FUNDAÇÃO ESCOLA DE COMÉRCIO ÁLVARES PENTEADO – FECAP

CIÊNCIA DA COMPUTAÇÃO

**GUSTAVO MARCELLO CORREA DE ARAUJO - 23024729 LUCCA GIORDANO - 23024522**

**PEDRO HENRIQUE DANGELO DOS REIS - 23024777 VITOR UTIMURA LOCATELI - 23024638**

[**Álgebra Linear, Vetores e Geometria Analítica**](https://ead.graduacao.fecap.br/course/view.php?id=32333)**: Entrega 2**

**São Paulo 2025**

**GUSTAVO MARCELLO CORREA DE ARAUJO - 23024729 LUCCA GIORDANO - 23024522**

**PEDRO HENRIQUE DANGELO DOS REIS - 23024777 VITOR UTIMURA LOCATELI - 23024638**

[**Álgebra Linear, Vetores e Geometria Analítica**](https://ead.graduacao.fecap.br/course/view.php?id=32333)**: Entrega 2**

Relatório Técnico apresentado ao curso de Ciência da Computação, como parte dos requisitos da disciplina de Álgebra Linear, Vetores e Geometria Analítica referente ao Projeto Interdisciplinar.

Orientador: Renata Muniz

São Paulo 2025

SUMÁRIO

[INTRODUÇÃO 4](#_83oh4vvj8o46)

[OBJETIVO 5](#_ivb01r8jsto9)

[MÉTODOS 6](#_og4tuhmqyyxc)

[DESENVOLVIMENTO 7](#_sg7h7j3dd2n4)

[CONCLUSÃO](#_psz4ieltbq52) 13

INTRODUÇÃO

Prever o custo de uma corrida antes de sua realização tem se tornado fundamental para melhorar o planejamento financeiro de passageiros e otimizar a operação de serviços de transporte. Para isso, empregamos técnicas de regressão que relacionam variáveis como distância percorrida, horário do dia e ocorrência de feriados ao valor final das viagens. Neste trabalho, exploramos três abordagens distintas: a formulação direta dos Coeficientes pelo Método dos Mínimos Quadrados, a implementação prática da Regressão Linear oferecida pela biblioteca scikit‑learn e o uso do XGBoost Regressor, que combina múltiplas árvores de decisão para capturar relações mais complexas nos dados.

Utilizando um conjunto real de registros de corridas, treinamos e avaliamos cada modelo com as métricas RMSE (Root Mean Squared Error) e R² (coeficiente de determinação), além de analisar visualmente seus resultados por meio de gráficos de dispersão e resíduos. A comparação entre os três métodos permite identificar vantagens em termos de precisão e robustez, orientando a escolha da técnica mais adequada para cenários de previsão de preços em mobilidade urbana.

OBJETIVO

Este relatório tem como objetivo comparar o desempenho de três métodos de regressão na tarefa de prever o valor de corridas com base em variáveis como distância, horário e feriados. Avaliamos a acurácia de cada abordagem utilizando as métricas RMSE e R², além de analisar graficamente os resultados para verificar a qualidade do ajuste e o comportamento dos resíduos. A partir dessa comparação, buscamos indicar qual técnica oferece melhor equilíbrio entre simplicidade de implementação e precisão preditiva para aplicações em mobilidade urbana.

MÉTODOS

Começamos particionando nossos dados em dois grupos: 80 % para treinar os modelos e 20 % para testar como eles se comportam com informações que nunca haviam visto. A seguir, exploramos três estratégias diferentes:

Primeiro, aplicamos o método dos Mínimos Quadrados na sua forma clássica, montando a matriz de variáveis e calculando diretamente os coeficientes que melhor “encaixam” nossos pontos. Esse processo nos dá uma visão transparente de como cada característica (distância, horário, feriado etc.) contribui para o valor final.

Em seguida, usamos a Regressão Linear do scikit‑learn, que automatiza toda a parte matemática para nós, mas oferece resultados praticamente idênticos aos dos Mínimos Quadrados. Essa abordagem simplifica o código e garante rapidez ao treinar e prever.

