**Project 2**

**Advanced Analysis**

# Personal Loan Modeling

Prediction of whether a customer will respond to a Personal Loan Campaign:

A case study to devise campaigns with better target marketing

*Data based on Thera Bank*

**GROUP 2**

**K.A.I. Kandewatte-s13994**
**S.H. Karunadhika- s13995**
**B.D.S. Suwaris –s14024**

## Abstract

This report is based on the advanced analysis carried out on the Thera Bank, USA dataset sourced from the Kaggle website. The analysis aims to identify the potential clients for personal loans depending on the demographic information (age, income, etc.) and the customer's relationship with the bank (mortgage, securities account, etc.), thereby to identify the most influential predictors to predict whether a customer will respond to a Personal Loan Campaign. Data can be considered as an asset in the process of marketing. By utilizing such data, marketing department of the Thera Bank can begin to build better marketing campaigns in addition to creating stronger product offerings. Univariate and bivariate plots between the response and predictor variables led the pathway to identify the different techniques that could be used for the advanced analysis. The results showed that XGBoost method performs better than other machine learning algorithms with an accuracy of 98.9% and F1 score of 0.99 and 0.95 for class 0 and 1 respectively. The class wise error rates of all other models were higher than that of our final model.

## Table of Content

Page No.

1.Introduction……………………………………………………………………………………………2
2.The Question Description……………………………………………………………………………2
3.Description of the dataset……………………………………………………………………………2
4.Data cleaning ……………………………………………………………………………………………3
5 Important results of the Descriptive Analysis. ………………………………………………3
6.Important results of the advanced Analysis……………………………………………………4
7. Issues Encountered and proposed solutions……………………………………………………8
8. Discussion and conclusion…………………………………………………………………………8
9.Appendix including code and technical details………………………………………………9

## List of figures

5.1.1 Pie chart of personal loan……………………………………………………………………3
5.2.1 Mean income of customers who accepted the personal loan………………………………3
5.3.1 Scatter plot of income by personal loan……………………………………………………3
5.6.1 Boxplot of CCAvg by personal loan..…………………………………………………………3
5.8.1 Bar plot of securities account by personal loan……………………………………………3
5.9.1 Bar plot of customers who transact online……………………………………………………4
5.11.1 Boxplot of mortgage by personal loan………………………………………………………4
5.13.1 Spearman rank correlation plot………………………………………………………………4
6.1.1 Confusion matrix of the final model…………………………………………………………6
6.1.2 ROC curve of the final model…………………………………………………………………6
6.1.3 Variable importance plot of the final model…………………………………………………6
6.1.4 Partial dependency plots of income……………………………………………………………6
6.1.5 Partial dependency plots of CCAvg……………………………………………………………7
6.1.6 Partial dependency plot of Family………………………………………………………………7
6.1.7 Partial dependency plot of Education…………………………………………………………7

## List of tables

1. Table 3.1 Table containing description of the variables…………………………………………2
2. Table 6.1.1 Summarized results of the Logistic Regression with Shrinkage techniques…………5
3. Table 6.1.2 Summarized results of the KNN and SVM models……………………………………5
4. Table 6.1.3 Summarized results of tree based………………………………………………………5
5. Table 6.1.4 Summarized results of fitted models for under-sampling and over-sampling…………5
6. Table 6.1.5 Tuned optimal parameters for XGBoost model…………………………………………6
7. Table 6.1.6 Table of target market……………………………………………………………………7

## 1. Introduction

There is a rising role of marketing in the banking sector and it the function which gets personality and image for bank on its customers' mind. Data can be considered as an asset in the process of marketing. By utilizing such data, marketing department of the Thera Bank can begin to build better marketing campaigns in addition to creating stronger product offerings. This case study focuses on predicting whether a customer will respond to a personal loan campaign to devise campaigns with better target marketing. Unlike a business or a commercial loan, a personal loan is an advance given to an individual for his or her personal use that can be used for large purchases, debt consolidation, emergency expenses and much more. The personal loan market is exploding. According to U.S. web-based lending company Lending Club, personal loan originations from 2017 to 2018 were up 20%. American consumer credit reporting agency TransUnion says personal loans are by far the fastest-growing U.S. consumer-lending category. This explains the competitiveness that prevail in personal loan market. The Thera Bank which is located in USA provides such personal loans and this analysis will provide insights to identify potential clients for loans. In this report, we provide you an advanced analysis on the factors that are associated with the conversion of liability customers of the Thera Bank to personal loan customers while retaining them as depositors.

## 2.The Question Description

Our main attention was paid to the variable "Personal Loan". We, as the marketing department of Thera Bank is concerned about exploring ways of converting liability customers of our bank to personal loan customers while retaining them as depositors. Furthermore, it is important to know which customers are likely to accept a personal loan in order to perform a market segmentation. This will result in lowering of marketing costs and optimization of advertising efforts since we can target the potential highest-yield customers in the next campaigns.
The main objectives of this project are as follows:

1. To identify the potential clients for loans by minimizing loss of resources and opportunities.
2. To construct a suitable model to predict whether a customer will respond to a Personal Loan Campaign based on customer demographic information and the customer's relationship with the bank.

## 3. Description of the Dataset

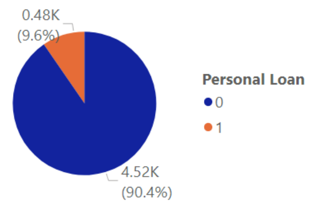
The Thera Bank dataset obtained from Kaggle website is a collection of observations from 5000 customers and 14 variables. The data include customer demographic information (age, income, etc.), the customer's relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan).

| No. | Variable | Description |
|---|---|---|
| 1 | ID | Customer ID |
| 2 | Age | Customer's age in completed years |
| 3 | Experience | Number of years of professional experience |
| 4 | Income | Annual income of the customer ($000) |
| 5 | ZIP Code | Home Address ZIP code. |
| 6 | Family | Family size of the customer |
| 7 | CCAvg | Avg. spending on credit cards per month ($000) |
| 8 | Education | Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional |
| 9 | Mortgage | Value of house mortgage if any. ($000) |
| 10 | Personal Loan | Did this customer accept the personal loan offered in the last campaign? |
| 11 | Securities Account | Does the customer have a securities account with the bank? |
| 12 | CD Account | Does the customer have a certificate of deposit (CD) account with the bank? |
| 13 | Online | Does the customer use internet banking facilities? |
| 14 | CreditCard | Does the customer use a credit card issued by this Bank? |

*Table 3.1- Table -description of the variables*

**4. Data Cleaning**

Data cleaning was performed to correct and remove inaccurate and corrupt data since they can drive the analysis to wrong conclusions. There were no missing values or duplicates in this dataset. The variable "ID" does not add any benefit to the analysis, so it was removed. Some values of the variable "Experience" were negative and they converted them into their absolute values. There were more than 400 unique zip codes available and some zip codes were not valid. Thus, the variable ZIPCode was removed in the advanced analysis. Further, many of the customers have not obtained a housing mortgage and thus had 0 values. So we converted the mortgage variable to a categorical variable such that 1 indicates customers with mortgage and 0 indicates customers without mortgage.



*5.1.1 Pie chart of personal Loan*

**5.Important results of the Descriptive Analysis**

**1.** The conversion rate of the personal loan marketing campaign is 9.6%. By carefully identifying the potential clients for personal loans, marketing costs of the department can be optimized. Also, the dataset obtained is highly imbalanced.

**2** Average income is very low in parts of the country where none of the customers have obtained a personal loan.

**3.** Targeting the wrong market (i.e. customers with low income) will costs a lot for the marketing department. The minimum income requirement of Thera Bank is approximately $50,000 per year.

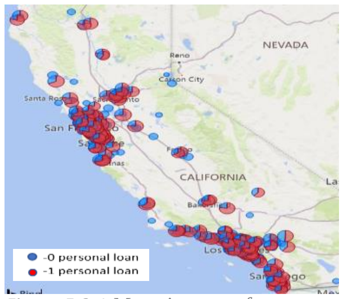


*Figure 5.2.1 Mean income of customers who accepted the personal loan vs. who did not*

**4.** There is an interesting pattern that shows larger families feel the need to get personal loans more. The expenses of large families are high.

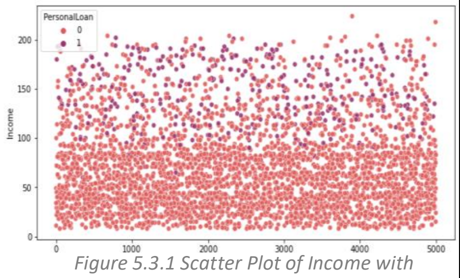


*Figure 5.3.1 Scatter Plot of Income with personal loan*

**5.** The customers with Advanced/professional education have accepted the personal loan more than the undergraduates and graduates. Mainly, the undergraduates do not have a proper source of income to repay the loans. Also, higher the education level the more the customer is confident and open to opt for personal loans.

**6.** The website cnbc.com which provides high-quality personal finance advice states the solution for the high interest rates of credit cards is to use a personal loan to consolidate the credit card debt into one monthly payment. Thus, customers who have a high average spending on Credit Card per month have decided to get a Personal Loan offered in the last campaign.

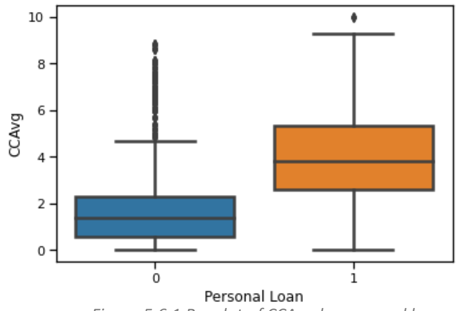
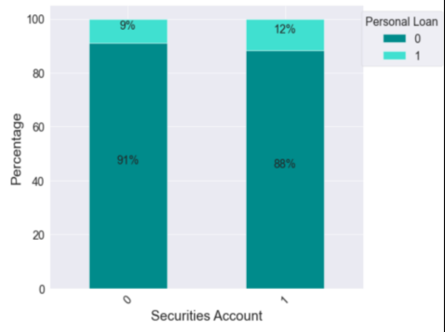


*Figure 5.6.1 Boxplot of CCAvg by personal loan*

**7.** Since CD accounts restrict access to money, customers tend to buy personal loans to satisfy their monetary requirements. Also, the interest rates from CD accounts are much higher which in result aids the customers to repay the loan amount. Thus, a significant percentage i.e. 46.36% of customers who have CD accounts have accepted a personal loan after the last campaign.



