Machine Learning Assignment 2: Naive Bayes & ANNs

30.04.2022

—

Done By

Maitreya Manohar (2021A7PS2663H)

Ayush Bhauwala (2021A7PS0180H)

Arunachala Amuda Murugan (2021A7PS0205H)

Part 1: Naive Bayes

Model 1: Naive Bayes

Notes

1. Conversion of continuous columns to categorical using **pandas.cut**, allowed us to speed up the training process as well as increase the accuracy from 80% to 83.5%.
2. Data preprocessing: imputed categorical columns using its mode and numerical columns with its mean.
3. Smoothing techniques allowed in the model: Laplace smoothing, Ignoring features not present in distribution.
4. We create a Distribution object for every class (here 2, but extendable to n classes as well), where we store its prior probability and the likelihood of each feature in a distribution in a dictionary, whose keys are in the format **(column, value)**.

Accuracies

split 1	0.8335194490973385
split 2	0.8289596128792108
split 3	0.8324958123953099
split 4	0.8322166387493021
split 5	0.8334263912153359
split 6	0.8351944909733855
split 7	0.8273776288851665
split 8	0.8331472175693281
split 9	0.8351944909733855
split 10	0.8335194490973385

Mean Accuracy	0.8325051181835101
Standard deviation	0.0025106281502815394

Average precision	0.8253973136540113
Average recall	0.8029594125987934

Model 2: KNN

Accuracies

split 1	0.8323096966313047
split 2	0.828680439233203
split 3	0.8306346547552578
split 4	0.8259817606551275
split 5	0.824213660897078
split 6	0.8310999441652708
split 7	0.8325888702773125
split 8	0.8327749860413177
split 9	0.830820770519263
split 10	0.8306346547552578

Mean Accuracy	0.8299739437930391
Standard deviation	0.0028633513601374907
Average precision	0.8606935915399496
Average recall	0.9257837842067571

Model 3 : Logistic Regression

Accuracies

split 1	0.8528754885538805
split 2	0.844965568583659
split 3	0.8517587939698492
split 4	0.8484087102177554
split 5	0.8467336683417085
split 6	0.8463614368136981
split 7	0.853247720081891
split 8	0.8503629257398102
split 9	0.8481295365717476
split 10	0.8494323469197841

Mean Accuracy	0.8492276195793783
Standard deviation	0.0028221762132697053
Average precision	0.8787840738371813
Average recall	0.9257837842067571

	index	accuracy	precision	recall	f1 score	knn accuracy	knn recall	knn f1 score	knn precision	logreg accuracy	logreg recall	logreg f1 score	logreg precision
0	Accuracy 1	0.833519	0.849771	0.922736	0.884752	0.832310	0.923707	0.893448	0.865109	0.852875	0.931165	0.905966	0.882094
1	Accuracy 2	0.828960	0.844551	0.922321	0.881725	0.828680	0.920850	0.891187	0.863376	0.844966	0.925369	0.900939	0.877766
2	Accuracy 3	0.832496	0.845331	0.929412	0.885380	0.830635	0.925163	0.892320	0.861730	0.851759	0.929088	0.904833	0.881812
3	Accuracy 4	0.832217	0.796911	0.617779	0.696004	0.825982	0.926305	0.889205	0.854962	0.848409	0.931366	0.902566	0.875493
4	Accuracy 5	0.833426	0.800391	0.615038	0.695578	0.824214	0.926364	0.888456	0.853528	0.846734	0.928703	0.901560	0.875958
5	Accuracy 6	0.835194	0.849188	0.928438	0.887046	0.831100	0.928343	0.892737	0.859761	0.846361	0.926991	0.901344	0.877079
6	Accuracy 7	0.827378	0.843742	0.922480	0.881356	0.832589	0.922429	0.893557	0.866437	0.853248	0.927681	0.905935	0.885185
7	Accuracy 8	0.833147	0.783888	0.621049	0.693032	0.832775	0.927064	0.894288	0.863750	0.850363	0.931089	0.904717	0.879797
8	Accuracy 9	0.835194	0.794942	0.621539	0.697627	0.830821	0.930215	0.892642	0.857986	0.848130	0.933785	0.902892	0.873978
9	Accuracy 10	0.833519	0.845257	0.928803	0.885063	0.830635	0.927398	0.892587	0.860296	0.849432	0.929973	0.903599	0.878679

Part 2: ANNs

Dataset: MNIST

Used the inbuilt MNIST (80-20 split) train and test datasets provided in **keras.datasets**

Models Used

Using the various combinations from the given parameter grid, we have trained, tested and documented the results of 15 models on 10 splits each.

The output layer is a 10 neuron layer, with a softmax activation function.

Hyperparameters

Number of hidden layers	Number of neurons	Activation functions
2	100	relu
3	150	tanh
		sigmoid

Model details with accuracies

Model no.	No. of Inner Layers	Neurons	Activation function	Accuracy
1	2	150, 150	sigmoid, sigmoid	95.77%
2	2	100, 100	sigmoid, sigmoid	95.21%
3	2	100, 150	sigmoid, sigmoid	95.52%
4	2	150, 100	sigmoid, sigmoid	95.68%
5	2	150, 150	relu, relu	97.48%
6	2	100, 100	relu, relu	97.85%
7	2	100, 150	relu, relu	97.99%
8	2	150, 100	relu, relu	97.96%
9	2	150, 150	tanh, tanh	94.06%
10	2	100, 100	tanh, tanh	94.16%
11	2	100, 150	tanh, tanh	94.48%
12	2	150, 100	tanh, tanh	94.56%
13	3	150, 150, 150	tanh, tanh, tanh	94.11%
14	3	150, 150, 150	relu, relu, relu	98.25%
15	3	150, 150, 150	sigmoid, sigmoid, sigmoid	95.92%

Analysis

On analysing the accuracies and confusion matrices of all 15 models, we find that the models with the best performance are the ones that use the ReLU activation function.

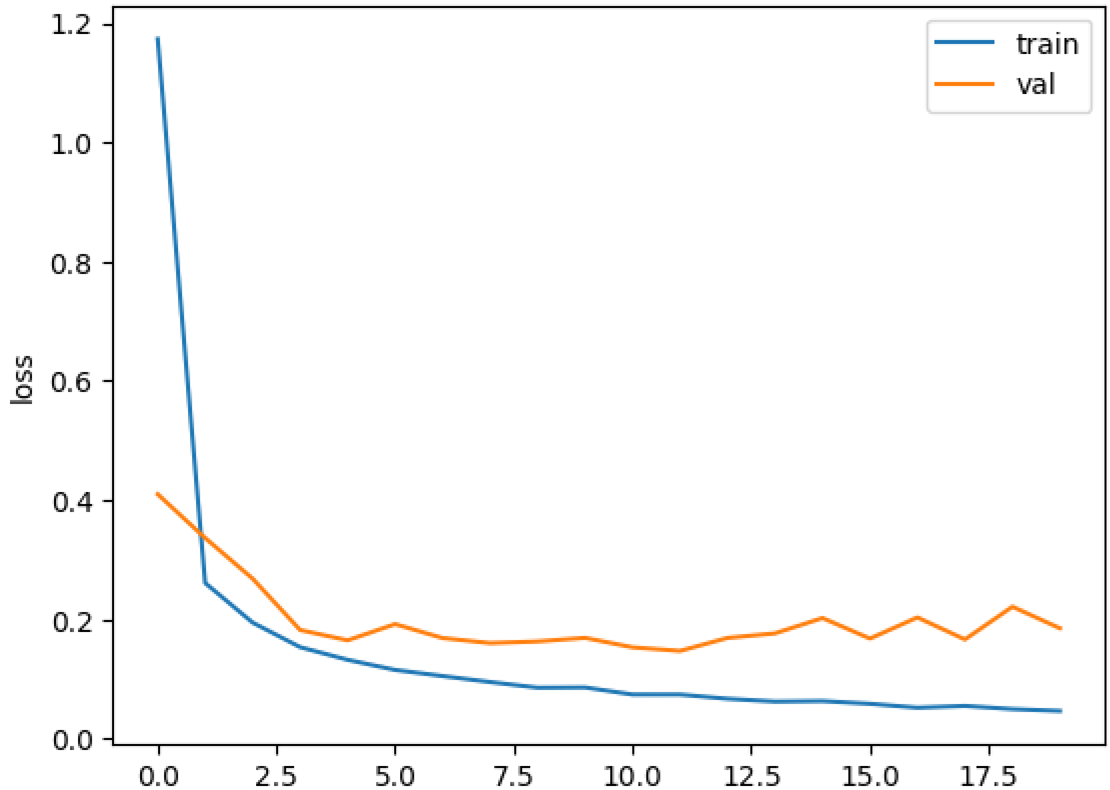
The best performing model is model 14 which has 3 inner layers and uses the ReLU activation function.

