

**Centro Universitário - Fundação Escola de Comércio Álvares Penteado
(FECAP)**

Curso: Análise e Desenvolvimento de Sistemas

Título: Fast and Cheap: Comparação em Tempo Real de Preços em Aplicativos
de Transporte

Alunos: Vinícius Brandão, Murilo Dias, Guilherme Rodrigues e João Henrique
Albuquerque

Professor: Lucy Mari Tabuti

Disciplina: Inteligência Artificial e Machine Learning

São Paulo

2025

Sumário:

1. Introdução

2. DESCRIÇÃO DO ALGORITMO

3. IMPLEMENTAÇÃO

3.1 Consulta SQL para encontrar o menor preço

3.2 Uso de Pandas (Python) para filtragem de dados

3.3 Implementação do *Random Forest Regressor*

4. RESULTADOS

5. Conclusão

IMPLEMENTAÇÃO DE ALGORITMO DE IA NO PROJETO INTEGRADOR

O presente trabalho descreve a aplicação de um algoritmo de Inteligência Artificial (IA) no Projeto Integrador, cujo objetivo é comparar preços de aplicativos de corrida (Uber, 99, Táxi) e fornecer ao usuário a opção mais vantajosa. Para isso, utilizamos técnicas de busca e filtragem de dados, além de uma possível aplicação de aprendizado de máquina para análise preditiva.

1 INTRODUÇÃO Este trabalho tem como objetivo apresentar a implementação de um algoritmo de IA no Projeto Integrador, visando a otimização da escolha do menor preço de corridas em aplicativos de transporte. Para isso, utilizamos técnicas de consulta de dados e aprendizado de máquina para prever variações de preço, proporcionando ao usuário uma tomada de decisão mais eficiente.

2 DESCRIÇÃO DO ALGORITMO O principal foco do projeto é encontrar o menor preço dentro de uma base de dados existente. Para isso, utilizamos:

- Consultas SQL otimizadas, que realizam a busca do menor preço disponível para um determinado trajeto;
- Uso da biblioteca Pandas (Python) para manipulação dos dados, possibilitando a filtragem eficiente das corridas e ordenação por preço;
- Algoritmo de IA escolhido: *Random Forest Regressor*, um modelo de aprendizado de máquina baseado em árvores de decisão, capaz de prever variações de preços com base em padrões históricos.

3 IMPLEMENTAÇÃO A seguir, apresenta-se a implementação do algoritmo para busca do menor preço.

3.1 Consulta SQL para encontrar o menor preço

```
SELECT plataforma, preco  
FROM corridas  
WHERE origem = 'Ponto A' AND destino = 'Ponto B'  
ORDER BY preco ASC  
LIMIT 1;
```

3.2 Uso de Pandas (Python) para filtragem de dados

```
import pandas as pd

df = pd.read_csv("corridas.csv")
df_filtrado = df[(df["origem"] == "Ponto A") & (df["destino"] == "Ponto B")]
corrida_mais_barata = df_filtrado.nsmallest(1, "preco")
print(corrida_mais_barata)
```

3.3 Implementação do *Random Forest Regressor*

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import numpy as np

df["origem"] = LabelEncoder().fit_transform(df["origem"])
df["destino"] = LabelEncoder().fit_transform(df["destino"])

X = df[["origem", "destino", "distancia", "horario"]]
y = df["preco"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

modelo = RandomForestRegressor(n_estimators=100, random_state=42)
modelo.fit(X_train, y_train)
previsoes = modelo.predict(X_test)
print(previsoes)
```

4 RESULTADOS A abordagem escolhida garante que o sistema forneça ao usuário a opção mais econômica disponível. O uso do *Random Forest Regressor* permite prever flutuações nos preços e sugerir o melhor momento para solicitar uma corrida, aumentando a eficiência da recomendação.

5 CONCLUSÃO A implementação do algoritmo permite uma busca eficiente do menor preço em uma base de dados de corridas, otimizando a experiência do usuário e garantindo economia. Além disso, a implementação do *Random Forest Regressor* possibilita previsões mais assertivas e melhora a capacidade do sistema em recomendar momentos ideais para solicitar uma corrida.