**Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**UCS2612 Machine Learning Laboratory**

**Assignment 4**

**Name:** V.S. Pranav **Reg No:** 3122 21 5001 069

**Classification of Email spam and MNIST data**

**Github Link:**

1. Develop a python program to classify Emails as Spam or Ham using Support Vector Machine (SVM) Model. Visualize the features from the dataset and interpret the results obtained by the model using Matplotlib library.

**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

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

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score, confusion\_matrix

import seaborn as sns

df = pd.read\_csv('email.csv')

# Visualize the features from the dataset

df.iloc[::,-1].value\_counts().plot(kind='bar')

# Preprocessing

X = df.iloc[::,:-1]

y = df.iloc[::,-1]

labels = ['Spam','Ham']

explode=np.zeros(len(labels))

explode[-1]=0.1

plt.title("Ham vs Spam")

plt.pie(df.iloc[:,-1].value\_counts(), labels=labels, counterclock=False, shadow=True,

explode=explode, autopct='%1.1f%%', radius=1, startangle=0)

plt.show()

corr=df.corr()

plt.figure(figsize=(12, 10))

sns.heatmap(corr,

xticklabels=corr.columns,

yticklabels=corr.columns,

cmap=sns.diverging\_palette(250, 10, as\_cmap=True),

square=True,

linewidths=.1)

plt.show()

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

model\_linear = SVC(kernel='linear')

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

model\_rb = SVC(kernel='rbf')

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

model\_poly = SVC(kernel='poly')

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

model\_sig = SVC(kernel='sigmoid')

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

def pred\_plot(model):

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

print('Training Accuracy:', model.score(X\_train, y\_train))

print('Accuracy Score:', accuracy\_score(y\_test, y\_pred))

# Confusion matrix

confusion = confusion\_matrix(y\_test, y\_pred)

# Visualising the results

plt.matshow(confusion)

plt.colorbar()

plt.xlabel('Actual')

plt.ylabel('Predicted')

plt.show()

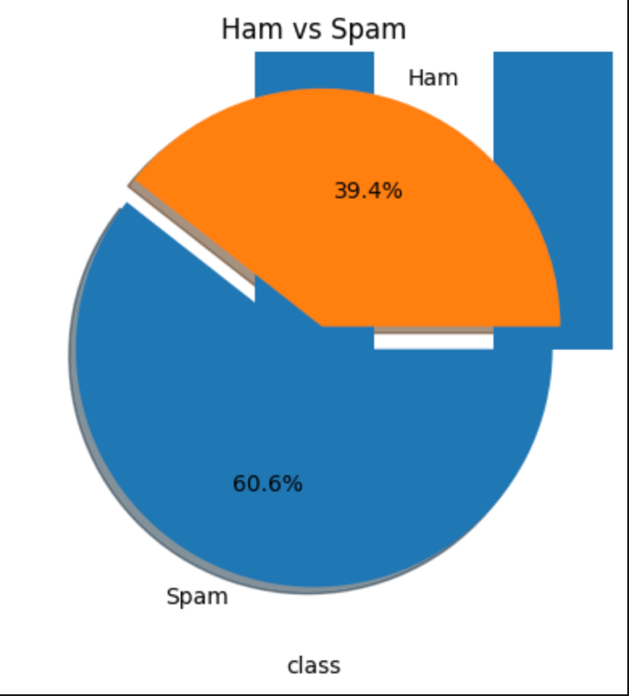
pred\_plot(model\_linear)

pred\_plot(model\_rb)

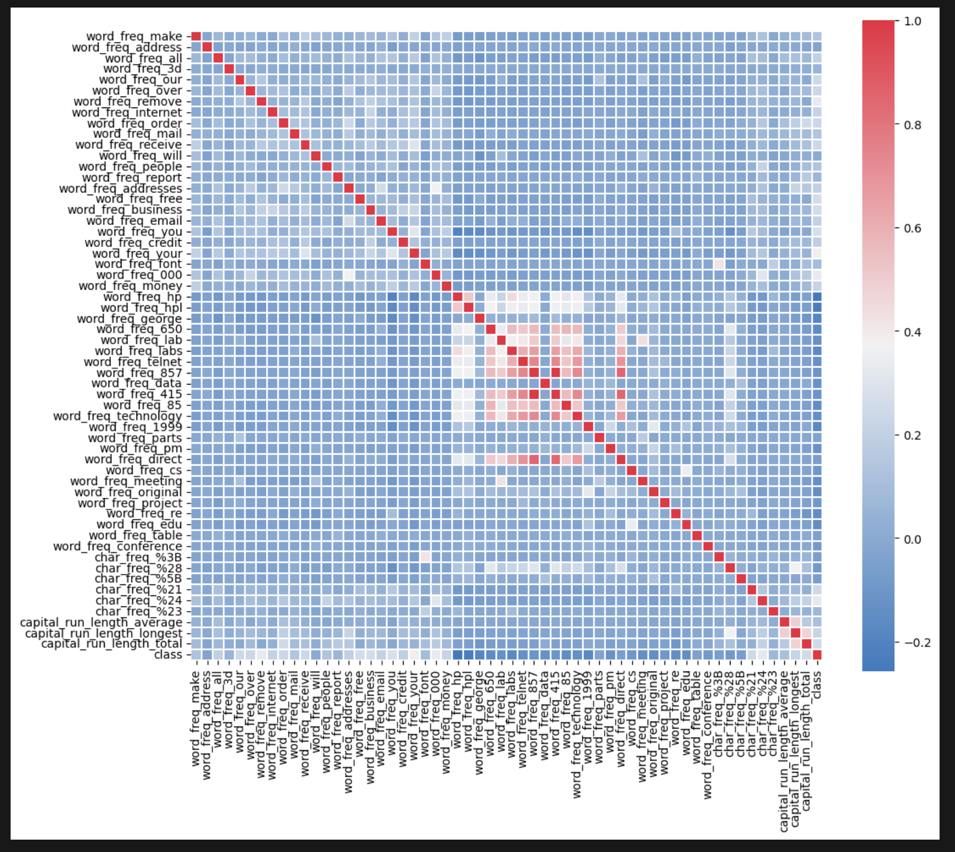
pred\_plot(model\_poly)

pred\_plot(model\_sig)

**Sample Screenshots:**



A screenshot of a graph

Description automatically generated

**Inference:** The linear kernel of the SVM model performs best with a testing accuracy of 91% followed by rbf, polynomial and sigmoid respectively.

1. Develop a python program to recognize the digits using Support Vector Machine (SVM) Model. Visualize the features from the dataset and interpret the results obtained by the model using Matplotlib library.

**Code:**

import numpy as np

import matplotlib.pyplot as plt

from sklearn.datasets import fetch\_openml

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

from sklearn.preprocessing import StandardScaler

from sklearn.svm import SVC

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

mnist = fetch\_openml('mnist\_784', version=1)

X, y = mnist.data, mnist.target

X = X / 255.0

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

# Initialize the SVM classifier

svm\_clf = SVC(kernel='linear', gamma='scale', C=1.0, random\_state=42)

# Train the SVM model

svm\_clf.fit(X\_train, y\_train)

# Evaluate the trained model

y\_pred = svm\_clf.predict(X\_test)

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

print("Accuracy:", accuracy)

X = np.array(X)

plt.figure(figsize=(10, 4))

for i in range(10):

plt.subplot(2, 5, i + 1)

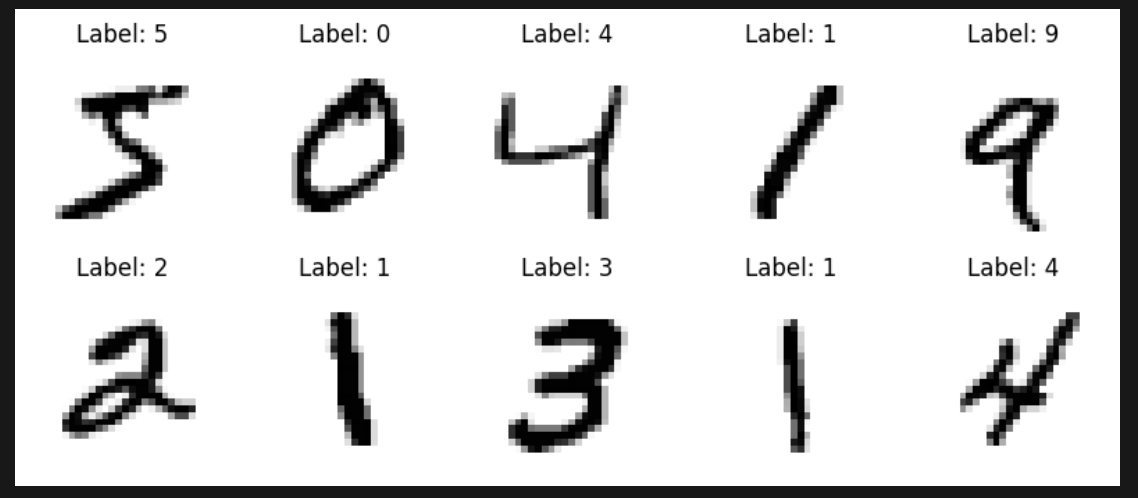
plt.imshow(X[i].reshape(28, 28), cmap='binary', interpolation='nearest')

plt.title('Label: ' + str(y[i]))

plt.axis('off')

plt.show()

**Sample Screenshot:**



1. Develop a python program to classify Emails as Spam or Ham using Naïve Bayes Model. Visualize the features from the dataset and interpret the results obtained by the model using Matplotlib library.

**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

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

from sklearn.metrics import accuracy\_score, confusion\_matrix

import seaborn as sns

from sklearn.naive\_bayes import MultinomialNB

df = pd.read\_csv('email.csv')

# Visualize the features from the dataset

df.iloc[::,-1].value\_counts().plot(kind='bar')

# Preprocessing

X = df.iloc[::,:-1]

y = df.iloc[::,-1]

labels = ['Spam','Ham']

explode=np.zeros(len(labels))

explode[-1]=0.1

plt.title("Ham vs Spam")

plt.pie(df.iloc[:,-1].value\_counts(), labels=labels, counterclock=False, shadow=True,

explode=explode, autopct='%1.1f%%', radius=1, startangle=0)

plt.show()

corr=df.corr()

plt.figure(figsize=(12, 10))

sns.heatmap(corr,

xticklabels=corr.columns,

yticklabels=corr.columns,

cmap=sns.diverging\_palette(250, 10, as\_cmap=True),

square=True,

linewidths=.1)

plt.show()

# Splitting the dataset

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

model\_nb = MultinomialNB()

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

def pred\_plot(model):

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

print('Training Accuracy:', model.score(X\_train, y\_train))

print('Accuracy Score:', accuracy\_score(y\_test, y\_pred))

confusion = confusion\_matrix(y\_test, y\_pred)

# Visualising the results

plt.matshow(confusion)

plt.colorbar()

plt.xlabel('Actual')

plt.ylabel('Predicted')

plt.show()

pred\_plot(model\_nb)

**Sample Screenshots:**

