Credit Fraud Risk Modeling

Madhumitha Santhana Krishnan
Drexel University
ms5226@drexel.edu

Maopin Yan
Drexel University
my492@drexel.edu

Laura Quinlan
Drexel University
lgq23@drexel.edu

Hristina Shangova
Drexel University
hhs35@drexel.edu

***Abstract*—The aim of this study is to construct an efficient risk prediction model using the dataset shared by Continental Finance Company(CFC) to detect the possible defaults during the credit card application process. The chosen dataset contains the account information of 19030 applicants issued over the recent three years all captured at the time of application process. The dataset has a lot of redundant features and is highly skewed which tends to increase the complexity of the model and can reduce the overall accuracy of the model. Hence, the following feature selection and transformation techniques: F-test, Backward Elimination, Encoding are applied on the raw and preprocessed dataset to extract the meaningful insights. Cross-validation using stratified sampling is applied to split the training and testing data and resampling techniques are used to balance the dataset before applying the classifiers as the dataset is highly imbalanced.**

This paper investigates and checks the performance of Logistic Regression, XGBoost Classifier by applying the following resampling techniques such as Down sampling, Up Sampling and Synthetic Minority Oversampling Technique (SMOTE) across the splits on the data resulted from feature transformation and selection. The performance of these techniques is evaluated based on mean accuracy, precision, recall, f1 score and auc_roc scores. The best metrics are presented in the results for every data output. We were able to achieve a very good accuracy of 97.21% from the XGBoost model implemented on One-Hot and Label encoded dataset Up sampled to 34610 from 19030 rows.

Keywords: Credit Fraud Risk Modeling; Stratified K-fold; Logistic Regression; XGBoost; Up Sampling

I. INTRODUCTION

Financial threats are a growing concern with far reaching consequences in the government, corporate organizations, finance industry. One of the biggest threats faced by commercial banks is the risk prediction of credit clients [1]. Financial institutions such as Continental have to evaluate the credit card risk management to minimize financial losses. Effective management of credit card risk becomes a critical core competence for the long-term success of the financial and banking institutions. Since the factors resulting in credit card defaults are very complex and full of uncertainty, the default risks are difficult to be prevented and controlled in advance [2]. Therefore, it is essential that the company implement a successful credit card risk models to manage fraudulent applicants.

Credit risk modeling is the determination of how likely a consumer will pay back a particular loan by using data about the person. It is crucial for banks and financial institutions to try to decrease the number of consumers' defaults. Continental Finance Company (CFC) is one of America's leading marketers and servicers of credit cards. They presented us with the problem of best predicting the likelihood of an applicant being a fraud risk if issued their $750, $500 or $300 credit limit products. The goal of this project is to build a classifier that helps to classify the applicants as fraud risk or not based on the historical performance on upwards of 100K accounts along with several hundred credit bureau attributes/features that were pulled on the applicants before issuing the card.

II. DATASET DESCRIPTION

The dataset shared by Continental Finance Company(CFC) consisted of two files:

1) *fraud_risk_dataset.csv*: The csv file contains 19030 unique rows which is identified by the unique identifier (record_nb) and 2329 columns. Each row in the csv file gives the information about the year and month of the application (portfolio_id), credit limit product issued (product_term_credit_limit), paid back the balance or not (NP), withdrawn the cash from the card or not (cash_intent) and the rest of the columns are credit bureau attributes(ALJ0300 to TSTU4908) which were pulled for the applicants at the time of application by Experian credit reporting agency. It also includes the account issuance for several months providing an opportunity for seasonal fraud rate comparison.

   Out of the 2329 columns, 1,493 of them are of int datatype while 836 columns are of float datatype making all of the columns numerical. In general, the first 4 columns (portfolio_id to NP) are considered to be categorical features and the credit bureau attributes as continuous valued features except for the features for the data pre-processing techniques. Few exceptions were made based on the context of the technique.

   Notable features in the dataset:

   a) Portfolio_id - This variable is a four-digit number where the first two digits represent the year and the last two digits represent the month.
   b) product_term_credit_limit - The credit limit customers can have on their creditcard. All of Continental users have either a 300, 500, 750, or 1000 dollar credit limit.
   c) NP - NP stands for nonpayment and is the target variable of our dataset. If a customer paid their bill,

NP would be marked 0, if they didn't it would be marked 1.

d) cash_intent - It indicates the customer intention of withdrawing cash after they receive a credit card. When a customer has cash intent they are marked 1, when they don't they are marked 0.

2) *Experian Dictionary3.xlsx*: The Excel file acts as the data dictionary and provides the Name, Description, Units, Length, Valid values, Default value and description of all the features belonging to credit bureau attributes (ALJ0300 to TSTU4908) displayed in the Fraud Risk Dataset csv file.

III. RELATED WORK

Recent studies have shown that artificial intelligence has become a significant part of credit assessment and management. Artificial Intelligence (AI) methods are competitive to the traditional statistical methods for credit assessment [2].

