**Classification with Random Forest**

**Explanation of the code:**

1. **Importing libraries:** The required libraries for data manipulation, visualization, and machine learning are imported.
2. **Loading the dataset:** The dataset is loaded from a CSV file using the Pandas library. The na\_values parameter is used to handle any missing values in the dataset.
3. **Handling missing values:** The code checks for missing values in the dataset and sums up the missing values for each column.
4. **Exploratory data analysis (EDA):** Descriptive statistics and information about the dataset, such as data types of each column, are provided.
5. **Data visualization:** Plots are created to visualize the distribution of various columns in the dataset using histograms and a pie chart.
6. **Preprocessing:** Missing values are dropped from the dataset. The features (all columns except the target column) are assigned to the input variable X, and the target variable is assigned to the output variable y. The dataset is then split into training and testing sets.
7. **Random Forest Classifier:** The Random Forest classifier is imported from the scikit-learn library. An instance of the classifier is created with specified hyperparameters such as the number of estimators, maximum depth, minimum samples required to split a node, and random state. The model is then trained on the training data.
8. **Model evaluation:** The accuracy of the model is calculated on the test set to evaluate its performance.
9. **Making predictions:** The model is used to make predictions on the test set, and the actual and predicted values are stored for comparison.
10. **Making a single prediction:** The trained model is used to predict the class for a single sample by providing its feature values.