Assignment 3 – sxb180048 – Sai Pratheek Banda

Loan Status Classification

Classification Problem:

The aim of the problem is to predict the classification of the loan status to **'Fully Paid'** and '**Charged OFF'**. The interesting part of this classification is that it will help us understand the patterns for customers if they are going to get "**Charged OFF'**. **Class 0(Fully Paid)**: 28972 Observations| **Class 1 (Charged OFF)**: 7451 Observations

EDA:

1) Features 'Customer ID', 'Loan ID' were insignificant due to its inability to explain the target variable therefore dropped
2) Though the Dataset contains 100000 Records, after dropping the not available rows, we get consistent data of 26423.
3) The Features Term, Home Ownership, Purpose and Target Variable were converted into dummy variables.
4) In the process I have chosen to use F-1 Score as metric because it's of utmost importance that we get the Class 1 classification better than other measures or accuracy.

Data Sampling:

Since the data is heavily unbalanced in terms of the target classification '**Charged OFF',** The consistent data was split into train and test. Due to heavy imbalance and for the quest of a better model I have under sampled the train data to a **Class 0(Fully Paid)**: 7000 Observations| **Class 1 (Charged OFF)**: 5194 Observations.
It is a tradeoff I have chosen to for training the model to the information lost in the process.

Artificial Neural Network:

We are using a Multi-Level Perceptron for our Neural Network with the library Keras to create a Sequential Model.

Parameters:

In search for the best network of Perceptron's we experiment with 4 parameters while other parameters are at default:
i) Neurons ii) Epochs iii) Layers iv) Activation

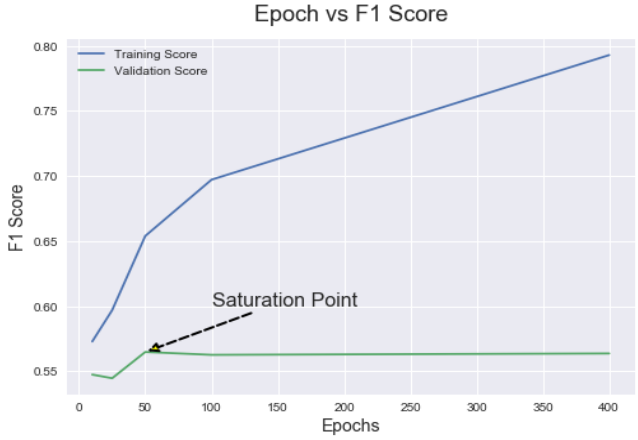
Neurons---------- Experimental Values: [Layer=3,4; Neurons=23,28,33]

As best practice and to avoid underfitting, we choose the number of neurons to be in and around the number of features in the dataset after under sampling and adding dummy variables. Experimenting with a short set of neurons ie [23,28,33] for each hidden layer, number of layers together to find the best outline. The best network was found to be at **(33,28,28)**
First Hidden Layer: 33 Neurons| Second Hidden layer: 28 neurons Third Hidden Layer: 28 Neurons which scored an F-1 Score of **45.73%** Though the F-1 Score is low, it is attributed to the low records of the target variable.

	F1Score	LayerNumber	Neurons
0	0.457305	3	(33, 28, 28)
0	0.454154	3	(33, 33, 23)
0	0.454	4	(23, 28, 23, 28)
0	0.453909	4	(23, 28, 33, 33)
0	0.452754	3	(28, 33, 23)
0	0.452742	4	(23, 28, 33, 23)
0	0.452505	4	(23, 33, 33, 28)
0	0.452499	4	(23, 33, 23, 33)
0	0.451269	3	(28, 23, 28)
0	0.451169	4	(33, 28, 23, 23)

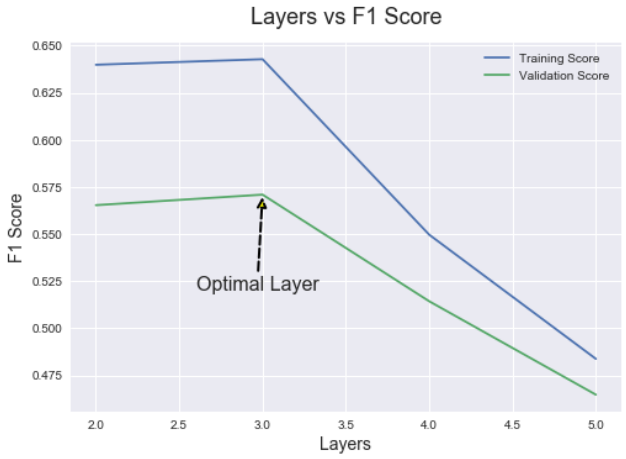
Epochs ---------- Experimental values: [10,25,50,100]----------

Experimenting with multiple Epochs with Cross Validation of 5 Sets, we redistribute weight to each neuron, resulting in a gradual increase in accuracy for both train and validation dataset, but reaches a saturation point at 50 Epochs with an F1 of **56.48%** on validation Data, which implies as to minute change in loss. Though Training sample increase with more iterations, it tends to fit the noise resulting in overfitting. At **50 Epoch** the train and validation seem to model well thus is chosen as the best epoch parameter.



Layers ---------- Experimental Values: [2,3,4,5]----------

An iterative expensive experimentation with Cross Validation of 5 sets was done to find the apt number of layers for the network with 50 epochs, which was found at 3 Hidden Layers, the train and test data take a significant dip of accuracy at 4 ,5 Layers, which implies the increase in loss after 3 Layers due memorization , This result is verified by first parameter experimentation with number of both Neurons and layers.
Thus, we choose the optimum parameter for **Number of Layers as 3** which resulted **57.09%** F1 Score on validation data.

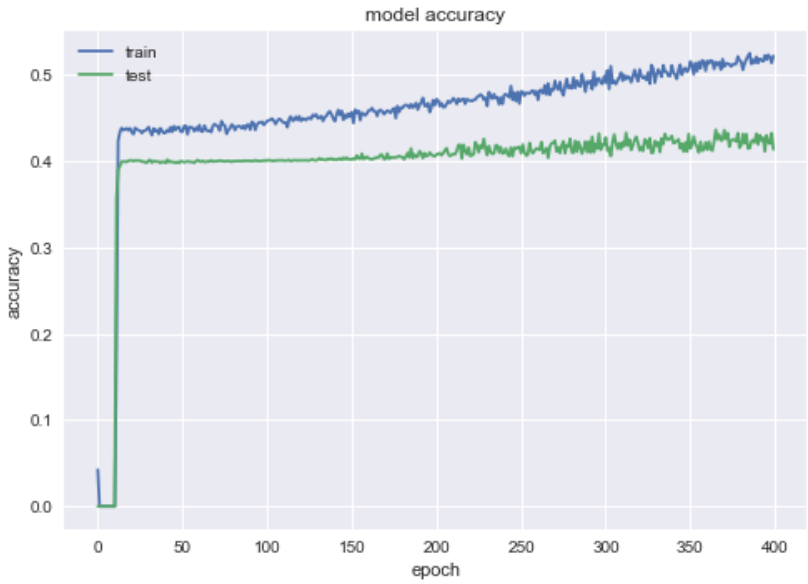


Activation ---------- Experimental Values:[Relu, Sigmoid, Tanh]------

While the other parameters were chosen at its optimal value, experimentation with activation function on both train and validation data were run to find the best activation function to be Tanh. As Tanh is set between [-1,1], it depicts the centered dataset better than sigmoid which makes it easier to learn for the model. Though the Activation of Train is higher in Relu, the validation does not follow it as this activation function is vulnerable o the noise in the data and shows signs of overfitting, while Tanh shows better F1 score with **58.43%** which also models well with validation. Thus, **Tanh** is chosen to be the optimal parameter.

Leaning Curve ----------

The learning curve here depicts the model learning as we increase the number of iterations to resample the weights across the perceptron's, we observe in the graph that the train data and test tends to saturate at 50 epochs, thus the reason we have chosen the value as our optimal value, The train F1 score on train data tends to increase over the following epochs which can be attributed to the noise around which it tends to fit the data. With the validation data the model saturates and does not learn further with increase in epochs.



Train and Test with the Best Parameters ----------

The model fit on the train data and predicted on Train Datset gives an **accuracy of 75% and an F1 Score of 62%** on the **'Charged OFF'** target variable. Test Dataset Preiction gives an **accuracy of 75% and an F1 Score of 46%** on **Charged OFF'**.

Though the accuracy of the model is something which can live by, the F1 score of the model is not commendable , this issue arises due to the high imbalance in the dataset, the undersampling or oversampling does not improve by much, neither does the change of neurons , layers,weights,activation,batch size. Atmost iterations were run on the dataset to find the optimal value.But very low records of the target variable is a major reason for the low performance of the value.

Test Classification Report

	precision	recall	f1-score	support
0	0.86	0.82	0.84	8670
1	0.42	0.50	0.46	2257
accuracy			0.75	10927

Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	7091	1579
Actual 1	1124	1133

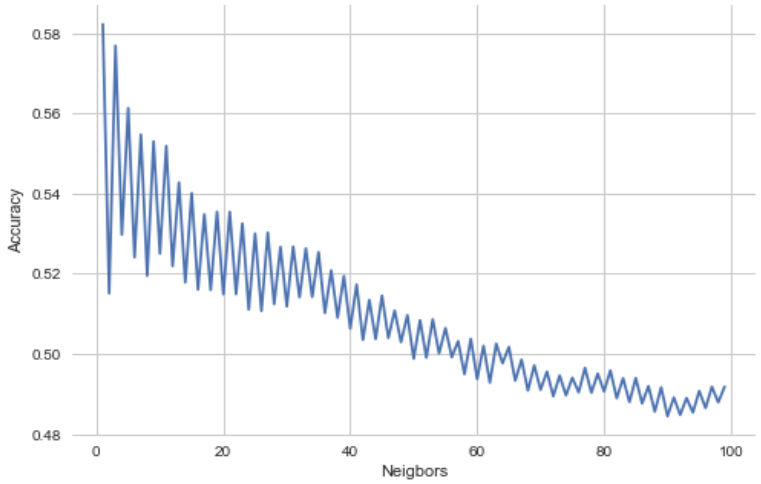
K Nearest Neighbors

Parameters:

In search for the best classification we experiment with 2 parameter i) Neighbors ii) Distance Metric

Neighbors ---------- Experimental Values: [1-100] ----------

The decision of choosing the neighbors is the most of utmost importance to draw clear boundaries, as the best practice suggests an odd value to break the majority vote, we run iterations of the model from 1 to 100 neighbors with the default distance measure. We find the highest accuracy with 1 Neighbor, but we know that having 1 neighbor tends to overfit the data by not having clear boundaries and memorizing the points of train data. Thus, the next best neighbor of 3 is chosen as the optimal.



Distance Metric---------- Experimental Values: [Manhattan, Minkowski, chebyshev]

At the optimal value of 3 Neighbors, the highest accuracy is shown by the 'Minkowski' distance. A Minkowski distance is more generalized distance measure of Euclidian, Manhattan and Chebyshev distance, thus saying that the 'p' value is higher than 1 or 2. As explained above, we are not choosing neighbor as 1 due to overfitting. Cross Validation is implemented not to have bias in our result. We can also verify the neighbors to be three as it shows highest accuracy with 10 values of neighbors.

Model with the best Parameters on Test----------

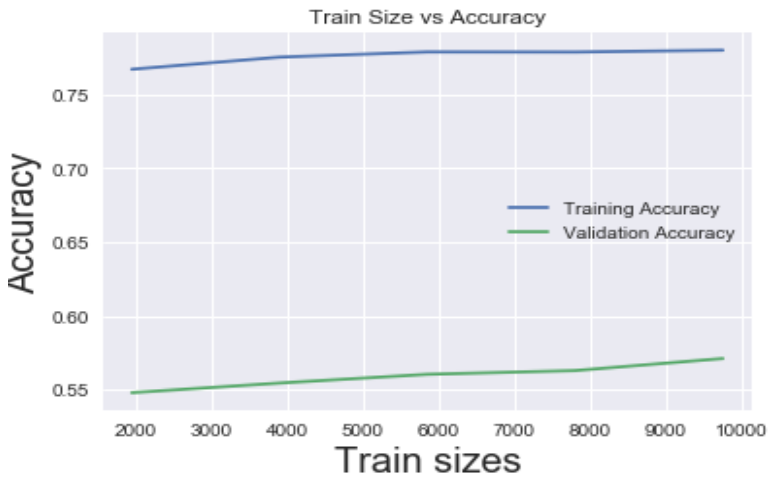
The KNN Classifier is used to classify the data into Loan Status of **Charged OFF' or Fully Paid** , the model does not fare well on the dataset with an accuracy of 69% and an F1 Score of 43% on the target variable. Thus isnt a suitable classifier to choose on this dataset.

```
------------------Classification Report-----------------------
              precision    recall  f1-score   support

           0       0.86      0.73      0.79      8670
           1       0.35      0.55      0.43      2257

    accuracy                           0.69     10927
   macro avg       0.60      0.64      0.61     10927
weighted avg       0.76      0.69      0.72     10927
```

Learning Curve----------

For this model we plot a training curve which results in slow increase in F1 Score up to 57% of train data, the validation data accuracy slow gradual increase as it understands the data better with increase of influx training data.

As learning curve implements cross validation, we can conclude that model trying not to overfit but also does not model well.



---------- *How All the Models Fare* ----------

Model	F1 Score	Comparison
Support Vector Machines	44%	Support Vectors do not fare well when there is more noise in the data which is the case in this dataset as low number of target variables tend to have noise and probability of finding it is high thus does not do well.
Decision Tree	47%	The Decision tree is comprehensive in nature to derive possible outcomes of a decision and traces path to a conclusion which is very important to an unbalanced dataset. It is easy to use and is versatile which is exactly the reason it performs well on this data.
Ensemble Method (Boosting)	47%	The Ensemble method return the same result as that of Decision Tree thus not "boosting" the result. The weak learners though were given high values in further iterations had too less of a number of records to learn and implement.
Artificial Neural Network	46%	Artificial Neural Networks is a design which is suited well for large datasets for which it can learn and assign weight to each perceptron, under sampled data to train on is a great disadvantage going into ANN. Though it does almost as well as Decision Tree, the computational requirement did not validate the results found. The right network to be found is hard to compute as it has high variance in-between.
K Nearest Neighbor	43%	The simplest implementation of all the models which can separate data which is nonlinear. This aside it is very sensitive to noise and unbalanced dataset. The distance measure with an unbalanced dataset deteriorates the classification as it's the only parameter. The learning curve stays with increase in the size of dataset.