CS 6350.002 Big Data Management and Analytics

Project Final Report

December 2017

Two Sigma Connect: Rental Listing Inquiries

How much interest will a new rental listing on RentHop receive?

Team members:

Vishal Addala(vxa162530)

Pruthvi Vooka (pxv162030)

Shravya Kuncha (sxk151632)

Likhitha Nanda (lxn160430)

1. Introduction

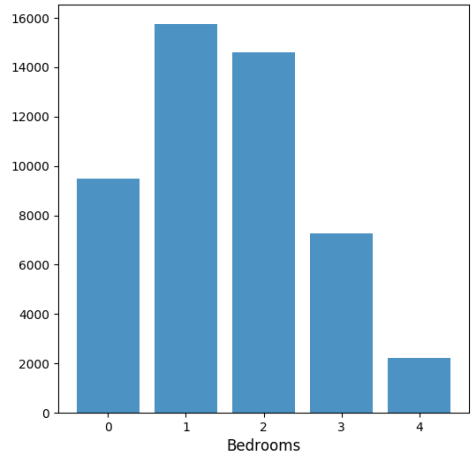
Every person in this world wants to find the perfect place to call it Home. They search through endless listings of the houses to find their perfect place. The search is difficult since structuring and making sense of all the data of the houses in the listings is hard. People tend to look at features such as bedrooms, bathrooms and price to select their houses. The better these features, the more likely that the person is interested in the listing. By exploring and analyzing the previous listing interest levels and their features, we can predict the interest level of any new listing. This will help people find listings easier and help people know what changes they must make to their listings to increase its popularity. Our project uses the data from renthop.com which is an apartment listing website in the New York City to predict if a listing has high, low or medium interest rating.

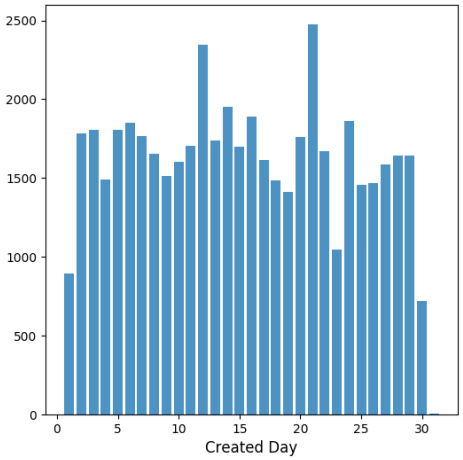
2. Dataset Description

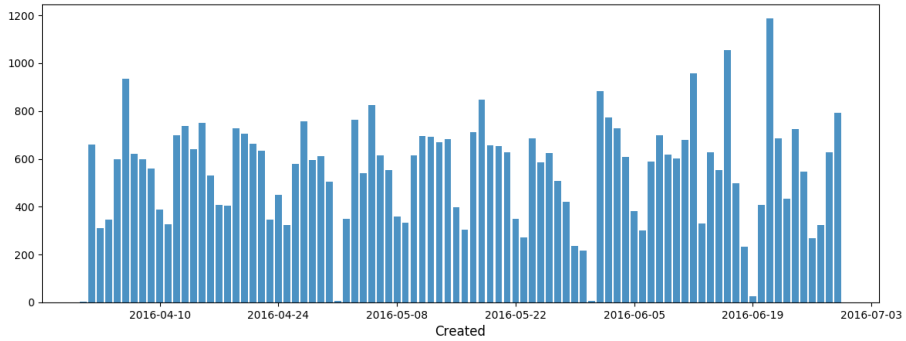
The data of this project comes from renthop.com which is an apartment listing website. The apartments in the data are in New York city. The train data contains the listings and their interest level. The test data contains the listings whose interest levels are to be predicted. The train data set consists of 49352 listings and the test dataset consists of 74659 listings of apartments. Each listing contains the following data fields,

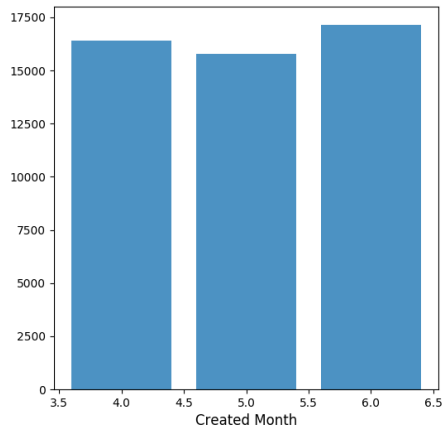
1. Bathrooms: number of bathrooms.
2. Bedrooms: number of bedrooms.
3. Building_id: identifier for the building.
4. Created: when the listing was created.
5. Description: short description of the apartment.
6. Display_address: address displayed to the user.
7. Features: List of features about the apartment.
8. Latitude: The latitude of the location of the apartment
9. Longitude: The longitude of the location of the apartment.
10. Manager_id: the management in charge of the apartment.
11. Photos: links of photos of the apartment.
12. Price: the price of the apartment in USD.
13. Street_address: the street address of the apartment.
14. Interest_level: the target variable which tells us how popular the apartment is.

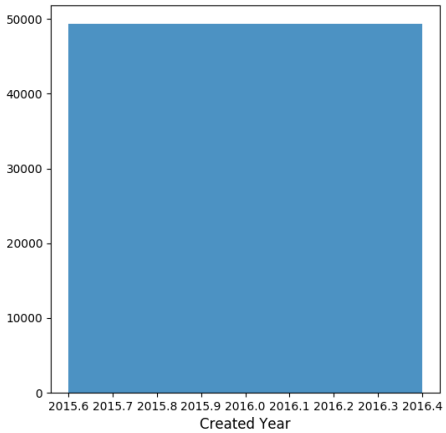
The following charts describe the dataset based on the features,

