

Revolucionando o ensino com a IA de correção de redações

Imagine um futuro onde cada aluno recebe feedback personalizado e detalhado em suas redações, aprendendo e aprimorando sua escrita de forma mais eficiente, conforme as solicitações de cada vestibular. Essa é a promessa de nosso projeto, uma IA de correção de redações, uma ferramenta que tem o potencial de revolucionar o ensino e democratizar o acesso à educação de qualidade.

Apresentação da Equipe

Alunos responsáveis por este projeto, desenvolvido na disciplina de Projeto Aplicado 2.

Willians Carvalho da Silva – 10416087

João Pedro Santos Oliveira – 10423752

Cesar Valentim Silva – 10416087

Luciano Guimaraes Costas – 10289655





Como a IA pode melhorar a educação

1 Feedback Personalizado

A IA analisa cada redação, identificando erros gramaticais, de ortografia, pontuação e estilo, fornecendo feedback individualizado e específico para cada aluno.

2 Aprendizado Adaptativo

Com base no feedback da IA, os alunos podem identificar seus pontos fracos e focar em áreas específicas para aprimoramento, personalizando seu aprendizado de acordo com suas necessidades.

3 Liberação de Tempo

Professores podem dedicar mais tempo a atividades de ensino mais interativas e personalizadas, como debates e projetos em grupo, liberando-os de tarefas repetitivas de correção.

Benefícios da IA para escolas públicas e comunidades

Acesso Equitativo

A IA de correção de redações pode ser disponibilizada gratuitamente para escolas públicas e comunidades, democratizando o acesso à educação de qualidade e proporcionando oportunidades iguais para todos.

Aprimoramento da Aprendizagem

Com feedback constante e personalizado, os alunos podem desenvolver habilidades de escrita mais sólidas, preparando-os para o sucesso acadêmico e profissional.

Comunidades Fortalecimento

A IA pode ser utilizada em programas de alfabetização e letramento, fortalecendo as comunidades e proporcionando oportunidades de desenvolvimento pessoal e profissional.



O potencial da IA para a educação gratuita

1

Acessibilidade Global

A IA pode ser disponibilizada online, tornando a educação de qualidade acessível para todos, independentemente de localização geográfica ou recursos financeiros.

2

Aprendizado Autônomo

A IA permite que os alunos aprendam no seu próprio ritmo, oferecendo feedback e suporte personalizados para cada etapa do processo de aprendizado.

3

Desenvolvimento de Habilidades Essenciais

A IA pode ajudar os alunos a desenvolver habilidades de escrita e comunicação essenciais para o sucesso no mercado de trabalho.

Vantagens competitivas para escolas privadas

1

Diferenciação Acadêmica

A IA de correção de redações oferece uma vantagem competitiva para escolas privadas ou cursinhos, diferenciando-as pela qualidade do ensino e do feedback oferecido aos alunos. Isso se mostra um argumento muito forte de venda, visto que o foco delas é o vestibular e a redação é a parte que mais pesa na nota do aluno.

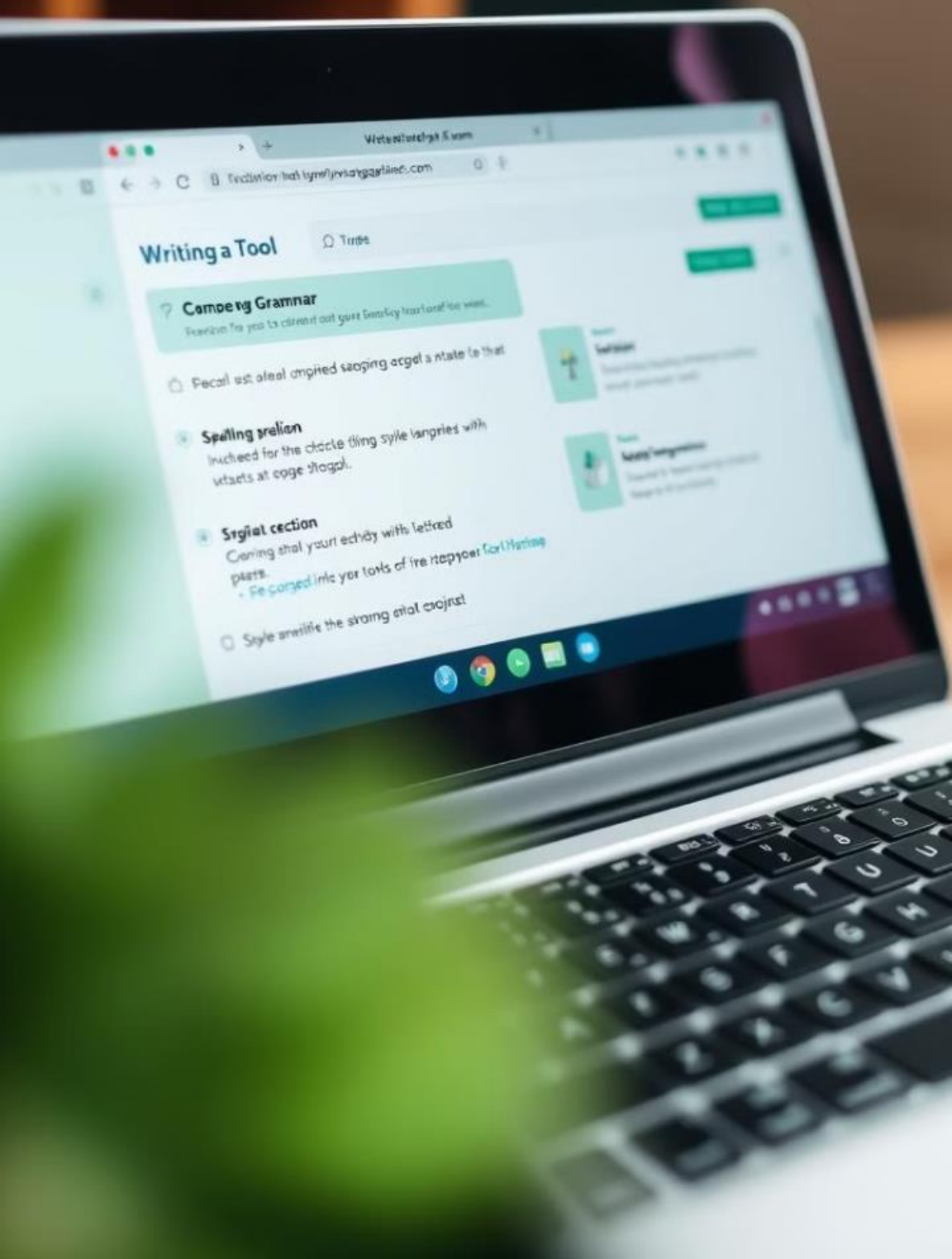
2

Aprimoramento da Qualidade

O feedback personalizado da IA permite que os alunos melhorem suas habilidades de escrita de forma mais rápida e eficiente, elevando o nível geral da qualidade da escrita na escola, além de mostrar os pontos de melhoria necessários.



Destques da funcionalidade da IA



Análise Gramatical

Identifica e corrige erros gramaticais, como concordância verbal e nominal, uso de tempos verbais e colocação pronominal.

Verificação Ortográfica

Detecta e corrige erros de ortografia, incluindo palavras mal escritas, homófonos e grafia incorreta de nomes próprios.

Análise de Estilo

Avalia o estilo da escrita, identificando problemas de clareza, concisão, coesão e fluidez, oferecendo sugestões para melhorar a qualidade do texto.

