Personalized medicine for cancer treatment

by
Yaswanth Reddy Manukonda (G01337128)
Aravind Kommineni (G01327447)

Abstract

A cancer tumor growth comprises of thousands of hereditary transformations. Indeed, even after progression in innovation, the errand of recognizing hereditary transformations, which go about as driver for the growth of the tumor with passengers (Neutral Genetic Mutations), is yet being done physically. This is a tedious interaction where pathologists decipher each hereditary transformation from the clinical proof physically. These clinical proofs have a total of nine classes, yet the basis of grouping is unclear. The primary point of this exploration is to propose a multiclass classifier to arrange the hereditary changes dependent on clinical proof (i.e., the text depiction of these hereditary transformations) utilizing Natural Language Processing (NLP) procedures. The dataset for this exploration is taken from Kaggle and is given by the Memorial Sloan Kettering Cancer Center (MSKCC). The top-notch scientists and oncologists contribute the dataset. The AI arrangement models, specifically, Naïve Bayes, K Nearest Neighbor, Logistic Regression (balanced and unbalanced), Support Vector Machine, Random Forest Classifier, XGBoost, and LSTM are applied to the sparse matrix (keywords count representation) of text depictions. We have also used CountVectorizer, TfidfVectorizer, and Word2Vec for the transformation of text to a matrix of token counts. The accuracy score of all the proposed classifiers is assessed by utilizing the multi class log loss and confusion matrix. At the end of the project, the exact outcomes show that the XGBoost model has performed better compared to other proposed classifiers with a log loss of 0.913.

1. Introduction

Gene sequencing has rapidly moved from the research domain into the clinical phase. Now a days, lot of research is being conducted to genetically understand the disease and selecting the treatment that is best suitable for an individual based on their gene sequence. Gene mutation is characterized as the interminable variety in the typical DNA sequence that is liable for making up a gene so that the arrangement is unique in relation to the one that is seen in most individuals. These gene mutations have varieties in sizes, and they can impact each DNA. A cancer tumor can have thousands of genetic mutations, but the challenge is to distinguish the mutation that is causing tumor growth. The dataset we are using is from Kaggle where we have a clinical research text data and we need to classify the class of the gene mutation, as of now this is being done manually by the pathologists in the lab. To take personalized medicine to its full potential, MSKCC who hosted this competition have provided this dataset. Further, we will be discussing about the exact problem to be solved, research on the existing solutions and our approach in solving this problem with results obtained.

2. Problem Statement

Currently, the interpretation of genetic mutations that are making the cancer-causing tumor grow is identified manually by the pathologists with the help of research literature for every single genetic mutation. This is very tedious and time-consuming task. So, we need to develop a machine learning model that can read the research literature text given in the dataset and classify the class of a genetic mutation. Once the class of the mutation is identified, the personalized medicine can be developed at ease by the medical experts. As the errors in classifying the gene class could be very costly, our model needs to classify the data with as lowest log loss possible.

3. Literature Review

We have read Research paper titled ***"Gene Expression Classification based on Deep Learning Techniques***" [1]. In this study, for classification they assessed the accuracy for most powerful deep learning's algorithms such as Deep Neural Network, Recurrent Neural Network, Convolutional Neural Network and improved Deep Neural Network with the preprocessing technique. The DNN was improved by adding Dropout to it by which the overfitting problem was overcame. Their results showed that the proposed improved-DNN outperforms the other algorithms among all used datasets with a log loss of 1.231. However, we got even better result with XGBoost.

In the next paper, ***"Gene Mutation Classification Using CNN and BiGRU Network"*** [2], the three RNNs, LSTM, Bi-LSTM and BiGRU, outperform the CNN. For text, word order is very important (the keywords extracted by tf-idf are also arranged in order of frequency size). RNN takes that into account, it can capture long-term dependencies, so the effect is much better than CNN. TEXT-CNN model performs well. It is different from CNN in convolution layer and pooling layer and is more suitable for text prediction and classification. It has been found that the network can be strengthened by adding convolution layers on the cyclic layer. So, they try to use the hybrid neural network model of CNN and RNN. they find that the model works well both in series and in parallel. That is because RNN allows the embedding of sequences and previous words, and CNN can extract local features. If two data files, it is found that the CNN-BiGRU parallel hybrid neural network proposed in this paper outperforms the traditional CNN, TEXT-CNN, LSTM, and other single neural network models and CNN-GRU series hybrid neural network model.

