IRIS FLOWER CLASSIFICATION

INTRODUCTION:

The Iris flower classification problem is a well-known machine learning task in the field of pattern recognition and classification. It involves predicting the species of an Iris flower based on certain measurements of its petals and sepals.

The Iris dataset, introduced by British biologist and statistician Ronald Fisher in 1936, contains measurements of three species of Iris flowers: setosa, versicolor, and virginica. The dataset includes four features: sepal length, sepal width, petal length, and petal width.

The primary goal is to build a model that can accurately classify Iris flowers into one of the three species based on these four features.

There are various machine learning algorithms that can be used to solve the Iris flower classification problem, including but not limited to:

1. **K-Nearest Neighbors (KNN)**: This algorithm classifies a data point by finding the majority class among its K nearest neighbors in the feature space.
2. **Support Vector Machines (SVM)**: SVM aims to find the hyperplane that best separates different classes in the feature space.
3. **Decision Trees**: These are tree-like structures where internal nodes represent the features, branches represent decisions, and leaf nodes represent the class labels.
4. **Random Forest**: A collection of decision trees where each tree is built from a subset of the data and features, and the final prediction is made by averaging the predictions of individual trees.
5. **Logistic Regression**: Despite its name, logistic regression is used for classification. It models the probability of a certain class using a logistic function.
6. **Neural Networks**: Deep learning models like neural networks can also be used for Iris classification tasks, employing multiple layers of interconnected nodes to learn complex patterns in the data.

The Iris dataset is commonly used to demonstrate and compare the performance of different machine learning algorithms due to its simplicity and availability. It's often used as a beginner's dataset for learning classification techniques and assessing model performance.

To work on the Iris classification problem, data preprocessing, model selection, hyperparameter tuning, and evaluation of model performance using techniques like cross-validation and metrics such as accuracy, precision, recall, and F1-score are essential steps.

Researchers and practitioners use this dataset not only for educational purposes but also as a benchmark to test the effectiveness of new algorithms or methodologies in the field of machine learning and pattern recognition.

LOADING DATASET:

Step 1 – Load the data

- Pandas help to load data from various sources like local storage, database, excel file, CSV file, etc.

PROGRAM

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy_score, classification_report

from sklearn.datasets import load_iris

iris = load_iris()

X = iris.data  # Features (sepal length, sepal width, petal length, petal width)

y = iris.target  # Target variable (species)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)

knn = KNeighborsClassifier(n_neighbors=3) # You can adjust the number of neighbors as needed

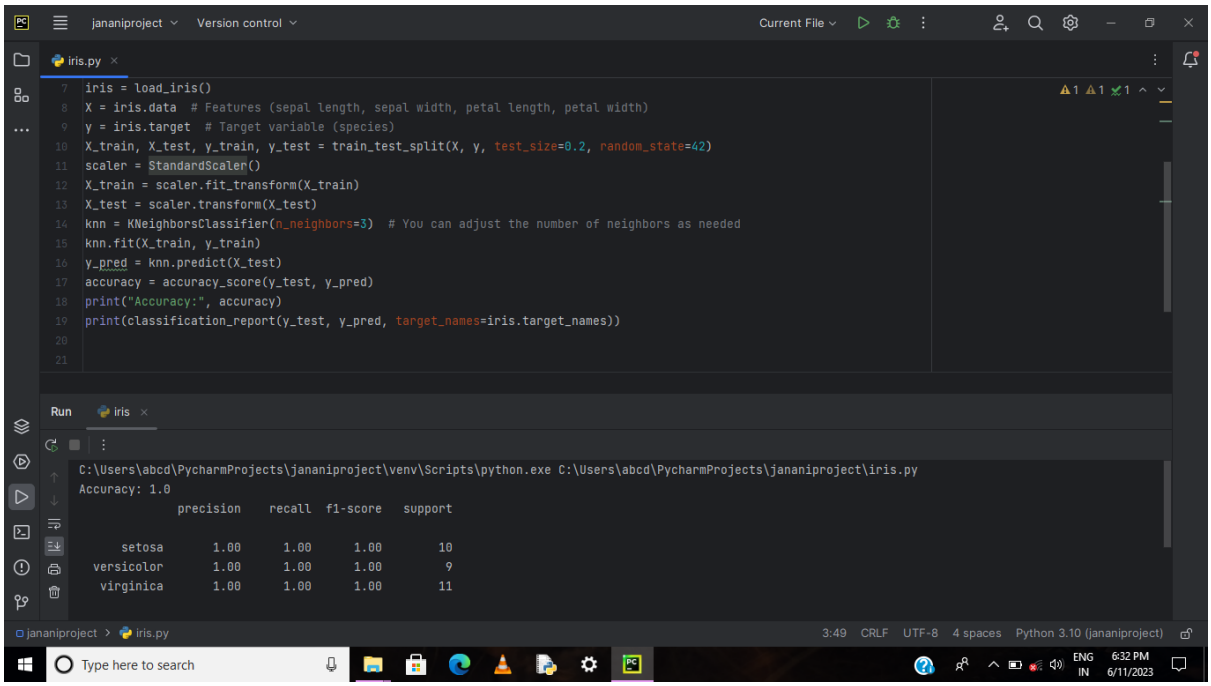
knn.fit(X_train, y_train)

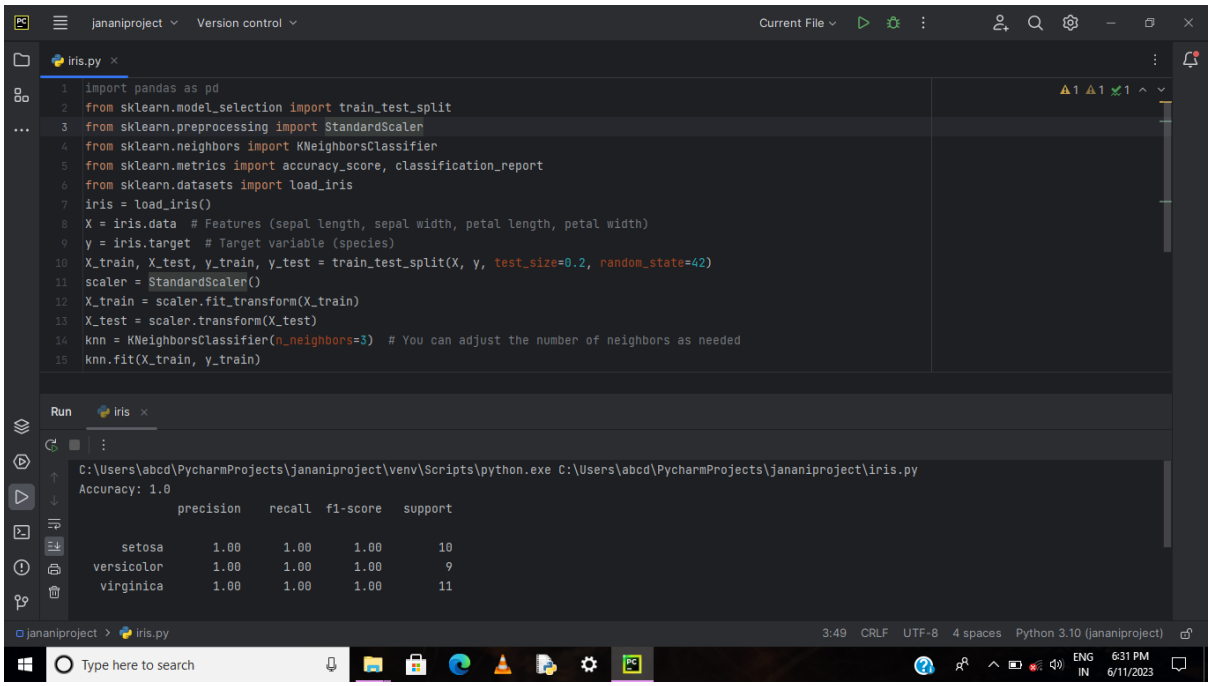
y_pred = knn.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

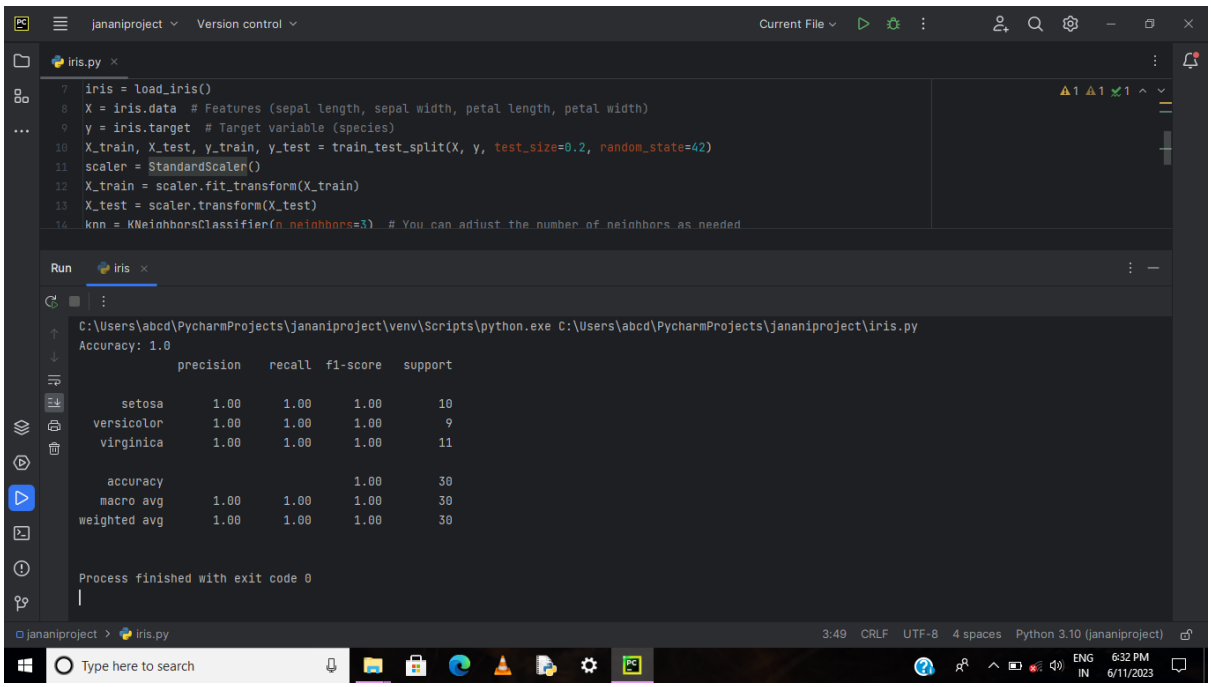
print("Accuracy:", accuracy)

print(classification_report(y_test, y_pred, target_names=iris.target_names))





OUTPUT



CONCLUSION

This is the program of iris flower classification. And output is here.