Benchmark de IA para Correção de Redações

Realizar um **benchmark** é essencial para entender como as melhores soluções de IA para correção de textos estão estruturadas e identificar práticas que se destacam no mercado. Este estudo comparativo examina plataformas de referência que combinam tecnologia e métodos de ensino, oferecendo feedback detalhado e personalizado. Usando a Aprova Total, ProWritingAid e Scribbr como inspirações, nosso projeto visa aprimorar a experiência do usuário, fornecendo correções adaptadas às necessidades de cada estudante e auxiliando na melhoria contínua de suas habilidades de escrita.

Aprova Total

Diferenciais:

- **Foco em Redação para ENEM e Vestibulares:** Correção adaptada às exigências específicas de diferentes exames, com foco em competências-chave para o ENEM, como argumentação, coesão, e domínio da língua.
- **Feedback Detalhado e Interativo:** Além de uma análise minuciosa, a plataforma fornece links para videoaulas e permite a reescrita da redação, um diferencial para fixação do aprendizado.
- **IA e Correção por Professores:** A correção usa IA e uma revisão humana final para garantir que os erros e pontos fortes sejam bem compreendidos. O processo une precisão e humanização, oferecendo insights além da gramática e pontuação.

Grammarly

Diferenciais:

- **Popularidade e Variedade de Ferramentas:** Grammarly é amplamente utilizado devido à sua versatilidade, com feedback instantâneo em gramática, concisão e até tom.
- **IA Multinível:** Usa IA para sugerir desde correções gramaticais até estilo e fluência, além de analisar o tom e adequação ao contexto.
- **Extensões e Integrações:** Com integração em navegadores e software de edição de texto, é uma ferramenta prática para revisar qualquer conteúdo rapidamente.

Limitações:

- **Sem Foco em Exames ou Competências Específicas:** Não foca em competências como argumentação e coesão, essenciais para redações de exames, e falta feedback pedagógico.

PaperRater

Diferenciais:

- **Sistema de Análise de Pontuação:** Avalia o nível da escrita e dá uma “nota” para que o usuário tenha uma ideia do nível do texto.
- **Análise Plagiária:** Além de correções gramaticais, PaperRater verifica plágio, o que é útil para estudantes e profissionais.

Pontos Fracos:

- **Correção Geral:** Não possui uma abordagem específica para ENEM ou vestibulares e nem fornece feedback pedagógico detalhado, sendo uma ferramenta mais técnica e voltada para plágio e gramática.

Parte técnica

Introdução e Objetivo

Introdução

- A análise de redações usando aprendizado de máquina visa prever a pontuação de textos com base em características como número de palavras e conteúdo textual.
- Modelos utilizados: RandomForest e XGBoost.

Objetivo

- Avaliar o desempenho dos modelos e identificar pontos de acerto e de melhoria.
- Propor estratégias de otimização para aumentar a precisão das previsões.

Desempenho dos Modelos

RandomForest

- Acurácia Atingida: 66.73
- **MSE:** 2087.28
- **R²:** 0.26 (26% da variação das notas explicada)
- Observações: Desempenho superior ao XGBoost, mas com espaço para melhorias.

XGBoost

- **MSE:** 2143.59
- **R²:** 0.24 (24% da variação explicada)
- Observações: Desempenho levemente inferior, com potencial em cenários complexos.

