STA 545/EAS 506
STATISTICAL DATA MINING 1

Final Project

University at Buffalo The State University of New York

ANALYSIS AND MODELLING OF FMINST

Group 15

Satish Varma Bhupathiraju

Akshey Ram Murali

SriHarsha Teja Nallamala

Vaishnav Tammadwar

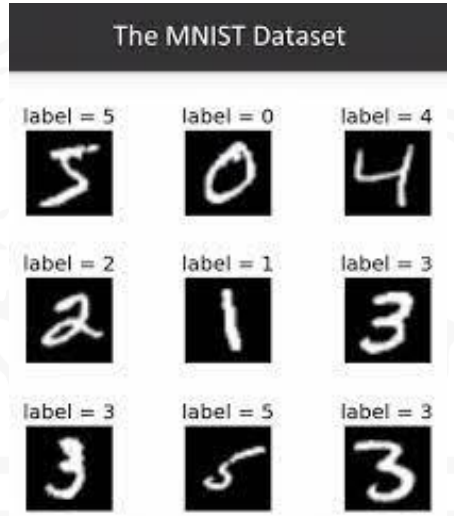
University at Buffalo The State University of New York

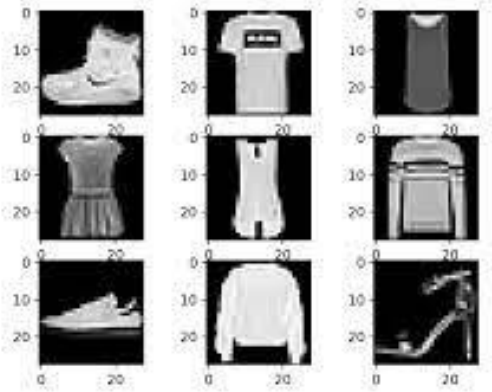
University at Buffalo The State University of New York

CONTENT

Introduction

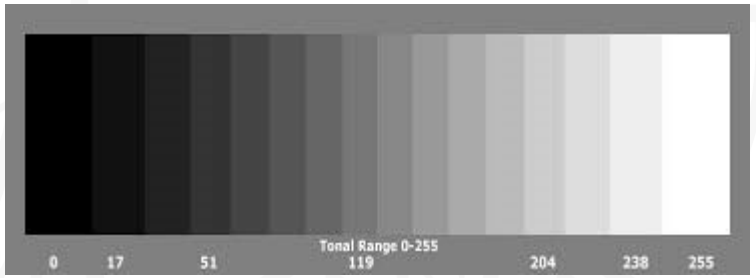
- MNIST data is popular data set for image classification machine learning algorithms.
- Initially MNIST Data set is compromising of 10 class handwritten digits later it is modified into Fashion MNIST Data.
- Every fashion product on Za-lando has a set of pictures, demonstrating different aspects of the product.
- They have used the front look thumbnail images of 70,000 unique products to build Fashion-MNIST dataset. Those products come from different gender groups: men, women, kids and neutral.





