Prepared by: KHUSHBOO BHATIA (kxb190007) Date: 04/10/2020

BUAN 6341 Applied Machine Learning

ASSIGNMENT NO 3

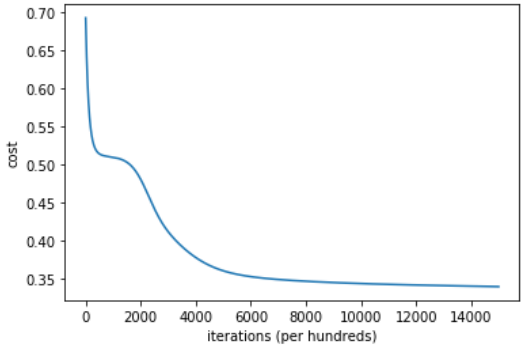
Introduction:

In this report, I have implemented Artificial Neural Networks (ANNs) and K Nearest Neighbors with Python and Scikit-Learn and build a classifier to predict whether or not it will rain tomorrow in Australia. I have used the **Rain in Australia** dataset as my second dataset for this project. This Data set is obtained from Kaggle, link to the dataset is given below –

https://www.kaggle.com/jsphyg/weather-dataset-rattle-package

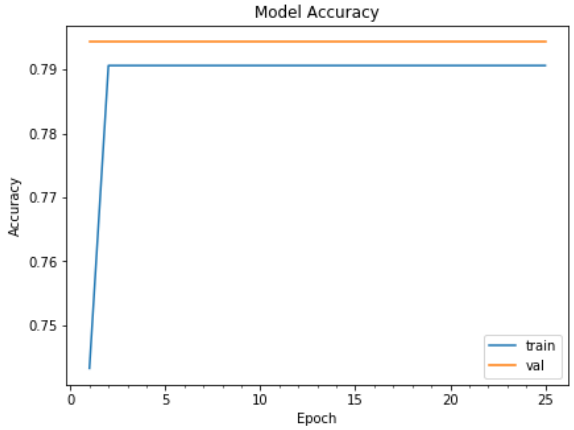
Rain in Australia Next Day Prediction:

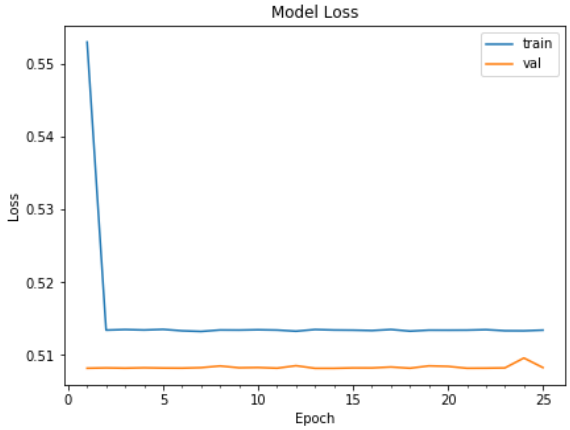
1. **Shallow Neural Network:** I have built the Neural Network using one hidden layer and by writing forward propagation and backpropagation functions to compute cost and predict the binary output and calculating its respective weights and biases.



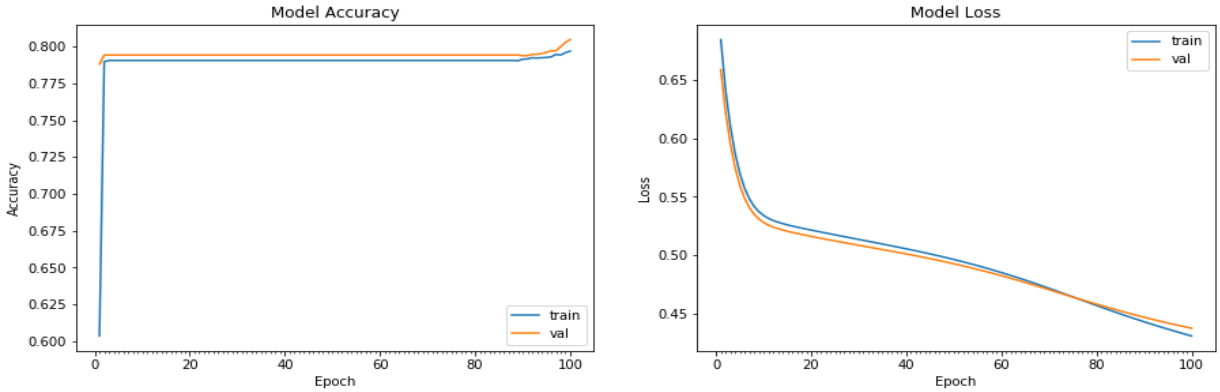
Train Accuracy = 85%, Test Accuracy = 85%. Hence the model is generalized for both training and test set.

