**Object:** Predict customer churn in a bank using Deep Learning(Artificial Neural Network)

Description of data:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Non Null count | Data type | Description |
| CustomerId | 10000 | int64 | custmer id of each cusomer |
| Surname | 10000 | object | same of customer |
| CreditScore | 10000 | int64 | Credit score of each customer |
| Geography | 10000 | object | Country of each customer(France, Germany, Spain) |
| Gender | 10000 | object | Gender of customer eithter Male or Female |
| Age | 10000 | int64 | Age of customer |
| Tenure | 10000 | int64 | No of year spent by customer with bank |
| Balance | 10000 | float64 | bank balance of each customer |
| NumOfProducts | 10000 | int64 | No. of products availed by customer |
| HasCrCard | 10000 | int64 | does customer has credit card ( 1 indicates "YES", 0 indicates "NO") |
| IsActiveMember | 10000 | int64 | Present status of customer |
| EstimatedSalary | 10000 | float64 | Salary of customer |
| Exited | 10000 | int64 | Wheather customer stayed with the bank or left the bank ( 0-> "NO", 1-> "YES") |

**Learnings/Outcomes:**

* Handling Imbalance in data
* Building Neural network with hidden layers Created user one defined Function to automate process of Building a neural network with handling imbalance approaches, which resulted in reduction in execution time by 80%
* Visualizing neural network, epoch accuracy,epoch loss using tensorboard, leading to 50% less time in understanding performance.(how to show tensor board in GitHub or online)
* Evaluated model with different metrics

**Project Workflow**  
Preprocessing Data:

* Deleted unnecessary columns
* Converted all text values into numeric data type
* Normalized data using Min Max scalar

Splitting data:

* Splitted data into train and test data with test size as 20%

Model Building:

* Built Deep Learning Neural Network model with 5 layers in total including input, output and Hidden layers

* Model accuracy on train data is 93%

Model Evaluation:

* The above model is evaluated with test data across metrics like Accuracy, precision, recall, f1- score and AUC.
* On comparing metrics, concluded that model is suffering from Imbalance.

Handling Imbalance:

The following approaches have been adopted to handle the Imbalance in data.

* + - * Undersampling
      * Oversampling
      * SMOTE approach

**Conclusion:**

On comparing metrics,  
***The oversampling approach is considered as best one for the given problem,***  
***as it has better accuracy, better AUC, and better f1-score than other approaches***

Scop for improvement:

Dropout layer approach for overfitting

Conclusion: