Problem Statement: To develop algorithms to classify genetic mutations based on clinical evidence.

#---

// loading the required libraries

```
library(data.table) // to convert the object into data.table
library(Matrix)  // for conversion into matrix
library(caret)   // for application of k-fold cross validation
library(tm)      // To process text NLP(Natural Language Processing)
library(forcats)  // To club the levels of a factor variable
library(e1071)    // To build SVM classsifier
```

#---

// GENERATING THE DATA SETS USING THE GIVEN TEXT FILES

After loading all the required libraries into R environment, the working directory is set to the location where all the Data is present and all the data is imported

Initially the train_text and test_text data files which are in text format are loaded and then transformed into data.table.

the columns in these data sets are ID, Text in which ID is a numeric type and Text is character type

Consequently, the train_variants and test_variants are loaded into R in such a way that all the strings are turned into factors by using 'stringsAsFactors' argument set to TRUE

the columns in these data sets are ID,Gene,Variants,Class.

in which ID is a numeric type

Gene is a factor variable

Variants is a factor variable

Class is a factor variable

Now, test_text and test_variants are merged by ID column to generate test data frame

train_text and train_variants are merged by ID column to generate train data frame

The train dataset has 5 columns but on the other hand test dataset has only 4 columns

so, a dummy column named CLASS with value -1 was generated so that the test and train datasets can be combined to form the full data set DATA which has 5 columns

#---

// NATURAL LANGUAGE PROCESSING

The genarated dataset is made to run though a series of data preprocessing steps in NLP

Now, the data which which has the class value = -1 is ignored and the rest is considered for run though NLP

Initially, a corpus is build on the datasource followed by several preprocessing steps which includes:

-> Removal of white spaces

-> Conversion of the existing text to lowercase

-> Removing Punctiations

-> Removing Stopwords

-> Conversion of all the words into Base words by stemdocumentation(Stemming)

-> Removing any numbers that are present in the corpus

-> Removal of any special characters that are present in the corpus using iconv function

-> Conversion of the corpus into Term Frequency- Inverse Document frequency to give weightage to the words present in the document

If a particular words' frequency is High then the weightage is given low as the name suggests

-> To remove sparse terms from term document matrix

Now, the DocumentTermMatrix is column binded to the original dataset which ends up at 3507 variables

#---

Building the SVM classifier

Initially the classifier is build on the train dataset to predict the Class variable outcome of the test dataset

and the accuracy was 63.855% which changes with change in the values in the training dataset which may lead to inconsistencies in the accuracies obtained

So, to overcome this K-FOLD CROSS VALIDATION MODEL is built on top of it.

#---

// Building K-FOLD CROSS VALIDATION

The resultant dataset obtained after NLP processing is divided into TRAIN AND TEST samples by random sampling into 70 % and 30 % respectively

To Balance the BIAS-TRADE off issue the dataset is split into 10 FOLDS using K-FOLD validation

The whole dataset is split into 10 parts in which 9 parts are made as training set and 1 part is made as test set and this changes in every

iterations using lapply in such a way that the model is trained on every possible data point.

ultimately, the predictions are made on each of these test sets and their respective accuracies are stored using a confusion matrix

The mean of the whole 10 individual accuracies is considered as the final accuracy of the model. and 1 - acuuracy gives the error %

Conclusion : An Accuracy of 63% was achieved.

#---