Computational Journal of Mathematical and Statistical Sciences
3(2), 280–315
DOI:10.21608/CJMSS.2024.267064.1040
https://cjmss.journals.ekb.eg/

Research article

Multi-Class Classification of Genetic Mutation Using Machine Learning Models

Barikisu Ntiwaa Ankrah[1*], **Lewis Brew**[1] **and Joseph Acquah** [1]

[1] Department of Mathematical Sciences, Faculty of Engineering, University of Mines and Technology, Tarkwa, Ghana.

* **Correspondence:** pg-bnankrah9021@st.umat.edu.gh

Abstract: The challenge of distinguishing genetic mutations that contribute to tumor growth is crucial in cancer treatment. Cancer is responsible for millions of deaths annually, hence the need for early detection of tumors to improve treatment efficacy and survival rates. However, manual classification is prone to errors and inefficiencies due to human limitations and the complexity of domain knowledge, leading to time-intensive processes. In response, machine learning models improve accuracy and efficiency for cancer prognosis and prediction. However, the lack of theoretical understanding of algorithms may limit the interpretability and applicability of results, where insights into model prediction are crucial to making informed decisions, especially in the biomedical domain. To address these challenges, our study employed four supervised machine learning algorithms, namely Support Vector Machine (SVM), Naïve Bayes (NB), Logistic Regression (LR), and Random Forest (RF). The performance of these algorithms was assessed using log-loss and misclassification rates. Logistic regression emerged as the optimal classifier with a log loss of 1.0125 and a misclassification rate of 30.97%.

Keywords: Logistic Regression, Cancer, Term Frequency Inverse Document Frequency (TF-IDF), One-hot encoding, Log loss.
Mathematics Subject Classification: 62J12, 03C45
Received: 3 February 2024; **Revised:** 29 March 2024; **Accepted:** 4 April 2024; **Published:** 26 April 2024.
Copyright: © 2024 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Advancements in genomics and bioinformatics have revolutionized our understanding of genetic mutations and their potential implications for human health and disease. Gene mutations are fundamental genetic changes that can alter the structure and function of genes and play a vital role in various

biological processes, including diseases such as cancer, drug response, and evolutionary adaptations. Cancer is responsible for most fatalities in the developed world and ranks second in the developing world, causing a loss of nearly 8 million lives annually [24].
According to [31], cancer is the uncontrolled growth and spread of abnormal cells in the body, which can form tumors. Tumors are either cancerous (malignant) or non-cancerous (benign). Malignant tumors can invade surrounding tissues and spread to other parts of the body, while benign tumors typically do not spread and can be removed, as shown in Figure 1. Cancer can occur almost anywhere in the body and is caused by disruptions in the process of cell division, often due to genetic mutations. These alterations in cell division are illustrated in Figure 2.

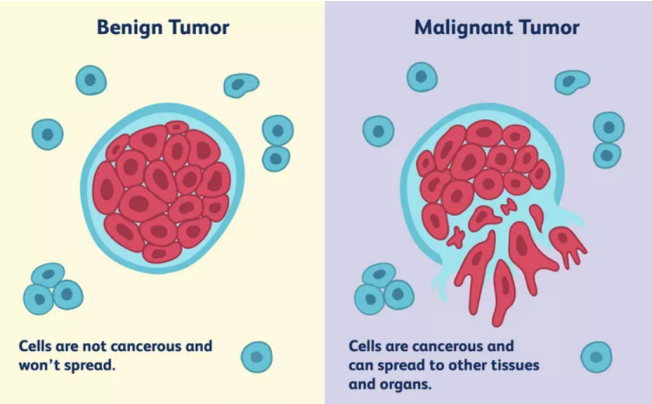


Figure 1. Benign vs Malignant Tumor [4]

[7] defines genetic mutations as alterations in the DNA sequence that occur randomly, either due to environmental factors or inherited from birth, and can be categorised into two main types. The first type is hereditary or germline mutation, where inherited variants are passed from parent to child and are present throughout a person's life in virtually every cell in the body. The second type, known as somatic mutation is acquired during a person's lifetime due to factors such as ultraviolet light, X-rays, cigarette smoke, and copying mistakes during cell division. Somatic mutation cannot be transmitted to offspring as they occur after conception and in cells other than sperm or eggs (somatic cells). Other types of genetic mutations are deletion, duplication, insertion, translocation, inversion, and frameshift among others.

Understanding genetic variations is of paramount importance to unraveling the intricate complexities of biological processes, disease, and drug response [32], thereby facilitating personalized medicine. Personalized medicine can lead to more comfortable cancer treatment for patients by utilizing information about the genetic composition of their tumor [14]. This knowledge allows for a more informed understanding of which treatments are less likely to cause adverse side effects [65]. Gene expression data generally comprise a huge number of genes, yet not all of them are associated with cancer. Therefore, the need to classify cancer tumors accurately becomes crucial in cancer treatment. Researchers have carefully examined the difficulties in classifying cancer by making use of data mining techniques, statistical procedures, and machine learning algorithms to effectively analyse the information [16].

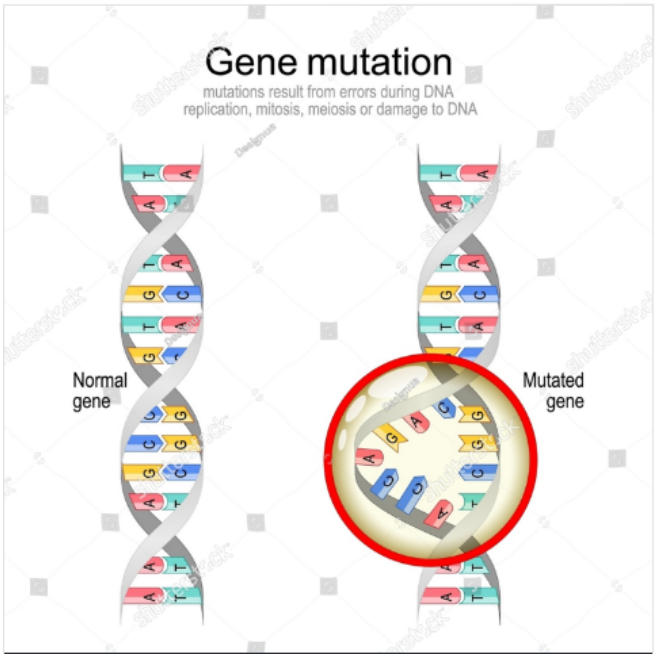


Figure 2. Normal vs. Mutated Gene [57]

[46] and [61] affirmed the intensification of using machine learning algorithms for molecular classification of tumors in recent years since these algorithms can analyze large amounts of genetic data to identify patterns and characteristics associated with cancer. Using these informations, the algorithms can classify tumors as benign or malignant and subsequently identify specific subtypes of cancer, such as breast cancer, lung cancer, etc.

The advent of machine learning models in this domain has presented novel opportunities for accurate and efficient classification of gene mutations [18, 6, 34, 41], paving the way for transformative breakthroughs in personalized medicine and genetic research.
A branch of artificial intelligence (AI) called machine learning (ML) enables algorithms to learn from previous datasets (training data) by utilizing statistical, probabilistic, and optimization tools to automatically improve its performance, classify new data points, and detect new patterns or trends [8]. Although machine learning relies heavily on statistics and probability, it is fundamentally more powerful, as it allows inferences or decisions that may not be feasible using traditional statistical approaches [44, 27].
According to [42], the choice of algorithm depends on multiple factors, such as the type of problem to be solved, the number of variables involved, and the model that would be the most suitable among others. Hence, there is no universal algorithm that fits all circumstances. Machine learning algorithms are classified primarily based on the intended outcome they aim to achieve according to [27] and [43]. There are two general types of machine learning algorithms; unsupervised and supervised learning. In unsupervised learning, no labelled set of training data is provided, and the output during the learning process is unknown as the algorithm attempts to identify patterns or relationships in the data without any specific guidance or supervision. Supervised learning involves using a labelled set of training

data to create or approximate a function that can map an input data to the desired output [34]. It has been observed that almost all machine learning algorithms used in cancer prediction and prognosis are based on supervised machine learning.

In the domain of text classification, machine learning models emerge as the optimal replacement for conventional approaches [51]. Several significant techniques employed in this field include Nave Bayes, Support Vector Machines (SVM), Decision Trees, J48, K-Nearest Neighbours (KNN), and IBK [20]. According to [56], there has been a surge in interest regarding the automated categorisation (or classification) of texts into predefined categories over the past decade, as this growing enthusiasm can be attributed to the proliferation of digital documents and the subsequent demand for effective organisational solutions.
Natural language processing (NLP) technology enables computers to interact with humans through deep learning and linguistic analysis techniques, extracting knowledge from unstructured text [19]. Furthermore, with the progression of technological innovations, text classification and document categorisation have found widespread application in various domains, covering fields such as medicine [35, 3], social sciences [49, 48, 47], healthcare [38, 50], business, and marketing [69], and law [62].

According to [35], medical coding involves the assignment of medical diagnoses to distinct class values extracted from an extensive array of categories, and text classification techniques hold substantial promise and value in performing such tasks. As online information continues to expand rapidly, especially in text form, the practice of text classification has emerged as a crucial method to efficiently organize text data [30].
However, manual classification is prone to numerous inaccuracies due to human errors and a lack of understanding of domain knowledge, as indicated by [51] and also very time-consuming [26]. The use of computer-related technology in medical diagnostics has improved physicians' ability to effectively diagnose diseases and analyse patient physiological data using innovative signal processing techniques [71, 72].

The remainder of the article is structured as follows. Section 2 presents the review of relevant literature. Section 3 describes the data and methods used. The description and pre-processing of the data are performed in Section 4. The development of model and evaluation are presented in Section 5 followed by the discussion of the results in Section 6. Finally, the conclusion and recommendations of the study are captured in Section 7.

2. Related Literature

Several studies have explored the use of various natural processing techniques and machine learning algorithms such as the study by [18]. The researchers explored the application of machine learning in cancer diagnosis, detection, prognosis, and prediction. Their study highlighted the growing trend of personalized predictive medicine in cancer care and covered a wide range of machine learning methods, types of data, and the performance of these methods in cancer prediction and prognosis.
The study further stated that, while some studies lack appropriate validation or testing, others were well-designed and validated. [18] demonstrated machine learning methods can significantly

improve precision (15- 25%) in predicting cancer susceptibility, recurrence, and mortality. And also emphasized the potential of machine learning in advancing cancer research and clinical practice, opening avenues for improved diagnosis, treatment, and patient outcomes.

Furthermore, the study by [68] addressed the need to distinguish between triple-negative breast cancer (TNBC) and non-triple negative breast cancer, as TNBC is the most aggressive and lethal form of breast cancer. The researchers proposed the use of a machine learning (ML) approach to classify breast cancer patients, based on gene expression data. To develop and validate the classification models, [68] analyzed RNA-Sequence data from 110 TNBC and 992 non-TNBC tumor samples from The Cancer Genome Atlas. They selected specific genes as features for training the ML models and evaluated four different algorithms: Support Vector Machines, K-nearest neighbour, Naïve Bayes, and Decision tree. Among the four ML algorithms tested, the Support Vector Machine (SVM) algorithm outperformed the others in accurately classifying breast cancer into TNBC and non-TNBC categories, with fewer misclassification errors. The results indicate that ML algorithms, particularly SVM, are efficient and effective in classifying breast cancer patients based on gene expression data, distinguishing between TNBC and non-TNBC subtypes. The researchers' approach showed machine learning models can positively contribute to precision medicine in the clinical management of breast cancer, helping to tailor treatment strategies based on molecular types and subtypes of the disease.

Subsequently, [54] discussed the significant progress made in the detection and treatment of cancer using machine assistance over the past few decades. Their study presented a systematic review of various techniques used in the diagnosis and cure of several types of cancer that affect the human body and focused on six types of cancer, namely lung cancer, breast cancer, brain tumor, liver cancer, leukemia, and skin cancer. The researchers categorized the methodologies used in each case and highlighted existing limitations. The four primary stages of automated cancer diagnosis discussed were image pre-processing, tumor segmentation, feature extraction, and classification using benchmark datasets.
The study provided valuable insights to new researchers entering the field of cancer detection and diagnosis, by offering a comprehensive review of current state-of-the-art machine-assisted techniques along with their advantages and disadvantages. However, [54] also acknowledged despite the progress made, the accuracy of cancer detection methods for each cancer category is still not at its peak. Many researchers have not used benchmark datasets or used small datasets to test their techniques and emphasized the importance of using benchmark datasets. Finally, their study sheds light on the need for continued research and development in the field to improve cancer diagnosis and treatment outcomes.