We find that after ReLU activation function, sigmoid function performs the best followed by tanh function.

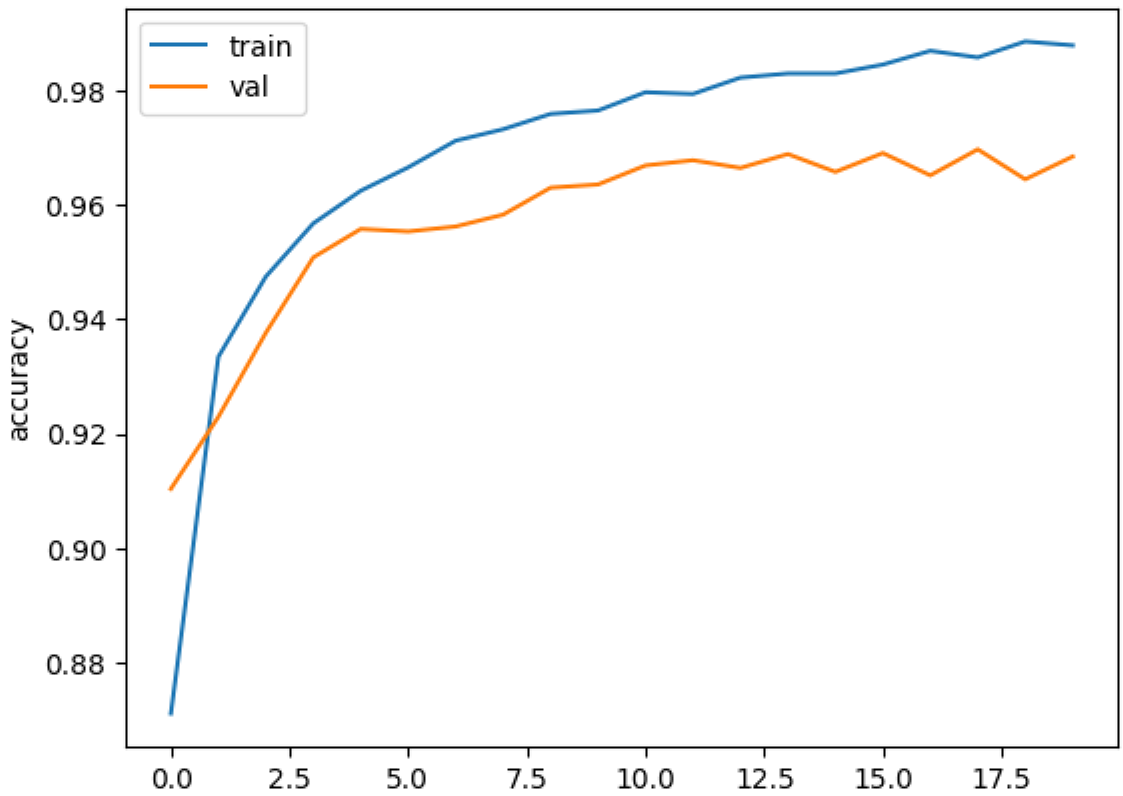
All models though had different performances, were statistically significant, as the losses were reducing with increase in epochs. (as seen below)

