Drug Classification Project Documentation

Overview

This project focuses on classifying drugs based on patient characteristics using machine learning models. The dataset used contains information about patients, including their age, sex, blood pressure (BP), cholesterol levels, sodium-to-potassium ratio (Na_to_K), and the drug they were prescribed. The goal is to build a model that can accurately predict the appropriate drug for a patient based on these features.

Dataset

The dataset used in this project is named drug200.csv and contains the following columns:

- **Age**: Age of the patient.
- **Sex**: Gender of the patient (Male or Female).
- **BP**: Blood pressure level (HIGH, LOW, or NORMAL).
- **Cholesterol**: Cholesterol level (HIGH or NORMAL).
- **Na_to_K**: Sodium-to-potassium ratio in the blood.
- **Drug**: The drug prescribed to the patient (DrugY, drugX, drugA, drugB, or drugC).

The dataset contains 200 entries with no missing values.

Data Preprocessing

1. Loading the Dataset

- The dataset is loaded using pandas.read_csv().

2. Exploratory Data Analysis (EDA)

- **Data Inspection**: The first few rows of the dataset are displayed to understand its structure.
- **Missing Values**: Checked for missing values using isnull().sum(). No missing values were found.
- **Data Types**: Verified the data types of each column using info().
- **Descriptive Statistics**: Generated summary statistics for numerical columns using describe().
- **Distribution of Na_to_K**: Visualized the distribution of the Na_to_K column using a histogram.

3. Feature Engineering

- **Categorical Encoding**:
    - The categorical columns (Sex, BP, Cholesterol, and Drug) were encoded using LabelEncoder to convert them into numerical values.
- **Feature and Target Separation**:
    - The dataset was split into features (X) and the target variable (Y).

    - X contains all columns except Drug.
    - Y contains the Drug column.

4. Train-Test Split

- The dataset was split into training and testing sets using train_test_split() with a test size of 20%.

Model Training and Evaluation

1. Model Selection

- Three machine learning models were selected for evaluation:
    - **Logistic Regression**
    - **Decision Tree Classifier**
    - **Random Forest Classifier**

2. Model Training and Evaluation

- Each model was trained on the training data and evaluated on the test data.
- The accuracy of each model was calculated using the r2_score() function.
- The results were as follows:
    - **Logistic Regression**: 80.40% accuracy.
    - **Decision Tree Classifier**: 100% accuracy.
    - **Random Forest Classifier**: 100% accuracy.

3. Model Selection

- Based on the evaluation, the **Decision Tree Classifier** was selected for further tuning due to its perfect accuracy.

Hyperparameter Tuning

1. Grid Search Cross-Validation

- A **GridSearchCV** was performed on the Decision Tree Classifier to find the optimal hyperparameters.
- The hyperparameters tuned were:
    - max_depth: [1, 2, 3, 4, 5]
    - min_samples_split: [0.1, 0.2, 0.3, 0.4]
    - criterion: ['gini', 'entropy', 'log_loss']

2. Model Evaluation After Tuning

- The tuned model was evaluated on both the training and test datasets.
- The R² score for both the training and test datasets was **1.0**, indicating perfect accuracy.

Conclusion

- The **Decision Tree Classifier** achieved perfect accuracy on both the training and test datasets after hyperparameter tuning.
- The model is capable of accurately predicting the appropriate drug for a patient based on their age, sex, blood pressure, cholesterol levels, and sodium-to-potassium ratio.
- This model can be deployed in a clinical setting to assist healthcare professionals in prescribing the correct drug based on patient characteristics.

Dependencies

- **Python Libraries**:
  - pandas
  - numpy
  - matplotlib
  - seaborn
  - scikit-learn

How to Run the Code

1. **Install Dependencies**:
   Ensure that all the required libraries are installed. You can install them using pip:

   ```
   pip install pandas numpy matplotlib seaborn scikit-learn
   ```

2. **Load the Dataset**:
   Place the drug200.csv file in the same directory as the notebook or script.
3. **Run the Code**:
   Execute the code cells in the provided Jupyter Notebook (main.ipynb) to preprocess the data, train the models, and evaluate their performance.
4. **Hyperparameter Tuning**:
   The hyperparameter tuning section can be modified to experiment with different parameters and models.

References

- **Scikit-learn Documentation**: https://scikit-learn.org/stable/
- **Pandas Documentation**: https://pandas.pydata.org/pandas-docs/stable/
- **Matplotlib Documentation**: https://matplotlib.org/stable/contents.html