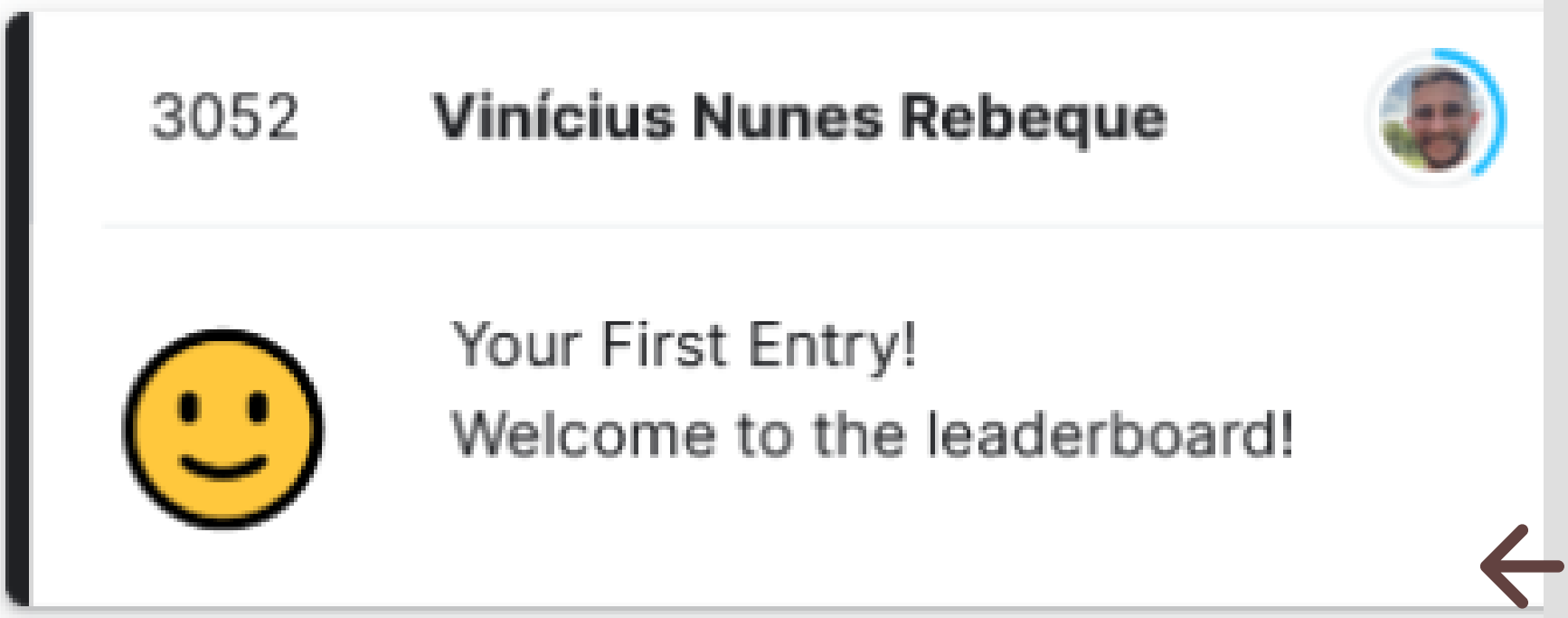
Machine Learning Competitvo

Como criei meu primeiro modelo de predição usando dados públicos do Kaggle e me inscrevi na competição da comunidade

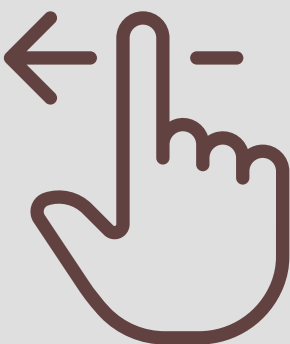
Agosto/2022



Entradas utilizadas:

- Qualidade geral do material e acabamento da casa;
- Tamanho total da área construida;
- Metros quadrados do primeiro andar;
- Tamanho total da sala de estar;
- Número de carros que a garagem suporta;
- Tamanho da garagem.

Mas por quê essas entradas?



Usei o **método Pearson** de **correlação linear** para criar uma matriz de correlação.

$$\frac{\operatorname{cov}(X,Y)}{\sqrt{\operatorname{var}(X)\cdot\operatorname{var}(Y)}}$$

A matriz nos mostra quais as variáveis tem mais **correlação entre si** e podem nos **aproximar do objetivo**.

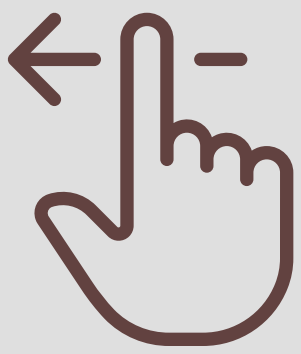


gráfico de correlação

Cores **mais fortes** significam **maior correlação**. As características estão plotadas nos eixos do gráfico.

Teste no seu dataset:

```
corr = df.corr()
corr
```

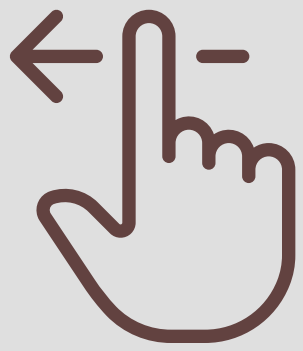
Out[17]:

	Id	MSSubClass
Id	1.000000	0.011156
MSSubClass	0.011156	1.000000
LotFrontage	-0.010601	-0.386347
LotArea	-0.033226	-0.139781
OverallQual	-0.028365	0.032628

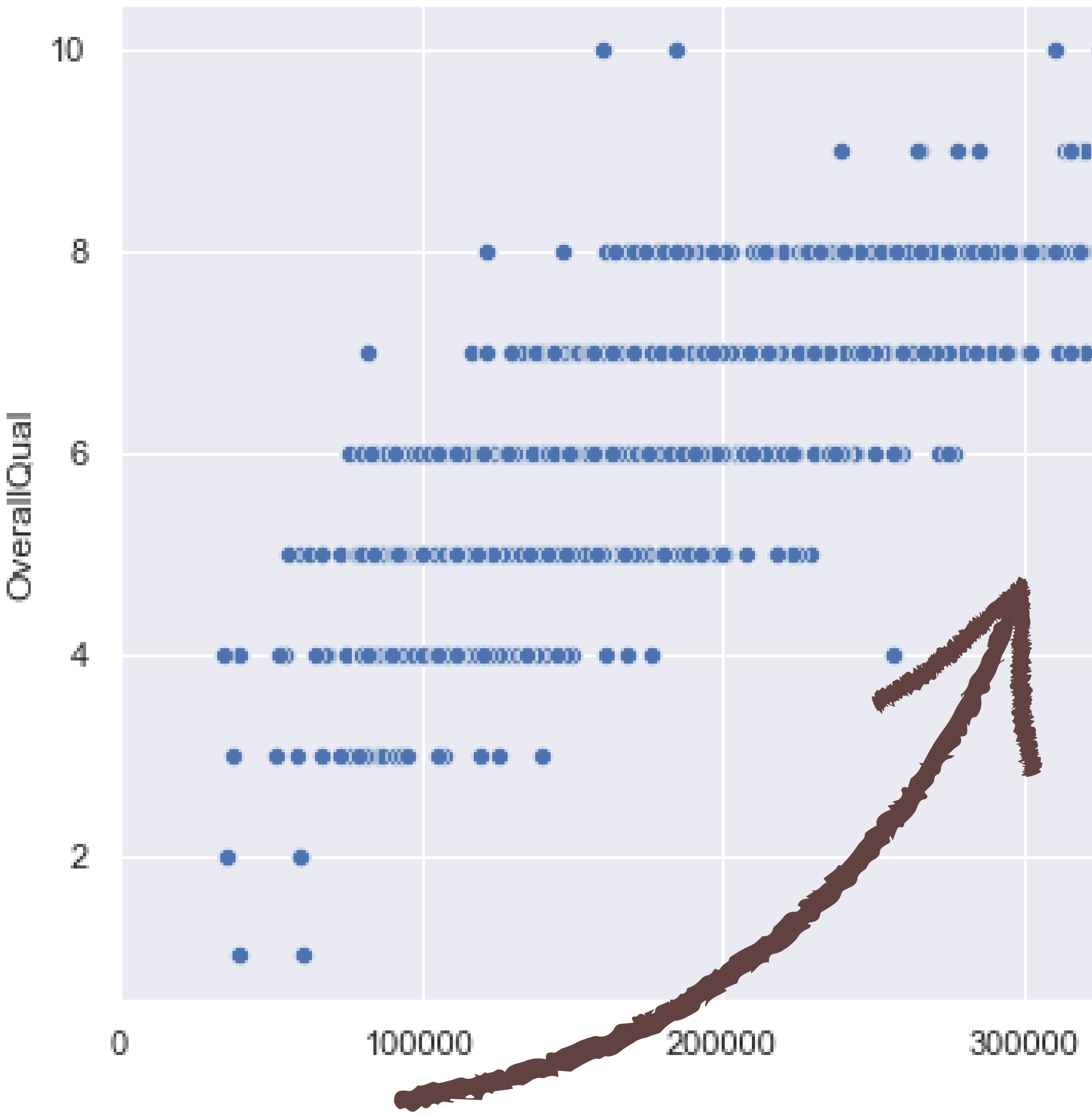
```
plt.figure(figsize=(10,10))
ax = sns.heatmap(corr,
                 annot=False,
                 linewidths=.5,
                 cmap='YlGnBu')
```

Nosso **objetivo** está na coluna '**SalePrice**'.

Tudo que temos que fazer é **verificar onde há maior correlação** entre SalePrice e as **outras colunas**.

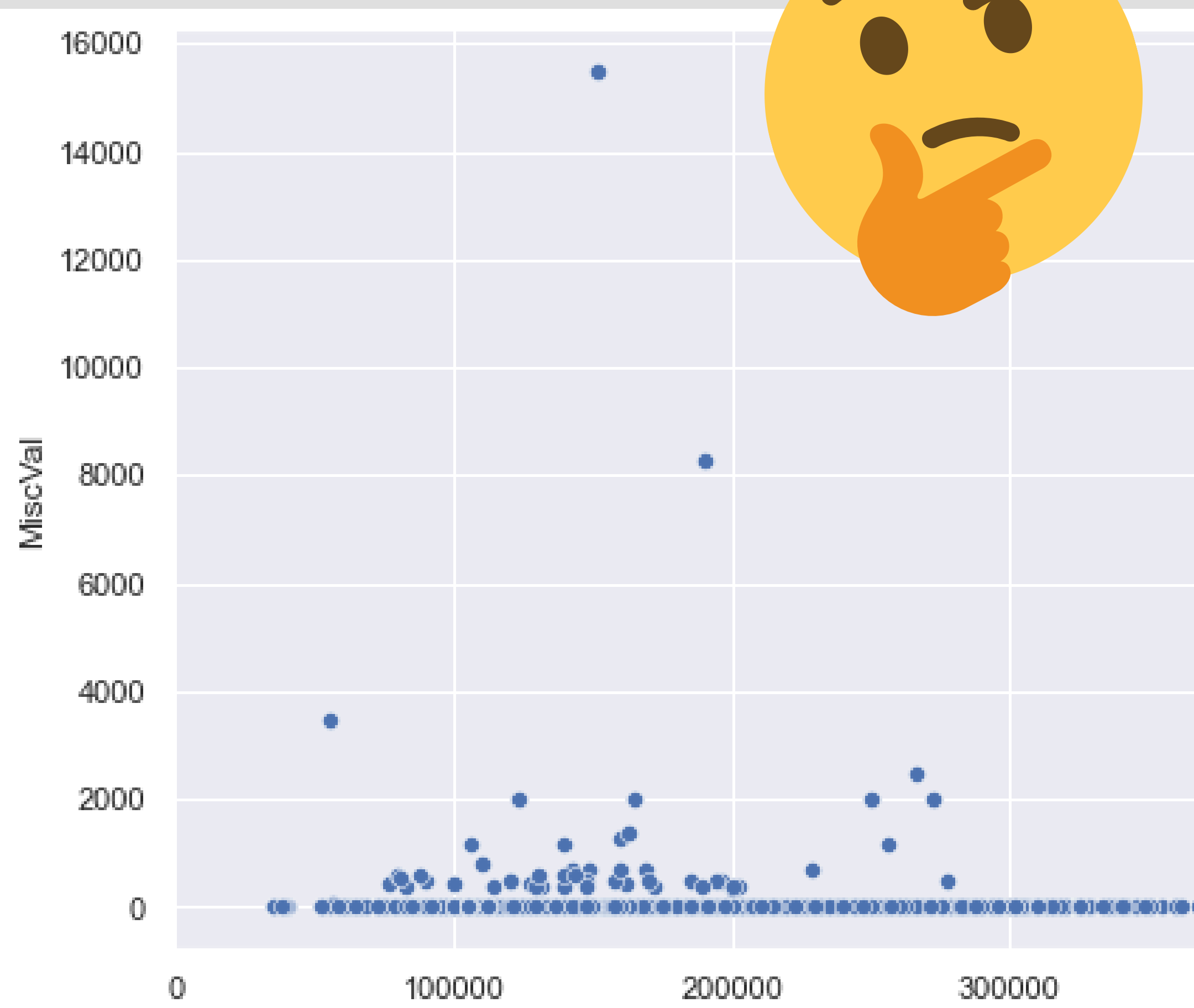


Exemplo 1: Feature COM correlação clara.



Qualidade Geral vs. Preço de Venda

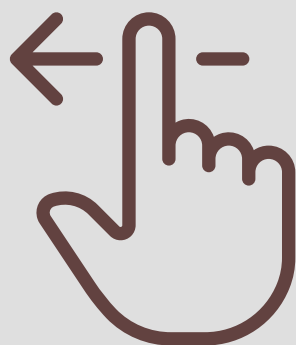
Exemplo 2: Feature SEM correlação clara.



Valor de 'Opcionais' vs. Preço de Venda

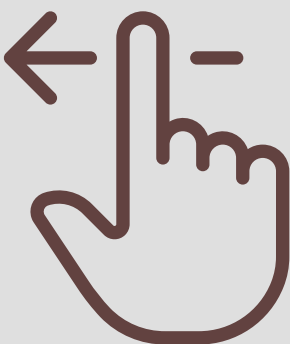
Cross validation

Agora que já sabemos as features chave vamos **dividir** nossos dados em **treino** e **teste**.



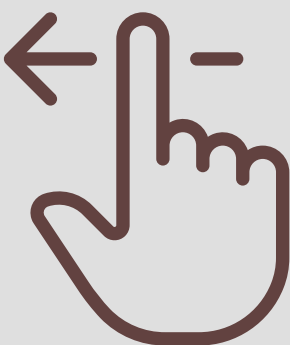
Pra quê?!

A ideia é separar uma porcentagem dos dados para validar se ele aprendeu mesmo ou só decorou.



O **método** que vamos usar **apresentará os dados** ao modelo e ele vai **aprender a relação** entre as variáveis e o objetivo target '**SalePrice**'.

É impossível **validá-lo** com os mesmos dados que ele **aprendeu**. Por isso vamos fazer essa **divisão**.

