A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name* | Derek O’Hara |
| *Student Number* | sbs24018 |
| *Module Title* | Machine Learning |
| *Assessment Title* | CA1 Project |
| *Assessment Due Date* | 28th April 2024 |
| *Date of Submission* | 28th April 2024 |

**Declaration**

By submitting this assessment, I confirm that I have read the CCT policy on academic misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source.

I declare it to be my own work and that all material from third parties has been appropriately referenced.

I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Contents

[Introduction 1](#_Toc165147067)

[Data Characterization and Pre-processing 1](#_Toc165147068)

[References 12](#_Toc165147069)

[Dataset: 12](#_Toc165147070)

[Github: 12](#_Toc165147071)

[Code comments: 12](#_Toc165147072)

# Introduction

Analysing and predicting student performance is crucial for educational institutions to improve efficiency. It helps identify students with low academic achievements early on, high dropout rates, and delays in graduation (Daniel, B. 2015). It is very important for educational institutions, to understand the potential of using collected data to improve the learning efficacy and academic achievements of both the individual student and institutions themselves (M. Nachouki and M. Abou Naaj, 2022).

This report aims to explore and demonstrate the application of machine learning algorithms to predict academic success. I will be focusing on predicting a group of students' final grade “G3” in a course based on their prior academic data and demographic features.

The dataset I have chosen is called the “Student Performance Dataset” and can be found at [kaggle.com](https://www.kaggle.com/datasets/devansodariya/student-performance-data). As I become more familiar with the data and using prediction and classification algorithms where appropriate, I hope to be able to identify patterns and insights from the data to achieve my goal. (word count = 160)

# Data Characterization and Pre-processing

This dataset was compiled through a survey of students' math course in secondary school. It is made up of 33 columns/features of both numerical and categorical types and 395 rows. The features comprise of several considerations such as the students past academic performance, demographics, and social factors.

Data preprocessing refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models (K Goyal 2024). My first steps are to load and display the main features of the dataset to get an initial understanding and identify any obvious relationships or anomalies.

A screenshot of a computer

Description automatically generated

Displaying data types tells me encoding categorical variables will be necessary. There are no null values in my dataset, so no need to implement any missing value handling techniques.

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Ihe initial description of the metrics of the numerical columns also confirms that there are no null values in the numerical features.

A screenshot of a computer

Description automatically generated

A Boxplot of the numerical columns helps to identify any distint outliers (Individual Dots) in the data.

A screen shot of a graph

Description automatically generated

The variable with the widest range of values with a lot of outliers is **absences**, indicating that most students have few absences, but some have a lot more than typical.

The grades categories **G1, G2 & G3** are spread out with several outliers, particularly for the final grade **G3**. This suggests varied performance among students, with some scoring much higher or lower than the typical range.

To deal with outliers effectively, I can:

* **Keep them**: If they represent valid variations.
* **Remove them**: If they're due to errors or if their extreme values could skew my analysis.
* **Transform them**: Apply a transformation to reduce the impact of outliers, for instance, using a log transformation (S. Anuganti, 2020).

I visualise the distribution of **G3** values by means of a histogram using matplotlib and seaborn libraries. We can see the significant left skew of the **G3** grade distribution caused by 38 entries where **G3** is zero.

A graph with blue lines and a blue line

Description automatically generated

I similarly plot the distribution of all the numerical features.

A screenshot of a graph

Description automatically generated

The correlation matrix heatmap shows that the strongest correlation between **G3** marks is with the marks obtained in **G1** & **G2**. There is a risk of multicollinearity here.

A screenshot of a computer

Description automatically generatedWe also see negative correlation between **failures** and **G1**, **G2**, **G3**, which indicates that students with more past failures tend to have lower current grades.

**studytime** is negatively correlated with **failures**, suggesting that more study time is associated with fewer failures.

I perform a thorough exploratory data analysis by creating visual relationships between each feature and the target variable, **G3**.

Firstly, I use scatter plots to visualise the numerical features. A screenshot of a computer screen

Description automatically generatedA group of graphs with numbers

Description automatically generated with medium confidence

A group of graphs with dots

Description automatically generated with medium confidence

I used bar charts to plot the mean of **G3** for each categorical feature and visualise the relationship.

