**CSE1006 PROJECT SUBMISSION Template**

|  |  |
| --- | --- |
| All Roll Numbers of your Team: | 17BCE7009,17BCE7052 |
| Student names in your team : | Kartick Khoker  Nihar Chauhan |
| Contribution by each team member in percentage to complete this project: (Try not to mention equal proportions) | Kartick Khoker-51%  Nihar Chauhan-49% |
| Slot (E2/F1/F2) | F1 |
| Project Title | Santander Customer transaction prediction |
| Did you complete the Project? If No, how much percentage have you done? | Yes |
| Is it a competition (YES/NO)? | Yes |
| If YES, competition link | <https://www.kaggle.com/c/santander-customer-transaction-prediction> |
| Data set used and its link | <https://www.kaggle.com/c/santander-customer-transaction-prediction/data> |
| How many rows and columns in the data set(s) and what are its data types | Rows - 200000  Columns – 403  Data Types- Decimal, String, Integer |
| Describe about your project (What, Why, Final Goal) | In this problem, Santander Bank poses a challenge to Kagglers in order to help them with the problem of identification of the customers who will make a transaction with the bank in future, irrespective of the amount of money transacted previously with the bank. The data set provided is similar to the real data that is available to solve the problem, although the data that is provided to us for solving the problem is masked completely with only numeric values  The data is anonymous with no Customer details been revealed to the participants of the competition. The data sheet contains 200000 rows for both train and test data. The Train Data set has 202 columns with 200 columns having values for var\_1 to var\_200, one column for ID code and one column for target, which are the outcome of the transaction. The same columns are present for test data except for the target. |
| Challenges Identified in your project | Challenging task was to determine the purpose of each column in the dataset so that we could apply pre-processing techniques on it as we didn’t want to lose important data. |
| Pre-processing techniques used | Cleaning, Transformation, Reduction |
| Did your Team spend more time on Pre-processing or for implementing ML algorithms? | Pre-processing as lot of time goes in to decide how to deal with columns, how to optimize the data. |
| What Machine Learning Algorithms Implemented and their accuracies | With linear regression we got, ML Accuracies:0.672  Therefore we switched to LGBM  After Pre-processing(LGBM), ML Accuracies:  0.89983 |
| Which machine learning algorithm you implemented gave highest accuracy and WHY?  (Explain this in very detail) | We used LightGBM. It is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed and efficient with the following advantages:   * Faster training speed and higher efficiency. * Lower memory usage. * Better accuracy. * Support of parallel and GPU learning. * Capable of handling large-scale data.   **Light GBM** grows tree verticallywhile other algorithms grow trees horizontally which means that Light GBM grows tree leaf-wisewhile other algorithms grow level-wise. It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm.  Light GBM is prefixed as ‘Light’ because of its high speed**.** Light GBM can handle the large siz**e** of data and takes lower memory to run. Another reason why Light GBM is popular is because it focuses on accuracy of results. LGBM also supports GPU learning and thus data scientists are widely using LGBM for data science application development. |
| Highest accuracy identified from the Competition site existing submissions and what ML algorithm they have used and how they achieved high performance?  (Explain this in very detail)  (Mention the links of the python code of the earlier competition submissions of the same project from which you have taken inspiration) | Highest accuracy identified by user - 0.9263  He has used Deep NN (**NN wo pseudo, 1 fold seed)**  Deep neural networks solve the problem more globally and can draw conclusions or predictions depending on the information supplied and the desired result. It can solve a problem without a significant amount of marked data.  Deep NN shines when you have excessive amounts of data.  With only a little bit of data it can easily overfit. The big difference between training and test performance shows that your network is overfitting badly. This is likely also because your network model has too much capacity (variables, nodes) compared to the amount of training data. A smaller network (fewer nodes) may overfit less.  Deep neural networks represent the type of machine learning when the system uses many layers of nodes to derive high-level functions from input information. It means transforming the data into a more creative and abstract component.  That’s the reason, the competitor achieved a good accuracy.  **Their project link:**  <https://www.kaggle.com/fl2ooo/nn-wo-pseudo-1-fold-seed>  <https://www.kaggle.com/gunesevitan/santander-customer-transaction-eda-fe-lgb>  <https://www.kaggle.com/gunesevitan/santander-customer-transaction-eda-fe-lgb> |
| Final Conclusion | This was a classification problem on a typically unbalanced dataset with no missing values. Predictor variables are anonymous and numeric and the target variable is categorical. Visualising descriptive features and finally we got to know that these variables are not correlated among themselves. After that we decided to treat imbalanced dataset and built different models with original data and choosen LightGBM as our final model then using the same model we got accuracy of 0.89983 |