2. **Deep Neural Network Using Linear Activation function:** I have used the Sigmoid Activation function. 4 hidden layers are used and Dense is 50 neurons in each layer.



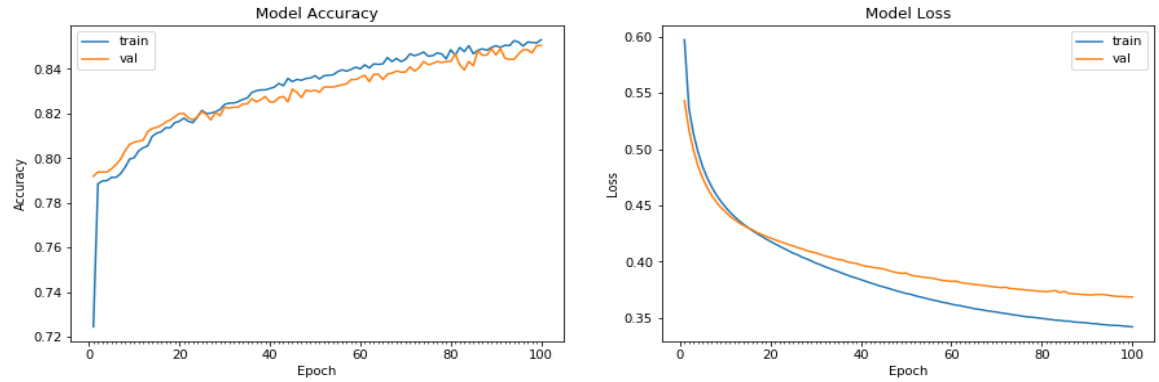


3. **Deep Neural Network Using Non-Linear Activation function (relu):**



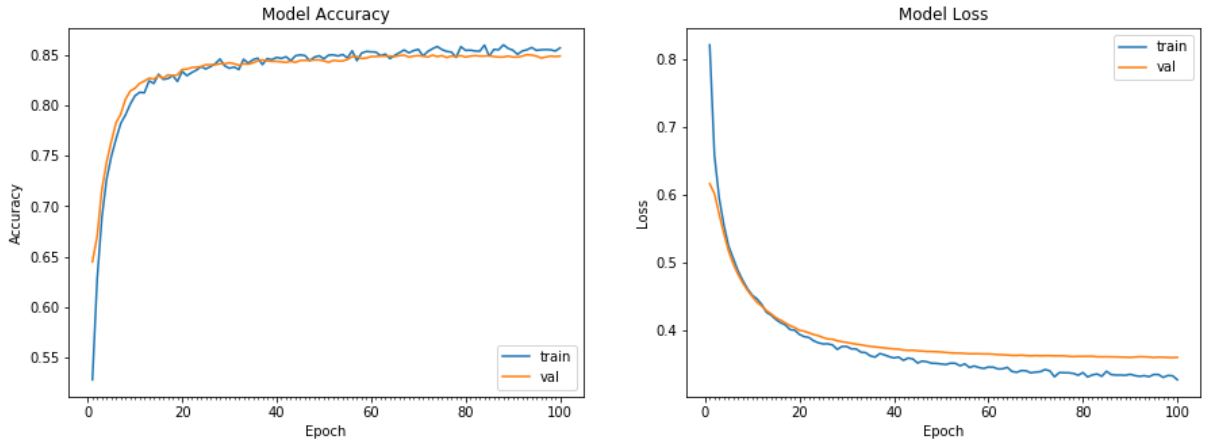
Both the train and validation accuracy increase after 70 epochs with accuracy of 79.92% and 80.06% respectively.

