ML ASSIGNMENT 2 REPORT

HARISH CHANDRA THUWAL
2017MCS2074

1. NAIVE BAYES

Default Pre-processing

For every sample review:

- Strip. Remove all whitespace from beginning and end of the review.
- Create two sets of words from the line
  - Set1: Replace every non-alphanumeric character by whitespace. Split on whitespace to get set of words.

    E.g. word1,word2 => word1 word2

  - Set2: Replace every non-alphanumeric character by empty string. Split on whitespace to get set of words.

    E.g. word1,word2 -> word1word2

  Return the union of set1 and set2 as bag of words for this review.

a) Un-stemmed data (no stemming or stop word removal)

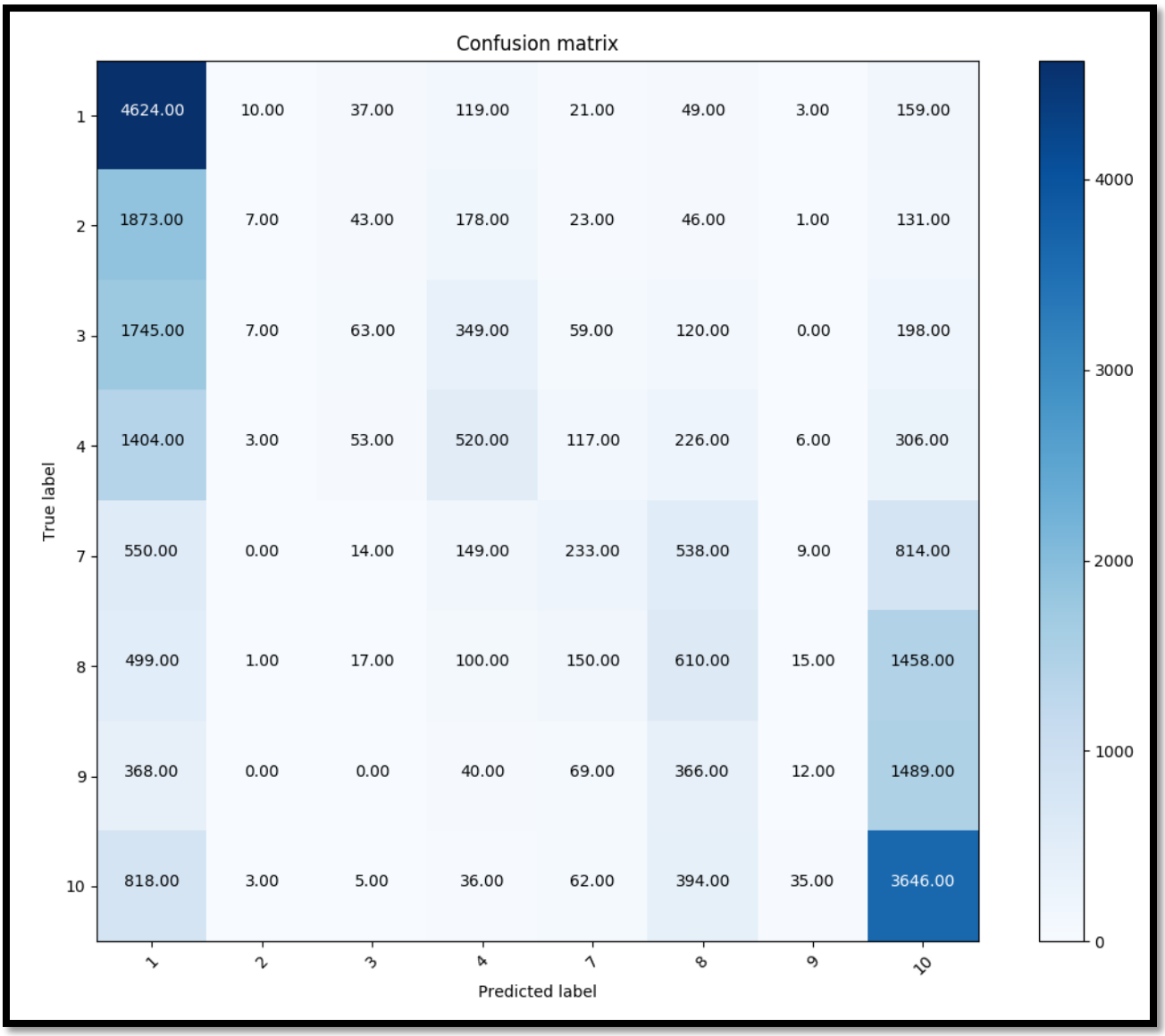
Dataset	Accuracy
Train Set	67.044
Test Set	38.856

b) Un-stemmed data

Algorithm	Accuracy on Test Set
Random Prediction	12.748
Majority Prediction	20.088

The Naive Bayes model have an accuracy almost twice of the Majority Prediction and more than three times the accuracy of the Random Prediction

c) Confusion Matrix



- **1** is the category with the highest diagonal entry of **4624** followed by **10** which has the value of **3646.** Diagonal Entry corresponds to the True Positive (number of samples correctly classified).

