Battle of The Neighborhoods: New York Edition

By: Hassan Sarhan

Date: 19/06/2019

1. Introduction

New York City is the definition of a global city. It is the financial center of the United States and the main immigrant landing point for many immigrants to the United States. No city is represented in media as much as New York is. People across the states and all over the world envy the people that live there. However, living in New York has its downsides. The biggest downside for most people is that it is expensive. According to Investopedia.com the cost of living in Manhattan was 138.6% of the U.S. average in 2018 while Brooklyn was a little bit lower at 82% of the U.S average. The second most important thing for people who want to move to New York City is the crime rate. This is mostly because crime in New York has been overly exaggerated in the media, New York is a relatively safe city to live in. However, people moving to New York would still like to be sure that they will be safe when they arrive. The third most thing that is on people's minds is that they want to be close to the trendiest bars, cafes, and venues. So, what I am suggesting is that there is this one perfect neighborhood that has it all. A neighborhood that is close to trendy venues, has a low crime rate, and is relatively affordable to live in. That mythical neighborhood is real, and this report will be about how I found it. We will also create a rudimentary ranking system and we will introduce a multiple linear regression model to predict the ranking of a neighborhood based on three different features.

2. Data

In this report I will be working with five main data sets representing New York City's neighborhoods, venue details, crime rate, average price by neighborhood, and neighborhood ranking. This will involve scraping websites, cleaning the data, feature selection, and merging the data

2.1. New York City Neighborhood Names

The New York City Neighborhood Names data set comes from this NYU website. It has many features for each neighborhood, but we are only interested in four of them: Borough name, Neighborhood name, Latitude, and Longitude

2.2. Foursquare Venue Data

The data about the location of each venue comes from the Foursquare Places API. Each venue comes with a bucket filled with features. In this report we'll only extract the name, latitude, and longitude.

2.3. New York City Criminal Activity Data

This data set comes from the City of New York website. It details all crimes reported by the NYPD in 2016. It contains the details of the crime and the location the crime occurred; however, it does not list the name of the neighborhood, making it difficult to merge with the other data sets. We will address this issue later

2.4. New York City Rent Cost by Neighborhood

This dataset cannot be downloaded from a singular website. It had to be scraped from multiple pages. I will explain the process later in the report, but what you need to know now is that this dataset contains various neighborhoods around New York City and the average rent cost in each one.

2.5. New York City Neighborhood Ranking

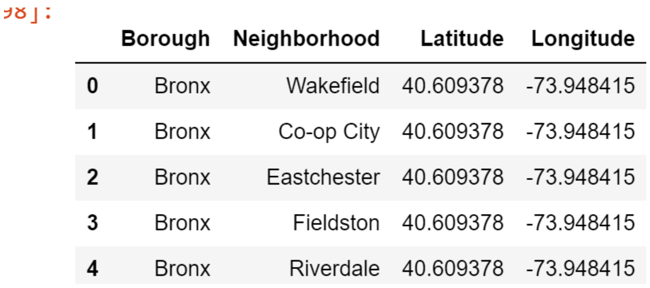
This dataset comes from niche.com and it ranks each neighborhood in New York City based on a wide number of features, including proximity to schools, safety of the neighborhood, and job opportunities. This dataset also cannot be obtained through normal means from a particular website. It had to be scraped from multiple pages. I will explain the process later in the report.

3. Methodology

The methodology section of this report covers all the steps of the data science process. These steps are: 1. Data collection 2. Data cleaning 3. Data merging 4. Exploratory data analysis 5. Predictive modelling 6. Model evaluation

3.1. Data Collection, Cleaning, and Merging

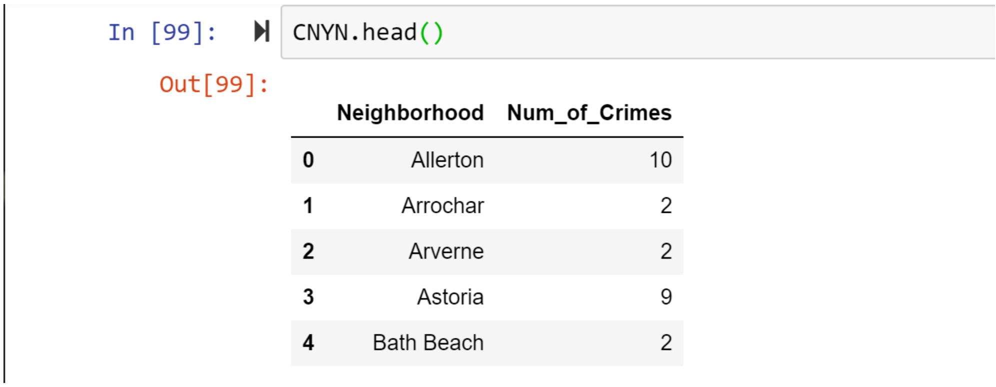
We first begin with collecting the New York City neighborhood data. The data can be downloaded from the website linked above. The data is in the form of a JSON file. For python to read a JSON file we first need to import some libraries. The way I read the JSON file is that I made 4 lists representing each feature. Using a for loop I appended each feature to their respective list, then I created a data frame and added four columns and for each column I assigned it a feature. The four features that I chose are Borough, Neighborhood, Latitude, and Longitude



	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.609378	-73.948415
1	Bronx	Co-op City	40.609378	-73.948415
2	Bronx	Eastchester	40.609378	-73.948415
3	Bronx	Fieldston	40.609378	-73.948415
4	Bronx	Riverdale	40.609378	-73.948415

After that I used the Foursquare API to access venue data in a radius of 1000m around each neighborhood. Each search with the API had a limit of 100 venues. The data collected from the API request is also in the form of a JSON file. To read it I used a function that was used in the labs for this course. This function, which is called getNearbyVenues(), gets some features from each venue and stores it in a data frame. I modified the function to only return two features, the neighborhood name, and the venue name. I did not run the function immediately since we had over 300 neighborhoods in the neighborhood data data frame. I waited until the number of neighborhoods decreased through the merging of data frames.

After that I collected the information on criminal activity in New York City the information was available as a .csv file on the website linked above. The csv file stores a lot of information about the details of each crime but what is most important to us are the following features, borough name, neighborhood name, and the coordinates. Unfortunately, this data set does not provide the neighborhood name, so I used reversed geocoding with the LocationIQ API to get the neighborhood names. Geocoding is the process of obtaining the coordinates of a location based on its name, and reverse geocoding is the process of obtaining the name of a location based on its coordinates. The LocationIQ API returns a JSON file and I was able to get the neighborhood name for all the crimes with a for loop, and after some cleaning and grouping and summing (described in the notebook) the data frame looked like this.

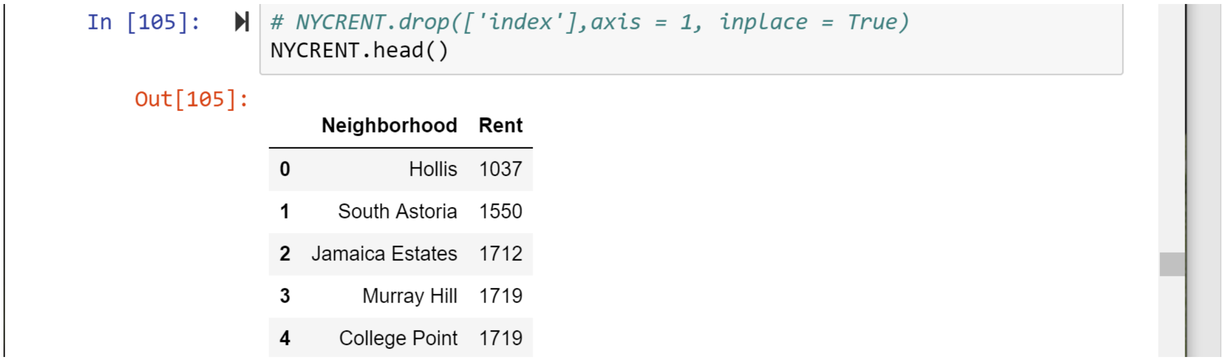


```
In [99]:  CNYN.head()
```

Out[99]:

	Neighborhood	Num_of_Crimes
0	Allerton	10
1	Arrochar	2
2	Arverne	2
3	Astoria	9
4	Bath Beach	2

The next data set that will have to be collected is the rental cost data set. This data set will have to be scraped from four separate websites. With each one listing the average rent in Queens, Bronx, Brooklyn, or Manhattan. I used the beautifulsoup package to scrape the websites. For each borough I made two lists representing the neighborhood names and the average rent costs. I then used a for loop to scrape the websites appending each neighborhood name and its rent to their respective lists. After that I made a data frame for each borough and added the two lists to each data frame as columns. Then I concatenated the data frames together, did some cleaning, and the final result looks like this.

