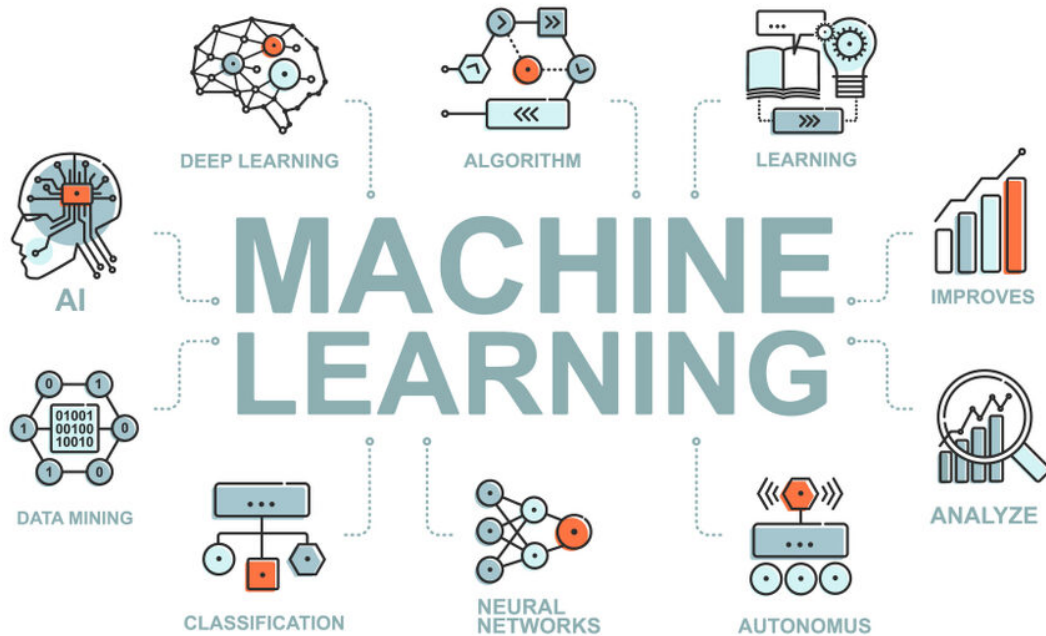


Assignment 2

MACHINE LEARNING



Team Members

- Divyateja Pasupuleti 2021A7PS0075H
- Kumarasamy 2021A7PS0096H
- Adarsh Das 2021A7PS1511H

Introduction

In the second assignment, we were tasked with creating 2 models, one from scratch, where we had to implement Naive Bayes' and the second being Artificial Neural Network using Pytorch.

Part A

Data preprocessing

We first replace all the '?' with the most frequently occurring value in that feature, then we convert the categorical data to nominal data, for example, the salaries are converted from >50K and <=50K to 0 and 1.

Model 1: Naive Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem, which assumes that the features are conditionally independent given the class label. It calculates the probability of each class and the conditional probability of each feature given the class. Then it uses these probabilities to predict the class with the highest probability for a new data point. It is simple, fast, and works well with high-dimensional data.

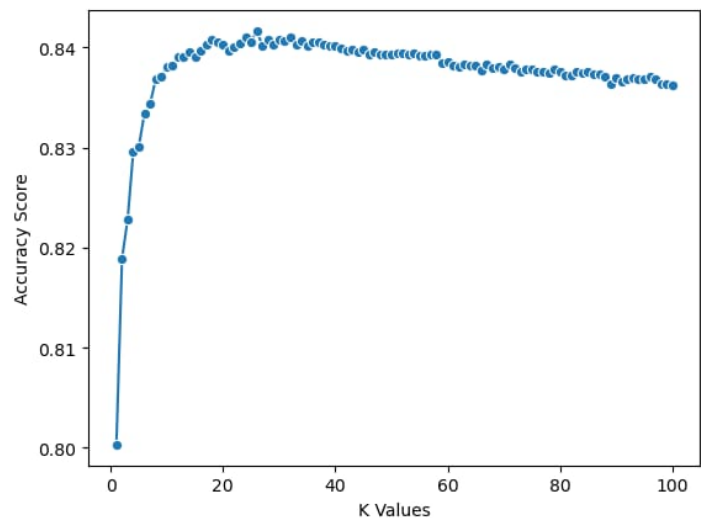
In Naive Bayes's Classifier, we assume that the features are all independent. Therefore, we can multiply the probabilities of the various features.

