SpringBoard's Data Science Intensive program using Python

Capstone Project Report

Predicting interest level of new rental listings for RentHop

By Sidhant Bhardwaj

Problems to be solved and Motivation:

Finding a perfect place to rent usually results in browsing of endless listing on the internet. But while looking for the perfect apartment is difficult enough, structuring and making sense of all available real estate data programmatically is even harder. Using the data provided by RentHop, we can predict the interest level of the new rental listings based on the different characteristics and features of the listings thereby helping people to narrow down their search for apartments according to the needs. Doing so will also help RentHop better handle fraud control, identify potential listing quality issues and allow owners and agents to better understand renter's needs and preferences.

Data:

The data was obtained from one of Kaggle Competitions.

https://www.kaggle.com/c/two-sigma-connect-rental-listing-inquiries/data

Important Fields and Information in the Dataset:

The data provided by Kaggle consists of different files such as train.json, test.json, images_sample.zip. The data is about apartments located in the New York City. The different features contained in the Data are as follows:

- Bathrooms: The number of bathrooms
- Bedrooms: The number of bedrooms
- Display address: The address of the listings
- Features: The list of features about the apartment

- Latitude
- Longitude
- Manager_id
- Price: The price of each apartment in USD
- building_id
- Description
- created
- Interest_level: The target variable

The objective of the project is to predict the interest level of the listings i.e. Interest level is our target variable. Some of the important features given in our data set can be used for this prediction.

With the help of decision tree classifier, we were able to find the important features in the data set and remove the unimportant ones.

Features Used in the models:

1. Bathrooms
2. Bedrooms
3. Price
4. Display_address
5. Num_photos
6. Num_features
7. Manager_id
8. Latitude
9. Longitude
10. Description
11. Created_day

12. Created_month
13. Feature dummy variables

Data Wrangling and Cleaning:

The original data had three interest levels (High, Medium and low). For our purpose, we combined the medium and high levels together so that the data now had only two levels i.e. High and Low. The **Price** variable in the data had a couple of outliers which was affecting the prediction results. We trimmed down the outlier values to the normal values. By looking at the distribution of **latitudes** and **longitudes**, it was noted that both the variables had outliers and thus were trimmed down as well. I cleaned the **Display_address** variable by stripping all the character like -, &, #,*, ! and making all the alphabets lowercase so that we don't have any duplicates in the field. I was thus able to find out the top addresses for Low and High levels in the Exploratory Data Analysis. I performed the process of cleaning for the **Features** variable to find the top features in the dataset.

The training data has approximately 49k listings and there are no blank or empty values in the data.

No additional datasets or sources have been used other than the provided by Kaggle.

Feature Engineering:

Some of the features have been used as it is such as bedrooms, bathroom and price. I created dummy variable for **Features** variable for only those features that have more than 1000 listings to avoid high dimensions and to help in increasing the prediction score. I also created new features from **Features** and

photos variables, consisting of number of photos and number of features. From the **created** feature, I made three more features called **created_year**, **created_month**, **created_day** and **created_hour.** From **description** feature, I created a new feature containing the count of number of words in the description. I used label encoding and then one hot encoding on the **display_address** field to

Convert the text data into numerical data for better classification. I used one hot encoding for **manager_id** as well.

Exploratory Data Analysis:

The data consists of numerical, text and categorical features. There are 15 features and approx 50k listings in the dataset. The target variable is '**interest_level'.** The low interest level has the highest number of listing of around 35k.

Bathrooms: Most of the listings have only 1 bathroom and have interest level 'low'.

Bedrooms: Most of the listings have 1 bedroom followed closely by 2 bedrooms and then 3. We see that for 1 bedroom, the interest level is lowest and for 2 bedrooms the interest level is highest. Also for 2 bedrooms, the interest level is medium.

