UNIVERSIDADE FEDERAL DO PAMPA

Victor Hugo Schneider Lopes

Análise Comparativa do Pré-Processamento de Dados na Classificação de Sementes

Alegrete
2024

Victor Hugo Schneider Lopes

Análise Comparativa do Pré-Processamento de Dados na Classificação de Sementes

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Alessandro Bof de Oliveira

Alegrete
2024

VICTOR HUGO SCHNEIDER LOPES

ANÁLISE COMPARATIVA DO PRÉ-PROCESSAMENTO DE DADOS NA CLASSIFICAÇÃO DE SEMENTES

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Dissertação defendida e aprovada em: 02 de julho de 2024.

Banca examinadora:

_____

Prof. Dr. Alessandro Bof de Oliveira

Orientador

UNIPAMPA

Prof. Dr. Anderson Priebe Ferrugem

UFPEL

Prof. Dr. Mauricio Braga de Paula

UFPEL

Assinado eletronicamente por **ALESSANDRO BOF DE OLIVEIRA**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 02/07/2024, às 19:27, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **Mauricio Braga de Paula**, **Usuário Externo**, em 02/07/2024, às 19:28, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **Anderson Priebe Ferrugem**, **Usuário Externo**, em 02/07/2024, às 19:28, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1475454** e o código CRC **66B12BFF**.

Este trabalho é dedicado à minha família, amigos e todos os demais que me apoiaram nesta jornada.

AGRADECIMENTOS

Em primeiro lugar eu agradeço à minha família que sempre me apoiou de todas as formas possíveis e imagináveis, frente a todas as dificuldades e desafios que encontrei pelo caminho. Sem o amor, o carinho e o apoio de vocês, eu jamais teria conseguido alcançar os meus objetivos.

Agradeço também aos meus amigos que estiveram presentes durante todo o processo de graduação, seja nos momentos de alegria, de estudo ou em diversas outras circunstâncias.

Por fim, agradeço aos meus professores e mentores, que compartilharam seu conhecimento e experiência, e que foram fundamentais para a minha formação acadêmica e profissional. Em especial, agradeço ao professor Alessandro Bof de Oliveira pelo apoio inestimável durante o desenvolvimento deste trabalho. Também gostaria de expressar minha gratidão ao professor Marcelo Caggiani Luizelli pelo suporte contínuo e por me proporcionar uma perspectiva acadêmica valiosa ao longo dessa jornada.

“Ideias, e somente ideias, podem iluminar a escuridão.”
Ludwig Von Mises

RESUMO

Definir o tipo e a qualidade de uma semente antes do seu plantio é uma tarefa vital para a colheita, uma vez que o uso de sementes de baixa qualidade pode resultar em baixa produtividade, mesmo em condições de cultivo favoráveis. Essa tarefa tem sido realizada de maneira manual, tornando-a árdua, demorada e propensa a erros. Técnicas de aprendizado de máquina podem solucionar esse problema ao utilizar algoritmos de classificação para rotular sementes em diferentes classes, reduzindo a dificuldade, o tempo consumido e a quantidade de erros. No entanto, essas técnicas estão sujeitas à qualidade dos dados fornecidos a elas, como a presença de dados faltantes, *outliers* e dados não normalizados, impactando diretamente o desempenho da classificação, que geralmente é sensível à presença dessas características nos conjuntos de dados fornecidos. Para aumentar a qualidade dos dados fornecidos aos classificadores, técnicas de pré-processamento de dados podem ser aplicadas. Este trabalho investiga o impacto das técnicas de pré-processamento de dados no desempenho de modelos de aprendizado de máquina na classificação de sementes de feijão seco. Utilizando um conjunto de dados do Repositório de Aprendizado de Máquina da UCI, derivado de um experimento realizado por Koklu e Ozkan (KOKLU; OZKAN, 2020), foram aplicadas várias técnicas de pré-processamento, como imputação de valores faltantes, remoção de outliers e normalização de dados. Métodos de classificação k-Vizinhos Mais Próximos (kNN) e Perceptron Multicamadas (MLP) foram usados para avaliar a eficácia dessas técnicas de pré-processamento. Além disso, foi proposto um modelo MLP aprimorado com parâmetros otimizados, incluindo a taxa de aprendizado e a configuração das camadas ocultas. Os resultados experimentais demonstram o papel crítico do pré-processamento de dados, especialmente da normalização de dados. O modelo MLP aprimorado superou significativamente o modelo de base, destacando a importância da utilização de um modelo de rede neural otimizado para resolver o problema.

Palavras-chave: Aprendizado de Máquina. Pré-processamento de Dados. Classificação de Sementes.

ABSTRACT

Defining the type and quality of a seed before planting is vital for the harvest, as the use of low-quality seeds can result in low productivity, even under favorable growing conditions. This task has traditionally been performed manually, making it arduous, time-consuming, and prone to errors. Machine learning techniques can solve this problem by using classification algorithms to label seeds into different classes, reducing difficulty, time consumption, and the number of errors. However, these techniques are subject to the quality of the data provided to them, such as the presence of missing data, outliers, and unnormalized data, which directly impact the performance of the classification, as it is generally sensitive to these characteristics in the datasets provided. To increase the quality of the data provided to the classifiers, data preprocessing techniques can be applied. This paper investigates the impact of data preprocessing techniques on the performance of machine learning models in the classification of dry bean seeds. Using a dataset from the UCI Machine Learning Repository, derived from an experiment by Koklu and Ozkan (KOKLU; OZKAN, 2020), various preprocessing techniques, such as missing value imputation, outlier removal, and data normalization, were applied. k-Nearest Neighbors (kNN) and Multi-Layer Perceptron (MLP) classification methods were used to evaluate the effectiveness of these preprocessing techniques. Additionally, an enhanced MLP model with optimized parameters, including the learning rate and hidden layer configuration, was proposed. The experimental results demonstrate the critical role of data preprocessing, especially data normalization. The enhanced MLP model significantly outperformed the baseline model, highlighting the importance of using an optimized neural network model to solve the problem.

Key-words: Machine Learning. Data Preprocessing. Seed Classification.

LISTA DE FIGURAS

LISTA DE TABELAS

UCI Universidade da Califórnia, Irvine

ML Aprendizado de Máquinas

k**NN** k Vizinhos Mais Próximos

SVM Máquina de Vetores de Suporte

MLP Perceptron Multicamadas

CNN Rede Neural Convolucional

e.g. por exemplo

i.e. isto é

MAD Desvio Mediano Absoluto

std Desvio Padrão

k**NN-I** Imputor k Vizinhos Mais Próximos

IL Interpolação Linear

TAI Taxa de Aprendizado Inicial

NME Número Máximo de Épocas

NPC Neurônios da Primeira Camada

NSC Neurônios da Segunda Camada

TP Positivo Verdadeiro

FP Falso Positivo

FN Falso Negativo

TN Verdadeiro Negativo

SUMÁRIO

1 INTRODUÇÃO

Em 2018, cerca de 2,5 quintilhões de *bytes* de dados foram produzidos diariamente e a quantidade de dados produzida deve dobrar a cada ano (HARIRI; FREDERICKS; BOWERS, 2019). Essa tendência é impulsionada por diversas fontes como, por exemplo: (i) dispositivos móveis, (ii) *e-commerce*, (iii) redes sociais e (iv) transações online (SADINENI, 2020). Esse avanço exponencial da tecnologia deu origem a uma era de dados massivos, onde informações valiosas são encontradas em conjuntos de dados cada vez maiores e mais complexos. Essa proliferação de informações desafia não apenas nossa capacidade de armazenamento e processamento, mas também a nossa capacidade de extrair conhecimento desses dados (MANYIKA et al., 2011).

A análise de dados tornou-se uma ferramenta indispensável nesse cenário de abundância de informações. À medida que os dados continuam a se acumular, a capacidade de filtrar, organizar e extrair conhecimento valioso se torna vital. Ela desempenha um papel fundamental em diversas áreas, como Saúde (BENHAR; IDRI; FERNÁNDEZ-ALEMÁN, 2020), Economia (DHARMA et al., 2020), Sismologia (XIE et al., 2020), Agricultura (KOKLU; OZKAN, 2020), entre outras. Em resposta à esse crescimento no volume de dados, técnicas mais avançadas, como as técnicas baseadas em aprendizados de máquina (*machine learning* ou ML), são necessárias para que informações úteis possam ser extraídas. Técnicas de ML, entretanto, são bastante sensíveis à qualidade dos dados que são fornecidos (BENHAR; IDRI; FERNÁNDEZ-ALEMÁN, 2020) e, apesar da quantidade de dados disponíveis ser imensa, a qualidade dos mesmos pode não ser ótima. Isso pode ocorrer por diferentes razões, como: (i) erro durante a coleta de dados em uma pesquisa, (ii) falha mecânica durante a captura de informações, ou por (iii) armazenamento incorreto, visto que muitos dos conjuntos de dados encontrados podem conter erros como dados faltantes, não normalizados ou discrepantes (LITTLE; RUBIN, 2019). Cerca de 80% do trabalho de análise de dados é usado apenas no tratamento dos dados crus, à fim de torná-los interpretáveis (JAMSHED et al., 2019). Algumas das tarefas do pré-processamento de dados são:

- **Tratamento de dados faltantes**: Encontrar e tratar dados faltantes por meio de técnicas de imputação como *k-Nearest Neightbors* e Interpolação Linear, por exemplo.
- **Detecção de valores discrepantes (*Outliers*)**: Encontrar e remover dados que se diferenciam muito do restante do conjunto de dados. Técnicas como 3 sigmas e intervalo de confiança, por exemplo, são muito utilizadas para realizar a detecção de *Outliers*.
- **Normalização de dados**: Padronizar as variáveis de um conjunto de dados para que todas tenham uma mesma escala. Técnicas como *min-max* e *z-score*, por exemplo, são amplamente utilizadas para essa finalidade.

Uma área que depende de análise de um grande volume de dados é a Agriltura, em especial a classificação de sementes (MACUÁCUA; CENTENO; AMISSE, 2023). A classificação de sementes, em especial a classificação de sementes de feijões, é fundamental para garantir que as sementes utilizadas garantam um alto nível de produtividade (KOKLU; OZKAN, 2020). Essa classificação pode ser realizada manualmente, mas é um processo demorado e propenso a erros e a automação desse processo por meio de modelos de técnicas baseadas em ML pode ajudar a aumentar a eficiência e a precisão da classificação de sementes (SARIJALOO et al., 2021).

Apesar de diversos avanços recentes na área de classificação de sementes (HAMID et al., 2022), o impacto dessas técnicas de pré-processamento de dados ainda é pouco discutido. A ideia de que técnicas de pré-processamento de dados afetam positivamente o desempenho de técnicas de ML é amplamente aceita, porém um baixo número de estudos foram feitos para medir essee impacto (ZELAYA, 2019). O presente trabalho se propõe a analisar o impacto do pré-processamento em modelos baseados em ML, investigando a importância da qualidade dos dados para a eficiência desses métodos.

1.1 Objetivos

1.1.1 Objetivos Gerais

O principal objetivo deste trabalho consiste em realizar uma investigação aprofundada sobre a influência do pré-processamento de dados nas abordagens de classificação de sementes de feijão por meio de modelos de classificação baseados em ML. Em particular, o impacto de técnicas de tratamento de dados faltantes, detecção de *outliers* e normalização de dados serão abordadas no processo de classificação de sementes de feijões com as técnicas baseadas em ML *k Nearest Neighbors* e rede reural MLP (*multi layer perceptron*) .

1.1.2 Objetivos específicos

- Compreender o efeito de diferentes técnicas de imputação de valores faltantes na classificação de sementes de feijões;

- Investigar o impacto do pré-processamento de dados no desempenho da classificação de sementes de feijões com o método kNN;

- Investigar o impacto do pré-processamento de dados no desempenho da classificação de sementes de feijões por uma rede neural MLP.

1.2 Organização deste trabalho

Esta seção apresenta a estrutura do documento. No Capítulo 2, será apresentada uma visão geral sobre os tópicos de interesse deste trabalho. No Capítulo 3, será feita uma revisão abrangente da literatura, focando em trabalhos recentes que abordam os conceitos de pré-processamento de dados e classificação de sementes, bem como trabalhos que visam estabeler uma relação entre o uso de técnicas de pré-processamento de dados e o desempenho da classificação de sementes. No Capítulo 4 será apresentada a metodologia utilizada para alcançar os objetivo desse trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo fornecer uma visão geral dos tópicos relevantes para o presente trabalho. Primeiramente será apresentada uma visão geral de técnicas de pré-processamento de dados, seguida por uma explicação de técnicas de classificação de dados, incluindo técnicas de aprendizado de máquina (*machine learning* ou ML) e redes neurais artificiais.

2.1 Pré-processamento de Dados

O crescimento rápido da quantidade de dados disponíveis criou um ambiente onde uma enorme quantidade de dados pode ser encontrada (HARIRI; FREDERICKS; BOWERS, 2019). Esses dados, entretanto, podem estar dispostos de maneira que dificulte seu uso. Dados faltantes, discrepantes ou duplicados são comumente encontrados nesses conjuntos de dados "sujos" (RIDZUAN; ZAINON, 2019).

Dados faltantes podem ser encontrados em diversas bases de dados de aplicações cotidianas, frequentemente levando à resultados inapropriados e reduzindo a eficácia de métodos de algoritmos de aprendizado de máquina expostos à esses dados (ZHANG et al., 2017). A Tabela 1 apresenta um exemplo de dados faltantes em um conjunto de dados.

Tabela 1 – Exemplo de conjunto de dados com valores faltantes.

Registro	Nome	Idade	Cidade
0	Derrick Bridges	26.0	Orlando
1	Tyndall Hunter	34.0	None
2	Varnan Taylor	NaN	Nova Iorque

Bases de dados podem, também, apresentar valores discrepantes (normalmente chamados de *outliers*) em seus dados. Por um lado, eles podem ser ferramentas valiosas na detecção de fraudes em cartões de crédito, sistemas de segurança e detecção de anomalias, uma vez que representam eventos raros e suspeitos (AGGARWAL; AGGARWAL, 2017). No entanto, os *outliers* também podem ter impactos negativos na análise de dados, distorcendo medidas estatísticas e afetando o desempenho de técnicas de aprendizado de máquina. Sua presença pode levar a conclusões incorretas e modelos de classificação imprecisos (SUN et al., 2018). A Tabela 2 apresenta um exemplo de conjunto de dados que possui *outliers*. Nesse caso, todas as notas possuem valores similares, com excessão do valor $26,8$ (marcado em negrito).

Tabela 2 – Exemplo de conjunto de dados que contém um *outlier*.

Registro	Matemática	Geografia	Português
0	8,5	7,9	9,6
1	7,2	7,5	8,9
2	9,1	**26,8**	7,9

Dados não normalizados também podem ser encontrados em conjuntos de dados do mundo real. Esses conjuntos possuem colunas com valores em escalas diferentes e garantir que características que valores númericos menores não tenham sua importância dominada por características com escalas de valores maiores é importante para garantir que técnicas de ML não os tratem com menor importância (SINGH; SINGH, 2020). A Tabela 3 apresenta um exemplo de conjunto com dados não normalizados. Nesse caso, a altura é uma caracterísitca que normalmente varia entre 0 e 2, enquanto Peso e Idade são características que possuem valores que podem ultrapassar uma centena. Essa diferença pode causar um viés em técnicas de ML, dando importância maior a colunas com faixas de valores maiores (NAYAK; MISRA; BEHERA, 2014).

Tabela 3 – Exemplo de dados não normalizados.

Registro	Altura	Peso	Idade
0	1,81	102,55	49
1	1,59	62,42	18
2	1,68	61,10	25

O pré-processamento de dados engloba técnicas de identificação e correção dessas inconsistências, tendo um papel crítico na garantia da qualidade e eficácia da análise de dados, aprimorando a qualidade da interpretação dos dados e aumentando o desempenho de técnicas de ML (JAMSHED et al., 2019).

2.1.1 Tratamento de Valores Faltantes

Valores faltantes são informações que estão ausentes ou incompletas em um conjunto de dados. Eles podem acontecer por várias causas, como por entrevistados que se recusam a relatar sua renda ou uma falha mecânica que impede uma máquina de reportar dados, por exemplo (LITTLE; RUBIN, 2019). A presença de dados faltantes é comum em conjuntos de dados do mundo real e pode representar um desafio significativo na análise de dados, pois a falta de informações pode afetar a qualidade e a integridade dos resultados obtidos (CHOUDHURY; PAL, 2019).

A abordagem para tratar valores faltantes em um conjunto de dados pode seguir duas estratégias distintas. Algoritmos de análise de dados encontram dificuldades ao operar em ambientes que possuam essas ocorrências, o que torna a eliminação dessas observações uma técnica útil (KNOL et al., 2010). A segunda alternativa, mais sofisticada, consiste na aplicação de técnicas de inferência de dados, que visam preencher essas lacunas por meio de métodos estatísticos ou de aprendizado de máquina. De maneira abrangente, essas técnicas (chamadas de técnicas de imputação de valores faltantes) podem ser categorizadas em dois grupos distintos, conforme explicado por (FAN et al., 2021): imputação univariada e imputação multivariada.

O primeiro grupo envolve a atribuição de valores faltantes com base apenas na informação da varíavel faltante e utiliza técnicas como imputação de valores médios e medianos de outras observações que não apresentam falta do valores desta variável, ou técnicas de substituição baseadas nos dados seguintes e anteriores (*i.e.*, *Moving Foward*, *Moving Backward*, *Moving Average*) (FAN et al., 2021). Apesar de serem técnicas úteis, não são recomendadas quando a quantidade de valores faltantes é muito alta (*i.e.*, 5-15%) (JENGHARA et al., 2018).

O grupo de métodos de imputação multivariado utiliza informações de múltiplas características simultaneamente para realizar a inferência e utiliza técnicas como *k-nearest neighbour* (KNN) e Interpolação Linear, por exemplo. Esses métodos utilizam um conjunto de um ou mais valores (chamados de independentes) para determinar o resultado de um outro valor (chamado de dependente) (DHARMA et al., 2020).

Tabela 4 – Avaliação de séries por telespectadores

Registro	GOT	Breaking Bad	The Office	The Blacklist
0	78	92	68	45
1	13	72	27	68
2	NaN	44	45	79
3	67	25	86	12
4	52	89	42	95

A técnica de imputação de valores kNN é uma abordagem de imputação multivariada que se baseia na similaridade entre observações em um espaço multidimensional. Ela consiste em calcular a semelhança entre uma observação com valor faltante e todas as outras observações que possuem valor atribuído à essa característica. Depois de definirmos a similaridade das observações, as k observações mais próximas são selecionadas, onde k é um parâmetro definido previamente. Para inferir o valor ausente, os valores da característica que possuem o valor faltante das k observações mais próximas são considerados, e isso pode ser feito de várias maneiras, como calculando a média, a mediana ou ponderando os valores (como usando Distância Euclidiana, onde observações mais próximas são consideradas mais relevantes) (SILVA; HRUSCHKA, 2009). A imputação kNN é eficaz quando existe uma relação significativa entre as observações, no entanto, a escolha cuidadosa do valor de k é fundamental, uma vez que um valor inadequado pode afetar a qualidade da imputação (ZHANG et al., 2017).

A Tabela 4 apresenta um exemplo de um conjunto de dados referente à uma pesquisa feita com 5 pessoas que assistiram as séries apresentadas na figura. Podemos ver que o terceiro respondente deixou de dar sua avaliação à série *Game of Thrones*. Para se decidir o valor aproximado que servirá como substituição para esse dado de acordo com a técnica *k-Nearest Neighbor* deveremos seguir os seguites passos:

- Considere i como número da linha e j como número da coluna. x_{ij} representa o

valor da posição ij no conjunto de dados;

- Selecionar o valor faltante (x_{ij}), nesse caso x_{20};
- Selecionar todos os outros valores da mesma observação (x_{21}, x_{22} e x_{23});
- Escolher um valor para k, ou seja, o número de vizinhos mais próximos considerados para definirmos o valor final de x_{20}.
- Calcular a distância euclidiana de x_{21}, x_{22} e x_{23} para cada outra observação y (Equação 2.1);
- Tendo a distância de x para cada outra observação, as k mais próximas são escolhidas;
- Por fim, fazer a média dos valores da coluna de x_1 de cada observação selecionada.

$$dist(x, y) = \left((x_2 - y_2)^2 + (x_3 - y_3)^2 + (x_4 - y_4)^2\right)^{1/2} \tag{2.1}$$

A técnica de imputação de valores faltantes por interpolação linear é uma abordagem que tenta preencher valores ausentes em um conjunto de dados de maneira sequencial, usando uma relação linear entre as observações próximas (de preferência vizinhas) com dados disponíveis. Isso envolve o cálculo de uma linha reta que melhor se ajusta aos pontos de dados anteriores e posteriores do valor ausente, permitindo a estimativa do valor com base na progressão linear entre as observações conhecidas (NOOR et al., 2014).

Tabela 5 – Exemplo de conjunto de dados com dados faltantes.

Registro	X	Y
0	12,5	137,5
1	14	154
2	15	165,0

Considerando a Tabela 5, para decidirmos o valor aproximado do valor faltante, os seguintes passos devem ser seguidos:

- Considere y como valor à ser descoberto na característica Y à partir dos valores da característica X e x como valor da mesma observação para característica X;
- Considere x_0 e y_0 como valores da obersavação anterior e x_1 e y_1 como valor da posterior, representando as características X e Y;
- O valor utilizado para substituição de y é dado por:

$$novo\_y = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) \tag{2.2}$$

A interpolação linear é especialmente útil quando os dados apresentam uma tendência ou padrão linear e quando a sequencialidade das observações é relevante, apesar de apresentar resultados melhores que imputações de valores faltantes univariados (*e.g.*, imputação pela média ou pela mediana), já que estes causam uma disrupção na estrutura dos dados (NOOR et al., 2015).

É viável, também, a introdução intencional de dados faltantes em um conjunto de dados que originalmente não apresenta esse tipo de ocorrência. Essa prática é frequentemente adotada com o propósito de avaliar o desempenho de algoritmos de análise de dados em situações de dados incompletos (RIEGER; HOTHORN; STROBL, 2010; KNOL et al., 2010). Ferramentas avançadas, como a biblioteca Pandas (MCKINNEY, 2010), oferecem meios para inserir deliberadamente dados ausentes em um conjunto de dados, permitindo a realização de experimentos controlados que podem revelar como os algoritmos se comportam diante dessa situação.

2.1.2 Técnicas de Detecção e Tratamento de Valores Discrepantes

Um valor discrepante (comumente chamado de *outlier*) é:

> um ponto de dados que é significativamente diferente dos outros pontos de dados, ou não se conforma ao comportamento normal esperado, ou se conforma bem a um comportamento anormal definido (CHANDOLA; BANERJEE; KUMAR, 2009).

Os *outliers* podem ser classificados em *outliers* de Ponto ou *outliers* Coletivos. O primeiro acontece quando uma instância de dados está extremamente afastada do restante do conjunto de dados. Já um *outlier* coletivo pode ser observado quando um grupo de instâncias de dados está anormal do restante do conjunto, mesmo que cada instância esteja próxima de outra dentro do mesmo grupo (BOUKERCHE; ZHENG; ALFANDI, 2020).

Técnicas de detecção de outliers desempenham um papel fundamental na identificação de observações incomuns ou discrepantes em conjuntos de dados. Duas abordagens comuns incluem o uso do método de "3 Sigma" (3σ), que se baseia no cálculo de desvio padrão, e Desvio Absoluto Mediano (*Median Absolute Deviation* ou MAD). Essas técnicas, amplamente aplicadas em diversas áreas como Medicina (YONG; WARD; BIRCH, 2008), Consumo de Energia (NASCIMENTO et al., 2021) e Agricultura (SHAHRIAR et al., 2016), por exemplo, permitem a identificação de valores atípicos que podem indicar erros, anomalias ou eventos significativos, contribuindo para a melhoria da qualidade e integridade dos dados analisados.

A técnica de detecção de *outliers* por 3 Sigma, também conhecida como regra 68-95-99.7, é uma abordagem estatística que identifica valores discrepantes em um conjunto de dados com base na premissa de que quase todos os valores de uma distribuição normal ($99,73\%$) estão em até 3 desvios padrão (*std*) da média (SHAHRIAR et al., 2016) como

mostrado na subseção 2.1.2. Sendo assim, pode-se assumir que qualquer valor que esteja além desse intervalo é considerado um possível *outlier*.

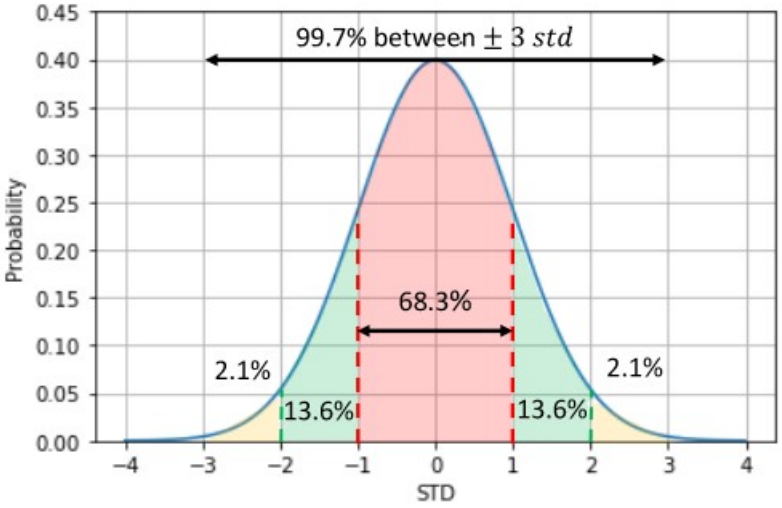


Figura 1 – Porcentagens em uma distribuição normal entre desvios padrão

Fonte: (NASCIMENTO et al., 2021)

Sendo μ a média e σ o desvio padrão de uma característica X com N observações, para descobrir se um valor x seria considerado um *outlier* de acordo com essa técnica, primeiro definimos os limites superior (l_{sup}) e inferior (l_{inf}) aceitos. Caso $x < l_{inf}$ ou $x > l_{sup}$, então dizemos que x é um *outlier*.

$$\mu = \frac{\sum_{i=0}^{N} X_i}{N} \tag{2.3}$$

$$\sigma = \sqrt{\frac{\sum_{i=0}^{N} (X_i - \mu)}{N}} \tag{2.4}$$

$$l_{sup} = \mu + 3\sigma \tag{2.5}$$

$$l_{inf} = \mu - 3\sigma \tag{2.6}$$

Apesar de ser uma abordagem simples e eficaz para identificar observações que se desviam significativamente da tendência central dos dados, essa técnica acaba por apontar apenas os casos de *outliers* mais óvios (NASCIMENTO et al., 2021). Além disso, por se utilizar da média como forma de tendência central, é uma técnica que é afetada diretamente pela presença de *outliers* (LEYS et al., 2013).

Outra técnica muito utilizada como forma de detecção de *outliers* é o Desvio Absoluto Mediano (*Mean Absolute Deviation* ou MAD). A mediana também funciona como uma medida de centralidade de uma distribuição de valores porém é muito mais resistente à presença de *outliers* (LEYS et al., 2013).

