

1 METODOLOGIA

O SPAECE (Sistema Permanente de Avaliação da Educação Básica do Ceará) é uma iniciativa do Governo do Estado do Ceará, através da Secretaria da Educação (SEDUC), para avaliar o desempenho dos estudantes das escolas públicas estaduais. O objetivo do SPAECE é monitorar a qualidade da educação e fornecer informações que possam subsidiar políticas públicas e estratégias pedagógicas para a melhoria do ensino. O SPAECE avalia estudantes de diversas etapas da educação básica, incluindo os anos finais do Ensino Fundamental (5º e 9º anos) e o 3º ano do Ensino Médio. As avaliações focam principalmente nas áreas de Língua Portuguesa e Matemática, buscando verificar as competências e habilidades dos alunos nessas disciplinas (JÚNIOR; FARIAS, 2016; LIMA; ANDRADE, 2008). Na edição de 2018 do SPAECE, os resultados para os alunos do 3º ano do Ensino Médio foram analisados considerando os níveis de proficiência em Língua Portuguesa e Matemática. Os dados são apresentados em níveis de desempenho que vão de "Muito Crítico" a "Desejável".

- Língua Portuguesa:

- Desempenho Geral: A maioria dos alunos se encontrava nos níveis "Crítico" e "Intermediário", com uma porcentagem menor atingindo os níveis mais altos de proficiência ("Adequado" e "Desejável").
- Distribuição de Níveis: Uma pequena fração dos alunos atingiu o nível "Desejável", que representa uma proficiência adequada para a conclusão do Ensino Médio e preparação para o ensino superior ou mercado de trabalho.

- Matemática:

- Desempenho Geral: Similar à Língua Portuguesa, a maioria dos estudantes ficou nos níveis "Crítico" e "Intermediário".
- Distribuição de Níveis: Poucos alunos alcançaram os níveis "Adequado" e "Desejável", indicando que a Matemática é uma área com maiores desafios

para os alunos do 3º ano do Ensino Médio.

1.1 PROPOSTA

Os resultados do SPAECE 2018 apontam para a necessidade de intervenções mais robustas e direcionadas no ensino médio, especialmente nas áreas de Língua Portuguesa e Matemática. Realizar a previsão das proficiências em Língua Portuguesa e Matemática dos estudantes cearenses no SPAECE, comparando as técnicas de Aprendizagem de Máquina (AM) e Aprendizagem Profunda (AP), além de indicar os principais atributos preditores para o desempenho dos alunos. Principais questões de pesquisa:

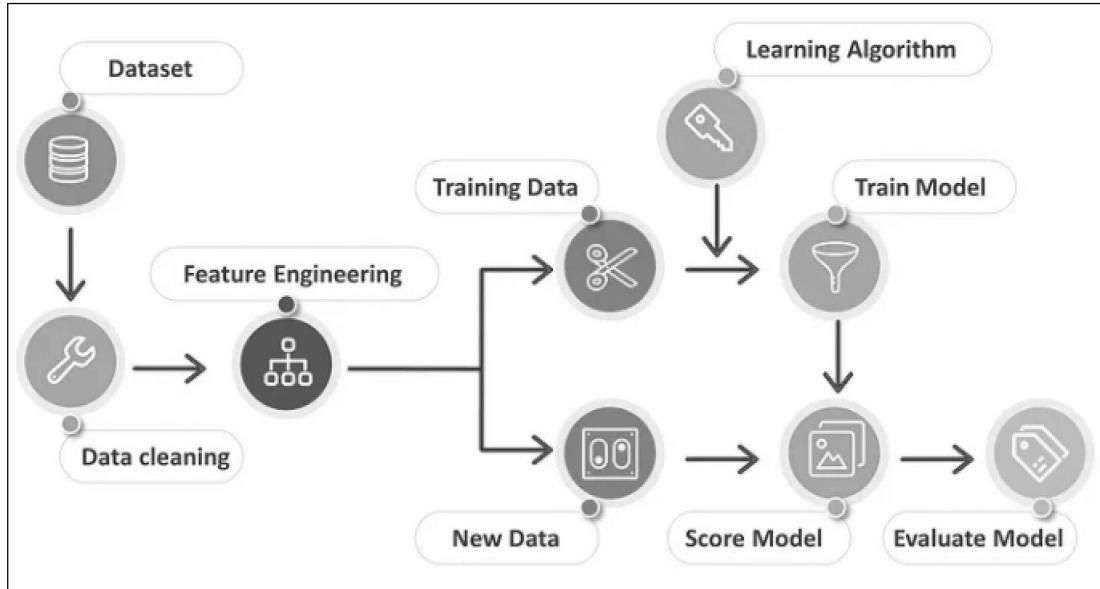
1. Qual conjunto de atributos tem mais influência na previsão do desempenho dos estudantes?
2. Qual a eficácia de modelos gerados a partir de algoritmos baseados em Aprendizagem de Máquina e Aprendizagem Profunda na previsão das proficiências de aprendizagem dos estudantes no SPAECE?
3. Os modelos baseados em algoritmos de Aprendizagem Profunda têm uma melhor eficiência quando comparados aos modelos baseados em algoritmos de Aprendizagem de Máquina utilizados na Mineração de Dados Educacionais?

Os resultados obtidos podem auxiliar as escolas e a Secretaria da Educação a desenvolver ações específicas para melhorar a qualidade de ensino, como:

- Formação Continuada de Professores: Investir na capacitação dos professores para melhorar as práticas pedagógicas.
- Programas de Reforço Escolar: Implementar programas de reforço e recuperação para alunos com dificuldades de aprendizagem.
- Apoio e Monitoramento: Fortalecer o acompanhamento pedagógico e psicológico dos estudantes para identificar e intervir precocemente em dificuldades.
- Incentivo ao Uso de Tecnologias: Utilizar ferramentas tecnológicas e metodologias inovadoras que possam tornar o aprendizado mais atraente e eficiente.

A abordagem do projeto pode ser simplificada no diagrama da figura 1 e melhor detalhada nos tópicos posteriores. O fluxo de um processo de aprendizado de máquina aplicado ao contexto educacional descrito. A seguir, uma descrição dos principais passos do processo:

Figura 1 – Diagrama do Processo de Aprendizagem de Máquina



Fonte: <https://medium.datadriveninvestor.com/data-preprocessing-3cd01eef438>

1. **Dataset:** Coleta de dados dos alunos, incluindo resultados de avaliações, informações socioeconômicas, entre outros.
2. **Data Cleaning:** Limpeza dos dados para remover inconsistências e valores ausentes.
3. **Feature Engineering:** Seleção e transformação das variáveis que serão usadas no modelo.
4. **Training Data:** Divisão dos dados em conjuntos de treino e teste.
5. **Train Model:** Treinamento do modelo utilizando algoritmos de aprendizado de máquina.

6. **Score Model:** Avaliação do desempenho do modelo nos dados de teste.
7. **Evaluate Model:** Avaliação final do modelo, considerando métricas de desempenho e validação.

A utilização de técnicas avançadas de análise de dados permitirá identificar padrões e informações valiosas que podem orientar intervenções mais eficazes e direcionadas para melhorar a educação no estado do Ceará.

1.1.1 Conjunto de Dados

O SPAECE (Sistema Permanente de Avaliação da Educação Básica do Ceará) coleta uma ampla gama de dados por meio de seus questionários, com o objetivo de entender melhor os fatores que influenciam o desempenho dos alunos e, consequentemente, melhorar a qualidade da educação no estado. Esses dados são geralmente divididos em duas categorias principais: dados contextuais socioeconômicos e dados das avaliações de desempenho acadêmico.

1.1.1.1 Dados Contextuais Socioeconômicos

Os dados contextuais socioeconômicos são coletados para obter um panorama abrangente sobre o ambiente em que os alunos estão inseridos. Esses dados incluem informações sobre:

- Perfil do Aluno:
 - Idade
 - Sexo
 - Raça/etnia
- Condições Socioeconômicas:
 - Renda familiar
 - Nível de escolaridade dos pais ou responsáveis

- Ocupação dos pais ou responsáveis
- Número de pessoas na residência
- Condições de moradia (tipo de habitação, posse de bens, acesso a serviços básicos como água e eletricidade)
- Aspectos Educacionais:
 - Tipo de escola (urbana ou rural)
 - Tipo de transporte utilizado para chegar à escola
 - Frequência de faltas escolares
 - Participação em atividades extracurriculares (esportes, música, etc.)
- Ambiente de Estudo:
 - Disponibilidade de um lugar adequado para estudar
 - Acesso a materiais escolares e livros
 - Uso de tecnologias como computadores, tablets e internet para fins educacionais

1.1.1.2 Dados das Avaliações

Os dados das avaliações são obtidos através de provas padronizadas aplicadas aos alunos. Esses dados incluem:

- Desempenho Acadêmico:
 - Notas obtidas nas avaliações de Língua Portuguesa e Matemática
 - Níveis de proficiência, que variam de "Muito Crítico" a "Desejável"
 - Comparação de desempenho entre diferentes escolas e regiões
- Distribuição de Desempenho:
 - Percentual de alunos em cada nível de proficiência
 - Identificação de padrões de desempenho em diferentes grupos demográficos e socioeconômicos
- Progressão Escolar:

- Taxas de aprovação e reprovação
- Evolução do desempenho ao longo dos anos

1.1.1.3 Dados Aplicados no Projeto

Foram realizados testes em 2 conjuntos de dados. O primeiro conjunto de dados aplicado nos testes consiste em 54.500 instâncias, sendo cada uma delas correspondente a um aluno; e 18 atributos, 14 atributos categóricos textuais correspondentes à algumas perguntas de questionário socioeconômico e 4 atributos numéricos correspondentes aos escores e níveis de proficiência obtidos nas avaliações de Matemática e Português. Dados estes, que foram coletados na edição de 2018 do exame. O *dataset* foi denominado de “SPAECE 2018” e estruturado no formato “.csv” e não possuía dados faltantes ou inválidos. O segundo *dataset* consiste em um conjunto de dados composto também por 54.500 instâncias, mas com 55 categorias correspondentes ao questionário socioeconômico completo. Nomeado como "Dados Contextuais Codificados" e estruturado também no formato ".csv". Exemplificação do processo de codificação categórica das pontuações:

- **Prova de Matemática:**

- Número total de questões: 40
- Cada questão correta vale 1 ponto.
- Aluno A acertou 30 questões.
- Pontuação bruta do Aluno A: 30 pontos.

- **Escalonamento:**

- Pontuação bruta de 30 é convertida para uma escala de 0 a 500.
- Após o escalonamento, a pontuação do Aluno A pode ser, por exemplo, 300 pontos.

- **Nível de Proficiência:**

- A pontuação de 300 pontos pode ser classificada como "Intermediário" conforme

a definição dos níveis de proficiência do SPAECE.

Dados numéricos são úteis para previsões usando regressão e dados categóricos podem ser aplicados em problemas de classificação. Podemos também transformar os dados numéricos em codificação de categorias e vice-versa. Por exemplo:

- Escalonamento:
 - **Muito Crítico:** 0 a 125 pontos
 - **Crítico:** 126 a 200 pontos
 - **Intermediário:** 201 a 275 pontos
 - **Adequado:** 276 a 350 pontos
 - **Desejável:** 351 a 500 pontos

Assim, dividimos os dados numéricos em 5 categorias diferentes, fazendo a possível a classificação dos mesmos. É possível aplicar o mesmo tipo de codificação para os dados numéricos de proficiência, criando assim variações do *dataset* original que melhor se aplicam a diferentes aplicações e modelos de aprendizagem de máquina. O ajuste e codificação do *dataset* será melhor abordado no tópico posterior: “1.1.4.2 Tipos de Codificações Aplicadas”.

1.1.2 Limpeza de Dados

A limpeza de dados é uma etapa crucial no processo de análise de dados, especialmente em projetos de aprendizado de máquina e mineração de dados, como no contexto do SPAECE. Essa fase assegura que os dados utilizados são de alta qualidade, livres de erros e inconsistências, e adequados para a análise e aplicação subsequentes (HARIHARAKRISHNAN *et al.*, 2017; GARCÍA; LUENGO; HERRERA, 2015; RAHM; DO *et al.*, 2000). O processamento dos dados foi simplificado nas seguintes etapas:

- **Verificação de Integridade:** Inicialmente, procedeu-se à verificação da integridade dos dados coletados, garantindo a presença de todos os registros esperados e a ausência de duplicações ou perdas significativas de informações. Esta verificação

foi conduzida por meio de planilhas, considerando que o conjunto de dados original está no formato ".csv", bem como através de *datagramas* gerados pelo pacote *Python* Pandas. Dado que cada entrada corresponde a um aluno e possui um identificador único (nome), esta etapa do processamento foi simplificada.

- **Verificação e Tratamento de Valores Ausentes:** Valores ausentes são comuns em grandes conjuntos de dados e podem ocorrer devido a diversos motivos, como falhas na coleta de dados ou respostas incompletas dos questionários. No conjunto de dados abordado não foram identificados valores ausentes.
- **Correção de Erros e Padronização dos Dados:** Os dados podem conter erros, como valores fora do intervalo esperado, formatação incorreta ou entradas duplicadas. Neste caso, foram encontrados erros na notação numérica de ponto flutuante dos atributos de Pontuação e Proficiência, uma possuía notação de ponto flutuante com "."(notação americana) e outra em ponto flutuante com ","(notação numérica nacional). Este erro de formatação e padronização dos dados pode impactar negativamente na interpretação dos dados, no treinamento e predição dos modelos. A formatação foi corrigida usando *Python*.

Figura 2 – Código *Python* para Tratamento de Ponto Flutuante.

```
# Formatando colunas que estão com float separados por vírgula
gnetDf['PROFICIENCIA EM PORTUGUES'] = gnetDf['PROFICIENCIA EM PORTUGUES'].str.replace(',', '.').astype(float)
gnetDf['PROFICIENCIA EM MATEMATICA'] = gnetDf['PROFICIENCIA EM MATEMATICA'].str.replace(',', '.').astype(float)
✓ 0.0s
```

Python

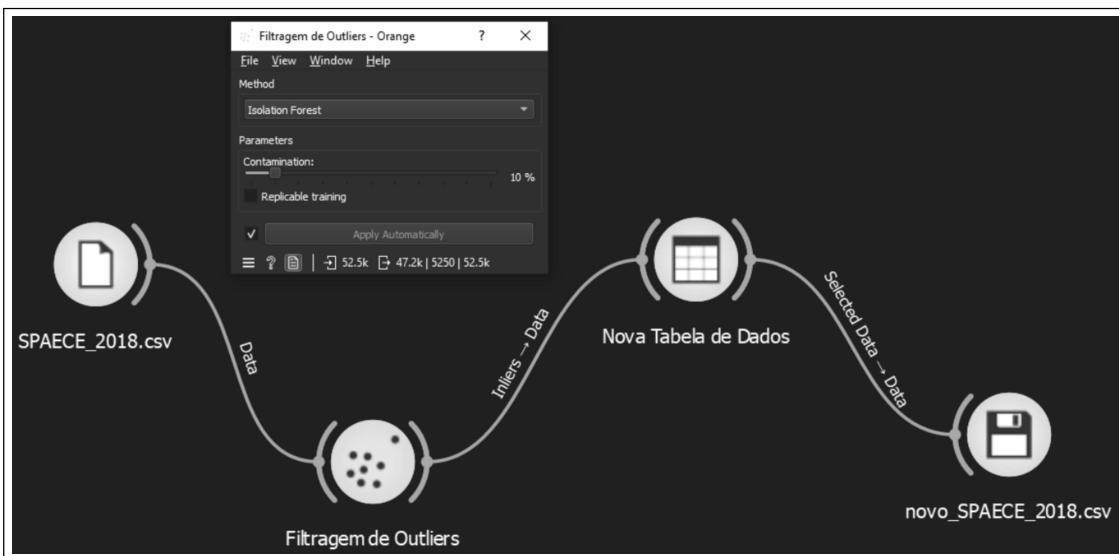
Fonte: Elaborado pelo Autor

1.1.2.1 Remoção de *Outliers* com *Orange Data Mining*

Outliers são valores que se desviam significativamente do restante dos dados em um conjunto de dados. Eles podem surgir por várias razões, incluindo erros de coleta de dados, variabilidade natural ou fenômenos extremos. A detecção e o tratamento

de *outliers* são etapas essenciais na limpeza de dados, pois os *outliers* podem afetar negativamente a análise e os resultados dos modelos de aprendizado de máquina. Para tratar possíveis *outliers* no conjunto de dados, foi aplicada a ferramenta *Orange Data Mining*. Esta é uma plataforma de código aberto desenvolvida para análise de dados, aprendizado de máquina, e mineração de dados. É conhecida por sua interface gráfica amigável, que permite aos usuários construir fluxos de trabalho de análise de dados de maneira intuitiva e visual, sem necessidade de programação e que oferece várias opções para a detecção e tratamento de *outliers*, uma das quais é o uso do método *Isolation Forest* que baseia-se em árvores para detectar *outliers*. Diferentemente de outros algoritmos de detecção de *outliers* que modelam a distribuição dos dados, o *Isolation Forest* explicitamente isola observações (LIU; TING; ZHOU, 2008; CHANDOLA; BANERJEE; KUMAR, 2009; AGGARWAL; AGGARWAL, 2017). A premissa é que *outliers* são poucos e diferentes, portanto, mais fáceis de isolar. Abaixo o diagrama de *Widgets* do *Orange*:

Figura 3 – Diagrama de Filtragem de *Outliers* com *Orange Data Mining*



Fonte: Elaborado pelo Autor

Como resultado obteve-se um novo conjunto de dados do primeiro *dataset* com 47.200 instâncias e livre de *outliers*. Este novo conjunto de dados foi salvo como “novo SPAECE.csv”. E um novo conjunto de dados derivado do segundo *dataset* de dados contextuais, contendo aproximadamente 54.000 amostras e salvo como “novo Dados Contextuais Codificados.csv”

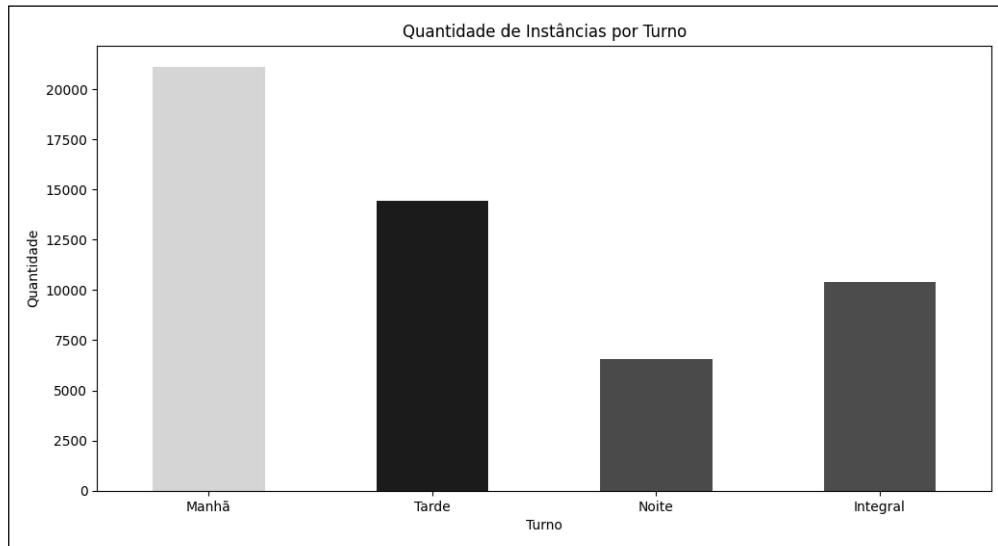
1.1.2.2 Visualização de Dados

A exploração e visualização dos dados é uma etapa fundamental para entender o comportamento das variáveis e suas relações. Utilizamos diferentes tipos de gráficos e análises estatísticas para identificar padrões, tendências e possíveis *outliers*. Através do “*pyplot*” pode-se plotar gráficos e visualizar informações como distribuição de dados e possíveis correlações de atributos. O “*pyplot*” é um módulo dentro da biblioteca “*matplotlib*” que oferece uma interface de programação semelhante ao *MATLAB* para plotagem de gráficos em *Python*. Ele fornece uma maneira conveniente de criar figuras, eixos, gráficos de linha, de dispersão, histogramas, barras, entre outros tipos de visualizações (YIM; CHUNG; YU, 2018; SIAL; RASHDI; KHAN, 2021; SAHOO *et al.*, 2019). Alguns dos gráficos gerados e dados extraídos:

No gráfico da Figura 4 podemos observar a distribuição das instâncias (eixo y) nos grupos definidos pelos turnos escolares dos alunos (eixo x). Sendo a maioria dos alunos, estudantes do turno diurno.

