# Ahsanullah University of Science and Technology

## *Department of Computer Science & Engineering*

| | |
|---|---|
| Course No. | CSE 4108 |
| Course Name | Artificial Intelligence Lab |
| Project | Television Price Prediction |

**Submitted To:**

| | |
|---|---|
| Faisal Md Shah | Md. Siam Ansary |
| Department of CSE, AUST | Department of CSE, AUST |

**Submitted By:**

| | |
|---|---|
| Shaiuf Sadique | 18.01.04.111 |
| Alphi Shahrin | 18.01.04.121 |
| Section: | C1 |
| Group: | 05 |

## Introduction

In this project, we are going to predict the price of television by training regression models. We collected the data from numerous websites where we found the specifications and the price of the televisions. After collecting data we trained our model with some machine learning algorithms so that we can predict the price of television. In the ‘performance table’ section we have compared the performance of each algorithm and concluded the best regression model for our operation.

## Brief Description of the Dataset

| | |
|---|---|
| Name of the Dataset | Dataset |
| File format of the Dataset | csv |
| Dimension of the Dataset | 333x12 |
| Number of Total Columns | 12 |
| Number of Total Rows | 333 |
| Number of Feature Columns | 12 |
| Name of Feature Columns | Brand, Screen Size, Resolution, Device Type, Power Supply, Audio Output, Speaker System, HDMI, USB, Smart TV, Resolution Upscale, Price |
| Number of Target Columns | Price |

## Dataset Description

From different online store websites we collected a good collection of data consisting of 333 rows and 12 columns. The columns are: Brand, Screen Size, Resolution, Device Type, Power Supply, Audio Output, Speaker System, HDMI, USB, Smart TV, Resolution Upscaler, Price We encrypted the columns according to their own specific values. Like for the Brand column we put 12 for Samsung, 11 for Sony, 10 for LG, 9 for Xiaomi, 8 for Konka, 7 for Vision, 6 for Walton, 5 for Jamuna, 4 for Minister, 3 for Singer, 2 for Marcel, 1 for MyOne. The final target column is called Price.

Splitting data set

- Data used for training = 80% (266 rows are used
- Data used for testing = 20% (67 rows are used)

## Description of the Models

1. **Linear Regression:** The first model we used in our project is Linear Regression. For the training 75 percent data is used and the rest is used in the test set. It is perhaps one of the most well-known and well understood algorithms in statistics and machine learning. Linear regression is a linear model, e.g., a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x). When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, literature from statistics often refers to the method as multiple linear regression.

2. **XGBoost:** XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models. XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a penalty term for model complexity (in other words, the regression tree functions). The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction.

3. **Lasso Regression:** Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This regression is well-suited for models showing high levels of multi co linearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination. Lasso Regression uses L1 regularization technique.

4. **Random Forest:** The fourth model is used in our project is Random Forest algorithm. Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

5. **Nearest Neighbours Regression:** The fifth model is used in my project is KNN. KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same neighbourhood. The size of the neighbourhood needs to be set by

the analyst or can be chosen using cross-validation to select the size that minimizes the mean squared error.

**Comparison of the performance scores:**

For “Dataset.csv”, five below models are used to justify and measures the result of the dataset.

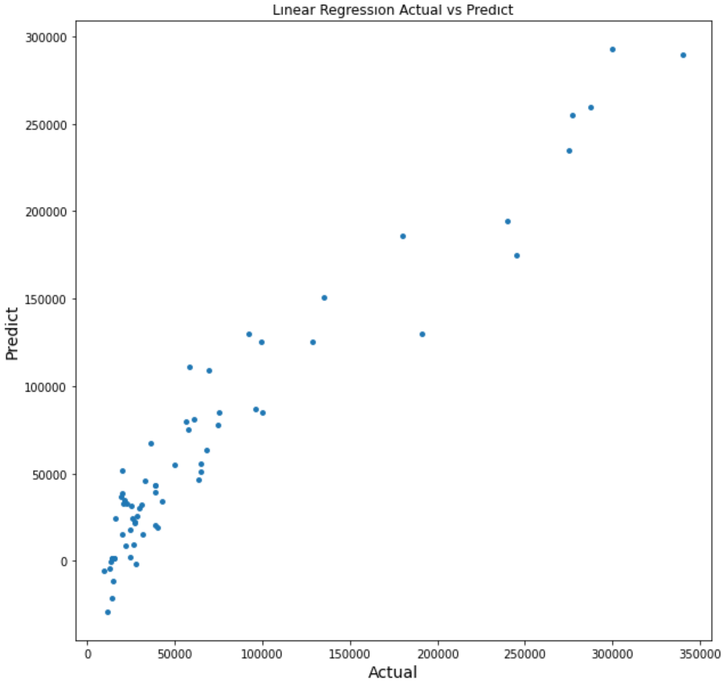
- Linear Regression
- XGBoost
- Lasso
- Random Forest
- KNN

To compare the results four performance metric scores are executed. They are Accuracy, Recall, Precision, F1 score. The result comparison table is given below.

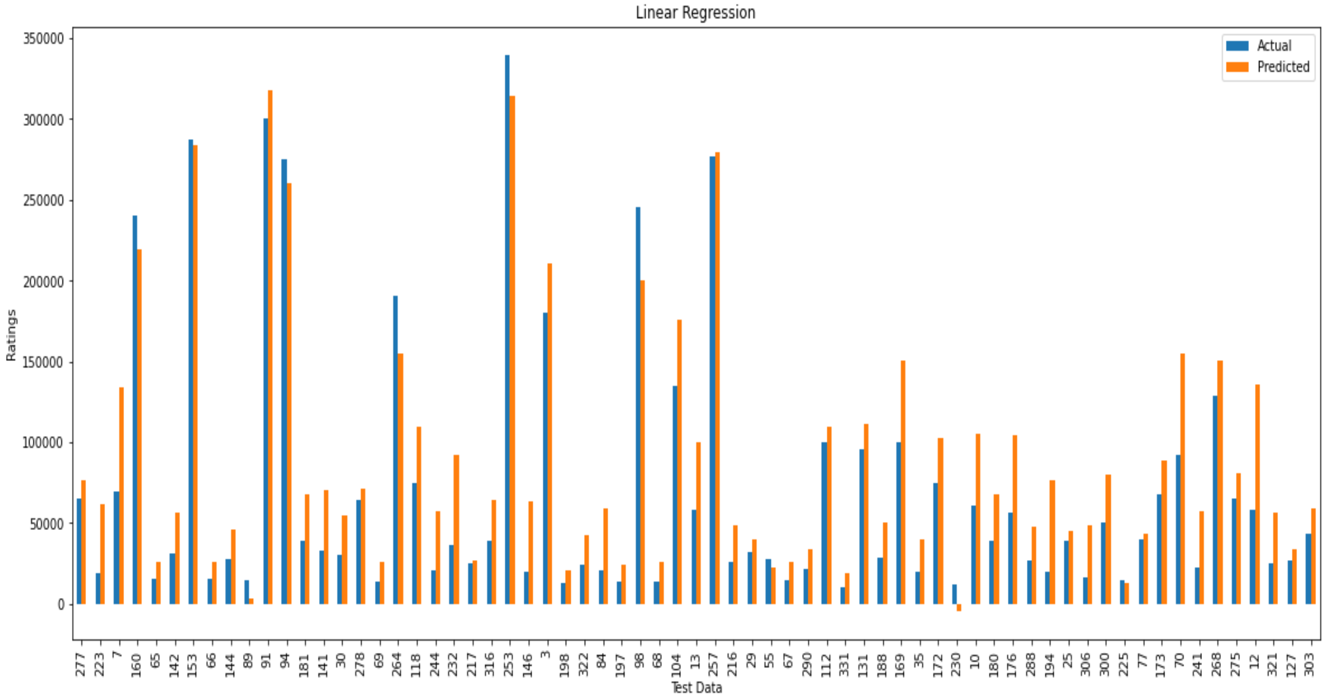
| Name of the Algorithm | Algorithm Score | Mean Absolute Error | Mean Squared Error | Root Mean Squared Error | R2 Score |
|---|---|---|---|---|---|
| **Linear Regression** | 0.925 | 18103.48 | 549974021.97 | 23451.52 | 0.916 |
| **XGBoost** | 0.992 | 5882.40 | 92272183.70 | 9605.84 | 0.986 |
| **Lasso** | 0.925 | 24785.972 | 898193461.15 | 29969.87 | 0.864 |
| **Random Forest** | 0.993 | 6670.72 | 149533492.93 | 12228.38 | 0.977 |
| **KNN** | 0.980 | 9486.97 | 307209411.97 | 17527.39 | 0.953 |

