Cancer Detection by predicting the effect of Genetic Variants

Submitted in partial fulfillment of the requirements of the degree

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

By

Roll No	Name
1804004	Ishan Agarwal
1804007	Musharraf Alam
1804008	Asawa Aryan
1804016	Yash Balchandani

Supervisor

Dr. Tanuja Sarode

(Head of Department, Department of Computer Engineering, TSEC)

Computer Engineering Department

Thadomal Shahani Engineering College

Bandra(w), Mumbai - 400 050

University of Mumbai

(AY 2020-21)

CERTIFICATE

This is to certify that the Mini Project entitled **"Cancer Detection by predicting the effect of Genetic Variants"** is a bonafide work of

Roll No	Name
1804004	Ishan Agarwal
1804007	Musharraf Alam
1804008	Asawa Aryan
1804016	Yash Balchandani

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering".**

(Dr. Tanuja Sarode)

Supervisor

Dr. Tanuja Sarode
Head of Department

Dr. G.T. Thampi
Principal

Mini Project Approval

This Mini Project entitled "---------Cancer Research and Diagnosis--------- "

by

Roll No	**Name**
1804004	Ishan Agarwal
1804007	Musharraf Alam
1804008	Asawa Aryan
1804016	Yash Balchandani

is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Examiners

1……
(Internal Examiner Name & Sign)

2……
(External Examiner name & Sign)

Date: 8th May 2018

Place: Thadomal Shahani Engineering College- (Bandra)

Contents

1. INTRODUCTION

1.1 GENERAL INTRODUCTION ON CANCER RESEARCH AND DIAGNOSIS

Cancer is not a single disease, but rather many related diseases that all involve uncontrolled cellular growth and reproduction. It is leading cause of death in the developed world and second in the developing world, killing almost 8 million people a year [7]. The early diagnosis and prognosis of a cancer type have become a necessity in cancer research, as it can facilitate the subsequent clinical management of patients.
For better clinical decisions, it is important to accurately distinguish between benign and malignant tumors [8]. Conventionally, statistical methods have been used for classification of high risk and low risk cancer, despite the complex interactions of high dimensional medical data [9]. To overcome the drawbacks of conventional statistical methods, more recently machine learning has been applied to cancer prognosis and prediction [10]. During the past few years, the increase in scientific knowledge and the massive data production have caused an exponential growth in databases and repositories. The knowledge discovery in databases, the ability to extract useful hidden knowledge and the development of methods and techniques for making use of data are becoming increasingly important in today's competitive world. BIOMEDICAL DOMAIN REPRESENTS ONE OF THE RICH DATA DOMAINS. An extensive amount of biomedical data is currently available with wealth of information (SEE FIG.1), ranging from details of clinical symptoms to various types of biochemical data and outputs of imaging devices.[3]

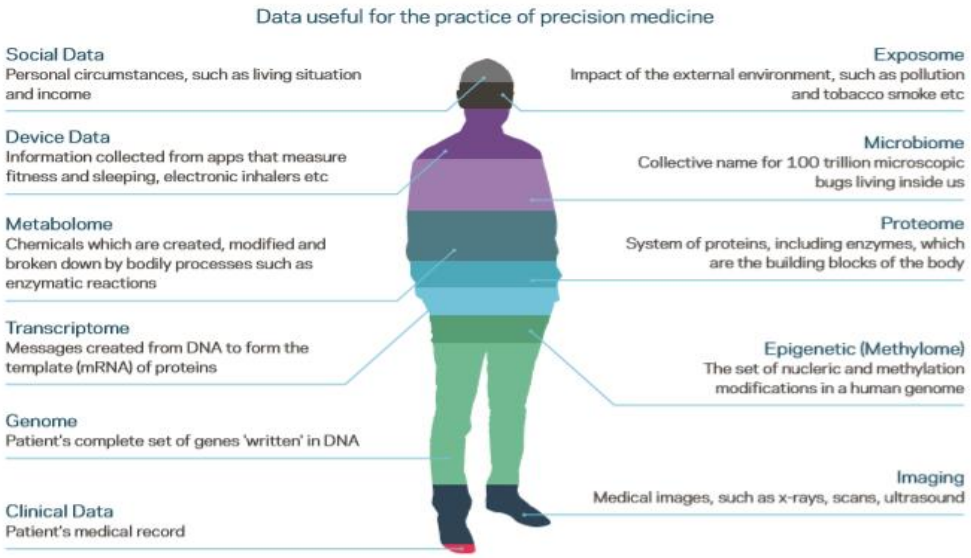


Fig 1: DATA USEFUL FOR THE PRACTICE OF PRECISION MEDICINE

The fundamental goals of cancer prediction and prognosis are distinct from the goals of cancer detection and diagnosis. In cancer prediction/prognosis one is concerned with three predictive foci:
1) the prediction of cancer susceptibility (i.e. risk assessment)

2) the prediction of cancer recurrence and
3) the prediction of cancer survivability.
In the first case, one is trying to predict the likelihood of developing a type of cancer prior to the occurrence of the disease. In the second case one is trying to predict the likelihood of redeveloping cancer after to the apparent resolution of the disease. In the third case one is trying to predict an outcome (life expectancy, survivability, progression, tumor-drug sensitivity) after the diagnosis of the disease. In the latter two situations the success of the prognostic prediction is obviously dependent, in part, on the success or quality of the diagnosis. However a disease prognosis can only come after a medical diagnosis and a prognostic prediction must take into account more than just a simple diagnosis.

1.2 PROBLEM DEFINITION

Cancer is not a monotonous disease; it consists of many different subtypes. Early diagnosis is a requirement in cancer, as it can facilitate the subsequent clinical management of patients into high or low-risk groups. The factors influencing the probability of a patient having cancer is huge. Genetic and mutations are a crucial factor in determining and diagnosis of cancer. To read, analyze and predict variation in such a huge dataset, the use of machine learning becomes necessary. With techniques such as the I Bayes Algorithm and KNN algorithm to store and classify the dataset as per similarity measures. In amidst the increase of the ageing population, increased availability of diagnostic tests and growing emphasis on precision medicine, machine learning could help them to do their jobs by identifying the high-risk cases they should focus on and helping them to make decisions about uncertain diagnoses. The goal is to use the Machine's Learning ability to recognize patterns that are too subtle for the human eye to detect to guide physicians towards better-targeted therapies and to improve outcomes for patients. Some scientists are even applying Machine Learning to screening tests in the hope of identifying people with an increased cancer risk or catching the disease sooner.

1.3 DOMAIN

- **MACHINE LEARNING**

  Machine Learning has many applications one of which is pointing subtle variation in the dataset and predicting results based on those variations. This spotting of subtle differences is necessary for the medical field as it can prevent incorrect diagnostic of any disease and hence provide correct medicine based on it. **DNA methylation in cancer** plays a variety of roles, helping to change the healthy regulation of gene expression to a disease pattern. Several research works have been done in this area. Here a classifier algorithm named "Logistic Regression" has been modified to detect the malignancy or benignancy of the tumorous cell more accurately. The computer's ability to spot those cancer types could cut hospitals' error rates. In the initial study, the algorithm found that 12% of brain tumours had been misdiagnosed by pathologists.

- **DATA MINING**
  Data Mining. In simple words, data mining is defined as a process used to extract usable data from a larger set of raw data. It implies analysing data patterns in large batches of data using one or more software. Data mining has applications in multiple fields, like science and research. Data mining has various techniques (such as Classification, Clustering, Regression, Association Rules, etc.,) and algorithms (such as Decision Trees, Genetic Algorithm, Nearest Neighbor method etc.,) for analyzing a huge amount of raw or multi-dimensional data. In the other words, data mining has capabilities for intelligent data analysis to extract hidden knowledge from large databases of medical or clinical data that are collected from medical centers or hospitals. This knowledge provides useful information to improve decision support, prevention, diagnosis and treatment in the medical world.

1.4 NEEDS

Cancer is the second leading cause of death globally and accounted for 8.8 million deaths in 2015. As of 2019, about 18 million new cases occur annually. Annually, it caused about 15.7% of total deaths. It has been characterized as a heterogeneous disease consisting of many different subtypes. Due to the poor implementation of healthcare system and lack of necessary medical equipment cancers are often diagnosed at later stages when the condition is beyond the curable stage. Machine learning is a branch of artificial intelligence that employs a variety of statistical, probabilistic and optimization techniques that allows computers to “learn” from past examples and to detect hard-to discern patterns from large, noisy or complex data sets. This capability is particularly well-suited to medical applications, especially those that depend on complex proteomic and genomic measurements. With the help of Machine Learning, we can perform diagnosis at an early stage where most of the symptoms may go unnoticed and hence can bring down the number of cancer patients by a considerable amount. Since Cancer has many subtypes, it can be very prelexical for the medical practitioner to identify it and this can lead to incorrect diagnosis, Treatment for the wrong cancer could have ill effects without actually destroying the cancer. As a result, machine learning is frequently used in cancer diagnosis and detection. It also assists to avoid running an unnecessary diagnostic test on the patient which successively reduces the cost and time of both, the practitioner and the patient.

1.5 SCOPE

Cancer Research and Diagnosis Model first read the Gene and Variation Data. Then it preprocesses the data by removing Stop words, punctuations and the Null values; while replacing every special char with space and multiple spaces

with a single space and converting all the chars into lower-case, also retaining the word if it is not a stop word from the data. Later it processes the text and merges the gene variations and text data based on ID. Then test, Train and Cross-validation spilt the dataset to 6hosphor it to a less. This in turn helps us calculate the Gene Feature. Since the gene is a categorical variable, we featurize it using ‘One hot Encoding ‘ or ‘Response Coding’ and repeat the same with Variation. Then Train a Logistic regression+Calibration model using text features which re on-hot encoded. The data goes through multiple algorithm and mining sessions. And the highest probability is selected to know if the person has cancer or not.

1.6 APPLICATION

In assembling this review a number of trends are noted, including a growing dependence on protein biomarkers and microarray data, a strong bias towards applications in prostate and breast cancer, and a heavy reliance on “older” technologies such artificial neural networks (ANNs) instead of more recently developed or more easily interpretable machine learning methods. A number of published studies also appear to lack an appropriate level of validation or testing. Among the better designed and validated studies, it is clear that machine learning methods can be used to substantially (15–25%) improve the accuracy of predicting cancer susceptibility, recurrence and mortality. At a more fundamental level, it is also evident that machine learning is also helping to improve our basic understanding of cancer development and progression**.**

2. DESIGN

2.1 LITERATURE OF SURVEY

Over the past decades, a continuous evolution related to cancer research has been performed. Scientists applied different methods, such as screening in early stage, in order to find types of cancer before they cause symptoms. Moreover, they have developed new strategies for the early prediction of cancer treatment outcome. With the advent of new technologies in the field of medicine, large amounts of cancer data have been collected and are available to the medical research community. However, the accurate prediction of a disease outcome is one of the most interesting and challenging tasks for physicians. As a result, ML methods have become a popular tool for medical researchers. These techniques can discover and identify patterns and relationships between them, from complex datasets, while they are able to effectively predict future outcomes of a cancer type.

Given the significance of personalized medicine and the growing trend on the application of ML techniques, we here present a review of studies that make use of these methods regarding the cancer prediction and prognosis. In these studies, prognostic and predictive features are considered which may be independent of a certain treatment or are integrated in order to guide therapy for cancer patients, respectively. In addition, we discuss the types of ML methods being used, the types of data they integrate, the overall performance of each proposed scheme while we also discuss their pros and cons.

An obvious trend in the proposed works includes the integration of mixed data, such as clinical and genomic. However, a common problem that we noticed in several works is the lack of external validation or testing regarding the predictive performance of their models. It is clear that the application of ML methods could improve the accuracy of cancer susceptibility, recurrence and survival prediction. The accuracy of cancer prediction outcome has significantly improved by 15%–20% the last years, with the application of ML techniques.

Several studies have been reported in the literature and are based on different strategies that could enable the early cancer diagnosis and prognosis. Specifically, these studies describe approaches related to the profiling of circulating miRNAs that have been proven a promising class for cancer detection and identification. However, these methods suffer from low sensitivity regarding their use in screening at early stages and their difficulty to discriminate benign from malignant tumors. Various aspects regarding the prediction of cancer outcome based on gene expression signatures are discussed. These studies list the potential as well as the limitations of microarrays for the prediction of cancer outcome. Even though gene signatures could significantly improve our ability for prognosis in cancer patients, poor progress has been made for their application in the clinics. However, before gene expression profiling can be used in clinical practice, studies with larger data samples and more adequate validation are needed.

In the present work only studies that employed ML techniques for modeling cancer diagnosis and prognosis are presented.

2.2 TECHNOLOGY USED

ML, a branch of Artificial Intelligence, relates the problem of learning from data samples to the general concept of inference [10], [11], [12]. Every learning process consists of two phases: (i) estimation of unknown dependencies in a system from a given dataset and (ii) use of estimated dependencies to predict new outputs of the system. ML has also been proven an interesting area in biomedical research with many applications, where an acceptable generalization is obtained by searching through an n-dimensional space for a given set of biological samples, using different techniques and algorithms [13]. There are two main common types of ML methods known as (i)

supervised learning and (ii) unsupervised learning. In supervised learning a labeled set of training data is used to estimate or map the input data to the desired output. In contrast, under the unsupervised learning methods no labeled examples are provided and there is no notion of the output during the learning process. As a result, it is up to the learning scheme/model to find patterns or discover the groups of the input data. In supervised learning this procedure can be thought as a classification problem. The task of classification refers to a learning process that categorizes the data into a set of finite classes. Two other common ML tasks are regression and clustering. In the case of regression problems, a learning function maps the data into a real-value variable. Subsequently, for each new sample the value of a predictive variable can be estimated, based on this process. Clustering is a common unsupervised task in which one tries to find the categories or clusters in order to describe the data items. Based on this process each new sample can be assigned to one of the identified clusters concerning the similar characteristics that they share.

Suppose for example that we have collected medical records relevant to breast cancer and we try to predict if a tumor is malignant or benign based on its size. The ML question would be referred to the estimation of the probability that the tumor is malignant or no (1 = Yes, 0 = No). Fig. 1 depicts the classification process of a tumor being malignant or not. The circled records depict any misclassification of the type of a tumor produced by the procedure.

2.3 REQUIREMENTS

FUNCTIONAL REQUIREMENTS:-

Functional requirements are the functions or features that must be included in any system to satisfy the business needs and be acceptable to users. Based on this, the functional requirements that the system must require are as follows: -

- The system should be able to generate approximately 80% accurate diagnostic result
- The system should be able to analyse the data and make a good prediction.
- The system should be able to simplify the early raw input data so that it could be preprocessed and mined further.

NON – FUNCTIONAL REQUIREMENTS: -

Non-Functional requirements are a description of features, characteristics and attributes of the system as well as any constraint that may limit the boundaries of the proposed system. The Non-Functional requirements are essentially based on the performance, information, economy, control and security efficiency and services. Based on this, the non-functional requirements are as follows: -

• Performance Requirements: -As for this prototype version we will keep on detecting if the system is crashed, hanged or an operating system error has occurred. Also detecting the performance of the system in terms of efficiency of the integration of different components

• Safety Requirements: - For the safety requirements nothing but an operation of weekly backups for the database should take place

• Security and Privacy Requirements: -There are no specific security requirements.

• Software Quality Attributes: -

Reliability – The solution should provide reliability to the user that the product will run all the features mentioned in this document are perfectly available and executing perfectly. It should be tested and debugged completely. All exceptions should be well handled.

SYSTEM REQUIREMENTS: -

- Software Requirements: -
    1) Python 3.x
    2) Google Colab or Jupyter Notebook
    3) Operating System
    4) Anaconda
- Hardware Requirements: -
    1) Core i5/i7 processor
    2) At least 8GB RAM
    3) At least 60 GB of Usable Hard Disk Space

2.4 DESIGN

When predicting a result for cancer one needs to be as accurate as possible. A slight carelessness on diagnosis can lead to the administer of an inappropriate treatment that could worsen the health state of the person. In such a case one cannot rely on one algorithm to predict the better result, therefore multiple algorithms are needed to find the better result and the Ensembling method is required to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.
For our Case Study, we have the raw text data from the Memorial Sloan Kettering Cancer Center (MSKCC) containing an expert-annotated knowledge base where world-class researchers and oncologists have manually annotated thousands of mutations.
Our Machine Learning model will take that raw text data as the primary input and apply data preprocessing and mining to separate Gene, Variation and Class from the data. So that our Machine Learning model can classify them and thereby predict the class to which the variation belongs. The result can be interpreted by the Oncologist present at hand who can interpret the result and formulate the conclusion for diagnosis.

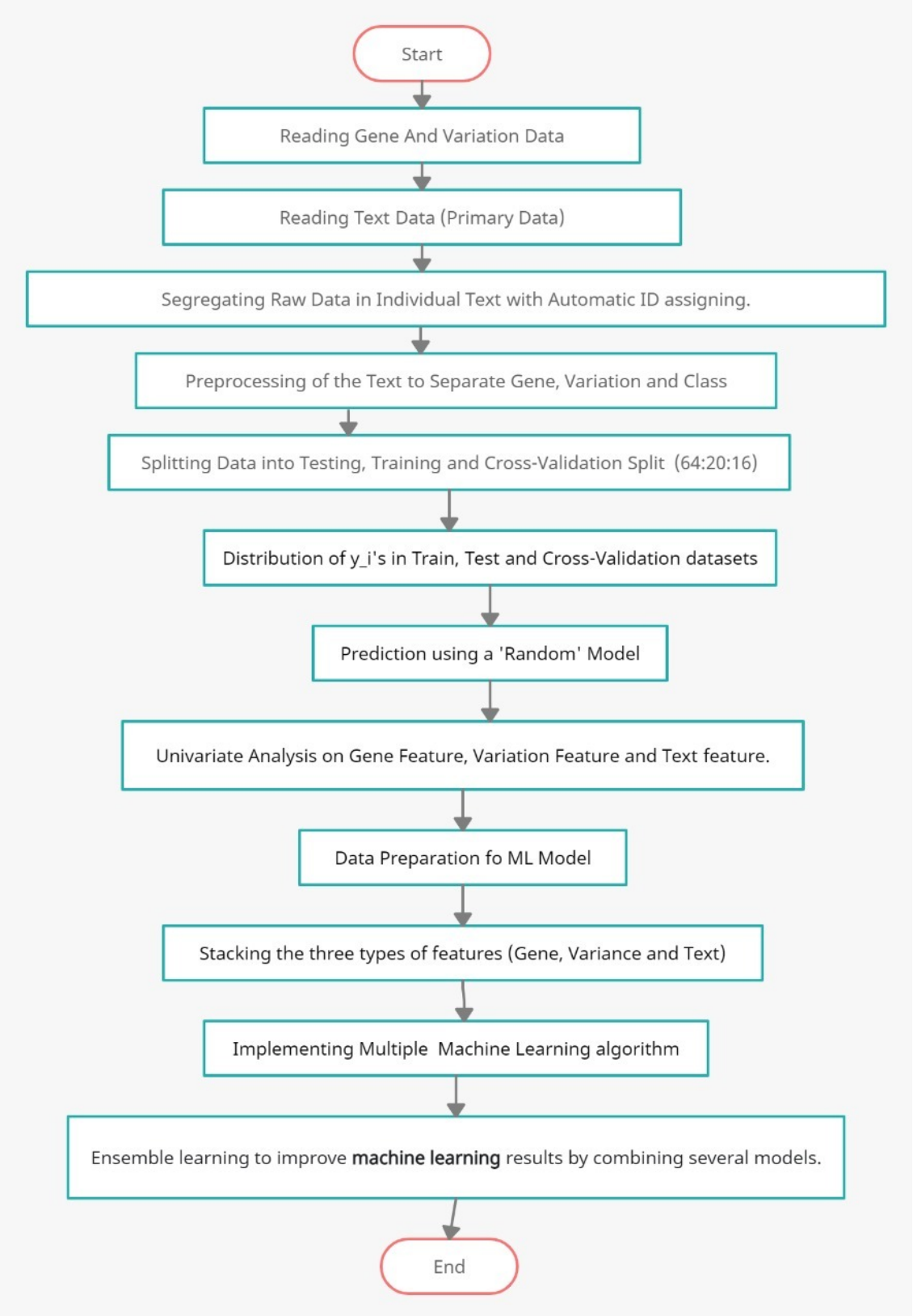


Fig2. Flowchart for Design of our case study

The Design flow of the Model is :

1. Setting Fields on which the data operation will be performed
   which are as mentioned below:

- **ID :** the id of the row used to link the mutation to the clinical evidence
- **Gene :** the gene where this genetic mutation is located
- **Variation :** the aminoacid change for this mutations
- **Class :** 1-9 the class this genetic mutation has been classified on

2. Reading the raw data *training_text.txt*

```
Number of data points :  3321
Number of features :  2
Features :  ['ID' 'TEXT']
```

3. Extracting a Proper Text and automatically assigning ID to it.

	ID	*TEXT*
0	*0*	*Cyclin-dependent kinases (CDKs) regulate a var...*
1	*1*	*Abstract Background Non-small cell lung canc...*
2	*2*	*Abstract Background Non-small cell lung canc...*
3	*3*	*Recent evidence has demonstrated that acquired...*
4	*4*	*Oncogenic mutations in the monomeric Casitas B..*

4. Preprocessing of the Text to Separate Gene, Variation & Class from the Text.
This ensures to remove all the stop word and redundant element in data set. Ensuring we have the most accurate data to work on.

ID	Gene	Variation	Class	TEXT	
0	0	FAM58A	Truncating Mutations	1	cyclin dependent kinases cdks regulate variety...
1	1	CBL	W802*	2	abstract background non small cell lung cancer...
2	2	CBL	Q249E	2	abstract background non small cell lung cancer...
3	3	CBL	N454D	3	recent evidence demonstrated acquired uniparen...
4	4	CBL	L399V	4	oncogenic mutations monomeric casitas b lineag...

5. Splitting data into train, test and cross validation (64:20:16)

We split the data into train, test and cross validation data sets, preserving the ratio of class distribution in the original data set.

```
Number of data points in train data: 2124
Number of data points in test data: 665
Number of data points in cross validation data: 532
```

6. Distribution of y_i's in Train, Test and Cross Validation datasets

This returns a dict, keys as class labels and values as the number of data points in that class.

The graphs below shows 9 class Variation in Gene and the data Distributed accordingly in training, testing and cross validation Respectively.

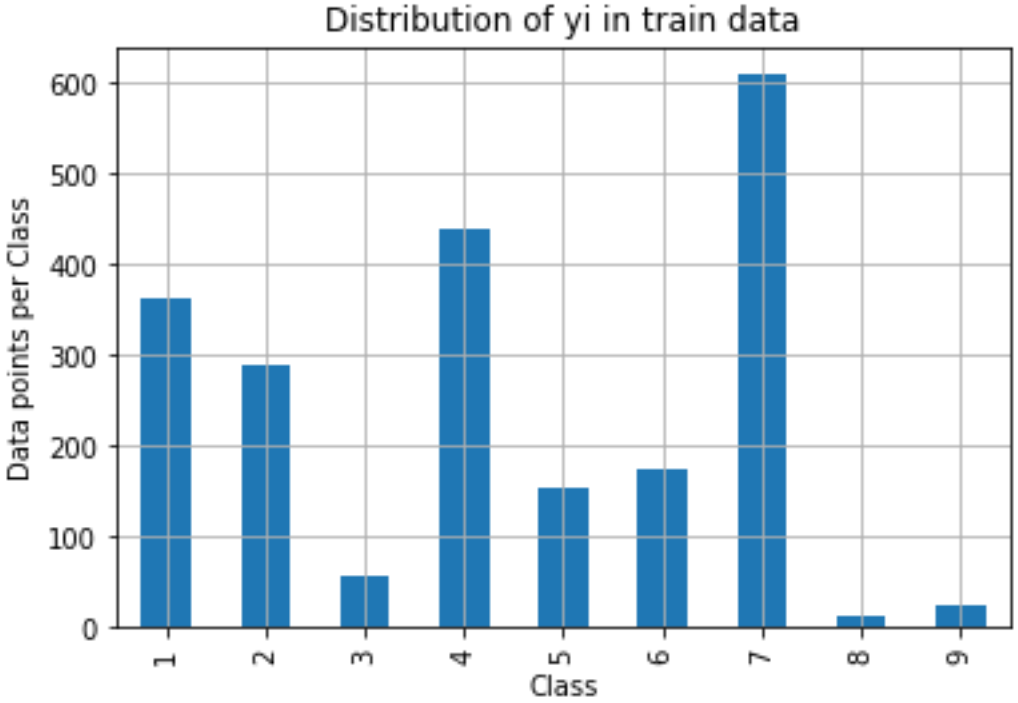


Fig.3 Data in training

```
Number of data points in class 7 : 609 ( 28.672 %)
Number of data points in class 4 : 439 ( 20.669 %)
Number of data points in class 1 : 363 ( 17.09 %)
Number of data points in class 2 : 289 ( 13.606 %)
Number of data points in class 6 : 176 ( 8.286 %)
Number of data points in class 5 : 155 ( 7.298 %)
Number of data points in class 3 : 57 ( 2.684 %)
Number of data points in class 9 : 24 ( 1.13 %)
Number of data points in class 8 : 12 ( 0.565 %)
```

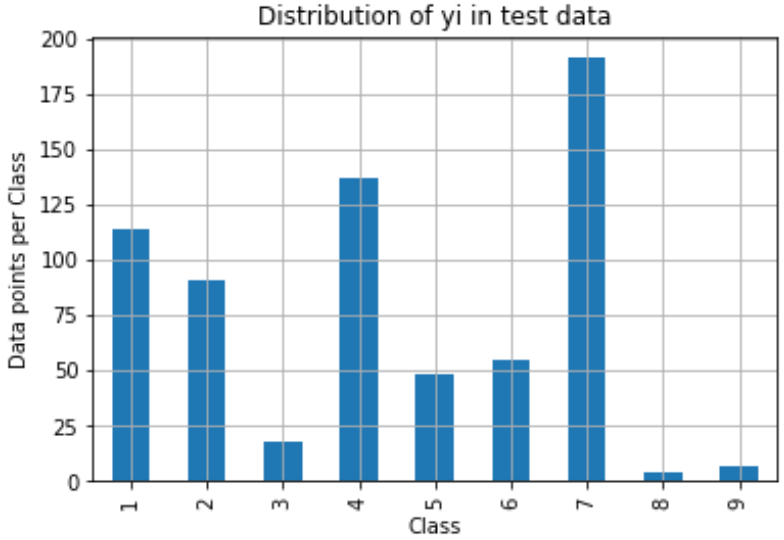


Fig.4 Data in Testing

```
Number of data points in class 7 : 191 ( 28.722 %)
Number of data points in class 4 : 137 ( 20.602 %)
Number of data points in class 1 : 114 ( 17.143 %)
Number of data points in class 2 : 91 ( 13.684 %)
Number of data points in class 6 : 55 ( 8.271 %)
Number of data points in class 5 : 48 ( 7.218 %)
Number of data points in class 3 : 18 ( 2.707 %)
Number of data points in class 9 : 7 ( 1.053 %)
Number of data points in class 8 : 4 ( 0.602 %)
```

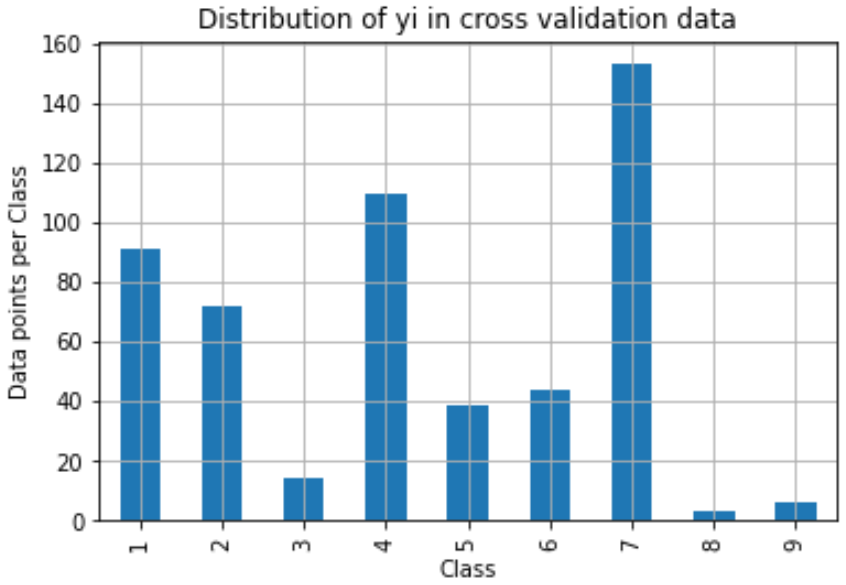


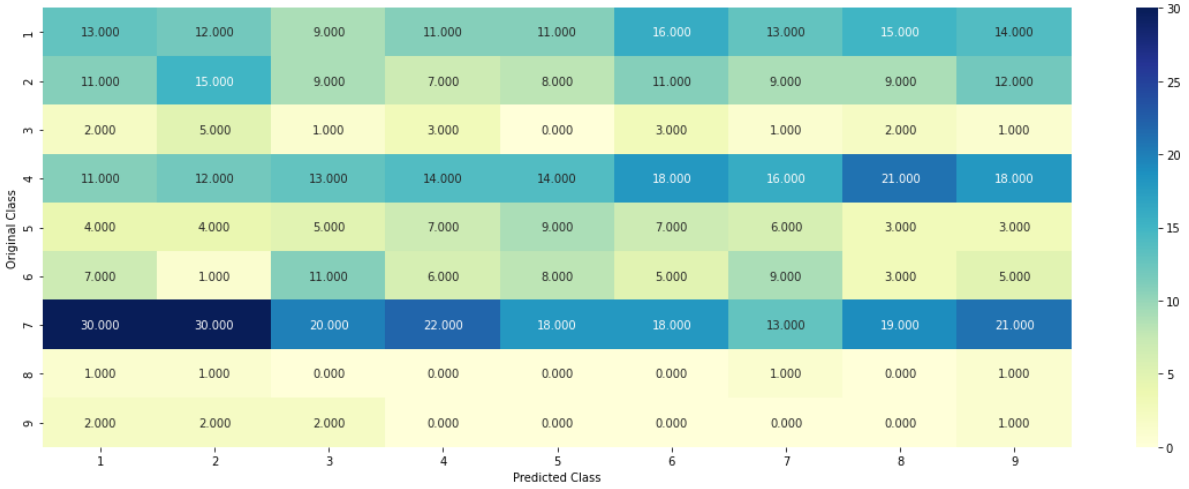
Fig.5 Data In Cross Validation

```
Number of data points in class 7 : 153 ( 28.759 %)
Number of data points in class 4 : 110 ( 20.677 %)
Number of data points in class 1 : 91 ( 17.105 %)
Number of data points in class 2 : 72 ( 13.534 %)
Number of data points in class 6 : 44 ( 8.271 %)
Number of data points in class 5 : 39 ( 7.331 %)
Number of data points in class 3 : 14 ( 2.632 %)
Number of data points in class 9 : 6 ( 1.128 %)
Number of data points in class 8 : 3 ( 0.564 %)
```

7. Prediction using a 'Random' Model

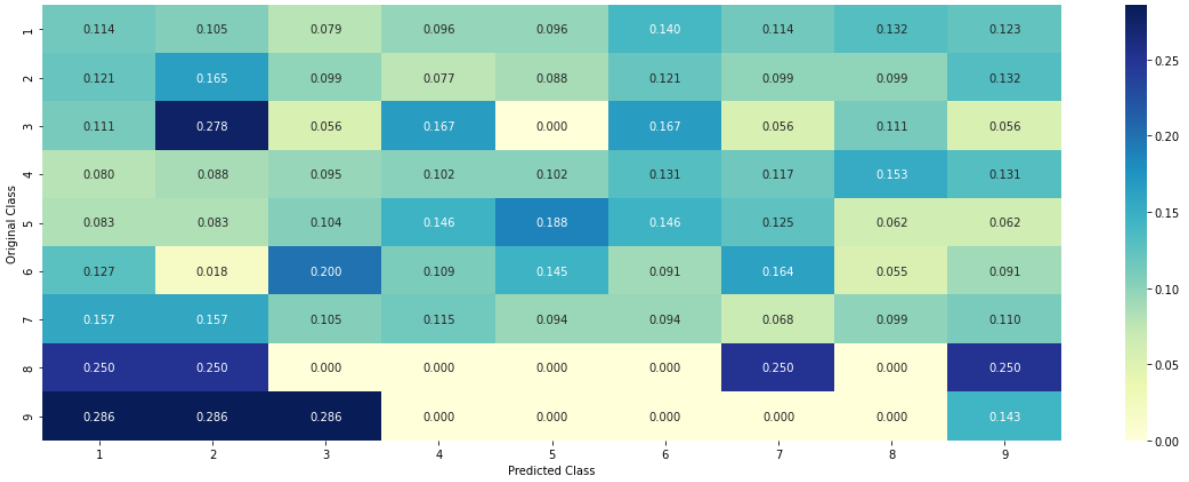
In a 'Random' Model, we generate the NINE class probabilities
randomly such that they sum to 1.
This model randomly generates points for prediction.
We generate 9 numbers and the sum of numbers should be 1.
one solution is to generate 9 numbers and divide each of the numbers by their sum.
we create a output array that has exactly same size as the CV data.
we create a output array that has exactly same as the test data

```
Log loss on Cross Validation Data using Random Model 2.5421750872938165
Log loss on Test Data using Random Model 2.4439151610259313
-------------------- Confusion matrix --------------------
```



---------- Precision matrix (Columm Sum=1) ----------

---------- Recall matrix (Row sum=1) ----------



8. Univariant Analysis on Gene Feature, Variation Feature and Text feature.

Univariate analysis is the simplest form of analyzing data. “Uni” means “one”, so in other words your data has only one variable. It doesn't deal with causes or relationships (unlike regression) and it's major purpose is to describe; It takes data, summarizes that data and finds patterns in the data.

We apply this on our training variant database.