- Category 2 and 9 have the least number of correct classification.
- The model seems to identify the two extremes that is 1 and 10 pretty accurately while it fails drastically for others.

- Precision and Recall based on confusion matrix

Class	Precision	Recall	F Score
1	0.38919	0.9207	0.54712
2	0.22580	0.00304	0.00600
3	0.271551	0.02479	0.04543
4	0.34875	0.19734	0.2520
7	0.31743	0.10099	0.1532
8	0.25968	0.2140	0.2346
9	0.14148	0.005	0.00989
10	0.44457	0.729345	0.5524

As can be seen from the above table the Recall for category 1 which had the maximum diagonal entry is about 92%. This indicates that most of the samples belonging to the category 1 were correctly classified.

Category 10, had 72% Recall and the highest precision of 44.4%.

Category 2 and 9 had a very low recall of 0.3 and 0.5 percent. That is almost none of the samples of this category were correctly classified.

d) Stemmed data (Stemming + Stopwords Removal)

Dataset	Accuracy
Test Set	39.3960

After Stemming both the training and test data, accuracy over test set improved slightly from **38.860 to 39.400**. Also, the time required for training and testing almost halved. This is because stemming reduces the size of the vocabulary.

Vocabulary size reduces after stemming because stopwords are removed and words are reduced to their root form. This reduces the number of unique words.

e) Feature Engineering

- Deal with **negations** (not, never, neither etc...) - **F1**
  → A list of words that have inverting affect.
  → These words tend to inverse the sense/emotion of the upcoming words.
  → So, the effect of the next two words is inverted i.e subtracted instead of adding to the total probability of the doc belonging to a class.

Data Type	Naïve Bayes (Test Accuracy)	Naïve Bayes + F1 (Test Accuracy)
Un-stemmed	38.856	39.04
Stemmed	39.396	39.336

Adding Feature F1 improved the accuracy for un-stemmed data, while the accuracy in case of stemmed data remained more or less the same.

- F1 + **More Importance of First three words**
  → People tend to express more with the starting words of the movie review.
  → For e.g. Excellent. Best movie. It had ....
  → For e.g. Worst movie ever. It was as bad as ....
  → So, give more weight to the first three words of the review.
  → That is counting first three words twice while predicting
  → This increased the test set accuracy to 39.652

Data Type	Naïve Bayes (Test Accuracy)	Naïve Bayes + F1 (Test Accuracy)	F1 + F2 (Test Accuracy)
Un-stemmed	38.856	39.04	39.012
Stemmed	39.396	39.336	39.652

2. SVM

a) Pegasos Algorithm using mini batch gradient Descent

$$w \leftarrow w - \eta * Gradient_w$$
$$b \leftarrow w - \eta * Gradient_b$$

$$Gradient_w = c * \sum_{i=1}^{B} \delta(t^i) * y^i * x^i$$

$$Gradient_b = c * \sum_{i=1}^{B} \delta(t^i) * x^i$$

$$t^i = y^i (w^T x^i + b)$$

$$\delta(t^i) = 0 \text{ if } t^i > 1 \text{ else } -1$$

- Stopping Criteria -> maximum change in w, b < threshold(10e-4)
- C = 1
- Eeta = 1 / iteration number.
- B = Batch size = 100

b) One-vs-One

Data Set	Accuracy
Train	96.2350
Test	92.4900

c) LIBSVM (Test Accuracies)

Kernel	Test Accuracy
Linear (C = 1.0)	**92.78**
Gaussian (C = 1.0, gamma = 0.05)	**97.23**

The accuracy obtained by the linear kernel of the libsvm module is almost same as the accuracy obtained by our implementation of the pegasos algorithm.

