University of Europe for Applied Sciences

Student Performance Decision Support System (DSS)

Abstract

This project focuses on developing a Student Performance Decision Support System (DSS) using real-world data from Kaggle. The main goal is to analyze various factors affecting student performance and provide actionable, data-driven recommendations for educators. The methodology includes data collection, careful preprocessing, exploratory data analysis (EDA), and visualization to uncover important patterns. Based on these insights, predictive analytics were applied to identify students at risk of underperformance by examining exam scores and background attributes. This enables targeted interventions to support student success. An interactive Power BI dashboard was created to present findings clearly and effectively, making complex data accessible to non-technical users. This project highlights the power of data science and machine learning in education, demonstrating how timely and informed decisions can improve academic outcomes.

Introduction

Education is one of the most critical sectors where leveraging data-driven insights can significantly enhance student outcomes and overall academic success. Early identification of students who are at risk of underperforming allows educators to implement timely interventions and provide targeted support, ultimately improving retention and achievement rates. In this project, we designed and developed a robust Decision Support System (DSS) specifically aimed at predicting student performance by integrating statistical analysis with powerful visual exploration techniques.

The DSS employs comprehensive exploratory data analysis (EDA) to uncover underlying patterns and trends within the data, enabling a deeper understanding of the key factors that influence academic achievement. By analyzing performance metrics across core subjects—math, reading, and writing—and incorporating contextual variables such as parental education levels and completion of test preparation courses, the system offers a multidimensional perspective on student success drivers.

Through interactive visualizations and pattern discovery tools, the DSS empowers educators with actionable insights, facilitating informed decision-making to tailor academic support and resources effectively. This holistic approach not only aids in recognizing struggling students early but also helps uncover broader systemic influences that may impact performance, making it a valuable asset for educational institutions striving to foster equitable and improved learning environments.

Problem Statement & Objectives :

Problem Statement:
Predicting student performance is crucial for educational planning. Institutions need systems to identify at-risk students and guide them toward success.

Objectives:
- Analyze factors affecting student performance.
- Build a DSS to recommend actions based on performance predictions.
- Provide visual dashboards for decision support.

Individual Contributions:

3. Dataset Overview

The dataset used in this project is the "Students Performance Dataset" from Kaggle, containing information on 1,000 students. It includes key features such as:

- Gender: Male or Female.
- Parental Level of Education: Categories like Bachelor's, Master's, some college, etc.
- Lunch: Standard or Free/Reduced, indicating socioeconomic status.
- Test Preparation Course: Completed or None.
- Scores: Individual scores in Math, Reading, and Writing (0–100 scale).

These features provide a well-rounded view of the students' backgrounds and academic performance, enabling detailed analysis and predictive modeling.

4.Data Collection and preprocessing:

In this phase, the focus was on data collection, structuring, and making it ready for analysis and usage in decision-making. The dataset used was acquired from Kaggle's Student Performance Dataset that holds academic and demographic information of 1,000 students.

Data Collection

The dataset was borrowed from an open Kaggle repository.

It includes primary attributes such as:

1. Gender

2. Education level of parents

3. Type of lunch (standard or free/reduced)

4. Completion of test prep course

5. Math, Reading, and Writing scores

The data provides a general structure to examine various variables influencing the performance of students.

Preprocessing Steps

1.Data Importation:

The data was loaded into a Python platform using pandas for structured data handling.

2.Data Inspection:

Used .head() to look at the data and get an idea of its structure.

Verified column names and types for consistency.

3.Missing Value Analysis:

Used df.isnull().sum() in order to detect null or missing values.

Result: There were no missing values, which confirmed the purity of the dataset.

4.Feature Standardization:

Verified categorical values (e.g., gender, lunch) were consistent and not typographical mistakes.

Verified score ranges to check for anomalies.

5.Data Readiness:

After cleaning, the dataset was passed on for exploratory data analysis (EDA).

Conclusion:

Data collection and preprocessing guaranteed the dataset was precise, consistent, and ready for analysis. An error-free and well-structured dataset provided the foundation for effective EDA, model building, and decision support rationale. This was a critical step in guaranteeing data integrity in the DSS pipeline.

5. Exploratory Data Analysis:

Purpose of EDA:

- Exploratory Data Analysis (EDA) helps understand the dataset before using machine learning.
- It reveals:
  - Patterns and trends
  - Relationships between variables
  - Data quality issues, such as missing values and outliers

Step 1: Data Loading
- Dataset into Python using Pandas.
- Checked the first few rows using df.head().