4. **Deep Neural Network Using Non-Linear Activation function (selu):**



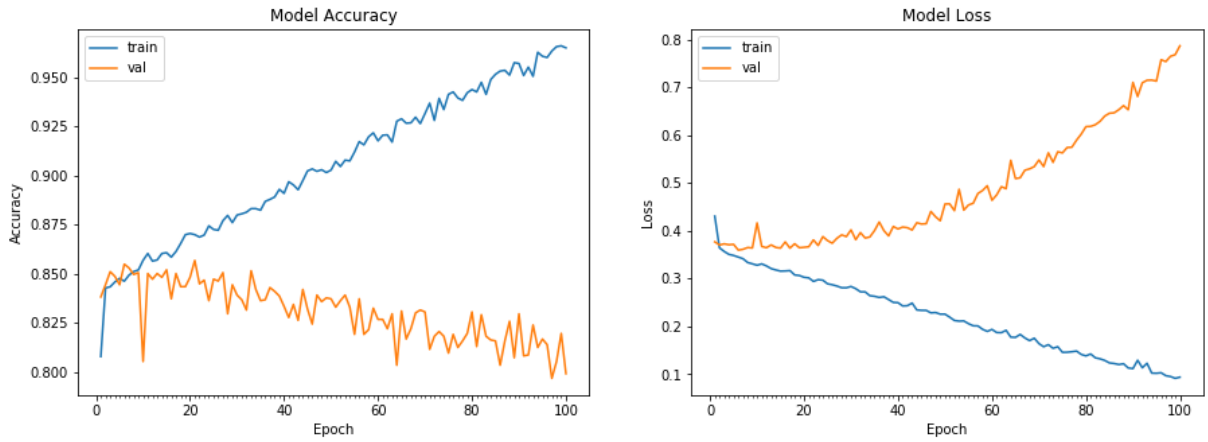
Both the train and validation accuracy increase after 5 epochs with accuracy of 85.31% and 84.86% respectively. The model has generalized well.

5. **Deep Neural Network Using Batch Normalization before Non-Linear Activation:**



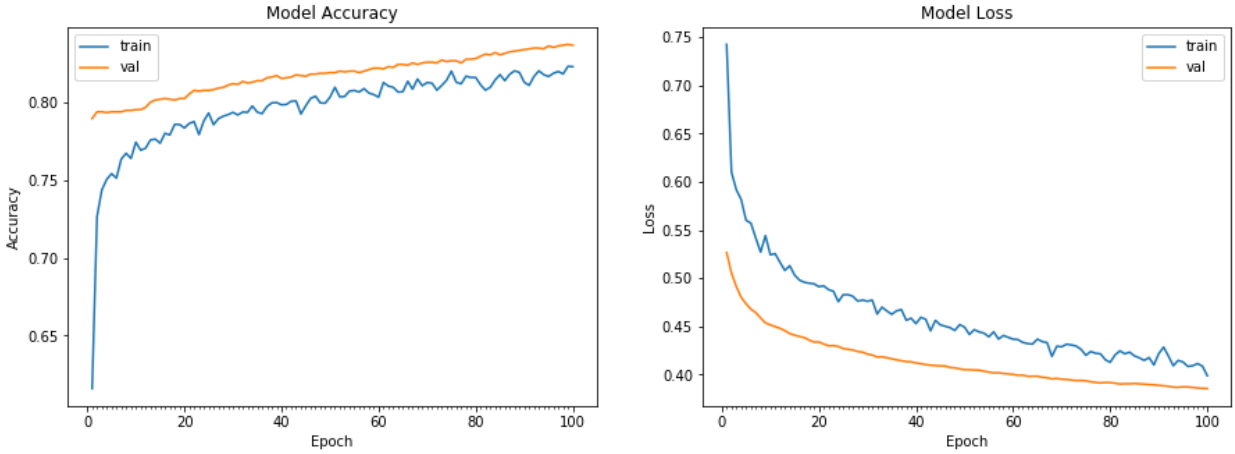
Both the train and validation accuracy increase after 5 epochs with accuracy of 86.17% and 84.96% respectively. The model has generalized well. The AUC is 0.7034.

6. **Deep Neural Network Using Adam Optimizer:**



The train accuracy tremendously increased to 91.7% and test accuracy of 80.6%.

7. **Deep Neural Network Using Drop out regularization:**



Both the train and validation accuracy increase after 5 epochs, with accuracy of 83.42% and 83.76% respectively. The validation accuracy is higher than the train accuracy.

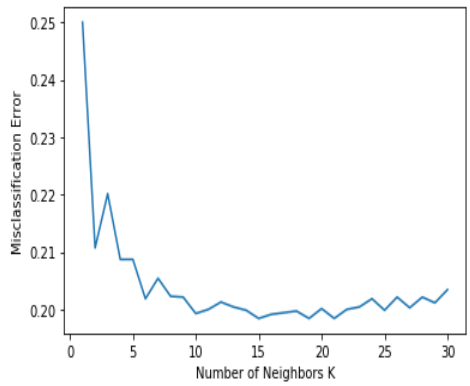
8. **Ensemble method (Adam Optimizer Model):** The train accuracy tremendously increased to 98.5% whereas test accuracy is 83.96%. The model is not generalized well.

9. **Ensemble method (Batch Normalization Model):** The train accuracy is 85.7% whereas test accuracy is 85.13%. The model generalized well.

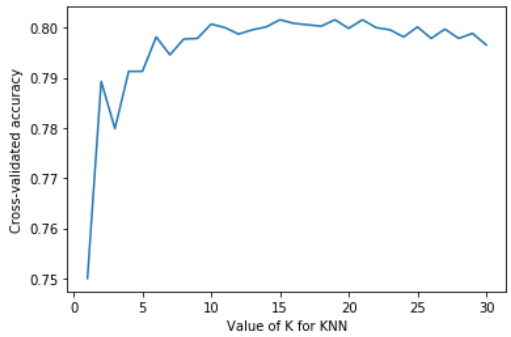
KNN (K – Nearest Neighbor):

10. **KNN (Random value for K):** The k-nearest algorithm(k-NN) is a non-parametric method. The train accuracy is 84.8% and test accuracy is 80.26% for K=5. The Area under curve is 0.5893.
11. **KNN (Finding optimal value for N by changing N):**

```
The optimal number of neighbors is 15
```

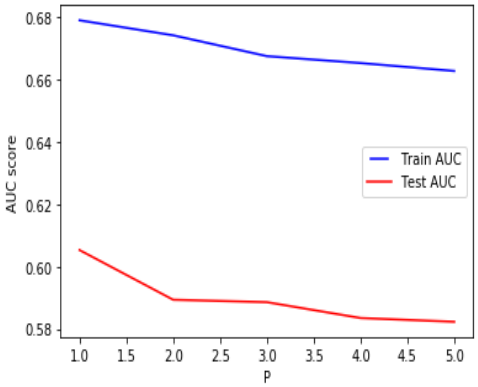


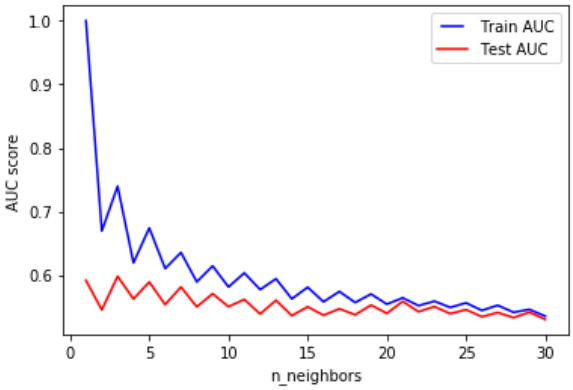

```
Text(0, 0.5, 'Cross-validated accuracy')
```



The optimal value for K is 21. The train accuracy is 81.2% and test accuracy is 81.4%. The Area under curve is 0.558.

12. <u>KNN (Finding optimal value for K by changing distance):</u>





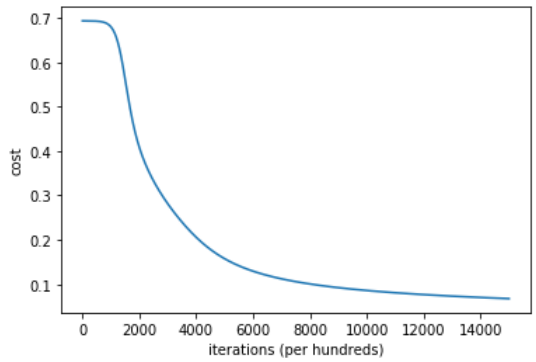
13. <u>KNN (Finding optimal value for N by changing weights):</u> The train accuracy is 100% and test accuracy is 80.7%. The Area under curve is 0.555.

Model	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score	AUC
Shallow Neural Network	85%	85%	0.64	0.8	0.71	0.5
Linear Activation function	79.17%	79.40%	0.79	1.00	0.89	0.5
Non-Linear Activation function (relu)	79.92%	80.06%	0.8	0.99	0.89	0.528
Non-Linear Activation function (selu)	85.31%	84.86%	0.87	0.95	0.91	0.706
Batch Normalization	86.17%	84.96%	0.87	0.95	0.91	0.703
Adam Optimizer	91.78%	80.66%	0.88	0.87	0.88	0.711
Drop out regularization	83.42%	83.76%	0.85	0.96	0.9	0.665
Ensemble method (Adam Optimizer Model)	98.50%	83.96%	0.88	0.92	0.9	0.722
Ensemble method (Batch Normalization Model)	85.70%	85.13%	0.87	0.96	0.91	0.701
KNN (Random value for K)	84.80%	80.20%	0.83	0.95	0.88	0.589
KNN (Optimal value for N by changing N)	81.20%	81.40%	0.81	0.99	0.89	0.558
KNN (Optimal value for K by changing distance)	100%	80.76%	0.81	0.98	0.89	0.555

In terms of generalization and Neural Network using Selu activation function, batch normalization model and Ensemble using batch normalization are the best models. KNN using distance as weight metric and Ensemble using Adam optimizer model are the best in terms of Train accuracy. Neural Network using Adam optimizer and Ensemble Adam optimizer are the best in terms of best Area under the curve.

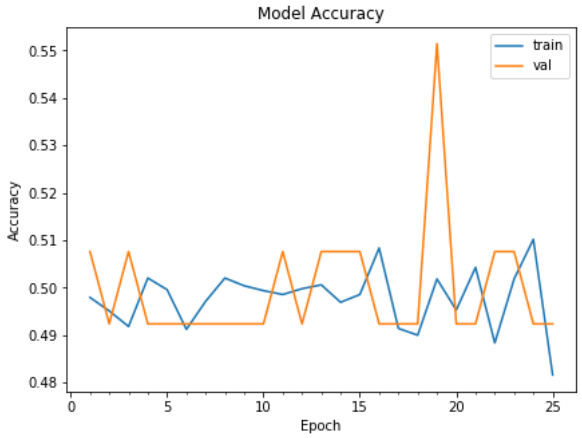
SGEMM GPU kernel performance Data Set:

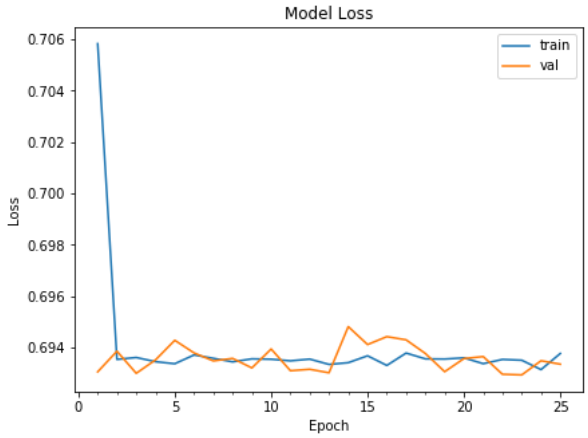
1. **Shallow Neural Network:** I have built the Neural Network using one hidden layer and by writing forward propagation and backpropagation functions to compute cost and predict the binary output and calculating its respective weights and biases.



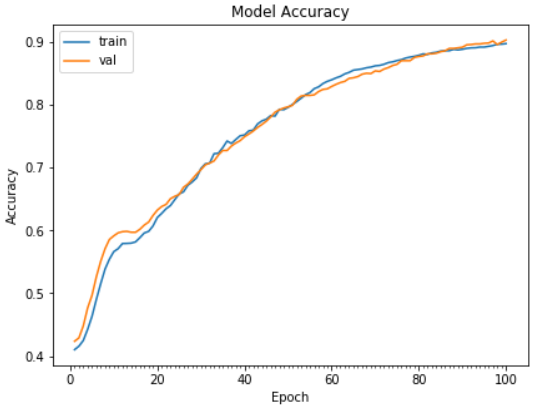
Train Accuracy = 98%, Test Accuracy = 98%. Hence the model is generalized for both training and test set.