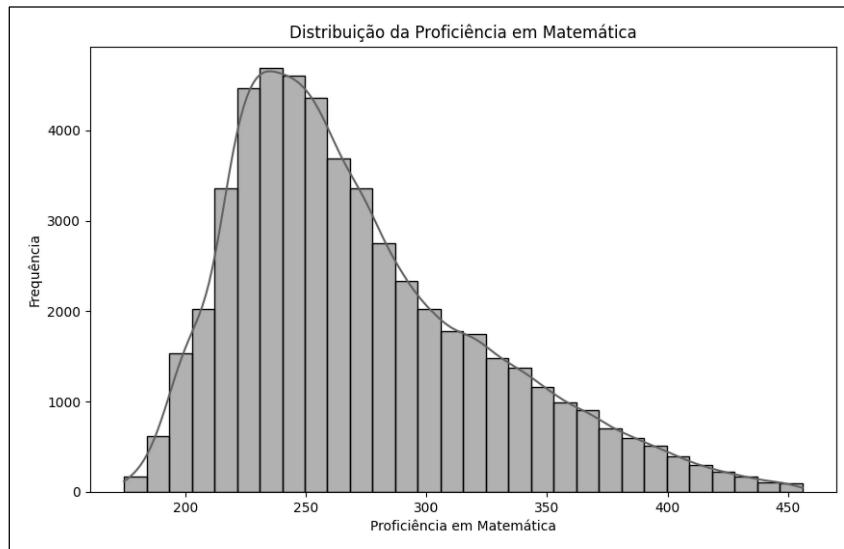
O gráfico da Figura 5 representa os diferentes valores de proficiência em matemática no eixo x e o número de alunos que possuem determinada proficiência no eixo y. É notável uma distribuição assimétrica à direita, ou seja, a cauda se estende mais para a direita. Apontando que a maioria dos alunos possui uma proficiência em matemática entre 200 e 300 pontos. O valor mais frequente (pico do histograma), ou moda, está em torno de 250 pontos e há uma variação considerável na proficiência, com uma cauda que se estende até cerca de 450 pontos, indicando uma diversidade

Figura 4 – Gráfico de Barras da Quantidade de Instâncias por Turno Escolar



Fonte: Elaborado pelo Autor

Figura 5 – Histograma de Distribuição da Proficiência em Matemática.

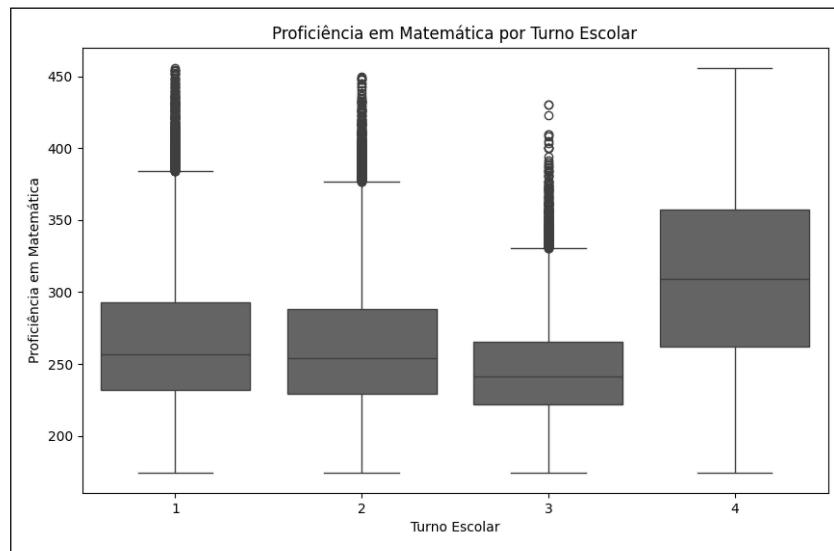


Fonte: Elaborado pelo Autor

significativa no desempenho dos alunos. A cauda da direita se estende para valores mais altos de proficiência, mas com menor frequência. Isso indica que, embora existam alunos com alta proficiência, eles são menos comuns. Com essas informações é possível

observar:

Figura 6 – Histograma de Distribuição da Proficiência em Matemática.



Fonte: Elaborado pelo Autor

- **Desempenho Médio:** a maioria dos alunos tem uma proficiência em matemática concentrada em torno de 250 pontos, o que pode ser interpretado como a média geral dos alunos. A assimetria à direita sugere que há mais alunos com proficiência abaixo da média, mas com uma presença notável de alunos com desempenho acima da média.
- **Variabilidade:** A presença de uma cauda longa indica que há uma variabilidade significativa no desempenho, com alguns alunos apresentando proficiência muito alta. A dispersão larga sugere que diferentes fatores podem influenciar o desempenho dos alunos.

Na tentativa de encontrar o fator mais impactante no desempenho dos alunos, o Gráfico da Figura 6 foi gerado. Onde o eixo x representa os diferentes turnos escolares (1: Manhã, 2: Tarde, 3: Noite, 4: Integral) e o eixo y representa os valores de proficiência em matemática dos alunos. Observa-se a seguinte distribuição:

- **Turno 1 (Manhã)::**

- A mediana está em torno de 275.
- O intervalo interquartil (IQR) é aproximadamente de 250 a 300.
- Muitos *outliers* acima de 300.
- A dispersão dos dados é maior em comparação com outros turnos, indicando uma variabilidade significativa na proficiência dos alunos.

- **Turno 2 (Tarde)::**

- A mediana está ligeiramente abaixo de 275.
- O IQR é similar ao do turno da manhã, mas a dispersão parece um pouco menor.
- Também possui muitos *outliers* acima de 300, mas menos do que o turno da manhã.

- **Turno 3 (Noite)::**

- A mediana está abaixo de 250.
- O IQR vai de aproximadamente 225 a 275.
- Menos dispersão comparada aos turnos da manhã e da tarde, mas ainda com *outliers* significativos.

- **Turno 4 (Integral)::**

- A mediana está acima de 300, a maior entre todos os turnos.
- O IQR vai de aproximadamente 275 a 350.
- Menor número de outliers e menos dispersão, indicando uma maior consistência na proficiência dos alunos nesse turno.

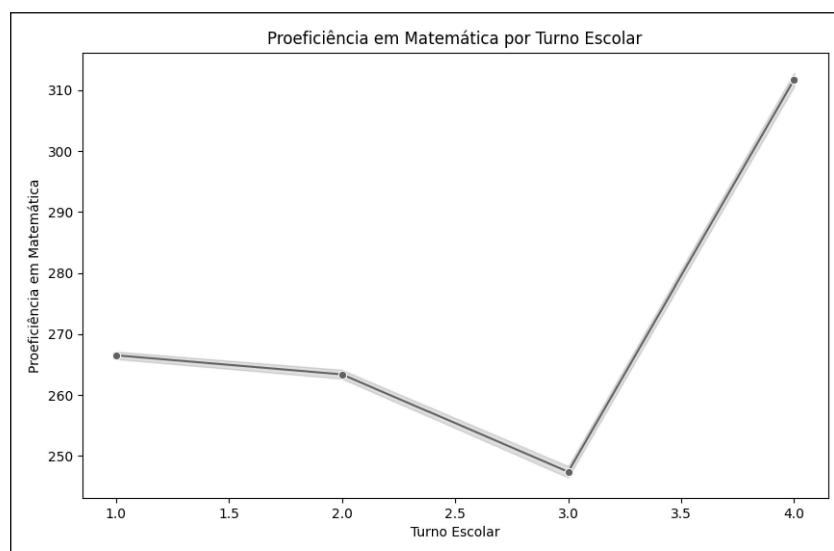
Estas informações proporcionam as seguintes interpretações:

- **Consistência e Desempenho:** Alunos do turno integral (4) têm, em média, uma proficiência maior em matemática comparada aos alunos dos outros turnos. A consistência é maior no turno integral, sugerindo que os alunos nesse turno têm desempenho mais uniforme.

- **Variabilidade:** Os turnos da manhã (1) e tarde (2) mostram grande variabilidade nos resultados dos alunos, o que pode indicar diferenças significativas no ensino ou nas condições de aprendizagem. O turno da noite (3) tem menor desempenho mediano e uma menor variabilidade interna, mas ainda apresenta *outliers* significativos.
- **Impacto do Turno:** O turno escolar parece ter um impacto significativo na proficiência em matemática dos alunos. Programas integrais (turno 4) podem oferecer um ambiente mais favorável para a aprendizagem de matemática, possivelmente devido a mais tempo de estudo, recursos ou métodos de ensino.

O que sugere que o turno escolar influencia significativamente a proficiência dos alunos em matemática. Alunos em turnos integrais tendem a ter melhores resultados, com menor variabilidade e menos *outliers* negativos. No gráfico da Figura 7 podemos confirmar que a situação se repete quando observamos os dados de Pontuação em Matemática e os dados de Proficiência. Relações como esta são observadas pelo algoritmo de Aprendizagem de Máquina.

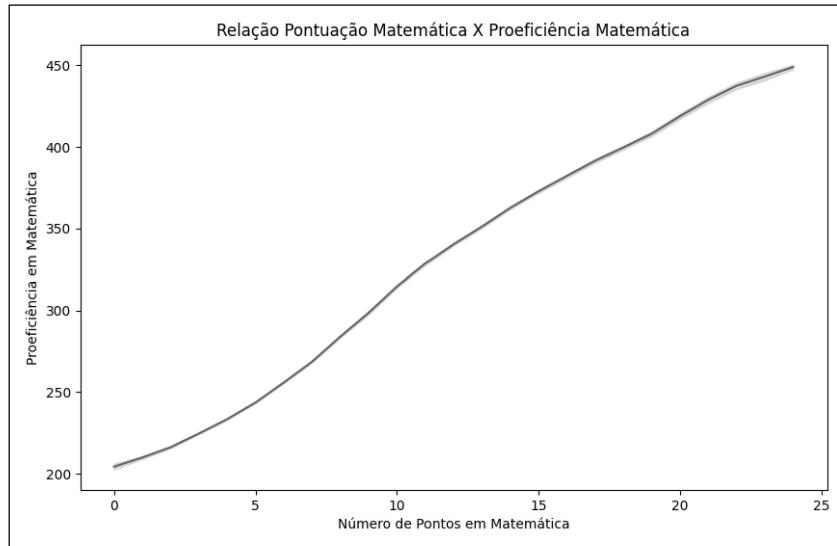
Figura 7 – Relação Proficiência em Matemática x Turno Escolar.



Fonte: Elaborado pelo Autor

No gráfico da Figura 8 tem-se a pontuação direta dos alunos em matemática como eixo x e a medida de proficiência em matemática dos alunos (derivada da pontuação) como eixo y . O gráfico mostra uma relação aproximadamente linear entre a pontuação direta e a proficiência. Isso indica que conforme a pontuação dos alunos aumenta, sua proficiência também aumenta de maneira proporcional. Dada a natureza dos dados, onde a proficiência é derivada diretamente da pontuação, é esperado observar uma forte correlação linear. Este é um exemplo claro de multicolinearidade, onde duas variáveis estão intimamente relacionadas.

Figura 8 – Relação de Proficiência em Matemática e Pontuação em Matemática.



Fonte: Elaborado pelo Autor

Como a proficiência é calculada diretamente a partir da pontuação, a forte correlação não é surpreendente. Contudo, esta relação pode introduzir viés ao utilizar esses dados em modelos de aprendizado de máquina, pois o modelo pode "aprender" esta relação artificialmente inflada. Então a presença de uma alta correlação entre pontuação e proficiência pode resultar em problemas de multicolinearidade, especialmente em modelos de regressão linear ou em técnicas que assumem independência das *features*.

Isso pode levar a coeficientes instáveis e a dificuldades na interpretação dos resultados. A avaliação do modelo deve considerar o potencial viés introduzido por essa correlação. É crucial utilizar técnicas de validação cruzada ou desconsiderar um dos atributos para garantir que o modelo não esteja super ajustado aos dados devido à alta correlação entre essas variáveis. O mesmo cenário é válido para a relação Pontuação em Português e Proficiência em Português.

1.1.2.3 Matriz de Correlação

Para observar com maior precisão a correlação entre os atributos, foi criada uma matriz de correlação. A matriz de correlação é uma representação tabular que mostra as relações lineares entre todas as variáveis em um *dataset*. Cada célula na matriz contém o coeficiente de correlação entre duas variáveis. O coeficiente de correlação é uma medida estatística que indica o grau de relação linear entre duas variáveis. Ele varia de -1 a 1, onde:

- 1 indica uma correlação positiva perfeita: à medida que uma variável aumenta, a outra também aumenta de forma proporcional.
- -1 indica uma correlação negativa perfeita: à medida que uma variável aumenta, a outra diminui de forma proporcional.
- 0 indica ausência de correlação linear: não há relação linear entre as variáveis.

A matriz de correlação é frequentemente visualizada como um *heatmap*, onde cores diferentes representam diferentes níveis de correlação. Cores mais claras indicam correlação mais forte (positiva ou negativa), enquanto cores mais escuras indicam correlação mais fraca ou ausência de correlação. A análise da matriz de correlação pode fornecer dados valiosos, como:

- **Identificação de variáveis fortemente correlacionadas:** variáveis altamente correlacionadas podem fornecer informações redundantes e podem ser candidatas para redução de dimensionalidade.

- **Identificação de relações interessantes:** correlações entre variáveis podem revelar padrões ou relações importantes no dataset.
- **Seleção de variáveis para modelagem:** variáveis altamente correlacionadas podem ser eliminadas para simplificar modelos e evitar multicolinearidade.

No contexto de um projeto de análise de dados ou modelagem preditiva, a matriz de correlação é uma ferramenta poderosa para explorar as relações entre as variáveis e tomar decisões informadas sobre o processo de modelagem (STEIGER, 1980; GU, 2022). A Matriz de Correlação resultante pode ser visualizada na Figura 9:

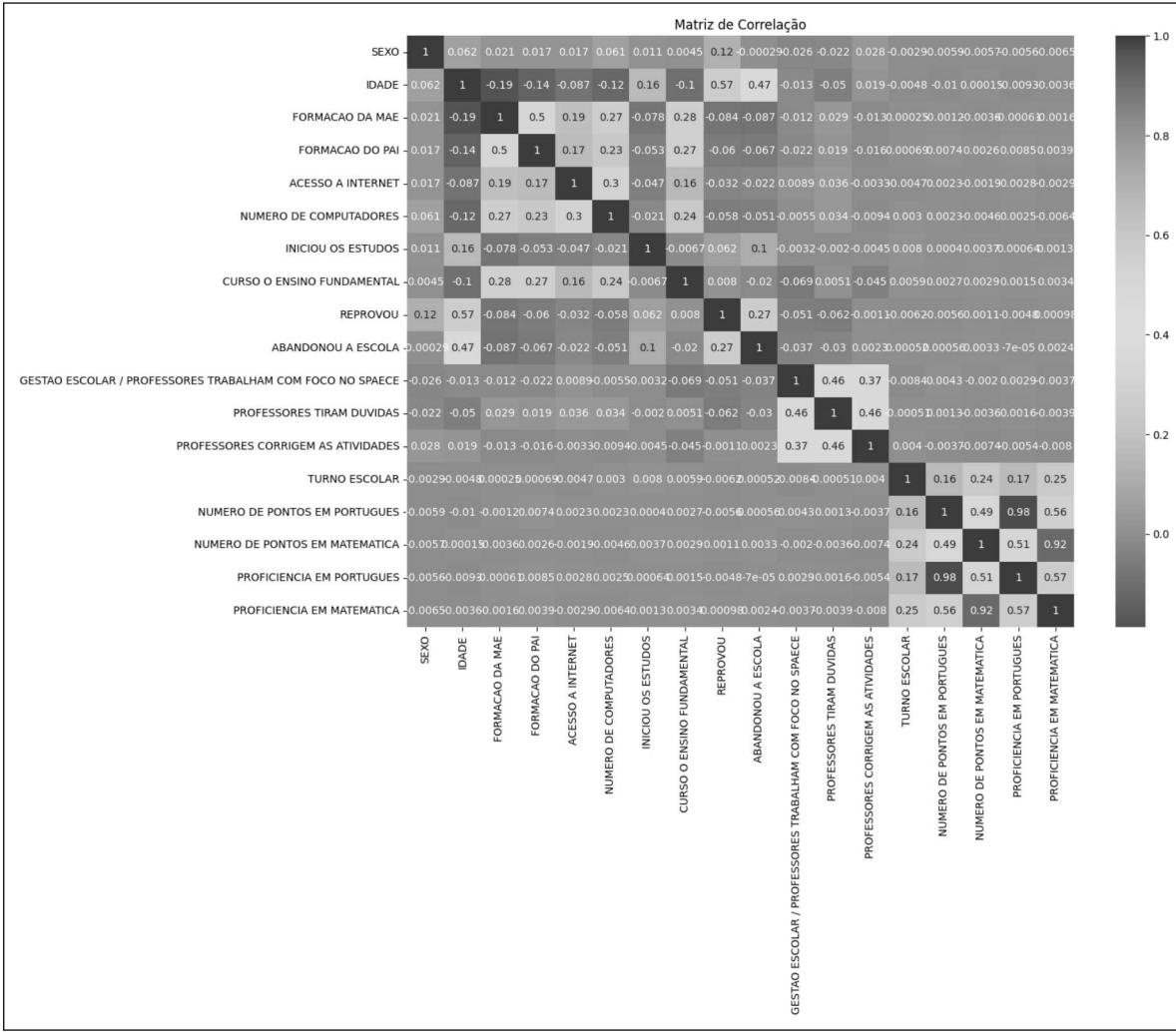
1.1.3 Seleção de Atributos

Conjuntos de dados frequentemente contêm muitos atributos, alguns dos quais podem não contribuir significativamente para a predição do modelo. A seleção de *features* ajuda a reduzir o número de atributos, simplificando assim a complexidade do modelo e melhorando o desempenho computacional. Ao remover atributos irrelevantes ou redundantes, a seleção de atributos pode melhorar a precisão, a generalização e a interpretabilidade dos modelos de ML. Modelos treinados com um conjunto de *features* mais relevantes tendem a ter um desempenho melhor em novos dados. A seleção de *features* também pode reduzir o *overfitting*, fenômeno no qual o modelo se ajusta muito bem aos dados de treinamento, mas falha em generalizar para novos dados (YU; LIU, 2003; GUYON; ELISSEEFF, 2003; JOVIĆ; BRKIĆ; BOGUNOVIĆ, 2015; TUV *et al.*, 2009). Reduzir a dimensionalidade dos dados pode ajudar a mitigar esse problema. Os métodos de Seleção de *Features* aplicados no projeto foram os seguintes:

1.1.3.1 Seleção de Atributos com Árvores de Decisão

A árvore de decisão é uma técnica popular para seleção de *features*, especialmente em problemas de classificação e regressão. Ela funciona dividindo o conjunto de

Figura 9 – Matriz de Correlação de Atributos



Fonte: Elaborado pelo Autor

dados em subconjuntos menores com base nos atributos mais importantes. O critério de divisão pode ser medido usando diferentes métodos, como ganho de informação, índice Gini ou erro quadrático médio.

- **Régressão:** Para problemas de regressão, as árvores de decisão são usadas para identificar as *features* mais importantes para prever a variável alvo. As *features* são selecionadas com base em sua capacidade de reduzir a variabilidade na variável de

resposta.

- **Classificação:** Para problemas de classificação, as árvores de decisão são usadas para classificar os dados com base nas *features* mais discriminativas. As *features* são selecionadas com base em sua capacidade de separar as diferentes classes no conjunto de dados.

1.1.3.2 Seleção de Atributos com *Features* Automáticas do *Sklearn*

O *SKlearn* oferece várias técnicas para seleção automática de *features*, incluindo *SelectKBest*, *SelectPercentile*, *Recursive Feature Elimination* (RFE) e *SelectFromModel*. Essas técnicas utilizam diferentes critérios para selecionar as melhores *features*, como pontuação de teste estatístico, importância de atributos em modelos de aprendizado de máquina ou recursão sobre subconjuntos de *features*.

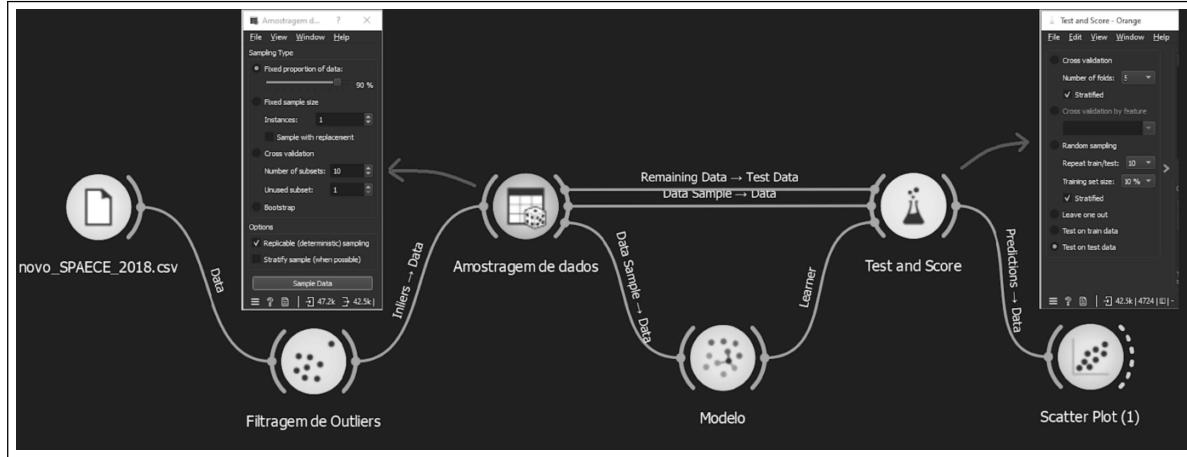
- **Regressão:** Para problemas de regressão, essas técnicas podem ser usadas para identificar as *features* mais importantes com base em sua contribuição para a precisão do modelo de regressão.
- **Classificação:** Para problemas de classificação, essas técnicas podem ser usadas para selecionar as *features* mais discriminativas com base em sua capacidade de separar as classes no conjunto de dados.

1.1.4 Dados de Treino

Os dados de Teste e Treino foram distribuídos de acordo com o Diagrama da Figura 10, novamente montado com o auxílio da ferramenta *Orange Datamining*. Onde o conjunto de dados foi dividido na proporção 10% para testes e 90% para treino, aplicando *data sampling*, que é o processo de selecionar uma parte dos dados de um conjunto maior de dados para realizar análises. A amostragem aleatória é o método pelo qual cada observação do conjunto de dados tem a mesma probabilidade de ser selecionada. Não há

qualquer regra ou padrão específico na escolha dos dados; eles são escolhidos de maneira aleatória. O procedimento é mesmo para qualquer conjunto de dados.

Figura 10 – Diagrama de Distribuição de Dados de Treino e Teste com Orange Data Mining



Fonte: Elaborado pelo Autor

A opção "*Deterministic*" refere-se à consistência na amostragem. Quando a amostragem é determinística, ela produz o mesmo subconjunto de dados todas as vezes que o procedimento é executado, dada uma mesma semente de aleatoriedade. Isso é útil para garantir a reproduzibilidade dos resultados. Com isso podemos criar diferentes *datasets* com diferentes proporções de dados para testes e treinos de modelos.

1.1.4.1 Tipos de Codificações Aplicadas

Ao aplicar modelos de Machine Learning (ML) a problemas de classificação, regressão e associação, é essencial entender como lidar com diferentes tipos de dados: categóricos, textuais e numéricos. Cada tipo de dado requer uma abordagem específica para pré-processamento e codificação. No conjunto de dados aplicado foram encontrados 2 tipos de dados: categóricos (algumas respostas textuais dos questionários) e numéricos

(notas, pontuações de proficiência e algumas respostas numéricas dos questionários).

Então foram aplicadas as seguintes codificações:

- **Dados Categóricos:** dados categóricos são valores discretos que representam categorias ou classes. Exemplos incluem cores ("vermelho", "verde", "azul"), estados civis ("solteiro", "casado", "divorciado"), e tipos de produtos.

1. Label Encoding:

- Descrição: Converte cada categoria em um número inteiro único.
- Aplicação: Simples de implementar e pode ser usado quando as categorias têm uma ordem implícita.
- Uso: Frequentemente usado em árvores de decisão e florestas aleatórias.

2. One-Hot Encoding:

- Descrição: Cria uma nova coluna binária (0 ou 1) para cada categoria.
- Aplicação: Evita atribuir ordens às categorias. É especialmente útil quando não há hierarquia entre as categorias.
- Uso: Usado em regressão linear, redes neurais e qualquer modelo que não interprete bem valores ordenados.

- **Dados Numéricos:** dados numéricos são valores quantitativos que podem ser contínuos ou discretos. Exemplos incluem idade, salário, e temperatura.

1. Discretização (Binning):

- Descrição: Converte dados contínuos em categorias discretas.
- Aplicação: Pode ser usado para transformar variáveis contínuas em categóricas.
- Uso: Usado em árvores de decisão e quando a relação entre os valores numéricos e a target variável é não-linear.

• Aplicações para Problemas de ML

– Classificação:

- * Dados Categóricos: One-Hot Encoding, Target Encoding.

- * Dados Textuais: TF-IDF, Word Embeddings.
- * Dados Numéricos: Normalização, Padronização.

- Regressão:

- * Dados Categóricos: One-Hot Encoding, Target Encoding.
- * Dados Textuais: TF-IDF, Word Embeddings.
- * Dados Numéricos: Padronização, Discretização.

Na Figura 11 podemos observar um exemplo de codificação *Label Encoding* usando o *Widget Edit Domain* do *Orange Data Mining* e na Figura 4 pode-se observar o exemplo de codificação usando o mapeamento com *datagramas* do *Python Pandas*.

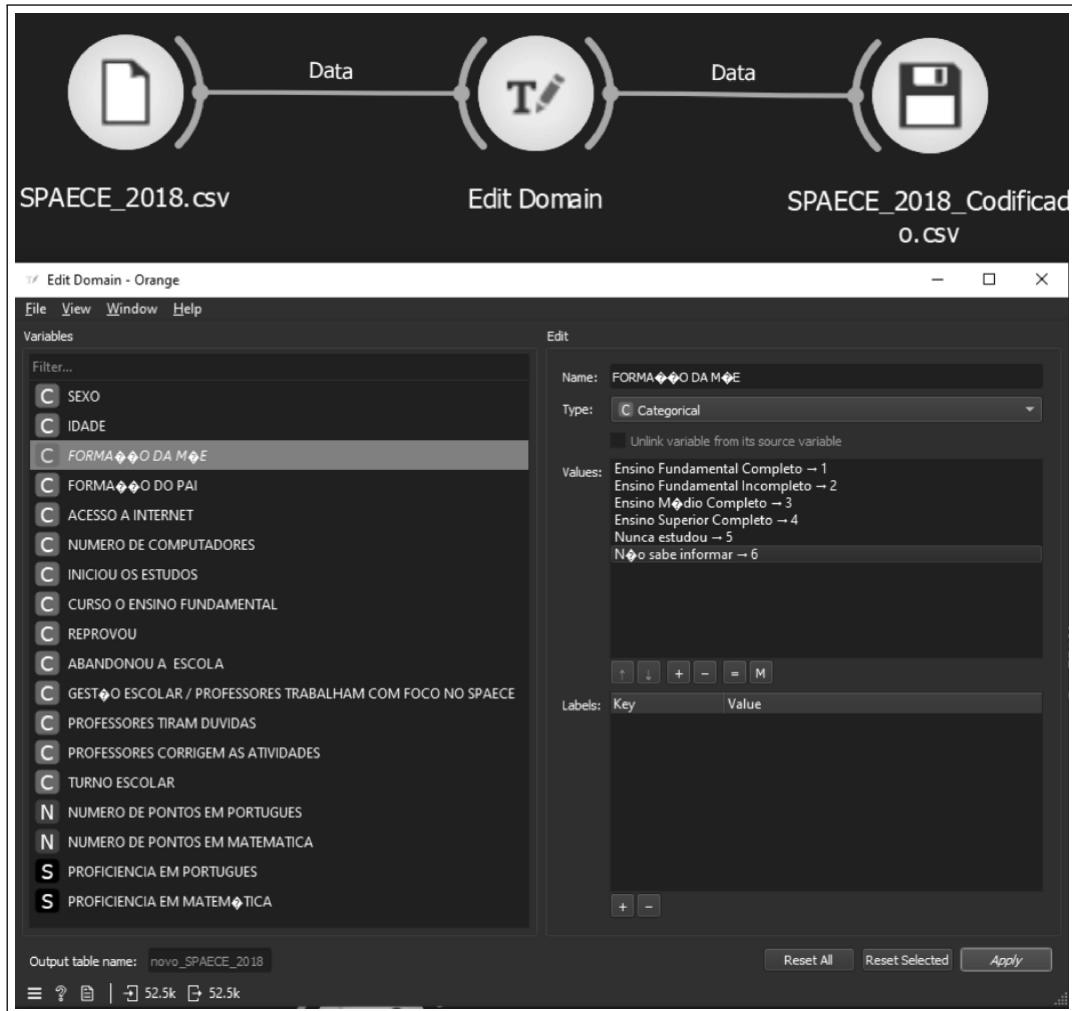
Com isso, a partir do *dataset* “novo SPAECE 2018.csv” processado foram criadas variações com diferentes codificações para diferentes aplicações. Sendo alguns deles:

- “**SPAECE_2018_codificado.csv**”: *dataset* codificado em *Label Encoding* e com os valores numéricos das notas e proficiências em formato *float*. Aplicável em problemas de Regressão.
- “**CLASS.csv**”: *dataset* codificado em *Label Encoding* com os valores das notas e proficiências distribuídas em 5 grupos ou classes (sendo 1 = pontuação mínima / 5 = pontuação máxima). Aplicável em problemas de Classificação.
- “**Apriori.csv**”: *dataset* com colunas codificadas em *One-Hot Encoding* (binários *bool*) para aplicar em algoritmos de associação Apriori.
- “**PCY_binário.csv**”: é o *dataset* "Apriori.csv" no formato *One-Hot Encoding* numérico (0-1) para aplicar em algoritmos de Associação PCY, FP-Growth e ECLAT.

1.1.5 Modelos de Aprendizagem de Máquina

O conjunto de dados SPAECE contém informações variadas sobre os alunos, incluindo características demográficas, socioeconômicas e acadêmicas. Esta etapa do

Figura 11 – Figura 3: Diagrama e Widget de Edição de Domínios do *Orange Data Mining*



Fonte: Elaborado pelo Autor

projeto testa como o conjunto de dados SPAECE pode ser utilizado para resolver diversos problemas de machine learning, desde previsão de valores contínuos (regressão), classificação de categorias e descoberta de padrões (associação). As etapas anteriores de preparação de dados, seleção de *features*, modelagem e avaliação são cruciais para obter modelos eficazes (KIM *et al.*, 2022; GARCIA-GUTIERREZ *et al.*, 2014; MODARESI; ARAGHINEJAD; EBRAHIMI, 2018). Exemplos de abordagens:

Figura 12 – Figura 4: Exemplo de Código para Mapeamento de Categorias com Python e Pandas

```
# Dicionários de mapeamento
formacao_mae_mapping = {
    'Ensino Fundamental Completo': 3,
    'Ensino Fundamental Incompleto': 2,
    'Ensino Médio Completo': 4,
    'Ensino Superior Completo': 5,
    'Nunca estudou': 0,
    'Não sabe informar': 1
}
```

Fonte: Elaborado pelo Autor

- **Regressão:** Prever a proficiência em matemática com base em características demográficas e socioeconômicas dos alunos.
- **Classificação:** Prever se um aluno irá abandonar a escola (sim ou não) com base em suas características e histórico acadêmico.

1.1.5.1 Modelos comparados

- **Machine Learning - Regressão apenas:**
 1. **Regressão Linear:** Modelo simples e interpretável que assume uma relação linear entre as *features* e a *target*. Funciona bem quando há uma relação linear ou quase linear entre as variáveis.
- **Machine Learning - Regressão e Classificação:**
 1. **Random Forests:** Poderoso método de aprendizagem de máquina, amplamente utilizado para problemas de classificação e regressão. Combinam a ideia de "*bagging*" (*bootstrap aggregating*) com o uso de árvores de decisão para criar um modelo robusto e eficaz. São menos propensas a *overfitting* do que árvores de decisão individuais, devido ao processo de *bagging* e à diversidade introduzida pela seleção aleatória de amostras e *features*. Elas

podem lidar bem com datasets com classes desbalanceadas, pois a votação majoritária ajuda a suavizar previsões extremas.

2. **Árvores de Decisão (Decision Trees):** Divide os dados em subconjuntos com base em valores de *features*, criando uma estrutura de árvore. Capaz de capturar relações não lineares e interações entre *features*. Fácil de interpretar e visualizar, não requer muita preparação de dados, pode capturar relações não lineares. Pode se tornar complexa e sobre ajustada facilmente, especialmente com muitos dados.
3. **K-Nearest Neighbors Regression (KNN):** Prediz o valor de um ponto com base nos valores dos k pontos mais próximos. Simples e eficaz para pequenos conjuntos de dados com relações não lineares. ão escala bem com grandes *datasets*, sensível à escolha dos *hiperparâmetros*. Computacionalmente intensivo para grandes *datasets*, sensível ao ruído nos dados.
4. **Support Vector Machine (SVM):** Utilizando um conjunto de funções de base para criar um hiperplano de regressão. Bom para problemas de alta dimensionalidade e quando a relação entre as variáveis não é linear. Não escala bem com grandes *datasets*, sensível à escolha dos *hiperparâmetros*.

- **Machine Learning - Classificação Apenas:**

1. **Naive Bayes:** Calcula a probabilidade de uma instância pertencer a uma determinada classe com base na probabilidade condicional das *features*. A principal suposição é que as *features* são independentes entre si, o que significa que a presença de uma *feature* não afeta a presença de outra. Simples e rápido de treinar, eficaz para grandes *datasets*, funciona bem com dados categóricos. Assume independência entre as *features*, o que pode não ser verdadeiro.

- **Deep Learning - Classificação e Regressão:**

1. **Multilayer Perceptron (MLP):** Rede neural *feedforward* com uma ou mais

camadas ocultas. Utiliza *backpropagation* para ajustar os pesos durante o treinamento. Capaz de capturar relações complexas e não lineares. Funciona bem com grandes conjuntos de dados e é altamente flexível devido ao ajuste dos *hiperparâmetros*, como número de camadas e neurônios por camada. Requer mais dados e poder computacional, mais difícil de interpretar e ajustar, sensível à escolha dos *hiperparâmetros* e à arquitetura da rede. MLP com múltiplas camadas ocultas é considerado uma rede neural "deep"(profunda), entrando na categoria de deep learning. Redes profundas (com mais camadas) têm a capacidade de aprender representações hierárquicas dos dados, onde camadas iniciais podem aprender características básicas e camadas mais profundas podem aprender combinações mais abstratas e complexas dessas características.