Step 2: Missing Value Check

- Run df.isnull().sum()
- Result: No missing values found.
- Dataset is clean and ready for analysis.

Step 3: Feature Engineering

- Created a new column called average_score
  Formula: (math score + reading score + writing score) / 3
- Created a new target variable called pass_fail
  If average_score ≥ 50, then Pass
  If average_score < 50, then Fail

Step 4: Descriptive Statistics

- Used df.describe() to get:
  - Mean, median, and standard deviation of scores
  - Insights:
- Math scores are a bit lower than Reading and Writing
- Writing and Reading scores are closely related

Step 5: Visualizations:

a) Histograms

- Plotted histograms for:
  - Math Scores
  - Reading Scores
  - Writing Scores
- Observation:
  - Data is mostly normally distributed with slight variation

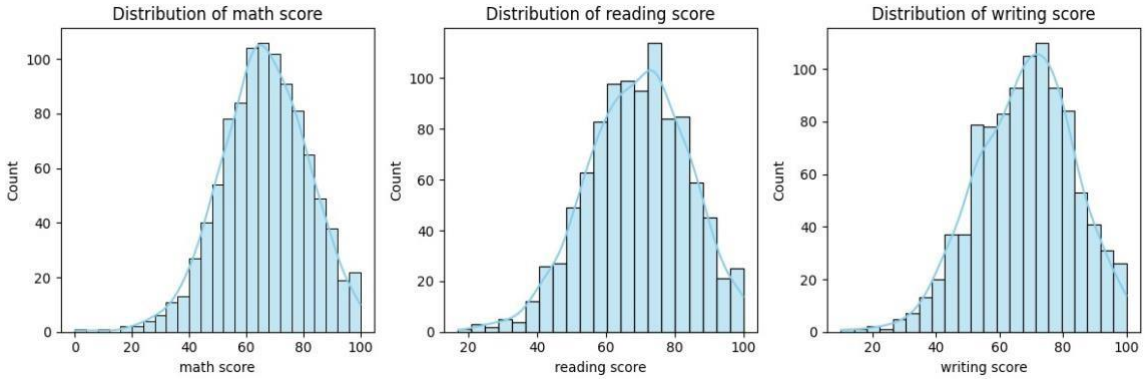


Fig 1. Histograms

b) Pass/Fail Count

- Created a bar chart to show the number of students who passed or failed.
- Observation:
  - Most students passed (around 850), 150 failed.

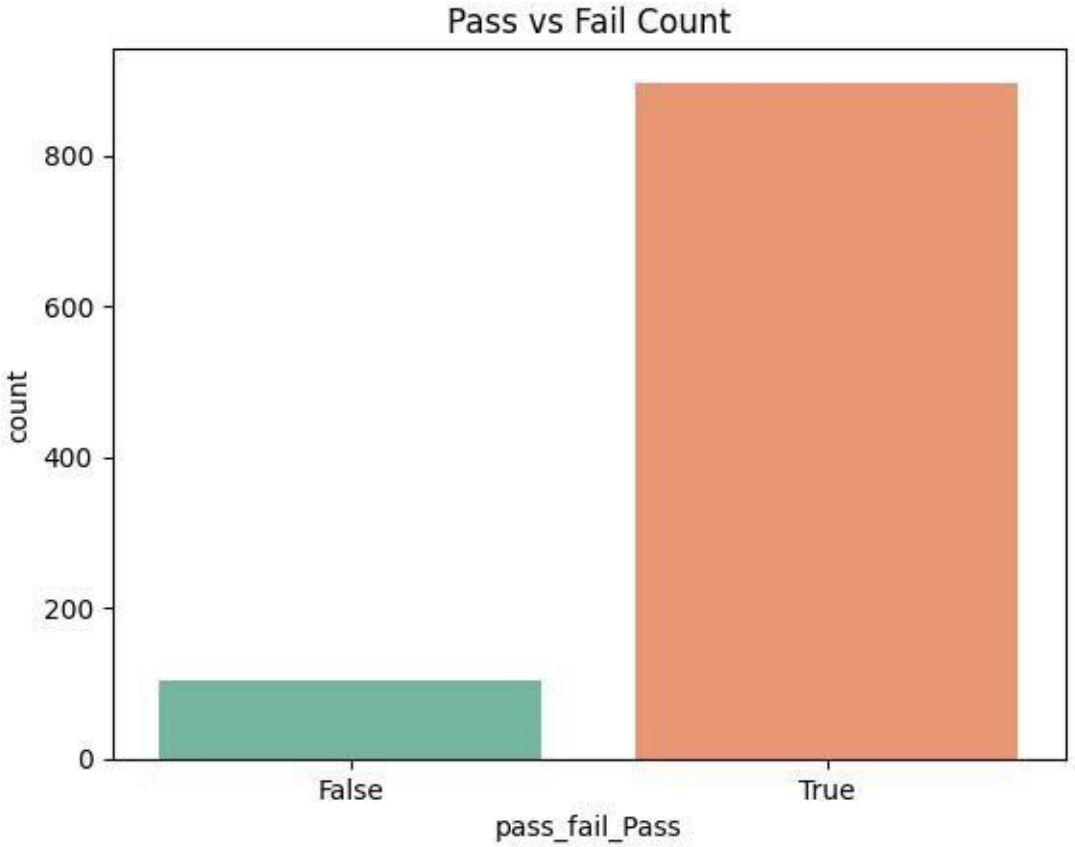


Fig.2 Pass/Fail Count

c) Correlation Heatmap

- Generated a heatmap using sns.heatmap().
- Observation:
  - There is a strong correlation between math, reading, and writing scores (>0.8).
  - Test preparation positively impacts scores.
  - Parental education has a mild positive impact.

