Traditional Machine Learning Method vs Neural Networks (on the MNIST Handwritten Dataset)

Arish Izhar
Computer Science
Birla Institute of Technology Mesra, Ranchi
New Delhi, India

Abstract— **In this paper we compare the accuracies of solving the task of number identification of the MNIST handwritten dataset with two types of modelling approaches. On the one hand, we use well known traditional machine learning classification which is Logistic Regression in this case; and on the other hand we use Artificial Neural Networks to do the same. Traditional Machine Learning algorithms tend to perform at the same level when the data size increases but ANN outperforms traditional Machine Learning algorithms. Three different groups of models are trained. For the entire dataset, to detect '2' and to detect 'not 2'. In addition we add another hidden layer to the neural networks to offer more insight.**

I. INTRODUCTION

The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems. The database is also widely used for training and testing in the field of machine learning. It was created by "re-mixing" the samples from NIST's original datasets. The creators felt that since NIST's training dataset was taken from American Census Bureau employees, while the testing dataset was taken from American high school students, it was not well-suited for machine learning experiments. Furthermore, the black and white images from NIST were normalized to fit into a 28x28 pixel bounding box and anti-aliased, which introduced grayscale levels.

The MNIST database contains 60,000 training images and 10,000 testing images. Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of the training set and the other half of the test set were taken from NIST's testing dataset.

II. METHOD

A. Methodology Overview

The MNIST data is freely available online. It is split into 60,000 training images and 10,000 testing images. The data was flattened and shuffled before it was used for both the Logistic Regression classifier as well as the Neural network. The dataset was also shuffled before being used to train the two models. The code was written using Sklearn, Tensorflow and Keras in Python

In order to gain a better insight into the difference between the methods, three different sets of models were trained for the dataset. One set was a general model for the entire dataset, the second was to identify just the number '2' and the third was a model to identify 'not 2'. In addition, another neural network with a hidden layer was trained in each set.

The sets are as follows-

1. Set 1 -
   a. Logistic Regression classifier on the entire dataset
   b. Neural Network trained on the entire dataset
2. Set 2 -
   a. Logistic Regression classifier to find '2'
   b. Neural Network trained to identify '2'
3. Set 3 -
   a. Logistic Regression classifier to find 'not 2'
   b. Neural Network trained to find 'not 2'

Additionally, each model had K-fold validation done on it with three folds. To evaluate the performance of any machine learning model we need to test it on some unseen data. Based on the models' performance on unseen data we can say whether our model is Under-fitting/Over-fitting/Well generalized. This is the reason we have used K-fold cross-validation.

Once the models are trained we then find out their accuracy and find the confusion matrix along with our performance metrics such as precision, recall and f1 score. The same was done for the cross-validation score and cross-validation predictions.

B. Logistic Regression Model

The logistic regression classifier was trained using the 'lbfgs' solver and a tolerance of 0.1.

C. Neural Network Model

The neural networks were trained using the 'sigmoid' activation function, the 'adam' optimizer and the 'sparse_categorical_crossentropy' loss function and five epochs. They had 10 dense units/neurons.

The neural network with the hidden layer had an additional layer with 100 dense units using the 'relu' activation function.

III. RESULTS

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

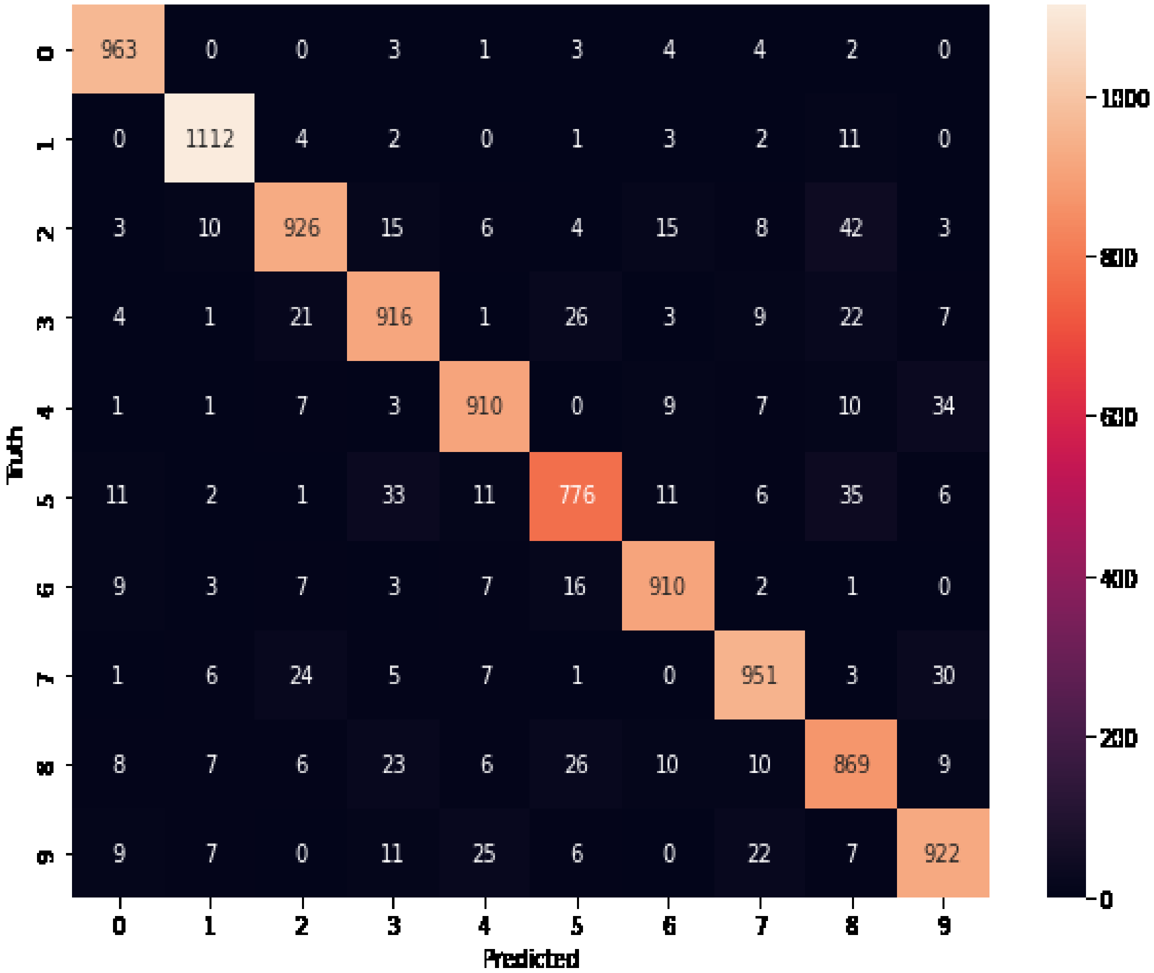
Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. Results for the entire dataset

Regression Model

Accuracy = 0.9255

Confusion Matrix



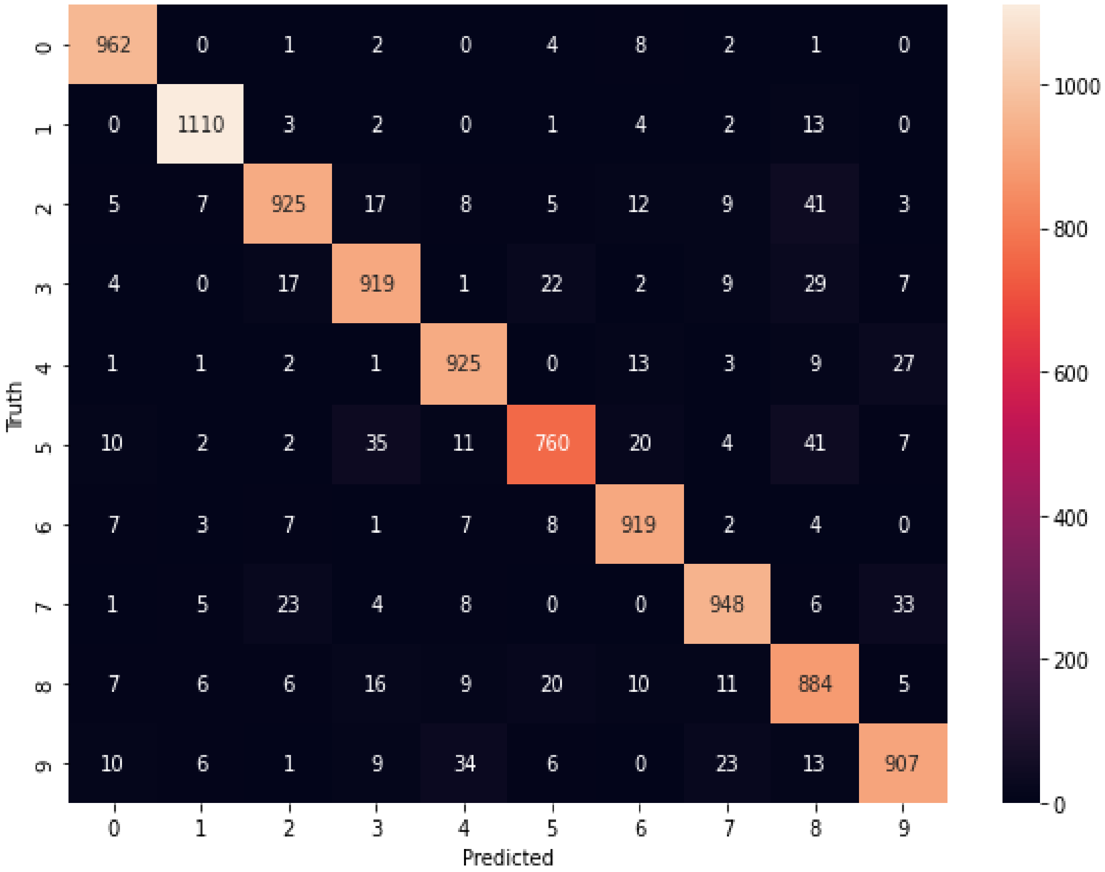
Truth \ Predicted	0	1	2	3	4	5	6	7	8	9
0	963	0	0	3	1	3	4	4	2	0
1	0	1112	4	2	0	1	3	2	11	0
2	3	10	926	15	6	4	15	8	42	3
3	4	1	21	916	1	26	3	9	22	7
4	1	1	7	3	910	0	9	7	10	34
5	11	2	1	33	11	776	11	6	35	6
6	9	3	7	3	7	16	910	2	1	0
7	1	6	24	5	7	1	0	951	3	30
8	8	7	6	23	6	26	10	10	869	9
9	9	7	0	11	25	6	0	22	7	922

Neural Network
Accuracy = 0.9258999824523926

Classification Report

```
              precision    recall  f1-score   support

           0       0.95      0.98      0.97       980
           1       0.97      0.98      0.97      1135
           2       0.93      0.90      0.91      1032
           3       0.90      0.91      0.91      1010
           4       0.93      0.93      0.93       982
           5       0.90      0.87      0.89       892
           6       0.94      0.95      0.95       958
           7       0.93      0.93      0.93      1028
           8       0.87      0.89      0.88       974
           9       0.91      0.91      0.91      1009

    accuracy                           0.93     10000
   macro avg       0.92      0.92      0.92     10000
weighted avg       0.93      0.93      0.93     10000
```

Confusion Matrix



	0	1	2	3	4	5	6	7	8	9
0	962	0	1	2	0	4	8	2	1	0
1	0	1110	3	2	0	1	4	2	13	0
2	5	7	925	17	8	5	12	9	41	3
3	4	0	17	919	1	22	2	9	29	7
4	1	1	2	1	925	0	13	3	9	27
5	10	2	2	35	11	760	20	4	41	7
6	7	3	7	1	7	8	919	2	4	0
7	1	5	23	4	8	0	0	948	6	33
8	7	6	6	16	9	20	10	11	884	5
9	10	6	1	9	34	6	0	23	13	907

