**UNIVERSIDADE FEDERAL DE ALAGOAS**

**INSTITUTO DE COMPUTAÇÃO**

**COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA**


Dissertação Mestrado

# Proposta de modelo de previsão de estadiamento em pacientes diagnosticados com PDAC

Fabiano Santos Conrado
fsc2@ic.ufal.br

Orientador:
Dr. Rafael de Amorim Silva

Maceió
Agosto 29, 2022

Fabiano Santos Conrado

# Proposta de modelo de previsão de estadiamento em pacientes diagnosticados com PDAC

Dissertação apresentada ao curso de Mestrado em Informática do Programa de Pós Graduação em Informática da Universidade Federal de Alagoas, como requisito para obtenção do grau de Mestre em informática.

Orientador:

Dr. Rafael de Amorim Silva

Maceió
Agosto 29, 2022

**Catalogação na Fonte**
**Universidade Federal de Alagoas**
**Biblioteca Central**
**Divisão de Tratamento Técnico**

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

C754p Conrado, Fabiano Santos.
Proposta de modelo de previsão de estadiamento em pacientes diagnosticados com PDAC / Fabiano Santos Conrado. – 2022.
116 f. : il.

Orientador: Rafael de Amorim Silva.
Dissertação (mestrado em informática) - Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2022.

Bibliografia: f. 67-72.
Apêndices: f. 73-116.

1. Neoplasias pandreáticas - Diagnóstico. 2. Carcinoma ductal pancreático. 3. Estadiamento de neoplasias. 4. Biomarcadores. 5. Aprendizado de máquina. 6. KNN (Algoritmo). 7. Máquinas de vetor de suporte (Algoritmo). 8. Floresta aleatória (Algoritmo). 9. Neoplasias. 10. Urina. 11. LYVE1 (Biomarcadores). 12. Litostatina. 13. Fator trefoil-1. 14. Creatinina. 15. Antígeno CA-19-9. I. Título.

CDU: 004.81:159.953.5:616-006.6

UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
**Programa de Pós-Graduação em Informática – PPGI**
**Instituto de Computação/UFAL**
Campus A. C. Simões BR 104-Norte Km 14 BL 12 Tabuleiro do Martins
Maceió/AL - Brasil CEP: 57.072-970 | Telefone: (082) 3214-1401

## Folha de Aprovação

FABIANO SANTOS CONRADO

PROPOSTA DE MODELO DE PREVISÃO DE ESTADIAMENTO EM PACIENTES DIAGNOSTICADOS COM PDAC

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática da Universidade Federal de Alagoas e aprovada em 29 de agosto de 2022.

**Banca Examinadora:**

_____
**Prof. Dr. RAFAEL DE AMORIM SILVA**
UFAL – Instituto de Computação
**Orientador**

_____
**Prof. Dr. BRUNO ALMEIDA PIMENTEL**
UFAL – Instituto de Computação
**Examinador Interno**

_____
**Prof. Dr. ALMIR PEREIRA GUIMARÃES**
UFAL – Instituto de Computação
**Examinador Externo**

# Agradecimentos

Durante esses anos de mestrado, não foi fácil alcançar o fim desta trajetória repleta de inúmeros percalços, tristezas, incertezas, esperanças, alegrias e muitos outros sentimentos que as palavras não são suficientes para expressar. Todavia, apesar do processo solitário ao qual qualquer pesquisador está destinado, existem pessoas que contribuíram de forma direta ou indireta, dando o suporte necessário para vencer cada obstáculo.

Trilhar esse caminho só foi possível com o apoio, energia e força dessas pessoas, a quem dedico especialmente este projeto de vida.

A Deus, pela dádiva da vida e por Seu imensurável amor, te amo.

Ao meu orientador, Prof. Dr. Rafael de Amorim Silva, que sempre acreditou em mim e aceitou ser meu orientador mesmo com um prazo tão curto, agradeço pela orientação pautada em elevado rigor científico, empenho exemplar e saudável exigência, estando dedicado e presente em cada passo dessa caminhada.

À minha esposa, Roseane Conrado, pelo amor, companheirismo e apoio incondicional. Agradeço pela enorme compreensão nos momentos em que não pude estar presente, pela generosidade e alegria que me acompanharam constantemente, contribuindo para que eu chegasse ao fim dessa caminhada. Sem ela, essa jornada teria sido ainda mais longa.

Ao meu filho, Felip Conrado, pela alegria que trouxe às nossas vidas, um presente iluminado que sempre me motivou nos momentos mais difíceis. Sua felicidade sempre aqueceu meu coração. Perdoe-me pelos momentos em que não pude estar presente; vou recompensar cada um deles.

À minha mãe, Josefa Conrado, por ter escolhido me amar; sem ela, não sei o que seria de mim. À minha irmã, Maria Conrado, e à minha sobrinha, Camila Beatriz, que com muita alegria, amor e carinho zelaram pelo bem do meu filho quando foi necessário. Foram fundamentais nessa caminhada.

Agradeço, em memória, ao meu pai, Cícero Conrado, que tanto me ensinou, e ao meu sobrinho, Carlos André, que sempre foi uma inspiração de vencedor.

Aos meus coordenadores do Núcleo de Tecnologia da Informação, Reinaldo Cabral e Bruno César, pela motivação, compreensão e apoio.

Agradeço aos professores Dr. Ig Ibert Bittencourt, Dr. Geiser Chalco Challco, Dr. Alan Pedro da Silva e Dr. Diego Dermeval Medeiros da Cunha Matos, representando os demais professores e servidores do PPGI, que, com dedicação e qualidade, contribuíram generosamente no meu processo de metamorfose, tornando-me um pesquisador.

Agradeço também aos alunos Marcos Bento, Victor Holanda, Ítalo Arruda, Lucas Cezar e Willian da Silva, pelo companheirismo e incentivo prestados durante essa jornada.

Por fim, agradeço aos professores Dr. Almir Pereira Guimarães Dr. Bruno Almeida Pimentel pela avaliação deste trabalho.

*"Porque a loucura de Deus é mais sábia que a sabedoria humana, e a fraqueza de Deus é mais forte que a força do homem."*

*Tarso, Paulo de.*

# Resumo

*O câncer de pâncreas (CP) é de difícil diagnóstico precoce, uma vez que evolui de forma silenciosa, sem apresentar sinais específicos, e responde mal à maioria dos tratamentos. Noventa por cento dos casos de CP são do tipo adenocarcinoma ductal pancreático (PDAC), e a sobrevida global em cinco anos após o diagnóstico é de apenas 12,8%. Esse baixo índice leva os pacientes diagnosticados a questionarem quanto tempo lhes resta de vida. O sistema de classificação TNM para tumores malignos tem sido o método mais comum para avaliar a sobrevida e apoiar a tomada de decisão médica em relação a intervenções curativas ou paliativas. Entretanto, essa classificação só pode ser realizada após exames de imagem avançados, exigindo que os pacientes se submetam a novos testes para monitorar alterações no estadiamento. Nem todos os pacientes dispõem de recursos, disponibilidade física e/ou emocional para reavaliações constantes. Dada a alta taxa de mortalidade e a dificuldade na detecção dessa neoplasia, diversas pesquisas têm surgido em busca de biomarcadores para um diagnóstico precoce. No entanto, poucos desses trabalhos focam no desenvolvimento de métodos para prognósticos prévios. Esta pesquisa propõe e avalia um modelo de prognóstico prévio de estadiamento para PDAC com base em biomarcadores urinários utilizados no diagnóstico de PDAC, combinados com idade e sexo. Para isso, foram coletados dados de vários centros de saúde e analisados utilizando técnicas de Aprendizado de Máquina (Machine Learning, ML). As técnicas adotadas para a classificação prévia dos estadiamentos foram K-Nearest Neighbors (KNN), Support Vector Machines (SVM) e Random Forest.* ***Resultados encontrados:*** *O classificador KNN alcançou uma acurácia máxima de 0,62, o SVM atingiu uma acurácia de 0,58 e o Random Forest apresentou os melhores resultados, com acurácia de 0,81. Isso indica que o uso de biomarcadores para a classificação prévia de estadiamento pode auxiliar na tomada de decisão médica e no monitoramento da progressão da neoplasia.*

***Palavras-chave****: Câncer de Pâncreas, diagnóstico, Adenocarcinoma Ductal Pancreático, estadiamento, Classificação de Tumores Malignos, biomarcadores, Machine Learning, KNN, SVM, Random Forest, PDAC, TNM, neoplasia, urina, LYVE1, REG1B, TFF1, creatinina, plasma CA199, REG1A.*

# Abstract

*Pancreatic cancer (PC) is difficult to diagnose early because it progresses silently, without specific symptoms, and responds poorly to most treatments. Ninety percent of pancreatic cancer cases are pancreatic ductal adenocarcinoma (PDAC), and the overall five-year survival rate after diagnosis is only 12.8%. This low survival rate leads diagnosed patients to question how much time they have left. The TNM classification system for malignant tumors has been the most common method for assessing survival and supporting medical decision-making regarding curative or palliative interventions. However, this classification can only be performed after advanced imaging exams, requiring patients to undergo new imaging tests to monitor changes in staging. Not all patients have the resources, physical availability, or emotional capacity for constant re-evaluation. Given the high mortality rate and difficulty in detecting this neoplasm, several research studies have emerged in search of biomarkers for early diagnosis. However, few of these studies focus on developing methods for early prognosis. This research proposes and evaluates a prior staging prognosis model for PDAC based on urinary biomarkers used for PDAC diagnosis, combined with age and gender. To this end, data from various health centers were collected and analyzed using Machine Learning (ML) techniques. The adopted techniques for prior staging classification were K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Random Forest.* ***Results:*** *The KNN classifier achieved a maximum accuracy of 0.62, SVM reached an accuracy of 0.58, and Random Forest produced the best results with an accuracy of 0.81. This indicates that the use of biomarkers for prior staging classification can assist in medical decision-making and monitoring the progression of the neoplasm.*

***Keywords****: Pancreatic Cancer, Diagnosis, Pancreatic Ductal Adenocarcinoma, Staging, Malignant Tumor Classification, Biomarkers, Machine Learning, KNN, SVM, Random Forest, PDAC, TNM, Neoplasm, Urine, LYVE1, REG1B, TFF1, Creatinine, Plasma CA199, REG1A.*

# Lista de Figuras

# Lista de Tabelas

# Sumário

# 1

# Introdução

## 1.1 Contextualização

*Nos seres humanos, o Pâncreas é um órgão de aproximadamente 12 a 25cm (Sulochana and Sivakami, 2012), localizado no abdômen, atrás do estômago, entre o duodeno e o baço. Apesar de seu pequeno tamanho, o pâncreas é ricamente irrigado por artérias e veias. Ele é uma glândula mista, possuindo duas funções: (i) Uma endócrina cuja principal função é a regulação da glicose no sangue, produzindo insulina para baixar os níveis de glicose e glucagon para subir o nível da glicose (i.e. o pâncreas está estritamente ligado a Diabetes Mellitus (DM)); (ii) A função exócrina do pâncreas é responsável na produção das enzimas: Lipase (que quebra as gorduras), a protease (que quebra as proteínas) e a amilase (que quebra os carboidratos), auxiliando no processo de ingestão de alimentos e transformando-os em nutrientes, ver Figura 1.1.*

*Dado o tamanho, localização e funções do pâncreas as doenças associadas a ele são de percepção tardia, sendo possível conviver anos com elas até o surgimento de sintomas mais graves, dentre os principais problemas pancreáticos o Câncer de Pâncreas (CP) se destaca, pois, embora raro quando ocorre, é difícil diagnóstico prévio, evolui de maneira silenciosa e sem apresentar sinais específicos. Além disso, tem uma baixa resposta a maioria dos tratamentos, sua taxa de sobrevida global em 5 anos é de 12.8% (Cancer Institute, 2022) após serem diagnosticados, tendo uma maior sobrevida para aqueles pacientes que não apresentam doença metastática. Segundo organização mundial da saúde somente em 2020, surgiram 495.773 (quatrocentos e noventa e cinco mil, setecentos e setenta e três) novos casos e 466.003 (qua-*

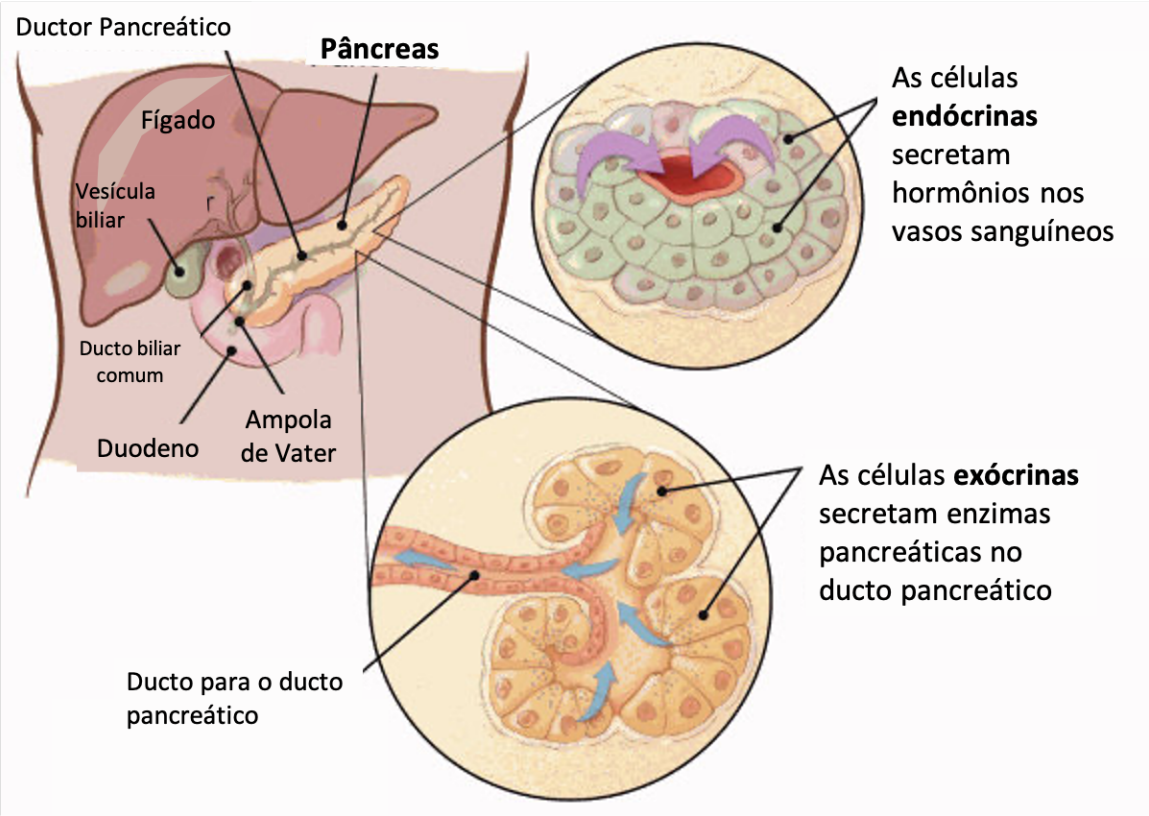



Figura 1.1: Localização do pâncreas e visualização das células endócrinas e exócrinas, imagem traduzida (Society, 2022)

*trocentos e sessenta e seis mil e três) casos de mortalidade, de câncer de pâncreas em todo mundo, (World Cancer Research, 2022).*

*As tendências de incidência e mortalidade por CP variam consideravelmente no mundo. Uma causa conhecida de câncer de pâncreas é o tabagismo. Este fator de risco provavelmente explica algumas das variações internacionais e diferenças de gênero (A incidência é mais significativa no sexo masculino). A taxa de sobrevida varia pouco entre países desenvolvidos e em desenvolvimento. Até o momento, as causas do CP ainda são insuficientemente conhecidas, embora alguns fatores de risco tenham sido identificados além do tabagismo, tais como: obesidade, genética, diabetes, dieta, sedentarismo. Não há recomendações atuais de rastreamento para câncer de pâncreas, portanto, a prevenção primária é de extrema importância. (Kenner et al., 2021). O CP é raro antes dos 30 anos, tornando-se mais comum a partir dos 60 anos. Segundo a União Internacional para o Controle do Câncer (UICC), os casos de câncer de pâncreas aumentam com o avanço da idade: de 10/100.000 habitantes entre 40 e 50 anos para 116/100.000 habitantes entre 80 e 85 anos (INCA, 2022), ou seja, embora o CP tenha uma causa variada, o grupo de maior risco são os idosos.*

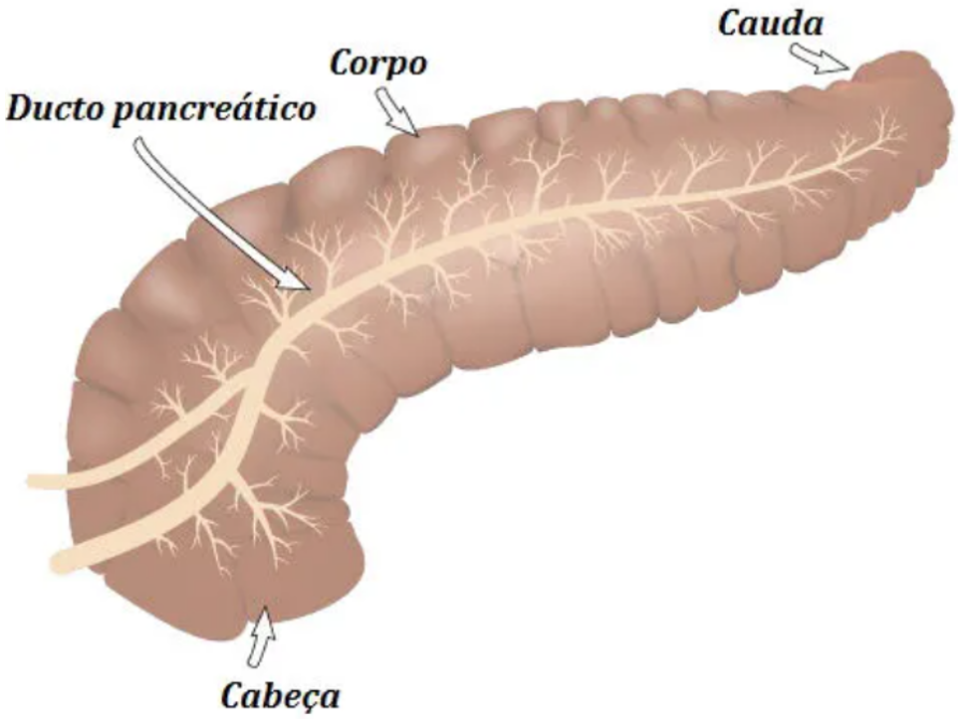



Figura 1.2: Principais divisões do pâncreas: Cabeça, Corpo e Cauda, (dos Santos, 2022)

## 1.2 Problemática

*As previsões prognósticas e as estratégias de tratamento para pacientes com CP são baseadas no sistema de estadiamento TNM(Classificação de Tumores Malignos) padrão globalmente reconhecido para classificar a extensão de disseminação do câncer, sendo desenvolvido e mantido pela União Internacional para o Controle do Câncer (UICC) sendo também utilizado pela American Joint Committee on Cancer (AJCC) e pela Federação Internacional de Ginecologia e Obstetrícia (INCA, 2022). Esse sistema avalia o câncer pela extensão anatômica da doença com base na profundidade da invasão, número de linfonodos em metástase e status de metástase à distância (denominado estágio). Este estágio vem se tornando cada vez mais um componente de importância de vigilância e controle do câncer, logo, um ponto final para a avaliação da triagem populacional, sendo amplamente utilizado para predizer a sobrevida de pacientes com câncer (UICC).*

*Todavia, a sobrevida pode ser diferente em pacientes com o mesmo estágio TNM. De fato, outros fatores do paciente como biomarcadores de câncer específicos, idade, raça ou estado civil também podem estar associados a sobrevida em vários tipos de câncer. Portanto, é necessário um sistema de estadiamento que combine às características do tumor e o status do paciente, visando uma maior acurácia ou ainda o uso desses dados clínicos para um prévio*

*estadiamento, tendo em vista, o longo processo para se obter o TNM inicial e respectivo acompanhamento. Com o avanço do poder computacional e do aprimoramento das técnicas de Inteligência Artificial (IA), em especifico da aprendizagem de Máquina (Machine Learning (ML)), o diagnóstico apoiado por ML promete revolucionar a saúde(Richens et al., 2020), fazendo uso do grande volume de dados disponíveis do paciente visando fornecer diagnósticos precisos e personalizados.*

## 1.3 Proposta

*Esta pesquisa* ***propõe e avalia um modelo para predição do estadiamento em pacientes diagnosticados com câncer de adenocarcinoma ductal pancreático (PDAC)****, principal tipo de câncer de pâncreas, com base na classificação dos graus do Score do TNM como referencia. Este modelo considera informações como idade, sexo e os biomarcadores que vêm sendo estudados para diagnóstico para predizer o estadiamento dos pacientes. Consequentemente, gera-se um estimador prévio baseado em exames não invasivos, não nocivos e de baixo custo. O modelo visa servir de apoio a decisão médica.*

*Para isso serão utilizadas a técnica de ML: (I) K-ésimo Vizinho mais Próximo (k-nearest neighbors algorithm – KNN), um método de aprendizado supervisionado não paramétrico que utiliza a correlação com seus vizinhos mais próximos, sendo utilizado tanto para classificação como para regressão de dados, com inúmeros usos aplicados no prognostico de doença (Ow and Kuznetsov, 2016; Parry et al., 2010); (II) Floresta Aleatória (Random Forest), um método de aprendizado conjunto para classificação e regressão que opera construindo árvores de decisão em tempo de treinamento (Breiman, 2001); e (III) Máquinas de Vetor de Suporte (Support-Vector Machine - SVM), um método de aprendizado que utiliza um conjunto de modelos matemáticos de aprendizado supervisionado com algoritmos de aprendizado associados que analisam dados para classificação e análise de regressão (Cortes and Vapnik, 1995).*

## 1.4 Estrutura de Trabalho

*O restante desta dissertação está organizado da seguinte forma: Capítulo 2 apresenta a fundamentação teórica necessária para a compreensão deste trabalho. O Capítulo 3 descreve a problemática a ser resolvida relacionada ao prognóstico de pacientes com câncer de pâncreas. O Capítulo 4 apresenta a relevância e descrição do modelo proposto neste trabalho. O Capítulo*

*5 define os aspectos utilizados para a elaboração do experimento. O Capítulo 6 apresenta e discute os resultados obtidos no experimento. O capítulo 7 apresenta as considerações finais.*

# 2

# Fundamentação Teórica

*O objetivo deste capítulo é apresentar a fundamentação teórica referente ao foco deste trabalho, fazendo uma pequena introdução sobre conceitos importantes da área que auxiliam na compreensão da pesquisa desenvolvida.*

## 2.1 Pâncreas

*O pâncreas é um órgão que faz parte do sistema digestivo e sistema endócrino dos vertebrados, sendo uma glândula mista ou heterócrina que atua de forma exócrina (99%) e endócrina (1%). Sua função exócrina é vinculada à secreção de suco pancreático, que contém enzimas digestivas. Já sua função endócrina atua na produção de diversos hormônios importantes, tais como: insulina, glucagon, amilina e somatostatina. Este órgão é um produtor de enzimas e proteínas que aumenta a rapidez nas transformações químicas.*

### 2.1.1 Anatomia

*O pâncreas mede entre 12 e 25 cm de comprimento e entre 3 e 5 cm de altura (Sulochana and Sivakami, 2012), com peso entre 60 g e 170 g. É um órgão retroperitoneal, lobular e dividido comumente em três partes: cabeça (proximal), corpo e cauda (distal), conforme ilustrado na Figura 1.2. A primeira parte encontra-se junto ao duodeno e a última parte está em contato com o hilo esplênico e a flexura cólica esquerda, como mostrado na Figura 2.1.*

*O canal de Wirsung é um ducto excretório que acompanha toda a extensão do pâncreas. Ele se conecta ao duodeno através da ampola de Vater, onde se junta ao ducto biliar. O esfíncter de Oddi, juntamente com a ampola de Vater, regula a secreção pancreática no trato gastrointestinal, como mostrado na Figura 2.2.*

***Artérias e veias***

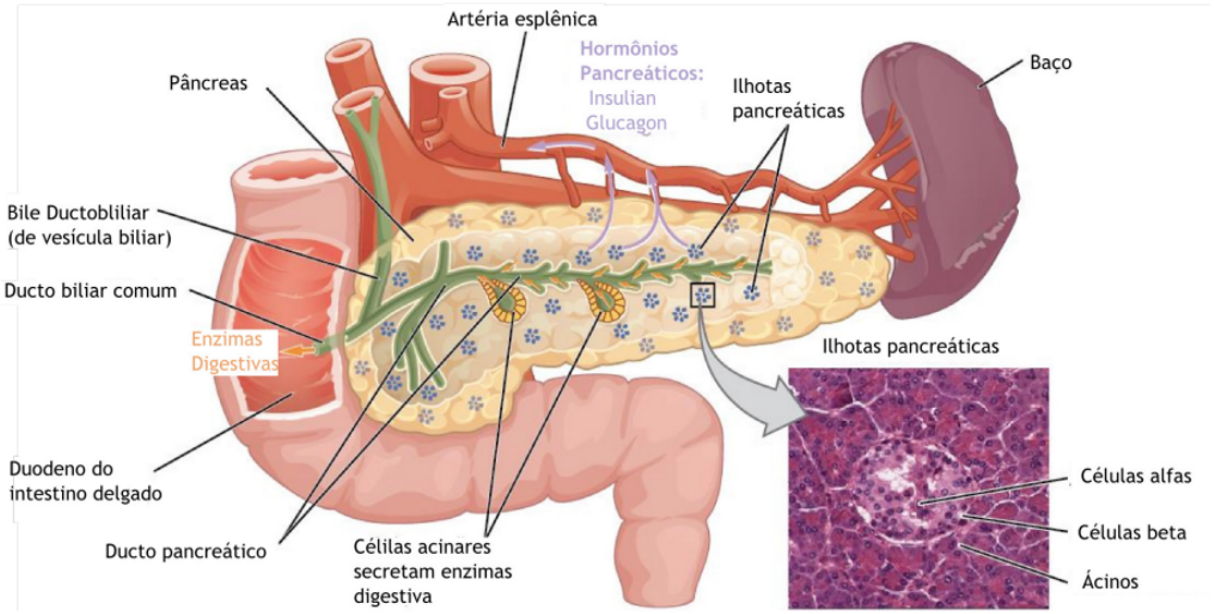



Figura 2.1: Estrutura anatômica do pâncreas.

*O pâncreas é suprido pelas artérias pancreaticoduodenais: A artéria mesentérica superior que origina as artérias pancreaticoduodenais inferiores, a artéria gastroduodenal que origina as artérias pancreaticoduodenais superiores, a artéria esplênica que origina as artérias pancreáticas.*

*A drenagem venosa é feita através das veias pancreáticas que são tributárias das veias esplênica e mesentérica superior, no entanto a maioria delas terminam na veia esplênica. A veia porta hepática é formada pela união da veia mesentérica superior e veia esplênica posteriormente ao colo do pâncreas. Assim, o fígado se torna exposto a altas concentrações dos hormônios pancreáticos, sendo o principal órgão-alvo dos seus efeitos fisiológicos. Geralmente a veia mesentérica inferior se une à veia esplênica atrás do pâncreas.*

***Microanatomia***

*A maioria do tecido pancreático tem um papel digestivo (cerca de 95%). As células com*

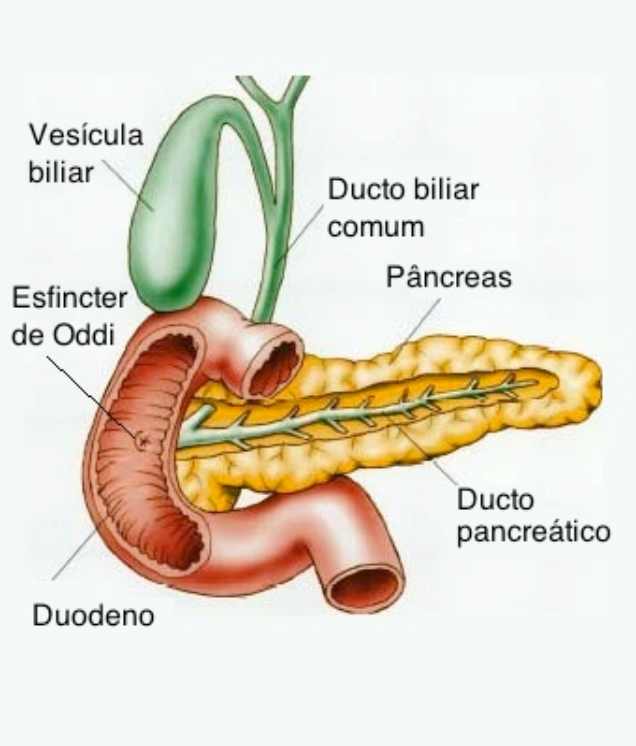
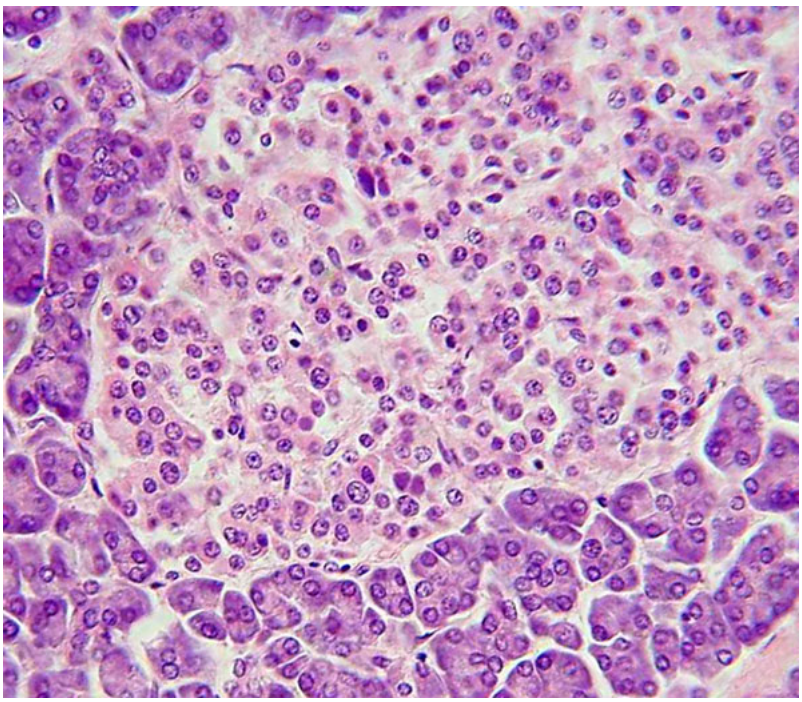



Figura 2.2: Esfincter de Oddi

*esse papel formam aglomerados ao redor de pequenos ductos e estão dispostas em lobos que possuem paredes fibrosas finas. As células de cada ácino secretam enzimas digestivas inativas, chamadas zimogênios, nos pequenos ductos intercalados ao seu redor (Medicine, 2021a). Em cada ácino, as células são em forma de pirâmide e situadas ao redor dos ductos intercalares, com os núcleos apoiados na membrana basal, um grande retículo endoplasmático e vários grânulos de zimogênio visíveis no citoplasma (Jamieson, 2021). Os ductos intercalados drenam para ductos intralobulares maiores dentro do lóbulo e, finalmente, para ductos interlobulares. Os dutos são revestidos por uma única camada de células em forma de coluna. Há mais de uma camada de células à medida que o diâmetro dos ductos aumenta (Brelje and Sorenson, 2021).*

*Os tecidos com papel endócrino dentro do pâncreas existem como aglomerados de células chamadas ilhotas pancreáticas (também conhecidas como: ilhotas de Langerhans ) constituindo cerca de 1-2% essas ilhotas desempenham um papel crucial na regulação da glicose no sangue, por meio da secreção de hormônios e são distribuídas por todo o pâncreas (Britannica, 2021). As ilhotas pancreáticas contêm células alfa($\alpha$), células beta($\beta$) e células delta($\delta$), cada uma das quais libera um hormônio diferente. Estas células têm posições características, com as células alfa (secretoras de glucagon) tendendo a situar-se na periferia da ilhota, e as células beta (secretoras de insulina ) mais numerosas e encontradas em toda a ilhota (Learning, 2021). Células enterocromafins também estão espalhadas por todas as ilhotas.*

Figura 2.3: Ilhota pancreática quando o tecido pancreático é corado e visto ao microscópio



### 2.1.2 Função

*O pâncreas desempenha um papel fundamental tanto no sistema endócrino quanto no exócrino. Ele é responsável pela regulação do açúcar no sangue e pelo metabolismo geral do corpo. Essa regulação ocorre através da secreção de dois hormônios principais: insulina e glucagon. A insulina, secretada pelas células beta das ilhotas pancreáticas, reduz os níveis de glicose no sangue, promovendo a absorção de glicose pelas células. O glucagon, produzido pelas células alfa, tem o efeito oposto, aumentando os níveis de glicose ao estimular a liberação de glicose armazenada no fígado (Medicine, 2021b; Manual, 2021).*

*Além dessas funções endócrinas, o pâncreas também atua no processo digestivo através da secreção de suco pancreático, que contém enzimas digestivas como amilase, lipase e proteases. Essas enzimas são fundamentais para a digestão de carboidratos, lipídios e proteínas no intestino delgado (Manual, 2021; Pancreapedia, 2021). O suco pancreático também contém bicarbonato, que neutraliza o ácido gástrico no duodeno, protegendo as paredes intestinais (Manual, 2021).*

*Outro componente importante do pâncreas é a secreção de polipeptídeo pancreático (PP) e peptídeo intestinal vasoativo (VIP), ambos influenciando a motilidade intestinal e a secreção de fluidos no trato gastrointestinal (Pancreapedia, 2021).*

*As células enterocromafins estão presentes no pâncreas, embora não sejam a principal*

*fonte do hormônio motilina, que é geralmente secretado por células do intestino delgado. No entanto, essas células secretam serotonina e substância P, que são importantes para a regulação da motilidade intestinal e outras funções hormonais (Pancreapedia, 2021).*

***Pâncreas Exócrino***

*O pâncreas desempenha um papel vital no sistema digestivo, secretando um fluido que contém enzimas digestivas no duodeno, a primeira parte do intestino delgado que recebe o conteúdo gástrico do estômago. Essas enzimas são essenciais para a digestão de carboidratos, proteínas e lipídios (gorduras). Esse papel é denominado função "exócrina"do pâncreas. As células responsáveis por essa função estão organizadas em aglomerados chamados ácinos. As secreções produzidas nos ácinos se acumulam nos ductos intralobulares, que drenam para o ducto pancreático principal, o qual se conecta diretamente ao duodeno. Aproximadamente 1,5 a 3 litros de fluido pancreático são secretados diariamente Brock et al. (2003).*

*As células de cada ácino são preenchidas com grânulos que contêm enzimas digestivas. Essas enzimas são secretadas em sua forma inativa, denominadas zimogênios ou proenzimas. Quando liberadas no duodeno, são ativadas pela enzima enteropeptidase (anteriormente conhecida como enteroquinase), presente no revestimento intestinal. As proenzimas são clivadas, iniciando uma cascata de ativação enzimática que resulta na digestão eficaz dos nutrientes. (Boron and Boulpaep, 2009).*

- *As enzimas que quebram as proteínas começam com a ativação do tripsinogênio em tripsina. A tripsina livre então cliva o restante do tripsinogênio, bem como o quimotripsinogênio em sua forma ativa quimotripsina;*
- *As enzimas secretadas envolvidas na digestão de gorduras incluem lipase, fosfolipase A2, lisofosfolipase e colesterol esterase;*
- *As enzimas que quebram o amido e outros carboidratos incluem amilase.*

*Essas enzimas são secretadas em um fluido alcalino, rico em bicarbonato. O bicarbonato contribui para manter o pH alcalino no fluido, condição em que a maioria das enzimas digestivas atua de forma mais eficiente. Além disso, o bicarbonato ajuda a neutralizar os ácidos gástricos que entram no duodeno, protegendo o intestino delgado e permitindo a digestão adequada (Guyton and Hall, 2016).A secreção pancreática é influenciada por hormônios, incluindo secretina, colecistocinina e peptídeo intestinal vasoativo (VIP), bem como pela estimulação da acetilcolina liberada pelo nervo vago. A secretina é liberada pelas células do revestimento do*

