Data 606: An Investigation into Breast Cancer Genes

Sean Anselmo, Matthew Haddad

December 3, 2024

Contents

1 Introduction

In the world of Big Data, the explosion and centralization of information has created pathways in industries never before seen. This is exceptionally true in the field of genetics. More data allows geneticists to draw conclusions about genes that would have otherwise not been possible. This project aims to investigate the Breast Cancer associated gene BRCA2.
The leading consortium of genetic information is a Harvard led German database known as gnomAD[2]. GnomAD has absorbed several other genetic bases to produce the most comprehensive set of genetic data available. The newest 2024 version of gnomAD includes a prediction of the gene's impact from ClinVar[1].

ClinVar is a clinical laboratory database that houses the real-world impact of variants. Clinics and labs analyze the impact of a variant (a person's given type of the gene) and upload their opinion of the variant to ClinVar. The opinions can be generalized into three categories; Benign (non-damaging), Unknown significance, and Pathogenic (damaging). The figure below demonstrates the severity tier list.

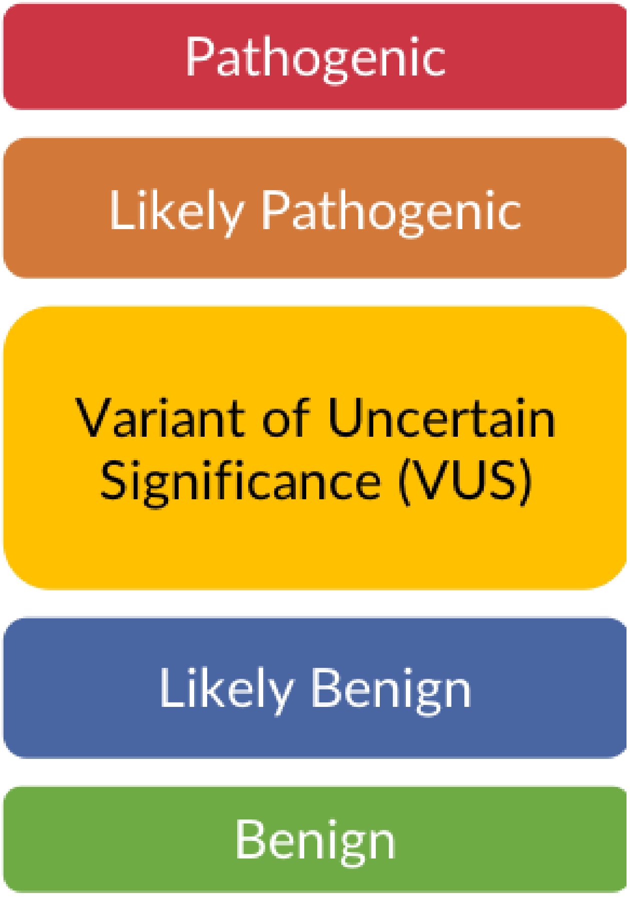


Figure 1: ClinVar Classification List

The combination of the massive genomic data from gnomAD and the real-world impact from

ClinVar allows us to make important conclusions and investigations into this dataset. We will explore the dataset further, before moving towards analyzing.

2 Methodology

2.1 Data

The dataset contains the following relevant fields:

i) ClinVar Clinical Significance: Laboratory database ClinVar's prediction on the impact of the variant.

ii) Allele Frequency: The frequency of the allele aggregated across the populations in gnomAD.

iii) CADD: Combined Annotation Dependant Deletion is a tool used to score single nucleotide mutations.

iv) PhyloP: A measurement of evolutionary conservation for each alignment.

v) Pangolin: A deep learning prediction tool that provides an assessment on a variant's potential pathogenicity.

vi) SpliceAI: A score that represents a variants effect on splicing.

vii) Group Max FAF Frequency: The highest allele frequency for said variantof any of the observed population groups

viii) Group Max FAF Group: The group associated with the highest allele frequency for each variant.

2.2 Approach

Transform the ClinVar predictions into a binary response variable, being either ”Benign” to ”Damaging”. Other predictors such as ”Unknown Significance” are to be removed. We are then to utilize different statistical techniques to determine the validity and accuracy of the aforementioned predictors. In silico predictors are often useful across a specific set of mutations. These have different purposes and goals.

2.3 Workload Distribution

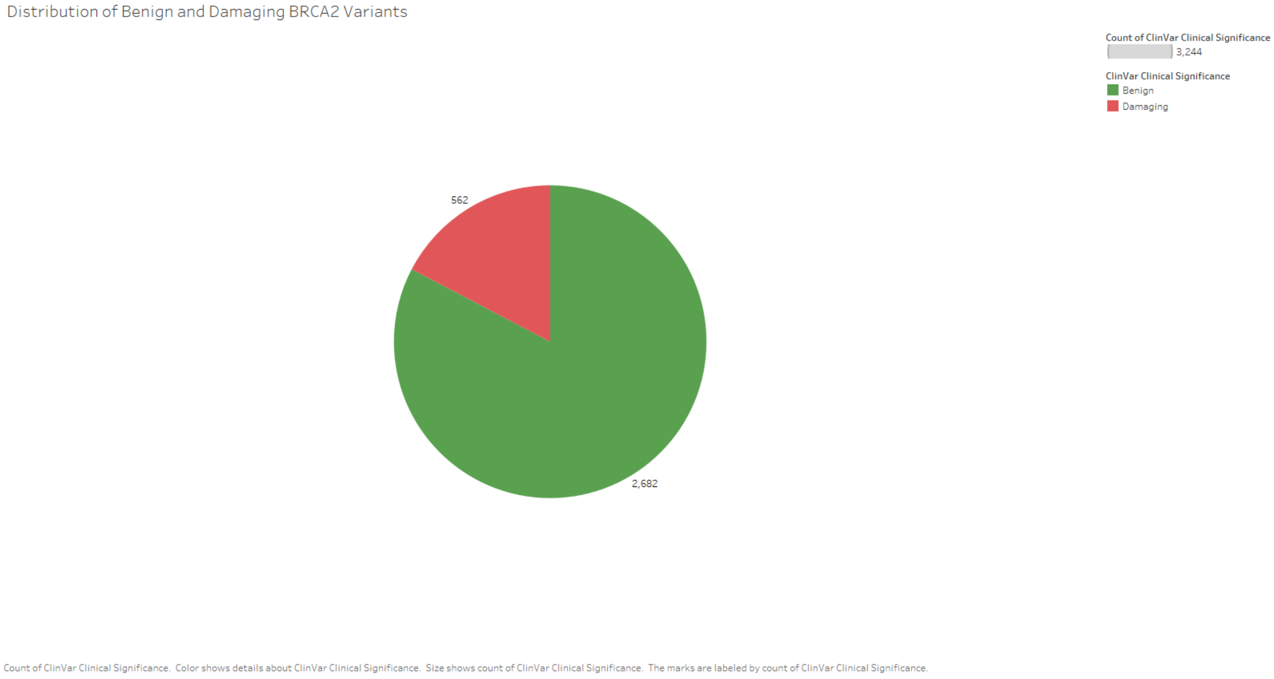
Sean will do methodology, data-wrangling. Matthew will do Results approach conclusion

3 Analysis

3.1 Data Cleaning and Wrangling

The data collected from gnomAD contained an immense amount of variables. These variables range from molecular information regarding the variant , to population metrics. We only kept the information that was pertinent to us. This includes the eight variables outlined above.

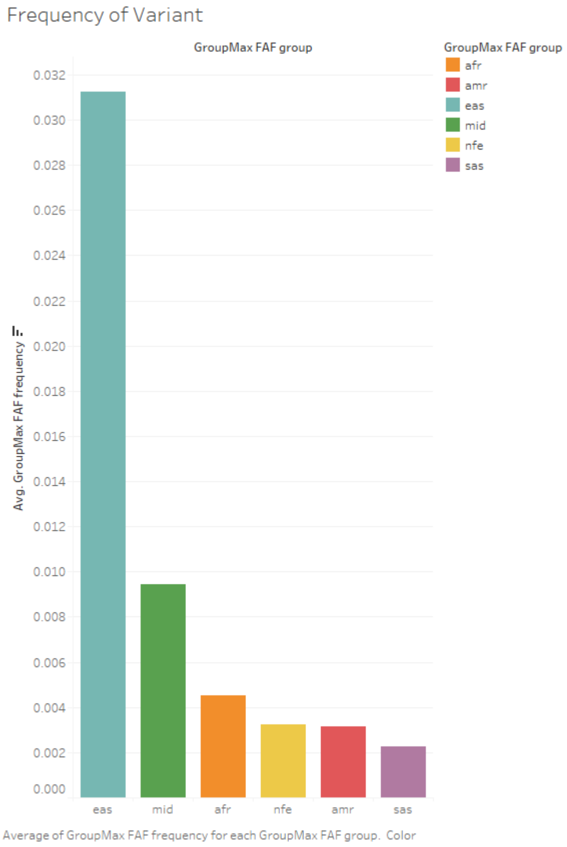
3.2 Exploratory Data Analysis


Distribution of Benign and Damaging BRCA2 Variants

Count of ClinVar Clinical Significance. Color shows details about ClinVar Clinical Significance. Size shows count of ClinVar Clinical Significance. The marks are labeled by count of ClinVar Clinical Significance.

