Approach used for solving this hackathon

Problem Statement: Predicting Employee Attrition

To aid staffing, you are provided with the monthly information for a segment of employees for 2016 and 2017 and tasked to predict whether a current employee will be leaving the organization in the upcoming two quarters (01 Jan 2018 - 01 July 2018) or not.

Given: A training data set with 19104 observations and 13 features,

Train Data

Variable	Definition
MMMM-YY	Reporting Date (Monthly)
Emp_ID	Unique id for employees
Age	Age of the employee
Gender	Gender of the employee
City	City Code of the employee
Education_Level	Education level : Bachelor, Master or College
Salary	Salary of the employee
Dateofjoining	Joining date for the employee
LastWorkingDate	Last date of working for the employee
Joining Designation	Designation of the employee at the time of joining
Designation	Designation of the employee at the time of reporting
Total_Business_Value	The total business value acquired by the employee in a month (negative business indicates cancellation/refund of sold insurance policies)
Quarterly Rating	Quarterly rating of the employee: 1,2,3,4 (higher is better)

In this dataset target label is missing but LastWorkingDate of employee is there out of which employees which had left the company their last working date is mentioned and for the rest it has null values.

So, I have created a target column which will be used as label for the training dataset and replaced the null values with date 31st December 2017 as we have the data of year 2016 and 2017.

Then I have changed the values in the target column to 1 and 0 where <u>0: if the employee does not leave the organization,</u>
<u>1: if the employee leaves the organization</u>

Then, I have created a new feature Vintage in Days by subtracting target column and date of joining column in the data frame df.

Then, I have read the test dataset given and found that it has 741 observations and only 1 feature Emp_ID

Then I have merged the train and test dataset using inner join and then removed the duplicate observations in the test data frame and saved it in test_new data frame.

Now, test_new data frame has 741 unique observations and 15 features with no null and duplicate entries.

Next, I have removed the columns : 'MMM-YY','Emp_ID','City','Dateofjoining','LastWorkingDate','Target' from test_new data frame

Now, the test_new dataframe has 741 observation with no null and duplicate values and 9 features out of which 7 are numerical and 2 are categorical and no target variable.

Similarly I have remove the columns : 'MMM-YY','Emp_ID','City','Dateofjoining','LastWorkingDate' from training data frame df.

Now, I have 19104 observations and 10 columns out of which 9 are features and 1 Target variable which is label.

Out of the 9 features 2 are categorical variables and 7 are numerical variables.

Then I checked the null values and duplicate values in the training data frame df and found that there are no null values but 2409 duplicate values.

Next, I removed the duplicate values and then I got 15279 observations in which 2 are categorical variables and 8 are numerical variables out of which Target variable is our label and rest are features.

Then I visualised the categorical variables and target variable and got the following insights:

- More male employees are present in the company then females.
- More employees are having Bachelor then master and then college as their level of education.
- By plotting the target variable and then subsequent calculating the values of 0 and 1 I found that there are 13663 employees that did not left the company and only 1616

employees that left the company and here I got the understanding that it is a clear case of Imbalanced classification.

Next I performed distribution plots to numerical variables and found that some are normally distributed.

Next, I plot the correlation heatmap to observe the correlation between different variables.

Next, I used Label encoder to convert categorical variables of both train and test data frame to numerical variables.

Next, I split the training data to features and label.

Then I scaled the features using standard scaler.

Next, I split the training data frame df into train and validation set using train test split keeping validation set as 20% and rest as train.

Since my target class is Imbalanced I have to make the class balanced before training my model as if I did not do that then my model will be trained mostly on one class and will cause the problem of accuracy paradox, to avoid such problem I used the up sampling technique SMOTE

After performing up sampling my target class became balanced and then I fitted the following 3 classification models:

1. Logistic Regression
2. Random Forest classifier
3. XGBoost classifier.

Out of which XGBoost classifier performed best with accuracy of 95% and f1 score of 0.7883.

So I have used XGBoost classifier as my final model for predictions in the test set.