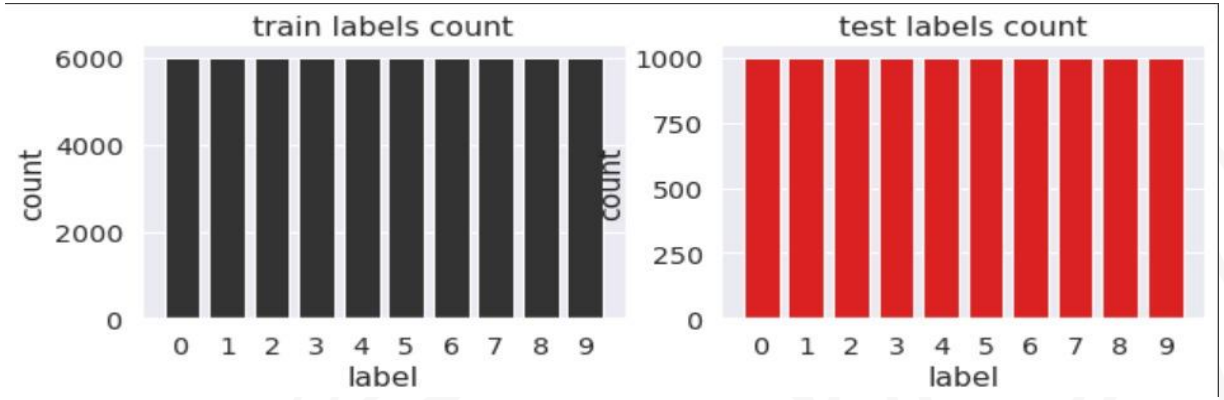
Data Understanding

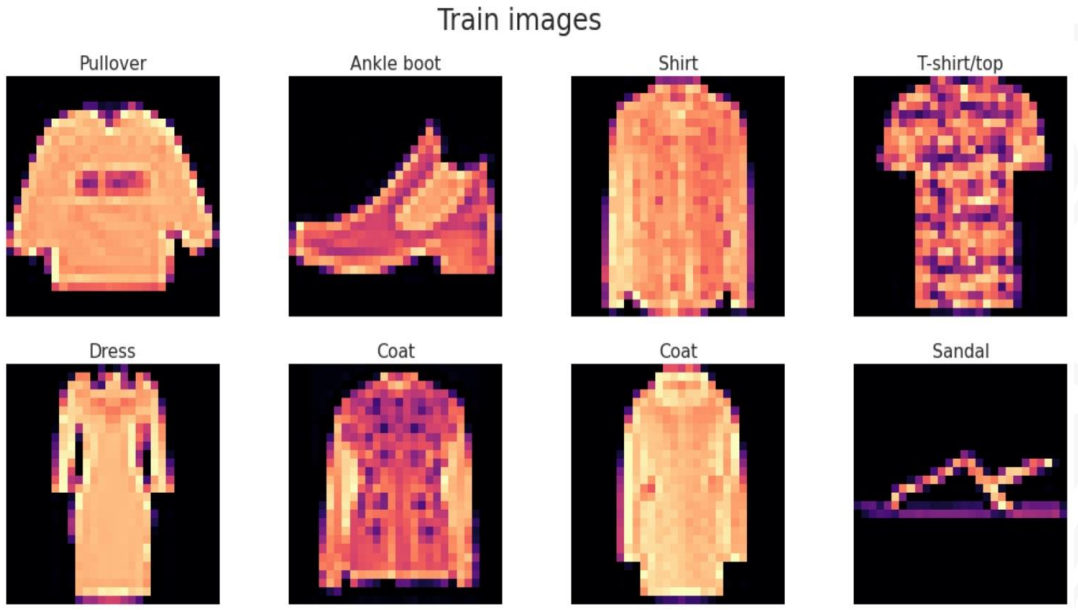
- Each image is 28 pixels in width and 28 pixels in height and a total of 784 pixels in total.
- Each pixel is associated with a single pixel-value, indicating the lightness or darkness of that pixel
- This pixel-value is an integer between 0 and 255.
- The training and test data sets have 785 columns.
- First column contains the class label (T shirt, Trouser..)
- The Training data set has 60000 rows
- The testing data set has 10000 rows



Exploratory Data Analysis

- The data set is split into train data and test data in the ratio 0.85% and 0.15% respectively.
- From the graph it is evident that all labels are equally spread in the data set both test and train.
- Each Train and test sample is assigned to one of the following labels.
- 0 – T-shirt; 1 – Trouser etc.,
- After Reshaping and setting the Cmap to magma we would get the corresponding image to labels.





Data Cleaning and Data Pre-Processing

- The Data set has no Null values.
- The Data and the labels are separated by removing the label column from the data set.
- Considering Fashion Mnist to be an image dataset, data preprocessing is the vital step for the models to yield the expected accuracy.
- The Pixel Values are often stored as Integer Numbers in the range 0 to 255, the range that a single 8-bit byte can offer.
- Scaling all the contents in the data sets [0,1] to Optimize Algorithms to work much faster. Here, we achieve Zero Mean and Unit Variance.
- Reduction is done using PCA in order to compress the corelated and collinear data so that it can be more effectively modelled.
- Sklearn decomposition function is used to reduce the dimensionality.

Predictive Models

To predict the label of the given image data by applying machine learning models mentioned below.

- Logistic Regression
- Support Vector Machine
- Convolution Neural Network
- Decision Tree
- Random Forest
- KNN
- Naïve Bayes

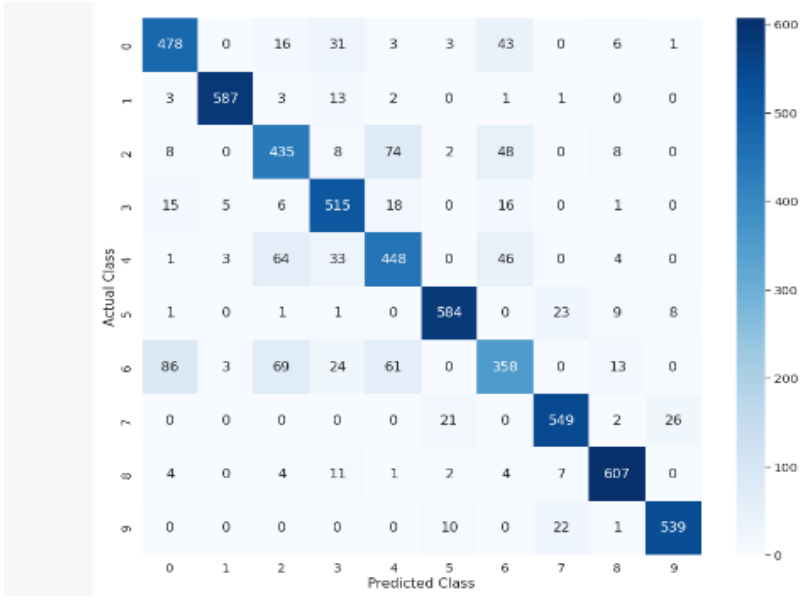
Logistic Regression

- Logistic regression is used to classify the label 0 to 9 to the corresponding pixels.
- The predefined function from sklearn is used for logistic regression and “libliner” solver is used. Since it works well for mid level data sets.
- The maximum iteration is set for 200.

Test Accuracy : 85.24%

----------THE CLASSFICATION REPORT FOR LOGISTIC REGRESSION IS----------

	precision	recall	f1-score	support
0	0.80	0.82	0.81	581
1	0.98	0.96	0.97	610
2	0.73	0.75	0.74	583
3	0.81	0.89	0.85	576
4	0.74	0.75	0.74	599
5	0.94	0.93	0.94	627
6	0.69	0.58	0.63	614
7	0.91	0.92	0.92	598
8	0.93	0.95	0.94	640
9	0.94	0.94	0.94	572
accuracy			0.85	6000
macro avg	0.85	0.85	0.85	6000
weighted avg	0.85	0.85	0.85	6000



