LAB :LOGISTIC REGRESSION

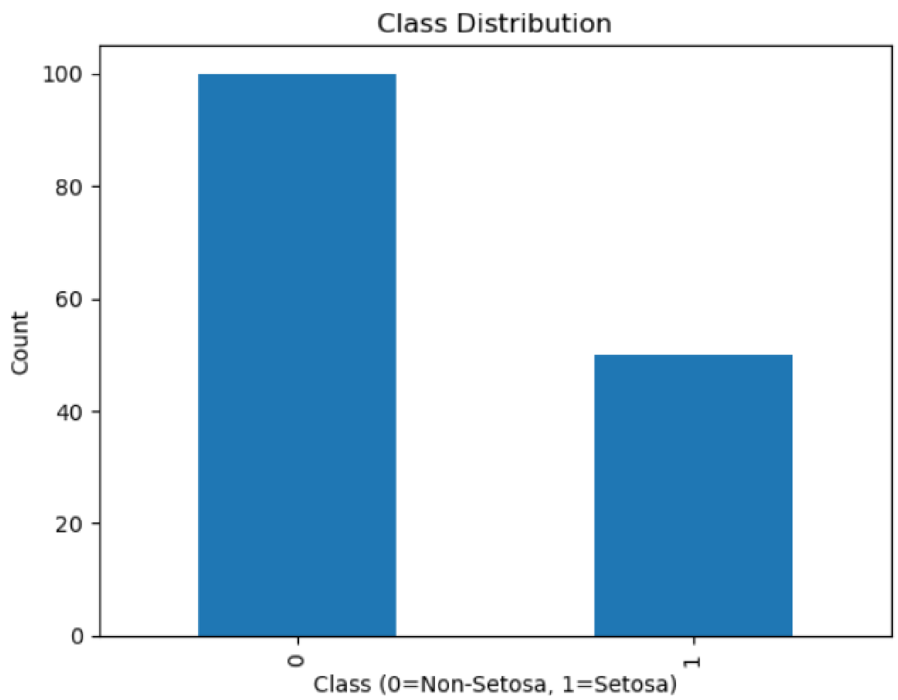
Lab – Logistic Regression

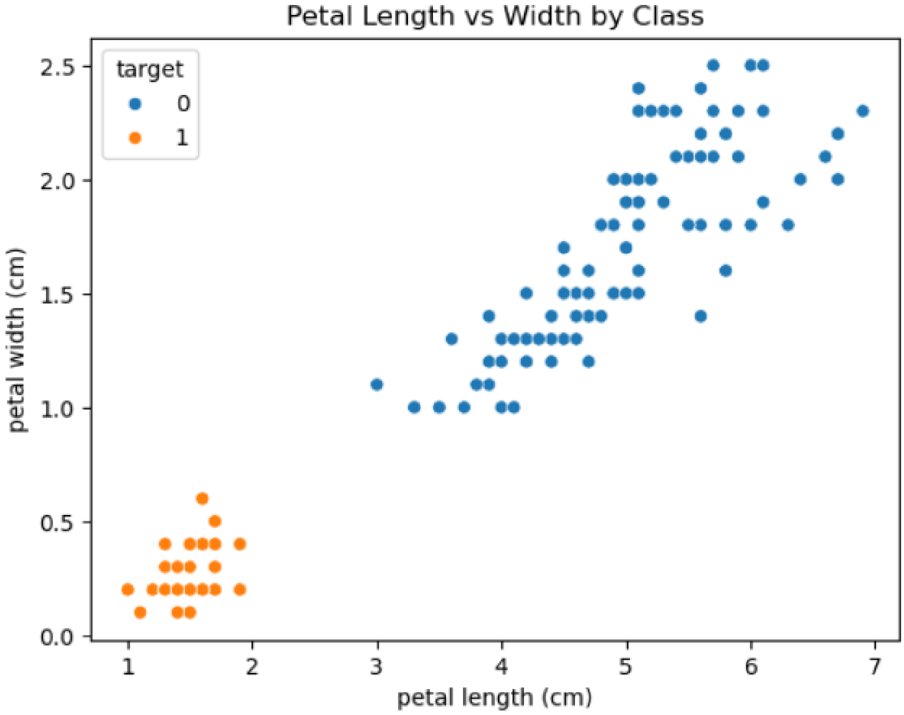
- **Problem:** Classify Iris flowers as Setosa vs Non-Setosa
  - Link: https://www.kaggle.com/datasets/uciml/iris
- **Task B1: Data Loading and Preparation**
  - Load the Iris dataset from sklearn
  - Convert the 3-class problem to binary: Setosa (1) vs Others (0)
  - How many samples are in each class? Is the dataset balanced?
- **Task B2: Exploratory Data Analysis**
  - Create bar plot of class distribution
  - Create scatter plot of petal length vs petal width, colored by class
  - Compare feature means between the two classes
    - Which features seem most different between Setosa and non-Setosa flowers?
    - Based on the scatter plot, do you expect the classification to be easy or difficult?
- **Task B3: Model Building and Evaluation**
  - Split data into training (70%) and testing (30%) sets
  - Train a Logistic Regression model
  - Make predictions and calculate accuracy
  - ***Create confusion matrix – Self research!***
  - Examine feature coefficients (importance)
  - What is your model's accuracy?
  - Which feature has the largest absolute coefficient? What does this mean?
  - Looking at the confusion matrix, which class does the model predict better?