We then calculate the likelihood of each value of each feature for both salary >50K and <= 50K. We then use these likelihood values to predict the salary of the testing examples. We've split the data into training(67%) and testing(33%).

Feature	Statistic
Accuracy	87.66%
Precision	70.76%
Recall	83.10%
F1	76.44%

Model 2: KNN

KNN is a non-parametric, lazy learning algorithm used for classification and regression tasks. Given a new data point, KNN finds the K training examples closest to it in the feature space, and predicts the class label based on the majority vote of the K neighbors. The choice of K affects the classifier's performance, with smaller values of K leading to more flexible models and larger values leading to smoother decision boundaries.



For KNN we did an analysis of the data to get the best 'K' value and took the model with the same.

Feature	Statistic
Accuracy	83.76%
Precision	71.57%
Recall	52.73%
F1	60.72%

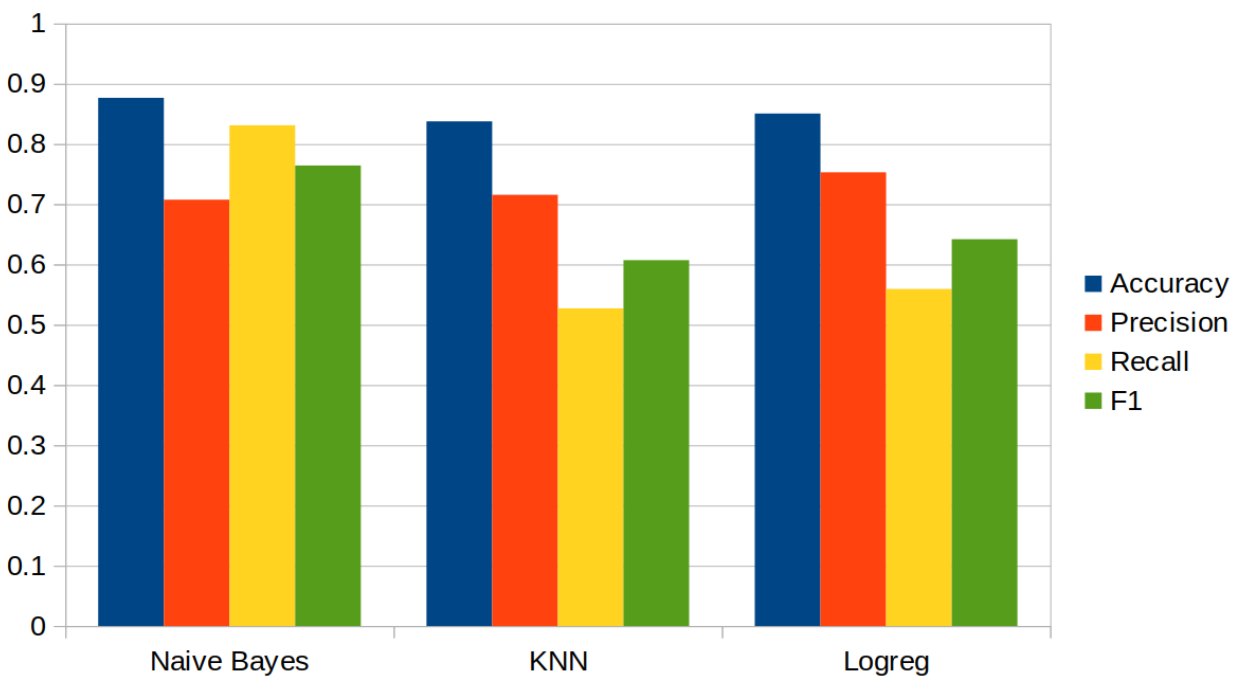
Model 3: Logistic Regression

Logistic regression is a linear binary classification model can be generalized to multi-class classification using techniques like one-vs-rest or multinomial logistic regression. It models the probability of a data point belonging to a particular class using the logistic function, and fits the model parameters by minimizing the log-loss (cross-entropy) between the true labels and the predicted probabilities. It is simple, interpretable, and works well for linearly separable data.

