Homework 3 **Zafar Ahmad, 111195037**
CSE 519 (Data Science Fundamentals) **Ubaid Ullah Hafeez, 111195374**
Submission Date: 10-01-2017

Part 1

Our Scoring function, f takes data about a property as input and returns a pair of non_geographical score and geographical score:

f(**data**) = (**non_geographical score**, **geographical score**)

Non-Geographical Score

Some Interesting Facts About Real-Estate :

Our Scoring function is based on some of the observations about houses being sold in last few years by some real-estate researchers which are as follows :

- Buyers generally prefer centrally air-conditioned homes and are willing to pay more
- Houses sold in last three years have 2-3 average number of bathrooms
- Houses sold in last three years have 3-4 average number of bedrooms
- Most of the Houses sold have a garage which has space for 2 cars
- In California, heat pumps are preferred heating system because of energy efficiency and providing necessary amount of heating when needed
- Most of the buyers prefer a single story or a double story house, not more than that. In addition to that people prefer to have an attic or basement for extra space and storage
- Other than bedrooms, people prefer to have some extra rooms like laundry room, a dine-in kitchen, some common sitting room etc and are willing to pay more for these
- Most preferred unit count is one
- People prefer to have a patio and storage room and are in most cases willing to pay more for these.

Some Deductions:

Following are some considerations based on our own sniff test after looking at the data:

- A newer house is more desirable to an old one, unless the old house is an antique one and is well maintained and well kept
- Areas where tax is high are those with good neighborhoods and have really good facilities for residents, so the houses with higher tax / sqft of area should be more desirable

Scoring Function

Keeping the above things in mind, we picked a subset of 15 columns from the given data and scored each of them individually such that score reflects the above mentioned points. General technique we used is to find the best in the each column and assign decreasing score from there to all the other values based on how far apart these are from the best. After assigning scores, normalize each of the column individually. Now, to find the accumulated score, we just sum the score of all the columns. Following is the list of columns that we used:

- **airconditioningtypeid**
- **bathroomcnt**
- **bedroomcnt**
- **buildingqualitytypeid**
- **garagetotalsqft**
- **heatingorsystemtypeid**
- **numberofstories**
- **propertylandusetypeid**
- **roomcnt**
- **unitcnt**
- **yardbuildingsqft17**
- **yardbuildingsqft26**
- **yearbuilt**
- **landtaxvaluedollarcnt**

How good is the scoring function

We think our scoring function is decent based on the two observations. First one being the difference in the scores of houses on the extreme values being higher as compared to

differences in the scores close to mean being lower. Second observation is that the distribution looks close to normal distribution which suggests that it scores houses in a way that we are used to see, which decent enough reasoning to say that this scoring function worked just fine.

Geographical Score

Finding Correct Zip-Codes

Zillow's zipcode are not the zip code but a mapping to it. They mapped each zip code to a table id. We used a data-set (https://gist.github.com/erichurst/7882666) that gives us the latitude and longitude of all Zipcodes of United States. From that datasets we filtered out the zip codes of our desired counties. Now we did clustering based on the Zillow zip-ids. We measures the distance of each points from the zipcode geographical coordinates and assigned the correct zip code ids based on majority voting technique. We also ran randomized experiments to see if our predicted zip code is correct. Now we have correct Zip code of each house. We found a dataset **Forbes 500 Most Expensive Zip Codes** (https://www.forbes.com/sites/samanthasharf/2016/12/08/full-list-americas-most-expensive-zip-codes-2016/), conducted by **Altos Research**. They considered the following factors:

- **Median Price**
- **Average Days on Market**
- **Inventory**
- **Previous year rank**
- **Different facilities in that Zip code etc**
- We found 192 most expensive zip in california. Among these 192 zipcode 64 of them was our desired zipcodes for the three counties. We did imputation based on the geographic distance to measure the relative expensiveness score for the rest of the zip codes in our dataset.
- Now we created a data-set that compared the price of the houses similar to the best house given by the above non geographical score across all the zipcodes and based on this, we assigned score to to each of the zipcodes such that the zipcode where a similar house is cheaper is given a lower score as compared to the expensive one.

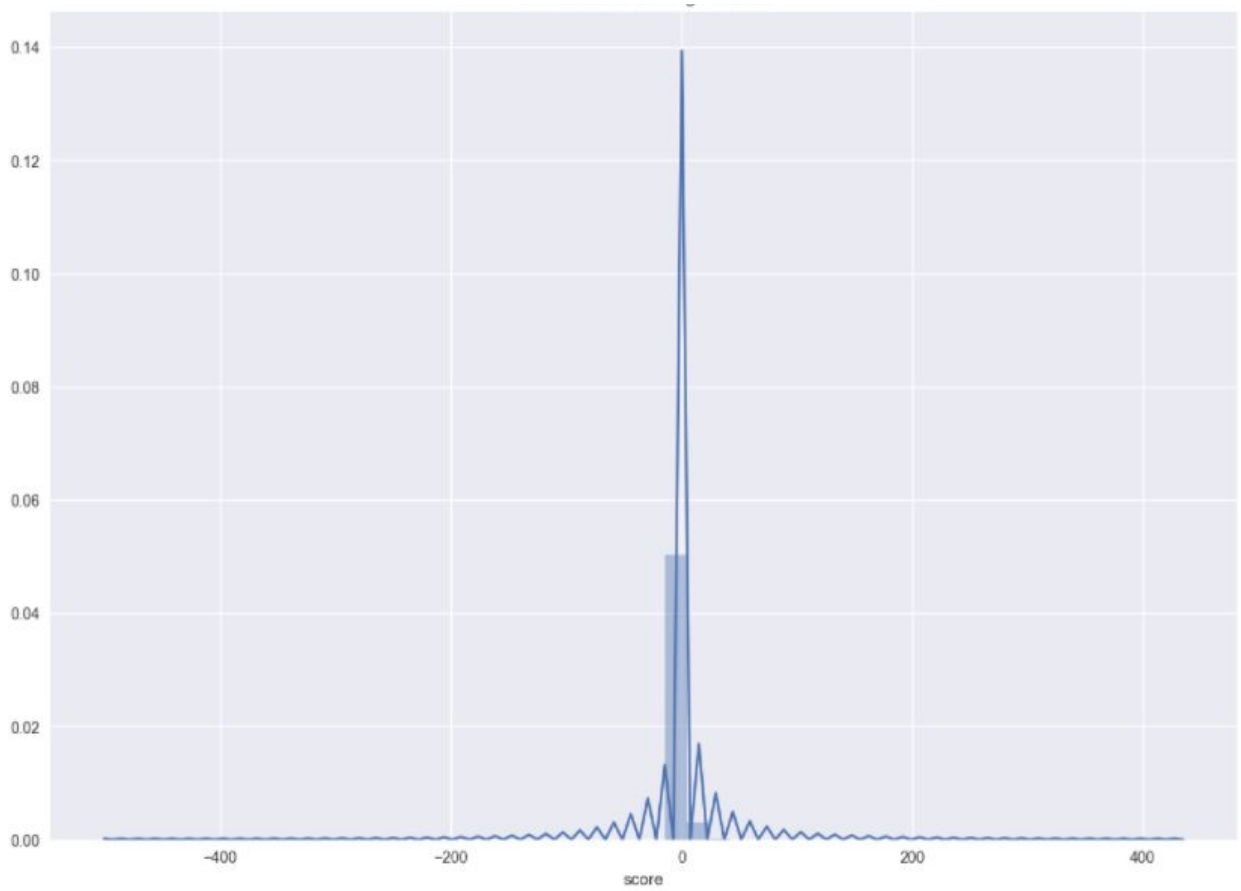

Figure: Distribution of Scoring Function

Part 2

Our scoring function returns a pair i.e. (Non-Geo_Score, Geo_Score). We represented each property with this ordered pair and our distance metric is just the euclidean distance of these ordered pairs of each pair of properties. We will discuss about the goodness of this metric in **Part 3** when we discuss clustering.

