Project – 2 Phase-1

Group: Smiles

Objective:

The objective of this project is to use statistical learning models to predict how often individuals wear a mask around people who do not live in their home using survey data.

Purpose:

The goal of this project is to use statistical learning models to predict mask-wearing frequency among individuals living alone, based on survey data. Amid the COVID-19 pandemic, mask-wearing is crucial to prevent the virus spread, but adherence varies across locations. This project seeks to examine the factors affecting mask-wearing among A&M students and create a predictive model to understand mask-wearing frequency.

Problem:

The COVID-19 outbreak has prompted widespread mask-wearing as a preventive measure, but adherence varies and its efficacy is debated. This project aims to predict mask-wearing frequency among individuals living alone and examine the factors influencing A&M students' mask-wearing behaviour to develop a predictive model for understanding mask-wearing frequency.

This Report Contains:

- Technical report of model 1.
- Technical report of model 2.
- Technical report of model 3.
- Comparison of the 3 models and specifying the best model you want to use for testing.

Group Members:

1. Lakshmi Sai Deepthi Vanasarla
2. Siddharth Thazhathedathu
3. Mahati Aditya Pisipati
4. Satya Prakash Kodamanchili

Introduction:

The first step is to examine the data and interpret it as much as possible. We applied the dataset to several models to find out which approach worked best. Having reviewed a variety of models, including linear regression and complex trees, and comparing the performance of each model, we have selected three that perform best. As a result, three top models will be evaluated and tuned further to optimize their performance: Forward stepwise selection, LDA and bagging.

For effective model development and selection, we have started by handling the raw data correctly, focusing on the missing values. The training data was first thoroughly checked and preprocessed to ensure a solid foundation for model building. In the first step, we inspected the dataset's basic statistics, dimensions, and column data types. The numeric columns were imputed using median for every parameter to deal with missing values. To ensure the data was suitable for modeling, we transformed all columns into integers. A new data frame 'traindata_omitna' was created after imputation, containing the processed data, and we examined the types of columns and statistics for essential variables. To make future analyses easier, we recorded the dimensions of the final dataset and the number of rows and predictor columns to check if we are going ahead in the right direction or not.

Model – 1: Forward Stepwise Selection

To identify the best subset of predictors for our target variable, we choose forward stepwise selection. With 154 predictors (nvmax = 154), the forward selection algorithm is implemented using the 'traindata_omitna' dataset. The model fits are then summarized, and diagnostic plots are created for evaluation metrics. So, the diagnostic plots below are displaying the relationship between number of variables and the Residual Sum of Squares (RSS), Adjusted R-squared (Adj-R2), Mallows' Cp (Cp), and Bayesian Information Criterion (BIC) respectively.