Figure 2: Pie plot of total counts of benign and damaging variants

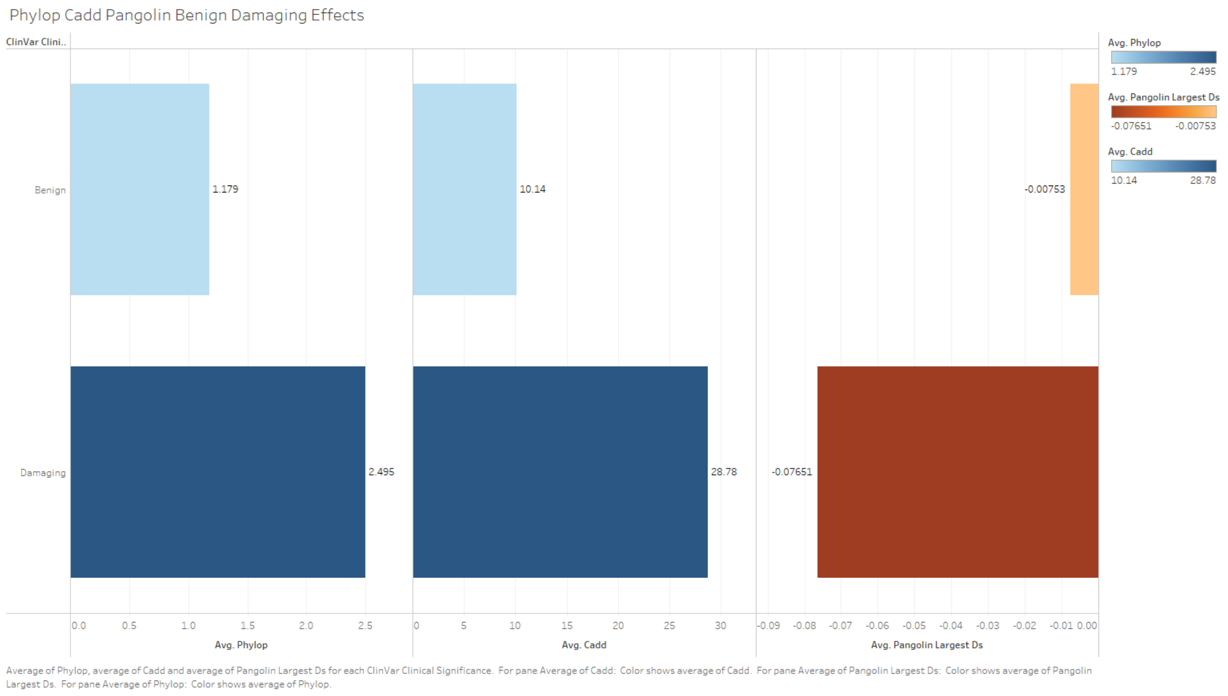
The above figure shows that there is an uneven distribution of variants, with more benign variants present in this dataset. Since it is a cancer gene BRCA2, there are a higher number of damaging variants than otherwise.



Average of GroupMax FAF frequency for each GroupMax FAF group. Color shows details about GroupMax FAF group. The view is filtered on GroupMax FAF group, which excludes Null.

Figure 3: Bar chart showing the frequency of variant for each world location

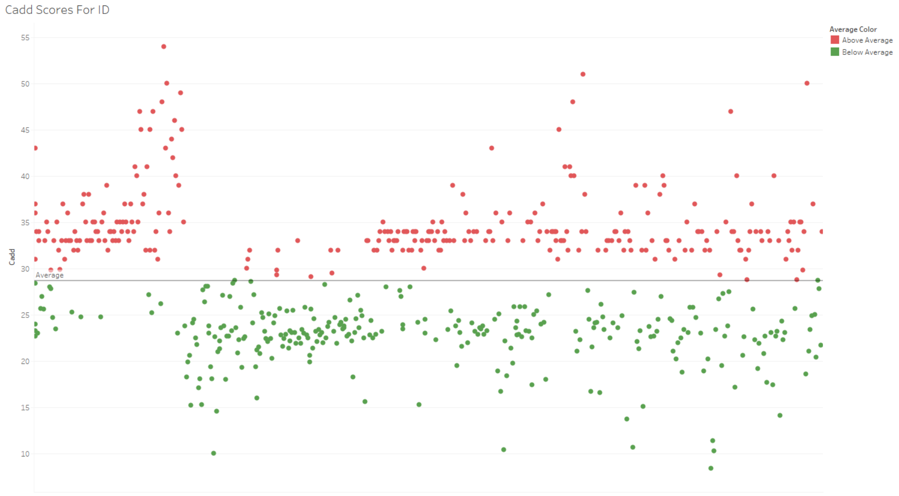
Eastern Asian populations have substantially higher average frequency for their variants than the other populations. This means, on average, the homogeneity of Eastern Asian populations are higher.



Average of Phylop, average of Cadd and average of Pangolin Largest Ds for each ClinVar Clinical Significance. For pane Average of Cadd: Color shows average of Cadd. For pane Average of Pangolin Largest Ds: Color shows average of Pangolin Largest Ds. For pane Average of Phylop: Color shows average of Phylop.

Figure 4: Predictors and counts of Benign and Damaging

The above chart shows the scores of in silico predictors CADD, PhyloP, and Pangolin against the amount of damaging and benign variants. Boxes represent average score for both Benign and Damaging variants. Pangolin scores in the reverse manner of the rest, with more negative scores being scored as more damaging.



Cadd for each Rs I Ds. Color shows details about Average Color. The data is filtered on ClinVar Clinical Significance, which keeps Damaging.

Figure 5: Damaging Cadd scores relative to average score

The CADD score figure above the distribution of damaging variants. The average line is shown across the centre, and variants above the average line are colored red, and variants below are colored green. This shows a semi-even distribution of below and above average damaging variants, with few outliers of note. We can conclude from this plot that there are two groups of CADD scores for damaging variants.

This initial exploration into damaging versus benign variants of the BRCA2 gene highlights significant disparities in their distribution and frequency across populations, particularly in Eastern Asian populations where damaging variants are more prevalent. In the proceeding portion of the report, we construct robust models that try to unravel the underlying factors influencing variant severity and prevalence. These models will help in understanding the genetic landscape of BRCA2 mutations, guiding future research and clinical strategies aimed at managing cancer risk associated with these variants.

3.3 Statistical Models

In this portion of the report we will propose four unique statistical models to predict malignancy of the BRCA2 genome.

1. Logistic Regression
2. Linear Discriminant Analysis (LDA)
3. Quadratic Discriminant Analysis (QDA)
4. Regression Tree

For each model the data was split into a 75% training set and 25% testing set.

3.4 Logistic Regression

Our logistic regression tests the prediction strength of a population genetics (Allele Frequency) and in silico predictors. To ensure the in silico predictors do not have influence on one another, we will check for multicolinearity. We did this via the "VIF" function, and returned values below 2 for each of our variables. With this, we can conclude there is no multicolinearity between our predictors.

Predictor	VIF
Allele Frequency	1.001864
CADD	1.924123
PhyloP	1.977230
Pangolin	1.385613
SpliceAI	1.398761

Table 1: VIF Values for Logistic Regression

We then created the Logistic regression model with Allele Frequency, CADD, PhyloP, Pangolin, SpliceAI. All but one variable was found to be significant, with SpliceAI recording a p value of 0.81. This is understandable that SpliceAI would be insignificant, since this predictor primarily focuses on splice site interactions and may miss deleterious content that other predictors like CADD may catch.

We trained the logistic regression on the first 75 percent of the dataset. Once the regression was applied to the test set, we compared our predicted to actual values. The following confusion matrix illustrates the results of our fitted regression.

	Actual	
Predicted	Benign	Damaging
Benign	653	40
Damaging	13	105

Table 2: Confusion Matrix for Logistic Regression

The misclassificaton rate for the confusion matrix is 0.06535142.

The model was created once more with the insignificant variable "Splice AI" removed. The confusion matrix remained the same, and thus the misclassification rate was unchanged (0.0653142).