1.1.5.2 Problemas de Regressão

Problemas de regressão envolvem prever um valor contínuo baseado em uma ou mais variáveis independentes. Ao contrário dos problemas de classificação, onde a saída é uma classe ou categoria, em problemas de regressão, a saída é um valor numérico. Modelos de regressão são usados em diversas áreas, incluindo economia, finanças, ciências naturais, engenharia e muitas outras, para prever valores como preços, temperaturas, pontuações, etc (FUMO; BISWAS, 2015; ALIZAMIR *et al.*, 2020). No contexto do *dataset* do SPAECE 2018, um exemplo de problema de regressão seria prever a "PROFICIENCIA EM MATEMATICA"dos estudantes com base em várias características como sexo, idade, formação dos pais, acesso à internet, e outras variáveis presentes no *dataset*.

1.1.5.2.1 Métricas para Avaliação de Modelos de Regressão

- **Erro Médio Absoluto (Mean Absolute Error - MAE):** O MAE é a média da diferença absoluta entre os valores previstos pelo modelo e os valores observados. É calculado pela fórmula:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Onde y_i são os valores observados, \hat{y} são os valores previstos pelo modelo e n é o número total de observações. O MAE mede a magnitude média dos erros em uma escala similar aos dados originais, sendo menos sensível a *outliers* comparado ao MSE.

- **Erro Quadrático Médio (Mean Squared Error - MSE):** O MSE é a média dos quadrados dos erros entre os valores previstos e os valores observados. É calculado pela fórmula:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

O MSE penaliza erros grandes mais do que o MAE, pois eleva os erros ao quadrado. É frequentemente usado em problemas onde erros maiores são críticos.

- **Raiz do Erro Quadrático Médio (Root Mean Squared Error - RMSE):** O RMSE é a raiz quadrada do MSE e é uma das métricas mais comuns para avaliação de modelos de regressão. É calculado pela fórmula:

$$RMSE = \sqrt{MSE}$$

O RMSE fornece uma interpretação na mesma escala das variáveis dependentes, o que facilita a compreensão do quanto bem o modelo está performando.

- **Coeficiente de Determinação (R^2):** O R^2 é uma medida estatística que indica a proporção da variância dos dados que é explicada pelo modelo. É uma medida de

quão bem os pontos de dados se ajustam à linha de regressão ajustada. Valores de R_2 variam de 0 a 1, sendo 1 indicativo de um ajuste perfeito.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Onde \bar{y} é a média dos valores observados y_i . Um valor de R_2 próximo a 1 indica que o modelo explica bem a variabilidade dos dados, enquanto valores próximos a 0 indicam que o modelo não explica bem a variabilidade dos dados.

- **Erro Absoluto Percentual Médio (Mean Absolute Percentage Error - MAPE):**

O MAPE é uma métrica útil para problemas onde a magnitude do erro é significativa em relação ao valor real. É calculado pela fórmula:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

O MAPE expressa o erro médio como uma porcentagem do valor real, sendo útil em contextos onde a escala dos dados é importante.

(FREUND; WILSON; SA, 2006)

1.1.5.3 Problemas de Classificação

Problemas de classificação envolvem a previsão de categorias ou rótulos para novas observações com base em um conjunto de dados de treinamento. Ao contrário dos problemas de regressão, que preveem valores contínuos, os problemas de classificação preveem rótulos discretos (NOVAKOVIĆ *et al.*, 2017). No contexto do *dataset* do SPAECE 2018, podemos formular problemas de classificação para predizer uma categoria baseada em outras *features*. Por exemplo:

- **Previsão de Performance Acadêmica:** Classificar os alunos em categorias de desempenho com base em suas características e respostas ao questionário.
- **Previsão de Abandono Escolar:** Classificar se um aluno tem probabilidade de abandonar a escola com base nas características demográficas e educacionais.

1.1.5.3.1 Métricas para Avaliação de Modelos de Classificação

Ao avaliar modelos de classificação, é importante escolher métricas de desempenho apropriadas para entender como o modelo está se saindo em diferentes aspectos. As métricas e métodos de avaliação que iremos aplicar serão:

- **Acurácia (Accuracy):** A acurácia é a proporção de exemplos classificados corretamente pelo modelo em relação ao total de exemplos. Sendo uma métrica geral que indica a proporção de previsões corretas do modelo. Calculada por:

$$(TP + TN) / (TP + TN + FP + FN)$$

- **Precisão (Precision):** A precisão é a proporção de verdadeiros positivos (TP) em relação a todos os exemplos classificados como positivos pelo modelo (verdadeiros positivos mais falsos positivos). É calculada pela fórmula:

$$TP / (TP + FP)$$

- **Revocação (Recall ou Sensibilidade):** A revocação é a proporção de verdadeiros positivos (TP) em relação a todos os exemplos que são realmente positivos (verdadeiros positivos mais falsos negativos). A revocação mede a capacidade do modelo em identificar corretamente exemplos positivos. É calculada pela fórmula:

$$TP / (TP + FN)$$

- **O F1-Score:** É a média harmônica da precisão e da revocação e fornece um único número que representa o balanceamento entre essas duas métricas. Sendo útil quando há desequilíbrio entre as classes. É calculado pela fórmula:

$$F1 = 2 \times \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}}$$

- **Especificidade:** A especificidade mede a proporção de verdadeiros negativos em relação ao total de reais negativos. Essencial para entender como o modelo se comporta em termos de detecção de classes positivas e negativas. Importante quando diferentes classes têm diferentes custos de erros.
- **Matriz de Confusão:** A matriz de confusão é uma tabela que mostra a frequência de classificações corretas e incorretas feitas pelo modelo. É uma ferramenta fundamental para entender o desempenho do modelo em cada classe.
- **Intervalo de Confiança:** Um intervalo de confiança fornece uma estimativa do intervalo no qual a verdadeira métrica de desempenho do modelo se encontra com um certo nível de confiança (por exemplo, 95%). Importante para entender a variabilidade e a robustez das métricas de desempenho. Indica a precisão da estimativa da métrica de desempenho.
- **Kappa:** O coeficiente Kappa mede a concordância entre as previsões do modelo e as classes reais, ajustando para a concordância ao acaso. Útil em conjuntos de dados desbalanceados, proporcionando uma medida ajustada de acurácia que considera a ocorrência ao acaso. Relevante para este projeto ao comparar diferentes modelos.
- **Valores Preditivos Positivos e Negativos:** O Valor Preditivo Positivo (VPP) mede a proporção de verdadeiros positivos em relação ao total de previsões positivas. Valor Preditivo Negativo (VPN) mede a proporção de verdadeiros negativos em relação ao total de previsões negativas. Úteis para avaliar a confiabilidade das previsões positivas e negativas do modelo. Relevante para garantir que as previsões de desempenho dos estudantes sejam precisas.
- **Prevalência** A prevalência é a proporção de instâncias de uma determinada classe

no conjunto de dados. Importante para entender a distribuição das classes no conjunto de dados. Ajuda a interpretar outras métricas, especialmente em conjuntos de dados desbalanceados.

- **Taxa de Detecção:** A taxa de detecção é equivalente à sensibilidade, medindo a capacidade do modelo de identificar corretamente os verdadeiros positivos. Específica para casos em que é crucial detectar todos os verdadeiros positivos, como em problemas de saúde ou segurança. Relevante para entender a capacidade do modelo de identificar estudantes com alta proficiência.
- **Prevalência da Detecção:** Refere-se à proporção de verdadeiros positivos em relação ao total de instâncias detectadas. Similar ao valor preditivo positivo, útil para avaliar a precisão das detecções do modelo.
- **Precisão Balanceada** A precisão balanceada é a média aritmética da sensibilidade e especificidade, proporcionando uma métrica equilibrada para conjuntos de dados desbalanceados. Muito útil em cenários de classes desbalanceadas, fornecendo uma métrica que considera tanto a capacidade de detectar positivos quanto negativos.

..

2 RESULTADOS

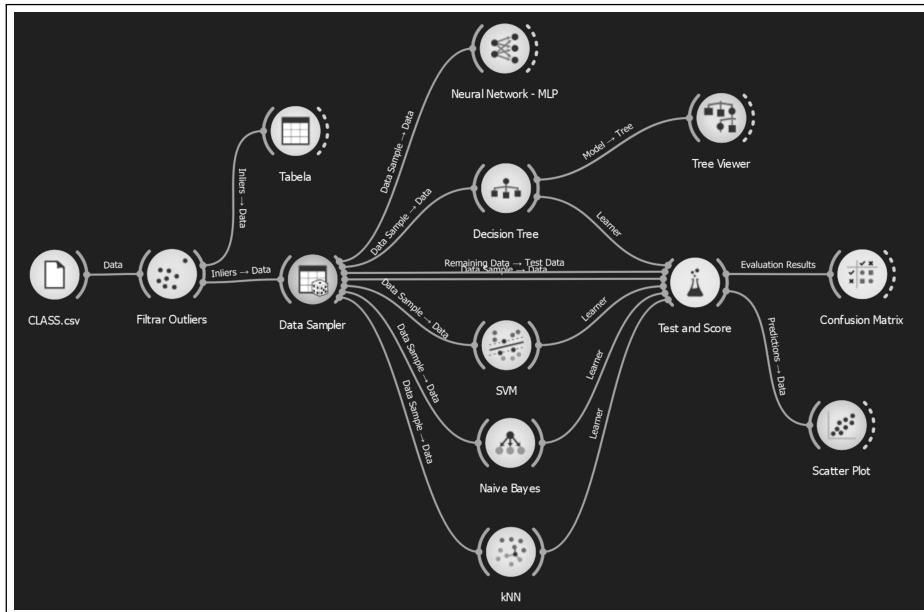
Este capítulo descreve os procedimentos experimentais conduzidos para avaliar os modelos de Aprendizado de Máquina escolhidos no contexto da aplicação do conjunto de dados processado.

2.1 CONFIGURAÇÃO DOS EXPERIMENTOS

A seleção dos atributos foi orientada pelos atributos selecionados. O processo de escolha dos atributos envolveu a visualização e análise dos dados gerados nas fases precedentes, bem como a realização de múltiplos testes com os modelos selecionados, com o objetivo de maximizar a acurácia das previsões. O *SelectFromModel* é uma técnica de seleção de características (*feature selection*) em aprendizado de máquina do *Python sklearn* que utiliza modelos baseados em árvores, como árvores de decisão ou florestas aleatórias, para identificar e selecionar as características mais importantes do conjunto de dados. A ideia básica por trás do *SelectFromModel* é treinar um modelo no conjunto de dados original e, em seguida, utilizar a importância das características atribuídas pelo modelo para selecionar um subconjunto das características mais relevantes. Isso é particularmente útil quando você lida com conjuntos de dados que contêm muitas características e deseja reduzir a dimensionalidade para melhorar o desempenho do modelo ou simplificar a interpretação. O treinamento e teste dos modelos foram conduzidos tanto utilizando o ambiente de programação nativo em *Python* quanto empregando a ferramenta *Orange Datamining* para problemas de classificação e regressão. Para mineração de regras de Associação foi-se aplicado a biblioteca *mlxtend*, que possui ferramentas voltadas para este tipo de algoritmo. Ambas as abordagens utilizaram as bibliotecas disponíveis no ecossistema *Python*, notadamente o *Scikit-Learn*. Este processo metodológico proporcionou uma avaliação abrangente e comparativa da performance dos modelos de Aprendizado de Máquina selecionados, oferecendo informações sobre

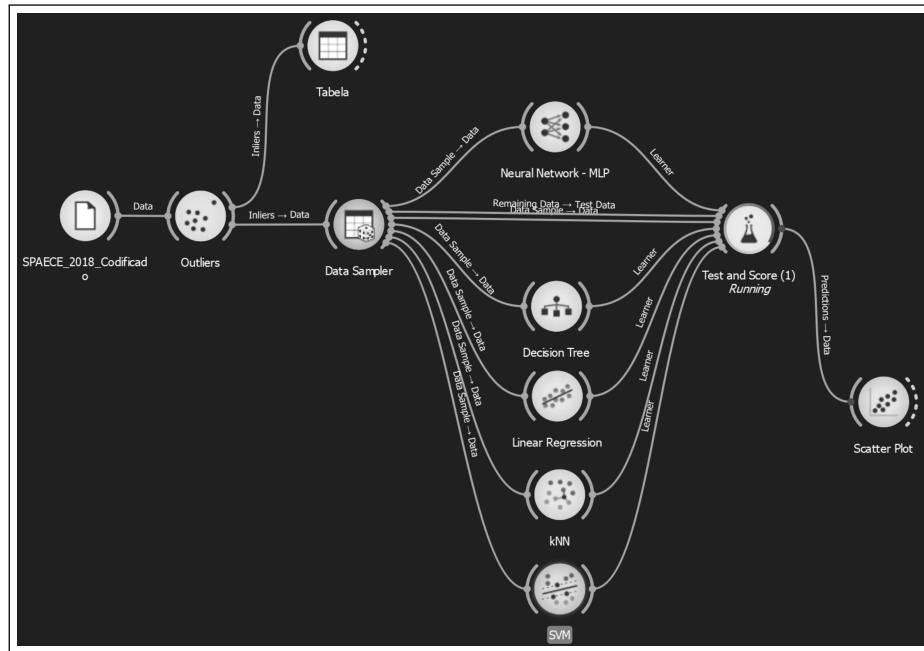
a eficácia desses modelos ao trabalhar com os dados da SPAECE. Foram realizados experimentos utilizando diferentes conjuntos de dados, sendo estes variações dos dados originais, mas com codificações e ajustes para melhor se adequarem à cada abordagem e aplicação nos modelos. Sendo executados em um computador doméstico equipado com um processador *Ryzen 7 5800X* de 8 núcleos *4.5ghz (gigahertz)* e 32gb (*gigabytes*) de memória RAM (*Random Access Memory*). O conjunto de dados de maneira geral possui 2 partes: uma contendo dados categóricos provenientes de testes socioeconômicos e outra dos resultados dos exames de Matemática e Português. Desta forma é possível direcionar os testes dos modelos para que os mesmos aprendam padrões de dados específicos de cada um dos dois tipos.

Figura 13 – Diagrama na Ferramenta *Orange Data Mining* para problemas de Classificação.



Fonte: Elaborado pelo Autor

Figura 14 – Diagrama na Ferramenta *Orange Data Mining* para problemas de Regressão.



Fonte: Elaborado pelo Autor

2.1.1 Objetivos dos Experimentos

Os resultados do SPAECE 2018 apontam para a necessidade de intervenções mais robustas e direcionadas no ensino médio, especialmente nas áreas de Língua Portuguesa e Matemática. A presente pesquisa tem como objetivo explorar e comparar a eficácia de modelos gerados a partir de algoritmos de Aprendizagem de Máquina (Machine Learning) e Aprendizagem Profunda (Deep Learning) na previsão das proficiências de aprendizagem dos estudantes no Sistema Permanente de Avaliação da Educação Básica do Ceará (SPAECE). No intuito de:

- Determinar a eficácia relativa dos modelos de Aprendizagem de Máquina e Aprendizagem Profunda na previsão das proficiências de aprendizagem dos estudantes.
- Identificar se os modelos de Aprendizagem Profunda superam os modelos de

Aprendizagem de Máquina em termos de eficiência e precisão.

- Estabelecer um conjunto de atributos que mais influenciam o desempenho dos estudantes, fornecendo dados importantes que possibilitem intervenções educacionais.

Em suma, a aplicação de técnicas avançadas de análise de dados e inteligência artificial no contexto educacional pode proporcionar dados que proporcionem a melhoria do sistema de ensino. As intervenções sugeridas, baseadas nos resultados do SPAECE 2018, visam criar um ambiente educacional mais inclusivo, eficaz e adaptado às necessidades dos alunos, contribuindo assim para a formação de uma geração mais bem preparada e capaz de enfrentar os desafios futuros.

2.1.1.1 Aplicando o Fluxo da Metodologia

1. Coleta e Preparação dos Dados: Os dados do SPAECE coletados são limpos e preparados para análise. Esta etapa inclui a seleção de *features* relevantes e a transformação de variáveis categóricas em numéricas, se necessário.
2. Desenvolvimento de Modelos: Serão desenvolvidos diversos modelos de Aprendizagem de Máquina (como *Decision Trees*, *Random Forests*, *Support Vector Machines*, e *K-Nearest Neighbors*) e Aprendizagem Profunda (como Redes Neurais Profundas).
3. Análise de Atributos Influentes: Serão utilizados métodos de seleção de *features* e análise de importância de *features* para identificar quais atributos têm maior influência na previsão do desempenho dos estudantes.
4. Avaliação de Modelos: A eficácia dos modelos será avaliada utilizando métricas como Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) e o Coeficiente de Determinação (R^2) para problemas de regressão. Para problemas de classificação, serão utilizadas métricas como Accuracy, Precision, Recall, F1-Score, e Matriz de Confusão.

5. Comparação entre Modelos: Os resultados dos modelos de Aprendizagem de Máquina e Aprendizagem Profunda serão comparados para determinar quais apresentam melhor desempenho na previsão das proficiências dos estudantes.

2.1.1.2 Parâmetros dos Modelos

Os parâmetros dos modelos foram os seguintes:

1. SVM:

- Custo: prazo de penalidade por perda e aplica-se a tarefas de classificação e regressão. Ajustado para $C = 1$.
- ε : um parâmetro do modelo epsilon-SVR, aplica-se a tarefas de regressão. Define a distância dos valores verdadeiros dentro da qual nenhuma penalidade está associada aos valores previstos. Ajustado para $\varepsilon = 0,1$.
- Número limite de Iterações: Define o número máximo de iterações permitidas. Ajustado para $max = 10000$.

2. KNN:

- Número de Vizinhos: Define o número de vizinhos mais próximos. Ajustado para $n = 5$.
- Métrica: Ajustada para *Manhatam*, ou seja, soma das diferenças absolutas de todos os atributos.
- Pesos: Ajustada para Uniforme, ou seja, todos os pontos em cada vizinhança têm peso igual.