Most of the works that can be found in the literature propose and implement specific algorithms based on artificial intelligence and neural networks to predict and detect the credit card fraudulent transactions [3]. The most relevant study that we found in literature is the credit scoring model(assign credit risk score to determine if a customer is likely to default on the financial obligation) which was published by Yap Bee Wah and Irma Rohaiza Ibrahim, "*Using data mining predictive models to classify credit card applicants*". This paper illustrates the construction and comparison of three credit scoring models: logistic regression (LR) model, classification and regression tree (CART) model and neural network (NN) model to discriminate between rejected and accepted credit card applicants of a bank [4]. Another interesting study that came up is "*An Investigation of Credit Card Default Prediction in the Imbalanced Datasets*," by T. M. Alam et al published in IEEE Access, vol. 8.

The objective of this paper is to train various supervised learning algorithms to predict the client's behavior in paying off the credit card balance. Different resampling techniques were also used to balance the dataset [1].

IV. EXPLORING DATA:

A. Imbalanced Data:

Imbalanced data refers to classification problems where we have unequal instances for different classes of target variable(NP). Having imbalanced data is actually very common in financial datasets. In our dataset, there are 2328 independent variables, and one is a dependent target variable which is NP. We considered all the 2328 variables as X and NP column as Y(label) in our data pre-processing. This dataset is extremely imbalanced where most credit card users are non-defaulters while only very few are defaulters. To generate the bar graph and pie chart (shown in Fig.1), matplotlib python was used by plotting the classes on X-axis and number of applicants on Y-axis.

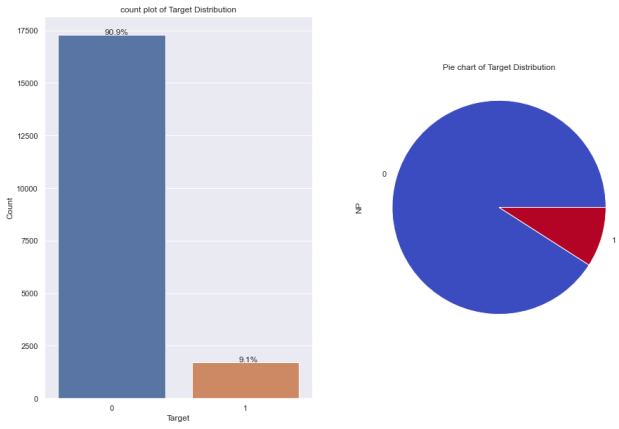

Fig. 1. Target Class Distribution

Only 1725 (or 9.1%) out of 19030 applicants are defaulters. That means the data is highly unbalanced with respect with target variable Class.

B. Null Values:

This dataset has 1,672 null values and all of those values come from 836 columns with 2 rows of missing data as shown in Fig. 2. The missing values on a few features were either handled by filling the null values with 0 or dropped based on the context of the data pre-processing technique used.

	absolute	percent
portfolio_id	0	0.00
product_term_credit_limit	0	0.00
NP	0	0.00
cash_intent	0	0.00
ALJ0300	0	0.00
...	...	...
TSTU3907	2	0.12
TSTU3908	2	0.12
TSTU4906	2	0.12
TSTU4907	2	0.12
TSTU4908	2	0.12

2329 rows × 2 columns

Fig. 2. Percentage of total missing value per column

C. Data Cleaning:

To clean the data, we either dropped the rows or filled with 0's where all the values were missing because they are not useful for our analysis. We also changed the value of the variable 'portfolio_id' by first transforming it into a string and placing a twenty in front of it. Then we used pandas to_datetime function to transform the variable into the datetime data type as shown in Fig.3. Ultimately, we extracted the

year, month, quarter from this value so we could reference them as well as have a dataset with exclusively numeric data types.

```
cc_Data['portfolio_id'] = '20' + cc_Data['portfolio_id'].astype(str)
```

```
cc_Data['portfolio_id'] = pd.to_datetime(cc_Data['portfolio_id'], format='%Y%m')
cc_Data.head()
```

Fig. 3. Python code to convert to date-time format

D. Categorical features:

The following features were considered as categorical for data exploration and analysis: issue_yr, issue_month, Quarter, product_term_credit_limit, cash_intent, ALL9950, ALL9951, ALL6310, ALL6320, MTF6326. Bar charts were created to analyze the data distribution of applicants with respect to year, month, quarter and product term credit limit.

Description of each of the credit bureau attribute code that are included in categorical are provided below with their respective units and valid values.

TABLE I
DESCRIPTION OF CREDIT BUREAU ATTRIBUTES

Name	DESCRIPTION	UNITS	VALID	DEFAULT VALUES AND DESCRIPTIONS
ALL9950	Presence of outstanding federal debts, including default student loans, federal tax lien, unpaid child/family support, or other miscellaneous debts excluding collections including indeterminates	Flag	0-1	99: No trade excluding collections and no public record
ALL9951	Presence of outstanding governmental agency debts, including default student loans, tax lien, unpaid child/family support, or other miscellaneous debts excluding collections including indeterminates	Flag	0-1	99: No trade excluding collections and no public record
ALL6310	Type of trade industry of the oldest opened trade excluding collections including indeterminates (1=MTA, 2=BCA, 3=AUA, 4=RTA, 5=STU, 6=ILN, 7=UTI, 8=CRU, 9=others)	Rank	1-9	99: No trade excluding collections
ALL6320	Type of trade industry of the most recently opened trade excluding collections including indeterminates (1=MTA, 2=BCA, 3=AUA, 4=RTA, 5=STU, 6=ILN, 7=UTI, 8=CRU, 9=others)	Rank	1-9	99: No trade excluding collections
MTF6326	Type of mortgage industry of the most recently opened first mortgage reported in the last 6 months including indeterminates (1=VA, 2=FHA, 3=conventional and others)	Rank	1-3	99: No trade excluding collections

