Hello World of Machine Learning

The best small project to start with on a new tool is the classification of iris flowers (e.g. the iris dataset).

- Attributes are numeric so you have to figure out how to load and handle data.
- It is a classification problem, allowing you to practice with perhaps an easier type of supervised learning algorithm.
- It is a multi-class classification problem (multi-nominal) that may require some specialized handling.
- It only has 4 attributes and 150 rows, meaning it is small and easily fits into memory (and a screen or A4 page).
- All of the numeric attributes are in the same units and the same scale, not requiring any special scaling or transforms to get started.

To do

1. Installing the Python and SciPy platform.
   pip install numpy pandas matplotlib seaborn scikit-learn
2. Loading the dataset.
3. Summarizing the dataset.
   - Dimensions of the dataset.
   - Peek at the data itself.
   - Statistical summary of all attributes.
   - Breakdown of the data by the class variable.
4. Visualizing the dataset.
   - Univariate plots to better understand each attribute.
   - Multivariate plots to better understand the relationships between attributes.
5. Evaluating some algorithms.
   - Separate out a validation dataset.
   - Set-up the test harness to use 10-fold cross validation.
   - Build multiple different models to predict species from flower measurements
   - Select the best model.

     test 6 different algorithms:

     - Logistic Regression (LR)
     - Linear Discriminant Analysis (LDA)
     - K-Nearest Neighbors (KNN).
     - Classification and Regression Trees (CART).
     - Gaussian Naive Bayes (NB).
     - Support Vector Machines (SVM).
6. Making some predictions.

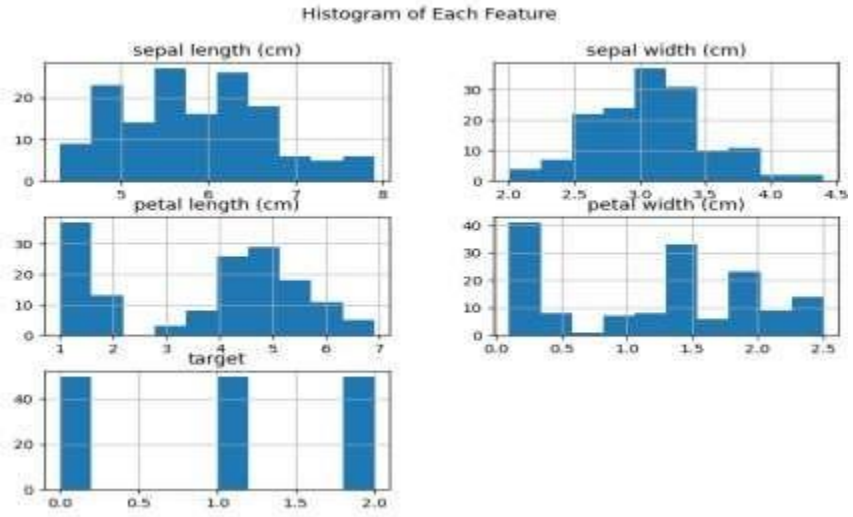
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split, KFold, cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.neighbors import KNeighborsClassifier
```

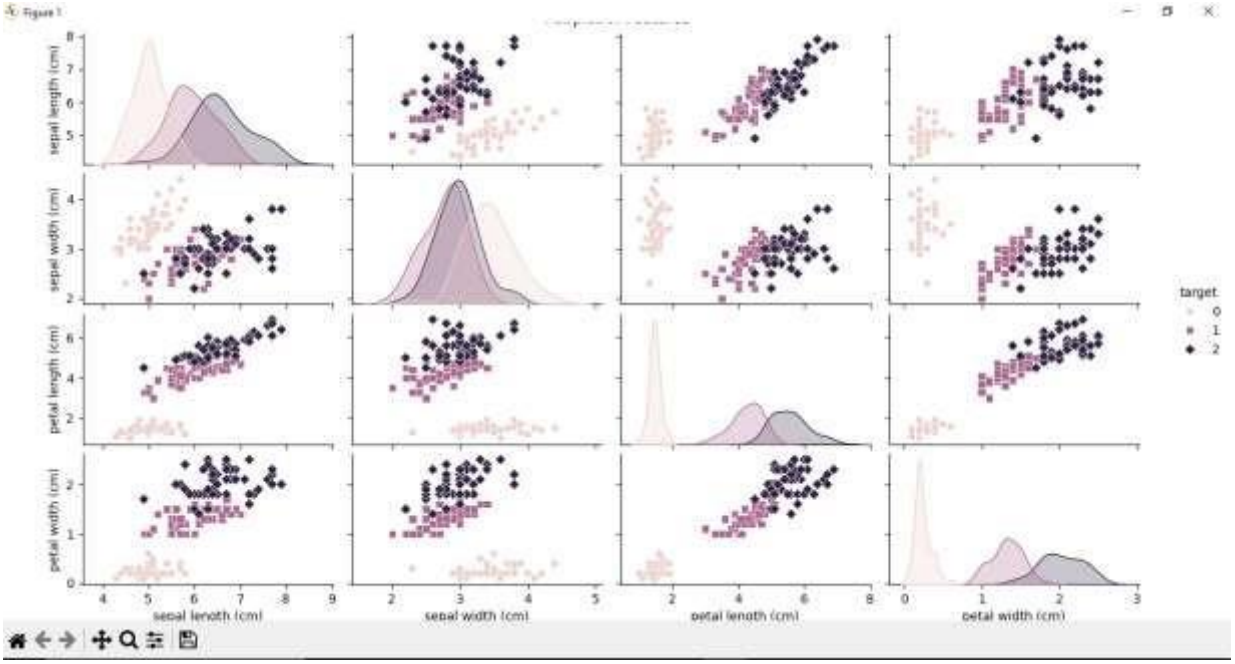
```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score

# Step 2: Load the Iris Dataset
iris = load_iris()
data = pd.DataFrame(data=iris.data, columns=iris.feature_names)
data['target'] = iris.target

# Step 3: Summarize the Dataset
# Check dimensions
print("Dataset Dimensions:", data.shape)

# Peek at the data
print("\nFirst 5 Rows of the Dataset:")
print(data.head())

# Statistical summary of all attributes
print("\nStatistical Summary:")
print(data.describe())

# Breakdown of the data by class variable
print("\nClass Distribution:")
print(data['target'].value_counts())

# Step 4: Visualize the Dataset
# Univariate Plots (Histograms)
data.hist(figsize=(10, 8))
plt.suptitle('Histogram of Each Feature')
plt.show()

# Multivariate Plots (Pairplot)
sns.pairplot(data, hue='target', markers=["o", "s", "D"])
plt.suptitle('Pairplot of Features', y=1.02)
plt.show()

# Step 5: Evaluate Algorithms
X = data.drop('target', axis=1)
y = data['target']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
kfold = KFold(n_splits=10, random_state=42, shuffle=True)
models = {
    'Logistic Regression (LR)': LogisticRegression(max_iter=200),
    'Linear Discriminant Analysis (LDA)': LinearDiscriminantAnalysis(),
    'K-Nearest Neighbors (KNN)': KNeighborsClassifier(),
    'Classification and Regression Trees (CART)': DecisionTreeClassifier(),
    'Gaussian Naive Bayes (NB)': GaussianNB(),
    'Support Vector Machines (SVM)': SVC()
}
results = {}
print("\nCross-Validation Results:")
for name, model in models.items():
```

```python
    cv_results = cross_val_score(model, X_train, y_train, cv=kfold, scoring='accuracy')
    results[name] = cv_results.mean()
    print(f"{name}: {cv_results.mean():.4f}")

# Step 6: Make Predictions
best_model_name = max(results, key=results.get)
best_model = models[best_model_name]
print(f"\nBest Model: {best_model_name}")
best_model.fit(X_train, y_train)
predictions = best_model.predict(X_test)
print("\nAccuracy Score on Test Set:")
print(accuracy_score(y_test, predictions))

print("\nClassification Report:")
print(classification_report(y_test, predictions))
```

```
PS C:\Users\Rudradeep\Desktop\Python\pythonCollage> python -u "c:\Users\Rudradeep\Desktop\Python\pythonCollage\Collage_python\Lab 6\1.py"
Dataset Dimensions: (150, 5)

First 5 Rows of the Dataset:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  target
0                5.1               3.5                1.4               0.2       0
1                4.9               3.0                1.4               0.2       0
2                4.7               3.2                1.3               0.2       0
3                4.6               3.1                1.5               0.2       0
4                5.0               3.6                1.4               0.2       0

Statistical Summary:
       sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)      target
count         150.000000        150.000000         150.000000        150.000000  150.000000
mean            5.843333          3.057333           3.758000          1.199333    1.000000
std             0.828066          0.435866           1.765298          0.762238    0.819232
min             4.300000          2.000000           1.000000          0.100000    0.000000
25%             5.100000          2.800000           1.600000          0.300000    0.000000
50%             5.800000          3.000000           4.350000          1.300000    1.000000
75%             6.400000          3.300000           5.100000          1.800000    2.000000
max             7.900000          4.400000           6.900000          2.500000    2.000000

Class Distribution:
target
0    50
1    50
2    50
Name: count, dtype: int64

Cross-Validation Results:
Logistic Regression (LR): 0.9500
Linear Discriminant Analysis (LDA): 0.9667
K-Nearest Neighbors (KNN): 0.9500
Classification and Regression Trees (CART): 0.9250
Gaussian Naive Bayes (NB): 0.9417
Support Vector Machines (SVM): 0.9583
```

```
Best Model: Linear Discriminant Analysis (LDA)

Accuracy Score on Test Set:
1.0

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30

PS C:\Users\Rudradeep\Desktop\Python\pythonCollage>
```


Histogram of Each Feature
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
target


sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
target
0
1
2