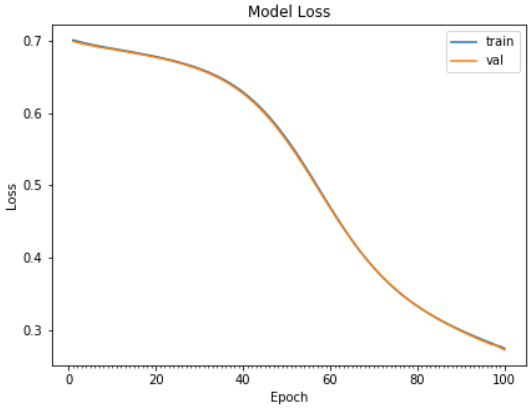
2. **Deep Neural Network Using Linear Activation function:** I have used the Sigmoid Activation function. 4 hidden layers are used and Dense is 50 neurons in each layer.





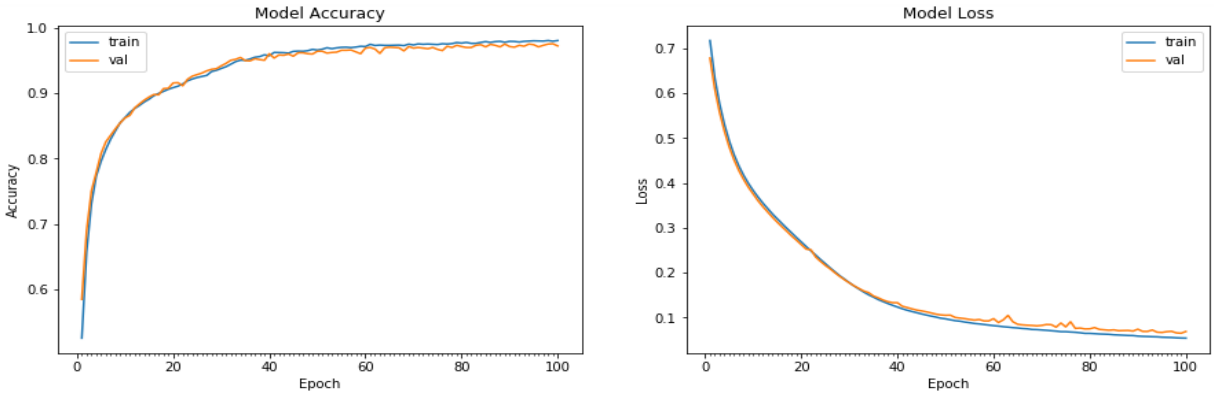
3. **Deep Neural Network Using Non-Linear Activation function (relu):**





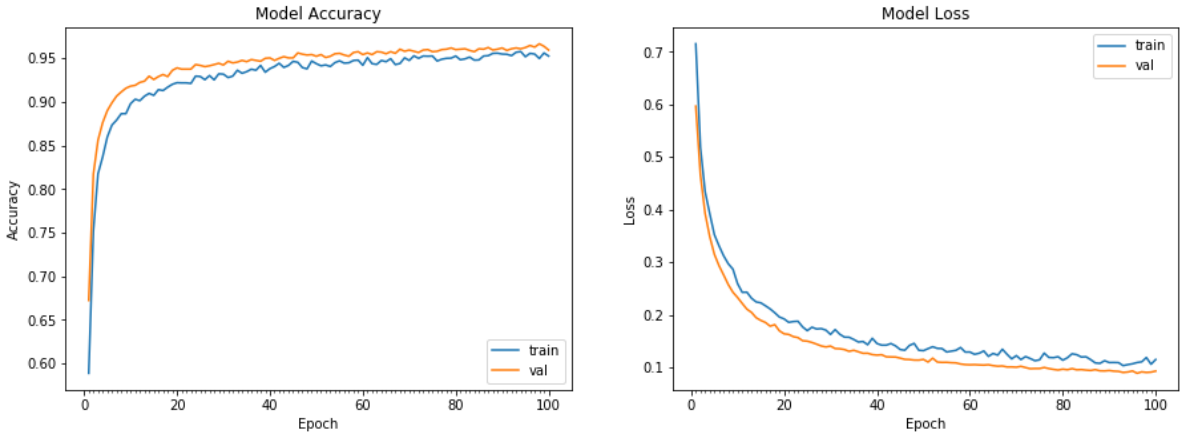
Both the train and validation accuracy increase after 70 epochs with accuracy of 89.97% and 90.86% respectively.

