PROBLEM STATEMENT

Basically, the problem is all about personalised treatment regarding personalised treatment for cancer through genetic testing. Genetic map is unique for every individual. Mutations in genes leads to cancer. So our problem in precise is all about classifying clinically actionable mutations that led predefined treatment for cancer. Challenge lies in distinguishing mutilations that contribute to cancer growth. Currently this interpretation of genetic mutations is being done manually. This is a very time-consuming task where a clinical pathologist has to manually review and classify every single genetic mutation based on evidence from text-based clinical literature.

HYPOTHESIS GENERATION

Every individual is unique in nature their body structure and all.so each every one has different genetic makeup. Cancer is all about abnormal proliferation of cells due to mutation of genetic material. Generally any mutation in gene sequence may result in the variation of expressing protein which is is not normal in nature this led to have a chance for cancer may be a cancer producing gene .so we need to study the abnormal genetic mutations which gives information about type of cancer it cause and then accordingly treatment is done. Also the variation caused as a result of this genetic mutation is also to be taken into consideration for the cancer causing gene, so maximum this data will be form of text.with this this help pf these two descriptors we can classify the cancer treatment. Already doctors will keep record of this two descriptors abnormal genes and the variations that led to cancer so we can collect the data related to these two attributes in classifying. **Apart from these we can get access of information related to cancer causing agents from the research institutes , scientific journals, articles and various research papers related to cancer study.so we can collect that information analyse and get more features that can explain the target variable for better classification for more precise for classification of mutations that led to cancer.** Also we hear various types of cancers that were diagnosed eg blood cancer, lung cancer, throat cancer.so if we find the reasons that why this person effected with the cancer also why only that type of cancer in particular.like wise if we have information then we can predict to classify the actionable genetic mutations causing cancer for effective accuracy.

Exploratory Data Analysis(EDA)

Our main aim is to classify actionable genetic mutations that will result in precise customised treatment.treatment.If we see the data given

The data comes in 4 different files. Two csv files and two text files:

- training/test variants: These are csv catalogues of the gene mutations together with the target value Class, which is the (manually) classified assessment of the mutation. The feature variables are Gene, the specific gene where the mutation took place, and Variation, the nature of the mutation. The test data of course doesn't have the Class values. This is what we have to predict. These two files each are linked through an ID variable to another file each, namely:

- training/test text: Those contain an extensive description of the evidence that was used (by experts) to manually label the mutation classes.

The text information holds the key to the classification problem and will have to be understood/modelled well to achieve a useful accuracy.so how much effectively we model the clinical text evidence better we reach the aim in contributing in classifying for cancer predefined treatment

ID	Gene	Variation	Class
Min. : 0	BRCA1 : 264	Truncating Mutations: 93	1:568
1st Qu.: 830	TP53 : 163	Deletion : 74	2:452
Median :1660	EGFR : 141	Amplification : 71	3: 89
Mean :1660	PTEN : 126	Fusions : 34	4:686
3rd Qu.:2490	BRCA2 : 125	Overexpression : 6	5:242
Max. :3320	KIT : 99	G12V : 4	6:275
	BRAF : 93	E17K : 3	7:953
	ALK : 69	Q61H : 3	8: 19
	(Other):2241	(Other) :3033	9: 37

Glimpse(train)

Observations: 3,321

Variables: 4

$ ID <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15...

$ Gene <fctr> FAM58A, CBL, CBL, CBL, CBL, CBL, CBL, CBL, CBL, CBL...

$ Variation <fctr> Truncating Mutations, W802*, Q249E, N454D, L399V, V...

$ Class <fctr> 1, 2, 2, 3, 4, 4, 5, 1, 4, 4, 4, 4, 4, 4, 5, 4, 1, ...

