Developing Ensemble Model Using Greedy Approach

MEHAK AGGARWAL (MCA, 03104092021) AND TANIYA JAIN (MCA, 06404092021)

ABSTRACT

Machine learning is a technique to predict future results based on observed data from past, on which the classifier is trained to discover patterns, spot anomalies, test hypotheses, and check assumptions with the help of summary statistics and graphical representations. In today's world, Machine Learning is a technology that is rapidly developing. Without even knowing or realizing it, we use machine learning on a day-to-day basis in applications like Google Maps, Google Assistant, and Alexa. Designing an efficient predictive model has been the main aim of researchers throughout. Various techniques of optimization have been developed to improve prediction accuracy. Ensemble learning is a relatively new technique that is widely adopted to improve the performance of machine learning models. In this research, we have developed an ensemble model by combining popular machine learning models. To develop the ensemble model, we have used an incremental greedy approach which gives us an accuracy of 88.99%. This result can be increased significantly in the future by simulating other machine-learning models which have not been tested in this paper.

KEYWORDS

Ensemble learning, Optimisation Technique, Greedy, Majority Voting, Confusion Matrix, Support Vector Machine, K-Nearest Neighbors, Decision Tree

---

I. INTRODUCTION

Ensemble learning is a technique where the results of various models are combined and a common set of output is generated based on certain parameters. Ensemble Techniques improve the accuracy of results significantly. Based on the classifiers used, ensemble models can be homogeneous or heterogeneous. Homogeneous is when results of the same base model are combined. The model is called heterogeneous when base models are different.

An ensemble model ensures that the result predicted is the best result out of all the base models. Results can be decided based on three fundamental techniques - max voting, averaging ,and weighted average. An ensemble model can be passed through a voting classifier which selects the output suggested by the majority of models. While this technique is favored in classification problem statements, averaging is best suited in predicting the results of regression problems. Averaging can also be used in case of categorical variables to calculate probabilities of nominal

values. The weighted average technique can be useful in favoring specific models by defining the importance of each model.

Ensemble Techniques can be categorized into three categories - Bagging, Boosting and Stacking. Bagging or Bootstrap Aggregation trains multiple weak learners such as Decision Tree parallelly and outputs the aggregated result of all base models. Sample generated to train each weak learner may or may not be identical as samples are fetched with replacement technique. Hence, base models can be trained independently. Random Forest is the common bagging algorithm which overcomes the limitations of overfitting in its base learner, Decision Tree. Unlike bagging, base learners in Boosting depend on the output of each other, the output of one base model is fed to the other in a sequential manner. All base learners are not given equal importance. Incorrectly classified data points are given more importance in training the next model. Gradient boosting and AdaBoost are common boosting algorithms. While bagging aims to reduce variance, the aim of boosting is to reduce the bias in prediction. In Stacking, 2 or more base models are trained and results are sent to a meta-model to combine predictions. Meta-models are trained on data that was not sent to base-models. Stacking takes place in levels where base models are at level 0 and meta-model is at level 1.

Optimisation in machine learning is to adjust hyperparameters in order to improve performance of the machine learning algorithm. Hyperparameters decide the trade-off between bias and variance. Optimisation is conducted by following certain techniques. Gradient Descent, Stochastic Gradient Descent, and Genetic algorithms are some of the common optimization techniques. Gradient Descent technique is applied to minimize the cost function, thereby reducing the error. The aim is to reach a local minima where the cost function cannot be reduced further. The hyperparameter learning rate, which is the step size at each iteration, decides when it is adequate for the model to stop looking for a global minima. Selection of a learning rate that is neither too small nor too large plays a key role in this process. Stochastic Gradient Descent introduces an element of randomness in Gradient Descent where it randomly selects data points in each iteration. Genetic algorithms are based on the theory of evolution. In this technique, models that give better results are kept and others are discarded. These models are further mutated with other models in order to get better results. When this process continues up to some iterations models are said to be adaptations of its previous form.

The objective of this project is to create a new ensemble model using a greedy optimization technique so that there is an improvement in F-measure. We have formulated three research questions for our research.

RQ1: What is the performance of various classifiers as compared to a basic ensemble model?

RQ2: What are the simulations applied to execute the proposed greedy approach to develop an ensemble model using various classifiers?

