Serviço Nacional de Seguro de Saúde da Coreia

Jian Licio de Oliveira*

[1] Ciência da Computação – Universidade Tecnológica Federal do Paraná (UTFPR)
Medianeira – PR – Brasil

Oliveira, Jian Licio jian.licio@gmail.com

Abstract. *This study aims to analyze and classify individuals who consume alcoholic beverages using a dataset collected by the National Health Insurance Service of Korea. To achieve this, various machine learning algorithms were employed, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Trees, Naive Bayes, Perceptron, and Linear Regression. The dataset underwent several preprocessing steps, such as normalization and duplicate removal, to assess their impact on model performance.*
Among the experiments conducted, the KNN algorithm stood out as the best approach, with the optimal value of k=126k=126 identified using the Euclidean distance metric, achieving an accuracy of 71.61%. To determine the best kk, values ranging from 30 to 300 were tested in increments of 10, followed by a fine-tuning process within the range of 100 to 140. The comparative analysis of the algorithms revealed that adjustments in hyperparameters and data preprocessing significantly influence model performance, highlighting the importance of an experimental approach to optimizing classification.

Resumo. *Este estudo tem como objetivo a análise e classificação de indivíduos consumidores de bebidas alcoólicas a partir de um conjunto de dados coletado pelo Serviço Nacional de Seguro de Saúde da Coreia. Para isso, foram utilizados diferentes algoritmos de aprendizado de máquina, incluindo K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Árvores de Decisão, Naive Bayes, Perceptron e Regressão Linear. Os dados passaram por diversas etapas de pré-processamento, como normalização e remoção de duplicatas, para avaliar seu impacto no desempenho dos modelos.*
Dentre os experimentos realizados, o algoritmo KNN se destacou como a melhor abordagem, sendo identificado o valor ideal de k=126k=126 utilizando a métrica de distância Euclidiana, alcançando uma acurácia de 71,61%. Para a seleção do melhor kk, foram testados valores variando de 30 a 300 em incrementos de 10, e posteriormente, um ajuste fino no intervalo de 100 a 140. A análise comparativa dos algoritmos revelou que ajustes nos hiperparâmetros e no pré-processamento dos dados influenciam significativamente a performance dos modelos, reforçando a importância de uma abordagem experimental para otimização da classificação.

1. Definição

O presente artigo tem como base um conjunto de dados coletado pelo Serviço Nacional de Seguro de Saúde da Coreia. Para garantir a privacidade dos indivíduos, todas as informações pessoais e dados sensíveis foram removidos. Esse banco de dados é utilizado

*Prof. Dr. Jorge Aikes Junior, Universidade Tecnológica Federal do Paraná (UTFPR).

para a análise de sinais corporais e a classificação de determinados hábitos, como o consumo de tabaco e bebidas alcoólicas. No contexto deste estudo, o objetivo principal é identificar e classificar indivíduos que consomem bebidas alcoólicas, utilizando os atributos disponíveis no conjunto de dados. A seguir, se encontra a lista de atributos iniciais disponibilizadas pelo Serviço Nacional de Seguro de Saúde da Coreia:

- `Sex`: Masculino, Feminino
- `Age`: Idade arredondada para cima a cada 5 anos
- `Height`: Altura arredondada para cima a cada 5 cm [cm]
- `Weight`: Peso [kg]
- `Sight_left`: Visão (olho esquerdo)
- `Sight_right`: Visão (olho direito)
- `Hear_left`: Audição (ouvido esquerdo), 1 (normal), 2 (anormal)
- `Hear_right`: Audição (ouvido direito), 1 (normal), 2 (anormal)
- `SBP`: Pressão arterial sistólica [mmHg]
- `DBP`: Pressão arterial diastólica [mmHg]
- `BLDS`: BLDS ou FSG (glicose em jejum) [mg/dL]
- `Tot_chole`: Colesterol total [mg/dL]
- `HDL_chole`: Colesterol HDL [mg/dL]
- `LDL_chole`: Colesterol LDL [mg/dL]
- `Triglyceride`: Triglicerídeos [mg/dL]
- `Hemoglobin`: Hemoglobina [g/dL]
- `Urine_protein`: Proteína na urina, 1 (-), 2 (+/-), 3 (+1), 4 (+2), 5 (+3), 6 (+4)
- `serum_creatinine`: Creatinina no sangue [mg/dL]
- `SGOT_AST`: SGOT (Glutamato-oxaloacetato transaminase) AST (Aspartato transaminase) [IU/L]
- `SGOT_ALT`: ALT (Alanina transaminase) [IU/L]
- `Gamma_GTP`: y-Glutamil transpeptidase [IU/L]
- `SMK_stat_type_cd`: Estado de fumante, 1 (nunca), 2 (fumava, mas parou), 3 (ainda fuma)
- `DRK_YN`: Bebe ou não

2. Seleção

A inclusão de atributos irrelevantes ou redundantes pode aumentar o tempo de processamento, sobrecarregar algoritmos e levar a resultados menos precisos, além de dificultar a interpretação dos dados. Por outro lado, a escolha criteriosa dos atributos permite reduzir a complexidade do modelo, melhorar sua capacidade de generalização e destacar os fatores realmente relevantes para a análise. Assim, a seleção de atributos não apenas otimiza os recursos computacionais, mas também contribui para uma maior qualidade e confiabilidade dos insights gerados.

2.1. Atributos Removidos

Neste estágio, a remoção de atributos de baixa ou nenhuma relevância é uma etapa essencial. No entanto, para este conjunto de dados, todos os atributos foram inicialmente considerados importantes para a análise. Por essa razão, até o momento, nenhum atributo foi excluído.

3. Pré-processamento

Consiste em preparar dados brutos em um formato adequado para análise, envolvendo tarefas como remoção de ruídos (remoção de valores nulos ou inconsistentes). Sua importância está no fato de que dados de baixa qualidade podem comprometer a precisão e a eficácia dos modelos, pois algoritmos de aprendizado de máquina são sensíveis a inconsistências nos dados. Um pré-processamento bem realizado aprimora a qualidade dos dados e melhora o desempenho do modelo, assegurando que ele capture padrões relevantes e seja capaz de generalizar para novos dados, o que é essencial para obter resultados precisos e confiáveis.

3.1. Normalização dos atributos

Todos os atributos e instâncias que possuíam valores textuais, que possuíam letras maiúsculas, passaram a ter valor minusculo, para facilitar o entendimento e não haver enganos de caracteres.

3.2. Valores Flutuantes

Todos os valores decimais foram convertidos para inteiros, garantindo uniformidade nos dados e evitando inconsistências durante a análise.

3.3. Ruídos nos Dados

No processo de descoberta de conhecimento e aprendizado de máquina, a qualidade dos dados é um fator determinante para a obtenção de modelos confiáveis e eficientes. Dados ruidosos, que incluem valores inconsistentes, atípicos ou corrompidos, podem comprometer significativamente a precisão das análises e a capacidade preditiva dos algoritmos. Esses ruídos podem surgir de diversas fontes, como erros de medição, falhas na coleta de dados, inserções manuais incorretas ou até mesmo registros anômalos que não refletem o comportamento real da população analisada.

A remoção ou correção desses ruídos é uma etapa fundamental no pré-processamento dos dados, pois garante maior integridade ao conjunto de informações utilizado. Ignorar a presença de valores inconsistentes pode levar a interpretações equivocadas, impactando diretamente a tomada de decisões baseada nos modelos gerados. Assim, a identificação e o tratamento adequado dos ruídos são essenciais para aprimorar a qualidade dos resultados obtidos, tornando o processo de extração de conhecimento mais confiável e robusto.

3.3.1. Waistline

Foram identificadas 57 instâncias com o valor de `waistline` igual a 999, claramente um erro, seja intencional ou não. Considerando que o segundo maior valor registrado foi 149, a discrepância é extremamente elevada. Dessa forma, tais valores foram considerados inconsistentes e tratados como nulos.

3.3.2. Colesterol Total

Foram encontradas 17 instâncias com valores de `tot_chole` superiores a 1000. Devido à alta improbabilidade desses valores serem reais, eles foram classificados como erros e substituídos por valores nulos.

3.3.3. Colesterol HDL

Valores de `hdl_chole` superiores a 200 foram removidos por serem considerados inconsistentes. No total, 42 registros foram afetados por essa correção.

3.4. Colesterol LDL

Foram removidos todos os valores de `ldl_chole` superiores a 1100, pois ultrapassavam significativamente os limites esperados. Ao todo, 25 instâncias foram afetadas.

3.5. Creatinina no Sangue

Valores de `serum_creatinine` acima de 41 foram considerados erros e removidos, totalizando 19 instâncias corrigidas.

3.6. Glutamato-Oxaloacetato Transaminase (AST)

Foram detectadas 21 instâncias com valores de `sgot_ast` superiores a 5000, um nível extremamente alto e inconsistente com a normalidade. Como resultado, esses valores foram removidos da análise.

Por fim, um total de 162 instâncias foram removidas do conjunto de dados, resultando em um total final de 991.345 instâncias. A remoção dessas observações inconsistentes foi essencial para garantir a qualidade e a confiabilidade da análise, minimizando a influência de valores extremos ou possivelmente errôneos nos resultados. Esse processo contribui diretamente para a melhoria do desempenho dos modelos preditivos e para a obtenção de conclusões mais precisas e representativas da população analisada.

4. Transformação

A etapa de Transformação consiste em modificar ou consolidar os dados para que estejam no formato adequado para a análise. Esse processo envolve converter os dados brutos em variáveis relevantes e com formatos padronizados, facilitando o uso de algoritmos de aprendizado de máquina e técnicas de inteligência artificial. A transformação pode incluir normalização, agregação, e criação de novas variáveis que representam combinações ou decomposições de dados existentes. Essa etapa é essencial para aprimorar a qualidade dos dados e a precisão dos modelos de IA, tornando-os mais eficientes na extração de padrões relevantes.

4.1. Valores Descritivos

No contexto de aprendizado de máquina e descoberta de conhecimento, a conversão de valores descritivos para numéricos é uma etapa essencial do pré-processamento dos dados.

Muitos algoritmos de modelagem estatística e inteligência artificial não operam diretamente com dados categóricos, exigindo que esses valores sejam transformados em representações numéricas. Além de garantir compatibilidade com os modelos, essa conversão também melhora a eficiência computacional e facilita a análise matemática das relações entre os atributos. Dessa forma, categorias textuais são substituídas por valores numéricos que mantêm a mesma informação sem introduzir viés indevido ao modelo. Essa abordagem permite que os algoritmos processem os dados corretamente, evitando problemas como dificuldade na comparação e inconsistências no tratamento dos atributos.

4.1.1. Sexo

O atributo `sex` possuía duas categorias: `male` e `female`. Para permitir sua utilização nos modelos de análise, esses valores foram convertidos para representações numéricas, sendo `male` substituído por 0 e `female` por 1. Essa transformação preserva a informação original e facilita sua interpretação nos algoritmos.

4.1.2. Beber

O atributo `drk_yn`, responsável por indicar se a pessoa consome bebidas alcoólicas ou não, originalmente possuía os valores `n` e `y`. Para tornar esse dado compatível com os modelos de aprendizado de máquina, os valores foram convertidos para 0 e 1, respectivamente, representando a resposta negativa e positiva. Essa transformação simplifica o processamento e mantém a integridade da informação sem alterar seu significado.

4.2. Agrupamento de Valores com Referências Clínicas

Para determinadas variáveis contínuas, a simples utilização de valores numéricos brutos pode dificultar a análise e interpretação dos dados. Assim, optou-se por agrupar esses atributos em categorias baseadas em valores de referência amplamente aceitos na literatura médica. Esse agrupamento permite a redução da complexidade dos dados, facilita a identificação de padrões relevantes e melhora a coerência das análises. Além disso, a categorização minimiza a influência de valores extremos, tornando os modelos mais estáveis e robustos. Como parte desse processo, os atributos originais foram substituídos por suas versões categorizadas, garantindo uma estrutura mais adequada para a modelagem e interpretação dos resultados.

