Data Glacier

Your Deep Learning Partner

Exploratory Data Analysis

Bank Marketing (Campaign)

15 August 2021

Week 13:Final Project Report

Individual Project:
This project is done individually

Name:Nkululeko Freedom Mqadi

Email:mqadinf@gmail.com

Country:South Africa

College/Company:Deviare

Specialization:Data Science

Problem statement/Case Study

- 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).

- **Objective:**
- **What is the reason for developing a machine learning model?**
- **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.**

- **This will save resource and their time (which is directly involved in the cost (resource billing)).**

- The analysis is divided into the following sections:
- Business Understanding
- Data Understanding
- Data Exploration
- Exploratory Data Analysis
- Recommendation summary
- Model building
- Model selection

Business understanding

- ABC Bank is a banking institution in Portuguese that offers direct marketing campaigns which are phone call based.
- As the marketing campaigns are phone call based, this presented some limitations
- This resulted in decreasing the operating efficiency of the company.
- Contacting many people takes a lot of time and therefore success rate is less.
- The consumers that are being offered deposits, mostly are people using phone calls as a method of communication.
- This poses some threats in the company as this will jeopardize the performance of the company,
- So the company is coming up with innovative ways to increase the efficiency by making fewer calls but improves the success rate.

Data understanding

- The dataset consists of direct marketing campaigns data of a banking institution.
- Picked from UCI Machine Learning Repository which is an amazing source for publicly available datasets.
- There were four variants of the datasets out of which we chose “ bank-additional-full.csv”
- Consists of 41188 data points with 20 independent variables out of which 10 are numeric features and 10 are categorical features. The list of features available to us are given below:

Data set description:

- 1.**age (numeric)**
- 2.**job :** type of job (categorical: ‘admin.’,’blue-collar’,’entrepreneur’,’housemaid’,’management’,’retired’,’self-employed’,’services’,’student’,’technician’,’unemployed’,’unknown’)
- 3.**marital :** marital status (categorical: ‘divorced’,’married’,’single’,’unknown’; note: ‘divorced’ means divorced or widowed)
- 4.**education (categorical:** ‘basic.4y’,’basic.6y’,’basic.9y’,’high.school’,’illiterate’,’professional.course’,’university.degree’,’unknown’)
- 5.**default:** has credit in default? (categorical: ‘no’,’yes’,’unknown’)
- 6.**housing:** has housing loan? (categorical: ‘no’,’yes’,’unknown’)
- 7.**loan:** has personal loan? (categorical: ‘no’,’yes’,’unknown’)
- **Related with the last contact of the current campaign:**
- 8**. contact:** contact communication type (categorical: ‘cellular’,’telephone’)
- 9. **month:** last contact month of year (categorical: ‘jan’, ‘feb’, ‘mar’, …, ‘nov’, ‘dec’)
- 10. **day_of_week:** last contact day of the week (categorical: ‘mon’,’tue’,’wed’,’thu’,’fri’)
- 11. **duration:** last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y=’no’). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

other attributes:

12. **campaign:** number of contacts performed during this campaign and for this client (numeric, includes last contact)

13. **pdays:** number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)

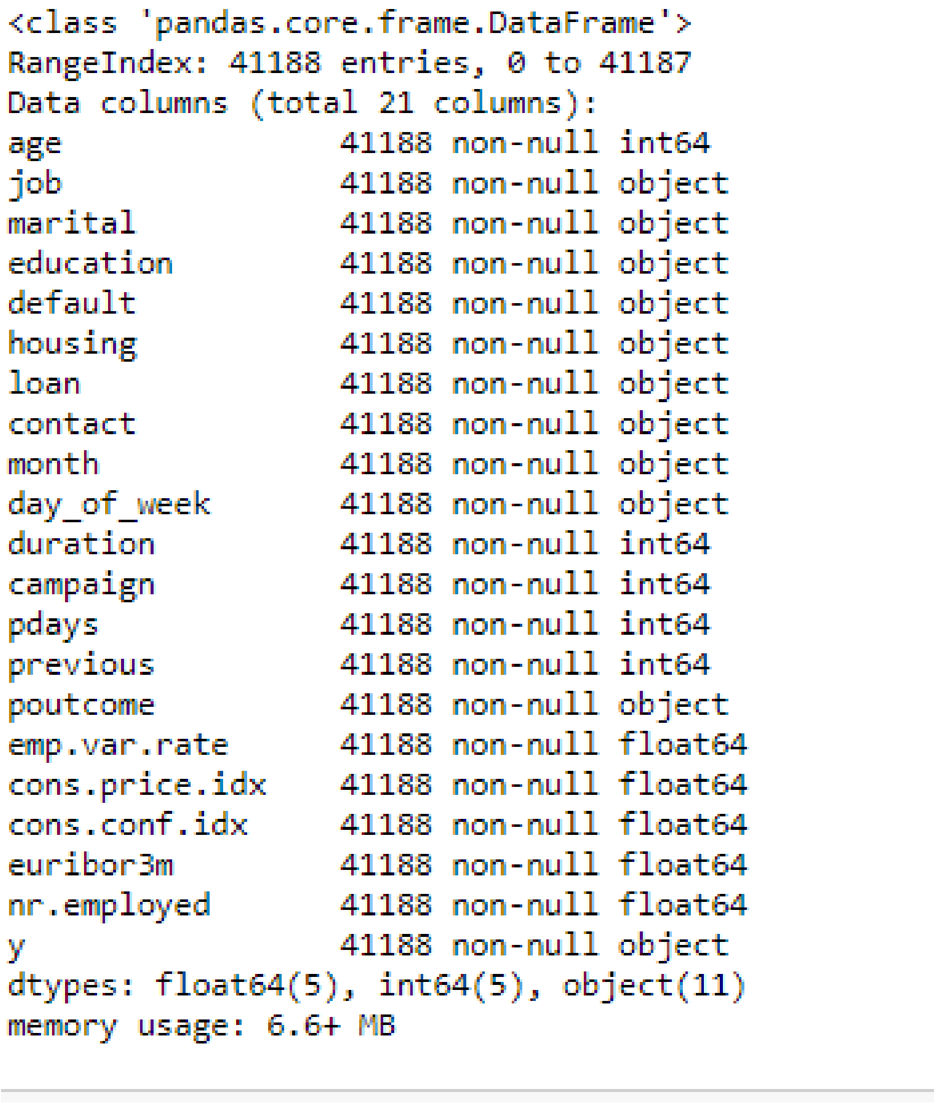
14. **previous:** number of contacts performed before this campaign and for this client (numeric)

15. **poutcome:** outcome of the previous marketing campaign (categorical: ‘failure’,’nonexistent’,’success’)

social and economic context attributes:

- 16. **emp.var.rate:** employment variation rate — quarterly indicator (numeric)
- 17. **cons.price.idx:** consumer price index — monthly indicator (numeric)
- 18. **cons.conf.idx:** consumer confidence index — monthly indicator (numeric)
- 19. **euribor3m:** euribor 3 month rate — daily indicator (numeric)
- 20. **nr.employed:** number of employees — quarterly indicator (numeric)
- **Output variable (desired target):**
- 21 - **y** - has the client subscribed a term deposit? (binary: 'yes','no')

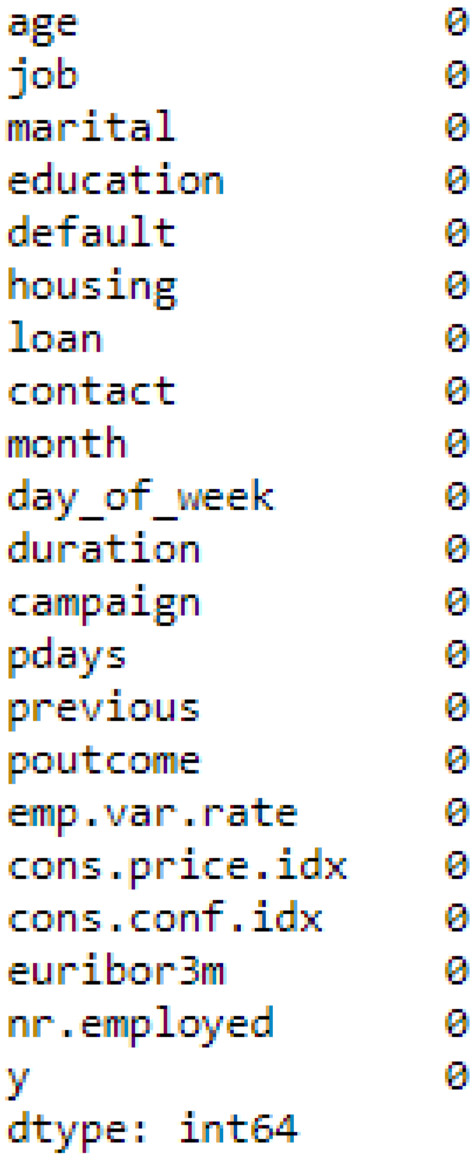
Data Exploration



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 21 columns):
age                 41188 non-null int64
job                 41188 non-null object
marital             41188 non-null object
education           41188 non-null object
default             41188 non-null object
housing             41188 non-null object
loan                41188 non-null object
contact             41188 non-null object
month               41188 non-null object
day_of_week         41188 non-null object
duration            41188 non-null int64
campaign            41188 non-null int64
pdays               41188 non-null int64
previous            41188 non-null int64
poutcome            41188 non-null object
emp.var.rate        41188 non-null float64
cons.price.idx      41188 non-null float64
cons.conf.idx       41188 non-null float64
euribor3m           41188 non-null float64
nr.employed         41188 non-null float64
y                   41188 non-null object
dtypes: float64(5), int64(5), object(11)
memory usage: 6.6+ MB
```

- Above we can see that there are 10 numerical columns having 5 columns of integer datatype and 5 columns on float datatype and 11 are categorical datatype including target variable which is named as y in the dataset.

Checking for null/missing values



