CONTEXTO ANALÍTICO

Neste desafio, você deve resolver um case de análise de desempenho acadêmico. Esse desafio foi construído em parceria pela TERA, onde o objetivo é simular um projeto de Machine Learning com características semelhantes ao que ocorre no dia a dia de uma empresa.

Imagine-se na seguinte situação: a área de marketing de uma EdTech quer mapear os perfis dos candidatos do ENEM e, nesse projeto, os analistas precisam entender quais características estão associadas a um bom ou mau desempenho de um candidato na prova. Você é o cientista de dados que atuará na resolução desse case.

Para tanto, existem dois objetivos principais:

1. **INTERPRETABILIDADE -** Construir uma regressão linear simples, com poucas variáveis importantes, visando gerar insights para os analistas no quesito desempenho do candidato no exame. Ou seja, o foco será na interpretação dos coeficientes (exemplo: se o candidato for de uma escola pública o valor Y da sua nota média irá aumentar quanto?).
2. **PREDIÇÃO -** Construir um modelo com alto poder preditivo, com mais variáveis, visando um bom desempenho e com o intuito de ser usado em uma página web como preditora de desempenho. Note que, em casos como esse, queremos ter o menor erro possível, mesmo que o modelo seja complexo e tenha uma interpretação mais difícil.

Base de dados

O conjunto de dados descreve o desempenho de candidatos do ENEM, no ano de 2021. Ele contém **168.979 observações** e um **76 variáveis explicativas** provenientes do questionário socioeconômico preenchido pelos candidatos antes do exame, além da nota média (a **target**) dos candidatos.

Entre as variáveis explicativas, existem features contínuas e discretas, mas deve-se ter atenção ao que elas significam, como por exemplo, o código de um município que é dado por um valor numérico.

Vale ressaltar que a variável target é composta diretamente por outras variáveis do banco (trata-se de um resumo da informação de desempenho) atente-se isso na hora de construir o modelo de forma coerente.

Para iniciar seu trabalho, abaixo você encontrará a base de dados, o dicionário de dados e o arquivo de notebook do google colab, que também vai te auxiliar na execução do desafio.

Análise exploratória

Algumas sugestões do que você pode fazer:

- Verificar a distribuição da variável de interesse (nota média).
- Contar o número de valores faltantes.
- Verificar a matriz de correlação entre as features contínuas.
- Scatterplots são úteis para visualizar duas variáveis contínuas.
- Plotar a distribuição da nota média (histogramas e boxplots) para as diferentes variáveis - categóricas.
- Ao final, escreva um pouco sobre o que você conseguiu entender, extrapolar e interpretar a partir da análise exploratória.

O número de variáveis é alto, então é **importante ser criterioso na montagem dos gráficos exploratórios**. Use sua intuição e raciocínio crítico para mostrar as variáveis e encontrar a informação que importa para prosseguir com sua modelagem.

Pré processamento, limpeza dos dados e construção das features

Algumas sugestões do que você pode fazer:

- Dropar colunas(features) com muitos valores faltantes.
- Buscar algum erro de preenchimento no dataset.
- Tente criar features (Exemplo: transformar variável contínua em categórica e vice-versa).
- Não remova linhas com valores faltantes, já que isso pode modificar a distribuição do dataset de validação.

Objetivo 1 - interpretabilidade usando uma regressão linear

Algumas sugestões de como trabalhar esse objetivo:

- Utilize a biblioteca statsmodel para fitar a regressão linear; use a função summary para conseguir interpretar os coeficientes.
- Aplique alguma transformação na sua variável de interesse (nota media).
- Importante lembrar que a interpretação do coeficiente muda ao aplicar uma transformação logaritmica(Saiba mais AQUI e também AQUI).
- Selecione até **6 features** para o seu modelo.
- Trate os valores faltantes.

- Aplique as transformações nas variáveis categóricas que você julgar necessárias (One hot encoding, ordinal encoding, etc…).
- Verifique a distribuição dos resíduos da regressão linear, e quais as implicações do resultado obtido.
- Reporte o R^2 dessa regressão, e a sua interpretação desse resultado. Note que o R^2 nesse tipo de dado **tende sempre** a ser muito baixo (menor que 0.4).
- As features não podem ter alta correlação (utilizar o EDA feito previamente para encontrar as features que você julgue relevantes).
- Verifique se os pressupostos da regressão linear estão sendo atendidos (Há dicas AQUI e AQUI).
- Em um breve sumário, discorra sobre a interpretação dos coeficientes obtidos pela regressão linear (Você pode ver mais sobre interpretação AQUI).

OBJETIVO 2 - PODER PREDITIVO, REGRESSÃO VIA SCIKIT-LEARN (SGD)

A ideia da segunda parte é treinar um modelo mais robusto visando o poder preditivo e a obtenção de um modelo para uso em produção (uso real em uma aplicação web).

