* **Project Title:-**

Housing Trends Prediction

* **Group Information:-**

**Member 1** : Saad Azhar

**Member 2 :** Pranav Mahesh Makhijani

* **Introduction and Problem Formulation:-**

**Problem:** Prediction of housing trends and prices in a certain zip codes.

The purpose of project is to provide customers with a system which would enable them to know about the prices of the houses in future. It will enable them to make calculated decisions to invest on a particular house. Moreover another purpose of this project is to provide customers with houses which have similar characteristics while buying or renting. This feature would allow them to have multiple options to choose from. For course project we are dealing more specifically with house price prediction.

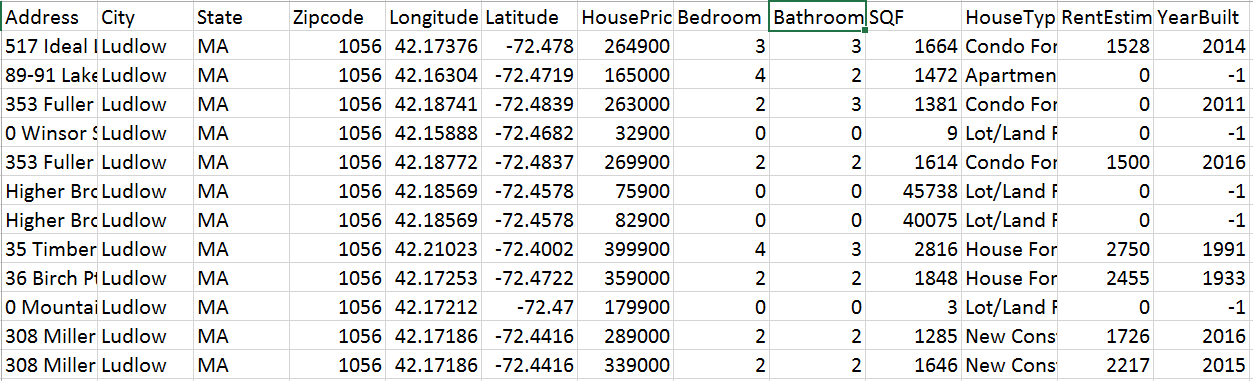
* **Dataset Description:-**

We had 2 separate Zillow house datasets which we used for price prediction. Both the datasets were from Massachusetts but they were having different cities and zip codes. In these dataset we had a total of 13 attributes. Out of which some were numerical and some were textual attributes. These final 13 attributes are as follows:-

1. ID
2. Address
3. Zip Code
4. City
5. Bedrooms
6. Bathrooms
7. Estimated Price
8. Area Space
9. Year Built
10. Estimated Rent
11. Mortgage
12. Price Per Square Feet
13. Status of House(Condo/Town House/ Full house)

**Pre-Processing and Partitioning of data set:**

In our datasets there were a number of records which had Rent Estimate to be zero. To fill up those numerical missing values we used the median technique to impute them. We confirmed this in the Data Exploration phase. We cannot simply impute these values with mean or median. It is always good to leverage the existing knowledge from the data to impute missing values. Following image explains the issue.

****

We imputed missing values based on the house price. For example - if the house price falls in the $100,000-$200,000 bracket, the median rent estimate is $1421. We calculated the median rent estimate based on its price bracket. We calculated this median based on price bracket because whenever the rent was increasing then house price was increasing too in our data set but there could be a scenario when rent estimate is not directly proportional to price. In that particular case we will be needing to come up with a more efficient technique. Following is a code snippet of how we achieved this:

****

We partitioned the data into training and validation set with a ratio of 80% training data and 20% validation data. I had no test data set for this project because it was a case of unsupervised learning.

* **Project Milestones:-**

1. **Metadata Extraction and Imputation:**
   * 1. **Extracting metadata:**

We were provided with a data sets for which there was no need to extract metadata as it was already preprocessed till metadata extraction point. We started with the imputation step in this project.

* + 1. **Dealing with missing data:**

There were a lot of numerical missing values in the data set. For that we used median value imputations. After that we used python’s standard scalar normalization to normalize the data. This enabled us to further improve model’s accuracy. We used Z-score normalization too but we achieved better accuracy using standard scalar Normalization.