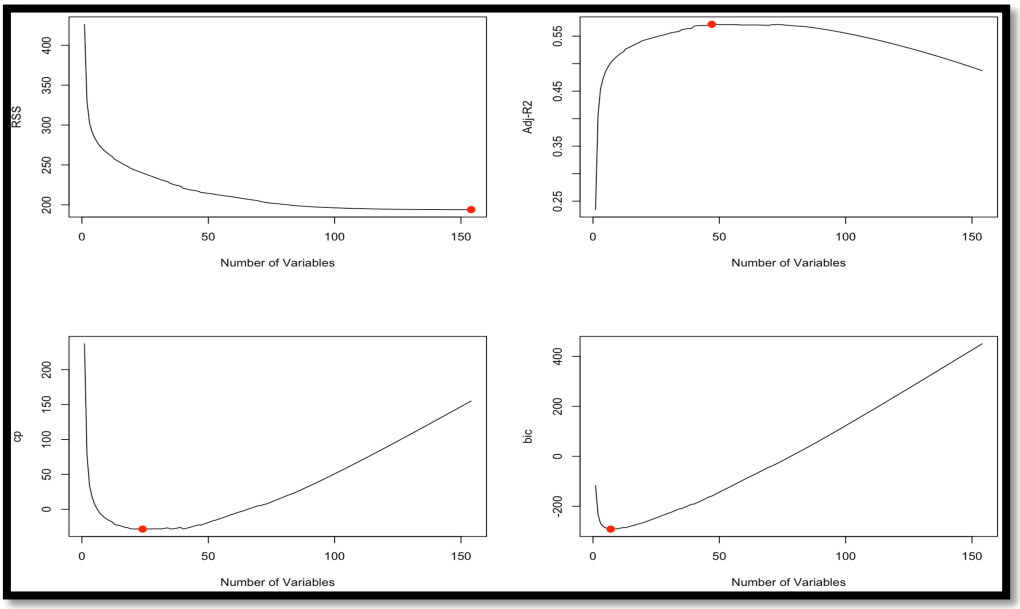

Figure 1: Diagnostic plots of RSS, Adj-R2, Cp, BIC with variables 154,47,24 and 7 respectively.

```
> coef(fss_fits, 7)
(Intercept)       Q5_1       Q25_6       Q29_4         Q38         Q42        Q100
 0.95805933  0.06992544 -0.12801009  0.09655461 -0.26248923  0.21236563  0.48044899
      Q66_3
 0.10086736
```

Figure 2: The coefficients of the optimal model, containing 7 predictors, identified by the lowest BIC.

Looking at the above plots, we find the models which are giving us lowest RSS, Cp and BIC values and highest Adj-R^2 (Indicated in red in Figure 1). From the analysis, it has become evident that the model with lowest BIC criterion value is most appropriate to move ahead. The BIC metric helps us identify

an optimal model, which we will further analyze and optimize to improve our model's prediction accuracy, by using forward stepwise selection.

Next, we perform a k-fold cross validation analysis for our approach. We intend to proceed with 10-fold approach by giving the 'k' value as 10. To identify the optimal model, we average the cross-validation errors across all folds and plot them against the number of predictors. Considering the lowest cross-validation error, we can determine that the forward stepwise selection model with seven predictors is the best fit. In addition, this analysis demonstrates the effectiveness of cross-validation in evaluating model performance and selecting the most suitable model based on the data.

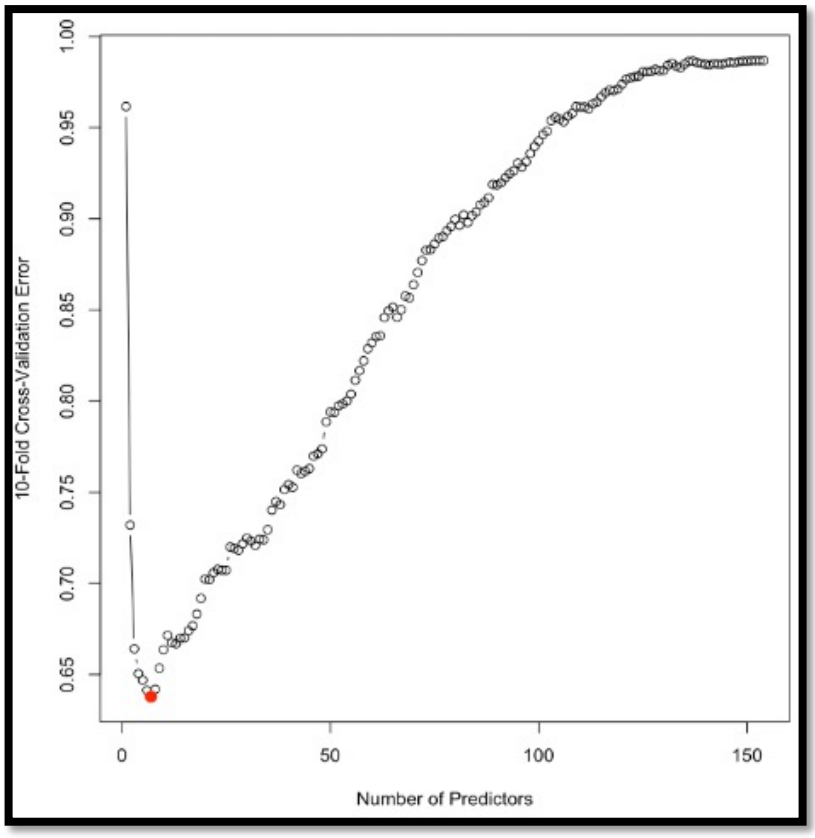

Figure 3: The 10-fold cross-validation error plotted against the number of predictors