From the plots shown in Fig.4 for various categories, it can be seen from the 1st plot that customers has been steadily increasing from 2019 to 2021. In terms of months in the 2nd plot, more applications were distributed during the month of February which appear distinctively high. Hence, in the 3rd plot, applicants seems to be intuitively high in first quarter. Of all the credit limit products distributed, there were more applicants who have purchased product with credit limit of $300 which can be seen in 4th plot. Most of the customers did not have intention of withdrawing the cash as shown in 5th plot.

Fig.5 displays the plot of all the credit bureau attributes described in Table 1 with categories on the X-axis and the count of applicants for each of those categories on the Y-axis. Plot 1 and 2 shows less number of applicants who have presence of outstanding federal/government debts. Plot 3 and 4 displays the applicants with open trades belonging to which type of trade industry and most of them seems to belong to 2-BCA. The last plot displays the number of applicants with open mortgage belonging to which type of mortgage industry, most of applicants are belonging to 98-Others.

Correlation heatmaps were created on the categorical columns to see their correlation between those features. There does not seem to be strong correlation between the categorical

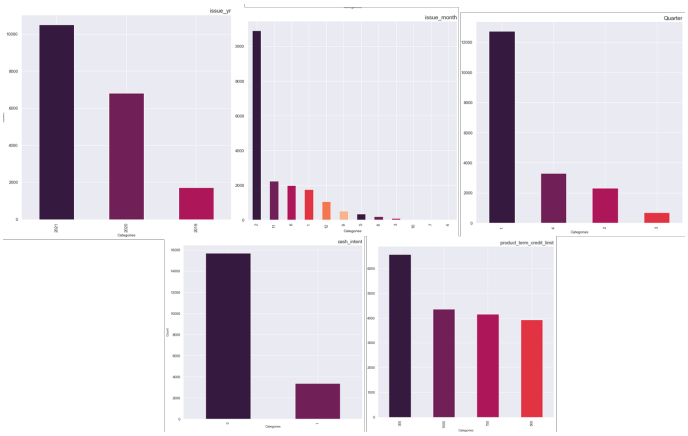

Fig. 4. Categoric Features

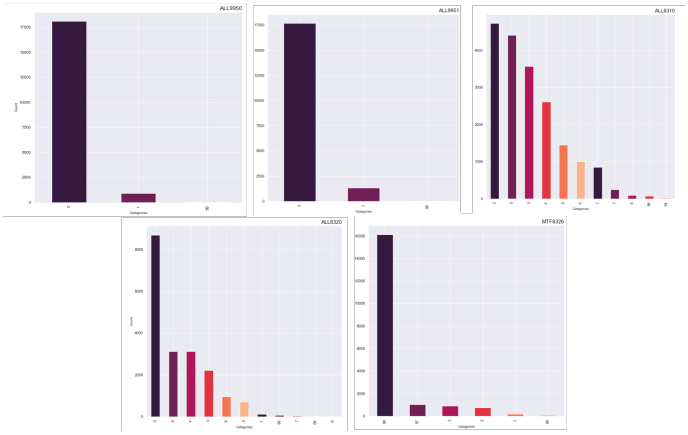

Fig. 5. Categoric Features from credit bureau attributes

features chosen for analysis as displayed in Fig.6 except for few.

E. Continuous Numeric features:

Most of the features in the dataset are continuous numerical values belonging to int and float datatypes. The problem of working with raw, continuous numeric features is that the distribution of values in these features are highly skewed as shown in Figures 7 and 8. Besides this, there is also another problem of the varying range of values in any of these features as seen in Table II [5]. For instance, there are columns with $ and Months of the range 0-999999990 which is abnormally large. Directly using these features can cause a lot of issues and adversely affect the model. Hence, we used some transformation and selection strategies to deal with this kind of data.

F. Bivariate Analysis:

In order to determine the empirical relationship between the categorical variable and the target variable, we performed bivariate analysis by plotting the histogram plots of Not Paid(Fraud) and Paid(Non-Fraud) Vs year, month, product term credit limit, cash intent as shown in Fig.9 and 10.

Fig. 6. Correlation heatmap of Categoric Features

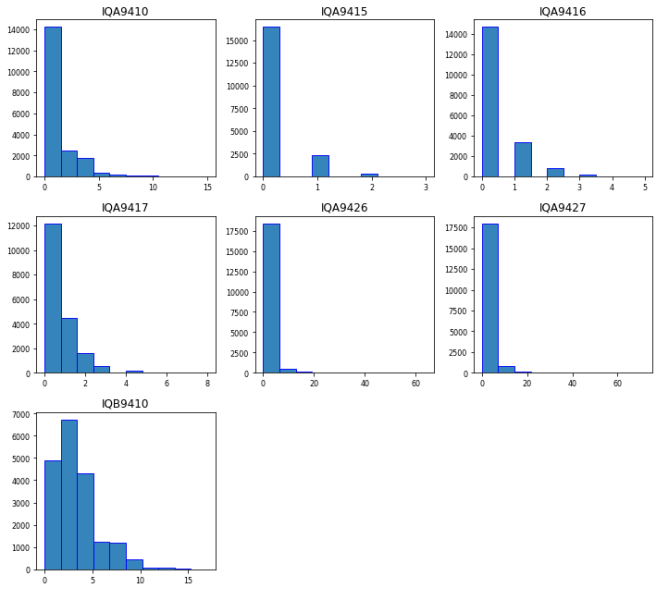

Fig. 7. Distribution plot of Credit bureau attribute with unit of Inquiries

The percentage of fraud vs non-fraud ratio is with respect to the count of the total applicants under the particular category. The interesting insight that once can gain by looking at the graph plotted with month as seen in Fig. 9 plot 2, the applicants who received on the month of June and November did not pay back the balance compared to month March and August where all the applicants paid back the balance. The plot also demonstrates that February is by far the most popular month to sign up for a credit card and very few applicants have not paid back the balance for the products issued in this month.

As displayed in plot 1 of Fig.10, count of Not paid applicants seems to very less in the $750 and $1000 product term credit limits. And as expected, applicants with the lowest product_term_credit_limit ($300) are the most likely to be fraud applicants

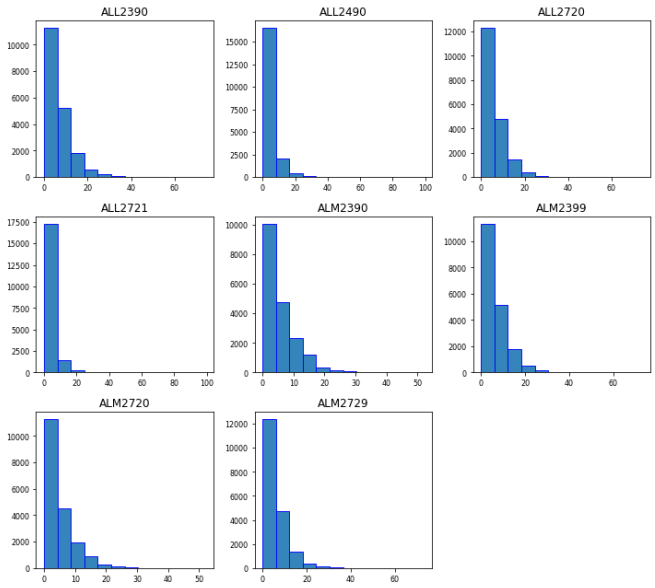

Fig. 8. Distribution plot of Credit bureau attribute with unit of trade

TABLE II
VALID VALUE RANGES OF CREDIT BUREAU ATTRIBUTES ALONG WITH UNITS

Units/Valid Values	Count of Features
$	**648**
0-999999990	578
-999999990-999999990	9
-999999990-0	11
-999999990-999999990	50
%	**224**
0-100	108
0-110	2
0-990	105
-990-0	1
-990-990	8
Days	**29**
0-990	12
0-9990	6
0-99990	11
Flag	**3**
0-1	3
Inquiries	**39**
0-90	39
Months	**391**
0-12	48
0-13	1
0-24	44
0-25	1
0-3	20
0-6	37
0-90	5
0-990	2
0-9990	223
0-99990	2
0-999999990	4
1-12	4
Occurrences	**128**
0-90	33
0-990	83
0-9990	12
PubRec	**29**
0-90	29
Rank	**98**
0-400	95
1-3	1
1-9	2
Trades	**728**
0-90	728
Trades, PubRec	**8**
0-90	8
Grand Total	**2325**

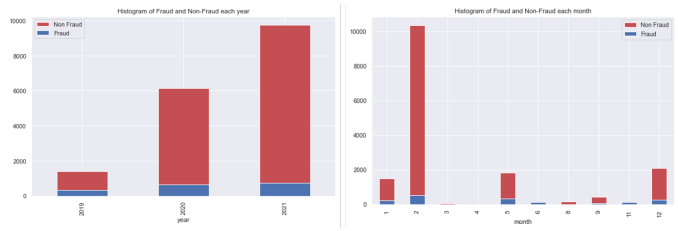

Fig. 9. Histogram plot of year and month with NP

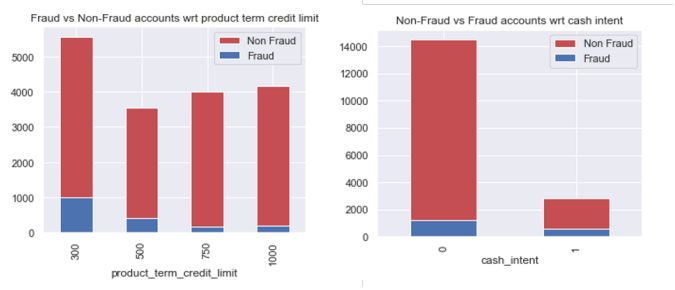

Fig. 10. Histogram plot of product term credit limit and cash intent with NP

V. BACKGROUND:

In this section, a brief description of each of the used data pre-processing and machine learning techniques is presented. In the results section, the accuracy of each classifier/feature combination and the timing performance of each of the classifier will be reported as well.