Top Gene and its count in train data set

```
<fctr> <int>
##  1                BRCA1   264
##                A tibble: 264 x 2
##                   Gene    ct
##                   TP53   163
##  3                EGFR   141
##  4                PTEN   126
##  5               BRCA2   125
##  6                KIT    99
##  7                BRAF    93
##  8                ALK    69
##  9               ERBB2    69
##                10 PDGFRA    60
## # ...         with 254 more rows
```

Top Gene and its count in Test data set

```
## # A tibble: 1,397 x 2
##      Gene    ct
##    <fctr> <int>
##  1     F8   134
##  2   CFTR    57
##  3     F9    54
##  4   G6PD    46
##  5    GBA    39
##  6     AR    38
##  7    PAH    38
##  8   CASR    37
##  9   ARSA    30
## 10  BRCA1    29
## # ... with 1,387 more rows
```

Top variations and its count in the train data set

```
train %>%

## # A tibble: 5,628 x 2

##                Variation    ct

##                   <fctr> <int>

##  1 Truncating Mutations    18

##  2              Deletion    14

##  3         Amplification     8

##  4               Fusions     3

##  5                  G44D     2

##  6                 A101V     1

##  7                A1020P     1

##  8                A1028V     1

##  9                A1035V     1

## 10                A1038V     1

## # ... with 5,618 more rows
```