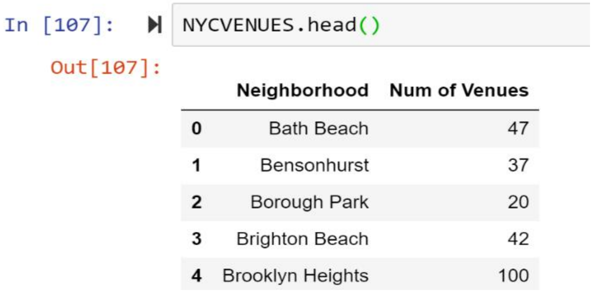


```
In [105]:  # NYCRENT.drop(['index'],axis = 1, inplace = True)
           NYCRENT.head()
```

Out[105]:

	Neighborhood	Rent
0	Hollis	1037
1	South Astoria	1550
2	Jamaica Estates	1712
3	Murray Hill	1719
4	College Point	1719

Now it is time to merge the data frames. The data has been cleaned so this turned out to be a relatively simple process. First, we merge the rent and crimes data frames, and then we merge the result of that with the neighborhood data. This data frame, which is called NYC in the notebook, has 54 rows representing 54 different neighborhoods. Now that we know which neighborhoods to target, we can now get the venue details for each neighborhood. We use the getNearbyVenues() method and assign its value to a data frame called NYCVENUES. Since we are only interested in the number of venues per neighborhood, we group NYCVENUES by neighborhood and count the result of grouping NYCVENUES, this is shown in the notebook. When we're done, the data frame looks like this.

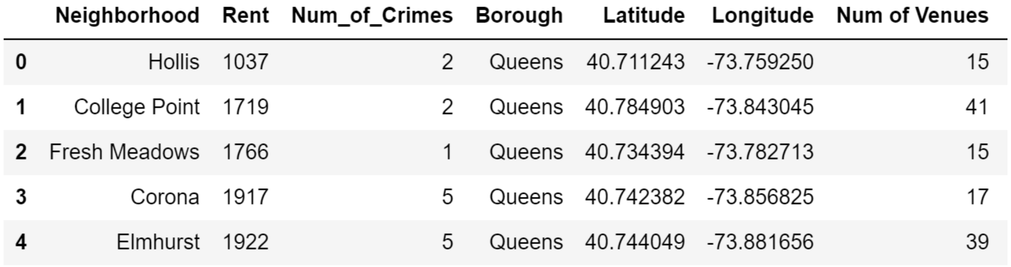


```
In [107]:  NYCVENUES.head()

Out[107]:
```

	Neighborhood	Num of Venues
0	Bath Beach	47
1	Bensonhurst	37
2	Borough Park	20
3	Brighton Beach	42
4	Brooklyn Heights	100

The final step in the data cleaning/merging process is to merge NYC and NYCVENUES together.



	Neighborhood	Rent	Num_of_Crimes	Borough	Latitude	Longitude	Num of Venues
0	Hollis	1037	2	Queens	40.711243	-73.759250	15
1	College Point	1719	2	Queens	40.784903	-73.843045	41
2	Fresh Meadows	1766	1	Queens	40.734394	-73.782713	15
3	Corona	1917	5	Queens	40.742382	-73.856825	17
4	Elmhurst	1922	5	Queens	40.744049	-73.881656	39

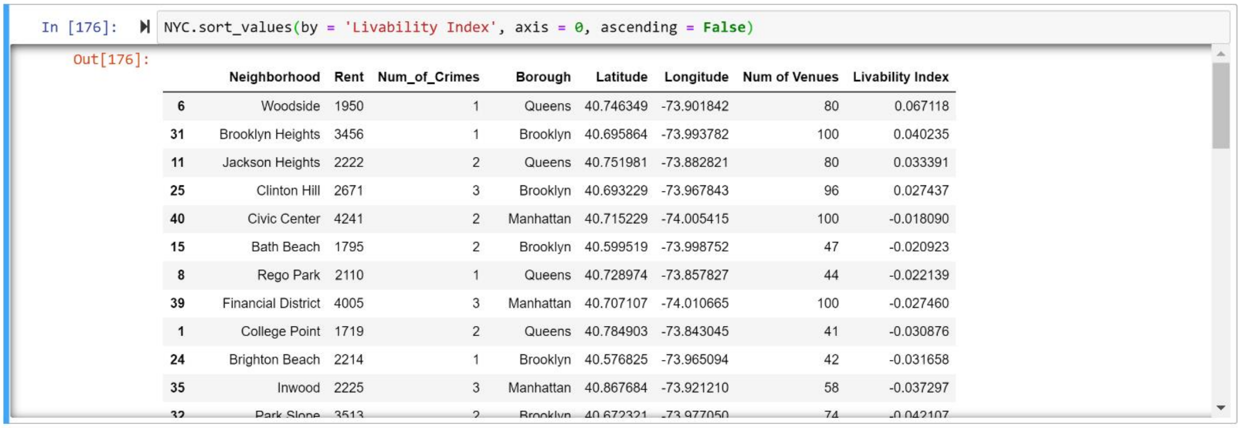
3.2. Introducing the Livability Index

The "Livability Index" is a term I coined to calculate how livable a neighborhood is based on its cost of living, its popularity, and its safety. The cost of living is determined by the average rent price of a neighborhood, the popularity of a neighborhood is determined by the number of nearby venues it has, and the safety is determined by the number of recent crimes in the neighborhood. The higher the livability index is the more livable it is. A more technical definition of the livability index is, the number of venues (normalized) minus the average rent price (normalized) minus the number of crimes (normalized)

$$Livability\ Index = norm(Venues)\ +\ norm(-rent)\ +\ norm(-crimes)$$

We must normalize all the values to give equal weight to each of them. The rent and the crime data must be negative because they have a negative effect on the livability index. For example, a neighborhood with a high rent and a high number of crimes must have a low livability index.

