Coursera - IBM Data Science Specialization

Sebastian Mack

mack.seb@gmail.com

Keyword: Data Science; Prediction; GBM; Tuning

---

1. Introduction

The background for this capstone project is to have the opportunity to be as creative as possible and come up with ideas to leverage the Foursquare location data to come up with a problem that can make a benefit from the Foursquare location data.

1.1. Project Overview

As the field of study, I selected an interesting topic from the rental industry that caught my attention when I was exploring topics on the data science platform Kaggle. The initiator of this contest is RentHop (a portfolio company of TwoSigma Ventures) which has the objective to make apartment search smarter by using data to sort rental listings by quality (refer to https://www.renthop.com/).

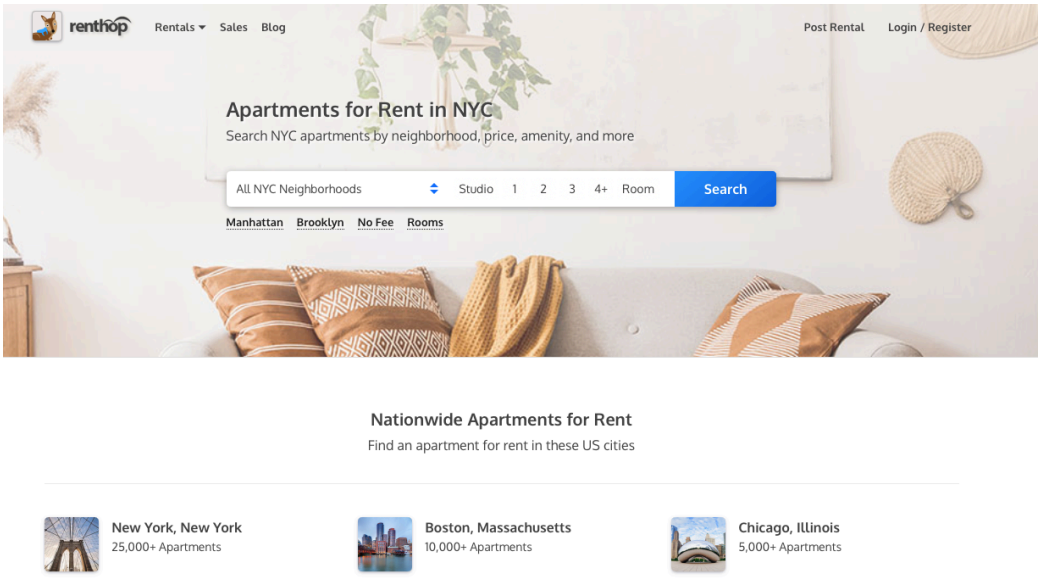


Fig. 1: Renthop Portal

1.2. Problem Statement

In this project we want to predict how popular an apartment rental listing is based on the listing content like text description, photos, number of bedrooms, price, etc. The data comes from renthop.com, an apartment listing website. These apartments are located in New York City. The target variable, interest level, is defined by the number of inquiries a listing has in the duration that the listing was live on the site. For me personally this problem is very interesting because it represents a real life problem that not only Renthop but many other companies are facing right now. In this situation companies are already acquiring and collecting data with their existing services but still struggle to create a business value out of it. In this particular case we can see that by developing a model that can predict how much interest a new rental listing on RentHop will receive, new business values can be proposed. Both the consumer and the merchants could benefit from such a situation. Furthermore, the offering company could gain new clients with this value adding service.

1.3. Approach

The problem that is to be solved can be described as a supervised machine learning problem because the model will be trained based on a given feature. Since this target variable has categorical values it can be further characterized as a classification problem. A problem of this type can be solved and modeled with various approaches but in this study the most promising will be applied which will be a desicion tree model. I will outline the most important steps within my theoretical workflow in order to find a solution for the described problem. A structured approach will be helpful for a reasonable result and to have a scientific discussion on the final model. My planned workflow includes the following steps:

1.3.1. **Exploratory Data Analysis (EDA):** As a first step I will explore the provided data and make an analysis. This includes summarizing properties and visualizing important outcomes. It will be also very useful to identify features that are relevant for the model and also to give hints for transformations that are required for fitting the data.

1.3.2. **Feature Engineering**: Within this step the knowledge gained from the previous step will be applied to clean the data set and to select important features as well as to define new ones.

1.3.3. **Train Baseline Model**: When the previous step is completed, the obtained transformed and extended dataset can be used to train the baseline model (or benchmarking model). It will be an implementation of an Ensemble model with gradient boosting for classification.

1.3.4. **Tune parameters**: Since there are lots of parameters available in order to train a sophisticated model, it will be necessary to repeat some of the steps and fine tune the model until it is able to produce the desired scores.

1.3.5. **Evaluate Metrics**: In the last step, the results and scores of all generated models will be evaluated and compared to one another.

1.4. Metrics

The chosen evaluation metric will be the root mean squared error (RMSE)

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

2. Data Description

For this project the publicly available data for the" Two Sigma Connect: Rental Listing Inquiries" Kaggle competition will be considered. It consists out of the following files:

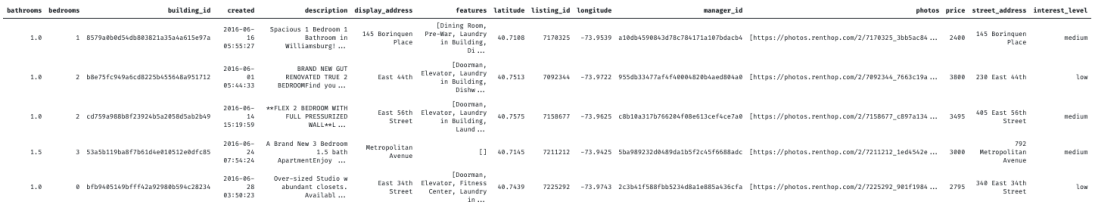
(https://www.kaggle.com/c/two-sigma-connect-rental-listing-inquiries/data)

- train.json: the training set
- test.json: the test set
- samplesubmission.csv: a sample submission file

For both the train and the test set, the following features are provided in thecompetition data. In total we have 49352 datapoints.

