CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS TIMÓTEO

Silas Eduardo de Souza

APLICAÇÃO DE TÉCNICAS DE MACHINE LEARNING EM UM DATASET DE DOADORES DE SANGUE

Timóteo

2022

Silas Eduardo de Souza

APLICAÇÃO DE TÉCNICAS DE MACHINE LEARNING EM UM DATASET DE DOADORES DE SANGUE

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Maurílio Alves Martins da Costa
Coorientador: Prof. Me. Marcelo de Souza Balbino

Timóteo

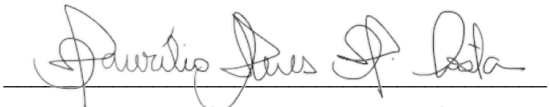
2022

Silas Eduardo de Souza

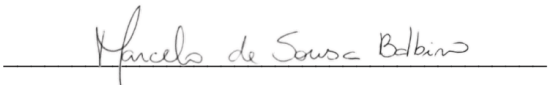
APLICAÇÃO DE TÉCNICAS DE MACHINE LEARNING EM UM DATASET DE DOADORES DE SANGUE

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo, como requisito parcial para obtenção do título de Engenheiro de Computação.

Trabalho aprovado. Timóteo, 21 de julho de 2022:



Prof. Dr. Maurílio Alves Martins da Costa
Orientador



Prof. Me. Marcelo de Sousa Balbino
Professor Co-orientador

Prof. Me. Aléssio Miranda Júnior
Professor Convidado

Timóteo
2022

Dedico aos meus pais, Silas e Ilcimar, por sempre me apoiarem nos estudos e pelas palavras de conforto e coragem nos momentos de dificuldades.

Agradecimentos

Agradeço ao meu orientador Orientador Dr. Maurílio Alves Martins da Costa pela orientação no desenvolvimento deste trabalho, agradeço também ao meu Coorientador Me. Marcelo de Souza Balbino pela ajuda na parte técnica do trabalho e também pelos conhecimentos compartilhados.

Agradeço a minha família por me apoiarem nos momentos mais difíceis seja com palavras de conforto ou de incentivo, sem vocês eu não teria conseguido.

“Por meio da tecnologia de inteligência artificial, podemos ter um conhecimento além da superficialidade, um conhecimento que revoluciona o mundo”.

Thauane Mendes

Resumo

O uso do aprendizado de máquina na área da saúde começou a pouco tempo mas está em bastante acensão. Seu uso está mais concentrado em área administrativas e financeiras mas já está entrando para dentro de consultórios, laboratórios e centros cirúrgicos. Hoje o aprendizado de máquina auxilia no diagnóstico de doenças e até em cirurgias executadas por robôs automatizados. O baixo estoque de sangue nos centros de coleta é um grave problema que o sistema de saúde brasileiro atravessa, a ausência de sangue em hospitais e centros de coleta pode definir se uma paciente sobreviva ou não em casos de acidentes graves, e um dos motivos é a fata de informação e de comunicação entre os centros de coletas e doadores. E com a pandemia da Covid-19 o número total de doadores diminuiu por causa do medo das pessoas de contraírem a doença em ambientes hospitalares. O incentivo e campanhas podem auxiliar no aumento do número de doadores. Com todo esse cenário o estudo e criação de um modelo de aprendizagem de máquina para predição e classificação de doadores junto com outras ferramentas podem auxiliar hospitais e centro de coletas a encontrar estes doadores. Para criar estes modelos foram usados e testados diversos algoritmos e ferramentas além da validação dos modelos por meios de métricas como avaliação da Acurácia, Precisão, Revocação, F1 Score e Matriz de Confusão. O modelo Decision Jungle apresentou o melhores números em relação as métricas usando a ferramenta Azure Machine Learning Studio. Usando o ecossistema Python o modelo Decision Tree foi o que obteve o melhor desempenho em relação aos outros algoritmos testados. Este trabalho demonstrou que é possível a utilização de modelo de Machine Learning para classificação e predição de doadores de sangue.

Palavras-chave: Aprendizado de máquina, Doação de sangue, Árvores de decisão.

Abstract

The use of machine learning in healthcare has just started but is on the rise. Its use is more concentrated in administrative and financial areas, but is already entering offices, laboratories, and surgical centers. The low stock of blood in the collection centers is a serious problem that the Brazilian health system goes through, the absence of blood in hospitals and collection centers can define whether a patient survives or not in cases of serious accidents, and one of the reasons is the lack of information and communication between the collection centers and donors. And with the Covid-19 pandemic the total number of donors has decreased because of people's fear of contracting the disease in hospital environments. Incentives and campaigns can help increase the number of donors. With all this scenario the study and creation of a machine learning model for donor prediction and classification along with other tools can help hospitals and collection centers to find these donors. To create these models, several algorithms and tools were used and tested, besides the validation of the models through metrics such as accuracy, precision, recall, F1 Score and confusion matrix. The Decision Jungle model showed the best performance using the Azure Machine Learning Studio tool, reaching an accuracy of 85%. Using the Python ecosystem the Decision Tree model was the one that obtained the best performance also reaching a percentage around 85%. This work demonstrated that it is possible to use Machine Learning models for classification and prediction of blood donors.

Keywords: Machine learning, Blood donation, Decision trees.

Lista de ilustrações

Figura 1 – Percentual de registros de consulta em relação às faixas de idades dos pacientes.. 20
Figura 2 – Centro de Usinagem CNC Feeler VMP-30A. 22
Figura 3 – Orientação dos eixos de medição do aplicativo Sci Journal e do smartphone em relação a direção do processo de usinagem. 23
Figura 4 – Comparação dos valores de rugosidade obtidos nos testes do algoritmo e dos valores medidos. 23
Figura 5 – Etapas do trabalho. 26
Figura 6 – Etapas para criação de modelo usando Azure ML Studio (Classic) 28
Figura 7 – Exemplo de Árvore de decisão 32
Figura 8 – Gráfico de regressão linear 34
Figura 9 – Representação do serviço de Computação em Nuvem 36
Figura 10 – Tela Inicial da criação do experimento no Azure ML Studio 40
Figura 11 – Visualização do Dataset no Azure ML Studio 41
Figura 12 – Tabela de cálculo do resultado da avaliação de algoritmo. 43
Figura 13 – Gráfico de resultados de avaliação do modelo de ML. 44
Figura 14 – Pipeline do projeto 45
Figura 15 – Primeiros cinco do Dataset 46
Figura 16 – Alteração do nome das colunas 46
Figura 17 – Tabela de dados estatísticos do Dataset 47
Figura 18 – Tabela de informações do Dataset 48
Figura 19 – Matriz de correlação 49
Figura 20 – Tela de Resultados do Two-Class Boosted Decision Tree no Azure Studio. . . 54
Figura 21 – Gráficos das métricas do algorítimo Two-class Boosted Decision Tree 55
Figura 22 – Tela de Resultados do Two-Class Bayes Point Machine no Azure Studio. . . 56
Figura 23 – Gráficos das métricas do algorítimo Two-class Bayes Point Machine 56
Figura 24 – Tela de Resultados do Two-Class Neural Network no Azure Studio. 57
Figura 25 – Gráficos das métricas do algorítimo Two-class Neural Network 58
Figura 26 – Tela de Resultados do Locally DSVM no Azure Studio. 59
Figura 27 – Gráficos das métricas do algorítimo Two-class Locally DSVM 60
Figura 28 – Tela de Resultados do Two-class Decision Jungle no Azure Studio. 61
Figura 29 – Gráficos das métricas do algorítimo Two-class Decision Jungle 62
Figura 30 – Tela de Resultados do Two-class Decision Forest no Azure Studio. 63
Figura 31 – Gráficos das métricas do algoritmo Two-Class Decision Forest 63
Figura 32 – Tela de Resultados do Two-class Averaged Perceptron no Azure Studio. . . . 64
Figura 33 – Gráficos das métricas do algoritmo Two-class Averaged Perceptron 65
Figura 34 – Tela de Resultados do Two-class Logistic Regression Perceptron no Azure Studio. 66
Figura 35 – Gráficos das métricas do algoritmo Two-class Logistic Regression 66

Figura 36 – Tela de Resultados do Two-class Support Vector Machine no Azure Studio ML 67
Figura 37 – Gráficos das métricas do algoritmo Two-Class Support Vector Machine . . . 68
Figura 38 – Tela Run do web Service do Azure ML Studio 70
Figura 39 – Formulário web do Azure ML Studio 71
Figura 40 – Resultado do teste feito na interface web do Azure 72
Figura 41 – Resultado do teste feito no Workbook do Excel 72
Figura 42 – Métricas Logistic Regression 73
Figura 43 – Métricas Classificador SGD 74
Figura 44 – Métricas KNN 75
Figura 45 – Métricas Algorítimo Navie Bayes 76
Figura 46 – Árvore sem poda 77
Figura 47 – Árvore com poda 78
Figura 48 – Métricas Algorítimo Decision Tree 78
Figura 49 – Métricas Algorítimo Random Forest 79
Figura 50 – Métricas Algorítimo Extra Tree 80
Figura 51 – API flask 82
Figura 52 – Método POST usando Software Insomnia 83

Lista de tabelas

Tabela 1 – Parâmetros de usinagem utilizados no experimento. 22
Tabela 2 – Testes do experimento. 24
Tabela 3 – Precisão dos resultados Previstos pelo modelo Decision Tree Regresion . . 24
Tabela 4 – Amostras dos Atributos e valores do Dataset 39
Tabela 5 – Parâmetros definidos para o Two-Class Boosted Decision Tree 51
Tabela 6 – Parâmetros definidos para o Two-Class Neural Network 51
Tabela 7 – Parâmetros definidos para Two-Class Bayes Point Machine 52
Tabela 8 – Parâmetros definidos para o Two-class locally deep support vector machine 52
Tabela 9 – Parâmetros definidos para Two-Class Decision Forest 52
Tabela 10 – Parâmetros definidos para Two-Class Averaged Perceptron 53
Tabela 11 – Parâmetros definidos para Two-class Logistic Regression 53
Tabela 12 – Parâmetros definidos para Two-Class Support Vector Machine 53
Tabela 13 – Parâmetros definidos para Two-Class Decision Jungle 54
Tabela 14 – Resultados Two-Class Boosted Decision Tree 54
Tabela 15 – Resultados Two-Class Bayes Point Machine 55
Tabela 16 – Resultados Two-Class Neural Network 57
Tabela 17 – Resultados Two-Class Locally Deep Support Vector Machine 58
Tabela 18 – Resultados Two-class Decision Jungle 60
Tabela 19 – Resultados Two-class Decision Forest 62
Tabela 20 – Resultados Two-class Averaged Perceptron 64
Tabela 21 – Resultados Two-class Logistic Regression 65
Tabela 22 – Resultados Two-class Support Vector Machine 67
Tabela 23 – Comparação dos resultados das métricas dos algorítimos 68
Tabela 24 – Resultados Logistic Regression 73
Tabela 25 – Resultados Classificador SGD 74
Tabela 26 – Resultados Classificador KNN 75
Tabela 27 – Resultados Classificador Navie Bayes 76
Tabela 28 – Resultados Classificador Decision Tree 79
Tabela 29 – Resultados Classificador Random Forest 79
Tabela 30 – Resultados Classificador Random Forest 80
Tabela 31 – Resultados Classificador Random Forest 81
Tabela 32 – Sensibilidade para casos negativos e Positivos. 81
Tabela 33 – Resultados dos testes na API flask. 83

Lista de abreviaturas e siglas

AM	Aprendizado de máquina
CNC	Computer Numeric Control
IA	Inteligência Artificial
IBM	International Business Machines Corporation
ML	Machine Learning
ONU	Organização das Nações unidas
SAE	Society of Automotive Engineers – sistema de identificação da composição química do aço
SCI	Single Column Inch
SIMCO	Comércio Importação Exportação de Máquinas Ltda
SUS	Sistema Único de Saúde
TP	True Positives(Verdadeiros Positivos)
TN	True Negatives(Verdadeiros Negativos)
FP	False Positives(Falsos Positivos)
FN	False Negatives(Falsos Negativos)
API	Interface de Programação de Aplicações

Sumário

1 INTRODUÇÃO 15
1.1 Objetivos 17
1.1.1 Objetivo geral 17
1.1.2 Objetivo específico 17
1.2 Justificativa 17
1.3 Organização dos capítulos 18

2 TRABALHOS RELACIONADOS 19
2.1 APLICAÇÃO DE MACHINE LEARNING EM DATASET DE CONSULTAS MÉDICAS DO SUS 19
2.2 MACHINE LEARNING APLICADO A USINAGEM: Previsão de informações de um processo de usinagem por Machine Learning com dados de vibração obtidos com aplicativo Sci Journal 21

3 PROCEDIMENTO METODOLÓGICO 26
3.1 Coleta de dados 26
3.2 Levantamento bibliográfico 27
3.3 Análise de artigos científicos 27
3.4 Testar Modelos no Azure ML 27
3.4.1 Encontrar modelos com melhores desempenhos 28
3.5 Implementação dos algoritmos usando Python 28

4 REFERENCIAL TEÓRICO 29
4.1 Doação de sangue 29
4.1.1 Busca por doadores de sangue no Brasil atualmente 29
4.2 Inteligencia Artificial 30
4.2.1 História da IA 30
4.3 Machine Learning 30
4.3.1 Aprendizado Supervisionado 31
4.3.1.1 Algoritmos de classificação 31
4.3.1.2 Árvores de decisão 31
4.3.1.3 Two-class Boosted Decision Tree 33
4.3.1.4 Two-Class Decision Jungle 33
4.3.1.5 Two-class Bayes point Machine 33
4.3.1.6 Two-Class Neural Network 33
4.3.1.7 Two-Class Averaged Perceptron 33
4.3.1.8 Algoritmos de Regressão 33
4.3.2 Aprendizado não Supervisionado 34
4.3.3 O aprendizado reforçado 34

4.4 Computação em Nuvem . . . 35
4.4.1 Tipos de Computação em Nuvem . . . 35
4.5 Azure machine Learning Studio . . . 36
4.6 Python . . . 37
4.6.1 Bibliotecas Python . . . 37

5 APLICAÇÃO DE TÉCNICAS DE IA EM UM DATASET DE DOADORES DE SANGUE . . . 38
5.1 Coleta de dados . . . 38
5.2 Treinamento e avaliação de algoritmos e criação de modelo usando Azure Machine Learning Studio . . . 39
5.2.1 Conjunto de dados . . . 40
5.2.2 Seleção do Dataset no Azure ML . . . 40
5.2.3 Padronizando os dados no Azure ML . . . 41
5.2.4 Separando as amostras de teste e de treinamento . . . 41
5.2.5 Escolha dos modelos de classificação a serem testados . . . 42
5.2.6 Resultados . . . 42
5.3 Criando modelos de Machine Learning usando Python e seu ecossistema 43
5.4 Estudo e definição dos métodos de machine Learning . . . 44
5.4.1 Preparação do ambiente . . . 45
5.4.2 Análise do conjunto de dados usando Pandas . . . 45
5.4.3 Análise de dados usando o Pandas-Profiling . . . 48
5.4.4 Separação dos dados . . . 49
5.4.5 Treinamento e métricas dos modelos . . . 49

6 RESULTADOS DOS EXPERIMENTOS . . . 50
6.1 Resultados do Treinamento e avaliação de algoritmos usando Azure Machine Learning Studio . . . 50
6.1.1 Escolha do Dataset no Azure Studio ML e preparação dos dados . . . 50
6.1.2 Two-Class Boosted Decision Tree . . . 51
6.1.3 Two-Class Neural Network . . . 51
6.1.4 Two-Class Bayes Point Machine . . . 52
6.1.5 Two-class locally deep support vector machine . . . 52
6.1.6 Two-Class Decision Forest . . . 52
6.1.7 Two-Class Averaged Perceptron . . . 52
6.1.8 Two-class Logistic Regression . . . 53
6.1.9 Two-Class Support Vector Machine . . . 53
6.1.10 Two-Class Decision Jungle . . . 53
6.1.11 Resultados . . . 54
6.1.11.1 Two-Class Boosted Decision Tree . . . 54
6.1.11.2 Two-Class Bayes Point Machine . . . 55
6.1.11.3 Two-Class Neural Network . . . 57
6.1.11.4 Two-Class Locally Deep Support Vector Machine . . . 58

6.1.11.5 Two-class Decision Jungle 60
6.1.11.6 Two-clas Two-class Decision Forest 62
6.1.11.7 Two-class Averaged Perceptron 64
6.1.11.8 Two-class Logistic Regression 65
6.1.11.9 Two-class Support Vector Machine 67
6.1.12 Comparação de Resultados 68
6.2 Criando serviço web do Azure Studio 69
6.3 Resultados dos modelos de Machine Learning usando Python e seu ecossistema 72
6.3.1 Logistic Regression 73
6.3.2 Classificador SGD 74
6.3.3 Classificação dos Vizinhos mais Próximos KNN - KNeighborsClassifier 74
6.3.4 Algoritmo Naive Bayes 75
6.3.5 Decision Tree Classifier 76
6.3.6 Random Forest 79
6.3.7 Classificador Extra Tree 79
6.4 Resultados finais dos Classificadores 80
6.5 Criando API utilizando Flask 82
6.5.1 Resultado dos testes dos testes feitos usando API 83

7 CONCLUSÃO 84
7.1 Limitações 84
7.2 Trabalhos Futuros 84

REFERÊNCIAS 85

1 Introdução

"A nova onda de inteligência artificial não nos traz propriamente a inteligência, mas um componente crítico dela: a previsão."
Ajay Agrawal

Como Cerri e Carvalho (2017) citam no artigo, o uso da aprendizagem de máquina (em inglês, Machine Learning) na resolução de problemas do dia-a-dia se popularizou muito nos últimos anos tanto no mundo acadêmico como em todos os setores da economia. A necessidade de otimização e automação de atividades antes realizados por pessoas originou a criação máquinas que assumem a responsabilidade de realizar tais atividades. Para a realização dessas atividades os profissionais se depararam com um problema: como fazer uma máquina ser capaz de resolver problemas e tomar decisões, características presentes no ser humano.

O ser humano possui características que foram decisivas e importantes para sua sobrevivência, ele aprende a tomar decisões para a resolução de problemas através de experiências adquiridas ao longo dos anos, entre erros e acertos nós conseguimos detectar padrões que ajudam nas tomadas de decisão e na resolução de problemas. A tomada de decisão é um fator primordial nas organizações pois ela afeta o rumo do negócio e o sucesso ou não na venda de um produto ou serviço (SBCOACHING, 2020). A atividade humana tem limites naturais, algumas atividades manuais com o tempo geram um alto volume de dados ficando cada vez mais complexo, demorado e humanamente impossível a execução de atividades sem a introdução de máquinas no processo auxiliando ou até mesmo substituindo a atividade humana. Reproduzir essa característica em máquinas e sistemas é o desafio que vários profissionais buscam alcançar. Elaborar e implantar sistemas que produzam auto-conhecimento através de experiências passadas, que é o conceito da inteligência artificial mais especificamente do aprendizado de máquina que é a capacidade de sistemas aprenderem e executarem tarefas de acordo com sua própria experiência (CERRI; CARVALHO, 2017).

A primeira vez que o termo inteligencia artificial surgiu foi com o pai da computação o matemático britânico Alan Turing na década de 1950 que conceituou a capacidade dos computadores de tomar decisões baseado em auto-conhecimento aprendido em experiências de reconhecimento de padrões e comportamentos, numa semelhança ao comportamento humano. Com as limitações da época não foi possível um aproveitamento deste conceito pelo baixo número de informações e dados suficientes para processamento e tecnologia da época limitava e inviabilizava o uso de treinamento de sistemas de inteligência artificial (CADE, 2020).

Os métodos de aprendizado de máquina são algoritmos computacionais que tem o objetivo de emular a inteligência humana aprendendo com experiências do dia-a-dia. No mundo tecnológico atual é nítido o aumento no uso de machine learning em diversas áreas de conhecimento, vemos sua utilização para reconhecimento de padrões, engenharias, saúde em diversos tratamentos onde precisa de um controle de acompanhamento médico, mercado fi-

nanceiro na previsão de tendências de mercado, indústrias como por exemplo o cálculo de valores de variáveis de entradas em processos siderúrgicos e etc (HASTIE; TIBSHIRANI; FRIEDMAN, 2008).

Na área da saúde o uso de aprendizado de máquina para análise preditiva de dados é muito utilizado para estimar o risco de algum evento acontecer como por exemplo o maior risco de determinados indivíduos desenvolverem algum tipo de doença baseado em análise de dados físicos, características socioeconômicas,demográficas, relacionadas ao hábito de vida e às condições de saúde, entre outras (SANTOS et al., 2019).

Como vimos anteriormente o aprendizado de máquina não é uma área de estudo recente da tecnologia, mas o aumento no número e no volume de dados fez surgir recentemente algorítimos capazes de aplicar cálculos matemáticos ao Big Data. As grandes industrias e corporações que trabalham com um tamanho elevado de dados viram no aprendizado de máquina a oportunidade de ganhar eficiência nas operações e assim ter uma vantagem competitiva analisando e identificando padrões são capazes de captar oportunidades de negócio ou de evitar riscos(SAS, s.d). As características dos dados analisados e o tipo de resposta é o que define o tipo de método de aprendizado de máquina que será usado.

Entre os tipos de aprendizado de máquina os mais conhecidos são : método de aprendizado supervisionado que se caracteriza pelo sistema treinado a partir de um conjunto de dados pré-definidos, aprendizado não supervisionado em que o sistema encontra padrões a partir de um conjunto de dados automaticamente e o aprendizado por reforço em que o sistema deve aprender a identificar o caminho a seguir e tomar decisões baseados no método de tentativa e erro (MELO, 2020).

O uso dessas técnicas de aprendizado de máquina permite o treinamento de sistemas pelo estudo de padrões e características de dados que tornam possível a tomada de decisões e reconhecimento de padrões. Esse tipo de técnica consegue por conta própria aprender a tomar decisões para determinados problemas e ou ações, conseguindo também aprimorá-los através da repetição e análise de dados comparando seu desempenho e sendo alimentado com novas informações (PATEL, 2021).

A escolha das técnicas a serem utilizadas leva em conta o objetivo e as características dos dados a serem manipulados, pois não existe um algoritmo perfeito que resolva todos os problemas mas sim um ideal que vai alcançar resultados mais indicado para a necessidade do sistema (BERTOZZO, 2019).

Como explicado por Santos et al. (2019) o uso de tais métodos na área da saúde no Brasil ainda está na fase inicial, o aumento da população e o risco de colapso do sistema de saúde faz se necessário o uso da tecnologia que auxilie na melhora do atendimento para população. Entre o uso de aprendizado de máquina no sistema de saúde no Brasil temos como exemplo o estudo realizado por Olivera et al. (2017) que consiste no estudo de modelo de diabete não identificada a partir de dados de 12.447 adultos utilizados algoritmos de aprendizado de máquina g (regressão logística, redes neurais,naive bayes, método dos K vizinhos mais próximos e random forest)e o que demonstrou a importância dessa ferramenta no auxílio para

profissionais de saúde do Brasil.

A doação de sangue no Brasil está abaixo do recomendado pela Organizações das Nações Unidas (ONU), a recomendação na taxa de doação de sangue em relação ao total da população é de 3%, dados do Ministério da Saúde indicam uma taxa abaixo do recomendado,no Brasil a taxa de doação não chega a 2% do total de nossa população (CATHO, 2015).A falta de incentivos e a falta de informações são problemas que podem explicar a baixa porcentagem de doadores em relação a população total . A pandemia de Covid-19 fez esses números piorarem, dados da Agência Brasil (MELO, 2021) mostraram que em 2020 houve uma queda de 20% no número de doações em comparação ao período da pesquisa ano anterior.

Baseado nesse cenário, utilizando uma base de dados real de doadores de sangue o aprendizado de máquina pode ser usado para classificação dos indivíduos em doadores em potencial ?

Assim o objetivo deste trabalho é fazer uma análise exploratória de diferentes técnicas de Machine Learning usando um conjunto de dados históricos reais de doadores de sangue para classificação e previsão de doadores em potencial.

1.1 Objetivos

1.1.1 Objetivo geral

Explorar diferentes técnicas de Aprendizado de máquina em um conjunto de dados de doadores de sangue.

1.1.2 Objetivo específico

1. Apresentar e treinar diferentes algoritmos de aprendizado de máquina e apresentar os resultados.

1.2 Justificativa

Esta pesquisa justifica-se pelo próprio objetivo, avaliar características de doadores de sangue e descobrir padrões utilizando um Dataset com informações de doadores aplicando algoritmos de machine learning supervisionados criando os modelos preditivos e avaliando a performance de cada um com a finalidade de escolher um modelo que seja capaz de classificar doadores de sangue em potencial.

O uso de aplicações de aprendizado de máquina na área da saúde é bem recente mas que está em constante crescimento, e o uso delas está ajudando pacientes e profissionais, as áreas mais usadas ainda são as áreas administrativas como faturamento hospitalar e algumas tomadas de decisões.

Recentemente o uso de algorítimos de aprendizado de máquina começou a ser estudado e utilizado também em consultórios, laboratórios e centros cirúrgicos na detecção de

algumas doenças, diagnósticos médicos e cirurgia por robôs automatizados (NEXXTO, 2021) .

Em situação normal os níveis de estoque de sangue no Brasil estava abaixo do encontrado em outros países da América latina, a situação só piorou com a chegada da pandemia, pois a falta de informação sobre essa nova doença trouxe o receio e o medo que fez com que muitas pessoas deixassem de fazer suas doações. O aumento no número de atendimentos causada pelo aumento de atendimentos de pessoas diagnosticadas com o covid-19 fez com que a rotina hospitalar mudasse completamente, agora o foco está definido para tratamento da covid-19 de modo emergencial e migrando com isso profissionais das mais diversas áreas para dar um apoio no atendimento dessas pessoas, o que ocasionou em alguns lugares a suspensão das doações de sangue. Esse aumento de pacientes nos hospitais causou nas pessoas o medo de contrair a doença no ambiente hospitalar e até mesmo a falta de comunicação reduziu em muito o número de doações, só no Rio de janeiro o estoque de sangue caiu 38% em relação aos números do ano passado (PIMENTA; SOUZA, 2020).

Diversos periódicos tratam o problema da falta de captação de doadores e da fidelização dos mesmos pelo pouco uso de estratégias de captação de doadores. Algumas das estratégias mais citadas em artigos é o estudo dos perfis psicológicos, comportamentais e sociais dos doadores afim de encontrar padrões que ajudem a identificar potenciais doadores (RODRIGUES; REIBNITZ, 2011).

A falta de uma proximidade entre doadores e os centros de coleta de sangue dificulta a comunicação entre as partes o que explica em muitos casos a falta de doações em tempos normais e no agravamento em tempos de pandemia como a que estamos presenciando. Esses fatos mostram a importância de se desenvolver mecanismos que otimizem a busca por doadores e que deem um auxílio nas tomadas de decisões e escolha de estratégias para alcançar doadores em potencial, e o uso de técnicas de machine learning para mapear os perfis de doadores classificando de acordo com o potencial de doações recorrentes.

1.3 Organização dos capítulos

Este trabalho está divido da seguinte forma: o capítulo 2 apresenta como será o procedimento metodológico do trabalho e explicando cada etapa do trabalho; o capítulo 3 fará uma descrição de trabalhos relacionados; o capítulo 4 fará uma descrição dos assuntos relacionados a inteligência artificial, algorítimos e o ecossistema da tecnologia; no capítulo 5 será apresentado os passos para aplicação das técnicas de aprendizado de máquina no dataset usando o Azure Studio e no Jupyter Notebook; no capítulo 6 será apresentados os resultados dos treinamentos e o capítulo 7 as conclusões dos estudos e ideias para trabalhos futuros.