Our goal now is to implement the Livability Index equation to the features of each neighborhood. We start this process by using the preprocessing module from sci-kit learn. We use the normalize function from preprocessing on the rent, crime, and venue data and we assign each of them to a list. Then we add the lists as columns to a new data frame called normal_NYC. We can now easily create a new column in normal_NYC called "Livability Index" by using the .sum() function along the columns. To see which neighborhood has the best livability index we must add the "Livability Index" column to the NYC data frame, doing so is very simple. To get the best neighborhood we can sort the values of NYC by the "Livability Index" column in descending order, and this is what it looks like.



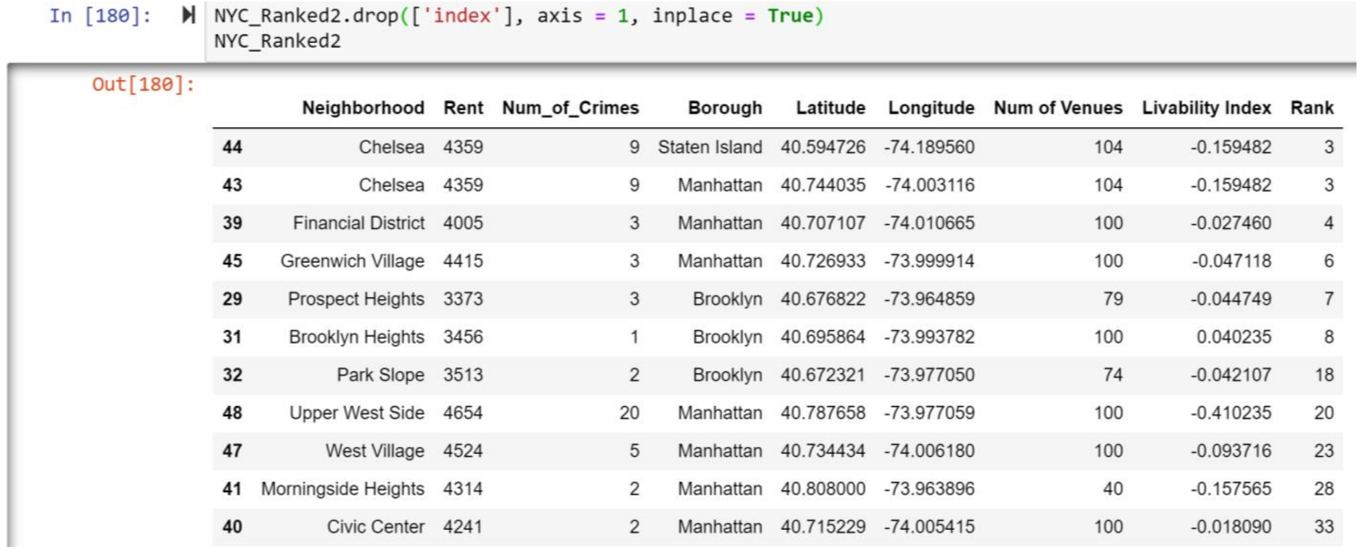
```
In [176]:  NYC.sort_values(by = 'Livability Index', axis = 0, ascending = False)
```

Out[176]:

	Neighborhood	Rent	Num_of_Crimes	Borough	Latitude	Longitude	Num of Venues	Livability Index
6	Woodside	1950	1	Queens	40.746349	-73.901842	80	0.067118
31	Brooklyn Heights	3456	1	Brooklyn	40.695864	-73.993782	100	0.040235
11	Jackson Heights	2222	2	Queens	40.751981	-73.882821	80	0.033391
25	Clinton Hill	2671	3	Brooklyn	40.693229	-73.967843	96	0.027437
40	Civic Center	4241	2	Manhattan	40.715229	-74.005415	100	-0.018090
15	Bath Beach	1795	2	Brooklyn	40.599519	-73.998752	47	-0.020923
8	Rego Park	2110	1	Queens	40.728974	-73.857827	44	-0.022139
39	Financial District	4005	3	Manhattan	40.707107	-74.010665	100	-0.027460
1	College Point	1719	2	Queens	40.784903	-73.843045	41	-0.030876
24	Brighton Beach	2214	1	Brooklyn	40.576825	-73.965094	42	-0.031658
35	Inwood	2225	3	Manhattan	40.867684	-73.921210	58	-0.037297
32	Park Slope	3513	2	Brooklyn	40.672321	-73.977050	74	-0.042107

