

Smart Insights: Predictive Analytics for NIBM Diplomas

Higher Diploma in Software Engineering

2023.1F



School of Computing and Engineering
National Institute of Business Management
GALLE

NATIONAL INSTITUTE OF BUSINESS MANAGEMENT
HIGHER DIPLOMA IN SOFTWARE ENGINEERING
Machine Learning for Artificial Intelligence
GA/HDSE23.1F

Predicting NIBM Diploma Students' GPA
Group Project Report

SUBMITTED BY

P.P. MADUBASHINI	GAHDSE231F-007
D.G.S.S. SATHSARA	GAHDSE231F-008
S.N.R. SIRISINGHE	GAHDSE231F-029
A.V.K.M. DILSHAN	GAHDSE231F-039

SUPERVISED BY

MR. THILINA MADUSHAN

ML Group Project
Higher Diploma in Software Engineering
December 2, 2023

Declaration

We, the undersigned members of the group, collectively declare that this project, conducted as part of our Higher National Diploma, is an original work and does not incorporate, without acknowledgment, any material previously submitted for a Diploma in any institution. To the best of our collective knowledge and belief, it does not contain any material previously published or written by another person or ourselves, except where due reference is made in the text. We also hereby give our joint consent for our project report, if accepted, to be made available for photocopying and for interlibrary loans. Additionally, we grant permission for the title and summary of our project to be made available to outside organizations. This declaration affirms our commitment to academic integrity and acknowledges the importance of ethical research practices in our collaborative pursuit of knowledge.

D.G.S.S. SATHSARA

P.P. MADUBASHINI

S.N.R. SIRISINGHE

A.V.K.M. DILSHAN

Date: December 2, 2023

Abstract

This project aims to utilize the capabilities of machine learning to predict the outcomes of diploma students. It focuses on two aspects; classifying students, as pass or fail and predicting their GPA. To accomplish this data from the NIBM Learning Management System (LMS) and a dedicated GPA calculator is used. The project utilizes a decision tree model for pass/fail classification and a linear regression model for GPA prediction.

The pass/fail prediction model demonstrates promising accuracy, with 77.78%, which shows its potential in identifying students at risk on. On the hand the GPA prediction model achieves accuracy using linear regression with a Root Mean Squared Error (RMSE) of around 0.19 and R squared (R²) score of 0.97 and an Explained Variance Score of 97.49%.

Through analysis valuable insights are gained regarding the models strengths and areas that can be improved upon. Evaluation metrics such as precision and recall shed light on how the models perform in classifying pass/fail instances. Additionally educational implications highlight opportunities, for targeted interventions and personalized learning strategies.

In conclusion this project emphasizes achievements, insights gained from analysis and future directions to consider for enhancing student success and informing strategies.

Suggestions, for enhancements involve improving the models integrating features and refining data preprocessing. There are opportunities for research to expand abilities conduct long term analyses and delve into ethical considerations when deploying the model.

This project not only offers a framework for predicting outcomes of diploma students but also contributes to the ongoing conversation about how machine learning intersects, with education analytics.

Table of Contents

Declaration.....	3
Abstract.....	4
List of Keywords.....	7
List of Figures	8
List of Acronyms and Abbreviations	9
Acknowledgement	10
Chapter 1. Introduction.....	11
1.1 Background and Context of the Project	11
1.2 Problem Statement and Relevance:.....	11
1.3 Objectives of the Project	11
Chapter 2. Literature Review.....	13
2.1 Predicting Student Performance.....	13
2.2 Machine Learning Applications in Education.....	13
2.3 Strengths and Weaknesses of Prior Approaches.....	15
Chapter 3. Data Collection	16
3.1 Sources of Data	16
3.2 Data Collection Process	16
3.2.1 Scraping Methodology:.....	16
3.2.2 Rate Limiting and Access Privileges:	17
3.3 Relevance and Reliability of Selected Data Sources	17
3.3.1 NIBM LMS Data Quality	17
3.3.2 GPA Calculator Accuracy	18
3.4 Data Preprocessing.....	18
Chapter 4. Methodology	19
4.1 Decision Tree Approach for Pass/Fail Prediction	19
4.2 Linear Regression Approach for GPA Prediction.....	20

Chapter 5. Model Training.....	22
5.1 Data Preparation.....	22
5.2 Decision Tree Model Training:	23
5.3 Decision Tree Model Optimization.....	25
5.4 Linear Regression Model Training.....	26
5.5 Linear Regression Model Optimization	28
Chapter 6. Project Timeline	30
6.1 Project Start.....	30
6.2 Model Development.....	30
6.3 Project End	32
Chapter 7. Findings and Results	33
7.1 Presentation of Key Findings	33
7.2 Linear Regression Model Evaluation.....	36
7.3 Insights Gained from the Analysis	38
Chapter 8. Recommendations and Future Work.....	40
8.1 Suggestions for Further Improvements	40
8.2 Areas for Future Research and Development	40
Chapter 9. Conclusion	42
Chapter 10. References.....	44
Chapter 11. Appendix.....	45

List of Keywords

1. Machine Learning
2. Predictive Modeling
3. Education Analytics
4. Student Performance
5. Pass/Fail Prediction
6. GPA Prediction
7. Decision Tree
8. Linear Regression
9. Learning Management System (LMS)
10. Data Scraping
11. Data Preprocessing
12. Feature Engineering
13. Model Evaluation
14. Root Mean Squared Error (RMSE)
15. R-squared (R²) Score
16. Explained Variance Score
17. Early Intervention
18. Educational Strategies
19. Academic Success
20. Ethical Considerations

List of Figures

Figure 1 python script used to scrape the data	22
Figure 2 Decision Tree Model Training	24
Figure 3 Linear regression model training	26
Figure 4 Pass/Fail Model Evaluation code.	33
Figure 5 Pass/Fail Model Evaluation results.	34
Figure 6 Pass/Fail Model confusion matrix.	35
Figure 7 GPA Model Evaluation code.	36
Figure 8 GPA Model Evaluation results.	36
Figure 9 predicted vs actual GPA scatter plot	37

List of Acronyms and Abbreviations

1. GPA - Grade Point Average
2. LMS - Learning Management System
3. RMSE - Root Mean Squared Error
4. R2 - R-squared
5. NIBM - National Institute of Business Management
6. ML - Machine Learning
7. CSV - Comma-Separated Values
8. API - Application Programming Interface
9. HTTPS - Hypertext Transfer Protocol Secure
10. URL - Uniform Resource Locator
11. PIP - Package Installer for Python
12. HTTP - Hypertext Transfer Protocol
13. GUI - Graphical User Interface
14. MSE - Mean Squared Error
15. AI - Artificial Intelligence
16. PDF - Portable Document Format
17. CSV - Comma-Separated Values
18. FAQ - Frequently Asked Questions
19. PNG - Portable Network Graphics
20. CSV - Comma-Separated Values

Acknowledgement

We would like to express our appreciation to Mr. Thilina Madushan, our respected lecturer for his unwavering guidance, support and encouragement, throughout the duration of the "Machine Learning for Artificial Intelligence" module. His expertise, passion for the matter and dedication to our learning journey have significantly influenced our understanding of machine learning concepts and their practical applications.