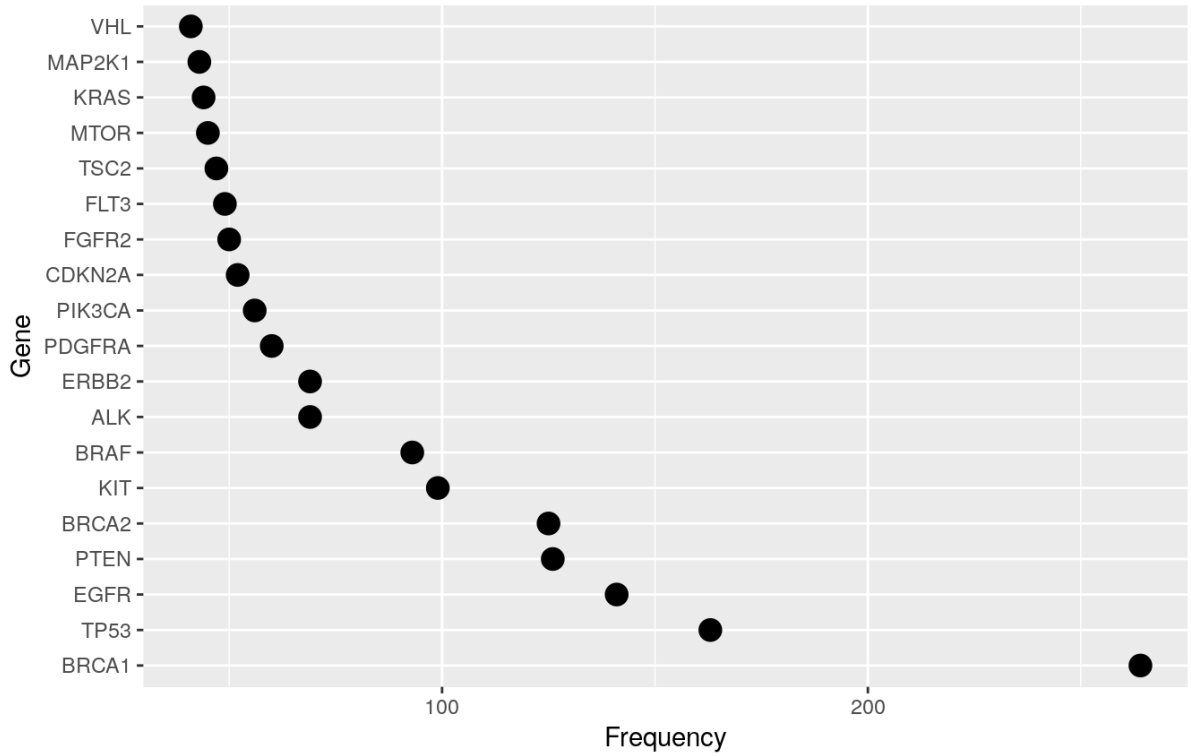
Findings

- There are 3321 different *IDs* in the training set containing 264 different *Gene* expressions with 2996 different *Variations*. There are 9 different *Classes* indicated by integer levels.
- The *Gene* and *Variation* features contain character strings of various lengths.
- There is 70% more test data than train data. The data description tells us that “Some of the test data is machine-generated to prevent hand labeling.”, which should explain this otherwise curious imbalance.
- There are no missing values in the variants data.
- The most frequent *Genes* in the train vs test data are complete different. In addition, the test data seems to contain significantly more different *Genes* and fewer high-frequency *Genes* than the train data. To some extent, this might be an effect of the added machine-generate entries in the test data (by adding many different random levels). Thereby, the difference in frequency might mirror the true fraction of effective test data over train data.
- In contrast, the most frequent *Variations* in train vs test are largely identical; although, again, the corresponding frequencies are lower in the test data (by a factor of 5 - 10).

Individual feature visualisations

This is the frequency distribution of the most frequent *Gene* values

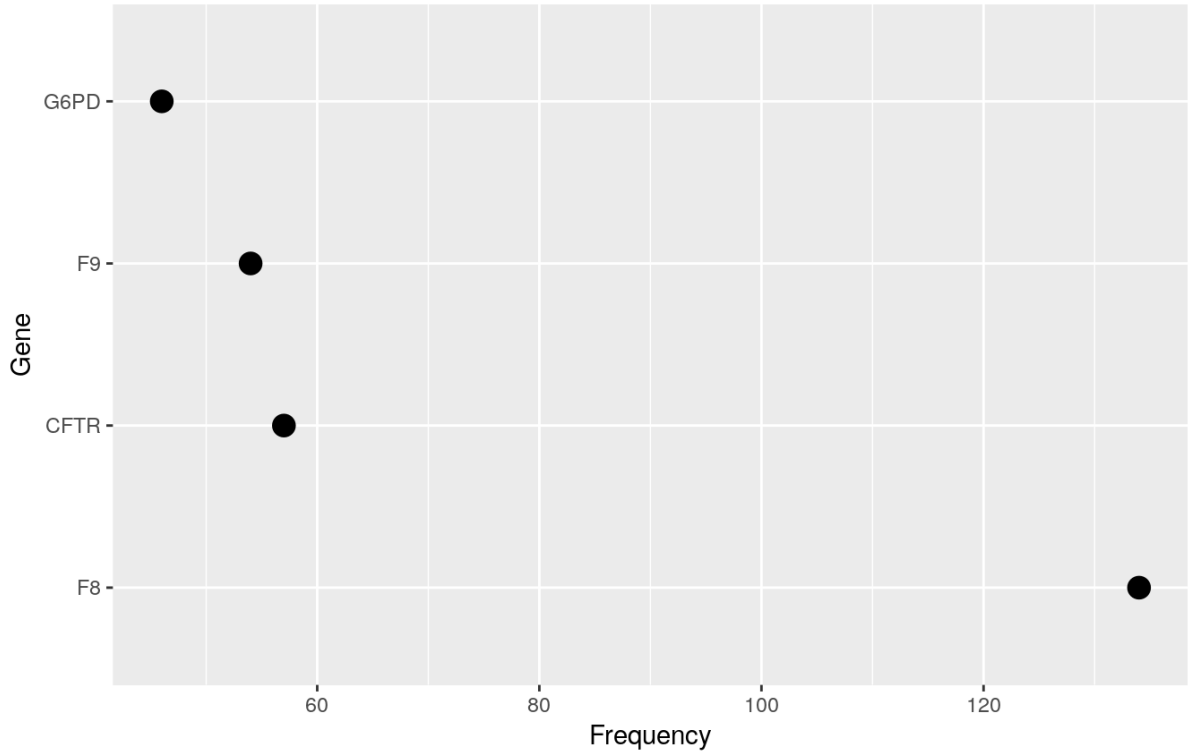
Top gene in train data set



Findings

- Some Genes, like “PTEN”, are predominatly present in a single Class (here: 4).
- Other Genes, like “TP53”, are mainly shared between 2 classes (here: 1 and 4).
- Classes 8 and 9 contain none of the most frequent Genes.

