Capstone Project

Machine Learning Engineer

Nanodegree

Student Performance Prediction in order Organize Specialized Training to Improve Teaching Learning Process

Overview

The target environment of our project is students studying Bachelor of Technology in Electronics and Communication Engineering, JNT University, Kakinada, India. The courses consists of Eighth Semesters out of which First and Second semesters are common for all branches (e.g. Mechanical Engineering, Electronics and Communication Engineering, Civil Engineering, Computer science Engineering etc). Every semester consists of labs and theory subjects. Eighth semester consists of project work. In order to improve quality of teaching learning process, it is required to identify students that would pass or fail in semester examinations and provide special attention to students based on their performance. For example, for the students who may fail one or two theory subjects special classes to be conducted. For the students who pass all the subjects, placement training should be provided

Existing Methods

Zhivu Liu et al in "Prediction and Analysis for Students Marks Based on Decision Tree Algorithm" 2010 3 rd international conference on intelligent networks and intelligent systems proposed model using Decision Tree to predict student marks. G.Narasinga Rao, Srinivasan Nagaraj in "A Study on the Prediction of Student's Performance by applying straight-line regression analysis using the method of least squares" proposed model to predict student performance using straight-line regression analysis Farhana Sarker, Thanassis Tiropanis and Hugh C Davis in "Students Performance Prediction by Using Institutional Internal and External Open Data Sources" predicted students performance using institutional internal and open source data.
Cortez, P. & Silva, A. "Using data mining to predict secondary school student performance"

Proceedings of 5th Annual Future Business Technology Conference, Porto, Portugal proposed model to predict secondary school student performance. B.K. Bharadwaj & Pal., S. 2011 in “Mining Educational Data to Analyze Students Performance”, International Journal of Advance Computer Science and Applications (IJACSA) presented model to analyze students performance.

Problem Statement

Teaching learning process would be improved by giving special attention or specialized training to students based on their expected results in final examinations and finding groups of students. There would be 60 students attending graduation courses in a classroom, but all these 60 students won’t be having the same outcome. Some students may require specialized training in specific subjects and others in some other set of subjects. So identifying students based on their expected results may allow professors to conduct specialized courses to only required students.

The problem of predicting students performance can be accomplished using machine learning algorithms, the problem can be a classification problem since the inputs are having only two classes pass and fail. The expected outputs are also pass or fail. I would train the classifiers using historical data of student performance and predict new student performance.

Benchmark Model

Since this dataset was collected recently and was not public, there are no existing benchmark models for this dataset. So, Naive model may serve as benchmark model for this project. Naive model I am planning to use is predicting all students will pass in all 5 th semester subjects.

Evaluation Metrics

The output would be prediction for 4 subjects, so I am planning to train four classifiers. For the classification models F-scores serve as evaluation metrics. The class labels are imbalanced because the number of students pass may not be equal to number of students fail, so using accuracy as evaluation metric is not a good idea. Alternatively using F score (2*(precision*recall/precision+recall))will give more accurate measure of accuracy in our case where the classes are imbalanced.

Analysis

Data Exploration

The dataset consists of record of results by students of 2014-2017, 2013- 2016, 2012-2015,2011-2014,2010-2013 batches in all semesters (1 to 8). The data is categorical i.e. it consists of two classes (either 'Pass' or 'Fail'). Every batch consists of 120 students of two sections (Section A and B). These mark sheets are provided by university to department of Electronics and Communication Engineering and are original. The main aim of this project was topredict the result/performance of students in fourth sem and to provide specialized training forstudents based on their performance, so, data of results of students of first, second, third and fourth sem was considered in modelling.

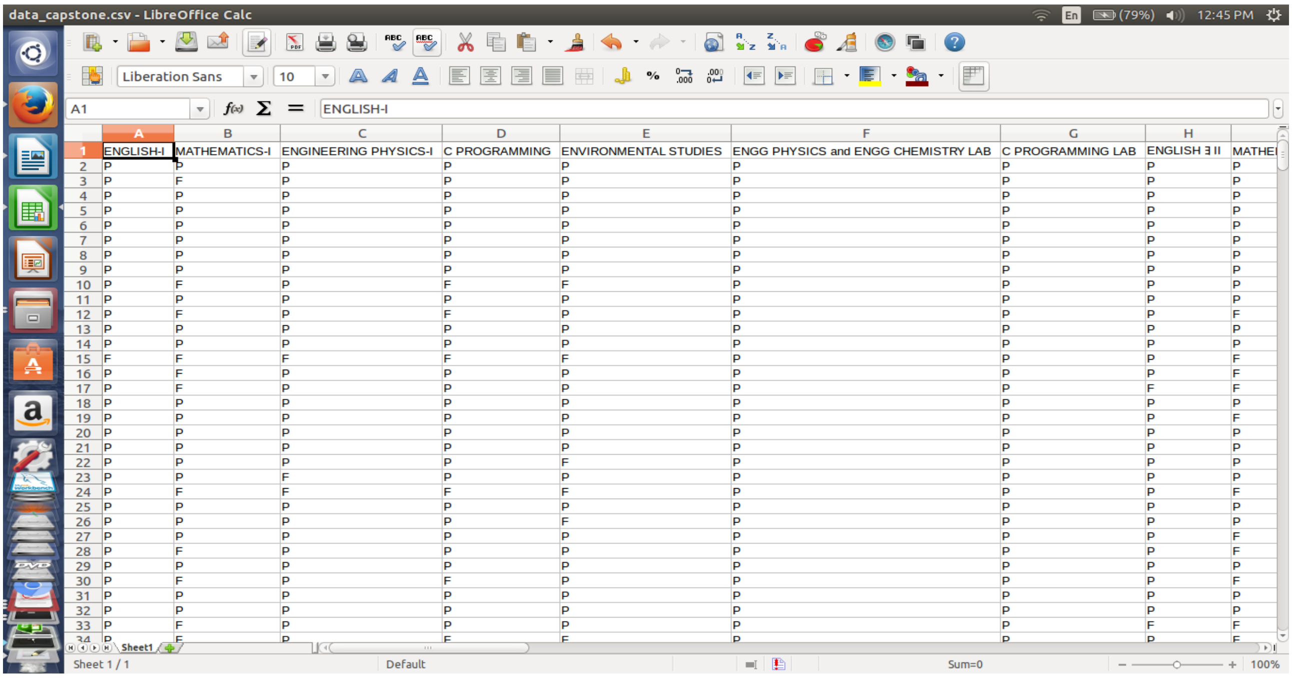
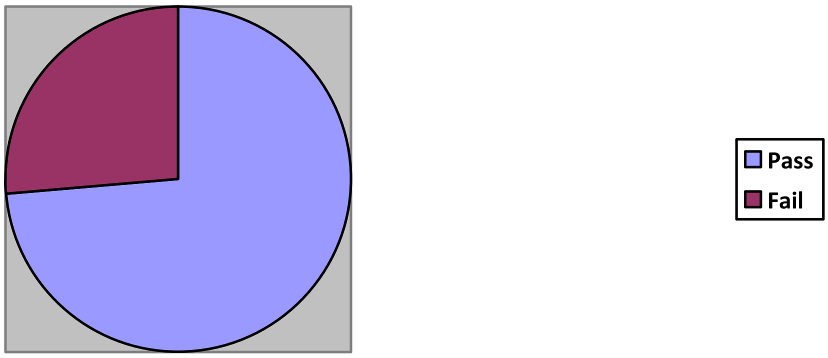


Figure1:Data Used in project

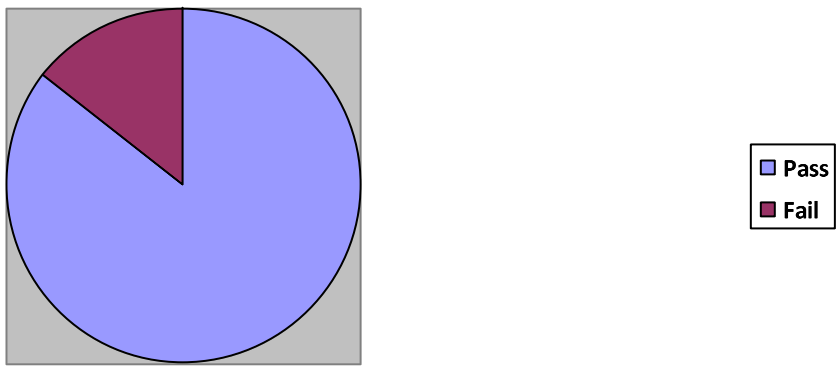
The syllabus and subjects of 2010-2013, 2011-2014 and 2012-2015 was different from 2013-2016, 2014-2017 due to revision of syllabus in 2013. Some new subjects are introduced in 2013 and some subjects are removed. The target variables (subjects interested to predict) are six, namely Switching Theory and Logic Design, Analog Communications, Electronic Circuit Analysis, Electronics Circuits and PDC Lab, Analog Communications Lab, EMWTL, English Communication Practice Lab. Pass percentage of students in Labs are more than 95%, so four theory subjects “Switching Theory and Logic Design”, “Analog Communications”, “Electronic Circuit Analysis”, “EMWTL” are taken as target variables. Pass percentage of Switching Theory and Logic Design was 73.28%, pass percentage of Analog Communications was 85.59%, pass percentage of Electronic Circuit Analysis was 77.66%, pass percentage of EMWTL was 73.49%.