3. Decision Trees:

- Induzir árvore binária: construir uma árvore binária (dividida em dois nós filhos).
- Min. número de instâncias nas folhas: o algoritmo nunca construirá uma divisão que colocaria menos do que o número especificado de exemplos de treinamento em qualquer uma das ramificações. Ajustado para $min = 2$.

- Não dividir subconjuntos menores que: proíbe o algoritmo de dividir os nós com menos que o número de instâncias determinado. Ajustado para $min = 5$.
- Limitar a profundidade máxima da árvore: limita a profundidade da árvore de classificação ao número especificado de níveis de nó. Ajustado para $deep = 100$.

4. Random Forests:

- Número de Árvores: Este parâmetro indica que o modelo de *Random Forest* consiste em 10 árvores de decisão. Um número maior de árvores geralmente melhora a performance do modelo, pois ele pode capturar mais variabilidade nos dados. No entanto, mais árvores também aumentam o tempo de treinamento e o consumo de memória. Parâmetro ajustado para 10.
- Número máximo de atributos considerados: Este parâmetro sugere que todas as *features* disponíveis serão consideradas para a divisão em cada nó da árvore. Isso pode resultar em árvores mais complexas e potencialmente *overfitting*, pois não há restrição sobre quantas *features* são consideradas em cada divisão. Ajustado para "Ilimitado"
- Profundidade máxima da árvore: A profundidade máxima das árvores definida como ilimitada, permitindo que as árvores cresçam até que cada folha contenha um número mínimo de amostras.

5. MLP:

- Neurônios por camada oculta: definido como o i -ésimo elemento representa o número de neurônios na i -ésima camada oculta. Por exemplo. Ajustado para 20, 20, 20, ou seja, três camadas ocultas de 20 neurônios.
- Função de ativação da camada oculta: Ajustada para *ReLU*, a função da unidade linear retificada
- Solucionador para otimização de peso: Ajustada para *Adam*, otimizador estocástico baseado em gradientes.

- Max iterações: número máximo de iterações. Ajustado para $max = 1000$

6. Regressão Linear:

- L2 Regularization (Ridge): Adiciona a soma dos quadrados dos valores dos coeficientes à função de custo. Isso penaliza os coeficientes maiores mais fortemente do que os menores, forçando-os a serem distribuídos de maneira mais uniforme e evitando que se tornem muito grandes.
- Força da Regularização: ajustado para $\alpha = 1$.

2.1.2 Seleção de Features

O método de seleção de *features* a Seleção de Features com Árvores de Decisão. Esta é uma abordagem poderosa e eficaz, especialmente quando se deseja entender quais variáveis têm maior impacto na predição de um determinado *target* ou variável de interesse. Durante o treinamento de uma árvore de decisão, o algoritmo calcula automaticamente a importância de cada *feature* para fazer previsões precisas. Essa importância é medida pela redução média da impureza (como Gini impurity ou entropia) que cada *feature* contribui em todas as divisões da árvore. É facilmente implementada com o *python sklearn*.

Durante os testes de avaliações e seleção das *features*, foram identificados casos em que uma *feature* teria muito mais importância (score) que as demais. Isso pode indicar algumas situações que precisam ser avaliadas antes de decidir se deve ou não ser excluída do treinamento do modelo:

- **Impacto real da *feature*:** Primeiramente, é importante entender por que uma *feature* específica tem uma importância tão alta. Pode ser que essa *feature* realmente seja muito relevante para o problema em questão e seu alto score de importância seja um reflexo legítimo do seu papel na predição.
- **Overfitting:** Se uma *feature* específica tem uma importância muito alta, isso pode ser um indicativo de que o modelo está sob reajustando-se a essa *feature*,

especialmente se outras *features* relevantes estão sendo negligenciadas. Nesse caso, é importante considerar ajustes nos hiper parâmetros do modelo para evitar *overfitting*, como limitar a profundidade da árvore ou o número mínimo de amostras por folha.

- **Correlação com o target:** A alta importância de uma *feature* pode indicar uma forte correlação com o *target*. Se essa correlação é genuína e relevante para o problema, então essa *feature* pode ser crucial para a precisão do modelo. No entanto, se essa correlação é espúria (por exemplo, devido a vazamento de dados), pode ser necessário investigar mais a fundo.

Estratégias aplicadas para lidar com *features* de importâncias desbalanceadas foram as seguintes:

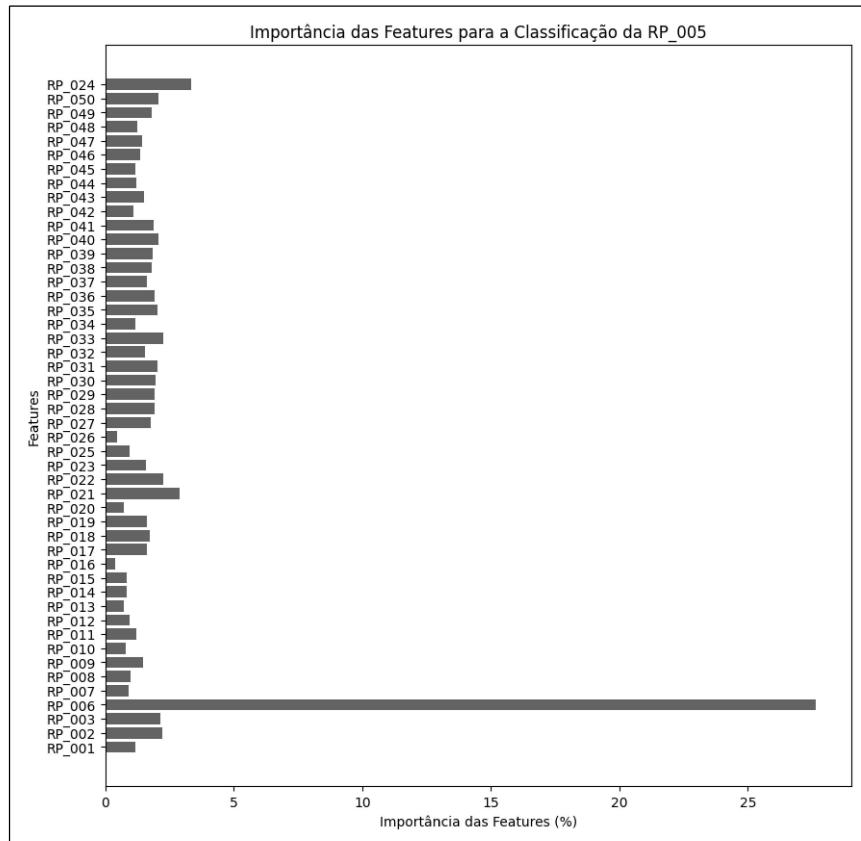
- **Seleção de um conjunto diversificado de features:** Se você identificar que apenas algumas poucas *features* têm importância significativa, pode ser benéfico considerar a seleção de um conjunto mais diversificado de *features*. Isso pode ser feito manualmente, escolhendo um conjunto de atributos que abranja diferentes aspectos do problema, ou utilizando técnicas automatizadas de seleção de *features* que considerem a importância relativa das *features*.
- **Reavaliação das features:** Periodicamente, é útil reavaliar as *features* utilizadas no modelo à medida que novos dados são disponibilizados ou conforme se ganha mais dados sobre o problema. Isso pode envolver a exclusão de *features* que se mostram irrelevantes ou redundantes e a inclusão de novas *features* que são mais informativas.

2.1.2.1 Resultados da Seleção de Melhores Atributos

Os resultados obtidos dos testes com *features* apresentados a seguir podem ser encontrados dos gráficos das Figuras 16 a 29

Features aplicadas:

Figura 15 – Exemplo do Resultado de Seleção de Features para a RP_005



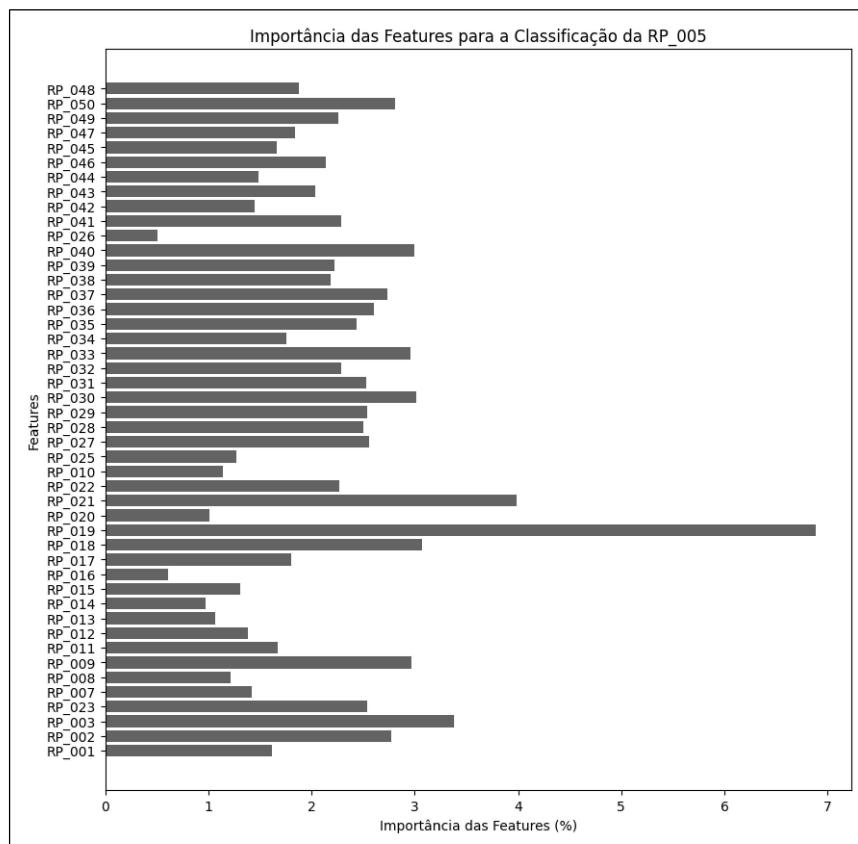
Fonte: Elaborado pelo Autor

- "RP_005": "Até que série/ano sua mãe ou a responsável por você estudou?"
- "RP_006": "Até que série/ano seu pai ou o responsável por você estudou?"
- "RP_010": "Você tem acesso à internet na sua casa?"
- "RP_023": "Quando você frequentou a escola pela primeira vez?"
- "RP_024": "A partir do 1º ano do ensino fundamental, em que tipo de escola você estudou?"
- "RP_025": "Você já foi reprovado?"
- "RP_026": "Você já abandonou a escola durante o período de aulas e ficou fora da escola o resto do ano?"
- "RP_041": "Meus professores utilizam diferentes recursos para apresentar a maté-

ria (vídeo, música, computador etc.)."

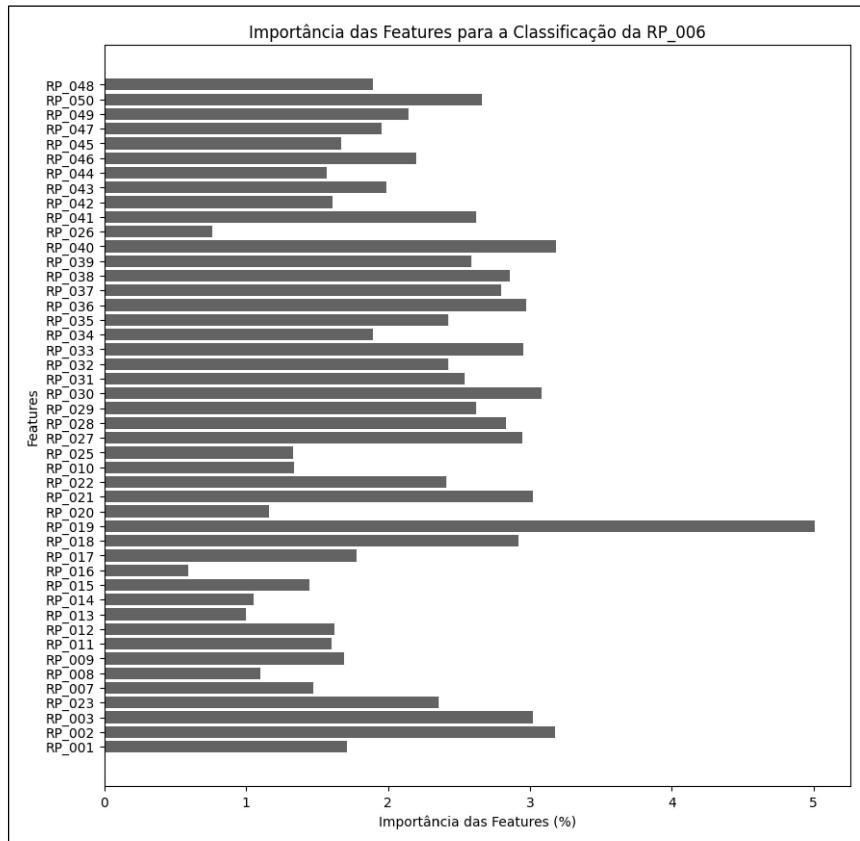
- "RP_042": "Meus professores respondem às dúvidas que aparecem durante as aulas."
- "RP_043": "Meus professores utilizam o livro didático."
- "RP_044": "Meus professores falam sobre a importância de estudar."
- "RP_045": "Meus professores revisam a matéria para as provas."
- "RP_047": "Meus professores corrigem os exercícios que passam em sala de aula com os estudantes."
- "RP_048": "Meus professores ajudam os estudantes com dificuldade."

Figura 16 – Seleção de Features para Regras de Classificação com a RP_005



Fonte: Elaborado pelo Autor

Figura 17 – Seleção de Features para Regras de Classificação com a RP_006

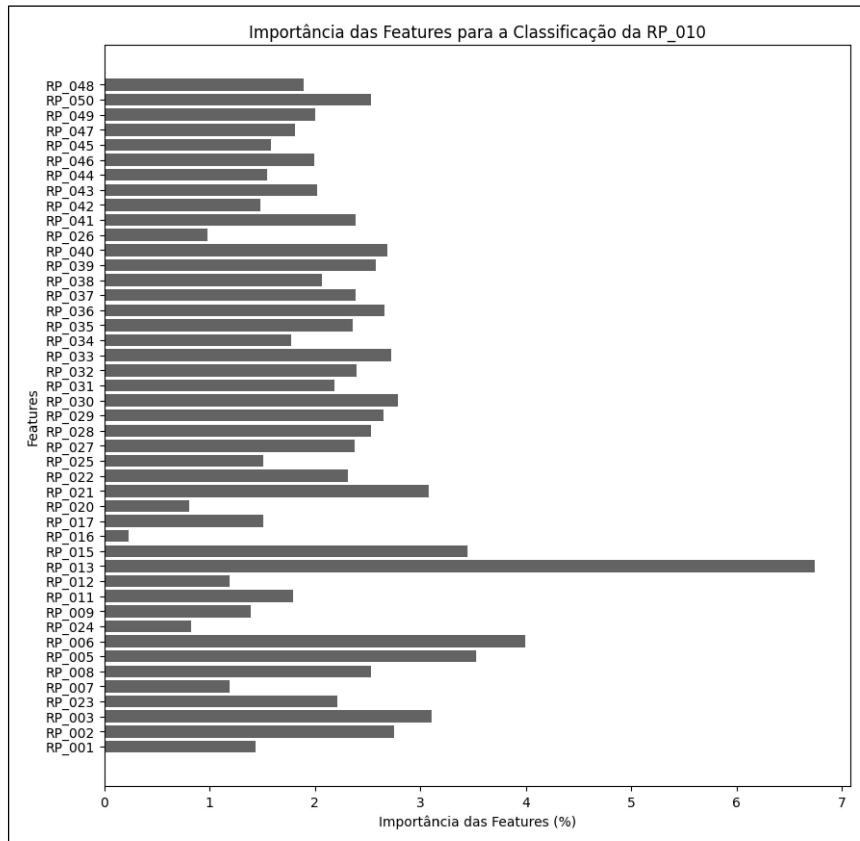


Fonte: Elaborado pelo Autor

2.1.3 Comparação dos Resultados Obtidos com Modelos

O Experimento foi conduzido utilizando a parte que continham dados referentes ao questionário socioeconômico para testes de classificação e o conjunto de dados contendo dados numéricos (pontuações dos testes) juntamente com os dados do questionário socioeconômico para testes de regressão, conforme as especificações e modelos previamente delineados na seção anterior. O procedimento experimental consistiu na realização de 10 ensaios distintos para cada modelo, cujo resultado final corresponde à média dos 10 resultados obtidos. Inicialmente, a abordagem adotada contemplou a alocação de 90% dos dados para a fase de treinamento, reservando os restantes 10% para

Figura 18 – Seleção de Features para Regras de Classificação com a RP_010



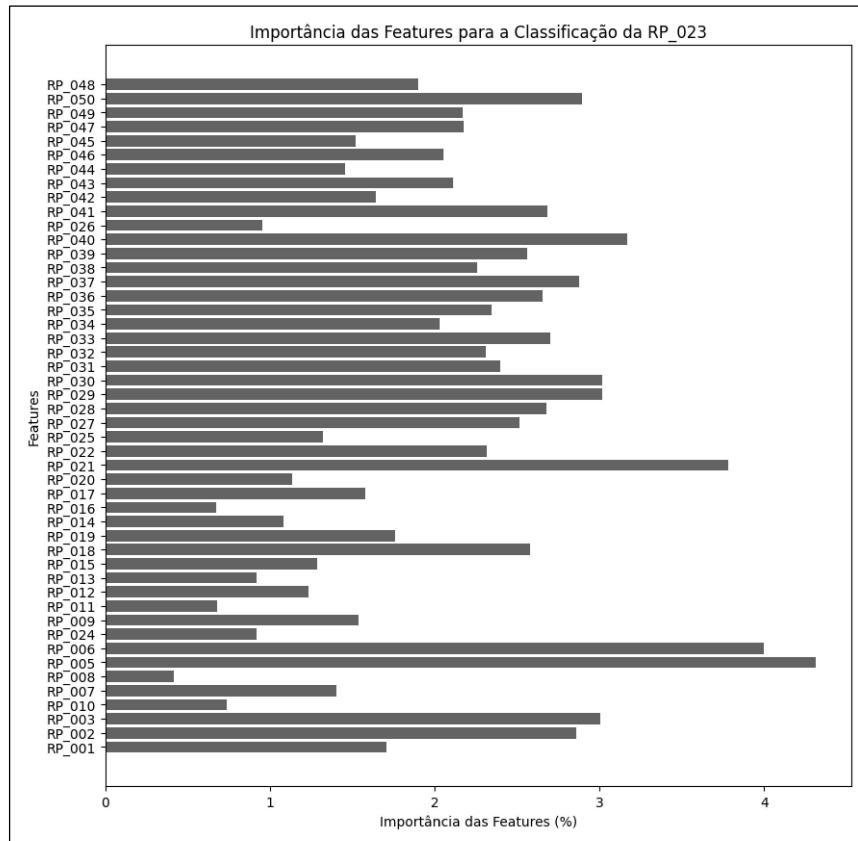
Fonte: Elaborado pelo Autor

a fase de teste. Tornando possível, por exemplo, a tentativa da classificação do aluno como estudante de turno integral e com média de pontuação em Matemática igual a 12 pontos. Diversas baterias de testes foram realizadas com o processo de seleção de atributos e treinamento de modelos.

