Laboratory Practice 1 :
Data Analytics (DA) Mini Project
Bank Customer Churn Prediction

PID 26

Project created by :
Ashutosh Srivastava (29)
Vaibhav Bhavsar (30)
Vaibhav Chalse (31)
Vinay Deokar (32)

Project Guide: Prof. S. M. Malao

December 14, 2020

Contents

List of Figures

List of Tables

Chapter 1

Problem Statement

A Bank wants to take care of customer retention for its product: savings accounts. The bank wants us to identify customers likely to churn balances below the minimum balance. We have the customer's information such as age, gender, demographics along with their transactions with the bank.

Chapter 2

Motivation

As we know, it is much more expensive to sign in a new client than keeping an existing one.It is advantageous for banks to know what leads a client towards the decision to leave the company. Churn prevention allows companies to develop loyalty programs and retention campaigns to keep as many customers as possible.

Given the importance of customers as the most valuable assets of organizations, customer retention seems to be an essential, basic requirement for any organization. Banks are no exception to this rule. The competitive atmosphere within which electronic banking services are provided by different banks increases the necessity of customer retention.

Chapter 3

Objectives

- Analyzing the dataset by using data preprocessing techniques and libraries.
- Impute the missing values in some variables using data pre-processing techniques.
- Scaling the numerical features using data pre-processing techniques.
- Analyzing the dataset by plotting various graphs and visual representations.

Chapter 4

Outcomes

- A Machine Learning Model to predict propensity to churn for each customer is developed.
- Implemented 2 ML algorithms for better efficiency score.
- Random Forest model is giving the best result for each fold after cross validation.

Chapter 5

DataSet

5.1 DataSet Details

There are multiple variables in the dataset which can be cleanly divided into 3 categories:

- Demographic information about customers
    1. customer_id - Customer id
    2. vintage - Vintage of the customer with the bank in a number of days
    3. gender - Gender of customer
    4. age - Age of customer
    5. dependents - Number of dependents
    6. occupation - Occupation of the customer
    7. city - City of the customer (anonymized)
- Customer Bank Relationship
    1. customer_nw_category - Net worth of customer (3: Low 2: Medium 1: High)
    2. custome ° _nw_category - Net worth of customer (3: Low 2: Medium 1: High)
    3. days_since_last_transaction - No of Days Since Last Credit in Last 1 year
- Transactional Information
    1. current_balance - Balance as of today
    2. previous_month_end_balance - End of Month Balance of previous month
    3. average_monthly_balance_prevQ - Average monthly balances (AMB) in Previous Quarter

4. average_monthly_balance_prevQ2 - Average monthly balances (AMB) in previous to the previous quarter
5. current_month_credit - Total Credit Amount current month
6. previous_month_credit - Total Credit Amount previous month
7. current_month_debit - Total Debit Amount current month
8. previous_month_debit - Total Debit Amount previous month
9. current_month_balance - Average Balance of current month
10. previous_month_balance - Average Balance of previous month
11. churn - Average balance of customer falls below minimum balance in the next quarter (1/0)

- Source of the dataset: Kaggle

Chapter 6

Software and Hardware Requirements

6.1 Software Requirements

1. Python Version 3.8
2. Anaconda Navigator
3. Jupyter Notebook

6.2 Hardware Requirements

1. OS : Windows 10 / Linux
2. Processor : i3 5^{nd} Gen (Min)
3. RAM : 4 GB (Min)

Chapter 7

Libraries / Frameworks Used

1. Numpy
2. Pandas
3. Matplotlib
4. Seaborn
5. Scikit Learn

7.0.1 Numpy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

7.0.2 Pandas

pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way toward this goal.

7.0.3 Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib produces publication-quality figures in a variety of

hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers, and various graphical user interface toolkits.

7.0.4 Seaborn

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures. Here is some of the functionality that seaborn offers:
A dataset-oriented API for examining relationships between multiple variables Specialized support for using categorical variables to show observations or aggregate statistics
Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data
Automatic estimation and plotting of linear regression models for different kinds dependent variables
Convenient views onto the overall structure of complex datasets
High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations
Concise control over matplotlib figure styling with several built-in themes
Tools for choosing color palettes that faithfully reveal patterns in your data
Seaborn aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

7.0.5 Scikit Learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.
is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.
The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn.

Chapter 8

Block Diagram

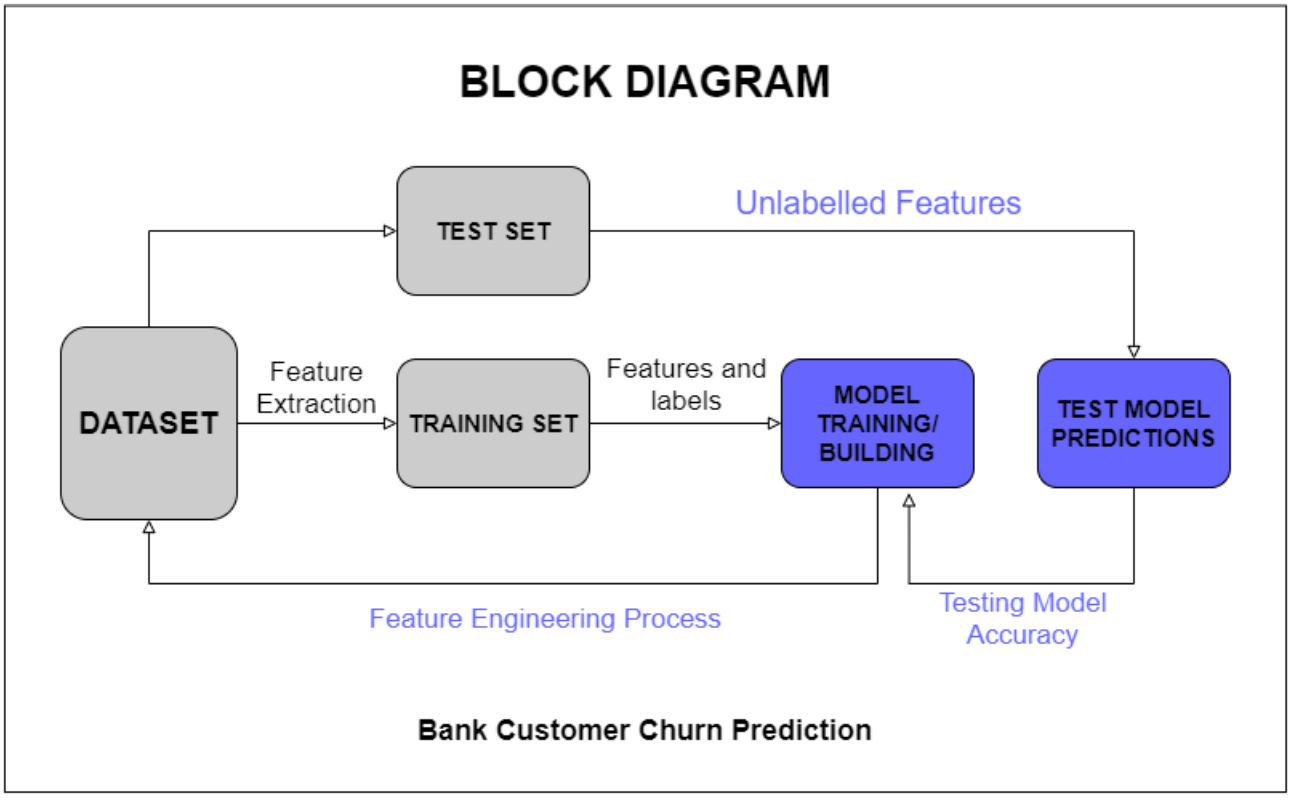

Figure 8.1: Block Diagram

Chapter 9

Algorithms Used

1. Logistic Regression
2. Random Forest

9.1 Logistic Regression

Logistic regression is a statistical analysis method used to predict a data value based on prior observations of a data set. Logistic regression has become an important tool in the discipline of machine learning. The approach allows an algorithm being used in a machine learning application to classify incoming data based on historical data. As more relevant data comes in, the algorithm should get better at predicting classifications within data sets. Logistic regression can also play a role in data preparation activities by allowing data sets to be put into specifically predefined buckets during the extract, transform, load (ETL) process in order to stage the information for analysis.

A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. For example, a logistic regression could be used to predict whether a political candidate will win or lose an election or whether a high school student will be admitted to a particular college.

The resulting analytical model can take into consideration multiple input criteria. In the case of college acceptance, the model could consider factors such as the student's grade point average, SAT score and number of extracurricular activities. Based on historical data about earlier outcomes involving the same input criteria, it then scores new cases on their probability of falling into a particular outcome category.

Logistic regression is one of the most commonly used machine learning algorithms for binary classification problems, which are problems with two class values, including predictions such as "this or that," "yes or no" and "A or B." The purpose of logistic regression is to estimate the probabilities of events, including determining a relationship between features and the probabilities of particular outcomes. One example of this is predicting if a student will pass or fail an exam when the number of hours spent studying is provided as a feature and the variables for the response has two values: pass and fail.

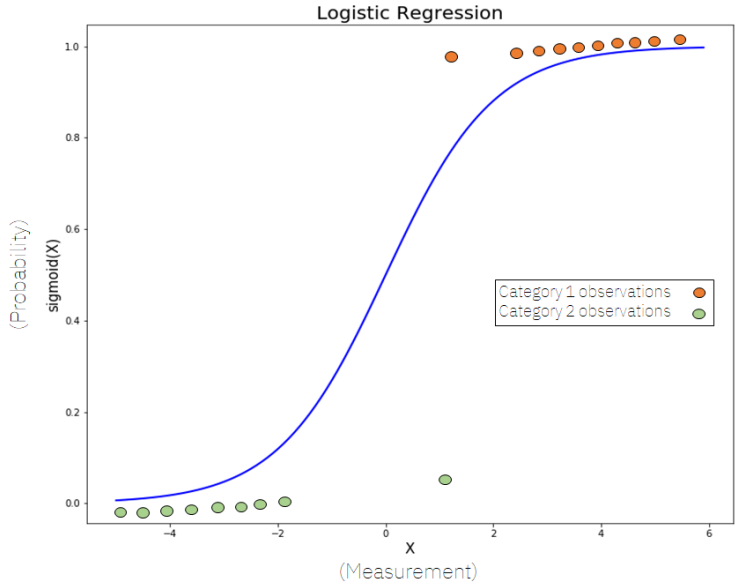

Figure 9.1: Logistic Regression

9.2 Random Forest Classifier

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result.

Put simply: random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Let's look at random forest in classification, since classification is sometimes considered the building block of machine learning. Below you can see how a random forest would look like with two trees:

Random forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, there's no need to combine a decision tree with a bagging classifier because you can easily use the classifier-class of random forest. With random forest, you can also deal with regression tasks by using the algorithm's regressor.

Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

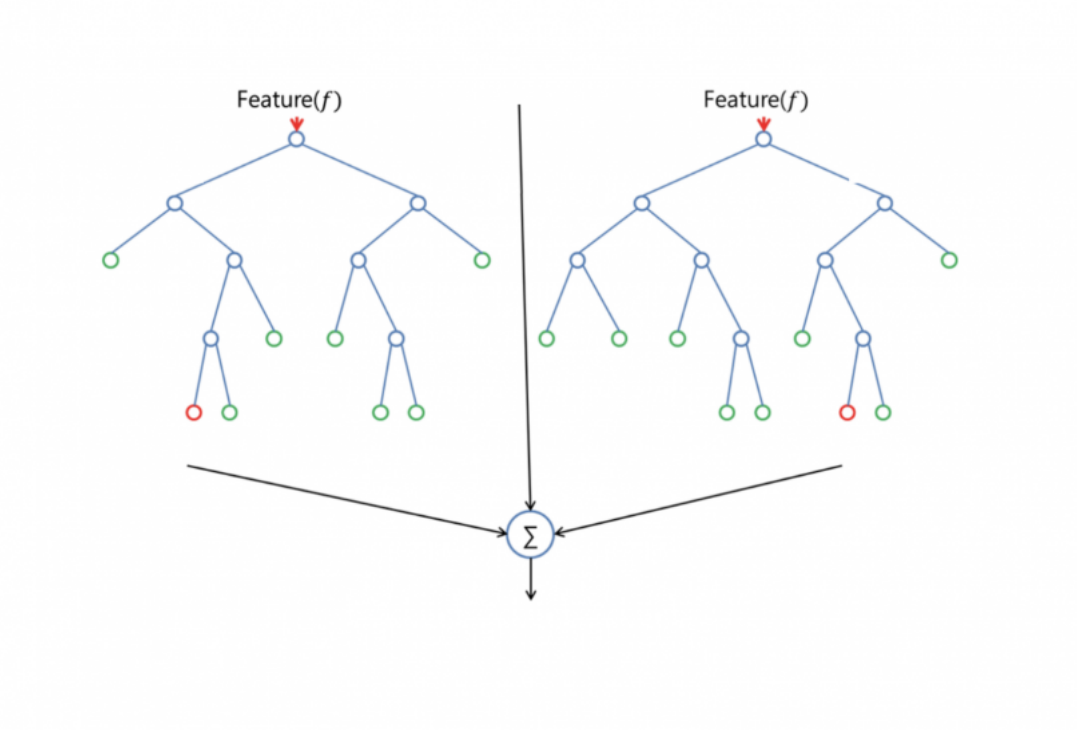

Figure 9.2: Random Forest Classifier

Therefore, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

Chapter 10

Evaluation Metrics Comparison

Table 10.1: Comparison of Evaluation Metrics

Comparison of Evaluation Metrics			
ALGORITHM	AUC ROC SCORE	RECALL SCORE	PRECISION SCORE
Linear Regression (Baseline Features)	0.7624	0.1122	0.5813
Linear Regression (All Features)	0.7588	0.1730	0.5987
Random Forest	0.8240	0.3498	0.7390

Random Forest model is giving the best result for each fold.

Chapter 11

Applications

- Financial Institutions: Banks/NBFC, Lending Companies can predict propensity to churn for each customer and take necessary actions to attract the customers.
- Similar models can also be developed to predict probability to churn for each customers in different domains.

Figure 11.1: Why Customers Leave and What Banks can do?

Figure 11.2: Bank Customer Churning

Chapter 12

Conclusion

- A Machine Learning Model to predict propensity to churn for each customer is developed.
- Missing value imputation, numerical value transformation are performed on dataset.
- Logistic Regression and Random Forest algorithm is used while building model.
- Random Forest model is giving the best result for each fold after cross validation.