4.2.1. Pressão Arterial Sistólica e Diastólica

Os valores da pressão arterial sistólica (`sbp`) e diastólica (`dbp`) foram transformados em categorias baseadas em faixas de referência clínicas. A pressão arterial sistólica foi classificada nos seguintes grupos: hipotensão (`sbp` $<$ 90 mmHg), normal (90-120 mmHg), limítrofe (121-129 mmHg), hipertensão estágio 1 (130-139 mmHg), hipertensão estágio 2 (140-179 mmHg) e crise hipertensiva (`sbp` $\geq$ 180 mmHg). Da mesma forma, a pressão arterial diastólica foi categorizada em: hipotensão (`dbp` $<$ 60 mmHg), normal (60-80 mmHg), limítrofe (81-89 mmHg), hipertensão estágio 1 (90-99 mmHg), hipertensão

estágio 2 (100-119 mmHg) e crise hipertensiva (dbp $\geq$ 120 mmHg). Após essa transformação, os atributos sbp e dbp foram removidos e substituídos pelos novos atributos categorizados sbp_grupo e dbp_grupo, respectivamente.

4.2.2. Glicose no Sangue em Jejum

O atributo blds, que representa a glicose no sangue em jejum, foi categorizado em três grupos principais: normoglicemia (blds $<$ 100 mg/dL), pré-diabetes (100-125 mg/dL) e diabetes (blds $\geq$ 126 mg/dL). Essa categorização permite uma análise mais estruturada dos perfis glicêmicos, eliminando pequenas variações que poderiam impactar negativamente a modelagem preditiva. O atributo original foi removido e substituído por blds_grupo.

4.2.3. Hemoglobina

Os valores de hemoglobina (hemoglobin) foram classificados em três categorias com base em referências clínicas: baixa hemoglobina (hemoglobin $<$ 13.5 g/dL), normal (13.5-17.5 g/dL) e elevada (hemoglobin $>$ 17.5 g/dL). Essa categorização foi ajustada levando em conta variações fisiológicas esperadas, tornando a análise mais robusta. O atributo original foi removido e substituído pela variável categorizada hemoglobin_grupo.

Após essas transformações, os atributos numéricos originais foram descartados, sendo substituídos por suas versões agrupadas. Essa abordagem minimiza o impacto de valores extremos, melhora a interpretabilidade dos dados e mantém a relevância clínica das informações, contribuindo para a construção de modelos mais estáveis e representativos.

4.3. Agrupamento com K-Means

Para alguns atributos, não havia valores de referência clínicos que permitissem um agrupamento direto. Nesses casos, foi utilizado o método do cotovelo para determinar a quantidade ideal de clusters e, posteriormente, o algoritmo K-Means para segmentação dos dados. Essa abordagem possibilita a identificação de padrões latentes nos atributos contínuos, organizando-os em grupos distintos que refletem melhor a distribuição dos dados.

A análise indicou que o número ideal de clusters variava entre **4 e 6**, conforme identificado pelo ponto de inflexão no gráfico do método do cotovelo. Dessa forma, os seguintes atributos passaram por essa transformação:

- waistline foi categorizado em 6 grupos, originando o atributo waistline_grupo.
- weight foi categorizado em 6 grupos, originando o atributo weight_grupo.
- serum_creatinine foi categorizado em 4 grupos, originando o atributo serum_creatinine_grupo.
- tot_chole foi categorizado em 6 grupos, originando o atributo tot_chole_grupo.
- hdl_chole foi categorizado em 6 grupos, originando o atributo hdl_chole_grupo.

- `ldl_chole` foi categorizado em 6 grupos, originando o atributo `ldl_chole_grupo`.
- `triglyceride` foi categorizado em 6 grupos, originando o atributo `triglyceride_grupo`.
- `sgot_ast` foi categorizado em 6 grupos, originando o atributo `sgot_ast_grupo`.
- `gamma_gtp` foi categorizado em 6 grupos, originando o atributo `gamma_gtp_grupo`.

Assim como na abordagem anterior, os atributos originais foram removidos e substituídos por suas versões agrupadas. Esse processo melhora a interpretação dos dados e reduz a influência de variações individuais, tornando os modelos mais estáveis e menos sensíveis a outliers. O uso do K-Means permitiu que os dados fossem organizados de maneira estruturada, mantendo a coerência na modelagem e garantindo uma melhor adaptação aos algoritmos de aprendizado de máquina.

4.4. Análise de Correlação e Similaridade

A análise de correlação é uma etapa essencial no pré-processamento de dados, especialmente em contextos de aprendizado de máquina e modelagem estatística. A correlação entre atributos indica o grau de relação linear existente entre eles, ajudando a identificar possíveis redundâncias ou relações fortes que podem impactar a performance dos modelos. Quando dois atributos apresentam alta correlação, um deles pode ser removido para reduzir a dimensionalidade dos dados, evitando informações redundantes e melhorando a eficiência computacional.

No presente estudo, a matriz de correlação foi utilizada para verificar a existência de atributos altamente similares que poderiam ser eliminados sem prejuízo à representatividade dos dados. Como critério de análise, buscou-se identificar pares de atributos com coeficiente de correlação próximo de **1.0** (correlação positiva perfeita) ou **-1.0** (correlação negativa perfeita), pois tais atributos carregariam informações praticamente idênticas ou simetricamente opostas.

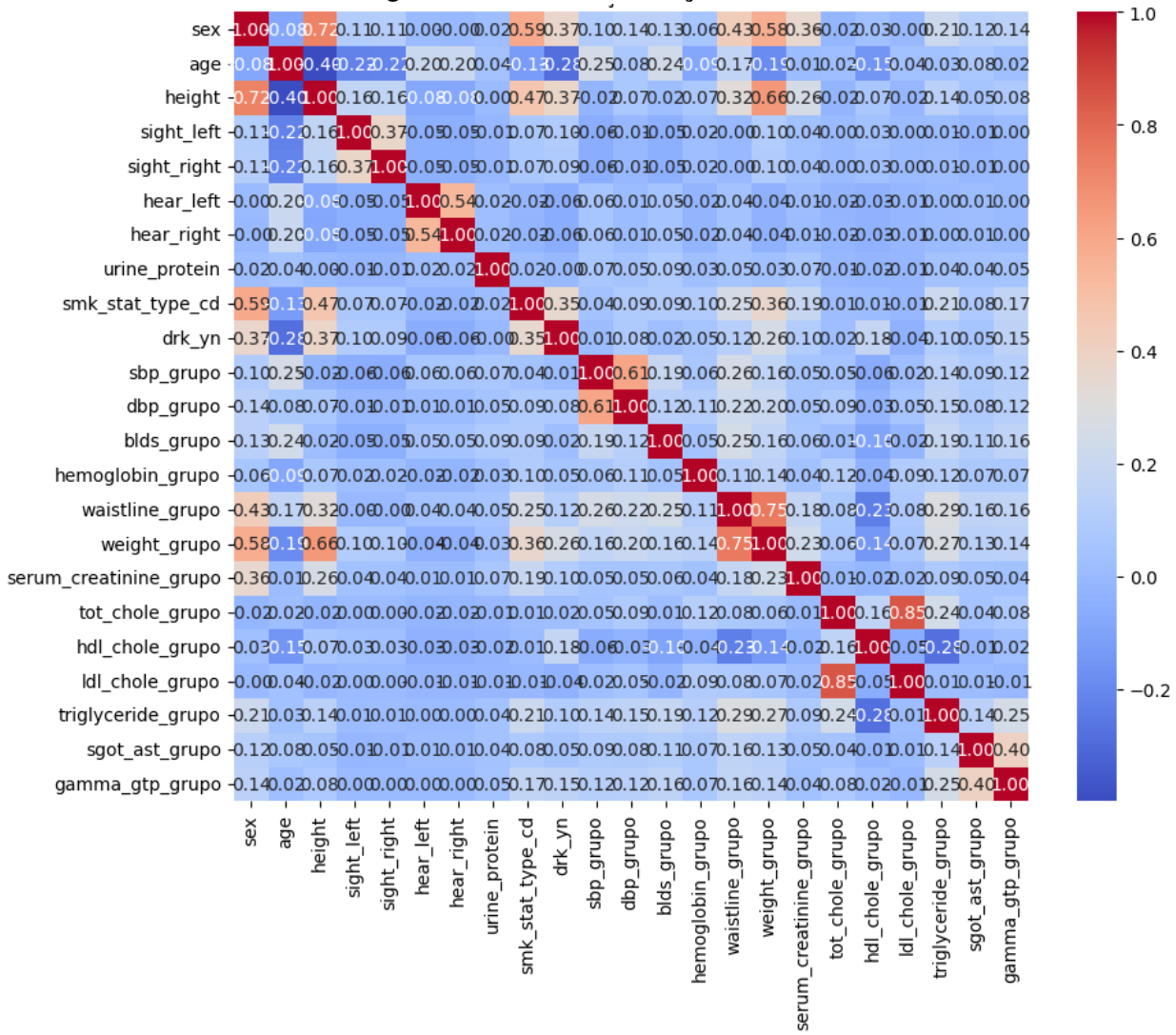
A análise dos coeficientes revelou que, embora alguns atributos apresentem correlações moderadas, não houve nenhuma relação forte o suficiente para justificar a remoção de atributos. Isso indica que cada variável contribui com informações distintas para a modelagem, evitando redundâncias significativas no conjunto de dados. Dessa forma, nenhum atributo foi excluído com base nessa análise, garantindo que todas as características continuem sendo utilizadas nos processos de aprendizado de máquina e descoberta de conhecimento.

Na Figura 1, é possível visualizar a matriz de correlação, onde as cores representam a relação entre os atributos: tons mais próximos do vermelho indicam correlações positivas mais fortes (valores próximos de 1), enquanto tons em azul intenso representam correlações negativas mais acentuadas (valores próximos de -1). Essa representação gráfica auxilia na identificação de padrões de similaridade entre as variáveis.

4.5. Ranking de Importância dos Atributos

A análise da importância dos atributos é uma etapa essencial no pré-processamento de dados para aprendizado de máquina. Através dessa abordagem, é possível identificar quais

Figura 1. Matriz de Correlação entre os Atributos

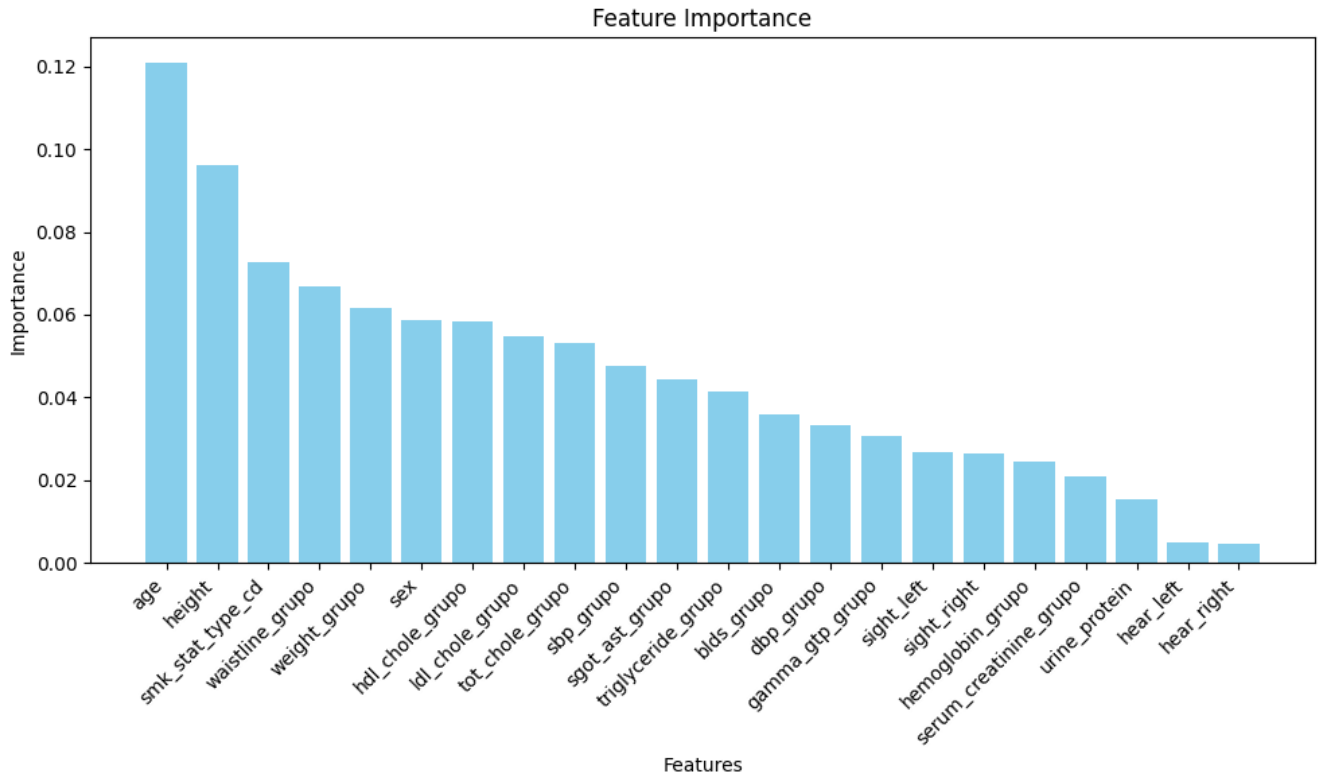


variáveis possuem maior impacto na classificação e quais têm pouca ou nenhuma relevância. Esse processo permite otimizar o desempenho dos modelos, reduzindo a complexidade computacional e eliminando atributos desnecessários que podem introduzir ruído ou redundância na análise.