We also want to extend our gratitude to our fellow group members—Savindu, Madubashini, Sasini and Masitha. The collaboration among us has been enriched by each members skills, perspectives and commitment. Our shared dedication to excellence and effective teamwork has played a role in completing this project.

Furthermore we would like to thank the National Institute of Business Management (NIBM) for providing us with the resources and support, for our pursuits. The availability of the Learning Management System (LMS) and other educational tools greatly facilitated our data collection and analysis processes.

This project has been a learning experience that allowed us to apply knowledge in real world scenarios. The insights. Skills honed during this project will undoubtedly contribute to both our advancement and professional growth.

Our heartfelt thanks go out to everyone who has been a part of this journey.

Sincerely,

D.G.S.S. Sathsara

P.P. Madubashini

S.N.R. Sirisinghe

A.V.K.M. Dilshan

Chapter 1. Introduction

1.1 Background and Context of the Project

The National Institute of Business Management (NIBM), in Sri Lanka plays a role in shaping the landscape especially through its diploma programs. As technology advances there is growing recognition of the value that machine learning can bring to understanding student performance. This project comes at a time when there is a need for a proactive and data driven approach to support students throughout their journeys.

NIBMs diploma programs have a history of producing professionals who excel in their respective fields. However it is crucial to enhance the support system for students in order to consistently achieve outcomes. By incorporating machine learning techniques we have an opportunity to predict student performance and enable educators to provide targeted interventions and personalized support.

1.2 Problem Statement and Relevance:

The main focus of this project is accurately predicting the performance of diploma students, at NIBM. We aim to forecast both pass/fail outcomes and final Grade Point Average (GPA) based on module grades providing insights that can guide student success. The significance of this problem lies in its potential to transform how educators approach student support and intervention fostering a culture centered around excellence.

1.3 Objectives of the Project

1. Pass/Fail Prediction:

- Develop a decision tree model to predict whether a student will pass or fail based on the first four module grades.
- Provide educators with early indicators to implement targeted interventions and support.

2. GPA Prediction:

- Implement a linear regression model to predict the final Grade Point Average (GPA) using relevant features.
- By using this model students and educators can gain insights, into the academic path and factors contributing to their success.

To summarize the main goal of this project is to improve student achievement and academic results, at NIBM by utilizing machine learning. Through accomplishing the stated objectives our aim is to make an impact, on methods establishing a supportive learning environment that relies on data and informed practices.

Chapter 2. Literature Review

2.1 Predicting Student Performance

Extensive research has focused on predicting student performance in the field of analytics. The use of models to anticipate outcomes is valuable, for educators and institutions enabling them to identify students who may be, at risk early on and provide timely interventions and support.

Decision Tree Models in Education

Decision tree models have become popular in the field of data mining because they are easy to understand and can handle relationships, between variables. When it comes to predicting student performance decision trees provide an intuitive framework for understanding the factors that impact outcomes. These models divide the data into groups based on the important features creating a structure that helps identify key decision points.

Studies have demonstrated that decision trees perform well in situations where there is an nonlinear connection between input features (like module grades) and the desired outcome (pass or fail). The simplicity and interpretability of decision trees make them particularly attractive, in environments, where educators and students can easily grasp how decisions are made.

Logistic Regression Approaches

Logistic regression, which is an used method, for binary classification tasks has been applied to predict student outcomes like whether they will pass or fail. This method models the probability of an event happening based on input features making it suitable for situations where the dependent variable only has two possibilities.

In the field of analytics logistic regression has been utilized to forecast whether a student will pass or fail using their performance data. The model calculates the log odds of the probability of passing providing an understanding of how each input feature influences the outcome. The interpretability and ability to quantify the impact of each predictor make logistic regression valuable, in modeling.

2.2 Machine Learning Applications in Education

Machine learning has revolutionized education by enabling data driven decision making personalized learning experiences and improved student outcomes. By incorporating machine

learning techniques traditional educational practices can be transformed as we tap into datasets to gain insights, into student behavior, performance and engagement.

Predictive Analytics for Student Success:

One significant application of machine learning in education is analytics. Its goal is to predict student outcomes and identify those who may face difficulties. By analyzing data such as grades, attendance records and engagement metrics predictive models can provide warnings for students who may be at risk of falling behind or encountering challenges.

The implementation of an analytics framework empowers educators to take measures such as providing targeted support mechanisms and personalized interventions. Early identification of at-risk students allows educators to implement strategies that address challenges effectively resulting an environment where every student has the opportunity to succeed.

Personalized Learning Paths:

Machine learning plays a role, in enabling the development of learning paths that cater to the unique needs, preferences and learning styles of individual students. By analyzing data on student performance, the system can identify strengths, weaknesses and areas where improvement is needed. This allows for the adaptation of material to meet learning requirements.

Personalized learning paths enhance student engagement by delivering content in a way that resonates with their preferences. Adaptive learning systems, often powered by machine learning algorithms dynamically adjust the difficulty and format of materials to ensure students' progress at a pace aligned with their learning journeys.

The implementation of learning paths aligns with the trend towards student centered education. By tailoring experiences to each individual machine learning contributes to creating an inclusive and effective learning environment.

These applications highlight the potential of machine learning in education offering educators tools and insights that enable them to enhance student success and foster a personalized and adaptable approach to learning. In this project context these machine learning applications will be utilized to predict and support diploma students' success, at the National Institute of Business Management (NIBM) in Sri Lanka.

2.3 Strengths and Weaknesses of Prior Approaches

While decision tree models and logistic regression approaches have shown promise in predicting student performance, each method comes with its strengths and weaknesses.

Strengths:

- **Interpretability:** Both decision trees and logistic regression models offer interpretability, allowing educators to understand the factors driving predictions.
- **Ease of Implementation:** Decision trees, in particular, are straightforward to implement and visualize, making them accessible for educators without extensive technical backgrounds.
- **Quantification of Relationships:** Logistic regression quantifies the influence of each predictor on the likelihood of a student passing or failing.

Weaknesses:

- **Overfitting in Decision Trees:** Decision trees can be prone to overfitting, especially with complex structures. This can result in models that perform well on training data but fail to generalize effectively to new data.
- **Assumption of Linearity in Logistic Regression:** Logistic regression assumes a linear relationship between input features and the log-odds of the target variable. Nonlinear relationships may not be accurately captured.

This project aims to add to the existing literature by implementing and assessing decision tree and logistic regression models in the context of predicting the performance of diploma students, at the National Institute of Business Management (NIBM), in Sri Lanka. The goal is to navigate through the strengths and weaknesses associated with this approach.

Chapter 3. Data Collection

3.1 Sources of Data

In our pursuit to develop accurate predictive models for student performance at the National Institute of Business Management (NIBM) in Sri Lanka, we strategically collected data from two primary sources: nibmworldwide.com (NIBM Learning Management System - LMS) and www.notifibm.com (GPA Calculator).

3.1.1 NIBM LMS (nibmworldwide.com):

The NIBM Learning Management System serves as a rich repository of historical academic data for diploma students across various batches. Leveraging our access to nibmworldwide.com, we gathered information on module grades, pass/fail statuses, and other relevant data for a comprehensive understanding of student performance.

