PROJECT 2: IDENTICATION OF PATIENT WITH ACUTE LYMPHOCYTIC LEUKEMIA BASED ON THEIR GENES

1. EXECUTIVE SUMMARY

In 2015, over two million people had leukemia, and leukemia caused over 350,000 deaths. Acute forms of leukemia are faster progressing and typically more aggressive than chronic forms, and if left untreated, could be fatal in a few months, so early diagnosis is especially important. Currently, cytochemistry and lineage phenotypes are usually used in combination to identify which type of leukemia the patient has. Both methods are subject to certain limitations and even when used in combination, may not yield a reliable diagnosis. A correct diagnosis is key to determine a patient's likely response to certain treatments. In this project, we are thus interested in increasing the diagnostic accuracy by exploring the possibility of diagnosing/identifying leukemia based on the patient's genes. The provided Chiaretti dataset which consist of 12625 explanatory variables (patient's genes) and 128 observations was analyzed using 3 classification methods which are: Lasso logistic regression, Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM) methods in R. After carrying out the analysis using these methods, we discovered that both Lasso and SVM provided the best model that helped to predict patients with leukemia, and also helped to identify the genes that are responsible for influencing the possibility of having leukemia. We selected the top 10 genes that help to predict patients with leukemia for each method. From our result, both LASSO and LDA performed the best in analyzing these datasets, they also provided the best predictions in terms of identifying patients with leukemia with model accuracy of 92% and Area Under the Curve (AUC) of 0.9 which are our criteria for selecting the best performing method. While SVM is the least performing method with model accuracy of 88% and AUC of 0.85.

2. INTRODUCTION

Leukemia is a group of cancers that affect blood cells. The most common form of leukemia in children is ALL, and the most common form of acute leukemia in adults is AML.
ALL is a type of cancer that develops in immature lymphocytes, a type of white blood cell. There are two main types of lymphocytes: B cells and T cells. B cells can mature into plasma cells, which produce antibodies or become memory B cells, which are important to secondary immune responses. T cells have a diverse set of functions including directly killing virus-infected cells.
Our project aims to diagnose whether a patient has acute lymphoblastic leukemia (ALL) based on their genes using the Chiaretti dataset.

3. EXPLORATORY DATA ANALYSIS

The Chiaretti et al. 2004 dataset has 128 observations along with the expression levels of 12,625 genes assayed using HGU95aV2 gene chips. We will try to classify patients into two subcategories: Patients with ALL (Either ALL affecting B cells or ALL affecting T cells) and Patients without ALL (Those that tested negative). The following bar plot helps to visualize number of patients with different types of ALL and those without ALL.

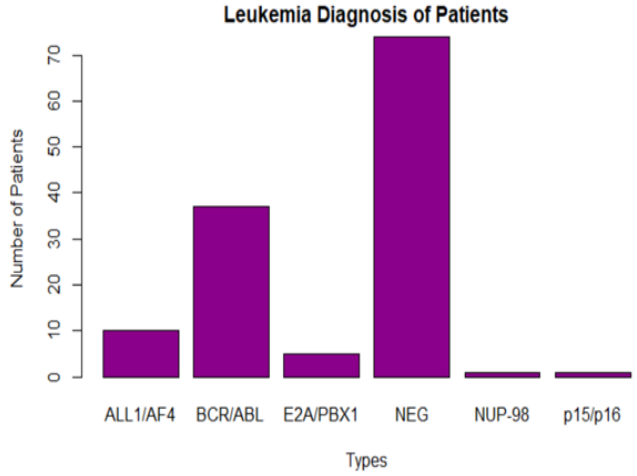

Figure 1: Leukemia Diagnosis of the patients

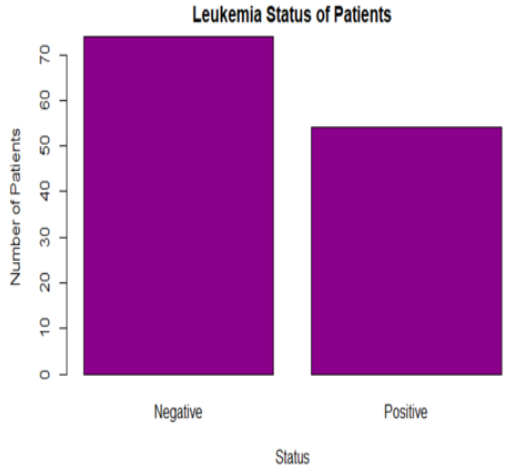

Figure 2: Patients with & without Leukemia

We visualized some predictors using histogram to see how they are distributed.

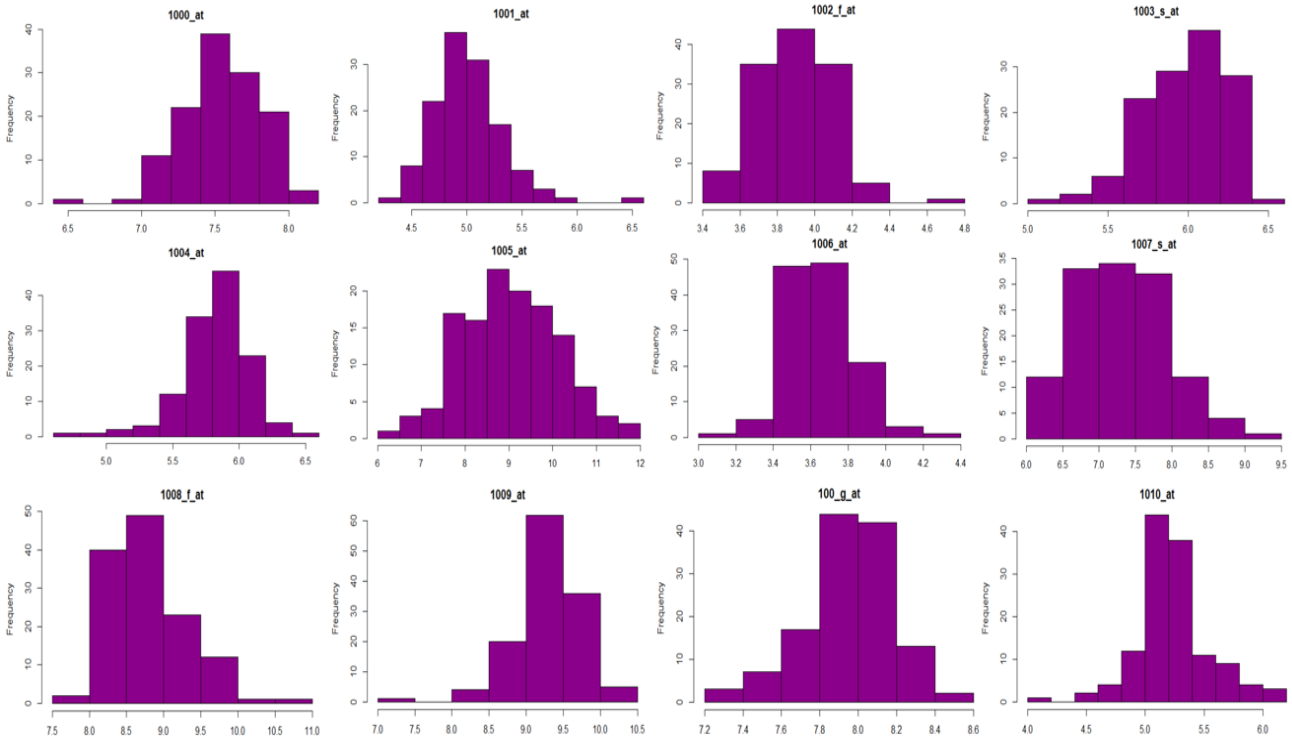

Figure 3: Histogram of some Predictors

4. METHOD

This regression analysis was done using three different classification methods which are Lasso Logistic Regression, LDA, Support Vector Machine (SVM). We then compared these methods using Receiver Operating Characteristics (ROC), Area Under Curve (AUC) and the model accuracy so as to determine the best performing classifier (method) that helped to predict patients with Leukemia.

The response variable we are trying to predict states the type of Leukemia the observed patients have, it also states if the patient is negative (i.e patient without Leukemia), this variable consists of 6 levels: ALL1/AF4, BCR/ABL, E2A/PBX1, NEG, NUP-98, p15/p16 which are recoded into 2 dummy variables: 0s and 1s.

$$Y_i = \begin{cases} 0 & patient\ wtihout\ Leukemia \\ 1 & patient\ with\ Leutkmia \end{cases}$$

The dataset was divided into training data (80% of the total observations) and test data (20% of total observations). The models (Lasso, LDA, SVM) were fitted on the training data and predictions were made on the test data. We fitted a Lasso logistic regression model on the training data using a 10 folds cross validation to pick the best lambda value that minimizes the cross-validation error and gives the best model, we then make predictions

on the test data and check for model accuracy which gives us 92%. Lasso method shrinks the coefficients of less important explanatory variables to zero and select the most important ones. Lasso selected only 39 out of the 12625 explanatory variables (genes) present in the dataset.

We also carried out LDA and SVM regression on the training data and make prediction on the test data for each method, the model accuracy was also checked for both LDA & SVM which are 92% and 88% respectively.

Finally, we compared the 3 classifiers using AUC, ROC and the model accuracy to check for the best performing classifier/method. Criteria for selecting the best method based on AUC, ROC and model accuracy are listed below:

SELECTION CRITERIA	
AUC	A higher AUC or AUC closer to 1 The higher the better
ROC	A higher ROC or ROC closer to 1 The higher the better
Model Accuracy	Accuracy closer to 100% The higher the better

Figure 4: Criteria for selecting the best performing Classifier/Method

5. RESULTS

After carrying out the regression analysis on the dataset using Lasso, LDA and SVM, the following are the results we derived:

- LASSO LOGISTIC REGRESSION RESULT

For Lasso, we applied 10-fold cross validation to determine the best tuning parameter, λ = 0.03197895 that minimizes the cross-validation error and produces the best model

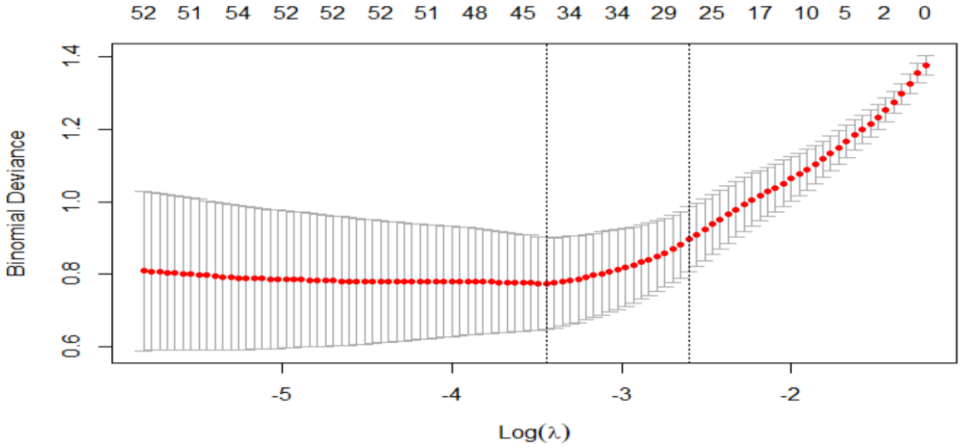

Figure 5: Binomial Deviance vs Lambda values

Lasso logistic regression shrunk the coefficients of 12586 (99.69% of all predictors) explanatory variables to 0 selecting only 39 variables which influence the possibility of a patient being diagnosed of leukemia. We will select the top 10 variables (genes) with high coefficients out of the 39 selected variables that greatly influence the possibility of having leukemia. These top 10 variables/genes are visualized in the bar plot below:

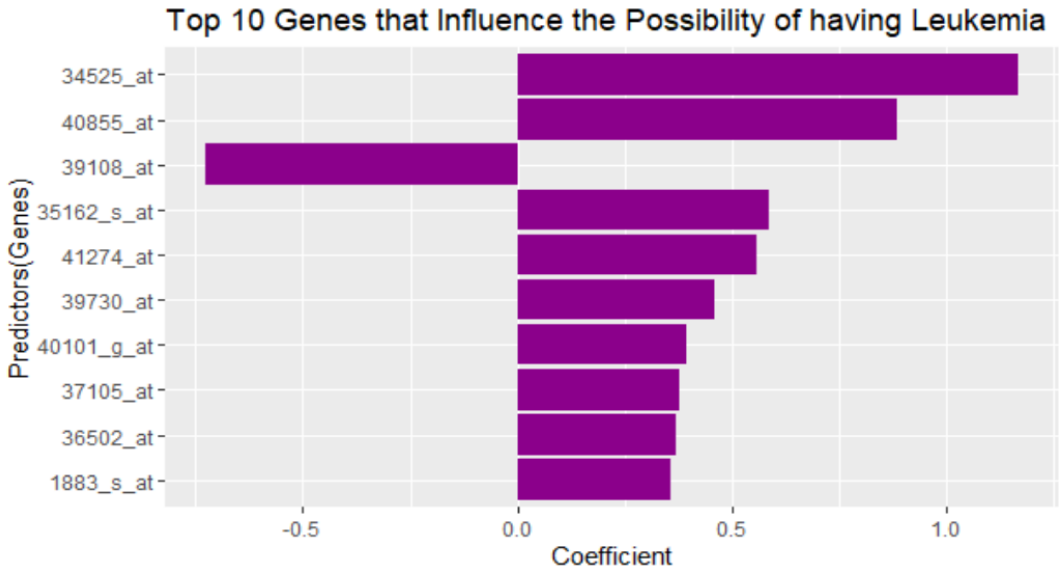

Figure 5: Top 10 Genes that influence the possibility of having Leukemia selected by Lasso

After fitting the model on the training data, we also make some predictions on the test data and check the model accuracy which is 92%.

- LINEAR DISCRIMINANT ANALYSIS (LDA) RESULT

Unlike Lasso regression, LDA model does not shrink coefficients of less significant explanatory variables to 0. Hence, to select the best performing variables/genes that influence the probability of having leukemia, we used the coefficients of Linear discriminants, and we selected the predictors with the highest coefficient. After fitting an LDA model on the training data, we selected the top 10 predictors/genes with the highest coefficient which is visualized in the following bar plot:

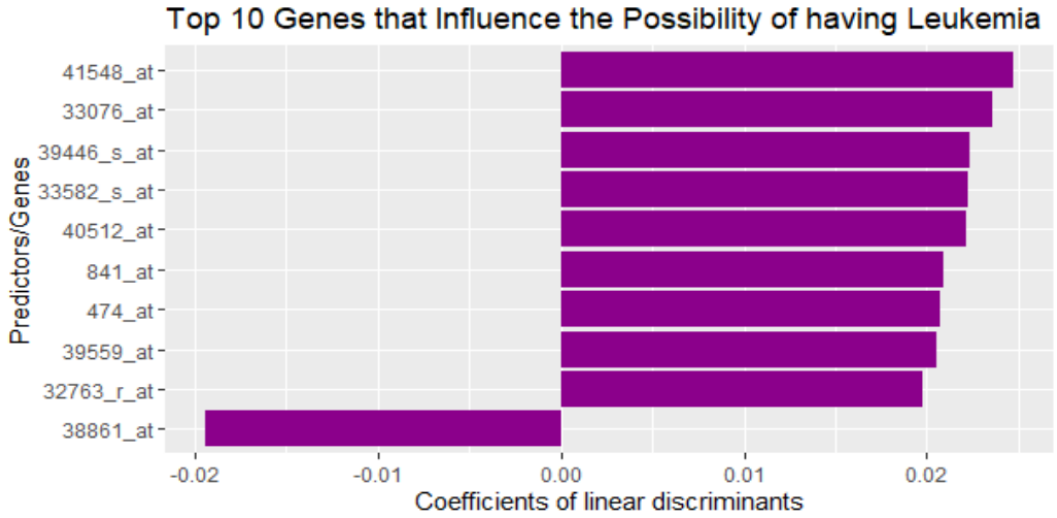

Figure 5: Top 10 Genes that influence the possibility of having Leukemia selected by LDA

We also used the fitted LDA model to make prediction on the test data and the model accuracy is also 92% just like that of lasso.

- SUPPORT VECTOR MACHINE (SVM) RESULT

Similar to LDA, SVM model also does not shrink coefficients of less significant explanatory variables to 0. Therefore, we carried out variable selection by selecting the top 10 predictors/genes with the highest coefficient reason being that predictors with high coefficients tend to have significant influence on the response variable. The top 10 genes selected is displayed below.

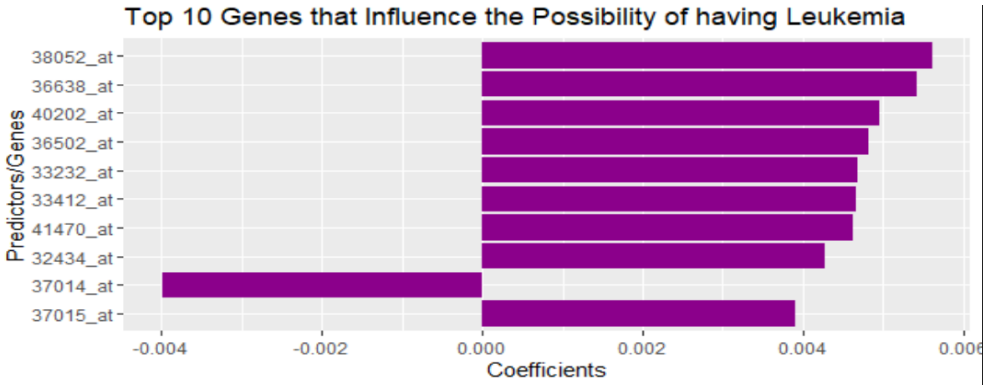

Figure 6: Top 10 Genes that influence the possibility of having Leukemia selected by SVM

The fitted SVM model was used to make prediction on the test data and the model accuracy is 88%.

- COMPARISON OF THE 3 METHODS

The three methods are quite straight forward and easy to use in R and they all performed very well on the Chiaretti dataset. Nevertheless, we would like to know the best performing classifier/method that best predict the possibility of a patient having Leukemia. In doing this, we compared the 3 classifiers using AUC, ROC, and the model accuracy to check for the best performing method following the criteria listed in figure 4 from "Method" chapter. We calculated the AUC, model accuracy and plotted the ROC of the 3 methods used.