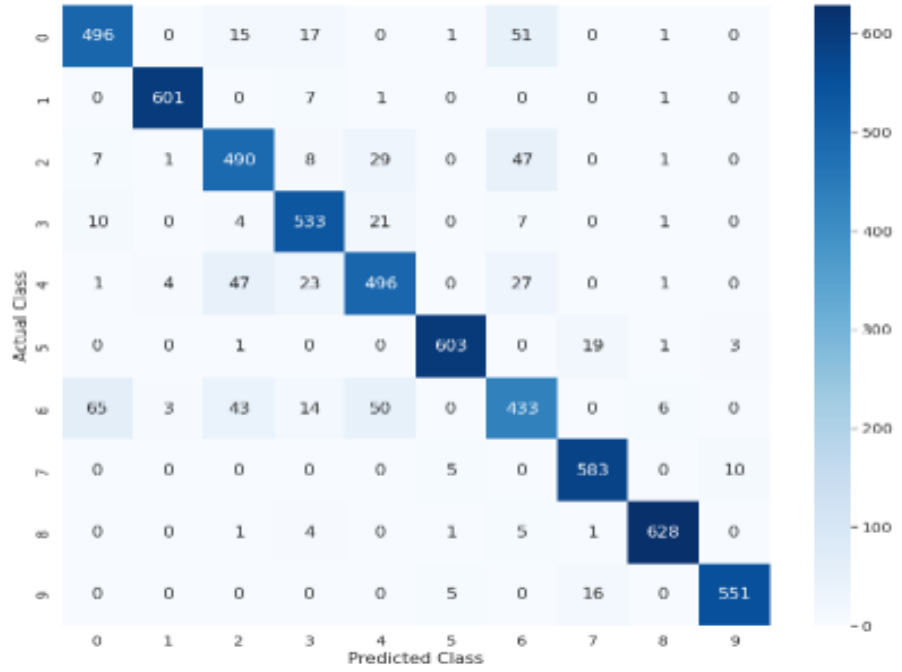
Support Vector Machine

- One common supervised learning technique, support vector machine is used here to classify the labels.
- SVC, The predefined function from sklearn is used.
- The default kernel type of rbf is used to precompute the kernel matrix to data matrix.
- Gamma is set to auto and uses 1/n where n is the number of features.

Test Accuracy : 90.05%

------THE CLASSFICATION REPORT FOR SUPPORT VECTOR CLASSIFIER IS------

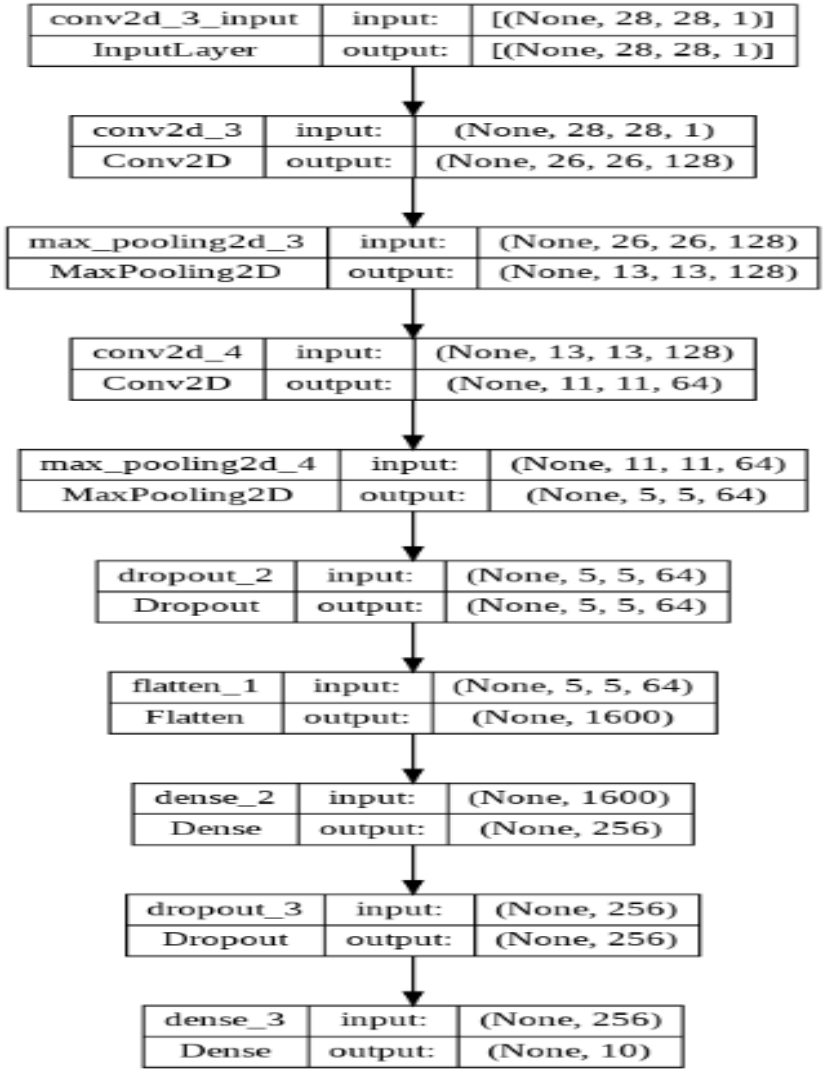
	precision	recall	f1-score	support
0	0.86	0.85	0.86	581
1	0.99	0.99	0.99	610
2	0.82	0.84	0.83	583
3	0.88	0.93	0.90	576
4	0.83	0.83	0.83	599
5	0.98	0.96	0.97	627
6	0.76	0.71	0.73	614
7	0.94	0.97	0.96	598
8	0.98	0.98	0.98	640
9	0.98	0.96	0.97	572
accuracy			0.90	6000
macro avg	0.90	0.90	0.90	6000
weighted avg	0.90	0.90	0.90	6000



Convolution Neural Network

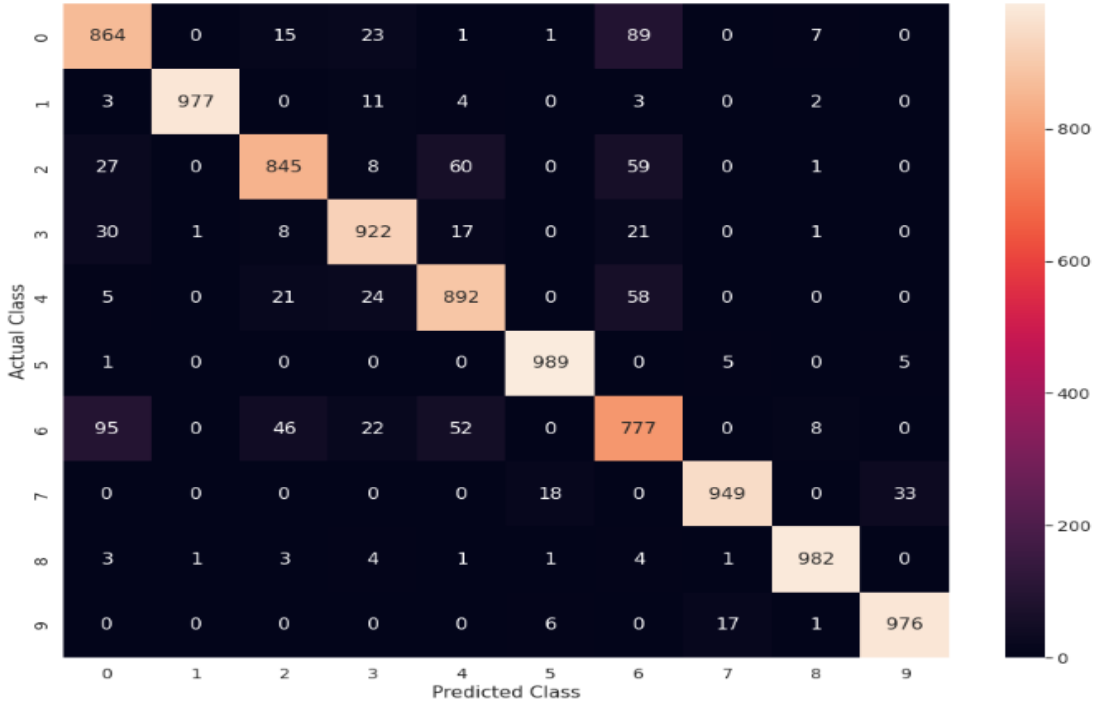
- CNN is a widely used for image classification.
- The Architecture of CNN is shown here.
- reLU is used as an Activation Function adam used as an optimizer and categorical class entropy is used as a metric to calculate the loss.
- Max-pooling, drop out value of 0.2 and flatten layer is used.

CNN Model

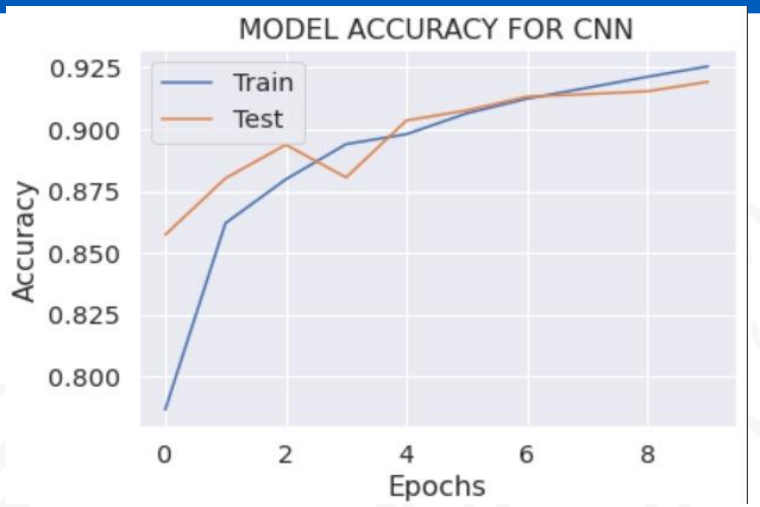


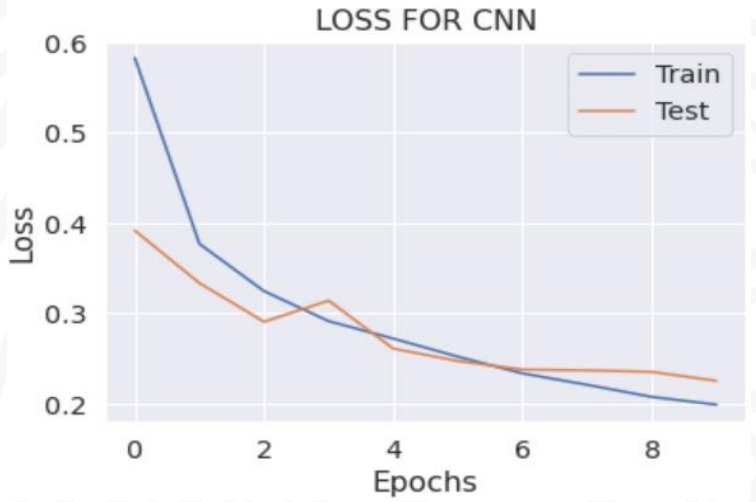
Convolution Neural Network

Confusion Matrix



	0	1	2	3	4	5	6	7	8	9
0	864	0	15	23	1	1	89	0	7	0
1	3	977	0	11	4	0	3	0	2	0
2	27	0	845	8	60	0	59	0	1	0
3	30	1	8	922	17	0	21	0	1	0
4	5	0	21	24	892	0	58	0	0	0
5	1	0	0	0	0	989	0	5	0	5
6	95	0	46	22	52	0	777	0	8	0
7	0	0	0	0	0	18	0	949	0	33
8	3	1	3	4	1	1	4	1	982	0
9	0	0	0	0	0	6	0	17	1	976






```
test_loss, test_acc = cnn_model.evaluate(X_test, y_test)
test_acc= test_acc*100
print("The test accuracy :",test_acc)
```

```
313/313 [==============================] - 1s 3ms/step -
The test accuracy : 91.8500006198883
```

Test Accuracy : 91.85%

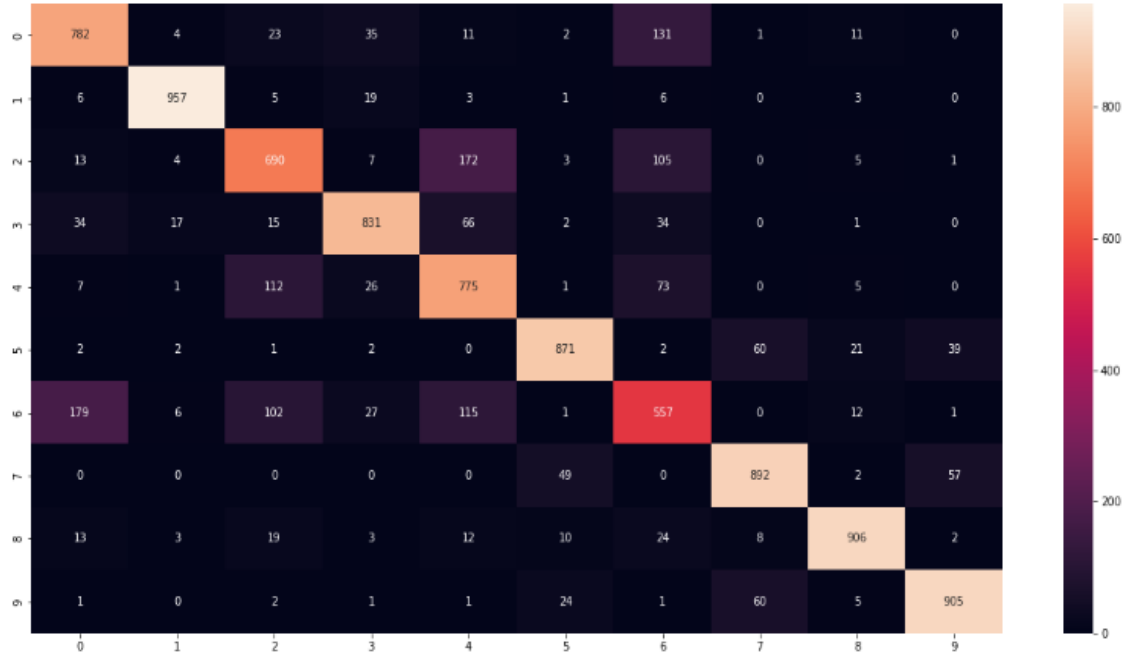
Decision Tree

- Represents upside down tree with its root at the top.
- Data is split from the root and each non-leaf node contains a condition.
- The final leaf node contains a prediction.
- we partition the data into resulting regions and repeat the same process of splitting on each region.
- We use the predefined function from sklearn (Decision Tree Classifier) to built this model.

Test Accuracy : 82.07%

----------THE CLASSFICATION REPORT FOR DECISION TREE IS----------

	precision	recall	f1-score	support
0	0.75	0.78	0.77	1000
1	0.96	0.96	0.96	1000
2	0.71	0.69	0.70	1000
3	0.87	0.83	0.85	1000
4	0.67	0.78	0.72	1000
5	0.90	0.87	0.89	1000
6	0.60	0.56	0.58	1000
7	0.87	0.89	0.88	1000
8	0.93	0.91	0.92	1000
9	0.90	0.91	0.90	1000
accuracy			0.82	10000
macro avg	0.82	0.82	0.82	10000
weighted avg	0.82	0.82	0.82	10000