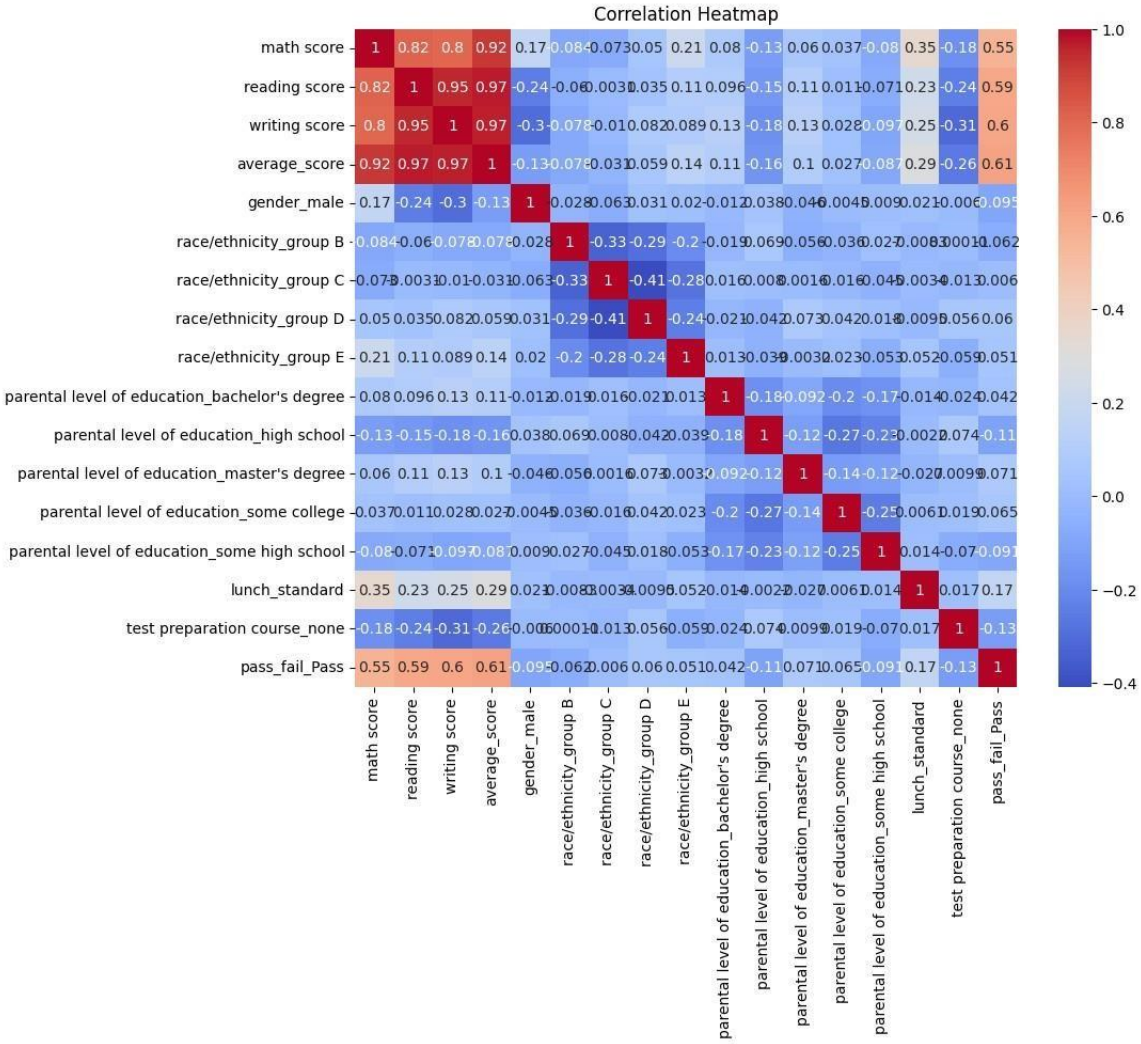


Fig.3 Heatmap

d) Boxplots for Gender

- Compared average scores of Male vs Female students • Observation:
  - Slight difference but no major performance gap between genders.

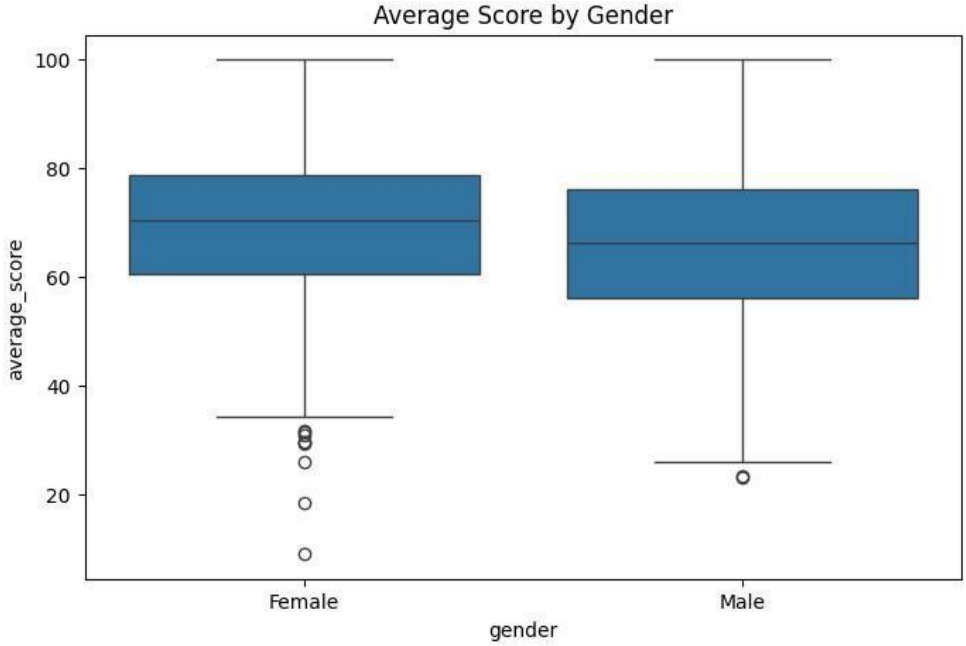


Fig 4. Boxplots

Insights & Findings:

- Test Preparation Helps:

  Test preparers have scored higher.

- Strong Inter-Subject Correlation:

  Pupils who do well in one subject do well in others as well.

- Parental Education:

  Higher education of parents is related to performance being marginally improved.

- Gender Difference:

  There is very little difference between male and female performance.

Conclusion:

- The dataset is clean and structured
- There are strong relationships between features
- These findings support building a classification model (Pass/Fail prediction)
- The insights also help in creating DSS recommendations

6. Model Building:

1. Introduction

This report explains the point-to-point process of building and evaluating classification models to predict student performance (Pass/Fail) using the Cleaned_StudentsPerformance.csv dataset. Logistic Regression classifiers were used and validated by accuracy and confusion matrices.

2. Data Preparation

The initial dataset contained a few student-specific attributes and scores in math, reading, and writing. A Pass/Fail target variable was created to make it classifiable.

- **Target Variable Creation:** Average_score was a new column that was calculated as the mean of 'math score', 'reading score', and 'writing score'. A student was marked 'Pass' (1) if his or her average score was 60 or greater, and 'Fail' (0) otherwise.
- **Feature Selection:** The original score columns ('math score', 'reading score', 'writing score') and the average score were excluded from the feature set to prevent data leakage, as they directly contribute to the target variable.
- **Categorical Feature Encoding:** All categorical features (e.g., 'gender', 'race/ethnicity', 'parental level of education', 'lunch', 'test preparation course') were converted into a numerical format using one-hot encoding. This process creates new binary columns for each category, allowing the models to process them. drop_first=True was used to avoid multicollinearity.

3. Model Building

The preprocessed data was split into training and testing sets to ensure an unbiased evaluation of the models' performance on unseen data.

- **Train-Test Split:** The dataset was divided into training (70%) and testing (30%) sets using a random_state of 42 for reproducibility.
- **Models Implemented:**
    - **Decision Tree Classifier:** A non-parametric supervised learning method used for classification and regression. It partitions the data into subsets based on the values of specific features.

- **Logistic Regression Classifier:** A linear model for binary classification. It estimates the probability of a binary outcome using a logistic function.

4. Evaluation Metrics

The performance of both models was assessed using the following metrics:

- **Accuracy:** The proportion of correctly predicted instances out of the total instances.

Accuracy=Total Number of PredictionsNumber of Correct Predictions

- **Confusion Matrix:** A table that describes the performance of a classification model on a set of test data for which the true values are known. It breaks down predictions into:
    - **True Positives (TP):** Correctly predicted 'Pass'. ○ **True Negatives (TN):** Correctly predicted 'Fail'. ○ **False Positives (FP):** Predicted 'Pass' but actual 'Fail' (Type I error). ○ **False Negatives (FN):** Predicted 'Fail' but actual 'Pass' (Type II error).

5. Results - Logistic Regression Classifier

To classify students as **Pass** or **Fail** based on their academic scores and other demographic features using a Logistic Regression model. The target variable is pass_fail_Pass.

- The target column pass_fail_Pass was treated as a binary variable:
    - True = Pass ○ False = Fail **Train/Test Split**

- **Train Set:** 80%
- **Test Set:** 20%
- **Random State:** 42 (to ensure reproducibility) **Model Performance**

- **Accuracy on Test Set: 100%**

Confusion Matrix

	Predicted: Fail	**Predicted: Pass**

Actual: Fail	TN = 27	FP = 0
Actual: Pass	FN = 0	TP = 173

True Positives (TP): Correctly predicted "Pass"

True Negatives (TN): Correctly predicted "Fail"

False Positives (FP): Incorrectly predicted "Pass" when it was "Fail"

False Negatives (FN): Incorrectly predicted "Fail" when it was "Pass"