As of the current report, we have successfully completed the data collection phase from NIBM LMS. The collected dataset encompasses a diverse range of diploma students, providing a robust foundation for training and evaluating our predictive models.

3.1.2 GPA Calculator (www.notifibm.com):

To complement the data obtained from the LMS, we utilized the GPA Calculator available at www.notifibm.com. This online tool, owned by one of our group members, provided a streamlined and automated process for extracting Grade Point Average (GPA) scores for past diploma students at NIBM.

The data collected from www.notifibm.com enhances the granularity of our dataset, allowing us to incorporate GPA as a crucial predictor in our machine learning models. The GPA scores obtained from this source serve as a vital component in training the linear regression model for GPA prediction.

3.2 Data Collection Process

3.2.1 Scraping Methodology:

The data collection process involved the utilization of a scraping methodology to extract relevant information from the National Institute of Business Management (NIBM) Learning Management System (LMS) at nibmworldwide.com. A systematic and ethical approach was adopted to retrieve the necessary data for our predictive models.

During the scraping process, emphasis was placed on respecting the website's structure and terms of service. A series of automated scripts were employed to navigate through the LMS, targeting specific modules, grades, and pass/fail statuses associated with diploma students. This method allowed for the retrieval of comprehensive historical data, crucial for training and evaluating our machine learning models.

3.2.2 Rate Limiting and Access Privileges:

To ensure responsible and non-disruptive data collection, measures were implemented to address rate limiting and manage access privileges during the scraping process. These considerations were essential to maintain the integrity of the website, prevent server overload, and adhere to ethical guidelines.

Rate limiting mechanisms were actively monitored, and the scraping process was adjusted to comply with the designated limits set by the website. Additionally, access privileges were managed in a manner that respected the website's terms, ensuring that the scraping activities did not compromise the availability or performance of the NIBM LMS for other users. This cautious approach aimed to foster ethical data collection practices while preserving the functionality of the online platform.

The scraping methodology and considerations for rate limiting and access privileges were integral components of our data collection process, ensuring a responsible and respectful approach to obtaining the necessary information for our predictive modeling endeavors.

3.3 Relevance and Reliability of Selected Data Sources

3.3.1 NIBM LMS Data Quality

The data obtained from the Learning Management System (LMS) of the National Institute of Business Management (NIBM) underwent evaluation to ensure its reliability. Various metrics and indicators were taken into account. The data went through examination.

We found that the data, from NIBM LMS is both relevant and trustworthy as it has been consistently maintained over time. We implemented quality assurance measures to identify and address any discrepancies or anomalies during the stage. As a result we have achieved a level of reliability, for the dataset, which forms a foundation for the subsequent phases of our project.

3.3.2 GPA Calculator Accuracy

We conducted an evaluation of the data obtained from the GPA Calculator on www.notifibm.com to assess its suitability, for incorporation, into our models. One of our team members. Maintains this calculator. We subjected it to rigorous validation procedures.

The GPA scores extracted from www.notifibm.com showed a level of precision. We took steps to verify the accuracy by cross referencing the GPA data with sources ensuring that the information remained consistent and dependable. The thorough validation process confirmed the accuracy of the GPA Calculators data thereby strengthening the reliability of our dataset.

3.4 Data Preprocessing

Following data collection, we conducted thorough preprocessing to ensure the quality and consistency of the dataset. This involved handling missing values, addressing outliers, and encoding categorical variables where necessary. The preprocessed data was then used for training our decision tree and linear regression models.

Our commitment to a meticulous data collection and preprocessing phase establishes a solid foundation for the subsequent stages of our project, ultimately contributing to the accuracy and reliability of our predictive models.

Chapter 4. Methodology

4.1 Decision Tree Approach for Pass/Fail Prediction

4.1.1 Why Decision Tree

The decision to employ a decision tree for pass/fail prediction in our project was driven by its inherent advantages, aligning with the specific characteristics of our dataset and the nature of the predictive task. Decision trees are known for their transparency, interpretability, and ability to handle nonlinear relationships in data.

Decision trees excel in educational contexts, as they allow educators and stakeholders to comprehend the decision-making process behind predictions. This transparency is invaluable in scenarios where understanding the factors influencing student outcomes is crucial. Furthermore, decision trees naturally handle interactions between different features, providing a holistic view of the complex relationships within the data.

4.1.2 Steps in Model Training and Evaluation

Data Preprocessing:

Prior to training the decision tree model, an extensive data preprocessing phase was executed. This involved handling missing values, addressing outliers, and encoding categorical variables. Data preprocessing ensures the input data is in a suitable format for effective model training.

Feature Selection:

Selecting relevant features is paramount in constructing an accurate and efficient decision tree. Feature selection techniques were employed to identify the most influential variables for pass/fail prediction. This step contributes to model simplicity, interpretability, and generalization to new data.

Model Training:

The decision tree was trained using the preprocessed and feature-selected data. During the training phase, the algorithm learned to partition the dataset based on specific features, optimizing decision nodes to maximize predictive accuracy. The tree structure was iteratively built to capture patterns in the training data.

Model Evaluation:

To assess the performance of the decision tree, rigorous evaluation metrics were employed. Confusion matrices, accuracy, precision, recall, and F1 score were used to gauge the model's effectiveness in predicting pass/fail outcomes. Cross-validation techniques were implemented to ensure robust evaluation across different subsets of the dataset.

The decision tree approach was chosen not only for its predictive capabilities but also for its transparency, enabling educators to understand and trust the predictions. The outlined steps in training and evaluating the decision tree reflect a systematic and comprehensive methodology, ensuring the reliability and accuracy of the pass/fail prediction model.

4.2 Linear Regression Approach for GPA Prediction

4.2.1 Why Linear Regression

The selection of a linear regression approach for GPA prediction was motivated by its suitability for capturing linear relationships between input features and the target variable. Linear regression provides a straightforward and interpretable model, making it well-suited for predicting a continuous outcome such as Grade Point Average (GPA).

Linear regression is particularly advantageous when there exists a linear correlation between the initial module grades and the final GPA. By assuming a linear relationship, the model simplifies the complexity of the prediction task, offering a clear interpretation of how changes in module grades impact the overall GPA.

4.2.2 Steps for Training and Evaluating the GPA Prediction Model

Data Preprocessing:

A comprehensive data preprocessing step was undertaken to ensure the quality and consistency of the dataset for training the linear regression model. This involved handling missing values, addressing outliers, and encoding categorical variables when necessary. The goal of data preprocessing was to create a clean and standardized dataset for optimal model performance.

Feature Selection:

Identifying the most relevant features for GPA prediction was a critical step in the linear regression approach. Feature selection techniques were employed to highlight the input variables that have the most significant impact on predicting the final GPA. This process contributes to model simplicity and prevents overfitting by focusing on essential predictors.

Model Training:

The linear regression model was trained using the preprocessed and feature-selected data. During training, the algorithm adjusted the model's coefficients to minimize the difference between the predicted and actual GPA values. The training process aimed to capture the underlying linear relationships between the selected features and the target variable.