Para determinar a importância dos atributos, foi utilizado o algoritmo **Random Forest**, que avalia a contribuição de cada variável na tomada de decisão do modelo. Como resultado dessa análise, observou-se que os atributos `hear_left` e `hear_right` apresentaram uma relevância muito baixa na classificação, conforme ilustrado na Tabela 1. Dessa forma, optou-se por remover essas variáveis para evitar a inclusão de características irrelevantes no modelo final. A figura 2 apresenta o gráfico com a importância de cada atributo, permitindo visualizar a relevância relativa de cada variável no modelo.

A análise demonstrou que variáveis como `age`, `height` e `smk_stat_type_cd` possuem maior influência na classificação. Já os atributos `hear_left` e `hear_right` apresentaram um impacto extremamente reduzido,

Figura 2. Ranking de Importância dos Atributos



reforçando a decisão de removê-los. Esse processo de seleção de atributos contribui para modelos mais eficientes e interpretáveis, garantindo que apenas as variáveis mais relevantes sejam utilizadas nas etapas posteriores de aprendizado de máquina.

4.6. Balanceamento dos Dados

O balanceamento dos dados é um aspecto crucial na construção de modelos de aprendizado de máquina, especialmente em problemas de classificação. Quando há um desbalanceamento significativo entre as classes, o modelo pode ter dificuldades em aprender corretamente os padrões dos dados minoritários, resultando em viés na predição e reduzindo a capacidade de generalização. Em casos extremos, um modelo treinado com dados desbalanceados pode simplesmente favorecer a classe majoritária, comprometendo a eficácia da análise.

No entanto, no presente estudo, verificou-se que as classes estavam naturalmente equilibradas. O conjunto de dados apresenta **495.759 instâncias** de indivíduos que não ingerem bebidas alcoólicas e **495.441 instâncias** de indivíduos que ingerem, o que representa uma distribuição bastante próxima. Dessa forma, não foi necessária a aplicação de técnicas de balanceamento, como oversampling ou undersampling, garantindo que os modelos de aprendizado de máquina possam aprender de forma justa e representativa sem necessidade de ajustes adicionais.

4.7. Normalização de Dados

A normalização de dados é uma etapa fundamental no pré-processamento para aprendizado de máquina, pois garante que os atributos numéricos sejam escalonados para uma

Tabela 1. Ranking de Importância dos Atributos

Posição	Atributo	Importância (%)
1	age	12.09
2	height	9.62
3	smk_stat_type_cd	7.28
4	waistline_grupo	6.69
5	weight_grupo	6.15
6	sex	5.88
7	hdl_chole_grupo	5.82
8	ldl_chole_grupo	5.48
9	tot_chole_grupo	5.32
10	sbp_grupo	4.77
11	sgot_ast_grupo	4.43
12	triglyceride_grupo	4.15
13	blds_grupo	3.60
14	dbp_grupo	3.31
15	gamma_gtp_grupo	3.05
16	sight_left	2.68
17	sight_right	2.64
18	hemoglobin_grupo	2.44
19	serum_creatinine_grupo	2.10
20	urine_protein	1.54
21	hear_left	0.48
22	hear_right	0.47

faixa específica, melhorando a estabilidade e eficiência dos algoritmos. Essa técnica é especialmente útil quando os dados possuem escalas diferentes, o que pode impactar negativamente o desempenho de modelos baseados em distância, como redes neurais, k-Nearest Neighbors (k-NN) e métodos baseados em gradiente.

No presente estudo, foi utilizada a técnica de **normalização Min-Max**, que transforma os valores dos atributos para um intervalo entre **0 e 1**. Esse método preserva a distribuição original dos dados, garantindo que os valores sejam reescalonados proporcionalmente sem alterar sua relação entre si. A normalização Min-Max é amplamente utilizada devido à sua simplicidade e eficácia, tornando o treinamento dos modelos mais estável e reduzindo possíveis viéses gerados por atributos com magnitudes muito diferentes.

Com essa abordagem, todas as variáveis numéricas foram transformadas para garantir que estivessem na mesma escala, facilitando a convergência dos algoritmos de aprendizado de máquina e evitando que atributos com valores absolutos mais altos dominassem a modelagem.

4.8. Remoção de Dados Duplicados

A presença de instâncias duplicadas em um conjunto de dados pode comprometer a qualidade da análise e afetar o desempenho dos modelos de aprendizado de máquina. Dados repetidos podem enviesar os resultados, tornando alguns padrões estatísticos super-

representados e reduzindo a capacidade do modelo de generalizar corretamente para novos exemplos. Além disso, a duplicação desnecessária pode aumentar o tempo de processamento e o consumo de memória, dificultando a eficiência computacional.

Para garantir a integridade dos dados, foi realizada uma verificação e remoção de instâncias duplicadas. Após esse processo, o conjunto de dados final passou a conter **49.148 instâncias** de indivíduos que ingerem bebidas alcoólicas e **48.878 instâncias** de indivíduos que não ingerem. Essa limpeza assegura que cada observação no dataset seja única, eliminando redundâncias que poderiam impactar negativamente a modelagem e a extração de conhecimento.

5. Mineração de Dados

A mineração de dados é um processo essencial na descoberta de padrões e relações ocultas em grandes volumes de informações. No contexto do aprendizado de máquina, essa etapa permite extrair conhecimento significativo dos dados, contribuindo para a construção de modelos preditivos eficientes. A análise dos dados brutos isoladamente muitas vezes não é suficiente para compreender a complexidade das relações existentes, tornando a aplicação de algoritmos de mineração fundamental para a obtenção de insights mais precisos e úteis.

Para garantir uma abordagem abrangente, a mineração de dados foi realizada em três diferentes cenários:

- Com dados sem normalização e sem remoção de duplicatas;
- Com dados normalizados e sem remoção de duplicatas;
- Com dados normalizados e com remoção de duplicatas.

Essa variação nos conjuntos de dados permitiu avaliar o impacto da normalização e da eliminação de registros redundantes no desempenho dos modelos preditivos. Além disso, foram utilizados diversos algoritmos de aprendizado supervisionado, abrangendo diferentes abordagens e técnicas de modelagem.

Os algoritmos empregados foram:

- **Árvore de Decisão CART**: Utilizando o classificador `DecisionTreeClassifier(random_state=42)`;
- **Support Vector Machine (SVM)**: Avaliando a versão padrão do algoritmo;
- **Support Vector Machine (SVM) com Kernel Diferenciado**: Testando variações no kernel para verificar impactos na performance;
- **K-Nearest Neighbors (KNN)**: Com experimentação de diferentes métricas de distância (`euclidean`, `manhattan`, `minkowski`) e variação no número de vizinhos (`k` variando de 1 a 30 e posteriormente expandido para `30 a 300`, além de uma análise refinada entre `100 a 140`);
- **Naive Bayes**: Testando tanto a versão *Gaussian Naive Bayes* quanto a *Categorical Naive Bayes*;
- **Regressão Linear**: Aplicando o método dos mínimos quadrados;
- **Perceptron**: Utilizado para classificação binária.

Para avaliar o desempenho de cada modelo, foram analisadas a **acurácia** e a **matriz de confusão**, permitindo uma compreensão detalhada da capacidade de classificação de cada algoritmo. Essa abordagem possibilitou comparar as diferentes técnicas e identificar quais apresentaram melhor desempenho na tarefa de prever o consumo de bebidas alcoólicas a partir das variáveis disponíveis no conjunto de dados.

Para viabilizar a análise e reduzir o custo computacional dos experimentos, foi utilizada uma amostra representativa correspondente a 10% do conjunto de dados original. Essa amostragem foi realizada de forma aleatória, garantindo que a distribuição das variáveis fosse preservada. Dessa maneira, foi possível avaliar o impacto das diferentes abordagens de pré-processamento e modelagem sem comprometer a representatividade dos dados, assegurando que os resultados obtidos possam ser generalizados para o conjunto completo.

5.1. KNN

O K-Nearest Neighbors (KNN) é um algoritmo de aprendizado supervisionado amplamente utilizado em tarefas de classificação e regressão. Ele funciona baseado na ideia de proximidade, classificando uma nova amostra de acordo com as classes majoritárias de seus k vizinhos mais próximos no espaço de características. A definição de "proximidade"é geralmente feita utilizando métricas como a distância Euclidiana, Manhattan ou Minkowski.

5.1.1. Implementação KNN Sem Normalizar e Sem Remoção de Duplicatas

A Figura 3 apresenta a matriz de confusão gerada para o modelo KNN, configurado com o melhor valor de k identificado na análise. A matriz ilustra a distribuição dos acertos e erros na classificação, possibilitando uma avaliação detalhada do desempenho do modelo. A acurácia obtida foi de **0,63** com o k = 28 e a distância Manhattan, indicando a capacidade preditiva do classificador dentro do conjunto de teste.

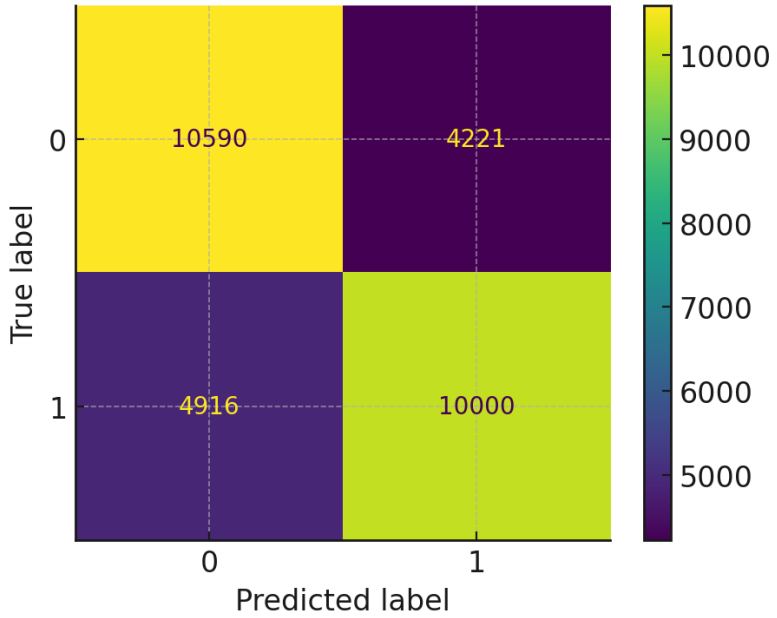


Figura 3. Matriz de Confusão KNN Sem Normalização e Sem Remoção de Duplicatas

Além disso, a Tabela3 exibe os **dez melhores resultados** para diferentes valores de k e distâncias utilizadas na métrica do KNN. Essa análise comparativa permitiu

identificar a configuração mais eficiente para o modelo, auxiliando na escolha do melhor hiperparâmetro para a classificação.