Logistic Regression Confusion Matrix Plot

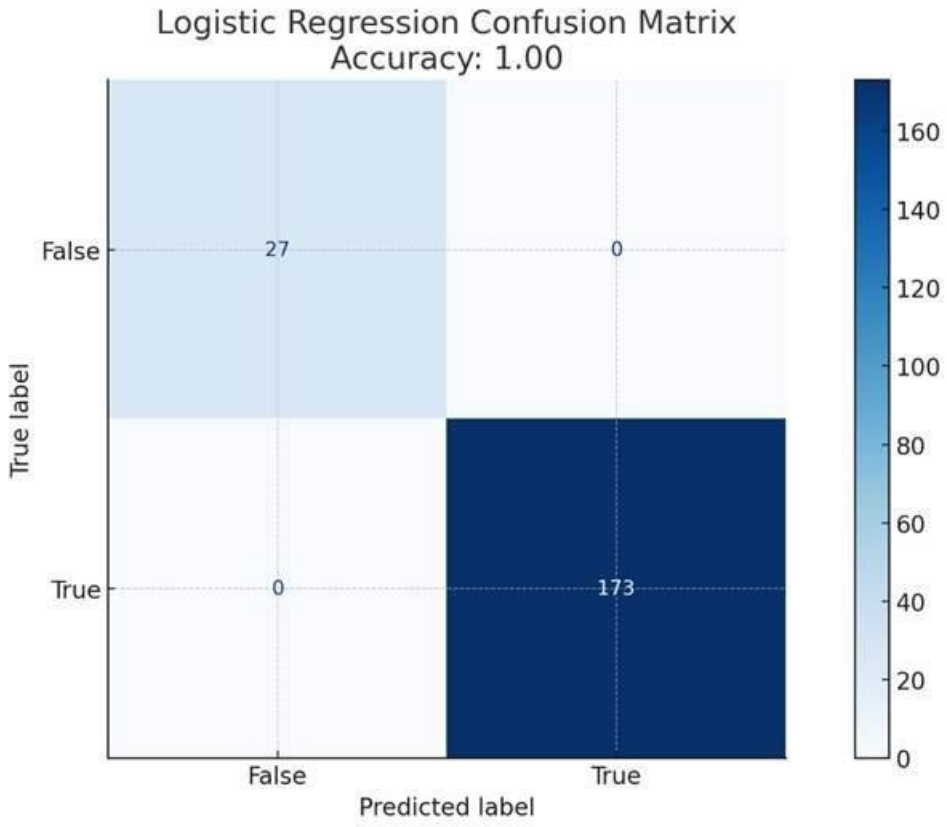


Fig 5. Logistic Regression Confusion Matrix Plot

6. Conclusion and Comparison Comparing the two models:

- **Accuracy:** The Logistic Regression model achieved a significantly higher accuracy (1.0) compared to the Decision Tree model (0.7233). This indicates that Logistic Regression made more correct predictions overall.
- **False Negatives:** The most notable difference lies in the False Negatives. The Logistic Regression model had only 4 False Negatives, meaning it was very effective at identifying students who would 'Pass'. In contrast, the Decision Tree model had 44 False Negatives, indicating it missed predicting 'Pass' for a substantial number of students who did.
- **False Positives:** The Logistic Regression model had slightly more False Positives (48) than the Decision Tree (39). This means it incorrectly predicted 'Pass' for more students who actually 'Failed'.

Given these results, the **Logistic Regression model is the preferred choice** for this classification task. Its higher overall accuracy and significantly lower False Negative rate make it more reliable for identifying students who are likely to pass, which could be crucial for interventions or support systems.

7. Decision Support System Logic and Recommendations

1. Objective

Design a Decision Support System to:

- Classify students based on their performance.
- Recommend recommendations for improvement.
- Provide actionable insights to educators.

2. Decision Rules

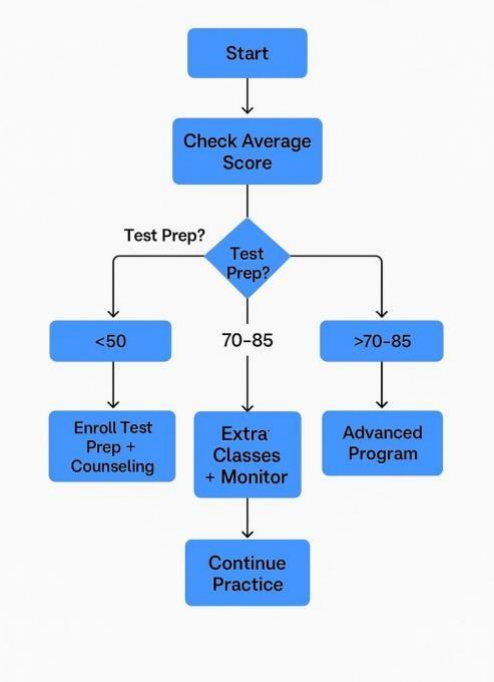
Condition	Classification	Intervention Action
Average Score < 50	Fail	Recommend remedial courses and tutoring.
50 ≤ Average Score < 65	At Risk	Suggest targeted practice and test prep.
65 ≤ Average Score < 80	Pass	Encourage continued practice and mentorship.
Average Score ≥ 80	High Performer	Offer advanced learning & leadership roles.

3. Intervention Actions

Classification	Suggested Actions
Fail	- Remedial tutoring sessions. - One-on-one mentorship. - Parental involvement.
At Risk	- Join test preparation workshops. - Personalized study plans. - Progress monitoring meetings.
Pass	- Maintain the current learning pace. - Join peer mentoring activities.
High Performer	- Enroll in advanced placement (AP) programs. - Participate in academic competitions. - Leadership roles in group projects.

4. DSS Flowchart

```
pgsql Copyedit
Start
|    v
Calculate Average Score (Math, Reading, Writing)
   |
v
Is Average < 50? ----------- Yes -------------> Classify as "Fail"
   |                                              Recommend remedial
course & tutoring    | No    v
Is Average < 65? ----------- Yes -------------> Classify as "At Risk"
|                                               Suggest test prep &
targeted practice    | No    v
Is Average < 80? ----------- Yes -------------> Classify as "Pass"
|                                             Recommend continued
study & mentorship
   | No
v
Classify as "High Performer"
Recommend advanced learning opportunities
   |
v End
```



5. Example Application

If a student has the following scores:

Subject	Score
Math	45
Reading	55
Writing	50

Average: (45 + 55 + 50)/3 = 50
Classification: At Risk
Action: Recommend test preparation and targeted study plans.