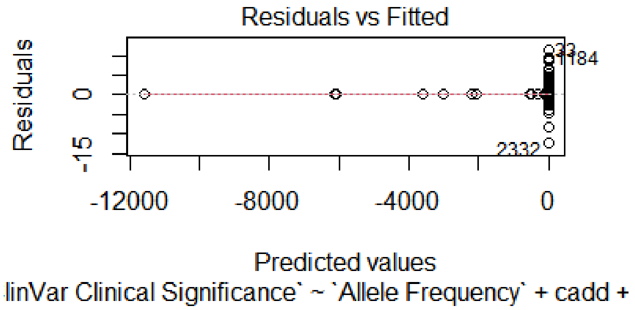


Figure 6: Uncleaned Residuals versus Fitted

The residual versus fitted plot revealed outliers that may be having an impact on the model. These outliers were removed, and the regression was performed on the cleaned data. The regression with insignificant variables removed produced a misclassification rate of 0.05925926, and the following confusion matrix.

	Actual	
Predicted	Benign	Damaging
Benign	671	34
Damaging	14	91

Table 3: Confusion Matrix for Cleaned Logistic Regression

The QQ plot on the residuals of the regression showed that the residuals followed a normal distribution, but not strictly. The following plot illustrates this:

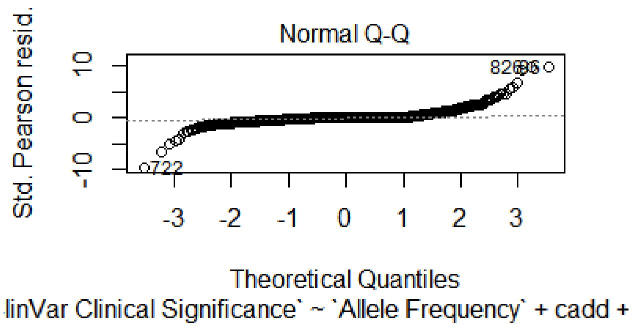


Figure 7: QQ Plot of Logistic Regression Residuals

The residuals versus fitted plot on the cleaned data with the extreme outliers did reveal a pattern. This may indicate higher order terms exist, however for this model interoperability is key and we have chosen to maintain it. We believe our model performs, as seen in the confusion matrix.

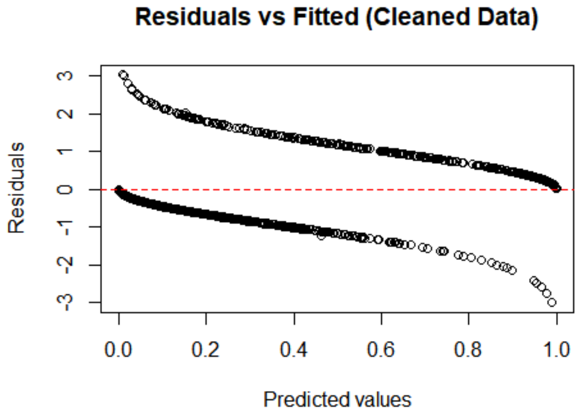


Figure 8: Cleaned Residuals versus Fitted

3.5 Linear Discriminant Analysis

In order to assume valid analysis of the LDA model, we must first check normality within each of our variables sets of data.

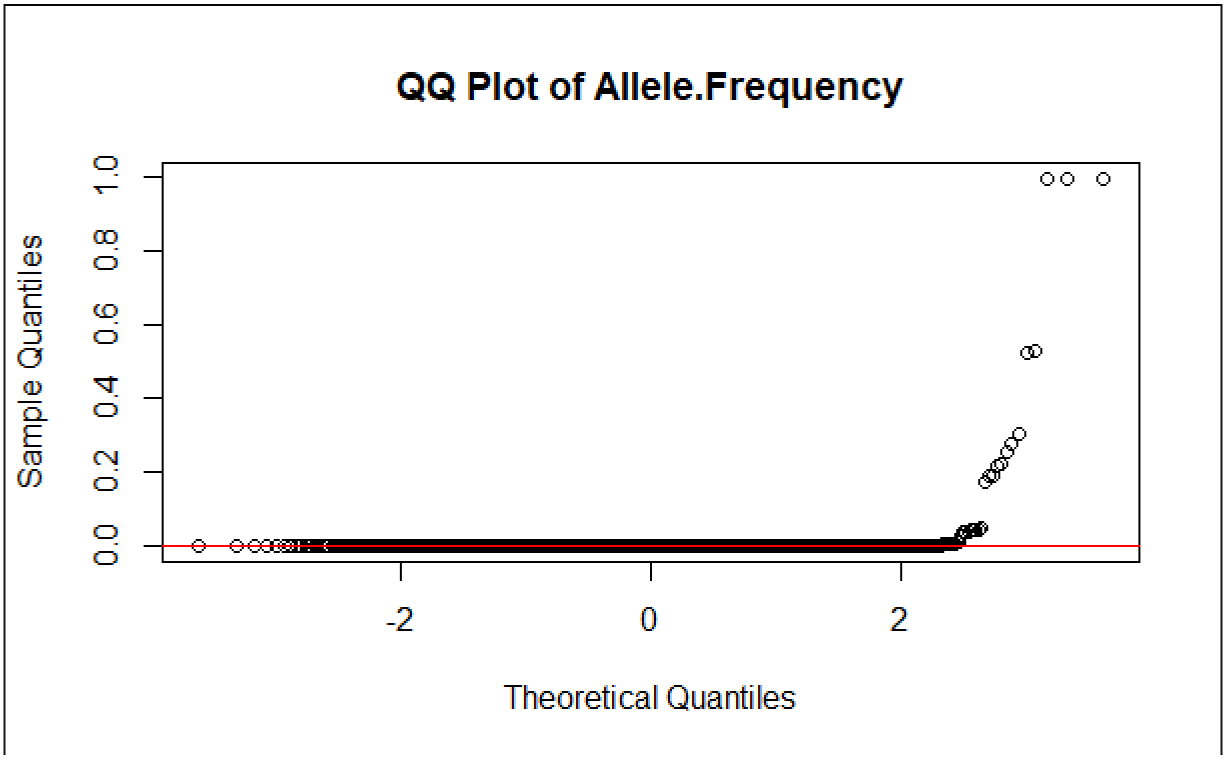


Figure 9: QQ plot of Allele Frequency variable

Most of the data seems to be aligned with the normal line. Removing the right tail outier data points will allow for a better look at normality within the allele frequency data.