The linear kernel was mere 0.29 % more accurate than the pegasos algorithm.

d) Cross Validation for Gaussian Kernel

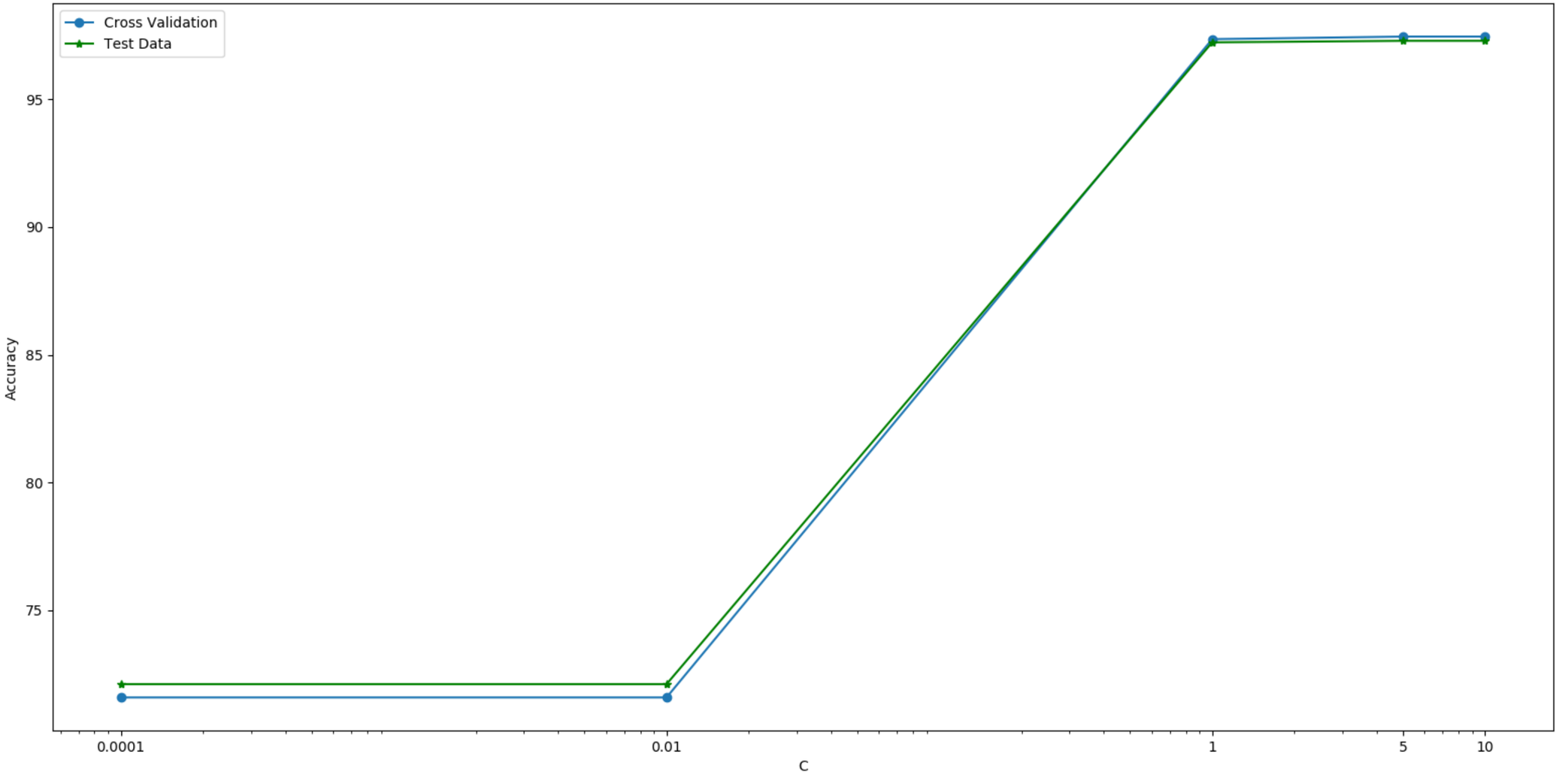
Gamma fixed to 0.05, C belongs to $\{10^{-5}, 10^{-3}, 1, 5, 10\}$

Accuracy	$C=10^{-5}$	$C=10^{-3}$	C=1.0	C=5.0	C=10.0
CV	71.59	71.59	97.355	97.455	97.455
Test Set	72.11	72.11	97.23	97.29	97.29

