PROJECT: IRIS FLOWER CLASSIFICATION

Project Problem Statement:

The Iris flower dataset consists of three species: setosa, versicolor, and virginica. These species can be distinguished based on their measurements. Now, imagine that you have the measurements of Iris flowers categorized by their respective species. Your objective is to train a machine learning model that can learn from these measurements and accurately classify the Iris flowers into their respective species.

Use the Iris dataset to develop a model that can classify iris flowers into different species based on their sepal and petal measurements. This dataset is widely used for introductory classification tasks.

PROJECT SOLUTION DESCRIPTION

➔ Technologies, Language, and Libraries Used

The technologies, languages, and libraries used in the project is mentioned below:

Technologies and Languages:

1. Python: The programming language used for writing the code.
2. Jupyter Notebook: The environment in which this Python code is likely executed.

Libraries and Modules:

1. **Pandas (`pd`):** For data manipulation and analysis. Used to read the CSV file and handle data.
2. **NumPy (`np`):** For numerical operations. Used here primarily for array manipulations.
3. **Matplotlib (`plt`):** For creating static, animated, and interactive visualizations. Used for plotting graphs.
4. **Seaborn (`sns`):** For statistical data visualization. Used for creating various types of plots.
5. **Scikit-learn (`sklearn`):**
   a. **LabelEncoder:** For encoding categorical labels.
   b. **train_test_split:** For splitting data into training and test sets.
   c. **StandardScaler:** For feature scaling.
   d. **SVC:** Support Vector Classification for building the classification model.
   e. **classification_report & confusion_matrix:** For evaluating the model's performance.
6. **Warnings:** Used to filter out user warnings during execution.

Features and Functions Used:

1. **Data Loading and Manipulation:** `pd.read_csv()`, `iris_df.info()`, `iris_df.describe()`
2. **Data Encoding:** `LabelEncoder()`
3. **Data Filtering:** Filtering based on species.

4. **Data Visualization:** `sns.heatmap()`, `sns.pairplot()`, `sns.violinplot()`, `sns.countplot()`, `plt.show()`
5. **Model Building and Evaluation:** `train_test_split()`, `StandardScaler()`, `SVC()`, `confusion_matrix()`, `classification_report()`
6. **User Input Handling:** Using `input()` for interactive data entry.

➔ Workflow Description

1. Data Import and Exploration

- The script starts by importing necessary libraries and reading the Iris dataset from a CSV file ('IRIS.csv') into a Pandas DataFrame.
- It displays basic information about the dataset using `info()` and summary statistics using `describe()`.

2. Data Preprocessing:

- Label encoding is applied to convert the categorical species labels (setosa, versicolor, virginica) into numerical values.
- The dataset is then divided into separate DataFrames for each species (iris_seto, iris_vers, iris_virg).
- Percentage calculations show the distribution of each species in the dataset.

3. Data Visualization:

Several data visualizations are created to explore relationships between features:

- A correlation heatmap is generated to visualize feature correlations.
- A pairplot displays pairwise scatterplots colored by species for initial data exploration.
- Violin plots show the distribution of sepal and petal measurements by species.
- A countplot displays the count of each species.

4. Data Splitting:

- The dataset is split into training and testing sets using `train_test_split()`.
- Standardization is applied to scale the feature values using `StandardScaler`.

5. Model Training:

A Support Vector Machine (SVM) model, best for such classification problems, is created and trained using the IRIS training data.

6. Model Evaluation:

- Model predictions are made on the test data, and a confusion matrix and classification report are generated using `classification_report()` and `confusion_matrix()`.
- A heatmap visualizes the confusion matrix.

7. User Interaction:

- The script allows the user to input sepal and petal measurements for an iris flower.
- The user's input is used to make a prediction using the trained SVM model.

- The predicted species (setosa, versicolor, virginica) is displayed to the user.

8. **Warnings Suppression:**

User warnings are suppressed using `warnings.filterwarnings()` to prevent displaying warnings related to feature names.

➔ Conclusion

Overall, the code demonstrates a complete workflow for data preprocessing, model training, evaluation, and user interaction for classifying Iris flowers based on their sepal and petal measurements using a Support Vector Machine (SVM) model. It also includes visualizations to help understand the dataset and model performance.