Due to the nature of logistic regression, all categorical data were converted using one-hot encoding to achieve high accuracy and remove unnecessary bias.

Feature	Statistic
Accuracy	85.06%
Precision	75.31%
Recall	55.95%
F1	64.20%

Conclusion



As seen above Naive Bayes classifier is the better option compared to KNN and Logreg. This is because Logreg works well with continuous data that is normalized. KNN also works better on continuous data. But Naive Bayes is defined for categorical data and works well in our case.

Part B: Artificial Neural Network: MNIST

We are considering **100 Epochs** and dataset splits of 67% for Training and 33% for Testing. We have 2 variants of **Hidden layers 2 and 3**, 2 variants of the **Number of neurons in each hidden layer, 100 and 150**, and 3 variants of **Activation functions TanH, Sigmoid, and ReLU**.

In the ANN Classification Reports, We have macro and micro accuracies. Macro accuracy takes an average of all accuracies, while micro adds all numerators and adds all denominators and does the division. It also contains each class's precision, recall, and f1 score.

We used Pytorch to make these artificial neural networks and used the torch datasets repository only. In ANN, we convert the image, which is a 28x28, into a 784-sized tuple which we feed into the network, and using backpropagation on the training data, we settle the weights, and then we use this network with on the testing dataset and get probabilities of the number being each number and we use the argmax function to find which of them has the highest probability and take that as output

2 HIDDEN LAYERS, 100 NEURONS, TanH ACTIVATION FUNCTION:

Model #1

```
[ [ 972    0    0    0    0    2    2    1    2    1]
  [   0 1128    1    1    0    0    1    1    3    0]
  [   4    1 1016    2    1    0    2    3    3    0]
  [   0    0    4  991    0    5    0    3    3    4]
  [   1    1    1    0  966    0    5    2    0    6]
  [   2    0    0   12    1  867    3    2    3    2]
  [   3    3    2    1    5    4  938    0    2    0]
  [   0    1   12    0    3    1    1 1005    1    4]
  [   2    0    4    2    3    2    2    3  951    5]
  [   0    2    0    3    9    4    0    6    2  983]]
```

Classification report for Hidden Layers: 2, Neurons in Each Hidden Layer: 100, Activation Function: t :

	precision	recall	f1-score	support
0	0.987805	0.991837	0.989817	980
1	0.992958	0.993833	0.993395	1135
2	0.976923	0.984496	0.980695	1032
3	0.979249	0.981188	0.980218	1010
4	0.977733	0.983707	0.980711	982
5	0.979661	0.971973	0.975802	892
6	0.983229	0.979123	0.981172	958
7	0.979532	0.977626	0.978578	1028
8	0.980412	0.976386	0.978395	974
9	0.978109	0.974232	0.976167	1009
accuracy			0.981700	10000
macro avg	0.981561	0.981440	0.981495	10000
weighted avg	0.981700	0.981700	0.981694	10000

2 HIDDEN LAYERS, 100 NEURONS, Sigmoid ACTIVATION FUNCTION:

Model #2

```
[[ 973    0    1    1    0    0    2    1    2    0]
 [   0 1117    0    0    0    2    6    1    8    1]
 [   4    1 1002    2    4    0    4    4    9    2]
 [   0    0    3  969    1    9    0   10    4   14]
 [   1    0    2    0  967    0    4    1    0    7]
 [   1    0    0    6    2  875    4    1    2    1]
 [   3    1    2    0    4    3  945    0    0    0]
 [   2    2    9    0    4    0    0  996    4   11]
 [   3    0    1    3    5    9    4    3  938    8]
 [   1    2    0    0   20    6    1    1    1  977]]
```

Classification report for Hidden Layers: 2, Neurons in Each Hidden Layer: 100, Activation Function: s :

	precision	recall	f1-score	support
0	0.984818	0.992857	0.988821	980
1	0.994657	0.984141	0.989371	1135
2	0.982353	0.970930	0.976608	1032
3	0.987768	0.959406	0.973380	1010
4	0.960278	0.984725	0.972348	982
5	0.967920	0.980942	0.974388	892
6	0.974227	0.986430	0.980290	958
7	0.978389	0.968872	0.973607	1028
8	0.969008	0.963039	0.966014	974
9	0.956905	0.968285	0.962562	1009
accuracy			0.975900	10000
macro avg	0.975632	0.975963	0.975739	10000
weighted avg	0.976029	0.975900	0.975906	10000

2 HIDDEN LAYERS, 100 NEURONS, ReLU ACTIVATION FUNCTION:

Model #3

```
[[ 967    0    1    1    0    0    4    3    3    1]
 [   0 1126    1    2    0    0    1    1    4    0]
 [   2    1 1014    1    0    0    0    6    7    1]
 [   0    0    7 986    0    5    1    3    3    5]
 [   2    1    3    0 953    0    4    5    0 14]
 [   4    0    0    8    1 862    5    1 10    1]
 [   2    2    2    0    6    0 941    0    5    0]
 [   0    2    8    0    1    0    0 1001    3 13]
 [   1    0    2    3    4    1    1    2 956    4]
 [   0    4    0    1    9    1    1    5    3 985]]
```

Classification report for Hidden Layers: 2, Neurons in Each Hidden Layer: 100, Activation Function: r :

	precision	recall	f1-score	support
0	0.988753	0.986735	0.987743	980
1	0.991197	0.992070	0.991634	1135
2	0.976879	0.982558	0.979710	1032
3	0.984032	0.976238	0.980119	1010
4	0.978439	0.970468	0.974438	982
5	0.991945	0.966368	0.978989	892
6	0.982255	0.982255	0.982255	958
7	0.974684	0.973735	0.974209	1028
8	0.961771	0.981520	0.971545	974
9	0.961914	0.976214	0.969011	1009
accuracy			0.979100	10000
macro avg	0.979187	0.978816	0.978965	10000
weighted avg	0.979195	0.979100	0.979113	10000

2 HIDDEN LAYERS, 150 NEURONS, TanH ACTIVATION FUNCTION:

Model #4

```
[[ 974    0    1    0    0    0    2    1    2    0]
 [   0 1127    1    1    0    0    2    1    3    0]
 [   1    2 1016    2    2    0    1    7    1    0]
 [   0    0    4  993    0    2    0    4    1    6]
 [   1    0    3    1  965    0    3    1    0    8]
 [   2    0    0    9    0  874    2    1    3    1]
 [   3    2    0    1    3    2  946    0    1    0]
 [   0    2    10    0    0    0    0 1009    3    4]
 [   4    0    1    4    3    2    0    3  954    3]
 [   1    2    0    1    7    2    1    2    3  990]]
```

Classification report for Hidden Layers: 2, Neurons in Each Hidden Layer: 150, Activation Function: t :

	precision	recall	f1-score	support
0	0.987830	0.993878	0.990844	980
1	0.992952	0.992952	0.992952	1135
2	0.980695	0.984496	0.982592	1032
3	0.981225	0.983168	0.982196	1010
4	0.984694	0.982688	0.983690	982
5	0.990930	0.979821	0.985344	892
6	0.988506	0.987474	0.987990	958
7	0.980564	0.981518	0.981040	1028
8	0.982492	0.979466	0.980977	974
9	0.978261	0.981169	0.979713	1009
accuracy			0.984800	10000
macro avg	0.984815	0.984663	0.984734	10000
weighted avg	0.984809	0.984800	0.984800	10000

2 HIDDEN LAYERS, 150 NEURONS, Sigmoid ACTIVATION FUNCTION:

Model #5

```
[[ 970    0    1    2    1    0    1    0    2    3]
 [   0 1124    3    2    0    0    2    1    3    0]
 [   1    0 1017    5    1    0    1    3    3    1]
 [   1    0    1 994    0    3    0    3    7    1]
 [   1    1    2    0 966    0    4    2    1    5]
 [   2    0    0    8    2 870    3    1    4    2]
 [   3    1    1    0    4    5 943    0    1    0]
 [   1    2   11    0    1    0    0 1004    3    6]
 [   2    0    3    5    4    3    0    4 948    5]
 [   2    3    0    2    5    1    1    4    4 987]]
```