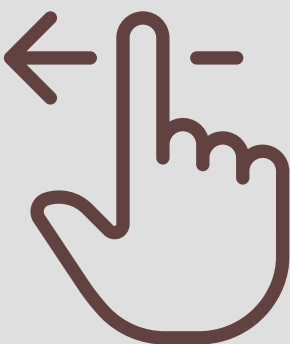


Imagine passar uma prova com o gabarito no verso.

Não queremos que nosso modelo entenda esses dados em si e sim a relação entre eles.

Queremos sempre usar novos dados.

Os antigos nós já sabemos o valor.



A aplicação prática desse modelo pode ser um corretor de imóveis, por exemplo.

Ele insere os dados do imóvel e o algorítmo calcula de maneira estatística o valor desse imóvel.

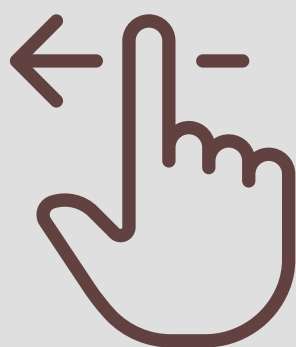
Mãos à obra!

```
# Feature arrange
df = df[['OverallQual',
         'TotalBsmtSF',
         '1stFlrSF',
         'GrLivArea',
         'GarageCars',
         'GarageArea']]
```

```
if df.isnull().any() is True:
    print('Temos valores ausentes')

else:
    print('Estamos prontos para ir ;-)')
```

```
Estamos prontos para ir ;-)
```



Definindo X e y.
X são os dados de entrada.
y o objetivo.

```
y = df_backup1.SalePrice
X = df


print(X.columns)
print('\n----\n')
print(y)
print('\n----')
X.describe()
```

```
Index(['OverallQual', 'TotalBs
       'GarageArea'],
      dtype='object')


----


0          208500
1          181500
2          223500
3          140000
4          250000
```

Divisão em Treino e Teste

```
train_X, val_X,
train_y, val_y =
train_test_split(X, y, random_state=1)
```