Tabela 2. Resultado do KNN Sem Normalizar e Com Duplicatas

K	Distância	Acurácia (%)
28	manhattan	69,27
25	manhattan	69,20
22	manhattan	69,03
19	manhattan	68,85
25	minkowski	68,52
25	euclidean	68,52
28	minkowski	68,43
28	euclidean	68,43
16	manhattan	68,35
19	minkowski	68,30

5.1.2. Implementação KNN Com Normalização e Sem Remoção de Duplicatas

Nesta abordagem, os dados foram submetidos a um processo de normalização antes da aplicação do modelo KNN. A normalização é uma técnica essencial para algoritmos baseados em distância, como o KNN, pois garante que todas as variáveis tenham a mesma escala, evitando que atributos com valores numéricos maiores dominem a influência sobre a métrica de similaridade.

A Tabela 3 apresenta os **dez melhores resultados** obtidos após a normalização, permitindo uma comparação direta com a versão sem normalização. A matriz de confusão correspondente é exibida na Figura 4, onde é possível visualizar a distribuição dos acertos e erros na classificação.

Tabela 3. Resultado do KNN Normalizado

K	Distância	Acurácia (%)
28	manhattan	69,27
25	manhattan	69,20
22	manhattan	69,03
19	manhattan	68,85
25	minkowski	68,52
25	euclidean	68,52
28	minkowski	68,43
28	euclidean	68,43
16	manhattan	68,35
19	minkowski	68,30

Embora a normalização dos dados geralmente melhore o desempenho do KNN, **neste caso específico, os resultados permaneceram praticamente inalterados**. A acurácia máxima obtida ainda foi de **69,27%** com $k = 28$ e distância Manhattan, **idêntica ao cenário sem normalização**. Esse comportamento pode ser explicado pelo fato de que os

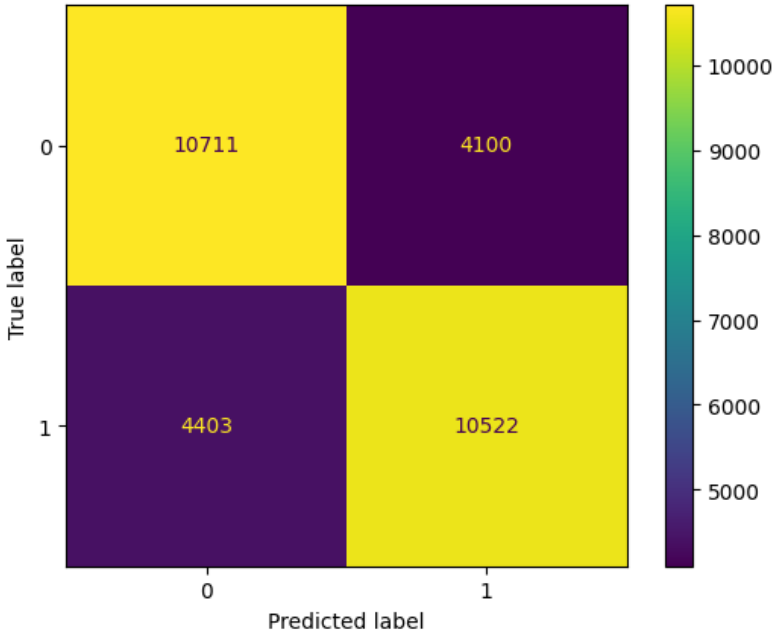


Figura 4. Matriz de Confusão KNN Com Normalização e Sem Remoção de Duplicatas

dados já possuíam uma distribuição numérica favorável ao modelo, ou que as relações entre os atributos e a variável alvo não foram significativamente impactadas pela escala dos dados. Assim, a normalização, apesar de ser uma boa prática geral, não trouxe melhorias perceptíveis para este conjunto de dados.

5.1.3. Implementação KNN Com Normalização e Com Remoção de Duplicatas

Nesta abordagem, além da normalização dos dados, foi realizada a remoção de instâncias duplicadas. A eliminação de registros redundantes é uma prática essencial no pré-processamento de dados, pois reduz a redundância informacional e pode melhorar a generalização do modelo, evitando que padrões artificiais influenciem a aprendizagem.

A Tabela 4 apresenta os **dez melhores resultados** obtidos após a aplicação da normalização e da remoção de duplicatas. Comparando esses valores com os obtidos na abordagem anterior (com normalização, mas sem remoção de duplicatas), nota-se que a acurácia máxima aumentou de **69,27%** para **71,26%**, representando uma **pequena, porém relevante melhoria no desempenho do modelo**.

Esse resultado indica que a presença de instâncias duplicadas pode ter introduzido viés nos dados, afetando a capacidade do KNN de generalizar corretamente. Com a remoção das duplicatas, o modelo foi capaz de aprender de maneira mais representativa, refletindo um melhor desempenho na classificação. A matriz de confusão correspondente é exibida na Figura 5, evidenciando a distribuição dos acertos e erros após essa etapa de refinamento dos dados.

Tabela 4. Resultado do KNN Normalizado Sem Duplicatas

K	Distância	Acurácia (%)
28	minkowski	71.26
28	euclidean	71.26
28	manhattan	71.15
25	minkowski	71.09
25	euclidean	71.09
22	manhattan	71.00
25	manhattan	71.00
19	minkowski	70.94
19	euclidean	70.94
22	minkowski	70.93

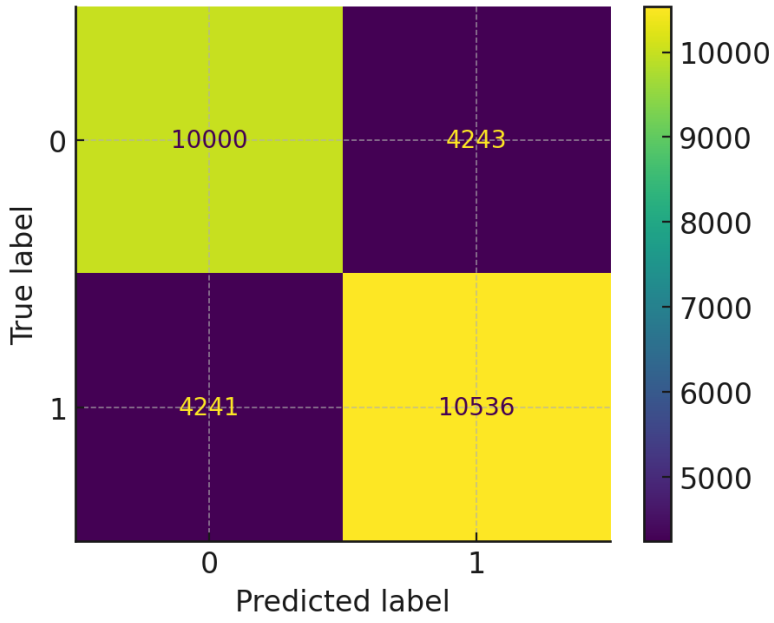


Figura 5. Matriz de Confusão KNN Com Normalização e Com Remoção de Duplicatas

5.2. Árvores de Decisão

A árvore de decisão é um modelo de aprendizado de máquina utilizado para tarefas de classificação e regressão, que funciona de forma intuitiva e visualmente interpretável. Ela segmenta os dados em diferentes "nós", criando uma estrutura hierárquica em forma de árvore. Cada nó de decisão representa uma pergunta ou condição baseada nos atributos dos dados, e os ramos correspondem aos possíveis resultados dessas condições. Esse processo é repetido até que a árvore atinja um nível de profundidade pré-determinado ou até que todos os dados sejam classificados corretamente nos nós finais (folhas). A principal vantagem das árvores de decisão é a simplicidade e interpretabilidade, mas elas podem se tornar suscetíveis a overfitting, especialmente em árvores muito profundas. Para contornar essa limitação, técnicas como poda ou o uso de ensembles, como Random Forests, são frequentemente aplicadas.

5.2.1. Implementação Árvore de Decisão Sem Normalização e Sem Remoção de Duplicatas

A Figura 6 apresenta a matriz de confusão para o modelo de **Árvore de Decisão**, aplicada sem a normalização dos dados e sem a remoção de instâncias duplicadas. A matriz permite visualizar a distribuição dos acertos e erros na classificação, auxiliando na interpretação do desempenho do modelo.

A acurácia obtida foi de **63,32%**, indicando um desempenho consistente na separação das classes. Como as Árvores de Decisão funcionam por meio de divisões sequenciais nos dados com base em critérios de impureza, essa taxa de acerto reflete a capacidade do modelo de identificar padrões nos atributos disponíveis e tomar decisões baseadas nas regras inferidas a partir do conjunto de treinamento.

Vale ressaltar que a Árvore de Decisão possui a vantagem de ser um modelo interpretável, permitindo a análise das decisões tomadas em cada nível da estrutura hierárquica gerada. Essa característica torna o modelo útil para a extração de conhecimento e compreensão dos fatores mais relevantes na distinção entre as classes.

5.2.2. Implementação Árvore de Decisão Com Normalização e Sem Remoção de Duplicatas

A Figura 7 apresenta a matriz de confusão do modelo de **Árvore de Decisão**, treinado com os dados normalizados, porém sem a remoção de instâncias duplicadas. A matriz permite visualizar a quantidade de acertos e erros cometidos pelo modelo durante a classificação.

A acurácia obtida foi de **63,00%**, o que demonstra um desempenho estável na separação das classes. A normalização dos dados não impactou significativamente os resultados, o que pode indicar que o modelo de Árvore de Decisão não é fortemente influenciado pela escala dos atributos, já que seu funcionamento se baseia em regras de decisão e divisões nos dados em vez de cálculos de distância.

Além disso, a Árvore de Decisão continua apresentando a vantagem de interpretabilidade, permitindo a análise das regras geradas e identificando quais atributos possuem

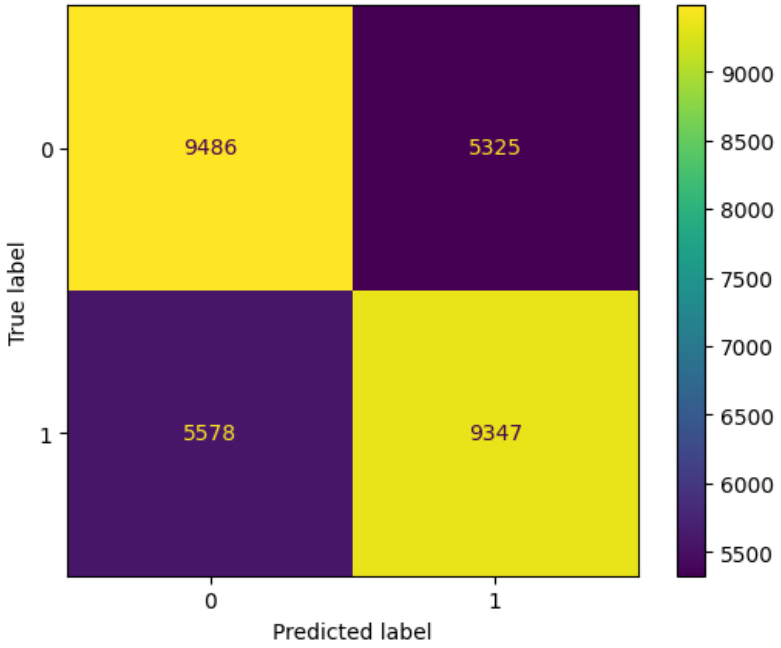


Figura 6. Matriz de Confusão Árvore de Decisão

maior importância na decisão final. Essa característica reforça a utilidade do modelo em aplicações que exigem transparência na tomada de decisões.

5.2.3. Implementação Árvore de Decisão Com Normalização e Com Remoção de Duplicatas

A Figura 8 apresenta a matriz de confusão do modelo de **Árvore de Decisão**, treinado com os dados normalizados e após a remoção de instâncias duplicadas. A matriz ilustra a quantidade de acertos e erros cometidos pelo modelo na classificação.

A acurácia obtida foi de **62,00%**, representando uma leve redução no desempenho em relação às abordagens anteriores. A remoção de duplicatas geralmente contribui para reduzir viéses no conjunto de dados, proporcionando um modelo mais generalizável. No entanto, neste caso específico, a remoção das instâncias redundantes pode ter eliminado informações que estavam auxiliando a tomada de decisão da árvore, resultando em um leve impacto na performance.

Ainda assim, a **interpretabilidade da Árvore de Decisão permanece um fator positivo**, permitindo a análise das divisões geradas e a identificação dos atributos mais relevantes no processo de classificação. Essa característica reforça a utilidade do modelo em cenários que exigem explicabilidade na tomada de decisões.

