Data Centric AI Competition 2023 [Image Data]

Introduction

Build machine learning model to predict the character of each image.

Dataset Information:

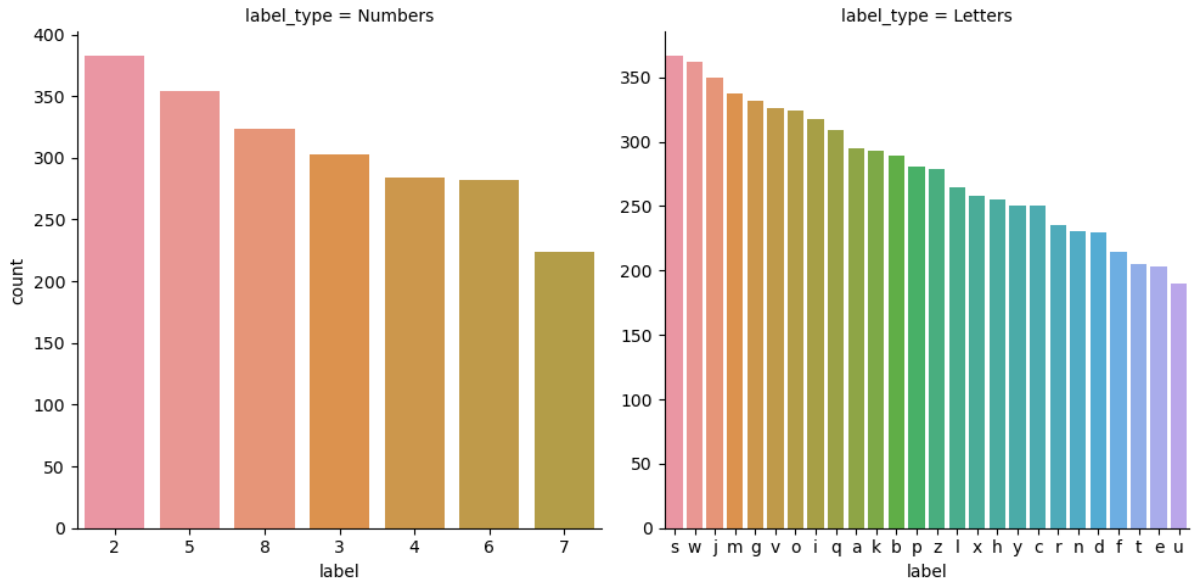
Train data:

Train dataset contains 9402 images.

Test data:

Test dataset contains 1109 images.

Target label distribution:



	label_type	label	count
0	Numbers	2	383
1	Numbers	3	303
2	Numbers	4	284
3	Numbers	5	354
4	Numbers	6	282
5	Numbers	7	224
6	Numbers	8	324

	label_type	label	count
0	Letters	a	295
1	Letters	b	289
2	Letters	c	250
3	Letters	d	230
4	Letters	e	203
5	Letters	f	214
6	Letters	g	332
7	Letters	h	255
8	Letters	i	318
9	Letters	j	350
10	Letters	k	293
11	Letters	l	265
12	Letters	m	337
13	Letters	n	231
14	Letters	o	324
15	Letters	p	281
16	Letters	q	309
17	Letters	r	235
18	Letters	s	367
19	Letters	t	205
20	Letters	u	190
21	Letters	v	326
22	Letters	w	362
23	Letters	x	258
24	Letters	y	250
25	Letters	z	279

All the images are single channel(black & white) image. The width and height of the image is 60 pixels.

Sample images from number type label

Sample images from letter type label

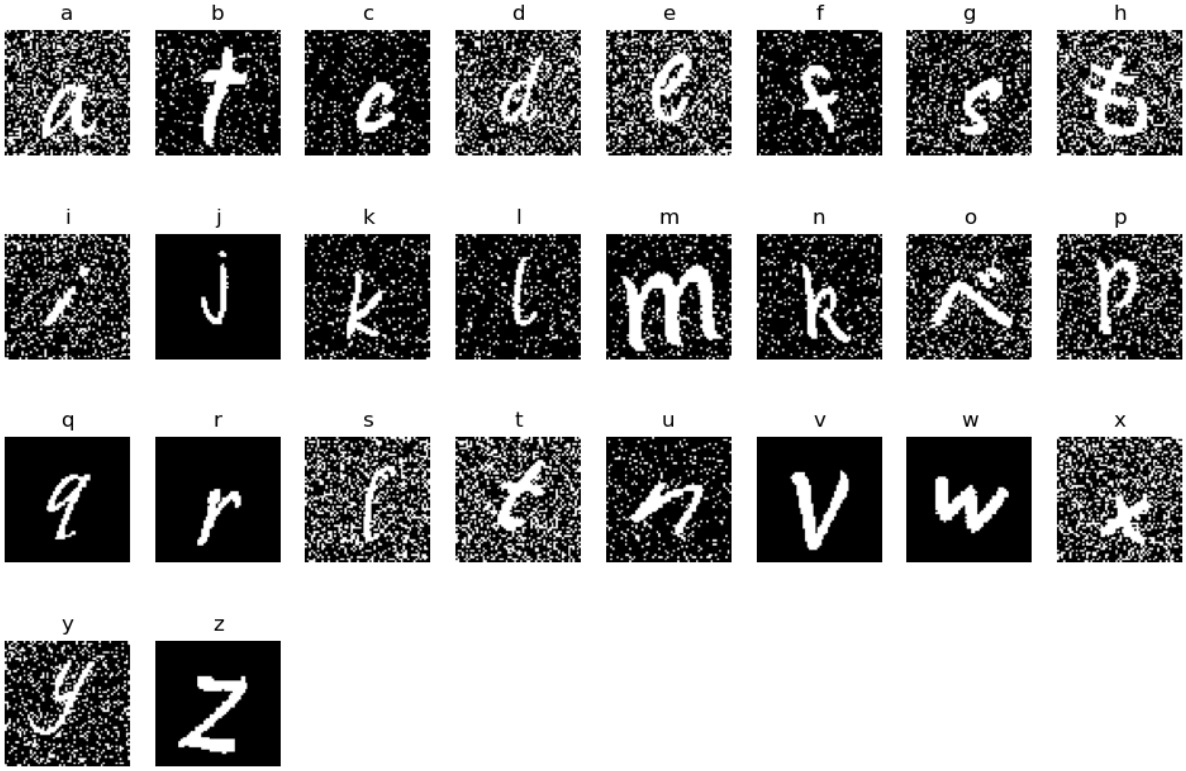


Image similarity analysis

Image similarity analysis had been done by using the various image hash algorithms,

- Average hash
- Perceptual hash
- difference hash
- Wavelet hash
- Color hash

From the above different hashing algorithms, the perceptual and difference hashing algorithms have identified a significant amount of similar images.

