

RELATÓRIO FINAL DE INICIAÇÃO CIENTÍFICA

NOME DO ALUNO: Willian Penteado

ORIENTADORA: Sandra Mara Guse Scós Venske

REFERENTE AO PERÍODO: 01/09/2021 a 31/08/2022

TÍTULO: Previsão de Risco de Diabetes em Estágio Inicial Usando Técnica de Aprendizado de Máquina

PALAVRAS-CHAVE: aprendizado supervisionado, problemas de classificação, seleção de características, classificadores.

RESUMO

O Aprendizado de Máquina (AM) é o campo de estudo que dá aos computadores a habilidade de aprender sem ser explicitamente programado. O objetivo deste trabalho foi o estudo de técnicas de AM a serem aplicadas no problema de classificação de *Risco de Diabetes em Estágio Inicial*. O Diabetes é uma doença causada pela produção insuficiente ou má absorção da insulina e atinge mais de 13 milhões de pessoas no Brasil. Onze classificadores foram testados no problema, sendo eles: Regressão Logística, K-ésimo Vizinho mais Próximo, Máquinas de Vetores de Suporte núcleo Linear e RBF, Processo Gaussiano, Árvores de Decisão, Florestas Aleatórias, *Perceptron* Multicamadas, *AdaBoost*, *Naive Bayes* e Análise Discriminante Quadrática. Além disso, foram estudadas duas técnicas de seleção de características e utilizadas duas métricas de avaliação dos modelos obtidos. O modelo de Florestas Aleatórias foi o classificador que obteve melhor desempenho e o *Naive Bayes* obteve o pior desempenho.

Sumário

1	INTRODUÇÃO	2
2	OBJETIVOS	2
2.1	Objetivo Geral	2
2.2	Objetivos específicos	2
3	METODOLOGIA	2
3.1	Python e <i>framework Scikit-Learn</i>	3
3.2	Aprendizado de Máquina	3
3.2.1	Aprendizado Supervisionado	3
3.3	Pré-Processamento	3
3.3.1	Transformação de Dados: Conversão Simbólico-Numérico	4
3.3.2	Redução da Dimensionalidade: Seleção de Características	4
3.4	Algoritmos de Classificação	4
3.5	Validação do Modelo	4
3.6	Métricas de Avaliação	6
3.6.1	Métrica Acurácia	6
3.6.2	Métrica F_1 -score	6
3.7	<i>Dataset</i> Utilizado	7
3.8	Algoritmo Desenvolvido	7
4	RESULTADOS E DISCUSSÃO	8
4.1	Resultados dos Métodos de Seleção de Características utilizando <i>Holdout</i>	9
4.2	Resultados com a Validação Cruzada	9
5	CONCLUSÕES	11

1 INTRODUÇÃO

Nas últimas décadas, os problemas começaram a ter uma maior complexidade e o volume de dados gerados também vem crescendo. Assim sendo, houve necessidade de ferramentas e técnicas computacionais mais sofisticadas, que reduzissem a intervenção humana e dependência de especialistas. Com isso, tais técnicas deveriam ser capazes de tratar esses problemas, partindo de experiências anteriores, hipóteses, ou funções criadas pelas mesmas. Dá-se o nome de Aprendizado de Máquina (AM) o processo de indução de uma hipótese a partir de experiências anteriores (Faceli et al., 2021).

O AM pode ser definido como “a ciência (e a arte) da programação de computadores para que eles possam aprender com os dados”. Uma definição mais abrangente “é o campo de estudo que dá aos computadores a habilidade de aprender sem ser explicitamente programado” (Géron, 2019).

O Diabetes é uma doença causada pela produção insuficiente ou má absorção da insulina, que é um hormônio que regula a glicose no sangue e ajuda na produção de energia. Esta doença pode causar o aumento da glicemia, bem como pode levar a complicações no coração, nas artérias, nos olhos, nos rins e nos nervos. Segundo, a Sociedade Brasileira de Diabetes, existem atualmente, no Brasil, mais de 13 milhões de pessoas vivendo com a doença. Isto representa 6,9% da população brasileira. Neste trabalho foi usado um conjunto de dados (*dataset*) público sobre a doença obtido originalmente do *National Institute of Diabetes and Digestive and Kidney Diseases*. O objetivo do *dataset* é prever com base em medidas de diagnóstico se um paciente tem diabetes (de Saúde, 2022).

O objetivo deste trabalho foi o estudo de técnicas de AM supervisionado a serem aplicadas no problema de classificação de *Risco de Diabetes em Estágio Inicial*.

Neste estudo foi utilizada a técnica de *holdout* na fase de pré-processamento, sendo o *dataset* dividido em conjuntos de treinamento e teste. Foram estudadas e aplicadas técnicas de seleção de características utilizando o *SelectKBest(chi2)* e *Recursive Feature Elimination* (RFE). Foram testados onze classificadores, sendo eles: Regressão Logística (RL), K-ésimo Vizinho mais Próximo (KNN), Máquinas de Vetores de Suporte (SVM) núcleo Linear e RBF, Processo Gaussiano (PG), Árvores de Decisão (AD), Florestas Aleatórias (FA), *Perceptron* Multicamadas (MLP), *AdaBoost*, *Naive Bayes* (NB) e Análise Discriminante Quadrática (QDA). A avaliação desses classificadores utilizou validação cruzada com duas métricas, a acurácia e o F_1 -score.

Este relatório está organizado da seguinte forma. A Seção 2 mostra os objetivos do trabalho. Na Seção 3 são apresentados os conceitos e métodos utilizados para a realização do trabalho, bem como o algoritmo desenvolvido. Na Seção 4 são discutidos os resultados da proposta. Na Seção 5, são listadas as conclusões e trabalhos futuros. Alguns resultados complementares são apresentados no Apêndice A.

2 OBJETIVOS

2.1 Objetivo Geral

O objetivo deste projeto de iniciação científica foi o estudo de técnicas de aprendizado supervisionado de máquina aplicadas para a previsão de risco de diabetes em estágio inicial (problema de classificação).

2.2 Objetivos específicos

Os objetivos específicos relacionados a este projeto foram:

- Estudo da linguagem de programação Python com foco em bibliotecas e *frameworks* de AM;
- Aprofundamento de conhecimentos relativos a diferentes técnicas e algoritmos de AM, bem como métricas relacionadas.

3 METODOLOGIA

A metodologia utilizada neste projeto baseou-se nas seguintes etapas principais:

- Etapa 1. Estudo da linguagem de programação Python;
- Etapa 2. Estudo dos conceitos sobre AM envolvidos;
- Etapa 3. Estudo do *dataset* de previsão de risco de diabetes em estágio inicial;
- Etapa 4. Desenvolvimento e implementação de algoritmos;
- Etapa 5. Realização de experimentos computacionais;
- Etapa 6. Análise dos resultados;
- Etapa 7. Elaboração de artigo/resumo e do relatório final.