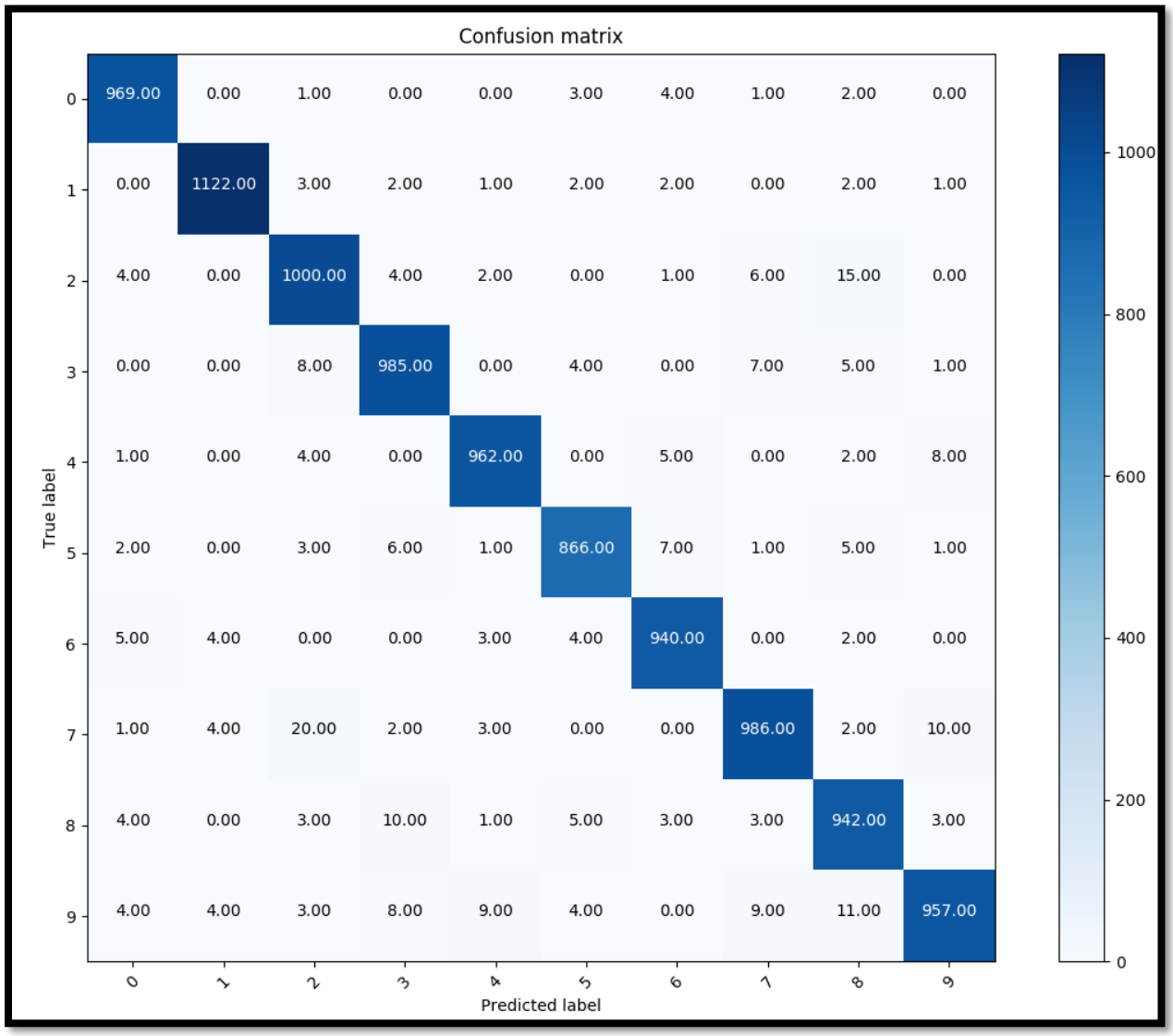
It can be observed from the table and the plot ahead that both the test accuracy and the cross validation follow a similar curve on changing the hyperparameters.

Both C=5.0 and 10.0 give the best cross validation accuracy. Same is the case for test accuracy.

Therefore, in the scenarios where we don't have a test set, we can trust the best model obtained by cross validation to perform best on the test set as well.