*duodeno em resposta à presença de ácido gástrico. Juntamente com o VIP, ela aumenta a secreção de enzimas e bicarbonato. A colecistocinina é liberada pelas células Ito do revestimento do duodeno e jejuno, principalmente em resposta aos ácidos graxos de cadeia longa, e amplifica os efeitos da secretina (Guyton and Hall, 2016). Em nível celular, o bicarbonato é secretado pelas células centroacinares e ductais por meio de um cotransportador de sódio e bicarbonato, que atua devido à despolarização da membrana causada pelo regulador de condutância transmembrana da fibrose cística (CFTR). A secretina e o VIP aumentam a abertura do CFTR, resultando em maior despolarização da membrana e, consequentemente, em mais secreção de bicarbonato (Boron and Boulpaep, 2009; Guyton and Hall, 2016; Brock et al., 2003).*

*Uma variedade de mecanismos atua para garantir que a ação digestiva do pâncreas não resulte na digestão do próprio tecido pancreático. Esses mecanismos incluem a secreção de enzimas inativas (zimogênios), a liberação da enzima inibidora da tripsina, que neutraliza a tripsina ativa, as mudanças de pH associadas à secreção de bicarbonato, que promovem a digestão apenas quando o pâncreas é estimulado, e o fato de que o baixo teor de cálcio intracelular contribui para a inativação da tripsina (Leung, 2010; Gorelick and Jamieson, 2012).*

***Pâncreas Endócrino***

*As células do pâncreas desempenham um papel crucial na manutenção da homeostase glicêmica. As células que regulam os níveis de glicose no sangue estão localizadas dentro das ilhotas pancreáticas, que estão distribuídas por todo o pâncreas. Quando os níveis de glicose no sangue estão baixos, as células alfa secretam glucagon, que aumenta a glicose sanguínea. Quando os níveis de glicose no sangue estão elevados, as células beta secretam insulina, que promove a diminuição da glicose no sangue. As células delta das ilhotas também secretam somatostatina, a qual inibe a liberação de insulina e glucagon (James et al., 2018).*

*O glucagon atua para aumentar os níveis de glicose no sangue, promovendo a gliconeogênese (produção de glicose) e a glicogenólise (quebra do glicogênio em glicose) no fígado. Além disso, o glucagon diminui a captação de glicose pelas células adiposas e musculares. A liberação de glucagon é estimulada por baixos níveis de glicose ou insulina no sangue, bem como durante o exercício (Müller et al., 2017).A insulina atua para diminuir os níveis de glicose no sangue, facilitando sua captação pelas células, especialmente no músculo esquelético, e promovendo seu uso na síntese de proteínas, lipídios e carboidratos. A insulina é inicialmente sintetizada como um precursor chamado pré-pró-insulina. Esta é convertida em pró-insulina e,*

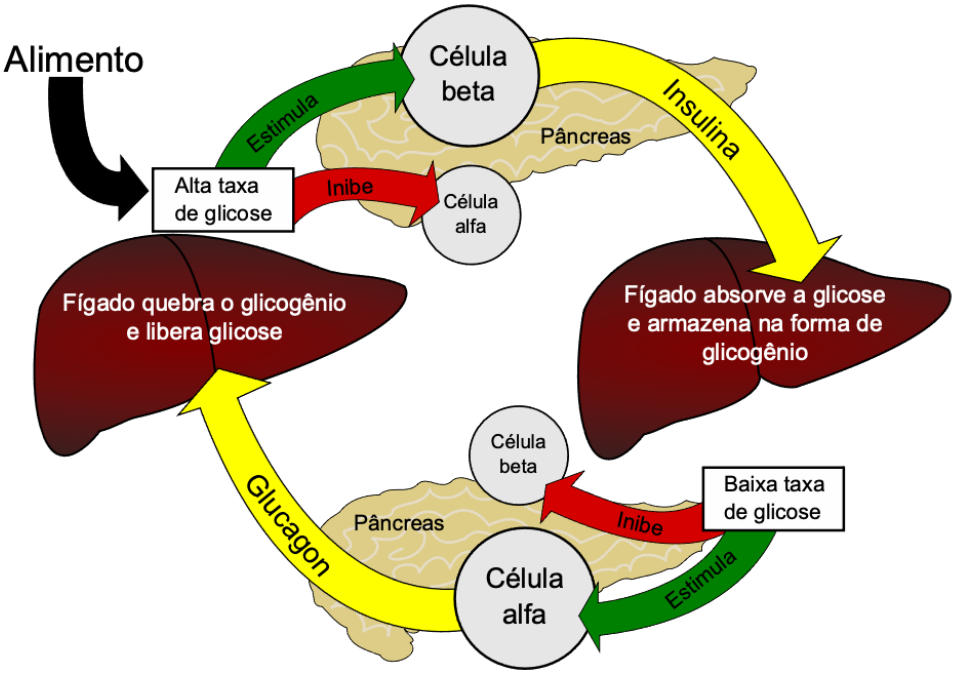



Figura 2.4: Processo de regulação da insulina

*em seguida, clivada pelo peptídeo C para formar a insulina ativa, que é armazenada em grânulos nas células beta. A glicose é transportada para as células beta e metabolizada, o que resulta na despolarização da membrana celular, estimulando a liberação de insulina (Ruiz et al., 2015).*

*O principal fator que influencia a secreção de insulina e glucagon são os níveis de glicose no plasma sanguíneo. Baixos níveis de glicose no sangue estimulam a liberação de glucagon, enquanto altos níveis estimulam a secreção de insulina. Outros fatores também influenciam a secreção desses hormônios. Alguns aminoácidos , que são subprodutos da digestão de proteínas , estimulam a liberação de insulina e glucagon. A somatostatina atua como um inibidor da insulina e do glucagon. O sistema nervoso autônomo também desempenha um papel. A ativação dos receptores beta-2 do sistema nervoso simpático, por meio das catecolaminas secretadas, estimula a secreção de insulina e glucagon, enquanto a ativação dos receptores Alfa-1 inibe a secreção. Os receptores M3 do sistema nervoso parassimpático agem quando estimulados pelo nervo vago direito para estimular a liberação de insulina das células beta.*

### 2.1.3 Doenças Pancreáticas

#### O câncer de pâncreas

*Surge quando as células do pâncreas começam a se multiplicar descontroladamente formando uma massa. Essas células tem a capacidade de invadir outras partes do corpo afetando seu funcionamento, se não tratável pode levar a morte.*

#### Câncer de Adenocarcinoma Ductal Pancreático (PDAC)

*O Câncer pancreático pode ser gerado tanto nas áreas responsáveis pelas duas principais funções do pâncreas: (i) região endócrina, (ii) região exócrina. Sendo a parte exócrina responsável 95% dos cânceres de pâncreas e o Adenocarcionama ductal pancreático (PDAC) sendo responsável por 90% do total, os cânceres da região endócrina costumam ter um prognóstico melhor. Existem várias categorias de cânceres, carcinoma são o tipo mais comum de câncer. Eles são formados por células epiteliais, que são as células que cobrem as superfícies internas e externas do corpo. Existem muitos tipos de células epiteliais, que muitas vezes têm uma forma de coluna quando vistas ao microscópio. Os carcinomas que começam em diferentes tipos de células epiteliais têm nomes específicos:*

- ***O adenocarcinoma** é um câncer que se forma nas células epiteliais que produzem fluidos ou muco. Tecidos com esse tipo de célula epitelial são às vezes chamados de tecidos glandulares. A maioria dos cânceres de mama, cólon e próstata são adenocarcinomas.*
- ***O carcinoma basocelular** é um câncer que começa na camada inferior ou basal (base) da epiderme, que é a camada externa da pele de uma pessoa.*
- ***O carcinoma de células escamosas** é um câncer que se forma em células escamosas, que são células epiteliais que se encontram logo abaixo da superfície externa da pele. As células escamosas também revestem muitos outros órgãos, incluindo o estômago, intestinos, pulmões, bexiga e rins. As células escamosas parecem planas, como escamas de peixe, quando vistas ao microscópio. Os carcinomas de células escamosas são às vezes chamados de carcinomas epidermoides.*
- ***O carcinoma de células de transição** é um câncer que se forma em um tipo de tecido epitelial chamado epitélio de transição ou urotélio. Este tecido, que é composto de muitas camadas de células epiteliais que podem ficar maiores e menores, é encontrado nos revestimentos da bexiga, ureteres e parte dos rins (pelve renal) e alguns outros ór-*

*gãos. Alguns cânceres da bexiga, ureteres e rins são carcinomas de células transicionais (Soldan, 2017).*

*O PDAC se forma no ducto pancreático, sendo o tipo de câncer com maior letalidade. Os sinais e sintomas da forma mais comum podem incluir pele amarelada, dor abdominal ou nas costas, perda de peso inexplicável, fezes claras, urina escura e perda de apetite. Geralmente, nenhum sintoma é observado nos estágios iniciais da doença, e os sintomas que são específicos o suficiente para sugerir câncer de pâncreas geralmente não se desenvolvem até que a doença atinja um estágio avançado. No momento do diagnóstico, o câncer de pâncreas já se espalhou para outras partes do corpo (BW and CP, 2014).*

## 2.2 Classificação de Tumores Malignos - TNM

*A Classificação TNM de Tumores Malignos (TNM) é um padrão globalmente reconhecido para classificar a extensão da disseminação do câncer. É um sistema de classificação da extensão anatômica dos cânceres tumorais. Ganhou ampla aceitação internacional para muitos cânceres de tumores sólidos, mas não é aplicável à leucemia e aos tumores do sistema nervoso central . Os tumores mais comuns têm sua própria classificação TNM. Às vezes também descrito como o sistema AJCC.*

*O TNM foi desenvolvido e é mantido pela União Internacional para o Controle do Câncer (UICC). Também é usado pelo American Joint Committee on Cancer (AJCC) e pela Federação Internacional de Ginecologia e Obstetrícia (FIGO). Em 1987, os sistemas de encenação UICC e AJCC foram unificados em um único sistema de encenação TNM. TNM é um sistema de notação que descreve o estágio de um câncer, que se origina de um tumor sólido, usando códigos alfanuméricos :*

- *T descreve o tamanho do tumor original (primário) e se ele invadiu o tecido próximo;*
- *N descreve os linfonodos próximos (regionais) que estão envolvidos;*
- *M descreve metástase à distância (disseminação do câncer de uma parte do corpo para outra).*

*O sistema de estadiamento TNM para todos os tumores sólidos foi idealizado por Pierre Denoix entre 1943 e 1952, usando o tamanho e extensão do tumor primário, seu envolvimento linfático e a presença de metástases para classificar a progressão do câncer. (Asare et al., 2019)*

### 2.2.1 Os Princípios do Sistema TNM

*A prática de se dividir os casos de câncer em grupos, de acordo com os chamados estádios, surgiu do fato de que as taxas de sobrevida eram maiores para os casos nos quais a doença era localizada do que para aqueles nos quais a doença tinha se estendido além do órgão de origem. Esses grupos eram frequentemente referidos como casos iniciais e casos avançados, inferindo alguma progressão regular com o passar do tempo. Na verdade, o estádio da doença, na ocasião do diagnóstico, pode ser um reflexo não somente da taxa de crescimento e extensão da neoplasia, mas também do tipo de tumor e da relação tumor-hospedeiro. O estadiamento do câncer é consagrado por tradição, e para o propósito de análise de grupos de pacientes é frequentemente necessário usar tal método. A UICC acredita que é importante alcançar a concordância no registro da informação precisa da extensão da doença para cada localização anatômica, porque a descrição clínica precisa e a classificação histopatológica das neoplasias malignas podem interessar a um número de objetivos correlatos, a saber:*

1. *Ajudar o médico no planejamento do tratamento;*
2. *Dar alguma indicação do prognóstico;*
3. *Ajudar na avaliação dos resultados de tratamento;*
4. *Facilitar a troca de informações entre os centros de tratamento;*
5. *Contribuir para a pesquisa contínua sobre o câncer humano.*

*O principal propósito a ser conseguido pela concordância internacional na classificação dos casos de câncer pela extensão da doença é fornecer um método que permita comparações entre experiências clínicas sem ambiguidade. Existem muitas bases ou eixos de classificação dos tumores, por exemplo: a localização anatômica e a extensão clínica e patológica da doença, a duração dos sinais ou sintomas, o gênero e idade do paciente, o tipo e grau histológico. Todas essas bases ou eixos representam variáveis que, sabidamente, têm uma influência na evolução da doença. O sistema TNM trabalha prioritariamente com a classificação por extensão anatômica da doença, determinada clínica e histopatologicamente (quando possível). A primeira tarefa do clínico é fazer uma avaliação do prognóstico e decidir qual o tratamento mais efetivo a ser realizado. Este julgamento e esta decisão requer, entre outras coisas, uma avaliação objetiva da extensão anatômica da doença. Isto feito, a tendência é divergir do estadiamento, quanto a uma descrição significativa, com ou sem alguma forma de sumarização. Para conseguir os objetivos estabelecidos, um sistema de classificação necessita que: 1. Os princípios básicos sejam aplicáveis a todas as localizações anatômicas, independentemente do tratamento; e 2. Possa ser complementado, mais tarde, por informações que se tornem disponíveis pela histopatologia e/ou cirurgia.*

### 2.2.2 Estágios tradicionais

*Os estágios (ou estádios) são numerados de 0 a IV de acordo com a facilidade de remover em cirurgia com sucesso:*

- ***Estágio 0:*** *Carcinoma in situ, ou seja, restritos a área inicial. É um tipo de displasia.*
- ***Estágio I:*** *Tumor restrito a uma parte do corpo, sem comprometimento linfático.*
- ***Estágio II:*** *Localmente avançado com comprometimento do sistema linfático ou espalhado por mais de um tecido.*
- ***Estágio III:*** *Localmente avançado, espalhado por mais de um tecido e causando comprometimento linfático.*
- ***Estágio IV:*** *Metástase a distância, ou seja, espalhando para outros órgãos ou todo o corpo.*

*A designação como estágio II ou estágio III pode depender do tipo específico de câncer. Por exemplo, na doença de Hodgkin, o estágio II indica linfonodos afetados em apenas um lado do diafragma, enquanto o Estágio III indica linfonodos afetados acima e abaixo do diafragma. Os critérios específicos para os estágios II e III, portanto, diferem de acordo com o diagnóstico.*

*Um tratamento curativo ou paliativo dependerá da situação do tumor. Até o estágio III é geralmente possível remover o câncer completamente via quimioterapia, radioterapia ou cirurgia, o que é entendido como uma cura, porém a partir do estágio IV o tratamento se restringe a promover o mínimo de sintomas, o máximo de sobrevida e a melhor qualidade de vida ao paciente sendo a cura altamente improvável.*

### 2.2.3 Regras Gerais do Sistema TNM

*A classificação TNM compreende algoritmos de estadiamento para quase todos os cânceres, com exceção primária dos cânceres pediátricos. O esboço geral para a classificação TNM está abaixo. Os valores entre parênteses fornecem um intervalo do que pode ser usado para todos os tipos de câncer, mas nem todos os cânceres usam esse intervalo completo.*

***Parâmetros obrigatórios:***

- ***T**: Tamanho ou extensão direta do tumor primário*
  - ***Tx:** Tumor não pode ser avaliado*
  - ***Tis:** Carcinoma in situ*
  - ***T0:** Sem evidência de tumor*
  - ***T1, T2, T3, T4:** Tamanho e/ou extensão do tumor primário*
- ***N:** Grau de disseminação para linfonodos regionais*
  - ***Nx:** Os linfonodos não podem ser avaliados*
  - ***N0:** Sem metástase em linfonodos regionais*
  - ***N1:** Presença de metástase em linfonodo regional; em alguns locais, o tumor se espalhou para o número mais próximo ou pequeno de linfonodos regionais*
  - ***N2:** Tumor se espalhou entre N1 e N3 (N2 não é usado em todos os locais)*
  - ***N3:** Disseminação do tumor para linfonodos regionais mais distantes ou numerosos (N3 não é usado em todos os locais)*
- ***M:** presença de metástase à distância*
  - ***M0:** sem metástase à distância*
  - ***M1:** metástase para órgãos distantes (além dos linfonodos regionais).*

*A designação Mx, foi removida da 7ª edição do sistema AJCC/UICC, mas se referia a cânceres que não podiam ser avaliados para metástase à distância (O sistema está na 8ª Edição lançada em dezembro/2016) .*

*O sistema TNM é usado para registrar a extensão anatômica da doença. É útil condensar essas categorias em grupos. O carcinoma in situ é classificado como estágio 0; muitas vezes os tumores localizados no órgão de origem são estadiados como I ou II, dependendo da extensão, disseminação localmente extensa, para linfonodos regionais são estadiados como III, e aqueles com metástase à distância são estadiados como estágio IV. No entanto, em alguns tipos de tumores, os grupos de estágio não estão em conformidade com esse esquema simplificado. O grupo de estágios é adotado com a intenção de que as categorias dentro de cada grupo sejam mais ou menos homogêneas em relação à sobrevivência, e que as taxas de sobrevivência sejam distintas entre os grupos. A União Internacional para o Controle do Câncer (UICC) usa o termo Estágio para definir a extensão anatômica da doença. O American Joint Committee on Cancer (AJCC) usa o termo Prognostic Stage Group, que também pode incluir fatores prognósticos adicionais, além da extensão anatômica da doença.*

### 2.2.4 Regras do Sistema TNM para o câncer de pâncreas

*Apesar das definições das regras do sistema para todo os tipos de câncer de forma geral, o sistema de estadiamento TNM também definiu as regras de classificação especificas de cada neoplasia, conforme é possível observar na tabela 2.1.*

| Definições da 8ª edição do sistema de estadiamento TNM | |
|---|---|
| T1 | Tumor $\leq$ 2cm na maior dimensão |
| T1a | Tumor $\leq$ 0.5cm na maior dimensão |
| T1b | Tumor $\leq$ 0.5cm e > 1cm na maior dimensão |
| T1c | Tumor 1-2cm na maior dimensão |
| T2 | Tumor > 2cm e $\leq$ 4 na maior dimensão |
| T3 | Tumor > 4cm na maior dimensão |
| T4 | Tumor envolve CA, SMA e ou CHA, independente do tamanho |
| N1 | Metástase em 1-3 nódulos tumorais regionais |
| N2 | Metástase em 4 $\geq$ nódulos tumorais regionais |

Tabela 2.1: Resumo Esquemático: CA = Indica o eixo celíaco (Celiac Axis), SMA = Indica artéria mesentérica superior( superior mesenteric artery ), CHA = Indica artéria hepática comum (common hepatic artery

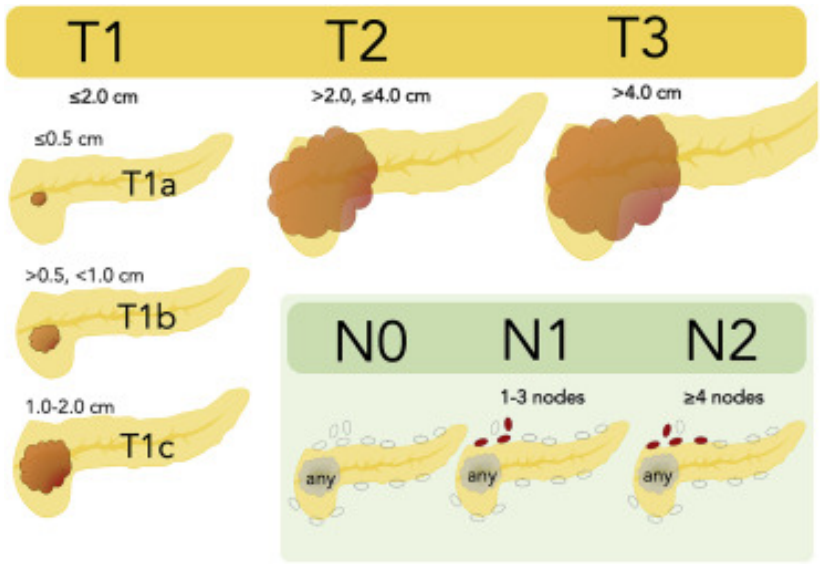



Figura 2.5: **Ilustração dos estágios T e N atuais:** Critérios baseados em tamanho para estágios T com subestadiamento para T1 nas categorias T1a, T1b e T1c. O estágio N é baseado em diferenças numéricas nos linfonodos metastáticos , com 3 linfonodos como ponto de corte.

## 2.3 Biomarcadores

*Um marcador biológico é uma característica objetivamente mensurada e avaliada como um indicador de processo biológico normal, processos patogênicos, ou de respostas farmacológicas*

*a uma intervenção terapêutica. Na área da saúde existe uma diversidade de biomarcadores conhecidos: fisiológicos, bioquímicos, histológicos e anatômicos. Podem ser células específicas, DNA e RNA livres de células, peptídeos, proteínas, enzimas, metabólitos e hormônios. Entre esses, os mais relevantes são os biomarcadores bioquímicos, por causa da relativa facilidade em obtenção a partir de fluidos corporais. A natureza dinâmica do sistema circulatório e seus constituintes reflete diversos estados fisiológicos ou patológicos, e a facilidade com que o sangue pode ser coletado faz com que seja uma escolha lógica para aplicações de biomarcadores (Oliveira, 2020).*

#### Classificação dos biomarcadores

*Segundo (Shaw et al., 2015) Os biomarcadores possuem 4 classificações, são elas:*

- ***Biomarcadores de Diagnóstico:*** *permitem a detecção precoce de um câncer de forma não invasiva.*
- ***Biomarcadores de Prognóstico:*** *é uma característica clínica ou biológica que fornece informações sobre o curso provável da doença.*
- ***Biomarcadores Preditivos:*** *é um parâmetro que pode ser usado para predizer o resultado diferencial a uma terapia ou de algum tratamento especial.*
- ***Biomarcadores Terapêuticos:*** *é geralmente uma proteína que pode ser usada como alvo para uma terapia.*

*A função primordial do sistema urinário é a excreção de compostos tóxicos para o nosso organismo, compostos que são gerados a partir do sangue. Portanto, a urina pode ser uma rica fonte de biomarcadores de processos patogênicos. Por exemplo, (Debernardi et al., 2020) utilizou Biomarcadores urinários no diagnóstico prévio do PDAC.*

### 2.3.1 Biomarcadores x PDAC

#### CA 19-9

*Antígeno Sérico de Carboidratos (Serum carbohydrate antigen) ou ainda Antígeno de Carboidrato 19-9 foi isolado a mais de 30 anos, sendo o biomarcador mais conhecido do PDAC. Extensas pesquisas sobre as funções biológicas dos ácidos siálicos mostraram que o CA19-9 está*

*envolvido no crescimento e diferenciação celular, transdução de sinal, apoptose, espermatogênese e imunomodulação. Especificamente, foi relatado que CA19-9 propaga o recrutamento de leucócitos mediando a adesão e migração de leucócitos para um foco inflamatório. Suspeita-se que CA19-9 desempenhe um papel vital na adesão de células malignas às células endoteliais, transmigração e desenvolvimento de metástases. Isso é apoiado por observações in vivo de pacientes com câncer de pâncreas, câncer colorretal e câncer de mama, em que CA19-9 elevado está associado à progressão da doença e resultados inferiores.*

*CA19-9 tem sido extensivamente estudado em câncer de pâncreas desde sua descoberta nas últimas três décadas. Como marcador tumoral, o CA19-9 continua a ser solicitado rotineiramente pelos médicos durante o tratamento do câncer de pâncreas. Embora o teste seja amplamente acessível, é pertinente reconhecer as armadilhas potenciais para garantir a utilidade adequada e a interpretação precisa de quaisquer anormalidades. Há uma série de fatores que podem confundir a interpretação do CA19-9. Primeiro, até 20% da população tem uma deficiência hereditária de fucosiltransferase e não expressa CA19-9. Segundo, CA19-9 não é exclusivamente específico para câncer de pâncreas e inúmeras etiologias benignas podem falsamente elevar CA19-9. Em geral, CA19-9 pode ser usado no cenário clínico com cautela e como adjuvante de outras investigações (Tabela 1). CA19-9 no rastreamento de câncer de pâncreas é inadequado. No entanto, a utilidade do CA19-9 com outras modalidades de imagem pode conferir vantagens de detecção no subconjunto da população com fatores de alto risco. Atualmente, um CA19-9 pré-operatório elevado deve levar o clínico a reavaliar a ressecabilidade. Dada a especificidade limitada do CA19-9, imagens adicionais com ressonância magnética e EUS podem delinear melhor a vasculatura, anatomia e extensão da doença, com o objetivo de orientar a ressecabilidade e reduzir a carga de laparotomias desnecessárias. Apesar dessas insuficiências, há um papel do CA19-9 no prognóstico. Estudos têm demonstrado que a redução dos níveis pós-operatórios está associada a maior sobrevida. Um nível elevado de CA19-9 pré-operatório está associado a achados patológicos ruins, como linfonodos positivos e doença de alto grau histológico e sobrevida significativamente reduzida. Da mesma forma, uma falha na normalização do CA19-9 dentro de 3 a 6 meses após a cirurgia está associada a resultados piores. No entanto, a evidência da resposta do CA19-9 à quimioterapia como marcador de prognóstico no câncer de pâncreas permanece discutível. Há uma necessidade urgente de ferramentas clínicas que possibilitem a detecção precoce do câncer de pâncreas, o prognóstico após a terapia e o monitoramento da terapia em pacientes com câncer de pâncreas. Apesar do amplo uso clínico do CA19-9 nos últimos 30 anos, seu valor em influenciar as direções terapêuticas permanece limitado e precisa ser integrado às avaliações clínicas e radiológicas (Goh et al., 2017).*

### REG1B

*REG1B (Regenerating islet-derived 1 beta) é uma proteína codificada pelo gene REG1B, localizado no cromossomo 2p12. Esse gene faz parte da família de proteínas regeneradoras que desempenham papéis importantes na regeneração celular e na proliferação tecidual, particularmente no pâncreas. As proteínas REG estão envolvidas em processos inflamatórios e na regeneração de tecidos após danos, o que explica a elevação de REG1B em condições patológicas, como o câncer de pâncreas. Essa proteína tem sido utilizada como biomarcador para detecção de adenocarcinoma ductal pancreático (PDAC), sendo amplamente pesquisada por sua capacidade de auxiliar no diagnóstico precoce e no monitoramento da progressão da doença (Xu et al., 2019; Radon et al., 2015; Comitê de Nomenclatura de Genes HUGO, 2022). Estudos demonstram que o REG1B pode ser detectado em amostras de urina e soro, fornecendo uma alternativa não invasiva para o rastreamento de pacientes com risco de PDAC (Radon et al., 2015; Xu et al., 2019).*

### REG1A

*REG1A (Regenerating islet-derived 1 alpha) é uma proteína codificada pelo gene REG1A, localizado no cromossomo 2p12, na mesma região do gene REG1B, ambos pertencentes à família de proteínas regeneradoras. O gene REG1A está envolvido em processos de regeneração celular e resposta a danos teciduais, particularmente no pâncreas. Estudos indicam que o REG1A está superexpresso em lesões precursoras do câncer pancreático e promove a proliferação celular, contribuindo para a progressão tumoral, particularmente em pacientes com diabetes (Zhou et al., 2010). A expressão de REG1A foi observada também em outros tipos de câncer, como o câncer de pulmão de células não pequenas (NSCLC), onde sua superexpressão foi associada a um pior prognóstico (Minamiya et al., 2008). Embora o REG1A tenha menor sensibilidade diagnóstica em comparação com o REG1B no contexto do adenocarcinoma ductal pancreático (PDAC), ele desempenha um papel importante quando combinado com outros biomarcadores, como o CA 19-9, melhorando a precisão diagnóstica (Zhou et al., 2010). O REG1A também está envolvido em processos de reparo tecidual e angiogênese, tornando-se relevante para estudos sobre progressão tumoral e regeneração celular (Minamiya et al., 2008; Zhou et al., 2010).*

### LYVE1

*LYVE1 (Lymphatic Vessel Endothelial Hyaluronan Receptor 1) é uma proteína codificada pelo gene LYVE1, localizado no cromossomo 11p15.1, sendo parte da família de receptores endoteliais de ácido hialurônico. Este gene desempenha um papel crucial na homeostase dos*

*vasos linfáticos, facilitando o transporte de ácido hialurônico e estando intimamente envolvido na linfangiogênese. A expressão do LYVE1 tem sido estudada no contexto de diversos tipos de câncer, incluindo o adenocarcinoma ductal pancreático (PDAC). Estudos identificaram que níveis elevados de LYVE1 na urina estão associados à presença de PDAC, sugerindo que o LYVE1 pode ser utilizado como biomarcador não invasivo para detecção precoce da doença (Minamiya et al., 2008). O painel de biomarcadores urinários que inclui LYVE1 demonstrou melhorar significativamente a precisão diagnóstica para PDAC em estágios iniciais. Além de seu potencial diagnóstico, o papel do LYVE1 na linfangiogênese o torna um alvo de interesse para pesquisas sobre metástase linfática em cânceres pancreáticos.*

### Creatinina

*A creatinina é um subproduto do metabolismo da creatina, que é utilizada no fornecimento de energia para a contração muscular. Ela é filtrada pelos rins e excretada pela urina, sendo comumente usada como um marcador confiável da taxa de filtração glomerular (TFG), o que reflete a função renal. A medição dos níveis de creatinina em amostras de urina é essencial em estudos de biomarcadores urinários, pois permite a correção para a diluição urinária, garantindo que as variações observadas nos níveis de biomarcadores sejam devidas às condições patológicas, como o câncer pancreático, e não a alterações na função renal (Radon et al., 2015).*

*Essa normalização é fundamental porque a concentração de biomarcadores, como REG1B e LYVE1, pode variar em função do volume urinário, que por sua vez é influenciado pela hidratação e função renal. Ao ajustar os níveis de biomarcadores em relação à concentração de creatinina, é possível minimizar esses fatores de confusão e garantir que as diferenças observadas nos biomarcadores reflitam com maior precisão a presença e a progressão da doença, como no adenocarcinoma ductal pancreático (PDAC) (Radon et al., 2015).*

*Além disso, a creatinina é amplamente reconhecida por sua estabilidade e consistência em termos de excreção diária, tornando-a um excelente parâmetro para padronização em estudos que envolvem biomarcadores urinários. Isso assegura que as flutuações nos biomarcadores sejam analisadas de maneira mais confiável, levando a uma melhor acurácia diagnóstica em doenças como o PDAC.*

# 3

# Relato do Problema

*O objetivo deste capítulo é apresentar o relato do problema referente ao foco deste trabalho, fazendo uma pequena introdução sobre conceitos importantes da área que auxiliam na compreensão da pesquisa desenvolvida.*

## 3.1 Definição

*Conforme as fundamentações apresentadas nas seções anteriores, quanto mais recente for o diagnóstico, maiores são as chances de cura de qualquer tipo de câncer, incluindo o CP. Quanto mais rapidamente descoberto, maiores são as chances de ressecção cirúrgica, a única forma de cura dessa neoplasia maligna, ou tratamento de paliação que pode elevar a expectativa de sobrevida. Entretanto, devido à raridade do CP na população em geral e em comparação com outros tipos de câncer, o maior volume de pesquisas historicamente foi concentrado nos tipos mais comuns de neoplasias. Isso contribuiu para a falta de clareza sobre o diagnóstico e prognóstico prévio para a população.*

*Mesmo com uma alta taxa de mortalidade, existe uma grande indefinição das formas de diagnóstico prévio e, consequentemente, um prognóstico tardio na população em geral. As formas tradicionais de detecção e acompanhamento incluem: (i) ultrassonografia, (ii) tomografia computadorizada, (iii) ressonância magnética, (iv) ultrassonografia endoscópica, (v) colangiopancreatografia retrógrada endoscópica, (vi) laparoscopia e (vii) biópsia para confirmar o tipo de neoplasia (benigna ou maligna). No entanto, não é recomendável que esses exames sejam aplicados na população em geral ou em grupos de potencial risco, pois seria necessário submeter-se com frequência regular a esses exames ou uma intercalação deles, o que poderia*

*ser ainda mais nocivo à população, tendo em vista que alguns desses exames, se aplicados com frequência, podem levar a várias complicações de saúde, inclusive para algum tipo de câncer.*

*Outra dificuldade são os altos custos desses exames, que não são acessíveis à boa parte da população, principalmente em países considerados pobres. Mesmo em populações com potencial de risco de desenvolverem PDAC, existem muitas perguntas a serem respondidas. Por exemplo, alguém que tem 3 parentes próximos que desenvolveram algum tipo de CP é considerado uma pessoa de alto risco, no entanto, não há consenso sobre a partir de que idade essa pessoa será submetida aos exames de diagnóstico, qual a frequência e até que idade investigar. Essas e outras questões ainda carecem de mais pesquisas.*

*Se o diagnóstico, que possui um maior volume de pesquisas, ainda chega de forma tardia para a maioria dos pacientes com PDAC, é ainda pior com o prognóstico e a definição do melhor tratamento possível. Isso ocorre porque o estadiamento, que é usado como base para inferir o melhor prognóstico ao paciente, só ocorre mediante a realização dos exames de imagens e biópsias, o que leva algum tempo para ser realizado. Além disso, mesmo com os primeiros sinais da doença, é necessário que o paciente tenha acesso a hospitais modernos, disponibilidade financeira e/ou serviços públicos ágeis, o que não é a realidade de muitos pacientes.*

*Outrossim, dado o aumento da expectativa de vida e o envelhecimento da população mundial, são esperados 28,4 milhões de novos casos para todos os tipos de cânceres em 2040, um aumento de 47% em relação a 2020 (Sung et al., 2021). Se nenhuma revolução for feita no diagnóstico e no tratamento nos próximos anos, o estadiamento e o tempo de sobrevida dos pacientes afetados pelo CP se tornarão ainda mais relevantes, dada a quantidade crescente de pacientes que virão a óbito dentro dos primeiros cinco anos. Isso porque é sobre eles que é desenvolvido o tratamento de paliação.*

*Entretanto, dada toda a dificuldade no diagnóstico precoce do PDAC, várias pesquisas paralelas vêm sendo realizadas ao longo dos anos, como muitos estudos de corte visando descobrir potenciais biomarcadores para o diagnóstico prévio do CP. A descoberta desses biomarcadores pode representar a resolução mencionada no parágrafo anterior. Como mencionado anteriormente, existe um biomarcador conhecido para cânceres da região do cólon, que inclui o PDAC.*

## 3.2 Trabalhos Relacionados

### 3.2.1 Seleção dos artigos

*Este pesquisa contou com uma revisão exploratório. Dentre os trabalhos encontrados destacaremos neste capitulo os trabalhos mais relevantes segundo critérios de seleção, foram definidas quatro questões de avaliação, que permitem verificar a proximidade com a proposta.*

| **Questões de Inclusão** | **Descrição e Motivação** |
|---|---|
| QA1: Qual escopo da utilização? | Esta pergunta refere-se ao contexto das neoplasias. |
| QA2: Uso de biomarcador para prognóstico? | Esta pergunta refere-se se o estudo buscou utilizar os biomarcadores para descobrir o estadiamento. |
| QA3: Uso de biomarcadores urinários? | Esta pergunta refere-se ao uso de biomarcadores urinários ainda que em contextos de não prognóstico e/ou classificação. |
| QA4: Utiliza de estadiamento TNM? | Esta pergunta refere-se as técnicas utilizadas para criar o modelo de classificação. |

Tabela 3.1: Fonte: Autor

### 3.2.2 Lista de Trabalhos

- *(Lesko et al., 2022): A pesquisa utiliza o biomarcador MicroRNA 371a-3p, pertencente ao grupo de pequenos RNAs não codificantes, visando identificar precocemente o câncer de testículo no estágio I, sem estabelecer uma relação específica com os demais estágios TNM.*
  - *Este estudo se relaciona com o trabalho atual, demonstrando a ampla aplicabilidade dos diversos biomarcadores existentes e o uso desses biomarcadores para estimar um estágio TNM. Nesse caso, o biomarcador é usado para diagnosticar uma neoplasia e estimar o estágio I do TNM. No entanto, a neoplasia investigada é diferente: (Lesko et al., 2022) tem como objetivo identificar apenas o estágio I do TNM, enquanto a dissertação atual busca identificar os quatro principais estágios do TNM do PDAC, ver Tabela 3.2.*