Para calular o MAD de um valor x pertencente à uma característica X com N observações, é necessário calcular a mediana da característica X(med_X, apresentada na Equação 2.7) dessa característica, seguido do cálculo da mediana das deviações absolutas em relação à mediana (MAD, apresentada na Equação 2.8) onde b é um valor constante (sugerido como 1.4826 (NASCIMENTO et al., 2021)) e i é uma observação da característica X sendo calculada.

$$med_X = \frac{N+1}{2} \tag{2.7}$$

$$MAD = b * med(|X_i - med_X|) \tag{2.8}$$

Em seguida, é necessário definir os limites inferior (l_{inf}) e superior (l_{sup}). Para isso considera-se a a soma entre a mediana e o valor de MAD dessa característica multiplicado por um valor de conservadorismo da detecção, 3 para muito conservador, 2,5 para moderadamente conservador ou até mesmo 2 para pouco conservador (MILLER, 1991). Caso $X_i < lim_{inf}$ ou $X_i > lim_{sup}$ a observação testada será considerada um *outlier*.

$$l_{inf} = med_X - 3 * MAD \tag{2.9}$$

$$l_{sup} = med_X + 3 * MAD \tag{2.10}$$

2.1.3 Normalização de dados

A normalização de dados tem como objetivo escalar os valores de variáveis de um conjunto de dados para um intervalo específico. Isso é feito para garantir que as características do conjunto tenham escalas comparáveis e para evitar distorções na análise estatística, principalmente em algoritmos de aprendizado de máquina e em redes neurais (NAYAK; MISRA; BEHERA, 2014). Diferentes técnicas podem ser utilizada para normalizar os dados, como a normalização *min-max* e a normalização *Z-Score*.

A normalização de dados pela técnica *min-max* transforma os valores dados recebidos em uma faixa pré definida (normalmente $[0, 1]$ ou $[-1, 1]$. Esse método normaliza os valores de uma característica X de acordo com os valores máximos e mínimos dessa característica. Ele transforma um valor x dessa característica em uma faixa $[minimo, máximo]$ usando as fórmulas apresentadas na Equação 2.11 e Equação 2.12, onde a primeira transforma os valores em uma nova faixa $[0, 1]$ e a segunda transforma o valor em uma faixa $[-1, 1]$.

$$x' = \frac{x - min}{max - min} \tag{2.11}$$

$$x' = 2 * \frac{x - min}{max - min} - 1 \tag{2.12}$$

A normalização de dados pela técnica *Z-Score*, por sua vez, se utiliza dos valores de média (μ) e desvio padrão (σ) de uma característica X para definir um novo valor para

x. Para chegar no novo valor de x, a equação apresentada na Equação 2.14 é utilizada. Os novos valores são uma representação da distância dos valores em relação à média.

$$x' = \frac{x - \mu}{\sigma} \tag{2.13}$$

$$x' = \frac{104 - 86.75}{15.12} \tag{2.14}$$

A normalização de dados é uma etapa extremamente importante tanto para a vizualização do conjunto de dados quanto para o seu uso em algoritmos de ML e de redes neurais. Técnicas sólidas como o *min-max* e *Z-Score* podem aumentar o desempenho de tais algoritmos significativamente (SINGH; SINGH, 2020); (SUPRAJITNO et al., 2022); (PATEL et al., 2022).

2.2 Métodos de Classificação

Métodos de classificação são técnicas responsáveis por definir classes (também chamados de categorias) a dados de um conjunto de dados (KRISHNAIAH; NARSIMHA; CHANDRA, 2014). Esses métodos empregam algoritmos e modelos estatísticos para treinar um sistema de classificação capaz de fazer previsões precisas em novos dados não rotulados. Diferentes técnicas podem ser empregadas para realizar essa tarefa, entre elas se destacam:

- Árvores de Decisão (*Decision Trees*): Árvores de decisão são estruturas que classificam instâncias ao organizá-las com base nos valores das características. Cada nó em uma árvore de decisão representa uma característica em uma instância a ser classificada, e cada ramo representa um valor que o nó pode assumir. As instâncias são classificadas a partir do nó raiz e ordenadas com base em seus valores de características (KOTSIANTIS et al., 2007).

- Floresta Aleatória (*Random Forest*): Baseada no conceito de *bagging*, à fim de reduzir a variância, a floresta aleatória é um método de aprendizado supervisionado proposto por Ho (HO, 1995), que pode ser utilizado para resolver problemas de classificação. O algoritmo consiste em uma combinação de classificadores de árvores de decisão. Cada árvore emite um voto unitário a fim de classificar a entrada e a classe mais popular é determinada. Quanto maior o número de árvores na floresta, maior a precisão e menor a probabilidade de sobreajuste (*overfitting*) (KOKLU; OZKAN, 2020).

- Máquina de Vetores de Suporte (*Support Vector Machines* ou SVM): é um método baseado em kernel com alta capacidade computacional para problemas de classificação com o uso de um separador de margem máxima (IZMAILOV; VAPNIK;

VASHIST, 2013). Por meio do uso de uma técnica chamada de truque de *kernel* (*kernel trick*) ele pode resolver problemas que utilizam de dados não lineares ao separá-los em um espaço de características de alta dimensão (SCHÖLKOPF, 2000). As SVMs têm a capacidade de representar problemas complexos e são resistentes ao sobreajuste.

- k Vizinhos mais Próximos (k *Nearest Neighbors* ou kNN): Este algoritmo realiza a classificação de acordo com o valor k fornecido, conforme a classe do vizinho mais próximo. No algoritmo kNN, a classificação de uma observação é feita usando o conjunto de todas as observações com classe conhecida. A observação a ser testada é processada individualmente com cada amostra no conjunto de treinamento, então, para determinar a classe da observação a ser testada, são selecionadas as k observações mais próximas dessa amostra no conjunto de treinamento (DENG et al., 2016). Essa distância pode ser medida de diferentes formas, como (i) Distância Euclidiana, (ii) Distância de Manhattan e (iii) Distância de Minkowski, por exemplo. No agrupamento formado pelas amostras selecionadas, a classe com mais amostras será definida como a classe da observação.

- *Perceptron* Multicamadas (*Multi Layer Perceptron* ou MLP): MLP são redes neurais compostas por camadas de *perceptrons* capazes de aprender eventos e determinar respostas de maneira similar à tomada de decisão do cérebro humano (KOKLU; OZKAN, 2020). Essas redes são compostas por uma camada de entrada (responsável por receber os valores de entrada), uma ou mais camadas ocultas (que definem o fluxo dos dados pela rede) e uma camada de saída (onde os resultados podem ser encontrados).

Dentre essas técnicas, MLP e kNN são amplamente empregadas na literatura atual devido à sua eficácia reconhecida em lidar com problemas complexos de classificação, cada uma destacando-se por características específicas que as tornam robustas e versáteis em diferentes contextos de análise de dados.

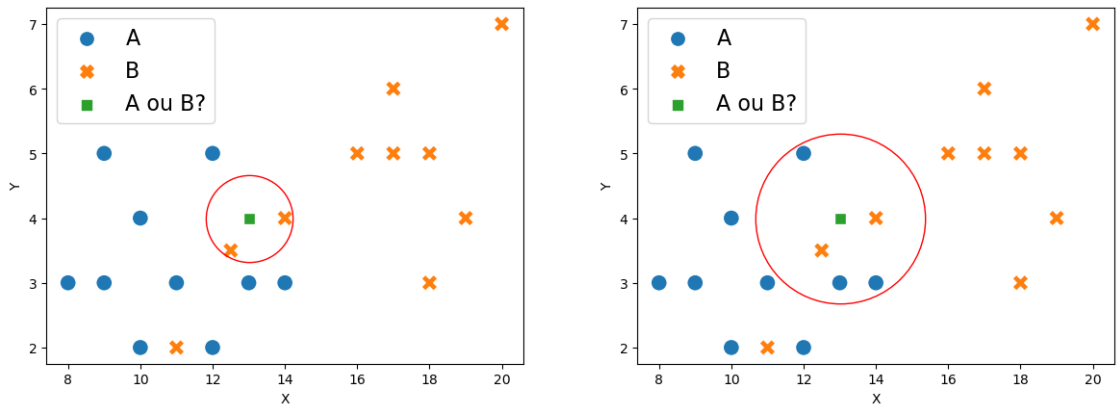
2.2.1 *k-Nearest Neighbour* (kNN)

k-Nearest Neighbors (k-NN) é uma técnica de aprendizado supervisionado que opera com base na proximidade espacial entre observações no espaço de características multidimensionais. Seu uso é bastante relevante em áreas como segurança de informações (LIAO; VEMURI, 2002), agricultura (HOSSAIN; HOSSAIN; RAHAMAN, 2019) e classificação de textos (MOLDAGULOVA; SULAIMAN, 2017), entre outras.

O k-NN classifica novos pontos de dados atribuindo a eles a classe predominante entre os k vizinhos mais próximos, onde k é um hiperparâmetro definido pelo usuário. A proximidade é geralmente calculada usando uma métrica de distância, como a distância

euclidiana. Durante o treinamento, o algoritmo não cria um modelo explícito, mas mantém todo o conjunto de dados de treinamento na memória. Quando um novo ponto de dados precisa ser classificado, o k-NN calcula as distâncias para todos os pontos de treinamento, identifica os k vizinhos mais próximos e determina a classe com base na maioria dos votos entre esses vizinhos (TAUNK et al., 2019). O k-NN é simples de entender e implementar, sendo especialmente útil em problemas de classificação em que a estrutura dos dados não é facilmente parametrizável por modelos matemáticos. No entanto, a escolha do valor de k é crítica, pois pode afetar significativamente o desempenho do algoritmo (ZHANG et al., 2017).

Escolher um valor muito pequeno para k (por exemplo, $k = 2$) pode resultar em classificações excessivamente sensíveis a ruídos e flutuações nos dados, levando a uma alta variância. Por outro lado, escolher um valor muito grande para k (por exemplo, $k = \text{N}$, onde N é o tamanho do conjunto de dados de treinamento) pode suavizar a fronteira de decisão e levar a um alto viés . A escolha apropriada de k é muitas vezes determinada por meio de técnicas de validação cruzada ou otimização de hiperparâmetros, com o objetivo de encontrar o equilíbrio entre viés e variância que melhor se ajusta ao conjunto de dados específico (TAUNK et al., 2019).



(a) Valor de $k = 2$. (b) Valor de $k = 5$.

Figura 2 – Classificação Pelo método k-Nearest Neightbour com valores de k diferentes

Fonte: Próprio Autor

A Figura 2 apresenta um exemplo de algoritmo de classificação de dados baseado na técnica kNN. No exemplo, os dados devem ser classificados como “A” ou “B”. Nesse caso, temos um dado sem classificação que foi chamado de “A ou B”. Na Figura 2(a) foi usado um valor de $k = 2$. Nesse caso, o algoritmo buscaria, à partir da posição do dado “A ou B”, os dois vizinhos mais próximos, retornando o conjunto de dados dentro do círculo. Nesse caso, como todos os dados retornados são da mesma classe, o dado “A ou B” seria classificado com “B”. Já no exemplo Figura 2(b) foi usado o valor de $k = 5$. Dessa maneira, o algoritmo busca pelos 5 vizinhos mais próximos, retornando o conjunto

de dados circulados. Sendo assim, como a maioria dos dados pertence a classe "A", essa seria a classificação do dado "A ou B".

Apesar de apresentar bons resultados, kNN tem um alto custo na classificação de novas instâncias uma vez que quase toda a computação ocorre no momento da classificação, em vez de quando os exemplos de treinamento são inicialmente encontrados (ZHANG, 2020). Além disso, devido ao custoso armazenamento de todo o conjunto de treinamento ser necessário para novas classificações, conjuntos de dados muito grandes podem representar um problema para esse modelo (GUO et al., 2003).

2.2.2 Redes Neurais Artificiais

As redes neurais são um componente fundamental na área de aprendizado de máquina e inteligência artificial, modeladas a partir do funcionamento do cérebro humano. Essas redes são projetadas para realizar tarefas complexas de processamento de informações, incluindo reconhecimento de padrões, previsão e classificação. Elas consistem em uma coleção de unidades interconectadas, chamadas neurônios artificiais, que são organizados em camadas. Uma rede neural é composta por uma camada de entrada, que recebe os dados de entrada, uma ou mais camadas ocultas (exceto no caso de um *perceptron*, que não possui camadas ocultas) que processam esses dados usando funções de ativação, e uma camada de saída que produz o resultado final. O processo de treinamento envolve a apresentação de exemplos de entrada junto com as saídas desejadas, ajustando iterativamente os pesos das conexões para minimizar a diferença entre as saídas previstas e as saídas desejadas, geralmente usando algoritmos de otimização.

Um *perceptron* é um exemplo de rede neural básica, proposto por Rosenblatt (ROSENBLATT, 1958) em 1958, ele é constituído apenas de uma camada de entrada com nós completamente conectados com um nó na camada de saída. A Figura 3 apresenta o modelo do *perceptron* de Rosenblatt. Nesse caso, $x_1 \ \ldots \ x_n$ são os valores de entrada da rede neural, $w_1 \ \ldots \ w_n$ são os pesos atribuídos pela rede neural para a entrada, b é o viés e Σ é o somatório do produto entre as entradas e os pesos definidos à elas com o viés definido pela rede neural. Esse valor é então aplicado à uma função de ativação (σ) resultando na saída da rede neural.

O *perceptron*, entretanto, é capaz somente de classificar problemas lineares, limitando seus casos de uso em muitos problemas do mundo real onde a distribuição dos dados é não-linear (FU; ROBLES-KELLY; ZHOU, 2010). Para resolver problemas desse tipo, redes neurais mais avançadas devem ser utilizadas, como por exemplo um *Perceptron* Multicamadas (*Multilayer Percetron* ou MLP). Um MLP é um tipo de rede neural que consiste em uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída (Figura 4), onde cada neurônio de uma camada está conectado a todos os neurônios da camada seguinte em uma arquitetura chamada de propagação direta (*feedfoward propagation*), e essas conexões têm pesos associados que são ajustados durante o treinamento

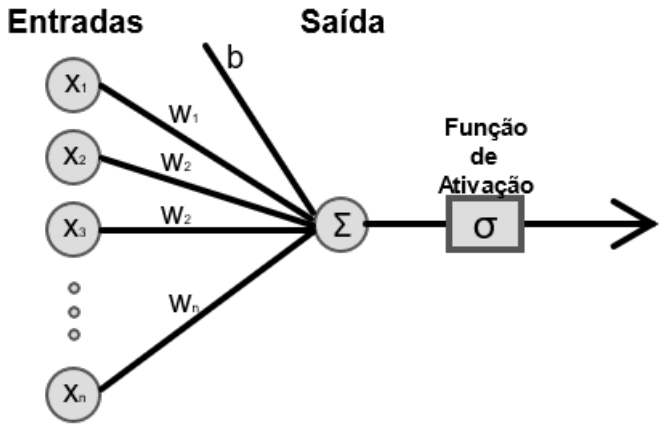


Figura 3 – *Perceptron* de Rosenblatt

Fonte: Próprio Autor

por meio de uma técnica chamada de retropropagação (*backpropagation*) (POPESCU et al., 2009). O resultado do somatório dos produtos das entradas de cada neurônios são adicionados à uma contantes (chamada de viés ou *bias*) e aplicados à uma função de ativação, o resultado dessa função define se a conexão com o neurônio da próxima camada será ativada (GARDNER; DORLING, 1998).

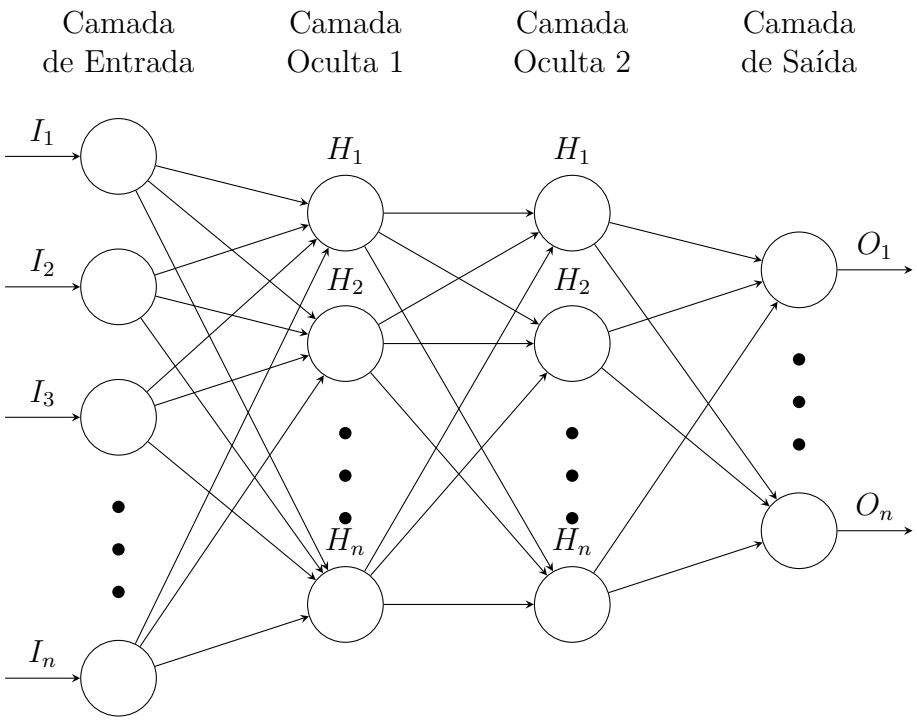


Figura 4 – Exemplo de Rede Neural de Múltiplas Camadas

Fonte: Próprio Autor

Treinar uma rede neural consistem em definir os valores associados aos pesos e aos vieses da rede. Para isso é necessário é necessário prover à rede neural um conjunto de dados que possuem o valor de saída esperado, chamado de conjunto de treinamento (*training set*). Então, a rede escolhe valores aleatórios pequenos para os pesos e para os vieses (GARDNER; DORLING, 1998). Assim, é possível calcular a diferença entre o valor de saída nessa rede, comparado com o valor de saída esperado, valor chamado de erro. Então, por meio da técnica de retropropagação, o valor do erro é transferido para as camadas anteriores. Em seguida os valores dos pesos e dos vieses são ajustados por meio de uma técnica de redução de erro, como por exemplo o método do gradiente. A taxa de aprendizado é um parâmetro que determina o tamanho do passo durante o processo iterativo de aprendizado pelo método do gradiente, um valor muito alto de taxa de aprendizado pode levar a rede neural a omitir valores de pesos ou vieses que levariam à menor taxa de erro, enquanto valores muito pequenos podem tornar o treinamento muito lento (GARDNER; DORLING, 1998). Com os pesos ajustados, a rede é testada novamente, seguindo os mesmos passos. A etapa de treinamento pode ter milhares de iterações, dependendo da complexidade da rede neural, até que a taxa de erro esteja em um nível adequado.

A função de ativação de um neurônio de uma MLP deve ser escolhido com cuidado, uma vez que a escolha de uma função linear não aumenta a capacidade computacional de uma MLP comparada com um *perceptron*, portanto funções não lineares como a sigmoide logística (Equação 2.15) e a tangente hiperbólica (Equação 2.16) se tornam escolhas comuns (POPESCU et al., 2009). Outras funções não lineares, como unidade linear retificada (Equação 2.17) ou *softplus* (Equação 2.18), também são amplamente utilizadas.

$$f(s) = \frac{1}{1 + e^{-s}} \tag{2.15}$$

$$f(s) = \frac{1 - e^{-a \cdot s}}{1 + e^{-a \cdot s}} \tag{2.16}$$

$$f(s) = max(0, s) \tag{2.17}$$

$$f(s) = \log_e (1 + e^s) \tag{2.18}$$

Redes neurais são utilizadas em aplicações como reconhecimento de imagem (GUO et al., 2017), detecção de fraudes (ASHA; KR, 2021), diagnóstico médico (MARQUES; AGARWAL; DÍEZ, 2020) e muito mais. A capacidade de adaptabilidade e generalização das redes neurais artificiais as torna uma escolha extremamente útil para problemas de classificação em uma variedade de domínios (LIU; ZHANG; PHILIP, 2018).

3 TRABALHOS RELACIONADOS

Neste capítulo será realizada uma revisão da literatura recente, explorando trabalhos que versam sobre as aplicações de aprendizado de máquina e técnicas de processamento de dados em contextos semelhantes aos abordados neste estudo. Na seção 3.1 serão investigados trabalhos que exploram o impacto do pré-processamento de dados. Na seção 3.2 serão vistos trabalhos que investigam métodos de classificação de sementes. Já na seção 3.3 serão abordados trabalhos recentes que abordam o impacto do pré-processamento na área de classificação de sementes.

3.1 Técnicas de Pré-processamento de Dados

Técnicas de pré-processamento de dados podem aprimorar drasticamente o desempenho de técnicas (ZELAYA, 2019). Recentemente, a estimativa desse impacto tem se tornado foco de diferentes estudos. Dong et al. (DONG et al., 2019) apresentam um modelo de imputação de valores faltantes baseado em *k-NN* para conjuntos de dados trans-ômicos (que possuem informações sobre diversos tipos de organismos diferentes) e o compara com outras técnicas (*e.g.*, imputação pela média e deleção de valores faltantes) no contexto de ocorrências de 4 tipos diferentes de câncer em pacientes, onde esse modelo apresentou resultados melhores, reduzindo o erro de imputação. De forma similar, Daberdaku et al. (DABERDAKU; TAVAZZI; CAMILLO, 2020) comparam o uso das técnicas de interpolação linear, *k-NN* e da técnica proposta (uso combinado de interpolação linear e *k-NN* para imputação de dados laboratoriais de pacientes, onde os resultados apontam um aumento de perforance de $7,9\%$ comparado com o algoritmo base (3D-MICE) quando a interpolação linear é combinada com o *k-NN*.

Nascimento et al. (NASCIMENTO et al., 2021) propõem um método de detectção de *outliers* que combina técnicas de regressão com técnicas estatísticas para a predição de consumo de energia, onde o método de regressão floresta aleatória foi utilizado, juntamente com métodos estatísticos 3σ, MAD e *boxplot* ajustado, entre outros, obtendo o melhor resultado com o uso do *boxplot* ajustado. Saleem, Aslam e Shaukat (SALEEM; ASLAM; SHAUKAT, 2021) compararam o desempenho de diversas técnicas de detecção de *outliers* em cenários de dados com diferentes números de observações e tipos de distribuições. Foi observado que técnicas baseadas no desvio padrão (como a técnica 3σ, por exemplo) não têm uma boa taxa de acerto na detecção de *outliers* em conjuntos pequenos e em conjuntos que possuem uma distribuição muito próxima da distribuição padrão. Por outro lado, para conjuntos de dados maiores e mais distorcidos, a taxa de acerto é maior. Além disso, técnicas que se utilizam da mediana (como a técnica MAD) não foram afetadas pelo tamanho e tipo de distribuição do conjunto. Li et al. (LI et al., 2021) estuda o impacto de diferentes técnicas de pré-processamento de dados (*i.e.*, imputação de valores faltantes, tratamento de *outliers*, tratamento de rótulos errados, tratamento de inconsistências e remoção de observações duplicadas). Pôde ser observado que a imputação de valores

faltantes aumentou a performance de técnicas de ML em mais casos, comparado com a deleção das obserções que possuem esse tipo de ocorrência. Também foi observado que a escolha do método de detecção de *outliers* é bastante importante quando se espera aumentar o desempenho de técnicas de ML.

D. Singh e B. Singh (SINGH; SINGH, 2020) investigaram o impacto da normalização de dados em diferentes conjuntos de dados. No estudo foram utilizados 24 conjuntos de dados diferentes e 12 métodos de normalização foram comparados em relação à classificação pelo método kNN. O estudo também aborda o impacto da seleção e exclusão de características do conjunto de dados. Foi observado que a faixa de valores escolhidas para o método *min-max* (*e.g.*, [0, 1], [-1, 1]) tem baixo impacto na acurácia da classificação, mas que o uso dessa técnica aumentou a taxa de acurácia em relação ao conjunto de dados não normalizado. Ainda, o uso do método de normalização *Z-Score* teve boa performance quando o conjunto de dados não foi submetido à seleção de características e, em geral, teve performance melhor do que o método *min-max*. Em relação à seleção de características, os resultado apontam um aumento na acurácia média quando combinada com todos os métodos de normalização de dados. De maneira semelhante Raju et al. (RAJU et al., 2020) compara diferentes técnicas de normalização e transformação de dados em três métodos de classificação diferentes (*i.e.*, kNN, SVM Radial e SVM Sigmoidal), onde pode ser observado que a utilização de normalização de dados aumentou a acurácia dos classificadores em até 10%, com o método *min-max* obtendo a melhor performance quando utilizado com o classificador SVM Radial.

3.2 Classificação de Sementes

A utilização de técnicas de *machine learning* no setor de agricultura tem se tornado mais proeminente nos últimos anos (TANTALAKI; SOURAVLAS; ROUMELIOTIS, 2019). Nesse contexto, a classificação de sementes tem sido objeto de estudo, uma vez que a utilização de sementes de baixa qualidade pode resultar em baixa produtividade, mesmo em condições de cultivo favoráveis (KOKLU; OZKAN, 2020). Salimi e Boelt (SALIMI; BOELT, 2019) utilizaram um classificador baseado em espectroscopia (MSI) para classificar sementes de beterraba-sacarina (*Beta vulgaris*) em diferentes tipo de dano à semente, chegando a taxas de acurácia de até 85%. Keya, Majumdar e Islam (KEYA; MAJUMDAR; ISLAM, 2020) realizaram a segmentação de imagens com uso de uma rede neural convolucional (CNN) para classificar 5 tipos diferentes de sementes (*i.e.*, arroz, milho, abóbora, abóbora-d'água e porongo), obtendo uma acurácia superior à 90%. Kiratiratanapruk et al. (KIRATIRATANAPRUK et al., 2020) avaliou o impacto de técnicas de pré-processamento de imagem (orientação e triagem das sementes) no resultado final da classificação de sementes de arroz, obtendo resultados que apontaram uma melhora de até 1,3% na classificação por redes neurais e 2-3% na classificação por SVM e kNN. Além disso, comparou métodos de classificação SVM e kNN com o uso de redes neurais profun-

das na classificação de sementes de arroz, onde a acurácia obtida pela rede neural utilizada foi $11,24\%$ maior. Medeiros et al. (MEDEIROS et al., 2020a) investigou a capacidade de predição de germinação e vigor de sementes de *U. brizantha* (conhecida comumente como grama paliçada) por métodos de ML. Com o auxílio de captura de características por espectroscopia e raio-X, a acurácia em prever a capacidade de germinação das sementes chegou à 85%. Entretando, a acurácia em estimar o vigor das sementes foi de apenas 62%. Medeiros et al. (MEDEIROS et al., 2020b) apresentou um modelo *machine learning* interativo para classificação de 7 classes de sementes de soja, onde o treinamento foi realizado com a supervisão humana, à fim de reduzir o custo computacional e o tamanho do conjunto de treinamento. Dessa forma comparada com o método de Floresta Aleatória e SVM o modelo proposto apresentou a melhor acurácia entre os três, apesar de utilizar apenas 10% das observações totais como conjunto de treinamento (comparado com os 70% utilizados pelos outros métodos). O trabalho também propões um método de classificar as sementes de acordo com sua qualidade fisiológica, onde a melhor taxa de acurácia foi obtida pelo método de floresta aleatória (*random forest*). Koklu e Ozkan (KOKLU; OZKAN, 2020) investigaram o desempenho de técnicas de classificação baseadas em *machine learning* para a classificação de sementes de feijão, obtendo resultados que indicam que as técnicas de SVM e MLP apresentam alto desempenho (taxa de acurácia de 93,13% e 91,73%, respectivamente) na classificação deste tipo de semente. Koklu, Sarigil e Ozbek (KOKLU; SARIGIL; OZBEK, 2021) compararam o uso de diversas técnicas de classificação baseadas em *machine learning* para a classificação de sementes de abóbora. Dentre as técnicas utililizadas no estudo, SVM e MLP obtiveram os melhores resultados, com taxa de acurácia de $88,64\%$ e $88,52\%$, respectivamente.

