



HIGH LEVEL DESIGN

Bank Marketing Analytics

Chetan Patil, Mr Pushpendra Dudi, Abhishek Gupta, Rupam Gupta

HIGH LEVEL DOCUMENTATION

Contents

Abstract.....	2
1. Introduction.....	2
.	
1.1 Why this High-Level Design Document?	2
1.2 Scope	3
2. General Descriptions	4
2.1. Product Perspective & Problem Statement	4
2.2. Tools used	5
3. Pre-processing	5
4. Methodology	5
5. Conclusion	13

Abstract

Bank Marketing :-

Bank marketing deals with providing services to satisfy customers' financial needs and wants.

To satisfy these financial needs, customers want specific services. All the techniques and strategies of marketing are used so that ultimately they induce the people to do business with a particular bank.

Marketing is important for growing market share as well as sales in banking and insurance. Marketing is essential for any business. Since the Banking sector is moving towards customer-centric, Marketing is very important for that.

Traditional banking method is changed to digital banking. Service marketing might include the process of selling telecommunications, health treatment, financial, hospitality, car rental, air travel, and professional services.

The marketing of bank services is the activity of presenting, advertising and selling of bank's products in the best possible way in order to satisfy consumers' requirement profitable. Marketing of banks services is one of the services rendered by financial industry (bank).

Term Deposit

With a term deposit, you lock away an amount of money for an agreed length of time (the 'term') – that means you can't access the money until the term is up. In return, you'll get a guaranteed rate of interest for the term you select, so you'll know exactly what the return on your money will be

1. Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
 - Security
 - Reliability
 - Maintainability
 - Portability
 - Reusability
 - Application compatibility
 - Resource utilization
 - Serviceability

1.2 Scope

HIGH LEVEL DOCUMENTATION

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

2 General Descriptions

2.1 Product Perspective

The marketing of bank services is the activity of presenting, advertising and selling of bank's products in the best possible way in order to satisfy consumers' requirement profitable. Marketing of banks services is one of the services rendered by financial industry (bank).

Problem Statement:

The data is related to direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe to a term deposit.

The data is related to direct marketing campaigns of a Portuguese banking institution.

The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be subscribed or not.

HIGH LEVEL DOCUMENTATION

2.2 Tools used

Business Intelligence tools and python libraries works such as numpy, Pandas, Excel and Power BI are used to build the whole framework.



PREPROCESSING

Dataset

For this research, we have used '**Bank - Full**' dataset which has been provided by '**Ineuron.ai**'. This dataset contains more than 45,211 data rows with detailed information of Campaign output from 2008 to 2010 with respect different ages, job groups , Educations and others

A lot of pre-processing was required to handle missing values, noise and outliers. We have considered 16 different attributes for this research: **Age, Job, Marital, Education, Default, Balance, Housing (Housing Loan), Loan(Personal Loan), Contact (Communication Type), Day, Month, Duration(Call Duration), Campaign (# Calls), Pdays, previous, poutcome.**

METHODOLOGY

First we imported necessary libraries of python which we needed to precede our work.

```
import pandas as pd,os
```

HIGH LEVEL DOCUMENTATION

After loading our data set we checked is there any null value present on any attributes and we discovered that for production attribute there were 0 null values. Since it is mentioned in Document that null values are presented as an unknown instead of nan/NaN/Missing

```
bank_df.isnull().sum()
```

```
age          0
job          0
marital      0
education    0
default      0
balance      0
housing      0
loan         0
day          0
month        0
duration     0
campaign     0
pdays       0
previous     0
poutcome     0
y            0
```

Changing column names for better understanding

```
bank_df.rename(columns
={ 'age': 'Age_Group', 'job': 'Job_type', 'housing': 'housing_loan', 'loan': 'P
ersonal_loan'}, inplace = True)
bank_df.rename(columns
={ 'duration': 'last_call_duration', 'campaign': 'Current_followUp', 'pdays'
: 'Contact_Day_diff', 'previous': 'previous_followUp', }, inplace = True)
bank_df.rename(columns
={ 'poutcome': 'Previous_camp_Status', 'y': 'Current_camp_status'}, inplace
= True)
```

creating function for age group to ease of better analysis

```
def age_grp(x):
    if x >=18 and x<=30:
        return 'Young Adults'
    elif x>30 and x<=45:
        return 'Mid Age Adults'
    elif x>45 and x<=60:
        return 'Veterans'
    else:
        return 'Senior Citizen'
```

Applying Age_Group Function to change data accordingly.

```
bank_df['Age_Group'] = bank_df['Age_Group'].apply(age_grp)
```

Checking value count after categorizing.

```
bank_df['Age_Group'].value_counts()
```

```
Mid Age Adults      23733
Veterans            13260
Young Adults        7030
Senior Citizen      1188
Name: Age_Group, dtype: int64
```

Checking Job Count for the dataset.

```
bank_df.Job_type.value_counts()
```

```
blue-collar      9732
management      9458
technician       7597
admin.           5171
services         4154
retired          2264
self-employed    1579
entrepreneur     1487
unemployed       1303
housemaid        1240
student          938
unknown          288
Name: Job_type, dtype: int64
```

Job has 288 unknown values so will replace it with Blue-Collar as it has maximum count in the dataset

```
def unknown_bluecollar(x):
    if x == 'unknown':
        return 'blue-collar'
    else:
        return x

bank_df['Job_type'] = bank_df['Job_type'].apply(unknown_bluecollar)
```


Creating and applying function for job group where we considered
Desk job as white collar job,
Field job as blue collar job,
Self-employed as Entrepreneur.

```
def job_group(x):  
    if x == 'admin.' or x == 'management' or x == 'services':  
        return 'White Collar'  
    elif x == 'blue-collar' or x == 'housemaid' or x == 'technician':  
        return 'Blue Collar'  
    elif x == 'entrepreneur' or x == 'self-employed':  
        return 'Entrepreneur'  
    else:  
        return x  
bank_df['Job_type'] = bank_df['Job_type'].apply(job_group)
```

Checking Count of Job_Type

```
bank_df['Job_type'].value_counts()  
Blue Collar      18857  
White Collar     18783  
Entrepreneur      3066  
retired          2264  
unemployed       1303  
student          938  
Name: Job_type, dtype: int64
```

We have 1857 unknown values in the Education Column

```
bank_df['education'].value_counts()  
secondary      23202  
tertiary       13301  
primary        6851  
unknown        1857  
Name: education, dtype: int64
```

Creating and Applying function in Education Column and replacing
unknown with secondary as highest count in the dataset.

```
def unknown_secondary(x):  
    if x == 'unknown':  
        return 'secondary'  
    else:  
        return x  
bank_df['education'] = bank_df['education'].apply(unknown_secondary)
```

Creating and applying function for balance group

considered value less than 0 as **negative balance**.

values greater than 0 and less than equal to 500 as **low balance**.

values greater than 500 and less than equal to 4000 as **average balance**.

values greater than 4000 as **high balance**.

```
def balance_grp(x):  
    if x <= 0 :  
        return 'Negative balance'  
    elif (x > 0 and x <=500):  
        return 'Low Balance'  
    elif (x > 500 and x <= 4000):  
        return 'Average Balance'  
    else:  
        return 'High Balance'  
  
bank_df['balance']=bank_df['balance'].apply(balance_grp)
```

Removing Contact Column as it has no use in analysis

```
bank_df.drop(['contact'],axis=1 ,inplace = True)
```

Changing last_call_duration in Minutes in ease of analysis

```
bank_df['last_call_duration'] = (bank_df['last_call_duration'] /  
60).round(0)
```

Creating and applying function for Duration

Short Call Time duration >= and <= 2.

Medium Call Time duration >= 2 and <= 5

High Call Time duration <=5.

```
def duration(x):  
    if (x >= 0 and x <= 2):  
        return 'Short Call Time'  
    elif (x>2 and x <=5):  
        return 'Medium Call Time'  
    else:  
        return 'High Call Time'
```

```
bank_df['last_call_duration'] =  
bank_df['last_call_duration'].apply(duration)
```

Creating and Applying function for Campaign

Upto 5 follow ups value ≥ 0 and ≤ 5

More than 5 follow ups value ≥ 5 as more than 5 follow ups

```
def campaign_grp(x):  
    if x > 0 and x <=5:  
        return 'Upto 5 Follow Ups'  
    else:  
        return 'More Than 5 Follow Ups'  
  
bank_df['Current_followUp'] =  
bank_df['Current_followUp'].apply(campaign_grp)
```

Creating and applying Function For Contact_day_diff

Not Contacted when pdays = -1.

0-3 Months when between 0 to 90 days.

3-6 Months when between 90 to 180 days.

More than 6 Months when > 180 days.

```
def pdays_grp(x):  
    if x == -1:  
        return 'Not Contacted'  
    elif x >= 0 and x <= 90:  
        return '0-3 Months Back'  
    elif x > 90 and x <= 180:  
        return '3-6 Months Back'  
    else:  
        return 'More Than 6 Months'  
  
bank_df['Contact_Day_diff'] =  
bank_df['Contact_Day_diff'].apply(pdays_grp)
```

Creating & applying function for pdays column group.
Upto 5 Follow Ups when calls done are between 0 to 5.
More than 5 Follow Ups when Calls are more than 5.

```
def previous_grp(x):  
    if x >= 0 and x <=5:  
        return 'Upto 5 Follow Ups'  
    else:  
        return 'More Than 5 Follow Ups'  
  
bank_df['previous_followUp'] =  
bank_df['previous_followUp'].apply(previous_grp)
```

Replacing Unknown values with not contacted and other with failure.

```
bank_df['Previous_camp_Status'] =  
bank_df['Previous_camp_Status'].replace('unknown', 'Not  
Contacted').replace('other', 'failure')
```

Replacing failure and success with some meaningful names

```
bank_df['Previous_camp_Status'] =  
bank_df['Previous_camp_Status'].replace('failure', 'P N  
Subscribed').replace('success', 'P Subscribed ')
```

Saving final Output as a csv file

```
bank_df.to_csv('final_file.csv')
```

Dax Queries Used.

- For Current Campaign
 - New Measures

```
CURRENT SUBSCRIBED = COUNTROWS(FILTER('Banking Market Analysis  
final_file', 'Banking Market Analysis  
final_file'[Current_camp_status]="Subscribed"))
```

```
CURRENT UNSUBSCRIBED = COUNTROWS(FILTER('Banking Market Analysis  
final_file','Banking Market Analysis  
final_file'[Current_camp_status]="Not Subscribed"))
```

- New Column

```
Loan Takers = IF('Banking Market Analysis  
final_file'[housing_loan]="no" && 'Banking Market Analysis  
final_file'[Personal_loan]="no", "NO", "YES")
```

- For Previous Campaign Status-
 - New Measures

```
PREVIOUS SUBSCRIBED = COUNTROWS(FILTER('Banking Market Analysis  
final_file','Banking Market Analysis  
final_file'[Previous_camp_Status]="P Subscribed"))
```

```
PREVIOUS UNSUBSCRIBED = COUNTROWS(FILTER('Banking Market Analysis  
final_file','Banking Market Analysis  
final_file'[Previous_camp_Status]="P N Subscribed"))
```

```
PREVIOUS NOT CONTACTED = COUNTROWS(FILTER('Banking Market Analysis  
final_file','Banking Market Analysis  
final_file'[Previous_camp_Status]="Not Contacted"))
```

```
TOTAL LEADS = COUNT('Banking Market Analysis  
final_file'[Current_camp_status])
```

Conclusion

- For Both Previous & Current Campaign Maximum Conversion in Age_Group is for Senior Citizen & Minimum Conversion is for Veteran.
- As per Education maximum conversion for current campaign & previous campaign is Tertiary & Minimum for both campaigns is Primary.
 - This Concludes more the education more is conversion.

- For Job Group Top 3 conversions for Both Campaigns are Retired, Student & Unemployed.
 - Here we can easily conclude that those who don't have any active source of income.
- As per Marital Status , maximum conversion for both previous and Current campaigns is Single and Minimum is Married.
 - Conclusion Single > Divorced > Married
- For both previous and current Campaign maximum conversion is in High Balance Group.
 - Higher the Balance group higher the chance of Conversion.
- Those who don't have loans have a higher chance of conversion for term deposit.
- There is more probability of conversion for those who are having personal loan than home loan.
- Maximum people subscribed to the scheme up to 5 follow up calls.
- Most people agreed when call duration is high.
 - Bank should focus on high call duration in 5 followup calls.
- Defaulter didn't subscribe for term deposit.
- Maximum conversion month is March & Min is May.
- Top 3 Months for Conversion is March, September & December.
- People who subscribed in the previous month have a higher chance of conversion in Current Campaign (64% Retention Rate).
- Chances of subscription is higher when follow up calls are done within 3 months.
- As Contact Day difference is increasing then Rate of subscription is decreasing.