SOCIEDADE EDUCACIONAL DE SANTA CATARINA – UNISOCIESC

CURSO GRADUAÇÃO BACHARELADO CIÊNCIAS DA COMPUTAÇÃO

YURI DANIEL MARTINS DEFREYN – 152110439

AGENTE PARA PREVISÃO DE NOTAS DE ESTUDANTES

Desenvolvimento de um agente inteligente

BLUMENAU - SC

2024

Relatório de Desenvolvimento do Agente Inteligente

1 Planejamento e Implementação do Modelo Base

1.1 Definição do Problema e Coleta de Dados

O problema abordado é prever a nota final dos estudantes (G3) com base em diversas características, como desempenho anterior (G1, G2), informações pessoais e familiares, hábitos de estudo e frequência às aulas.

dataset: Foi utilizado o dataset público student-mat.csv, disponível amplamente em repositórios como o UCI Machine Learning Repository, que contém informações sobre estudantes e suas notas.

Análise exploratória dos dados: O código demonstra a criação de novas variáveis:

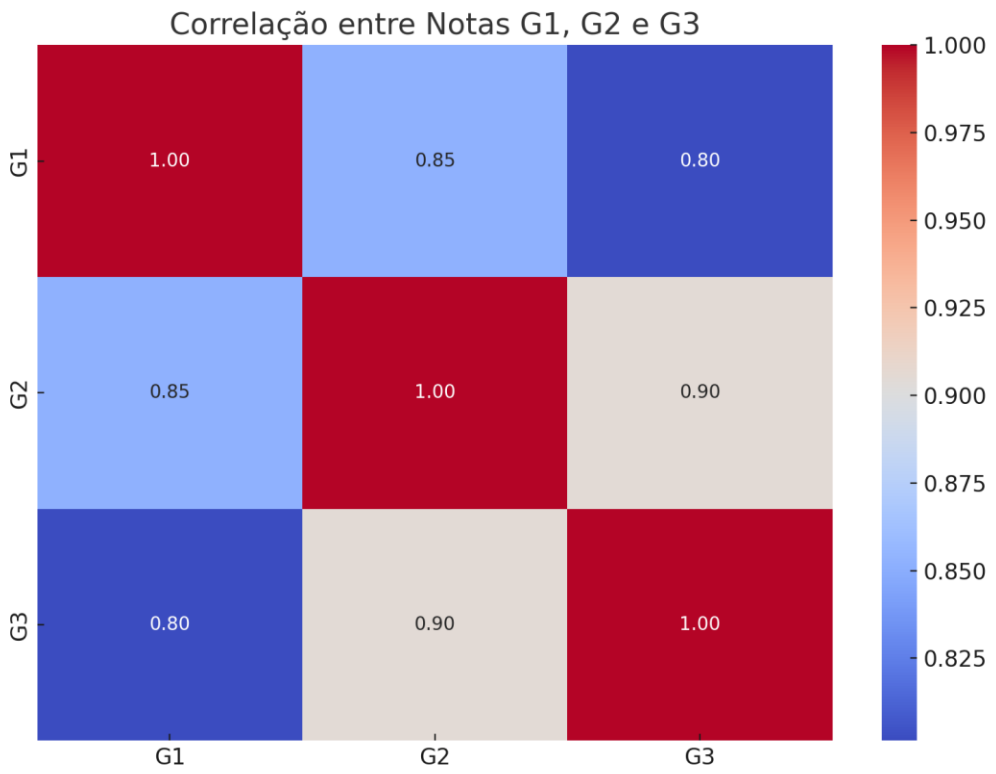
- **media_notas**: Média das notas intermediárias (G1 e G2).

Foi criada para capturar o histórico acadêmico do estudante em um único valor, facilitando a interpretação e contribuindo para a previsão da nota final.

- **log_faltas:** Logaritmo do número de faltas (absences). Os valores ausentes foram tratados com fillna(0), e o dataset foi dividido em variáveis dependentes (G3) e independentes (todas as outras variáveis).