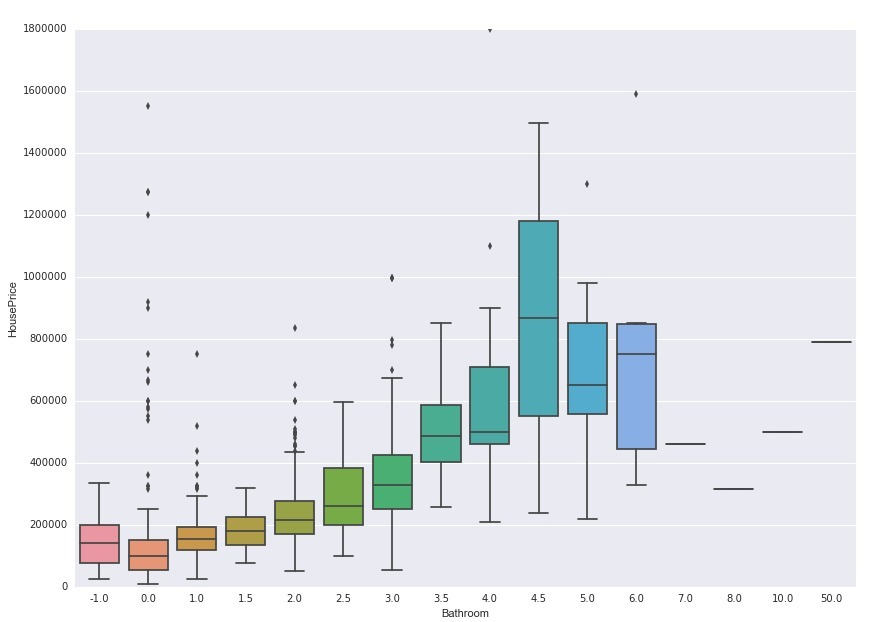
* + 1. **Dealing with outliers:**

In case of outliers we straight away removed those entries as it caused very absurd and variable end results. Moreover outliers have a big effect on accuracy too. So we decided altogether to get rid of them.

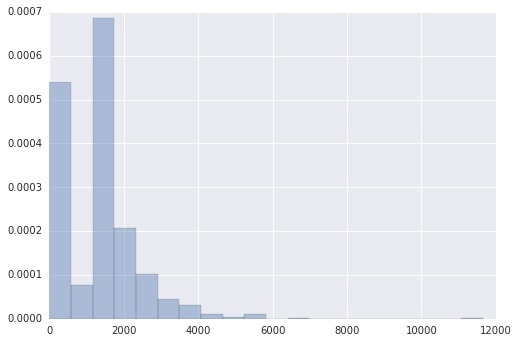
* **Summarization and Visualization:-**

1. **Data Exploration:**

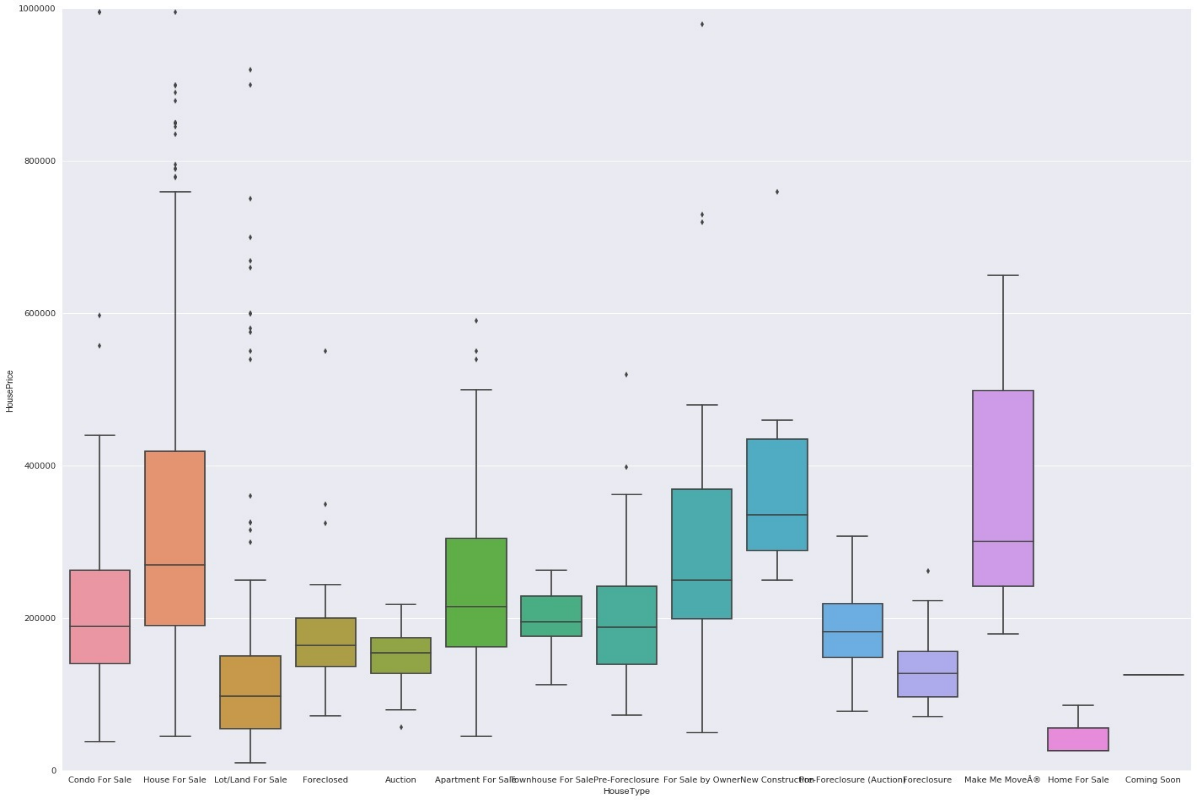
We started this project with certain assumptions about the dataset which we wanted to validate. The best way to do that is through Data Exploration.

