Data Glacier

Data Science Intern at Data Glacier

Week 10: Report on Group Project

Topic: Bank Marketing (Campaign)

Group Name: Campaign Catalysts

Specialization: Data Science

Batch Code: LISUM19

Date: 9th May 2023

Submitted to: Data Glacier

Group Member Details

S.No.	Name	Email	Country	College/ Company
1.	Yash Jayesh Doshi	yashjdoshi99@gmail.com	UAE	Orpheuss LLC
2.	Anuj Singh	dsanuj21@gmail.com	India	Mumbai University
3.	Yash Jadwani	yash.jadwani1998@gmail.com	United Kingdom	Kingston University
4.	Harold Wilson	haroldwilson537@gmail.com	United Kingdom	University of Buckingham

1. Problem Description

ABC Bank wants to sell its term deposit product to customers and before launching the product they want to develop a model which help them in understanding whether a particular customer will buy their product or not (based on customer's past interaction with bank or other Financial Institution).

Bank wants to use ML model to shortlist customer whose chances of buying the product is more so that their marketing channel (tele marketing, SMS/email marketing etc.) can focus only to those customers whose chances of buying the product is more.

2. EDA of Categorical Features (Harold Wilson and Yash Jadwani)

Data cleaning of categorical features:

Introduction:

Machine learning (ML) projects typically start with a comprehensive exploration of the provided datasets. It is critical that ML practitioners gain a deep understanding of:

- The properties of the data: schema, statistical properties
- The quality of the data: missing values, inconsistent data types
- The predictive power of the data: for example, the correlation of features with the target

2.1 Handling missing data for Bank marketing dataset

Data cleaning is an important step in data pre-processing due to its ability to help improve the quality of the dataset for a more reliable output. The presence of impurities in real-world data application has brought about the development of several methods to eradicate this problem to help improve the accuracy and usability of existing data (Müller and Freytag, 2005). [3] The data cleaning process involves the detection or removal of outliers, smoothing noisy data, filling in missing values and resolving inconsistency within a dataset (Han, Pei and Kamber, 2011). [4]

There is exactly no one way of dealing with missing data. There are different solutions for data imputation depending on the kind of problem and it always difficult to provide a general solution, and care should be taken when it comes to removing missing values in any given data set since doing so will introduce biasness in the model.

Imputing or Deleting missing values of the Data:

Before we decide to remove, replace or impute the data, we have to understand and establish the reason why data is missing.

- Missing at Random: This means that the tendency for a data point to be missing is not related to the missing data, but it is related to some of the observed data.

- Missing Completely at Random: The fact that a certain value is missing has nothing to do with its hypothetical value and with the values of other variables.

- Missing not at Random: Possible reasons are that the missing value depends on the hypothetical value or missing value is dependent on some other variable's value.

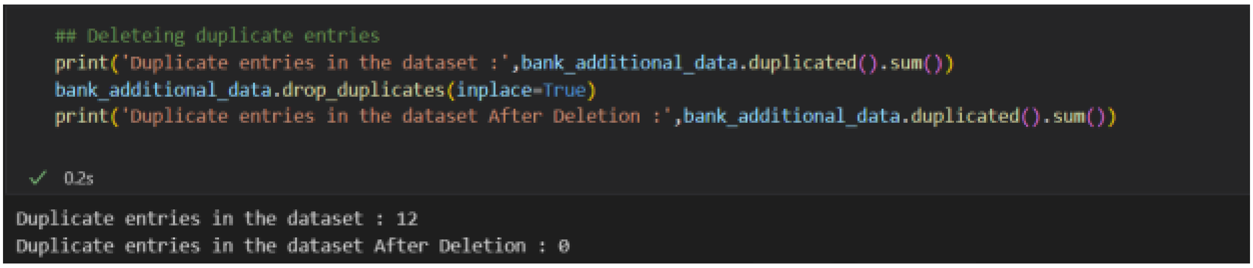
2.2 Data cleaning of categorical features in the data set:

We have the following unknown values for some of the features in the data set:

Features	Unknown Vaules	Minimum value	Maximum value
Job	990	unknown	admin
Marital	80	unknown	married
Education	1731	illiterate	university degree
Default	8597	yes	no
Housing	990	unknown	yes
Loan	990	unknown	no
contact	0	Nil	Nil
month	0	Nil	Nil
day_of_week	0	Nil	Nil
poutcome	35563	success	nonexistence
y	0	yes	no

Mode Imputation for Unknown/Missing values

a) Deleting Duplicate values

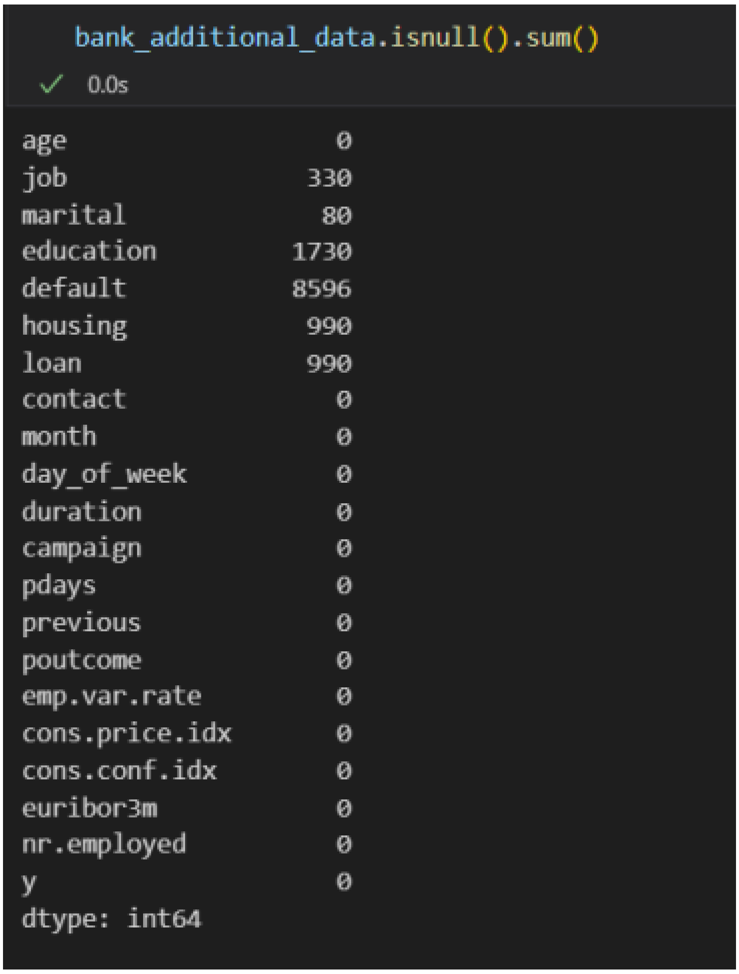


```
## Deleteing duplicate entries
print('Duplicate entries in the dataset :',bank_additional_data.duplicated().sum())
bank_additional_data.drop_duplicates(inplace=True)
print('Duplicate entries in the dataset After Deletion :',bank_additional_data.duplicated().sum())
```

✓ 0.2s

```
Duplicate entries in the dataset : 12
Duplicate entries in the dataset After Deletion : 0
```

b) Replacing Unknown/missing values with N/A and Checking for null values



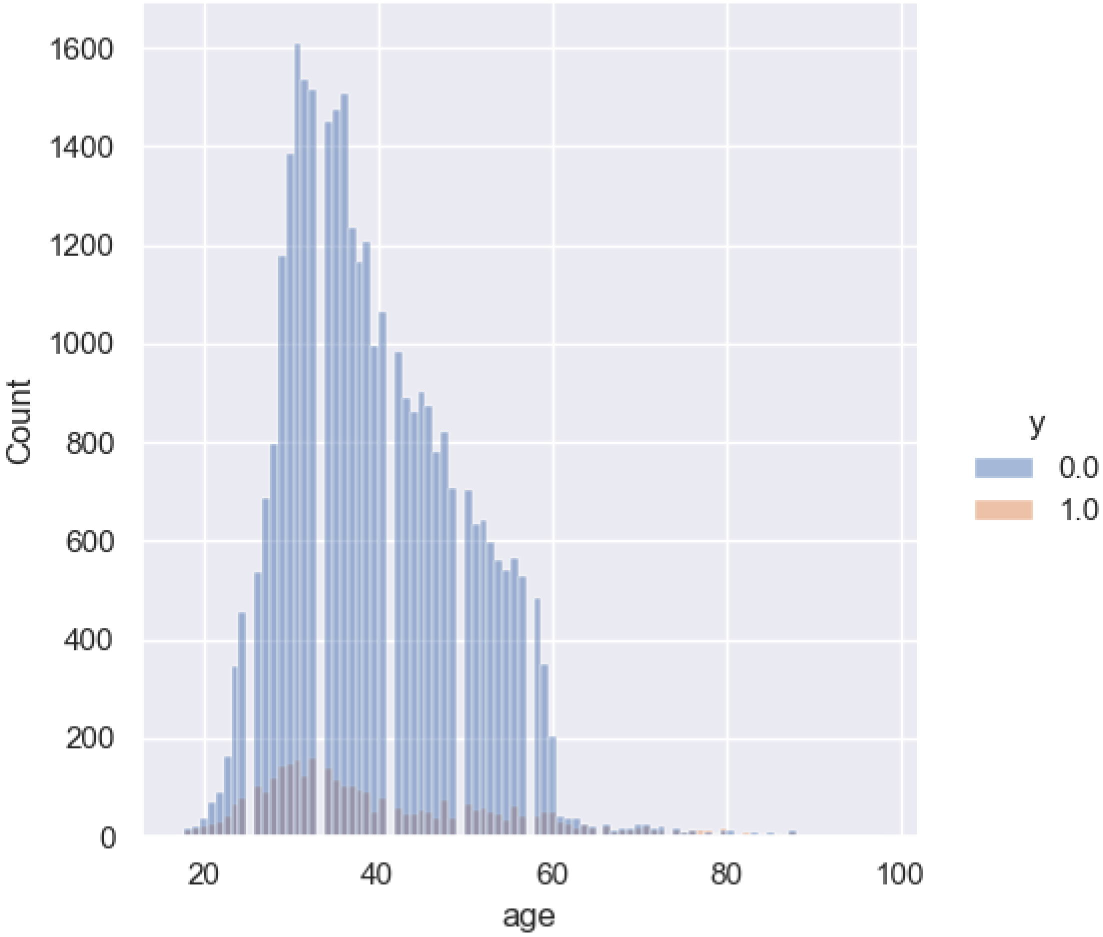
```
bank_additional_data.isnull().sum()
```

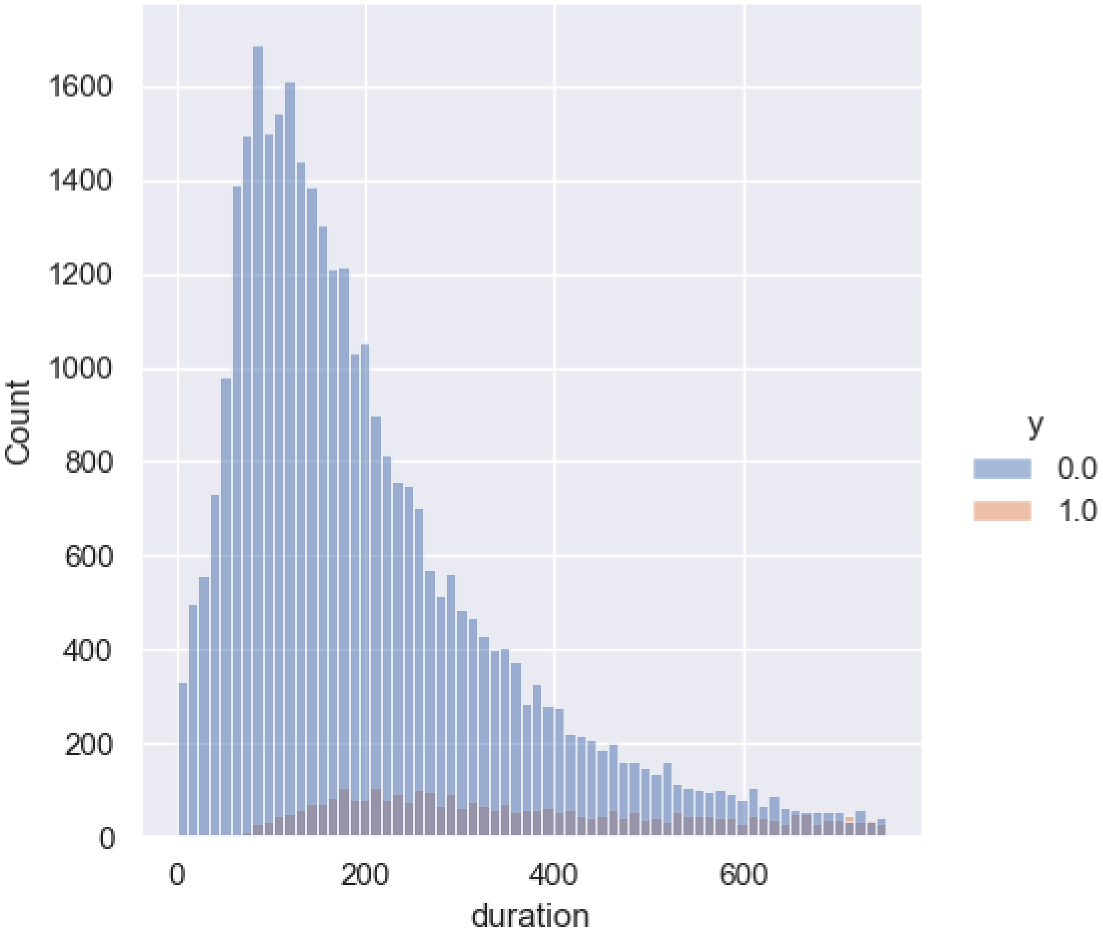
```
age                0
job              330
marital           80
education       1730
default         8596
housing          990
loan             990
contact            0
month              0
day_of_week        0
duration           0
campaign           0
pdays              0
previous           0
poutcome           0
emp.var.rate       0
cons.price.idx     0
cons.conf.idx      0
euribor3m          0
nr.employed        0
y                  0
dtype: int64
```

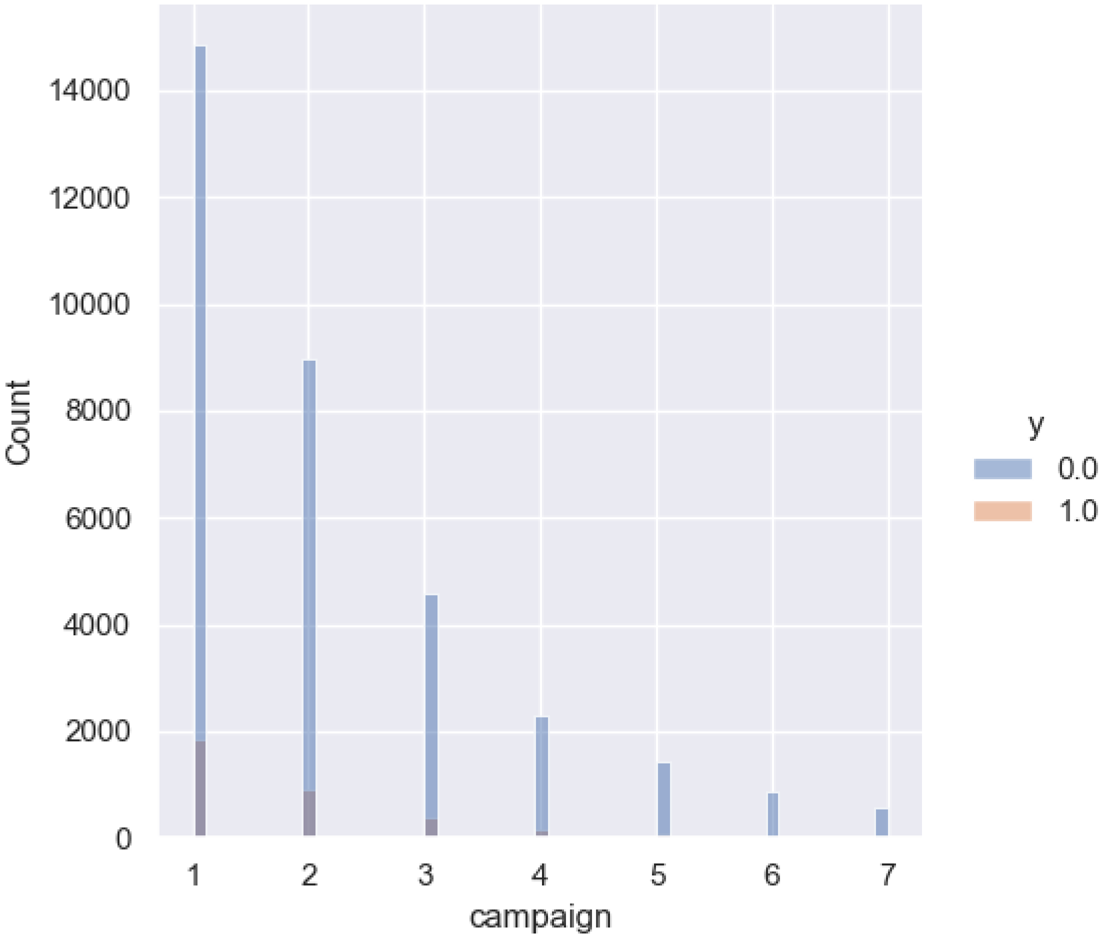
c) Mode Imputation Steps

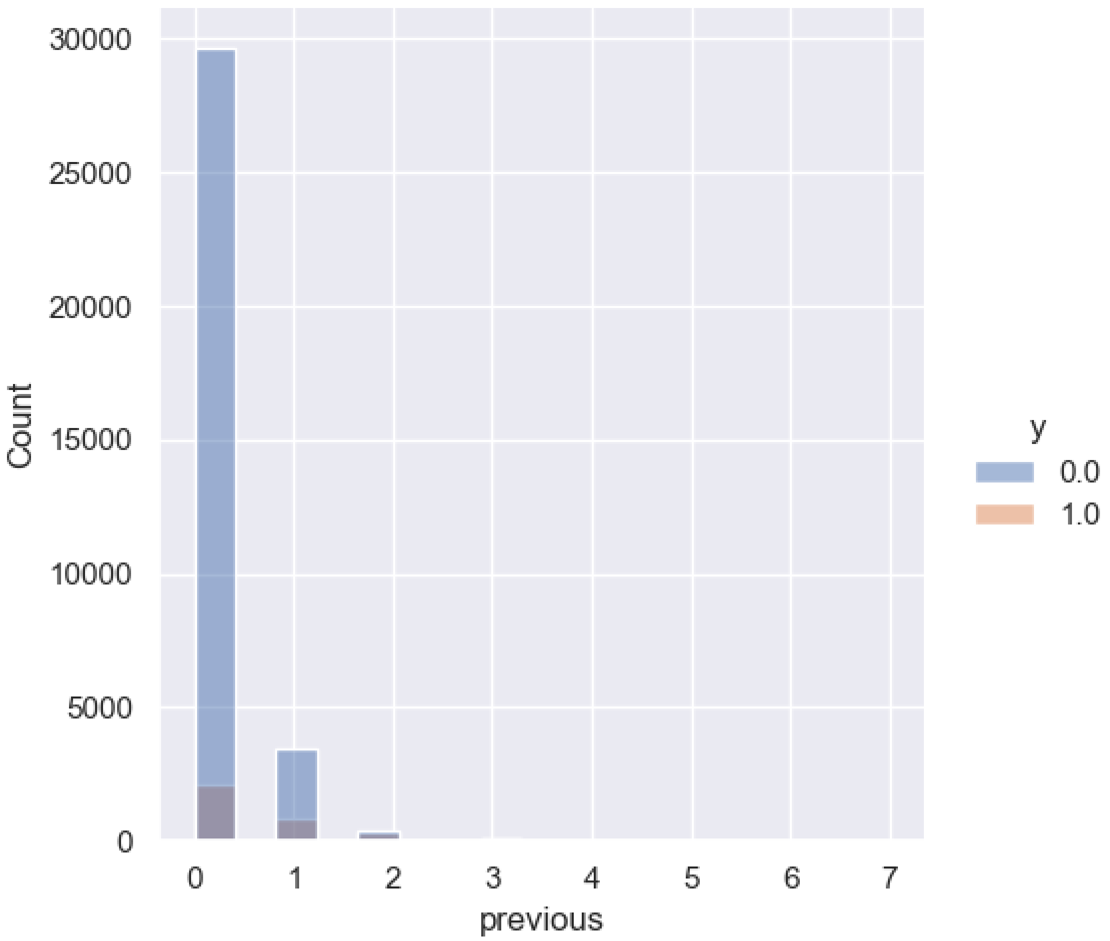
- We removed all the rows where job, default and housing are null.
- We removed all the rows where job and education are null.
- We encountered 75 missing values in the marital status variable and addressed this issue by creating an age marital map to impute these values.
- In the job variable, we found 197 instances of missing data, which we resolved by generating an age education job map and using it to impute the missing values.
- We found approximately 1600 instances of missing education records, which we addressed by generating a job education map and using it to impute the missing values.
- We removed all the rows where loan, default and housing are null.
- There was highly imbalance in default feature, only 3 individuals had defaulted. This suggests that the vast majority of individuals in the dataset have not defaulted on their payments. Therefore, we removed the default feature from the analysis as it may not be useful in making any meaningful conclusions
- We identified 763 missing values in both the loan and housing variables, which we imputed using information from the marital status, job, and education variables

2.3 Visualizing distributions for each category of target variable:

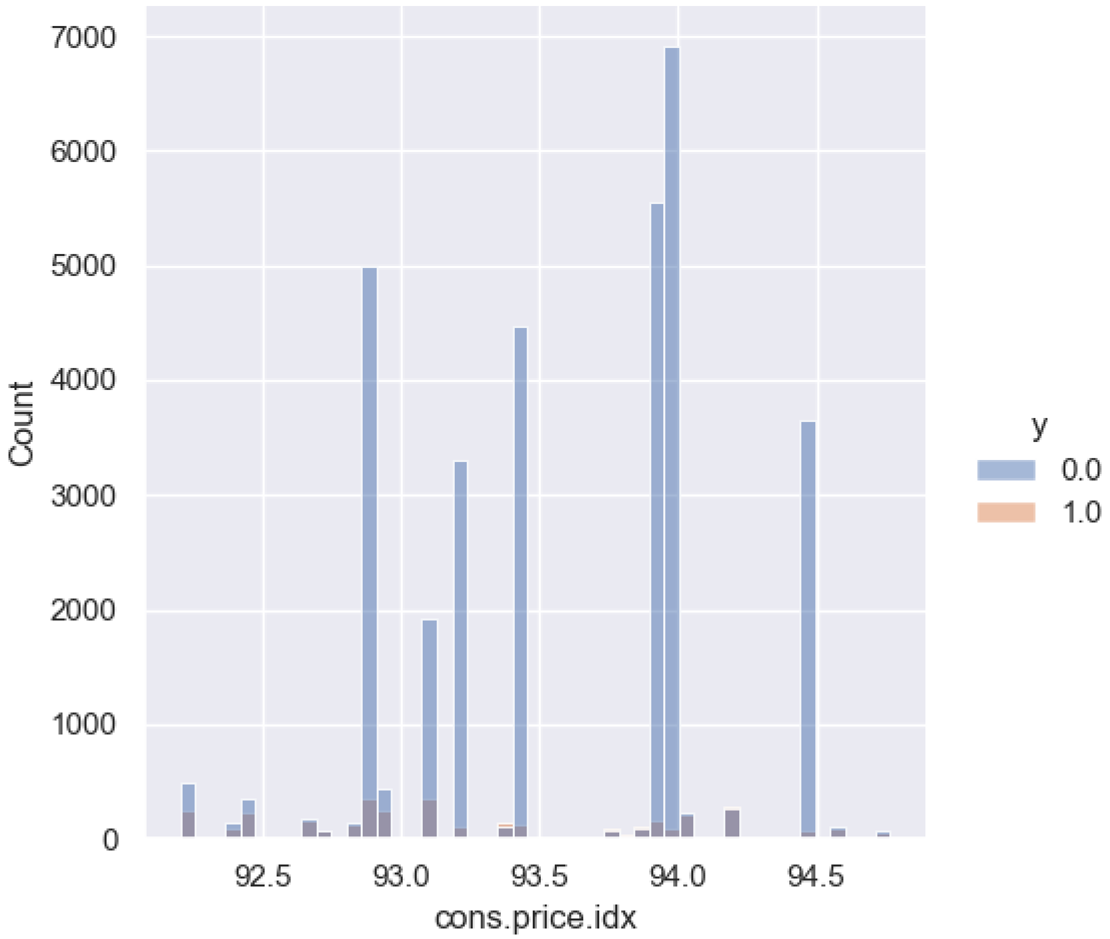



Count
1600
1400
1200
1000
800
600
400
200
0
0
200
400
600
duration
y
0.0
1.0

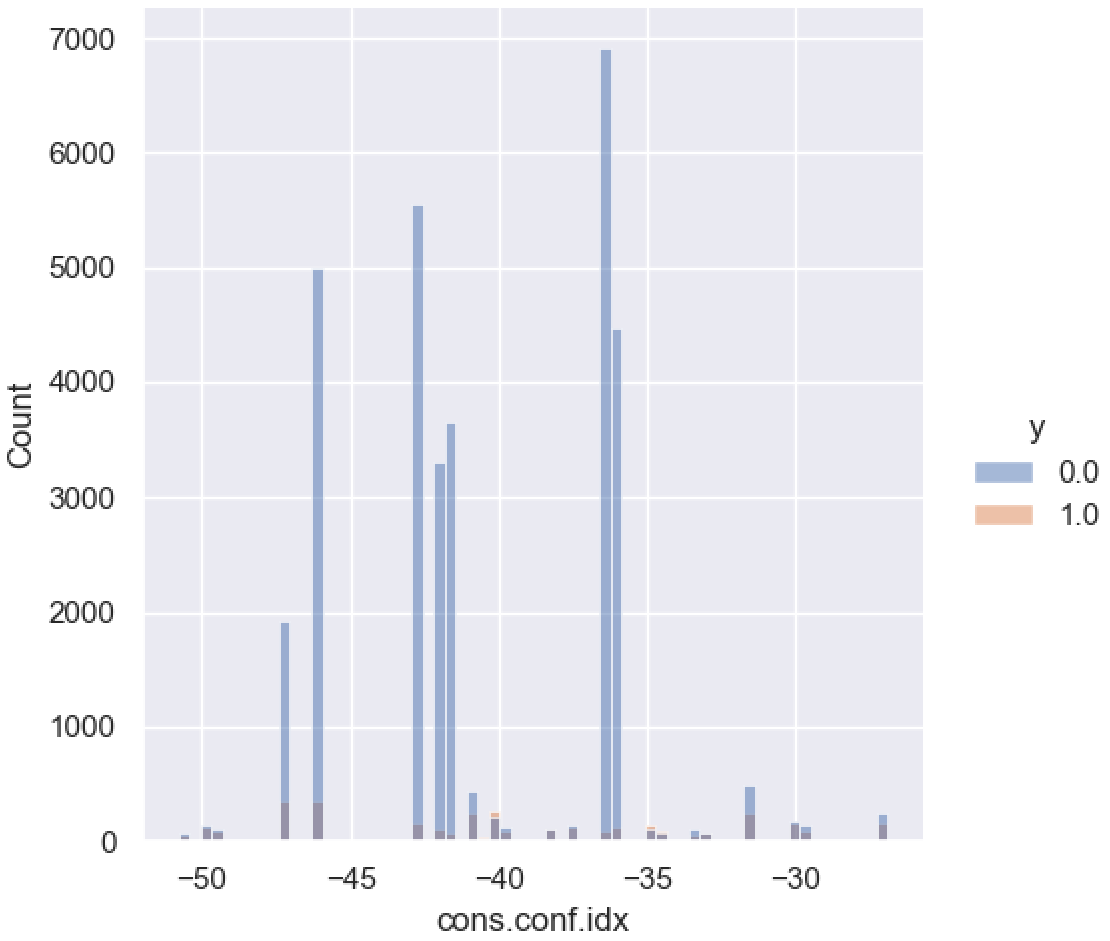

Count
14000
12000
10000
8000
6000
4000
2000
0
1
2
3
4
5
6
7
campaign
y
0.0
1.0

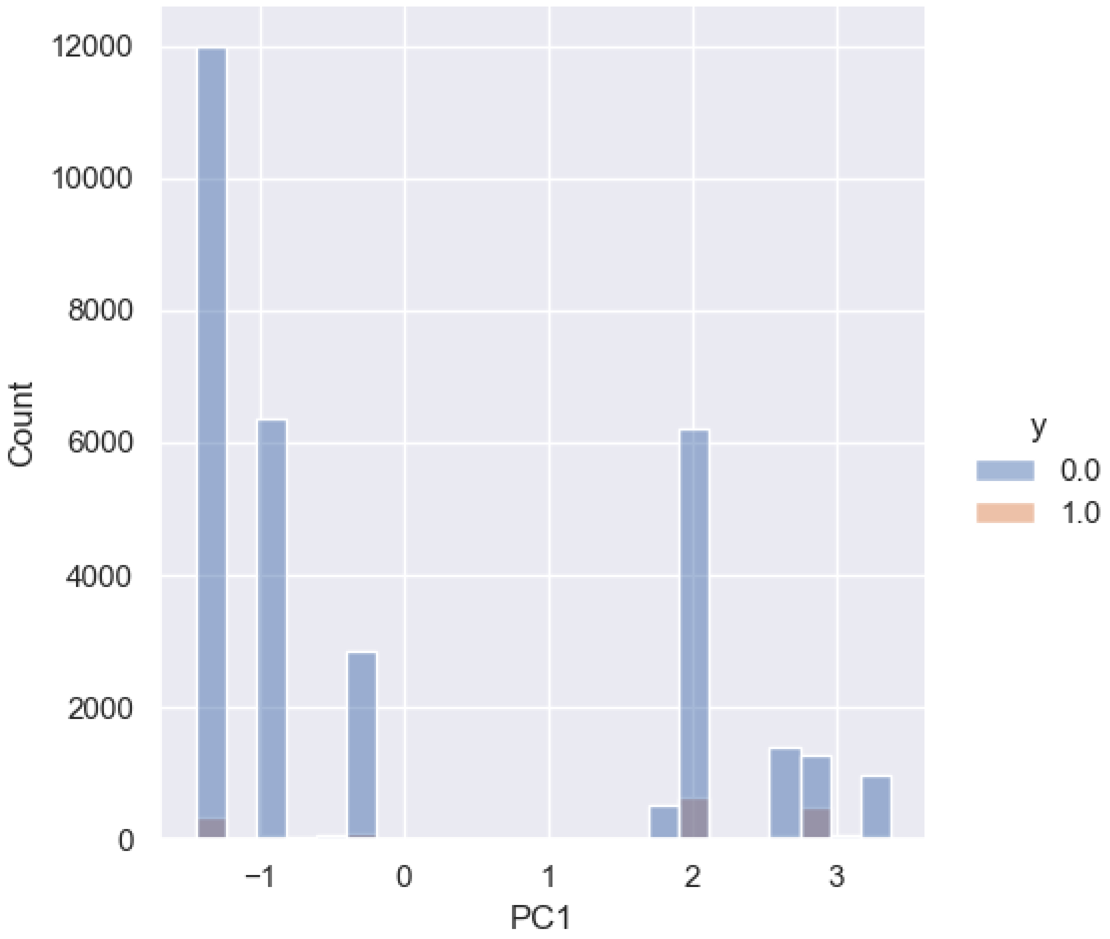


30000
25000
20000
15000
10000
5000
0
Count
0
1
2
3
4
5
6
7
previous
y
0.0
1.0



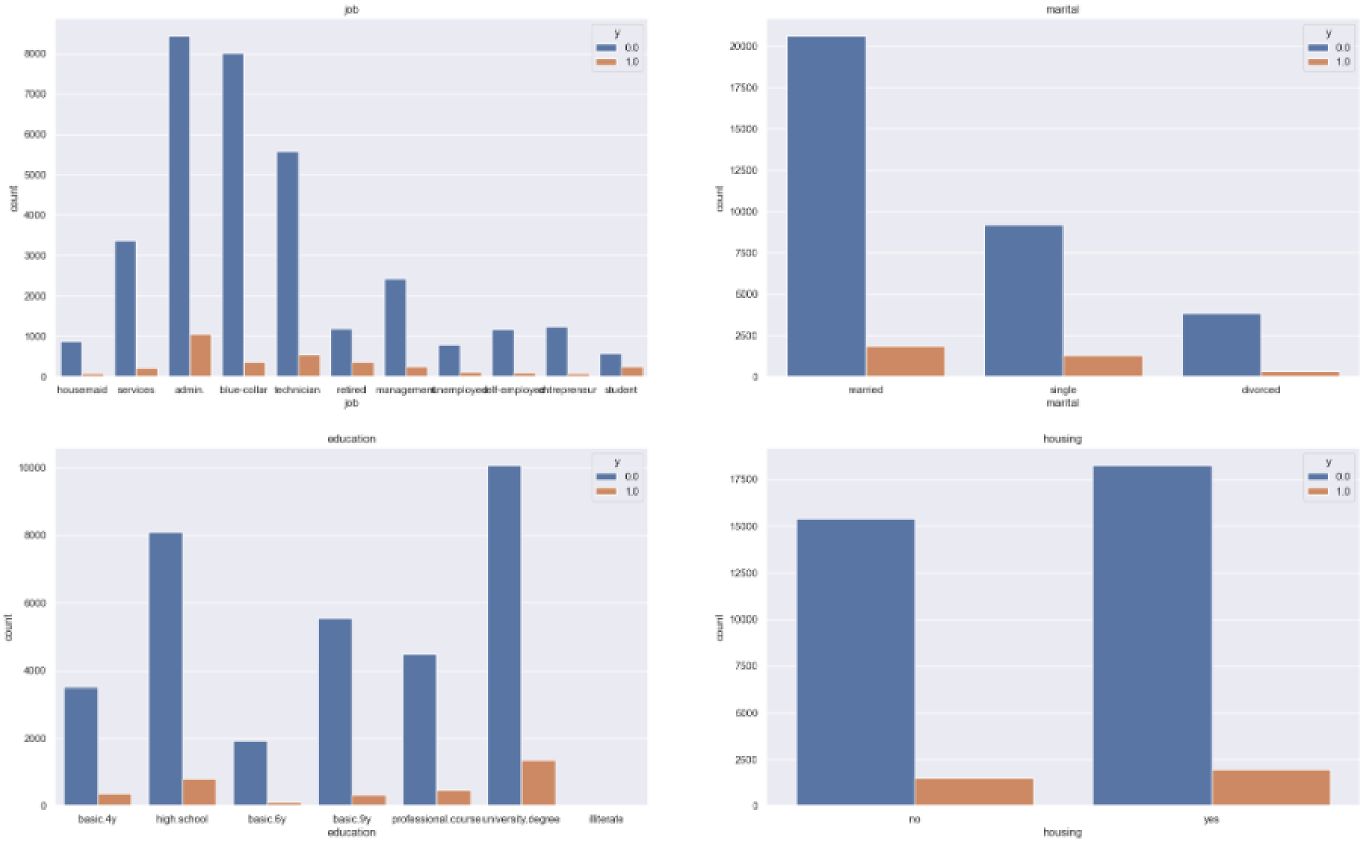
7000
6000
5000
4000
3000
2000
1000
0
Count
92.5
93.0
93.5
94.0
94.5
cons.price.idx
y
0.0
1.0

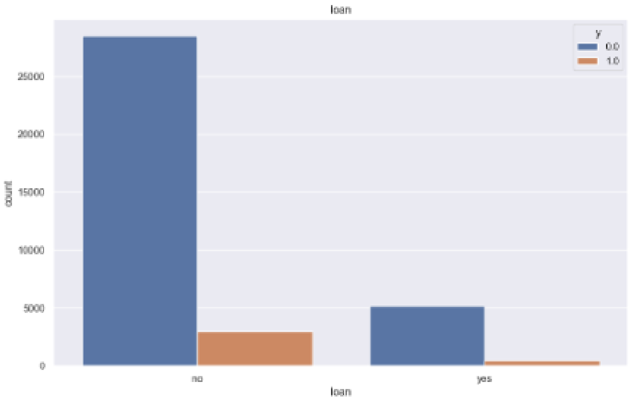

Count
7000
6000
5000
4000
3000
2000
1000
0
−50
−45
−40
−35
−30
cons.conf.idx
y
0.0
1.0

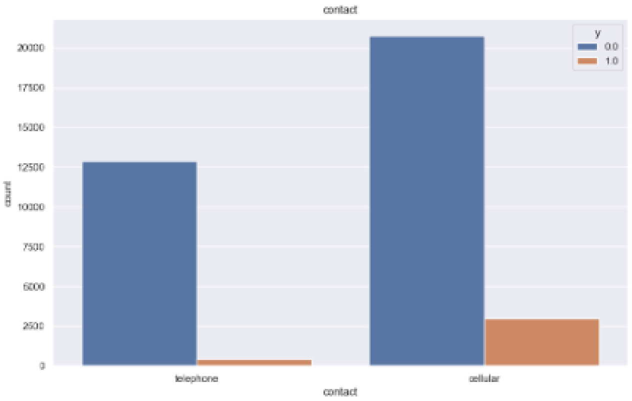

Count
12000
10000
8000
6000
4000
2000
0
−1
0
1
2
3
PC1
y
0.0
1.0

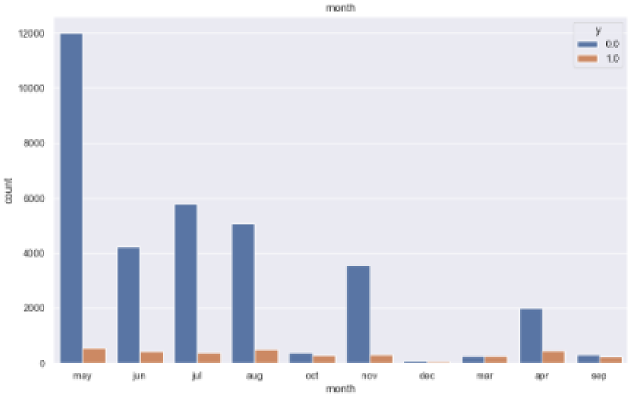
Observations 1:

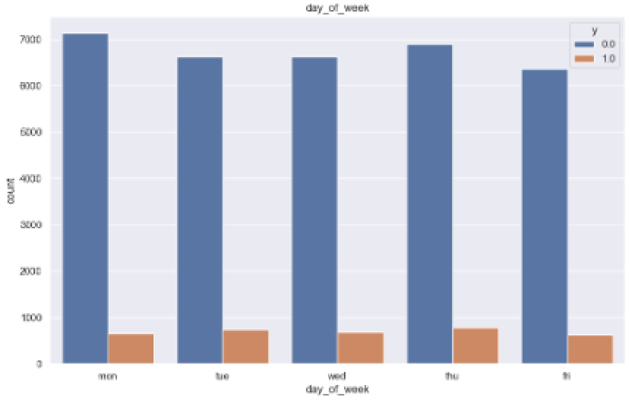
- Age: Most of the calls were made to people aged 25-50. Percentage of subscriptions seems to be approximately constant across all ages.
- Duration: As expected, percentage of subscriptions increases with the increase in call duration.
- Campaign: There seems to be almost no subscriptions for more than 4 contacts in the current campaign
- Previous: Data is heavily skewed to number of contacts = 1. Percentage of conversion seems to be consistent with overall value.
- cons.price.idx, cons.conf.index and PC1: There seems to be some correlation with target variable. But the trend is not clear, this may be clearer when coupled with month and year values.

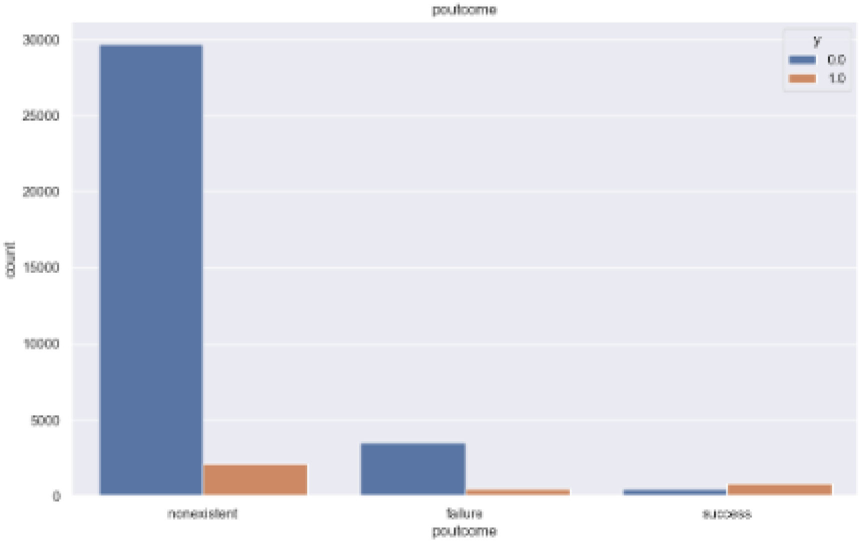



loan
count
0
5000
10000
15000
20000
25000
no
yes
loan
y
0.0
1.0


contact
count
0
2500
5000
7500
10000
12500
15000
17500
20000
telephone
cellular
contact
y
0.0
1.0


month
count
0
2000
4000
6000
8000
10000
12000
may
jun
jul
aug
oct
nov
dec
mar
apr
sep
month
y
0.0
1.0


day_of_week
count
0
1000
2000
3000
4000
5000
6000
7000
mon
tue
wed
thu
fri
day_of_week
y
0.0
1.0


poutcome
count
0
5000
10000
15000
20000
25000
30000
nonexistent
failure
success
poutcome
y
0.0
1.0

Observations 2:

- job: 'admin','blue-collar' and 'technician' jobs were contacted most. 'Retired' and 'Student' categories gave the highest percentage of subscriptions
- marital: most of the people contacted were married. The percentage of subscriptions didn't seem to change much with marital status
- education: most of the people contacted had either 'university. degree' or 'high. school' as their highest level of education. Though, 'illiterate' customers gave the highest percentage of subscriptions
- housing: There is no imbalance observed with respect to housing. The percentage of subscription also seems to be constant.
- loan: Most of the people contacted didn't have a personal loan. People without personal loan did seem to be more likely to subscribe but the difference between the two categories is small.
- contact: most of the people were contacted through a cellphone. This did result in a significantly higher percentage of subscriptions.
- month: Most of the contacts were made in the second quarter. Some months gave a significantly higher percentage of subscriptions than other months, but the trend is not very clear and there may be other factors at play here.
- day_of_week: Number of contacts and percentage of subscriptions doesn't seem to change much with day of the week.
- poutcome: The outcome of previous campaigns was "nonexistent" for most of the contacts. Although, the success of previous campaigns did seem to positively impact the subscriptions of current campaign.

3.0 EDA of Numerical Features (Yash Doshi and Anuj Singh)

Data cleaning of numerical features:

As discussed in the previous week's report, going forward, we will be using the bank-additionalfull.csv for this project.

Basic data exploration revealed that the dataset has 10 numerical features. In the data exploration performed in the last week, it was found that some of these features have outliers and some have high correlation amongst themselves. These issues need to be dealt with at this stage because these issues can significantly affect the accuracy and performance of the machine learning model that will be created later.

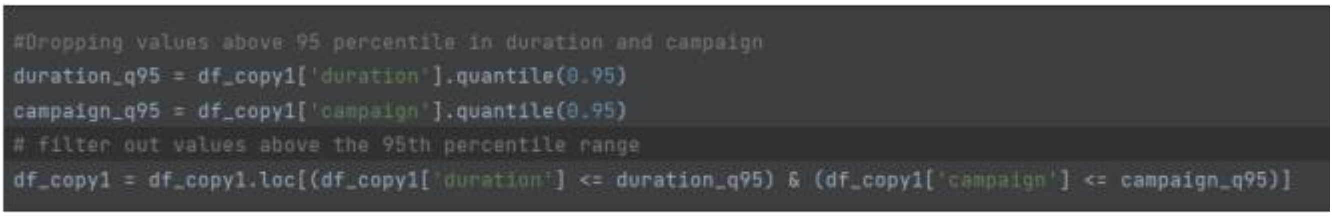
Outlier detection:

The following table shows the outliers for numerical features and rare categories for categorical features of the dataset. It also shows how we intend to deal with them.