Part 3

How Many Clusters?

There is an algorithm for clustering called DBSCAN which don't take the number of clusters as an input parameter, instead it takes another parameter called eps which is related to the distances among the points that we want to cluster. We read upon this and found that if we find the distance of each point to its 3rd neighbor and plot it. The point where we find the 'knee' in the plot is a suitable value for eps for DBSCAN.
So, 'knee' of the said graph appears at about 0.6 so we used this as eps. After that, we ran

DBSCAN with eps=0.6 and DBSCAN clustered the points into two clusters. This way, we found out that classifying the above data into 2 clusters is a good idea.

Clustering

DBSCAN don't cluster outliers and researchers say it's not good at prediction. So, we clustered the data using KMeans into two clusters

Comments About the Distance Metric

- A good distance metric should have just one 'knee' in the nearest neighbors plot and it should be towards the end (it is clear in the plot below).
- Another comment about the distance metric is that it should be easy to calculate which ours is.
- When you cluster using a distance metric, if you see clear clusters and few outliers. This gives us the sense that the metric we used works well with the clustering technique. In our case, clusters are not very clear, but outliers are very few. So, our distance metric is not very good, but it's not bad either.

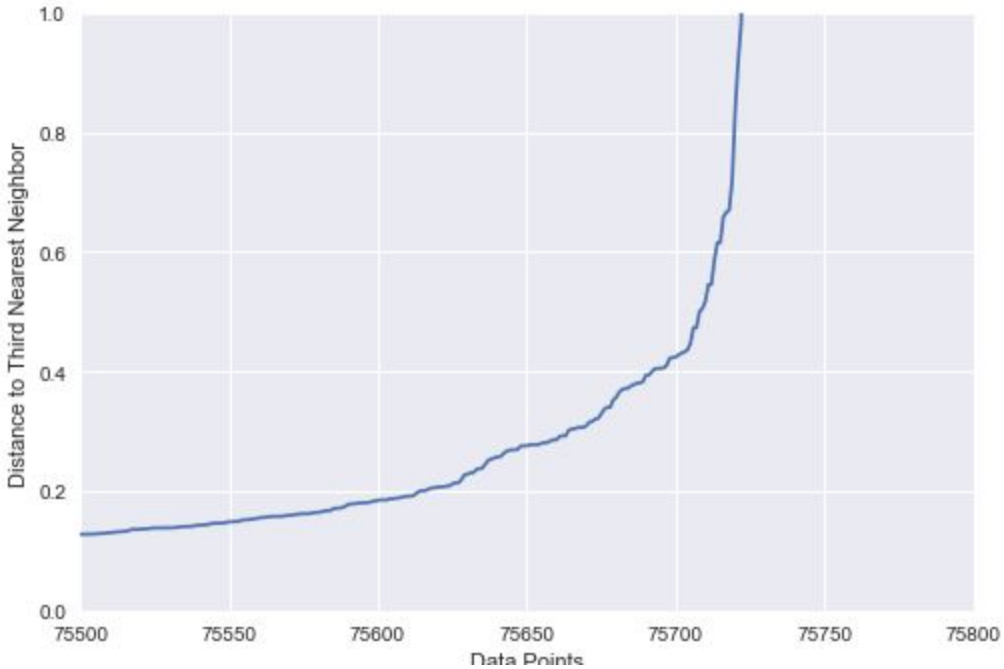

Figure: 'knee' in Nearest Neighbor Plot

Looking at the graph, you can see that 'knee' occurs ~0.6, so we use eps=0.6 for DBSCAN to find number of clusters. Estimated number of clusters using DBSCAN is 2.

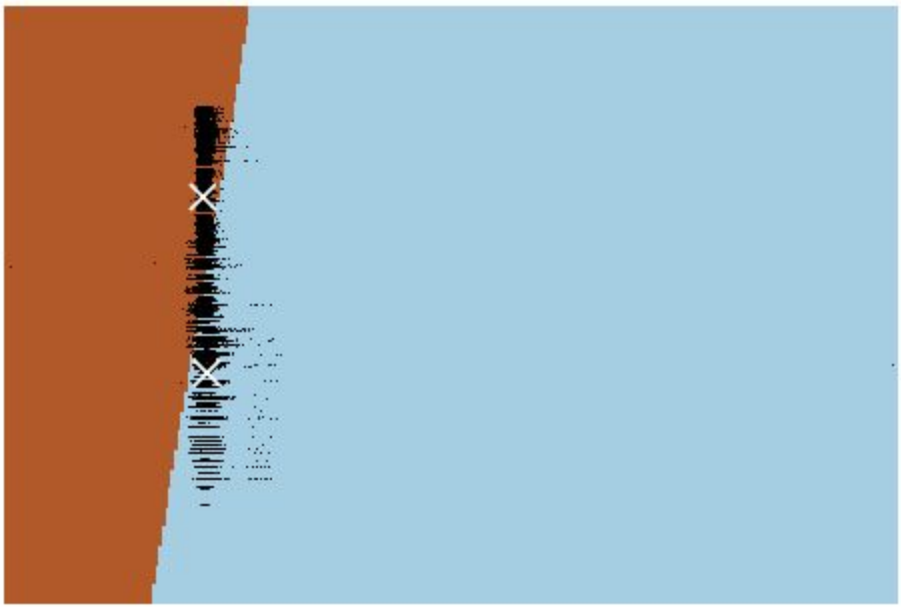

Figure: K-means clustering (Centroids are marked with white cross)

Part 4

Since we have used the geographical score (generated using **Altos Research** dataset) in finding the most/least desirable house in previous parts, we explained the external dataset (**Altos Research** dataset) in **Part 2** in *"Geographical Score"* section. We have also showed the impact of external dataset in **Part 5** that our best model is using Geographical score clustering which gives us a very nice improvement.

Part 5

We tried a bunch of things which are as follows

- Clustering based on Desirability and Regression on Features
- Predicting logerror just using the desirability
- Regression on Desirability
- Regression Using some Selected Features
- XGBoost on Features
- Taking care of Outliers while training
- Different combination of features
- Polynomial feature engineering
- Polynomial Regression and Boosting
- Clustering based on Desirability and regression using XGBoost on selected features of properties already given to us (This worked best)

Best Model

Our best model was based on Clustering and using multiple predictors based on XGBoost. What we did is as follows:

- Created clusters using desirability scores of training data using KMeans clustering and to decide on the parameters, we used the same technique as explained in part 3
- For each cluster train a predictor based on XGBoost using all the points that lie in that cluster. This predictor was based on all the actual features of the data not just desirability scores
- Predict the clusters of all data in the sample_submission file on Zillow using KMeans predictor
- For each data point, use the predictor of the predicted cluster to predict the log_error

Why this Model is Best?

We think this model works best because our desirability score is based mostly on the features provided already in the data-set and clustering based on desirability, clusters the similar houses together. So, if we use the desirability of a house to find the houses that are similar and then use the regressor that is trained only based on similar houses is more likely to predict a better log-error as compared to a regressor that is trained using all the houses. The reason behind this is that the predictor used by zillow is most likely dealing the houses with similar desirability in a similar way.

Interesting Findings:

In our first assignment we ignored the geographical features. It is said that, three most important feature of real estate business are Location, Location and Location. After analyzing the geographical features and creating scores using zip code clustering we see that certain zip codes desirability changes over time. And this change has some good effect in the Zestimate/Logerr as well. We incorporated these values and make cluster. Instead of a global prediction clustered prediction gives us a better prediction. We have a strong feeling that if we refine our desirability score a bit more and get more fine grain clustering and spend some more time exploring the best parameters for the tools we used in this assignment, it's likely that we can get a score in top 10 or at least top 100, but I think that's out of the scope of this assignment considering the time allotted.