2 Trabalhos Relacionados

"Não se possui o que não se compreende".
Johann Goethe

A utilização de de inteligência artificial tanto nas indústrias e empresas de tecnologia e no meio acadêmico vem numa crescente nos últimos anos e estudos envolvendo o tema caracterizam por diversos tipos de abordagens. Foi feito um levantamento bibliográfico para compreender e analisar as metodologias adotadas em trabalhos que envolviam o uso de métodos de machine learning.

As pesquisas foram feitas utilizando a plataforma de pesquisa Google Scholar lançada em 2004 e que foi desenvolvida pela Google. O Google Scholar é uma plataforma que reúne um acervo de publicações, monografias, teses, fontes para referencial teórico e muito mais (LINS, 2021).

Os trabalhos escolhidos foram : APLICAÇÃO DE MACHINE LEARNING EM DATASET DE CONSULTAS MÉDICAS DO SUS (BERTOZZO, 2019) e MACHINE LEARNING APLICADO A USINAGEM: Previsão de informações de um processo de usinagem por Machine Learning com dados de vibração obtidos com aplicativo Sci Journal (PAZ, 2019).

2.1 APLICAÇÃO DE MACHINE LEARNING EM DATASET DE CONSULTAS MÉDICAS DO SUS

Em seu trabalho BERTOZZO estudou o problema do sistema do Sus que apresenta sérios problemas que afetam a população, entre esses problemas está a das filas que são tão comuns em postos e hospitais. O objetivo do trabalho é desenvolver um método que possibilite fazer uma previsão se o paciente é mais provável a faltar ou não. A primeira etapa do trabalho foi a coleta de dados e para isso usando a plataforma Kaggle foi feito o download da base de dados que ele utilizou. Essa base de dados possui informações de consultas no SUS e possuía um total de 110.527 registros da cidade de Vitória - ES.

A segunda etapa consistiu na escolha da ferramenta para aplicação, a linguagem escolhida foi o Python pois possui um ambiente de desenvolvimento que possui uma base muito grande de bibliotecas que auxiliam no aprendizado de máquina. Foi escolhido utilizar também o Google Colab pela facilidade de poder através do navegador e sem precisar instalar nada na máquina local realizar o processamento de dados de onde quer que esteja, precisando apenas de um acesso a internet.

A próxima etapa consistiu na preparação dos dados para deixar pronto para serem utilizados conhecendo melhor os dados, definição de quais características utilizar e fazer a padronização dos valores.

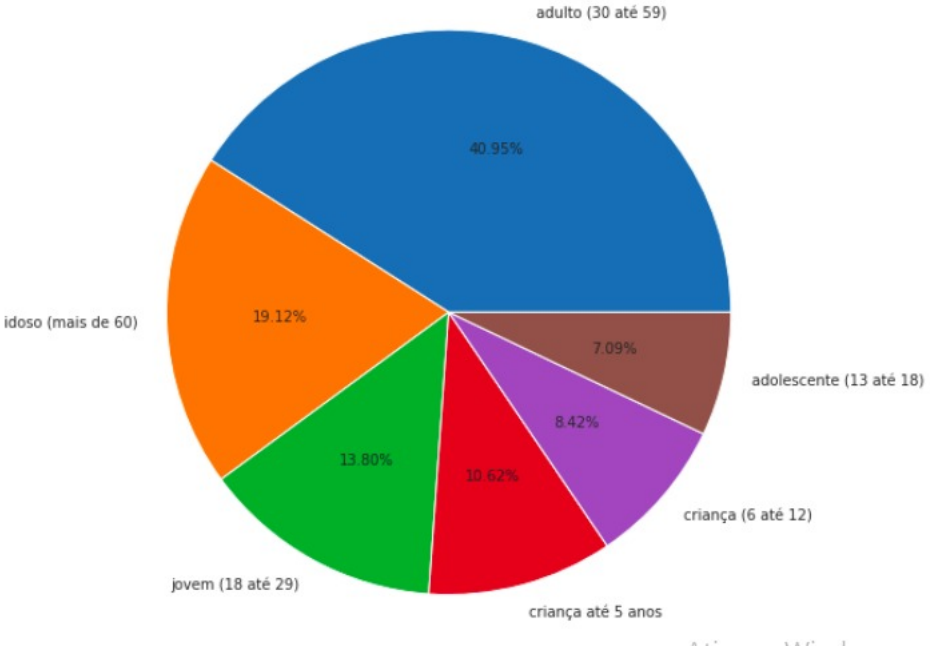
Com a base de dados preparada a próxima etapa foi a utilização de várias abordagens de AM como supervisionado e não supervisionado utilizando alguns algorítimos de cada tipo para encontrar o que teria a melhor performance.

De posse dos resultados de cada uma das aplicações realizadas por cada algoritmo foram feitas uma exploração dos resultados obtendo as métricas das aplicações para que seja feito a comparação de performance, acurácia e outras características importantes para essa abordagem.

Para esta exploração de dados utilizou-se a biblioteca do Python o Pandas carregando o arquivo formato CSV e com essa biblioteca é possível trabalhar de várias maneiras pois ela fornece ferramentas que possibilitam uma exploração do data-frame. Para a geração de gráficos foram utilizados as bibliotecas Matiplotlib e Seaborn.

Como exemplo dos gráficos criados temos a figura 1 que é um gráfico que mostra a distribuição em porcentagem de registros de consultas do conjunto de dados em relação a coluna de faixa de idade dos pacientes.

Figura 1 – Percentual de registros de consulta em relação às faixas de idades dos pacientes..



Fonte: (BERTOZZO, 2019)

A próxima etapa agora é aplicação dos algoritmos utilizando duas bibliotecas do Python o scikit-learn que possui um número grande de algoritmos de machine learning e o Keras para a aplicação de redes neurais.

As características estuadas e definidas para serem usadas no treinamento foram as seguintes :

- IsMale - se o paciente é do gênero masculino ou do gênero feminino;

- ZoneAge - faixa de idade do paciente (exemplo: criança);
- Hypertension - se o paciente tem problemas de hipertensão;
- Diabetes - se o paciente tem problemas de diabetes;
- Alcoholism - se o paciente é alcoólatra;
- Handicap - quantidade de deficiências que o paciente tem (0 até 4);
- SMS-received - se o paciente recebeu mensagem de aviso da consulta;
- ZoneAwaitingTime - faixa de tempo espera para a consulta (exemplo: rápido);
- NeighborhoodNum - bairro onde acontece a consulta (número dele);

Após a execução dos algoritmos foram feitas análises bem elaboradas no trabalho em questão com gráficos e tabelas que mostravam entre outros resultados a acurácia de cada algoritmo, a matriz de confusão, o número de acertos e erros entre outros resultados. Os resultados finais mostraram que os algoritmos supervisionados tiveram uma porcentagem de acerto de 70% ou mais de acurácia e métricas sem muitas diferenças. No melhor modelo estudado chegou a uma acurácia de 80% com o algoritmo de árvore de decisão. Os autores concluíram que os algoritmos estudados podem ser usados para apoio a gestão de consultas do SUS para amenizar a situação de filas, atingindo assim o objetivo do trabalho.

2.2 MACHINE LEARNING APLICADO A USINAGEM: Previsão de informações de um processo de usinagem por Machine Learning com dados de vibração obtidos com aplicativo Sci Journal

O problema explorado pelo autor é a aquisição de informações e monitoramento em tempo real de processos nas indústrias mais especificamente o de sensores em máquinas industriais. O trabalho consiste em coletar informações de vibração em um processo de usinagem pelo esforço de corte de material, com os sensores é possível coletar informações sobre o estado da máquina, do material que está sendo processado, da qualidade do produto resultante e etc. O trabalho em questão propõe utilizar sensores de acelerômetros presentes na maioria dos smartphones para obter os dados de processo de usinagem através de aplicativo de coleta de dados Sci Journal para uso destes dados em um algoritmo de Machine Learning que receberá as informações obtidas e com esses dados prever informações a cerca do resultado desse processo.

O trabalho foi desenvolvido utilizando um Centro de Usinagem CNC Feeler VMP-30A como mostrado na figura 2 usando um bloco de aço SAE 1045 e uma fresa insertada com 2 arestas de corte e 16 mm de diâmetro. O aparelho celular que foi utilizado no trabalho foi um smartphone modelo IPhone 7 com o aplicativo Sci Journal desenvolvido pela Google instalado. Para se obter as informações da peça foi utilizado um Rugosímetro SJ-310.

Figura 2 – Centro de Usinagem CNC Feeler VMP-30A.

Fonte: (PAZ, 2019)

Foram definidos os parâmetros para a usinagem com os valores presentes na tabela 1.

Tabela 1 – Parâmetros de usinagem utilizados no experimento.

Parâmetros de Processo	Nível 1	Nível 2	Nível 3
Profundidade (mm)	0,1	0,2	0,5
Avanço (mm/min)	500	1000	2000
Rotação da Ferramenta (RPM)	1000	2500	5000

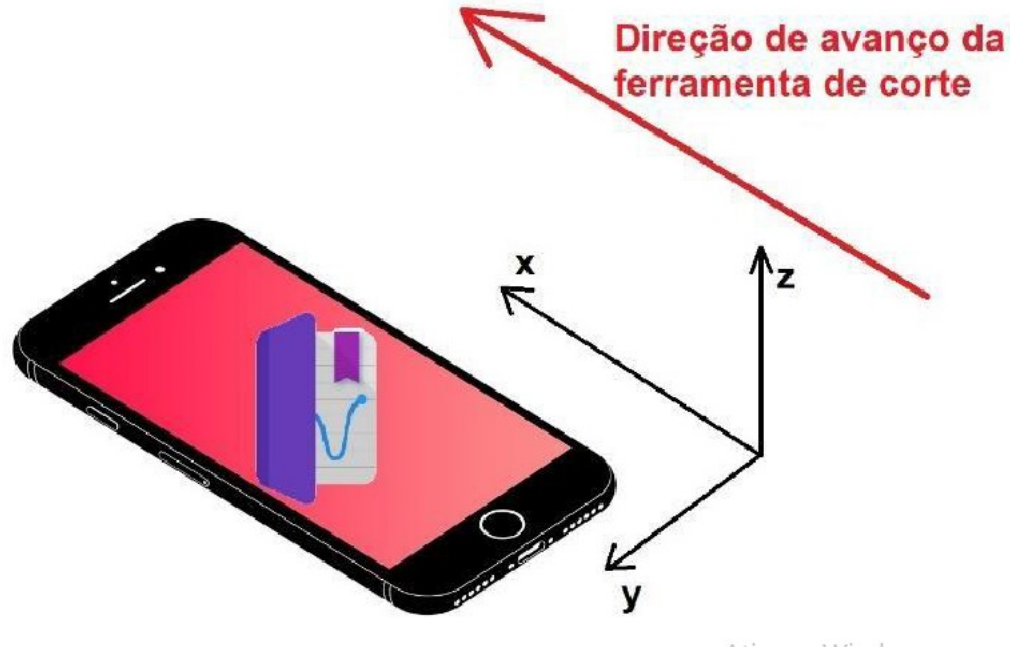
Fonte: (PAZ, 2019).

Foram realizadas 27 usinagens para a obtenção de dados utilizando pares de avanço e rotação em que cada combinação foi realizada 3 vezes, os resultados obtidos estão na tabela 2:

Para se obter os dados o o smartphone foi fixado na morsa de fixação do centro de usinagem para que os sensores juntamente com o aplicativo Sci journal medissem a vibração, foram obtidos as informações referentes as vibrações nos eixos x,y,z, além da vibração linear. O modo de fixação foi feito de acordo com a ilustração feita pelo autor na figura 3.

A próxima etapa foi a criação de um código para aplicação do algoritmo de Aprendizado de Máquina, utilizando a linguagem Python e sua biblioteca Scikit.Learn utilizou-se o módulo

Figura 3 – Orientação dos eixos de medição do aplicativo Sci Journal e do smartphone em relação a direção do processo de usinagem.

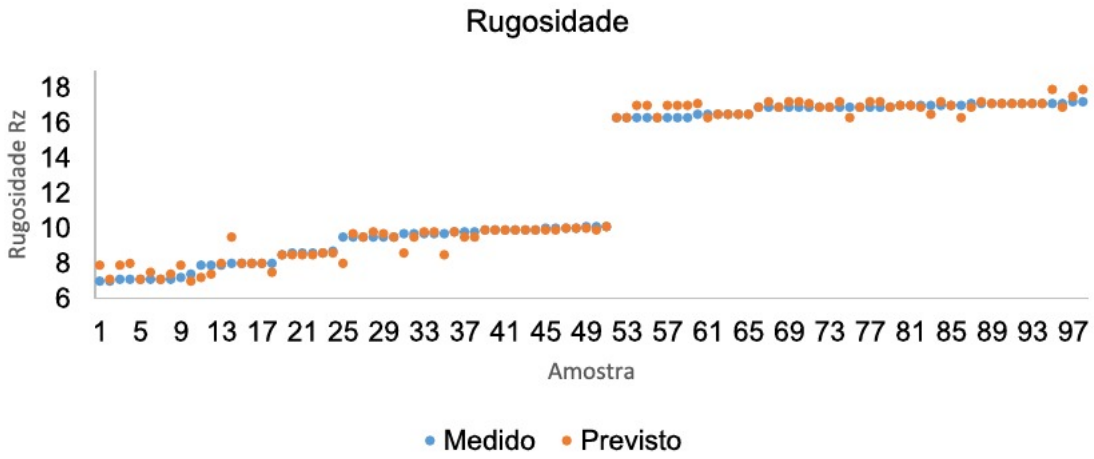


Fonte: (PAZ, 2019).

decision.tree.regressor. Os resultados da usinagem foram transformados em dados para a utilização do treinamento no modelo de Machine Learning.

Com os resultados do treinamento em mão foi feito um gráfico com os resultados obtidos durante a usinagem comparados aos dados previstos no modelo Decision Tree Regression para rugosidade da peça. Os dados obtidos estão apresentados na figura 4.

Figura 4 – Comparação dos valores de rugosidade obtidos nos testes do algoritmo e dos valores medidos.



Fonte: (PAZ, 2019).

Foi feito o cálculo dos valores previstos com os valores reais das principais saídas do

modelo, os valores calculados podem ser encontrados na tabela 2.

Tabela 2 – Testes do experimento.

Teste	Profundidade (mm)	RPM	Avanço (mm/min)
1	0,1	2500	1000
2	0,1	2500	1000
3	0,1	2500	1000
4	0,3	2500	1000
5	0,3	2500	1000
6	0,3	2500	1000
7	0,5	2500	1000
8	0,5	2500	1000
9	0,5	2500	1000
10	0,1	5000	500
11	0,1	5000	500
12	0,1	5000	500
13	0,3	5000	500
14	0,3	5000	500
15	0,3	5000	500
16	0,5	5000	500
17	0,5	5000	500
18	0,5	5000	500
19	0,1	1000	2000
20	0,1	1000	2000
21	0,1	1000	2000
22	0,3	1000	2000
23	0,3	1000	2000
24	0,3	1000	2000
25	0,5	1000	2000
26	0,5	1000	2000
27	0,5	1000	2000

Fonte: (PAZ, 2019).

O autor conclui que o modelo Decision Tree Regresion obteve uma precisão de 96,15%(como visto na tabela 3) para a rugosidade da peça com o auxílio do aplicativo Sci Jounarl, e conclui que um aumento no número de dados para usar no algoritmo poderia melhorar esses números já que o algoritmos de Machine Learning tendem a ser cada vez mais precisos com um maior número de dados e de treinamentos. Como conclusão do trabalho ele conseguiu demonstrar que o aplicativo Sci Jounral é capaz de obter dados de processo usando os acelerômetros presentes nos celulares e concluiu também que o algorítimo Decision Tree Regresion se mos-

Tabela 3 – Precisão dos resultados Previstos pelo modelo Decision Tree Regresion

Variável de saída	Precisão
Rotação (RPM)	96,78%
Avanço 9mm/min)	96,64%
Rugosidade (µm)	96,15%

Fonte: (PAZ, 2019).

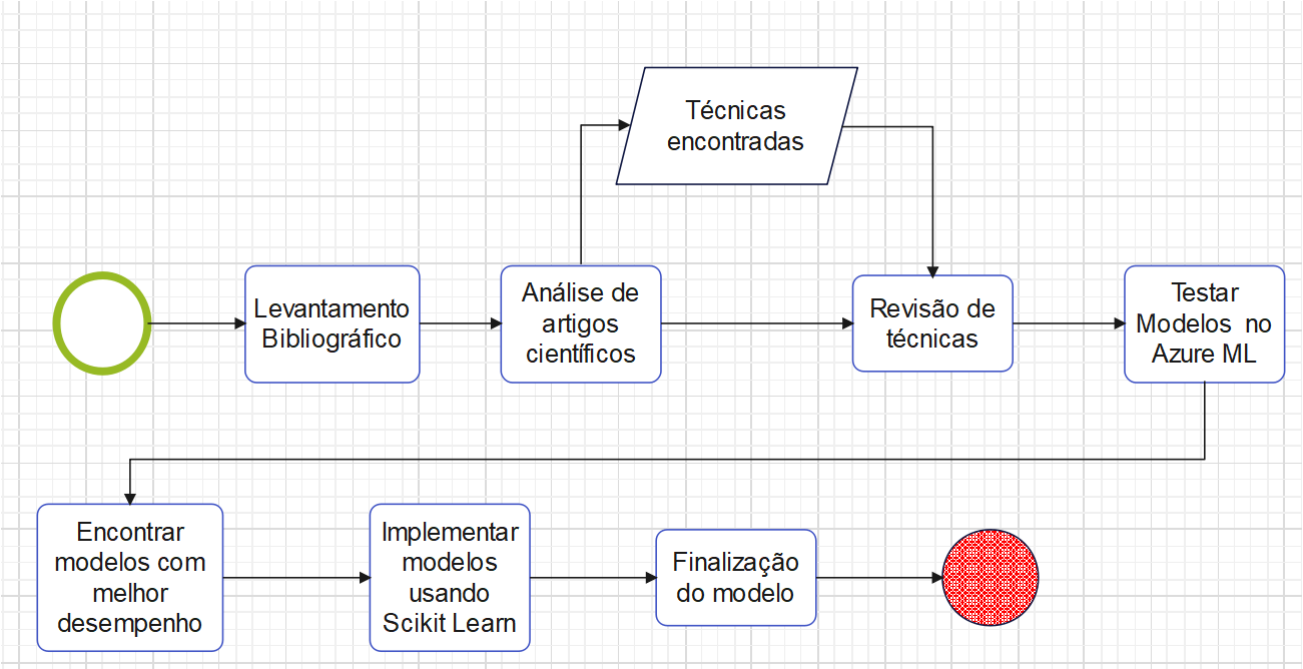
trou eficaz na previsão de parâmetros de qualidade a partir de dados de entrada obtidos do processo de usinagem como a utilizada que no caso em questão foi a vibração.

3 Procedimento Metodológico

“A longo prazo, inteligência artificial e automação tomarão muito do que dá aos humanos um sentimento de propósito”.
Matt Bellamy

Esta seção descreve como foi a metodologia adotada neste trabalho. A figura 5 mostra todas as fases seguidas e a seguir nesta seção uma explicação de cada uma das fases citadas.

Figura 5 – Etapas do trabalho.



Fonte:O autor.

3.1 Coleta de dados

A coleta de dados pode se realizar de várias formas como uma pesquisa feita com formulário para a coleta de dados de forma manual, pode se coletar informações através de banco de dados de uma organização e assim criar uma tabela com dados históricos e também pode-se coletar dados em repositórios online em plataformas que disponibilizam estes arquivos em diversos formatos. Através dos levantamentos bibliográficos os bancos de dados mais usados são os sites Kaggle, UCI Machine Learning Repository e Datahub.Io que disponibilizam alguns dos conjuntos de dados mais utilizados no meio acadêmico.

3.2 Levantamento bibliográfico

Nesta etapa formulamos o objetivo e as estratégias para a criação do trabalho acumulando conhecimento através de trabalhos acadêmicos e pesquisas científicas realizadas por pesquisadores e alunos. Com a definição do tema devemos pesquisar soluções variadas realizadas para servir de base para nossas soluções.

A escolha de uma boa fonte de banco de dados para se levantar fontes bibliográficas é uma importante etapa para garantir uma qualidade dos materiais que será usado. Para garantir uma melhor qualidade e confiabilidade deve-se ter uma boa organização na pesquisa, ter uma base de dados reconhecida e de boa reputação e dar preferência para trabalhos e artigos mais recentes.

3.3 Análise de artigos científicos

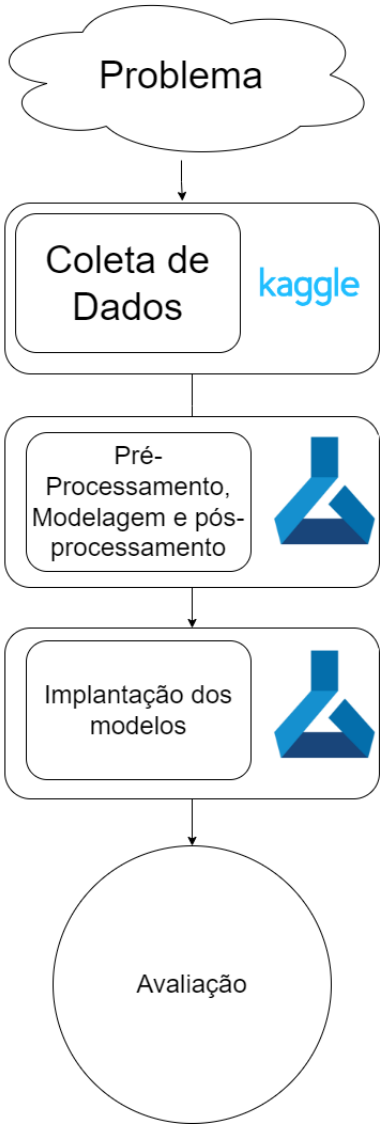
Após o levantamento bibliográfico poderemos fazer uma análise crítica dos artigos científicos, analisando atividades e técnicas usadas para resolução de problemas semelhantes ao tema central do nosso trabalho.

3.4 Testar Modelos no Azure ML

Em um dos artigos encontrados foi usado como ferramenta de treinamento e de criação de serviço o Azure ML Studio, após leitura da documentação, tutoriais e alguns cursos ele foi escolhido para ser usado como plataforma de ciência de dados que cria em um alto nível de programação em nuvem o modelo de Machine Learning o que garante um melhor desempenho e sem o custo de Hardware pelo usuário além de garantir uma disponibilidade do serviço em qualquer lugar através da internet. Uma explicação mais detalhada do Azure Studio será feita em capítulos posteriores.

O Azure Machine Learning Studio é uma ferramenta disponibilizada pela Microsoft que permite a criação e treinamento de modelo de aprendizado de máquina usando a nuvem, garantindo assim uma performance e disponibilidade que os serviços de nuvem da Microsoft oferecem. A descrição das etapas para criação de treinamento de modelos de machine learning usando o Azure ML Studio será descrito nas seções a seguir. Na figura 6 mostra as etapas que serão realizadas desde a coleta de dados até a avaliação dos modelos usando os serviços de nuvem do Azure ML Studio.

Figura 6 – Etapas para criação de modelo usando Azure ML Studio (Classic)



Fonte: O autor

3.4.1 Encontrar modelos com melhores desempenhos

Nesta etapa será comparado utilizando os dados gerados pelo Azure Studio a performance de cada algorítimo para escolher o que melhor performou na criação do modelo de predição. E com essa definição criar também um modelo usando Python.

3.5 Implementação dos algoritmos usando Python

Nesta etapa será implementado os principais algoritmos encontrados nos levantamentos bibliográficos utilizando o Python, suas bibliotecas como a Pandas, Pandas-Profiling, NumPy, SciPy, Matplotilib, um micro-framework Flask e a principal biblioteca do trabalho, a biblioteca com as principais ferramentas para aprendizado de máquina: o Scikit Learn.

4 Referencial Teórico

“Você quer passar o resto da sua vida vendendo água com açúcar ou quer uma chance de mudar o mundo?”.
Steve Jobs

4.1 Doação de sangue

Desde os anos iniciais da era moderna da humanidade são realizados estudos sobre a importância do sangue para a cura de males e doenças. A primeira transfusão de sangue foi realizado entre dois cães em 1665. A primeira transfusão de sangue em humanos é datado de 1667 por foi Jean Baptiste Denis, professor de filosofia e matemática em Montpellier e médico do rei Luis XIV onde foi introduzido sangue de carneiro em um homem com problemas mentais. A primeira transfusão com sangue humano foi realizada por James Blundell, em 1818 que, após realizar com sucesso experimentos em animais, transfundiu sangue humano em mulheres com hemorragia pós-parto (PROSANGUE, s.d).

4.1.1 Busca por doadores de sangue no Brasil atualmente

A doação de sangue no Brasil é feita de quatro formas : voluntária (o doador faz a doação para um banco de sangue), vinculada (o doador repõe a quantidade de sangue utilizada no tratamento de um conhecido), específica (sangue destinado a uma paciente específico) e autotransfusão (a pessoa pode guardar seu próprio pangue para ser utilizado futuramente) (VEJASAUDE, 2016).

De acordo com VEJASAUDE para ser um doador no Brasil os interessados devem seguir os seguintes passos :

1. Os interessados devem se encaminhar a um centro de doação de sangue com os documentos pessoais onde será feito o cadastro pessoal.
2. Serão realizados as triagem clínicas e hematológicas para averiguação das características físicas para saber se o candidato está apto a doar.
3. A ultima etapa é a própria doação onde serão feitas ainda a verificação de doenças infecciosas.

Após a doação passando por todos os testes será feito o envio e autorizado para transfusão. O sangue retirado para doação é reposto no corpo humano em até oito semanas, cada doação pode salvar a vida de até quatro pessoas(VEJASAUDE, 2016).

4.2 Inteligencia Artificial

A Inteligencia artificial não tem uma definição exata nos meios acadêmicos, ela é um ramo da tecnologia que tem como alvo o desenvolvimento de sistemas que sejam capazes de resolver por conta própria problemas sem nenhum ou com o mínimo de intervenção humana (SICHMAN, 2021). A IA vem com o propósito de resolver problemas que a computação convencional não tem capacidade de realizar e também de exercer atividades que atualmente são desenvolvidas por seres humanos.

4.2.1 História da IA

Desde o início do século 20 já se falava em inteligência artificial. Obras acadêmicas e filmes já tinham como tema a criação de máquinas que tinham inteligência e que tomavam decidões como por exemplo no filme *Metrópolis* de 1927 onde nesse filme já apareceia um adróide. Anos antes em 1921 uma peça de teatro produzida pelo escritor checo Karel Čapek apresentava como tema seres humanóides artificiais que possuiam inteligência. Em 1950 Alan Turing desenvolveu o **teste de Turing** que era um teste em que era analisado se uma máquina conseguia se passar por um humano. Alguns outros acontecimentos que sucederam foram os seguintes (TOTVS, 2019):

- Em 1951, Marvin Minsky desenvolveu uma calculadora de operações matemáticas imitando sinapses — o SNARC;
- Em 1952, Arthur Samuel desenvolveu um jogo de damas no primeiro computador científico comercial da IBM, o IBM 701. Esse jogo conseguia se otimizar por conta própria;
- Em 1956, ocorreu uma conferência no campus da Dartmouth College, em que se reuniram alguns dos citados com outros nomes importantes, como Nathan Rochester e John McCarthy. Esse último batizou a área de Inteligência Artificial. Na conferência, também surgiram alguns eixos que conceituaram e passaram a nortear o campo de pesquisa da IA;
- Em 1957, é apresentado, por Frank Rosenblatt, o perceptron. Trata-se de um algoritmo que funciona como um tipo de rede neural artificial, de uma camada, que classifica resultados. É um classificador linear.

