Assignment 2
ICS322 – Machine Learning

Name: Arya Suneesh Roll No. : 2021BCS0005

Project Steps:
1.Data Collection: Download the MNIST dataset which contains 70,000 images of handwritten digits. The dataset is available in different formats like CSV, JSON, etc.
Choose a format that is compatible with the programming language you'll be using. For example, if you'll be using Python, you can download the dataset in CSV format.
2. Data Preprocessing: Once you have the dataset, preprocess the data by dividing it into training and testing sets. The training set will be used to train the model, and the testing set will be used to evaluate its performance.
3. Model Building: Next, build a Support Vector Machine (SVM) model using a suitable library such as scikit-learn in Python. Choose appropriate hyperparameters for the SVMmodel, such as the kernel function, regularization parameter, etc.
4. Model Training: Train the SVM model on the training dataset.
5. Model Evaluation: Evaluate the performance of the SVM model on the testing dataset by calculating metrics such as accuracy, precision, recall, and F1-score.

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image:
https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input
directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'): for filename in
        filenames:
                print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output
when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the
current session

/kaggle/input/mnist-csv/mnist_test.csv
/kaggle/input/mnist-csv/mnist_train.csv
```

Data preprocessing

```
train_data = pd.read_csv("/kaggle/input/mnist-csv/mnist_train.csv")  test_data =
pd.read_csv("/kaggle/input/mnist-csv/mnist_test.csv")

X_train = train_data.drop("label", axis=1)  y_train =
train_data["label"]

X_test = test_data.drop("label", axis=1)  y_test =
test_data["label"]
```

Build SVM Model

```
from sklearn.svm import SVC
```

Model Training

```
svm_model = SVC(kernel='rbf', C=1.0)

svm_model.fit(X_train, y_train)
```

```
SVC()
```

Evaluate model

```
from sklearn.metrics import accuracy_score, precision_score,  recall_score, f1_score

y_pred = svm_model.predict(X_test)

# Calculate evaluation metrics
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')
f1 = f1_score(y_test, y_pred, average='weighted')

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)  print("F1 Score:", f1)
```

```
Accuracy: 0.9792
Precision: 0.9792006017788697
Recall: 0.9792
F1 Score: 0.9791856837674859
```

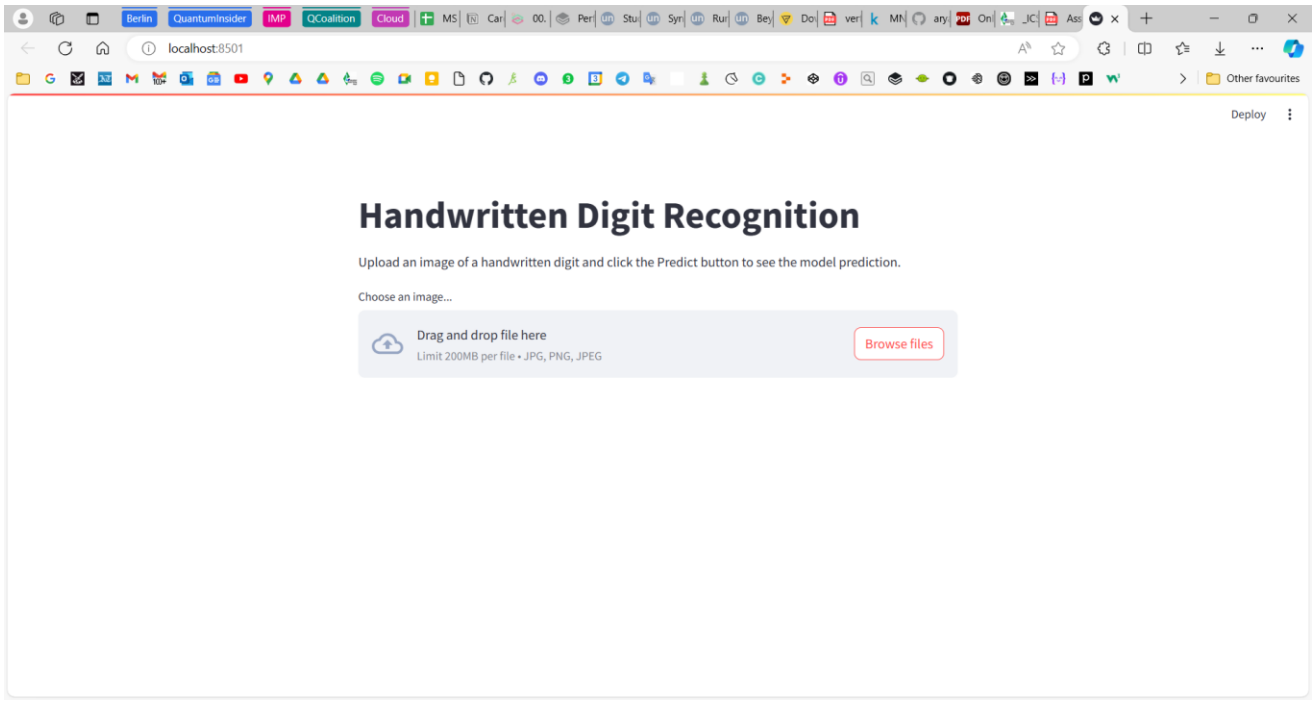
Export model

```
import joblib

# Save the model
joblib.dump(svm_model, 'svm_model.pkl')
```

```
['svm_model.pkl']
```

6. Model Deployment: Once the model is trained and tested, deploy it in a suitable environment such as a web application or a mobile application.
7. User Interface: Build a simple user interface that allows users to draw a digit on the screen, and then the SVM model predicts the digit based on the drawn image.
8. Testing: Finally, test the application thoroughly to ensure that it is working correctly and giving accurate predictions


localhost:8501
Other favourites
Deploy
Handwritten Digit Recognition
Upload an image of a handwritten digit and click the Predict button to see the model prediction.
Choose an image...
Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG
Browse files

Handwritten Digit Recognition

Upload an image of a handwritten digit and click the Predict button to see the model prediction.

Choose an image...

Drag and drop file here

Limit 200MB per file • JPG, PNG, JPEG

Browse files

hw_5.png 13.2KB

Uploaded Image

`st.cache` is deprecated. Please use one of Streamlit's new caching commands, `st.cache_data` or `st.cache_resource`. Based on this function's return value of type `int`, we recommend using `st.cache_data`.

More information in our docs.

Prediction: 5