**CSE 519 - DATA SCIENCE FUNDAMENTALS**

**Report - Homework 3**

Team members:

1. Jitendra Savanur (111491676, [jitendra.savanur@stonybrook.edu](mailto:jitendra.savanur@stonybrook.edu))

2. Abhinav Jain (111495982, [abhinav.jain@stonybrook.edu](mailto:abhinav.jain@stonybrook.edu))

3. Chaitanya Kalantri (111446728, [chaitanya.kalantri@stonybrook.edu](mailto:chaitanya.kalantri@stonybrook.edu))

**Desirability Function**

This function returns the desirability score for a given property instance in the data set.We have assigned a weight to each of the below features, which decides the impact that feature will have on the final desirability score.

We have considered the following features to compute the desirability of a property:

|  |  |
| --- | --- |
| **Feature** | **Weight** |
| Building quality | 10 |
| Bathroom count | 14 |
| Finished Square Feet | 12 |
| Garage car count | 10 |
| Swimming pool count | 10 |
| Room count | 13 |
| Total tax assessed for the property (sum of land tax and structure tax) | 8 |
| Total property tax for an assessment year | 8 |
| Number of units the house is built on (duplex, triplex etc) | 12 |
| Year built | 3 |

For each property instance, the final desirability score is calculated as follows:

* Divide the feature value for that property with the maximum value for that feature in the data
* Multiply the above number with the weight
* Add it to the total desirability

Finally, the sum of all the above such terms will give us the desirability score for that property instance.

After sorting the data based on desirability, we get the properties sorted in the order of their desirability.

Observations:

* There are some property instances whose feature values look like outliers.
  + For example, the four most desirable houses in the data have a unit count of more than 400. Perhaps, these are actually building/office spaces on sale.
  + Another example is that ninth most desirable house in the data has a finished square feet value of 3273 but has 96 rooms in all.
* Apart from the above outliers, the desirability function did well in giving a good score to houses which were built recently, had some good number of bathrooms and rooms, with a decent building quality etc.

**Clustering using Distance Function:**

The pairwise distance (which computes similarity between two houses) function took into account the following features, with some scaling involved for each of them:

1. Bathroom count
2. Room count
3. Building Quality
4. Year in which it was build
5. Swimming pool count
6. Units upon which the house is built
7. Garage car count

The following attributes had too big values, therefore had to be scaled down. The method used to scale them down and transform them into manageable values was through by taking a logarithm of the values:

1. Finished Square Feet
2. Total tax assessed for the property
3. Total property tax for the assessment year

Apart from the above features, the longitude and latitude of the houses was taken into consideration for the distance.

After scaling the appropriate feature values, we computed the difference between each of the features and summed up the absolute values of the differences between them to come up with the final distance value.

**Clustering:**

We used KMeans clustering algorithm to cluster our data. We clustered our data into 10 clusters and plotted the results on a scatter plot.

The limitation with KMeans clustering in sklearn is that it uses Euclidean distance as its default distance measure. And sklearn does not allow us to provide our own distance function to its KMeans clustering implementation.

We found another clustering mechanism provided by scipy, called fcluster, which is a type of hierarchical clustering. This clustering method allows us to have our own distance function for the clustering algorithm.

Since this clustering method calls our distance function for every pair of property instances, it is very computationally heavy to run it on 3 million records. Therefore, we tried it for 10000 records and plotted the results on scatter plot

**External Dataset:**

In the original dataset provided by Kaggle, the zip codes were masked. Hence, we made use of a Python library called uszipcode in order to find data based on the longitude and latitude of the property instances. We extracted the following features for the longitude and latitude from the above library:

1. Population

2. Population density

3. Average Wage

4. Zip code

We believe the above features have helped us build a better predictive model

**Discussion:**

This new dataset with geographical data will be useful because:

1. The dataset which we selected is based on the geographic location. Considering the zip codes with several other relevant fields on that zip code.
2. If we know the zip code, we can estimate the house value prices. Because each area has different tax value, and land value price, based on the locality.
3. Moreover, knowing the population of a particular area, we can understand whether the area has rich demand for house purchase or not. More the population, more will be the need for houses. Hence, the prices of the houses can increase if the demand is more than the supply.
4. The salary range of a locality will help to understand the overall wealth of the individuals in the locality. For instance, if the public of a specific locality are rich, they may add a lot of cost in maintenance, security and the interior of the house. So, for a house with a given value for the finished square feet, the price of the house in this locality will be greater.
5. The supply and demand of the houses can also be measured by the density of the population in that area. Hence, even if a house is small in area, but there is no space for the building of new houses, in such situation, the same small house will earn more, than a big house with less demand.

**Analysis:**

1. This dataset was surely useful, because we were able to understand a better picture of the location.
2. Hence, now we were able to distinguish the prices of multiple houses with the same square feet area, pool count, bed room count, and room count; just because of the popularity of the area.
3. For instance, a house in New York may cost more than a house of same features in North Carolina.

**Best Prediction Model:**

For the prediction of the model as per the competition, we implemented two models:

1. Linear Regression
2. Lasso Regression

**Linear Regression:**

Based on the experience of homework 2, linear regression was giving the best predictions. Hence, we went with the same model. However, this time the dataset in operation is a much more cleaned and merged with external dataset. Hence, we were expecting a good rank.