Algumas sugestões de como trabalhar esse objetivo:

- Transforme o sua variável de interesse (ex: y_log=np.log(y)).
- Separe seu dataset original em **treino**, **teste**, **validação** (ver instruções de como fazer a separação abaixo).
- Impute os valores faltantes das variáveis **numéricas com a mediana** e os valores faltantes das **variáveis categóricas com a moda**, os imputers devem ser fitados usando o dataset de treino para depois serem aplicadas nos datasets de validação e teste, isso evitará data leakage (Dica: usar os **simpleimputer** do sklearn).
- Aplique as transformações nas variáveis categóricas que você julgar necessárias (One hot encoding, ordinal encoding e etc…), as transformações também devem ser fitadas usando o dataset de treino para depois serem aplicadas nos datasets de validação e teste, isso evitará data leakage (Dica: usar os **transformers** do sklearn).
- **Treine um modelo (modelo baseline)**, usando todas as features e sem mexer nos hiperparâmetros do modelo, compute as métricas de avaliação no dataset de validação para você ter um **baseline**.
- Tente tunar o seu modelo. **Teste diferentes hiperparâmetros**, veja as instruções abaixo, use a documentação do sklearn para entender os hiperparâmetros que você for testar.
- Compute a **importância das features** no dataset de validação (use o permutation_importance do sklearn). Usando a importância das features ,remova do seu treinamento as features menos importantes para que o seu modelo tenha no máximo 40 features, verifique novamente a performance com esse número reduzido de features (isso pode melhorar a performance e a velocidade do seu modelo)
- Adicione um breve texto com **sua interpretação** em relação à **importância das features.**

- Finalmente, compute as **métricas de avaliação no dataset de teste** para obter o proxy de performance do seu modelo em um ambiente em produção (ambiente real online).
- Adicione uma **conclusão** para fechar o seu case.

Dica: sempre que você for avaliar o seu modelo, você deve reverter suas predições da escala log para a escala normal usando uma função exponencial (ex: y_pred = np.exp(y_pred)).

Separação dos dados em treinamento e validação: os dados devem ser separados em treino, validação e teste, na fase de exploração e modelagem você pode avaliar o modelo usando o dataset de validação para evitar overfitting, e depois, com estudo fechado aplicar as métricas de avaliação no dataset de teste (simulando a performance em exemplos nunca vistos). Para esse caso você deve separar os datasets usando a função **train_test_split** do sklearn, usando como **random state o número 42**:

- Primeiro use a função função train_test_split para separar 70% para treino e 30% para validação e teste.
- Segundo, aplique novamente essa função para quebrar esses 30% em dois datasets, sendo 50% para teste e 50% para validação. Assim obtendo 70% para treino, 15% para validação e 15% para teste.

Testando diferentes formas de melhorar a regressão: Primeiro, aumente a complexidade do modelo considerando uma regressão polinomial (grau 2 ou grau 3). Avalie o desempenho do modelo e identifique situação de overfitting.

Em seguida, considere usar alguma regularização para corrigir possíveis casos de overfitting (Ridge, Lasso, ElasticNet).

Avaliação do modelo de regressão: Para fazer a avaliação do seu modelo você deve aplicar métricas de avaliação no dataset de validação (e no final do estudo no dataset de teste). As seguintes métricas são comuns em modelos de regressão:

- **R^2**: pense nesse score como uma medida do desempenho do nosso modelo em comparação com um modelo trivial que retorna sempre a média para qualquer previsão solicitada. (o valor de 1.0 representa um modelo perfeito, já um valor de 0.0 representa um modelo equivalente a um modelo aleatório)
- **Valor absoluto médio (MAE)**: que é apenas a diferença absoluta média entre os valores previstos e verdadeiros. O valor absoluto evita que desvios negativos e positivos se cancelem.
- Em vez de tomar o valor absoluto, poderíamos elevar ao quadrado as diferenças, dando-nos o **erro quadrático médio (MSE)**. Elevar a diferença também tem o efeito de enfatizar quaisquer previsões que estejam muito longe de seus verdadeiros valores.
- Para ignorar algumas previsões significativamente desviantes (outliers), é melhor usar o MAE no lugar MSE. Tudo depende do que você está buscando
- Como as unidades do MSE são o quadrado das unidades da variável de interesse, é útil usar a **raiz do erro quadrático médio** (RMSE) como métrica de avaliação

TOOLBOX E REFERÊNCIAS

Algumas referências que podem te auxiliar com o objetivo 1:

Modelo: Statsmodels

Visualização: Seaborn

Manipulação do dataset: Pandas e Numpy (use np.log para realizar transformações logarítmicas)

Algumas referências que podem te auxiliar com o objetivo 2:

Modelo: Sklearn

RandomForest

Imputar valores faltantes: SimpleImputer

Enconders de categóricas: OrdinalEncoder e OneHotEncoder

Importância das features: Permutation Importance

Métricas de avaliação: R^2, MAE, MSE, RMSE

Visualização: Seaborn

Manipulação do dataset: Pandas e Numpy (use np.log para realizar transformações logarítmicas)

Referências

Exploring and Cleaning the Bulldozer Dataset

Linear Regression