Tabela 3.2: Comparação entre o estudo de Lesko et al. (2022) e a dissertação atual

| **Aspecto** | **Lesko et al. (2022)** | **Dissertação atual** |
|---|---|---|
| Biomarcador | MicroRNA 371a-3p | LYVE1, REG1A, TFF1 |
| Neoplasia | Câncer de testículo | Adenocarcinoma ductal pancreático (PDAC) |
| Estágio TNM investigado | Estágio I | Estágios I, II, III e IV |
| Objetivo do biomarcador | Identificar precocemente o câncer de testículo no estágio I | Identificar os quatro principais estágios do TNM do PDAC |
| Relação com outros estágios TNM | Não estabelece relação específica com os demais estágios TNM | Estuda a relação com os quatro principais estágios do TNM |

2020 *(Debernardi et al., 2020): Utiliza biomarcadores urinários para diagnóstico prévio do PDAC, com um total de 590 amostras. a) Essa pesquisa focou na melhoria da acurácia do diagnóstico prévio do PDAC por meio da substituição do REG1A por REG1B, apresentando uma acurácia de 0.99.*

- *O conjunto de dados deste trabalho foi utilizado nesta pesquisa. No entanto, o foco das pesquisas é distinto. Enquanto (Debernardi et al., 2020) atuou no diagnóstico usando biomarcadores urinários, a presente pesquisa atua no prognóstico para estimar o estadiamento TNM. Ou seja, esta dissertação é um* ***complemento a esse estudo****, dado que ambas pesquisas utilizam-se dos mesmos biomarcadores, mas atuam em momentos distintos. Comprovando-se a eficácia de ambas as pesquisas, é possível, em um único exame, obter/estimar tanto o diagnóstico quanto o prognóstico do PDAC.*

Tabela 3.3: Comparação entre o estudo de Debernardi et al. (2020) e a dissertação atual

| **Aspecto** | **Debernardi et al. (2020)** | **Dissertação atual** |
|---|---|---|
| Biomarcadores | LYVE1, REG1B, TFF1 | LYVE1, REG1B, TFF1 |
| Neoplasia | Adenocarcinoma ductal pancreático (PDAC) | Adenocarcinoma ductal pancreático (PDAC) |
| Foco da pesquisa | Diagnóstico prévio do PDAC | Prognóstico e estadiamento TNM do PDAC |
| Acurácia | 0.99 (com a substituição do REG1A por REG1B) | 0.81 (com uso de Random Forest) |
| Complementaridade | - | Complementa o estudo de Debernardi et al. (2020) |

2015 *(Honda et al., 2015): Utiliza o biomarcador plasmático apoAII-ATQ/AT, no diagnóstico prévio e na busca por identificar o estágio do DPAC. Com à acurácia de 0,94 para estadiamento I/II e 0,93 para a classe III/IV.*

- *Artigo apresenta um grande desbalanceamento das amostras (I = 2, II = 10, III = 12 e IV = 102).*
- *Agrupou as classes I/II e III/IV, divergindo da atual pesquisa que busca classificar os 4 graus da TNM.*
- *Usa o biomarcador ATQ/AT, não contemplado nesta pesquisa.*

## 3.3 Impacto

*A maioria dos estudos atuais relacionados do PDAC e aos demais tipos de cânceres de pâncreas tem se concentrado na descoberta de biomarcadores para um diagnostico precoce, pois sabe-se, que mesmo sem o desenvolvimento de novas técnicas para o tratamento do PDAC, se for possível diagnosticar precocemente o PDAC, as taxas de sobrevida e sucesso na resseção e por sua vez a cura definitiva do PDAC subirá drasticamente. No entanto, poucos estudos tem buscado uma avanço em um precoce prognóstico. É um prática conhecida que para qualquer tipo de tratamento das neoplasias se faz necessária uma analise do estadiamento do paciente, por vezes, o diagnostico só ocorre quando resta poucos meses de sobrevida. Quanto mais precoce for realizado o estadiamento, mais tempo e opções de tratamento e/ou paliação estarão disponíveis para o paciente.*

# 4

# Proposta

*O objetivo desta capítulo é apresentar a proposta do trabalho, bem como os detalhes a serem considerados na elaboração do modelo de aprendizagem de máquina, e nas correlações das variáveis utilizadas.*

## 4.1 Fundamentação

*O objetivo deste trabalho é a elaboração de uma Proposta e Avaliação de um Modelo prévio de estadiamento TNM em pacientes diagnosticados com câncer de pâncreas PDAC com base em biomarcadores. A proposta dessa pesquisa se difere na busca do prévio estadiamento com base em biomarcadores inicialmente propostos para diagnóstico do PDAC.*

*A fundamentação tem como base a utilização de técnicas de aprendizagem de máquina, visando analisar a viabilidade do uso de biomarcadores cancerígenos com objetivo de prever o estadiamento tumoral no TNM ou parte deste estadiamento, que tradicionalmente é feito com base em exames de imagens. Entretanto, esta dissertação avalia a viabilidade de uso de biomarcadores para prever o estadiamento sem o uso de imagens, dando apoio à decisão médica de ações de prognóstico. Este estudo se alicerça no avanço de inúmeras pesquisas para prévio diagnóstico do CP com base em biomarcadores em específico em pacientes com PDAC.*

*O TNM é utilizado na elaboração de um estadiamento para pacientes com tumores malignos, baseando-se no tamanho e localização do tumor. Os dados utilizados nesta pesquisa*

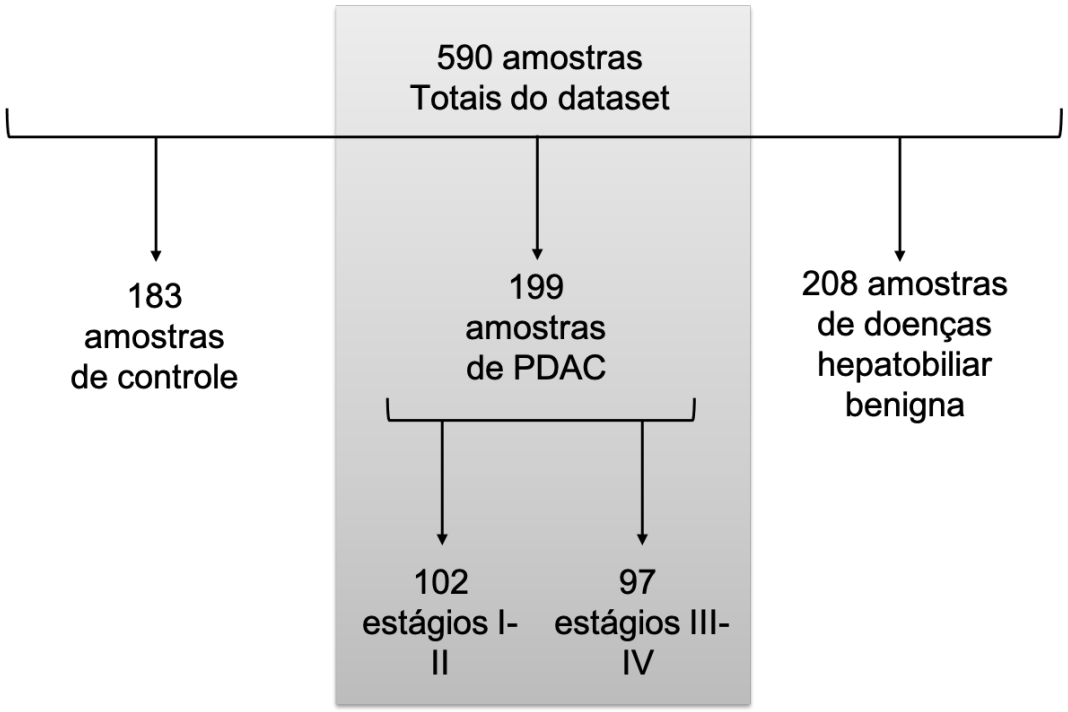



Figura 4.1: Fluxo de separação da base de dados

*foram obtidos de amostras clínicas de vários centros: Barts Pancreas Tissue Bank, University College London, University of Liverpool, Spanish National Cancer Research Center, Cambridge University Hospital e University of Belgrade. O uso de dados reais é altamente relevante em qualquer pesquisa. Entretanto, a disponibilidade de datasets de pacientes com CP não é tão abrangente como em outros tipos de cânceres mais comuns, mesmo, em centros especializados, uma vez que a amostra é reduzida ou o acesso é permeado por vários critérios de acesso. O dataset utilizado nesta pesquisa foi publicado no artigo de (Debernardi et al., 2020), contendo um total de 590 amostras, com dados sobre sexo, idade, e 5 biomarcadores.*

### 4.1.1 Fluxo de normalização dos dados

*O dataset obtido inicialmente proposto para diagnóstico prévio de CP do tipo PDAC, com um total de 590 amostras, sendo separadas em 3 grupos: (i) Amostra de controle com 183 pacientes, (ii) amostra de PDAC com 199 pacientes, (iii) amostra de doenças hepatobiliar benigna com 208, conforme figura 4.1; destas amostras somente os dados relacionados ao PDAC foram utilizados neste estudo, dos quais estão contam com 116 amostras do sexo masculino e 83 do sexo masculino, com uma faixa etária mediana global de 67 anos.*

## 4.2 Hipóteses

*Esta seção reúne e descreve as hipóteses que foram elaboradas no decorrer da revisão da literatura, desde os modelos que serviram para diagnóstico, até os modelos para prognóstico.*

*As hipóteses que foram elaboradas a partir da revisão exploratória, a resolução final e as que foram testadas, foram as hipóteses presentes na tabela 4.1*

| | Hipóteses |
|---|---|
| 0 | É possível prever o estadiamento com base em biomarcadores. |
| 1 | Não É possível prever o estadiamento com base em biomarcadores. |
| 2 | Aplicação de algoritmos de aprendizagem de máquina com base em biomarcadores, obtém resultados iguais, superiores ou correlacionados ao estadiamento TNM por imagem. |
| 3 | Aplicação de algoritmos de aprendizagem de máquina com base em biomarcadores, não obtém resultados iguais, superiores ou correlacionados ao estadiamento TNM por imagem. |

Tabela 4.1: Hipóteses.

# 5

# Experimentação

*Este capítulo descreve a metodologia que será aplicada no sistema, como será desenvolvido, técnicas, instrumentos, ou, dispositivos que serão utilizados para obtenção dos resultados, ao desenvolver o experimento proposto nesta dissertação.*

## 5.1 Ferramentas

### 5.1.1 O que é Machine Learning?

*É uma subárea da IA(Inteligência Artificial), que refere-se ao estudo de algoritmos que aprendem seu comportamento a partir de dados. Para ver por que esses algoritmos são importantes, considere a seguinte tarefa básica, construir um programa para prever se uma imagem contém um cachorro ou um gato. Embora seja uma tarefa árdua especificarmos manualmente as regras exatas para determinar que um cachorro é um cachorro, é comparativamente simples preparar um conjunto de referência de imagens e rótulos (ou seja, cachorros ou gatos). Essa configuração, em que o conhecimento é mais facilmente codificado em dados do que em um conjunto descritivo de regras, é o foco dos algoritmos de ML. Dado um conjunto de referência, ou seja,dados de treinamento, e uma métrica de desempenho para otimizar, ou seja, um objetivo do modelo, os algoritmos de ML geralmente começam com uma estimativa aleatória. Por exemplo, considere o seguinte modelo simplificado em Fig 5.1.*

### 5.1.2 Algoritmo KNN

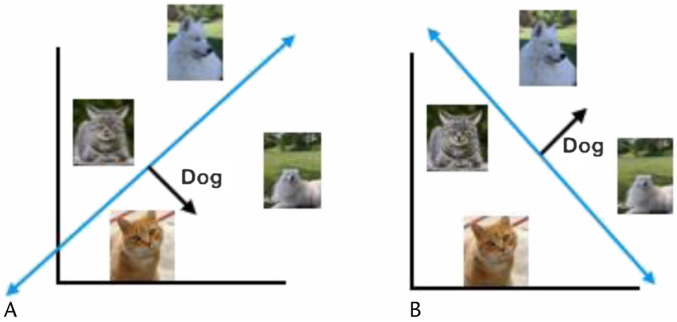



Figura 5.1: Aprendendo a classificar imagens de cães versus gatos. A, Suposição inicial do modelo. B, A estimativa refinada após medir seu desempenho no conjunto de referência (Kenner et al., 2021).

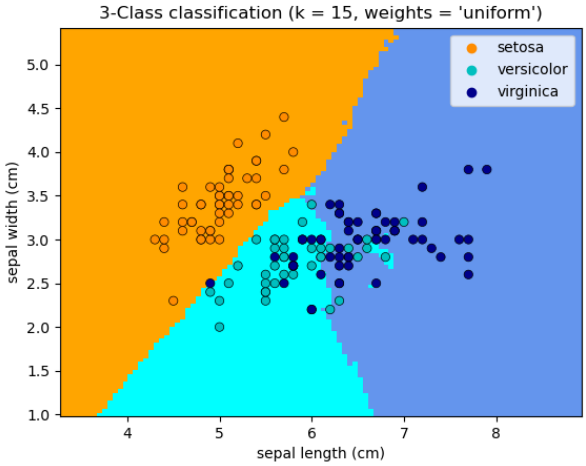



Figura 5.2: Exemplo de classificação KNN (The scikit-learn developers, 2022).

$$d(x,y) = \sqrt{\sum_{n=1}^{n} (x_i - y_i)^2} \tag{5.1}$$

*Existe vários algoritmos dentro da ML para classificar objetos, coisas, etc. A escolha de um algoritmo adequado para solução do problema é de fundamental importância. Dentre estes, o algoritmo de k-vizinhos mais próximos (k-NN, do inglês: k-nearest neighbors) é um método de aprendizado supervisionado não paramétrico desenvolvido pela primeira vez por Evelyn Fix e Joseph Hodges em 1951, e posteriormente expandido por Thomas Cover (Fix and Hodges, 1989). É usado para classificação e regressão . Em ambos os casos, a entrada consiste nos k exemplos de treinamento mais próximos em um conjunto de dados, Fig 5.2.*

### 5.1.3 Algoritmo Floresta Aleatória (Random Forest)

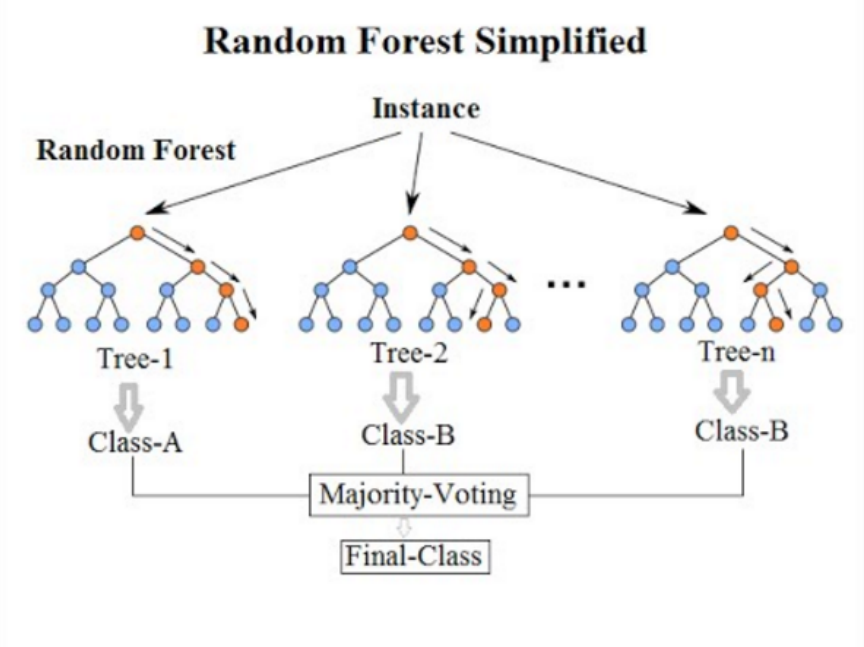



Figura 5.3: Exemplo de classificação SVM (The scikit-learn developers, 2022).

*Florestas aleatórias ou florestas de decisão aleatória é um método de aprendizado conjunto para classificação , regressão e outras tarefas que opera construindo uma infinidade de árvores de decisão em tempo de treinamento. Para tarefas de classificação, a saída da floresta aleatória é a classe selecionada pela maioria das árvores. Para tarefas de regressão, a previsão média ou média das árvores individuais é retornada. (Ho, 2016) Florestas de decisão aleatória corrigem o hábito das árvores de decisão de se ajustarem ao seu conjunto de treinamento.*

*A Floresta Aleatória é um algoritmo que pertence a classe dos algoritmos supervisionados de Aprendizagem de Máquina. Ele é um dos algoritmos mais utilizados pela sua simplicidade e diversidade, e também pelos bons resultados apresentados mesmo sem a utilização de hiper-parâmetros. Além disso, ele pode ser utilizado tanto para tarefas de classificação ou de regressão (Breiman, 2001).*

### 5.1.4 Máquinas de Vetor de Suporte (Support-Vector Machine - SVM)

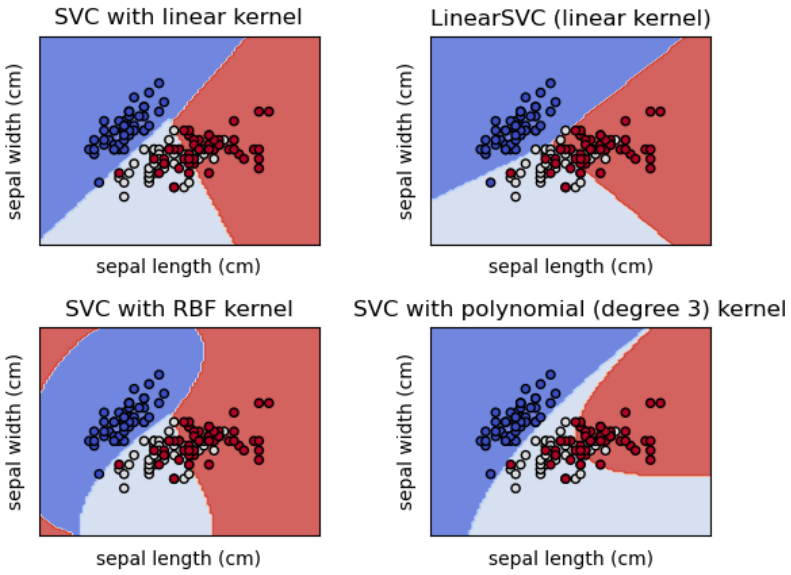



Figura 5.4: Exemplo de classificação SVM (The scikit-learn developers, 2022).

*São um conjunto de métodos de aprendizado supervisionado usados para classificação, regressão e detecção de outliers(Cortes and Vapnik, 1995).*

## 5.2 Ferramentas Utilizadas

*O experimento foi realizado no ambiente virtual online Colaboratory (Colab). Esta ferramenta permite que qualquer pessoa escreva e execute código Python arbitrário por meio do navegador e é especialmente adequado para aprendizado de máquina, análise de dados e educação. Mais tecnicamente, o Colab é um serviço de notebook Jupyter hospedado que não requer configuração para uso, enquanto fornece acesso gratuito a recursos de computação, incluindo GPUs.*

### 5.2.1 Equipamentos:

*Para a realização desta dissertação, os seguintes equipamentos foram utilizados: (i) um notebook Macos Air, conforme Fig. 5.5*


Figura 5.5: Detalhamento das configurações do computador (notebook) utilizado

### 5.2.2 Linguagem de programação:

*Nesse experimento utilizamos a linguagem de programação python em sua versão 3.10. Esta linguagem foi selecionada por ter um amplo aparato de bibliotecas voltadas à ciência de dados e por sua ampla utilização no meio acadêmico e profissional de pessoas que atuam com ciência de dados.*

#### Bibliotecas

*Bibliotecas utilizadas nos scripts escritos em python:*

- ***Pandas**: é uma biblioteca para uso em Python, open-source e de uso gratuito (sob uma licença BSD), que fornece ferramentas para análise e manipulação de dados.*
- ***Numpy**: significa Numerical Python (Python Númerico), tudo isso devido ao fato de ser baseado nos projetos Numeric e Numarray que foi feito com objetivo de reunir a comunidade em torno de um framework de processamento Arrays.*
- ***Seabourn**: é uma biblioteca usada principalmente para plotagem estatística em Python. Ele é construído em cima do Matplotlib e fornece belos estilos padrão e paletas de cores para tornar os gráficos estatísticos e atraentes.*
- ***Matplotlib.pyplot**: é uma biblioteca da linguagem de programação Python, utilizada para visualização de dados e plotagem gráfica.*
- ***Plotly.express**: é uma biblioteca de visualização de dados para Python, Javascript e R.*
- ***klearn.preprocessing import LabelEncoder**: é uma biblioteca de transforma os valores das variáveis rotuladas iniciando em 0 até o número de classes menos um, 0:n_classes-1.*

- ***sklearn.preprocessing import OneHotEncoder****: Essa codificação é necessária para alimentar dados categóricos para muitos estimadores scikit-learn, principalmente modelos lineares e SVMs com os kernels padrão.*

- ***sklearn.compose import ColumnTransformer****: Este estimador permite que diferentes colunas ou subconjuntos de colunas da entrada sejam transformados separadamente e as características geradas por cada transformador serão concatenadas para formar um único espaço de características. Isso é útil para dados heterogêneos ou colunares, para combinar vários mecanismos de extração de recursos ou transformações em um único transformador.*

## 5.3 Metodologia

### 5.3.1 Fase I Aquisição de Dados

**Dataset:**

*O Dataset utilizado foi baseado em dados reais visando validar os modelos propostos, o que não seria possível de averiguar com dados simulados. Sendo assim, foi selecionado o dataset elaborado por (Debernardi et al., 2020), sendo que não foram encontrados outros dados públicos de interesse desta pesquisa. As amostras clínicas usadas no dataset foram obtidas de vários centros: Barts Pancreas Tissue Bank, University College London, University of Liverpool, Spanish National Cancer Research Center, Cambridge University Hospital e University of Belgrade. O painel de biomarcadores foi testado em 590 amostras de urina: 183 amostras de controle, 208 amostras de doença hepatobiliar benigna e 199 amostras de PDAC (102 estágio I–II e 97 estágio III–IV); 50,7% eram de indivíduos do sexo feminino. Amostras de PDAC foram coletadas de pacientes antes do tratamento, Fig 4.1.*

*A disponibilização dos dados é feita através de arquivo comprimido com as colunas que contêm os tipos de dados clínicos que referenciam os pacientes em Comma-separated values (CSV), sendo possível selecionar os dados que serão relevantes para a pesquisa.*

### 5.3.2 Fase II: Organização dos Dados

*Destes dados foram selecionados apenas os campos de interesse da pesquisa, ou seja, os biomarcadores, os estadiamento e por fim idade e sexo do paciente, ficando com as seguintes colunas utilizadas na pesquisa:*

- *Age*
- *Sex*
- *stage*
- *Plasma CA 19-9*
- *Creatinine*
- *LYVE1*
- *REG1B*
- *TFF1*
- *REG1A*

### 5.3.3 Fase III: Divisão dos Cenários

*Os biomarcadores, idade e sexo foram divididos em dois grupos para avaliação do impacto da Age(Idade) e Sex(Sexo) em contraste com Biomarcadores na predição dos graus de estadiamento. Outra questão levada em consideração na criação dos cenários foi o balanceamento dos dados, ou seja, foram criados cenários com os dados sem balanceamento e com balanceamento.*

### 5.3.4 Fase IV: Aplicação do algoritmos de ML

*Fase III: Aplicação do KNN em busca de prever o estadiamento. Os dados depois de coletados e selecionados, foram analisados enfatizando a previsão do tempo de sobrevida e classificação do câncer de pâncreas em vários estágios:*

- ***Cenário 1: Sem Balanceamento + (CA 19-9, CREATITINE, LYVE1, TFF1, REG1A, REG1B)***

  - *Aplicação do algoritmo KNN*
  - *Aplicação do algoritmo Random Forest*
  - *Aplicação do algoritmo SVM*

- ***Cenário 2: Com Balanceamento + (CA 19-9, CREATITINE, LYVE1, TFF1, REG1A, REG1B)***
  - *Aplicação do algoritmo KNN*
  - *Aplicação do algoritmo Random Forest*
  - *Aplicação do algoritmo SVM*

- ***Cenário 3: Sem Balanceamento + (CA 19-9, CREATITINE, LYVE1, TFF1, REG1A, REG1B)+Age+Sex***
  - *Aplicação do algoritmo KNN*
  - *Aplicação do algoritmo Random Forest*
  - *Aplicação do algoritmo SVM*

- ***Cenário 4: Com Balanceamento + (CA 19-9, CREATITINE, LYVE1, TFF1, REG1A, REG1B)+Age+Sex***
  - *Aplicação do algoritmo KNN*
  - *Aplicação do algoritmo Random Forest*
  - *Aplicação do algoritmo SVM*

## 5.4 Parâmetros

*Colunas de entrada do dataset: O dataset conta com os seguinte campos:*

- *Sample_id = Id da amostra*
- *Patient_cohort = Coorte de pacientes*
- *Sample_origin = Origem da amostra (centro médico)*
- *Age = Idade*
- *Sex = Sexo(M = Masculino, F = Feminino)*
- *diagnosis = Diagnostico (1= Grupos de amostra,2=Doenças benignas e 3= PDAC)*

- *stage = Estadiamento NTM*
- *benign_sample_diagnosis = diagnóstico de amostra benigna*
- *Plasma CA 19-9 = Biomarcado tumoral CA 19-9*
- *Creatinine = Creatinina*
- *LYVE1 = biomarcadores de proteínas na Urina*
- *REG1B = biomarcadores de proteínas na Urina*
- *TFF1 = biomarcadores de proteínas na Urina*
- *REG1A = biomarcadores de proteínas na Urina*

***OBS.*** *Nos scripts foram tratadas as variáveis nominais, transformando-as em valores numéricos, sendo esta ação necessária para o correto seguimento do algoritmo.*

# 6

# Resultados

*Esta seção descreve os resultados obtidos no experimento proposto, comparando o resultado das técnicas de ML aplicadas (KnnClassifier, Random forest e SVM) para avaliar a possibilidade de previamente estimar o grau do estadiamento por biomarcadores promissores no diagnostico de PDAC, antes das realização de exames de imagens.*

*No cenário 1, o algoritmo KNN apresentou a maior acurácia de 0.55. no cenário 2, o algoritmo Random Forest apresentou a maior acurácia alcançando 0.75. No cenário 3, o Random Forest novamente apresentou a maior acurácia de 0.575, embora inferior ao cenário 2. Por fim, no cenário 4 o algoritmo RandomForest alcançou a maior acurácia de todos os cenários, atingindo o score de 0.81. Os scripts dos algoritmos utilizados foram disponibilizados no Apêndice A e B deste trabalho, assim, como compartilhados por meio da plataforma do github:*

- *ScriptA (Conrado, 2022a): Contempla os cenários 1 e 2*
- *ScriptB (Conrado, 2022b): Contempla os cenários 3 e 4*

## 6.1 Métricas

*As métricas utilizadas para avaliação dos resultados são:*

- ***Precision (Precisão)** (também chamada de valor preditivo positivo) é a fração de instân-*

*cias relevantes entre as instâncias recuperadas.*

$$Precision = \frac{VP}{VP+FP} \tag{6.1}$$

- ***Recall (Revocação)*** *(também conhecido como sensibilidade) é a fração de instâncias relevantes que foram recuperadas.*

$$Recall = \frac{VP}{VP+FN} \tag{6.2}$$

- ***Accuracy (Acurácia)*** *é uma média aritmética ponderada entre precisão e precisão inversa (ponderada por viés), bem como uma média aritmética ponderada entre revocação e revocação inversa (ponderada por prevalência)*

$$Accuracy = \frac{VP+VN}{VP+VN+FP+FN} \tag{6.3}$$

- *VP = Verdadeiro Positivo*
- *VN = Verdadeiro Negativo*
- *FP = Falso Positivo*
- *FN = Falso Negativo*

## 6.2 Descrição

*Conforme é possível observar na Fig.6.1 existe um número grande de classes, podendo gerar uma maior imprecisão na classificação. Diante disso, agrupamos essas classes pelos graus I, II, II e IV, conforme é possível observar na Fig. 6.2. É possível observar outro problema: existe desbalanceamento nos dados no dataset utilizado, o que podem prejudicar o treinamento e por consequência a correta classificação das amostras. Por esse motivo, foi necessário criar mais cenários, que avaliassem os algoritmos com os dados sem balanceamento e os algoritmos com balanceamento de dados, conforme Fig. 6.3.*

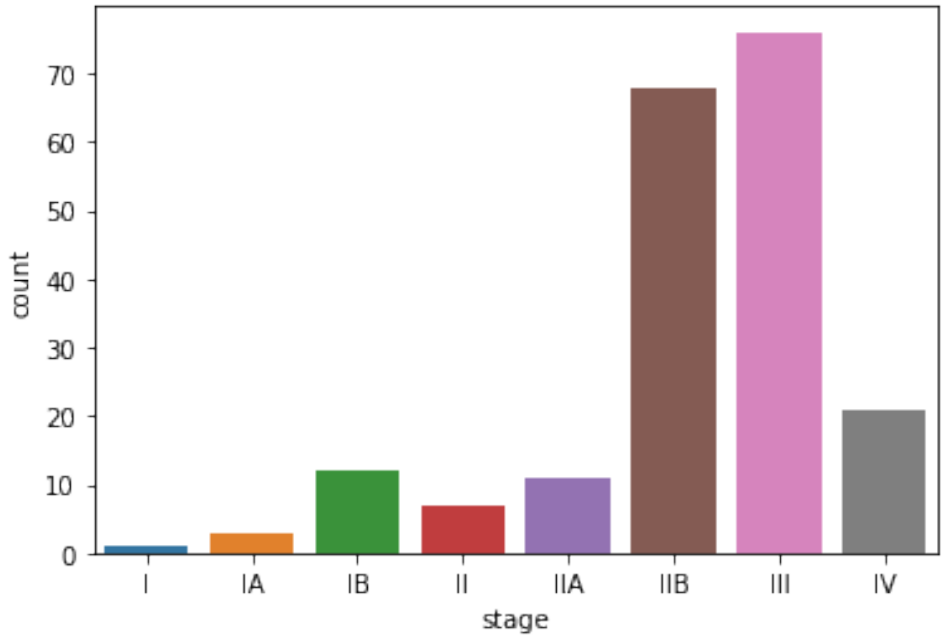



Figura 6.1: Os estágios do TNM no dataset está apresenta um grande número de classificadores

*É possível observar que os estadiamentos não estão uniformemente distribuídos. Ainda que agrupados, o estadiamento I e seus subgrupos (IA e IB) não possuem a mesma representação numérica que as amostras II e III, o mesmo se aplica ao grau IV, Fig. 6.1. Já no gráfico da Fig.*

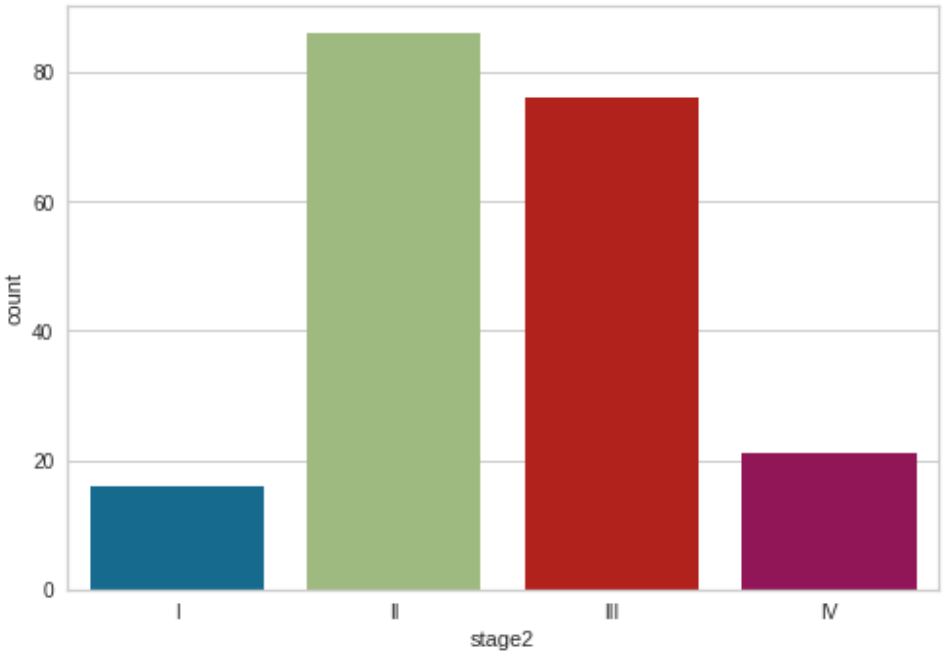



Figura 6.2: Gráfico das classes após agrupamento para classificação via algoritmos de ML

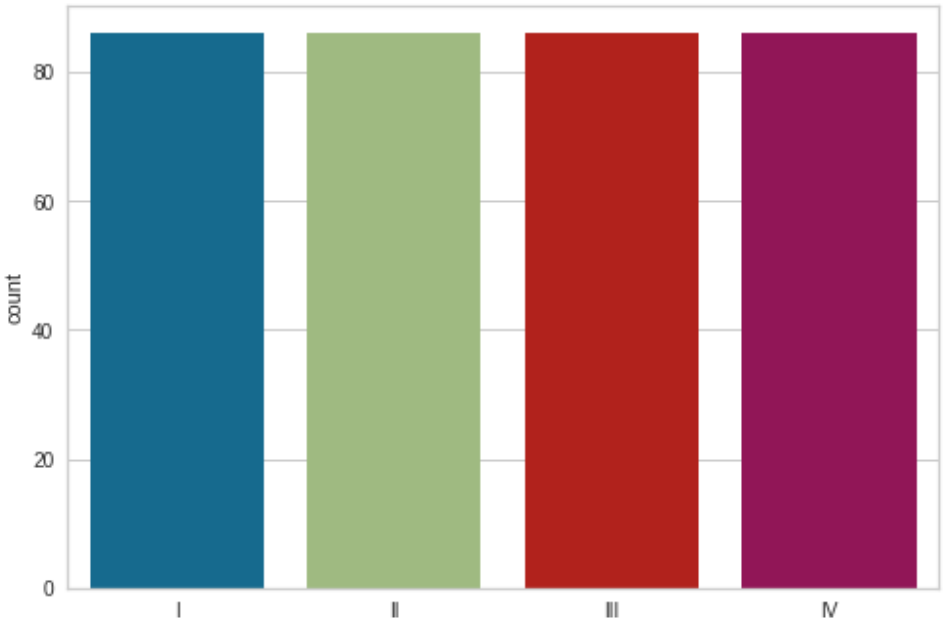



Figura 6.3: O Gráfico das classes após o uso da SMOTE, técnica de sobreamostragem onde as amostras síntetizadas são geradas para a classe de sobreamostragem

*A seguir um historiograma com as medianas e distribuição de frequência por Biomarcador.*

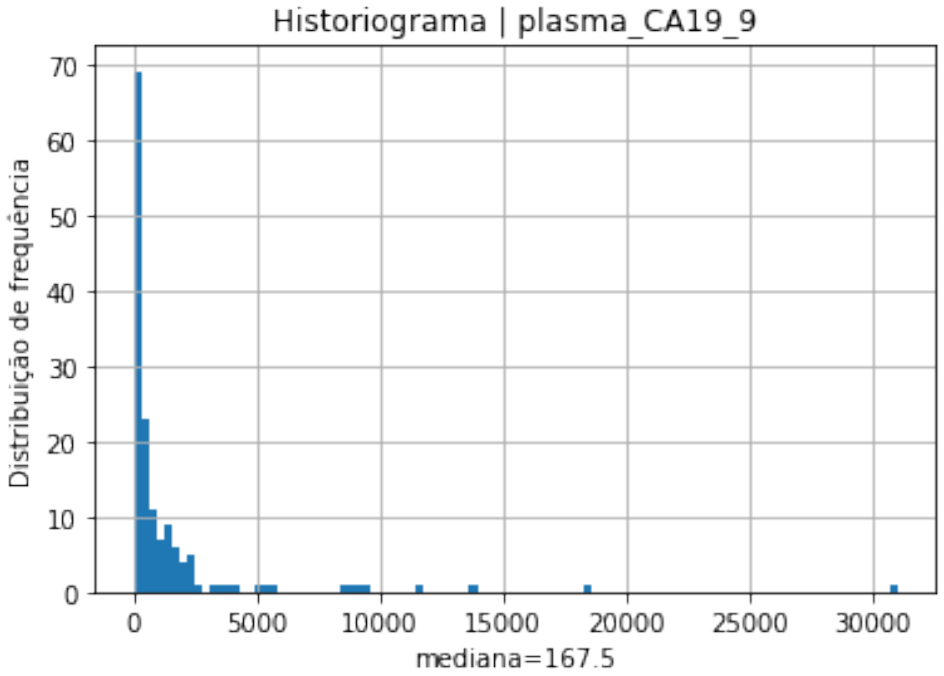



Figura 6.4: Historiograma do CA 19-9

*O antígeno carboidrato CA 19-9, quando acima de 37U/mL (Unidades por mililitro, FIG 6.4.*