Data Visualization

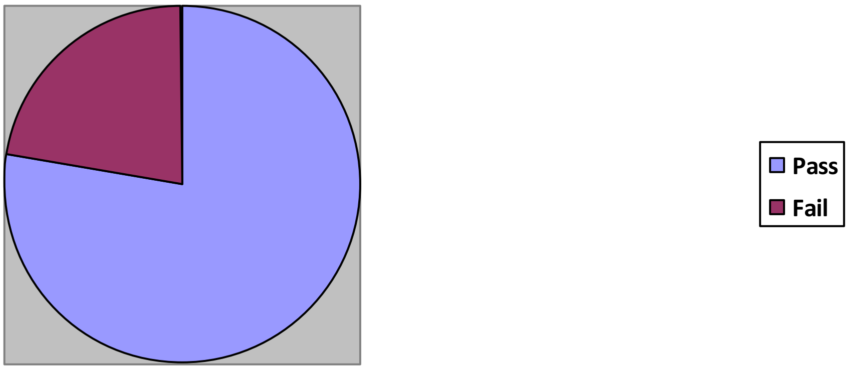
The pass and fail percentage of students in “Switching Theory and Logic Design” was shown in FigA, the pass and fail percentage of students in “Analog Communications” was shown in FigB, the pass and fail percentage of students in “Electronic Circuit Analysis” was shown in FigC, the pass and fail percentage of students in “EMWTL” was shown in FigD.



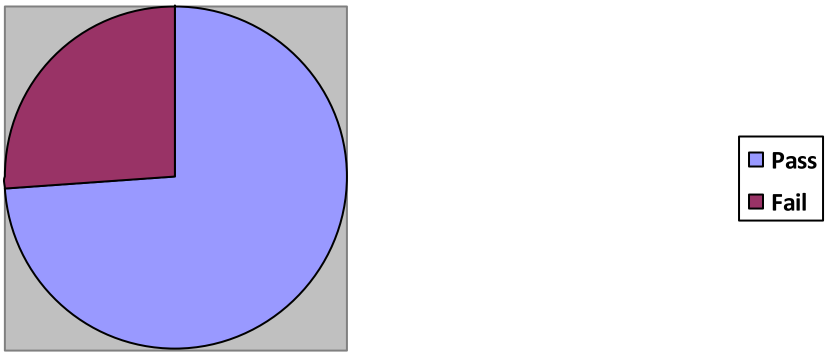
FigA: Pass percentage of students in “Switching Theory and Logic Design”



FigB: Pass percentage of students in “Analog Communications”



FigC: Pass percentage of students in “Electronic Circuit Analysis”



FigD: Pass percentage of students in “EMWTL”

Data Pre-processing

The dataset consists of categorical data. In order to get good results categorical data should be converted to numerical values. Categorical data was converted to numerical data using pd.get_dummies(input data), which resulted in 40 input variables (original data consists of 28 input variables). The four output variables are also categorical, so they are also converted to numerical values using LineEncoder(). The output variables are having only two categories. The dataset was divided to training set and test dataset (80% of data to test set and 20% data to test set).

Naive Predictor

Naive model is used as benchmark model for this project. In Naive model all the students are predicted to pass the exam. The accuracy and F-Score of Naive model for Switching Theory and Logic Design was 0.7328 and 0.7742, the accuracy and F-Score of Naive model for Analog Communications was 0.8559 and 0.8813, the accuracy and F-Score of Naive model for Electronic Circuit Analysis was 0.7766 and 0.8129,the accuracy and F-Score of Naive model for EMWTL was 0.7349 and 0.7760.

Methodology

Step 1: Selection of Classifiers

I selected Decision tree classifier, SVM classifier and Gradient Boost Classifiers. I selected these classifiers because

Decision tree classifier

Decision tree classification is a non parametric supervised learning algorithm that maps observations about an item to conclusions about the items target value. Decision trees classifies the data using sequence of questions, the next question asked depends on answer to the current question. Decision trees can handle both numerical and categorical data. Decision trees are used in face detection (https://www.researchgate.net/publication/4232792_Component-based_robust_face_detection_using_AdaBoost_and_decision_tree), detection of network intrusion (https://www.researchgate.net/publication/224377436_Network_intrusion_detection_using_feature_selection_and_Decision_tree_classifier), digital image forgery detection(http://www.ijera.com/special_issue/Humming%20Bird_March_2014/Version%20%204/DD1823.pdf) etc.

The strengths of Decision tree are

1.Decision trees are simple to understand and interpret

2.Decision trees can handle numeric and categorical data

3.Decision trees require less data preparation

4. Decision trees performs well on large data sets

Weakness of Decision Trees

1. Decision trees may over fit

2.Decision trees may be unstable for small variations of data

The data given is a subset of student data and a random sample of performance. The given data consists of categorical *features. So I expect that Decision trees can produce good model*

SVM classifier

Support Vector Machines is a maximum margin classifier, it performs linear classification well with largest possible margin. It also performs non linear classification well using kernel trick. Support vector machines are used in industrial applications, credit score analysis(http://www.sciencedirect.com/science/article/pii/S095741740600217X), credit rate analysis (http://www.sciencedirect.com/science/article/pii/S0167923603000861), bankrupty prediction (http://www.sciencedirect.com/science/article/pii/S095741740400096X) , hand written text recognition, etc.

Strengths of SVMs

1. Effective in high dimension spaces

2.Different kernels can be specified for decision functions

Weakness of SVMs

1. SVMs may take more time to train and predict

2. If number of features are greater than samples, SVMs may exhibit low performance

The data consists of less features compared to amount of data. The data set is not very large, So i think SVMs may perform well

Ensemble Learning(Gradient Boosting)

Ensemble learners uses multiple learning algorithms to get more performance, Ensemble learners combine many weak learners to classify data. Ensemble learners are used in anomaly detection (http://www.ise.bgu.ac.il/faculty/liorr/EnsOfFeatureChains.pdf), Android Malware detection (http://www.academia.edu/12348488/High_Accuracy_Android_Malware_Detection_Using_Ensemble_Learning). Gradient Boosting is a type of Ensemble learner that divides the data to train weak learners and uses weighted average to model the predictor. Gradient Boosting is used in mobile recommendation system (http://ieeexplore.ieee.org/document/7727431/), web search ranking, ecology (http://scikit-learn.org/stable/modules/ensemble.html)Biometric finger vein authentication(http://ieeexplore.ieee.org/document/7754239/) etc

Strengths of Gradient Boosting

1. Very less chances of over fitting

2. Natural handling of mixed type of data

3.Robustness to outliers

Weakness of Gradient Boosting

1. Scalability due to sequential nature of boosting

The problem is a binary classification problem, the data is categorical, *since the chance of over* *fitting for ensemble algorithms is less, I selected Gradient Boosting algorithm*

Step2: Selecting best classifier

The selected classifiers namely Decision tree classifier, SVM classifier and Gradient Boost Classifiers is used to model the train data and test data was tested on these trained models. Output results such as Training Time, Prediction Time, Train Dataset Accuracy (Acc Train), F-Score on Training Data (F Train), Test Data Set Accuracy (Acc Test), F-Score(F Test) on Test Data of the above mentioned classifiers are given below