**Bathrooms Vs House Price: -**

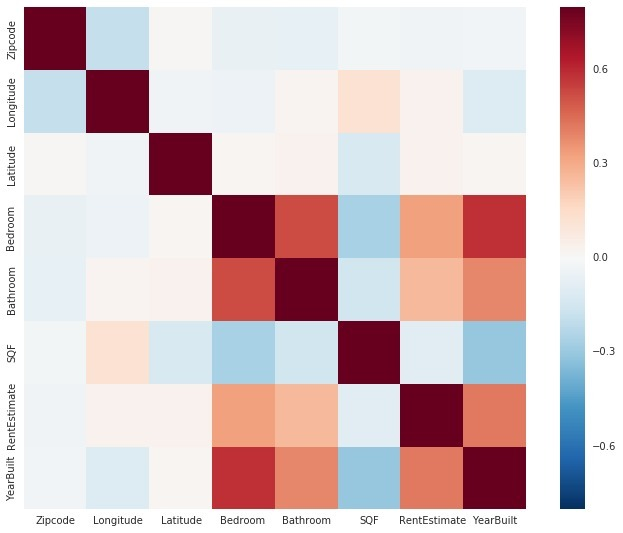
Here we can see that as the number of bathrooms increase, the house price increases which makes intuitive sense. We can see some outliers too which explains sparsity of data.

**Rent Estimate Plot: -**

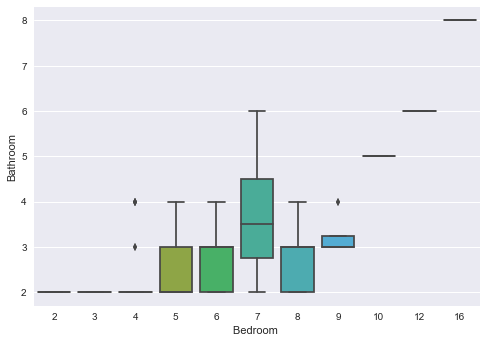
The rent estimate has a lot of missing values. This bar plot shows that majority of the values are zero. We intend to use feature engineering techniques to impute missing ‘rent estimate’ values.

**House Type Vs House Price:-**

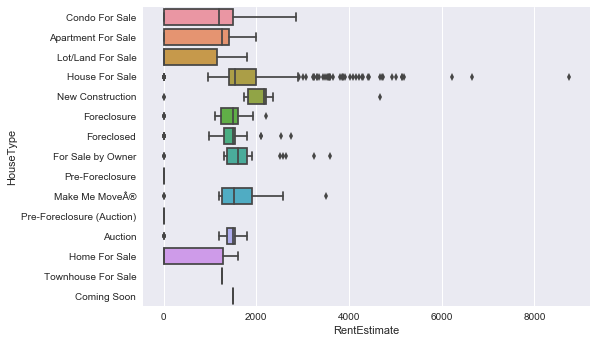
We plotted this box plots to understand the distribution of house prices according to house types. In addition to the distribution of house prices, this also helps us detect outliers and gives us statistical metrics like mean, median, 25th percentile, 75th percentile, etc.