3.1 Python e *framework Scikit-Learn*

Neste trabalho foi usado o *framework Scikit-Learn*¹ que é baseado na linguagem Python².

Python é uma linguagem de programação interpretada de código aberto que possui um grande número de bibliotecas complementares úteis para aprendizado de máquina.

Scikit-learn é uma biblioteca de aprendizado de máquina de código aberto em Python que oferece suporte ao aprendizado supervisionado e não supervisionado. Ele também fornece várias ferramentas para ajuste de modelos de AM, pré-processamento de dados, seleção de modelos, avaliação de modelos e outras utilidades (Pedregosa et al., 2011).

3.2 Aprendizado de Máquina

O conceito de aprendizagem de máquina é a ciência da programação de computadores para que eles possam aprender com base nos dados, sem serem exclusivamente programados para isso (Géron, 2019).

O Aprendizado de Máquina se destaca nos problemas muito complexos para abordagens tradicionais ou que não possuem um algoritmo conhecido. Como exemplos pode-se citar o reconhecimento de fala e o filtro de *spam* (Géron, 2019).

3.2.1 Aprendizado Supervisionado

Existem diferentes tipos de sistemas de Aprendizado de Máquina sendo que pode-se classificar em diferentes categorias com base em (Géron, 2019):

- Serem ou não treinados com supervisão humana, como aprendizado supervisionado, não supervisionado, semi-supervisionado e aprendizado por reforço;
- Se podem ou não aprender rapidamente, de forma incremental, como aprendizado *online* e aprendizado por lotes;
- Se funcionam simplesmente comparando novos pontos de dados com pontos de dados conhecidos, ou se detectam padrões em dados de treinamento e criam um modelo preditivo, como o aprendizado baseado em instâncias e o aprendizado baseado em modelo.

Neste trabalho foi optado pelo Aprendizado de Máquina Supervisionado, para implementação de um classificador de diagnóstico para diabetes em estágio inicial. Neste tipo de aprendizado os dados carregados foram divididos em dois conjuntos, um para treino do algoritmo e outro para realização de testes. A parte do *dataset* que é composta pelos resultados é chamada de rótulo. Os rótulos são a principal diferença desse tipo de aprendizado dos demais.

3.3 Pré-Processamento

O Pré-Processamento é uma das partes fundamentais no aprendizado de máquina, a partir dele pode-se treinar e testar os algoritmos. A fim de suprir diferentes necessidades no tratamento de dados, existem várias técnicas, como: eliminação manual de atributos, integração de dados, amostragem de dados, dados desbalanceados, limpeza de dados, transformações de dados e redução de dimensionalidade (Faceli et al., 2021).

As técnicas utilizadas neste trabalho são descritas nas subseções seguintes.

¹<https://scikit-learn.org/stable/>

²<https://www.python.org>

3.3.1 Transformação de Dados: Conversão Simbólico-Numérico

Algumas técnicas e algoritmos não conseguem trabalhar bem com dados não numéricos. Por isso, torna-se necessário realizar algumas conversões, como a que foi feita no *dataset* utilizado neste trabalho, onde todas as instâncias que eram “Sim” e “Não” foram convertidas para “1” e “0”, respectivamente. Após essa transformação o *dataset* ficou binário, com isso, se tornou possível trabalhar com mais técnicas e algoritmos de AM (Faceli et al., 2021).

3.3.2 Redução da Dimensionalidade: Seleção de Características

Esta técnica analisa quais são as características mais relevantes ao conjunto de dados. Assim, ela pode melhorar a qualidade dos dados e o desempenho do algoritmo de AM. Desta forma é possível avaliar a qualidade de cada característica, podendo ser dividida em três abordagens: Embutida, Filtro e *Wrapper*. Neste trabalho foram utilizadas as técnicas *SelectKBest(chi2)* que é uma técnica do tipo Filtro e *Recursive Feature Elimination* que é do tipo *wrapper*. Elas foram escolhidas considerando o *dataset* ser do tipo binário (Ferreira and de Castro Jorge, 2007).

Neste trabalho foi utilizado o *SelectKBest(chi2)* que é uma técnica com abordagem do tipo Filtro, isto é, ela recebe todo o conjunto de treino, com todas as características, então faz uma avaliação dos dados e produz um subconjunto de maior relevância e com menos características. Também foi utilizado o *Recursive Feature Elimination* (RFE) que é uma abordagem do tipo *wrapper*. Essa abordagem produz como resultado um subconjunto de características como candidato, então executa o algoritmo de AM e mede a acurácia do modelo. Esse processo é repetido com cada subconjunto de atributos gerado até que um critério de parada seja atendido (Ferreira and de Castro Jorge, 2007). Resumidamente, as duas técnicas podem ser definidas como:

- *SelectKBest(chi2)*: Esta abordagem calcula estatísticas *chi-squared*, entre cada característica e a sua classe. Este valor resultante pode ser usado para selecionar algumas características com os valores mais altos de *chi-squared* (Pedregosa et al., 2011).
- *Recursive Feature Elimination*: Esta abordagem faz um *ranking* das características, com base em um estimador que associa pesos às características. O objetivo do RFE é selecionar as características considerando recursivamente conjuntos menores delas (Pedregosa et al., 2011).

3.4 Algoritmos de Classificação

Neste trabalho foram utilizados onze classificadores, disponíveis no *framework Scikit Learn*. A Tabela 1 descreve os classificadores utilizados: Regressão Logística (RL), K-ésimo Vizinho mais Próximo (KNN), Máquinas de Vetores de Suporte (SVM) núcleo Linear e RBF, Processo Gaussiano (PG), Árvores de Decisão (AD), Florestas Aleatórias (FA), *Perceptron* Multicamadas (MLP), *AdaBoost*, *Naive Bayes* (NB) e Análise Discriminante Quadrática (QDA).

A Figura 1 apresenta os algoritmos classificados de acordo com os métodos principais utilizados, segundo (Faceli et al., 2021) baseados em: Distância, Simbólicos, Probabilísticos, Maximização de Margens e Conexionalistas.

Os parâmetros dos algoritmos utilizados são apresentados na Tabela 2.

3.5 Validação do Modelo

Uma forma de validar um modelo de AM é conhecida como método *Holdout*. O *Holdout* separa uma parte dos dados de treinamento e usa para obter previsões do modelo treinado sobre o restante dos dados. Em seguida, uma estimativa de erro mede como o modelo está funcionando em dados não vistos (Géron, 2019).

