A Project Report

on

Student Performance Assessment and Prediction System using Machine Learning

carried out as part of the course CS1634 Submitted by

Mehil B Shah
169105105
6th Semester, B.Tech - CSE

and

Maheeka Kaistha
169105101
6th Semester, B.Tech - CSE

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

In

Computer Science & Engineering

MANIPAL UNIVERSITY
JAIPUR

Department of Computer Science & Engineering,
School of Computing and IT,
Manipal University Jaipur,
April, 2019

CERTIFICATE

This is to certify that the project entitled "**Student Performance Prediction and Assessment System**" is a bonafide work carried out as part of the course ***CS1634 Minor Project***, under my guidance by ***Mehil B Shah and Maheeka Kaistha***, student of ***B.Tech, VIth Semester*** at the Department of Computer Science & Engineering , Manipal University Jaipur, during the academic semester ***VIth***, in partial fullfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering, at MUJ, Jaipur.

Place: Manipal University, Jaipur

Date: 26th April, 2019 Signature of the Instructor (s)

DECLARATION

I hereby declare that the project entitled **"Student Performance Prediction and Assessment System**" submitted as part of the partial course requirements for the course ***CS 1634 Minor Project***, for the award of the degree of Bachelor of Technology in Computer Science & Engineering at Manipal University Jaipur during the ***B.Tech, VIth*** semester, has been carried out by me. I declare that the project has not formed the basis for the award of any degree, associate ship, fellowship or any other similar titles elsewhere.

Further, I declare that I will not share, re-submit or publish the code, idea, framework and/or any publication that may arise out of this work for academic or profit purposes without obtaining the prior written consent of the Course Faculty Mentor and Course Instructor.

Signature of the Student:

Place:

Date:

Abstract

This project is aimed at developing models to predict student's performance based on estimating the proportion of grades scored from their psychographic and lifestyle attributes. It uses various Machine Learning and Deep Learning Techniques to predict the performance of students, and basic exploratory data Analysis to derive various correlations of student's performance with psychographic attributes.

The Machine Learning Techniques used are : Decision Tree, SVM, Random Forest and Logistic Regression, AdaBoost, Gradient Boosting and XGBoost Classifier and the Deep Learning Techniques used are : Artificial Neural Networks and Recurrent Neural Networks.

The data needed for the project is collected from the UCI Machine Learning Repository.

Performances of the models were measured using the coefficient of Correlation R, Cross Validation Score, Precision, Recall and F1-Score and the Mean Confidence Interval.

Index

1. Introduction

1. Performance evaluations for students are an integral part to a student's individual and professional development.
2. Although performance evaluations are not mandatory, they serve as an important tool that assists students in further developing their skills by highlighting their strengths and constructively identifying areas for improvement.
3. The performance review process serves as the foundation that allows teachers to discuss students' contributions towards the achievement of goals and objectives, ask for feedback, make suggestions, and possibly reward a student for exceeding expectations.
4. The time spent in discussion with the students about his/her performance signals the student that you are personally invested in his/her development.

1.1 Motivation

1. Student Performance Evaluation systems usually use academic records of students' ongoing degrees, but we can use the past academic records of students' to assess their performance i.e. using their 10^{th} and 12^{th} marks to predict their performance in their respective degrees.
2. We can use other attributes related to the students' extracurricular activities, communication skills, social media activities and their family information to evaluate and predict performance.
3. After this analysis, we can determine suitable career fields for the students.
4. It will be helpful to students as well as teachers for academic performance evaluation instead of the classical evaluation approaches.
5. It can also act as a warning system for the students to improve their study performance.

2. Literature Review

- Pauziah Mohd Arsad, Norlida Buniyamin and Jamalul-lail Ab Manan. "A Neural Network Students' Performance Prediction Model (NNSPPM)" used the method of Artificial Neural Network for the prediction of academic performance of students. The cumulative grade points is used as the measuring criterion. The first semester result of students is taken as the input predictor variable (Independent variable) and eighth semester grade points are taken as the output variable (Dependent variable). Performances of the models were measured using the coefficient of Correlation R and Mean Square Error (MSE). The outcomes from the study showed that fundamental subjects at semester one and three have strong influence in the final CGPA.

- Midhun Mohan M G, Siju K Augustin and Dr. Kumari Roshni V S "A BigData Approach for Classification and Prediction of Student Result Using MapReduce" mainly uses Learning Analytics and Predictive Analytics for the overall prediction of the students' performance over a huge volume of data. They started with data collection and data preprocessing phase after collecting data from the CBSE schools, using MySQL server. The data pre-processing, data cleaning, data transformation etc. are done by using apache HIVE framework. MapReduce algorithm in the Hadoop framework is used to retrieve all the informative data followed by the Predictive analytics part where the actual predictions are made using the multiple linear regression model.

- Madhav S. Vyas and Reshma Gulwani. "Predicting Student's Performance using CART approach in Data Science" uses a decision tree model for predicting the academic performance of students. The data collection and data preprocessing is performed where the continuous values are converted to discrete values and the null values are eliminated. Then by using CART algorithm to the data, the decision tree prediction model was built and the students with poor performance are predicted out.

- Huda Al-Shehri, Amani Al-Qarni, Leena Al-Saati, Arwa Batoaq, Haifa Badukhen, Saleh Alrashed, Jamal Alhiyafi and Sunday O. Olatunji. "Student Performance Prediction Using Support Vector Machine and K-Nearest Neighbor" used the dataset provided by the University of Minho in Portugal, which relate to the performance in math subject and it consists of 395 data samples. Most of earlier work on the same dataset used K-Nearest Neighbor algorithm and achieved low results. To ensure better comparison, both Support Vector Machine algorithm and KNearest Neighbour algorithm were applied on the dataset to predict the student's grade. Empirical studies outcome indicated that Support Vector Machine achieved slightly better results with correlation coefficient of 0.96, while the K-Nearest Neighbor achieved correlation coefficient of 0.95.

2.1 Outcome of Literature Review

- After the Literature Review phase, we found out that very few projects have taken in consideration the psychographic and emotional attributes of the student, these attributes are very important in determining the performance of the student.
- So, we decided to take these factors in the consideration, and we got the dataset from the UCI Machine Learning Repository.
- We also found that the projects had only 1 or 2 models for the performance prediction, so we decided to implement more number of models, as this will help us in building a proper solution for the problem statement.
- We also decided to do assessment of student's performance and deriving correlations and trends between various attributes and performance.

2.2 Problem Statement

- After going through the Literature Review Phase, and writing down the outcomes of the Literature Review, we decided to frame the Problem Statement : **"Student Performance Prediction and Assessment System using Machine Learning Techniques"**
- We finalized on the workflow of the project
    - Dataset Acquisition
    - Data Cleaning and Preprocessing
    - Exploratory Data Analysis
    - Implementing Machine Learning Models
    - Implementing Deep Learning Models
    - Saving the best model and applying that on another dataset, to check for validity of the solution delivered.

3. Methodology and Framework

3.1 Dataset Description

Our dataset is of secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school related features) and it was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por). The two datasets were modelled under binary/three-level classification and regression tasks.