Here, we featurized Gene, Variation and Text. For Categorizing them accordingly

A. GENE as categorical variable

Q. How many categories are there and How they are distributed?

```
Number of Unique Genes : 232
BRCA1      176
TP53       102
PTEN        81
EGFR        80
BRCA2       78
KIT         65
BRAF        64
ALK         49
ERBB2       46
PDGFRA      41
Name: Gene, dtype: int64
```

There are 232 different categories of genes in the train data, and they are distibuted as follows

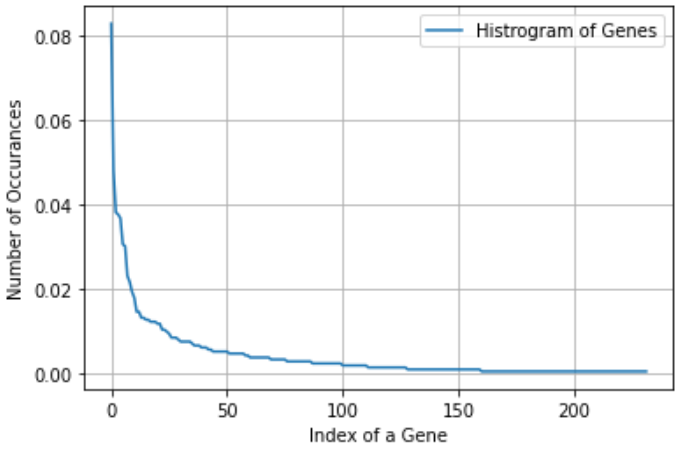


Fig 6. Histogram of genes

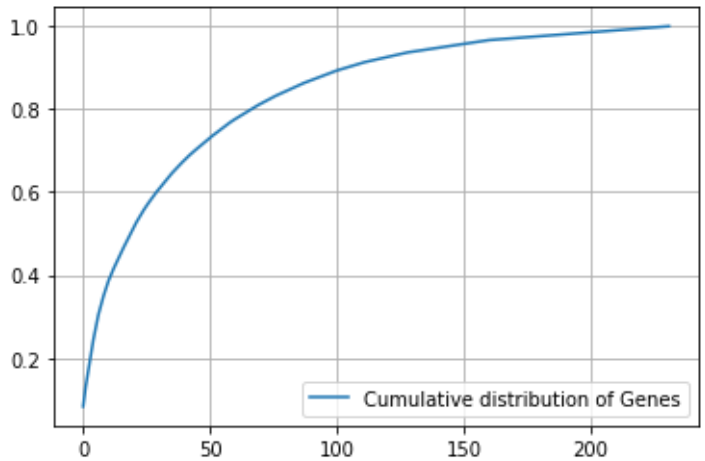


Fig 7, Cumulative Distribution of Genes

We will choose the appropriate featurization based on the ML model we use. For this problem of multi-class classification with categorical features, one-hot encoding is better for Logistic regression while response coding is better for Random Forests.

```
train_gene_feature_responseCoding is converted feature using respone c
oding method. The shape of gene feature: (2124, 9)
146      EGFR
745     ERBB2
```

```
1736     MSH2
2110      B2M
2956     GNAS
Name: Gene, dtype: object
train_gene_feature_onehotCoding is converted feature using one-hot enc
oding method. The shape of gene feature: (2124, 231)
```

There are many ways to estimate how good a feature is, in predicting y_i. One of the good methods is to build a proper ML model using just this feature. In this case, we will build a logistic regression model using only Gene feature (one hot encoded) to predict y_i.

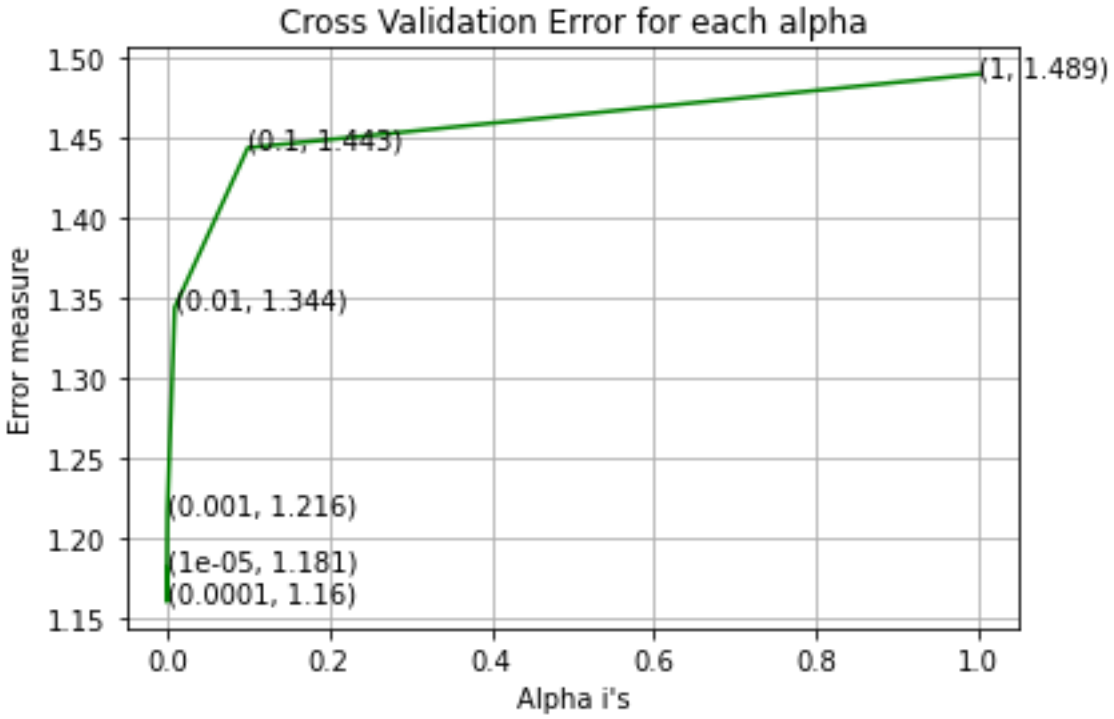


Fig 8. C-V Error for each alpha Gene as Feature

```
For values of best alpha =  0.0001 The train log loss is: 0.9961740987
606205
For values of best alpha =  0.0001 The cross validation log loss is: 1
.1604745933653107
For values of best alpha =  0.0001 The test log loss is: 1.18919835714
01996
```

This procedure checks the stability of the Gene Feature across all the data sets.
It comes out as Stable or else the CV and Test errors would be significantly more than train error, if it wasn't stable.

B. Variation as a categorical variable

Q. How many categories are there?

```
Number of Unique Variations : 1920
Truncating_Mutations    58
Deletion                53
Amplification           50
Fusions                 22
Q61L                     3
T58I                     3
Overexpression           2
EWSR1-ETV1_Fusion        2
F28L                     2
G35R                     2
```

```
Name: Variation, dtype: int64
```
There are 1920 different categories of variations in the train data, and they are distributed as follows.

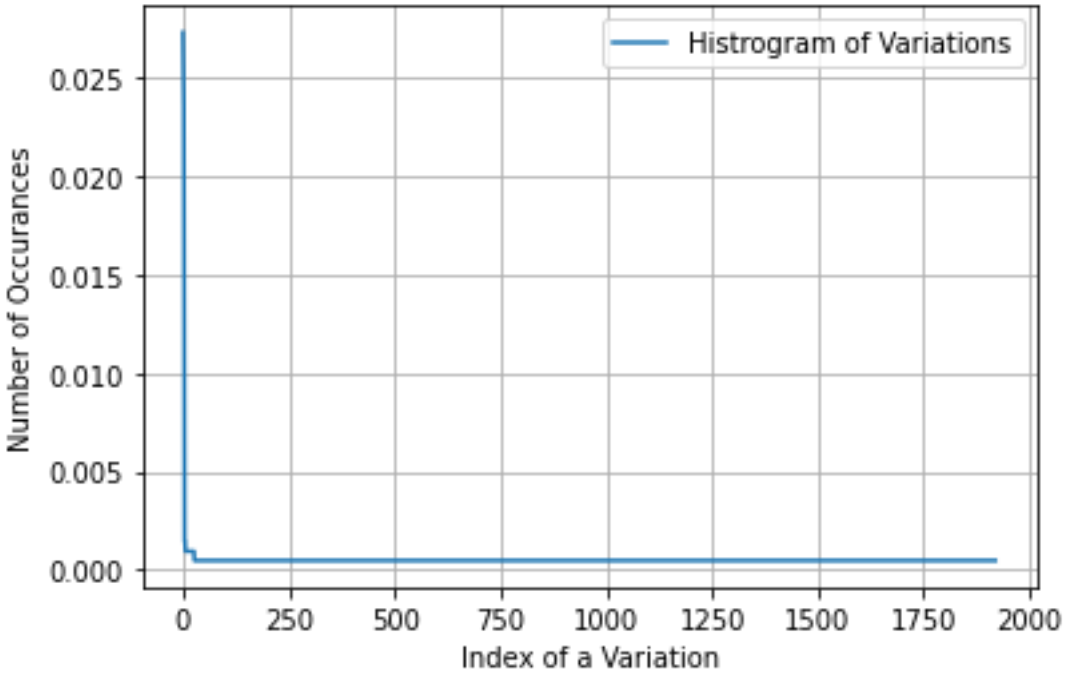


Fig 9. Histogram of Variation

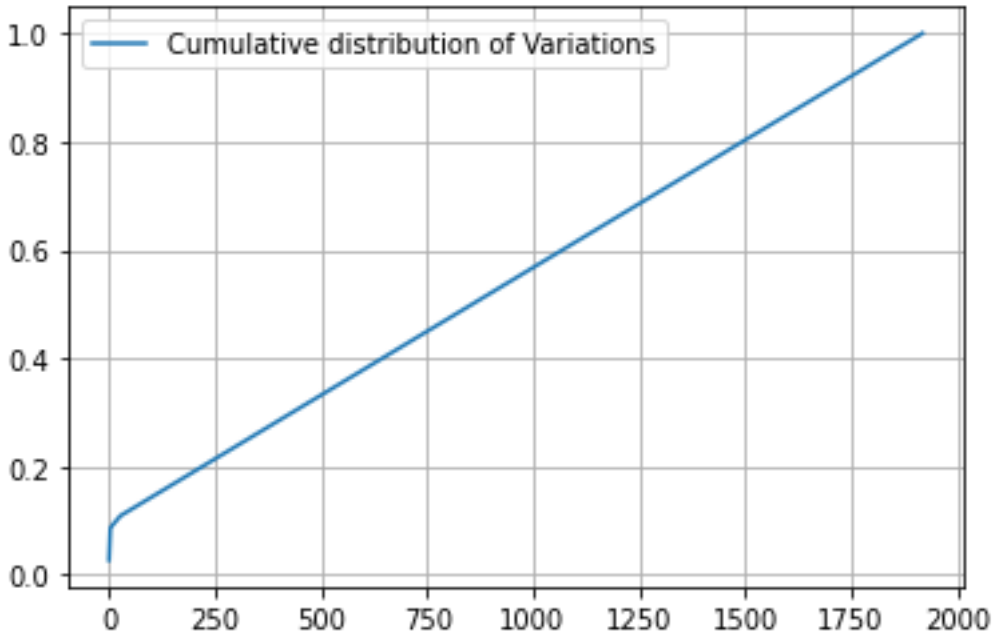


Fig 10. Cumulative distribution of Variations

We again featurize it using:

1. One hot Encoding
2. Response coding

for the different ML algorithm we will use,

```
train_variation_feature_responseCoding is a converted feature using th
e response coding method. The shape of Variation feature: (2124, 9)
```

```
train_variation_feature_onehotEncoded is converted feature using the o
nne-hot encoding method. The shape of Variation feature: (2124, 1950)
```

Variation feature in predicting y_i:

We build a model like before to check the stability of the feature across the data set.

```
For values of best alpha =  0.0001 The train log loss is: 0.6699481195
548089
For values of best alpha =  0.0001 The cross validation log loss is: 1
.6976576222146533
For values of best alpha =  0.0001 The test log loss is: 1.71621954491
19637
```

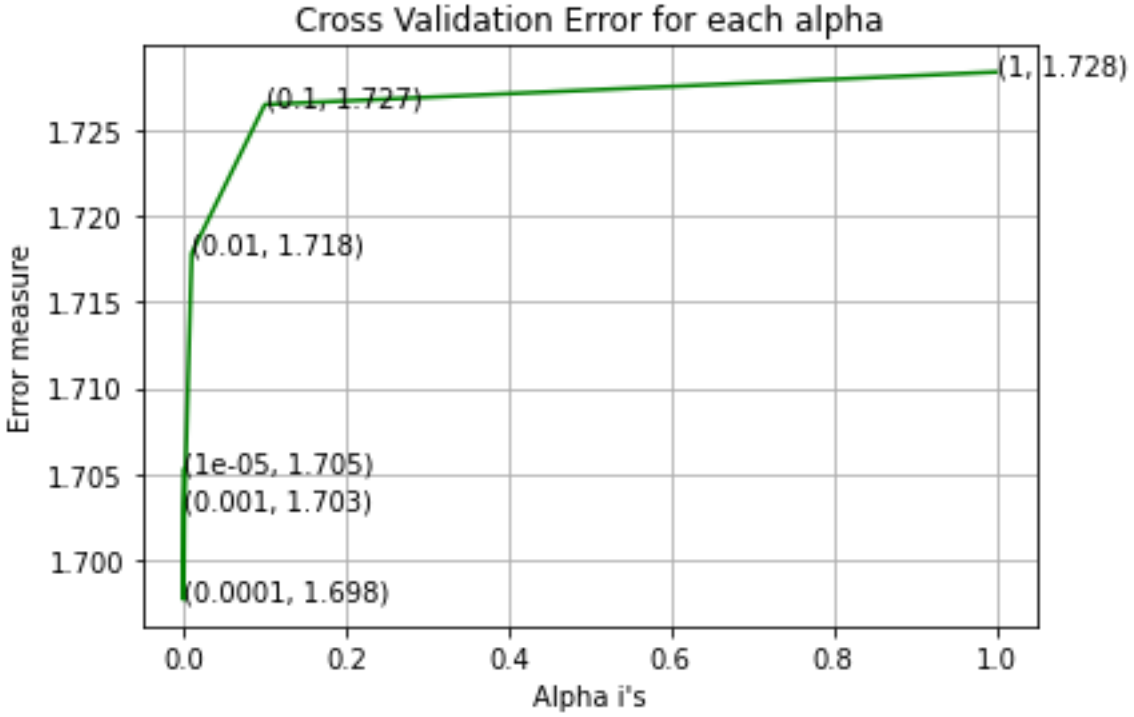


Fig 11. CV Error for each alpha Variation as Feature.

Variation is not a stable feature to use as:

```
1. In test data 68 out of 665 : 10.225563909774436
2. In cross validation data 47 out of  532 : 8.834586466165414
```

C. Text as a feature.

We repeat the same procedure from the above two Gene and variation as feature.

Q. How many unique words are in the training data set.

```
Total number of unique words in train data : 52988
```

We again apply response coding and one hot coding on the feature to adjust it as per our ML model.

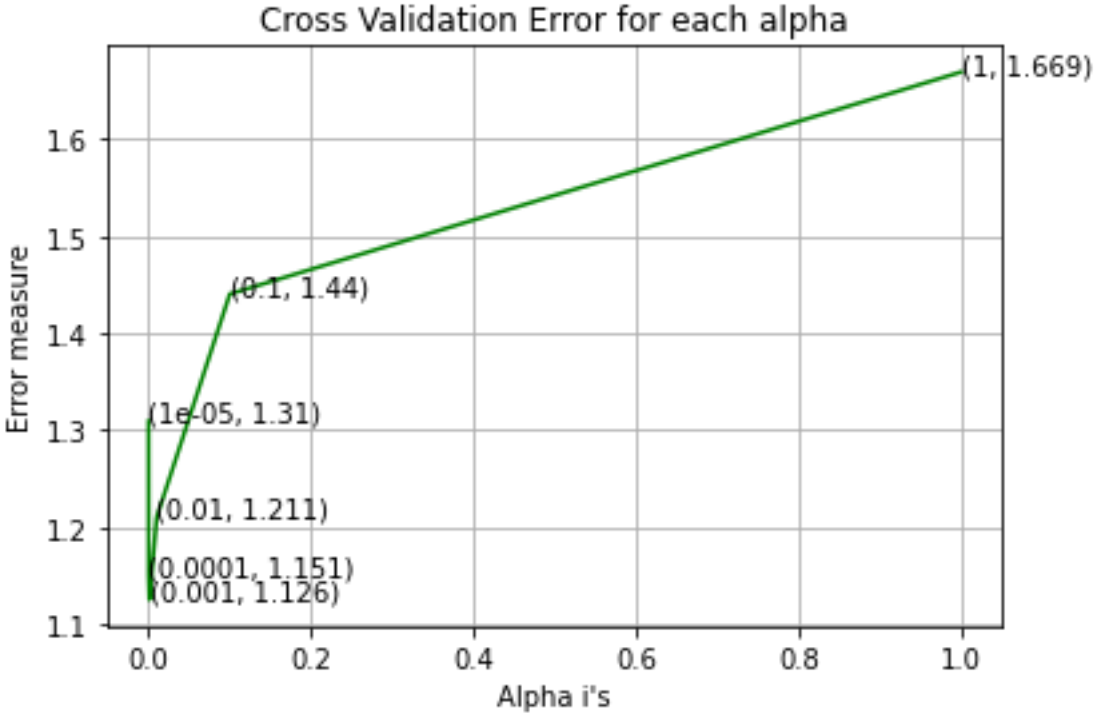


Fig 12. CV error for each alpha Text as feature

```
For values of best alpha =  0.001 The train log loss is: 0.67576530522
22209
For values of best alpha =  0.001 The cross validation log loss is: 1.
1257509731524722
For values of best alpha =  0.001 The test log loss is: 1.072780878164
5205
```

On checking the stability of Text as feature:

```
96.831 % of word of test data appeared in train data
97.878 % of word of Cross Validation appeared in train data
```

Therefore, Yes Text is a stable feature.

9. Stacking the features and Preparation of data for ML algorithms.

The next step involves in preparing data for ML algorithms that will plot the confusion Matrix, report the log loss and we will check whether the feature present in the test point text or not.

After that we are merging all 3 features merging gene, variance and text features, using stacking to improve the performance for predicting.