A. Principal component analysis (PCA):

Principal component analysis (PCA) is fundamental multivariate data analysis method which is used in various area in neural network and machine learning. It is used to reduce the dimensionality of the existing dataset. PCA can be applied to the digit images by projecting the item onto smaller dimension [6]. PCA is an "unsupervised pre-processing task that is carried out before applying any ML algorithm. PCA is based on "orthogonal linear transformation" which is a mathematical technique to project the attributes of a data set onto a new coordinate system. The attribute which describes the most variance is called the first principal component and is placed at the first coordinate. Similarly, the attribute which stands second in describing variance is called a second principal component and so on. In short, the complete dataset can be expressed in terms of principal components. Usually, more than 90% of the variance is explained by two/three principal components. Principal component analysis, or PCA, thus converts data from high dimensional space to low dimensional space by selecting the most important attributes that capture maximum information about the dataset" [7].

B. One-Hot Encoding:

One-hot Encoding is a type of vector representation in which all of the elements in a vector are 0, except for one, which has 1 as its value, where 1 represents a boolean specifying a category of the element [8]. Scikit-learn has OneHotEncoder for this purpose which has been used for this study. All categoric variables are represented by N-1 (N= No of Category and N-1 = drop first of One Hot Coded new feature).

C. Label Encoding:

In this encoding, each category is assigned a value from 1 through N (where N is the number of categories for the feature. Since the credit bureau feature attributes falls within a valid range of values but represented by very large numbers, Scikit-learn Label encoding is used to represent in simple number format that can be better readable by a classifier model.

D. Scaling:

Two kinds of Scaling are used in one of the methods of feature transformation: Scikit-learn's RobustScaler() and StandardScaler(). By using RobustScaler(), we were able to remove the outliers and then use either StandardScaler for prepossessing the dataset.

E. F-Test:

We used p-values and f-values to select features. We selected features with the highest f-values and with the lowest p-values. F-test compares the joint effect of all the variables together, large f-values means the features are significant; features with small p-values mean the results are significantly [9].

F. Backward Feature Elimination:

Backward Feature Elimination can be used to select the important features from dataset [10]. In this project, we do the backward feature elimination after f-test feature selections, because too many features will cost a lot of time in running the results. The steps of backward feature elimination are as follows: Firstly, select the top 4% (93) features after F-test. Secondly, remove some outliers from the 93 features, and we got 87 numerical features. Thirdly, we used backward features elimination to select the 60 most important numerical features.

G. Stratified K-Fold Cross Validation:

A Cross Validation Technique that may be considered as a derived version of the K-Fold Cross Validation Technique. This technique maintains the ratio of each labels in a Fold (any K-1 Dataset) constant. Hence, each fold essentially has the same ratio of each labels. Stratified sampling is implemented with k-fold cross-validation using the 'StratifiedKFold' class of Scikit-Learn [11].

H. Down Sampling:

Down-sampling is one of the effective ways to handle imbalanced data. Downsampling means training on a disproportionately low subset of the majority class examples [12]. In this project, with 1723 (9.1%) fraud to 17305 (90.1%) non-fraud, we randomly down-sampled the 17305 to 1723 non-fraud data. We improved the balance to 1 fraud to 1 non-fraud (50%).

I. Up Sampling:

Up-sampling means adding specific weight to the down-sampled class equal to the factor by the down-sampled. In this project, we used RandomOverSampler[13] to up-sample the minority class (1723 fraud data) to 17305. We improved the balance to 1 fraud to 1 non-fraud (50%).

J. Synthetic Minority Oversampling Technique(SMOTE):

Synthetic Minority Over-sampling Technique (SMOTE) [12] is considered as an effective upsampling algorithm to generate synthetic samples. Different from making copies of existing samples, SMOTE learns the topological properties of the neighbourhood of points in the minority class [14].

K. Logistic Regression:

Logistic regression is a "supervised machine learning classifier that extracts real-valued features from the input, multiplies each by a weight, sums them, and passes the sum through a sigmoid function to generate a probability. A threshold is used to make a decision"[6]. It is the baseline classifiers used widely for classification problems which made it suitable for this study.

L. XGBoost Classifier:

XGBoost stands for "Extreme Gradient Boosting" and it is a supervised learning tree ensemble model that consists of a set of classification and regression trees (CART). XGBoost is an effective machine learning model, even on datasets where the class distribution is skewed [15]. Hence, this classifier is used for this study.

M. Description of metrics used:

All the below metrics was calculated by using scoring parameter on scikit-learn's model_selection.cross_val_scores.

1) *Accuracy*: Accuracy represents the number of correctly classified data instances over the total number of data instances. Accuracy may not be a good measure if the dataset is not balanced (both negative and positive classes have different number of data instances)[16]. Hence, we calculated the other metrices as well in order to evaluate the performance of the classifiers. Accuracy can be represented as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

2) *F-1 score*: F1 is the harmonic mean of precision and recall. F1 is useful in situations where the analysis views precision and recall as equally valuable. The formula for calculating f1 is as follows:

$$\text{F-1} = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN}$$

3) *Precision*: How close measured values are to each other. Measurements can be accurate but not precise and vice versa. Precision can be calculated by dividing true positive by the sum of the true positive and false negative as below:

$$\text{Precision} = \frac{TP}{TP+FP}$$

4) *Recall(sensitivity)*: Recall is the number of values that were classified correctly. The recall values can be calculated by dividing the true positives by the sum of the true positives and false negatives as follows:

$$\text{Sensitivity} = Recall = \frac{TP}{TP+FN}$$

5) *AUC - ROC*: Receiver operating characteristic curve(AUC) curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1 [17].