```
age               0
job               0
marital           0
education         0
default           0
housing           0
loan              0
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
```

- There are no null/missing values in a dataset.

Type of data available for analysis:

- The type of data available for analysis is the **Exploratory Data Analysis (EDA).** The goal of using **Exploratory Data Analysis(EDA**) is:

- To explore data and find relationships between variables which were previously unknown.
- It’s also helps you to discover relationships between measures in your data, which are evidence for the existence of the correlation.

Exploratory Data Analysis

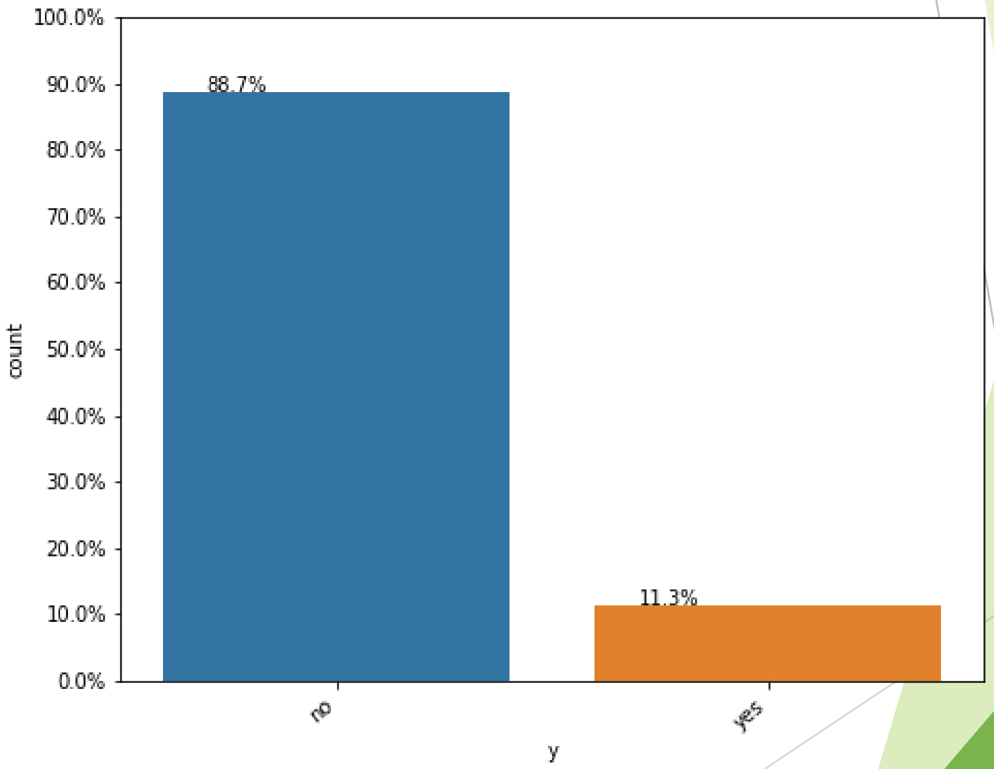
- Why **Exploratory Data Analysis(EDA) ?**
- **One of the reason for doing some Exploratory Data Analysis(EDA are as follows:**
- To ease the burden when doing some machine learning tasks. Machine learning tasks typically start with a comprehensive exploration of the datasets.
- Its helps ML practitioners to gain a deeper understanding of the properties of the data(schema, statistical properties, and so on),the quality of the data(missing values, inconsistent data types, and so on) and the predictive power of the data(for example, the correlation of features with the target).
- Can accomplish the task of performing some descriptive analysis which provides an understanding of the characteristics of the dataset and performing some visualizations which presents data in a pictorial or graphical format.

- I’m going to perform some Exploratory Data Analysis(EDA) using the **“bank-additional-full.csv”** dataset.
- With the help of the libraries such as matplotlib and seaborn, our goal to perform some exploratory data analysis can be easily accomplished. I will reveal some insights which I’ve developed.

EDA

Categorical Variables:

- I will first perform some exploratory data analysis on the categorical variables.
- **In which class does majority data points belongs to? Yes, or no?**
- As shown from the plot, we can see that majority of datapoints belong to No class labels with 88.7% and minority of class belongs to 11.3% so the ratio of No:Yes is 8:1.

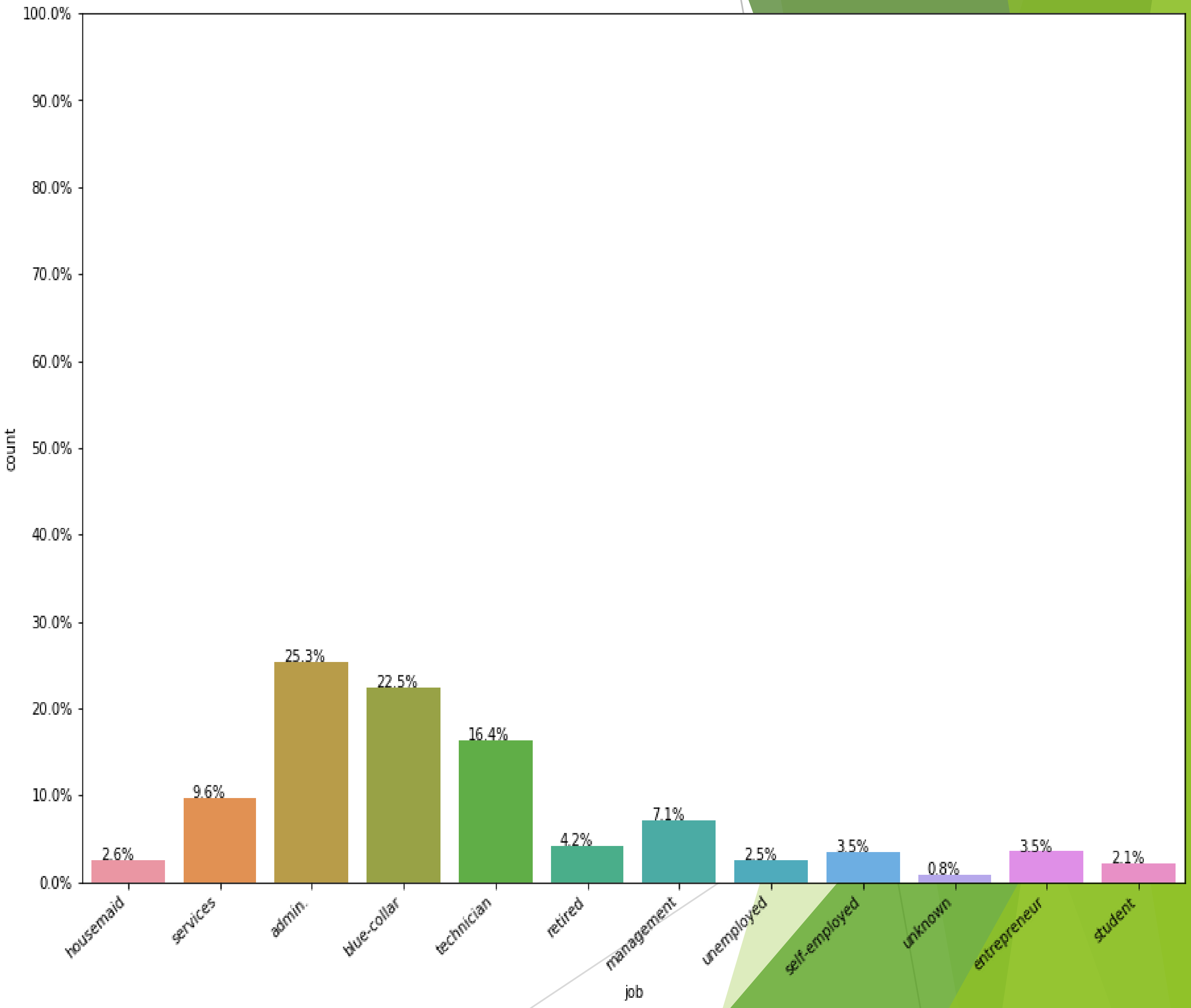


Univariate Analysis EDA

Let’s start doing EDA on rest of the columns of the datapoints.

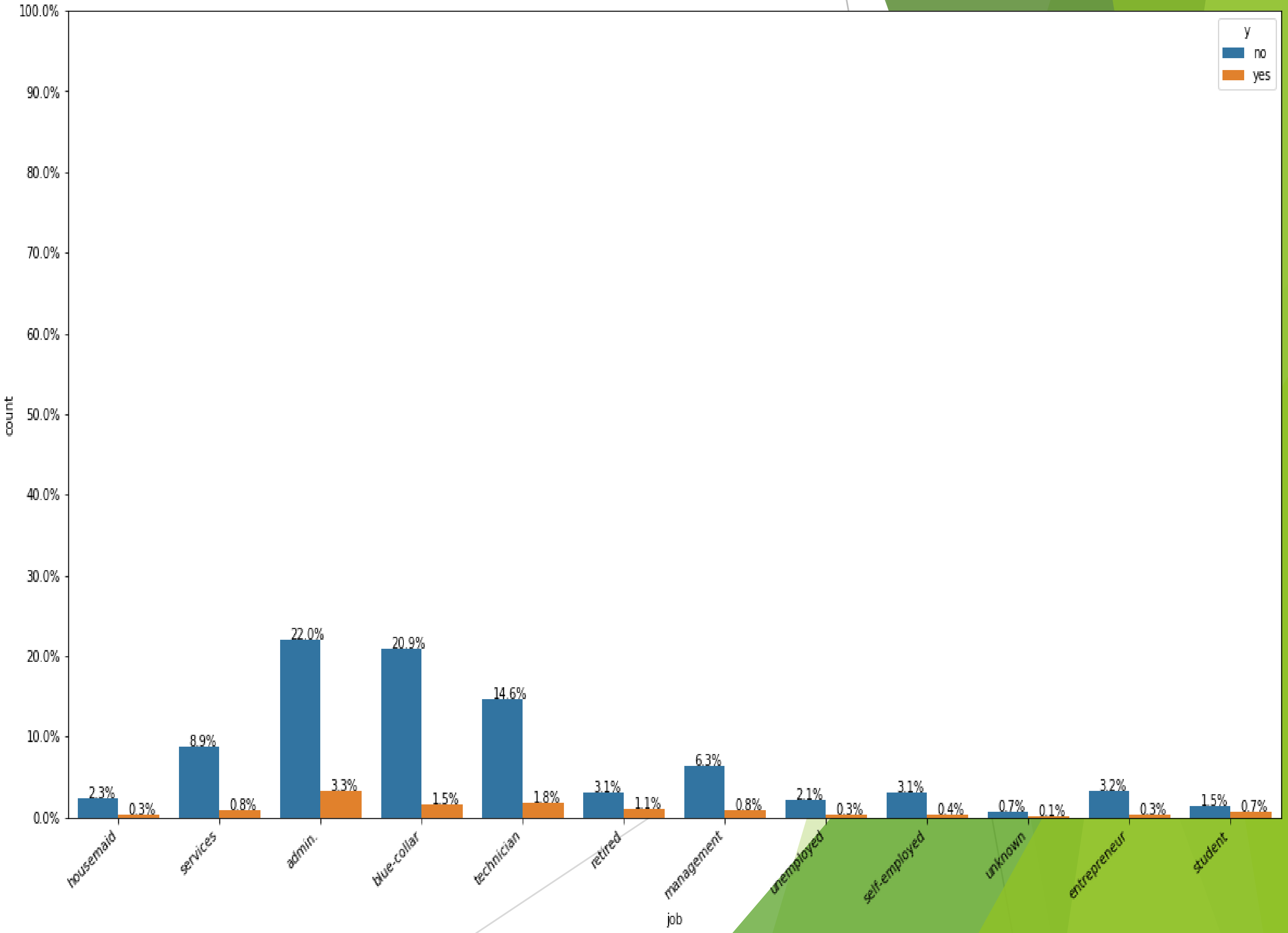
Feature: Job (Categorical variable)

- From the graph we can see that most of the customers that have jobs as **"admin", "blue-collar"** or **"technician"** have been contacted by the bank.
- One interesting thing to find out would be to see the distribution for each classes as well.
- For example, how many people who work as an admin have subscribed a term deposit.



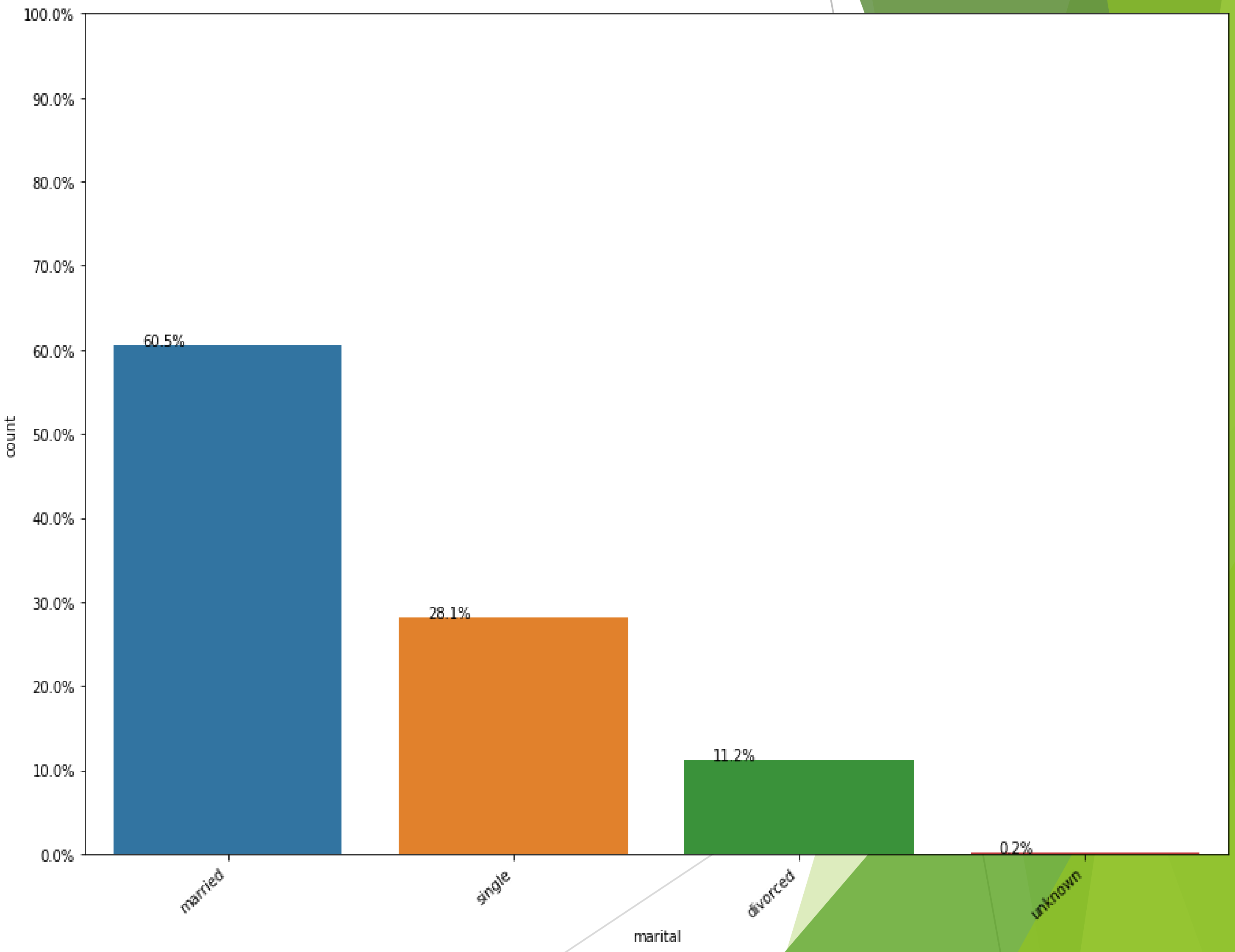
Which job profession has the highest rate for subscribing a term deposit and which has the highest rate of not subscribing?

- From the plot, we can see that the customers who have jobs of admin have the highest rate of subscribing a term deposit
- They are also the highest when it comes to not subscribing.
- This is simply because we have more customers working as admin than any other profession.



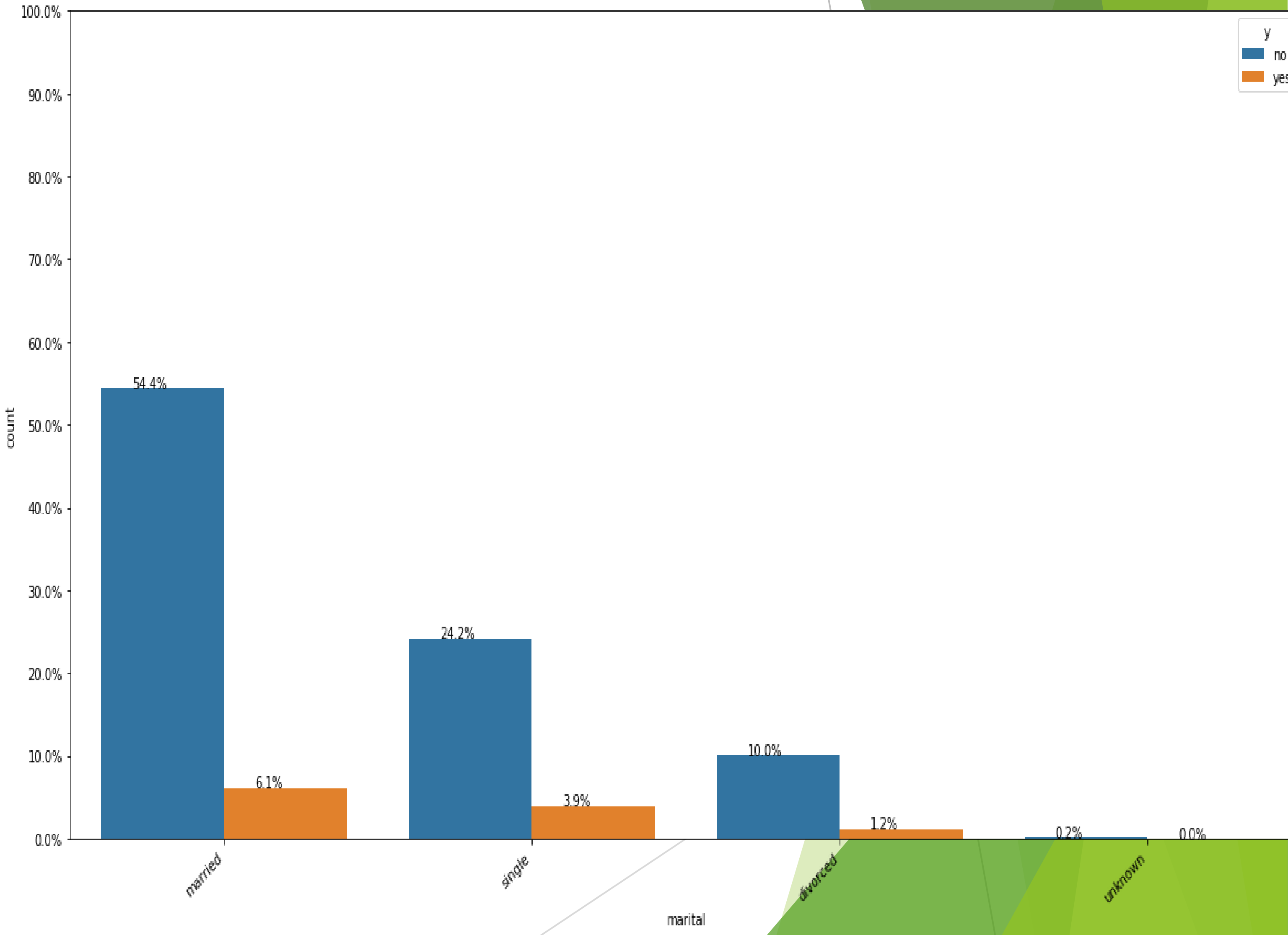
➢ Feature: Marital (Categorical feature)

- From the plot as we can see, customers who have been contacted the most are married.
- About 0.2% of marital status of customer is unknown.

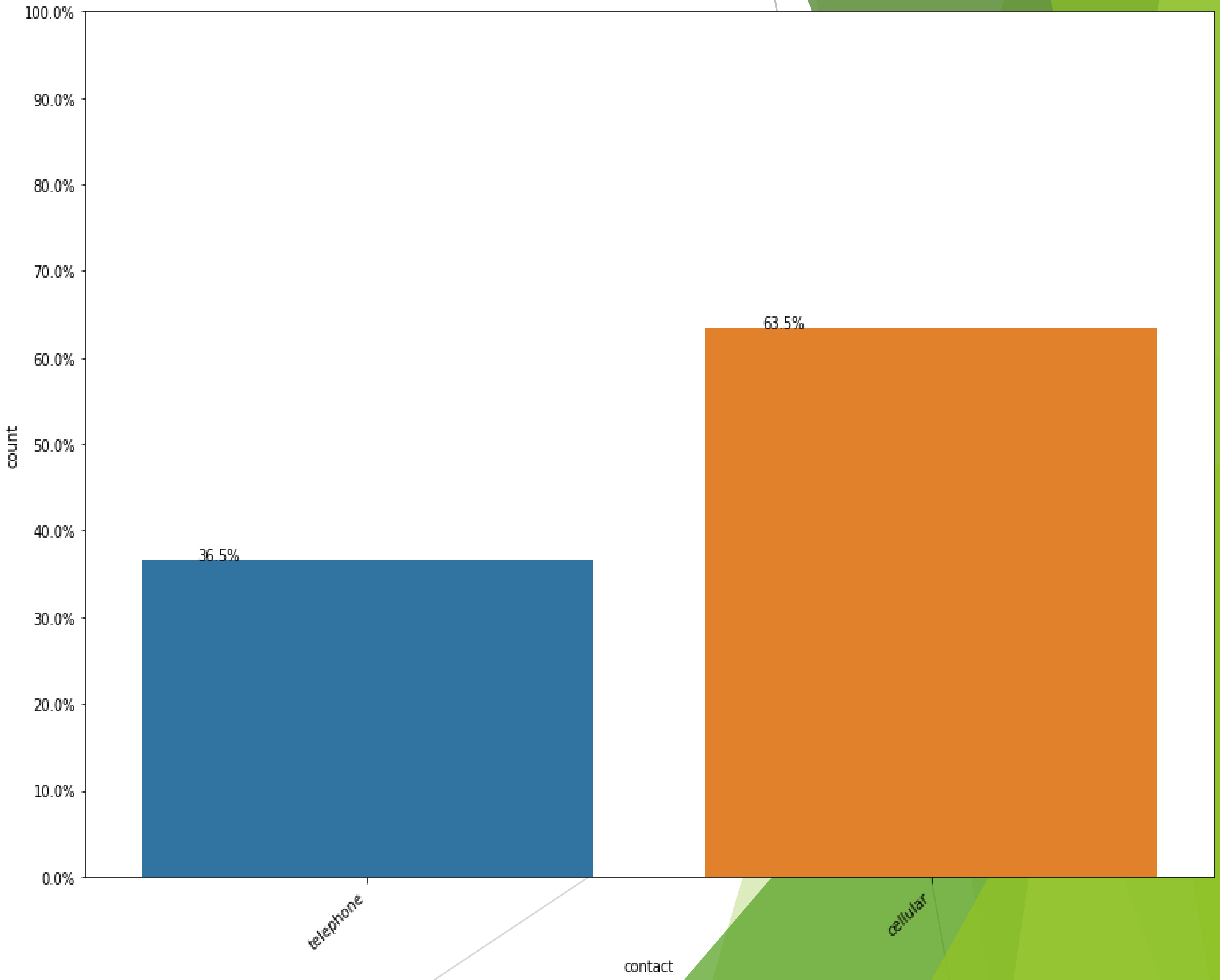


Which marital status subscribes the most for long term deposits?

- From the plot we can see that married people has subscribed for long term deposits more than any other people with any marital status.
- They are also the most one's who have turned down the deposits offered by the bank.
- People whose status is unknown has not subscribed to the long-term deposits at all.



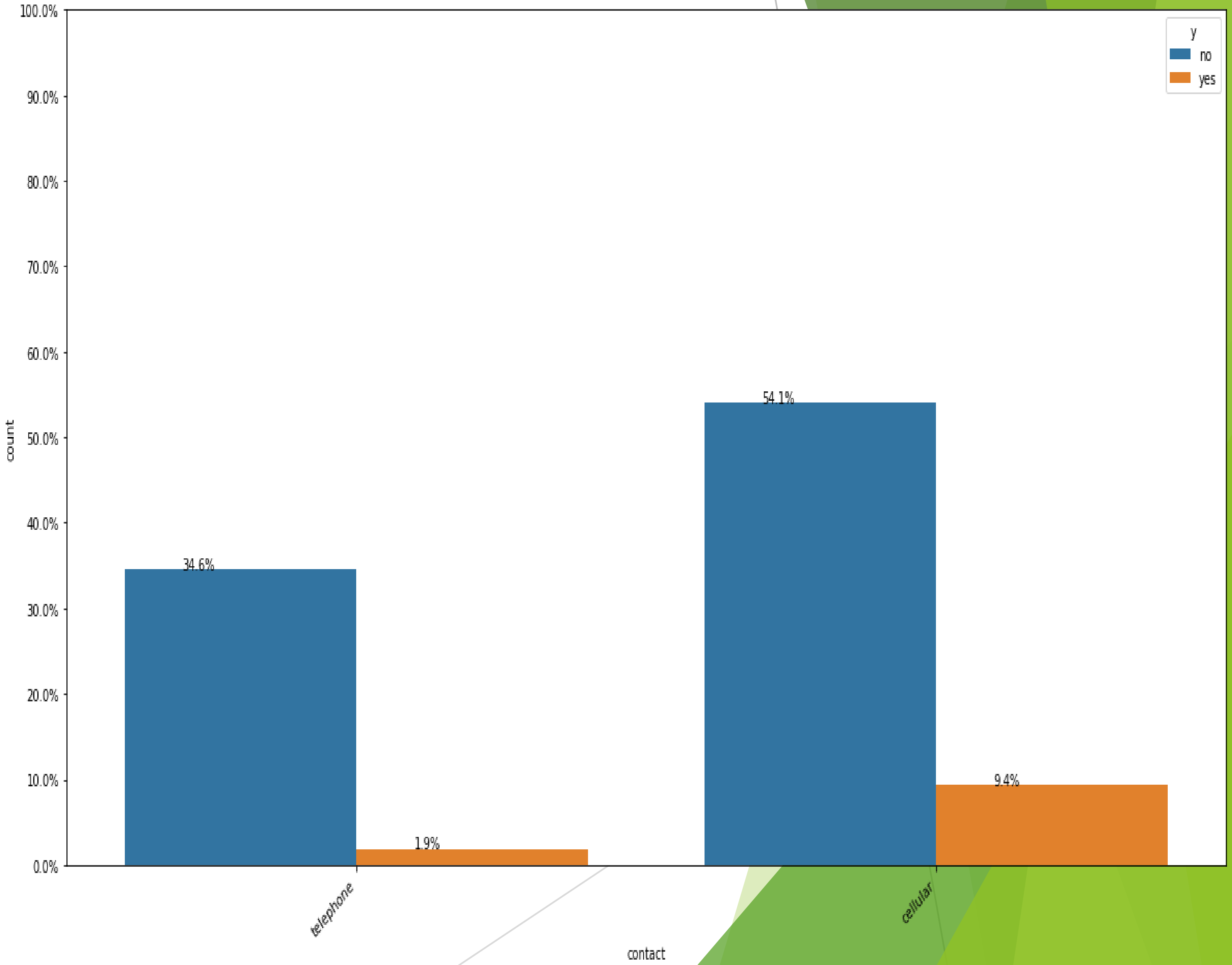
➢ Feature: contact (Categorical)



- As shown above, most people are contacted more in cellular than telephone.

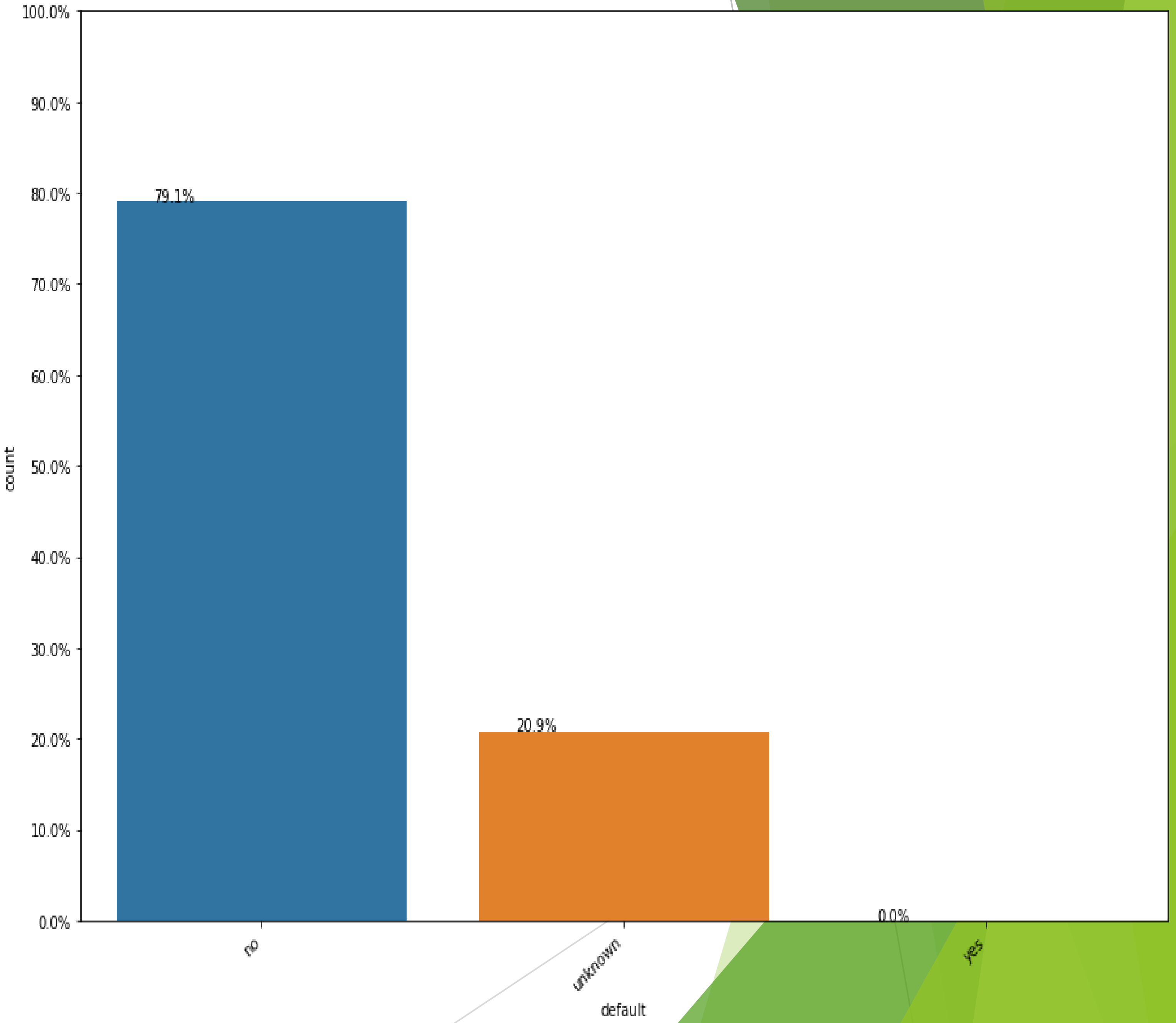
Which contact type has subscribed the most for long term deposits?

- People with contact type cellular have subscribed more for long term deposits than telephone.



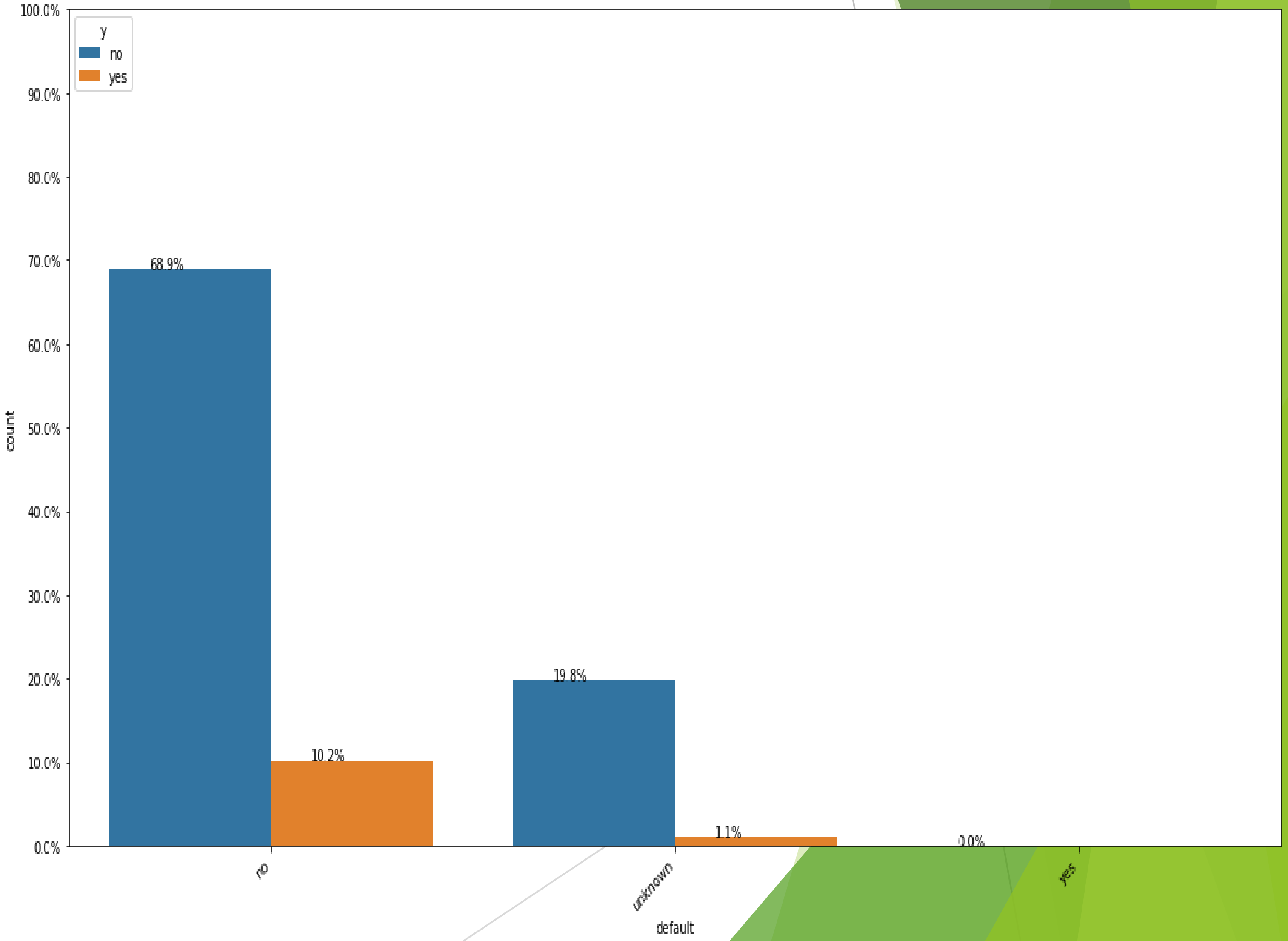
➢ Feature: default (categorical)

- As shown from the plot, we can clearly see that the people with default status as ‘no’ are the most who have been contacted by the bank for the deposits.
- People with default status ‘yes’ have not been contacted by the bank at all.
- While very few people with unknown default status have been contacted by the bank.



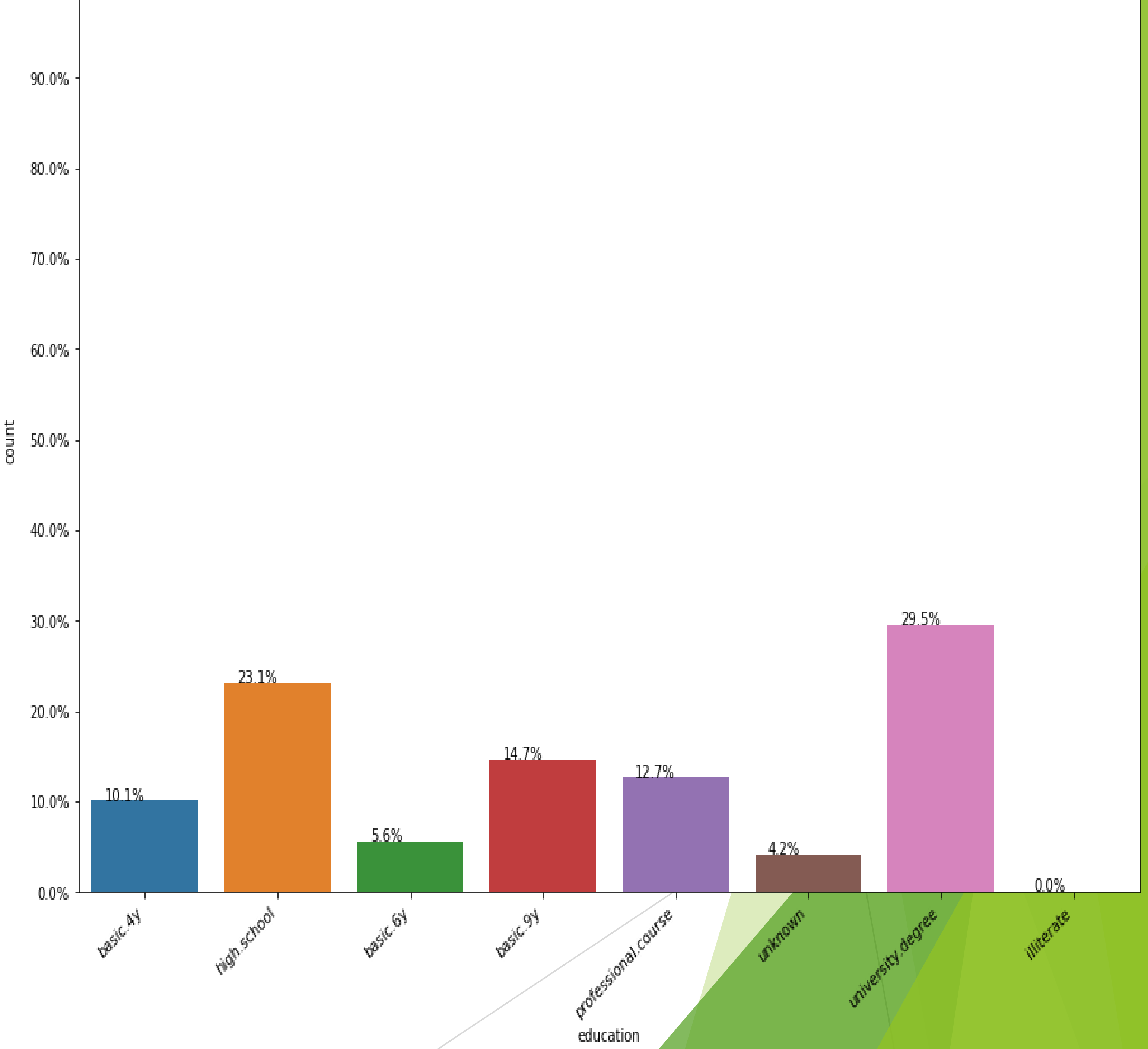
Do people who have or who don’t have credit in default subscribe the most for long term deposits?

- From the plot we can observe that people with default status as no are the most one’s who have and have not subscribed for bank deposits.