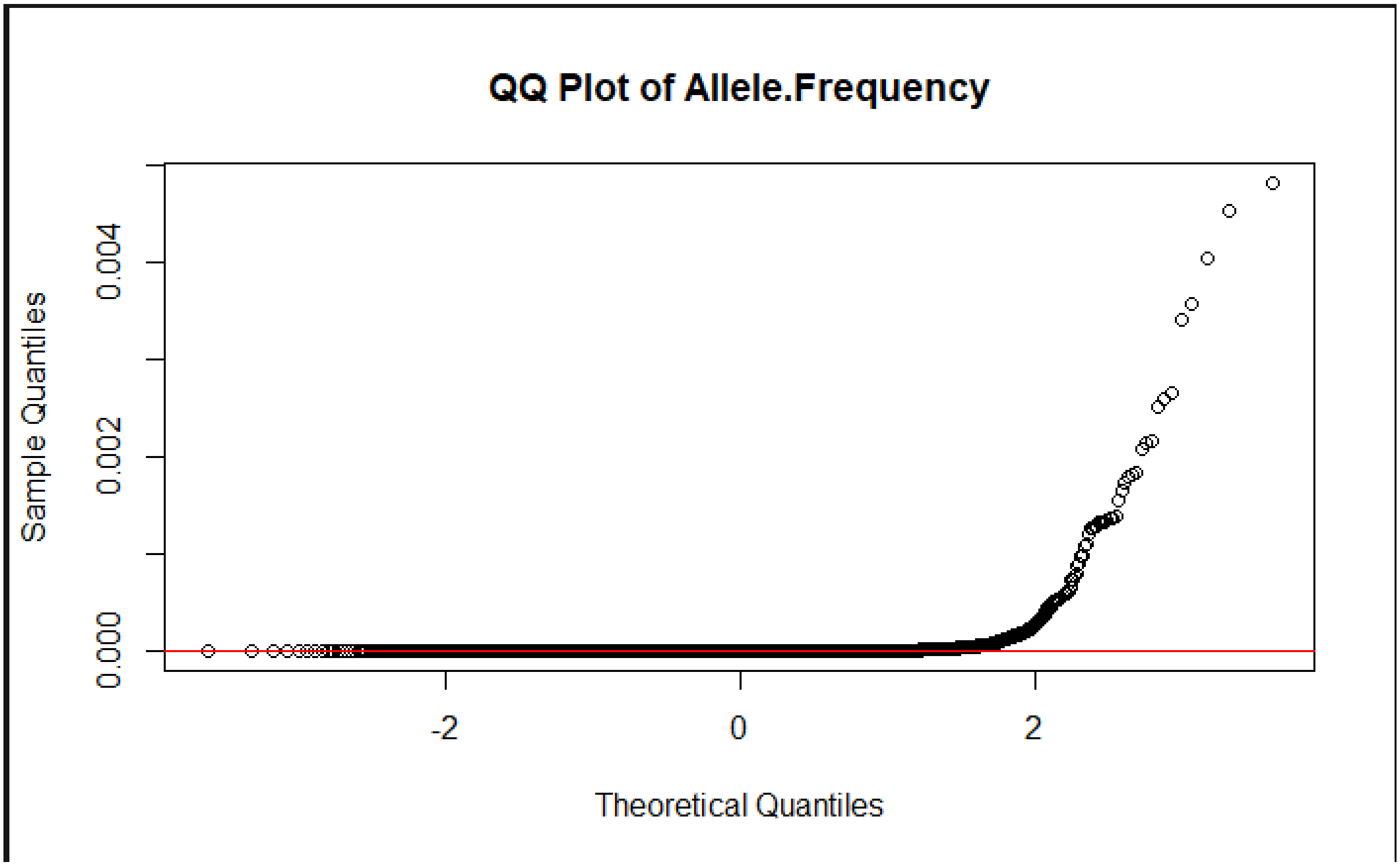


Figure 10: QQ plot of Allele Frequency variable with outliers removed

With the removed outliers we can see that the data deviates heavily at the right tail of the normal line. This data does not follow a normal distribution.

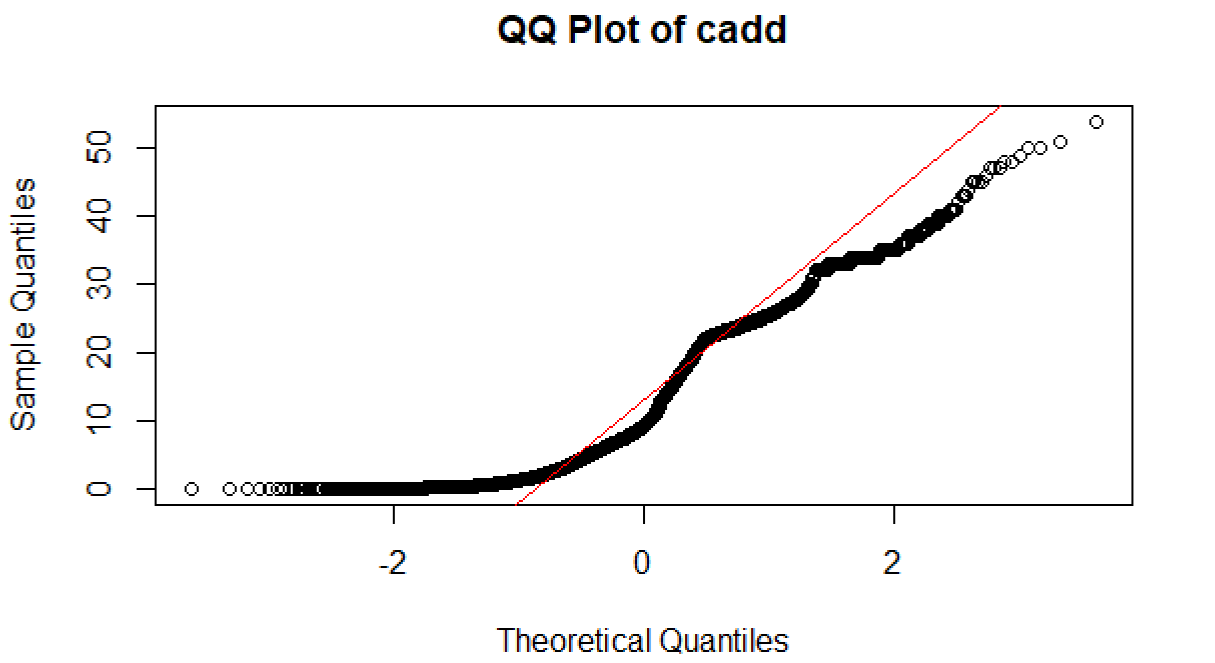


Figure 11: QQ plot of cadd variable

The left tail of the data along the normal line is significantly not aligned. To a lesser degree but still significant, the data along the right tail of the normal line is not aligned as well. The data for cadd scores does not follow a normal distribution.

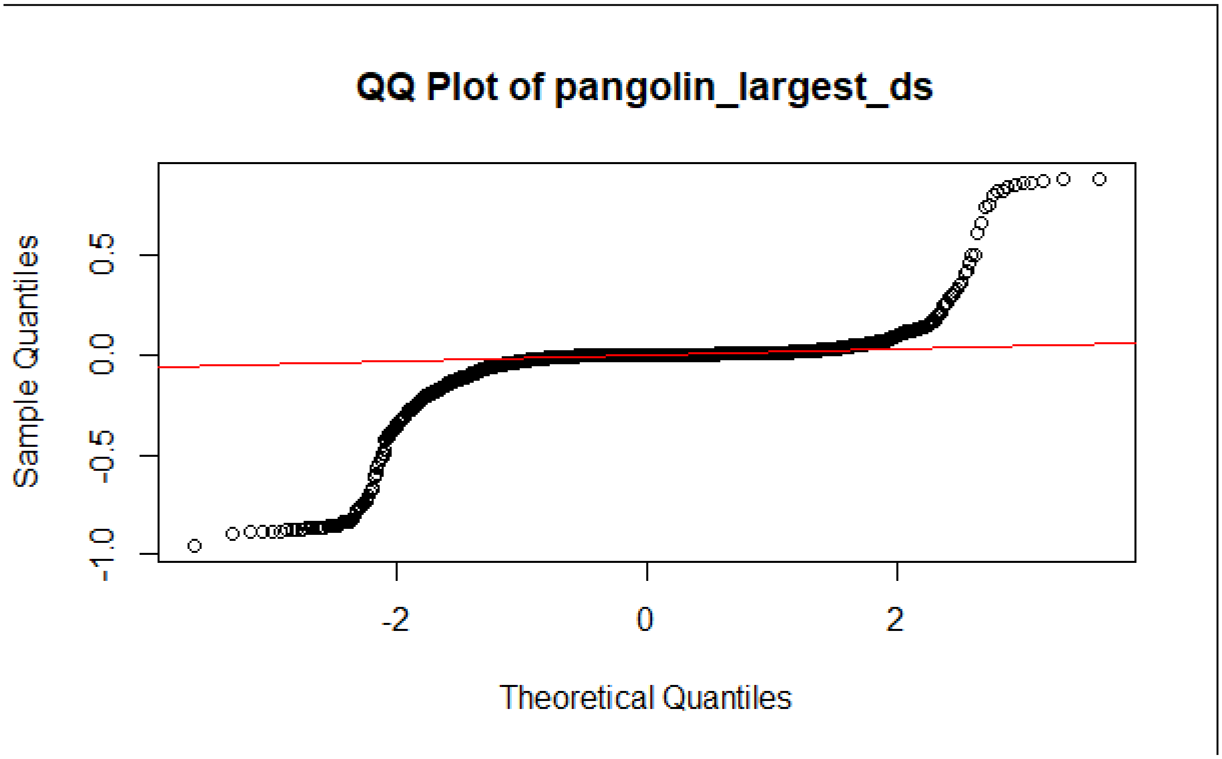


Figure 12: QQ plot of pangolin_largest_ds variable

The pangolin data follows the normal distribution between theoretical quantiles -0.5 and 0.5. The deviation about the normal line is symmetric and opposite with the left tail data skewing negatively and the right tail positively. The pangolin data does not follow a normal distribution.