In addition, [50] integrated natural language processing (NLP) techniques into the development of conversational health diagnosis systems to enhance patient's access to medical information through text. Their study presented creating a chatbot service, named CUDoctor, within the Covenant University Doctor telehealth system. The chatbot utilised fuzzy logic rules and fuzzy inference to assess the symptoms of tropical diseases in Nigeria. Leveraging the Telegram Bot Application Programming Interface (API) and Twilio API, the chatbot established connectivity with users via both messaging interfaces and short messaging service (SMS).

The system's knowledge base draws from established medical ontologies, encompassing disease-symptom associations. Fuzzy support vector machine (SVM) techniques were used to predict diseases based on symptoms entered by the user. Natural language processing interprets user input, channelling it to CUDoctor for decision support. The researchers' diagnostic process ended with the transmission of a notification message to the user.
The resulting system embodied a personalised medical diagnosis approach, using user input data to effectively identify the disease. Usability assessment, measured using the system usability scale (SUS), demonstrated a favorable mean score of 80.4, underscoring the system's positive evaluation and usability. The study by [50] underscored the potential of NLP-powered chatbot systems to facilitate medical diagnoses and foster improved patient interactions with healthcare resources.

Further, [25] also addressed the challenging task of manually distinguishing genetic mutations in cancer tumors that act as drivers for tumor growth from genetic mutations known as passengers which is time-consuming and involves pathologists interpreting clinical evidence related to genetic mutations, belonging to nine different classes.
To automate this classification process, the researchers proposed a multiclass classifier that uses Natural Language Processing (NLP) techniques to classify genetic mutations based on clinical evidence, which is in the form of text descriptions. Three text transformation models, namely Count Vectorizer, Tfidf Vectorizer, and Word2Vec, were employed to convert the text descriptions into a matrix of token counts and used three machine learning classification models (Logistic Regression, Random Forest, and XGBoost) along with a Recurrent Neural Network (RNN) model from deep learning. The researchers evaluated the accuracy scores of all proposed classifiers using the accuracy score from the confusion matrix. [25] results showed that, the RNN model of deep learning outperforms the other proposed classifiers, achieving the highest accuracy of 70% in classifying genetic mutations based on clinical evidence.
Their research highlighted the potential of using NLP techniques and machine learning classifiers, particularly the RNN model, to automate and improve the accuracy of genetic mutation classification in cancer tumors based on clinical evidence.

Several studies such as [37, 25, 2] have explored gene mutation classification using machine learning and deep learning to improve classification accuracy but these studies lack a thorough insight into how the model performs. Recognising the importance of model interpretability in biomedical applications, this study aims to bridge the gap between high performance and transparency in gene mutation classification through the clinical literature using machine learning models. By creating a multiclass classifier with enhanced interpretability, the study aims to provide pathologists with insights into the reasons behind the model predictions. This transparency not only aids in efficient gene mutation classification but also minimises misdiagnosis rates, potentially saving patients' lives and reducing adverse consequences.

To achieve these goals, the study adopts a hybrid approach, integrating categorical and text transformation techniques. These methods, combined with four well-established machine learning classifiers: Logistic regression, Naive Bayes, Support Vector Machines (SVM), and Random Forest aim to create a model that not only exhibits high efficiency but also ensures the pathologist's ability to comprehend

and trust its decisions. Through this comprehensive approach, the study aspires to contribute to the field by offering a multiclass classification solution that bridges the gap between performance and interpretability in genetic mutation classification.

3. Methods Used

The model training was done in five stages: (a) data cleaning and exploratory; (b) feature transformation or extraction; (c) train-text split (d) classifier training; and finally (e) evaluation of the trained classifier to select the best classifier. Below are the proposed algorithm steps followed by the framework for the study illustrated in Figure 3

Proposed Algorithm Steps

- Data Collection: Obtain cancer patients dataset from Kaggle.
- Data Preprocessing: Preprocess the data by removing punctuation and special characters, eliminating stopwords, and converting all text to lowercase.
- Data Splitting: Divide the dataset into train, cross-validation, and test sets in the ratios 48:12:40, 56:14:30, and 60:20:20 such the ratio with the best performance is selected.
- Exploratory Data Analysis: Explore train, cross-validation and test dataset under each ratio to know the number of observations in that dataset and how the features are distributed.
- Feature Extraction: Gene, Variation, and Text features were extracted and transformed into numerical values such that they can be used for model training.
- Classifier Training: Train selected clasifiers namely Naive Bayes, Logistic Regression, Support Vector Machine and Random Forest with the train data. Then cross-validate with cross-validation data in hyperparameter tuning to choose the best parameter value.
- Classifier Evaluation: Evaluate the trained classifier with the test data using evaluation metrics like log loss and misclassification rate to select the best classifier.

3.1. Data Cleaning and Exploratory

In various algorithms, particularly those involving statistics and probabilities, the presence of noise and unnecessary features can detrimentally impact the performance of the system [35]. Therefore, the following data-cleaning techniques were employed to preprocess and prepare text data for analysis. Tokenization which is a fundamental step in text pre-processing, was used to convert the continuous stream of text into manageable units. Also, all the text were converted to lowercase which helps in standardising the data and ensures that the model does not treat the same word with different cases as distinct.

Stop words are common words such as; (the, is, and, after) that occur frequently in text but usually do not contribute much to the meaning. All stop words were removed as mentioned in the work of [53] to reduce noise in the data and improve processing speed. In addition, punctuation and special characters were all removed

Subsequently, exploratory data analysis (EDA) was performed to visually and statistically explore the data set to gain insight, discover patterns, and understand the underlying structure of the data.

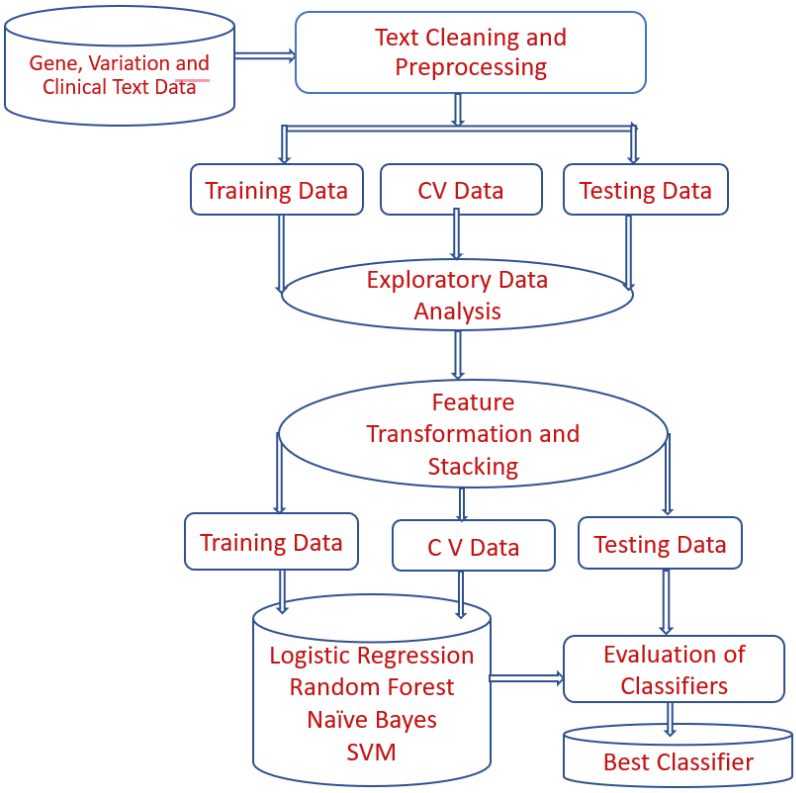


Figure 3. Framework for The Study

Therefore, stratified sampling was used in data splitting since the distribution of the outcome variable (class) was unbalanced.
The study used 60: 40, 70: 30, and 80: 20 train/test ratios, where a larger portion of the data set (training data) was used to train the classifier. We used holdout cross-validation where the validation set was used for initial model assessment and hyperparameter tuning, while the test set was used to get a final evaluation of the model's performance after all tuning was performed. The separation of data into training, validation, and test sets was done to prevent overfitting which occurs when a model learns to perform exceptionally well on the training data but does not generalize well to unseen data.
Univariate feature analysis was used to examine each feature separately to determine how significant they are in predicting class labels as indicated in the work of [55].

3.2. Feature Transformation or Feature Extraction

The categorical features (Gene and variation) were transformed using one-hot encoding or response coding while text features were transformed using TFIDF or Bag of words. Feature transformation was performed to transform categorical and text features into numerical vector formats that can be used by machine learning algorithms.

One-hot encoding involves generating new variables that represent the original categories using binary values of 0 or 1 [12].

Response coding is defined as a technique for converting categorical data that involves calculating the probability of a data point belonging to a specific class given a category [58]. This is expressed mathematically as ;

$$P(class = Y|category = A) = \frac{P(category = A \cap class = Y)}{P(category = A)} \tag{3.1}$$

where;
Y = class label
A = category of gene or variation

Count Vectorizer serves as a prevalent text preprocessing method in natural language processing (NLP) applications. It is employed to transform a set of textual documents into a numerical representation. It involves tallying the occurrences of each word within a document [60]. Count vectorizer offers a clear method for tokenizing a set of text documents, creating a vocabulary of known words, and encoding new documents using that vocabulary[5, 17, 23].

Term Frequency Inverse Document Frequency (TF-IDF)
TF-IDF is a measure used to assess the significance of a word in a document within a given collection or corpus[13]. It employs a metric that gauges the frequency of words within the documents, and the word count is adjusted based on this metric [37]. Term Frequency (TF) is a metric that indicates the frequency with which a term appears within a document. Since documents can vary in length, a term can occur more frequently in longer documents than in shorter ones. This is expressed mathematically as;

$$TF = \frac{Number\ of\ times\ the\ term\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document} \tag{3.2}$$

Inverse Document Frequency (IDF) is a measure of the significance of a term. When calculating TF, all terms are given equal importance. However, it is recognized that certain terms, such as ”the”, ”of”, and ”is”, may appear frequently but have little value in terms of meaning. As a result, it is necessary to decrease the weight of the frequent terms while increasing the weight of the rare ones. This is achieved through the following computation:

$$IDF = \frac{Number\ of\ the\ document\ in\ the\ corpus}{Number\ of\ document\ in\ the\ corpus\ contain\ the\ term} \tag{3.3}$$

The mathematical representation of the weight of a term in a document By utilizing this approach, we can impose a penalty on words that have high frequency. This is achieved by multiplying two metrics: the count of a word in a document and the inverse document frequency of the word across a set of documents [70, 59] as expressed in Equation (3.4).

$$W(d,t) = TF(d,t) * \log\left(\frac{N}{df\,(t)}\right) \tag{3.4}$$

The number of documents, N, is represented by df (t), which is the count of documents containing the term t within the corpus. The first part of Equation (3.4) increases recall, while the second part increases the precision of the word embedding [59].

3.3. Supervised Learning Algorithm

The study used supervised learning, a type of machine learning algorithm since the data set is labelled, and also a classification task because the target variable is categorical (classes 1 - 9). Hence, we are dealing with a multi-class classification problem using the following algorithms .

Logistic Regression

Logistic regression is a supervised machine learning algorithm that performs classification tasks by predicting the probability of an outcome, event, or observation [1]. Logistic regression can also analyze the relationship between one or more independent variables and classifies the data into discrete classes. Based on the number of categories, logistic regression can be classified as binomial, multinomial, or ordinal.

Let $X \in \mathbb{R}^{n \times d}$ in a given data set, the multinomial (or multilabeled) logistic classification uses the probability that x belongs to class i [36] as defined in Equation (3.5).

$$P(y^{(i)} = 1|x, \theta) = \frac{exp\left(\theta^{(i)^T} x\right)}{\sum_{i=1}^{m} exp\left(\theta^{(i)^T} x\right)} \tag{3.5}$$

where;
x = The input feature
$\theta^{(i)}$ = the parameter vector corresponding to class i.
$P(y^{(i)} = 1|x, \theta)$ = The conditional probability that the target variable y takes the value 1 given the input features x. This represents the probability of the outcome being class 1.
$\sum_{i=1}^{m} exp\left(\theta^{(i)^T} x\right)$ = The sum of the exponential inner products of the parameter vectors θ with the input feature x for all classes i.
The denominator term normalizes the probabilities to ensure they sum up to 1.

For binary classification (m = 2) which is known as a basic LR, but for multinomial logistic regression ($m > 2$) usually uses the softmax function [35]. The normalization function is written as;

$$\sum_{i=1}^{m} p\left(y^{(i)} = 1\middle|x, \theta\right) \tag{3.6}$$

For classification task in a supervised learning , the component of θ is calculated from the subset of the training data D which belongs to class i where i $\in 1, .., n\}$.

Naive Bayes Classifier

Naïve Bayes classifier approach is rooted in the principles of Bayes theorem, originally developed by Thomas Bayes during the years 1701–1761 [52, 28]. Numerous types of NB exist, including multinomial NB, Bernoulli NB, and Gaussian NB. Among these, multinomial NB finds extensive application in text classification [51].

The output of the Multinomial Naïve Bayes classifier is a predicted class, c, from a set of k categories, $C = \{c_1, c_2, ..., c_k\}$, when the number of documents (n) is given. The Naïve Bayes algorithm can

be expressed as:

$$P(c|d) = \frac{P(c)\prod_{w\in d} P(d|c)^{n_{wd}}}{P(d)} \tag{3.7}$$

$P(c|d)$ represents the probability of class **c** given the document d, while c is the prior probability of class c, and $P(d)$ is the probability of document d. The number of times a word, w appears in a document, d is denoted as n_{wd}. The conditional probability of document d given class c, is represented as $P(d|c)$.

The Laplace-smoothed version of the Multinomial Naïve Bayes algorithm is expressed as $P(w|c)$ and is given by equation. $P(w|c)$ is calculated as the proportion of times word w appears in documents of class c [22].

$$P(w|c) = \frac{1 + \sum_{d\in D_c} n_w d}{k + \sum_{w'} \sum_{d\in D_c} n_{w'} d} \tag{3.8}$$

where D_c is the collection of all training documents in class c, and k is a smoothing parameter. The number of times a word, w appears in a document, d is denoted as n_{wd} while $\sum_{w'} \sum_{d\in D_c} n_{w'} d$ represents the total count of all words in all documents in class c.

k is the size of the vocabulary (i.e. the number of distinct words in all training documents). The additional one in the numerator is the Laplace correction and corresponds to initializing each word count to one instead of zero. It requires the addition of k in the denominator to obtain a probability distribution that sums to one. This kind of correction is necessary because of the zero-frequency problem, a single word in test document d that does not occur in any training document about a particular category c will otherwise render $P(c|d)$ zero.

Support Vector Machine (SVM)
[64] developed the initial version of Support Vector Machines in 1963. In the early 1990s, [10] introduced a nonlinear variation of this model. Originally designed for binary classification tasks, SVM has been widely adopted for tackling multiclass problems through various research efforts [9]. They are popular due to their ability to handle high-dimensional data and their versatility in handling both linear and nonlinear data.
Multi-class SVM: Given that SVMs have traditionally been employed for binary classification, an extension called Multiple-SVM (MSVM) is employed to handle multi-class problems [45]. In the One-vs-Rest approach, N binary classifiers are trained, each representing one class versus the rest of the classes. The prediction for a new sample x is made by all binary classifiers, and the class with the highest confidence (i.e., the highest decision score) is selected as the final prediction.

The decision boundary equation for the One-vs-Rest approach is expressed mathematically as

$$f(x) = w_i x + b_i \tag{3.9}$$

Where: x is the input feature vector representing a data point to be classified.
w_i This is the weight vector corresponding to the i-th class in the SVM classifier.

b_i is the bias term or intercept corresponding to the i-th class in the SVM classifier.

Points lying on the decision boundary satisfy the equation $w_i x + b_i = 0$, and their classification depends on which side of the boundary they fall.

Addressing the i-class issue basically involves developing a decision function encompassing all i-classes alongside [66, 15]. Generally, a multiclass SVM can be formulated as an optimization problem with the following structure;

$$\min_{w_1,w_2,....,w_k,} \zeta \frac{1}{2} \sum_k w_k^T w_k + C \sum_{(x_i y_i) \in D} \zeta_i \tag{3.10}$$

$$st. \;\; w_{y_i}^T x - w_k^T x \leq i - \zeta_{i,} \tag{3.11}$$

$\forall (x_i, y_i) \in D, k \in 1, 2, .., K, k \neq y_i$

The training data points (x_i, y_i) are part of a dataset D, and C is a penalty parameter. Additionally, ζ is a slack parameter introduced to handle misclassifications or instances that fall within the margin. $w_1, w_2, ..., w_k$, are the weight vectors associated with each class k. The margin associated with the correct class y_i for the i-th instance is given by $w_{y_i}^T x$ and the margin associated with class k for the i-th instance is also given by $w_k^T x$.

Random Forest

Random forest is one of the most popular and powerful machine learning algorithms. It is a type of ensemble machine learning algorithm called Bootstrap Aggregation or Bagging. In 1995, T. Kam Ho developed a method that utilised t trees in parallel [29]. [11] later enhanced it, demonstrating convergence for RF as shown in Figure 4 with margin measures mg (X, Y):

$$mg\,(X, Y) = av_k I\,(h_k\,(X) = Y) - max\; av_k I\,(h_k\,(X) = J)\,,\; j \neq i \tag{3.12}$$

where $mg\,(X, Y)$ denotes the margin for a given instance X with true label Y, av_k represents the average over all decision trees k in the random forest ensemble and $I\,(h_k\,(X) = Y)$ is an indicator function that equals 1 if the prediction of the k-th decision tree h_k for the instance X matches the true label Y, and 0 otherwise. $max\; av_k I\,(h_k\,(X) = J)$ denotes the maximum average probability of incorrect predictions for labels J other than the true label Y

Once all trees in the forest have been trained, the predictions are determined through a voting process, as outlined in the work of [67]. The final classification result is decided by a majority vote by all DTs and it is expressed in Equation (3.13).

$$\delta v\; arg\max_i \sum_{j:j \neq j} I_{\{r_{ij} > r_{ji}\}} \tag{3.13}$$

Such that $r_{ij} + r_{ji} = 1$

Where;

δv is the final classification result.

$arg\max_i$ represents the class index that maximises the expression.

i and j are indices representing different classes.

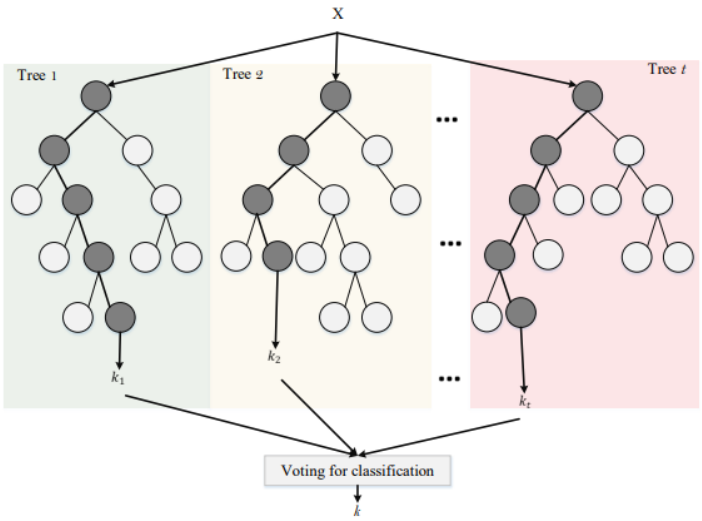


Figure 4. Random Forest [11]

r_{ij} is a measure of the number of times class i is chosen over class j in the majority votes of the decision trees.
$I_{\{r_{ij}>r_{ji}\}}$ is an indicator function that equals 1 if $r_{ij} > r_{ji}$ and 0 otherwise.

3.4. Model Evaluation Metric

An evaluation metric in machine learning is a measure used to evaluate the performance of a machine learning model.

Logarithmic Loss (Log loss)
Log loss, also called cross-entropy loss, is one of the most reliable single number metrics that uses the probability score between 0 and 1 in its calculation [40]. In a multi-classification problem, we define the logarithmic loss function F as:

$$F = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{C} y_{ij} \cdot log\left(p_{ij}\right) \tag{3.14}$$

Where
N is the number of instances.
C is the number of different labels.
y_{ij} is the binary variable with the expected labels.
p_{ij} is the classification probability output by the classifier for the i-instance and the j-label.

Equation 3.14 calculates the negative logarithm of the predicted probabilities for the true label. Log loss score ranges from 0 to infinity, when the predicted probability aligns well with the true value , then log loss will be close to 0, indicating a more accurate prediction. On the other hand, if the predicted probabilities deviate significantly from the true values, the log loss will be larger, indicating

poorer model performance [40].

Since the log loss of a perfect model is 0 and not upper bound, a random model can be simulated using the data to obtain a log loss value such that any better model developed should have a log loss less than the log loss of the random model[55]. Also, the number of data points that deviated from the true values after prediction (misclassified points) were computed.

Confusion matrix

A confusion matrix is a tabular representation that provides a comprehensive summary of the performance of a machine learning model on a specific set of test data. It is mainly used to evaluate the effectiveness of classification models and is designed to predict categorical labels for individual input instances. The confusion matrix displays the counts of true and false predictions obtained with known data [39]. The matrix shows the counts generated by the model when applied to the test data which are briefly defined below.

True Positives (TP) - Both actual and predicted values are Positive. True Negatives (TN) - Both actual and predicted values are Negative. False Positives (FP) - The actual value is negative, but was predicted as positive. False Negatives (FN) - The actual value is positive but was predicted as negative.

These values offer crucial insights into the model's accuracy and error rates, facilitating a deeper understanding of its classification capabilities. While accuracy is the typical measure for classification, it can be misleading if the dataset is skewed [21]. Precision and recall are also extensively employed to gauge the efficiency of text classifiers [35].

Precision: is the measure of all actual positives out of all predicted positive values. Precision aims to minimize the number of false positives and does not concern itself with false negatives. The value of precision ranges between 0 and 1 and is presented by Equation 3.15.

$$Precision = \frac{TP}{TP + FP} \tag{3.15}$$

Recall: also known as Sensitivity and True Positive Rate is the measure of positive values that are predicted correctly out of all actual positive values. Unlike precision, Recall aims to minimise the number of false negatives and does not concern itself with false positives, and also ranges between 0 and 1. Recall is expressed mathematically in Equation (3.16).

$$Recall = \frac{TP}{TP + FN} \tag{3.16}$$

Precision and recall matrices are extensions of precision and recall, computed in a matrix form to get a deeper understanding of how well the model developed performs [63].

4. Data Description and Exploratory

The data set was obtained from the Memorial Sloan Kettering Cancer Centre (MSKCC), a renowned institution in the field of oncology. This data set was made available through Kaggle and its compilation involved contributions from distinguished researchers and oncologists [33]. The total number of

observations was 3321 which is represented by the first column (ID). The dataset has three features: ”Gene”, ”Variation”, and ”TEXT” (Clinical Literature). With nine distinct labels (Class), making it a multiclass classification task.
Figures 5 and 6 shows the first and last 10 observations of the data set respectively.

ID	Gene	Variation	Class	TEXT
0	FAM58A	Truncating Mutations	1	cyclin dependent kinases cdks regulate variety...
1	CBL	W802*	2	abstract background non small cell lung cancer...
2	CBL	Q249E	2	abstract background non small cell lung cancer...
3	CBL	N454D	3	recent evidence demonstrated acquired uniparen...
4	CBL	L399V	4	oncogenic mutations monomeric casitas b lineag...
5	CBL	V391I	4	oncogenic mutations monomeric casitas b lineag...
6	CBL	V430M	5	oncogenic mutations monomeric casitas b lineag...
7	CBL	Deletion	1	cbl negative regulator activated receptor tyro...
8	CBL	Y371H	4	abstract juvenile myelomonocytic leukemia jmml...
9	CBL	C384R	4	abstract juvenile myelomonocytic leukemia jmml...

Figure 5. First Ten Observation of Data Set

ID	Gene	Variation	Class	TEXT
3311	RUNX1	RUNX1-EVI1 Fusion	4	aml1 evi 1 chimeric gene generated 3 21 q26 q2...
3312	RUNX1	TEL-RUNX1 Fusion	4	balanced chromosomal translocations frequently...
3313	RUNX1	H78Q	4	bcr abl fusion protein generated 9 22 q34 q11 ...
3314	RUNX1	G42R	6	introduction myelodysplastic syndromes mds het...
3315	RUNX1	RUNX1-RUNX1T1 Fusion	4	runx gene family includes three evolutionarily...
3316	RUNX1	D171N	4	introduction myelodysplastic syndromes mds het...
3317	RUNX1	A122*	1	introduction myelodysplastic syndromes mds het...
3318	RUNX1	Fusions	1	runt related transcription factor 1 gene runx1...
3319	RUNX1	R80C	4	runx1 aml1 gene frequent target chromosomal tr...
3320	RUNX1	K83E	4	frequent mutations associated leukemia recurre...

Figure 6. Last Ten Observation of Data Set

The total number of unique genes and variation were 264 and 2996 respectively with text having 65956 words. The distribution of class labels in ascending order is shown in Table 1.

The imbalanced nature of the class distribution is represented in Figure 7, followed by the frequency distribution of the top 20 genes and variations in Figure 8 and Figure 9 respectively.

In various algorithms, particularly those that involve statistics and probabilities, the presence of noise and unnecessary features can negatively impact the performance of the system [35]. Hence, the text data was cleaned using data-cleaning techniques such as stopword removal, lower casing, and special character removal among others. Five observations for text feature with ID (1109, 1277, 1407, 1639, and 2755) had null values, and also there was no correlation among the features.

Table 1. Class Distribution in Dataset

Class	Number of data points	Percentage
8	19	0.572%
9	37	1.114%
3	89	2.680%
5	242	7.287%
6	275	8.281%
2	452	13.610%
1	568	17.103%
4	686	20.656%
7	953	28.698%

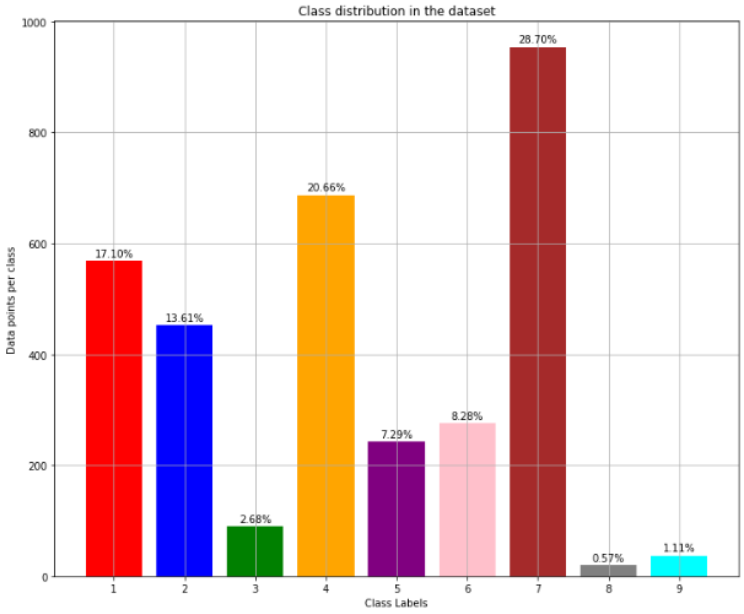


Figure 7. Imbalanced Class Distribution

4.1. Train/Test Split

To train machine learning models that generalises with accurate predictions, the data set was divided into training, cross-validation, and testing. A comparative analysis was done on data splitting to select the most suitable split for our data using the following ratios 48:12:40, 56:14:30, and 60:20:20. The number of observations in each ratio were (1593, 399, 1329), (1859, 456, 997) and (1992, 664, 665) respectively. Data splitting was done using stratified random sampling to maitain the distribution of class label in the data set to address the imbalanced distribution. Hence each split can be a true representation of the entire data set. Figure 10, presents the distribution of class labels. Furthermore, the probability density function for gene and variation were plotted in Figure 11 and Figure 12. This distribution characterises the density or concentration of genes and variation in our data set, and a

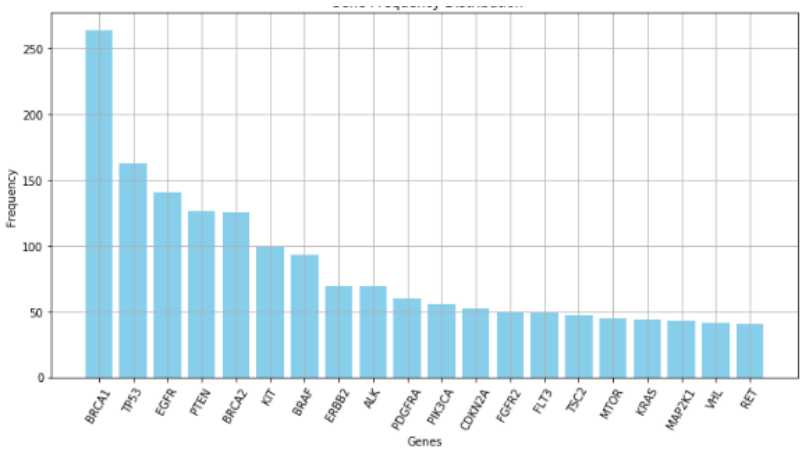


Figure 8. Variation Frequency Distribution for Top 20 Genes

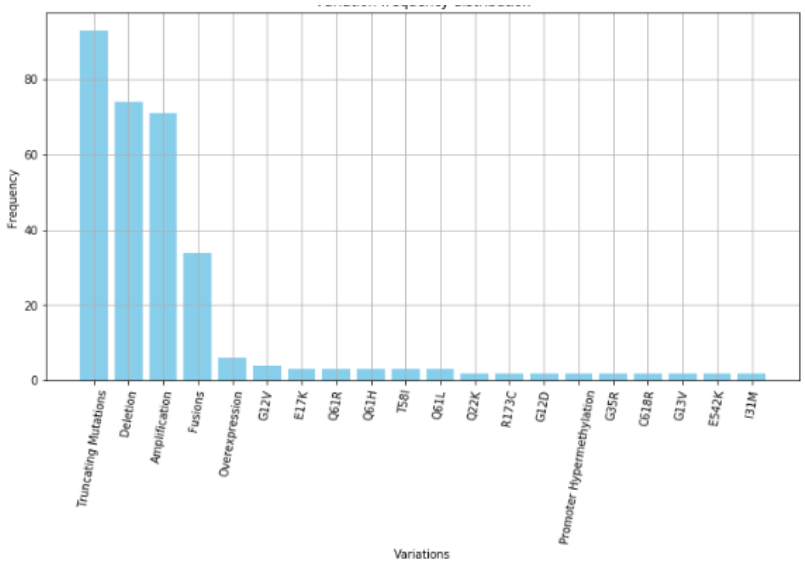


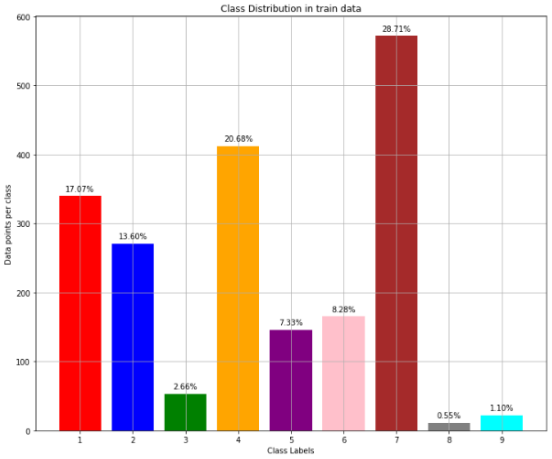
Figure 9. Variation Frequency Distribution for Top 20 Variations

similar distribution is replicated for each split.

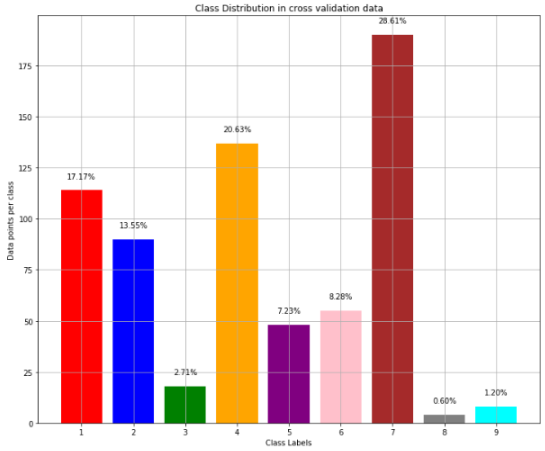
4.2. Random Model Simulation

Since log loss is a performance metric that would be used to evaluate the classifiers and is not upper bound, a random model is simulated to obtain log losses from the test and cross-validation so that any better model developed must have smaller log losses. The random model log losses for cross-validation and testing were 2.44 and 2.55 respectively. The output is displayed in Figure 13

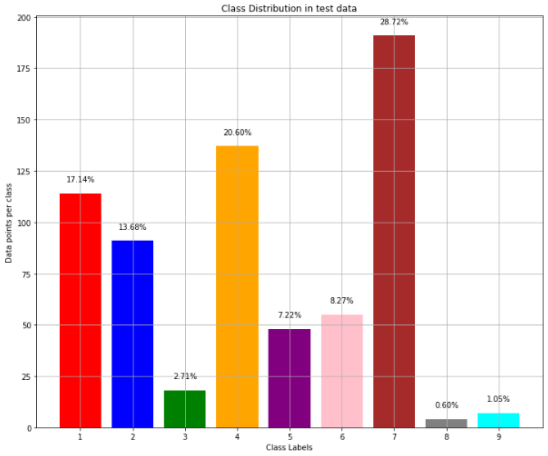
Therefore, any good model developed is expected to perform better than the random model with cross-validation and log losses less than 2.5.



Train data



Cross-validationn data



Test data

Figure 10. Distribution of Class Labels in Each Data Split

4.3. Univariate Feature Analysis

The study explored each feature individually to know how they are distributed and also fitted a logistic regression model using each feature (gene, variation, and text) separately. Their results gave log losses less than 2.5 which means all the features are significant in predicting class labels. A summary of the result is shown in Table 2.

4.4. Feature Extraction

Gene and variation features are categorical variables and were transformed using one-hot encoding and response coding, the text feature was also transformed using TFIDF and Count vectorizer. The dimensions of the transformed features are shown in Tables 3,4 and 5 respectively. Each one-hot encoded feature and response coded feature were stacked horizontally with TFIDF and count vectorizer which is presented in Table 6

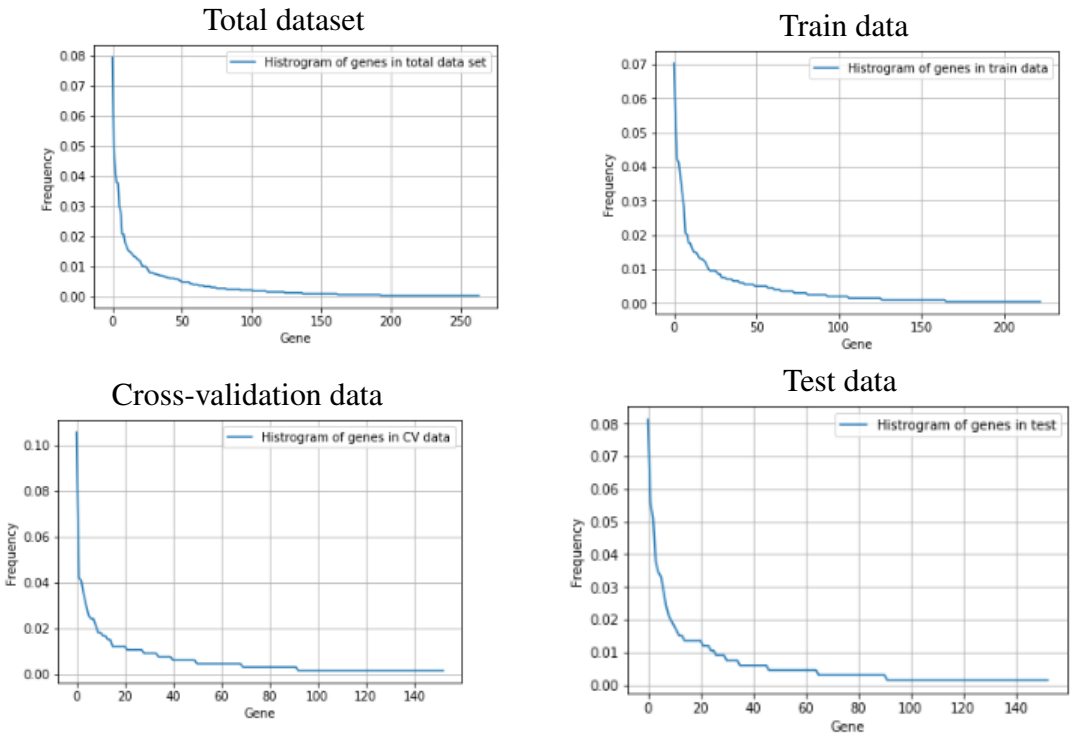


Figure 11. Distribution of Gene in Each Data Split

Table 2. Log Losses for Univariate Analysis

Feature Name	Cross Validation Log Loss	Test Log Loss
Random model	2.4469	2.5532
Gene	1.2153	1.2481
Varaition	1.6904	1.7418
Text	1.0446	1.1075

5. Model Training and Evaluation

The machine learning algorithms used in developing our multiclass classifiers were Multinomial Naïve Bayes(MNB), Support Vector Machine(SVM), Logistic Regression(LR) and Random Forest(RF). These supervised learning models were selected because the data set is labelled with discrete outcome variable (classification problem). Since we are dealing with a classification problem these models are termed as classifiers. The selected classifiers offer the advantage of providing feature importance, allowing for a better understanding of the features that contribute significantly to the predictions.

