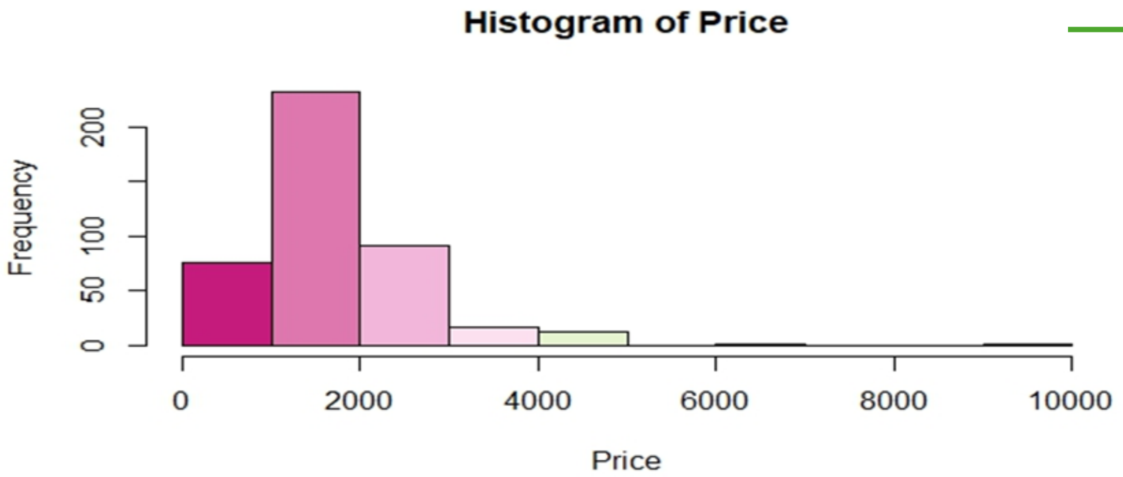
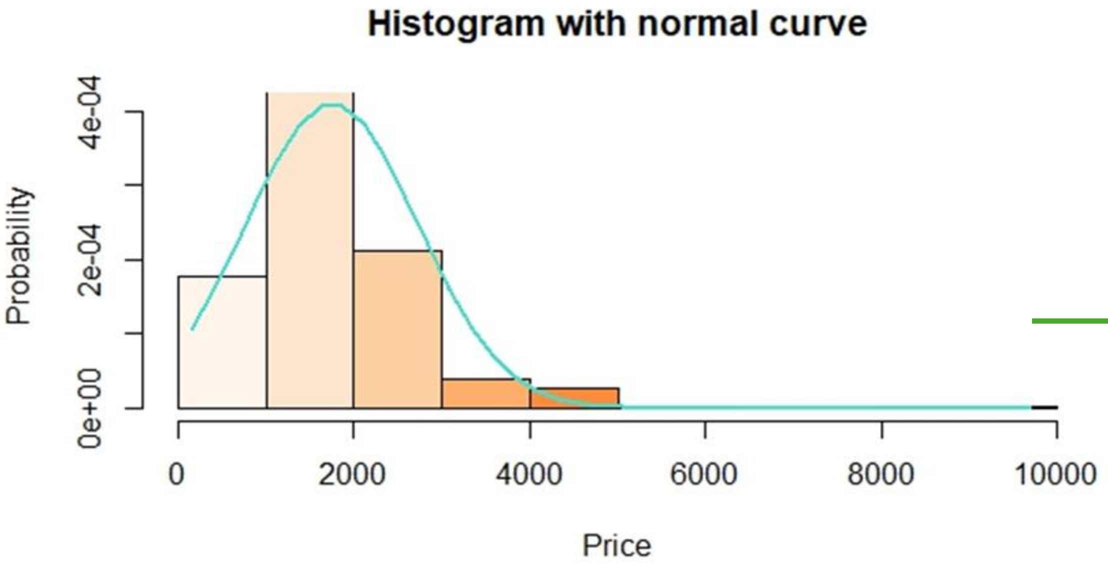
MACHINE LEARNING & REGRESSION ANALYSIS

- Amrita Mondal

- Normality Checking (Histogram)
- Factors Influencing Product Price (Scatter Plot, Boxplot, Kruskal-Wallis Test)
- SVM Model
- Random Forest
- RMSE : SVM vs. Random Forest

