**I) Summary:**

**1. Problem Definition:**

A lot has been said during the past several years about how precision medicine and, more concretely, how genetic testing is going to disrupt the way diseases like cancer are treated.

But this is only partially happening due to the huge amount of manual work still required.

Memorial Sloan Kettering Cancer Center (MSKCC) launched this competition to take personalized medicine to its full potential.

A cancer tumor can have thousands of genetic mutations. And distinguishing the mutations that contribute to tumor growth is challenging.

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.

**Workflow is as follows:**

i. A molecular pathologist selects a list of genetic variations of interest that he/she want to analyze

ii. The molecular pathologist searches for evidence in the medical literature that somehow are relevant to the genetic variations of interest.

iii. Finally this molecular pathologist spends a huge amount of time analyzing the evidence related to each of the variations to classify them.

Our goal here is to replace step 3 by a 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.

**2. Objective:**

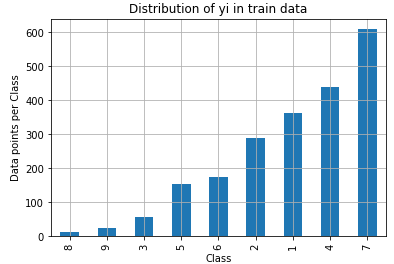
Predict the probability of each data-point belonging to each of the nine classes.

**3.Constraints:**

\* Interpretability, \* Class probabilities are needed, \* Penalize the errors in class probabilites => Metric is Log-loss. \* No Latency constraints.

**4. EDA:**

i. More number of data points belong to class 7, 4, 1 and 2 .



ii. Prediction of class using Random model and plot confusion matrix.

Log loss on CV data using Random model = 2.459

Log loss on Test Data using Random model = 2.498

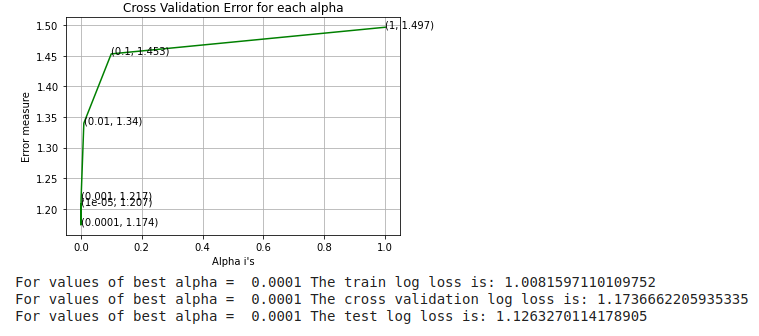
**5. Univariate Analysis of Gene Feature:**

**Observations:**

i. Top 50 genes contribute to 70% of data and remaining genes contribute 30% of data.

ii. Lots of genes occur very few times and very few genes occur lot of times.

iii. How good Gene feature in predicting class label? : Train Logistic Regression model using only the “Gene Feature” .



**iv. Stability of Gene feature :**

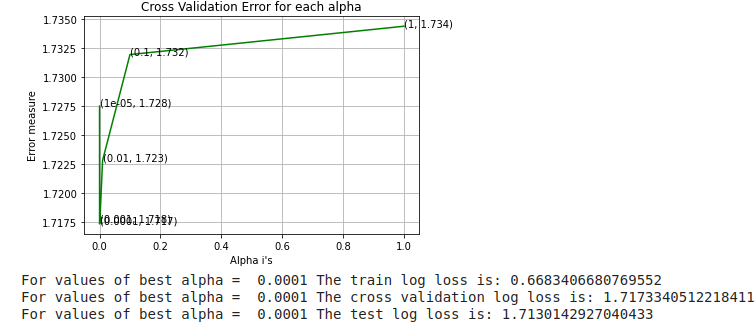
In test data out of 665 data points , 664 are present in the train dataset also.

In CV data out of 532 data points, 519 are present in the train dataset also.

**6. Univariate Analysis of Variation Feature:**

i. How good Gene feature in predicting class label? : Train Logistic Regression model using only the “Veriation Feature” .

Train Logistic Regression model with “Variation Feature” alone.



**ii. Stability of Variation feature :**

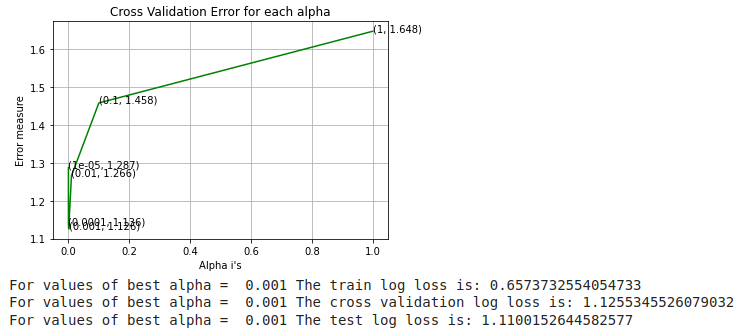
In test data 66 out of 665 data points are present in train data.