A group of colored bars

Description automatically generated with medium confidence

A group of colored bars

Description automatically generated

A group of colored bars

Description automatically generated with medium confidence

I picked some of the main features and created pair plots to further analyse patterns. The strongest relationships to **G3** are with **G1** & **G2**.

A screenshot of a computer screen

Description automatically generated

With **studytime**, we can see increasing trend in median grades as study time increases but the overall spread is similar. We also see the outliers across all levels were study time did not influence a zero mark.

A screenshot of a graph

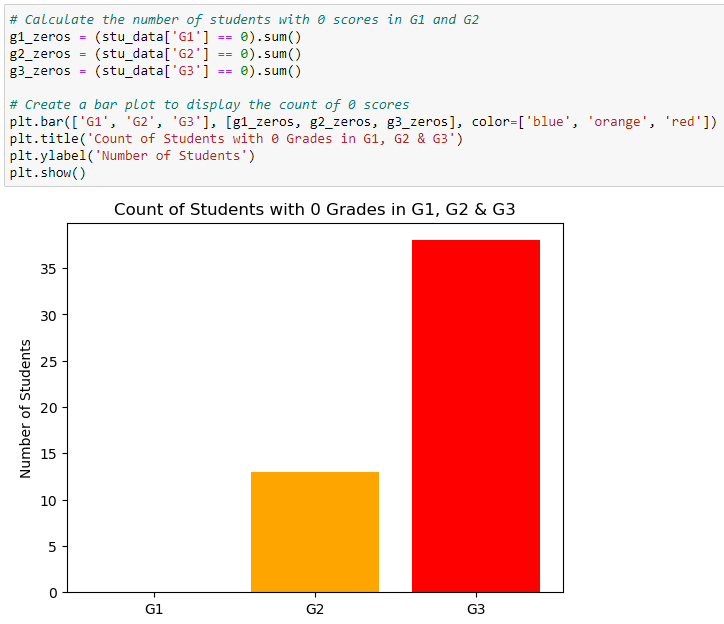
Description automatically generated

I encoded the categorical variables into numerical variables using one hot encoding.

A screenshot of a computer

Description automatically generated

Between **G1**, **G2** & **G3,** the number of zero grades have increased to 38.



Based on my analysis, I assume that the 38 students did not sit the exam.

A graph with blue lines and numbers

Description automatically generated

After dropping the 38 zero **G3** grades, we can see that the distribution skew has improved.

(word count = 568)

# Analysis and Machine Learning Models

Multiple regression models were developed to predict the final grade **G3**.

The models included:

* Linear Regression.
* Support Vector Machine Regression.
* Decision Tree Regressor.
* Random Forest Regressor.

These models were chosen based on their ability to handle different types of data distributions and their robustness in regression tasks.