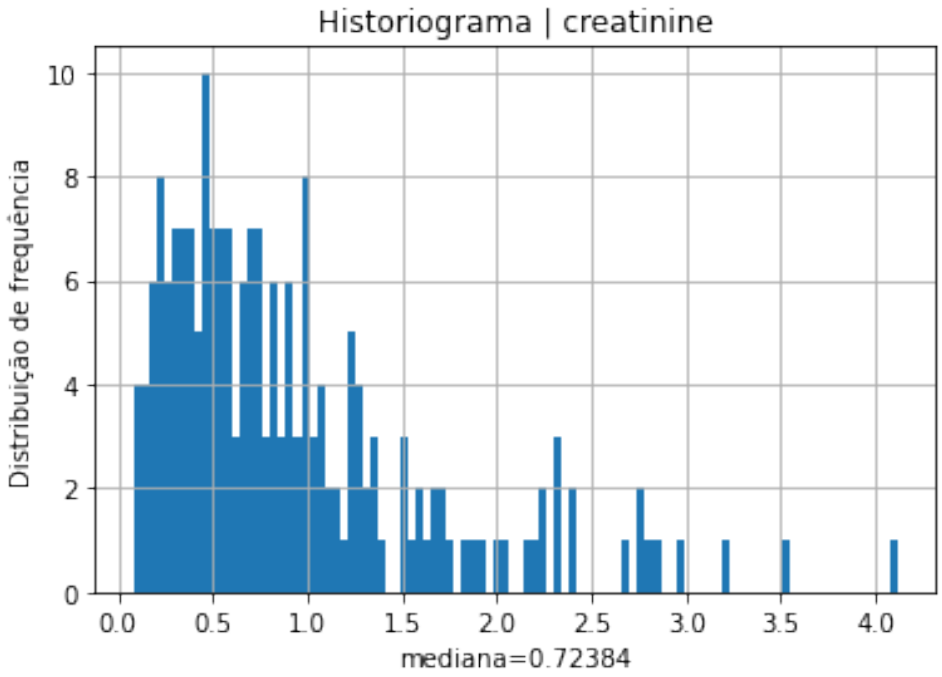



Figura 6.5: Historiograma da creatinina

*O valor da medida da creatinina encontra-se no intervalo de valores normais para ambos os sexos. 6.5.*

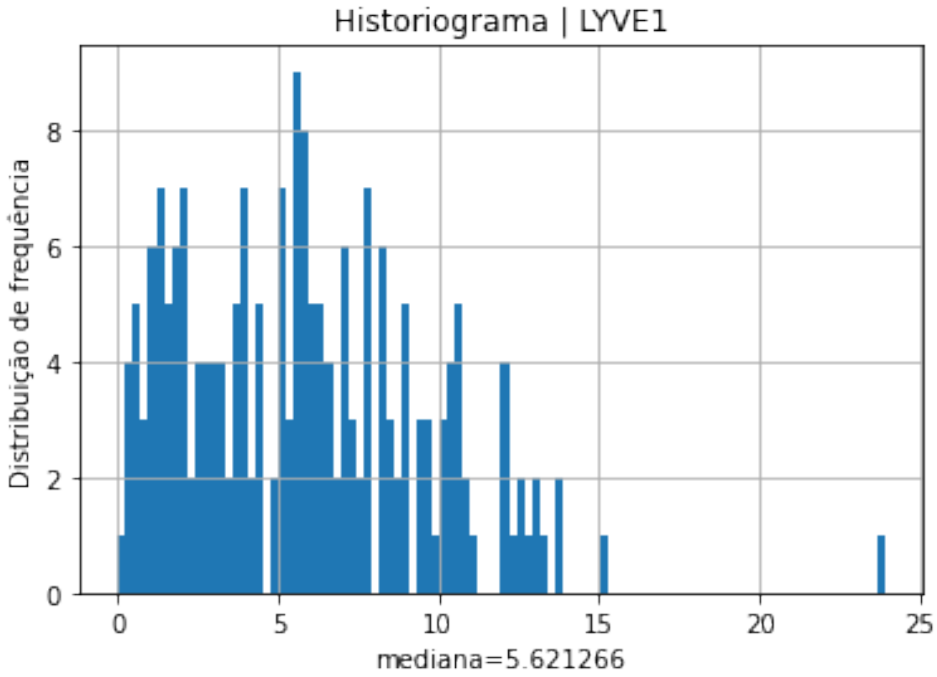



Figura 6.6: Historiograma do LYVE1

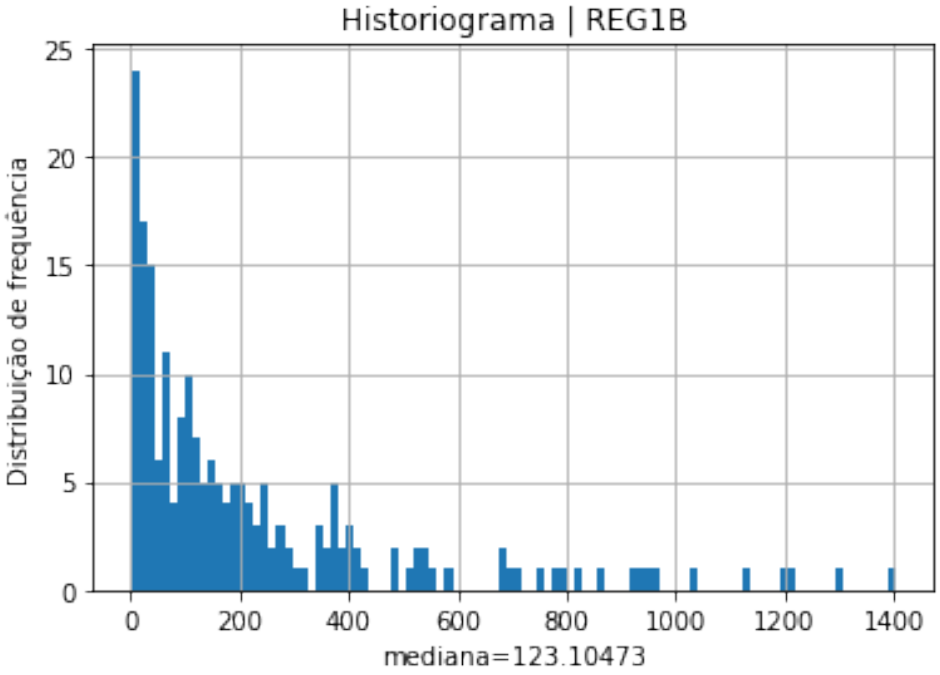



Figura 6.7: Historiograma do REG1B

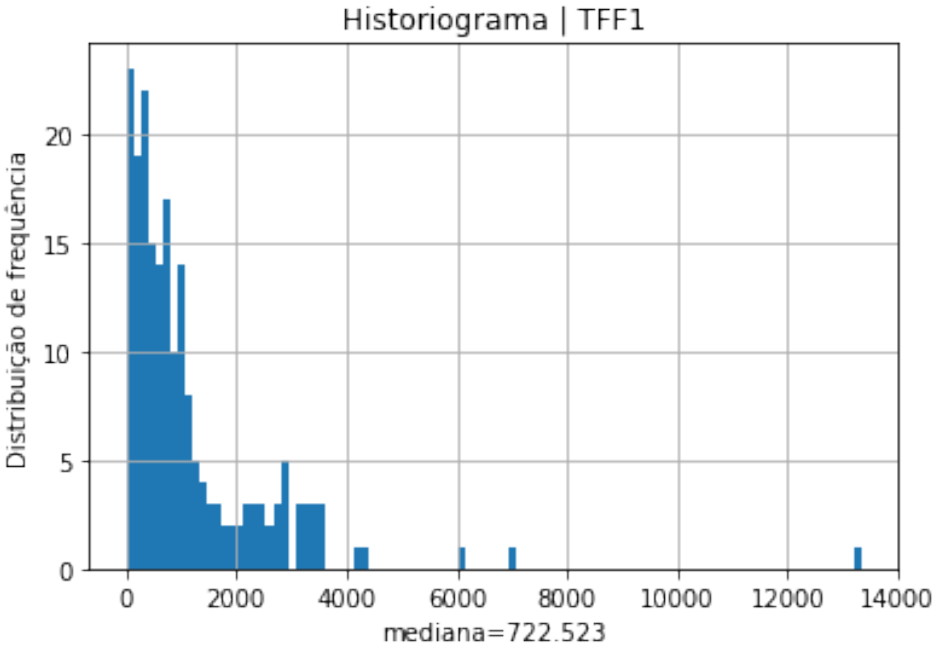



Figura 6.8: Historiorama do TFF1

.

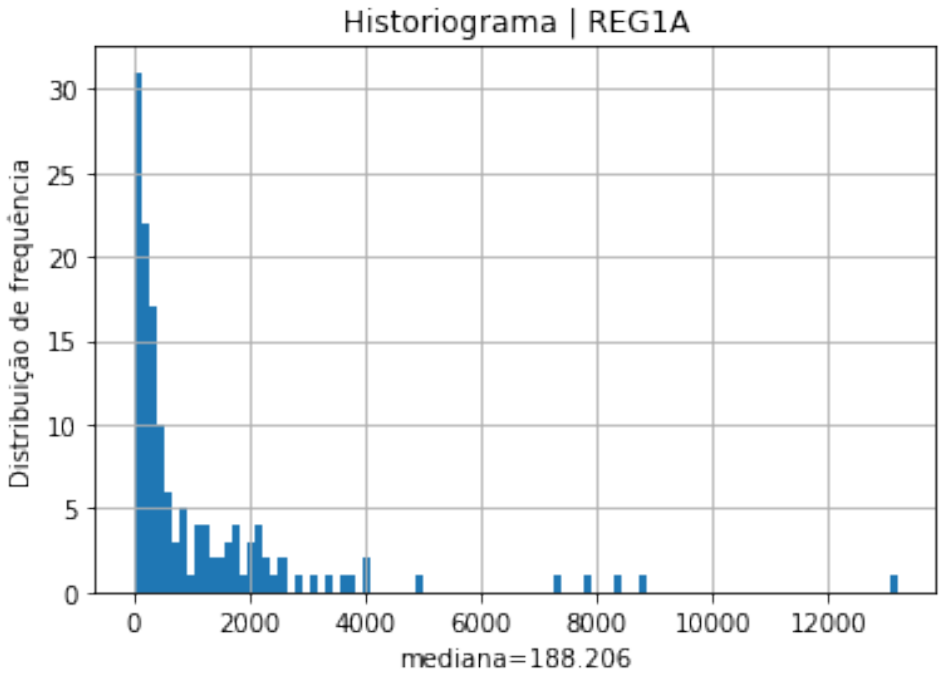



Figura 6.9: Historiorama do REG1A

*A proteína CA 19-9, FIG 6.9.*

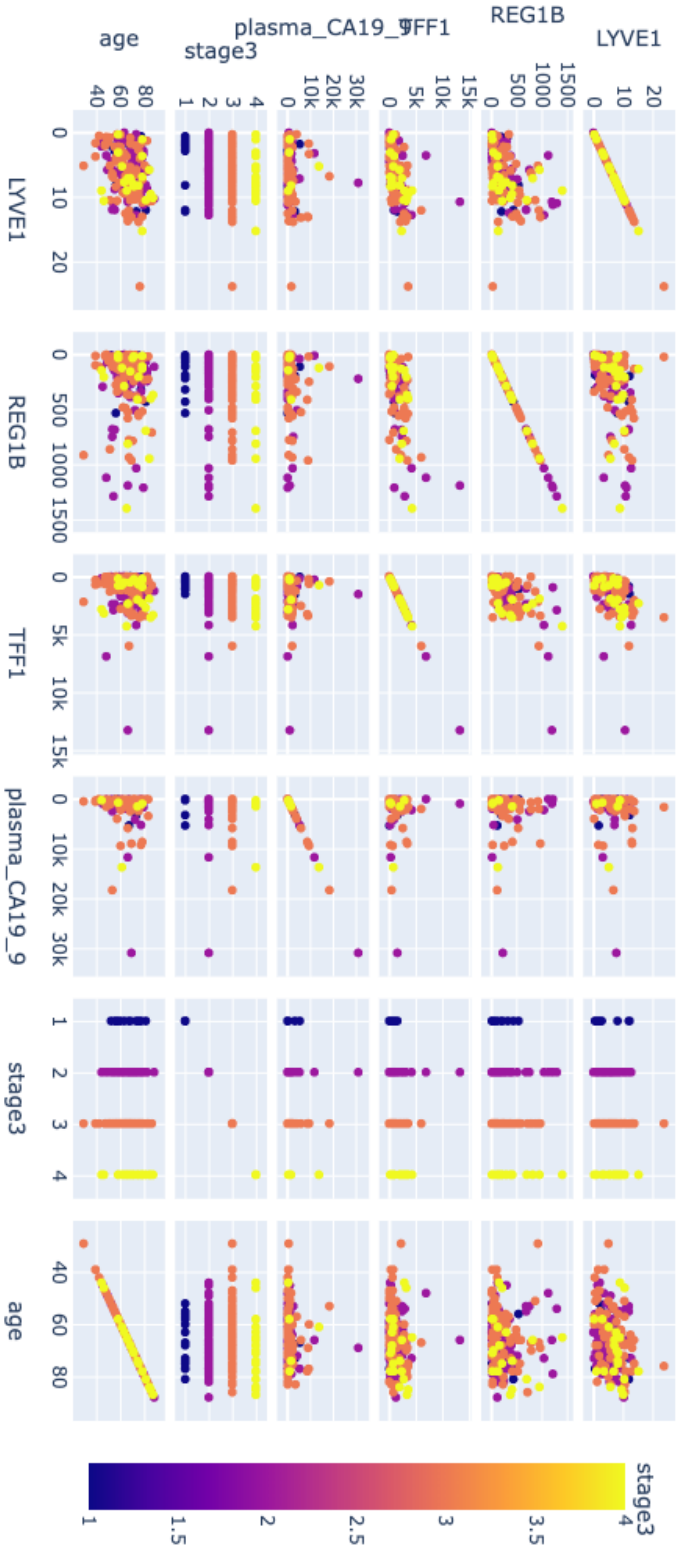



Figura 6.10: Matriz de dispersão (Scatter Matrix) com 6 dimensões para visualizarmos as tendências nos dados

*Conforme é possível observar na Matriz de dispersão ou do inglês, Scatter Matrix na Fig. 6.10, 4 dos biomarcadores são apresentados (excluímos o REG1A, para melhorar a visuali-*

*zação dos dados, pois, o excesso de dimensões dificultaria a analise e por essa proteína ter uma correlação menor que o REG1B já representado no gráfico), o estadiamento (stage3, para agrupar a visualização dos estadiamentos) e a idade (age). Os gráficos na diagonal da matriz representam a intersecção da próxima dimensão, existe uma aparente não correlação dos estadiamentos com os biomarcadores.*

## 6.3 Análise

*Em cada cenário foram aplicadas no mínimo 3 algoritmos de ML: KnnClassifier, Random forest, e SVM.*

### 6.3.1 Cenários

*Como nas amostras coletadas, foram disponibilizados poucas informações além dos biomarcadores. Neste trabalho foi decidido que não seriam descartadas nenhuma variável, visando conservar o maior número de variáveis possíveis sobre os pacientes. Todavia, foi decidido avaliar o impacto da idade e sexo na classificação conjunta com os biomarcadores do dataset em comparação com a classificação somente dos biomarcadores. Além disso, conforme apresentado na seção de descrição deste capitulo, foi criada outra bifurcação para avaliar se a utilização de técnicas de balanceamento poderiam apresentar uma melhoria nos resultados.*

### 6.3.2 Cenário 1: Biomarcadores urinários sem (idade, sexo) e sem balanceamento

#### A) C1-01 KnnClassifier - Unbalanced (acurácia = 0.35)

*No primeiro experimento do cenário 1 conforme as Figs. 6.11 e 6.12, foi encontrada uma acurácia de 0.35. É possível observar que existe um problema na distribuição de dados que foi gerada para o y_test, pois, o mesmo não enviou nenhuma amostra da classe IV, e as 5 amostras identificadas pelo algoritmo estão como pertencentes a classe IV (sendo falsos positivos). Mesmo com 50% de precisão, a classe I também contou com um baixo número de amostras, o que gera pouca confiabilidade nessa classificação. As classes 3 e 4 mesmo contando com mais amostras, apresentaram baixa precisão e um alto número de falsos positivos. Em suma, o maior f1-score foi da classe I com 0.57 e o menor da classe IV com 0.00.*