Gene vs frequency in Test Data set



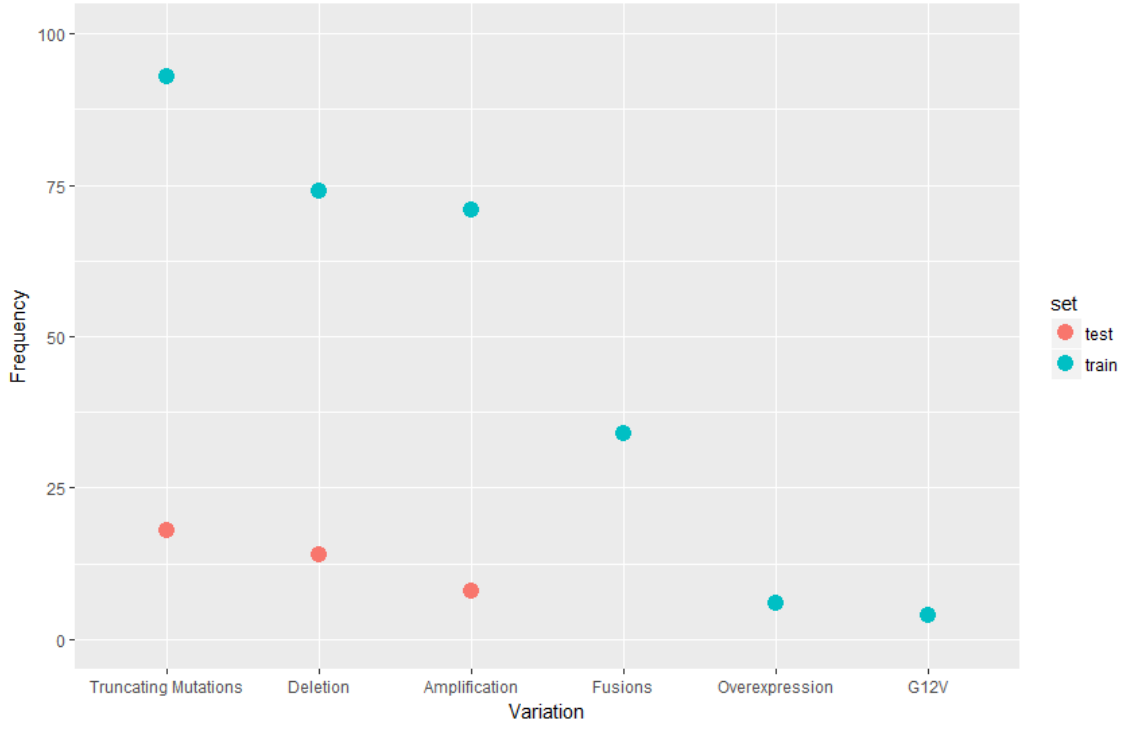
Findings

- A relatively small group of *Gene* levels make up a sizeable part of the feature values in both train and test data.
- The test data has fewer high-frequency *Gene*

Following below is the plot showing variation and its frequency of top variationsof both train and test data sets

