**Credit Card Approval Prediction**

### **Dataset**

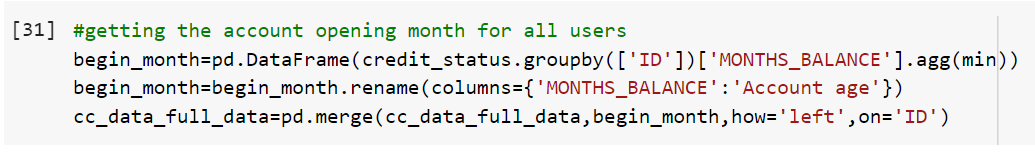
There're two tables could be merged by ID:

| **application\_record.csv** |  |  |
| --- | --- | --- |
| Feature name | Explanation | Remarks |
| ID | Client number |  |
| CODE\_GENDER | Gender |  |
| FLAG\_OWN\_CAR | Is there a car |  |
| FLAG\_OWN\_REALTY | Is there a property |  |
| CNT\_CHILDREN | Number of children |  |
| AMT\_INCOME\_TOTAL | Annual income |  |
| NAME\_INCOME\_TYPE | Income category |  |
| NAME\_EDUCATION\_TYPE | Education level |  |
| NAME\_FAMILY\_STATUS | Marital status |  |
| NAME\_HOUSING\_TYPE | Way of living |  |
| DAYS\_BIRTH | Birthday | Count backwards from current day (0), -1 means yesterday |
| DAYS\_EMPLOYED | Start date of employment | Count backwards from current day(0). If positive, it means the person currently unemployed. |
| FLAG\_MOBIL | Is there a mobile phone |  |
| FLAG\_WORK\_PHONE | Is there a work phone |  |
| FLAG\_PHONE | Is there a phone |  |
| FLAG\_EMAIL | Is there an email |  |
| OCCUPATION\_TYPE | Occupation |  |
| CNT\_FAM\_MEMBERS | Family size |  |

| **credit\_record.csv** |  |  |
| --- | --- | --- |
| Feature name | Explanation | Remarks |
| ID | Client number |  |
| MONTHS\_BALANCE | Record month | The month of the extracted data is the starting point, backwards, 0 is the current month, -1 is the previous month, and so on |
| STATUS | Status | 0: 1-29 days past due 1: 30-59 days past due 2: 60-89 days overdue 3: 90-119 days overdue 4: 120-149 days overdue 5: Overdue or bad debts, write-offs for more than 150 days C: paid off that month X: No loan for the month |

**Feature Engineering**

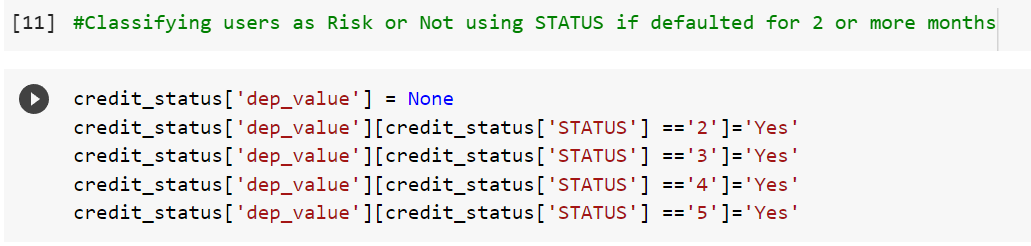
**1: Begin Month:** From the MONTHS\_BALANCE field in credit\_record we can find out how many months ago the user’s account was opened.