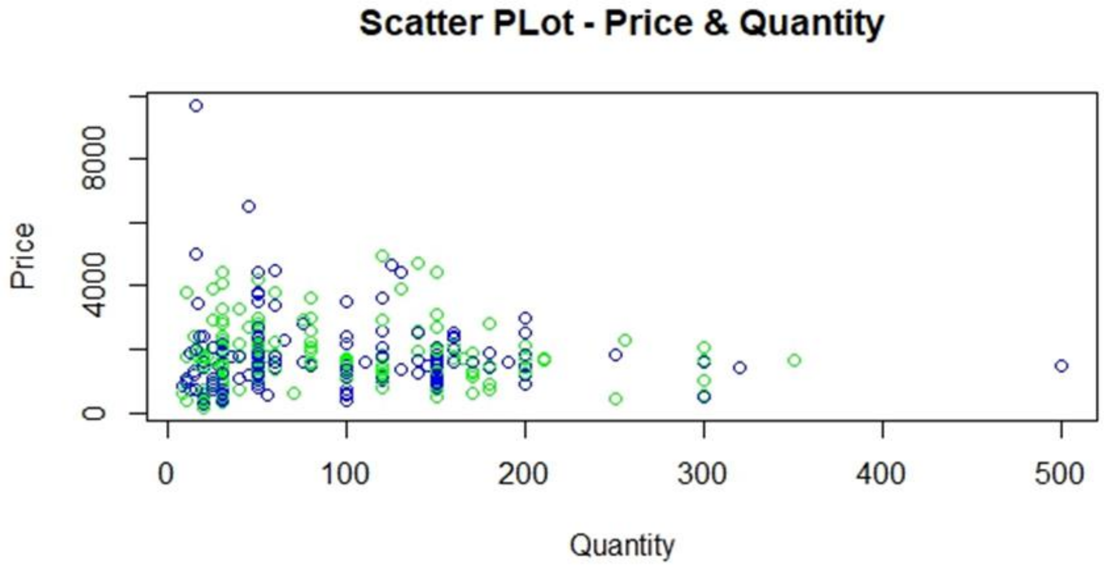
NORMALITY CHECKING & SCATTER PLOT


Histogram of Price


Histogram with normal curve

i. Product Price - positively skewed, not Normal distribution.

ii. Shapiro-Wilk test confirms the acceptance of Alternative Hypothesis that the distribution is not Normal, with 5% level of significance. (W = 0.8529, p-value = 2.2e-16)


Scatter PLot - Price & Quantity

- No presence of linear relationship, but non-linear relation may exist.

KRUSKAL-WALLIS TEST

Assumptions :

i. Data are assumed to be non-Normal or a skewed distribution.

ii. The variable of interest should have two or more independent groups. For two groups, Mann-Whitney and for three or more, Kruskal-Wallis Test is used.

iii. Similar distribution across the groups.

iv. Randomly selected independent samples.

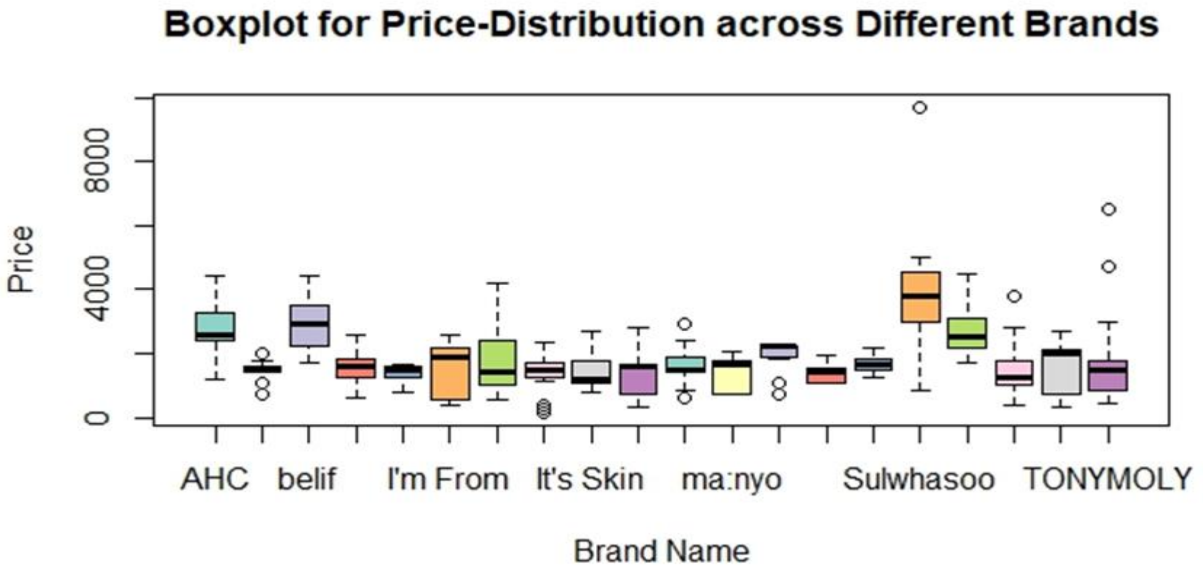
v. Each group sample should have at least 5 observations for a sufficient sample size.

Null Hypothesis : There is no significant difference between the two groups being compared, meaning the populations from which the samples were drawn are essentially the same, with equal distributions (or medians) across both groups.

Alternative Hypothesis : There is significant difference between the two groups being compared.

The test is carried out at $\alpha = 0.05$, level of significance.

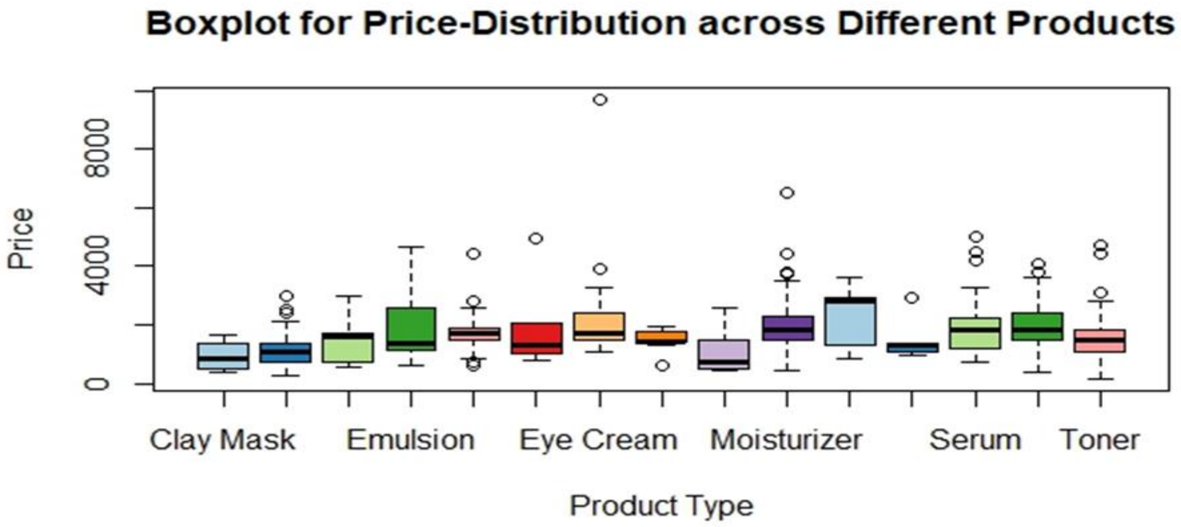
FACTORS INFLUENCING K-BEAUTY PRICE

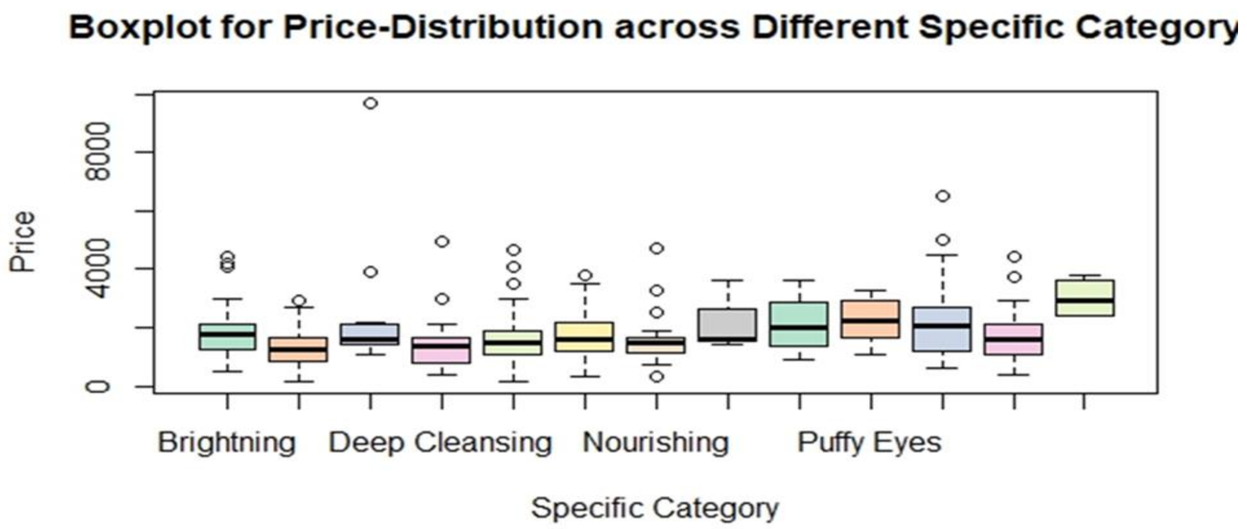

Boxplot for Price-Distribution across Different Brands

Price significantly differs for the followings (p value < 0.05) –

1. K-Beauty Brands
2. Products
3. Product Specialities

- Skin Types, Vegan Label, Product Formulation do not affect the K-Beauty product price.


Boxplot for Price-Distribution across Different Products


Boxplot for Price-Distribution across Different Specific Category

SVM MODEL

i. Transforming all the categorical data into factors using encoding methods.