Classification report for Hidden Layers: 2, Neurons in Each Hidden Layer: 150, Activation Function: s :

	precision	recall	f1-score	support
0	0.986775	0.989796	0.988283	980
1	0.993811	0.990308	0.992056	1135
2	0.978826	0.985465	0.982134	1032
3	0.976424	0.984158	0.980276	1010
4	0.981707	0.983707	0.982706	982
5	0.986395	0.975336	0.980834	892
6	0.987435	0.984342	0.985886	958
7	0.982387	0.976654	0.979512	1028
8	0.971311	0.973306	0.972308	974
9	0.977228	0.978196	0.977712	1009
accuracy			0.982300	10000
macro avg	0.982230	0.982127	0.982171	10000
weighted avg	0.982319	0.982300	0.982302	10000

2 HIDDEN LAYERS, 150 NEURONS, ReLU ACTIVATION FUNCTION:

Model #6

```
[[ 975    1    1    0    0    0    1    1    1    0]
 [   0 1129    0    0    0    1    1    1    3    0]
 [   3    1 1013    3    1    0    0    4    7    0]
 [   0    0    2 996    1    1    0    4    5    1]
 [   1    0    4    0 964    0    2    2    1    8]
 [   2    0    0    6    1 870    5    1    5    2]
 [   2    2    2    1    3    4 944    0    0    0]
 [   1    2    8    1    1    0    0 1006    2    7]
 [   2    0    4    2    1    3    1    3 955    3]
 [   1    2    0    0    5    5    2    4    0 990]]
```

Classification report for Hidden Layers: 2, Neurons in Each Hidden Layer: 150, Activation Function: r :

	precision	recall	f1-score	support
0	0.987842	0.994898	0.991357	980
1	0.992964	0.994714	0.993838	1135
2	0.979691	0.981589	0.980639	1032
3	0.987116	0.986139	0.986627	1010
4	0.986694	0.981670	0.984176	982
5	0.984163	0.975336	0.979730	892
6	0.987448	0.985386	0.986416	958
7	0.980507	0.978599	0.979552	1028
8	0.975485	0.980493	0.977983	974
9	0.979228	0.981169	0.980198	1009
accuracy			0.984200	10000
macro avg	0.984114	0.983999	0.984052	10000
weighted avg	0.984203	0.984200	0.984197	10000

3 HIDDEN LAYERS, 100 NEURONS, TanH ACTIVATION FUNCTION:

Model #7

```
[[ 973    0    2    0    0    0    2    1    2    0]
 [   0 1130    1    0    0    0    2    0    2    0]
 [   2    0 1014    5    4    0    0    3    4    0]
 [   1    0    3  986    0    5    0    9    4    2]
 [   1    0    2    0  961    0    5    4    3    6]
 [   2    0    0    8    1  871    4    1    5    0]
 [   2    3    2    1    2    3  945    0    0    0]
 [   0    2    9    0    0    0    0 1012    2    3]
 [   1    1    3    6    3    3    2    4  946    5]
 [   2    2    0    2   10    5    1    8    3  976]]
```

Classification report for Hidden Layers: 3, Neurons in Each Hidden Layer: 100, Activation Function: t :

	precision	recall	f1-score	support
0	0.988821	0.992857	0.990835	980
1	0.992970	0.995595	0.994281	1135
2	0.978764	0.982558	0.980658	1032
3	0.978175	0.976238	0.977205	1010
4	0.979613	0.978615	0.979114	982
5	0.981962	0.976457	0.979202	892
6	0.983351	0.986430	0.984888	958
7	0.971209	0.984436	0.977778	1028
8	0.974253	0.971253	0.972751	974
9	0.983871	0.967294	0.975512	1009
accuracy			0.981400	10000
macro avg	0.981299	0.981173	0.981222	10000
weighted avg	0.981410	0.981400	0.981391	10000

