A Machine Learning Approach to Prediction of deaths due to Monoclonal Gammopathy

Shashank Reddy B*, Sreedhar Reddy V*, Dr. Santhi B*
* School of Computing
SASTRA Deemed University
{shashank1872, reddyvsree}@gmail.com, shanthi@cse.sastra.edu

***Abstract*—Monoclonal Gammopathy of Undetermined Significance (MGUS) occurs in up to 2% of persons of the age 50 or older. Machine Learning methods are needed to learn the vital information embedded in the genetic samples, which in turn can be helpful to develop more robust and accurate models of clinical diagnostics. We use SEVEN methods with classification trees as one of our models. For each of the methods, we employed 10-fold cross validation to estimate the prediction error. Our results produced are the comparison between the machine learning algorithms. The results we achieved, proved to be better when certain preprocessing techniques were used for the specific application of an algorithm. The results varied when missing values are replaced with mode, mean, mean with class label etc. We considered removing the tuples where the missing values are found, with consequence of losing valuable data. The accuracy of 82.6% is achieved by Support Vector Machines, followed by the K-Nearest Neighbor, Logistic Regression .**

Keywords: **Monoclonal Gammopathy, Cancer, Machine Learning, Disease progression to death, MGUS.**

I. Introduction

MM, Multiple myeloma is a malignant plasma component of white blood cells (plasma cells) in the bone marrow. MM is related with an overproduction of an abnormal protein known as monoclonal protein (M-protein). Though this protein itself is not harmful to most people, if too much of this protein accumulates it often causes characteristic osteolytic lesions, anemia, renal failure, and hypercalcemia. [**?**] In contrast, monoclonal gammopathy of unknown significance (MGUS) is an asymptomatic plasma cell dyscrasia that is present in more than 3% of the general white population older than age 50 and has an average multiple myeloma progression risk of 1% per year. [**?**]

Traditional clinical practices rely on close surveillance as the tool for management of multiple myeloma precursor disease. Indeed, definition of MGUS has clearly evolved, and characterization of clinical subtypes and risk stratification models have led to better understanding of disease biology and probability of progression. Despite this, a number of inherent diagnostic challenges continue to plague physicians while managing their patients. Molecular tools used to better identify MGUS signature profiles provide answers to long-sought questions and dilemmas. Of course, more research effort is required in this area. Gaining molecular insight into multiple myeloma precursor disease may have a dramatic impact on clinical management in the future. As expected, several clinical investigations have already begun to study treatment options.

Though there is no known particular cause for accumulating of M-protein, the study found out some risk factors that might increase risk of developing MGUS. They include

- **Age**: Mean age at diagnosis of MGUS is 70.
- **Race**: Africans and African-Americans people are likely to be affected.
- **Sex**: It is more common in men
- **Family History**: Possibility of getting affected is considerable, if anyone of the family member have this condition.

Machine Learning techniques have now been used both to classify different kinds of cancers which are morphologically indistinguishable and to predict response to therapy. In this paper, we discuss the possible Machine Learning techniques to predict the risk of progression of MGUS which has caused death in many people. Empirical results suggested that 10-fold cross validation may provide better accuracy estimates than the more common leave one out cross validation. We evaluate and compare each model against a set of metrics, resulted through the analysis of data from the MGUS data-set. The features that are included in the data set are age, sex, hgb (haemoglobin) etc. The data-set is explained in the Section **??**. Through this analysis using machine learning, can help in diagnosis of a patient. This method efficiently finds out if the progression of MGUS is probable for a certain case.

II. Related Works

Patients with MGUS are at increased risk for progression to multiple myeloma or a related plasma-cell cancer. The risk of progression of MGUS to multiple myeloma or related disorders is about 1% per year. [**?**], [**?**] In a comparison of patients with various monoclonal protein values, in which 0.5 g per deciliter or less was used as a reference value, it is found that the initial concentration of monoclonal protein was a statistically significant predictor of progression to multiple myeloma.