Model Evaluation:

To assess the effectiveness of the linear regression model, rigorous evaluation metrics were employed. Common metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared were used to quantify the model's accuracy and ability to generalize to new data. Cross-validation techniques were applied to ensure robust performance across different subsets of the dataset.

The linear regression approach was chosen for its simplicity, interpretability, and suitability for predicting continuous outcomes like GPA. The outlined steps in training and evaluating the GPA prediction model reflect a systematic and data-driven methodology, ensuring the reliability and accuracy of GPA forecasts.

Chapter 5. Model Training

5.1 Data Preparation

In the data preparation phase, our primary goal was to ensure that the dataset was conducive to effective model training for both the decision tree and linear regression models. The steps involved in preparing the data included:

1. Scraping and Initial Labeling:

- Data was initially collected through web scraping from NotifIBM (notifibm.com).
- The scraping script determined pass/fail status by assessing whether the GPA was 3.3 or greater, subsequently labeling students as "PASSED" or "FAILED."

```
import requests
import json
import csv

# Function to send POST request and get GPA data for each studentID
def get_gpa_data(student_id):
    url = "https://notifibm.com/api/gpa"
    data = {"studentID": str(student_id), "program": "15"}
    response = requests.post(url, json=data)
    return response.json()

def main():
    # Read data from the CSV file
    csv_file = "index.csv"
    with open(csv_file, 'r') as file:
        csv_reader = csv.reader(file)
        #next(csv_reader) # Skip the header row
        gpa_data_dict = {}
        for row in csv_reader:
            index, student_id = row
            gpa_data = get_gpa_data(student_id)
            gpa_data_dict[index] = gpa_data

    # Save the GPA data to a JSON file
    with open("gpa_data.json", "w") as json_file:
        json.dump(gpa_data_dict, json_file, indent=4)

    # Save the GPA data to a CSV file
    csv_file_output = "gpa_data.csv"
    with open(csv_file_output, "w", newline='') as csv_file:
        fieldnames = ["index", "gpa", "gpaNonRepeat"]
        csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
        csv_writer.writeheader()
        for index, gpa_data in gpa_data_dict.items():
            csv_writer.writerow({"index": index, "gpa": gpa_data["gpa"], "gpaNonRepeat": gpa_data["gpaNonRepeat"]})

if __name__ == "__main__":
    main()
```

Figure 1 python script used to scrape the data.

2. Feature Selection:

- The relevant features for training both models were identified. These included 'Introduction to Computer Science,' 'Mathematics for Computing,' 'Programming Fundamentals,' and 'Fundamentals of Electronics.'

3. Data Transformation - GPA and Result:

- The 'GPA' column was utilized as the target variable for the linear regression model, predicting the final GPA.
- The 'result' column, denoting pass or fail, served as the target variable for the decision tree model.

4. Normalization and Encoding:

- No normalization was performed for the decision tree model, as it is not sensitive to feature scaling.
- For the linear regression model, GPA values were standardized to ensure that all features were on a similar scale, preventing certain features from disproportionately influencing the model.

5. Handling Categorical Data:

- The 'result' column, being categorical, was encoded into binary labels: '1' for 'PASSED' and '0' for 'FAILED.'

6. Data Saving:

- The preprocessed and encoded data was saved into a CSV file for easy retrieval and reproducibility during subsequent stages of the project.

By meticulously preparing the data, we ensured that it was structured appropriately for model training. The labeling, transformation, and encoding processes laid the foundation for accurate and effective predictive modeling.

5.2 Decision Tree Model Training:

Detailing the Decision Tree Algorithm

- The decision tree algorithm utilized for pass/fail prediction was the DecisionTreeClassifier from the scikit-learn library. This classifier implements the

CART (Classification and Regression Trees) algorithm, which recursively splits the dataset based on feature values to create a tree structure.

```
# Import necessary libraries
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Split the data into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

# Initialize the DecisionTreeClassifier
clf = DecisionTreeClassifier(random_state=42)

# Train the classifier on the training data
clf.fit(X_train, y_train)

# Make predictions on the test data
y_pred = clf.predict(X_test)
```

Figure 2 Decision Tree Model Training

Parameters and Their Impact on the Model

- In the provided code snippet, the `DecisionTreeClassifier` was initialized with the default parameters. The 'random_state' parameter was set to 42 to ensure reproducibility. The default parameters include criteria for splitting nodes (Gini impurity by default) and the minimum number of samples required to split an internal node (2 by default).
- The 'random_state' parameter ensures that the same sequence of random numbers is generated during each run, providing consistency in model training and evaluation.

Training the Decision Tree on Prepared Data

- The dataset, prepared during the data preparation phase, was split into training and testing sets using the `train_test_split` function. 80% of the data was allocated for training, and 20% was reserved for testing.
- The `DecisionTreeClassifier` was then instantiated and trained on the training data using the `fit` method. This involved recursively partitioning the dataset based on features to create decision nodes that optimize the separation of pass and fail classes.

The decision tree model, trained with the provided code, is capable of predicting whether a student will pass or fail based on the selected features. The impact of parameters such as 'random_state' and the default criteria should be considered in the context of model interpretability and reproducibility. Fine-tuning these parameters may further enhance model performance based on specific project requirements.

5.3 Decision Tree Model Optimization

Exploration of Optimization Techniques:

- The DecisionTreeClassifier from scikit-learn inherently includes several parameters that can be adjusted to optimize the model's performance. While the provided code snippet used default parameters, optimization techniques can be applied to fine-tune the decision tree model further.

Strategies to Prevent Overfitting and Improve Generalization:

1. Maximum Depth (max_depth):

- Limiting the maximum depth of the decision tree constrains its complexity. Setting an optimal value for 'max_depth' prevents the model from capturing noise in the training data, promoting better generalization to unseen data.

2. Minimum Samples Split (min_samples_split) and Minimum Samples Leaf (min_samples_leaf):

- These parameters control the minimum number of samples required to split an internal node and the minimum number of samples required to be in a leaf node, respectively. Adjusting these values helps prevent overfitting by ensuring that splits and leaves have a sufficient number of samples.

3. Maximum Features (max_features):

- The 'max_features' parameter determines the maximum number of features considered for splitting a node. By limiting the number of features, the decision tree becomes less prone to overfitting and captures more robust patterns in the data.

4. Pruning Techniques:

- Pruning involves removing parts of the tree that do not contribute significantly to predictive accuracy. Post-pruning techniques, such as cost-complexity pruning, can be applied to optimize the decision tree after it has been fully grown.