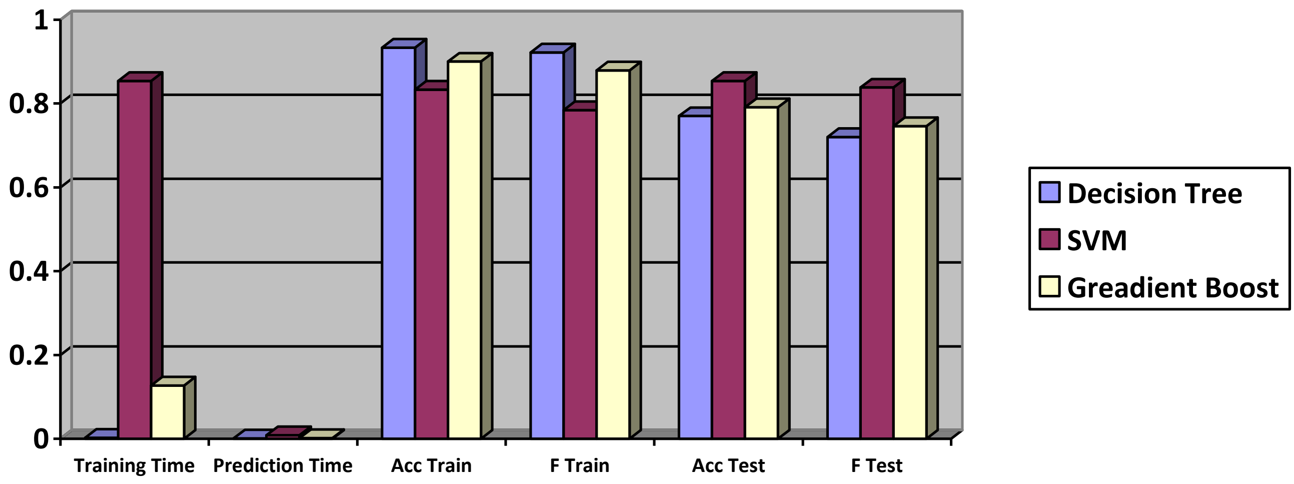


Fig2: Result for “Switching Theory and Logic Design”

Fig2 shows the output of target variable Switching Theory and Logic Design, SVM classifiers has taken more Train time compared to other classifiers, the performance (accuracy and F-score) of decision tree classifier is better on training set, but its performance on test set was bad, decision tree classifier has shown signs of over fitting. Similarly the performance Gradient Boost classifier was good on training set and not up to mark on test data set. The performance of SVM classifier was good on training data set and test dataset. Even though Train Time is more for SVM, based on its performance, SVM classifier was selected to model target variable “Switching Theory and Logic Design”

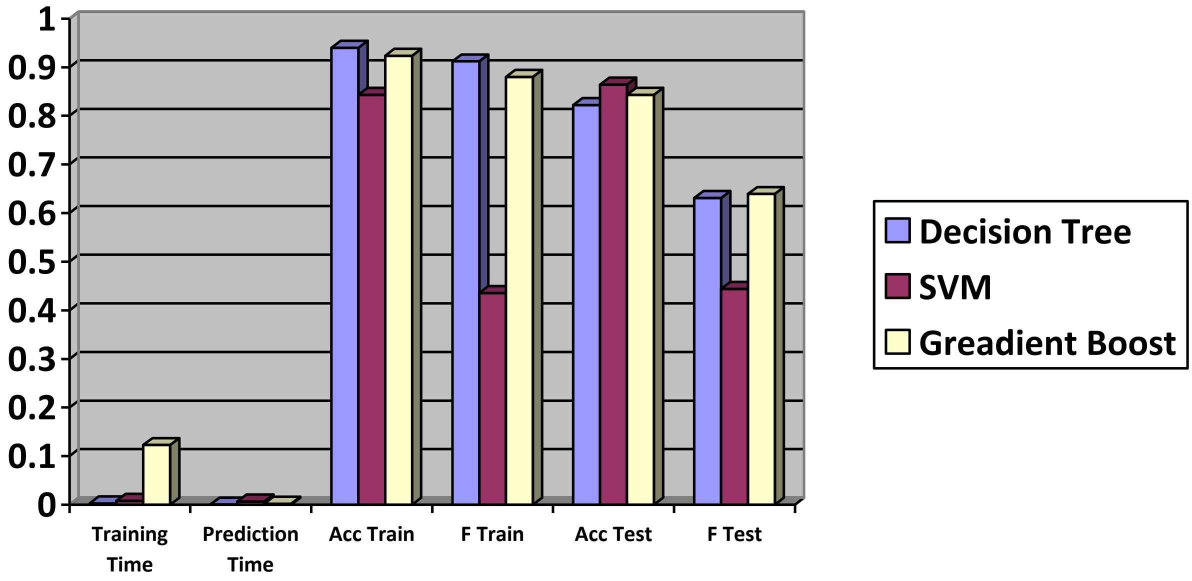


Fig3: Result for "Analog Communications"

Fig3 shows the results for target variable "Analog Communications", the Training Time of Gradient Boost classifier is slightly more compared to Decision tree and SVM classifiers. The performance of Decision tree classifier on training data and test data was good. The performance of SVM on training data and test data was not good. The performance of Gradient Boost classifier was good on training and test datasets. In this case Decision tree or Gradient Boost classifiers can be selected, but considering Test Data Accuracy(Acc Test) and F-Score Test(F Test) gradient boost algorithm was selected(considering its slightly better performance on test data)

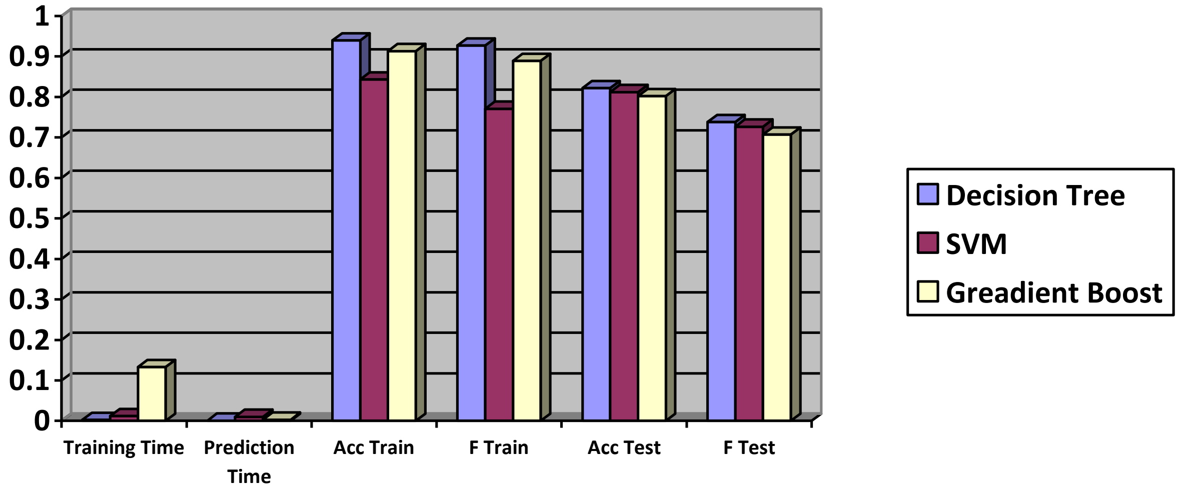


Fig4: Result for "Electronic Circuit Analysis"

Fig4 shows the result of target variable "Electronic Circuit Analysis", the Training Time of Gradient Boost classifier is slightly more compared to Decision tree and SVM classifiers. The performance of Decision tree classifier on training data and test data was good. The performance of SVM on training data was not good, but it has shown good performance on test dataset. The performance of

Gradient Boost classifier was good on training and test datasets. In this case Decision tree classifier is selected, because it shown better performance on training and test datasets compared to SVM and gradient boost classifiers.

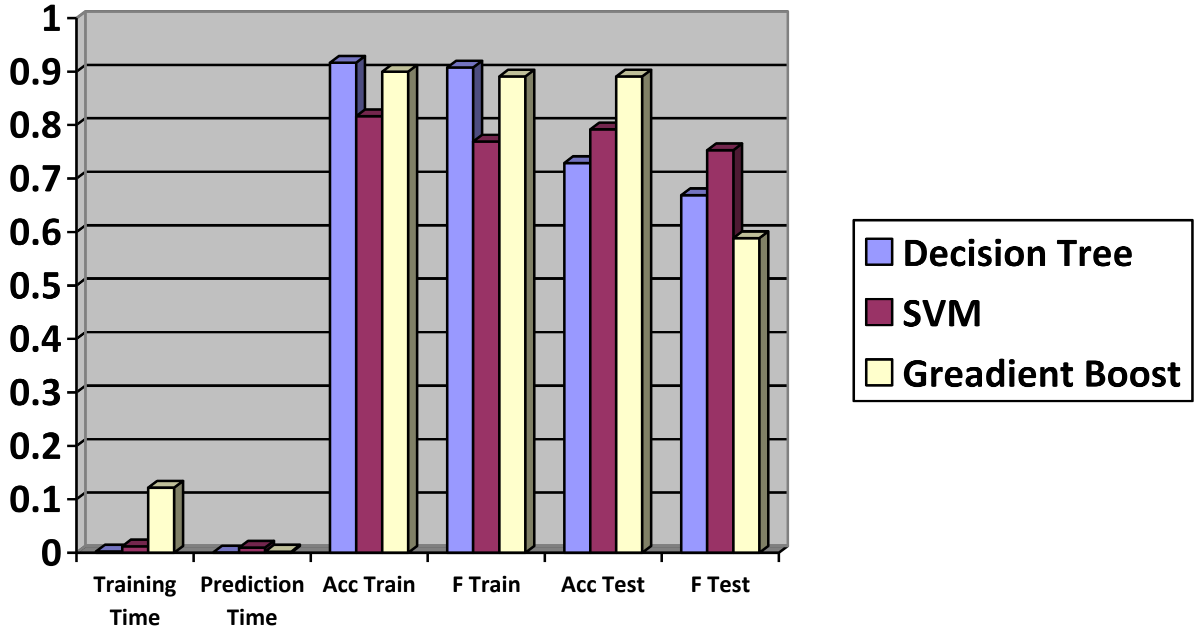


Fig5: Result for "EMWTL"

Fig5 shows the result of target variable "EMWTL", the Training Time of Gradient Boost classifier is slightly more compared to Decision tree and SVM classifiers. The performance of Decision tree classifier on training data was good and moderate performance on test data. The performance of SVM on training data was moderate and it shown good performance on test dataset. The performance of Gradient Boost classifier was good on training and bad performance on test datasets. In this case SVM classifier is selected, because it shown better performance on training and test datasets compared to Decision tree and gradient boost classifiers.

Step3: Optimization of Classifiers

Selected classifiers for output variables are optimized using GridSearchCV( ). For output variable "Switching Theory and Logic Design", SVM classifier was optimized. Parameters such as 'C', 'decision_function_shape','kernel','shrinking' are optimized. The hyper parameter 'kernel' is tuned for 'rbf','poly','linear','sigmoid', the hyper parameter 'C' is tuned for 0.2,0.5,1.0,1.5,2.0, the hyper parameter 'decision_function_shape' is tuned for 'ovo', 'ovr','None', the hyper parameter 'degree' is tuned for 2,3,4,5,6,7. The hyper parameter 'shrinking' is tuned for True and False, the hyper parameter 'probability' tuned for True and False, the hyper parameter 'coef0' tuned for 0.0,0.5,1.0.

Fig6 shows the comparison of un-optimized and optimized classifiers for “Switching Theory and Logic Design”. The accuracies of optimized and un-optimized classifiers are almost same, but there was large variation in F-Score. The F-Score of optimized SVM classifier is much better compared to un-optimized SVM classifier. The optimized SVM classifier outperformed un-optimized SVM classifier.

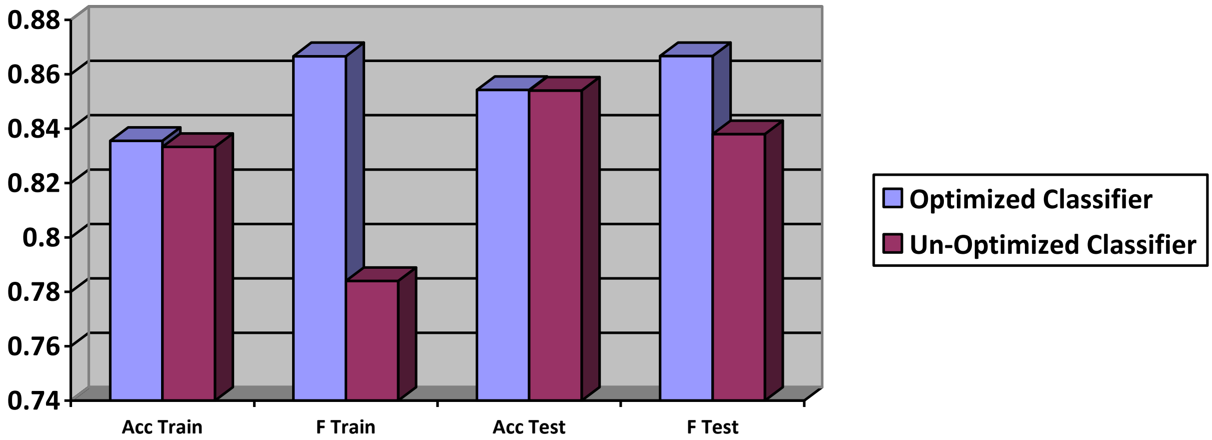


Fig6: Comparison or results of Optimized and un-Optimized SVM classifier for “Switching Theory and Logic Design”

Gradient Boost classifier was optimized for output variable “Analog Communications”. Parameters such as ‘criterion’’, ‘learning_rate’,’ max_leaf_nodes ’,’ min_samples_leaf’,’ min_samples_split’,’ n_estimators’ are optimized. The hyper parameter 'learning_rate' is tuned for 0.1,0.5,1, the hyper parameter 'n_estimators' is tuned for 50,100,150, the hyper parameter 'min_samples_leaf' is tuned for 1,3,5,7,15, the hyper parameter 'min_samples_split' is tuned for 2,10,20. The hyper parameter 'max_depth' is tuned for 1,3,5. Fig7 shows the comparison of un-optimized and optimized classifiers for “Analog Communications”. The accuracies of optimized and un-optimized classifiers are almost same, but there was large variation in F-Score. The F-Score of optimized Gradient Boost classifier is much better compared to un-optimized SVM classifier. The optimized Gradient Boost classifier outperformed un-optimized Gradient Boost classifier.

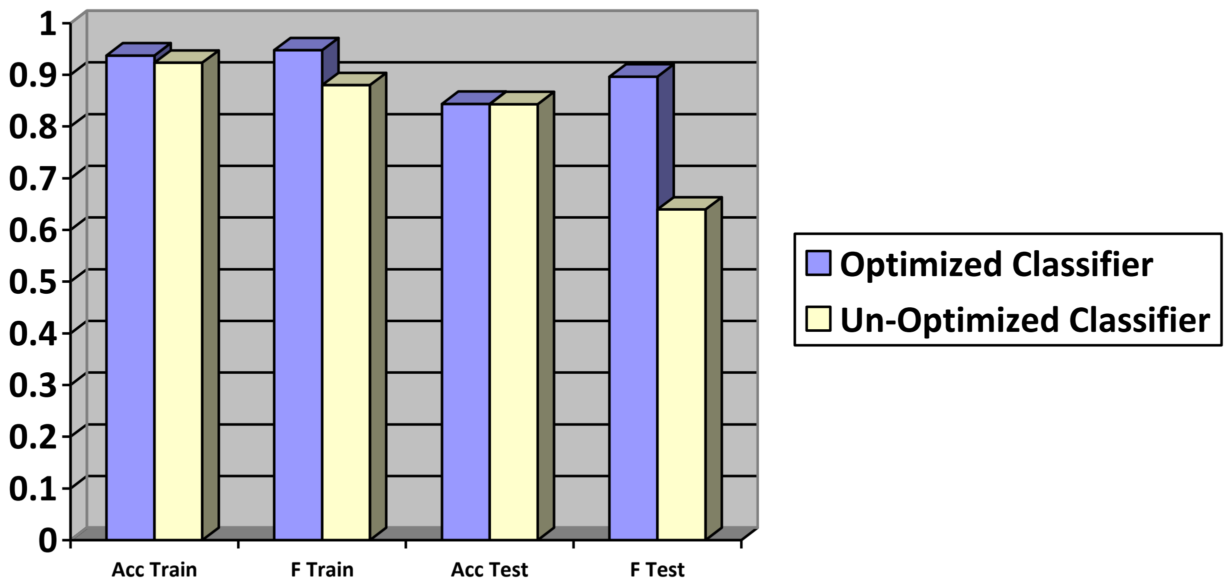


Fig7: Comparison or results of Optimized and un-Optimized Gradient Boost classifier for “Analog Communications”

Decision Tree classifier was optimized for output variable “Electronic Circuit Analysis”. Parameters such as ‘criterion’’, ‘max_depth’’,’ max_leaf_nodes ’,’ min_samples_leaf’,’ min_samples_split’ are optimized. The hyper parameter 'criterion' is tuned for 'gini','entropy', the hyper parameter 'splitter' is tuned for 'best','random', the hyper parameter 'min_samples_leaf' is tuned for 1,3,5,7,15, the hyper parameter 'min_samples_split' is tuned for 2,10,20. The hyper parameter 'max_depth' is tuned for 1,3,5.Fig8 shows the comparison of un-optimized and optimized classifiers for “Electronic Circuit Analysis”. The accuracies of training and test data ofun-optimized classifier is better compared to optimized classifier. Further F-Score on training data of optimized and un-optimized classifiers are almost same (with a small difference), but the F-Score on test data of optimized classifier is much better compared to un-optimized classifier. Keeping in mind F-Score (evaluation metric for this project), optimized classifier performed better compared to un-optimized classifier.

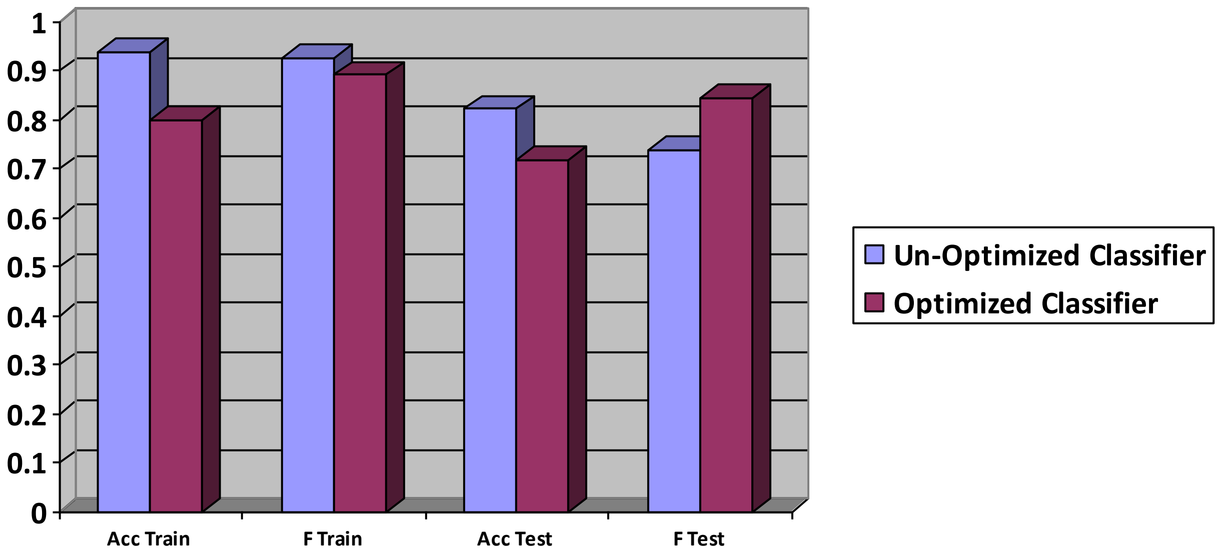


Fig8: Comparison or results of Optimized and un-Optimized Decision Tree classifier for "Electronic Circuit Analysis"

For output variable "EMWTL", SVM classifier was optimized. Parameters such as 'C', 'decision_function_shape','kernel','shrinking' are optimized. The hyper parameter 'kernel' is tuned for 'rbf','poly','linear','sigmoid', the hyper parameter 'C' is tuned for 0.2,0.5,1.0,1.5,2.0, the hyper parameter 'decision_function_shape' is tuned for 'ovo', 'ovr','None', the hyper parameter 'degree' is tuned for 2,3,4,5,6,7. The hyper parameter 'shrinking' is tuned for True and False, the hyper parameter 'probability' tuned for True and False, the hyper parameter 'coef0' tuned for 0.0,0.5,1.0

Fig9 shows the comparison of un-optimized and optimized classifiers for "EMWTL". The accuracies of optimized and un-optimized classifiers are almost same, but there was large variation in F-Score. The F-Score of optimized SVM classifier is much better compared to un-optimized SVM classifier. The optimized SVM classifier outperformed un-optimized SVM classifier.

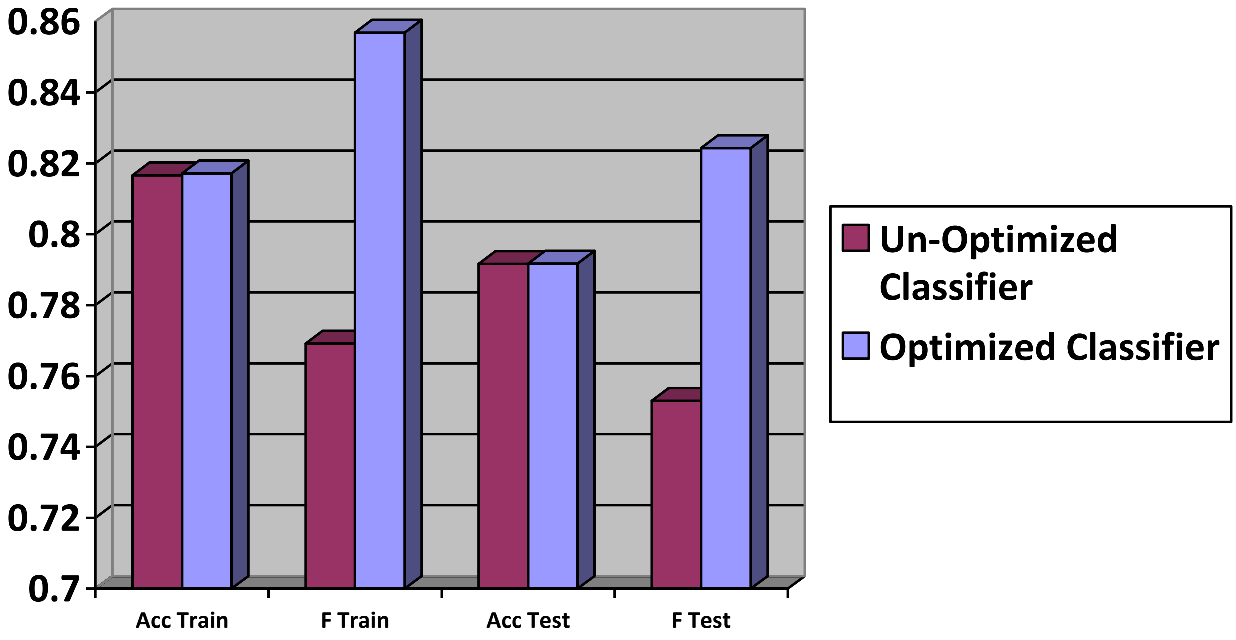


Fig9: Comparison or results of Optimized and un-Optimized SVM classifier for “EMWTL”

Step4: Feature Selection

Most predictive input features are derived form Gradient Boost model using model.feature_importances_. Fig 11 shows normalized weights for first five most predictive features of “Switching Theory and Logic Design”. . Fig11 shows normalized weights for first five most predictive features of “Analog Communications”. Fig12 shows normalized weights for first five most predictive features of “Electronic Circuit Analysis”. Fig13 shows normalized weights for first five most predictive features of “EMWTL”. These five most predictive features are used to train the train data on selected Optimized classifiers (given in step3) for the four target variables. After training these trained models are tested on test dataset. The results of predicted test data would be compared with optimized and un-optimized models given in step3 and step2 respectively.

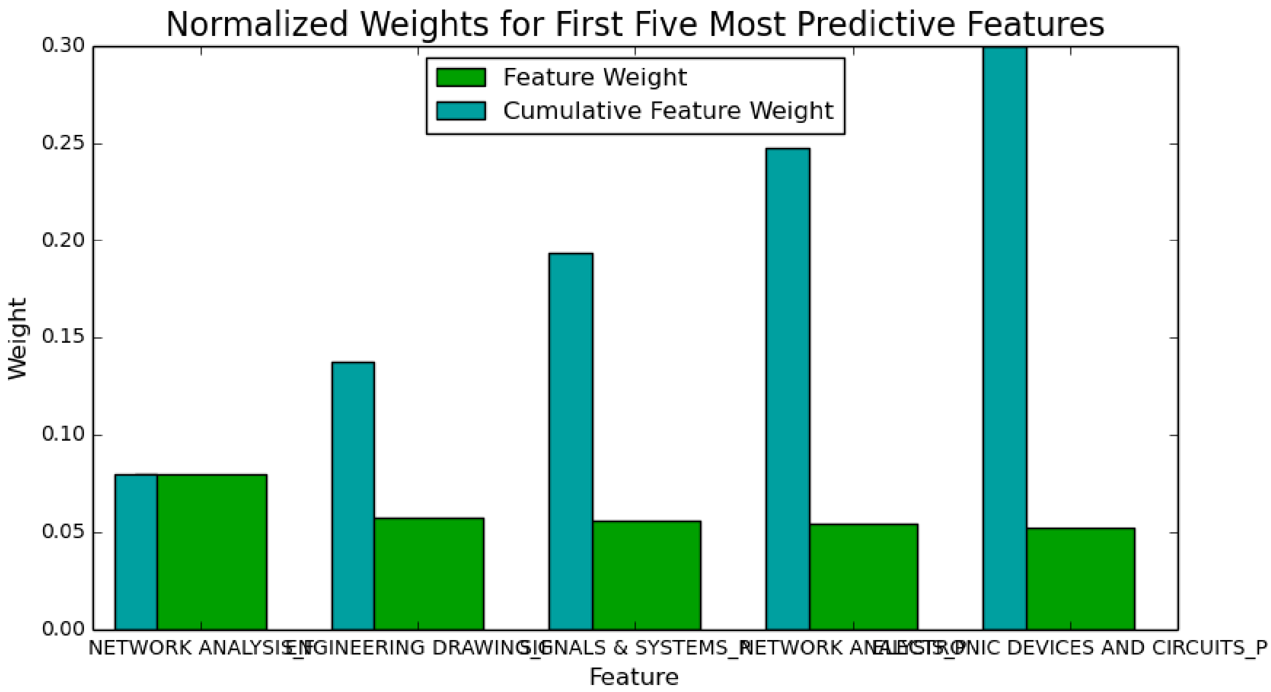


Fig10: Normalized Weights for First Five Most Predictive Features of “Switching Theory and Logic Design”

