Código-fonte: https://colab.research.google.com/drive/15TR9WOSjg9Ke3MruOln4-Ot4ljLvH8gB?usp=sharing

Análise de modelos:

Foi realizada a análise de cinco diferentes modelos, para ver qual tinha melhor performance diante de nossos dados. Para avaliá-los, usamos as seguintes métricas:

- Acurácia: essa métrica verifica, a partir da base de teste e do treino que realizamos, quantos dos y presentes no teste o modelo foi capaz de acertar.
- MAE: quantifica o quanto em média o modelo erra em cada previsão.
- MSE: é a média de cada erro ao quadrado, penalizando erros maiores.
- RMSE: é a raiz do MSE, para entendermos a média de fato dos erros.
- R^2: Mede quanto da variabilidade dos dados o modelo consegue explicar.

**Não usamos as métricas F1-Score, Recall e precisão, por se tratar de um modelo treinado com variáveis contínuas (preço) e não com variáveis discretas (0 e 1).

```
 ====== GradientBoost =======
Acuracia: 87.55%
MAE: 7.03
MSE: 173.09
RMSE: 13.16
R²: 0.88

 ====== Regressão Linear =======
Acuracia: 59.33%
MAE: 14.59
MSE: 565.27
RMSE: 23.78
R²: 0.59

 ====== DecisionTree =======
Acuracia: 93.41%
MAE: 3.20
MSE: 91.58
RMSE: 9.57
R²: 0.93

 ====== KNN =======
Acuracia: 86.47%
MAE: 7.28
MSE: 188.02
RMSE: 13.71
R²: 0.86

 ====== RandomForest =======
Acuracia: 95.20%
MAE: 3.25
MSE: 66.66
RMSE: 8.16
R²: 0.95
```

Após a análise, escolhemos como melhor modelo e o qual será usado para o trabalho, por enquanto, o RandomForestRegressor, por possuir os melhores resultados diante dos modelos analisados.

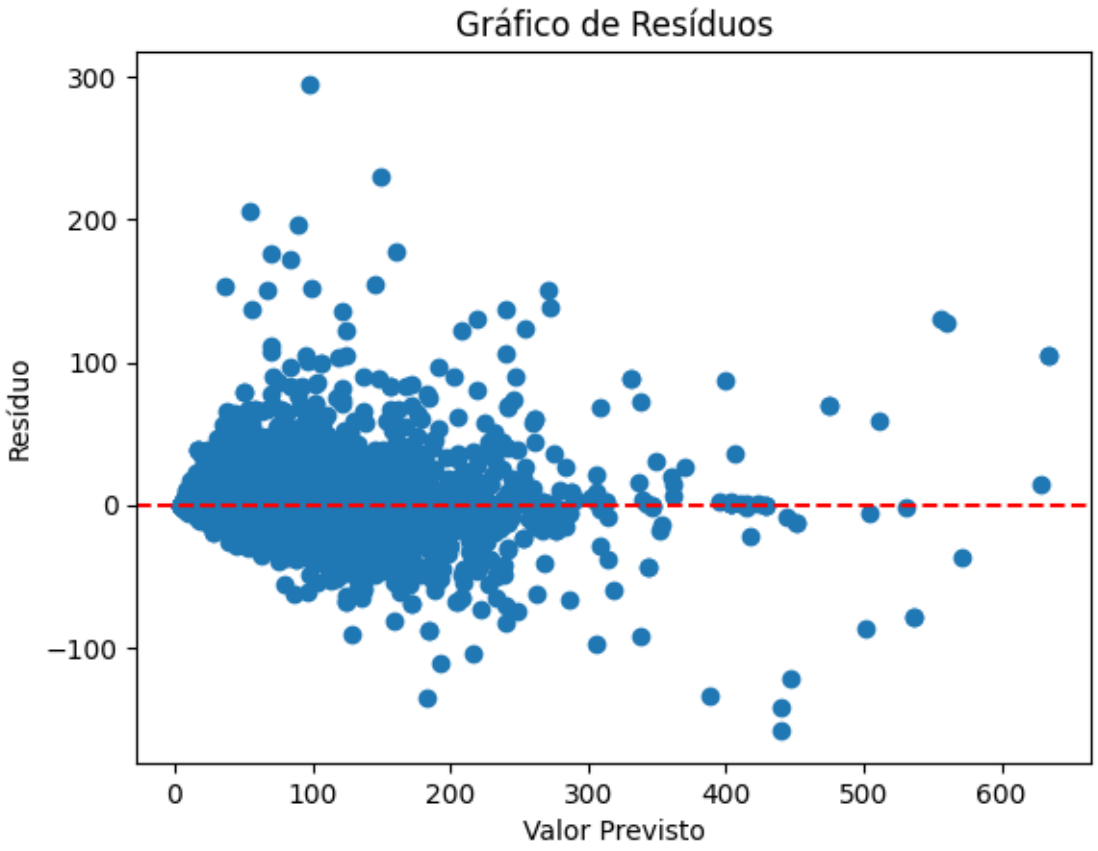
```
modelo = RandomForestRegressor(n_estimators=100)
x = df[['distancia_m', 'tempo_estim_minutos', 'ProductID_encoded', 'dia', 'mes', 'e_fim_de_semana']]
y = df['Price']
treino_x, teste_x, treino_y, teste_y = train_test_split(x, y, test_size=0.3, random_state=42)
modelo.fit(treino_x, treino_y)
```