Train and Test plots

Loss curves

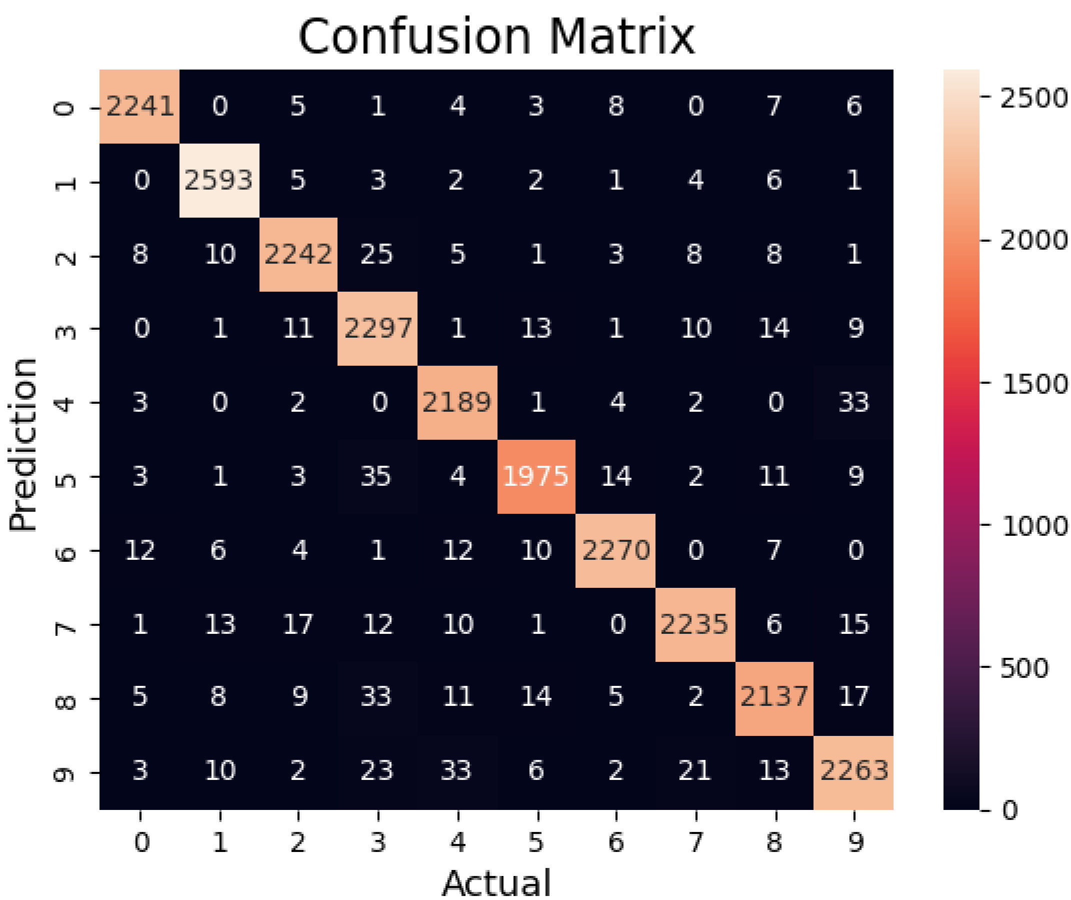


Accuracy curves

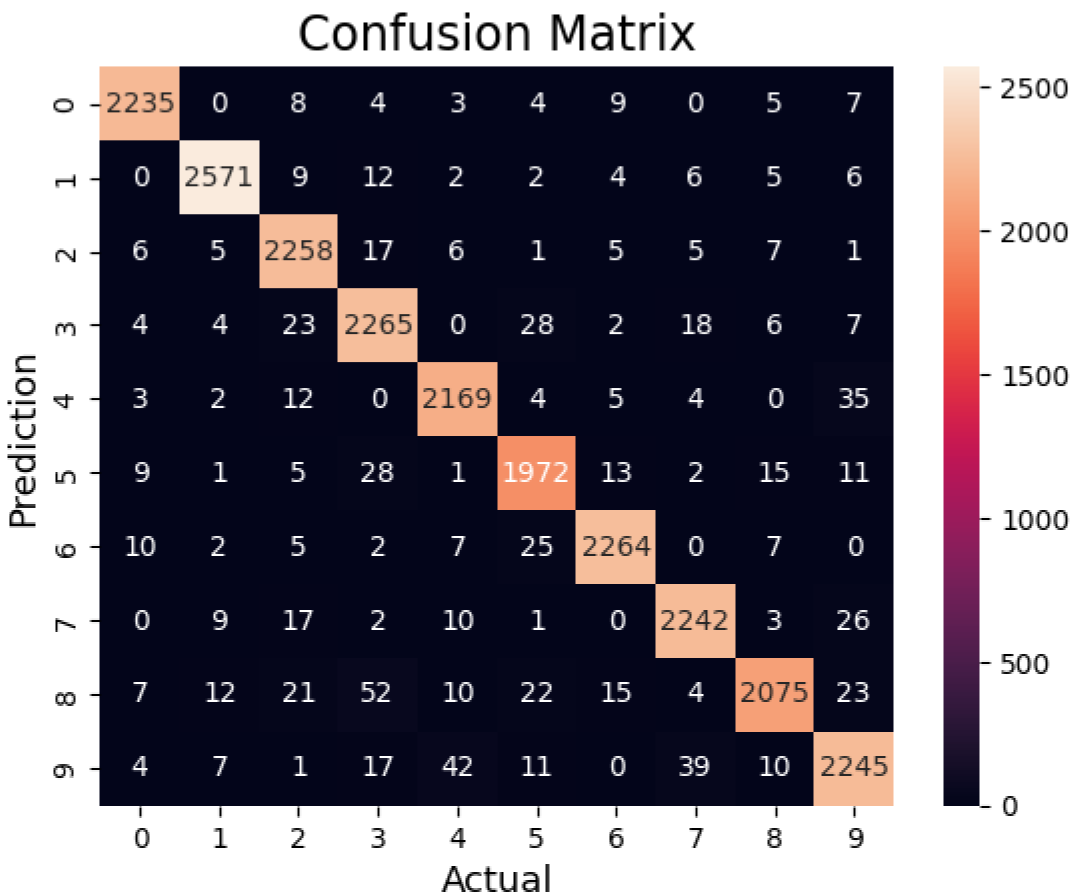


Confusion Matrices

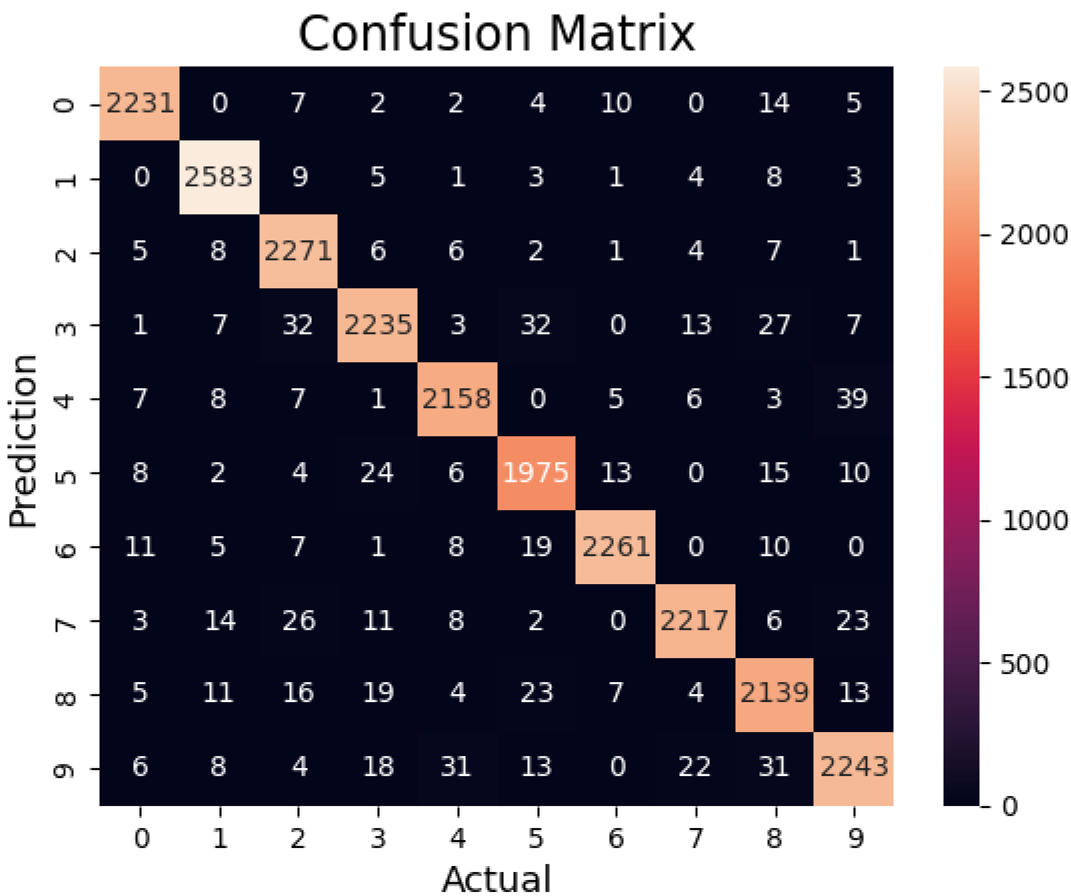
Model 1



Model 2

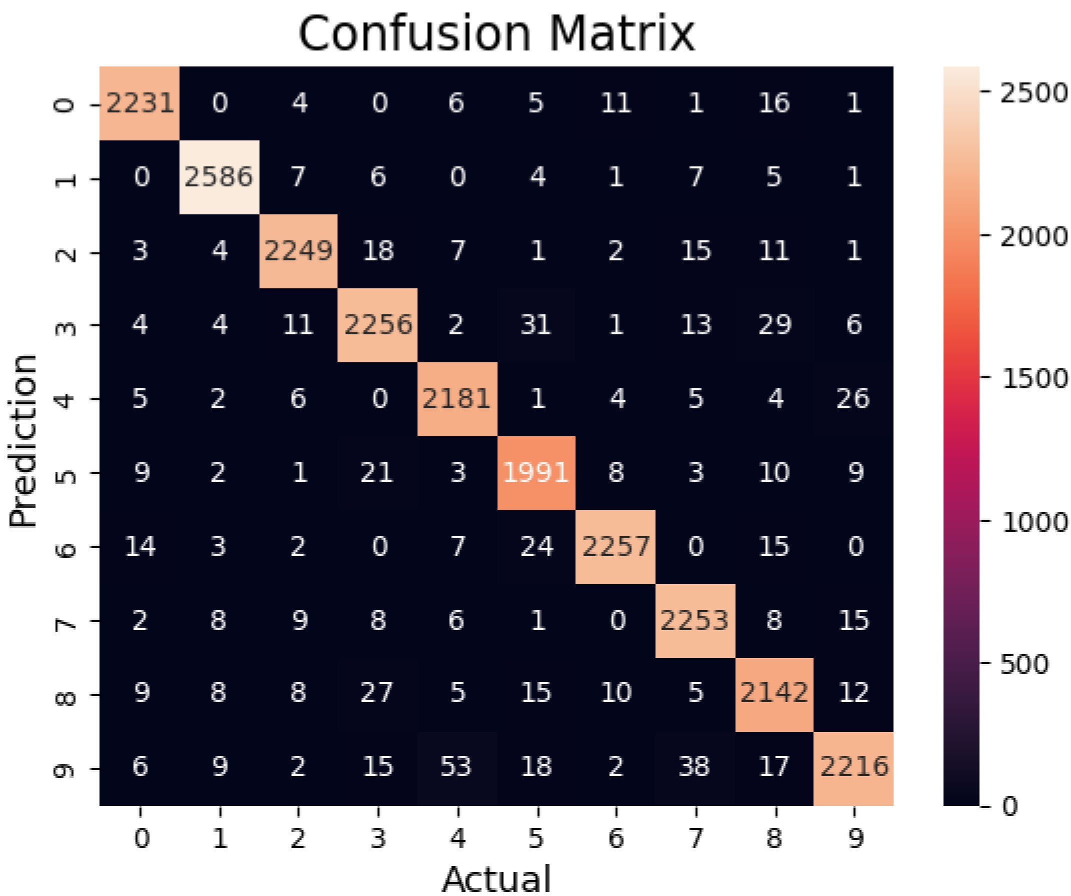

Confusion Matrix
Prediction
Actual
0 1 2 3 4 5 6 7 8 9
2235 0 8 4 3 4 9 0 5 7
0 2571 9 12 2 2 4 6 5 6
6 5 2258 17 6 1 5 5 7 1
4 4 23 2265 0 28 2 18 6 7
3 2 12 0 2169 4 5 4 0 35
9 1 5 28 1 1972 13 2 15 11
10 2 5 2 7 25 2264 0 7 0
0 9 17 2 10 1 0 2242 3 26
7 12 21 52 10 22 15 4 2075 23
4 7 1 17 42 11 0 39 10 2245
2500
2000
1500
1000
500
0