Some matched images have noise and rotation from the actual reference images.

The train dataset contains miss labeled images, the incorrectly labeled images are identified by using ,

Simple pytorch convolutional neural network

Skorch - Scikit-Learn compatible neural network library

Cleanlab - No code library to fix the errors in dataset

The simple CNN model run 4 iterations(50 epochs/iteration) and cross-validated on 10-kfold split train dataset.In each iteration, the actual label and the predicted probability for the labels are compared by using the find_label_issue function from cleanlab tool.Then, based on the self-confidence level the miss labeled images are collected.

Four iteration results,

Cross-validated estimate of accuracy on held-out data: 0.6171027440970006
Iteration: 0, Number of Misslabled Images: 2440

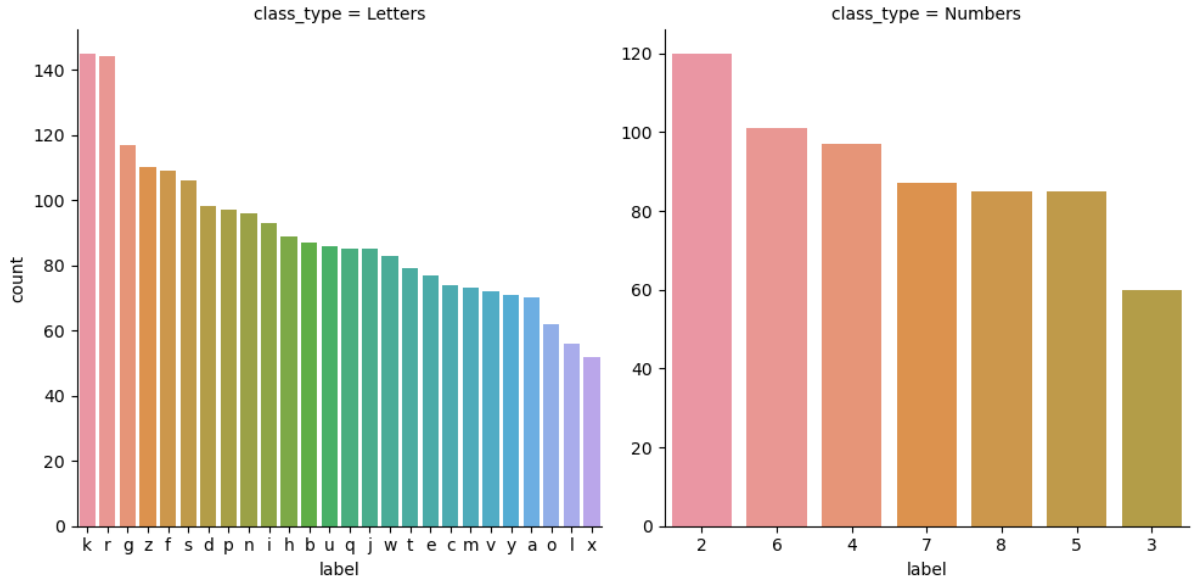
Cross-validated estimate of accuracy on held-out data: 0.859379488652686
Iteration: 1, Number of Misslabled Images: 228

Cross-validated estimate of accuracy on held-out data: 0.8921888921888922
Iteration: 2, Number of Misslabled Images: 87

Cross-validated estimate of accuracy on held-out data: 0.8942380021062133
Iteration: 3, Number of Misslabled Images: 79

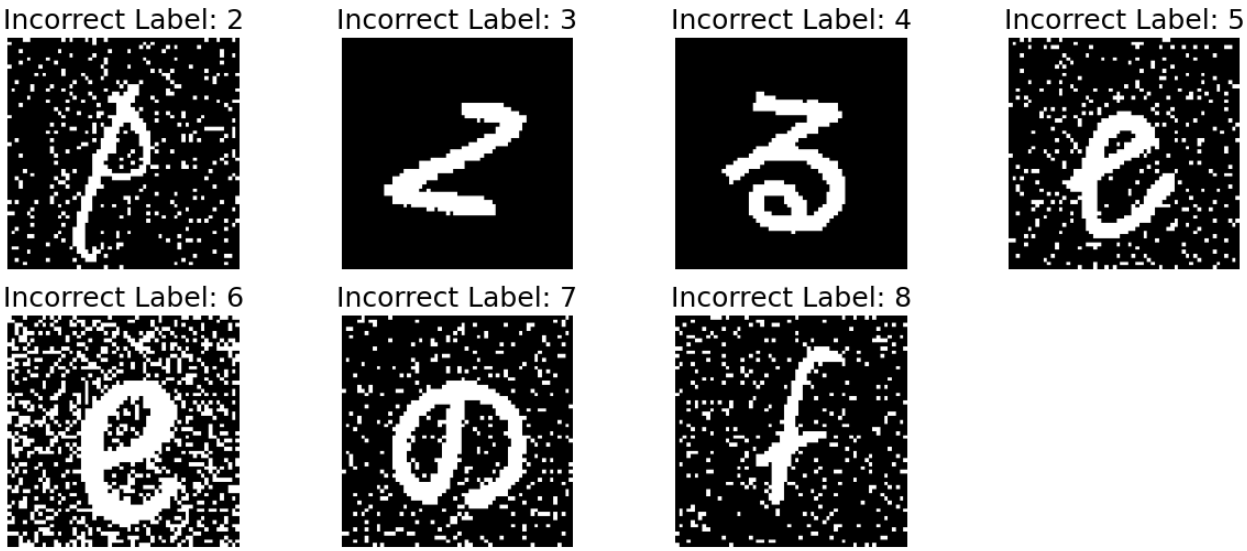
After the 4 iteration,the total number of misslabled images are 2834.

Miss Labeled images



In the letter class images, the images which are labeled k, r, and in the number class images 2, 6 labeled images are mislabeled more than 100.