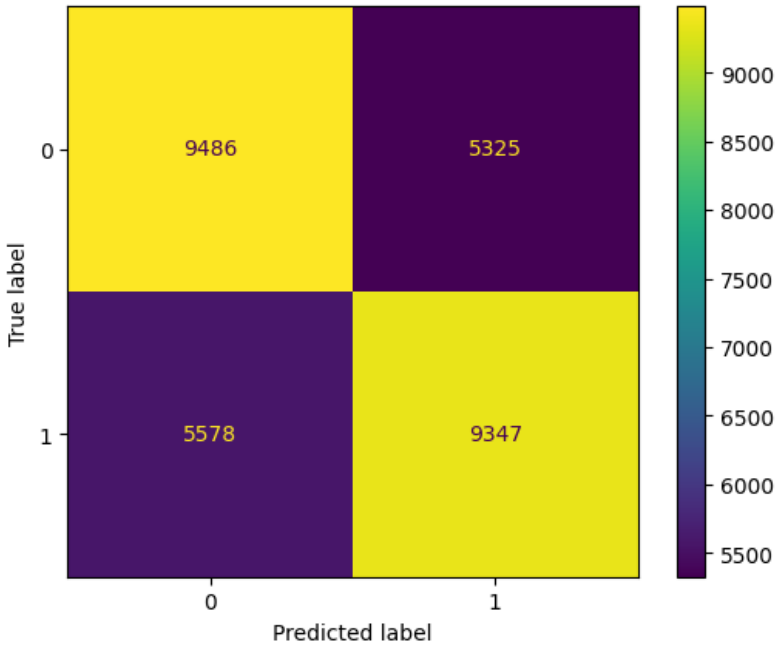


Figura 7. Matriz de Confusão Árvore de Decisão Normalizado

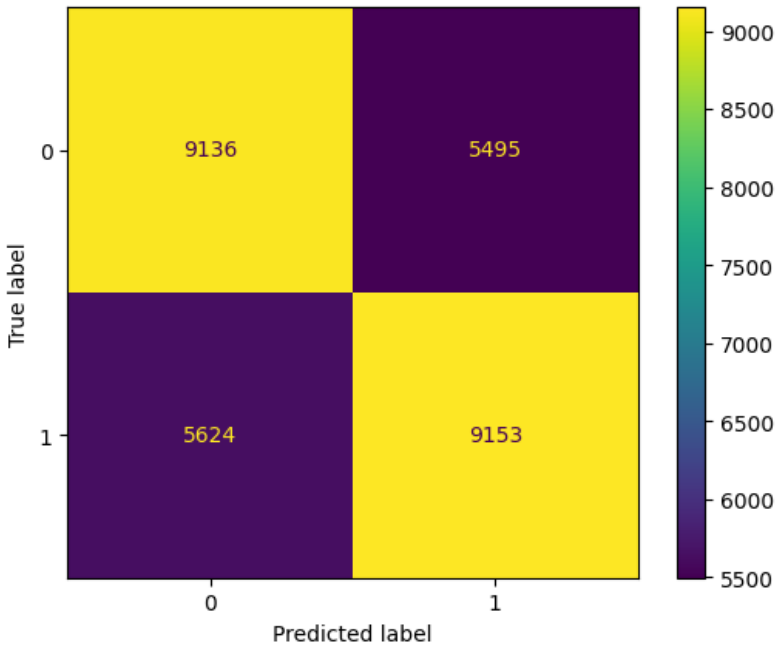


Figura 8. Matriz de Confusão Árvore de Decisão Normalizado Sem Duplicatas

5.3. SVM

O SVM (Support Vector Machine) é um modelo de aprendizado de máquina usado principalmente para classificação e regressão, conhecido por sua capacidade de lidar com problemas complexos de separação de classes. Ele funciona identificando um hiperplano que separa as classes no espaço dos atributos de forma ótima, maximizando a margem entre as instâncias de classes diferentes mais próximas desse hiperplano, chamadas de "vetores de suporte". Essa margem máxima ajuda a aumentar a generalização do modelo em novos dados. O SVM também pode utilizar kernels, que são funções que mapeiam os dados para espaços de maior dimensionalidade, permitindo a separação de classes de forma não linear em casos onde as classes não são linearmente separáveis. É um modelo robusto e eficaz, especialmente em problemas com alta dimensionalidade, mas pode ser computacionalmente intenso em grandes conjuntos de dados.

5.3.1. Implementação SVM Sem Normalização e Sem Remoção de Duplicatas

A Figura 9 apresenta a matriz de confusão do modelo de **Máquina de Vetores de Suporte (SVM)**, treinado com os dados em sua forma original, sem normalização e sem remoção de duplicatas. A matriz ilustra a distribuição dos acertos e erros, permitindo uma análise detalhada do desempenho do modelo.

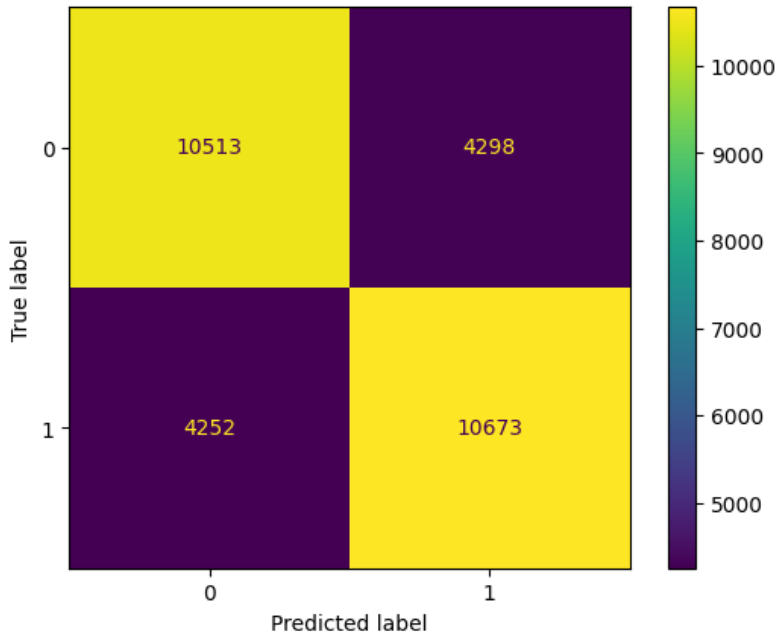


Figura 9. Matriz de Confusão SVM Sem Normalização e Sem Duplicatas

A acurácia obtida foi de **71,00%**, indicando um desempenho satisfatório na classificação. O algoritmo SVM é amplamente utilizado para problemas de classificação devido à sua capacidade de encontrar hiperplanos que maximizam a separação entre as classes. Mesmo sem a normalização dos dados, o modelo conseguiu atingir um resultado expressivo, o que sugere que a escala dos atributos não teve um impacto significativo na performance da classificação.

Entretanto, como o SVM se baseia em cálculos de distância, a normalização dos dados geralmente é recomendada para evitar que atributos com valores numéricos maiores tenham maior influência na decisão do modelo. Dessa forma, nas próximas etapas, será avaliado o impacto da normalização e da remoção de duplicatas no desempenho da classificação.

5.3.2. Implementação SVM Com Normalização e Sem Remoção de Duplicatas

A Figura 10 apresenta a matriz de confusão do modelo de **Máquina de Vetores de Suporte (SVM)**, treinado com os dados normalizados, porém sem a remoção de instâncias duplicadas. A matriz ilustra a distribuição dos acertos e erros, permitindo uma análise detalhada da classificação.

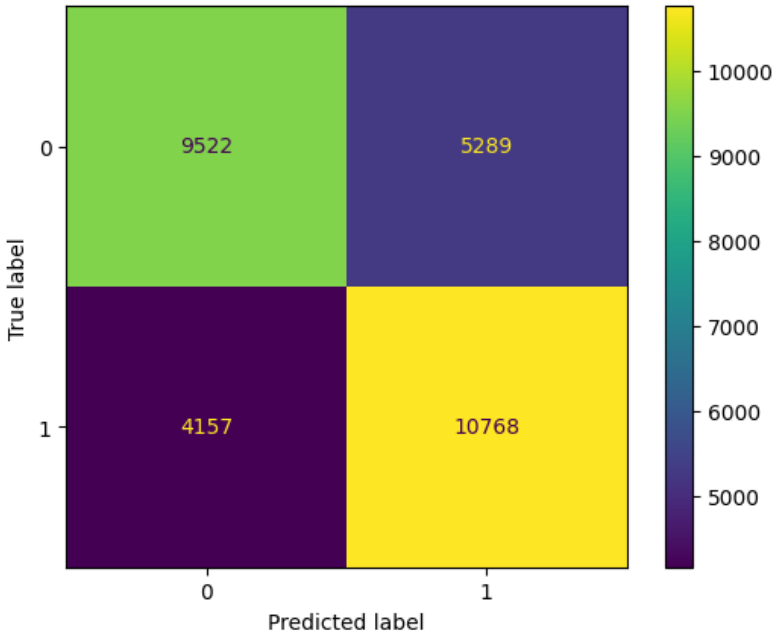


Figura 10. Matriz de Confusão SVM Normalizado Com Duplicatas

A acurácia obtida foi de **68,00%**, representando uma leve queda em relação à abordagem anterior sem normalização. Embora o SVM seja um algoritmo sensível à escala dos atributos, a normalização, que geralmente melhora seu desempenho, não resultou em um ganho significativo neste caso específico. Isso pode indicar que a distribuição original dos dados já favorecia a separação das classes ou que a normalização introduziu pequenas variações que afetaram o modelo.

Ainda assim, a normalização continua sendo uma etapa importante no pré-processamento, especialmente para modelos que dependem de medidas de distância, como o próprio SVM. Nas próximas etapas, será analisado o impacto da remoção de duplicatas no desempenho do modelo, possibilitando uma melhor compreensão do comportamento da classificação.

5.3.3. Implementação SVM Com Normalização e Com Remoção de Duplicatas

A Figura 11 apresenta a matriz de confusão do modelo de **Máquina de Vetores de Suporte (SVM)**, treinado com os dados normalizados e após a remoção de instâncias duplicadas. A matriz demonstra a distribuição dos acertos e erros, permitindo uma análise detalhada do desempenho do modelo.

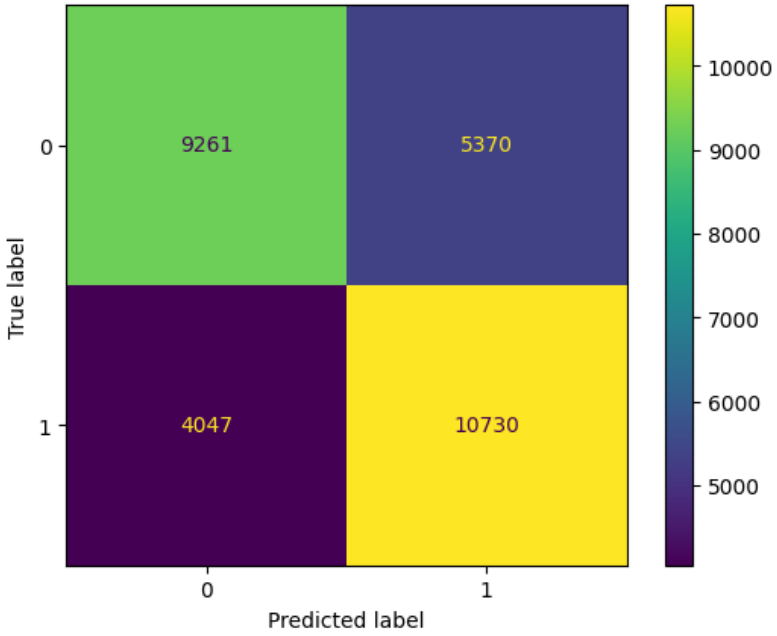


Figura 11. Matriz de Confusão SVM Normalizado e Sem Duplicatas

A acurácia obtida foi de **68,00%**, mantendo-se estável em relação à abordagem anterior, na qual os dados foram normalizados, mas as duplicatas não foram removidas. A remoção de instâncias duplicadas geralmente contribui para reduzir viéses no conjunto de dados, tornando o modelo mais generalizável. No entanto, neste caso específico, a eliminação de registros repetidos não trouxe um impacto significativo no desempenho da classificação.

