Prepared by : DharmaTeja Chelamalasetty, id : DXC210022 Date : 07/23/2022

BUAN 6341 Applied Machine Learning

ASSIGNMENT NO 3

Seoul Rental Bike prediction Part III

Executive Summary

- **Converted the data set to 2 classes based on Rented Bike Count median value, if <= Median it is treated as class 0, otherwise class 1.**
- **After Experimenting with Neural Network MLP classifier below are the best tuned hyper parameters :- Nr of hidden layers =2, Nr of nodes =10 per each layer, activation function = 'relu' has best performance, solver = 'sgd' has best performance, learning rate = 0.001 has best convergence and tol has no much significance .**
- **Model accuracies have been 93.3 and 93.4 for train and test accuracies respectively with 0.93 as AUC and having 98 Type 1 and 75 Type II Errors .**
- **Model performed better during cross validation and achieved 91~93 score , and baselined cv = 10 .**
- **When KPI/Statistics compared with assignment II and assignment III Neural network model performed better w.r.t Accuracies, Type 1, Type II Errors, Cross validation, AUC , but only limitation is very hard to interpret and uses machine capabilities for performance .**

Introduction

In this project, the objective is to use the dataset that is converted during assignment II and implement neural networks package for classifying either class 0(less than median value) or class 1(more than median value) and perform different experimentation w.r.t **nr of hidden layers, nr of nodes, different activation functions, different values of learning_rate_init, different values of threshold** where model is converging etc., and identify the best tuned hyper parameter for the model and evaluate the model using **confusion matrix, AUC** and compare the performance w.r.t models implemented as part of assignment II .

About the Data

The dataset consists of 14 features and 8760 records. The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information. In order to remove the serial correlation with time, hours variable is hot encoded and converted to dummies and ran the model .

Project Outline

The Project is outlined to have 9 parts:

Part 1 : Implement Sklean Neural Network MLP classifier and experiment with different nr of layers .

Part 2 : Implement Sklean Neural Network MLP classifier and experiment with different nr of nodes .

Part 3: Implement Sklean Neural Network MLP classifier and experiment with different activation functions.

Part 4: Implement Sklean Neural Network MLP classifier and experiment with different solvers.

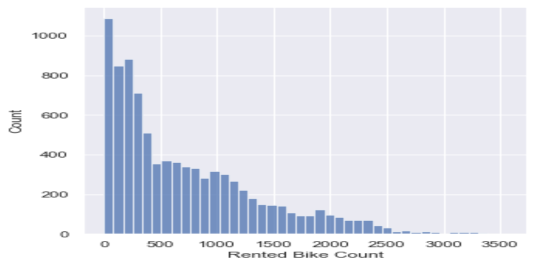
Part 5: Implement Sklean Neural Network MLP classifier and experiment with different learning_rate_init.

Part 6: Implement Sklean Neural Network MLP classifier and experiment with different Thresholds.

Part 7: Evaluate best hyper parameter tuned model with Confusion matrix, AUC and cross validation techniques .

Part 8 : Conclusion, learning curves and comparison w.r.t model implemented in Assignment II .

Part 1:

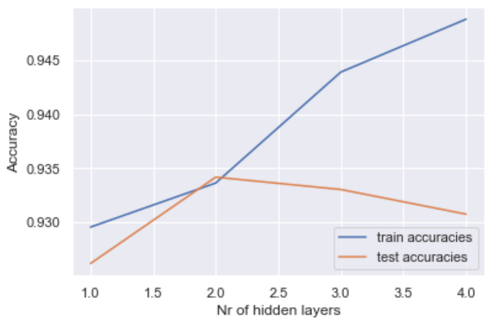


We can observe from the figure that the data of the rented bike count is right skewed. So, we apply a median technique to divide the data in 2 distributions and to handle outliers.
Since the Median value is 504.5., Less than or equal to the value is considered as class 0 and greater than the value is considered as class 1.

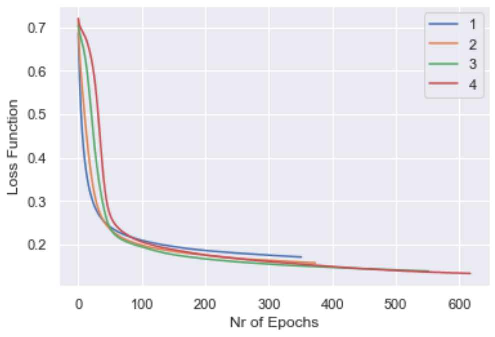
Neural Network MLP Classifier :

We will use neural network MLP classifier as logistic regressor .

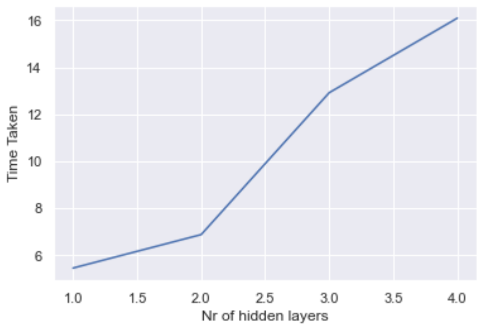
In this experiment we tried with different nr of layers i.e..,(1,2,3,4)



After conducting experiment with different nr of hidden layers we can observe that when nr of hidden layers increase train score is increasing and test score is increasing till 2 layers and after that it is decreasing which thus increasing the bias of the model.
Hence, we can conclude **that nr of hidden layers as 2** can be baselined.



After conducting experiment with different nr of hidden layers we can observe that Loss function is **converging faster when nr of hidden are less** and is **converging slower when hidden layers are more**.



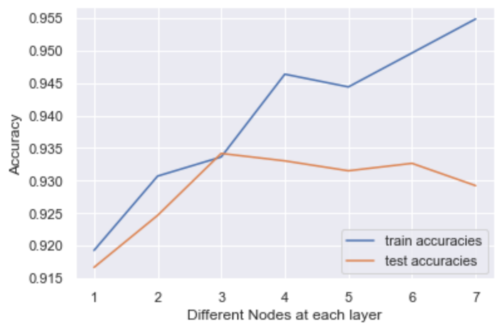
After conducting experiment with different nr of hidden layers we can observe that time taken for convergence of the model is **increasing** with increase in nr of hidden layers.

Conclusion :-

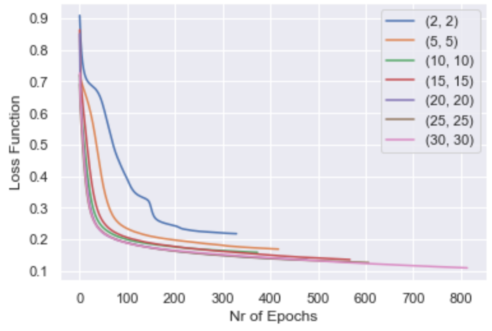
1. After conducting experiments with different layers it is observed that when **nr of hidden layers = 2** , model performance is good with less bias and less variance .
2. When **nr of hidden layers =2**, model convergence is faster when compared to more nr of hidden layers .

Part 2 :

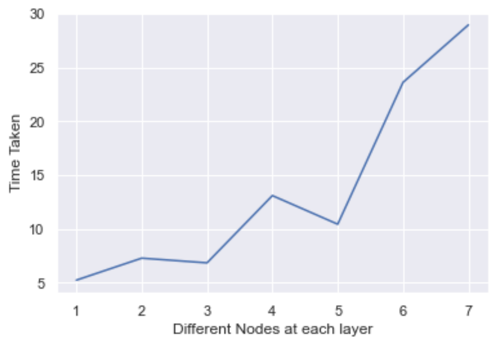
In this experiment we tried with different nr of Nodes i.e..,
1:(2,2),2:(5,5),3:(10,10),4:(15,15),5:(20,20),6:(25,25),7:(30,30) by keeping nr of hidden layers = 2 .



After conducting experiment with different nr of nodes keeping hidden layers as 2, we can observe that when **nr of nodes are increasing train score is increasing,** whereas **test score is increasing and after (10,10) it starts decreasing** hence increasing the **bias** in the model.
So, we can baseline **nr of nodes as (10,10).**



After conducting experiment with different nr of nodes keeping hidden layers as 2, we can observe that with increase in the nr of nodes model convergence is taking more time and iterations.



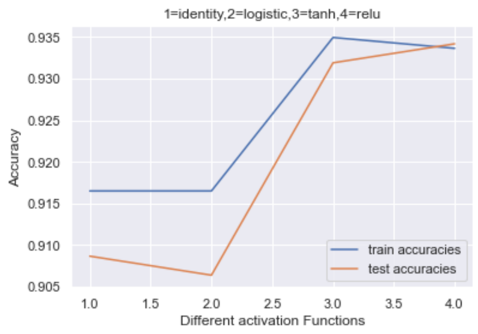
After conducting experiment with different nr of nodes keeping nr of layers as 2 we can observe that with increase in nr of nodes there is increase in time except at (20,20).
But we can see when nr of nodes are 10 per each layer model is converging in less time when compared to other nr of nodes.

Conclusion :-

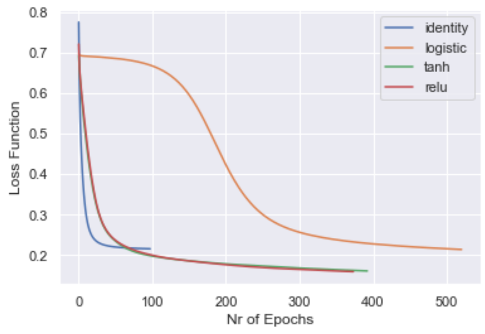
1. After conducting experiments with different **nr of nodes as 10** in each node, model performance is good with less bias and less variance .
2. When **nr of nodes as 10**, model convergence is faster when compared to more nr of nodes per hidden layer .

Part 3 :

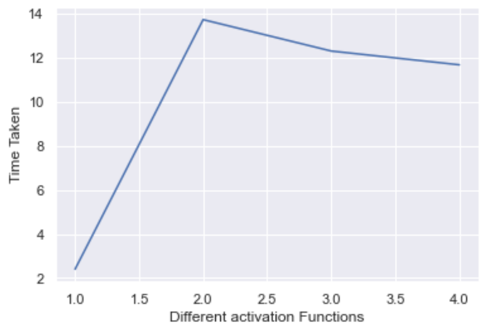
In this experiment we tried with different activation functions i.e..,
'identity','logistic','tanh','relu' , keeping nr of **hidden layers as 2** and **nr of nodes as 10** per each layer .



After conducting experiment with different activation functions, we can observe that with activation function as **relu** model has **low variance and low bias** when compared to other activation functions.
So, we can baseline **relu as best activation function** for the given data set.



After conducting experiment with different activation functions, it is observed that **logistic function is taking more iteration for convergence and identity function is taking less iterations for convergence but tanh and relu has lower loss function and relu is convergence is faster than tanh**



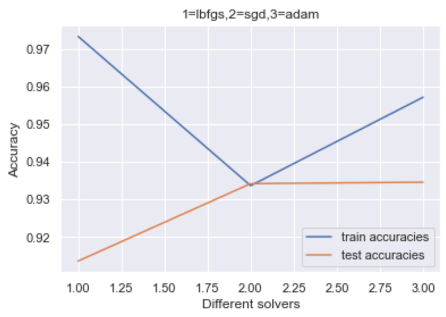
After conducting experiment it is observed that logistic function is taking more time when compared to others and relu function is taking less time after identity function .

Conclusion :-

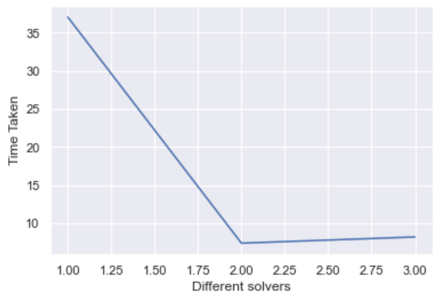
1. After conducting experiments with different activation functions it is observed that when **activation function = ‘relu’ ,** model performance is good with less bias and less variance .
2. When **activation function = ‘relu’** , model convergence is faster when compared to other activation functions .

Part 4 :

In this experiment we tried with different solvers i.e.., **'lbfgs','sgd','adam',** keeping **activation function as 'relu'**, **hidden layers as 2** and **nr of nodes as 10** per each layer .



After conducting experiment with different solvers, it is observed that when solver = 'sgd' model has **low variance and low bias** whereas with solver as 'lbfgs' model is **overfit** and when solver = 'adam' model is observed to have **slightly higher bias** than sgd.
So, we can **baseline solver = 'sgd'** as best solver for the model.



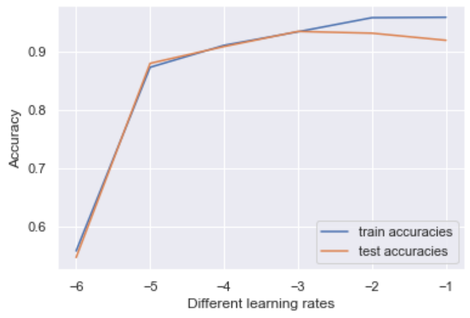
After conducting experiment it is observed that solver = 'sgd' took less time compared to other solvers for the model to convergence .

Conclusion :-

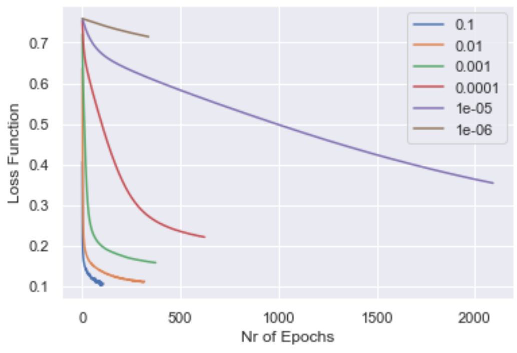
1. After conducting experiments with different solvers it is observed that when **solver = sgd**, model performance is good with less bias and less variance .
2. When **solver='sgd'**, model convergence is faster when compared to other activation functions .

Part 5 :

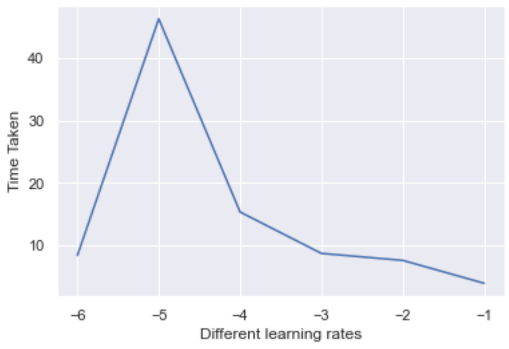
In this experiment we tried with different learning rates i.e.., 0.1,0.01,0.001,0.0001,0.00001,0.000001, keeping **solver='sgd',activation function as 'relu'**, **hidden layers as 2** and **nr of nodes as 10** per each layer .



After conducting experiment with different learning rates, it is observed that with **decrease in learning rate model performance is getting decreased** and we can observe that at **alpha = 0.001 we can see model has low variance and low bias with better accuracy**.
Hence, we can baseline **alpha = 0.001** as best learning rate parameter for the model.



After conducting experiment with when alpha = 0.1 model is converging faster,1e-5, model is not converging after 2000 iterations, at 1e-6 we can see that model is not reaching to its minima.



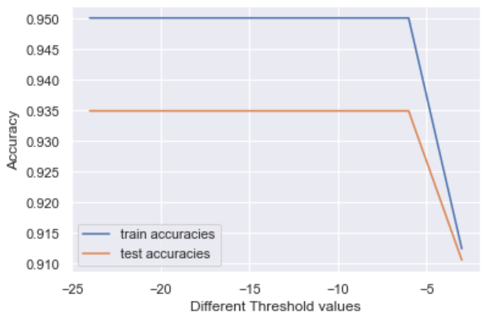
After conducting experiment it is observed that with decrease in alpha time taken for model for execution is more except at 1e-6 and at 1e-6 we can see that model is not reaching to its minima .

Conclusion :-

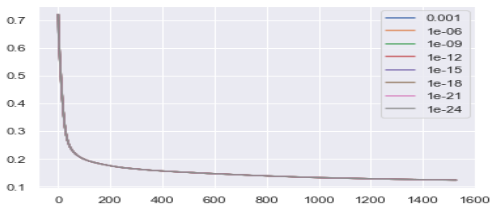
1. After conducting experiments with different learning rates it is observed that when **alpha=0.001,** model performance is good with less bias and less variance .
2. When **alpha = 0.001,** model has better convergence reaching to its minimum and time taken is considerably low .

Part 6 :

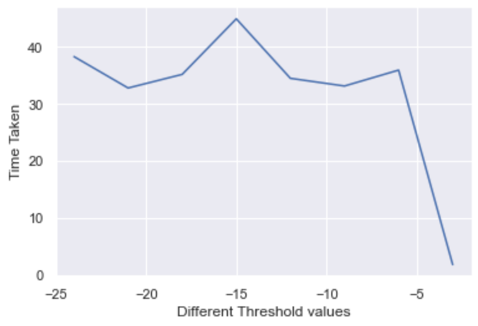
In this experiment we tried with different thresholds i.e.., 1e-3,1e-6,1e-9,1e-12,1e-15,1e-18,1e-21,1e-24, keeping **alpha = 0.001,solver='sgd',activation function as 'relu'**, **hidden layers as 2** and **nr of nodes as 10** per each layer



After conducting experiment with different thresholds , it is observed that this parameter has **no much significance** on the model performance .
So, we can baseline tol=1e-3 as best parameter for the model.



After conducting experiment with different thresholds , it is observed that this parameter has **no much significance** on the loss function it is almost similar for different values of threshold .



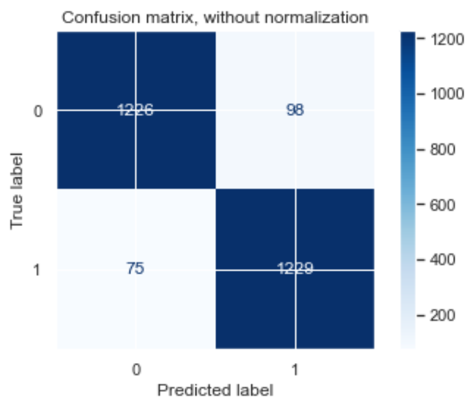
After conducting experiment with different thresholds, it is observed with decrease in threshold time taken for model is increasing.

Conclusion :

1. After this experiment it is observed that tol has no much significance on model performance so default value of 1e-3 is taken as best parameter .

Part 7 :

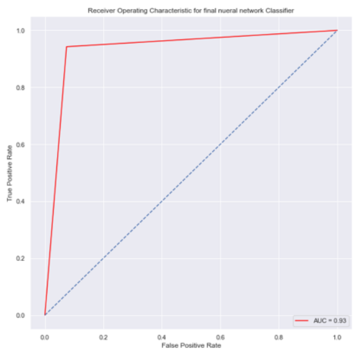
After thorough experimentation with **nr of hidden layers as 2, nr of nodes as 10 per each node, activation function as 'relu', solver = 'sgd', learning rate as 0.001,tol as 1e-3** below are the conclusions :

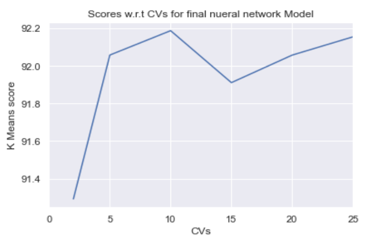


With the best hyper tuned parameter model below are the model performance :

Model/Accuracy	Train	Test
Nueral Network MLP Classifier	93.3	93.4

We can observe model has 98 Type 1 Errors and 75 Type II Errors, since this is a prediction on rental bike count so, technically we can consider Type 1 Errors should be minimal. When we see the ROC and AUC curve, we can observe that neural network model using MLP classifier has AUC of 0.93.





With K Fold 2 we found cross validation score of ~91
With K Fold 5 we found cross validation score of ~92
With K Fold 10 we found cross validation score of ~92
With K Fold 15 we found cross validation score of ~91
With K Fold 20 we found cross validation score of ~92
With K Fold 25 we found cross validation score of ~92
On Avg score is varying between 91~93 and **K-Fold of 2 or 10** can be baselined.

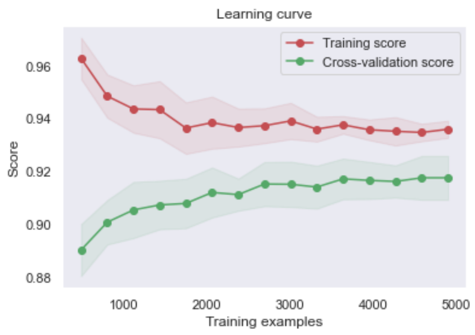
Results :-

1. After thorough experimentation it is observed below are the best hyperparameters :
    a. **Nr of hidden layers** = 2 (if more hidden layers we can observe that model has high bias) .
    b. **Nr of Nodes** = 10 per each node (if more nr of nodes we can observe that model is tending towards high bias) .
    c. **Activation Function** = 'relu' has best model performance .
    d. **Solver** = 'sgd' has the best model performance .
    e. At **alpha** = 0.001 model has best performance
    f. **Tol** (threshold) varying has no much significance on the model performance .

Model/Accuracy	Train	Test	Type 1	Type II	AUC	CV Score
Neural Network MLP Classifier	93.3	93.4	98	75	0.93	91~93

With Neural Network MLP classifier we are able to achieve **good accuracies, AUC and we can also observe that it has low Type 1, Type II Errors**, since this is a prediction on rental bike count so, technically we can consider Type 1 Errors should be minimal.

2. After implementing cross validation, it is observed that MLP classifier has a **cross validation score of 91~93** which is better than the models which we conducted during Assignment II.
3. Learning Curve: -



We can observe that with increase in train size data set model is converging to **low bias and low variance.**

4. Below is the metric comparison post assignment II and assignment III :

Model/Accuracy	Train	Test	Type 1	Type II	AUC	CV Score
SGD Classifier	91.7	91.4	120	105	0.91	67~79
SVM Classifier	97.8	92.5	108	89	0.93	73~79
DT Classifier	88.8	88.8	93	201	0.89	87~88
Neural Network MLP Classifier	93.3	93.4	98	75	0.93	91~93

From the above Statistics we can conclude that Neural Networks is considered to be best model w.r.t accuracies, Type 1 and Type II Errors, AUC and cross validation scores but it is very hard to interpret as it uses machine computational capabilities for modelling .

Apart from the mentioned limitation neural network is the best fit model .