Project Title: Stay Recommendations

Course: Programming for Big Data Analytics

Submitted by:
Dhaval Patel (djp526)
Vishnu Thakral (vvt223)
Yash Rathod (yhr208)

Guided by:
Prof. Juan Rodriguez
Computer Science and Engineering Department,
Tandon School of Engineering,
New York University

Introduction

When tourists visit cosmopolitan cities like New York, the first problem they face is of accommodation, which is mostly due to the Skyrocket prices of Hotels. Startups like Airbnb have come forward to help provide cheaper accommodations to visitors in form of temporary living spaces. This option does come with a trade off in the form of Security and Safety, especially when the guest has no idea of the neighborhood. We have come up with a recommendation system for tourists to have a peaceful stay in New York. The recommendations consider Safety of the Neighborhood, nearby restaurants listing (both of which are presently not considered by hospitality chains in giving recommendations), property's past price History, properties past booking and ratings (if present).

Datasets

The system is currently developed for New York City using below three datasets, we can develop a large-scale recommendation system if we consider data from other cities as well.

Datasets used:

1. The Airbnb Dataset contains all the info of the property like price, location, rating etc. http://insideairbnb.com/new-york-city/
2. NYC crime dataset contains info of the crime reports made to NYPD. https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i
3. NYC Restaurant dataset contains the info of restaurants in NYC https://www.kaggle.com/new-york-city/ny-department-of-health-and-mental-hygiene#dohmh-newyork-city-restaurant-inspection-results.csv

Data Insights:

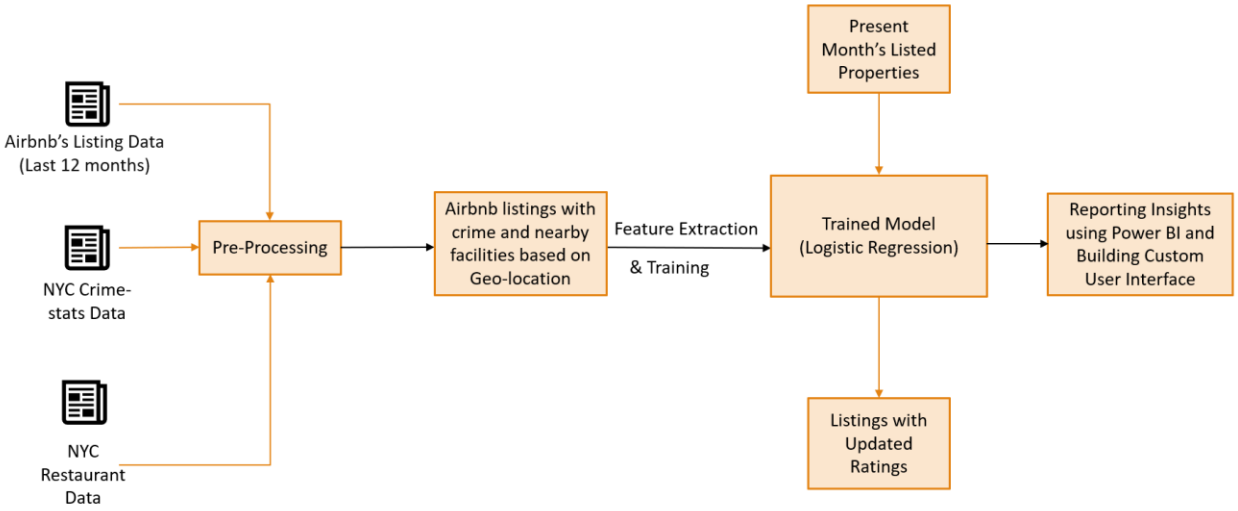
Dataset	CSV File	Rows	Col
Airbnb	listings.csv	50K/month	16
	reviews.csv	1.1M/month	6
Crime	NYPD_Complaint_Historic_DataDictionary.xlsx	6.5M	35
Restaurant	dohmh-new-york-city-restaurant-inspection-results.csv	396K	26

As these datasets were too large and takes computation time to give recommendations, we came up with a solution where we use Big Data technologies to reduce computation time and to give fast retrieval results to customer/tourists.