Orange colour represents test dataset

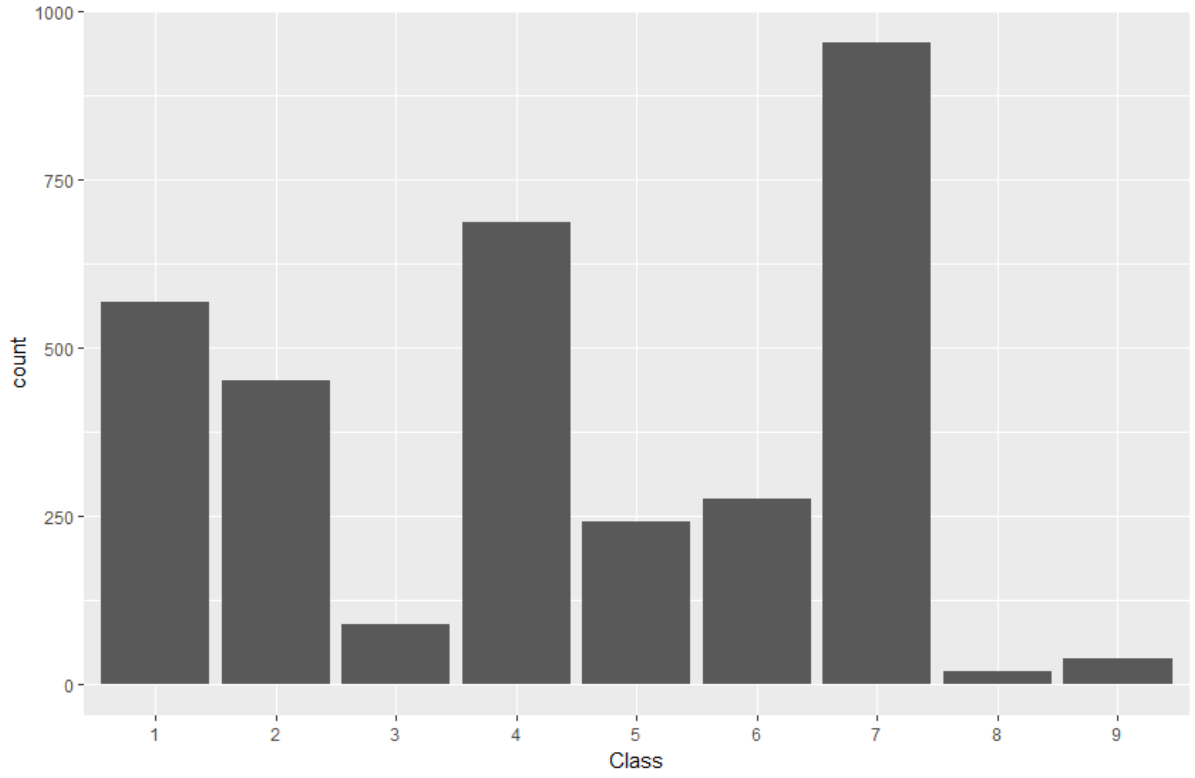
Blue colour represents train dataset



Findings:

Truncating variations, Deletion and Amplification are the top variation types in Train data when compared with Test data sets

Here we see how the *Class* target is distributed in the train data



Findings

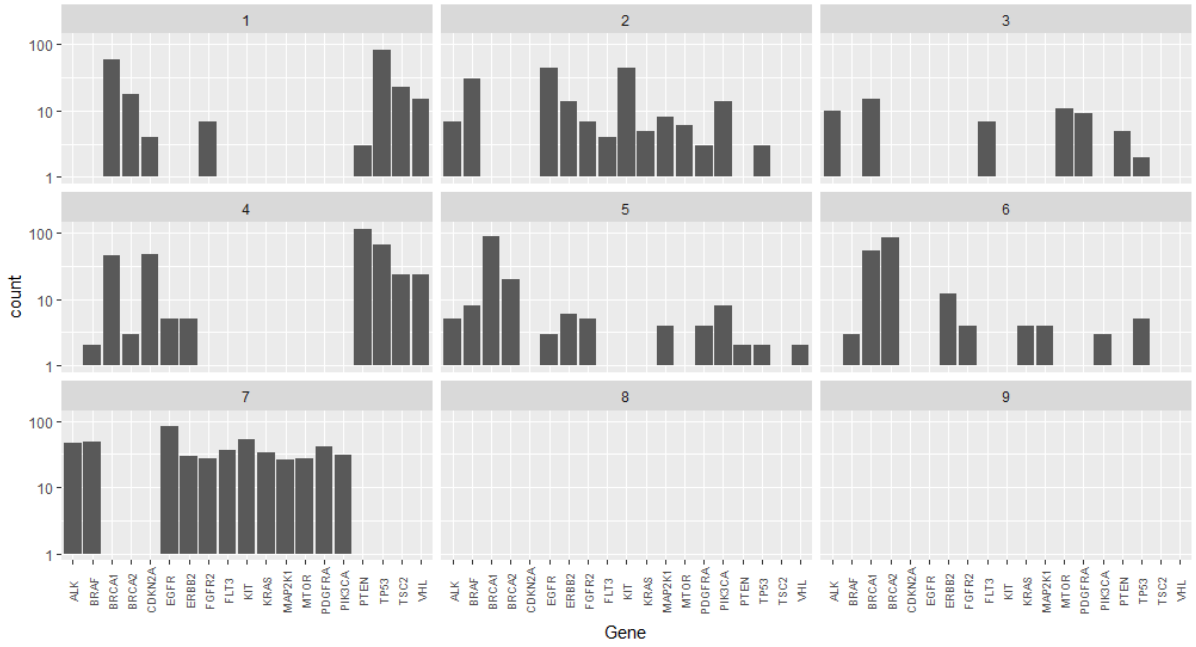
- *Class* levels 3, 8, and 9 are notably under-represented
- Levels 5 and 6 are of comparable, medium-low frequency
- Levels 1, 2, and 4 are of comparable, medium-high frequency
- Level 7 is clearly the most frequent one