RQ3: What is the efficacy of the proposed ensemble model in comparison to the base ensemble model?

This paper is organised as follows: Section II describes the dataset used, Section III explaines the project methodoly, Section IV is the analysis if research work, Section V concludes research work.

II. DATASET

The Dataset we have used for this study is the 'collegePlace' dataset [https://www.kaggle.com/datasets/tejashvi14/engineering-placements-prediction]. These are the actual statistics shared by a university on on-campus hiring for its engineering programme (2013 and 2014). This data on college placements is gathered from the kaggle. This information is utilized to forecast and evaluate a student's placement based on his or her background. There are of 2966 observations(records) and 8 variables. Independent variables contain information about students. Dependent variable which is the categorical variable refers to student placement status. There are 2 categorical variables, 3 numerical variables and 3 boolean variables in the dataset. There is no missing or incorrect value in the dataset. The dataset is balanced with 1639 students marked placed and 1327 students marked as unplaced.

- **Age** – This is a numerical variable. It mentions the age of the student sitting for placements.
- **Gender** – A categorical variable, that specifies weather a student is a male or female.
- **Stream** – A categorical variable, that specifies the branch of the student. There are six categories - Civil, Computer Science, Electrical, Electronics and Communication, Information Technology, Mechanical.
- **Internships –** Specifies the number of internships done by the student.
- **CGPA -** CGPA plays an important factor in telling if a student will get placed or not. If the CGPA is low, there is less probability for the student to get placed.
- **Hostel –** A boolean attribute, to mention whether or not a student stays at a hostel. The student's location can affect the probability of getting placed or not.
- **HistoryOfBacklogs –** Specifies if there are one or more subjects as backlog. As with the marks, Students with backlogs are more likely to not get placed compared to those with no backlogs.
- **PlacedOrNot** – This is a dependent variable that predicts if the student got placed or not.

III. METHODOLOGY

This research mainly focuses on developing ensemble models to increase the efficiency and reliability of predictions made. The newly developed ensemble model is then compared with a basic ensemble model. We consider the prediction models developed by five machine learning classifiers which are Support Vector Machine (SVM), Logistic Regression (LR), Gaussian Naive Bayes (NB), Decision Tree (DT), and K-Nearest Neighbor (KNN). Both the base ensemble model and optimised ensemble model use these 5 classifiers as base learners. Figure 1 shows a brief overview of the research design. The optimiser is viewed as a black box in this figure.

A. Selection and collection of Dataset

Dataset is selected based on the problem statement. The required criteria was to select a dataset containing one dependent variable which is categorical(binary) and a number of independent variables. A clean dataset was preferred in order to minimize the time required for data cleaning and preprocessing.

B. Data Preprocessing

When a model is trained on preprocessed data, it is said to give better result as preprocessing removes various inconsistencies in the dataset allowing the model to identify patterns easily. Preprocessing improves the quality of data by giving the data a structured format. It can be applied as per the requirements of the dataset and problem statement. We have listed the commonly used preprocessing techniques –

- **Handling missing or incorrect values**: Missing values produce ambiguity while training the model, the record either has to be fabricated or removed. Incorrect data can lead to misleading results. Hence, it is important to handle this kind of data. However, there was no such value in our dataset.
- **Handling categorical variables**: Many machine learning models (except Decision Tree) don't identify the categorical arguments supplied to them. Hence, it is advisable to convert categorical attributes into numerical attributes. It can be done in two ways - LabelEncoder and OneHotEncoder. LabelEncoder is used when categories are related and can be assumed to be ranked by models. OneHotEncoder is for conditions when categories are not ranks but independent values.
  We have converted two categorical attributes - Gender and Stream, to numerical attributes using OneHotEncoding.
- **Dimension Reduction:** In machine learning, a large number of dimensions is considered a curse for model training. This is because it increases the chances of overfitting. There are merely 7 independent attributes in our dataset, hence we need not reduce the dimensionality.
- **Data Sampling:** A dataset can be divided into a number of classes based on its dependent variable. It is ideal for each class to have a similar number of data points so that a majority class is not favored. This kind of dataset is common in fraud detection models. Sampling the data will increase the chances of correctly classifying minority classes. The two classes in our dependent variable have almost similar number of datapoints. Hence, our dataset is a balanced dataset.
- **Normalization and Standardization:** Normalization is done when the values of the dataset are scattered. Normalizing data scales it into a smaller range. Standardization does not bring the data in a range, rather it focuses more on centering the values so that its mean becomes zero. There is no need for feature scaling in our dataset.