2.1.3.1 Resultados dos Primeiros Testes

Começando pelos dados mais dispersos. Os testes com os atributos numéricos e regressão tiveram resultados abaixo do esperado, já que os modelos tiveram poucos atributos numéricos para trabalhar.

Figura 19 – Seleção de Features para Regras de Classificação com a RP_023



Fonte: Elaborado pelo Autor

Tabela 1 – Testes com Regressão em Predições de Proficiência em Matemática

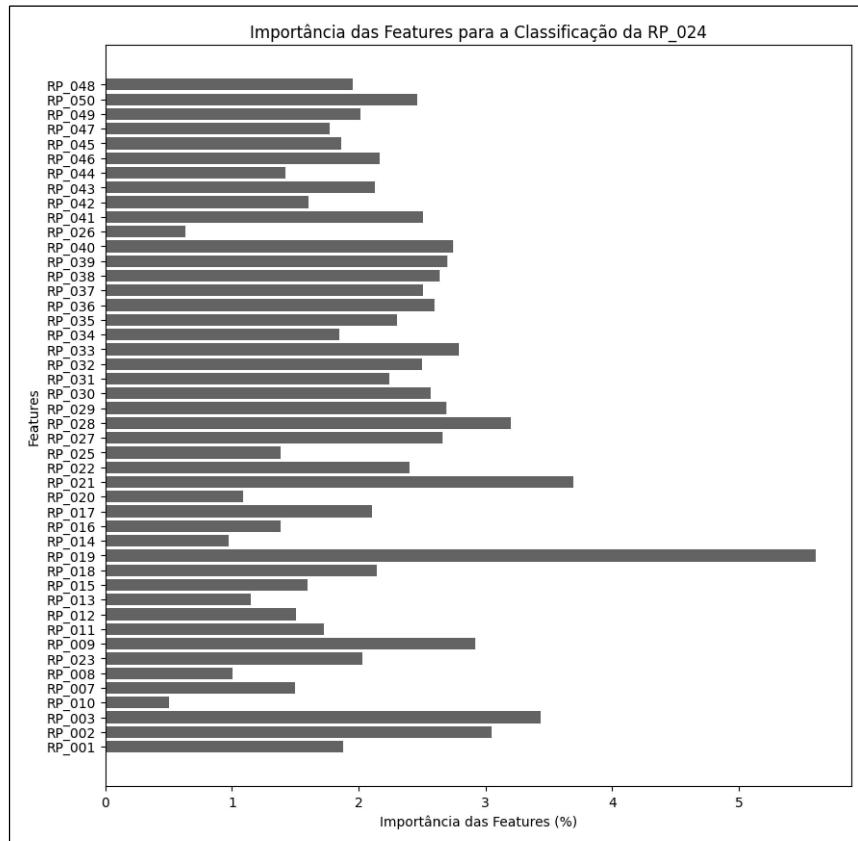
Modelos Regressão	MSE	RMSE	MAE	MAPE	R ²
Decision Trees	2069.2	45.49	35.4	0.131	0.252
Linear Regression	1794.9	42.4	33.5	0.124	0.351
SVM	2159.5	46.5	38.5	0.148	0.219
KNN	2025.2	45.0	35.2	.129	0.268
MLP	1705.1	41.3	32.42	0.121	0.383

Fonte: Elaborado pelo autor

• **Interpretando resultados da tabela 1:**

- A Decision Tree apresenta um MSE e RMSE relativamente altos, indicando maior erro quadrático médio. O R² é 0.252, o que significa que cerca de 25.2% da variabilidade é explicada pelo modelo, indicando um desempenho

Figura 20 – Seleção de Features para Regras de Classificação com a RP_024

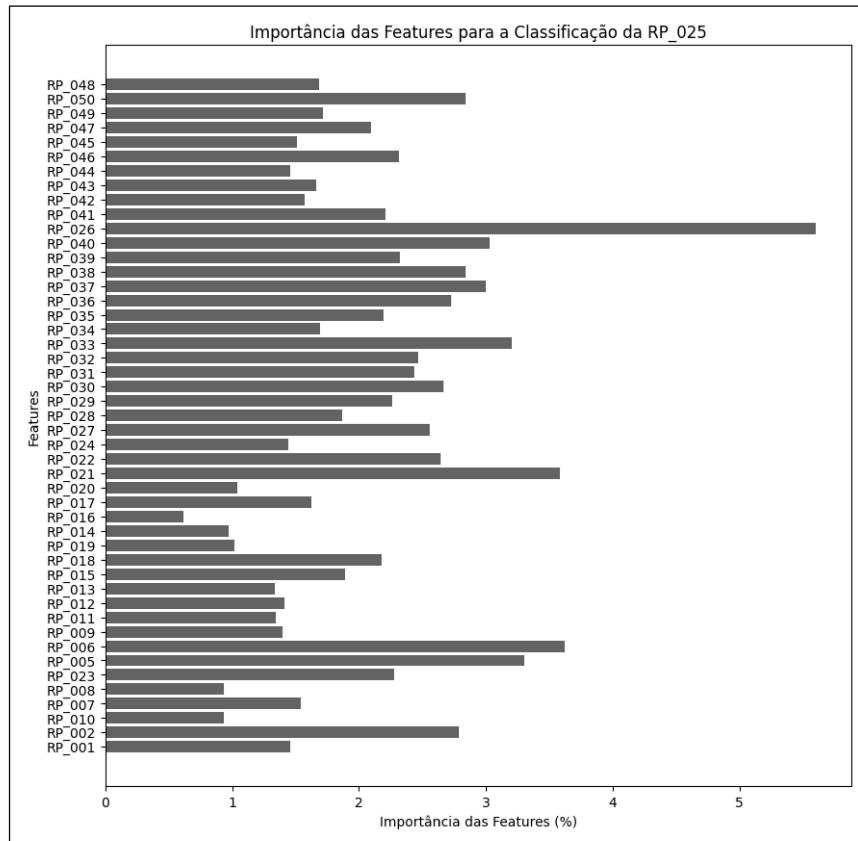


Fonte: Elaborado pelo Autor

moderado.

- A Regressão Linear apresenta um MSE menor que a Decision Tree, com um RMSE de 42.4, indicando melhor precisão. O R^2 de 0.351 indica que o modelo explica 35.1% da variabilidade, mostrando um desempenho superior à Decision Tree.
- O modelo SVM tem o maior MSE e RMSE entre todos os modelos testados, indicando maior erro quadrático médio. Com um R^2 de 0.219, é o modelo menos eficiente, explicando apenas 21.9% da variabilidade dos dados.
- O modelo KNN apresenta um desempenho um pouco melhor que a Decision Tree, com MSE e RMSE ligeiramente menores. O R^2 de 0.268 indica que o

Figura 21 – Seleção de Features para Regras de Classificação com a RP_025



Fonte: Elaborado pelo Autor

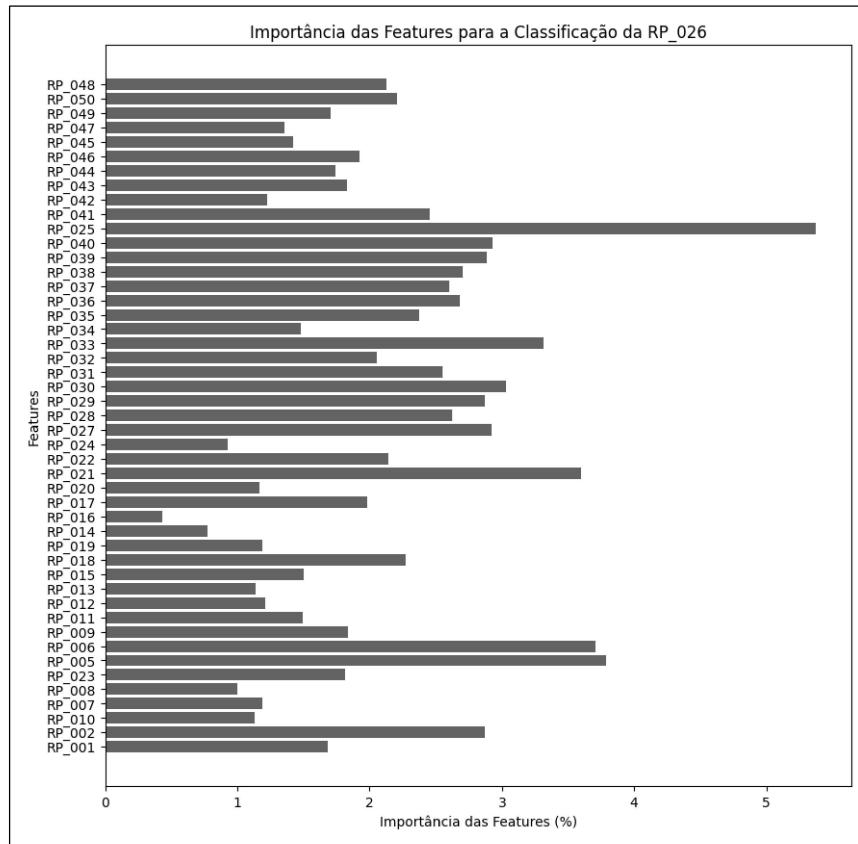
modelo explica 26.8% da variabilidade dos dados.

- O modelo MLP tem o menor MSE e RMSE, indicando a melhor precisão entre os modelos testados. Com um R^2 de 0.383, o MLP explica 38.3% da variabilidade dos dados, sendo o modelo mais eficiente.

• **Conclusões sobre a tabela 1:**

- MLP (Multi-Layer Perceptron) é o melhor modelo para a regressão de Proficiência em Matemática, com menor MSE, RMSE, MAE e MAPE, além do maior R^2 , indicando que ele explica melhor a variabilidade dos dados.
- Linear Regression também apresentou um bom desempenho, sendo uma opção viável, especialmente pela simplicidade e interpretabilidade.

Figura 22 – Seleção de Features para Regras de Classificação com a RP_026



Fonte: Elaborado pelo Autor

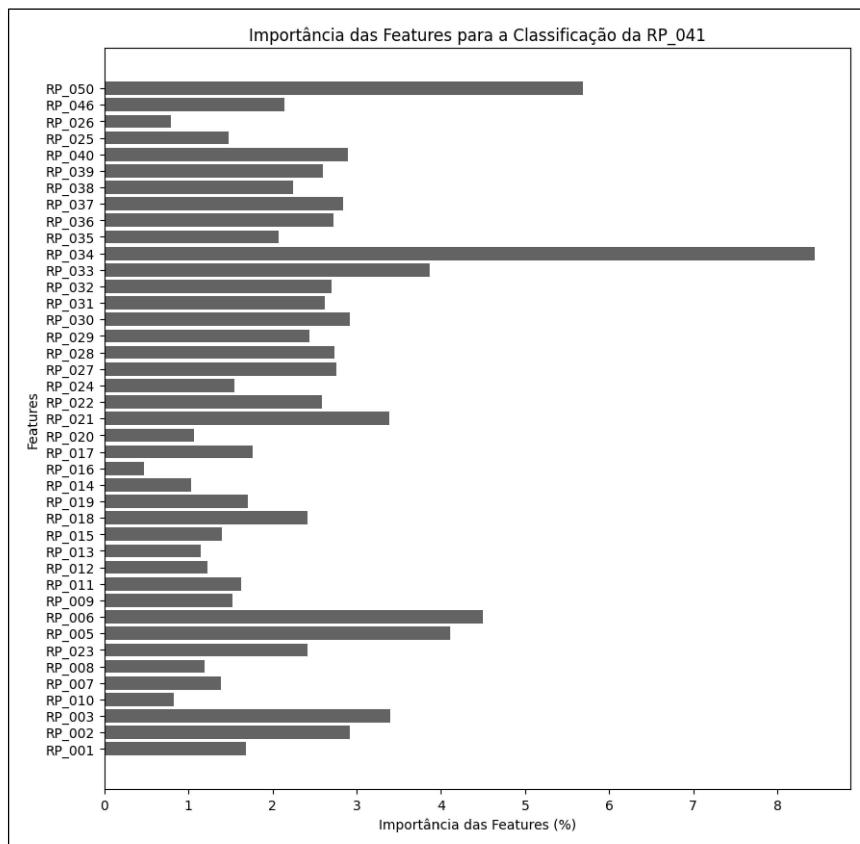
- SVM teve o pior desempenho, indicando que pode não ser a melhor escolha para este problema específico.
- Decision Trees e KNN apresentam desempenho moderado, mas ainda são menos eficientes comparados ao MLP e à Regressão Linear.

A segunda bateria de testes trabalhou com os dados numéricos no formato discreto, ou seja, distribuídos em categorias. O mapeamento foi feito para 5 categorias de desempenho em pontuações e proficiência. Por exemplo:

Pontuações:

- Categoria 1 - Muito Crítico: 0 a 200 pontos
- Categoria 2 - Crítico: 201 a 250 pontos

Figura 23 – Seleção de Features para Regras de Classificação com a RP_041



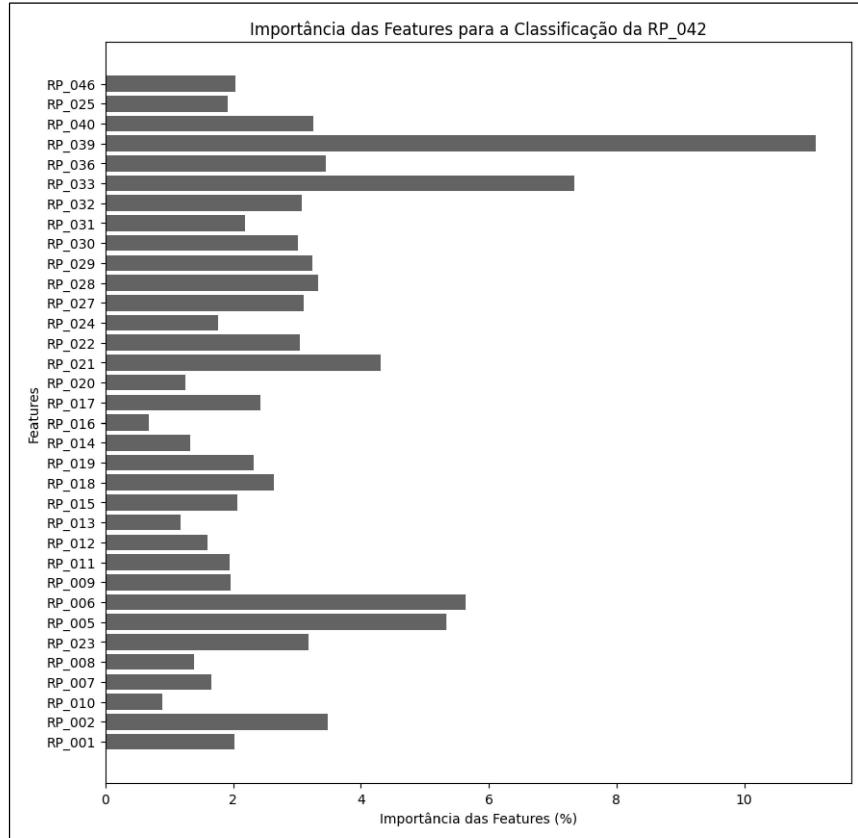
Fonte: Elaborado pelo Autor

- Categoria 3 - Intermediário: 251 a 300 pontos
- Categoria 4 - Adequado: 301 a 400 pontos
- Categoria 5 - Desejável: 401 a 500 pontos

O objetivo dos testes era criar regras de classificação para cada categoria de desempenho com base nos dados disponíveis. Sendo este um teste de Classificação. Os resultados Tabela 2 mostram o desempenho de diferentes modelos de *machine learning* na tarefa de predizer a proficiência em matemática. Cada modelo foi avaliado com base em várias métricas de desempenho, incluindo Acurácia (CA), F1 Score, Precisão (Prec) e Recall.

- **Interpretando resultados da tabela 2:**

Figura 24 – Seleção de Features para Regras de Classificação com a RP_042



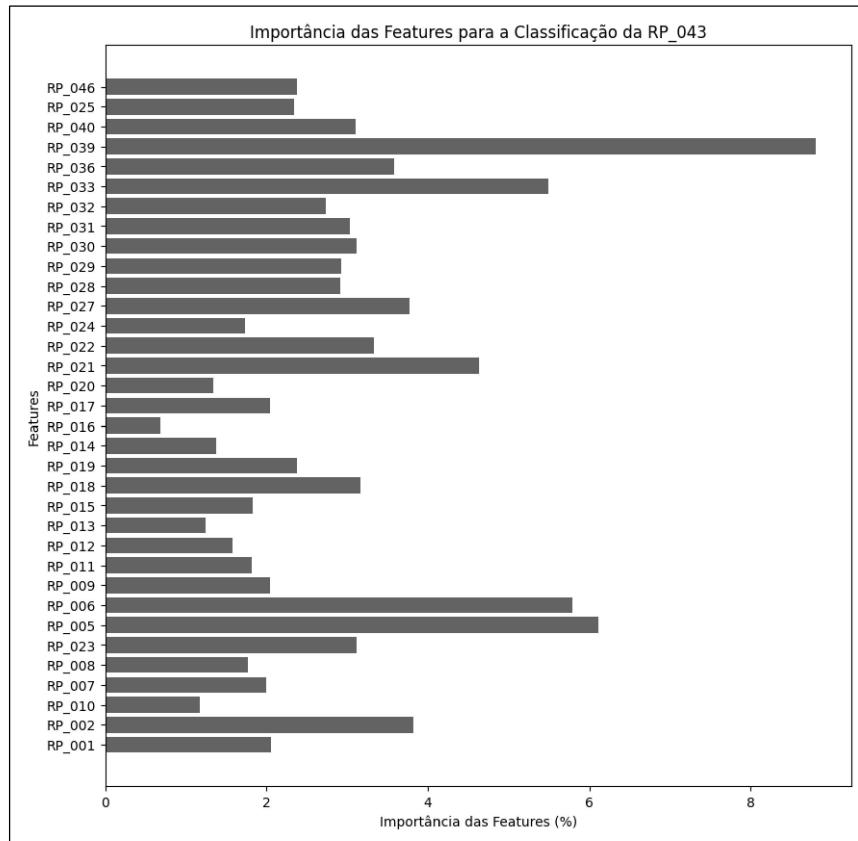
Fonte: Elaborado pelo Autor

Tabela 2 – Testes com Classificação em Predições de Proficiência em Matemática

Modelos de Classificação	CA	F1	Prec	Recall
SVM	0.280	0.280	0.295	0.280
Naive Bayes	0.448	0.330	0.308	0.448
kNN	0.376	0.343	0.345	0.376
Decision Tree	0.370	0.346	0.336	0.370
Neural Network - MLP	0.414	0.355	0.355	0.414

- O modelo SVM apresentou a menor acurácia e F1-score entre todos os modelos testados. Isso indica que o SVM não conseguiu separar adequadamente as classes no conjunto de dados, resultando em um desempenho insatisfatório.
- O modelo Naive Bayes teve a melhor acurácia e recall entre todos os modelos testados, embora a precisão e o F1-score estejam um pouco mais baixos. Isso

Figura 25 – Seleção de Features para Regras de Classificação com a RP_043

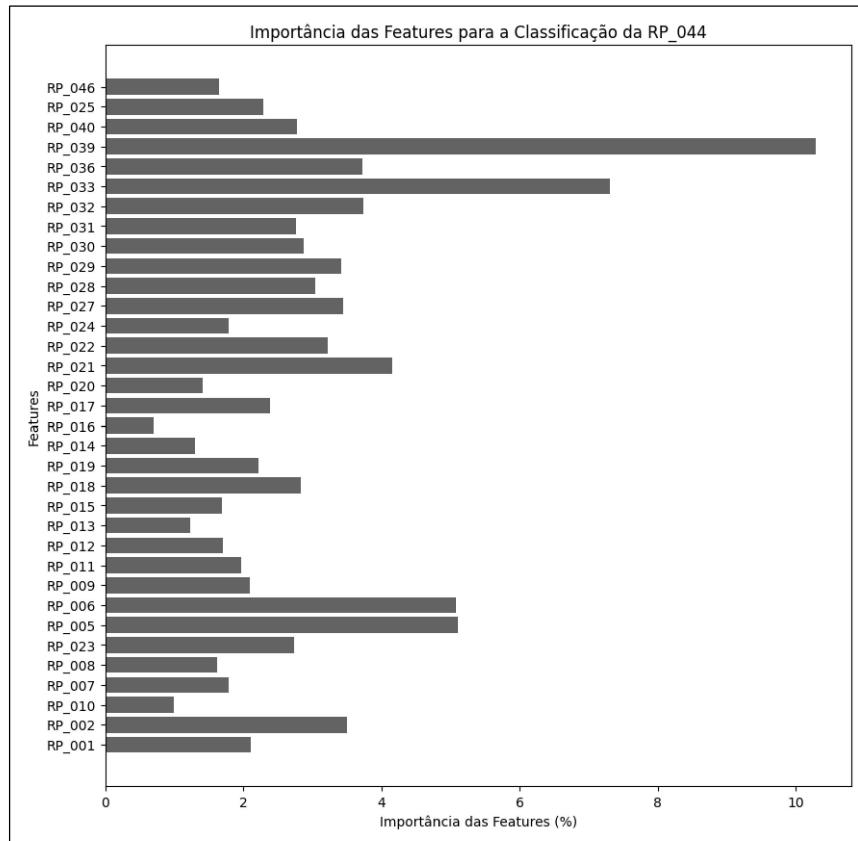


Fonte: Elaborado pelo Autor

sugere que o Naive Bayes foi capaz de identificar corretamente a maioria das instâncias de uma classe específica, mas houve mais falsos positivos.

- O kNN apresentou um desempenho intermediário, com acurácia, F1-score, precisão e recall moderados. Isso indica que o modelo conseguiu balancear razoavelmente bem entre a precisão e o recall.
- A árvore de decisão teve um desempenho semelhante ao do kNN, com todas as métricas sendo muito próximas. Isso sugere que o modelo de árvore de decisão também teve um desempenho balanceado, mas não o suficiente para superar o Naive Bayes.
- A rede neural MLP teve um desempenho superior ao do SVM, kNN e árvore

Figura 26 – Seleção de Features para Regras de Classificação com a RP_044



Fonte: Elaborado pelo Autor

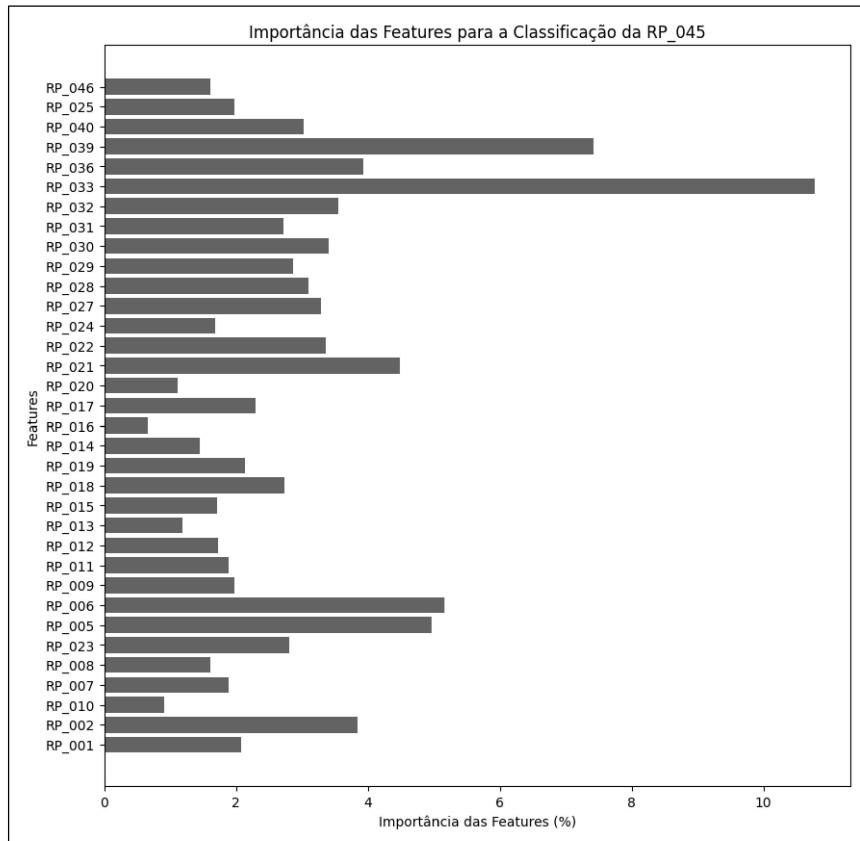
de decisão, mas inferior ao do Naive Bayes em termos de acurácia e recall.

No entanto, o MLP teve um F1-score e precisão melhores que o Naive Bayes, sugerindo um desempenho mais equilibrado entre identificar corretamente as classes positivas e minimizar os falsos positivos.

- **Conclusões sobre a tabela 2:**

- Naive Bayes se destacou em termos de acurácia e recall, sendo mais eficaz em identificar corretamente as classes positivas.
- Neural Network - MLP teve um desempenho geral equilibrado, com boas métricas de F1-score e precisão.
- kNN e Decision Tree apresentaram desempenhos similares e intermediários.

Figura 27 – Seleção de Features para Regras de Classificação com a RP_045



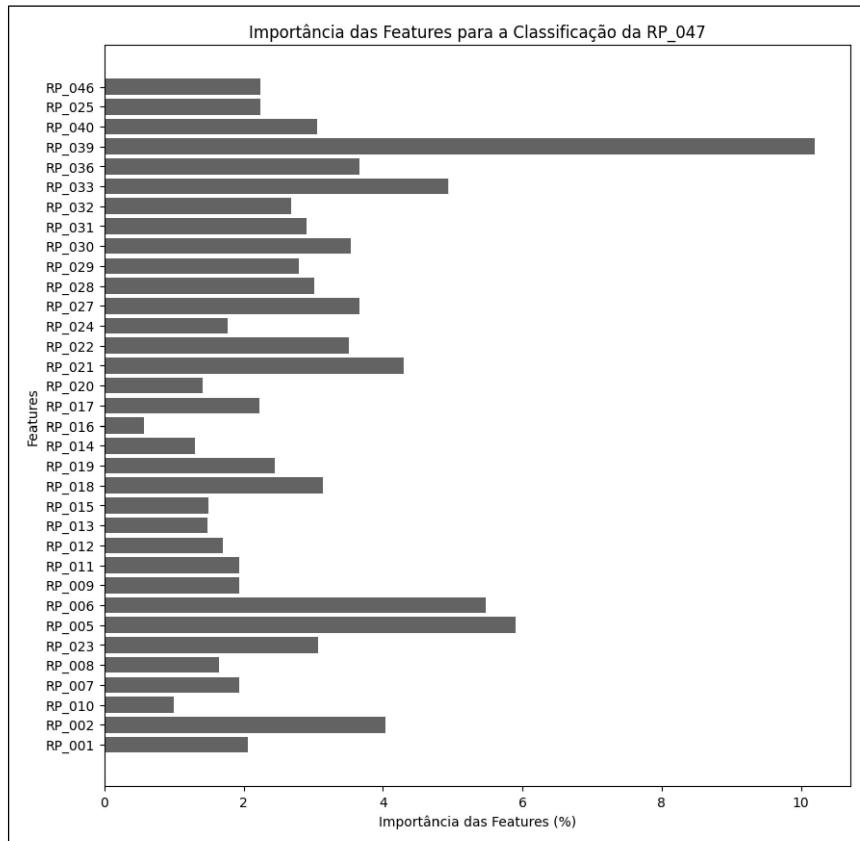
Fonte: Elaborado pelo Autor

- SVM foi o modelo com o pior desempenho entre os testados, sugerindo que pode não ser a melhor escolha para este conjunto de dados específico.

2.1.3.2 Resultados dos Testes Finais

A terceira bateria de testes trabalhou apenas com os dados do questionário socioeconômico contendo 49 atributos e aplicando os dados categóricos em um problema de Classificação. As *features* selecionadas foram os atributos demonstrados na Sub sessão 2.1.2.1. Nesta sessão serão apresentados os resultados mais promissores.

Figura 28 – Seleção de Features para Regras de Classificação com a RP_047



Fonte: Elaborado pelo Autor

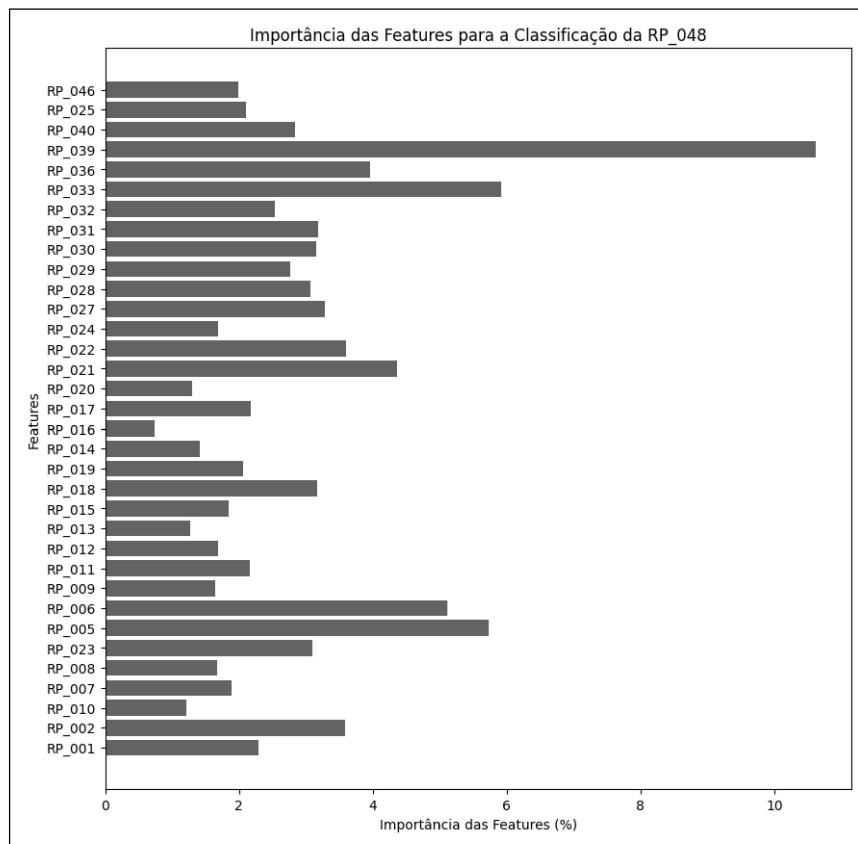
2.1.3.2.1 RP_024

Um dos resultados obtidos são observados na Tabela 3, 4, 5 e 6. O experimento em específico tentou predizer a resposta da questão "RP_024" do questionário: "A partir do 1º ano do ensino fundamental, em que tipo de escola você estudou?". As *features* descartadas foram: "RP_005" e "RP_006"

A Tabela 4 apresenta as métricas de *precision*, *recall*, *f1-score* e *support* para cada um dos modelos de classificação: *Random Forest*, *Decision Tree*, *KNN*, *SVM*, *Naive Bayes* e *MLP*.

- *Precision*: Mede a exatidão das previsões positivas. O *MLP* e o *Naive Bayes*

Figura 29 – Seleção de Features para Regras de Classificação com a RP_048



Fonte: Elaborado pelo Autor

Tabela 3 – Matriz de Confusão dos Modelos de Classificação para RP_24

Modelos de Classificação	Matriz de Confusão
<i>Random Forest</i>	[[4009, 2, 9], [287, 3, 3], [535, 2, 17]]
<i>Decision Tree</i>	[[3507, 178, 335], [192, 36, 65], [380, 73, 101]]
<i>KNN</i>	[[3936, 26, 58], [269, 9, 15], [501, 19, 34]]
<i>SVM</i>	[[4006, 0, 14], [278, 0, 15], [547, 0, 7]]
<i>Naive Bayes</i>	[[3256, 275, 489], [102, 92, 99], [251, 124, 179]]
<i>MLP</i>	[[3901, 47, 72], [239, 19, 35], [474, 32, 48]]

Tabela 4 – Precision, Recall, F1-Score e Support dos Modelos de Classificação para RP_024

Modelos de Classificação	Precision	Recall	F1-Score	Support
Random Forest	0.61	0.35	0.33	4867
Decision Tree	0.40	0.39	0.39	4867
KNN	0.44	0.36	0.35	4867
SVM	0.34	0.34	0.31	4867
Naive Bayes	0.44	0.48	0.45	4867
MLP	0.45	0.37	0.38	4867

Tabela 5 – Outras Métricas dos Modelos de Classificação de RP_024 (Parte 1)

Modelos de Classifica- ção	Kappa	Sensibilidade (Recall)	Especificidade	Precisão (PPV)
Random Forest	0.04	[0.997/ 0.010/ 0.031]	[0.997/ 0.010/ 0.031]	[0.830/ 0.429/ 0.586]
Decision Tree	0.14	[0.872/ 0.123/ 0.182]	[0.872/ 0.123/ 0.182]	[0.860/ 0.125/ 0.202]
KNN	0.08	[0.979/ 0.031/ 0.061]	[0.979/ 0.031/ 0.061]	[0.836/ 0.167/ 0.318]
SVM	0.02	[0.997/ 0.000/ 0.013]	[0.997/ 0.000/ 0.013]	[0.829/ 0.000/ 0.194]
Naive Bayes	0.24	[0.810/ 0.314/ 0.323]	[0.810/ 0.314/ 0.323]	[0.902/ 0.187/ 0.233]
MLP	0.13	[0.970/ 0.065/ 0.087]	[0.970/ 0.065/ 0.087]	[0.845/ 0.194/ 0.310]

Tabela 6 – Outras Métricas dos Modelos de Classificação de RP_024 (Parte 2)

Modelos de Classifica- ção	VPN	Precisão Ba- lanceada	Prevalência	Taxa de Detecção	Intervalo de Con- fiança (95%)
Random Forest	[0.830/ 0.429/ 0.586]	0.35	1.29	0.35	(0.817/ 0.838)
Decision Tree	[0.860/ 0.125/ 0.202]	0.39	1.29	0.39	(0.737/ 0.761)
KNN	[0.836/ 0.167/ 0.318]	0.36	1.29	0.36	(0.807/ 0.828)
SVM	[0.829/ 0.000/ 0.194]	0.34	1.29	0.34	(0.814/ 0.835)
Naive Bayes	[0.902/ 0.187/ 0.233]	0.48	1.29	0.48	(0.712/ 0.737)
MLP	[0.845/ 0.194/ 0.310]	0.37	1.29	0.37	(0.804/ 0.826)

apresentam os valores mais altos de *precision*, indicando um bom desempenho na previsão de classes positivas.

- *Recall*: Mede a capacidade do modelo de encontrar todas as instâncias positivas. Naive Bayes apresenta o melhor *recall*, sugerindo que é mais eficiente em identificar corretamente as instâncias positivas.
- *F1-Score*: É a média harmônica de *precision* e *recall*, oferecendo um equilíbrio entre os dois. Novamente, Naive Bayes e MLP se destacam, mostrando um bom equilíbrio entre *precision* e *recall*.
- *Support*: Refere-se ao número de ocorrências reais de cada classe no conjunto de

dados. Todos os modelos são avaliados com o mesmo suporte total de 4867. Os valores de precision, recall e f1-score são essenciais para entender a performance geral dos modelos em diferentes aspectos.

A Tabela 5 apresenta métricas adicionais como Kappa, Sensibilidade (Recall), Especificidade e Precisão (PPV) para cada modelo.

- *Kappa*: Mede a concordância entre as previsões do modelo e as classes reais. Random Forest e Decision Tree têm valores de Kappa relativamente baixos, indicando baixa concordância.
- *Sensibilidade (Recall)*: Reflete a capacidade do modelo de identificar verdadeiros positivos. Random Forest tem uma alta sensibilidade para a classe majoritária.
- *Especificidade*: Mede a proporção de verdadeiros negativos. Valores são semelhantes aos de sensibilidade, destacando a dificuldade de alguns modelos em identificar corretamente as classes minoritárias.
- *Precisão (PPV)*: Reflete a exatidão das previsões positivas, onde Random Forest e Naive Bayes mostram melhores resultados.