Isso sugere que a presença de duplicatas não estava influenciando negativamente o aprendizado do modelo, possivelmente porque as informações repetidas refletiam corretamente padrões existentes nos dados. Dessa forma, a normalização continua sendo um fator relevante no pré-processamento para modelos baseados em distância, enquanto a remoção de duplicatas teve um efeito neutro no desempenho do SVM.

5.4. Naive Bayes

O algoritmo **Naive Bayes** é um classificador probabilístico baseado no **Teorema de Bayes**, que calcula a probabilidade de uma instância pertencer a uma determinada classe com base nas características dos dados. Ele assume que todos os atributos são **independentes entre si**, uma hipótese conhecida como **ingenuidade (naive)**, o que simplifica os cálculos probabilísticos e torna o algoritmo extremamente eficiente, mesmo para grandes volumes de dados. Essa abordagem permite que o modelo tenha um bom desempenho mesmo em cenários onde a suposição de independência não é totalmente válida.

Existem diferentes variações do Naive Bayes, adaptadas para diferentes tipos de dados. O **Gaussian Naive Bayes** é utilizado para atributos numéricos contínuos, assumindo que os dados seguem uma distribuição Gaussiana. Já o **Categorical Naive Bayes** é mais adequado para atributos categóricos, modelando as probabilidades de cada classe diretamente com base nas frequências observadas. Devido à sua simplicidade e eficiência computacional, o Naive Bayes é amplamente utilizado em aplicações como **filtragem de spam, análise de sentimentos e classificação de textos**, sendo uma escolha robusta para tarefas de classificação onde os dados possuem uma estrutura probabilística bem definida.

5.4.1. Implementação Sem Normalização e Sem Remoção de Duplicatas - Naive Bayes Qualitativo

A Figura 12 apresenta a matriz de confusão do modelo de **Naive Bayes (qualitativo)** aplicado aos dados sem normalização e sem remoção de instâncias duplicadas. A matriz permite observar a distribuição dos acertos e erros, proporcionando uma visão clara do desempenho do modelo.

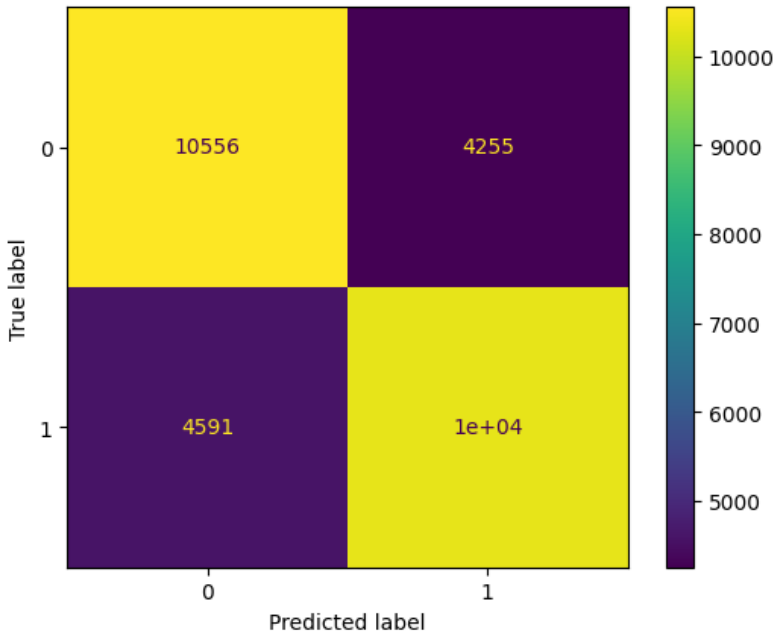


Figura 12. Matriz de Confusão Naive Bayes Qualitativo Sem Normalização e Sem Remoção de Duplicatas

A acurácia obtida foi de **70,25%** demonstrando um desempenho competitivo para um modelo probabilístico. O **Naive Bayes** é um classificador baseado no **Teorema de Bayes** assumindo independência condicional entre os atributos. Essa característica o torna eficiente em grandes conjuntos de dados e em cenários onde os atributos são representados por variáveis categóricas.

Mesmo sem normalização ou remoção de duplicatas, o modelo conseguiu um bom nível de acerto, o que pode indicar que os atributos qualitativos fornecem informações relevantes para a classificação. A seguir, será analisado o impacto da normalização e da remoção de duplicatas na performance do modelo.

5.4.2. Implementação Sem Normalização e Sem Remoção de Duplicatas - Naive Bayes Quantitativo

A Figura 13 apresenta a matriz de confusão do modelo de **Naive Bayes (quantitativo)** aplicado aos dados sem normalização e sem remoção de instâncias duplicadas. A matriz permite observar a distribuição dos acertos e erros, fornecendo uma análise detalhada da performance do modelo.

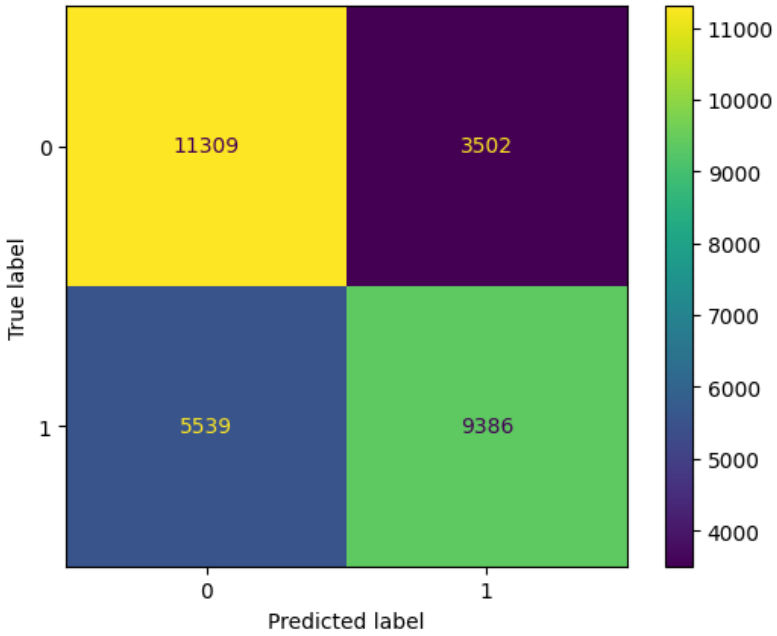


Figura 13. Matriz de Confusão Naive Bayes Quantitativo Sem Normalização e Sem Remoção de Duplicatas

A acurácia obtida foi de **69,60%** demonstrando um desempenho sólido para um modelo probabilístico baseado no **Teorema de Bayes**. Diferente do Naive Bayes qualitativo, que lida melhor com atributos categóricos, o modelo quantitativo assume que os atributos seguem uma distribuição estatística específica, como a distribuição normal (Gaussiana). Esse método é eficiente para modelar relações probabilísticas entre os atributos numéricos e a variável alvo.

Apesar de não ter sido aplicada normalização, o modelo conseguiu um bom ní-

vel de acerto, indicando que os dados numéricos não apresentaram grandes discrepâncias que prejudicassem a suposição de normalidade do Naive Bayes. As próximas análises avaliarão o impacto da normalização e da remoção de duplicatas no desempenho da classificação.

5.4.3. Implementação Com Normalização e Sem Remoção de Duplicatas - Naive Bayes Qualitativo

A Figura 14 apresenta a matriz de confusão do modelo de **Naive Bayes (qualitativo)** treinado com os dados normalizados, mas sem a remoção de instâncias duplicadas. A matriz evidencia a distribuição dos acertos e erros, proporcionando uma visão detalhada da performance do modelo.

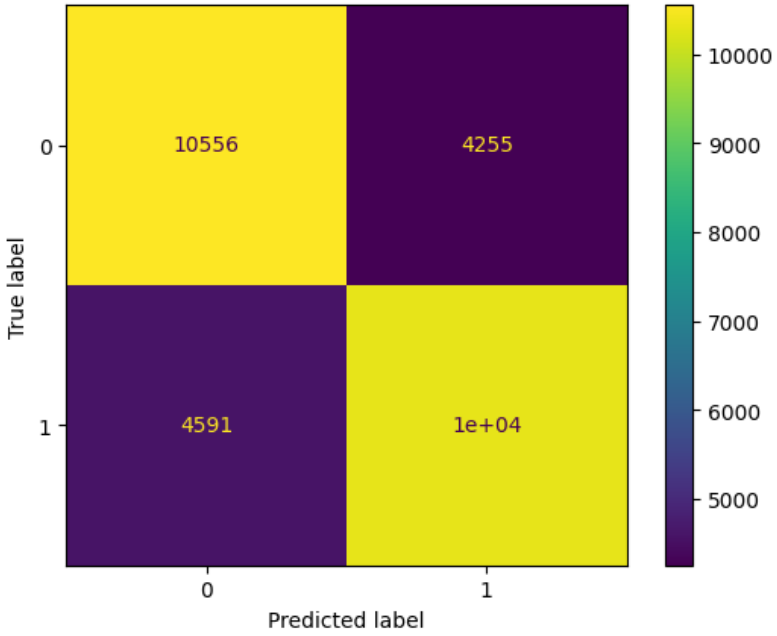


Figura 14. Matriz de Confusão Naive Bayes Qualitativo Normalizado e Sem Remoção de Duplicatas

A acurácia obtida foi de **70,25%** mantendo-se estável em relação ao modelo treinado sem normalização. Como o Naive Bayes qualitativo lida com atributos categóricos, a normalização não afeta diretamente seu desempenho, pois o modelo baseia-se em distribuições de probabilidade para cada classe em relação aos atributos.

Esse resultado reforça a robustez do Naive Bayes qualitativo para dados categóricos, demonstrando que a normalização pode não ser um fator determinante nesse tipo de abordagem. No entanto, em experimentos futuros, será analisado se a remoção de duplicatas pode influenciar positivamente o desempenho do modelo.

5.4.4. Implementação Com Normalização e Sem Remoção de Duplicatas - Naive Bayes Quantitativo

A Figura 15 apresenta a matriz de confusão do modelo de **Naive Bayes (quantitativo)** treinado com os dados normalizados, mas sem a remoção de instâncias duplicadas. A matriz evidencia a distribuição dos acertos e erros, proporcionando uma análise detalhada da classificação.

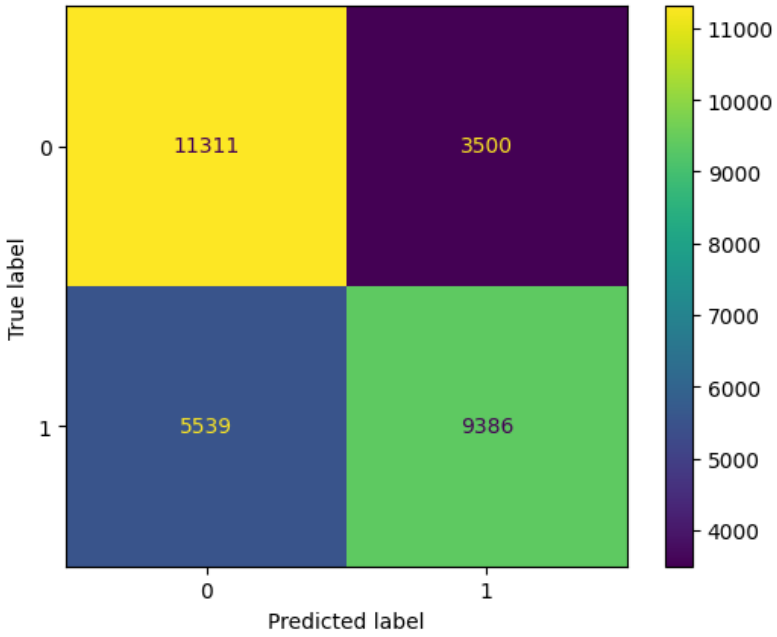


Figura 15. Matriz de Confusão Naive Bayes Quantitativo Normalizado e Sem Remoção de Duplicatas

A acurácia obtida foi de **69,60%** mantendo-se praticamente inalterada em relação ao modelo treinado sem normalização. O Naive Bayes quantitativo assume que os atributos seguem uma distribuição Gaussiana, e a normalização pode, em alguns casos, melhorar a suposição de normalidade dos dados. No entanto, neste caso específico, a normalização não teve um impacto significativo no desempenho da classificação.