ii. Split the dataset into Train and Test dataset - 80% for Training and 20% as Testing.

```
set.seed(123)

> split <- sample.split(data, SplitRatio = 0.8)

> data_train <- subset(data, split == "TRUE")

> data_test <- subset(data, split == "FALSE")
```

iii. Fit the model to the Train data. For non-linear data, SVM model with RBF Kernel effectively maps data into a higher dimensional space where complex relationships between data points can be linearly separated.

```
> model <- svm(data.cs.Price.1 ~.,data= data_train,+ kernel='radial')

> summary(model)

Call:

svm(formula = data.cs.Price.1 ~ ., data = data_train, kernel = "radial")

Parameters:

SVM-Type:  eps-regression

SVM-Kernel:  radial

cost:  1

gamma:  0.01754386

epsilon:  0.1

Number of Support Vectors:  344
```

iv. Predict on Test data.

```
preds <- predict(model,data_test)
```

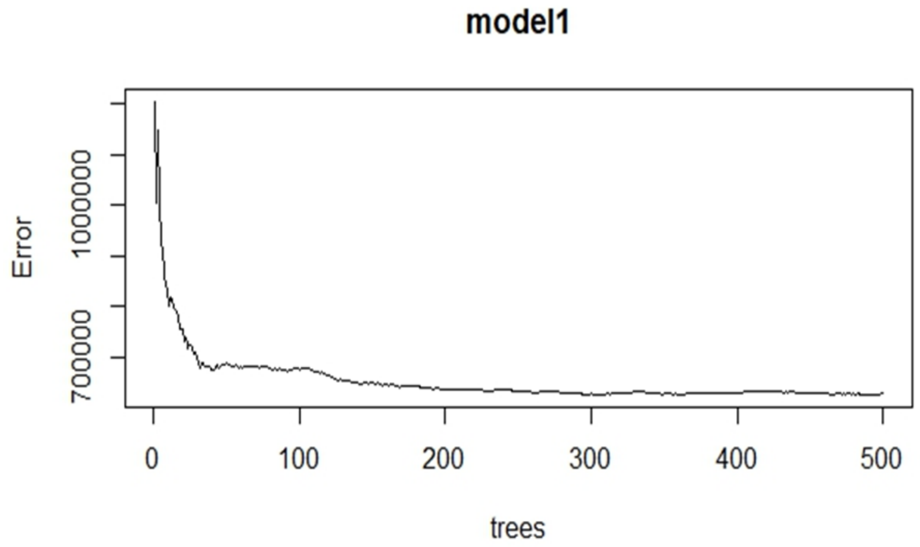
RANDOM FOREST

i. Transforming all the categorical data into factors using encoding methods.

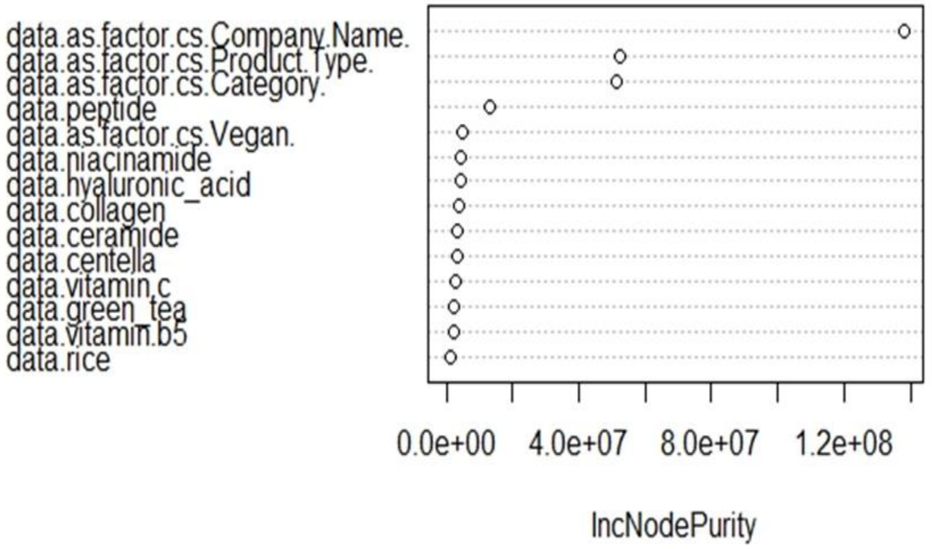
ii. Split the dataset into Train and Test dataset - 80% for Training and 20% as Testing as SVM.

iii. After this we fit a Random Forest used for regression on Train data.

```
> library(randomForest)

> model1 <- randomForest(data.cs.Price.1 ~.,data= data_train)

Call:

randomForest(formula = data.cs.Price.1 ~ ., data = data_train)

               Type of random forest: regression

                     Number of trees: 500

No. of variables tried at each split: 4

          Mean of squared residuals: 627359.9

                    % Var explained: 38.06
```



iii. Check error stabilizer rate and "varImp" to find importance of all regressors in Random Forest model. Error rate is stabilized with increasing number of trees. Brand Name is the most important feature followed by Product Type, Category etc.



iv. Finally, predict on Test data.

```
pred_test <- predict(model1, newdata = data_test)
```