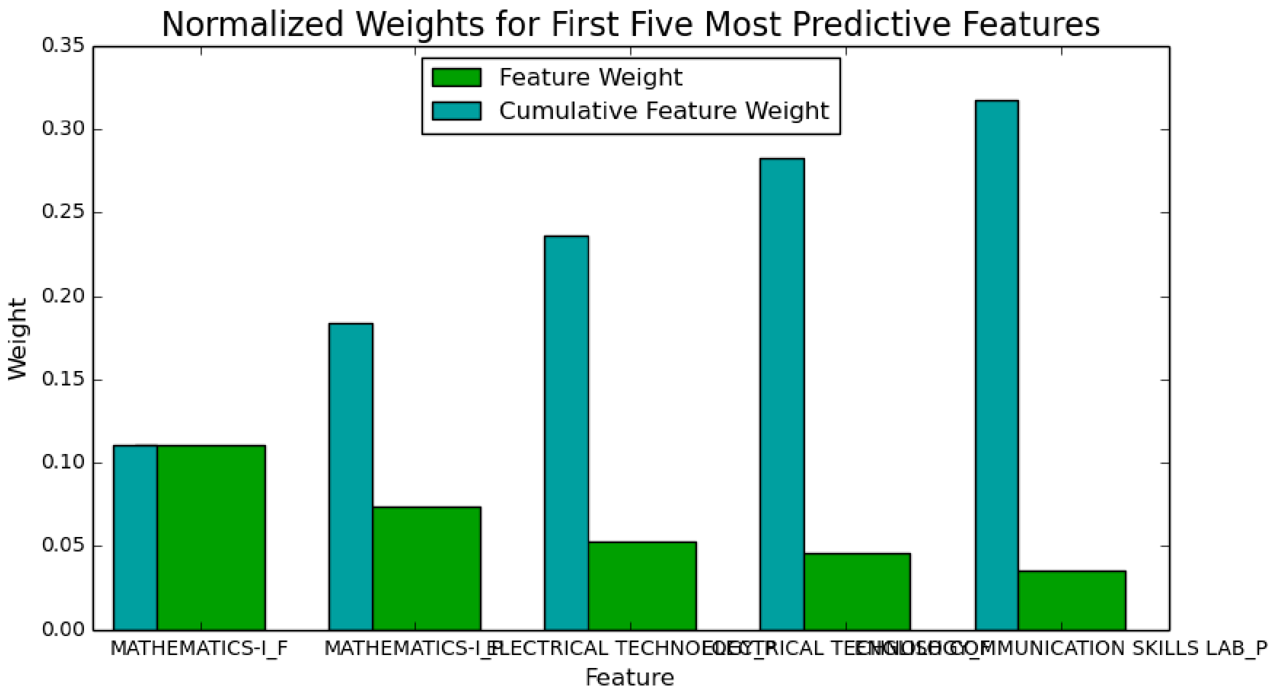


Fig11: Normalized Weights for First Five Most Predictive Features of “Analog Communications”

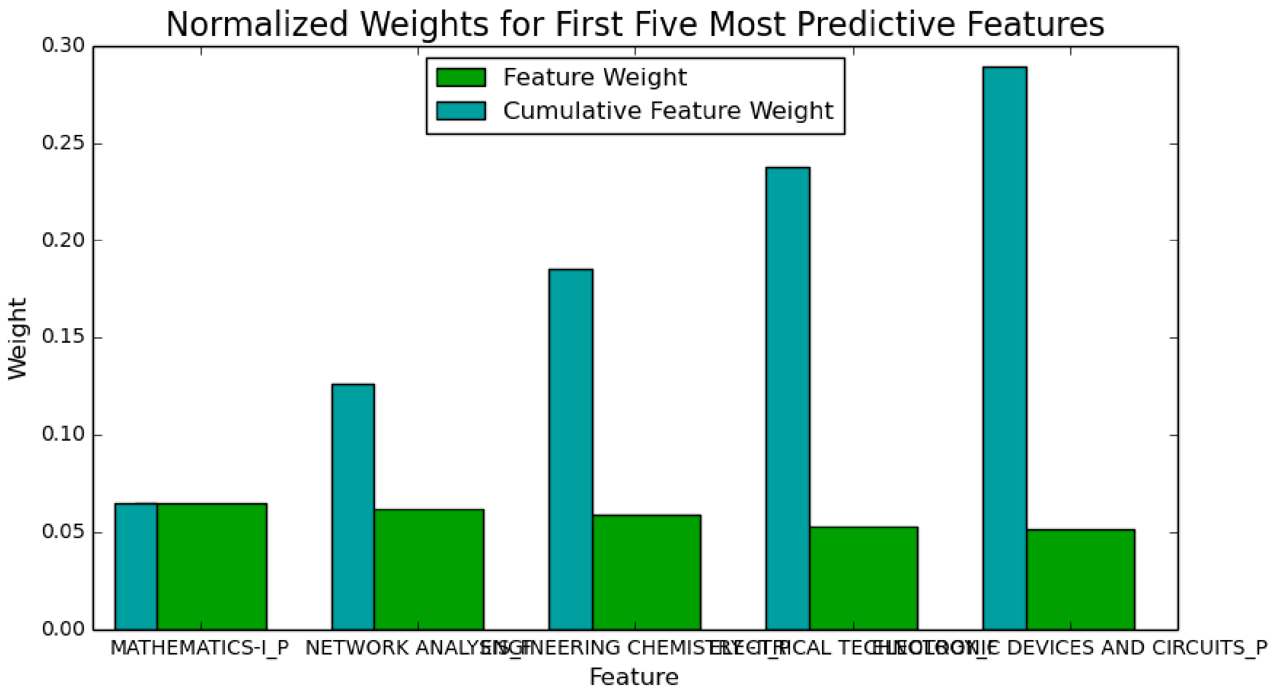


Fig12: Normalized Weights for First Five Most Predictive Features of "Electronic Circuit Analysis"

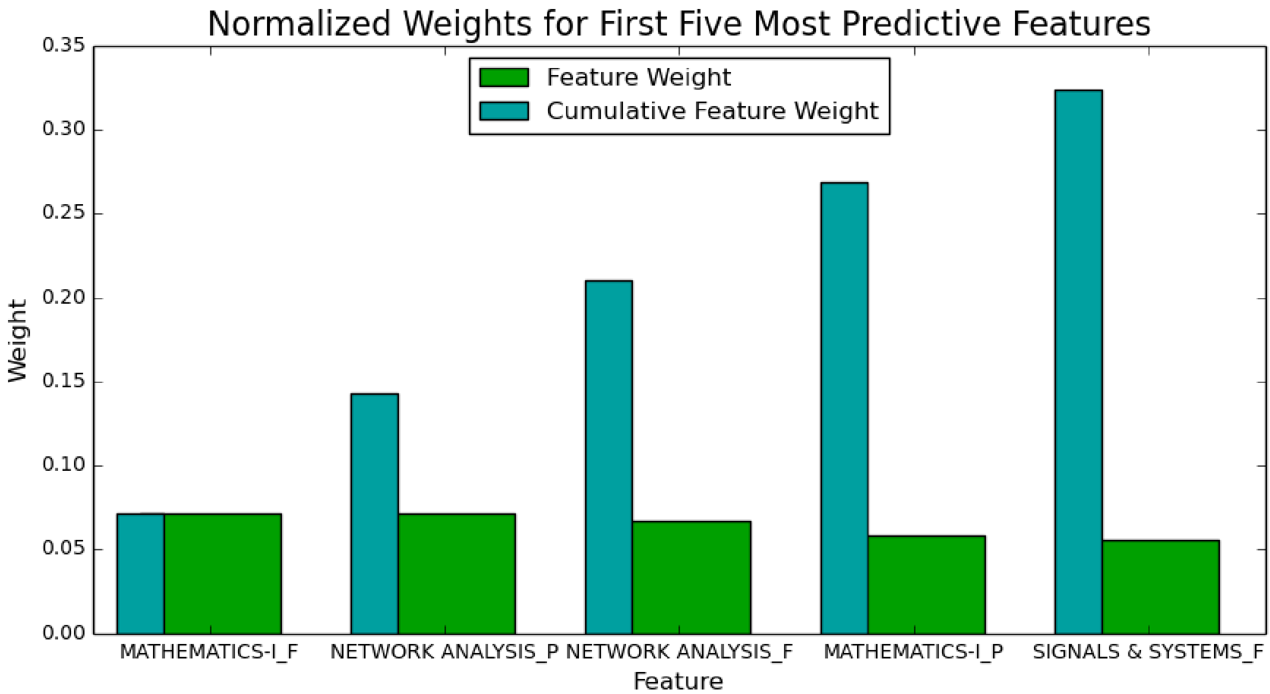


Fig13: Normalized Weights for First Five Most Predictive Features of "EMWTL"

Step5: Results (Comparison of Optimized, Un-Optimized and Feature Selected Models)

Comparison or test data prediction results of Optimized, un-Optimized and Feature Selected models for target variable “Switching Theory and Logic Design” is shown in Fig14. Accuracy of optimized and un-optimized models are much higher compared to feature selected model. Similarly F-Score of feature selected and un-optimized models are very less compared to optimized model. Optimized model shown best performance for target variable “Switching Theory and Logic Design”.