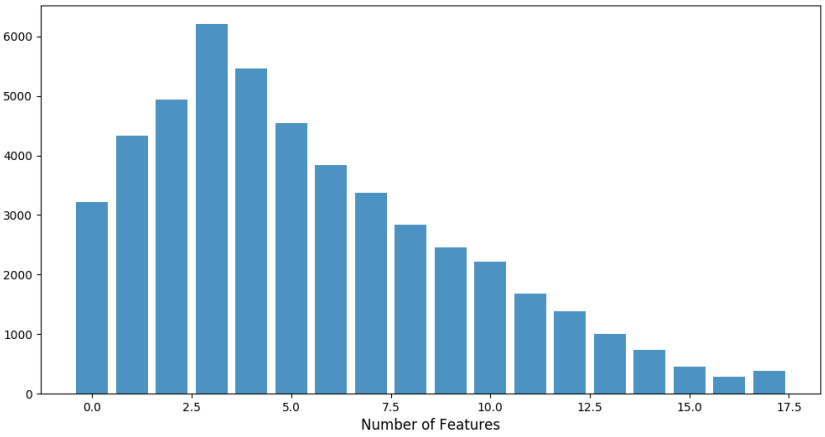

16000
14000
12000
10000
8000
6000
4000
2000
0
0
1
2
3
4
Bedrooms

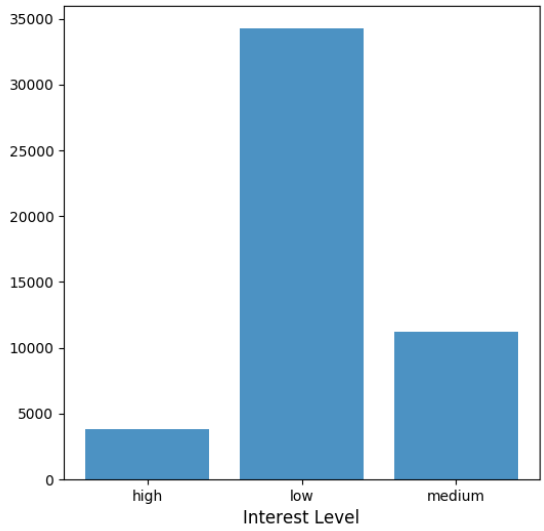

2500
2000
1500
1000
500
0
0
5
10
15
20
25
30
Created Day

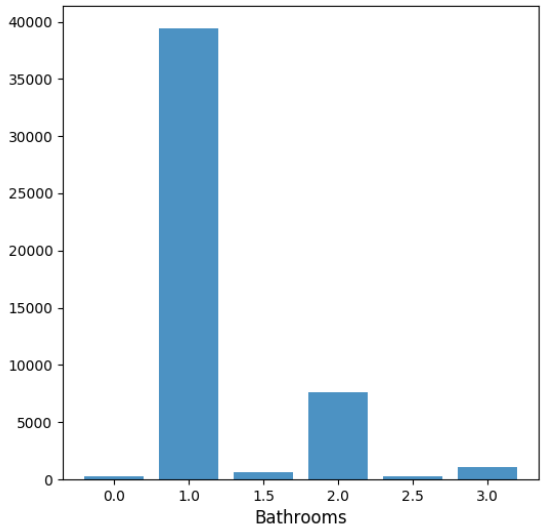

1200
1000
800
600
400
200
0
2016-04-10
2016-04-24
2016-05-08
2016-05-22
2016-06-05
2016-06-19
2016-07-03
Created

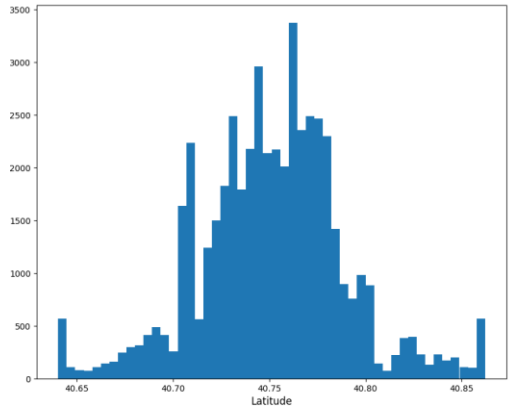

17500
15000
12500
10000
7500
5000
2500
0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
Created Month

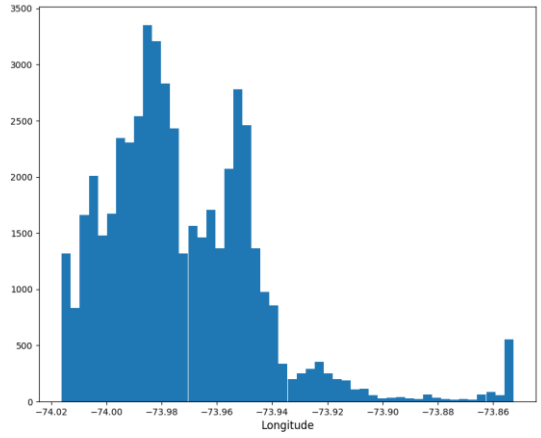

50000
40000
30000
20000
10000
0
2015.6 2015.7 2015.8 2015.9 2016.0 2016.1 2016.2 2016.3 2016.4
Created Year

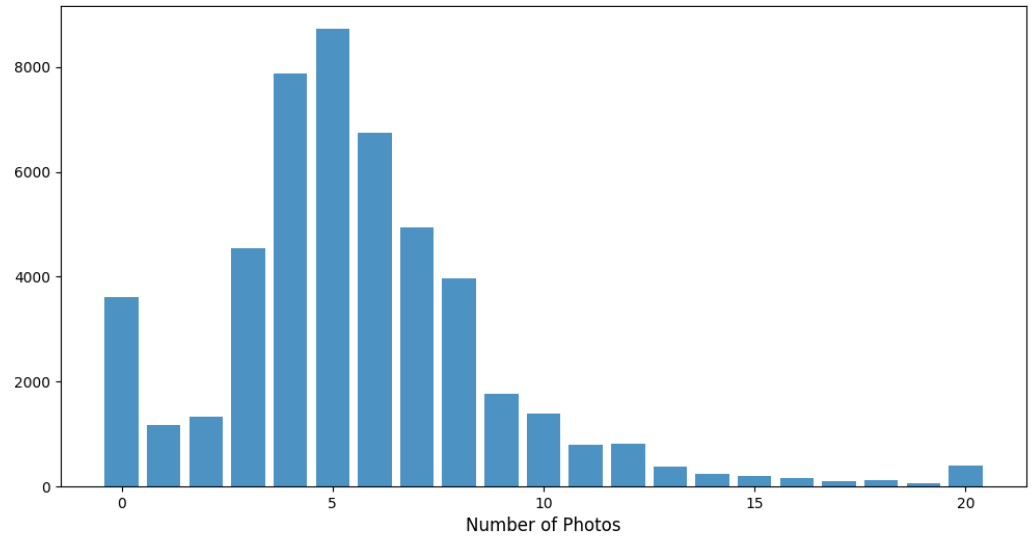

6000
5000
4000
3000
2000
1000
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Number of Features

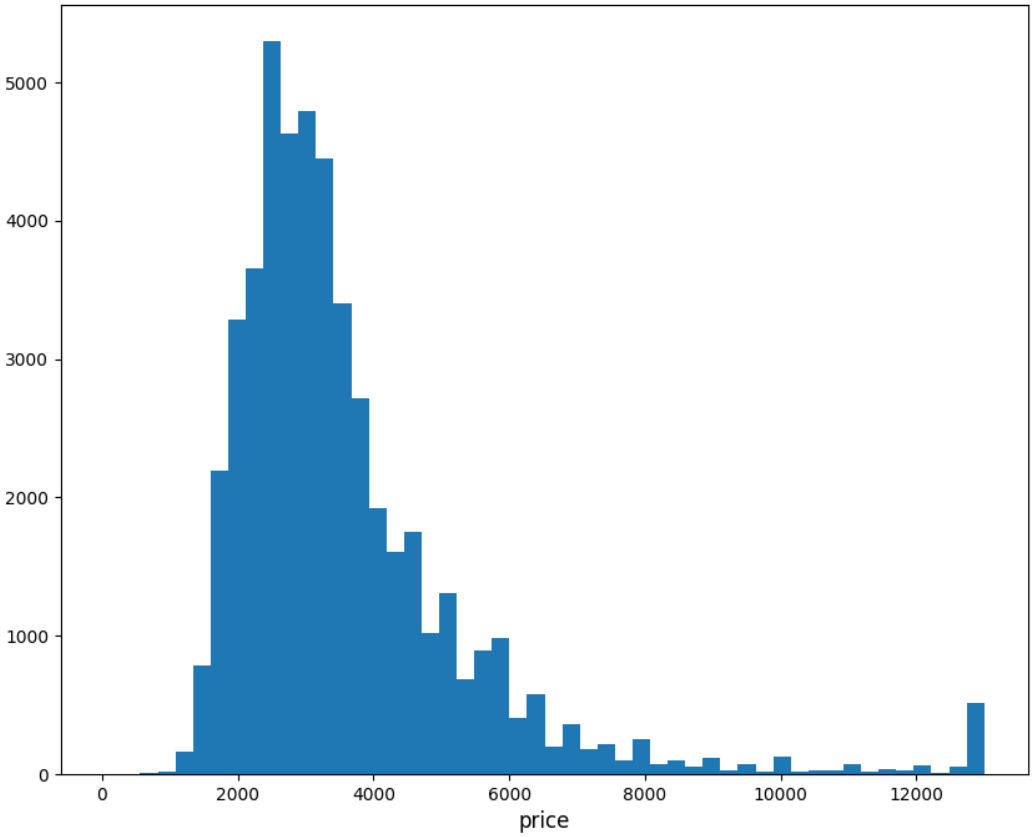

35000
30000
25000
20000
15000
10000
5000
0
high
low
medium
Interest Level


40000
35000
30000
25000
20000
15000
10000
5000
0
0.0
1.0
1.5
2.0
2.5
3.0
Bathrooms


3500
3000
2500
2000
1500
1000
500
0
40.65
40.70
40.75
40.80
40.85
Latitude


3500
3000
2500
2000
1500
1000
500
0
−74.02
−74.00
−73.98
−73.96
−73.94
−73.92
−73.90
−73.88
−73.86
Longitude





There are outliers in all the attributes and they need to be removed. The attribute created_year has only one value, so it can be removed. The target variable interest_level has a lot of difference

in the proportions between the low interest levels and high interest levels. This makes the predictability of the high class hard.