1. school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
2. sex - student's sex (binary: 'F' - female or 'M' - male)
3. age - student's age (numeric: from 15 to 22)
4. address - student's home address type (binary: 'U' - urban or 'R' - rural)
5. famsize - family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
6. Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
7. Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
8. Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
9. Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
10. Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
11. reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
12. guardian - student's guardian (nominal: 'mother', 'father' or 'other')
13. traveltime - home to school travel time (numeric: 1 - less than 15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - greater than 1 hour)
14. studytime - weekly study time (numeric: 1 - less than 2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - greater than 10 hours)
15. failures - number of past class failures (numeric: n if n is between 1 and 3, else 4)
16. schoolsup - extra educational support (binary: yes or no)
17. famsup - family educational support (binary: yes or no)
18. paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
19. activities - extra-curricular activities (binary: yes or no)
20. nursery - attended nursery school (binary: yes or no)
21. higher - wants to take higher education (binary: yes or no)
22. internet - Internet access at home (binary: yes or no)
23. romantic - with a romantic relationship (binary: yes or no)
24. famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
25. freetime - free time after school (numeric: from 1 - very low to 5 - very high)

26. goout - going out with friends (numeric: from 1 - very low to 5 - very high)
27. Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
28. Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
29. health - current health status (numeric: from 1 - very bad to 5 - very good)
30. absences - number of school absences (numeric: from 0 to 93)
31. G1 - first period grade (numeric: from 0 to 20)
32. G2 - second period grade (numeric: from 0 to 20)
33. G3 - final grade (numeric: from 0 to 20, output target)

3.2 Algorithms and Techniques Implemented

We implemented four classification Machine Learning Models and further three Machine Learning Boosting Algorithms on our dataset

3.2.1 Classification Models

1. Decision Tree Model

Decision Trees are a type of Supervised Machine Learning Algorithm where the data is continuously split according to a certain parameter. There are two main types of decision trees, Classification trees (with Yes/No types) and Regression trees (continuous data types).

The tree can be explained by two entities, namely Decision Nodes and Leaves. The Leaves are the decisions or the final outcomes and the Decision Nodes are where the data is split.

Using this algorithm, we have to identify which attributes we need to consider as the root node at each level, this is called attributes selection. There are different attributes selection measures like Information Gain and Gini Index. Information gain is used for categorical attributes, to estimate the information contained by each attribute.

2. SVM Model

"Support Vector Machine" or SVM is a supervised machine learning algorithm which is mostly used in classification problems. The objective of this algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. We need to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes.

Hyperplanes are decision boundaries that help classify the data points; data points falling on either side of the hyperplane can be attributed to different classes. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these, we maximize the margin of the classifier and deleting these

support vectors will change the position of the hyperplane. These are the points that help us in building an SVM model.

However, it doesn't perform well when the dataset is large because the required training time is higher and when the data set has more noise i.e. target classes are overlapping. SVM doesn't directly provide probability estimates, these are calculated using five-fold cross-validation.

3. **Random Forest Model**

Random Forest is a flexible, easy to use machine learning algorithm that produces a great result most of the time and is one of the most used algorithms, because of it's simplicity and the fact that it can be used for both classification and regression tasks.

The forest is an ensemble of Decision Trees, most of the time trained with the "bagging" method, according to which, a combination of learning models increases the overall result. Random Forest, instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features which results in a wider diversity that generally results in a better model.

Overfitting is not a problem faced by this algorithm as most of the time this won't happen that easily to a random forest classifier because if there are enough trees in the forest, the classifier won't overfit the model.

However, a large number of trees can make the algorithm to slow and ineffective for real-time predictions. A more accurate prediction requires more trees, which results in a slower model.

4. **Logistic Regression Model**

Logistic regression is the most famous machine learning algorithm after linear regression and is used for classification tasks. It is named after the function it uses, the logistic function also called the sigmoid function. This function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

Logistic regression can be classified as binomial (target variable can have only 2 possible types), multinomial (target variable can have 3 or more possible types which are not ordered) or ordinal (deals with target variables with ordered categories).

However, it tends to underperform when there are multiple or non-linear decision boundaries as they are not flexible enough to naturally capture more complex relationships. It is highly reliable on a proper presentation of your data and is not a useful tool unless all the important independent variables have already been identified. It is also an Algorithm that is known for its vulnerability to overfitting.

3.2.2 Boosting Algorithms

After applying the basic classification models, we applied Boosting Algorithms to improve the model predictions. The idea of boosting is to train weak learners sequentially, each trying to correct its predecessor, and become strong learners.

These attempt to create a better model, by learning from errors of previous model. A strong classifier is created by combining weak classifiers, and then building a second model on the top of it, so it can learn from the errors of first model.

1. AdaBoost Classifier

Adaptive Boosting or ***"AdaBoost"*** used for Binary Classification, combines multiple weak learners into a single strong learner. The weak learners are decision trees with a single split, called ***decision stumps***. When the first decision stump is created, all observations are weighted equally. To correct the previous error, the observations that were incorrectly classified now carry more weight than the observations that were correctly classified. AdaBoost algorithms can be used for both classification as well as regression problem.

A drawback of AdaBoost is that it can't handle noisy data and that the efficiency of the algorithm is highly affected by outliers as the algorithm tries to fit every point perfectly, but there is no proof that this algorithm overfits.

2. XGBoost Classifier

Extreme Gradient Boosting or ***"XGBoost"*** is an implementation of gradient boosted decision trees which is faster and gives better performance. It uses hyper parameters which can be used to fine tune the performance.

It cuts off Gradient Boosting when it reaches a certain level, because Gradient Boosting takes a lot of time and XGBoost tends to reduce the computation time by cutting off Gradient Boosting at a certain satisfactory level.

3. Gradient Boosting Classifier

Gradient Boosting also works similarly to AdaBoost, but instead of changing the weights for every incorrect classified observation at every iteration, it tries to fit the new predictor to the residual errors made by the previous predictor. This algorithm uses Gradient Descent to find the shortcomings in the previous learner's predictions.

Gradient boosting machines are generally very slow in implementation because of sequential model training, so they are not very scalable.

3.2.3 Classifying Data using various Parameters

Further, classification of data was done by calculating the Precision, Recall, F-1 score and Support values. Finally, the mean confidence interval was also calculated.

- Precision-Recall is a useful measure of success of prediction when the classes are very imbalanced. In information retrieval, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

- Precision (***P***) is defined as the number of true positives (T_P) over the number of true positives plus the number of false positives (F_P).

$$P = T_P / (T_P + F_P)$$

- Recall (***R***) is defined as the number of true positives (T_P) over the number of true positives plus the number of false negatives (F_N).

$$R = T_P / (T_P + F_N)$$

- (***F1***) score is defined as the harmonic mean of precision and recall.

$$F1 = 2 \times [(P \times R) / (P + R)]$$

- Precision and Recall are often in tension. Improving precision typically reduces recall and vice versa.

Based on these values, a confusion matrix can be plotted for each model. This helps in describing the performance of a classification model (or classifier).

Mean Confidence Interval

A confidence interval often referred to as margin of error, is a bounds on the estimate of a population parameter like mean and is used to quantify the uncertainty on an estimate. The basic format for calculating error is given as: ***error +/- const * sqrt((error*(1 - error) /n)***

The narrower the confidence interval, the more precise the estimate is.

3.2.4 Implementing Neural Networks

Neural networks are a set of algorithms that work similar to the human brain, and are designed to recognize patterns. They interpret sensory data in numerical format contained in

vectors, into which all real-world data, must be translated through a kind of machine perception, labelling or clustering raw input.

Neural networks act as a clustering and classification layer on top of the data. They help to group unlabelled data according to similarities among the example inputs, and they classify data when they have a labelled dataset to train on. Deep neural networks or ***"Deep Learning"*** can basically be thought of as a component of larger machine-learning applications involving algorithms for reinforcement learning, classification and regression.