4.3 Machine Learning

Usando a tradução do nome Machine Learning (Aprendizagem de máquina) podemos definir como a aprendizagem que uma máquina vai adquirir através de processos que envolvem a repetição de atividades mudando com isso seu comportamento habitual com base nessas experiências (ALECRIM, 2018).

A aprendizagem de máquina vem sendo usada cada vez mais nos dias atuais como por exemplo em plataformas de streaming como Netflix, Youtube, Spotify por exemplo, você recebe

recomendações de conteúdos baseado no que você em mais consumido nessas plataformas (ALECRIM, 2018).

A aprendizagem de máquina pode ser realizado de três formas diferentes que serão explicadas a seguir:

4.3.1 Aprendizado Supervisionado

A aprendizagem supervisionada soluciona problemas e a partir de dados ela treina algoritmos para realizar alguma tarefa. No aprendizado supervisionado o ser humano passa para máquinas as características mais importantes que a máquina precisa para por exemplo identificar o que é um carro, ensinando que um carro tem quatro rodas, volante, motor e etc. A máquina passa a reconhecer padrões aprendendo assim a definir se tal objeto é um carro ou não (FREITAS, 2019).

Como citado em seu artigo LAURETTO os métodos de aprendizagem supervisionada necessita de conjunto de exemplos também denominado conjunto de treinamento aprendizado. O problema de aprendizado supervisionado é encontrar, a partir do conjunto de treinamento um classificador que associe a cada vetor de atributos uma classe.

O Aprendizado supervisionado é dividido em algoritmos de classificação e de regressão.

4.3.1.1 Algoritmos de classificação

Tem como finalidade utilizar um conjunto de dados previamente classificados e através de treinamento criar um modelo capaz de prever a classificação de dados futuros (AWARI, 2020) . Os algorítimos de classificação podem ser de dois tipos : De duas classes chamadas de Binárias e Multi-classe que possuem a partir de três classes. Faremos a seguir uma descrição de alguns dos algorítimos utilizados em nosso trabalho

4.3.1.2 Árvores de decisão

De forma resumida algorítimos do tipo árvore de decisão são mapas de resultados de escolhas com base em valores, probabilidades e custos. A semelhança como uma árvore vem na ramificação dos resultados onde cada resultado funcionam como um nó onde se dividem em dois possíveis resultados e os resultados também se dividem em mais ramificações.

Cada nó de uma árvore de decisão tem três funções: nó de decisão que é uma ação a ser tomada; nó de probabilidade: que mostra a probabilidade de um certo resultado e o nó final que mostra o resultado da decisão.

O funcionamento de uma árvore de decisão é de fácil compreensão, começamos com uma nó de decisão com a regra do sim ou não onde é analisado um valor x com uma regra tendo como resposta sim ou não, por exemplo: "Esse carro é de duas ou quatro portas?", após a escolha do resultado é feito uma nova ramificação com uma nova decisão("Esse carro é novo ou usado?"), e assim vai até chegar em um nó final.

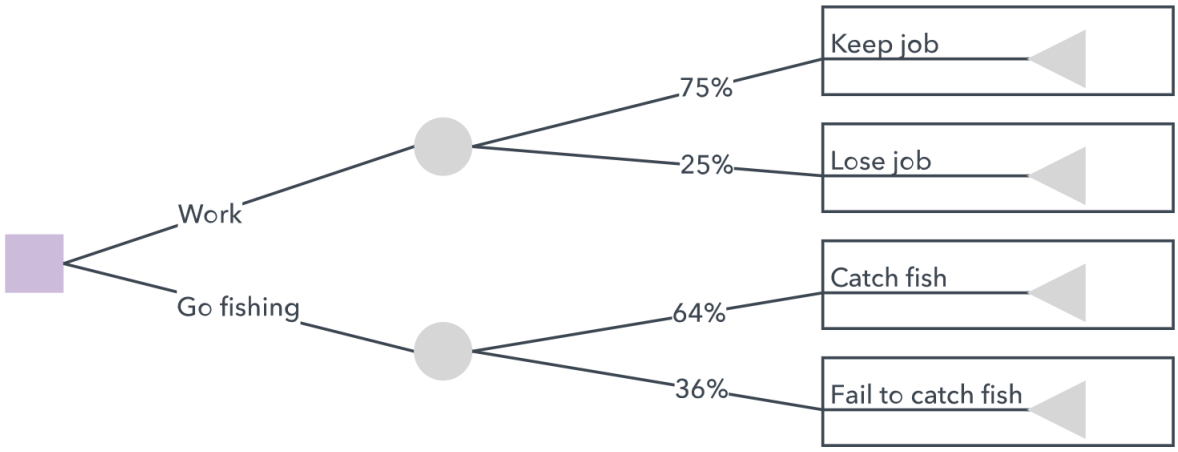
A árvore tem como função organizar os dados de modo a encaixar as características do dado em cada uma dessas posições na árvore, quem será o nó raiz e os nós resultantes, fazendo uma série de cálculos matemáticos para realizar esse agrupamento calculando a entropia das classes de saída e a ganho de informação dos atributos da base de dados.

Por ter uma semelhança com um fluxograma a árvore de decisão facilita a visualização e compreensão das etapas do processo de decisão como por exemplo na figura 7.

Na árvore de decisão usada para criar modelos preditivos os nós são dados e cada ramificação um conjunto de atributos associado a um rótulo de classe. Essas regras de decisão são do tipo se então, em que cumpridos as condições das regras terá um desfecho x com certeza y. Quanto mais dados adicionais a árvore tiver mais precisão terá a árvore na hora de decidir em qual grupo tal sujeito pertence. Para aumentar a precisão várias árvores são usadas e combinadas em métodos de conjunto:

- Ensacamento : Constituem de várias árvores de decisão onde elas em conjunto escolhem o melhor resultado.
- classificador de floresta aleatória (Random Florest): Cria várias árvores de decisão e as combina para obter uma predição com maior acurácia e mais estável.´
- Boosted Tree: Cada nova árvore é construída considerando os erros das árvores anteriores.
- Floresta de rotação: Treinadas usando PCA em uma parte aleatória dos dados.

Figura 7 – Exemplo de Árvore de decisão



Fonte: (LUCIDCHART, 2022)

È descrito a seguir alguns algoritmos que usam árvores de decisão e que foram usadas nesse trabalho.

4.3.1.3 Two-class Boosted Decision Tree

O algoritmo Two-class Boosted Decision Tree (Árvore de decisão impulsionada de duas classes) funciona de uma maneira resumida onde uma árvore corrige os erros da árvore anterior, a próxima árvore corrige a da atual e assim por diante. Arvores de decisão são gráficos que ilustram todos os resultados possíveis de uma decisão usando uma abordagem de ramificação. A previsão é baseada no conjunto total das árvores que fazem a previsão.

4.3.1.4 Two-Class Decision Jungle

É um algoritmo de classificação que consiste em muitas árvores de decisão combinadas para obter um resultado mais preciso em comparação com uma única árvore. Em relação ao Decision Tree ele consome mais computação. O processo de geração e análise consome muito tempo. E é mais difícil de visualizar em relação a árvore de decisão simples (TALARI, 2022).

4.3.1.5 Two-class Bayes point Machine

O pesquisador JOSHI detalha em seu Blog o funcionamento do algoritmo seguindo a definição como: um classificador Bayes ótimo faz a predição para qualquer ponto a classe que minimiza o erro sobre todos os limites possíveis e todas as amostras. Como os dados pode ser infinitamente grandes o limite é encontrado em um espaço fixo que está mais próximo do ponto. É sorteado valores de alguns pontos onde é feita a avaliação d previsão de Bayes nesses pontos, escolhendo a mais próxima das previsões ótimas de Bayes. è feito uma linha reta e a partir dos dados de entrada verifica-se em que lado da fronteira esse ponto está .

4.3.1.6 Two-Class Neural Network

O uso de redes neurais é indicado para prever resultados binários como o caso da previsão de doadores em potencial onde o objetivo é saber se um doador irá doar em um dia específico ou não, para isso uma rede neural necessita de dados marcados. Este algorítimo permite a criação de um modelo de previsão de destino que tenha dois valores.

4.3.1.7 Two-Class Averaged Perceptron

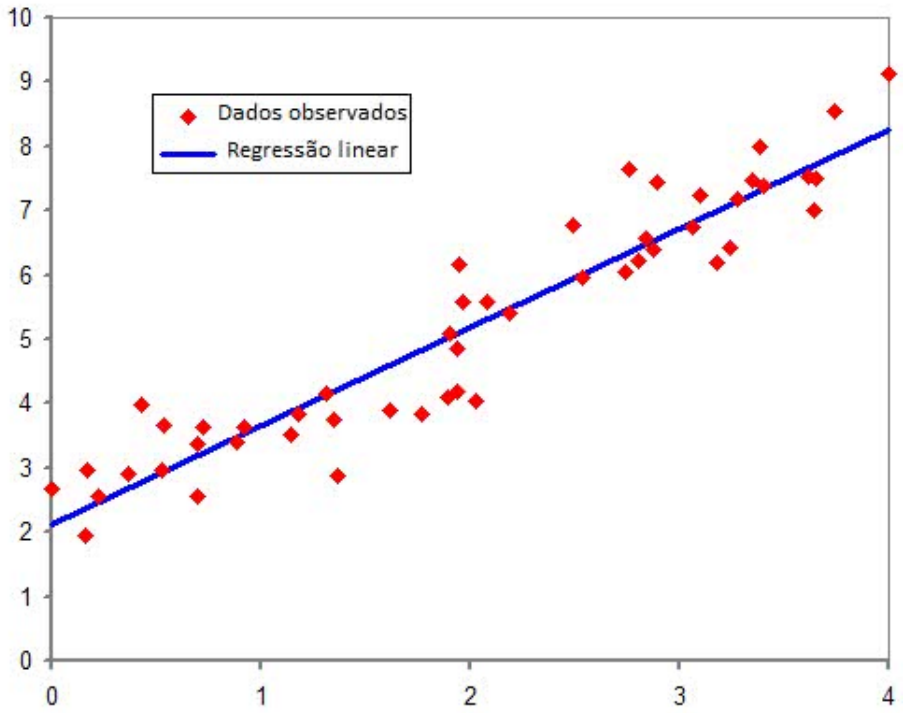
è um algorítimo de classificação supervisionado sendo uma versão mais simples de uma rede neural. Seu funcionamento baseia-se na entrada de valores que são classificados em muitas saídas possíveis baseado numa função linear combinando cada uma delas com um número de pesos. Perceptron são mais simples e mais rápidos e podem ser usados para treinamento contínuo (TAI, 2019).

4.3.1.8 Algoritmos de Regressão

Neste tipo de algoritmo o objetivo é fazer uma regressão com o intuito de prever um número. Este algoritmo verifica a relação entre duas variáveis utilizando pontos de dados para traçar uma linha fazendo os ajustes necessários para encontrar a melhor relação entre eles

assim demonstrado no gráfico 8. A regressão pode ser simples quando ela utiliza apenas uma variável independente e múltipla quando ela utiliza mais de uma variável independente (GOMES, 2019).

Figura 8 – Gráfico de regressão linear



Fonte: (GOMES, 2019)

4.3.2 Aprendizado não Supervisionado

De acordo com Tibco (s.d), neste tipo de método a máquina aprende a identificar novos padrões e detectar anomalias. Os dados que são utilizados em algoritmos de aprendizagem não supervisionados não são rotulados . A máquina aprende por conta própria a reconhecer padrões analisando sozinha características que mais se repetem por exemplo. Com o tempo ela consegue a fazer a separação por exemplo do que é um carro e um ônibus baseando-se no tamanho. Como não há uma aprendizagem inicial o processo tende a ser mais demorado (FREITAS, 2019).

4.3.3 O aprendizado reforçado

O aprendizado reforçado também conhecido como otimização encontra a melhor solução mesmo quando há restrições complexas. O aprendizado reforçado é caracterizado pelo uso da experiencia da máquina onde ela usa os erros e acertos para ir se adequando e aprimorando cada vez mais (FREITAS, 2019).

4.4 Computação em Nuvem

A computação em nuvem é uma revolução computacional que possui várias vantagens aos usuários. Entre elas está a disponibilização de um sistema computacional potente para qualquer usuário através da internet diminuindo a necessidade de o usuário ou a organização de gastar com uma estrutura de Hardware robusta. Como por exemplo no nosso experimento estaremos utilizando a computação em nuvem do Azure através do Azure Studio Machine Learning que faz o processamento de dados e treinamento do modelo e a criação do serviço web disponibilizado pelo Azure o que facilita a criação de todo o sistema poque garante a disponibilidade do serviço e a velocidade nos processos. As vantagens da utilização da computação em nuvem de acordo com a documentação do Azure (2022b) são os seguintes:

Custo : O uso de um serviço de computação em nuvem retira a necessidade de um custo para compra e implantação de Hardware e Software e também na capacitação de profissionais para usar o sistema ou mesmo a contração de pessoal de TI. Usa-se o poder computacional da provedora do serviço de cloud além pagar apenas pelos serviços que usar.

Velocidade : Os serviços de cloud são oferecidos com uma estrutura muito grande de Hardware e são disponibilizados sob demanda para garantir que os recursos computacionais estejam disponíveis em minutos ou mesmo em segundos.

Escala Global : O serviço de computação em nuvem fornecem uma quantidade de recursos de na medida certa e em qualquer localização geográfica com servidores lotados em várias partes do mundo.

Produtividade : Com o foco dos profissionais de TI voltados apenas nos resultados e no trabalho em si já que a preocupação de montagem de um sistema de Hardware e Software fica a cargo da computação em nuvem há com isso um aumento na produtividade das equipes.

Segurança : Com um grande número de servidores espalhados pelo mundo com políticas de segurança garante-se assim a disponibilização e proteção dos dados contra ameaças.

4.4.1 Tipos de Computação em Nuvem

Dependendo da estrutura de computação e da filosofia do trabalho a computação em nuvem terá características diferentes para cada uma delas. O tipo de implantação dos serviços em nuvem poderá ser: nuvem pública, privada e Híbrida:

Nuvem Pública : Os serviços de Hardware e Software são todos disponibilizados pela prestadora de serviço e com o papel de gerenciamento por ela. O sistema que usarei neste trabalho é uma nuvem pública do Microsoft Azure.

Nuvem Privada : Uma nuvem privada são recursos de computação em nuvem de software e Hardware exclusivo de uma única organização, geralmente esta estrutura está localizada

dentro da infraestrutura particular da empresa. Alguma empresas pagam para armazenar sua estrutura de nuvem privada mas dentro de provedores de uma empresa terceirizada especializada nestes serviços.

Nuvem Híbrida : è a combinação de estruturas de nuvem pública e privada com uma tecnologia que permite a troca de informações entre elas, o que permite uma flexibilidade e garantia na segurança e proteção de dados ao mesmo tempo que otimiza a estrutura de computação.

Figura 9 – Representação do serviço de Computação em Nuvem

Fonte (AZURE, 2022b)

4.5 Azure machine Learning Studio

Para auxiliar nos testes dos algoritmos de Aprendizado de máquina será utilizado o Azure Machine Learning Studio (AZURE, 2022a) que é um serviço em nuvem que possibilita a criação e implementação de modelos de aprendizado de máquina de uma maneira automática e rápida, utilizaremos este serviço para encontrar os modelos com melhores resultados e com isso implementar no nosso código os modelos com os melhores resultados a fim de comparar com o que foi testado por este serviço.

Uma das vantagens de utilizar o Azure ML é a facilidade que este serviço possui pois ele segue os padrões Microsofit de projetos em módulos usando um estilo de "arrastar e soltar". Na hora que é criado o experimento é inciado um pipeline onde cada parte desse pipeline são módulos contendo cada parte do construtor do modelo bastando selecionar e arrastar esses módulos para o pipeline e em seguida configurar cada módulo com as características e dados do seu treinamento específico. O Azure ML permite a criação e teste de modelos usando as linguagens R, Python e SQL, além de possuir em seu banco de dados Data sets encontrados em outros serviços de Aprendizado de Máquina, inclusive ele também possui uma versão do Dataset usado neste trabalho (DAVID, 2016).

4.6 Python

Linguagem de programação implementada no final de 1989 é a linguagem mais utilizada por cientistas de dados. MATOS lista em seu artigo algumas das razões para isso são descritas a seguir:

- Python tem umas das maiores comunidades de usuários, oferecendo um grande número de materiais e especialistas que podem auxiliar na resolução de algum problema ou objetivo usando a linguagem.
- Possui uma enorme variedade de bibliotecas de ciência de dados e a cada dia ela é incrementada.
- Fácil aprendizagem. Python possui uma linha de aprendizagem muito mais agradável que outras linguagens, possibilitando a quem nunca teve contato com programação uma facilidade maior de aprender a utilizar
- Escalabilidade. Python é muito rápido, principalmente em relação a outras linguagens de programação voltadas para a ciência de dados.

4.6.1 Bibliotecas Python

Um dos motivos da escolha do Python para a criação dos modelos entre outras características é a sua vasta biblioteca para a área de ciência de dados, entre elas está o Scikit_Learn uma biblioteca criada especialmente para a prática de machine Learning e desenvolvida sobre as bibliotecas NumPy, SciPy e Matplotilib (DIDATICATECH, 2022).

O Scikit-Learning é criado em módulos que nos disponibilizam ferramentas para atividade específicas dependendo de nossas necessidades, como pré-processamento de dados, ajuste de parâmetros e os algoritmos, são ferramentas simples e preparadas para análise preditiva de dados.

5 Aplicação de técnicas de IA em um Dataset de Doadores de sangue

Este trabalho foi separado em duas etapas, a primeira é a criação e avaliação de modelos de machine learning usando a ferramenta Azure ML Studio onde serão avaliados os modelos disponibilizados pela plataforma. Também será criado modelos usando a biblioteca Scikit-Learn do Python para avaliação dos algoritmos.

5.1 Coleta de dados

Nesta etapa foram pesquisados artigos e trabalhos acadêmicos que tinham o tema e problemas parecidos para que fossem usados como base para encontrar as técnicas e as base de dados mais usadas. Foram feitos levantamentos em plataformas renomadas para garantir a confiabilidade das fontes. Algumas das plataformas utilizadas foram : Google Acadêmico, Biblioteca Digital de teses da USP e o Portal de periódicos da Capes.

Foi realizado uma pesquisa em plataformas de dados e ferramentas para Machine Learning afim de encontrar Datasets mais atuais e completos sobre doação de sangue para ser usado no trabalho.

O Dataset usado no trabalho está disponível na plataforma online kaggle que é uma plataforma onde a comunidade de cientista de dados tem acesso a competições de Machine Learning e onde os estudantes e cientistas de dados do mundo inteiro podem publicar seus trabalhos e disponibilizar conjuntos de dados para serem usados pela comunidade acadêmica e científica (DAVID, 2020).

Este Dataset é o conjunto de dados mais usado em trabalhos e estudos que envolvem o tema de doadores de sangue e Machine Learning. Ele foi retirado do banco de dados de um Centro de Serviços de Transfusão de Sangue na cidade de Hsin-Chu em Taiwan no ano de 2008. São registros de 748 doadores escolhidos aleatoriamente. Os atributos do Dataset são os seguintes (NISHAT, 2019) :

Os atributos e valores de alguns exemplos estão mostrados na tabela 4.

1. R (Recência - meses desde a última doação),
2. F (Frequência - número total de doações),
3. M (Monetário (Quantidade) - total de sangue doado em centímetros cúbicos,
4. T(Tempo - meses desde a primeira doação)),
5. e uma variável binária que representa se ele / ela doou sangue em março de 2007 (1 significa doar sangue; 0 significa não doar sangue).

Tabela 4 – Amostras dos Atributos e valores do Dataset

Recência (meses)	Frequência (Dias)	Total Sangue doado (cm^3)	Tempo (meses)	Doou? (0/1)
2	50	12500	98	1
0	13	3250	28	1
1	16	4000	35	1
2	20	5000	45	1
1	24	6000	77	0
4	4	1000	4	0

Fonte: (NISHAT, 2019).

5.2 Treinamento e avaliação de algoritmos e criação de modelo usando Azure Machine Learning Studio

O Azure ML já disponibiliza em sua biblioteca um conjunto de algorítimos de Aprendizado de máquina de cada categoria. Os algorítimos de classificação que serão usados para testar com nosso Dataset serão os seguintes:

- Multiclass decision forest.
- Multiclass Decision Jungle.
- Multiclass Logistic Regression.
- Multiclass Neural Network.
- two-class bayes point machine.
- two-class boosted decision.
- tree two-class decision forest.
- two-class decision jungle.
- two-class locally deep support vector machine.
- two-class logistic regression.
- two-class neural network.
- two-class support vector machine.

As características e o funcionamento de cada um dos modelos pode ser visto na documentação da Microsoft do Azure Machine Learning Studio (MICROSOFT, 2022a). Neste artigo mostra o passo a passo para configurar os algorítimos os parâmetros de ajustes e os resultados esperados entre outras informações mais relevantes.

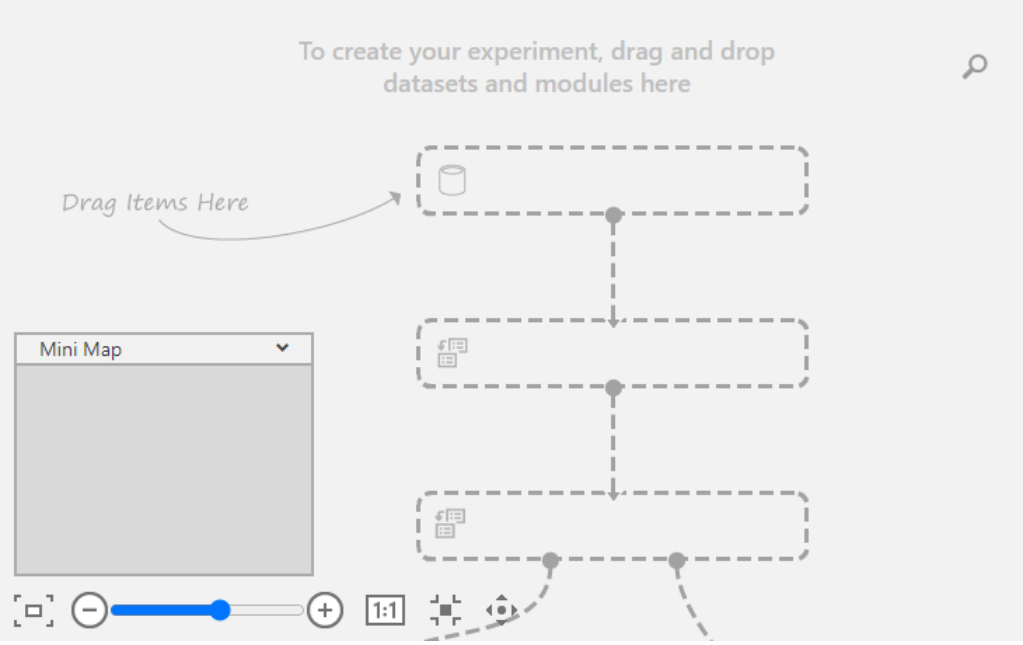
5.2.1 Conjunto de dados

Um Dataset é um conjunto de dados que nada mais é do que uma coleção de dados. Os dados são os valores presentes no Dataset. Cada elemento do Dataset corresponde a uma característica, as colunas são as variáveis e cada linha é um membro do conjunto de dados.

5.2.2 Seleção do Dataset no Azure ML

Após fazer o cadastro no site do Azure ML para criar o projeto basta iniciar o projeto que na plataforma é chamado de experiments. Após a criação ele irá mostrar de maneira bem didática um modelo em forma de diagrama com cada etapa e cada parte do experimento bastando selecionar o que você quer e arrastar para o centro da tela como mostrado na figura 10.

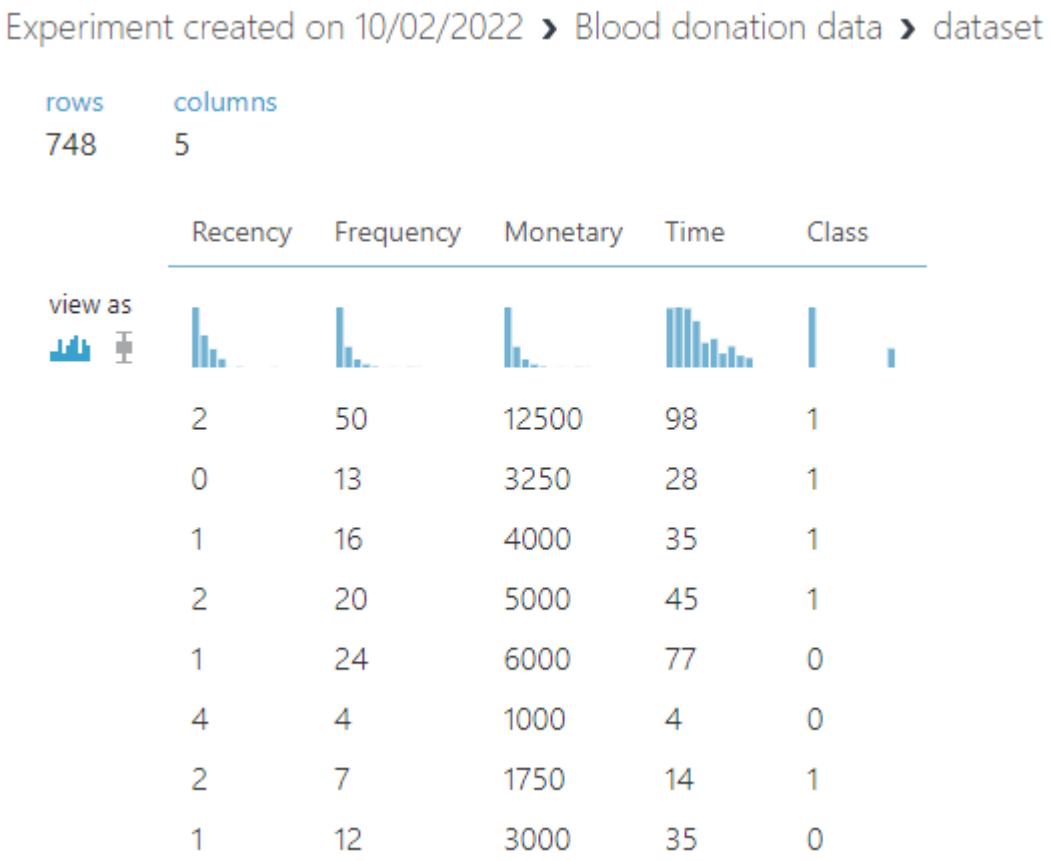
Figura 10 – Tela Inicial da criação do experimento no Azure ML Studio .



Fonte: (AZURE, 2022a)

Como foi dito o Azure ML possui alguns Datasets salvos em seu banco de dados e o Dataset utilizado no trabalho está presente nela bastando ir em Saved Datasets e buscar pelo Dataset Blood Donation Data. Se caso o Data Set não estivesse na plataforma bastava fazer o upload do Dataset para a plataforma. Depois de escolhido o Dataset é possível visualizar os atributos pela própia plataforma clicando com o botão direito e visualizar, o Dataset é visualizado conforme a figura 11.

Figura 11 – Visualização do Dataset no Azure ML Studio .



Fonte: (AZURE, 2022a)

5.2.3 Padronizando os dados no Azure ML

A segunda etapa é a seleção do módulo "Edit Metadata"que tem como objetivo a padronização dos dados do Dataset definindo quais atributos serão utilizados, quais devem ser descartados, podendo também nessa etapa fazer a renomeação das colunas substituindo os nomes originais de cada atributos para melhor entendimento (MICROSOFT, 2022b).