Methodology

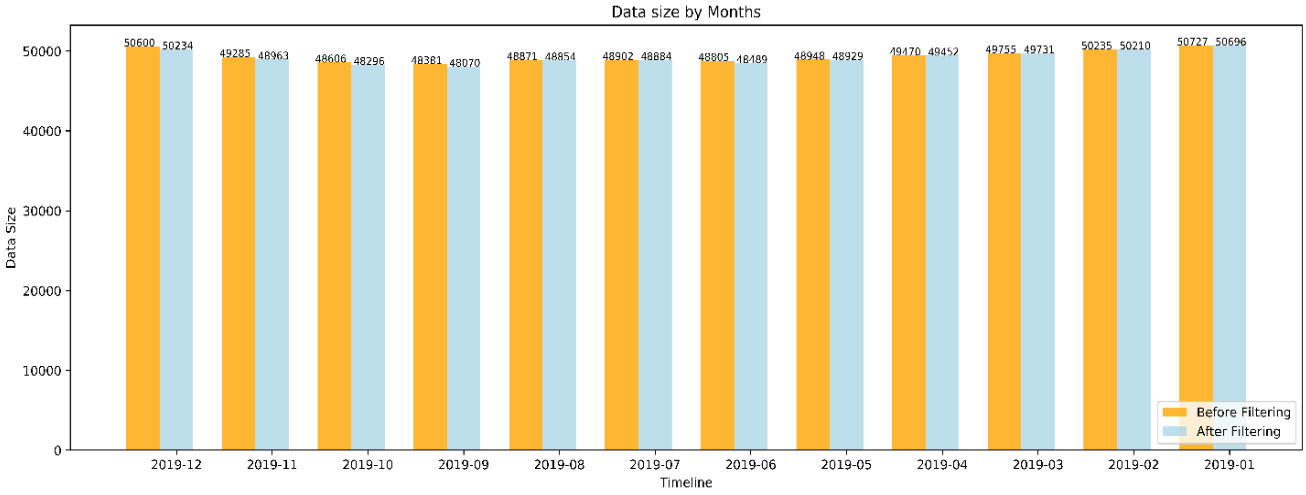
As we work in collaborative environment where many developers work on same project and share their expertise on specific task, we divided our project into major 4 tasks.

Task 1: Preprocessing (Removing null values, cleaning datasets etc.)
Task 2: Combing datasets and Computing Feature set
Task 3: Training Machine Learning Model
Task 4: Reporting Insights and User Interface



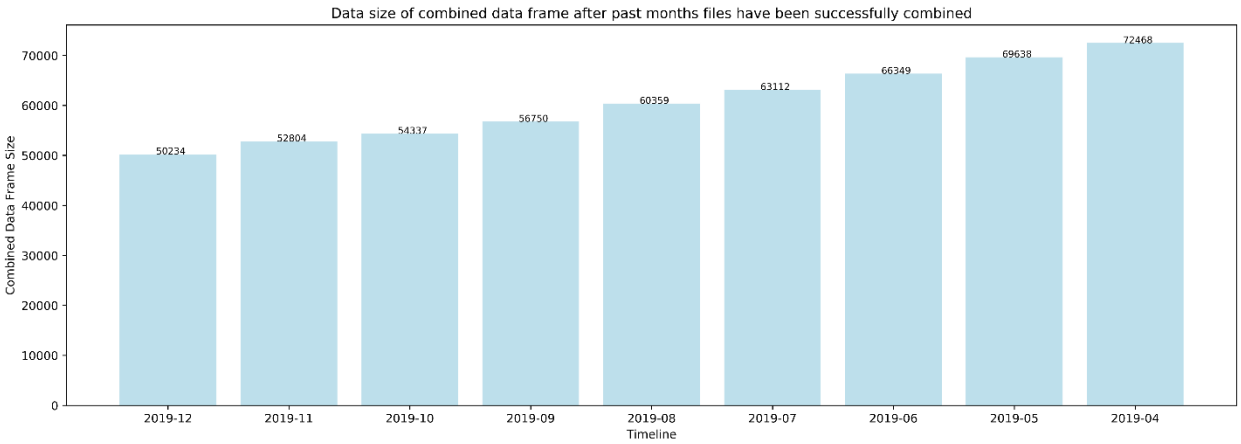
Task 1: Preprocessing

In this stage we remove listings which has null/not valid pricing values. Below graph shows the data loss in each month after the removal of null values.



Once the null values are filtered, then we combine each month's listings with their previous months listings. During this process, the following set of sub datasets gets created,

Sub Data 1: Listings which were present last month but right now got delisted
Sub Data 2: listings which are present in current month and were also present in last month
Sub Data 3: listings which are present only in this month and were not present last month



Above graph shows that once we start combining those Sub datasets for every month the size of dataset increases and the final dataset is created for computing feature set for Task 2.

Task 2: Combing datasets and Computing Feature set

After merging the Airbnb Listing Data of the last 12 months, we clean it further by removing the Null Valued/Zero Price Entries and changing the datatype of many columns to Integer / Float Datatype as required.