- bathrooms: number of bathrooms
- bedrooms: number of bathrooms
- buildingid
- created
- description
- displayaddress
- features: a list of features about this apartment
- latitude
- listingid
- longitude
- managerid
- photos: a list of photo links
- price: in USD
- streetaddress
- interestlevel: this is the target variable. It has 3 categories: 'high', 'medium', 'low'

In the following figures we can find the head of the data frame as well as a statistical summary of the numerical features in the dataset.



bathrooms	bedrooms	building_id	created	description	display_address	features	latitude	listing_id	longitude	manager_id	photos	price	street_address	interest_level
1.0	1	8579a0b0d54db803821a35a4a615e97a	2016-06-16 05:55:27	Spacious 1 Bedroom 1 Bathroom in Williamsburg! ...	145 Borinquen Place	[Dining Room, Pre-War, Laundry in Building, Di ...	40.7108	7170325	-73.9539	a10db4590843d78c784171a107bdacb4	[https://photos.renthop.com/2/7170325_3bb5ac84 ...	2400	145 Borinquen Place	medium
1.0	2	b8e75fc949a6cd8225b455648a951712	2016-06-01 05:44:33	BRAND NEW GUT RENOVATED TRUE 2 BEDROOMFind you ...	East 44th	[Doorman, Elevator, Laundry in Building, Dishw ...	40.7513	7092344	-73.9722	955db33477af4f40004820b4aed804a0	[https://photos.renthop.com/2/7092344_7663c19a ...	3800	230 East 44th	low
1.0	2	cd759a988b8f23924b5a2058d5ab2b49	2016-06-14 15:19:59	**FLEX 2 BEDROOM WITH FULL PRESSURIZED WALL**L ...	East 56th Street	[Doorman, Elevator, Laundry in Building, Laund ...	40.7575	7158677	-73.9625	c8b10a317b766204f08e613cef4ce7a0	[https://photos.renthop.com/2/7158677_c897a134 ...	3495	405 East 56th Street	medium
1.5	3	53a5b119ba8f7b61d4e010512e0dfc85	2016-06-24 07:54:24	A Brand New 3 Bedroom 1.5 bath ApartmentEnjoy ...	Metropolitan Avenue	[]	40.7145	7211212	-73.9425	5ba989232d0489da1b5f2c45f6688adc	[https://photos.renthop.com/2/7211212_1ed4542e ...	3000	792 Metropolitan Avenue	medium
1.0	0	bfb9405149bfff42a92980b594c2823a	2016-06-28 03:50:23	Over-sized Studio w abundant closets. Availabl ...	East 34th Street	[Doorman, Elevator, Fitness Center, Laundry in ...	40.7439	7225292	-73.9743	2c3b41f588fbb5234d8a1e885a436cfa	[https://photos.renthop.com/2/7225292_981f1984 ...	2795	340 East 34th Street	low

Fig. 2: Head of Competition Data

	bathrooms	bedrooms	latitude	listing_id	longitude	price
count	49352.00000	49352.000000	49352.000000	4.935200e+04	49352.000000	4.935200e+04
mean	1.21218	1.541640	40.741545	7.024055e+06	-73.955716	3.830174e+03
std	0.50142	1.115018	0.638535	1.262746e+05	1.177912	2.206687e+04
min	0.00000	0.000000	0.000000	6.811957e+06	-118.271000	4.300000e+01
25%	1.00000	1.000000	40.728300	6.915888e+06	-73.991700	2.500000e+03
50%	1.00000	1.000000	40.751800	7.021070e+06	-73.977900	3.150000e+03
75%	1.00000	2.000000	40.774300	7.128733e+06	-73.954800	4.100000e+03
max	10.00000	8.000000	44.883500	7.753784e+06	0.000000	4.490000e+06

Fig. 3: Description Numeric Features Competition Data

In addition to the previously described data, we use the Foursquare api (https://developer.foursquare.com/places-api) to gather additional information for the respective geographic coordinates (latitude, longitude) that can be found for each row in the main data frames.

The hypothesis of this project is that we can use the foursquare api for exploring a location for improving the model. In order to do so, we have to specify an url and pass the coordinates of the relevant data point and make a request. The resulting json response can be extracted and put into a new pandas data frame. The new created features are:

- categories: number of unique foursquare venue categories
- distance: mean distance of foursquare venues

	index	categories	mean_distance
5611	11283	17.0	57.421053
5612	112830	5.0	72.200000
5613	112831	9.0	58.333333
5614	112832	13.0	48.533333
5616	112838	15.0	67.800000

Fig. 4: Head of Crawled Foursquare Data

	index	categories	mean_distance
count	6985.000000	6985.000000	6985.000000
mean	76069.943021	8.201432	66.905929
std	31460.057838	5.705742	13.218888
min	66.000000	1.000000	7.000000
25%	64862.000000	4.000000	59.285714
50%	69575.000000	7.000000	67.192308
75%	113743.000000	12.000000	74.714286
max	118406.000000	38.000000	99.000000

Fig. 5: Description of numeric features of Crawled Foursquare Data

Both of the two previously introduced data frames are merged into a single data frame for consistency. Afterwards additional steps like data cleaning and feature selection can be performed as well as engineering of features. We can extract with the help from pandas datetime module new features from the date and time. In addition, we can use word count techniques from the text feature. The total number of extracted additional features is only 6985 because of quota limits for free usage of the foursquare api.

3. Methodology

In this chapter we discuss the methodology that has been applied to the given problem. It includes the steps for preprocessing the data in order to address any abnormalities or characteristics. Furthermore, it documents the implemented metrics, algorithms and techniques. The general methodology is described by the Team Data Science Process (TDSP) shown in the following figure:

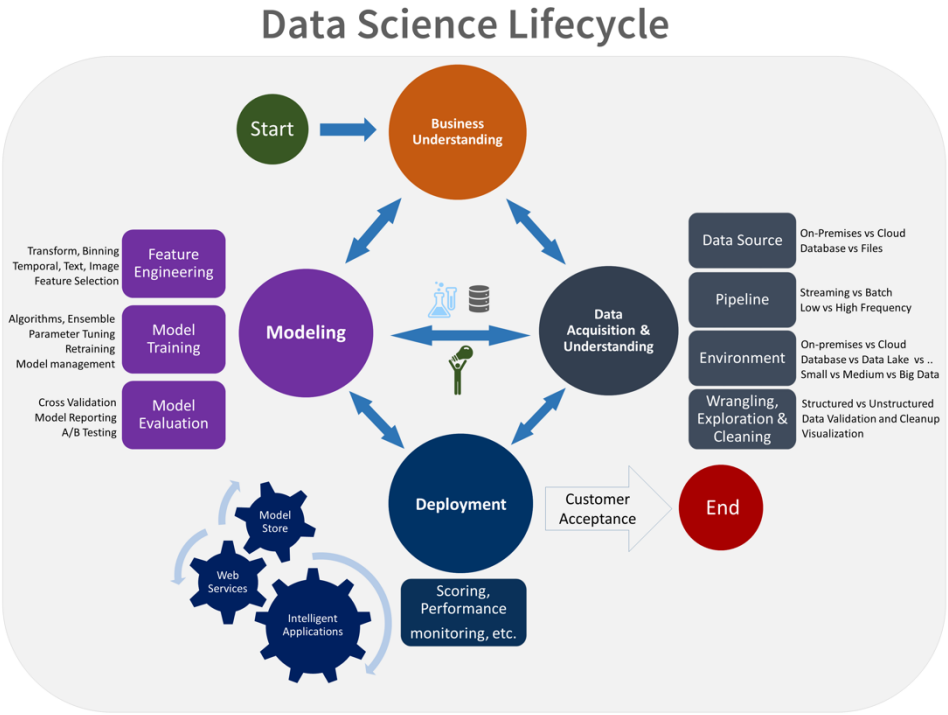


Fig. 6: Data Science Lifecycle (TDSP)

3.1. Exploratory Data Analysis (EDA)

Before we begin to model the given problem, an analysis of the given data sets needs to be performed in order to understand which algorithms are going to be used in the next steps of the project. This section includes a data exploration which describes characteristic properties of the data as well as visualizations that help to summarize the most important outcomes.

As can be seen from the following figures, there are significantly more samples with low interest levels (5000) than medium (1500) and low (500) interest levels. The map plotted with folium illustrates this observation quite well.

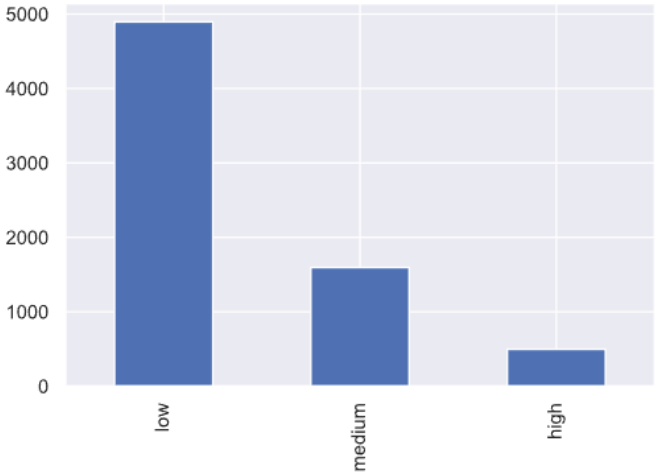
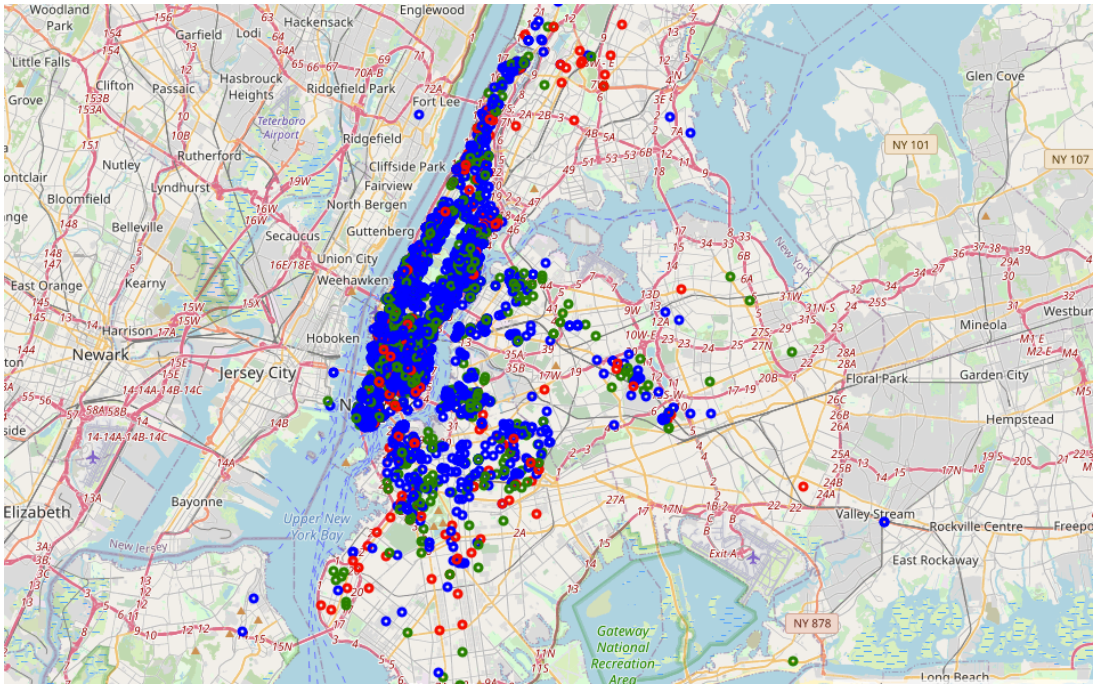
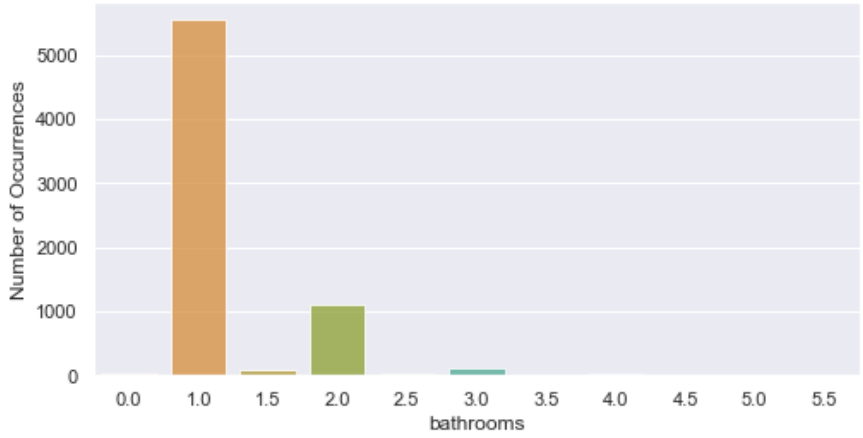