Nessa parte definimos que a coluna alvo será a coluna referente a informação da doação ou não de cada doador no dia selecionado na pesquisa que criou o Dataset, informaremos que nosso modelo será baseado nessa informação. Nessa etapa também será descartado a coluna referente ao ID de cada Doador pois essa informação não é importante para nosso experimento.

5.2.4 Separando as amostras de teste e de treinamento

O módulo "Split Data"vem logo a seguir e a função é a separação dos dados em partes de dados para fim de treinamento e teste. Para usar esse módulo basta arrastar o módulo para o pipeline ele se encontra na categoria "Data Transformation", "Sample and Split". Após selecionar e arrastar basta clicar nele e fazer as configurações necessárias. o primeiro atributo se refere a que tipo de divisão será usada, são elas:

separação por linhas Separação dos dados em duas partes. Escolhendo a porcentagem dessa divisão. Por padrão a divisão é 50/50.

expressão regular quando queremos testar uma determinada coluna por um determinado valor. Por exemplo se selecionar uma coluna de tamanhos e selecionamos a expressão "médio"os dados serão divididos em duas partes, uma parte com linhas que contenham a expressão médio na coluna altura e uma outra parte de dados onde não contenha essa expressão. Essa expressão pode ser uma palavra inteira ou também um cadeia de caracteres.

Expressão de linguagem podemos aplicar uma condição a uma coluna, como por exemplo a separação de produtos pela coluna data, onde serão separados os produtos pela data de venda, é um dos exemplos de uso (MICROSOFT, 2022c).

Nosso módulo neste item irá separar em dois os dados do Dataset onde serão criados a amostra para teste correspondendo a 80% para o treinamento dos modelos e 20% para teste. Bastando apenas selecionar no Azure o tipo de separação que no nosso caso é por linhas e a porcentagem da divisão colocando o valor de 0,8 nas propriedades, as outras opções não serão utilizadas.

5.2.5 Escolha dos modelos de classificação a serem testados

O Azure ML Studio possui em seu banco de dados os principais modelos de Aprendizado de máquina separados pelo tipo, são eles, classificação (classification),Detecção de anomalias (Anomaly detection), Análise de cluster (Clustering) e Regressão (Regression). Usaremos todos os algoritmos de classificação disponíveis no banco de dados para posteriormente os com os melhores resultados serão testados no projeto manual para comparação. Como o processo de criação é mais rápido do que no projeto manual esta etapa ajudará na escolha dos algoritmos a serem testados no projeto. Esses módulos estão presentes na categoria Machine Learninig/Initialize Model/Classification.

5.2.6 Resultados

A penúltima etapa do Azure ML Studio é a parte de resultados onde podemos visualizar os resultados do treinamento mostrados na figura 12. Ela descreve o resultado de alguns cálculos obtidos do teste como acurácia, precisão negativa e precisão positiva, score, falso positivo e falso negativo ou seja informações que ajudarão na comparação dos modelos para que sirva de auxílio na decisão final.

A página de resultados calculam as métricas a seguir:

- Accuracy: Calcula a qualidade do modelo de classificação como uma proporção de resultados verdadeiros do número de casos totais.Accuracy= (TP+FN)/(TP+TN+FP+FN).
- Precision: Proporção de resultados verdadeiros sobre o total de resultados positivos. Precisão = TP/(TP+FP).

Figura 12 – Tabela de cálculo do resultado da avaliação de algoritmo.

Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	0	0	0.000	0.780	0.000	1.000	0.000	0.780	1.000	0.000
(0.800,0.900]	0	0	0.000	0.780	0.000	1.000	0.000	0.780	1.000	0.000
(0.700,0.800]	0	0	0.000	0.780	0.000	1.000	0.000	0.780	1.000	0.000
(0.600,0.700]	0	0	0.000	0.780	0.000	1.000	0.000	0.780	1.000	0.000
(0.500,0.600]	1	0	0.007	0.787	0.059	1.000	0.030	0.785	1.000	0.000
(0.400,0.500]	2	0	0.020	0.800	0.167	1.000	0.091	0.796	1.000	0.000
(0.300,0.400]	14	19	0.240	0.767	0.493	0.472	0.515	0.860	0.838	0.066
(0.200,0.300]	11	52	0.660	0.493	0.424	0.283	0.848	0.902	0.393	0.381
(0.100,0.200]	5	45	0.993	0.227	0.363	0.221	1.000	1.000	0.009	0.745
(0.000,0.100]	0	1	1.000	0.220	0.361	0.220	1.000	1.000	0.000	0.754

Fonte: (AZURE, 2022a)

- Recall: Porcentagem das instâncias relevantes que foram recuperadas. Capacidade do método de detectar com sucesso resultados classificados como positivos. Recall= TP/(TP+FN).
- F1 Score: Média ponderada de Precisão e recall entre 0 e 1. Sendo o 1 o valor ideal.
- AUC: Curva plotada em um gráfico com os verdadeiros positivos no eixo Y e falsos positivos no eixo X. mede a qualidade das previsões dos modelos independente do limite de classificação.

True Positivos (TP) : são as instâncias em que o modelo previu corretamente uma classe positiva .

True Negatives (TN) : são as instâncias em que o modelo previu corretamente uma classe negativa .

Falsos positivos (FP) : são as instâncias em que o modelo previu incorretamente uma classe positiva . .

Falsos negativos (FN) : são as instâncias em que o modelo previu incorretamente uma classe negativa .

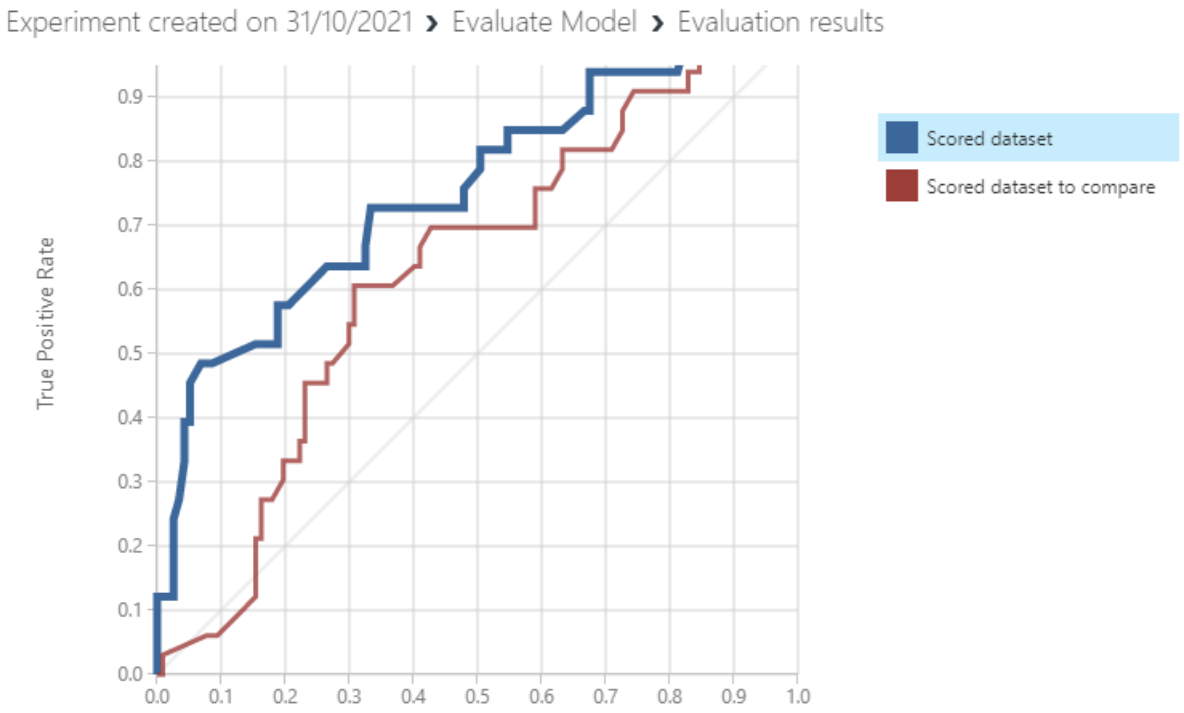
Como podemos treinar dois algoritmos ao mesmo tempo a tela de resultados nos mostra também um gráfico que projeta os resultados de dois modelos testados e faz uma comparação gráfica o que facilita bem na etapa de comparação como mostrado na figura 13.

Criaremos também um modelo de Aprendizado de máquina usando a biblioteca do Python chamada Scikit-Learn que foi desenvolvidas para análise preditiva de dados.

5.3 Criando modelos de Machine Learning usando Python e seu ecossistema

Nesta etapa é definido a linguagem escolhida para a criação de nosso projeto de aprendizagem de máquina. A linguagem foi escolhida através de pesquisas na rede mundial de computadores das linguagens para aprendizagem de máquina mais utilizadas e mais em alta

Figura 13 – Gráfico de resultados de avaliação do modelo de ML.



Fonte: (AZURE, 2022a)

no momento da pesquisa. Foi realizado uma pesquisa no meio acadêmico e no ramo industrial analisando projetos implantados. A linguagem mais utilizadas em todos esses casos foi o Python. A escolha do Python como citam West e Borges (2017) está no fato de sua criação ter como objetivo ser de fácil aprendizado por qualquer pessoa com qualquer nível de conhecimento de programação mesmo que seja iniciante, e de ser mais legível para as pessoas e não só para máquina, e também pelo fato do Python possuir uma biblioteca muito ampla que abrange todas as etapas do aprendizado de máquina.

5.4 Estudo e definição dos métodos de machine Learning

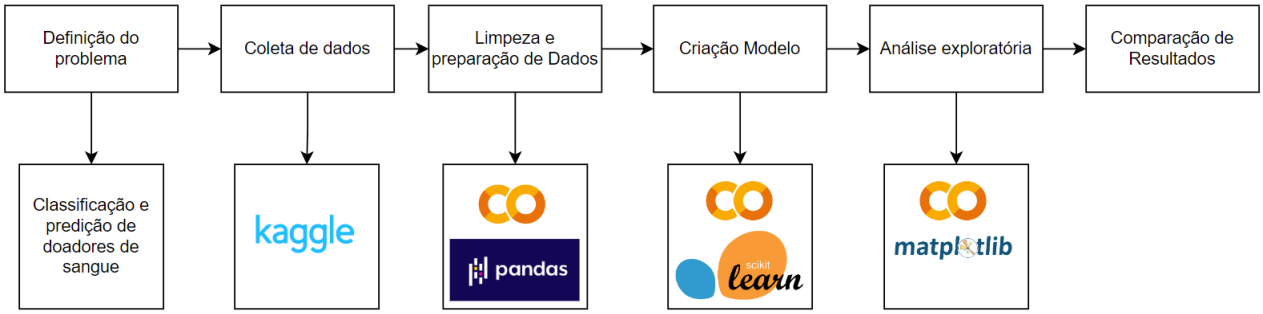
A escolha das ferramentas é uma etapa crucial para o sucesso do trabalho. Foi feito um estudo profundo sobre as ferramentas mais usadas no mercado. A linguagem Python foi a escolhida por ser a mais indicada para estes tipo de abordagem pelo número de bibliotecas e também por ter todas as ferramentas necessárias para o uso de aprendizagem de máquina, para manipulação de diversos tipos de dados e para desenvolvimento. Para o desenvolvimento de códigos e processamento foi escolhido o Jupyter Notebook que permite o uso de todo o poder do Python sem precisar instalar localmente as extensões e bibliotecas do Python e também faz todo o processamento do código.

Nesta unidade estaremos criando modelos de aprendizagem de máquina utilizando a biblioteca scikit-Learn e fazendo as análises usando funções da biblioteca Pandas e gerando gráficos utilizando o Matplotlib.

Usaremos como ferramentas o Google Colab, um produto da Google que permite criação e e execução de código Python diretamente no navegador sem a necessidade de instalação do Anaconda para rodar o Jupyter, tudo será instalado de modo on-line. AS vantagens de fazermos usando o Google Colab é o compartilhamento do projeto com mais pessoas e também a possibilidade de rodar o projeto em nuvem, fazendo com que o processamento seja feito em nuvem assim não sendo necessário uma estrutura robusta para o projeto.

Toda a metologia que será utilizada nesta etapa está representada na figura 14.

Figura 14 – Pipeline do projeto



fonte : O autor

5.4.1 Preparação do ambiente

Para a criação e salvamento dos projetos em Jupyter Notebook no Google Colab é preciso uma conta Gmail para que os dados do projeto sejam salvos no Google Drive.

Na plataforma Kaggle além de termos disponível o Dataset que usaremos temos também informações sobre o nosso Dataset para melhor compreender os dados que possamos trabalhar. Para a inclusão do Dataset no nosso Google Colab temos um ícone de pasta onde será o local onde poderemos fazer o upload do arquivo do Dataset.

5.4.2 Análise do conjunto de dados usando Pandas

Para fazer a análise de atributos do nosso Dataset usaremos a biblioteca Pandas, para isto basta importar a biblioteca no cabeçalho do nosso código. Com a importação da biblioteca Pandas poderemos fazer a importação do nosso Dataset com o comando "read_csv()"do Pandas passando como parâmetro o caminho do diretório onde fizemos o upload do nosso Dataset. A função "df.head()"nos retorna os primeiros cinco dados de nosso Dataset como a figura 15 mostra.

Figura 15 – Primeiros cinco do Dataset

	Recency (months)	Frequency (times)	Monetary (c.c. blood)	Time (months)	whether he/she donated blood in March 2007
0	2	50	12500	98	1
1	0	13	3250	28	1
2	1	16	4000	35	1
3	2	20	5000	45	1
4	1	24	6000	77	0

fonte : O autor

Para melhor entendimento dos nossos dados precisamos editar os dados . A primeira mudança é alteração dos nomes das colunas para melhor entendimento, para isso primeiro usamos a função "df.columns()"para imprimir o nome das variáveis, com essa informação em mão criamos um dicionário de nome colunas que altera cada nome da coluna. A figura 16 trás o Dataset com o nome das colunas modificados.

Figura 16 – Alteração do nome das colunas

```
col = df.columns
col
```

```
Index(['Recency (months)', 'Frequency (times)', 'Monetary (c.c. blood)',
       'Time (months)', 'whether he/she donated blood in March 2007'],
      dtype='object')
```

```
[23] colunas = {
        'Recency (months)' : 'Recência (meses)', 'Frequency (times)': 'Nº de doações',
        'Monetary (c.c. blood)': 'Total de sangue doado(ml)',
           'Time (months)' : 'Meses última doação',
           'whether he/she donated blood in March 2007': 'Doou ou não'
    }

    df = df.rename(columns = colunas)
    col = df.columns
    col
```

```
Index(['Recência (meses)', 'Nº de doações', 'Total de sangue doado(ml)',
       'Meses última doação', 'Doou ou não'],
      dtype='object')
```

fonte : O autor

Após esta etapa poderemos agora com o comando "describe()". A figura 17 traz as seguintes informações geradas pela função que mostra os dados estatísticos do nosso Dataset. Neves (2021) define assim os atributos da função: O campo count traz o tamanho das

amostras dos campos não nulos ou vazios.O tamanho do nosso Dataset é 748 linhas. Não existe nenhum campo com valor nulo ou vazio ou seja todos os campos estão preenchidos.

Figura 17 – Tabela de dados estatísticos do Dataset

	Recência (meses)	Nº de doações	Total de sangue doado(ml)	Meses última doação	Doou ou não
count	748.000000	748.000000	748.000000	748.000000	748.000000
mean	9.506684	5.514706	1378.676471	34.282086	0.237968
std	8.095396	5.839307	1459.826781	24.376714	0.426124
min	0.000000	1.000000	250.000000	2.000000	0.000000
25%	2.750000	2.000000	500.000000	16.000000	0.000000
50%	7.000000	4.000000	1000.000000	28.000000	0.000000
75%	14.000000	7.000000	1750.000000	50.000000	0.000000
max	74.000000	50.000000	12500.000000	98.000000	1.000000

fonte : O autor

O campo "mean"traz a média aritmética para cada atributo,o atributo "Recência"tem como média 9,5 que é a média do tempo de meses desde a última doação. O campo "Nº de doações"traz uma média de 5,5 vezes no total de vezes doadas. O campo "Total de sangue doado(ml)"tem como média 1370 ml de sangue doado por cada doador. O campo "Meses última doação"tem como média 34 meses desde a primeira doação. O campo "Doou ou não"possui como média 0,23 ou seja possui mais doadores que não doaram na data pesquisada específica.

O campo "std"corresponde ao desvio padrão, nos mostra o quanto nossos dados estão dispersos sobre a média do atributo específico. Os campos "min"e "max"mostram os valores máximos e mínimos de cada campo. Os campos "25%","50%"e "75%"nos mostram como nossos dados estão distribuídos em relação ao valor desse campo. Exemplo o campo "50%"do atributo Recência nos mostra que 50% das pessoas doaram pela última vez a 7 meses atrás ou menos e os outros 50% doaram a mais de 7 meses atrás. As próximas funções que analisaremos são as funções "df.info()"e "df.groupby('Doou ou não').count()".

A próxima função utilizada em nosso código é o "info()", essa função traz mais algumas informações importantes do nosso Dataset como o número de dados, número de colunas e o tipo de cada dado. De acordo com a tabela resultante nós temos 748 dados, sendo todos os 748 não nulos, e todos os dados são do tipo "int64"ou seja de 9,223,372,036,854,775,808 até +9,223,372,036,854,775,807.

A próxima função usaremos pra saber informações sobre nosso campo target "Doou ou não"que nos dará o valor exato de número de doadores do dia em questão. O número de doadores que doaram foram 178 contra 570 que não doaram. Os dados gerados pelas funções estão presentes na figura 18.