According to Figure 3, as the number of predictors increases, the cross-validation error initially decreases, reaching a minimum value, and then increases as more predictors are added. This plot helps us to determine what the appropriate number of predictors should be in the forward stepwise selection model (indicated in red in Figure 3). In this case, we observe that the model with 7 predictors has the lowest cross-validation error, which indicates that it is the most generalizable.

```
> min(mean.cv.errors)
[1] 0.637861
> points(7, mean.cv.errors[7], col ="red",cex =2, pch =20)
> coef(fss_fits, 7)
(Intercept)        Q5_1       Q25_6       Q29_4         Q38         Q42        Q100        Q66_3
 0.95805933  0.06992544 -0.12801009  0.09655461 -0.26248923  0.21236563  0.48044899  0.10086736
```

Figure 4: Optimal forward step-wise selection model with 7 predictors, showing the lowest mean cross-validation error

The cross-validation error value of 0.6378 represents the minimal mean error across all models tested using the k-fold cross-validation process. A model's predicted values are calculated by averaging the squared differences between the actual target values (traindata) for each predictor. When compared to other models with different numbers of predictors, the model with 7 predictors provides the best data with the lowest mean cross-validation error.

Model – 2: Linear Discriminant Analysis

We have chosen LDA as one of the models because it is classification modelling and we compared it to QDA (Quadratic Discriminant Analysis) we have found LDA has a lower error rate. LDA is a linear model that tries to find the optimal linear combination of features that maximizes the separation between different classes. The goals are to find a hyperplane that separates the classes with greatest margin. LDA uses Bayes' theorem.

Analysis I:

There are 154 variables and 479 observations (data points). In the beginning we took all the variables into account and fitted the LDA model for the observations. After train-test split of 75% training data we have found results such as:

We can see that there is a misclassification rate of 64% if we take the all the variables and an accuracy of 36%. We have converted the values into the integer and calculated the test MSE for comparison. We have got an MSE of 1.558.

```
> ##### LDA
> lda.fit = lda(Q46_2~., data = train)
> lda.pred = predict(lda.fit, test)
```

Figure 5: LDA model

```
   1  2  3  4  5
1  1  0  5  0  1
2  3  6  5  4  1
3  3  6 15 10  2
4  0  2  7 14  6
5  0  3  5 14  7
```

Figure 6: LDA confusion matrix

```
> mean(lda.pred$class != test$Q46_2)
[1] 0.6416667
> mean(((as.integer(as.vector(lda.pred$class))) - test$Q46_2)^2)
[1] 1.558333
```

Figure 7: Test mean error

Analysis II - Improved Approach:

Due to the high misclassification rate and MSE, one of the approaches to reduce the error rate is using only the most important variables and that in turn could reduce the misclassification. To find the most important variables, we have taken a few different approaches to solve this issue. One of the best methods to find those variables is to use Shrinkage Methods. We have tried both ridge and lasso to find the most important variables. Lasso Regression due to its L1 norm as the penalty term and it can completely reduce the coefficients to give a more defined set of variables compared to ridge which has L2 norm in its penalty term. and we have tried this approach with λ=0.05126 (found using cross validation). We found 21 important variables which drastically reduced the misclassification rate and gave us above 49% misclassification rate and a test MSE of 0.9583.

We tried to find other algorithms that would reduce the number variables by using selection algorithms as well to find the most important variables. After analysis we have found that using Forward Selection and Cross-validation gave us 7 variables. We have used these variables for the LDA model which further reduced our misclassification rate.

```
> lda.fit = lda(Q46_2~Q5_1+Q25_6+Q29_4+Q38+Q42+Q100+Q66_3, data = train)
> mean(lda.pred$class != test$Q46_2)
[1] 0.4333333
> mean(((as.integer(as.vector(lda.pred$class))) - test$Q46_2)^2)
[1] 0.6333333
```

Figure 8: Improved approach error

Confusion Matrix:

The LDA output indicates $\hat{\pi}1 = 0.0445$, $\hat{\pi}2 = 0.09749$, $\hat{\pi}3 = 0.28133$, $\hat{\pi}4 = 0.3481$ and $\hat{\pi}5 = 0.22841$; in other words 4.45% belong to class 1, 9.74% belong to class 2, 28.133% belong to class 3, 34.81% belong to class 4 and 22.841% belong to class 5 calculated among the training data set. It gives us a group means. This is used by LDA as estimate of μk.

```
   1  2  3  4  5
1  3  1  0  0  0
2  1  5  2  1  0
3  3  9 24  7  2
4  0  2 11 31 10
5  0  0  0  3  5
```

```
> lda.fit
Call:
lda(Q46_2 ~ Q5_1 + Q25_6 + Q29_4 + Q38 + Q42 + Q100 + Q66_3,
    data = train)

Prior probabilities of groups:
          1           2           3           4           5
0.04456825 0.09749304 0.28133705 0.34818942 0.22841226

Group means:
      Q5_1    Q25_6    Q29_4      Q38      Q42     Q100    Q66_3
1 4.750000 2.750000 3.375000 3.062500 1.812500 2.125000 5.187500
2 5.085714 2.342857 3.771429 2.628571 1.800000 2.314286 5.314286
3 5.653465 2.128713 4.297030 2.425743 1.900990 2.970297 5.742574
4 7.056000 1.592000 5.480000 1.984000 1.600000 3.384000 6.312000
5 7.829268 1.390244 6.243902 1.585366 1.365854 3.658537 6.682927

Coefficients of linear discriminants:
             LD1         LD2         LD3         LD4
Q5_1  -0.1075021  0.15914574 -0.15672053  0.19640283
Q25_6  0.3150363  0.43429481 -0.63870468 -0.92865447
Q29_4 -0.1719956  0.30034437 -0.01882913 -0.07838902
Q38    0.4933542  0.76966448 -0.96470294  0.76182493
Q42   -0.4805990 -1.26953410  0.13151830  0.05386083
Q100  -0.9282298 -0.30392212 -0.43104519 -0.09010144
Q66_3 -0.1734540 -0.07124126 -0.50921109 -0.28100047

Proportion of trace:
   LD1    LD2    LD3    LD4
0.9560 0.0317 0.0078 0.0045
```

Figure 9: Summary of the LDA improved fit

Key insights from summary:

- The variable Q100 has the most influence on LD1 when we are separating the classes because the absolute coefficient value is -0.92822
- The variable Q42 has the most influence on LD2 when we are separating the classes because the absolute the coefficient value is -1.269953
- The variable Q38 has the most influence on LD3 when we are separating the classes because the absolute the coefficient value is -0.96470
- The variable Q25_6 has the most influence on LD4 when we are separating the classes because the absolute the coefficient value is -0.9286544

The proportion of traces shows that:

- LD1 explains 95.6% of the total variance between the classes.
- LD2 explains 3.17% of the total variance between the classes.
- LD3 explains 0.78% of the total variance between the classes.
- LD4 explains 0.45% of the total variance between the classes.

Model – 3: Bagging:

A study was conducted to compare the performance of bagging and random forest regression methods using the dataset. Initially, 75% of the data was randomly selected as the training set, and the remaining 25% was used for testing. Both random forest models were trained using the training set and the mean squared error (MSE) was evaluated on the test set.

To determine the optimal value of the m parameter in random forest, which specifies the number of variables randomly sampled as candidates at each split, the models were trained with different values of m ranging from 1 to 154. The test MSE was evaluated for each value of m, and a plot was created to visualize the relationship between m and the test MSE.

The plot shows that the test MSE decreased as m increased. The lowest test MSE was obtained when m was set to 154, indicating that the bagging method outperformed the random forest method in this dataset.

