Machine Learning Classification Project

This document outlines the various stages of a machine learning classification project, focusing on data preprocessing, model building, and evaluation. Each section is detailed to present the methodology, techniques, and results involved in the classification task.

1. Data Definition and Preprocessing

1.1 Initial Data Analysis

The initial steps include defining the dataset by checking data types (categorical or numerical), counting unique values, and identifying missing values. The mean of each feature is also calculated to understand the central tendencies of the data.

1.2 Handling Missing Values and Noise

Missing values were detected in the 'semester' feature and were subsequently removed. Additionally, noisy data points, such as an incorrect GPA value of 99, were identified and dropped from the dataset.

1.3 Normalization and Encoding

Before proceeding to modeling, all data were normalized. This step ensures that the scale of each feature is consistent, which is crucial for many machine learning algorithms. Categorical string values were converted to numeric codes for proper processing.

1.4 Data Visualization and Correlation

Countplots and boxplots were generated for categorical and numerical features respectively to better understand the distribution of the data. Additionally, a correlation matrix was computed and visualized using a heatmap to investigate the relationships between features.

2. Model Building and Evaluation

2.1 Logistic Regression

A logistic regression model was built by first defining the features (X) and the target (Y). The data were split into 80% training and 20% testing sets. Cross-validation with 10 folds was applied, with fold 6 showing the lowest accuracy. The final model was evaluated using accuracy and a confusion matrix.

2.2 K-Nearest Neighbors (KNN)

For the KNN model, X and Y were defined similarly. Parameter tuning was performed on the number of neighbors, with plots drawn to compare training and test accuracy. The goal was to find the K value where these accuracies converge, achieving the best balance between simplicity and performance. Evaluation metrics included accuracy and confusion matrix.

2.3 Decision Tree

A Decision Tree Classifier was implemented using the entropy criterion, with a max depth of 6. The model was tuned using these parameters, and performance was evaluated using accuracy and confusion matrix.

2.4 Random Forest

A Random Forest model was built with 100 estimators and a max depth of 6. As with previous models, accuracy and confusion matrix were used to assess performance.

2.5 Deep Neural Network (DNN)

A DNN model was developed with multiple layers, using the ReLU activation function. Adam was chosen as the optimizer, with a batch size of 5 and 100 epochs. This model was evaluated with accuracy and confusion matrix.

3. Feature Engineering and Optimization

A custom function was created to check the 2-dimensional nature of features, allowing for feature selection based on their impact on the model's accuracy. Features that positively impacted accuracy were added to the dataset as 2D features, leading to an approximate 10% improvement in accuracy.

Conclusion

This project covered the entire pipeline from data preprocessing to model evaluation, focusing on optimizing the accuracy of the final machine learning models. Feature engineering and hyperparameter tuning played crucial roles in improving the overall performance. Each model was rigorously tested and evaluated to ensure robust results.