Figura 18 – Tabela de informações do Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 748 entries, 0 to 747
Data columns (total 5 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Recência (meses)           748 non-null    int64
 1   Nº de doações              748 non-null    int64
 2   Total de sangue doado(ml)  748 non-null    int64
 3   Meses última doação        748 non-null    int64
 4   Doou ou não                748 non-null    int64
dtypes: int64(5)
memory usage: 29.3 KB
```

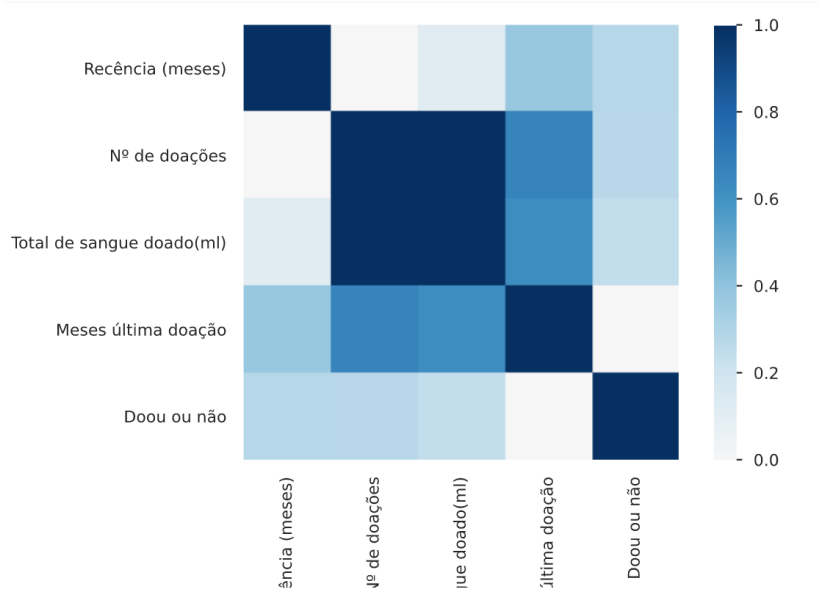
Doou ou não	Recência (meses)	Nº de doações	Total de sangue doado(ml)	Meses última doação
0	570	570	570	570
1	178	178	178	178

fonte : O autor

5.4.3 Análise de dados usando o Pandas-Profiling

O Pandas-Profiling é uma biblioteca do Pandas que faz uma análise completa dos dados sem a necessidade de usar muitas células para isso. Para usar a biblioteca no Google Colab é necessário a instalação no Google Colab pois ela é nativa. Com o comando "ProfileReport()"ele executa o comando e começa a criação do relatório executando a sumarização e a criação de um HTML com as informações. Com essa biblioteca temos algumas informações importantes como o número de zeros em cada atributos, a porcentagem de ocorrência em relação ao total. Essa função também nos mostra sem a necessidade de executar qualquer comando a correlação entre variáveis, três das nossas variáveis possuem uma alta correlação que são "Total de sangue doado(ml)", "Nº de doações"e "Meses desde a última doação"como nos mostra a matriz da figura 19.

Figura 19 – Matriz de correlação



fonte : O autor

5.4.4 Separação dos dados

Para iniciarmos o treinamento dos algoritmos deveremos agora separar os dados em dados de treino e de teste para que nosso Dataset não fique enviesado se treinássemos com todos os dados de nosso conjunto. Para fazer isso separamos nosso Dataset em dois onde separamos o nosso Target que é o dado se o indivíduo doou ou não. Isso é feito usando o comando "df.f=drop() passando como parâmetro a coluna "Doou ou não", isso salva o Dataset atual menos a coluna do nosso campo Target. A outra parte do Dataset com os dados restante serão salvos em outro conjunto de dados.

Após instalar e importar a biblioteca do Scikit - Learn para nosso Jupyter Notebook usaremos uma função disponibilizada dessa biblioteca a train_test_split que faz a separação de nossos dados em dados de treino e teste passando como parâmetros os dois conjunto de dados que acabamos de criar e o parâmetro test_size com o valor da porcentagem de separação dos casos de teste para nossos algoritmos que no nosso caso é de "test_size = 0.25"ou seja 25% dos nossos dados serão usados para treinar o modelo após o treinamento com os outros 75%.

Os próximos passos é a importação dos modelos da biblioteca do Scikit - Learn e o treinamento onde serão feitos os ajustes necessários para alcançarmos as melhores métricas.

5.4.5 Treinamento e métricas dos modelos

O primeiro passo para a criação dos modelos foi a pesquisa das abordagens utilizadas em pesquisas e trabalhos tanto em artigos como trabalhos acadêmicos encontrados na pesquisas bibliográficas e na documentação da biblioteca do Python Scikit-Learn. Os modelos, suas características e as métricas serão detalhadas nesta unidade.

6 Resultados dos experimentos

Os resultados dos treinamentos serão apresentados nesta seção. Em uma primeira execução usando o Azure ML Studio os modelos foram treinados usando a configuração padrão dos seus parâmetros e foram modificados os parâmetros realizando novos testes até alcançar a melhor performance para cada um deles. Após esta etapa foram criados os modelos usando o Python e seu ecossistema seguindo a etapa de criação e avaliação dos modelos com os atributos padrões e o refinamento de cada um deles por meio de novos testes.

6.1 Resultados do Treinamento e avaliação de algoritmos usando Azure Machine Learning Studio

O Azure ML já disponibiliza em sua biblioteca um conjunto de algorítimos de Aprendizado de máquina de cada categoria. Os algorítimos de classificação que serão usados para testar com nosso Dataset serão os seguintes:

- two-class bayes point machine.
- two-class boosted decision.
- tree two-class decision forest.
- two-class decision jungle.
- two-class locally deep support vector machine.
- two-class logistic regression.
- two-class neural network.
- two-class support vector machine.

As características e o funcionamento de cada um dos modelos pode ser visto na documentação da Microsoft do Azure Machine Learning Studio (MICROSOFT, 2022a). Neste artigo mostra o passo a passo para configurar os algorítimos os parâmetros de ajustes e os resultados esperados entre outras informações mais relevantes.

6.1.1 Escolha do Dataset no Azure Studio ML e preparação dos dados

Foi escolhido e posicionado o módulo com o DataSet "Blood Donation data"dos doadores de sangue que estamos usando neste trabalho, logo após foi escolhido o módulo "Edit Metadata"que permite editar os metadados modificando as colunas, é através desse módulo que podemos alterar o valor e os tipos de dados do Dataset. É neste módulo onde mudaremos o nome das colunas para melhor entendimento dos valores que ela contém e também é neste

módulo que selecionaremos a coluna classe que é a coluna que queremos categorizar ou prever que é a coluna com nome original "Class"que alteramos para "Potencial Doador". Após selecionamos o módulo de separação de dados para treino e testes o módulo "Split Data".

O módulo com o nome de "Split Data"é necessário para separar dados quando queremos uma parte de dados para treino e outra para teste. Os três seletores do primeiro atributo do módulo são os seguintes:

No nosso caso serão separados por linhas e na proporção 75% para treino e 25% para testes. A próxima etapa é a escolha e configuração dos módulos com os algorítimos para treino. Serão treinados todos os algorítimos de classificação presentes na biblioteca do Azure ML Studio com os valores dos atributos de cada algorítimo sendo modificados para encontrar os melhores resultados. A configuração de cada módulo nos melhores casos serão descritos na próxima sessão.

6.1.2 Two-Class Boosted Decision Tree

Foi adicionado os módulos de cada algorítimo. Primeiro foi feito o teste com os parâmetros com os valores padrões e depois foram modificados os valores dos mesmo para conseguir o melhor resultado possível neste experimento. As configurações que obtiveram os melhores resultados tanto de quantificação e performance foram:

Tabela 5 – Parâmetros definidos para o Two-Class Boosted Decision Tree

Parâmetro	Valor
Modo de treinamento:	Parâmetro único
Número máximo de folhas por árvore:	200
Número mínimo de amostras por nó folha:	100
Taxa de Aprendizagem:	0.1
Número de árvores construídas:	100

Fonte: O Autor.

6.1.3 Two-Class Neural Network

Depois de adicionado o módulo. Foi configurado cada um com as seguintes atributos no melhor resultado possível encontrado:

Tabela 6 – Parâmetros definidos para o Two-Class Neural Network

Parâmetro	Valor
Modo de treinamento:	Intervalo de parâmetros
Função de perda:	CrossEntropy
Especificação de camada oculta:	Caso totalmente conectado
Número iterações:	1000
Taxa de aprendizagem:	0.1
Valor inicial dos pesos de aprendizagem:	0.1
Tipo de Normalizador:	Normalizador Gaussiano

Fonte: O Autor.

6.1.4 Two-Class Bayes Point Machine

Após alguns testes a melhor configuração para esse algoritmo foi:

Tabela 7 – Parâmetros definidos para Two-Class Bayes Point Machine

Parâmetro	Valor
Número de iterações de treinamento:	100
Include bias:	não
Permitir valores desconhecidos em features categóricas:	não

Fonte: O Autor.

6.1.5 Two-class locally deep support vector machine

Alterando algumas métricas nos treinamentos a configuração com o melhor resultado foi o seguinte:

Tabela 8 – Parâmetros definidos para o Two-class locally deep support vector machine

Parâmetro	Valor
Modo de treinamento:	Parâmetro único
Profundidade da árvore:	3
valor de lambda:	0.1
Lambda Teta:	0.01
Lambda Theta Prime:	0.1
Número de iterações:	1500
Tipo de Normalizador:	Normalizador Gaussiano

Fonte: O Autor.

6.1.6 Two-Class Decision Forest

Parâmetros com melhor desempenho e resultado do módulo do Decision Forest:

Tabela 9 – Parâmetros definidos para Two-Class Decision Forest

Parâmetro	Valor
Modo de treinamento:	Parâmetro único
Modo de Re-amostragem:	Acondicionamento
Número de árvores de decisão:	24
Profundidade máxima:	64
Número de divisões aleatórias:	256

Fonte: O Autor.

6.1.7 Two-Class Averaged Perceptron

Parâmetros com melhor desempenho e resultado do algoritmo Two-Class Averaged Perecptron:

Tabela 10 – Parâmetros definidos para Two-Class Averaged Perceptron

Parâmetro	Valor
Modo de Treinamento:	Parâmetro único
Taxa de aprendizagem:	0.1
Número máximo de iterações:	100
Tamanho do batch:	256
Expoente de Decaimento da Taxa de Aprendizagem:	0.5
Peso médio:	0.5
Tolerância:	1E-05

Fonte: O Autor.

6.1.8 Two-class Logistic Regression

Parâmetros com melhor desempenho e resultado do algoritmo Two-class Logistic Regression:

Tabela 11 – Parâmetros definidos para Two-class Logistic Regression

Parâmetro	Valor
Modo de Treinamento:	Intervalo de parâmetro
tolerância de otimização:	1.58489880198727E-06
Peso de regularização:	0.00288402035984128
Peso de regularização:	0.10009
Tamanho da memória:	11

Fonte: O Autor.

6.1.9 Two-Class Support Vector Machine

Parâmetros com melhor desempenho e resultado do algoritmo Two-Class Support Vector Machine:

Tabela 12 – Parâmetros definidos para Two-Class Support Vector Machine

Parâmetro	Valor
Modo de Treinamento:	Parâmetro único.
Lambda:	0.01
Normalizar features:	habilitado

Fonte: O Autor.

6.1.10 Two-Class Decision Jungle

Parâmetros com melhor desempenho e resultado do módulo do Decision Jungle:

Tabela 13 – Parâmetros definidos para Two-Class Decision Jungle

Parâmetro	Valor
Contagem de etapas de otimização:	2048
Modo de treinador:	intervalo de parâmetros
Número de gráficos de decisão:	103 a 922
Contagem de elementos do conjunto:	8
Profundidade máxima:	32
Contagem de classes:	2
Largura máxima :	128

Fonte: O Autor.

6.1.11 Resultados

6.1.11.1 Two-Class Boosted Decision Tree

O algoritmo Two-Class Boosted Decision Tree ficou com os seguintes resultados:

Tabela 14 – Resultados Two-Class Boosted Decision Tree

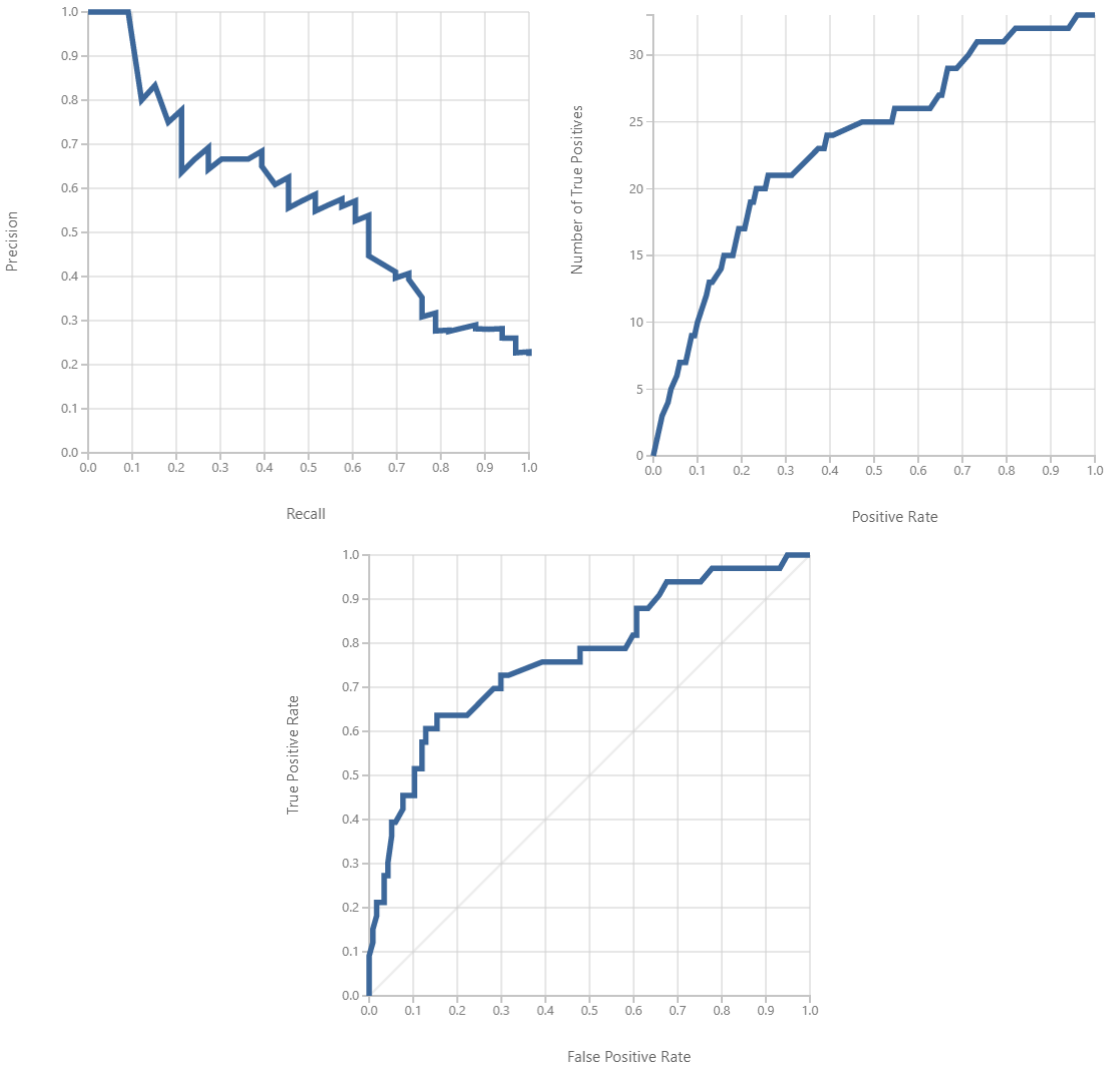
Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
13	20	7	110
Accuracy	Precision	Recall	F1 Score
0.820	0.650	0.394	0.491

Fonte: O autor

Figura 20 – Tela de Resultados do Two-Class Boosted Decision Tree no Azure Studio.

Fonte: (AZURE, 2022a)

Figura 21 – Gráficos das métricas do algorítimo Two-class Boosted Decision Tree



Fonte: (AZURE, 2022a)

6.1.11.2 Two-Class Bayes Point Machine

O algoritmo Two-Class Bayes Point Machine ficou com os seguintes resultados:

Tabela 15 – Resultados Two-Class Bayes Point Machine

Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
5	28	3	114
Accuracy	Precision	Recall	F1 Score
0.793	0.625	0.152	0.244

Fonte: O autor

Figura 22 – Tela de Resultados do Two-Class Bayes Point Machine no Azure Studio.

Experiment created on 05/04/2022 › Evaluate Model › Evaluation results

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
5	28	0.793	0.625	0.5	0.736

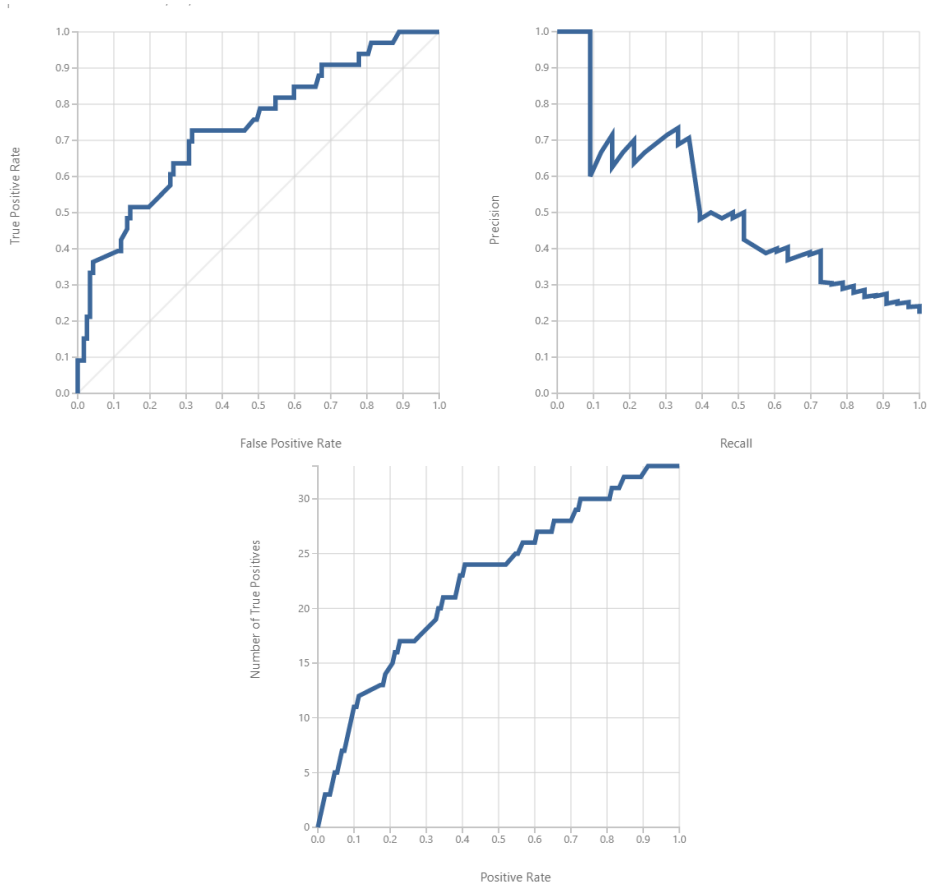
False Positive	True Negative	Recall	F1 Score
3	114	0.152	0.244

Positive Label	Negative Label
1	0

Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	0	0	0.000	0.780	0.000	1.000	0.000	0.780	1.000	0.000
(0.800,0.900]	0	0	0.000	0.780	0.000	1.000	0.000	0.780	1.000	0.000
(0.700,0.800]	0	0	0.000	0.780	0.000	1.000	0.000	0.780	1.000	0.000
(0.600,0.700]	2	0	0.013	0.793	0.114	1.000	0.061	0.791	1.000	0.000
(0.500,0.600]	3	3	0.053	0.793	0.244	0.625	0.152	0.803	0.974	0.003
(0.400,0.500]	12	18	0.253	0.753	0.479	0.447	0.515	0.857	0.821	0.066
(0.300,0.400]	6	15	0.393	0.693	0.500	0.390	0.697	0.890	0.692	0.140
(0.200,0.300]	1	18	0.520	0.580	0.432	0.308	0.727	0.875	0.538	0.251
(0.100,0.200]	6	34	0.787	0.393	0.397	0.254	0.909	0.906	0.248	0.495
(0.000,0.100]	3	29	1.000	0.220	0.361	0.220	1.000	1.000	0.000	0.736

Fonte: (AZURE, 2022a)

Figura 23 – Gráficos das métricas do algorítimo Two-class Bayes Point Machine



Fonte: (AZURE, 2022a)

6.1.11.3 Two-Class Neural Network

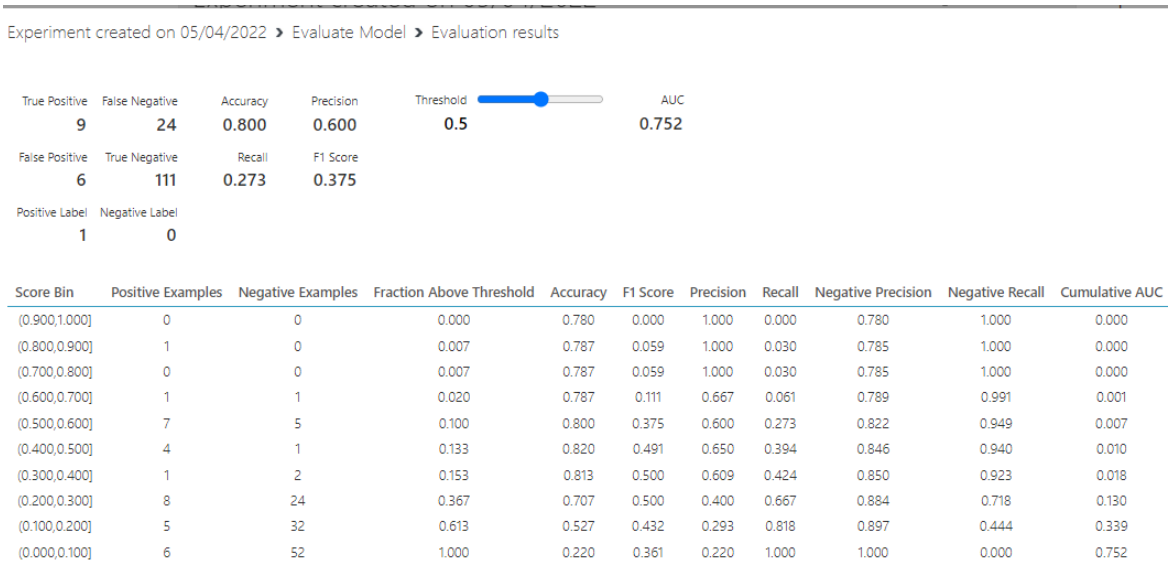
Resultados do treinamento do módulo do algorítmo do Two-Class Neural Network:

Tabela 16 – Resultados Two-Class Neural Network

Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
9	24	6	111
Accuracy	Precision	Recall	F1 Score
0.800	0.600	0.273	0.375

Fonte: O autor

Figura 24 – Tela de Resultados do Two-Class Neural Network no Azure Studio.



Experiment created on 05/04/2022 › Evaluate Model › Evaluation results

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
9	24	0.800	0.600	0.5	0.752

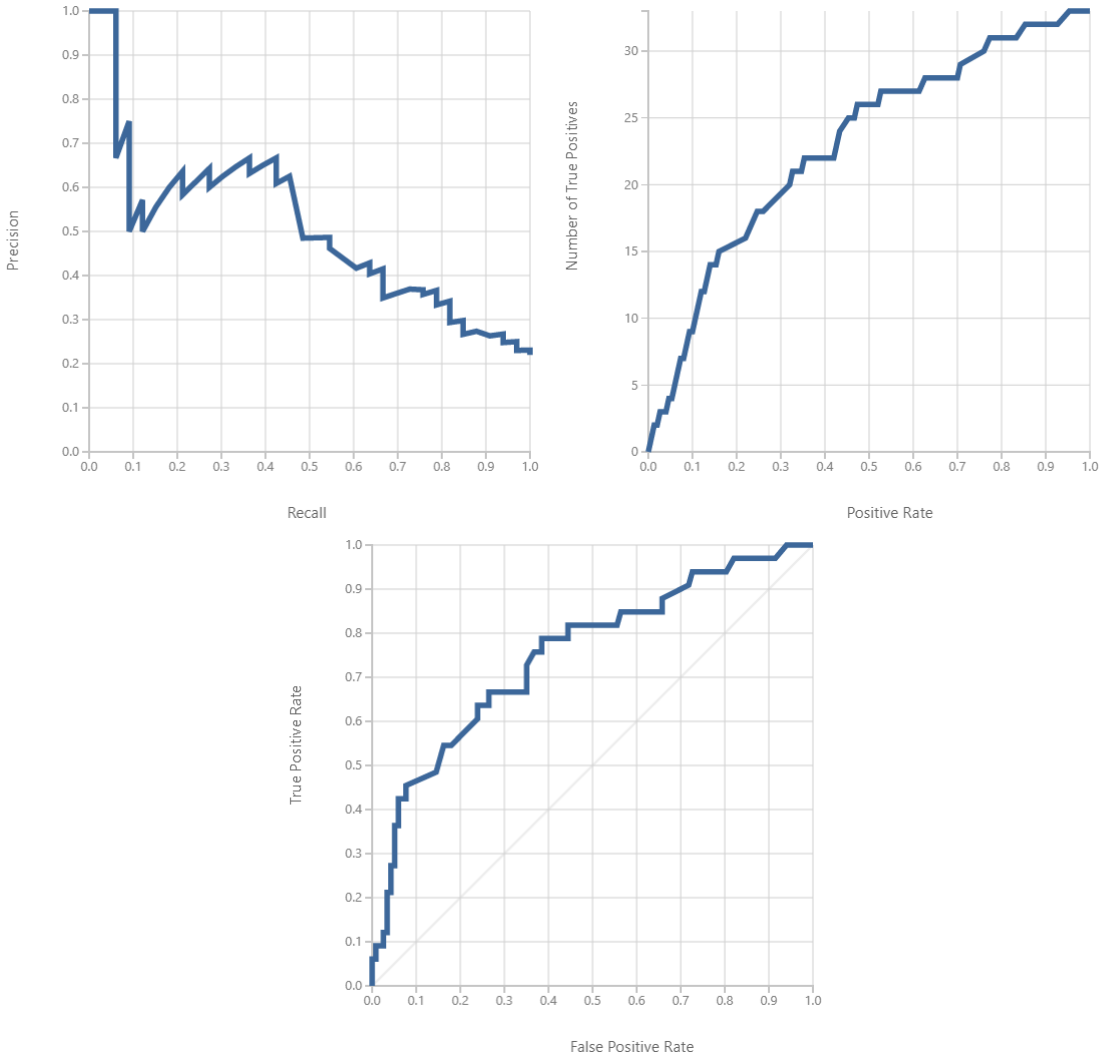
False Positive	True Negative	Recall	F1 Score
6	111	0.273	0.375

Positive Label	Negative Label
1	0

Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	0	0	0.000	0.780	0.000	1.000	0.000	0.780	1.000	0.000
(0.800,0.900]	1	0	0.007	0.787	0.059	1.000	0.030	0.785	1.000	0.000
(0.700,0.800]	0	0	0.007	0.787	0.059	1.000	0.030	0.785	1.000	0.000
(0.600,0.700]	1	1	0.020	0.787	0.111	0.667	0.061	0.789	0.991	0.001
(0.500,0.600]	7	5	0.100	0.800	0.375	0.600	0.273	0.822	0.949	0.007
(0.400,0.500]	4	1	0.133	0.820	0.491	0.650	0.394	0.846	0.940	0.010
(0.300,0.400]	1	2	0.153	0.813	0.500	0.609	0.424	0.850	0.923	0.018
(0.200,0.300]	8	24	0.367	0.707	0.500	0.400	0.667	0.884	0.718	0.130
(0.100,0.200]	5	32	0.613	0.527	0.432	0.293	0.818	0.897	0.444	0.339
(0.000,0.100]	6	52	1.000	0.220	0.361	0.220	1.000	1.000	0.000	0.752

Fonte: (AZURE, 2022a)

Figura 25 – Gráficos das métricas do algorítimo Two-class Neural Network



Fonte: (AZURE, 2022a)

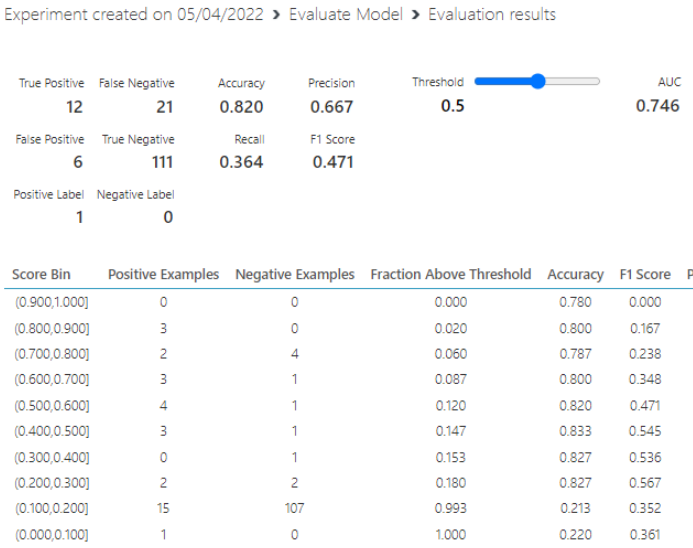
6.1.11.4 Two-Class Locally Deep Support Vector Machine

Tabela 17 – Resultados Two-Class Locally Deep Support Vector Machine

Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
12	21	5	111
Accuracy	Precision	Recall	F1 Score
0.820	0.667	0.364	0.471

Fonte: O autor

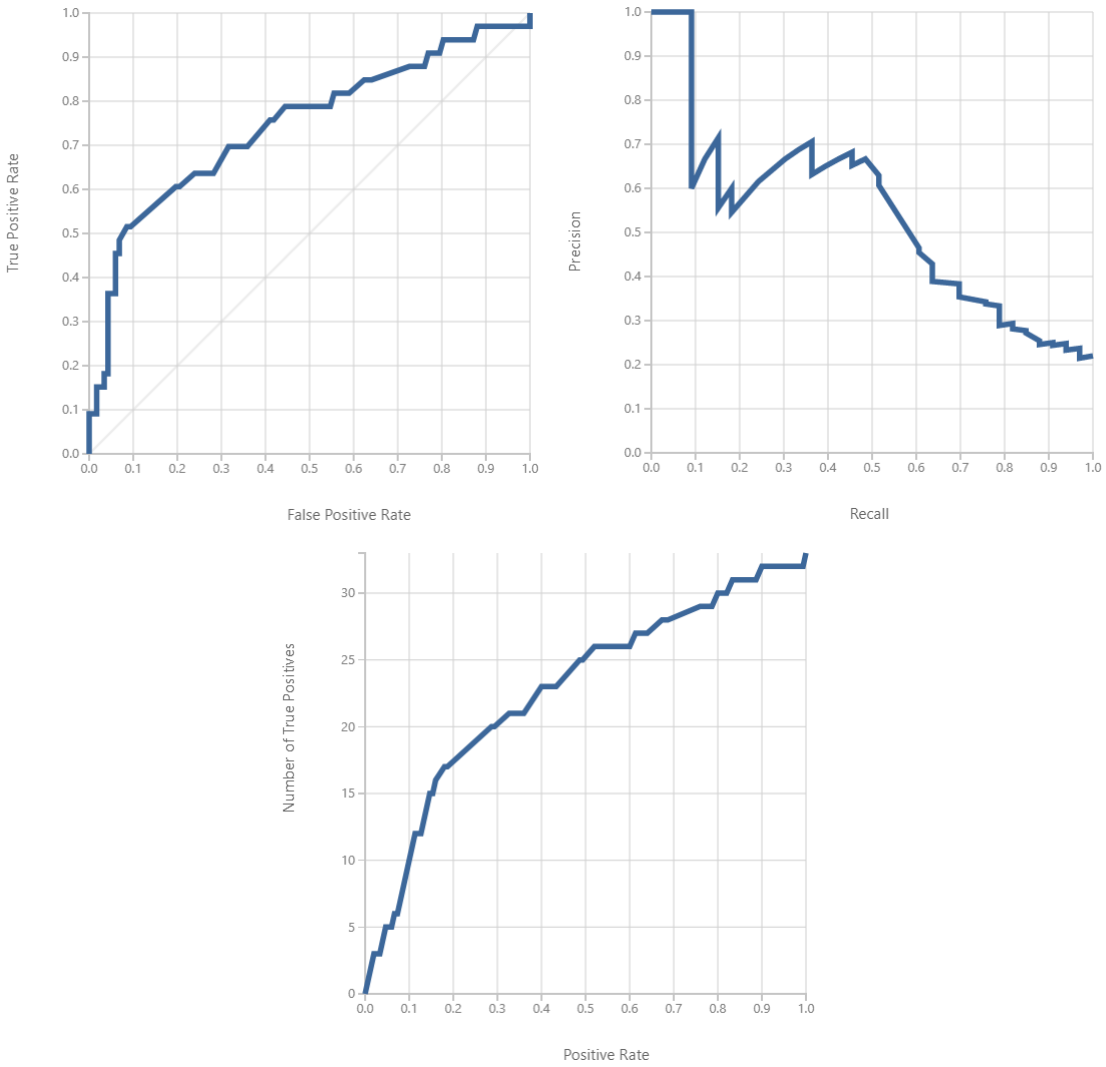
Figura 26 – Tela de Resultados do Locally DSVM no Azure Studio.



Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	0	0	0.000	0.780	0.000	1.000	0.000	0.780	1.000	0.000
(0.800,0.900]	3	0	0.020	0.800	0.167	1.000	0.091	0.796	1.000	0.000
(0.700,0.800]	2	4	0.060	0.787	0.238	0.556	0.152	0.801	0.966	0.004
(0.600,0.700]	3	1	0.087	0.800	0.348	0.615	0.242	0.818	0.957	0.006
(0.500,0.600]	4	1	0.120	0.820	0.471	0.667	0.364	0.841	0.949	0.009
(0.400,0.500]	3	1	0.147	0.833	0.545	0.682	0.455	0.859	0.940	0.012
(0.300,0.400]	0	1	0.153	0.827	0.536	0.652	0.455	0.858	0.932	0.016
(0.200,0.300]	2	2	0.180	0.827	0.567	0.630	0.515	0.870	0.915	0.024
(0.100,0.200]	15	107	0.993	0.213	0.352	0.215	0.970	0.000	0.000	0.746
(0.000,0.100]	1	0	1.000	0.220	0.361	0.220	1.000	1.000	0.000	0.746

Fonte: (AZURE, 2022a)

Figura 27 – Gráficos das métricas do algorítimo Two-class Locally DSVM



Fonte: (AZURE, 2022a)

6.1.11.5 Two-class Decision Jungle

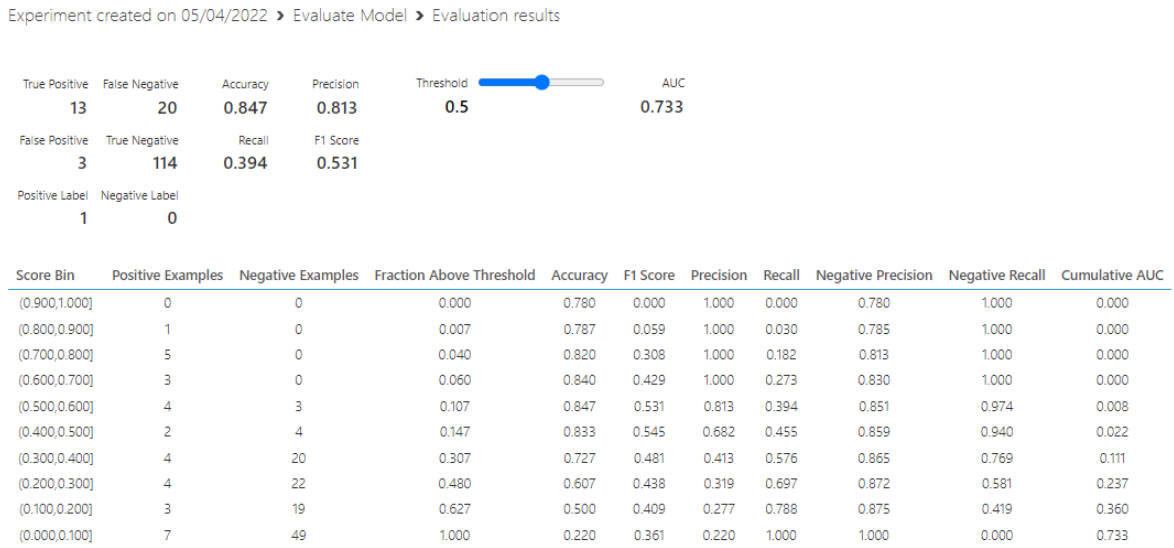
Resultados do treinamento do algorítimo Two-class Decision Jungle:

Tabela 18 – Resultados Two-class Decision Jungle

Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
13	20	3	114
Accuracy	Precision	Recall	F1 Score
0.847	0.813	0.394	0.531

Fonte: O autor

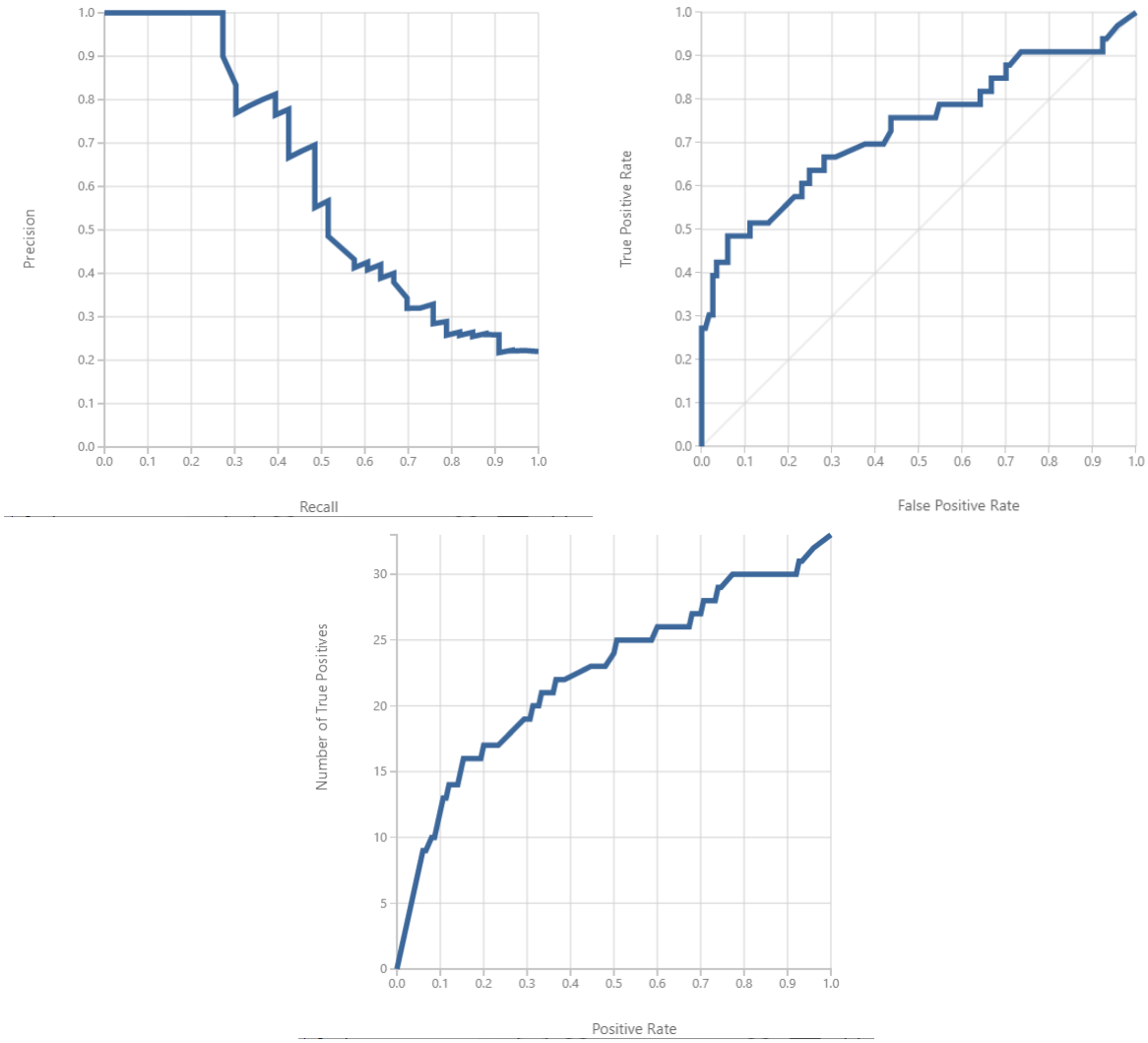
Figura 28 – Tela de Resultados do Two-class Decision Jungle no Azure Studio.



Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	0	0	0.000	0.780	0.000	1.000	0.000	0.780	1.000	0.000
(0.800,0.900]	1	0	0.007	0.787	0.059	1.000	0.030	0.785	1.000	0.000
(0.700,0.800]	5	0	0.040	0.820	0.308	1.000	0.182	0.813	1.000	0.000
(0.600,0.700]	3	0	0.060	0.840	0.429	1.000	0.273	0.830	1.000	0.000
(0.500,0.600]	4	3	0.107	0.847	0.531	0.813	0.394	0.851	0.974	0.008
(0.400,0.500]	2	4	0.147	0.833	0.545	0.682	0.455	0.859	0.940	0.022
(0.300,0.400]	4	20	0.307	0.727	0.481	0.413	0.576	0.865	0.769	0.111
(0.200,0.300]	4	22	0.480	0.607	0.438	0.319	0.697	0.872	0.581	0.237
(0.100,0.200]	3	19	0.627	0.500	0.409	0.277	0.788	0.875	0.419	0.360
(0.000,0.100]	7	49	1.000	0.220	0.361	0.220	1.000	1.000	0.000	0.733

Fonte: (AZURE, 2022a)

Figura 29 – Gráficos das métricas do algorítimo Two-class Decision Jungle



Fonte: (AZURE, 2022a)

6.1.11.6 Two-clas Two-class Decision Forest

Resultados do treinamento do Two-class Decision Forest:

Tabela 19 – Resultados Two-class Decision Forest

Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
17	20	12	105
Accuracy	Precision	Recall	F1 Score
0.787	0.520	0.394	0.448

Fonte: O autor

Figura 30 – Tela de Resultados do Two-class Decision Forest no Azure Studio.

Experiment created on 05/04/2022 › Evaluate Model › Evaluation results

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
13	20	0.787	0.520	0.5	0.704

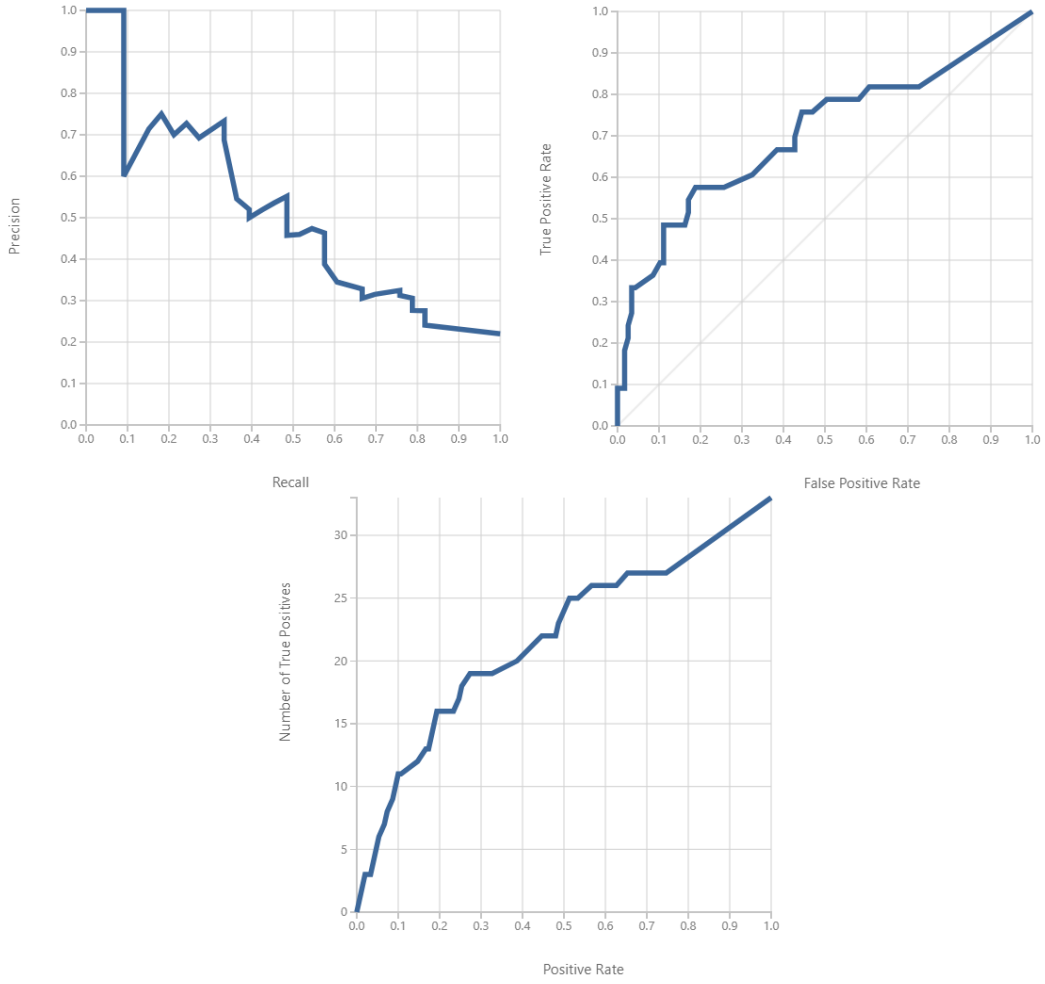
False Positive	True Negative	Recall	F1 Score
12	105	0.394	0.448

Positive Label	Negative Label
1	0

Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	2	0	0.013	0.793	0.114	1.000	0.061	0.791	1.000	0.000
(0.800,0.900]	1	0	0.020	0.800	0.167	1.000	0.091	0.796	1.000	0.000
(0.700,0.800]	5	3	0.073	0.813	0.364	0.727	0.242	0.820	0.974	0.003
(0.600,0.700]	3	2	0.107	0.820	0.449	0.688	0.333	0.836	0.957	0.008
(0.500,0.600]	2	8	0.173	0.780	0.441	0.500	0.394	0.839	0.889	0.033
(0.400,0.500]	3	0	0.193	0.800	0.516	0.552	0.485	0.860	0.889	0.033
(0.300,0.400]	1	7	0.247	0.760	0.486	0.459	0.515	0.858	0.829	0.062
(0.200,0.300]	5	30	0.480	0.593	0.419	0.306	0.667	0.859	0.573	0.218
(0.100,0.200]	4	9	0.567	0.560	0.441	0.306	0.788	0.892	0.496	0.276
(0.000,0.100]	7	58	1.000	0.220	0.361	0.220	1.000	1.000	0.000	0.704

Fonte: (AZURE, 2022a)

Figura 31 – Gráficos das métricas do algoritmo Two-Class Decision Forest



Fonte: (AZURE, 2022a)

6.1.11.7 Two-class Averaged Perceptron

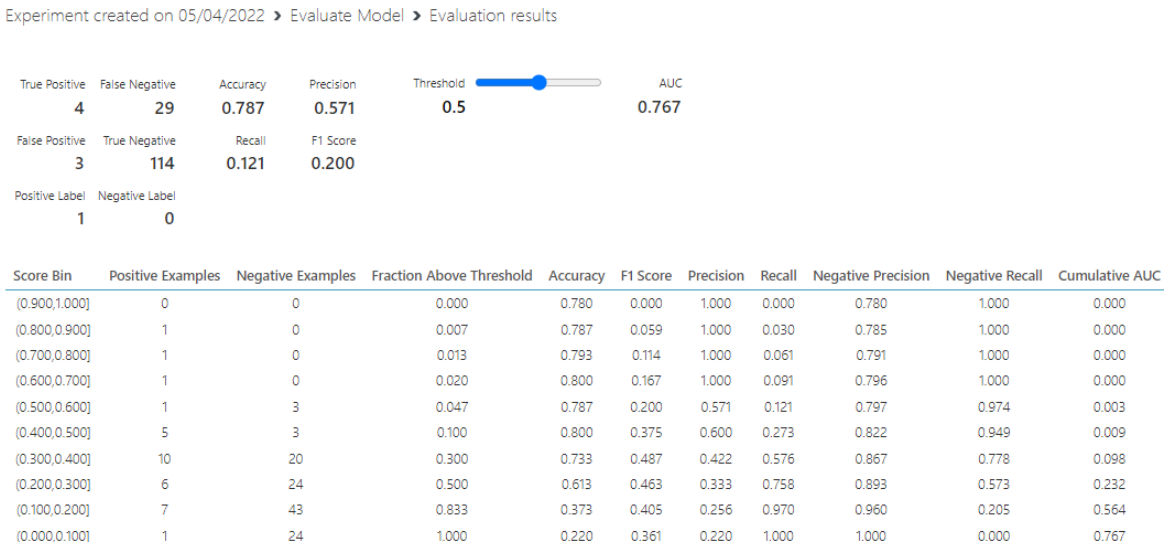
Resultados do treinamento do algorítimo Two-class Averaged Perceptron :

Tabela 20 – Resultados Two-class Averaged Perceptron

Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
4	29	3	114
Accuracy	Precision	Recall	F1 Score
0.787	0.571	0.121	0.200

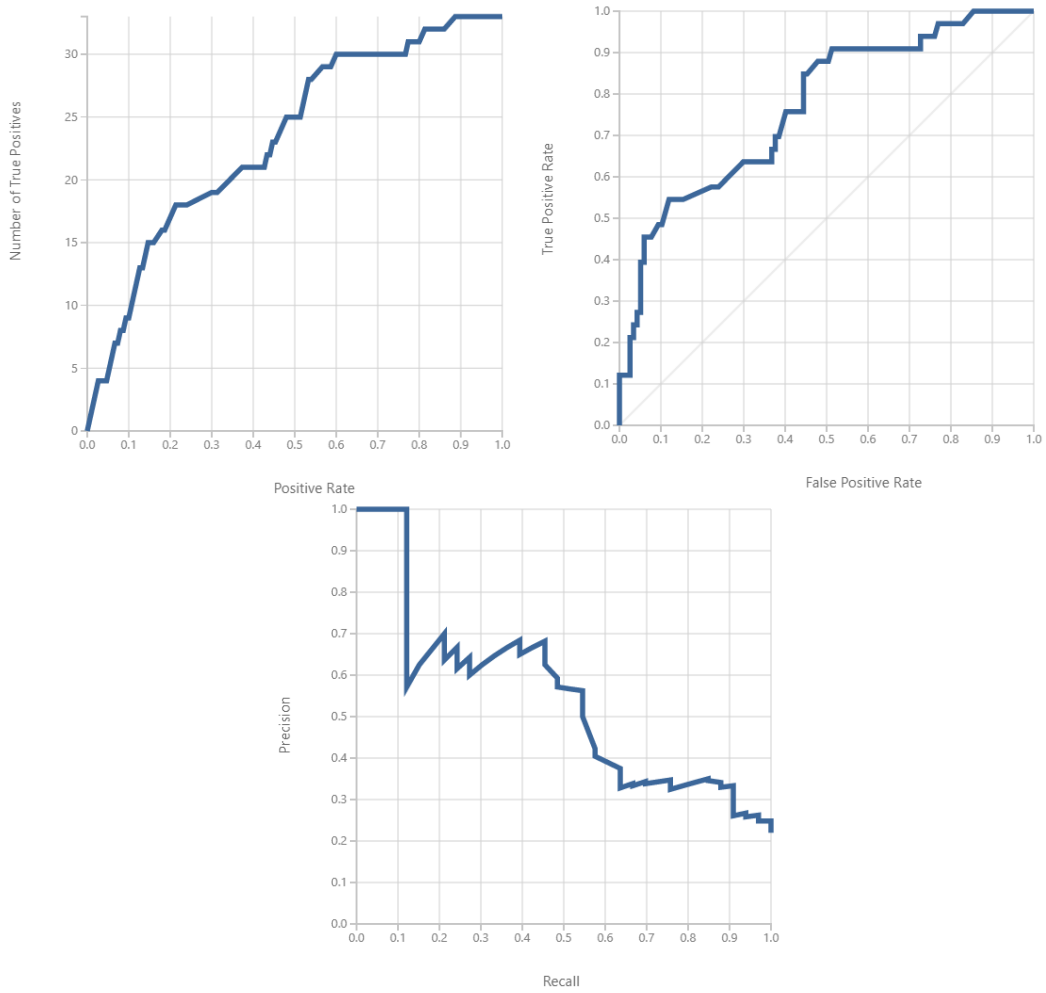
Fonte: O autor

Figura 32 – Tela de Resultados do Two-class Averaged Perceptron no Azure Studio.



Fonte: (AZURE, 2022a)

Figura 33 – Gráficos das métricas do algoritmo Two-class Averaged Perceptron



Fonte: (AZURE, 2022a)

6.1.11.8 Two-class Logistic Regression

Resultados de treinamento do Two-class Logistic Regression:

Tabela 21 – Resultados Two-class Logistic Regression

Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
4	29	2	115
Accuracy	Precision	Recall	F1 Score
0.793	0.667	0.121	0.205

Fonte: O autor

Figura 34 – Tela de Resultados do Two-class Logistic Regression Perceptron no Azure Studio.

Experiment created on 05/04/2022 › Evaluate Model › Evaluation results

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
4	29	0.793	0.667	0.5	0.751

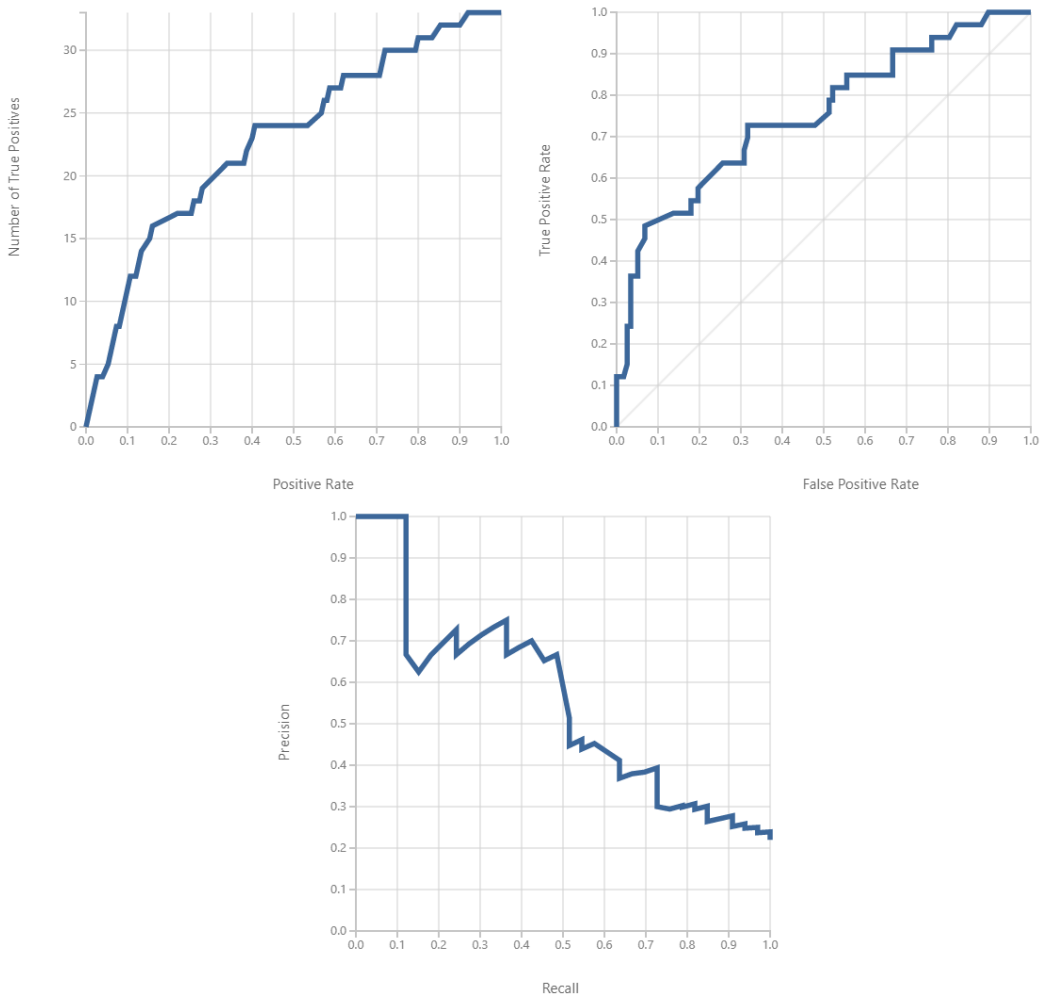
False Positive	True Negative	Recall	F1 Score
2	115	0.121	0.205

Positive Label	Negative Label
1	0

Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	0	0	0.000	0.780	0.000	1.000	0.000	0.780	1.000	0.000
(0.800,0.900]	0	0	0.000	0.780	0.000	1.000	0.000	0.780	1.000	0.000
(0.700,0.800]	1	0	0.007	0.787	0.059	1.000	0.030	0.785	1.000	0.000
(0.600,0.700]	1	0	0.013	0.793	0.114	1.000	0.061	0.791	1.000	0.000
(0.500,0.600]	2	2	0.040	0.793	0.205	0.667	0.121	0.799	0.983	0.002
(0.400,0.500]	8	3	0.113	0.827	0.480	0.706	0.364	0.842	0.957	0.009
(0.300,0.400]	9	28	0.360	0.700	0.483	0.389	0.636	0.875	0.718	0.137
(0.200,0.300]	3	17	0.493	0.607	0.449	0.324	0.727	0.882	0.573	0.240
(0.100,0.200]	7	40	0.807	0.387	0.403	0.256	0.939	0.931	0.231	0.525
(0.000,0.100]	2	27	1.000	0.220	0.361	0.220	1.000	1.000	0.000	0.751

Fonte: (AZURE, 2022a)

Figura 35 – Gráficos das métricas do algoritmo Two-class Logistic Regression



Fonte: (AZURE, 2022a)

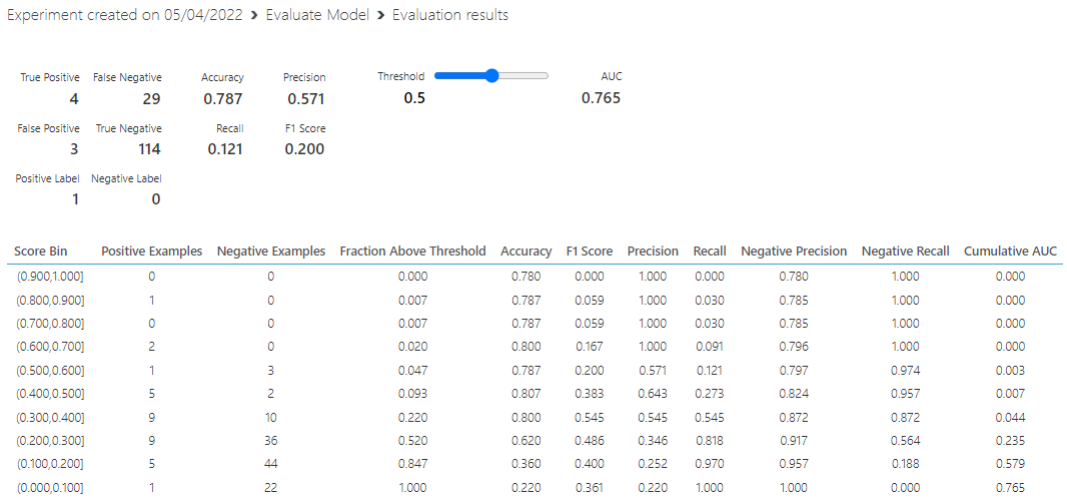
6.1.11.9 Two-class Support Vector Machine

Tabela 22 – Resultados Two-class Support Vector Machine

Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
4	29	3	114
Accuracy	Precision	Recall	F1 Score
0.787	0.571	0.121	0.200

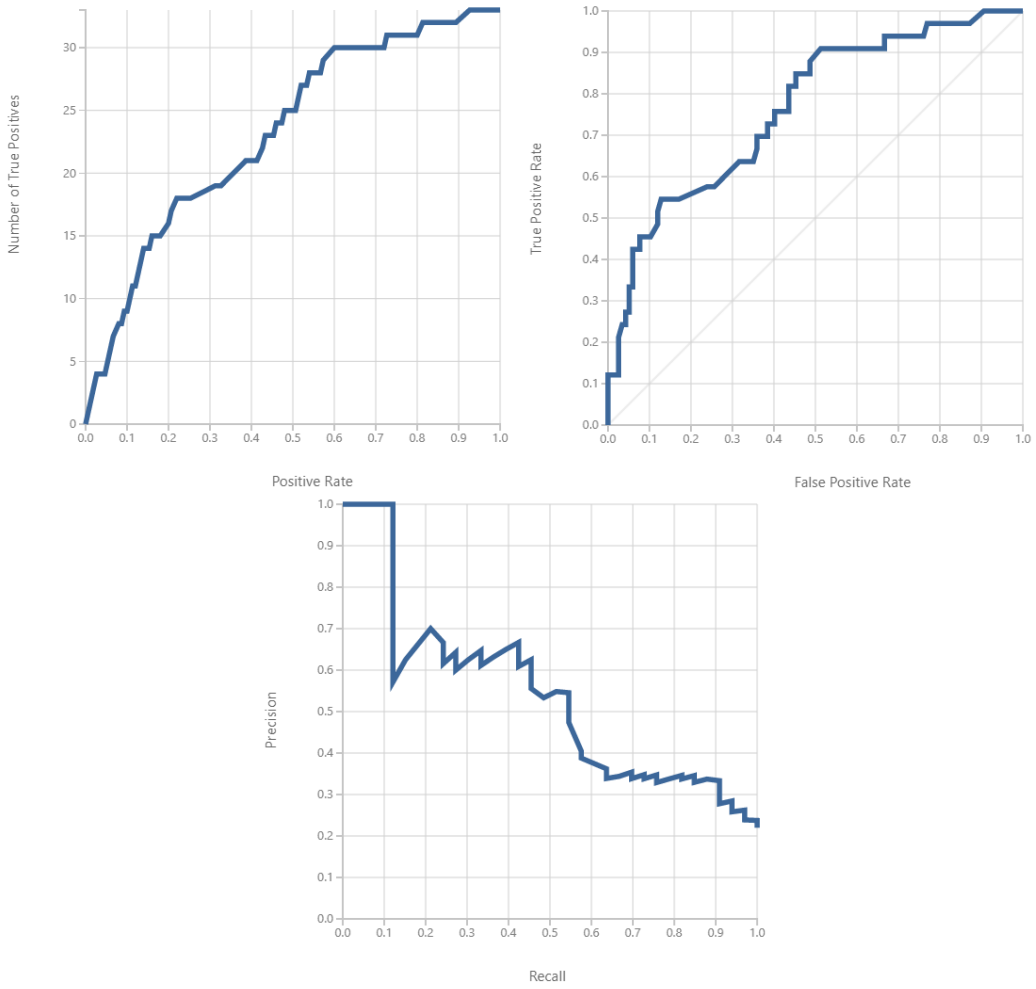
Fonte: O autor

Figura 36 – Tela de Resultados do Two-class Support Vector Machine no Azure Studio ML



Fonte: (AZURE, 2022a)

Figura 37 – Gráficos das métricas do algoritmo Two-Class Support Vector Machine



Fonte: (AZURE, 2022a)

6.1.12 Comparação de Resultados

Para fazer a avaliação foi agrupado todas as métricas conseguidas em cada treinamento e seus valores serão comparados para fazer uma análise qualitativa.

Tabela 23 – Comparação dos resultados das métricas dos algorítimos

Métrica	Maior valor	Menor valor
Accuracy	Decision Jungle	Logistic Regression
Precision	Decision Jungle	Decision Forest
Recall	Decision Jungle	Logistic Regression
F1 Score	Decision Jungle	Averaged Perceptron

Fonte: O autor

- Decision Jungle obteve os melhores resultados para Accuracy (0.847) ou seja em relação

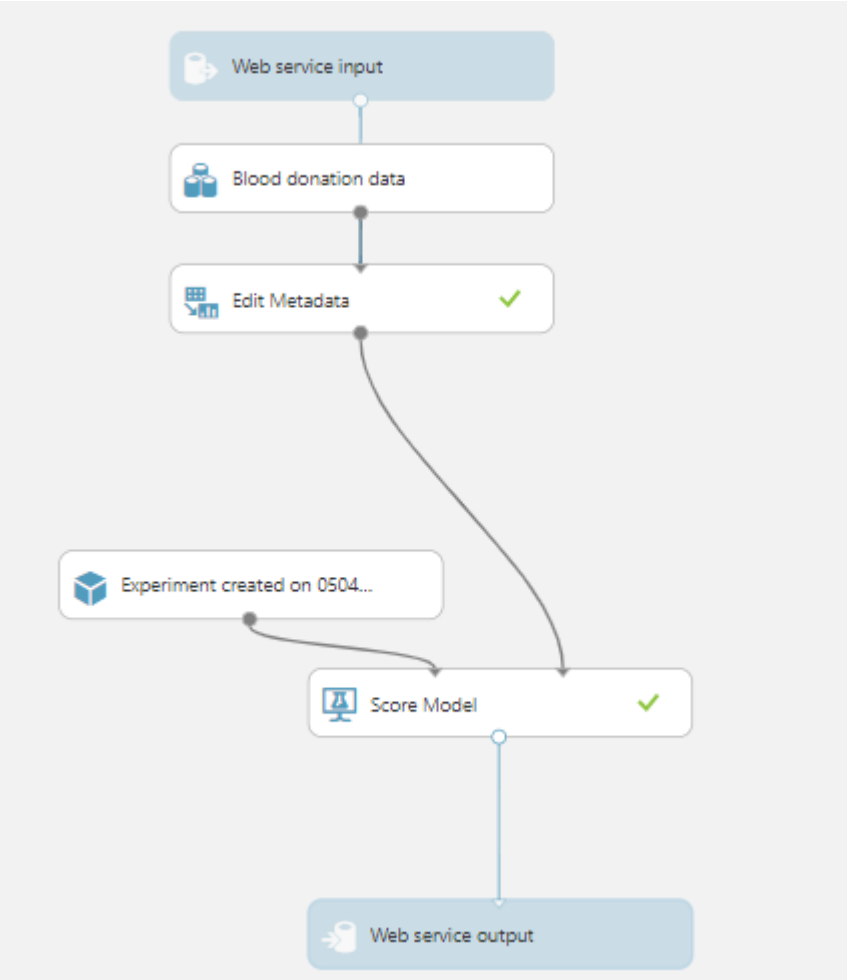
ao total de casos a porcentagem de acertos foi de quase 85%, em cada 100 indivíduos a classificação em doador potencial ou não foi correta em 85 casos, Precision (0.813) ou seja em relação ao número total de casos positivos ou de classificador como doador em potencial a classificação estava realmente correta em 81,3% dos casos, Recall capacidade do método de detectar com sucesso resultados classificados como potenciais doadores (0.394) e F1 Score (0.531).

Após a análise de métricas, o algoritmo Decision Jungle foi escolhido para criar nosso modelo. O Azure Studio ML disponibiliza uma ferramenta completa para publicar nosso modelo online permitindo criar uma API para consumir em nossa aplicação seja em um aplicativo web ou um serviço excel e etc. O Azure Studio ML cria toda a estrutura para desfrutarmos de nosso modelo podendo apenas se preocupar com a interface com o usuário. Para demonstrar usaremos o serviço de teste na própria interface web que o Azure Studio ML disponibiliza e também uma aplicação consumindo a API no Excel.

6.2 Criando serviço web do Azure Studio

Após rodarmos com sucesso nosso projeto e criar o modelo com o algorítimo do Decision Jungle é disponibilizado uma nova ferramenta com o nome de Deploy Web Service. Executando essa ferramenta o Azure irá criar um serviço web completo para nosso modelo preditivo criando uma API onde poderemos consumi-la em nossa aplicação. Nosso pipiline fica com a aparência da figura 39.

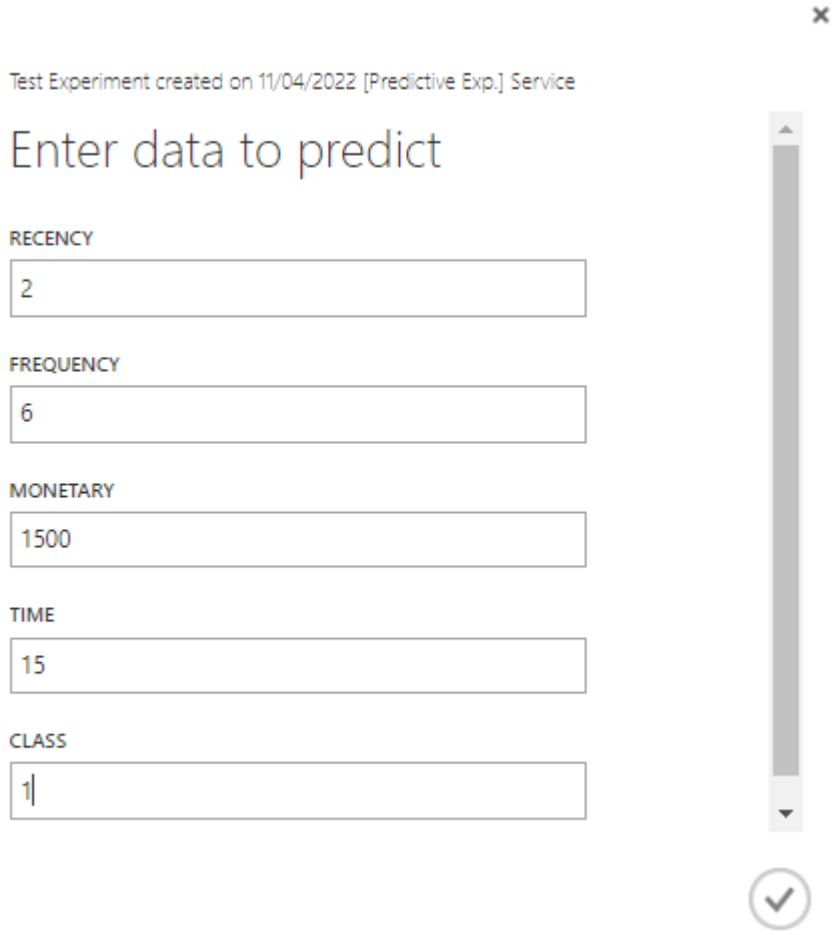
Figura 38 – Tela Run do web Service do Azure ML Studio



fonte : O autor

Podemos ver 6 módulos nessa tela. O web service Input onde é através dele que nosso serviço web irá receber os parâmetros para usar nosso serviço e assim poder fazer a predição do doador potencial. A seguir temos o módulo do nosso DataSet Blood Donation Data, o módulo Edit Metadata, o módulo do experimento onde treinamos o algítmo e o módulo de score model que é o resultado do treinamento. Temos por fim o web service output que é através dele que receberemos os resultados da nossa API com a precisão se o doador é um doador potencial ou não. O Azure cria também uma interface web que permite o uso e o teste ali mesmo de nosso modelo preditivo bastando apenas preencher os campos com valores de cada atributo executar e em poucos segundos já nos mostra a previsão do nosso teste. como nas figuras a seguir.

Figura 39 – Formulário web do Azure ML Studio



fonte : O autor

Nesse teste usamos os dados de um dos doadores reais do nosso Dataset que tinha como parâmetros:

- Meses desde a ultima doação(Recency) = 2
- Total de doações feitas(Frequency) = 6
- Quantidade de sangue doado no total(ml)(Monetary)= 1500
- Meses desde a última doação (Time) = 15

Após isso é só executar a aplicação e nosso serviço web retorna objeto result com atributos que são entre eles os parâmetros de entrada do nosso teste e os atributos do resultado do teste. O resultado do nosso teste é a imagem 41 .

Figura 40 – Resultado do teste feito na interface web do Azure

Result: {"Results":{"output1":{"type":"table","value":{"ColumnNames":
["Recency","Frequency","Monetary","Time","Potential Doador","Scored Labels","Scored Probabilities"],"ColumnTypes":
["Int32","Int32","Int32","Int32","Int32","Int32","Double"],"Values":[["2","6","1500","15","1","1","0.855872844827586"]]}}}

fonte : O autor

O objeto tem como parâmetros entre eles os seguintes atributos: o tipo: tabela, que é o modo como podemos mostrar os resultados, o nome das colunas e os respectivos valores. Destaco aqui o "Scored Labels"que é o atributo que mostra o resultado da previsão do campo doador potencial, o valor desse atributo nos dirá se tal pessoa é um potencial doador(1) ou não(2). Neste teste o resultado foi igual ao resultado retirado do Dataset. O atributo "Scored Probabilities"que mostra a probabilidade do doador pertencer a classe 1 que no caso foi de 0,85/1.