3 HIDDEN LAYERS, 100 NEURONS, Sigmoid ACTIVATION FUNCTION:

Model #8

```
[[ 966    0    2    2    2    2    3    1    1    1]
 [   0 1126    0    2    0    1    1    1    4    0]
 [   3    4 1010    6    2    0    2    3    2    0]
 [   0    0    3  989    0    3    0    5    4    6]
 [   3    0    1    0  963    0    4    2    1    8]
 [   2    1    0    6    2  870    5    0    3    3]
 [   2    3    2    1    7    4  935    0    4    0]
 [   2    4   10    3    1    0    0 1000    2    6]
 [   2    0    4    5    2    2    2    3  950    4]
 [   3    2    0    3    9    3    0    0    3  986]]
```

Classification report for Hidden Layers: 3, Neurons in Each Hidden Layer: 100, Activation Function: s :

	precision	recall	f1-score	support
0	0.982706	0.985714	0.984208	980
1	0.987719	0.992070	0.989890	1135
2	0.978682	0.978682	0.978682	1032
3	0.972468	0.979208	0.975826	1010
4	0.974696	0.980652	0.977665	982
5	0.983051	0.975336	0.979178	892
6	0.982143	0.975992	0.979058	958
7	0.985222	0.972763	0.978953	1028
8	0.975359	0.975359	0.975359	974
9	0.972387	0.977205	0.974790	1009
accuracy			0.979500	10000
macro avg	0.979443	0.979298	0.979361	10000
weighted avg	0.979518	0.979500	0.979499	10000

3 HIDDEN LAYERS, 100 NEURONS, ReLU ACTIVATION FUNCTION:

Model #9

```
[[ 970    0    2    1    0    0    1    3    1    2]
 [   0 1130    0    1    0    1    0    1    2    0]
 [   4    0 1013    2    2    0    1    7    3    0]
 [   0    0    4  993    0    2    0    3    4    4]
 [   0    0    2    0  964    1    5    2    1    7]
 [   3    0    0  11    0  863    5    1    3    6]
 [   3    2    1    1    5    6  932    0    8    0]
 [   0    1    8    0    0    0    0 1008    4    7]
 [   1    0    4    6    2    3    3    3  947    5]
 [   2    2    0    1  11    5    0    1    4  983]]
```

Classification report for Hidden Layers: 3, Neurons in Each Hidden Layer: 100, Activation Function: r :

	precision	recall	f1-score	support
0	0.986775	0.989796	0.988283	980
1	0.995595	0.995595	0.995595	1135
2	0.979691	0.981589	0.980639	1032
3	0.977362	0.983168	0.980257	1010
4	0.979675	0.981670	0.980671	982
5	0.979569	0.967489	0.973491	892
6	0.984161	0.972860	0.978478	958
7	0.979592	0.980545	0.980068	1028
8	0.969294	0.972279	0.970784	974
9	0.969428	0.974232	0.971824	1009
accuracy			0.980300	10000
macro avg	0.980114	0.979922	0.980009	10000
weighted avg	0.980312	0.980300	0.980298	10000

3 HIDDEN LAYERS, 150 NEURONS, TanH ACTIVATION FUNCTION:

Model #10

```
[[ 972    1    0    1    0    1    4    1    0    0]
 [   0 1128    1    2    0    0    3    1    0    0]
 [   0    0 1017    1    1    0    2    5    5    1]
 [   0    0    4 993    0    2    0    2    2    7]
 [   1    0    1    0 961    0    4    2    1   12]
 [   2    0    0    7    1 872    2    2    4    2]
 [   1    2    1    1    4    3 944    0    2    0]
 [   1    3    7    1    0    0    0 1009    1    6]
 [   3    0    4    5    1    2    1    2 950    6]
 [   3    2    0    2    5    2    1    3    0 991]]
```

Classification report for Hidden Layers: 3, Neurons in Each Hidden Layer: 150, Activation Function: t :

	precision	recall	f1-score	support
0	0.988810	0.991837	0.990321	980
1	0.992958	0.993833	0.993395	1135
2	0.982609	0.985465	0.984035	1032
3	0.980257	0.983168	0.981710	1010
4	0.987667	0.978615	0.983120	982
5	0.988662	0.977578	0.983089	892
6	0.982310	0.985386	0.983846	958
7	0.982473	0.981518	0.981995	1028
8	0.984456	0.975359	0.979887	974
9	0.966829	0.982161	0.974435	1009
accuracy			0.983700	10000
macro avg	0.983703	0.983492	0.983583	10000
weighted avg	0.983735	0.983700	0.983704	10000