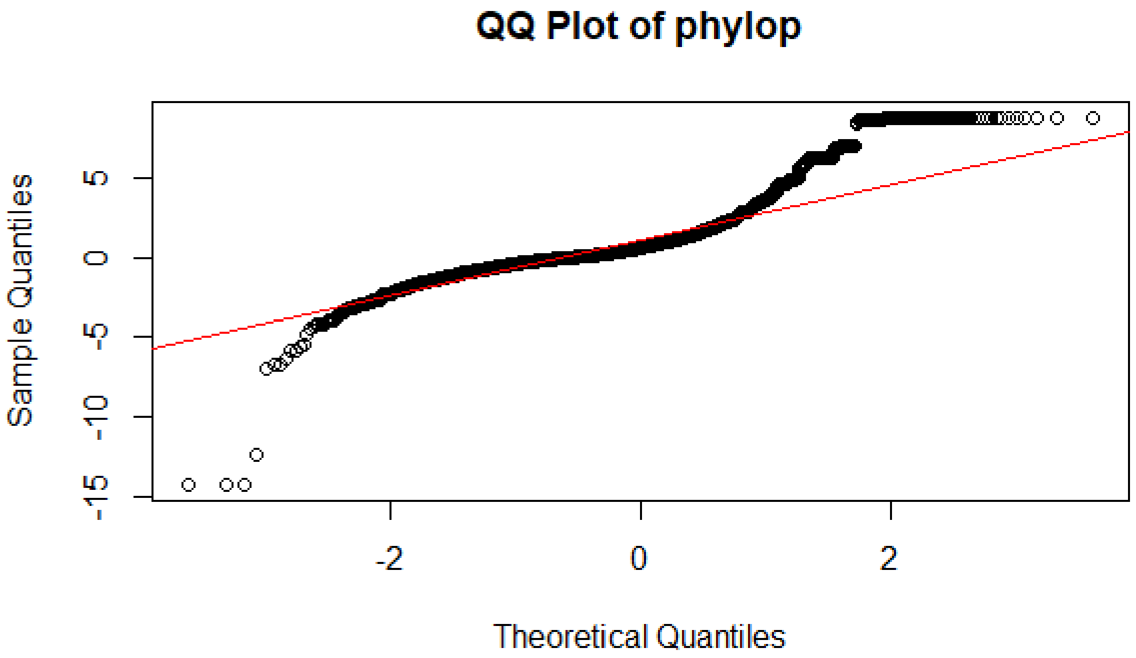


Figure 13: QQ plot of phylop variable

The phylop data is relatively normal between theoretical quantiles -2 and 0.5 but deviates about the left tail negatively and about the right tail positively. The phylop data does not follow a normal distribution.

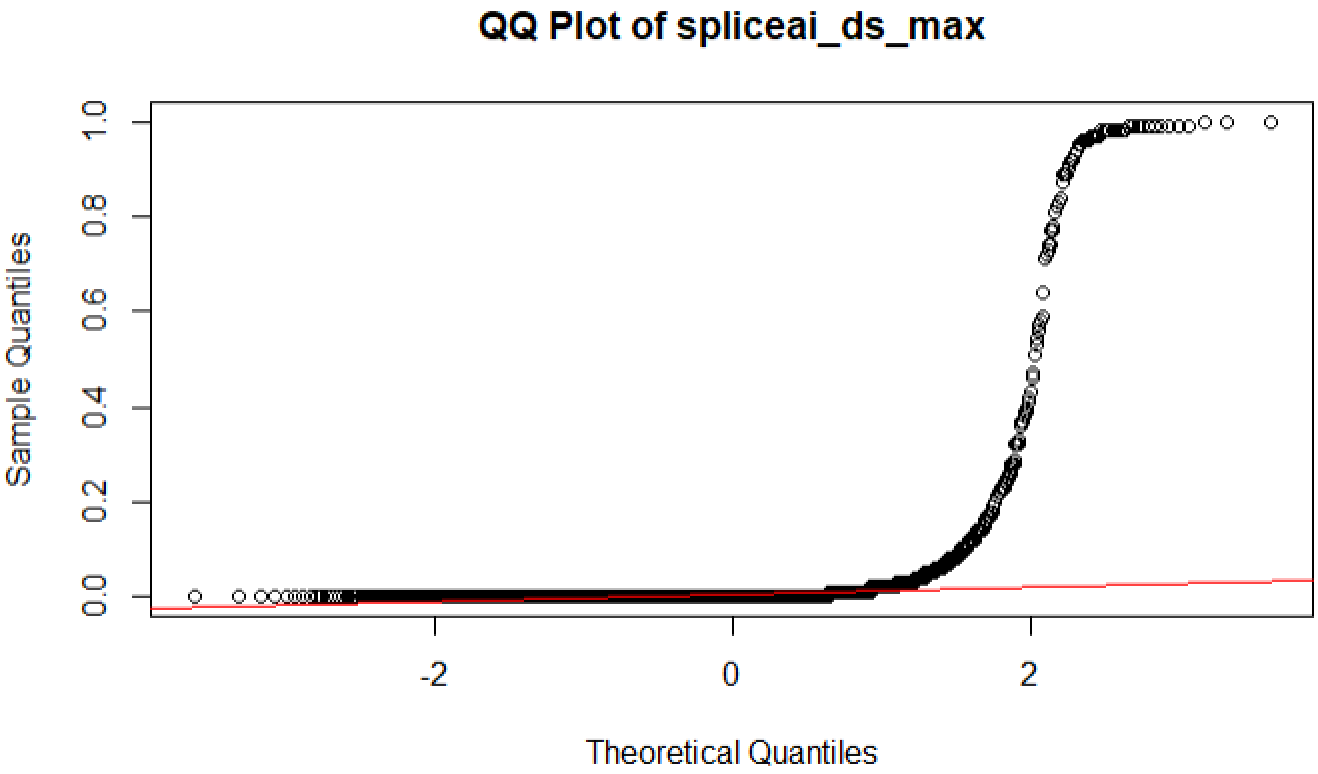


Figure 14: QQ plot of spliceai_ds_max variable

The spliceai data seems to follow a similar distribution to the allele frequency data. The data appears to follow the normal line up until theoretical quantile 0.5 and then deviates strongly positively. The spliceai data does not follow a normal distribution.

Due to the large deviation in the normal quantile distributions, we can say that all predictors are NOT normally distributed. Thus, the normal assumption for LDA does not hold. The continuation of study using this data is held because of the accuracy of prediction in the analysis further.

From completing an LDA model we find the prior probabilities of both groups are 83% and 17% for benign and damaging classes respectively. This tells us that the probability that an observation coming from a particular class is 83% and 17% for the respective classes benign and damaging.

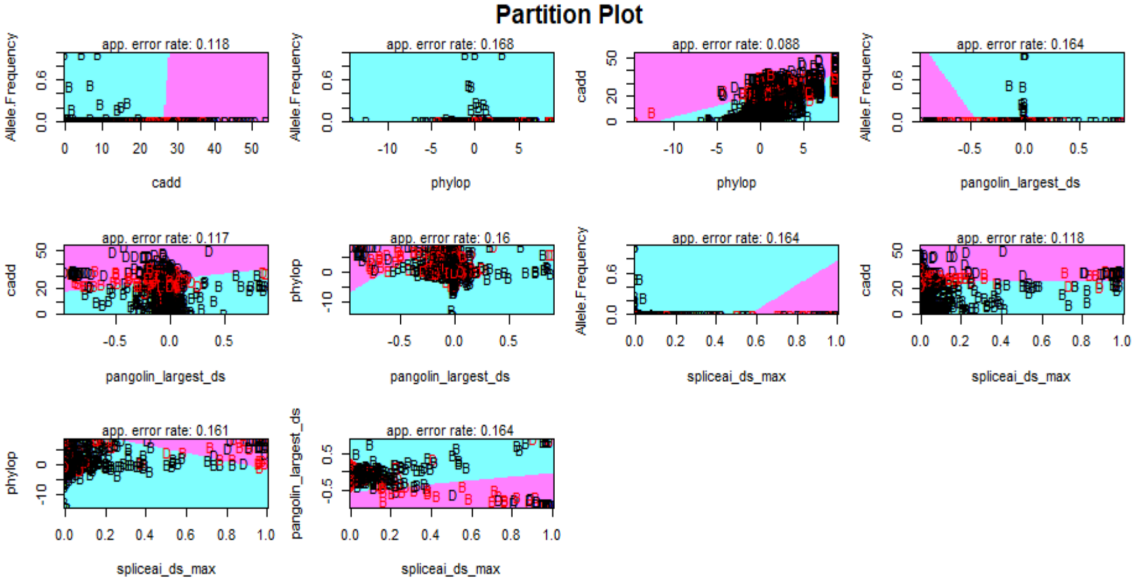


Figure 15: Partition plots of LDA model and each predictor

By looking at the partition plot of the LDA model we can see that the classification with the least error when based off of two variables is between cadd scores and phylop with an error rate of 0.09. The largest error at 0.17 is between predictors spliceai and allele frequency.

The confusion matrix will tell us the accuracy of the model:

	Actual	
Predicted	Benign	Damaging
Benign	641	37
Damaging	29	104

Table 4: Confusion Matrix for LDA

The accuracy for this model is 92% with a missclassification rate of 8%.

3.6 Quadratic Discriminant Analysis (QDA)

Before building the QDA model we must satisfy the assumption that each class has its own covariance matrix. We can analyze the matrices below:

BENIGN					
	Allele Frequency	Cadd	Phylop	Pangolin	Spliceai
Allele Frequency	1.477567e-03	-0.00967200	-0.001966973	-1.305772e-05	-0.0000459907
Cadd	-9.672000e-03	85.06081055	16.506296611	-8.542212e-02	0.1759315832
Phylop	-1.966973e-03	16.50629661	6.228672439	-3.200163e-02	0.0259986060
Pangolin	-1.305772e-05	-0.08542212	-0.032001634	9.031428e-03	-0.0001225203
Spliceai	-4.599070e-05	0.17593158	0.025998606	-1.225203e-04	0.0086092207

Table 5: Covariance matrix of Benign class

DAMAGING					
	Allele Frequency	Cadd	Phylop	Pangolin	Spliceai
Allele Frequency	6.36603e-11	-3.34423e-06	-1.19859e-06	9.74255e-08	-9.87210e-08
Cadd	-3.34423e-06	0.506205	0.114649	-2.96746e-01	0.337500
Phylop	-1.19859e-06	0.114649	0.860062	-3.26741e-01	0.320179
Pangolin	9.74255e-08	-0.296746	-0.326741	4.32916e-02	-0.0432916
Spliceai	-9.87210e-08	0.337500	0.320179	-0.0432916	0.058738

Table 6: Covariance matrix of Damaging class

We can see that the off diagonal values differ from the the covariance matrices in the benign and damaging classes. We can then take the assumption that each class has its own covariance matrix to be valid in this case.

From completing an QDA model we find the prior probabilities of both groups are 83% and 17% for benign and damaging classes respectively. This tells us that the probability that an observation coming from a particular class is 83% and 17% for the respective classes benign and damaging. This value remains the same as the LDA model analysis.

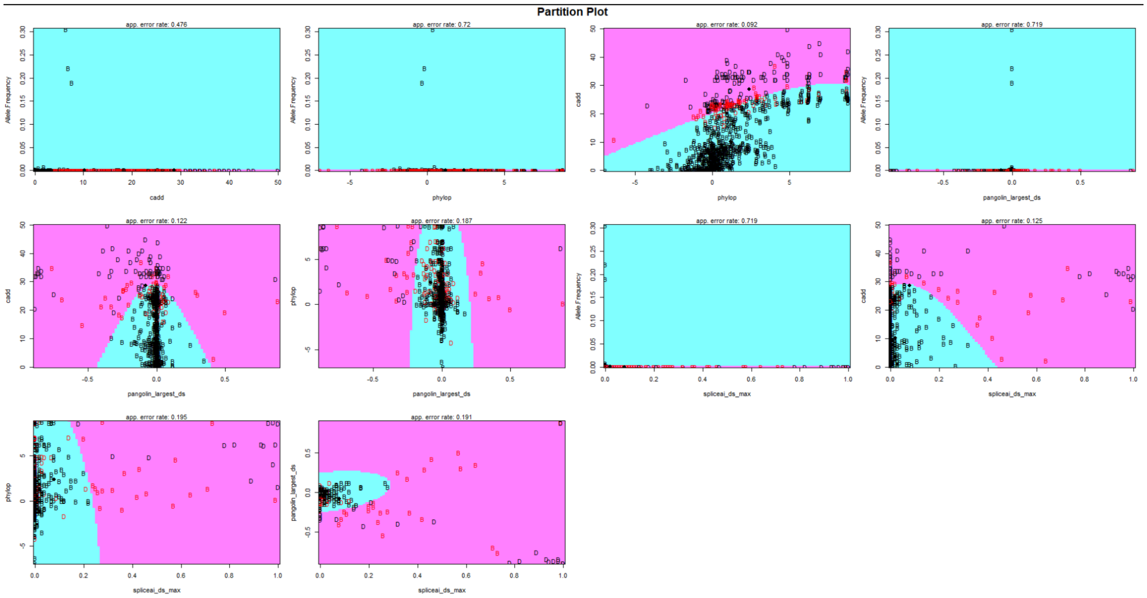


Figure 16: Partition plots of QDA model and each predictor

By looking at the partition plot of the LDA model we can see that the classification with the least error when based off of two variables is between cadd scores and phylop with an error rate of 0.1. The largest error at 0.758 is tied between predictor combinations spliceai : allele frequency and pangolin_largest_ds and Allele frequency.

The confusion matrix will tell us the accuracy of the model:

	Actual	
Predicted	Benign	Damaging
Benign	57	18
Damaging	613	123

Table 7: Confusion Matrix for QDA

The accuracy for this model is 28% with a missclassification rate of 72%.

3.7 Regression Tree

The un-pruned tree model can be seen in the figure below.

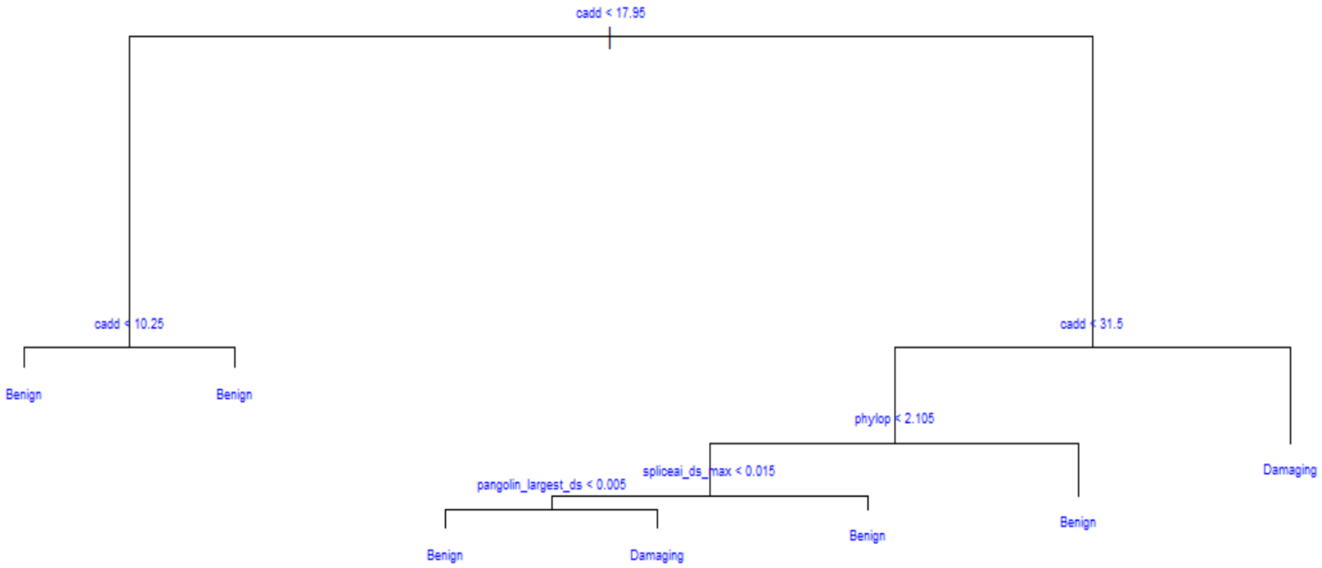


Figure 17: Diagram of regression tree

The regression tree has seven nodes with an accuracy of 93% and missclassification rate of 7%.

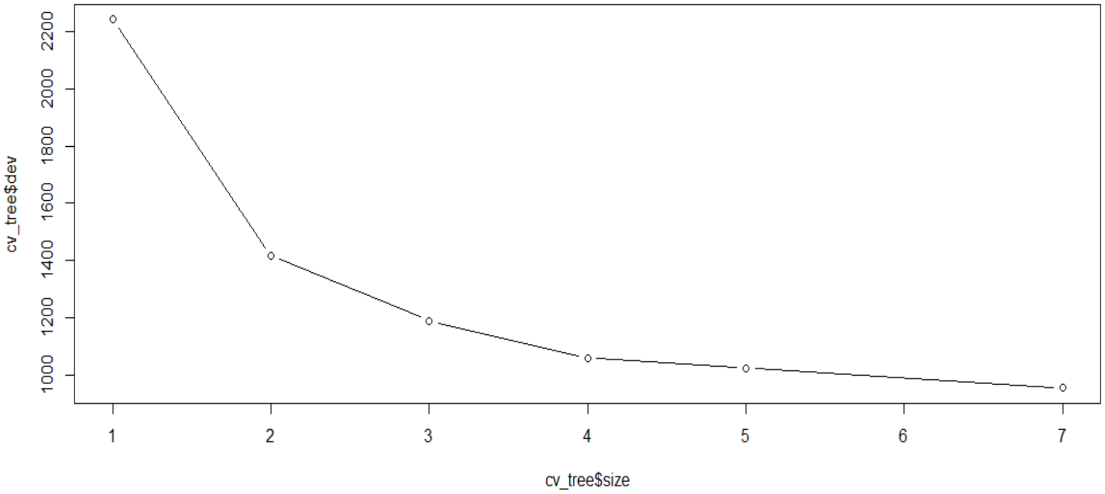


Figure 18: Error vs nodes in cross validation

Using cross validation to select the best number of nodes we can see that seven nodes has the least error. Due to the severity in classifying someone with cancer damaging gene and with how quickly the model takes to compute, we will continue to use seven nodes.