3.3 Impacto do Pré-processamento na Classificação de Sementes

Apesar do impacto positivo de técnicas de pré-processamento de dados em métodos de classificação por ML ser bastante aceito, poucos estudos foram feitos para quantificar esse impacto (HAMID et al., 2022). Xu et al. (XU et al., 2022) estudaram o impacto da seleção de características na classificação do vigor de sementes de milho, e obteram o melhor resultado com a utilização do método UVE (*Uninformative Variable Elimination* ou Elimição de Variável Não Informativa). Macuácua, Centeno e Amisse et al. (MACUÁCUA; CENTENO; AMISSE, 2023) estudaram o efeito do balanceamento de classes das sementes de feijão, obtendo um aumento na taxa de acurácia de até $1,6\%$, quando utilizada a técnica SVM comparada com o estudo de Koklu e Ozkan (KOKLU; OZKAN, 2020). Slowinski (SŁOWIŃSKI, 2021) estuda o efeito da redução de características com alta taxa de correlação na classificação de sementes de feijão, obtendo resultados que apontam que a eliminação dessas características não afetou negativamente a acurácia da classificação. De maneira similar, Khan et al. (KHAN et al., 2023) compara o resultado de técnicas de ML na classificação de sementes de feijão após o balanceamento das classes

do conjunto de entrada, além de realizar a remoção de *outliers*, obtendo aumentos na taxa de acurácia de até 6% se comparados com os dados não balanceados. Esse trabalho não estuda o impacto que a remoção dos *outliers* causou no resultado final dos métodos de classificação.

Dentre os estudos abordados neste capítulo, é evidente que poucas pesquisas realizam a avaliação do impacto das técnicas de pré-processamento de dados sobre os métodos de aprendizado de máquina empregados. Portanto, o presente trabalho visa preencher essa lacuna de maneira técnica, direcionando-se especificamente para a avaliação detalhada do impacto das técnicas de pré-processamento de dados (*i.e.*, imputação de valores faltantes, tratamento de *outliers* e normalização de dados) sobre os métodos de ML kNN e MLP.

4 METODOLOGIA

Neste capítulo será apresentada a metodologia aplicada no desenvolvimento desse trabalho, tendo como objetico final analisar o impacto das técnicas de pré-processamento de dados na classificação de sementes de feijão.

A seção 4.1 apresenta a base de dados escolhida para a realização dos experimentos, a seção 4.2 descreve o ambiente de desenvolvimento dos experimentos, a seção 4.3 detalha os métodos de classificação que serão utilizados nos experimentos, a seção 4.5 apresenta as técnicas de pré-processamento de dados utilizadas e os experimentos que serão realizados. A seção 4.4 explica o método de validação cruzada que será utilizado para garantir a qualidade dos resultados obtidos. Por fim, a seção 4.7 descreve as métricas de avaliação de desempenho dos experimentos.

4.1 Base de Dados e suas características

O estudo utilizou um conjunto de dados do Repositório de Aprendizado de Máquina da UCI (Universidade da California, Irvine) proveniente do experimento conduzido por (KOKLU; OZKAN, 2020). Esse conjunto de dados corresponde a imagens de 13.611 feijões secos de sete variedades diferentes determinadas pelo Instituto Turco de Padrões, obtidas de produtores de sementes certificados (REPOSITORY, 2020).As variedades incluem Barbunya, Bombay, Cali, Dermason, Horoz, Seker e Sira. Essas imagens foram capturadas por uma câmera RGB de 2.2 megapixels chamada Prosilica GT2000C, com uma resolução de 2048 $\times$ 1088, um sensor CMOS e uma faixa efetiva de temperatura operacional de -20 °C a +65 °C. Antes de capturar as imagens, os feijões foram colocados em um fundo escuro para facilitar o processo de segmentação. Durante o processo de captura de imagem, a câmera foi posicionada a 15 cm acima das amostras na parte superior da caixa, visando proporcionar um ambiente de iluminação homogêneo. As lâmpadas na parte superior da caixa permitiram a eliminação de ruídos ambientais. A Figura 5 apresenta o resultado da captura de imagens das sementes segundo o método descrito.

Segundo (KOKLU; OZKAN, 2020) a próxima parte do experimento foi a remoção das sombras e de ruídos do fundo da imagem, seguido pela separação das sementes por meio de segmentação das imagens. Para esse fim, a imagem foi convertida para escala de cinza, e o método de limiarização global de Otsu foi utilizado. Esse método permite binarizar uma imagem, maximizando a separabilidade de duas classes: primeiro plano e plano de fundo. A partir da imagem binária resultante de cada semente, 16 características espaciais foram computadas com o software MATLAB (INC., 2022), incluindo algumas medidas geométricas diretas, como distâncias ou área (apresentadas na Figura 6), e outras na forma de índices calculados a partir delas, como a compacidade, conforme mostrado abaixo:

- Área (A): A área de uma zona de feijão e o número de pixels dentro de seus limites;

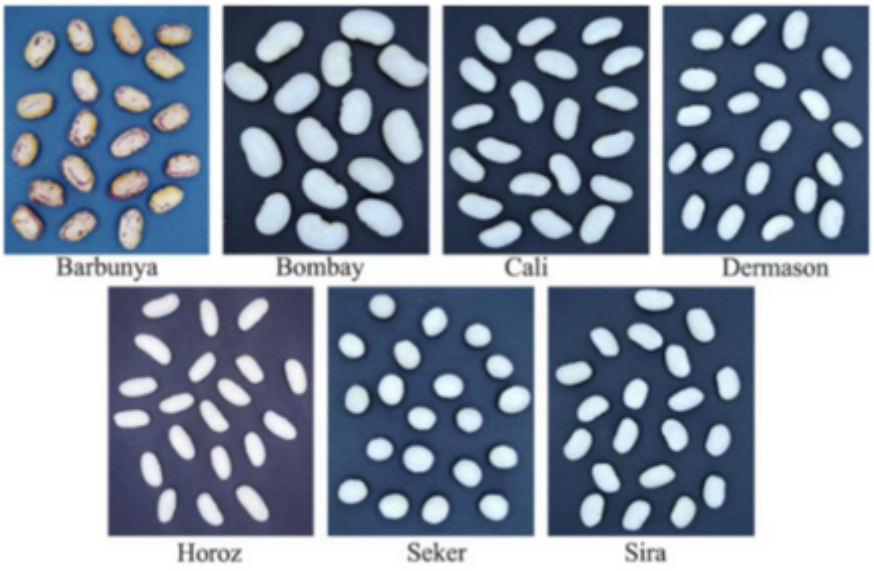


Figura 5 – Imagens das sementes de feijões capturadas.

Fonte: (KOKLU; OZKAN, 2020)

- Perímetro (P): A circunferência do feijão é definida como o comprimento de sua borda;
- Comprimento do eixo maior (L): A distância entre as extremidades da linha mais longa que pode ser desenhada a partir de um feijão;
- Comprimento do eixo menor (l): A linha mais longa que pode ser desenhada a partir do feijão enquanto está perpendicular ao eixo principal;
- Razão de aspecto (K): Define a relação entre L e l:

$$K = \frac{L}{l}$$

- Excentricidade (Ec): Excentricidade da elipse que tem os mesmos momentos que a região;
- Área convexa (C): Número de pixels no polígono convexo mínimo que pode conter a área de uma semente de feijão;
- Diâmetro equivalente (Ed): O diâmetro de um círculo que tem a mesma área que a área de uma semente de feijão. Calculada pela seguinte fórmula:

$$Ed = \sqrt{\frac{4 \cdot A}{\pi}}$$

- Extensão (Ex): A razão dos pixels na caixa delimitadora para a área do feijão. Calculada pela seguinte fórmula;

$$Ex = \frac{A}{A_B} \text{ onde } A_B = \text{Área do retângulo delimitador}$$

- Solidez (S): Também conhecida como convexidade. A razão dos pixels na casca convexa para aqueles encontrados nos feijões. Calculada pela seguinte fórmula;

$$S = \frac{A}{C}$$

- Circularidade (R): Calculada pela fórmula seguinte:

$$R = \frac{4 \cdot \pi \cdot A}{P^2}$$

- Compacidade (CO): Mede a circularidade de um objeto. Dada pela fórmula seguinte:

$$CO = \frac{Ed}{L}$$

- Fator de Forma 1 (SF1) - calculado por:

$$SF1 = \frac{L}{A}$$

- Fator de Forma 2 (SF2) - calculado por:

$$SF2 = \frac{l}{A}$$

- Fator de Forma 3 (SF3) - calculado por:

$$SF3 = \frac{A}{\frac{L}{2} \cdot \frac{L}{2} \cdot \pi}$$

- Fator de Forma 4 (SF4) - calculado por:

$$SF4 = \frac{A}{\frac{L}{2} \cdot \frac{l}{2} \cdot \pi}$$

4.2 Ambiente de Desenvolvimento

No desenvolvimento deste trabalho, foi utilizado o ambiente de desenvolvimento Google Colaboratory (Colab)[1], uma plataforma baseada em nuvem que oferece suporte à execução de código Python[2] em notebooks Jupyter. O Colab proporciona uma infraestrutura computacional gratuita com aceleração por GPU, permitindo a implementação e

[1] https://colab.research.google.com/
[2] https://www.python.org/

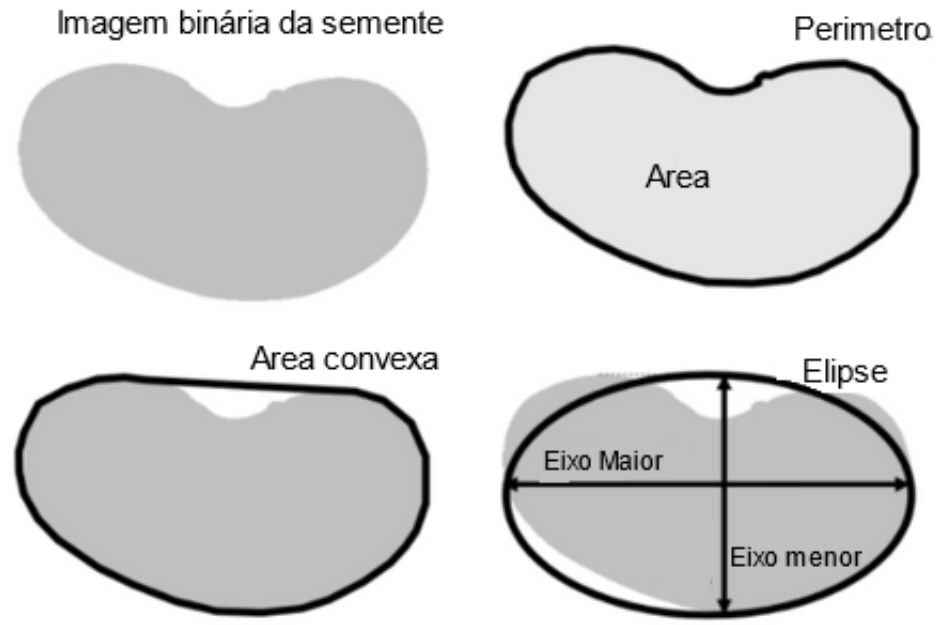


Figura 6 – Exemplo de caracterísitcas espaciais obtidas da imagem binária

Fonte: Adaptado de (MACUÁCUA; CENTENO; AMISSE, 2023)

execução eficiente de modelos de aprendizado de máquina e análises de dados. A escolha pelo Colab se deve à sua acessibilidade, facilidade de uso e capacidade de integração com bibliotecas populares de *machine learning*, proporcionando um ambiente eficaz para o desenvolvimento e experimentação ao longo do projeto. Além disso, as seguintes bibliotecas e pacotes serão utilizados:

- *Pandas*: biblioteca de limpeza, transformação, agregação e manipulação de dados (TEAM, 2020);
- *Numpy*: biblioteca fundamental para operações numéricas eficientes, especialmente em contextos de aprendizado de máquina (HARRIS et al., 2020);
- *matplotlib*: biblioteca em Python para criação de gráficos e visualizações de dados (HUNTER, 2007);
- *scikit-learn*: biblioteca em Python que oferece ferramentas simples e eficientes para análise de dados e aprendizado de máquina, incluindo algoritmos de classificação, regressão, clustering e pré-processamento de dados (PEDREGOSA et al., 2011).

4.3 Métodos de Classificação utilizados

Neste trabalho optou-se por empregar os métodos de classificação kNN (k-vizinhos mais próximos) e rede neural MLP (perceptron de múltiplas camadas), conforme descritos

no trabalho de (KOKLU; OZKAN, 2020). Essa abordagem visa realizar uma comparação abrangente e justa do impacto de técnicas de pré-processamento aplicadas no conjunto de dados.

Dessa maneira, será utilizado um modelo de classificação por *machine learning* do tipo *k Nearest Neighbors*, como descrito na subseção 2.2.1, utilizando a distância euclidiana como forma de cálculo de distância e valor de $k = 10$ e uma rede neural MLP (*Multi Layer Perceptron*) que segue o modelo de (KOKLU; OZKAN, 2020), possuindo 4 camadas, dispostas da seguinte maneira:

- Camada de Entrada: 16 perceptrons ($x_1 - x_{16}$), representando as características das sementes de feijão, como descritas na seção 4.1;
- Camadas Ocultas: De acordo com (KOKLU; OZKAN, 2020) a melhor estrutura de camadas ocultas para essa finalidade possui duas camadas, a primeira com 12 perceptrons e a segunda com 3 perceptrons;
- Camada de Saída: 7 perceptrons ($O_1 - O_7$) representando os tipos de sementes de feijão, sendo elas *Barbunya*, *Bombay*, *Cali*, *Dermason*, *Horoz*, *Seker* e *Sira*.

A Figura 7 exibe uma representação da rede neural utilizada. Os parâmetros utilizados nessa rede neural estão apresentados na Tabela 6.

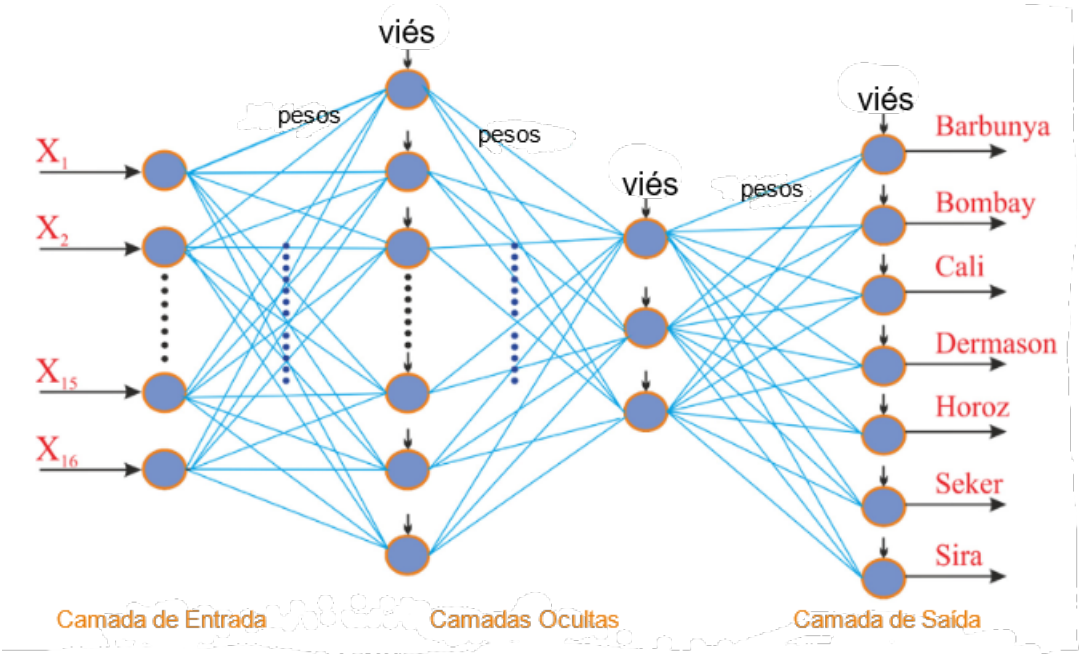


Figura 7 – Rede Neural MLP utilizada.

Fonte: Adaptado de (KOKLU; OZKAN, 2020)

Tabela 6 – Parâmetros da Rede Neural.

Nome do Parâmetro	Valor
Número de Camadas Ocultas	2
Função de Ativação das Camadas Ocultas	Sigmóide
Número de Ativação da Camada de Saída	Sigmóide
Taxa de Aprendizado	0,3
Gradiente Mínimo de Desempenho	$1e^{-5}$
Meta de Desempenho	$1e^{-3}$
Número de Épocas Máximo para Treinamento	500

4.4 Validação Cruzada

A validação cruzada é uma abordagem desenvolvida para aumentar a robustez da classificação. Nesse método, o conjunto de dados é aleatoriamente dividido em um número especificado de conjuntos de tamanho idêntico. Um dos subconjuntos é utilizado como conjunto de teste e o sistema é treinado com os conjuntos restantes. Esse processo é repetido até que todos os conjuntos tenham sido testados no sistema (ARLOT; CELISSE, 2010). Esse método aumenta a generalização dos testes, garantindo a qualidade dos resultados. Nesse trabalho o número de subconjuntos utlizados foi 10, ou seja, cada subconjunto terá um tamanho igual a 10% do tamanho do conjunto total.

4.5 Técnicas de Pré-processamento de Dados e Experimentos

Como forma de avaliar o impacto do pré-processamento no desempenho dos métodos de classificação, as seguintes técnicas de pré-processamento de dados serão testadas:

- Imputação de Valores Faltantes por kNN (kNN-I), com os parâmetros apresentados na Tabela 7;

Tabela 7 – Parâmetros do método kNN para Imputação.

Nome do Parâmetro	Valor
Número de Vizinhos Observados	5
Métrica de Cálculo de Distância	*nan_euclidean*
Cálculo de Peso	*uniform*

- Imputação de Valores Faltantes por Interpolação Linear (IL), com os parametros apresentados na Tabela 8;
- Remoção de *Outliers* por 3σ (3σ);
- Remoção de *Outliers* por Desvio Absoluto Mediano (MAD);
- Normalização de dados por *min-max* (*min-max*);

Tabela 8 – Parâmetros do método Interpolação Linear para Imputação.

Nome do Parâmetro	Valor
Método	*nearest*
Ordem	3
Limite	*None*
Direção	Ambas

- Normalização de dados por *Z-Score* (*Z-Score*).

Para a realização dos experimentos, diferentes técnicas serão combinadas, dessa maneira teremos 16 experimentos diferentes, dispostos de acordo com a Tabela 9, onde cada experimento recebe um nome (*e.g.*, E1, E2, E3...) e é composto pela aplicação de uma técnica de imputação de valores faltantes (IVF), uma técnica de remoção de *outliers*, uma técnica de normalização de dados e um método de classificação.

Tabela 9 – Descrição dos experimentos à serem realizados.

Experimento	IVF	*Outliers*	Normalização	Classificação
Experimento 1 (E1)	kNN-I	3σ	*min-max*	kNN
Experimento 2 (E2)	kNN-I	3σ	*min-max*	MLP
Experimento 3 (E3)	kNN-I	3σ	*Z-Score*	kNN
Experimento 4 (E4)	kNN-I	3σ	*Z-Score*	MLP
Experimento 5 (E5)	kNN-I	MAD	*min-max*	kNN
Experimento 6 (E6)	kNN-I	MAD	*min-max*	MLP
Experimento 7 (E7)	kNN-I	MAD	*Z-Score*	kNN
Experimento 8 (E8)	kNN-I	MAD	*Z-Score*	MLP
Experimento 9 (E9)	IL	3σ	*min-max*	kNN
Experimento 10 (E10)	IL	3σ	*min-max*	MLP
Experimento 11 (E11)	IL	3σ	*Z-Score*	kNN
Experimento 12 (E12)	IL	3σ	*Z-Score*	MLP
Experimento 13 (E13)	IL	MAD	*min-max*	kNN
Experimento 14 (E14)	IL	MAD	*min-max*	MLP
Experimento 15 (E15)	IL	MAD	*Z-Score*	kNN
Experimento 16 (E16)	IL	MAD	*Z-Score*	MLP

4.6 Aperfeiçoamento da Rede Neural

Também serão realizados testes de maneira à aprimorar a rede neural proposta por Koklu e Ozkan (KOKLU; OZKAN, 2020) onde os seguintes parâmetros serão alterados:

- Taxa de Aprendizado Inicial (TAI);
- Número Máximo de Épocas (NME);
- Neurônios da Primeira Camada (NPC);

- Neurônios da Segunda Camada (NSC).

Para a realização dos experimentos, diferentes tamanhos de redes neurais serão testados, gerando um total de 16 experimentos diferentes, apresentados na Tabela 10, onde cada experimento recebeu um nome (*e.g.*, A1, A2, A3...) e diferentes combinações de parâmetros de execução da rede neural.

Tabela 10 – Descrição dos experimentos à serem realizados.

Experimento	TAI	NME	NPC	NSC
Koklu e Ozkan	0.3	500	12	3
Experimento A1	0.001	2000	10	5
Experimento A2	0.001	2000	10	10
Experimento A3	0.001	2000	10	15
Experimento A4	0.001	2000	10	20
Experimento A5	0.001	2000	15	5
Experimento A6	0.001	2000	15	10
Experimento A7	0.001	2000	15	15
Experimento A8	0.001	2000	15	20
Experimento A9	0.001	2000	20	5
Experimento A10	0.001	2000	20	10
Experimento A11	0.001	2000	20	15
Experimento A12	0.001	2000	20	20
Experimento A13	0.001	2000	25	5
Experimento A14	0.001	2000	25	10
Experimento A15	0.001	2000	25	15
Experimento A16	0.001	2000	25	20

4.7 Métricas de Avaliação de Desempenho

A avaliação de um modelo de classificação é feita a partir da comparação entre as classes preditas pelo modelo e as classes verdadeiras de cada exemplo. Todas as métricas de classificação têm como objetivo comum medir quão distante o modelo está da classificação perfeita, porém fazem isto de formas diferentes.

Uma forma bastante simples de visualizar a performance de um modelo de classificação é através de uma matriz de confusão. Esta matriz indica quantos exemplos existem em cada grupo: falso positivo (FP), falso negativo (FN), verdadeiro positivo (VP) e verdadeiro negativo (NV). A Tabela 11 apresenta o funcionamento da matriz de confusão em relação aos grupos descritos.

Outra métrica importante para a avaliação de desempenho de uma classificação é a acurácia, que nos diz quantos de nossos exemplos foram de fato classificados corretamente, independente da classe. Por exemplo, se temos 100 observações e 90 delas foram classificados corretamente, nosso modelo possui uma acurácia de 90%. A acurácia é definida pela Equação 4.1.

Tabela 11 – Matriz de Confusão

	Positivos Predito	Negativos Predito
Positivos Reais	Positivo Verdadeiro (PV)	Falso Negativo (FN)
Negativos Reais	Falso Positivo (FP)	Negativo Verdadeiro (NV)

Embora seja uma métrica valiosa, a acurácia pode distorcer a avaliação do desempenho do modelo. Em um cenário em que um conjunto de dados contém 1000 observações de pacientes, dos quais 990 estão saudáveis e 10 têm câncer, a criação de um modelo para prever a presença ou ausência de câncer pode levar a interpretações enganosas. Se o modelo simplesmente rotular todos os pacientes como saudáveis, ainda assim alcançaria uma acurácia de 99%.

$$Acurácia = \frac{\text{predições corretas}}{\text{todas predições}} = \frac{VP + NV}{VP + NV + FP + FN} \tag{4.1}$$

A precisão, dada pela Equação 4.2, é uma métrica que corrige essa representação inadequada, considerando as observações que foram erroneamente classificadas como positivas. Ela representa a quantidade de vezes que o modelo classificou de maneira correta uma observação positiva. Essa representação ignora as classificações negativas, não representando de maneira completa o desempenho do modelo.

$$Precisão = \frac{\text{predições de positivos verdadeiros corretas}}{\text{todas as predições positivas}} = \frac{VP}{VP + FP} \tag{4.2}$$

A revocação, por sua vez, dá maior ênfase para os erros por falso negativo. Esta métrica é definida pela razão entre a quantidade de exemplos classificados corretamente como positivos e a quantidade de exemplos que são de fato positivos, conforme a Equação 4.3.

$$Revocação = \frac{\text{predições de positivos verdadeiros corretas}}{\text{todos positivos reais}} = \frac{VP}{VP + FN} \tag{4.3}$$

A métrica *F1-score* representa uma média harmônica entre a precisão e a revocação, sendo calculada de acordo com a Equação 4.4. Os valores do *F1-score* variam entre 0 e 1, assumindo o valor 0 quando todos os valores foram classificados incorretamente e 1 quando todos os valores foram classificados corretamente (CHICCO; JURMAN, 2020).

$$F_1 = 2 * \frac{Precisão * Revocação}{Precisão + Revocação} \tag{4.4}$$

Uma característica fundamental da média harmônica é que, quando a precisão ou a revocação se aproximam de zero, o *F1-score* também terá um valor próximo de zero. Em outras palavras, para alcançar um *F1-score* elevado, é necessário que tanto a precisão quanto a revocação sejam altas. Assim, um modelo com um bom *F1-score* demonstra

habilidade não apenas em realizar previsões corretas (alta precisão), mas também em recuperar exemplos da classe de interesse (alta revocação). Portanto, essa métrica se destaca como um indicador mais abrangente da qualidade do modelo.

5 DESENVOLVIMENTO

Neste capítulo será apresentada o desenvolvimento deste trabalho, tendo como objetico final analisar o impacto das técnicas de pré-processamento de dados na classificação de sementes de feijão.

A seção 5.1 apresenta os resultados dos experimentos propostos no Capítulo 4. A seção 5.2 avalia os resultados de aperfeiçoamentos aplicados à *Multi Layer Perceptron* utilizada nos experimentos. Por fim, a seção 5.3 apresenta a comparação entre os resultados obtibos neste trabalho com os resultados de Koklu e Ozkan (KOKLU; OZKAN, 2020), além de uma breve discussão sobre os resultados obtidos.

5.1 Resultados dos Experimentos

Conforme apresentado na Tabela 9 do Capítulo 4, foram realizados 16 experimentos, abrangendo todas as combinações possíveis entre as técnicas de pré-processamento e os classificadores propostos. Como o conjunto de dados utilizado não possui dados faltantes, foram inseridos 5% de dados faltantes de maneira totalmente aleatória para todos os experimentos. Cada um dos experimentos foi repetido 50 vezes, de maneira à garantir resultados consistentes e minimizar os efeitos de variabilidade.

Os experimentos descritos por Koklu e Ozkan (KOKLU; OZKAN, 2020) foram refeitos conforme a metodologia proposta. Contudo, os resultados obtidos divergem dos apresentados em seu estudo. Essas disparidades possivelmente decorrem de procedimentos de pré-processamento de dados não documentados ou incompletamente descritos na obra em questão.

Os resultados obtidos pelos experimentos propostos serão comparados com os resultados obtidos pela repetição dos experimentos de Koklu e Ozkan (KOKLU; OZKAN, 2020). Adicionalmente, foi realizada a normalização dos dados antes da aplicação dos métodos de Koklu, o que resultou em números mais próximos dos relatados em seu estudo. Durante este capítulo, os resultados que aplicam a normalização antes da classificação serão descritos com o sufixo *min-max*, uma vez que foi esse método de normalização que foi utilizado.

A Tabela 12 apresenta os resultados dos experimentos que aplicam classificadores kNN. Os resultados apresentam um impacto grande do uso da normalização de dados na classificação, especialmente quando comparamos os resultados de Koklu-KNN e Koklu-KNN-minmax. Apesar disso, parece que as outras técnicas tem um impacto pequeno nas métricas de avaliação.

A Tabela 13 apresenta os resultados dos experimentos envolvendo classificadores MLP. Os achados sugerem uma baixa eficácia da MLP ao lidar com dados não normalizados. Além disso, a aplicação de outras técnicas demonstrou contribuir para a melhoria das métricas de avaliação de desempenho da classificação.

Dentre os resultados apresentados, o experimento 11 teve o melhor desempenho

Tabela 12 – Resultados dos Experimentos com Classificadores KNN.

Nome	Acurácia (%)	Precisão (%)	Revocação (%)	F1-Score (%)
Experimento 1	91.84	93.31	92.76	93.00
Experimento 3	92.07	93.54	93.04	93.26
Experimento 5	91.84	93.33	92.76	93.01
Experimento 7	92.10	93.56	93.07	93.29
Experimento 9	92.41	93.85	93.35	93.57
Experimento 11	**92.60**	**94.04**	**93.60**	**93.79**
Experimento 13	92.43	93.87	93.36	93.59
Experimento 15	92.59	94.03	93.57	93.78
Koklu-KNN	71.96	73.26	71.63	72.09
Koklu-KNN-minmax	92.18	93.64	93.15	93.36

Tabela 13 – Resultados dos Experimentos com Classificadores MLP.

Nome	Acurácia (%)	Precisão (%)	Revocação (%)	F1-Score (%)
Experimento 2	86.70	87.27	87.08	86.58
Experimento 4	90.41	91.97	91.40	91.51
Experimento 6	86.57	87.02	87.01	86.45
Experimento 8	90.33	91.80	91.31	91.38
Experimento 10	87.71	88.42	88.17	87.79
Experimento 12	90.89	92.40	91.93	92.00
Experimento 14	87.52	87.94	87.82	87.35
Experimento 16	**90.91**	**92.38**	**91.97**	**92.02**
Koklu-MLP	26.05	3.72	14.29	5.91
Koklu-MLP-minmax	87.40	88.22	87.99	87.61

dentre os experimentos que utilizam o kNN como método de classificação, apontando a efetividade das técnicas IL, 3σ e *min-max* para o pré-processamento de dados para esse classificador. De maneira similar, o experimento 16 teve o melhor desempenho dentre os experimentos que utilizam a rede neural MLP como método de classificação. Isso aponta uma efetidade grande da combinação das técnicas de Interpolação Linear, *Median Absolute Deviation* e *Z-Score*. O Apêndice O apresenta a Matriz de Confusão do Experimento 15 e o Apêndice P apresenta a Matriz de Confusão, o Gráfico de Acurácia por época e o Gráfico de Perda por Época, para referência.

5.2 Aperfeiçoamentos da Rede Neural

Embora os resultados da classificação da MLP sejam favoráveis, ainda há margem para aprimoramentos por meio de ajustes de parâmetros. A rede neural proposta apresenta certos parâmetros que se destacam por sua singularidade, especialmente a taxa de aprendizado inicial e a configuração das camadas ocultas.

Segundo o site oficial da biblioteca *scikit-learn* (PEDREGOSA et al., 2011), o valor padrão de taxa de aprendizado para uma *Multi Layer Perceptron* é de 0.001, dessa

maneira o valor utilizado por Koklu e Ozkan (KOKLU; OZKAN, 2020)(0.3) é bastante chamativo. Esse valor alto causa ajustes repentinos durante o treinamento da rede neural, sendo facilmente percebido nos gráficos de perda e de acurácia contidos nos anexos deste trabalho.

Outro parâmetro interessante foi o número máximo de épocas parametrizado no trabalho original. A utilização de apenas 500 épocas para treinamento da rede neural pode interromper o processo de aprendizado da rede neural antes que a taxa de erro seja baixa o suficiente. Nesse caso, foram realizados testes iterativos para encontrar um ponto de convergência da rede neural, que levaram ao resultado de que a rede neural precisava de um número próximo de 2000 épocas para chegar à um erro mínimo.

Além disso, a escolha da topologia das camadas ocultas da rede neural também chama a atenção, o ajuste repentino da primeira camada (que contém 12 neurônios) para a segunda camada (que contém apenas 3 neurônios) pode ocasionar a perda de informações durante o processo de treinamento.

Levando em consideração os pontos destacados, uma nova bateria de experimentos foi realizada, onde foram testados diversos parâmetros da rede. Os parâmetros destes testes são apresentados na Tabela 10 do Capítulo 4. Para estes experimentos foram utilizadas as ténicas de pré-processamento do Experimento 16, uma vez que esse foi o experimento que apresentou melhor resultado durante os testes iniciais.

Tabela 14 – Resultados dos Experimentos de Aperfeiçoamento da MLP.

Experimento	Acurácia (%)	Precisão (%)	Revocação (%)	F1-Score (%)
Koklu e Ozkan (baseline)	91.73	93.11	92.68	92.88
Experimento A1	92.95	94.11	93.86	93.97
Experimento A2	93.12	94.32	94.03	94.15
Experimento A3	93.07	94.27	94.01	94.12
Experimento A4	93.08	94.28	94.04	94.14
Experimento A5	93.07	94.29	94.03	94.14
Experimento A6	93.14	94.35	94.09	94.20
Experimento A7	93.15	94.37	94.09	94.21
Experimento A8	93.16	94.37	94.09	94.21
Experimento A9	93.13	94.39	94.11	94.23
Experimento A10	93.11	94.35	94.07	94.19
Experimento A11	93.14	94.37	94.10	94.22
Experimento A12	93.16	94.39	94.11	94.23
Experimento A13	93.05	94.31	94.00	94.13
Experimento A14	93.10	94.32	94.02	94.15
Experimento A15	93.18	94.40	94.14	94.25
Experimento A16	93.12	94.36	94.08	94.20

A Tabela 14 apresenta os resultados dos experimentos realizados. Todos os experimentos realizados tiveram uma acurácia significativamente maior do que os experimentos iniciais. A pequena diferença entre as diferentes configurações de camadas da rede neu-

ral sugere um interferência maior pela taxa inicial de aprendizado e limite máximo de iterações.

5.3 Comparação e Discussão

Neste estudo, buscamos avaliar o impacto das técnicas de pré-processamento de dados na classificação de sementes de feijão, comparando nossos resultados com aqueles obtidos por Koklu e Ozkan (KOKLU; OZKAN, 2020). Utilizamos duas abordagens principais de classificação: kNN e MLP, aplicando diversas técnicas de pré-processamento.

Inicialmente, reproduzimos os experimentos de Koklu e Ozkan (KOKLU; OZKAN, 2020) conforme a metodologia descrita por eles. No entanto, observamos divergências nos resultados, o que indicou possíveis lacunas na documentação dos procedimentos de pré-processamento de dados realizados no estudo original. Ao aplicar a normalização min-max, obtivemos resultados mais próximos dos relatados por Koklu e Ozkan, destacando a importância dessa etapa para a performance dos classificadores.

Os experimentos denominados Koklu-KNN e Koklu-KNN-minmax ilustram que a normalização min-max melhorou substancialmente a performance. Entretanto, outras técnicas de pré-processamento não demonstraram melhorias significativas quando combinadas com kNN. O melhor desempenho foi observado no experimento 15, que utilizou uma combinação de Interpolação Linear, *Median Absolute Deviation* e *Z-Score*. A Figura 8 apresenta o resultado das métricas destes experimentos.

A Figura 9 revela que os classificadores MLP são altamente sensíveis à normalização dos dados. A MLP apresentou uma performance inferior com dados não normalizados, mas mostrou melhorias significativas ao aplicar diversas técnicas de pré-processamento. O experimento 16, que também utilizou Interpolação Linear, *Median Absolute Deviation*, e *Z-Score*, teve o melhor desempenho, corroborando a efetividade dessa combinação de técnicas. Podemos notar a maior eficiência ao se utilizar a técnica de normalização de dados *Z-Score* em relação à técnica *min-max*. Os experimentos 4, 8, 12 e 16 utilizam dessa técnica e apresentam métricas de avaliação consideravelmente mais altas que os experimentos 2, 6, 10 e 14.

Com base nos resultados iniciais, realizamos uma nova série de experimentos focados em ajustar parâmetros da rede neural, como taxa de aprendizado e configuração das camadas ocultas. Conforme a Tabela 14, todos os ajustes resultaram em melhorias significativas na acurácia em comparação com os experimentos iniciais. Nota-se que a taxa de aprendizado inicial e o número máximo de iterações tiveram um impacto mais relevante do que a configuração das camadas ocultas.

A Figura 10 apresenta o resultado dos experimentos realizados. Foi possível observar que o ajuste dos parâmetros da rede neural resultou em um perceptível aumento de desempenho da rede neural, principalmente os valores de taxa de aprendizado inicial e de máximo de iterações.

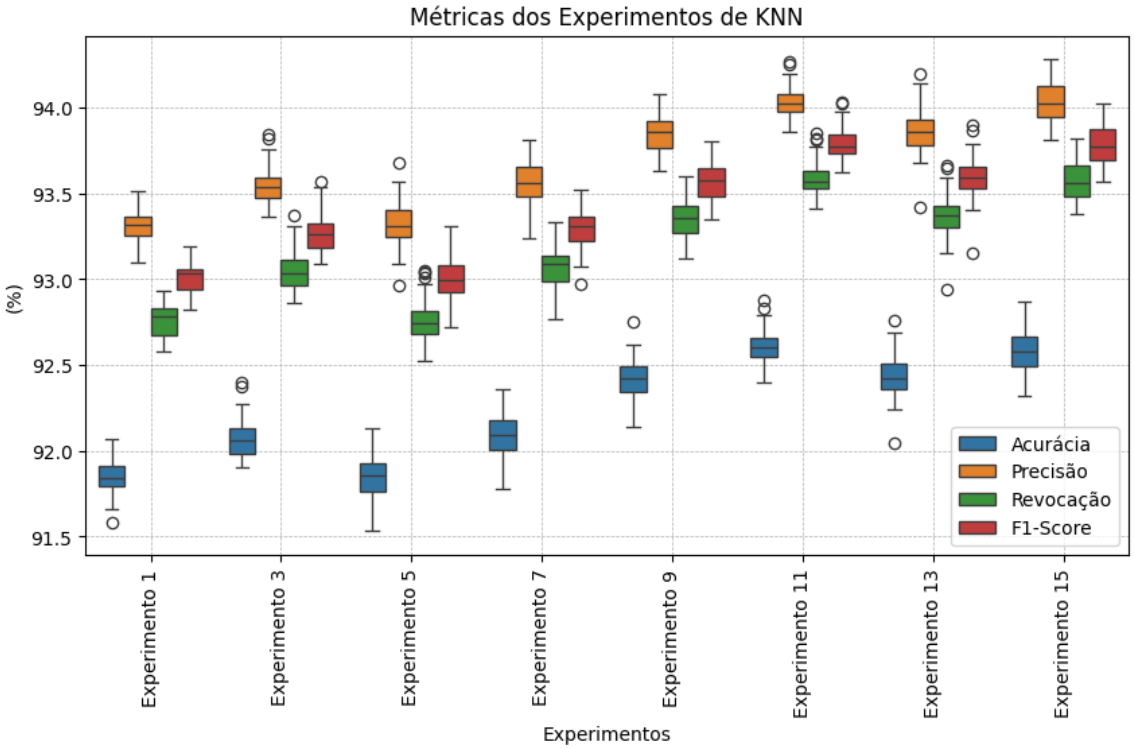


Figura 8 – Métricas dos Experimentos Iniciais com KNN.

Fonte: Próprio Autor

Por fim, foi feita uma comparação dos resultados obtidos em ambas baterias de experimentos em relação ao resultado apresentado por Koklu e Ozkan (KOKLU; OZKAN, 2020). A Figura 11 apresenta a comparação entre os resultados obtidos. Os resultados obtidos apontam que os experimentos iniciais tiveram um resultado levemente inferior aos descritos por Koklu e Ozkan (KOKLU; OZKAN, 2020), podendo ser resultado da baixa documentação em seu trabalho. Entretanto, foi possível observar uma melhora significativa na classificação das sementes pela rede neural aprimorada.

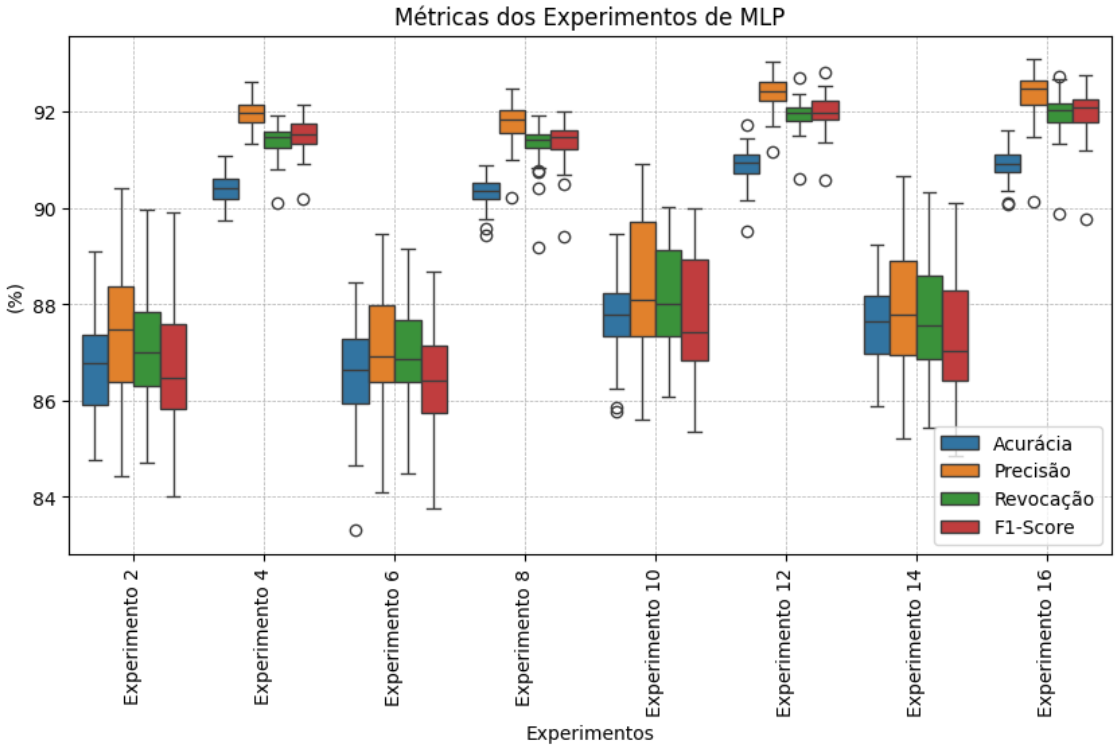


Figura 9 – Métricas dos Experimentos Iniciais com MLP.

Fonte: Próprio Autor

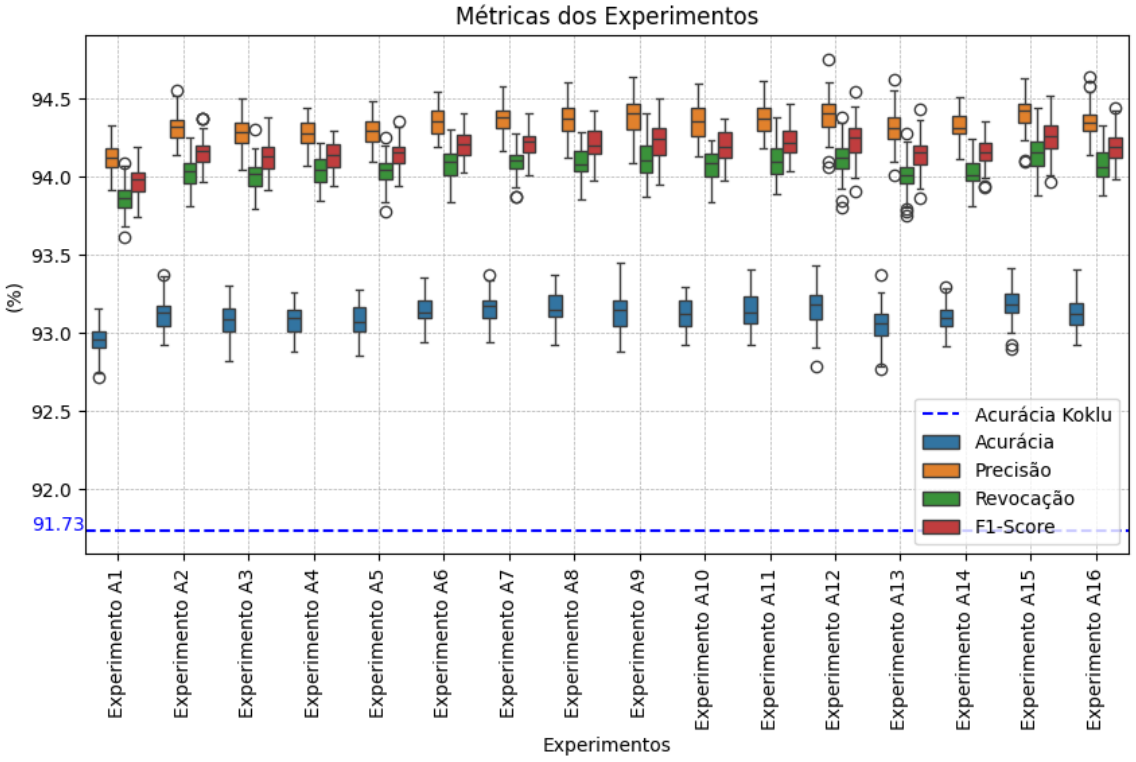


Figura 10 – Métricas dos Experimentos de Aperfeiçoamento da MLP.

Fonte: Próprio Autor

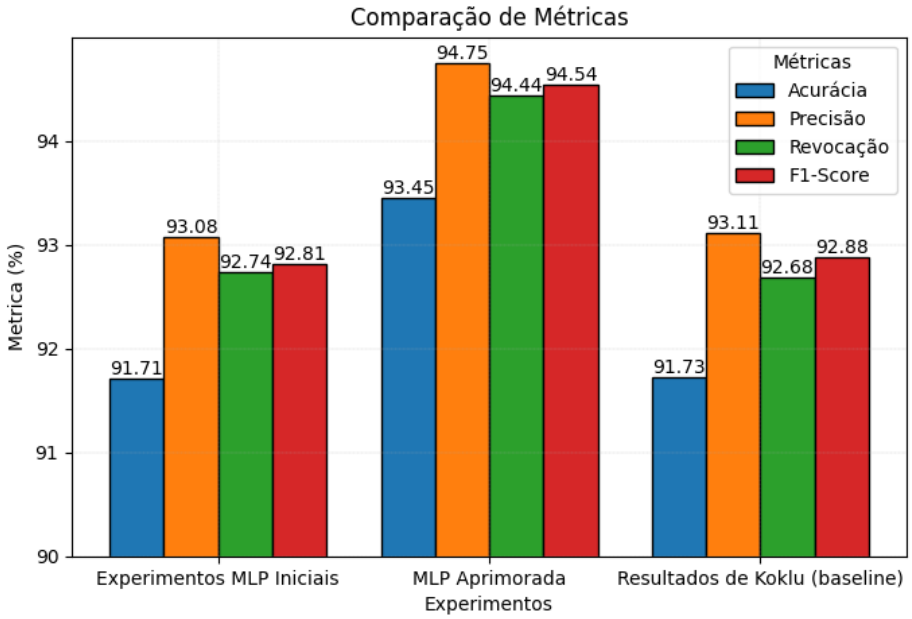


Figura 11 – Comparação dos Resultados dos Experimentos.

Fonte: Próprio Autor

6 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo avaliar o impacto das técnicas de pré-processamento de dados na classificação de sementes de feijão e propor um modelo aprimorado de Perceptron Multicamadas (MLP) capaz de melhorar o desempenho da classificação em comparação com o modelo proposto por Koklu e Ozkan (KOKLU; OZKAN, 2020). Através de experimentos extensivos, incluindo a replicação dos métodos de Koklu e Ozkan e testes adicionais focados no pré-processamento de dados e ajustes nos parâmetros da rede neural, resultados importantes foram obtidos.

Os experimentos demonstraram o papel crítico do pré-processamento de dados na melhoria do desempenho dos modelos de aprendizado de máquina. Especificamente, técnicas de normalização de dados impactaram significativamente o desempenho dos classificadores k-Nearest Neighbors (kNN) e MLP. Os resultados indicaram que os classificadores MLP são particularmente sensíveis à normalização de dados, sendo o método *Z-Score* mais eficaz do que o *min-max*.

O Experimento 16, que combinou Interpolação Linear, Desvio Absoluto Mediano e normalização *Z-Score*, alcançou o melhor desempenho, destacando a sinergia dessas técnicas. As descobertas ressaltam a necessidade de um pré-processamento de dados minucioso para maximizar o desempenho dos modelos de aprendizado de máquina.

Além disso, o modelo aprimorado de MLP, com parâmetros otimizados como taxa de aprendizado inicial e configuração das camadas ocultas, mostrou melhorias consideráveis em relação aos experimentos iniciais e ao modelo de base de Koklu e Ozkan (KOKLU; OZKAN, 2020). Essa otimização levou a ganhos significativos nas métricas de avaliação, demonstrando a importância de ajustar cuidadosamente os parâmetros da rede neural para alcançar resultados superiores na classificação.

Os resultados encontrados nesse trabalho revelam um espaço grande de avanço de pesquisa. Trabalhos futuros podem explorar métodos de aprendizado de máquina diferentes, como o SVM e DT realizados no trabalho de Koklu e Ozkan (KOKLU; OZKAN, 2020). Além disso, o impacto do pré-processamento de dados pode ser avaliado em outros conjuntos de dados, como de outros tipos de sementes (e.g., sementes de arroz, sementes de abóbora). Por fim, o conjunto de dados utilizado nesse trabalho não é balanceado, podendo ser analisado o impacto do balanceamento de dados utilizando diferentens técnicas.

REFERÊNCIAS

AGGARWAL, C. C.; AGGARWAL, C. C. **An introduction to outlier analysis**. [S.l.]: Springer, 2017. 1–2 p. Citado na página 27.

ARLOT, S.; CELISSE, A. A survey of cross-validation procedures for model selection. 2010. Citado na página 50.

ASHA, R.; KR, S. K. Credit card fraud detection using artificial neural network. **Global Transitions Proceedings**, Elsevier, v. 2, n. 1, p. 35–41, 2021. Citado na página 39.

BENHAR, H.; IDRI, A.; FERNÁNDEZ-ALEMÁN, J. Data preprocessing for heart disease classification: A systematic literature review. **Computer Methods and Programs in Biomedicine**, Elsevier, v. 195, p. 105635, 2020. Citado na página 23.

BOUKERCHE, A.; ZHENG, L.; ALFANDI, O. Outlier detection: Methods, models, and classification. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 53, n. 3, jun 2020. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3381028>. Citado na página 31.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 41, n. 3, jul 2009. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/1541880.1541882>. Citado na página 31.

CHICCO, D.; JURMAN, G. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. **BMC genomics**, BioMed Central, v. 21, n. 1, p. 1–13, 2020. Citado na página 53.

CHOUDHURY, S. J.; PAL, N. R. Imputation of missing data with neural networks for classification. **Knowledge-Based Systems**, Elsevier, v. 182, p. 104838, 2019. Citado na página 28.

DABERDAKU, S.; TAVAZZI, E.; CAMILLO, B. D. A combined interpolation and weighted k-nearest neighbours approach for the imputation of longitudinal icu laboratory data. **Journal of Healthcare Informatics Research**, Springer, v. 4, p. 174–188, 2020. Citado na página 41.

DENG, Z. et al. Efficient knn classification algorithm for big data. **Neurocomputing**, Elsevier, v. 195, p. 143–148, 2016. Citado na página 35.

DHARMA, F. et al. Prediction of indonesian inflation rate using regression model based on genetic algorithms. **Jurnal Online Informatika**, v. 5, n. 1, p. 45–52, 2020. Citado 2 vezes nas páginas 23 e 29.

DONG, X. et al. Tobmi: trans-omics block missing data imputation using a k-nearest neighbor weighted approach. **Bioinformatics**, Oxford University Press, v. 35, n. 8, p. 1278–1283, 2019. Citado na página 41.

FAN, C. et al. A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. **Frontiers in Energy Research**, Frontiers Media SA, v. 9, p. 652801, 2021. Citado 2 vezes nas páginas 28 e 29.

FU, Z.; ROBLES-KELLY, A.; ZHOU, J. Mixing linear svms for nonlinear classification. **IEEE Transactions on Neural Networks**, IEEE, v. 21, n. 12, p. 1963–1975, 2010. Citado na página 37.

GARDNER, M. W.; DORLING, S. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. **Atmospheric environment**, Elsevier, v. 32, n. 14-15, p. 2627–2636, 1998. Citado 2 vezes nas páginas 38 e 39.

GUO, G. et al. Knn model-based approach in classification. In: SPRINGER. **On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings**. [S.l.], 2003. p. 986–996. Citado na página 37.

GUO, T. et al. Simple convolutional neural network on image classification. In: IEEE. **2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)**. [S.l.], 2017. p. 721–724. Citado na página 39.

HAMID, Y. et al. Smart seed classification system based on mobilenetv2 architecture. In: IEEE. **2022 2nd International Conference on Computing and Information Technology (ICCIT)**. [S.l.], 2022. p. 217–222. Citado 2 vezes nas páginas 24 e 43.

HARIRI, R. H.; FREDERICKS, E. M.; BOWERS, K. M. Uncertainty in big data analytics: survey, opportunities, and challenges. **Journal of Big Data**, SpringerOpen, v. 6, n. 1, p. 1–16, 2019. Citado 2 vezes nas páginas 23 e 27.

HARRIS, C. R. et al. Array programming with NumPy. **Nature**, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: <https://doi.org/10.1038/s41586-020-2649-2>. Citado na página 48.

HO, T. K. Random decision forests. In: IEEE. **Proceedings of 3rd international conference on document analysis and recognition**. [S.l.], 1995. v. 1, p. 278–282. Citado na página 34.

HOSSAIN, E.; HOSSAIN, M. F.; RAHAMAN, M. A. A color and texture based approach for the detection and classification of plant leaf disease using knn classifier. In: IEEE. **2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)**. [S.l.], 2019. p. 1–6. Citado na página 35.

HUNTER, J. D. Matplotlib: A 2d graphics environment. **Computing in Science & Engineering**, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007. Citado na página 48.

INC., T. M. **MATLAB version: 9.13.0 (R2022b)**. Natick, Massachusetts, United States: The MathWorks Inc., 2022. Disponível em: <https://www.mathworks.com>. Citado na página 45.

IZMAILOV, R.; VAPNIK, V.; VASHIST, A. Multidimensional splines with infinite number of knots as svm kernels. In: IEEE. **The 2013 International Joint Conference on Neural Networks (IJCNN)**. [S.l.], 2013. p. 1–7. Citado na página 35.

JAMSHED, H. et al. Data preprocessing: A preliminary step for web data mining. **3c Tecnología: glosas de innovación aplicadas a la pyme**, 3ciencias, v. 8, n. 1, p. 206–221, 2019. Citado 2 vezes nas páginas 23 e 28.

JENGHARA, M. M. et al. Imputing missing value through ensemble concept based on statistical measures. **Knowledge and Information Systems**, Springer London, v. 56, p. 123–139, 7 2018. ISSN 02193116. Disponível em: <https://link.springer.com/article/10.1007/s10115-017-1118-1>. Citado na página 29.

KEYA, M.; MAJUMDAR, B.; ISLAM, M. S. A robust deep learning segmentation and identification approach of different bangladeshi plant seeds using cnn. In: IEEE. **2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)**. [S.l.], 2020. p. 1–6. Citado na página 42.

KHAN, M. S. et al. Comparison of multiclass classification techniques using dry bean dataset. **International Journal of Cognitive Computing in Engineering**, Elsevier, v. 4, p. 6–20, 2023. Citado na página 43.

KIRATIRATANAPRUK, K. et al. Development of paddy rice seed classification process using machine learning techniques for automatic grading machine. **Journal of Sensors**, Hindawi, v. 2020, 2020. Citado na página 42.

KNOL, M. J. et al. Unpredictable bias when using the missing indicator method or complete case analysis for missing confounder values: an empirical example. **Journal of clinical epidemiology**, Elsevier, v. 63, n. 7, p. 728–736, 2010. Citado 2 vezes nas páginas 28 e 31.

KOKLU, M.; OZKAN, I. A. Multiclass classification of dry beans using computer vision and machine learning techniques. **Computers and Electronics in Agriculture**, Elsevier, v. 174, p. 105507, 2020. Citado 21 vezes nas páginas 11, 13, 23, 24, 34, 35, 42, 43, 45, 46, 49, 51, 55, 57, 58, 59, 63, 107, 109, 111 e 113.

KOKLU, M.; SARIGIL, S.; OZBEK, O. The use of machine learning methods in classification of pumpkin seeds (cucurbita pepo l.). **Genetic Resources and Crop Evolution**, Springer, v. 68, n. 7, p. 2713–2726, 2021. Citado na página 43.

KOTSIANTIS, S. B. et al. Supervised machine learning: A review of classification techniques. **Emerging artificial intelligence applications in computer engineering**, Amsterdam, v. 160, n. 1, p. 3–24, 2007. Citado na página 34.

KRISHNAIAH, V.; NARSIMHA, G.; CHANDRA, N. S. Survey of classification techniques in data mining. **International Journal of Computer Sciences and Engineering**, v. 2, n. 9, p. 65–74, 2014. Citado na página 34.

LEYS, C. et al. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. **Journal of experimental social psychology**, Elsevier, v. 49, n. 4, p. 764–766, 2013. Citado na página 32.

LI, P. et al. Cleanml: A study for evaluating the impact of data cleaning on ml classification tasks. In: IEEE. **2021 IEEE 37th International Conference on Data Engineering (ICDE)**. [S.l.], 2021. p. 13–24. Citado na página 41.

LIAO, Y.; VEMURI, V. R. Use of k-nearest neighbor classifier for intrusion detection. **Computers & security**, Elsevier, v. 21, n. 5, p. 439–448, 2002. Citado na página 35.

LITTLE, R. J.; RUBIN, D. B. **Statistical analysis with missing data**. [S.l.]: John Wiley & Sons, 2019. v. 793. 3–5 p. Citado 2 vezes nas páginas 23 e 28.

LIU, Z.-M.; ZHANG, C.; PHILIP, S. Y. Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections. **IEEE Transactions on Antennas and Propagation**, IEEE, v. 66, n. 12, p. 7315–7327, 2018. Citado na página 39.

MACUÁCUA, J. C.; CENTENO, J. A. S.; AMISSE, C. Data mining approach for dry bean seeds classification. **Smart Agricultural Technology**, Elsevier, v. 5, p. 100240, 2023. Citado 3 vezes nas páginas 24, 43 e 48.

MANYIKA, J. et al. Big data: The next frontier for innovation, competition, and productivity. McKinsey & Company, 2011. Citado na página 23.

MARQUES, G.; AGARWAL, D.; DÍEZ, I. De la T. Automated medical diagnosis of covid-19 through efficientnet convolutional neural network. **Applied soft computing**, Elsevier, v. 96, p. 106691, 2020. Citado na página 39.

MCKINNEY Wes. Data Structures for Statistical Computing in Python. In: WALT Stéfan van der; MILLMAN Jarrod (Ed.). **Proceedings of the 9th Python in Science Conference**. [S.l.: s.n.], 2010. p. 56 – 61. Citado na página 31.

MEDEIROS, A. D. d. et al. Machine learning for seed quality classification: An advanced approach using merger data from ft-nir spectroscopy and x-ray imaging. **Sensors**, MDPI, v. 20, n. 15, p. 4319, 2020. Citado na página 43.

MEDEIROS, A. D. de et al. Interactive machine learning for soybean seed and seedling quality classification. **Scientific reports**, Nature Publishing Group UK London, v. 10, n. 1, p. 11267, 2020. Citado na página 43.

MILLER, J. Reaction time analysis with outlier exclusion: Bias varies with sample size. **The Quarterly Journal of Experimental Psychology Section A**, SAGE Publications Sage UK: London, England, v. 43, n. 4, p. 907–912, 1991. Citado na página 33.

MOLDAGULOVA, A.; SULAIMAN, R. B. Using knn algorithm for classification of textual documents. In: IEEE. **2017 8th international conference on information technology (ICIT)**. [S.l.], 2017. p. 665–671. Citado na página 35.

NASCIMENTO, G. F. M. et al. Outlier detection in buildings' power consumption data using forecast error. **Energies**, MDPI, v. 14, n. 24, p. 8325, 2021. Citado 4 vezes nas páginas 31, 32, 33 e 41.

NAYAK, S.; MISRA, B. B.; BEHERA, H. S. Impact of data normalization on stock index forecasting. **International Journal of Computer Information Systems and Industrial Management Applications**, v. 6, n. 2014, p. 257–269, 2014. Citado 2 vezes nas páginas 28 e 33.

NOOR, M. et al. Filling missing data using interpolation methods: Study on the effect of fitting distribution. **Key Engineering Materials**, Trans Tech Publ, v. 594, p. 889–895, 2014. Citado na página 30.

NOOR, N. M. et al. Comparison of linear interpolation method and mean method to replace the missing values in environmental data set. In: TRANS TECH PUBL. **Materials Science Forum**. [S.l.], 2015. v. 803, p. 278–281. Citado na página 31.

PATEL, C. et al. Forecasting nonstationary wind data using adaptive min-max normalization. In: IEEE. **2022 1st International Conference on Sustainable Technology for Power and Energy Systems (STPES)**. [S.l.], 2022. p. 1–6. Citado na página 34.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado 2 vezes nas páginas 48 e 56.

POPESCU, M.-C. et al. Multilayer perceptron and neural networks. **WSEAS Transactions on Circuits and Systems**, World Scientific and Engineering Academy and Society (WSEAS) Stevens Point ..., v. 8, n. 7, p. 579–588, 2009. Citado 2 vezes nas páginas 38 e 39.

RAJU, V. G. et al. Study the influence of normalization/transformation process on the accuracy of supervised classification. In: IEEE. **2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)**. [S.l.], 2020. p. 729–735. Citado na página 42.

REPOSITORY, U. M. L. **Dry Bean Dataset**. 2020. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C50S4B. Citado na página 45.

RIDZUAN, F.; ZAINON, W. M. N. W. A review on data cleansing methods for big data. **Procedia Computer Science**, Elsevier, v. 161, p. 731–738, 2019. Citado na página 27.

RIEGER, A.; HOTHORN, T.; STROBL, C. Random forests with missing values in the covariates. 2010. Citado na página 31.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citado na página 37.

SADINENI, P. K. Comparative study on query processing and indexing techniques in big data. In: IEEE. **2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)**. [S.l.], 2020. p. 933–939. Citado na página 23.

SALEEM, S.; ASLAM, M.; SHAUKAT, M. R. A review and empirical comparison of univariate outlier detection methods. **Pakistan Journal of Statistics**, v. 37, n. 4, 2021. Citado na página 41.

SALIMI, Z.; BOELT, B. Classification of processing damage in sugar beet (beta vulgaris) seeds by multispectral image analysis. **Sensors**, MDPI, v. 19, n. 10, p. 2360, 2019. Citado na página 42.

SARIJALOO, F. B. et al. Yield performance estimation of corn hybrids using machine learning algorithms. **Artificial Intelligence in Agriculture**, Elsevier, v. 5, p. 82–89, 2021. Citado na página 24.

SCHÖLKOPF, B. The kernel trick for distances. **Advances in neural information processing systems**, v. 13, 2000. Citado na página 35.

SHAHRIAR, M. S. et al. Detecting heat events in dairy cows using accelerometers and unsupervised learning. **Computers and electronics in agriculture**, Elsevier, v. 128, p. 20–26, 2016. Citado na página 31.

SILVA, J. de A.; HRUSCHKA, E. R. Eacimpute: an evolutionary algorithm for clustering-based imputation. In: IEEE. **2009 Ninth International Conference on Intelligent Systems Design and Applications**. [S.l.], 2009. p. 1400–1406. Citado na página 29.

SINGH, D.; SINGH, B. Investigating the impact of data normalization on classification performance. **Applied Soft Computing**, Elsevier, v. 97, p. 105524, 2020. Citado 3 vezes nas páginas 28, 34 e 42.

SŁOWIŃSKI, G. Dry beans classification using machine learning. **Proceedings http://ceur-ws. org ISSN**, v. 1613, p. 0073, 2021. Citado na página 43.

SUN, L. et al. Outlier data treatment methods toward smart grid applications. **IEEE Access**, IEEE, v. 6, p. 39849–39859, 2018. Citado na página 27.

SUPRAJITNO, H. et al. Investigations on impact of feature normalization techniques for prediction of hydro-climatology data using neural network backpropagation with three layer hidden. **International Journal of Sustainable Development & Planning**, v. 17, n. 7, 2022. Citado na página 34.

TANTALAKI, N.; SOURAVLAS, S.; ROUMELIOTIS, M. Data-driven decision making in precision agriculture: The rise of big data in agricultural systems. **Journal of agricultural & food information**, Taylor & Francis, v. 20, n. 4, p. 344–380, 2019. Citado na página 42.

TAUNK, K. et al. A brief review of nearest neighbor algorithm for learning and classification. In: IEEE. **2019 international conference on intelligent computing and control systems (ICCS)**. [S.l.], 2019. p. 1255–1260. Citado na página 36.

TEAM, T. pandas development. **pandas-dev/pandas: Pandas**. Zenodo, 2020. Disponível em: <https://doi.org/10.5281/zenodo.3509134>. Citado na página 48.

XIE, Y. et al. The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. **Earthquake Spectra**, SAGE Publications Sage UK: London, England, v. 36, n. 4, p. 1769–1801, 2020. Citado na página 23.

XU, P. et al. Vigor identification of maize seeds by using hyperspectral imaging combined with multivariate data analysis. **Infrared Physics & Technology**, Elsevier, v. 126, p. 104361, 2022. Citado na página 43.

YONG, X.; WARD, R. K.; BIRCH, G. E. Robust common spatial patterns for eeg signal preprocessing. In: IEEE. **2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society**. [S.l.], 2008. p. 2087–2090. Citado na página 31.

ZELAYA, C. V. G. Towards explaining the effects of data preprocessing on machine learning. In: IEEE. **2019 IEEE 35th international conference on data engineering (ICDE)**. [S.l.], 2019. p. 2086–2090. Citado 2 vezes nas páginas 24 e 41.

ZHANG, S. Cost-sensitive knn classification. **Neurocomputing**, Elsevier, v. 391, p. 234–242, 2020. Citado na página 37.

ZHANG, S. et al. Learning k for knn classification. **ACM Transactions on Intelligent Systems and Technology (TIST)**, ACM New York, NY, USA, v. 8, n. 3, p. 1–19, 2017. Citado 3 vezes nas páginas 27, 29 e 36.

Anexos

ANEXO A – EXPERIMENTO 1

No experimento 1 foi utilizado kNN como método de imputação de valores faltantes, 3σ como método de detecção de *outliers*, *min-max* como método de normalização e kNN como método de classificação. A Figura 12 mostra a matriz de confusão do Experimento 1.

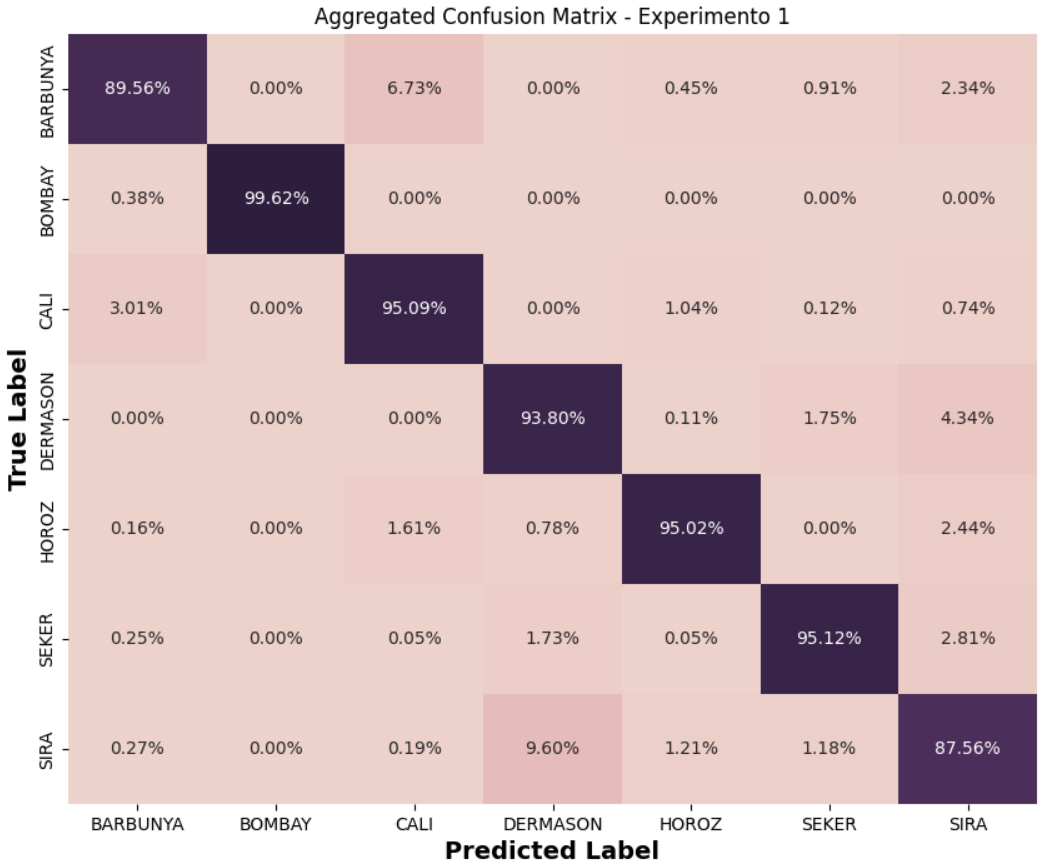


Figura 12 – Matriz de Confusão do Experimento 1.

ANEXO B – EXPERIMENTO 2

No experimento 2 foi utilizado kNN como método de imputação de valores faltantes, 3σ como método de detecção de *outliers*, *min-max* como método de normalização e *Multi Layer Perceptron* como método de classificação. A Figura 13 mostra a matriz de confusão do Experimento 2, enquanto a Figura 14 e a Figura 15 apresentam os gráficos de acurácia e de perda por época.

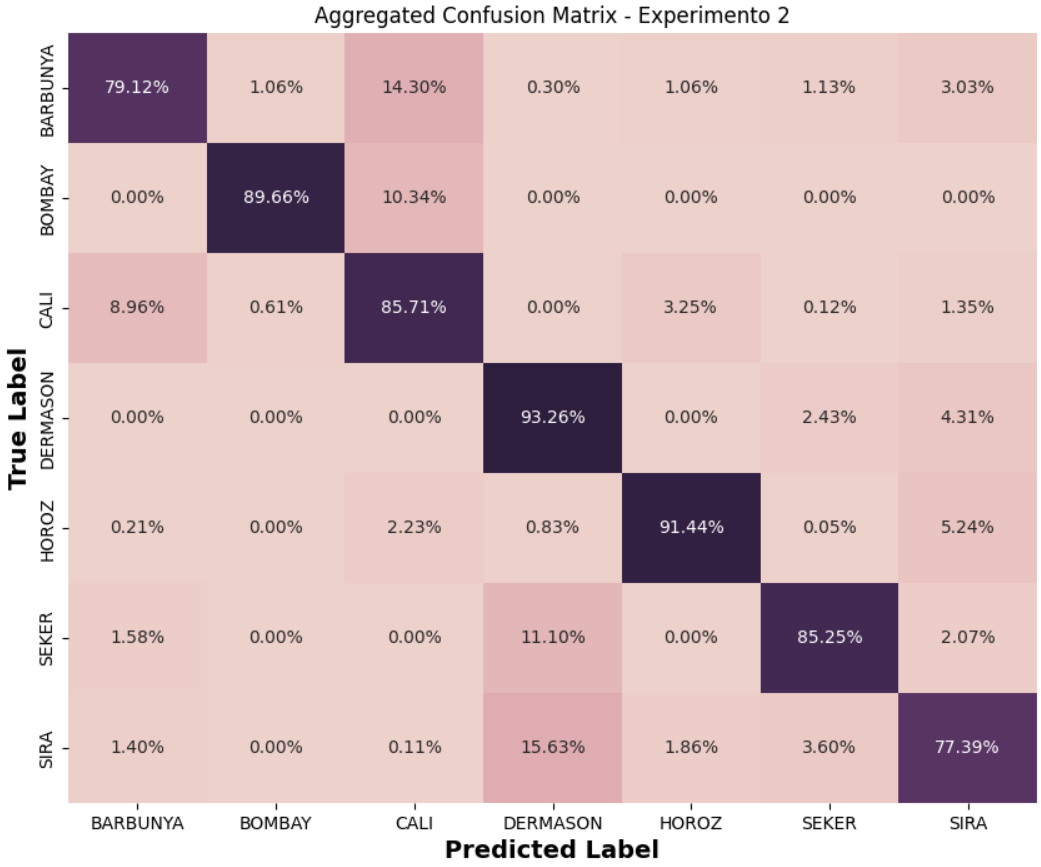


Figura 13 – Matriz de Confusão do Experimento 2.

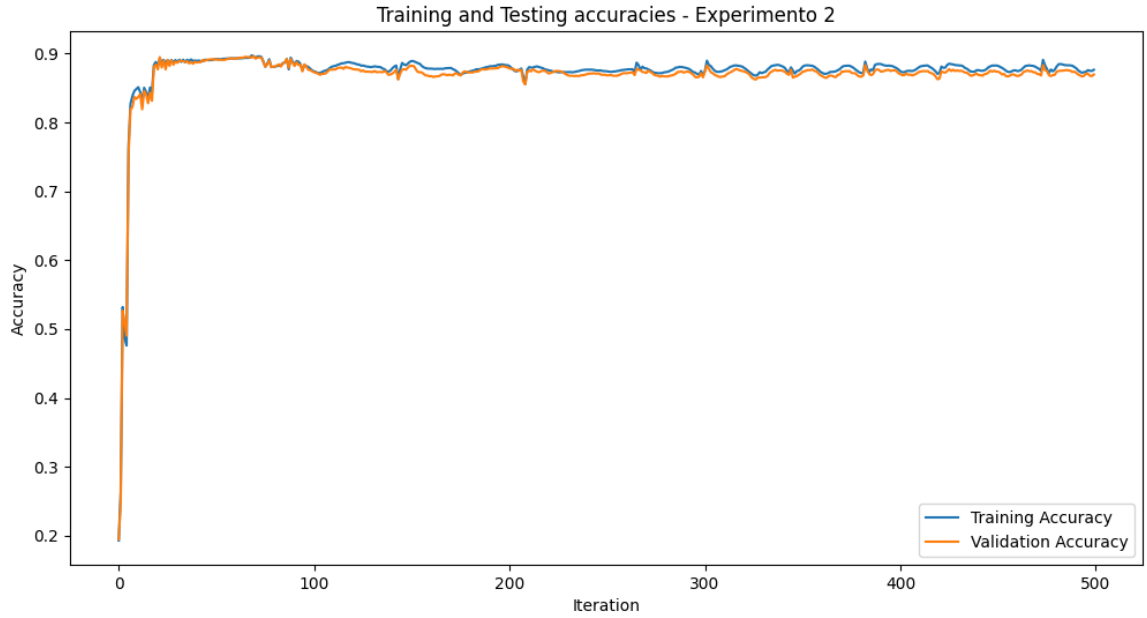


Figura 14 – Acurácia por época do Experimento 2.

Figura 15 – Perda por época do Experimento 2.

ANEXO C – EXPERIMENTO 3

No experimento 3 foi utilizado kNN como método de imputação de valores faltantes, 3σ como método de detecção de *outliers*, *Z-Score* como método de normalização e kNN como método de classificação. A Figura 16 mostra a matriz de confusão do Experimento 3.

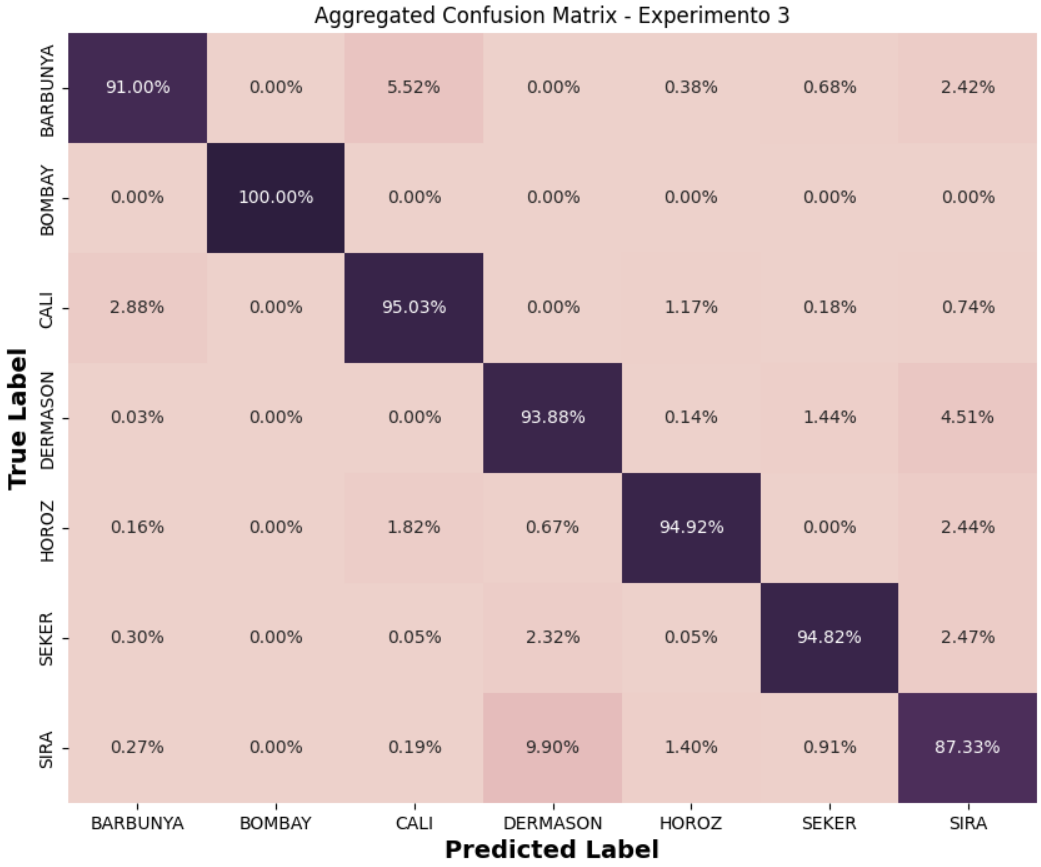


Figura 16 – Matriz de Confusão do Experimento 3.

ANEXO D – EXPERIMENTO 4

No experimento 4 foi utilizado kNN como método de imputação de valores faltantes, 3σ como método de detecção de *outliers*, *Z-Score* como método de normalização e *Multi Layer Perceptron* como método de classificação. A Figura 17 mostra a matriz de confusão do Experimento 4, enquanto a Figura 18 e a Figura 19 apresentam os gráficos de acurácia e de perda por época.