4. **Deep Neural Network Using Non-Linear Activation function (selu):**



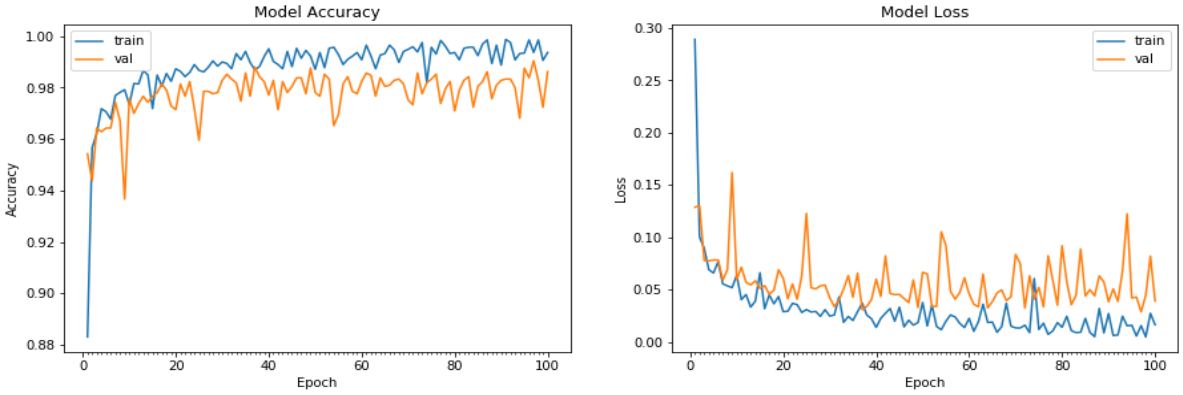
Both the train and validation accuracy increase after 5 epochs with accuracy of 98% and 97.36% respectively. The model has generalized well.

5. **Deep Neural Network Using Batch Normalization before Non-Linear Activation:**



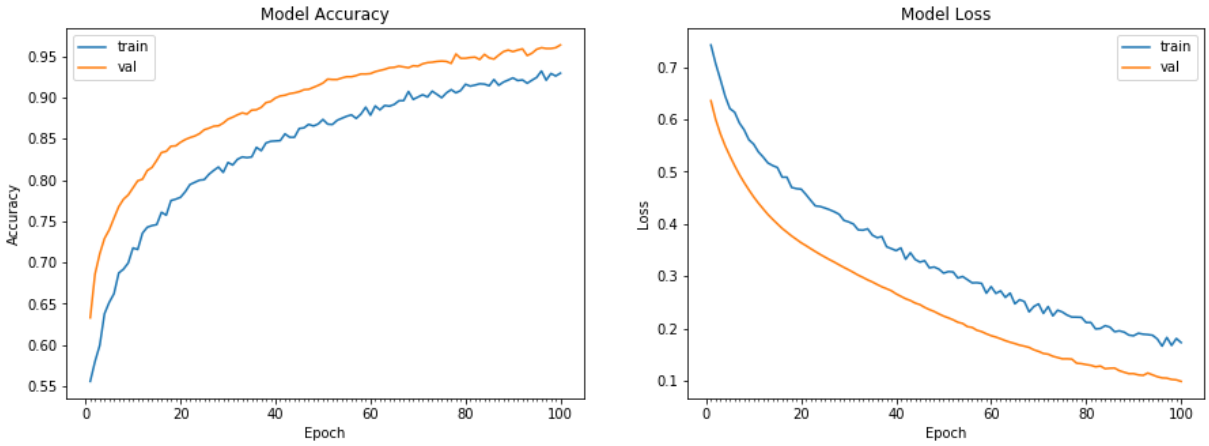
Both the train and validation accuracy increase after 5 epochs with accuracy of 86.17% and 84.96% respectively. The model has generalized well. The AUC is 0.9688.

6. **Deep Neural Network Using Adam Optimizer:**



The train accuracy tremendously increased to 99.47% and test accuracy of 98.56%.

7. **Deep Neural Network Using Drop out regularization:**



Both the train and validation accuracy increase after 5 epochs, with accuracy of 96.51% and 96.73% respectively. The validation accuracy is higher than the train accuracy.

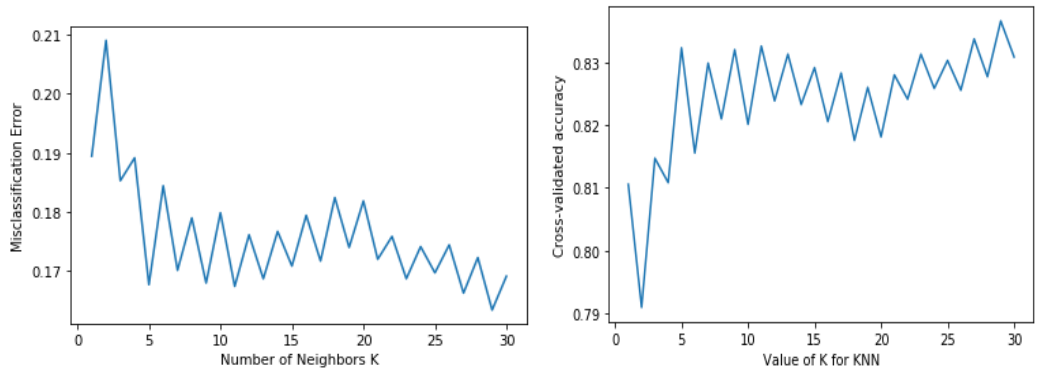
8. **Ensemble method (Adam Optimizer Model):** The train accuracy tremendously increased whereas test accuracy is 99.3%.

9. **Ensemble method (Batch Normalization Model):** The train accuracy is 97.9% whereas test accuracy is 97.1%. The model generalized well.

KNN (K – Nearest Neighbor):

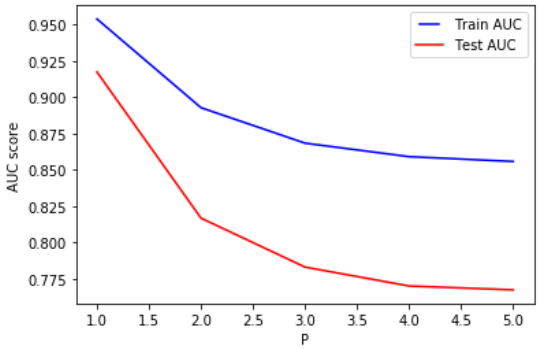
10. **KNN (Random value for K):** The k-nearest algorithm(k-NN) is a non-parametric method. The train accuracy is 89.3% and test accuracy is 81.63% for K=5. The Area under curve is 0.8166.

11. **KNN (Finding optimal value for N by changing N):**



The optimal value for K is 29. The train accuracy is 85.2% and test accuracy is 83.5%. The Area under curve is 0.8361.

12. **KNN (Finding optimal value for K by changing distance):**



13. **KNN (Finding optimal value for N by changing weights):** The train accuracy is 100% and test accuracy is 83.7%. The Area under curve is 0.8376.

Model	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score	AUC
Shallow Neural Network	98%	98%	0.64	0.8	0.71	0.5
Linear Activation function	49.78%	49.56%	0.5	1.00	0.66	0.5
Non-Linear Activation function (relu)	89.97%	90.86%	0.87	0.96	0.91	0.9
Non-Linear Activation function (selu)	98.00%	97.36%	0.98	0.97	0.97	0.9
Batch Normalization	86.17%	84.96%	0.95	0.99	0.97	0.9
Adam Optimizer	99.47%	98.56%	0.98	1	0.99	0.985
Drop out regularization	96.51%	96.73%	0.95	0.99	0.97	0.967
Ensemble method (Adam Optimizer Model)	100.00%	99.30%	0.99	0.99	0.99	0.992
Ensemble method (Batch Normalization Model)	97.90%	97.10%	0.95	0.99	0.97	0.971
KNN (Random value for K)	89.30%	81.63%	0.79	0.86	0.82	0.816
KNN (Optimal value for N by changing N)	85.20%	83.50%	0.78	0.92	0.85	0.836
KNN (Optimal value for K by changing distance)	100%	83.70%	0.79	0.91	0.85	0.837

What additional things we can do to get better results:

- We can change the number of layers and number of neurons in each layer and find the optimal value.
- Neural network has various activation functions (Sigmoid, tanh, relu, selu, soft plus, soft sign etc). We have used 4 of them and found selu has given the best results. We can try different activation functions in different layers.
- Neural network has dropout regularization. We have used dropout value of 0.25. We can vary dropouts and find the optimal value.
- We can try different activation functions with different optimizers such as Adam optimizers.