VIDEO GAMES SALES

Final Report

Syed Qavi

Introduction

Video games are a very popular form of entertainment. Every year companies make hundreds of new video games. Popular video games are sold more than video games with less popularity. The popular video games also get more recognition by the users and critics which means that they get more people reviewing it. The dataset looks at video games sales of around 16720 games. The dates range from 1980 to end of 2016. The dataset looked at the name, genre, platform (xbox, playstation), publisher (company that released the game), Developer, Rating (i.e. T for teen, E for everyone), year of release, NA sales, EU sales, Japan sales, other sales (sales in rest of world), global sales (accumulation of all the sales), User count, User score (rating given by user), Critic count, Critic Score (rating given by critics) of the video game. We will take a look at various regression models along with cluster and PCA.

Dataset and Data Preprocessing

The dataset had a lot of missing variables. So, I decided to remove all rows that had missing variables. After dropping missing variables using pandas, the dataset had total of 6825 rows. While doing data analysis, I noticed that there were a lot of outliers, so using scipy stats package, I used z score to remove all the extreme outliers which were around 500. Also, before performing SGD, I standardized the features. The final dataset had around 6325 rows. I also removed variables that were not important for analysis such as name, year, platform, publisher, developer. I also created dummy variables for genre and rating of game. The final breakdown of variables are listed below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Variable Name | Variable Type | Variable Information |
| 1 | NA\_Sales | numeric | Sales in North America |
| 2 | EU\_Sales | numeric | Sales in Europe |
| 3 | Other\_Sales | numeric | Sales in rest of world |
| 4 | JP\_Sales | numeric | Sales in Japan |
| 5 | Critic\_Score | numeric | Score given by critics |
| 6 | Critic\_Count | numeric | Number of critics |
| 7 | User\_Score | numeric | Score given by user |
| 8 | User\_Count | numeric | Number of users |
| 9-13 | Rating | categorical | Total of 4 (T, E, M, E10+) |
| 14-22 | Genre | categorical | Total of 9 (Action, Fighting, Misc, Platform, Role-playing, Shooter, Sports, Simulation, Strategy) |

Methodology

Regression algorithms were used for supervised learning to discover how well variables can predict the target variable. Two target variables were used separately which include NA\_Sales and User\_Score. I wanted to see how these two different variables perform, the sales variables and the scoring variable. The regression methods include standard linear regression, Ridge regression, Lasso regression, simple linear regression (closed form), SGDRegresssor regression and Random forestation regression. Most of these regression methods were performed using scikit learn package. Matplotlib was used to plot train and test set. Tree was plotted using sklearn package. For unsupervised learning, k means cluster algorithm was used from scikit learn package along with PCA to discover the number of components that explain at least 95% of the variance using the sklearn decomposition package.

Data exploration was performed along with data visualization of all variables. Data was preprocessed by getting rid of unnecessary attributes, normalizing of all numeric attributes, removing outliers using z-score from stats package from sklearn and creating dummy variables. Also splitting data into train and test set into 80-20% was used before performing machine learning algorithm using model selection from sklearn package using random state of 33. Functions from examples were used such as calc\_params, standRegres, cross\_validate and others to perform model evaluation.

Data Exploration

Histograms were made for each of the numeric variables. The histograms showed a similar for all sales variables. The highest sales were observed in NA Sales where the range was from 0 to 5 millions sold for most games with some games going as high as 40. The lowest range was in Japan Sales range from 0 to 1 going up to 6 millions products sold. Global sales showed the accumulation histogram of all the sales variables.

As for user score and count, critics score and count, I got more users count than critics count with most games having 20 critics and about 1000 users scoring it. The distribution of critics and user score are very similar with users giving slightly higher score than critics which was around 8.

Scatterplot were made for Global Sales vs. NA Sales where I saw a positive correlation which is expected since they are similar in distribution. User score/Critics Score vs. Global Sales showed a similar distribution where games with higher sales were given higher score.

Bar charts of the categorical variables were created. For the Rating, T for teen had the highest while lowest were K-A, AO and RP which had no values since they were removed because of missing variables. For Genre, I saw that Action was the highest count while puzzle had the lowest count.

