Final Project

Supervised Machine Learning: *Classification*

Summary

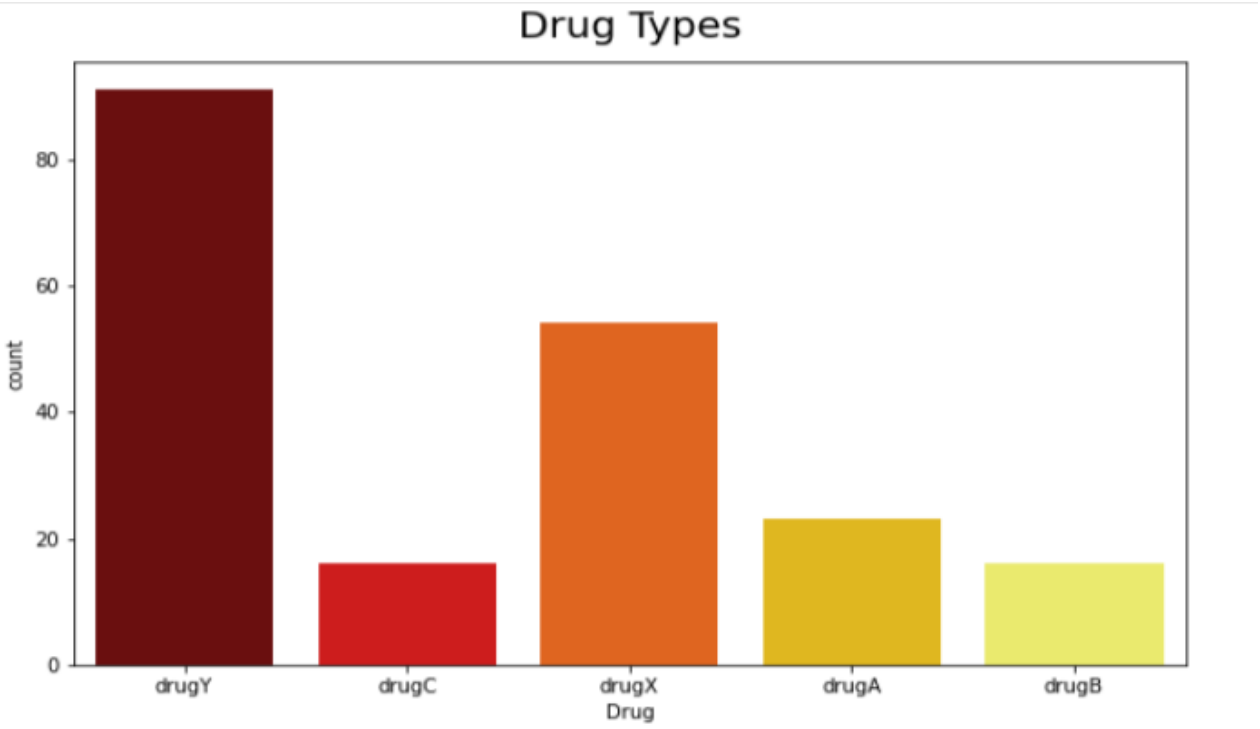
The below analyzed data are from a scientific research compiled for a medical study. They have been collected from a set of patients, all of whom suffered from the same illness. During their course of treatment, each patient responded to one of 5 medications, Drug A, Drug B, Drug c, Drug x and y.

Our job is to build three different classification models to find out which one of them predicts with the highest accuracy the drug that is appropriate for a future patient with the same illness. The features of this dataset are Age, Sex, Blood Pressure, and the Cholesterol of the patients, and the target is the drug that each patient responded to.

It is a sample of multiclass classifier, and we will use the training part of the dataset to build 1) Decision tree, 2) Random Forest and 3) Logistic Regression and then use it to predict the class of an unknown patient, or to prescribe a drug to a new patient. Our models will be focused on **prediction** rather than interpretation cause we have a simple dataset with few variables. The dataset is available on Kaggle ('/kaggle/input/drug-classification/drug200.csv').

Exploratory Data Analysis

We cleaned our data finding missing values and selected the *features Age, Sex, Blood Pressure, Sodium to Potassium ratio in patient's blood and Cholesterol* as the independent variables to train the models after conducting correlation and causation tests. Our dependent variable is *Drug*. Also we transformed our object data types (Sex,BP,Cholesterol) into numerical ones to be used from our machine learning models with LabelEncoder. Although our target variable is little imbalanced we did not performed a class weight or resampling method on the drug class cause the accuracy of the three models was quite high.



For multiclass classification, all three models—RandomForestClassifier, LogisticRegression, and DecisionTreeClassifier—are capable of handling multiclass problems. However, their performance can vary based on the dataset and the specific problem. Here's a general overview of their suitability for multiclass classification:

- RandomForestClassifier:

Pros: Random Forest is usually one of the best performers for multiclass classification because it can capture complex relationships in the data through ensemble learning.

Cons: Can be computationally expensive and slower to train as the number of trees and the dataset size increases.

Best Use Case: When you need high accuracy and have enough computational resources

- LogisticRegression:

Pros: It works well for linearly separable data and can handle multiclass classification using the one-vs-rest (OvR) or one-vs-one (OvO) strategies. It is also relatively simple and efficient.

Cons: It struggles when the decision boundaries are complex, as it is based on a linear model.

Best Use Case: When the data is relatively simple or when you need a fast, interpretable model with decent performance.

- DecisionTreeClassifier:

Pros: Decision Trees can model complex decision boundaries, making them a good choice for many types of data. They're easy to interpret and visualize.

Cons: They can overfit if not tuned properly (e.g., with max depth or pruning), and they may not generalize well on unseen data.

Best Use Case: When you need an interpretable model that can handle both numerical and categorical features.

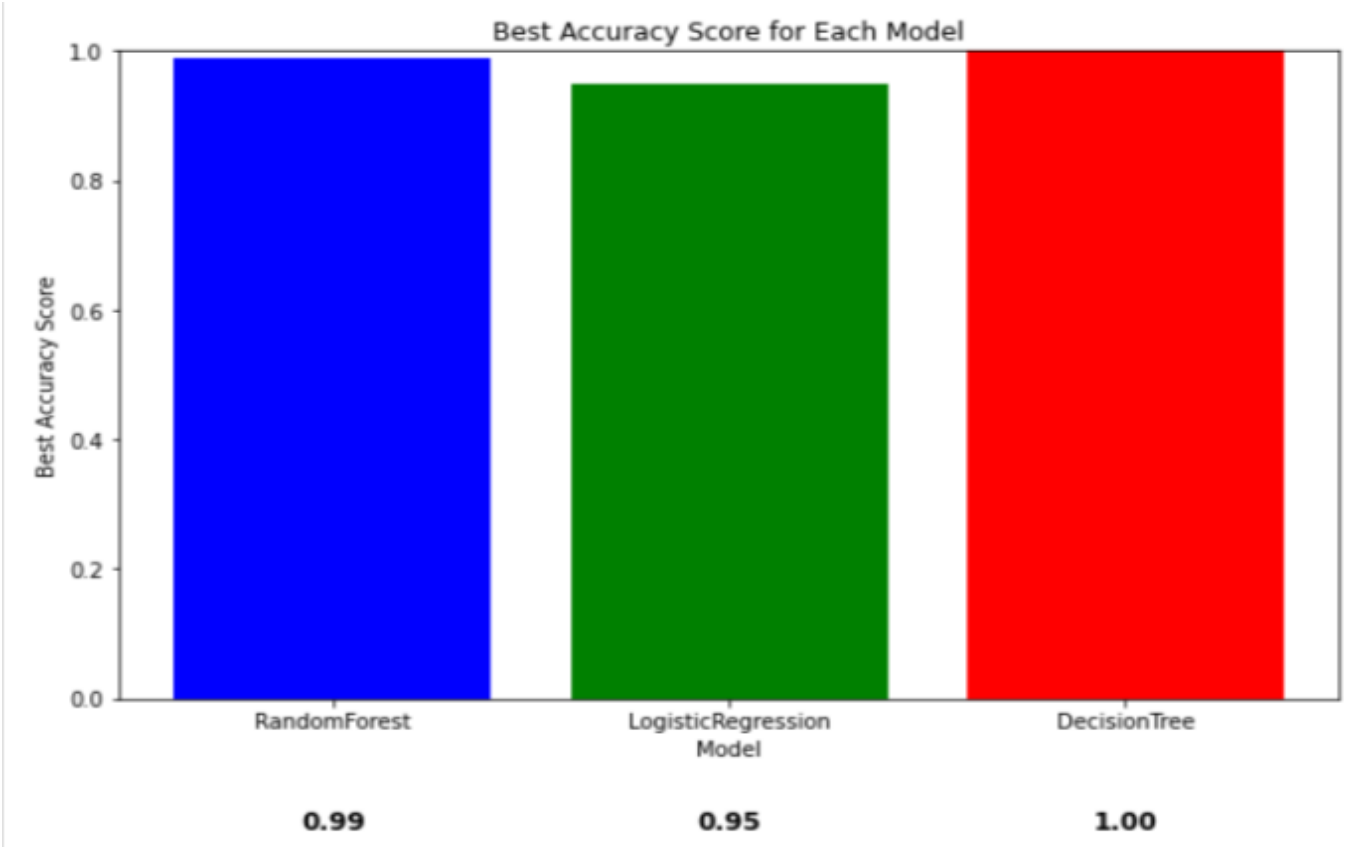
RandomForestClassifier is typically the best choice for most multiclass classification tasks because it combines the power of multiple decision trees, which helps to improve generalization and reduce overfitting.

LogisticRegression is effective for simpler problems with linear decision boundaries, but might not perform as well as Random Forest for more complex patterns.

DecisionTreeClassifier can work well but requires more careful tuning to avoid overfitting.

Conclusions:

For most multiclass classification tasks, RandomForestClassifier tends to be the most robust choice, especially when you have a complex dataset with non-linear decision boundaries, but as we observe from the below image has the best accuracy score 100%, followed by RandomForest 99% and LogistiRegerssion 95%, cause here we have s simple dataset with few predictors-variables and clear relationships between them and the tagret variable. We found the best hyperparameters for each model by tuning them with GridSearchCV in Python. You can see the detailed code in the attached .ipynb file.


Best Accuracy Score for Each Model
Best Accuracy Score
1.0
0.8
0.6
0.4
0.2
0.0
RandomForest
LogisticRegression
DecisionTree
Model

0.99 **0.95** **1.00**