	LASSO	**LDA**	SVM
AUC	**0.9**	**0.9**	0.85
Model Accuracy	**92%**	**92%**	88%

Figure 8: Table showing the results of AUC & Model Accuracy of the 3 methods

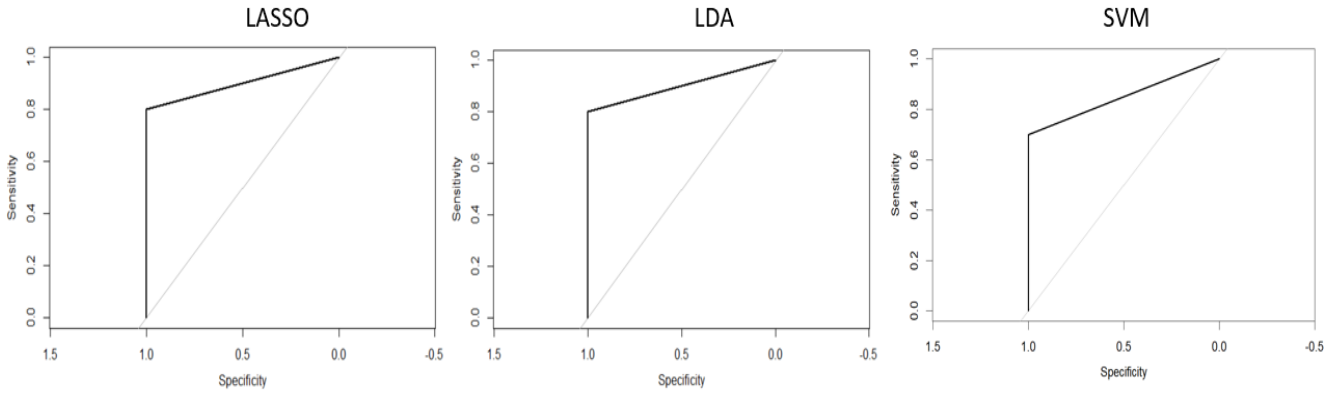

Figure 9: ROC plot of the 3 methods

6. CONCLUSION

From the results of the 3 methods given in the previous chapter, we can see that both LASSO and LDA have the highest AUC which is 0.9 which is closer to 1 while SVM has the least AUC which is 0.85. Also, after making prediction on the test data using each model, we discovered that the model accuracy of LASSO and LDA are the highest which is 92% very close to 100% accuracy while that of SVM is the least which is 88%. Clearly from the criteria we used, LASSO and LDA are both the best performing methods, they performed excellently on the dataset and give the best prediction in identifying patients with Leukemia (both B-ALL and T-ALL). While SVM is the least performing method.