Feature Name	Outliers/ Rare Categories	How to deal with them?
age	has 468 outliers greater than upper bound (69.5) or lower than lower bound(9.5).	Convert into categorical feature by binning the values.
job	7 rare categories: ['retired', 'entrepreneur', 'self-employed', 'housemaid', 'unemployed', 'student', 'unknown'].	These can be grouped into a single category.
marital	1 rare categories: ['unknown'].	This will not be changed/ transformed.
education	2 rare categories: ['unknown', 'illiterate'].	This will not be changed/ transformed.
default	1 rare categories: ['yes'].	This will not be changed/ transformed.
housing	1 rare categories: ['unknown'].	This will not be changed/ transformed.
loan	1 rare categories: ['unknown'].	This will not be changed/ transformed.
contact	No issue	
month	4 rare categories: ['oct', 'sep', 'mar', 'dec'].	This will not be changed/ transformed.
day_of_week	No issue	
duration	has 2963 outliers greater than upper bound (644.5) or lower than lower bound(-223.5).	Convert into categorical feature by binning the values.
campaign	has 2406 outliers greater than upper bound (6.0) or lower than lower bound(-2.0).	Records with values greater than 15 will be removed as outliers.
pdays	has 1515 outliers greater than upper bound (999.0) or lower than lower bound(999.0)	Since 999 is just a placeholder, it will be replaced by -1.
previous	has 5625 outliers greater than upper bound (0.0) or lower than lower bound(0.0).	This will not be changed/ transformed.
poutcome	1 rare categories: ['success'].	This will not be changed/ transformed.
emp.var.rate	No issue	
cons.price.idx	No issue	
cons.conf.idx	has 446 outliers greater than upper bound (-26.949999999999992) or lower than lower bound(-52.150000000000006).	This will not be changed/ transformed.
euribor3m	No issue	
nr.employed	No issue	
y	has 4639 outliers greater than upper bound (0.0) or lower than lower bound(0.0). Cap them or remove them.	Sampling methods will be used to deal with this imbalance

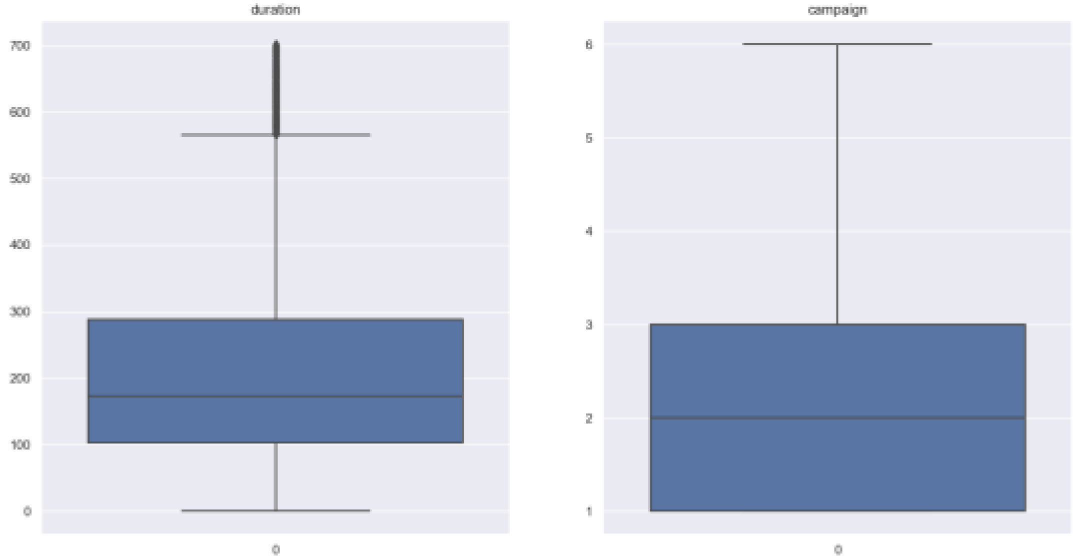
3.1 Dealing with outliers (Anuj Singh)

Checking the boxplots and the distribution plots of the numerical features revealed that three numerical features had a large number of outliers, namely, ‘duration’, ‘pdays’, and ‘campaign’. The outliers in the ‘duration’ and ‘campaign’ column were removed by only keeping the values less than the 95th percentile value for these features.

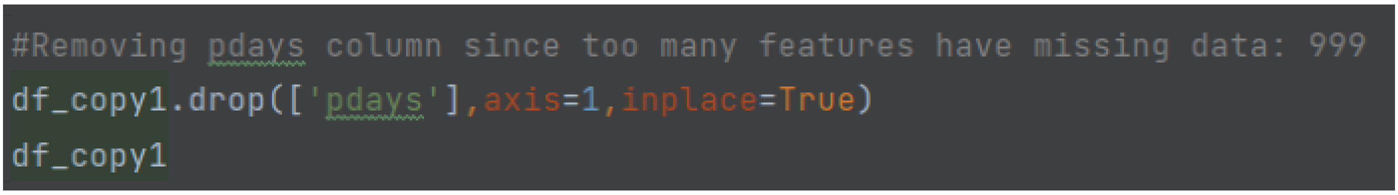


```
#Dropping values above 95 percentile in duration and campaign
duration_q95 = df_copy1['duration'].quantile(0.95)
campaign_q95 = df_copy1['campaign'].quantile(0.95)
# filter out values above the 95th percentile range
df_copy1 = df_copy1.loc[(df_copy1['duration'] <= duration_q95) & (df_copy1['campaign'] <= campaign_q95)]
```

The boxplots below reveal that majority of the outliers were successfully removed



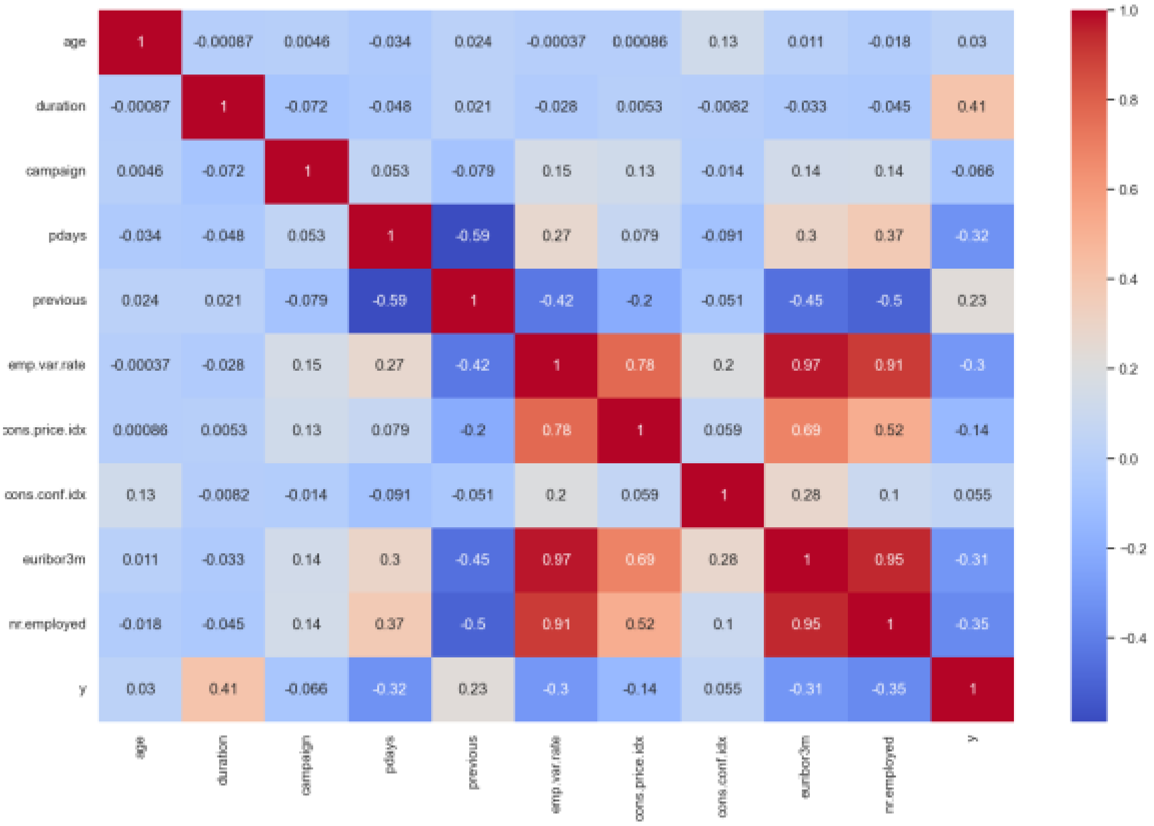
For the ‘pdays’ column, it was found that the outliers were due to 999 values, which were placeholders for missing data. Since more than 95% of the values are 999, we will drop the ‘pdays’ column entirely.



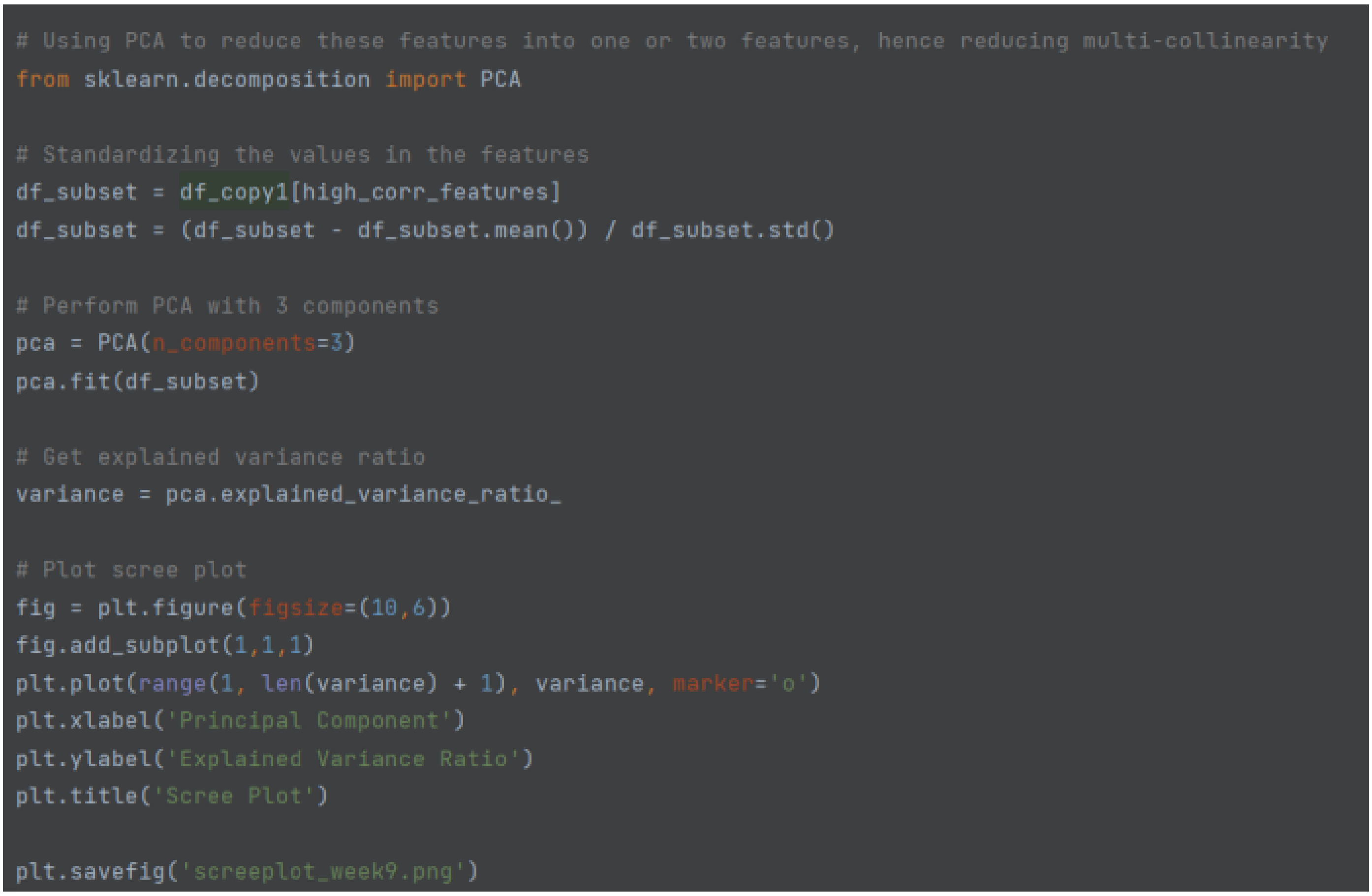
```
#Removing pdays column since too many features have missing data: 999
df_copy1.drop(['pdays'],axis=1,inplace=True)
df_copy1
```

3.2 Dealing with highly correlated features (Yash Doshi)

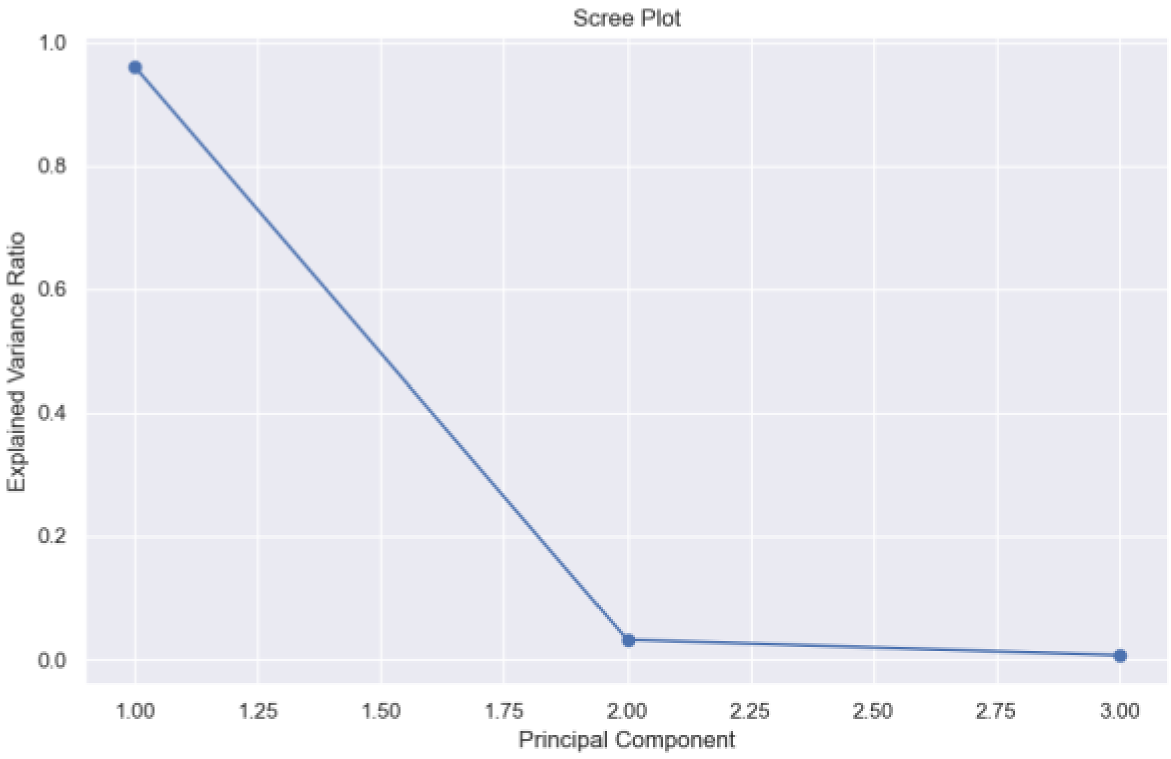
The heatmap of the Pearson's correlation matrix of the dataset shows that 'euribor3m', 'emp.var.rate', and 'nr.employed' are highly correlated. Note that the below figure also includes the target variable 'y', which was a categorical feature in the original dataset but was converted to a numerical feature using Label Encoding to study its correlation with the other numerical features.



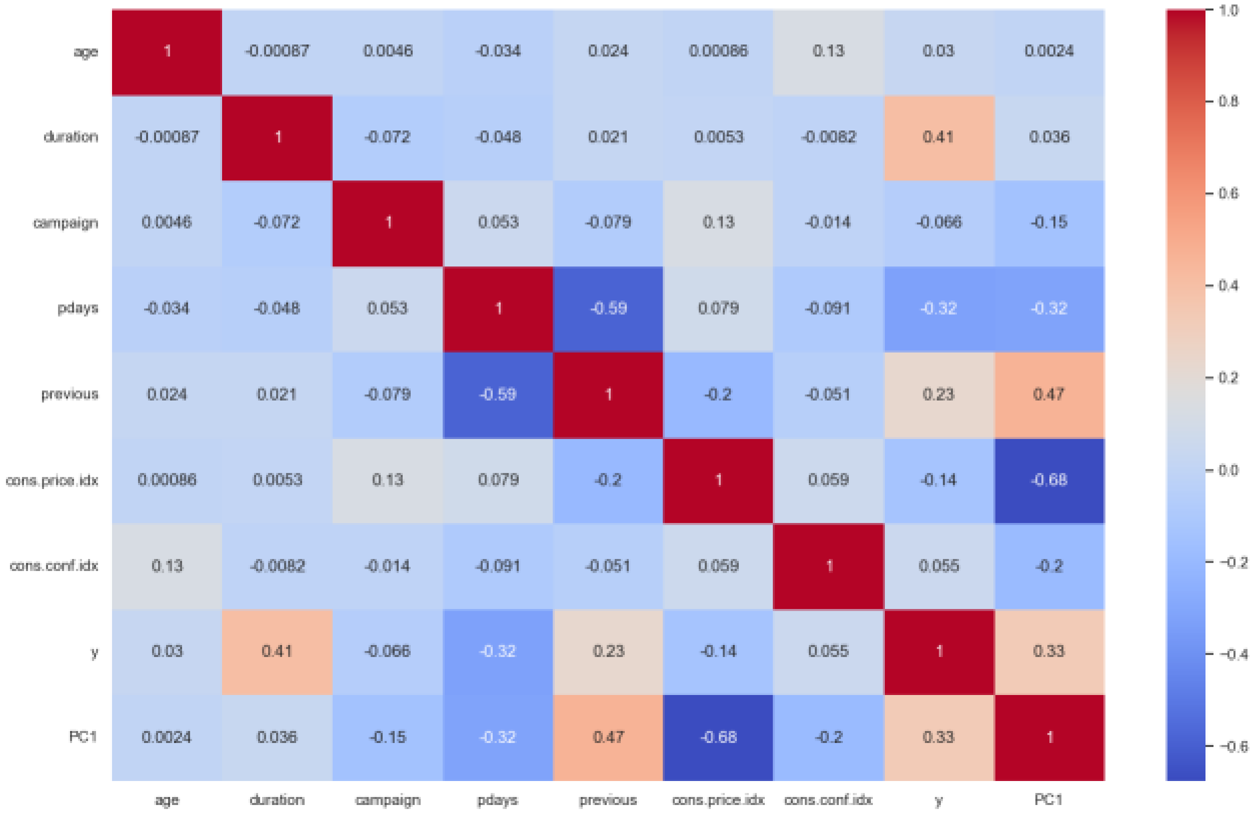
We will be performing Principle Component Analysis (PCA) on these highly correlated features to reduce them into one or two transformed features. The below Scree plot shows that one principle component explains more that 90% of the variance in the data and hence, the three highly correlated features can be transformed into one feature without losing the information of the three features.



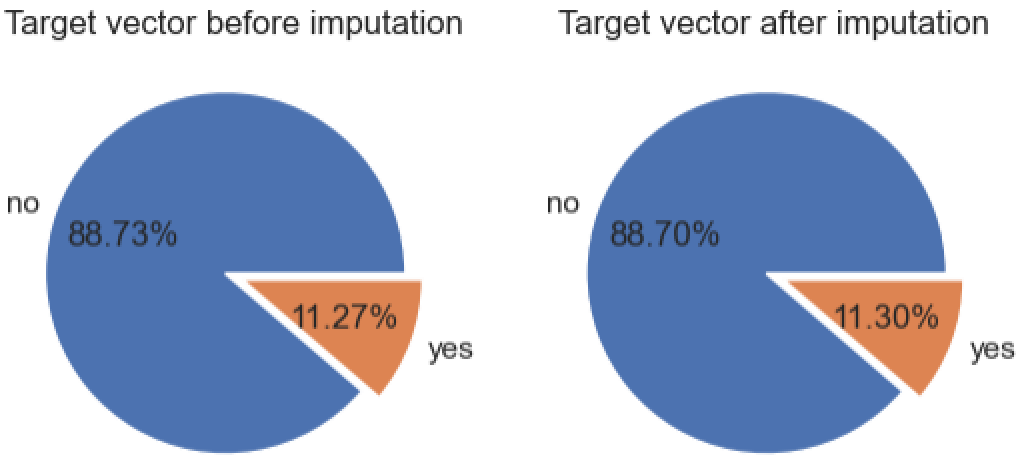
```
# Using PCA to reduce these features into one or two features, hence reducing multi-collinearity
from sklearn.decomposition import PCA

# Standardizing the values in the features
df_subset = df_copy1[high_corr_features]
df_subset = (df_subset - df_subset.mean()) / df_subset.std()

# Perform PCA with 3 components
pca = PCA(n_components=3)
pca.fit(df_subset)

# Get explained variance ratio
variance = pca.explained_variance_ratio_

# Plot scree plot
fig = plt.figure(figsize=(10,6))
fig.add_subplot(1,1,1)
plt.plot(range(1, len(variance) + 1), variance, marker='o')
plt.xlabel('Principal Component')
plt.ylabel('Explained Variance Ratio')
plt.title('Scree Plot')

plt.savefig('screeplot_week9.png')
```



After transforming the three highly correlated features into one feature using PCA, the below image shows the heatmap of the Pearson's Correlation matrix of the updated dataframe. It is evident from the similar correlation value between the new feature PC1 and y that the correlation problem has been solved without losing the information contained in the features.

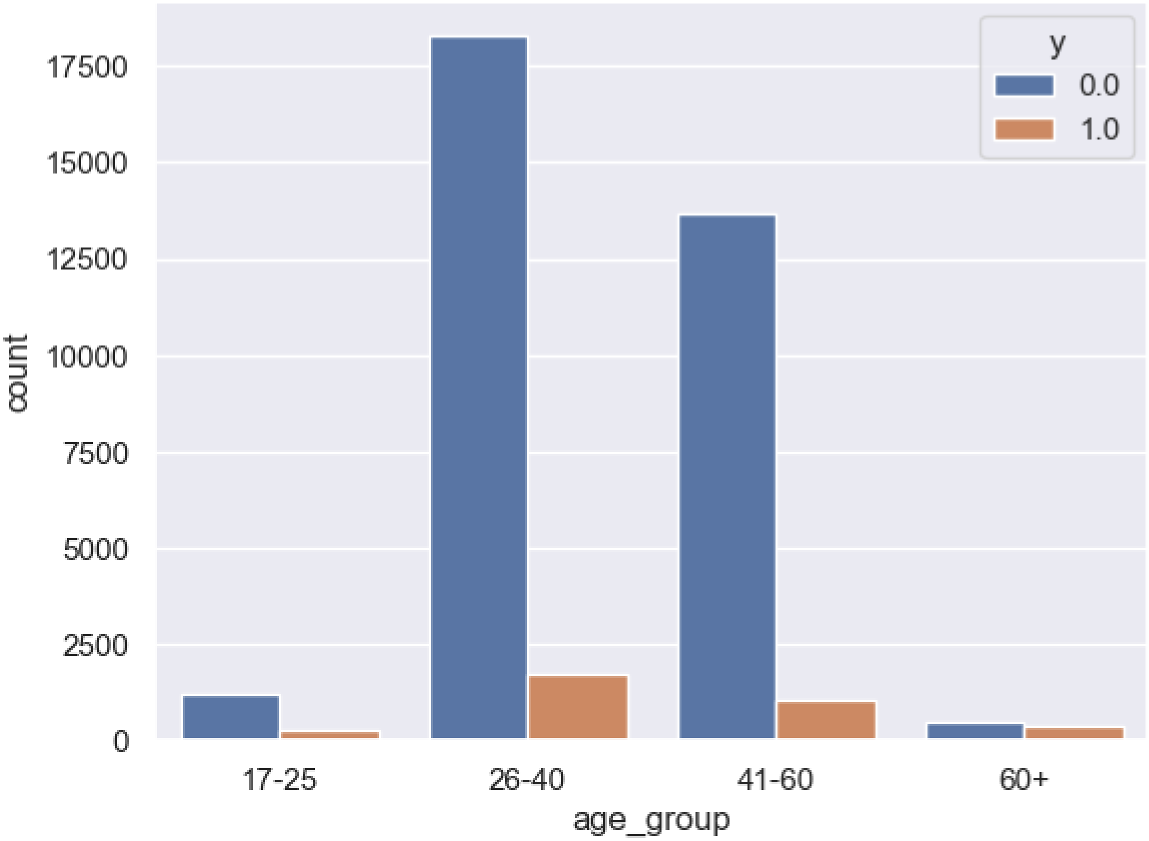


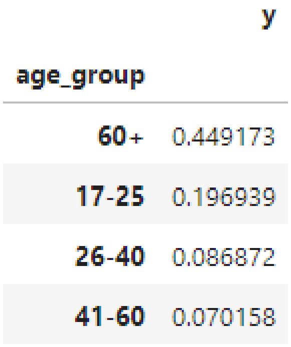
Target vector share before and after Imputation



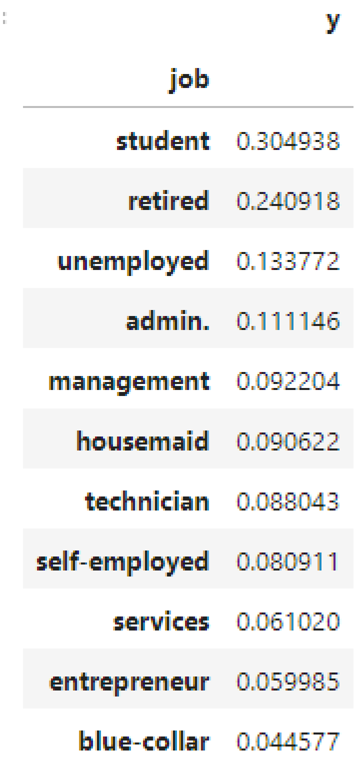
3.3 Individual feature analysis:

Age:

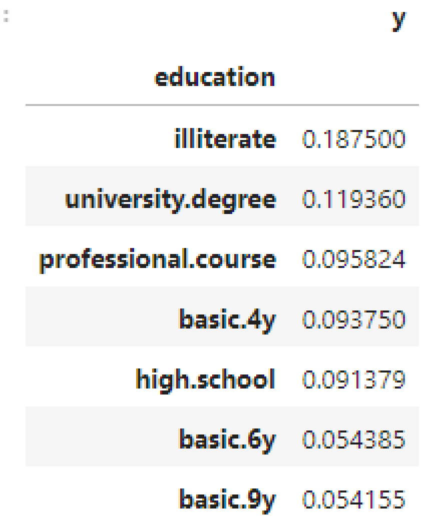




	y
age_group	
60+	0.449173
17-25	0.196939
26-40	0.086872
41-60	0.070158

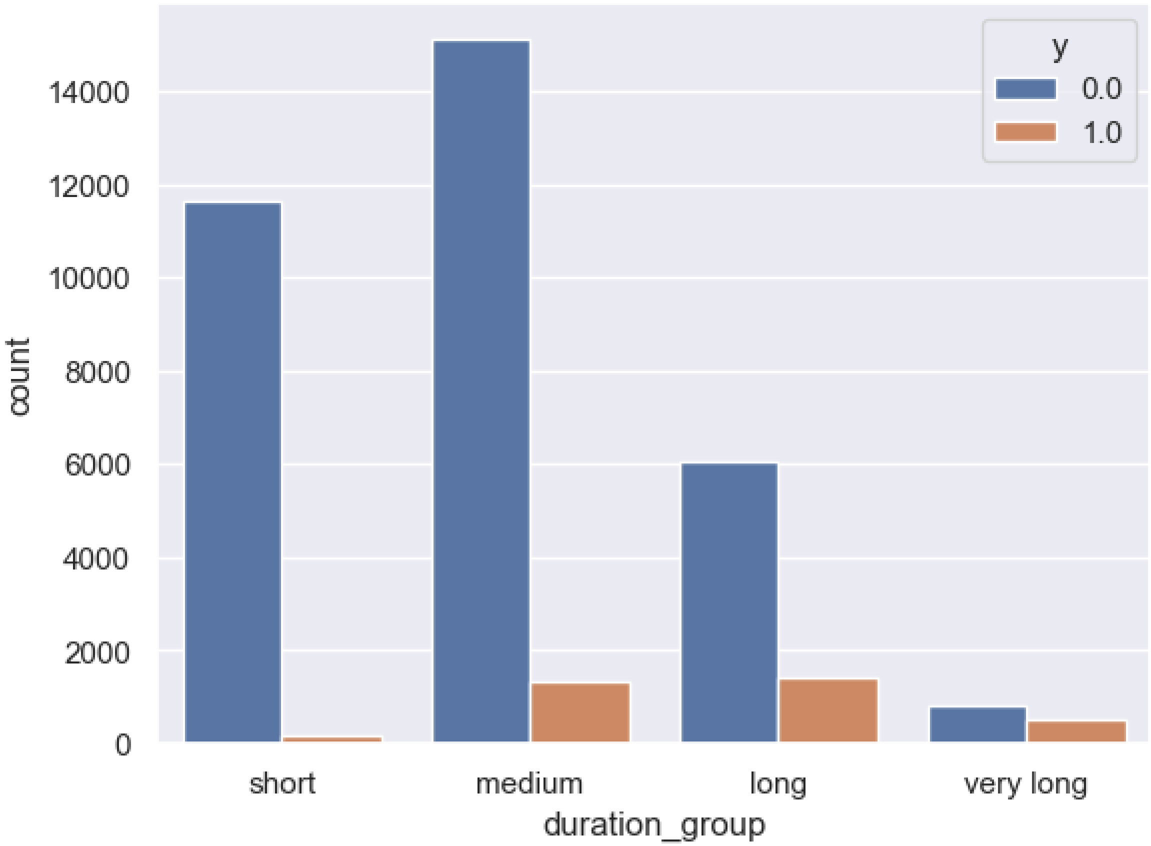


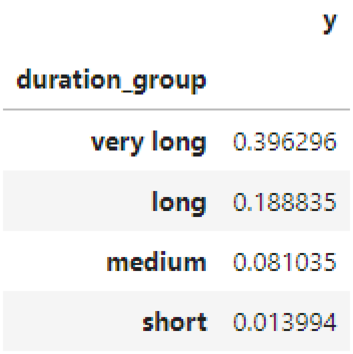
	y
job	
student	0.304938
retired	0.240918
unemployed	0.133772
admin.	0.111146
management	0.092204
housemaid	0.090622
technician	0.088043
self-employed	0.080911
services	0.061020
entrepreneur	0.059985
blue-collar	0.044577



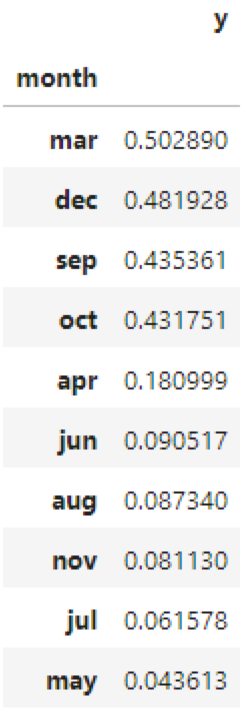
	y
education	
illiterate	0.187500
university.degree	0.119360
professional.course	0.095824
basic.4y	0.093750
high.school	0.091379
basic.6y	0.054385
basic.9y	0.054155

Duration:



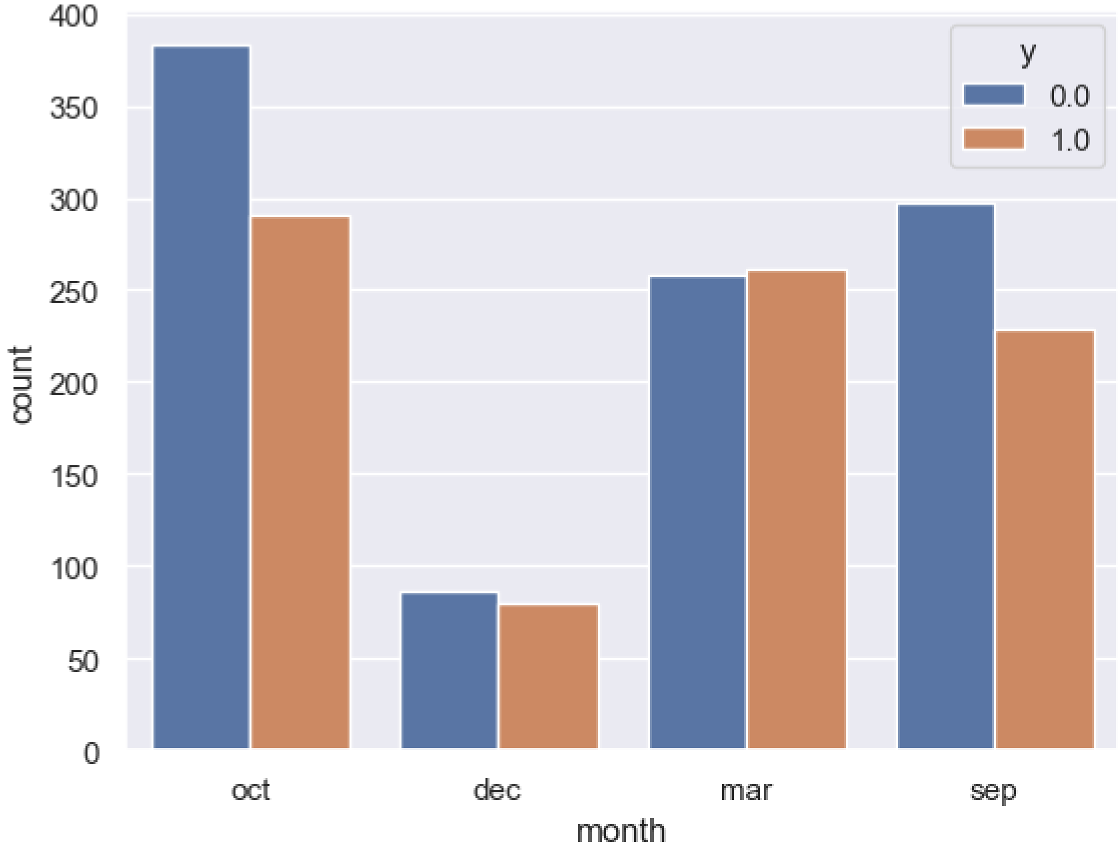


	y
duration_group	
very long	0.396296
long	0.188835
medium	0.081035
short	0.013994



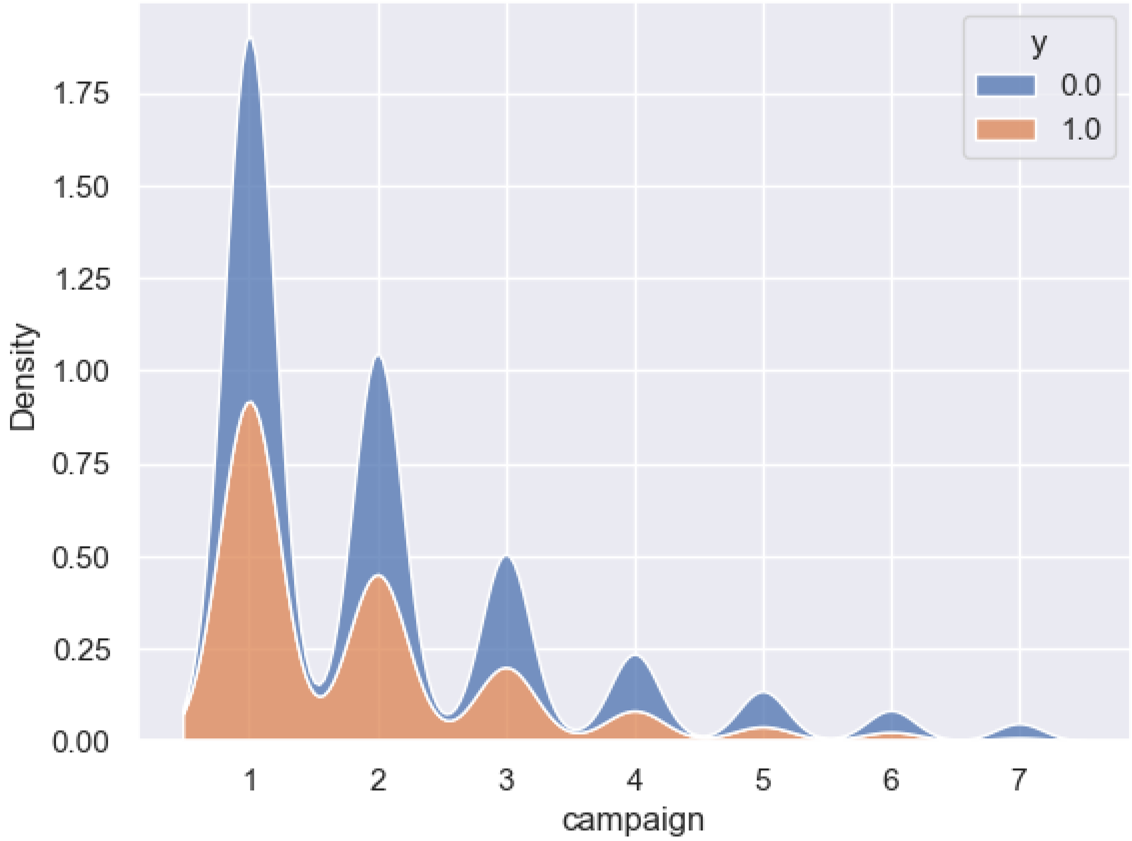
	y
month	
mar	0.502890
dec	0.481928
sep	0.435361
oct	0.431751
apr	0.180999
jun	0.090517
aug	0.087340
nov	0.081130
jul	0.061578
may	0.043613

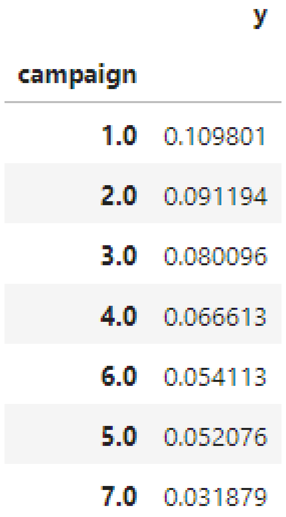
Month:



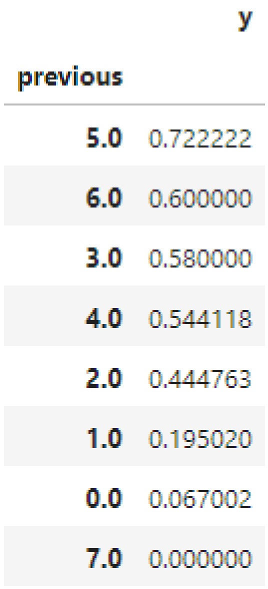
March had the most conversions for the bank's term deposit product despite only around 500 calls. The success may have been influenced by factors like interest rates, promotions, and sales representatives.

Campaign:

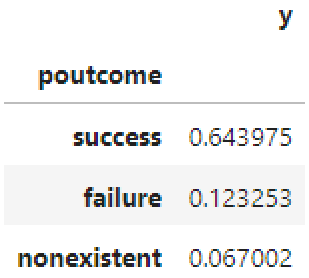




	y
campaign	
1.0	0.109801
2.0	0.091194
3.0	0.080096
4.0	0.066613
6.0	0.054113
5.0	0.052076
7.0	0.031879



	y
previous	
5.0	0.722222
6.0	0.600000
3.0	0.580000
4.0	0.544118
2.0	0.444763
1.0	0.195020
0.0	0.067002
7.0	0.000000

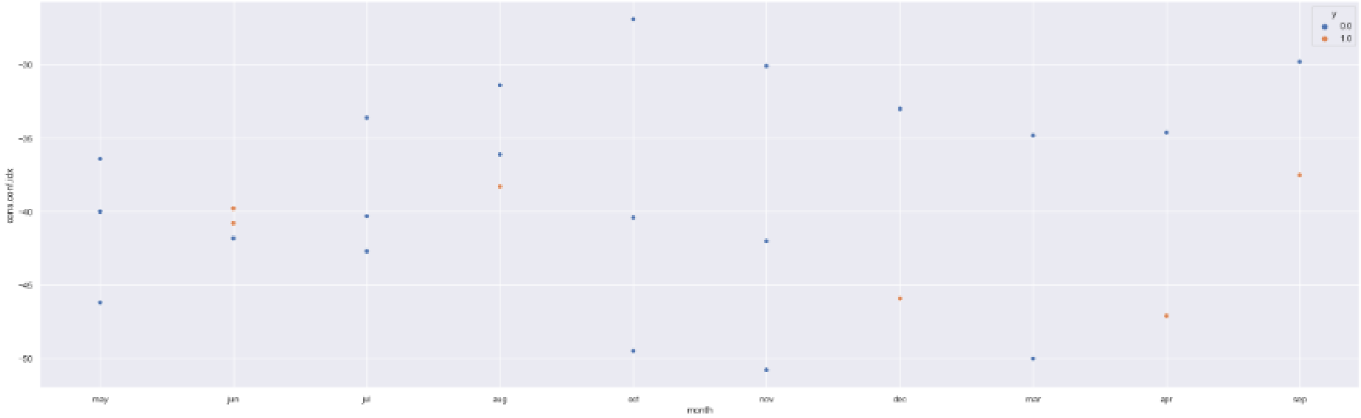


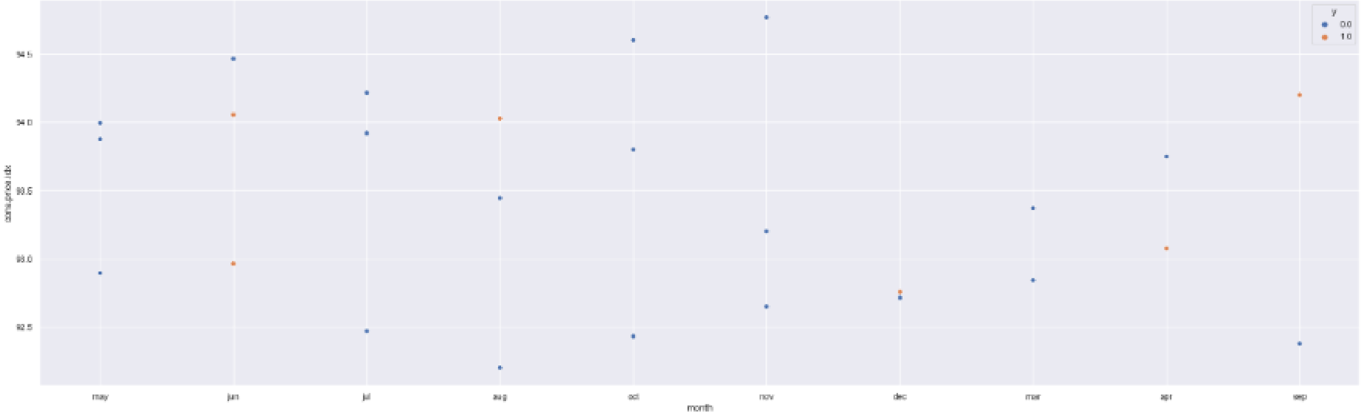
	y
poutcome	
success	0.643975
failure	0.123253
nonexistent	0.067002

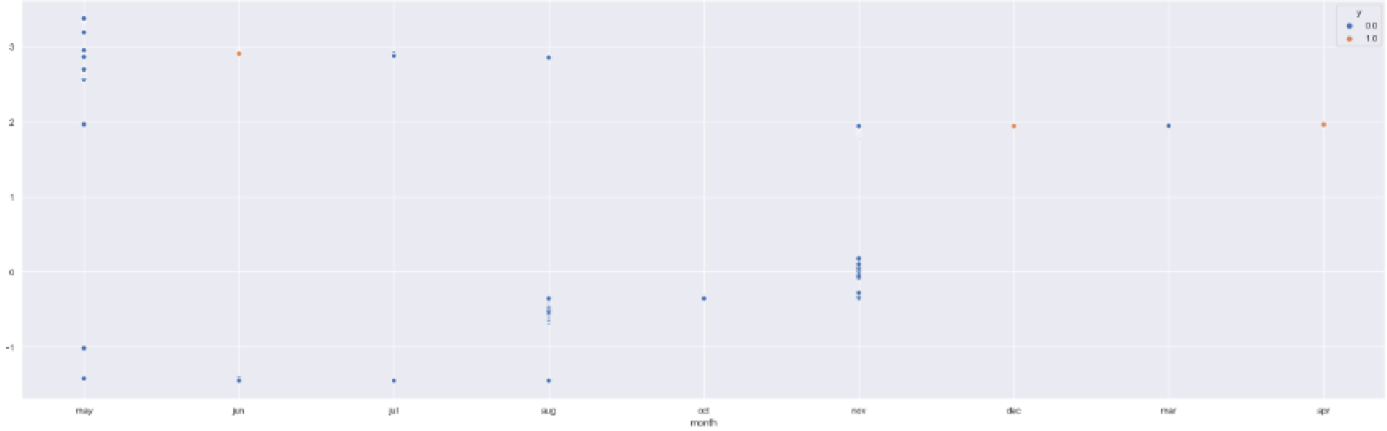
3.4 Hypothesis testing:

Testing correlation between month and economical indicators:

```
Relation between cons.conf.idx and month:
Chi-square statistic: 333819.0
P-value: 0.0
---------------------------------------------------------------------------------------
The p-value is below the threshold of 0.02. There is significant difference between mon
th and cons.conf.idx.
===============================================================================
Relation between cons.price.idx and month:
Chi-square statistic: 333818.99999999994
P-value: 0.0
---------------------------------------------------------------------------------------
The p-value is below the threshold of 0.02. There is significant difference between mon
th and cons.price.idx.
===============================================================================
Relation between PC1 and month:
Chi-square statistic: 155791.85127762248
P-value: 0.0
---------------------------------------------------------------------------------------
The p-value is below the threshold of 0.02. There is significant difference between mon
th and PC1.
===============================================================================
```


y
0.0
1.0
cons.conf.idx
−30
−35
−40
−45
−50
may
jun
jul
aug
oct
nov
dec
mar
apr
sep
month


y
0.0
1.0
cons.price.idx
94.5
94.0
93.5
93.0
92.5
may
jun
jul
aug
oct
nov
dec
mar
apr
sep
month


y
0.0
1.0
3
2
1
0
−1
may
jun
jul
aug
oct
nov
dec
mar
apr
month

References:

1. [Moro et al., 2011] S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimarães, Portugal, October, 2011. EUROSIS.

2. [Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

GitHub Repo Link:

https://github.com/singhanuj695/Data-glacier-Group-Project