Moreover, we tried to consider the best parameters which will impact the house value prediction. Such as Tax value dollar count, zip codes, building quality type, transaction data and few more. Linear regression was implemented using sklearn.linear\_regression in Python.

However, because there were few records in the field of longitude and latitude which were null; we were not able to compute the model on the raw data. Then, again we had to replace the null values with the median of the column. As there were around 3000 records with null values, it was not possible to identify the exact longitude and latitude values for each of them.

To our surprise the model didn’t give the expected results, the reason being replacement of the null values with the approximate median values. The model was ranked 2672, among 3800+ participants.

**Evaluation of how it works:**

1. In more technical way, we are able to predict whether the quantities on x-axis and y-axis are positively or negatively correlated
2. Statistically, more the value of R-square, and it's respective Adjusted R-square; better is the correlation between these variables
3. If the slope of the plot is positive. Then, we can say that the variables are positively correlated. And vice-versa.

**Lasso Regression:**

In order to improve the accuracy of the model, we implemented the Lasso model on the same dataset. The Lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces (Wiki definition).

Lasso regression was implemented using sklearn.linear\_regression\_lasso. The model was able to better predict the results, but not with a significant count. The rank improved by 39, and helped to reach 2639 rank.

**Evaluation of how it works: (**Source: Wikipedia**)**

1. It is a regression method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces
2. Lasso’s ability to perform subset selection relies on the form of the constraint and has a variety of interpretations including in terms of geometry, Bayesian statistics, and convex analysis
3. Lasso regularization can be extended to a wide variety of objective functions such as those for generalized linear models, generalized estimating equations, proportional hazard models, and M-estimators.
4. In addition to fitting the parameters, choosing the regularization parameter is also a fundamental part of using lasso. Selecting it well is essential to the performance of lasso since it controls the strength of shrinkage and variable selection, which, in moderation can improve both prediction and interpretability

**Surprises or Interesting Experiences:**

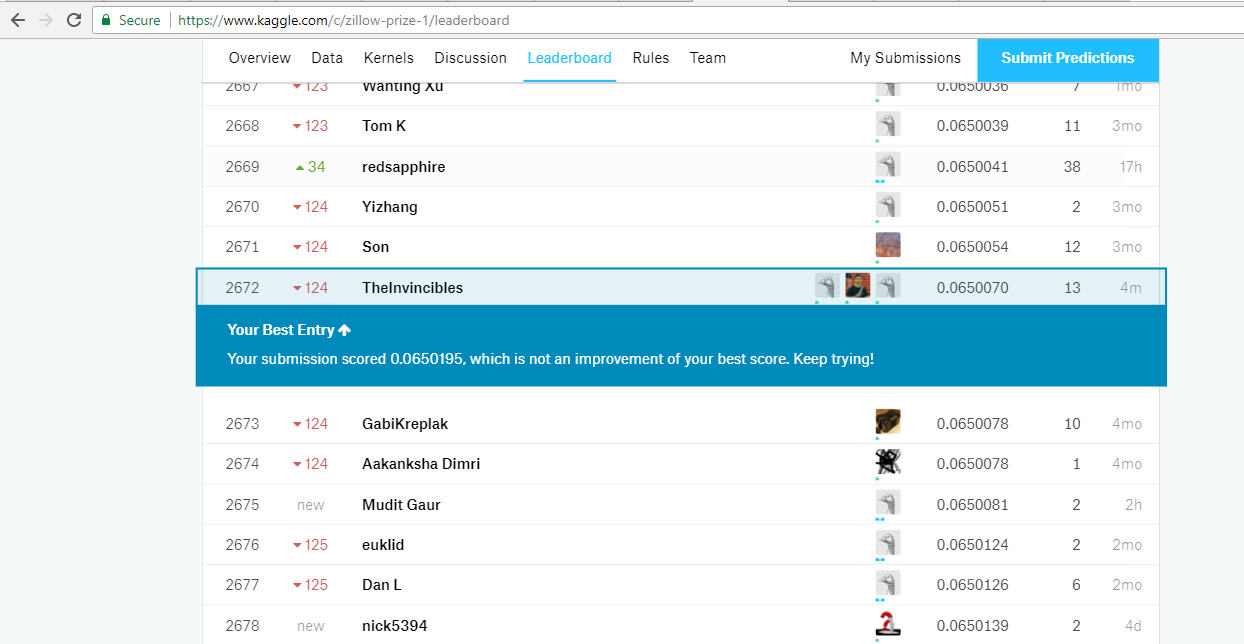
1. When we were calculating the p-value, most of the times the p-value is coming as 0. Which means the data is over fitting. However, when we checked the difference between the predicted values and the original values. The difference is very less (around 0.001).

However, as we were shuffling the data multiple times, we were able to get the p-value till 0.02.

1. Another interesting experience was during merging of data. There were few incorrect zip code values. Hence, we tried figuring the zip code of several records based on the longitude and latitude of the record. Even though it was labor work, still it helped us complete the cleaning quickly as compared to merging the incorrect values through some external dataset.

**Screenshots for Kaggle ranks:**

1. Using Linear Regression



2. Using Lasso regression:

