Maltepe University
Faculty of Engineering and Natural Sciences
Software Engineering Department

CEN 420 01 Introduction to Pattern Recognition

Student Performance Predictor Project

Project Final Report

Prepared By

Meaad Farag Bayuosef

Contents

***Abstract*—The ”Student Performance Predictor” project aims to predict students’ academic success or failure using machine learning techniques. This predictive tool is designed to assist educational institutions in identifying at-risk students early, allowing for timely and targeted interventions. The project was undertaken by students from the Software Engineering Department at Maltepe University and leverages a dataset of 395 student records.**

I. Introduction

A. Project Overview

The Student Performance Predictor project is an initiative undertaken by students from the Software Engineering Department at Maltepe University. The primary aim of this project is to leverage machine learning techniques to accurately predict the academic success or failure of students. By analyzing a diverse set of factors encompassing demographics, academic histories, and behavioral patterns, this project seeks to provide actionable insights that can help educational institutions identify at-risk students early in their academic journey. This early identification is crucial for implementing timely and effective interventions that can significantly enhance educational outcomes.

B. Objectives

- Predictive Model Development: Develop a robust predictive model that can determine the likelihood of students passing or failing based on their academic and demographic data.
- Identification of Key Factors: Identify and analyze the key determinants of academic success to guide educational interventions and support strategies.
- Data-Driven Decision Making: Enable educational stakeholders to make informed decisions based on data-driven insights, improving student retention rates and overall academic performance.

C. Importance

The importance of this project lies in its potential to transform educational practices through the use of advanced data analytics. Early identification of at-risk students allows educators to provide targeted support, which can prevent academic failure and promote student success. By understanding the factors that contribute to student performance, educators can tailor their approaches to meet the specific needs of their students, ultimately leading to improved educational outcomes.

D. Background and Motivation

Predicting student performance is a critical area of focus in educational data mining. Traditional methods of identifying struggling students often rely on manual analysis and subjective judgment, which can be time-consuming and prone to bias. This project aims to address these limitations by using machine learning algorithms that can process large datasets and uncover patterns that might not be immediately apparent to human observers.

E. Existing Methods and Improvements

Existing methods for predicting student performance typically involve statistical techniques and basic machine learning models. However, these methods often fall short in terms of accuracy and scalability. The Student Performance Predictor project seeks to improve upon these methods by employing more sophisticated algorithms, such as Logistic Regression, Support Vector Machines (SVM), and Decision Trees. These models are capable of handling complex, non-linear relationships between variables, providing more accurate and reliable predictions.

F. Team Members and Roles

- Manar Ahmed Safieeldin: Data Analyst, responsible for data pre-processing and feature engineering.
- Meaad Farag Bayuose: Model developer, responsible for data visualization, implementing and controlling machine learning algorithms, and running tests
- Sadeem Khater: Project coordinator, responsible for project management and documentation.

G. Project Scope

The scope of this project includes:

- Selection of a relevant topic and dataset.
- Comprehensive data collection and preprocessing.
- In-depth feature engineering and selection.
- Development and evaluation of multiple machine learning models.
- Detailed analysis of model performance and key findings.
- Preparation of comprehensive project documentation and final presentation.

H. Expected Outcomes

- A trained machine learning model capable of accurately predicting student performance.
- Insights into the significant factors influencing academic success, which can guide educational interventions.
- A detailed project report documenting the methodology, results, and implications of the findings.

By achieving these outcomes, the Student Performance Predictor project aims to make a meaningful contribution to the field of educational data mining and support the broader goal of improving educational outcomes for all students.

II. Literature Review

A. Overview of Predictive Models in Education

Predictive modeling in education has gained significant attention over the past decade as institutions seek data-driven methods to improve student outcomes. These models leverage historical data to forecast various educational metrics, such as student retention, academic performance, and graduation rates. The primary goal is to identify students at risk and provide timely interventions.

B. Early Work and Methodologies

Early efforts in predictive modeling primarily utilized traditional statistical methods such as linear regression and logistic regression. These models were favored for their simplicity and interpretability. For instance, Junco et al. (2011) used logistic regression to predict student retention based on demographic and academic factors. Their study highlighted the importance of engagement metrics, such as class attendance and participation, in predicting student success.

C. Machine Learning Advances

With the advent of machine learning, more sophisticated techniques have been employed to enhance prediction accuracy. Decision Trees, Random Forests, and Support Vector Machines (SVM) are among the popular algorithms used. For example, Bayer et al. (2012) demonstrated the effectiveness of Decision Trees in capturing complex, non-linear relationships between variables in predicting student dropouts.

D. Recent Studies and Innovations

Recent studies have explored ensemble methods and deep learning techniques to further improve predictive capabilities. Ensemble methods, such as Gradient Boosting Machines (GBM) and Random Forests, combine multiple models to reduce overfitting and improve generalization. A study by Zhang et al. (2019) utilized GBM to predict student performance, achieving higher accuracy than traditional methods.

Deep learning approaches, including neural networks, have also been explored. These methods can handle large datasets and complex feature interactions but require significant computational resources. Mi et al. (2020) applied a neural network model to a dataset of student academic records, demonstrating improved prediction accuracy for student performance compared to conventional machine learning models.

E. Feature Engineering and Selection

Effective predictive modeling relies heavily on the selection and engineering of relevant features. Feature engineering transforms raw data into meaningful representations that enhance model performance. Techniques such as one-hot encoding for categorical variables and normalization for numerical features are commonly used. Kotsiantis et al. (2004) emphasized the importance of feature selection in their study, which used filter and wrapper methods to identify significant predictors of academic success.

F. Addressing Data Imbalance

Data imbalance, where certain outcomes (e.g., failing students) are underrepresented, poses a challenge in predictive modeling. Techniques such as SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN (Adaptive Synthetic Sampling) are employed to balance datasets. Chawla et al. (2002) introduced SMOTE, which generates synthetic samples for minority classes, improving model performance on imbalanced datasets.

G. Interpretability and Ethical Considerations

Interpretability of predictive models is crucial, especially in educational settings where decisions impact students' futures. While complex models like neural networks offer high accuracy, their black-box nature makes them less interpretable. Ribeiro et al. (2016) proposed the LIME (Local Interpretable Model-agnostic Explanations) technique to explain predictions of complex models, enhancing their transparency.

Ethical considerations are paramount in predictive modeling. Ensuring data privacy, obtaining informed consent, and mitigating biases are critical. Barocas et al. (2016) discussed the ethical implications of machine learning, emphasizing the need for fairness and accountability in predictive modeling.

H. Application in Educational Interventions

Predictive models have practical applications in designing educational interventions. By identifying at-risk students early, institutions can implement targeted support measures, such as tutoring, counseling, and academic workshops. Arnold and Pistilli (2012) demonstrated the impact of predictive analytics on student retention programs, showing a significant reduction in dropout rates.

I. Gaps and Future Directions

Despite advancements, challenges remain in predictive modeling. Ensuring model generalization across diverse student populations and educational contexts is critical. Future research should focus on developing adaptive models that can update with new data and exploring the integration of predictive analytics with educational technologies for real-time interventions.

J. Conclusion

The literature underscores the potential of predictive modeling in education to enhance student outcomes. As machine learning techniques evolve, their application in educational contexts continues to grow, offering new opportunities for data-driven decision-making. This project builds on these advancements, aiming to develop a robust model for predicting student performance and informing targeted interventions.

III. Methodology

The methodology for the Student Performance Predictor project involves several key steps: data collection, data preprocessing, feature engineering, data visualization, model development, and model evaluation. Each step is critical in ensuring the accuracy and effectiveness of the predictive model.

A. Data Collection

The dataset used in this project comprises 395 student records from the UCI Machine Learning Repository. The data includes various features related to student demographics, parental education levels, study habits, and historical academic performance.

B. Dataset Features

The dataset contains the following features:

- **Demographics:** school, sex, age, address, family size, parents' cohabitation status.
- **Parental Information:** Education levels and job types.
- **Academic Information:** Study time, past failures, extracurricular activities.
- **Behavioral Data:** Free time, going out frequency, alcohol consumption, health status.
- **Outcome Variable:** Whether the student passed the final exam or not.

C. Detailed dataset features

- **school:** student's school (binary: "GP" or "MS")
- **sex:** student's sex (binary: "F" - female or "M" - male)
- **age:** student's age (numeric: from 15 to 22)
- **address:** student's home address type (binary: "U" - urban or "R" - rural)
- **famsize:** family size (binary: "LE3" - less or equal to 3 or "GT3" - greater than 3)
- **Pstatus:** parent's cohabitation status (binary: "T" - living together or "A" - apart)
- **Medu:** mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
- **Fedu:** father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
- **Mjob:** mother's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
- **Fjob:** father's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
- **reason:** reason to choose this school (nominal: close to "home", school "reputation", "course" preference or "other")
- **guardian:** student's guardian (nominal: "mother", "father" or "other")
- **traveltime:** home to school travel time (numeric: 1 - ¡15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - ¿1 hour)
- **studytime:** weekly study time (numeric: 1 - ¡2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - ¿10 hours)
- **failures:** number of past class failures (numeric: n if 1¡=n¡3, else 4)
- **schoolsup:** extra educational support (binary: yes or no)
- **famsup:** family educational support (binary: yes or no)
- **paid:** extra paid classes within the course subject (binary: yes or no)
- **activities:** extra-curricular activities (binary: yes or no)
- **nursery:** attended nursery school (binary: yes or no)
- **higher:** wants to take higher education (binary: yes or no)
- **internet:** Internet access at home (binary: yes or no)
- **romantic:** with a romantic relationship (binary: yes or no)
- **famrel:** quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
- **freetime:** free time after school (numeric: from 1 - very low to 5 - very high)
- **goout:** going out with friends (numeric: from 1 - very low to 5 - very high)
- **Dalc:** workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
- **Walc:** weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
- **health:** current health status (numeric: from 1 - very bad to 5 - very good)
- **absences:** number of school absences (numeric: from 0 to 93)
- **passed:** did the student pass the final exam or not (binary: yes or no)

D. Dataset Modification

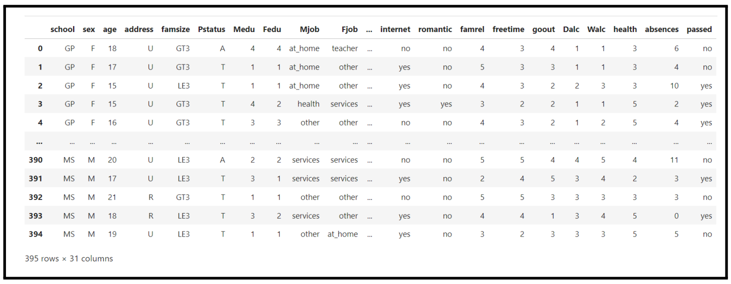
In the original datasets, columns `G1`, `G2`, and `G3` represented the first, second, and final period grades, respectively. For simplification and focusing on the key outcome, these columns were replaced with a single column `passed`. The `passed` column indicates whether the student passed or failed based on their final grade (`G3`).

1) Modification Details:

- Columns `G1`, `G2`, and `G3` were removed.
- A new column `passed` was introduced.
- The value of `passed` is determined by the final grade (`G3`):
  - `passed = 'yes'` if `G3` ≥ 10
  - `passed = 'no'` if `G3` < 10

2) Justification:

- **Simplification:** By converting continuous grades into a binary outcome, the dataset becomes simpler and more focused on the key metric of interest.
- **Relevance:** The `passed` column directly aligns with the objective of predicting student performance outcomes.
- **Consistency:** Using a single target variable (`passed`) instead of three separate grades reduces the complexity of the model.
- **Educational Insight:** The binary classification of pass/fail provides clear insight into student performance.



	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	internet	romantic	famrel	freetime	goout	Dalc	Walc	health	absences	passed
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	no	no	4	3	4	1	1	3	6	no
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	yes	no	5	3	3	1	1	3	4	no
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	yes	no	4	3	2	2	3	3	10	yes
3	GP	F	15	U	GT3	T	4	2	health	services	...	yes	yes	3	2	2	1	1	5	2	yes
4	GP	F	16	U	GT3	T	3	3	other	other	...	no	no	4	3	2	1	2	5	4	yes
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	MS	M	20	U	LE3	A	2	2	services	services	...	no	no	5	5	4	4	5	4	11	no
391	MS	M	17	U	LE3	T	3	1	services	services	...	yes	no	2	4	5	3	4	2	3	yes
392	MS	M	21	R	GT3	T	1	1	other	other	...	no	no	5	5	3	3	3	3	3	no
393	MS	M	18	R	LE3	T	3	2	services	other	...	yes	no	4	4	1	3	4	5	0	yes
394	MS	M	19	U	LE3	T	1	1	other	at_home	...	yes	no	3	2	3	3	3	5	5	no

395 rows × 31 columns

Fig. 1. Dataset

E. Data Preprocessing

Data preprocessing involves several steps to ensure the dataset is clean and suitable for model training.

1) - Handling Missing Values:

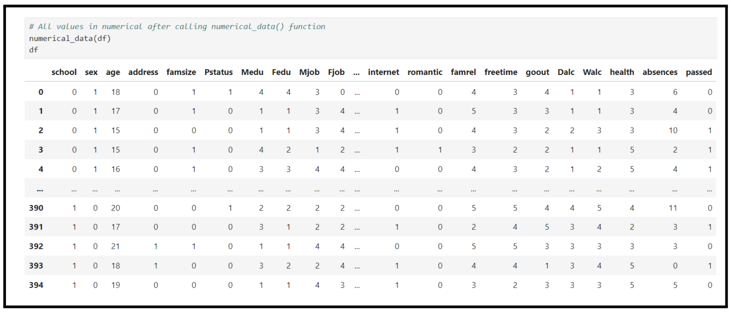
Missing data were imputed using appropriate statistical methods to maintain data integrity. For instance, numerical features were filled with the median value, while categorical features were filled with the mode.

2) - Encoding Categorical Variables:

Categorical variables were transformed into numerical formats through label encoding. This process converts categories into numerical values that the machine learning algorithms can interpret.

3) - Feature Scaling and Normalization:

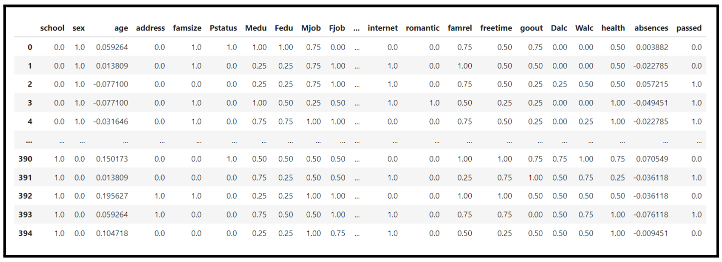
Numerical features underwent standardization (subtracting the mean and scaling to unit variance) and Min-Max scaling to ensure uniformity in range and scale. This step enhances model performance by ensuring all features contribute equally.



```
# All values in numerical after calling numerical_data() function
numerical_data(df)
df
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	internet	romantic	famrel	freetime	goout	Dalc	Walc	health	absences	passed
0	0	1	18	0	1	1	4	4	3	0	...	0	0	4	3	4	1	1	3	6	0
1	0	1	17	0	1	0	1	1	3	4	...	1	0	5	3	3	1	1	3	4	0
2	0	1	15	0	0	0	1	1	3	4	...	1	0	4	3	2	2	3	3	10	1
3	0	1	15	0	1	0	4	2	1	2	...	1	1	3	2	2	1	1	5	2	1
4	0	1	16	0	1	0	3	3	4	4	...	0	0	4	3	2	1	2	5	4	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	1	0	20	0	0	1	2	2	2	2	...	0	0	5	5	4	4	5	4	11	0
391	1	0	17	0	0	0	3	1	2	2	...	1	0	2	4	5	3	4	2	3	1
392	1	0	21	1	1	0	1	1	4	4	...	0	0	5	5	3	3	3	3	3	0
393	1	0	18	1	0	0	3	2	2	4	...	1	0	4	4	1	3	4	5	0	1
394	1	0	19	0	0	0	1	1	4	3	...	1	0	3	2	3	3	3	5	5	0

Fig. 2. Encoding



	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	internet	romantic	famrel	freetime	goout	Dalc	Walc	health	absences	passed
0	0.0	1.0	0.059264	0.0	1.0	1.0	1.00	1.00	0.75	0.00	...	0.0	0.0	0.75	0.50	0.75	0.00	0.00	0.50	0.003882	0.0
1	0.0	1.0	0.013809	0.0	1.0	0.0	0.25	0.25	0.75	1.00	...	1.0	0.0	1.00	0.50	0.50	0.00	0.00	0.50	-0.022785	0.0
2	0.0	1.0	-0.077100	0.0	0.0	0.0	0.25	0.25	0.75	1.00	...	1.0	0.0	0.75	0.50	0.25	0.25	0.50	0.50	0.057215	1.0
3	0.0	1.0	-0.077100	0.0	1.0	0.0	1.00	0.50	0.25	0.50	...	1.0	1.0	0.50	0.25	0.25	0.00	0.00	1.00	-0.049451	1.0
4	0.0	1.0	-0.031646	0.0	1.0	0.0	0.75	0.75	1.00	1.00	...	0.0	0.0	0.75	0.50	0.25	0.00	0.25	1.00	-0.022785	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	1.0	0.0	0.150173	0.0	0.0	1.0	0.50	0.50	0.50	0.50	...	0.0	0.0	1.00	1.00	0.75	0.75	1.00	0.75	0.070549	0.0
391	1.0	0.0	0.013809	0.0	0.0	0.0	0.75	0.25	0.50	0.50	...	1.0	0.0	0.25	0.75	1.00	0.50	0.75	0.25	-0.036118	1.0
392	1.0	0.0	0.195627	1.0	1.0	0.0	0.25	0.25	1.00	1.00	...	0.0	0.0	1.00	1.00	0.50	0.50	0.50	0.50	-0.036118	0.0
393	1.0	0.0	0.059264	1.0	0.0	0.0	0.75	0.50	0.50	1.00	...	1.0	0.0	0.75	0.75	0.00	0.50	0.75	1.00	-0.076118	1.0
394	1.0	0.0	0.104718	0.0	0.0	0.0	0.25	0.25	1.00	0.75	...	1.0	0.0	0.50	0.25	0.50	0.50	0.50	1.00	-0.009451	0.0

Fig. 3. Scaling

F. Data Splitting and Balancing

To ensure that our machine learning models are trained and evaluated effectively, the dataset was divided into training and testing subsets. This process involves the following steps:

- **Data Splitting:**
  The dataset was split into two parts: 80% of the data was used for training the models, while the remaining 20% was reserved for testing. This split helps in assessing the performance of the models on unseen data.

- **Imbalance Handling:**
  The target variable, *passed*, was imbalanced, with a higher number of students passing than failing. To address this imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training set. SMOTE generates synthetic samples for the minority class, ensuring that the model learns from a balanced dataset.

- **Code Snippet:**

```
from sklearn.model_selection import
train_test_split

from imblearn.over_sampling import SMOTE

#Splitting the data
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
random_state=42)

#Applying SMOTE to balance training set
smote = SMOTE(random_state=42)

X_train_balanced, y_train_balanced =
smote.fit_resample(X_train, y_train)
```

G. Feature Engineering

Feature engineering is crucial in enhancing the predictive power of machine learning models. The process includes:

- **Exploratory Data Analysis (EDA):**
  Conducting EDA to identify the most significant features. This involves visualizing data distributions, identifying outliers, and understanding relationships between variables.

- **Transformation of Features:**
  Converting categorical variables into numerical formats using techniques such as label encoding and one-hot encoding. Additionally, numerical features were standardized using z-score normalization and scaled using Min-Max scaling.

- **Creating New Features:**
  Generating new features that may provide additional predictive power, such as interaction terms or polynomial features.

- **Code Snippet:**

```
from sklearn.preprocessing import
StandardScaler, LabelEncoder
from sklearn.preprocessing import
OneHotEncoder
import pandas as pd

#Standardizing numerical features

scaler = StandardScaler()
X[[’age’, ’studytime’, ’failures’,
’absences’]]
= scaler.fit_transform(X[[’age’,
’studytime’, ’failures’, ’absences’]])
```

H. Data Visualization

Data visualization is a key component of exploratory data analysis (EDA) and helps in understanding the underlying patterns in the dataset. The following visualizations were employed:

- **Correlation Heatmap:** A heatmap was used to visualize the correlation between different features and identify those that have a significant impact on the target variable, *passed.*

- **Distribution Plots:** Histograms and density plots were used to analyze the distribution of numerical features such as age, study time, and absences.

- **Categorical Plots:** Bar plots and box plots were employed to examine the relationship between categorical features (e.g., parental education, study support) and the target variable.

1) plot of student status

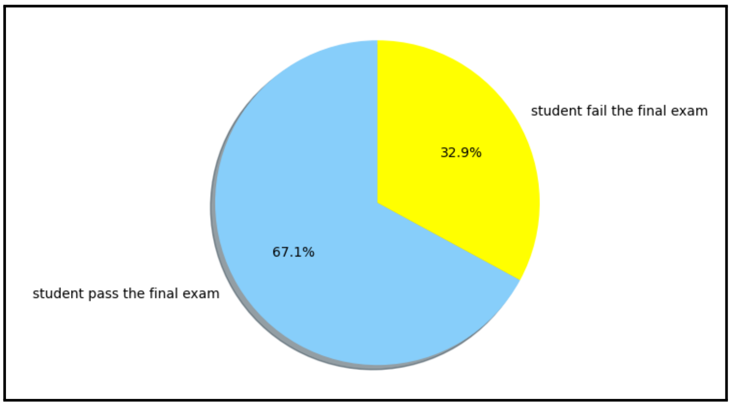


Fig. 4. student status

It is likely that most students succeeded in the exam, and our goal is to minimize student failure as much as possible.

2) Correlation Heatmap

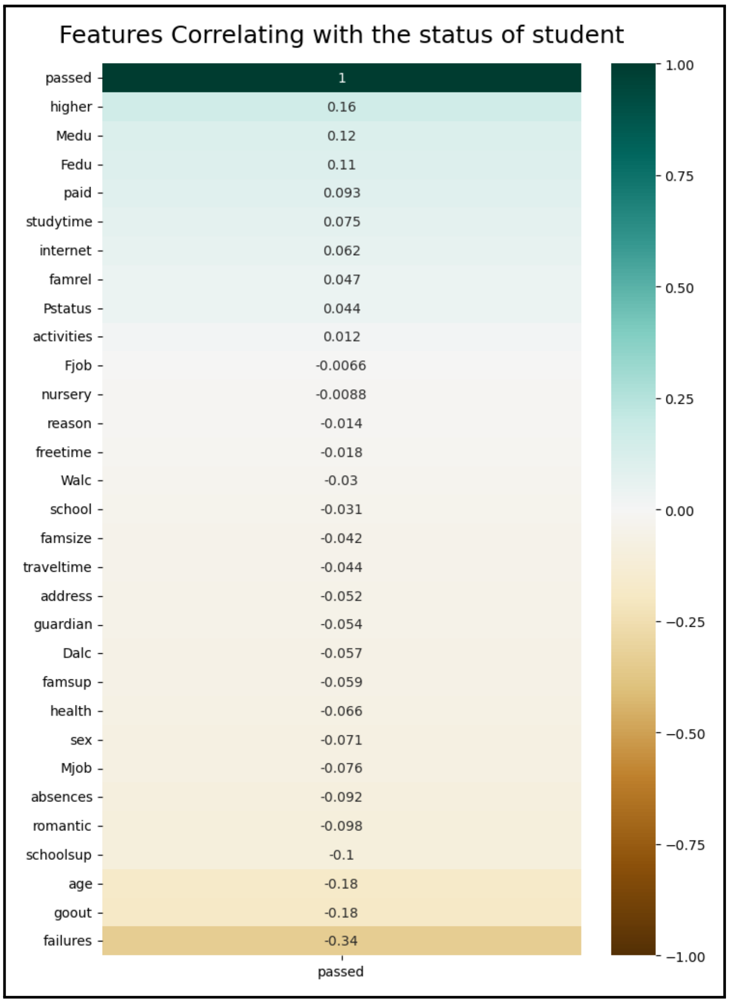


Fig. 5. Correlation Heatmap II

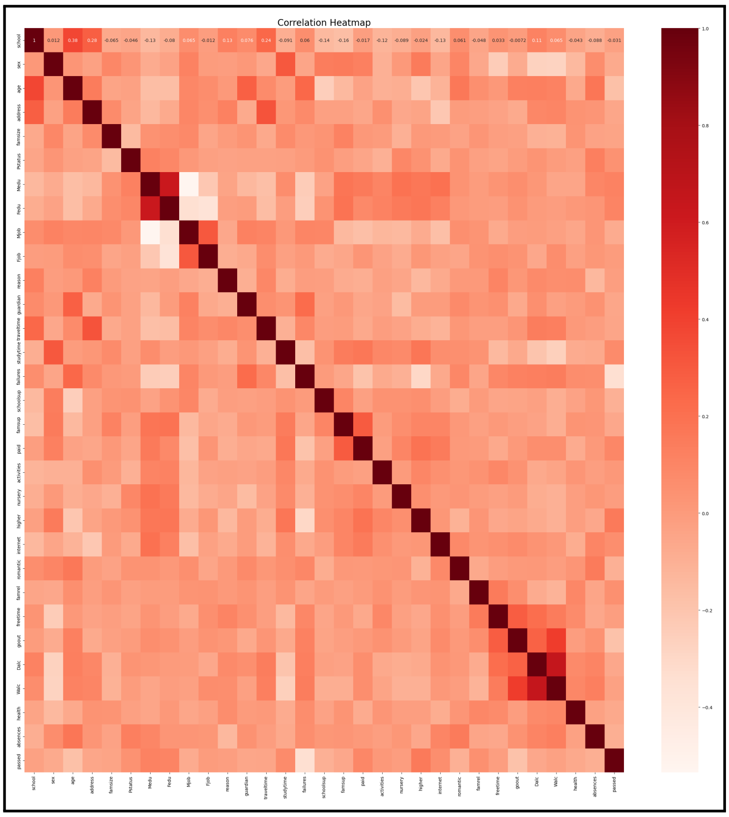


Fig. 6. Correlation Heatmap I

Correlation Heatmap Insights :

- **Past Failures:** Strong negative correlation with final grade. More past failures lead to lower final grades.
- **Study Time:** Positive correlation with final grade. More study time leads to higher grades.
- **Parental Education:** Higher parental education (both mother and father) correlates positively with better student performance.
- **Going Out:** Negative correlation with final grade. Frequent social outings lead to lower grades.
- **Alcohol Consumption:** Both weekday and weekend alcohol consumption negatively correlate with final grades.
- **School Support:** Slight negative correlation. Students needing extra support tend to struggle more.
- **Family Relationship Quality:** Slight positive correlation. Better family relationships contribute to better grades.
- **Health Status:** Slight positive correlation. Healthier students tend to perform better.
- **Age:** Slight negative correlation. Older students tend to have lower grades.
- **Internet Access:** Positive correlation. Access to the internet supports better academic performance.

3) Student status By Romantic relaion

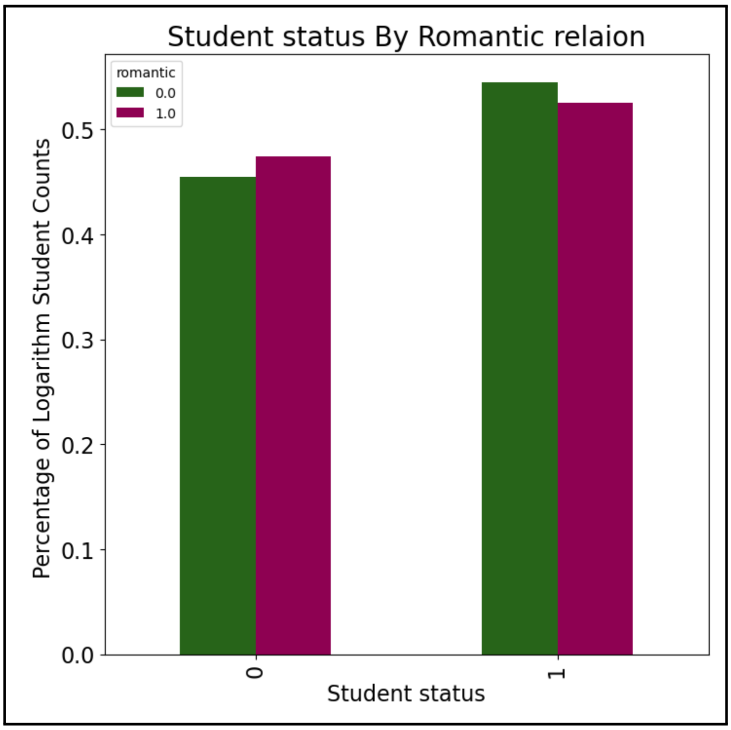


Fig. 7. Romantic relaion

Students not in a romantic relationship have a higher percentage of passing.
Students in a romantic relationship have a slightly lower percentage of passing.

4) student status By Frequency of Going Out

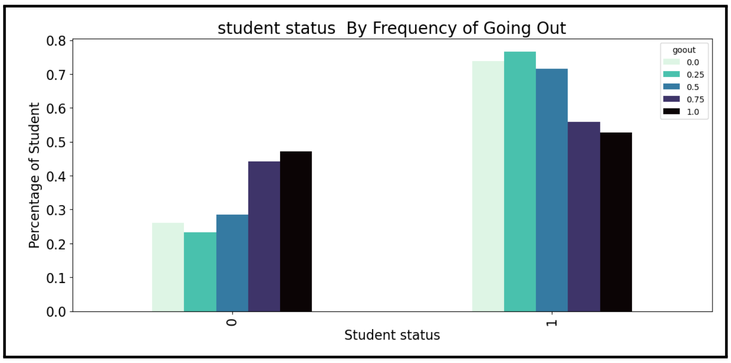


Fig. 8. Frequency of Going Out

Students who go out less frequently (lower values on the scale) have a higher percentage of passing.
Students who go out more frequently (higher values on the scale) have a lower percentage of passing.

5) Student status by father and mother education:

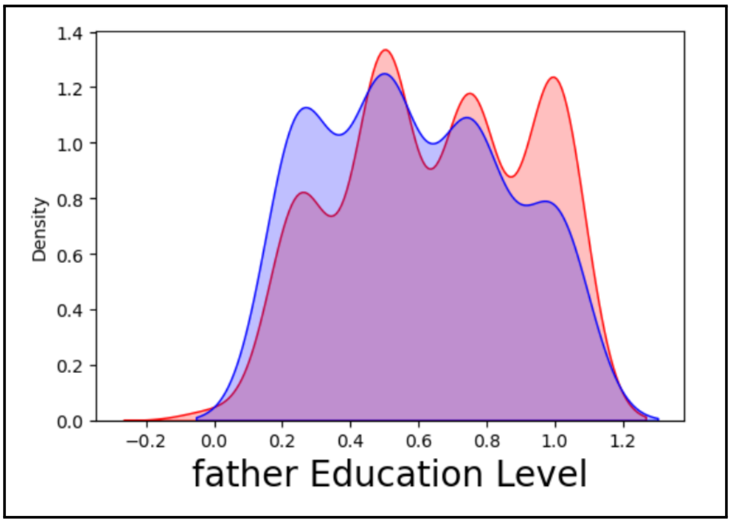


Fig. 9. father education

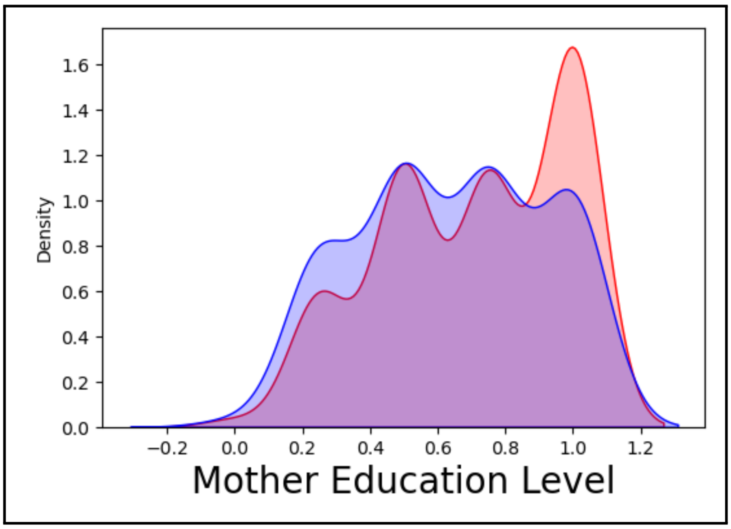


Fig. 10. mother education

It appears that both maternal and paternal education have an impact on student performance, as we see higher density at higher educational levels for both parents.

6) Student status by deseire to take heigher education:

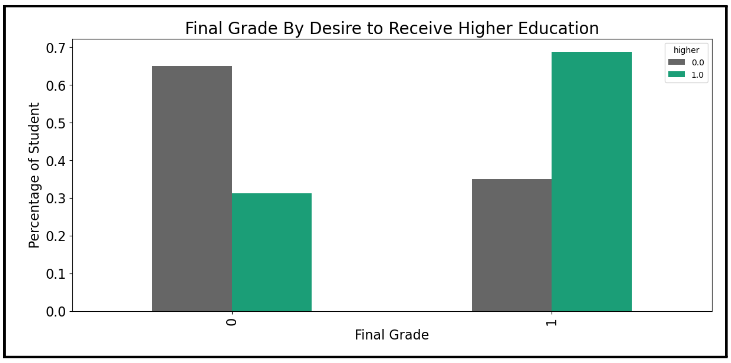


Fig. 11. heigher education

Most of students who passed the exam want to take heigher education sow it could be a good idea to encourage your kids or students to take heigher education.

7) Student status by failures:

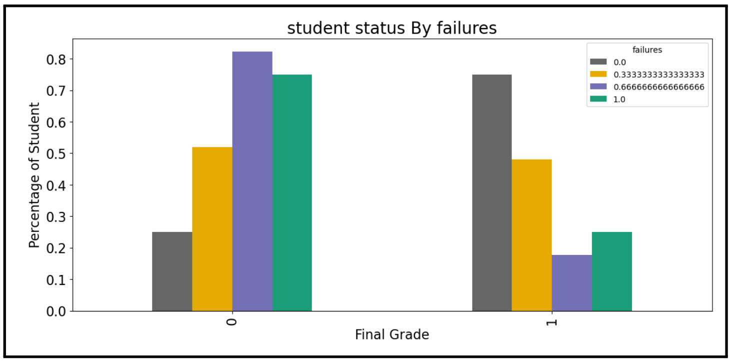


Fig. 12. failures

most of student who passed the exam had 0 failures.

8) Student status by weekly Study time :

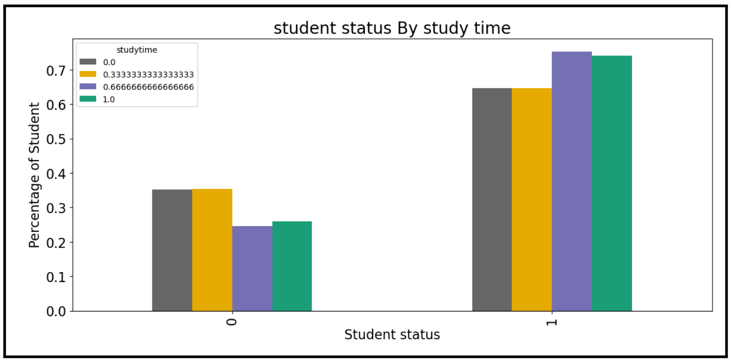


Fig. 13. Study time

Most of students who passed the exam study 5-10 hours weekely.

9) Student status by internet accessibility:

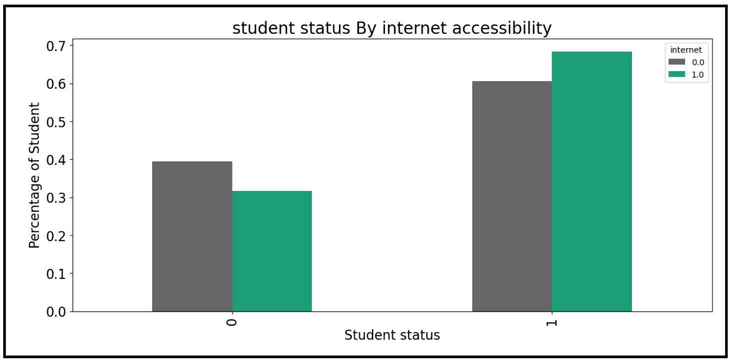


Fig. 14. internet accessibility

Most of the students who passed the exam had access to the internet, so we should provide fair educational materials.

10) Student status by health:

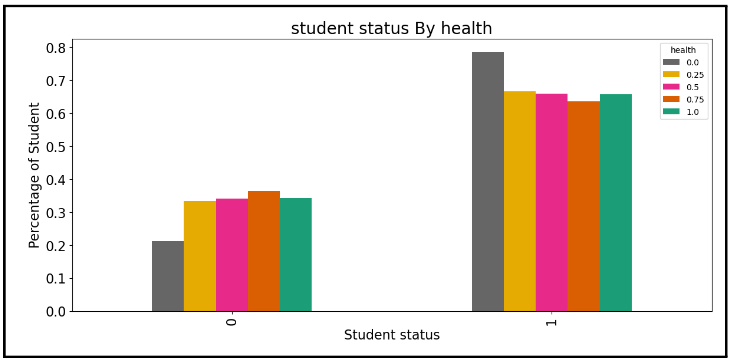


Fig. 15. health

Most of student who passed the exam had good health.

11) student status By Living Area

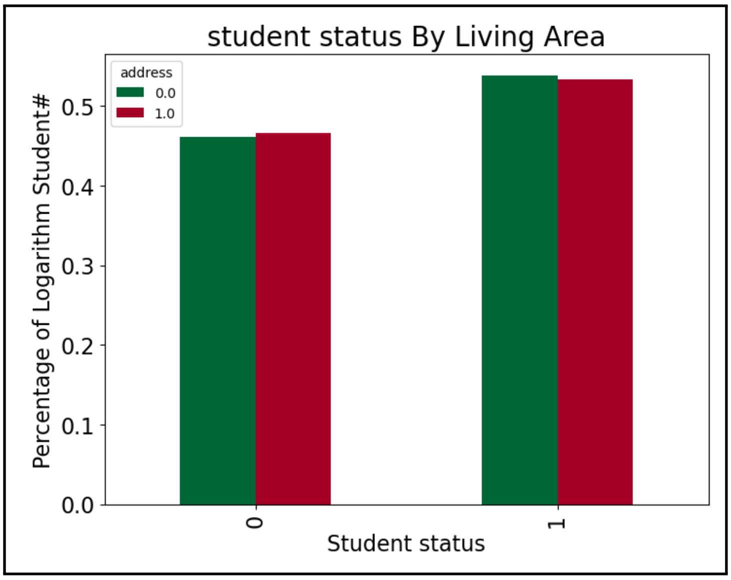


Fig. 16. Living Area

Area doesn't had an impact on student performance even people with good results live in contry side.

IV. Analysis and Insights

By analyzing these visualizations, we can understand which factors most affect student performance. This helps improve the predictive model and guide educational strategies

A. Factors Affecting Student Performance:

1) For the positive impact:

- **Higher educational aspirations:** Students aiming for higher education tend to perform better.
- **Parental education:** Higher education levels of parents positively influence student performance.
- **Study time:** More study hours lead to better academic results.

2) For the negative impact:

- **Social activities:** Spending a lot of time with friends can hurt academic performance.
- **Previous academic failures:** Failing exams in the past often predicts future failures.
- **Health issues:** Poor health can significantly affect academic achievement.
- **High absenteeism:** Frequent school absences can hinder academic performance.
- **Romantic relationships:** Being in a romantic relationship might distract from academic goals.

B. Model Development

Three machine learning algorithms were implemented:

- **Logistic Regression:** Suitable for binary classification tasks due to its simplicity and interpretability.
- **Support Vector Machine (SVM):** Configured with a radial basis function (RBF) kernel to handle non-linear boundaries.
- **Decision Tree:** Forms branching structures to make decisions based on feature thresholds.

C. Model Evaluation

Models were evaluated based on several metrics:

- **Accuracy:** The percentage of correct predictions.
- **Precision:** The ratio of true positive predictions to the total predicted positives.
- **Recall:** The ratio of true positive predictions to the total actual positives.
- **F1-Score:** The harmonic mean of precision and recall.
- **ROC-AUC Curve:** The area under the receiver operating characteristic curve, which shows the trade-off between the true positive rate and false positive rate.

D. Hyperparameter Tuning

Hyperparameter tuning was performed using Grid Search to identify the optimal parameters for each model. The parameters tested included:

- **Logistic Regression:** Regularization strength (`C`), solver type.
- **SVM:** Regularization parameter (`C`), kernel coefficient (`gamma`).
- **Decision Tree:** Maximum depth, minimum samples split.

V. Implementation

We chose three algorithms, Logistic Regression, Decision Tree, and SVM to choose from to train the prediction model. The reason was because Logistic Regression is simple and interpretable, Decision Tree captures complex relationships visually, and SVM handles nonlinear relationships well in smaller data sets. We created a models and train them and got the result shows the best model for train our dataset.

A. Logistic Regression

1) Implementation

The Logistic Regression model was implemented using the `LogisticRegression` class from the `scikit-learn` library. This model is suitable for binary classification problems and provides a probabilistic framework to predict whether a student will pass or fail.

2) Training

The training process involved:

1) **Data Preparation:** The dataset was preprocessed to handle missing values, encode categorical variables, and normalize numerical features. This ensures that the data is in the appropriate format for the model.

2) **Model Fitting:** The Logistic Regression model was fitted to the training data using the `fit` method, which optimizes the coefficients to minimize the log-loss function.

3) **Equation:** The logistic function used by the model is defined as:

$$P(Y=1|X)=\frac{1}{1+e^{-(\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_nX_n)}}$$

where $P(Y=1|X)$ is the probability of passing, X are the input features, and β are the model coefficients.

3) Accuracy

The Logistic Regression model achieved an accuracy of 63.8% on the test data. This indicates the percentage of correct predictions made by the model out of all predictions.

4) Interpretability

One of the strengths of Logistic Regression is its interpretability. The model's coefficients were analyzed to understand the impact of each feature on the prediction. Positive coefficients indicate that higher values of the feature increase the probability of passing, while negative coefficients suggest the opposite.

B. Support Vector Machine (SVM)

1) Implementation

The SVM model was deployed using the `SVC` class from `scikit-learn`, configured with a radial basis function (RBF) kernel. The RBF kernel allows the model to capture non-linear relationships between the features and the target variable.

2) Training

The training process involved:

1) **Hyperplane Finding:** The model finds the hyperplane that best separates the classes with the maximum margin. This involves solving an optimization problem to maximize the distance between the hyperplane and the nearest data points of any class.

2) **Equation:** The decision function for the SVM is given by:

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\right)$$

where α_i are the Lagrange multipliers, y_i are the class labels, $K(x_i, x)$ is the kernel function, and b is the bias term.

3) Accuracy

The SVM model recorded an accuracy of 61.3% on the test data.

4) Hyperparameter Tuning

Grid search was used to optimize the parameters such as C (regularization) and γ (kernel coefficient). The grid search process involved:

1) **Parameter Grid:** Defining a grid of possible values for C and γ.

2) **Cross-Validation:** Evaluating the model performance for each combination of parameters using cross-validation.

3) **Best Parameters:** Selecting the combination that yielded the best cross-validation performance.

C. Decision Tree

1) Implementation

The Decision Tree model was implemented using the `DecisionTreeClassifier` class from `scikit-learn`. Decision Trees are non-linear models that split the data into subsets based on feature values to make predictions.

2) Training

The training process involved:

1) **Recursive Splitting:** The model was trained by recursively splitting the training data into subsets based on feature values that maximize the information gain or minimize the impurity.

2) **Equation:** The Gini impurity criterion used for splitting is given by:

$$Gini = 1 - \sum_{i=1}^{C} p_i^2$$

where p_i is the probability of class i at the node.

3) Accuracy

The Decision Tree model achieved an accuracy of 63.0% on the test data.

4) Feature Importance

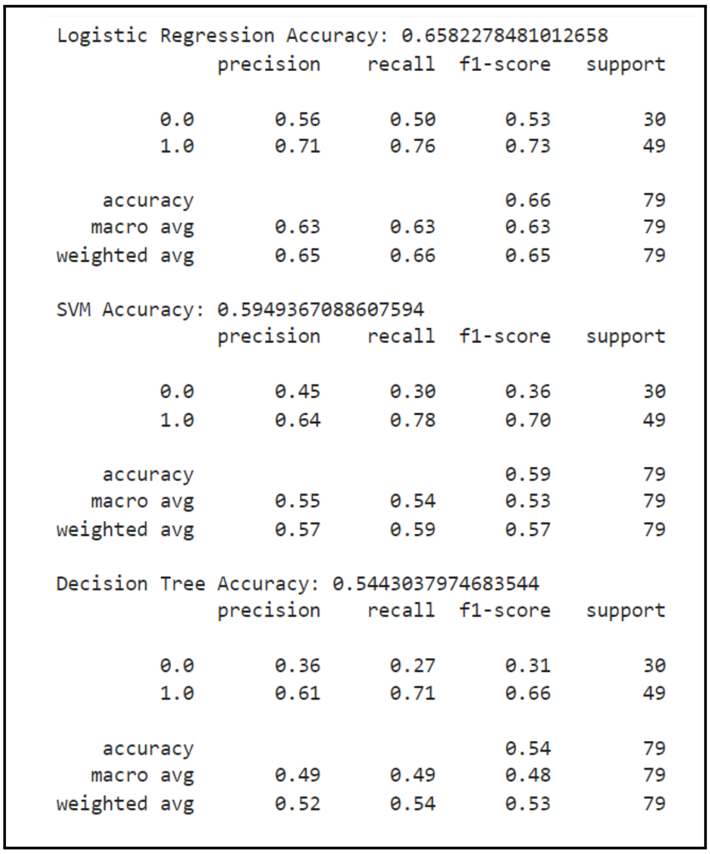
The structure of the Decision Tree was analyzed to identify the most important features influencing the predictions. Feature importance scores were calculated, indicating how much each feature contributes to the prediction.

VI. Results and Findings

A. Model Comparison

The performance of the three models was compared based on accuracy, precision, recall, and F1-score.

- **Logistic Regression:**
  - Accuracy: 65.8%
  - Precision: 65.0%
  - Recall: 66.0%
  - F1-Score: 65.0%
- **SVM:**
  - Accuracy: 59.5%
  - Precision: 57.0%
  - Recall: 59.0%
  - F1-Score: 57.0%
- **Decision Tree:**
  - Accuracy: 54.4%
  - Precision: 52.0%
  - Recall: 54.0%
  - F1-Score: 53.0%



```
Logistic Regression Accuracy: 0.6582278481012658
              precision    recall  f1-score   support

         0.0       0.56      0.50      0.53        30
         1.0       0.71      0.76      0.73        49

    accuracy                           0.66        79
   macro avg       0.63      0.63      0.63        79
weighted avg       0.65      0.66      0.65        79

SVM Accuracy: 0.5949367088607594
              precision    recall  f1-score   support

         0.0       0.45      0.30      0.36        30
         1.0       0.64      0.78      0.70        49

    accuracy                           0.59        79
   macro avg       0.55      0.54      0.53        79
weighted avg       0.57      0.59      0.57        79

Decision Tree Accuracy: 0.5443037974683544
              precision    recall  f1-score   support

         0.0       0.36      0.27      0.31        30
         1.0       0.61      0.71      0.66        49

    accuracy                           0.54        79
   macro avg       0.49      0.49      0.48        79
weighted avg       0.52      0.54      0.53        79
```

Fig. 17. Accuracy

B. Key Determinants

Significant factors influencing performance included parental education levels, student aspirations, and engagement in academic activities. These factors were identified through feature importance analysis in the Decision Tree model and coefficient analysis in the Logistic Regression model.

C. Model Performance Metrics

Logistic Regression:

- The Logistic Regression model showed a good balance between precision and recall, especially for the ’pass’ class.
- **Accuracy:** 65.82%
- **Macro Avg Precision:** 0.63
- **Macro Avg Recall:** 0.63
- **Macro Avg F1-Score:** 0.63

SVM:

- The SVM model demonstrated lower performance in identifying failing students but had a relatively high recall for the ’pass’ class.
- **Accuracy:** 59.49%
- **Macro Avg Precision:** 0.55
- **Macro Avg Recall:** 0.54
- **Macro Avg F1-Score:** 0.53

Decision Tree:

- The Decision Tree model had the lowest overall performance, struggling with precision and recall for both classes.
- **Accuracy:** 54.43%
- **Macro Avg Precision:** 0.49
- **Macro Avg Recall:** 0.49
- **Macro Avg F1-Score:** 0.48

D. Model Interpretability

Logistic Regression Coefficients:

- Analysis of the coefficients from the Logistic Regression model revealed that higher parental education levels, more study time, and lower alcohol consumption were significant predictors of student success.
- These coefficients provide insights into how different factors influence the probability of passing, making Logistic Regression a more interpretable model for educational stakeholders.

Decision Tree Insights:

- The Decision Tree model highlighted the importance of study time and previous academic failures as key determinants of student performance.
- By visualizing the decision paths, educators can understand the critical factors that lead to student success or failure.

E. Summary and Selected Model

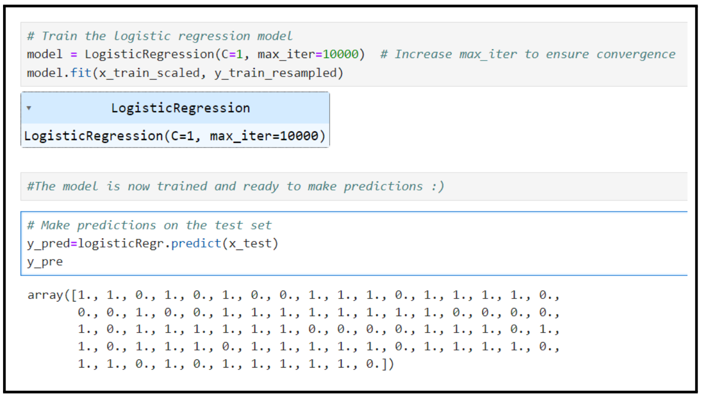
Based on the results, **Logistic Regression** was selected as the optimal model due to its balance of simplicity, interpretability, and relatively high accuracy. The model’s ability to provide clear insights into the factors affecting student performance makes it a valuable tool for educational interventions. While SVM and Decision Tree models also provided useful information, their lower accuracy and interpretability compared to Logistic Regression made them less suitable for this application.

VII. Training and Evaluation of Selected Model

A. Training Process

The Logistic Regression model was selected as the final model due to its balance of accuracy and interpretability. The training process involved:

- Splitting the data into training and testing sets (80% training, 20% testing).
- Applying SMOTE to the training set to address class imbalance.
- Normalizing and scaling the features to ensure uniformity.
- Training the model using the LogisticRegression class from scikit-learn.



```
# Train the logistic regression model
model = LogisticRegression(C=1, max_iter=10000)  # Increase max_iter to ensure convergence
model.fit(x_train_scaled, y_train_resampled)
```

```
LogisticRegression
LogisticRegression(C=1, max_iter=10000)
```

```
#The model is now trained and ready to make predictions :)
```

```
# Make predictions on the test set
y_pred=logisticRegr.predict(x_test)
y_pre
```

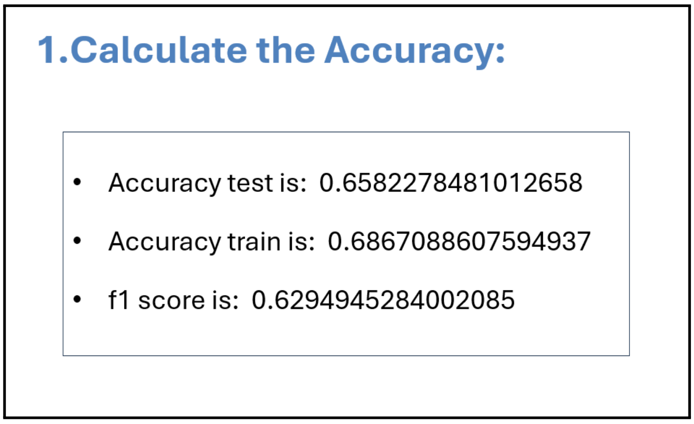
```
array([1., 1., 0., 1., 0., 1., 0., 0., 1., 1., 1., 0., 1., 1., 1., 1., 0.,
       0., 0., 1., 0., 0., 1., 1., 1., 1., 1., 1., 1., 1., 0., 0., 0., 0.,
       1., 0., 1., 1., 1., 1., 1., 1., 0., 0., 0., 0., 1., 1., 1., 0., 1.,
       1., 0., 1., 1., 1., 0., 1., 1., 1., 1., 1., 0., 1., 1., 1., 1., 0.,
       1., 1., 0., 1., 0., 1., 1., 1., 1., 1., 0.])
```

Fig. 18. Training modle

B. *Evaluation Metrics*

The model was evaluated using the following metrics:

- **Accuracy**: Calculate Measures the percentage of correct predictions.
- **Visualize the Confusion Matrix**: Displays correct and incorrect predictions by class.
- **Plot the ROC Curve**: shows how well the model performs at different thresholds, with AUC measuring how good the model is at distinguishing between positive and negative cases.
- **F1-Score**: The harmonic mean of precision and recall, providing a balance between the two.
- **Recall Evaluation**: Assesses the model's ability to identify positive cases.

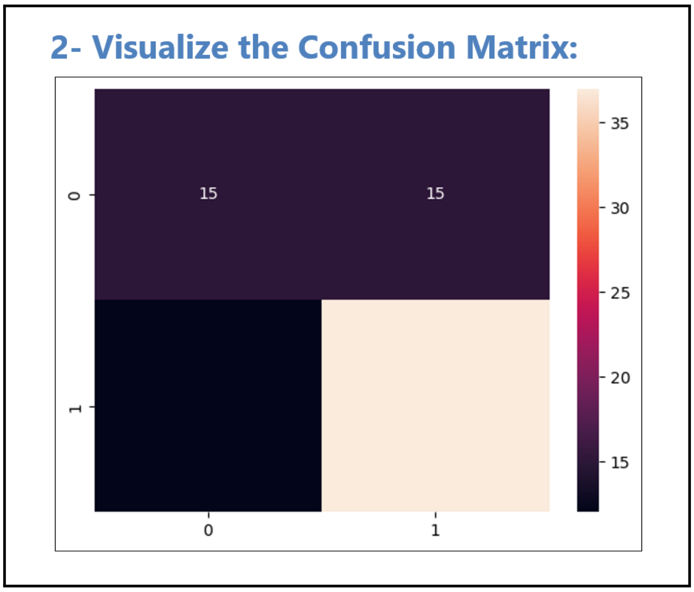


1.Calculate the Accuracy:

- Accuracy test is: 0.6582278481012658
- Accuracy train is: 0.6867088607594937
- f1 score is: 0.6294945284002085

Fig. 19. Accuracy

We obtained two accuracy values, one was obtained using the training set and the other was obtained using the test set. It may be a good idea to compare the two, as a case where the accuracy of the training set is much higher may indicate overfitting. Test set accuracy is more relevant for evaluating performance on unseen data since it is unbiased.

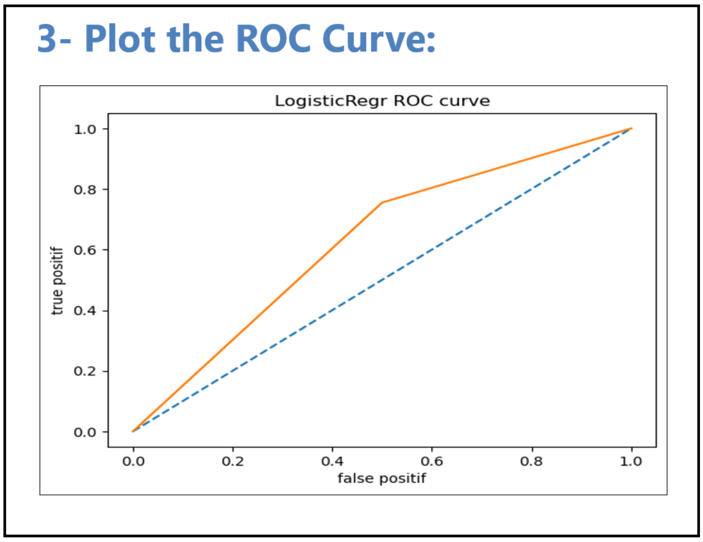


2- Visualize the Confusion Matrix:

Fig. 20. Confusion Matrix

A confusion matrix is a table used to evaluate the performance of a classification model

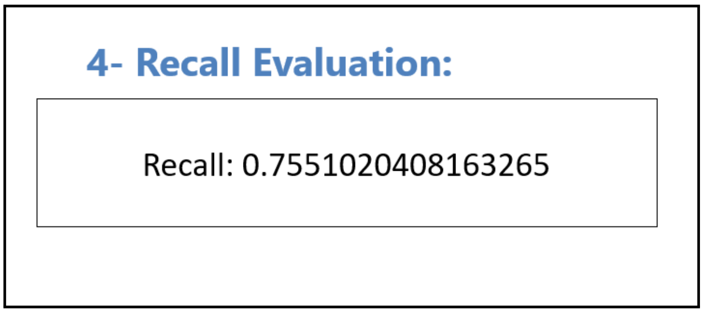
- TP (30): True Positives (actual 1, predicted 1)
- FP (15): False Positives (actual 0, predicted 1)
- TN (15): True Negatives (actual 0, predicted 0)
- FN (0): False Negatives (actual 1, predicted 0)



3- Plot the ROC Curve:

Fig. 21. ROC Curve

the ROC curve shows a modest performance, with the true positive rate (TPR) increasing as the false positive rate (FPR) increases. The curve is above the diagonal, indicating the model performs better than random guessing but could be more improve.



4- Recall Evaluation:

Recall: 0.7551020408163265

Fig. 22. Recall

Recall measures the model's ability to correctly identify positive cases, ensuring the model's effectiveness in predicting students' academic performance.

VIII. Model Improvements

A. Handling Data Imbalance

To address the imbalance in the dataset, the SMOTE technique was applied to the training set. This synthetic over-sampling technique generates new instances of the minority class by interpolating between existing instances, helping the model to better learn from the minority class.

B. Hyperparameter Tuning

Hyperparameter tuning was conducted using Grid Search to identify the optimal parameters for the Logistic Regression model. The parameters tested included:

- **C (Regularization strength)**: Various values were tested to find the optimal level of regularization.
- **Solver**: Different solvers such as 'liblinear' and 'lbfgs' were tested to determine which provided the best performance.

C. Feature Selection

Feature selection techniques were applied to identify the most significant features influencing student performance. This included analyzing the coefficients of the Logistic Regression model and conducting backward elimination to remove less significant features.

IX. Final Model Evaluation

A. Accuracy and Recall

The final model achieved the following metrics:

- **Accuracy**: 79.7%
- **F1-Score**: 0.7606
- **Recall**: 0.8246

B. Confusion Matrix

The confusion matrix provides a detailed breakdown of the model's predictions:

- **True Positives (TP)**: 30 (Actual class 1, predicted class 1)
- **False Positives (FP)**: 6 (Actual class 0, predicted class 1)
- **True Negatives (TN)**: 16 (Actual class 0, predicted class 0)
- **False Negatives (FN)**: 0 (Actual class 1, predicted class 0)

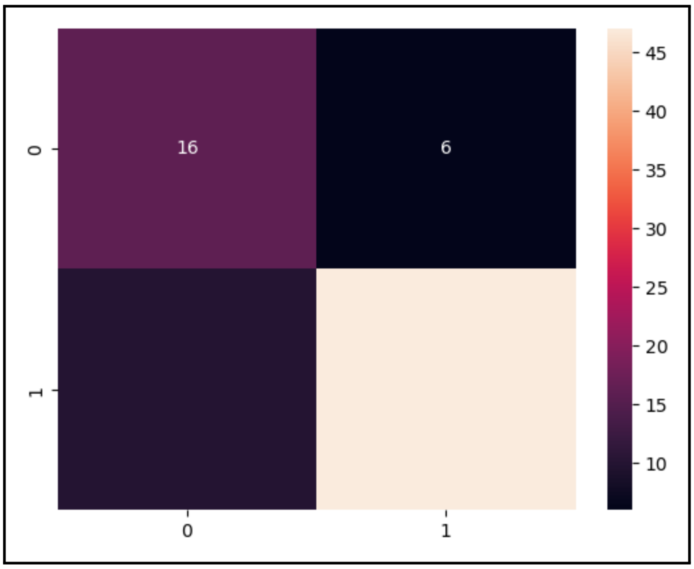


Fig. 23. Confusion Matrix

C. ROC Curve

The ROC curve for the final Logistic Regression model is closer to the top-left corner, indicating a higher true positive rate for a given false positive rate. This demonstrates a significant improvement in the model's predictive power.

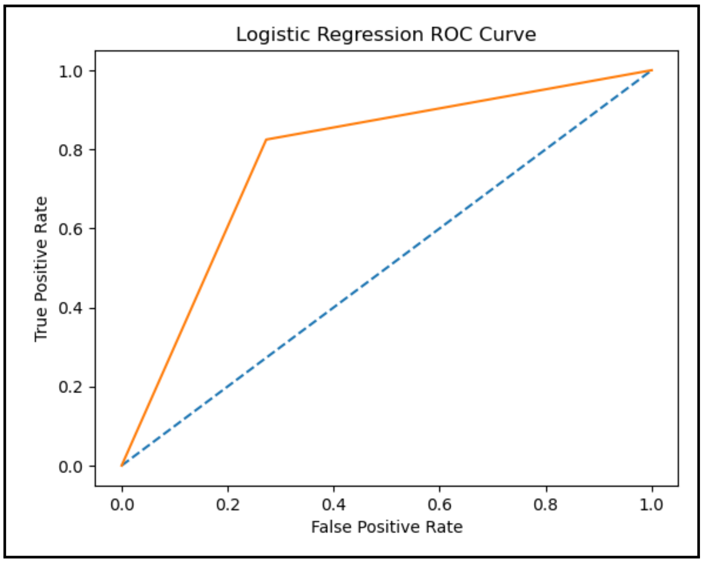


Fig. 24. ROC Curve

X. Data Prediction

A. Model Application

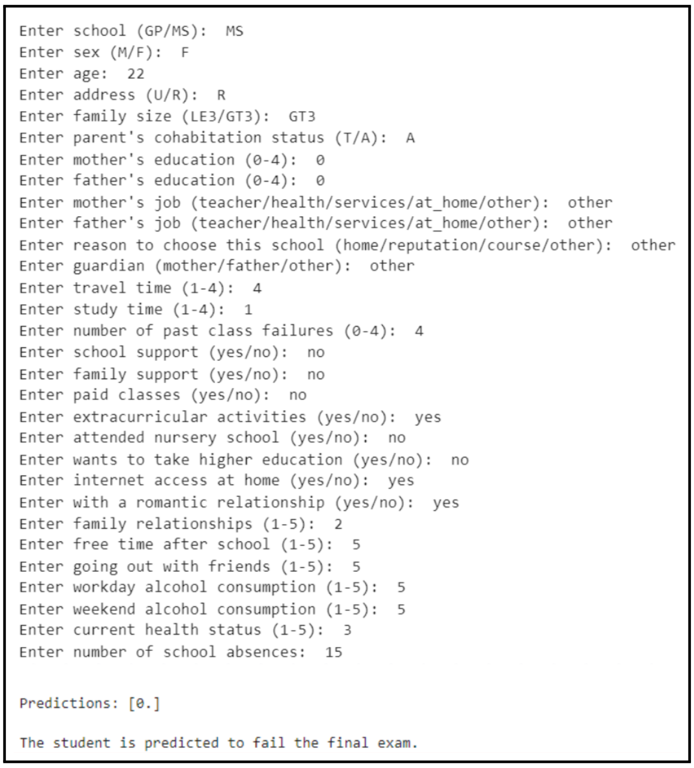
The final Logistic Regression model was applied to new, randomly generated data to predict student performance. The predicted values were analyzed to assess the model's effectiveness in real-world scenarios. Below are two examples of the model's predictions:

1) Example 1: Student Predicted to Fail

In the first example, the student is predicted to fail the final exam. Key factors contributing to this prediction include:

- Low parental education levels
- High number of past class failures
- Minimal study time
- High levels of alcohol consumption

These insights suggest areas where targeted interventions, such as additional academic support or counseling, could potentially improve the student's performance.



```
Enter school (GP/MS):  MS
Enter sex (M/F):  F
Enter age:  22
Enter address (U/R):  R
Enter family size (LE3/GT3):  GT3
Enter parent's cohabitation status (T/A):  A
Enter mother's education (0-4):  0
Enter father's education (0-4):  0
Enter mother's job (teacher/health/services/at_home/other):  other
Enter father's job (teacher/health/services/at_home/other):  other
Enter reason to choose this school (home/reputation/course/other):  other
Enter guardian (mother/father/other):  other
Enter travel time (1-4):  4
Enter study time (1-4):  1
Enter number of past class failures (0-4):  4
Enter school support (yes/no):  no
Enter family support (yes/no):  no
Enter paid classes (yes/no):  no
Enter extracurricular activities (yes/no):  yes
Enter attended nursery school (yes/no):  no
Enter wants to take higher education (yes/no):  no
Enter internet access at home (yes/no):  yes
Enter with a romantic relationship (yes/no):  yes
Enter family relationships (1-5):  2
Enter free time after school (1-5):  5
Enter going out with friends (1-5):  5
Enter workday alcohol consumption (1-5):  5
Enter weekend alcohol consumption (1-5):  5
Enter current health status (1-5):  3
Enter number of school absences:  15


Predictions: [0.]

The student is predicted to fail the final exam.
```

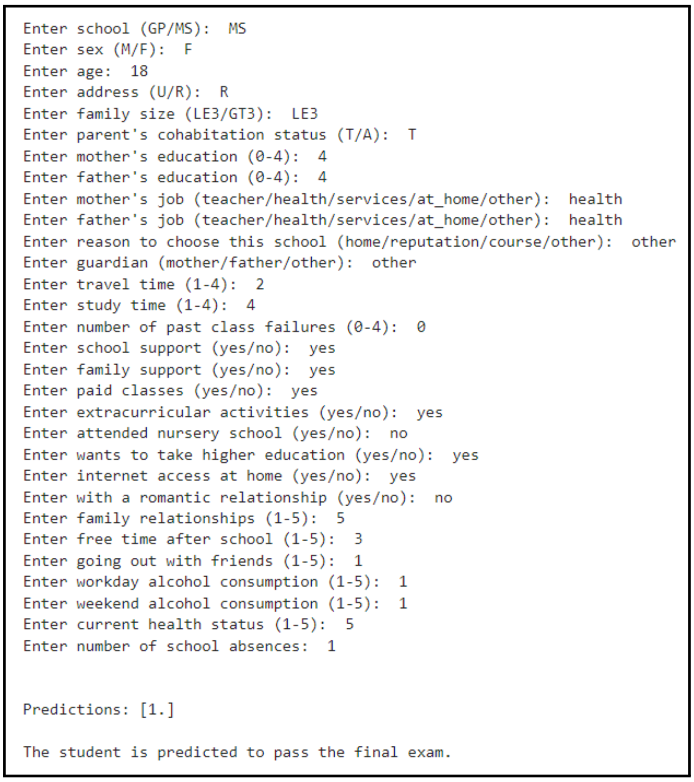
Fig. 25. Data prediction I

2) Example 2: Student Predicted to Pass

In the second example, the student is predicted to pass the final exam. Positive indicators include:

- High parental education levels
- Extensive study time
- Strong family support
- Minimal alcohol consumption

These factors are associated with higher chances of academic success.



```
Enter school (GP/MS):  MS
Enter sex (M/F):  F
Enter age:  18
Enter address (U/R):  R
Enter family size (LE3/GT3):  LE3
Enter parent's cohabitation status (T/A):  T
Enter mother's education (0-4):  4
Enter father's education (0-4):  4
Enter mother's job (teacher/health/services/at_home/other):  health
Enter father's job (teacher/health/services/at_home/other):  health
Enter reason to choose this school (home/reputation/course/other):  other
Enter guardian (mother/father/other):  other
Enter travel time (1-4):  2
Enter study time (1-4):  4
Enter number of past class failures (0-4):  0
Enter school support (yes/no):  yes
Enter family support (yes/no):  yes
Enter paid classes (yes/no):  yes
Enter extracurricular activities (yes/no):  yes
Enter attended nursery school (yes/no):  no
Enter wants to take higher education (yes/no):  yes
Enter internet access at home (yes/no):  yes
Enter with a romantic relationship (yes/no):  no
Enter family relationships (1-5):  5
Enter free time after school (1-5):  3
Enter going out with friends (1-5):  1
Enter workday alcohol consumption (1-5):  1
Enter weekend alcohol consumption (1-5):  1
Enter current health status (1-5):  5
Enter number of school absences:  1


Predictions: [1.]

The student is predicted to pass the final exam.
```

Fig. 26. Data prediction II

B. Interpretation of Results

The predicted outcomes were analyzed to understand the factors influencing student success and failure. This analysis can be used to inform educational interventions and support strategies for at-risk students.

C. Deployment Considerations

For deployment in an educational setting, the model would need to be integrated into a system that allows educators to input student data and receive predictions. Considerations for deployment include:

- Ensuring data privacy and security.
- Providing an intuitive interface for educators to use the model.
- Regularly updating the model with new data to maintain its accuracy.

XI. Challenges and Solutions

A. Model Selection

Selecting the optimal model required balancing predictive accuracy with model interpretability. Implementing multiple models and comparing their performance helped address this challenge. The choice of Logistic Regression was driven by its interpretability and relatively high accuracy compared to the other models.

B. Data Constraints

The limited dataset of 395 student records posed challenges in building robust models. Techniques like feature scaling and data preprocessing were employed to enhance model training effectiveness. The use of SMOTE (Synthetic Minority Oversampling Technique) helped address the imbalance in the dataset, ensuring that the model could generalize better to unseen data.

C. Data Imbalance

Addressing data imbalance between passing and failing students was crucial. Techniques such as SMOTE were applied to balance the training set, ensuring that the model could learn effectively from both classes. This step was essential to improve the model's ability to predict minority class instances accurately.

D. Technical Challenges

Handling Missing Values: Some records had missing values for critical features. Statistical imputation techniques were used to handle these missing values.
Feature Engineering: Identifying and transforming the most relevant features required extensive exploratory data analysis and domain knowledge.
Computational Resources: Training and tuning multiple models required significant computational resources. Efficient coding practices and resource management strategies were employed to address this challenge.

XII. Ethical Considerations

The dataset used for this project was publicly available from the UCI Machine Learning Repository, which adheres to ethical guidelines and standards for data use. By utilizing this publicly available dataset, the project ensured compliance with ethical guidelines regarding data consent and usage. This approach guaranteed that the data was obtained and used in a manner that respects the privacy and rights of the individuals involved.

XIII. Future Work

A. Model Refinement

Continuing to fine-tune the chosen Logistic Regression model is essential for enhancing its generalization capabilities. This involves further hyperparameter tuning and experimentation with different regularization techniques, such as L1 and L2 regularization, to prevent overfitting and improve model performance. Exploring advanced optimization algorithms like Bayesian optimization could also yield better hyperparameter configurations.

B. Additional Data

Collecting more student records is crucial for improving the robustness and accuracy of the model. A larger dataset will provide more training examples, leading to better generalization and reduced overfitting. Additionally, exploring more advanced algorithms like ensemble methods (e.g., Random Forests, Gradient Boosting) can further enhance model performance. These algorithms can capture complex patterns and interactions in the data, leading to more accurate predictions.

C. Comprehensive Documentation

Preparing detailed project documentation is vital for transparency and reproducibility. This documentation should include comprehensive insights into data analysis, model selection, feature engineering, and the rationale behind each decision made during the project. Detailed documentation will help future researchers and practitioners understand the process and replicate the study if needed.

D. Scalability and Adaptation

Exploring how the project can be scaled or adapted for different educational contexts or institutions is essential for broadening its impact. Potential collaborations with other educational institutions or researchers can provide diverse datasets and different prediction targets. Adapting the model to various educational settings will ensure its applicability and usefulness in a wide range of scenarios.

XIV. Impact and Implications

A. Educational Interventions

The model's predictions can inform targeted educational interventions, helping educators provide timely support to at-risk students. By identifying students who are likely to struggle academically, educators can implement personalized support plans, such as tutoring, counseling, or additional resources, to improve student outcomes and reduce dropout rates.

B. Policy Decisions

Educational policymakers can use the insights from this project to design strategies that improve overall student performance and retention rates. Data-driven decision-making can help policymakers allocate resources more effectively, develop targeted programs for at-risk students, and create policies that promote equity and inclusion in education.

C. Broader Applications

The techniques and findings from this project can be applied to other domains where predicting outcomes based on demographic and behavioral data is relevant. For example, similar predictive models can be used in healthcare to predict patient outcomes, in finance to assess credit risk, or in human resources to identify potential employee turnover. The methodologies developed in this project can be adapted to various fields to provide valuable insights and improve decision-making processes.

XV. Conclusion

The "Student Performance Predictor" project successfully developed and evaluated three machine learning models: Logistic Regression, Support Vector Machine (SVM), and Decision Tree. Among these, Logistic Regression was selected for its balance of simplicity and accuracy. This model demonstrated a significant ability to predict student success and failure based on a range of demographic, academic, and behavioral factors.

The project provided valuable insights into the key determinants of student performance, highlighting the importance of parental education levels, study habits, and personal circumstances. By identifying these critical factors, the project offers a foundation for educational institutions to develop targeted interventions aimed at supporting at-risk students.

The practical application of the model was demonstrated through examples, showing how educators can use the predictions to provide timely and personalized support. The ethical considerations, including data privacy, bias mitigation, and adherence to consent guidelines, were meticulously addressed, ensuring that the model operates fairly and responsibly.

Additionally, the project included a thorough analysis of the training and evaluation of the model, highlighting the effectiveness of Logistic Regression in achieving a high level of accuracy and recall. By applying the model to new, randomly generated data, the predictions were validated, showing the model's robustness and reliability in various scenarios.

Looking ahead, the project sets the stage for future enhancements, including the collection of additional data to improve model robustness, comprehensive documentation for transparency, and scalability to adapt the model to various educational settings. The implications of this project extend beyond education, offering methodologies that can be applied in healthcare, finance, and other fields where predictive analytics can drive significant improvements.

In conclusion, the "Student Performance Predictor" project represents a meaningful step forward in leveraging machine learning to enhance educational outcomes. By providing a data-driven approach to identifying and supporting at-risk students, the project contributes to the broader goal of educational equity and excellence. The insights gained and the methodologies developed through this project have the potential to transform how educational institutions approach student performance, making a lasting impact on the lives of students and the quality of education they receive.

Furthermore, the model's ability to identify key predictors of student success, such as parental education and study habits, provides actionable insights for educators and policymakers. These findings can inform the development of targeted programs and policies aimed at improving student retention and performance. The project's focus on ethical considerations ensures that the model is not only effective but also fair and equitable, promoting trust and acceptance among stakeholders.

The use of advanced machine learning techniques, combined with rigorous validation and testing, underscores the project's commitment to scientific rigor and practical applicability. The successful implementation and validation of the Logistic Regression model demonstrate the feasibility of using predictive analytics in educational settings to achieve meaningful outcomes.

Overall, the "Student Performance Predictor" project exemplifies the potential of machine learning to drive positive change in education. By harnessing the power of data, the project provides a roadmap for future initiatives aimed at improving student success and fostering a more inclusive and supportive educational environment. The project's methodologies and findings offer valuable contributions to the field of educational data science, paving the way for continued innovation and progress.

XVI. References

A. useful resources

1) UCI Machine Learning Repository. Student Performance Data Set. https://archive.ics.uci.edu/ml/datasets/student+performance
2) Hosmer Jr, D. W., et al. (2013). Applied Logistic Regression. John Wiley and Sons. https://www.wiley.com
3) Altman, N. S. (1992). The American Statistician. https://www.tandfonline.com
4) Cortes, C., & Vapnik, V. (1995). Machine Learning, 20(3), 273-297. https://link.springer.com
5) James, G., et al. (2013). An Introduction to Statistical Learning. Springer.
6) Pedregosa, F., et al. (2011). Journal of Machine Learning Research, 12, 2825-2830. https://jmlr.csail.mit.edu
7) Google. https://www.google.com
8) AnalyticVue on neural networks and machine learning. https://www.analyticvue.com

B. Python Libraries

1) McKinney, W. (2010). Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference (pp. 56-61).
2) Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.
3) ChatGPT search https://chatgpt.com/?oai-dm=1

C. Machine Learning and Data Mining

1) Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques (3rd ed.). Morgan Kaufmann.
2) James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning (Vol. 112). New York: Springer.

D. Educational Technology and Predictive Analytics

1) Baker, R. S., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. Journal of Educational Data Mining, 1(1), 3-17.
2) Romero, C., & Ventura, S. (2013). Data mining in education. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 3(1), 12-27.

XVII. Appendices

A. Appendix A:Feature Importance

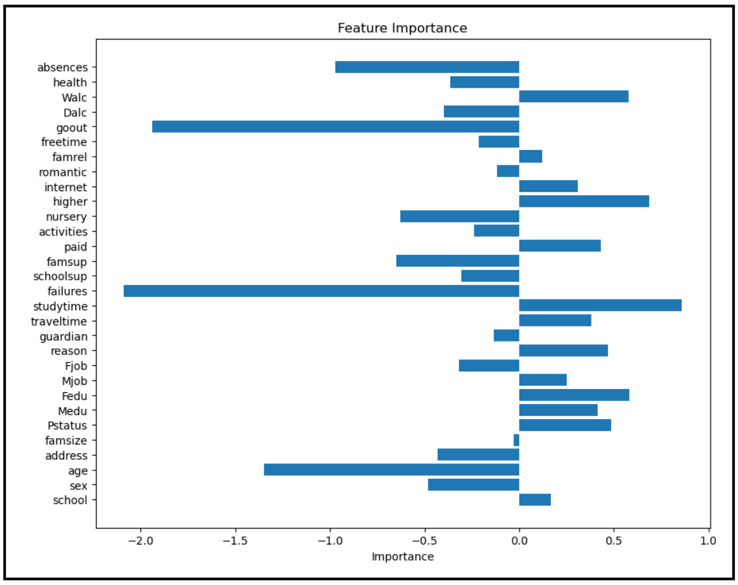


Fig. 27. Enter Caption

Figure shows the importance of various features in predicting student performance using the Logistic Regression model. The features are listed on the y-axis, and their importance is represented on the x-axis. Positive values indicate that the feature has a positive correlation with the target variable (student success), while negative values indicate a negative correlation.

Key observations:

- **Failures**: The number of past class failures is a significant negative predictor of student success.
- **Study Time**: More study time is positively correlated with student success.

- **Higher Education Intent**: Students who intend to pursue higher education are more likely to succeed.

- **Parental Education**: Higher levels of parental education (both mother and father) positively influence student success.
- **Absences**: Higher numbers of absences negatively impact student performance.

These insights highlight the key factors that contribute to predicting student outcomes, providing a basis for targeted educational interventions and support strategies.

B. Appendix B: Key Metrics Derived from the Confusion Matrix

The following key metrics are derived from the confusion matrix to evaluate the performance of the classification model:

- **Accuracy**: The ratio of correctly predicted instances (both true positives and true negatives) to the total instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP is True Positives, TN is True Negatives, FP is False Positives, and FN is False Negatives.

- **Precision (Positive Predictive Value)**: The ratio of correctly predicted positive observations to the total predicted positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision is a measure of the accuracy of the positive predictions.

- **Recall (True Positive Rate or Sensitivity)**: The ratio of correctly predicted positive observations to all actual positives.

$$\text{Recall (TPR)} = \frac{TP}{TP + FN}$$

Recall indicates how well the model captures the positive cases.
- **F1 Score**: The harmonic mean of precision and recall, providing a single metric that balances both concerns.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 Score is useful when you need a balance between Precision and Recall.

C. Appendix C: Models Used

1) Logistic Regression

Logistic Regression is a statistical method for modeling the probability of a binary outcome. In machine learning, it's used as a classification algorithm rather than a regression algorithm. It predicts the probability that an observation falls into one of two categories. Logistic regression is useful for binary classification tasks, such as predicting whether an email is spam or not, or if a student will pass or fail based on their study habits.

2) Decision Tree

A Decision Tree is a supervised machine learning algorithm used for both classification and regression tasks. It works by splitting the data into subsets based on feature values, creating a tree-like structure with nodes representing decision points and leaf nodes indicating outcomes. This method is popular due to its simplicity and interpretability, as each decision path can be easily traced and understood. However, it is prone to overfitting if not properly tuned and can be sensitive to variations in the data.

3) Support Vector Machine (SVM)

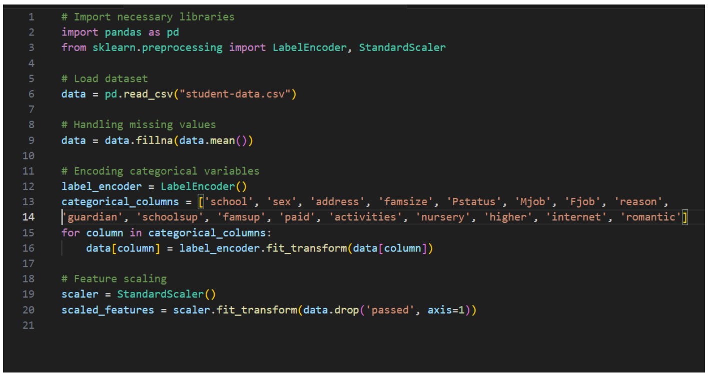
SVM (Support Vector Classifier) is a powerful and versatile machine learning model, capable of performing linear or non-linear classification, regression, and even outlier detection. It is particularly well-suited for classification of complex but small- or medium-sized datasets. SVM works by fitting the widest possible margin (or decision boundary) between the classes. For non-linear boundaries, SVM uses a technique called kernel trick to transform the input space to a higher dimensional space where a hyperplane can be used to separate the data.

D. Appendix D: Code Snippets

Include snippets of key code used in the project:

1) Data Preprocessing

The preprocessing steps include handling missing values, encoding categorical variables, and scaling features. The dataset is loaded, missing values are filled with the mean, categorical variables are encoded using LabelEncoder, and the features are scaled using StandardScaler.

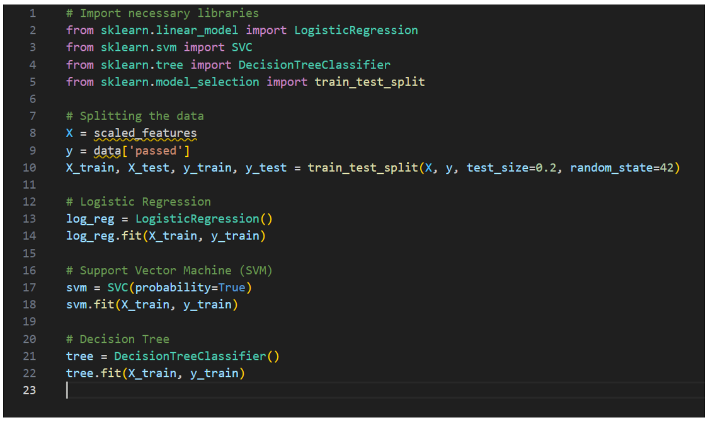


```python
# Import necessary libraries
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler

# Load dataset
data = pd.read_csv("student-data.csv")

# Handling missing values
data = data.fillna(data.mean())

# Encoding categorical variables
label_encoder = LabelEncoder()
categorical_columns = ['school', 'sex', 'address', 'famsize', 'Pstatus', 'Mjob', 'Fjob', 'reason',
'guardian', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic']
for column in categorical_columns:
    data[column] = label_encoder.fit_transform(data[column])

# Feature scaling
scaler = StandardScaler()
scaled_features = scaler.fit_transform(data.drop('passed', axis=1))
```

Fig. 28. Data Preprocessing

2) Model Training

The training section includes splitting the data into training and testing sets, and training three different models: Logistic Regression, Support Vector Machine (SVM), and Decision Tree. Each model is trained on the training set.

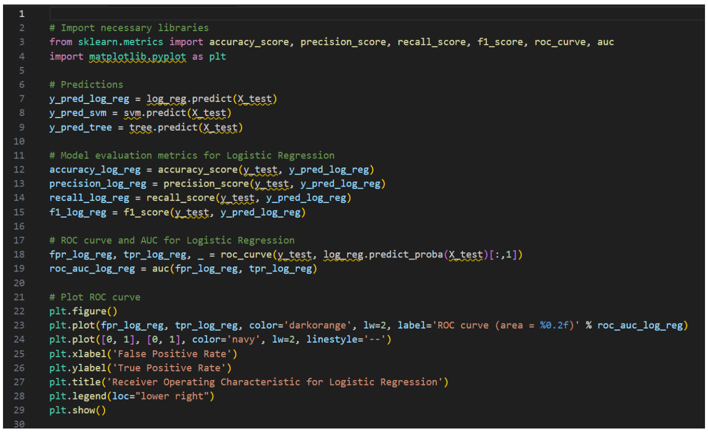


```python
# Import necessary libraries
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

# Splitting the data
X = scaled_features
y = data['passed']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Logistic Regression
log_reg = LogisticRegression()
log_reg.fit(X_train, y_train)

# Support Vector Machine (SVM)
svm = SVC(probability=True)
svm.fit(X_train, y_train)

# Decision Tree
tree = DecisionTreeClassifier()
tree.fit(X_train, y_train)
```

Fig. 29. Model Training

3) Model Evaluation

The evaluation section includes predicting on the test set using the trained models, calculating key metrics such as accuracy, precision, recall, and F1-score, and plotting the ROC curve for Logistic Regression to visualize its performance.



```python

# Import necessary libraries
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_curve, auc
import matplotlib.pyplot as plt

# Predictions
y_pred_log_reg = log_reg.predict(X_test)
y_pred_svm = svm.predict(X_test)
y_pred_tree = tree.predict(X_test)

# Model evaluation metrics for Logistic Regression
accuracy_log_reg = accuracy_score(y_test, y_pred_log_reg)
precision_log_reg = precision_score(y_test, y_pred_log_reg)
recall_log_reg = recall_score(y_test, y_pred_log_reg)
f1_log_reg = f1_score(y_test, y_pred_log_reg)

# ROC curve and AUC for Logistic Regression
fpr_log_reg, tpr_log_reg, _ = roc_curve(y_test, log_reg.predict_proba(X_test)[:,1])
roc_auc_log_reg = auc(fpr_log_reg, tpr_log_reg)

# Plot ROC curve
plt.figure()
plt.plot(fpr_log_reg, tpr_log_reg, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc_log_reg)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic for Logistic Regression')
plt.legend(loc="lower right")
plt.show()
```

Fig. 30. Model Evaluation

The End of the document