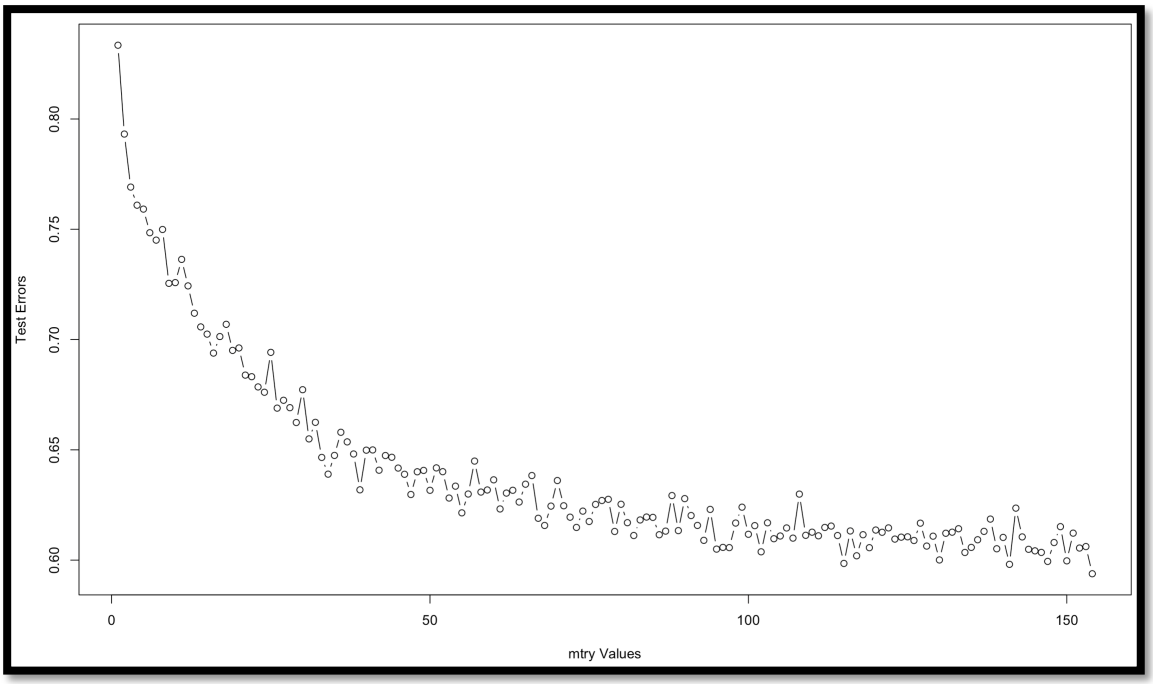

Figure 10: m values vs. Test MSE

In the same way, another study was conducted to evaluate the impact of the number of trees on the test mean squared error (MSE) in the model. The experiment involved incrementally increasing the value of ntree, representing the number of trees to be grown, from 5 to 500 while maintaining m, representing the number of variables randomly sampled at each split, at a constant value of 154. The results indicated that the test MSE reached a saturation point when the number of trees reached 100, as shown in the plot below. Based on these findings, the m value was set at 154 and the maximum number of trees was fixed at 100 for further analysis.