Each of these classifiers was trained using transformed training data, designed to learn patterns and relationships in the data. Also, the transformed cross-validation data was used for hyperparameter tuning to regulate the learning process in which the best parameter values were selected for optimal model training, while the transformed test data was used to get a final evaluation of the model's performance after all tuning was done. Additionally, all classifiers were calibrated to obtain probability output, rather than just class labels to minimize errors. The class probabilities allow pathologists to

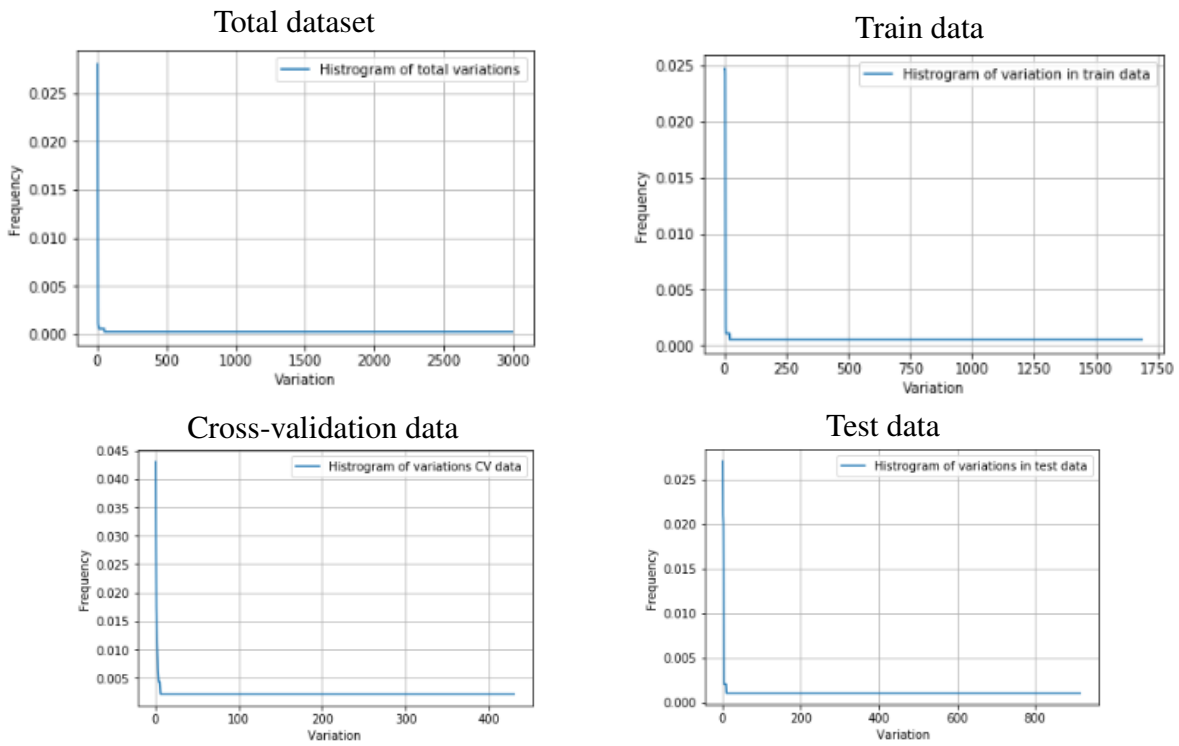


Figure 12. Distribution of Variation in Each Data Split

Table 3. Dimension of Response Coded Features

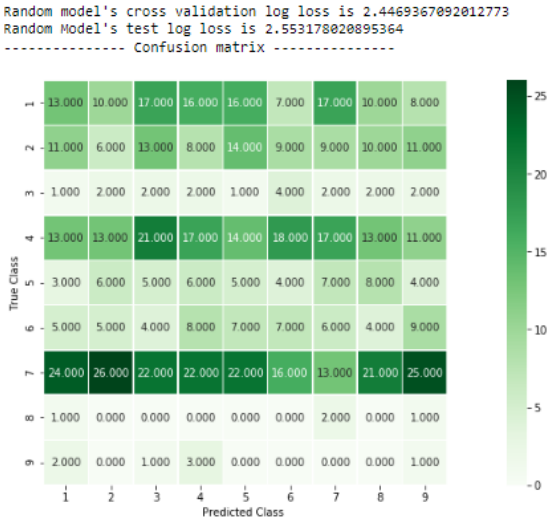
Response Coding of Categorical Features		
Split	Gene	Variation
Train	1895, 9	1895, 9
Cross validation	465, 9	465, 9
Test	997, 9	997, 9

make informed decisions about whether further testing is necessary. Hence, log loss was chosen as an evaluation metric. Further, to gain insight into the classification process, recall, and precision matrices were obtained from the confusion matrix to get a deeper understanding of the developed classifiers. Logistic regression and support vector machine were both tuned with balanced and unbalanced class distributions from the sklearn library. In summary, 16 classifiers from four machine learning algorithms were trained using four feature transformation techniques. The model development output from ”Random Forest + Response + TFIDF’ and ”Logistic Regression + Response + TFIDF(WCB)” are displayed below since our optimum decision was based on these two classifiers

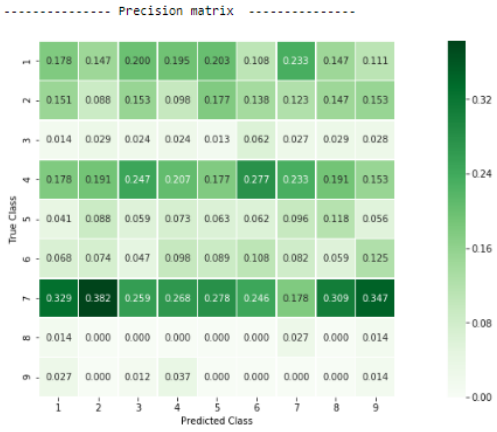
5.1. Random Forest + Response Coding + TFIDF Vectorizer

The third column in Table 6, was used to train random forest with 100, 200, 500, 1000, and 2000 estimators with maximum depth of 5 and 10. The minimum log loss was obtained at the n estimate = 2000 with a maximum depth of 10 and the tuning output is shown in Figure 14.

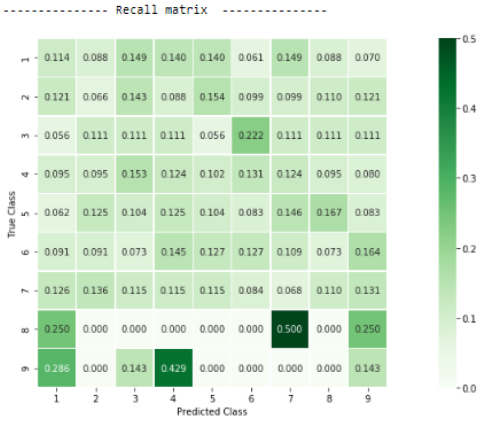
Subsequently, Figure15 shows precision, recall, and confusion matrices thereby giving a deeper un-



Confusion Matrix



Precision Matrix



Recall Matrix

Figure 13. Random Model for log loss

derstanding of the developed model.

5.2. Logistic Regression + Response Coding + TFIDF (WCB)

Also, a logistic regression model was trained using response coded (gene and variation features) and TFIDF encoded (text feature). An alpha value of 10^{-6} to 10^{3} was used in hyperparameter tuning and the minimum log loss occurred at alpha equals 10^{-4} which was used to train the model. The tuning output can be seen in Figure 16 followed by precision recall and confusion matrix in Figure 17.

5.3. Feature Importance

Since the study aims to train classifiers with class probabilities that are easily interpretable, the importance of the features was assessed. A test point is selected for two correctly classified points in Figures 18 and 19. The predicted class probability and the features present in query point add more insight into the model's prediction.

Table 4. Dimension of One-hot Encododed Features

Onehot Encoding of Categorical Features		
Split	Gene	Variation
Train	1895, 224	1895, 1724
Cross validation	465, 224	465, 1724
Test	997, 224	997, 1724

Table 5. Dimension of Count and TFIDF Vectorizer

Encoding of TEXT Feature		
Split	TFIDF vectorizer	Count Vectorizer
Train	1895, 49767	1895, 65956
Cross validation	465, 49767	465, 65956
Test	997, 49767	997, 65956

6. Results and Discussion

Among the three data split ratios used, the 56: 14: 30 split gave the minimum log loss and the output is shown in Table 7 followed by the graphical representation of the results displayed in Figure 20

Table 7. Summary of Log Losses and Percentages Misclassified

Classifier	**Train Loss**	**CV Loss**	**Test Loss**	**Misclassified %**
MNB + Onehot + TFIDF	0.6494	1.1202	1.1976	33.12%
MNB + Response + TFIDF	0.8574	1.1465	1.2363	35.05%
MNB + Onehot + CountVec	0.5806	1.1132	1.2027	32.26%
SVM+ Onehot + TFIDF(CB)	0.5366	1.0479	1.1247	33.12%
SVM + Onehot + TFIDF (WCB)	0.4874	1.0115	1.0799	32.90%
SVM + Response + TFIDF (CB)	0.6697	1.0252	1.1014	31.61%
SVM + Response + TFIDF (WCB)	0.6457	1.0134	1.0747	30.75%
SVM + Onehot + CountVec	0.4971	1.0136	1.0861	33.76%
LR + Onehot + TFIDF (CB)	0.4157	0.9575	1.0238	32.47%
LR + Onehot + TFIDF (WCB)	0.4046	0.9499	1.0183	32.04%
LR + Response + TFIDF (CB)	0.5897	0.9561	1.0196	30.96%
LR + Response + TFIDF (WCB)	0.5935	0.9536	1.0125	30.97%
LR + Onehot + CountVec	0.4193	0.9648	1.0364	31.18%
RF + Onehot + TFIDF	0.6219	1.0416	1.0813	33.54%
RF + Response + TFIDF	0.3859	0.9517	1.0023	29.68%
RF + Onehot + CountVec	0.6128	1.0378	1.0794	32.90%

Green = Classifier with the least log losses and misclassified percent.
Blue = Best classifier
Red = Worst classifier

Table 6. Dimension of Stacked Features

Split	Onehot + TFIDF	Response + TFIDF	Onehot + Count Vectorizer
Train	1895, 51715	1895, 49785	1895, 67904
Cross validation	465, 51715	465, 49785	465, 67904
Test	997, 51715	997, 49785	997, 67904

```
When number of estimators = 100 and maximum depth =  5 , og Loss is 1.0616972233832889
When number of estimators = 100 and maximum depth =  10 , og Loss is 0.9616820489257133
When number of estimators = 200 and maximum depth =  5 , og Loss is 1.0475092532293255
When number of estimators = 200 and maximum depth =  10 , og Loss is 0.9572501047399846
When number of estimators = 500 and maximum depth =  5 , og Loss is 1.0450549721455784
When number of estimators = 500 and maximum depth =  10 , og Loss is 0.9563386328085142
When number of estimators = 1000 and maximum depth =  5 , og Loss is 1.0419605889866121
When number of estimators = 1000 and maximum depth =  10 , og Loss is 0.9558053269434805
When number of estimators = 2000 and maximum depth =  5 , og Loss is 1.039119678287059
When number of estimators = 2000 and maximum depth =  10 , og Loss is 0.9516859615434174
The train log loss for best value of alpha at 2000 is 0.38586488825179943
The cross validation log loss for best value of alpha at  2000 is 0.9516859615434174
The test log loss for best value of alpha at 2000 is 1.0023270963708224
```

