Project Name - IRIS FLOWER Classification Model

Project Summary

The Iris Flower project is a classic machine learning project that involves the classification of iris flower species based on their sepal length, sepal width, petal length, and petal width. The dataset used in this project contains 150 instances with 4 attributes each, where each instance corresponds to a particular flower sample.

The main steps involved in this project are:

- Importing the necessary libraries and loading the dataset into a pandas DataFrame.
- Data wrangling and exploratory data analysis (EDA), which involves checking for missing values, duplicates, and outliers, visualizing the distribution of each feature, and identifying any correlation between the features.
- Feature selection and engineering, which involves selecting the relevant features that contribute most to the classification task and creating new features that may improve the performance of the model.
- Splitting the dataset into training and testing sets.
- Building a classification model, which can be done using various machine learning algorithms such as Logistic Regression, Support Vector Machines (SVM), Decision Trees, Random Forests, or KNN.
- Evaluating the model's performance using various metrics such as accuracy, precision, recall, F1 score, and ROC-AUC score.
- Improving the model's performance by tuning hyperparameters with the help of grid search cross-validation.

EDA Insights:

- Distribution of target variable: The target variable of the iris flower classification model is the species of the flower (Setosa, Versicolor, or Virginica). One of the first steps in EDA would be to examine the distribution of the target variable in the dataset. This could involve creating a histogram or bar chart of the species counts to ensure that the classes are relatively balanced and to check for any potential class imbalance issues.

- Correlation between features: The iris dataset contains measurements of the sepal length, sepal width, petal length, and petal width of each flower. EDA could involve calculating pairwise correlations between these features to identify any strong correlations between them. For example, the length and width of the petals may be highly correlated, while the sepal length and petal length may be less correlated.
- Distribution of features: EDA could also involve examining the distribution of each feature in the dataset. This could involve creating histograms or density plots of each feature to check for normality or skewness. If any features are skewed, transformations such as log or square root transformations could be applied to make them more normally distributed.
- Box plots: Another useful visualization for EDA is the box plot, which can be used to identify any outliers in the dataset. Outliers may need to be removed or dealt with in some other way to ensure that they do not negatively impact the performance of the classification model.

ML Results:

- Feature Importance: By analyzing the feature importance of the model, we can understand which physical characteristics of iris flowers are most important in determining their species. In this project, we found that petal length and petal width are the two most important features for iris flower classification.
- Model Performance: By evaluating the performance of different machine learning models, we can understand which algorithms work best for the Iris Flower dataset. In this project, we found that decision tree classifier, random forest classifier, and support vector machine classifier perform best for iris flower classification.
- Data Preprocessing: By performing data preprocessing techniques such as data cleaning, data transformation, and feature selection, we can improve the performance of the machine learning models. In this project, we used techniques such as removing outliers, scaling the data, and performing feature selection using the VIF method to improve the model's performance.
- Cross-validation: By performing cross-validation, we can evaluate the performance of the model on multiple splits of the data. In this project, we used 3-fold cross-validation to evaluate the performance of the models.
- We achieve highest accuracy 1.0 by Random Forest Classification Algorithm.