

# Regressão Linear

## Prática 02: Predizendo o preço das casas

Prof<sup>a</sup> Deborah Magalhães

Monitor: Davi Luis de Oliveira



UNIVERSIDADE  
FEDERAL DO PIAUÍ



## **Curso: Bacharelado em Sistema de Informação**

Disciplina: Sistemas Inteligentes

### ▷ **Predição com RL**

Você pode me encontrar em [deborah.vm@gmail.com](mailto:deborah.vm@gmail.com)

(Dúvidas e sugestões serão bem-vindas =D)

## Passo 1: Baixar o dataset

```
# fazer o download do dataset home_data.gl.zip  
$ git pull https://github.com/daviluis321/Sistemas-Inteligentes.git
```

## Passo 2: Abrir notebook

```
$ source activate gl-env  
$ jupyter notebook
```

```
deborah@deborah-Lenovo-ideapad-300S-14ISK:~/Sistemas Inteligentes/Regressão/Pratica 01$ jupyter notebook  
[I 09:54:06.321 NotebookApp] Serving notebooks from local directory: /home/deborah/Sistemas Inteligentes/Regressão/Pratica 01  
[I 09:54:06.321 NotebookApp] 0 active kernels  
[I 09:54:06.321 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/  
[I 09:54:06.322 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

5

## Passo 2: Abrir notebook

 jupyter

Files

Running

Clusters

Select items to perform actions on them.

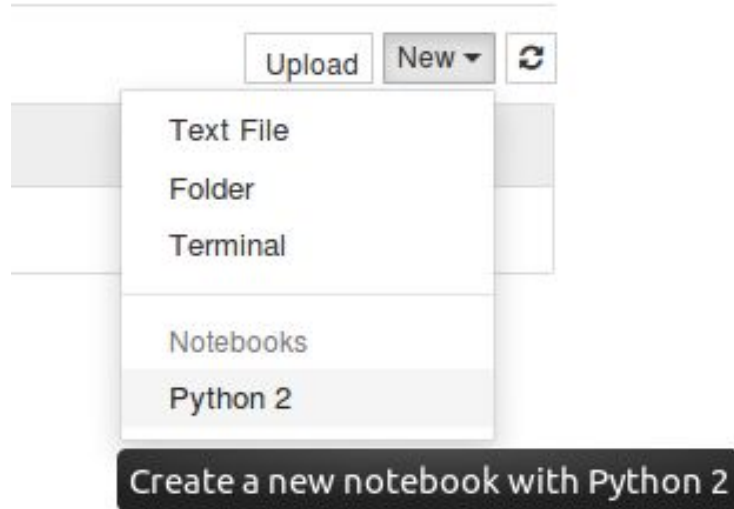
Upload

New ▾



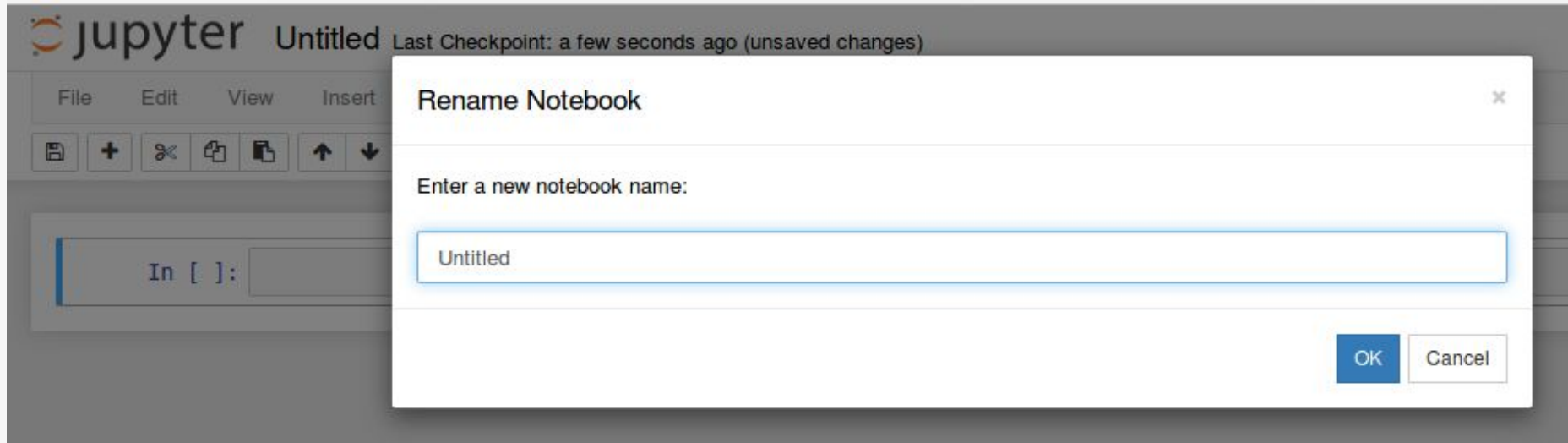
home\_data.gl

## Passo 3: Criar notebook



7

## Passo 4: Renomear notebook



## Passo 5: Carregar os dados