In CV data 51 out of 532 data points are present in Train data.

**7. Univariate Analysis of Gene Feature:**

i. Total number of unique words in train data : 53491

ii. Train Logistic Regression model with “Text Feature” alone.



i**ii. Stability of Text feature :**

97.148 % of word of test data appeared in train data

97.602 % of word of Cross Validation appeared in train data

**8. Combining all the features using hstack and Training the different models with data.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Hyperparameter (alpha)** | **Logloss (Train data)** | **Logloss (CV data)** | **Logloss (Test Data)** | **% Mis-classified points** |
| Random Model | \_\_ | \_\_ | 2.561 | 2.435 |  |
| Logistic Regression(with Balancing) | 0.001 | 0.523 | 1.086 | 1.034 |  |
| Logistic Regression(without Balancing) | 0.001 | 0.521 | 1.088 | 1.039 |  |
| Naive Bayes | 1e-5 | 0.798 | 1.279 | 1.244 |  |
| SVM | 0.001 | 0.530 | 1.128 | 1.073 |  |
| Random Forest | 2000 (no of trees) | 0.689 | 1.158 | 1.104 |  |
| KNN | 15 (nearest neighbors) | 0.667 | 1.058 | 0.948 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Hyperparameter (alpha)** | **Logloss (Train data)** | **Logloss (CV data)** | **Logloss (Test Data)** | **% Mis-classified points** |
| Naive Bayes (One hot encoding) | 1e-5 | 0.798 | 1.279 | 1.244 |  |
| K-NN (Response Coding) | 15 (nearest neighbors) | 0.667 | 1.058 | 0.948 |  |
| Logistic Regression (Balancing + One hot encoding) | 0.001 | 0.521 | 1.088 | 1.039 |  |
| Logistic Regression (without balancing + One hot encoding) |  |  |  |  |  |
| Lineara SVM (One hot encoding + Balancing) | 0.001 | 0.530 | 1.128 | 1.073 |  |
| Random Forest (One hot encoding) | 2000 (no of trees) | 0.689 | 1.158 | 1.104 |  |
| Random Forest (Response encoding) |  |  |  |  |  |

**Model Interpretation:**

**1.** **Naive Bayes :**

i.There is some gap between Train and CV log loss, this might be because naive bayes is simpler model.

ii. We can get Feaure imortance thus satisfies interpretability of the model. Ie, gives the reason for predicting the query point belonging to a particular class.

**2. K-NN (Response Coding):**

i. K-NN can’t work with high dimensional data, hence use response coding for the features.

ii. There is some gap in train log loss and cv log loss.

Iii. More easy to make mistakes in K-NN than Naive Bayes, because K-NN is not interpretable.

**3. Logistic Regression (Balancing + One hot encoding) :**

i. Logistic regression easily takes high dimensional data, hence use onehot encoding of the features.

ii. Is very interpretable and we can get feature importance with absolute values of Weights.

Iii. Class balancing helps to improve the prediction even for minority class labeled data points.

**4. Logistic Regression (without balancing + One hot encoding) :**

i. without class balancing precision and recall values will be nearly to zero.

ii. percentage of misclassified points increases witout class balancing.

**5. Lineara SVM (One hot encoding + Balancing) :**

i. When data is high dimensional, then Linear SVM works very well.

ii. Interpretability of the model is good (very simillar to logistic Regression).

Iii. We are not using RBF SVM because it is not easily interpretable and also we don’t know which kernel works well here.

**6. Random Forest (One hot encoding) :**

i. Random Forest works well when the dimensionality is small.

ii. There are 2 hyper parameters,

a. Number of tress / number of base estimators

b. Maximum depth of the tree.

Iii. As number of trees increase, model will generalize better.

iv. In precision matrix we are getting all diagonal elements =1 and also for all minor classes prediction has improved.

**7. Random Forest (Response encoding) :**

i. The difference between train log loss and cv log loss is very high and this means that model has overfit.

ii. And also the percentage of misclassified points is very high.

**Conclusion:**

By comparing all the model log loss, percentage of misclassified points and intepretability of the model, we conclude that **Logistic Regression (One hot encoded features) and with class balancing** gives the best results.

**Business Impact:**

1. As the Doctor receives a lot of samples to test, in that some of them model predicts (with very high probability of belonging to a particular class), then it would reduce the time taken by the doctor.

2. Doctor can concentrate only on the samples which the model can’t predict accurately, so that chance of making error by doctor reduces drastically.

3. We can make contribution to the humanity by saving a life