Feature Interactions

Now we want to examine how the features interact with each other and with the target *Class* variable.

Gene vs *Class*

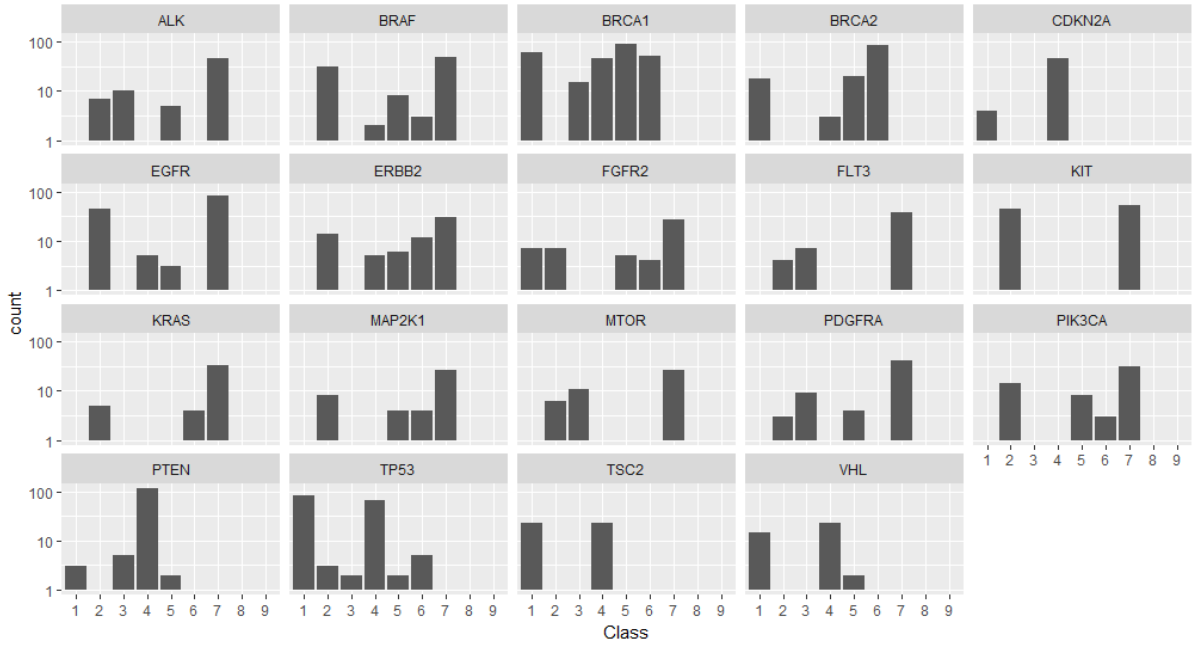
First, we will look at the frequency distribution of the overall most frequent *Genes* for the different *Classes*. Note the logarithmic frequency scale.



Findings

- Some *Genes*, like “PTEN”, are predominantly present in a single *Class* (here: 4).
- Other *Genes*, like “TP53”, are mainly shared between 2 classes (here: 1 and 4).
- *Classes* 8 and 9 contain none of the most frequent *Genes*.