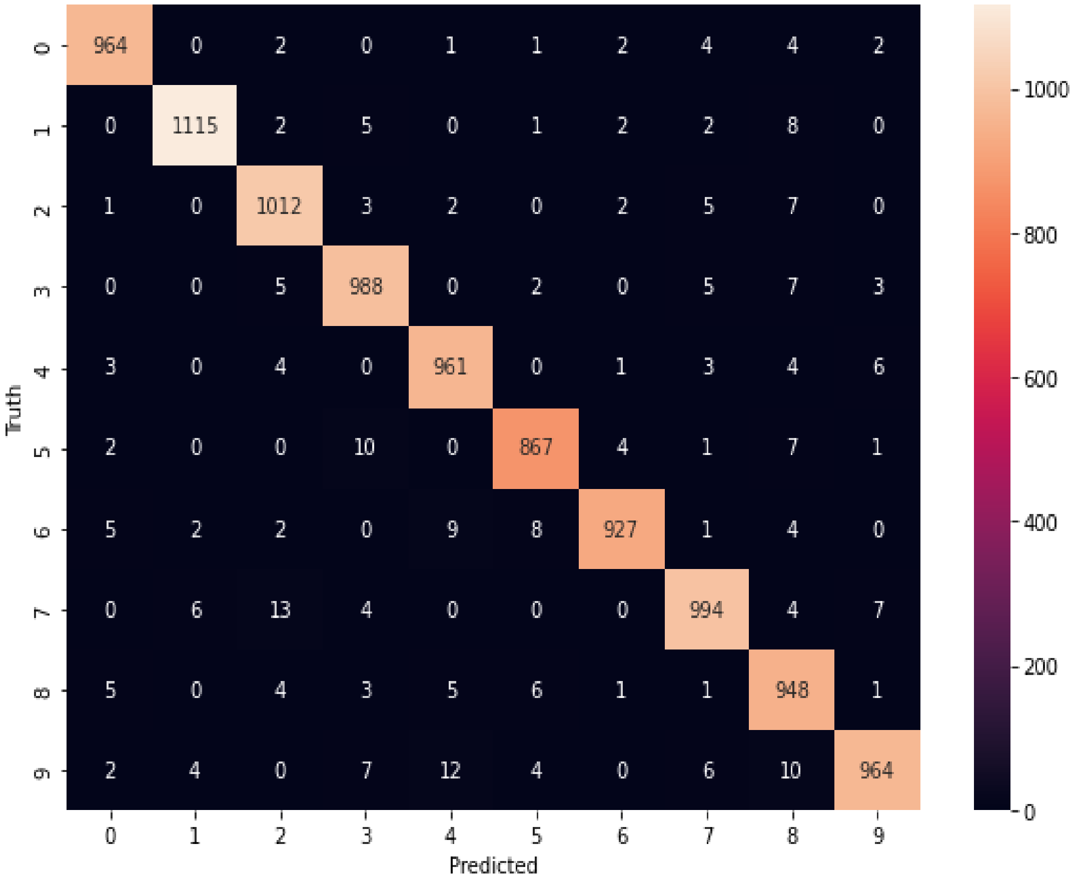
Classification Report

```
              precision    recall  f1-score   support

           0       0.96      0.98      0.97       980
           1       0.97      0.98      0.98      1135
           2       0.94      0.90      0.92      1032
           3       0.91      0.91      0.91      1010
           4       0.92      0.94      0.93       982
           5       0.92      0.85      0.88       892
           6       0.93      0.96      0.94       958
           7       0.94      0.92      0.93      1028
           8       0.85      0.91      0.88       974
           9       0.92      0.90      0.91      1009

    accuracy                           0.93     10000
   macro avg       0.93      0.92      0.92     10000
weighted avg       0.93      0.93      0.93     10000
```

Neural Network with a hidden layer

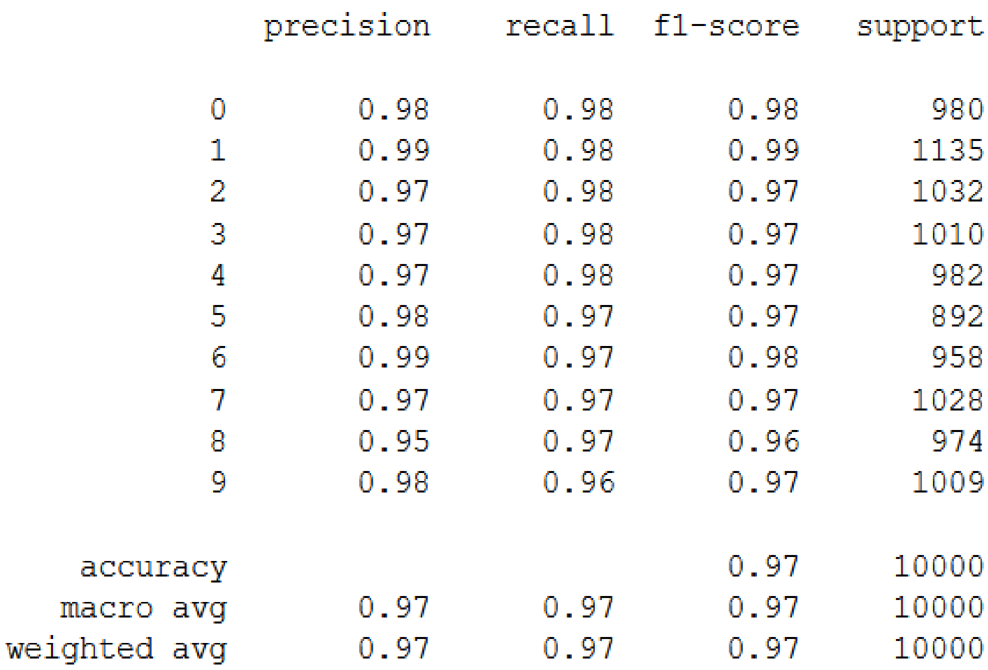
Accuracy score = 0.9739999771118164

Confusion Matrix



Truth \ Predicted	0	1	2	3	4	5	6	7	8	9
0	964	0	2	0	1	1	2	4	4	2
1	0	1115	2	5	0	1	2	2	8	0
2	1	0	1012	3	2	0	2	5	7	0
3	0	0	5	988	0	2	0	5	7	3
4	3	0	4	0	961	0	1	3	4	6
5	2	0	0	10	0	867	4	1	7	1
6	5	2	2	0	9	8	927	1	4	0
7	0	6	13	4	0	0	0	994	4	7
8	5	0	4	3	5	6	1	1	948	1
9	2	4	0	7	12	4	0	6	10	964

Classification Report

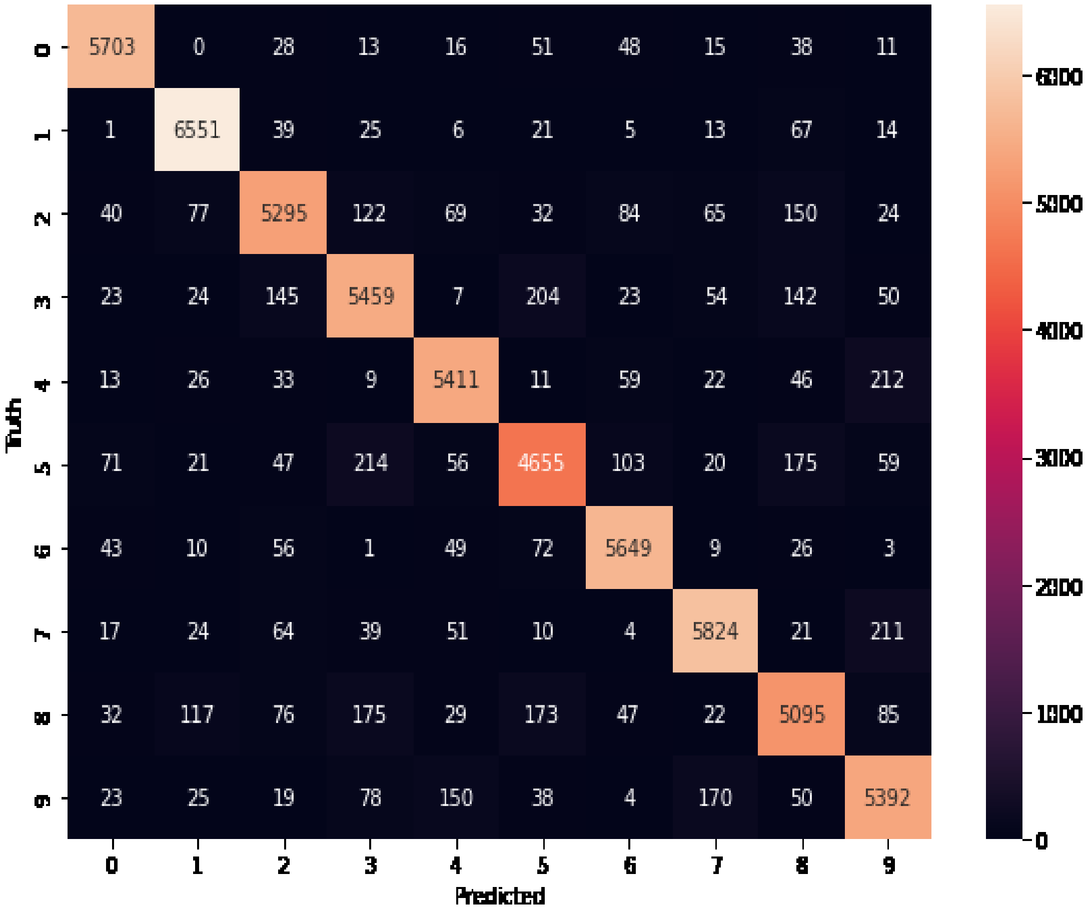


```
              precision    recall  f1-score   support

           0       0.98      0.98      0.98       980
           1       0.99      0.98      0.99      1135
           2       0.97      0.98      0.97      1032
           3       0.97      0.98      0.97      1010
           4       0.97      0.98      0.97       982
           5       0.98      0.97      0.97       892
           6       0.99      0.97      0.98       958
           7       0.97      0.97      0.97      1028
           8       0.95      0.97      0.96       974
           9       0.98      0.96      0.97      1009

    accuracy                           0.97     10000
   macro avg       0.97      0.97      0.97     10000
weighted avg       0.97      0.97      0.97     10000
```

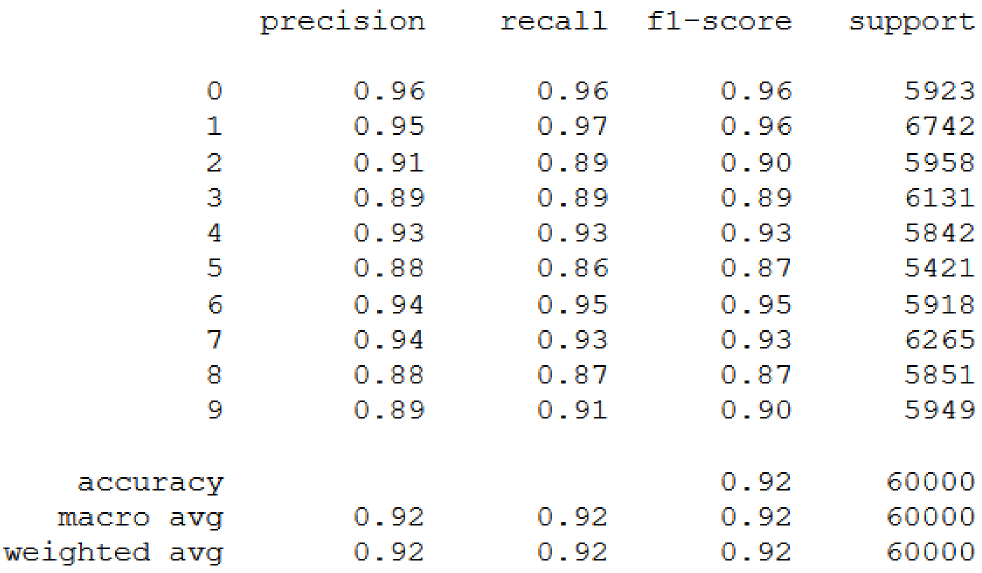
Regression Model Cross-Validation Scores

Accuracy = 0.9172333333333

Confusion Matrix



	0	1	2	3	4	5	6	7	8	9
0	5703	0	28	13	16	51	48	15	38	11
1	1	6551	39	25	6	21	5	13	67	14
2	40	77	5295	122	69	32	84	65	150	24
3	23	24	145	5459	7	204	23	54	142	50
4	13	26	33	9	5411	11	59	22	46	212
5	71	21	47	214	56	4655	103	20	175	59
6	43	10	56	1	49	72	5649	9	26	3
7	17	24	64	39	51	10	4	5824	21	211
8	32	117	76	175	29	173	47	22	5095	85
9	23	25	19	78	150	38	4	170	50	5392