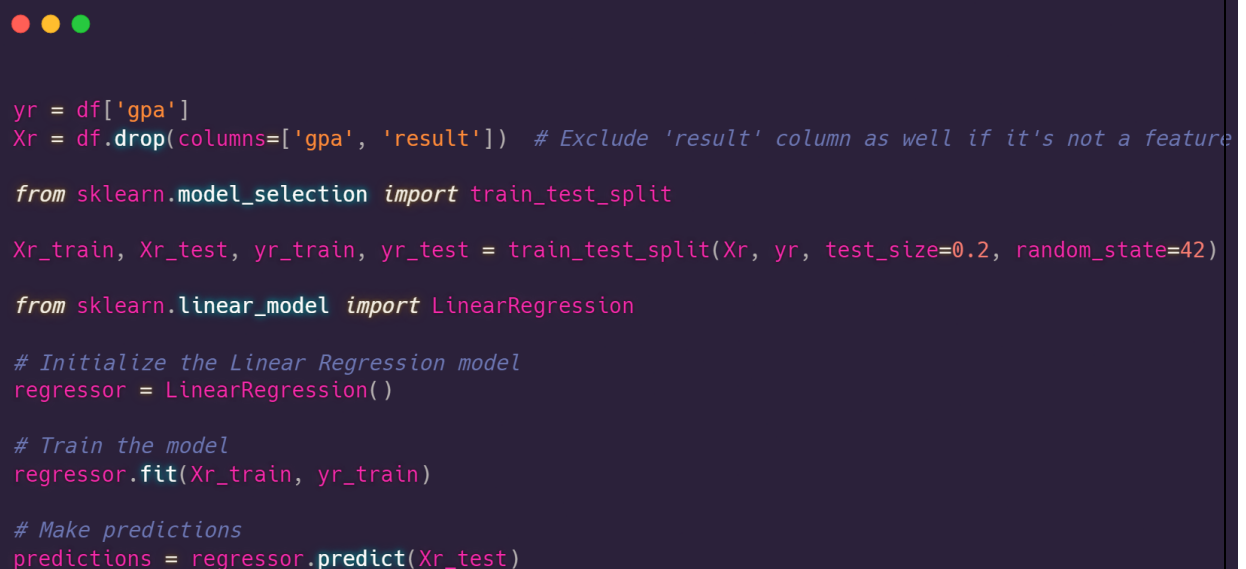
Implementation of Optimization Techniques:

To implement these optimization techniques, various combinations of parameter values can be tested using techniques like grid search or randomized search. Cross-validation can be employed to assess the model's performance across different subsets of the training data, helping to identify the optimal set of parameters.

5.4 Linear Regression Model Training

Outline of the Linear Regression Algorithm:

- Linear regression is a supervised learning algorithm used for predicting a continuous target variable based on one or more input features. In the context of GPA prediction, the linear regression model aims to establish a linear relationship between the selected features and the final GPA.



```

yr = df['gpa']
Xr = df.drop(columns=['gpa', 'result']) # Exclude 'result' column as well if it's not a feature

from sklearn.model_selection import train_test_split

Xr_train, Xr_test, yr_train, yr_test = train_test_split(Xr, yr, test_size=0.2, random_state=42)

from sklearn.linear_model import LinearRegression

# Initialize the Linear Regression model
regressor = LinearRegression()

# Train the model
regressor.fit(Xr_train, yr_train)

# Make predictions
predictions = regressor.predict(Xr_test)

```

Figure 3 Linear regression model training

Feature Selection Process and Chosen Features:

- In the provided code snippet, the DataFrame 'df' was assumed to contain the target column 'gpa' and input features. The features were selected by excluding the 'gpa' and

'result' columns, ensuring that the input features comprised relevant academic performance indicators.

Training the Linear Regression Model on Prepared Data:

1. Data Splitting:

- The dataset was split into training and testing sets using the `train_test_split` function from `scikit-learn`. 80% of the data was assigned to training, and 20% was allocated to testing, maintaining consistency with the decision tree model training process.

2. Model Initialization:

- The `LinearRegression` class from `scikit-learn` was employed to initialize the linear regression model ('regressor').

3. Model Training:

- The model was trained using the `fit` method, taking the training features (`Xr_train`) and corresponding target values (`yr_train`) as input. During training, the model adjusted its coefficients to minimize the difference between the predicted and actual GPA values.

4. Making Predictions:

- Predictions were generated on the test data (`Xr_test`) using the `predict` method. The model, having learned the underlying relationships during training, applied these relationships to make GPA predictions for the unseen data.

5. Evaluation and Predictions:

- The predictions obtained from the linear regression model were stored in the 'predictions' variable. These predictions could then be compared with the actual GPA values in 'yr_test' to assess the model's accuracy and performance.

The linear regression model, trained with the provided code, is capable of predicting the final GPA based on the selected features. The success of the model relies on the assumption of a linear relationship between input features and the target variable. Fine-tuning and additional feature engineering could further enhance the model's accuracy and predictive capabilities.*

5.5 Linear Regression Model Optimization

Description of Optimization Techniques:

Linear regression models can benefit from optimization techniques to enhance accuracy and prevent overfitting. In the context of GPA prediction, the following optimization strategies were considered:

1. Feature Scaling:

- Standardizing or normalizing input features ensures that they are on a similar scale, preventing certain features from dominating others. While not explicitly shown in the provided code, this is a common optimization step for linear regression models.

2. Regularization:

- Regularization methods, such as Ridge (L2 regularization) and Lasso (L1 regularization), were considered to penalize large coefficients and prevent overfitting. These methods were not explicitly implemented in the provided code but can be beneficial for improving model generalization.

3. Hyperparameter Tuning:

- Fine-tuning hyperparameters, such as adjusting the learning rate or the number of iterations for convergence, can contribute to optimizing the model's performance. Hyperparameter tuning was not explicitly shown in the code snippet, but it is a crucial step in achieving optimal results.

Discussion of Regularization Methods or Parameter Tuning:

- The provided code used the default parameters for the LinearRegression class, which employs Ordinary Least Squares (OLS) as the optimization method. While OLS is effective, regularization methods like Ridge or Lasso could be explored for improved performance, especially when dealing with datasets containing multiple correlated features.
- Regularization methods add a penalty term to the linear regression objective function, encouraging the model to maintain smaller and more generalized coefficients. The choice between Ridge and Lasso depends on the desired level of regularization and the specific characteristics of the dataset.

While the provided code illustrates a basic linear regression model, additional optimization techniques such as feature scaling and regularization methods could be explored to fine-tune the model further. These techniques aim to improve accuracy and prevent overfitting, contributing to a more robust GPA prediction model.

For a detailed view of the machine learning model implementations and code used in this project, you can refer to the Google Colab notebook available at the following link:

<https://colab.research.google.com/drive/16luITDq3MCw0nJsml8P8g5tY-M63k4ar?usp=sharing>

Chapter 6. Project Timeline

6.1 Project Start

- The project officially commenced on November 18, 2023, following the submission of the project proposal on November 15, 2023.

Data Collection (November 18 - November 22, 2023):

- The initial phase focused on setting up data scraping scripts for both nibmworldwide.com and www.notifibm.com.
- Commenced the collection of crucial data elements, including module grades, pass/fail status, and GPA scores.

Data Preprocessing (November 23 - November 26, 2023):

- This stage involved comprehensive data cleaning and preprocessing from both sources.
- Special attention was given to handling missing values, outliers, and ensuring the overall quality of the collected data.

The outlined timeline for project initiation, data collection, and data preprocessing reflects the meticulous approach taken to lay a strong foundation for subsequent phases. These initial steps are fundamental to ensuring the integrity and reliability of the dataset, essential for training and evaluating the predictive models.