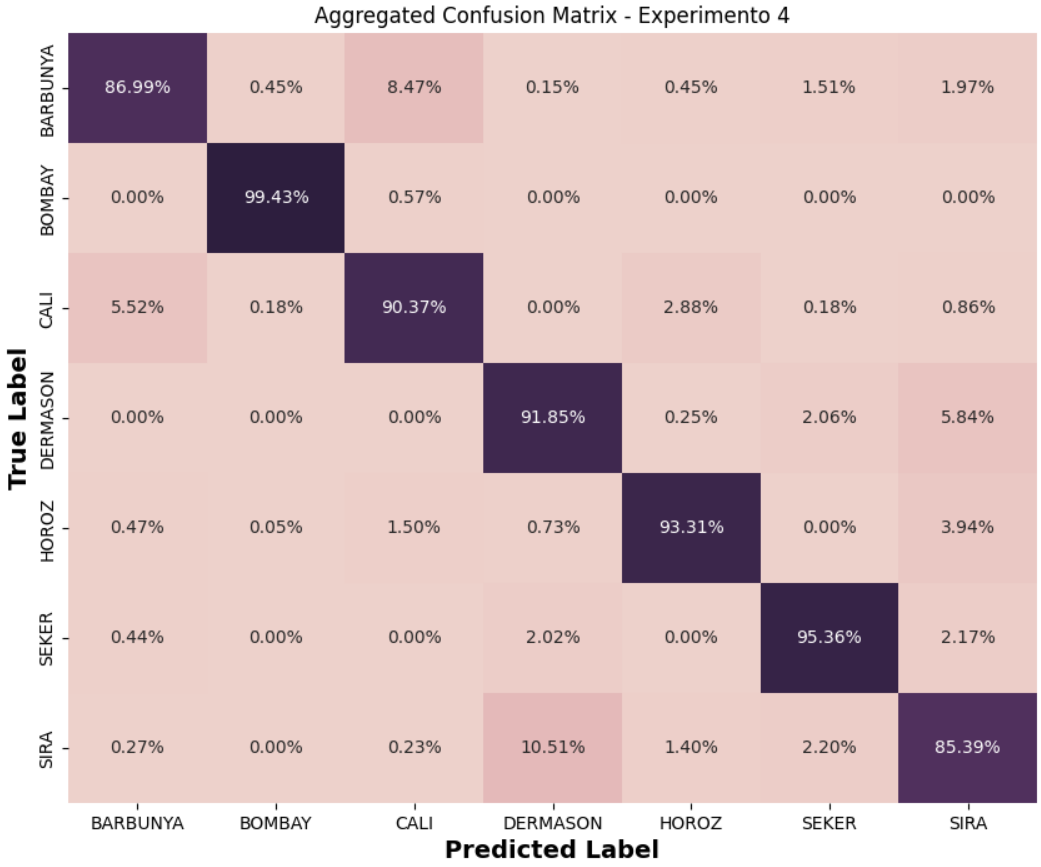


Figura 17 – Matriz de Confusão do Experimento 4.

Figura 18 – Acurácia por época do Experimento 4.

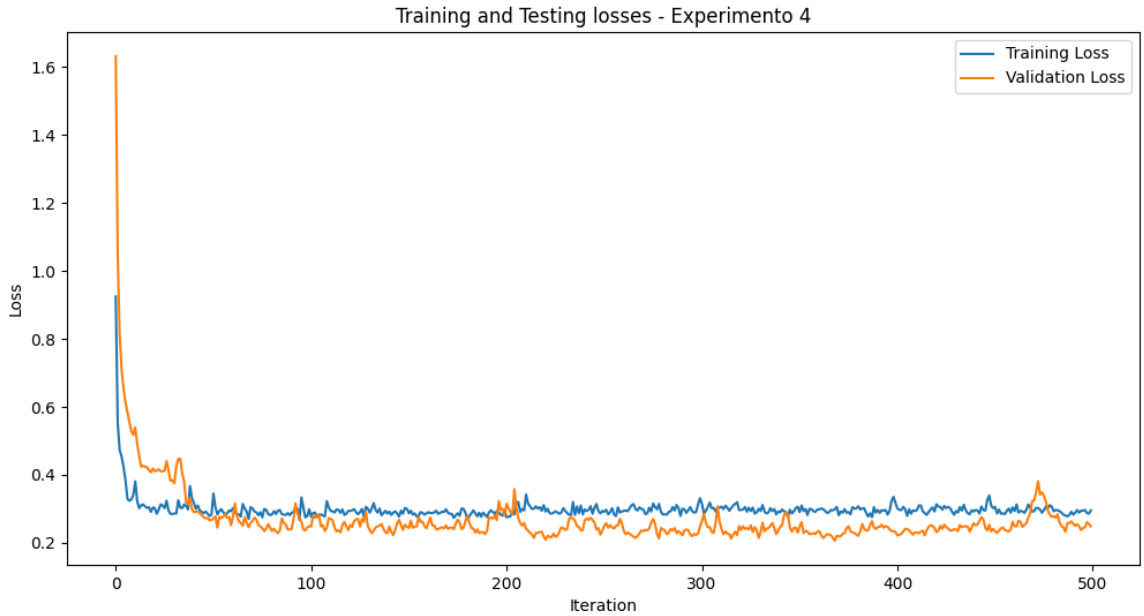


Figura 19 – Perda por época do Experimento 4.

ANEXO E – EXPERIMENTO 5

No experimento 5 foi utilizado *k*NN como método de imputação de valores faltantes, *MAD* como método de detecção de *outliers*, *min-max* como método de normalização e *k*NN como método de classificação. A Figura 20 mostra a matriz de confusão do Experimento 5.

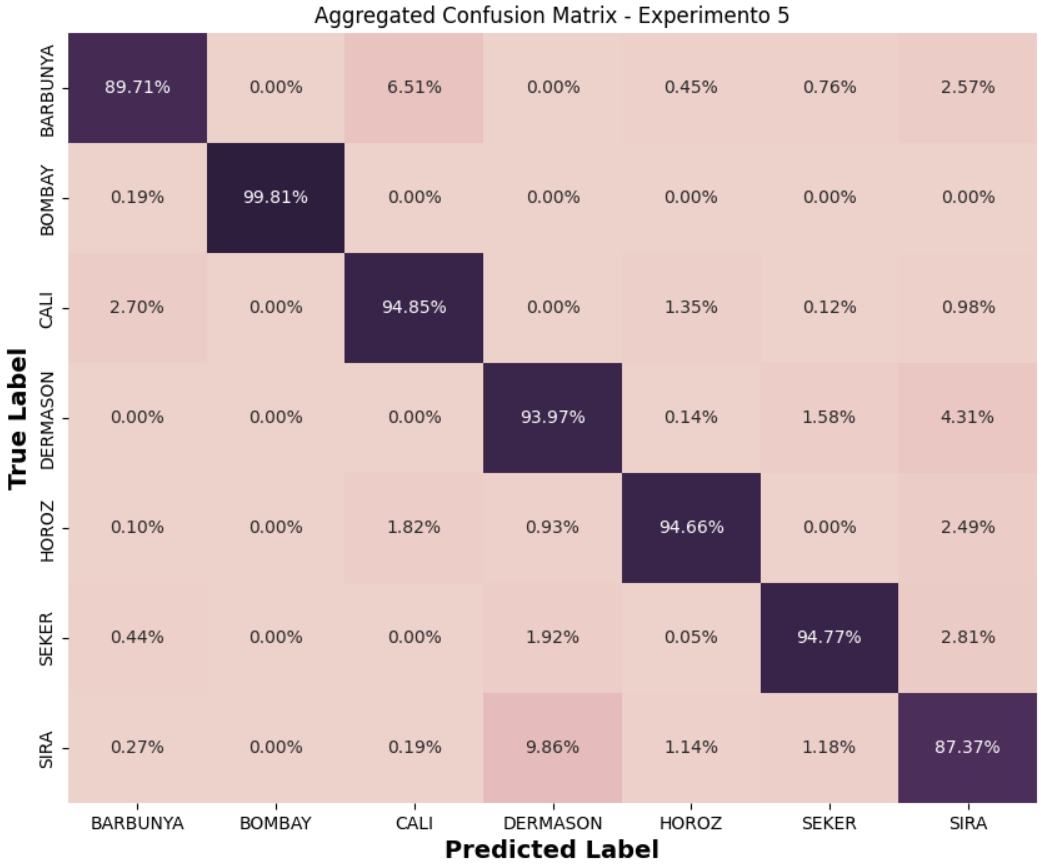


Figura 20 – Matriz de Confusão do Experimento 5.

ANEXO F – EXPERIMENTO 6

No experimento 6 foi utilizado *k*NN como método de imputação de valores faltantes, *MAD* como método de detecção de *outliers*, *min-max* como método de normalização e *Multi Layer Perceptron* como método de classificação. A Figura 21 mostra a matriz de confusão do Experimento 6, enquanto a Figura 22 e a Figura 23 apresentam os gráficos de acurácia e de perda por época.

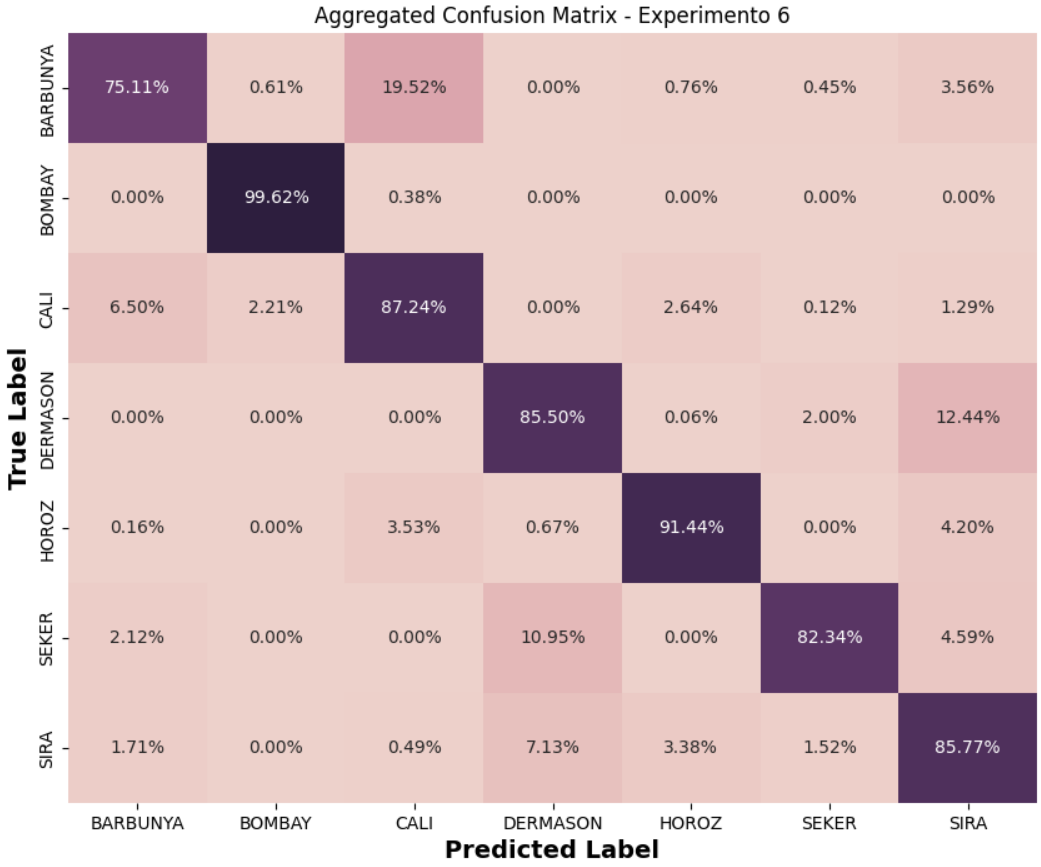


Figura 21 – Matriz de Confusão do Experimento 6.

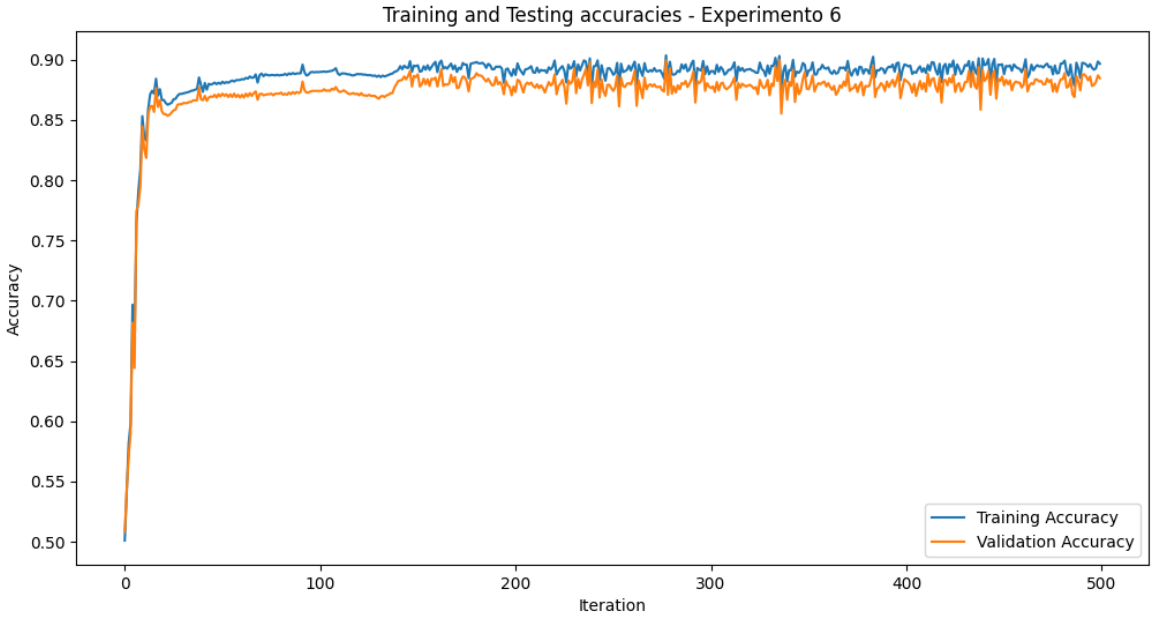


Figura 22 – Acurácia por época do Experimento 6.

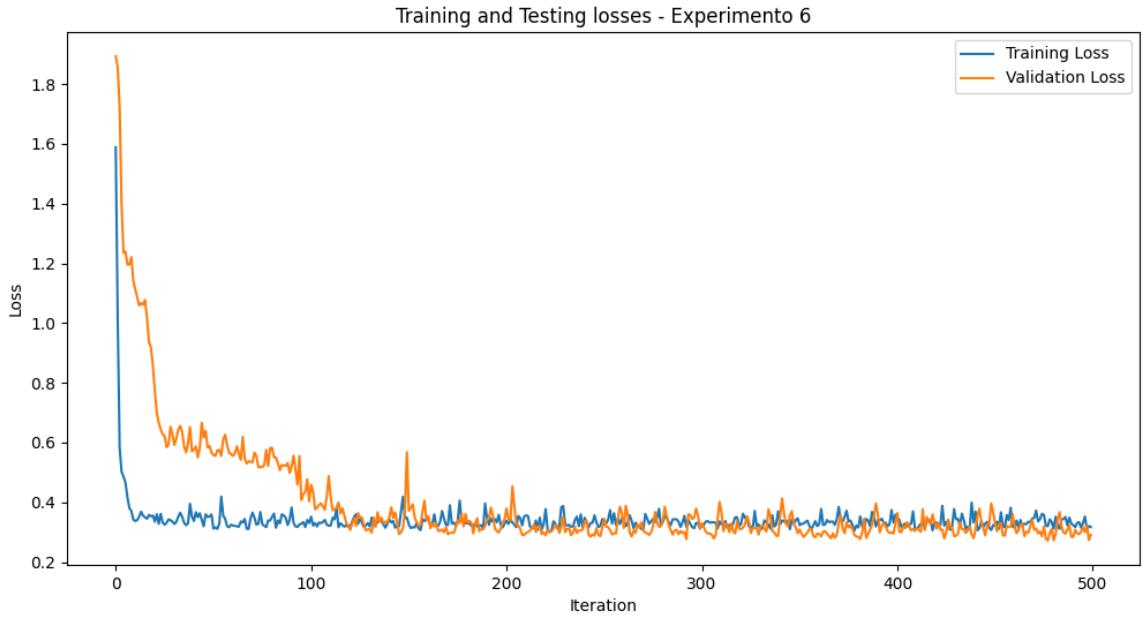


Figura 23 – Perda por época do Experimento 6.

ANEXO G – EXPERIMENTO 7

No experimento 7 foi utilizado *k*NN como método de imputação de valores faltantes, *MAD* como método de detecção de *outliers*, *Z-Score* como método de normalização e *k*NN como método de classificação. A Figura 24 mostra a matriz de confusão do Experimento 7.

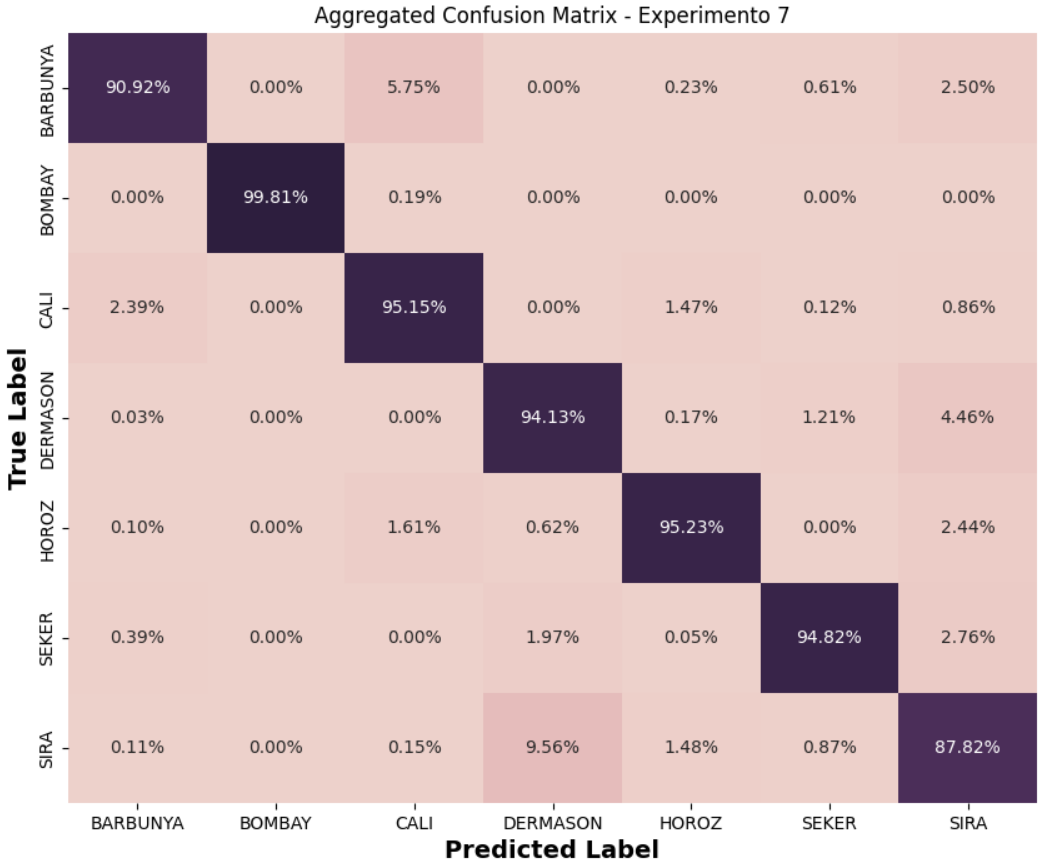


Figura 24 – Matriz de Confusão do Experimento 7.

ANEXO H – EXPERIMENTO 8

No experimento 8 foi utilizado *k*NN como método de imputação de valores faltantes, *MAD* como método de detecção de *outliers*, *Z-Score* como método de normalização e *Multi Layer Perceptron* como método de classificação. A Figura 25 mostra a matriz de confusão do Experimento 8, enquanto a Figura 26 e a Figura 27 apresentam os gráficos de acurácia e de perda por época.

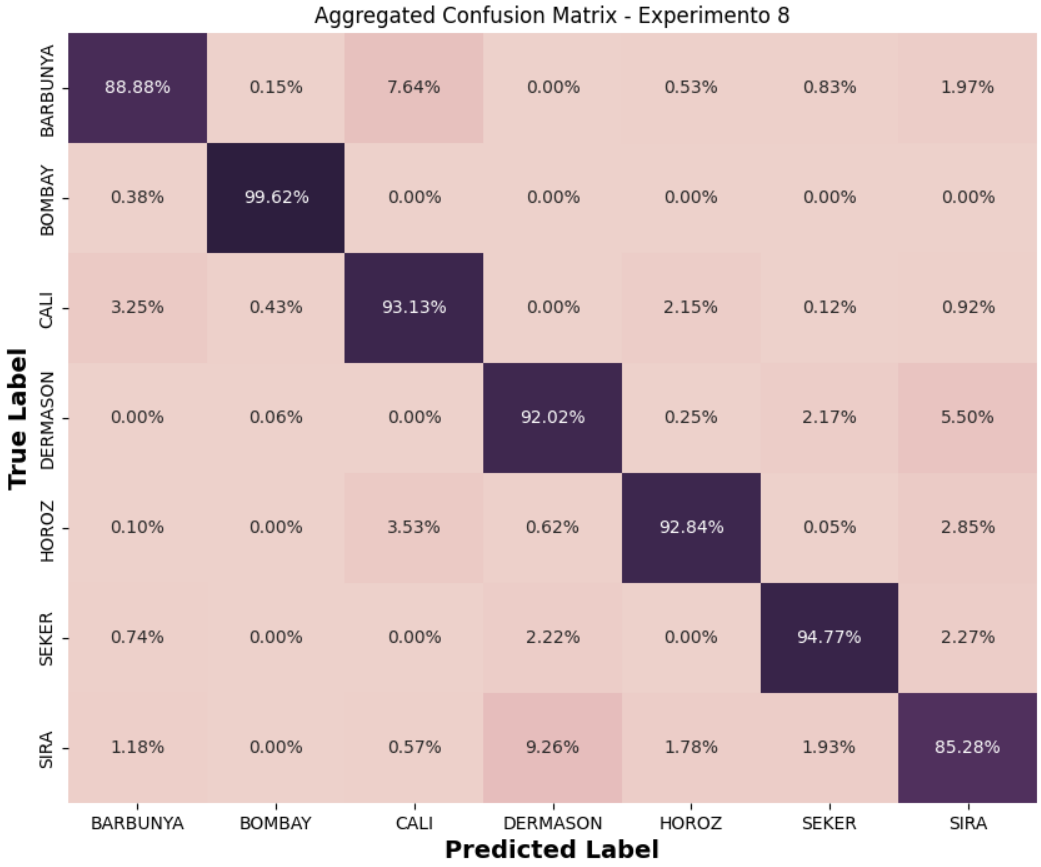


Figura 25 – Matriz de Confusão do Experimento 8.

Figura 26 – Acurácia por época do Experimento 8.

Figura 27 – Perda por época do Experimento 8.

ANEXO I – EXPERIMENTO 9

No experimento 9 foi utilizado Interpolação Linear como método de imputação de valores faltantes, 3σ como método de detecção de *outliers*, *min-max* como método de normalização e kNN como método de classificação. A Figura 28 mostra a matriz de confusão do Experimento 9.

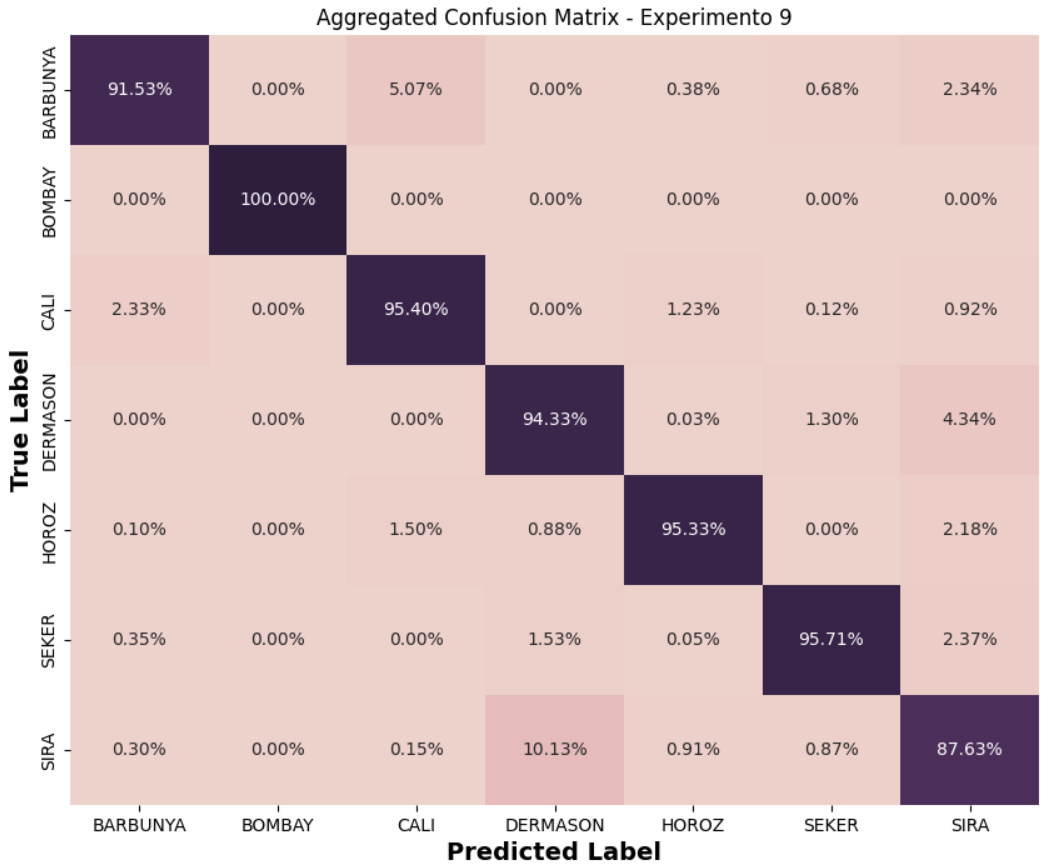


Figura 28 – Matriz de Confusão do Experimento 9.

ANEXO J – EXPERIMENTO 10

No experimento 10 foi utilizado Interpolação Linear como método de imputação de valores faltantes, 3σ como método de detecção de *outliers*, *min-max* como método de normalização e *Multi Layer Perceptron* como método de classificação. A Figura 29 mostra a matriz de confusão do Experimento 8, enquanto a Figura 30 e a Figura 31 apresentam os gráficos de acurácia e de perda por época.

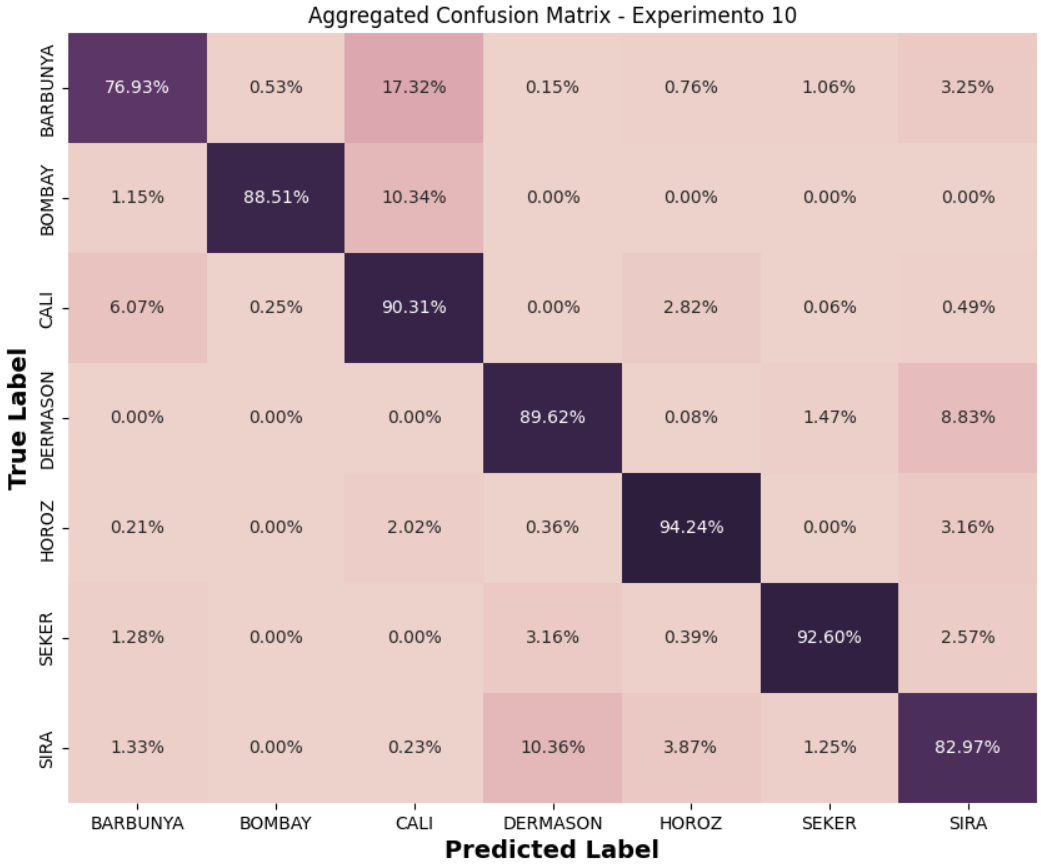


Figura 29 – Matriz de Confusão do Experimento 10.

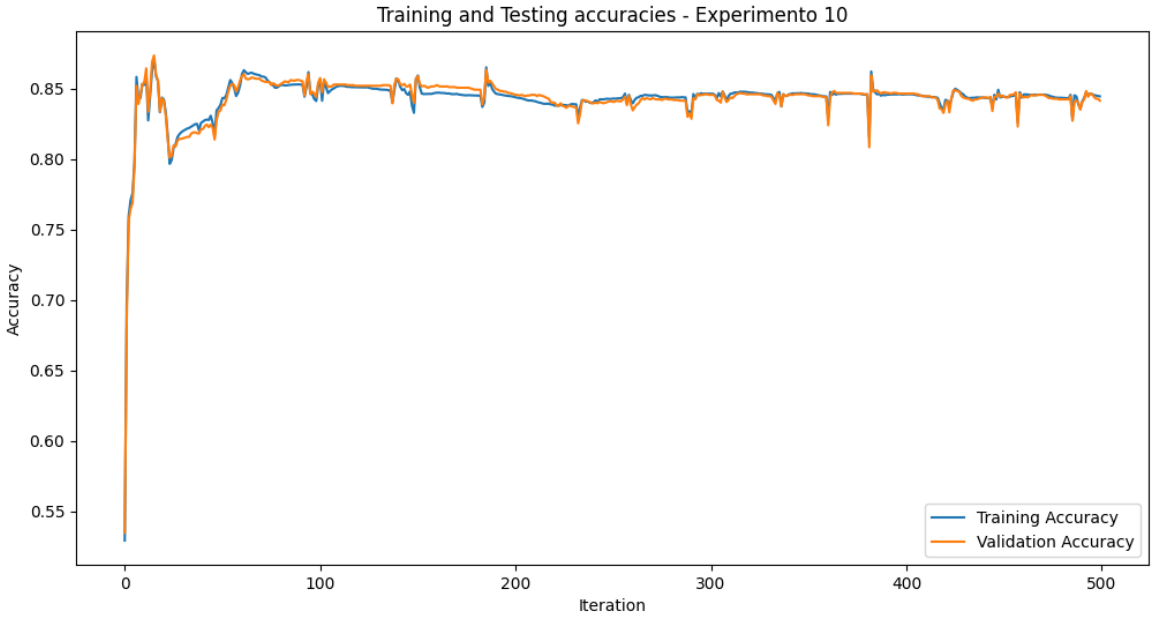


Figura 30 – Acurácia por época do Experimento 10.

Figura 31 – Perda por época do Experimento 10.

ANEXO K – EXPERIMENTO 11

No experimento 11 foi utilizado Interpolação Linear como método de imputação de valores faltantes, 3σ como método de detecção de *outliers*, *Z-Score* como método de normalização e kNN como método de classificação. A Figura 32 mostra a matriz de confusão do Experimento 11.

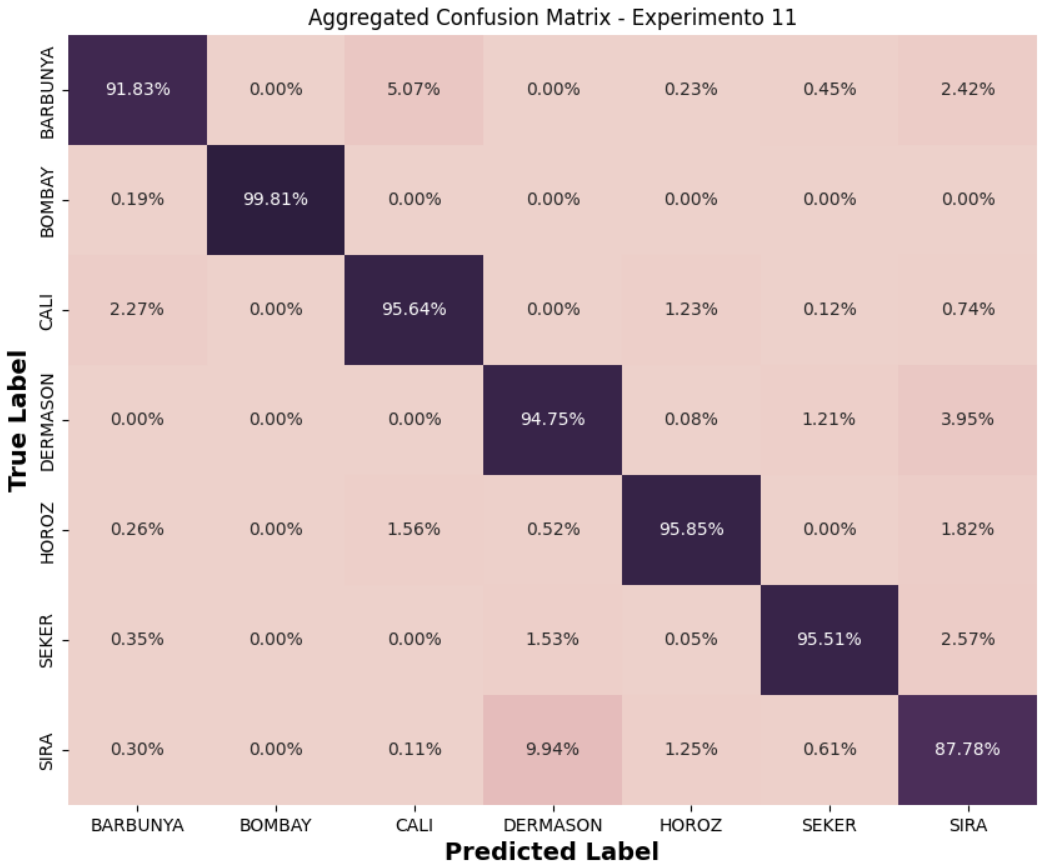


Figura 32 – Matriz de Confusão do Experimento 11.

ANEXO L – EXPERIMENTO 12

No experimento 12 foi utilizado Interpolação Linear como método de imputação de valores faltantes, 3σ como método de detecção de *outliers*, *Z-Score* como método de normalização e *Multi Layer Perceptron* como método de classificação. A Figura 33 mostra a matriz de confusão do Experimento 8, enquanto a Figura 34 e a Figura 35 apresentam os gráficos de acurácia e de perda por época.

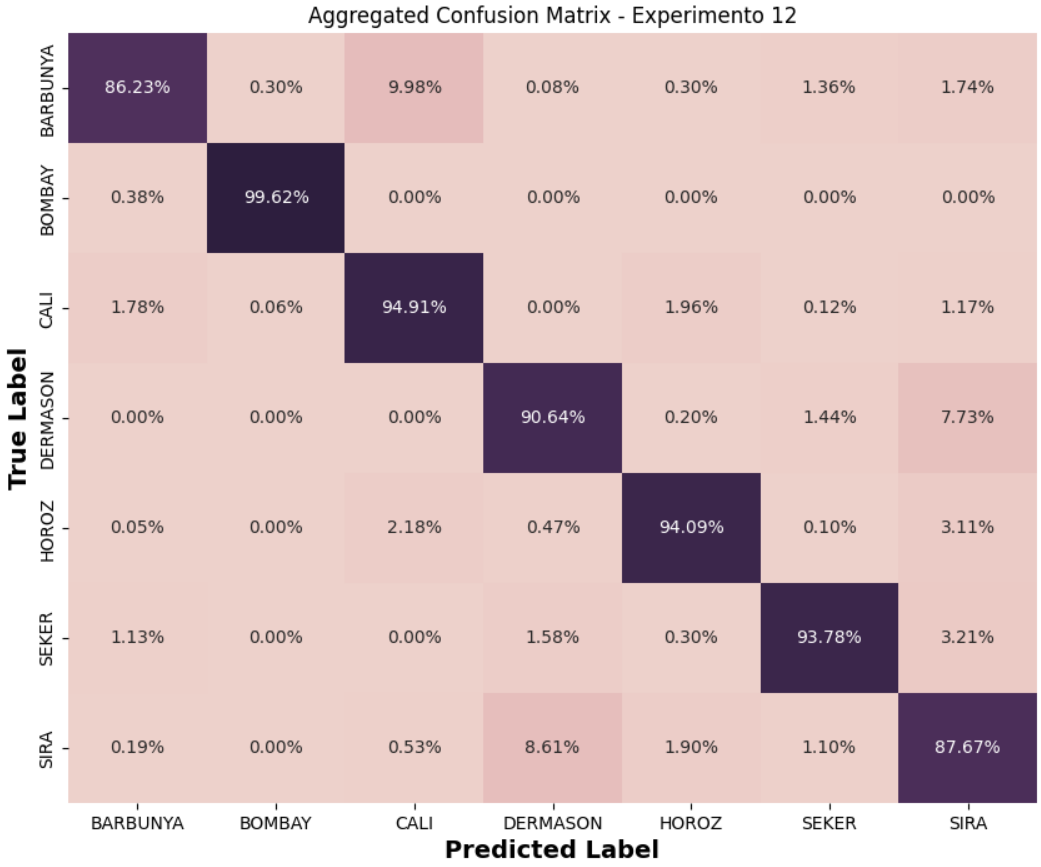


Figura 33 – Matriz de Confusão do Experimento 12.

Figura 34 – Acurácia por época do Experimento 12.

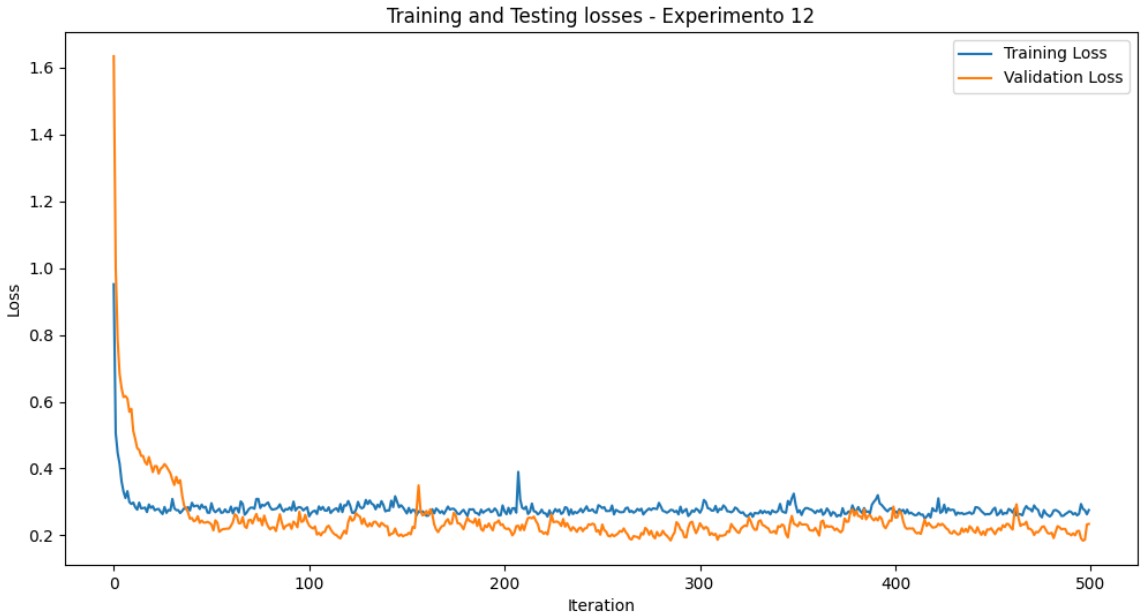


Figura 35 – Perda por época do Experimento 12.

ANEXO M – EXPERIMENTO 13

No experimento 13 foi utilizado Interpolação Linear como método de imputação de valores faltantes, MAD como método de detecção de *outliers*, *min-max* como método de normalização e kNN como método de classificação. A Figura 36 mostra a matriz de confusão do Experimento 13.

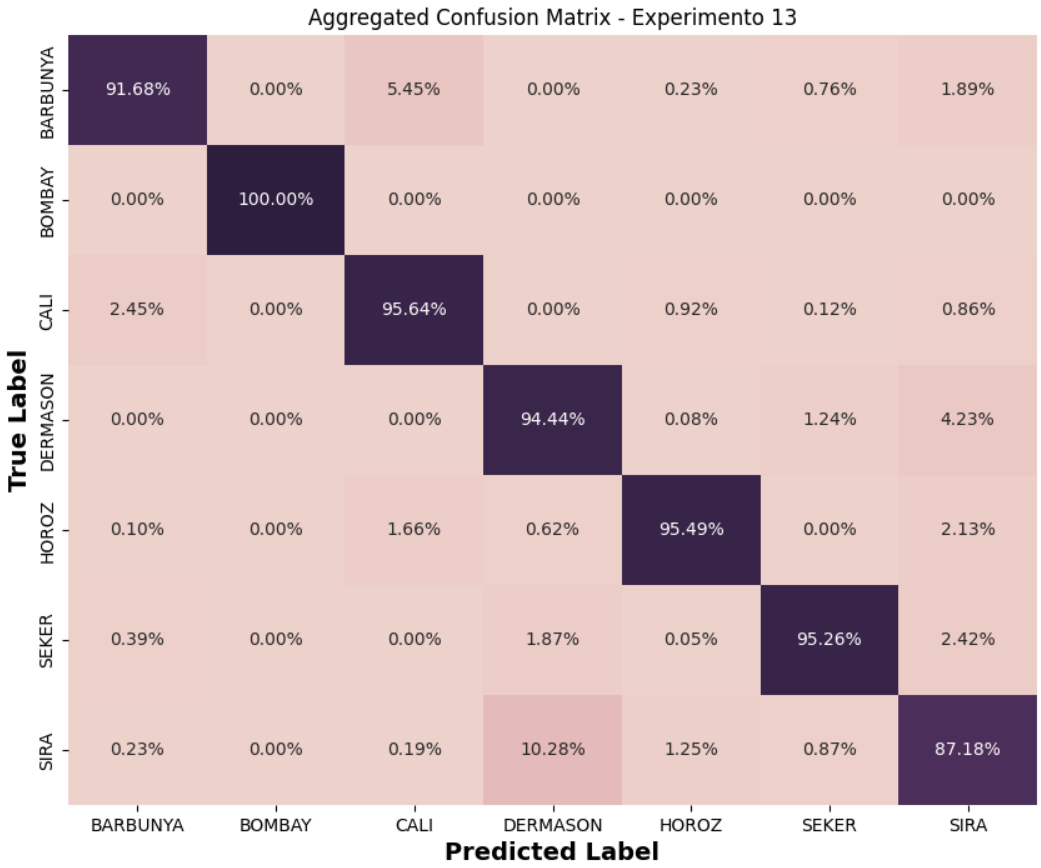


Figura 36 – Matriz de Confusão do Experimento 13.

ANEXO N – EXPERIMENTO 14

No experimento 14 foi utilizado Interpolação Linear como método de imputação de valores faltantes, MAD como método de detecção de *outliers*, *min-max* como método de normalização e *Multi Layer Perceptron* como método de classificação. A Figura 37 mostra a matriz de confusão do Experimento 4, enquanto a Figura 38 e a Figura 39 apresentam os gráficos de acurácia e de perda por época.

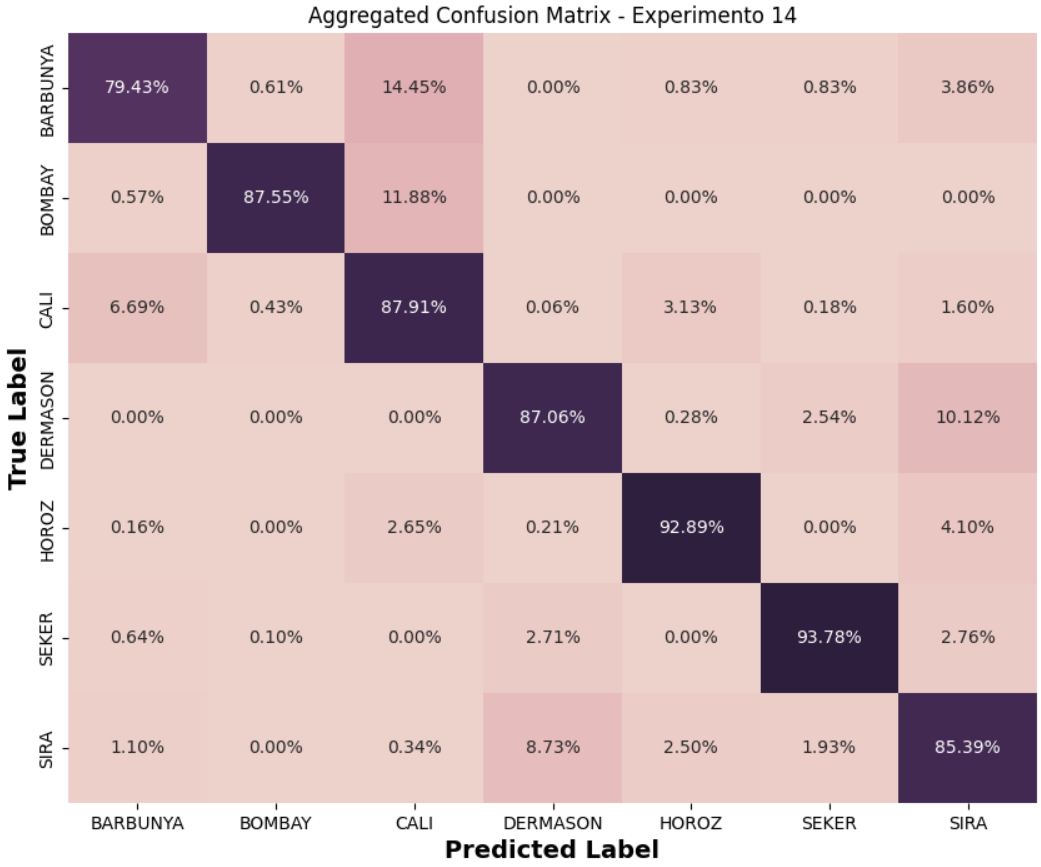


Figura 37 – Matriz de Confusão do Experimento 14.

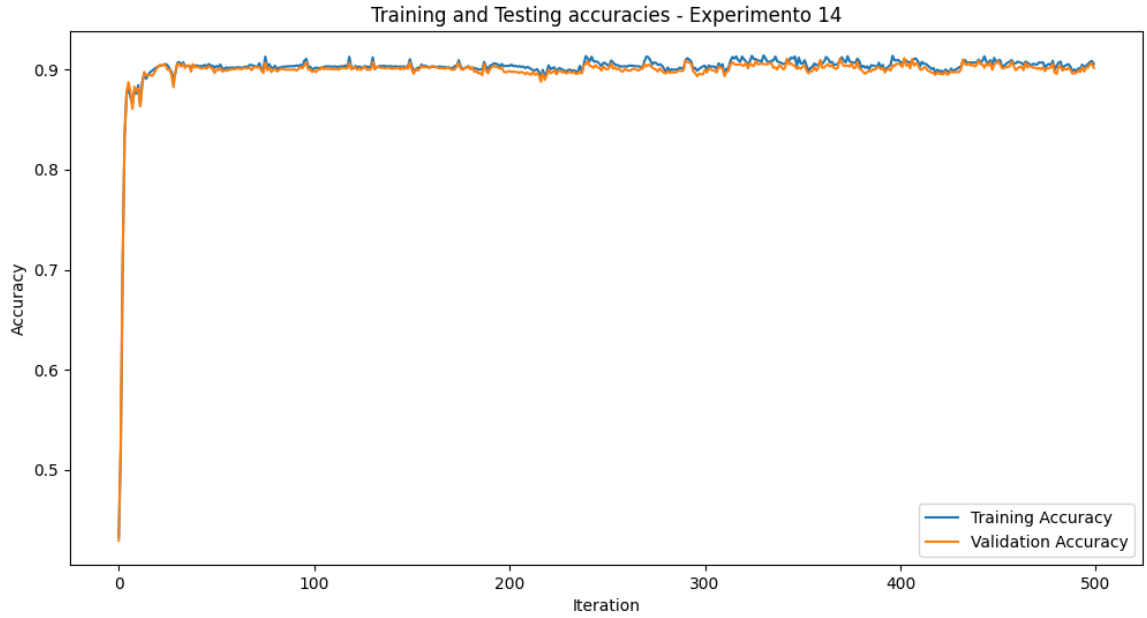


Figura 38 – Acurácia por época do Experimento 14.

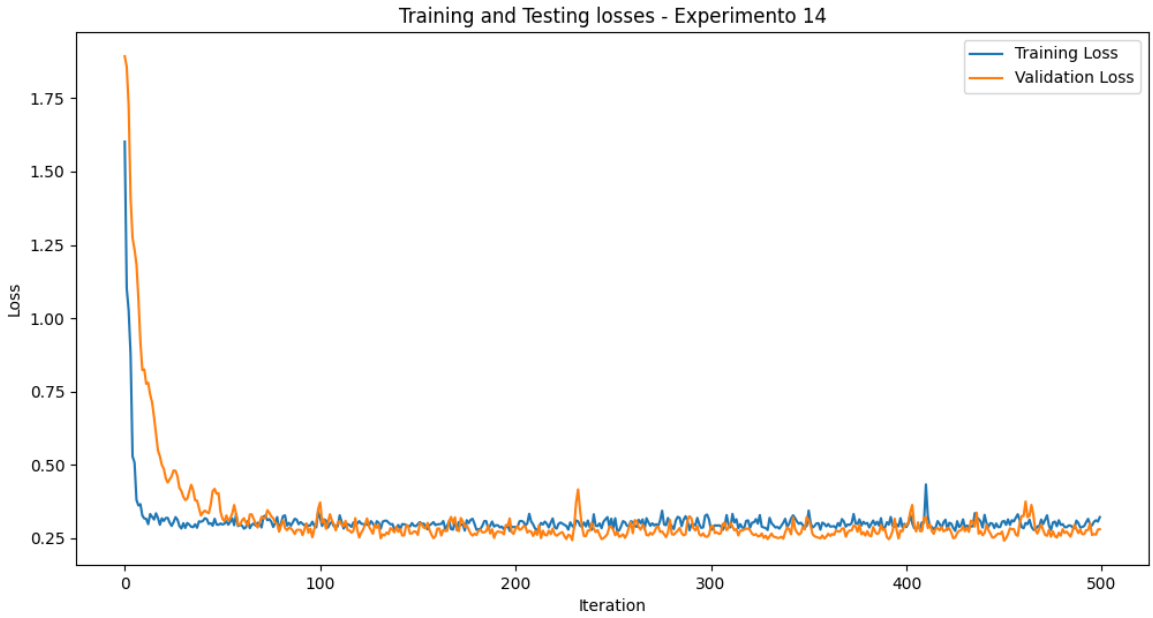


Figura 39 – Perda por época do Experimento 14.

ANEXO O – EXPERIMENTO 15

No experimento 15 foi utilizado Interpolação Linear como método de imputação de valores faltantes, MAD como método de detecção de *outliers*, *min-max* como método de normalização e kNN como método de classificação. A Figura 40 mostra a matriz de confusão do Experimento 15.

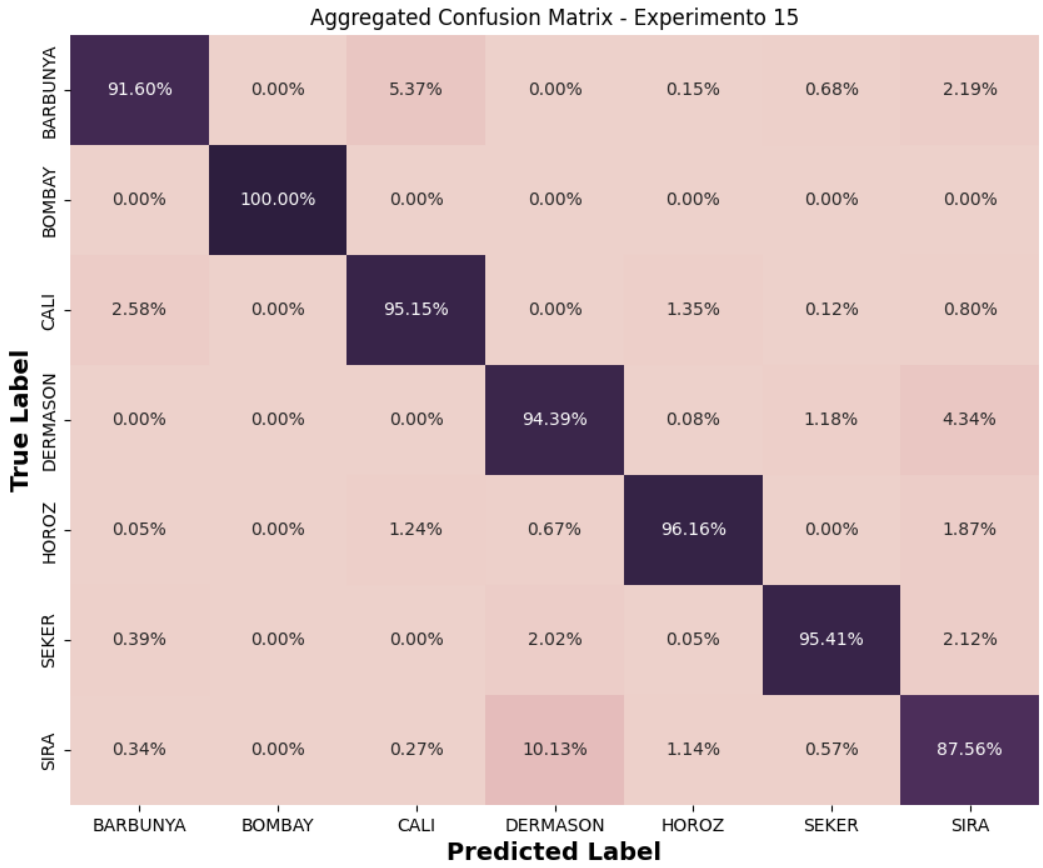


Figura 40 – Matriz de Confusão do Experimento 15.