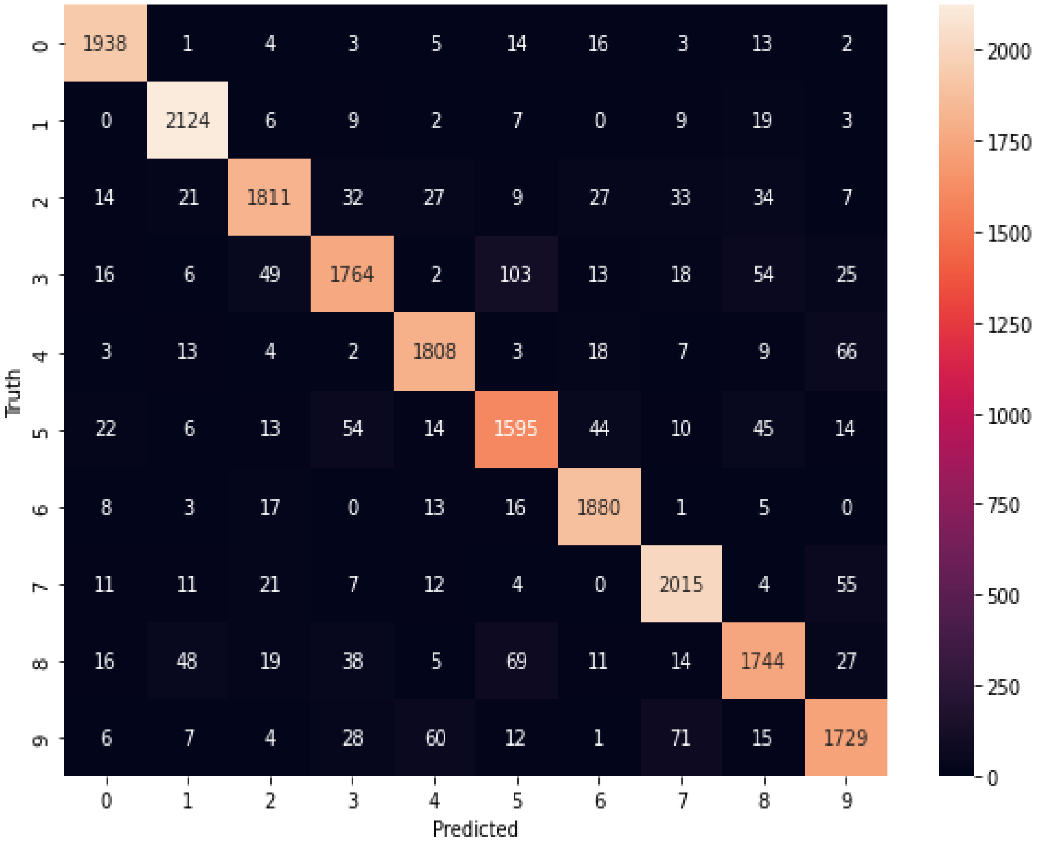
```
              precision    recall  f1-score   support

           0       0.96      0.96      0.96      5923
           1       0.95      0.97      0.96      6742
           2       0.91      0.89      0.90      5958
           3       0.89      0.89      0.89      6131
           4       0.93      0.93      0.93      5842
           5       0.88      0.86      0.87      5421
           6       0.94      0.95      0.95      5918
           7       0.94      0.93      0.93      6265
           8       0.88      0.87      0.87      5851
           9       0.89      0.91      0.90      5949

    accuracy                           0.92     60000
   macro avg       0.92      0.92      0.92     60000
weighted avg       0.92      0.92      0.92     60000
```

Classification Report

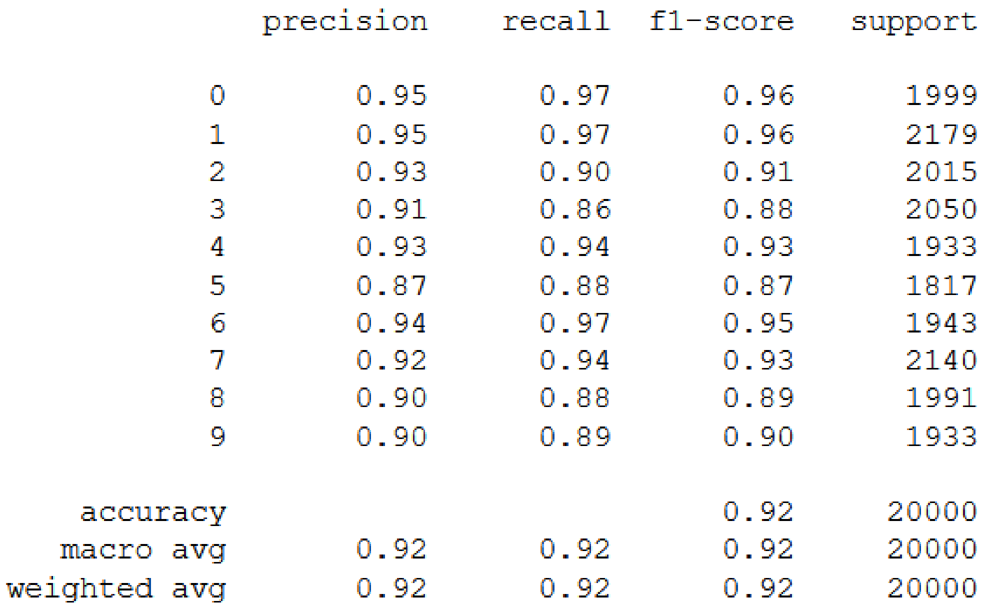
Neural Network Cross Validation Scores

Accuracy = 0.9191500147183737

Confusion Matrix



Truth \ Predicted	0	1	2	3	4	5	6	7	8	9
0	1938	1	4	3	5	14	16	3	13	2
1	0	2124	6	9	2	7	0	9	19	3
2	14	21	1811	32	27	9	27	33	34	7
3	16	6	49	1764	2	103	13	18	54	25
4	3	13	4	2	1808	3	18	7	9	66
5	22	6	13	54	14	1595	44	10	45	14
6	8	3	17	0	13	16	1880	1	5	0
7	11	11	21	7	12	4	0	2015	4	55
8	16	48	19	38	5	69	11	14	1744	27
9	6	7	4	28	60	12	1	71	15	1729



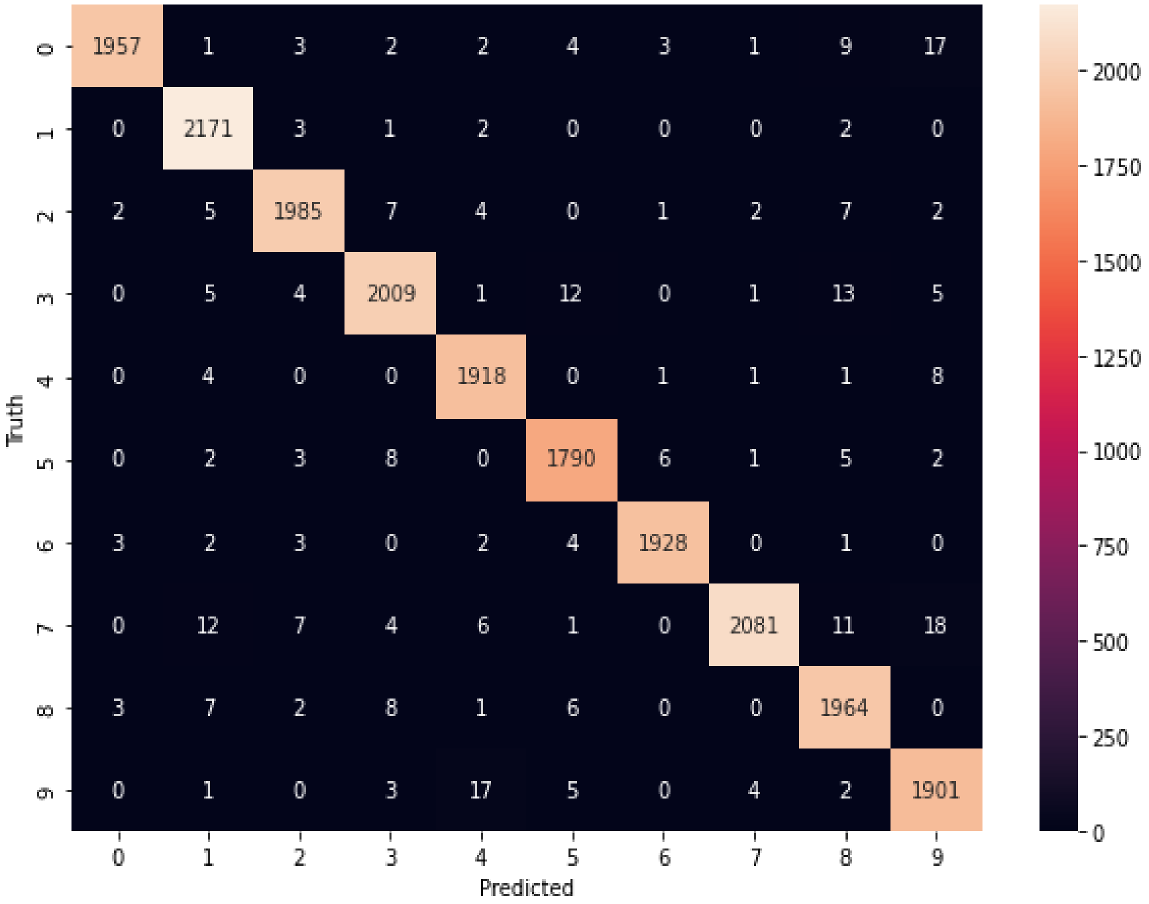
```
              precision    recall  f1-score   support

           0       0.95      0.97      0.96      1999
           1       0.95      0.97      0.96      2179
           2       0.93      0.90      0.91      2015
           3       0.91      0.86      0.88      2050
           4       0.93      0.94      0.93      1933
           5       0.87      0.88      0.87      1817
           6       0.94      0.97      0.95      1943
           7       0.92      0.94      0.93      2140
           8       0.90      0.88      0.89      1991
           9       0.90      0.89      0.90      1933

    accuracy                           0.92     20000
   macro avg       0.92      0.92      0.92     20000
weighted avg       0.92      0.92      0.92     20000
```

Classification Report

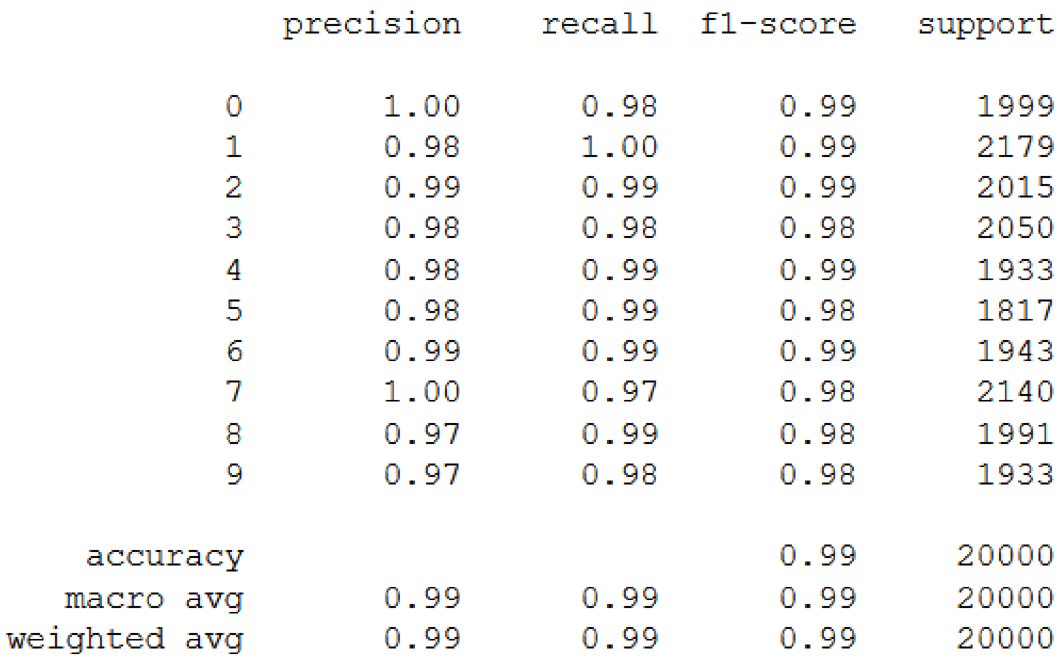
Neural Network with a hidden layer Cross-Validation Scores