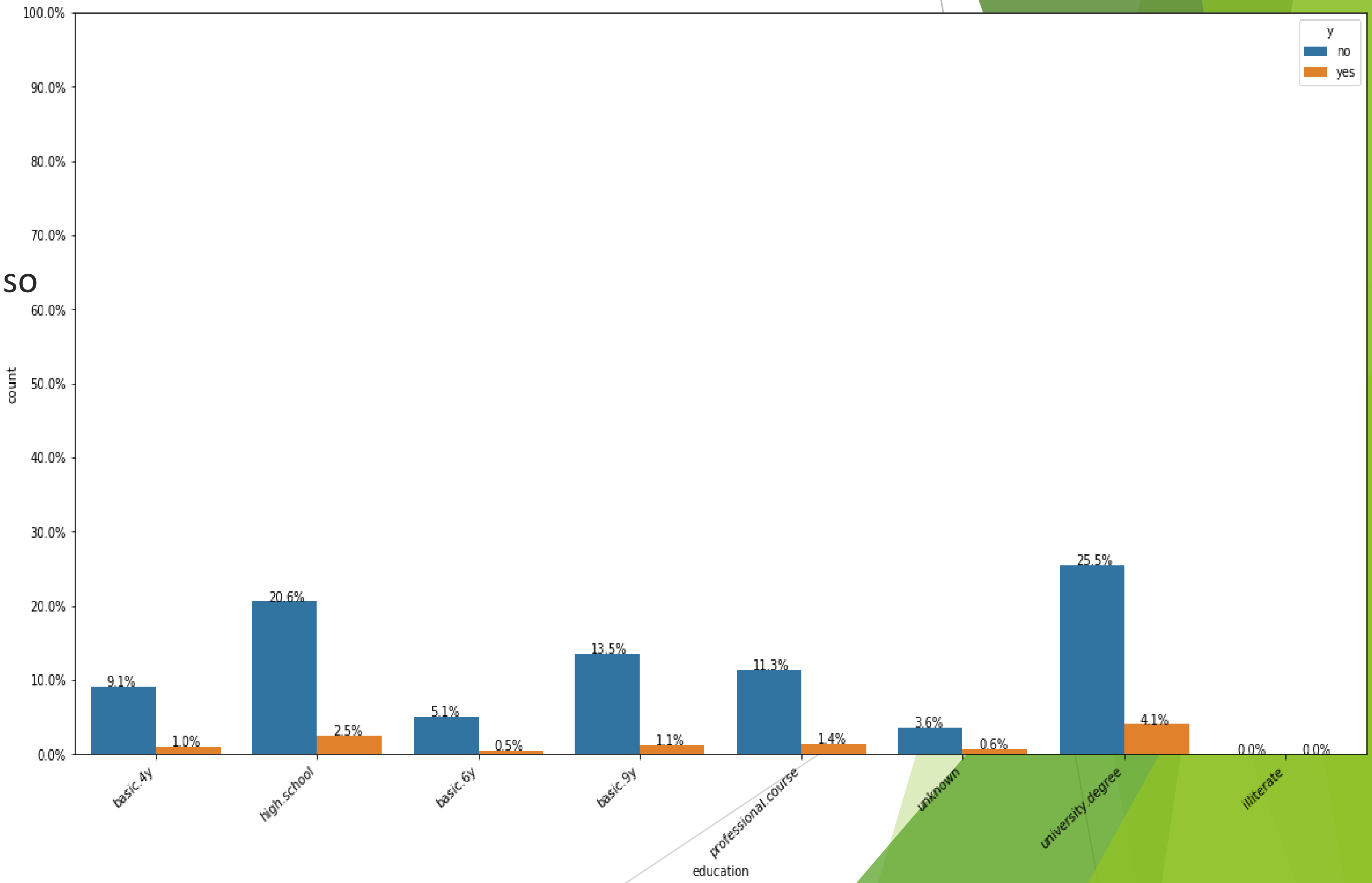
➢ Feature: Education

- As shown from the plot, people contacted by the bank with university degree as their educational qualification are more than the people with any other educational qualification.



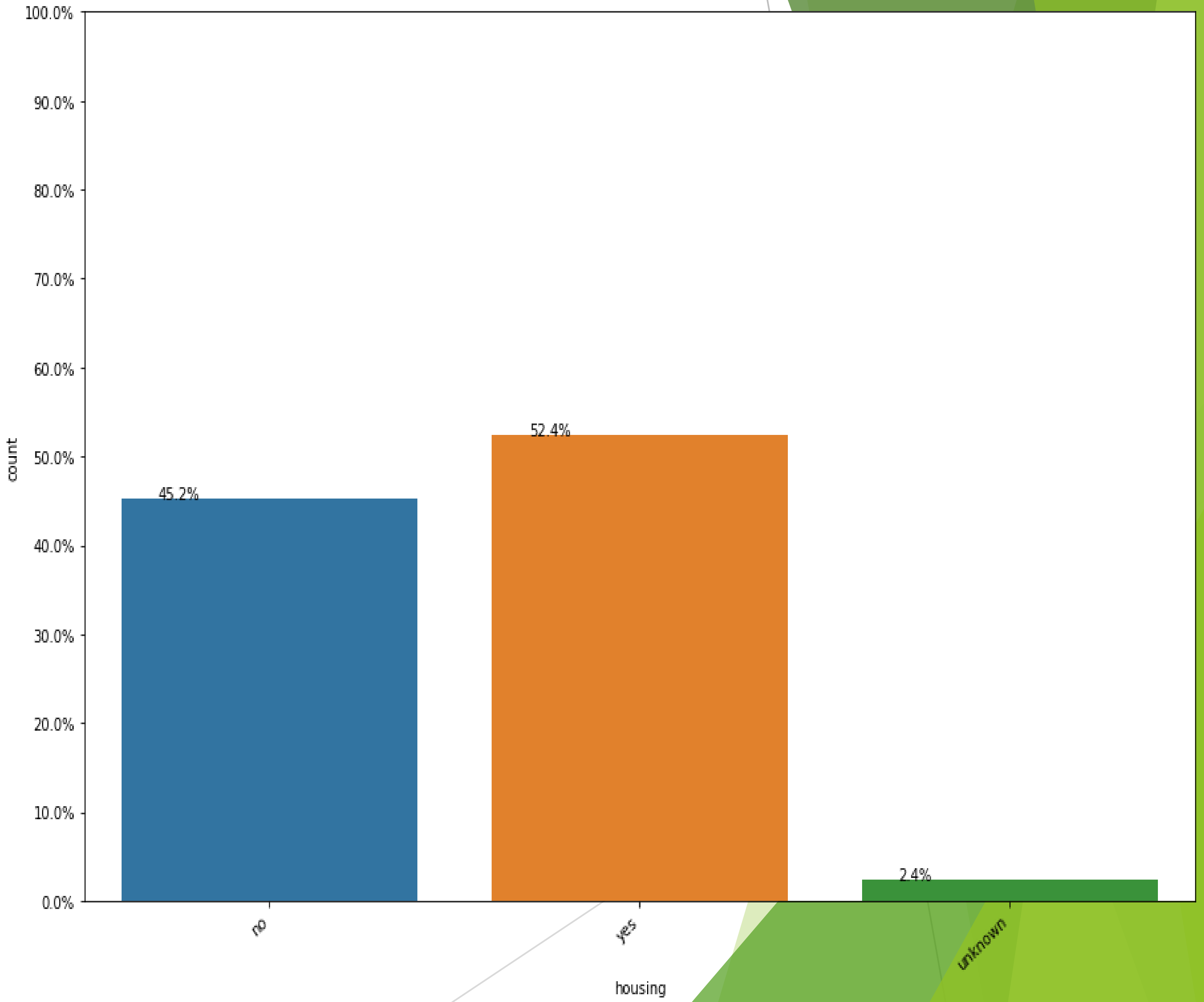
Do people who have completed their university degrees tend to subscribe the most for long term deposits?

- As shown from the plot, people with university degree as education qualification are the most who have subscribed for the deposits. They are also the most who have not subscribed for deposits.



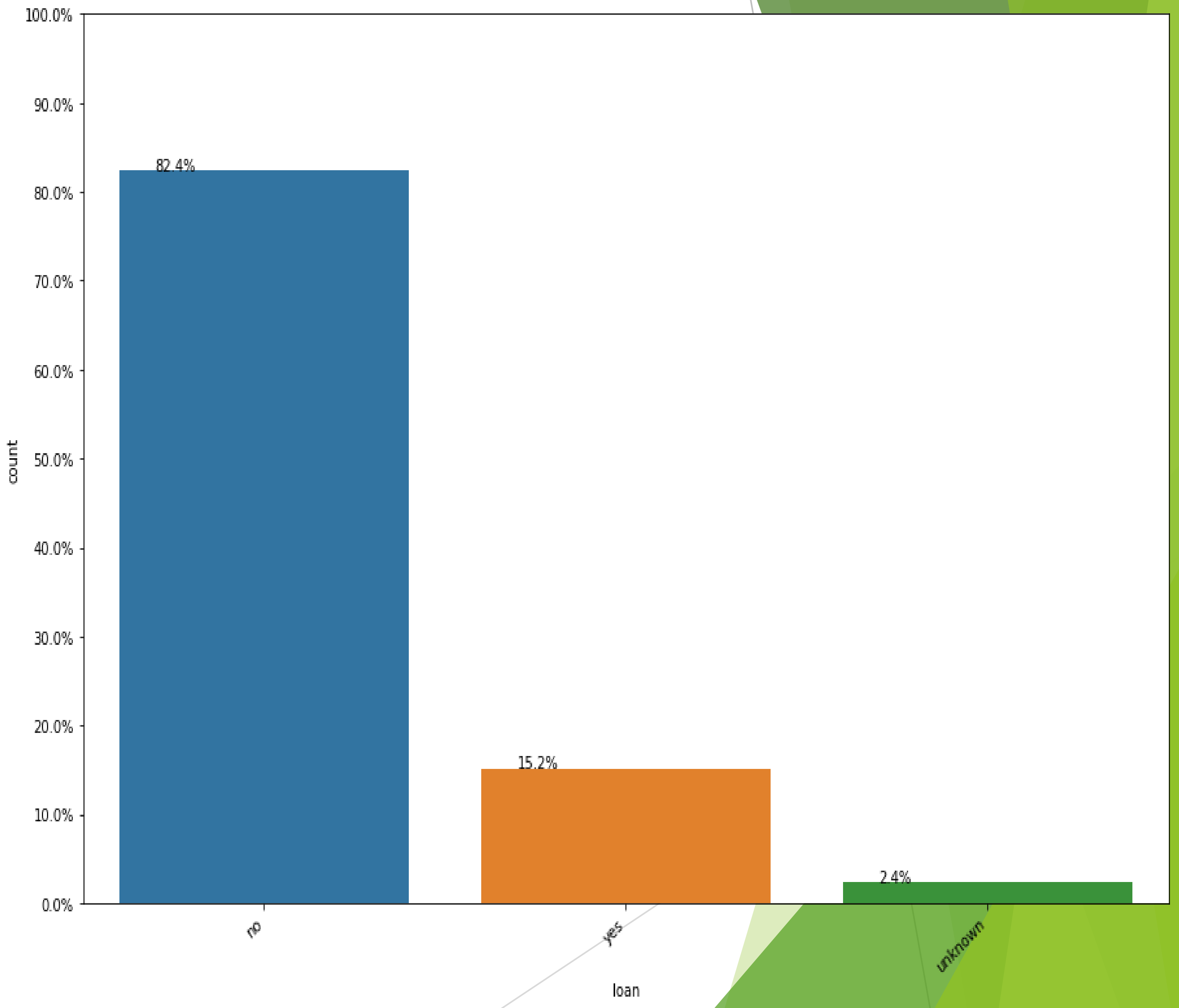
➢ Feature: housing (Categorical)

- People who have housing loan have been contacted the most by the bank.
- People who have no housing loan has also been contacted pretty much, and people who have status unknown have been least contacted.



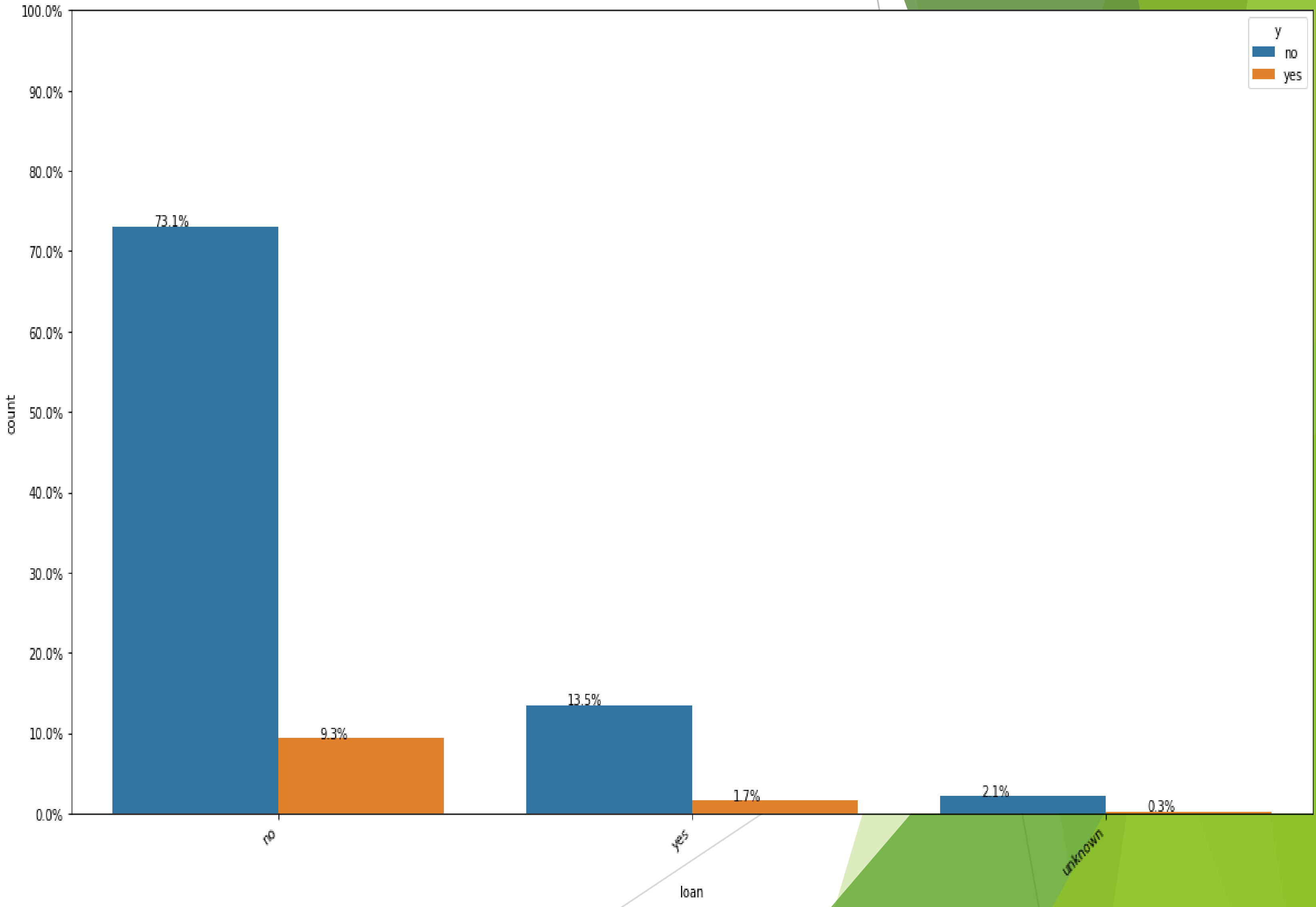
➢ Feature: loan (Categorical)

- As shown from the plot,people who do not have loans have been most contacted for longer term deposits than the ones who have loans.



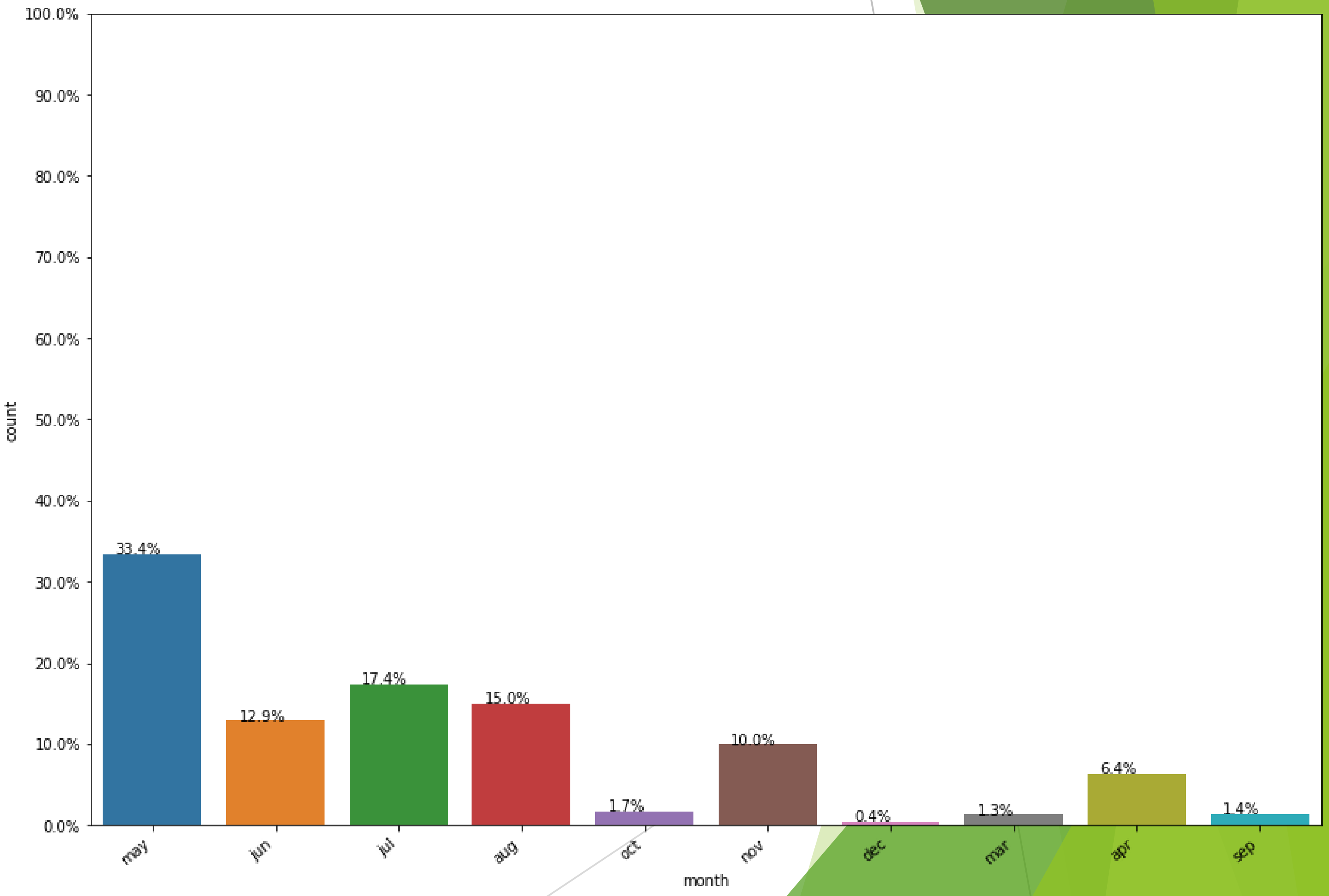
➢ Do people with or without loans subscribed the most for long term deposits?

- People with no personal loan have subscribed the most for long term deposits.
- They are also the ones who have not subscribed for long term deposits.
- People are being contacted the most in the month of May than any other months.
- It is followed by July, August, June.
- Very few people have been contacted in the month of December.
- People have not been contacted in the month of January and February.



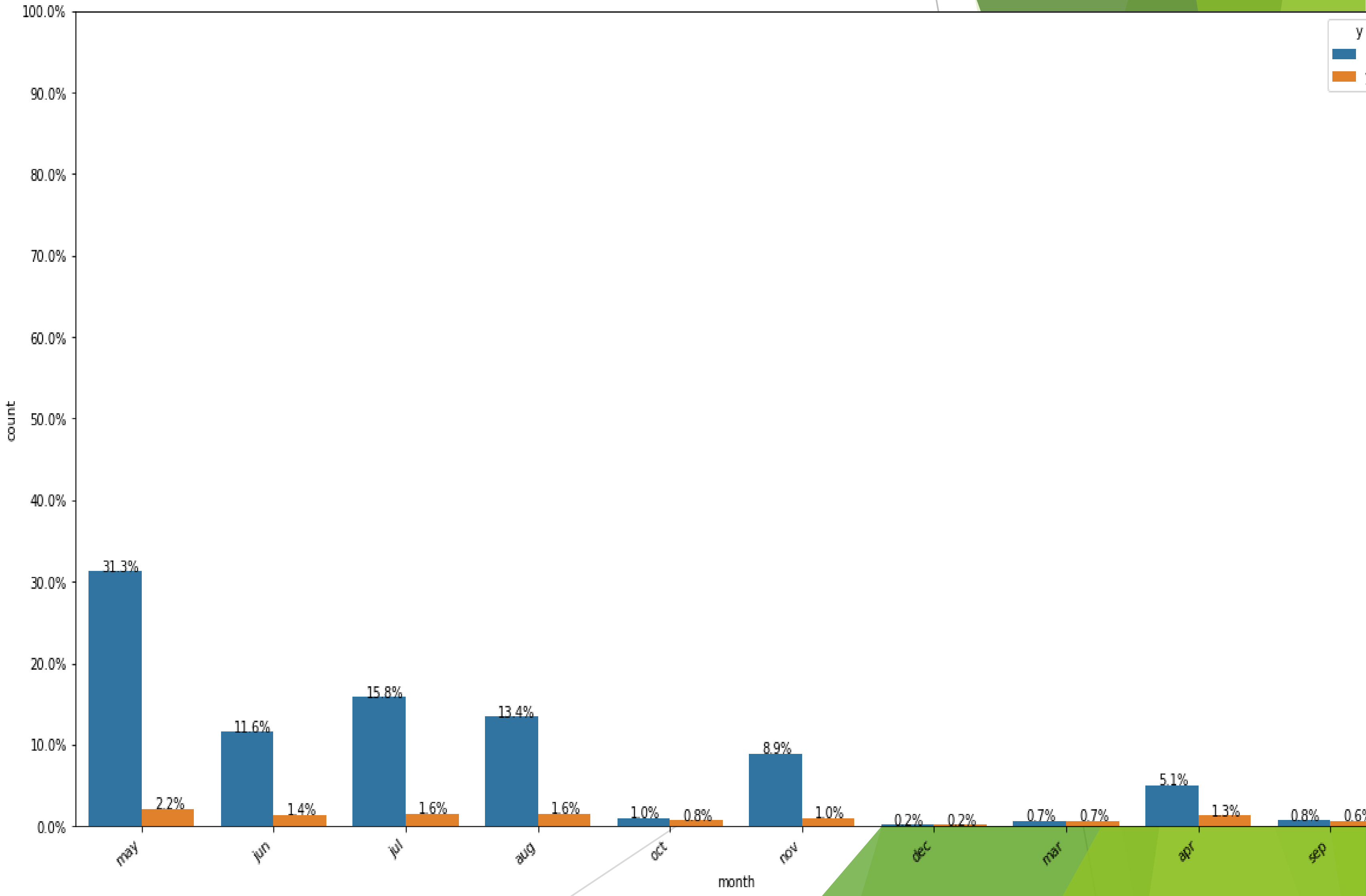
➢ Feature: month (Categorical)

- People are being contacted the most in the month of May than any other months.
- It is followed by July, August, June.
- Very few people have been contacted in the month of December.
- People have not been contacted in the month of January and February.



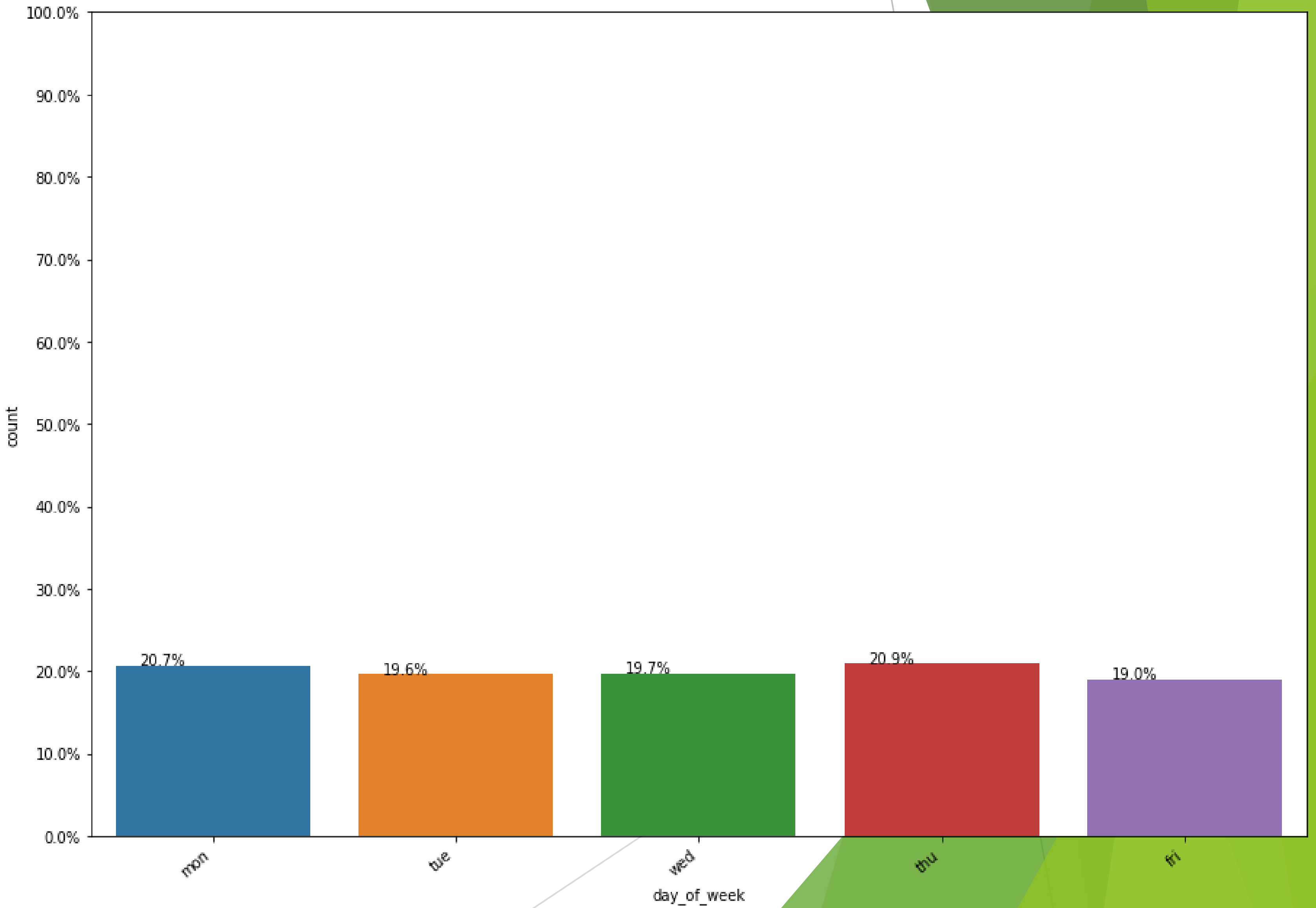
➢ In which month do people have higher chances to subscribe for longer term deposits?

- People who have been contacted in May have higher chances to subscribe for longer term deposits but have also higher chances for not subscribing the long-term deposits.
- Very few people are contacted in the month of December, March, September, and October and have almost equal chances for subscribing the deposits or not.



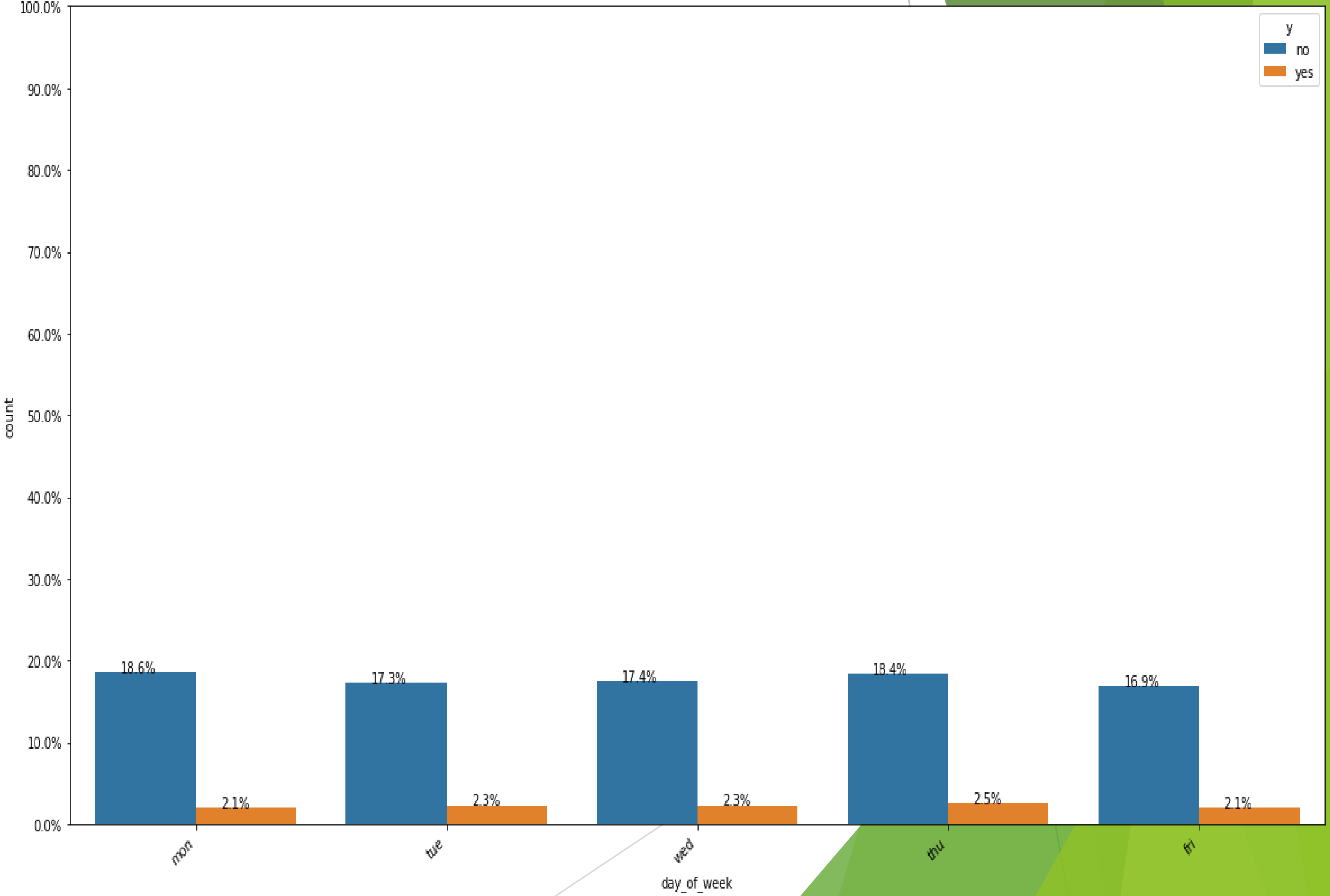
➢ Feature: day_of_week (Categorical)

- From the plot we can see that people are contacted from Monday to Friday but not on Saturday and Sunday.



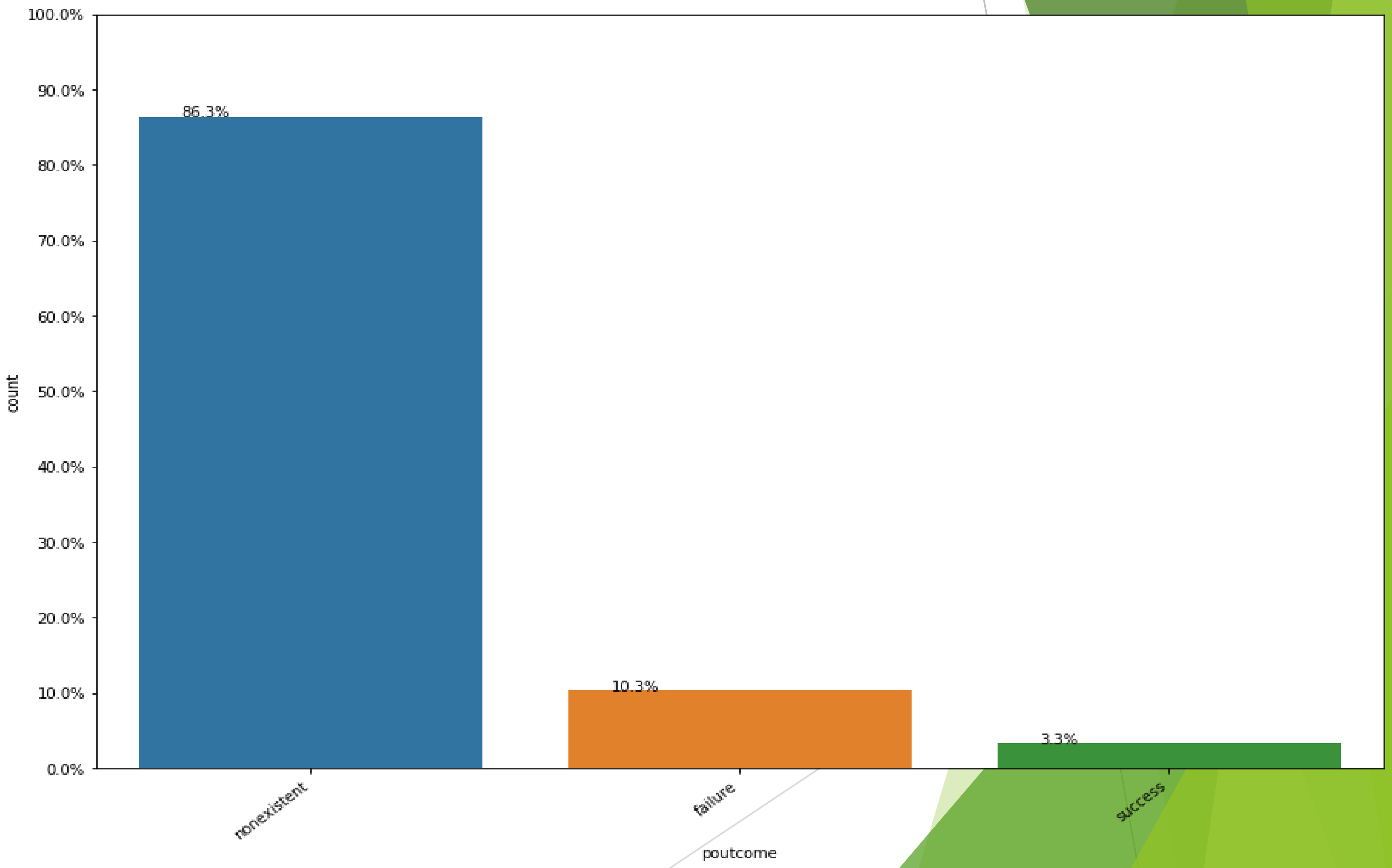
➢ Does in all days people have equal chances for subscribing or not subscribing for the long-term deposits?

- As shown from the plot, in all the days they have equal chances for subscribing and not subscribing the term deposits.
- Day_of_week may not be very helpful in predicting whether the customer will subscribe for long term deposits or not.



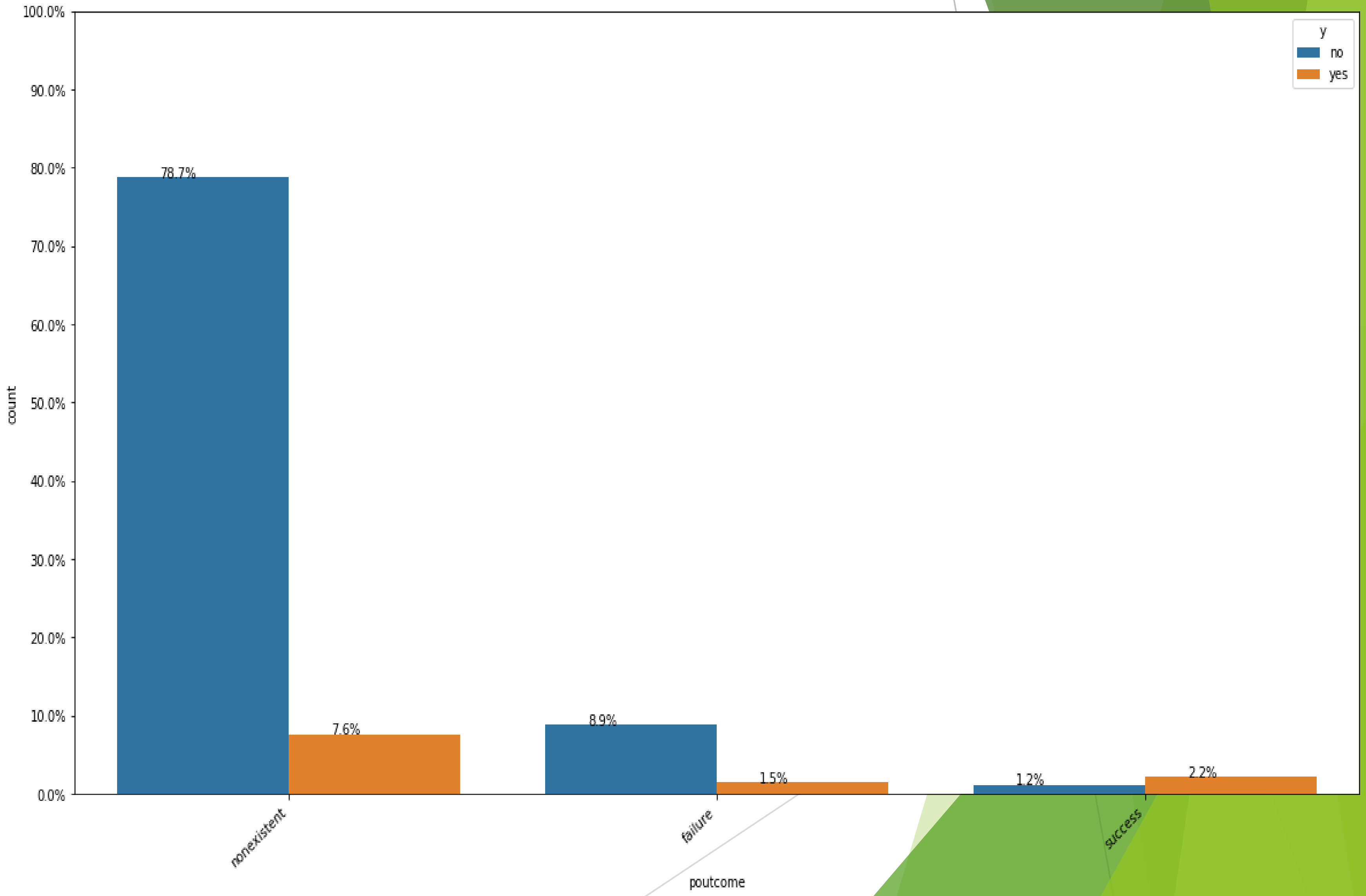
➢ Feature: poutcome (Categorical)

- As shown from the plot it is evident that majority of outcome of previous campaigns are nonexistent.
- Very few people from previous marketing campaigns have subscribed for long term deposits.



Which previous outcome has subscribed the most?

- As we can see, people whose previous outcome is non-existent has subscribed the most than any other group of people belonging to previous outcome.
- It is also clear that people belonging to success category of previous outcome has turned down for longer deposits.

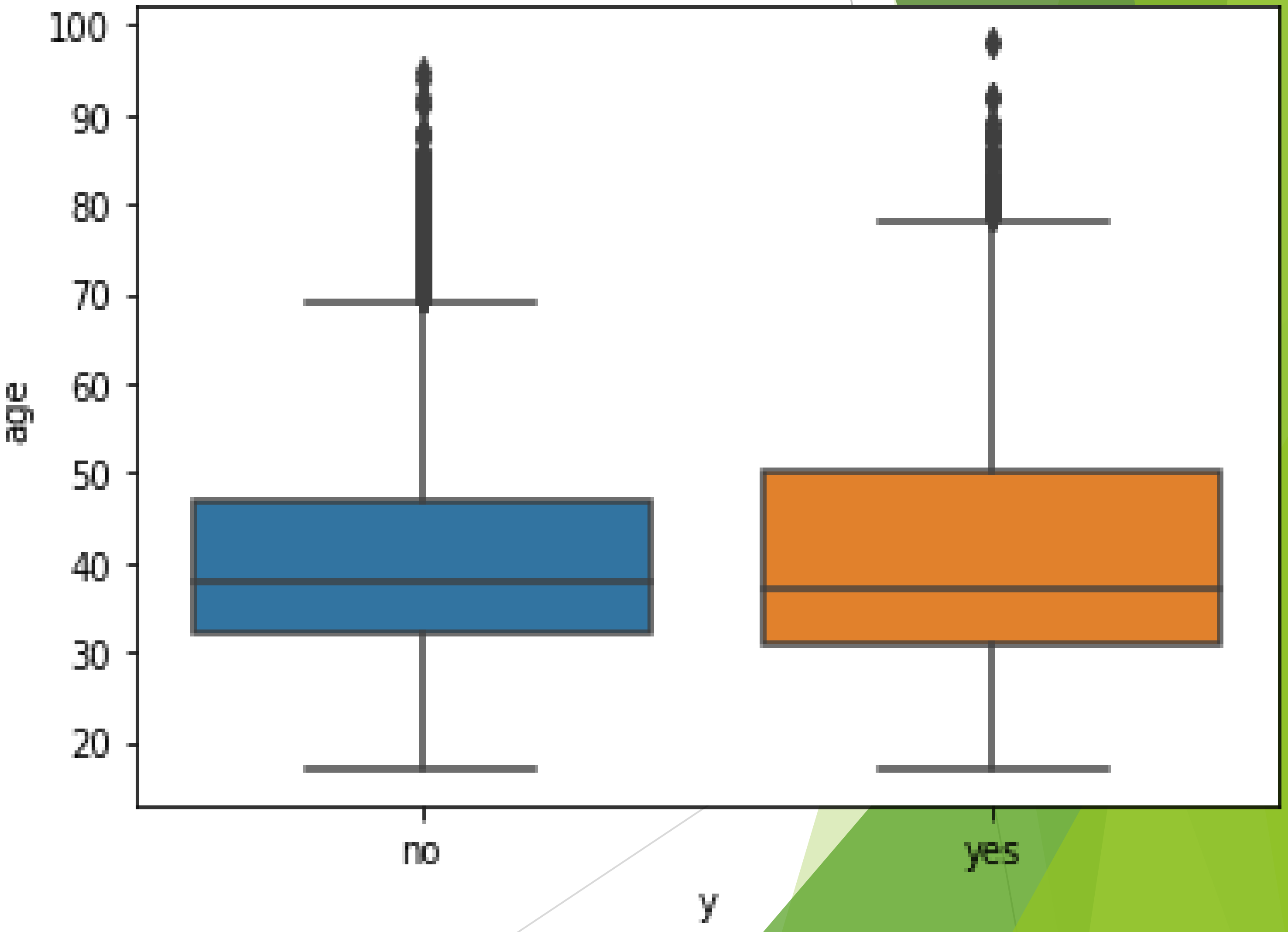


EDA
Numerical Variables:

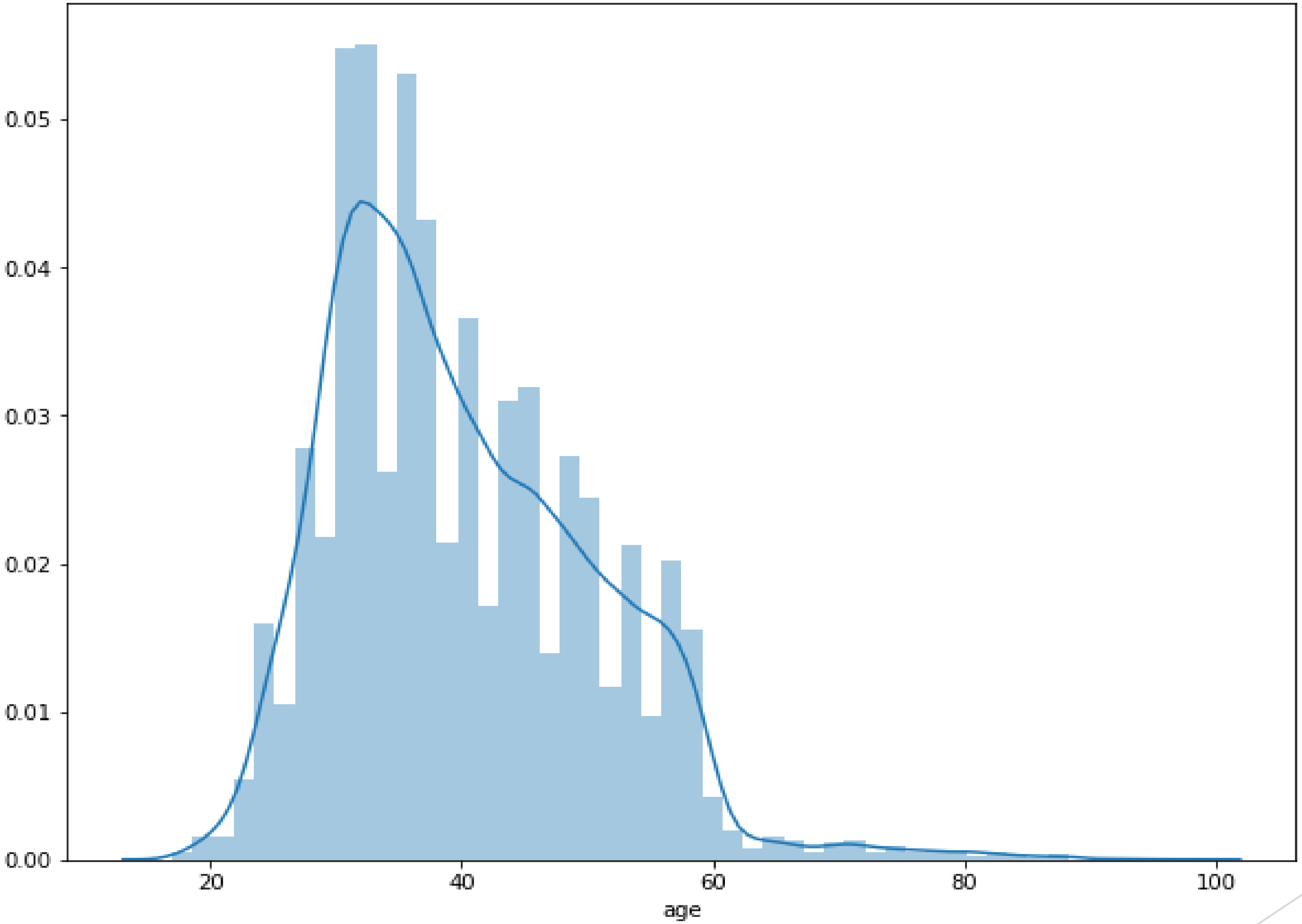
- Now will I perform some exploratory data analysis on the numerical variables.

➢ Feature: age (Numeric)

- From the plot as shown ,we know that for both the customers that subscribed or didn't subscribe a term deposit, has a median age of around 38-40, and the boxplot for both the classes overlaps quite a lot, which means that age isn't necessarily a good indicator for which customer will subscribe and which customer will not.

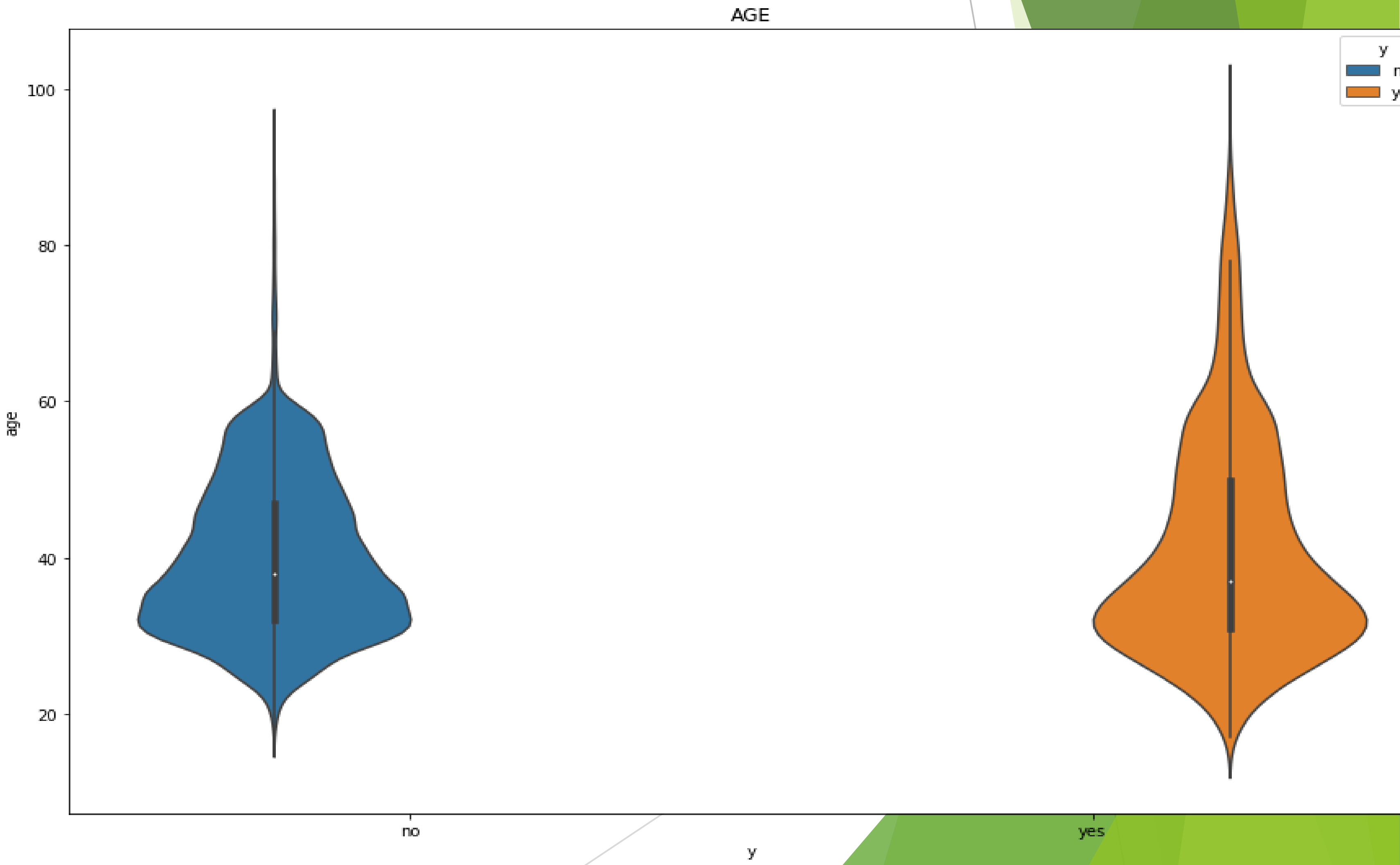


- From the plot as shown below ,it is a right skewed graph and there is an evidence of outliers after the age of 60.
  We also see in the distribution that most of the customers are in the age range of 30-40.

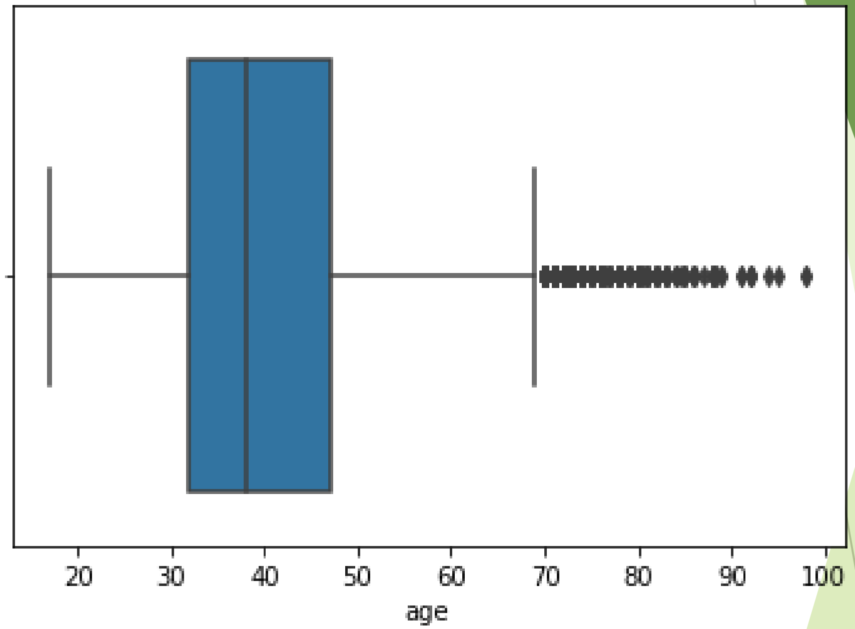


However, we can use other plots to check for outliers ,Violin plot and box plot.

- Using the Violin plot as shown , it is clearly visible that there are outliers present for both the class.
- In No class, outliers are present above age 70 and for Yes class, outliers are present above age 75.
- Median for No class is around 40 which is same for Yes class.
- Also, it is visible that IQR range is almost overlapping so age might not be very helpful in predicting class label.