The etiology of MGUS remains unclear and it is a current topic of investigation. Race seems to play a role given the observation that prevalence of MGUS is 2 to 3-fold higher in African-Americans and blacks from Africa compared with whites. [**?**], [**?**] Other identified risk factors for MGUS include older age, male sex, exposure to pesticides, and family history

of MGUS or MM [**?**], [**?**], [**?**]. Thus, previous studies support a role for both genetic and environmental factors in the development of multiple myeloma and its precursor states. [**?**]

A significant increase of the monoclonal protein or development of myeloma, macroglobulinemia or amyloidosis occurred in 18 % of the patients with monoclonal immunoglobulin G (IgG), in 28% with immunoglobulin A (IgA) and in 25% with immunoglobulin M (IgM). [**?**] Two independent studies have demonstrated that most cases of multiple myeloma are preceded by MGUS. Among 71 patients who during a 10-year follow-up time developed multiple myeloma, serum samples consistently demonstrated MGUS in the years before the malignant diagnosis. [**?**] Another study, based on the Department of Defense Serum Repository, showed a very similar finding. [**?**]

When different presenting features were analyzed for predictive value of the malignant transformation, the IgA type of MGUS was the only variable associated with a higher probability of such an event (P less than 0.025). [**?**] The relative risks of developing MM are the following: 2.4 for each 1 g/dL increase of IgG, serum MC, 3.5 for detectable light chain proteinuria, 4.4 for the increase of 1 unit in log. BMPC percentage, 6.1 for age $>$ 70, 3.6 and 13.1 for a reduction in one or two polyclonal Ig. In conclusion, a study allowed the identification of a particular subset of MGUS patients at very low-risk of evolution, who can be considered as having benign monoclonal gammopathy. [**?**]

III. Methodology

Fig. **??** shows us an overview of the methodology pipeline. Our methodology consists of various steps: data preprocessing, feature selection, feature scaling and classification where we start with preparing the dataset, cleaning, training, fine tuning the algorithm, model training and testing. The dataset is taken from UCI [**?**]. We use scikit-learn [**?**] and pandas library for all the steps involved in the methodology. Section **??** to section **??** consist of detailed information of the steps.

A. Data Description

We have a total of 10 attributes including class label ***death*** which is a binary value. The other 9 attributes are,

- ***id*** is the patient identifier.
- ***age*** is the age of person in years at the time of detection of MGUS.
- ***sex*** tells if person is male or female
- ***creat*** is creatinine level when MGUS is diagnosed.
- ***hgb*** is the amount of haemoglobin present in blood at time of MGUS diagnosis.
- ***mspike*** is the size of the monoclonal protein spike at diagnosis.
- ***ptime*** tells us the no.of days from MGUS until diagnosis of a plasma cell malignancy.
- ***pstat*** is a binary value which indicates if there is an interval end in an event.
- ***futime*** tells us the no.of days from diagnosis to last follow-up.

B. Data Preprocessing

Incomplete and noisy data are common properties of real world databases. Data needs to be cleaned before it is processed for better efficiency and accurate outputs. There are a lot of pre-processing techniques that include cleaning and preparing the data for classification. Other pre-processing techniques include Integration and Transformation, Aggregation and Discretization.

The pre-processing done for our data are,

- ***Filling missing values*** Missing values are filled in many ways and we used the mean of the total and substituted with our missing values.
- ***Removal of unnecessary attributes*** Unnecessary attributes are removed so that it doesn't affect the performance of the model.
- ***Encoding the attribute values*** The values in the label *sex* are in the form of characters *M/F* which are transformed to *1/0* using the help of label encoder.

C. Feature Selection

Selecting the right features for classification is very important and it is the key factor for achieving a good performance for a model.

The models are having a very high deviation due to the presence of the *id* label because of its sequence ordering nature. So, the *id* label gets high feature importance, but, it is just a patient identifier which should be dropped from the data-set. The remaining labels are all important to process the output label. So the remaining 8 labels with 1384 patients are taken for classification.

D. Feature Scaling

Feature Scaling is important when the bounds between the labels are different from each other. The upper and lower bounds of the labels are very important and if they vary a lot in the data-set that could affect the performance of the model. There are many feature scaling techniques like standard scalar which makes use of mean and standard deviation, min-max scalar which normalizes based on the minimum and maximum values and robust scalar which makes use of the quartile ranges to normalize.