3. Pre-processing

The following pre-processing techniques are used:

1. Unnecessary attributes, such as street_address, description, display_address, building_id, manager_id are removed.
2. The maximum number of bathrooms are limited to 3 as the number of instances which have more than 3 bathrooms are low.
3. The maximum number of bedrooms are limited to 4 as the number of instances which have more than 4 bedrooms are low.
4. “created_day”, “created_month”, “created_year” attributes are extracted from “created” attribute. “created_year” attribute is removed as it has only one value.
5. Number of photos is extracted from photos attribute.
6. The maximum number of photos is limited to 20 as the number of instances which have more than 20 photos are low.
7. Number of features is extracted from features attribute.
8. The maximum number of features is limited to 17 as the number of instances which have more than 17 features are low.
9. Outliers in price attribute are removed.
10. The attribute “interest_level” is String categorical. It is indexed using String Indexer to convert it a double attribute.
11. The input features (bathrooms, bedrooms, features, latitude, longitude, longitude, photos, price, created_month, created_day) is converted into a vector using a vector assembler.
12. A PCA is used to convert possible correlated variables to linearly uncorrelated variables.
13. A Min Max Scaler is used to scale the features to the range [0, 1].

4. Proposed Solution

The attribute we need to predict has 3 classes. Hence multi-class classifiers are used to solve this problem. The following multi-class classifiers are used in our project,

1. Random Forests: Random Forests are a type of ensemble methods which use decision trees. They aim at aggregating many decision trees to combat overfitting. The parameters we tune to find the best classification are,
    a. numTrees: The number of trees to train.
    b. maxDepth: The maximum depth that each tree can grow to during training.
2. Decision Trees: Decision trees are a type of classifiers that aim to achieve classification using axis parallel cuts. They can be used for data which is not feature scaled and herein lies the effectiveness of decision trees. The parameters we tune are,
    a. maxBins: We can set the power of discretizing continuous features by increasing this parameter
    b. maxDepth: The maximum depth that each tree can grow during fitting.

3. Logistic Regression: It is a case of generalized linear models. It predicts the probability of the outcomes. The parameters we tune are,
    a. maxIterations: The maximum number of iterations that this classifier runs.
    b. Regularization: To control overfitting.

Each classifier is tuned against different parameter values. The "f1-measure" metric is used to select the best parameter values.

Cross validation technique is used to assess how the results will generalize to an independent dataset. Different values of "k" are used where k is the number of folds.

The following metrics are used to choose the best classifier among them,

1. Accuracy
2. Precision
3. Recall
4. F1 – Score
5. False positive Rate

We plan to use Python, Scala, Spark 2.2 to achieve the functionalities of our project.

5. Experimental Results and Analysis

Num of Features used	Classifier	Best Parameters	Num of Folds	Accuracy	Weighted Precision	Weighted Recall	Weighted F1-Score	Weighted False Positive Rate
9	Random Forests	Max Depth: 8 Num Trees: 20	10	0.70631	0.64375	0.70631	0.60892	0.63492
8	Random Forests	Max Depth: 8 Num Trees: 30	10	0.70499	0.63894	0.70499	0.60323	0.64431
6	Random Forests	Max Depth: 8 Num Trees: 30	10	0.70468	0.64368	0.70468	0.60865	0.63220
8	Random Forests	Max Depth: 8 Num Trees: 30	8	0.70448	0.63838	0.70448	0.60249	0.64552
9	Random Forests	Max Depth: 8 Num Trees: 30	12	0.70874	0.64708	0.70874	0.61535	0.62656
9	Random Forests	Max Depth: 8 Num Trees: 50	6	0.70719	0.63297	0.70719	0.61134	0.63467
9	Decision Trees	Max Bins: 50 Max Depth: 8	10	0.70377	0.64816	0.70377	0.64919	0.53691
8	Decision Trees	Max Bins: 50 Max Depth: 8	10	0.70154	0.63742	0.70154	0.64195	0.55120
6	Decision Trees	Max Bins: 50 Max Depth: 8	10	0.69504	0.63703	0.69504	0.64461	0.53891