4. Methods and Techniques

4.1. Data Analysis

Our dataset contains the file ***training_text*** with features ID, TEXT. ***training_variants*** file has features ID, GENE, VARIATION and CLASS represented in the tables below. We found that there are 9 unique items in the CLASS feature. As we need to classify the CLASS of the test data to help doctors identifying the personalized medicine. So, our problem is a multi-class classification problem. As part of the analysis, we have also checked if there are any duplicates, or the null values present in the datasets.

	ID	Gene	Variation	Class
0	0	FAM58A	Truncating Mutations	1
1	1	CBL	W802*	2
2	2	CBL	Q249E	2
3	3	CBL	N454D	3
4	4	CBL	L399V	4

training_variants

	ID	TEXT
0	0	Cyclin-dependent kinases (CDKs) …
1	1	Abstract Background Non-small cell …
2	2	Abstract Background Non-small cell …
3	3	Recent evidence has demonstrated that…
4	4	Oncogenic mutations in the monomeric…

training_text

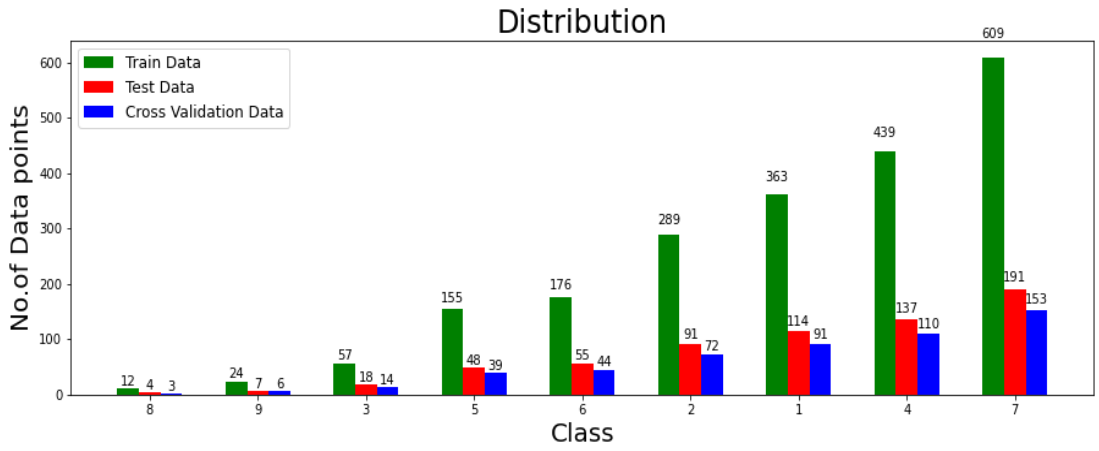
4.2. Data Pre-processing

To achieve less log loss for the model, we need to preprocess the data before sending it to the machine learning models. So, we have used Natural Language Toolkit (NLTK) to perform the preprocessing of the data. We found that the TEXT feature basically consists of the research literature of every class that is being used to predict the class of the genetic mutation manually.

So, we decided to perform some cleaning on this TEXT feature as it has lot of numbers, stop words and unnecessary spaces and indentations. To eliminate these, we have performed tokenization, parsing, classification, stemming, tagging and semantic reasoning available in the NLTK library, which is used to convert human understandable language into Statistical Natural Language processing (NLP, machine understandable language). Now that the TEXT feature has been preprocessed, we have merged the ***training_text*** and ***training_variants*** files with ID feature being the common key between the data points. Now, in the combined dataset, we found that for few of the datapoints the TEXT feature has the empty values. So, we decided to replace these null values with a string that is the concatenation of the features GENE and VARIATION which resulted in the increase of accuracy in predicting the gene CLASS.

4.3. Data Distribution

For the models to be accurate enough we need to eliminate overfitting and underfitting of the data. To achieve this, we need to split the preprocessed train data into 80% of train and 20% of test data. Furthermore, this 80% of train data is split into 80% of train and 20% of cross validation data. Now, the final training dataset after the split contains the known output and the model learns from this data to be generalized to other data later. The "validation dataset" is predominately used to depict the assessment of models when tuning hyperparameters and data preparation, and the "test dataset" is predominately used to portray the assessment of a final tuned model when contrasting it with other final models. We have used stratify parameter in ***train_test_split*** to preserve the ratio of class distribution. Following is the distribution plot obtained after splitting the data.