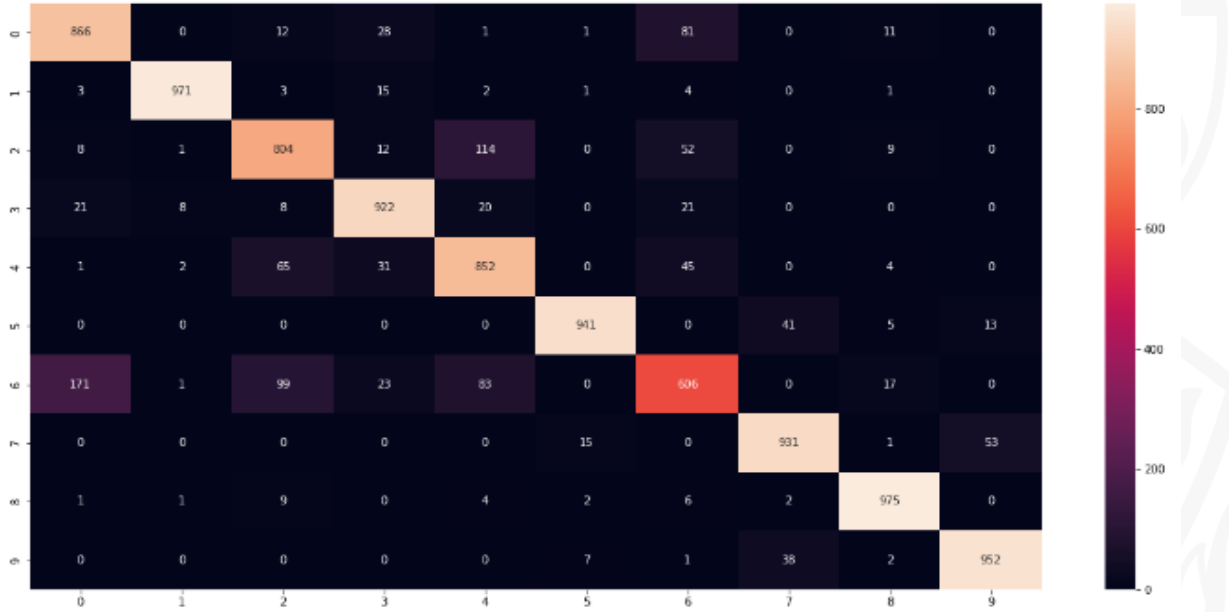
RANDOM FOREST

- Consists of a large number of individual decision trees.
- Each individual tree in this splits out a class prediction and best is considered.
- The predefined function from sklearn is used this model(Random Forest Classifier).
- Default function(gini) is used to measure the quality of a split.

Test Accuracy : 88.03%

----------THE CLASSFICATION REPORT FOR RANDOM FOREST IS----------

	precision	recall	f1-score	support
0	0.81	0.87	0.84	1000
1	0.99	0.97	0.98	1000
2	0.80	0.80	0.80	1000
3	0.89	0.92	0.91	1000
4	0.79	0.85	0.82	1000
5	0.97	0.94	0.96	1000
6	0.74	0.61	0.67	1000
7	0.92	0.93	0.93	1000
8	0.95	0.97	0.96	1000
9	0.94	0.95	0.94	1000
accuracy			0.88	10000
macro avg	0.88	0.88	0.88	10000
weighted avg	0.88	0.88	0.88	10000

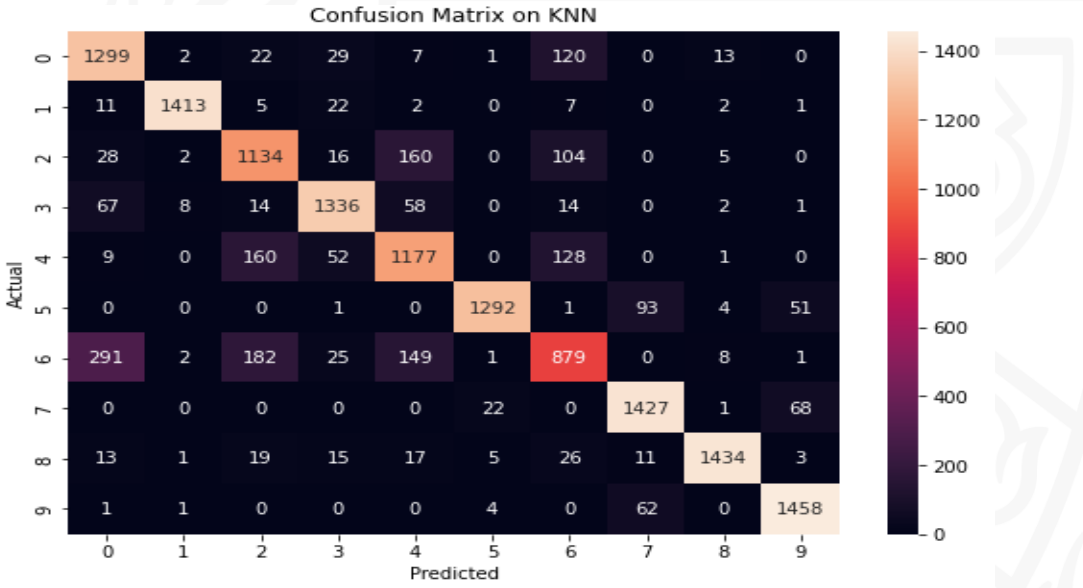


KNN

- K-Nearest neighbor algorithm is a non-parametric classification algorithm.
- Basic Idea, Given a test image(vector) find k images(vector) from train that are close to test image.
- Finding K closest vectors in Train set.
- Optimizing Parameter K.
- Classification using Euclidean Distance.
- The predefined function from sklearn is used this model(K-Neighbors Classifier)

Test Accuracy : 86.34%

	precision	recall	f1-score	support
0	0.76	0.87	0.81	1493
1	0.99	0.97	0.98	1463
2	0.74	0.78	0.76	1449
3	0.89	0.89	0.89	1500
4	0.75	0.77	0.76	1527
5	0.98	0.90	0.93	1442
6	0.69	0.57	0.62	1538
7	0.90	0.94	0.92	1518
8	0.98	0.93	0.95	1544
9	0.92	0.96	0.94	1526
accuracy			0.86	15000
macro avg	0.86	0.86	0.86	15000
weighted avg	0.86	0.86	0.86	15000