**8.** The risks associated with securities accounts demotivate the securities account holders to buy a personal loan since it adds another risk. Thus, most of the customers who have accepted the personal loan do not have a securities account.

*Figure 5.8.1Stacked bar plot of securities account by personal loan*

**9.** According to the author Jamie Gonzalez-Garcia, in 2016, 62% of Americans cited digital banking as their primary method of banking (Bank of America Trends in Consumer Mobility Report 2016). Our analysis provides further evidence to this as 60% of the customers transact online.

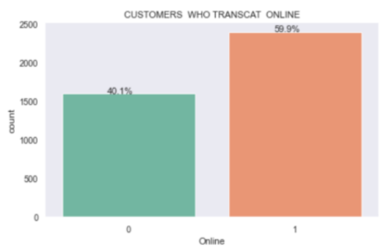


Figure 5.9.1 Bar plot of customers who transact online

**10**. Another fascinating insight that was gained through this analysis is that a greater number of online users have obtained the personal loan than the non-online users. By making use of online platforms to market the personal loans will attract more customers which in result would give a healthy conversion rate.

**11.** The median mortgage amount is relatively high among the customers who have accepted the personal loans. This behavior may be due to the inability to pay the installments for the mortgages, especially among those who have obtained high mortgages.

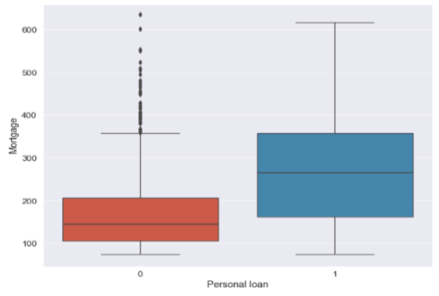


Figure 5.11.1 boxplot of mortgage by personal loan

**12.** The customers with higher incomes have obtained mortgages with higher values. This implies that the house price of the house they are intending to refinance is also high.

**13.** It can be observed that there exist several multicollinearities between some predictors. Especially between the years of experience and Age which indicates almost perfect multicollinearity (0.99). Thus, Logistic Regression with shrinkage methods such as ridge, lasso, elastic net and classification trees such as XGBoost and random forest will be used for the advanced analysis.

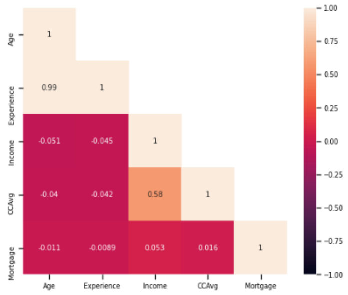


Figure 5.13.1 spearman rank correlation plot

## <u>6. Important results of the Advanced Analysis</u>

Our main objective through this analysis is to identify the potential clients for loans without losing any potential customer opportunity. Thus, we should avoid both loss of resources and loss of opportunity. Therefore, F1-Score which considers both precision and recall is used to select the best model. It's the harmonic mean of the precision and recall. F1 Score is used when we need a balance between precision and recall. Since we need to avoid both the cost of loss of opportunity and loss of resources, F1 score would be the ideal to consider. As the very first step of our advanced analysis, we fitted models without performing any treatment to the prevailing imbalanced dataset.

Achieving 90% classification accuracy, or even 99% classification accuracy, may be trivial on an imbalanced classification problem, misleading the practitioner into thinking that a model has good or even excellent performance when it, in fact, does not. Thus, F1- score will be considered in evaluating the below models. For each of the fitted models, the grid search method with cross validation was used to tune the parameters.

### Logistic Regression with Shrinkage techniques, KNN and SVM

| Model (Logistic) | Accuracy | F1 Score | |
|---|---|---|---|
| | | 0 | 1 |
| Ridge | 0.942 | 0.9680 | 0.6882 |
| Lasso | 0.943 | 0.9689 | 0.6952 |
| Elastic Net | 0.921 | 0.9566 | 0.5587 |

Table 6.1.1-Summarized results of the Logistic Regression with Shrinkage techniques,

Through the descriptive analysis, a strong collinearity between the variables "age" and "experience" was observed. As a remedy to this existing multicollinearity, shrinkage methods such as Ridge, Lasso and Elastic Net were performed. It can be noticed that the misclassification error is high for class 1. This might be due to the dataset being imbalanced.