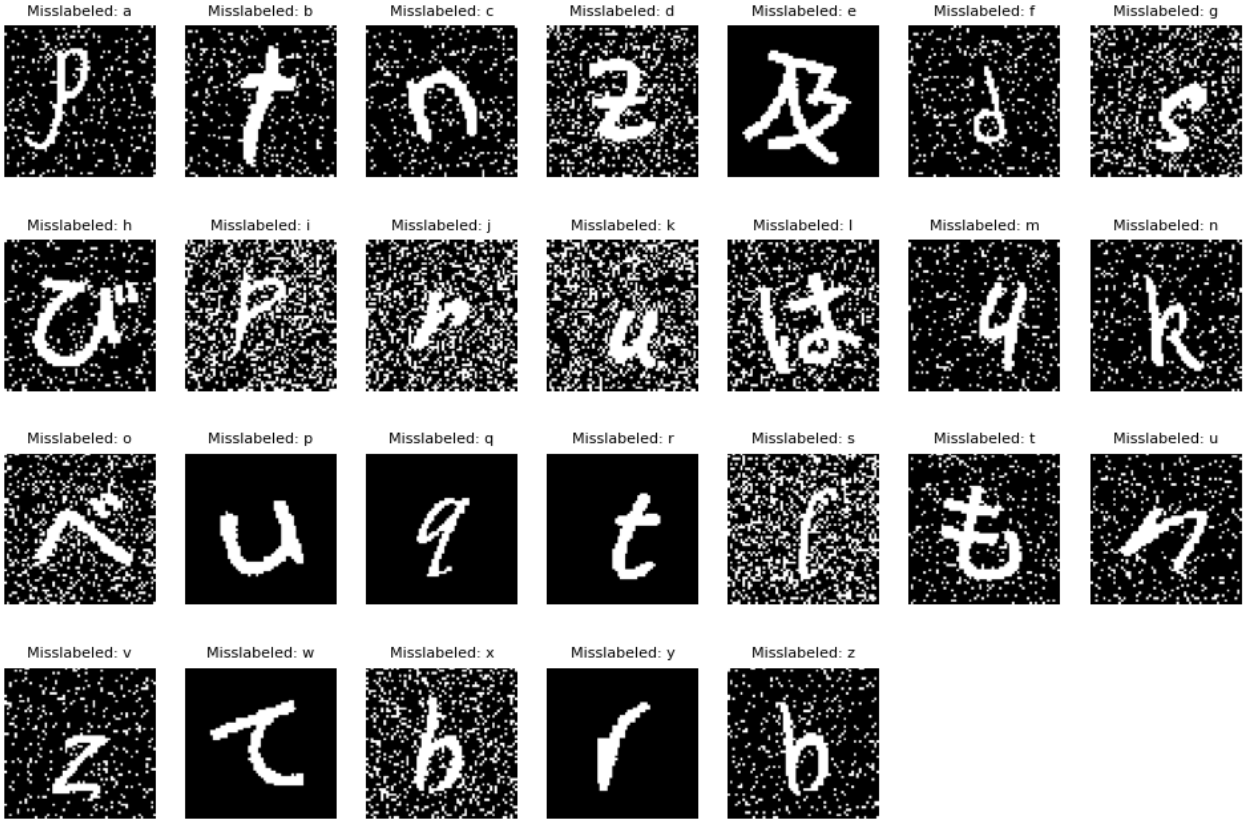
Misslabeled images from number class



The model predicted significant miss-labeled images from the train dataset.

	class_type	label	count
0	Numbers	2	124
1	Numbers	3	54
2	Numbers	4	97
3	Numbers	5	92
4	Numbers	6	90
5	Numbers	7	77
6	Numbers	8	85

Misslabeled images from letter class



	class_type	label	count
0	Letters	a	56
1	Letters	b	89
2	Letters	c	65
3	Letters	d	101
4	Letters	e	72
5	Letters	f	102
6	Letters	g	113
7	Letters	h	87
8	Letters	i	95
9	Letters	j	74
10	Letters	k	138
11	Letters	l	53
12	Letters	m	69
13	Letters	n	87
14	Letters	o	61
15	Letters	p	90
16	Letters	q	73
17	Letters	r	140
18	Letters	s	92
19	Letters	t	75
20	Letters	u	91
21	Letters	v	67
22	Letters	w	80
23	Letters	x	48
24	Letters	y	78
25	Letters	z	119

The model predicted significant miss-labeled images from the train dataset.

Data preparation for model

Miss labeled images are removed from the actual train dataset.Each image are resized into 28 pixels and converted to a 1-D vector then create a dataframe from the converted image array.

The converted dataframe shape is 6451 rows and 784(28*28) columns.

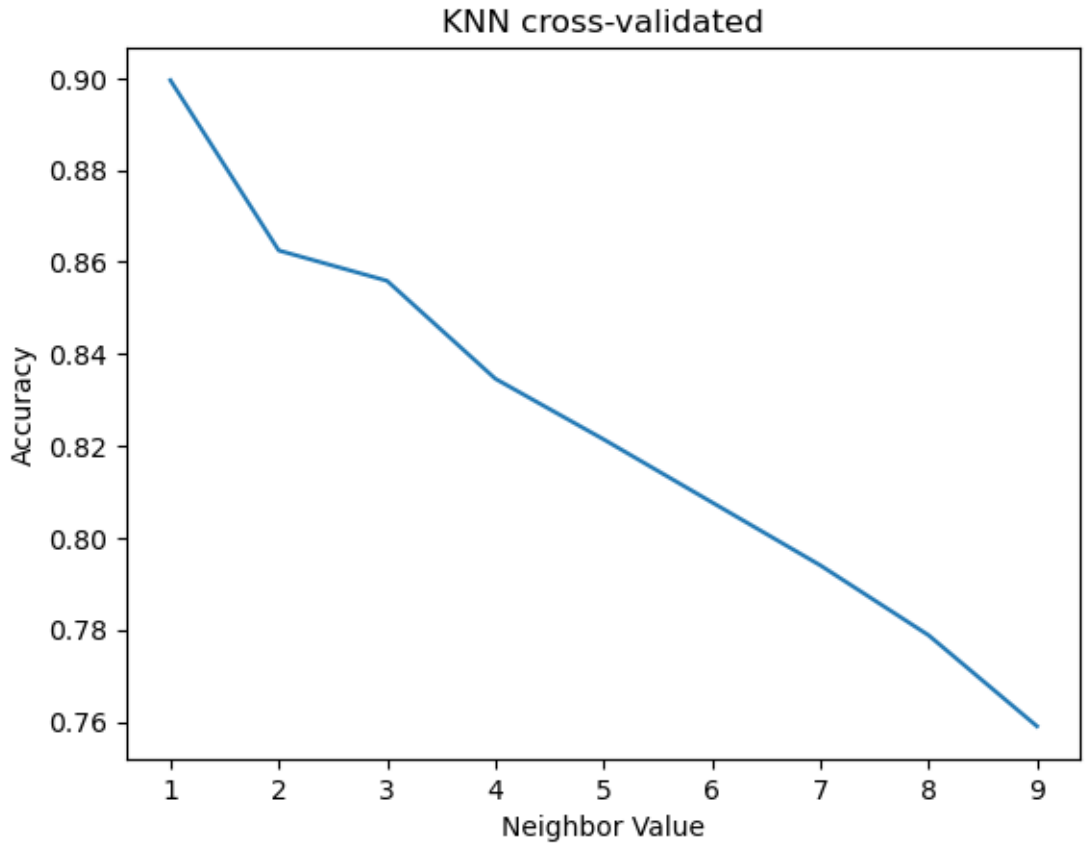
The converted dataframe is split(stratified) into train and test data.