Foi criado também 0 Workbook Excel que é uma ferramenta disponibilizada pelo Azure Studio ML, este Workbook possui macros que consomem a API e assim podemos utilizar esta ferramenta para fazer a previsão através do excel dependendo da necessidade. Os resultados da previsão usando o Workbook está mostrado na figura 41.

Figura 41 – Resultado do teste feito no Workbook do Excel

PARAMETERS					PREDICTED VALUES						
Recency	Frequency	Monetary	Time	Class	Recency	Frequency	Monetary	Time	PotentialDoador	ScoredLabels	ScoredProbabilities
2	6	1500	15	1	2	6	1500	15	1	1	0,855872845

fonte : O autor

Como podemos ver o funcionamento do workbook do Excel é bem simples bastando preencher as células dos “PARAMETERS” com os atributos do doador que será feito a previsão e automaticamente ele calcula os resultados e preenche as células das colunas do “PREDICTED VALUES” :o “Scored Labels” que é o atributo que nos mostra se o doador é um potencial doador e o “Score Probabilities” que é a porcentagem de chance do doador pertencer a classe prevista. Os resultados como esperado foram os mesmos que encontramos usando a interface web do serviço do Azure ML Studio.

6.3 Resultados dos modelos de Machine Learning usando Python e seu ecossistema

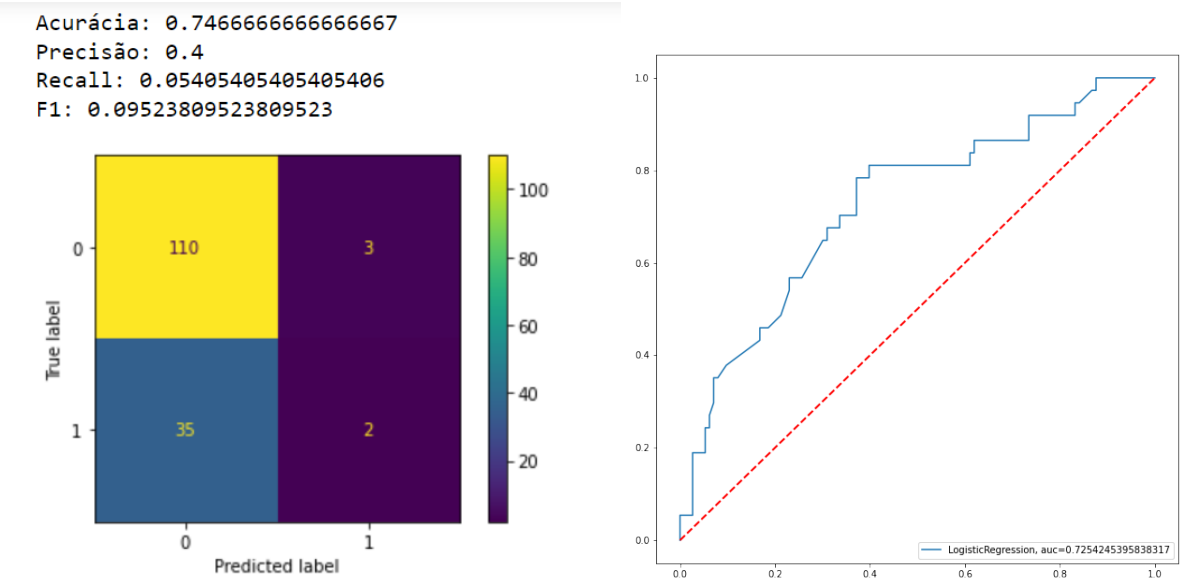
Para usar os modelos importamos do Scikit-Learn e instanciamos, logo a seguir faremos o treinamento usando a função ".fit()"passando como parâmetros o conjunto de dados de treino separados tanto o x como o y.Para predição usaremos a função ".predict()"passando os dados de testes separados. Usaremos a seguir as funções de métricas disponibilizadas pelo

Scikit - Learn que ajudará a análise da performance do algorítimo. Cada algoritmo utilizado passou por vários testes sendo o primeiro teste o treinamento com os parâmetros default listados na documentação do Scikit - Learn do modelo. Foi impresso as métricas como acurácia, precisão, Recall, F1 Score, matriz de correlação e curva AUC.

6.3.1 Logistic Regression

Foram realizados vários testes com esse modelo fazendo aprimoramentos e ajustes. No primeiro teste o Logistic Regression foi treinado com os parâmetros padrões listados na documentação, após isso os parâmetros foram alterados e o algoritmo foi novamente testados a medida que foram feito mudanças. O melhor resultado dos testes foram com os parâmetros C=1e5 que por padrão é 1.0, e maxiter=1000 que por padrão é 100, o parâmetro C refere-se a suavização da função aprendida quanto maior esse valor maior a regularização. E o parâmetro max-iter limita o número máximo de iterações tomadas para os solucionadores convergirem, os demais parâmetros ou não tiveram uma alteração significativa nos resultados ou tiveram uma redução na performance . Na figura 42 está presente a tela de saída com a métrica de melhor performance do Logistic Regression e os valores detalhados para melhor visualização na tabela 24.

Figura 42 – Métricas Logistic Regression



fonte : O autor

Tabela 24 – Resultados Logistic Regression

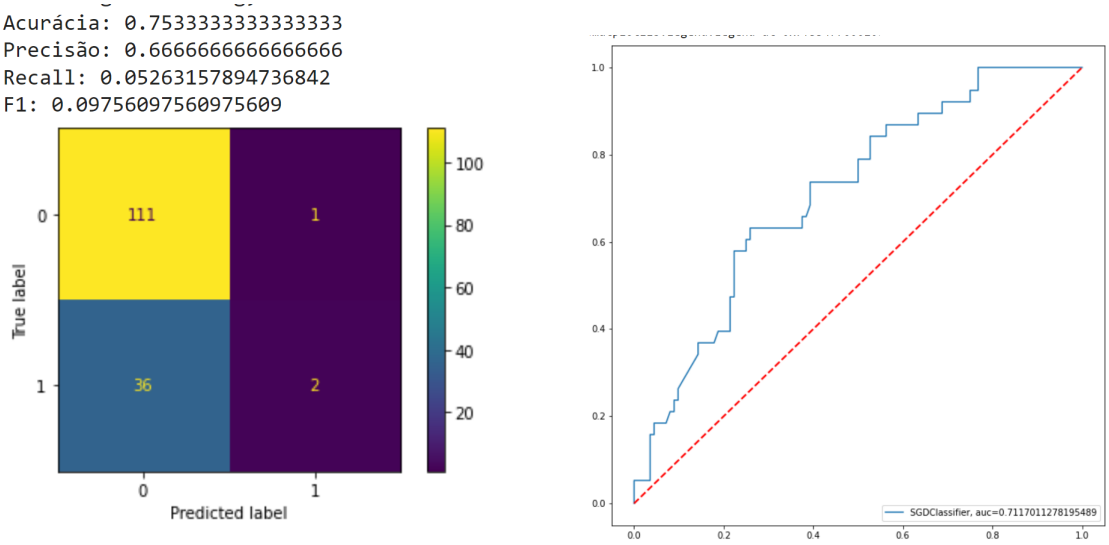
Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
3	35	2	110
Accuracy	Precision	Recall	F1 Score
0.746	0.4	0.05	0.095

Fonte: O autor

6.3.2 Classificador SGD

A função SGD "Stochastic Gradient Descent"tem como característica selecionar uma parte aleatória do conjunto de dados em vez de todo o conjunto em cada uma das iterações. Isso diminui o custo operacional quando por exemplo temos um conjunto de dados muito grande. Após executar SGDClassifier cm os parâmetros com valores default foram feito vários testes e mudanças nos parâmetros até alcançar a melhor métrica mostrada na figura 43 e tabela 25.Os parâmetros foram loss='log',alpha=1000,max_iter=10000, tol=1e-3. "loss='log'"significa que estamos implementando uma regressão logística, "alpha"é uma constante que multiplica o termo de regularização, "tol"é o critério de parada e "max_iter"é a limitação de iterações. O valor de precisão foi maior que o do logistic regression mas as outras métricas ficaram com valores inferiores.

Figura 43 – Métricas Classificador SGD



fonte : O autor

Tabela 25 – Resultados Classificador SGD

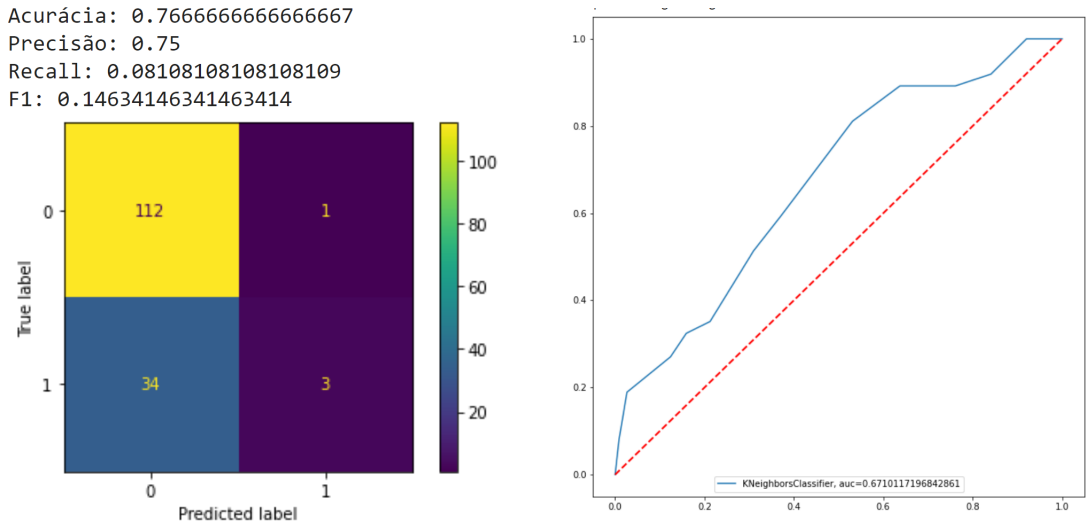
Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
1	36	2	111
Accuracy	Precision	Recall	F1 Score
0.753	0.666	0.052	0.09

Fonte: O autor

6.3.3 Classificação dos Vizinhos mais Próximos KNN - KNeighborsClassifier

Para o classificador KneighborsClassifier as melhores métricas foram usando os seguintes parâmetros descritos da figura 44 e tabela 26: "n_neighbors"= 25, onde é selecionado o número de vizinhos ou o número de "k"como é comumente usado na literatura acadêmica e "leaf_size"que determina o tamanho das "folhas"controla o número mínimo de pontos em um determinado nó.

Figura 44 – Métricas KNN



fonte : O autor

Tabela 26 – Resultados Classificador KNN

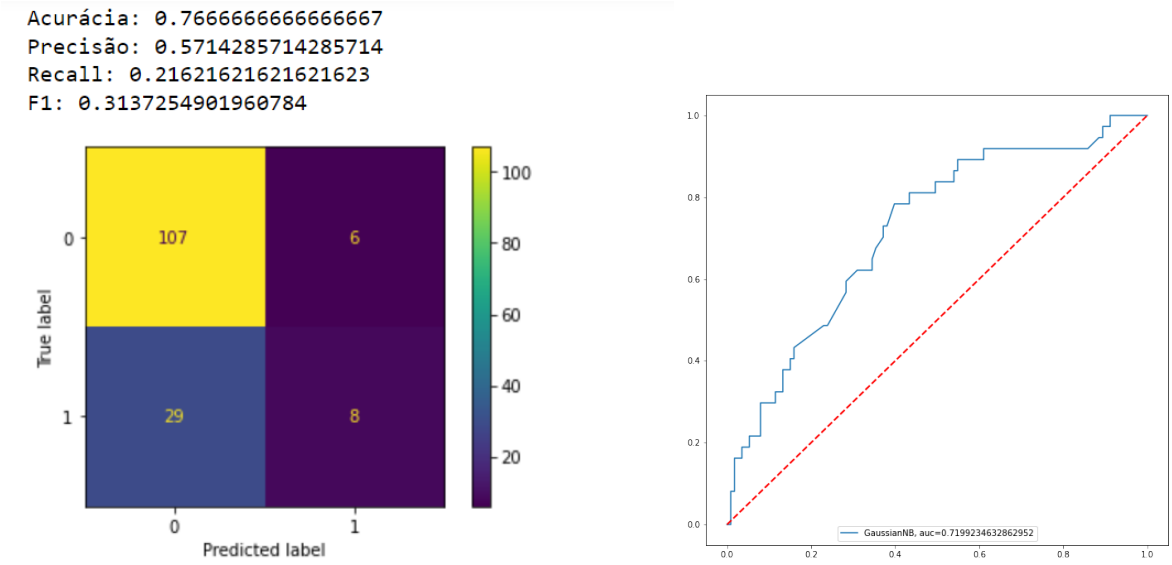
Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
1	34	5	112
Accuracy	Precision	Recall	F1 Score
0.766	0.75	0.08	0.146

Fonte: O autor

6.3.4 Algoritmo Naive Bayes

Para esse algoritmo os parâmetros de default obtiveram os melhores resultados. O parâmetro modificado foi "var_smoothing", foram feitos diversos testes em que as métricas tiveram valores piores que os parâmetros de default. Foram eles "var_smoothing": [1e-9, 1e-8, 1e-5, 1e-4]. A figura 45 e a tabela 27 mostram as melhores métricas alcançadas:

Figura 45 – Métricas Algorítimo Navie Bayes



fonte : O autor

Tabela 27 – Resultados Classificador Navie Bayes

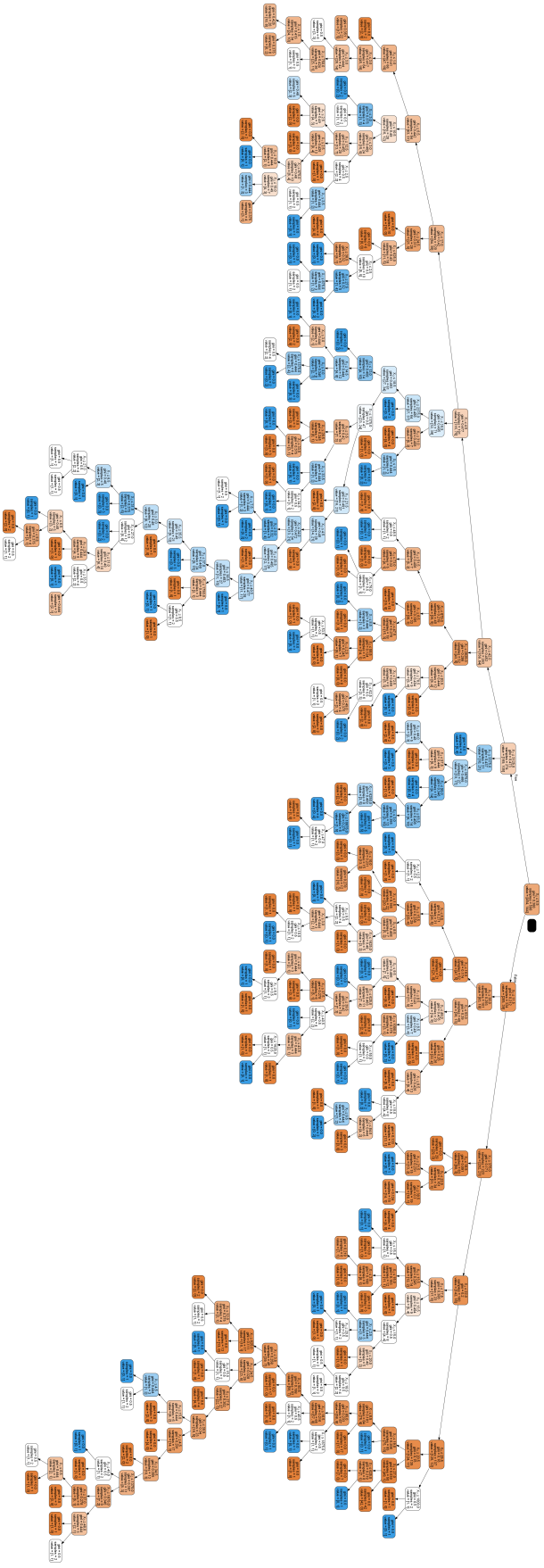
Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
6	29	8	107
Accuracy	Precision	Recall	F1 Score
0.766	0.571	0.216	0.313

Fonte: O autor

6.3.5 Decision Tree Classifier

A métrica alcançada com o uso do algorítimo Decision Tree com os parâmetros de default gerou como resultado uma acurácia de 78%, depois de vários testes foi encontrado o melhor resultado usando o parâmetro min_samples_split=0.2, esse parâmetro defini o número mínimo de amostras necessárias para dividir um nó interno. A primeira árvore sem o parâmetro selecionado gerou a árvore da figura 46 e os dados listados na tabela 28:

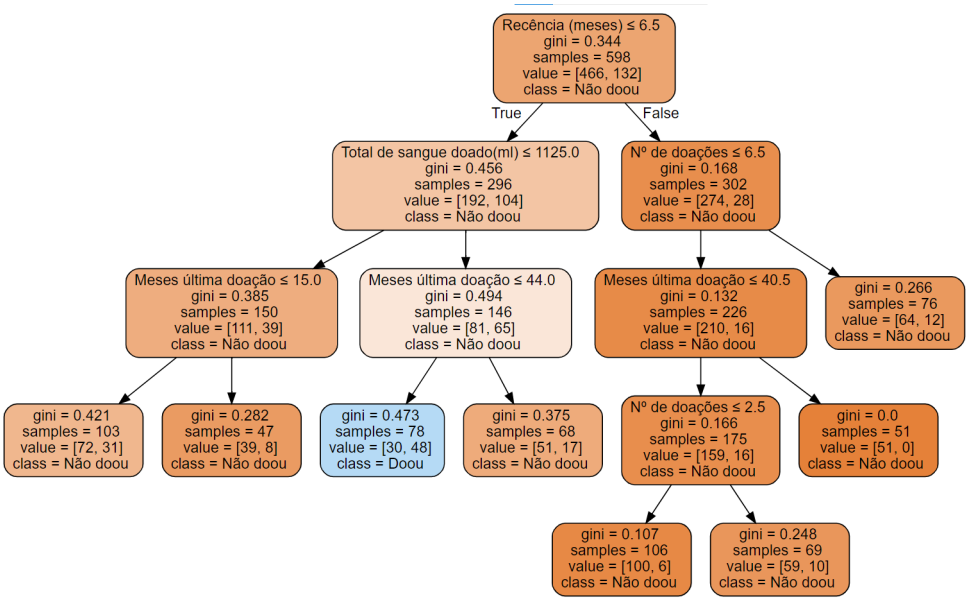
Figura 46 – Árvore sem poda



fonte : O autor

E o algoritmo otimizado com melhor performance gerou a seguinte árvore da figura 47:

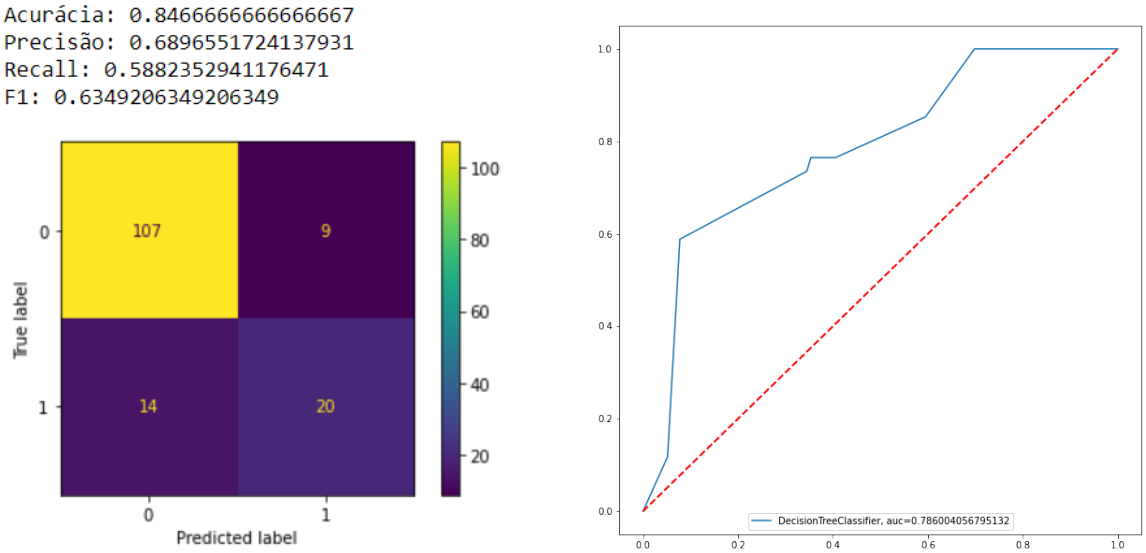
Figura 47 – Árvore com poda



fonte : O autor

As melhores métricas estão mostradas na figura 48:

Figura 48 – Métricas Algorítimo Decision Tree



fonte : O autor

Tabela 28 – Resultados Classificador Decision Tree

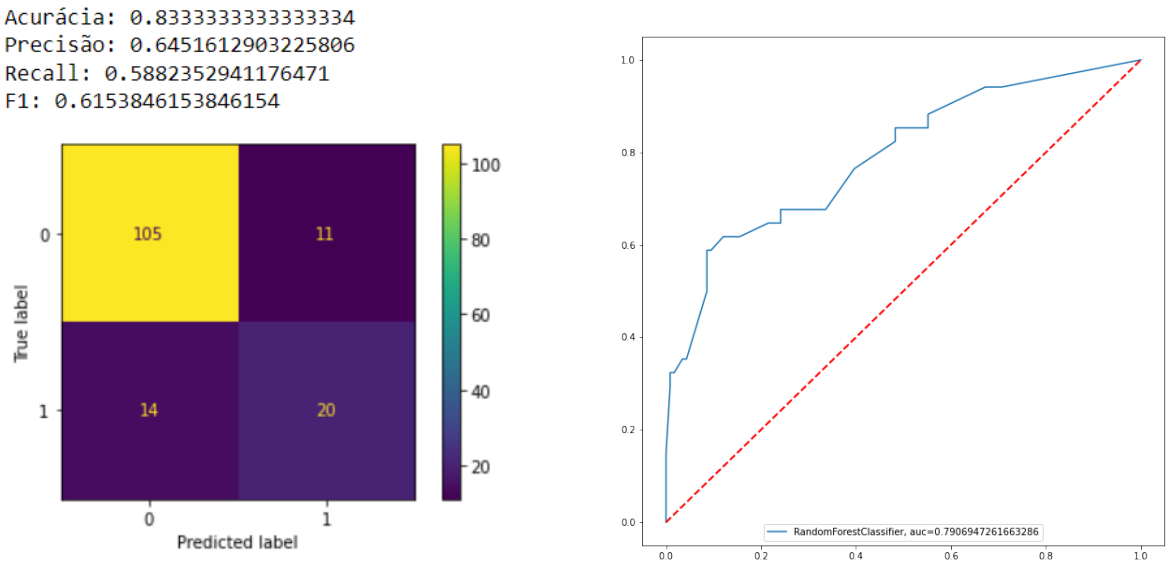
Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
9	14	20	107
Accuracy	Precision	Recall	F1 Score
0.846	0.689	0.588	0.634

Fonte: O autor

6.3.6 Random Forest

Para este classificador a primeira etapa de testes com valores padrões os resultados já apresentavam bons em relação a a maioria dos outros algoritmos mesmo otimizados. Com valores de default obtivemos nos testes uma acurácia de 80.6%, com Precisão=0.56, Recall=0.64 e F1Score = 0.60. Com valores otimizados a precisão aumentou para 83.3% com a mudança do parâmetro n_estimators=10. As demais métricas são apresentadas na figura 49 e na tabela 31:

Figura 49 – Métricas Algorítimo Random Forest



fonte : O autor

Tabela 29 – Resultados Classificador Random Forest

Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
11	14	20	105
Accuracy	Precision	Recall	F1 Score
0.833	0.645	0.588	0.615

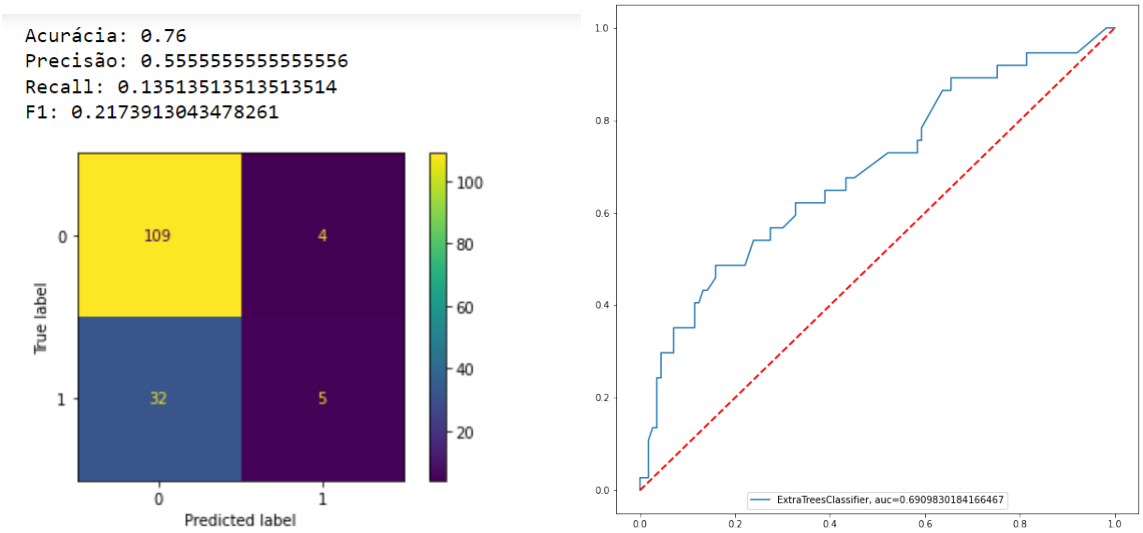
Fonte: O autor

6.3.7 Classificador Extra Tree

Este classificador usa a estratégia de criar várias árvores de decisão e assim com a técnica de combinação escolher a melhor combinação para encontrar um resultado. Os primeiros

testes por Default obtiveram uma performance com os seguintes métricas : 73.3% de acurácia com 56% de precisão. Os parâmetros que melhoraram a performance desse algorítimo foram os parâmetros: min_samples_split= 0.1, n_estimators=10. Que obtiveram a seguinte métrica detalhada na figura 50:

Figura 50 – Métricas Algorítimo Extra Tree



fonte : O autor

Tabela 30 – Resultados Classificador Random Forest

Verdadeiro Positivo	Falso Negativo	Falso Positivo	Verdadeiro Negativo
4	32	5	109
Accuracy	Precision	Recall	F1 Score
0.76	0.555	0.135	0.217

Fonte: O autor

6.4 Resultados finais dos Classificadores

Pela exploração de dados e análise gráfica as principais características que definem se um doador é um doador em potencial é o total de doações em números absolutos e o o volume de sangue doado em ml de sangue. Em relação as diferenças de desempenho e métricas dos classificadores não foi muito grande, todos estiveram uma porcentagem de acurácia acima de 70%. A tabela com os valores relativos a cada algoritmo estão descrito na tabela 31:

Tabela 31 – Resultados Classificador Random Forest

Métrica	Maior Valor	Menor Valor
Accuracy	Decision Tree	Navie Bayes
Precision	Decision Tree	Navie Bayes
Recall	Decision Tree	SGD
F1 Score	Decision Tree	SGD

Fonte: O autor

- O Algorítimo Decision Tree obteve os melhores resultados para todos os parâmetros analisados. Accuracy (0,846), Precisão (0,689), Recall (0,588) e F1 Score(0,634).

Guama (2019) cita em seu artigo três fórmulas para analisar os valores de taxa de acertos para casos negativos e positivos :

- Sensibilidade (Sensitivity, Recall): porção de VP em relação ao total de positivos. Em outras palavras, quão bom o seu classificador é para classificar corretamente a classe de interesse. [VP/(VP + FN)]
- Especificidade (Specificity): porção de VN em relação ao total de negativos. Em outras palavas, quão bom o classificador é para classificar corretamente a outra classe. [VN / (FP + VN)]
- Eficiência (Efficiency): média aritmética entre sensibilidade e especificidade. [(Sens + Espec)/2]

Os valores calculados para estas métricas estão descritas na tabela 32:

Tabela 32 – Sensibilidade para casos negativos e Positivos.

Modelos	Taxa de Acertos para classe doou	Taxa de Acertos para classe Não doou	Eficiência
LR	7,89%	98,21%	53,05%
SGD	2,70%	98,23%	50,47%
KNN	2,86%	95,73%	49,29%
Bayes	17,14%	93,04%	55,09%
DTree	39,13%	84,25%	61,69%
RForest	44,00%	84,00%	64,00%
ExtraTree	11,11%	95,61%	53,36%

Fonte: O autor.

Quanto maior for a eficiência melhor será a classificação para as duas classes, quanto menor o modelo estará fazendo a classificação para apenas uma das classes. Essas informações tem que ser levadas em conta na hora de fazer a avaliação do modelo.

A próxima etapa foi a criação de uma API que usará todos os modelos criados com cada um dos algoritmos para que seja feito mais um teste de acerto da previsão e para demonstrar o funcionamento dessa API. Para que fosse possível a criação oda API foram salvos todos os modelos criados com a função "dump()"que recebe como parâmetros o modelo já treinado e o nome do arquivo que daremos para salvar o modelo.

6.5 Criando API utilizando Flask

Para criar a API utilizaremos a IDE Visual Studio Code. Iniciamos criando um arquivo do tipo Python e salvamos. Abrimos nosso arquivo criado no Visual Studio Code e através do terminal instalamos flask e flask_restful. Nas primeiras linhas de código importamos os componentes necessários e carregamos o modelo salvo no Jupyter Notebook e que foi salvo em nosso diretório.

Criamos dois métodos um tipo GET que nos retorna um valor quando chamado e um método do tipo POST onde enviamos valores e recebemos uma resposta, o método POST enviará os atributos de nossos doadores de teste e a resposta será a classificação prevista, o código criado esta mostrado na figura 51. Após rodarmos nossa aplicação será gerado um endereço URL onde ao acessarmos iremos obter a mensagem criada no método GET. Para realizarmos a requisição do tipo POST utilizaremos um software chamado Insomnia que é uma ferramenta que permite fazer testes de API_Rest.

Figura 51 – API flask

apiblood.py 4 ●

C: > Users > silas > apiblood.py > ...

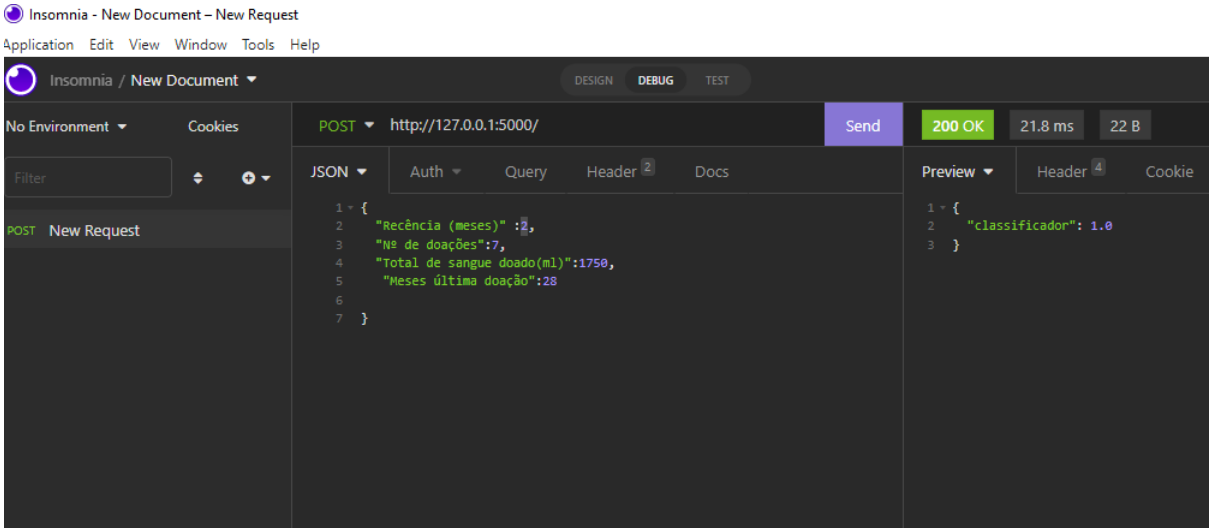
```python
from flask import Flask, jsonify, request
from flask_restful import Resource, Api
from joblib import load
import numpy as np

# from flask import Flask
# app = Flask(__name__)

# @app.route('/')
# def hello_world():
#     return 'Hello, World!'

app = Flask(__name__)

api = Api(app)

model = load('C:/Users/silas/ET.joblib')

class BloodDonation(Resource):
    def get(self):
        return {'Trabalho de Conclusao de Curso': 'Classificacao de doadores de sangue'}
    def post(self):
        args = request.get_json(force=True)
        input_values = np.asarray(list(args.values())).reshape(1, -1)
        predict = model.predict(input_values)[0]

        return jsonify({'classificador': float(predict)})

api.add_resource(BloodDonation, '/')

if __name__ == '__main__':
    app.run()
```

fonte : O autor

Para utilizar o Insomnia basta fazer o download do software no site oficial e após o download iniciamos o software e na tela inicial clicar na opção "New Request", temos que dar nome para a requisição criada e no painel seguinte adicionar a URL da nossa Api criada. Nosso método POST recebe um dicionário de dados de tipo JSON e nesse dicionário está presente os atributos do doador que queremos fazer a classificação. Então para realizar a requisição selecionaremos o tipo de texto para JSON e escrevemos no corpo da mensagem o dicionário de dados com as características do doador. A criação da API e o uso do software Insomnia foi realizada seguindo tutorial do (GELATTI, 2020).

Após o envio da requisição receberemos como retorno um valor que é a previsão 0 ou 1 do doador em potencial como mostrado na figura 52. Utilizaremos essas ferramentas para fazer algumas previsões e avaliar os resultados.

Figura 52 – Método POST usando Software Insomnia



fonte : O autor

6.5.1 Resultado dos testes dos testes feitos usando API

Testamos cinco casos de testes sendo três casos em que o doador é um doador não potencial, ele não doou na data da coleta de dados e outros dois foi um doador q doou na data especificada, testamos os cinco casos para ver se nosso modelo conseguia prever se a pessoa doaria ou não e comparar com os dados reais.

Os resultados dos testes mostrados na tabela 33 que todos os modelos classificaram corretamente os doadores de classe 0 ou seja que não doaram na data especificada, mas apenas o modelo criado com Random Forest e Extra Tree conseguiram prever corretamente os doadores que realmente foram doar na data específica.

Tabela 33 – Resultados dos testes na API flask.

Pessoa ID	Doou ou não	LR	KNN	NB	DT	RF	ET
668	Não	Não	Não	Não	Não	Não	Não
317	Não	Não	Não	Não	Não	Não	Não
612	Sim	Não	Não	Não	Sim	Sim	Não
57	Sim	Não	Não	Não	Sim	Sim	Sim
719	Não	Não	Não	Não	Não	Não	Não

Fonte: O autor.

7 Conclusão

Neste trabalho foi demonstrado diversas técnicas de machine Learning com o objetivo de explorar diversas opções de técnicas de aprendizado de máquina, criar modelos baseados em dados históricos reais de doações. O trabalho foi divido em duas frentes criando experimentos usando o Azure Machine Learning Studio, uma plataforma de ciência de dados que oferece uma ferramenta robusta e facilitada para criação de projetos de machine learning, onde foi demonstrado as etapas de criação dos modelos e dos serviços web.

Outra parte do trabalho foi a criação de modelos de aprendizado de máquina usando ecossistema Python como a biblioteca scikit-Learn e softwares de apoio. O desempenho geral dos algoritmos foram demonstrados e comparados entre si para análise exploratória de suas métricas.

Pelas características do conjunto de dados objeto de nosso trabalho todos os algoritmos tiveram uma acertabilidade acima de 75% nas situações onde o doador doou e não doou, já para os casos exclusivamente positivos ou negativos tivemos dois cenários completamente diferentes, de acordo com a tabela 32 podemos ver estes diferentes cenários e concluir que a avaliação para casos em que o doador não doou foram melhores que para casos em que o doador não doou.

7.1 Limitações

As limitações do trabalho foram relativos as características do conjunto de dados, por ser um conjunto com informações de doadores de outro país e por ser um conjunto de dados já com um certo tempo da coleta, 15 anos desde a coleta destes dados, os resultados refletem a realidade dos doadores de outro país, que podem não ser iguais a do Brasil. Outra limitação foi o número de características coletadas dos doadores que ficaram limitadas as informações sobre as doações e nenhuma sobre características pessoais e físicas das pessoas.

7.2 Trabalhos Futuros

Para trabalhos futuros sugere-se que possa ser criado um novo conjunto de dados que possa ser compartilhado com a comunidade cientifica, já que como comentado nas limitações, o conjunto de dados público mais atual completou mais de 15 anos de sua criação, sugere-se que este conjunto de dados consiga acrescentar informações sobre as características dos doadores como faixa etária, sexo, idade, entre outras informações relevantes, que tenha uma quantidade maior de dados em relação ao número de doadores e que se possível sejam doadores da região ou lolalidades próximas.

Outro sugestão é a realização da pesquisa utilizando os algoritmos que ficaram de fora deste trabalho.

Referências

ALECRIM, E. Machine learning: o que é e por que é tão importante. Tecnoblog, 2018. Último acesso em Maio de 2021. Disponível em: <https://tecnoblog.net/responde/machine-learning-ia-o-que-e/>. Citado nas páginas 30 e 31.

AWARI. Algoritmos de classificação: O que são e como funcionam. Awari.com, Outubro 2020. Acessado em Fevereiro de 2022. Disponível em: <https://awari.com.br/algoritmos-de-classificacao/>. Citado na página 31.

AZURE, M. Microsoft machine learning studio (classic). 2022. Disponível em: <https://studio.azureml.net/>. Citado nas páginas 36, 40, 41, 43, 44, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67 e 68.

AZURE, M. *O que é computação em nuvem?* azure.microsoft.com, 2022. Disponível em: <https://azure.microsoft.com/pt-br/overview/what-is-cloud-computing/#benefits>. Citado nas páginas 35 e 36.

BERTOZZO, R. Aplicação de machine learning em dataset de consultas médicas do sus. *UFSC - UNIVERSIDADE FEDERAL DE SANTA CATARINA*, 2019. Citado nas páginas 16, 19 e 20.

CADE, J. Como a inteligência artificial pode ser usada a favor da medicina. Medicina S/A, Outubro 2020. Acessado em Setembro de 2021. Disponível em: <https://medicinasa.com.br/ia-medicina/>. Citado na página 15.

CATHO, C. ao. O dia do doador de sangue e a conscientização das empresas. Novembro 2015. Disponível em: <https://www.catho.com.br/carreira-sucesso/colunistas/noticias/o-dia-do-doador-de-sangue-e-a-conscientizacao-das-empresas>. Citado na página 17.

CERRI, R.; CARVALHO, A. C. P. d. L. F. d. Aprendizado de mÁquina: Breve introduÇÃo e aplicaÇÕes. Cadernos de Ciência Tecnologia, 2017. Citado na página 15.

DAVID, M. Microsoft azure machine learning. CienciaEDados, Julho 2016. Disponível em: <https://ww.cienciaedados.com/microsoft-azure-machine-learning/>. Citado na página 37.

DAVID, M. O kaggle é realmente válido para aprender data science? cienciaedados, Outubro 2020. Citado na página 38.

DIDATICATECH. A biblioteca scikit-learn – pyhton para machine learning. 2022. Acessado em Junho de 2022. Disponível em: <https://didatica.tech/a-biblioteca-scikit-learn-pyhton-para-machine-learning/>. Citado na página 37.

FREITAS, T. *Os três tipos de aprendizado no machine learning, um ramo da inteligência artificial*. 2019. Disponível em: <hhttps://www.startse.com/noticia/nova-economia/machine-learning-inteligencia-artificial-aprendizado>. Citado nas páginas 31 e 34.

GELATTI, H. S. Criando uma api utilizando flask. Medium.com, Julho 2020. Acessado em Junho de 2022. Disponível em: <https://medium.com/@henrique.gelatti/criando-uma-api-utilizando-flask-40a0d7ab2371>. Citado na página 82.

GOMES, P. C. T. Regressão linear: entenda como utilizar. DataGeeks, 2019. Disponível em: <https://www.datageeks.com.br/regressao-linear/>. Citado na página 34.

GUAMA, J. Métricas de avaliação de classificadores. Medium.com, Março 2019. Acessado em Junho de 2022. Disponível em: <https://medium.com/pyladiesbh>. Citado na página 81.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*: Data mining, inference, and prediction. [S.l.]: Springer, 2008. (Springer Series in Statistics). Citado na página 16.

JOSHI, P. Bayes point machines. prateekvjoshi.com, March 2013. Disponível em: <https://prateekvjoshi.com/2013/03/05/bayes-point-machines/>. Citado na página 33.

LAURETTO, M. Árvores de decisão. *EEACH - Escola de Artes, Ciências e Humanidades da Universidade de São Paulo*, USP - Universidade de São Paulo, São Paulo, SP , Brasil, p. 1–2, Novembro 2010. Citado na página 31.

LINS, F. *Google Scholar: o que é e como utilizar em sua vida acadêmica*. 2021. Disponível em: <https://doity.com.br/blog/google-scholar-o-que-e-e-como-utilizar-em-sua-vida-academica/>. Citado na página 19.

LUCIDCHART. O que é um diagrama de árvore de decisão? Lucid Software Inc., 2022. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-arvore-de-decisao>. Citado na página 32.

MATOS, D. Por que cientistas de dados escolhem python? cienciaedados, Abril 2019. Acessado em Abril de 2022. Disponível em: <https://www.cienciaedados.com/por-que-cientistas-de-dados-escolhem-python>. Citado na página 37.

MELO, K. Covid-19: doações de sangue caem 20% e governo lança campanha: Meta é melhorar a informação sobre a segurança da doação. Agência Brasil, Março 2021. Citado na página 17.

MELO, L. C. Conheça os métodos de machine learning mais populares. SGA TI em Nuvem, 2020. Citado na página 16.

MICROSOFT. Documentação do azure machine learning. Microsoft, 2022. Disponível em: <https://docs.microsoft.com/pt-br/azure/machine-learning/>. Citado nas páginas 39 e 50.

MICROSOFT. Edit metadata component. Microsoft, 2022. Disponível em: <https://docs.microsoft.com/pt-br/azure/machine-learning/component-reference/edit-metadata>. Citado na página 41.

MICROSOFT. Split data component. Microsoft, 2022. Disponível em: <https://docs.microsoft.com/pt-br/azure/machine-learning/component-reference/split-data>. Citado na página 42.

NEVES, D. Ampliando a análise com o describe. Alura, Maio 2021. Acessado em Junho de 2022. Disponível em: <https://www.alura.com.br/artigos/ampliando-a-analise-com-describe>. Citado na página 46.

NEXXTO. Machine learning: aplicações do aprendizado de máquina na área médica. Nexxto Tecnologia, Abril 2021. Acessado em Junho de 2022. Disponível em: <https://nexxto.com/machine-learning-aplicacoes-do-aprendizado-de-maquina-na-area-medica/>. Citado na página 18.

NISHAT, N. L. *Blood Transfusion Dataset*. kaggle, 2019. Disponível em: <https://www.kaggle.com/ninalabiba/blood-transfusion-dataset>. Citado nas páginas 38 e 39.

OLIVERA, A. R. et al. Comparison of machine-learning algorithms to build a predictive model for detecting undiagnosed diabetes – elsa-brasil: accuracy study. *Sao Paulo Medical Journal*, UFRG - Universidade Federal do Rio Grande do Sul, 2017. Citado na página 16.

PATEL, N. Machine learning: O que É, para que serve, benefícios e muito mais! Neil Patel Digital, 2021. Citado na página 16.

PAZ, M. R. Machine learning aplicado a usinagem: Previsão de in-formações de um processo de usinagem por machine learning com dados de vibração obtidos com aplicativo sci journal. *UNIVERSIDADE DO VALE DOS RIOS DOS SINOS - UNISINOS*, 2019. Citado nas páginas 19, 22, 23 e 24.

PIMENTA, I.; SOUZA, T. Desafios da doação de sangue durante a pandemia no brasil. Escola de Medicina Souza Marques da Fundação Técnico-educacional Souza Marques, Rio de Janeiro, RJ , Brasil, p. 1–2, 2020. Citado na página 18.

PROSANGUE. Estudanrtes. prosangue.sp.gov.b, s.d. Citado na página 29.

RODRIGUES, R.; REIBNITZ, K. Estratégias de captação de doadores de sangue : uma revisão integrativa da literatura. *UFSC - Universidade Federal de Santa Catarina*, SciELO - Scientific Electronic Library Online, Florianópolis, SC , Brasil, 2011. Citado na página 18.

SANTOS, H. G. d. et al. Machine learning para análises preditivas em saúde: exemplo de aplicação para predizer óbito em idosos de são paulo, brasil. Cadernos de Saúde Pública, 2019. Citado na página 16.

SAS. *Machine Learning*: O que é e qual sua importância? s.d. Disponível em: <https://www.sas.com/pt_br/insights/analytics/machine-learning.html>. Citado na página 16.

SBCOACHING. Tomada de decisão: entenda o que é e qual a sua importância. SBCOACHING Group, 2020. Citado na página 15.

SICHMAN, J. S. Inteligência artificial e sociedade: avanços e riscos. SciELO - Scientific Electronic Library Online, 2021. Citado na página 30.

TAI, E. Perceptron algorithms for linear classification. Towards Data Science, Agosto 2019. Disponível em: <https://towardsdatascience.com/perceptron-algorithms-for-linear-classification-e1bb3dcc7602>. Citado na página 33.

TALARI, S. Random forest® vs decision tree: Key differences. KDnuggets, Fevereiro 2022. Acessado em Junho de 2022. Disponível em: <https://www.kdnuggets.com/2022/02/random-forest-decision-tree-key-differences.html>. Citado na página 33.

TIBCO. What is supervised learning? TIBCO Software Inc., s.d. Acessado em Abril de 2022. Disponível em: <https://www.tibco.com/pt-br/reference-center/what-is-supervised-learning>. Citado na página 34.

TOTVS, E. *O que é Inteligência artificial? Como funciona, exemplos e aplicações*. 2019. Disponível em: <https://www.totvs.com/blog/inovacoes/o-que-e-inteligencia-artificial/>. Citado na página 30.

VEJASAUDE, R. Entenda o processo de doação de sangue e seja uma doadora. Grupo Abril, Outubro 2016. Último acesso em Abril de 2021. Disponível em: <https://saude.abril.com.br/bem-estar/entenda-o-processo-de-doacao-de-sangue-e-seja-uma-doadora/>. Citado na página 29.

WEST, M.; BORGES, A. Por que o python é mais poderoso que outras linguagens de programação? quora, 2017. Citado na página 44.