Deep learning is used for "stacked neural networks", that is, networks composed of several layers made of nodes. Each layer of nodes trains on a distinct set of features based on the previous layer's output and more and more complex the features can be recognized, since they aggregate and recombine features from the previous layers.

Keras is a high-level neural networks API which runs seamlessly on CPU and GPU and focuses on enabling fast experimentation. It supports both convolutional networks and recurrent networks, as well as combinations of the two and allows easy and fast prototyping. Keras models are trained on Numpy arrays of input data and labels.

The Sequential model in Keras is a linear stack of layers. The first layer in a Sequential model needs to receive information about its input shape and the following layers can do automatic shape inference. It uses the ***"fit"*** function for training a model for a given number of ***epochs*** (iterations on a dataset). Some 2D and 3D layers, such as ***"Dense"***, support the specification of their input shape via the argument ***"input_dim"***.

4. Work Done

4.1 Working Process

Data Collection

We collected the datasets from various sources, after studying the datasets, we finalized on the dataset, which we obtained from UCI, which contained 33 parameters, and around 1000 records.

Data Preprocessing

Since the data was already clean, we didn't have to do any preprocessing on data, and then, we began with the analysis of the data, where we generated various graphs and derived many correlations.

Data Preparation

In order to prepare the data for Machine Learning processes, we had to prepare the data. In this process, we label encoded the variables, dropped the target variable and split the data into training/test sets.

Machine Learning

We used Decision Tree, Random Forest, Support Vector Classification and Logistic Regression Models to our data. We applied various methods to ensure maximum accuracy. We also applied various boosting algorithms like Gradient Boosting, AdaBoost, and XGBoost to improve accuracy, the highest accuracy recorded was 93.8%

Deep Learning

We used Artificial Neural Networks and Recurrent Neural Networks techniques to the dataset, and optimized various parameters to improve the accuracy, but due to the lack of training data, the maximum accuracy recorded was 88.8%.

4.2 Results and Conclusion

4.2.1 Correlation Matrix and its Conclusions

From the correlation matrix, we can make certain observations:

1. The grades of future have a very high correlation with the previous grades.
2. Grades have a strong positive correlation with mother’s education, study time and desire to receive higher education.
3. Grades have a strong negative correlation with alcohol consumption, past failures and going out.
4. There is a strong positive correlation between Mother’s Education and Father’s Education
5. There is a strong positive correlation between Daily Alcohol Consumption, Weekend Alcohol Consumption and Going Out.

These correlations help us setup a base for future trend predictions, and it also helps us in feature selection.

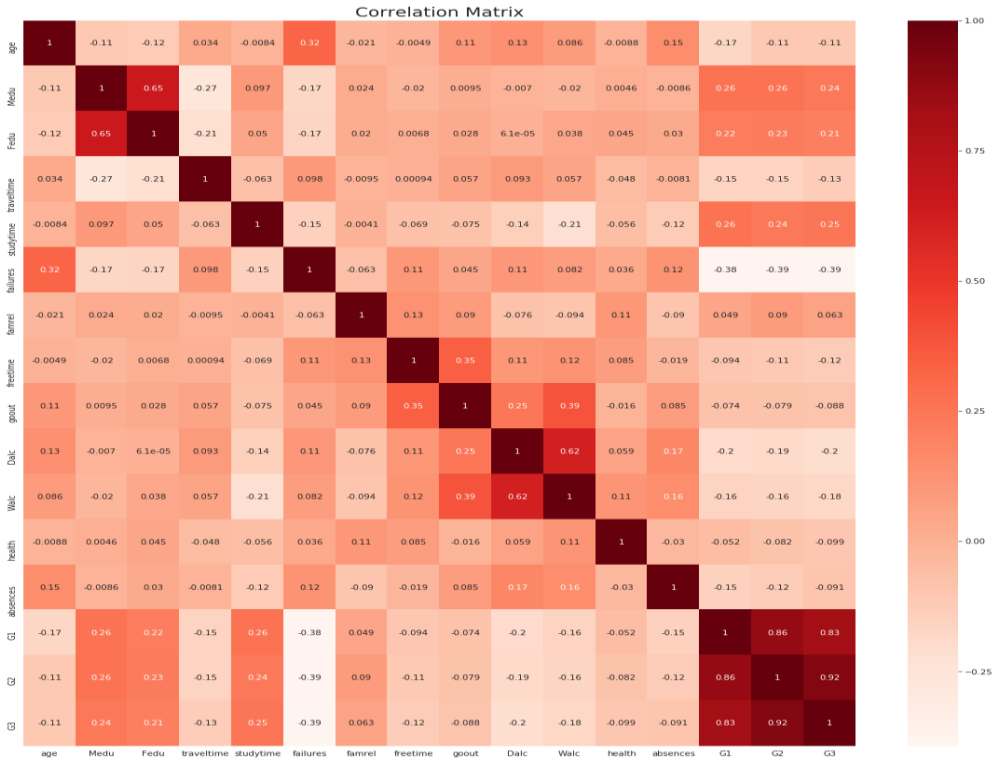

Figure-1: Correlation Matrix

4.2.2 Descriptive Analysis and Related Graphs

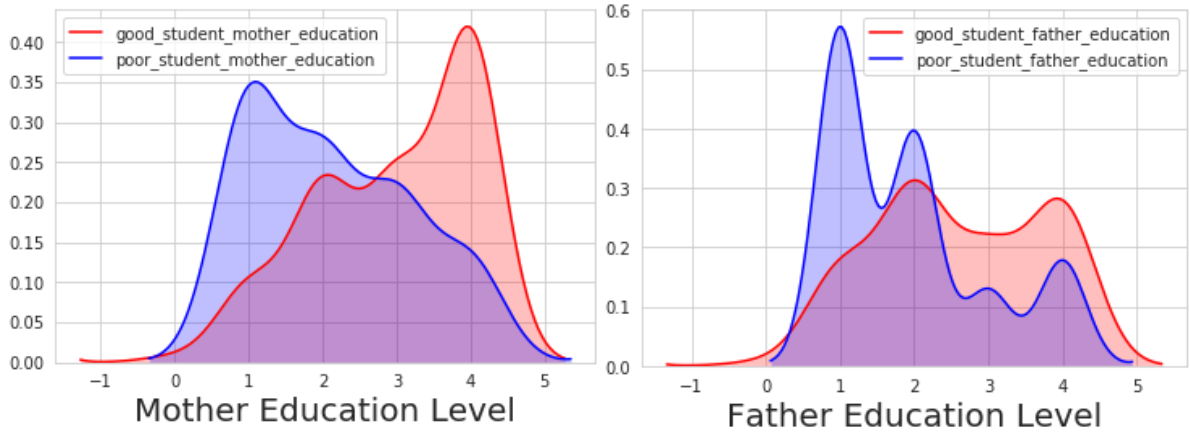

Figure - 2: Correlation with Education Level of Parents

From the given two Kernel Density Estimation graphs it can be inferred that the level of education of a student's mother plays a significant role in the student's academic performance, whereas the level of education of the father is comparatively less significant.

