Muhammad Abdullah

STU-DS-251-930

Student Performance Prediction

Project Overview:

This project aims to build a predictive model for student academic performance using machine learning techniques. The dataset contains academic and behavioral attributes such as gender, ethnicity, parental education level, tutoring, extracurricular involvement, and more. The objective is to predict student performance (Low, Average, or High GPA) based on these features.

1. Importing Libraries:

We begin by importing the essential Python libraries for data analysis, visualization, and machine learning:

- `numpy` and `pandas` for numerical operations and data manipulation
- `matplotlib.pyplot` and `seaborn` for data visualization
- `sklearn` for preprocessing, model building, evaluation, and train-test splitting

2. Load Dataset:

The dataset (e.g., `student_performance.csv`) is loaded using `pandas.read_csv()`. Basic inspection using `.head()`, `.info()`, and `.describe()` helps understand data types, structure, and completeness.

3. Data Cleaning:

To ensure clean input for models:

- Missing values are checked and handled accordingly (if any)
- Duplicate rows are dropped to eliminate bias

4. Categorical Data Handling:

Many features such as Gender, Ethnicity, Parental Education, and Extracurriculars are categorical. These are encoded into numerical format using `LabelEncoder` for compatibility with machine learning algorithms.

5. Feature Scaling:

Numerical features are scaled using `StandardScaler`, which normalizes them to a standard normal distribution (mean = 0, standard deviation = 1). This ensures fair contribution of all features in model training.

6. Train-Test Split:

The dataset is split into:

- Training Set (80%) – used to train machine learning models
- Testing Set (20%) – used to evaluate performance on unseen data We use `train_test_split()` from `sklearn.model_selection`.

7. Exploratory Data Analysis (EDA):

We analyze patterns and relationships between features and the target:

- A countplot of GPA categories (Low, Average, High)
- A heatmap to show correlation between features
- Boxplots and bar plots to visualize GPA variation with categorical features

8. Feature Selection:

Using correlation analysis, we observe which features are most relevant to the GPA target. Highly correlated features are preferred during training to improve accuracy and reduce noise.

9. Model Training & Evaluation:

Four classification models are trained and evaluated:

Logistic Regression

- A linear model suitable for classification
- Controlled using regularization to prevent overfitting

Decision Tree Classifier

- A tree-based model that learns decision rules from the data
- Risk of overfitting is reduced using `max_depth`

Random Forest Classifier

- An ensemble of decision trees with better generalization
- Less prone to overfitting and more stable

Support Vector Machine (SVM)

- Attempts to find the best boundary (hyperplane) to classify classes
- Performs well on scaled, high-dimensional data

For each model, we:

- Fit on the training set
- Predict on the test set
- Evaluate using the following metrics:
  - **Accuracy** – Overall correct predictions
  - **Precision** – Correctness of positive predictions
  - **Recall** – Ability to capture actual positives
  - **F1 Score** – Harmonic mean of precision and recall
  - **Confusion Matrix** – Matrix showing TP, TN, FP, FN

10. Model Comparison:

All models are compared using a metric summary table to analyze which performs best overall. Depending on accuracy, F1 score, and interpretability, the most effective model is selected.

Conclusion:

This notebook successfully implements a full machine learning pipeline to predict student academic performance based on various academic and behavioral attributes. Visualizations and metrics assist in interpreting model strengths. Future improvements could involve:

- Hyper parameter tuning
- Adding more advanced features like study hours or attendance data
- Converting this into a real-world predictive tool for schools or educators