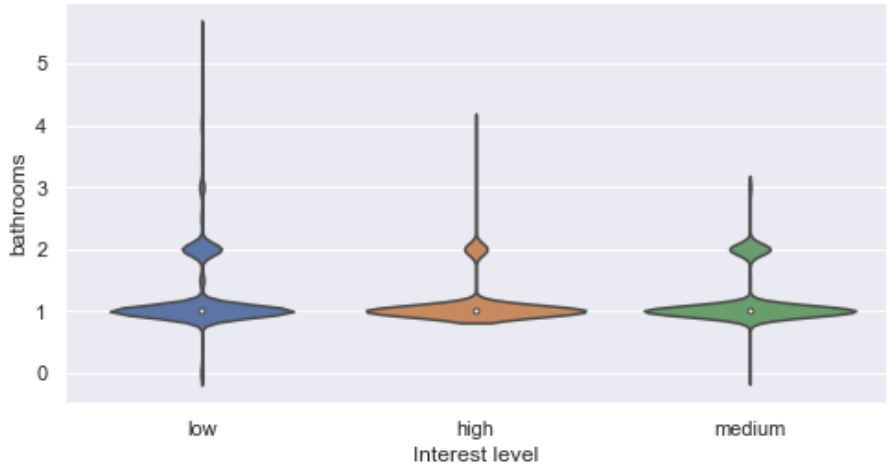
Fig. 7: Number of Occurrences of target



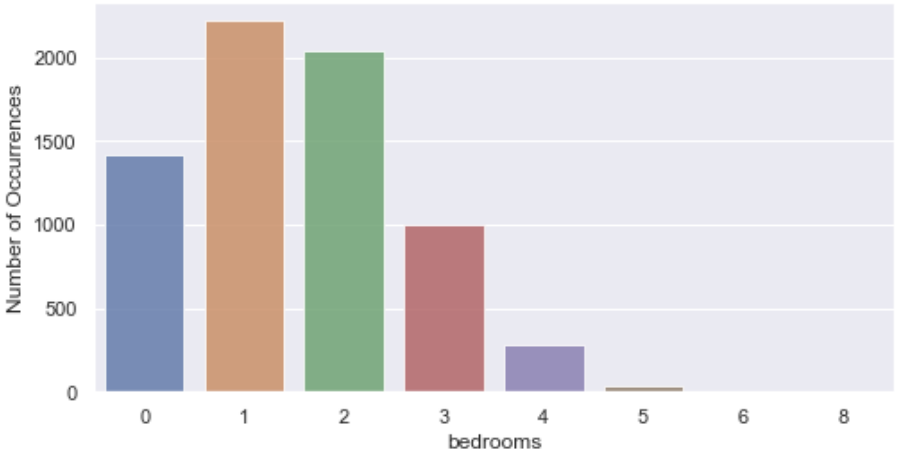
In order to get a better understanding for the predicting features we plot their distributions and relations to the target variable “interest level”.

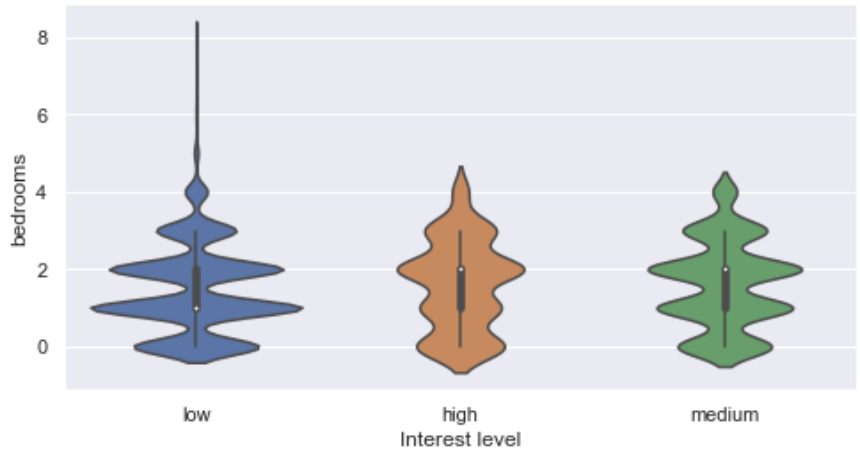
Bathrooms:



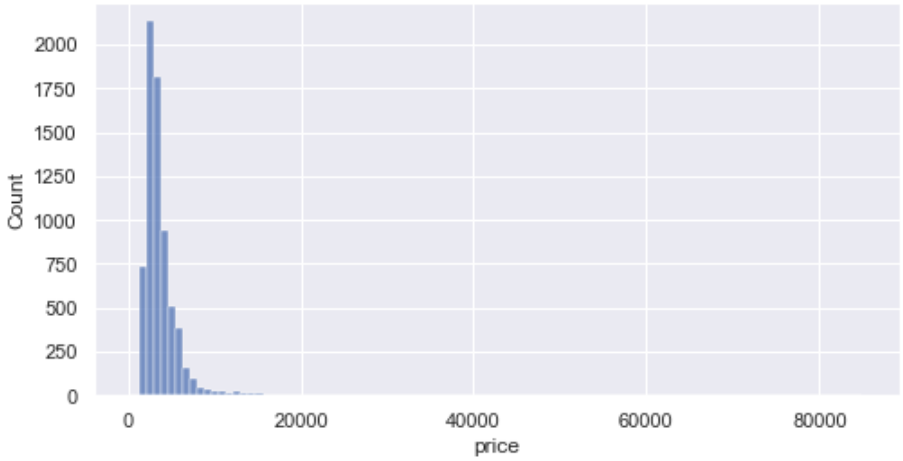


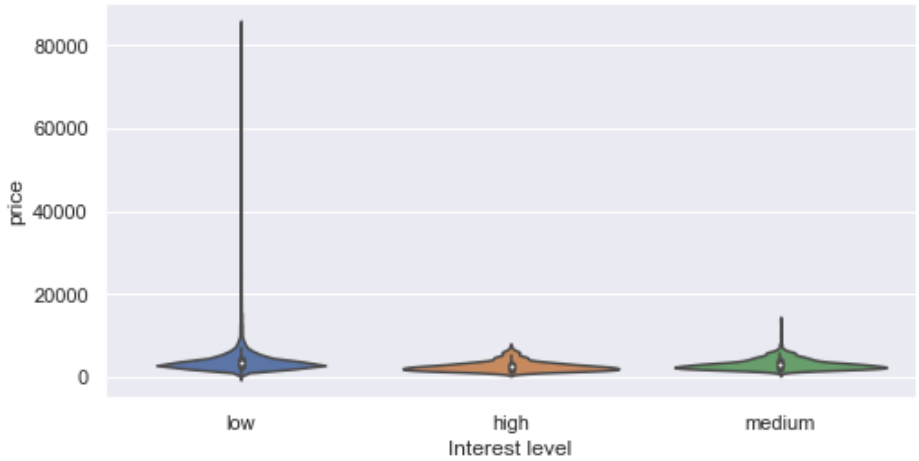
Bedrooms:



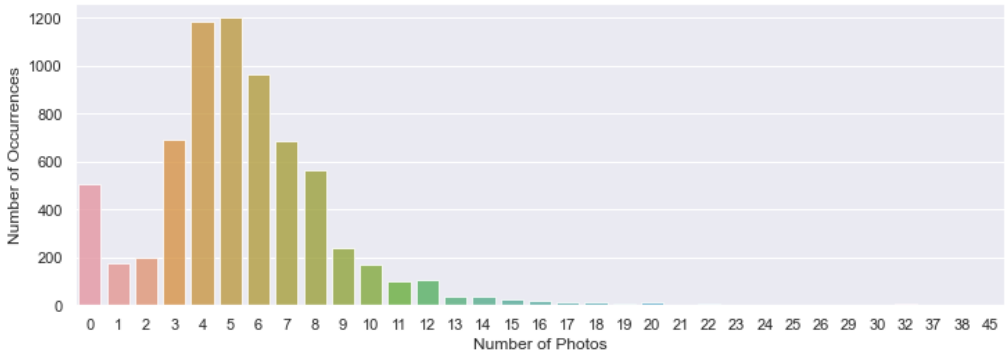

bedrooms
8
6
4
2
0
low
high
medium
Interest level

Price:

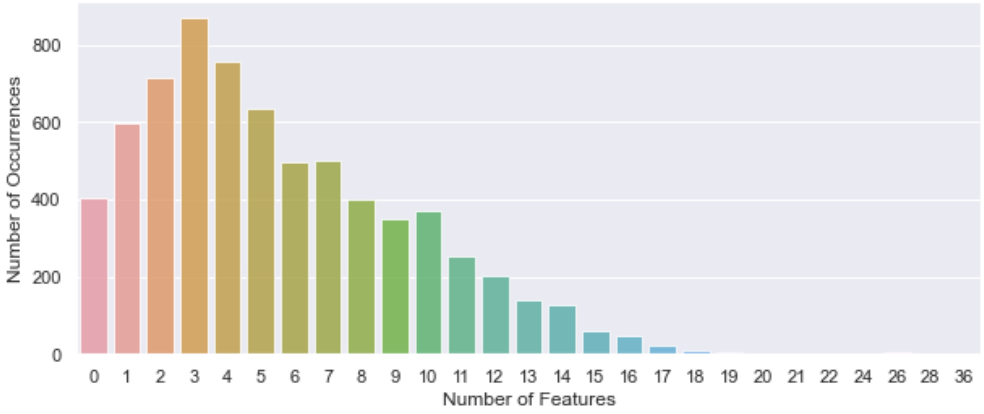

Count
2000
1750
1500
1250
1000
750
500
250
0
0
20000
40000
60000
80000
price


price
80000
60000
40000
20000
0
low
high
medium
Interest level

Number of photos:

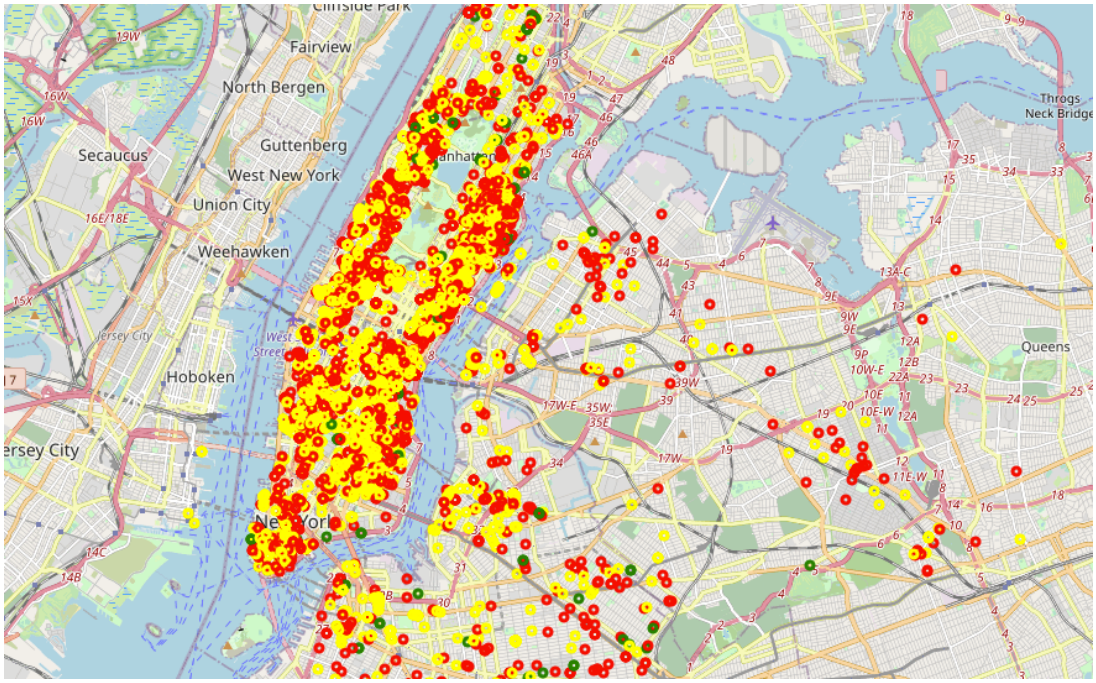


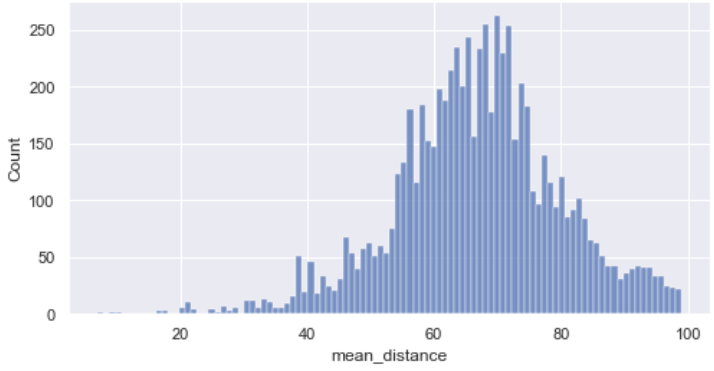
Number of features:

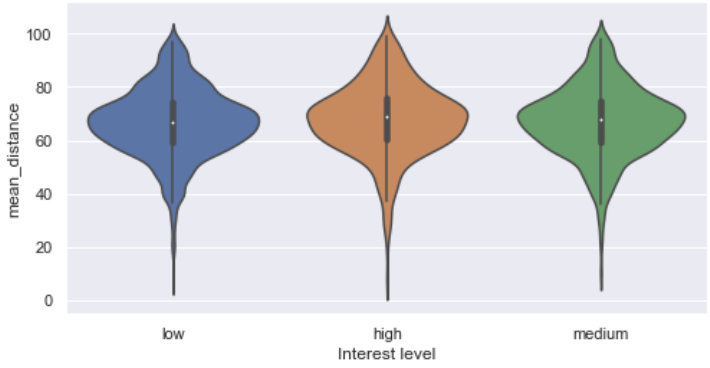


In addition, we examine the foursquare data features. For the geographical visualisations, green points represent a low value, yellow a medium value and red a high value.

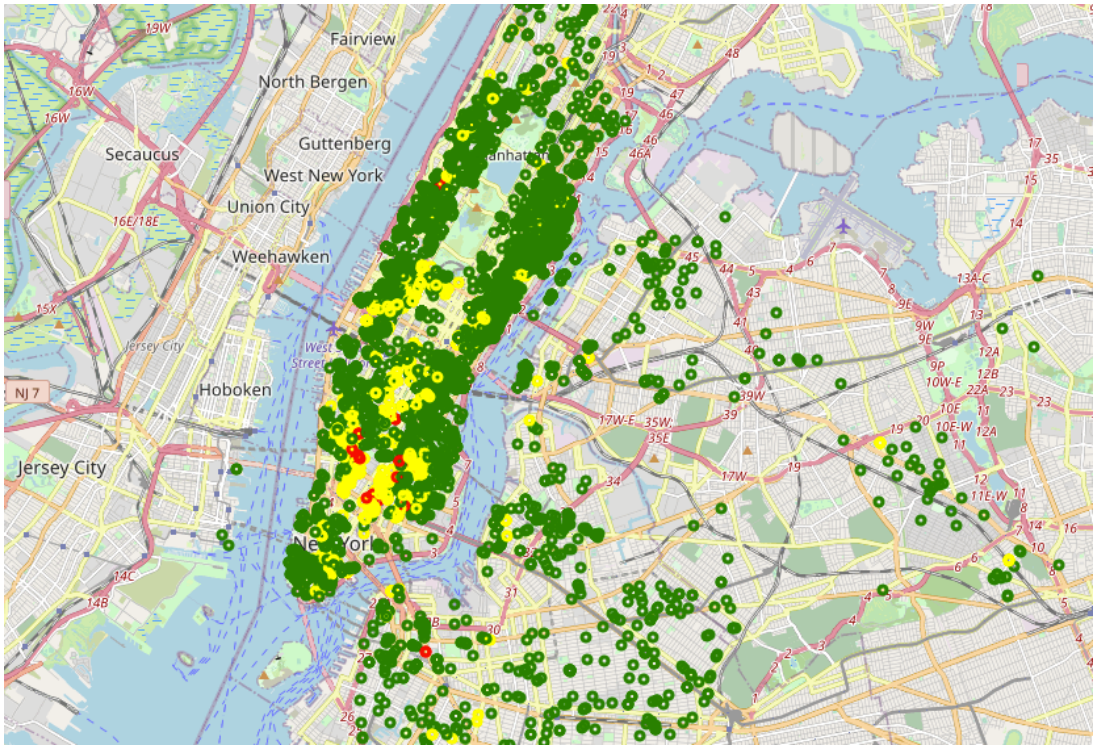
Mean distance of foursquare venues

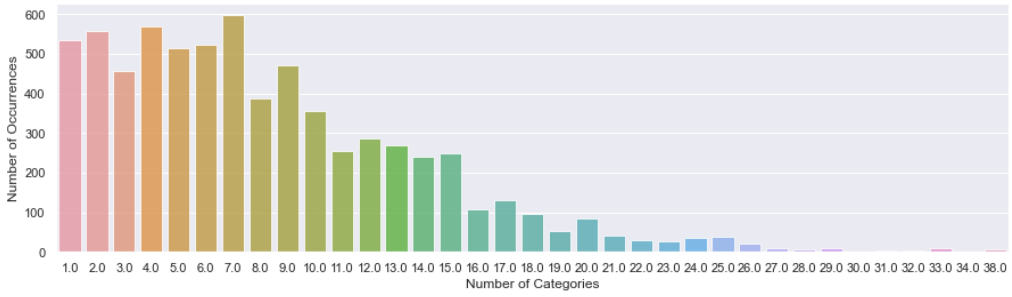


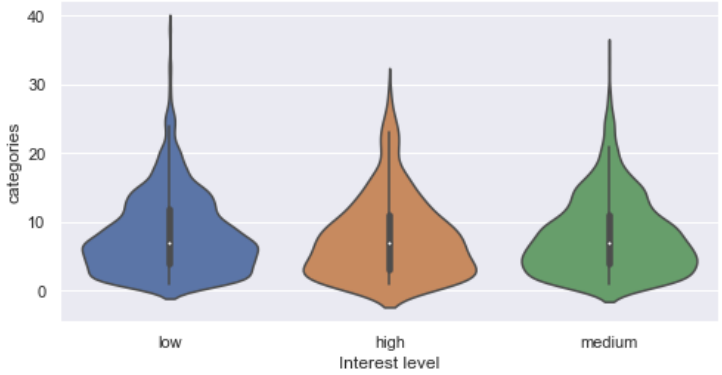




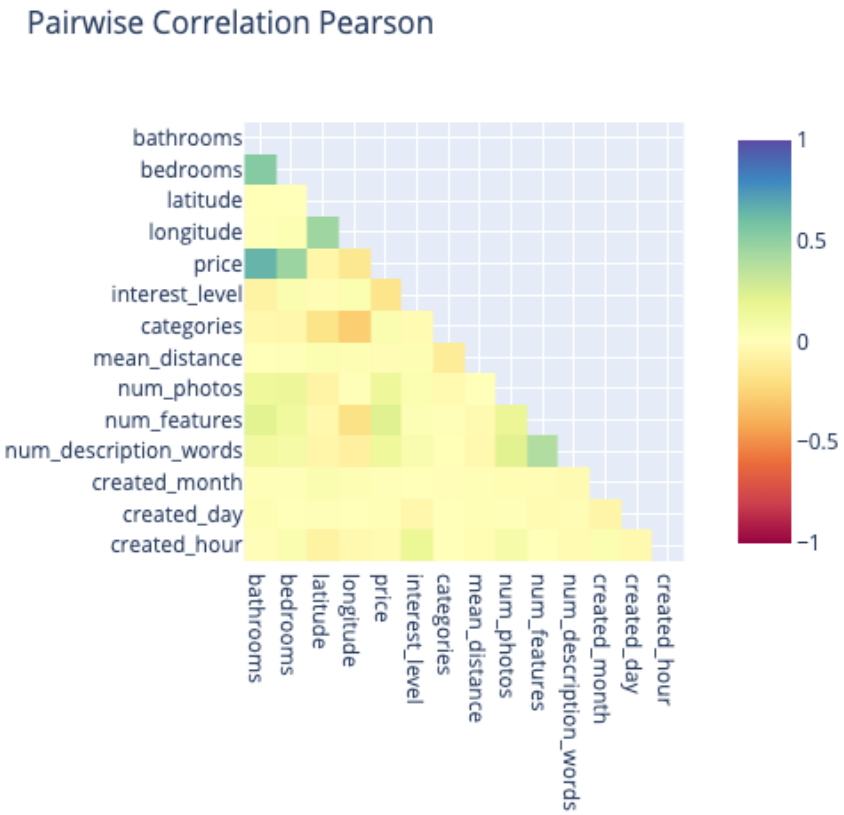
Number of unique foursquare venue categories







3.2. Correlation



Unfortunately, the newly created features "categories" and "mean_distance" have very low correlation values with the target variable (-0.03 and 0.02 respectively). Maybe the model can identify more complex patterns.

3.3. Algorithms

For the modelling process of this project, we will use the h2o.ai framework. H2O is a Java-based software for data modeling and general computing. The H2O software is many things, but the primary purpose of H2O is as a distributed (many machines), parallel (many CPUs), in memory (several hundred GBs Xmx) processing engine. (http://docs.h2o.ai/h2o/latest-stable/h2o-py/docs/intro.html)

In particular the H2OXGBoostEstimator will be used to train a baseline model and compare it to an advanced model with the newly created features. XGBoost is a supervised learning algorithm that implements a process called boosting to yield accurate models. Boosting refers to the ensemble