**Co-relation Heat Map: -**

The purpose of this heat map is to identify the correlation between all variables of our dataset. Bedroom is positively correlated to Year Built which makes intuitive sense. Other than that, there is no correlation which stands out.

**Bedroom Vs Bathrooms: -**

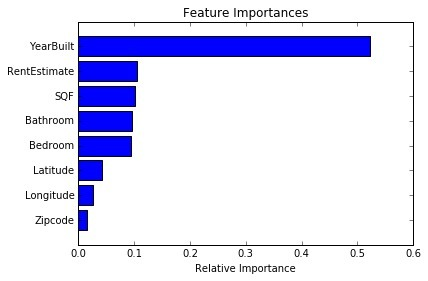
This image shows as the number of bedrooms increase, the bathrooms also increase.

**House Type Vs Rent Estimate Challenges: -**

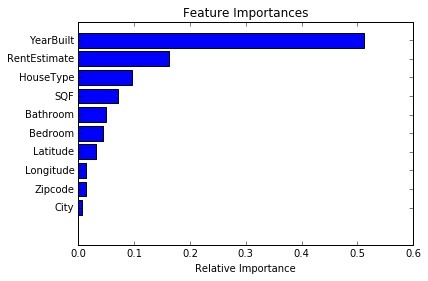
There are a number of outliers in the House for sale section. Since the number is very high, we cannot simply remove them. We have to come up with an effective technique to impute the missing values.

1. **Statistical summary:**
2. **Feature Selection:**

For feature selection we used Random Forest Algorithm. That algorithm gave us the importance of each feature in two separate datasets. Ultimately we went with all the important feature in both data sets as shown in the following figures. Feature with importance closer to 1 is the most important feature.

**Saad’s Dataset:**

**Pranav’s Dataset:**

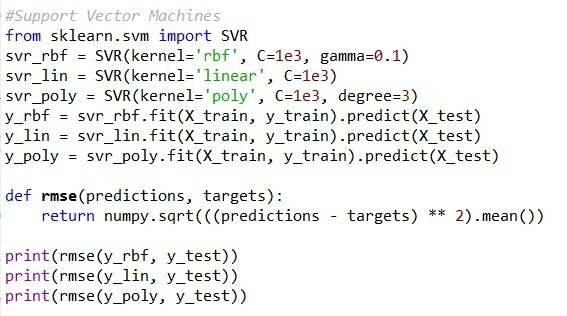
****

1. **Algorithms for prediction**

Algorithms which we used for prediction were as follows:

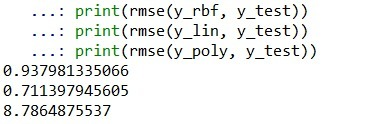
* Support Vector Machine
* Artificial Neural Networks

1. **Results: -**

**Support Vector Machine:**

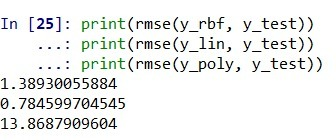
For SVM we used 3 different types of kernels linear kernel, Polynomial kernel and RBF kernel. Linear Kernel is used for linear hyperplanes, Polynomial kernel is used for fitted hyperplanes and RBF kernel is used for transformed hyperplanes. Apart from that we used 4 different cost functions which were C: 0.1, 1, 100, 500, 1000. The best accuracy we achieved was with C:1000. The following results are with 1000 cost function.

**Saad’s Dataset:**

****

The least RMSE we got was 0.711 and that was with linear kernel with 1000 cost function value. Which tells us that linear kernel performs the best in this data set.