3 HIDDEN LAYERS, 150 NEURONS, Sigmoid ACTIVATION FUNCTION:

Model #11

```
[[ 972    0    2    0    0    1    3    0    1    1]
 [   0 1124    3    3    0    0    1    1    3    0]
 [   0    2 1014    5    2    0    2    3    4    0]
 [   0    0    6 985    0    6    0    4    5    4]
 [   0    1    4    0 952    1    7    3    1   13]
 [   3    0    0    8    0 872    2    0    5    2]
 [   3    2    1    0    3    4 943    0    2    0]
 [   1    4    6    2    1    0    0 1006    0    8]
 [   2    0    5    6    1    4    0    4 948    4]
 [   2    2    0    3    8    3    0    2    2 987]]
```

Classification report for Hidden Layers: 3, Neurons in Each Hidden Layer: 150, Activation Function: s :

	precision	recall	f1-score	support
0	0.988810	0.991837	0.990321	980
1	0.990308	0.990308	0.990308	1135
2	0.974063	0.982558	0.978292	1032
3	0.973320	0.975248	0.974283	1010
4	0.984488	0.969450	0.976911	982
5	0.978676	0.977578	0.978127	892
6	0.984342	0.984342	0.984342	958
7	0.983382	0.978599	0.980985	1028
8	0.976313	0.973306	0.974807	974
9	0.968597	0.978196	0.973373	1009
accuracy			0.980300	10000
macro avg	0.980230	0.980142	0.980175	10000
weighted avg	0.980323	0.980300	0.980300	10000

3 HIDDEN LAYERS, 150 NEURONS, ReLU ACTIVATION FUNCTION:

Model #12

```
[[ 974    0    0    1    0    0    1    1    3    0]
 [   0 1125    2    0    0    1    1    0    6    0]
 [   3    4 1011    2    1    0    0    1   10    0]
 [   2    0    2  990    0    4    0    5    6    1]
 [   1    2    3    0  963    0    3    2    0    8]
 [   2    1    0    8    1  868    4    1    5    2]
 [   4    1    0    1    6    3  943    0    0    0]
 [   1    9    7    0    0    0    0 1002    3    6]
 [   0    0    2    4    1    2    1    1  960    3]
 [   2    5    0    0    7    7    0    2    4  982]]
```

Classification report for Hidden Layers: 3, Neurons in Each Hidden Layer: 150, Activation Function: r :

	precision	recall	f1-score	support
0	0.984833	0.993878	0.989335	980
1	0.980820	0.991189	0.985977	1135
2	0.984421	0.979651	0.982030	1032
3	0.984095	0.980198	0.982143	1010
4	0.983657	0.980652	0.982152	982
5	0.980791	0.973094	0.976927	892
6	0.989507	0.984342	0.986918	958
7	0.987192	0.974708	0.980910	1028
8	0.962889	0.985626	0.974125	974
9	0.980040	0.973241	0.976629	1009
accuracy			0.981800	10000
macro avg	0.981824	0.981658	0.981715	10000
weighted avg	0.981854	0.981800	0.981800	10000

3 HIDDEN LAYERS, 100 + 150 NEURONS, TanH ACTIVATION FUNCTION:

```

[[ 952    0    1    2    2    2   15    2    4    0]
 [   0 1111    4    2    0    2    2    6    8    0]
 [   3    0 989   10    6    1    6    8    8    1]
 [   0    0    7 969    0    4    0   10   13    7]
 [   1    1    3    0 942    0    6    3    1   25]
 [   6    0    0   19    2 833    4    4   19    5]
 [   3    3    0    1    8   16 916    0   11    0]
 [   0    3   13    6    2    1    0 989    0   14]
 [   2    0    5   12    3    5    2    6 937    2]
 [   4    2    0    7   17    4    1   10   10 954]]