KNN and Support Vector Machine algorithms were also implemented in our attempt to increase the F1 scores of the model. The grid search method was used to tune the parameters and manhattan metric with 5 n_neighbors and distance as the weights was used in the final tuned KNN model.

| Model | Accuracy | F1 Score | |
|---|---|---|---|
| | | 0 | 1 |
| KNN | 0.938 | 0.9662 | 0.6265 |
| SVM | 0.965 | 0.9807 | 0.8087 |

Table 6.1.2-Summarized results of the KNN and SVM models

## Decision Trees

| Model | Accuracy | F1 Score | |
|---|---|---|---|
| | | 0 | 1 |
| Bagged Decision Trees | 0.9850 | 0.99 | 0.93 |
| Random Forest | 0.9830 | 0.99 | 0.92 |
| Gradient Boosting | 0.9850 | 0.99 | 0.93 |
| XG Boosting | 0.9890 | 0.99 | 0.95 |

Table 6.1.3-Summarized results of tree based

Since our objective is to correctly classify the potential loan customers without any loss in resources and opportunities, we should try to increase the F1 score for both class 1 and class 0. We can observe that the F1 score for both classes have drastically increased for the decision trees.

## Treating imbalanced data

### SMOTE

Oversampling the minority class is one approach to address the imbalanced datasets. The simplest approach involves duplicating examples in the minority class. It is important to notice that these examples don't add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling Technique (SMOTE).

### Random Under Sampling with and without SMOTE

Random under- sampling involves randomly selecting examples from the majority class to delete from the training dataset. Thus, all the above models were refitted by resampling the dataset. The combination of SMOTE and under-sampling performs better than plain under-sampling. N. V. Chawla(2011). Thus, SMOTE with under-sampling technique was also employed in this analysis.

| Model Fitted | SMOTE | | | Under Sampling | | | Under Sampling+SMOTE | | |
|---|---|---|---|---|---|---|---|---|---|
| | | F1 Score | | | F1 Score | | | F1 Score | |
| | Accuracy | 0 | 1 | Accuracy | 0 | 1 | Accuracy | 0 | 1 |
| Logistic Ridge | 0.864 | 0.92 | 0.60 | 0.877 | 0.93 | 0.61 | 0.875 | 0.93 | 0.62 |
| Logistic Lasso | 0.864 | 0.92 | 0.60 | 0.869 | 0.92 | 0.6 | 0.885 | 0.93 | 0.64 |
| Logistic Elastic Net | 0.860 | 0.92 | 0.58 | 0.857 | 0.91 | 0.58 | 0.921 | 0.96 | 0.56 |
| KNN | 0.898 | 0.94 | 0.60 | 0.822 | 0.89 | 0.53 | 0.894 | 0.94 | 0.6 |
| SVM | 0.901 | 0.91 | 0.75 | 0.936 | 0.96 | 0.76 | 0.918 | 0.95 | 0.71 |
| Bagged Decision Trees | 0.974 | 0.99 | 0.88 | 0.984 | 0.98 | 0.75 | 0.972 | 0.98 | 0.88 |
| Random Forest | 0.981 | 0.99 | 0.90 | 0.961 | 0.98 | 0.75 | 0.976 | 0.99 | 0.7 |
| Gradient Boosting | 0.981 | 0.99 | 0.91 | 0.968 | 0.98 | 0.87 | 0.979 | 0.99 | 0.91 |
| XG Boosting | 0.979 | 0.99 | 0.91 | 0.967 | 0.98 | 0.87 | 0.974 | 0.99 | 0.89 |

Table 6.1.4-summarized results of fitted models for under-sampling and over-sampling techniques

## Final Best Model: XGBoost

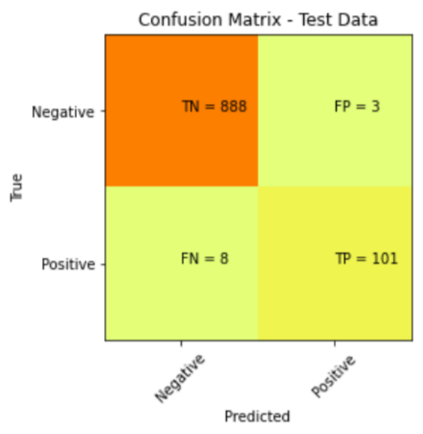


*Figure 6.1.1 confusion matrix of the final model*

| | |
|---|---|
| learning_rate | 0.1 |
| max_depth | 7 |
| n_estimators | 1000 |
| subsample | 1.0 |

*Table 6.1.5-tuned optimal parameters for XGBoost model*

It is worthy to note that, even though our dataset was imbalanced, the tree models obtained without applying any resampling technique performed better than the models which were obtained through SMOTE and under sampling techniques**.** One major disadvantage of under sampling is loss of information. As discussed previously, random under sampling involved randomly selecting examples from the majority class to delete from the training dataset.

According to the confusion matrix of the final model, the marketing team would waste their resources on 0.3% of customers in test data who will not be buying a personal loan. According the False Negatives, 0.8% of potential customers in test data are missed by the sales team. This is loss of opportunity. The purpose of campaign was to target such customers without any loss of resources. If we knew about these customers, we could have offered some good APR /interest rates. By considering all the fitted models, it could be observed that FP and FN are the lowest for XGBoost with high F1 scores for both class 1 and class 0. Thus, it would be our final model. The AUC value of our final model is 0.96 which is quite good. Since mtry=7, 7 variables were used for splitting at each tree node in our final best model.