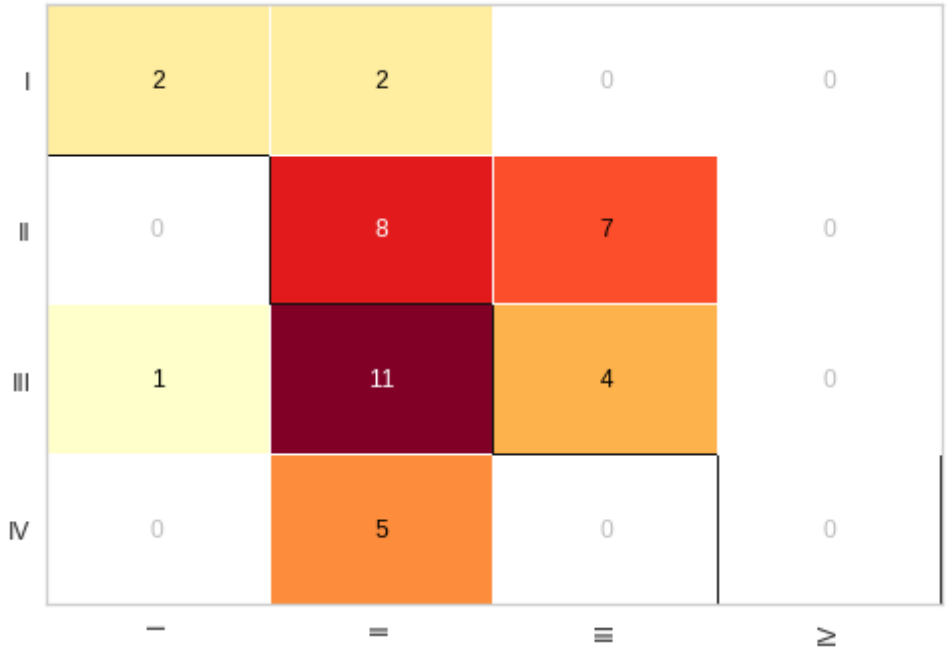



Figura 6.11: Matriz de confusão do Cenário C1-01, com acurácia de 0.35

```
print(classification_report(y_dados_test, previsoes))
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| I | 0.67 | 0.50 | 0.57 | 4 |
| II | 0.31 | 0.53 | 0.39 | 15 |
| III | 0.36 | 0.25 | 0.30 | 16 |
| IV | 0.00 | 0.00 | 0.00 | 5 |
| accuracy | | | 0.35 | 40 |
| macro avg | 0.33 | 0.32 | 0.31 | 40 |
| weighted avg | 0.33 | 0.35 | 0.32 | 40 |

Figura 6.12: Relatório de classificação do Cenário C1-01, com acurácia de 0.35

| | Precisão | Revocação | F1-Score | Suporte |
|---|---|---|---|---|
| **I** | 0.67 | 0.50 | 0.57 | 4 |
| **II** | 0.31 | 0.53 | 0.39 | 15 |
| **III** | 0.36 | 0.25 | 0.30 | 16 |
| **IV** | 0.00 | 0.00 | 0.00 | 5 |
| **Acurácia** | | | 0.35 | 40 |
| **Média Aritmética** | 0.33 | 0.32 | 0.31 | 40 |
| **Média ponderada** | 0.33 | 0.35 | 0.32 | 40 |

Tabela 6.1: Relatório de classificação do Cenário C1-01, com acurácia de 0.35

**B) C1-02 RandomForestClassifier - Unbalanced (acurácia = 0.475)**

*No segundo experimento do cenário 1 conforme a figura 6.13 e a tabela 6.2, foi encontrada uma acurácia de 0.475, com leve aumento comparado ao cenário anterior. Embora tenha tido uma piora na classe I, todas as demais classes tiveram um progresso tanto no na precisão (precision) quanto na revogação (recall). É possível observar que na classe IV o y_test selecionou amostras dessa classe, possibilitando um aumento em seu score. Em suma o maior f1-score foi da classe II com 0.54 e o menor da classe I com 0.25.*

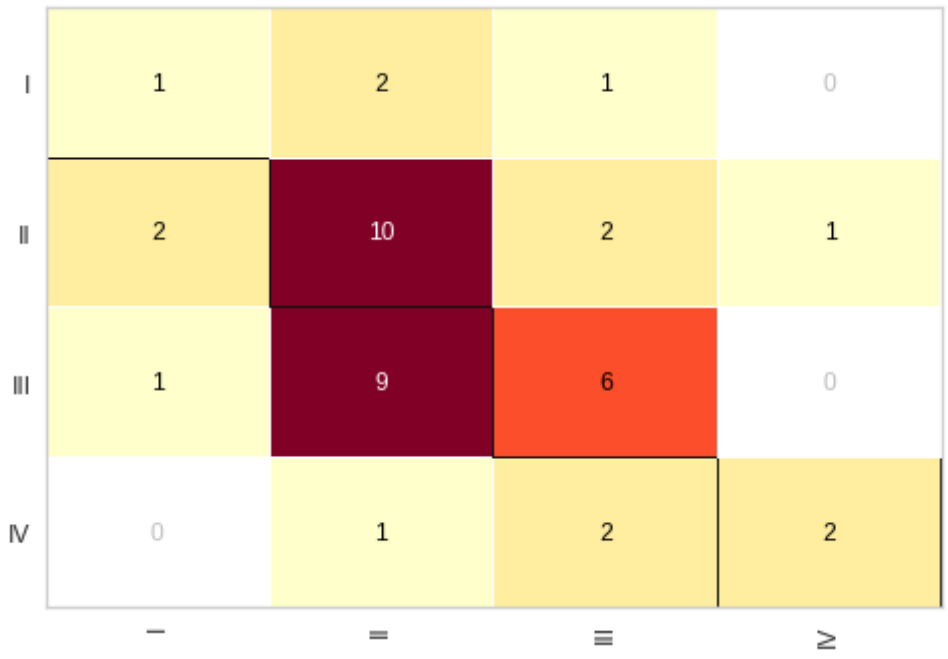



Figura 6.13: Matriz de confusão do Cenário C1-02, com acurácia de 0.475

| | Precisão | Revocação | F1-Score | Suporte |
|---|---|---|---|---|
| **I** | 0.25 | 0.25 | 0.25 | 4 |
| **II** | 0.45 | 0.67 | 0.54 | 15 |
| **III** | 0.55 | 0.38 | 0.44 | 16 |
| **IV** | 0.67 | 0.40 | 0.50 | 5 |
| **Acurácia** | | | 0.48 | 40 |
| **Média Aritmética** | 0.48 | 0.42 | 0.43 | 40 |
| **Média ponderada** | 0.50 | 0.47 | 0.47 | 40 |

Tabela 6.2: Relatório de classificação do Cenário C1-02, com acurácia de 0.475

**C) C1-03 SVM - Unbalanced (acurácia = 0.55)**

*No terceiro experimento do cenário 1 conforme a figura 6.14 e a tabela 6.3 foi encontrada com uma acurácia de 0.55, a maior do cenário 1. Entretanto, ao observar a classe I e IV, percebe-se que não foram selecionadas amostras dessas classes, prejudicando o score do algoritmo. É possível observar que o alto valor do recall da classe II elevou o valor o valor final da acurácia. Em suma o maior f1-score foi encontrado na classe II com 0.65 e o menor das classes I e IV com 0.00.*

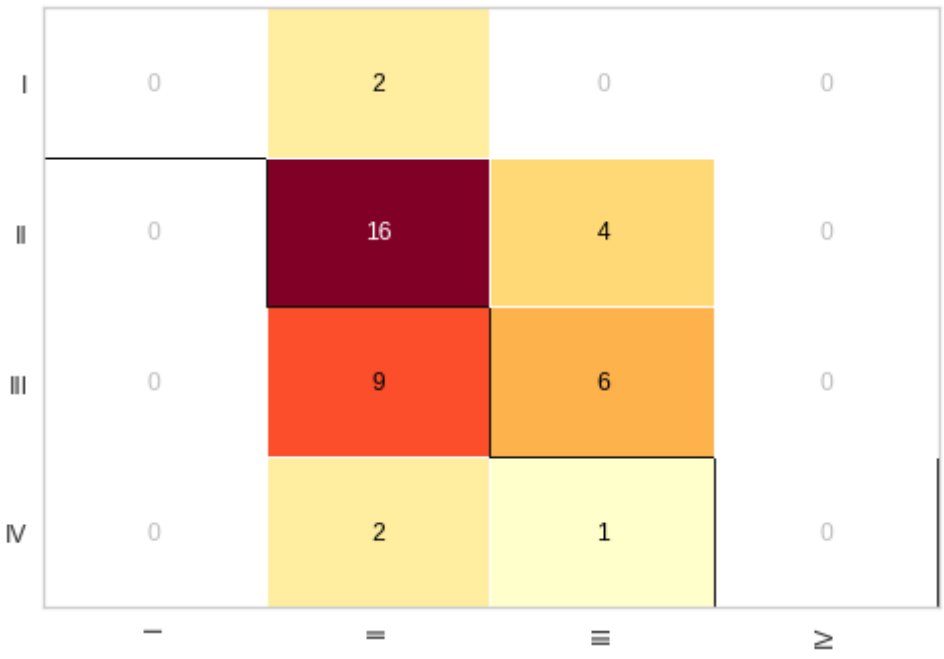



Figura 6.14: Matriz de confusão do Cenário C1-03, com acurácia de 0.55

| | Precisão | Revocação | F1-Score | Suporte |
|---|---|---|---|---|
| **I** | 0.00 | 0.00 | 0.00 | 2 |
| **II** | 0.55 | 0.80 | 0.65 | 20 |
| **III** | 0.55 | 0.40 | 0.46 | 15 |
| **IV** | 0.00 | 0.00 | 0.00 | 3 |
| **Acurácia** | | | 0.55 | 40 |
| **Média Aritmética** | 0.27 | 0.30 | 0.28 | 40 |
| **Média ponderada** | 0.48 | 0.55 | 0.50 | 40 |

Tabela 6.3: Relatório de classificação do Cenário C1-03, com acurácia de 0.55

### 6.3.3 Cenário 2: Biomarcadores urinários sem (idade, sexo) e com balanceamento de dados

#### A) C2-01 KnnClassifier - Balanced (acurácia = 0.62)

*No primeiro experimento do cenário 2 conforme a figura 6.15 e a tabela 6.4, foi encontrada uma acurácia de 0.62 bem superior a técnica equivalente ao primeiro cenário. Além disso, é possível notar na matriz de confusão uma melhor distribuição das classes, I e IV sem que fosse afetadas as classificações da classe II e III. Em suma, o maior f1-score foi da classe IV com 0.79 e o menor da classe III com 0.40.*

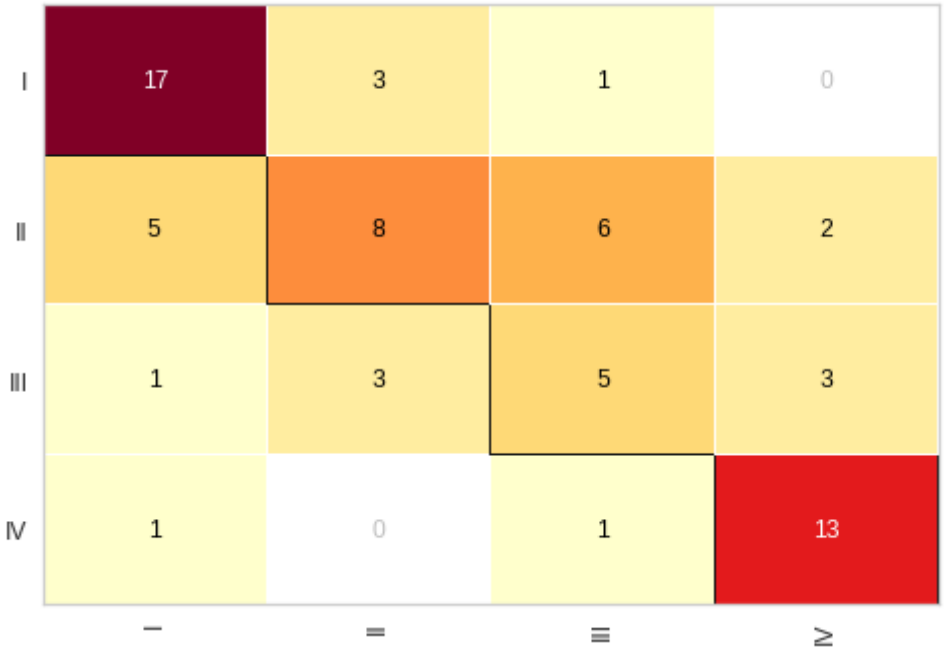



Figura 6.15: Matriz de confusão do Cenário C2-01, com acurácia de 0.62

| | Precisão | Revocação | F1-Score | Suporte |
|---|---|---|---|---|
| **I** | 0.71 | 0.81 | 0.76 | 21 |
| **II** | 0.57 | 0.38 | 0.46 | 21 |
| **III** | 0.38 | 0.42 | 0.40 | 12 |
| **IV** | 0.72 | 0.87 | 0.79 | 15 |
| **Acurácia** | | | 0.62 | 69 |
| **Média Aritmética** | 0.60 | 0.62 | 0.60 | 69 |
| **Média ponderada** | 0.61 | 0.62 | 0.61 | 69 |

Tabela 6.4: Relatório de classificação do Cenário C2-01, com acurácia de 0.62

**B) C2-02 RandomForestClassifier - Balanced (acurácia = 0.75)**

*No segundo experimento do cenário 2 conforme a figura 6.16 e a tabela 6.5 foi encontrada uma acurácia de 0.75, sendo um aumento expressivo comparado aos cenários anteriores. É possível observar seus impactos em todos os scores. Em suma, o maior f1-score foi da classe I com 0.89 e o menor da classe II com 0.56.*

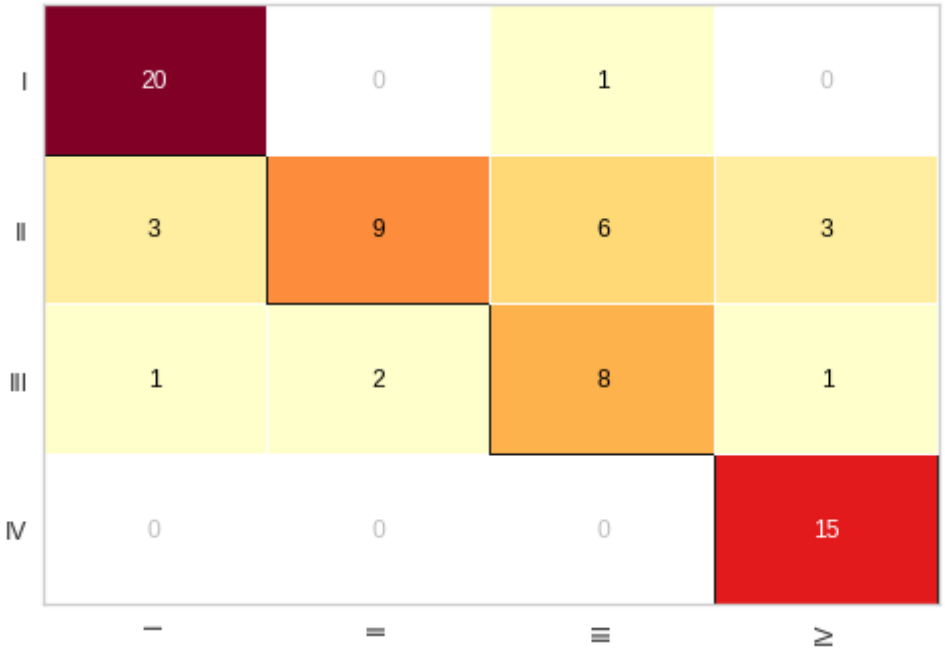



Figura 6.16: Matriz de confusão do Cenário C2-02, com acurácia de 0.75

| | Precisão | Revocação | F1-Score | Suporte |
|---|---|---|---|---|
| **I** | 0.83 | 0.95 | 0.89 | 21 |
| **II** | 0.82 | 0.43 | 0.56 | 21 |
| **III** | 0.53 | 0.67 | 0.59 | 12 |
| **IV** | 0.79 | 1.00 | 0.88 | 15 |
| **Acurácia** | | | 0.75 | 69 |
| **Média Aritmética** | 0.74 | 0.76 | 0.73 | 69 |
| **Média ponderada** | 0.77 | 0.75 | 0.74 | 69 |

Tabela 6.5: Relatório de classificação do Cenário C2-02, com acurácia de 0.75

**C) C2-03 SVM - Balanced (acurácia = 0.52)**

*No terceiro experimento do cenário 2 conforme a figura 6.17 e a tabela 6.6 foi encontrada uma acurácia de 0.52, menor do que o cenário 1 para o mesma técnica de ML e a menor do cenário 2 se comparada com as demais técnicas. No entanto, não houve diferenças significativas neste algoritmo mesmo com o balanceamento dos dados. Um fator a se destacar é o baixo recall no cenário III. Em suma o maior f1-score foi da classe IV com 0.68 e o menor das classes III com 0.23.*

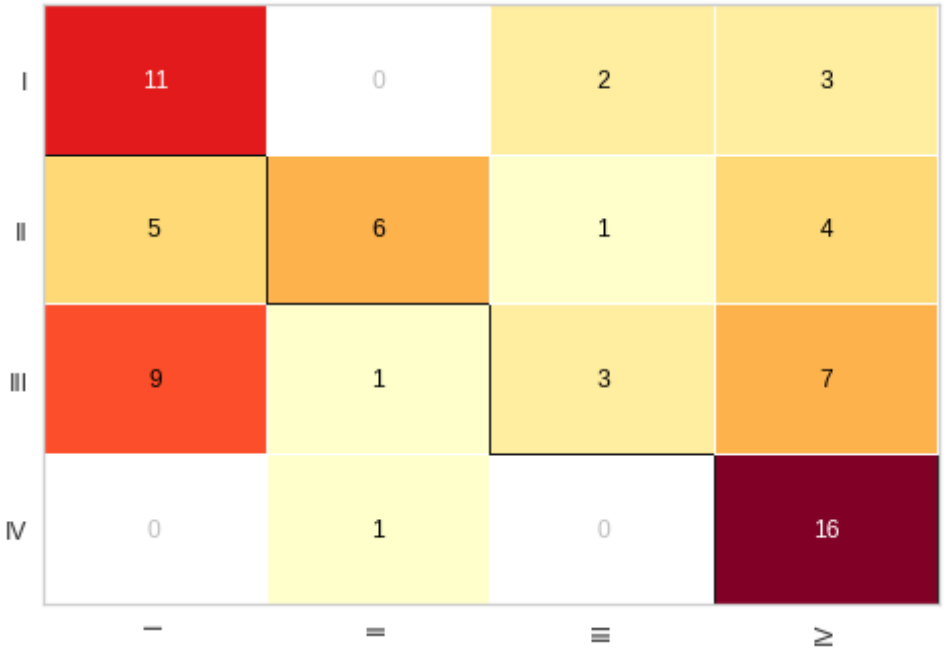



Figura 6.17: Matriz de confusão do Cenário C2-03, com acurácia de 0.52

| | Precisão | Revocação | F1-Score | Suporte |
|---|---|---|---|---|
| **I** | 0.44 | 0.69 | 0.54 | 16 |
| **II** | 0.75 | 0.38 | 0.50 | 16 |
| **III** | 0.50 | 0.15 | 0.23 | 20 |
| **IV** | 0.53 | 0.94 | 0.68 | 17 |
| **Acurácia** | | | 0.52 | 69 |
| **Média Aritmética** | 0.56 | 0.54 | 0.49 | 69 |
| **Média ponderada** | 0.55 | 0.52 | 0.48 | 69 |

Tabela 6.6: Relatório de classificação do Cenário C2-03, com acurácia de 0.52

### 6.3.4 Cenário 3: Biomarcadores urinários com (idade, sexo) e sem balanceamento

#### A) C3-01 KnnClassifier - Unbalanced (acurácia = 0.425)

*No primeiro experimento do cenário 3 conforme a figura 6.18 e a tabela 6.7, foi encontrada uma acurácia de 0.425. É possível notar que o desbalanceamento de dados tem selecionado das classes números totais distintos, o que afeta a analise do algoritmo neste cenário. A classe I*

*não realizou nenhuma classificação (em duas amostras possíveis) e a classe IV tem a precisão 1.0 mas classificou um única amostra correta possível. Em suma, o maior f1-score foi da classe II com 0.52 e o menor da classe I com 0.00.*

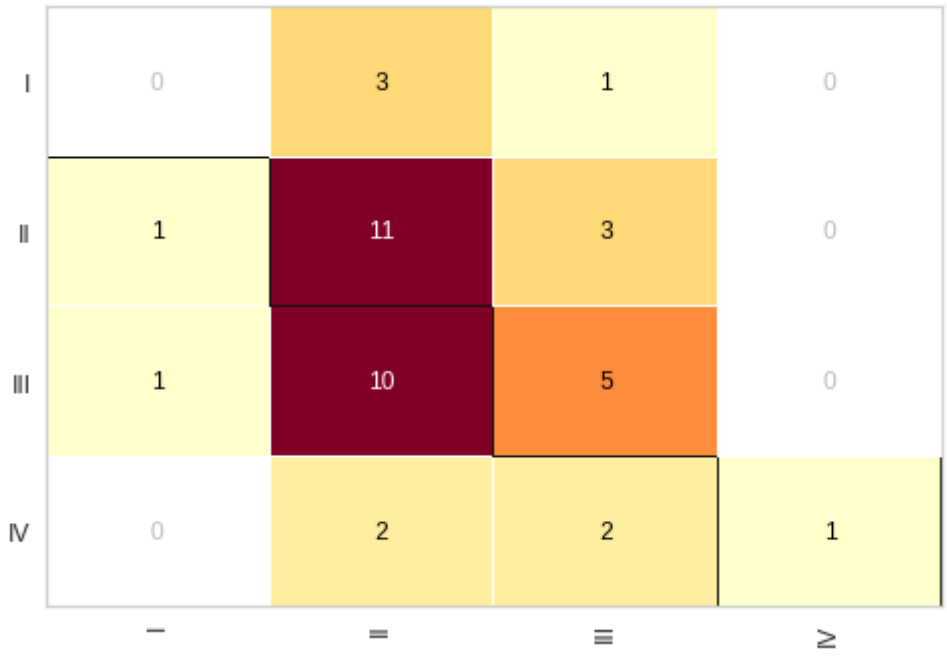



Figura 6.18: Matriz de confusão do Cenário C3-01, com acurácia de 0.425

| | Precisão | Revocação | F1-Score | Suporte |
|---|---|---|---|---|
| **I** | 0.00 | 0.00 | 0.00 | 4 |
| **II** | 0.42 | 0.73 | 0.54 | 15 |
| **III** | 0.45 | 0.31 | 0.37 | 16 |
| **IV** | 1.00 | 0.20 | 0.33 | 5 |
| **Acurácia** | | | 0.42 | 40 |
| **Média Aritmética** | 0.47 | 0.31 | 0.31 | 40 |
| **Média ponderada** | 0.47 | 0.42 | 0.39 | 40 |

Tabela 6.7: Relatório de classificação do Cenário C3-01, com acurácia de 0.425

**B) C3-02 RandomForestClassifier - Unbalanced (acurácia = 0.57)**

*No segundo experimento do cenário 3 conforme a figura 6.19 e a tabela 6.8, foi encontrada uma acurácia de 0.57, menor do que o cenário 2 para o mesmo algoritmo, no entanto, é o melhor resultado dos cenários não balanceados. Já é esperado um baixo número de amostras*

*das classes I e IV o que justifica o baixo recall da primeira classe e o baixo recall da segunda mesmo com o precision em 1.0. Em suma, o maior f1-score foi da classe II com 0.54 e o menor da classe I com 0.25.*

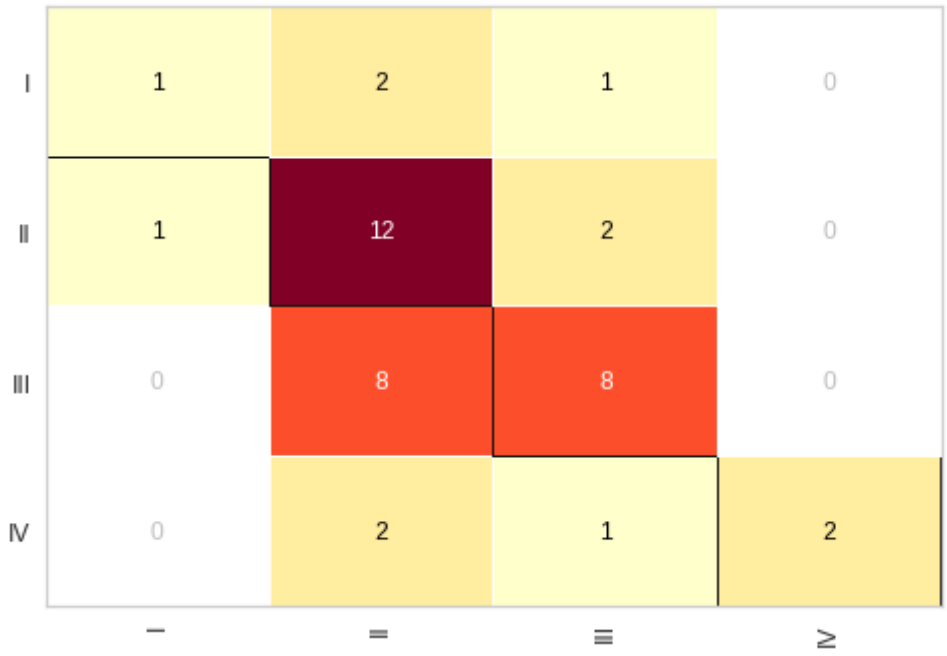



Figura 6.19: Matriz de confusão do Cenário C3-02, com acurácia de 0.57

| | Precisão | Revocação | F1-Score | Suporte |
|---|---|---|---|---|
| **I** | 0.50 | 0.25 | 0.33 | 4 |
| **II** | 0.50 | 0.80 | 0.62 | 15 |
| **III** | 0.67 | 0.50 | 0.57 | 16 |
| **IV** | 1.00 | 0.40 | 0.57 | 5 |
| **Acurácia** | | | 0.57 | 40 |
| **Média Aritmética** | 0.67 | 0.49 | 0.52 | 40 |
| **Média ponderada** | 0.63 | 0.57 | 0.56 | 40 |

Tabela 6.8: Relatório de classificação do Cenário C3-02, com acurácia de 0.57

**C) C3-03 SVM - Unbalanced (acurácia = 0.57)**

*No terceiro experimento do cenário 3 conforme a figura 6.20 e a tabela 6.9, foi encontrada uma acurácia de 0.57. É possível notar que a macro avg é mais baixa no relatório de classificação, na matriz de confusão é ainda mais notória o desbalanceamento dos dados, pois as poucas*

*amostras das classes I e IV foram erroneamente classificadas. Em suma, o maior f1-score foi da classe II com 0.68 e o menor das classes I e IV com 0.00.*

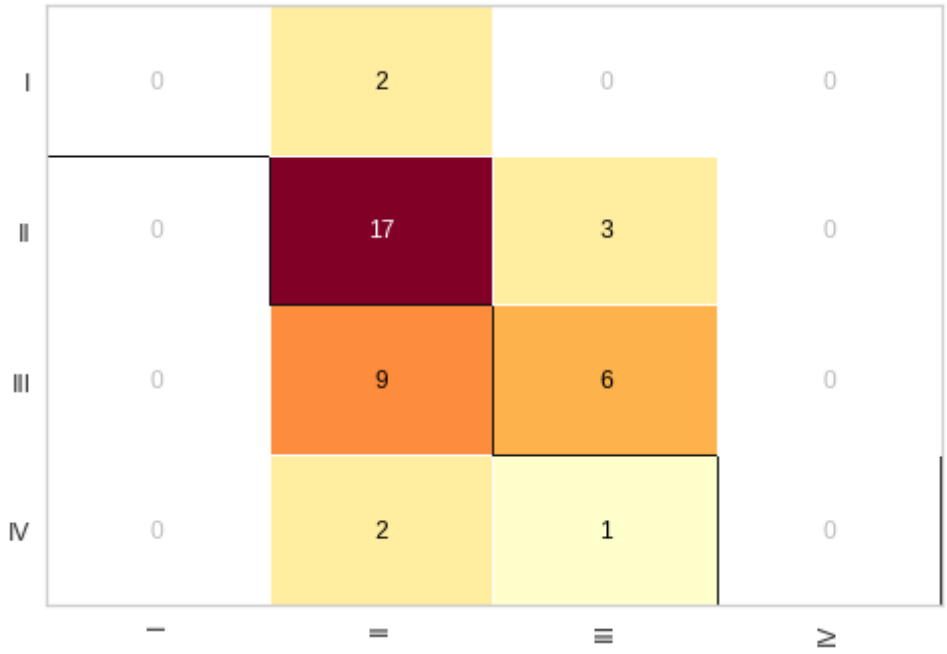



Figura 6.20: Matriz de confusão do Cenário C3-03, com acurácia de 0.57

| | Precisão | Revocação | F1-Score | Suporte |
|---|---|---|---|---|
| **I** | 0.00 | 0.00 | 0.00 | 2 |
| **II** | 0.57 | 0.85 | 0.68 | 20 |
| **III** | 0.60 | 0.40 | 0.48 | 15 |
| **IV** | 0.00 | 0.00 | 0.00 | 3 |
| **Acurácia** | | | 0.57 | 40 |
| **Média Aritmética** | 0.29 | 0.31 | 0.29 | 40 |
| **Média ponderada** | 0.51 | 0.57 | 0.52 | 40 |

Tabela 6.9: Relatório de classificação do Cenário C3-03, com acurácia de 0.57

### 6.3.5 Cenário 4: Biomarcadores urinários com (idade, sexo) e com balanceamento de dados

#### A) C4-01 KnnClassifier - Balanced (acurácia = 0.61)

*No primeiro experimento do cenário 4 conforme a figura 6.21 e tabela 6.10, foi encontrada uma acurácia de 0.61. É possível notar um espalhamento maior e melhora na classificação. Com*

*uma maior quantidade de amostras, a classe I e IV tem apresentado melhores resultados. Em suma, o maior f1-score foi da classe I com 0.74 e o menor da classe IV com 0.40.*

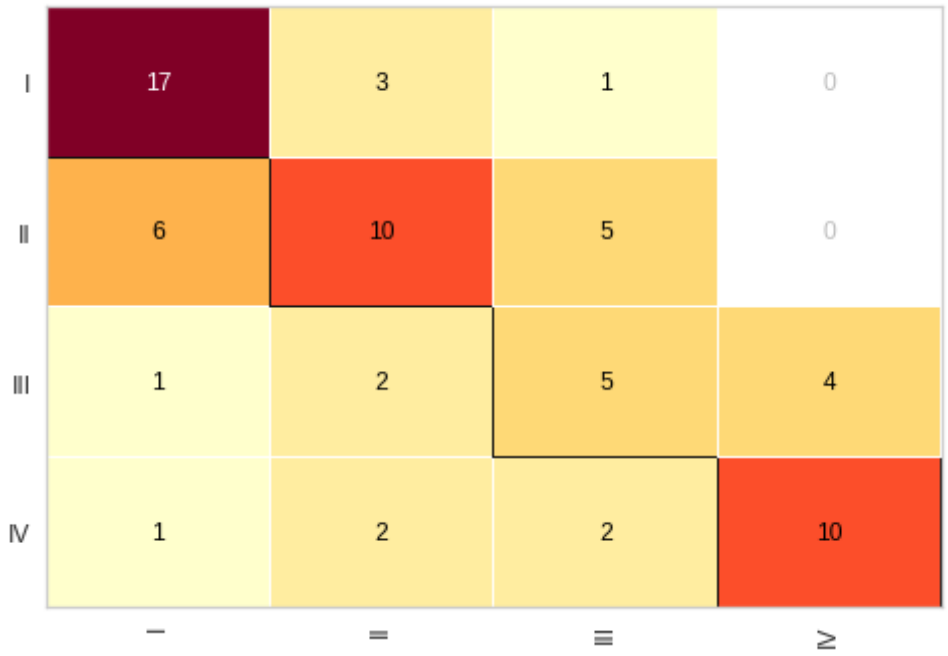



Figura 6.21: Matriz de confusão do Cenário C4-01, com acurácia de 0.61

| | Precisão | Revocação | F1-Score | Suporte |
|---|---|---|---|---|
| **I** | 0.68 | 0.81 | 0.74 | 21 |
| **II** | 0.59 | 0.48 | 0.53 | 21 |
| **III** | 0.38 | 0.42 | 0.40 | 12 |
| **IV** | 0.71 | 0.67 | 0.69 | 15 |
| **Acurácia** | | | 0.61 | 69 |
| **Média Aritmética** | 0.59 | 0.59 | 0.59 | 69 |
| **Média ponderada** | 0.61 | 0.61 | 0.60 | 69 |

Tabela 6.10: Relatório de classificação do Cenário C4-01, com acurácia de 0.61

**B) C4-02 RandomForestClassifier - Balanced (acurácia = 0.81)**

*No segundo experimento do cenário 4 conforme a figura 6.22 e a tabela 6.11, foi encontrada a maior acurácia de 0.81, fator que é refletido na matriz de confusão. O balanceamento de dados aumentou o desempenho de classes que não tiveram nenhum ajuste. Em suma, o maior f1-score foi da classe I com 0.90 e o menor da classe II com 0.67.*

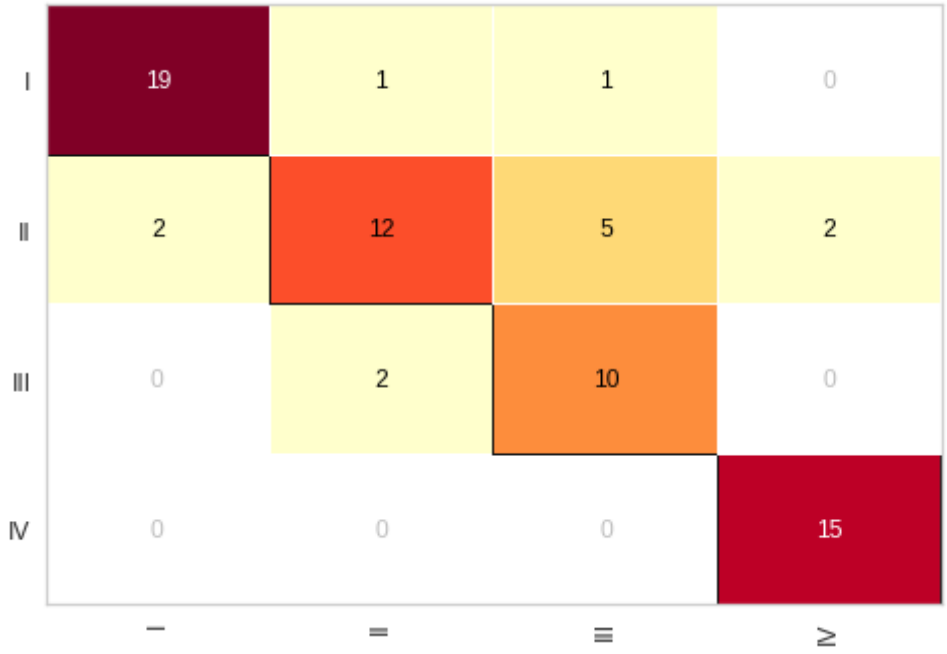



Figura 6.22: Matriz de confusão do Cenário C4-02, com acurácia de 0.81

| | Precisão | Revocação | F1-Score | Suporte |
|---|---|---|---|---|
| **I** | 0.90 | 0.90 | 0.90 | 21 |
| **II** | 0.80 | 0.57 | 0.67 | 21 |
| **III** | 0.62 | 0.83 | 0.71 | 12 |
| **IV** | 0.88 | 1.00 | 0.94 | 15 |
| **Acurácia** | | | 0.81 | 69 |
| **Média Aritmética** | 0.80 | 0.83 | 0.81 | 69 |
| **Média ponderada** | 0.82 | 0.81 | 0.81 | 69 |

Tabela 6.11: Relatório de classificação do Cenário C4-02, com acurácia de 0.81

**C) C4-03 SVM - Balanced (acurácia = 0.58)**

*No terceiro experimento do cenário 4 conforme a figura 6.23 e tabela a 6.12, foi encontrada uma acurácia de 0.58. Os falsos positivos da classe III puxaram a acurácia para baixo. Em suma, o maior f1-score foi da classe IV com 0.67 e o menor das classes III com 0.35.*

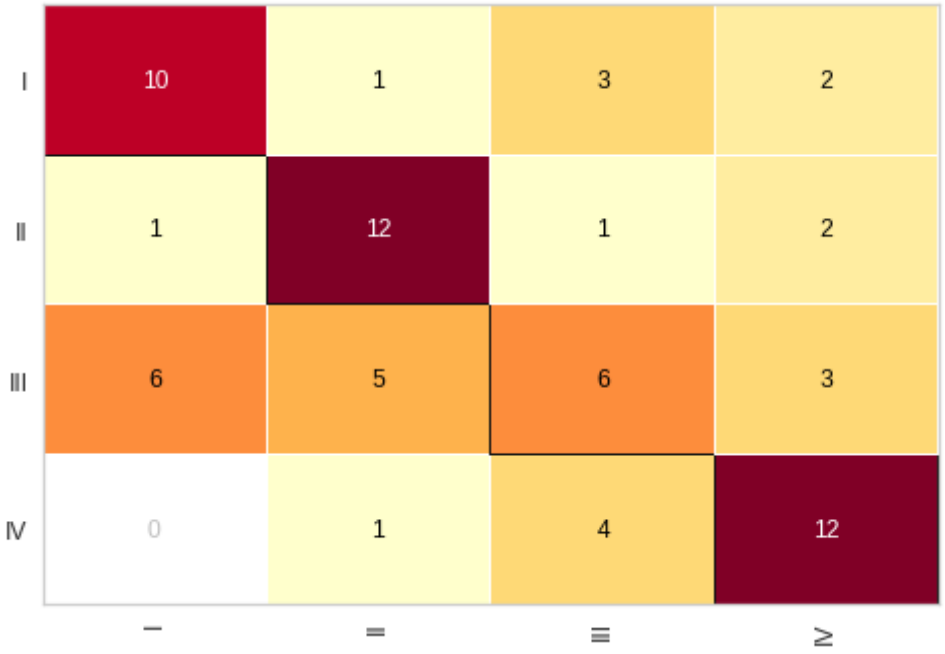



Figura 6.23: Matriz de confusão do Cenário C4-03, com acurácia de 0.58

| | Precisão | Revocação | F1-Score | Suporte |
|---|---|---|---|---|
| **I** | 0.59 | 0.62 | 0.61 | 16 |
| **II** | 0.63 | 0.75 | 0.69 | 16 |
| **III** | 0.43 | 0.30 | 0.35 | 20 |
| **IV** | 0.63 | 0.71 | 0.67 | 17 |
| **Acurácia** | | | 0.58 | 69 |
| **Média Aritmética** | 0.57 | 0.60 | 0.58 | 69 |
| **Média ponderada** | 0.56 | 0.58 | 0.57 | 69 |

Tabela 6.12: Relatório de classificação do Cenário C4-03, com acurácia de 0.58

# 7

# Discussão

## 7.1 Implicações

*Esta seção discorre sobre os resultados encontrados no experimento em busca das respostas as hipóteses dessa pesquisa.*

### 7.1.1 Desbalanceamento de Dados

*Claramente o desbalanceamento de dados conforme Fig. 6.2 afetou o desempenho dos algoritmos classificadores. Por exemplo, nos cenários 1 e 3 (que representam os cenários desbalanceados), as classes I e IV que são as classes com maior desbalanceamento apresentam um baixo número de amostras nos testes de todos experimentos destes cenários. Esses cenários contaram com o total de 40 amostras no y_test sendo que o número de amostras das classes I e IV somados representam apenas 9 amostras, ou seja, 22.5%, enquanto as outras duas classes II e III cada uma tinha em média 38.75% das amostras, que somadas representam 77.5%,. Em outras palavras, isso explica a baixa acurácia para essas duas classes e a baixa da acurácia geral nos cenários desbalanceados.*

*Entretanto é possível notar os algoritmos Random Forest e KNN chegarem a acurácia de 0.57, mesmo com o grande desbalanceamento dos dados, ou seja, o que nos leva ao questionamento se é possível melhorar o desempenho desses classificadores com um maior número de amostras.*

### 7.1.2 Balanceamento de dados

*Claramente o desbalanceamento dos dados implicou em um baixo resultado nos cenários onde não foram aplicadas técnicas de oversampling. No entanto, mesmo nessas bases, o algoritmo de classificação Random Forest e KNN apresentaram resultados promissores dado o contexto.*

*Diante desses resultados ainda na fase de experimentos, implementamos novos cenários visando avaliar o comportamento do algoritmos na classificação dos estadiamento e assim responder se é possível ou não utilizar-se de biomarcadores que podem ou não ser utilizados no diagnóstico dos CPs. Entretanto, ainda que esses biomarcadores não sejam aplicados para o diagnóstico, resta a pergunta se é possível usa-los para acompanhamento e evolução do estadiamento.*

*Aplicamos o balanceamento de dados nos cenários 2 e 4, sendo notório o melhor desempenhos dos 3 algoritmos comparando-os aos experimentos sem balanceamento. O desempenho foi de uma acurácia no SVM de 0.58, no KNN 0.62 e no Random Forest chegando a 0.75 no cenário 3 e 0.81 no cenário 4. Mesmos nas classes onde o número de amostras não foi aumentada, dado o melhor treinamento da IA, o desempenho dessa classe também levou uma melhor acurácia.*

*Os resultados são promissores, entretanto, já destacamos que eles também apontam a necessidade de mais pesquisas utilizando mais biomarcadores e um maior número de amostras para que a resposta de pesquisa deste trabalho tenha um grau precisão ainda mais assertivo. O uso de algoritmos de IA mais sofisticados podem apresentar resultados ainda melhores, elevando o valor da acurácia da classificação da variável alvo se comparada as técnicas mais básicas. Com a utilização de algoritmos de Random Forest, foi possível alcançar uma acurácia de 0.81 utilizando os biomarcadores urinários juntamente com a idade e sexo.*

### 7.1.3 O uso dos biomarcadores somados a idade e sexo dos pacientes tem alguma diferença no uso exclusivos dos biomarcadores?

*Dado o baixo número de variáveis, foram considerados 6 biomarcadores, além de idade e sexo. Todavia, separamos os cenários com apenas os biomarcadores e outro com os biomarcadores acrescidos da idade e sexo, visando avaliar se a idade e sexo teriam alguma influencia na classificação e por sua vez na acurácia. Os resultados das acurácias nos cenários gerados foi superior em quase todos os cenários que continham a idade e sexo, a única exceção apresentada não teve uma diferença significativa.*

# 8

# Conclusão

## 8.1 Recapitulação

*O Pâncreas é uma glândula hibrida, possuindo uma função endócrina que representa apenas 2% da composição celular deste órgão e também possui uma função exócrina responsável pelos 98% restantes. Suas principais funções respectivamente são: controle da glicose no sangue e produção do suco pancreático. Por sua localização mais interna no abdômen humano, as complicações pancreáticas só comumente se apresentam quando não é possível mais um cura. Dentre os vários possíveis problemas que afetam essa glândula destaca-se o Câncer de Pâncreas, pois, embora raro, é uma das neoplasias mais letais que existem. Das neoplasias pancreáticas, o PDAC é responsável por 90% dos casos, por essa razão um maior volume de pesquisas relacionadas ao CP são especializadas nesse tipo.*

*Por sua elevada taxa de mortalidade, a maioria desses estudos se concentram no diagnóstico prévio. Além da falta de sintomas mais expressivos nas fases iniciais, outro fator que corrobora para não detecção precoce do PDAC é que seu método padrão de detecção é por meio de exames de imagens específicos e por fim a biopsia. No entanto, os estudos atuais relacionados ao PDAC e aos demais tipos de cânceres de pâncreas tem se concentrado na descoberta de* ***biomarcadores*** *para um diagnostico precoce, pois sabe-se, que mesmo sem o desenvolvimento de novas técnicas para o tratamento do PDAC, se for possível o diagnóstico precocemente do PDAC por detecção de biomarcadores encontrados no sangue, na urina ou nas fezes, o baixo custo e a alta oferta desses tipos de exames, certamente aumentariam as taxas de sobrevida e sucesso na resseção e por sua vez a cura definitiva do PDAC. Entretanto, poucos estudos tem buscado uma avanço em um precoce prognóstico. É de comum conheci-*

*mento que para iniciar algum tipo de tratamento das neoplasias, faz-se necessária uma analise do estadiamento do paciente, por vezes, o diagnostico só ocorre quando resta poucos meses de sobrevida. A definição do estadiamento inicial pode demorar pouco mais. Quanto mais precoce for realizado o estadiamento, mais tempo e opções de tratamento e/ou paliação estarão disponíveis para o paciente, e ainda que sobrevida dos pacientes seja maior, utilização de uma estimativa de estadiamento baseada em biomarcadores urinários, além de diminuir os custos de acompanhamento, pode se tornar a única forma de acompanhamento médico em lugares com poucos recursos, dada a facilidade e baixo custo destes exames em comparação com os de imagem.*

## 8.2 Contribuições

1. *Contribui com a resposta a ciência que é possível utilizar biomarcadores urinários combinados para prévio estadiamento do grau da neoplasia;*
2. *Diminuição dos custos de acompanhamento do PDAC;*
3. *Contribui para o acompanhamento médico da evolução dos pacientes com PDAC;*
4. *Contribui com um prévio prognóstico do PDAC.*

## 8.3 Limitações

1. *Limitações do baixo número de amostras do Dataset, apenas 199 de PDAC no total.*
2. *Limitações da falta de informações clinicas que podem vir completar a proposta e auxiliar no aperfeiçoamento da precisão na classificação.*
3. *Desbalanceamento do Dataset: o desbalanceamento do número de amostras da classe I e IV em comparação as bases II e III. Embora tenha sido aplicada técnica para resolução de sobreamostragem, tanto o desbalanceamento como as técnicas de sobreamostragem podem afetar acurácia.*
4. *Após aplicar técnicas de balanceamento, ficamos limitados as poucas amostras disponíveis, dado que as amostras geradas foram baseadas nas amostras existentes.*
5. *A limitação de datasets com dados sobre CP disponíveis de forma gratuita.*

6. *Classificação apenas do Grau do TNM que tem como base o tamanho do carcinoma, não atendendo a todas subdivisões do TNM, limitando assim a previsão do estadiamento Grau do TNM.*

## 8.4 Trabalhos Futuros

1. *Utilizar outros biomarcadores que podem apresentar uma maior acurácia.*
2. *Submeter os algoritmos aqui utilizados em uma amostra maior.*
3. *Utilizar algoritmos de ML não supervisionados.*
4. *Segmentar os dados em buscar das subclasses do TNM.*

## 8.5 Conclusão

*O estudo realizado teve como foco a aplicação da área da computação de aprendizagem de máquina como proposta de solução para o problema de estadiamento prévio do TNM para prognóstico de PDAC. Este tipo de estadiamento se dá somente após o diagnóstico do câncer de pâncreas e exames de imagens para realização da classificação.*

*Neste trabalho, apresentamos o resultado da avaliação do uso dos algoritmos de ML como KNN, RandomForest e SVM para prognóstico de previsão de estadiamento com base no TNM para PDAC em biomarcadores urinários sem o uso de exames de imagens. O resultado dos achados científicos foram utilizados com base para definição do objeto de estudo desenvolvido no trabalho aqui proposto.*

*A metodologia seguida contou com uma revisão exploratória da narrativa sobre os principais temas da pesquisa, o desenvolvimento e realização dos experimentos ocorreram no google colabory, com utilização da linguagem python3.*

*O dataset de dados utilizados foi extraído do estudo sobre diagnóstico prévio do PDAC com base em biomarcadores urinários (Debernardi et al., 2020). As amostras utilizadas foram recolhidas de vários centros de saúde.*

*Os principais resultados indicam que a aplicação dos algoritmos de classificação foram capazes de estimar com certo grau de precisão a classificação do grau de TNM (I, II, III e IV) que são relacionados principalmente ao tamanho do tumor, permitindo assim estimar o estadiamento dos pacientes, direcionando as demais ações de combate e paliação da neoplasia e diminuindo os custos dos cuidados oncológicos.*

*Além disso, os resultados sugerem que o uso de uma amostra maior e a aplicação de outros algoritmos podem apresentar uma acurácia mais aprimorada.*

*Com a acurácia de 0.81 do algoritmo RandomForest, entendemos que está dissertação tem uma proposição positiva, alcançando o objetivo desta pesquisa.*

# Referências


*Elliot A. Asare, Elizabeth G. Grubbs, Jeffrey E. Gershenwald, Frederick L. Greene, and Thomas A. Aloia. Setting the "stage" for surgical oncology fellows: Pierre denoix and tnm staging.* Journal of surgical oncology*, 119(7):823–823, 2019. ISSN 0022-4790.*

*Walter F. Boron and Emile L. Boulpaep.* Medical Physiology: Principles for Clinical Medicine*. Saunders Elsevier, 2009.*

*Leo Breiman. Random forests.* Machine learning*, 45(1):5–32, 2001.*

*T. Clark Brelje and Robert L. Sorenson. Exocrine pancreas and ductal system.* Histology Guide*, 2021. URL https://histologyguide.com/EM-view/EM-074-interlobular-duct/12-photo-1.html. Acessado em julho de 2022.*

*Encyclopaedia Britannica. Islets of langerhans | definition, function, location, & facts.* Encyclopaedia Britannica*, 2021. URL https://www.britannica.com/science/islets-of-Langerhans. Acessado em julho de 2022.*

*Christoph Brock et al.* The Pancreas: An Integrated Textbook of Basic Science, Medicine, and Surgery*. Blackwell Publishing, 2003.*

*Stewart BW and Wild CP.* World Cancer Report 2014*. World Health Organization, International Agency for Research on Cancer, 2014. ISBN 978-92-832-0429-9 978-92-832-0443-5 978-92-832-0432-9. URL https://publications.iarc.fr/Non-Series-Publications/World-Cancer-Reports/World-Cancer-Report-2014.*

*National Cancer Institute. Cancer of the Pancreas - Cancer Stat Facts, September 2022. URL https://seer.cancer.gov/statfacts/html/pancreas.html.*

*Comitê de Nomenclatura de Genes HUGO. Relatório de símbolos de genes | Comitê de Nomenclatura de Genes HUGO, 2022. URL https://www.genenames.org/data/*

gene-symbol-report/#!/hgnc_id/HGNC:9952. https://www.genenames.org/data/gene-symbol-report/#!/hgnc_id/HGNC:9952 *[acessado em 28 de agosto de 2022].*

*Fabiano Santos Conrado. MScript2, August 2022a. URL* https://github.com/fsconrado/MScriptA. *original-date: 2022-08-26T11:18:54Z.*

*Fabiano Santos Conrado. MScript1, August 2022b. URL* https://github.com/fsconrado/MScriptB. *original-date: 2022-08-25T21:17:57Z.*

*Corinna Cortes and Vladimir Vapnik. Support-vector networks.* Mach Learn*, 20(3):273–297, September 1995. ISSN 0885-6125, 1573-0565.* **DOI** 10.1007/BF00994018*. URL* http://link.springer.com/10.1007/BF00994018*.*

*Silvana Debernardi, Harrison O'Brien, Asma S. Algahmdi, Nuria Malats, Grant D. Stewart, Marija Plješa-Ercegovac, Eithne Costello, William Greenhalf, Amina Saad, Rhiannon Roberts, Alexander Ney, Stephen P. Pereira, Hemant M. Kocher, Stephen Duffy, Oleg Blyuss, and Tatjana Crnogorac-Jurcevic. A combination of urinary biomarker panel and PancRISK score for earlier detection of pancreatic cancer: A case–control study.* PLOS Medicine*, 17 (12):e1003489, 2020. ISSN 1549-1676.* **DOI** 10.1371/journal.pmed.1003489*. URL* https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.1003489*. Publisher: Public Library of Science.*

*Vanessa Sardinha dos Santos. Pâncreas. Características e funções do pâncreas, 2022. URL* https://brasilescola.uol.com.br/biologia/pancreas.htm*.*

*Evelyn Fix and J. L. Hodges. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties.* International Statistical Review / Revue Internationale de Statistique*, 57(3):238–247, 1989. ISSN 0306-7734.* **DOI** 10.2307/1403797*. URL* https://www.jstor.org/stable/1403797*. Publisher: [Wiley, International Statistical Institute (ISI)].*

*Su Kah Goh, Grace Gold, Christopher Christophi, and Vijayaragavan Muralidharan. Serum carbohydrate antigen 19-9 in pancreatic adenocarcinoma: a mini review for surgeons.* ANZ journal of surgery*, 87(12):987–992, 2017. ISSN 1445-1433.*

*Fred S. Gorelick and John D. Jamieson.* Structure-function relationships in the pancreatic acinar cell*, pages 1341–1360. Academic Press, 2012.*

*Arthur C. Guyton and John E. Hall.* Guyton and Hall Textbook of Medical Physiology*. Elsevier Health Sciences, 2016.*

*Tin Kam Ho. Wayback Machine, April 2016. URL* https://web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf.

*Kazufumi Honda, Michimoto Kobayashi, Takuji Okusaka, Jo Ann Rinaudo, Ying Huang, Tracey Marsh, Mitsuaki Sanada, Yoshiyuki Sasajima, Shoji Nakamori, Masashi Shimahara, Takaaki Ueno, Akihiko Tsuchida, Naohiro Sata, Tatsuya Ioka, Yohichi Yasunami, Tomoo Kosuge, Nami Miura, Masahiro Kamita, Takako Sakamoto, Hirokazu Shoji, Giman Jung, Sudhir Srivastava, and Tesshi Yamada. Plasma biomarker for detection of early stage pancreatic cancer and risk factors for pancreatic malignancy using antibodies for apolipoprotein-AII isoforms.* Sci Rep*, 5(1):15921, November 2015. ISSN 2045-2322.* **DOI** *10.1038/srep15921. URL* https://www.nature.com/articles/srep15921*. Number: 1 Publisher: Nature Publishing Group.*

*INCA. Tipos de câncer | INCA - Instituto Nacional de Câncer, September 2022. URL* https://www.inca.gov.br/tipos-de-cancer/cancer-de-pancreas.

*Steven R. James, Patricia Smith, and George H. White. Pancreatic islet physiology and pathophysiology.* Journal of Endocrinology and Metabolism*, 107:1012–1024, 2018.*

*James Jamieson. Anatomy and histology of the pancreas.* Pancreapedia*, 2021. URL* https://www.pancreapedia.org/research/acinar-cell-structure*. Acessado em julho de 2022.*

*Barbara Kenner, Suresh T Chari, David Kelsen, David S Klimstra, Stephen J Pandol, Michael Rosenthal, Anil K Rustgi, James A Taylor, Adam Yala, Noura Abul-Husn, Dana K Andersen, David Bernstein, Søren Brunak, Marcia Irene Canto, Yonina C Eldar, Elliot K Fishman, Julie Fleshman, Vay Liang W Go, Jane M Holt, Bruce Field, Ann Goldberg, William Hoos, Christine Iacobuzio-Donahue, Debiao Li, Graham Lidgard, Anirban Maitra, Lynn M Matrisian, Sung Poblete, Laura Rothschild, Chris Sander, Lawrence H Schwartz, Uri Shalit, Sudhir Srivastava, and Brian Wolpin. Artificial intelligence and early detection of pancreatic cancer: 2020 summative review.* Pancreas*, 50(3):251–279, 2021. ISSN 0885-3177.*

*Lumen Learning. The endocrine pancreas.* Anatomy and Physiology II*, 2021. URL* https://courses.lumenlearning.com/suny-ap2/chapter/the-endocrine-pancreas/*. Acessado em julho de 2022.*

*Peter Lesko, Michal Chovanec, and Michal Mego. Biomarkers of disease recurrence in stage I testicular germ cell tumours.* Nat Rev Urol*, pages 1–22, August 2022. ISSN 1759-4820.* **DOI** *10.1038/s41585-022-00624-y. URL* https://www.nature.com/articles/s41585-022-00624-y*. Publisher: Nature Publishing Group.*

*Po Sing Leung.* The Renin-Angiotensin System: Current Research Progress in the Pancreas*. Springer, 2010.*

*MSD Manual. Pancreas - digestive system.* MSD Manual Consumer Version*, 2021. URL* https://www.msdmanuals.com/home/digestive-disorders/biology-of-the-digestive-system/pancreas*. Acessado em julho de 2022.*

*Johns Hopkins Medicine. The digestive process: What is the role of your pancreas in digestion?* Johns Hopkins Medicine*, 2021a. URL* https://www.hopkinsmedicine.org/health/conditions-and-diseases/the-digestive-process-what-is-the-role-of-your-pancreas-in-digestion*. Acessado em julho de 2022.*

*Libretexts Medicine. Overview of pancreatic islets.* LibreTexts*, 2021b. URL* https://med.libretexts.org/Bookshelves/Anatomy_and_Physiology/Book%3A_Human_Biology_(Wakim_and_Grewal)/15%3A_The_Endocrine_System/15.11%3A_Overview_of_Pancreatic_Islets*. Acessado em julho de 2022.*

*Yoshihiro Minamiya, Hideki Kawai, Hajime Saito, Manabu Ito, Yukiko Hosono, Satoru Motoyama, Yoshihisa Katayose, Naoko Takahashi, and Jun-ichi Ogawa. Reg1a expression is an independent factor predictive of poor prognosis in patients with non-small cell lung cancer.* Lung Cancer*, 60:98–104, 2008.* ***DOI*** *10.1016/j.lungcan.2007.09.012.*

*Tobias D. Müller, Brian Finan, Christoffer Clemmensen, Richard D. DiMarchi, and Matthias H. Tschöp. Glucagon physiology and pathophysiology.* Cell Metabolism*, 25(2):545–557, 2017.*

*Grasieli de Oliveira.* Meta-análise integrativa secretoma-proteoma para identificação de potenciais biomarcadores de adenocarcinoma ductal pancreático*. PhD thesis, Universidade Estadual Paulista (UNESP), São Paulo, 2020. URL* https://repositorio.unesp.br/handle/11449/192618*.*

*Ghim Siong Ow and Vladimir A. Kuznetsov. Big genomics and clinical data analytics strategies for precision cancer prognosis.* Scientific Reports*, 6(1):36493, November 2016. ISSN 2045-2322.* ***DOI*** *10.1038/srep36493. URL* https://doi.org/10.1038/srep36493*.*

*Pancreapedia. Vasoactive intestinal polypeptide or vip.* Pancreapedia*, 2021. URL* https://www.pancreapedia.org/research/vasoactive-intestinal-polypeptide*. Acessado em julho de 2022.*

*R M Parry, W. Jones, T H Stokes, J H Phan, R A Moffitt, H. Fang, L. Shi, A. Oberthuer, M. Fischer, W. Tong, and M D Wang. k-Nearest neighbor models for microarray gene*

*expression analysis and clinical outcome prediction.* The Pharmacogenomics Journal*, 10(4): 292–309, August 2010. ISSN 1473-1150.* **DOI** *10.1038/tpj.2010.56. URL* https://doi.org/10.1038/tpj.2010.56.

*Thomas P Radon, Nicholas J Massat, Rachel Jones, Wadah Alrawashdeh, Laetitia Dumartin, Daniel Ennis, S William Duffy, Hemant M Kocher, Stephen P Pereira, Luis Guarner, et al. Identification of a three-biomarker panel in urine for early detection of pancreatic adenocarcinoma.* Clinical Cancer Research*, 21(15):3512–3521, 2015.* **DOI** *10.1158/1078-0432.CCR-14-2467.*

*Jonathan G. Richens, Ciarán M. Lee, and Saurabh Johri. Improving the accuracy of medical diagnosis with causal machine learning.* Nature Communications*, 11(1):3923, August 2020. ISSN 2041-1723.* **DOI** *10.1038/s41467-020-17419-7. URL* https://doi.org/10.1038/s41467-020-17419-7.

*Emma Ruiz, Pedro Hernandez, and Raul Alvarez. The insulin signaling pathway: Mechanisms of action and regulation.* Journal of Molecular Endocrinology*, 54(1):R17–R31, 2015.*

*Abigail Shaw, Marcus D Bradley, Sean Elyan, and Kathreena M Kurian. Tumour biomarkers: diagnostic, prognostic, and predictive.* BMJ (Online)*, 351:h3449–h3449, 2015. ISSN 0959-8138.*

*American Cancer Society. Pancreas anatomy, 2022. URL* https://www.cancer.org/content/dam/cancer-org/images/illustrations/medical-illustrations/en/pancreas-anatomy.gif/jcr:content/renditions/aem-thumbnail-980-980.jpeg.

*Mônica Soldan. Rastreamento do câncer de pâncreas.* Rev. Col. Bras. Cir.*, 44:109–111, April 2017. ISSN 0100-6991, 1809-4546.* **DOI** *10.1590/0100-69912017002015. URL* http://www.scielo.br/j/rcbc/a/pQqHW6scP4yK3QySPJvbLSH/?lang=pt. *Publisher: Colégio Brasileiro de Cirurgiões.*

*S Sulochana and T Sivakami. A gross morphological study of the pancreas in human cadavers.* National Journal of Clinical Anatomy*, 1(1):1–5, 2012.* **DOI** *10.1055/s-0039-3401669.*

*Hyuna Sung, Jacques Ferlay, Rebecca L. Siegel, Mathieu Laversanne, Isabelle Soerjomataram, Ahmedin Jemal, and Freddie Bray. Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries.* CA: A Cancer Journal for Clinicians*, 71(3):209–249, 2021. ISSN 1542-4863.* **DOI** *10.3322/caac.21660. URL* https://onlinelibrary.wiley.com/doi/abs/10.3322/caac.21660. *_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.3322/caac.21660.*

*The scikit-learn developers. Nearest Neighbors Classification, 2022. URL* https://scikit-learn/stable/auto_examples/neighbors/plot_classification.html.

*Fund international World Cancer Research. Pancreatic cancer statistics, September 2022. URL* https://www.wcrf.org/cancer-trends/pancreatic-cancer-statistics/.

*Qian Xu, Wen Xia, Xiang Zhang, Bing Peng, and Cheng Peng. Regenerating islet-derived protein 1b (reg1b): a serum biomarker candidate for pancreatic ductal adenocarcinoma.* Cancer Science*, 110(3):1023–1033, 2019.* **DOI** 10.1111/cas.13940.

*Lin Zhou, Ruifeng Zhang, Lishun Wang, Shaoming Shen, Hiroshi Okamoto, Akira Sugawara, Li Xia, Xiaoling Wang, Naoya Noguchi, Takeo Yoshikawa, Akira Uruno, Weiyan Yao, and Yaozong Yuan. Upregulation of reg ia accelerates tumor progression in pancreatic cancer with diabetes.* International Journal of Cancer*, 127:1795–1803, 2010.* **DOI** 10.1002/ijc.25188.

# 9

# Apêndices

## 9.1 Apêndices A

*Na seção de Apêndices A, encontra-se o script completo utilizado nesta dissertação. O ambiente utilizado foi o Google Colaboratory e a linguagem de programação utilizada foi Python. Além disso, é possível acessar o referido script nos links do GitHub: (Conrado, 2022a) e (Conrado, 2022b).*

# **Cenário 1:**{ Somente os Biomarcadores | Dados Não balanceados}

## **Visão dos Dados