The dataset description table showed that NA sales ranged from 0 to 41, EU Sales from 0 to 29, Japan Sales from 0 to 6.5, Other Sales from 0 to 10.5. Global Sales from 0 to 82.5. Critic score from 13 to 98. The max count of critics was 113 compared to user count which was 10665. The std was highest for user count. After normalizing the dataset, the description table showed that highest mean and highest std was Rating Teen.

I also did a correlation of the variables after doing the min max normalization which showed that Global Sales had multicollinearity with the other sales variables. Also, some of the dummy variables showed nan which means that they had no data.

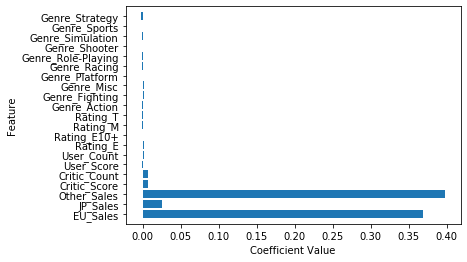
Regression

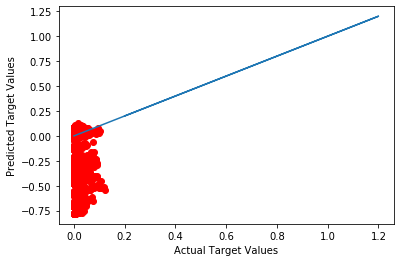
Target Variable: NA Sales

Data was split into 80 to 20 with train and test set having 5060 and 1265 rows.

Standard linear regression:

standRegres method was used from textbook to fit the model. RMSE was calculated on the full training set which was the 80% part. plot\_coefficients function was used to plot the coefficients where Other sales and EU sales were observed to have the highest. Also correlation between the predicted and actual values of the target attribute were plotted. All the points were from 0 to 0.2 actual target values and 0.25 to -0.75 predicted target values. Cross\_validate function was used from examples notebook to perform 10-fold cross-validation. I used it to compare the cross-validation RMSE to the training RMSE. Sklearn KFold package was used to perform this. Results were:



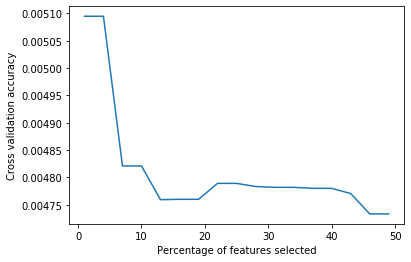


|  |  |
| --- | --- |
| RMSE on Training | 0.3949 |
| RMSE on 10-fold CV | 0.0740 |

Feature selection:

I used the scikit-learn regression model from sklearn.linear\_model package to perform linear regression. I took as input the training data, target variable, the model. I also used 10-fold cross-validation on the training data. I also used feature\_selection SelectPercentile package to find the most informative variables. Then, I plot the model's MAE on cross-validation using only the selected features. I trained the model on the full 80% training data and evaluated it on 20% test data. Optimal number of features were found to be 9 and Optimal percentile of features was 46.

|  |  |
| --- | --- |
| Optimal number of features | 46 |
| Optimal percentile of features | 9 |



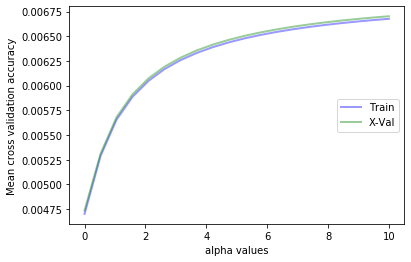
Mean Absolute Error on Test: 0.004575112079702553

Ridge and Lasso Regression:

They were performed using sklearn linear model package. Model Selection was used to find the optimal alpha parameter. calc\_params function was used to find the optimal value for alpha for both ridge and lasso regression. The function also performed K-fold cross validation and a plot was created for the error values on training and cross validation splits for the alpha parameter that was specified as the input. After finding the optimal alpha values, I trained the model to the 80% training set and evaluated it on the 20% test set. The results were:

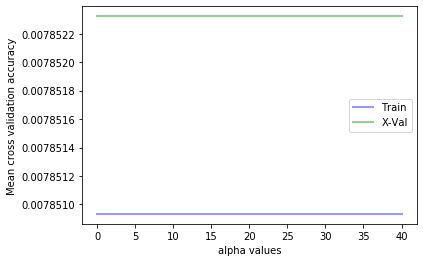
**Ridge Regression** optimal alpha value: 0.00001

|  |  |
| --- | --- |
| RMSE on Test | 0.0083 |
| MAE on Test | 0.0046 |



**Lasso Regression** optimal alpha value: 0.01

|  |  |
| --- | --- |
| RMSE on Test | 0.0117 |
| MAE on Test | 0.0076 |



Stochastic Gradient Descent:

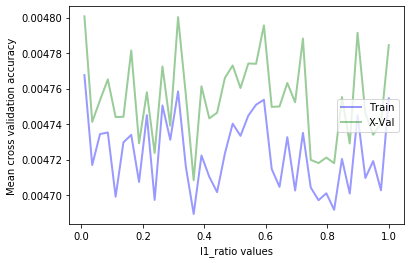
Next, regression was performed using SGD. This is also found in the sklearn linear model package. Before doing this, features were standardized using the standard scaler from sklearn preprocessing package. Grid search was also performed using GridSearchCV from sklearn grid search package to find the optimal parameters for SGD. The grid search compared the combinations of l2 (ridge), l1 (lasso) penalty parameters. Then, model selection was performed similar to ridge and lasso regression where I found the best alpha value as 0.01 and penalty parameter as l2 which I used to fit the model using SGD for both train and test set separately which resulted in:

|  |  |
| --- | --- |
| RMSE on Train | 0.0085 |
| MAE on Train | 0.0050 |

|  |  |
| --- | --- |
| RMSE on Test | 0.0083 |
| MAE on Test | 0.0046 |

When comparing both, I find that train and test set give similar results so I don’t have overfitting. Then I used calc\_params method using elacticnet as penalty parameter which combines both l2 and l1 for SGDRegressor. The results after using the function were:

|  |  |
| --- | --- |
| Minimum MAE | 0.004689 |
| L1\_ratio | 0.36548 |



I used the l1 ratio to fit the model to set aside test data which resulted in:

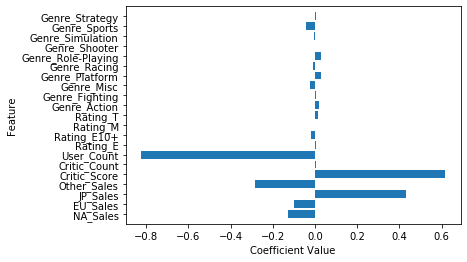
|  |  |
| --- | --- |
| RMSE on Train | 0.0083 |
| MAE on Train | 0.0045 |

Stochastic Gradient Descent had lowest MAE at 0.0045 for NA\_Sales.

Target Variable: User Score

Same methodology was used to perform regression for target variable User Score except for the addition of Random Forest regression. To avoid redundancy, the results for the specified regression are listed below:

Standard linear regression:



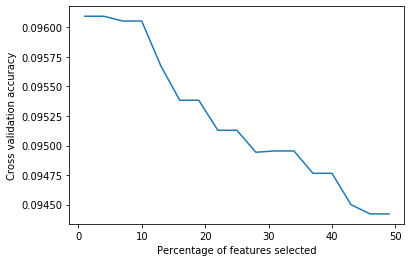


|  |  |
| --- | --- |
| RMSE on Training | 20.6288 |
| RMSE on 10-fold CV | 9.1301 |

We can observe from the results that the RMSE value was much higher compared to NA Sales variable.

Feature selection:

|  |  |
| --- | --- |
| Optimal number of features | 46 |
| Optimal percentile of features | 9 |



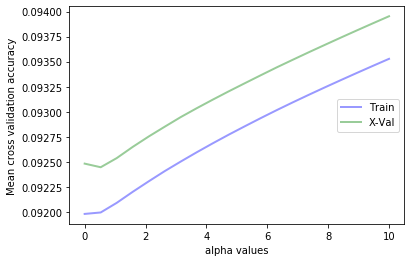
Mean Absolute Error: 0.09219826520249533

The number of optimal features and percentiles were the same as NA sales variable. The graph is different. The MAE value was also higher.

Ridge and Lasso Regression:

**Ridge Regression** optimal alpha value: 0.526

|  |  |
| --- | --- |
| RMSE on Test | 0.1217 |
| MAE on Test | 0.0923 |



The graph different than NA sales, both Train and X-val further apart and MAE values higher.