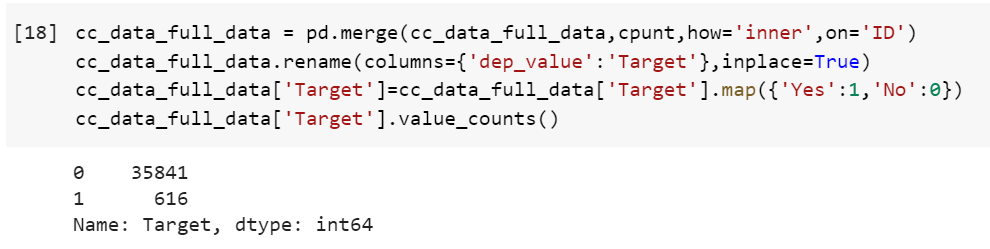


**2:Target:** From the STATUS field we can find out the Credit Card delinquency of all the users.

* Credit card delinquency refers to falling behind on required monthly payments to credit card companies.
* Being late by more than one month is considered delinquent, but the information is typically not reported to credit reporting agencies until two or more payments are missed.
* Delinquent accounts on a credit report can lower credit scores and reduce an individual’s ability to borrow in the future.
* Missing four or five payments likely will move the account into collections, but making just one minimum payment can stop the progression of late payments.

**We can consider users who have defaulted for 2+ months as a Risk. (Target - 1)**

****

****

Data Cleaning and Feature Engineering using Pandas:-

cc\_data\_full\_data=pd.read\_csv('/content/drive/MyDrive/DBDA Final Project/application\_record.csv')

credit\_status = pd.read\_csv('/content/drive/MyDrive/DBDA Final Project/credit\_record.csv')

#getting the account opening month for all users

begin\_month=pd.DataFrame(np.abs(credit\_status.groupby(['ID'])['MONTHS\_BALANCE'].agg(min)))

begin\_month=begin\_month.rename(columns={'MONTHS\_BALANCE':'Account age'})

cc\_data\_full\_data=pd.merge(cc\_data\_full\_data,begin\_month,how='left',on='ID')

#Classifying users as Risk or Not using STATUS if defaulted for 2 or more months

credit\_status['dep\_value'] = None

credit\_status['dep\_value'][credit\_status['STATUS'] =='2']='Yes'

credit\_status['dep\_value'][credit\_status['STATUS'] =='3']='Yes'

credit\_status['dep\_value'][credit\_status['STATUS'] =='4']='Yes'

credit\_status['dep\_value'][credit\_status['STATUS'] =='5']='Yes'

#grouping records from credit status on ID and aggregating using count() to find for how many times the users have defaulted on payment.

cpunt=credit\_status.groupby('ID').count()

cpunt['dep\_value'][cpunt['dep\_value'] > 0]='Yes'

cpunt['dep\_value'][cpunt['dep\_value'] == 0]='No'

cpunt = cpunt[['dep\_value']]

cc\_data\_full\_data = pd.merge(cc\_data\_full\_data,cpunt,how='inner',on='ID')

cc\_data\_full\_data.rename(columns={'dep\_value':'Target'},inplace=True)

cc\_data\_full\_data['Target']=cc\_data\_full\_data['Target'].map({'Yes':1,'No':0})

cc\_data\_full\_data['Target'].value\_counts()

**EDA**:-

**Data Preparation**

Transformations to be done on each feature

* \*\*ID\*\*:Drop the feature
* \*\*CODE\_GENDER\*\*:One hot encoding
* \*\*Age\*\*:Min-max scaling and Fix skewness
* \*\*NAME\_FAMILY\_STATUS\*\*:One hot encoding
* \*\*CNT\_FAM\_MEMBERS\*\*:Fix outliers
* \*\*CNT\_CHILDREN\*\*:Drop feature
* \*\*Housing type\*\*:One hot encoding
* \*\*AMT\_INCOME\_TOTAL\*\*:Remove outliers and Fix skewness and Min-max scaling
* \*\*OCCUPATION\_TYPE\*\*:One hot encoding and Impute missing values
* \*\*Employment status:\*\*One hot encoding
* \*\*NAME\_EDUCATION\_TYPE\*\*:Ordinal encoding
* \*\*Employment length\*\*:Remove outliers and Min-max scaling
* \*\*FLAG\_OWN\_CAR\*\*:Change it numerical and One-hot encoding
* \*\*FLAG\_OWN\_REALTY\*\*:Change it numerical and One-hot encoding
* \*\*FLAG\_MOBIL\*\*:Drop feature
* \*\*FLAG\_WORK\_PHONE\*\*:One-hot encoding
* \*\*FLAG\_PHONE\*\*:One-hot encoding
* \*\*FLAG\_EMAIL\*\*:One-hot encoding
* \*\*Account age\*\*: Drop feature
* \*\*Target\*\*:Change the data type to numerical and balance the data

Dropping ID, FLAG\_MOBIL,Account Age,Children Count

Why are we dropping these features?

