Predictive Analysis of Housing Price in Washington D.C.

-Devarshi Tharwala

Table of Contents

Introduction 3
Data Exploration 3
Methodology 5
Analysis 6
Results 9
Conclusion 10

Introduction

Washington, D.C. is the capital of the United States. Washington's population is approaching 700,000 people and has been growing since 2000 following a half-century of population decline. The city is highly segregated and features a high cost of living. This dataset provides insight on the housing stock of the district. The average price of house in the Washington D.C. was $860,629 in the year 2018. The housing price shows increasing pattern since 1986 and it is critical to analyze the main variables causing this rise. This report takes deep dive into understanding the dataset and tries to predict the price of housing in the future.

Data Exploration

The dataset contains 46 columns and 28,900 records. Out of which 19 variables are categorical and 23 are numeric. There are also 4 variables which contains textual data. Figure 1 shows the average price of the house by different zip code in the Washington D.C. Also, the price of the house in the area is increasing year by year as it can be observed from figure 2, which shows line graph of housing price by year with respect to different quadrants of the city.

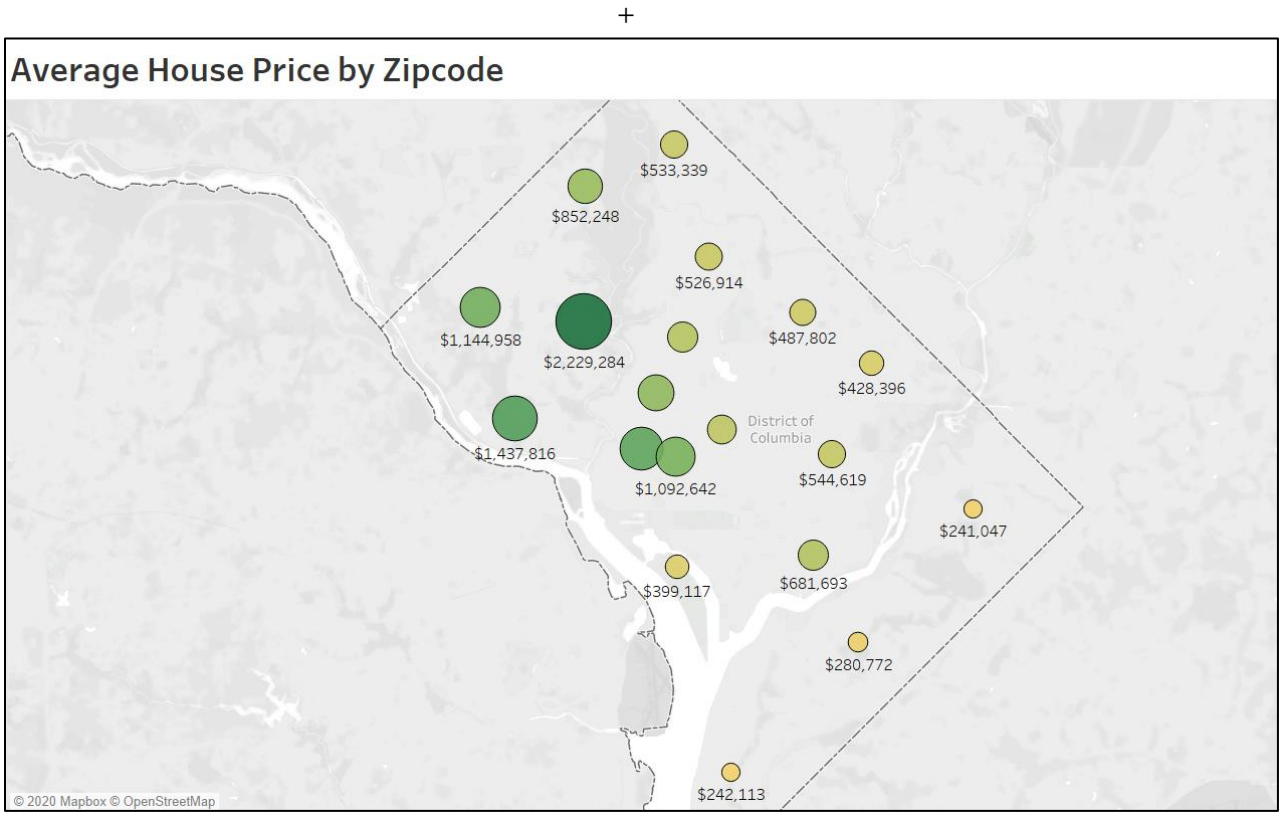

Figure 1: Average House Price by Zipcode

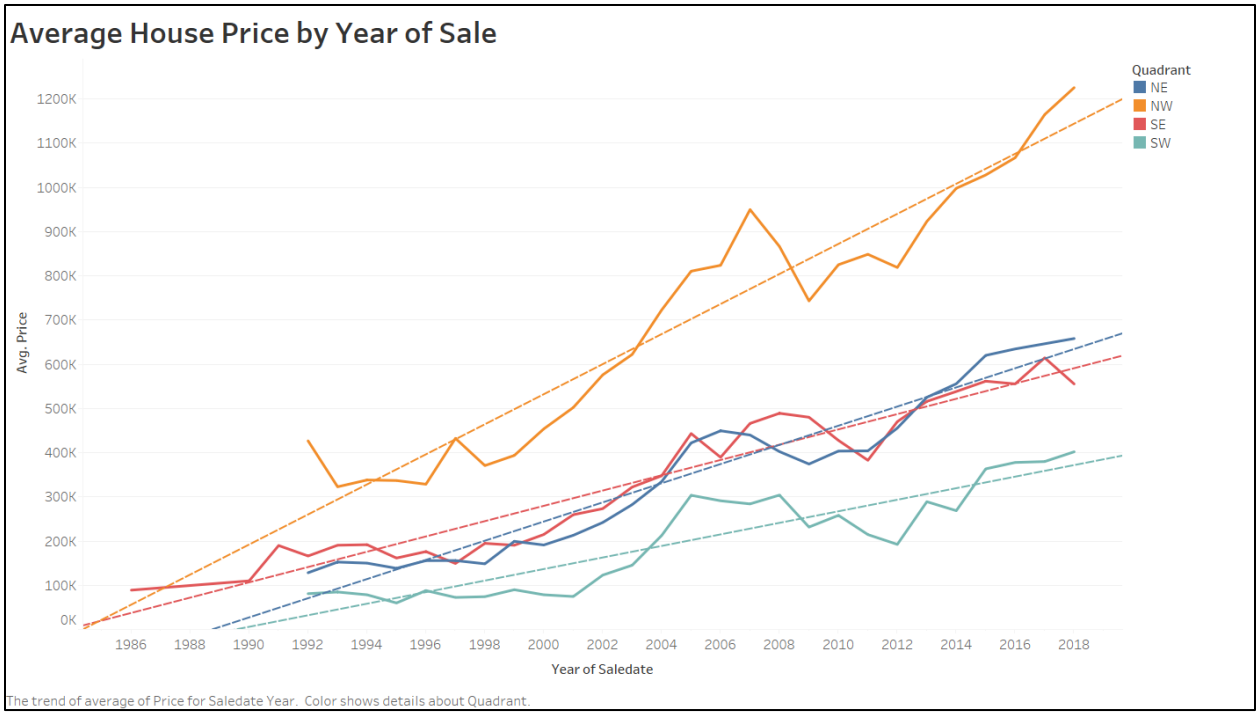

Figure 2: Average House Price by Year of Sale

Moreover, it was interesting to observer that 17% of the house that was sold was not qualified as it is shown in figure 3.