3.3. Using a Different Metric

This was a very simple analysis. Let's take a look at this website which ranks the best New York neighborhoods using a much more complicated method than what I did. We will need to scrape the site to produce a data frame. The process is outlined in the notebook and is similar to what we did for the neighborhood prices data frame. After we create NYC_Ranked as shown in the notebook we sort it by the rank that was scraped from niche.com and this is the result:



```
In [180]:  NYC_Ranked2.drop(['index'], axis = 1, inplace = True)
           NYC_Ranked2
```

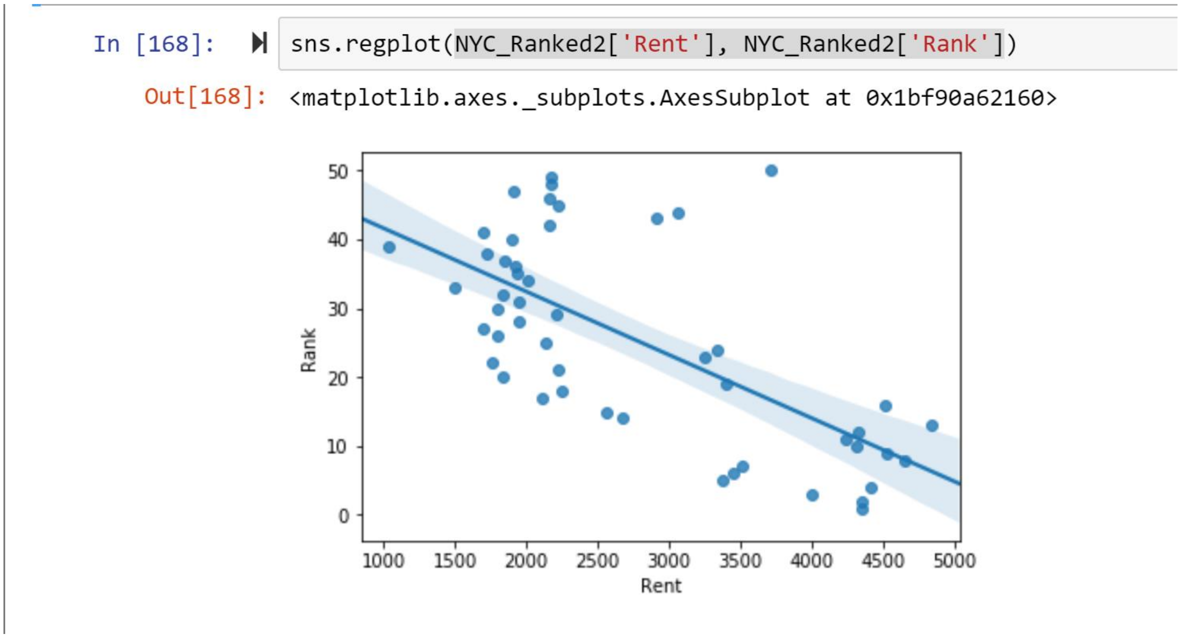
Out[180]:

	Neighborhood	Rent	Num_of_Crimes	Borough	Latitude	Longitude	Num of Venues	Livability Index	Rank
44	Chelsea	4359	9	Staten Island	40.594726	-74.189560	104	-0.159482	3
43	Chelsea	4359	9	Manhattan	40.744035	-74.003116	104	-0.159482	3
39	Financial District	4005	3	Manhattan	40.707107	-74.010665	100	-0.027460	4
45	Greenwich Village	4415	3	Manhattan	40.726933	-73.999914	100	-0.047118	6
29	Prospect Heights	3373	3	Brooklyn	40.676822	-73.964859	79	-0.044749	7
31	Brooklyn Heights	3456	1	Brooklyn	40.695864	-73.993782	100	0.040235	8
32	Park Slope	3513	2	Brooklyn	40.672321	-73.977050	74	-0.042107	18
48	Upper West Side	4654	20	Manhattan	40.787658	-73.977059	100	-0.410235	20
47	West Village	4524	5	Manhattan	40.734434	-74.006180	100	-0.093716	23
41	Morningside Heights	4314	2	Manhattan	40.808000	-73.963896	40	-0.157565	28
40	Civic Center	4241	2	Manhattan	40.715229	-74.005415	100	-0.018090	33

