Credit Card Lead Prediction Approach

Use the following steps to train our model.

Step 1: Understand the Problem Statement and the Data given

Firstly go through the problem statement and the data that is given to solve the problem. There will be two sets of data one is to train the model and other is to test the model. The results are to be predicted for the test data. Read the description that is given about each feature and understand the meaning of each feature by searching in google or asking a SME(subject matter Expert) related to the problem field.Understanding the problem and the data is the most important part of predicting.

Step 2: Load the Data and Understand the Data

After understanding the problem and the data given, load the data into an ipython Notebook using pandas for understanding the data. Check the description of each feature and understand whether it is categorical or continuous based on the Description. After loading the data go through each feature and see the dtype(type of feature int/float/object) and compare it to the one type based on the business importance of the feature. If they match there's no problem,else it has to be converted to the respective dtype. Now check the shape of the data for the number of features and the number of instances that are given. For fetching the features present in the data '.columns' can be used to view the features. Now based on the type of feature, divide them into categorical, continuous, Dates, unique Id's(if present).

Step 3: Descriptive statistics and Information

After understanding the data, focus on the information, statistics of the data. We need to understand the statistics of each feature. Statistics include variables such as mean,median,mode,standard deviation,min,max,etc… The mentioned stats are for continuous features. Categorical features have different stats to observe. They are the number of categories, Most occurred category and the frequency of most occurred category. Given data, date features are not there, so we need not go through date stats. It contains the start date,last date,most occurring date, etc…

Information of the data means the missing value content in the data. Missing data may lead to errors so the missing data needs to be identified. There are different ways to get the descriptive stats and information, the most used ones are '.describe()' and '.info'. Check the missing value percentage of features and if the feature has a missing percentage greater than 90 it can be discarded from the data. This assumption is made. And can visualize the missing percentage using plots.

Step 4: Univariate Analysis

After a detailed info and Descriptive stats,Analyze each one of the features using different plots to get meaningful information about each feature individually. Some of the plots are boxplot for outlier analysis, probability distribution curve(KDE) to see how the values are distributed,histogram to count in specific ranges. These are for continuous features. For univariate analysis of categorical features countplot to see the count of different categories,and the distribution of categories using swarmplot,Violin Plot,barplots. After observing each plot try to get information from it. For example, KDE plots the probability distribution if it can be observed. It can be of normally,lognormal.pareto distribution. Based on the observation we can analyze the distribution and try to convert into normal distribution using transformations as normal distribution is most natural occurring in nature. From the count plots of categorical features the most occurring category can be observed and it can replace null values of the feature if any.

Step 5: Bivariate Analysis

After univariate analysis of each feature, Analyze each feature with the output feature to see the relation between them and check the relation between each variable. Some of the plots are scatter plots,pair plots, hexbin plots between continuous features. For categorical and categorical features swarmplot, boxplots,barplots to view the relation between them. Based on the plots meaningful information like relation between features, distributions based on categories, etc.. The relation between the features should be understood as for the prediction. If the feature is not related to the output feature it can be discarded.The relation between the input features is not recommended.we can also analyse the relation using some statistical tests also. For checking the relation between two continuous features Pearson's correlation is used. For checking relation between categorical and

categorical Chi_sqaure test is used and for categorical and continuous ANOVA is used.

Step 6: Data Preparation

Step 6.1: Missing Values and outlier Treatment

Data may contain some missing values and some outliers. In this step data is cleaned and prepared for model training. Feature engineering can also be done. Missing values in the data can be treated using imputation methods like mean imputation,median imputation, mode imputation,Knn imputation. Based on the type of the feature, imputation methods are used. For Continuous if feature contains outliers median is imputed, else mean is imputed. For Categorical features mode imputation can be used. Now for outlier treatment based on the number of outliers and the distribution of features. If the outliers are less we can remove those instances. If they are in more percentage we can see the distribution and transform accordingly. Avg_Account_Balance is like normal_distibtion we can either use BoX_Cox or Log Transform to convert it.We can also impute median in place of that outlier if the distribution of feature is not like normal and pareto distribution.

Step 6.2: Encoding of Categorical Features

After missing values treatment and outliers treatment, the categorical features need to be Encoded as categorical cannot be passed as strings. There are two mostly used Encoding methods, One-Hot Encoding and Label Encoding. One-Hot encoding divides the categories into columns and it is 1 if it is that category, else 0. Label Encoder encodes them into numbers based on the occurrence 1 for 1st occurrence, 2for 2nd occurrence and so on….

Step 7: Model Training

After preparing the data, the model should be trained to predict on test data. Tried on different classification algorithms without using any hyper parameters and got some accuracy from that i saw ensemble model accuracy is better. So I have used Random Forest, Gradient Boosting, ExtremeGradient Boosting, LGBM boosting and Cat Boosting Algorithms to train the model.

Random Forest

As there are more instances, It took a very long time for training. Did a K-fold cross validation technique to do training and have taken parameters like n_estimators,criterion,max_depth,max_feartures. I have tried for 20000 estimators as instances are more. Leaning range 0.02 and max_depth as 6. It gave a Cross validation accuracy of 84.5%. From Random forest we can acquire the feature importance and from that based on the feature importance the least is of gender 0.001 so i discorded it for the training set.
Gradient Boosting(n_estimators=40000,learning_rate=0.02,max_depth=6)
XG Boosting(n_estimators=40000,learning_rate=0.02,max_depth=6)
LGBM Boosting(n_estimators=40000,learning_rate=0.02,max_depth=6)
CAT Boosting(n_estimators=40000,learning_rate=0.02,max_depth=14)

By using the above models I have trained the model.

Step 8: Evaluation

Evolution of the classification models is done using roc_auc score.

Step 9: Test Data

Do the same steps as before and prepare the data done in training the model and then pass through the models trained and predict it based on the test date. The probability is predicted and then I have taken values of XG_boosting, LGBM boosting, CAT_bossting and had an average of three and concluded the results. The average of predicted probabilities are the output from the models.

This is the approach that I have followed to get the accuracy that I got. First I tried on small parameters the accuracy was about 50-60. When I increased the n_estimators the results accuracy increased more.