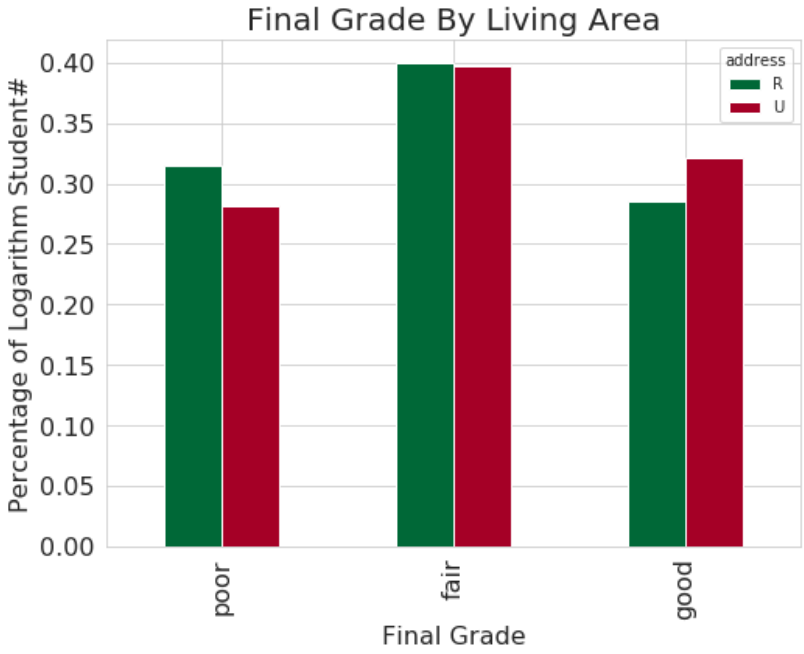

Figure - 3: Correlation with Living Area

From the given bar graph, it is noticed that there is not much significant difference between students living in rural and urban areas. A small difference can be seen under the poor grade and good grade category where students living in rural area perform poorly compared to students living in urban households

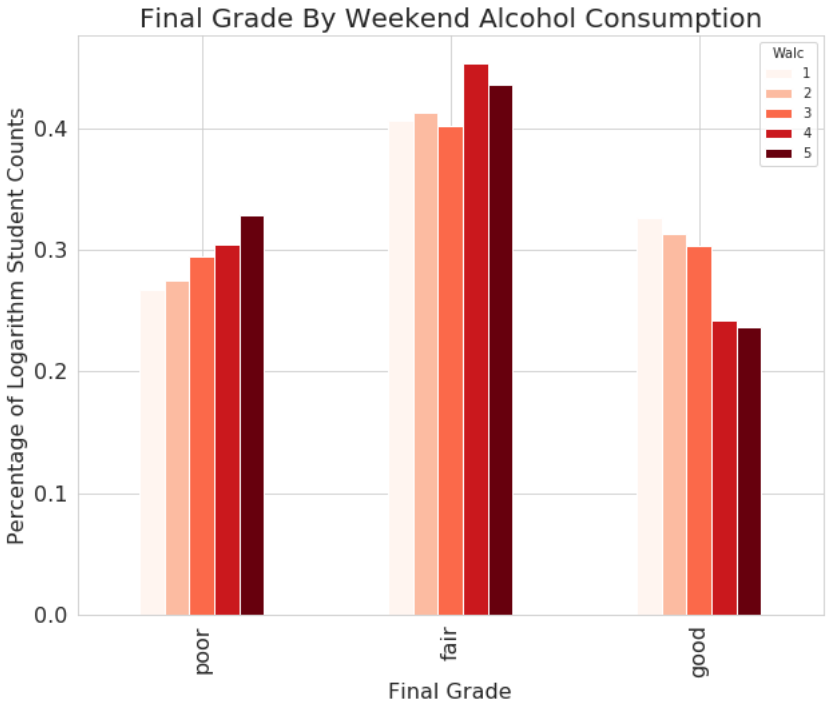

Figure - 4: Correlation with Alcohol Consumption

The above grouped histogram of Final Grade based on Weekend Alcohol Consumption indicates that, majority of students drinking more on the weekend score a Fair grade in the Final Exam.

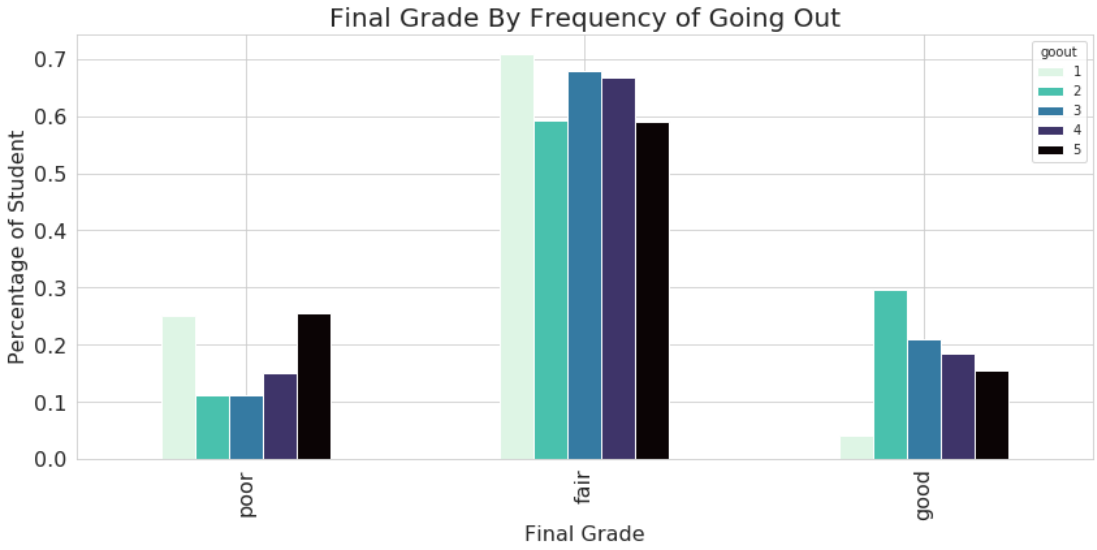

Figure - 5: Correlation with Frequency of Going Out

The above grouped histogram of Final Grade based on Frequency of weekly outings indicates that, students who go out regularly for social interactions perform fairly compared to students that don't. From this, it can be inferred that weekly outings should be an optimal amount based on the student's learning and focussing abilities (preferably limited to 2-3 a week).