Accuracy = 0.9242333372433981

Confusion Matrix



	0	1	2	3	4	5	6	7	8	9
0	1957	1	3	2	2	4	3	1	9	17
1	0	2171	3	1	2	0	0	0	2	0
2	2	5	1985	7	4	0	1	2	7	2
3	0	5	4	2009	1	12	0	1	13	5
4	0	4	0	0	1918	0	1	1	1	8
5	0	2	3	8	0	1790	6	1	5	2
6	3	2	3	0	2	4	1928	0	1	0
7	0	12	7	4	6	1	0	2081	11	18
8	3	7	2	8	1	6	0	0	1964	0
9	0	1	0	3	17	5	0	4	2	1901



```
              precision    recall  f1-score   support

           0       1.00      0.98      0.99      1999
           1       0.98      1.00      0.99      2179
           2       0.99      0.99      0.99      2015
           3       0.98      0.98      0.98      2050
           4       0.98      0.99      0.99      1933
           5       0.98      0.99      0.98      1817
           6       0.99      0.99      0.99      1943
           7       1.00      0.97      0.98      2140
           8       0.97      0.99      0.98      1991
           9       0.97      0.98      0.98      1933

    accuracy                           0.99     20000
   macro avg       0.99      0.99      0.99     20000
weighted avg       0.99      0.99      0.99     20000
```

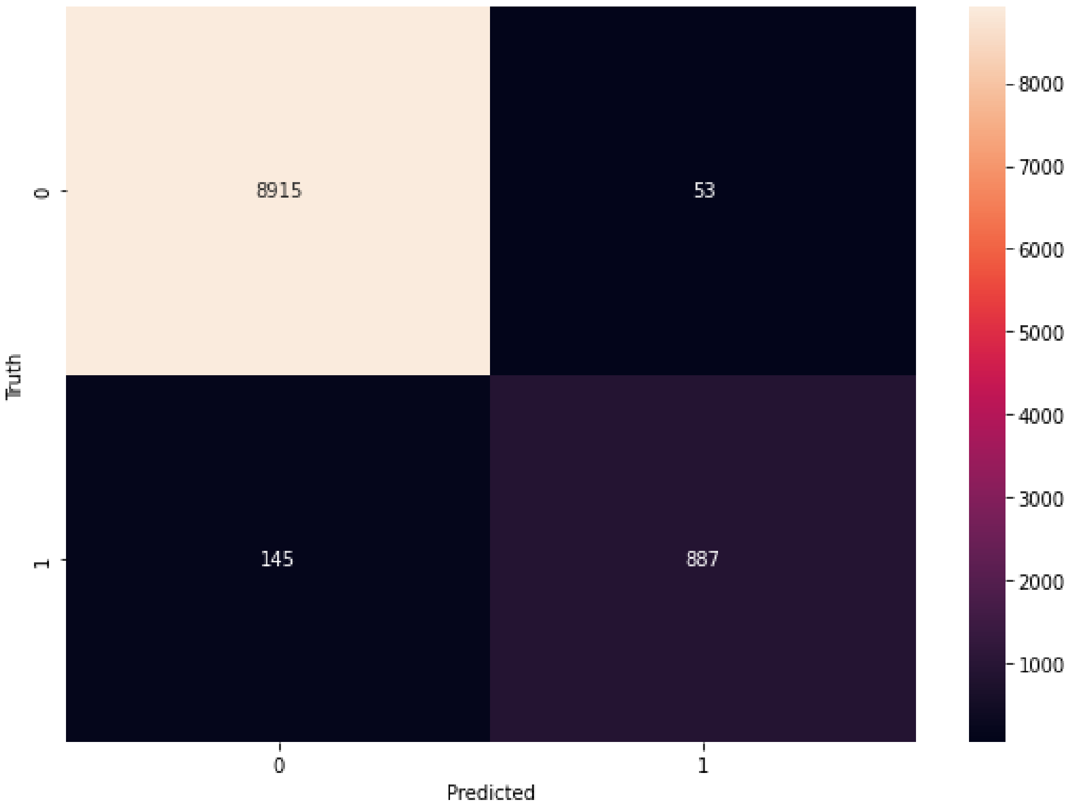
Classification Report

B. Results for the ‘2’ Detector

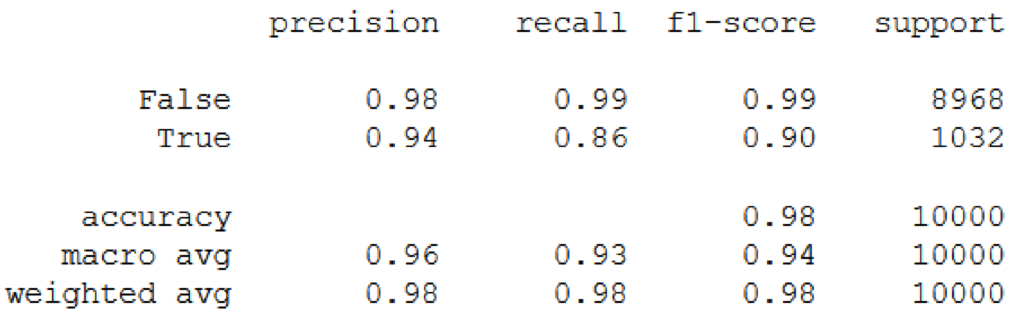
Regression Model

Accuracy = 0.9802

Confusion Matrix



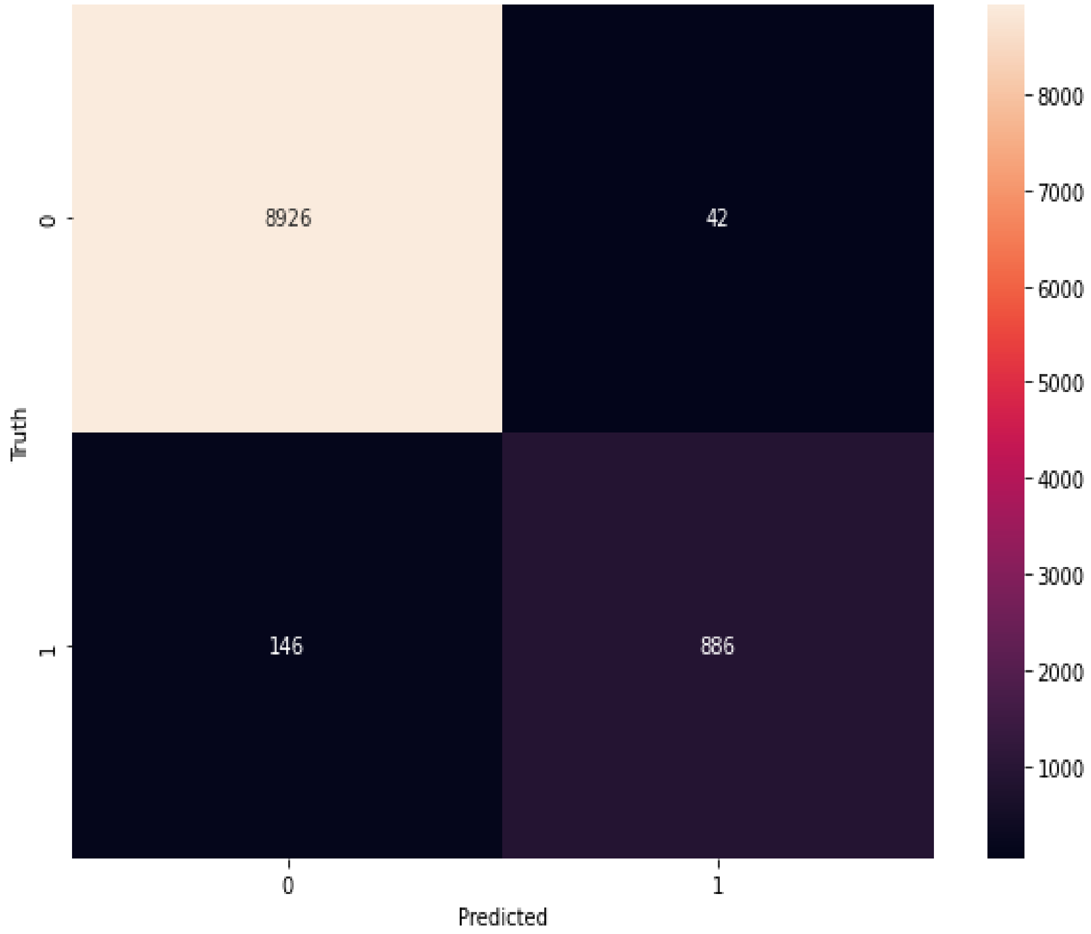
Classification Report



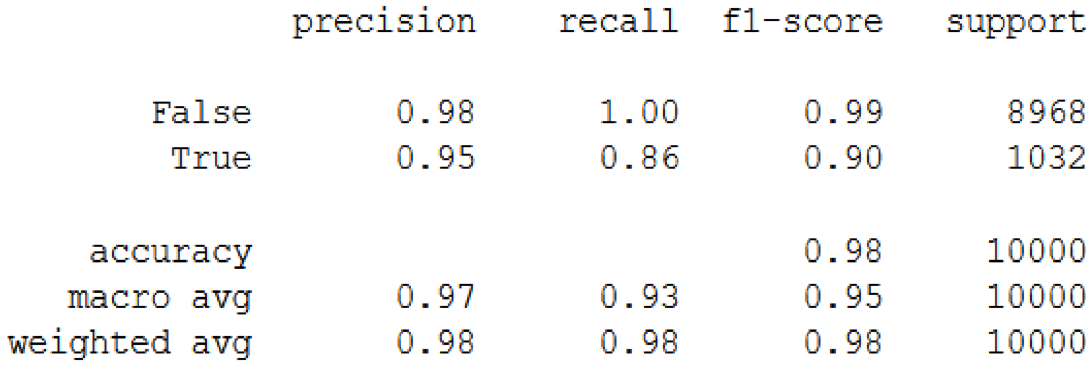
```
              precision    recall  f1-score   support

       False       0.98      0.99      0.99      8968
        True       0.94      0.86      0.90      1032

    accuracy                           0.98     10000
   macro avg       0.96      0.93      0.94     10000
weighted avg       0.98      0.98      0.98     10000
```

Neural Network Accuracy = 0.9811999797821045

Confusion Matrix



Classification Report



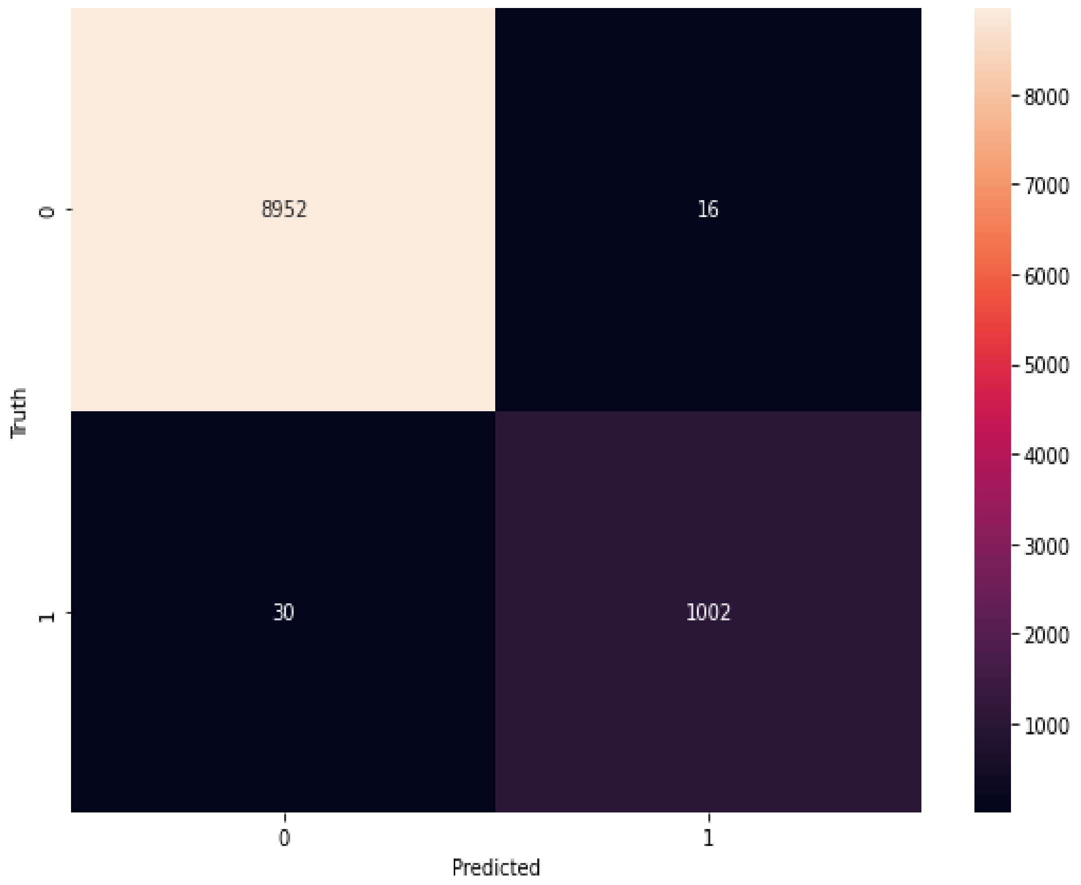
```
              precision    recall  f1-score   support

       False       0.98      1.00      0.99      8968
        True       0.95      0.86      0.90      1032

    accuracy                           0.98     10000
   macro avg       0.97      0.93      0.95     10000
weighted avg       0.98      0.98      0.98     10000
```