The split train data shape is 4515 rows and 784 columns.

The split test data shape is 1936 rows and 784 columns.

Model

Created KNN model and cross validated with different neighbor values.



neighbor:1, score:0.8995433789954338

neighbor:2, score:0.8625063419583967

neighbor:3, score:0.8559107052257737

neighbor:4, score:0.8346017250126839

neighbor:5, score:0.8214104515474379

neighbor:6, score:0.8077118214104515

neighbor:7, score:0.7940131912734653

neighbor:8, score:0.7787924911212583

neighbor:9, score:0.7590055809233891

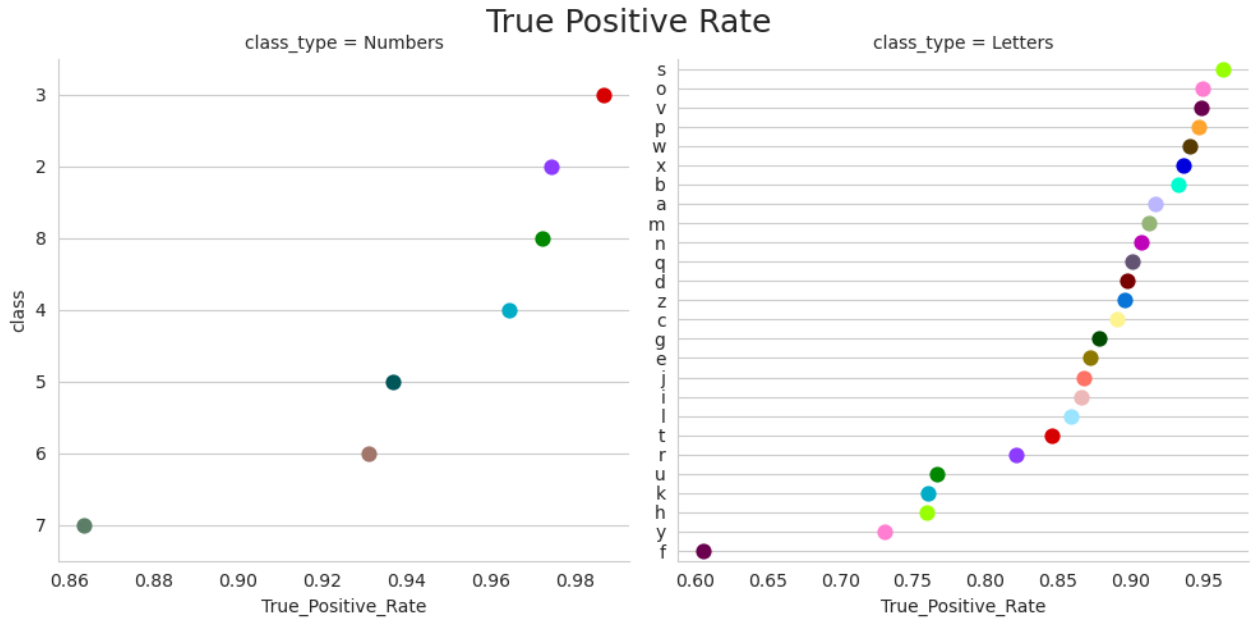
The KNN model with 1 neighbor value gives significant accuracy on test data.

KNN model created with 1 neighbor value and tested on test dataset.The model gives 0.8995 accuracy on test data.

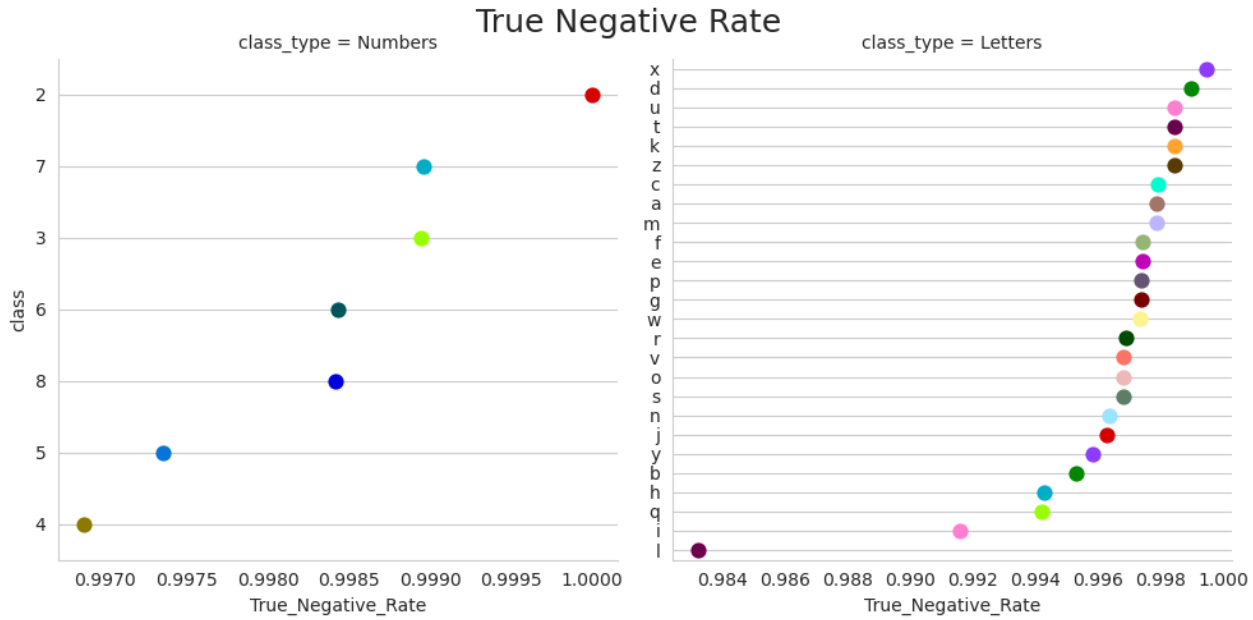
Predicted probability to calculate roc-auc score.