Uma alternativa para avaliar o modelo é utilizar o recurso da validação cruzada, sendo que o conjunto de treinamento é dividido em subconjuntos complementares e cada modelo é treinado com uma combinação diferente desses subconjuntos e validado em relação às partes restantes. Na validação cruzada *K-fold* o conjunto de treinamento é dividido aleatoriamente em *K* subconjuntos distintos chamados de partes (*folds*), então treina e avalia o modelo *K* vezes escolhendo uma parte (*fold*) diferente a cada uma delas para avaliação e treinando nas outras *K-1* partes (Géron, 2019).

Tabela 1: Descrição dos diferentes classificadores utilizados.

Classificador	Descrição
RL	Calcula a probabilidade estimada de uma instância pertencer a certa classe, ou seja, de acordo com os valores de todos os atributos, se a probabilidade estimada de uma instância for superior a 50%, ela será classificada como positiva “1”. Caso contrário, será negativa “0” (Géron, 2019).
KNN	Utiliza a técnica de memorização. Ele faz análise com base nas características de instâncias passadas, encontra as similaridades e com isso gera uma resposta com base na média dos valores vizinhos. O KNN utiliza a medida de distância euclidiana para calcular os vizinhos mais próximos (Mueller and Massaron, 2019).
SVM-Linear/RBF	O método usa uma linha divisória com características claras e definidas. A estratégia da SVM é examinar a maior margem de separação entre as classes “Positivo” e “Negativo”. Ela extrai uma função de classificação de uma amostra. Uma SVM não é influenciada por pontos distantes (<i>outliers</i>) (Mueller and Massaron, 2019). Uma SVM pode trabalhar com diferentes funções de núcleo, por exemplo Linear e RBF (<i>Radial Basis Function</i>).
PG	Esse algoritmo utiliza a estatística bayesiana. O PG seleciona instâncias de maneira aleatória e realiza previsões sobre os dados, então analisa a necessidade de ajustar os dados ao modelo. Subsequentemente faz uma distribuição normal para encontrar a média e o desvio padrão. Por último o PG calcula se é necessário reajustar os dados, esse processo é repetido até a média alcançar um valor mais próximo de zero (Santos et al., 2022).
AD	É um algoritmo simples e versátil de AM. Ele consiste em nós de decisão que se relacionam uns com os outros, seguindo uma hierarquia (raiz e folhas). Em cada nó é tomada uma decisão, podendo ir para esquerda, ou então para a direita da árvore. Cada característica do <i>dataset</i> é um nó, a característica mais importante é o nó raiz e os nós-folha são o resultado da classificação, que pode assumir os valores de “1” para positivo e “0” para negativo (Géron, 2019).
FA	Ele se baseia em respostas agregadas, ou seja, uma floresta aleatória consiste de várias árvores de decisão individuais, cada uma de um subconjunto diferente de dados do conjunto de treino. Então, com os resultados de todas essas árvores, a floresta aleatória faz a classificação. Assim as FAs têm uma probabilidade maior de obter uma melhor acurácia (Géron, 2019).
MLP	Esse algoritmo consiste de uma camada de entrada, no mínimo uma camada oculta e apenas uma camada final. Com exceção da camada de saída, toda camada possui um neurônio conectado à próxima camada. A qualquer instância de treino, o algoritmo calcula a saída de cada neurônio em cada camada de forma consecutiva, então ele mede o erro de saída e calcula o peso de cada neurônio e o erro, até a última camada oculta. Na sequência o algoritmo mede os pesos e erros de cada neurônio da camada oculta anterior até alcançar a camada de entrada (Géron, 2019).
AdaBoost	Esse algoritmo utiliza o método de <i>Boosting</i> que consiste em uma combinação de previsores fracos, com a finalidade de construir um previsor mais forte. Assim sendo, de maneira sequencial cada previsor busca corrigir as deficiências de seu antecessor. O <i>Adaptive Boosting</i> (<i>AdaBoost</i>) foca em instâncias que o seu antecessor subajustou (Géron, 2019).
NB	Esse algoritmo funciona com cálculos de probabilidade, fazendo uso do teorema de <i>bayes</i> e assume que cada característica é independente. Ele vai aprendendo as probabilidades baseado nas características e classificando a qual classe pertence, “Positiva” ou “Negativa”. Dessa forma, gera uma resposta com a classe que teve a probabilidade mais alta.
QDA	Esse algoritmo trabalha com a estimação que cada classe (Positiva ou Negativa) tem um vetor média e uma matriz de covariâncias próprios, o que resulta em funções de decisão quadráticas (FISHER, 1936).

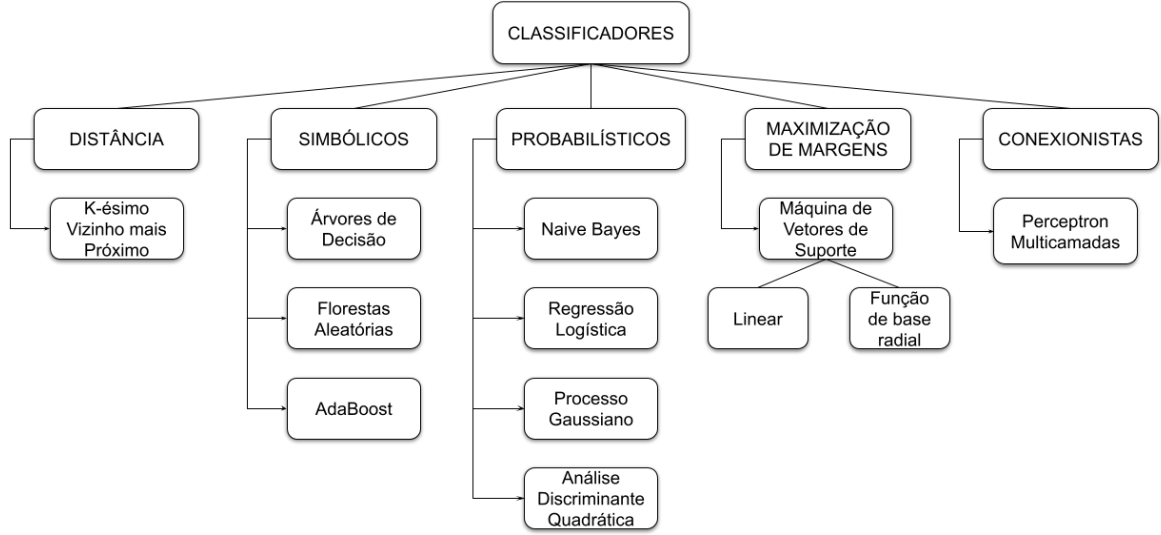


Figura 1: Classificadores Utilizados. Fonte: o autor.

3.6 Métricas de Avaliação

Neste trabalho foram utilizadas duas métricas para avaliar o desempenho dos modelos. Essas métricas são acurácia e F_1 -score. Tais métricas são detalhadas nas Subseções 3.6.1 e 3.6.2.