VI. APPROACH

The cleaned data has been used as an input for all the following transformation and feature selection techniques mentioned below. The target variable 'NP' is dropped before transformation and used while splitting the data using Stratified K-fold Cross Validation technique. The resulted output from each of those approaches are fed into the Logistic Regression and XGBoost Classifier models and performance are measured across the original unsampled imbalanced dataset and the dataset sampled using Up Sampling, Down Sampling and SMOTE resampling techniques as shown in Fig.12:

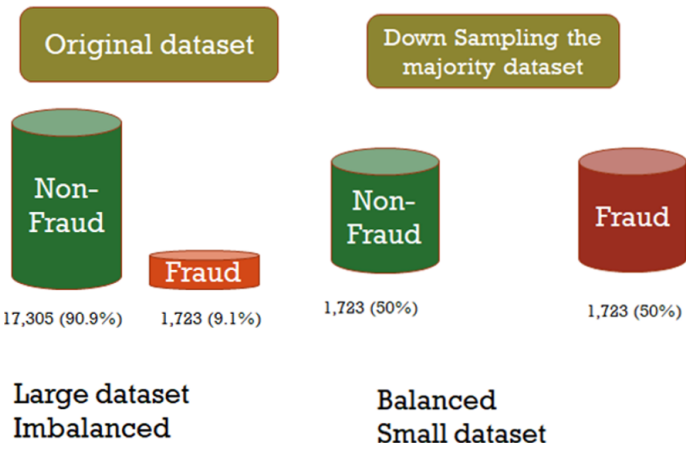

Fig. 11. Down Sampling the majority class

A. First Dataset using One Hot Encoding and Scaler:

The data is segregated into categorical (includes binary integer) and continuous(float and int values features). The categorical features includes the same features considered for our EDA analysis and those features have been one-hot encoded by dropping the first column. As the distribution of values on the continuous features are highly skewed, RobustScaler was used to remove the outliers and then StandardScaler is used for preprocessing the dataset. Both the resulted encoded and transformed dataframes have been concatenated into one

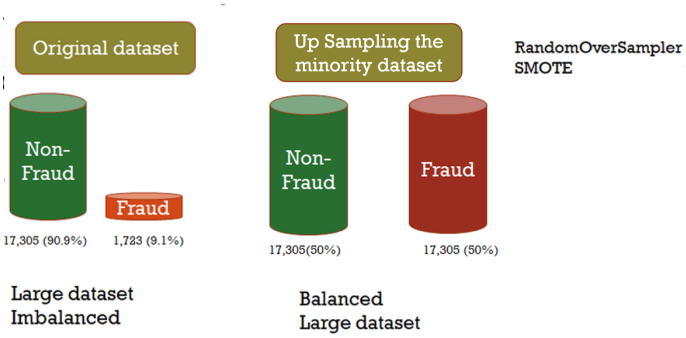

Fig. 12. Up Sampling the minority class

dataframe and consumed for model implementation. Finally, the metrics is evaluated across the splits using mean accuracy, F-1 score, precision, recall and roc_auc for the transformed dataset(imbalanced) as such and also on the dataset upsampled to 34610 rows.

B. Second Dataset using One Hot Encoding and Label Encoding:

The data is segregated into categorical (includes binary integer) and continuous(float and int values features). The categorical features have been encoded using one hot encoding technique similar to first method. Label Encoding is used for the continuous features. Both the resulted encoded dataframes have been concatenated into one dataframe and consumed for model implementation. Finally, the metrics is evaluated across the splits using mean accuracy, F-1 score, precision, recall and roc_auc for the transformed dataset(imbalanced) as such and also on the dataset upsampled to 34610 rows.

C. Third Dataset using Hybrid Feature Selection:

The Data is segregrated into float-valued, integer-valued and categorical (includes binary integer). Categorical features are encoded with one-hot encoding technique. The Correlation heatmap is used to identify the highly correlated float valued features and considered these reduction features for dimensionality reduction using PCA as shown in Fig.13.

The features importance of integer-valued and categorical valued features are visualized using Random Forest and Light GBM plots as shown in Fig.14 and 15.

The integer- valued features, catagorical features, pca transformed vectors for correlated float-valued features, rest of the float-valued features with no correlation have all been combined into one concatenated dataframe for model consumption. The metrics is evaluated across the splits using mean accuracy, F-1 score, precision, recall and roc_auc for the concatenated dataset(imbalanced) as such and also on the dataset upsampled to 34610 rows.