C. Training

Typically, when generating a machine-learning model, the dataset is split into a training set and a testing set. The majority of data is given for training and the rest of the data is used for testing the trained model. Splitting the data into testing and training is an important step in machine learning. In this project, we have split the dataset into 85% for training and 15% for testing. Other hyperparameters are: random_state = 0, stratify = y.

We have developed prediction models using these 5 machine-learning classifiers –

- **Support Vector Machine (SVM):** This algorithm tries to find support vectors such that the area of the hyperplane can be maximized. SVM can be of two types – linear and polynomial. Here it is used for classification problem.
- **Logistic Regression (LR):** This algorithm can only be used for classification problems. It is a supervised learning approach, where data points are classified based on a sigmoid function.

- **Gaussian Naive Bayes (NB):** This classification algorithm is based on Bayes' Theorem of probability. All the attributes are assumed to be independent while training. Other variations of NB are Multinomial NB and Bernoulli NB.
- **Decision Tree (DT):** This is a powerful machine learning algorithm which selects attributes as nodes of trees so that impurity is minimum. Impurity is measured in various ways such as – GainRatio, GinniIndex and InfoGain. We have used the cart decision tree model.
- **K-Nearest Neighbor (KNN):** This algorithm is a supervised machine-learning algorithm. For classification of a datapoint, its distance from known classes is calculated and the class at minimum distance is given as the result of classification.

D. Performance Metrics

The performance metrics used to evaluate various machine-learning models are Accuracy, Precision, Recall, F-measure. All these values can be derived from a performance measure confusion matrix which gives the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values of the tested dataset. In this research, we have considered F-measure as the primary metric to evaluate the performance of various machine-learning models.

- **Accuracy –** A measure of the number of classes predicted accurately from the total number of data points tested. This can however be sometimes misleading when majority and minority classes are present in dataset, classifier tend to predict all the classes as the majority class. Hence the number of true negatives will be zero and accuracy will still be high.

  Accuracy = (TP+TN)/(TP+TN+FP+FN)
- **Precision –** Measures the number of correctly classified positive classes out of all the classes classified as positive.

  Precision = (TP)/(TP+FP)
- **Recall –** Measure of the number of classes correctly classified as positive out of all the classes that actually positive.

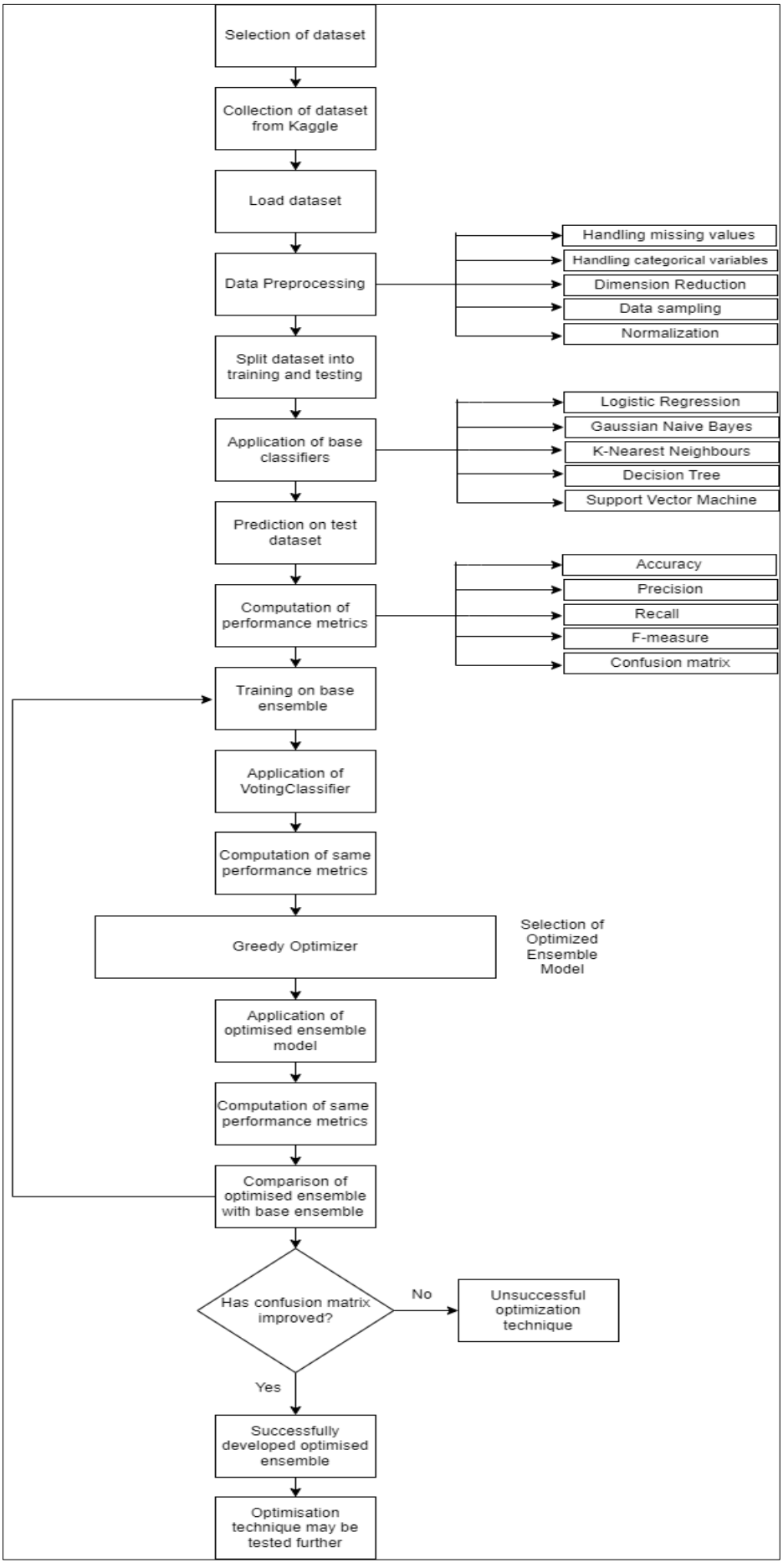
  Recall = (TP)/(TP+FN)
- **F-measure –** F1-measure gives a balanced value of FP and FN. This is calculated by taking a harmonic mean of precision and recall.

  F-measure = (precision*recall)/(precision+recall)

E. Ensemble Model

The concept of the ensemble model was introduced to improve the results of the prediction model by combining various models. The prediction results in this research are evaluated on a basic ensemble model first. The base ensemble used is a combination of all the classifiers described in the Training section above. Second, an optimised ensemble model is developed using an incremental greedy approach mentioned in the next section and illustrated in figure 2. We have used a majority voting classifier to combine all the base learners of an ensemble model. **Base Ensemble Model = SVM + DT + LR + NB +KNN.**

Figure 1. Flowchart representing research design



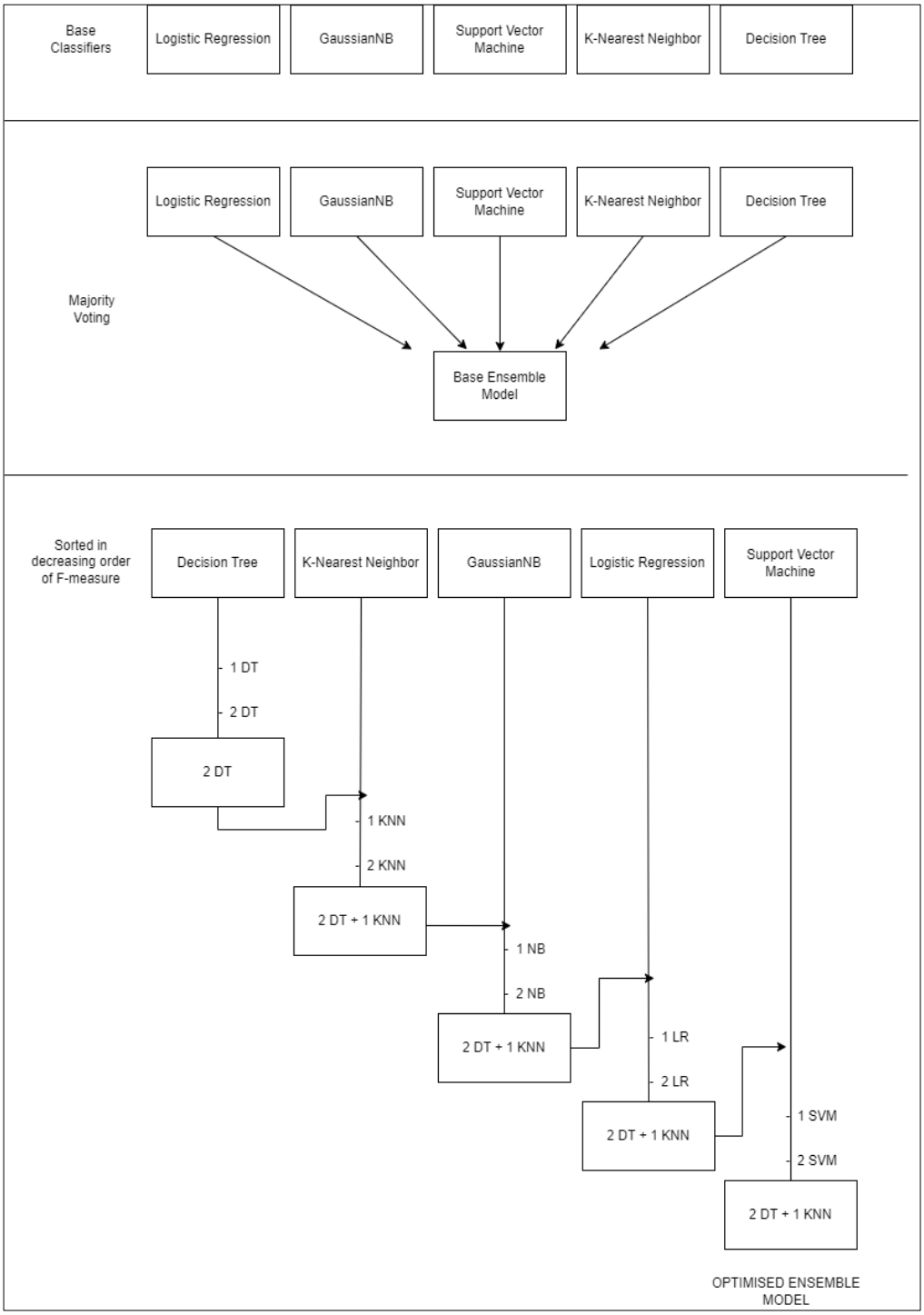


Figure 2. Various machine learning models, ensemble model and optimised

F. Optimization Technique

We have adopted an incremental greedy approach to find the best-suited ensemble model on our chosen dataset. F-measure is chosen as the greedy parameter, that is, the aim is to achieve a balanced value of recall and precision. First results from classifiers are taken into consideration and they are sorted in decreasing order of F-measure. This is done because models giving more balanced results are preferred compared to biased models. As the outputs of each step are taken

into account before going to the next step, this can be called a variation of Boosting Technique. Each base model from the sorted array is considered for maximum of two iterations, if none of the two iterations give a better F-measure than the current best value, then that classifier is discarded. Classifiers have been discarded because in a greedy approach any model which does not contribute in a positive way need not be adopted for further computation.

The F-measure of all possible ensembles from these is evaluated and the model giving a positive result, i.e. the model giving a better F-measure than the current model is selected. In this way, layers may keep on adding with each iteration and the required ensemble model is developed incrementally. After each iteration, the ensemble giving the best result is selected and fed as the first layer of the next base learner from the sorted array. This process is continued till the number of base learners in the sorted array exhaust. Finally, results from the generated ensemble are combined using the max voting approach. The optimizer is illustrated in Figure 2.

G. Algorithm of the proposed approach

This section contains a basic algorithm of the adopted optimization technique. Certain parts of the algorithm are assumed to be calculated with the help of helper functions.

```
Optimizer()
{
        A = Array of models of size n
        ReverseSort(A)                                  —------- (nlogn)

        bestFmes = 0
        Ensembler = []
        for i = 1 to n                                  —------(n+1)
        {
                Temp = Ensembler
                for j = 1 to 2                          ---------(2)*(n)
                {
                        Temp.append(A[i])
                        if(Temp.Fmes>bestFmes)
                        {
                                bestFmes = Temp.Fmes
                                Ensembler = Temp
                        }
                 }

        }
        return Ensembler
}
```

Time Complexity = O(nlogn)

IV. RESULTS AND DISCUSSIONS

This section discusses the obtained results and answers all the RQs mentioned in the Introduction section. The dataset is divided into 85% training samples and 15% testing samples.

