Centro Universitário - Fundação Escola de Comércio Alvares 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 T abuti

Disciplina: Inteligência Artificial e Machine Learning

São Paulo

2025

Sumário:

1. Introdução

A precificação dinâmica de serviços de transporte é um desafio estratégico para empresas do setor, especialmente em um contexto onde variáveis como distância, tipo de veículo e demanda sazonal impactam diretamente nos custos operacionais e na experiência do cliente. Este projeto propõe uma solução baseada em machine learning para prever o preço de uma viagem de forma automatizada, considerando três fatores-chave: a distância percorrida, o tipo de veículo utilizado (sedan, SUV ou van) e o dia da semana em que a viagem ocorre.

2. DESCRIÇÃO DO ALGORITMO

Este código utiliza um modelo de Random Forest (floresta aleatória) para prever o preço de uma viagem com base em três variáveis principais: distância percorrida, tipo de veículo (sedan, SUV ou van) e dia da semana (segunda a domingo). O modelo é treinado com dados sintéticos, gerados artificialmente para simular situações reais.

A escolha do Random Forest como modelo se deve à sua capacidade de lidar com relações não lineares entre variáveis e dados não normalizados, além de ser robusto. Para tratar as variáveis como *tipo de veículo* e *dia da semana*, o One-Hot Encoding foi usado, técnica que converte categorias em vetores binários, garantindo que o modelo interpretasse corretamente essas informações.

A geração de dados sintéticos foi necessária para simular um conjunto de treinamento realista. Foram criadas regras simples: multiplicação da distância por um valor base (dependente do tipo de veículo), acréscimo fixo para finais de semana e adição de ruído gaussiano para variabilidade. Isso permitiu testar o modelo sem depender de dados reais, mantendo a lógica de precificação transparente.

3. IMPLEMENTAÇÃO;

Funcionamento do código:

1. Geração de dados sintéticos: Cria um dataset simulando corridas com:
   - Distância: 1-100 km
   - Tipo de veículo: sedan, SUV ou van
   - Dia da semana: seg a dom
2. Pré-processamento:
   - OneHotEncoder para variáveis categóricas (tipo de veículo e dia)
   - Mantém a distância sem alterações
3. Modelo:
   - Random Forest Regressor (adequado para relações não-lineares)
4. Avaliação:
   - MAE (Erro Absoluto Médio)
   - R2 (Coeficiente de determinação)

Código:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_absolute_error, r2_score

# Gerar dados sintéticos (caso não tenha dados reais)
np.random.seed(42)
n_samples = 1000
```

```
data = {
   'distancia': np.random.uniform(1, 100, n_samples),
   'tipo_veiculo': np.random.choice(['sedan', 'suv', 'van'], n_samples),
   'dia_semana': np.random.choice(['seg', 'ter', 'qua', 'qui', 'sex', 'sab', 'dom'], n_samples),
}

# Criar preço baseado nas variáveis + ruído
data['preco'] = (
   data['distancia'] * np.where(data['tipo_veiculo'] == 'sedan', 2,
                        np.where(data['tipo_veiculo'] == 'suv', 2.5, 3)) +
   np.where(data['dia_semana'].isin(['sab', 'dom']), 20, 0) +  # Final de semana mais caro
   np.random.normal(0, 10, n_samples)  # Ruído
)

df = pd.DataFrame(data)

# Separar features e target
X = df.drop('preco', axis=1)
y = df['preco']

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

# Pré-processamento: OneHotEncoding para variáveis categóricas
preprocessor = ColumnTransformer(
   transformers=[
      ('cat', OneHotEncoder(), ['tipo_veiculo', 'dia_semana'])
   ],
   remainder='passthrough'  # Mantém a coluna numérica (distancia)
)
```

```
# Criar pipeline com pré-processamento e modelo
model = Pipeline([
   ('preprocessor', preprocessor),
   ('regressor', RandomForestRegressor(n_estimators=100, random_state=42))
])

# Treinar modelo
model.fit(X_train, y_train)

# Avaliar
y_pred = model.predict(X_test)
print(f"MAE: {mean_absolute_error(y_test, y_pred):.2f}")
print(f"R²: {r2_score(y_test, y_pred):.2f}")

# Exemplo de previsão
exemplo = pd.DataFrame({
   'distancia': [50],
   'tipo_veiculo': ['suv'],
   'dia_semana': ['sab']
})

previsao = model.predict(exemplo)
print(f"\nPrevisão para 50 km, SUV no sábado: R${previsao[0]:.2f}")
```

4. Conclusão

O código apresentado é uma solução para previsão de preços de viagens, combinando técnicas de pré-processamento, um modelo de machine learning adequado e avaliação clara. Ele serve como base para escalonamento, desde que adaptado a dados reais e contextos específicos, demonstrando como variáveis simples (distância, tipo de veículo e dia) podem ser usadas para criar sistemas de precificação inteligentes.