# **Part 2:**

## **1.1:**

Introduction:

The problem my project is trying to solve is to see if the preparation course in schools is needed to pass and get high marks or not. By comparing students’ grades in three exams (Math, Reading, and Writing) to whether they took a preparation course before or not.

**-The project impacts schools to know exactly if:**

**1.** the preparation course they are implementing is working to its intended purpose

**2.** if it needs to be improved on later stages.

**3.** If the preparation course implementation is needed in the school.

**-It can impact students by:**

**1.** Them knowing that they must take an extra exam that is expected to improve their grades in Math, reading, and writing exams later.

**2.** If they need to take the preparation course the school is implementing.

**3.** To see if it lowers students’ performance in later exams.

## **1.2:**

## **1.2.1: Materials.**

* The source of the dataset is:

<https://www.kaggle.com/datasets/whenamancodes/students-performance-in-exams>

originally taken from: <http://roycekimmons.com/tools/generated_data/exams>

The description of the dataset:

The dataset looks to solve whether the preparation course is needed for students to improve their grades while looking at other factors that may play a role in increasing their performance such as whether they had lunch before the exam, or if they come from a certain race or ethnicity, or whether they are male or female, and their parental level of education.

* The data inside the dataset was generated by a randomizer so it can be considered inaccurate.
* The attributes mentioned in the dataset were:

1. Gender: which identifies if the student is male or female
2. Race/ethnicity: which describes the race of the student, there are five races.
3. Parental level of education: Which mentions what level the students’ parents had.
4. Lunch: to know whether the students had lunch before taking the course
5. Test preparation course: this attribute mentions if the course was taken by the students or not.
6. Math score: their score in the math exam after taking the course.
7. Reading score: their score in the reading exam after taking the course
8. writing score: their score in the writing exam after taking the course.

* The dataset consisted of 8 features (columns) and 1000 records (rows) and it has no missing values.
* The number of projects the dataset was used in is 60 projects.
* The dataset had no missing values. But it had categorical values. I used the encoding method to turn them into numerical values, and I used normalization to make the values on the same scale. I explored the dataset and replaced the five races from the (race/ethnicity) feature into numbers from 0 to 4.
* The exploratory analysis I used for the dataset to get information about the data and the values inside, were:

The describe method of python which gives a brief description of the measures of the numerical values for each feature.

The info method of the data frames, which mentions how many missing values are inside each feature, and what’s the data type of the values inside each feature. I made sure to encode some of the features which data types were (object) or (string), to (int) or (float) so they can be used in training the models in a later stage. I managed to find the correlation of the features, and I used some visualization methods to find out more about the dataset. Such as bar plots to

A picture containing screenshot, line, plot, text

Description automatically generatedA picture containing screenshot, diagram

Description automatically generatedA picture containing screenshot

Description automatically generated

The relationship between the math score and reading score and writing score features. In a scatter plot, it shows that the features are in a correlation since both rise at the same time.

This is a bar plot that shows the count of the gender feature, it shows that the male count is nearly the same count as the female count.

A picture containing text, screenshot, plot, line

Description automatically generatedA picture containing screenshot, text, rectangle

Description automatically generated

This is a line plot that shows the writing score with math score with the addition of the test preparation course if taken. It shows that the scores of the tests were lower when the test preparation course was taken.

## **1.2.1: Methods.**

* The models used were The ANN (Artificial Neural Network) and the Random Forest Classifier.
* I used the random forest classification model because of its high accuracy and flexibility. And it’s also strong in identifying patterns inside datasets, and the dataset I used mostly consists of binary attributes.

The ANN model was used to predict the target inside the dataset because the model itself is strong when identifying patterns in the dataset and since it is a deep learning model, it can be used in complex datasets and domains, and it often results in high accuracy.

* The pipeline charts for both the models:

First, the following images were the start of both models. So, I will include it then include both the models:

This is the beginning of the pipeline for each model, it started with the collection of the dataset, then importing the libraries, exploration of the dataset was made, and preprocessing techniques were used to prepare the data for modeling, then data splitting was made.

A picture containing text, screenshot, design

Description automatically generated

For the ANN model:

A picture containing text, screenshot, diagram, circle

Description automatically generated

For the Random Forest Classifier model:

A picture containing text, screenshot, diagram, circle

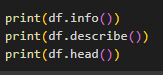
Description automatically generated

Both models start with importing libraries and exploring the dataset, along with the preprocessing techniques and the splitting of the data, then each model continues as it should. With the random forest taking the labeled data as a normal classification model. While the ANN model creates layers and uses unit numbers along with classifiers and epochs to predict the target feature. Both models end with evaluation measures that were later improved using different techniques such as changing the preprocessing, splitting, unit numbers, and layers.