As métricas consideradas utilizam as seguintes medições:

- VP são as instâncias classificadas como *Verdadeiros Positivos*;
- VN são as instâncias classificadas como *Verdadeiros Negativos*;
- FP são as instâncias classificadas como *Falsos Positivos*; e
- FN são as instâncias classificadas como *Falsos Negativos*.

3.6.1 Métrica Acurácia

A acurácia do diagnóstico é dada pelo número de exemplos classificados corretamente dividido pelo número total de exemplos classificados e é calculada conforme a Equação 1 (Shaikh et al., 2021).

$$Acuracia = \frac{VP + VN}{VP + VN + FP + FN} \quad (1)$$

3.6.2 Métrica F_1 -score

A métrica F_1 -score combina as métricas de precisão (Equação 2) e sensibilidade (Equação 3) em uma única métrica. A Equação 4 define a métrica F_1 (Shalev-Shwartz and Ben-David, 2014).

$$Precisao = \frac{VP}{VP + FP} \quad (2)$$

$$Sensibilidade = \frac{VP}{VP + FN} \quad (3)$$

$$F_1 = 2 \times \frac{Precisao \times Sensibilidade}{Precisao + Sensibilidade} = \frac{VP}{VP + \frac{FN+FP}{2}} \quad (4)$$

3.7 Dataset Utilizado

Os dados utilizados neste trabalho são do *dataset* “Early stage diabetes risk prediction” (previsão de risco de diabetes em estágio inicial)³ e foram coletados utilizando questionários de pacientes do *Sylhet Diabetes Hospital* em Sylhet, Bangladesh. O *dataset* é público e é composto por 520 instâncias e 17 atributos, dos quais continham dados que informavam a idade, sexo e se o paciente possuía poliúria, polidipsia, perda repentina de peso, fraqueza, polifagia, aftas genitais, desfoque visual, coceira, irritabilidade, cicatrização retardada, paresia parcial, rigidez muscular, alopecia, obesidade e por último como era classificado. A classificação pode ser positiva, caso o paciente tenha a doença, ou negativa, caso contrário. A Tabela 3.7 apresenta os atributos e valores do *dataset*.

Tabela 2: Atributos e Valores do *Dataset* utilizado.

Atributos	Valores
Idade	16 a 90 anos
Gênero	Feminino: 0 e Masculino: 1
Poliúria	Não: 0 e Sim: 1
Polidipsia	Não: 0 e Sim: 1
Perda de peso repentina	Não: 0 e Sim: 1
Fraqueza	Não: 0 e Sim: 1
Polifagia	Não: 0 e Sim: 1
Aftas genitais	Não: 0 e Sim: 1
Desfoque visual	Não: 0 e Sim: 1
Coceira	Não: 0 e Sim: 1
Irritabilidade	Não: 0 e Sim: 1
Cicatrização retardada	Não: 0 e Sim: 1
Paresia parcial	Não: 0 e Sim: 1
Rigidez muscular	Não: 0 e Sim: 1
Alopecia	Não: 0 e Sim: 1
Obesidade	Não: 0 e Sim: 1
Classe	Negativo: 0 e Positivo: 1

3.8 Algoritmo Desenvolvido

A Figura 2 mostra o fluxograma do algoritmo proposto.

As fases do algoritmo são as seguintes:

- Importação de Dados

O algoritmo inicia com esta etapa, que faz a importação do *dataset* utilizado neste trabalho, por meio de um comando da biblioteca Pandas⁴.

- Conversão de Dados

Já entrando no pré processamento de dados, nesta etapa foram feitas as conversões dos tipos de dados, deixando o *dataset* totalmente binário, conforme mencionado na Subseção 3.3.1.

- Seleção de Características

Nesta etapa, ainda na fase de pré processamento, foram implementadas as técnicas de seleção de características citadas na subseção 3.3.2. O *dataset* foi dividido em duas partes, sendo a primeira parte chamada de “atributos” onde estão contidas todas as características, com exceção da classificação. A outra parte chamada “rótulos” que contém somente a característica “Classe” que é como o paciente foi classificado. Ocorre outra divisão subsequentemente, pois é preciso separar o *dataset*, em conjunto de treino e teste. Então, foram reservados 67% do conjunto para treinamento e os outros 33% para testes do modelo. Os métodos de seleção de características RFE e *SelectKBest(chi2)* foram utilizados, sendo que conforme os resultados obtidos, a cada iteração era refinado o número alvo de características e analisados os resultados.

³Disponível em: <https://archive.ics.uci.edu/ml/datasets/Early+stage+diabetes+risk+prediction+dataset>.

⁴<https://pandas.pydata.org/>

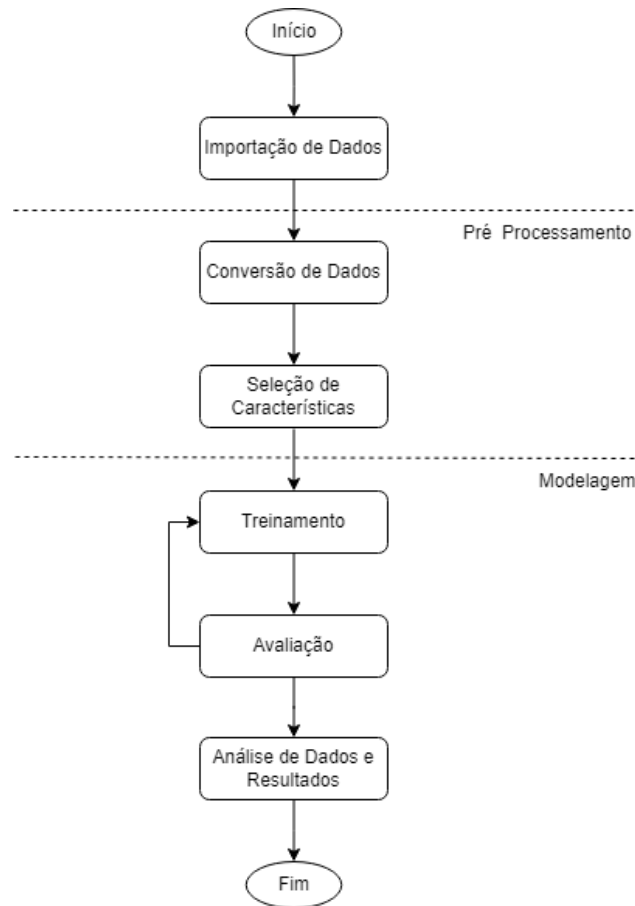


Figura 2: Fluxograma do algoritmo desenvolvido. Fonte: o autor.

- Treinamento do Modelo

Nesta etapa, a primeira da modelagem, é onde os algoritmos foram treinados, utilizando os classificadores citados na subseção 3.4. Eles aprendem conforme os dados que recebem e a sua eficiência está fortemente atrelada ao seu treinamento. Nesta fase foi utilizada a validação cruzada com subdivisões da base de três (*3-fold*), cinco (*5-fold*) e dez (*10-fold*).

