**Name: Omar Mohammed Rabie**

**No: 22**

**Machine Learning Project Documentation**

**a. Preprocessing**

**1. Data Handling**

* The dataset is read using pandas from the CSV file student\_version.csv.
* Features (X) and the target variable (y) are separated.

**2. Feature Types**

* Features are categorized into numerical and categorical types:
  + **Categorical Features**: Identified by checking for object data types.
  + **Numerical Features**: Identified by excluding object data types.

**3. Feature Selection**

* Numerical features with fewer than 8 unique values are moved to the categorical list, as these likely represent discrete categories.

**4. Outlier Handling**

* For each numerical feature, the interquartile range (IQR) is calculated.
* Outliers are identified as values outside the range: [Q1−0.5\*IQR, Q3+0.5\*IQR]. Outliers are replaced with the mean value of the feature.

**5. Scaling and Encoding**

* **Scaling**: StandardScaler is used for numerical features to standardize them.
* **Encoding**: OneHotEncoder is applied to categorical features to convert them into a binary form.

The preprocessing pipeline is implemented using ColumnTransformer, ensuring a consistent transformation of both numerical and categorical features.

**b. Model Selection**

Three types of models were explored:

1. **K-Nearest Neighbors (KNN)**: A simple distance-based classifier.
2. **Logistic Regression**: A classic statistical model commonly used for binary classification.
3. **Deep Learning (Keras)**: A neural network model consisting of dense layers.

**K-Nearest Neighbors (KNN)**

* Hyperparameters:
  + Number of neighbors: 5
  + Metric: Minkowski distance with p=2 (Euclidean distance)

**Logistic Regression**

* A widely used linear model for binary classification, suitable for datasets with linearly separable classes.

**Deep Learning (Keras)**

* A neural network with three layers:
  + Dense layer with 16 neurons and ReLU activation.
  + Dense layer with 8 neurons and ReLU activation.
  + Output layer with 1 neuron and sigmoid activation (suitable for binary classification).

**c. Evaluation Criteria**

Each model is evaluated based on its **accuracy** on the training set, which can be further evaluated on a test set in production. Accuracy measures the proportion of correctly classified instances.

Other evaluation metrics were included also:

* **Precision**: Proportion of true positives among all positive predictions.
* **Recall**: Proportion of true positives among all actual positives.
* **F1 Score**: Harmonic mean of precision and recall, useful when there is class imbalance.

For deep learning models, metrics like **loss** and **accuracy** during training are also tracked.

**d. Explanation of Your Work**

This project demonstrates a complete pipeline for a machine learning problem involving preprocessing, model selection, and training. Here's an overview:

* **Preprocessing**: The data was cleaned and transformed to handle outliers and prepare features for modeling.
* **Modeling**: Three different approaches (KNN, Logistic Regression, and Deep Learning) were implemented and trained.
* **Why These Models?**
  + **KNN** makes no assumptions about the underlying data distribution, making it a simple but powerful model for smaller datasets.
  + **Logistic Regression** is useful for binary classification problems and helps understand feature importance through coefficients.
  + **Deep Learning** can capture complex relationships between features, especially if the data has non-linear patterns.

**e. Comparison Between Approaches**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Pros** | **Cons** | **accuracy** |
| **KNN (K-Nearest Neighbors)** | Simple, intuitive, and effective for small datasets. No training phase required. | Slow prediction with large datasets, sensitive to irrelevant features and data scaling. | 88.6% |
| **Logistic Regression** | Interpretable and efficient for binary classification tasks. | Assumes linear relationship between features and target. Limited performance on complex data. | 89.6% |
| **Deep Learning (Keras)** | Powerful, flexible, and capable of capturing complex relationships. | Requires more data and computational resources, can be prone to overfitting if not properly regularized. | 88.04% |