KNN model evaluation results

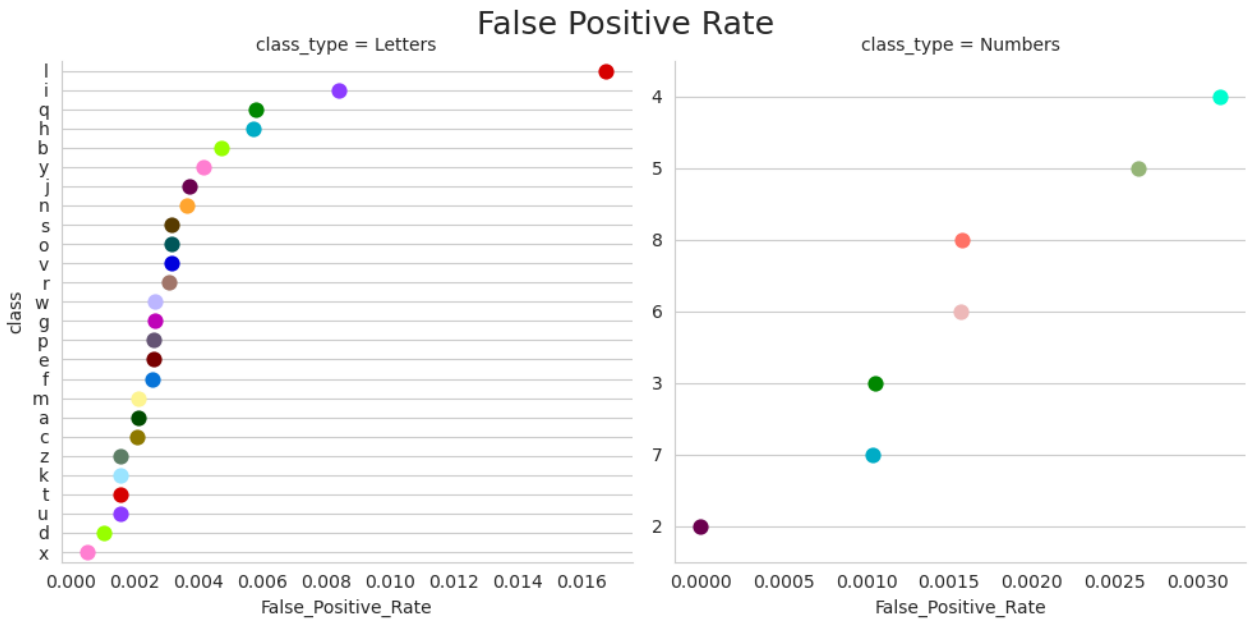
- **True positive rate**



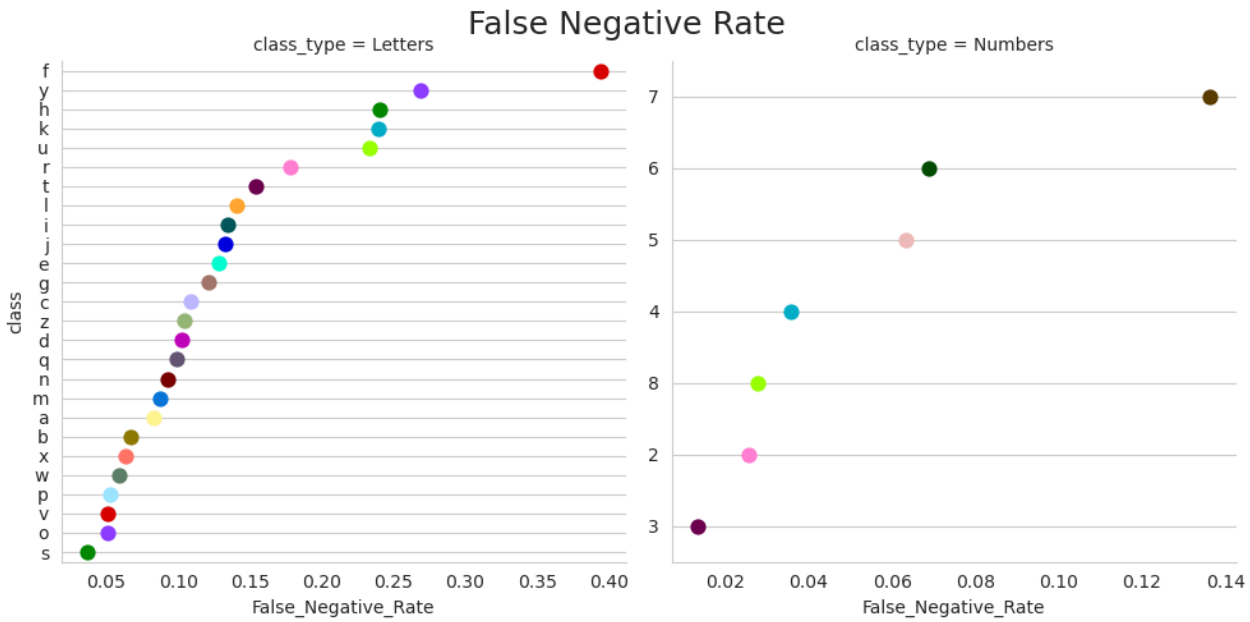
- **True negative rate**



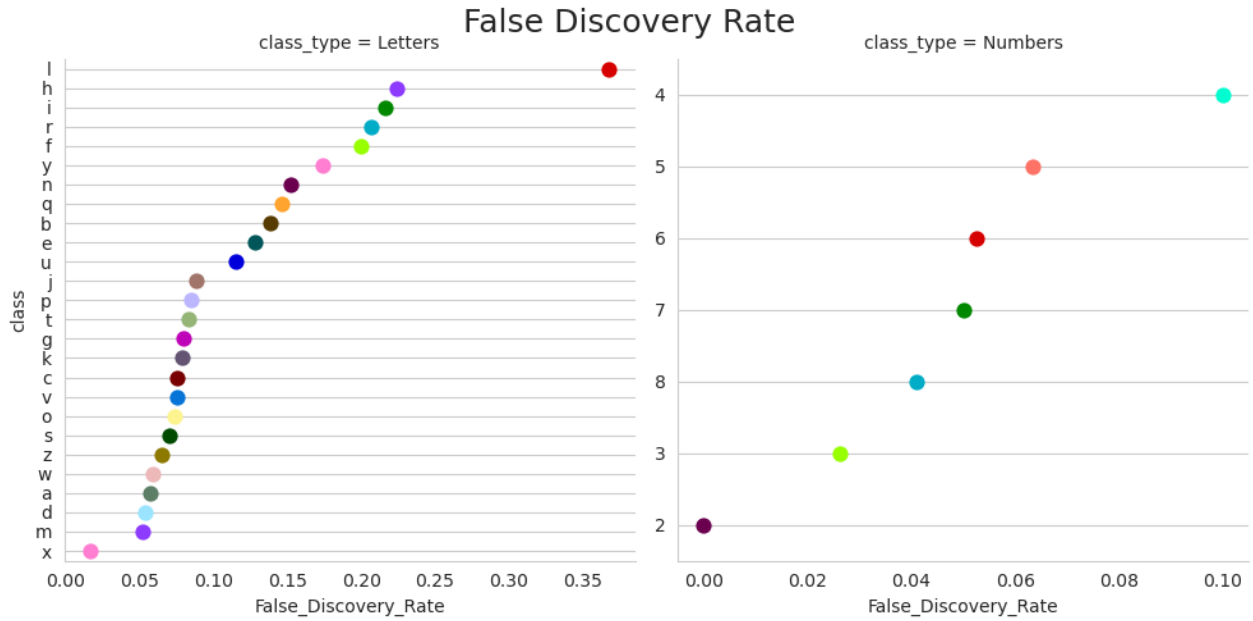
- **False positive rate**



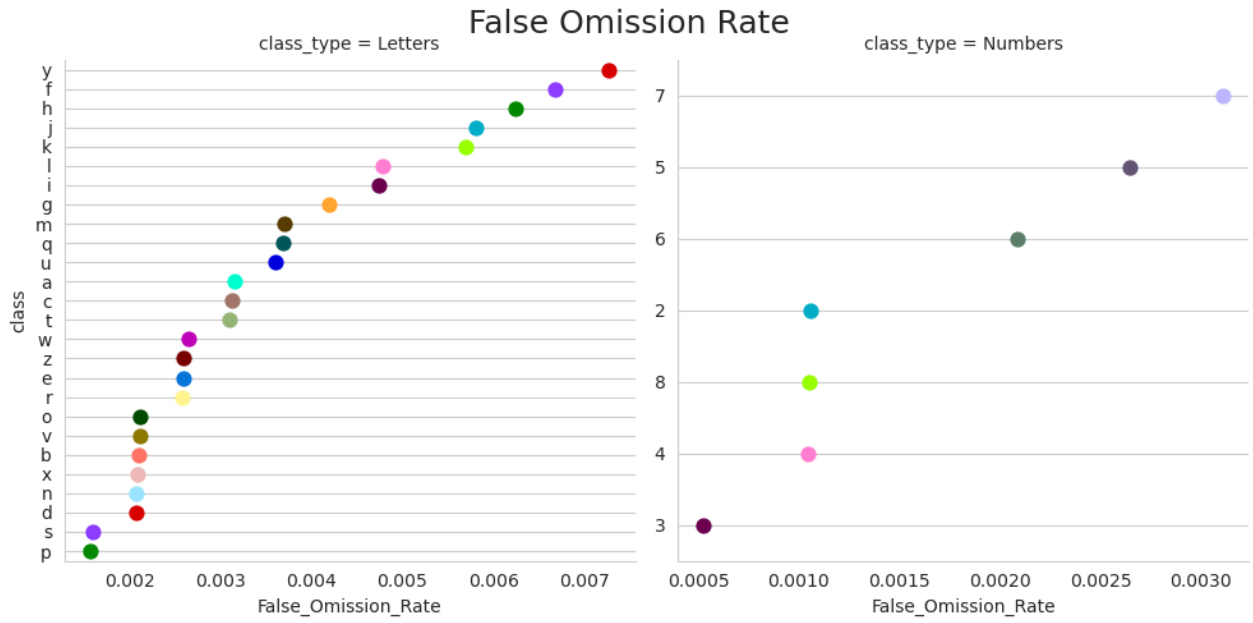