Para isso, separamos a base em 70% para treino e 30% para teste, e usamos as seguintes colunas para o treino do modelo: distância da corrida, tempo da corrida, o id do produto, o dia que ocorreu a corrida, o mês que ocorreu, e se era final de semana ou não. E o dado que queremos predizer é o preço.

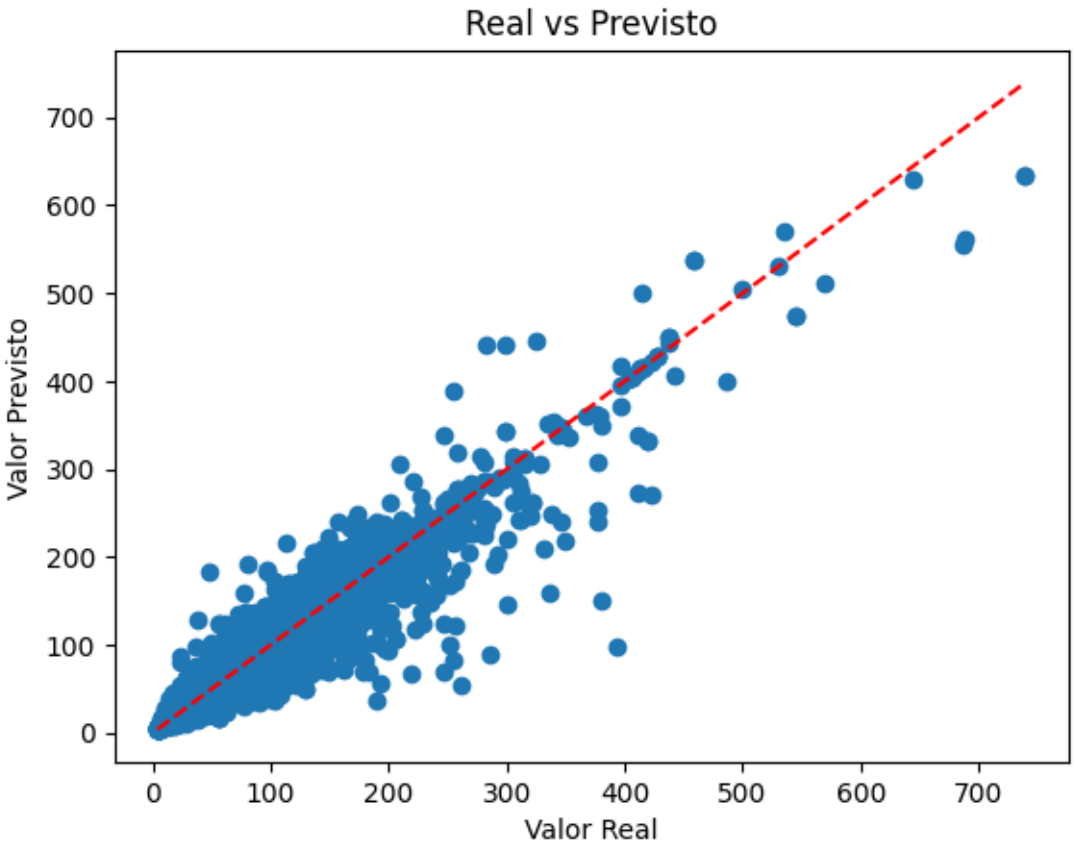
Com o restante da base, testamos e validamos o modelo, obtendo os seguintes resultados:

```
Acuracia: 95.19%
MAE: 3.24
MSE: 66.87
RMSE: 8.18
R²: 0.95
```

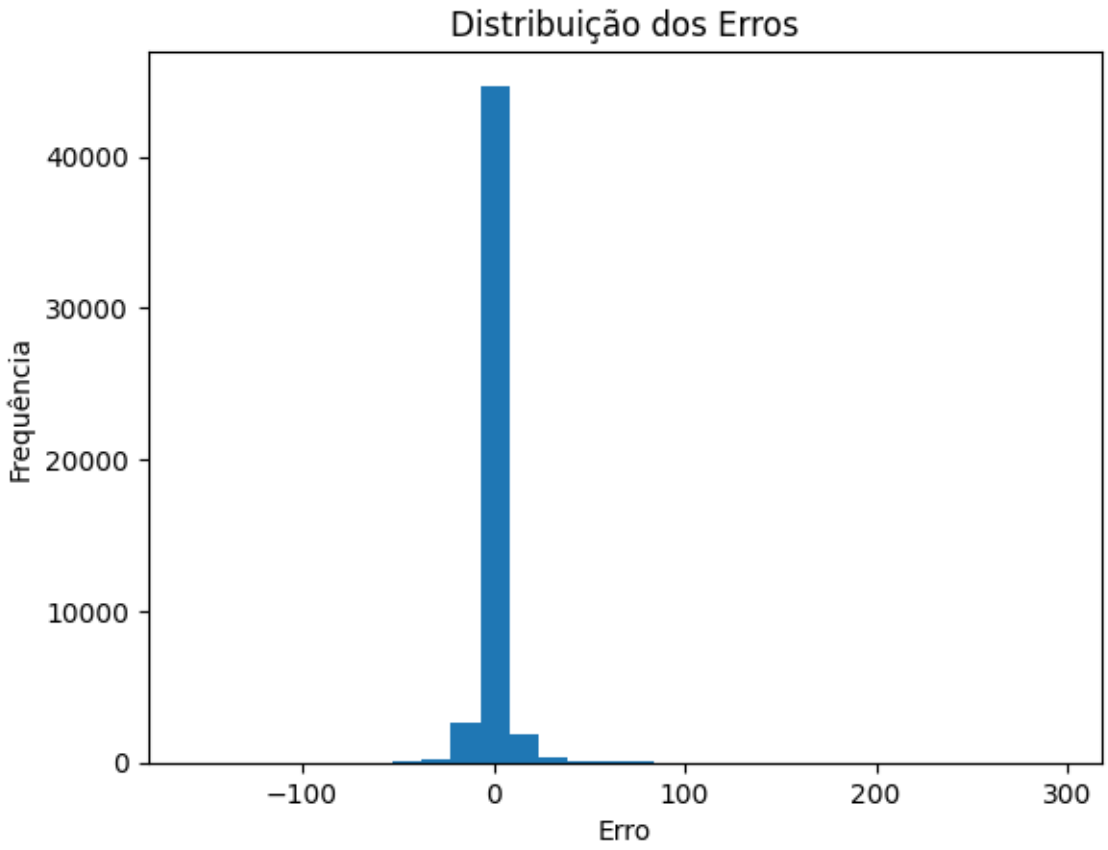
Gráficos de desempenho:



O gráfico acima analisa o quanto, em cada previsão, o modelo errou, para mais ou para menos. Quando está para baixo da linha, quer dizer que o modelo previso um preço superior ao real, e quando está acima, ele previu um preço abaixo do real. Em grande maioria, o modelo errou menos quando o valor real da corrida é mais baixo, mais especificamente, menos que R$ 300.



O gráfico acima evidencia mais o que foi analisado anteriormente, que para corridas abaixo de R$ 300 o modelo tende a se aproximar mais do valor real da corrida.



O gráfico acima mostra a frequência de erros, mostrando que, por conta de uma acurácia muito alta, a tendência é que a maior frequência se concentre em 0.