**Lasso Regression** optimal alpha value: 0.01

|  |  |
| --- | --- |
| RMSE on Test | 0.1425 |
| MAE on Test | 0.1099 |



Graph shows constant until 0.1096 then increases and stays constant. MAE value is higher than NA Sales.

Stochastic Gradient Descent:

Using alpha: 0.01 and penalty: l2

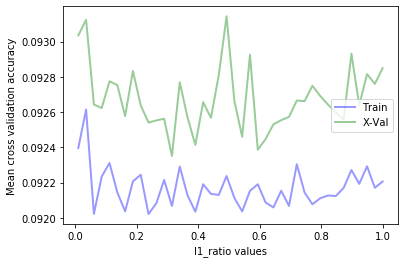
|  |  |
| --- | --- |
| RMSE on Train | 0.1234 |
| MAE on Train | 0.0924 |

|  |  |
| --- | --- |
| RMSE on Test | 0.1216 |
| MAE on Test | 0.0019 |

Again these values are higher than NA Sales.

Using calc\_params function with elacticnet as penalty parameter, results showed:

|  |  |
| --- | --- |
| Minimum MAE | 0.09206 |
| L1\_ratio | 0.3146 |



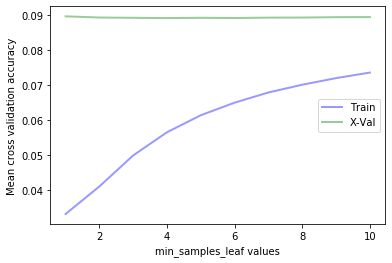
The Train and X-val are further apart than NA sales.

I used the l1 ratio to fit the model to set aside test data which resulted in:

|  |  |
| --- | --- |
| RMSE on Test | 0.1222 |
| MAE on Test | 0.0928 |

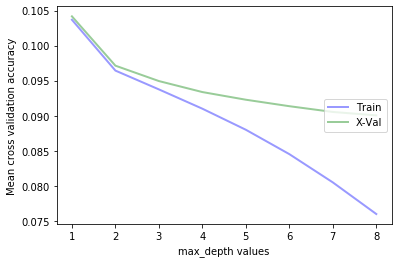
Random Forest:

This is an ensemble method which will be used to see if we can lower the MAE value for User\_Score. First, the data was split into train and test set and RandomForestRegressor from sklearn ensemble package was used to fit the model. Then the calc\_params function was used to find the min samples leaf which was observed to be 3.



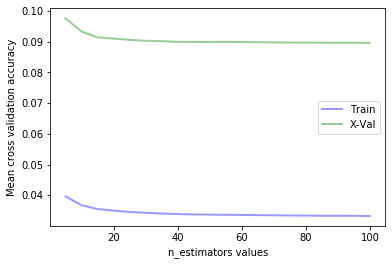
min\_sample\_leaf = 3

Then the calc\_params function was used to find the max depth which was approximated to be 3.



max depth = 3

Using range of 5-100 with interval of 5, calc\_param function was used to find number of estimators which was 45.



n\_estimators = 45

Finally, after finding the optimal parameters, I fitted the model to train and test set using RandomForestRegressor with the parameters 3 for min sample leaf and max depth and 45 as number of estimators and MAE as the criterion since I have been using MAE to compare the different regressions as opposed to the default MSE.

A plot of important features to the target variable User score was created:



Clearly critic score was the most important to user score which is expected since both had similarity in distribution.

I then looked at the three estimators for random forest:

DecisionTreeRegressor(criterion='mae', max\_depth=3, max\_features='auto',

max\_leaf\_nodes=None, min\_impurity\_decrease=0.0,

min\_impurity\_split=None, min\_samples\_leaf=3,

min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

presort=False, random\_state=1453202468, splitter='best'),

DecisionTreeRegressor(criterion='mae', max\_depth=3, max\_features='auto',

max\_leaf\_nodes=None, min\_impurity\_decrease=0.0,

min\_impurity\_split=None, min\_samples\_leaf=3,

min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

presort=False, random\_state=676685682, splitter='best'),

DecisionTreeRegressor(criterion='mae', max\_depth=3, max\_features='auto',

max\_leaf\_nodes=None, min\_impurity\_decrease=0.0,

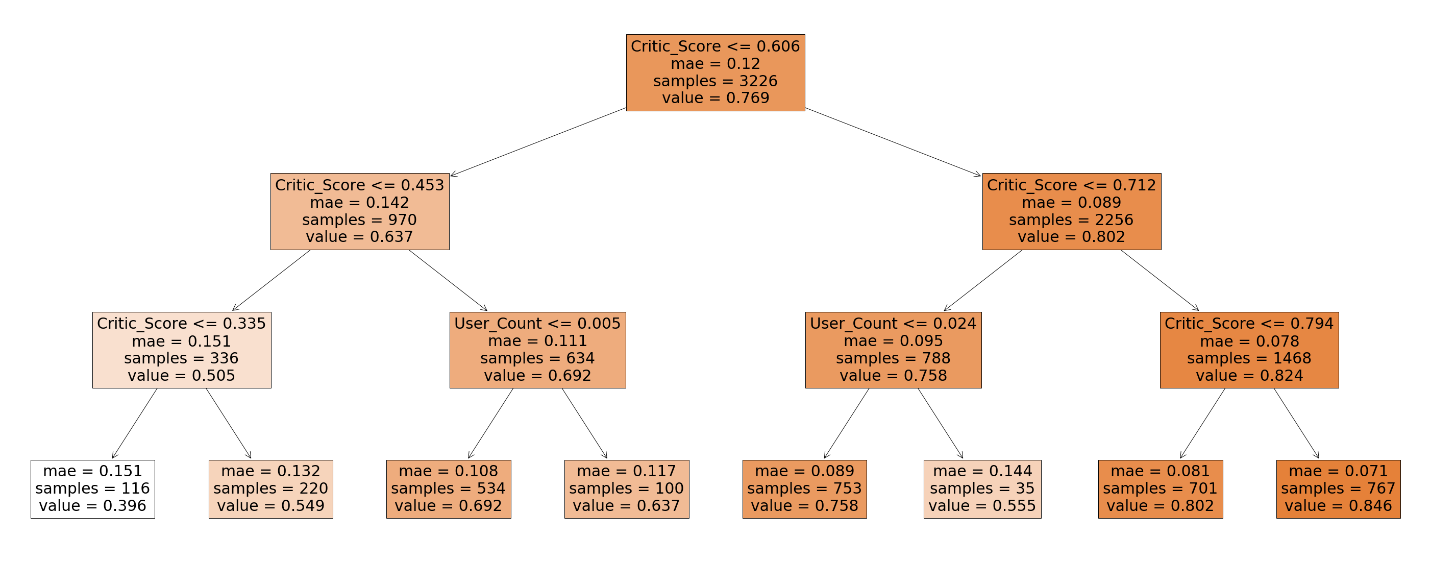
min\_impurity\_split=None, min\_samples\_leaf=3,

min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

presort=False, random\_state=1338583429, splitter='best')

All three are similar to each other.

A tree was plotted using tree from sklearn package.