3.4. Relationships Between Variables

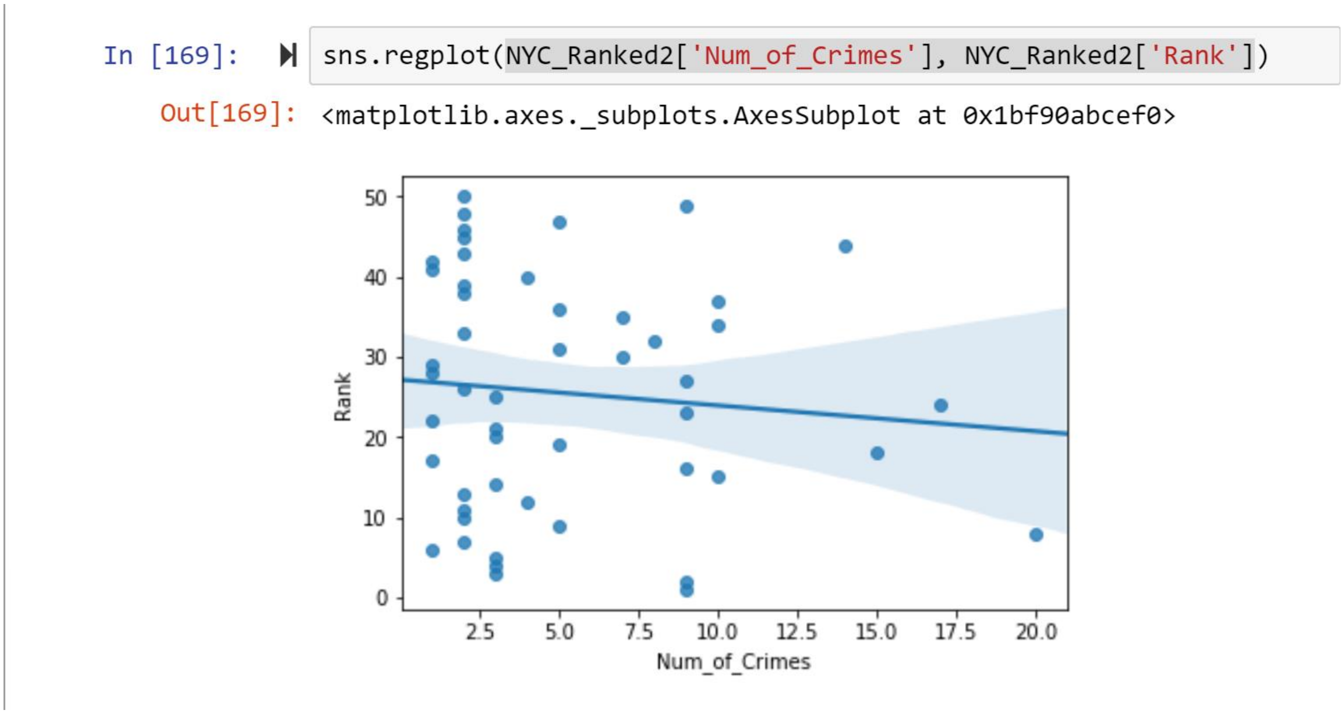
Using the Seaborn module, I created regression plots to show the relationship between the different features of a neighborhood and its niche.com rank

3.4.1. Rent vs. Rank



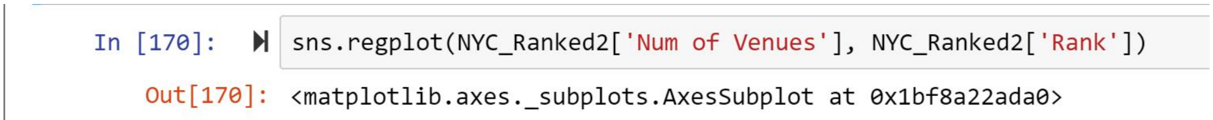
```
In [168]:  sns.regplot(NYC_Ranked2['Rent'], NYC_Ranked2['Rank'])
Out[168]: <matplotlib.axes._subplots.AxesSubplot at 0x1bf90a62160>
```