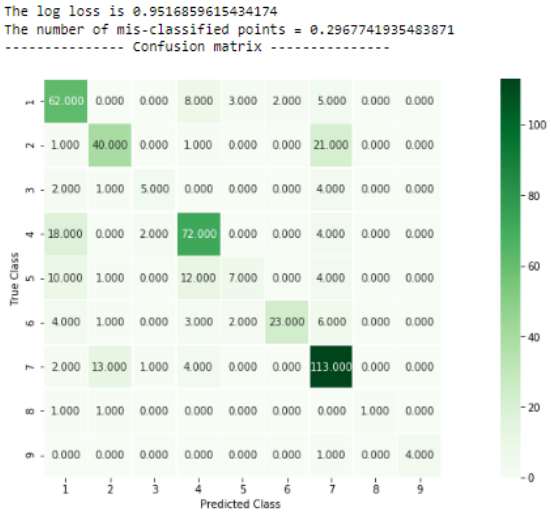
Figure 14. Hyperparameter Tuning for Random Forest + Response + TFIDF

The Train Loss represents the error of the model on the training data and indicates how well the model fits the training data. However, excessively low Train Loss may indicate overfitting, where the model memorizes the training data but fails to generalize well to unseen data. Although train loss is essential for understanding how well the model fits the training data, it alone is not sufficient to assess model performance

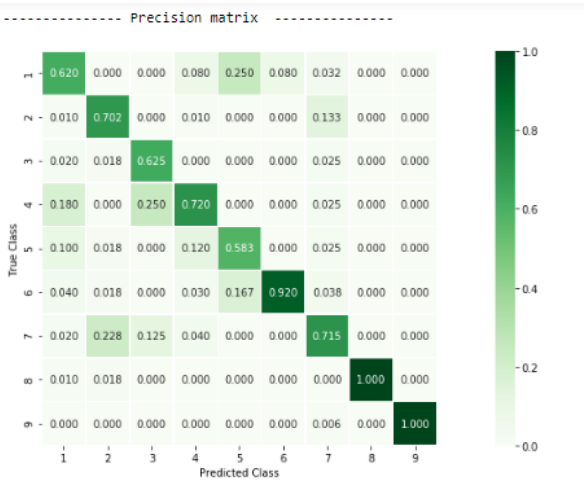
CV loss measures the model's performance on unseen data during cross-validation, which simulates how the model would perform on new data. It provides an estimate of how well the model generalizes to new, unseen data. A lower CV loss indicates better generalisation performance, as the model is better able to predict outcomes on unseen data.

Test loss evaluates the model's performance on an independent test data set that the model has not seen during training or cross-validation. Test loss provides a realistic estimate of how well the model will perform in real-world applications. Similarly to CV loss, a lower test loss indicates better generalisation and predictive performance.

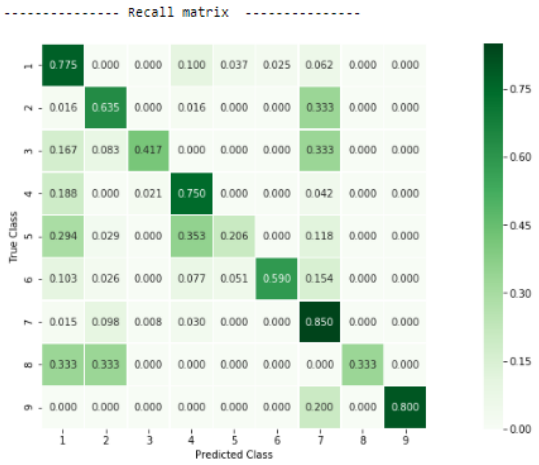
Misclassified points play a crucial role in evaluating the performance of machine learning models and selecting the best model for a given task. In domains where accurate predictions are crucial, such as healthcare, minimising misclassified points is paramount. Decision-makers rely on models with low misclassification rates to make informed decisions based on the model's output. Logistic Regression and Support Vector Machines were developed with CB (class balanced) and WCB (without class balanced), from Sk-learn.



Confusion Matrix



Precision Matrix



Recall Matrix

Figure 15. Matrices For Random Forest + Response Coding + TFIDF

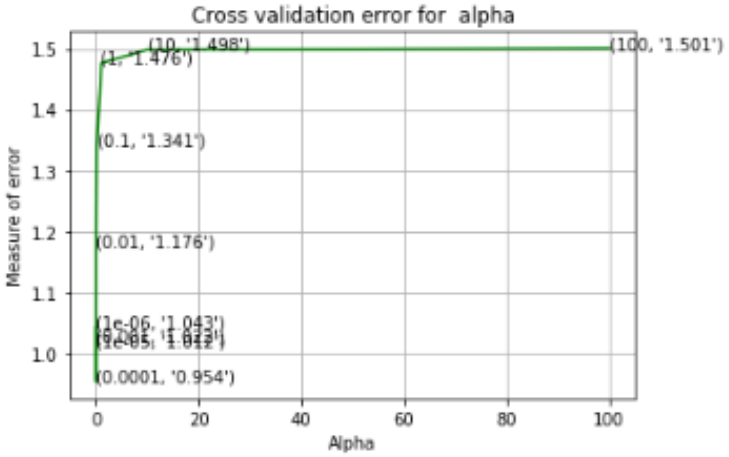
6.1. Multinomial Naive Bayes (MNB) Classifiers

MNB + Onehot + TFIDF: This model displayed moderate performance with a relatively high Train Loss of 0.6494. The CV and Test Loss values of 1.1202 and 1.1976 respectively were moderatly higher, indicating potential overfitting or model complexity. The misclassified percentage 33.12% was also moderate.

MNB + Response + TFIDF: Response coding with TFIDF encoding showed higher Train, CV, and test loss values than Onehot encoding, suggesting less optimal performance. The misclassified percent is relatively higher at 35.05%, indicating limitations in predictive accuracy.

MNB + Onehot + CountVec: This model exhibited moderate Train, CV and Test Loss values of 0.5806, 1.1132, and 1.2027, respectively. However, the misclassified percentage was relatively high (32.26%), indicating potential limitations in predictive accuracy compared to other variations of the

```
when alpha = 1e-06 , Log Loss is 1.0426976301592992
when alpha = 1e-05 , Log Loss is 1.0121837823020183
when alpha = 0.0001 , Log Loss is 0.9536081141927446
when alpha = 0.001 , Log Loss is 1.0217439700272573
when alpha = 0.01 , Log Loss is 1.1755706462001445
when alpha = 0.1 , Log Loss is 1.341049397957602
when alpha = 1 , Log Loss is 1.476426167578919
when alpha = 10 , Log Loss is 1.4983238062398072
when alpha = 100 , Log Loss is 1.5008070053037637
```




```
The train log loss for best value of alpha at   0.0001 is 0.593487858961164
The cross validation log loss for best value of alpha at  0.0001 is 0.9536081141927446
The test log loss for best value of alpha at  0.0001 is 1.012518135093537
```

Figure 16. Hyperparameter Tuning for Logistic Regression + Response + TFIDF

MNB.

Among all Multinomial Naive Bayes classifiers, MNB + Onehot + CountVec showed relatively better performance, followed by MNB + Onehot + TFIDF with MNB + Response + TFIDF exhibiting the worst performance.

6.2. Support Vector Machine (SVM) Classifiers

SVM + Onehot + TFIDF (CB): This model demonstrated moderate performance with a relatively high Train Loss of 0.5366. The CV and Test Loss values of 1.0479 and 1.1247 respectively, further increased. The Misclassified percentage was notably high at 33.12%.

SVM + Onehot + TFIDF (WCB): Similar to the CB variation, this model showed moderate performance across all metrics, with slightly improved Train, CV, and Test Loss values. However, the Misclassified percentage remained relatively low at 32. 90%.

SVM + Response + TFIDF (CB): This model exhibited competitive performance with a relatively high Train Loss of 0.6697. However, the CV and Test Loss of 1.0252 and 1.1014, respectively, were lower compared to the Onehot + TFIDF variations. The misclassified percentage was significantly

Confusion Matrix

Precision Matrix

Recall Matrix

Figure 17. Matrices For Logistic Regression+Response Coding+TFIDF (WCB)

low at 31. 61%, indicating improved predictive accuracy.

SVM + Response + TFIDF (WCB): Similar to the CB variation, this model demonstrated balanced performance across Train, CV, and Test Loss values. The Misclassified percent was lower at 30.75%, indicating improved predictive accuracy compared to other SVM variations.

SVM + Onehot + CountVec: This model exhibits moderate Train, CV, and Test Loss values. However, the Misclassified % was relatively high at 33.76%, indicating potential limitations in predictive accuracy compared to other SVM variations.

In summary, SVM + Response + TFIDF (WCB) emerges as the most promising classifier among SVM variations, demonstrating balanced performance and competitive predictive accuracy. SVM + Onehot + TFIDF (CB) performed worse among all SVM variation

```
Predicted Class : 1
Predicted Class Probabilities: [[0.7102 0.0071 0.0044 0.0024 0.0065 0.0105 0.2537 0.0025 0.0027]]
Actual Class : 1
--------------------------------------------------
263 Text feature [colorectal] present in test data point [True]
318 Text feature [binding] present in test data point [True]
332 Text feature [dna] present in test data point [True]
392 Text feature [residues] present in test data point [True]
408 Text feature [surface] present in test data point [True]
411 Text feature [function] present in test data point [True]
415 Text feature [region] present in test data point [True]
419 Text feature [supplementary] present in test data point [True]
445 Text feature [p21] present in test data point [True]
471 Text feature [rt] present in test data point [True]
481 Text feature [processing] present in test data point [True]
Out of the top  500  features  11 are present in query point
```

Figure 18. Correctly Classified Point

```
Predicted Class : 1
Predicted Class Probabilities: [[0.5107 0.0182 0.0036 0.0023 0.0093 0.0114 0.4407 0.0019 0.0019]]
Actual Class : 1
--------------------------------------------------
197 Text feature [binding] present in test data point [True]
274 Text feature [colorectal] present in test data point [True]
278 Text feature [supplementary] present in test data point [True]
322 Text feature [dna] present in test data point [True]
366 Text feature [residues] present in test data point [True]
380 Text feature [prostate] present in test data point [True]
421 Text feature [surface] present in test data point [True]
461 Text feature [fig] present in test data point [True]
462 Text feature [rt] present in test data point [True]
469 Text feature [region] present in test data point [True]
480 Text feature [function] present in test data point [True]
482 Text feature [terminal] present in test data point [True]
489 Text feature [processing] present in test data point [True]
498 Text feature [insertion] present in test data point [True]
Out of the top  500  features  14 are present in query point
```

Figure 19. Correctly Classified Point

6.3. Logistic Regression (LR) Classifiers

LR + Onehot + TFIDF (CB): This model exhibits relatively balanced performance with a moderate Train Loss of 0.4157. However, the CV and Test Loss values increase to 0.9575 and 1.0238, respectively. The Misclassified ercentage of 32.47% indicates poor model predictive accuracy.

LR + Onehot + TFIDF (WCB): Similar to the CB variation, this model demonstrated balanced performance across Train, CV, and Test Loss values. While the Train Loss is marginally lower at 0.4046, the Misclassified percent remains relatively high at 32.04%.

LR + Response + TFIDF (CB): This model showed competitive performance with a Train Loss of 0.5897 and CV and Test Loss values around 0.9561 and 1.0196, respectively. The misclassified percent is significantly lower at 30. 96%, indicating improved predictive accuracy compared to other variations of the LR.

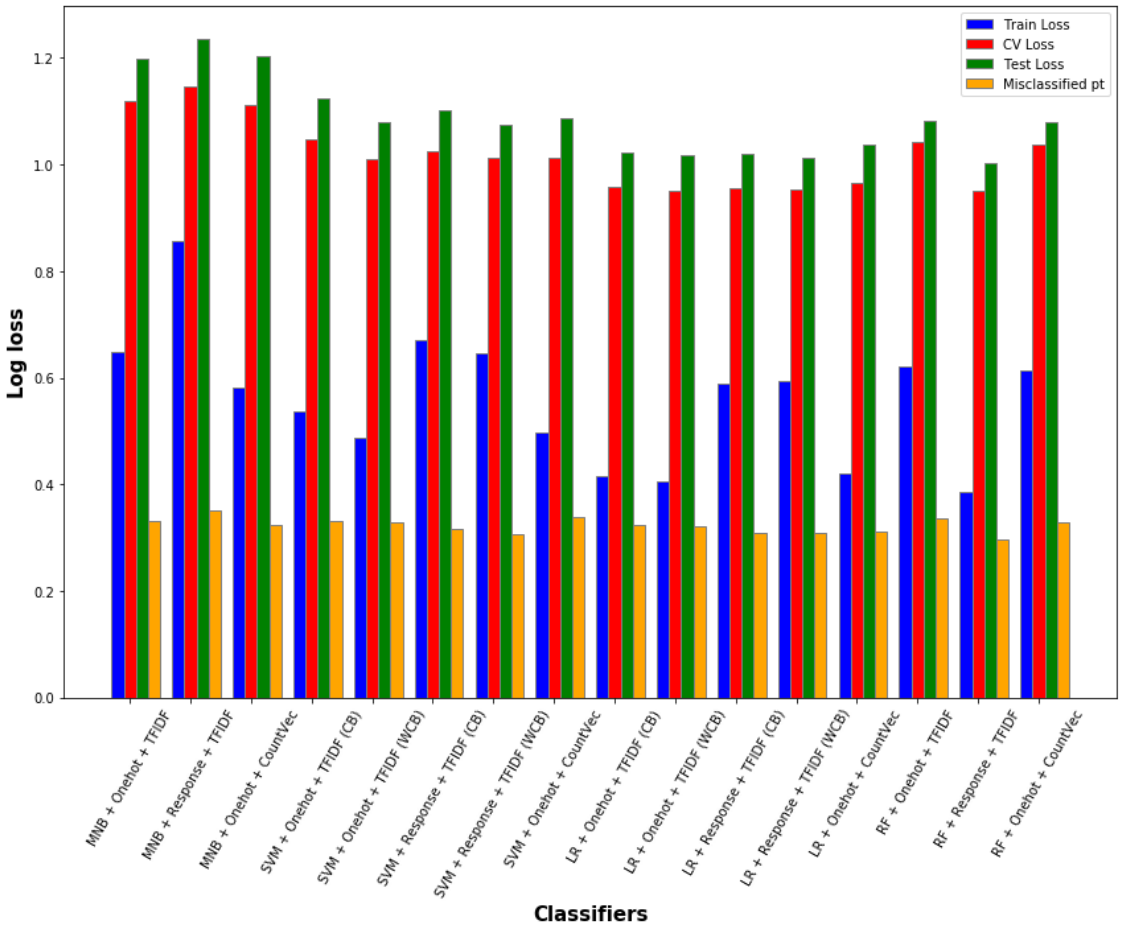


Figure 20. Grouped Bar Graph of Classifiers

LR + Response + TFIDF (WCB): Similar to the CB variation, this model displayed balanced performance across all metrics, with slightly improved Test Loss and Misclassified % values of 1.0125 and 30.97%, respectively.

LR + Onehot + CountVec: This model exhibited moderate Train, CV, and Test Loss values. However, the Misclassified % is slightly higher at 31.18%, indicating relatively lower predictive accuracy compared to LR + Response + TFIDF variations.

In summary, LR + Response + TFIDF (CB) and LR + Response + TFIDF (WCB) emerged as the most promising classifiers among the variations of LR, demonstrating balanced performance and competitive predictive accuracy. The worst performing model among all LRs was LR + Onehot + TFIDF (CB).

6.4. Random Forest (RF) Classifiers

RF + Onehot + TFIDF demonstrated competitive training performance with a training log loss of 0.6219. However, in the testing phase, it yields a test log loss of 1.0813 and a percentage misclassified of 33.54% Similar results for RF + Onehot + CountVec were obtained. RF + Response + TFIDF

displays the lowest training log loss of 0.3859, the lowest test log loss of 0.9517, with a misclassified percentage of 29. 68%.

RF + Response + TFIDF stands out as the most promising classifier among all RF and the classifier that performed worse was RF + Onehot + TFIDF.

It was noted that, within each category of the four primary classifiers, the most effective subtype of classifiers were trained using response-coded gene and variation, along with TFIDF-encoded text except MNB where the best classifier was trained with one-hot encoding and count vectorizer.
Subsequently, both logistic regression and support vector machine classifiers that were developed with a balanced class distribution performed slightly worse than those developed with an imbalanced class distribution. This suggest that changing the imbalance distribution of the data may affect model performance.

In conclusion, RF + Response + TFIDF stands out with a relatively low test log loss of 1.0023 and a percentage misclassified of 29.64%. However, "LR + Response + TFIDF (WCB)" emerged as the optimal classifier with a test loss of 1.0125 and a misclassification rate of 30.97%. This model achieved a commendable balance between training performance and generalization, as evidenced by its high training log loss of 0.6457 compared to RF + Response + TFIDF training log loss of 0.3859 which may lead to overfitting. Additionally, probability outputs were obtained which can be used by pathologists to justify their prediction and make accurate decisions.

6.5. Significance of Predicted Class Probabilities

Analyzing the probabilities for each test point is crucial for understanding the reliability of the model predictions and their implications for clinical decision-making. The predicted probability for some selected test points is shown in Table 8 . A wider gap between the highest and second-highest probabilities generally indicates more confident predictions, while narrow gaps suggest uncertainty that may require further investigation. Blue = Highest Probability
Green = Second highest probability

All test points from Table 8 were correctly predicted by our classifier. Notably, majority of these test points exhibited substantial gaps between the highest and second-highest probabilities, such as test point 3 which has the highest probability of 0.8826 and the second-highest probability of 0.0367 indicating a high level of certainty regarding the predicted class. However, the probabilities of 0.3889 and 0.1766 for test point 3, 0.4433 and 0.2119 for test point 6, and 0.4774 and 0.358 for test point 12 indicate closer class probabilities. This suggests a potential need for further assessment or additional tests to ensure the precision of the predictions and also reduce error.

6.6. Precision and Recall Matrices

Although log loss and misclassified points provided the basis for evaluating our classifiers, precision and recall matrices were adopted since log loss is a single metric and does not provide detailed information about the classifier's behavior. However, precision and recall matrices give precision and

Table 8. Class Probability For Correctly Classified Point

Test Pt	**Probabilities of Class Labels**								
	1	2	3	4	5	6	7	8	9
1	0.0722	0.0503	0.0102	0.082	0.0285	0.0319	0.7192	0.0033	0.0025
2	0.0177	0.082	0.0109	0.062	0.0392	0.0318	0.7495	0.0042	0.0026
3	0.1149	0.1708	0.0209	0.1766	0.0734	0.0471	0.3889	0.0045	0.0029
4	0.0804	0.0981	0.0154	0.0399	0.0522	0.0404	0.6613	0.0052	0.0068
5	0.8661	0.0536	0.0068	0.0071	0.023	0.018	0.0153	0.0048	0.0013
6	0.1584	0.2119	0.0097	0.0865	0.0637	0.0156	0.4433	0.0052	0.0056
7	0.0659	0.0731	0.0134	0.6596	0.0439	0.0365	0.098	0.0048	0.0049
8	0.0623	0.0908	0.0159	0.6665	0.0493	0.0284	0.0757	0.0059	0.0052
9	0.0444	0.2041	0.012	0.0725	0.0825	0.0347	0.5424	0.0053	0.0021
10	0.7763	0.0654	0.0109	0.0233	0.0408	0.031	0.0378	0.0121	0.0021
11	0.8826	0.0367	0.0062	0.0197	0.0234	0.0148	0.0126	0.0032	0.0099
12	0.0457	0.03713	0.013	0.0265	0.358	0.0242	0.4774	0.0039	0.0022
13	0.06	0.7424	0.0124	0.0735	0.0378	0.0282	0.0383	0.00454	0.006
14	0.1035	0.6435	0.0143	0.0963	0.0421	0.0306	0.0613	0.0042	0.0043

recall for each class label which can be useful in making decisions, especially in our case where the distribution of class labels are not balanced. This can be observed by the leading diagonal of each matrix representing precision and recall for each class label. A well-performing classifier is expected to have all values in the leading diagonal closer to one or display a darker green color, as observed in the precision and recall matrices in Figure 21 and 22.

Figure 22. Precision and Recall Matrices for Best classifier

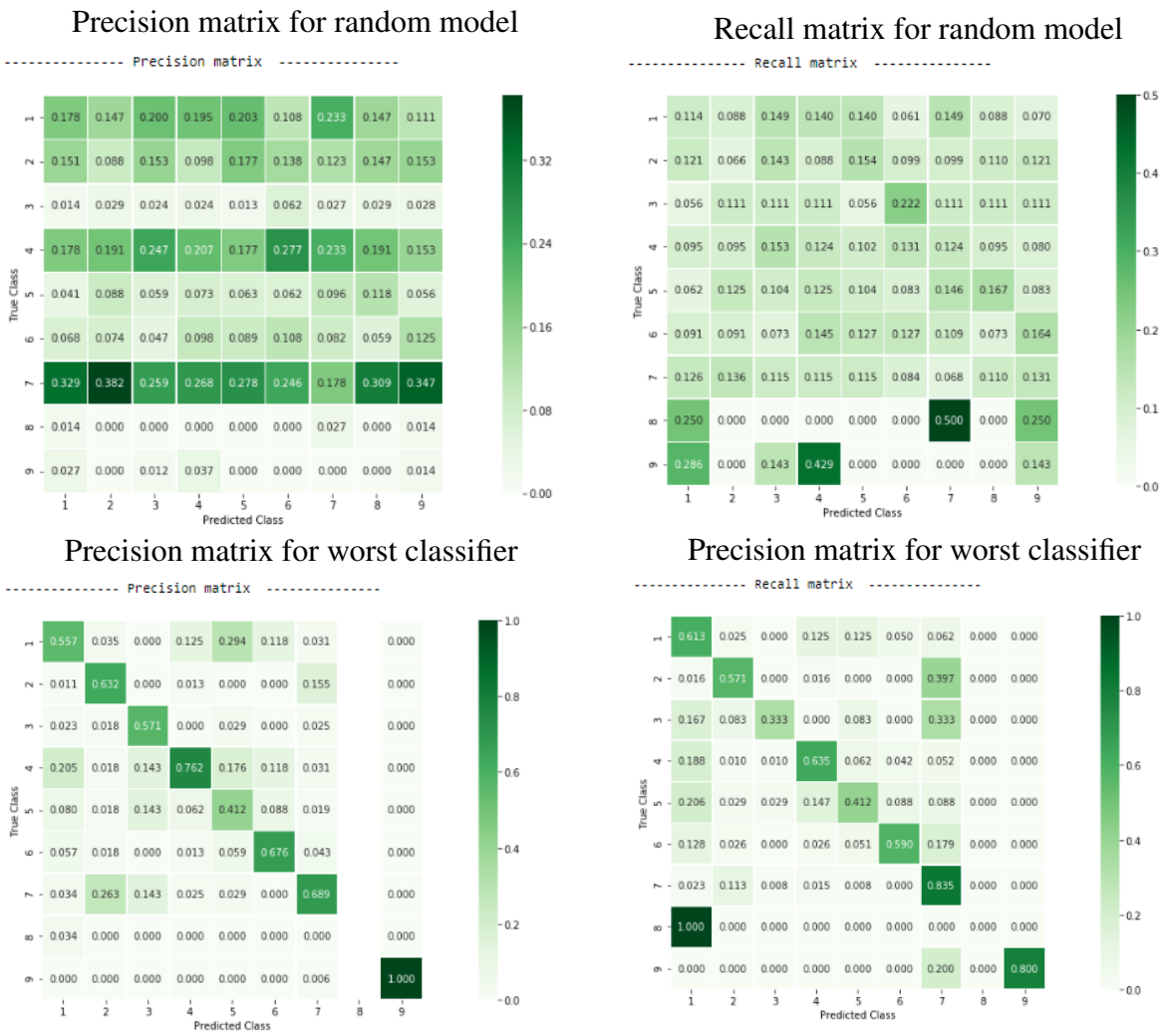


Figure 21. Precision and Recall Matrices for Random Model and Worst performing Classifier

7. Conclusion

In this research, we extensively explored machine learning models to improve efficiency and accuracy in genetic mutation classification by computing feature importance, precision and recall matrices, and probability values for the predicted class label, to enhance the model's predictive performance. LR + Response + TFIDF (WCB) emerged as the best classifier with a log-loss of 1.0125 and a misclassification rate of 30.97%. A noteworthy aspect of this model is the ability to provide molecular pathologists with reasons behind the model's predictions thereby minimizing misdiagnosis rates and improving patient outcomes. Our developed classifier was cross-validated to ensure robustness. However, it's essential to acknowledge the limitations posed by dataset availability.

Future research endeavors should prioritize expanding datasets and exploring advanced ML techniques to further improve classification accuracy and interpretability. Our developed classifier could be validated on external datasets to assess how well it can generalise. Collaboration between ML experts

and molecular pathologists can improve accuracy in genetic mutation classification and personalized medicine.

Conflict of interest

The authors state that they have no financial or other conflicts of interest to disclose with connection to this research.