Neural Network with a hidden layer

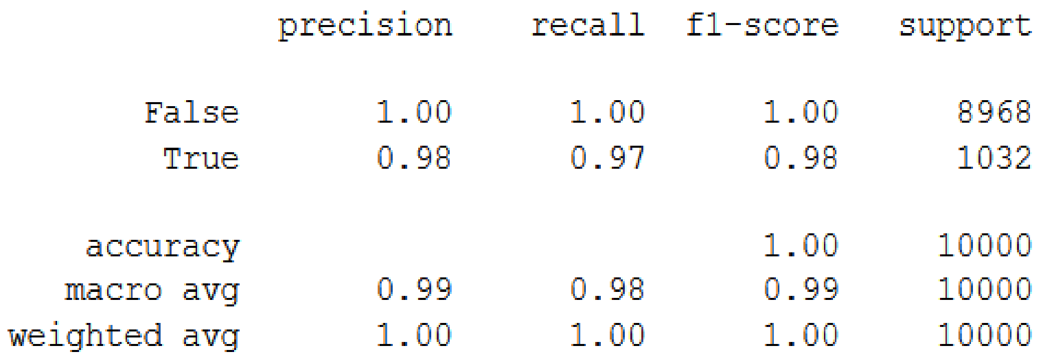
Accuracy =

0.9954000115394592

Confusion Matrix



Classification Report

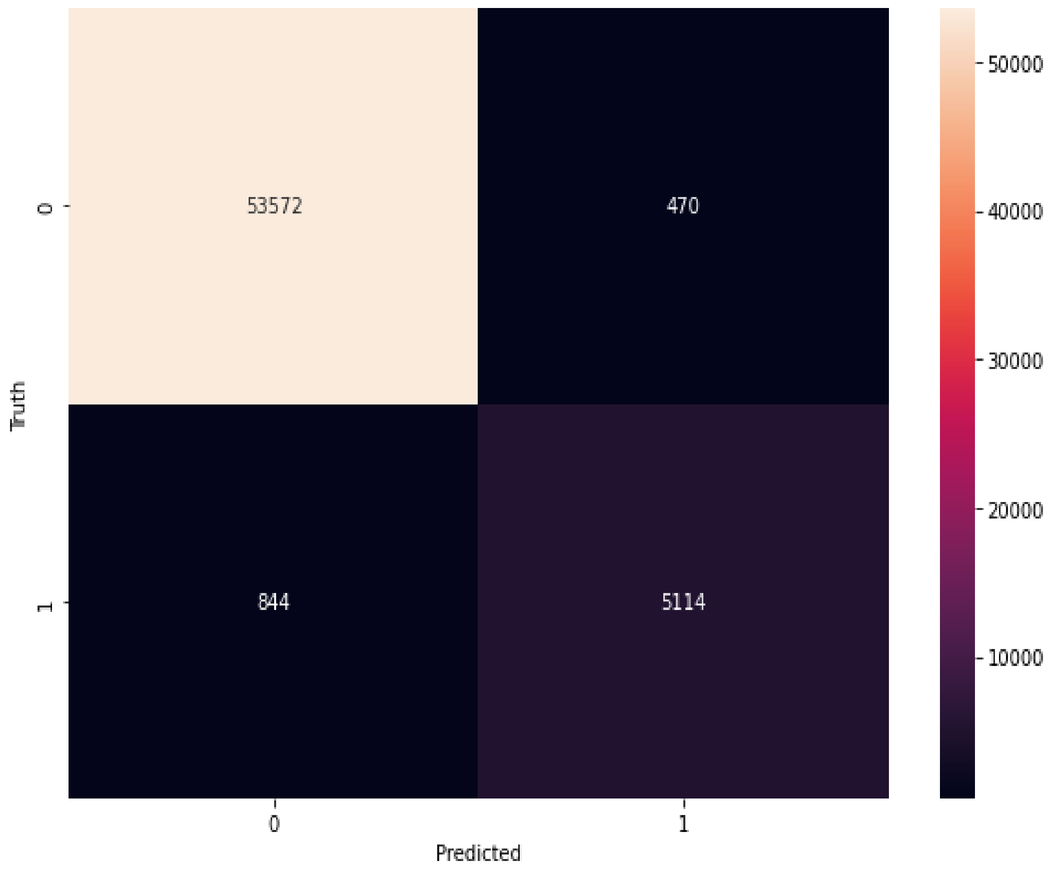


```
              precision    recall  f1-score   support

       False       1.00      1.00      1.00      8968
        True       0.98      0.97      0.98      1032

    accuracy                           1.00     10000
   macro avg       0.99      0.98      0.99     10000
weighted avg       1.00      1.00      1.00     10000
```

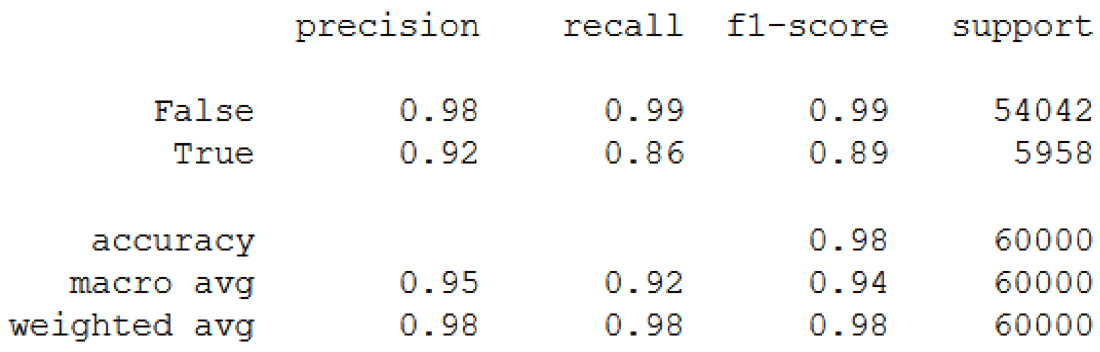
Regression Model Cross-Validation Scores

Accuracy = 0.9781

Confusion Matrix



Classification Report

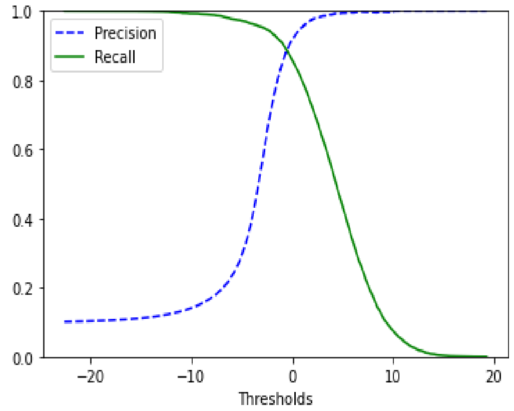


```
              precision    recall  f1-score   support

       False       0.98      0.99      0.99     54042
        True       0.92      0.86      0.89      5958

    accuracy                           0.98     60000
   macro avg       0.95      0.92      0.94     60000
weighted avg       0.98      0.98      0.98     60000
```

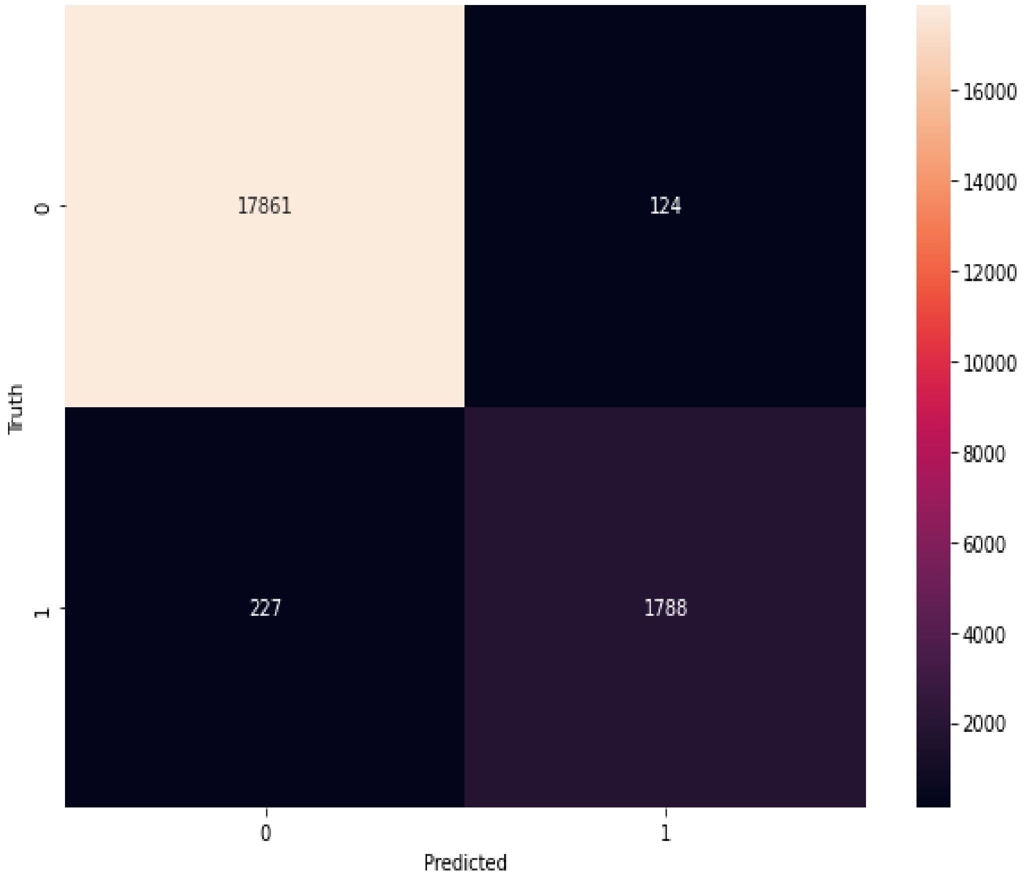
Neural Network Model Cross-Validation Scores