**Discussion:**

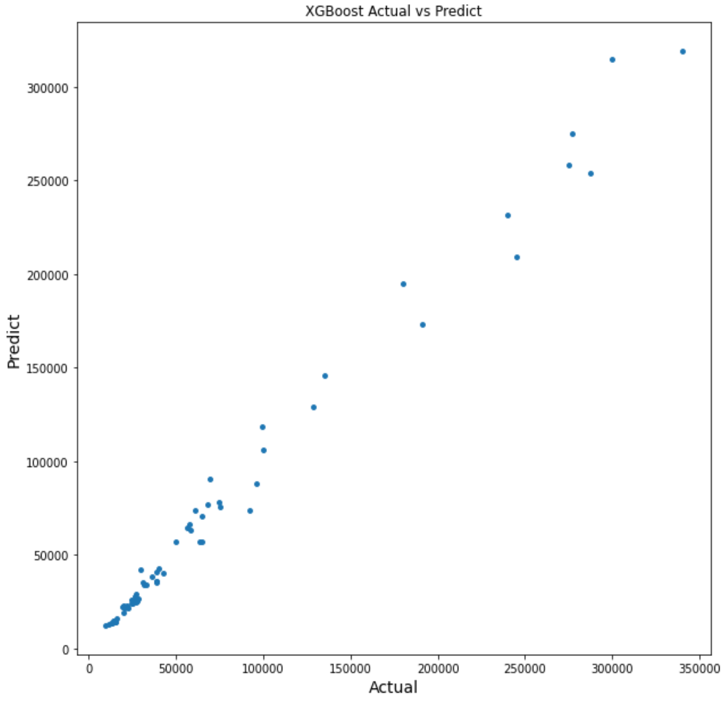
Now let’s see a graph of Actual vs Predict of the first model: Linear Regression Model:



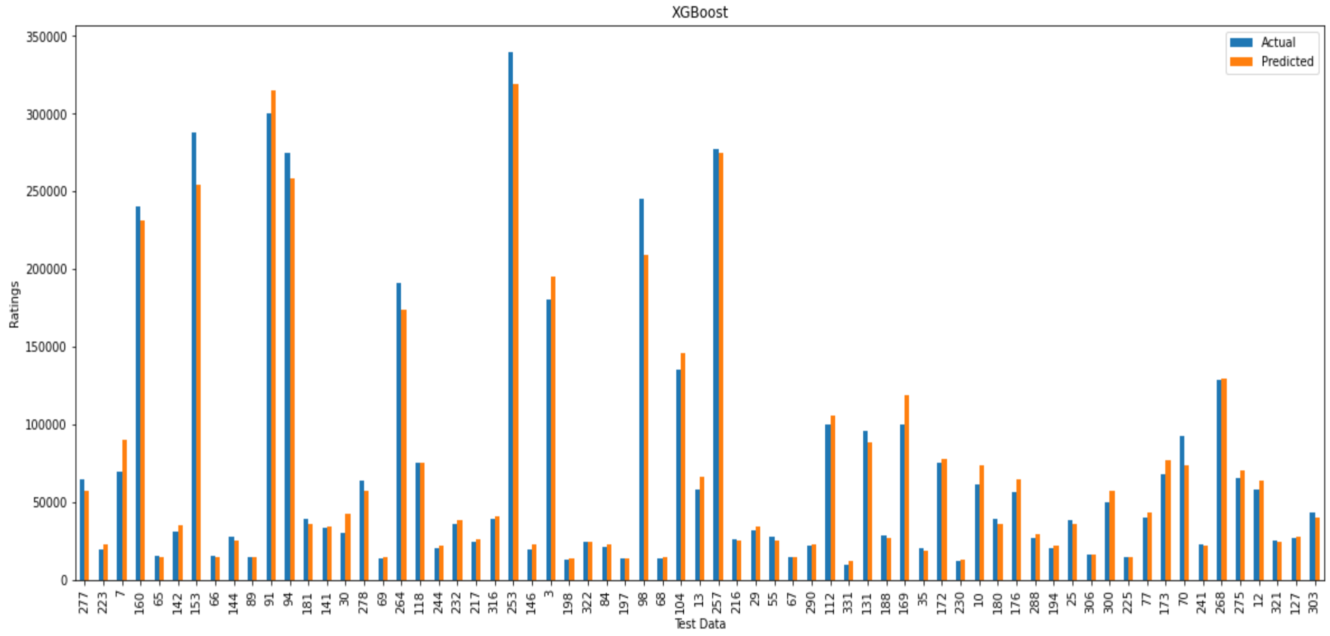
A histogram of Actual vs Predict of Linear Regression Model:



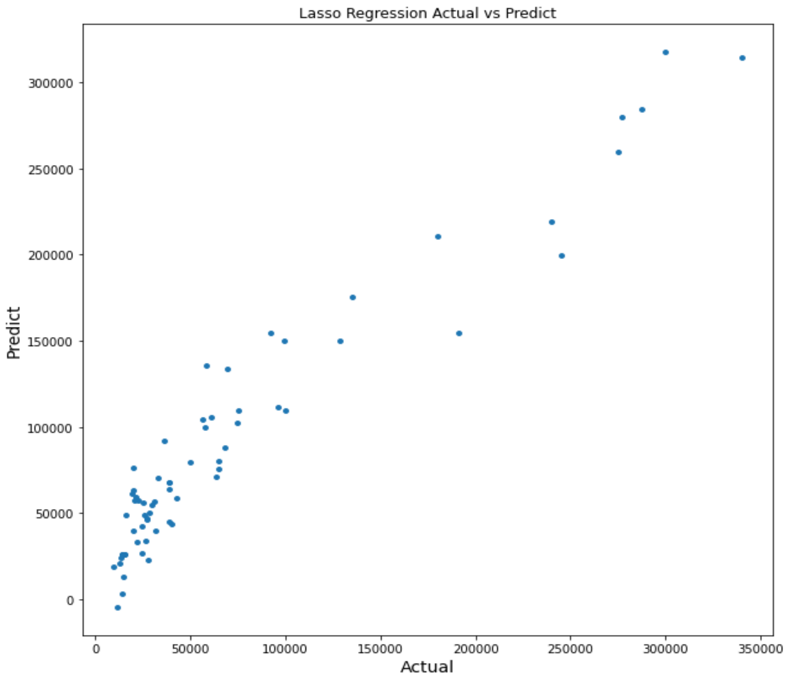
A graph of Actual vs Predict of the second model: XGBoost Model:



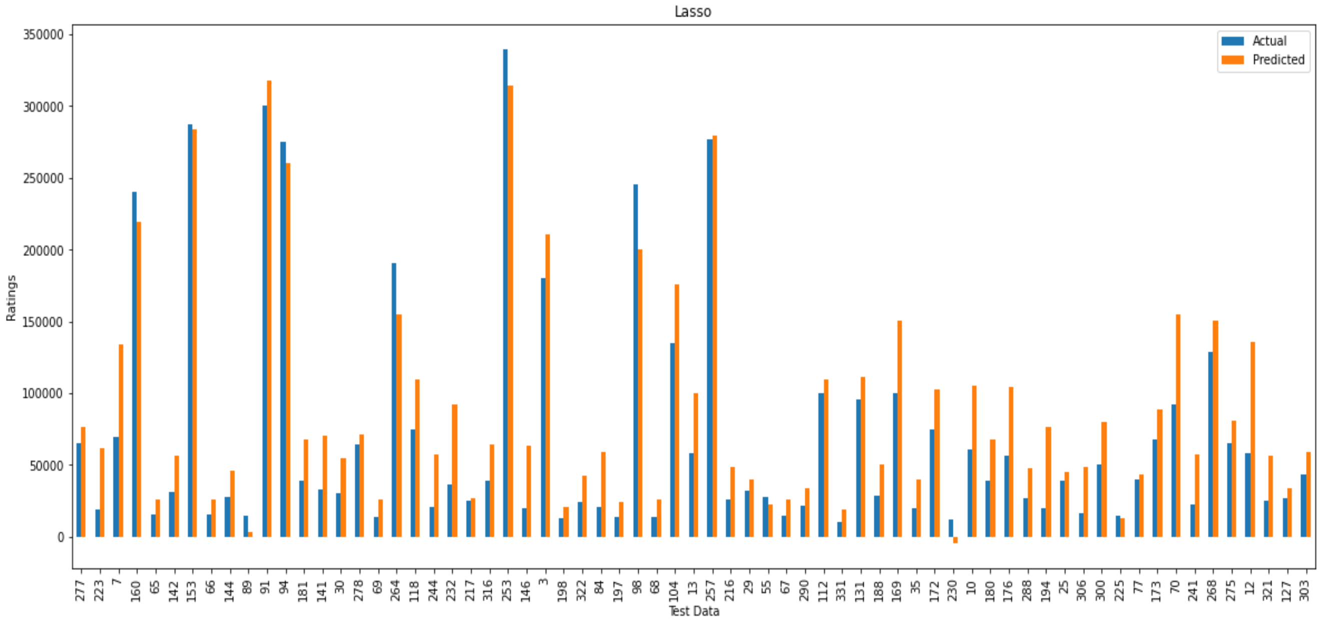
A histogram of Actual vs Predict of XGBoost Model:



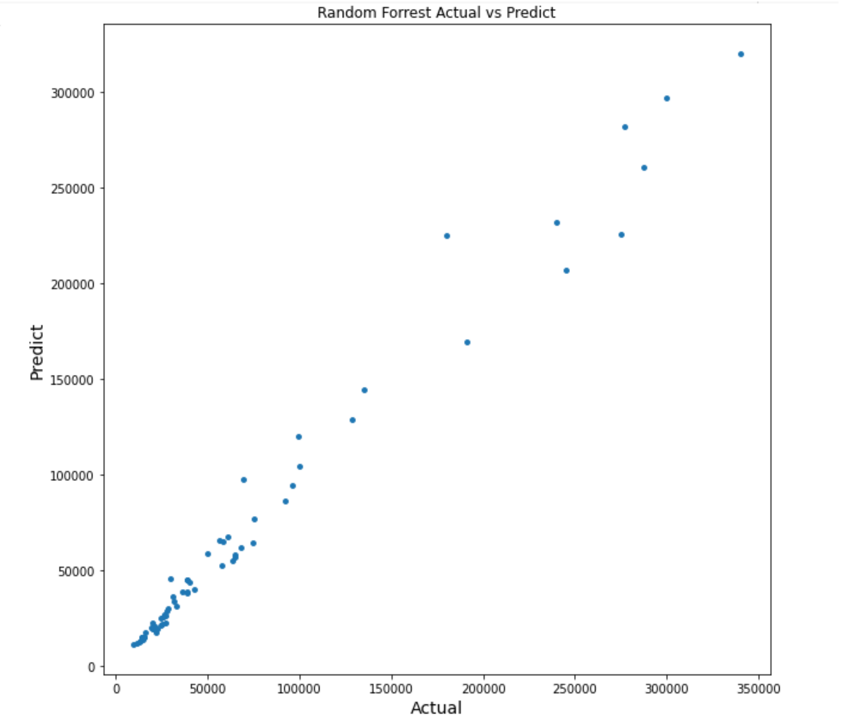
A graph of Actual vs Predict of the second model: Lasso Regression