learning technique of building many models sequentially, with each new model attempting to correct for the deficiencies in the previous model. In tree boosting, each new model that is added to the ensemble is a decision tree. XGBoost provides parallel tree boosting (also known as GBDT, GBM) that solves many data science problems in a fast and accurate way. For many problems, XGBoost is one of the best gradient boosting machine (GBM) frameworks today. (https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/xgboost.html)

H2O's Gradient Boosting Algorithms follow the algorithm specified by Hastie et al (2001):

Initialize $f_{k0} = 0, k = 1, 2, \ldots, K$

For $m = 1$ to M:

1. Set $p_k(x) = \frac{e^{f_k(x)}}{\sum_{l=1}^{K} e^{f_l(x)}}, k = 1, 2, \ldots, K$
2. For $k = 1$ to K:
   a. Compute $r_{ikm} = y_{ik} - p_k(x_i), i = 1, 2, \ldots, N$
   b. Fit a regression tree to the targets $r_{ikm}, i = 1, 2, \ldots, N$, giving terminal regions $R_{jim}, j = 1, 2, \ldots, J_m$
   c. Compute $\gamma_{jkm} = \frac{K-1}{K} \frac{\sum_{x_i \in R_{jkm}} (r_{ikm})}{\sum_{x_i \in R_{jkm}} |r_{ikm}|(1-|r_{ikm}|)}, j = 1, 2, \ldots, J_m.$
   d. Update $f_{km}(x) = f_{k,m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jkm} I(x \in R_{jkm})$.

Output $\hat{f}_k(x) = f_{kM}(x), k = 1, 2, \ldots, K$

4. Results

The following sections summarize the results for the baseline and extended model.

4.1. Baseline Model

ModelMetricsMultinomial: xgboost

** Reported on train data. **

MSE: 0.09305668048815223

RMSE: 0.30505193080548143

LogLoss: 0.3075453925310763

Mean Per-Class Error: 0.1623435251635366

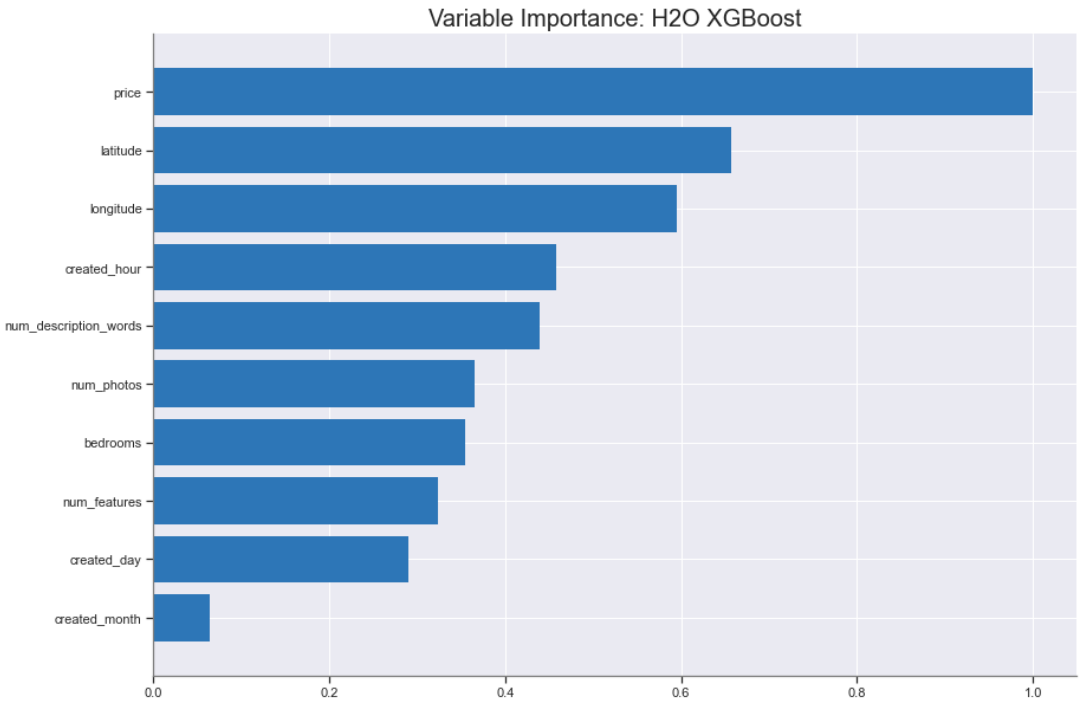
** Reported on validation data. **

MSE: 0.21445060605228666

RMSE: 0.4630881191007675

LogLoss: 0.6351238727480131

Mean Per-Class Error: 0.5622636243951696



	high	low	medium	Error	Rate
0	293.0	49.0	34.0	0.220745	83 / 376
1	2.0	3622.0	51.0	0.014422	53 / 3.675
2	1.0	303.0	903.0	0.251864	304 / 1.207
3	296.0	3974.0	988.0	0.083682	440 / 5.258

	high	low	medium	Error	Rate
0	18.0	47.0	54.0	0.848739	101 / 119
1	14.0	1100.0	106.0	0.098361	120 / 1.220
2	21.0	266.0	101.0	0.739691	287 / 388
3	53.0	1413.0	261.0	0.294152	508 / 1.727

4.2. Extended Model

ModelMetricsMultinomial: xgboost

** Reported on train data. **

MSE: 0.08892165485306923

RMSE: 0.29819734212945165

LogLoss: 0.2978544109530065

Mean Per-Class Error: 0.14100806913270514

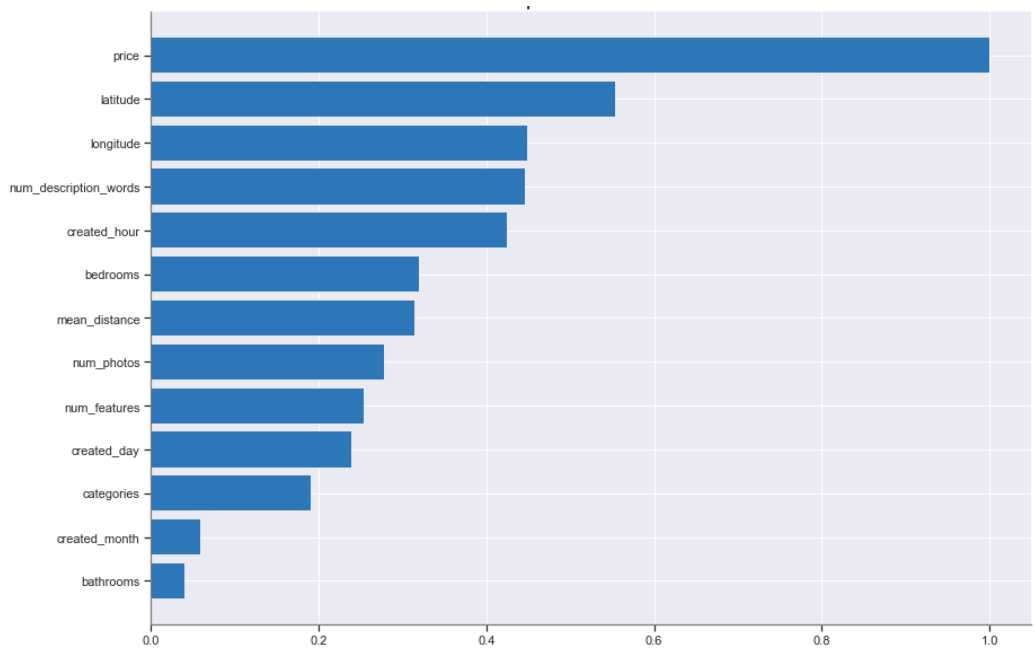
** Reported on validation data. **

MSE: 0.21528158906693382

RMSE: 0.46398447071743015

LogLoss: 0.6564431960940968

Mean Per-Class Error: 0.555266238294481



	high	low	medium	Error	Rate
0	345.0	40.0	19.0	0.146040	59 / 404
1	0.0	3886.0	53.0	0.013455	53 / 3.939
2	2.0	334.0	939.0	0.263529	336 / 1.275
3	347.0	4260.0	1011.0	0.079744	448 / 5.618

	high	low	medium	Error	Rate
0	13.0	39.0	39.0	0.857143	78 / 91
1	9.0	882.0	65.0	0.077406	74 / 956
2	14.0	220.0	86.0	0.731250	234 / 320
3	36.0	1141.0	190.0	0.282370	386 / 1.367

5. Discussion

In this section, we will discuss the obtained results from the modelling process. For the baseline model, we achieved a root mean squared error (RMSE) of 0.305 on the training set and 0.463 on the validation set. This results in an error rate for the classes of 8% on the training data and 29% on the validation data. For the extended model with 2 additional features the results are quite comparable with a RMSE of 0.298 on the training set and 0.463 on the validation set. The error rate for the training data is around 7% and 28% for the testing data. In total we could achieve a slight improvement by adding the additional features to our modelling process, but it was not as significant as expected. By locking at the plots for the variable importance of both new features, one can see that the importance of "mean_distance" lies within the top 7 features. The "category" feature was not so important after all, which was quite surprising.

6. Conclusion

The Purpose of this project was to identify new features for rental listings that can be used to predict the interest level of a customer. For this reason, two features have been proposed. Unfortunately, the importance for the model to predict the interest level of a customer was not significant and thus not really adding any new business value to the model. Nonetheless it was an interesting exercise and in a real-world case it is also quite helpful to reject a hypothesis and not invest in it. In addition, we have been quite limited by the free version of the foursquare api which only allows a specific quota for calls in a defined time period. For this reason, only a small subset of the dataset could be added to the competition data. It would be interesting to test more features with higher rate limits in the future.