

```
In [1]: import numpy as np
import pandas as pd
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler

X_data_train = pd.read_csv("bonus/train_input_data.txt", sep = ' ')
y_data_train = pd.read_csv("bonus/train_output_data.txt")

X_train = X_data_train.to_numpy()
y_train = y_data_train.to_numpy().flatten()
y_train_svm = np.where(y_train > 0.5, 1, -1)

X_data_test = pd.read_csv("bonus/test_input_data.txt", sep = ' ')
y_data_test = pd.read_csv("bonus/test_output_data.txt")

X_test = X_data_test.to_numpy()
y_test = y_data_test.to_numpy().flatten()
y_test_svm = np.where(y_test > 0.5, 1, -1)
```

```
In [2]: LR = make_pipeline(StandardScaler(), LogisticRegression(C = 100))
LR.fit(X_train, y_train)

LR_train_accuracy = LR.score(X_train, y_train) * 100
print("\nAccuracy of Logistic Regression model for training data: ", LR_train_accu

LR_test_accuracy = LR.score(X_test, y_test) * 100
print("Accuracy of Logistic Regression model for test data: ", LR_test_accuracy)
```

Accuracy of Logistic Regression model for training data: 87.98107897324607

Accuracy of Logistic Regression model for test data: 88.09523809523809

```
In [ ]: svm = make_pipeline(StandardScaler(), SVC(kernel = 'linear', gamma = 'scale'))
svm.fit(X_train, y_train_svm)

svm_train_accuracy = svm.score(X_train, y_train_svm) * 100
print("\nAccuracy of SVM with linear kernel for training data: ", svm_train_accurac

svm_test_accuracy = svm.score(X_test, y_test_svm) * 100
print("Accuracy of SVM with linear kernel for test data: ", svm_test_accuracy)

svm = make_pipeline(StandardScaler(), SVC(kernel = 'sigmoid', gamma = 'scale'))
svm.fit(X_train, y_train_svm)

svm_train_accuracy = svm.score(X_train, y_train_svm) * 100
print("\nAccuracy of SVM with sigmoid kernel for training data: ", svm_train_accura

svm_test_accuracy = svm.score(X_test, y_test_svm) * 100
print("Accuracy of SVM with sigmoid kernel for test data: ", svm_test_accuracy)

svm = make_pipeline(StandardScaler(), SVC(kernel = 'poly', gamma = 'scale'))
svm.fit(X_train, y_train_svm)
```

```
svm_train_accuracy = svm.score(X_train, y_train_svm) * 100
print("\nAccuracy of SVM with polynomial kernel for training data: ", svm_train_acc

svm_test_accuracy = svm.score(X_test, y_test_svm) * 100
print("Accuracy of SVM with polynomial kernel for test data: ", svm_test_accuracy)

svm = make_pipeline(StandardScaler(), SVC(kernel = 'rbf', gamma = 'scale'))
svm.fit(X_train, y_train_svm)

svm_train_accuracy = svm.score(X_train, y_train_svm) * 100
print("\nAccuracy of SVM with rbf kernel for training data: ", svm_train_accuracy)

svm_test_accuracy = svm.score(X_test, y_test_svm) * 100
print("Accuracy of SVM with rbf kernel for test data: ", svm_test_accuracy)
```

Accuracy of SVM with linear kernel for training data: 88.07432776101184

Accuracy of SVM with linear kernel for test data: 88.26312576312576