* ID: ID is not useful for prediction, it helped us when we were merging the two datasets but after that, there is no need to keep it.
* FLAG\_MOBIL: Since everyone has a mobile phone, this feature does not inform us about anything.
* Children count: is highly correlated with Family member count, and to avoid multicollinearity, we drop it.
* Account age: Because the account age was used to create the target, reusing will make our model overfit. Plus, this information is unknown while applying for a credit card.

1. **Objective of the project: To develop an application using predictive model that can accurately approve or reject credit card applications based on historical data?**
2. **Data used: Dataset was taken from Kaggle. There were two tables, the first one was called application records. It contained personal information like income, age, gender, family status, educational qualification etc. about various ‘users’ or you can say existing customers. The second table was called credit status. It contained monthly information about the credit card status of all the customers. The status column had values like 0,1,2,3,4,5,X,C where 0 meant 1-29 days payment was due for that month, 2 meant 30-60 days due and so on. X meant no bill for that month and C meant all dues were paid on time.The dataset was already clean with just one column in application records containing null values which was later dropped. We did feature engineering to make two new columns. One was the account age which we made from the months column of the credit status table and one the ‘Target’ which we made from the status column of the credit status table. Any customer who made late payments over 2 months were considered as target [1] and rest of the customers were target [0]. The data was highly imbalanced so we used SMOTE to make it balanced.**
3. **Methodology:As this was a binary classification problem we used various classification algorithms like Logistic Regression, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, K-NN ,LDA and Gaussian Naive Bayes.**
4. **Model performance:The decision tree, random forest , KNN, adaboost had accuracy scores over 0.98 and precision and recall were also over 0.95. The AUC-ROC over 0.95. The gradient Boosting model had accuracy score of 0.91and precision and recall of 0.91 and AUC-ROC of 0.98. Performance of LDA, Logistic Regressions and Naive Bayes were comparatively poor. The accuracy scores and other metrics were below 0.75.**
5. **Business implications: This app predicts if an applicant will be approved for a credit card or not. Each time there is a hard enquiry your credit score is affected negatively. This app predicts the probability of being approved without affecting your credit score. This app can be used by applicants who want to find out if they will be approved for a credit card without affecting their credit score.**
6. **Future work: Pipelining can be done for the whole process. A larger dataset with more information could be used for the same application which might give better results. All we did was fit the training data into models and check performance on test data. You can suggest any ML techniques that can be applied like hyperparameter tuning, feature selection etc which will improve the models performance**

**Based on the information provided, the objective of the project was to develop an application that uses a predictive model to accurately approve or reject credit card applications based on historical data. The data used for the project was obtained from Kaggle and included personal information such as income, age, gender, family status, educational qualification, and credit card status of existing customers. Feature engineering was performed to create new columns, including account age and a target column indicating customers who made late payments. The data was highly imbalanced, so SMOTE was used to balance it.**

**Various classification algorithms such as Logistic Regression, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, K-NN, LDA, and Gaussian Naive Bayes were used for the project. The Decision Tree, Random Forest, KNN, and Adaboost algorithms had high accuracy scores and AUC-ROC over 0.95. The Gradient Boosting model had an accuracy score of 0.91 and AUC-ROC of 0.98. However, the performance of LDA, Logistic Regression, and Naive Bayes was comparatively poor.**

**The application can have significant business implications as it can help potential applicants find out the probability of getting approved for a credit card without affecting their credit score. Future work could include pipelining the entire process and using a larger dataset with more information to improve the model's performance. Additionally, techniques such as hyperparameter tuning and feature selection could be applied to enhance the performance of the models. Based on the performance metrics provided, the decision tree, random forest, KNN, and adaboost algorithms could be considered for further analysis and evaluation to choose the most appropriate model for the project.**