# Machine Learning with Python

## Major Project

A project designed to identify digit classification using SVM algorithm from the MNIST dataset

Prepared by: Gurveer Singh Dhillon

Date: 20/7/21

## Problem Statement

Design a project from the MNIST dataset to identify digit classification using the SVM algorithm.

## Libraries and IDE used

The program was written and performed on Google Colab Notebook

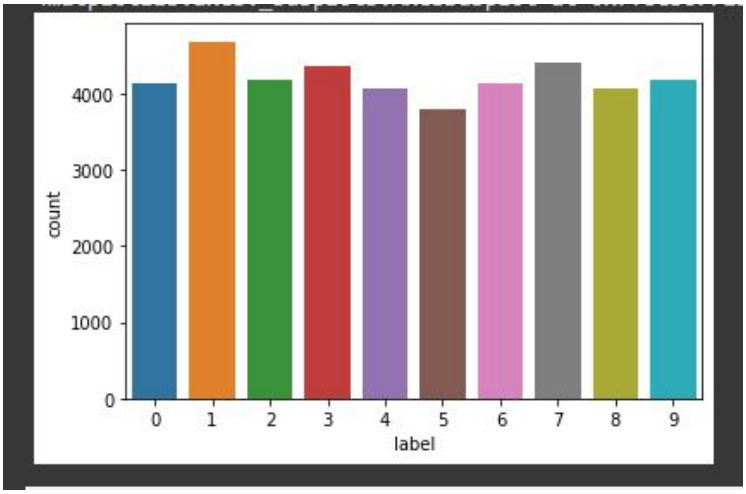
The Libraries used are as follows

- numpy
- pandas
- matplotlib.pyplot as plt, matplotlib.image as mpimg
- seaborn
- sklearn
- train_test_split from sklearn.model_selection
- StandardScaler from sklearn.preprocessing
- confusion_matrix, classification_report, accuracy_score from sklearn.metrics
- from sklearn.svm import SVC

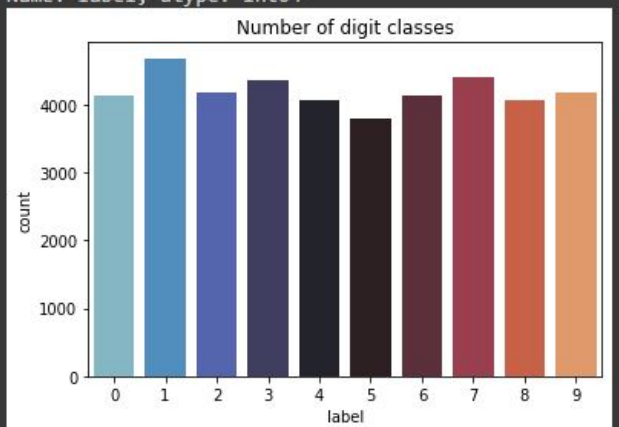
# Methodology

First of all we study the data given in the problem statement, after which we open our Google Colab Notebook to proceed with the problem and solve it.

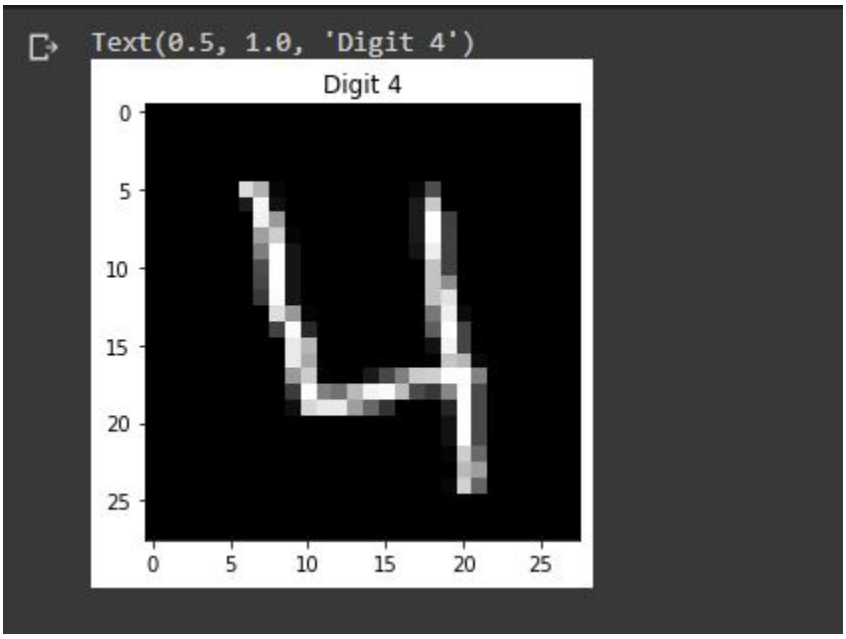
In our Notebook, we first import the aforementioned libraries which will be used in the program. Then we load our given dataset on our notebook. The file is then read and we print our Dataframe. We use .isnull().sum() function to check for null values in our dataset. We plot graphs to show the count of digits present in the dataset. The steps we followed are included in Exploratory Data Analysis.



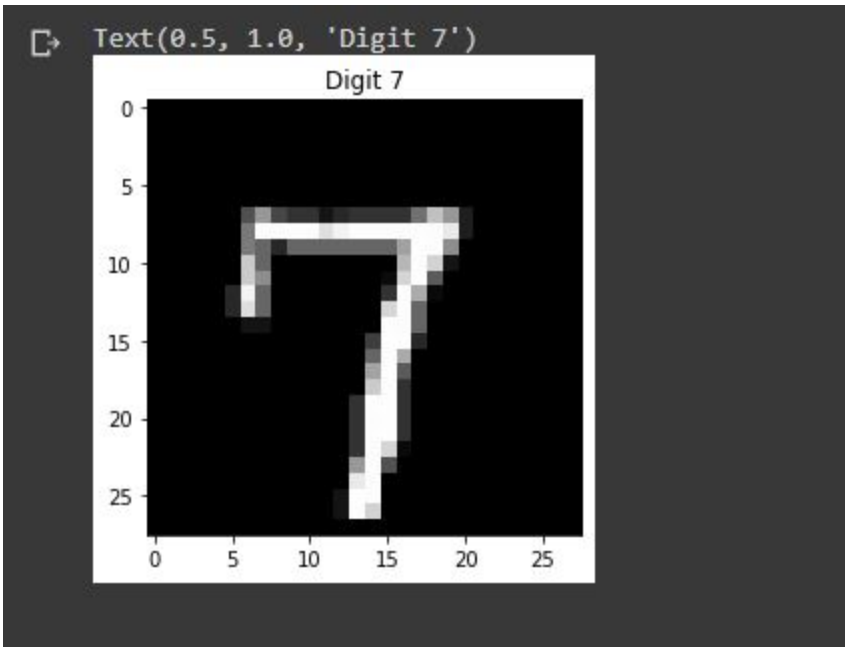

```
1    4684
7    4401
3    4351
9    4188
2    4177
6    4137
0    4132
4    4072
8    4063
5    3795
Name: label, dtype: int64
```



After plotting the graph we have plotted some samples as well as converted them into matrix. We have shown the digits ‘4’ and ‘7’.




**Digit ‘4’**




**Digit ‘7’**

Next, we move on to Preparation of our Data Models, we have used SVM algorithm in this project, hence we will be dealing with SVM Classifiers and use their different orientations for 'kernel' parameter like 'linear' and 'rbf'. We split the data into train and test for training and testing the data using train_test_split. We have used the train_size as 0.8 and random_state as 0. Then we have used StandardScaler() function to scale our x_train and x_test.

After preparing our data, we go towards Classification Models where we will use SVM algorithm to classify our model. Using SVM Classifier we implement our first model with kernel as linear and random_state as 0. We have created a classifier with the aforementioned parameters. Then we have used fit function to train our x_train and y_train. After this we predict x_test using predict function. Then we print the confusion matrix and accuracy score for our model. Then we start on with another model with kernel as rbf and random_state to be 0. We create another classifier with the given parameters. Then we repeat the procedure of using fit function to train the dataset and predict function to predict x_test. Then we print the confusion matrix and accuracy score for our second model.

# Model Codes

The Confusion matrix for SVC classifier with kernel as linear is

```
array([[103,   0,   0,   0,   0,   0,   0,   0,   0,   0],
       [  0,  96,   1,   1,   0,   0,   1,   1,   0,   0],
       [  1,   1,  92,   2,   3,   0,   3,   1,   1,   0],
       [  0,   1,   1,  90,   0,   1,   1,   1,   4,   1],
       [  0,   0,   1,   0, 100,   0,   1,   1,   0,   4],
       [  1,   2,   0,   3,   0,  80,   4,   0,   4,   0],
       [  2,   1,   1,   0,   0,   1,  88,   0,   0,   0],
       [  0,   5,   3,   0,   1,   1,   0, 100,   2,   4],
       [  1,   7,   2,   2,   0,   3,   1,   0,  69,   0],
       [  1,   0,   2,   2,   4,   1,   0,   2,   1,  85]])
```

The Confusion matrix for SVC classifier with kernel as rbf is

```
array([[100,   0,   1,   0,   0,   0,   1,   0,   1,   0],
       [  0,  96,   0,   1,   0,   0,   2,   1,   0,   0],
       [  0,   0,  95,   0,   3,   0,   0,   1,   4,   1],
       [  0,   0,   5,  89,   0,   0,   0,   1,   4,   1],
       [  0,   0,   6,   0,  99,   0,   0,   0,   0,   2],
       [  0,   2,   2,   2,   0,  83,   5,   0,   0,   0],
       [  1,   0,   1,   0,   0,   0,  90,   0,   1,   0],
       [  0,   6,   3,   0,   2,   0,   0, 101,   0,   4],
       [  0,   1,   3,   1,   0,   4,   0,   0,  76,   0],
       [  3,   0,   2,   1,   4,   1,   0,   4,   1,  82]])
```

# Conclusion

I hereby conclude my findings that SVC Classifier with kernel as rbf gives us accuracy of 91.1% which is higher than SVC Classifier with kernel as linear which had accuracy of 90.3%.