* **The implementation of the first & second model was as the following:**

**Step 1:**

I imported the libraries, such as numpy, pandas, scikitlearn, seaborn, and other libraries that are model related such as RandomForestClassifier and train\_test\_split



**Step 2:**

I made sure to explore the dataset I have by using different techniques like

A picture containing screenshot, rectangle, display, text

Description automatically generatedA picture containing text, screenshot, plot, line

Description automatically generatedA picture containing screenshot

Description automatically generatedAnd I used df.corr() to find the correlation between every feature. And I made some visualization charts to gather more information about the dataset.

Math score vs writing score count of each gender checking whether the course improved the score results or no

**Step 3:**

A screen shot of a computer program

Description automatically generated with low confidenceI started the preprocessing stage for the dataset by implementing a label encoder to turn categorical features into numerical features, so that they can be trained to the model. I used a feature scaler (Normalization) to ensure that the numerical data are on the same number scale.

**Step 4:**

After preprocessing, I split the data based on the correlation between the features to the features that are trained, and the target variable.

Then I split the data again to training and testing data, using the train\_test\_split method.



**Step 5:**

I started with the modeling process for the ANN, where I specified the model, I’m going to use and create the layers, an input layer, a single hidden layer, and an output layer

A screenshot of a computer program

Description automatically generated with medium confidence

Picture of the normalization process.



Input Layer



Hidden Layer

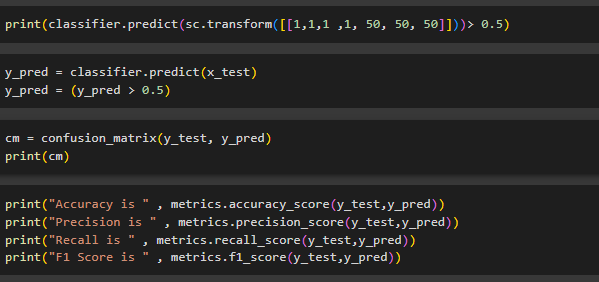


Output Layer

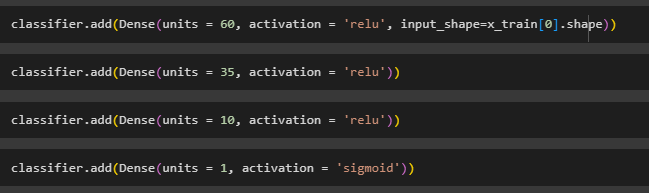
Here I specified the metric I wanted to predict, with the number of epochs and the batch size that the model will go through.

A screenshot of a computer

Description automatically generated with low confidence

Here’s the continuation of the modeling process, Then I used four evaluation measures so I know if the model is working as intended or not, accuracy, precision, recall, and f1 score.

The measures weren’t good at first, the accuracy was only 60%. So, I added another hidden layer, and increased the number of units inside each layer. And accuracy went up to nearly 75%.

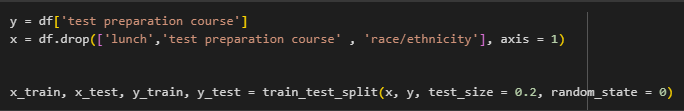


A screenshot of a computer program

Description automatically generated with medium confidence

**Step 6:**

Then I started with the modeling process for the Random Forest Classification model, which is an improved version of the decision tree model.

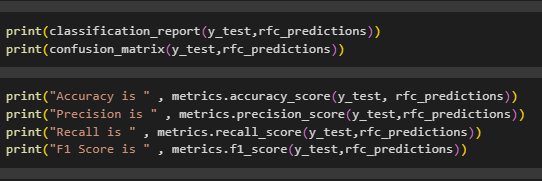
The splitting of the data change in the second model is due to ANN’s ability to work with missing or wrong features. But for the random forest model, the splitting must be done correctly, to get the best result.

A screenshot of a computer program

Description automatically generated with medium confidenceThis is the modeling process, I put the number of estimators up to 200 so it can give a more accurate prediction.

**Step 7:**

After the modeling process, I used the same evaluation measures I used for ANN for the random forest classifier, and I used a classification report and a confusion matrix for more details about the results.