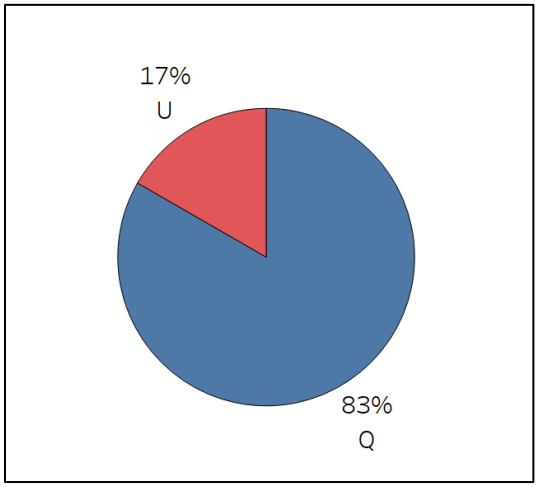

Figure 3: pie chart of the variable "QUALIFIED"

During the preliminary analysis on the data it was observes that in the "HEAT" column there are 5 records which has data entered as "No Data". In the "YR_RMDL" column which shows the year when structure was remodeled there was 1 record with data "20". In the "STORIES" data which shows how many stories building it is there are 4 records with data like "826, 250, and 275". These

data are dropped from the dataset because it will affect the analysis later in the process. Additionally, in the “AC” column there were 6 records with “0” as the data and it was imputed as “N” to match all the other data in the column.

There were 9 columns in the dataset which were either fully textual, redundant or same entirely throughout the column and was decided to drop in the very beginning. These columns are as follows: GIS_LAST_MOD_DTTM, SPURCE, FULLADDRESS, CITY, STATE, NATIONALGRID, X, Y, CENSUS_BLOCK

Methodology

The methodology followed to perform the predictive analysis can be easily understood by looking at the figure 4 which shows the flow chart of the whole process.

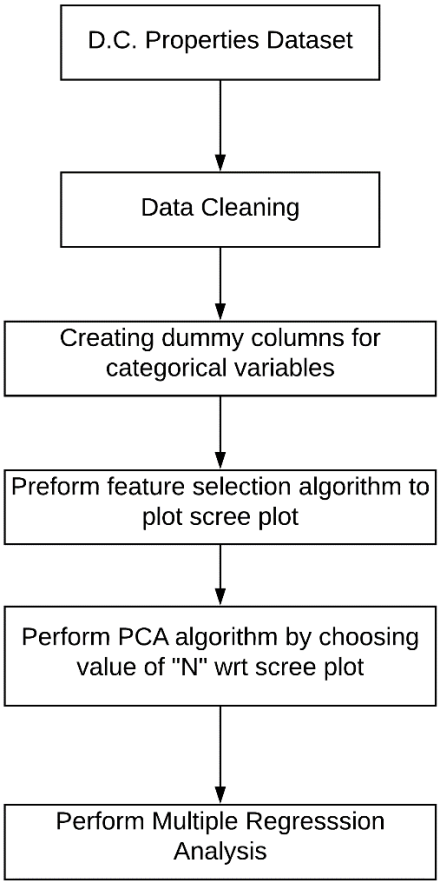

Figure 4: Flow Chart

To successfully fit he data into the model it is necessary to convert all the variables into numeric data thus all the categorical string data are converted into numeric data using Dummy Encoding methodology. There were total 15 categorical variables which were converted into numeric data. These variables are as follows: 'HEAT', 'AC', 'QUALIFIED', 'STYLE', 'STRUCT', 'GRADE', 'CNDTN', 'EXTWALL', 'ROOF', 'INTWALL', 'ASSESSMENT_NBHD', 'ASSESSMENT_SUBNBHD', 'WARD', 'QUADRANT'. After converting all these variables into numeric data, the total features in the dataset was 279.

Analysis

As there are 279 features in the dataset there must be some variables which shows high correlation with each other. The predictive analysis performs well when there is little correlation between the variables. Figure 5 shows top 20 pair of correlation exists in the dataset.