3.8 K Folds Cross Validation

K-fold cross-validation, is valuable for comparing logistic regression, LDA (Linear Discriminant Analysis), QDA (Quadratic Discriminant Analysis), and regression tree models because it provides a robust method to estimate model performance. By partitioning the data into k=10 subsets and iteratively training the models on k-1=9 subsets while validating on the remaining subset, k-fold cross-validation helps mitigate the risk of overfitting and provides a more reliable assessment of how well each model generalizes to unseen data. This approach also allows for a fair comparison of model performance metrics, such as accuracy or area under the ROC curve, ensuring informed model selection based on empirical validation rather than single-split data biases.
We can see the comparison of the k folds cross validation between the four models is:

	Model			
	Logistic Regression	LDA	QDA	Regression Tree
Missclassification Error	0.077	0.086	0.40	0.091

Table 8: Missclassification for each model in a K-Folds Cross Validation analysis

4 Conclusion

The distribution of variables in the dataset posed significant challenges, particularly in achieving the required assumptions for logistic regression and linear discriminant analysis. Performing logistic regression after removing outliers in the foremost residuals over fitted logistic regression plot, notably improved the misclassification rate from 6.5% to 6%. However, assumptions of homoscedasticity and normality were not met in logistic regression, highlighting limitations in the applicability of this model to our data. Calculating the variance inflation factor for the variables studied returned results all under two, indicating our model passes the assumption of multicollinearity.

Similarly, normality assumptions were not satisfied for both original and transformed data sets in the Linear Discriminant Analysis (LDA) model. Despite this, LDA demonstrated a reasonable misclassification rate of 8%, suggesting its potential utility in classification tasks. In contrast, the Quadratic Discriminant Analysis (QDA) model, while assuming independent covariance for each class (Benign and Damaging variants), resulted in a high misclassification rate of 72%, indicating extremely poor performance.

The regression tree model, with 7 nodes selected through cross-validation, provided efficient classification with a misclassification rate of 7%, highlighting its suitability for complex data structures.

Overall, in K-fold cross-validation analysis, the logistic regression model outperformed LDA, QDA, and regression tree models with the lowest misclassification rate of 0.077 (LDA: 0.086, QDA: 0.40, regression tree: 0.09) , demonstrating its superior accuracy in classifying cancerous genes based on BRCA2 variants. Interestingly, the K-fold cross-validation shows a significant decrease of the missclassification rate in the QDA model from 72% to 40% indicating that potentially, with more variable and more data, it could be a viable methodology.

In conclusion, while each model exhibited strengths and weaknesses in handling the dataset's complexities, the logistic regression model emerged as the most effective tool for accurately classifying BRCA2 gene variants, offering both practical efficiency and robust performance in predictive modeling.

4.1 Future Work

1. Implementing a Random Forest model following the above average performance of the regression tree, might offer significant advantages for classifying BRCA2 gene variants. The reduction in overfitting, and the betterment in capturing complex relationships and non-linearities present in the dataset are crucial advantages of random forest models that could help give firther insight into gene varient examination. Added robustness against outliers and noise is crucial for interpreting biological implications and guiding future research on BRCA2 variants as seen in this data and study. For these reasons we believe that this is why random forest modeling would be an asset.

2. Performing a chi-squared analysis would be instrumental in uncovering statistical relationships between benign and damaging effects of BRCA2 gene variants. By examining categorical data on variant classifications, this analysis could quantitatively validate patterns observed during exploratory data analysis (EDA), providing a deeper understanding of how different genetic variants contribute to benign or damaging outcomes. This statistical approach would enhance the insights gained from initial EDA, offering insight to explore potential associations crucial for characterization of BRCA2 variant effects.

5 Appendix

5.1 R Code

Following is the R Code used in computation of the model and the production of the figures

~~~
```{r Libraries and Data Import, include=FALSE}
library(ggplot2)
library(MASS)
library(ISLR)
library(klaR)
library(caret)
library(tree)
library(kmed)
library(car)
library(qqplotr)
library(Hotelling)
library(ggplot2)
library(dplyr)



~~~

~~~
data = read.csv("https://raw.githubusercontent.com/MHadd0/DataSets/main/BRCA2_filtered.csv")

data$ClinVar.Clinical.Significance <- as.factor(data$ClinVar.Clinical.Significance)
nrow(data)

# data <- data1[, c("ClinVar.Clinical.Significance", "Allele.Number", "Allele.Frequency", "Allele.Count")]
#
contrasts(data$ClinVar.Clinical.Significance)
head(data)

```

```{r Outliers, include=FALSE}

selected_data <- data[c(’Allele.Frequency’, ’cadd’, ’phylop’, ’pangolin_largest_ds’, ’spliceai_ds_max’)]

df_melt <- melt(selected_data)

colnames(df_melt) <- c("Variable", "Value")

colnames(df_melt)

# Create the box plot
ggplot(df_melt, aes(x = Variable, y = Value)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Box Plot of Variables",
       x = "Variable",
       y = "Value")



# Function to identify outliers based on IQR
identify_outliers <- function(x) {
  Q1 <- quantile(x, 0.25, na.rm = TRUE)
  Q3 <- quantile(x, 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1
  lower_bound <- Q1 - 1.5 * IQR
  upper_bound <- Q3 + 1.5 * IQR
  return(x < lower_bound | x > upper_bound)
}

# Apply the function to each column to get a logical matrix of outliers
outliers <- apply(selected_data, 2, identify_outliers)

# Get the indices of the rows containing outliers
outlier_indices <- which(rowSums(outliers) > 0)

# Remove the rows with outliers
selected_data_clean <- selected_data[-outlier_indices, ]
~~~

~~~

Melt the cleaned data frame for ggplot2
selected_data_clean_melt <- melt(selected_data_clean)

Rename the columns for better readability in the plot
colnames(selected_data_clean_melt) <- c("Variable", "Value")

Create the box plot for cleaned data
ggplot(selected_data_clean_melt, aes(x = Variable, y = Value)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Box Plot of Selected Variables (Outliers Removed)",
       x = "Variable",
       y = "Value")

Print outlier indices
print(outlier_indices)

clean_data = data[-outlier_indices,]


```


```{r, include=FALSE}

folds = createFolds(factor(data$ClinVar.Clinical.Significance), k=10)

training_amount = round(nrow(data)*0.75)
training_idx = sample(seq_len(nrow(data)), size=training_amount)

training_data = data[training_idx,]
test_data = data[setdiff(1:dim(data)[1], training_idx),]

test_data
head(training_data)
```


```{r LDA Model, include=FALSE}

lda_model = lda(ClinVar.Clinical.Significance ~ Allele.Frequency + cadd + phylop + pangolin_largest_ds + spliceai_ds_max, data=training_data)
lda_model


# contrasts(training_data$ClinVar.Clinical.Significance) # Benign 0, Damaging 1

plot(lda_model) # THE MEANS ARE NOT GOOD ON THE PLOT. PLOT BAD


~~~

```
‘‘‘

‘‘‘{r LDA Normality Assumption, include=FALSE}
par(mfrow=c(3, 2))

Normality within the predictor’s data
predictors_l = c(’Allele.Frequency’, ’cadd’, ’phylop’, ’pangolin_largest_ds’, ’spliceai_ds_max’)

create_qq_plot <- function(column, column_name) {
  qqnorm(column, main=paste("QQ Plot of", column_name))
  qqline(column, col = "red")
}

Identify the 15 largest values in "Allele.Frequency"
top_15_indices <- order(data$Allele.Frequency, decreasing = TRUE)[1:30]

Remove these rows from the dataframe
clean_data <- data[-top_15_indices,]
No transformation
for (i in 1:length(predictors_l)) {

  index = predictors_l[i]
  create_qq_plot(clean_data[[index]], index)
}

sqrt transformation (ASS)
for (i in 1:length(predictors_l)) {

  index2 = predictors_l[i]
  column_index2 <- data[[index2]]
  create_qq_plot(sqrt(column_index2), paste(index2, "- Sqrt Transformed"))
}

BOX-COX MODEL

Loop through each specified column and create QQ plots
for (i in 1:length(predictors_l)) {
  column_name <- predictors_l[i]
  column_data <- data[[column_name]]

  # Add a constant value (e.g., +10) to all data points
  column_data_translated <- column_data + 15

  # Apply Box-Cox Transformation if all values are positive
  if (any(column_data_translated <= 0, na.rm = TRUE)) {
    message(paste("Skipping column", column_name, "due to non-positive values for Box-Cox transformation"))
  } else {
```