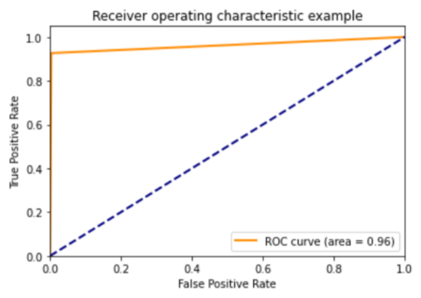


*Figure 6.1.2-ROC curve of the final model*

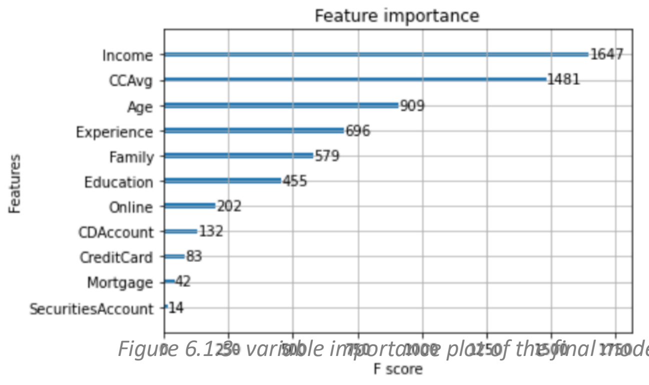


*Figure 6.1.3 variable importance plot of the final model*

Through the Feature Importance Plot, it can be observed that model ranks the variables income and CCAvg more important than the other variables. Also, it ranks the variables family and education as more important variables, aligning to what we observed through our descriptive analysis. Partial dependency plots were drawn for variables to get an insight about how the features affect the loan purchasing ability of customers.

### Income

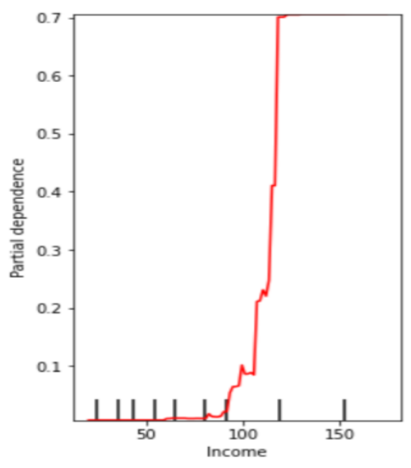


*Figure 6.1.4- pdp of income*

From the partial dependencies Vs income plot, it can be observed that the probability of obtaining a loan is very low for customers with an income ranging from $0 to $50,000. The probability has increased slightly after passing the income mark of $50,000 and also it shows a very high increment after $100,000. As explained in the descriptive analysis according to Forbes Advisor journal, lenders impose income requirements on borrowers to ensure they have the means to repay a new loan. Minimum income requirements vary by lender. The SoFi finance company in USA imposes a minimum salary requirement of $45,000 per year. As per Ascent personal loan statistics, Americans with income over $100,000 are more likely to consider taking out a personal loan than those with lower incomes. Our advanced analysis provides further evidence to these findings.

**Average Monthly Credit Card Spending**

The credit cards have the highest interest rates out of every kind of credit product. As mentioned in the descriptive analysis, website cnbc.com which provides high-quality personal finance advice states the solution for the above high interest rates of credit cards is to use a personal loan to consolidate the credit card debt into one monthly payment. Thus, when the average credit card spending increases, the probability of a customer obtaining a loan has increased. There is also a sudden drop. This may be due to high accumulated credit card average spending; the customers may not be able to afford to repay the personal loans with an interest. Also, when the credit card spending is high there is a high possibility of unpaid bills and high credit card balances. In these circumstances, it is important to refrain targeting such customers since they are unreliable.

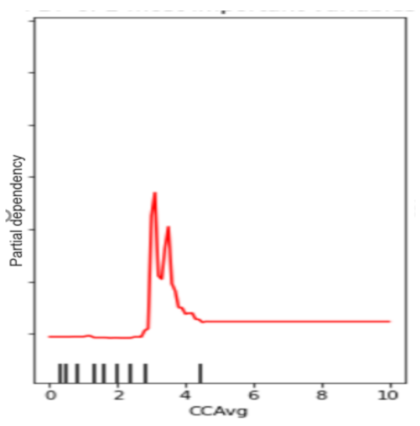


*Figure 6.1.5- pdp of CCAvg*

**Family**

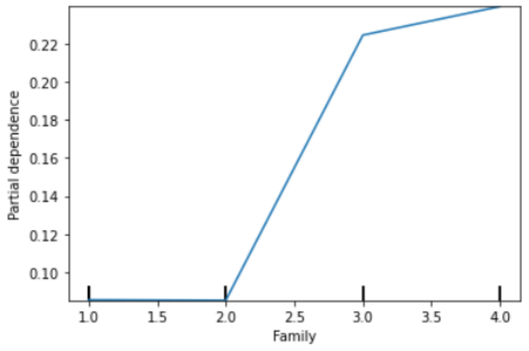


*Figure 6.1.6-pdp of Family*

The family size plays a major role in predicting the potential loan customers. As observed from the partial dependency plot (Fig 6.1.5), when the family size increases, the probability of a customer obtaining a loan also increases. There was an interesting pattern observed through the descriptive analysis that larger families feel the need to get personal loans more. The possible reason could be since the expenses of large families are high.