Accuracy = 0.9799000024795532

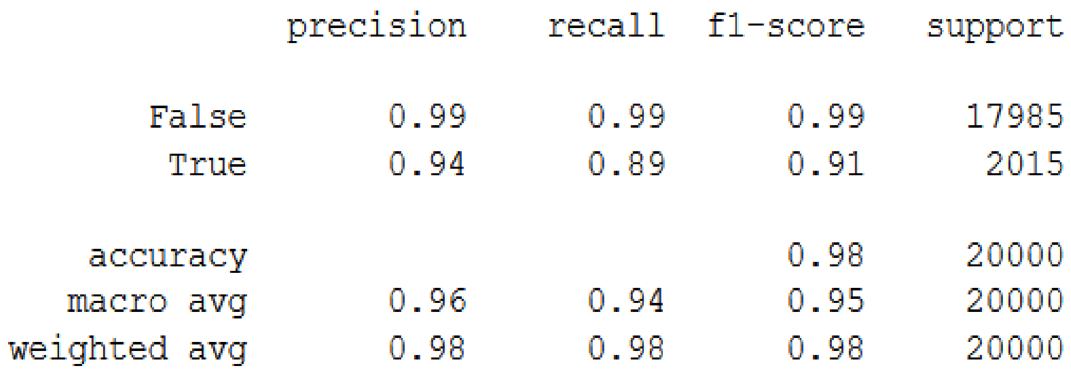
Precision Recall Curve



Confusion Matrix



Classification Report

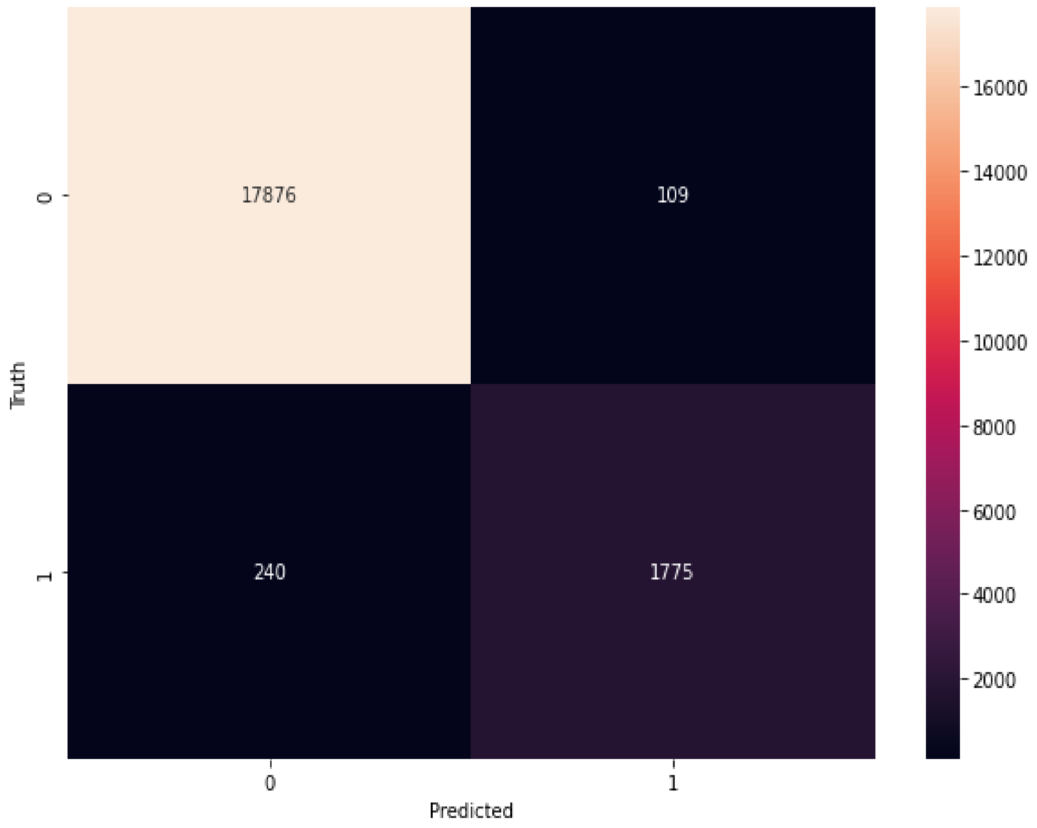


```
              precision    recall  f1-score   support

       False       0.99      0.99      0.99     17985
        True       0.94      0.89      0.91      2015

    accuracy                           0.98     20000
   macro avg       0.96      0.94      0.95     20000
weighted avg       0.98      0.98      0.98     20000
```

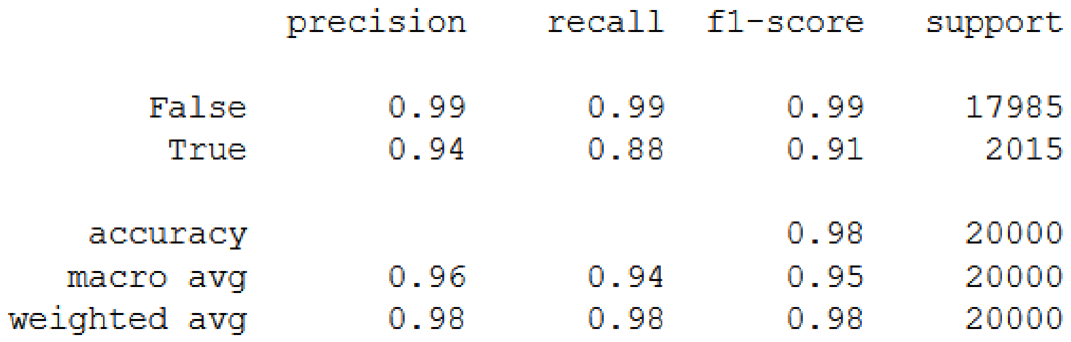
Neural Network with a hidden layer Model Cross-Validation Scores

Accuracy = 0.9799166719118754

Confusion Matrix



Classification Report

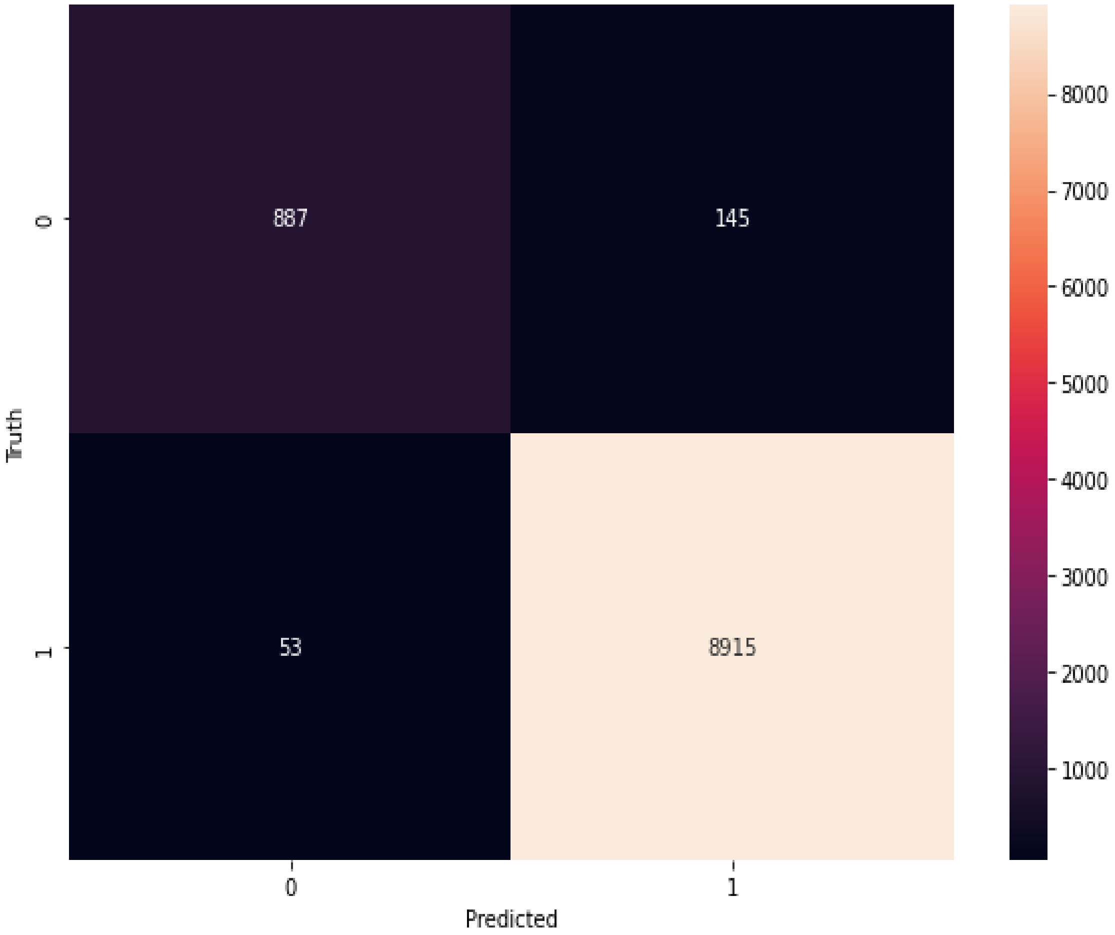

```
              precision    recall  f1-score   support

       False       0.99      0.99      0.99     17985
        True       0.94      0.88      0.91      2015

    accuracy                           0.98     20000
   macro avg       0.96      0.94      0.95     20000
weighted avg       0.98      0.98      0.98     20000
```

C. Results for the ‘not 2’ Detector

Regression Model

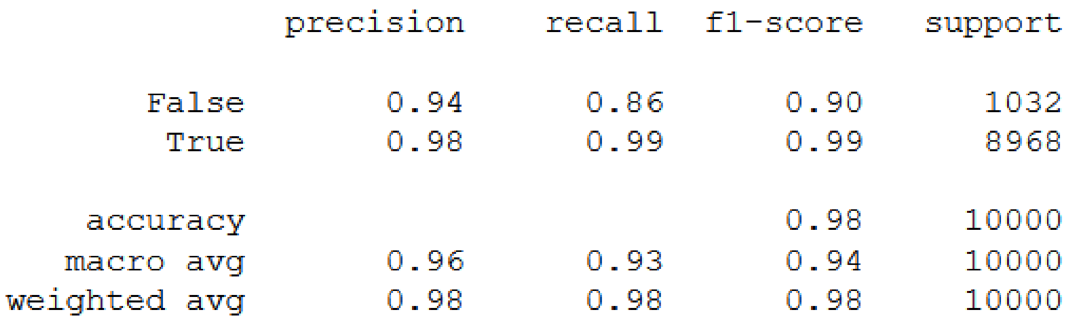
Accuracy = 0.9802

Confusion Matrix

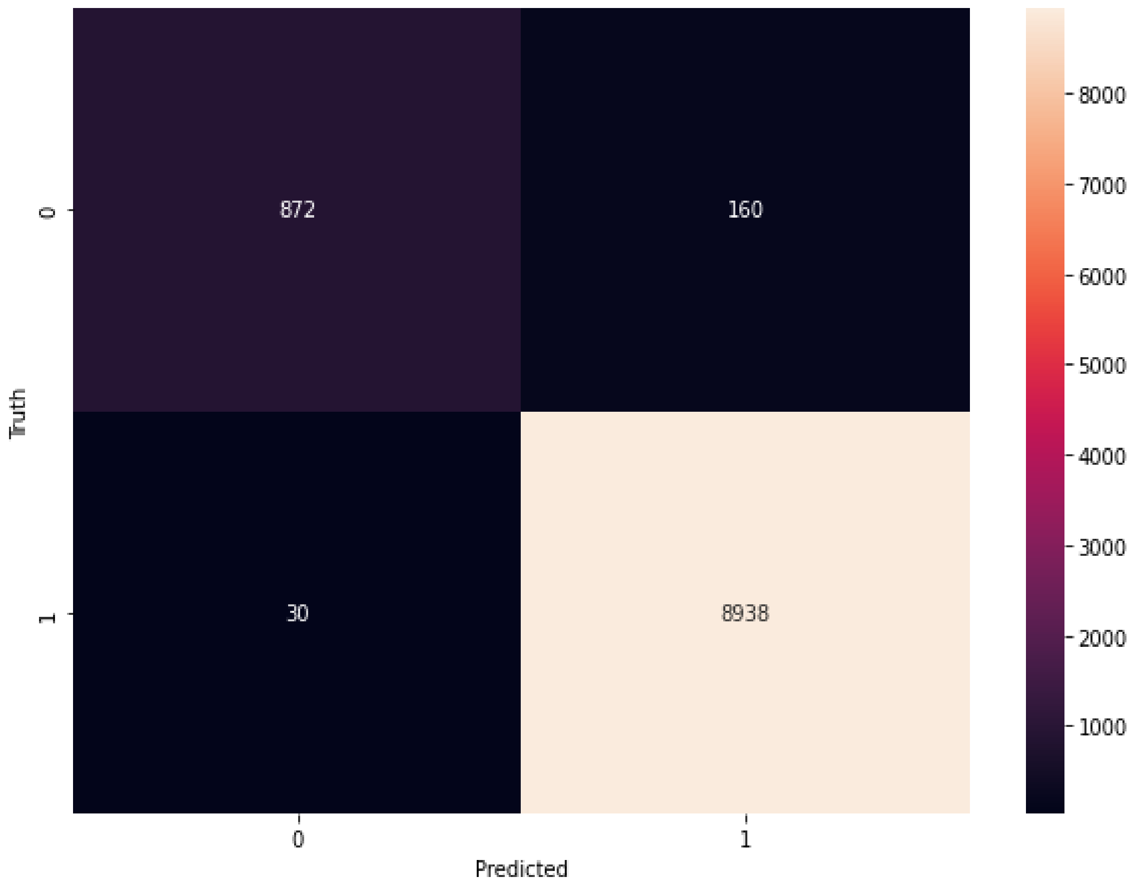


Neural Network Model
Accuracy = 0.9810000061988831

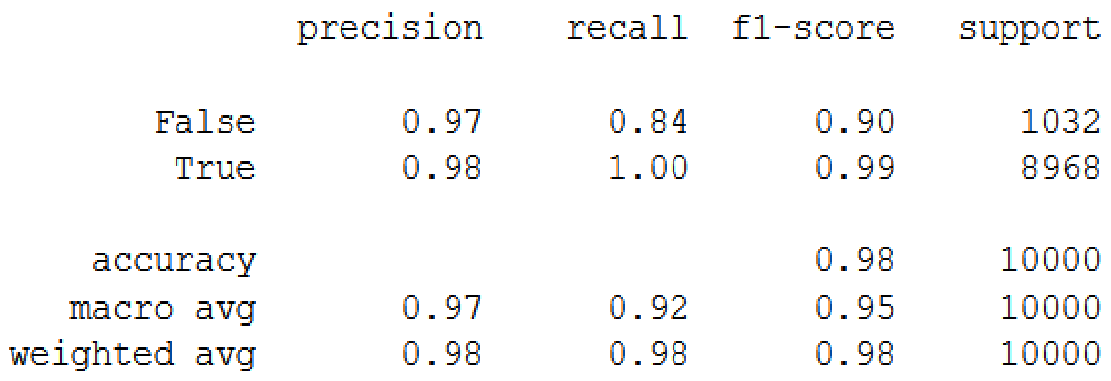
Classification Report


```
              precision    recall  f1-score   support

       False       0.94      0.86      0.90      1032
        True       0.98      0.99      0.99      8968

    accuracy                           0.98     10000
   macro avg       0.96      0.93      0.94     10000
weighted avg       0.98      0.98      0.98     10000
```