A Tabela 6 complementa a análise com métricas como VPN, Precisão Balanceada, Prevalência, Taxa de Detecção e Intervalo de Confiança (95%).

- *VPN (Valor Preditivo Negativo)*: Mede a precisão das previsões negativas. Naive Bayes e MLP apresentam altos valores de VPN, indicando boa performance na identificação de verdadeiros negativos.
- *Precisão Balanceada*: Média entre sensibilidade e especificidade. Naive Bayes tem o valor mais alto, indicando melhor equilíbrio na detecção de classes.
- *Prevalência*: Proporção de cada classe no conjunto de dados, constante em todos os modelos.
- *Taxa de Detecção*: Similar à precisão balanceada, Naive Bayes se destaca.
- *Intervalo de Confiança (95%)*: Indica a precisão das estimativas. Random Forest e SVM têm intervalos de confiança mais estreitos, mostrando menor variabilidade

nas suas previsões.

As análises das tabelas indicam que:

1. **Naive Bayes** e **MLP** apresentam um bom equilíbrio entre precision, recall e f1-score, sendo eficientes em detectar classes positivas e negativas.
2. **Random Forest** e **Decision Tree** têm dificuldades em identificar corretamente classes minoritárias, refletindo nos valores mais baixos de Kappa.
3. **SVM** tem a menor precisão geral, particularmente com dificuldades em lidar com classes minoritárias.
4. **KNN** mostra desempenho intermediário, com boa precisão nas classes majoritárias, mas dificuldades nas minoritárias.
5. **Naive Bayes** apresenta uma boa precisão balanceada e taxa de detecção, sendo eficaz em um contexto de classes desbalanceadas.

Cada modelo possui características específicas que podem ser exploradas dependendo do problema de classificação e da distribuição das classes no conjunto de dados.

2.1.3.2.2 RP_025

Outro dos testes teve os resultados das Tabela 7, 8, 9 e 10. O experimento em específico tentou predizer a resposta da questão RP_025 do questionário: "Você já foi reprovado?". As *features* descartadas foram: "RP_003"

A Tabela 7 apresenta a matriz de confusão para seis modelos de classificação (Random Forest, Decision Tree, KNN, SVM, Naive Bayes, MLP). Analisando cada matriz de confusão, podemos fazer as seguintes observações:

- Random Forest: Mostra um bom desempenho na classe majoritária (1), com 3760 verdadeiros positivos. As classes minoritárias (2 e 3) são menos bem preditas, com altos números de falsos negativos e falsos positivos.
- Decision Tree: Apresenta uma boa capacidade de previsão para a classe majoritária,

Tabela 7 – Matriz de Confusão dos Modelos de Classificação para RP_025

Modelos de Classificação	Matriz de Confusão
<i>Random Forest</i>	[[3760, 19, 1], [812, 40, 2], [215, 15, 3]]
<i>Decision Tree</i>	[[3093, 570, 117], [643, 169, 42], [151, 64, 18]]
<i>KNN</i>	[[3621, 157, 2], [802, 49, 3], [213, 20, 0]]
<i>SVM</i>	[[3754, 26, 0], [832, 22, 0], [213, 20, 0]]
<i>Naive Bayes</i>	[[3428, 289, 63], [640, 163, 51], [149, 47, 37]]
<i>MLP</i>	[[3647, 110, 23], [774, 68, 12], [197, 29, 7]]

Tabela 8 – Relatório de Classificação dos Modelos para RP_025

Modelo	Precision	Recall	F1-Score	Support
<i>Random Forest</i>	0.79	0.99	0.88	3780
<i>Decision Tree</i>	0.80	0.82	0.81	3780
<i>KNN</i>	0.78	0.96	0.86	3780
<i>SVM</i>	0.78	0.99	0.88	3780
<i>Naive Bayes</i>	0.81	0.91	0.86	3780
<i>MLP</i>	0.79	0.96	0.87	3780

Tabela 9 – Métricas Adicionais dos Modelos de Classificação de RP_025 (Parte 1)

Modelo	Kappa	Sensibilidade	Especificidade	Precisão (PPV)
<i>Random Forest</i>	0.06	0.99 / 0.05 / 0.01	0.99 / 0.05 / 0.01	0.79 / 0.54 / 0.50
<i>Decision Tree</i>	0.07	0.82 / 0.20 / 0.08	0.82 / 0.20 / 0.08	0.80 / 0.21 / 0.10
<i>KNN</i>	0.02	0.96 / 0.06 / 0.00	0.96 / 0.06 / 0.00	0.78 / 0.22 / 0.00
<i>SVM</i>	0.03	0.99 / 0.03 / 0.00	0.99 / 0.03 / 0.00	0.78 / 0.32 / 0.00
<i>Naive Bayes</i>	0.17	0.91 / 0.19 / 0.16	0.91 / 0.19 / 0.16	0.81 / 0.33 / 0.25
<i>MLP</i>	0.08	0.96 / 0.08 / 0.03	0.96 / 0.08 / 0.03	0.79 / 0.33 / 0.17

Tabela 10 – Métricas Adicionais dos Modelos de Classificação de RP_025 (Parte 2)

Modelo	VPN	F1 Score	Precisão Balanceada	Intervalo de Confiança (95%)
<i>Random Forest</i>	0.79 / 0.54 / 0.50	0.88 / 0.09 / 0.03	0.35	(0.77, 0.79)
<i>Decision Tree</i>	0.80 / 0.21 / 0.10	0.81 / 0.20 / 0.09	0.36	(0.66, 0.69)
<i>KNN</i>	0.78 / 0.22 / 0.00	0.86 / 0.09 / 0.00	0.34	(0.74, 0.77)
<i>SVM</i>	0.78 / 0.32 / 0.00	0.88 / 0.05 / 0.00	0.34	(0.76, 0.79)
<i>Naive Bayes</i>	0.81 / 0.33 / 0.25	0.86 / 0.24 / 0.19	0.42	(0.73, 0.76)
<i>MLP</i>	0.79 / 0.33 / 0.17	0.87 / 0.13 / 0.05	0.36	(0.75, 0.78)

mas também mostra muitas confusões entre as classes, especialmente para a classe 2, que tem um grande número de falsos negativos.

- KNN: Semelhante ao Random Forest na classe majoritária, mas com muitas dificuldades em prever corretamente as classes 2 e 3.
- SVM: Muito preciso na classe majoritária, mas praticamente incapaz de prever corretamente as classes 2 e 3, indicando um problema com desbalanceamento de classes.
- Naive Bayes: Melhor desempenho relativo nas classes minoritárias comparado a outros modelos, mas ainda com muitos erros.
- MLP: Apresenta uma boa performance na classe majoritária, mas falha nas classes minoritárias, com muitos falsos negativos.

A Tabela 8 apresenta métricas detalhadas de precisão, recall, F1-score e suporte para cada modelo:

- Random Forest: Alta precisão e recall para a classe 1, mas baixa performance nas classes 2 e 3. O F1-score reflete isso, com uma alta pontuação para a classe 1 e baixas para as outras.
- Decision Tree: Similar ao Random Forest, mas com menor precisão e recall para a classe 1 e ligeiramente melhor performance para as classes 2 e 3.
- KNN: Alta precisão e recall para a classe 1, mas praticamente incapaz de prever a classe 3 (F1-score de 0).
- SVM: Alta precisão e recall para a classe 1, mas muito fraco nas classes 2 e 3.
- Naive Bayes: Mostra um balanço melhor entre precisão e recall em todas as classes, mas ainda com baixa performance nas classes minoritárias.
- MLP: Similar ao KNN, com boa performance na classe 1, mas fraco nas outras classes.

Por fim, as Tabelas 9 e 10 apresentam métricas adicionais como Kappa, sensibilidade, especificidade, precisão (PPV), VPN, F1-score, precisão balanceada, prevalência, taxa de detecção e intervalos de confiança:

- Random Forest: Apresenta uma sensibilidade muito alta para a classe 1, mas muito

baixa para as outras classes, resultando em uma precisão balanceada de apenas 0.35. O Kappa é muito baixo (0.06), indicando pouca concordância além do acaso.

- Decision Tree: Melhor sensibilidade e especificidade para as classes minoritárias comparado ao Random Forest, mas ainda baixa precisão balanceada (0.36) e Kappa (0.07).
- KNN: Muito alta sensibilidade e especificidade para a classe 1, mas zero para a classe 3. A precisão balanceada é baixa (0.34) e o Kappa é muito baixo (0.02).
- SVM: Alta sensibilidade para a classe 1, mas zero para a classe 3. A precisão balanceada é muito baixa (0.34) e o Kappa é igualmente baixo (0.03).
- Naive Bayes: Melhor precisão balanceada (0.42) e Kappa (0.17) comparado aos outros modelos, mostrando um desempenho mais equilibrado.
- MLP: Semelhante ao KNN em termos de sensibilidade e especificidade, com uma precisão balanceada de 0.36 e Kappa de 0.08.

2.1.3.2.3 RP_026

Outro dos testes teve os resultados das Tabela 11, 12, 13 e 14. O experimento em específico tentou predizer a resposta da questão RP_026 do questionário: "Você já abandonou a escola durante o período de aulas e ficou fora da escola o resto do ano?". As *features* descartadas foram: "RP_003"

Tabela 3.1: Outras Métricas (Parte 1)

Tabela 3.2: Outras Métricas (Parte 2)

Tabela 11 – Matriz de Confusão dos Modelos de Classificação para RP_026

Modelos de Classificação	Matriz de Confusão
<i>Random Forest</i>	[[4510, 0, 0], [281, 10, 1], [63, 2, 0]]
<i>Decision Tree</i>	[[4196, 253, 61], [238, 46, 8], [54, 6, 5]]
<i>KNN</i>	[[4488, 22, 0], [280, 12, 0], [61, 4, 0]]
<i>SVM</i>	[[4510, 0, 0], [292, 0, 0], [65, 0, 0]]
<i>Naive Bayes</i>	[[3878, 591, 41], [173, 114, 5], [37, 25, 3]]
<i>MLP</i>	[[4446, 49, 15], [262, 24, 6], [57, 7, 1]]

Tabela 12 – Relatório de Classificação dos Modelos para RP_026

Modelo	Precision	Recall	F1-Score	Support
<i>Random Forest</i>	0.93 / 0.83 / 0.00	1.00 / 0.03 / 0.00	0.96 / 0.07 / 0.00	4510 / 292 / 65
<i>Decision Tree</i>	0.93 / 0.15 / 0.07	0.93 / 0.16 / 0.08	0.93 / 0.15 / 0.07	4510 / 292 / 65
<i>KNN</i>	0.93 / 0.32 / 0.00	1.00 / 0.04 / 0.00	0.96 / 0.07 / 0.00	4510 / 292 / 65
<i>SVM</i>	0.93 / 0.00 / 0.00	1.00 / 0.00 / 0.00	0.96 / 0.00 / 0.00	4510 / 292 / 65
<i>Naive Bayes</i>	0.95 / 0.16 / 0.06	0.86 / 0.39 / 0.05	0.90 / 0.22 / 0.05	4510 / 292 / 65
<i>MLP</i>	0.93 / 0.30 / 0.05	0.99 / 0.08 / 0.02	0.96 / 0.13 / 0.02	4510 / 292 / 65

Tabela 13 – Métricas Adicionais dos Modelos de Classificação de RP_026 (Parte 1)

Modelo	Kappa	Sensibilidade	Especificidade	Precisão (PPV)
<i>Random Forest</i>	0.06	1.00 / 0.03 / 0.00	1.00 / 0.03 / 0.00	0.93 / 0.83 / 0.00
<i>Decision Tree</i>	0.10	0.93 / 0.16 / 0.08	0.93 / 0.16 / 0.08	0.93 / 0.15 / 0.07
<i>KNN</i>	0.06	1.00 / 0.04 / 0.00	1.00 / 0.04 / 0.00	0.93 / 0.32 / 0.00
<i>SVM</i>	0.00	1.00 / 0.00 / 0.00	1.00 / 0.00 / 0.00	0.93 / 0.00 / 0.00
<i>Naive Bayes</i>	0.16	0.86 / 0.39 / 0.05	0.86 / 0.39 / 0.05	0.95 / 0.16 / 0.06
<i>MLP</i>	0.11	0.99 / 0.08 / 0.02	0.99 / 0.08 / 0.02	0.93 / 0.30 / 0.05

Tabela 14 – Métricas Adicionais dos Modelos de Classificação de RP_026 (Parte 2)

Modelo	VPN	F1 Score	Precisão Balanceada	Intervalo de Confiança (95%)
<i>Random Forest</i>	0.93 / 0.83 / 0.00	0.96 / 0.07 / 0.00	0.34	(0.92, 0.94)
<i>Decision Tree</i>	0.93 / 0.15 / 0.07	0.93 / 0.15 / 0.07	0.39	(0.86, 0.88)
<i>KNN</i>	0.93 / 0.32 / 0.00	0.96 / 0.07 / 0.00	0.35	(0.92, 0.93)
<i>SVM</i>	0.93 / 0.00 / 0.00	0.96 / 0.00 / 0.00	0.33	(0.92, 0.93)
<i>Naive Bayes</i>	0.95 / 0.16 / 0.06	0.90 / 0.22 / 0.02	0.43	(0.81, 0.83)
<i>MLP</i>	0.93 / 0.04	0.95 / 0.12 / 0.02	0.36	(0.91, 0.93)

REFERÊNCIAS

- AGGARWAL, C. C.; AGGARWAL, C. C. Outlier detection in categorical, text, and mixed attribute data. **Outlier analysis**, Springer, p. 249–272, 2017.
- ALIZAMIR, M.; KIM, S.; KISI, O.; ZOUNEMAT-KERMANI, M. A comparative study of several machine learning based non-linear regression methods in estimating solar radiation: Case studies of the usa and turkey regions. **Energy**, Elsevier, v. 197, p. 117239, 2020.
- CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM computing surveys (CSUR)**, ACM New York, NY, USA, v. 41, n. 3, p. 1–58, 2009.
- FREUND, R. J.; WILSON, W. J.; SA, P. **Regression analysis**. [S.l.]: Elsevier, 2006.
- FUMO, N.; BISWAS, M. R. Regression analysis for prediction of residential energy consumption. **Renewable and sustainable energy reviews**, Elsevier, v. 47, p. 332–343, 2015.
- GARCIA-GUTIERREZ, J.; MARTÍNEZ-ÁLVAREZ, F.; TRONCOSO, A.; RIQUELME, J. C. A comparative study of machine learning regression methods on lidar data: A case study. In: SPRINGER. **International Joint Conference SOCO'13-CISIS'13-ICEUTE'13: Salamanca, Spain, September 11th-13th, 2013 Proceedings**. [S.l.], 2014. p. 249–258.
- GARCÍA, S.; LUENGO, J.; HERRERA, F. **Data preprocessing in data mining**. [S.l.]: Springer, 2015. v. 72.
- GU, Z. Complex heatmap visualization. **Imeta**, Wiley Online Library, v. 1, n. 3, p. e43, 2022.
- GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. **Journal of machine learning research**, v. 3, n. Mar, p. 1157–1182, 2003.
- HARIHARA KRISHNAN, J.; MOHANAVALLI, S.; SRIVIDYA; KUMAR, K. B. S. Survey of pre-processing techniques for mining big data. In: **2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)**. [S.l.: s.n.], 2017. p. 1–5.
- JOVIĆ, A.; BRKIĆ, K.; BOGUNOVIĆ, N. A review of feature selection methods with applications. In: **2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)**. [S.l.: s.n.], 2015. p. 1200–1205.

JÚNIOR, A. G. M.; FARIAS, M. A. de. Spaece: Uma história em sintonia com avaliação educacional do governo federal. **Revista de Humanidades (Descontinuada)**, v. 31, n. 2, p. 525–547, 2016.

KIM, J.; LEE, Y.; LEE, M.-H.; HONG, S.-Y. A comparative study of machine learning and spatial interpolation methods for predicting house prices. **Sustainability**, v. 14, n. 15, 2022. ISSN 2071-1050. Disponível em: <<https://www.mdpi.com/2071-1050/14/15/9056>>.

LIMA, A. C.; ANDRADE, F. R. B. O sistema permanente de avaliação da educação básica do ceará (spaece) como expressão da política pública de avaliação educacional do estado. 2008.

LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. In: **2008 Eighth IEEE International Conference on Data Mining**. [S.l.: s.n.], 2008. p. 413–422.

MODARESI, F.; ARAGHINEJAD, S.; EBRAHIMI, K. A comparative assessment of artificial neural network, generalized regression neural network, least-square support vector regression, and k-nearest neighbor regression for monthly streamflow forecasting in linear and nonlinear conditions. **Water resources management**, Springer, v. 32, p. 243–258, 2018.

NOVAKOVIĆ, J. D.; VELJOVIĆ, A.; ILIĆ, S. S.; PAPIĆ, Ž.; TOMOVIĆ, M. Evaluation of classification models in machine learning. **Theory and Applications of Mathematics & Computer Science**, "Aurel Vlaicu"University of Arad Department of Mathematics and Computer . . . , v. 7, n. 1, p. 39, 2017.

RAHM, E.; DO, H. H. *et al.* Data cleaning: Problems and current approaches. **IEEE Data Eng. Bull.**, v. 23, n. 4, p. 3–13, 2000.

SAHOO, K.; SAMAL, A. K.; PRAMANIK, J.; PANI, S. K. Exploratory data analysis using python. **International Journal of Innovative Technology and Exploring Engineering**, v. 8, n. 12, p. 4727–4735, 2019.

SIAL, A. H.; RASHDI, S. Y. S.; KHAN, A. H. Comparative analysis of data visualization libraries matplotlib and seaborn in python. **International Journal**, v. 10, n. 1, p. 45, 2021.

STEIGER, J. H. Tests for comparing elements of a correlation matrix. **Psychological bulletin**, American Psychological Association, v. 87, n. 2, p. 245, 1980.

TUV, E.; BORISOV, A.; RUNGER, G.; TORKKOLA, K. Feature selection with ensembles, artificial variables, and redundancy elimination. **The Journal of Machine Learning Research**, JMLR. org, v. 10, p. 1341–1366, 2009.

YIM, A.; CHUNG, C.; YU, A. **Matplotlib for Python Developers: Effective techniques for data visualization with Python.** [S.l.]: Packt Publishing Ltd, 2018.

YU, L.; LIU, H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In: **Proceedings of the 20th international conference on machine learning (ICML-03).** [S.l.: s.n.], 2003. p. 856–863.