Treinamento e Validação

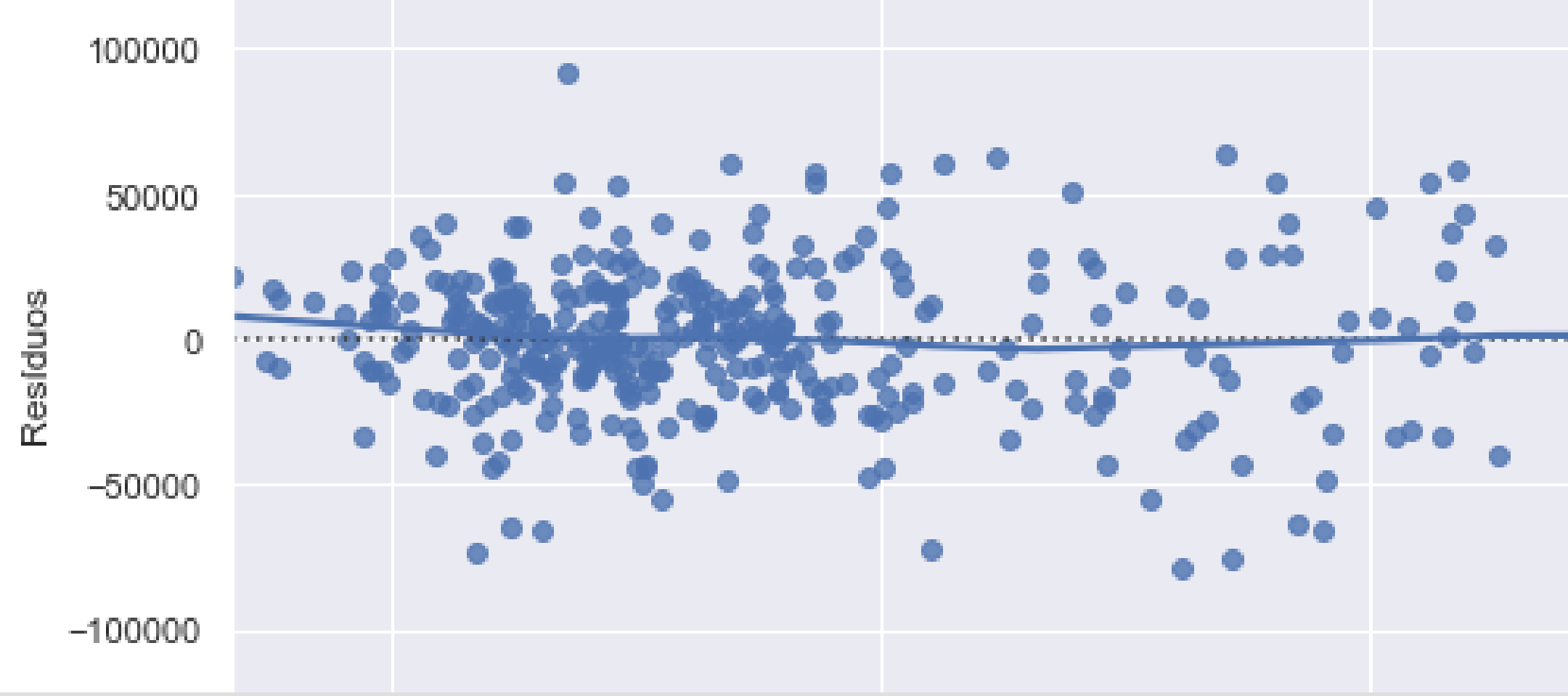
```
# Criando a versão 2 do modelo
modelo_v2 = RandomForestRegressor(random_state=1)

# Treinando modelo v2
modelo_v2.fit(train_X, train_y)

# Validação do modelo v2
modelo_v2_preds = modelo_v2.predict(val_X)

modelo_v2_mae = mean_absolute_error(val_y, modelo_v2_preds)
print(modelo_v2_mae)
print('O erro médio absoluto do modelo v2 (RandomForestRegre
```