References

1. Algamal, Z. (2017). An efficient gene selection method for high-dimensional microarray data based on sparse logistic regression. Electronic Journal of Applied Statistical Analysis, 10(1), 242–256.
2. Aburass, S., Dorgham, O., & Shaqsi, J. A. (2023). A hybrid machine learning model for classifying gene mutations in cancer using LSTM, BiLSTM, CNN, GRU, and GloVe. arXiv preprint arXiv:2307.14361.
3. Alshenawy, F. Y., & Almetwally, E. M. (2023). A COMPARATIVE STUDY OF STATISTICAL AND INTELLIGENT CLASSIFICATION MODELS FOR PREDICTING DIABETES. Advances and Applications in Statistics, 88(2), 201-223.
4. Splane, B. (2022). Differences between a malignant and benign tumor. verywellhealth.com. Retrieved from https://www.verywellhealth.com/what-does-malignant-and-benign-mean-514240
5. Barash, Y., Guralnik, G., Tau, N., Soffer, S., Levy, T., Shimon, O., Zimlichman, E., Konen, E., & Klang, E. (2020). Comparison of deep learning models for natural language processing-based classification of non-English head CT reports. Neuroradiology, 62, 1247–1256.
6. Bhola, A., & Tiwari, A. K. (2015). Machine learning based approaches for cancer classification using gene expression data. Machine Learning and Applications: An International Journal, 2(3/4), 01–12.
7. Atlas Biomed. (2019). What is a genetic mutation and how do gene variants affect health? Retrieved from https://atlasbiomed.com/blog/what-is-a-genetic-mutation.
8. Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4). Springer.
9. Bo, X. H. (2006). Svm multi-class classification. Technical report, DataAcquis.
10. Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory (pp. 144–152).
11. Breiman, L. (2001). Random forests. Machine learning, 45, 5–32.
12. Brownlee, J. (2017). Why one-hot encode data in machine learning. Machine Learning Mastery, 1–46.
13. Bruno, N., Jun, T., & Tessier, H. (2019). Natural language processing and classification methods for the maintenance and optimization of US weapon systems. In 2019 Systems and Information Engineering Design Symposium (SIEDS) (pp. 1–6). IEEE.

14. Chang, P., Grinband, J., Weinberg, B. D., Bardis, M., Khy, M., Cadena, G., ... Bota. (2018). Deep-learning convolutional neural networks accurately classify genetic mutations in gliomas. American Journal of Neuroradiology, 39(7), 1201–1207.

15. Chen, K., Zhang, Z., Long, J., & Zhang, H. (2016). Turning from tf-idf to tf-igm for term weighting in text classification. Expert Systems with Applications, 66, 245–260.

16. Cho, S. B., & Won, H.-H. (2007). Cancer classification using ensemble of neural networks with multiple significant gene subsets. Applied intelligence, 26, 243–250.

17. Cronin, A., Intepe, G., Shearman, D., & Sneyd, A. (2019). Analysis using natural language processing of feedback data from two mathematics support centres. International Journal of Mathematical Education in Science and Technology, 50(7), 1087–1103.

18. Cruz, J. A., & Wishart, D. S. (2006). Applications of machine learning in cancer prediction and prognosis. Cancer informatics, 2, 117693510600200030.

19. Doan, S., Maehara, C. K., Chaparro, J. D., Lu, S., Liu, R., Graham, A., ... Lloyd, D. D. (2016). Building a natural language processing tool to identify patients with high clinical suspicion for Kawasaki disease from emergency department notes. Academic Emergency Medicine, 23(5), 628–636.

20. Dwivedi, S. K., & Arya, C. (2016). Automatic text classification in information retrieval: A survey. In Proceedings of the second international conference on information and communication technology for competitive strategies (pp. 1–6).

21. Espejo-Garcia, B., Martinez-Guanter, J., Pérez-Ruiz, M., Lopez-Pellicer, F. J., & Zarazaga-Soria, F. J. (2018). Machine learning for automatic rule classification of agricultural regulations: A case study in Spain. Computers and Electronics in Agriculture, 150, 343–352.

22. Frank, E., & Bouckaert, R. R. (2006). Naive Bayes for text classification with unbalanced classes. In Knowledge Discovery in Databases: PKDD 2006: 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings 10 (pp. 503–510). Springer.

23. Goyal, P., Pandey, S., & Jain, K. (2018). Deep learning for natural language processing. New York: Apress.

24. Green, D. R. (2017). Cancer and apoptosis: Who is built to last? Cancer Cell, 31(1), 2–4.

25. Gupta, M., Wu, H., Arora, S., Gupta, A., Chaudhary, G., & Hua, Q. (2021). Gene mutation classification through text evidence facilitating cancer tumor detection. Journal of Healthcare Engineering, 2021, 1–16.

26. Kaggle. (2017). Personalized Medicine: Redefining Cancer Treatment. Retrieved from https://www.kaggle.com/c/msk-redefining-cancer-treatment

27. Hart, P. E., Stork, D. G., & Duda, R. O. (2000). Pattern classification. Wiley Hoboken.

28. Hill, B. M. (1968). Posterior distribution of percentiles: Bayes' theorem for sampling from a population. Journal of the American Statistical Association, 63(322), 677–691.

29. Ho, T. K. (1995). Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition (Vol. 1, pp. 278–282). IEEE.

30. Hoogeveen, D., Wang, L., Baldwin, T., Verspoor, K. M., et al. (2018). Web forum retrieval and text analytics: A survey. Foundations and Trends® in Information Retrieval, 12(1), 1–163.

31. National Cancer Institute. (2016). What Is Cancer? Retrieved from https://www.cancer.gov/about-cancer/understanding/what-is-cancer-definition

32. Jackson, S. E., & Chester, J. D. (2015). Personalised cancer medicine. International Journal of Cancer, 137(2), 262–266.

33. Kaggle. (2017). Personalized medicine: Redefining cancer treatment/data. Retrieved from https://www.kaggle.com/c/msk-

34. Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology Journal, 13, 8–17.

35. Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Information, 10(4), 150.

36. Krishnapuram, B., Carin, L., Figueiredo, M. A. T., & Hartemink, A. J. (2005). Sparse multinomial logistic regression: Fast algorithms and generalization bounds. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(6), 957–968./

37. Kumar, A., & Santhosh, K. S. (202). Personalized Medicine: Redefining Cancer Treatment Using Machine Learning. International Journal of Engineering Applied Sciences and Technology, 5(8), 207-210.

38. Lauría, E. J. M., & March, A. D. (2011). Combining Bayesian text classification and shrinkage to automate healthcare coding: A data quality analysis. Journal of Data and Information Quality (JDIQ), 2(3), 1–22.

39. Lever, J. (2016). Classification evaluation: It is important to understand both what a classification metric expresses and what it hides. Nature Methods, 13(8), 603–605.

40. Li, G., & Yao, B. (2018). Classification of genetic mutations for cancer treatment with machine learning approaches. International Journal of Design, Analysis and Tools for Integrated Circuits and Systems, 7(1), 63–66.

41. Libbrecht, M. W., & Noble, W. S. (2015). Machine learning applications in genetics and genomics. Nature Reviews Genetics, 16(6), 321–332.

42. Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR), 9(1), 381–386.

43. Mitchell, T. M. (1997). Does machine learning really work? AI Magazine, 18(3), 11–11.

44. Mitchell, T. M. (2006). The discipline of machine learning (Vol. 9). Carnegie Mellon University, School of Computer Science, Machine Learning.

45. Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of Machine Learning. MIT Press.

46. Montazeri, M., Montazeri, M., Montazeri, M., & Beigzadeh, A. (2016). Machine learning models in breast cancer survival prediction. Technology and Health Care, 24(1), 31–42.

47. Marie, H. S., Abu El-hassan, K., Almetwally, E. M., & El-Mandouh, M. A. (2022). Joint shear strength prediction of beam-column connections using machine learning via experimental results. Case Studies in Construction Materials, 17, e01463.

48. Nobles, A. L., Glenn, J. J., Kowsari, K., Teachman, B. A., & Barnes, L. E. (2018). Identification of imminent suicide risk among young adults using text messages. In Proceedings of the 2018 CHI conference on human factors in computing systems (pp. 1–11).

49. Ofoghi, B., & Verspoor, K. (2017). Textual emotion classification: An interoperability study on cross-genre data sets. In Australasian Joint Conference on Artificial Intelligence (pp. 262–273). Springer.

50. Omoregbe, N. A. I., Ndaman, I. O., Misra, S., Abayomi-Alli, O. O., Damaševičius, R., & Dogra, A. (2020). Text messaging-based medical diagnosis using natural language processing and fuzzy logic. Journal of Healthcare Engineering, 2020, 1–14.

51. Palanivinayagam, A., El-Bayeh, C. Z., & Damaševičius, R. (2023). Twenty years of machine-learning-based text classification: A systematic review. Algorithms, 16(5), 236.

52. Pearson, E. S. (1925). Bayes' theorem, examined in the light of experimental sampling. Biometrika, 388–442.

53. Saif, H., Fernandez, M., He, Y., & Alani, H. (2014). On stopwords, filtering and data sparsity for sentiment analysis of Twitter.

54. Saba, T. (2020). Recent advancement in cancer detection using machine learning: Systematic survey of decades, comparisons and challenges. Journal of Infection and Public Health, 13(9), 1274-1289.

55. Singh, A., & Jain, S. K. (2020, October). A personalized cancer diagnosis using machine learning models based on big data. In 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 763-771). IEEE.

56. Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys (CSUR), 34(1), 1–47.

57. Shwartz-Ziv, R., & Tishby, N. (2017). Opening the black box of deep neural networks via information. arXiv preprint arXiv:1703.00810.

58. Thakkar, D. (2019). Response Coding for Categorical Data. Retrieved from https://medium.comthewingedwolf.winterfell/response-coding-for-categoric

59. Takenobu, T. (1994). Text categorization based on weighted inverse document frequency. Information Processing Society of Japan, SIGNL, 94(100), 33-40.

60. Thompson, J., Hu, J., Mudaranthakam, D. P., Streeter, D., Neums, L., Park, M., ... Mayo, M. S. (2019). Relevant word order vectorization for improved natural language processing in electronic health records. Scientific Reports, 9(1), 9253.

61. Turkki, R., Byckhov, D., Lundin, M., Isola, J., Nordling, S., Kovanen, P. E., ... Lundin, J. (2019). Breast cancer outcome prediction with tumour tissue images and machine learning. Breast Cancer Research and Treatment, 177, 41–52.

62. Turtle, H. (1995). Text retrieval in the legal world. Artificial Intelligence and Law, 3 , 5–54.

63. Chekure, S. V. (2018). Personalized cancer diagnosis. Retrieved from https://www.appliedaicourse.com/course/7/cancer-diagnosis-using-medical-records

64. Vapnik, V., & Chervonenkis, A. Y. (1964). A class of algorithms for pattern recognition learning. Avtomat. i Telemekh, 25(6), 937–945.

65. Verma, M. (2012). Personalized medicine and cancer. Journal of Personalized Medicine, 2(1), 1–14.

66. Weston, J., & Watkins, C. (1998). Multi-class support vector machines. Technical report, Citeseer.

67. Wu, T.-F., Lin, C.-J., & Weng, R. (2003). Probability estimates for multiclass classification by pairwise coupling. Advances in Neural Information Processing Systems, 16.

68. Wu, J., & Hicks, C. (2021). Breast cancer type classification using machine learning. Journal of personalized medicine, 11(2), 61.

69. Yu, B., & Kwok, L. (2011). Classifying business marketing messages on Facebook. In Proceedings of the Association for Computing Machinery Special Interest Group on Information Retrieval (pp. 24–28). Bejing, China.

70. Zeng, Z., Espino, S., Roy, A., Li, X., Khan, S. A., Clare, S. E., Jiang, X., Neapolitan, R., & Luo, Y. (2018). Using natural language processing and machine learning to identify breast cancer local recurrence. BMC Bioinformatics, 19(17), 65–74.

71. Zhao, H., Li, D. Y., Deng, W., & Yang, X. H. (2017). Research on vibration suppression method of alternating current motor based on fractional order control strategy. Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering, 231(4), 786–799.

72. Zhao, H., Zuo, S., Hou, M., Liu, W., Yu, L., Yang, X., & Deng, W. (2018). A novel adaptive signal processing method based on enhanced empirical wavelet transform technology. Sensors, 18(10), 3323.

© 2024 by the authors. Disclaimer/Publisher's Note: The content in all publications reflects the views, opinions, and data of the respective individual author(s) and contributor(s), and not those of the scientific association for studies and applied research (SASAR) or the editor(s). SASAR and/or the editor(s) explicitly state that they are not liable for any harm to individuals or property arising from the ideas, methods, instructions, or products mentioned in the content.