```
#1. IMPORTAR O DATASET (COLUNAS IMPORTANTES: AGE, SEX, LYVE1, REG1B, TFF1 ==> ALVO PREVER A
#!pip install plotly --upgrade
#Baixar os dados e jogar na variável DADOS
import pandas as pd

dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")
dados = dados.fillna(1, inplace= False)
# 16 colunas
```

```
dados
```

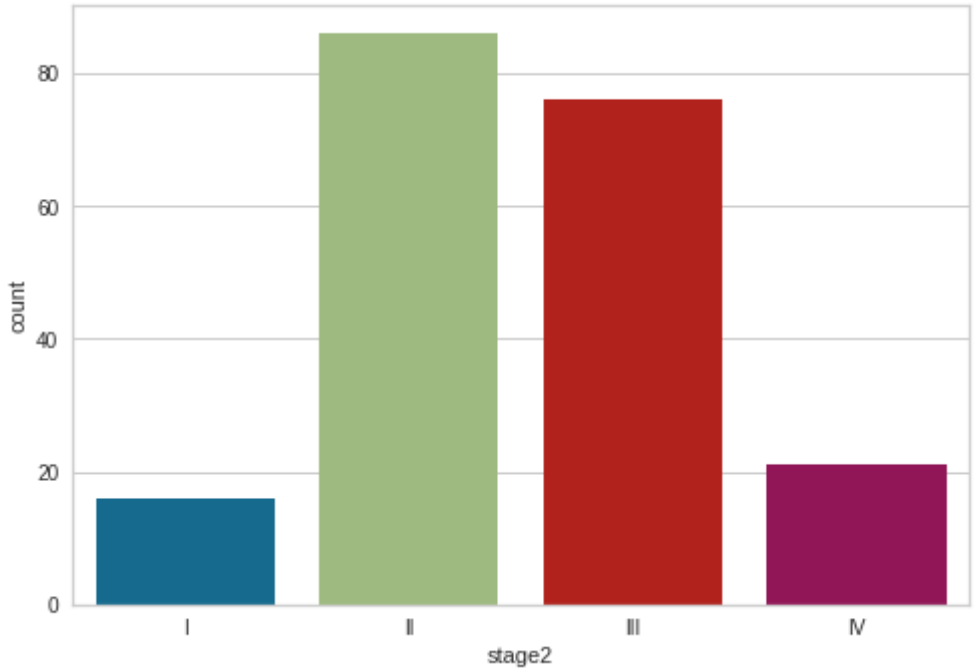
| | sample_id | patient_cohort | sample_origin | diagnosis | stage | sex | age | |
|---|---|---|---|---|---|---|---|---|
| **0** | S497 | Cohort1 | ESP | 3 | I | F | 81 | 1 |
| **1** | S456 | Cohort1 | LIV | 3 | IA | M | 57 | |
| **2** | S520 | Cohort1 | BPTB | 3 | IA | M | 55 | |
| **3** | S573 | Cohort2 | BPTB | 3 | IA | M | 58 | |
| **4** | S401 | Cohort1 | LIV | 3 | IB | M | 73 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **194** | S549 | Cohort2 | BPTB | 3 | IV | M | 68 | |
| **195** | S558 | Cohort2 | BPTB | 3 | IV | F | 71 | |
| **196** | S560 | Cohort2 | BPTB | 3 | IV | M | 63 | |
| **197** | S583 | Cohort2 | BPTB | 3 | IV | F | 75 | |
| **198** | S590 | Cohort1 | BPTB | 3 | IV | M | 74 | |

199 rows × 14 columns

```
#Avaliar o número de amostras para cada classe(estadiamento)
import numpy as np
np.unique(dados['stage2'], return_counts = True)
```

```
(array(['I', 'II', 'III', 'IV'], dtype=object), array([16, 86, 76, 21]))
```

```
#Gerar um gráfico para avaliar visualmente o número de amostras para cada classe, objetivo
import seaborn as sns
sns.countplot(x = dados['stage2']);
```




Clique duas vezes (ou pressione "Enter") para editar

## C1-01. KnnClassifier{*Unbalanced*} => 0.35

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | LYVE1    REG1B   TFF1
X_dados = dados.iloc[:,7:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

#label_encoder_sex = LabelEncoder()
#X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
#X_dados

#Onehot
#from sklearn.preprocessing import OneHotEncoder
#from sklearn.compose import ColumnTransformer
```

```
#onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remaind
#X_dados = onehotencorder.fit_transform(X_dados)
#X_dados

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler() #(with_mean=False)
X_dados = scaler_dados.fit_transform(X_dados)


#Separação dos dados SEM o balanceamento
from sklearn.model_selection import train_test_split

X_dados_train , X_dados_test, y_dados_train, y_dados_test = train_test_split(X_dados, y_dad
X_dados_train.shape, X_dados_test.shape
```

```
((159, 6), (40, 6))
```

```
#KNN CLASSIFICADOR
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix,plot_confusion_matrix
from sklearn.preprocessing import StandardScaler


#knn_modelc = KNeighborsClassifier(n_neighbors=5, metric='euclidean', p = 2)
knn_modelc = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p = 2)

knn_modelc.fit(X_dados_train, y_dados_train)
```

```
KNeighborsClassifier()
```

```
previsoes = knn_modelc.predict(X_dados_test)
previsoes
```

```
array(['II', 'II', 'II', 'III', 'II', 'II', 'II', 'III', 'I', 'II', 'III',
       'II', 'II', 'III', 'II', 'I', 'II', 'II', 'III', 'III', 'I', 'II',
       'II', 'II', 'II', 'III', 'II', 'II', 'II', 'III', 'II', 'II',
       'III', 'II', 'III', 'II', 'II', 'II', 'III', 'II'], dtype=object)
```

```
y_dados_test
```

```
array(['II', 'III', 'III', 'II', 'III', 'IV', 'I', 'III', 'I', 'III',
       'II', 'III', 'III', 'III', 'II', 'I', 'II', 'III', 'II', 'II',
       'III', 'III', 'II', 'III', 'II', 'III', 'IV', 'IV', 'II', 'II',
       'II', 'II', 'II', 'III', 'II', 'IV', 'III', 'I', 'III', 'IV'],
      dtype=object)
```

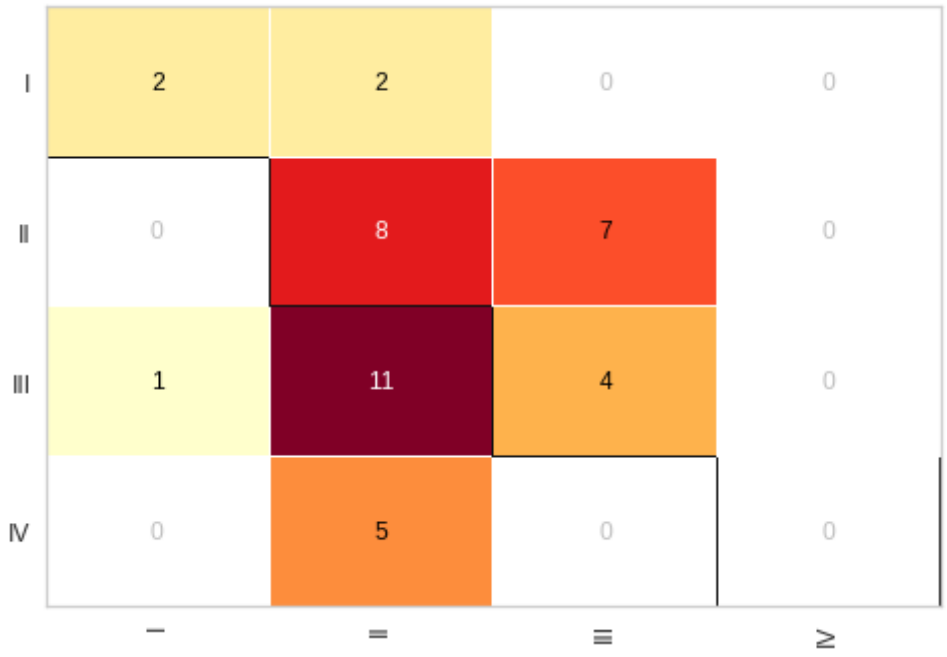
```
from sklearn.metrics import accuracy_score, classification_report
accuracy_score(y_dados_test, previsoes) # padronização
```

```
0.35
```

```
from yellowbrick.classifier import ConfusionMatrix
cm = ConfusionMatrix(knn_modelc)
```

```
cm.fit(X_dados_train, y_dados_train)
cm.score(X_dados_test, y_dados_test)
```

```
0.35
```




```
print(classification_report(y_dados_test, previsoes))
```

```
              precision    recall  f1-score   support

           I       0.67      0.50      0.57         4
          II       0.31      0.53      0.39        15
         III       0.36      0.25      0.30        16
          IV       0.00      0.00      0.00         5

    accuracy                           0.35        40
   macro avg       0.33      0.32      0.31        40
weighted avg       0.33      0.35      0.32        40

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undefi

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undefi

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undefi

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted
```

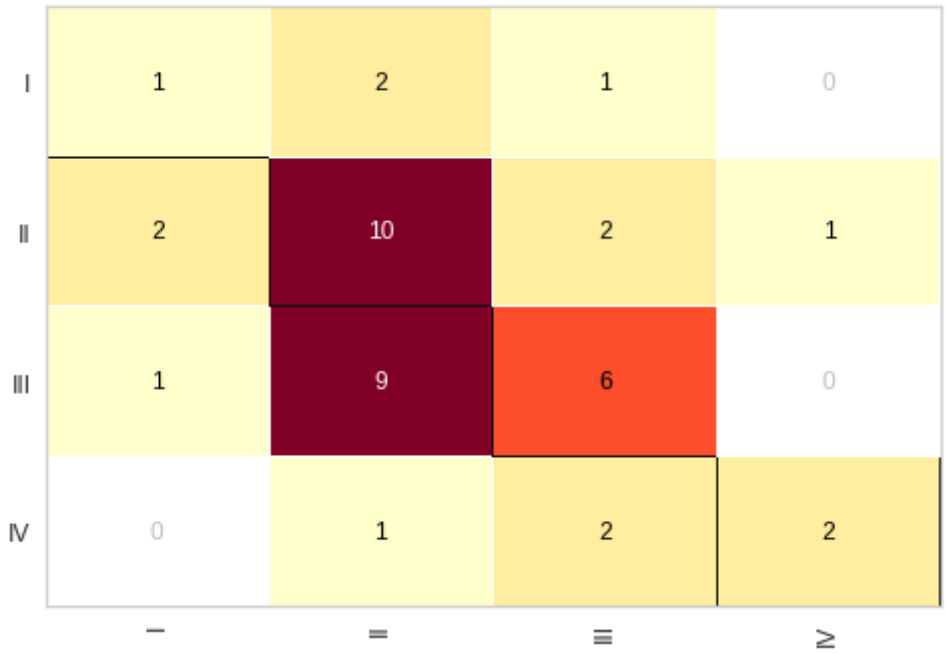
## C1-02. RandomForestClassifier{*Unbalanced*} => 0.475

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
```

```
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,7:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

#label_encoder_sex = LabelEncoder()
#X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
#X_dados

#Onehot
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

#onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remaind
#X_dados = onehotencorder.fit_transform(X_dados)
#X_dados

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler() #(with_mean=False)
X_dados = scaler_dados.fit_transform(X_dados)



#Separação dos dados SEM o balanceamento
from sklearn.model_selection import train_test_split

X_dados_train , X_dados_test, y_dados_train, y_dados_test = train_test_split(X_dados, y_dad
X_dados_train.shape, X_dados_test.shape
X_random_forest_dados_sb_teste = X_dados_test
y_random_forest_dados_sb_teste = y_dados_test

#RANDOM FOREST CLASSIFIER

from sklearn.ensemble import RandomForestClassifier
random_forest_dados_sb = RandomForestClassifier(criterion = 'entropy', min_samples_leaf =
random_forest_dados_sb.fit(X_dados_train, y_dados_train)


    RandomForestClassifier(criterion='entropy', min_samples_split=5,
                           n_estimators=200, random_state=0)


from sklearn.metrics import accuracy_score, classification_report
previsoes = random_forest_dados_sb.predict(X_dados_test)
accuracy_score(y_dados_test, previsoes)

    0.475

from yellowbrick.classifier import ConfusionMatrix
```

```
cm = ConfusionMatrix(random_forest_dados_sb)
cm.fit(X_dados_train, y_dados_train)
cm.score(X_dados_test, y_dados_test)
```

```
0.475
```




```
print(classification_report(y_dados_test, previsoes))
```

```
              precision    recall  f1-score   support

           I       0.25      0.25      0.25         4
          II       0.45      0.67      0.54        15
         III       0.55      0.38      0.44        16
          IV       0.67      0.40      0.50         5

    accuracy                           0.48        40
   macro avg       0.48      0.42      0.43        40
weighted avg       0.50      0.47      0.47        40
```

## C1-03. SVM{Unbalanced} => *0.55*, liner, 1, 1

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,7:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados
```

```
# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

#label_encoder_sex = LabelEncoder()
#X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
#X_dados

#Onehot
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

#onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remaind
#X_dados = onehotencorder.fit_transform(X_dados)
#X_dados

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler() #(with_mean=False)
X_dados = scaler_dados.fit_transform(X_dados)


#Separação dos dados SEM o balanceamento
from sklearn.model_selection import train_test_split

X_dados_train , X_dados_test, y_dados_train, y_dados_test = train_test_split(X_dados, y_dad
X_dados_train.shape, X_dados_test.shape
```

```
((159, 6), (40, 6))
```

```
from sklearn.svm import SVC



svm_dados = SVC(kernel='linear', random_state=1, C = 1.0) # 2 -> 4
svm_dados.fit(X_dados_train, y_dados_train)
```

```
SVC(kernel='linear', random_state=1)
```

```
previsoes = svm_dados.predict(X_dados_test)
previsoes
```

```
array(['II', 'II', 'II', 'II', 'III', 'III', 'II', 'II', 'III', 'II',
       'III', 'II', 'II', 'II', 'II', 'II', 'II', 'II', 'III', 'II', 'II',
       'II', 'III', 'III', 'II', 'II', 'II', 'III', 'II', 'II', 'II',
       'II', 'II', 'II', 'III', 'II', 'II', 'III', 'III', 'II'],
      dtype=object)
```

```
y_dados_test
```

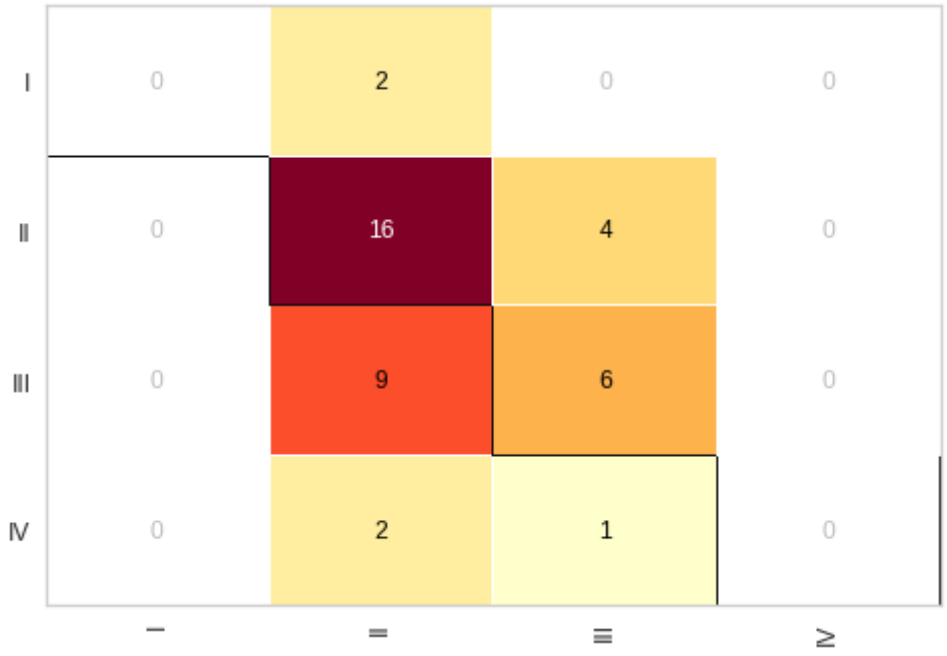
```
array(['III', 'II', 'III', 'III', 'III', 'III', 'III', 'I', 'II', 'II',
       'IV', 'II', 'II', 'II', 'II', 'III', 'I', 'II', 'III', 'III', 'IV',
       'II', 'II', 'III', 'II', 'II', 'III', 'II', 'II', 'II', 'II',
       'III', 'II', 'III', 'II', 'II', 'IV', 'III', 'III', 'II'],
      dtype=object)
```

```
from sklearn.metrics import accuracy_score, classification_report
#0.575, liner, 1, 1 | 0.425, poly, 1, 1 | #0.475, sigmoid, 1, 1 | 0.425, rbf, 1, 1
accuracy_score(y_dados_test, previsoes)
```

```
0.55
```

```
from yellowbrick.classifier import ConfusionMatrix
cm = ConfusionMatrix(svm_dados)
cm.fit(X_dados_train, y_dados_train)
cm.score(X_dados_test, y_dados_test)
```

```
0.55
```




```
print(classification_report(y_dados_test, previsoes))
```

```
                precision    recall  f1-score   support

             I       0.00      0.00      0.00         2
            II       0.55      0.80      0.65        20
           III       0.55      0.40      0.46        15
            IV       0.00      0.00      0.00         3

      accuracy                           0.55        40
     macro avg       0.27      0.30      0.28        40
  weighted avg       0.48      0.55      0.50        40

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undefi

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undefi

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undefi

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted
```

# **Cenário 2:**{ Somente os Biomarcadores | Dados Não balanceados}

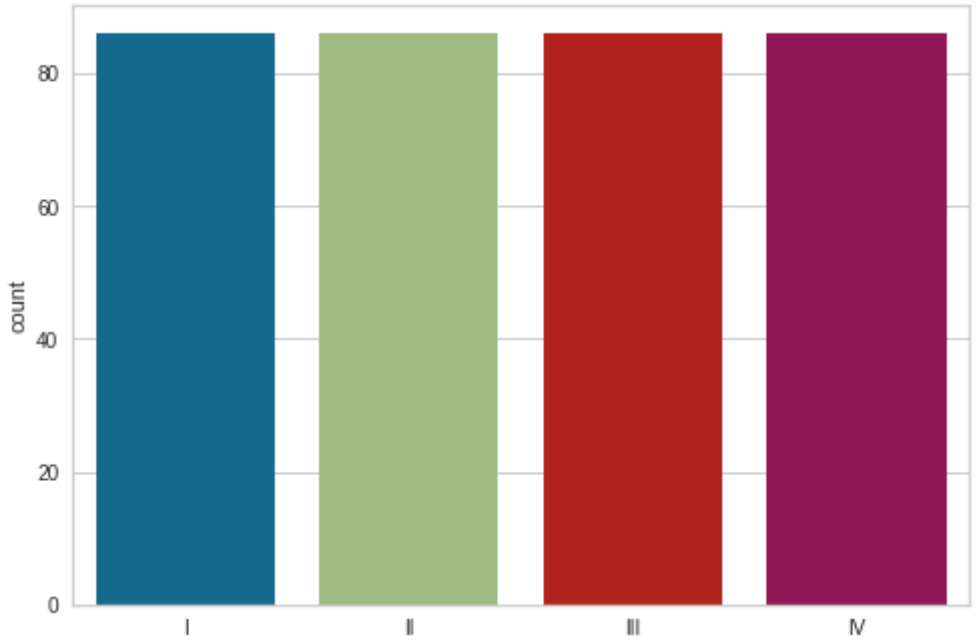
---

## **C2-01. KnnClassifier**{*Balanced*}0.623

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,7:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

#label_encoder_sex = LabelEncoder()
#X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
#X_dados

# Sobreamostragem com SMOTE
from imblearn.over_sampling import SMOTE
import numpy as np

smote = SMOTE(sampling_strategy='not majority', random_state=3)
X_over, y_over = smote.fit_resample(X_dados, y_dados)

#y_over.shape, X_over.shape
np.unique(y_dados, return_counts = True), np.unique(y_over, return_counts = True)

import seaborn as sns
sns.countplot(x = y_over);

#from sklearn.preprocessing import OneHotEncoder
#from sklearn.compose import ColumnTransformer
#onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remaind
#X_dados = onehotencorder.fit_transform(X_over).toarray()
X_dados = X_over # Se descomentar a linha de cima comentar essa.
#X_dados, X_dados.shape

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
```

```
scaler_dados = StandardScaler()
X_dados = scaler_dados.fit_transform(X_dados)
X_dados

#Separação dos dados BALANCEADOS
from sklearn.model_selection import train_test_split

X_dados_train_over, X_dados_test_over, y_dados_train_over, y_dados_test_over = train_test_s
X_dados_train_over.shape, X_dados_test_over.shape
```

```
((275, 6), (69, 6))
```




```
#Knn Classifier Balanced

from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix,plot_confusion_matrix
from sklearn.preprocessing import StandardScaler


knn_modelcb = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p = 2)
knn_modelcb.fit(X_dados_train_over, y_dados_train_over)
```

```
KNeighborsClassifier()
```

```
previsoes = knn_modelcb.predict(X_dados_test_over)
previsoes
```

```
array(['I', 'I', 'III', 'I', 'I', 'IV', 'III', 'I', 'I', 'I', 'III', 'II',
       'IV', 'IV', 'I', 'II', 'I', 'I', 'I', 'I', 'IV', 'III', 'I', 'IV',
       'I', 'I', 'III', 'I', 'II', 'I', 'I', 'IV', 'II', 'II', 'II', 'I',
       'IV', 'IV', 'II', 'II', 'III', 'II', 'III', 'IV', 'IV', 'I', 'IV',
       'IV', 'II', 'II', 'II', 'IV', 'II', 'III', 'III', 'I', 'III',
       'III', 'I', 'I', 'II', 'IV', 'III', 'IV', 'I', 'IV', 'IV', 'III',
       'IV'], dtype=object)
```

```
y_dados_test_over
```

```
array(['III', 'I', 'II', 'I', 'II', 'IV', 'III', 'I', 'I', 'II', 'II',
       'I', 'III', 'IV', 'I', 'II', 'II', 'I', 'II', 'I', 'IV', 'II', 'I',
       'III', 'I', 'I', 'III', 'II', 'II', 'I', 'I', 'IV', 'II', 'II',
```

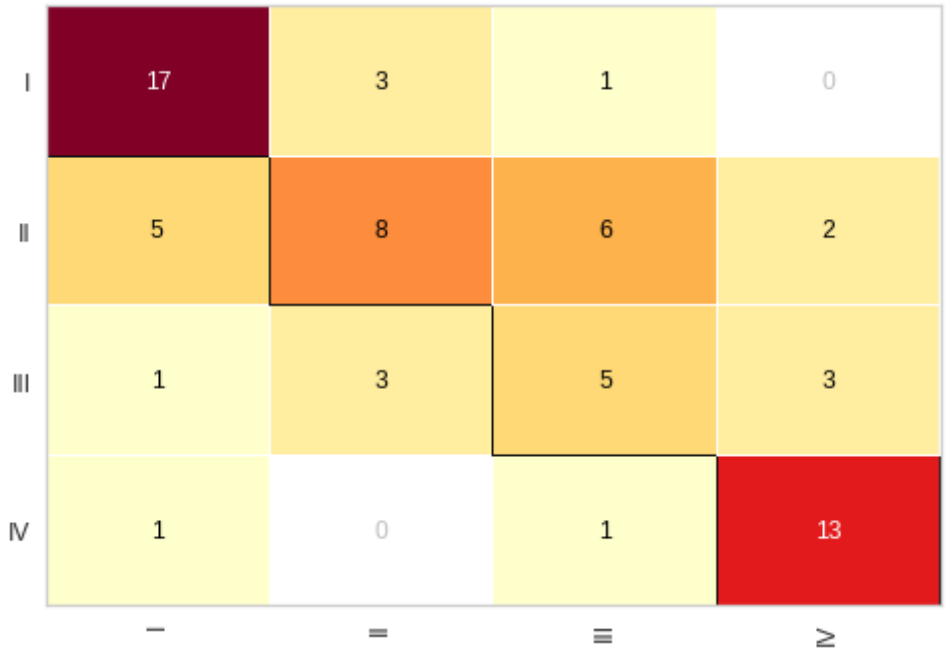
```
        'III', 'IV', 'IV', 'III', 'II', 'II', 'III', 'II', 'II', 'IV',
        'IV', 'I', 'IV', 'IV', 'III', 'I', 'II', 'II', 'III', 'III', 'III',
        'I', 'II', 'IV', 'I', 'I', 'I', 'IV', 'II', 'IV', 'I', 'IV', 'II',
        'I', 'IV'], dtype=object)
```

```
from sklearn.metrics import accuracy_score, classification_report
accuracy_score(y_dados_test_over, previsoes) # padronização
```

```
0.6231884057971014
```

```
from yellowbrick.classifier import ConfusionMatrix
cm = ConfusionMatrix(knn_modelcb)
cm.fit(X_dados_train_over, y_dados_train_over)
cm.score(X_dados_test_over, y_dados_test_over)
```

```
0.6231884057971014
```




```
print(classification_report(y_dados_test_over, previsoes))
```

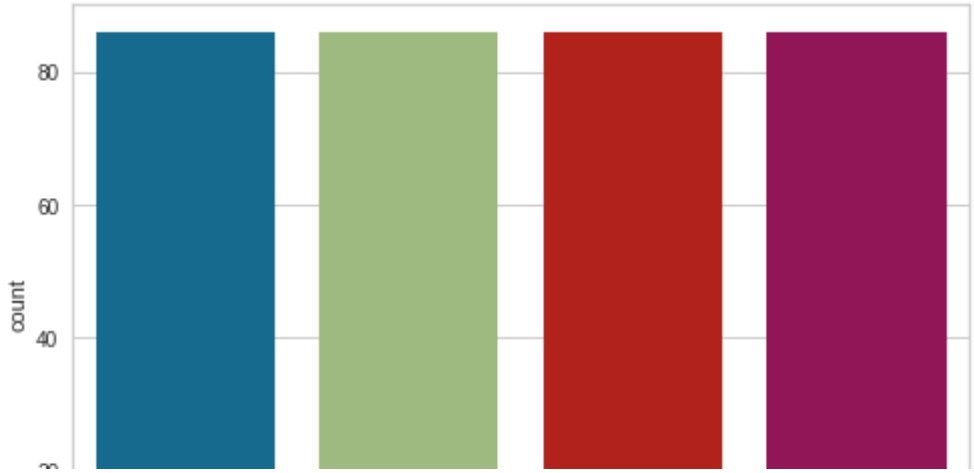
```
              precision    recall  f1-score   support

           I       0.71      0.81      0.76        21
          II       0.57      0.38      0.46        21
         III       0.38      0.42      0.40        12
          IV       0.72      0.87      0.79        15

    accuracy                           0.62        69
   macro avg       0.60      0.62      0.60        69
weighted avg       0.61      0.62      0.61        69
```

## *C2-02. RandomForestClassifier*{Balanced} => 0.753

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")
```

```
#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,7:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

#label_encoder_sex = LabelEncoder()
#X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
#X_dados

# Sobreamostragem com SMOTE

from imblearn.over_sampling import SMOTE
smote = SMOTE(sampling_strategy='not majority', random_state=29)
#9  0.7971014492753623

X_over, y_over = smote.fit_resample(X_dados, y_dados)

#y_over.shape, X_over.shape
np.unique(y_dados, return_counts = True), np.unique(y_over, return_counts = True)

import seaborn as sns
sns.countplot(x = y_over);

#from sklearn.preprocessing import OneHotEncoder
#from sklearn.compose import ColumnTransformer
#onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remaind
#X_dados = onehotencorder.fit_transform(X_over).toarray()
X_dados = X_over
#X_dados, X_dados.shape

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler()
X_dados = scaler_dados.fit_transform(X_dados)
X_dados

#Separação dos dados BALANCEADOS
from sklearn.model_selection import train_test_split

X_dados_train_over, X_dados_test_over, y_dados_train_over, y_dados_test_over = train_test_s
X_dados_train_over.shape, X_dados_test_over.shape
```

```
((275, 6), (69, 6))
```




```
from sklearn.ensemble import RandomForestClassifier #1,6,300,10 -->0.681 | 1,5,200, 17 -->0

random_forest_dados = RandomForestClassifier(criterion = 'entropy', min_samples_leaf =  1,
random_forest_dados.fit(X_dados_train_over, y_dados_train_over)
```