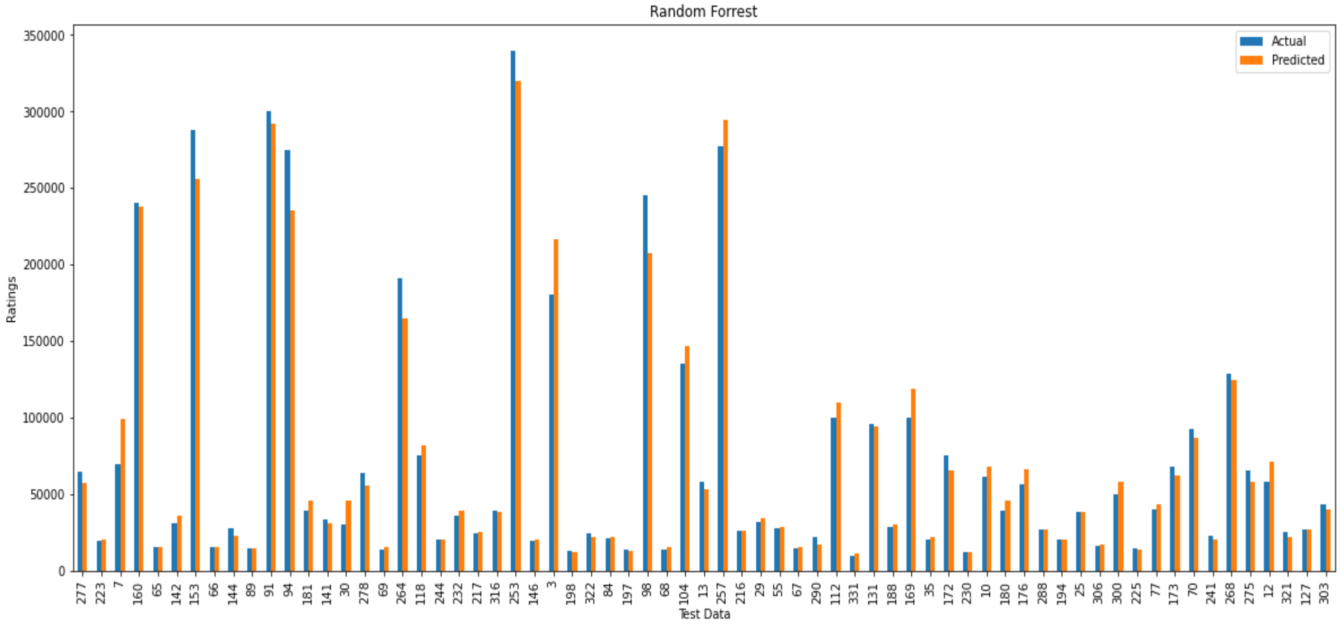
A histogram of Actual vs Predict of Lasso Regression