D. Fourth Dataset using F-Test Feature Selection:

The data is segregated into categorical (includes binary integer) and continuous(float and int values features). We used

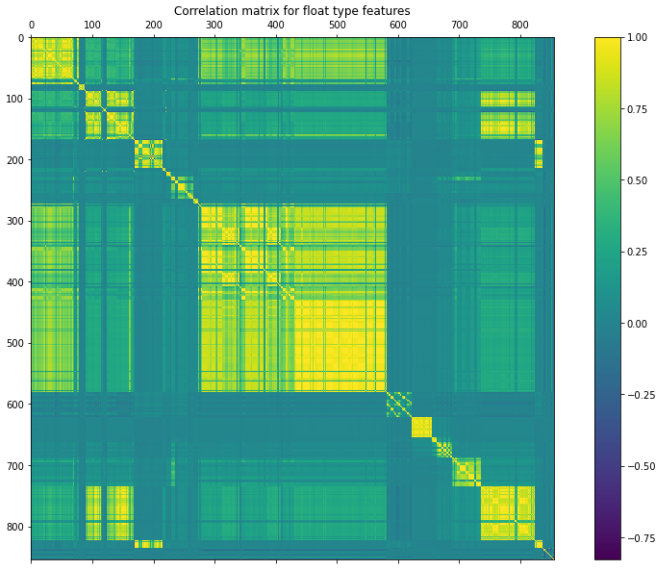

Fig. 13. Correlation Matrix for float-Valued features

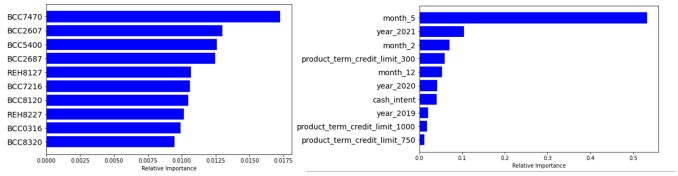

Fig. 14. Random Forest Plot for integer valued and Categorical features

correlation matrix to drop numerical features with correlation coefficients greater than 90%. Then, we used f-test to select 15% features with the highest f-values and the lowest p-values. We put these dataset into modeling. Finally, the metrics is evaluated across the splits using mean accuracy, F-1 score, precision, recall and roc_auc for the transformed dataset(imbalanced) as such and also on the dataset upsampled to 34610 rows.

E. Fifth Dataset using F-Test and Backward Feature Elimination:

The data is segregated into categorical (includes binary integer) and continuous(float and int values features). We used f-test to select 4% of the total features with the highest f-values and lowest p-values. Then we used backward features elimination to select the 60 most important numerical features. After that, we put the 64 features, including the four categorical features, into modeling. Finally, the metrics is evaluated across the splits using mean accuracy, F-1 score, precision, recall and roc_auc for the transformed dataset(imbalanced) as such and also on the dataset upsampled to 34610 rows. F-test features selection is shown in Fig.16.

VII. RESULTS

Table III represents the performance resulted from the implemented models for all the five upsampled datasets. The

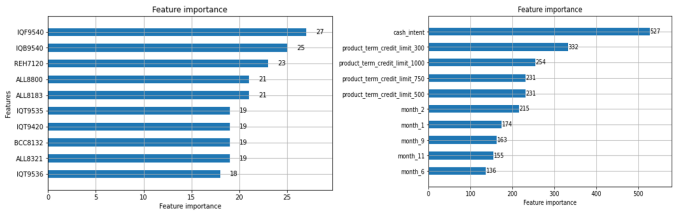

Fig. 15. LightGBM Plot for integer valued and Categorical features

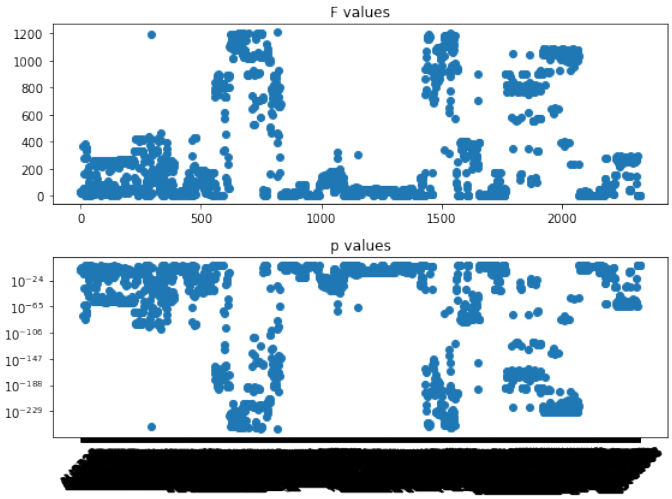

Fig. 16. F-values and P-values for whole features

other resampling techniques experimented did not provide any better results to get represented in the results. The results show that among the two classifiers used, XGBoost model worked the best on all the five up sampled datasets. From Table III, it can also be seen that the XGBoost model implemented on the second Dataset up sampled to 34610 rows (encoded with One Hot Encoding and Label Encoding) performed better when the data is split using Stratified K-fold Cross Validation technique than for the other datasets.