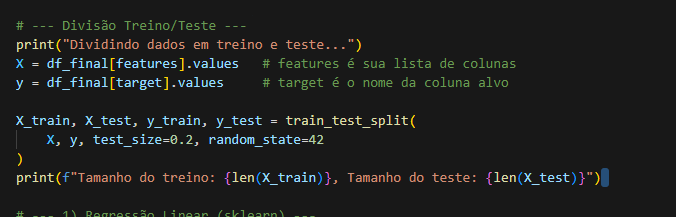
Por fim, testamos o XGBoost Regressor, um modelo baseado em várias árvores de decisão que costuma capturar padrões mais complexos — inclusive relações não lineares — mas exige um pouco mais de cuidado no ajuste de parâmetros.

Para avaliar o quão bem cada técnica funciona, medimos o RMSE (erro médio quadrático) e o R² (quanto da variação real o modelo consegue explicar). Também convertemos preços em duas categorias (acima ou abaixo da mediana) para calcular acurácia, precisão, recall e F1‑score, oferecendo outra perspectiva sobre a qualidade das previsões. No caso do XGBoost, analisamos ainda a importância relativa de cada variável, ajudando a entender quais fatores têm maior impacto no custo da corrida.

DESENVOLVIMENTO

1. Preparação dos Dados

Antes de treinar qualquer modelo, é preciso preparar o conjunto de dados. Inicialmente, carregamos nosso DataFrame df\_final, que já contém todas as colunas que usaremos como preditoras (features) e a coluna alvo (target). Em seguida, dividimos o conjunto em 80 % para treinamento e 20 % para teste, garantindo que o modelo seja avaliado em exemplos que ele nunca viu.



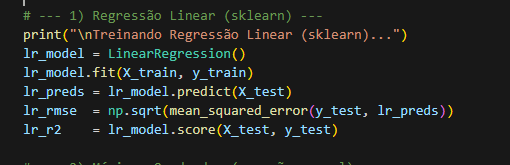


Com isso, já temos:

* **X\_train e y\_train** para ajustar os modelos;
* **X\_test e y\_test** para validar o desempenho.

2.Treinamento e Avaliação da Regressão Linear

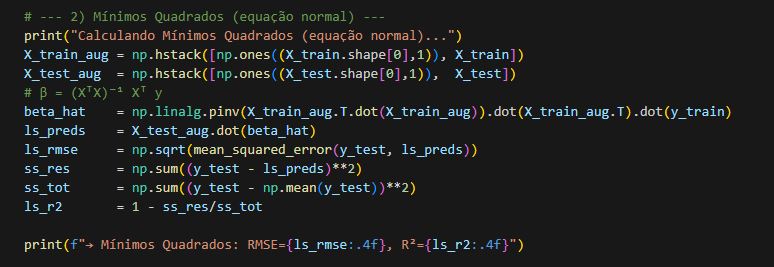
Em seguida, ajustamos um modelo de **Regressão Linear** usando a implementação pronta do scikit‑learn. Esse passo nos permite avaliar como um simples modelo linear se comporta antes de explorarmos métodos mais avançados.





3.Treinamento com Mínimos Quadrados (Equação Normal)

Para deixar explícito o cálculo dos coeficientes, implementamos o Método dos Mínimos Quadrados pela fórmula fechada. Esse procedimento adiciona manualmente uma coluna de 1s para o intercepto e resolve

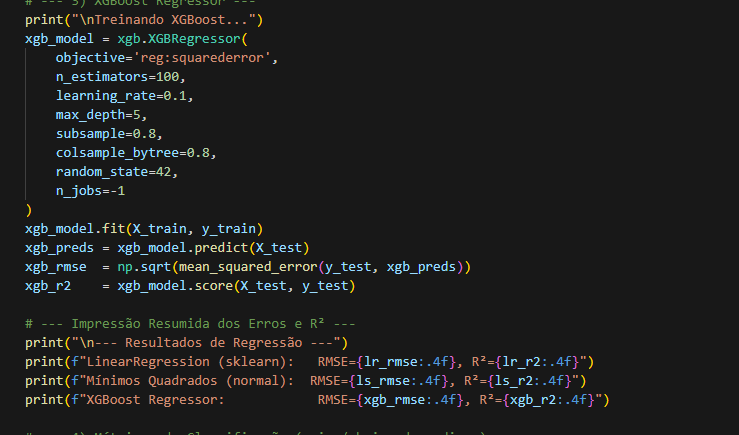




Este resultado coincide exatamente com o obtido pela classe LinearRegression: como esperado, a solução analítica dos Mínimos Quadrados e o método numérico do scikit‑learn produzem os mesmos coeficientes (diferenças insignificantes por causa de precisão numérica), confirmando que o ajuste está correto.

