APARTMENT FOR RENT CLASSIFIED ANALYSIS

Final project on Jungle's Data Science Academy

Shkumbim Mazrekaj

Table of Contents

Key issues to cover and solve

1. Preview the data and select the useful columns
2. Loading the dataset.
3. Cleaning the data.
4. Implement Linear Regression to look for correlation between square feet and the price of the apartment.
5. Discuss the finding of the regression.
6. Visualize the results.
7. Seeing the best states in terms of affordability and for space

Preview the data and select the useful columns

The data was available to download on:

https://archive.ics.uci.edu/dataset/555/apartment+for+rent+classified

Which is fulfills the rules in the gdpr. The data was available in 10 thousand rows and 100 thousand rows, i downloaded the 100 thousand rows dataset.

The dataset has 22 variables or columns which are:

'id', 'category', 'title', 'body', 'amenities', 'bathrooms', 'bedrooms', 'currency', 'fee', 'has_photo', 'pets_allowed', 'price', 'price_display', 'price_type', 'square_feet', 'address', 'cityname', 'state', 'latitude', 'longitude', 'source', 'time'

I don't need all of them so I will only use:

'id', 'title', 'bathrooms', 'bedrooms', 'currency', 'pets_allowed', 'price', 'price_type', 'square_feet' and 'state'

Loading the dataset

For my data processing i used python and more specifically pandas.

The libraries needed:

```
import pandas as pd
import numpy as np
from scipy import stats
import chardet
```

One of the beginning problems was the encoding on the data i had to find it manually using this code:

```
with open("dataset.csv", "rb") as f:
    result = chardet.detect(f.read(100000))  # Read a chunk of the file
    print(result["encoding"])  # Print the detected encoding
```

After finding the encoding we are safe to load the data into a pandas data frame that i named df

```
df=pd.read_csv("dataset.csv", encoding="Windows-1252", sep=";",usecols=[0,1,2,5,6,7,10,11,13,14,17])
df.head(100)
```

The usecols part selects the columns that we discussed

	id	category	title	bathrooms	bedrooms	currency	pets_allowed	price	price_type	square_feet	state
0	5668640009	housing/rent/apartment	One BR 507 & 509 Esplanade	1.0	1.0	USD	Cats	2195.0	Monthly	542	CA
1	5668639818	housing/rent/apartment	Three BR 146 Lochview Drive	1.5	3.0	USD	Cats,Dogs	1250.0	Monthly	1500	VA
2	5668639686	housing/rent/apartment	Three BR 3101 Morningside Drive	2.0	3.0	USD	NaN	1395.0	Monthly	1650	NC
3	5668639659	housing/rent/apartment	Two BR 209 Aegean Way	1.0	2.0	USD	Cats,Dogs	1600.0	Monthly	820	CA
4	5668639374	housing/rent/apartment	One BR 4805 Marquette NE	1.0	1.0	USD	Cats,Dogs	975.0	Monthly	624	NM
...	...	...	...	...	...	...	...	...	...	...	...
95	5668633801	housing/rent/apartment	Two BR 1917 S. 18th St.	1.0	2.0	USD	Cats,Dogs	1015.0	Monthly	845	NE
96	5668632658	housing/rent/apartment	Three BR 7312 South 81st Street	2.0	3.0	USD	Cats,Dogs	1495.0	Monthly	1850	NE
97	5668632537	housing/rent/apartment	One BR 4301 Grand Avenue Parkway	1.0	1.0	USD	NaN	1103.0	Monthly	652	TX
98	5668632393	housing/rent/apartment	One BR 2101 W. ANDERSON LN.	1.0	1.0	USD	NaN	1032.0	Monthly	600	TX
99	5668632355	housing/rent/apartment	Studio apartment 311 Bowie	1.0	2.0	USD	NaN	1729.0	Monthly	448	TX

Cleaning the data

In the price_type column i found out that we have mostly monthly bills and 3 weekly bills

```
price_type
Monthly           99488
Weekly                3
Monthly|Weekly        1
```

For the weekly bills we will make it even by multiplying the price with 4 to make it monthly, as for the monthly|weekly part i will just drop it.

```
df["price"]=df.apply( lambda row: row["price"]*4 if row["price_type"]=="Weekly" else row["price"],axis=1)
df["price_type"]=df.apply( lambda row: "Monthly" if row["price_type"]=="Weekly" else row["price_type"],axis=1)
df=df[df["price_type"]!= "Monthly|Weekly"]
```

Just for checking i will drop null values for the price column because that's the most needed variable together with square_feet

```
df = df.dropna(subset=["price"])
```

```
df["square_feet"].isna().value_counts()
#it doesnt have nan values so we will not perform anything
```

One important decision i had was to analyze for the 4 major populated states: New York, Texas, California and Florida. Since in my opinion doing a linear regression for the whole country won't do it justice because the prices may differ a lot between states. So, i made a filtering with 4 new data frames of the major states:

```
df_ny=df[df["state"]=="NY"]#Dataframe for new york
df_tx=df[df["state"]=="TX"]#Dataframe for Texas
df_ca=df[df["state"]=="CA"]#Dataframe for California
df_fl=df[df["state"]=="FL"]#Dataframe for Florida
```

Implement Linear Regression

What I want to do is make a linear regression where I take the square feet of the property as the independent variable and for the dependent variable to be the price for each of the states.

```
#x will be the independent variable
x_ny=df_ny['square_feet'].values.tolist()
x_tx=df_tx['square_feet'].values.tolist()
x_ca=df_ca['square_feet'].values.tolist()
x_fl=df_fl['square_feet'].values.tolist()
#y will be the dependent variable
y_ny=df_ny['price'].values.tolist()
y_tx=df_tx['price'].values.tolist()
y_ca=df_ca['price'].values.tolist()
y_fl=df_fl['price'].values.tolist()
```

After getting the variables we will perform linear regression and get the main stats:

```
slope_ny, intercept_ny, r_ny, p_ny, std_err_ny = stats.linregress(x_ny, y_ny)
slope_tx, intercept_tx, r_tx, p_tx, std_err_tx = stats.linregress(x_tx, y_tx)
slope_ca, intercept_ca, r_ca, p_ca, std_err_ca = stats.linregress(x_ca, y_ca)
slope_fl, intercept_fl, r_fl, p_fl, std_err_fl = stats.linregress(x_fl, y_fl)
```

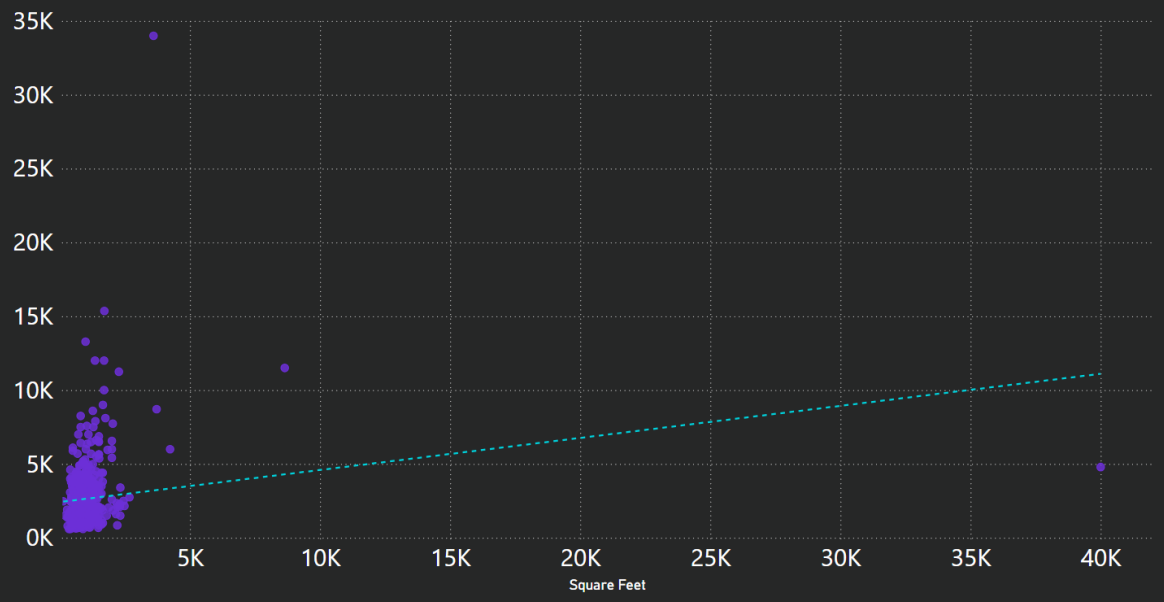
The function of regression has the form:

```
def regression_function(x):
    return slope*x+intercept
```

