# import libraries

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, classification\_report

import matplotlib.pyplot as plt

import seaborn as sns

# load dataset

df = pd.read\_csv('/content/Iris.csv')

df.info()

# check for missing values

print(df.isnull().sum())

# visualizing each feature using histogram

df.hist(edgecolor='black', linewidth=1.2, figsize=(12, 8))

plt.suptitle('Distribution of Features in Iris Dataset')

plt.show()

# box plot to identify outliners

plt.figure(figsize=(12, 8))

sns.boxplot(data=df)

plt.title('Box Plot of Iris Dataset Features')

plt.show()

# split dataset

X = df.drop('Species',axis=1)

y = df['Species']

# split data into training and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size =0.2, random\_state = 42)

# implementing logistic regression

model = LogisticRegression(random\_state=42)

model.fit(X\_train, y\_train)

# making prediction on test set

y\_pred = model.predict(X\_test)

# evaluating model performance

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred, average='weighted')

recall = recall\_score(y\_test, y\_pred, average='weighted')

# printing evaluation metrics

print(f'Accuracy: {accuracy:.2f}')

print(f'Precision: {precision:.2f}')

print(f'recall: {recall:.2f}')

# scatter plot to explore relationship between features

sns.pairplot(df, hue='Species', markers = ['o','s','D'])

plt.suptitle('Pair plot dataset features')

plt.show()

# Violin plot for feature distribution

plt.figure(figsize=(12, 8))

sns.violinplot(x='Species', y='SepalLengthCm', data=df)

plt.title('Violin Plot of Sepal Length by Species')

plt.show()

# predict flower types

classification\_rep = classification\_report(y\_test, y\_pred)

print('Classification Report:')

print(classification\_rep)

# documentation

# 1. Import the required libraries and the iris dataset is loaded using Pandas

# 2. Get the summary of the dataset using info() function.

# 3. Clean the dataset by checking for missing values and address outliners.

# 4. Visualize each feature using histogram.

# 5. The dataset is then split into features and target variable

# 6. Using the train\_test\_split function from scikit-learn create training and test sets(80% trainset, 20% testset).

# 7. Logistic regression algorithm is used for its simplicity and interpretability

# 8. The model is trained using training set. It is evaluated using accuracy, precision, recall on tesing set

# 9. Scatter plot, box plot and voilin plot is used to explore the relationship between different features and their distribution.

# 10. Used all available features for simplicity.