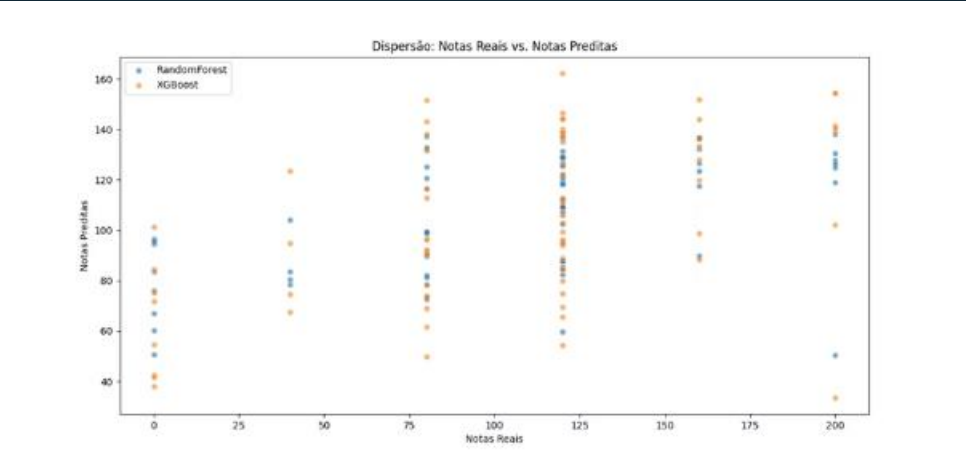
Distribuição dos Erros

- Relativamente centrada em torno de zero, porém dispersa.
- Superestimações e subestimações indicam necessidade de incluir mais características relevantes.
- Sugestões: Técnicas de NLP avançadas como Word2Vec ou BERT.

Visualizações e Interpretações

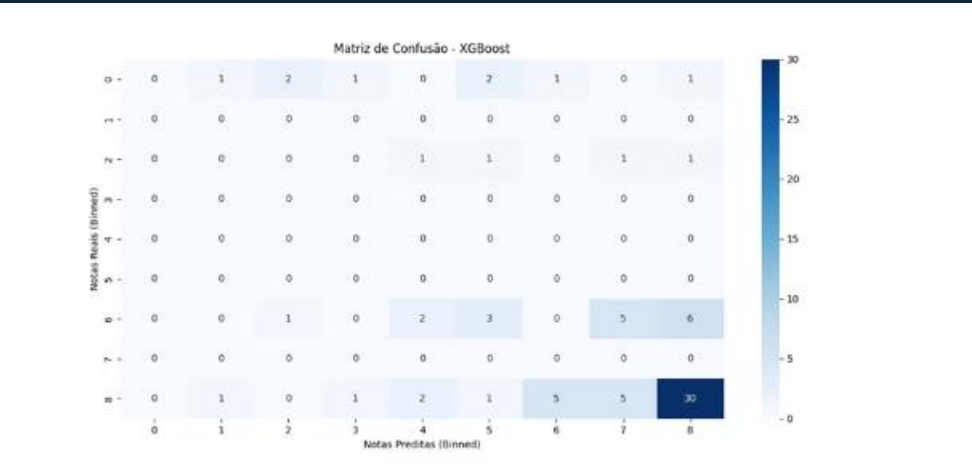
Gráfico de Dispersão: Notas Reais vs. Notas Preditas

- Descrição: Mostra a comparação entre notas reais e as previstas.
- Análise: Concentração em faixas específicas; dificuldade em prever extremos.



Matriz de Confusão (Binned)

- Descrição: Nota a precisão nas faixas médias e as dificuldades nas extremas.
- Análise: Tendência do modelo de "jogar seguro" com previsões próximas à média.



Distribuição dos Erros (Resíduos)

- Descrição: Histograma da diferença entre notas reais e previstas.
- Análise: Curva próxima a uma distribuição normal, sem viés, mas ainda com erros significativos.

Conclusão Geral

Os gráficos fornecem uma visão abrangente do desempenho dos modelos. Embora os modelos consigam capturar parcialmente os padrões das notas de redação, eles enfrentam dificuldades significativas com casos extremos e algumas faixas específicas. A matriz de confusão evidencia os erros mais frequentes, enquanto o gráfico de dispersão revela discrepâncias entre as previsões e as notas reais. Além disso, a análise da distribuição dos resíduos indica que, mesmo com um desempenho geral aceitável, ainda há áreas para aprimoramento.

Próximos Passos

- Aplicar técnicas de ensemble mais avançadas para melhorar a precisão.
- Incorporar recursos textuais mais ricos, como coesão, coerência e argumentação, usando processamento de linguagem natural (NLP) para lidar melhor com a subjetividade das avaliações de redação.
- Explorar métodos de ajuste fino e validação cruzada para reduzir erros em faixas de notas críticas.

Datasets utilizado

https://huggingface.co/datasets/kamel-usp/aes_enem_dataset (kamel-usp/aes_enem_dataset)

```
#bibliotecas necessárias
from datasets import load_dataset
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import mean_squared_error, r2_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#Carregar o dataset
dataset = load_dataset("kamel-usp/aes_enem_dataset", "sourceAOnly",
split="train", trust_remote_code=True)
df = pd.DataFrame(dataset)
```

```
#Adicionar novas features
df['word_count'] = df['essay_text'].apply(lambda x: len(x.split()))
df['year'] = df['essay_year']
```

```
#Separar textos e novas features
X_texts = df['essay_text'].tolist()
X_additional = df[['word_count', 'year']].values
y = [grade[0] for grade in df['grades']]
```

```
#Vetorizar o texto usando TF-IDF
vectorizer = TfidfVectorizer(max_features=5000, ngram_range=(1, 2))
X_text_tfidf = vectorizer.fit_transform(X_texts)
```

```
#Combinar as features de texto e numéricas
from scipy.sparse import hstack
X = hstack([X_text_tfidf, X_additional])
```

```
#Dividir em treino e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
```

```
#Modelo 1: RandomForestRegressor com GridSearchCV
rf_params = {'n_estimators': [100, 200], 'max_depth': [10, 20, None]}
rf_model = GridSearchCV(RandomForestRegressor(random_state=42), rf_params,
cv=5)
rf_model.fit(X_train, y_train)
```

```
#Avaliar RandomForest
rf_predictions = rf_model.predict(X_test)
print("Melhores parâmetros RandomForest:", rf_model.best_params_)
```

```
print("RandomForest - MSE:", mean_squared_error(y_test, rf_predictions))
print("RandomForest - R²:", r2_score(y_test, rf_predictions))
```

```
#Modelo 2: XGBoost Regressor
xgb_model = XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=6,
random_state=42)
xgb_model.fit(X_train, y_train)
```

```
#Avaliar XGBoost
xgb_predictions = xgb_model.predict(X_test)
print("XGBoost - MSE:", mean_squared_error(y_test, xgb_predictions))
print("XGBoost - R²:", r2_score(y_test, xgb_predictions))
```

```
#Calcular os resíduos (erros)
residuos = y_test - rf_predictions # Substituir por xgb_predictions, se necessário
```

```
#Gráfico: Distribuição dos Resíduos
plt.figure(figsize=(10, 6))
sns.histplot(residuos, kde=True, bins=30, color='skyblue', edgecolor='black')
plt.title('Distribuição dos Erros (Resíduos)')
plt.xlabel('Erro (Resíduo)')
plt.ylabel('Frequência')
plt.show()
```

```
#Gráfico de Dispersão: Notas Reais vs. Notas Preditas
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=rf_predictions, label='RandomForest', alpha=0.6)
sns.scatterplot(x=y_test, y=xgb_predictions, label='XGBoost', alpha=0.6)
plt.xlabel('Notas Reais')
plt.ylabel('Notas Preditas')
plt.title('Dispersão: Notas Reais vs. Notas Preditas')
plt.legend() plt.show()
```

```
#Matriz de Confusão (Binned)
bins = np.arange(0, 101, 10)
y_test_binned = np.digitize(y_test, bins)
rf_pred_binned = np.digitize(rf_predictions, bins)
```

```
#Plotar Matriz de Confusão para RandomForest
conf_matrix = confusion_matrix(y_test_binned, rf_pred_binned)
plt.figure(figsize=(10, 6))
sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='d')
plt.xlabel('Notas Preditas (Binned)')
plt.ylabel('Notas Reais (Binned)')
plt.title('Matriz de Confusão - RandomForest')
plt.show()
```