**Education**

As per the partial dependency plot (Fig 6.1.6), the probability of obtaining a loan differs with the education in a way such that when the education qualification increases, there is a higher chance of a customer obtaining a loan. This pattern was also observed in our descriptive analysis. The possible reason for this can be since the government of USA issues scholarships and grants for undergraduates, they may not find it necessary to buy a personal loan to fund their education (Federal Student Aid Office). Mainly, the undergraduates do not have a proper source of income to repay the loans. Most types of grants, unlike loans, are sources of free money that generally do not have to be repaid. Also, higher the education level the more the customer is confident and open to opt for personal loans.

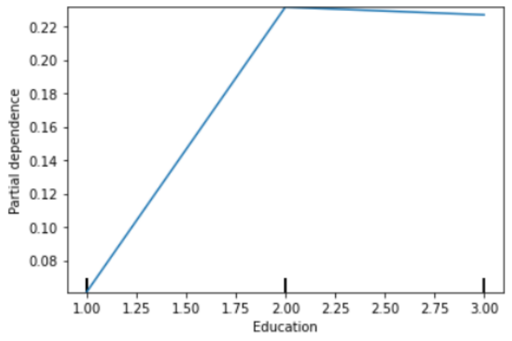


*Figure 6.1.7-Pdp plot of Education*

**Target Market**

For the next campaign, we could identify the high-profile customers which we must target. By targeting the below group of customers, our bank can minimize cost due to wrong marketing and also can avoid loss of opportunity of potential customers and thereby offer them good APR/ Interest Rates.

| Variable | High Profile Customer Ranges |
|---|---|
| Income | Higher than $50,000 |
| CCAvg | $3000-$4000 |
| Education | Advanced/Professional Level and Graduates |
| Family Size | Greater than 2 |

*Table 6.1-Target Market*

## 7. Issues Encountered and proposed solutions

- The variable “Mortgage” only indicated values to customers who have obtained a mortgage and 0 indicated that customer hasn’t obtained a mortgage which was misleading.
  **Solution:** We converted it to a categorical variable with 0 indicating customers without mortgages and 1 indicating customers with mortgages. This also increased the accuracy of our models rather than considering it as a continuous variable.
- Experience variable which indicated number of years of professional experience had negative values.
  **Solution:** Took the absolute value of the variable.
- Some of the zip codes were not valid and there were more than 400 unique zipcodes.
  **Solution:** Since removing observations with invalid zip codes cause loss of information, the variable ZIPCode was removed.
- The dataset was imbalanced with only 9.6% of the customers in the loan obtained class.
  **Solution:** We used under-sampling, SMOTE and under-sampling with SMOTE techniques as a remedy to this issue. In under-sampling it removes the observations from the majority class randomly while SMOTE overcomes imbalances in data by generating artificial data.
- The F1 score for class 1 was low for logistic regression models, KNN and SVM models that were fitted.
  **Solution:** The F1 score of both the classes increased after fitting tree-based models, especially in XGBoost model. According to Jason Brownlee, PhD. XGBoost is an effective machine learning model, even on datasets where the class distribution is skewed.
- The accuracy obtained for imbalanced dataset was higher than what we obtained through techniques such as SMOTE, under-sampling. This was opposed to what we expected.
  Since the source of the dataset was not mentioned, it is susceptible that this dataset is synthetic according to the dataset creator of kaggle.

## 8. Discussion and conclusion

Our main objective of this analysis was to identify the potential clients for loans by minimizing loss of resources and opportunities. Apart, to construct a suitable model to predict whether a customer will respond to a Personal Loan Campaign based on customer demographic information and the customer’s relationship with the bank. The factors income, family size, education, income, CCAvg, Security Account, Mortgage and CD Account were considered in this analysis in order to identify the potential loan customers. Shrinkage methods such as Ridge, Lasso and Elastic Net were fitted due to the strong collinearity that existed between the variables age an experience. Since our main concern was to reduce costs due to loss of resources and opportunities, tree-based models were also fitted. The final best model obtained was XGBoost, which gave an overall accuracy of **98.9%** and F1 score of **0.99** and **0.95** for class 0 and 1 respectively. The models that were obtained after applying resampling techniques such as under-sampling and SMOTE as a remedy to the existing class imbalance had a lower performance than this.

The model ranks the variables income and CCAvg more important than the others. The fascinating relationships and insights that were obtained through the descriptive analysis were further visible in our advanced analysis. Finally, a market segmentation was performed as originally intended by identifying high-profile customers.