The accuracy wasn’t that good at the beginning (50%), which was due to using the same splitting as the ANN, but as mentioned above, the splitting of the random forest model was later changed, and more features were dropped alongside the target variable. The accuracy after the change was as the following:

A screenshot of a computer program

Description automatically generated with medium confidence

* The ANN model and the random forest classification excel at feature engineering and selection, so if there are other models in the organization that are used to solve certain problems. The random forest and ANN models can be integrated with them to specify which features can be used and to which cause. The models can also be used as a final prediction to give the best accuracy if there are other models that have predicted before. CNN can be used with ANN with great efficiency since both are deep learning specified models. And CNN can be used for image classification which can be used in ANN to make high level representations of the data, and therefore high accuracy and accurate predictions.

This model can be used in schools because of its good prediction, the model works in these steps: First it will gather information regarding the students before they take the test preparation course that will help them get better marks in the three exams. When they take the test preparation course, they will do the exams. Therefore, their exam scores should increase. For the students that did not take the course, the school will do several events that’s goal is to encourage students to take the course because of its benefits for the students. The schools can also benefit from it by investing In the course and maybe evolve it to help the students in more than three exams in the future.

* The measures of performance I used were the default classification measures.

Accuracy

Precision

Recall

F1-score

The rationale behind each metric is that they can predict classification labeled data. Accuracy, for example can predict how accurate the model’s predictions were. Precision gives the predicted positive instances and their proportion. Recall compares the actual cases with the predicted samples. F1 score includes the precision and recall and compares them. Based on the dataset I acquired, the best metrics to use are these four.

* For the first model, ANN, the evaluation metrics gave a bad result with accuracy being only at 20%. I changed the targeted feature based on the correlation of the data, to a better target variable. And that made the model’s accuracy jump to 60%. Which is better than the previous accuracy, but still isn’t powerful enough. So, I added another hidden layer. Increased the unit numbers from 10 to 20 to 40 to 60. Based on the metrics I have been getting on each change, increasing unit numbers seemed to be the best solution for improving the model’s performance.

For the second model, Random Forest Classifier, the evaluation metrics at the start were on a medium scale, giving only 50% accuracy. That changed when I changed the splitting of the features, to add two more features to the df.drop statement on the x variable. This resulted in an improvement in performance for the random forest model, by 15% so the accuracy is now nearly 65%, to make more improvement. I started looking at the correlation between the features and came upon two features that work well with the targeted variable, so instead of the two features I already added, I changed them to two other features. And that resulted in the accuracy jumping to 75%, which is a good accuracy.

## **1.2.2: Results and Discussion:**

* **The results for the first & second models, with the four measures**

**(Accuracy / Precision / Recall / F1-score), Before & after changes:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models/Measures | Accuracy | Precision | Recall | F1-Score |
| ANN (Before) | 0.62 | 0.58 | 0.75 | 0.63 |
| Random Forest (Before) | 0.50 | 0.52 | 0.60 | 0.56 |
| ANN (After) | 0.75 | 0.75 | 0.90 | 0.82 |
| Random Forest (After) | 0.74 | 0.72 | 0.92 | 0.80 |

As described above, the results at first were bad, but after some changes to different factors that both the models rely on, the results improved to a state where they can predict better than a human guess.

Many factors played a part in these results, such as the pre-processing techniques used, the number of units and epochs and hidden layers for the ANN model. The selection of features for both models and the splitting of the data.

* Based on the accuracy for both the models. The accuracy is good but not efficient enough to be used in real-life applications. It can be used in small schools to predict if the course is needed to be taken and if it affects the performance of students for the three exams later. But its predictions shouldn’t be taken precisely due to its low accuracy.
* To implement the models with high efficiency in the future. We first need to address the limitations they carry; the models have a non-reliable performance due to the dataset’s features’ low correlations. The models’ predictions are not accurate enough to be used in the business field yet, it needs more data and more features that may increase the performance of the model.

The improvements for the model can be done by getting a more detailed dataset that focuses on a single target with high-correlation features. The results can be improved by using a model that is specialized in predicting low-correlation features. For the models, adding more hidden layers can be also beneficial, and changing the splitting of the data.

* To improve the model performance, I plan on changing the targeted variable, and gathering more information about the dataset, I also plan on changing the model used to a model that is good with low-correlation datasets. The gathering of the data could be done in real schools by real students and their scores. I intend to introduce the project to a big educational organization that would benefit from it, and we can work on improving the models with real-life scenarios and information.

Schools can benefit and invest in the project by implementing a course that prepares students for their upcoming exams, and if it grants high use and results. It can be invested in and introduced to the world.