4.4. Data Evaluation

In our dataset, GENE, VARIATION, and TEXT features are independent and CLASS feature is dependent on the other features. As few of the features have categorical data, we have featurized these categorical variables into One-hot encoding and Response encoding before feeding the data to the model. The problem with One-hot encoding is if the number of distinct values for a categorical feature is large, then One-hot encoding can create sparse and large vectors. So, we decided to 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. There are many ways to estimate how good a feature is, in predicting the class. One of the good methods is to build a proper ML model using just the features you would like to consider for the final model. In this case, we will build a logistic regression model and perform Univariate and Bivariate analysis on the features

GENE, TEXT, and VARIATION to predict the class. After performing all these steps, by using Count Vectorizer over TFIDF and One-hot encoding outperforms all the remaining models.

4.5. Classification Models Used

Now that we have done all the basics required to build a model, we have decided to work on hyper tuning the below Algorithms to see which one outperforms

4.5.1. Naïve Bayes

We have used multinomial naïve bayes that uses Bayesian learning approach which is popular in natural language processing. It is a classification technique based on Bayes theorem. It guesses the tags using bayes theorem.

4.5.2. K Nearest Neighbor

It is a supervised machine learning algorithm that can be used to solve both regression and classification problems that uses feature similarity to predict the values of new data points based on the number of nearest neighbors.

4.5.3. Logistic Regression

This algorithm is based on predictive dissection and the probability concept with the cost function being sigmoid that limits the cost function between 0 and 1 rather than a linear function. It is a probabilistic model which assigns probabilities for events or classes. Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). The binary logistic regression model has the pass or fail class representations. If the condition is “pass” that sample belongs to same class. If the condition is “failing” the sample belongs to another class.[3]

4.5.4. Support Vector Machine

It is a supervised learning framework which is used in classification and regression task analysis of data that is more powerful and flexible. It divides the datasets into classes to find a maximum marginal hyperplane (MMH). It is a non-probabilistic method of classification algorithm. Despite modelling the data into the labeled classes based on the probability of each class, SVM maps them as points in space. The points belonging to one class are closer and there will be a clear gap for points which are of different classes.[4]

4.5.5. Random Forest Classifier

It is a classification algorithm that consists of many decision trees, that uses bagging and features randomness. It takes the average of the subsets to improve the predictive accuracy of that dataset.

4.5.6. XGBoost

Extreme Gradient Boosting, otherwise called XGBoost, is an ensemble ML algorithm that depends on decision trees. It uses a gradient boosting algorithm, where new models are created to work out the residuals or errors of past models and afterward summarized to

deliver the final prediction. It utilizes a calculation of gradient descent to lessen the loss while presenting new models

4.5.7.LSTM

It is a type of recurrent neural network that can be interpreted as sequence of neural networks linked in chain manner. It is an extension to Recurrent Neural networks, the main issue of RNN is the need for large storage to remember many states and computations performed in the previous stage. As a remedy LSTM is activated with forget and remember gates. During the training the network evaluates the impacts of the previous and current states. The current input and the states that have impact on the output are kept while the rest are discarded.

5. Discussion and Results

5.1. Dataset

The dataset used for this project is provided by the Memorial Sloan Kettering Cancer Center (MSKCC). The top-notch scientists and oncologists contribute towards this dataset. This dataset is available in Kaggle [5].

5.2. Evaluation Metrics

Along with log loss, there are bunch of other methods to evaluate the performance of a classification model.

Log Loss: it considers the uncertainty of your prediction based on how much it varies from actual label. This gives us a more nuanced view of the performance of our model.

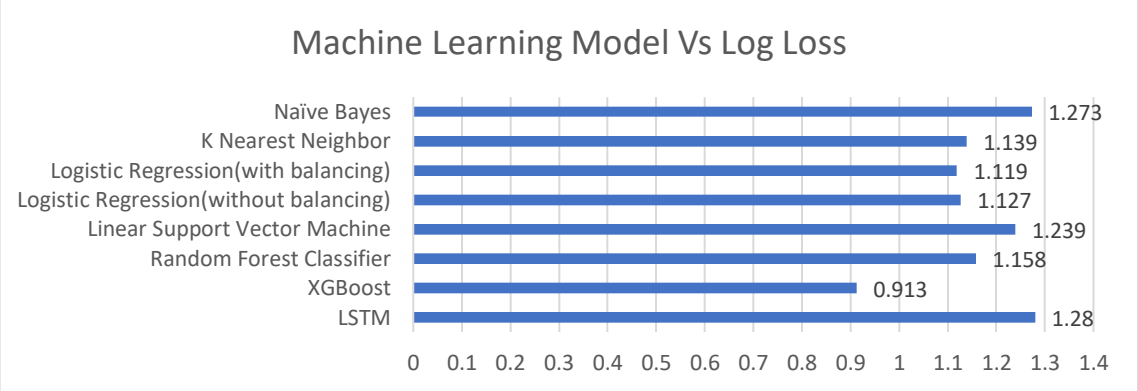
Confusion matrix: it's a table showing how good our model is at predicting examples of various classes. The axes of a confusion matrix would be actual class and predicted class.