- **False negative rate**



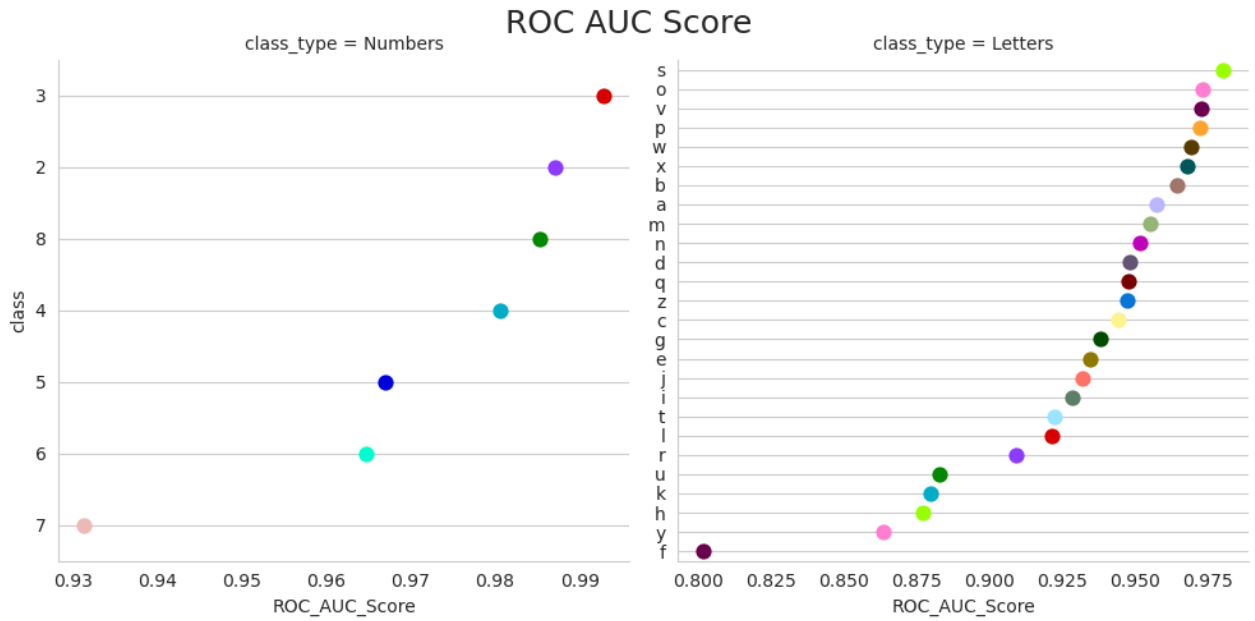
- **False discovery rate**



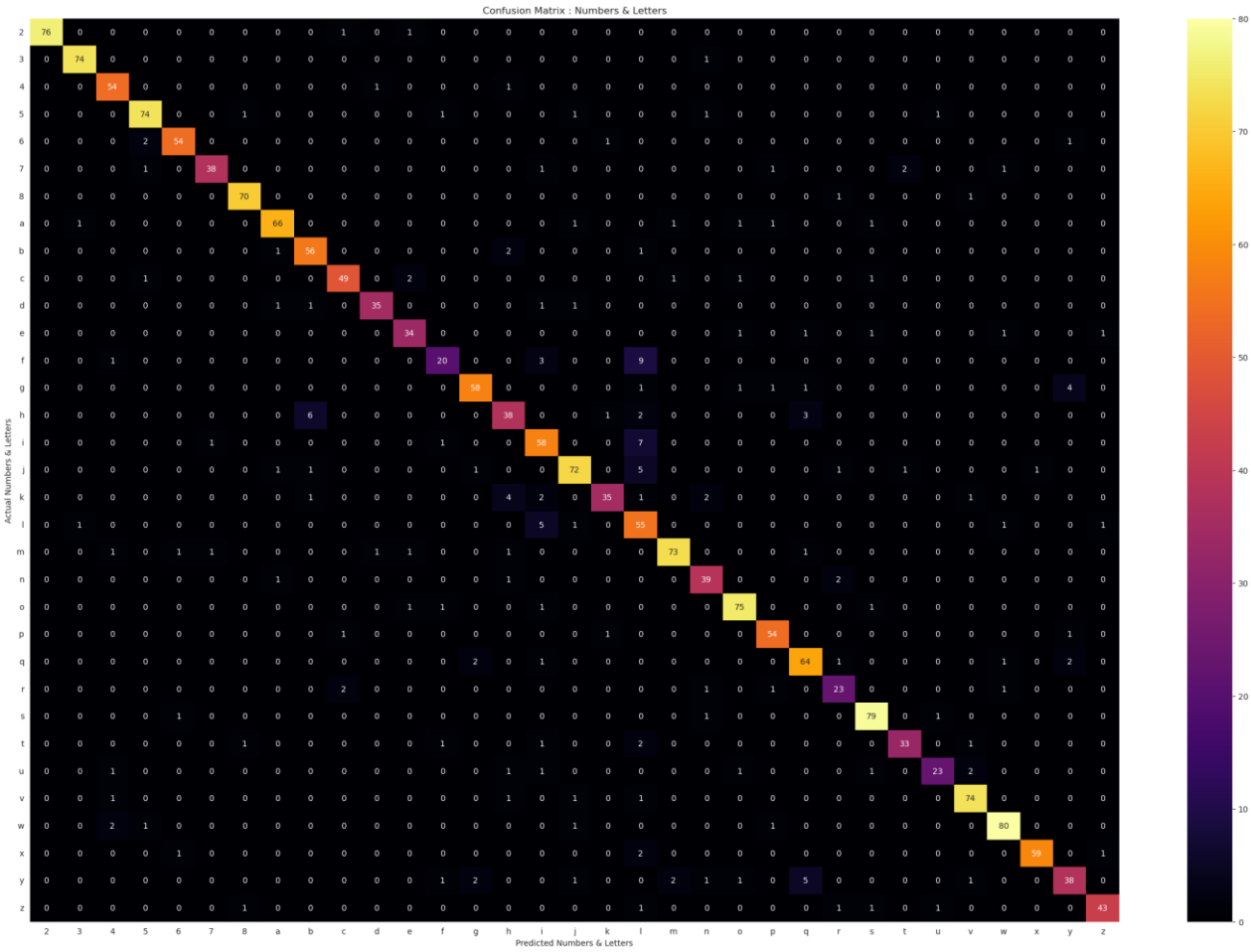
- **False omission rate**



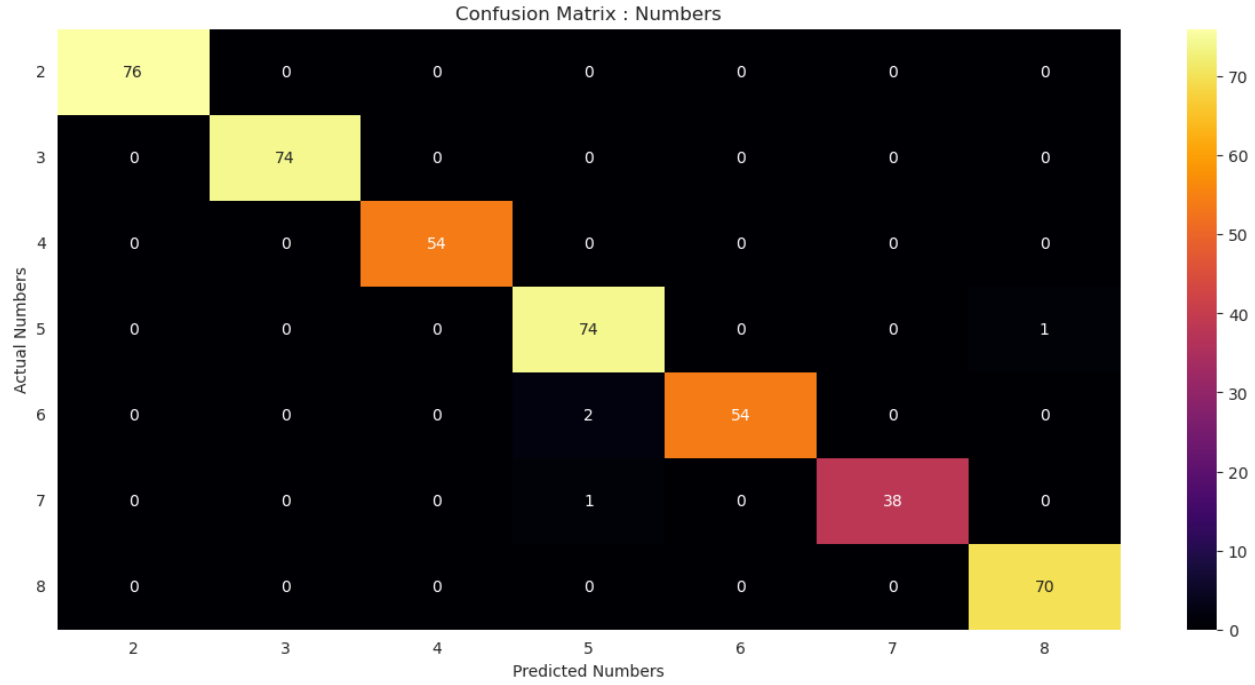
- **ROC-AUC score**

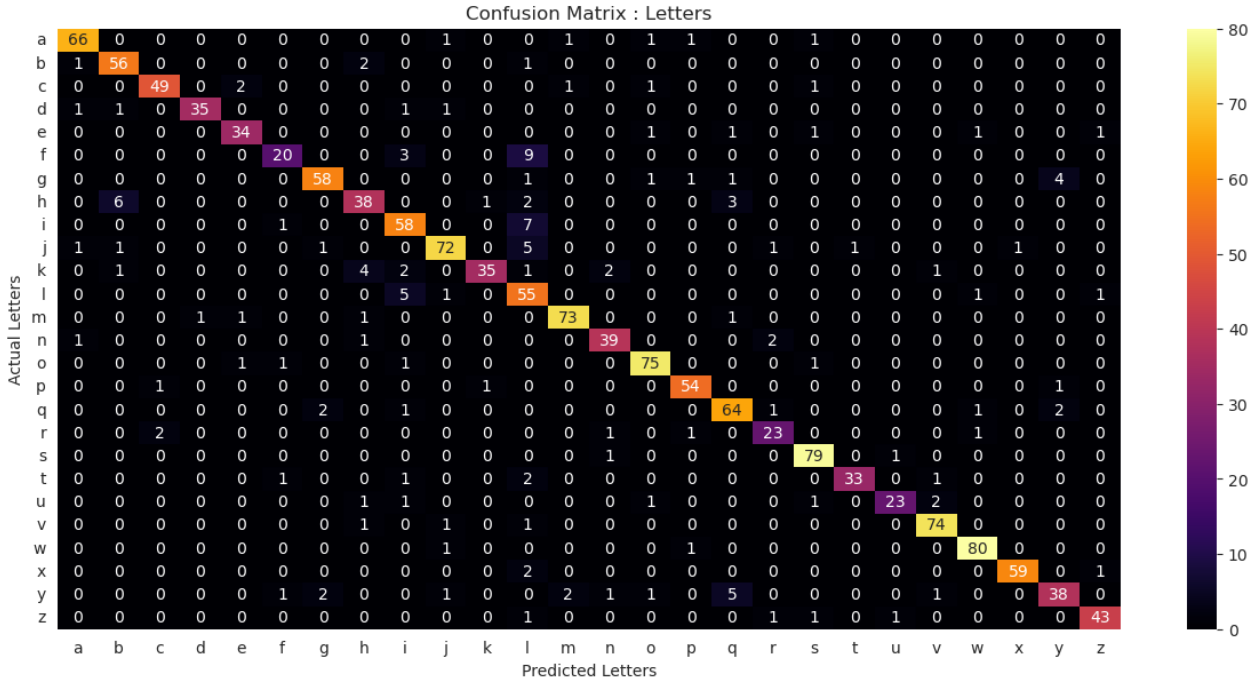


- **Confusion matrix**