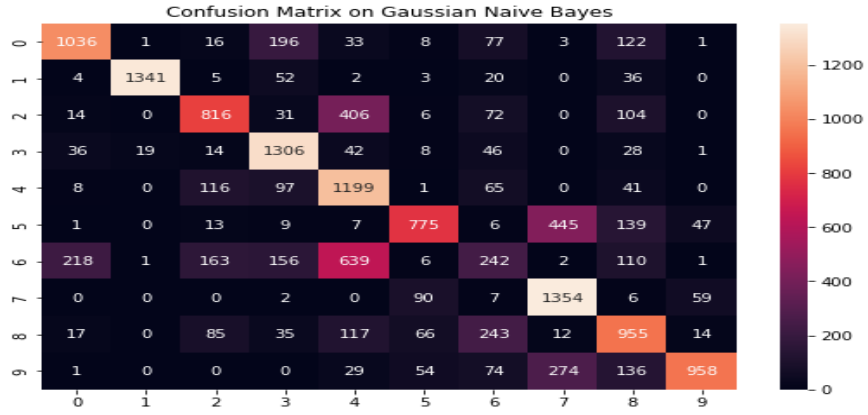


Naïve Bayes

- Naive-Bayes classifier uses Bayes theorem.
- Calculates the probability for membership of a data point to each class.
- Assumes that probability of each pixel is a gaussian distribution and probability of each digit is equal.
- The predefined function from sklearn is used this model(Gaussian NB)

Test Accuracy : 67.22%

	precision	recall	f1-score	support
0	0.78	0.69	0.73	1493
1	0.98	0.92	0.95	1463
2	0.66	0.56	0.61	1449
3	0.69	0.87	0.77	1500
4	0.48	0.79	0.60	1527
5	0.76	0.54	0.63	1442
6	0.28	0.16	0.20	1538
7	0.65	0.89	0.75	1518
8	0.57	0.62	0.59	1544
9	0.89	0.63	0.73	1526
accuracy			0.67	15000
macro avg	0.68	0.67	0.66	15000
weighted avg	0.67	0.67	0.66	15000

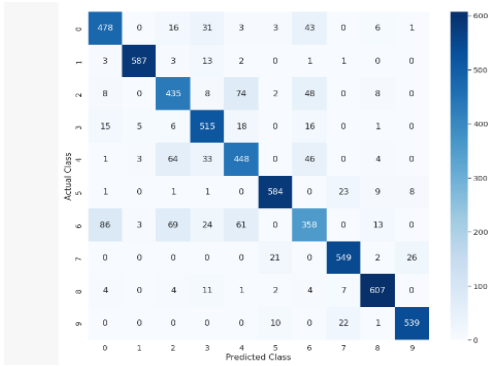


Results

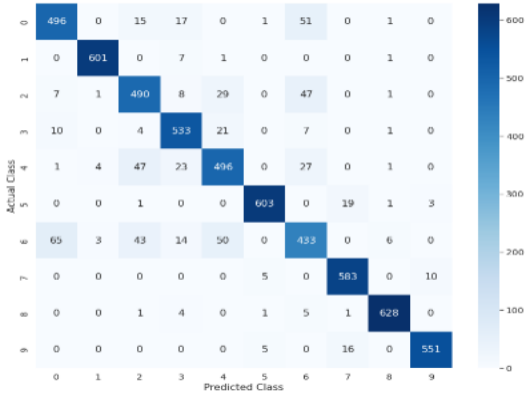
Accuracy for Fmnist Test data set.

Model	Accuracy
Logistic Regression	85.24%
KNN	86.34%
Naïve Bayes	67.22%
Decision Tree	82.07%
Random Forest	88.03%
SVM	90.05%
CNN	91.85%