RQ1: What is the performance of various classifiers as compared to a basic ensemble model? The classification models for this RQ were developed after changing the categorical attributes to numerical attributes. Models are analyzed based on their Accuracy, Precision, Recall, and F-Measure. The values of Accuracy, Precision, Recall, and F-Measure are in the range of 0.757 - 0.867, 0.770 - 0.956, 0.756-0.870, and 0.782-0.869 respectively. Base Ensemble Model shows better performance than SVM, LR, and NB but is not as good as DT AND KNN. Figure 3 shows the plot of various performance measures used in this RQ. Tables 1 and 2 show the values associated with the performance measures of classifiers and base ensemble model respectively.

Base Model	Accuracy	Precision	Recall	F-Measure
Support Vector Machine (SVM)	0.766	0.809	0.756	0.782
Logistic Regression (LR)	0.757	0.770	0.801	0.785
Gaussian NB (NB)	0.784	0.770	0.870	0.817
Decision Tree (DT)	0.867	0.956	0.797	0.869
K- Nearest Neighbor (KNN)	0.858	0.903	0.833	0.867

Table 1. Performance Measures of Base Classifiers

Base Ensemble	Accuracy	Precision	Recall	F-measure
SVM+DT+KNN +LR+NB	0.816	0.828	0.841	0.835

Table 2. Performance Measures of Base Ensemble Model

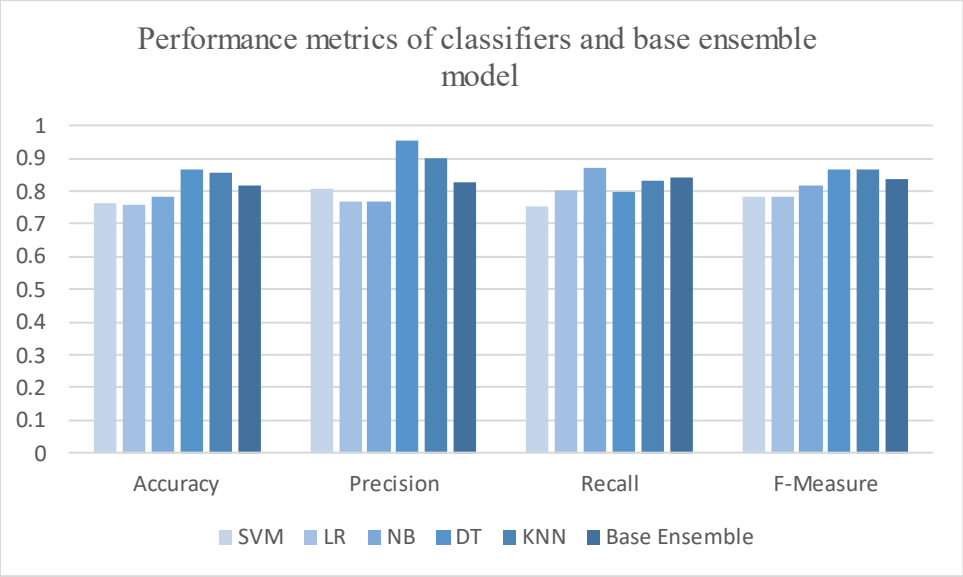


Figure 3. Performance Measures of classifiers as compared to base ensemble model

RQ2: What are the simulations applied to execute the proposed greedy approach to develop an ensemble model using various classifiers?

The Ensembler is initialized as an empty array, only the layers giving better results than the known best-value of F-measure are selected. As explained in the Methodology Section, an incremental greedy approach is adopted in developing ensemble models and the model with the best F-measure value is selected. Every base learner from the sorted array can be added a maximum of 2 times. The F-measure value of all the simulations is greater than the f-measure of the base ensemble (0.835) in RQ1. The results of each simulation are recorded in Table 3. **2DT + 1 KNN** is selected as the optimized ensemble model with an F-measure of **0.894** using the proposed greedy approach.

Simulations	**F-Measure**	**Result**
<u>xDT</u>		
1 DT	0.869	selected
2 DT	0.892	selected
<u>2DT + xKNN</u>		
2 DT + 1 KNN	**0.894**	**selected**
2 DT + 2 KNN	0.872	not selected

2DT + 1KNN + xNB		
2 DT + 1 KNN + 1 NB	0.883	not selected
2 DT + 1 KNN + 2 NB	0.878	not selected
2DT + 1KNN + xLR		
2 DT + 1 KNN + 1 LR	0.878	not selected
2 DT + 1 KNN + 2 LR	0.880	not selected
2DT + 1KNN + xSVM		
2 DT + 1 KNN + 1 SVM	0.878	not selected
2 DT + 1 KNN + 2 SVM	0.871	not selected

Table 3. Performance Measures of Heterogeneous Ensemble Models

Model	Accuracy	Precision	Recall	F-Measure
SVM+DT+KNN+LR+NB	0.816	0.828	0.841	0.835
2 DT + 1 KNN	0.890	0.958	0.837	0.894

Table 4. Comparison of base models with proposed ensemble model

RQ3: What is the efficacy of the proposed ensemble model in comparison to the base ensemble model?

We compare the performance of base ensemble model with the proposed ensemble model, developed to improve the F-measure. According to the results in Table 4, the accuracy of the optimized ensemble model shows an increase of 0.074 as compared to the accuracy of the base ensemble model. The targeted value of F-measure shows a significant increase from 0.835 in the base ensemble to 0.894. As illustrated in Figure 4, other performance measures also showed a significant improvement.

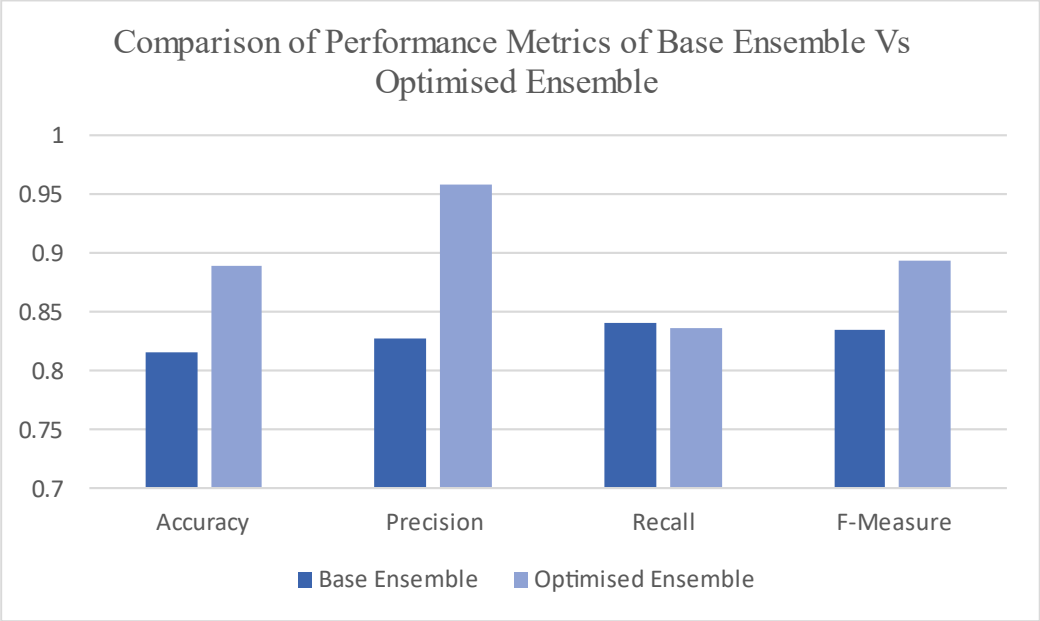


Figure 4. Comparison of base models with proposed ensemble model

V. CONCLUSION

The developed ensemble model shows significantly better results as compared to the base model. The accuracy of our ensemble model has increased to 88.99% from the previous range of 75.7% - 86.7% among classifiers and 81.6% in the base ensemble. Hence, this approach can be adopted in order to increase the reliability of the prediction model. In the future, more machine learning models can be tested to conform to the applicability of this greedy approach. This approach can be tested on varied datasets having a large number of data points.