**ML-MAJOR-MAY**

**Support Vector Machine: Digit Classification with Python;**

## The data-set is based on Gray-scale images of handwritten digits and, each image is 28 pixel in height and 28 pixel in width. Each pixel has a number associated with it, where 0 represents a dark pixel and, 255 represents a white pixel. Both the train and test data-set have 785 columns where, ‘label’ column represents the handwritten digit and remaining 784 columns represent the (28, 28) pixel values.

## I have used 42000 samples and 8400 samples from the training and test data-sets just to reduce the time of computation and it is recommended to use the full set to obtain a better score and avoid selection bias.

First we need to import the required libraries :

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

We can **check whether the training data-set is biased towards certain numbers or not by printing out the value\_counts()** and/or from the distribution plot of labels.

df['label'].value\_counts()

we can plot the graph for checking how count is differ from digit to Digit.

sns.countplot(df['label'],palette = 'icefire')

We have spitted the X and Y values from the given Data AS unique labels and their counts.

x = df.iloc[:,1:]

y = df.iloc[:, 0]

Model Building

1. Linear model:

As the pixel values vary in the range 0–255, it is time to use some standardization and I have used StandardScaler which standardize features by removing mean and scaling it to unit variance.

from sklearn.preprocessing import StandardScaler

I have separated training and test data with 20% samples reserved for test data.

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

train\_x,test\_x,train\_y,test\_y = train\_test\_split(x,y,test\_size=0.2)

After trying all the other models for gaining more accuracy and as mentioned in the project we Utilized the SVM algorithm.

#import SVC model from svm algorithm

from sklearn.svm import SVC

Now we are ready to test the model and find the best-fit parameters.

Finally we got the machine with accuracy of 92%.By using SVM algorithm.

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

accuracy\_score(test\_y, pred\_y)

0.9203571428571429

By,

Yerram Deekshith Kumar

ML-MAY-21