- Validação do Modelo

Nesta etapa é avaliada a precisão da predição do modelo. A fim de obter resultados mais precisos foi utilizada a validação cruzada com as medidas de acurácia e F_1 -score.

- Análise de Dados e Resultados

Nesta etapa foi feita a análise dos resultados dos modelos, com auxílio de tabelas.

A Tabela 3 apresenta os parâmetros utilizados em cada classificador. AdaBoost, NB e QDA não possuem parâmetros a alterar.

4 RESULTADOS E DISCUSSÃO

Nesta seção são apresentados os resultados para os 11 algoritmos testados: Regressão Logística (RL), K-ésimo Vizinho mais Próximo (KNN), Máquinas de Vetores de Suporte (SVM) núcleo Linear e RBF, Processo Gaussiano (PG), Árvores de Decisão (AD), Florestas Aleatórias (FA), *Perceptron* Multicamadas (MLP), *AdaBoost*, *Naive Bayes* (NB) e Análise Discriminante Quadrática (QDA).

Na Seção 4.1 são apresentados os resultados dos dois métodos de seleção de características utilizando *Holdout* em termo de acurácia.

Tabela 3: Parâmetros utilizados neste trabalho.

Classificador	Parâmetro
RL	random_state=42, max_iter=500, n_jobs=-1
KNN	n_neighbors=1, n_jobs=-1
SVM - Linear	random_state=43, kernel='linear'
SVM - RBF	gamma=0.01, C=100
PG	1.0 * RBF(1.0)
AD	max_depth=5, random_state=43
FA	max_depth=13, n_estimators=100, max_features=2, random_state=43, n_jobs=-1
MLP	alpha=1, max_iter=40000
AdaBoost	-
NB	-
QDA	-

Na Seção 4.2 são mostrados os resultados utilizando o melhor método de seleção de características, com validação cruzada 10-*fold* em termos de acurácia e F_1 -score.

4.1 Resultados dos Métodos de Seleção de Características utilizando *Holdout*

A seleção de características utilizando a técnica de RFE obteve melhores resultados, comparado ao *SelectKBest*(chi2). Os resultados obtidos, em termos de acurácia são apresentados nas Tabelas 4 e 5. Nas tabelas são apresentados os classificadores utilizados, a coluna *Original* apresenta os resultados sem a seleção de características com divisão de base de 67% de treinamento e 33% para teste, as colunas seguintes apresentam os resultados com as c características mais importantes. Os resultados com $c \leq 6$ foram removidos da tabela devido a questão de espaço e considerando que obtiveram uma piora nos resultados.

Tabela 4: Tabela de resultados em termos de acurácia utilizando o *SelectKBest*(chi2). Os melhores resultados estão destacados em uma escala de cinza, sendo o mais escuro o melhor.

Classificador	Original	c = 15	c = 14	c = 13	c = 12	c = 11	c = 10	c = 9	c = 8	c = 7
RL	0.936	0.941	0.912	0.930	0.924	0.924	0.918	0.918	0.906	0.906
KNN	0.918	0.889	0.906	0.895	0.901	0.924	0.912	0.936	0.959	0.959
SVM Linear	0.936	0.936	0.912	0.924	0.906	0.930	0.924	0.901	0.895	0.906
SVM RBF	0.959	0.953	0.941	0.947	0.959	0.941	0.941	0.936	0.930	0.906
PG	0.970	0.965	0.947	0.953	0.959	0.936	0.941	0.965	0.953	0.941
AD	0.924	0.953	0.953	0.953	0.936	0.936	0.912	0.936	0.924	0.924
FA	0.976	0.947	0.953	0.953	0.941	0.947	0.953	0.953	0.936	0.924
MLP	0.936	0.941	0.924	0.924	0.924	0.918	0.924	0.912	0.912	0.901
AdaBoost	0.941	0.941	0.906	0.936	0.924	0.918	0.918	0.918	0.924	0.924
NB	0.918	0.918	0.918	0.918	0.918	0.912	0.918	0.901	0.906	0.901
QDA	0.936	0.936	0.918	0.883	0.912	0.936	0.930	0.930	0.901	0.906

Neste trabalho a aplicação de seleção de características conseguiu melhorar o desempenho de alguns classificadores, principalmente o KNN. O RFE foi a técnica que mais contribuiu para eficácia dos modelos. Contudo, em alguns classificadores, a seleção de características não apresentou melhora no desempenho, como foi o caso do classificador Florestas Aleatórias (FA), que obteve a melhor acurácia dentre todos e sem fazer uso da seleção de características.

Comparando os dois métodos de seleção de características usando *Holdout*, pode-se concluir que, para este problema, a RFE foi a melhor técnica. No entanto, o algoritmo original ainda obteve melhores resultados em geral.

Na próxima subseção o método RFE foi testado utilizando validação cruzada.

4.2 Resultados com a Validação Cruzada

Nas Tabelas 6 e 7 são apresentados os resultados em termos de acurácia e F_1 -score, respectivamente. Elas utilizam a seleção de característica RFE e validação cruzada com $K = 10$, 10-*fold*. Na primeira coluna são

Tabela 5: Tabela de resultados em termos de acurácia utilizando o RFE. Os melhores resultados estão destacados em uma escala de cinza, sendo o mais escuro o melhor.

Classificador	Original	c = 15	c = 14	c = 13	c = 12	c = 11	c = 10	c = 9	c = 8	c = 7
RL	0.936	0.936	0.941	0.941	0.930	0.930	0.941	0.930	0.924	0.901
KNN	0.918	0.965	0.965	0.965	0.970	0.970	0.965	0.970	0.970	0.941
SVM Linear	0.936	0.947	0.936	0.941	0.936	0.936	0.941	0.941	0.930	0.889
SVM RBF	0.959	0.965	0.953	0.959	0.959	0.947	0.941	0.947	0.936	0.889
PG	0.970	0.970	0.970	0.965	0.965	0.959	0.965	0.953	0.947	0.912
AD	0.924	0.959	0.959	0.959	0.953	0.959	0.959	0.959	0.953	0.924
FA	0.994	0.970	0.976	0.970	0.976	0.965	0.970	0.953	0.953	0.930
MLP	0.936	0.965	0.959	0.959	0.947	0.947	0.947	0.947	0.930	0.901
AdaBoost	0.941	0.889	0.889	0.906	0.866	0.877	0.877	0.877	0.912	0.912
NB	0.918	0.924	0.924	0.912	0.924	0.918	0.918	0.889	0.877	0.906
QDA	0.936	0.930	0.918	0.895	0.906	0.906	0.906	0.901	0.918	0.912

apresentados os classificadores, a coluna “K = 10”, mostra o resultados sem seleção de características utilizando avaliação com validação cruzada. As colunas seguintes apresentam os resultados utilizando a seleção de características com as c características mais importantes. Os resultados com $c \leq 6$ foram removidos da tabela devido a questão de espaço e considerando que obtiveram uma piora nos resultados.