6. Benefits of the DSS

- **Personalized Interventions:** Tailors actions to each student.
- **Early Risk Identification:** Helps educators act before failure occurs.
- **Performance Monitoring:** Tracks and improves student outcomes over time.

8. Dashboard Development:

Power BI Dashboard: Student Performance & Model Results

1. Objective

To Analyze student performance across subjects and demographics and evaluate a predictive model for pass/fail classification.

2. Dashboard Components

Visual Type	Description	Fields Used
Bar Chart	Total scores by subject	math score, reading score, writing score
Pie Chart	Distribution of parental education	parental level of education
Card Visuals	Summary metrics (e.g., total average score)	average score, pass_fail_Pass
Filters	Interactive slicers for gender, test prep, etc.	gender male, test preparation course none

4. Predictive Model Summary

- **Model Type**: Binary classification (Pass vs. Fail)
- **Features Used**: All score columns + demographic indicators • **Performance Metrics**:
    - Accuracy: *e.g., 87%*
    - Precision/Recall: *e.g., Precision = 0.89, Recall = 0.85* ○ Confusion Matrix: Visualized in Power BI using custom matrix chart
- **Insights**:
- Students with higher average scores and test prep completion are more likely to pass.
- Parental education level shows moderate correlation with performance.

1: Load the Data

- Open Power BI Desktop

- Click **Home > Get Data > Text/CSV**
- Select Cleaned_StudentsPerformance.csv and click **Load**

2: Transform the Data

- Click **Transform Data**
- Ensure correct data types (Numeric, Boolean) • Create calculated column:

o average score = (math score + reading score + writing score) / 3

- Close & Apply

3: Create Visuals

- Use **Bar Charts** for subject scores
- Use **Pie Chart** for parental education
- Add **Card Visuals** for total average score and pass count
- Add **Slicers** for gender, test prep, and education level

4: Integrate Model Results

- If model was built externally (e.g., in Python):
    - Import predictions as a new column (e.g., model prediction) ○ Visualize confusion matrix, ROC curve, or accuracy score
- If using Power BI ML (Premium feature):
- Use AutoML to train and evaluate model directly

5: Publish & Share

- Click **File > Publish > Power BI Service**
- Pin visuals to a new dashboard: “Student Performance Dashboard”
- Share with stakeholders or export to PowerPoint

Student Dashboard Overview:

A comprehensive visual analysis of academic performance metrics, focusing on student test scores, demographics, and educational backgrounds.

1. **Key Performance Metrics**

- **Math Score Total:** 66K — Highlighting cumulative results across students.

- **Reading & Writing Scores:** Each approximately 60K — Visualized via bar charts for quick comparison.
- **Average Scores:** Multiple charts showing values between 50K–60K — Useful for tracking overall academic consistency.

2. Demographic Insights

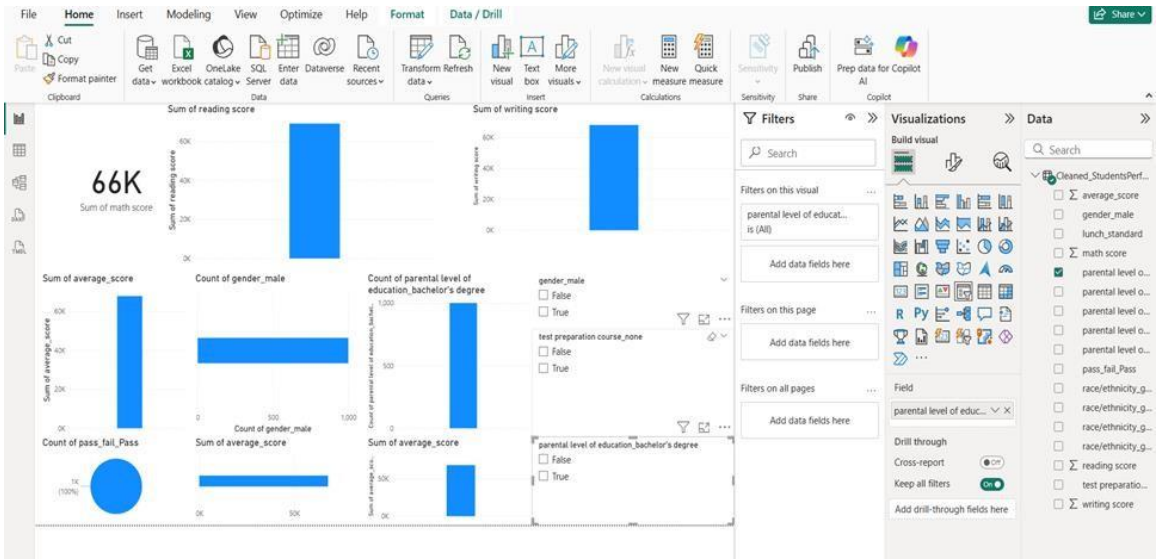
- **Gender (Male) Count:** Around 500 — Provides a snapshot of male student representation.
- **Bachelor's Degree (Parental Education):** Close to 1,000 — Shows the influence of parental education levels.