Yes Professor, You were Right!

If we don't use clustering based on desirability and use only one XGBoost regressor to predict logerrors for all the properties accuracy of prediction is less, which means since clustering based on desirability makes us more familiar with the data. It in a sense, enables us to guess logerrors for rest of data in a more educated manner.

Part 6

In this part, we subsampled our data. Because running permutation test for the whole data is very time consuming. We randomly selected 10000 point from our point distribution for the training set. We have randomly selected another 10000 non-overlapping points as our test set. In the first iteration, we do not change the order of the ylabels and record our MAE/MSE. In all the next iteration we randomly reorder the ylabels. Since clustering is a part of our hypothesis, we scramble the non-geographic/geographic scores as well.

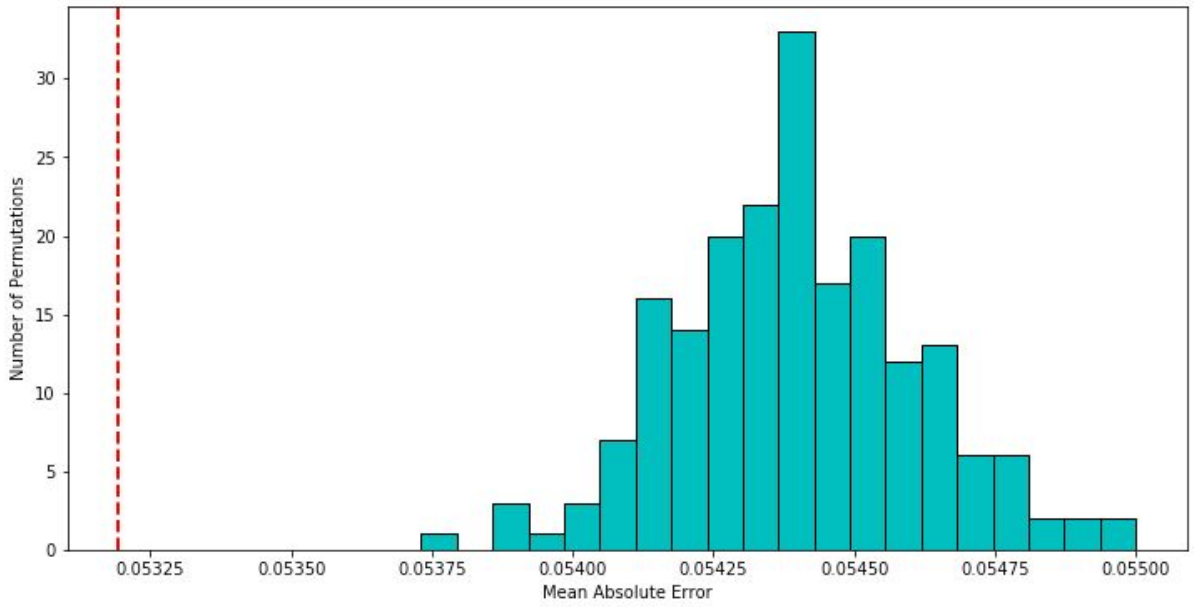

Figure: Permutation test histogram

Explanation

We ran our experiment for 200 permutations of training data. From the above figure it looks like our **P-value** is **Zero**. However, we are generating permutations for 10000 points and number of possible permutations of 10000 points is **10000!**, which is impossible or not feasible to generate. If we could have generated all these permutations we would have seen some permutation's MAE could be less

than ours. From the above figure we can infer that even if the P-value does not become Zero in long run but it will remain close to zero.

Part 7

We have submitted our prediction in Kaggle. Since we were unable to create team in Kaggle and talked to both TA and Professor about this, we used **Uhafeez** account to submit our prediction. Result are given below:

MAE: 0.0648731

Rank: 2425