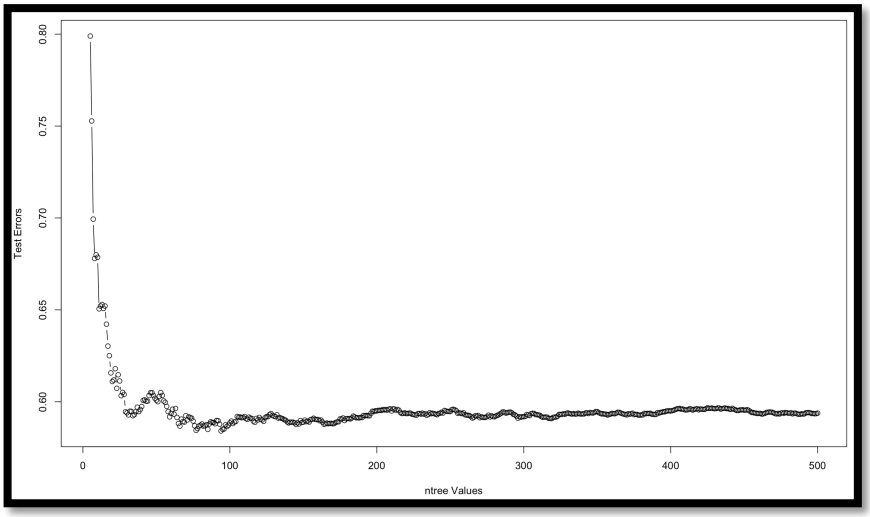

Figure 11: ntree values vs. Test Errors

Once the m and ntree values are found out, they are used in the building of bagging trees iteratively by bootstrap sampling the training data. The output variable Q46_2 is predicted using this model and the test MSE is calculated, which turned out to be 0.588 which is considerably less than Forward selection and LDA models' test error.

Because the bagging model reduces the variance by estimating the mean of many trees, it is less interpretable now. So, Variable importance plot is presented below.

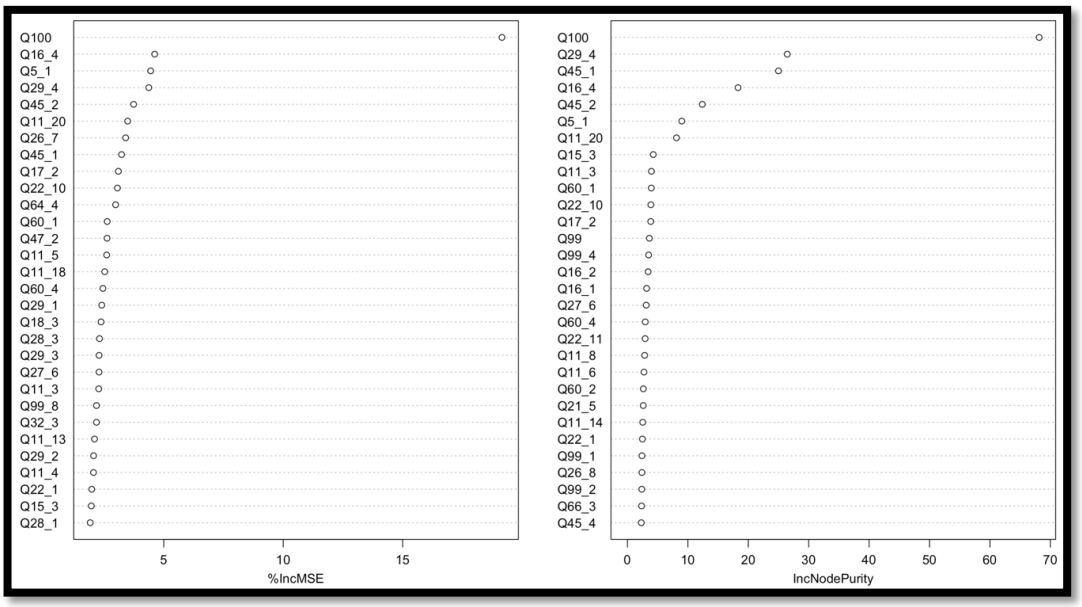

Figure 12: Variable Importance Plots

The above plot is the plot between each predictor variable and their %IncMSE, IncNodePurity numbers. The first measure, based on the mean decrease of accuracy in predictions on the out of bag samples when a given variable is excluded from the model, provides an estimate of the impact of each variable on the model's predictive accuracy. This measure is useful when the goal is to maximize the accuracy of the model's predictions. The second measure, based on the total decrease in node impurity that results from splits over that variable, provides an estimate of the contribution of each variable to the overall structure of the tree. This measure is useful when the goal is to understand the underlying relationships between the variables and the target variable. As our goal is to improve the model accuracy, the first measure explains which predictors are the most important. Of all the predictors, Q100 is the most significant one followed by Q16_4, Q5_1, Q29_4, Q45_2 and so on.

Comparison of Models:

We have used three models: Forward Selection, Bagging, and LDA. Forward Selection and Bagging are regression models we cannot directly compare to a classification model such as LDA. In Regression models, we mainly use the Test Mean Squared Errors to compare the models between them and for classification we calculate the misclassification rate (We have still provided test MSE for LDA). So, to compare we have used a heatmap to assess the predicted values.