Esse resultado sugere que a distribuição original dos atributos numéricos já estava dentro de um intervalo adequado para o modelo probabilístico, não necessitando de ajustes adicionais. Nas próximas etapas, será analisado o impacto da remoção de duplicatas no desempenho do modelo.

5.4.5. Implementação Com Normalização e Remoção de Duplicatas - Naive Bayes Qualitativo

A Figura 16 apresenta a matriz de confusão do modelo de **Naive Bayes (qualitativo)** treinado com os dados normalizados e após a remoção de instâncias duplicadas. A matriz evidencia a distribuição dos acertos e erros, permitindo uma avaliação mais detalhada do desempenho da classificação.

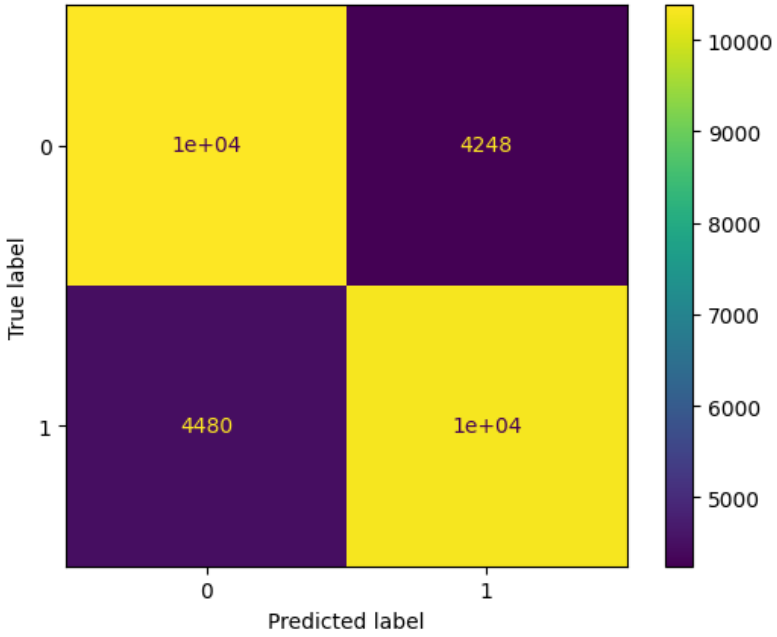


Figura 16. Matriz de Confusão Naive Bayes Qualitativo Normalizado e Com Remoção de Duplicatas

A acurácia obtida foi de **69,94%** apresentando uma leve queda em relação ao modelo anterior, que utilizava dados normalizados sem a remoção de duplicatas. O Naive Bayes qualitativo, que se baseia em probabilidades condicionais, pode ser impactado por alterações na distribuição dos dados quando instâncias repetidas são removidas, alterando ligeiramente as estimativas das probabilidades.

Esse resultado sugere que a remoção de duplicatas não trouxe um benefício significativo para o modelo, podendo indicar que os registros duplicados continham informações representativas da distribuição real dos dados. Assim, para o Naive Bayes qualitativo, a manutenção de duplicatas pode ser benéfica dependendo da estrutura dos dados. Ainda assim, a remoção de instâncias redundantes pode ter outros benefícios, como redução do tempo de processamento e melhoria na generalização do modelo em determinados cenários.

5.4.6. Implementação Com Normalização e Remoção de Duplicatas - Naive Bayes Quantitativo

A Figura 17 apresenta a matriz de confusão do modelo de **Naive Bayes (quantitativo)** treinado com os dados normalizados e após a remoção de instâncias duplicadas. A matriz ilustra a distribuição dos acertos e erros, possibilitando uma análise mais detalhada da classificação.

A acurácia obtida foi de **70,32%** representando um leve aumento em relação aos modelos anteriores. Esse resultado indica que a remoção de duplicatas teve um impacto positivo na performance do classificador, possivelmente reduzindo redundâncias e tornando o modelo mais generalizável.

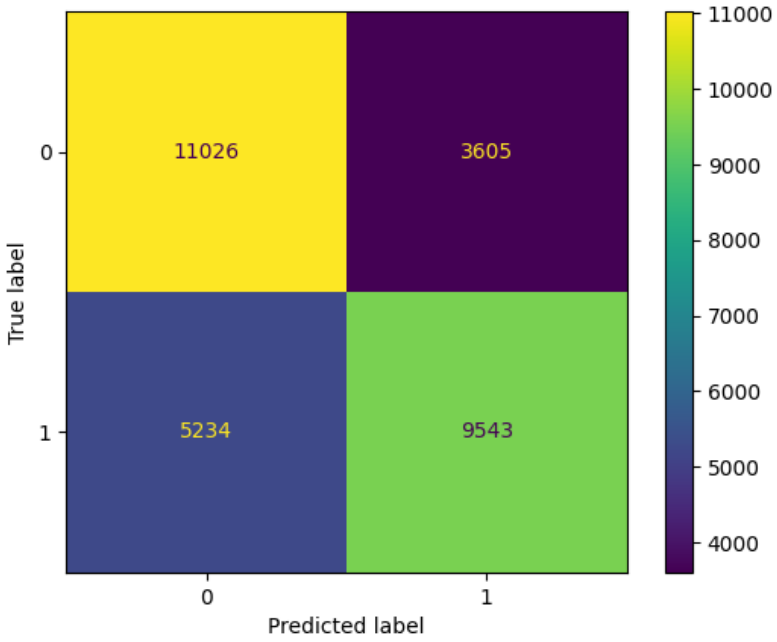


Figura 17. Matriz de Confusão Naive Bayes Quantitativo Normalizado e Com Remoção de Duplicatas

O Naive Bayes quantitativo assume que os atributos seguem uma distribuição Gaussiana, e a normalização pode contribuir para que os dados se aproximem dessa suposição, o que pode justificar a melhoria observada. Além disso, a remoção de instâncias duplicadas pode ter reduzido o viés presente no conjunto de dados, permitindo que o modelo aprenda padrões de forma mais eficiente.

Portanto, observa-se que a combinação da normalização com a remoção de duplicatas resultou em uma performance ligeiramente superior para o Naive Bayes quantitativo, destacando a importância dessas etapas no pré-processamento de dados.

5.5. Perceptron

O **Perceptron** é um dos algoritmos fundamentais no campo do aprendizado de máquina, sendo considerado a base das redes neurais artificiais. Desenvolvido por **Frank Rosenblatt** em 1958, ele é um modelo de **classificação linear** que busca separar os dados em diferentes classes por meio de um **hiperplano de decisão**. O algoritmo ajusta os pesos das conexões de entrada iterativamente, com base em um conjunto de exemplos rotulados, utilizando a **regra de aprendizado do Perceptron** que atualiza os pesos sempre que ocorre um erro na previsão. Esse processo continua até que todas as instâncias sejam classificadas corretamente ou que um número máximo de iterações seja atingido.

Apesar de sua simplicidade e eficiência computacional, o Perceptron possui algumas limitações. A principal delas é a incapacidade de lidar com **dados não linearmente separáveis** como demonstrado pelo **problema do XOR** que requer um modelo mais complexo para ser resolvido. Para contornar essa limitação, surgiram arquiteturas mais avançadas, como o **Perceptron Multicamadas (MLP)** que utiliza múltiplas camadas de neurônios e funções de ativação não lineares. No entanto, o Perceptron simples ainda é amplamente utilizado para tarefas de classificação binária e como um modelo introdutório para conceitos mais avançados em redes neurais.

5.5.1. Implementação Com Normalização e Sem Remoção de Duplicatas - Perceptron

A Figura 18 apresenta a matriz de confusão do modelo **Perceptron** treinado com os dados normalizados e sem remoção de instâncias duplicadas. Esse modelo de aprendizado supervisionado ajusta seus pesos iterativamente para encontrar um hiperplano que separe as classes de forma linear.

A acurácia obtida foi de **70,23%** indicando um desempenho competitivo na tarefa de classificação. Observando a matriz de confusão, nota-se que a **Classe 0** teve **12.313 previsões corretas** e **2.498 erros** enquanto a **Classe 1** obteve **8.570 acertos** e **6.355 previsões incorretas**. Essa diferença sugere que o modelo apresentou um **melhor recall para a Classe 0** capturando mais instâncias corretamente dessa categoria, porém com uma menor taxa de recuperação para a Classe 1.

Além disso, os valores de **precision** e **recall** reforçam que o Perceptron teve um equilíbrio razoável entre as classes, com um **f1-score médio de 70%**. Como o Perceptron é um classificador linear simples, seu desempenho pode ser afetado por padrões não linearmente separáveis nos dados, o que pode justificar algumas limitações na taxa de acertos.

5.5.2. Implementação Com Normalização e Remoção de Duplicatas - Perceptron

A Figura 19 apresenta a matriz de confusão do modelo **Perceptron** treinado com os dados normalizados e após a remoção das duplicatas. Esse modelo de aprendizado supervisionado ajusta iterativamente seus pesos para encontrar um hiperplano de separação entre as classes.

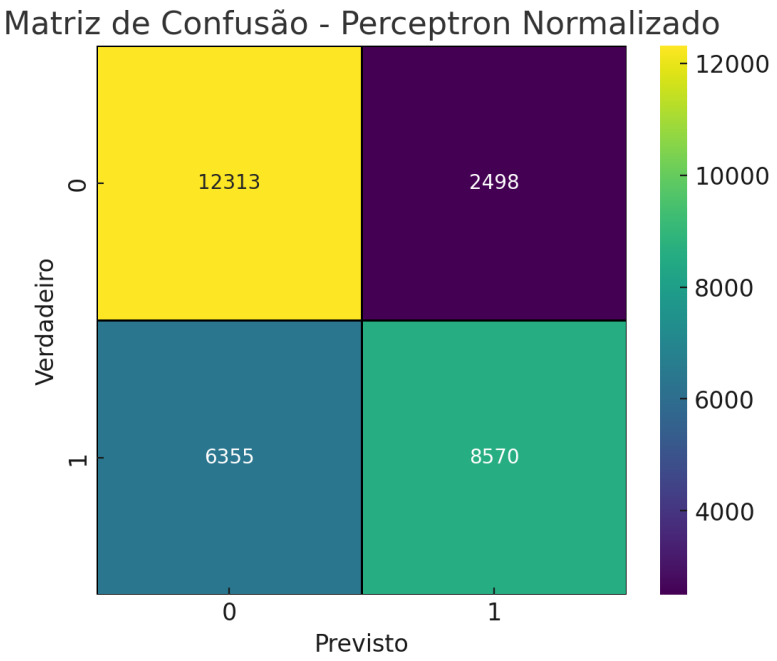


Figura 18. Matriz de Confusão Perceptron Normalizado e Sem Remoção de Duplicatas

No entanto, os resultados indicaram um desempenho insatisfatório, com uma acurácia de **50,25%** que equivale a uma classificação praticamente aleatória. Observando a matriz de confusão, percebe-se que o modelo classificou **todas as instâncias como pertencentes à Classe 1** sem identificar corretamente nenhuma instância da Classe 0. Esse comportamento sugere que o modelo não conseguiu encontrar um padrão de separação adequado após a remoção das duplicatas, resultando em um viés extremo para uma única classe.

Esse fenômeno pode estar relacionado à natureza linear do Perceptron, que pode ter sido impactado por uma distribuição menos informativa dos dados após a remoção das duplicatas. Além disso, a normalização pode ter influenciado a convergência do modelo, reforçando padrões inadequados na tomada de decisão.