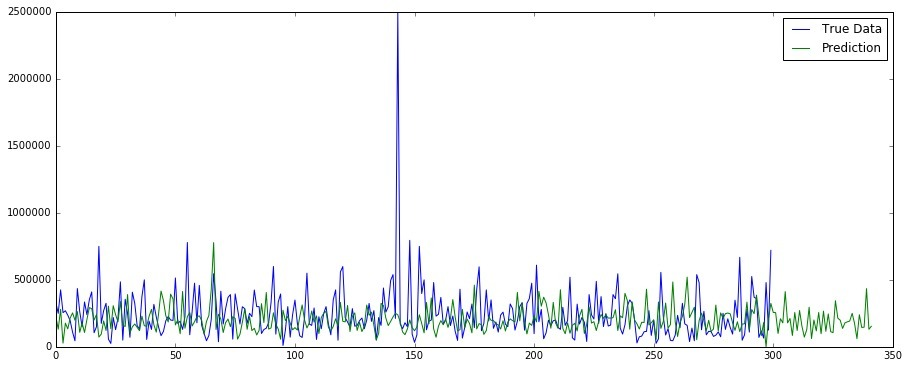
**Pranav’s Dataset:**

****

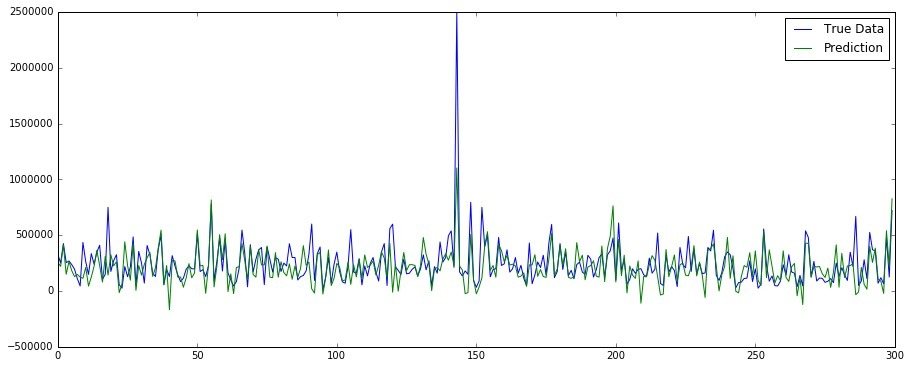
The least RMSE we got was 0.784 and that was with linear kernel with 1000 cost function value. Which tells us that linear kernel performs the best in this data set.

The less RMSE value in saad’s data set tells us that SVM perform better in that dataset.

**SVM Predicted Vs Actual Visualization:**

**Saad’s Dataset:**

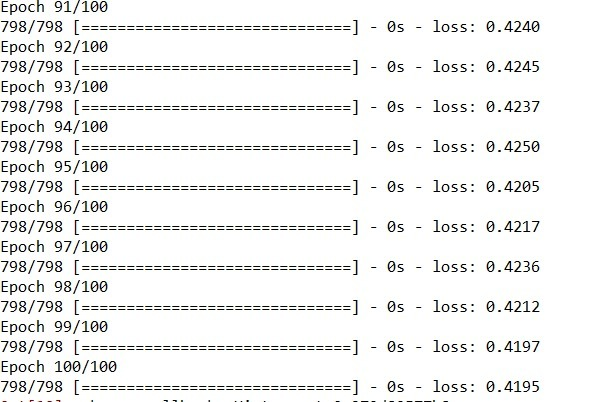
Here the blue line represents the original data and green line represents the predicted data. This plot shows a very respectable prediction trend. Which tells us that results are very reliable. We can spot one outlier here at around 2500000 price. The extra green line represents a longer prediction duration compared to the test set.

**Pranav’s Dataset:**

Here the blue line represents the original data and green line represents the predicted data. This plot shows a very respectable prediction trend. Which tells us that results are very reliable. We can spot one outlier here at around 2500000 price.

**Artificial Neural Network Results:**

For this algorithm we used loss function as Root Mean Squared Error and we used Adam Optimizer with Neurons equal to the number of features which we considered 8. We used 100 Epochs here. Epochs are the number of time the algorithm back propagates to reduce the error/cost function.

**Saad’s Data Set Loss Function:**

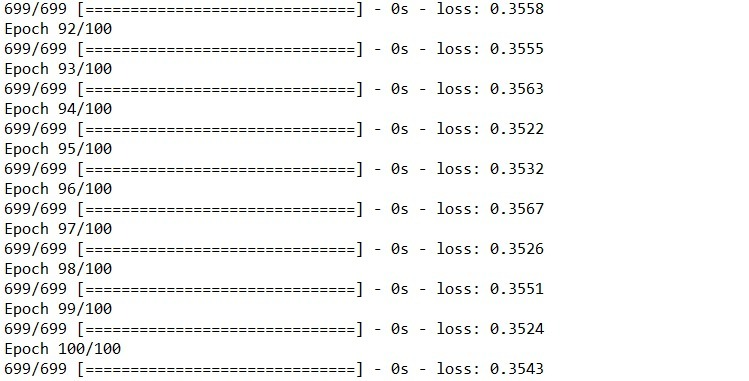
This above image depicts that with every epoch training loss function is decreasing. The last loss function value is 0.419

**Mean Squared Error:**

MSE value in Saad’s data set is 0.51

**Root Mean Squared Error:**

RMSE value in Saad’s data set is 0.644. Which is pretty less and it means algorithm is producing some very good results.