3.4.2. Number of Crimes vs. Rank



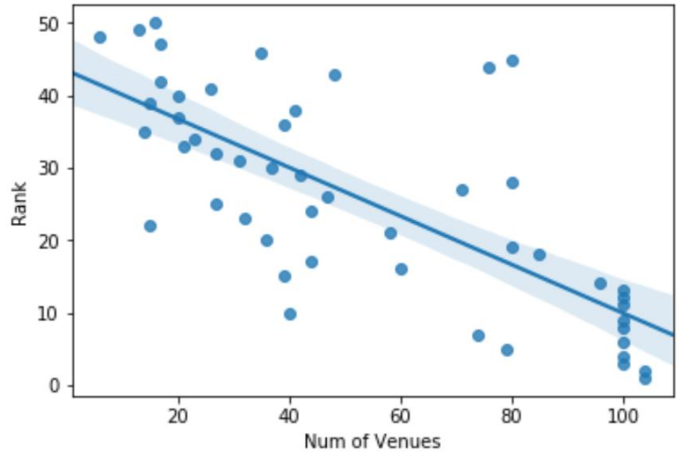
```
In [169]:  sns.regplot(NYC_Ranked2['Num_of_Crimes'], NYC_Ranked2['Rank'])
Out[169]: <matplotlib.axes._subplots.AxesSubplot at 0x1bf90abcef0>
```

3.4.3. Number of Venues vs. Rank



```
In [170]:  sns.regplot(NYC_Ranked2['Num of Venues'], NYC_Ranked2['Rank'])

Out[170]: <matplotlib.axes._subplots.AxesSubplot at 0x1bf8a22ada0>
```

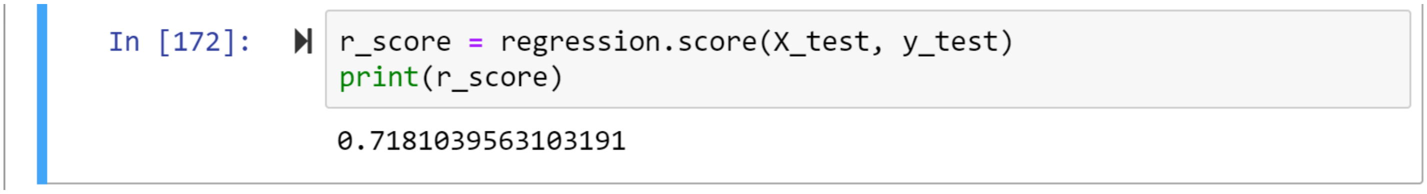


3.5. Predictive Modelling with Multiple Linear Regression

In this section we will create a linear model of the niche.com ranking system based on three features of each neighborhood. With this linear model we can predict the ranking of a neighborhood based on its rent, number of crimes, and number of venues. Creating the linear model is very straight-forward. We simply need to import the linear_model module from sci-kit learn. We create a model and call it regression. Then we use the .train_test_split() function, also from sci-kit learn, to split normal_NYC into training and testing data.

3.6. Model Evaluation

Evaluating a model is essential to know if it's worth using. This model is no different. Evaluating a regression model is simple and can be done with only one line of python code. I evaluated this model with the .score() function using the test data generated from the .train_test_split() function in the previous section. This is the result.



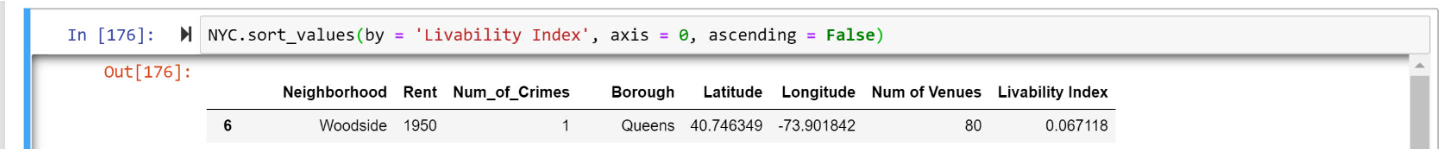
```
In [172]:  r_score = regression.score(X_test, y_test)
           print(r_score)

           0.7181039563103191
```

4. Results

4.1. The Best Neighborhood According to the Livability Index

From section 3.2 we sorted the neighborhoods by which one had the highest livability index. Let's take a look at the top result:



In [176]:
```
NYC.sort_values(by = 'Livability Index', axis = 0, ascending = False)
```

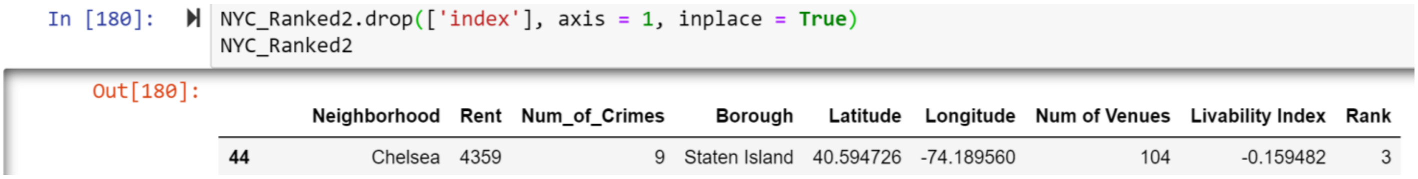
Out[176]:

	Neighborhood	Rent	Num_of_Crimes	Borough	Latitude	Longitude	Num of Venues	Livability Index
6	Woodside	1950	1	Queens	40.746349	-73.901842	80	0.067118

Woodside has a very reasonable rent for New York, the number of crimes is very low at 1, and there are 80 venues in the vicinity. It seems to me that Woodside is a very good neighborhood to live in and might be one of the best

4.2. The Best Neighborhood According to niche.com

From section 3.3 we sorted the neighborhoods by which one had the highest niche.com rank. Let's take a look at the top result:



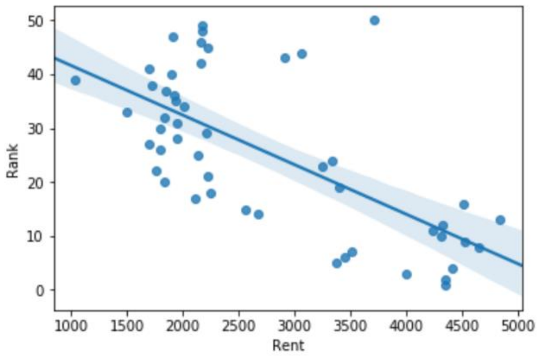
In [180]:
```
NYC_Ranked2.drop(['index'], axis = 1, inplace = True)
NYC_Ranked2
```

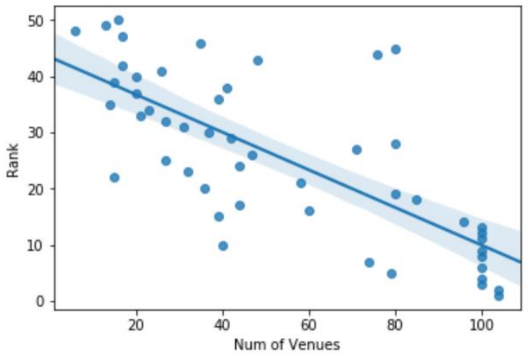
Out[180]:

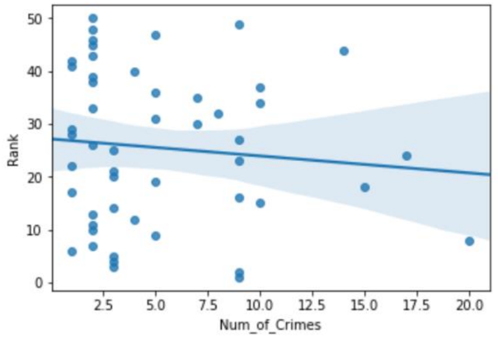
	Neighborhood	Rent	Num_of_Crimes	Borough	Latitude	Longitude	Num of Venues	Livability Index	Rank
44	Chelsea	4359	9	Staten Island	40.594726	-74.189560	104	-0.159482	3

Chelsea has a very expensive rent and a high number of crimes giving it a low livability index. However, it is the top ranking according to niche.com

4.3. Relationships Between Variables







4.3.1. Rent vs. Rank

As we can see the more expensive a neighborhood is the higher its ranking

4.3.2. Number of Venues vs. Rank

The more the number of venues a neighborhood has the higher its ranking

4.3.3. Number of Crimes vs. Rank

The regression line is almost horizontal meaning that the number of crimes has little effect on the rank

4.4. Multiple Linear Regression

The r-score for our linear model is ~0.72 which is not that great

5. Discussion

5.1. Livability Index vs Niche.com Ranking

Obviously, a data firm's efforts in creating a ranking system will be more comprehensive than an individual data scientist-to-be's attempt, but I don't think that my attempt was half bad. I would change some things about it though. Now that I know that the number of crimes has little effect on the ranking, I would replace the number of crimes feature with something else. Even though many people think otherwise, New York is relatively safe city to live in so crime data shouldn't matter and we should be looking at different features when deciding which neighborhood to live in like the level of traffic or job opportunities for example. One thing that I noticed is that the higher the rent of a neighborhood the higher its niche.com rank. You can see it in section 4.3.1. This leads me to believe that niche.com's ranking was targeted to a more affluent audience, while the livability index gave more weight to the cost of living of a neighborhood.

5.2. Improving Our Model

Our model's r-squared score wasn't that great. This is simply because we lacked enough data to train our model with. If I were to do this project again, I would collect way more data and I would replace the number of crimes with a more relevant feature.

6. Conclusion

This report was aimed at finding the best neighborhood in New York City to live. With that in mind I think it did a pretty good job of helping the reader come to a decision on where to live in New York, and what factors to consider and not to consider when moving here. I would also like to say that I enjoyed writing this report and I enjoyed taking this specialization on Coursera, it made the process of studying to become a data scientist very fun and challenging.