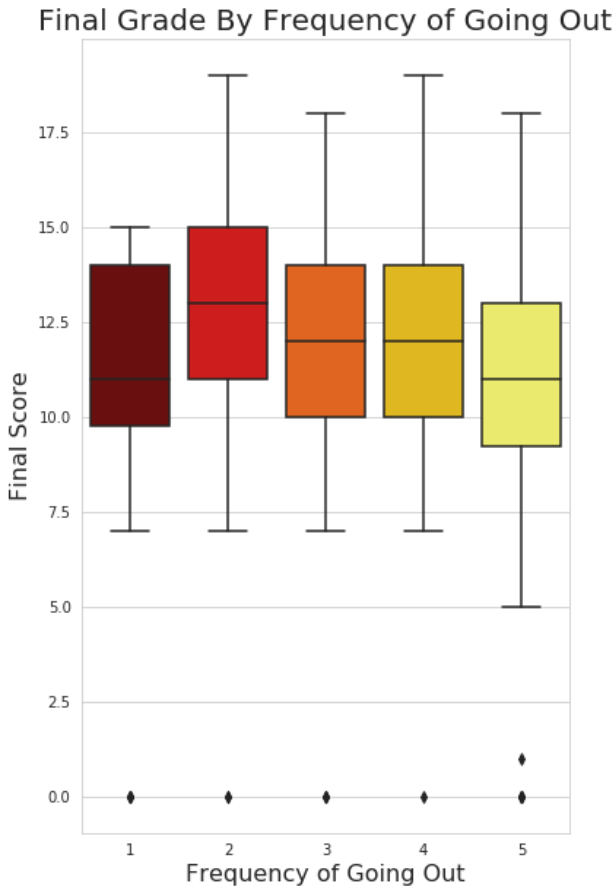

Figure - 6: Box Plot depicting the same as Figure - 5

The above plot depicting Final Grade by the Frequency of going out is a Box and Whisker plot. A Box and Whisker plot depicts groups of numerical data based on their quartiles and also indicates degree of dispersion, skewness and shows the outliers of the data. From this plot, it can be inferred that the students whose frequency of going out is 2 have the maximum median final score. Students going out 3-4 times also perform well but the students who go out only once or never have the least final grade. This indicates that it is also important for a student to enjoy co-curricular activities and have a healthy, social, outgoing life to be able to perform well as this helps the student to stay refreshed and helps them deal with stress.

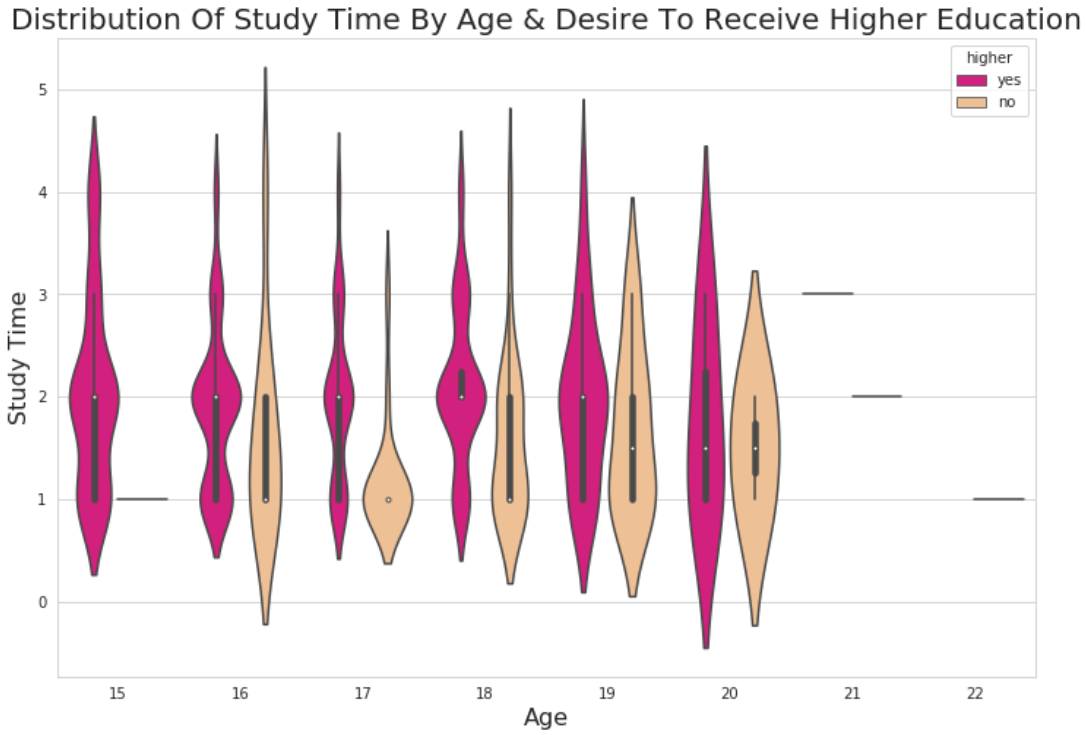

Figure - 7: Violin Plot showing the Correlation with Study Time and Age

Given above is a Violin Plot showing Distribution of Study Time with Age and Desire to Receive Higher Education. A Violin plot is a plot showing numeric data along with the probability density distribution at different values. This graph indicates that the students in age range of 18-19 are more inclined towards higher education and their study time is more compared to others. After the age of 19, it is observed that the median of study time reduces as less number of students are interested in higher education.

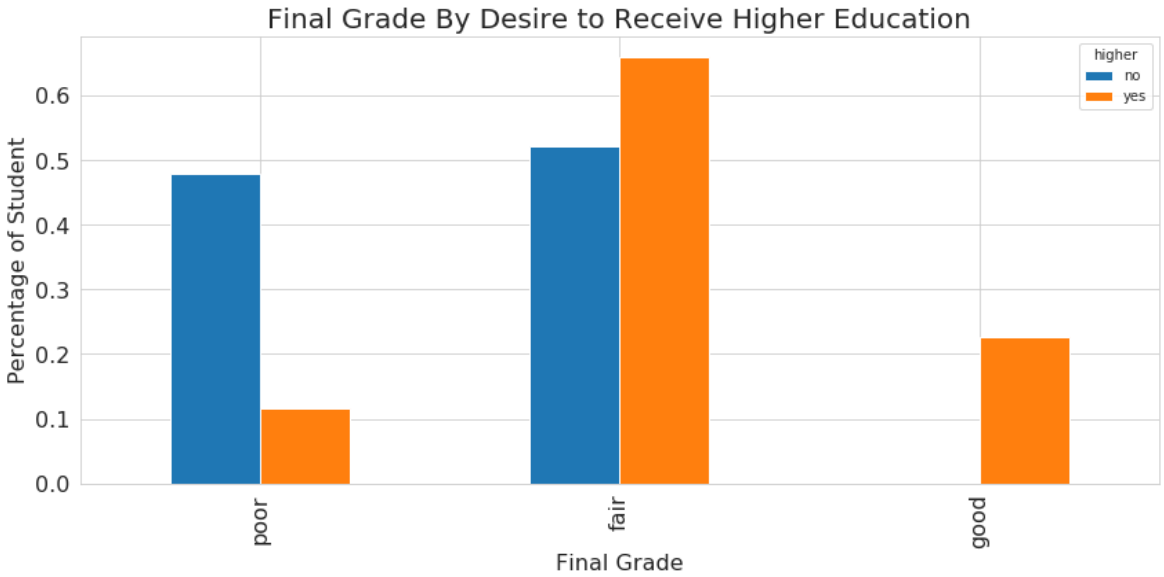

Figure - 8: Depicting the same fact as Figure 7

From the above grouped histogram of Final Grade based on Desire to Receive Higher Education, it can be observed that the students who are aiming at further higher studies perform better as compared to students who are not because they are putting in greater efforts and devoting more time to studies.