No Apêndice A são apresentados os testes realizados com validação cruzada com $K = 5$, 5-fold e com $K = 3$, 3-fold. Contudo, com o $K = 10$, 10-fold foram obtidos melhores resultados.

Tabela 6: Tabela de resultados utilizando o RFE e a validação cruzada 10-fold em termos de acurácia. Os melhores resultados estão destacados em uma escala de cinza, sendo o mais escuro o melhor.

Classificador	K = 10	c = 15	c = 14	c = 13	c = 12	c = 11	c = 10	c = 9	c = 8	c = 7
RL	0.9250	0.9250	0.9269	0.9250	0.9212	0.9173	0.9212	0.9135	0.9096	0.8885
KNN	0.9288	0.9750	0.9673	0.9596	0.9692	0.9673	0.9558	0.9423	0.9442	0.9038
SVM Linear	0.9269	0.9212	0.9269	0.9269	0.9192	0.9173	0.9154	0.9154	0.9115	0.8942
SVM RBF	0.9558	0.9308	0.9308	0.9288	0.9269	0.9308	0.9288	0.9115	0.9135	0.8942
PG	0.9731	0.9731	0.9712	0.9673	0.9635	0.9596	0.9596	0.9500	0.9423	0.9077
AD	0.9577	0.9404	0.9500	0.9481	0.9500	0.9519	0.9404	0.9481	0.9173	0.9115
FA	0.9885	0.9769	0.9769	0.9712	0.9692	0.9654	0.9596	0.9462	0.9327	0.9154
MLP	0.9250	0.9327	0.9327	0.9346	0.9404	0.9269	0.9231	0.9327	0.9212	0.8846
AdaBoost	0.9250	0.9115	0.9115	0.9077	0.8981	0.9096	0.8942	0.8981	0.9096	0.8846
NB	0.8885	0.8962	0.8981	0.8942	0.8942	0.8942	0.8885	0.8904	0.8904	0.8904
QDA	0.9404	0.9404	0.9404	0.9212	0.9212	0.9096	0.9000	0.8981	0.9058	0.9058

Analisando a Tabela 6, é possível perceber que o classificador com melhor desempenho foi o de Florestas Aleatórias (FA), seguido pelo K-ésimo Vizinho mais Próximo (KNN) e Processo Gaussiano (PG). O algoritmo FA obteve 98% de acurácia e o KNN, assim como o PG, obtiveram 97% de acurácia.

Analisando os resultados, percebe-se também a sensibilidade que o classificador KNN tem em relação a utilização da seleção de características, que com isso obteve uma melhora no seu desempenho que foi de 92% para 97%.

O classificador *Naive Bayes* obteve o pior desempenho dentre todos classificadores, com acurácia de 89%.

Os resultados apresentados na Tabela 7 demonstram que o melhor classificador continua sendo o de Florestas Aleatórias (FA) que obteve 99% na métrica de F_1 -score, seguido pelo K-ésimo Vizinho mais Próximo (KNN) e Processo Gaussiano (PG), ambos com 97%. Esses classificadores foram os mesmos que tiveram destaque na métrica de acurácia.

O KNN novamente demonstra sensibilidade a seleção de característica alcançando 97% na métrica F_1 -score e com isso têm o segundo melhor desempenho.

O classificador *Naive Bayes* teve o pior desempenho dentre todos classificadores, com o valor de 89% na métrica de F_1 -score.

Portanto, dentre todos os classificadores o de Florestas Aleatórias (FA) foi o que obteve melhor desempenho, em ambas as métricas aplicadas, com 98% de acurácia e 99% no F_1 -score, sem seleção de características e com

Tabela 7: Tabela de resultados utilizando o RFE e a validação cruzada 10-fold em termos de F_1 -score. Os melhores resultados estão destacados em uma escala de cinza, sendo o mais escuro o melhor.

Classificador	K = 10	c = 15	c = 14	c = 13	c = 12	c = 11	c = 10	c = 9	c = 8	c = 7
RL	0.9386	0.9385	0.9402	0.9385	0.9355	0.9321	0.9354	0.9287	0.9265	0.9087
KNN	0.9402	0.9796	0.9730	0.9668	0.9748	0.9733	0.9639	0.9534	0.9548	0.9225
SVM Linear	0.9397	0.9355	0.9399	0.9401	0.9331	0.9317	0.9301	0.9300	0.9286	0.9179
SVM RBF	0.9633	0.9428	0.9427	0.9416	0.9401	0.9431	0.9412	0.9268	0.9303	0.9179
PG	0.9781	0.9781	0.9765	0.9733	0.9700	0.9669	0.9670	0.9596	0.9531	0.9236
AD	0.9648	0.9504	0.9587	0.9570	0.9587	0.9605	0.9513	0.9572	0.9328	0.9251
FA	0.9907	0.9814	0.9814	0.9765	0.9749	0.9719	0.9674	0.9565	0.9457	0.9299
MLP	0.9335	0.9458	0.9427	0.9477	0.9465	0.9388	0.9401	0.9404	0.9364	0.9073
AdaBoost	0.9384	0.9272	0.9272	0.9238	0.9167	0.9253	0.9141	0.9175	0.9287	0.9089
NB	0.9097	0.9157	0.9174	0.9136	0.9143	0.9137	0.9085	0.9116	0.9116	0.9116
QDA	0.9525	0.9524	0.9525	0.9375	0.9376	0.9282	0.9212	0.9193	0.9259	0.9253

validação cruzada de 10-fold.

5 CONCLUSÕES

O objetivo deste trabalho de pesquisa foi o estudo de técnicas de Aprendizado de Máquina (AM) supervisionado a serem aplicadas em um problemas de classificação, como a previsão de riscos de diabetes em estágio inicial.

Neste trabalho foram aplicadas algumas técnicas de pré-processamento, como a seleção de características. Também onze classificadores foram testados utilizando duas métricas de desempenho.

Como seleção de características a RFE foi a melhor técnica comparada com a *SelectKBest(chi2)* utilizando *Holdout*. No entanto, o algoritmo original com modelo de Florestas Aleatórias ainda obteve melhores resultados em geral.

O modelo de Florestas Aleatórias também foi o classificador que obteve melhor desempenho utilizando validação cruzada.

Como trabalhos futuros pode-se fazer uma análise mais aprofundada com teste nos parâmetros dos modelos. Pode-se estudar outras técnicas de seleção de características e também aprofundar-se nos métodos de pré-processamento. Além disso, pode-se realizar testes com outros classificadores.