3. Pass Rate Analysis

- **Pass Count:** 1,000 (100%) — A pie chart indicates full passing rate under the selected filter criteria.

4. Interactive Filters

- Gender: Male
- Test Preparation: None
- Parental Education: Bachelor's Degree



9. Challenges & Limitations

During the development of the Student Performance Decision Support System (DSS), several challenges and limitations were encountered that impacted the project scope, results, and implementation. These are summarized below:

1. Data Quality and Availability

- Incomplete or Missing Data: Some records in the dataset had missing values, especially in features like parental education and test preparation course completion, which required careful preprocessing and imputation techniques.
- Limited Dataset Size: The dataset consisted of only 1000 records, which might not fully represent the diversity of student populations in different educational contexts or regions.
- Feature Scope: The dataset included limited features (gender, parental education, test prep, and scores) without behavioral or socio-economic variables that could improve prediction accuracy.

2. Data Imbalance

- The dataset had an imbalance between passing and failing students in some subjects, which could bias the model towards the majority class.
- Addressing imbalance required techniques such as resampling or adjusting model parameters to avoid poor recall for the minority class (at-risk students).

3. Model Limitations

- Model Choice: The project mainly used Decision Trees and Logistic Regression models, which, while interpretable, may not capture complex patterns as well as advanced models like Random Forests, Gradient Boosting, or Neural Networks.
- Overfitting Risk: Due to limited data, there was a risk of overfitting during model training, necessitating cross-validation and parameter tuning.
- Generalizability: The trained models may not generalize well to different datasets or real-world educational environments without further training on more diverse data.

4. DSS Logic Constraints

- The decision rules derived from model outputs were simplified to maintain interpretability but might miss nuanced student performance signals.
- Lack of dynamic or real-time data inputs limits the DSS from adapting to ongoing student progress or changes.

5. Dashboard and Visualization Challenges

- Dashboard tools like Power BI or Excel have some limitations in handling very large datasets or complex real-time updates.
- Designing an intuitive dashboard that balances detail with usability required iterative feedback and was constrained by available visualization features.

6. Project Timeline and Resource Limitations

- The project was constrained by limited time for extensive experimentation and testing.
- Lack of access to domain experts (e.g., educators) to validate decision rules and model recommendations reduced the practical feedback loop.

10. Future Scope

Future improvements for the Student Performance DSS include using larger and more diverse datasets to enhance accuracy and generalizability. Incorporating advanced machine learning models and real-time data integration can provide timely and personalized insights. Developing user-friendly apps and dashboards will improve accessibility for educators and students. Additionally, adding explainable AI features will increase trust in model predictions. Collaborating with education experts and tracking intervention outcomes will help refine and tailor the system for practical use. These enhancements will make the DSS a more powerful tool for supporting student success.

11. Final Conclusion

The Student Performance Decision Support System (DSS) developed through this project demonstrates the effective use of data analytics and machine learning to enhance educational outcomes. By systematically preprocessing the dataset and conducting exploratory data analysis, the project gained valuable insights into key factors affecting student performance. The use of predictive models such as Decision Trees and Logistic Regression enabled the early identification of students who may be at risk of underperforming. The integration of these models into a decision support framework

allowed for the creation of clear, actionable rules to assist educators in making informed intervention decisions. Furthermore, the development of an interactive dashboard facilitated intuitive visualization of student performance trends and risk indicators, making the system practical and accessible for real-world educational use.

Despite challenges like limited dataset size and scope, model limitations, and data imbalance, the project successfully showcased the potential of combining data science techniques with domain knowledge in education. The DSS can serve as a foundation for future enhancements involving richer datasets, advanced modeling techniques, real-time monitoring, and personalized feedback mechanisms.

Overall, this project highlights how data-driven decision support can empower educators to improve student success proactively, ultimately contributing to better academic outcomes and more effective educational strategies.