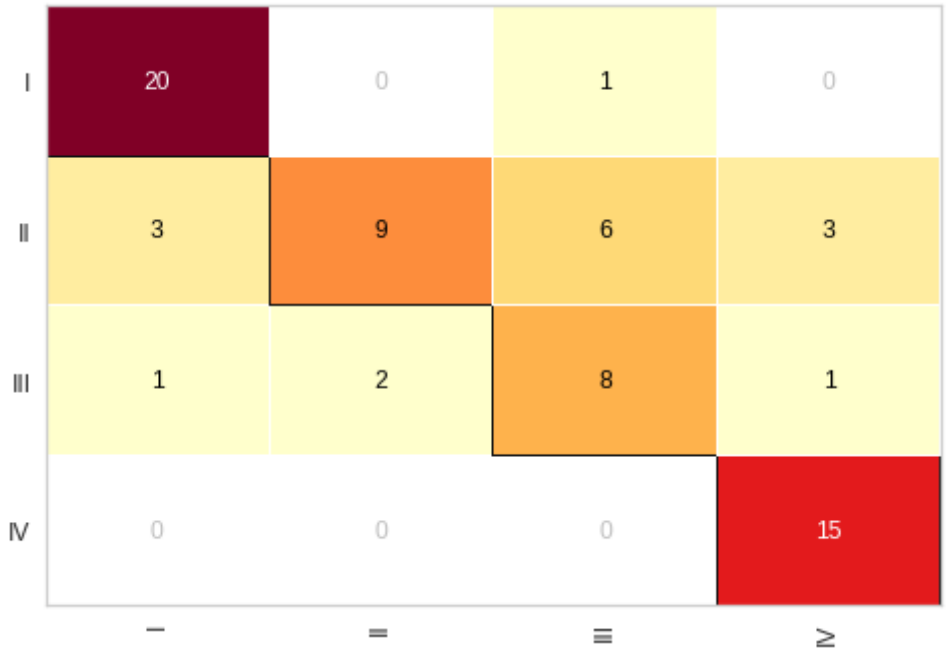
```
RandomForestClassifier(criterion='entropy', min_samples_split=5,
                       n_estimators=200, random_state=17)
```

```
from sklearn.metrics import accuracy_score, classification_report
previsoes = random_forest_dados.predict(X_dados_test_over)
accuracy_score(y_dados_test_over, previsoes)
```

```
0.7536231884057971
```

```
from yellowbrick.classifier import ConfusionMatrix
cm = ConfusionMatrix(random_forest_dados)
cm.fit(X_dados_train_over, y_dados_train_over)
cm.score(X_dados_test_over, y_dados_test_over)
```

```
0.7536231884057971
```




```
print(classification_report(y_dados_test_over, previsoes))
```

```
              precision    recall  f1-score   support
```

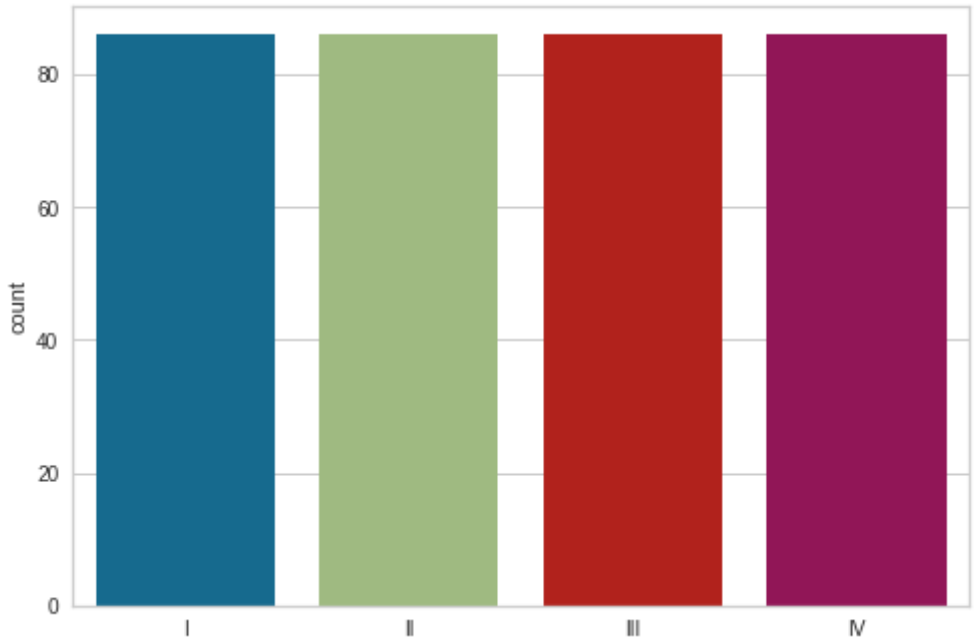
```
              I       0.83      0.95      0.89        21
             II       0.82      0.43      0.56        21
            III       0.53      0.67      0.59        12
             IV       0.79      1.00      0.88        15

       accuracy                           0.75        69
      macro avg       0.74      0.76      0.73        69
   weighted avg       0.77      0.75      0.74        69
```

## C2-03. SVM{Balanced} => 0.521, linear, 0, 10 |

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd


# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,7:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

#label_encoder_sex = LabelEncoder()
#X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
#X_dados

# Sobreamostragem com SMOTE

from imblearn.over_sampling import SMOTE
import numpy as np

lista = []
count = 1

smote = SMOTE(sampling_strategy='not majority', random_state=16)
#2 - 47
X_over, y_over = smote.fit_resample(X_dados, y_dados)

#y_over.shape, X_over.shape

np.unique(y_dados, return_counts = True), np.unique(y_over, return_counts = True)

import seaborn as sns
sns.countplot(x = y_over);
```

```
#from sklearn.preprocessing import OneHotEncoder
#from sklearn.compose import ColumnTransformer
#onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remaind
#X_dados = onehotencorder.fit_transform(X_over).toarray()
X_dados = X_over
#X_dados, X_dados.shape

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler()
X_dados = scaler_dados.fit_transform(X_dados)
X_dados

#Separação dos dados BALANCEADOS
from sklearn.model_selection import train_test_split

X_dados_train_over, X_dados_test_over, y_dados_train_over, y_dados_test_over = train_test_s
X_dados_train_over.shape, X_dados_test_over.shape
np.unique(y_dados, return_counts = True), np.unique(y_over, return_counts = True)
```

```
((array(['I', 'II', 'III', 'IV'], dtype=object), array([16, 86, 76, 21])),
 (array(['I', 'II', 'III', 'IV'], dtype=object), array([86, 86, 86, 86])))
```




```
from sklearn.svm import SVC


svm_dados = SVC(kernel='linear', random_state=0, C = 10.0) # 2 -> 4
svm_dados.fit(X_dados_train_over, y_dados_train_over)
```

```
SVC(C=10.0, kernel='linear', random_state=0)
```

```
previsoes = svm_dados.predict(X_dados_test_over)
previsoes
```

```
array(['IV', 'IV', 'IV', 'IV', 'IV', 'I', 'I', 'I', 'IV', 'I', 'I', 'IV',
       'IV', 'I', 'III', 'II', 'III', 'IV', 'II', 'II', 'I', 'I', 'IV',
       'IV', 'I', 'I', 'IV', 'I', 'IV', 'IV', 'I', 'IV', 'I', 'I', 'IV',
       'IV', 'III', 'IV', 'IV', 'I', 'I', 'IV', 'I', 'I', 'IV', 'III',
       'III', 'IV', 'IV', 'III', 'IV', 'IV', 'I', 'II', 'IV', 'I', 'II',
       'IV', 'II', 'I', 'I', 'I', 'IV', 'IV', 'IV', 'I', 'II', 'I', 'II'],
      dtype=object)
```

Clique duas vezes (ou pressione "Enter") para editar

```
y_dados_test_over
```

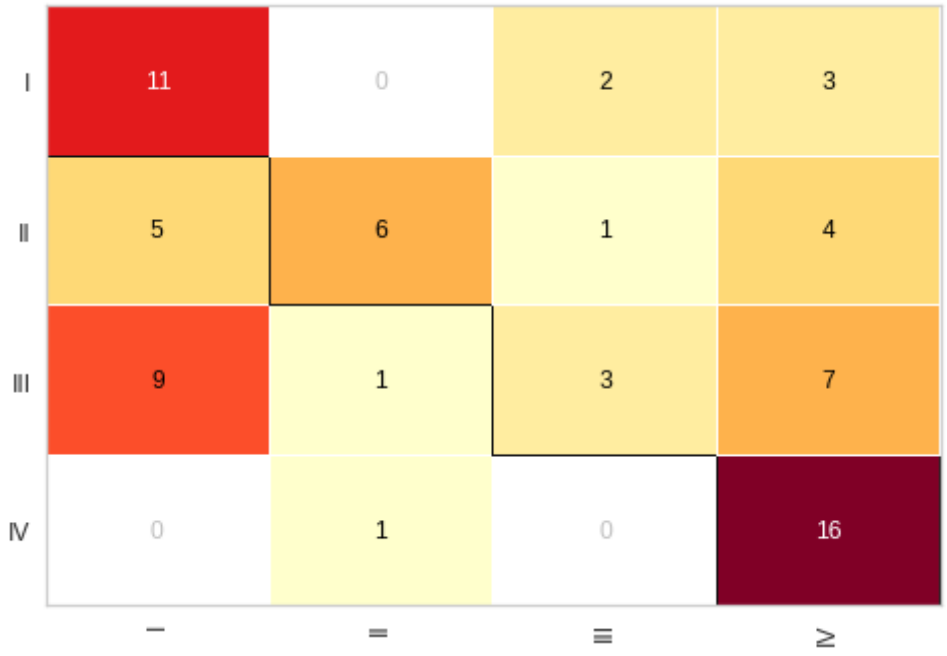
```
array(['IV', 'IV', 'IV', 'III', 'IV', 'I', 'III', 'I', 'III', 'I', 'III',
       'I', 'IV', 'III', 'II', 'II', 'I', 'I', 'II', 'II', 'I', 'I', 'IV',
       'IV', 'II', 'I', 'I', 'III', 'IV', 'II', 'III', 'III', 'I', 'III',
       'III', 'III', 'III', 'II', 'IV', 'II', 'II', 'III', 'III', 'I',
       'IV', 'I', 'III', 'IV', 'IV', 'III', 'III', 'IV', 'III', 'III',
       'IV', 'I', 'IV', 'II', 'II', 'I', 'III', 'II', 'IV', 'IV', 'II',
       'II', 'II', 'I', 'II'], dtype=object)
```

```
from sklearn.metrics import accuracy_score, classification_report
#0.475, liner, 1, 1 | 0.425, poly, 1, 1 | #0.475, sigmoid, 1, 1 | 0.425, rbf, 1, 1
#0.434, rbf 1, 2 |0.521 linear, 0, 10

accuracy_score(y_dados_test_over, previsoes)
```

```
0.5217391304347826
```

```
from yellowbrick.classifier import ConfusionMatrix
cm = ConfusionMatrix(svm_dados)
cm.fit(X_dados_train_over, y_dados_train_over)
cm.score(X_dados_test_over, y_dados_test_over)
```

```
0.5217391304347826
```




```
print(classification_report(y_dados_test_over, previsoes))
```

```
              precision    recall  f1-score   support

           I       0.44      0.69      0.54        16
          II       0.75      0.38      0.50        16
         III       0.50      0.15      0.23        20
          IV       0.53      0.94      0.68        17

    accuracy                           0.52        69
   macro avg       0.56      0.54      0.49        69
```

```
weighted avg       0.55      0.52      0.48        69
```

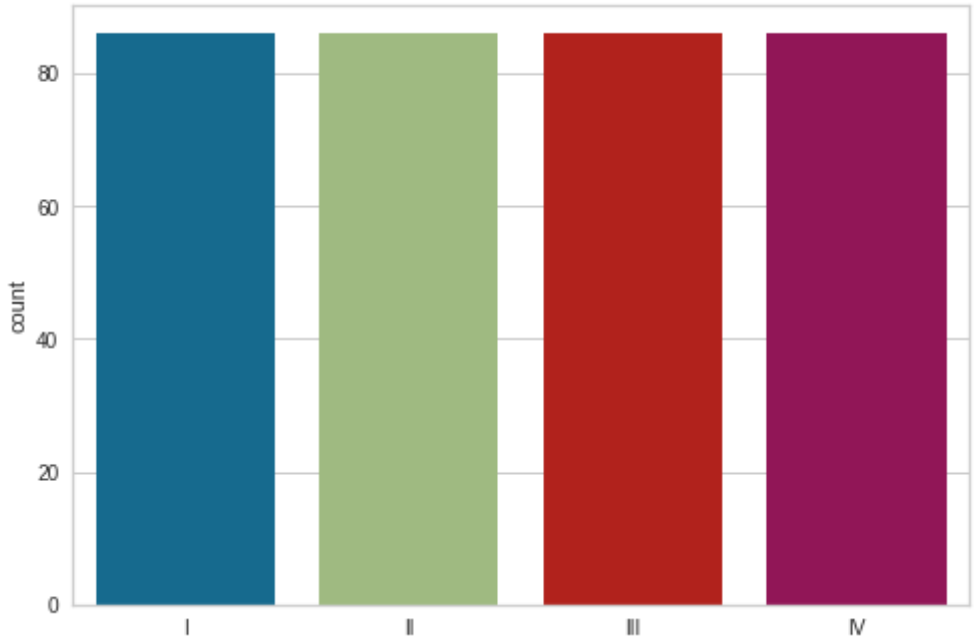
## C2-04. Agrupamento Hierarquico

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_

X_dados = dados.iloc[:,7:13].values

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

# label_encoder_sex = LabelEncoder()
# X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])

# Sobreamostragem com SMOTE

from imblearn.over_sampling import SMOTE

smote = SMOTE(sampling_strategy='not majority')
X_over, y_over = smote.fit_resample(X_dados, y_dados)

#y_over.shape, X_over.shape
np.unique(y_dados, return_counts = True), np.unique(y_over, return_counts = True)

import seaborn as sns
sns.countplot(x = y_over);

# from sklearn.preprocessing import OneHotEncoder
# from sklearn.compose import ColumnTransformer
# onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remain
# X_dados = onehotencorder.fit_transform(X_over).toarray()

#X_dados, X_dados.shape

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler()
X_dados = scaler_dados.fit_transform(X_over)
X_dados
```



```
#Separação dos dados BALANCEADOS
```

```
from sklearn.model_selection import train_test_split
X_dados_train_over, X_dados_test_over, y_dados_train_over, y_dados_test_over = train_test_s
```




```
#Reduzir as dimensões de dimensões.
from sklearn.decomposition import PCA
import plotly.express as px


pca = PCA(n_components=None)


X_dados_train_pca = pca.fit_transform(X_dados_train_over)
X_dados_test_pca = pca.transform(X_dados_test_over)


X_dados_train_pca.shape, X_dados_test_pca.shape,
```

```
    ((275, 6), (69, 6))
```

```
#X_dados_train_pca
```

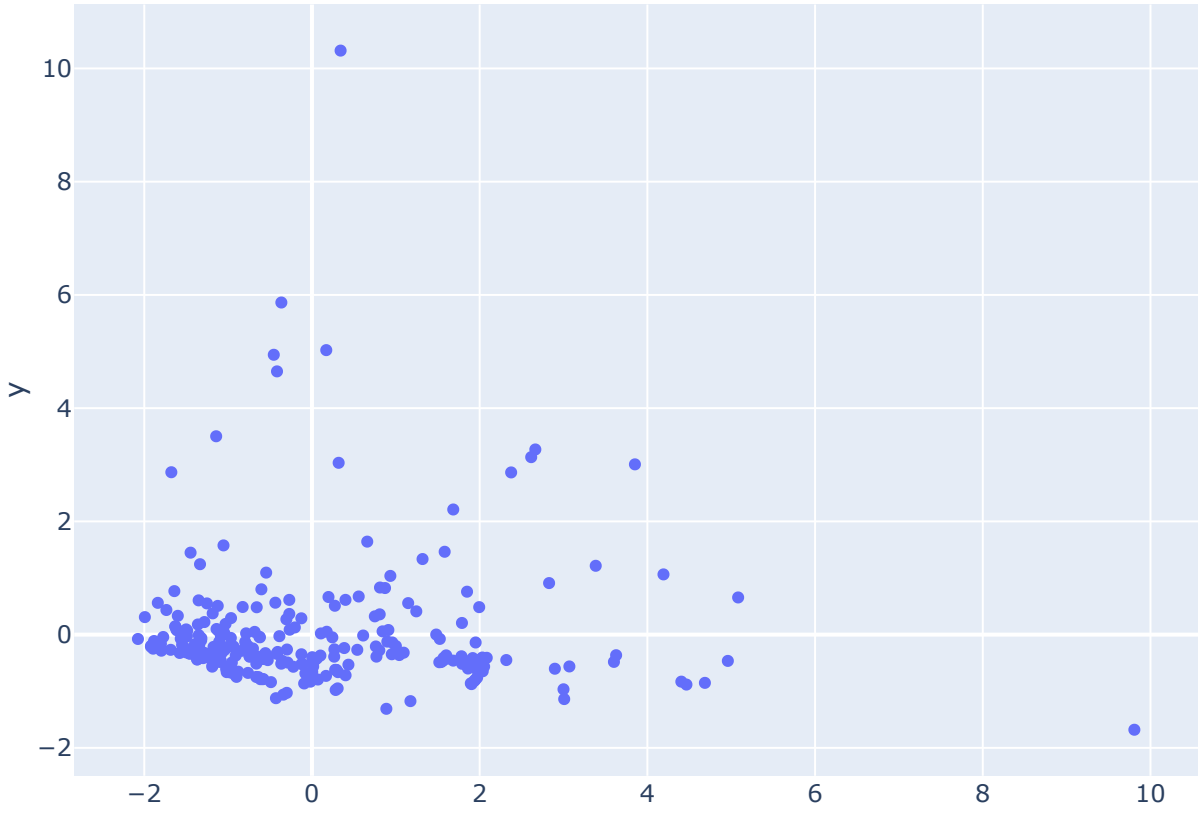
```
pca.explained_variance_ratio_
```

```
    array([0.39200402, 0.2054243 , 0.14428326, 0.12368825, 0.08509268,
           0.04950749])
```

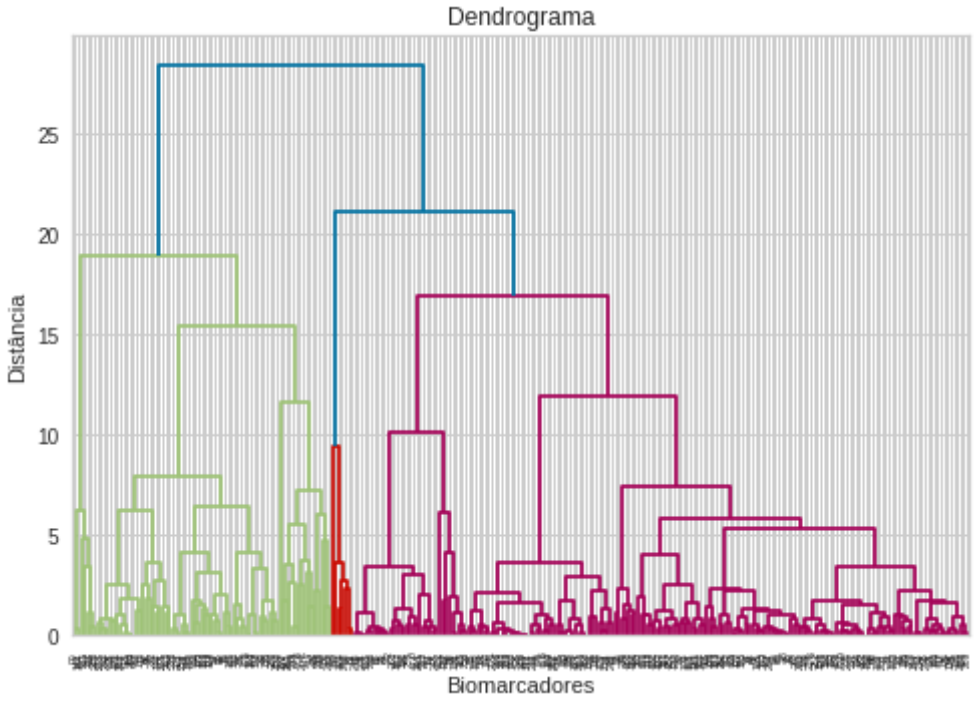
```
pca.explained_variance_ratio_.sum()
```

```
    1.0
```

```
grafico = px.scatter(x = X_dados_train_pca[:,0], y= X_dados_train_pca[:,1])
grafico.show()
```




```
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage


dendrograma = dendrogram(linkage(X_dados_train_pca, method='ward'))
plt.title('Dendrograma')
plt.xlabel('Biomarcadores')
plt.ylabel('Distância');
```




```
from sklearn.cluster import AgglomerativeClustering
```



```
hc_dados_agrup = AgglomerativeClustering(n_clusters=2, linkage='ward', affinity='euclidean'
```

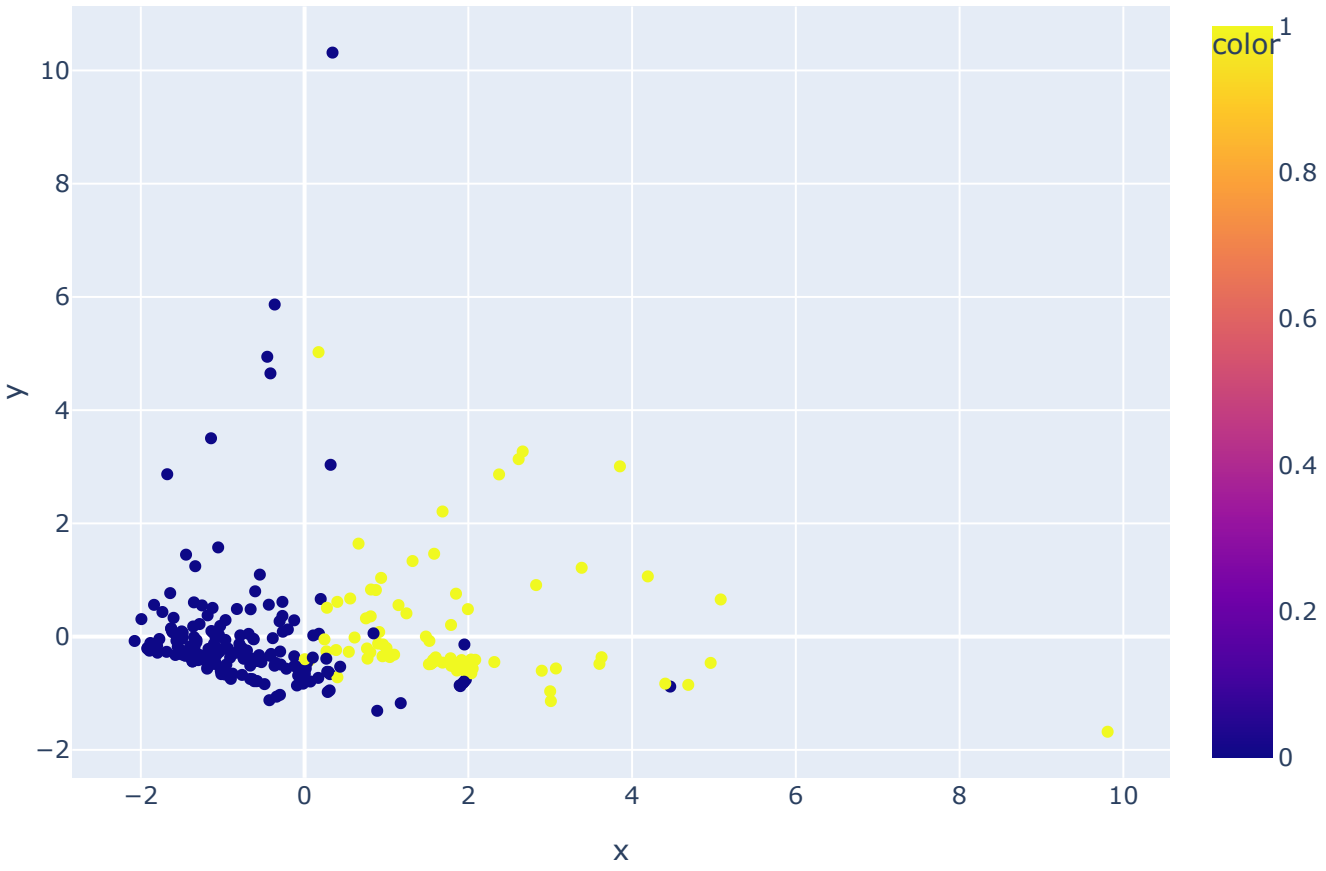
```
rotulos = hc_dados_agrup.fit_predict(X_dados_train_pca)
rotulos
```

```
array([0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0,
       1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1])
```

```
y_dados_train
```

```
array(['III', 'II', 'III', 'II', 'II', 'I', 'III', 'II', 'III', 'III',
       'III', 'III', 'II', 'III', 'III', 'II', 'III', 'III', 'II', 'III',
       'III', 'IV', 'II', 'IV', 'I', 'II', 'III', 'IV', 'II', 'II', 'II',
       'III', 'II', 'III', 'II', 'II', 'I', 'III', 'I', 'II', 'IV', 'III',
       'II', 'II', 'I', 'II', 'II', 'II', 'III', 'IV', 'II', 'III', 'IV',
       'III', 'I', 'III', 'III', 'II', 'II', 'II', 'II', 'II', 'II', 'II',
       'II', 'II', 'II', 'III', 'II', 'III', 'II', 'II', 'II', 'III',
       'III', 'II', 'III', 'II', 'IV', 'II', 'IV', 'II', 'III', 'IV',
       'IV', 'III', 'III', 'II', 'III', 'II', 'II', 'III', 'IV', 'III',
       'II', 'IV', 'I', 'III', 'I', 'II', 'II', 'III', 'II', 'II', 'II',
       'II', 'III', 'III', 'II', 'I', 'III', 'III', 'II', 'III', 'IV',
       'II', 'III', 'III', 'II', 'II', 'I', 'II', 'IV', 'III', 'III',
       'III', 'I', 'II', 'I', 'IV', 'III', 'III', 'II', 'III', 'III',
       'III', 'III', 'III', 'II', 'II', 'III', 'II', 'II', 'II', 'II',
       'IV', 'III', 'III', 'I', 'III', 'III', 'III', 'IV', 'I', 'II',
       'III', 'III', 'IV', 'II'], dtype=object)
```

```
grafico = px.scatter(x = X_dados_train_pca[:,0], y = X_dados_train_pca[:,1], color = rotulo
grafico.show()
```




✓ 0s conclusão: 14:07

# **Cenário 3:**{ Biomarcadores com Idade e Sexo | Dados Não balanceados}

## **Visão dos Dados -

```
#1. IMPORTAR O DATASET (COLUNAS IMPORTANTES: AGE, SEX, LYVE1, REG1B, TFF1 ==> ALVO PREVER A
#!pip install plotly --upgrade
#Baixar os dados e jogar na variável DADOS
import pandas as pd

dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")
dados = dados.fillna(1, inplace= False)
# 16 colunas
```

```
dados
```

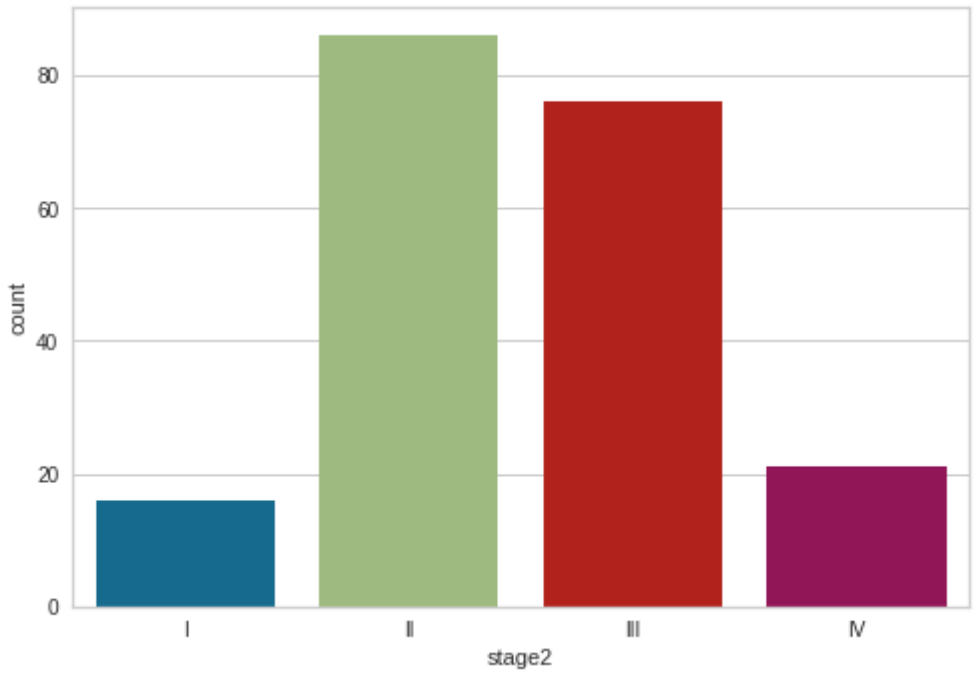
| | sample_id | patient_cohort | sample_origin | diagnosis | stage | sex | age | LYVE1 | |
|---|---|---|---|---|---|---|---|---|---|
| **0** | S497 | Cohort1 | ESP | 3 | I | F | 81 | 12.017150 | 431 |
| **1** | S456 | Cohort1 | LIV | 3 | IA | M | 57 | 2.628425 | 40 |
| **2** | S520 | Cohort1 | BPTB | 3 | IA | M | 55 | 2.830541 | 33 |
| **3** | S573 | Cohort2 | BPTB | 3 | IA | M | 58 | 0.632433 | 188 |
| **4** | S401 | Cohort1 | LIV | 3 | IB | M | 73 | 12.245820 | 196 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **194** | S549 | Cohort2 | BPTB | 3 | IV | M | 68 | 7.058209 | 156 |
| **195** | S558 | Cohort2 | BPTB | 3 | IV | F | 71 | 8.341207 | 16 |
| **196** | S560 | Cohort2 | BPTB | 3 | IV | M | 63 | 7.674707 | 289 |
| **197** | S583 | Cohort2 | BPTB | 3 | IV | F | 75 | 8.206777 | 205 |
| **198** | S590 | Cohort1 | BPTB | 3 | IV | M | 74 | 8.200958 | 411 |

199 rows × 14 columns

```
#Avaliar o número de amostras para cada classe(estadiamento)
import numpy as np
np.unique(dados['stage2'], return_counts = True)
```

```
(array(['I', 'II', 'III', 'IV'], dtype=object), array([16, 86, 76, 21]))
```

```
#Gerar um gráfico para avaliar visualmente o número de amostras para cada classe, objetivo
import seaborn as sns
sns.countplot(x = dados['stage2']);
```




## C3-01. KnnClassifier{*Unbalanced*} => 0.425

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,5:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

label_encoder_sex = LabelEncoder()
X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
X_dados

#Onehot
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remainde
X_dados = onehotencorder.fit_transform(X_dados)
```

```
X_dados

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler()
X_dados = scaler_dados.fit_transform(X_dados)


#Separação dos dados SEM o balanceamento
from sklearn.model_selection import train_test_split

X_dados_train , X_dados_test, y_dados_train, y_dados_test = train_test_split(X_dados, y_dad
X_dados_train.shape, X_dados_test.shape
```

```
((159, 9), (40, 9))
```

```
#KNN CLASSIFICADOR
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix,plot_confusion_matrix
from sklearn.preprocessing import StandardScaler


knn_modelc = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p = 3)
knn_modelc.fit(X_dados_train, y_dados_train)
```

```
KNeighborsClassifier(p=3)
```

```
previsoes = knn_modelc.predict(X_dados_test)
previsoes1 = previsoes
yprev1train = y_dados_test


y_dados_test
```

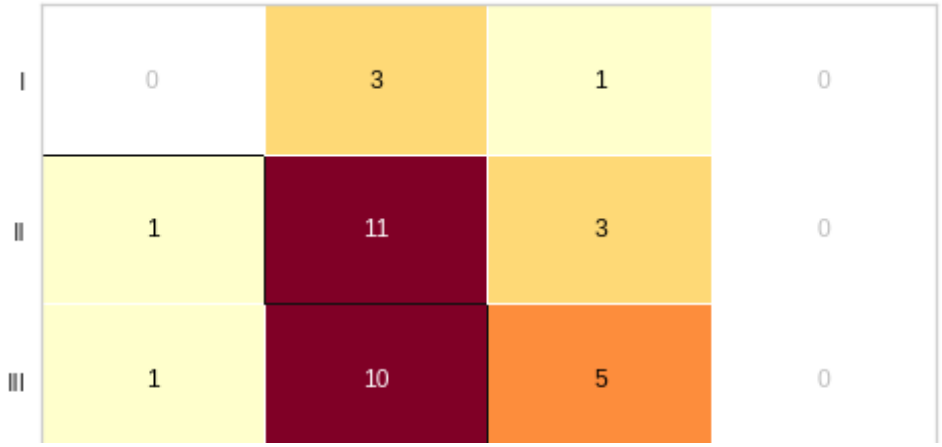
```
array(['II', 'III', 'III', 'II', 'III', 'IV', 'I', 'III', 'I', 'III',
       'II', 'III', 'III', 'III', 'II', 'I', 'II', 'III', 'II', 'II',
       'III', 'III', 'II', 'III', 'II', 'III', 'IV', 'IV', 'II', 'II',
       'II', 'II', 'II', 'III', 'II', 'IV', 'III', 'I', 'III', 'IV'],
      dtype=object)
```

```
from sklearn.metrics import accuracy_score, classification_report
accuracy_score(y_dados_test, previsoes) # padronização
```

```
0.425
```

```
from yellowbrick.classifier import ConfusionMatrix
cm = ConfusionMatrix(knn_modelc)
cm.fit(X_dados_train, y_dados_train)
cm.score(X_dados_test, y_dados_test)
```

0.425




```
print(classification_report(y_dados_test, previsoes))
```

```
              precision    recall  f1-score   support

           I       0.00      0.00      0.00         4
          II       0.42      0.73      0.54        15
         III       0.45      0.31      0.37        16
          IV       1.00      0.20      0.33         5

    accuracy                           0.42        40
   macro avg       0.47      0.31      0.31        40
weighted avg       0.47      0.42      0.39        40
```

## C3-02. RandomForestClassifier{*Unbalanced*} => 0.575

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,5:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

label_encoder_sex = LabelEncoder()
X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
X_dados

#Onehot
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
```

```
onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remainde
X_dados = onehotencorder.fit_transform(X_dados)
X_dados

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler()
X_dados = scaler_dados.fit_transform(X_dados)


#Separação dos dados SEM o balanceamento
from sklearn.model_selection import train_test_split

X_dados_train , X_dados_test, y_dados_train, y_dados_test = train_test_split(X_dados, y_dad
X_dados_train.shape, X_dados_test.shape
X_random_forest_dados_sb_teste = X_dados_test
y_random_forest_dados_sb_teste = y_dados_test


#RANDOM FOREST CLASSIFIER

from sklearn.ensemble import RandomForestClassifier
random_forest_dados_sb = RandomForestClassifier(criterion = 'entropy', min_samples_leaf =
random_forest_dados_sb.fit(X_dados_train, y_dados_train)


    RandomForestClassifier(criterion='entropy', min_samples_split=5,
                           n_estimators=200, random_state=0)


from sklearn.metrics import accuracy_score, classification_report
previsoes = random_forest_dados_sb.predict(X_dados_test)
accuracy_score(y_dados_test, previsoes)

    0.575


from yellowbrick.classifier import ConfusionMatrix
cm = ConfusionMatrix(random_forest_dados_sb)
cm.fit(X_dados_train, y_dados_train)
cm.score(X_dados_test, y_dados_test)
```

```
0.575
```

```
print(classification_report(y_dados_test, previsoes))
```

```
              precision    recall  f1-score   support

           I       0.50      0.25      0.33         4
          II       0.50      0.80      0.62        15
         III       0.67      0.50      0.57        16
          IV       1.00      0.40      0.57         5

    accuracy                           0.57        40
   macro avg       0.67      0.49      0.52        40
weighted avg       0.63      0.57      0.56        40
```

## C3-03. SVM{Unbalanced} => *0.575*, liner, 1, 1 | 0.575, sigmoid, 1, 1

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,5:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

label_encoder_sex = LabelEncoder()
X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
X_dados

#Onehot
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remainde
X_dados = onehotencorder.fit_transform(X_dados)
X_dados

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler() #(with_mean=False)
X_dados = scaler_dados.fit_transform(X_dados)
```

```
#Separação dos dados SEM o balanceamento
from sklearn.model_selection import train_test_split

X_dados_train , X_dados_test, y_dados_train, y_dados_test = train_test_split(X_dados, y_dad
X_dados_train.shape, X_dados_test.shape
```

```
((159, 9), (40, 9))
```

```
from sklearn.svm import SVC
```

```
svm_dados = SVC(kernel='linear', random_state=1, C = 1.0) # 2 -> 4
svm_dados.fit(X_dados_train, y_dados_train)
```