```

Classification report for Hidden Layers: 2, Neurons in Each Hidden Layer: 150, Activation Function: t :

	precision	recall	f1-score	support
0	0.980433	0.971429	0.975910	980
1	0.991964	0.978855	0.985366	1135
2	0.967710	0.958333	0.962999	1032
3	0.942607	0.959406	0.950932	1010
4	0.959267	0.959267	0.959267	982
5	0.959677	0.933857	0.946591	892
6	0.962185	0.956159	0.959162	958
7	0.952794	0.962062	0.957406	1028
8	0.926805	0.962012	0.944081	974
9	0.946429	0.945491	0.945959	1009
accuracy			0.959200	10000
macro avg	0.958987	0.958687	0.958767	10000
weighted avg	0.959435	0.959200	0.959250	10000

3 HIDDEN LAYERS, 100 + 150 NEURONS, Sigmoid ACTIVATION FUNCTION:

```

[[ 973    0    0    0    0    1    1    1    4    0]
 [   0 1130    0    1    0    0    1    1    2    0]
 [   1    2 1014    2    1    0    1    2    8    1]
 [   0    0    3  997    0    3    0    3    2    2]
 [   2    0    3    0  958    0    4    2    1   12]
 [   2    0    0    7    1  872    2    2    4    2]
 [   2    2    0    1    4    4  944    0    1    0]
 [   0    1    9    0    0    1    0 1009    2    6]
 [   3    0    3    6    2    1    0    3  951    5]
 [   3    2    0    3    9    2    0    1    4  985]]

```

Classification report for Hidden Layers: 2, Neurons in Each Hidden Layer: 150, Activation Function: t :

	precision	recall	f1-score	support
0	0.986815	0.992857	0.989827	980
1	0.993843	0.995595	0.994718	1135
2	0.982558	0.982558	0.982558	1032
3	0.980334	0.987129	0.983720	1010
4	0.982564	0.975560	0.979050	982
5	0.986425	0.977578	0.981982	892
6	0.990556	0.985386	0.987964	958
7	0.985352	0.981518	0.983431	1028
8	0.971399	0.976386	0.973886	974
9	0.972359	0.976214	0.974283	1009
accuracy			0.983300	10000
macro avg	0.983221	0.983078	0.983142	10000
weighted avg	0.983315	0.983300	0.983300	10000

3 HIDDEN LAYERS, 100 + 150 NEURONS, ReLU ACTIVATION FUNCTION:

```

[[ 972    0    3    0    0    0    4    1    0    0]
 [   1 1124    1    0    0    1    3    0    5    0]
 [   2    0 1014    2    1    0    3    4    6    0]
 [   0    0    4 986    0 10    0    2    6    2]
 [   4    0    3    0 949    1    5    1    1 18]
 [   2    0    0    5    1 874    3    2    4    1]
 [   2    2    0    1    1    5 944    0    2    1]
 [   1    2    6    1    1    0    0 1007    2    8]
 [   0    0    4    4    1    0    1    2 957    5]
 [   1    3    0    0    3    4    1    3    1 993]]

```

Classification report for Hidden Layers: 2, Neurons in Each Hidden Layer: 150, Activation Function: t :

	precision	recall	f1-score	support
0	0.986802	0.991837	0.989313	980
1	0.993811	0.990308	0.992056	1135
2	0.979710	0.982558	0.981132	1032
3	0.986987	0.976238	0.981583	1010
4	0.991641	0.966395	0.978855	982
5	0.976536	0.979821	0.978176	892
6	0.979253	0.985386	0.982310	958
7	0.985323	0.979572	0.982439	1028
8	0.972561	0.982546	0.977528	974
9	0.965953	0.984143	0.974963	1009
accuracy			0.982000	10000
macro avg	0.981858	0.981880	0.981836	10000
weighted avg	0.982078	0.982000	0.982006	10000

CONCLUSION

-
- We observe that over 100 epochs, neural networks perform way better than other Machine Learning algorithms.
 - The accuracy does not vary much between models, which would mean that these models have similar accuracies and there is no statistically significant model compared to the other models.
 - We also observe that ReLU tends to have higher accuracy than Sigmoid and TanH in most cases. We are not concluding anything here but are just noting our observations.