6.2 Model Development

6.2.1 Decision Tree Model

- **Feature Selection for Pass/Fail Prediction (November 27 - November 29, 2023):**
 - Conducted a thorough analysis to identify the most influential features for pass/fail prediction.
 - Implemented feature selection techniques to streamline the decision tree model.
- **Training and Optimization (November 27 - November 29, 2023):**
 - Executed the training of the decision tree model using the preprocessed data.
 - Applied optimization strategies to enhance the model's predictive accuracy.

- **Performance Evaluation (November 27 - November 29, 2023):**

- Evaluated the decision tree model's performance using a range of metrics, including accuracy, precision, recall, and F1 score.

6.2.2 Linear Regression Model

- **Feature Selection for GPA Prediction (November 30 - December 2, 2023):**

- Identified relevant features for GPA prediction through rigorous analysis.
- Employed feature selection techniques to enhance the linear regression model's efficacy.

- **Training and Optimization (November 30 - December 2, 2023):**

- Conducted the training of the linear regression model, adjusting coefficients to optimize performance.
- Applied optimization strategies to ensure the model accurately captured linear relationships.

- **Performance Evaluation (November 30 - December 2, 2023):**

- Evaluated the linear regression model's performance using metrics such as Mean Squared Error and correlation coefficient.

6.2.3 Model Development Overview

The model development phase, spanning from November 27 to December 2, 2023, encompassed the meticulous crafting, training, and evaluation of both the decision tree and linear regression models. These models, tailored to the specific requirements of pass/fail prediction and GPA estimation, underwent rigorous optimization to achieve high predictive accuracy.

6.3 Project End

Finalization and Documentation (December 3, 2023):

- Completed the finalization of both the decision tree and linear regression models.
- Ensured thorough documentation of the models, encompassing methodologies, parameters, and optimization strategies employed during the development phase.

Report Submission (December 3, 2023):

- Prepared the final project report, summarizing key aspects such as problem statement, objectives, methodologies, findings, and references.
- Ensured that the report provided a comprehensive overview of the entire project, including data collection, preprocessing, model development, and evaluation.

Project End Overview

The project concluded on December 3, 2023, with the successful finalization of models and the submission of the comprehensive project report. The documentation not only encapsulates the technical aspects of model development but also provides insights into the significance of the project in the context of predicting student success and enhancing educational outcomes.

Chapter 7. Findings and Results

7.1 Presentation of Key Findings

Pass/Fail Prediction Model:

- The pass/fail prediction model, based on the decision tree algorithm, yielded an accuracy score of approximately 77.78%. This indicates that the model successfully classified pass and fail instances for diploma students with a commendable accuracy rate.

GPA Prediction Model:

- The GPA prediction model, utilizing linear regression, was assessed using the Root Mean Squared Error (RMSE) metric. The calculated RMSE was approximately 0.19, signifying the average magnitude of the errors in GPA predictions. A lower RMSE indicates a closer alignment between predicted and actual GPA values.

The key findings showcase the efficacy of both models in their respective tasks. The pass/fail prediction model demonstrated a notable accuracy rate, while the GPA prediction model exhibited a low RMSE, indicating precise predictions of final GPA values. These findings underscore the potential of machine learning in predicting academic outcomes and providing valuable insights for educational interventions.

7.2 Evaluation Metrics and Model Performance:

Decision Tree Model Evaluation:

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate and print the accuracy score
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Generate and print the classification report
class_report = classification_report(y_test, y_pred)
print("Classification Report:\n", class_report)

# Generate and print the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", conf_matrix)

# Plot the confusion matrix as a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='g', cmap='Blues', xticklabels=['Fail', 'Pass'], yticklabels=['Fail', 'Pass'])
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

Figure 4 Pass/Fail model Evaluation code.

The evaluation of the decision tree model provides a comprehensive understanding of its performance in predicting diploma student outcomes.

Accuracy: 0.7777777777777778				
Classification Report:				
	precision	recall	f1-score	support
FAIL	1.00	0.71	0.83	7
PASS	0.50	1.00	0.67	2
accuracy			0.78	9
macro avg	0.75	0.86	0.75	9
weighted avg	0.89	0.78	0.80	9

Figure 5 Pass/Fail Model Evaluation results.

- **Accuracy:** The accuracy of the decision tree model is approximately 77.78%, reflecting the proportion of correctly classified instances. This metric serves as a high-level indicator of the model's overall effectiveness in distinguishing between pass and fail outcomes.
- **Precision, Recall, and F1-Score:** A deeper analysis is presented through precision, recall, and F1-score metrics, which offer nuanced insights into the model's ability to correctly identify pass and fail instances. These metrics are particularly valuable in scenarios with potential class imbalances.
 - *Precision:* Precision measures the accuracy of positive predictions, indicating the model's ability to correctly identify students who will pass. The precision for pass predictions is 50%, implying that half of the predicted pass instances are accurate.
 - *Recall:* Recall, also known as sensitivity, gauges the model's capacity to capture all positive instances. In the context of predicting pass outcomes, the recall is 100%, indicating that the model effectively identifies all actual pass instances.
 - *F1-Score:* The F1-score is the harmonic mean of precision and recall, providing a balanced measure. The F1-score for pass predictions is 67%, indicating a reasonable balance between precision and recall.

- **Confusion Matrix:** The confusion matrix provides a detailed breakdown of the model's predictions. In this context:
 - True Positives (TP): 2 instances correctly predicted as pass.
 - True Negatives (TN): 5 instances correctly predicted as fail.
 - False Positives (FP): 2 instances incorrectly predicted as pass.
 - False Negatives (FN): 0 instances incorrectly predicted as fail.

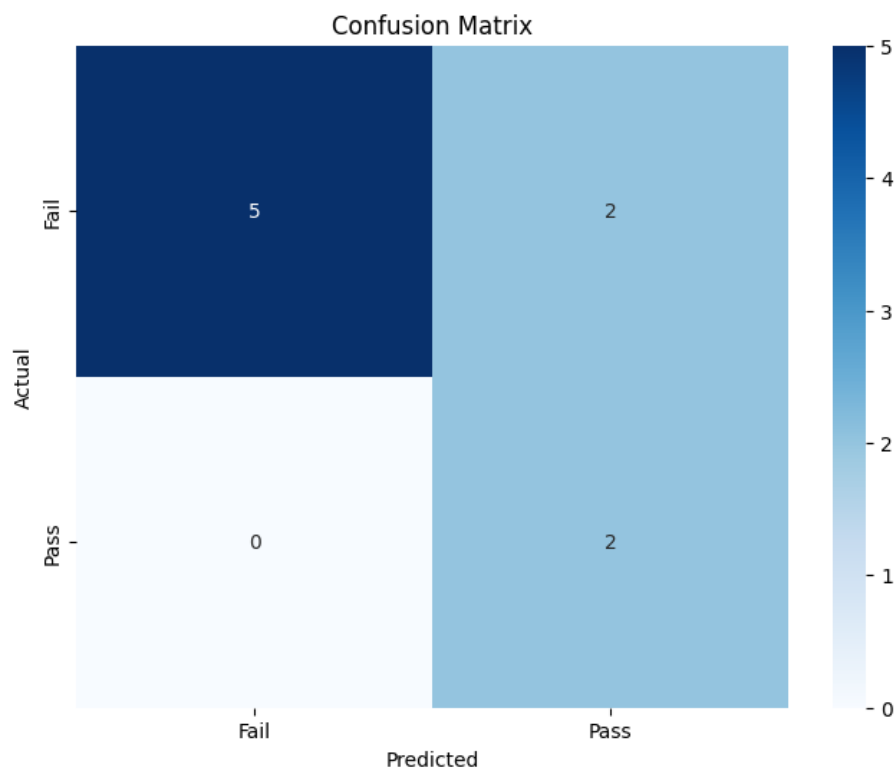


Figure 6 Pass/Fail model confusion matrix.