Price: By analyzing price feature in our dataset through a scatterplot, we found that there were some outliers present so we trimmed them down. The distribution of price was between 1000k and 13k in USD. Listings with low interest level have an average high price and listings with high interest levels have an average low price. By looking at the price, bedroom and bathrooms, we see that 'high' interest level have an average price of 2400 USD with 2

bedrooms and 1 bathrooms. 'Low' interest level has an average price of 3300 USD with 1 bedroom and 1 bathroom.

Latitudes: By analyzing the latitudes, we found that there were some outliers, so we trim it down. We found the distribution of latitudes between 40.65 and 40.85.

Longitudes: The longitude feature was trimmed downed to remove outliers. The distribution of longitudes was between -74.02 to -73.85.

I have done basemapping for listings based on longitudes and longitudes in the EDA notebook. I found that the low and high listings could not be distinguished solely on the latitudes and longitudes.

Display_address: I cleaned the display address feature and found the top addresses with majority listing in the dataset for each of the interest_levels. I created wordcloud for each of the levels for better understanding of the frequency of addresses. Some of the top addresses are Wall st,Broadway,E 34^{th} street.

Features: I used wordcloud for analyzing the features in the listing and using nltk package found the top 10 features in the dataset. Some of the top features are Elevators, Dogs_Allowed, Cats_allowed.

Num_Features: I created this feature using **Features** variable and found that majority of the listings had 3 features, followed by 4.

Num_photos: This feature contains the number of photos each listing has. I found out that majority of the listing had 5 photos, followed by 4.

Created_year: All the data was collected during the year 2016 from April to June.

Created_month: Majority of the listings with low interest level were created in the month of June (around 11k listings). The high and low interest level listings seems to have been created uniformly in the three months.

Created _Day: Majority of the listings with low interest level were created during 12 and 21 day of the months. The high and low interest level seems to be uniform throughout the month.

Created_hour: From the Exploratory data analysis, majority of the listings were created during early hours of the morning. (2 am to 6 am)

Approach and Findings:

The initial exploration and findings were concurrent with our claims that the features given in the dataset as well as the new features obtained through feature engineering would help us achieving good accurate results in predicting the interest level of the listings. The next step was to build machine learning models and obtain predictions on the test set. The first machine learning algorithm I applied was scikit learns Logistic Regression. Since we had converted out target variable 'Interest_level' which had three classes namely high, medium and low into two classes that is high (medium/high) and low, Logistic Regression was the appropriate approach.

In the project, our goal was to predict whether based on the characteristics of the listings, the interest level will be high or low. We have a binary classification problem, where the goal is to find the best line that has the maximum probability of classifying unseen points correctly. Since the ratio of listings for low and high are 75% to 25% i.e. 75% for low and 25% for high, we cannot use accuracy as metrics for our models. Instead with the help of confusion matrix, I was able to measure precision, recall and f1 score for the

models. F1 score will be a good metrics for models accuracy since it's a weighted average of precision and recall.

To find the best regularization parameter for our mode, we implemented gridsearch and found that a value of 10 is the most optimum. By splitting the data into train and test sets and by measuring the f1 score, we find that the f1 score for Logistic regression model is 69%.

The second model that I applied is the Support vector machine (SVM) model. The SVM model aims to maximize the margin between the positive and

Negative examples. I am using SVM with RBF (Radial basis function) to see how linearly inseparable data can be classified. By using the same features that we used in the Logistic Regression model, I found the f1 score to be around 62% (Lower than Logistic Regression).

The third model that I implemented is Extreme Gradient Boosting (XGBoost). I used this model as the implementation of the algorithm results in increased in efficiency time and memory resources. XGBoost model automatically handles missing data values and helps in continued training so that further boost to already fitted model on new data is possible.

By tuning the parameters of the model i.e. max_dephth, min_child_weight, we were able to increase our score. By looking at the confusion matrix, I found the f1 score of XGBoost model to be 79% which is greater than logistic regression (69%) and SVM (62%). Thus the best model for classification of listings into low and high would be XGBoost.