Machine Learning Lab (PMCA507P)

> **Reg No:** 23MCA1030
>
> **Name :** Vinayak Kumar Singh

Exercise 7a : Linear SVM

Collab url : https://colab.research.google.com/drive/1Nn6hweXLGDqth0tApZ6LuB0FnMCGPwWf?usp=sharing

Dataset url : https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database/data

Pima Indians Diabetes Database

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

```
#Load the dataset
data = pd.read_csv('/content/diabetes.csv')
```

```
# Print the first few rows of the dataset
print("First few rows of the dataset:")
print(data.head())
```

```
First few rows of the dataset:
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6   
1            1       85             66             29        0  26.6   
2            8      183             64              0        0  23.3   
3            1       89             66             23       94  28.1   
4            0      137             40             35      168  43.1   

   DiabetesPedigreeFunction  Age  Outcome  
0                     0.627   50        1  
1                     0.351   31        0  
2                     0.672   32        1  
3                     0.167   21        0  
4                     2.288   33        1  
```

```
# Separate features and target variable
features = data.drop('Outcome', axis=1)
target = data['Outcome']
```

```
# Print the column names of features and target variable
print("\nFeature columns:")
for column in features.columns:
    print(column)

print("\nTarget variable:")
print(target.name)
```

```

Feature columns:
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age

Target variable:
Outcome
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)
```

```
# Print the shapes of training and testing sets
print("\nShapes of training and testing sets:")
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
Shapes of training and testing sets:
X_train shape: (614, 8)
X_test shape: (154, 8)
y_train shape: (614,)
y_test shape: (154,)
```

```
# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Initialize the linear SVM classifier
linear_svm = LinearSVC(penalty='l2', dual=True, C=1.0)

# Train the model
linear_svm.fit(X_train_scaled, y_train)

# Make predictions on the test set
y_pred = linear_svm.predict(X_test_scaled)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1244: ConvergenceWarning: Liblinear failed to converge, increase the n
  warnings.warn(
```

```
# Evaluate the model
metrics = ['Accuracy', 'Precision', 'Recall', 'F1-score']
scores = [accuracy_score(y_test, y_pred),
           precision_score(y_test, y_pred),
           recall_score(y_test, y_pred),
           f1_score(y_test, y_pred)]

print("\nModel evaluation metrics:")
for metric, score in zip(metrics, scores):
    print(f"{metric}: {score:.2f}")
```

```
Model evaluation metrics:
Accuracy: 0.75
Precision: 0.65
Recall: 0.67
F1-score: 0.66
```

```
# Interpret the model
print("\nFeature weights:")
for idx, weight in enumerate(linear_svm.coef_[0]):
    print(f"Feature {features.columns[idx]}: {weight:.2f}")

print(f"\nIntercept: {linear_svm.intercept_[0]:.2f}")
```

```
Feature weights:
Feature Pregnancies: 0.08
Feature Glucose: 0.40
Feature BloodPressure: -0.09
Feature SkinThickness: 0.02
Feature Insulin: -0.07
Feature BMI: 0.28
Feature DiabetesPedigreeFunction: 0.08
Feature Age: 0.16

Intercept: -0.33
```