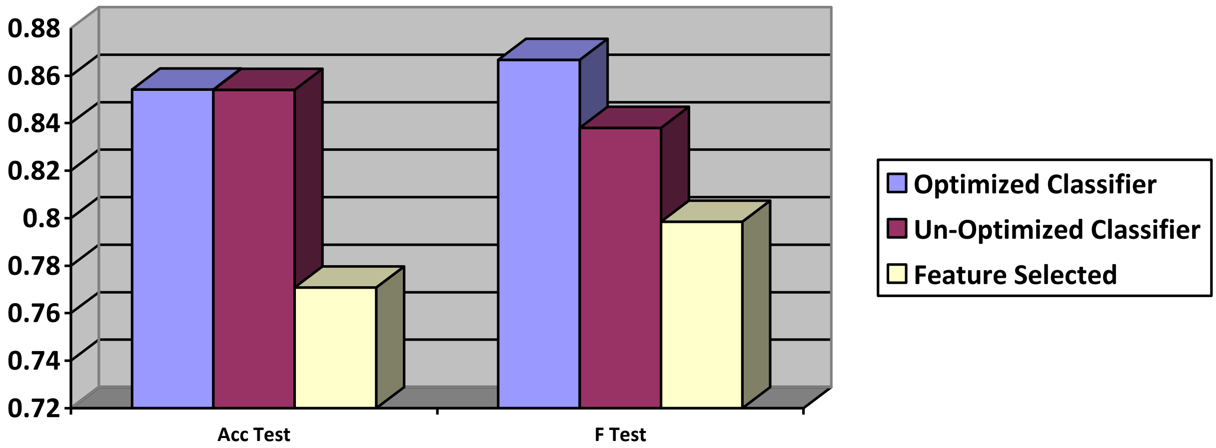


Fig14: Comparison or results of Optimized, un-Optimized and Feature Selected SVM classifiers for “Switching Theory and Logic Design”

Comparison or test data prediction results of Optimized, un-Optimized and Feature Selected models for target variable “Analog Communications” is shown in Fig15. Accuracy of optimized model is slightly higher compared to feature selected and un-optimized models. Further F-Score of un-optimized models are very less compared to optimized and feature selected models. Optimized model shown best performance for target variable “Analog Communications”.

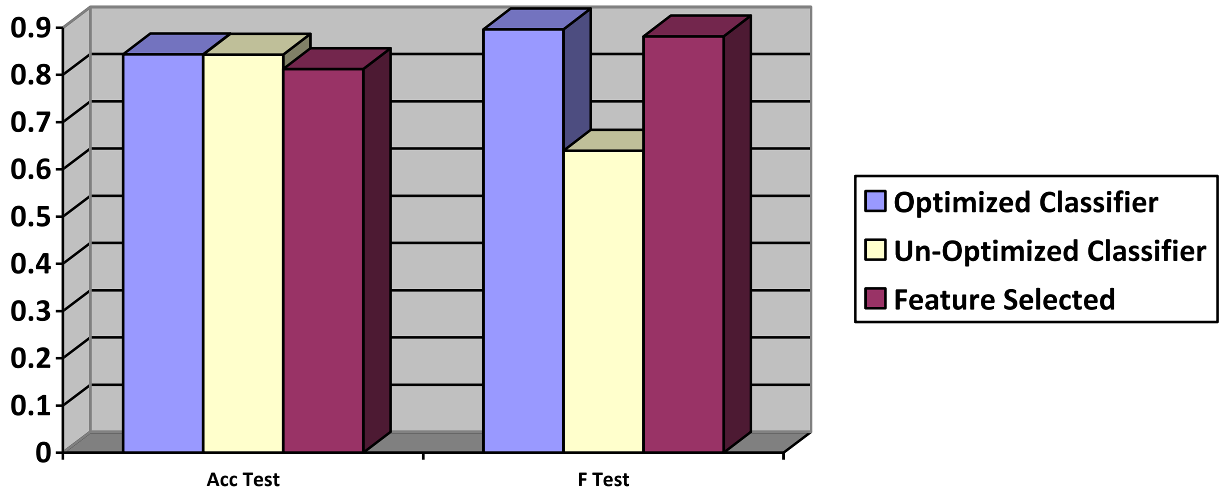


Fig15: Comparison or results of Optimized, un-Optimized and Feature Selected Gradient Boost classifiers for "Analog Communications"

Comparison or test data prediction results of Optimized, un-Optimized and Feature Selected models for target variable "Electronic Circuit Analysis" is shown in Fig16. Accuracy of un-optimized model is much higher compared to optimized and feature selected models. Similarly F-Score of feature selected and Optimized models are same and very high compared to un-optimized model. Since F-Score is the evaluation metric for this project, Optimized and Feature Selected models shown best performance for target variable "Electronic Circuit Analysis".

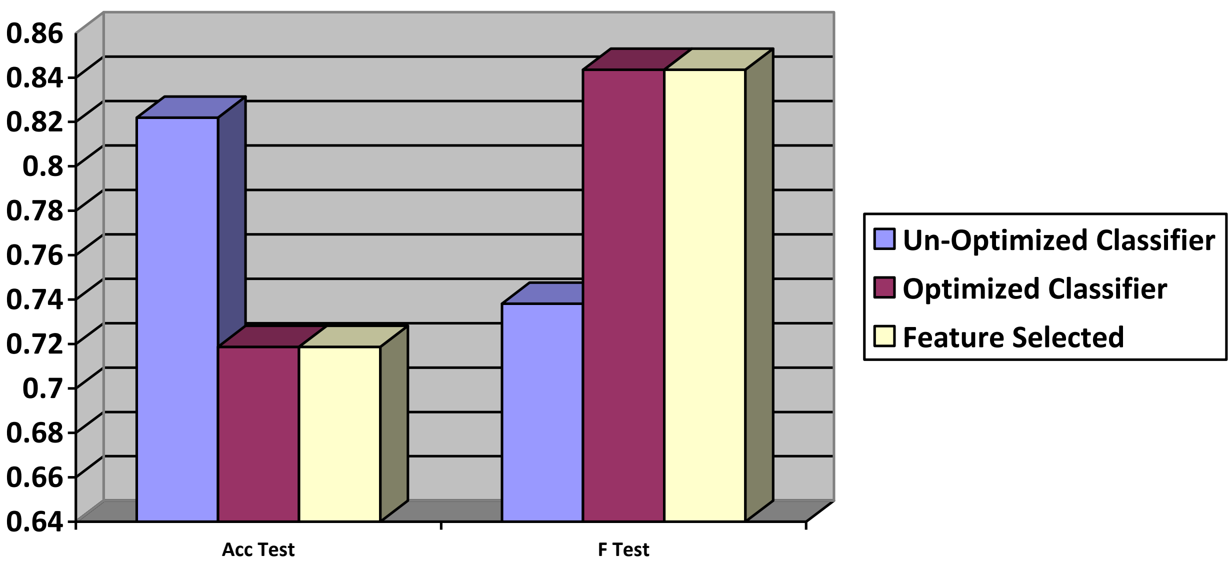


Fig16: Comparison or results of Optimized, un-Optimized and Feature Selected Decision Tree classifiers for "Electronic Circuit Analysis"

Comparison or test data prediction results of Optimized, un-Optimized and Feature Selected models for target variable “EMWTL” is shown in Fig17. Accuracy of un-optimized and Optimized models is much higher compared to feature selected models. Similarly F-Score of feature selected model is very high compared to un-optimized and Optimized models. Since F-Score is the evaluation metric for this project, Feature Selected model shown best performance for target variable “EMWTL”.