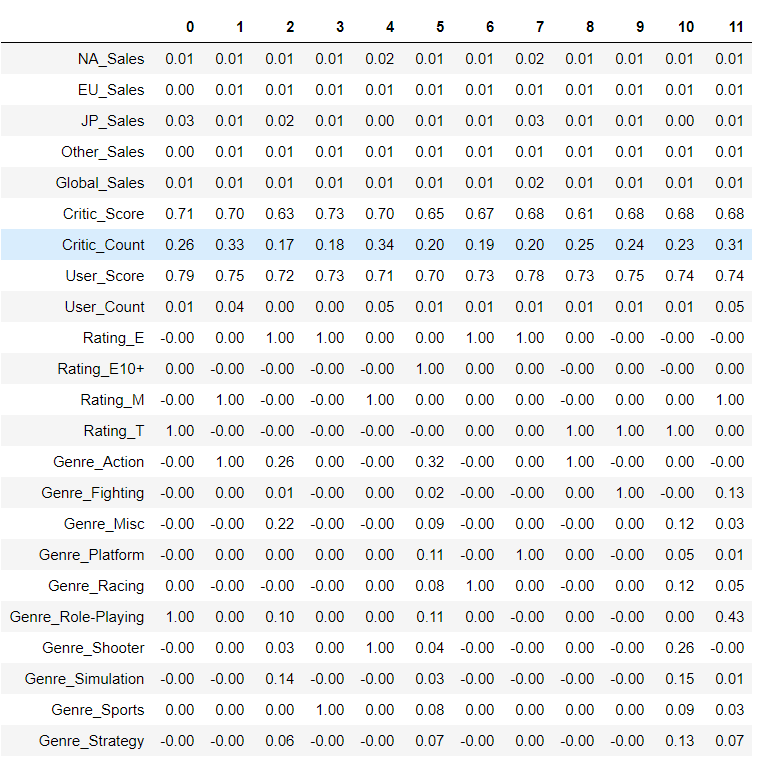


We can see that the MAE value is lowest on the right side of the tree. Recall that the data has been normalized to fit between 0 and 1. Doing a split between User count and Critic score which are the two highest related variables to user score. We can see that critic score <= 0.794 has lowest MAE and this is expected since both critic score and user score had highest count around 8.

For User\_Score, standard linear regression had by the lowest MAE value at 0.0921. But from the tree, we can observe that when you split, you can get MAE to 0.071.

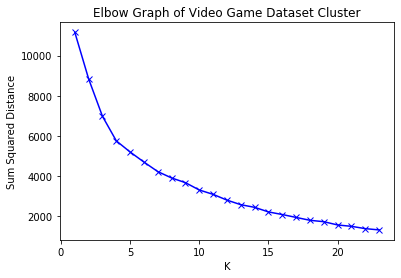
Cluster and PCA

After loading the data and performing the min max normalization to the full dataset, K means cluster was performed using the package from sklearn. K =12 was used to compare to PCA. After that I printed the centroids.



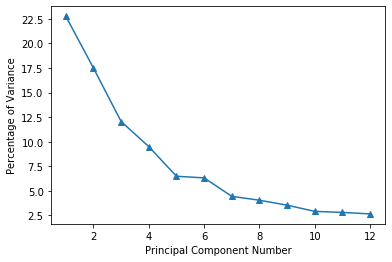
Highest centroids are seen in Critic score, critic count and user score. Lowest are seen in the sales variables.

Then I looked at the cluster sizes using the cluster\_sizes function which showed that the highest size was for cluster 10. Below is the elbow plot of the kmeans cluster.

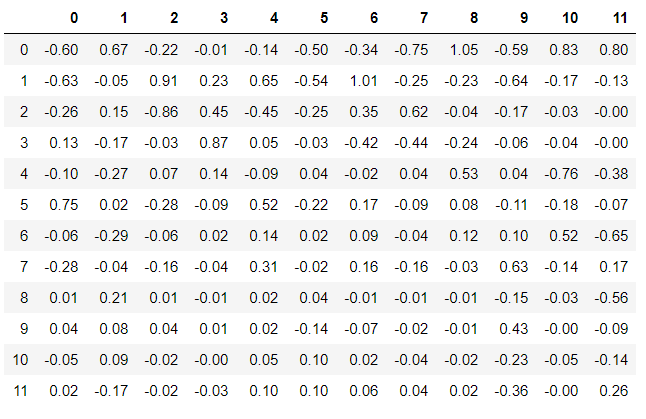


The number of K could be approximated between 5 and 10

For PCA, Decomposition module in scikit-learn was used to determine the number of components needed for at least 95% of variance in data. The number was 12 components explain at least 95% of variance. Then a plot of PC variances was made:



This is table of centroids of all 12 components attained from lower dimensional transformed data.



Conclusion

Machine learning algorithms used in the data analysis include various regression which include standard linear regression, ridge and lasso regression, Stochastic Gradient Descent, and random forest regression. K-means cluster and PCA was also performed. Regression was performed on two target variables separately, NA\_Sales and User\_Score. For NA\_Sales, Stochastic Gradient Descent had the lowest MAE value while for User\_Score had Simple linear regression had the lowest MAE value. Feature selection had 9 as optimal number of features for both User\_Score and NA\_Sales. NA\_Sales had much lower MAE value which means that other variables were better at predicting it compared to User\_Score. Cluster and PCA showed that 12 components of the 28 total variables explained 95% of the variance.

APPENDIX

All code and plots or any other information used in this final report can be found in the attached Jupyter notebook.