This is done by One hot Encoding the features and response encoding it:

```
One hot encoding features :
(number of data points * number of features) in train data =  (2124, 5
5169)
(number of data points * number of features) in test data =  (665, 551
69)
(number of data points * number of features) in cross validation data
= (532, 55169)


Response encoding features :
(number of data points * number of features) in train data =  (2124, 2
7)
(number of data points * number of features) in test data =  (665, 27)
```

```
(number of data points * number of features) in cross validation data
= (532, 27)
```

10. Implementing multiple algorithm and Ensemble learning.
This is explained more in detail in chapter 3.
The data-set prepared for the ML models are run by multiple algorithms. Giving out each of their prediction with log Loss report.
The Algorithms being:

- Naïve Bayes
- K Nearest Neighbor Classification
- Logistic Regression
- Linear Support Vector Machines
- Random Forest Classifier

And Finally Ensemble Learning Methods:

- Stacking Classifier
- Maximum Voting Classifier

are run to improve the Prediction accuracy for the model

3. IMPLEMENTATION

3.1 Algorithm Used

A) Naïve Bayes

Naive Bayes classifier for multinomial models

The multinomial I Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.

For classication using the 'Naive Bayes' Algorithm, we made use of:

1. Hyper parameter tuning to find the best parameters for the I Bayes Model.

Parameter: alpha = [0.00001, 0.0001, 0.001, 0.1, 1, 10, 100,1000]

Result:

for alpha = 1e-05
Log Loss : 1.2151111970450392
for alpha = 0.0001
Log Loss : 1.2106441761107478
for alpha = 0.001
Log Loss : 1.214516476883617
for alpha = 0.1
Log Loss : 1.2063521440346674
for alpha = 1
Log Loss : 1.2636301187204109
for alpha = 10
Log Loss : 1.372785671900146
for alpha = 100
Log Loss : 1.3578296195572752
for alpha = 1000
Log Loss : 1.3220932996346821

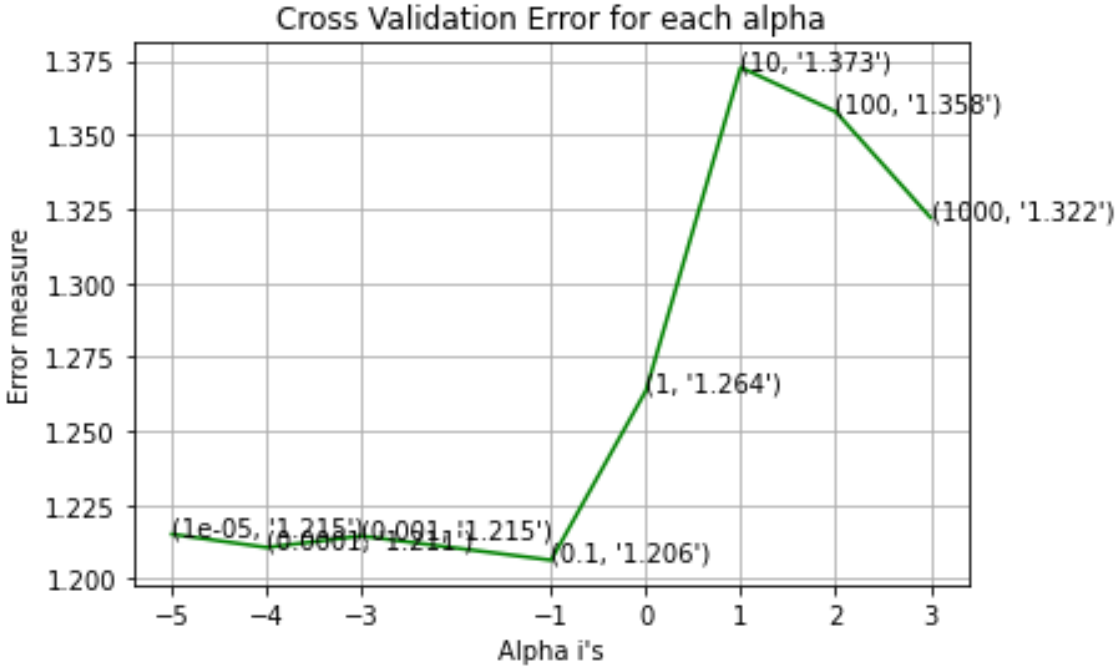


Fig. 13: Graph for finding best alpha Naïve Bayes

We found out the best value of alpha is 0.1 with the minimum log loss of 1.20635.
We used alpha=0.1 on 3 different data set i.e. train, cross-validation and test.
The results are as follows:
For values of best alpha = 0.1 The train log loss is: 0.8798370221316937
For values of best alpha = 0.1 The cross validation log loss is: 1.2063521440346674
For values of best alpha = 0.1 The test log loss is: 1.2379809338253713

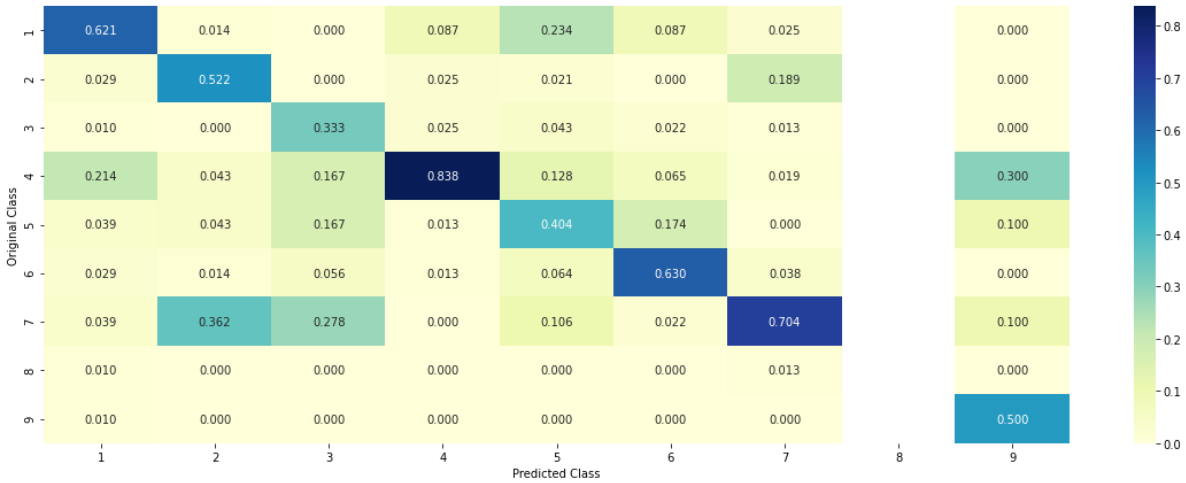
2. Training:

We trained the model on the best alpha parameters and here are the results:
Log Loss : 1.2063521440346674

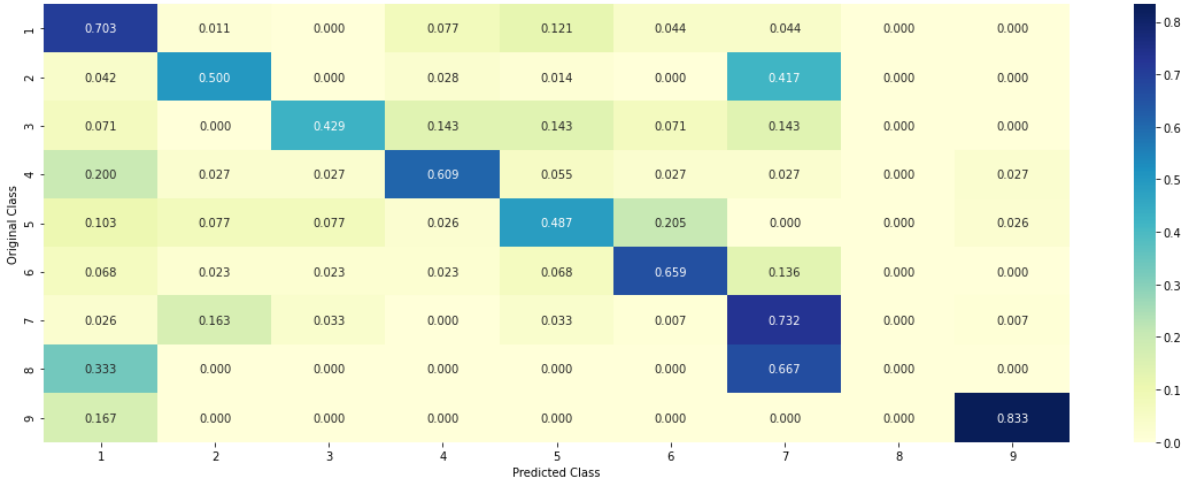
Number of misclassified point: 0.36466165413533835

---------- Confusion matrix ----------

---------- Precision matrix (Columm Sum=1) ----------



---------- Recall matrix (Row sum=1) ----------



3. Feature Importance, Correctly and incorrectly classified point

Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction

We tested the feature on test point index 1 and 100, for correctly and incorrectly classified points respectively:
The results for correctly classified points:
Predicted Class: 7

Predicted Class Probabilities: [[0.0777 0.0741 0.0116 0.1106 0.0349 0.0359 0.6463 0.0045 0.0043]]

Actual Class: 7

Out of the top 100 features 0 are present in query point

The results for incorrectly classified points:

Predicted Class: 7

Predicted Class Probabilities: [[0.0805 0.1593 0.0121 0.114 0.0359 0.0368 0.5522 0.0047 0.0045]]

Actual Class: 7

Out of the top 100 features 0 are present in query point

B) K Nearest Neighbor Classification

1. Hyper parameter tuning to find the best parameters for the 'K Nearest Neighbor Model'.
Parameter: alpha = alpha = [5, 11, 15, 21, 31, 41, 51, 99]
Result:
for alpha = 5
Log Loss : 1.0137121058888419
for alpha = 11
Log Loss : 1.0262018397608146
for alpha = 15
Log Loss : 1.0199121491963061
for alpha = 21
Log Loss : 1.0217029694043598
for alpha = 31
Log Loss : 1.033938745599289
for alpha = 41
Log Loss : 1.039449967630178

for alpha = 51
Log Loss : 1.0514281881103282
for alpha = 99
Log Loss : 1.0479089013904614

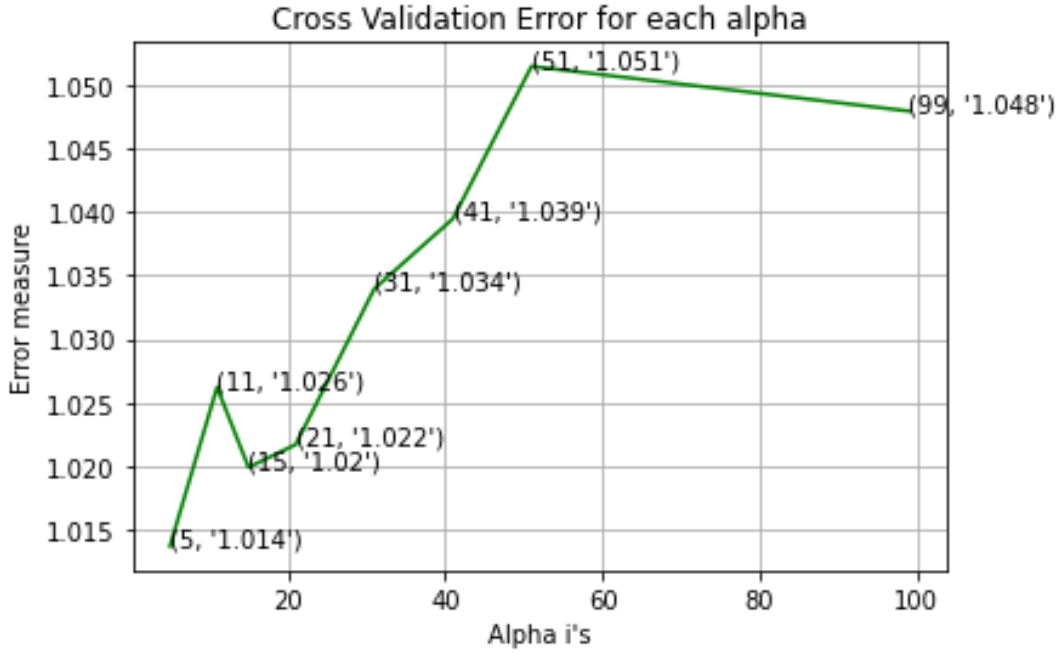
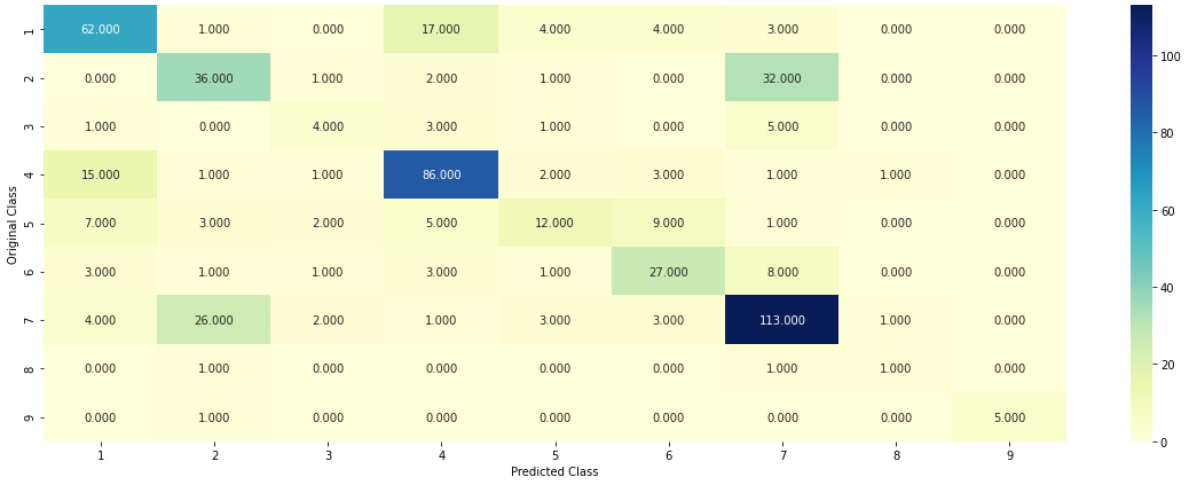


Fig.14: Graph for finding best alpha K-Nearest neighbour

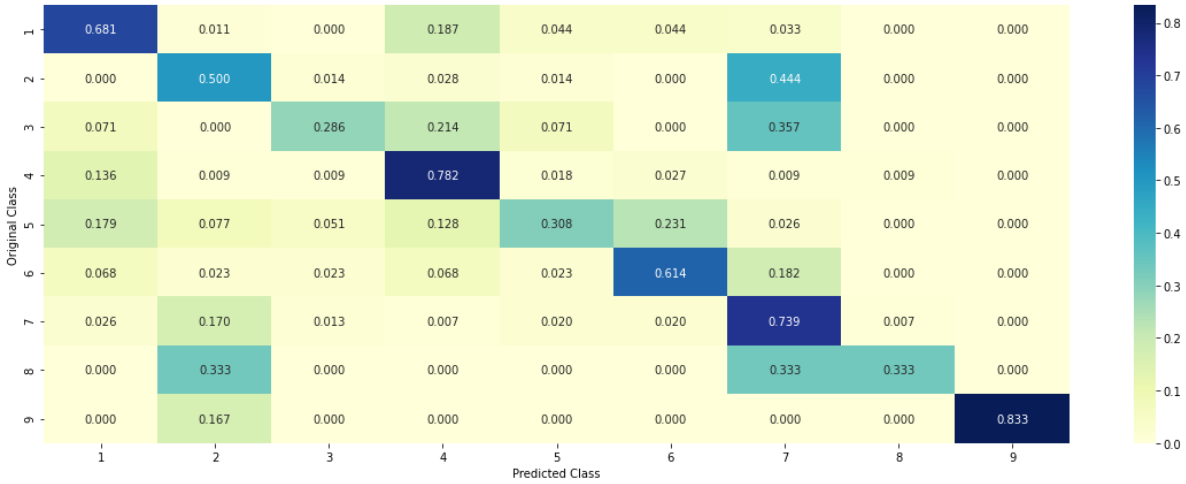
We found out the best value of alpha is 5 with the minimum log loss of 1.0137121058888419.
We used alpha=5 on 3 different data set i.e. train, cross-validation and test.
The results are as follows:
For values of best alpha = 5 The train log loss is: 0.4785148601133296
For values of best alpha = 5 The cross-validation log loss is: 1.0137121058888419
For values of best alpha = 5 The test log loss is: 1.0227274419805317

2. Training:
We trained the model on the best alpha parameters and here are the results:
Log loss : 1.0137121058888419
Number of mis-classified points : 0.34962406015037595
---------- Confusion matrix ----------



---------- Precision matrix (Column Sum=1) ----------

---------- Recall matrix (Row sum=1) ----------



3. Testing of the model

We tested the feature on test point index 1 and 100, for Sample Query Point-1 and 2 respectively:

The results for Sample Query point -1:

Predicted Class: 7

Actual Class: 7

The 5 nearest neighbors of the test points belong to classes [7 2 7 2 7]

Frequency of nearest points: Counter ({7: 3, 2: 2})

The results for Sample Query Point-2:

Predicted Class: 7

Actual Class: 7

the k value for ken is 5 and the nearest neighbors of the test points belongs to classes [7 7 7 7 2]

Frequency of nearest points: Counter ({7: 4, 2: 1})

C) Logistic Regression

Linear classifiers (SVM, logistic regression, etc.) with SGD training.

This estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate). SGD allows minibatch (online/out-of-core) learning via the partial_fit method. For best results using the default learning rate schedule, the data should have zero mean and unit variance.