Graph showing Classes sorted by *Genes*



Findings

This representation underlines our findings about the similar/dominating *Genes* in different *Classes.*

Next type of data is in the **form of text.** Huge data in the form of text is available. The main challenge also lies in modelling the text data.Here we analyse text in the form of document term matrix .More than 10lakh terms known as variables will be there **but we do a marvellous feature engineering technique called TF-IDF is used to extract best text features after doing various preprocees techniques also I had created my own stop list that would pose highly insignificant in explaining the target variable.**

Stopword List

"author","describe","find","found", "result",

"conclude","analyze","analysis","show","shown","resulted","concluded","described",

"concluded","evaluate","evaluated","discuss","discussed","demonstrate","demonstrated",

"the","this","that","these","those","illustrated","illustrate","list","fig","figure",

"et","al","data","determined","studied","indicated","research","method","determine",

"studies","study","indicate","research","researcher","medical","background","abstract",

"and","but","all","also","are","been","both","can","consider","describe","described",

"declar","determin","did","rt","http*"

Feature Engineering

Here in order to extract best features from the text we use TF-idf a feature engineering technique .

About TF-IDF technique

TFIDF, short for **term frequency–inverse document frequency**, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.[1] It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf-idf value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Nowadays, tf-idf is one of the most popular term-weighting schemes. For instance, 83% of text-based recommender systems in the domain of digital libraries use tf-idf. Text mining : To analyse the text data, we will first make it handy.

- For this purpuse the prepocessing of text data is required.
- The pre-processing should be done in the same way for train text and test text.
- Hence, the merging of both data set is required first.
- For this purpose, we will first merge variant data with text data for both and then will merge train and test.

By using TFIDF technique here on both train and text data we we extracted 2,670 best set features that could explain well about the target variable class.

Below are the top 10 highest frequency features

```
# A tibble: 10 x 2
      name    count
     <chr>    <dbl>
 1     alk 21.50574
 2    brca 40.53764
 3    egfr 25.08031
 4    fgfr 20.92278
 5    null 19.49332
 6    pten 25.57930
 7    smad 28.34689
 8     tsc 26.12717
 9 variant 20.95599
10     vus 20.15408
```

Now remaining variables apart from Gene , variation are taken as factor variables and our target variable (Class) is taken as a factor variable.

Since no data is missing, we combine these variables to build our model

Model Building

Since we are doing textual mining handling more than 2000 features as variables to predict our target variable. Taking an assumption that each and every feature is important in predicting the

target variable we build our model through **Naïve Bayes**. Since Naïve bayes works on bayes theorem of probability taking assumption that each and every feature has good probability in predicting the variable.

So we divide the train data into train and and validation set by simple random sampling by following 80/20 rule we train 80% of our data in building the model and remaining 20% of variable is taken as validation data.so in training data set of 3321 rows we takes 2657 rows of data to build model and remaining 664 rows of data as validation set to cross validate and find accuracy of the mode

Validation of the model

Validation of the model is done through validation set

After building model with train data and then used to predict on validation set.

Following are the various evaluation metrics of the model on validation data set

```
> ConfusionMatrix(pred,Validation_test$Class)
       y_pred
y_true   1   2   3   4   5   6   7   8   9
     1  45   2   0  29  18   9   8   2   0
     2   3  52   3   4   1   2  33   0   0
     3   0   0   8   2   1   0   4   0   0
     4  24   4  17  75  13   2  13   1   0
     5   5   5   3   3  24   8  10   0   1
     6   1   3   0   2   3  37   9   0   0
     7   4  26  13   4   4   0 113   3   0
     8   0   0   0   0   0   0   0   0   0
     9   0   1   0   0   0   0   1   1   5
> Accuracy(pred,Validation_test$Class)
[1] 0.5406627
> RMSE(y_pred=pred_prob,y_true=y_true)
[1] 0.3187591
> Logloss_score=MultiLogLoss(y_pred=pred_prob,y_true=y_true)
> Logloss
[1] 15.32905
>
```

Upon seeing the metrics, the model built so far is not upto the mark in predicting the target variable but so far fair enough to learn by applying different algorithms such as random forest, Xgb or otherwise need to pre-process the data well in enhanced manner before injecting to model after all learning is a continuous process.

Output

Now by building final model with complete training dataset on the competition data set i.e on test set in better way is saved as a **submission file containing prediction of classes with corresponding ID's.**

Github link for the project

https://github.com/GRMK11/Personalised-Medicine-project