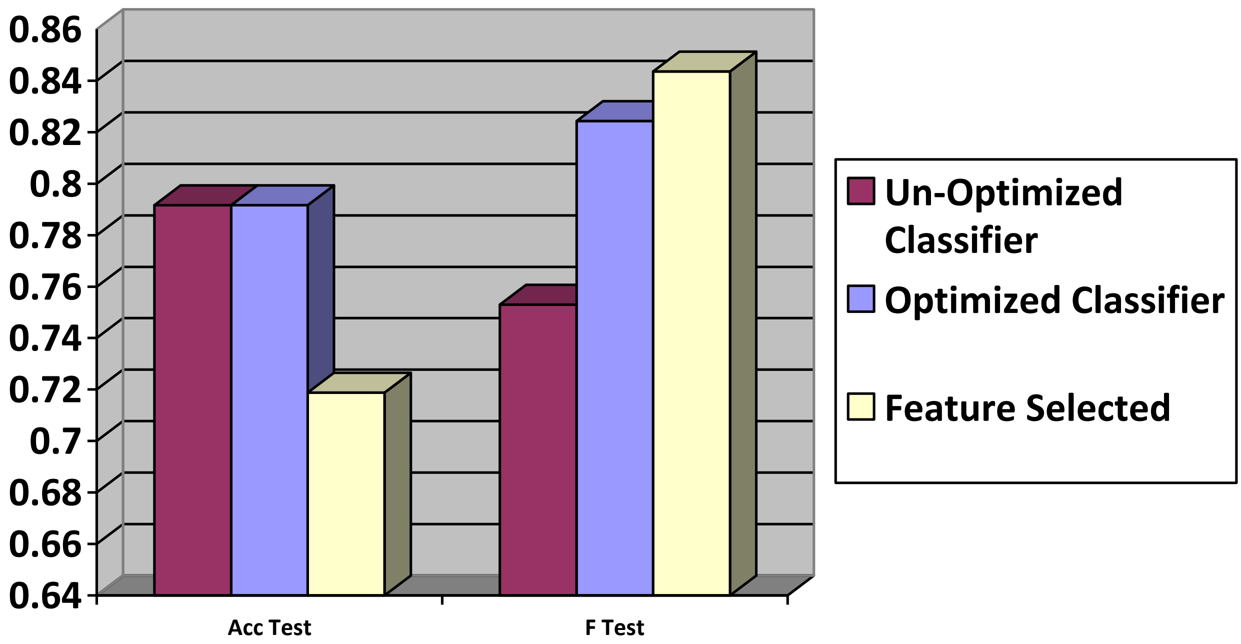


Fig17: Comparison or results of Optimized, un-Optimized and Feature Selected SVM classifiers for “EMWTL”

F-Score comparison of Benchmark, un-optimized, optimized and feature selected models for target variables “Switching Theory and Logic Design “,”Analog Communications”, “Electronic Circuit Analysis”, “EMWTL” is shown in Table1.

F-Score	Benchmark Predictor	Un-Optimized Predictor	Optimized Predictor	Feature Selected Predictor
Switching Theory and Logic Design	0.7742	0.8380	0.8667	0.7985
Analog Communications	0.8813	0.6394	0.8966	0.8817
Electronic Circuit Analysis	0.8129	0.7382	0.8436	0.8436
EMWTL	0.7760	0.5884	0.8243	0.8237

Table1: F-Score comparison of Benchmark, un-optimized, optimized and feature selected models

Conclusion

Teaching learning process would be improved by giving special attention or specialized training to

Students based on their expected results in final examinations. So identifying students based on their expected results may allow professors to conduct specialized courses to only required students. The problem of predicting students performance can be accomplished using machine learning algorithms, The dataset consists of record of results by students of 2014-2017, 2013- 2016, 2012-2015,2011-2014,2010-2013 batches in all semesters (1 to 8). The data is categorical i.e. it consists of two classes (either 'Pass' or 'Fail'). The class labels are imbalanced, so F-Score is used as evaluation metric. The data is split to train data (80%) and test data(20%). Naive model is used as benchmark predictor. Un-optimized, optimized and feature selected models are trained on train data and tested on test data for four target variables. Optimized SVM model shown best performance for target variable “Switching Theory and Logic Design”, Optimized gradient boost model shown best performance for target variable “Analog Communications”, Optimized and Feature Selected decision tree models shown best performance for target variable “Electronic Circuit Analysis”. Finally, feature selected SVM model shown best performance for target variable “EMWTL”.

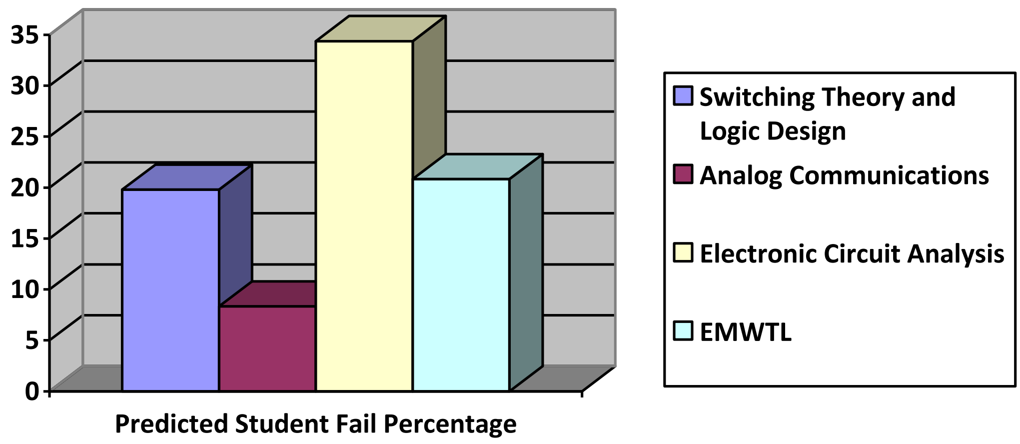


Fig18: Prediction of current batch students for four target variables

Fig18 shows the prediction of current batch students (2015-2019) for all four target variables, these prediction would help us to suggest new policies to improve student pass percentage, provide specialized training and improve teaching learning process. As per the predictions, the students predicted to fail in “Switching Theory and Logic Design” are 19.79%, the students predicted to fail in “Analog Communications” are 8.33%, the students predicted to fail in “Electronic Circuit Analysis” are 34.88%, the students predicted to fail in “EMWTL” are 20.83%. To gain more insight on data students who would pass all four subjects, students who would fail in four subjects, three subjects, two subjects and one subject are predicted and given in Fig19.

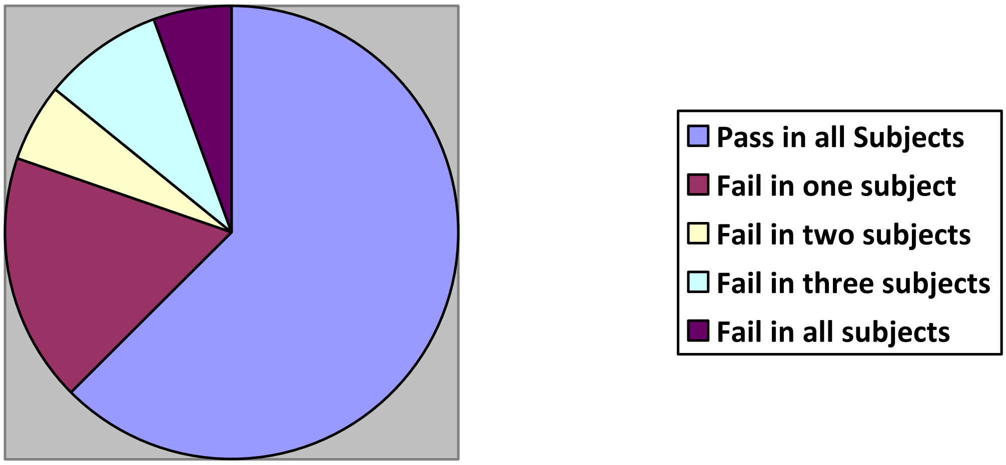


Fig19: Prediction of current batch students

It was predicted that 59.38% of students pass in all subjects, 16.67% of students would fail in one subject, 5.21% of students fail in two subjects, 8.33% of students fail in three subjects and 5.21% of students fail in all subjects. The predicted pass percentage of students was 59.38%, this percentage can easily improved by arranging tutorials for students who are predicted to fail in one subject. By arranging extra tutorials the pass percentage can be improved to (59.38+16.67) 76.05. Further placement training would be provided to 59.38% percentage of students predicted to pass all subjects. Special counselling should be provided to 5.21% of students predicted to fail in all subjects. Further training on fundamentals (fundamental subjects) would be given to students predicted to fail in two or three subjects.

I found interesting analysis which could be used to provide specialized training to students based on their predictions. The challenging part of this project was to collect and formatting the data. The model can further improved by training more data.