Confusion Matrix



Classification Report

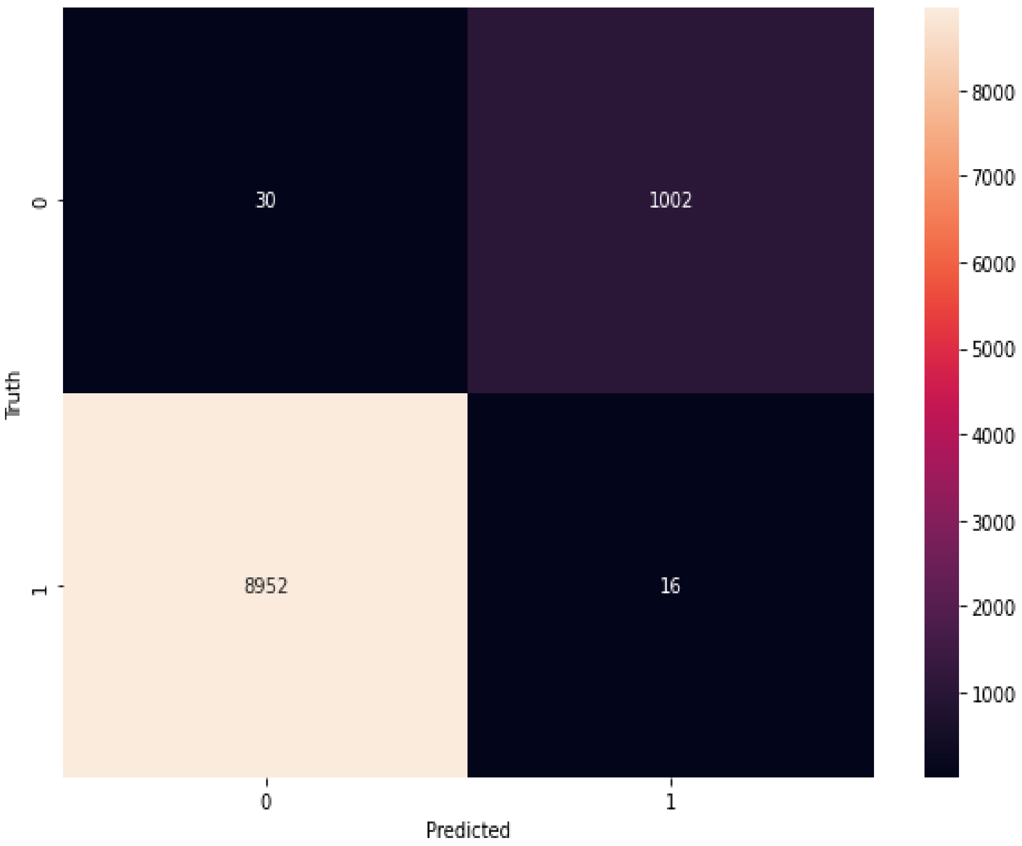


```
              precision    recall  f1-score   support

       False       0.97      0.84      0.90      1032
        True       0.98      1.00      0.99      8968

    accuracy                           0.98     10000
   macro avg       0.97      0.92      0.95     10000
weighted avg       0.98      0.98      0.98     10000
```

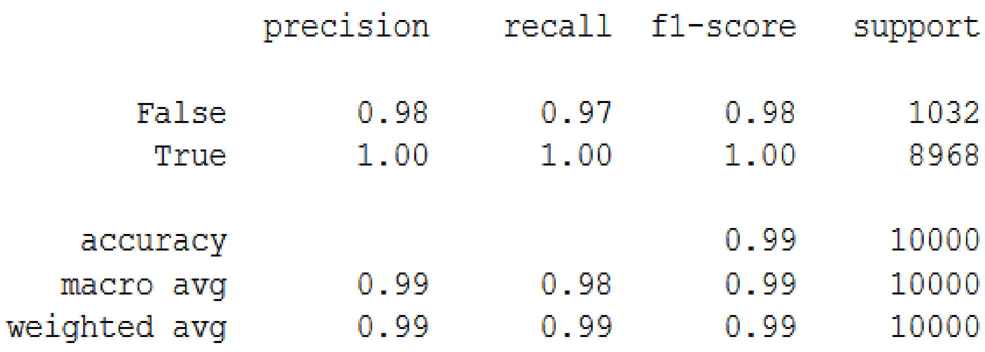
Neural Network with a hidden layer Model

Accuracy = 0.9948999881744385

Confusion Matrix



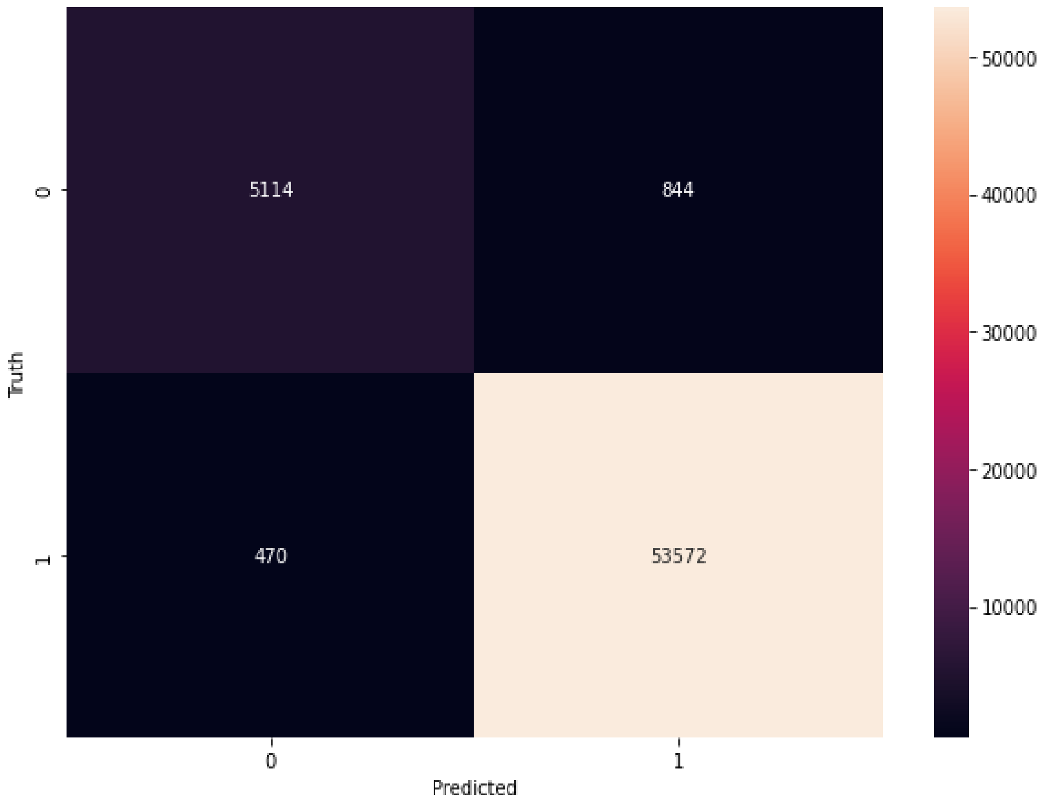
Classification Report



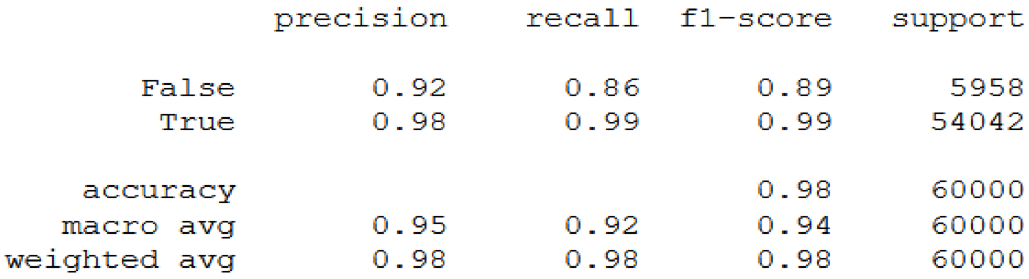
```
              precision    recall  f1-score   support

       False       0.98      0.97      0.98      1032
        True       1.00      1.00      1.00      8968

    accuracy                           0.99     10000
   macro avg       0.99      0.98      0.99     10000
weighted avg       0.99      0.99      0.99     10000
```