## Appendix

```python
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import seaborn as sns
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import SMOTE
from imblearn.pipeline import Pipeline
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import KFold
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.metrics import classification_report,roc_curve,auc
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier,plot_importance
from sklearn.inspection import partial_dependence
from sklearn.inspection import plot_partial_dependence
```

```python
df=pd.read_csv("./archive/Bank_Personal_Loan_Modelling.CSV")

# Remove space in Col names
df.columns = df.columns.str.replace(' ', '')

# Drop ID & Zipcode
df.drop(["ID","ZIPCode"],inplace=True,axis=1)

# Take absolute value of experiance
df['Experience']=df['Experience'].abs()

# Convert mortgage to categorical
df.loc[df['Mortgage'] > 0, 'Mortgage'] = 1

# Change data types
df.PersonalLoan = df.PersonalLoan.astype("category")
df.SecuritiesAccount = df.SecuritiesAccount.astype("category")
df.CDAccount = df.CDAccount.astype("category")
df.Online = df.Online.astype("category")
df.CreditCard= df.CreditCard.astype("category")
df.Mortgage= df.Mortgage.astype("category")

#Splitting
x = df.drop(['PersonalLoan'],axis=1)
y = df['PersonalLoan']

#spliting the data into 80/20
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=100)
```

```python
## To use SMOTE, uncomment below code & replace x_train,y_train with x_sm,y_sm
#smt=SMOTE()
#x_sm,y_sm=smt.fit_resample(x_train,y_train)
#-----------------------------------------------------------------------#
## To use RandomSampling, uncomment below code & replace x_train,y_train with x_rus,y_rus
#rus = RandomUnderSampler()
#x_rus, y_rus = rus.fit_resample(x_train,y_train)
#-----------------------------------------------------------------------#
## To use SMOTE+undersampling, uncomment below code & replace x_train,y_train with x_sumn, y_sumn
#over = SMOTE(sampling_strategy=0.6, random_state=1)
#under = RandomUnderSampler(sampling_strategy=0.9, random_state=1)
#steps = [('o', over), ('u', under)]
#pipeline = Pipeline(steps=steps)
#x_sumn, y_sumn = pipeline.fit_resample(x_train,y_train)
```

```python
def get_metrics(y_test,y_pred):
    print("Accuracy:",accuracy_score(y_test,y_pred))
    print()
    print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
    print()
    print("Classification Report:\n",classification_report(y_test,y_pred,digits=4))
```

```python
#Logistic Regression with Ridge

# define models and parameters
model = LogisticRegression(penalty='l2', random_state=100)
params={"solver": ['newton-cg', 'lbfgs', 'liblinear','sag','saga' ],
        "C": np.logspace(-3, 3, 20)}
cv = KFold(n_splits=10,random_state=100,shuffle=True)
grid_search = GridSearchCV(estimator=model, param_grid=params, n_jobs=-1, cv=cv,
                           scoring='accuracy',error_score=0)
grid_result = grid_search.fit(x_train, y_train)

# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))

# fit using tuned parameters
lrr=grid_result.best_estimator_
lrr.fit(x_train, y_train)
y_pred=lrr.predict(x_test)

get_metrics(y_test,y_pred) #get performance metrics
```

```python
#Logistic Regression with Lasso

# define models and parameters
model = LogisticRegression(penalty='l1',random_state=100)
params={"solver": ['liblinear','saga'],
        "C": np.logspace(-3, 3, 20)}
cv = KFold(n_splits=10,random_state=100,shuffle=True)
grid_search = GridSearchCV(estimator=model, param_grid=params, n_jobs=-1, cv=cv,
                           scoring='accuracy',error_score=0)
grid_result = grid_search.fit(x_train, y_train)

# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))

# fit using tuned parameters
lrl=grid_result.best_estimator_
lrl.fit(x_train, y_train)
y_pred=lrl.predict(x_test)

get_metrics(y_test,y_pred) #get performance metrics
```

```python
#Logistic Regression with Elasticnet

# define models and parameters
model = LogisticRegression(penalty='elasticnet',solver='saga',random_state=100)
params={"l1_ratio":[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1],
        "C": np.logspace(-3, 3, 20)}
cv = KFold(n_splits=10,random_state=100,shuffle=True)
grid_search = GridSearchCV(estimator=model, param_grid=params, n_jobs=-1, cv=cv,
                           scoring='accuracy',,error_score=0)
grid_result = grid_search.fit(x_train, y_train)

# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))

# fit using tuned parameters
lre=grid_result.best_estimator_
lre.fit(x_train, y_train)
y_pred=lre.predict(x_test)

get_metrics(y_test,y_pred) #get performance metrics
```

```python
#KNN

# define models and parameters
model = KNeighborsClassifier()
params={"n_neighbors":range(1, 21),
        "weights":['uniform', 'distance']
        "metric":['euclidean', 'manhattan', 'minkowski']}
cv = KFold(n_splits=10,random_state=100,shuffle=True)
grid_search = GridSearchCV(estimator=model, param_grid=params, n_jobs=-1, cv=cv,