$$\boldsymbol{Recall} = \frac{true\ positive}{true\ positive + false\ negative}$$

$$\boldsymbol{Precision} = \frac{true\ positive}{true\ positive + false\ positive}$$

		Predicted condition	
Actual condition		True positive	False negative
		False positive	True negative

5.3. Experimental Results



We have used the text transformation models, CountVectorizer, and TfidfVectorizer for the conversion of text to a matrix of token counts.

Count Vectorizer: The CountVectorizer class from the sklearn library's feature_extraction module is used to convert the text of clinical evidence to a matrix of token counts. It counts the number of times each word appears in a particular text. The accuracy score generated by the confusion matrix is then used to train few of the machine learning classifiers we used.

TfidfVectorizer: the term frequency-inverse document frequency also belongs to the feature_extraction module of the sklearn library is used to convert the text of clinical evidence to a matrix of token counts. TFIDF can normalize the word count in any document against the total number of documents containing that word in the entire corpus. The accuracy score generated by the confusion matrix is then used to train few of the machine learning classifiers we used.

The matrix of token counts obtained either from CountVectorizer or TfidfVectorizer for GENE, VARIATION and TEXT along with the One-hot encoded or Response encoded categorical values are being fed to the models and hyper tuned the models to attain the best lowest log loss possible.

5.3.1. Naïve Bayes

We have used MultinomialNB classifier that uses discrete counts. We have a text classification problem where we can consider Bernoulli trials which is one step further and instead of word occurrence, we count how often the word occurs in the document. With alpha (additive Laplace smoothing) = 0.1 gave us the lowest log loss.

5.3.2. K Nearest Neighbor

It uses the k nearest neighbor votes to predict the class, with the parameter n_neighbors (number of neighbors to use) =11 gave us the lowest log loss.

5.3.3. Logistic Regression

In this model, we used SGD (stochastic gradient descent) classifier over the existing logistic regression classifier in sklearn as it allows minibatch(online/out-of-core) learning. Therefore, it makes sense to use Stochastic Gradient Descent for large scale problem where it is efficient than the traditional logistic regression classifier. We have worked on both balanced and unbalanced data with the parameter of the classifier being alpha (the constant that multiplies the regularization term, the stronger the regularization the higher the value of alpha)=0.001, penalty (regularization term)='l2', loss (logistic regression)='log', random_state (used to shuffle the data)=42 gives us the optimum performance with the log loss of 1.119 in case of balanced class weight and a log loss of 1.127 in case of unbalanced class weight.

5.3.4. Support Vector Machine

We have attained a log loss of 1.239 in the SVM after hyper tuning the parameters to alpha (constant that multiplies the regularization term)=0.001, penalty (regularization term)='l2', loss(linear support vector machine)='hinge', random_state(helps shuffling the data)=42.

5.3.5. Random Forest Classifier

The count vectors obtained from the sparse matrix are fitted, the optimum values of the various parameters to achieve the optimum performance of the model are as follows: *n*_estimators = 2000, criterion='gini', max_depth=10, random_state=42, n_jobs=-1