ANEXO P – EXPERIMENTO 16

No experimento 16 foi utilizado Interpolação Linear como método de imputação de valores faltantes, MAD como método de detecção de *outliers*, *min-max* como método de normalização e *Multi Layer Perceptron* como método de classificação. A Figura 41 mostra a matriz de confusão do Experimento 16, enquanto a Figura 42 e a Figura 43 apresentam os gráficos de acurácia e de perda por época.

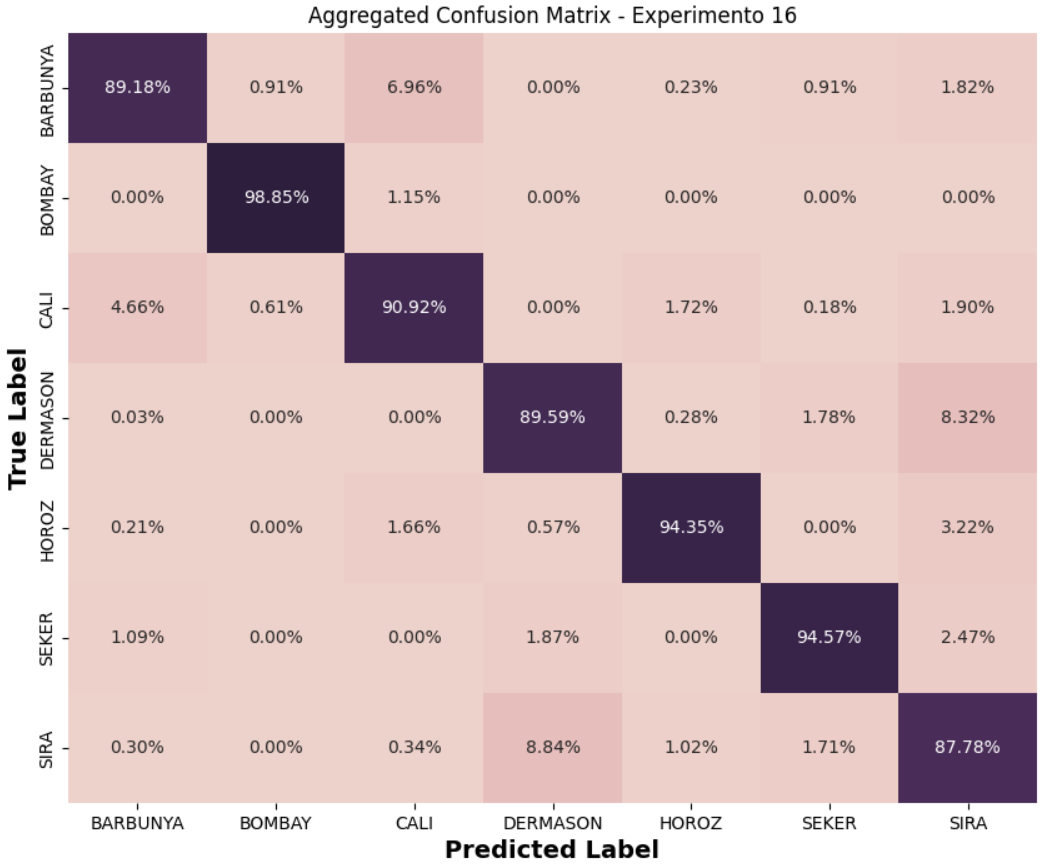


Figura 41 – Matriz de Confusão do Experimento 16.

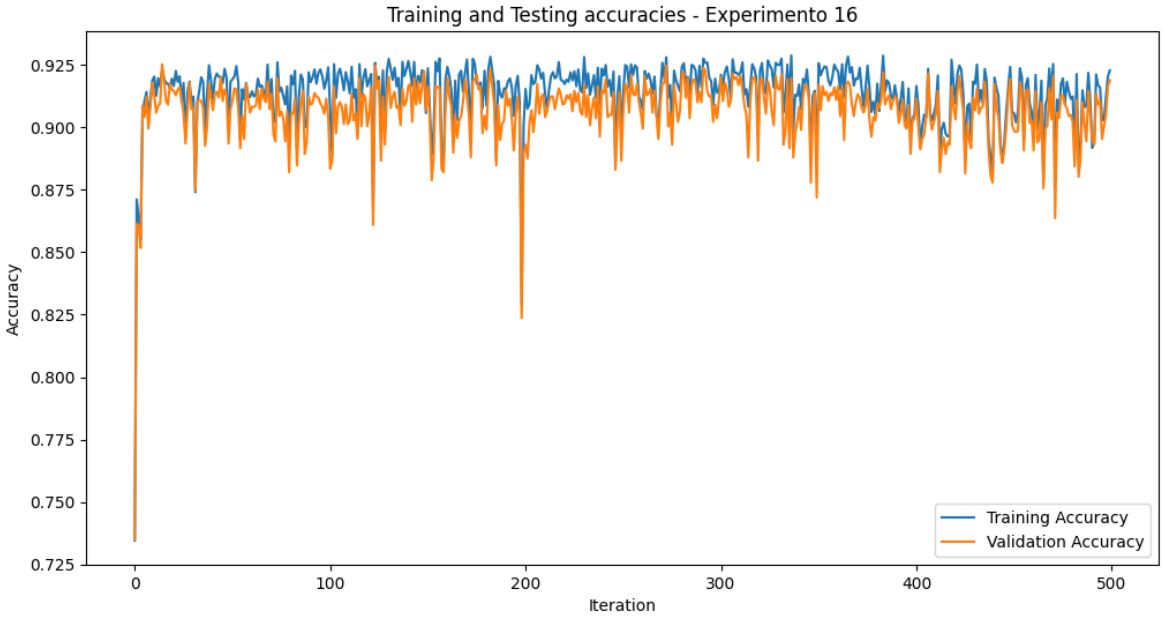


Figura 42 – Acurácia por época do Experimento 16.

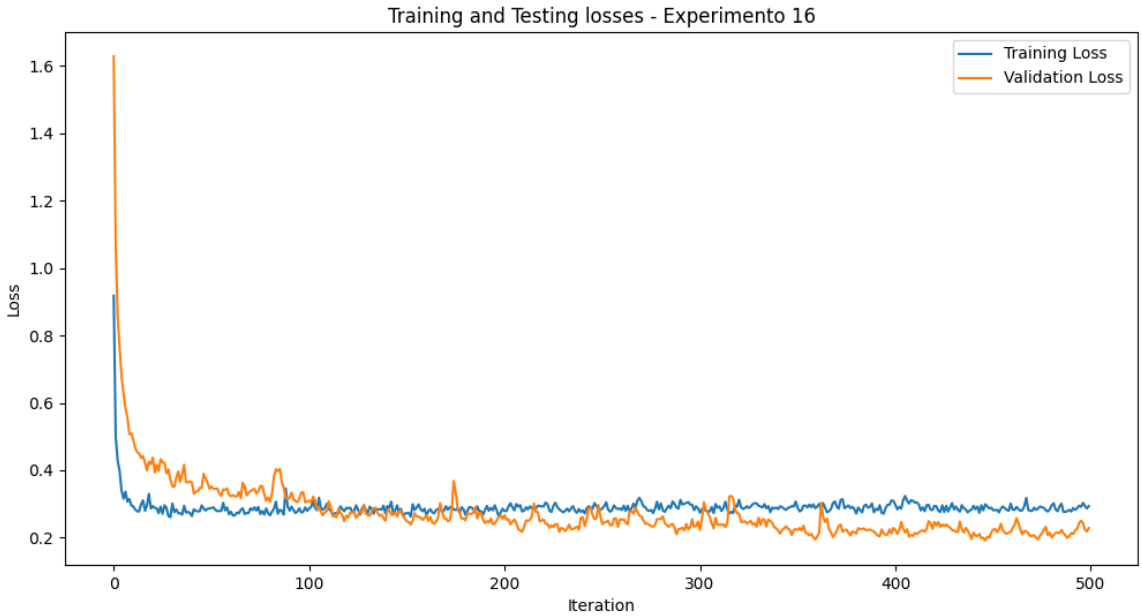


Figura 43 – Perda por época do Experimento 16.

ANEXO Q – KOKLU KNN

Nessa repetição do experimento descrito por Koklu e Ozkan (KOKLU; OZKAN, 2020), os dados brutos foram alimentados diretamente para o algoritmo de classificação kNN. A utilização de dados brutos geralmente não é indicada, provada pelo resultado atingido pela classificação desses dados. A Figura 44 apresenta a matriz de confusão resultante da classificação.

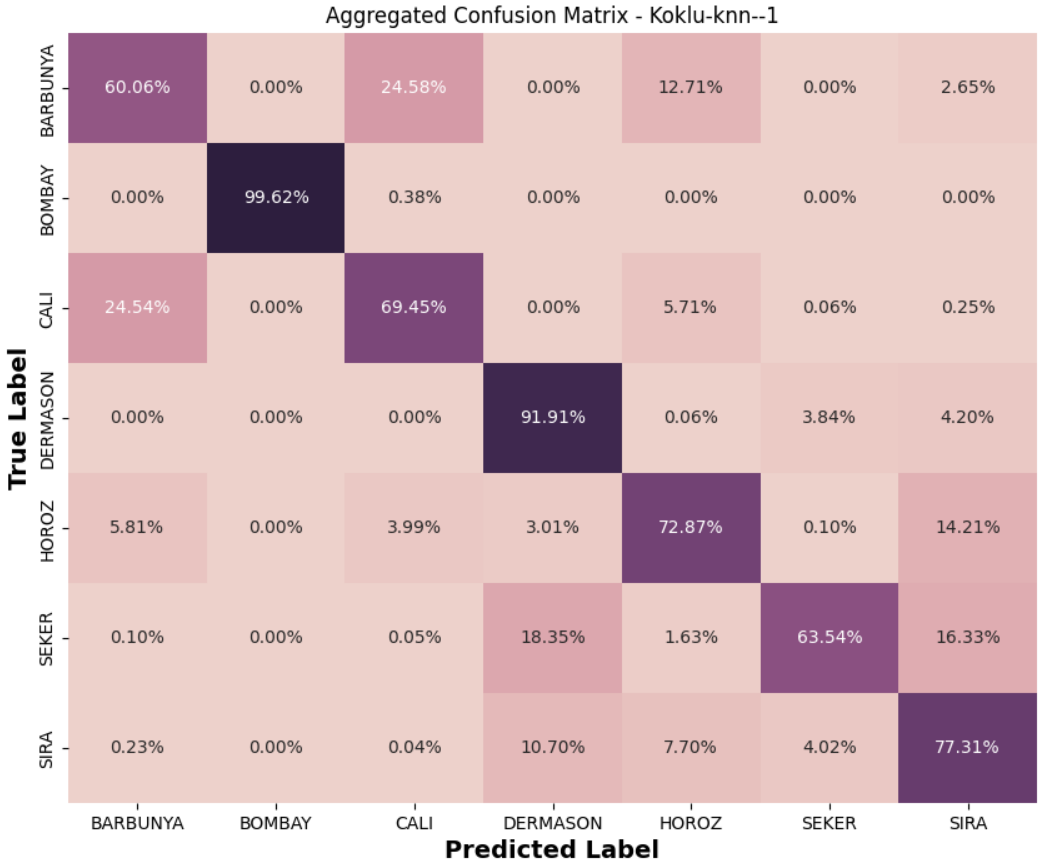


Figura 44 – Matriz de Confusão da Repetição do Experimento de Koklu KNN.

ANEXO R – KOKLU KNN COM NORMALIZAÇÃO

Nessa repetição do experimento descrito por Koklu e Ozkan (KOKLU; OZKAN, 2020), os dados brutos foram noramlizados antes de serem alimentados para o algoritmo de classificação kNN. Com os dados normalizados, o kNN conseguiu classificar de maneira melhor os dados. A Figura 45 apresenta a matriz de confusão resultante da classificação.

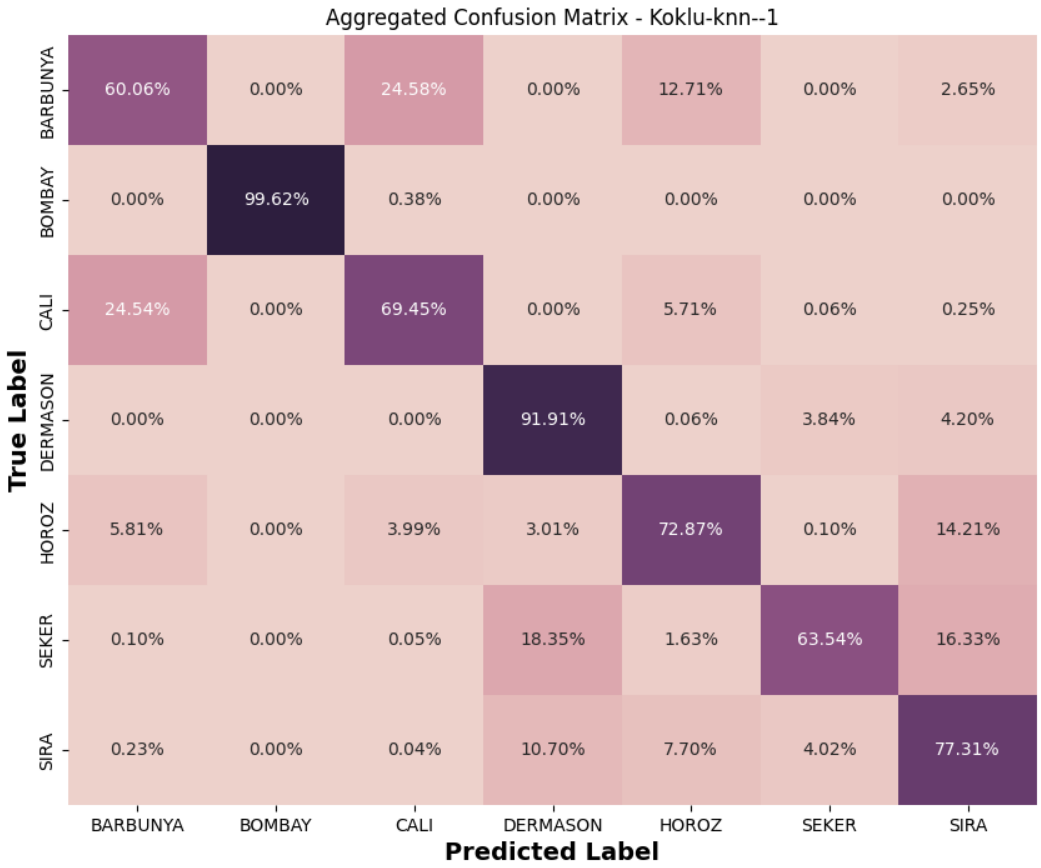


Figura 45 – Matriz de Confusão da Repetição do Experimento de Koklu KNN com Normalização.

ANEXO S – KOKLU MLP

Nessa repetição do experimento descrito por Koklu e Ozkan (KOKLU; OZKAN, 2020), os dados brutos foram noramlizados antes de serem alimentados para o algoritmo de classificação MLP. A utilização de dados brutos geralmente não é indicada, provada pelo resultado atingido pela classificação desses dados. A Figura 46 apresenta a matriz de confusão resultante da classificação, a Figura 47 apresenta o gráfico de acurácia por época (para treinamento e teste) e a Figura 48 apresenta o gráfico de perda por época (para treinamento e teste).

É possível percebem que com os dados não normalizados a rede neural apenas "chutou"uma classe para todos os dados recebidos, assim conseguindo classificar apenas uma das classes corretamente. Esse comportamente é facilmente percebido pela matriz de confusão e pelo comportamento dos gráficos de perda e de acurácia.

Figura 46 – Matriz de Confusão da Repetição do Experimento de Koklu MLP.

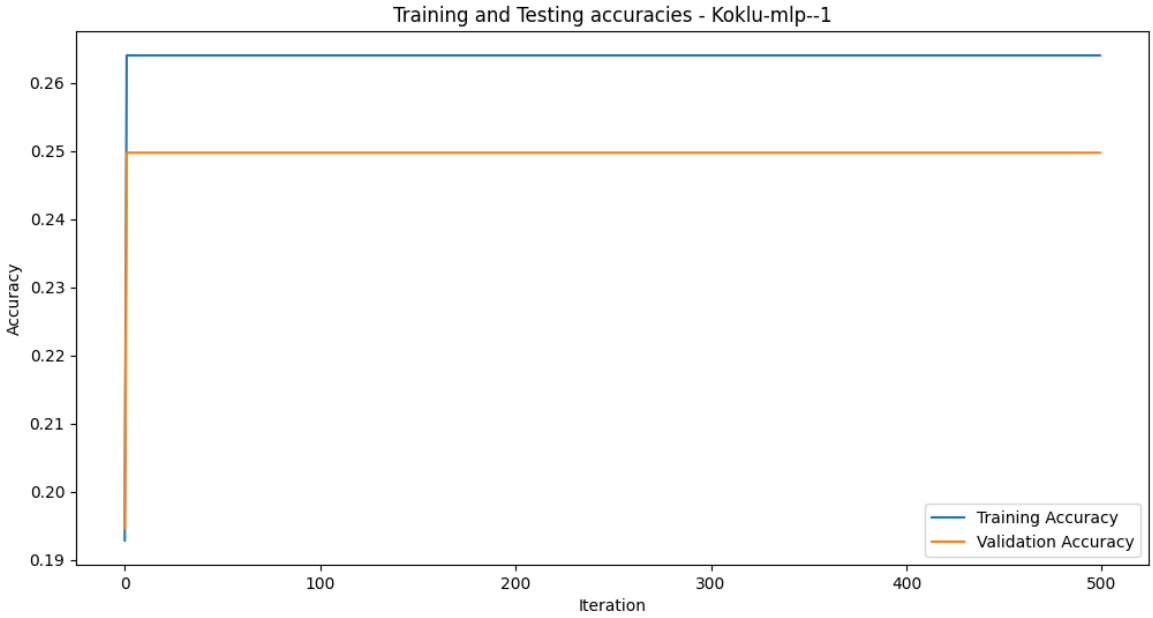


Figura 47 – Acurácia por época do Experimento de Koklu MLP.

Figura 48 – Perda por época do Experimento de Koklu MLP.

ANEXO T – KOKLU MLP COM NORMALIZAÇÃO

Nessa repetição do experimento descrito por Koklu e Ozkan (KOKLU; OZKAN, 2020), os dados brutos foram noramlizados antes de serem alimentados para o algoritmo de classificação MLP. A classificação com os dados já normalizados apresenta um resultado melhor, mas que poderia ser ainda melhor caso outras técnicas de pré-processamento de dados fossem aplicadas. A Figura 49 apresenta a matriz de confusão resultante da classificação, a Figura 50 apresenta o gráfico de acurácia por época (para treinamento e teste) e a Figura 51 apresenta o gráfico de perda por época (para treinamento e teste).

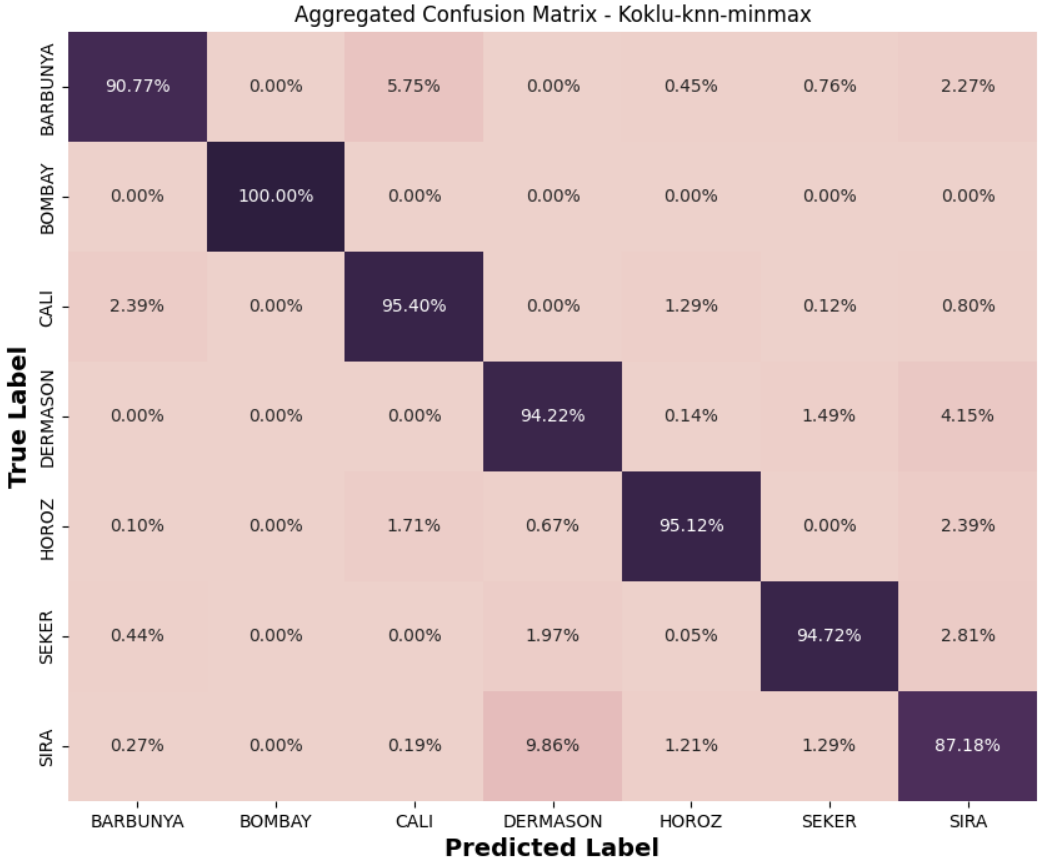


Figura 49 – Matriz de Confusão da Repetição do Experimento de Koklu MLP com Normalização.

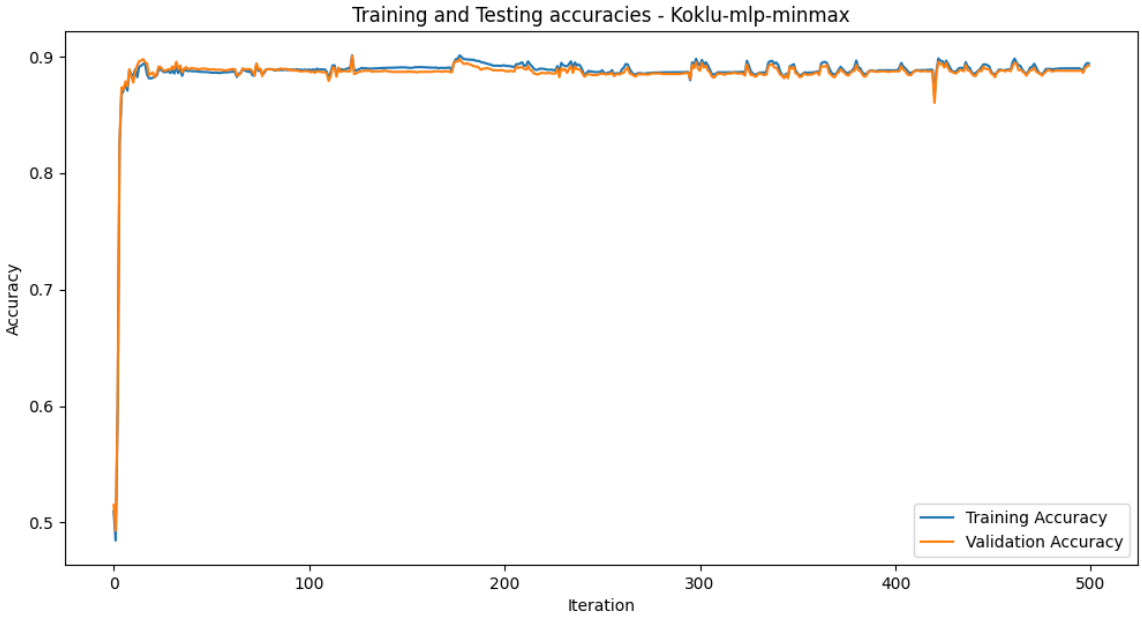


Figura 50 – Acurácia por época do Experimento de Koklu MLP com Normalização.

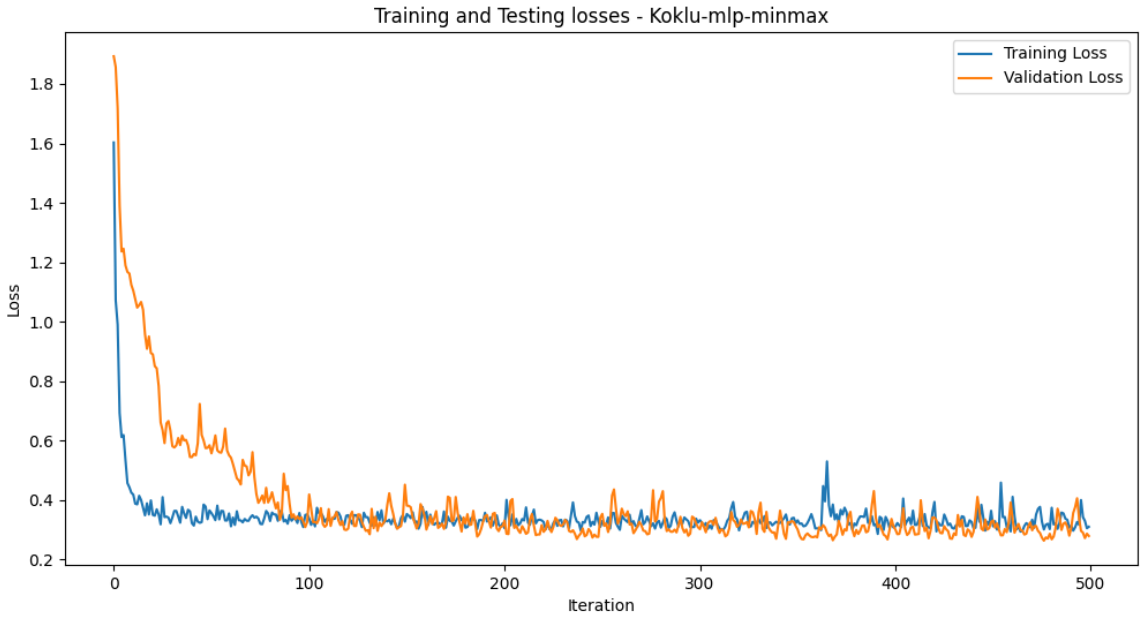


Figura 51 – Perda por época do Experimento de Koklu MLP como Normalização.