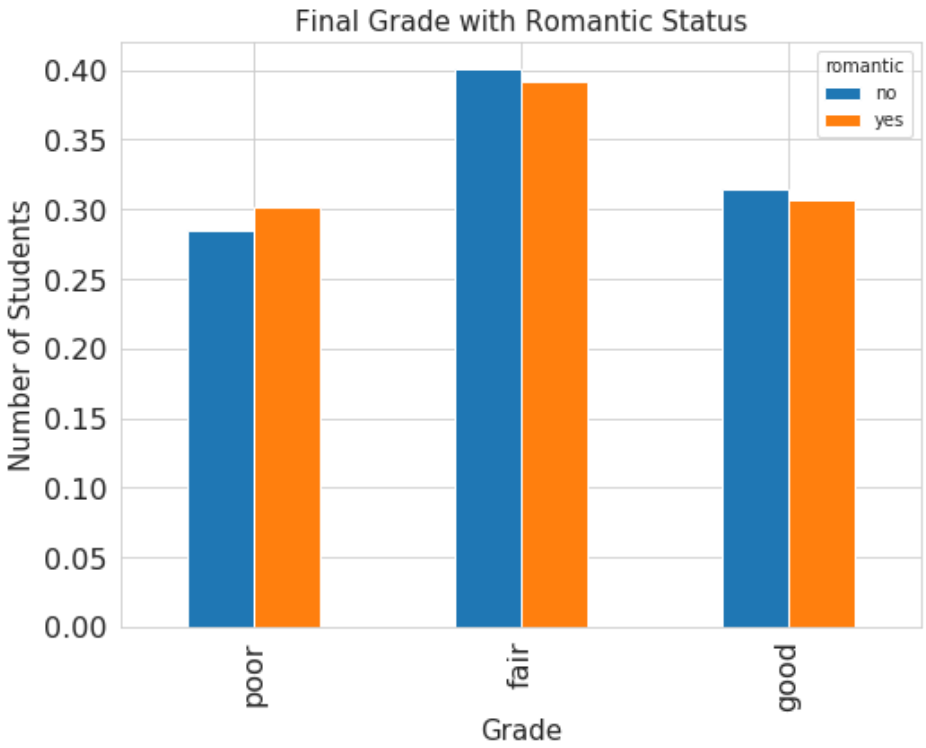

Figure - 9: Correlation with Romantic Status

The above grouped histogram indicates that majority number of students involved in a romantic relationship perform fairly. Whereas the number of students scoring poor and good

grades is approximately the same. There is not much difference observed in a particular given grade category of students who are involved in a relationship and those who are not.

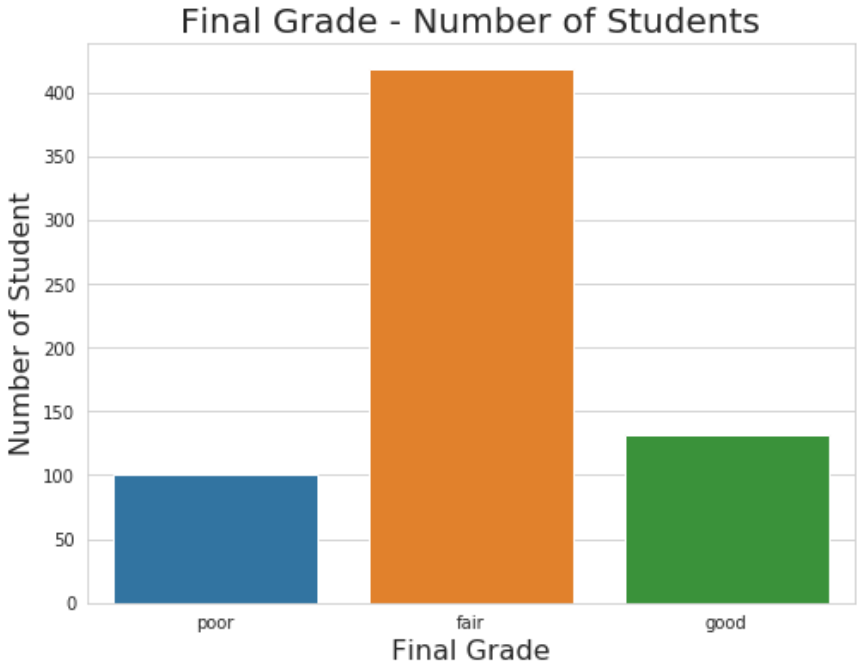

Figure - 10: Grade Distribution

From the above histogram, it is observed that majority of students score a Fair grade.

4.3 Results of Classification Machine Learning Models

Model	Model Score	Cross Validation Score	Correct Predictions	Incorrect Predictions	Training/ Test Split
Decision Tree	93.3%	88.2%	172	23	0.8/0.2
SVM	94.05%	89.8%	175	20	0.8/0.2
Random Forest	96.6%	91.7%	179	16	0.8/0.2
Logistic Regression	89.8%	59.89%	116	79	0.8/0.2

1. Decision Tree Model

Training Error: 8.14%
Test Error: 10.76%

Classifier	Precision	Recall	F1-Score	Support
Class 0	0.93	0.89	0.91	122
Class 1	0.91	0.96	0.93	45
Class 2	0.71	0.79	0.75	28
Micro Avg	0.89	0.89	0.89	195
Macro Avg	0.85	0.88	0.86	195
Weighted Avg	0.90	0.89	0.89	195

Confusion Matrix:

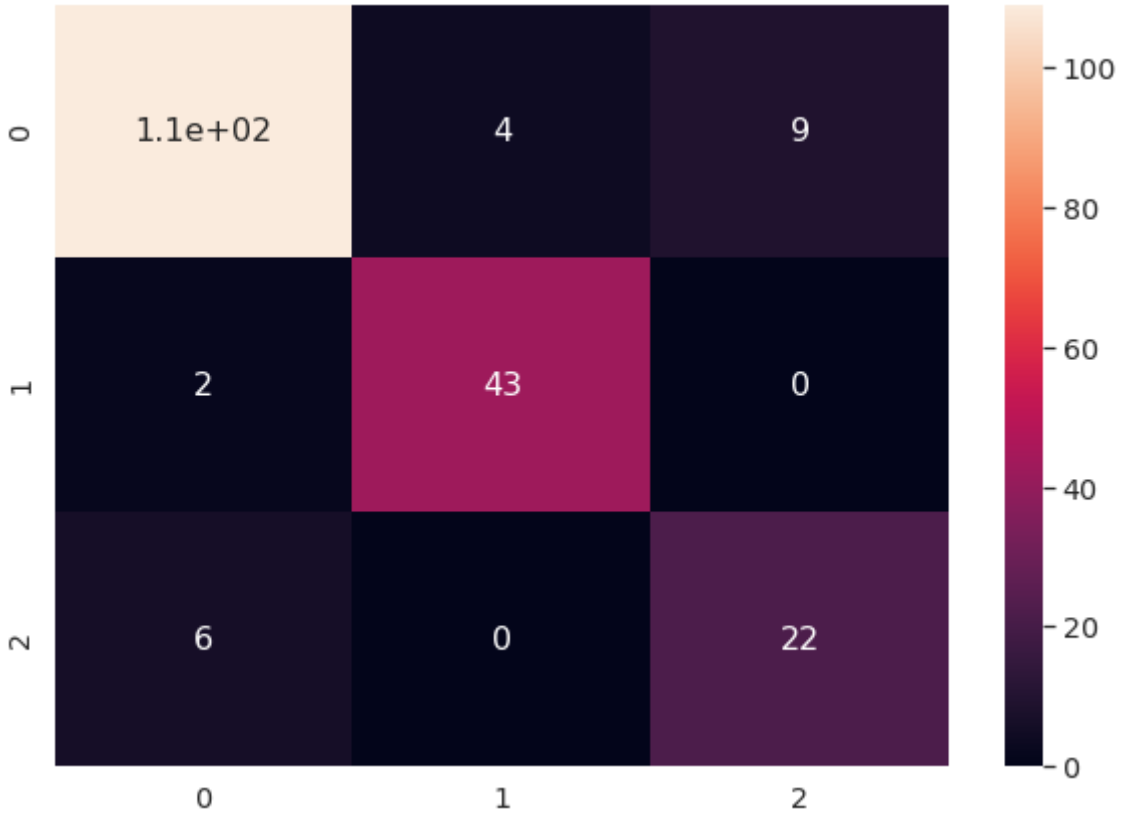

Figure 11: Confusion Matrix of Decision Tree Model

2. SVM Model

Training Error: 4.62%
Test Error: 11.79%

Classifier	Precision	Recall	F1-Score	Support
Class 0	0.89	0.93	0.91	122
Class 1	0.90	0.82	0.86	45
Class 2	0.81	0.75	0.78	28
Micro Avg	0.88	0.88	0.88	195
Macro Avg	0.87	0.84	0.85	195
Weighted Avg	0.88	0.88	0.88	195

