Student Performance Prediction - Model Training Documentation

# 1. Design

The objective of this project is to predict students' final grades based on various features like demographics, academic performance, and social factors. The dataset used for this task is the UCI Student Performance dataset, which contains information on Portuguese students. The target variable is the final grade (G3), and the features are a mix of categorical and numerical data.

**Description of the Dataset**

This data approach student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school-related features) and it was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por). In [Cortez and Silva, 2008], the two data sets were modeled under binary/five-level classification and regression tasks. Important note: the target attribute G3 has a strong correlation with attributes G2 and G1.

**2.1 Attribute Information:**

* school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
* 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 (Math or Portuguese) (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)

# 2. Implementation

## 2.1 Linear Regression

We trained a simple Linear Regression model to predict the students' final grades. The data was split into training and testing sets, and the model was evaluated using

M**ean Squared Error** and **R² Score**: These metrics will show how well the model performs. The lower the MSE and the closer the R² score is to 1, the better the model fits the data.

## 2.2 Logistic Regression

Training a Logistic Regression model on the student performance dataset requires transforming the problem into a classification task. Since Logistic Regression is used for classification (rather than regression), you need to convert the target variable (student grades, G3) into discrete classes.

One common approach is to categorize the final grades into distinct groups or classes (e.g., low, medium, and high performers) and then predict which class a student belongs to.

Confusion Matrix and Classification Report give insights into the model’s performance and where it might misclassify students

## 2.3 K-Means Clustering

Applying KMeans clustering to the student performance dataset will group students into clusters based on similarities in their features. However, since this dataset is structured for prediction tasks (i.e., predicting student performance), KMeans might reveal interesting clusters or patterns, but it won’t give direct prediction results. Clustering is an unsupervised learning technique, while performance prediction typically requires supervised learning.

## 2.4 Support Vector Machines (SVM)

## We use the SVR() class for regression with the Radial Basis Function (RBF) kernel. This kernel is commonly used when the data is non-linearly separable

## Preprocessing steps included scaling numerical features and One-Hot Encoding categorical features to prepare the data for SVM.

## The model's MSE and R² Score suggest it performs moderately well but could be improved with more feature engineering or hyperparameter tuning (e.g., grid search for the kernel, C, and gamma parameters in SVM

## 2.5 Decision Trees

A Decision Tree model was trained on the data. Decision trees can capture non-linear relationships, making them suitable for this dataset. We use DecisionTreeRegressor() from the sklearn.tree module for **regression** tasks  The model was evaluated using **MSE** and **R² score**.

You may consider tuning the hyperparameters of the Decision Tree (e.g., max\_depth, min\_samples\_split) to improve model performance.

## 2.6 Random Forest

Random Forests are ensemble models that combine the predictions of multiple decision trees to produce a more accurate and stable prediction.

Random Forests are less prone to overfitting than Decision Trees and generally provide better performance due to their ensemble nature. **Random Forests** tend to reduce overfitting and handle a mix of numerical and categorical data effectively, making it a good choice for this task.

You can further improve results by tuning hyperparameters such as n\_estimators, max\_depth, and others through techniques like **GridSearchCV** or **RandomizedSearchCV**.

## 2.7 XGBoost

XGBoost, a popular boosting algorithm, was trained on the dataset. XGBoost showed a significant improvement in both MSE and R² scores compared to other models. This was due to its ability to handle complex patterns and reduce bias.

**XGBoost** is a powerful tool for regression tasks and can be further tuned by adjusting hyperparameters such as learning\_rate, max\_depth, and n\_estimators for potentially better performance.

## 2.8 Neural Network

To train a model using a **Neural Network** for predicting student performance, we can use the **Keras** library (built on top of **TensorFlow**) to implement a feedforward neural network.

A Neural Network was built using Keras to predict students' final grades. The network consisted of several hidden layers with ReLU activations.

You can experiment with adding more layers, increasing/decreasing the number of neurons, or using different activation functions (e.g., tanh, sigmoid).

# 3. Challenges

1. Handling a mix of categorical and numerical features: The dataset contains a variety of data types, which required careful preprocessing (scaling for numerical features and one-hot encoding for categorical features).

2. Selecting appropriate models: Since the task is regression, we needed to avoid classification algorithms. Models like Logistic Regression and K-Means were not suitable.

But Instead, ), you need to convert the target variable (student grades, G3) into discrete classes.

One common approach is to categorize the final grades into distinct groups or classes (e.g., low, medium, and high performers) and then predict which class a student belongs to. Confusion Matrix and Classification Report give insights into the model’s performance and where it might misclassify students

3. Hyperparameter tuning: Optimizing models like XGBoost and Neural Networks required extensive tuning of hyperparameters to achieve optimal performance.

# 4. Results

The results of the different models varied based on complexity and ability to capture relationships in the data. Below is a summary of the performance metrics for the trained models:

|  |  |  |
| --- | --- | --- |
| Model | Mean Squared Error (MSE) | R² Score |
| Linear Regression | 5.65 | 0.72 |
| SVM (SVR) | 4.84 | 0.76 |
| Random Forest Regressor | 3.80 | 0.81 |
| XGBoost Regressor | 1.72 | 0.91 |
| Neural Network | 1.54 | 0.68 |
|  |  |  |