Referências

- de Saúde, S. (2022). Diabetes (diabetes mellitus). <https://www.saude.pr.gov.br/Pagina/Diabetes-diabetes-mellitus>. Accessed: 2022-08-16.
- Faceli, K., Lorena, A. C., Gama, J., and Carvalho, A. C. P. d. L. F. d. (2021). *Inteligência artificial: uma abordagem de aprendizado de máquina*. LTC, 2nd edition.
- Ferreira, E. J. and de Castro Jorge, L. A. (2007). Seleção de Características Aplicada ao Processamento de Imagens Digitais. Technical report, Embrapa Instrumentação Agropecuária, São Carlos.
- FISHER, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.
- Géron, A. (2019). *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow*. Alta Books.
- Mueller, J. and Massaron, L. (2019). *Aprendizado de Máquina Para Leigos*. Para Leigos. Alta Books.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

- Santos, D. R. d., Rosario, E. S. d., Leandro, J., Kobashikawa, M. M., Sergio, M. N., Nietto, P. R., Ferreira, C. P., and Wagner, M. S. (2022). Modelo preditivo para identificar pessoas infectadas com sars-cov-2 através de exames de sangue. Trabalho de Conclusão de Curso, Engenharia da Computação, Universidade Anhembi Morumbi.
- Shaikh, K., Krishnan, S., and Thanki, R. (2021). Breast cancer detection and diagnosis using ai. In *Artificial Intelligence in Breast Cancer Early Detection and Diagnosis*, pages 79–92. Springer.
- Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press.

A Resultados complementares

Este apêndice apresenta os testes realizados com validação cruzada com $K = 5$ (*5-fold*) - Tabelas 8 e 9 - e com $K = 3$ (*3-fold*) - Tabelas 10 e 11. Estes resultados foram inferiores aos obtidos pelo $K = 10$ (*10-fold*). As Tabelas 12 e 13 apresentam os resultados utilizando o *SelectKBest(chi2)* como seletor de características e validação cruzada *10-fold*.

Tabela 8: Tabela de resultados utilizando o RFE e a validação cruzada *5-fold* em termos de acurácia. Os melhores resultados estão destacados em uma escala de cinza, sendo o mais escuro o melhor.

Classificador	K = 5	c = 15	c = 14	c = 13	c = 12	c = 11	c = 10	c = 9	c = 8	c = 7
RL	0.9250	0.9231	0.9250	0.9231	0.9192	0.9173	0.9173	0.9077	0.9154	0.8865
KNN	0.9269	0.9635	0.9500	0.9385	0.9500	0.9519	0.9423	0.9231	0.9250	0.9115
SVM Linear	0.9135	0.9231	0.9231	0.9173	0.9115	0.9154	0.9096	0.9096	0.9135	0.8942
SVM RBF	0.9500	0.9327	0.9308	0.9231	0.9269	0.9269	0.9192	0.9135	0.9135	0.8942
PG	0.9673	0.9654	0.9654	0.9615	0.9596	0.9519	0.9538	0.9404	0.9404	0.9096
AD	0.9423	0.9423	0.9327	0.9481	0.9423	0.9462	0.9423	0.9404	0.9231	0.9096
FA	0.9827	0.9654	0.9712	0.9615	0.9654	0.9615	0.9577	0.9500	0.9365	0.9250
MLP	0.9038	0.9154	0.9154	0.9173	0.9173	0.9096	0.9096	0.8981	0.9038	0.8885
AdaBoost	0.9327	0.9096	0.9115	0.9173	0.9038	0.9058	0.9096	0.8942	0.9115	0.8808
NB	0.8962	0.8981	0.8942	0.8962	0.8923	0.8962	0.8865	0.8923	0.8923	0.8923
QDA	0.9404	0.9365	0.9346	0.9135	0.9212	0.9038	0.8942	0.8942	0.9115	0.9058

Tabela 9: Tabela de resultados utilizando o RFE e a validação cruzada *5-fold* em termos de F_1 -score. Os melhores resultados estão destacados em uma escala de cinza, sendo o mais escuro o melhor.

Classificador	K = 5	c = 15	c = 14	c = 13	c = 12	c = 11	c = 10	c = 9	c = 8	c = 7
RL	0.9388	0.9369	0.9386	0.9371	0.9340	0.9323	0.9328	0.9244	0.9316	0.9076
KNN	0.9388	0.9695	0.9583	0.9489	0.9585	0.9601	0.9516	0.9358	0.9376	0.9259
SVM Linear	0.9289	0.9372	0.9372	0.9328	0.9278	0.9306	0.9260	0.9260	0.9306	0.9178
SVM RBF	0.9585	0.9449	0.9434	0.9374	0.9401	0.9400	0.9340	0.9287	0.9304	0.9178
PG	0.9736	0.9717	0.9721	0.9689	0.9673	0.9609	0.9625	0.9516	0.9513	0.9249
AD	0.9512	0.9520	0.9446	0.9571	0.9522	0.9557	0.9526	0.9510	0.9370	0.9242
FA	0.9861	0.9721	0.9768	0.9689	0.9722	0.9689	0.9577	0.9598	0.9489	0.9372
MLP	0.9217	0.9273	0.9262	0.9308	0.9252	0.9197	0.9096	0.9235	0.9178	0.9093
AdaBoost	0.9450	0.9258	0.9274	0.9318	0.9209	0.9223	0.9096	0.9136	0.9302	0.9051
NB	0.9164	0.9178	0.9143	0.9153	0.9131	0.9157	0.9075	0.9134	0.9134	0.9134
QDA	0.9526	0.9491	0.9482	0.9321	0.9374	0.9235	0.9161	0.9165	0.9302	0.9253

Tabela 10: Tabela de resultados utilizando o RFE e a validação cruzada *3-fold* em termos de acurácia. Os melhores resultados estão destacados em uma escala de cinza, sendo o mais escuro o melhor.

Classificador	K = 3	c = 15	c = 14	c = 13	c = 12	c = 11	c = 10	c = 9	c = 8	c = 7
RL	0.9289	0.9250	0.9270	0.9270	0.9193	0.9173	0.9231	0.9250	0.8981	0.8866
KNN	0.9154	0.9750	0.9635	0.9539	0.9539	0.9539	0.9501	0.9424	0.9347	0.8981
SVM Linear	0.9289	0.9231	0.9269	0.9308	0.9250	0.9327	0.9269	0.9231	0.9077	0.8789
SVM RBF	0.9577	0.9385	0.9385	0.9366	0.9347	0.9366	0.9289	0.9289	0.9231	0.8827
PG	0.9616	0.9692	0.9654	0.9519	0.9615	0.9481	0.9443	0.9423	0.9423	0.8981
AD	0.9288	0.9385	0.9385	0.9481	0.9404	0.9385	0.9366	0.9385	0.9251	0.9115
FA	0.9712	0.9692	0.9731	0.9635	0.9673	0.9596	0.9558	0.9442	0.9385	0.9231
MLP	0.9289	0.9424	0.9404	0.9443	0.9385	0.9308	0.9328	0.9289	0.9154	0.8866
AdaBoost	0.9231	0.9058	0.9115	0.9115	0.8981	0.8981	0.8981	0.8962	0.9058	0.8846
NB	0.8885	0.8923	0.8942	0.8885	0.8904	0.8962	0.8942	0.8846	0.8846	0.8865
QDA	0.9327	0.9308	0.9173	0.9039	0.9135	0.9077	0.8981	0.9000	0.9057	0.8981

Tabela 11: Tabela de resultados utilizando o RFE e a validação cruzada 3-fold em termos de F_1 -score. Os melhores resultados estão destacados em uma escala de cinza, sendo o mais escuro o melhor.