Model 3

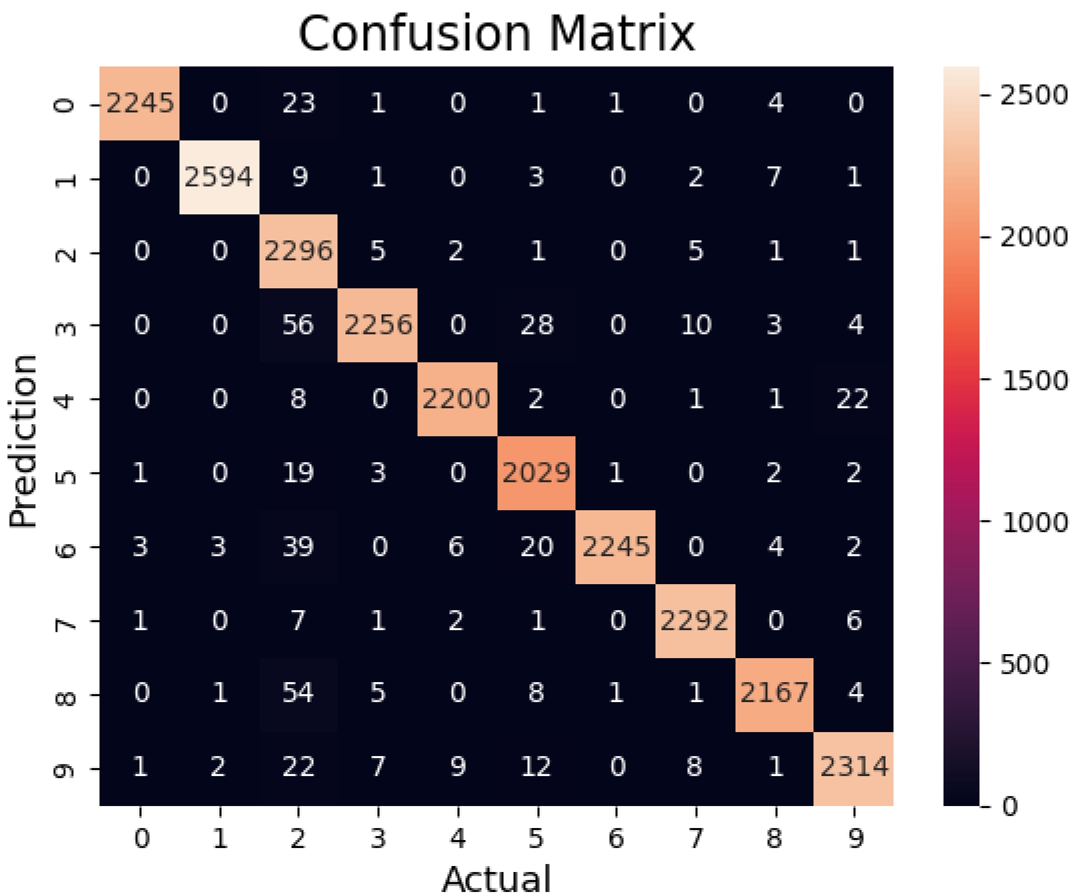


Confusion Matrix
Prediction
Actual
0 1 2 3 4 5 6 7 8 9
2231 0 7 2 2 4 10 0 14 5
0 2583 9 5 1 3 1 4 8 3
5 8 2271 6 6 2 1 4 7 1
1 7 32 2235 3 32 0 13 27 7
7 8 7 1 2158 0 5 6 3 39
8 2 4 24 6 1975 13 0 15 10
11 5 7 1 8 19 2261 0 10 0
3 14 26 11 8 2 0 2217 6 23
5 11 16 19 4 23 7 4 2139 13
6 8 4 18 31 13 0 22 31 2243
2500
2000
1500
1000
500
0

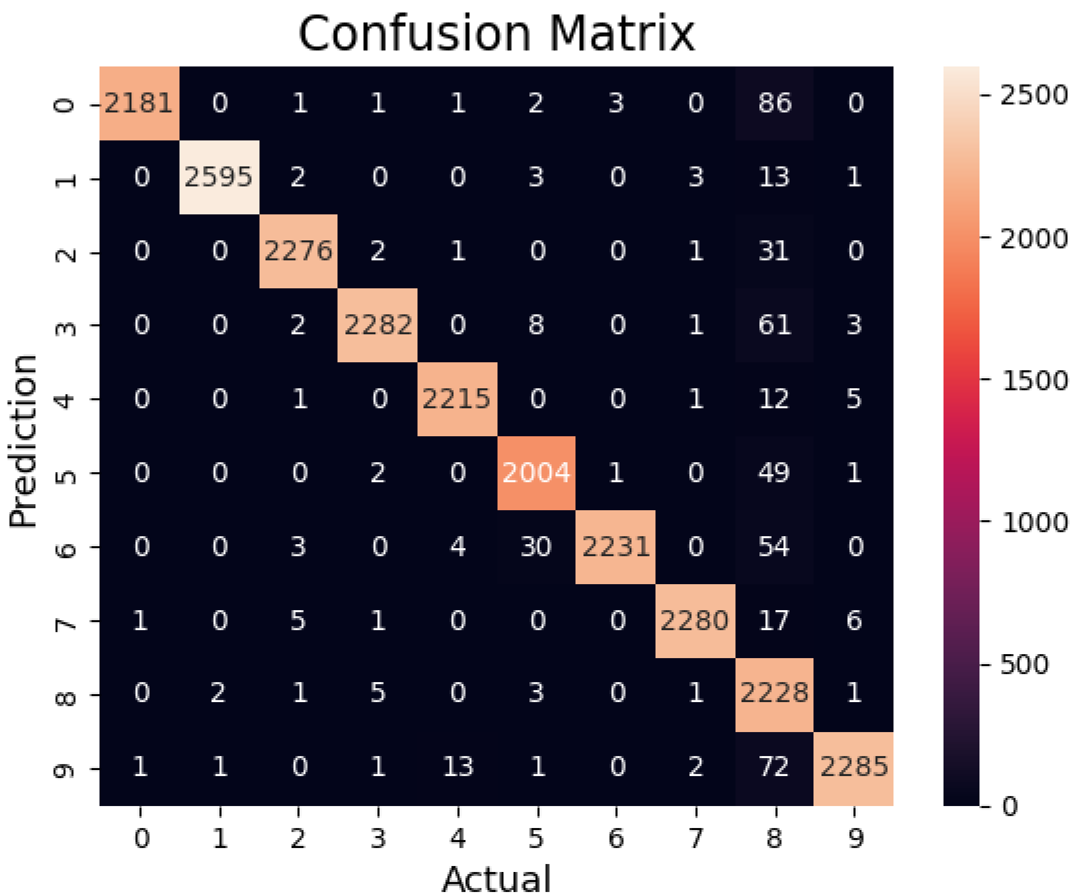
Model 4


Confusion Matrix
Prediction
Actual
0
1
2
3
4
5
6
7
8
9
2231 0 4 0 6 5 11 1 16 1
0 2586 7 6 0 4 1 7 5 1
3 4 2249 18 7 1 2 15 11 1
4 4 11 2256 2 31 1 13 29 6
5 2 6 0 2181 1 4 5 4 26
9 2 1 21 3 1991 8 3 10 9
14 3 2 0 7 24 2257 0 15 0
2 8 9 8 6 1 0 2253 8 15
9 8 8 27 5 15 10 5 2142 12
6 9 2 15 53 18 2 38 17 2216
2500
2000
1500
1000
500
0