8	Decision Trees	Max Bins: 50 Max Depth: 8	8	0.70154	0.63742	0.70154	0.64195	0.53691
9	Decision Trees	Max Bins: 35 Max Depth: 8	12	0.70302	0.65056	0.70302	0.65750	0.51018
9	Decision Trees	Max Bins: 50 Max Depth: 8	6	0.70012	0.64950	0.70012	0.65731	0.50682
9	Logistic Regression	Max Iterations: 100 Regularization: 0.0	10	0.69291	0.63518	0.69291	0.60509	0.63129
8	Logistic Regression	Max Iterations: 100 Regularization: 0.0	10	0.69382	0.64130	0.69382	0.60533	0.62942
6	Logistic Regression	Max Iterations: 100 Regularization: 0.0	10	0.68956	0.60659	0.68952	0.59688	0.64242
8	Logistic Regression	Max Iterations: 100 Regularization: 0.0	8	0.68382	0.64130	0.69382	0.60533	0.62942
9	Logistic Regression	Max Iterations: 100 Regularization: 0.0	12	0.68926	0.62201	0.68926	0.60021	0.63707
9	Logistic Regression	Max Iterations: 100 Regularization: 0.0	6	0.68968	0.62493	0.68968	0.60321	0.63812

The following observations are made from the above table,

1. The best parameters for Random Forest classifier are
    a. numTrees:30
    b. maxDepth:8
    c. Maximum Accuracy attained:70.87%
2. The best parameters for Decision trees classifier are
    a. maxBins:50
    b. maxDepth:8
    c. Maximum Accuracy attained:70.37%
3. The best parameters for Logistic regression classifier are:
    a. Max Iterations:100
    b. Regularization:0
    c. Maximum Accuracy attained:69.38%

4. The Random Forest classifier's prediction accuracy is the best, followed by the decision tree accuracy. The Logistic regression classifier has the least accuracy among them.
5. The precision of the Decision trees and Random Forests are similar and almost equal. Logistic Regression's precision is very low. Random forests and decision trees outperforms the logistic regression in precision metric.
6. The Recall of Random Forests and Decision trees are high compared to the Recall of Logistic Regression Classifier. Hence Random forests and decision trees outperform logistic regression.
7. The F1 score of Random Forest and Logistic Regression are low while the F1 score of the decision trees is the highest among them. Logistic regression has the least F1 score as expected.
8. The false positive rate of the Decision trees is the least among the classifiers. The FPR of Random Forests and Logistic Regression are similar.

6. Conclusion

After analyzing the results shown in the above table, we conclude that Random Forests works the best for our dataset with the following set of parameter settings: Number of features used = 9, Max Depth = 8.

We achieved an accuracy of 70.61% accuracy using Random Forest classifier as compared to decision tree which gave an accuracy of 70.3% and logistic regression which gave us 69.29% accuracy. Kaggle gave us a score based on the log loss of the classifier. Our random forest model got a log loss score of 0.66318 and the decision tree got a log loss score of 0.792996.

Therefore, Random forest is our choice of classifier which will perform best on our dataset to predict the interest levels of an apartment listing.

7. References

1. Spark Classification and regression docs: https://spark.apache.org/docs/latest/ml-classification-regression.html#classification
2. Matplotlib docs: https://matplotlib.org/gallery/index.html
3. Spark Feature selection docs: https://spark.apache.org/docs/latest/ml-features.html
4. Katarzyna Stapor: Evaluation of classifiers: current methods and future research directions, pp. 37–40 ISSN 2300-5963 ACSIS, Vol. 13