TABLE III
METRICS COMPARISON MATRIX ACROSS THE FIVE APPROACHES

Metrics		Accuracy		F1-Score		Precision		Recall		ROC_AUC	
ML algorithm	Resampling Technique Used	Logistic Regressio n	XGBoost	Logistic Regression	XGBoost	Logistic Regression	XGBoost	Logistic Regressi on	XGBoost	Logistic Regression	XGBoost
Original Dataset with Feature Transformation (Encoding and Scaling)	Up Sampling	78.98%	97.17%	0.79	0.97	78.73%	94.73%	79.43%	99.91%	0.87	0.99
Original Dataset with Feature Transformation (Encoding)	Up Sampling	71.52%	97.21%	0.71	0.97	71.87%	94.78%	70.73%	99.92%	0.78	0.99
Hybrid Feature Selection (2064 features)	Up Sampling	79.09%	96.99%	0.79	0.97	79.03%	94.36%	79.20%	99.97%	0.87	0.99
Correlation and F-Test	Up Sampling	61.33%	94.49%	0.64	0.94	59.56%	90.39%	70.56%	99.56%	0.68	0.98
F-Test and Backward Feature Elimination	Up Sampling	53.29%	92.84%	0.65	0.93	55.39%	87.89%	89.31%	99.36%	0.63	0.97

We have also presented the overall running time of the classifier for splitting the data using Stratified K-fold Cross Validation and to perform the model implementation and to calculate the performance metrics on Table IV for all the five datasets.

TABLE IV
RUNNING TIME OF THE ALGORITHMS

ML algorithm	Resampling Technique Used	Logistic Regression (in seconds)	XGBoost (in seconds)
Original Dataset with Feature Transformation (Encoding and Scaling)	UpSampling	214.32	1163.38
Original Dataset with Feature Transformation (Encoding)	UpSampling	339.23	2880.29
Hybrid Feature Selection (2064 features)	UpSampling	312.98	2396.94
Correlation and F-Test	UpSampling	2.03	32.62
F-Test and Backward Feature Elimination	UpSampling	2.72	32.03

VIII. CONCLUSION

Credit Card defaulters have become more and more rampant in the recent years. To improve the financial risk management level in an automatic, scientific and efficient way, building an accurate, efficient, and easy-handling credit fraud risk modeling system is one of the key tasks of the financial institutions.

In this study, two classifiers were used to evaluate the performance of the dataset created from the fraud risk dataset provided by Continental. The results show that XGBoost is the better classifier for credit fraud risk modeling compared to Logistic Regression. Among all the transformation and feature selection techniques that we used for this study, feature transformation techniques worked better. It can also be seen from the results that the re-sampling technique such as up sampling along with using Stratified K-fold cross validation has greatly increased the performance of the classifier.

IX. FUTURE WORK

One of the specific objectives of this project was to build a classifier to predict the defaulters on the following credit limit products of $300, $500 and $750. In the future, this built classifier can be trained for the specific credit line products and could then be tested on unseen data for the respective credit line products to see how the classifier works. A comparative evaluation of other ensemble algorithms such as Random Forest (Bagging), AdaBoost and LightGBM (Boosting) can also be done using the five datasets created from this study to determine if there are any visible improvement in the performance metrics.

X. APPENDIX

All the code for this project are uploaded to GitHub repository: https://github.com/madhusanthan/Credit-Fraud-Risk-Model

REFERENCES

[1] T.M. Alam; Kamran S; Ibrahim A. Hameed; Suhuai L; Muhammad U S; Shakir S; Jiaming L; Matloob K. An investigation of credit card default prediction in the imbalanced datasets, 2020.

[2] T. Chou. A novel prediction model for credit card risk management, 2007.

[3] A. Arora S. Khatri and A. P. Agrawal. Supervised machine learning algorithms for credit card fraud detection: A comparison, 2020.

[4] Yap Bee Wah and Irma Rohaiza Ibrahim. Using data mining predictive models to classify credit card applicants, 2010.

[5] Dipanjan (DJ) Sarkar. Continuous numeric data., 2019. URL https://towardsdatascience.com/understanding-feature-engineering-part-1-continuous-numeric-data-da4e47099a7b.

[6] V. Singh and S. P. Lal. Digit recognition using single layer neural network with principal component analysis., 2014. URL 10.1109/APWCCSE.2014.7053842.

[7] GeeksforGeeks. Implementing pca in python with scikit-learn., 2021. URL www.geeksforgeeks.org/implementing-pca-in-python-with-scikit-learn/.

[8] David Landup. One-hot encoding in python with pandas and scikit-learn., 2021.

[9] Stephanie Glen. F statistic / f value: Simple definition and interpretation, 2021. URL https://www.statisticshowto.com/probability-and-statistics/f-statistic-value-test/.

[10] Himanshi Singh. Backward feature elimination and its implementation. URL https://www.analyticsvidhya.com/blog/2021/04/backward-feature-elimination-and-its-implementation/.

[11] Anant Kumar. A complete guide to choose the correct cross validation technique., 2020. URL https://medium.com/analytics-vidhya/a-complete-guide-to-choose-the-correct-cross-validation-technique-d70810a02f27.

[12] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[13] The imbalanced-learn developers. Randomoversampler, 2014-2021. URL https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html.

[14] Shiyu Liu, Ming Lun Ong, Kar Kin Mun, Jia Yao, and Mehul Motani. Early prediction of sepsis via smote upsampling and mutual information based downsampling. In *2019 Computing in Cardiology (CinC)*, pages Page–1. IEEE, 2019.

[15] Xgboost 1.5.1 Documentation. Introduction to boosted trees. URL https://xgboost.readthedocs.io/en/stable/tutorials/model.html.

[16] Harikrishnan N B. Confusion matrix, accuracy, precision, recall, f1 score. URL https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd.

[17] Sarang Narkhede. Understanding auc - roc curve. URL https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.