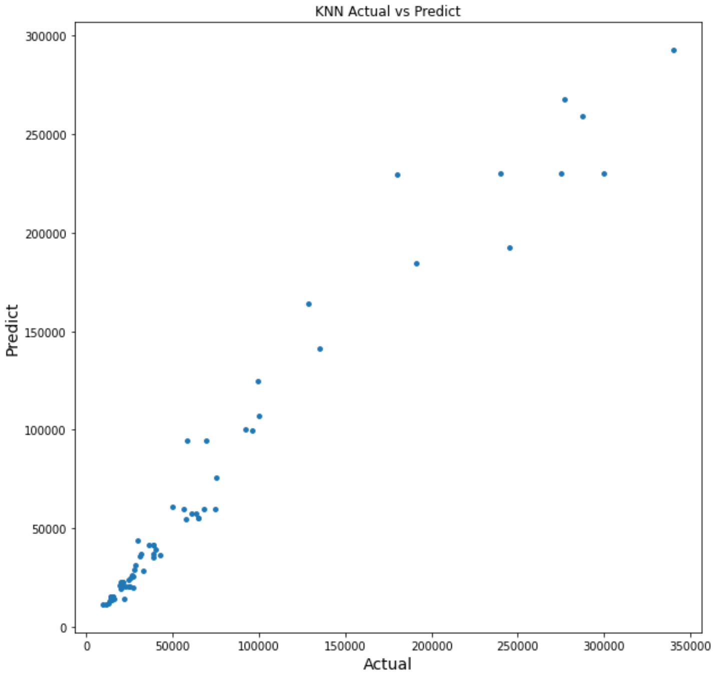
A graph of Actual vs Predict of the second model: Random Forest



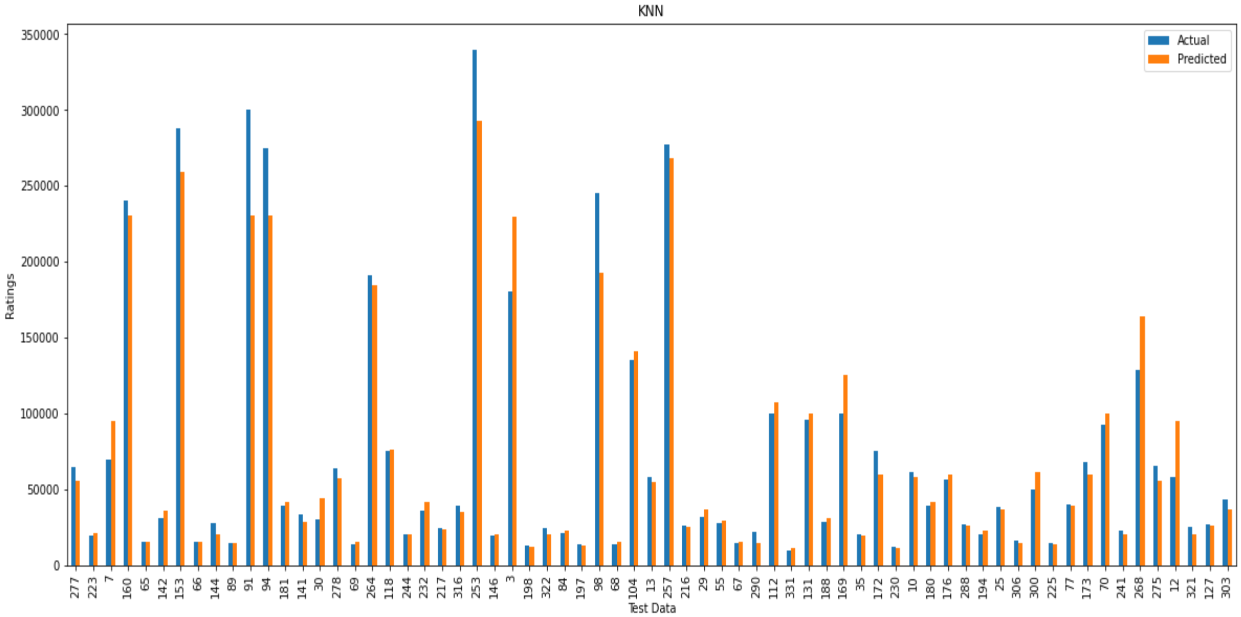
A histogram of Actual vs Predict of Random Forest



A graph of Actual vs Predict of the second model: KNN



A histogram of Actual vs Predict of KNN Model:



**Members Contribution:**

18.01.04.111: 50% - Dataset collection of Sony, Walton, Konka, Singer, Jamuna, Xiaomi, Vision. Code implementation of Linear Regression and KNN.

18.01.04.121: 50% - Dataset collection of Samsung, LG, MyOne, Marcel, Minister. Code implementation of XGBoost, Random Forest, Lasso.