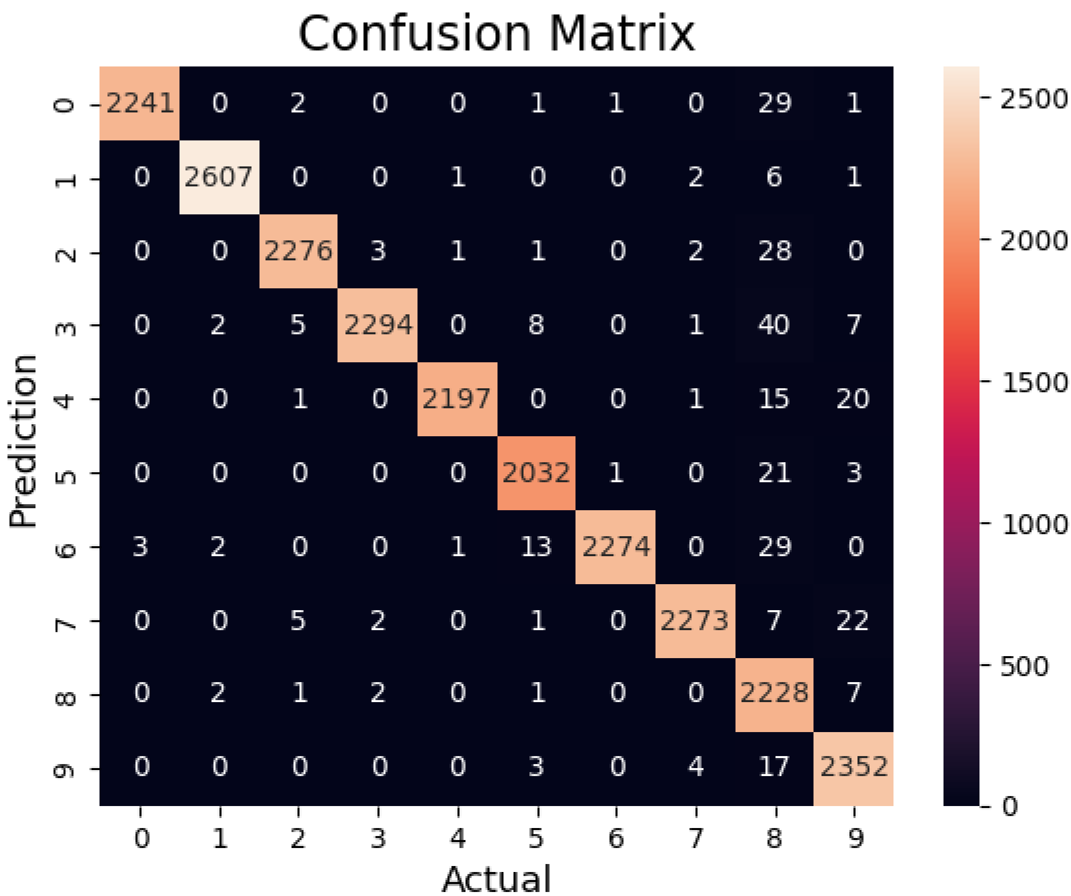
Model 5


Confusion Matrix
Prediction
Actual
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
2245 0 23 1 0 1 1 0 4 0
0 2594 9 1 0 3 0 2 7 1
0 0 2296 5 2 1 0 5 1 1
0 0 56 2256 0 28 0 10 3 4
0 0 8 0 2200 2 0 1 1 22
1 0 19 3 0 2029 1 0 2 2
3 3 39 0 6 20 2245 0 4 2
1 0 7 1 2 1 0 2292 0 6
0 1 54 5 0 8 1 1 2167 4
1 2 22 7 9 12 0 8 1 2314
2500
2000
1500
1000
500
0

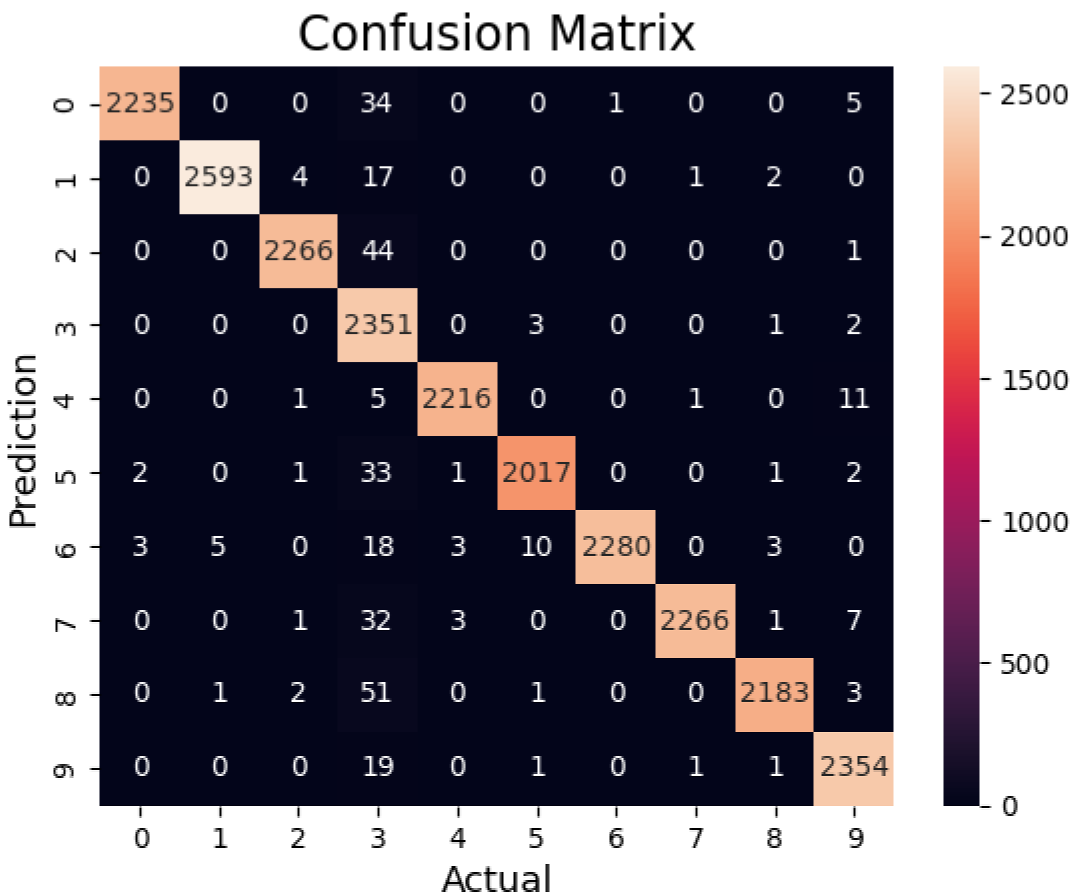
Model 6


Confusion Matrix
Prediction
Actual
2181 0 1 1 1 2 3 0 86 0
0 2595 2 0 0 3 0 3 13 1
0 0 2276 2 1 0 0 1 31 0
0 0 2 2282 0 8 0 1 61 3
0 0 1 0 2215 0 0 1 12 5
0 0 0 2 0 2004 1 0 49 1
0 0 3 0 4 30 2231 0 54 0
1 0 5 1 0 0 0 2280 17 6
0 2 1 5 0 3 0 1 2228 1
1 1 0 1 13 1 0 2 72 2285
0 1 2 3 4 5 6 7 8 9
2500
2000
1500
1000
500
0