```
In [1]: import graphlab
```

### Carregar os dados

```
In [4]: vendas = graphlab.SFrame("home_data.gl/")
```

This non-commercial license of GraphLab Create for academic use is assigned to deborah.vm@gmail.com and will expire on January 26, 2019.

[INFO] graphlab.cython.cy\_server: GraphLab Create v2.1 started. Logging: /tmp/graphlab\_server\_1522339428.log



## Passo 6: Visualizar os dados

In [5]: vendas

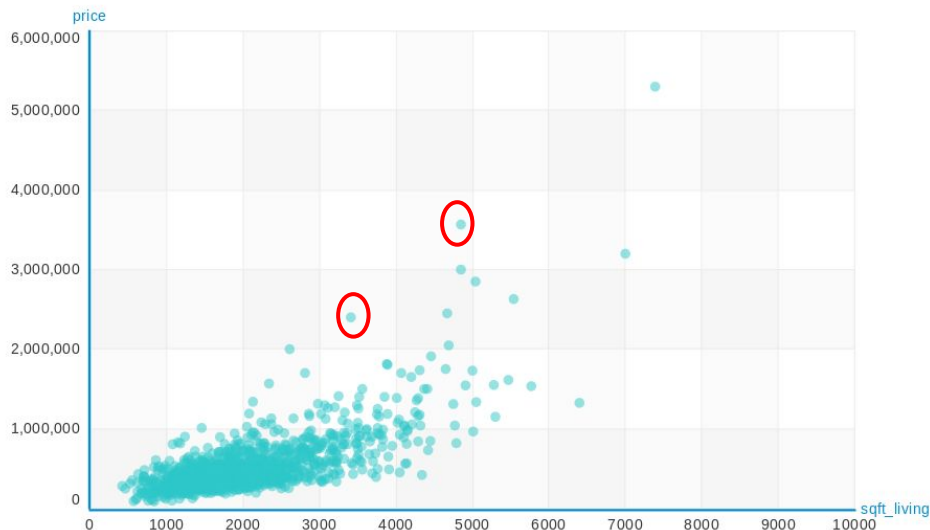
Out[5]:

id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors
7129300520	2014-10-13 00:00:00+00:00	221900	3	1	1180	5650	1
6414100192	2014-12-09 00:00:00+00:00	538000	3	2.25	2570	7242	2
5631500400	2015-02-25 00:00:00+00:00	180000	2	1	770	10000	1
2487200875	2014-12-09 00:00:00+00:00	604000	4	3	1960	5000	1
1954400510	2015-02-18 00:00:00+00:00	510000	3	2	1680	8080	1
7237550310	2014-05-12 00:00:00+00:00	1225000	4	4.5	5420	101930	1
1321400060	2014-06-27 00:00:00+00:00	257500	3	2.25	1715	6819	2
2008000270	2015-01-15 00:00:00+00:00	291850	3	1.5	1060	9711	1
2414600126	2015-04-15 00:00:00+00:00	229500	3	1	1780	7470	1

# Passo 6: Visualizar os dados

## Visualização dos Dados

```
In [16]: graphlab.canvas.set_target("ipynb")  
vendas["sqft_living", "price"].show(view="Scatter Plot")
```



# Passo 7: Divisão treino/teste

## Divisão treino/teste

```
In [18]: vendas_treino, vendas_teste = vendas.random_split(.8)
```

## Passo 8: Construindo o modelo

```
vendas_model = graphlab.linear_regression.create(vendas_treino,  
target='price', features=['sqft_living'])
```

# Passo 9: Avaliando o modelo

## Avaliar o modelo