APPENDIX

The R-code used in this project is given below:

```
#Load required R packages
library("devtools")
library("tidyverse")
library("datamicroarray")
library("caret")
library("glmnet")
library("ggplot2")
library("e1071")
library("pROC")
library("MASS")

#Step 1: load the chiaretti dataset
data('chiaretti', package = 'datamicroarray')

#Step 2: run exploratory data analysis on dataset
#barplot showing number of patients with different types of ALL
barplot(table(chiaretti$y),main="Leukemia Diagnosis of Patients",xlab = "Types",
ylab="Number of Patients",col ="darkmagenta")
#barplot showing number of patients with & without ALL
barplot(table(chiaretti$y.recode),names.arg =c("Negative","Positive"),main="Leukemia
Status of Patients",xlab = "Status",
ylab="Number of Patients",col ="darkmagenta")
#histogram showing the distribution of some predictors
hist(chiaretti$x[,1],main = colnames(chiaretti$x)[1],xlab = colnames(chiaretti$x)[1],col
="darkmagenta")
hist(chiaretti$x[,2],main = colnames(chiaretti$x)[2],xlab = colnames(chiaretti$x)[2],col
="darkmagenta")
hist(chiaretti$x[,3],main = colnames(chiaretti$x)[3],xlab = colnames(chiaretti$x)[3],col
="darkmagenta")
hist(chiaretti$x[,4],main = colnames(chiaretti$x)[4],xlab = colnames(chiaretti$x)[4],col
="darkmagenta")
hist(chiaretti$x[,5],main = colnames(chiaretti$x)[5],xlab = colnames(chiaretti$x)[5],col
="darkmagenta")
hist(chiaretti$x[,6],main = colnames(chiaretti$x)[6],xlab = colnames(chiaretti$x)[6],col
="darkmagenta")
hist(chiaretti$x[,7],main = colnames(chiaretti$x)[7],xlab = colnames(chiaretti$x)[7],col
="darkmagenta")
```

```
hist(chiaretti$x[,8],main = colnames(chiaretti$x)[8],xlab = colnames(chiaretti$x)[8],col ="darkmagenta")
hist(chiaretti$x[,9],main = colnames(chiaretti$x)[9],xlab = colnames(chiaretti$x)[9],col ="darkmagenta")
hist(chiaretti$x[,10],main = colnames(chiaretti$x)[10],xlab = colnames(chiaretti$x)[10],col ="darkmagenta")
hist(chiaretti$x[,11],main = colnames(chiaretti$x)[11],xlab = colnames(chiaretti$x)[11],col ="darkmagenta")
hist(chiaretti$x[,12],main = colnames(chiaretti$x)[12],xlab = colnames(chiaretti$x)[12],col ="darkmagenta")

#Step 3: Split the data into training and test set
#recode y variable into dummy variables 0 & 1
chiaretti$y.recode <- ifelse(chiaretti$y == "NEG", 0, 1)
#chiaretti.df <- data.frame(y=chiaretti$y.recode,chiaretti$x)
set.seed(123)
training.samples <- chiaretti$y.recode %>% createDataPartition(p = 0.8, list = FALSE)
#create x and y training and test data
x.train.data  <- chiaretti$x[training.samples, ]
y.train.data <- chiaretti$y.recode[training.samples]
x.test.data <- chiaretti$x[-training.samples, ]
y.test.data <- chiaretti$y.recode[-training.samples]

#LASSO LOGISTIC REGRESSION METHOD
#Step 4a: Fit Lasso model on training data
# Find the best lambda using cross-validation
set.seed(222)
cv.lasso <- cv.glmnet(x.train.data, y.train.data, alpha = 1, family = "binomial")
plot(cv.lasso)
# Fit the final model on the training data
lasso.classifier <- glmnet(x.train.data, y.train.data, alpha = 1, family = "binomial", lambda = cv.lasso$lambda.min)
# Display regression coefficients
lasso.coef <- coef(lasso.classifier)[,1]
# non-zero lasso coefficient
nonzero.lasso.coeff <- lasso.coef[lasso.coef != 0]
#plot top 10 predictors
lasso.top10 <- nonzero.lasso.coeff[order(-abs(nonzero.lasso.coeff))][2:11]
lasso.top10.df <- data.frame(predictors=names(lasso.top10),coefficent = lasso.top10)
ggplot(lasso.top10.df, aes(x = reorder(predictors,abs(coefficent)), coefficent)) +
  geom_bar(stat = "identity",fill ="darkmagenta") +
```

```
  coord_flip() +
  labs(y = "Coefficient", x = "Predictors(Genes)" ) +
  ggtitle("Top 10 Genes that Influence the Possibility of having Leukemia")

#Step 4b: Make predictions on the test data
lasso.probabilities <- lasso.classifier %>% predict(newx = x.test.data)
# Check model accuracy
lasso.predicted.classes <- ifelse(lasso.probabilities > 0.5, 1, 0)
# Model accuracy
mean(lasso.predicted.classes == y.test.data)

#Step 4c: calculate the AUC & ROC
auc(y.test.data, lasso.predicted.classes, plot=TRUE)


#LINEAR DISCRIMINANT ANALYSIS (LDA) METHOD
#Step 4a: Fit LDA model on training dataset
lda.classifier = lda(y~., data = data.frame(y=y.train.data,x.train.data))
#plot top 10 predictors
lda.top10<-
data.frame(predictors=substring(rownames(lda.classifier[["scaling"]]),2),coefficent       =
lda.classifier[["scaling"]])
lda.top10.df <- head(lda.top10[order(-abs(lda.top10$LD1)),],10)
ggplot(lda.top10.df, aes(x = reorder(predictors,abs(LD1)), LD1)) +
  geom_bar(stat = "identity",fill ="darkmagenta") +
  coord_flip() +
  labs(y = "Coefficients of linear discriminants", x = "Predictors/Genes" ) +
  ggtitle("Top 10 Genes that Influence the Possibility of having Leukemia")

#Step 4b: Make predictions on the test data
lda.probabilities <- lda.classifier %>% predict(data.frame(x.test.data))
# Check model accuracy
lda.predicted.classes                                                                    <-
as.numeric(levels(lda.probabilities$class))[lda.probabilities$class]
# Model accuracy
mean(lda.predicted.classes == y.test.data)

#Step 4c: calculate the AUC & ROC
roc(y.test.data, lda.predicted.classes, plot=TRUE)
```

```
#SUPPORT VECTOR MACHINE (SVM) METHOD
#Step 4a: Fit SVM model on training dataset
svm.classifier = svm(x = x.train.data, y = y.train.data,kernel = "linear")
beta = drop(t(svm.classifier$coefs)%*%x.train.data[svm.classifier$index,])
#plot top 10 predictors
svm.top10 <- data.frame(predictors=names(beta),coefficent = beta)
svm.top10.df <- head(svm.top10[order(-abs(svm.top10$coefficent)),],10)
ggplot(svm.top10.df, aes(x = reorder(predictors,abs(coefficent)), coefficent)) +
  geom_bar(stat = "identity",fill ="darkmagenta") +
  coord_flip() +
  labs(y = "Coefficients", x = "Predictors/Genes" ) +
  ggtitle("Top 10 Genes that Influence the Possibility of having Leukemia")

#Step 4b: Make predictions on the test data
svm.probabilities <- svm.classifier %>% predict(x.test.data)
# Check model accuracy
svm.predicted.classes <- ifelse(svm.probabilities > 0.5, 1, 0)
# Model accuracy
mean(svm.predicted.classes == y.test.data)

# Step 4c: calculate the AUC & ROC
roc(y.test.data, svm.predicted.classes, plot=TRUE)
```