```
SVC(kernel='linear', random_state=1)
```

```
previsoes = svm_dados.predict(X_dados_test)
previsoes
```

```
array(['II', 'II', 'II', 'III', 'III', 'III', 'II', 'II', 'III', 'II',
       'III', 'II', 'II', 'II', 'III', 'II', 'II', 'II', 'II', 'II', 'II',
       'II', 'II', 'III', 'II', 'II', 'II', 'III', 'II', 'II', 'II', 'II',
       'II', 'II', 'II', 'II', 'II', 'III', 'III', 'II'], dtype=object)
```

```
y_dados_test
```

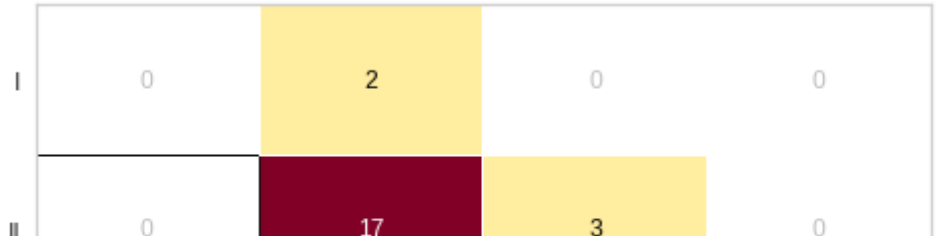
```
array(['III', 'II', 'III', 'III', 'III', 'III', 'III', 'I', 'II', 'II',
       'IV', 'II', 'II', 'II', 'II', 'III', 'I', 'II', 'III', 'III', 'IV',
       'II', 'II', 'III', 'II', 'II', 'III', 'II', 'II', 'II', 'II',
       'III', 'II', 'III', 'II', 'II', 'IV', 'III', 'III', 'II'],
      dtype=object)
```

```
from sklearn.metrics import accuracy_score, classification_report
#0.575, liner, 1, 1 | 0.425, poly, 1, 1 | #0.475, sigmoid, 1, 1 | 0.425, rbf, 1, 1
accuracy_score(y_dados_test, previsoes)
```

```
0.575
```

```
from yellowbrick.classifier import ConfusionMatrix
cm = ConfusionMatrix(svm_dados)
cm.fit(X_dados_train, y_dados_train)
cm.score(X_dados_test, y_dados_test)
```

```
0.575
```




```
print(classification_report(y_dados_test, previsoes))
```

```
              precision    recall  f1-score   support

           I       0.00      0.00      0.00         2
          II       0.57      0.85      0.68        20
         III       0.60      0.40      0.48        15
          IV       0.00      0.00      0.00         3

    accuracy                           0.57        40
   macro avg       0.29      0.31      0.29        40
weighted avg       0.51      0.57      0.52        40

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undefi

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undefi

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undefi

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted
```

---

# **Cenário 4:**{ Biomarcadores com Idade e Sexo | Dados Não balanceados}

---

## **02. KnnClassifier**{*Balanced*}0.608

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,5:13].values
X_dados
```



```
#Separando a classe alvo | stage2
```

```
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

label_encoder_sex = LabelEncoder()
X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
X_dados

# Sobreamostragem com SMOTE
from imblearn.over_sampling import SMOTE
import numpy as np

smote = SMOTE(sampling_strategy='not majority', random_state=3)
X_over, y_over = smote.fit_resample(X_dados, y_dados)

#y_over.shape, X_over.shape
np.unique(y_dados, return_counts = True), np.unique(y_over, return_counts = True)

import seaborn as sns
sns.countplot(x = y_over);

from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remainde
X_dados = onehotencorder.fit_transform(X_over).toarray()

#X_dados, X_dados.shape

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler()
X_dados = scaler_dados.fit_transform(X_dados)
X_dados

#Separação dos dados BALANCEADOS
from sklearn.model_selection import train_test_split

X_dados_train_over, X_dados_test_over, y_dados_train_over, y_dados_test_over = train_test_s
X_dados_train_over.shape, X_dados_test_over.shape
```

```
((275, 82), (69, 82))
```


```
#Knn Classifier Balanced

from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix,plot_confusion_matrix
from sklearn.preprocessing import StandardScaler
```


```
knn_modelcb = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p = 2)
knn_modelcb.fit(X_dados_train_over, y_dados_train_over)
```

```
KNeighborsClassifier()
```

```
previsoes = knn_modelcb.predict(X_dados_test_over)
previsoes
```

```
array(['I', 'II', 'II', 'I', 'I', 'IV', 'IV', 'I', 'I', 'I', 'I', 'II',
       'III', 'II', 'I', 'II', 'III', 'I', 'II', 'III', 'IV', 'II', 'I',
       'III', 'I', 'I', 'III', 'I', 'II', 'I', 'I', 'IV', 'III', 'III',
       'III', 'I', 'IV', 'IV', 'II', 'II', 'II', 'III', 'I', 'IV', 'IV',
       'I', 'IV', 'III', 'IV', 'I', 'II', 'II', 'II', 'IV', 'III', 'I',
       'III', 'III', 'I', 'I', 'II', 'II', 'I', 'IV', 'I', 'IV', 'II',
       'I', 'IV'], dtype=object)
```

```
y_dados_test_over
```

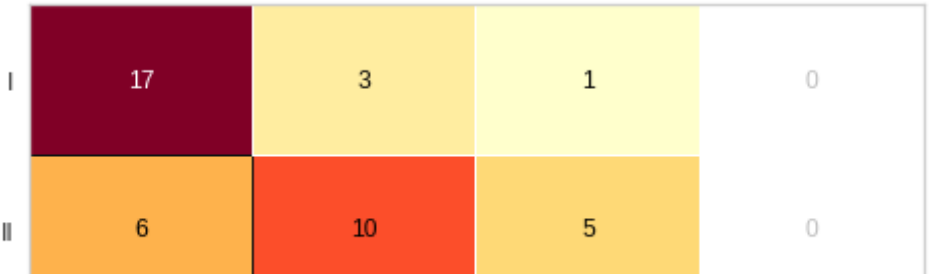
```
array(['III', 'I', 'II', 'I', 'II', 'IV', 'III', 'I', 'I', 'II', 'II',
       'I', 'III', 'IV', 'I', 'II', 'II', 'I', 'II', 'I', 'IV', 'II', 'I',
       'III', 'I', 'I', 'III', 'II', 'II', 'I', 'I', 'IV', 'II', 'II',
       'III', 'IV', 'IV', 'III', 'II', 'II', 'III', 'II', 'II', 'IV',
       'IV', 'I', 'IV', 'IV', 'III', 'I', 'II', 'II', 'III', 'III', 'III',
       'I', 'II', 'IV', 'I', 'I', 'I', 'IV', 'II', 'IV', 'I', 'IV', 'II',
       'I', 'IV'], dtype=object)
```

```
from sklearn.metrics import accuracy_score, classification_report
accuracy_score(y_dados_test_over, previsoes) # padronização
```

```
0.6086956521739131
```

```
from yellowbrick.classifier import ConfusionMatrix
cm = ConfusionMatrix(knn_modelcb)
cm.fit(X_dados_train_over, y_dados_train_over)
cm.score(X_dados_test_over, y_dados_test_over)
```

```
0.6086956521739131
```




```
print(classification_report(y_dados_test_over, previsoes))
```

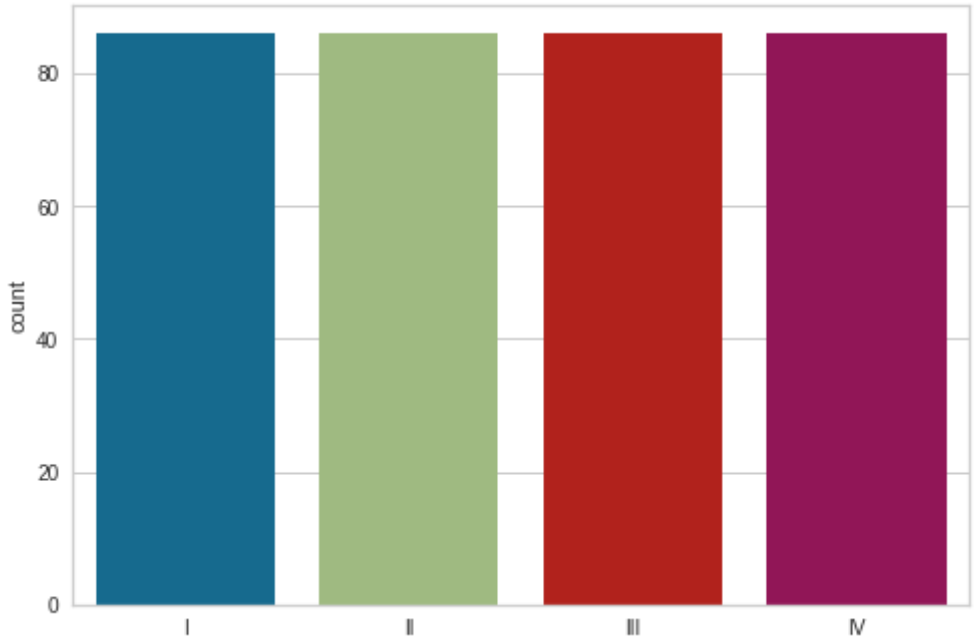
```
              precision    recall  f1-score   support

           I       0.68      0.81      0.74        21
          II       0.59      0.48      0.53        21
         III       0.38      0.42      0.40        12
          IV       0.71      0.67      0.69        15

    accuracy                           0.61        69
   macro avg       0.59      0.59      0.59        69
weighted avg       0.61      0.61      0.60        69
```

## *04. RandomForestClassifier*{Balanced} => 0.81

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,5:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

label_encoder_sex = LabelEncoder()
X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
X_dados

# Sobreamostragem com SMOTE

from imblearn.over_sampling import SMOTE
smote = SMOTE(sampling_strategy='not majority', random_state=29)
#9  0.7971014492753623

X_over, y_over = smote.fit_resample(X_dados, y_dados)
```

```
#y_over.shape, X_over.shape
np.unique(y_dados, return_counts = True), np.unique(y_over, return_counts = True)

import seaborn as sns
sns.countplot(x = y_over);

from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remainde
X_dados = onehotencorder.fit_transform(X_over).toarray()

#X_dados, X_dados.shape

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler()
X_dados = scaler_dados.fit_transform(X_dados)
X_dados

#Separação dos dados BALANCEADOS
from sklearn.model_selection import train_test_split

X_dados_train_over, X_dados_test_over, y_dados_train_over, y_dados_test_over = train_test_s
X_dados_train_over.shape, X_dados_test_over.shape
```

```
((275, 92), (69, 92))
```




```
from sklearn.ensemble import RandomForestClassifier #1,6,300,10 -->0.681 | 1,5,200, 17 -->0

random_forest_dados = RandomForestClassifier(criterion = 'entropy', min_samples_leaf =  1,
random_forest_dados.fit(X_dados_train_over, y_dados_train_over)
```

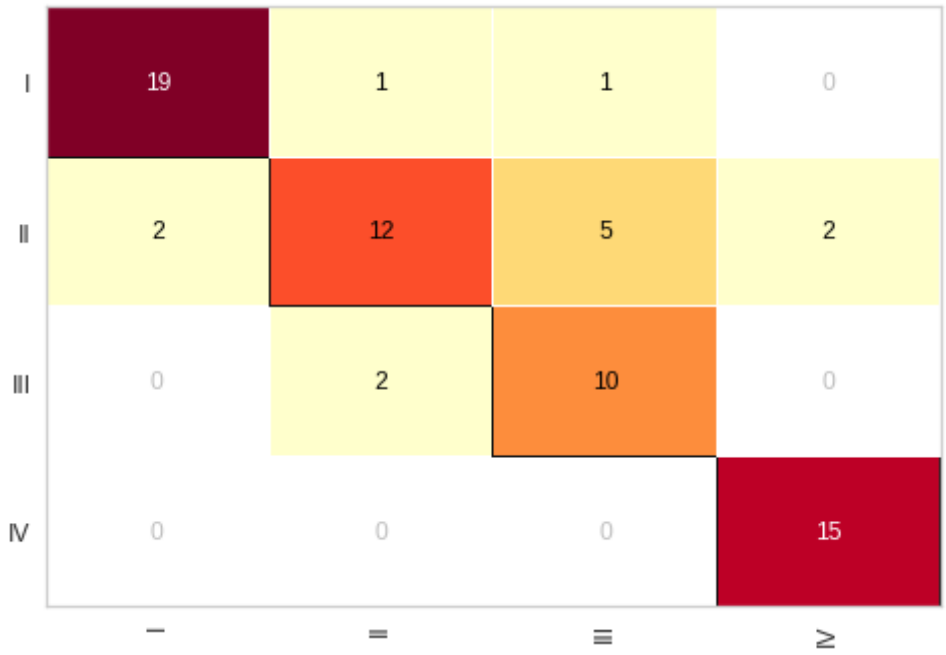
```
RandomForestClassifier(criterion='entropy', min_samples_split=5,
                       n_estimators=200, random_state=17)
```

```
from sklearn.metrics import accuracy_score, classification_report
previsoes = random_forest_dados.predict(X_dados_test_over)
accuracy_score(y_dados_test_over, previsoes)
```

```
0.8115942028985508
```

```
from yellowbrick.classifier import ConfusionMatrix
cm = ConfusionMatrix(random_forest_dados)
cm.fit(X_dados_train_over, y_dados_train_over)
cm.score(X_dados_test_over, y_dados_test_over)
```

```
0.8115942028985508
```




```
print(classification_report(y_dados_test_over, previsoes))
```

```
                precision    recall  f1-score   support

             I       0.90      0.90      0.90        21
            II       0.80      0.57      0.67        21
           III       0.62      0.83      0.71        12
            IV       0.88      1.00      0.94        15

      accuracy                           0.81        69
     macro avg       0.80      0.83      0.81        69
  weighted avg       0.82      0.81      0.81        69
```

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,5:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados
```

```
# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

label_encoder_sex = LabelEncoder()
X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
X_dados

#Onehot
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remainde
X_dados = onehotencorder.fit_transform(X_dados)
X_dados

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler()
X_dados = scaler_dados.fit_transform(X_dados)


#Separação dos dados SEM o balanceamento
from sklearn.model_selection import train_test_split

X_dados_train , X_dados_test, y_dados_train, y_dados_test = train_test_split(X_dados, y_dad
X_dados_train.shape, X_dados_test.shape
```

```
((159, 9), (40, 9))
```

```
from sklearn.svm import SVC



svm_dados = SVC(kernel='linear', random_state=1, C = 1.0) # 2 -> 4
svm_dados.fit(X_dados_train, y_dados_train)
```

```
SVC(kernel='linear', random_state=1)
```

```
previsoes = svm_dados.predict(X_dados_test)
previsoes
```

```
array(['II', 'II', 'II', 'III', 'III', 'III', 'II', 'II', 'III', 'II',
       'III', 'II', 'II', 'II', 'III', 'II', 'II', 'II', 'II', 'II', 'II',
       'II', 'II', 'III', 'II', 'II', 'II', 'III', 'II', 'II', 'II', 'II',
       'II', 'II', 'II', 'II', 'II', 'III', 'III', 'II'], dtype=object)
```

```
y_dados_test
```

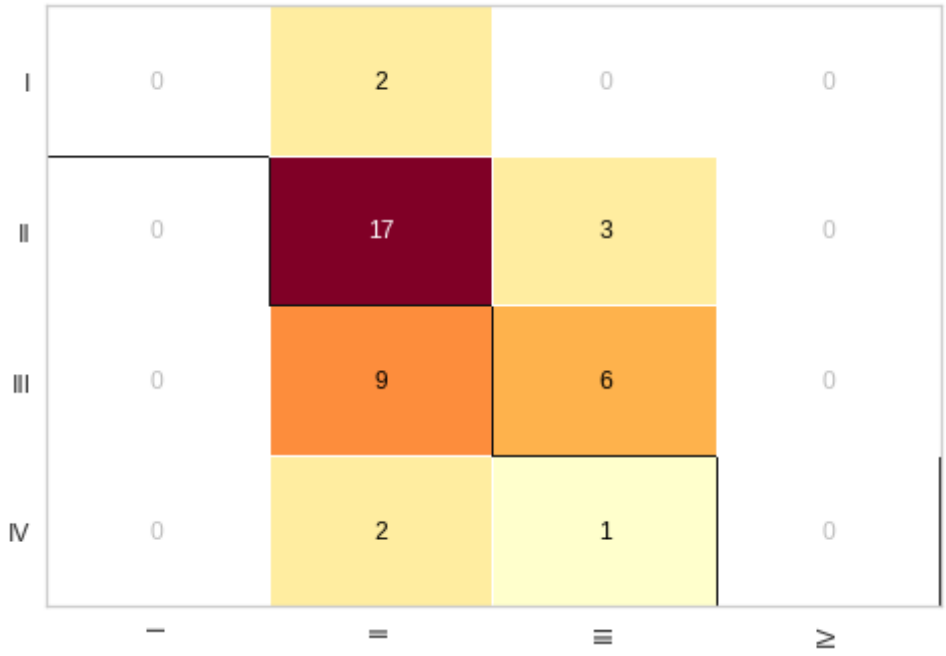
```
array(['III', 'II', 'III', 'III', 'III', 'III', 'III', 'I', 'II', 'II',
       'IV', 'II', 'II', 'II', 'II', 'III', 'I', 'II', 'III', 'III', 'IV',
       'II', 'II', 'III', 'II', 'II', 'III', 'II', 'II', 'II', 'II',
       'III', 'II', 'III', 'II', 'II', 'IV', 'III', 'III', 'II'],
      dtype=object)
```

```
from sklearn.metrics import accuracy_score, classification_report
#0.575, liner, 1, 1 | 0.425, poly, 1, 1 | #0.475, sigmoid, 1, 1 | 0.425, rbf, 1, 1
accuracy_score(y_dados_test, previsoes)
```

```
0.575
```

```
from yellowbrick.classifier import ConfusionMatrix
cm = ConfusionMatrix(svm_dados)
cm.fit(X_dados_train, y_dados_train)
cm.score(X_dados_test, y_dados_test)
```

```
0.575
```




```
print(classification_report(y_dados_test, previsoes))
```

```
                precision    recall  f1-score   support

             I       0.00      0.00      0.00         2
            II       0.57      0.85      0.68        20
           III       0.60      0.40      0.48        15
            IV       0.00      0.00      0.00         3

      accuracy                           0.57        40
     macro avg       0.29      0.31      0.29        40
  weighted avg       0.51      0.57      0.52        40

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undefi

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undefi

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undefi

Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted
```

## 06. SVM{Balanced} => 0.579, linear, 0, 10 |

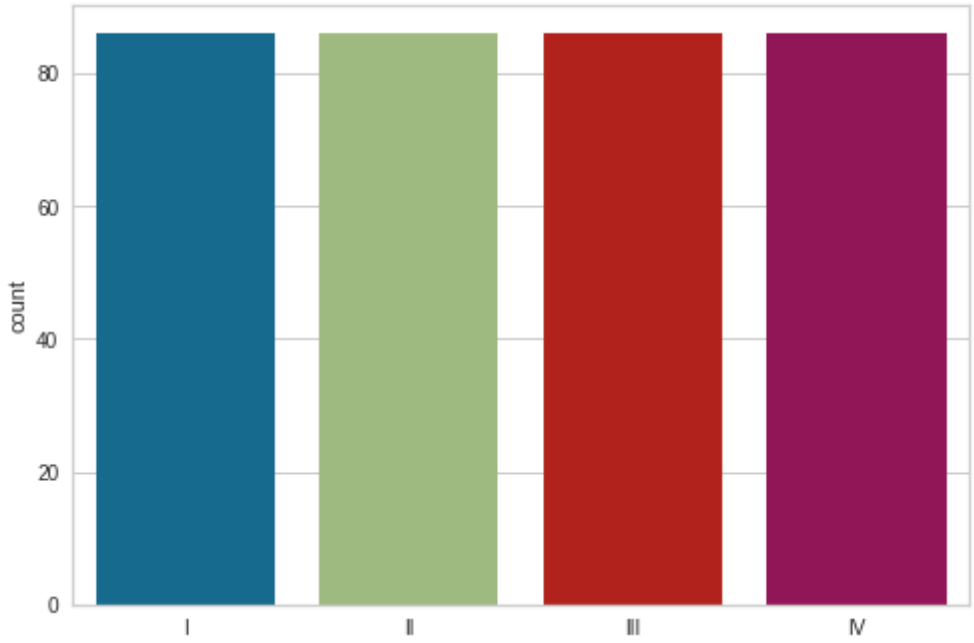


```
#PREPARAR OS DADOS ==> Importar e tratar os dados
```

```
import pandas as pd


# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")

#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,5:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

label_encoder_sex = LabelEncoder()
X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])
X_dados

# Sobreamostragem com SMOTE

from imblearn.over_sampling import SMOTE
import numpy as np

lista = []
count = 1

smote = SMOTE(sampling_strategy='not majority', random_state=16)
#2 - 47
X_over, y_over = smote.fit_resample(X_dados, y_dados)

#y_over.shape, X_over.shape

np.unique(y_dados, return_counts = True), np.unique(y_over, return_counts = True)

import seaborn as sns
sns.countplot(x = y_over);

from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remainde
X_dados = onehotencorder.fit_transform(X_over).toarray()

#X_dados, X_dados.shape

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler()
X_dados = scaler_dados.fit_transform(X_dados)
X_dados

#Separação dos dados BALANCEADOS
from sklearn.model_selection import train_test_split
```

```
X_dados_train_over, X_dados_test_over, y_dados_train_over, y_dados_test_over = train_test_s
X_dados_train_over.shape, X_dados_test_over.shape
np.unique(y_dados, return_counts = True), np.unique(y_over, return_counts = True)
```

```
((array(['I', 'II', 'III', 'IV'], dtype=object), array([16, 86, 76, 21])),
 (array(['I', 'II', 'III', 'IV'], dtype=object), array([86, 86, 86, 86])))
```




```
from sklearn.svm import SVC
```

```
svm_dados = SVC(kernel='linear', random_state=0, C = 10.0) # 2 -> 4
svm_dados.fit(X_dados_train_over, y_dados_train_over)
```

```
SVC(C=10.0, kernel='linear', random_state=0)
```

```
previsoes = svm_dados.predict(X_dados_test_over)
previsoes
```

```
array(['IV', 'IV', 'IV', 'III', 'II', 'I', 'I', 'II', 'IV', 'I', 'I',
       'IV', 'IV', 'I', 'II', 'II', 'III', 'III', 'II', 'II', 'I', 'I',
       'IV', 'IV', 'III', 'I', 'IV', 'II', 'IV', 'II', 'I', 'II', 'I',
       'III', 'IV', 'III', 'III', 'IV', 'III', 'II', 'II', 'IV', 'II',
       'I', 'III', 'III', 'II', 'IV', 'IV', 'III', 'III', 'III', 'I',
       'II', 'III', 'I', 'IV', 'IV', 'II', 'I', 'I', 'II', 'IV', 'IV',
       'II', 'I', 'II', 'I', 'II'], dtype=object)
```

Clique duas vezes (ou pressione "Enter") para editar

```
y_dados_test_over
```

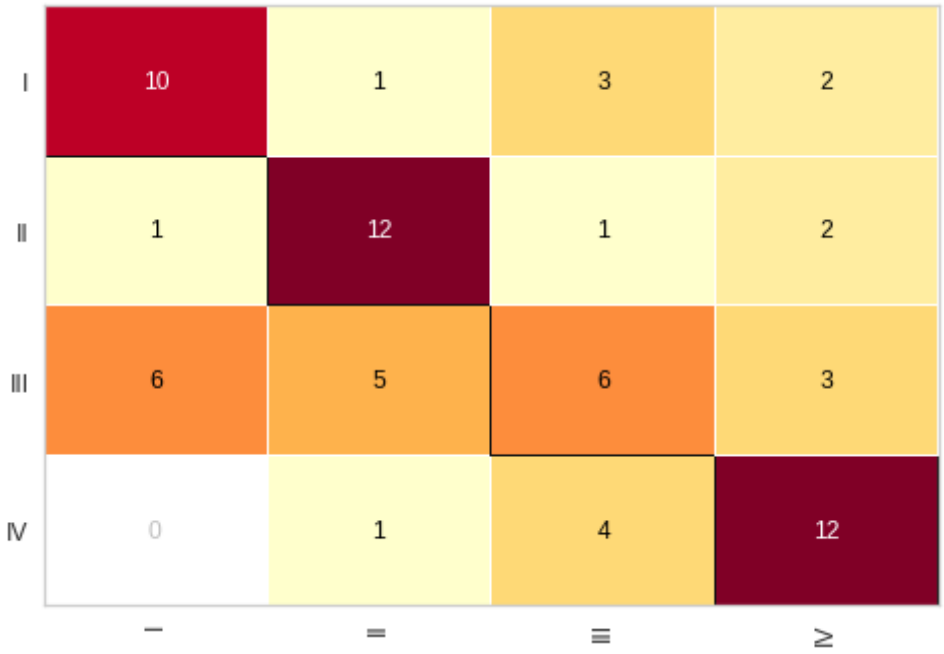
```
array(['IV', 'IV', 'IV', 'III', 'IV', 'I', 'III', 'I', 'III', 'I', 'III',
       'I', 'IV', 'III', 'II', 'II', 'I', 'I', 'II', 'II', 'I', 'I', 'IV',
       'IV', 'II', 'I', 'I', 'III', 'IV', 'II', 'III', 'III', 'I', 'III',
       'III', 'III', 'III', 'II', 'IV', 'II', 'II', 'III', 'III', 'I',
       'IV', 'I', 'III', 'IV', 'IV', 'III', 'III', 'IV', 'III', 'III',
       'IV', 'I', 'IV', 'II', 'II', 'I', 'III', 'II', 'IV', 'IV', 'II',
       'II', 'II', 'I', 'II'], dtype=object)
```

```
from sklearn.metrics import accuracy_score, classification_report
#0.475, liner, 1, 1 | 0.425, poly, 1, 1 | #0.475, sigmoid, 1, 1 | 0.425, rbf, 1, 1
#0.434, rbf 1, 2 |0.521 linear, 0, 10

accuracy_score(y_dados_test_over, previsoes)
```

```
0.5797101449275363
```

```
from yellowbrick.classifier import ConfusionMatrix
cm = ConfusionMatrix(svm_dados)
cm.fit(X_dados_train_over, y_dados_train_over)
cm.score(X_dados_test_over, y_dados_test_over)
```

```
0.5797101449275363
```




```
print(classification_report(y_dados_test_over, previsoes))
```

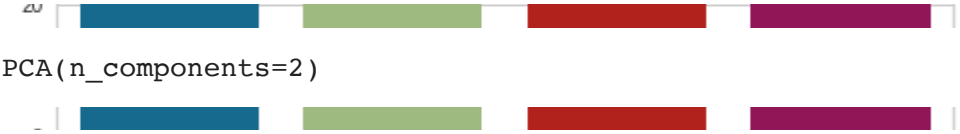
```
                precision    recall  f1-score   support

             I       0.59      0.62      0.61        16
            II       0.63      0.75      0.69        16
           III       0.43      0.30      0.35        20
            IV       0.63      0.71      0.67        17

      accuracy                           0.58        69
     macro avg       0.57      0.60      0.58        69
  weighted avg       0.56      0.58      0.57        69
```

# 07. Agrupamento Hierarquico

```
#PREPARAR OS DADOS ==> Importar e tratar os dados
import pandas as pd

# Importando o Arquivo
dados = pd.read_csv("dataMODIFICADOv10F.csv", sep=",")
```

```
#Transformando os nulos para 0
dados = dados.fillna(1, inplace= False)

#Separando as variaveis de interesse | sex  age LYVE1   REG1B   TFF1    creatinine  plasma_
X_dados = dados.iloc[:,5:13].values
X_dados

#Separando a classe alvo | stage2
y_dados = dados.iloc[:,13].values
y_dados

# Tratando as colunas que São rotulos | sex
from sklearn.preprocessing import LabelEncoder

label_encoder_sex = LabelEncoder()
X_dados[:,0] = label_encoder_sex.fit_transform(X_dados[:,0])

# Sobreamostragem com SMOTE

from imblearn.over_sampling import SMOTE

smote = SMOTE(sampling_strategy='not majority')
X_over, y_over = smote.fit_resample(X_dados, y_dados)

#y_over.shape, X_over.shape
np.unique(y_dados, return_counts = True), np.unique(y_over, return_counts = True)

import seaborn as sns
sns.countplot(x = y_over);

from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
onehotencorder = ColumnTransformer(transformers=[("OneHot", OneHotEncoder(), [0])],remainde
X_dados = onehotencorder.fit_transform(X_over).toarray()

#X_dados, X_dados.shape

#ESCALONAMENTO DE ATRIBUTOS
from sklearn.preprocessing import StandardScaler
scaler_dados = StandardScaler()
X_dados = scaler_dados.fit_transform(X_over)
X_dados

#Separação dos dados BALANCEADOS
from sklearn.model_selection import train_test_split
X_dados_train_over, X_dados_test_over, y_dados_train_over, y_dados_test_over = train_test_s
```


```
#Reduzir as dimensões de dimensões.
from sklearn.decomposition import PCA
import plotly.express as px
```



```
pca = PCA(n_components=2)
```



```
X_dados_train_pca = pca.fit_transform(X_dados_train_over)
X_dados_test_pca = pca.transform(X_dados_test_over)


X_dados_train_pca.shape, X_dados_test_pca.shape,
```

```
((275, 2), (69, 2))
```

```
#X_dados_train_pca
```

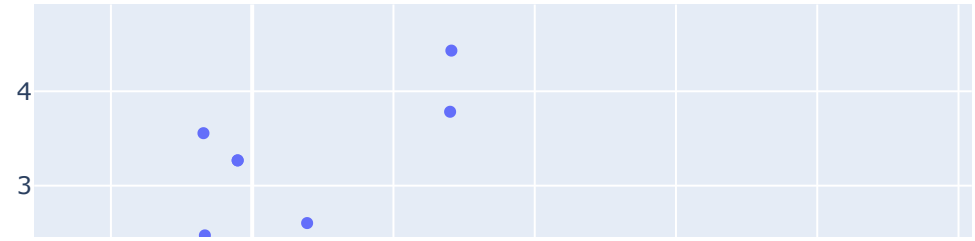
```
pca.explained_variance_ratio_
```

```
array([0.31860026, 0.15550037])
```

```
pca.explained_variance_ratio_.sum()
```

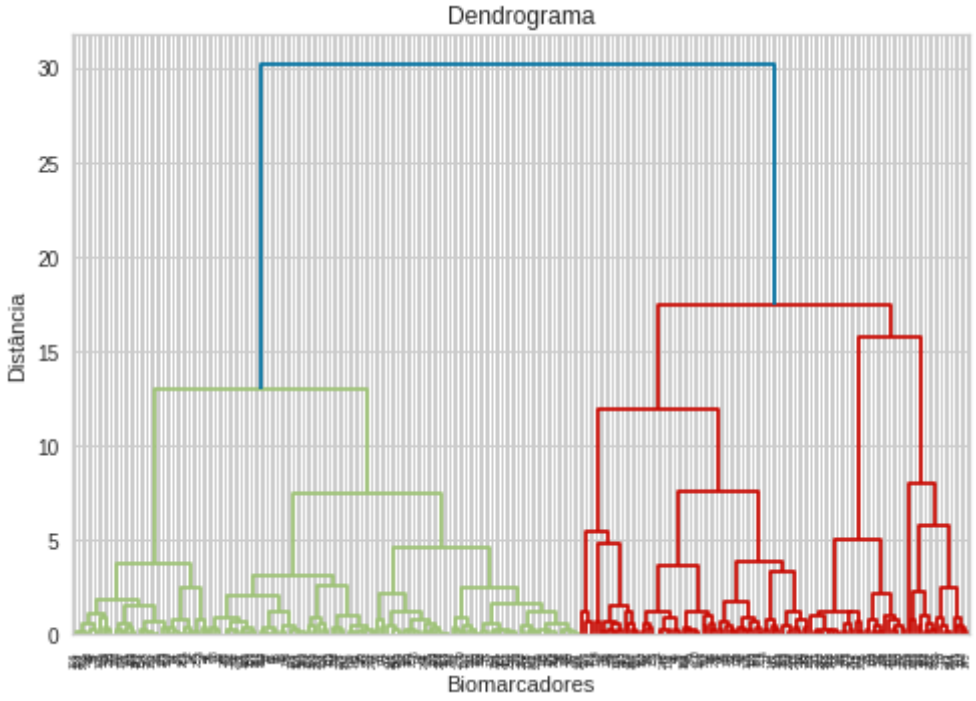
```
0.47410062534549513
```

```
grafico = px.scatter(x = X_dados_train_pca[:,0], y= X_dados_train_pca[:,1])
grafico.show()
```

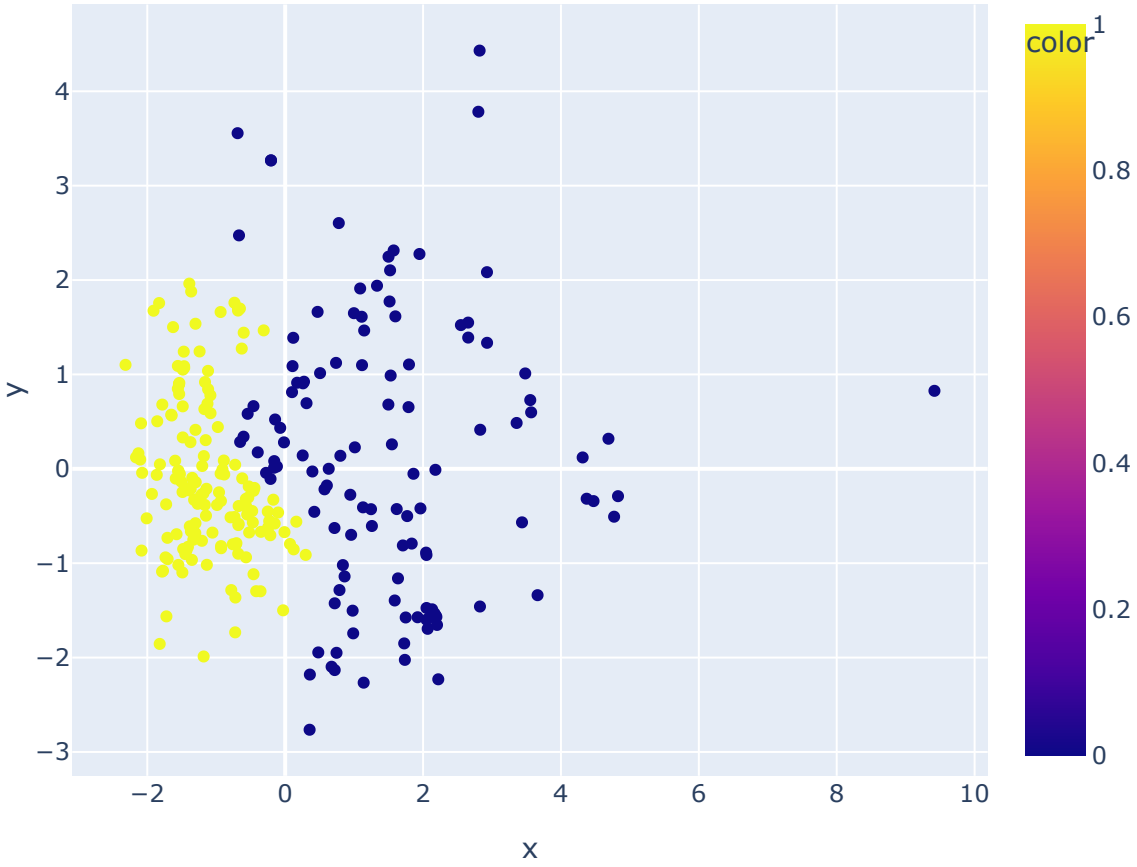



```
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage
```

```
dendrograma = dendrogram(linkage(X_dados_train_pca, method='ward'))
plt.title('Dendrograma')
plt.xlabel('Biomarcadores')
plt.ylabel('Distância');
```




```
from sklearn.cluster import AgglomerativeClustering


hc_dados_agrup = AgglomerativeClustering(n_clusters=2, linkage='ward', affinity='euclidean'
rotulos = hc_dados_agrup.fit_predict(X_dados_train_pca)


grafico = px.scatter(x = X_dados_train_pca[:,0], y = X_dados_train_pca[:,1], color = rotulo
grafico.show()
```




✓ 0s conclusão: 14:12