Classificador	K = 3	c = 15	c = 14	c = 13	c = 12	c = 11	c = 10	c = 9	c = 8	c = 7
RL	0.9419	0.9387	0.9403	0.9403	0.9342	0.9326	0.9372	0.9389	0.9174	0.9071
KNN	0.9304	0.9796	0.9702	0.9622	0.9624	0.9623	0.9588	0.9526	0.9456	0.9164
SVM Linear	0.9418	0.9371	0.9403	0.9438	0.9388	0.9447	0.9400	0.9369	0.9255	0.9039
SVM RBF	0.9652	0.9496	0.9498	0.9481	0.9464	0.9477	0.9422	0.9416	0.9370	0.9070
PG	0.9688	0.9750	0.9721	0.9612	0.9690	0.9581	0.9548	0.9531	0.9530	0.9168
AD	0.9404	0.9498	0.9498	0.9575	0.9515	0.9499	0.9480	0.9496	0.9372	0.9264
FA	0.9768	0.9753	0.9784	0.9706	0.9737	0.9674	0.9643	0.9551	0.9504	0.9361
MLP	0.9484	0.9591	0.9528	0.9469	0.9497	0.9436	0.9451	0.9423	0.9251	0.9071
AdaBoost	0.9371	0.9242	0.9284	0.9284	0.9179	0.9184	0.9184	0.9163	0.9250	0.9084
NB	0.9089	0.9118	0.9135	0.9086	0.9104	0.9152	0.9136	0.9057	0.9057	0.9071
QDA	0.9469	0.9453	0.9354	0.9250	0.9319	0.9263	0.9202	0.9213	0.9262	0.9201

Tabela 12: Tabela de resultados utilizando o *SelectKBest*(chi2) e a validação cruzada 10-fold em termos de acurácia. Os melhores resultados estão destacados em uma escala de cinza, sendo o mais escuro o melhor.

Classificador	K = 10	c = 15	c = 14	c = 13	c = 12	c = 11	c = 10	c = 9	c = 8	c = 7
RL	0.9250	0.9231	0.9173	0.9173	0.9154	0.9038	0.9115	0.9135	0.8904	0.8981
KNN	0.9288	0.9404	0.9308	0.9288	0.9231	0.9288	0.9308	0.9327	0.9308	0.9231
SVM Linear	0.9269	0.9288	0.9077	0.9173	0.9154	0.9000	0.9058	0.9096	0.8846	0.8846
SVM RBF	0.9558	0.9519	0.9481	0.9538	0.9500	0.9423	0.9404	0.9423	0.9019	0.8942
PG	0.9731	0.9692	0.9615	0.9577	0.9596	0.9596	0.9596	0.9538	0.9385	0.9212
AD	0.9577	0.9462	0.9519	0.9481	0.9385	0.9231	0.9269	0.9346	0.8981	0.9019
FA	0.9885	0.9885	0.9865	0.9827	0.9769	0.9750	0.9731	0.9692	0.9404	0.9288
MLP	0.9250	0.9192	0.9212	0.9135	0.9077	0.8942	0.8962	0.8962	0.9019	0.9000
AdaBoost	0.9250	0.9269	0.9115	0.9077	0.9115	0.9173	0.9192	0.9135	0.8981	0.9019
NB	0.8885	0.8942	0.8942	0.8942	0.8962	0.8923	0.9000	0.8846	0.8885	0.8942
QDA	0.9404	0.9404	0.9442	0.9327	0.9269	0.9135	0.9154	0.9000	0.8962	0.8942

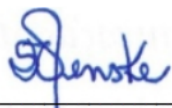
Tabela 13: Tabela de resultados utilizando o *SelectKBest*(chi2) e a validação cruzada 10-fold em termos de F_1 -score. Os melhores resultados estão destacados em uma escala de cinza, sendo o mais escuro o melhor.

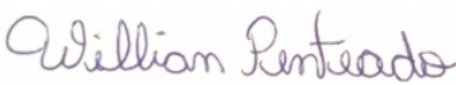
Classificador	K = 10	c = 15	c = 14	c = 13	c = 12	c = 11	c = 10	c = 9	c = 8	c = 7
RL	0.9386	0.9371	0.9320	0.9325	0.9308	0.9220	0.9278	0.9295	0.9113	0.9182
KNN	0.9402	0.9498	0.9414	0.9398	0.9349	0.9407	0.9422	0.9443	0.9431	0.9368
SVM Linear	0.9397	0.9413	0.9251	0.9326	0.9314	0.9196	0.9244	0.9276	0.9085	0.9086
SVM RBF	0.9633	0.9602	0.9572	0.9621	0.9589	0.9527	0.9508	0.9526	0.9219	0.9157
PG	0.9781	0.9748	0.9685	0.9654	0.9670	0.9670	0.9668	0.9623	0.9498	0.9354
AD	0.9648	0.9555	0.9604	0.9568	0.9490	0.9364	0.9397	0.9465	0.9171	0.9200
FA	0.9907	0.9907	0.9891	0.9860	0.9811	0.9796	0.9782	0.9750	0.9517	0.9425
MLP	0.9335	0.9357	0.9283	0.9273	0.9174	0.9167	0.9199	0.9139	0.9207	0.9177
AdaBoost	0.9384	0.9398	0.9270	0.9234	0.9274	0.9327	0.9342	0.9295	0.9187	0.9220
NB	0.9097	0.9143	0.9143	0.9143	0.9150	0.9127	0.9187	0.9064	0.9098	0.9152
QDA	0.9525	0.9525	0.9556	0.9467	0.9421	0.9317	0.9331	0.9217	0.9185	0.9168

B AVALIAÇÃO DO ORIENTADOR SOBRE O DESEMPENHO DO ORIENTADO

O acadêmico Willian Penteadó realizou todas as atividades previstas com dedicação e cumpriu todos os objetivos do projeto de iniciação científica proposto. Compareceu às reuniões semanais presenciais assídua e pontualmente, obtendo um ótimo desempenho.

Guarapuava, 28 de setembro de 2022.


Assinatura do Orientador


Assinatura do Aluno