Cross Validation
Test Data
Accuracy
95
90
85
80
75
0.0001
0.01
1
5
10
C

e) Confusion Matrix (c=5)

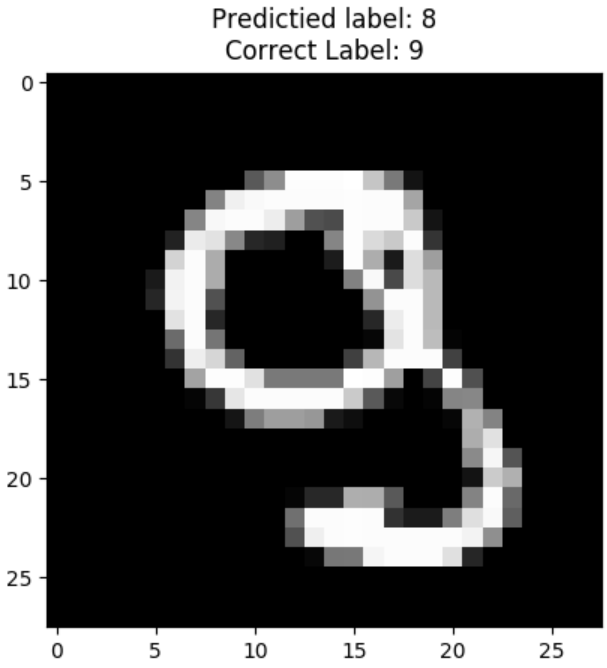


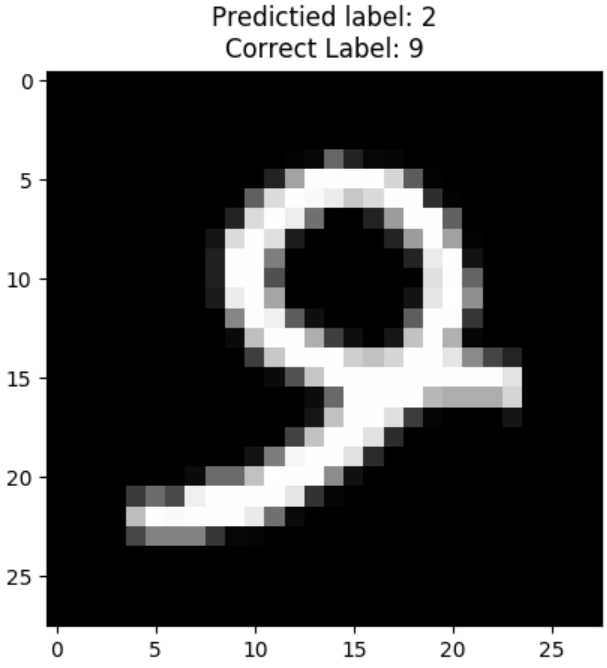
Class	Precision	Recall	F Score
0	0.978	0.988	0.9837
1	0.989	0.988	0.9889
2	0.956	0.968	0.9629
3	0.968	0.975	0.9718
4	0.979	0.979	0.9796
5	0.975	0.970	0.9730
6	0.977	0.981	0.9791
7	0.973	0.959	0.9661
8	0.953	0.967	0.9602
9	0.975	0.948	0.9618

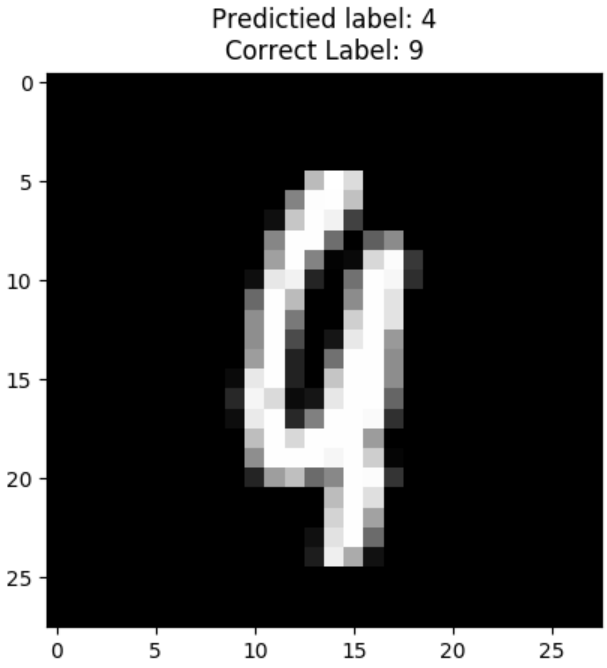
9 seems to be the most difficult class to classify it has the lowest recall.