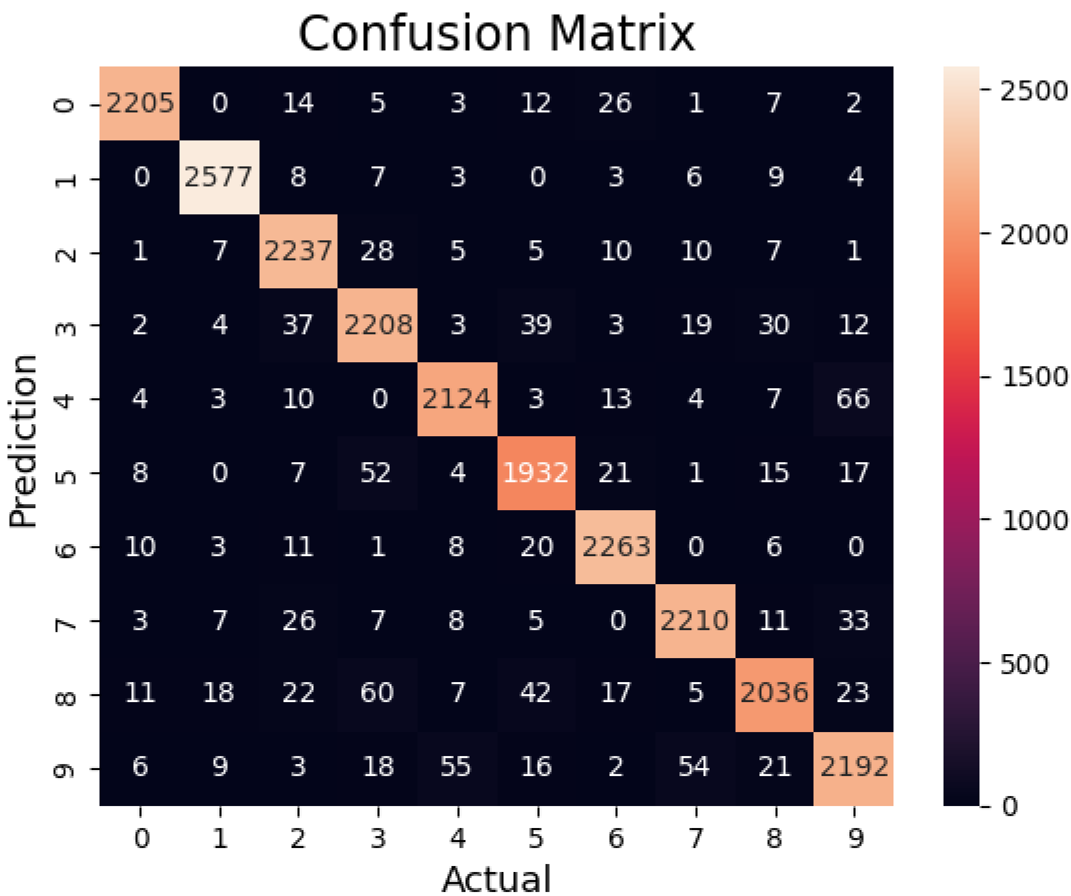
Model 7


Confusion Matrix
Prediction
Actual
0 1 2 3 4 5 6 7 8 9
2241 0 2 0 0 1 1 0 29 1
0 2607 0 0 1 0 0 2 6 1
0 0 2276 3 1 1 0 2 28 0
0 2 5 2294 0 8 0 1 40 7
0 0 1 0 2197 0 0 1 15 20
0 0 0 0 0 2032 1 0 21 3
3 2 0 0 1 13 2274 0 29 0
0 0 5 2 0 1 0 2273 7 22
0 2 1 2 0 1 0 0 2228 7
0 0 0 0 0 3 0 4 17 2352
2500
2000
1500
1000
500
0

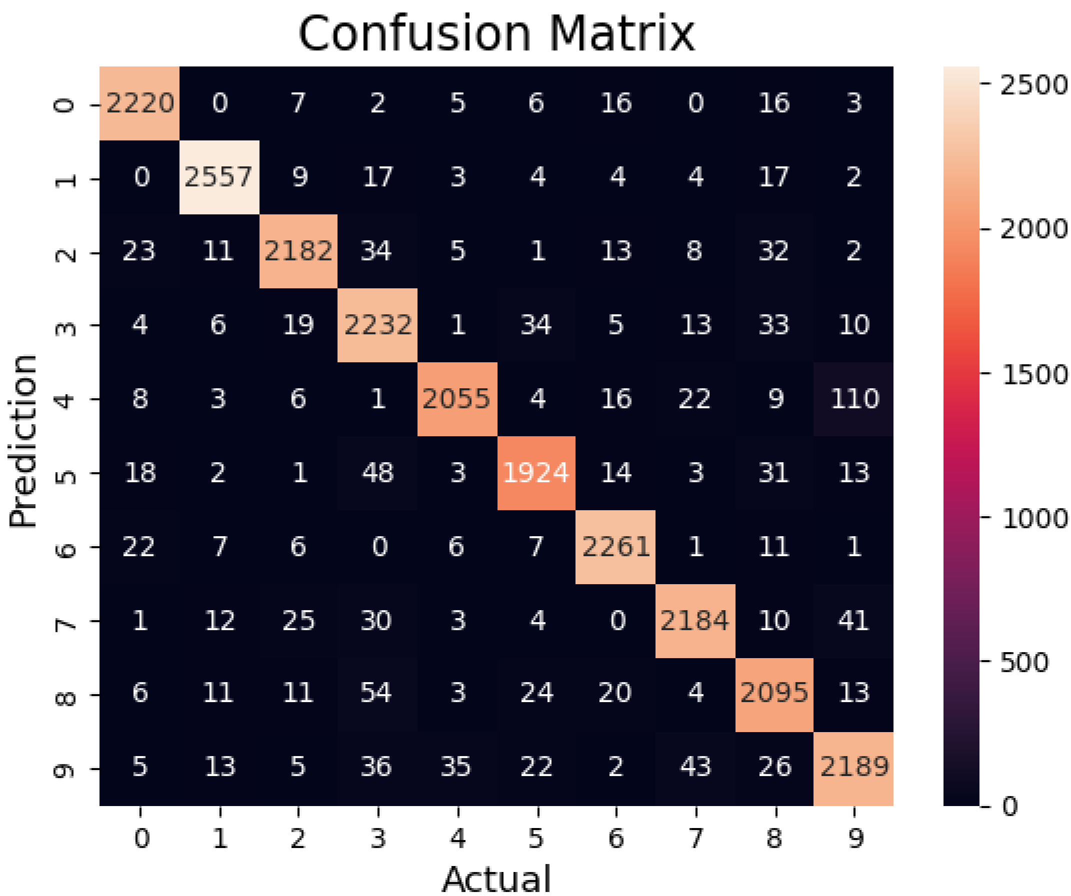
Model 8


Confusion Matrix
Prediction
Actual
2235 0 0 34 0 0 1 0 0 5
0 2593 4 17 0 0 0 1 2 0
0 0 2266 44 0 0 0 0 0 1
0 0 0 2351 0 3 0 0 1 2
0 0 1 5 2216 0 0 1 0 11
2 0 1 33 1 2017 0 0 1 2
3 5 0 18 3 10 2280 0 3 0
0 0 1 32 3 0 0 2266 1 7
0 1 2 51 0 1 0 0 2183 3
0 0 0 19 0 1 0 1 1 2354
0 1 2 3 4 5 6 7 8 9
2500
2000
1500
1000
500
0

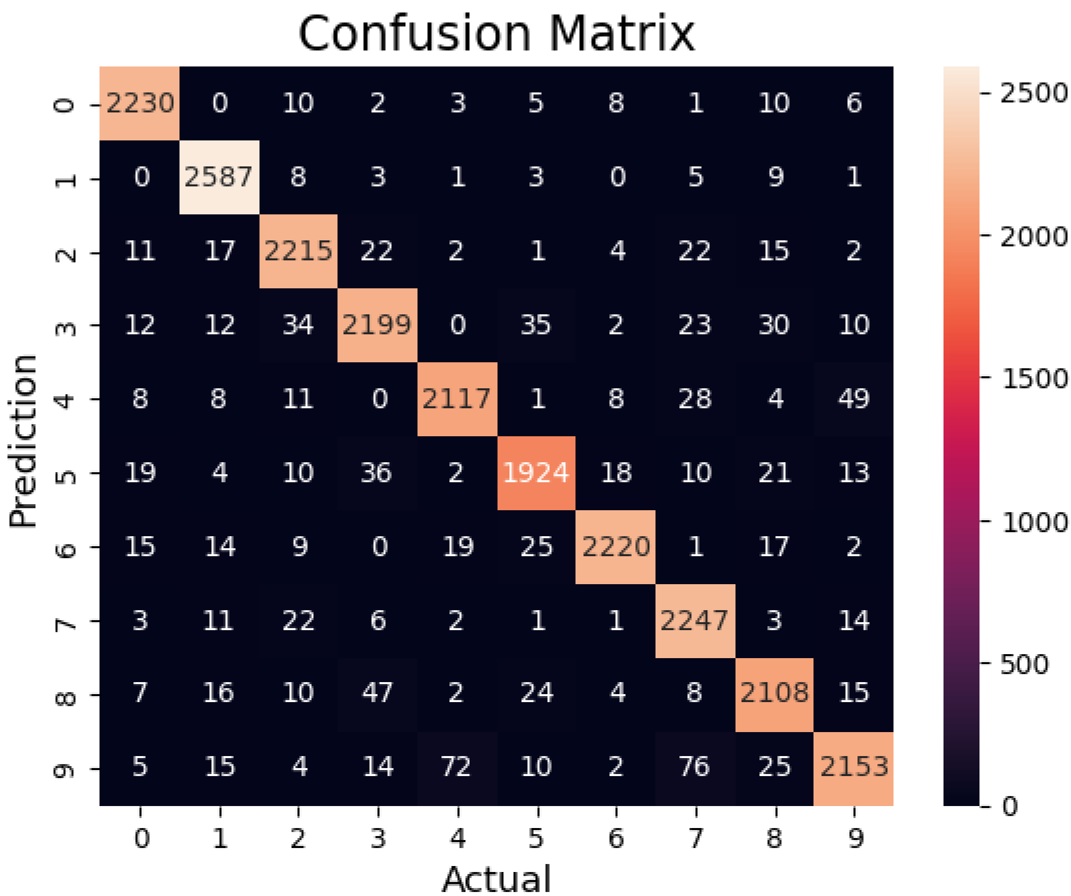
Model 9


Confusion Matrix
Prediction
Actual
0
1
2
3
4
5
6
7
8
9
2205 0 14 5 3 12 26 1 7 2
0 2577 8 7 3 0 3 6 9 4
1 7 2237 28 5 5 10 10 7 1
2 4 37 2208 3 39 3 19 30 12
4 3 10 0 2124 3 13 4 7 66
8 0 7 52 4 1932 21 1 15 17
10 3 11 1 8 20 2263 0 6 0
3 7 26 7 8 5 0 2210 11 33
11 18 22 60 7 42 17 5 2036 23
6 9 3 18 55 16 2 54 21 2192
2500
2000
1500
1000
500
0