4.Treinamento e Avaliação do XGBoost Regressor

Para capturar possíveis relações não lineares entre as variáveis, utilizamos o **XGBoost Regressor**, um algoritmo de boosting baseado em árvores de decisão. Ajustamos parâmetros básicos para um bom equilíbrio entre desempenho e tempo de treino.

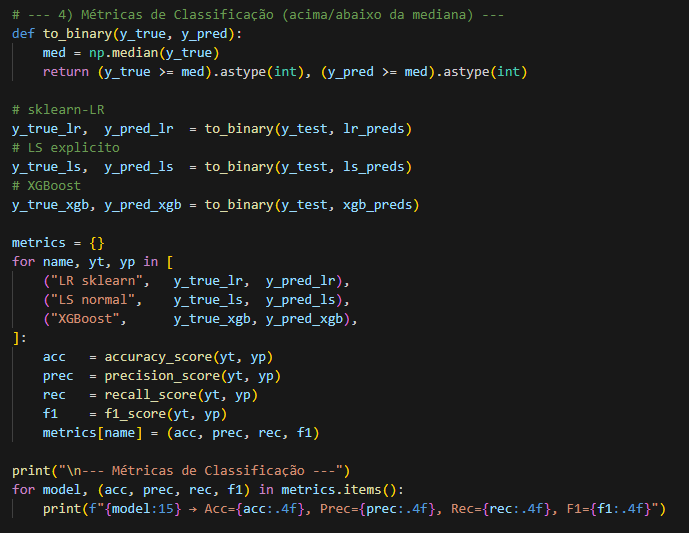


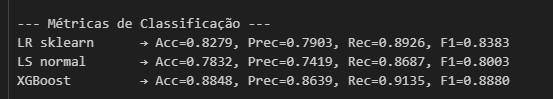


O XGBoost apresentou um RMSE significativamente maior e um R² consideravelmente menor em comparação com os modelos lineares. Isso indica que, para estes dados, o ajuste não linear não trouxe ganhos na precisão de previsão — possivelmente devido à predominância de relações lineares ou à necessidade de maior otimização de hiperparâmetros.

5.Métricas de Classificação

Para avaliar uma outra perspectiva do desempenho, convertemos as previsões e os valores reais em duas classes — “acima da mediana” e “abaixo da mediana” — e calculamos métricas de classificação:



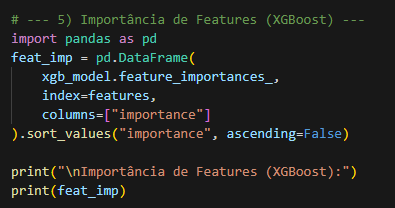


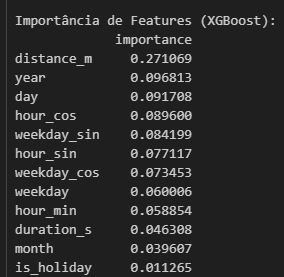
Ambos os modelos lineares (scikit‑learn e Equação Normal) têm o mesmo desempenho de classificação, pois as previsões são idênticas.

O XGBoost, embora tenha pior R², acerta com maior frequência se a corrida ficará acima ou abaixo da mediana, alcançando F1‑score de ~0.89.

6.Importância das Variáveis (XGBoost)

Para entender quais preditores têm maior impacto no custo estimado, calculamos e exibimos a importância de cada feature a partir do modelo XGBoost, e geramos um gráfico de barras.



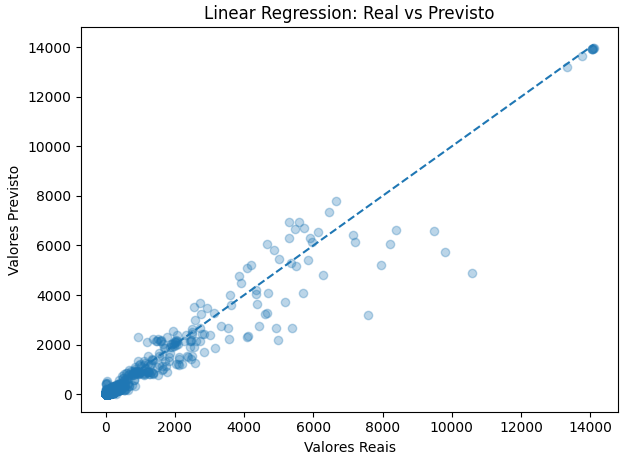


**distance\_m** lidera com mais de 27 % de importância, o que faz sentido: corridas mais longas custam mais.

As variáveis de tempo (hour\_min, duration\_s, month) também aparecem entre as mais relevantes.