                           scoring='accuracy',error_score=0)
grid_result = grid_search.fit(x_train, y_train)

# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))

#fit using tuned parameters
knncl = grid_result.best_estimator_
knncl.fit(x_train, y_train)
y_pred=knncl.predict(x_test)

get_metrics(y_test,y_pred) #get performance metrics
```

```python
#SVM

# define models and parameters
model = SVC()
params={"kernel":['linear','poly', 'rbf', 'sigmoid'],
        "C":[50, 10, 1.0, 0.1, 0.01],
        "gamma":['scale']}
cv = KFold(n_splits=10,random_state=100,shuffle=True)
grid_search = GridSearchCV(estimator=model, param_grid=params, n_jobs=-1, cv=cv,
                           scoring='accuracy',error_score=0)
grid_result = grid_search.fit(x_train, y_train)

# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))

#fit using tuned parameters
svcl=grid_result.best_estimator_
svcl.fit(x_train,y_train)
y_pred=svcl.predict(x_test)

get_metrics(y_test,y_pred) #get performance metrics
```

```python
#Random Forest

# define models and parameters
model = RandomForestClassifier()
params={"n_estimators":[10, 100, 1000],
        "max_features":['sqrt', 'log2']}
cv = KFold(n_splits=10,random_state=100,shuffle=True)
grid_search = GridSearchCV(estimator=model, param_grid=params, n_jobs=-1, cv=cv,
                           scoring='accuracy',error_score=0)
grid_result = grid_search.fit(x_train, y_train)

# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))

#fit using tuned parameters
rfcl = grid_result.best_estimator_
rfcl.fit(x_train, y_train)
y_pred=rfcl.predict(x_test)

get_metrics(y_test,y_pred) #get performance metrics
```

```python
#Gradient Boosting

# define models and parameters
model = GradientBoostingClassifier()
params={"n_estimators":[10, 100, 1000],
        "learning_rate":[0.001, 0.01, 0.1],
        "subsample":[0.5, 0.7, 1.0],
        "max_depth":[3, 7, 9]}
cv = KFold(n_splits=10,random_state=100,shuffle=True)
grid_search = GridSearchCV(estimator=model, param_grid=params, n_jobs=-1, cv=cv,
                           scoring='accuracy',error_score=0)
grid_result = grid_search.fit(x_train, y_train)

# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))

#fit using tuned parameters
gradcl=grid_result.best_estimator_
gradcl.fit(x_train, y_train)
y_pred=gradcl.predict(x_test)

get_metrics(y_test,y_pred) #get performance metrics
```

```python
le=LabelEncoder()
cat_cols=["SecuritiesAccount","CDAccount","Online","CreditCard","Mortgage"]
x_train[cat_cols]=x_train.loc[:,cat_cols].apply(lambda col : le.fit_transform(col))
x_test[cat_cols]=x_test.loc[:,cat_cols].apply(lambda col : le.fit_transform(col))
```

```python
#XGBoost

# define models and parameters
model = XGBClassifier()
params={"n_estimators":[10, 100, 1000],
        "learning_rate":[0.001, 0.01, 0.1],
        "subsample":[0.5, 0.7, 1.0],
        "max_depth":[3, 7, 9, 10, 12]}
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
grid_search = GridSearchCV(estimator=model, param_grid=params, n_jobs=-1, cv=cv,
                           scoring='accuracy',error_score=0)
grid_result = grid_search.fit(x_train, y_train)

# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))

#fit using tuned parameters
xgbcl=XGBClassifier(learning_rate= 0.1,
                    max_depth= 7,
                    n_estimators=1000,
                    subsample= 1.0,
                    use_label_encoder=False)
xgbcl.fit(x_train, y_train)
y_pred=xgbcl.predict(x_test)

get_metrics(y_test,y_pred) #get performance metrics
```

```python
# More Informative Confusion Matrix for Best Model(XGB)
plt.clf()
confusion_matrix=confusion_matrix(y_test,y_pred)
plt.imshow(confusion_matrix, interpolation='nearest', cmap=plt.cm.Wistia)
classNames = ['Negative','Positive']
plt.title('Confusion Matrix - Test Data')
plt.ylabel('True')
plt.xlabel('Predicted')
tick_marks = np.arange(len(classNames))
plt.xticks(tick_marks, classNames, rotation=45)
plt.yticks(tick_marks, classNames)
s = [['TN','FP'], ['FN', 'TP']]
for i in range(2):
    for j in range(2):
        plt.text(j,i, str(s[i][j])+" = "+str(confusion_matrix[i][j]))
plt.show()
```

```python
# ROC Curve
fpr, tpr, thresholds = roc_curve(y_test, y_pred)
a=auc(fpr,tpr)
plt.figure()
lw = 2
plt.plot(fpr, tpr, color='darkorange',
         lw=lw, label='ROC curve (area = %0.2f)' % a)
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()
```

```python
plot_importance(xgbcl, max_num_features=10) # top 10 most important features
plt.show()
```

```python
# Partial Dependence Plots
features = ['Income', 'CCAvg', ('Income', 'CCAvg')]
fig, ax = plt.subplots(figsize=(12, 6))
ax.set_title("PDP of 2 most important variables")
mlp_disp = plot_partial_dependence(xgbcl, x_train, features, ax=ax,
                                   line_kw={"color": "red"})
```

```python
plot_partial_dependence(xgbcl, x_train, ["Family"])
plot_partial_dependence(xgbcl, x_train, ["Education"])
```