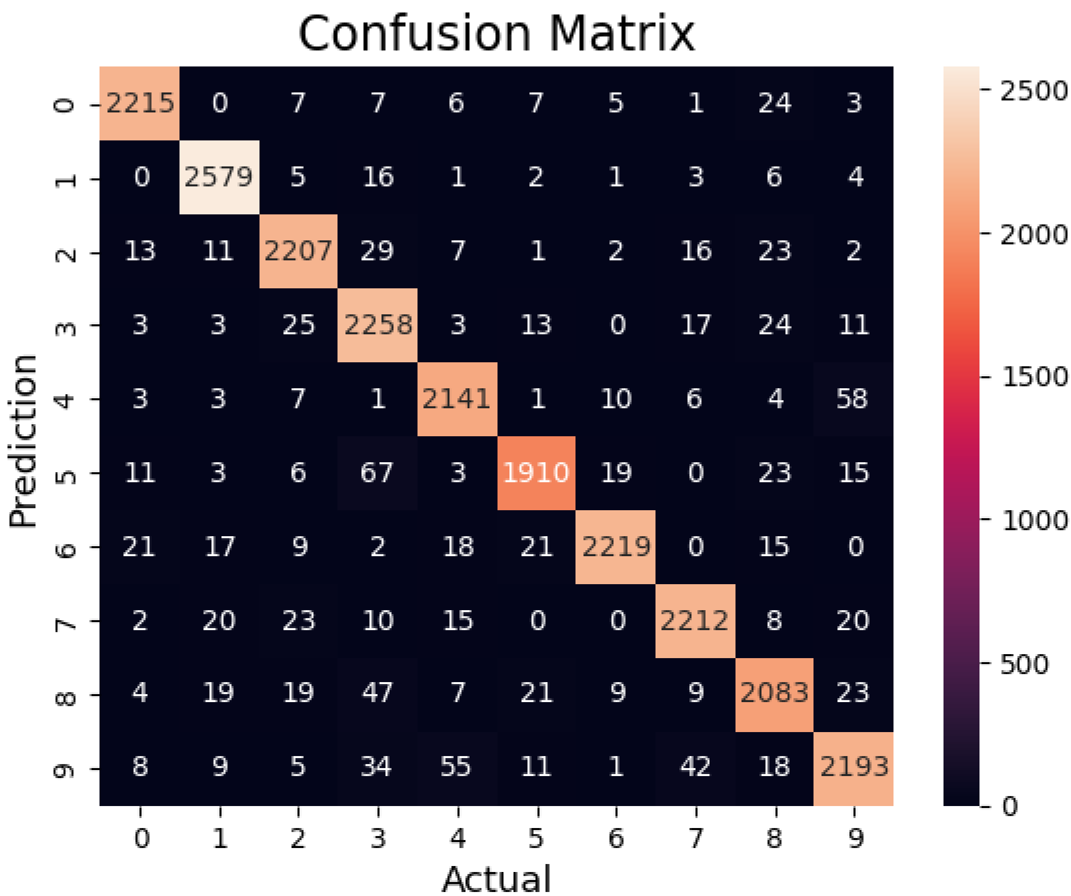
Model 10


Confusion Matrix
Prediction
Actual
0
1
2
3
4
5
6
7
8
9
2220 0 7 2 5 6 16 0 16 3
0 2557 9 17 3 4 4 4 17 2
23 11 2182 34 5 1 13 8 32 2
4 6 19 2232 1 34 5 13 33 10
8 3 6 1 2055 4 16 22 9 110
18 2 1 48 3 1924 14 3 31 13
22 7 6 0 6 7 2261 1 11 1
1 12 25 30 3 4 0 2184 10 41
6 11 11 54 3 24 20 4 2095 13
5 13 5 36 35 22 2 43 26 2189
2500
2000
1500
1000
500
0

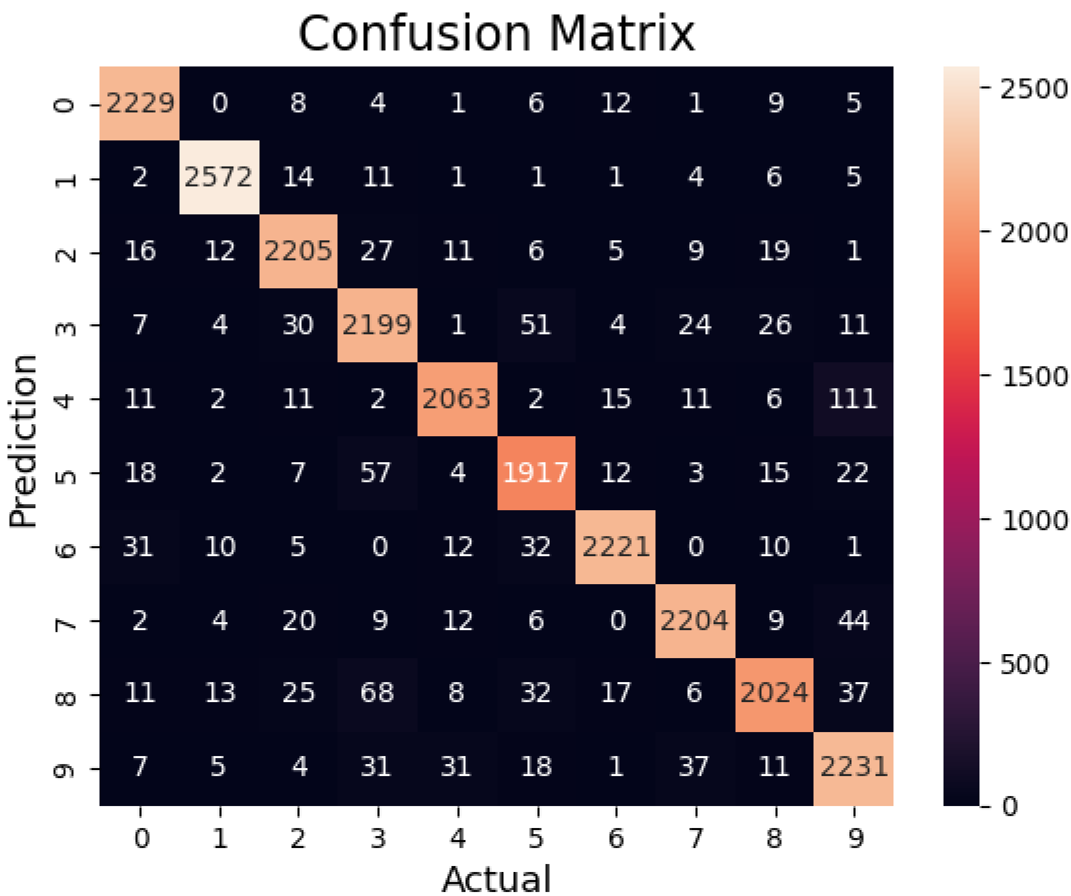
Model 11


Confusion Matrix
Prediction
Actual
2230
0
10
2
3
5
8
1
10
6
2587
8
3
1
9
11
17
2215
22
4
15
12
34
2199
35
23
30
2117
28
49
19
36
1924
18
21
13
14
25
2220
2247
47
24
2108
72
76
2153
2500
2000
1500
1000
500