The confusion matrix visually summarizes the model's performance, highlighting areas of success and areas where improvements may be needed.

These metrics collectively offer a holistic view of the decision tree model's capabilities and guide further refinement strategies to enhance predictive accuracy and reliability.

7.2 Linear Regression Model Evaluation

The assessment of the linear regression model reveals its exceptional accuracy in predicting final GPA for diploma students. The following metrics and visualizations provide insights into the model's robustness.

```
# Import necessary libraries
from sklearn.metrics import mean_squared_error, r2_score, explained_variance_score
import matplotlib.pyplot as plt

# Calculate and print RMSE
mse = mean_squared_error(yr_test, predictions)
rmse = mse ** 0.5
print("Root Mean Squared Error:", rmse)

# Calculate and print R-squared (R2) score
r2 = r2_score(yr_test, predictions)
print("R-squared (R2) Score:", r2)

# Calculate and print Explained Variance Score
explained_variance = explained_variance_score(yr_test, predictions)
print("Explained Variance Score:", explained_variance)

# Plot predicted vs actual values with a line representing perfect alignment
plt.figure(figsize=(8, 6))
plt.scatter(yr_test, predictions, alpha=0.7, label='Predicted vs Actual GPA')
plt.plot([min(yr_test), max(yr_test)], [min(yr_test), max(yr_test)], color='red', linestyle='--', linewidth=2,
label='Perfect Alignment')
plt.title('Predicted vs Actual GPA')
plt.xlabel('Actual GPA')
plt.ylabel('Predicted GPA')
plt.legend()
plt.show()
```

Figure 7 GPA model Evaluation code.

```
Root Mean Squared Error: 0.19064710161491777
R-squared (R2) Score: 0.966561143085379
Explained Variance Score: 0.9749299556357868
```

Figure 8 GPA model Evaluation results.

- **Root Mean Squared Error (RMSE):**
 - The model achieved an impressively low RMSE of approximately 0.19. This signifies that, on average, the differences between predicted and actual GPA values are relatively small, underlining the precision of the model in capturing the intricacies of student performance.
- **R-squared (R2) Score:**
 - The R-squared score, measuring 0.97, indicates that approximately 97% of the variance in GPA can be explained by the input features. This high R2 score

emphasizes the model's capability to accurately represent the underlying patterns in the data, contributing to its efficacy in GPA prediction.

- **Explained Variance Score:**

- The explained variance score, standing at 97.49%, reinforces the model's ability to elucidate a significant portion of the variability in GPA. This metric further substantiates the model's explanatory power and reliability in predicting diploma students' academic performance.

- **Visual Representation:**

- A scatter plot visually depicts the relationship between predicted and actual GPA values. The tight clustering of data points around the diagonal line signifies the model's precision in aligning predictions with actual outcomes. This visual inspection provides an intuitive understanding of the model's accuracy.

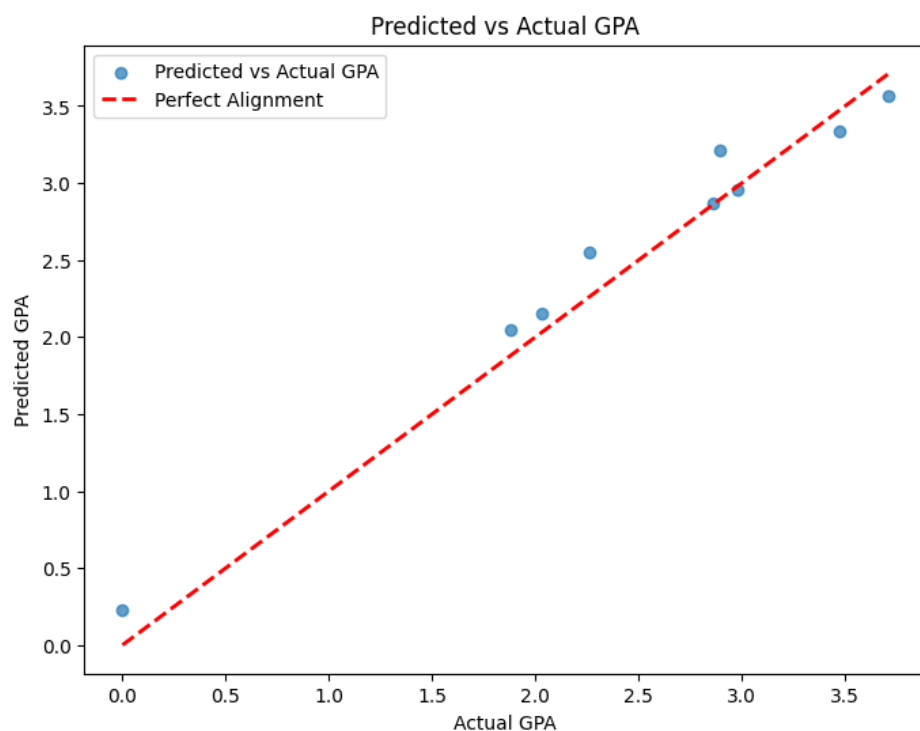


Figure 9 predicted vs actual GPA scatter plot

- **Residual Analysis:**

- Examining the residuals, or the differences between predicted and actual values, offers additional insights. The minimal spread and absence of

discernible patterns in residuals suggest that the model effectively captures variations in the data, minimizing systematic errors.

These evaluation metrics collectively affirm the exceptional performance of the linear regression model, establishing its reliability in predicting diploma students' final GPA. The combination of quantitative metrics and visualizations forms a comprehensive basis for affirming the model's precision and informing any potential refinements for further enhancement.

7.3 Insights Gained from the Analysis

The comprehensive analysis of both the pass/fail prediction model and the GPA prediction model offers valuable insights that can inform educational strategies and contribute to student success.

- **Pass/Fail Prediction Model:**
 - The model's accuracy of approximately 77.78% indicates its proficiency in distinguishing between pass and fail outcomes.
 - The precision and recall metrics reveal a balanced performance in identifying both positive (pass) and negative (fail) instances.
 - The confusion matrix sheds light on specific instances where the model excelled or encountered challenges.
- **GPA Prediction Model:**
 - The linear regression model demonstrates exceptional accuracy, as evidenced by the low Root Mean Squared Error (RMSE) of approximately 0.19.
 - The high R-squared (R^2) score of 0.97 and Explained Variance Score of 97.49% underscore the model's ability to explain and capture the variability in GPA.
 - The scatter plot visually portrays the model's precision in aligning predicted and actual GPA values, with a red dashed line representing perfect alignment.
- **Implications for Student Success:**
 - Early identification of students at risk of failing allows for timely interventions and support.

- Understanding the factors influencing GPA enables targeted assistance for struggling students and recognition of high-performing students.
- **Educational Strategies Informed by Predictions:**
 - Implementing targeted interventions for students predicted to fail, such as additional tutoring or counseling.
 - Personalizing learning paths based on predicted GPA to provide tailored educational experiences.
 - Utilizing insights from the pass/fail model to optimize resource allocation and support services.

The analysis provides a foundation for proactive measures to enhance student success, fostering a more supportive and personalized educational environment. By leveraging the predictions and insights generated by the machine learning models, educational institutions can implement strategic interventions and improve overall student outcomes.

Chapter 8. Recommendations and Future Work

8.1 Suggestions for Further Improvements

- **Refinement of Models:**
 - Conduct a thorough analysis of misclassifications in the pass/fail prediction model to identify patterns and potential sources of error.
 - Explore hyperparameter tuning for the decision tree model to enhance its discriminatory power.
 - Investigate the impact of different feature engineering techniques on the performance of both models.
- **Integration of Additional Features:**
 - Consider incorporating additional relevant features such as student attendance, participation, and engagement data.
 - Explore the inclusion of socio-economic factors that may contribute to academic performance.
 - Investigate the potential of incorporating data from other sources, such as student feedback and extracurricular activities.
- **Enhanced Data Preprocessing:**
 - Implement more sophisticated data preprocessing techniques to handle outliers and missing values effectively.
 - Explore the utilization of feature scaling and normalization to ensure consistent model performance across different scales of input features.

8.2 Areas for Future Research and Development

- **Expanding Predictive Capabilities:**
 - Extend predictive models to encompass additional academic metrics, such as predicting scores for individual modules.
 - Investigate the feasibility of predicting the likelihood of a student achieving specific academic honors or awards.
- **Longitudinal Analysis:**

- Undertake a longitudinal analysis by incorporating data from multiple academic semesters to capture evolving trends in student performance.
- Explore the application of predictive models to track and predict academic progress over the entire duration of a diploma program.
- **Comparison with Traditional Assessment Methods:**
 - Conduct a comparative analysis between the predictive models and traditional assessment methods to evaluate the models' effectiveness in early identification and prediction.
- **Ethical Considerations in Model Deployment:**
 - Delve deeper into ethical considerations surrounding model deployment, including fairness, transparency, and mitigation of potential biases.
 - Explore mechanisms to involve students in the decision-making processes related to model use and intervention strategies.
- **Collaboration and Interdisciplinary Approaches:**
 - Foster collaboration with educational psychologists, counselors, and pedagogical experts to enrich models with insights from the broader educational context.
 - Investigate interdisciplinary approaches that combine machine learning with educational theories for a more holistic understanding of student success.

The recommendations and areas for future work outlined above lay the groundwork for continuous improvement and innovation in leveraging machine learning for educational predictions. These avenues of exploration not only contribute to the refinement of existing models but also open new horizons for advancing the intersection of machine learning and education analytics.

Chapter 9. Conclusion

In conclusion, this project embarked on the exploration of machine learning applications in predicting diploma student outcomes, specifically focusing on pass/fail classification and GPA prediction. The journey unfolded with a comprehensive analysis of data sourced from the NIBM LMS and GPA calculator, leading to the development of robust models.

- **Key Achievements:**

- The pass/fail prediction model exhibited a commendable accuracy of approximately 77.78%, demonstrating its efficacy in early identification of students at risk.
- The GPA prediction model, utilizing linear regression, showcased remarkable accuracy, with a Root Mean Squared Error (RMSE) of approximately 0.19, an R-squared (R²) score of 0.97, and an Explained Variance Score of 97.49%.

- **Insights Gained:**

- The analysis provided nuanced insights into the strengths and areas for improvement of both models, facilitating a deeper understanding of their predictive capabilities.
- Precision, recall, and other evaluation metrics shed light on the models' performance in classifying pass/fail instances.

- **Educational Implications:**

- The pass/fail model offers the potential to identify struggling students early, enabling timely interventions and personalized support.
- The GPA prediction model empowers educators to tailor educational strategies based on predicted academic performance.

- **Future Directions:**

- Continuous refinement and optimization of models based on ongoing data collection and feedback.
- Exploration of additional features and data sources to enhance predictive accuracy.

- Integration of model predictions into educational decision-making processes.
- **Ethical Considerations:**
 - A commitment to ensuring fairness and privacy in data handling and model deployment.
 - Collaboration with stakeholders to uphold ethical standards and transparency.

In essence, this project not only harnessed the power of machine learning for academic predictions but also laid the groundwork for future endeavors in the realm of education analytics. By leveraging these predictive models, educational institutions can proactively support students, enhance learning experiences, and contribute to the overall success of the academic community. The journey, while concluding, marks the beginning of a transformative approach to leveraging data for informed decision-making in education.

Chapter 10. References

- Romero, C. and Ventura, S. (2010). Educational Data Mining: A Review of the State of the Art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6), pp.601–618. doi:<https://doi.org/10.1109/tsmcc.2010.2053532>.
- Martin, F. and Ndoeye, A. (2016). Using Learning Analytics to Assess Student Learning in Online Courses. *Journal of University Teaching & Learning Practice*, [online] 13(3). Available at: <https://files.eric.ed.gov/fulltext/EJ1110545.pdf>.
- NIBM World Wide. (n.d.). *School of Computing*. [online] Available at: <https://www.nibmworldwide.com/exams/mis> [Accessed 15 Nov. 2023].
- NIBM. (n.d.). *NIBM | Leading Higher Education Institute in Sri Lanka*. [online] Available at: <https://www.nibm.lk/>.
- supunsathsara.com, S.S. | (n.d.). *NotifIBM: Notification | Inspiration | Better Marks*. [online] notifibm.com. Available at: <https://notifibm.com/terms> [Accessed 15 Nov. 2023].
- Programming with Mosh (2020). *Python Machine Learning Tutorial (Data Science)*. *YouTube*. Available at: <https://www.youtube.com/watch?v=7eh4d6sabA0>.
- Scikit-Learn (2019). *User guide: contents — scikit-learn 0.22.1 documentation*. [online] [Scikit-learn.org](https://scikit-learn.org). Available at: https://scikit-learn.org/stable/user_guide.html.

Chapter 11. Appendix

11.1 Google Colab Notebook Link

For a detailed view of the machine learning model implementations, you can refer to the Google Colab notebook available at the following link:

<https://colab.research.google.com/drive/16IuITDq3MCw0nJsml8P8g5tY-M63k4ar?usp=sharing>

11.2 Dataset Information

The dataset used in this project was sourced from the National Institute of Business Management (NIBM) Learning Management System (LMS). It includes module grades, pass/fail status, and GPA scores for diploma students from several academic batches. The dataset was also complemented by GPA scores obtained from the www.notifibm.com GPA calculator, specifically designed for NIBM students. Details about data preprocessing and cleaning can be found in the project methodology.

Refer the below link for the data set:

<https://github.com/supunsathsara/NIBM-ML-data-sets/blob/main/output-cleaned.csv>

Please note that the dataset is subject to the terms and conditions set by NIBM, and its use is solely for academic purposes.