[1]:
```python
import pandas as pd
from sklearn.datasets import load_iris

# Load Iris dataset
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)

# Convert target to binary: Setosa (1) vs Others (0)
df['target'] = (iris.target == 0).astype(int)

print(df['target'].value_counts())  # Check class distribution
```

```
target
0    100
1     50
Name: count, dtype: int64
```

[2]:
```python
import matplotlib.pyplot as plt
import seaborn as sns

# Bar plot of class distribution
df['target'].value_counts().plot(kind='bar')
plt.xlabel('Class (0=Non-Setosa, 1=Setosa)')
plt.ylabel('Count')
plt.title('Class Distribution')
plt.show()

# Scatter plot petal length vs petal width, colored by class
sns.scatterplot(data=df, x='petal length (cm)', y='petal width (cm)', hue='target')
plt.title('Petal Length vs Width by Class')
plt.show()

# Compare means of features between the two classes
print(df.groupby('target').mean())
```



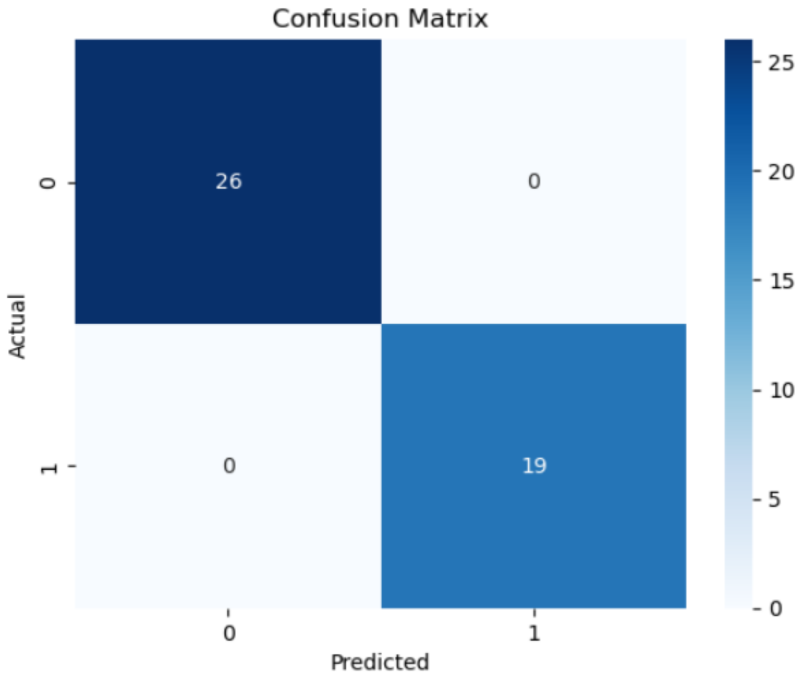



```
        sepal length (cm)  sepal width (cm)  petal length (cm)  \
target
0                   6.262             2.872              4.906
1                   5.006             3.428              1.462

        petal width (cm)
target
0                  1.676
1                  0.246
```

[3]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns

X = df.drop('target', axis=1)
y = df['target']

# Split 70% training, 30% testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Train logistic regression model
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)

# Predictions and accuracy
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

# Confusion matrix heatmap
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

# Feature coefficients
coeff_df = pd.DataFrame({'Feature': X.columns, 'Coefficient': model.coef_[0]})
print(coeff_df)

# Feature with largest absolute coefficient
largest_coef = coeff_df.loc[coeff_df['Coefficient'].abs().idxmax()]
print(f"Largest absolute coefficient feature: {largest_coef['Feature']} ({largest_coef['Coefficient']:.3f})")
```

```
Accuracy: 1.00
```




```
           Feature  Coefficient
0  sepal length (cm)    -0.431077
1   sepal width (cm)     0.845708
2  petal length (cm)    -2.156580
3   petal width (cm)    -0.889408
Largest absolute coefficient feature: petal length (cm) (-2.157)
```

Model Interpretation

- Coefficients indicate the importance and direction of features in deciding the class (Setose vs non-setose).
- Largest absolute coefficient means the most influential feature for classification.
- Confusion matrix diagonals show correct predictions; the class with more correct predictions is modelled better.
- Accuracy gives the overall success rate of prediction.