Marketing Data Analysis

Jaimin Patel (C0798277)

Janki Gajjar (C0806344)

Sai Akshay Muralidhar (C0806686)

Mohammed Abdul Moid Arif (C0797125)

Lambton College

Acknowledgement

We would like to express our special thanks of gratitude to the author Kaushik Varma whose dataset of market analysis we are using for our final project in subject Data Science and Machine Learning.

Secondly, we would like to thank professor Vahid who gave us the opportunity to do this wonderful data analysis project, which also helped us in doing a lot of research and we came to know about so many new things.

Table of Contents

Summary

The report entitled “Marketing Data Analysis” provides insights from the customer data by performing Exploratory Data Analysis. This dataset is from the banking sector. The bank's representative contacted their customer and got some data of the customers. It contains customer data and the call outcome with call duration time. We will analyse the data from banking customers and record the responses of the customers and gain insights from the data. We will build a machine learning model which will help us to predict whether the customers have taken loan or not.

Data understanding:

Original Data:

]:

	banking marketing	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnam
0	customer id and age.	NaN	Customer salary and balance.	NaN	Customer marital status and job with education...	NaN	particular customer before targeted or not	
1	customerid	age	salary	balance	marital	jobedu	targeted	de
2	1	58	100000	2143	married	management,tertiary	yes	
3	2	44	60000	29	single	technician,secondary	yes	
4	3	33	120000	2	married	entrepreneur,secondary	yes	
...	...	...	...	...	...	...	...	
45208	45207	51	60000	825	married	technician,tertiary	yes	
45209	45208	71	55000	1729	divorced	retired,primary	yes	

Figure 1.1

- The first row contains unnecessary information in the original data, and the second row contains redundant dataset column information. So, we replaced the third row as the first row as it has column headings after removing the first and second rows.

Dataset after changing columns:

5]:

1	customerid	age	salary	balance	marital	jobedu	targeted	default	housing	loa
0	1	58	100000	2143	married	management,tertiary	yes	no	yes	n
1	2	44	60000	29	single	technician,secondary	yes	no	yes	n
2	3	33	120000	2	married	entrepreneur,secondary	yes	no	yes	ye
3	4	47	20000	1506	married	blue-collar,unknown	no	no	yes	n

The dataset contains the below columns.

- **customerid:** Id of the customer.
- **age**: age of the customer.
- **salary**: monthly salary of the customer.
- **balance**: account balance of the customer.
- **marital**: marital status of the customer.
- **jobedu**: job and education of the customer, separated by a comma.
- **targeted**: whether the customer is a targeted one or not.
- **default**: does the customer have a default account in the bank or not.
- **housing**: Whether the customer has taken a housing loan.
- **loan**: whether the customer has taken any other loan except housing loan
- **contact**: how a banking representative contacted the customer, by cellular phone or telephone.
- **day**: day of the contact to the customer by a banking representative.
- **month**: the month and the year of the contact to the customer by a banking representative. A comma separates month and year.
- **duration**: The time duration for which the representative has been on a call with the customer. Call duration is in seconds and minutes.
- **pdays**: how many days has it been since the last contact with the customer.
- **previous**: number of times customer has been contacted before.
- **outcome of previous contact**: it shows whether the last call to the customer was successful or not.
- **response**: does the customer respond after the call?

- The dataset has 45211 records and 19 columns.
- There are some missing values found in the columns age, month, and response. (Used df.describe() method)
- The customer id is unique for each record, which makes sense.
- 'poutcome' column has almost 37000 missing values as 'unknown' text, which is around 81% of the total dataset.
- The dataset contains a combination of categorical and continuous values in columns.

Data Cleaning:

The dataset contains columns whose names are not descriptive to tell about the column. So, we renamed the column headings. All the renamed column names are as below.

- "customerid": "customer_id",
- "age": "age",
- "salary": "monthly_salary",
- "balance": "account_balance",
- "marital": "marital_status",
- "jobedu": "job_education",
- "targeted": "targeted_customer_yn",
- "default": "default_yn",
- "housing": "house_loan_yn",
- "loan": "other_loan_yn",
- "contact": "contact_carrier",
- "day": "day_of_contact",
- “month": "month_year_of_contact",
- "duration": "call_duration",
- "campaign": "campaign",
- "pdays": "last_contacted_days",
- "previous": "total_previous_contact",
- "poutcome": "last_contact_outcome",
- "response": "customer_responded_yn"

All the columns are of string object. So, the column value is converted to the appropriate format.

Our task for building a machine learning column is to predict whether the customer has taken a loan or not. We have two columns, 'housing', and 'loan', but we do not have a column for all the loans. So, we made a new column named 'loan_yn', which tells whether the customer has taken any loan or not. We made 'loan_yn' column from 'house_loan_yn' and 'other_loan_yn' columns. If either of the columns contains yes, means the customer has either taken a house loan or other type of loan, then 'loan_yn' column value will be yes, otherwise no.

t_contacted_days	total_previous_contact	last_contact_outcome	customer_responded_yn	loan
-1	0	unknown	no	yes
-1	0	unknown	no	yes
-1	0	unknown	no	yes
-1	0	unknown	no	yes
-1	0	unknown	no	no

In our dataset, if column name has a suffix as ‘yn’, this means that the column contains only yes and no data.

Column name ‘jobedu’ indicates two values, job and education of the customer (separated by comma). We removed that column and made 2 new column jobs and education (taking data from ‘jobedu’ column). Also, in the job column wherever value ‘admin.’ appeared, we changed it to ‘admin’. Also, deleted column **job_education.6**

ous_contact	last_contact_outcome	customer_responded_yn	loan	job	education
0	unknown	no	yes	management	tertiary
0	unknown	no	yes	technician	secondary
0	unknown	no	yes	entrepreneur	secondary
0	unknown	no	yes	blue-collar	unknown
0	unknown	no	no	unknown	unknown

Another column ‘month’ indicating, which month and the year the customer was contacted. We removed this column and separated data into 2 new columns **month_of_contact** and **year_of_contact**. Also, deleted column **month_year_of_contact.**

l_yn	loan	job	education	month_of_contact	year_of_contact
no	yes	management	tertiary	may	2017
no	yes	technician	secondary	may	2017
no	yes	entrepreneur	secondary	may	2017
no	yes	blue-collar	unknown	may	2017
no	no	unknown	unknown	may	2017

Duration column has the call duration time in the form of seconds and minutes. We have converted call duration time in seconds only to maintain same time format throughout the dataset.

Converted age and year columns data type from float to integer, which is a perfect representation of these data. Replaced all unknown values in dataframe to **NaN** value. So, in future, we can count missing values and handle it properly.

Change the categorical column to integer

This step is performed to analyse categorical column data using plots. Later while building the machine learning model, we will revert this conversion to use one-hot encoding for categorical values. We have converted below columns values:

1. marital_status
    - Single: 1
    - Married: 2
    - Divorced: 3

2. contact_carrier
    - Cellular: 1
    - Telephone: 2
3. Last_contact_outcome
    - Failure: 1
    - Success: 2
    - Other: 3
4. job
    - Unemployed: 1
    - Self-employed: 2
    - Student: 3
    - Entrepreneur: 4
    - Technician: 5
    - Management: 6
    - Blue-collar: 7
    - Admin: 8
    - Services: 9
    - Housemaid: 10
    - Retired: 11
5. education
    - Primary: 1
    - Secondary: 2
    - Tertiary: 3
6. month_of_contact
    - Jan: 1

- Feb: 2
- Mar: 3
- Apr: 4
- May: 5
- Jun: 6
- Jul: 7
- Aug: 8
- Sep: 9
- Oct: 10
- Nov: 11
- Dec: 12

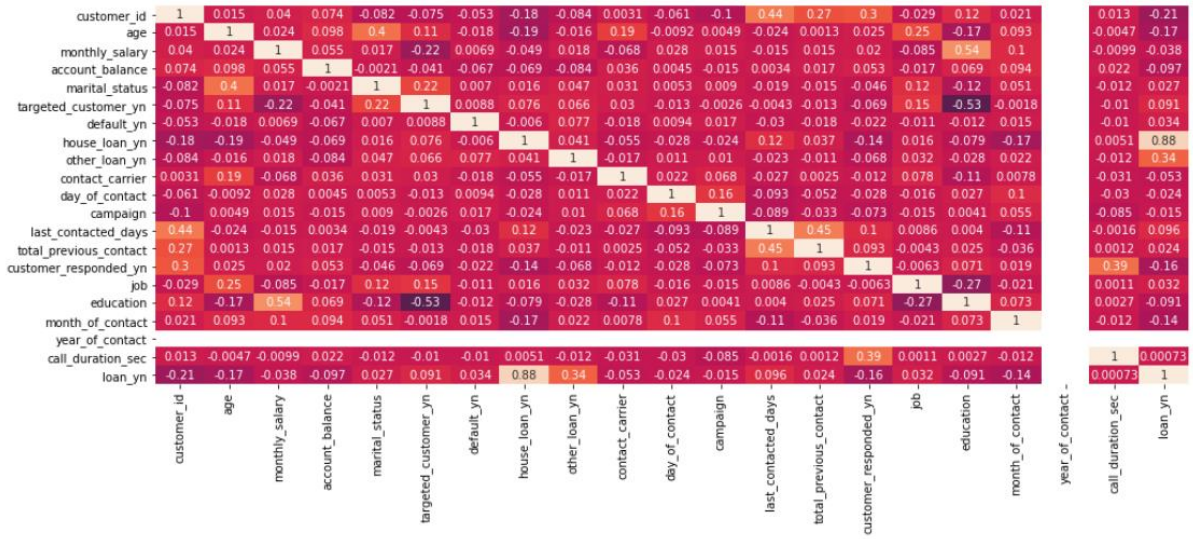
Columns like **loan_yn**, **customer_responded_yn, other_loan_yn, house_loan_yn, default_yn,** and **targeted_customer_yn** converted to 1 and 0. 1 for yes and 0 for no.

Handle Missing Values

Columns containing null values are **age, contact_carrier, customer_responded_yn, job, education, month_of_contact, year_of_contact,** and **last_contact_outcome.**

Handling missing values for above mentioned columns

1. last_contact_outcome
    - This column contains almost 81% of missing data. So, by going rule of thumb that if the column has 60-70% of data can be dropped. So, we dropped this column.
    - Other columns are having less missing data. So, before handling that values, we checked correlation of each column in a heatmap.



2. year_of_contact

    - This column has only one value that is **2017.** So, it is better to drop this column, because the model will not learn anything from this column even if we feed it as input. So, we dropped this column.

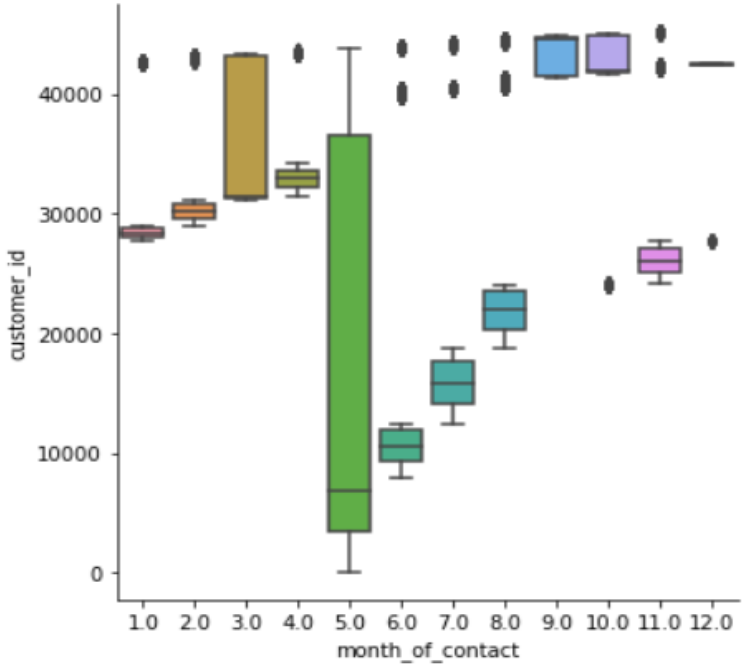
3. customer_responded_yn

    - This column has only 30 missing values.
    - This column is 14% negatively related to house_loan_yn and 39% positively related to call_duration_sec column
    - We have seen some statistical relationship between customer_responded_yn and call_duration_sec column. So, we derived that if the call_duration is less that 221 seconds, then impute the value 0, for call duration greater than 537 seconds, impute the value 1, and for call duration between 221 and 537, if the call duration seconds are near 221 then impute 0 and if it is near 537 impute 1 for null values.

4. month_of_contact

    - This column had only 50 missing values.

- From the box plot of this column we have seen that most of values are together. So, we can impute null values by nearby values.



- For this missing value handling, we used fill method of fillna function in pandas.

5. education

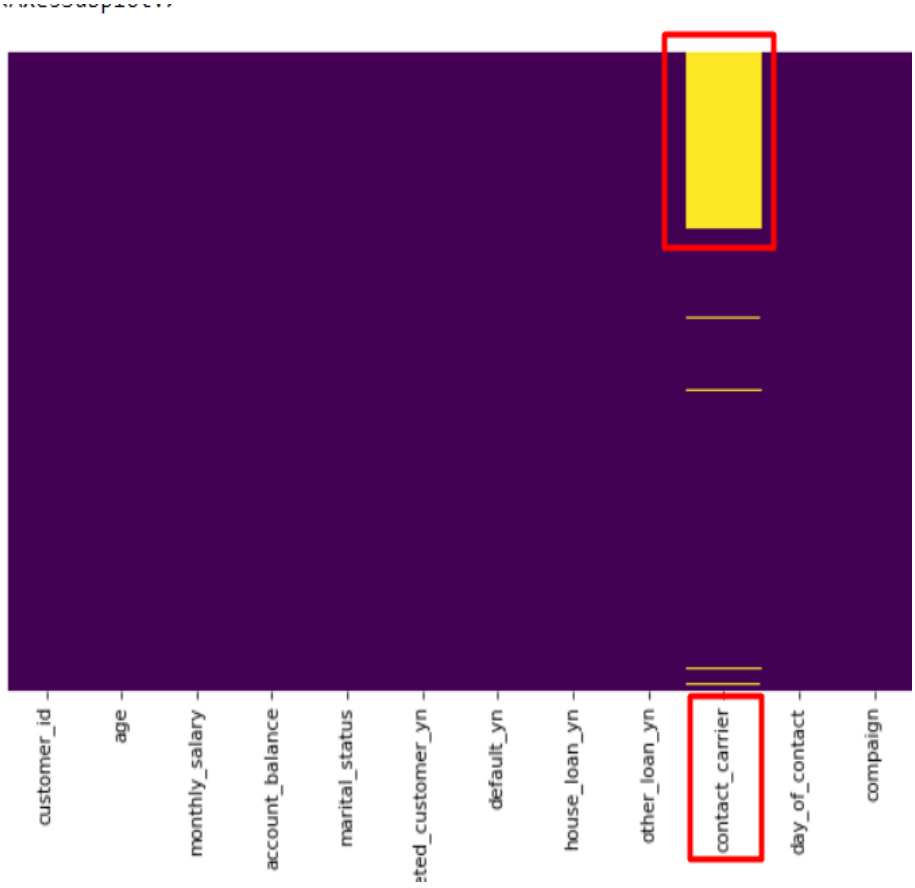
- This column has more missing values around 1857.
- Using heatmap, we have seen that it has 54% positive correlation with **monthly_salary**, 53% negative correlation with **targeted_customer_yn**, and 27% negative correlation with **job**.
- We used mean of monthly_salary for imputing null values for each type of education category.
- If the monthly_salary is near 34212, value **1 (primary)** is imputed.
- If the monthly_salary is near 49743, value **2 (secondary)** is imputed.
- If the monthly_salary is near 82873, value **3 (tertiary)** is imputed.

6. job

- This column has 288 missing values.

- Correlation shows, 25% positive correlated with age and 27% negative correlated with education.
- By looking at the data points in scatter plots, no observation has derived. So, we decided to drop 288 rows from the 45211 rows, which does not impact the machine learning model, because dropping records is relatively very less. And after dropping 288 rows, we have enough data to feed the model as an input.

7. age
    - This column has only 20 missing values.
    - It is 40% positively correlated with marital_status and 25% positively correlated with job column.
    - In this column also, no observation has been derived while looking tha data points using graphs. So, deciding to drop the record as 20 records will not affect this dataset.
8. contact_carrier
    - This column has so many missing values that is 12901.
    - By, looking at the heatmap, there is a lot of missing values at start, but after 30% of data, only small amount of missing values are there.



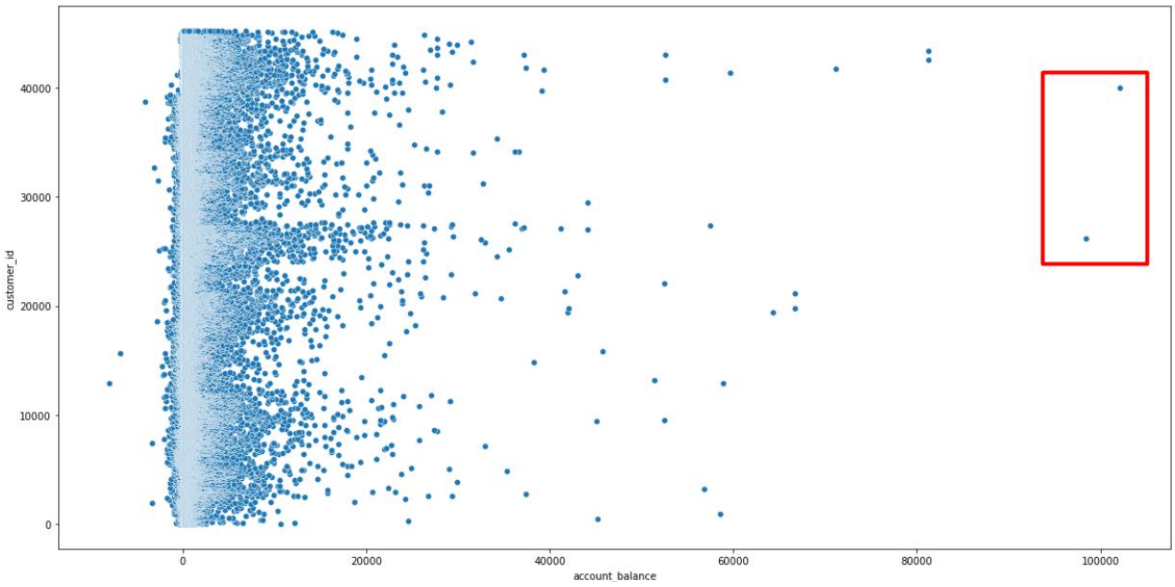
- From this column data, we have seen that 29142 customers contacted through cellular phone and only 2860 customers are contacted through telephone. So, every 10 cellular call there is 1 telephone call.
- So, we imputed 10 cellular data and 1 telephone data in the place of 11 missing value.

After, handling all the missing value, we checked correlation heatmap again and compared with old one. We found out that there is not much difference in correlation. So, we concluded that our methods of handling missing data are correct. After handling missing data our dataset is having 44903 records and 20 columns. That means we only lost a slight amount of data, which is good.

Outlier handling

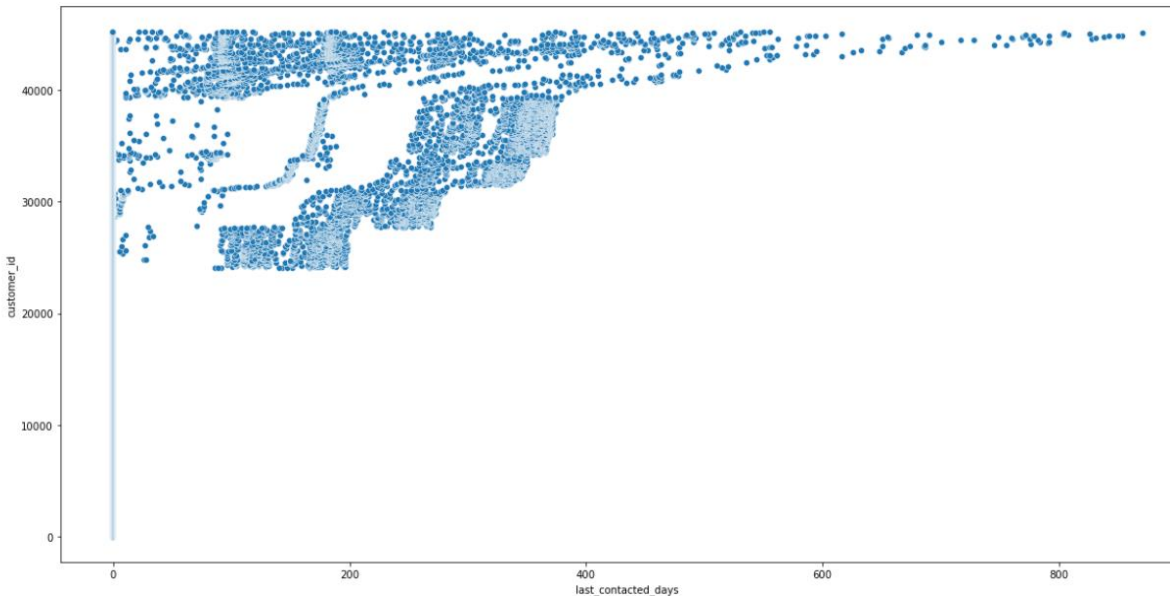
account_balance

- This column had big negative integers, which looks like outlier at first eye. But, after a little bit of googling, we have seen that account balance can be negative.



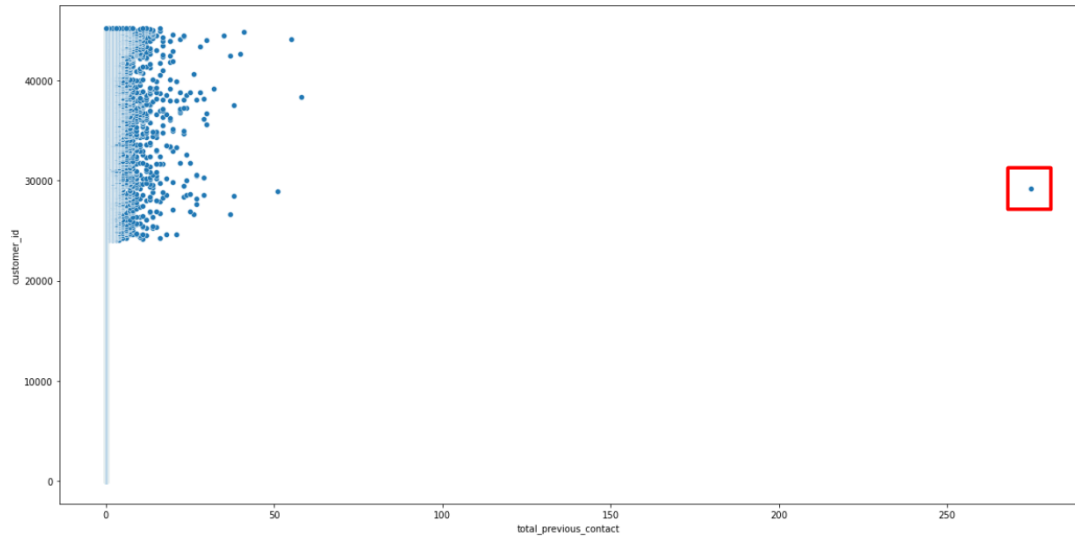
last_contacted_days

- The max value is 871 days, which can happen for a small amount of customer due to wrong contact information. The min value is -1, which indicates that the customer does not contacted before. So these are not outlier as per our understanding.



total_previous_contact

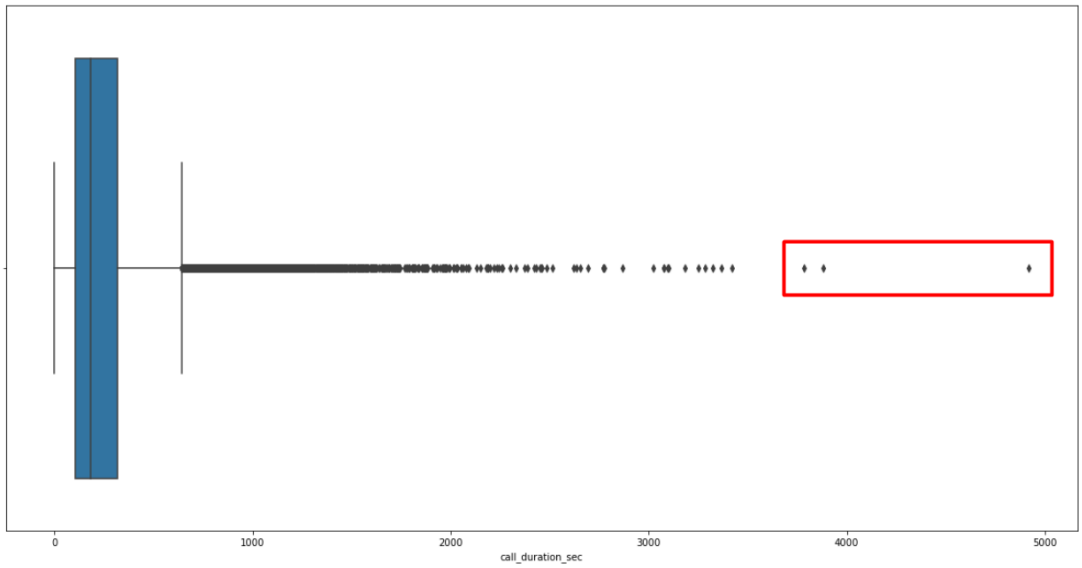
- All the values in this column lies between 0 to 58, except one value which is 275. So, 275 is an outlier.



- We replaced this outlier with the mean value of the column that is 1.

call_duration_Sec

- Most of the values are between 0 to 3500 seconds, except 3 values which are 3785, 3881 and 4918.
- 4918 seconds is around 82 minutes, which is clearly an outlier.



- We replaced these 3 outliers with the mean value of this column.
- After outlier handling, we converted all the float columns to integer columns.
- After cleaning the data, dataset has left with 44903 rows and 20 columns.

Exploratory Data Analysis

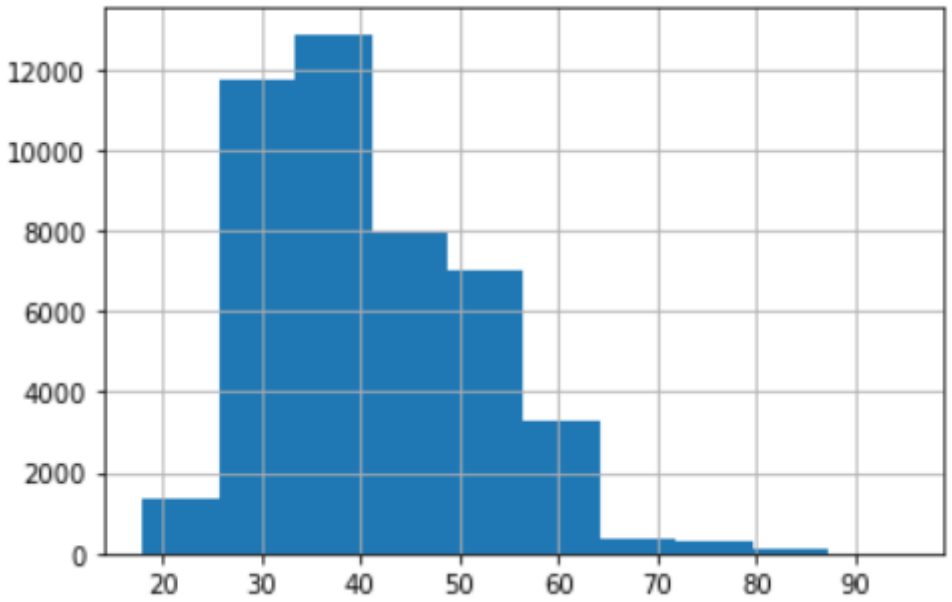
Measure Central Tendency of each feature

- Total 44903 records are in the dataset
- Standard deviation varies much between different continuous value columns. Monthly_salary has the biggest standard deviation followed by account_balance.
- minimum account balance is -8019 and maximum call duration is 3422 seconds.
- Total 63 campaigns conducted by banks.
- Most of the call conducted in the month of May, 2017.
- Around 66% of customers have taken a loan.

Distribution Plots

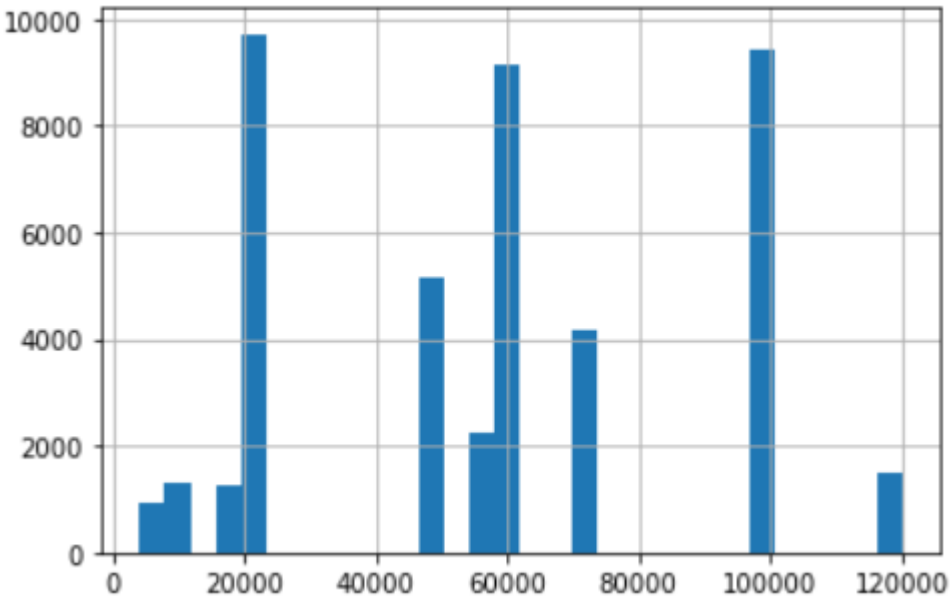
1. Age

- Normal distribution



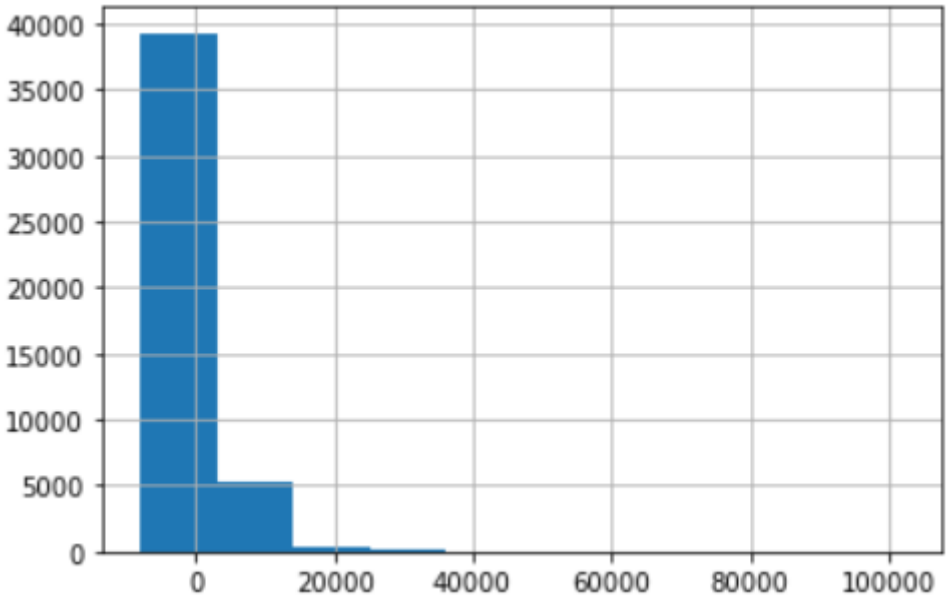
2. **Monthly_salary**

- Normal distribution with some gaps



3. **account_balance**

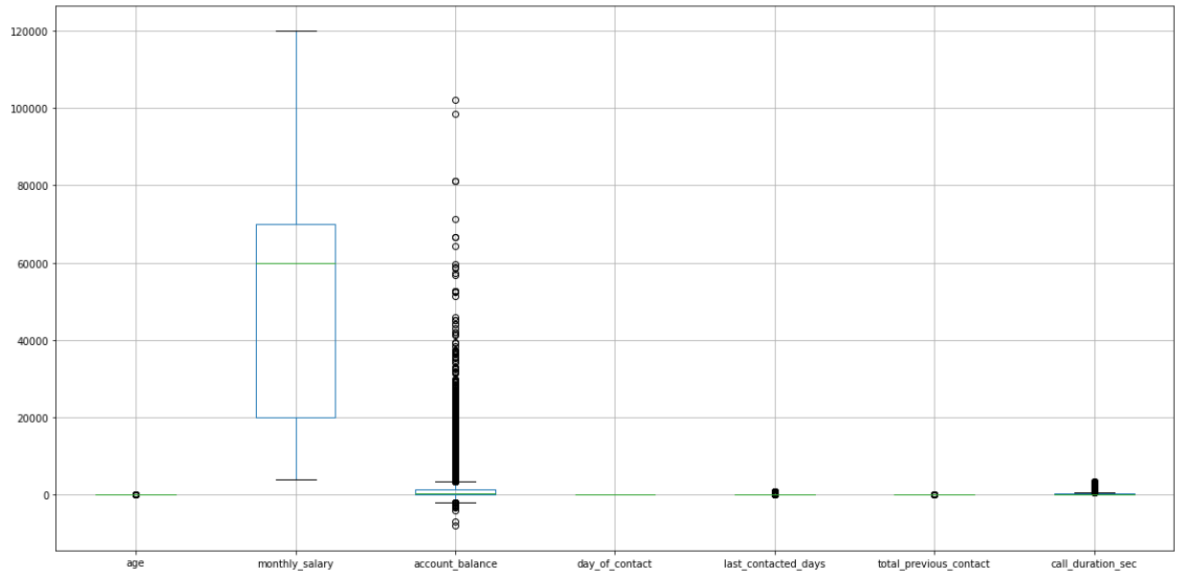
- Exponential distribution. This type of distribution is hard to understand for logistic and linear regression model.



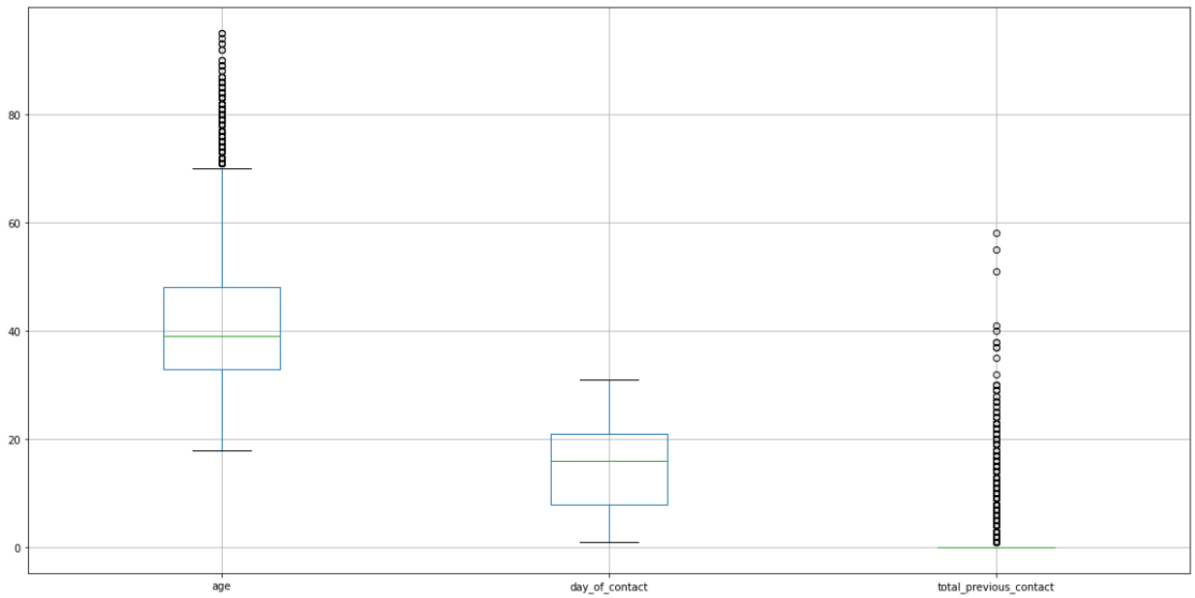
- Marital_status: Binomial distribution
- Targeted_customer_yn: Bernoulli distribution

- Last_contacted_days: exponential distribution
- Total_previous_contact: exponential distribution
- Call_duration_sec: exponential distribution
- Loan_yn: binomial distribution. This column has imbalance for each category data. So, it will affect model training.

From the boxplot of all the columns, we can say that, monthly_salary and account_balance columns has very high values compare to other continuous columns.

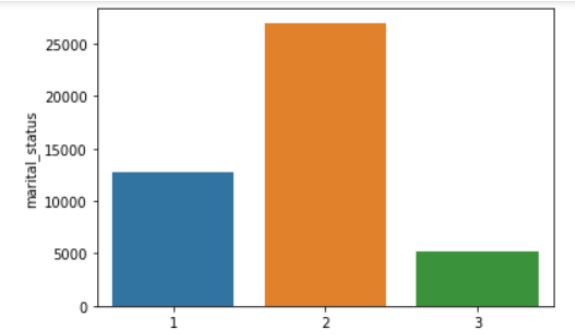


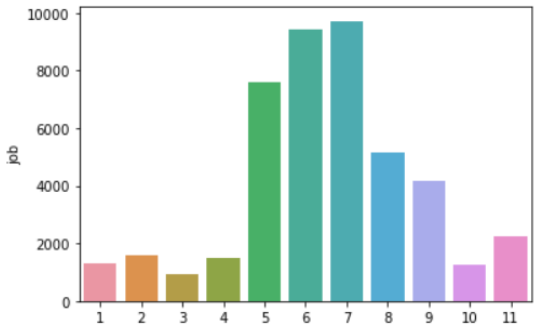
- There is also value difference between age, day_of_contact and total_previous_contact column. So, these data should be normalized to have a good performance of the model.



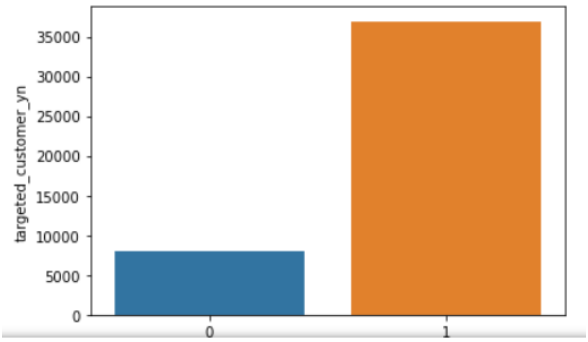
Visualizing categorical columns

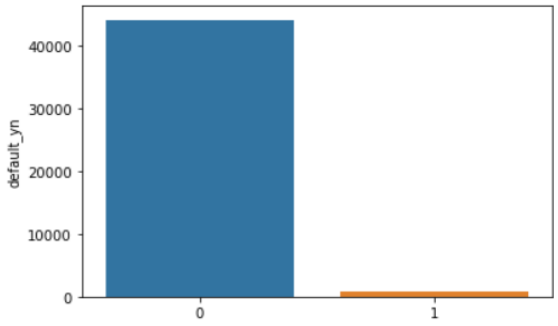
- Marital_status, job, education and month_of_contact has a normal distribution, which is good for model training.

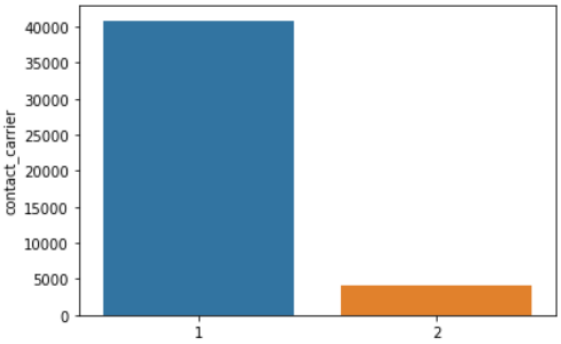




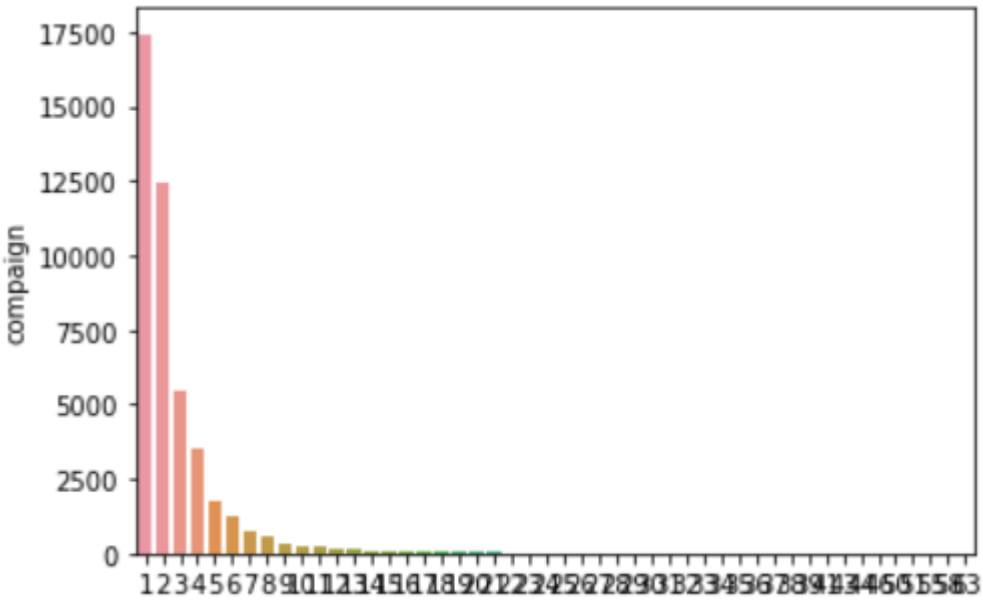
- targeted_customer_yn, default_yn, other_loan_yn, contact_carrier, customer_responded_yn, and loan_yn has an imbalance in the data, which will affect model training if given as an input.







- Only campaign column has an exponential distribution of values, which will be hard to understood by linear and logistic regression.



Statistical Analysis

- married customer tend to have more account balance, However monthly salary is relatively higher for divorced customer.
- single customer tend to have less age, while married and divorced customer has higher age.

marital_status	account_balance	age	monthly_salary
1	1299.095377	33.675499	57596.005661
2	1423.704344	43.356802	56622.162142
3	1175.909144	45.766590	60724.537037

- married and divorced customer tend to have taken house loan and other loan more than single person.

marital_status	house_loan_yn	other_loan_yn
1	0.541909	0.130445
2	0.566720	0.172290
3	0.558835	0.177469

- If married customer have taken a loan, then call duration will be less than if loan has not taken. But, for divorced and single customer this trend is opposite.

		call_duration_sec
marital_status	loan_yn	
1	0	262.285122
	1	268.937312
2	0	255.434935
	1	252.023193
3	0	258.193462
	1	265.142857

- If education of the customer is higher, the monthly salary goes higher too.

	monthly_salary
education	
1	32725.754742
2	50035.242477
3	83480.545031

- the targeted customers are who has less educated (primary and secondary).

	targeted_customer_yn
education	
1	0.907694
2	0.975900
3	0.510857

- most of the customer responded if the call duration time is high.

	call_duration_sec
customer_responded_yn	
0	221.019398
1	537.144297

- Less aged customer tend to take loan more.
- monthly salary is lower for customer who has taken a loan.
- account balance is lower for customer who has taken a loan.
- married customer tend to take a loan more.
- customers who have taken a loan are the one who are targeted.
- customers who have not taken a loan are contacted frequently.
- most contact has been done for customers who took the loan.
- customers don't responde much who took a loan.
- more educated customer take less loans.
- call duration time is almost similar for non loan taking and loan taking customers.

Build Machine Learning Model

- The task is to predict whether the customer has taken a loan or not.

Feature selection

- From the dataframe, house_loan and other_loan column is dropped because loan_yn column has been made from these 2 columns. So, if we add above 2 columns, then there will be high bias and model will learn only on those 2 columns.
- default_yn, contact_carrier, day_of_contact, campaign: these columns are not taken for model building, because it does not look relevent to our task.
- Changed all the integer categorical column to its original form. Because for categorical column, we need to do one-hot encoding, which can be done only the column is categorical and non-numeric.
- Model dataframe:

	loan_yn	age	monthly_salary	account_balance	marital_status	targeted_custon
0	1	58	100000	2143	married	
1	1	44	60000	29	single	
2	1	33	120000	2	married	
3	1	47	20000	1506	married	
4	1	35	100000	231	married	
...	...	...	...	...	...	
44898	0	51	60000	825	married	
44899	0	71	55000	1729	divorced	

- Categorical columns like marital_status needs to be encoded to feed in model.
- We used one-hot encoding from pandas to do encoding task.

```
# get dummy values for df_model
df_dum = pd.get_dummies(df_model)
df_dum
```

- After one-hot encoding dataframe dimensions are 44903 rows and 38 columns.

Split training and testing data

- The dataset is splitted 80% for training data and 20% for testing data using sklearn's train_test_split function.

Model Training and Testing

- We have trained total 4 machine learning classification model. Logistic regression, Support vector Machine, Random forest classifier and decision tree.
- The accuracy of all the models on test set are as below:
- Logistic regression: 62%
- Support vector machine: 53%
- Random forest classifier: 76.22%
- Decision tree classifier: 68.5%

Conclusion

The dataset is containing too many missing values and an imbalance in data. Some columns are not relevant to our task. Some columns containing substantial values compare to other columns, which affect the model training.

Classifiers like logistic regression achieved only 62% accuracy because we have seen that some column values are not normally distributed. However, the Random forest classifier performs the best with 76% accuracy, which can be increased if data like gender and job_sector will be added. Normalizing, transforming, and imbalance handling can increase the model performance.

References

- https://www.youtube.com/watch?v=KQ80oD_boBM&list=PL2zq7klxX5AQXzNSLtc_LEKFPh2mAvHIO&index=5
- https://www.youtube.com/watch?v=fhi4dOhmW-g&list=PL2zq7klxX5ASFejJj80ob9ZAnBHdz5O1t&index=3
- https://www.youtube.com/watch?v=QWgg4w1SpJ8&list=PL2zq7klxX5ASFejJj80ob9ZAnBHdz5O1t&index=4
- https://www.youtube.com/watch?v=lTc7NU9XpWE
- https://www.analyticsvidhya.com/blog/2020/03/pivot-table-pandas-python/
- https://www.youtube.com/watch?v=U_wKdCBC-w0
- https://www.analyticsvidhya.com/blog/2017/09/6-probability-distributions-data-science/
- https://medium.com/@szabo.bibor/how-to-create-a-seaborn-correlation-heatmap-in-python-834c0686b88e
- https://medium.com/@danberdov/dealing-with-missing-data-8b71cd819501#:~:text=As%20a%20rule%20of%20thumb,the%20variable%20should%20be%20considered
- https://github.com/Kaushik-Varma/Marketing_Data_Analysis