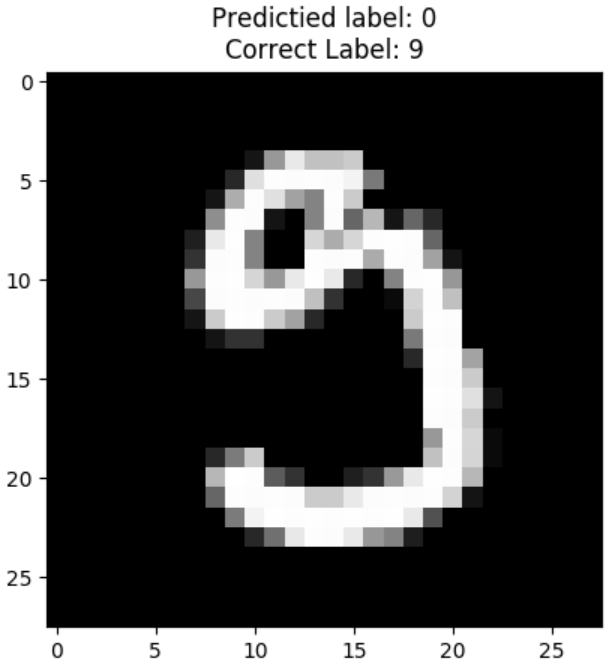
That is class 9, **has highest number of misclassified samples**.

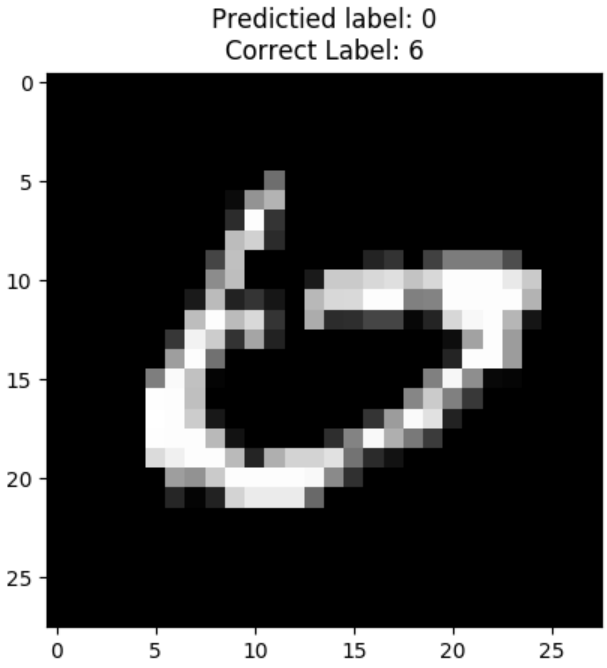
As can be seen from the upcoming visualization. The handwritten digit possesses some sort of relevance to the actually predicted class even though it's not the correct label. That is the model identified the common similarity and predicted the label. The predictions even though are wrong but are not delusional.

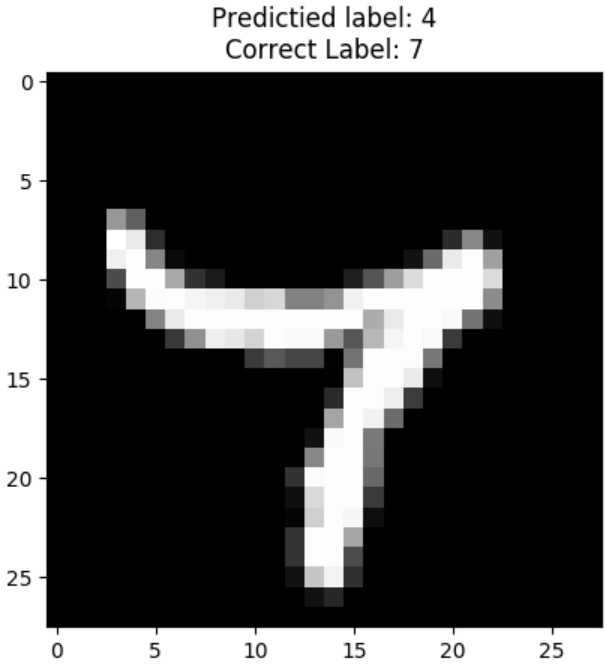

Predictied label: 8
Correct Label: 9
0
5
10
15
20
25
0
5
10
15
20
25


Predictied label: 2
Correct Label: 9
0
5
10
15
20
25
0
5
10
15
20
25


Predictied label: 4
Correct Label: 9
0
5
10
15
20
25
0
5
10
15
20
25


Predictied label: 0
Correct Label: 9
0
5
10
15
20
25
0
5
10
15
20
25


Predictied label: 0
Correct Label: 6
0
5
10
15
20
25
0
5
10
15
20
25


Predictied label: 4
Correct Label: 7
0
5
10
15
20
25
0
5
10
15
20
25