Data Science

Documentation

Regression-Primer

January 12, 2018

Eicker Niklas, Halastra Szymon

Contents

1 Introduction

For the following tasks we will use Graphlab and Numpy to read and analize data about sold houses. With several different models we will predict prices or other features of houses.

1.1 Tasks

- Load house sales data: kc_house_data.csv.zip
- What is the content, could you read it? do you understand collumns?
- Explore the data for housing
  - make scatter plot of selected features
  - create simple regression model of sqft_living to price
  - evaluate a simple model
  - is linear function good enough? try quadratic polynomial
- Split your data into training sample and test sample
  - what is trainign error and testing error of your model?
  - predict the house price for a given sqft_living
  - predict the sqft_living for a given price of the house
  - add more feaures
  - is the model better now?
  - maybe using range of data would work better?
  - predict house price for a house id = 5309101299 what is this house like?
  - predict house price for a house id = 1925069082

1.2 Dataset

We have 21613 rows of the following data given in the dataset:

Feature	Type
id	int
date	str
price	float
bedrooms	int
bathrooms	float
sqft_living	int
sqft_lot	int
floors	float
waterfront	int
view	int
condition	int
grade	int
sqft_above	int
sqft_basement	int
yr_built	int
yr_renovated	int
zipcode	int
lat	float
long	float
sqft_living15	int
sqft_lot15	int
const	float
sqft_living_sq	float

The houses are in Seattle, as can be seen with the zipcodes and were sold between 2014 and 2015, thus the data is quite new.

2 Simple regression models

For the beginning we concentrated on the features *price* and *sqft_living*. The distribution of these two features can be seen in figure 2.1.

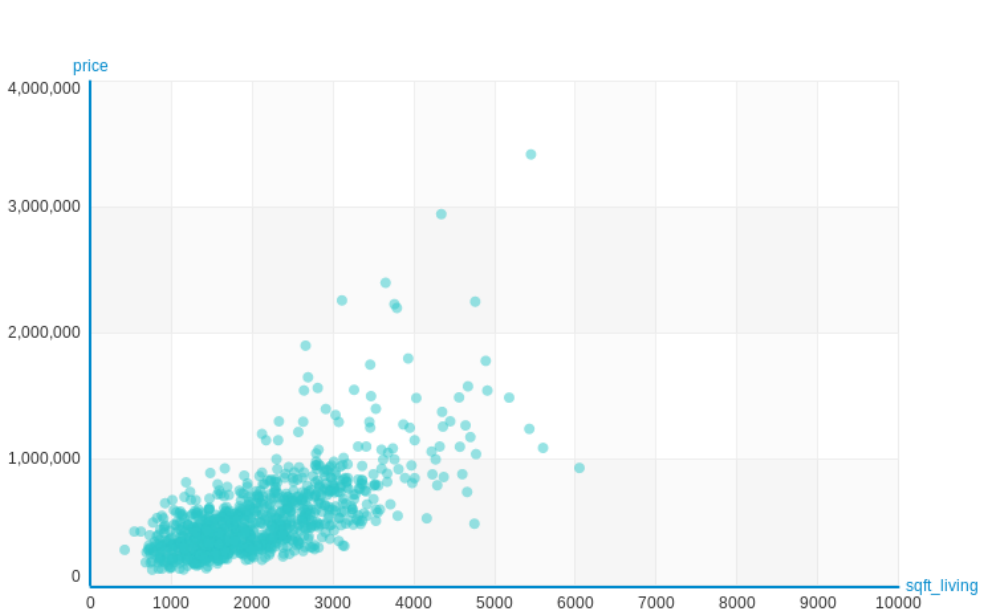

Figure 2.1: Scatter plot of the two features *price* and *sqft_living*.

2.1 Linear and quadratic model

As a first model we created a linear model with the target 'price' and fitted on a constant feature and 'sqft_living'. As one can see in the scatter plot 2.1 the distribution around a single, straight line will be quite broad, which we will see in the RMS.

To compare with the linear fit, we also created a quadratic fit with a constant feature, 'sqft_living' and squared 'sqft_living'. The squared squarefoot-living values were saved in an additional column of our data.

The evaluation of both does not show a significant improvement of the quadratic fit over the linear fit. The RMS improves from 261440 to 250948 as can be seen in the table 2.1.

Table 2.1: Comparison of the RMS of the linear and quadratic model

	Linear model	**Quadratic model**
RMS	261441	250948
max. Error	4362075	5913021

3 Improved regression models

3.1 Training and testing sample

The dataset is split up into a training sample and a testing sample. Randomly 80% of the data is placed in the training sample, the other 20% will be used to evaluate the model.

We created another linear model, but this time only on the training sample. The evaluation on the testing sample shows the same uncertainty as before, wich is what we expected for randomly picking the house data.

RMS testing sample: 255191

RMS training sample: 262943

RMS whole data: 261441

3.2 Predictions

To predict prices with given square foot values and the other way arround, we define two methods:

```
def get_house_price(sqft_living):
    return (sqft_model_lin.get('coefficients')['value'][0] + sqft_model_lin.get('coefficients')['value'][1] * sqft_living)
```

```
def get_house_sqft(price):
    coeff = [sqft_model_lin.get('coefficients')['value'][1], (sqft_model_lin.get('coefficients')['value'][0] - price)]
    return np.roots(coeff)[0]
```

With these two methods some test values are calculated, which can be seen in tables 3.1 and 3.2. The values are in the expected range and can be well compared with the scatter plot in chaper one 2.1.

Table 3.1: Prediction of square feet living

Given price	Predicted square foot living
500000	1940
1000000	3714
1500000	5487
2500000	9034

Table 3.2: Prediction of prices

Given square foot living	Predicted price
1000	234844
2000	516802
3000	798760
4000	1080717

3.3 Multi-feature model

In a new model we include additional features: 'sqft_living', 'sqft_lot', 'grade', 'yr_built'

The RMS goes down a bit to 232201, thus more features made the predictions more precise. To further improve on this we will eliminate outstanding houses from the data. From now on only data is used of houses with prices below 2000000 and square foot living below 5000.

The multi-feature model with the selected data has an RMS of 177179 on the testing samples. Meaning the RMS went down a considerable amount.

3.4 Selective houses

In the following we looked a bit closer at two selected houses.

3.4.1 House nr. 2008000270

The asked for house with the id 5309101299 does not exist. This is why we randomly chose another house from the data to take a closer look at.

The house with the ID 2008000270 is a rather small house, with 1060 sqft living and only one floor. The price is one of the lowest of the dataset with only 291850 dollar. There are 3 bedrooms and 1.5(?!) bathrooms. It was build in 1963.

3.4.2 Prediciton for house nr. 1925069082

The house number 1925069082 is worth more then 2 million and thus one of the data points that are far away from most of the others. It is even so far out of the bulk of data, that we cut it out of our dataset in section 3.3. The prediction with our model fails, as can be seen in table 3.3.

Table 3.3: Prediction and data for house with id 1925069082

Predicted price	**Real price**
1261170	2200000