5.3.6. LSTM

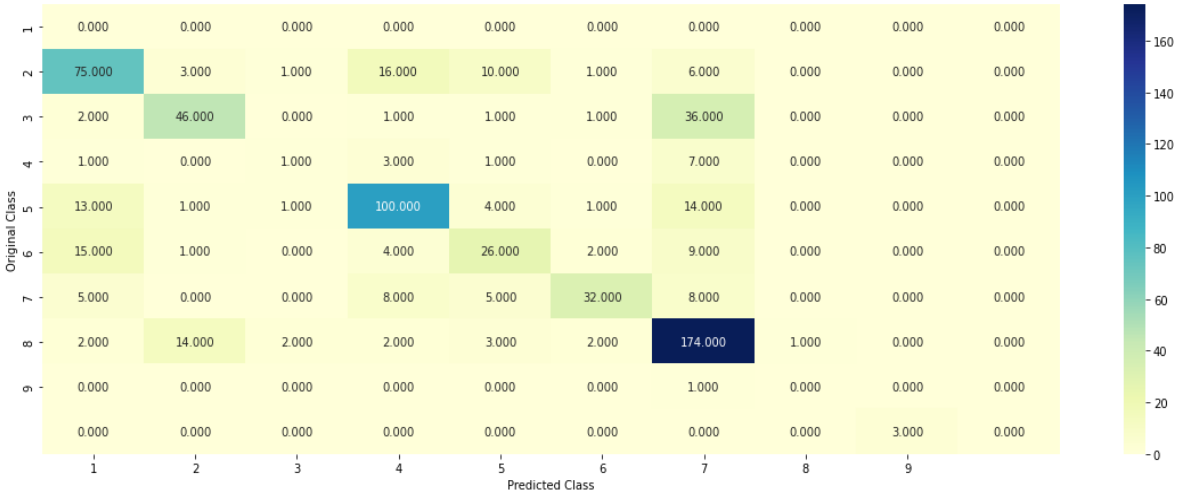
For multiple classification tasks, LSTMs have lately showed encouraging results in learning the long-term dependencies of sequences. We've created an LSTM model in this project. The suggested model is implemented in Keras, and performance is tested using multi-class log loss. There are 3321 entries in the dataset, with 2988 being used for training and 333 being used for testing. Variations in embedding size, number of neurons, and dropout rate are used to create this model. With 8 epochs and a batch size of 32, the model is being trained. The input data is processed one by one by the LSTM model. The test loss of 1.09 is also higher than the train loss of 0.56, according to the results. This is owing to the model's excessive fit. We also found that when the data size is minimal, the LSTM model overfits. When the training data amount is larger, the LSTM performance improves. The log loss seems to be increasing when the embedding size and number of neurons are less.

5.3.7. XGBoost Classifier

The count vectors resulting from the sparse matrix, and One-hot encoding of the gene attribute are fitted in the XGBoost classification method, and the log loss for this model is calculated by hyper tuning various parameters to achieve the best performance. After multiple considerations, we have achieved the lowest log loss for the following optimum values of parameters: eta (learning rate) = 0.05, minimum loss reduction = 0.4, max depth (tree maximum depth) = 5.

As part of text transformation, we have tried using both the CountVectorizer and TfidfVectorizer. However, the log-loss is a bit high if TfidfVectorizer is used as text transformation when compared to CountVectorizer.

We have achieved a log loss of 0.91 with the XGBoost classifier's and the CountVectorizer as text transformation is used which outperforms the rest of the classifiers.



6. Conclusion

This project is carried out to propose a multiclass classifier to classify the genetic mutations based on the clinical evidence (TEXT). Over time and with consistent literature review and clinical validity curation, we expect that many patients with candidate genetic etiologies will receive a definitive diagnosis via reclassification reports. Our machine learning model, and analysis highlights the importance of careful literature curation and evaluation using a system of clinical validity scoring optimized for use in a diagnostic laboratory. The minimum value for log loss is obtained on implementing XGBoost on Gene, Variation and Text data value points.

6.1. Directions for Future Work

The proposed model can be enhanced in the future by incorporating the other text transformation models like truncated singular value decomposition (SVD) and Doc2Vec for the text conversion. We can also use the advance recurrent neural networks to obtain better results. The similar gene mutation and classification techniques can be extended to find a cure for the diseases other than cancer and can be a breakthrough in a personalized medical space.

References

1. *O. Ahmed and A. Brifcani, "Gene Expression Classification Based on Deep Learning," 2019 4th Scientific International Conference Najaf (SICN), 2019, pp. 145-149, doi: 10.1109/SICN47020.2019.9019357.*
2. *J. Xu, X. Zheng and M. Jiang, "Gene Mutation Classification Using CNN and BiGRU Network," 2019 9th International Conference on Information Science and Technology (ICIST), 2019, pp. 397-401, doi: 10.1109/ICIST.2019.8836846.*
3. B*rownlee, J. (2020, August 14). Logistic regression for machine learning. Machine Learning Mastery. Retrieved December 02, 2021, from https://machinelearningmastery.com/logistic-regression-for-machine-learning/.*
4. *SVMstruct. SVM-Struct Support Vector Machine for Complex Outputs. (n.d.). Retrieved December 12, 2021, from https://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html.*
5. *Personalized medicine: Redefining cancer treatment*. Kaggle. (n.d.). Retrieved December 01, 2021, from https://www.kaggle.com/c/msk-redefining-cancer-treatment/data.