Model Name	Test MSE	Misclassification Error	Adj. R^2
Forward	0.637	-	0.5017
LDA	0.63	0.43	
Bagging	0.588	-	

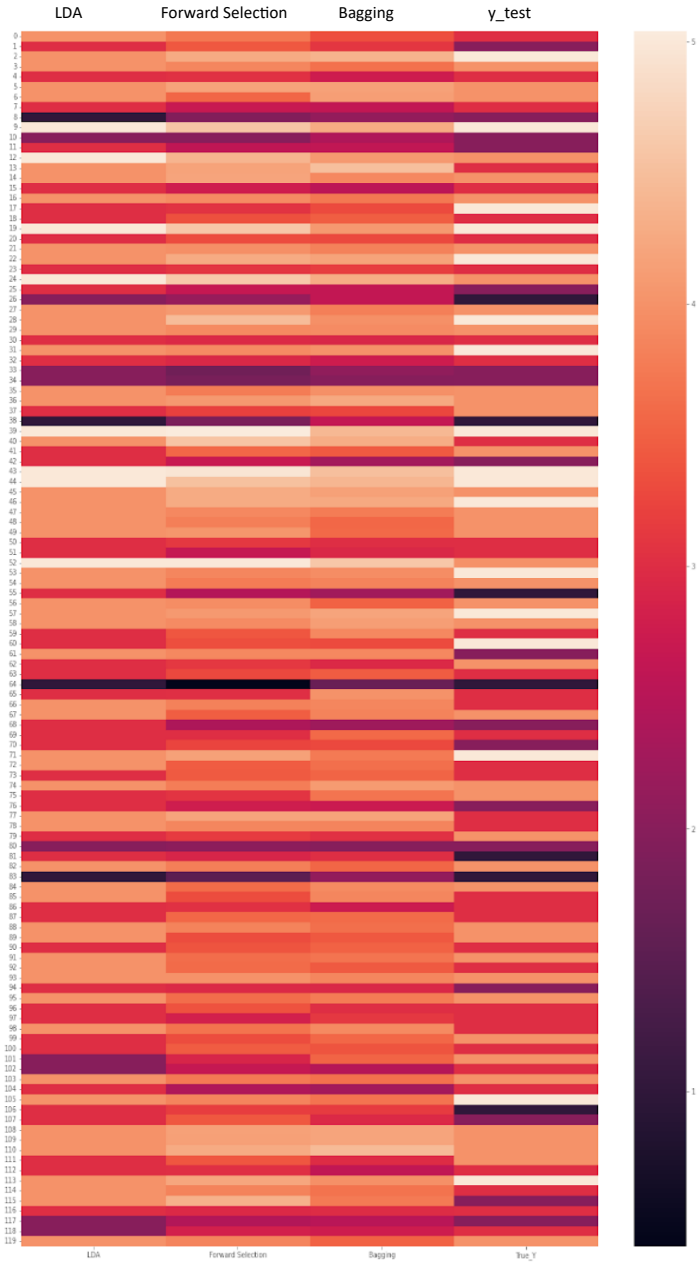

Figure 13: Heatmap- Assessing Predicted Variables

When comparing models using Test MSE as a metric, Bagging has the lowest score of 0.588, while LDA and Forward Selection follow closely with Test MSE scores of 0.63 and 0.637 respectively. It's worth noting that LDA is a classification model, so we can measure its performance in terms of misclassification rate, which in this case is 43%. Forward Selection, on the other hand, provides an Adj. R2 score.

The Heatmap shows the predicted values for different models, which we can compare to the true test data response variable (y_test).

However, the Heatmap ranges from 0 to 5, which means that none of the true response values in y_test can be below 1 since y_test ranges from 1 to 5. This constraint does not apply to the regression models, which can predict values ranging from 0 to 5 and can also produce float values. However, this does not apply to LDA, as it only predicts the class.

In conclusion, we have chosen Bagging model as the final model.