Confusion Matrix:

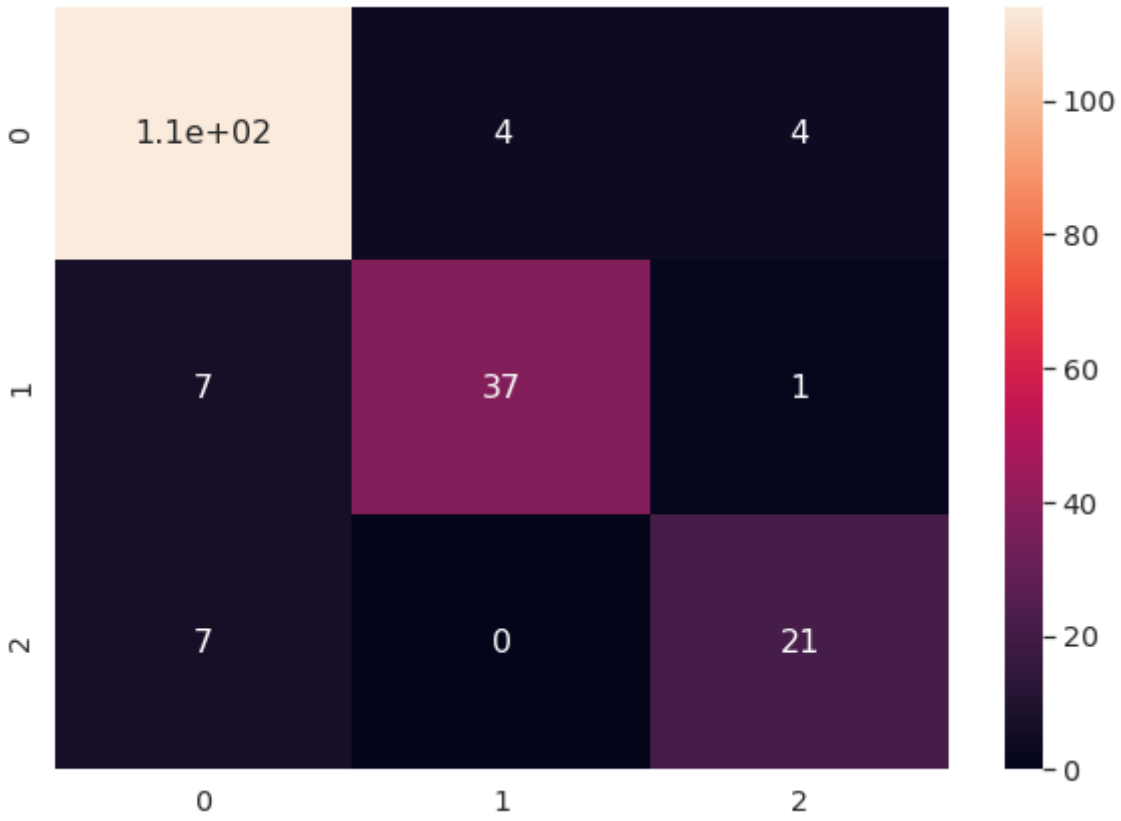

Figure 12: Confusion Matrix of SVM Model

3. Random Forest Model

Training Error: 2.42%
Test Error: 11.79%

Classifier	Precision	Recall	F1-Score	Support
Class 0	0.87	0.96	0.91	122
Class 1	0.95	0.87	0.91	45
Class 2	0.84	0.57	0.68	28
Micro Avg	0.88	0.88	0.88	195
Macro Avg	0.89	0.80	0.83	195
Weighted Avg	0.88	0.88	0.88	195

Confusion Matrix:

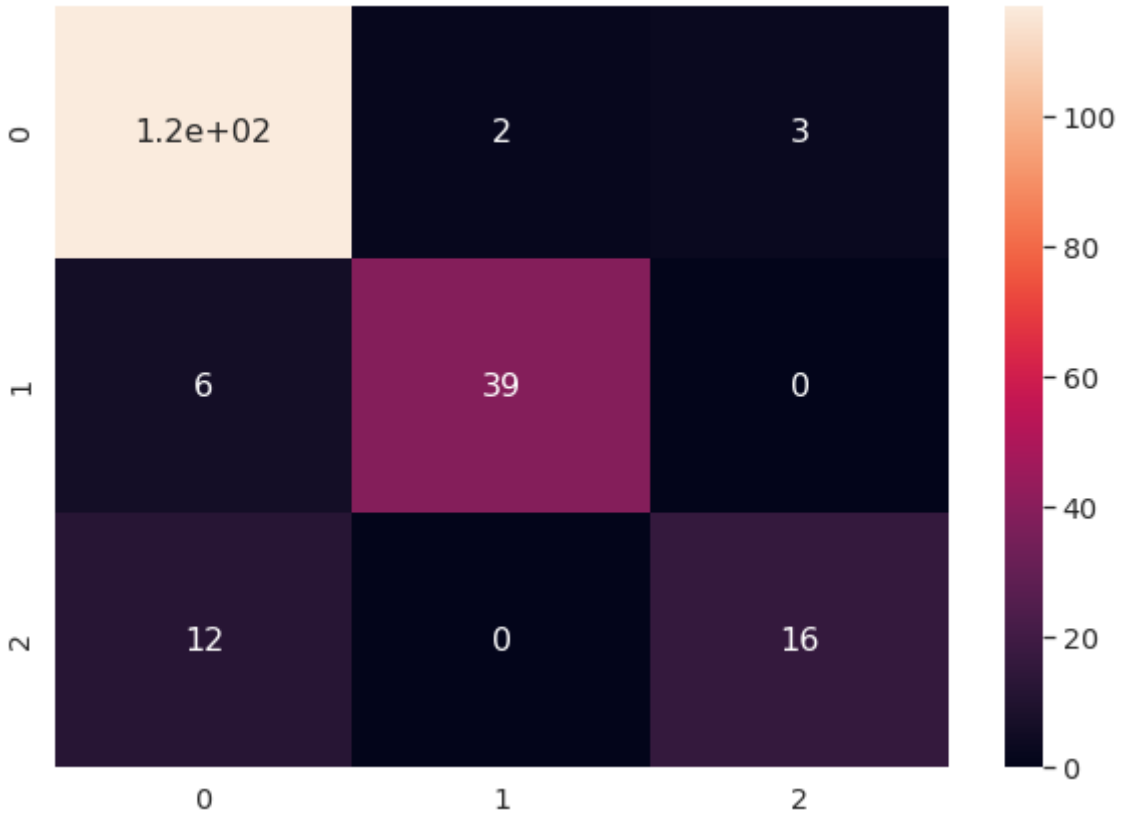

Figure 13: Confusion Matrix of Random Forest Model

4. Logistic Regression Model

Training Error: 8.81%
Test Error: 85.64%

Classifier	Precision	Recall	F1-Score	Support
Class 0	0	0	0	122
Class 1	0	0	0	45
Class 2	0.14	1.0	0.25	28
Micro Avg	0.14	0.14	0.14	195
Macro Avg	0.5	0.33	0.8	195
Weighted Avg	0.2	0.14	0.4	195