```
In [15]: print vendas_teste["price"].mean()
```

```
545521.140795
```

```
In [16]: print vendas_model.evaluate(vendas_teste)
```

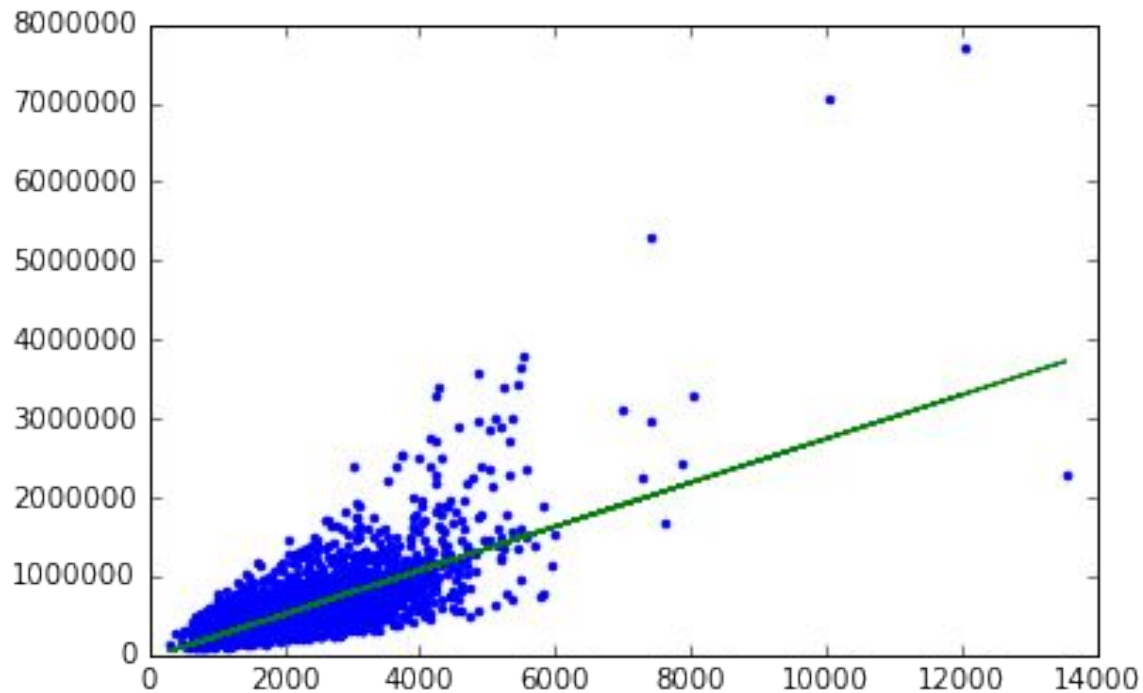
```
{'max error': 4389959.965411294, 'rmse': 278330.74677213107}
```

## Passo 10: Visualizando o modelo

```
import matplotlib.pyplot as plt  
%matplotlib inline
```

```
plt.plot(vendas_teste["sqft_living"], vendas_teste["price"], '.',  
vendas_teste["sqft_living"], vendas_model.predict(vendas_teste), '-')
```

## Passo 10: Visualizando o modelo



# Passo 11: Capturando os coeficientes do modelo

## Capturando os coeficientes do modelo

```
In [22]: vendas_model.get('coefficients')
```

```
Out[22]:
```

name	index	value	stderr
(intercept)	None	-37962.1561098	4985.53257957
sqft_living	None	277.842505452	2.19809468078

```
[2 rows x 4 columns]
```











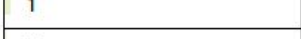
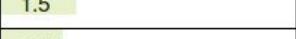




# Passo 12: Explorando outras características dos dados

## Explorando outras características das casas

```
In [24]: caracteristicas = ['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'zipcode']
```

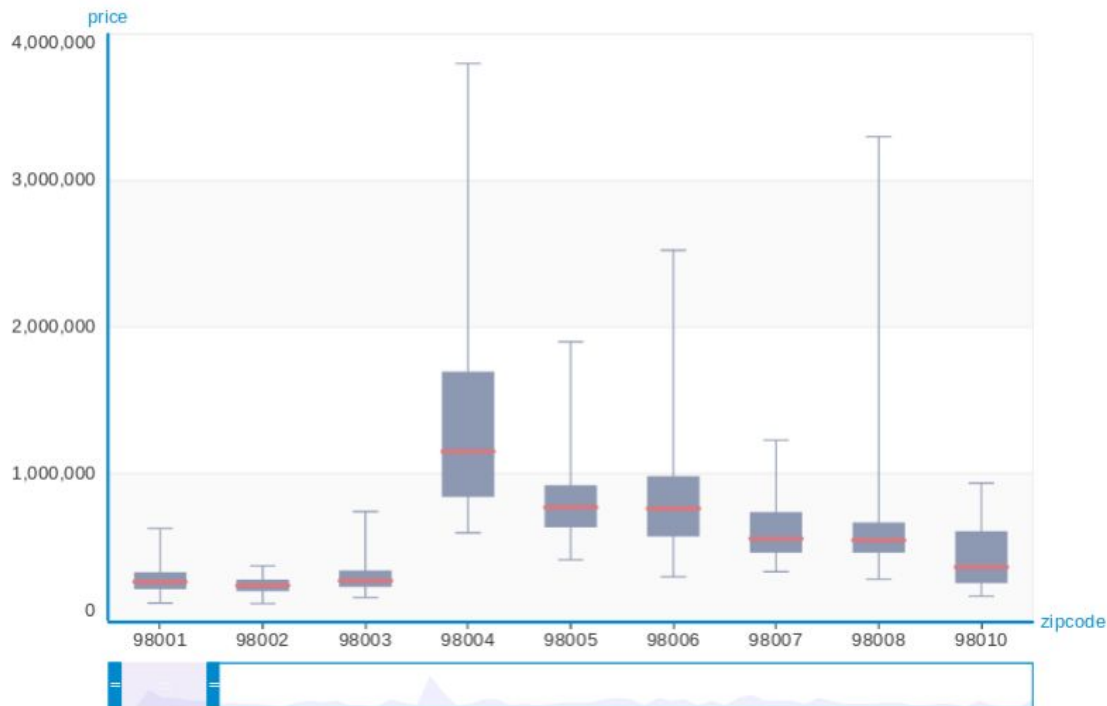
```
In [25]: vendas[caracteristicas].show()
```

bedrooms		bathrooms		sqft_living		sqft_lot	
dtype:	str	dtype:	str	dtype:	int	dtype:	int
num_unique (est.):	13	num_unique (est.):	30	num_unique (est.):	1,036	num_unique (est.):	9,74
num_undefined:	0	num_undefined:	0	num_undefined:	0	num_undefined:	0
frequent items:		frequent items:		min:	290	min:	520
				max:	13,540	max:	1,65
				median:	1,910	median:	7,61
				mean:	2,079.9	mean:	15,1
				std:	918.42	std:	41,4
				distribution of values:		distribution of values:	
							

## Passo 12: Explorando outras características dos dados

