**CS-401 Assignment 2 (Supervised Learning) Report**

**Objective:** The objective of this assignment is to train a classifier, attempting to get good generalization performance.

**Tools Used:** Python and Jupyter Notebook.

**Training Set:** train-io.txt (300,000 rows) has been divided into 2 set train set (80%) and validation set (20%) to check the accuracy of the classifier on the known outputs.

The dataset contains 12 predictor variables (continuous) and one response variable ( 0 | 1 ).

**Steps taken to achieve the objective:**

1. Preprocessing of the data.
   * Checking if the dataset is balanced or unbalanced
   * Looking for null or empty data.
   * Scaling the data.
   * Relation between predictors and response variables
   * Splitting the train data into train and validation set
2. Training various classifier.
   * Logistic regression
   * Random forest
   * Applying PCA to reduce dimensions for random forest
   * Support vector classifier
   * Artificial Neural Network
3. After the accuracy were observed for different models. The model with best accuracy was taken and various variants of that model was run to increase the accuracy.
4. Confusion metrics and various other visual aids were used to depict the results.
5. Output file (test-out.txt) was created for the best model.

**Elaboration on Steps taken:**

1. **Preprocessing of data**

* Dataset was balanced with the number of 0 and 1 as below:

1’s -161077

0’s -138923

* The data set has no missing values.
* The range is more alike in almost every continuous variable hence scaling would not make any difference. [As seen later while implementing the classifiers]
* No major relation between the variables in the dataset with each other or to the response.
* The trainset (300,000 rows) was split in train dataset (240,000) and validation set (60,000)

1. **Training various classifier**
   * **Logistic Regression**

The following parameters were used random\_state=42, penalty='elasticnet', solver='saga', l1\_ratio=0.7

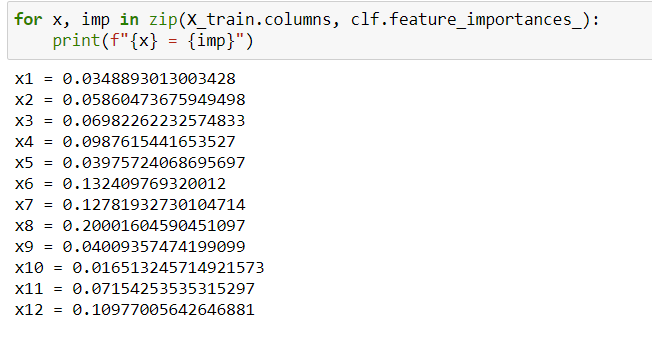
The accuracy of the model was computed to be 54%

* + **Random forest**

The following parameters were used n\_estimators=500, max\_depth=5, random\_state=42, n\_jobs=-1

The accuracy was calculated to be 55%.

The importance of predictor variable was calculated to find if a certain kind of predictor impacted the result. The result was not satisfactory as all none of the predictors have major impact on the accuracy.



Taking into consideration the predictors with highest importance [x4', 'x6', 'x7', 'x8', 'x12'] but no difference on the accuracy was seen (accuracy =55.41%).

* + **Applying PCA to reduce dimensions for random forest**

Applied PCA to limit the components to 3 and parameters to [max\_depth=5, n\_estimators=500, n\_jobs=-1, random\_state=42]

The accuracy in this model was calculated to be 54.67%.

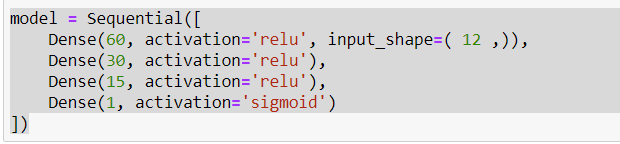
* + **Support vector classifier**

Using parameters as StandardScaler() to scale the data and gamma for the svc to be ‘auto’ .

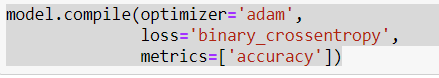
Accuracy on the validation set was computed to be 60%

* + **Artificial Neural Network**

Using 4 layers in the ANN with the following number of nodes and activation function as below :



Compile settings as below:



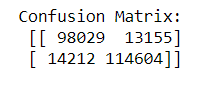
And fitting the model with batch\_size = 20 and epoch=50 there was a tremendous increase in the accuracy ie. 71%

**Conclusion:**

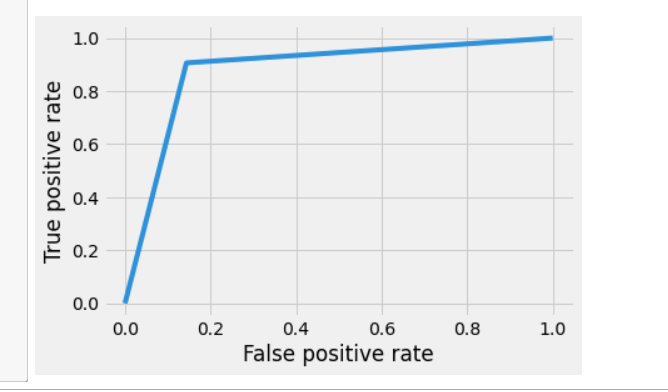
As it was seen that the highest accuracy was achieved by Artificial neural network. ANN was trained differently with some variation in the parameters to increase the efficiency of the model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Layer | Node | Compiler | parameter | Accuracy |
| 4 | 60,30,15,1 | optimizer='adam',  loss='binary\_crossentropy',  metrics=['accuracy'] | batch\_size=20, epochs=50 | 68.55% |
| 4 | 64,32,16 ,1 | optimizer='adam',  loss='binary\_crossentropy',  metrics=['accuracy'] | batch\_size=20, epochs=50 | 87.97% |
| 7 | 64,32,16,8,4,2,1 | optimizer=’adam’,  loss='binary\_crossentropy',  metrics=['accuracy'] | batch\_size=50, epochs=50 | 71.93% |
| 4 | 60,30,15,1 | optimizer='adam',  loss='binary\_crossentropy',  metrics=['accuracy'] | batch\_size=20, epochs=50 | 88.32% |

**Confusion matrix of the best of the above 4 models is:**



**ROC-Curve for the best ANN model (4):**



**The graph for the accuracy and loss for epoch are as follows:**