Now, after cleaning, we first combine the Airbnb Dataset with the Crime Stat Dataset. Using the geo-coordinates, we count the number of criminal activities reported within a 0.5Km radius of each listing

```
%spark.pyspark

combinedDF = combinedDF.groupBy("id").count().withColumnRenamed("id","id_1")

combinedDF.show()
```

```
+-------+-----+
|   id_1|count|
+-------+-----+
|  16974|  125|
|  55498|   32|
| 193105|   21|
| 233638|   41|
| 237127|  104|
| 258688|   65|
| 271083|   38|
| 466277|   34|
| 578941|   13|
| 622410|   49|
| 644575|   12|
| 882209|   35|
|1022204|   99|
|1266411|   42|
|1535236|   39|
```

We perform similar set of operations to clean the restaurant dataset

After cleaning, we combine the Airbnb Dataset with the Restaurant Stat Dataset. And again, using the geo-coordinates date we count the number of restaurant and facilities within a 0.5Km radius of each listing.

We then combine, the above two counts (Restaurant and Crime Stats) for each listing.

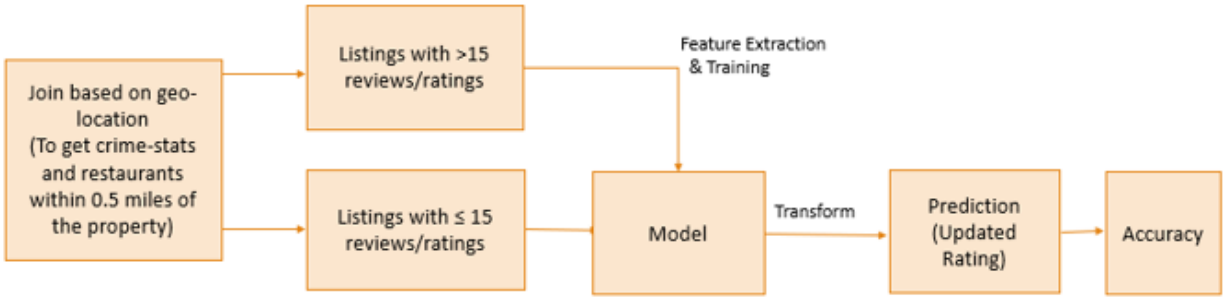
CAMIS	BORO	Latitude_rest	Longitude_rest
50078597	Manhattan	40.77837	-73.94829
40364179	Manhattan	40.801373	-73.96016
50069376	Queens	40.75847	-73.82961
50061519	Brooklyn	40.678616	-73.97898
41109700	Queens	40.736694	-73.86566
50057858	Brooklyn	40.707428	-73.95504
50002396	Brooklyn	40.673416	-73.95696
50000582	Manhattan	40.72706	-73.979774
40397305	Brooklyn	40.691387	-73.95127
50006240	Manhattan	40.72282	-74.00472
41329496	Brooklyn	40.60888	-73.97243
41477916	Manhattan	40.760384	-73.96997
50054573	Manhattan	40.709824	-74.012
40692961	Brooklyn	40.637756	-73.89672
50063311	Queens	40.76033	-73.82987

*Restaurant Data After Cleaning

The Final dataset has the required count data as shown below. This Data has more than 72K records and will be now sent to train and built a Machine Learning model.

count_restaurant	count_crime	id	neighbourhood_cleansed	neighbourhood_group_cleansed	Latitude_listing	Longitude_listing	price	number_of_reviews	review_scores_rating
987	37	10034090	Williamsburg	Brooklyn	40.71049	-73.94515	40	1	100
4375	99	1022204	Lower East Side	Manhattan	40.72198	-73.98932	110	49	98
686	24	10234090	Sunset Park	Brooklyn	40.6642	-73.99371	105	1	80
3341	81	10500222	West Village	Manhattan	40.73492	-74.00457	196	2	100
155	10	10537769	Greenpoint	Brooklyn	40.72963	-73.94937	45	0	null
3164	106	10692885	Chelsea	Manhattan	40.74237	-74.00063	120	2	100
599	12	10708986	Williamsburg	Brooklyn	40.71766	-73.94367	120	27	97
194	35	10974609	East Flatbush	Brooklyn	40.66209	-73.93947	42	1	null
529	21	11483117	Clinton Hill	Brooklyn	40.69483	-73.96403	54	30	95
825	97	11788559	Harlem	Manhattan	40.8014	-73.95469	450	9	93
362	35	11890343	Bushwick	Brooklyn	40.69332	-73.91216	5	92	86
623	67	11903561	East Village	Manhattan	40.72093	-73.97964	100	0	null
310	39	11948586	Flatbush	Brooklyn	40.63836	-73.95698	85	6	73
306	69	12211018	Bedford-Stuyvesant	Brooklyn	40.6886	-73.94153	79	0	null
868	108	12626663	Harlem	Manhattan	40.80854	-73.94305	125	1	100

Task 3: Training Machine Learning Model



From the dataset we received after preprocessing, we have 3 input features on which we can train our model: Crime stats of listings ("count_crime"), restaurant count of listings (“count_restuarant”), average price of listing(“price”).