**Pranav’s Data Set Loss Function:**

This above image depicts that with every epoch training loss function is decreasing. The last loss function value is 0.35.

**Mean Squared Error:**

MSE value in Pranav’s data set is 0.63.

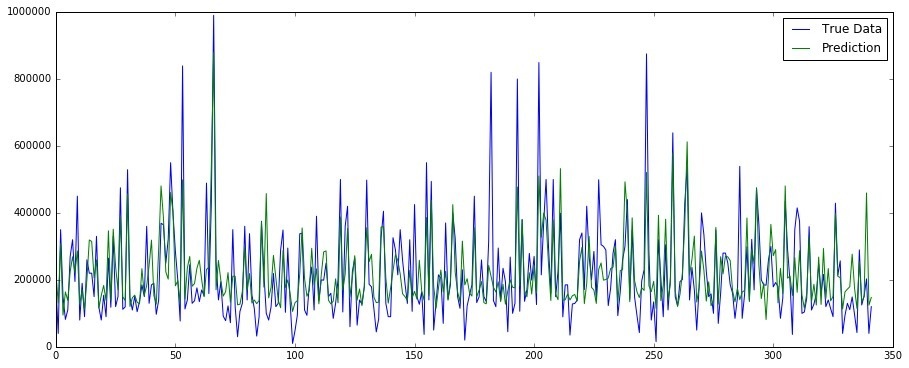
**Root Mean Squared Error:**

RMSE value in Pranav’s data set is 0.646. Which is pretty less and it means algorithm is producing some very good results.

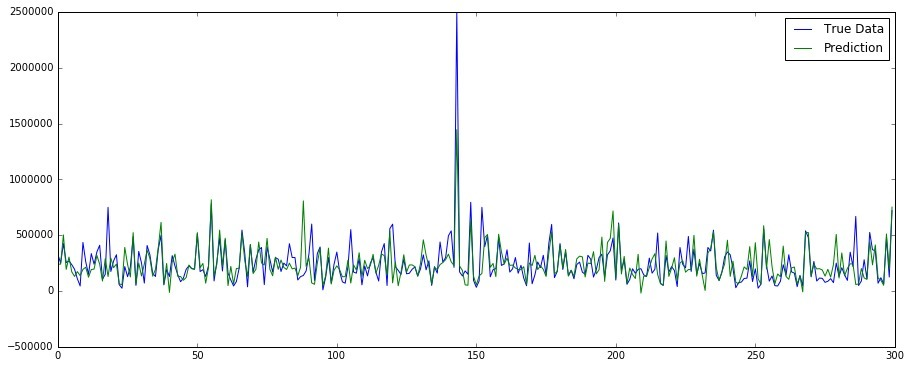
The less RMSE value in saad’s data set tells us that ANN perform better in that dataset. As RMSE for this algorithm and Saad’s data set is the least one so this means this ANN is producing the best result in this dataset compared to every other used algorithm.

**ANN Predicted Vs Actual Visualization:**

**Saad’s Dataset:**

****

Here the blue line represents the original data and green line represents the predicted data. This plot shows a very respectable prediction trend. Which tells us that results are very reliable.

**Pranav’s Dataset:**

Here the blue line represents the original data and green line represents the predicted data. This plot shows a very respectable prediction trend. Which tells us that results are very reliable.

1. **Challenges: -**
2. Dealing with inconsistency of data values.
3. Treating Outliers.
4. Dealing with interpretation of Neural Network Results.
5. Understanding and comparing the results from two different datasets.
6. **Future Work: -**
7. Understanding more about interpretation of Neural Network Results.
8. Extracting some constructive information using comparison between the results of both data sets.

* **References:-**

**Tools Used :** Excel, Python, Tableau, R-Studio

**Programming Language :** Python, R programming Language

**Links : https://www.zillow.com/**