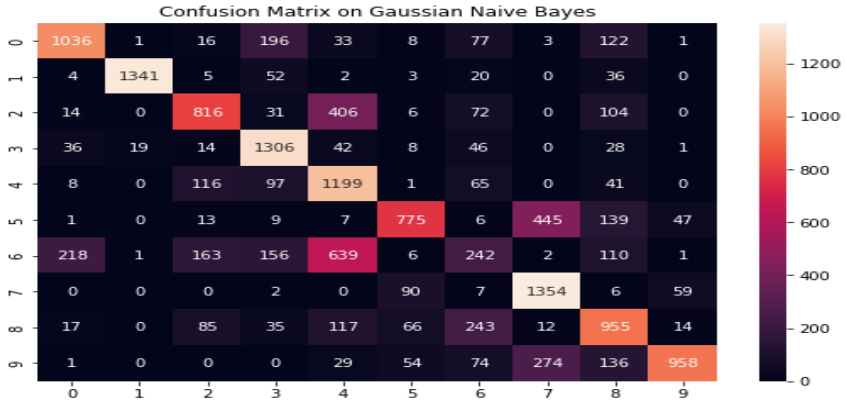
Logistic Regression



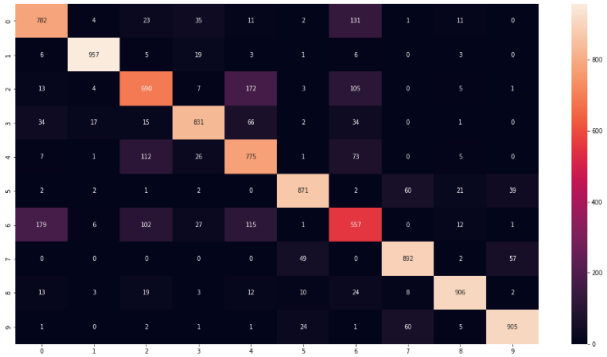
SVM



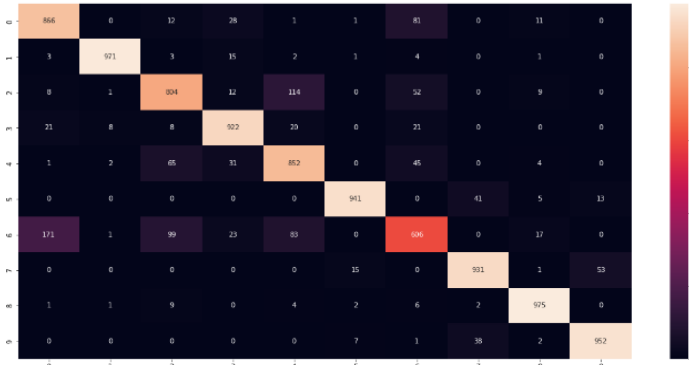
Naïve Bayes



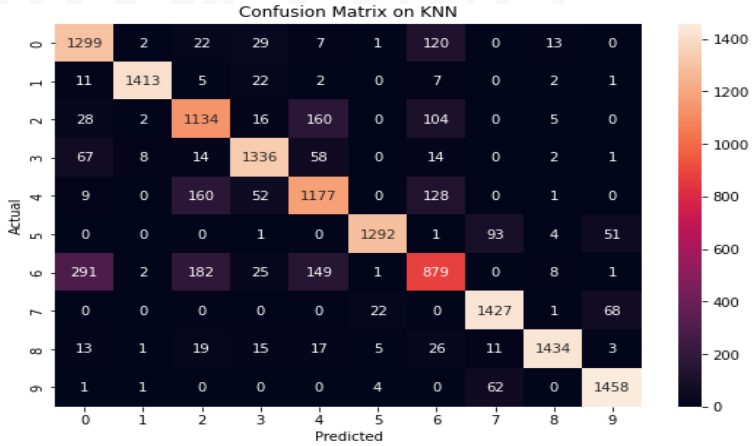
Decision Tree



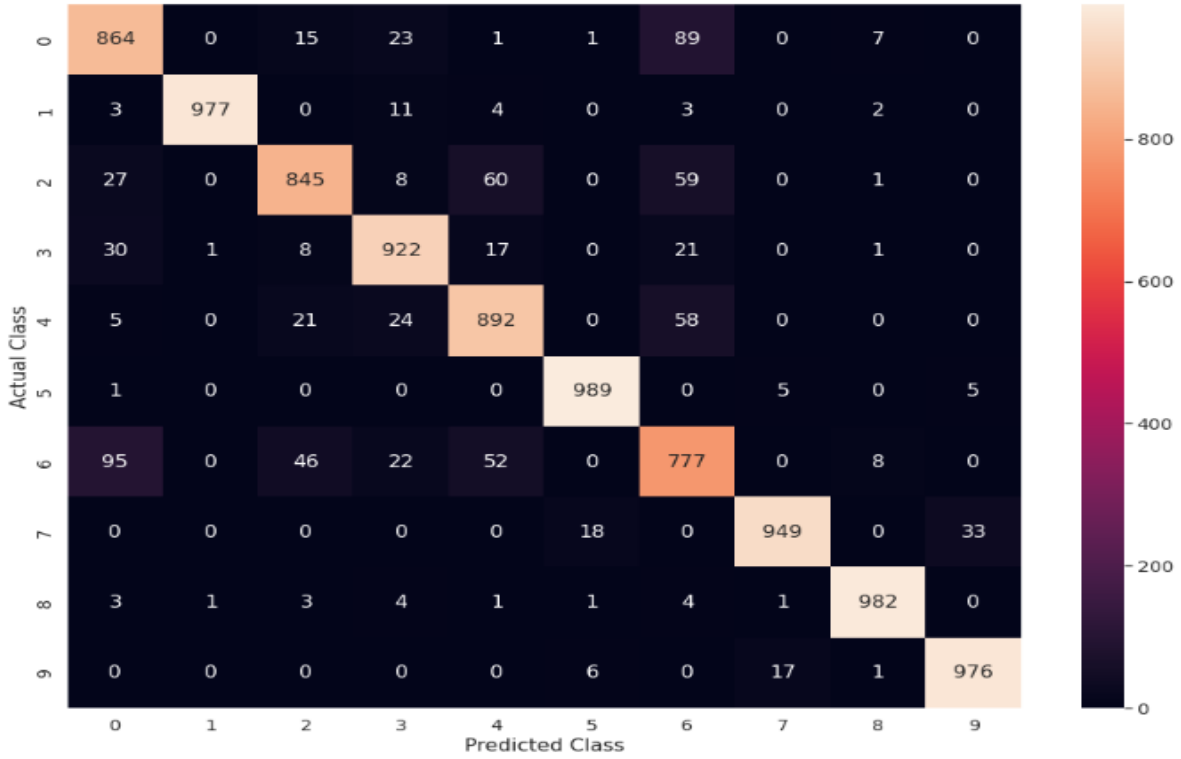
Random Forest



KNN



CNN



	0	1	2	3	4	5	6	7	8	9
0	864	0	15	23	1	1	89	0	7	0
1	3	977	0	11	4	0	3	0	2	0
2	27	0	845	8	60	0	59	0	1	0
3	30	1	8	922	17	0	21	0	1	0
4	5	0	21	24	892	0	58	0	0	0
5	1	0	0	0	0	989	0	5	0	5
6	95	0	46	22	52	0	777	0	8	0
7	0	0	0	0	0	18	0	949	0	33
8	3	1	3	4	1	1	4	1	982	0
9	0	0	0	0	0	6	0	17	1	976

- Based on the comparison of the results of the various models on the Train Data and Test Data, we can evidently say CNN is the well generalized model by **91.85%.**


THANK YOU