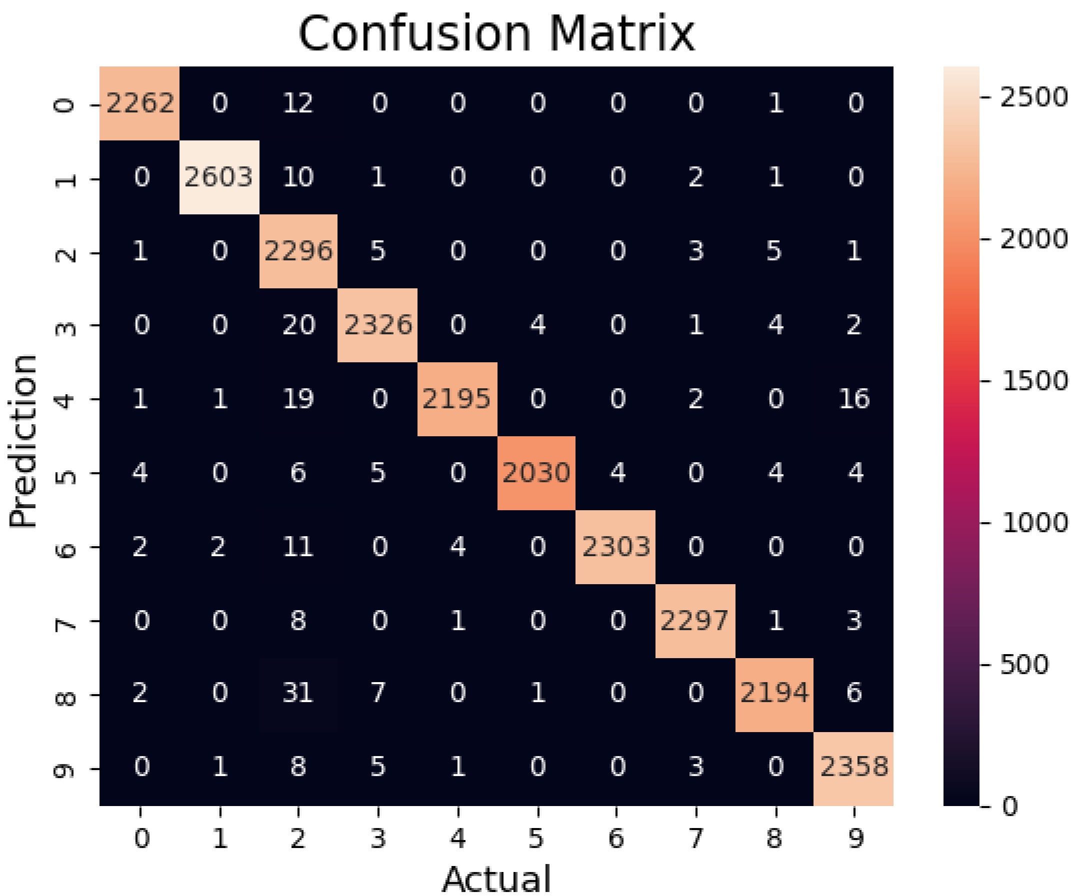
Model 12


Confusion Matrix
Prediction
Actual
2215 0 7 7 6 7 5 1 24 3
0 2579 5 16 1 2 1 3 6 4
13 11 2207 29 7 1 2 16 23 2
3 3 25 2258 3 13 0 17 24 11
3 3 7 1 2141 1 10 6 4 58
11 3 6 67 3 1910 19 0 23 15
21 17 9 2 18 21 2219 0 15 0
2 20 23 10 15 0 0 2212 8 20
4 19 19 47 7 21 9 9 2083 23
8 9 5 34 55 11 1 42 18 2193
0 1 2 3 4 5 6 7 8 9
2500
2000
1500
1000
500
0

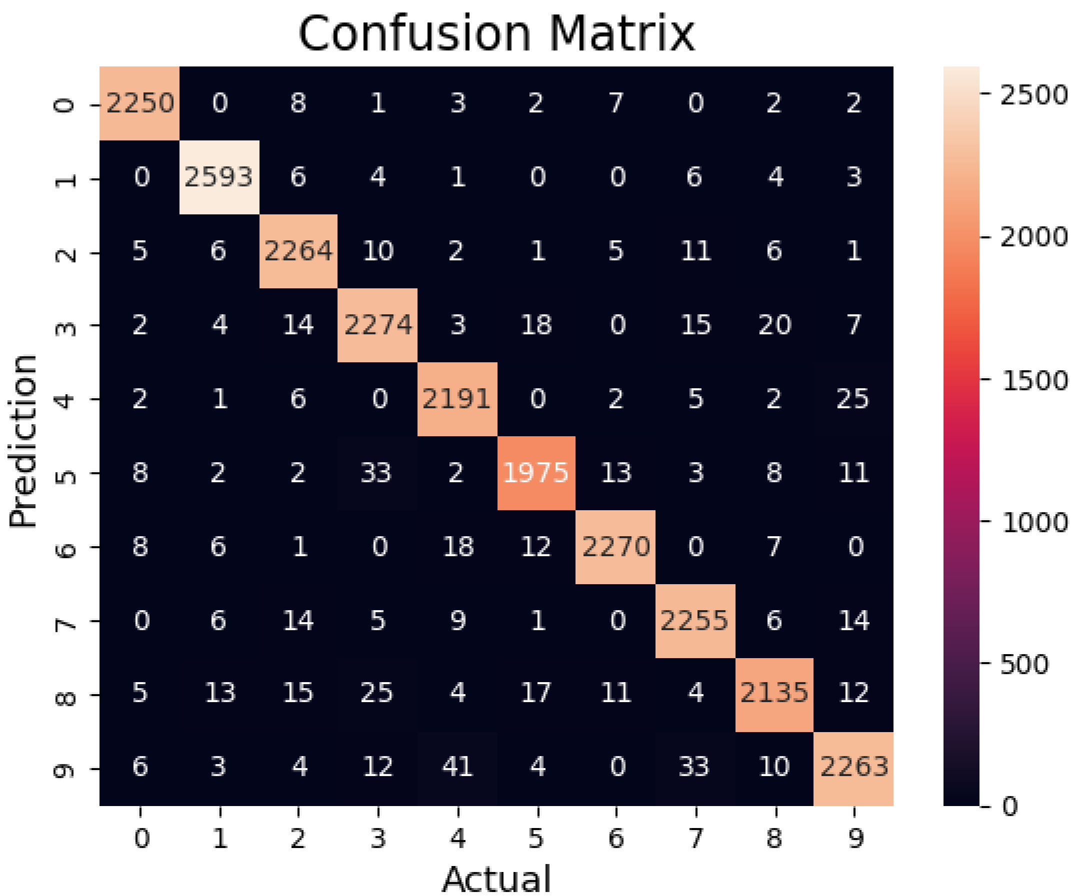
Model 13


Confusion Matrix
Prediction
Actual
0
1
2
3
4
5
6
7
8
9
2229 0 8 4 1 6 12 1 9 5
2 2572 14 11 1 1 1 4 6 5
16 12 2205 27 11 6 5 9 19 1
7 4 30 2199 1 51 4 24 26 11
11 2 11 2 2063 2 15 11 6 111
18 2 7 57 4 1917 12 3 15 22
31 10 5 0 12 32 2221 0 10 1
2 4 20 9 12 6 0 2204 9 44
11 13 25 68 8 32 17 6 2024 37
7 5 4 31 31 18 1 37 11 2231
2500
2000
1500
1000
500
0

Model 14


Confusion Matrix
Prediction
Actual
0
1
2
3
4
5
6
7
8
9
2262 0 12 0 0 0 0 0 1 0
0 2603 10 1 0 0 0 2 1 0
1 0 2296 5 0 0 0 3 5 1
0 0 20 2326 0 4 0 1 4 2
1 1 19 0 2195 0 0 2 0 16
4 0 6 5 0 2030 4 0 4 4
2 2 11 0 4 0 2303 0 0 0
0 0 8 0 1 0 0 2297 1 3
2 0 31 7 0 1 0 0 2194 6
0 1 8 5 1 0 0 3 0 2358
2500
2000
1500
1000
500
0

Model 15


Confusion Matrix
Prediction
Actual
0
1
2
3
4
5
6
7
8
9
2250 0 8 1 3 2 7 0 2 2
0 2593 6 4 1 0 0 6 4 3
5 6 2264 10 2 1 5 11 6 1
2 4 14 2274 3 18 0 15 20 7
2 1 6 0 2191 0 2 5 2 25
8 2 2 33 2 1975 13 3 8 11
8 6 1 0 18 12 2270 0 7 0
0 6 14 5 9 1 0 2255 6 14
5 13 15 25 4 17 11 4 2135 12
6 3 4 12 41 4 0 33 10 2263
2500
2000
1500
1000
500
0