This implementation works with data represented as dense or sparse arrays of floating point values for the features. The model it fits can be controlled with the loss parameter; by default, it fits a linear support vector machine (SVM).

The regularizer is a penalty added to the loss function that shrinks model parameters towards the zero-vector using either the squared 26hosphor26 norm L2 or the absolute norm L1 or a combination of both (Elastic Net). If the parameter update crosses the 0.0 value because of the regularizer, the update is truncated to 0.0 to allow for learning sparse models and achieve online feature selection.

Class Balancing:
Most real-world classification problems display some level of class imbalance, which is when each class does not make up an equal portion of your data-set. It is important to properly adjust your metrics and methods to adjust for your goals. If this is not done, you may end up optimizing for a meaningless metric in the context of your use case.

For example: Suppose you have two classes — A and B. Class A is 90% of your data-set and class B is the other 10%, but you are most interested in identifying instances of class B. You can reach an accuracy of 90% by simply predicting class A every time, but this provides a useless classifier for your intended use case. Instead, a properly calibrated method may achieve a lower accuracy, but would have a substantially higher true positive rate (or recall), which is really the metric you should have been optimizing for. These scenarios often occur in the context of detection, such as for abusive content online, or disease markers in medical data.

WITH CLASS BALANCING:

1. Hyper parameter tuning to find the best parameters for the Logistic Regression Model.
Parameter: alpha = [10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^{2}, 10^{3}]
Result:
for alpha = 1e-06
Log Loss: 1.2839090936006285
for alpha = 1e-05
Log Loss: 1.2574894726779975
for alpha = 0.0001
Log Loss: 1.0910533554946569
for alpha = 0.001
Log Loss: 1.0512204519237873
for alpha = 0.01
Log Loss: 1.126611029437376
for alpha = 0.1
Log Loss: 1.418985794517477
for alpha = 1
Log Loss: 1.6560157835482214
for alpha = 10
Log Loss: 1.6866767059501178
for alpha = 100
Log Loss: 1.6897991134633543

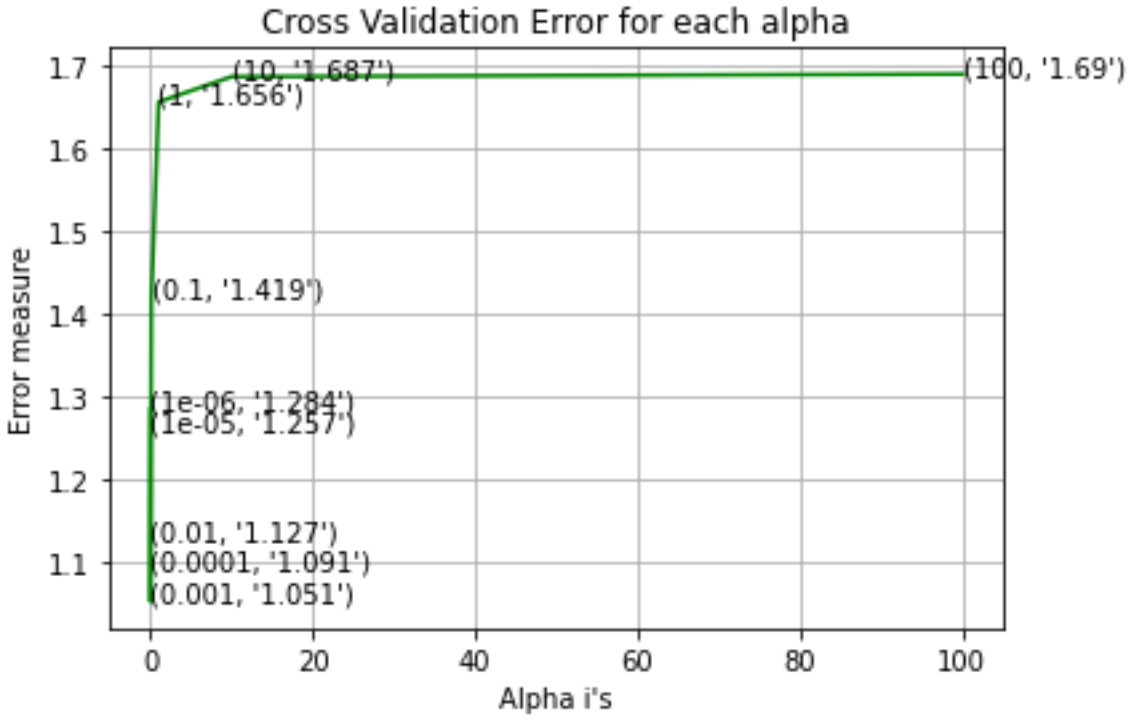


Fig. 15: Graph for finding best alpha Logistic Regression(Balanced Class)

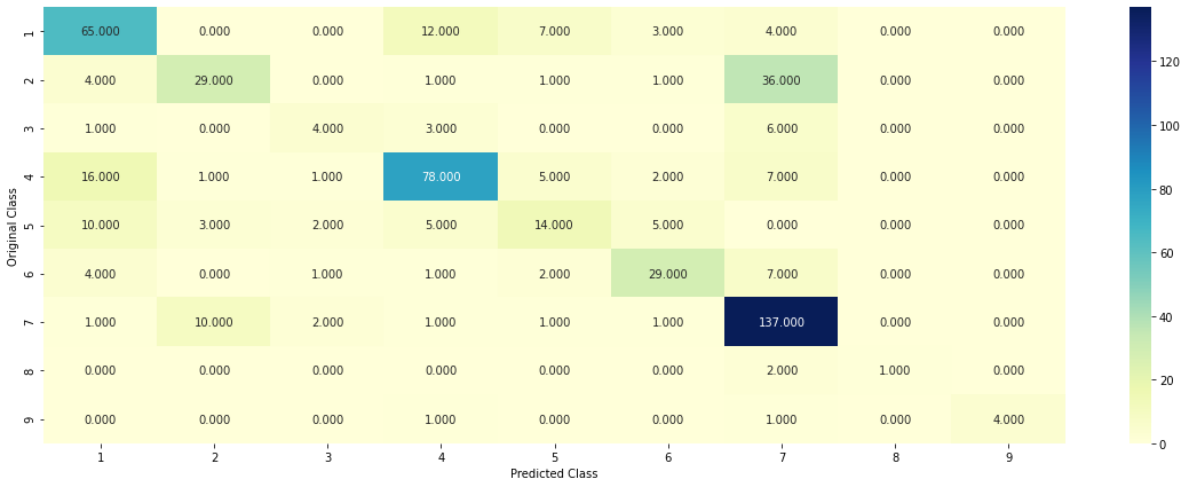
We found out the best value of alpha is 0.001 with the minimum log loss of 1.0512204519237873.
We used alpha=0.001 on 3 different data set i.e. train, cross-validation and test.
The results are as follows:
For values of best alpha = 0.001 The train log loss is: 0.5450433192307262
For values of best alpha = 0.001 The cross-validation log loss is: 1.0512204519237873
For values of best alpha = 0.001 The test log loss is: 1.027332286719511

2. Training:

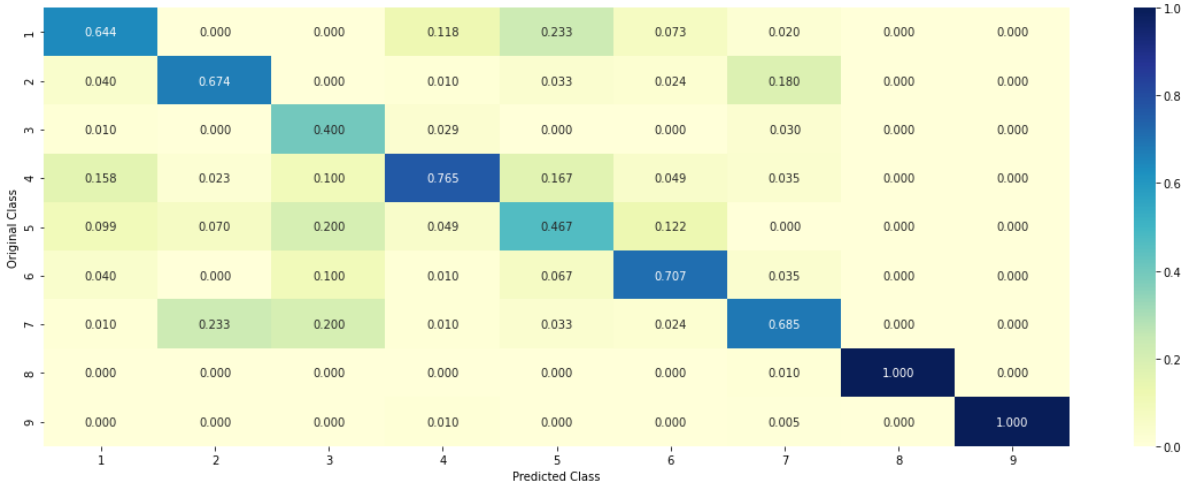
We trained the model on the best alpha parameters and here are the results:
Log loss: 1.0512204519237873

Number of mis-classified points: 0.32142857142857145

---------- Confusion matrix ----------



---------- Precision matrix (Columm Sum=1) ----------



---------- Recall matrix (Row sum=1) ----------

3. Feature Importance, Correctly and incorrectly classified point

Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction

We tested the feature on test point index 2 and 188, for correctly and incorrectly classified points respectively:
The results for correctly classified points:
Predicted Class: 7

Predicted Class: 4

Predicted Class Probabilities: [[0.1144 0.0607 0.0061 0.6029 0.0373 0.0249 0.1389 0.0077 0.0073]]

Actual Class: 1

---

168 Text features [suppressor] present in test data point [True]

229 Text features [ix] present in test data point [True]

467 Text features [kinase] present in test data point [True]
Out of the top 500 features 3 are present in query point.

The results for incorrectly classified points:
Predicted Class: 2

Predicted Class Probabilities: [[2.800e-03 8.070e-01 7.000e-04 3.700e-03 3.200e-03 1.600e-03 1.733e-01 4.900e-03 2.800e-03]]
Actual Class: 2

---

60 Text features [5v] present in test data point [True]
381 Text features [wild] present in test data point [True]
454 Text features [type] present in test data point [True]
474 Text features [50jc] present in test data point [True]
Out of the top 500 features 4 are present in query point

WITHOUT CLASS BALANCING:

3. Hyper parameter tuning to find the best parameters for the Logistic Regression Model.
Parameter: alpha = [10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10]
Result:
for alpha = 1e-06
Log Loss: 1.2623487052964188
for alpha = 1e-05
Log Loss: 1.219321805932207
for alpha = 0.0001
Log Loss: 1.0850708318603655
for alpha = 0.001
Log Loss: 1.0601486760104313
for alpha = 0.01
Log Loss: 1.1881319162042188
for alpha = 0.1
Log Loss: 1.3453381771086805
for alpha = 1
Log Loss: 1.5869164339643327

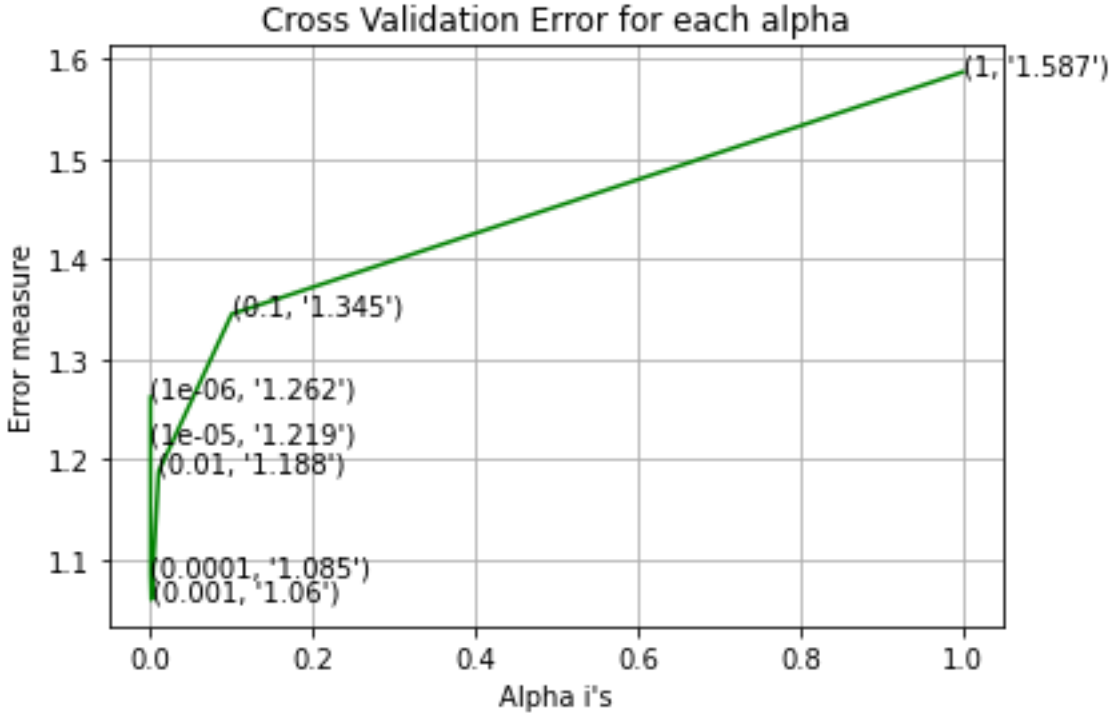


Fig. 16: Graph for finding best alpha Logistic Regression (Unbalanced Class)

We found out the best value of alpha is 0.001 with the minimum log loss of 1. 1.0601486760104313.
We used alpha=0.001 on 3 different data set i.e., train, cross-validation and test.

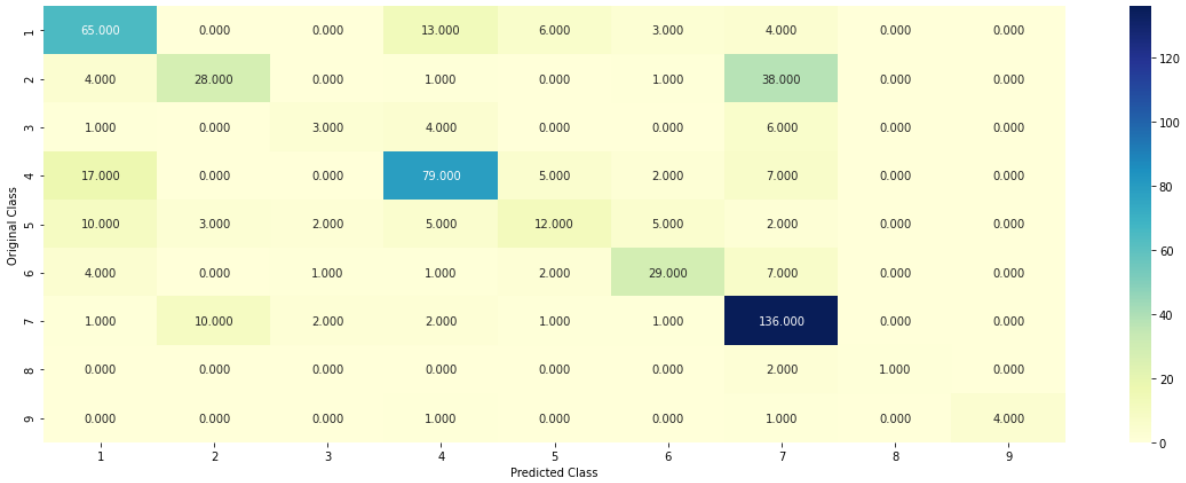
The results are as follows:
For values of best alpha = 0.001 The train log loss is: 0.537455848017654
For values of best alpha = 0.001 The cross-validation log loss is: 1.0601486760104313
For values of best alpha = 0.001 The test log loss is: 1.0612322986836904

2. Training:

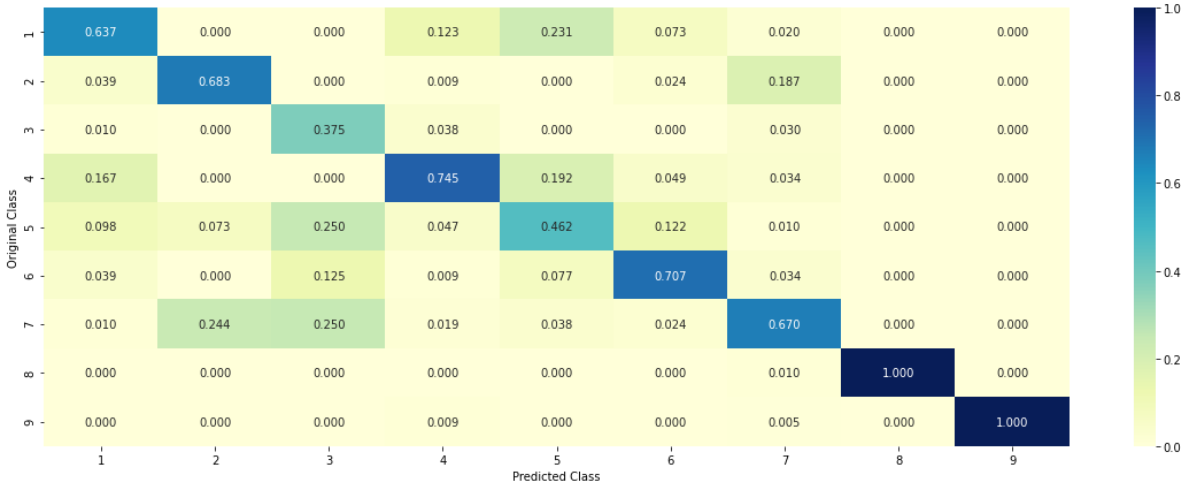
We trained the model on the best alpha parameters and here are the results:
Log loss: 1.0601486760104313

Number of mis-classified points: 0.32894736842105265

---------- Confusion matrix ----------



---------- Precision matrix (Columm Sum=1) ----------



---------- Recall matrix (Row sum=1) ----------

3. Feature Importance, Correctly and incorrectly classified point

Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction

We tested the feature on test point index 2 and 188, for correctly and incorrectly classified points respectively:

The results for correctly classified points:

Predicted Class: 7

Predicted Class Probabilities: [[0.0365 0.0724 0.0029 0.1036 0.0167 0.0161 0.7454 0.0048 0.0017]]

Actual Class: 7

---

176 Text features [constitutively] present in test data point [True]

263 Text features [transforming] present in test data point [True]

328 Text features [3t3] present in test data point [True]

346 Text features [downstream] present in test data point [True]

362 Text features [expressing] present in test data point [True]

369 Text features [rib] present in test data point [True]

376 Text features [activated] present in test data point [True]

440 Text features [phosphor] present in test data point [True]

442 Text features [murine] present in test data point [True]

453 Text features [Saudi] present in test data point [True]

489 Text features [oncogene] present in test data point [True]

493 Text features [missense] present in test data point [True]

Out of the top 500 features 12 are present in query point

The results for incorrectly classified points:

Predicted Class: 7

Predicted Class Probabilities: [[0.000e+00 3.369e-01 0.000e+00 0.000e+00 1.000e-04 0.000e+00 6.616e-01 1.400e-03 0.000e+00]]

Actual Class: 7

---

176 Text features [constitutively] present in test data point [True]

194 Text features [hki] present in test data point [True]
213 Text features [stat] present in test data point [True]
263 Text features [transforming] present in test data point [True]
328 Text features [3t3] present in test data point [True]
346 Text features [downstream] present in test data point [True]
351 Text features [requisite] present in test data point [True]
362 Text features [expressing] present in test data point [True]
376 Text features [activated] present in test data point [True]
384 Text features [272] present in test data point [True]
448 Text features [tk] present in test data point [True]
460 Text features [egfrs] present in test data point [True]
489 Text features [oncogene] present in test data point [True]
493 Text features [missense] present in test data point [True]
Out of the top 500 features 14 are present in query point

D) Linear Support Vector Machine

C-Support Vector Classification.