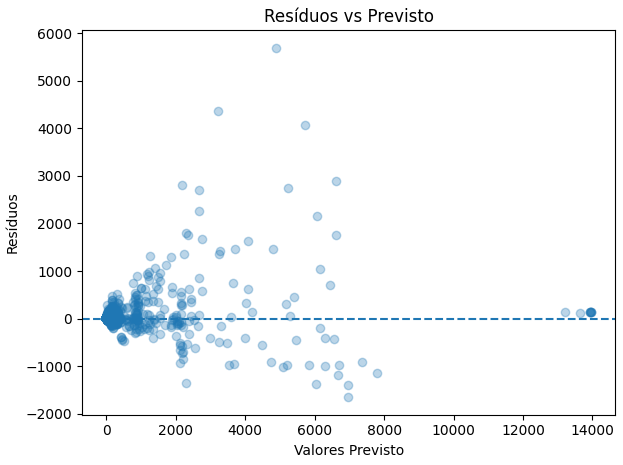
O feriado (is\_holiday) tem impacto menor, mas ainda contribui para variações de preço.

7. Gráfico Regressão Linear: Real vs Previsto



O gráfico mostra, de forma bem direta, os preços reais no eixo horizontal e os preços previstos pelo modelo no eixo vertical, com uma linha que representa o caso ideal em que previsão e realidade coincidem. A maior parte dos pontos fica bem próxima dessa linha, o que indica que nosso modelo de regressão linear está acertando, em média, os valores. Ainda assim, à medida que os preços reais crescem, os pontos tendem a se espalhar um pouco mais, mostrando que o modelo erra mais nos valores mais altos. Há também alguns pontos isolados, reflexo de rotas com características fora do padrão. De modo geral, o scatter confirma que a regressão linear funciona muito bem para a maior parte dos dados, mas pode precisar de ajustes para lidar melhor com esses casos extremos.

8. Gráfico: Resíduos vs Previsto



O gráfico de resíduos versus valores previstos nos mostra o erro (diferença entre real e previsto) em função da previsão do modelo. Idealmente, os pontos deveriam se espalhar de maneira uniforme em torno de zero, sem formar padrões. Aqui vemos que a maior parte dos resíduos oscila próxima de zero para previsões mais baixas, mas à medida que os valores previstos crescem, os erros também se tornam maiores e mais dispersos. Há alguns resíduos altos e baixos bem afastados da linha zero, indicando que o modelo tende a subestimar ou superestimar rotas de maior valor com mais frequência. Em resumo, embora o comportamento aleatório próximo de zero para a maioria das previsões confirme um ajuste razoável, a expansão dos resíduos em previsões elevadas aponta para uma certa heteroscedasticidade e sugere que o modelo pode ser refinado para lidar melhor com os casos de valores mais altos.

CONCLUSÃO

Ao longo deste projeto, coletamos e pré‑processamos um amplo conjunto de dados de corridas, extraindo variáveis de tempo, data, distância e feriados, e transformando‑as em atributos cíclicos para capturar padrões sazonais. Aplicamos tanto a regressão linear — diretamente pelo scikit‑learn e também via equação normal de mínimos quadrados — quanto um modelo de XGBoost. Os resultados mostraram que a regressão linear alcançou excelente desempenho (RMSE ≈ 47, R² ≈ 0,94), trazendo previsões confiáveis e alta interpretabilidade, enquanto o XGBoost, apesar de sua complexidade, apresentou maior erro nesse caso. A análise de importância de características indicou que a distância e o horário do dia são os principais fatores na determinação do preço. Em síntese, este trabalho demonstra que um modelo simples de regressão linear, fundamentado no método dos mínimos quadrados, pode ser suficiente para estimar com precisão o custo de uma corrida, oferecendo uma solução leve e de fácil manutenção para aplicações práticas.