We used standard scalar as it fits the data based on the mean and the deviation of the data. Standard scalar is given by

$$y_{Scaling} = \sum_{i=1}^{n} \frac{x_i - \mu(x)}{\sigma(x)} \tag{1}$$

where n is the number of patients, $\mu(x)$ is the mean and $\sigma(x)$ is the standard deviation for the label.

E. Classification

Classification is a 2-step process. In first step, a classifier is built to describe a predetermined set of data classes. This is the learning step and is done using the training data, where a classification algorithm builds the classifier by learning from that training set made up of data tuples and their associated class labels. It is also known as "Supervised learning".

Fig. 1. Methodology Pipeline Overview

The data-set consists of 1384 patient details of which 80% are taken for training the data and the remaining 20% are taken for testing the data. The following are the learning algorithms that we used to predict the output label *death*,

1) Support Vector Machine
2) Linear Discriminant Analysis
3) Logistic Regression
4) Decision Tree Classifier
5) Naive Bayes
6) Random Forest Classifier
7) K - Nearest Neighbor

Support Vector Machine: Support Vector Machines(SVM) are the supervised learning models that are used for classification of the data. SVMs are designed for both linear as well as non-linear classification. Linear data is rare and most of the existing data is non-linear. So, in [**?**], they designed the SVM to fit the non-linear data by using the kernel trick which uses kernels in SVM to fit the data accordingly. There are many kernels that can be used on a data. As it is a hyper-parameter we need to experiment it with all the available kernels to find the best fit.

Kernel	Train Accuracy	Test Accuracy
Linear Function	0.779	0.815
Polynomial Function	0.783	0.779
Radial Basis Function	0.807	0.826

TABLE I
SVM KERNELS AND THEIR ACCURACIES

The following kernels and its accuracies that are used for our data are listed in the Table **??**. As we can see Radial Basis Function(RBF) kernel fits the data better than other experimented kernels because it has proved to be a generalizer for many data-sets and in many experiments it is assumed as priors for uncertain situations. RBF is also called squared exponential kernel or gaussian kernel.

Linear Discriminant Analysis: Linear Discriminant Analysis(LDA) is a generalization of fisher's linear discriminant, a method in statistics and Machine Learning used to find a linear combination of features that are characterized by many classes of objects. LDA [**?**], can also be used for dimensionality reduction before classification. It is similar to analysis of variance(ANOVA) and principal component analysis(PCA) where it tries to attempt one dependent variable as a linear combination of other features.

Many different solvers which are used as the characteristic roots of discriminant functions. We used the solvers: *svd*, *lsqr*, *eigen* which are used for classification purpose.

Solver	Train Accuracy	Test Accuracy
Singular value decomposition(svd)	0.781	0.812
Least Squared Solution(lsqr)	0.781	0.812
Eigenvalue Decomposition(eigen)	0.786	0.815

TABLE II
LDA DISCRIMINANT FUNCTIONS AND THEIR ACCURACIES

Table **??**, describes the solvers and their accuracies with all the solvers giving a good performance measure because of their ability to classify well on the data.

Logistic Regression: Logistic Regression is a supervised learning algorithm which develops a model that predicts the binary value of a target. It uses a logistic function named as 'sigmoid' function that helps to determine the relationship between categorically dependent and one or more independent variables [**?**].