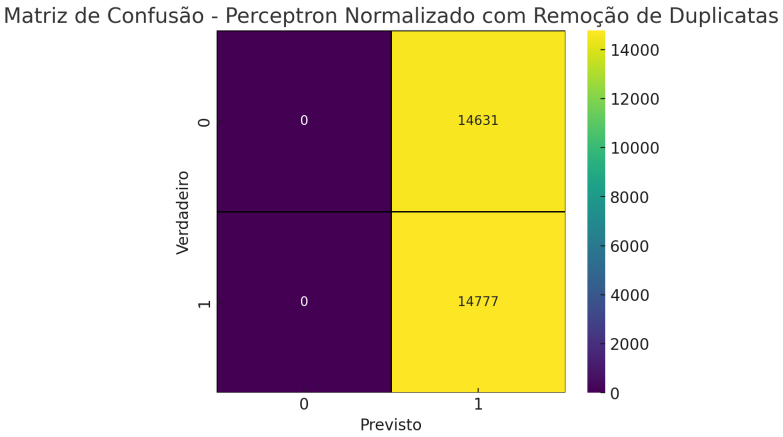


Figura 19. Matriz de Confusão Perceptron Normalizado e Com Remoção de Duplicatas

5.6. Regressão Linear

A regressão linear é um dos métodos estatísticos mais utilizados para modelagem preditiva, sendo amplamente aplicada em problemas de aprendizado de máquina supervisionado. Seu objetivo principal é encontrar a melhor relação entre uma variável dependente e uma ou mais variáveis independentes, minimizando a soma dos erros quadráticos entre os valores preditos e os valores reais. O modelo assume uma relação linear entre as variáveis, expressa por meio de uma equação do tipo:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon \tag{1}$$

onde y representa a variável alvo, x_i são as variáveis preditoras, β_i são os coeficientes do modelo e ϵ é o termo de erro.

No contexto da classificação, apesar da regressão linear ser tradicionalmente utilizada para problemas de regressão, sua aplicação pode ser estendida para problemas de classificação binária ao considerar um limiar para as previsões contínuas. Entretanto, a principal limitação dessa abordagem é a possibilidade de gerar previsões fora do intervalo $[0, 1]$, o que pode impactar sua interpretação probabilística. Ainda assim, sua simplicidade e interpretabilidade a tornam uma abordagem interessante para comparação com outros modelos mais complexos.

5.6.1. Implementação Sem Normalização e Sem Remoção de Duplicatas - Regressão Linear

A Figura 20 apresenta a matriz de confusão obtida a partir do modelo de **Regressão Linear** aplicado aos dados sem normalização e sem remoção de duplicatas. Esse modelo, embora tradicionalmente utilizado para tarefas de regressão, pode ser adaptado para classificação binária ao estabelecer um limiar de decisão sobre os valores contínuos preditos.

A análise dos resultados indica uma variância explicada de **0,72** sugerindo que o modelo conseguiu capturar uma parte significativa da variabilidade dos dados. No entanto, a matriz de confusão revela que o modelo ainda apresenta erros de classificação consideráveis, principalmente na distinção entre as classes. Esse comportamento pode ser atribuído ao fato de que a regressão linear não é ideal para problemas de classificação, pois não possui um mecanismo intrínseco para modelar relações não lineares ou para restringir suas previsões ao intervalo $[0, 1]$.

Apesar dessas limitações, a regressão linear continua sendo um modelo útil para fins comparativos, permitindo avaliar o desempenho de métodos mais sofisticados de aprendizado de máquina.

5.6.2. Implementação Com Normalização e Sem Remoção de Duplicatas - Regressão Linear

A Figura 21 apresenta a matriz de confusão obtida para o modelo de **Regressão Linear** aplicado aos dados normalizados, sem remoção de duplicatas. A normalização foi utili-

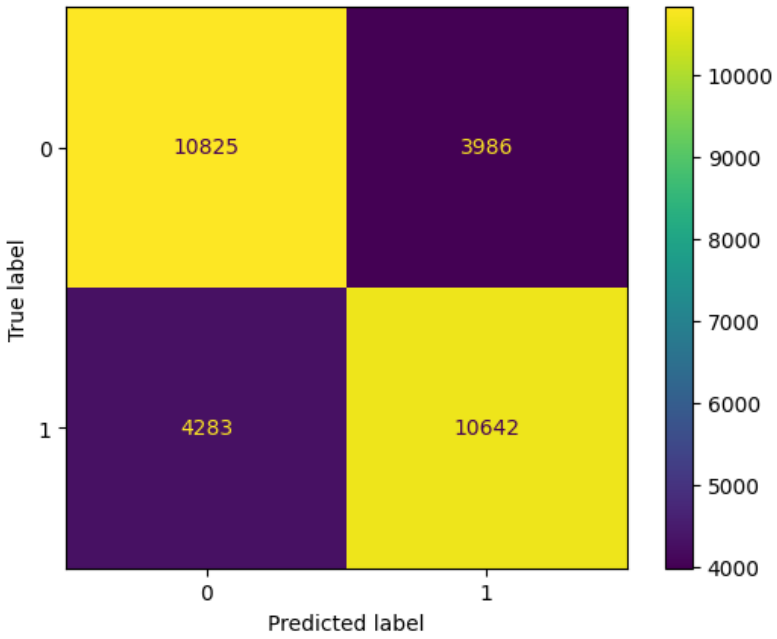


Figura 20. Matriz de Confusão Regressão Linear Sem Normalização e sem Remoção de Duplicatas

zada para padronizar as escalas das variáveis, permitindo que o modelo atribuísse pesos de maneira mais equilibrada.

A acurácia obtida foi de **0,72** demonstrando um desempenho satisfatório na tarefa de classificação binária. A normalização dos dados geralmente contribui para a melhora do desempenho de modelos baseados em distância e otimização contínua, como a regressão linear. No entanto, observa-se que, apesar da alta variância explicada e da melhoria na distribuição dos coeficientes do modelo, ainda há erros de classificação significativos.

Esse resultado reforça que, embora a regressão linear possa ser utilizada para classificação, sua aplicação apresenta limitações, especialmente quando comparada a algoritmos mais robustos projetados especificamente para esse tipo de problema.

5.6.3. Implementação Com Normalização e Remoção de Duplicatas - Regressão Linear

A Figura 22 exibe a matriz de confusão para o modelo de **Regressão Linear** aplicado aos dados normalizados e após a remoção de instâncias duplicadas. A remoção das duplicatas visa minimizar redundâncias e garantir que o modelo aprenda padrões mais representativos.

A acurácia obtida foi de **0,72** indicando que o desempenho permaneceu estável, mesmo após a eliminação de instâncias repetidas. Esse resultado sugere que a presença de duplicatas não influenciava significativamente o aprendizado do modelo, reforçando que a regressão linear, apesar de não ser um método clássico para classificação, ainda consegue obter um desempenho competitivo nesta tarefa.

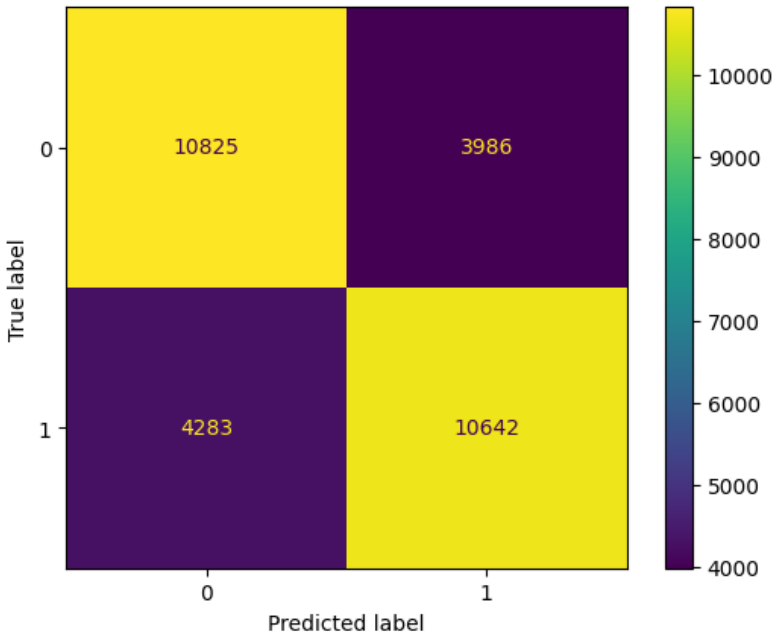


Figura 21. Matriz de Confusão Regressão Linear Normalizado e sem Remoção de Duplicatas

A remoção de dados repetidos pode ser benéfica para alguns algoritmos, reduzindo o viés introduzido por amostras idênticas e melhorando a generalização. No entanto, no caso da regressão linear, os resultados permaneceram consistentes, mostrando que o modelo já estava aprendendo de maneira eficiente a partir dos dados originais.

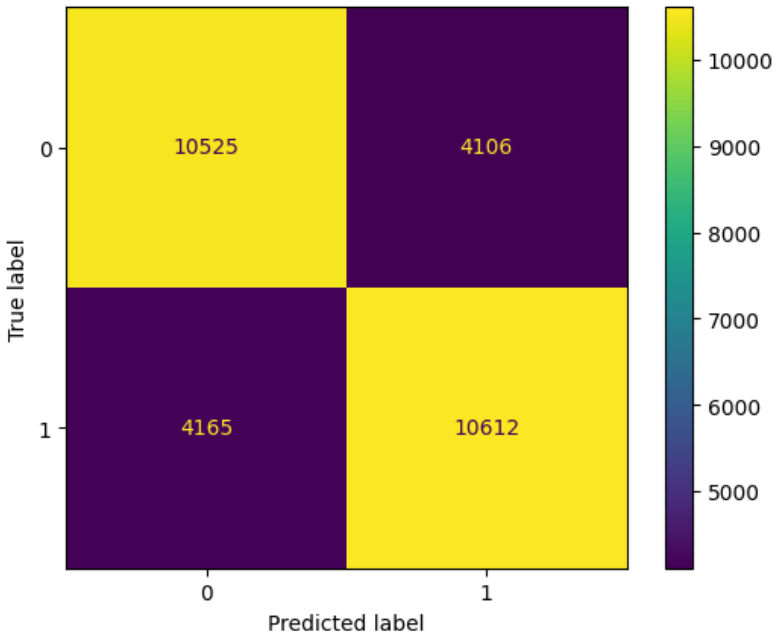


Figura 22. Matriz de Confusão Regressão Linear Normalizado e Com Remoção de Duplicatas

6. Conclusão

Neste estudo, foram explorados diversos algoritmos de aprendizado de máquina para a classificação de indivíduos que consomem ou não bebidas alcoólicas. A análise foi conduzida utilizando diferentes abordagens, considerando dados sem normalização, com normalização e com a remoção de duplicatas, permitindo avaliar o impacto dessas transformações nos resultados dos modelos.

Dentre os algoritmos testados, destacam-se a **Árvore de Decisão (CART)**, que mostrou um desempenho intermediário, o **Support Vector Machine (SVM)**, que apresentou bons resultados, especialmente em sua configuração padrão, e o **Naive Bayes**, tanto na versão qualitativa quanto quantitativa, que teve desempenho competitivo, embora com limitações na capacidade preditiva em certos cenários. Além disso, o **Perceptron**, um modelo linear simples, demonstrou limitações em certos cenários, especialmente ao classificar todas as instâncias em uma única classe quando aplicado a dados normalizados e sem duplicatas. A **Regressão Linear**, apesar de ser um modelo originalmente voltado para regressão, foi empregada para classificação e apresentou um desempenho consistente, alcançando uma acurácia competitiva.

O algoritmo **K-Nearest Neighbors (KNN)** foi analisado em profundidade, considerando diferentes métricas de distância e variações do hiperparâmetro k. Inicialmente, foram testados valores de k variando de 1 a 30, identificando um desempenho promissor para valores mais altos de k. Para um ajuste mais refinado, foram testados intervalos ampliados: primeiro k variando de **30 a 300** em incrementos de 10 e, posteriormente, um ajuste mais fino entre **100 e 140** com incrementos unitários. Como resultado dessa análise, identificou-se que o **melhor valor de k foi 126 utilizando a distância Euclidiana**, resultando em uma acurácia final de **71,61%**.

Com base nos experimentos conduzidos, o **KNN foi escolhido como o melhor algoritmo**para a tarefa em questão, considerando seu desempenho consistente e sua adaptabilidade ao conjunto de dados utilizado. A análise demonstrou que o ajuste correto do hiperparâmetro k e a escolha adequada da métrica de distância são fatores determinantes para a eficácia do modelo. Este estudo reforça a importância da experimentação e do ajuste fino de modelos em problemas de classificação, garantindo a obtenção de previsões mais precisas e confiáveis.