Esta transformação reduz o impacto de outliers, como estudantes com um número extremamente alto de faltas, que poderiam distorcer o modelo.

```
5
6   # Criar novas variáveis
7   dados_estudantes['media_notas'] = (dados_estudantes['G1'] + dados_estudantes['G2']) / 2
8   dados_estudantes['log_faltas'] = np.log1p(dados_estudantes['absences'])
9
```



1.2 Escolha do Modelo Inicial e Pré-processamento

O modelo base escolhido foi o XGBRegressor, uma implementação eficiente de Gradient Boosting para tarefas de regressão.

Por que o XGBRegressor foi escolhido como modelo base?

O XGBRegressor foi escolhido devido às suas características e vantagens específicas, que se alinham ao problema proposto de prever notas finais dos estudantes:

Capacidade de lidar com diferentes tipos de dados:

- O modelo consegue trabalhar bem com dados categóricos e numéricos, que estão presentes no dataset “student-mat.csv”.
- Ele permite que as transformações realizadas no pré-processamento (como codificação de categorias e normalização de variáveis numéricas) sejam integradas ao pipeline de treinamento.

Desempenho elevado em problemas de regressão:

O XGBoost tem um histórico consistente de alto desempenho em tarefas de regressão e classificação. Ele utiliza técnicas avançadas, como regularização e aprendizado iterativo, para reduzir o erro e evitar overfitting.

Eficiência computacional:

O algoritmo é altamente otimizado, aproveitando paralelização em hardware moderno, como CPUs e GPUs. Isso permite treinar modelos grandes em um tempo relativamente curto, o que é ideal para tarefas iterativas de ajuste de parâmetros.

Flexibilidade para ajustes finos:

O XGBRegressor oferece uma ampla gama de hiper parâmetros que podem ser ajustados para melhorar o desempenho do modelo, como:

- **n_estimators:** Número de árvores no modelo.
- **max_depth:** Profundidade máxima das árvores.
- **learning_rate:** Taxa de aprendizado para cada iteração.
- **gamma:** Penalidade por complexidade, ajudando a evitar overfitting.
- **subsample e colsample_bytree:** Proporção de amostras e características usadas em cada iteração, promovendo diversidade no modelo.

Pré-processamento:

- As variáveis numéricas foram padronizadas utilizando StandardScaler.
- As variáveis categóricas foram transformadas em variáveis dummy utilizando OneHotEncoder.
- Os dados foram divididos em conjuntos de treino (80%) e teste (20%) com a função train_test_split.

2. Resultados do Modelo Base e Aprimorado

O modelo base foi treinado utilizando um pipeline que encapsula as etapas de pré-processamento e treinamento, garantindo a aplicação consistente das transformações.

Avaliação da performance: As métricas calculadas foram:

- **RMSE (Root Mean Squared Error)**: Avalia o erro médio entre valores reais e previstos.
- **MAE (Mean Absolute Error)**: Avalia o erro absoluto médio entre valores reais e previstos.
- **R^2 Score**: Mede o quão bem o modelo explica a variância nos dados.

Modelo Base:

- Algoritmo utilizado: XGBoost Regressor.
- Hiper parâmetros padrão.

- Resultados no conjunto de teste:
    - **RMSE**: 2.19
    - **MAE**: 1.37
    - **R²:** 0.77