Regression Model Cross Validation Scores
Accuracy = 0.9781

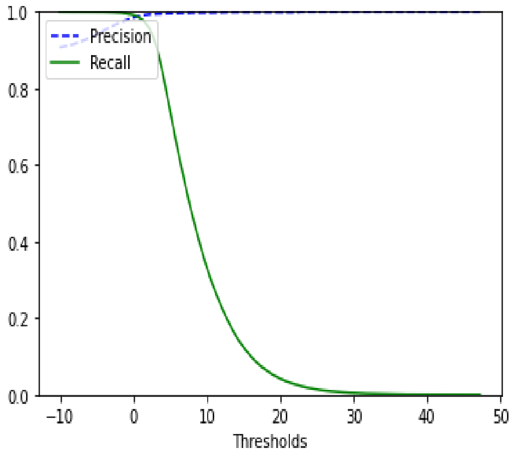
Confusion Matrix



Classification Report



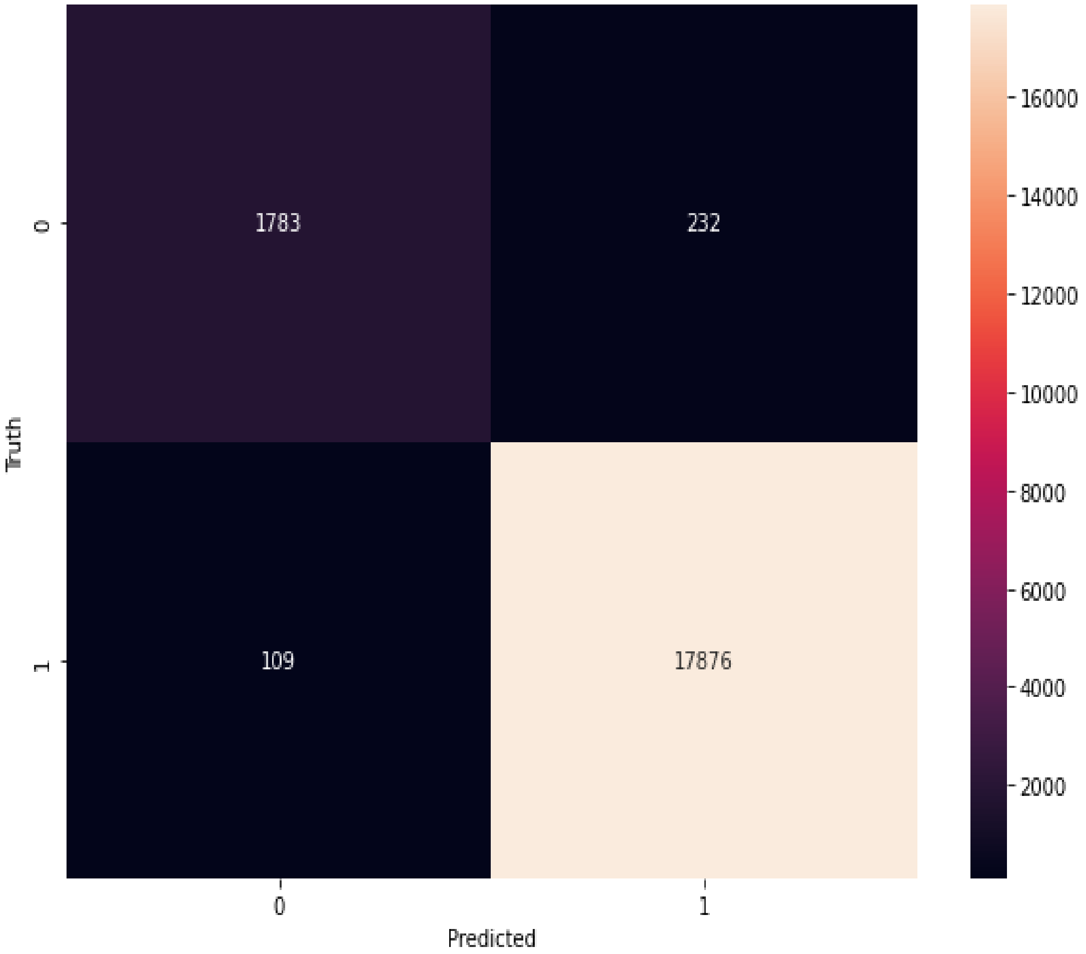
```
              precision    recall  f1-score   support

       False       0.92      0.86      0.89      5958
        True       0.98      0.99      0.99     54042

    accuracy                           0.98     60000
   macro avg       0.95      0.92      0.94     60000
weighted avg       0.98      0.98      0.98     60000
```



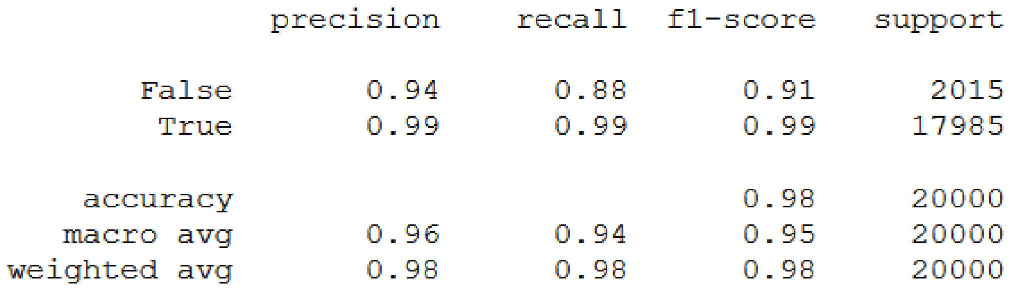
Precision-Recall Curve

Neural Network Model Cross Validation Scores
Accuracy = 0.9802666505177816

Confusion Matrix



Classification Report

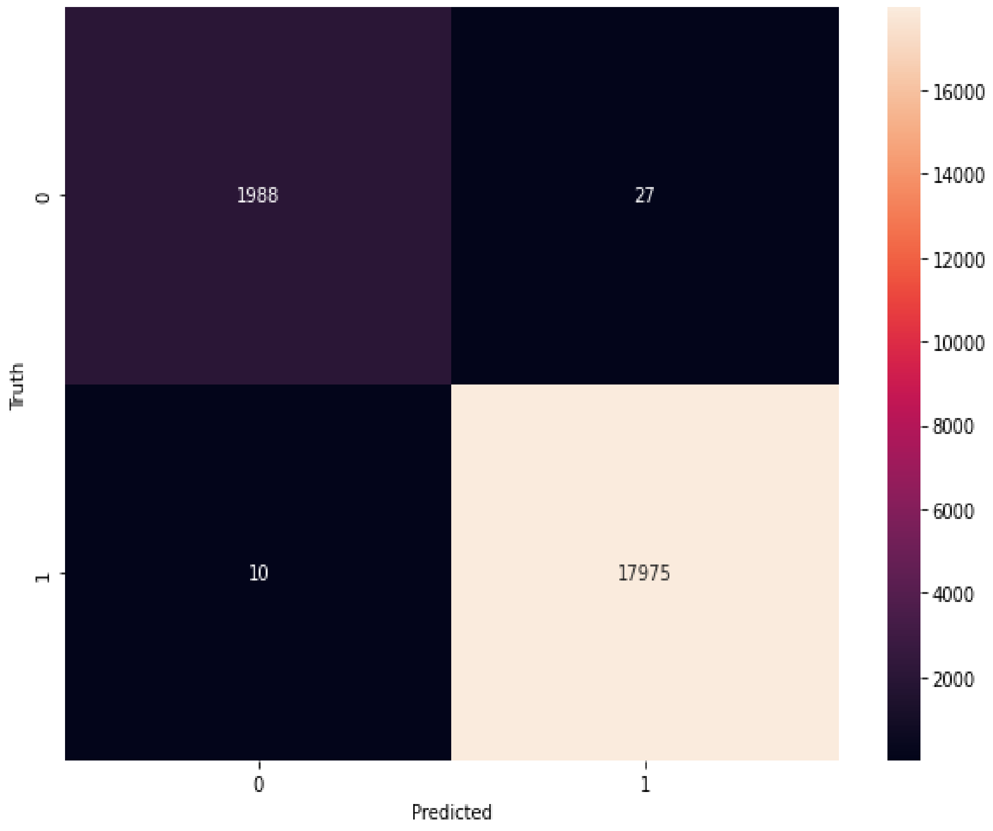


```
              precision    recall  f1-score   support

       False       0.94      0.88      0.91      2015
        True       0.99      0.99      0.99     17985

    accuracy                           0.98     20000
   macro avg       0.96      0.94      0.95     20000
weighted avg       0.98      0.98      0.98     20000
```

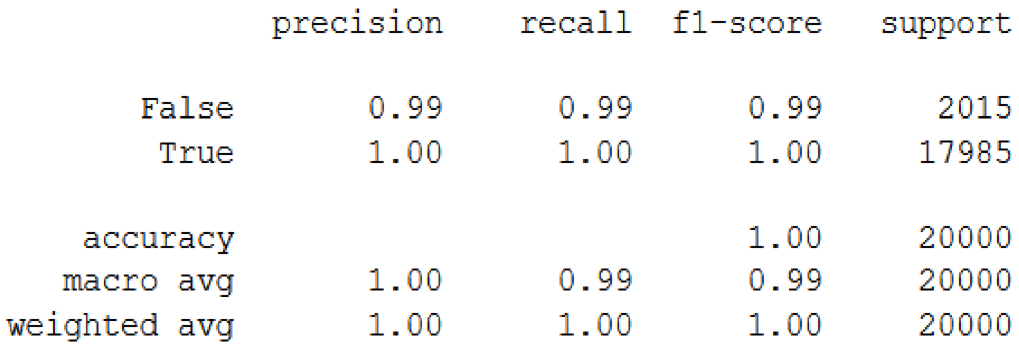
Neural Network with a hidden layer Model Cross Validation Scores

Accuracy = 0.9976666569709778

Confusion Matrix



Classification Report

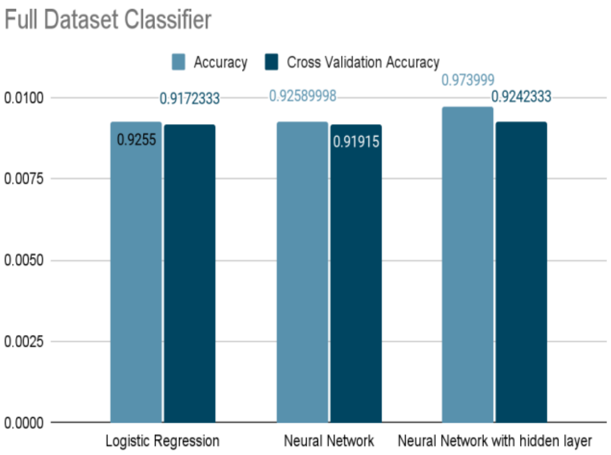


```
              precision    recall  f1-score   support

       False       0.99      0.99      0.99      2015
        True       1.00      1.00      1.00     17985

    accuracy                           1.00     20000
   macro avg       1.00      0.99      0.99     20000
weighted avg       1.00      1.00      1.00     20000
```

IV. CONCLUSION

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

A. *For the full dataset classifiers*



Looking at the accuracy scores

The neural network is 0.0432177% better than the regression classifier

The Neural network with a hidden layer is 5.2403% better than the regression classifier

The neural network with a hidden layer is 5.19484% better than the neural network with no hidden layers

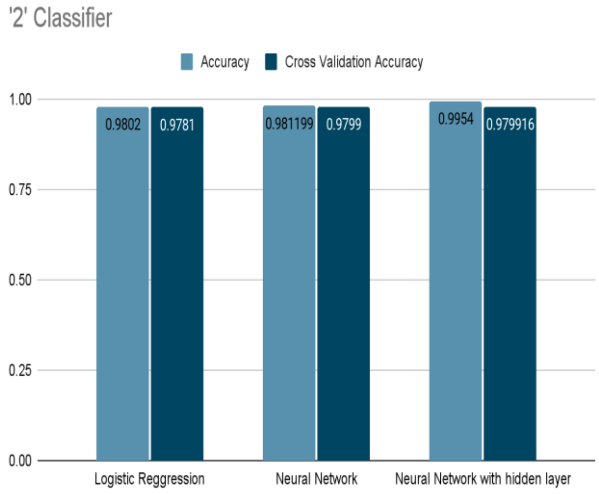
Now looking at the cross-validation accuracies

The neural network is 0.208965% better than the regression classifier

The Neural network with a hidden layer is 0.763165% better than the regression classifier

The neural network with a hidden layer is 0.553044% better than the neural network with no hidden layers

B. *For the detect '2' classifiers*



Looking at the accuracy scores

The neural network is 0.101918% better than the regression classifier

The Neural network with a hidden layer is 1.5507% better than the regression classifier

The neural network with a hidden layer is 1.44731% better than the neural network with no hidden layers

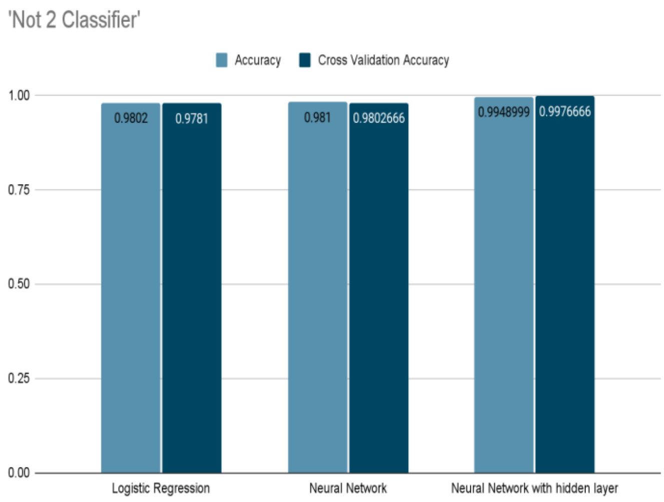
Now looking at the cross-validation accuracies

The neural network is 0.18403% better than the regression classifier

The Neural network with a hidden layer is 0.185666% better than the regression classifier

The neural network with a hidden layer is 0.00163282% better than the neural network with no hidden layers

C. *For the detect 'not 2' classifiers*



Looking at the accuracy scores

The neural network is 0.081616% better than the regression classifier

The Neural network with a hidden layer is 1.49968% better than the regression classifier

The neural network with a hidden layer is 1.41691% better than the neural network with no hidden layers

Now looking at the cross-validation accuracies

The neural network is 0.221511% better than the regression classifier

The Neural network with a hidden layer is 2.00047% better than the regression classifier

The neural network with a hidden layer is 1.77503% better than the neural network with no hidden layers

In each case, we can see that the Neural network with a single hidden layer seems to be better than the regression classifier while the neural network without any hidden layer is also better but only to a very small degree.