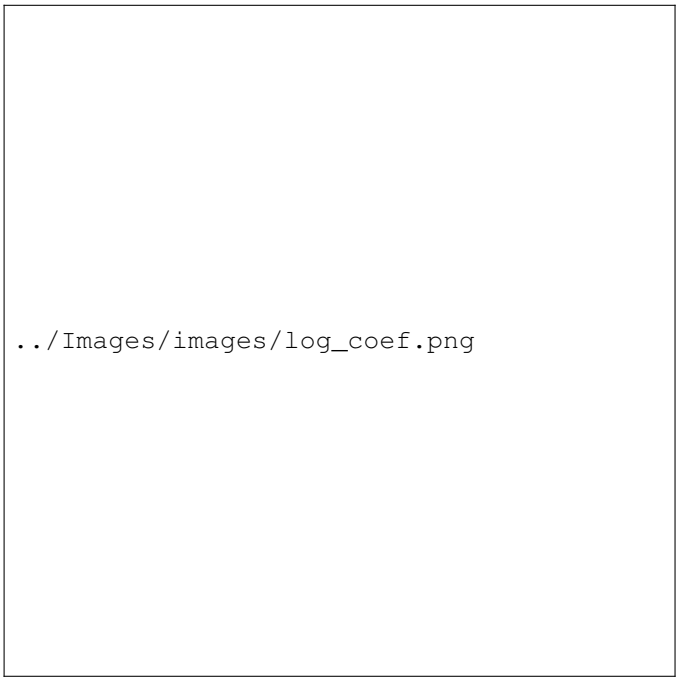
The regularization parameter 'C' passed to the model is a hyper-parameter that can be tweaked accordingly for the best performance. The values of 'C' can be like 0.01, 1, 100 etc. While '1' being the default value of the parameter 'C'. Stronger regularization (C=0.001) pushes the coefficients more towards zero. Figure **??** shows us the coefficient magnitudes for different regularizations when applied on different features.

Decision Tree Classifier: Decision tree is a, supervised learning algorithm, which is widely used classifier. Unlike Naïve Bayes, Logistic regression and other algorithms that belong to the supervised learning algorithm family, it can be used for solving both regression and classification problems. Even when the dataset has missing values, using Decision tree could result in a better output.

When applied directly, Decision Tree causes the overfitting on [**?**] and not being generalized well to the new data. Therefore we pre-pruned the tree by setting the maximum depth max_{depth} parameter to 3 as seen from [**?**]. Thus decreasing overfitting by limiting the depth. We observed a decrease in accuracy on the training set, but a significant increase in accuracy of test data. The feature importance plot of the decision tree classifier as shown in the figure **??** is given by,

Information Gain and Gini Index are the two attribute selection measures used for selecting which attribute can be considered as the root node at each level. Figure **??** shows use the visualization of how gini index acts on the data-set.

Naive Bayes: Naive Bayes, a highly scalable algorithm which is a special case of Bayesian Network where it is assumed that all the features are independent given the class label. Presence or absence of one feature will not influence



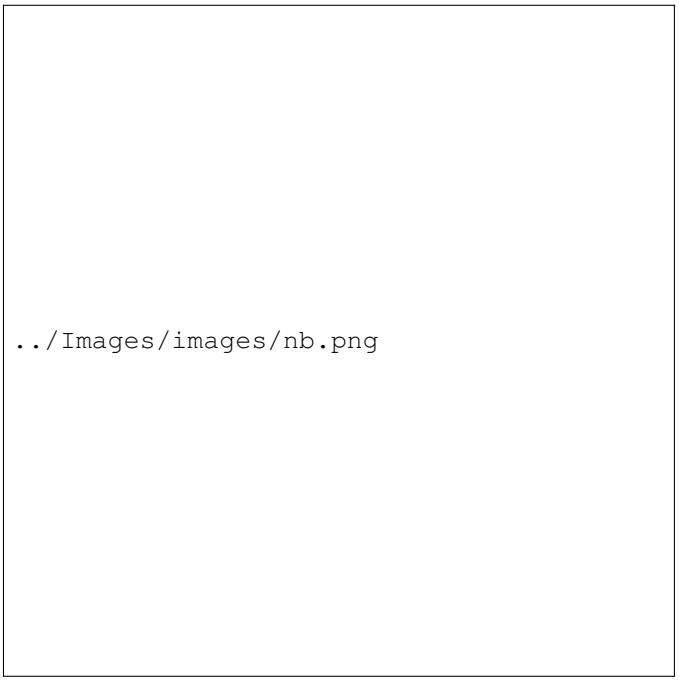


Fig. 2. Coefficient magnitudes for different values on Logistic Regression

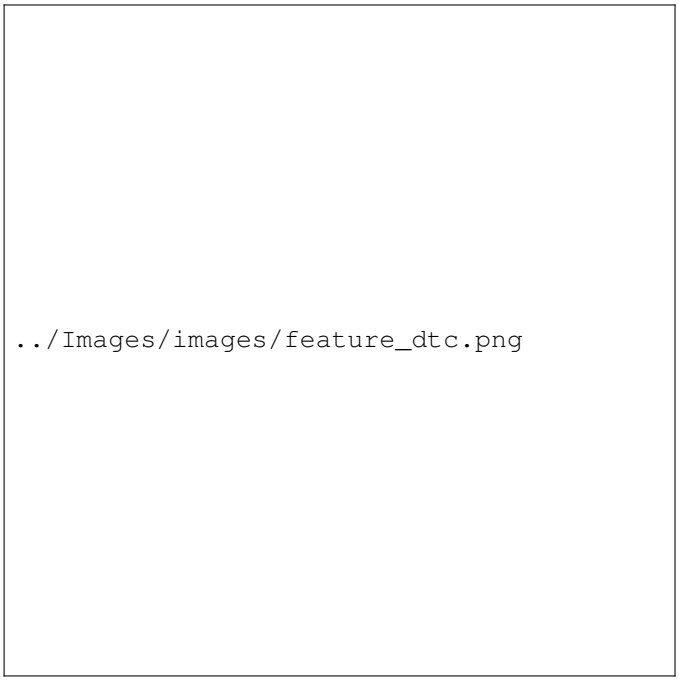


Fig. 3. Feature Importance plot of Decision Tree Classifier

the presence or absence of other features. Given a class label C, the relationship between conditionally independent data x_i can be given by the formula,

The algorithm predicts the class of a previously unseen test case $X = x_1, x_2...x_n$ by choosing the class that maximizes the above equation. In practice the predictor prior probability p(x) can be omitted because the classification focuses on choosing the class that maximizes the equation. From [**?**] we can say that, one of the advantages of the NB classifier is that it requires only few amounts of training data to estimate the class variables needed for classification. It is one of the popular choices for text classification, binary and multi-class classification problems.

Random Forest Classifier: Random Forest, similar to Decision Tree Classifier can be used for solving both regression and classification problems. It is an ensemble learning model. [**?**] The algorithm creates forest with a number of Decision Trees. More the trees in the forest, higher the accuracy of the classifier.

The algorithms gives importance to certain features to make a prediction. Not all the attributes are considered equal and thus the result vary for different data-sets. The randomness in building the random forest forces the algorithm to consider many possible explanations. Thus result of random forest captures a much broader picture of the data than a single tree. The feature Importance plot of the random forest classifier in figure **??**, is given by,

- In this algorithm, **k** features are randomly selected from a total of **n** features.
- Among the **k** features, using the best split point node **d** is calculated.
- The node is now split into daughter nodes using the best split.
- The above 3 steps are iterated until **l** number of nodes are formed.

Fig. 4. Visualization of Decision Tree Classifier using the gini index selection measure

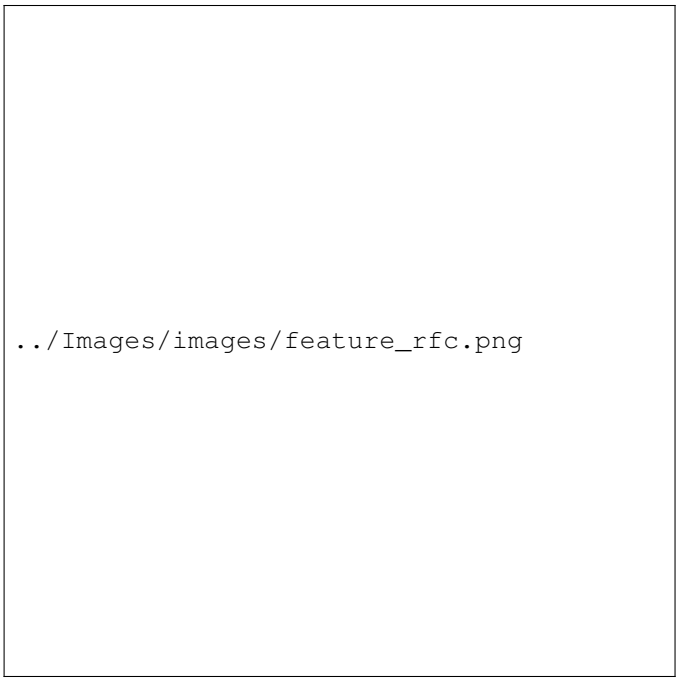


Fig. 5. Feature Importance plot of Random Forest Classifier

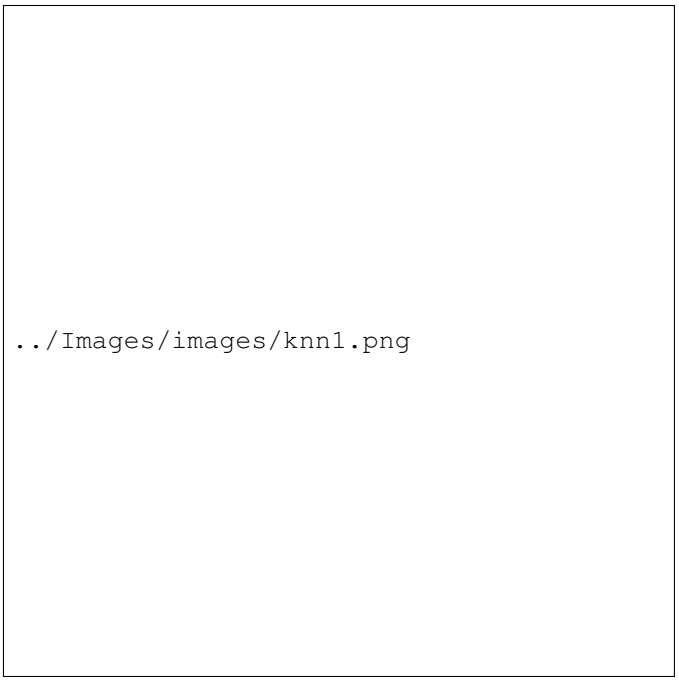


Fig. 6. Accuracies of KNN on train and test data

Repeat the above 4 steps **m** times until **m** decision trees are formed. Two important features of RF are:

1) Ability to achieve high prediction accuracy.
2) Usability of desired capabilities.

K - Nearest Neighbor: K Nearest neighbors is a classification algorithm that is used for regression predictive problems. As seen from [**?**], building the model will only store the training data set. Only if a new input is chosen, it predicts and finds the closest data points in the training data: *nearest neighbor*. KNN is commonly used for its easy of interpretation and minimal calculation time.

The figure below shows the training and test data accuracy on the y-axis against number of neighbors on the x-axis. Consider choosing one single nearest neighbor, accuracy of prediction on training data set is **1**, perfect. But if more neighbors are considered, the training accuracy falls, showing us that using the single nearest neighbor leads to a model that is much complex.

Figure **??** indicates us to choose 17 or 18 neighbors. In our case K=17 neighbors has produced an accuracy of 81.2% while K=13 neighbors has produced an accuracy of 82.3%.

Thus we know how much the choice of factor **K** in the algorithm influences the outcome. The boundary becomes smoother with the increasing value of **K**. The training accuracy and test accuracy are the two parameters needed to access on different K-value, as shown the plot figure. The accuracy of KNN classifier significantly increases by increasing the number of data rows in the training set.

F. Metrics

The combination of all the steps are evaluated under different performance metrics, including accuracy, precision, recall and F-score for the binary classification. The metrics are defined as following,

$$Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{|X_i \cap Y_i|}{|Y_i|} \tag{2}$$

$$Recall = \frac{1}{n}\sum_{i=1}^{n}\frac{|X_i \cap Y_i|}{|X_i|} \tag{3}$$

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2|X_i \cap Y_i|}{|X_i| + |Y_i|} \tag{4}$$

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n}\frac{|X_i \cap Y_i|}{|X_i \cup Y_i|} \tag{5}$$

where X_i is the set of predicted labels, Y_i is the set of true labels and n is the number of samples.

IV. Results and Analysis

Table **??** shows us the model performance for all the algorithms we have experimented. The metrics used for evaluation are accuracy, precision, recall and F_1-score as shown in the section **??**. As we can see, the comparison between the algorithms helped us understand the nature of the algorithm as well as the data used, based on the metrics. The accuracies of all the other algorithms are close to each other by a fraction of percentage as they are tweaked to get the best accuracy using kernels, solvers and hyper-parameters. Except Naive Bayes all the accuracies of other algorithms are above 80%.

The highest accuracy was achieved by Support Vector Machine with the radial basis function as the kernel. Its accuracy is 82.6%. Therefore, the data is non-linear and it takes the form of probabilistic density function which has a global maximum. The lowest accuracy is achieved by Gaussian Naive Bayes but the difference in accuracies between the highest and lowest is only 2.9%. When we consider the data to be linear, the accuracy falls below 60%. Finally, the overall precision, recall and F_1-score for the algorithms are similar for many cases with the highest achieved by Support Vector Machine.

Table **??** gives the performance measure for the output labels whose deaths is classified as **0**. As we can see, the values of the metrics, precision, recall and F_1-score are very less when compared to the metrics in table **??**. That is because the number of patient with the output label **0** are very less when compared to the number of patients with output labels **1**. As the F_1-score is a harmonic mean of precision and recall we can a reasonable values for it unlike the other two. The highest precision, recall and F_1-score is achieved by Random Forest Classifier, Decision Tree Classifier respectively.

Techniques	Precision	Recall	F1-score
Support Vector Machine	0.81	0.59	0.68
Linear Discriminant Analysis	0.77	0.60	0.68
Logistic Regression	0.79	0.59	0.68
K - Nearest Neighbor	0.77	0.64	0.70
Naive Bayes	0.66	0.75	0.70
Decision Tree Classifier	0.72	0.69	0.71
Random Forest Classifier	0.82	0.51	0.63

TABLE IV
PERFORMANCE MEASURES OF THE PREDICTION WHEN DEATHS LABEL VALUE IS **0**

The performance measures for prediction when the output labels is **1** is very high because of the dominance of the binary value **1** in the data. From Table **??**, we can see that the highest precision, recall and F_1-score is achieved by Decision Tree Classifier, Support Vector Machines respectively.

Techniques	Precision	Recall	F1-score
Support Vector Machine	0.83	0.94	0.88
Linear Discriminant Analysis	0.83	0.92	0.87
Logistic Regression	0.83	0.93	0.87
K - Nearest Neighbor	0.84	0.91	0.88
Naive Bayes	0.88	0.82	0.85
Decision Tree Classifier	0.86	0.87	0.87
Random Forest Classifier	0.81	0.95	0.87

TABLE V
PERFORMANCE MEASURES OF THE PREDICTION WHEN DEATHS LABEL VALUE IS **1**

V. Conclusion and Future Work

In this paper, the problem of summarizing different algorithms of Machine Learning used for the prediction of deaths due to Monoclonal Gammopathy is discussed. The focus is mainly on using different algorithms to effectively predict the deaths due to MGUS [**?**] . For prediction we used a refined version of the data-set with 9 attributes that are produced after applying a few data mining techniques. The outcome of the predictive results on the same data-set reveals that Support Vector Machine has outperformed all the other algorithms in terms of accuracy, precision, recall and F_1-score with a score of 82.6, 83, 83, 82 % respectively. The second conclusion gives us an intuition of how the output labels affect the metrics, by analyzing it in detail on the output labels.

The proposed work can be enhanced by expanding its scope of prediction by using the deep learning techniques [**?**], [**?**] like Artificial Deep Neural Networks, Multi-perceptron etc., which have been proven to have increased the performance of many data-sets due to its nature of deep computation.

Techniques	Training Accuracy	Test Accuracy	Precision	Recall	F_1-score
Support Vector Machine	0.807	0.826	0.83	0.83	0.82
Linear Discriminant Analysis	0.786	0.815	0.81	0.82	0.81
Logistic Regression	0.787	0.819	0.82	0.82	0.81
K - Nearest Neighbor	0.819	0.823	0.82	0.82	0.82
Naive Bayes	0.744	0.797	0.81	0.80	0.80
Decision Tree Classifier	0.796	0.815	0.81	0.82	0.82
Random Forest Classifier	0.799	0.808	0.81	0.81	0.79

TABLE III
MODEL PERFORMANCE OF ALGORITHMS