

## Faça o que eu fiz na aula

Para treinar o nosso modelo, vamos remover o fullVisitorId da nossa base.

```
visitas.drop('fullVisitorId', axis=1, inplace=True)
```

Separar variáveis preditoras e variável resposta.

```
y = visitas.transactionRevenue.copy()
X = visitas.drop('transactionRevenue', axis=1)
```

Tratamento das variáveis categóricas com LabelEncoder.

```
quali = visitas.dtypes[visitas.dtypes == object].keys()
from sklearn.preprocessing import LabelEncoder

for coluna in quali:
    lbl = LabelEncoder()
    strings = list(X[coluna].values.astype('str'))
    lbl.fit(strings)
    X[coluna] = lbl.transform(strings)
```

Separação em treino e teste.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Treinamento e predição de uma regressão linear.

```
reg = LinearRegression()
reg.fit(X_train, y_train)
reg_predict = reg.predict(X_test)
reg_predict[reg_predict < 0] = 0
```

Criação do dataframe para avaliação dos resultados.

```
resultados = pd.DataFrame()
resultados['revenue'] = y_test
resultados['predict'] = reg_predict
resultados['erro'] = reg_predict - y_test
resultados.head()
```

|      | revenue | predict  | erro     |
|------|---------|----------|----------|
| 4122 | 0.0     | 0.081534 | 0.081534 |
| 4065 | 0.0     | 0.000000 | 0.000000 |
| 1731 | 0.0     | 0.000000 | 0.000000 |
| 4740 | 0.0     | 0.478034 | 0.478034 |
| 6391 | 0.0     | 0.000000 | 0.000000 |

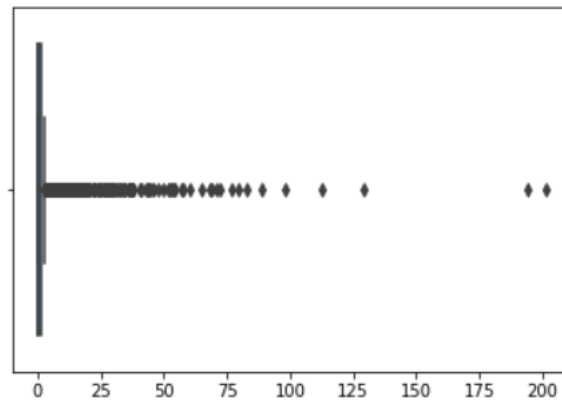
Cálculo do RMSE.

```
np.sqrt(mean_squared_error(y_test,reg_predict))
```

25,0196644

Boxplot das previsões.

```
sns.boxplot(reg_predict)
```



Treinamento do Gradient Boosting.

```
from sklearn.ensemble import GradientBoostingRegressor
gb = GradientBoostingRegressor(random_state=42)
gb.fit(X_train,y_train)
gb_predict = gb.predict(X_test)
gb_predict[gb_predict < 0 ] = 0
gb_predict
```

```
array([0.02242292, 0.          , ..., 0.32247215, 0.          ,
       0.          ])
```

Criação de um dataframe para avaliação dos resultados do Gradient Boosting.

```
resultados = pd.DataFrame()
resultados['revenue'] = y_test
resultados['predict'] = gb_predict
resultados['erro'] = gb_predict - y_test
resultados.head()
```

|      | revenue | predict  | erro     |
|------|---------|----------|----------|
| 4122 | 0.0     | 0.022423 | 0.022423 |
| 4065 | 0.0     | 0.000000 | 0.000000 |
| 1731 | 0.0     | 0.000000 | 0.000000 |
| 4740 | 0.0     | 0.000000 | 0.000000 |
| 6391 | 0.0     | 0.022423 | 0.022423 |

Calcular o RMSE.

```
np.sqrt(mean_squared_error(y_test,gb_predict))
```

23,659570