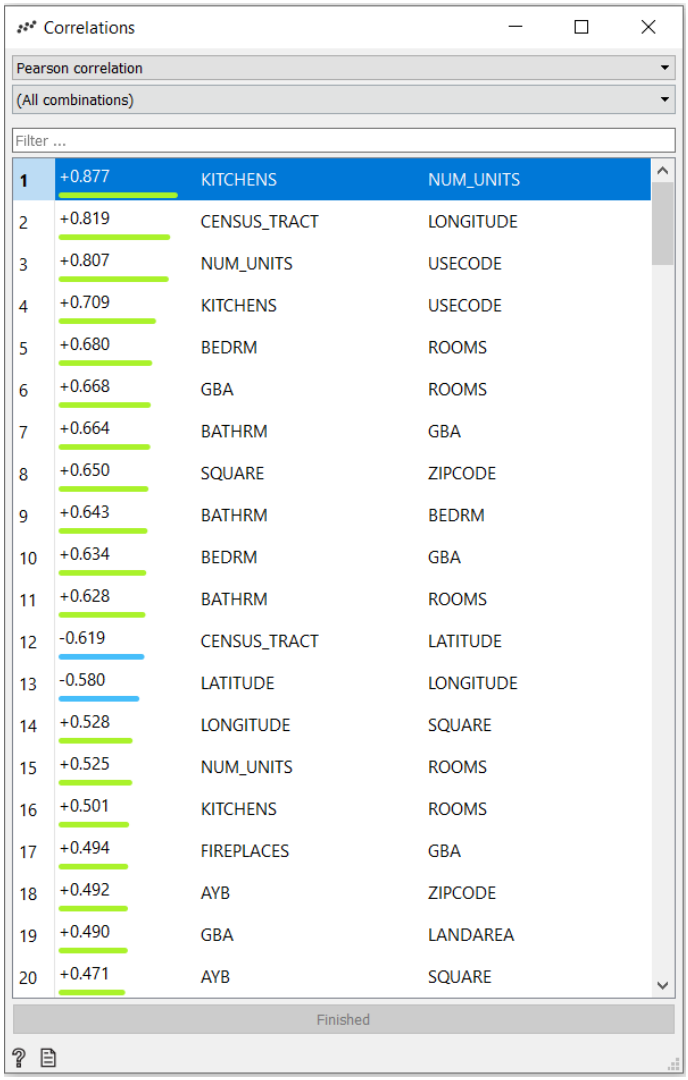

Figure 5: Correlation Table

The correlation graph of top 2 pair of features are shown in figure 6. It is essential to remove high correlation from the features. Principle Component Analysis (PCA) algorithm helps to reduce the dimensionality from the dataset and to reduce the correlation among the variables. However, PCA algorithm takes input from the user to decide how many numbers of variables we desire to keep in the output. Thus, feature selection algorithm is used to plot the scree plot to understand the eigen values of each variables, from which we can estimate what could be the input for PCA analysis for number of features to keep in the output.

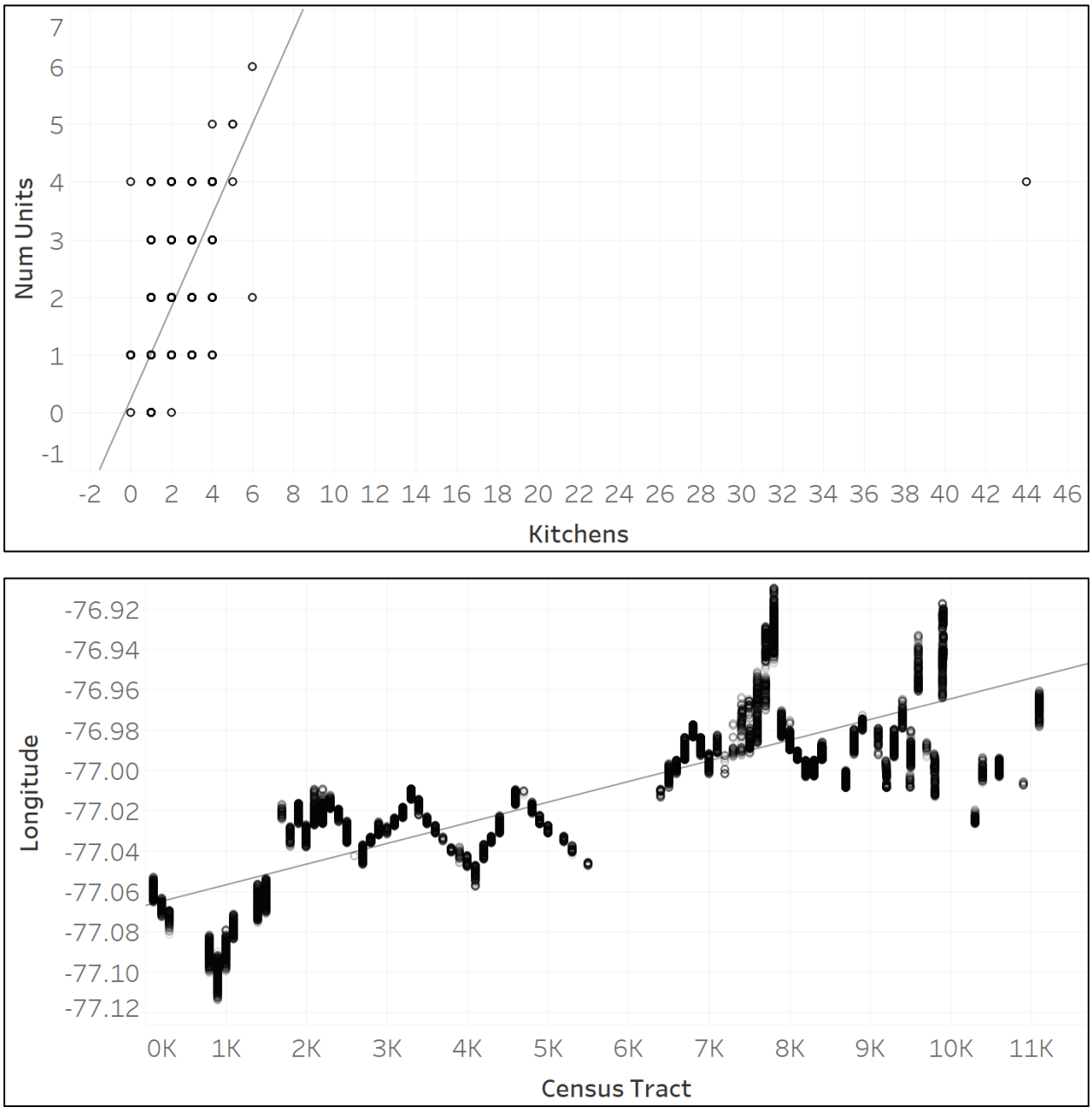

Figure 6: Scatter Plot of Top 2 Correlated variables

Figure 7 shows the output of factor analyzer, a feature selection algorithm to understand the eigen values of each variables. General practice to reduce the dimensionality and correlation from the dataset is by only considering the number of variables which has eigen value above value1. From figure 7 and 8 we can observe that there are 151 variables which shows eigen value above value1.

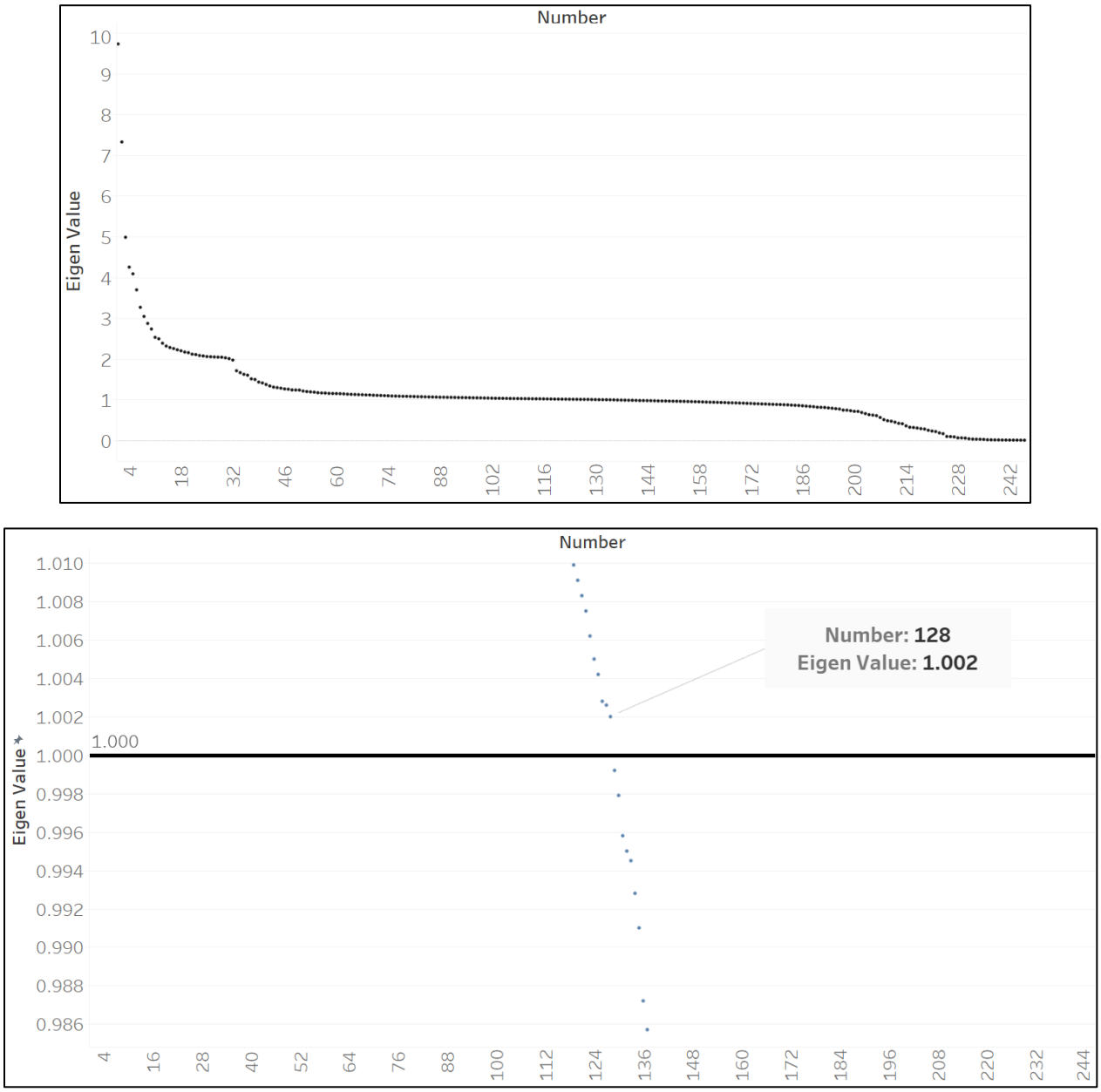

Figure 9: Scree Plot Normal and Zoomed Version

The next step is to perform the PCA analysis which will reduce the dimensionality from the dataset. Before running PCA it is essential to split the dataset into target variable and predictors variables. To perform PCA analysis it is also important to scale the dataset before performing the PCA algorithm. The output of the PCA algorithm is then used to split the dataset into training and testing

dataset to further run the Multiple Regression Algorithm for prediction analysis with “PRICE” as the target variable.

Results

After running the predictive analysis, the model performs decently well with 0.68 R^2 value and 340,940 as the Root Mean Squared Error (RMSE). Figure 8 shows the scatter plot of testing data and prediction data of our target variable i.e. “PRICE”. Figure 9 shows the residual plot of the target variable. The snapshot of the results of the predictive model is shown in figure 10.