~~~
    # Perform Box-Cox transformation
    bc <- boxcox(column_data_translated ~ 1, plotit = FALSE)
    lambda <- bc$x[which.max(bc$y)]  # Optimal lambda

    # Apply the transformation
    transformed_data <- if (lambda == 0) log(column_data_translated) else (column_data_translated^lambda - 1) / lambda

    # Remove NaNs and Inf values
    finite_transformed_data <- transformed_data[is.finite(transformed_data)]

    if (length(finite_transformed_data) > 0) {
      create_qq_plot(finite_transformed_data, paste(column_name, "- Box-Cox Transformed"))
    } else {
      message(paste("Skipping column", column_name, "due to all values being non-finite after transformation"))
    }
  }
}





```


```{r LDA Test, include=FALSE}

lda_pred = predict(lda_model, test_data)
names(lda_pred)


table(lda_pred$class, test_data$ClinVar.Clinical.Significance)

# misclassification rate
MCR = mean(lda_pred$class != test_data$ClinVar.Clinical.Significance)
MCR

# Accuracy
Accuracy = 1-MCR
Accuracy

print(paste("MCR:", MCR, "Accuracy:", Accuracy))
```

```{r LDA Partition Plot, include=FALSE}

partimat(ClinVar.Clinical.Significance ~ Allele.Frequency + cadd + phylop + pangolin_largest_ds + spliceai_ds_max, data=training_data, method="lda")

~~~

~~~
```


```{r QDA Model, include=FALSE}

qda_model = qda(ClinVar.Clinical.Significance ~ Allele.Frequency + cadd + phylop + pangolin_largest_ds + spliceai_ds_max, data=training_data)
qda_model

```


```{r QDA Assumption of Independant covariance matrix, include=FALSE}

# Split data by class
class_Benign <- subset(data, ClinVar.Clinical.Significance == "Benign")[, c('Allele.Frequency', 'cadd', 'phylop', 'pangolin_largest_ds', 'spliceai_ds_max')]
class_Damaging <- subset(data, ClinVar.Clinical.Significance == "Damaging")[, c('Allele.Frequency', 'cadd', 'phylop', 'pangolin_largest_ds', 'spliceai_ds_max')]


# Compute covariance matrices
cov_Benign <- cov(class_Benign)
cov_Damaging <- cov(class_Damaging)



# for (var_name in names(cov_matrices)) {
#   cat("Variable:", var_name, "\n")
#   result <- Hotelling.test(cov_matrices[[var_name]],
                              cov_Damaging[[var_name]])
#   print(result)
#   cat("\n")
# }

# Compare covariance matrices
print("Covariance Matrix for Class Benign:")
print(cov_Benign)
print("Covariance Matrix for Class Damaging:")
print(cov_Damaging)


```


```{r QDA Test, include=FALSE}

qda_pred=predict(qda_model, test_data)
names(qda_pred)


table(qda_pred$class, test_data$ClinVar.Clinical.Significance)
~~~

~~~

# misclassification rate
MCR_QDA = mean(qda_pred$class != test_data$ClinVar.Clinical.Significance)
MCR_QDA

# Accuracy
Accuracy_QDA = 1-MCR_QDA
Accuracy_QDA

print(paste("MCR:", MCR_QDA, "Accuracy:", Accuracy_QDA))
```

```{r QDA Partition PLot, include=FALSE}

partimat(ClinVar.Clinical.Significance ~ Allele.Frequency + cadd + phylop + pangolin_largest_ds + spliceai_ds_max, data=test_data,  method="qda")
```

TREE MODELS

```{r Tree, include=FALSE}

tree_model = tree(factor(ClinVar.Clinical.Significance)~Allele.Frequency + cadd + phylop + pangolin_largest_ds + spliceai_ds_max,data=training_data)
summary(tree_model)

```

```{r Plot Tree, include=FALSE}

plot(tree_model)
text(tree_model ,pretty =0, cex = 0.55, col="blue")

```

```{r Tree Leafs, include=FALSE}

cv_tree=cv.tree(tree_model)
plot(cv_tree$size,cv_tree$dev,type='b')

```

```{r Predict Tree, include=FALSE}

tree_pred = predict(tree_model,test_data, type = "class")
head(tree_pred)


a = table(tree_pred, test_data$ClinVar.Clinical.Significance)

~~~

~~~
print(head(a))

pred_class_counts <- table(tree_pred)
print(pred_class_counts)

confusion_matrix = confusionMatrix(tree_pred, test_data$ClinVar.Clinical.Significance)
confusion_matrix

MCR_TREE = mean(tree_pred != test_data$ClinVar.Clinical.Significance)
MCR_TREE

# Accuracy
Accuracy_TREE = 1-MCR_TREE
Accuracy_TREE

print(paste("MCR:", MCR_TREE, "Accuracy:", Accuracy_TREE))
```


```{r, include=FALSE}

# Fit the logistic regression model
lr_model <- glm(ClinVar.Clinical.Significance ~ Allele.Frequency + cadd + phylop + pangolin_largest_ds + spliceai_ds_max,
                family = binomial,
                data = train)

# Calculate the predicted values (fitted values)
predicted_values <- fitted(lr_model)

# Find the indices of the 50 lowest predicted values
indices_lowest_50 <- order(predicted_values)[1:50]

# Remove these rows from the training data
clean_train <- train[-indices_lowest_50, ]


clean_train
```


```{r USING FOLDS CV with tree, include=FLASE}

# Create folds for cross-validation
folds = createFolds(factor(data$ClinVar.Clinical.Significance), k = 10)

MCR_lda = numeric(10)
MCR_tree = numeric(10)
MCR_qda = numeric(10)
MCR_lr = numeric(10)

for (i in 1:length(folds)) {
  train_idx = unlist(folds[-i])
~~~

~~~
  test_idx = folds[[i]]

  train = data[train_indices, ]
  test = data[test_indices, ]

 lr_model = glm(ClinVar.Clinical.Significance ~ Allele.Frequency + cadd + phylop + pangolin_largest_ds + spliceai_ds_max, family = 'binomial', data = train)


   lr_predict = predict(lr_model, test, type = 'response')
   lr_class <- ifelse(lr_predict > 0.5, 1, 0)
   MCR_lr[i] = mean(lr_class != test$ClinVar.Clinical.Significance)

   # t = table(lr_class, test$ClinVar.Clinical.Significance)
   # print(t)


 lda_model = lda(ClinVar.Clinical.Significance ~ Allele.Frequency + cadd + phylop + pangolin_largest_ds + spliceai_ds_max, data = train)
   lda_predict = predict(lda_model, test)$class
   MCR_lda[i] = mean(lda_predict != test$ClinVar.Clinical.Significance)

   qda_model = qda(ClinVar.Clinical.Significance ~ Allele.Frequency + cadd + phylop + pangolin_largest_ds + spliceai_ds_max, data = train)
   qda_predict = predict(qda_model, test)$class
   MCR_qda[i] = mean(qda_predict != test$ClinVar.Clinical.Significance)

   tree_model = tree(ClinVar.Clinical.Significance ~ Allele.Frequency + cadd + phylop + pangolin_largest_ds + spliceai_ds_max, data = train)
   tree_predict = predict(tree_model, test, type = "class")
   MCR_tree[i] = mean(tree_predict != test$ClinVar.Clinical.Significance)


 }

cv_MCR_lda = mean(MCR_lda)
cv_MCR_tree = mean(MCR_tree)
cv_MCR_qda = mean(MCR_qda)
cv_MCR_lr = mean(MCR_lr)


cv_MCR_qda
cv_MCR_lda
cv_MCR_tree
cv_MCR_lr

```

```{r Logistic Regression, include=FALSE}

set.seed(123)
training_amount <- round(nrow(data) * 0.75)
training_idx <- sample(seq_len(nrow(data)), size = training_amount)
~~~

~~~

training_data <- data[training_idx, ]
test_data <- data[setdiff(seq_len(nrow(data)), training_idx), ]

test_data

Model.fit <- glm(ClinVar.Clinical.Significance ~ Allele.Frequency + cadd + phylop + pangolin_largest_ds + spliceai_ds_max,
                 family = binomial,
                 data = training_data)

predicted_values <- predict(Model.fit, newdata = test_data, type = "response")

clinical_significance_predict <- ifelse(predicted_values > 0.5, 1, 0)

actual <- test_data$ClinVar.Clinical.Significance
confusion_matrix <- table(clinical_significance_predict, actual)

print(confusion_matrix)

MCR <- mean(clinical_significance_predict != actual)

MCR


```
~~~

Bibliography

[1] *ClinVar*. URL: https://www.ncbi.nlm.nih.gov/clinvar/. (accessed 2024-06-01).

[2] *gnomAD browser*. URL: https://gnomad.broadinstitute.org/. (accessed 2024-06-01).