Our output feature is the ratings that users have already given to a particular Airbnb listing. We have classified listings with less than 15 ratings as unrated property. Out of all the dataset we received after preprocessing, we use random sampling to put 75% of data for training, and the other 25% for testing.We used Naïve Bayes and logistic regression for the ML model but the accuracy of Naïve Bayes was poor as it did not consider the fact that there would be relation between the customer review and the crime stat in the locality or rating of customer and the restaurant nearby.

Attached below is the list of properties, with their predicted ratings in the column “prediction”. Since we used 20 classifications, the values ranges from 0-20.

review_scores_rating	features	rawPrediction	probability	prediction
12	[0.001,0.022,0.06...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	16.0
19	[0.001,0.023,0.19...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
19	[0.001,0.024,0.05...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
19	[0.002,0.02,0.15,...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
19	[0.002,0.02,0.32,...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
19	[0.002,0.021,0.05...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
18	[0.002,0.021,0.06...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	18.0
18	[0.002,0.021,0.06...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	18.0
18	[0.002,0.021,0.2,...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	18.0
19	[0.002,0.021,0.2,...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
19	[0.002,0.021,0.25...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
19	[0.002,0.022,0.03...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
20	[0.002,0.035,0.21...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	20.0
20	[0.003,0.009,0.09...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	20.0
19	[0.003,0.027,0.3,...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
19	[0.004,0.009,0.07...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
18	[0.004,0.011,0.08...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	18.0
16	[0.004,0.012,0.05...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	18.0
19	[0.004,0.012,0.07...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
19	[0.004,0.012,0.25...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
17	[0.004,0.013,0.1,...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	18.0
19	[0.004,0.021,0.08...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0
19	[0.004,0.026,0.03...	[-577.44156795741...	[0.0,0.0,0.0,0.0,...	19.0

This prediction model gave us an accuracy of over 82%

```
In [216]: from pyspark.ml.evaluation import MulticlassClassificationEvaluator

          # Select (prediction, true label) and compute test error
          evaluator = MulticlassClassificationEvaluator(
              labelCol="review_scores_rating", predictionCol="prediction", metricName="accuracy")
          accuracy = evaluator.evaluate(predictions)
          print("Test Error = %g" % (1.0 - accuracy))

          Test Error = 0.171021
```

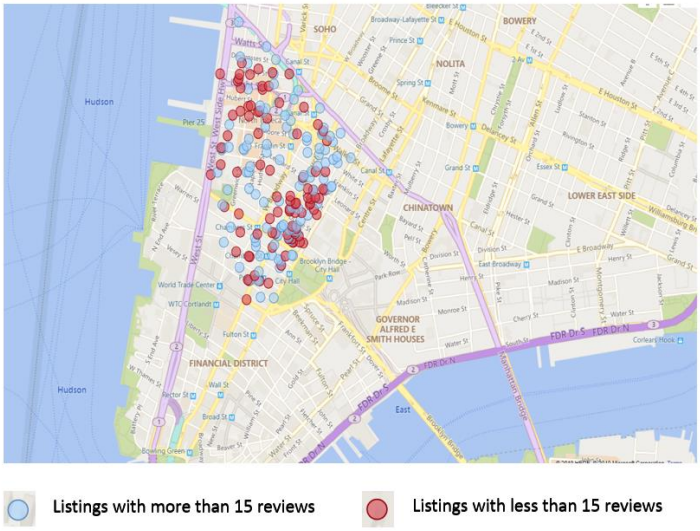
Now, this trained model is used to predict the values for the newly listed ratings in December.

Task 4: Reporting Insights and User Interface

Reporting Insights

To provide flexibility to our customer in selecting area's which they like and to apply various choice filters, we developed a Power BI Dashboard which customers can use to find their best suited housing. We used Map visuals to show the housings which were listed in Airbnb dataset.

Here are some example visuals which appear as customer puts the filters.



Customer selected the Area Tribeca, Manhattan and compared listings with more than 15 reviews

User Interface

Customers can use custom built user interface to select near by listings.
We give option for selecting listings according to 3 features.

1. Relevance (According to Machine Learning model)
2. Safe (Safety of Neighborhood)
3. Nearby Restaurants (Using distance threshold)

Scalability

We limited our computations only for Airbnb listings in NYC, but organizations like Airbnb have their presence all over the world. If this project were to be implemented on a large scale by an organization like Airbnb, it will have to build a prediction model for each of its city separately. **This training and prediction should be specific to every city because every city has different crime and restaurant stats specific to it**. Also, as we saw on an average more than 3K new listings get added in NYC every month, this model training & rating prediction task can be performed on a bi-weekly basis for each city.

Conclusion

It can be seen from the results of Task 3 that using Logistic Regression we are able to identify rating class for the new listings. The current system used PySpark, SparkML and Power BI as part of technology stack. We can deploy the same system for other cities.