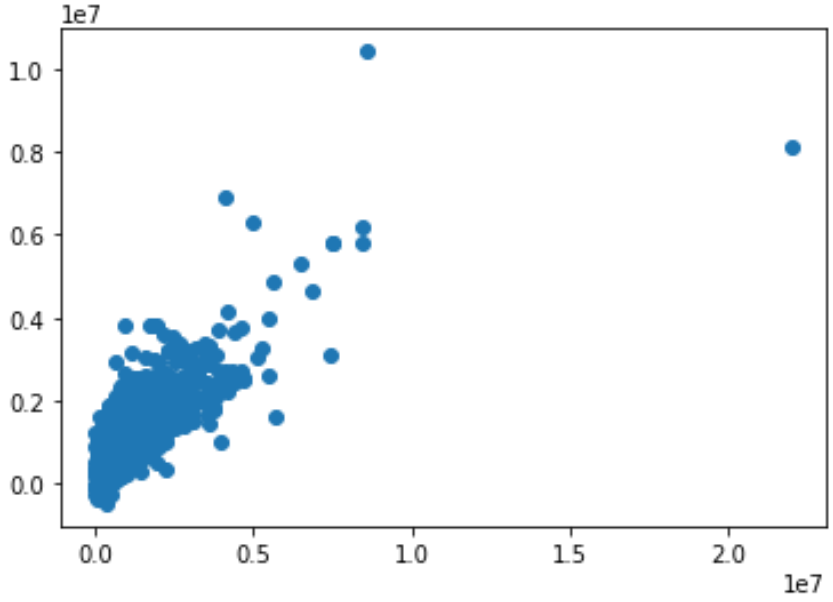

Figure 8: Testing vs Prediction Scatter Plot

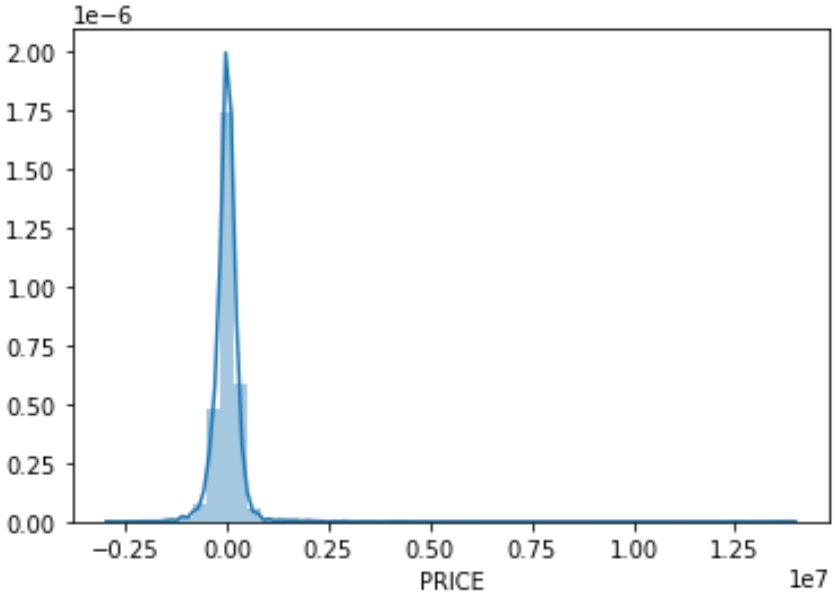

Figure 9: Residual Plot of Target Variable i.e. “PRICE”

```
In [22]: print('MAE:', metrics.mean_absolute_error(y_test, predictions))
         MAE: 198790.9939483259

In [23]: print('MSE:', metrics.mean_squared_error(y_test, predictions))
         MSE: 116240118959.55148

In [24]: print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
         RMSE: 340940.05185596994

In [25]: from sklearn.metrics import r2_score

In [26]: print('R2 Value: ', r2_score(y_test, predictions))
         R2 Value:  0.6767445172351785
```

Figure 10: Multiple Regression Predictive Model Results

Conclusion

Looking at the results, it can be concluded that the prediction model performs decently considering the huge dataset with 278 predictors which were reduced to 151 variables. The model shows 0.68 R^2 value and 340,940 as the Root Mean Squared Error (RMSE) and 198,791 as the Mean Absolute Error (MAE) when the target variable is widely spread with the range of Minimum value of 1 and Maximum value of 23,960,287 is consider fairly good.