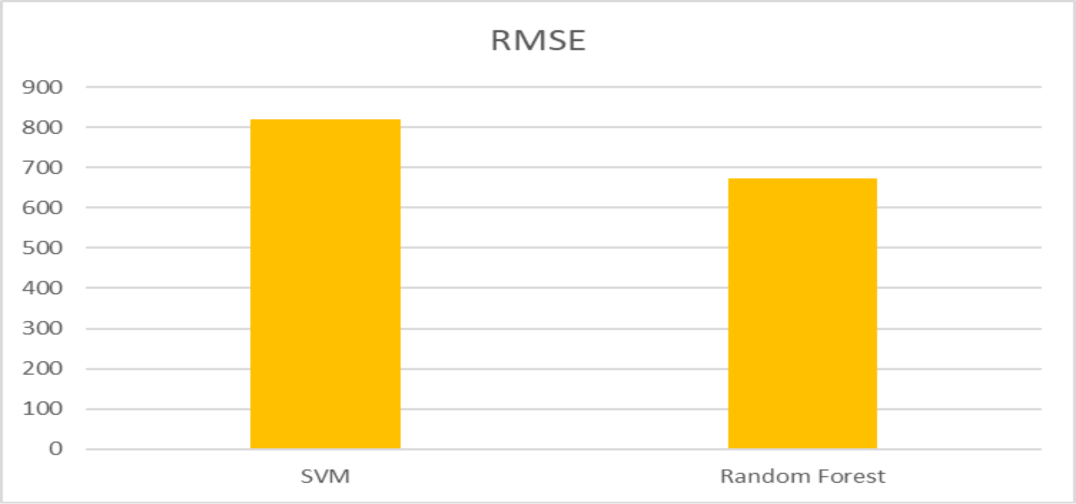
RMSE : SVM vs. RANDOM FOREST

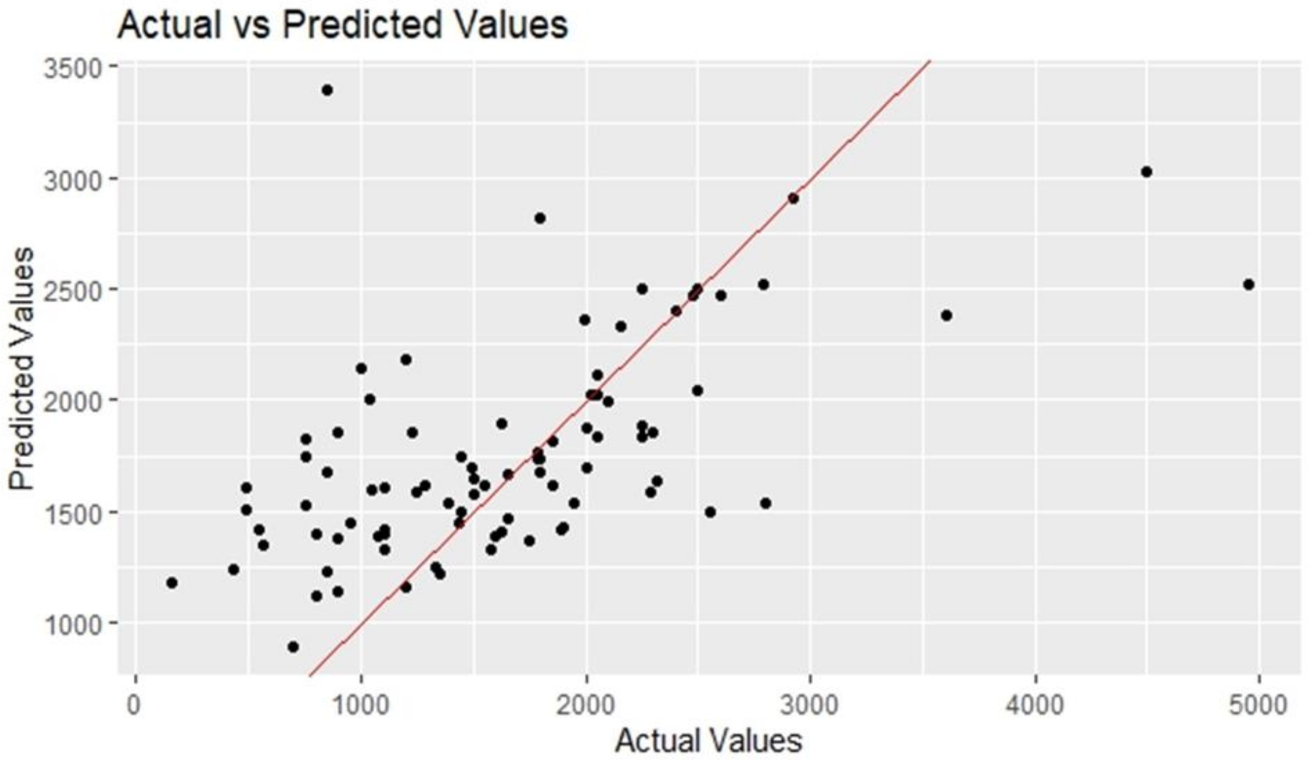
- Root Mean Squared Error - measures the average difference between values predicted by a model and the actual values. It provides an estimation of how well the model is able to predict the target value (accuracy).

```
> sqrt(mean((preds - data_test$data.cs.Price.1)^2))
[1] 819.317
> sqrt(mean((pred_test - data_test$data.cs.Price.1)^2))
[1] 672.1571
```

- Random Forest is better than SVM as RMSE is les ser.
- ggplot - visualize actual vs. predicted.

```
> plot_data <- data.frame(Actual = data_test$data.cs.Price.1,
                          Predicted = pred_test)
> ggplot(plot_data, aes(x = Actual, y = Predicted)) + geom_point() +
   geom_abline(intercept = 0, slope = 1, color = "red") +
     labs(title = "Actual vs Predicted Values",
     x = "Actual Values", y = "Predicted Values")
```


RMSE


Actual vs Predicted Values

THANK YOU