Verificação de Performance



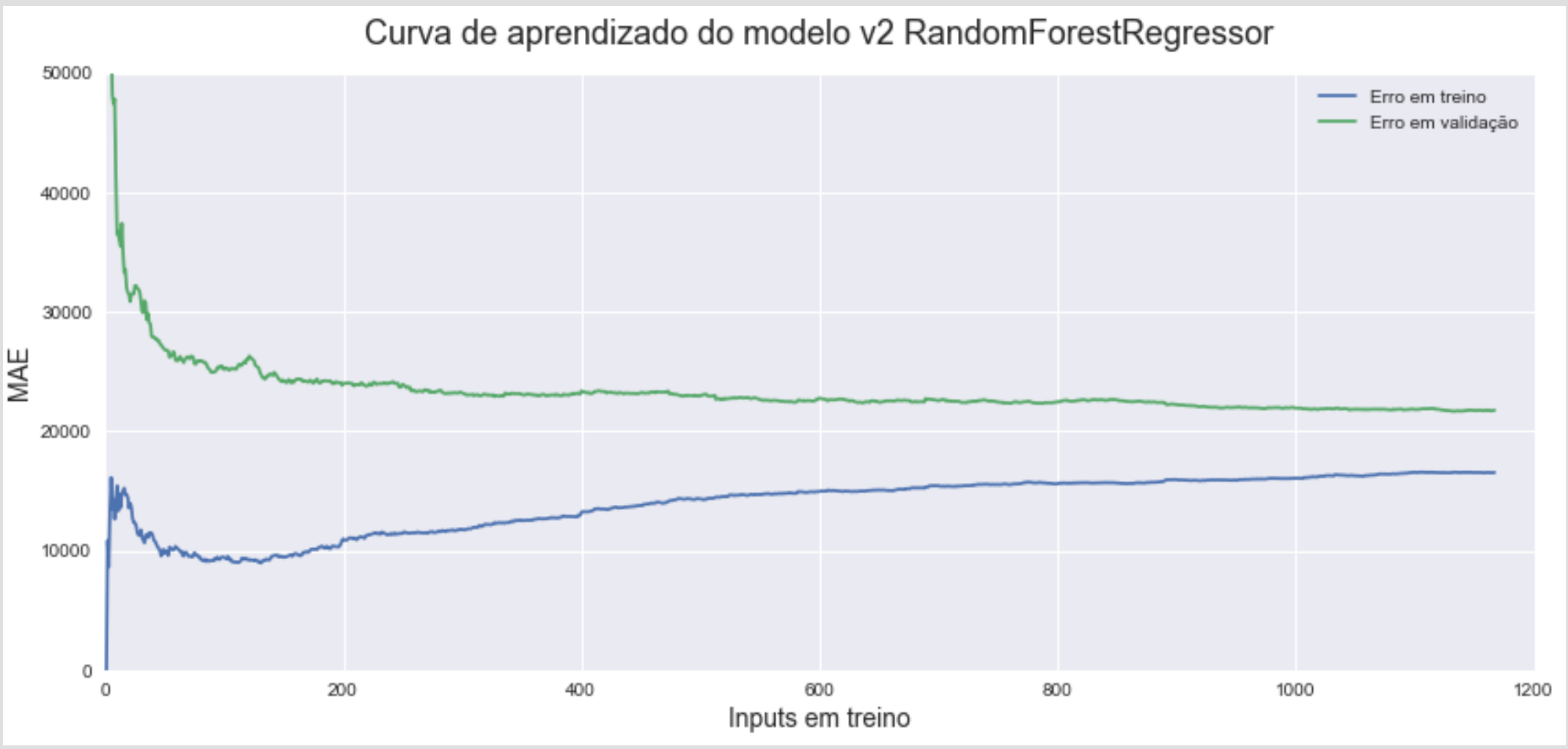
Buscando parâmetros para aprimoramento da perform.

```
def calcular_maev2(max_leaf_nodes, train_X, val_X, train_y, val_y):
    modelo_v2 = RandomForestRegressor(max_leaf_nodes=max_leaf_nodes
    modelo_v2.fit(train_X, train_y)
    modelo_v2_preds = modelo_v2.predict(val_X)
    modelo_v2_mae = mean_absolute_error(val_y, modelo_v2_preds)
    return(modelo_v2_mae)


for max_leaf_nodes in [5, 50, 500, 5000]:
    range_maev2 = calcular_maev2(max_leaf_nodes, train_X, val_X, tr
    print("Max leaf nodes: %d  \t\t Erro Médio Absoluto:  %d" %(max
```

```
Max leaf nodes: 5                Erro Médio Absoluto:  28392
Max leaf nodes: 50               Erro Médio Absoluto:  20168
Max leaf nodes: 500              Erro Médio Absoluto:  20483
Max leaf nodes: 5000             Erro Médio Absoluto:  20482
```

Verificando a curva de aprendizagem

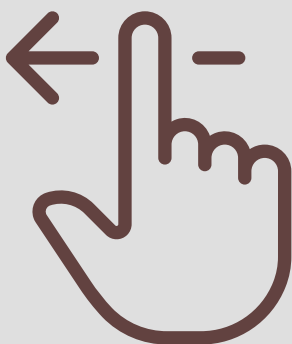


Resultados

Resumindo, em **validação** nosso modelo **atingiu** uma **precisão** de 82%. Existem várias formas de aprimorar esse valor.

Leu até aqui?

Então vai lá ver o arquivo completo e detalhado de todo esse processo. Esse PDF é só pra chamar sua atenção!



github.com/Rebeque

linkedin.com/in/vRebeque

Sua opinião é bem vinda!

VINÍCIUS NUNES REBEQUE

ANALYTICS | TECH FOR BI