The implementation is based on libsvm. The fit time scales at least quadratically with the number of samples and may be impractical beyond tens of thousands of samples. For large datasets consider using LinearSVC or SGDClassifier instead, possibly after a Nystroem transformer.

The multiclass support is handled according to a one-vs-one scheme.

For details on the precise mathematical formulation of the provided kernel functions and how gamma, coef0 and degree affect each other, see the corresponding section in the narrative documentation: Kernel functions.

1. Hyper parameter tuning to find the best parameters for the Linear Support Vector Machine Model.

Parameter: alpha = [10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^{2}]
Result:
for alpha = 1e-05
Log Loss: 1.2573504108652491
for alpha = 0.0001
Log Loss: 1.1861474220425674
for alpha = 0.001
Log Loss: 1.1179326652189527
for alpha = 0.01
Log Loss: 1.126528401228273
for alpha = 0.1
Log Loss: 1.361832899463871
for alpha = 1
Log Loss: 1.6759587406023402
for alpha = 10
Log Loss: 1.6902801511939116
for alpha = 100
Log Loss: 1.6902801441869568

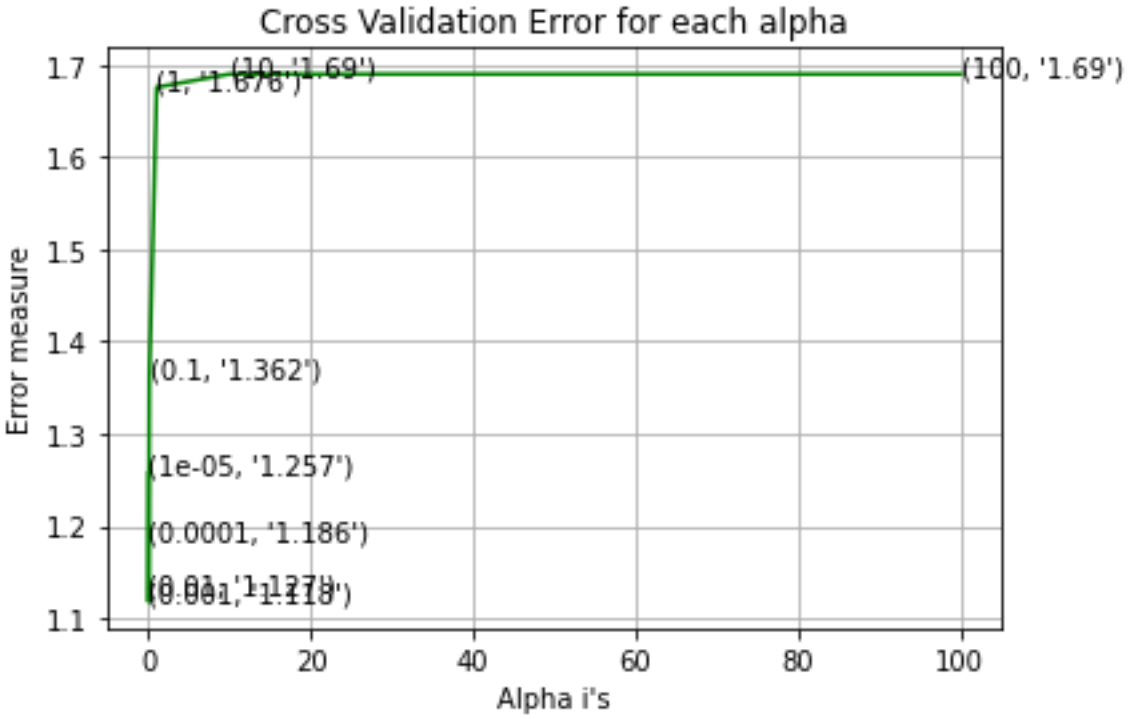


Fig. 17: Graph for finding best alpha Linear Support Vector Machine

We found out the best value of alpha is 0.001 with the minimum log loss of 1.1179326652189527.
We used alpha=0.1 on 3 different data set i.e., train, cross-validation and test.
The results are as follows:
For values of best alpha = 0.001 The train log loss is: 0.5571871527646932
For values of best alpha = 0.001 The cross-validation log loss is: 1.1179326652189527
For values of best alpha = 0.001 The test log loss is: 1.0908278397152973

2. Training:

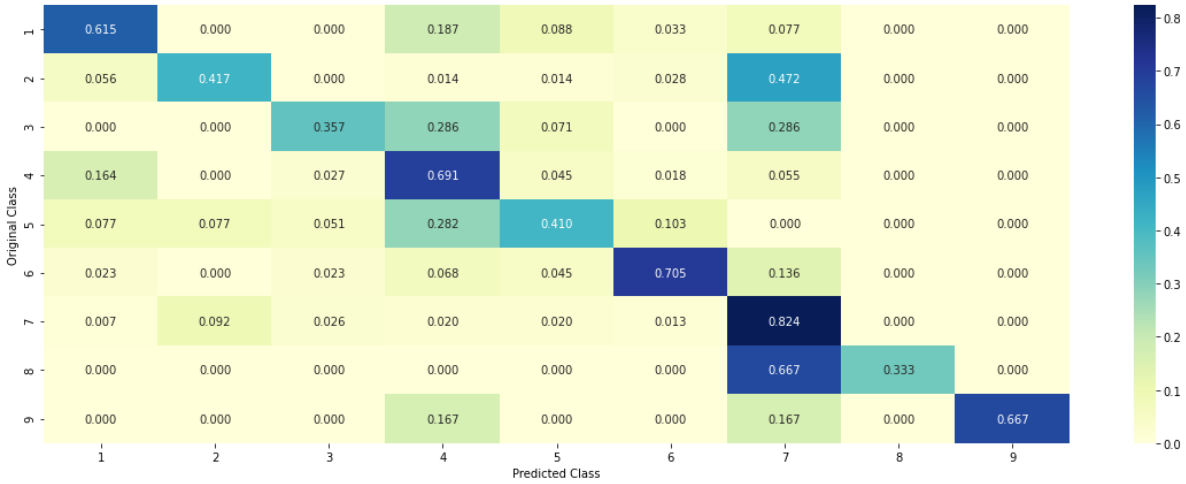
We trained the model on the best alpha parameters and here are the results:
Log loss : 1.1179326652189527

Number of mis-classified points: 0.35150375939849626

---------- Confusion matrix ----------

---------- Precision matrix (Column Sum=1) ----------

---------- Recall matrix (Row sum=1) ----------



3. Feature Importance, Correctly and incorrectly classified point

Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction

We tested the feature on test point index 1 and 100, for correctly and incorrectly classified points respectively:

The results for correctly classified points:

Predicted Class : 7

Predicted Class Probabilities: [[0.0465 0.0696 0.0076 0.1847 0.0275 0.0204 0.6338 0.004
0.0059]]

Actual Class: 7

---

451 Text features [35hosp] present in test data point [True]

Out of the top 500 features 1 are present in query point

The results for incorrectly classified points:

Predicted Class: 7

Predicted Class Probabilities: [[3.050e-02 3.298e-01 3.000e-04 8.000e-04 5.300e-03 9.000e-04 6.022e-01
2.740e-02 2.900e-03]]

Actual Class: 7

---

218 Text feature [hki] present in test data point [True]
263 Text feature [272] present in test data point [True]
292 Text feature [nonresponder] present in test data point [True]
447 Text feature [sirna1] present in test data point [True]
Out of the top 500 features 4 are present in query point

E) Random Forest Classifier

Probability calibration with isotonic regression or logistic regression.

This class uses cross-validation to both estimate the parameters of a classifier and subsequently calibrate a classifier. With default ensemble=True, for each cv split it fits a copy of the base estimator to the training subset, and calibrates it using the testing subset. For prediction, predicted probabilities are averaged across these individual calibrated classifiers. When ensemble=False, cross-validation is used to obtain unbiased predictions, via cross_val_predict, which are then used for calibration. For prediction, the base estimator, trained using all the data, is used. This is the method implemented when probabilities=True for sklearn.svm estimators.

Already fitted classifiers can be calibrated via the parameter cv="prefit". In this case, no cross-validation is used and all provided data is used for calibration. The user has to take care manually that data for model fitting and calibration are disjoint.

The calibration is based on the decision_function method of the base_estimator if it exists, else on predict_proba.

ONE-HOT ENCODING:

3. Hyper parameter tuning to find the best parameters for the Random Forest Model.

Parameter: alpha = [100,200,500,1000,2000]
max_depth = [5, 10]
Result:
for n_estimators = 100 and max depth = 5
Log Loss : 1.2117646915024245
for n_estimators = 100 and max depth = 10
Log Loss : 1.1463127691057031
for n_estimators = 200 and max depth = 5
Log Loss : 1.2023450816469876
for n_estimators = 200 and max depth = 10
Log Loss : 1.1319665547120128

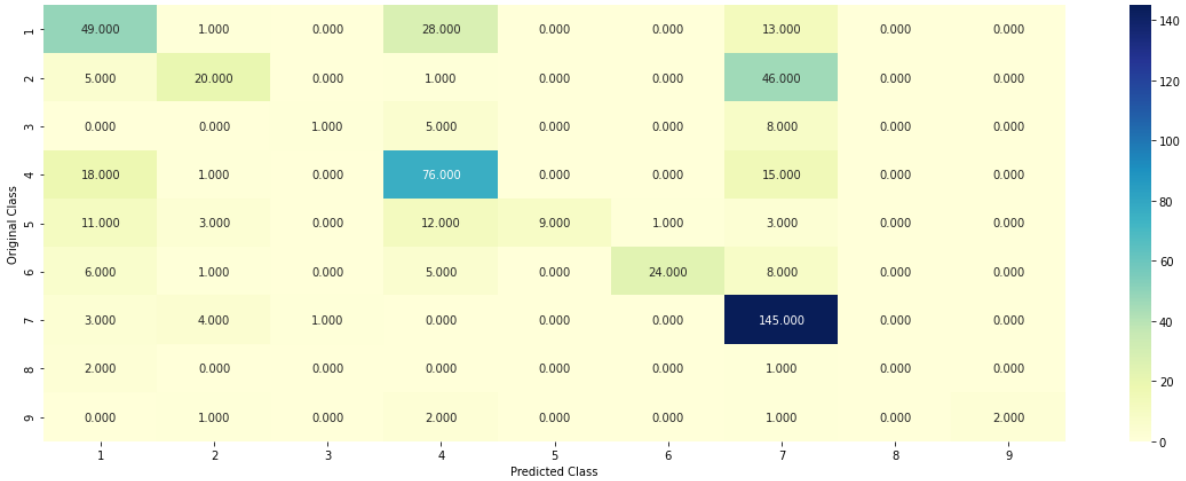
for n_estimators = 500 and max depth = 5
Log Loss : 1.1934757426410991
for n_estimators = 500 and max depth = 10
Log Loss : 1.1275368216726125
for n_estimators = 1000 and max depth = 5
Log Loss : 1.1931487964604819
for n_estimators = 1000 and max depth = 10
Log Loss : 1.1250140065886012
for n_estimators = 2000 and max depth = 5
Log Loss : 1.191842393327714
for n_estimators = 2000 and max depth = 10
Log Loss : 1.1247343186836187
We found out the best value of alpha is 2000 as the best estimator corresponding at the max depth of 5 with the minimum log loss of 1.1247343186836187.
We used estimator=2000 on 3 different data set i.e., train, cross-validation and test.
The results are as follows:
For values of best estimator = 2000 The train log loss is: 0.7020311125801458
For values of best estimator = 2000 The cross-validation log loss is: 1.124734318683619
For values of best estimator = 2000 The test log loss is: 1.1579329473110322

2. Training:

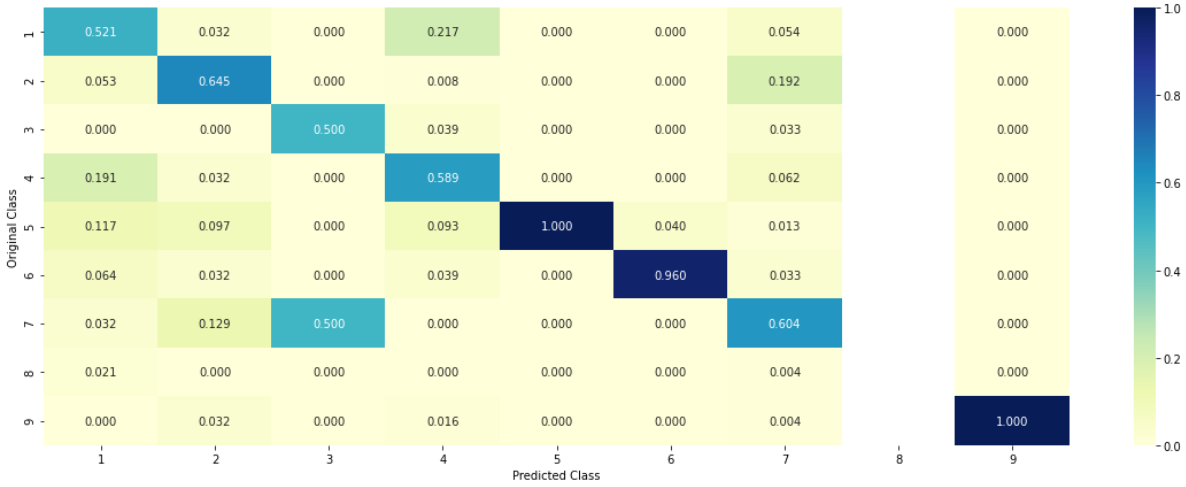
We trained the model on the best alpha parameters and here are the results:
Log loss: 1.124734318683619

Number of mis-classified points : 0.38721804511278196

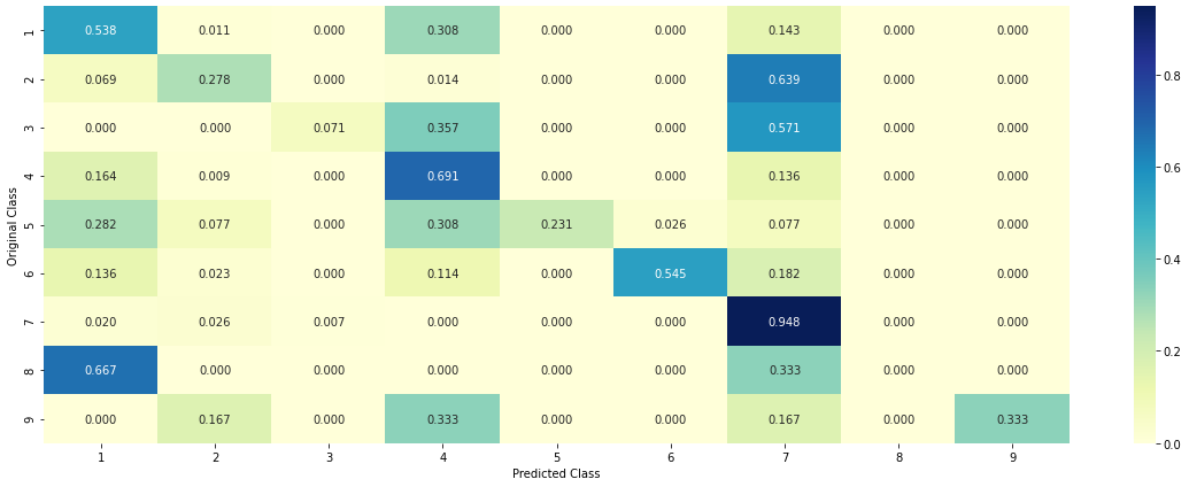
---------- Confusion matrix ----------



---------- Precision matrix (Columm Sum=1) ----------



---------- Recall matrix (Row sum=1) ----------



3. Feature Importance, Correctly and incorrectly classified point

Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction

We tested the feature on test point index 1 and 100, for correctly and incorrectly classified points respectively:

The results for correctly classified points:

Predicted Class : 7

Predicted Class Probabilities: [[0.0842 0.1113 0.018 0.0847 0.0506 0.0394 0.5964 0.0092 0.0063]]

Actual Class : 7

---

0 Text feature [kinase] present in test data point [True]

1 Text feature [activating] present in test data point [True]

2 Text feature [activation] present in test data point [True]

3 Text feature [tyrosine] present in test data point [True]

3 Text feature [inhibitors] present in test data point [True]

3 Text feature [activated] present in test data point [True]

3 Text feature [signaling] present in test data point [True]

7 Text feature [phosphorylation] present in test data point [True]

8 Text feature [oncogenic] present in test data point [True]

9 Text feature [treatment] present in test data point [True]

10 Text feature [missense] present in test data point [True]

13 Text feature [growth] present in test data point [True]

14 Text feature [inhibitor] present in test data point [True]

15 Text feature [erk] present in test data point [True]

16 Text feature [treated] present in test data point [True]

17 Text feature [cells] present in test data point [True]

18 Text feature [constitutively] present in test data point [True]

20 Text feature [function] present in test data point [True]

21 Text feature [yeast] present in test data point [True]

22 Text feature [trials] present in test data point [True]

24 Text feature [kinases] present in test data point [True]

26 Text feature [akt] present in test data point [True]

27 Text feature [receptor] present in test data point [True]

28 Text feature [functional] present in test data point [True]

30 Text feature [patients] present in test data point [True]

31 Text feature [downstream] present in test data point [True]

32 Text feature [transforming] present in test data point [True]

33 Text feature [loss] present in test data point [True]

37 Text feature [inhibition] present in test data point [True]

38 Text feature [activate] present in test data point [True]

40 Text feature [protein] present in test data point [True]

42 Text feature [3t3] present in test data point [True]

45 Text feature [advanced] present in test data point [True]

47 Text feature [variants] present in test data point [True]

48 Text feature [inhibited] present in test data point [True]

49 Text feature [resistance] present in test data point [True]

51 Text feature [therapeutic] present in test data point [True]

52 Text feature [mitogen] present in test data point [True]

55 Text feature [transform] present in test data point [True]

58 Text feature [drug] present in test data point [True]

59 Text feature [response] present in test data point [True]

63 Text feature [clinical] present in test data point [True]

65 Text feature [38hosphor] present in test data point [True]

67 Text feature [defective] present in test data point [True]

68 Text feature [mek] present in test data point [True]

69 Text feature [cell] present in test data point [True]

70 Text feature [expressing] present in test data point [True]

72 Text feature [stimulation] present in test data point [True]

74 Text feature [expression] present in test data point [True]

79 Text feature [serum] present in test data point [True]

80 Text feature [type] present in test data point [True]

81 Text feature [proteins] present in test data point [True]

82 Text feature [amplification] present in test data point [True]

83 Text feature [mapk] present in test data point [True]

86 Text feature [nuclear] present in test data point [True]

87 Text feature [sensitivity] present in test data point [True]

88 Text feature [ras] present in test data point [True]

90 Text feature [harboring] present in test data point [True]

99 Text feature [pathway] present in test data point [True]

Out of the top 100 features 59 are present in query point

The results for incorrectly classified points:

Predicted Class : 7

Predicted Class Probabilities: [[0.0216 0.134 0.011 0.0167 0.0326 0.0245 0.7515 0.0041 0.004]]

Actuall Class : 7

---

0 Text feature [kinase] present in test data point [True]
1 Text feature [activating] present in test data point [True]
2 Text feature [activation] present in test data point [True]
3 Text feature [tyrosine] present in test data point [True]

3 Text feature [inhibitors] present in test data point [True]

3 Text feature [activated] present in test data point [True]

3 Text feature [signaling] present in test data point [True]
7 Text feature [phosphorylation] present in test data point [True]
8 Text feature [oncogenic] present in test data point [True]
9 Text feature [treatment] present in test data point [True]
10 Text feature [missense] present in test data point [True]
13 Text feature [growth] present in test data point [True]
14 Text feature [inhibitor] present in test data point [True]
15 Text feature [erk] present in test data point [True]
16 Text feature [treated] present in test data point [True]
17 Text feature [cells] present in test data point [True]
18 Text feature [constitutively] present in test data point [True]
19 Text feature [suppressor] present in test data point [True]
20 Text feature [function] present in test data point [True]
22 Text feature [trials] present in test data point [True]

24 Text feature [kinases] present in test data point [True]
25 Text feature [therapy] present in test data point [True]
26 Text feature [akt] present in test data point [True]
27 Text feature [receptor] present in test data point [True]
28 Text feature [functional] present in test data point [True]
29 Text feature [stability] present in test data point [True]
30 Text feature [patients] present in test data point [True]
31 Text feature [downstream] present in test data point [True]
32 Text feature [transforming] present in test data point [True]
33 Text feature [loss] present in test data point [True]
35 Text feature [ba] present in test data point [True]
36 Text feature [months] present in test data point [True]
37 Text feature [inhibition] present in test data point [True]
38 Text feature [activate] present in test data point [True]
40 Text feature [protein] present in test data point [True]
42 Text feature [3t3] present in test data point [True]
44 Text feature [efficacy] present in test data point [True]
45 Text feature [advanced] present in test data point [True]
47 Text feature [variants] present in test data point [True]
48 Text feature [inhibited] present in test data point [True]
49 Text feature [resistance] present in test data point [True]
50 Text feature [extracellular] present in test data point [True]
51 Text feature [therapeutic] present in test data point [True]
52 Text feature [mitogen] present in test data point [True]
54 Text feature [f3] present in test data point [True]
55 Text feature [transform] present in test data point [True]
58 Text feature [drug] present in test data point [True]
59 Text feature [response] present in test data point [True]
62 Text feature [egfr] present in test data point [True]
63 Text feature [clinical] present in test data point [True]
68 Text feature [mek] present in test data point [True]
69 Text feature [cell] present in test data point [True]
70 Text feature [expressing] present in test data point [True]
73 Text feature [ic50] present in test data point [True]
74 Text feature [expression] present in test data point [True]
75 Text feature [tki] present in test data point [True]
79 Text feature [serum] present in test data point [True]
80 Text feature [type] present in test data point [True]
81 Text feature [proteins] present in test data point [True]
82 Text feature [amplification] present in test data point [True]
83 Text feature [mapk] present in test data point [True]
84 Text feature [likelihood] present in test data point [True]
87 Text feature [sensitivity] present in test data point [True]
89 Text feature [unclassified] present in test data point [True]
90 Text feature [harboring] present in test data point [True]
91 Text feature [phosphorylated] present in test data point [True]
95 Text feature [tkis] present in test data point [True]
96 Text feature [nsclc] present in test data point [True]
97 Text feature [dose] present in test data point [True]
98 Text feature [daily] present in test data point [True]
99 Text feature [pathway] present in test data point [True]
Out of the top 100 features 71 are present in query point

RESPONSE-CODING:

3. Hyper parameter tuning to find the best parameters for the Random Forest Model.
Parameter: alpha = [10,50,100,200,500,1000]
max_depth: = [2,3,5,10]
Result:
for n_estimators = 10 and max depth = 2
Log Loss : 1.9269743051199821
for n_estimators = 10 and max depth = 3
Log Loss : 1.5339420997947961
for n_estimators = 10 and max depth = 5
Log Loss : 1.4166356996191598
for n_estimators = 10 and max depth = 10
Log Loss : 1.8399036924005878
for n_estimators = 50 and max depth = 2
Log Loss : 1.5353929494228244
for n_estimators = 50 and max depth = 3
Log Loss : 1.294592653986634
for n_estimators = 50 and max depth = 5
Log Loss : 1.2828086404303864
for n_estimators = 50 and max depth = 10
Log Loss : 1.7078945157963192
for n_estimators = 100 and max depth = 2
Log Loss : 1.4611012841917075
for n_estimators = 100 and max depth = 3
Log Loss : 1.3560072036850284
for n_estimators = 100 and max depth = 5
Log Loss : 1.2759343444449007
for n_estimators = 100 and max depth = 10
Log Loss : 1.673812731602605
for n_estimators = 200 and max depth = 2
Log Loss : 1.4255234400766772
for n_estimators = 200 and max depth = 3
Log Loss : 1.3531749662824415
for n_estimators = 200 and max depth = 5
Log Loss : 1.3208612543663225
for n_estimators = 200 and max depth = 10
Log Loss : 1.668639515175503
for n_estimators = 500 and max depth = 2
Log Loss : 1.467846440522522
for n_estimators = 500 and max depth = 3
Log Loss : 1.3974278470068704
for n_estimators = 500 and max depth = 5
Log Loss : 1.2999230966925774
for n_estimators = 500 and max depth = 10
Log Loss : 1.7026710535615714
for n_estimators = 1000 and max depth = 2
Log Loss : 1.4575058978835775
for n_estimators = 1000 and max depth = 3
Log Loss : 1.413262080969932
for n_estimators = 1000 and max depth = 5
Log Loss : 1.2824347817603192
for n_estimators = 1000 and max depth = 10
Log Loss : 1.7008542563825004
We found out the best value of alpha is 100 as the best estimator corresponding at the max depth of 5 with the minimum log loss of Log Loss : 1.2759343444449007.
We used best alpha=100 on 3 different data set i.e., train, cross-validation and test.

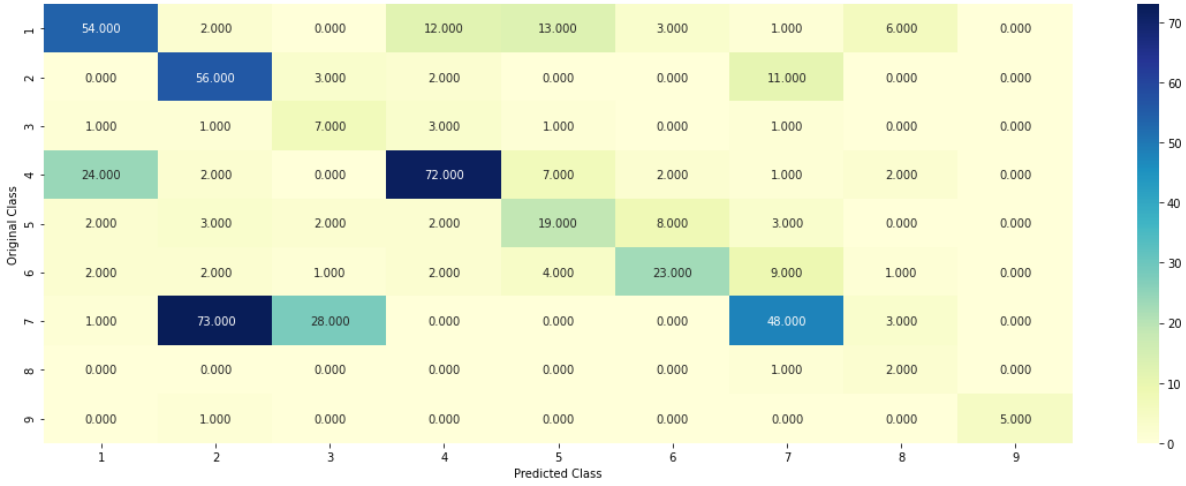
The results are as follows:
For values of best alpha = 100 The train log loss is: 0.06728724540519664
For values of best alpha = 100 The cross validation log loss is: 1.2759343444449007
For values of best alpha = 100 The test log loss is: 1.3237138197628435

2. Training:

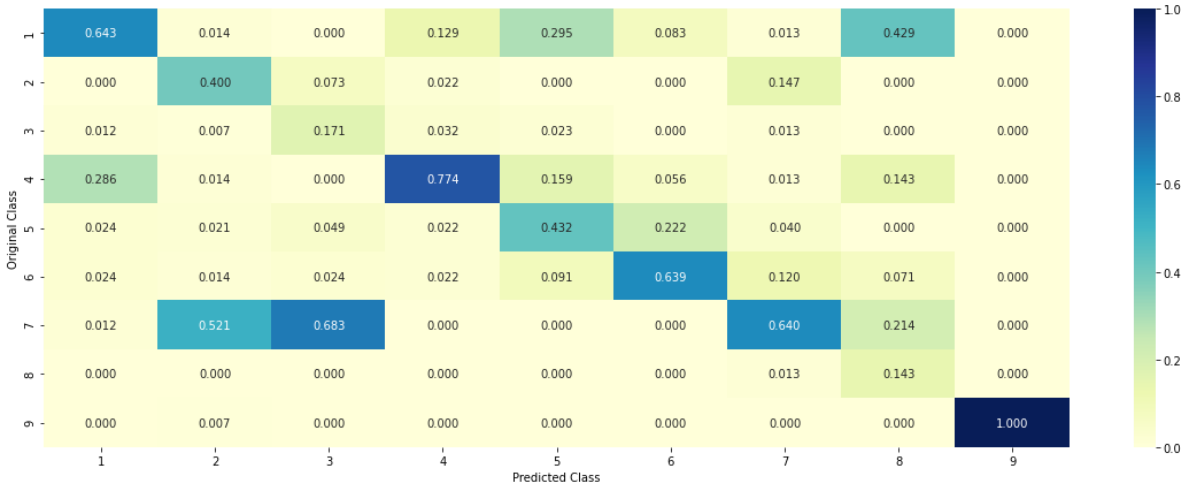
We trained the model on the best alpha parameters and here are the results:
Log loss: 1.2759343444449007

Number of mis-classified points: 0.462406015037594

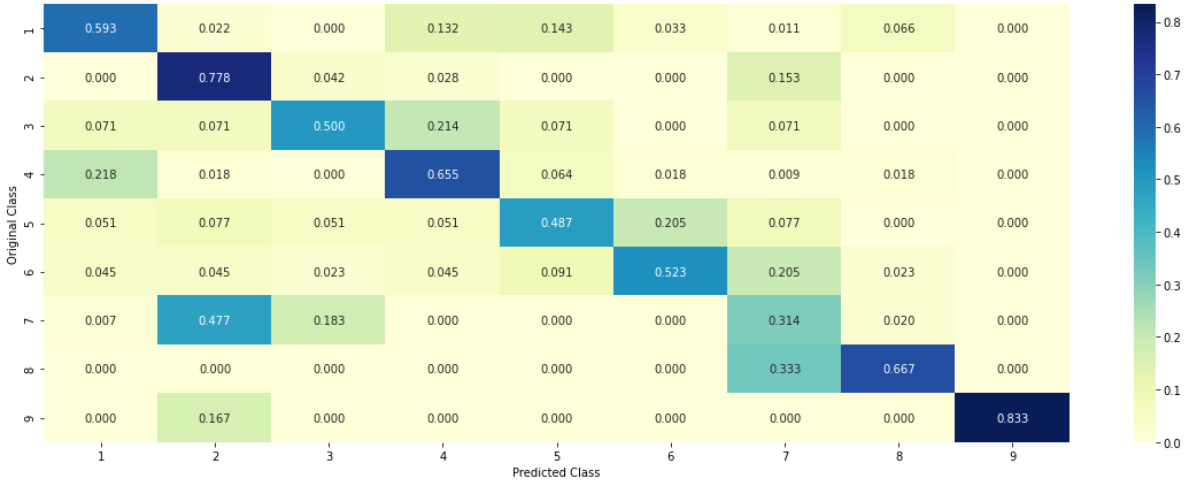
---------- Confusion matrix ----------



---------- Precision matrix (Columm Sum=1) ----------



---------- Recall matrix (Row sum=1) ----------



3. Feature Importance, Correctly and incorrectly classified point

Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction

We tested the feature on test point index 1 and 100, for correctly and incorrectly classified points respectively:

The results for correctly classified points:

Predicted Class : 7

Predicted Class Probabilities: [[0.0209 0.2515 0.1929 0.035 0.0437 0.0552 0.3285 0.0494 0.023]]

Actual Class : 7

Variation is important feature

Variation is important feature

Variation is important feature

Variation is important feature

Gene is important feature

Variation is important feature

Variation is important feature

Text is important feature

Text is important feature

Gene is important feature

Text is important feature

Text is important feature

Text is important feature

Gene is important feature

Variation is important feature

Gene is important feature

Text is important feature

Gene is important feature

Gene is important feature

Variation is important feature

Text is important feature

Variation is important feature

Text is important feature

Gene is important feature

Text is important feature

Gene is important feature

Gene is important feature

The results for incorrectly classified points:
Predicted Class: 2
Predicted Class Probabilities: [[0.01 0.674 0.0754 0.0145 0.02 0.03 0.1362 0.0289 0.011]]
Actual Class: 7

Variation is important feature
Variation is important feature
Variation is important feature
Variation is important feature
Gene is important feature
Variation is important feature
Variation is important feature
Text is important feature
Text is important feature
Gene is important feature
Text is important feature
Text is important feature
Text is important feature
Gene is important feature
Variation is important feature
Gene is important feature
Text is important feature
Gene is important feature
Gene is important feature
Variation is important feature
Text is important feature
Variation is important feature
Text is important feature
Gene is important feature
Text is important feature
Gene is important feature
Gene is important feature

3.2 Ensemble Method Learning

A. Stacking Classifier

To find the best prediction results from all the algorithms we implemented. We stacked the models.
Stacking is an *ensemble machine learning* algorithm. Stacking is an ensemble machine learning algorithm that learns how to best combine the predictions from multiple well-performing machine learning models.

Models stacked:

1. SGDClassifier
   Hyperparameters:
   a. alpha=0.001
   b. penalty='l2'
   c. loss='log'
   d. class_weight='balanced'
   e. random_state=0
2. SGDClassifier
   Hyperparameters:
   a. alpha=1
   b. penalty='l2'
   c. loss='hinge'
   d. class_weight='balanced'
   e. random_state=0
3. MultinomialNB
   Hyperparameter:
   a. alpha=0.001

We found logloss on individual models initially. The results for the same are as follows:

```
Logistic Regression :  Log Loss: 1.06
Support vector machines : Log Loss: 1.68
Naive Bayes : Log Loss: 1.21
```

Later we performed hyperparameter tuning on the stacked classifiers. Where alpha was taken as [0.0001,0.001,0.01,0.1,1,10]

The results for the same is as follows:

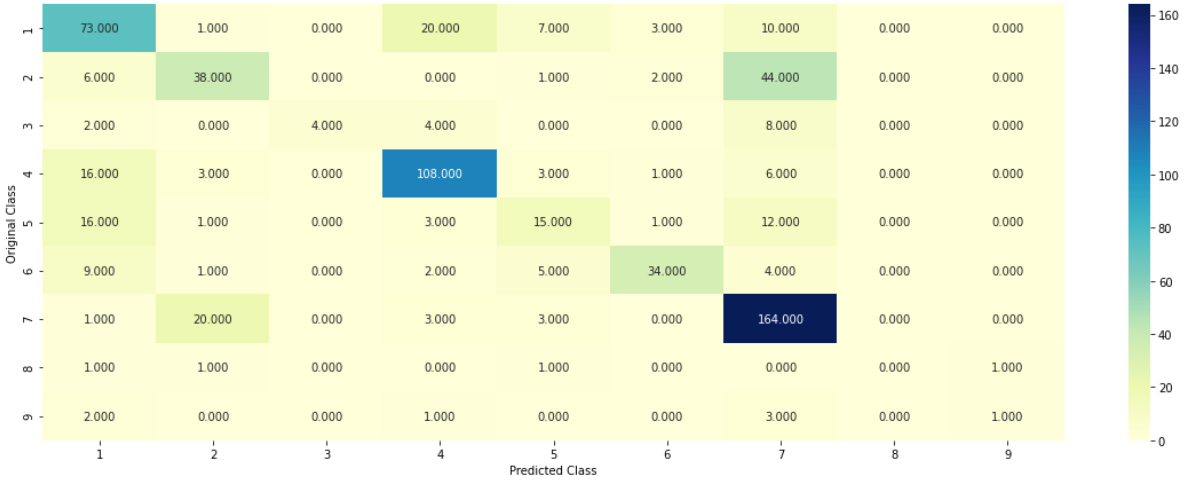
```
Stacking Classifer : for the value of alpha: 0.000100 Log Loss: 1.818
Stacking Classifer : for the value of alpha: 0.001000 Log Loss: 1.723
Stacking Classifer : for the value of alpha: 0.010000 Log Loss: 1.314
Stacking Classifer : for the value of alpha: 0.100000 Log Loss: 1.124
Stacking Classifer : for the value of alpha: 1.000000 Log Loss: 1.382
Stacking Classifer : for the value of alpha: 10.000000 Log Loss: 1.723
```

Testing stacked models with best hyperparameters:

```
Log loss (train) on the stacking classifier : 0.5117441250340428
```

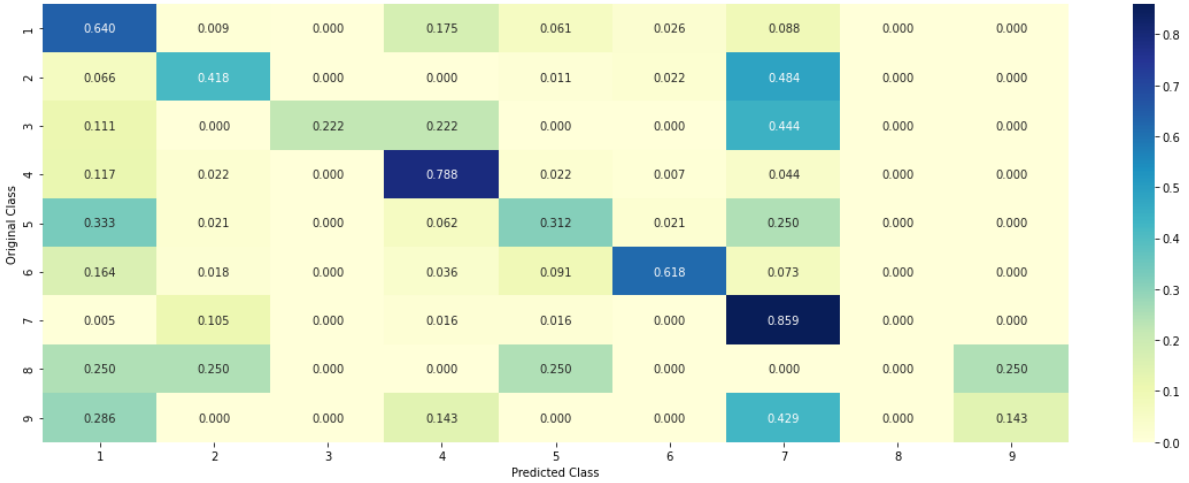
```
Log loss (CV) on the stacking classifier : 1.1244555008840966
Log loss (test) on the stacking classifier : 1.1357753343684847
Number of missclassified point : 0.34285714285714286
```

---------- Confusion matrix ----------



---------- Precision matrix (Columm Sum=1) ----------

---------- Recall matrix (Row sum=1) ----------



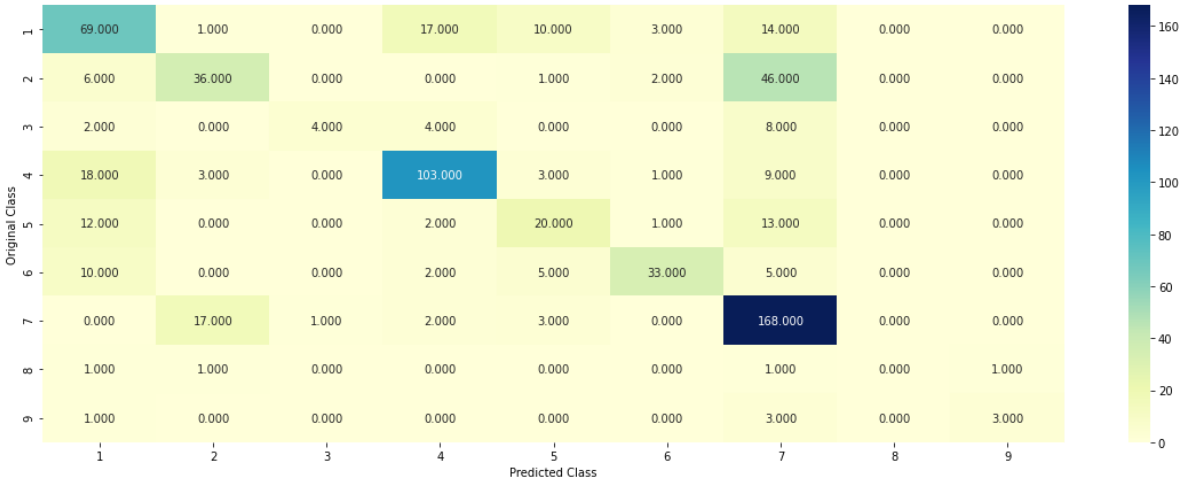
B. Maximum Voting Classifier

Voting Classifier supports two types of votings. Hard Voting: In hard voting, the predicted output class is a class with the highest majority of votes i.e the class which had the highest probability of being predicted by each of the classifiers**.**

We found the log loss for Training, CV and Testing respectively.:

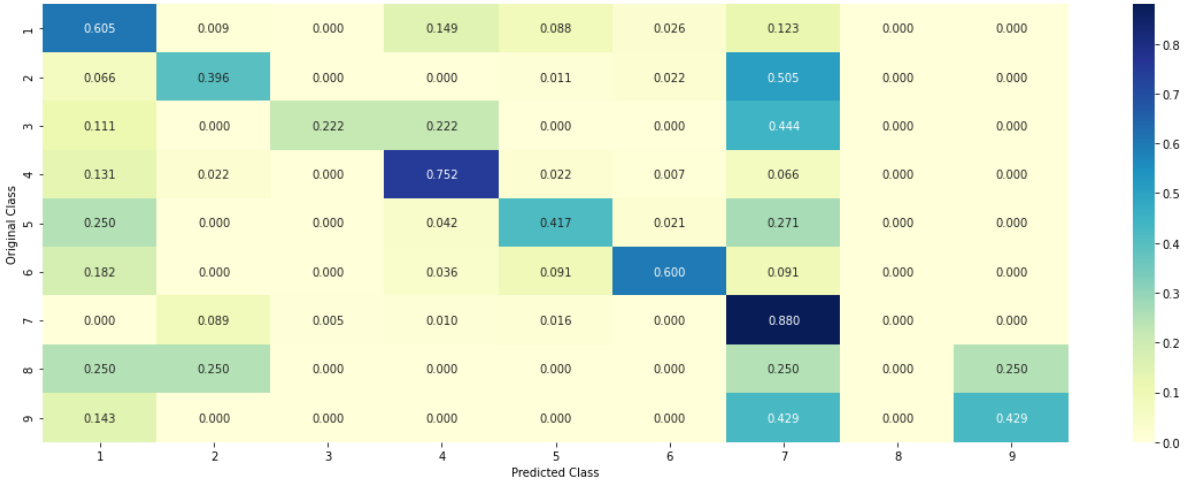
```
Log loss (train) on the VotingClassifier : 0.8793295217384377
Log loss (CV) on the VotingClassifier : 1.151274911188027
Log loss (test) on the VotingClassifier : 1.1812610571262407
Number of missclassified point : 0.3443609022556391
```

---------- Confusion matrix ----------



---------- Precision matrix (Column Sum=1) ----------

---------- Recall matrix (Row sum=1) ----------



3.3 Result

The Log Loss and Accuracy of all the models have been tabulated below.

Models	**Log loss**		
	Train data (64%)	**Cross validation data (16%)**	**Test data (20%)**
Naïve Bayes	0.855936	1.247139	1.255047
K Nearest Neighbor	0.454222	1.025018	1.057909
Logistic Regression (*Class Balancing*)	0.538460	1.035269	1.020829
Logistic Regression (*Without Class Balancing*)	0.533721	1.050540	1.027963
Linear Support Vector Machines	0.546302	1.095540	1.068224
Random Forest	0.701327	1.139863	1.141111

Classifier (One Hot Encoding)			
Random Forest Classifier (Response coding)	0.080826	1.306169	1.389727
Ensemble learning Method			
Stacking Method	0.522348	1.138801	1.118995
Maximum Voting Classifier	0.881708	1.162095	1.173151

Models	**Accuracy**		
	Train data (64%)	**Cross validation data (16%)**	**Test data (20%)**
Naïve Bayes	83.3804	62.2180	62.4060
K Nearest Neighbor	86.1581	65.03759	65.8646
Logistic Regression (*Class Balancing*)	89.4067	65.2255	67.6691
Logistic Regression (*Without Class Balancing*)	89.9717	64.8496	69.0225
Linear Support Vector Machines	96.0922	65.4135	68.2706
Random Forest Classifier (One Hot Encoding)	83.5216	62.5939	62.5563
Random Forest Classifier (Response coding)	98.6817	49.4360	48.8721
Ensemble learning Method			
Stacking Method	88.0885	63.9097	66.3157
Maximum Voting Classifier	88.6534	63.1578	66.0150

4. CONCLUSION AND FUTURE SCOPE

A. CONCLUSION

Understanding the genetic mutations that matter's in the evolution of cancer would be a tumour, which is a challenging task with a potentially huge impact on millions of lives. In this project, we have tried to fulfil our goal of automatically classifying the genetic variations and mutations by leveraging the knowledge of Machine Learning and using its algorithms to solve this problem. This model would be helpful in the field of Medical research as it will save the Molecular Pathologist's time. Now the molecular pathologist won't have to spend a huge amount of time analyzing the evidence related to each of the variations to classify them, it would be completely automated using our model.

For this model, we have taken the nine different classes a genetic mutation can be classified on. The training and test, data sets are provided via two different files. One (training/test_variants) provides the information about the genetic mutations, whereas the other (training/test_text) provides the clinical evidence (text) that our human experts used to classify the genetic mutations. Both are linked via the ID field. We have used five different algorithms for classification: Naive Bayes Classifier, K Nearest Neighbor Classification, Logistic Regression, Linear Support Vector Machine and Random Forest Classifier. Depending on the precision value of the confusion matrix of each of the five algorithms, the top three algorithms with the best performance are used for further operations and stacking of data.

After, training, Cross-Validating and Testing our models on various algorithms based models. We shifted our learning Model to Ensemble-based learning, two models were used Stacking Method and Maximum Voting Classifier.
The Accuracy for Training, CV and testing showed by both the individual model's were nearly equal to the average accuracy of all Algorithm based models.
This shows that the Ensemble-based algorithm works better and consistently for the data-set giving better and optimized result with fewer misclassified points.

Based on the analysis of their results, it is evident that the integration of multidimensional heterogeneous data, combined with the application of different techniques for feature selection and classification can provide promising tools for inference in the cancer domain

B. FUTURE SCOPE

For future works in this field, there are several components that we believe other researchers should adopt to their frameworks of scientific work. For this project, we have taken 3321 data points that contain various genes and it's variations. Ideally, for detecting cancer, a molecular pathologist works in this manner:-

STEP 1: A molecular pathologist selects a list of genetic variations of interest that he/she wants to analyze

STEP 2: The molecular pathologist searches for evidence in the medical literature that somehow is relevant to the genetic variations of interest

STEP 3: Finally this molecular pathologist spends a huge amount of time analyzing the evidence related to each of the variations to classify them

Analyzing the evidence related to each of the variations is very time consuming as here, we have just taken a small set of data points, but in real-world, there may be more than 105 potential data points available and analyzing all of them would be very time consuming for researchers, pathologists and hospitals. Therefore, we have tried to replace step 3 with our machine learning model. The molecular pathologist will still have to decide which variations

are of interest, and also collect the relevant evidence for them. But the last step, which is also the most time consuming, will be fully automated.

This would save the pathologists and researchers a huge amount of time and will also help the hospitals to detect and treat the patient at an early stage of cancer. It would prove to be cost-effective to both, the patient as well as the doctors. Although the accuracy of our model is __%, we still believe there is scope for improvement. Researchers can improve this model by including more data and performing more experiments. A better statistical analysis of the heterogeneous datasets used would provide more accurate results and would give reasoning to disease outcomes. Further research is required based on the construction of more public databases that would collect valid cancer dataset of all patients that have been diagnosed with the disease.

5. REFRENCES:

[1] Anoy Chowdhury, “Breast Cancer Detection and Prediction using Machine Learning”, Research Proposal - June 2020.

[2] Majid Murtaza Noor & Vinay Narwal, “Machine Learning Approaches in Cancer Detection and Diagnosis: Mini Review”, Research – November 2017.

[3] Maalel, A. & Hattab, M. (2019) "Literature review: Overview of Cancer Treatment and Prediction Approaches based on Machine Learning" Smart Systems for E-Health, Advanced Information and Knowledge Processing, Springer, ISBN : 978-3-030-14938-3, pp. 324.

[4] Joseph A. Cruz, David S. Wishart, “Applications of Machine Learning in Cancer Prediction and Prognosis”, Departments of Biological Science and Computing Science, University of Alberta Edmonton, AB, Canada T6G 2E8.

[5] Konstantina Kourou, Themis P. Exarchos, Konstantinos P. Exarchos, Michalis V.Karamouzis, Dimitrios I. Fotiadis, “Machine learning applications in cancer prognosis and prediction”, Computational and Structural Biotechnology Journal, Vol. 13, 2015, Pages 8-17.

[6] D. R. Green. (2017). Cancer and Apoptosis: Who Is Built to Last? .Cancer Cell, 31(1), 2-4.

[7] H. E. Bock & U. Dold. (1967). Early Diagnosis and its Importance for Prognosis and Therapy. New Trends in the Treatment of Cancer, 1-14.

[8] F. Toscani. (1996). Classification and staging of terminal cancer patients: Rationale and objectives of a multicentre cohort prospective study and methods used. Supportive Care in Cancer, 4(1), 56-60.

[9] A. Bhola, & A. K. Tiwari. (2015). Machine Learning Based Approaches for Cancer Classification Using Gene Expression Data. Machine Learning and Applications: An International Journal, 2(3/4), 01-12.

[10] A. H. Fielding. (1999). An introduction to machine learning methods. Machine Learning Methods for Ecological Applications, 1-35.

Acknowledgement

We would like to express our gratitude and thanks to **Dr. Tanuja Sarode** for her valuable guidance and help. We are indebted for her guidance and constant supervision as well as for providing necessary information regarding the project. We would like to express our greatest appreciation to our principal **Dr. G.T. Thampi** and head of the department **Dr. Tanuja Sarode** for their encouragement and tremendous support. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of the project.

Musharraf Alam

Ishan Agarwal

Asawa Aryan

Yash Balchandani