**Redefining the Cancer Treatment**

**Business Problem:**

The blood samples taken from Cancer suspect patient is analysed for genetic mutations. Some of the genetic mutation are cancerous and some are not. To exactly diagnose the cancer, the pathologist has to go through manually all research literature publish on this particular genetic mutation. It takes almost a day for the pathologist/domain expert to analyse the research text and classify the given gene and variation into one of the nine categories out of which some are cancerous and some are not.

Can AI help them to predict the given Gene, variation and Text research into one of the nine categories.

**Business Outcome:**

1. Saves the time spend by the pathologist significantly from almost a day to 3-4 seconds.
2. When there too many cancer suspect patients coming to the hospital, scarcity of the pathologist for analysing genetic mutation can be overcome.

**Problem statement:**

Classify the given Gene mutation, based on the evidence from text-based research literature, into one of the nine categories out of which some are Cancerous.

**Business constraints:**

1. Along with predicting the class, the probabilities of the classes are required.
2. Interpretability of the model is very important.
3. Reasons for predicting a particular class is required by the Doctor to be double sure about the predictions.

**Machine Learning Problem statement:**

Give a gene, variation and the research text on this given mutation, Build a model to classify it into one of the nine categories out of which some are Cancerous. Multi Class classification problem.

**Performance metric to be used:**

1. Multiclass log loss.

2. Precision Matrix.

3. Recall Metrix.

**Architecture of the Model building.**

Read the data in the Pandas DataFrame

Unzip the files and store it in the database

Read Data from the Data source

Data Source

Univariate analysis of Each features

Clean the Data

Perform Graphical Analysis

Perform Text Preprocessing

Model Training and Tunning

Building the Baseline Model

Feature Selection

Feature Encoding

Push to the Cloud

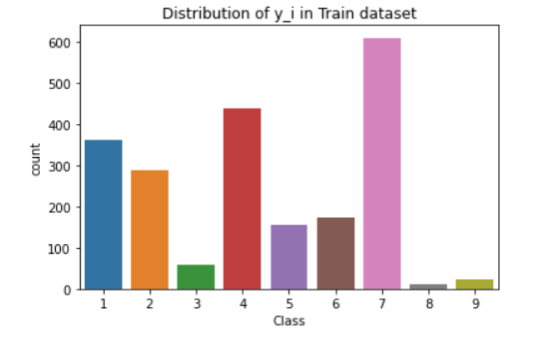
Deployment Readiness

Validation and Prediction

**Techniques used in Text Preprocessing.**

1. Removal of stop words.
2. In the text document remove all the special character and white spaces.
3. Removal of Null Values (total 5 data points having null values).

**Data visualization results:**

****

1. Severely imbalanced data.

2. Most of the Gene and Variation feature categories occurs once or twice.

**Methods tried to Encode the Gene, variation and text features:**

1. Response coding.

2. One Hot Encoding.

3. Tf-idf encoding.

4. Average Word2Vec.

**Built the Random model and calculated the multiclass log loss to set it as worst case. (multiclass-log loss= 2.3 approx.) Our final model should give much less log loss.**

**Univariate analysis of Gene feature:**

1. **Out of 2121** train data points there are **236 distinct Gene categories**.
2. Cumulative distribution graph showed that **top 50 gene categories constitute the 78% of the total Genes.** Which means that the remaining **186 Gene categories occurs ones or twice**
3. When build the model with only Gene feature, multi-class log loss came out to be around 1.12 for OneHot, Tfidf and Response coding techniques. Which means that the Gene feature definitely adds value to our data.
4. **Gene feature** is **stable** across train, cv and test data.

**Univariate analysis of Variation feature:**

1. **Out of 2121** train data points there are **1940 distinct Variation categories.**
2. Cumulative distribution graph showed that very few Variation feature categories occur frequently rest of them occurs once or twice.
3. When build the model with only Variation feature, multi-class log loss came out to be around 1.5 for OneHot and Tfidf encoding, and 0.45 for response encoding. Which means that the Variation feature definitely adds value to our data.
4. **Variation feature** is **highly unstable** across train, cv and test dataset. 1. In test data ,78 points out of total 664 test data points are present in train dataset: **11.2%**. In cross validation data 61 out of total 531 cv points are present in train datapoints**: 11.48%.**

**Univariate analysis of TEXT feature:**

1. **Total unique words** present are **53252** in whole train corpus.
2. When build the model with only Text feature, multi-class log loss came out to be around 1.1 for OneHot, Tfidf and Response coding techniques. Which means that the TEXT feature definitely adds value to our data.
3. **TEXT feature** is **stable** across train data, cv data and test data. **95.632 % of word of test** **data** appeared **in train data** and **97.576 % of word of Cross Validation** appeared **in train data**

**Models Tried:**

|  |  |  |
| --- | --- | --- |
|  | **Models** | **Multi-Class Log loss** |
| 1 | **KNN** Model with **response** Coding | 0.93 |
| 2 | **Logistic Regression** without Class balanced and **OneHot** Encoded feature | 0.99 |
| 3 | **Logistic Regression** with Class balanced and **Tf-idf** Encoded feature | 1 |
| 4 | **Logistic Regression** without Class balanced and **OneHot** Encoded feature | 1.01 |
| 5 | **Logistic Regression** with Class balanced and **OneHot** Encoded feature | 1.11 |
| 6 | **Naïve Bayes** Model with **OneHot** Encoded feature (**Baseline Model**) | 1.12 |
| 7 | **Linear SVM** with **Tf-idf** Encodedfeature | 1.12 |
| 8 | **Random Forest** with **Tf-idf** Encodedfeature | 1.14 |
| 9 | **Linear SVM** with **OneHot** Encodedfeature | 1.17 |
| 10 | **Naïve Bayes** Model with **Tf-idf** Encoded feature | 1.2 |
| 11 | **Random Forest** with **OneHot** Encodedfeature | 1.2 |

**Out of these models, I chose top 4 and used 3 models(Removed KNN) for building the final pipeline whichever gives the least Multi-Class log-loss.**

**In all the techniques I have used GridSearchCv to perform Hyperparameter tunning and cross validation.**

**Deployed the model using two techniques:**

1. Pushed the Model, Flask API file to the AWS EC2 cloud and deployed it.
2. Containerized the Model using Dockers and deployed on Google Kubernetes Engine.