```
vendas.show(view='BoxWhisker Plot', x='zipcode', y='price')
```

## Passo 12: Explorando outras características dos dados



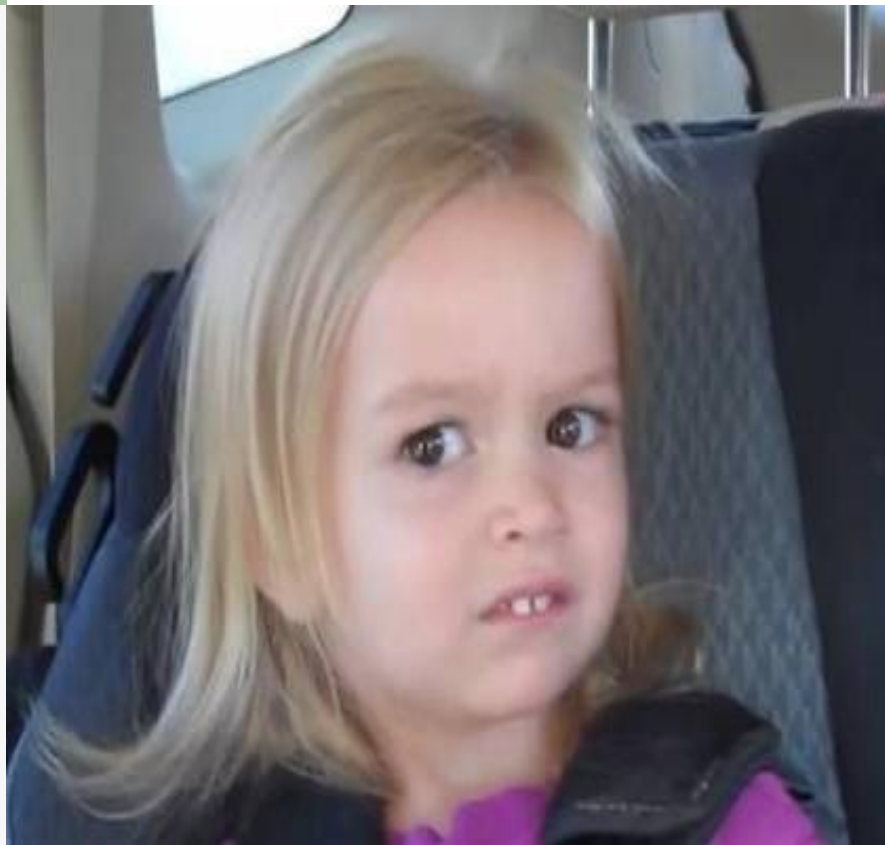
## Passo 13: Construindo o modelo com mais características

```
vendas_model_plus = graphlab.linear_regression.create(vendas_treino,  
target='price', features=caracteristicas)
```

## Passo 13: Construindo o modelo com mais características

```
In [30]: print vendas_model.evaluate(vendas_teste)
         print vendas_model_plus.evaluate(vendas_teste)

{'max_error': 4389959.965411294, 'rmse': 278330.74677213107}
{'max_error': 4468535.609018347, 'rmse': 202500.26054778424}
```



**Dúvidas? Sugestões?  
Inquietações?  
Aconselhamentos?**

- ▶ Desabafe em:  
[deborah.vm@gmail.com](mailto:deborah.vm@gmail.com)