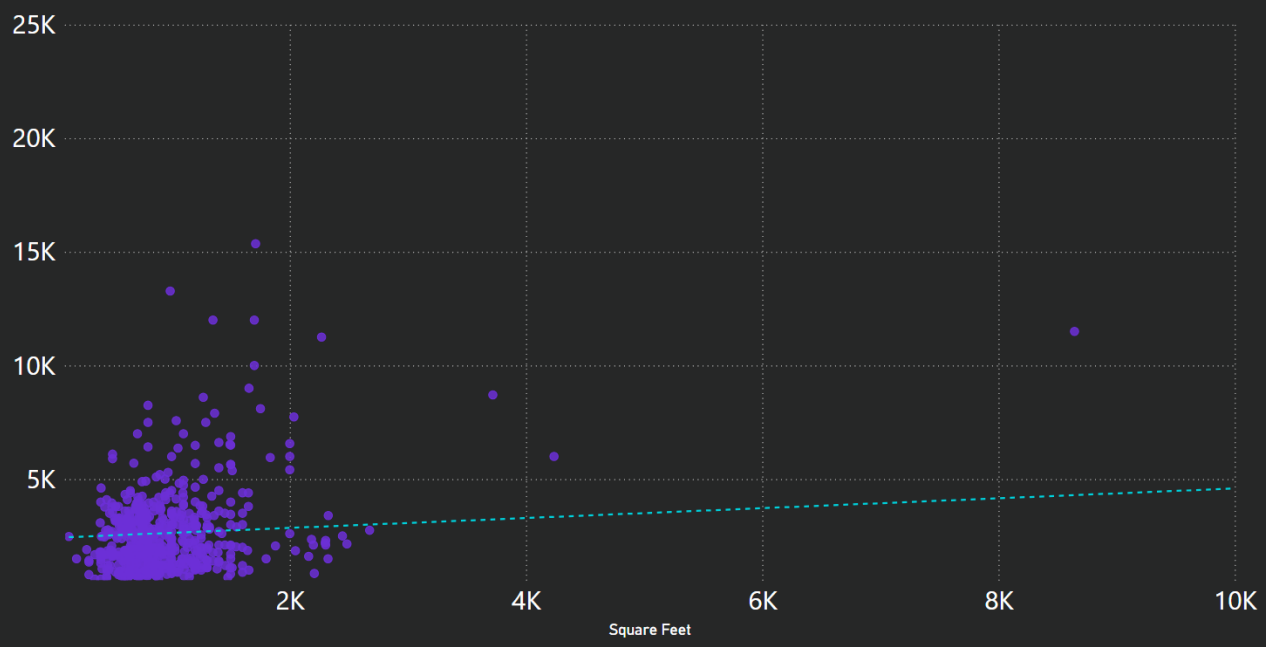
Data Visualizing

Let's see each of the data frames in Power Bi:

New York P-Value: `2.4388301279374208e-05` `Standard Error:` `0.05134658375275616`



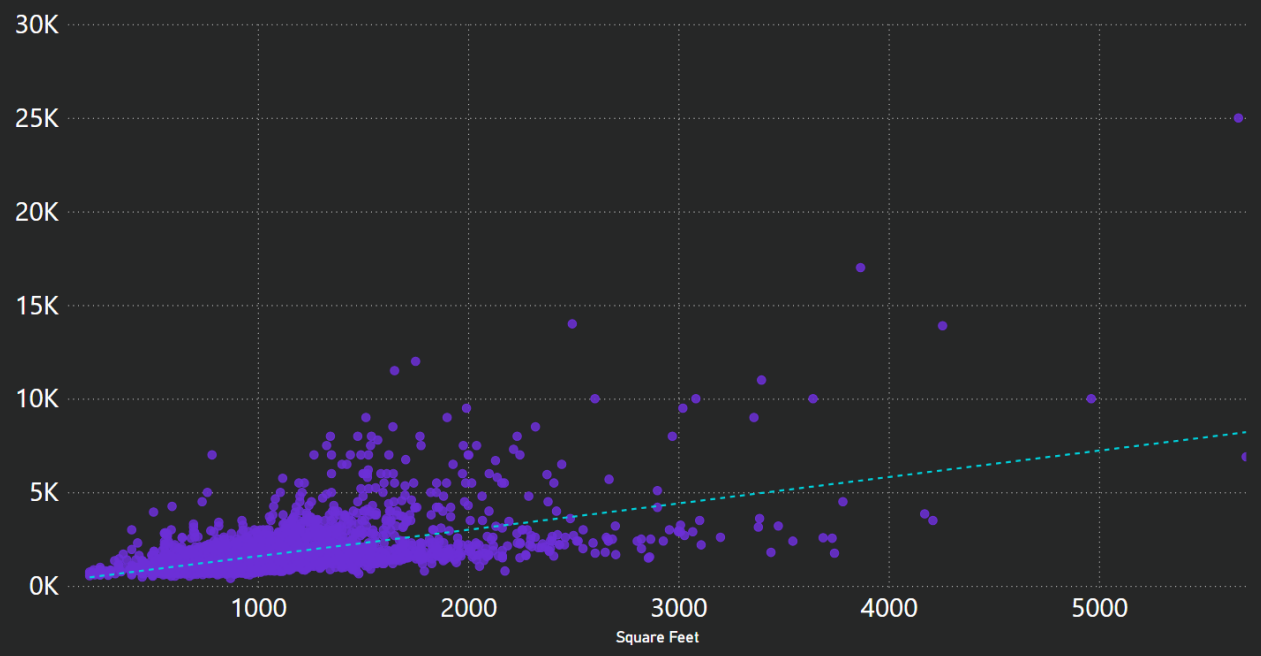
New York visual without the outliers:



Texas P-Value: very close to 0 Standard Error: 0.009442230020371 54

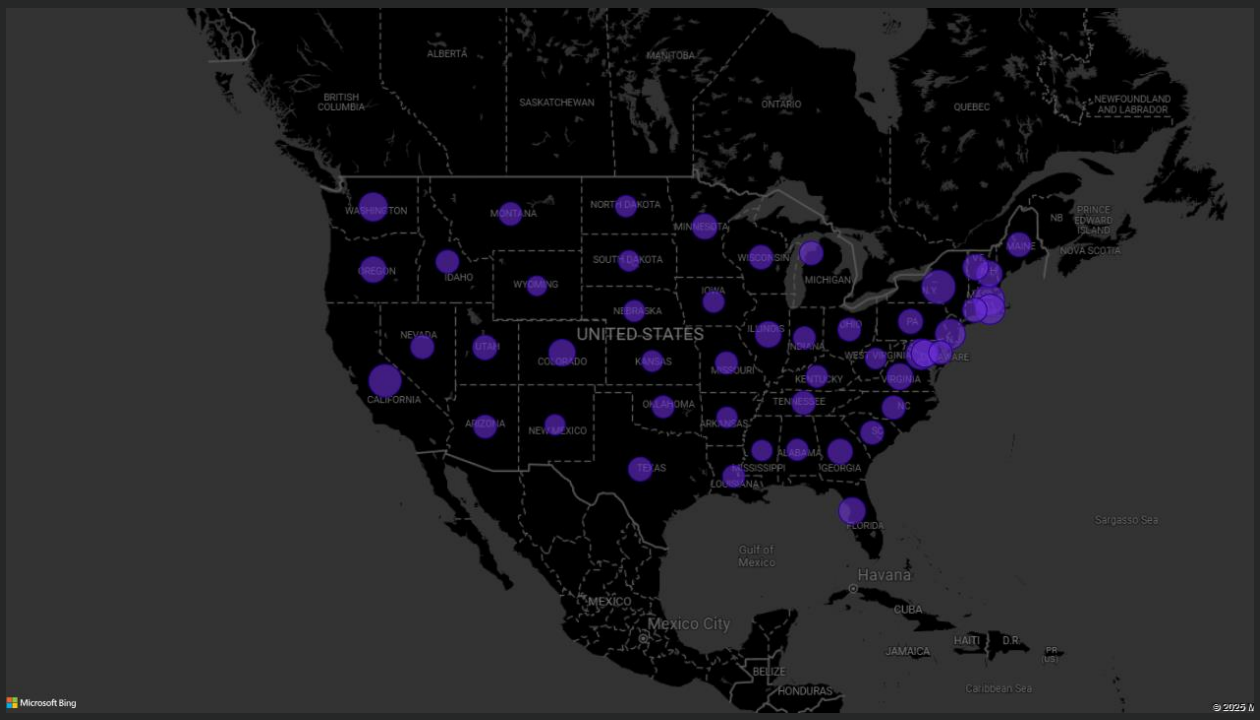
California P-Value: very close to 0 Standard Error: 0.0291989120513961

Florida P-Value: very close to 0 Standard Error: 0.026814904642868426



The best states in terms of affordability and for space

Here is a map showing the average prices for the apartments in terms of states



New York leads in terms of average price per apartment.