Each model was trained using different splits of the data (20%, 25%, and 30%) to ensure robustness and reliability of the predictions across various training scenarios. Cross-validation techniques were employed to validate the results, ensuring that the models were not overfitting and were generalisable to new data.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Metric** | **Split 20%** | **Split 25%** | **Split 30%** | **Hyperparameter Tuning 1 - 30%** | **Hyperparameter Tuning 2 - 30%** | |
| **Linear Regression** | Mean Absolute Error | 0.67482145 | 0.719230638 | 0.720122526 |  | | |
| Mean Squared Error | 0.80939602 | 0.883210251 | 0.884019061 |
| Root Mean Squared Error | 0.899664393 | 0.939792664 | 0.940222878 |
| R-squared | 0.915986045 | 0.908474451 | 0.912569543 |
| Cross-validation R2 scores | [0.92048806 0.9338324 0.91119051 0.93778319 0.91819327] | [0.92048806 0.9338324 0.91119051 0.93778319 0.91819327] | [0.92048806 0.9338324 0.91119051 0.93778319 0.91819327] |  | | |
| Mean CV R2 | 0.924297487 | 0.924297487 | 0.924297487 |
| Standard Deviation in CV R2 | 0.009963381 | 0.009963381 | 0.009963381 |
|  | | | | | | | |
| **SVM** | Mean Absolute Error | 0.678925458 | 0.700780294 | 0.742567348 |  | | |
| Mean Squared Error | 0.795795817 | 0.838678055 | 0.919241518 |
| Root Mean Squared Error | 0.892073885 | 0.915793675 | 0.958770837 |
| R-squared | 0.917397723 | 0.913089245 | 0.909086004 |
| Cross-validation R2 scores | [0.91920965 0.93456898 0.91065012 0.92628461 0.91095953] | [0.91920965 0.93456898 0.91065012 0.92628461 0.91095953] | [0.91920965 0.93456898 0.91065012 0.92628461 0.91095953] |  | | |
| Mean CV R2 | 0.920334576 | 0.920334576 | 0.920334576 |
| Standard Deviation in CV R2 | 0.009175694 | 0.009175694 | 0.009175694 |
|  | | | | | | | |
| **Decision Tree** | Mean Absolute Error | 0.736111111 | 0.811111111 | 0.787037037 |  | | |
| Mean Squared Error | 1.013888889 | 1.255555556 | 1.083333333 |
| Root Mean Squared Error | 1.006920498 | 1.120515754 | 1.040833 |
| R-squared | 0.894760026 | 0.869888951 | 0.892857143 |
| Cross-validation R2 scores | [0.86270158 0.88920617 0.8318575 0.84002832 0.85215178] | [0.86270158 0.88920617 0.8318575 0.84002832 0.85215178] | [0.86270158 0.88920617 0.8318575 0.84002832 0.85215178] |  | | |
| Mean CV R2 | 0.855189071 | 0.855189071 | 0.855189071 |
| Standard Deviation in CV R2 | 0.019985177 | 0.019985177 | 0.019985177 |
|  | | | | | | | |
| **Random Forest** | Mean Absolute Error | 0.611111111 | 0.638888889 | 0.630185185 | 0.977037625 | | 0.62158630044 |
| Mean Squared Error | 0.688830556 | 0.753157778 | 0.709588889 | 1.6710302 | | 0.70437644169 |
| Root Mean Squared Error | 0.829958165 | 0.867846633 | 0.842370992 | 1.292683333 | | 0.83927137547 |
| R-squared | 0.928500539 | 0.921951563 | 0.929820879 | 0.834733277 | | 0.93033639587 |
| Cross-validation R2 scores | [0.91232561 0.94254654 0.89575837 0.93189747 0.91709329] | [0.91339417 0.94391342 0.90000931 0.93431761 0.91579748] | [0.9137083 0.9413076 0.89519166 0.93093855 0.91358797] | [0.80674425 0.81211974 0.80560448 0.81309717 0.83063157] | | [0.91148225 0.94027011 0.89887866 0.93087874 0.9238684 ] |
| Mean CV R2 | 0.919924257 | 0.921486399 | 0.918946816 | 0.813639442 | | 0.92107563237 |
| Standard Deviation in CV R2 | 0.016154686 | 0.01566349 | 0.015901402 | 0.008982651 | | 0.01454006349 |

# References

Daniel, Ben. (2015). Big Data and analytics in higher education: Opportunities and challenges. British Journal of Educational Technology. 46. 10.1111/bjet.12230. Available at: <https://www.researchgate.net/publication/269936924_Big_Data_and_analytics_in_higher_education_Opportunities_and_challenges>

Nachouki, Mirna & abou naaj, Mahmoud. (2022). Predicting Student Performance to Improve Academic Advising Using the Random Forest Algorithm. International Journal of Distance Education Technologies. 20. 17. 10.4018/IJDET.296702. Available at: <https://www.researchgate.net/publication/362839521_Predicting_Student_Performance_to_Improve_Academic_Advising_Using_the_Random_Forest_Algorithm>

Kechit Goyal (2024) 'Data Preprocessing in Machine Learning'. Available at: [https://www.upgrad.com/blog/data-preprocessing-in-machine-learning](https://www.upgrad.com/blog/data-preprocessing-in-machine-learning/)

Suresh Anuganti (2020). How to remove outliers for machine learning. Available at: <https://medium.com/analytics-vidhya/how-to-remove-outliers-for-machine-learning-24620c4657e8>

## Dataset:

Student Performance Dataset (2022) <https://www.kaggle.com/datasets/devansodariya/student-performance-data/data>

## Github:

<https://github.com/derekoharacct/Machine-Learning-CA-1.git>

## Code comments: