Student name: Susanna Mkrtchyan

Project Report: Student Performance Analysis using kNN

I chose the Student Performance dataset for this project because it contains practical features-such as study time, absences, and previous failures-that logically influence a student's final grade. My objective was to apply the kNN algorithm to both classification and regression tasks. In the classification part, I predicted whether a student would pass or fail based on a threshold (G3 >= 10), while in the regression part, I aimed to forecast the exact numerical grade (G3).

Before training the model, I performed several preprocessing steps. I selected the most relevant features and checked for missing data. I used StandardScaler to normalize the data. This is a very important step because, without scaling, features with larger numbers (like absences) would confuse the model and make it ignore smaller but important features (like study time). To find the best settings, I used GridSearchCV with cross-validation. This process showed that the Euclidean distance metric worked best for both tasks, with an optimal k=31 for classification and k=17 for regression.

The model's performance is visualized through four key plots:

1. **Cross-validation scores plot:** This graph showed how the model's accuracy changed with different *k* values during training. It helped identify the "sweet spot" where the model is stable and not overfitting.
2. **Confusion Matrix Heatmap:** For classification, this heatmap revealed that the model is quite good at identifying students who pass. However, it also showed some "false positives," where the model predicted a pass for students who actually failed, likely due to similarities in their profiles.
3. **MSE vs k:** In the regression task, this plot showed that as k increases, the Mean Squared Error (MSE) initially drops and then levels off. This helped me confirm that k=17 is the point where more neighbors no longer improve the prediction.
4. **True vs Predicted G3 Grades:** This scatter plot shows the final accuracy of the regression. While the points follow the ideal red line, they tend to cluster around the average (10-12 points).

The final results showed that the model performs reliably on both tasks. The classification model effectively distinguished between pass and fail outcomes, while the regression model followed the general trend of actual grades.

However, there are some clear limits. The model tends to predict grades that are close to the average (10-12 points) and has a hard time with extreme cases. For example, it struggled to predict a grade of "0" because its nearest neighbors usually had much higher scores. Overall, this project taught me that while kNN is a simple and effective tool, its success depends heavily on how the data is scaled and how carefully we choose the k value and distance metric.