- **Confusion matrix for seperate class type**





Model blend

Created three shallow models and one simple convolutional model

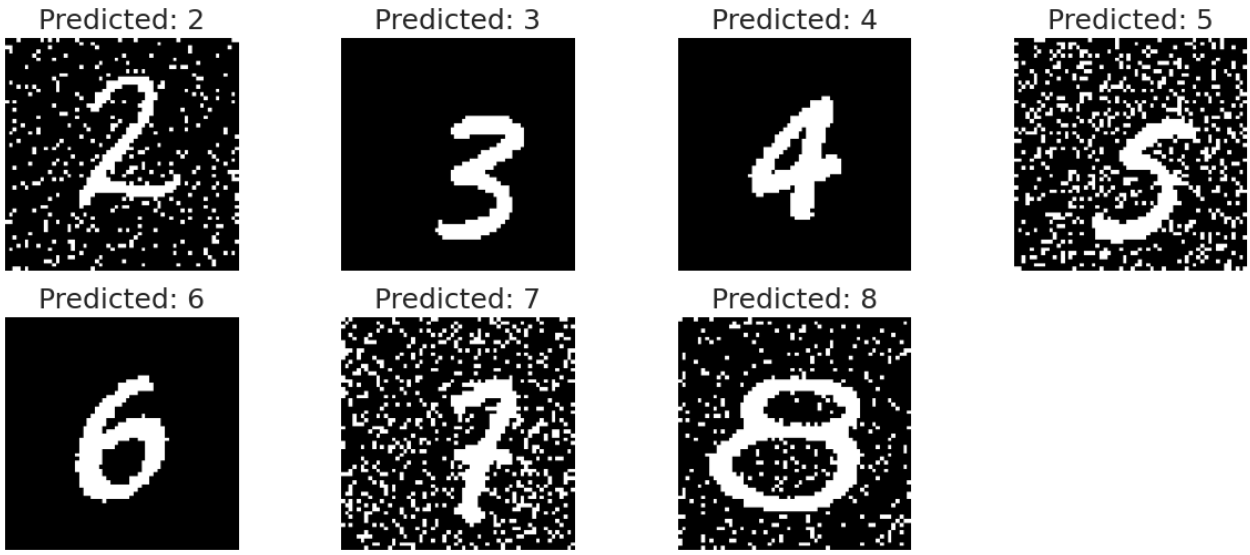
- KNN Classifier
- Random Forest Classifier
- Extratree Classifier
- CNN model

Models accuracy score,

- KNN accuracy_score: 0.9045203
- Random_forest accuracy_score: 0.9008303
- Extratree accuracy_score: 0.9035978
- Nnet accuracy_score: 0.895295

Model blending has not given expected results. So, the KNN model used to predict the test data.

Test prediction: Number class



Test prediction: Letter class