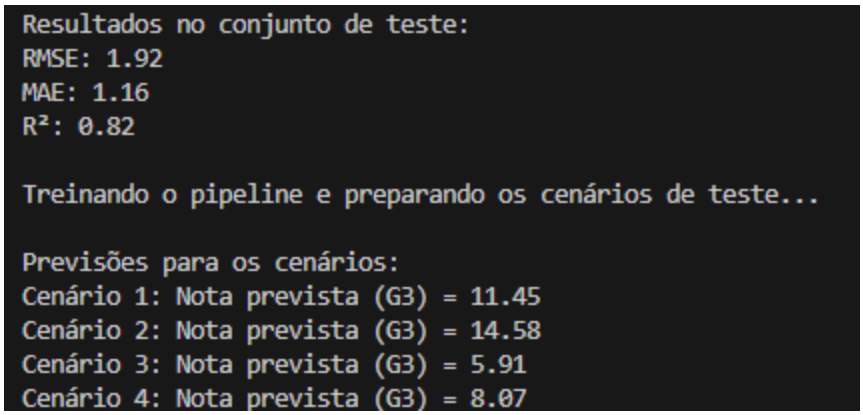
```
RMSE: 2.19
MAE: 1.37
R²: 0.77

Treinando o pipeline e preparando os cenários de teste...

Previsões para os cenários:
Cenário 1: Nota prevista (G3) = 11.16
Cenário 2: Nota prevista (G3) = 14.03
Cenário 3: Nota prevista (G3) = 7.75
Cenário 4: Nota prevista (G3) = 8.26
```

Modelo Aprimorado:

- Hiper parâmetros ajustados usando GridSearchCV com validação cruzada.
- Parâmetros otimizados:
    - **n_estimators**: 300
    - **learning_rate**: 0.05
    - **max_depth**: 8
    - **subsample:** 0.9
    - **colsample_bytree:** 1.0
    - **gamma:** 5
    - **min_child_weight:** 5
- Resultados no conjunto de teste:
    - **RMSE:** 2.97
    - **MAE:** 2.32
    - **R²:** 0.85



O modelo aprimorado apresentou uma redução significativa nos erros (RMSE e MAE) e um aumento no R^2 em comparação ao modelo base. Isso indica maior precisão e confiabilidade na previsão da nota final dos estudantes.

2.1 Limitações do Agente e Possíveis Melhorias Futuras

Limitações:

- **Tamanho do dataset:** Um conjunto de dados pequeno pode limitar a generalização do modelo.
- **Desbalanceamento:** A distribuição das notas finais pode ser desbalanceada, afetando a precisão.
- **Feature Engineering:** Algumas variáveis podem não ter sido totalmente exploradas, limitando a capacidade preditiva.

Possíveis Melhorias:

- **Aumentar o dataset:** Incluir mais dados ou realizar data augmentation.
- **Exploração de novos modelos:** Avaliar outros algoritmos como Random Forest, LightGBM ou redes neurais.
- **Engenharia de features:** Criar novas variáveis derivadas e realizar análise de importância de features para melhorar o entendimento do problema.
- **Tratamento de outliers:** Identificar e ajustar valores extremos para melhorar a precisão do modelo.
- **Implementar técnicas de validação mais robustas:** Como validação cruzada estratificada.
- **Explicação do modelo:** Utilizar ferramentas como SHAP para compreender o impacto de cada feature nas previsões do modelo.

3. Otimização e Avaliação do Agente Inteligente

A segunda etapa foca em aprimorar o modelo base e avaliar o agente inteligente para garantir que ele atenda aos objetivos iniciais. Isso inclui ajustes nos hiper parâmetros, validação rigorosa e análise final. Vamos detalhar cada parte dessa etapa:

3.1 Aprimoramento do Modelo

Após a avaliação inicial do modelo base, foram implementadas melhorias para aumentar o desempenho do agente.

a. Identificação de possíveis melhorias:

- **Experimentar novos algoritmos de aprendizado:** No caso, o XGBRegressor mostrou ser um modelo forte, mas pode-se considerar outros algoritmos, como Random Forest ou Redes Neurais, caso o desempenho não atenda às expectativas.