Confusion Matrix:

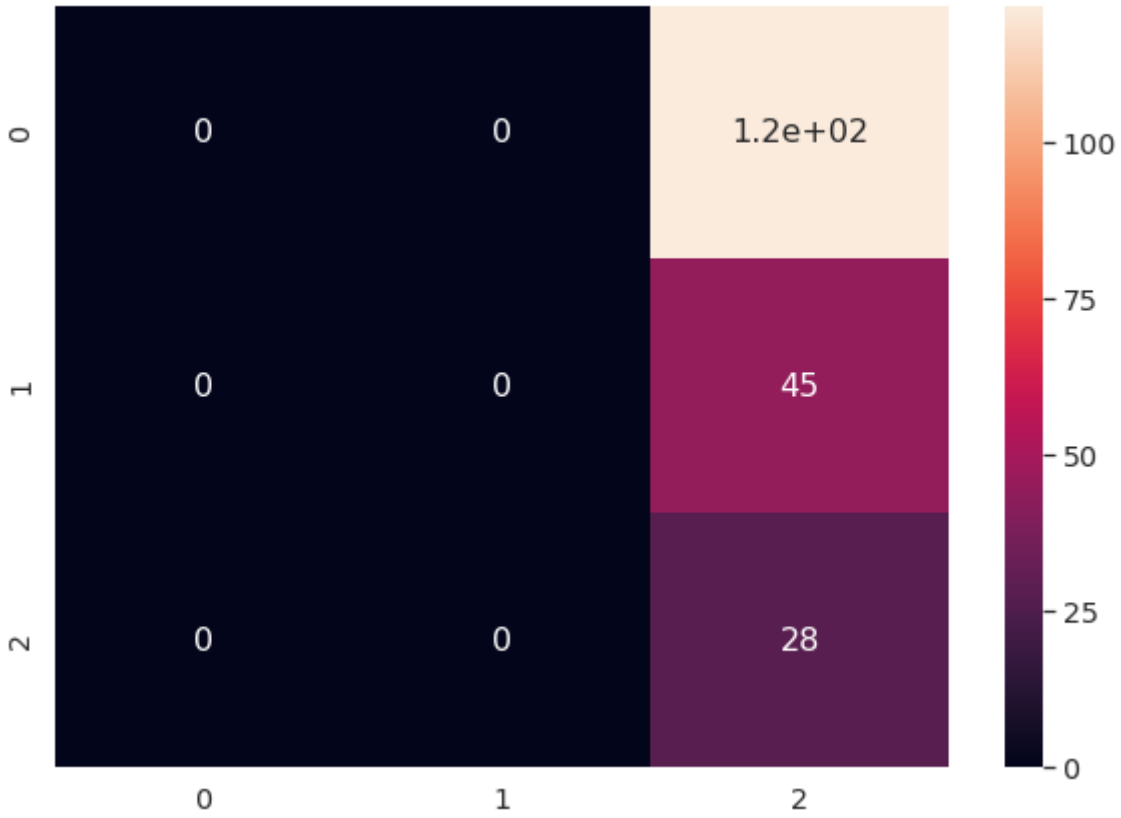

Figure 14: Confusion Matrix of Logistic Regression Model

4.4 Mean Confidence Interval

Interval : **11.906009244992296 +/- 0.24901666916238163**

4.5 Results for Boosting Algorithms

1. AdaBoost

Accuracy: **85.13%**

Classifier	Precision	Recall	F1-Score	Support
Class 0	0.90	0.86	0.88	122
Class 1	0.93	0.87	0.90	45
Class 2	0.61	0.79	0.69	28
Micro Avg	0.85	0.85	0.85	195
Macro Avg	0.81	0.84	0.82	195
Weighted Avg	0.86	0.85	0.86	195

2. XGBoost Classifier

Accuracy: **88.21%**

Classifier	Precision	Recall	F1-Score	Support
Class 0	0.92	0.89	0.90	122
Class 1	0.90	0.96	0.92	45
Class 2	0.70	0.75	0.72	28
Micro Avg	0.88	0.88	0.88	195
Macro Avg	0.84	0.86	0.85	195
Weighted Avg	0.80	0.88	0.88	195

3. Gradient Boosting Classifier

Accuracy: **93.80%**

Classifier	Precision	Recall	F1-Score	Support
Class 0	0.94	0.90	0.92	122
Class 1	0.91	0.96	0.93	45
Class 2	0.74	0.82	0.78	28
Micro Avg	0.90	0.90	0.90	195
Macro Avg	0.87	0.89	0.88	195
Weighted Avg	0.91	0.90	0.90	195

4.5 Results of Neural Networks

Accuracy of ANN - 88.6%
Accuracy of RNN - 69.66%

5. Conclusions and Future Work

Machine Learning Models work better than Neural Networks when the amount of training data is less.

- There is a high probability of overfitting in the Decision Tree Model and it generally gives low prediction accuracy for a dataset as compared to other machine learning algorithms.
- SVM works better on small datasets.
- Random Forest works better than Decision Tree.
- Logistic Regression doesn't work well when there is high correlation between variables.
- AdaBoost is primarily used for Binary Classification, hence gives a sub-optimal accuracy.
- XGBoost cuts off Gradient Boosting at the certain point, hence it gives a lower accuracy than Gradient Boosting.
- Gradient Boosting works the best out of all the algorithms applied.

5.1 Future of our project

Our plan for the future is initially to collect more data and training the models on that data. By feeding more data, we can improve the accuracy of Neural Networks and delve more into the Deep Learning side of our project.

Next, we are going to work towards making an end-to-end platform solution for colleges. This would not only help in predicting the performance of students, but it would also help in identifying their weak points.

Our project would also work as a useful tool for teachers, as predicting the students' performance will help them to take necessary actions and help to determine the sections of class where the teachers have to pay extra attention.

Finally, we are planning to write a paper regarding our project's workings, results and conclusions which, with the help of our guide, we would send to various conferences for publication.

6. References

- M. M. A. Tair, A. M. El-Halees, Mining educational data to improve students' performance: a case study, International Journal of Information2(2)
- V. Oladokun, A. Adebanjo, O. Charles-Owaba, Predicting students' academic performance using artificial neural network: A case study of an engineering course, The Pacific Journal of Science and Technology
- V. Ramesh, P. Parkavi, K. Ramar, Predicting student performance: a statistical and data mining approach, International Journal of Computer Applications 63 (8) (2013) 35–39.
- M. Christian, M. Ayub, Exploration of classification using nbtree for predicting students' performance, in: Data and Software Engineering(ICODSE), 2014 International Conference on, IEEE, 2014, pp. 1–6
- T. Wang, A. Mitrovic, Using neural networks to predict student's performance, in: Computers in Education, 2002. Proceedings. International Conference on, IEEE, 2002, pp. 969–97
- M. Wook, Y. H. Yahaya, N. Wahab, M. R. M. Isa, N. F. Awang, H. Y. Seong, Predicting ndum student's academic performance using datamining techniques, in: Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on,Vol. 2, EEE, 2009, pp.357–361