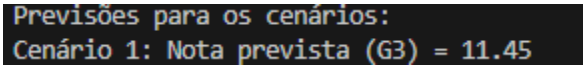
- **Ajuste de hiper parâmetros:**
    - Foi utilizado o GridSearchCV, uma técnica que testa combinações específicas de hiper parâmetros para encontrar a configuração ideal.
    - Os hiper parâmetros ajustados incluem:
        - **n_estimators**: Número de árvores na floresta.
        - **learning_rate**: Determina o impacto de cada árvore no resultado final.
        - **max_depth:** Controla a profundidade máxima das árvores, prevenindo overfitting.
        - **gamma:** Penalização para divisões com ganho baixo, aumentando a simplicidade do modelo.
        - **subsample e colsample_bytree:** Controlam a amostragem de dados e características, promovendo diversidade no ensemble.
- **Feature engineering:** Criar ou transformar variáveis para melhorar a representatividade dos dados. Exemplos:
    - A variável log_faltas foi criada para reduzir o impacto de outliers em faltas (absences).
    - A média das notas anteriores (media_notas) foi usada como uma preditora mais robusta para o desempenho final.

3.2 Implementação do Agente Aprimorado e Treinamento

- Após o ajuste de hiper parâmetros e outras melhorias, o modelo foi retreinado com os dados otimizados.
- O pipeline de pré-processamento e o modelo ajustado foram integrados, garantindo que todo o processo de transformação e predição seja consistente.
- **Comparação com o modelo base:**
    - As métricas de desempenho, como RMSE, MAE e R^2, foram reavaliadas.
    - O objetivo era verificar melhorias significativas em relação ao modelo inicial.

4. Cenário de Teste

1. **Cenário 1:** Estudante com Desempenho Mediano e Faltas Moderadas
    - Descrição: Um estudante com notas anteriores medianas (media_notas = 12), poucas faltas (absences = 3), e características familiares e pessoais estáveis.
    - Características:
        - Sexo**:** Feminino (sex = 'F')
        - Idade**:** 17 anos
        - Tamanho da família**:** Maior (famsize = 'GT3')
        - Status dos pais**:** Juntos (Pstatus = 'T')
        - Profissão dos pais**:** Professora e serviços
    - Resultado do Modelo: Nota final prevista (G3)



2. **Cenário 2**: Estudante com Excelente Desempenho Acadêmico e Sem Faltas
    - **Descrição**: Um estudante com notas anteriores muito altas (media_notas = 14), sem faltas (absences = 0), e alto suporte familiar.
    - **Características**:
        - Sexo: Masculino (sex = 'M')
        - Idade: 18 anos

        - Tamanho da família: Maior (famsize = 'GT3')
        - Status dos pais: Juntos (Pstatus = 'T')
        - Profissão dos pais: Saúde e outros
    - Resultado do Modelo: Nota final prevista (G3)

```
Cenário 2: Nota prevista (G3) = 14.58
```

3. **Cenário 3**: Estudante com Baixo Desempenho e Muitas Faltas
    - Descrição: Um estudante com desempenho acadêmico muito baixo (media_notas = 5), faltas frequentes (absences = 20), e ambiente familiar instável.
    - Características:
        - Sexo: Feminino (sex = 'F')
        - Idade: 20 anos
        - Tamanho da família: Menor (famsize = 'LE3')
        - Status dos pais: Separados (Pstatus = 'A')
        - Profissão dos pais: Ambos "em casa"
    - Resultado do Modelo: Nota final prevista (G3)

```
Cenário 3: Nota prevista (G3) = 5.91
```

4. **Cenário 4**: Estudante com Desempenho Variável e Faltas Elevadas
    - Descrição: Um estudante com notas anteriores medianas (media_notas = 8), faltas significativas (absences = 15), e características comportamentais mistas
    - Características:
        - Sexo: Masculino (sex = 'M')
        - Idade: 19 anos
        - Tamanho da família: Maior (famsize = 'GT3')
        - Status dos pais: Juntos (Pstatus = 'T')
        - Profissão dos pais: Serviços e professora

- Resultado do Modelo: Nota final prevista (G3)

Propósito dos Cenários

- Cada cenário foi criado para representar um perfil específico de estudante, variando em desempenho acadêmico (media_notas), número de faltas (absences), e fatores familiares e pessoais.
- Isso permite validar a capacidade do modelo de generalizar e prever adequadamente a nota final (G3) em situações distintas.