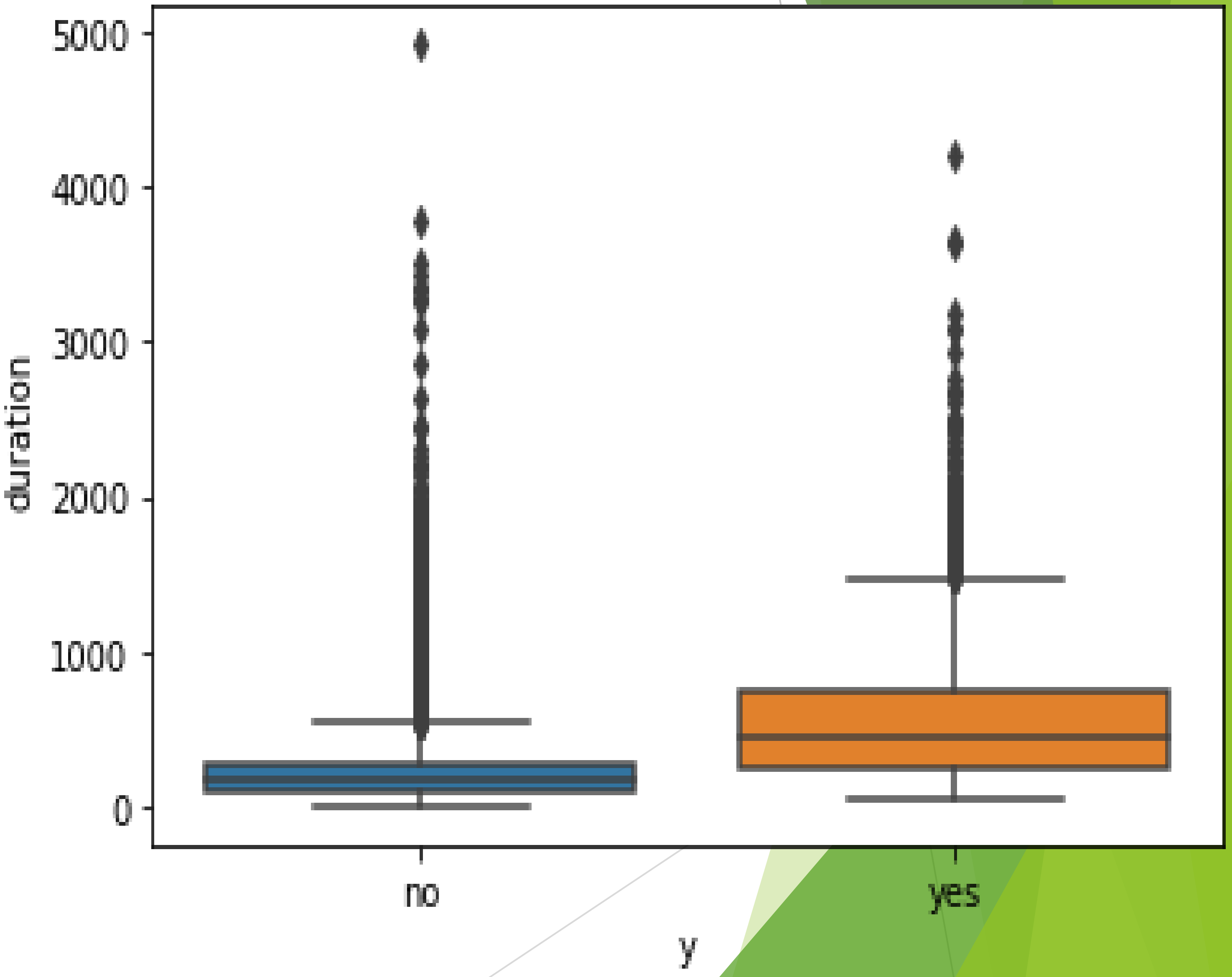


- Using the box plot as shown , we can say that outliers are present after the age of 70 years.

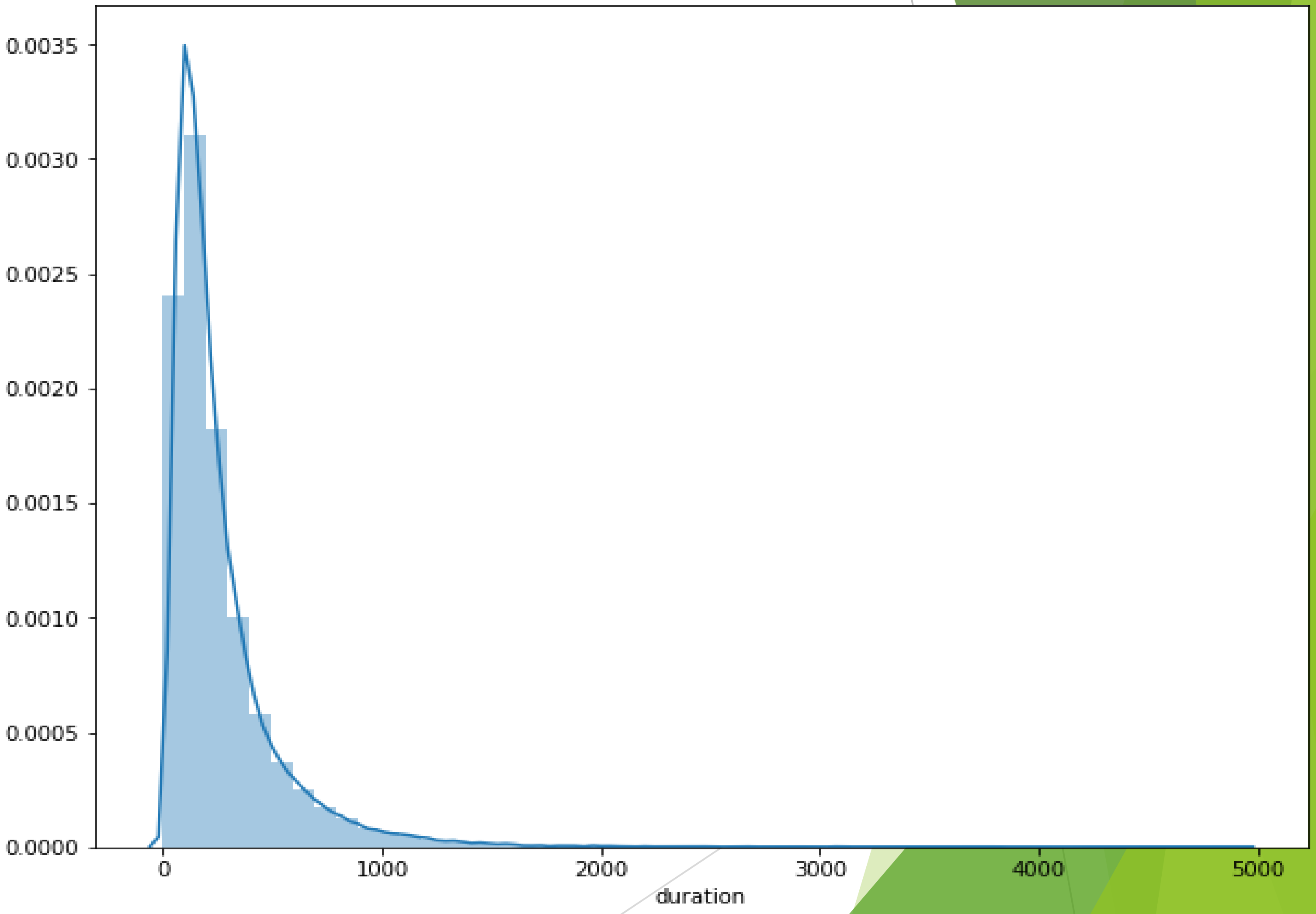


➢ Feature: duration (numeric)

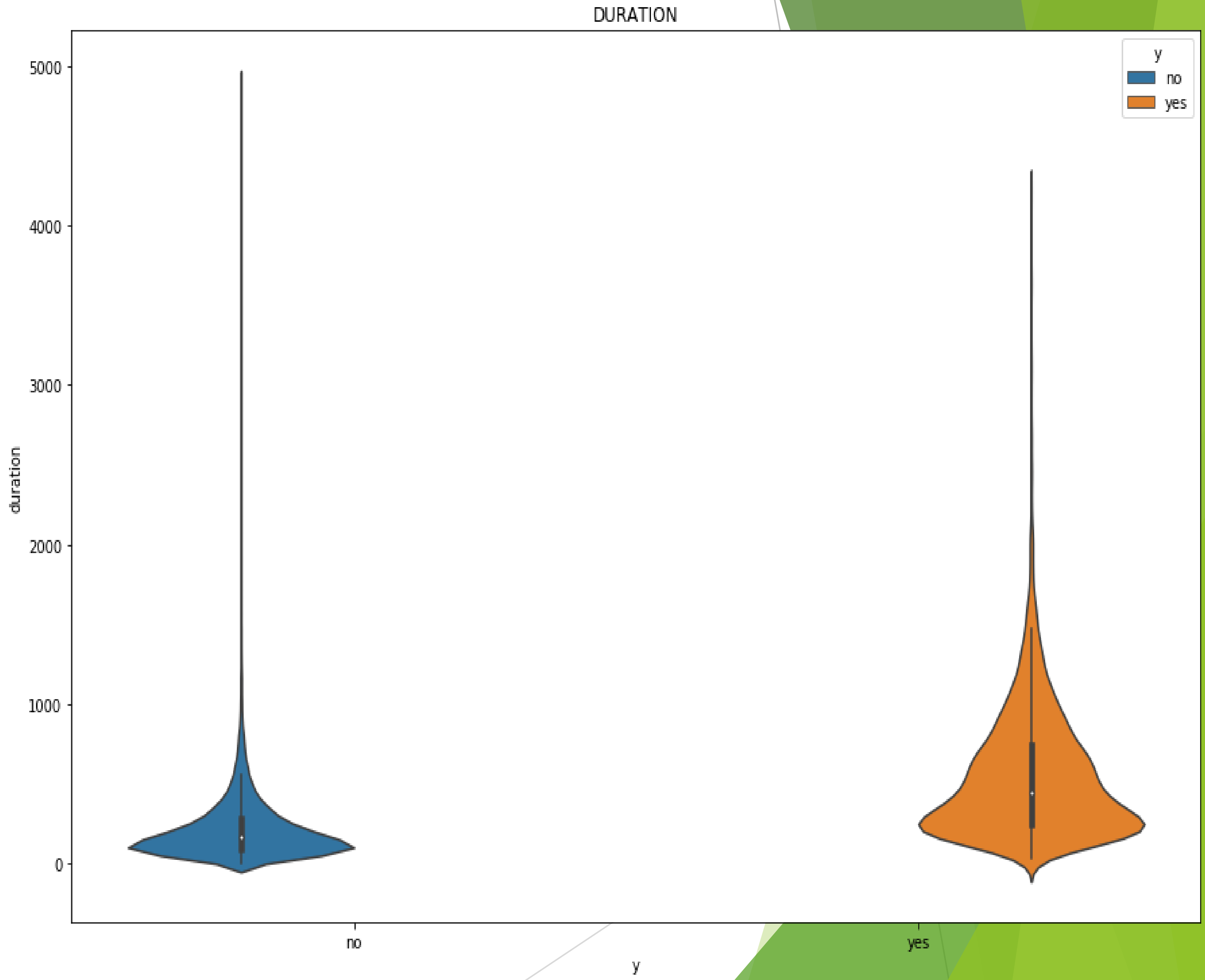
- From the plot as shown, the duration (last contact duration) of a customer can be useful for predicting the target variable.
- It is expected because it is already mentioned in the data overview that this field highly affects the target variable and should only be used for benchmark purposes.



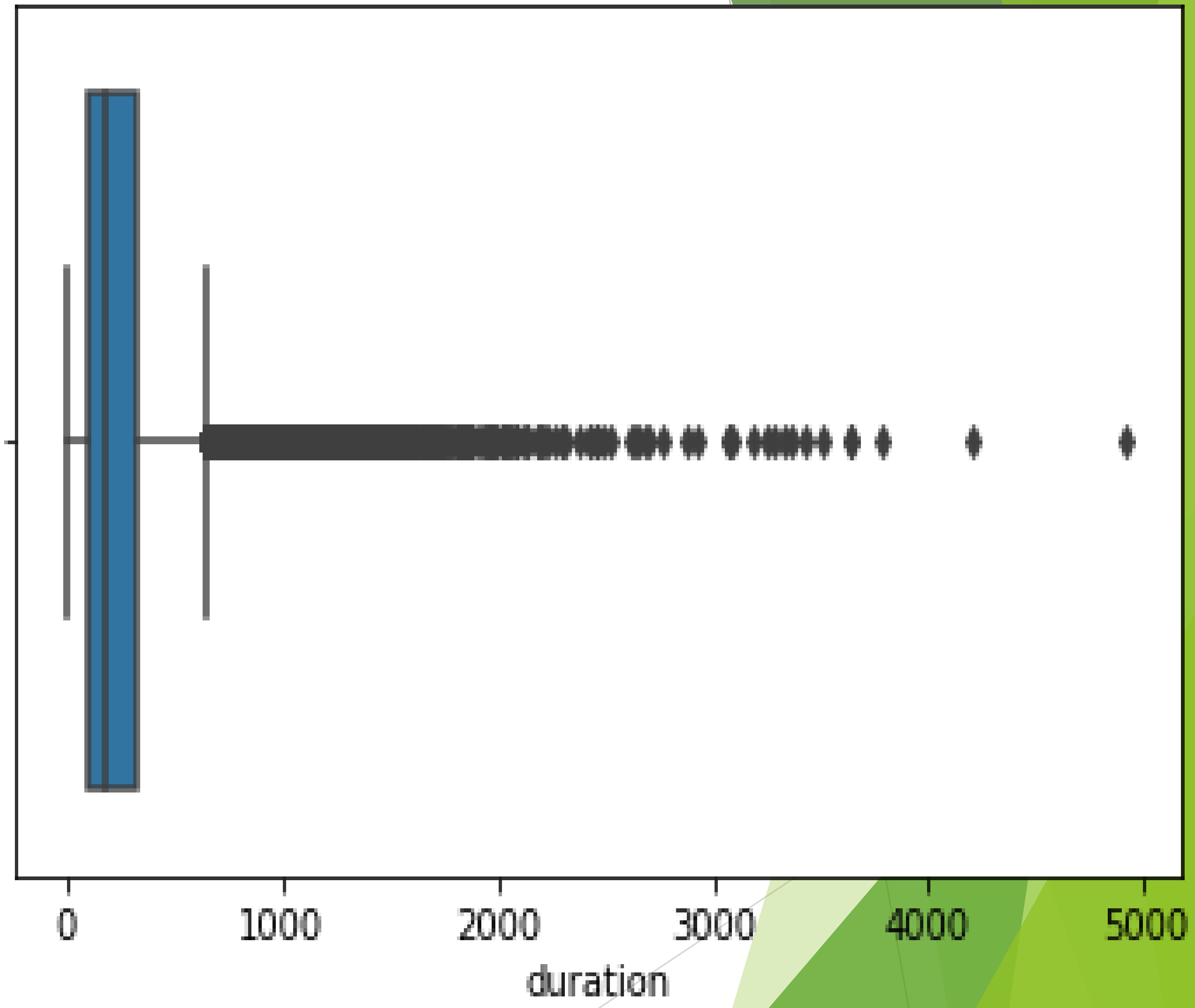
- As shown from the plot ,it is clear that from duration around 1500 onward outliers, are present and it is right skewed data.
- In this distribution above we find where most the values are very low and very few have high values.



- Using the Violin plot as shown , any duration of call with class labels as no, more than 1000 are considered as outliers while with class labels as yes, more than 1500 would be considered as outliers.

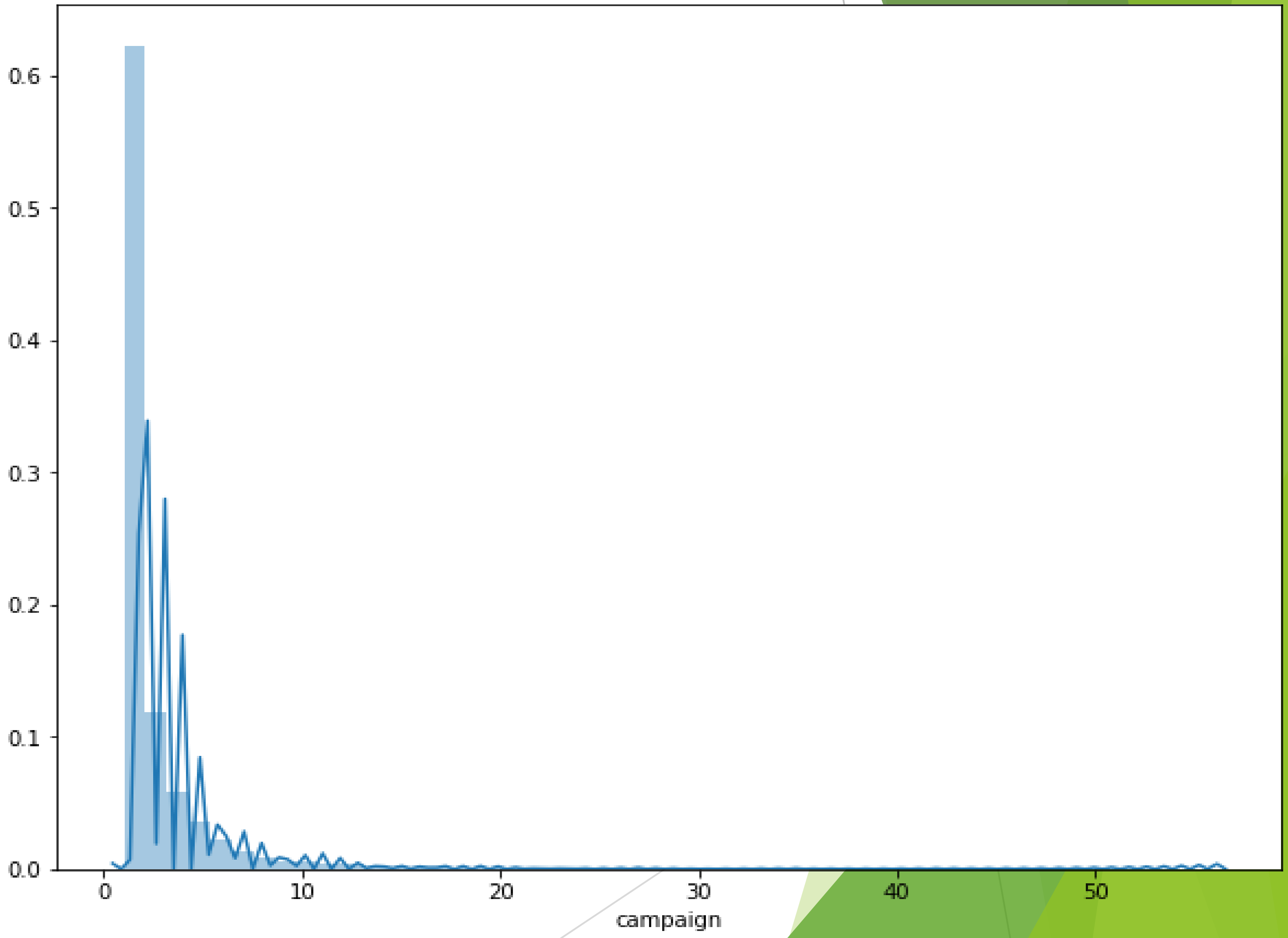


- Using the box plot as shown , we can say that outliers are present after the duration of calls more than 1000.

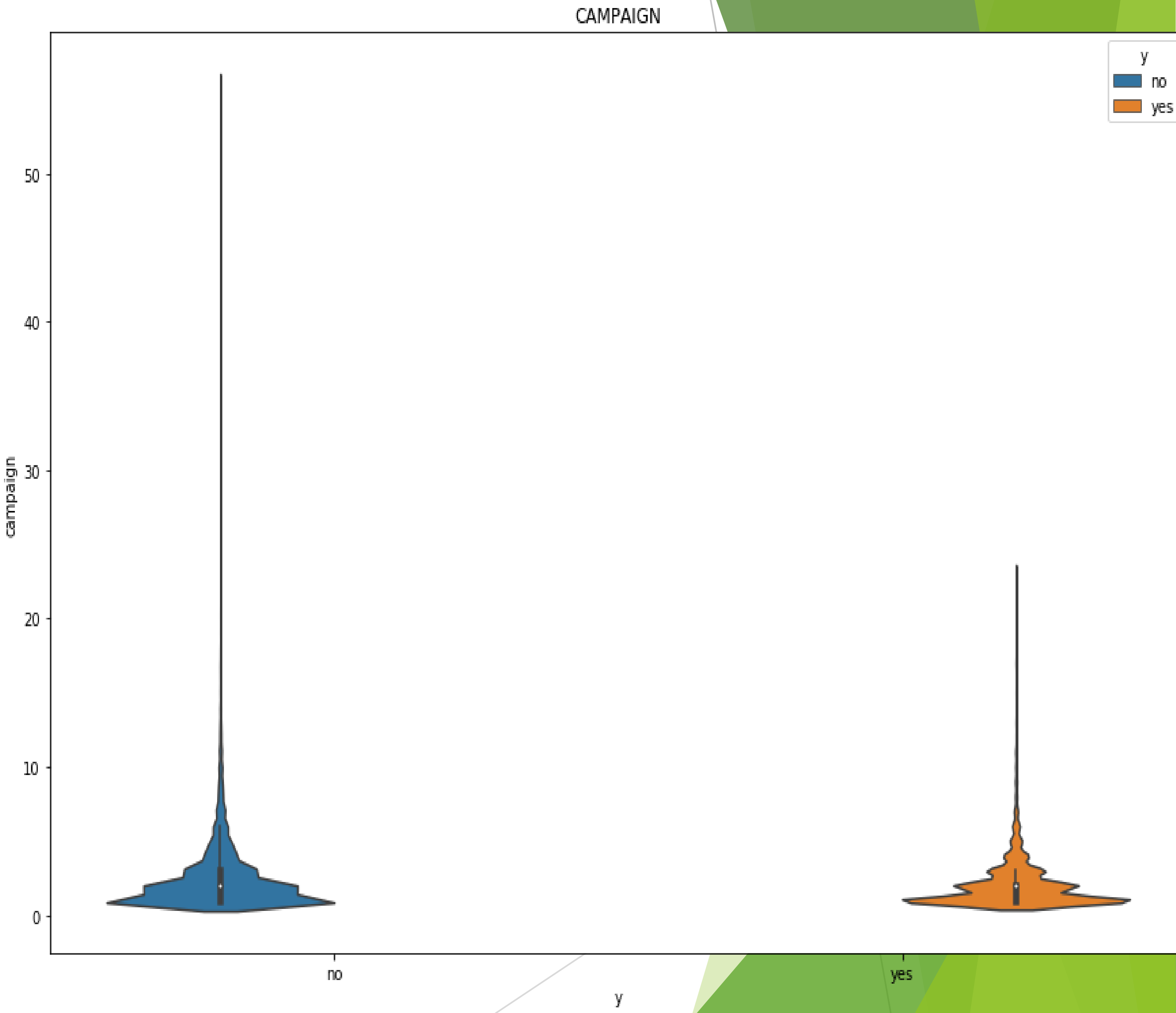


➢ Feature: campaign (numeric)

- As we can see from the plot, outliers might be present after the number of campaigns 10.

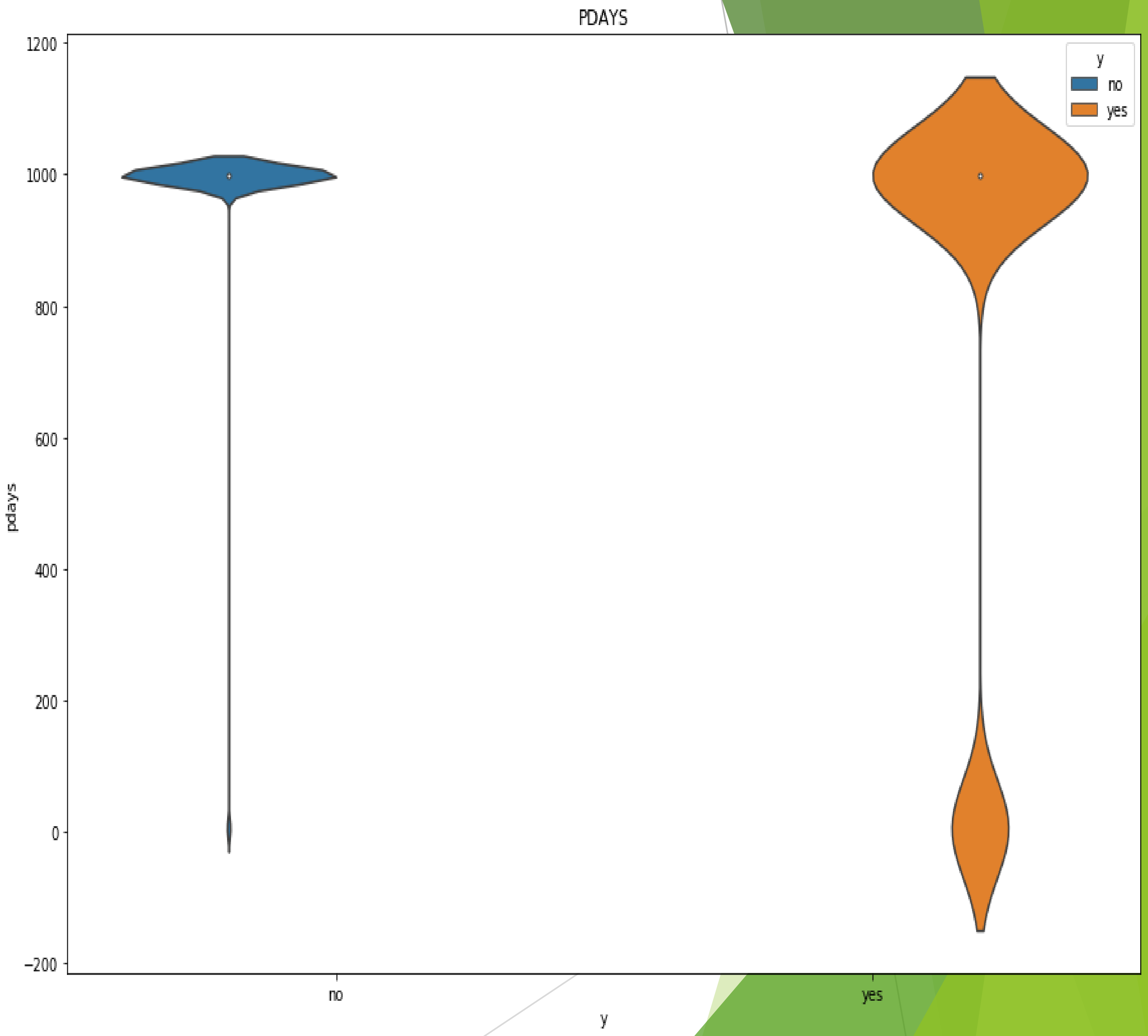


- Outliers are present when number of campaigns are more than 10 irrespective of any class labels.

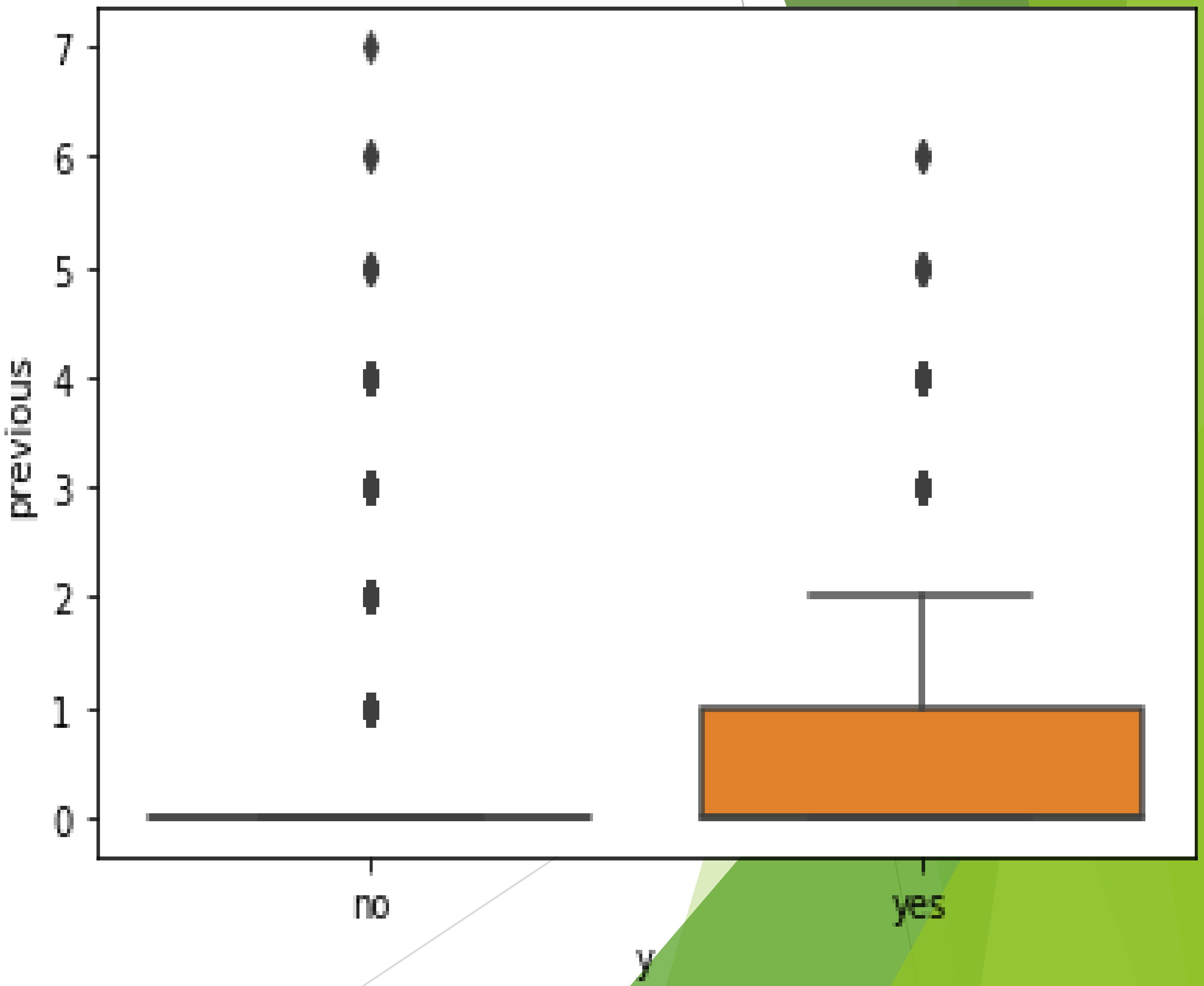


➢ Feature: pdays (numeric)

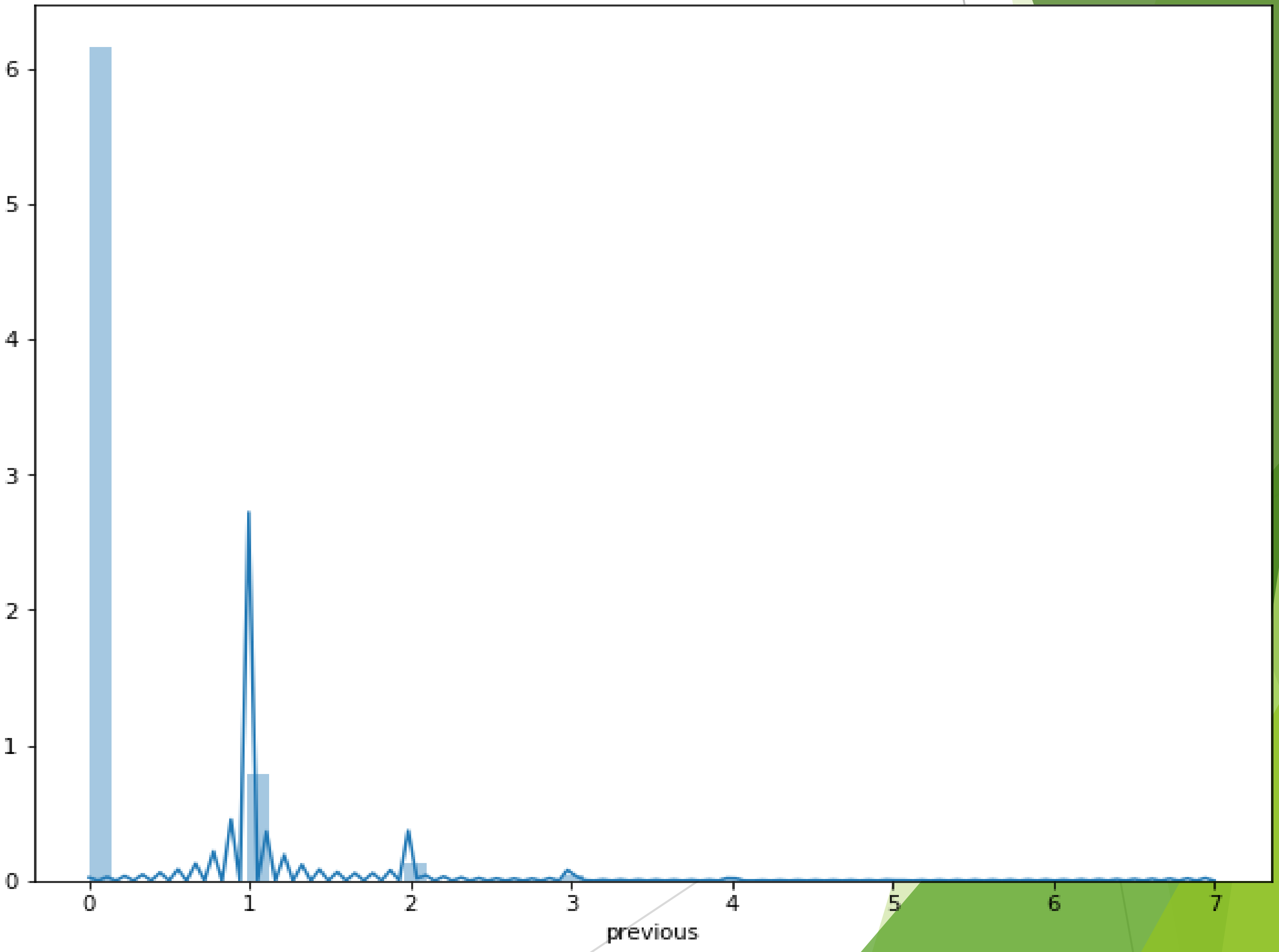
- From the Violin plot it is visible that irrespective of class labels, mostly people have not been contacted by the bank. Very people have been contacted by the bank and number of days passed for previous campaign is between 0–100.
- It means we either must compute pdays or drop the pdays depends on the percentage of values.
- Also, it is not very clear but the IQR range for the both the classes are overlapping.



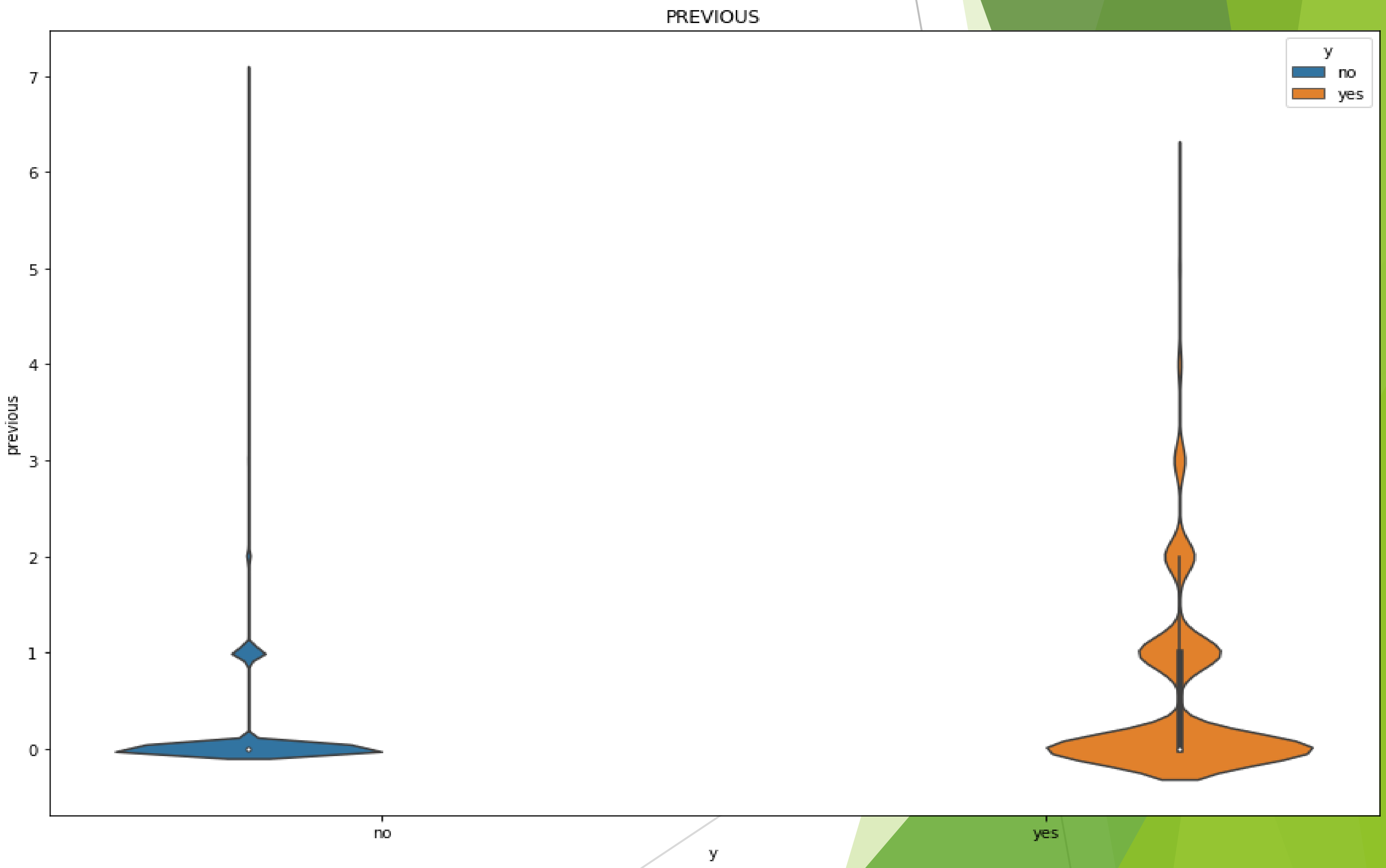
- **Feature: previous (numeric)**



- From the plot as shown , there might be outliers present after the number 4.

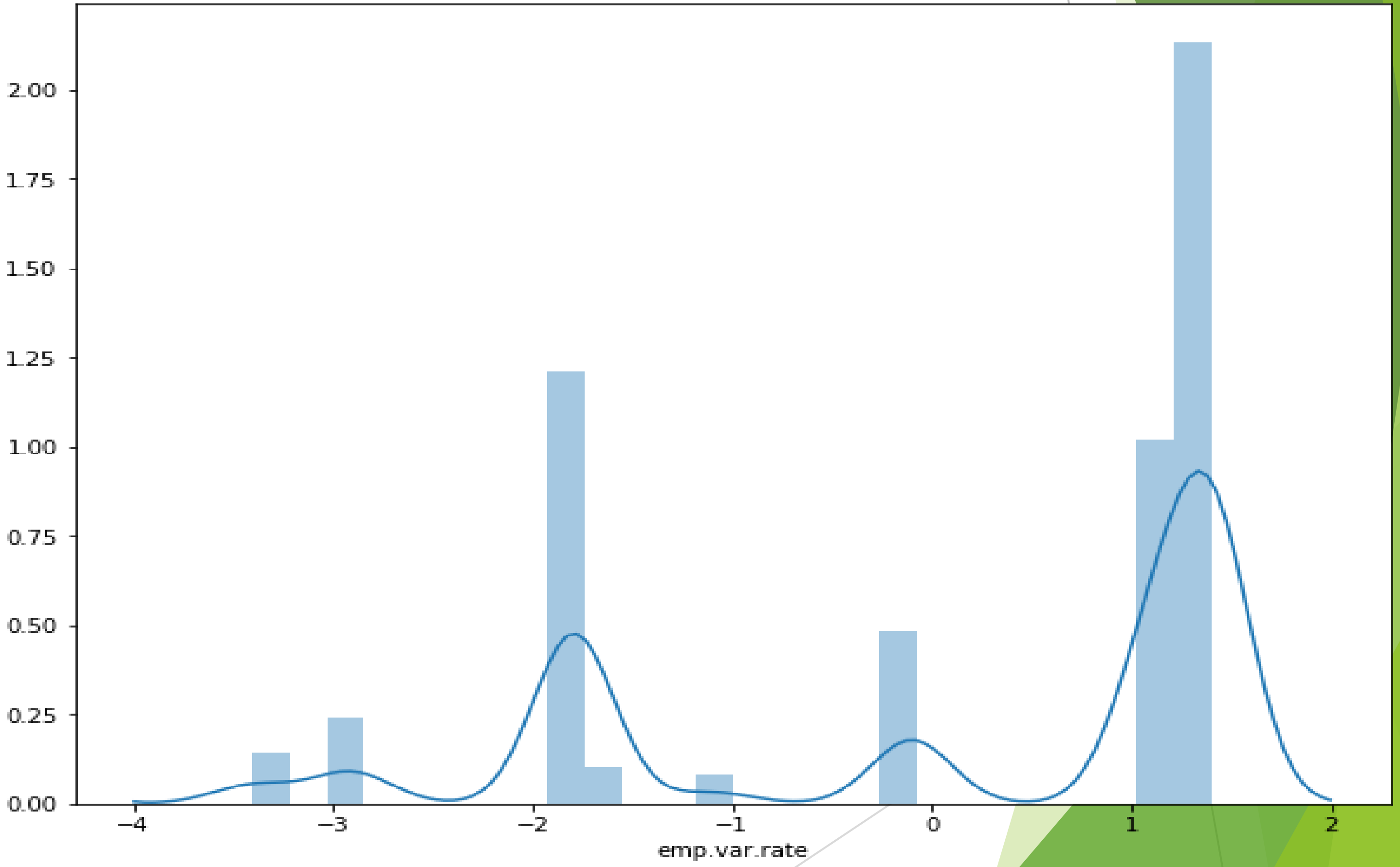


- Using the Violin plot as shown as we can see, people who have contacted once for the previous campaign has subscribed for long term deposits.
- For class no, there are so many outliers starting with value 1 but for yes class, outliers are present from value 3.
- From the plot it is visible that previous feature would be helpful in predicting the class labels.

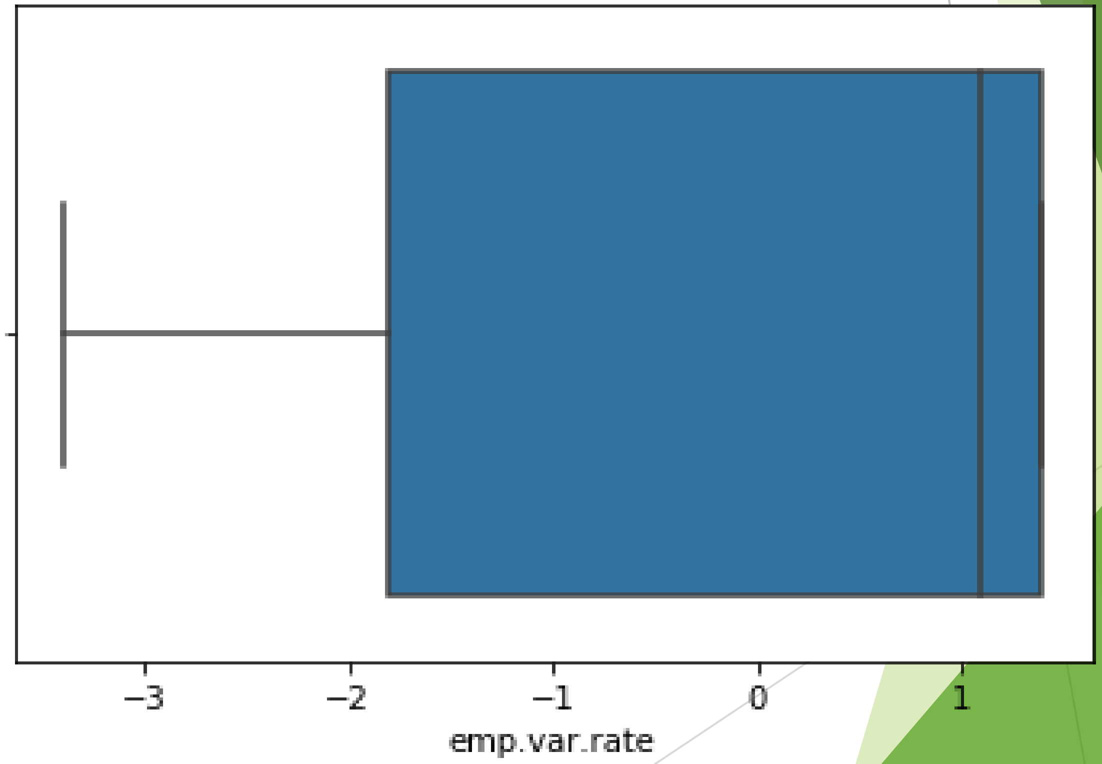


- **Feature:emp.Var.Rate(numeric)**

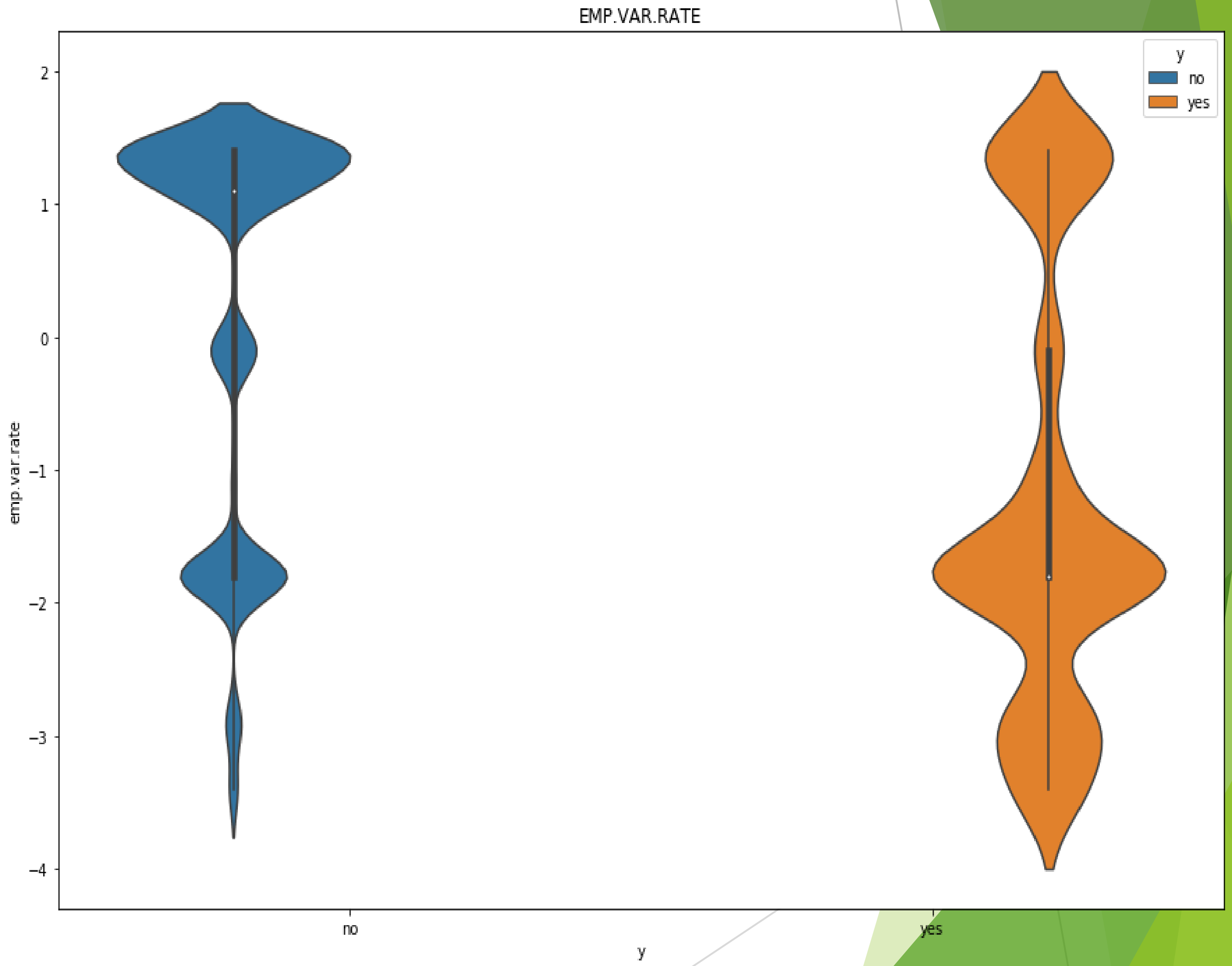
- From the plot it is visible that there are no outliers present.



- No outliers are present as shown on the boxplot.

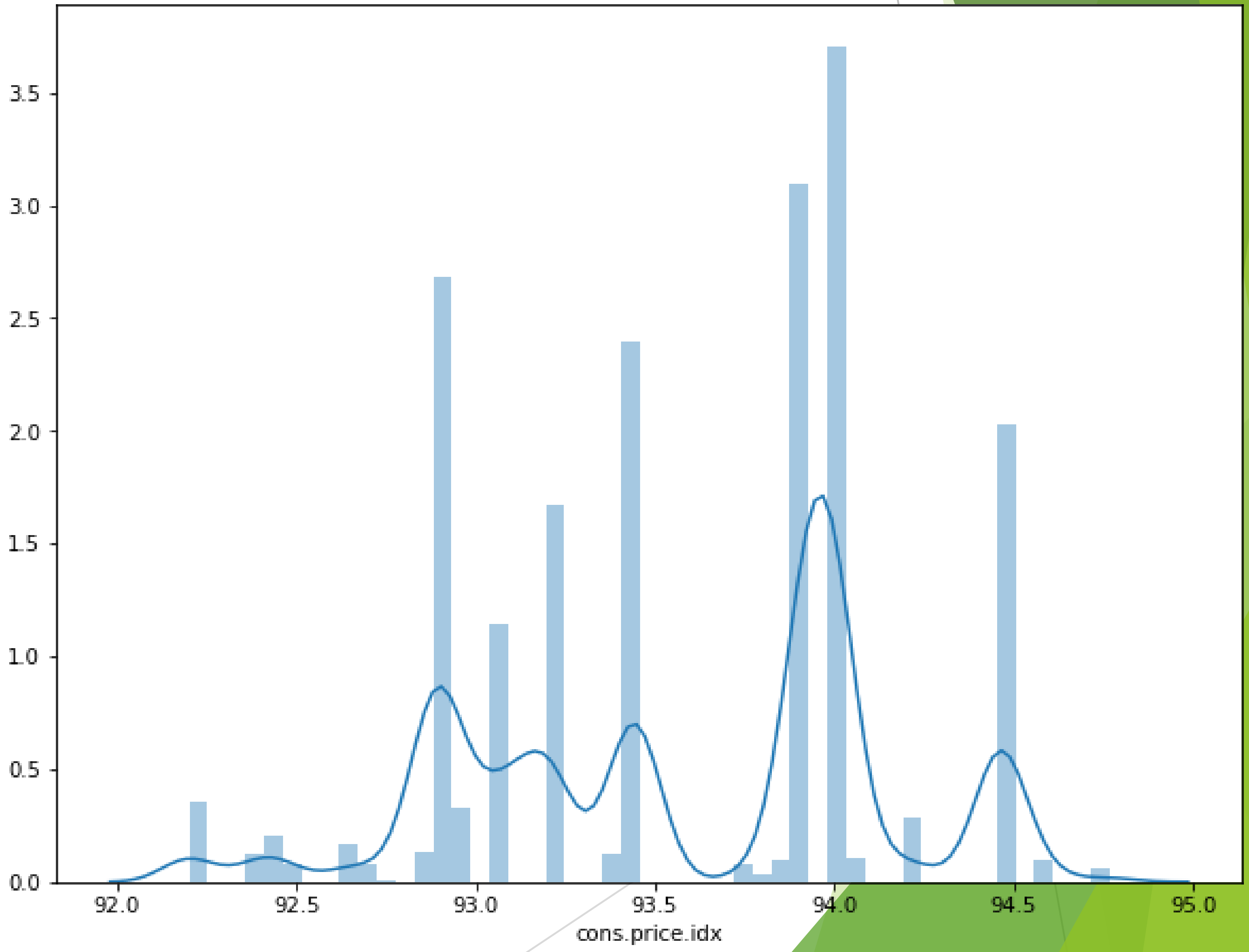


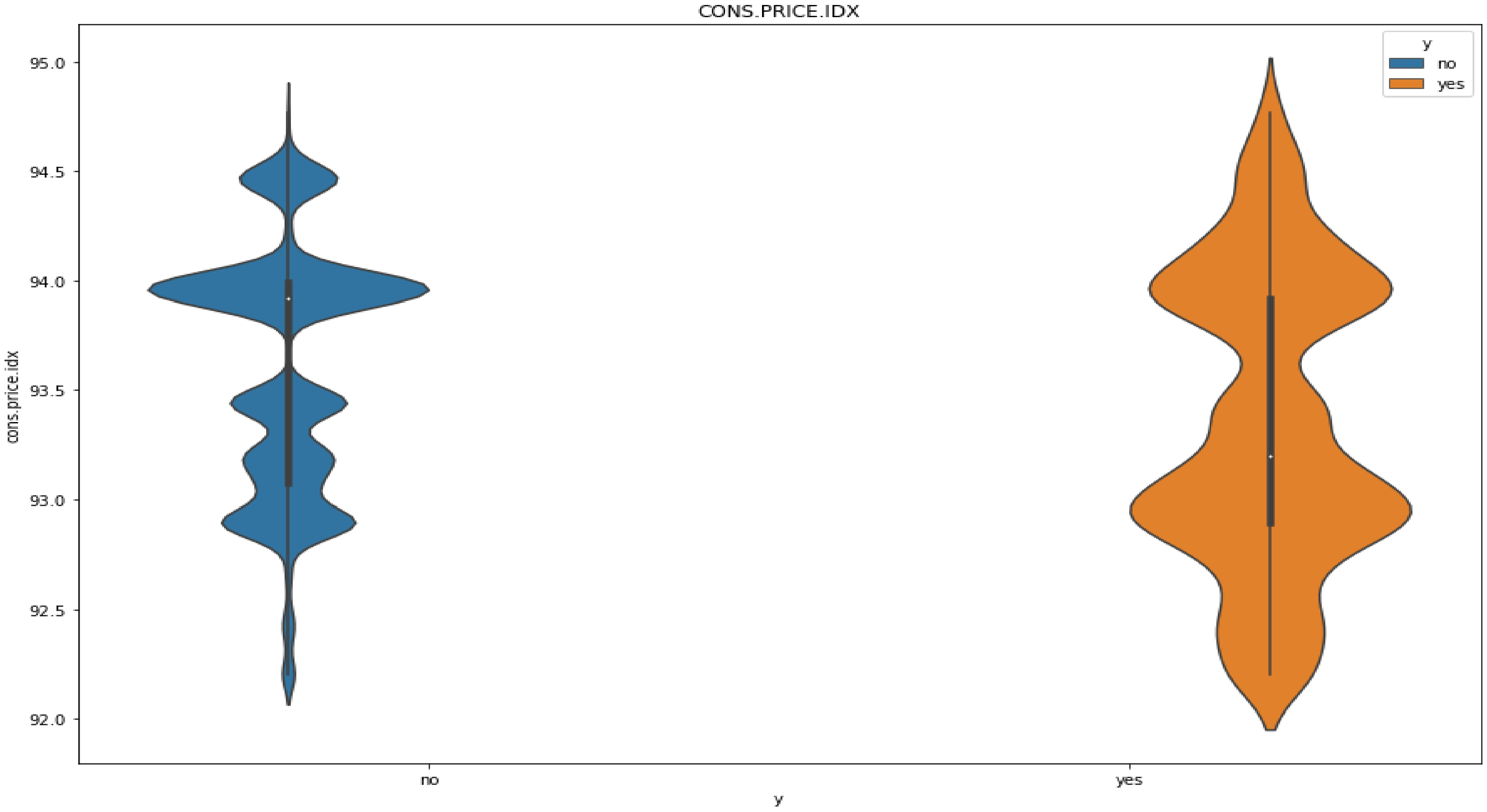
- There are no outliers present for any class for this feature and emp.var. rate feature would be very useful in predicting labels.

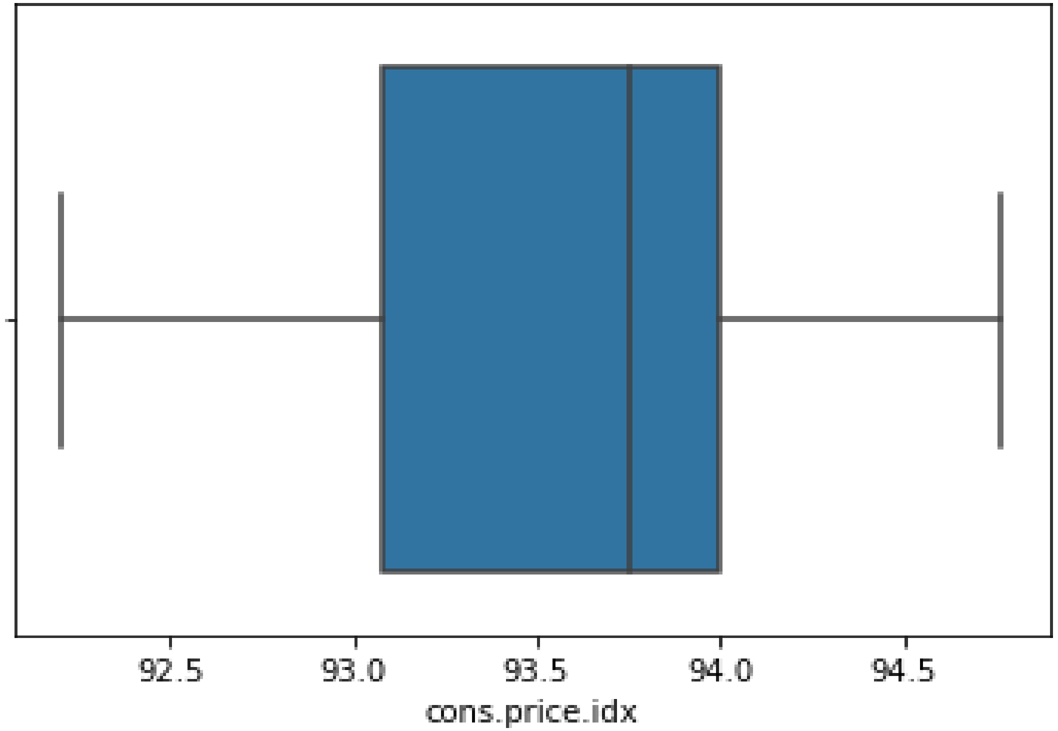


Feature:cons.price.idx(numeric)

- From the plot we can see that there are no outliers present.

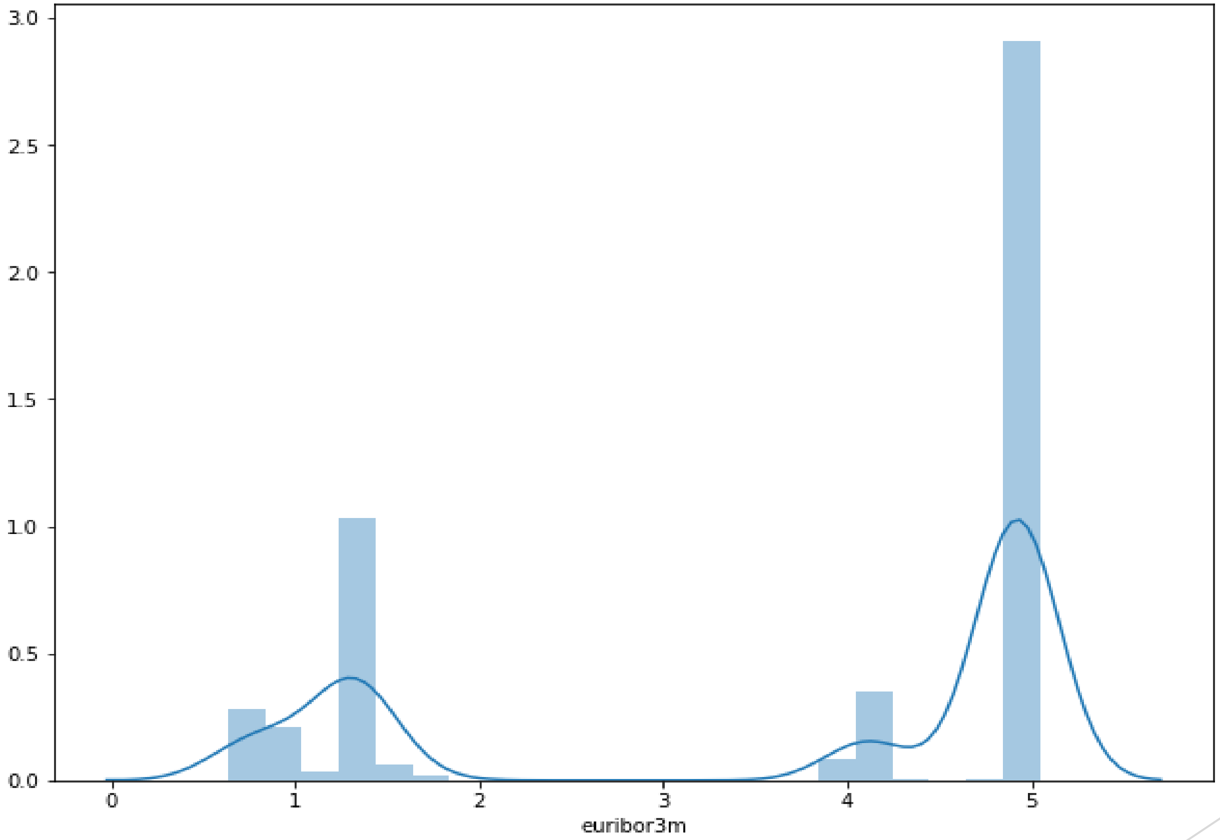


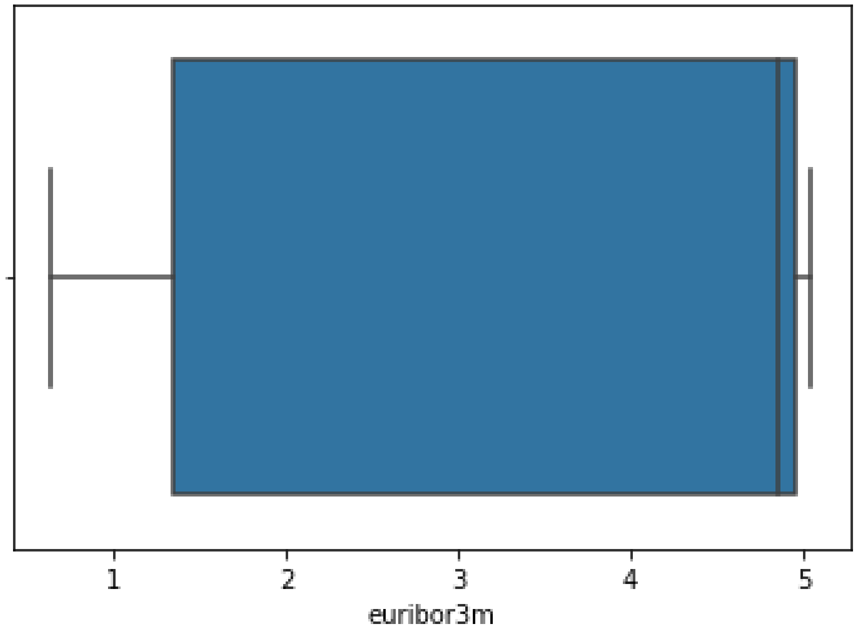

CONS.PRICE.IDX
cons.price.idx
95.0
94.5
94.0
93.5
93.0
92.5
92.0
y
no
yes
no
yes
y

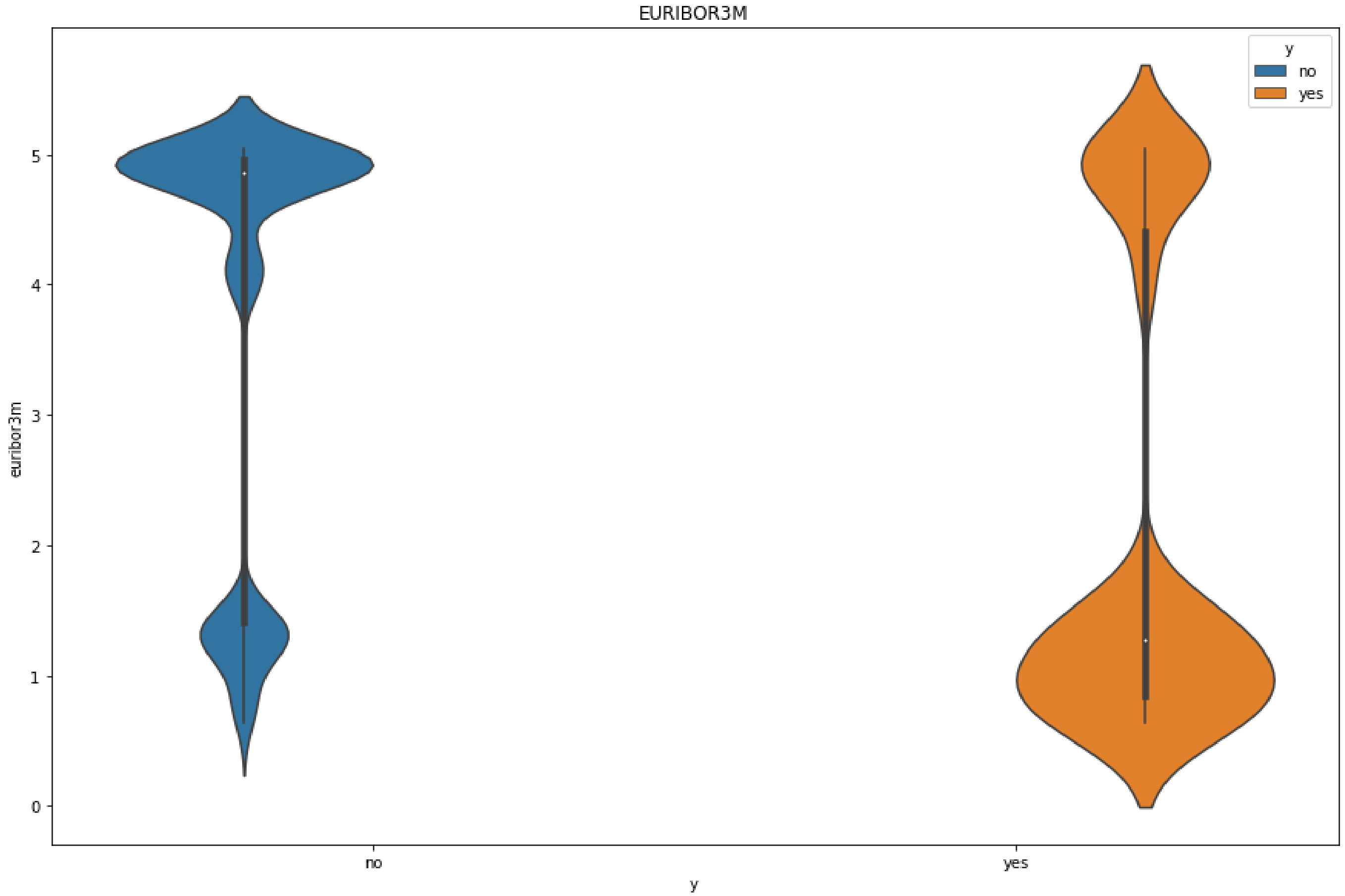

92.5
93.0
93.5
94.0
94.5
cons.price.idx

- From the 2 plots above, cons.price.idx does not contain any outliers and they would also be very much helpful in predicting class labels.

Feature:euribor3m(numeric)

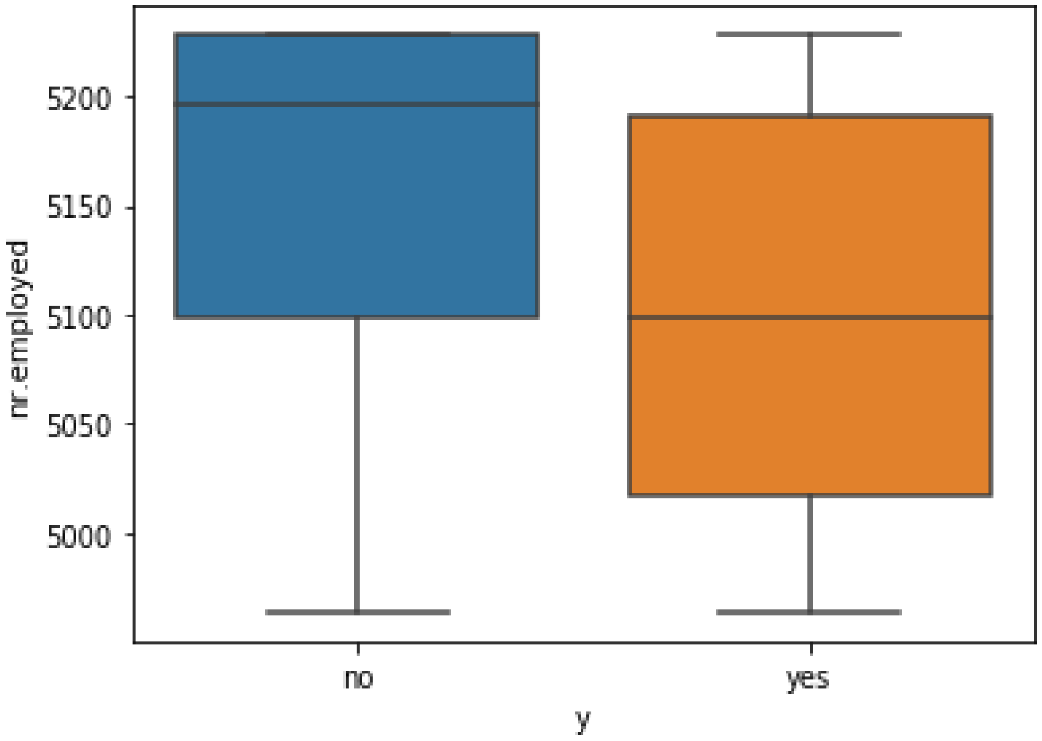


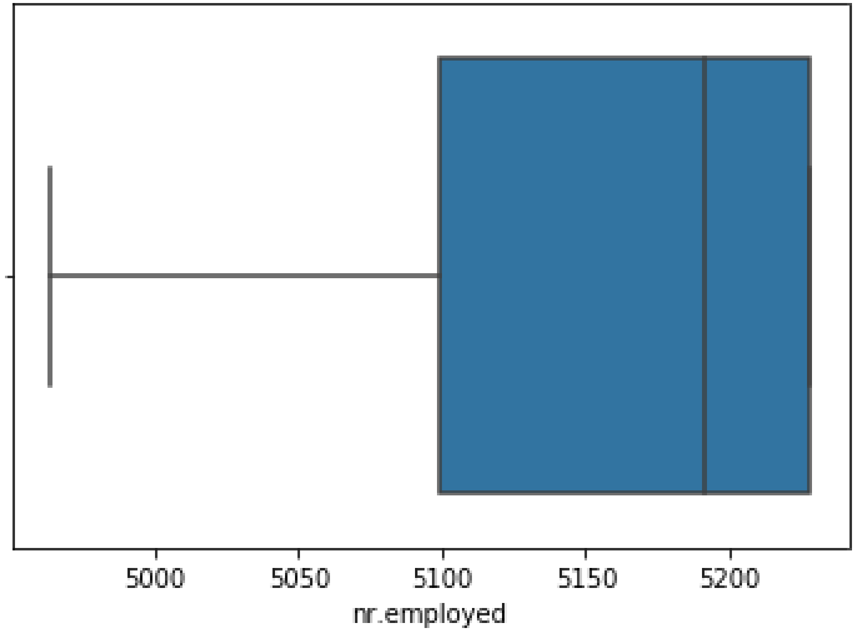

1
2
3
4
5
euribor3m

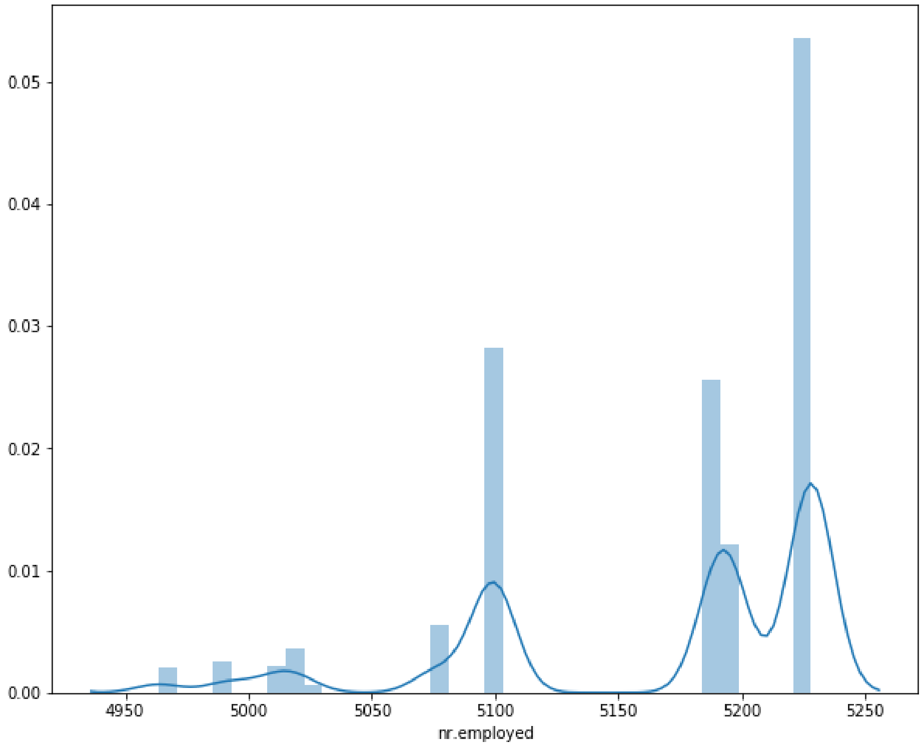

EURIBOR3M
y
no
yes
euribor3m
0
1
2
3
4
5
no
yes
y

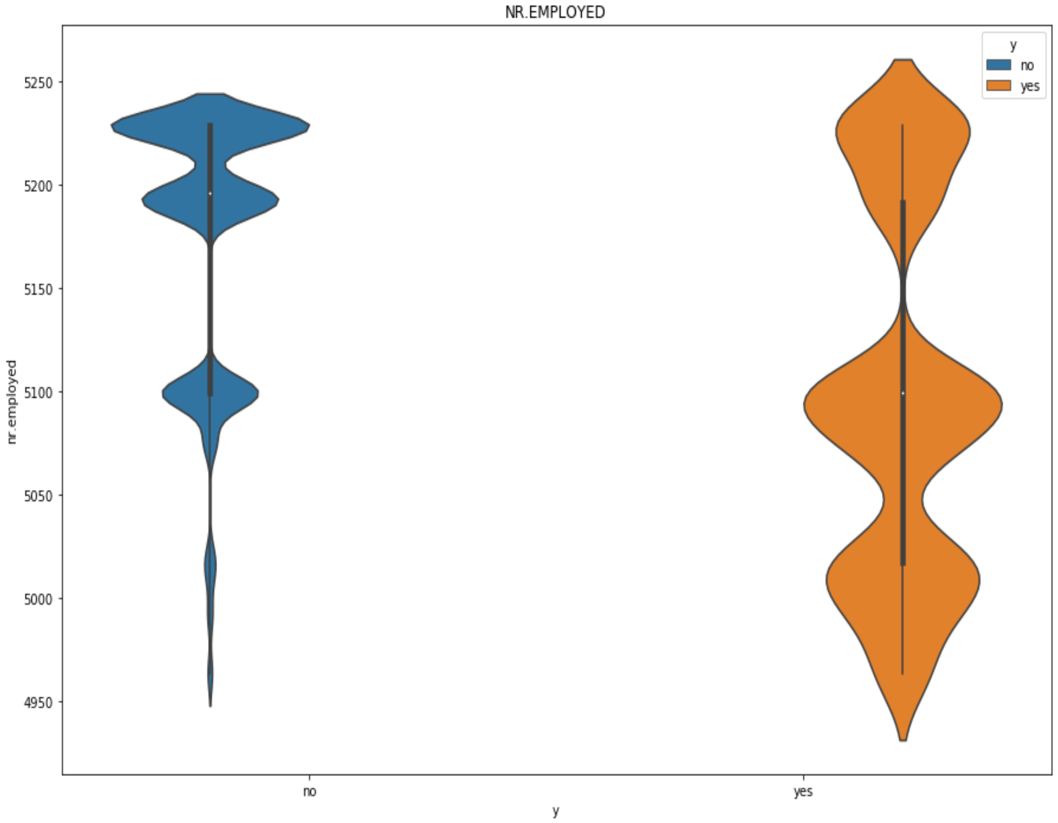
- From the above plots,Euribor3m does not contain outliers and would be very much helpful in predicting the class labels.

- **Feature:nr.employed(numeric)**



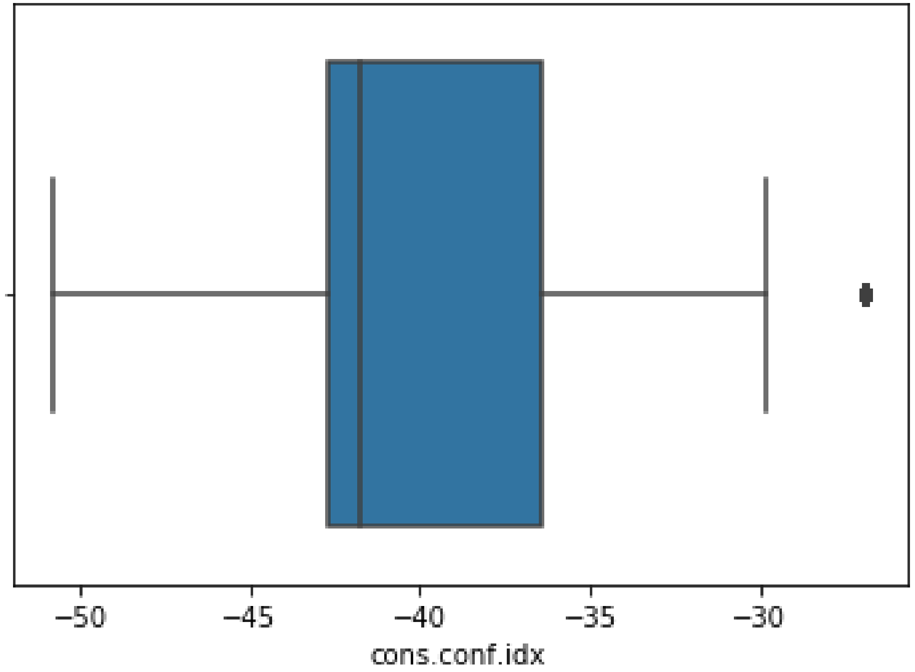

5000
5050
5100
5150
5200
nr.employed

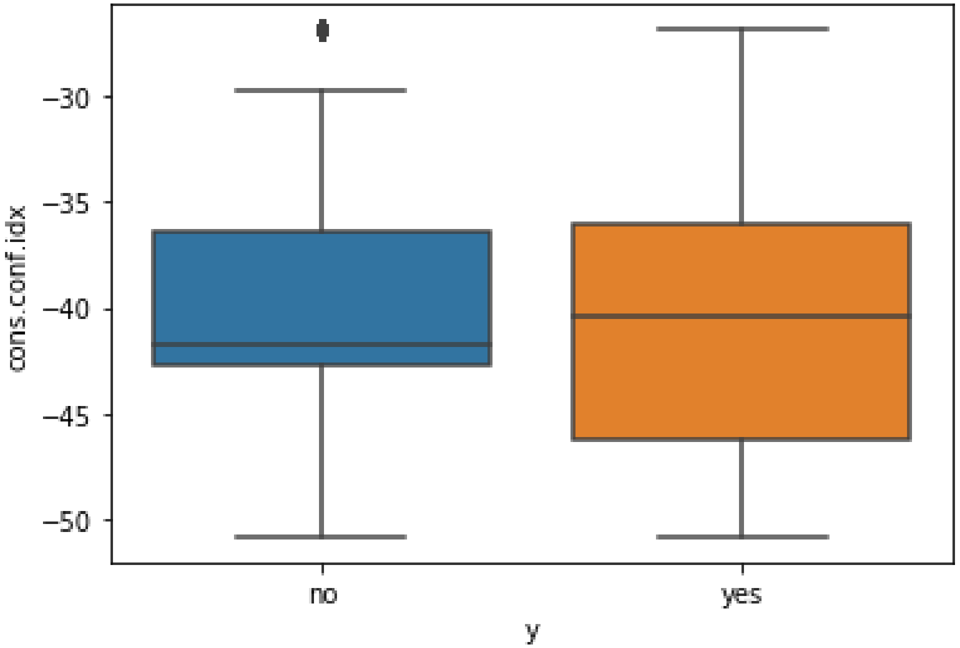

0.05
0.04
0.03
0.02
0.01
0.00
4950
5000
5050
5100
5150
5200
5250
nr.employed


NR.EMPLOYED
y
no
yes
5250
5200
5150
5100
5050
5000
4950
nr.employed
no
yes
y

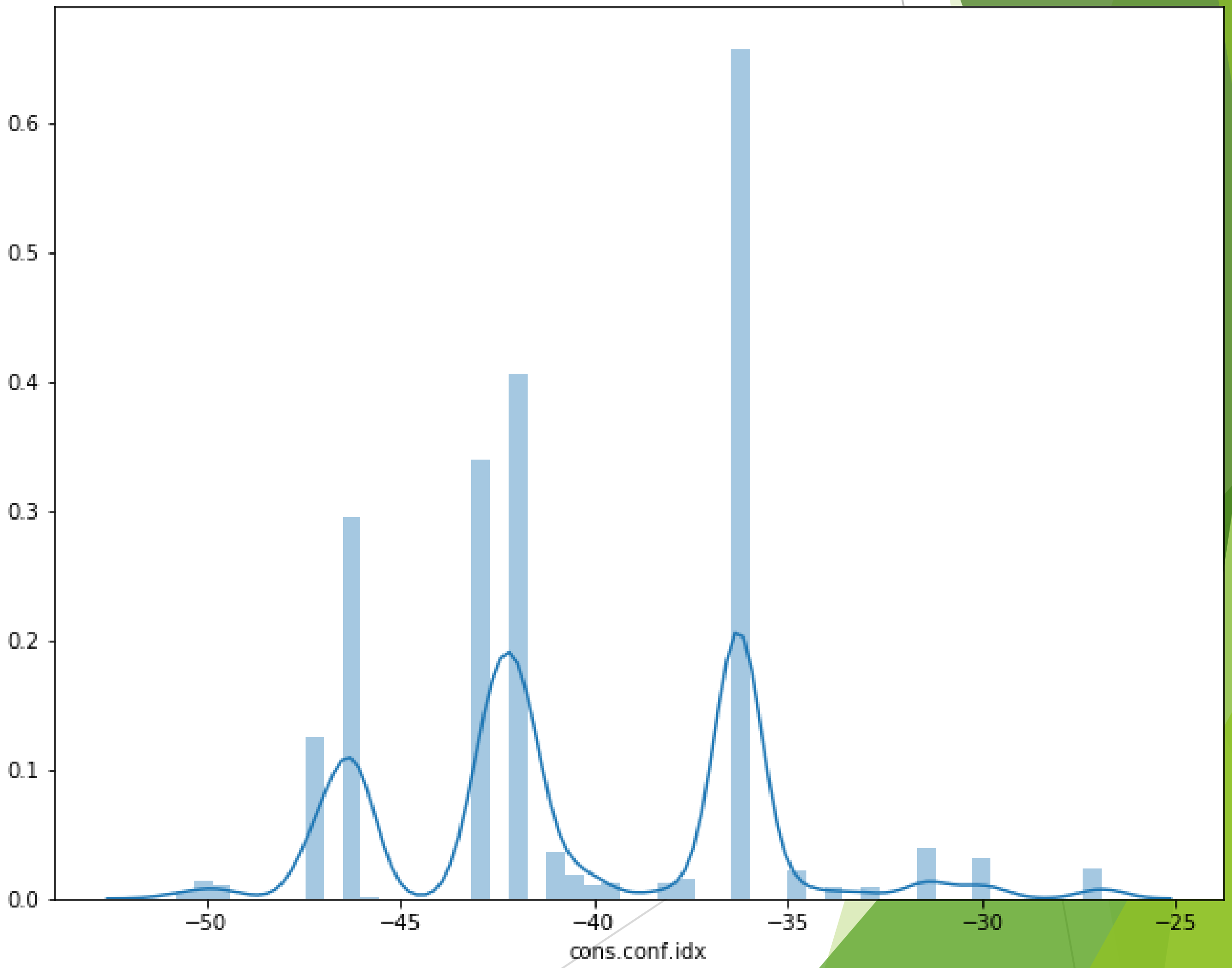
- From the plots above , nr. employed does not contain outliers and nr. employed would also be very much helpful in predicting class labels.

- **Feature:cons.conf.idx(numeric)**

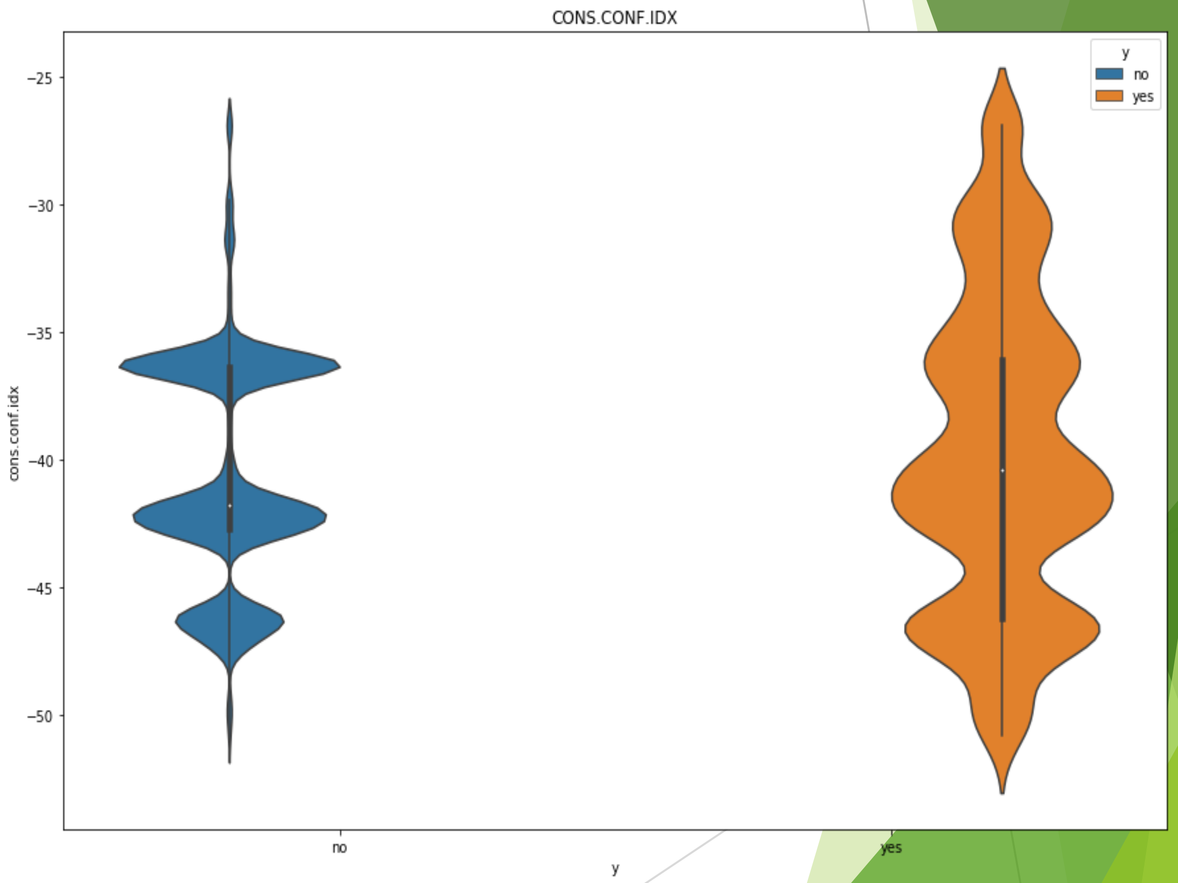



−30
−35
−40
−45
−50
cons.conf.idx
no
yes
y

- From the plot there might be a case of outliers. Let’s check with violinplot.

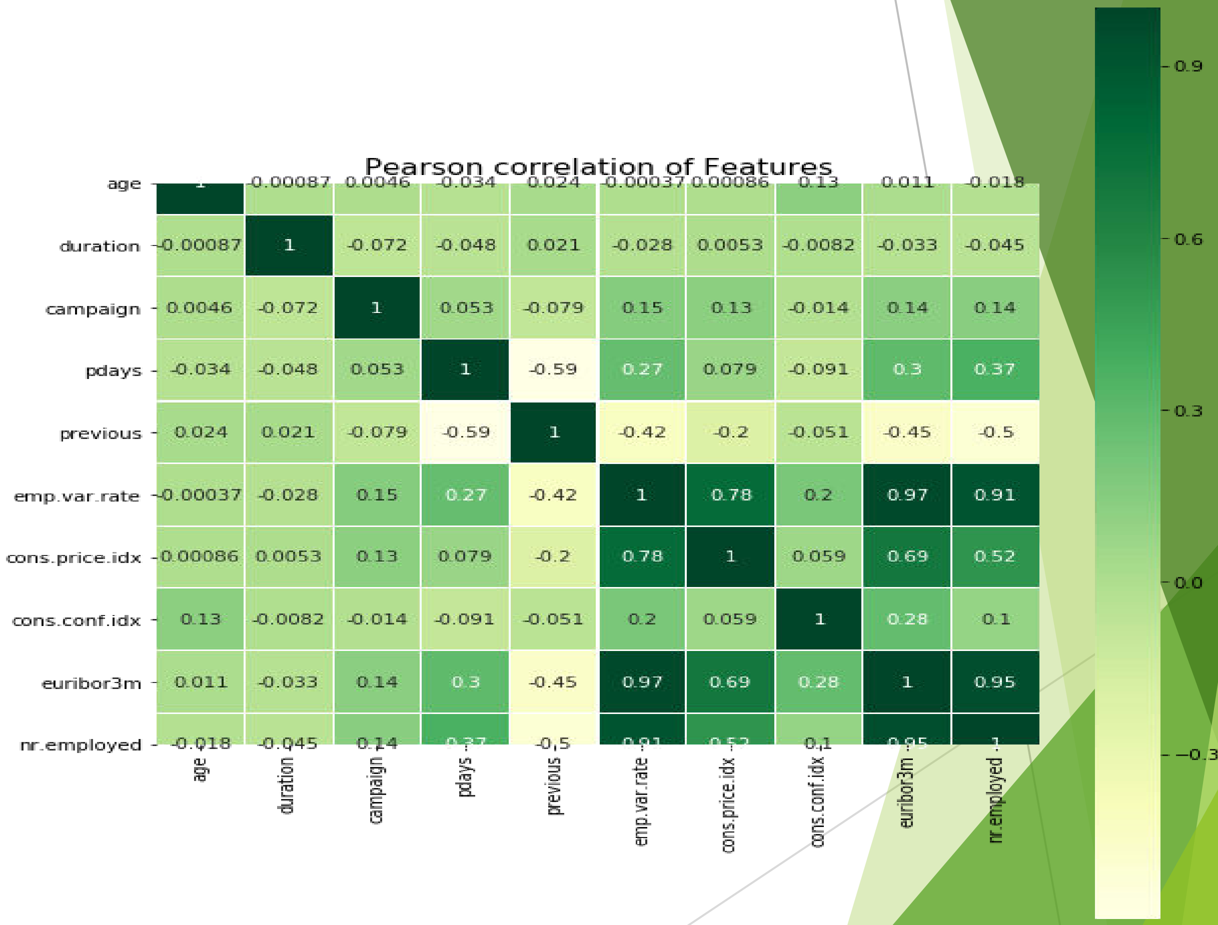


- In cons.conf.idx feature for class labels no, there is an outlier present when value above -30.



Correlation among features:

- Pearson correlation used.
- Let’s check for correlation of features between the numerical features.



Pearson correlation of Features

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
age	1	-0.00087	0.0046	-0.034	0.024	-0.00037	0.00086	0.13	0.011	-0.018
duration	-0.00087	1	-0.072	-0.048	0.021	-0.028	0.0053	-0.0082	-0.033	-0.045
campaign	0.0046	-0.072	1	0.053	-0.079	0.15	0.13	-0.014	0.14	0.14
pdays	-0.034	-0.048	0.053	1	-0.59	0.27	0.079	-0.091	0.3	0.37
previous	0.024	0.021	-0.079	-0.59	1	-0.42	-0.2	-0.051	-0.45	-0.5
emp.var.rate	-0.00037	-0.028	0.15	0.27	-0.42	1	0.78	0.2	0.97	0.91
cons.price.idx	0.00086	0.0053	0.13	0.079	-0.2	0.78	1	0.059	0.69	0.52
cons.conf.idx	0.13	-0.0082	-0.014	-0.091	-0.051	0.2	0.059	1	0.28	0.1
euribor3m	0.011	-0.033	0.14	0.3	-0.45	0.97	0.69	0.28	1	0.95
nr.employed	-0.018	-0.045	0.14	0.37	-0.5	0.91	0.52	0.1	0.95	1

- Some numerical features which share a high correlation between them, e.g., **nr. employed** and **euribor3m** these features share a correlation value of 0.95, and **euribor3m** and **emp.var. rate** share a correlation of 0.97, which is very high compared to the other features that we see in the heatmap.

Feature Engineering:

- Is the process of converting data into features that improves the prediction and performance of model in unseen data.

➢ Converting an age(numerical) variable to a categorical variable.

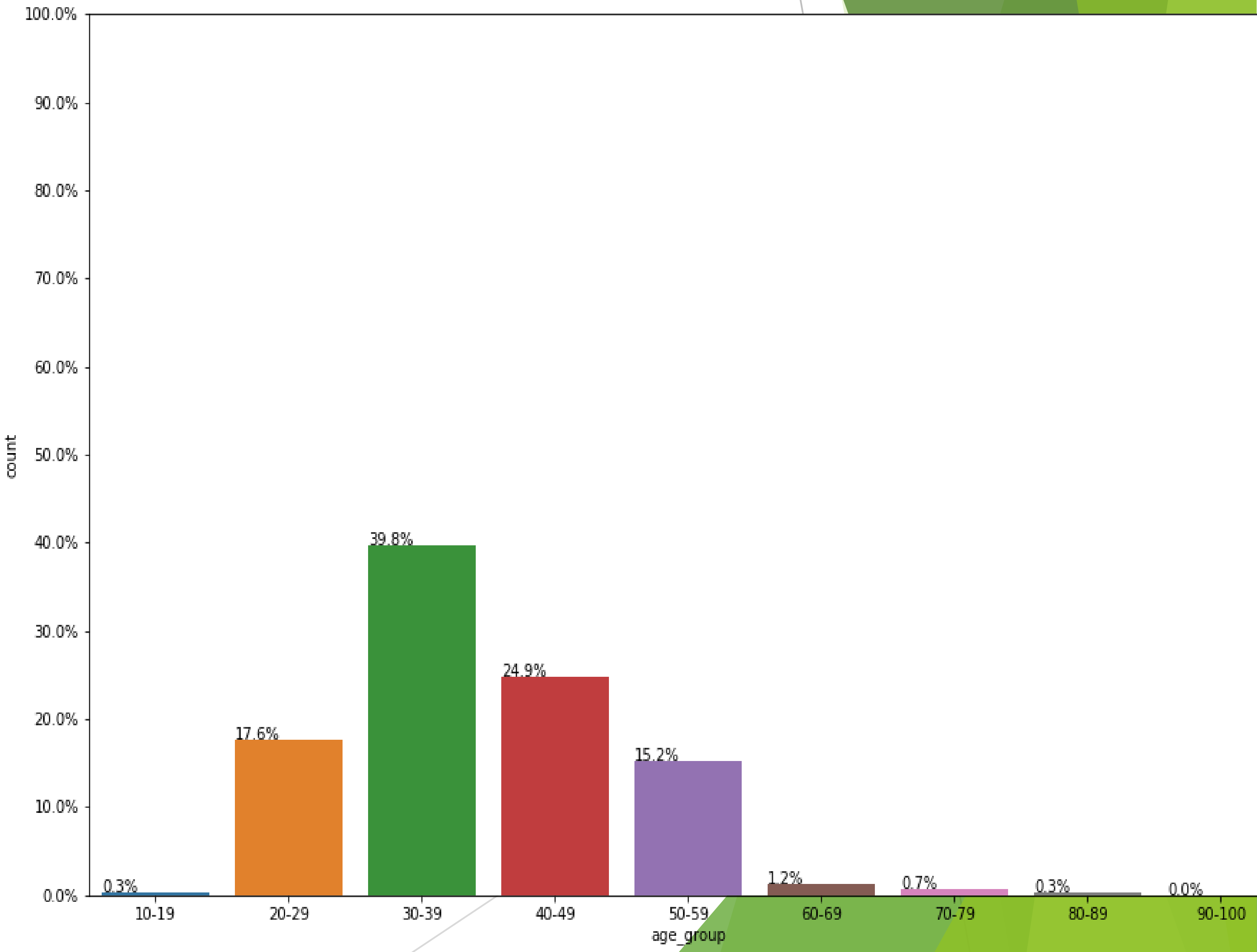
- Here I have created 9 groups from minimum age 10 to maximum age 100.
- After creating, inserted the age group into data frame and deleted the age column from the data frame.

➢ Creating i_loan column; Deleting Housing and Loan column

- I have created i_loan based on the columns['loan','housing'], status we have 3 statuses yes, no, unknown.
- If any 2 columns have status yes then ,i_loan will have yes as status, if any of the columns have no as status, then i_loan will have no as status or else unknown status.

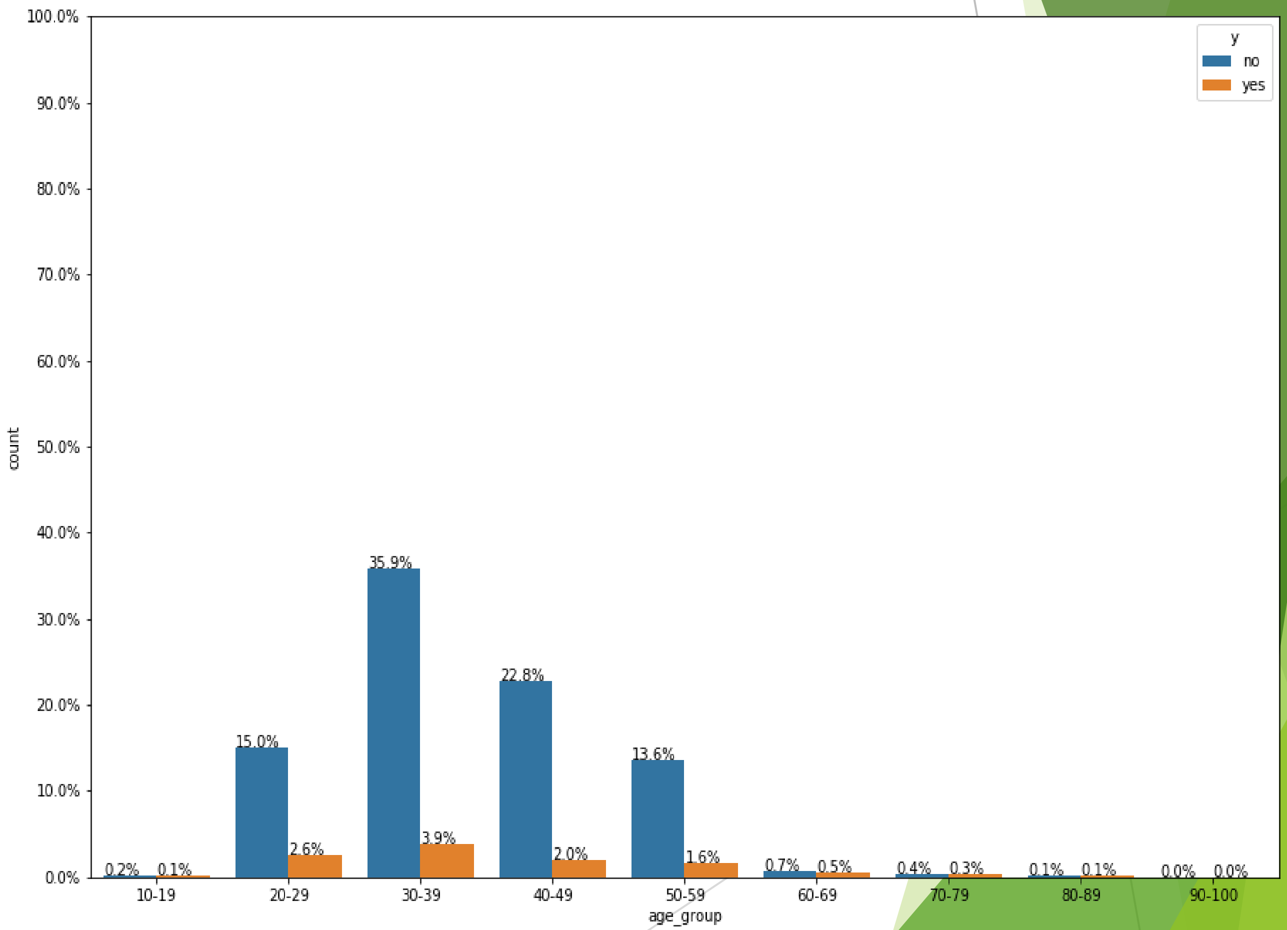
Which age group has the bank contacted the most?

- As shown , the bank has contacted to the most between the age group of 30-39 followed by 40-49.



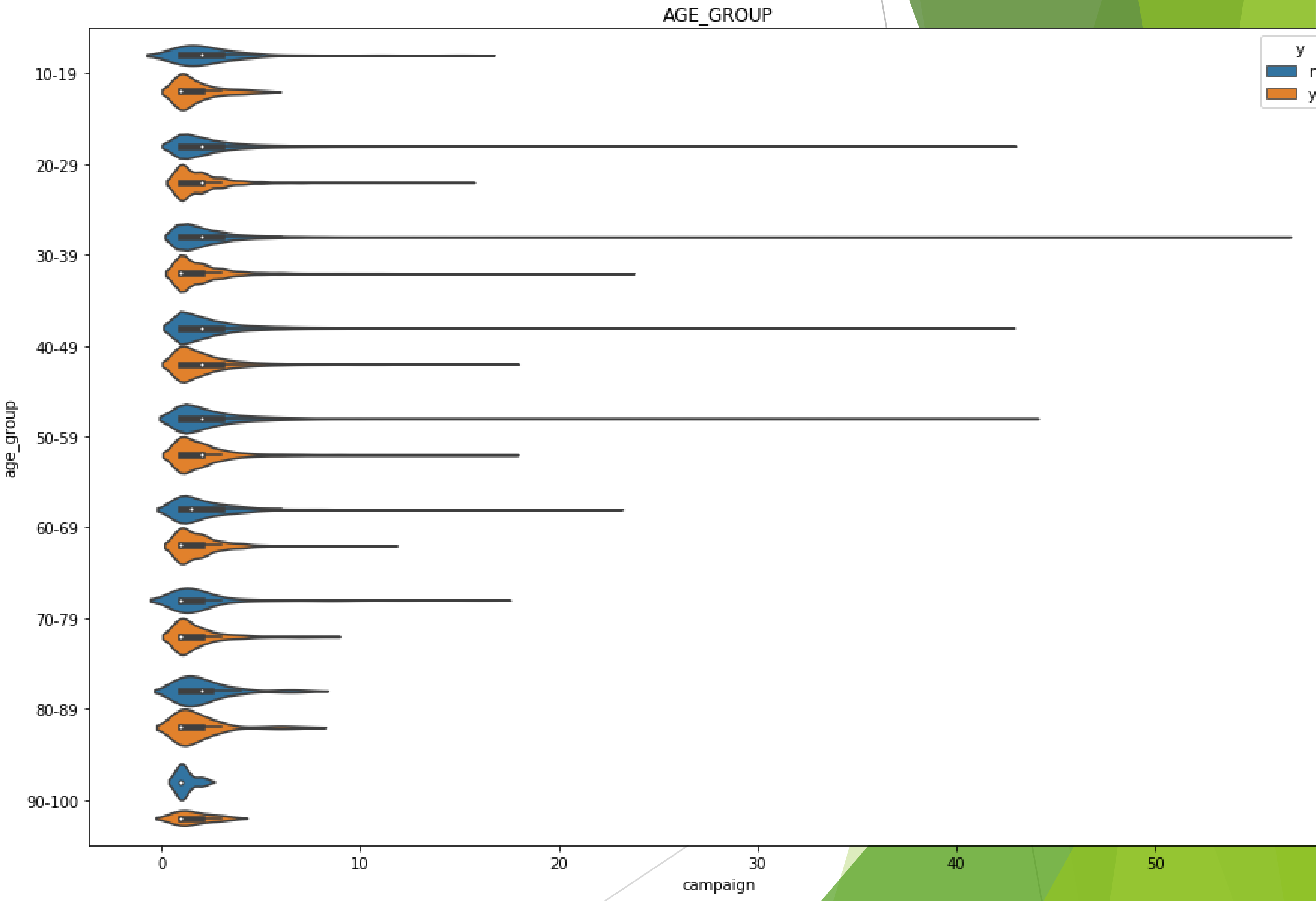
➢ Which age group is most likely to subscribe for long term deposits?

- Age group of 30-39 are the most people who have not subscribed for the deposits.
- They are also the most who have subscribed for the deposits



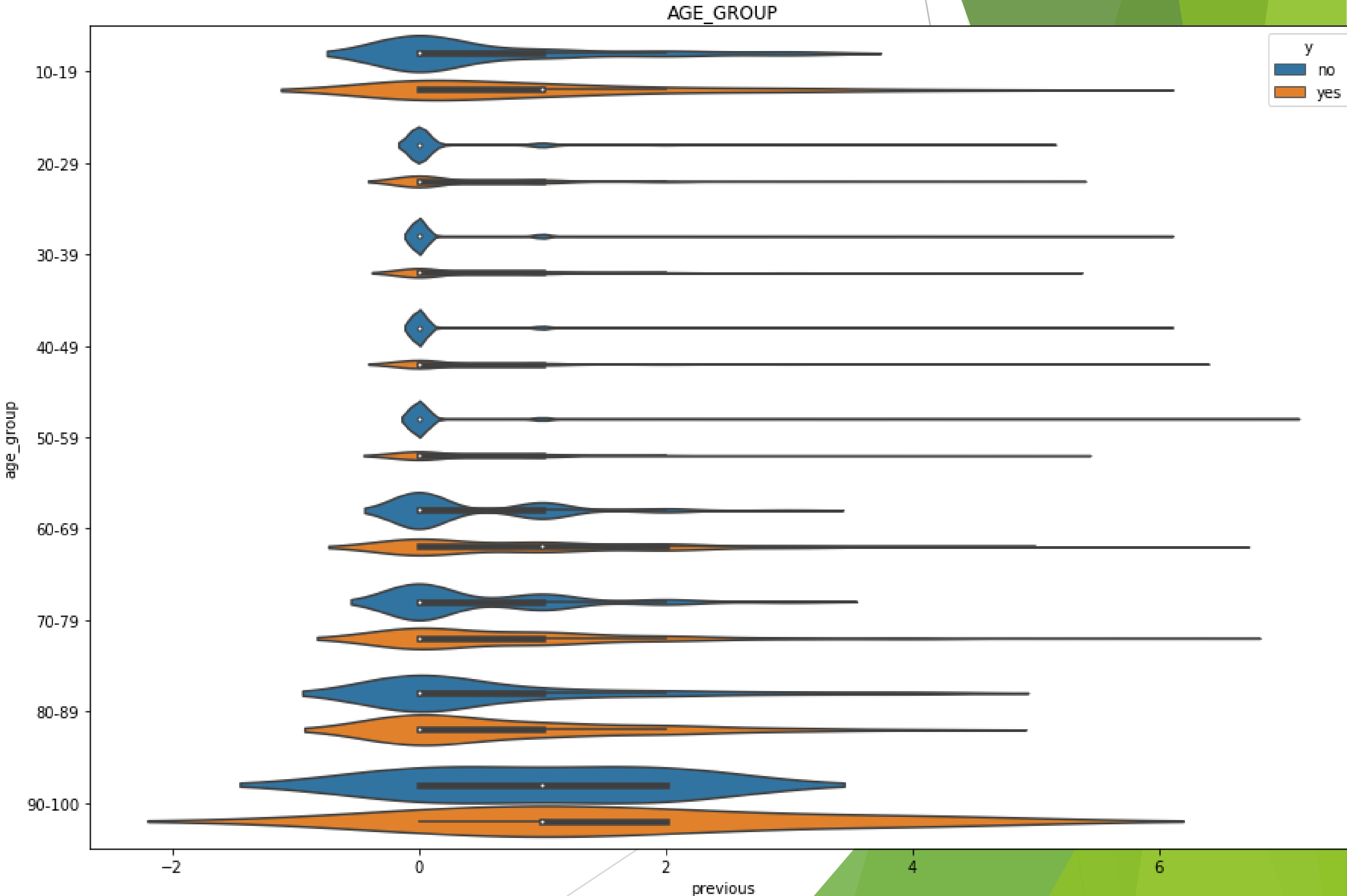
➢ Age_group and campaign

- Number of campaigns from 1-7 and age above 70 has possible outliers.



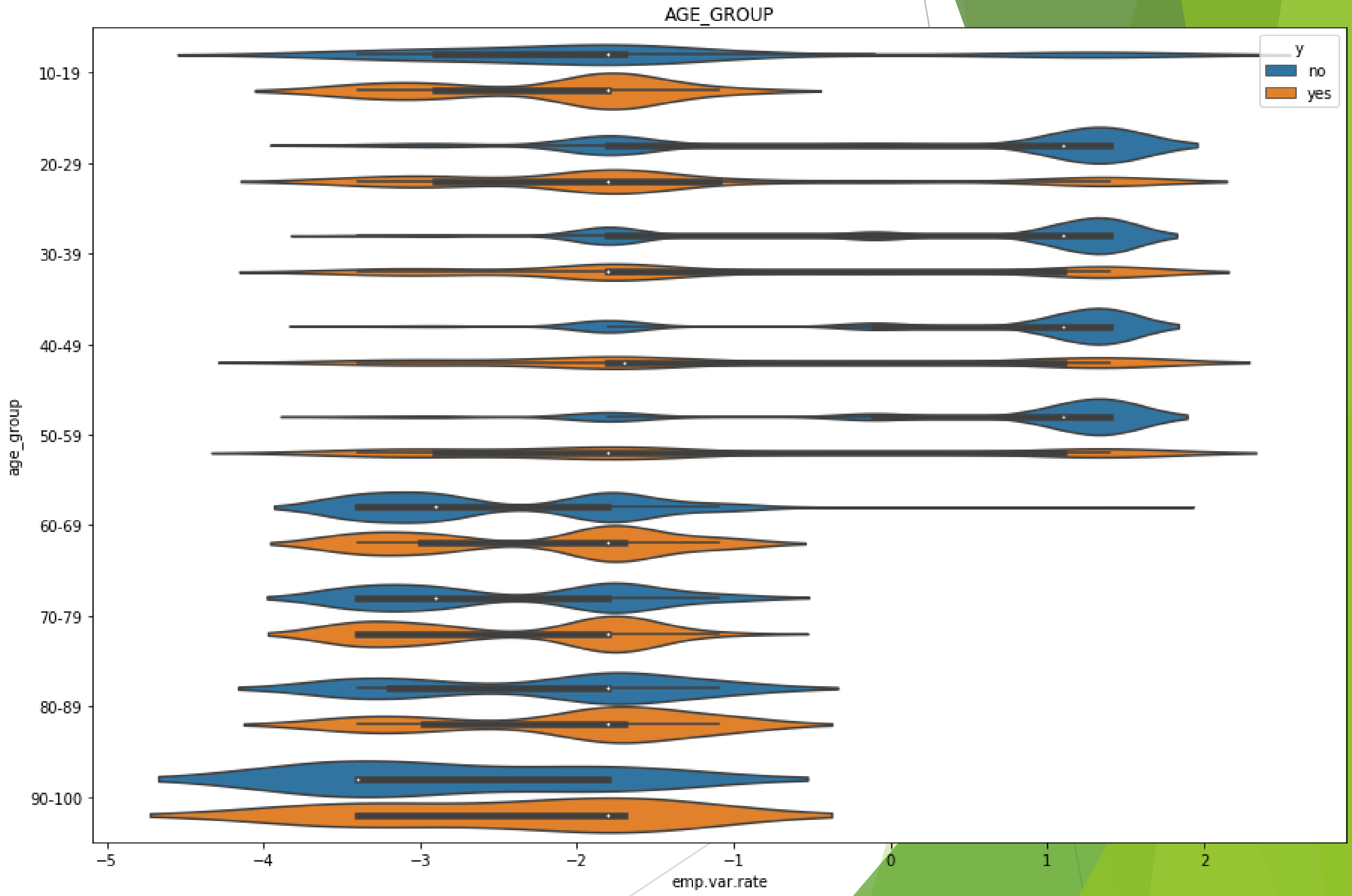
➢ Age_group and previous

- For number of previous campaigns from 0-1 age group above 70 are possible outliers.
- For number of contacts for previous campaigns as 2, age around 90 are possible outliers.
- For number of previous campaign from 3-4 age around 80 are possible outliers.



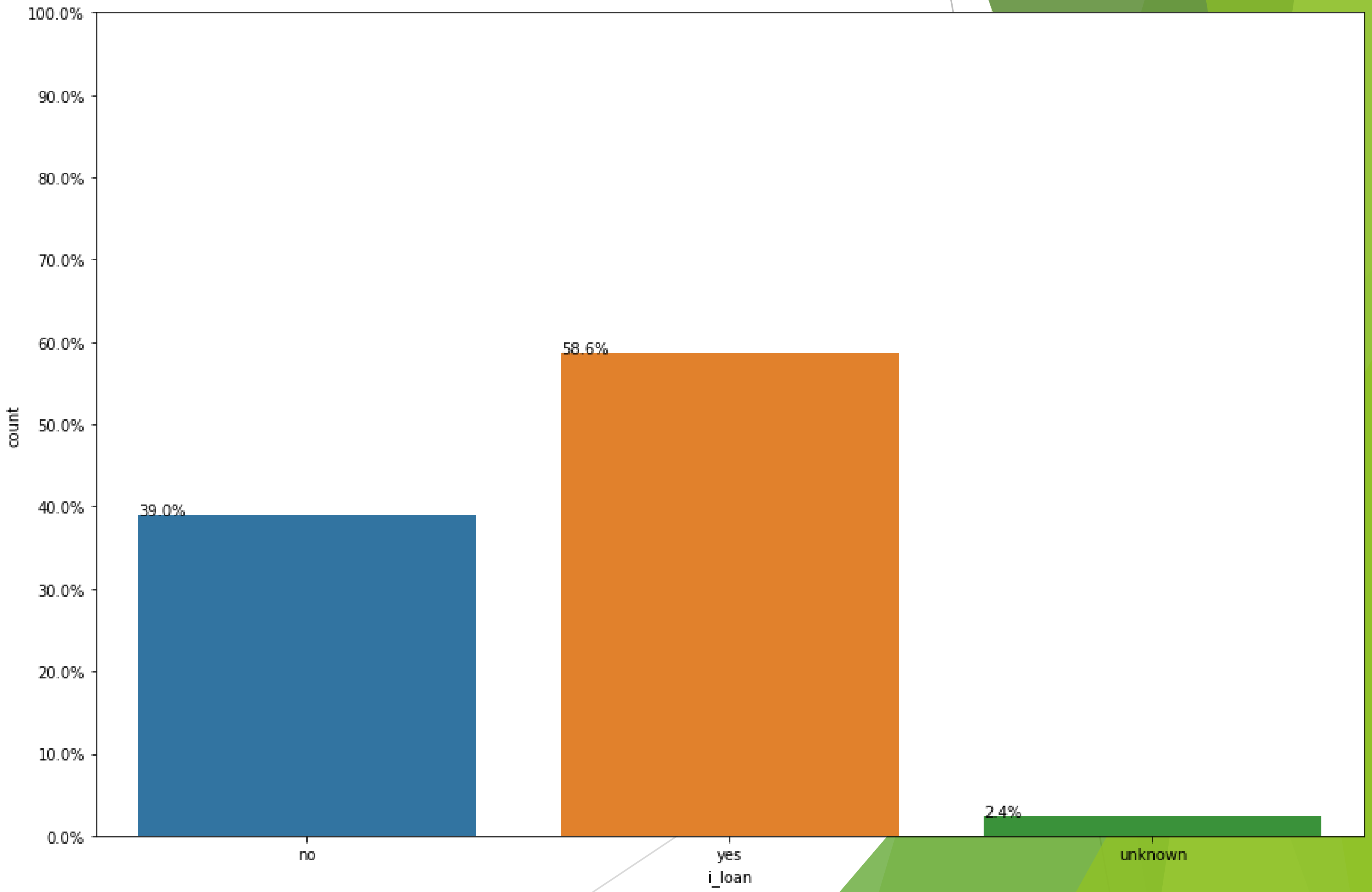
➢ Age_group and emp.var.rate

- For emp.var.rate with -1.8 has most outliers above age around 60.



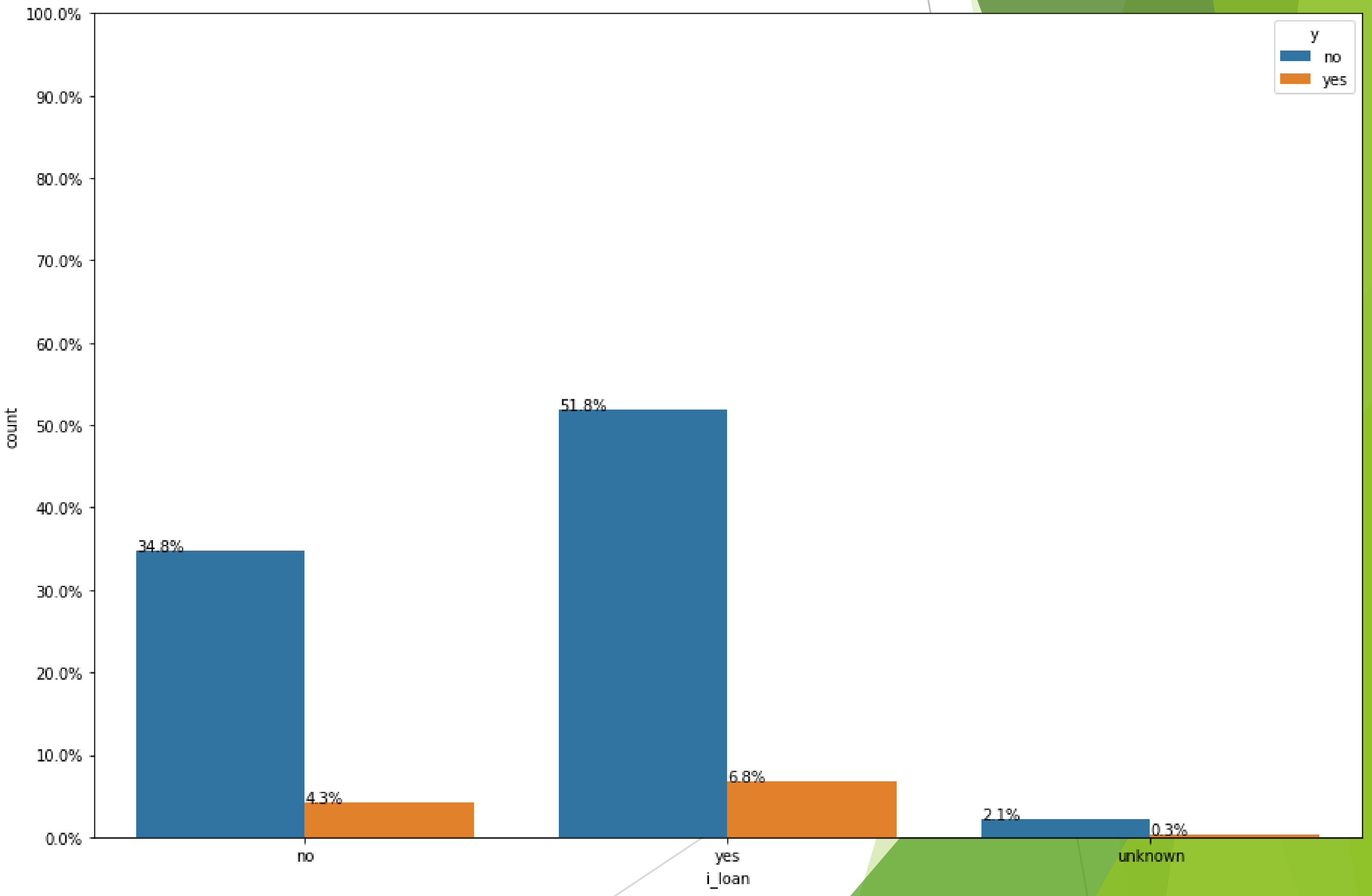
➢ i_loan

- Most of the people the bank has contacted either have personal or housing loan. Very few of the status of loan is unknown.



➢ Do people who have loans are the most that have subscribed for deposits?

- People who have loans are in majority who have subscribed for deposits.
- They are also the ones who have not subscribed for the deposits.



- For the other EDA of the following:
  - Age_group and job
  - Age_group and education
  - Age_group and i_loan
  - Education and job
  - Education and marital
  - I_loan and education
  - Marital and job
  - I_loan and marital
  - I_loan and job

you can check out and review the full code on the following GitHub repository link:

• Recommendation summary

- After I've conducted some analysis through visualizations using plots, it revealed as follows:
- People who are in admin job has been more contacted for the deposits by the bank.
- People who are married has been contacted more for the deposits by the bank.
- People who has been contacted more on the cellular than the telephone.
- People has been contacted more in the month of May than any other month. They have not been contacted in January and February at all.
- People has not been contacted on Saturday and Sunday.
- People with no default status has been contacted more by the bank.
- People who has housing loan has been contacted more by the bank
- People with no personal loan has been contacted more by the bank.
- People who are in university has been contacted more by the bank.
- Age,Duration,Campaign have outliers and are rightly skewed.
- Pdays have more than 70% of data imputed so it is better either to impute or remove the column.
- Euribor3m with nr.employed and emp.var.rate with nr.employed with the highest correlation.

• Recommended models

- Remember,our goal is to predict whether a customer will subscribe a term deposit or not given the data of the customer.

So, what type of machine learning problem is this?

This is a binary classification problem.

I have chosen the following models recommended for this problem:

- Dummy Classifier(Stratified)
- KNN-classifier

- Logistic Regression
- SGD-log loss
- Linear SVM
- Random Forest
- XGBoost
- Adaboost

Each of these models will be encoded with different encoding methods,with either One Hot Encoding or Response Encoding.

Recall:

- **One Hot Encoding:One hot encoding** creates column for each category and checks

whether the category is present in that row or not. If category is present, it would be marked as 1 else zero.

- **Response Encoding:** As part of this technique, we represent the probability of the data point

belonging to a particular class given a category. So, for a K-class classification problem, we get K new features which embed the probability of the datapoint belonging to each class based on the value of categorical data.

Data modelling steps:

- The duration variable will only be used for baseline model.Since to make the models realistic we would be dropping column for other machine learning models.
- Hyper tuning of models has been done, using calibrated classifier, where cross validation dataset is used to get best hyper parameter.
- After getting best hyper parameter, train and test data set is used for evaluating the model performance.
- Feature scaling of numerical data is done only for linear models. Tree based models does not require feature scaling and is also robust to outliers.
- For each encoding of categorical data, we have used models to compare which encoding would work better.
- Since the dataset is highly imbalanced, I have used class_weight='balanced' as parameter to balance the dataset internally.

- **Calibrated Classifier**

- What is the need to use Calibrated Classifier?

- Non-linear models like (KNN, Tree based Models) predicts uncalibrated probability. Though models predict probability, but they might not be the same like observed probability in training. It requires adjustment and that is done by calibration.

➢ Performance metric

- For this problem I will be using the **ROC-AUC curve.**
- The **ROC –AUC curve** shows the performance of binary class classifiers across the range of all possible threshold plotting between true positive rate and 1-false positive rate.
- Measures the likelihood of that given two random points one from positive and one from negative the classifier will rank the positive points above negative points.
- Is a popular classification metric that present the advantage of being independent of false positive and false negative.
- The ideal **AUC score** is 1 and **AUC** of 0.5 is for random classifier.

Other performance metrics required

- Macro-F1 score
- **Macro-F1 Score: F1 score** is the harmonic mean between Precision and Recall.
- **Macro F1 score** is used to know how our model works in overall dataset.

- **Confusion Matrix:**
- This matrix gives the count of true negative,true positive,false positive and false negative datapoints.

Model building

- I will first begin with the machine learning models encoded using **One Hot Encoding** method.
- The following machine learning models are:
- Dummy Classifier(Stratified)
- KNN-classifier
- Logistic Regression
- SGD-log loss
- Linear SVM
- Random Forest
- XGBoost
- Adaboost

- After I’m done with the machine learning models encoded using One Hot Encoding,then

I will go with machine learning models encoded using **Response Encoding** method.

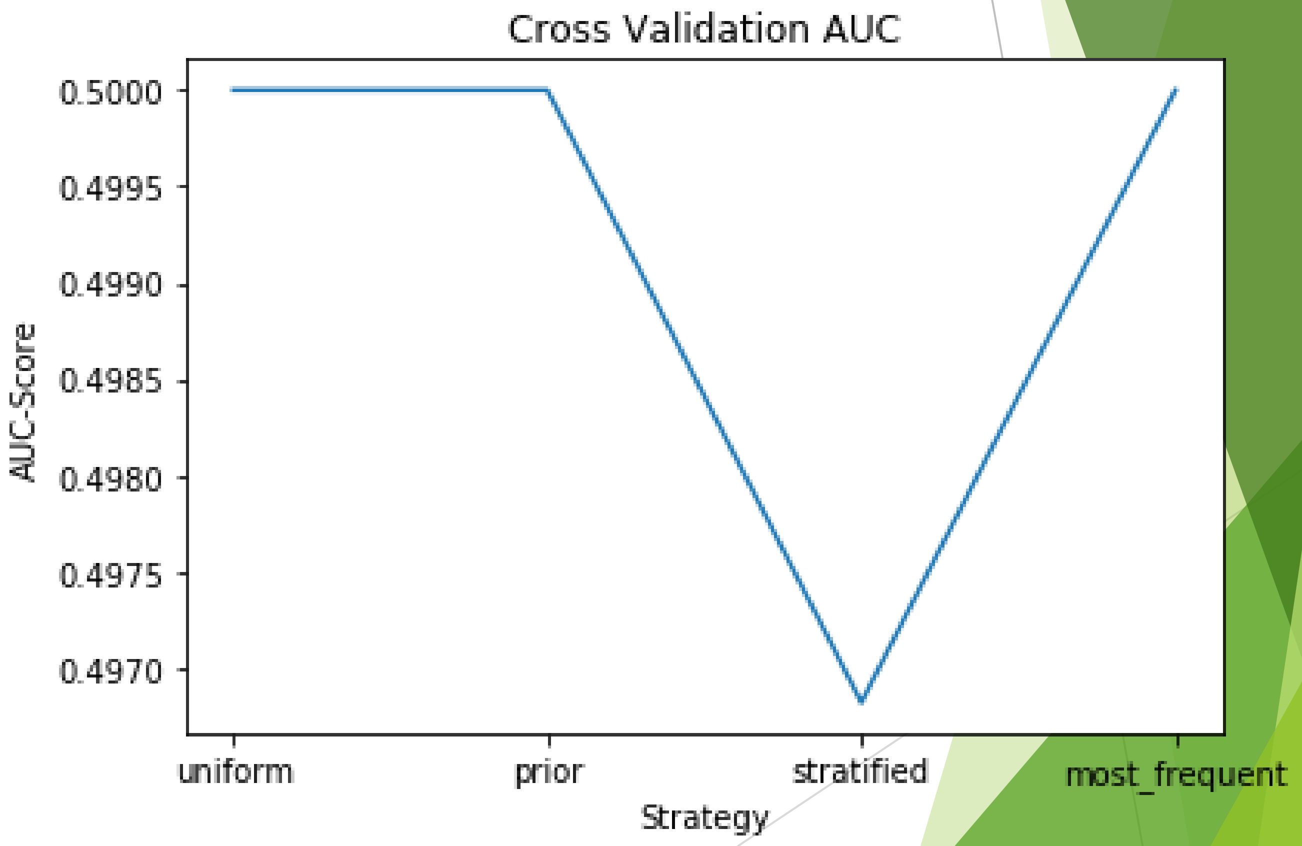
Then I will measure the performance of each of these machine learning models using the metrics ,

AUC score and F1-score.

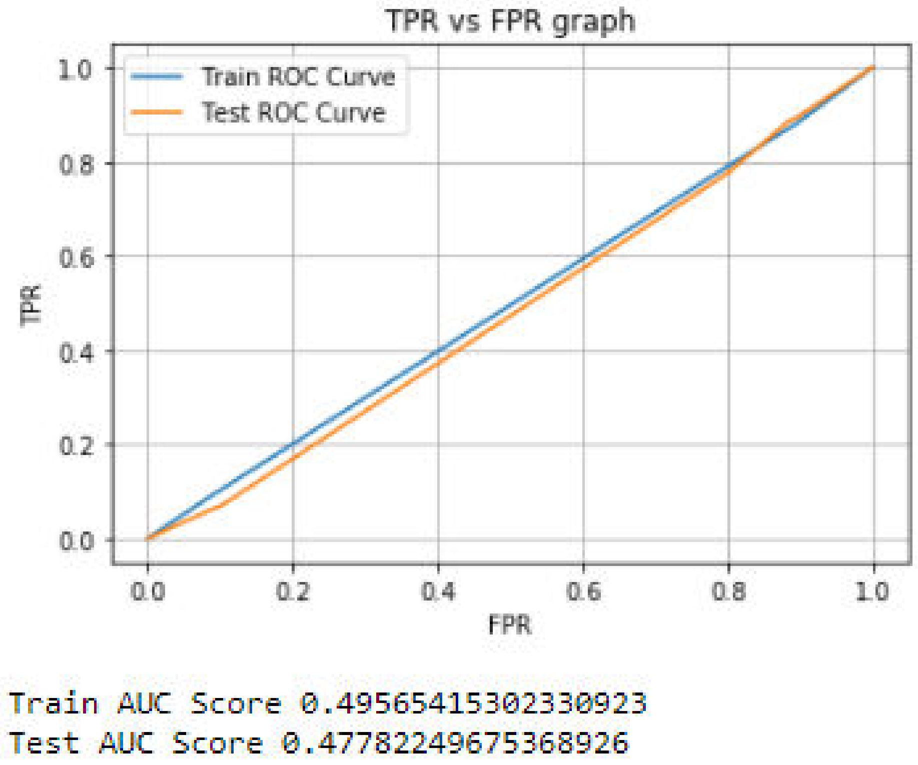
I will then select the best or champion model with the highest score.

1. Dummy Classifier(Stratified)
One Hot Encoding

- Strategy with Stratified has the best AUC score among other strategies,and that is why it is selected.



- For the Dummy Classifier ,we get a **Train AUC Score** of 0.4957 and a **Test AUC score** of **0.4778.** This is not a good performance. Now let's look at the F1 score .



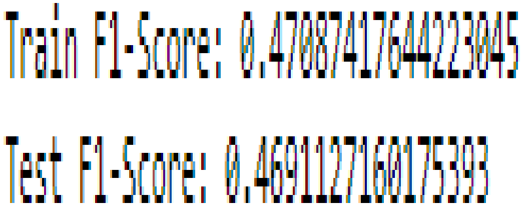
```
Train AUC Score 0.49565415302330923
Test AUC Score 0.47782249675368926
```

- We get a **Train F1-score** of **0.4709**

and a **Test F1-score** of **0.4691.**

- This is not a good performance either when measuring with the **F1-score.**
- Overall, the performance of the **Dummy Classifier(Stratified)** machine learning model encoded using **One Hot Encoding** method is not good.

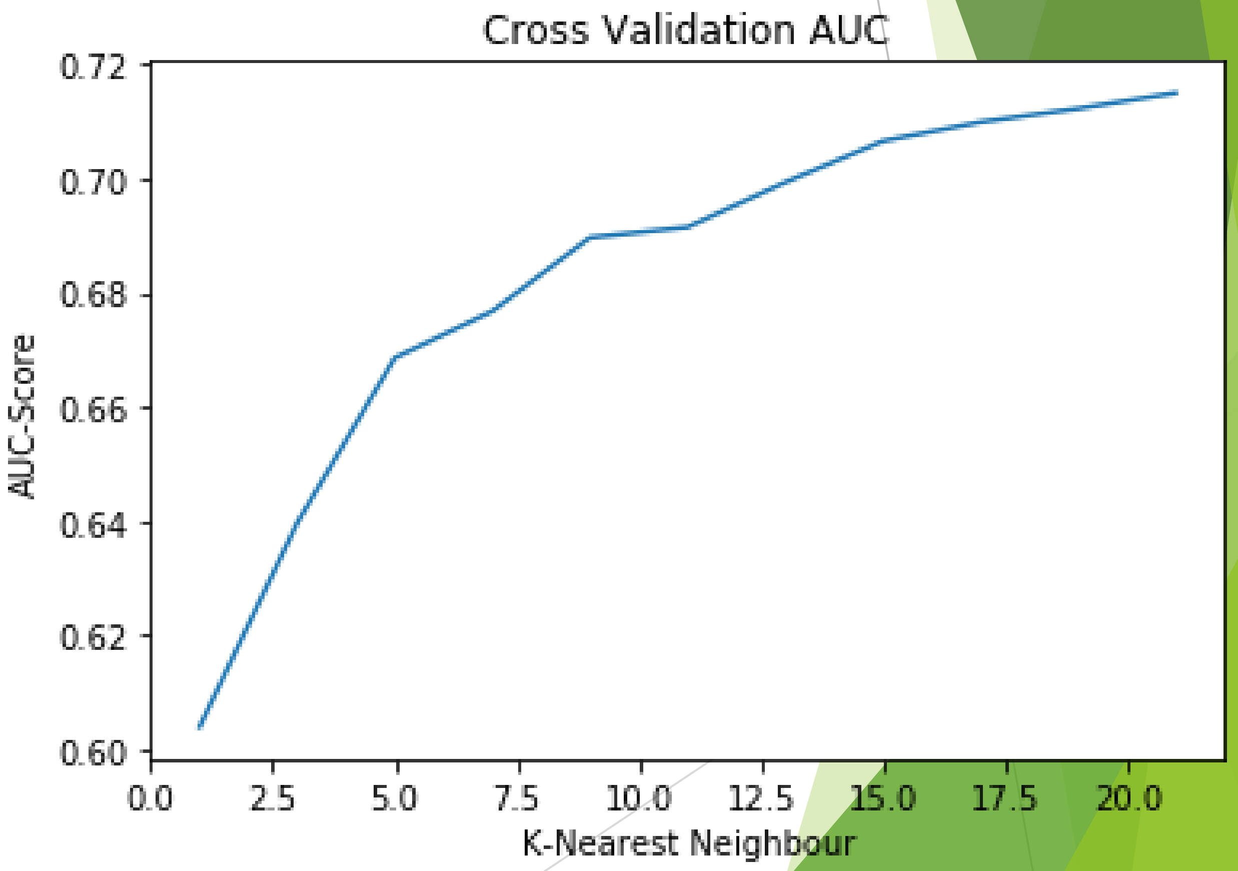
Let's work with the other machine learning models encoded with One Hot Encoding maybe they can perform better than this.



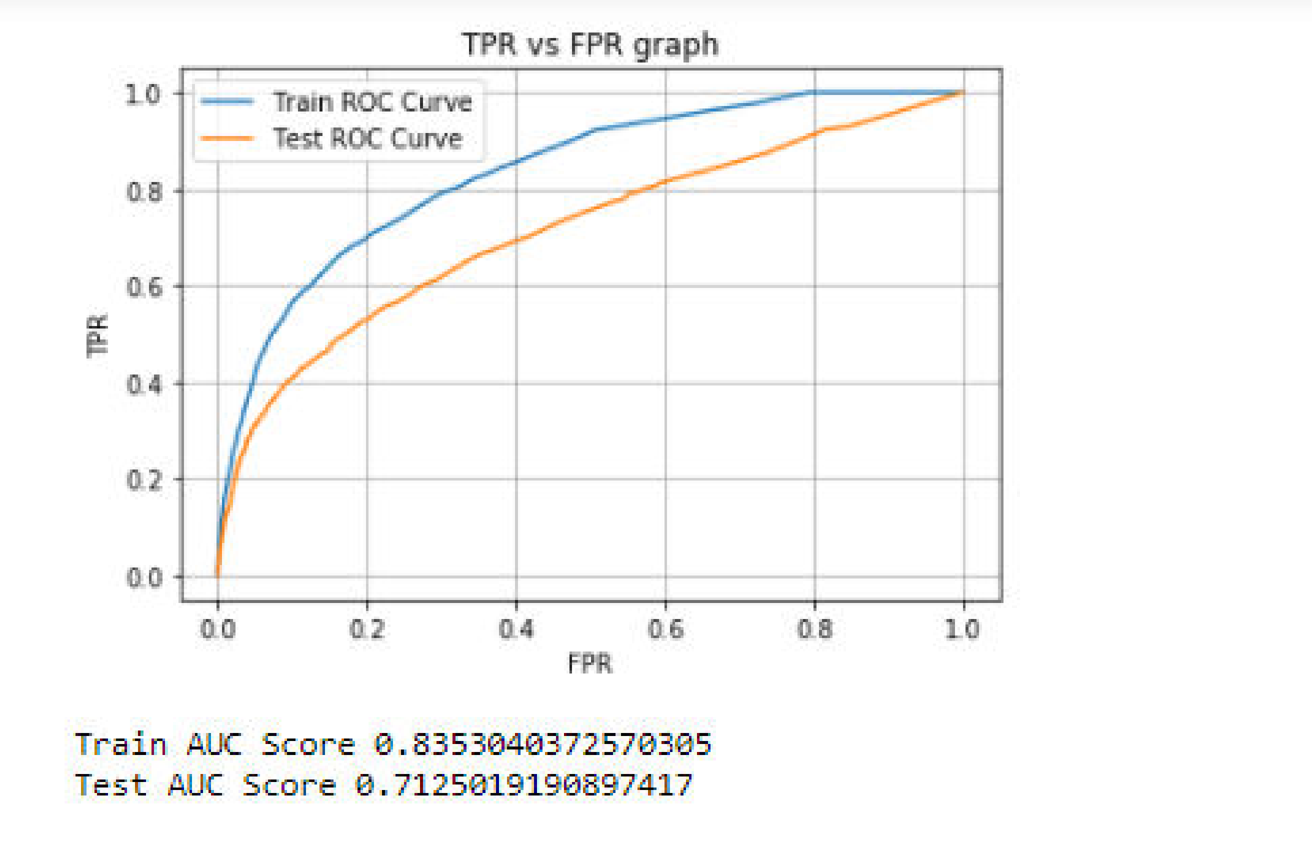
```
Train F1-Score: 0.47087417644223045

Test F1-Score: 0.4691127160175393
```

2.K-Nearest Neighbour

- Using the K-Nearest Neighbour,the best K value is 21.Lets find the **AUC score** and the **F1-score.**

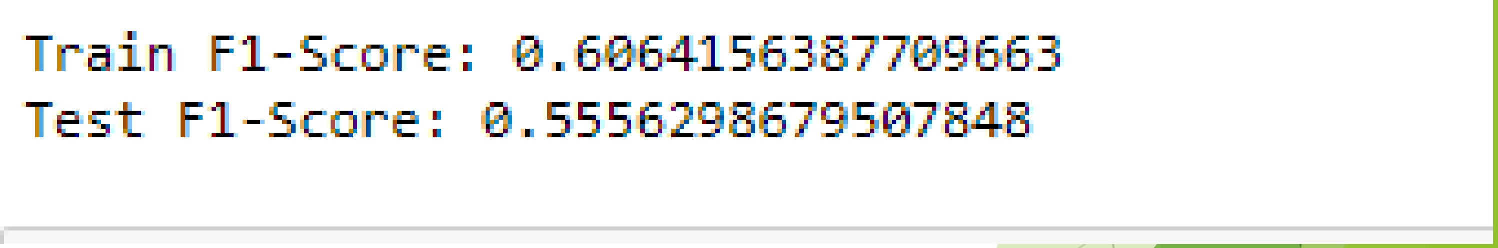


- For the K-Nearest Neighbour machine learning model we get a **Train AUC score** of **0.8353** and a **Test AUC score** of **0.7125.**
- This indicates a good performance.Now let's look at the F1-score.



```
Train AUC Score 0.8353040372570305
Test AUC Score 0.7125019190897417
```

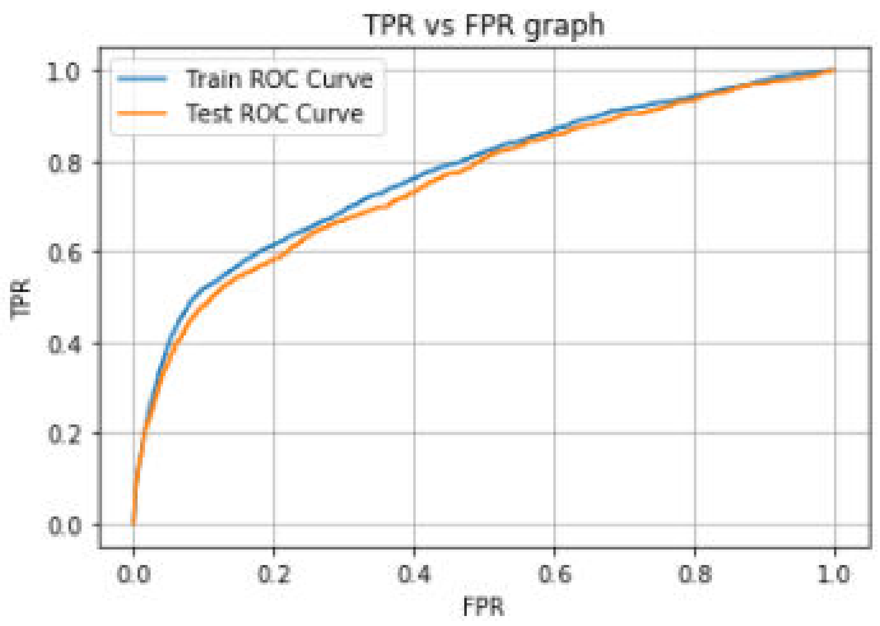
- We get a **Train F1-score** of **0.6064** and a **Test F1-score** of **0.5556** for the **K-NN model.**
- This is not bad.Now let's work with the **Logistic Regression model**.



```
Train F1-Score: 0.6064156387709663
Test F1-Score: 0.5556298679507848
```

3.Logistic Regression

- Using the Logistic Regression model, we get a **Train AUC score** of **0.7711** and a **Test AUC score** of **0.7530.**
- This also indicates a good performance.Now let's look at the F1-score.

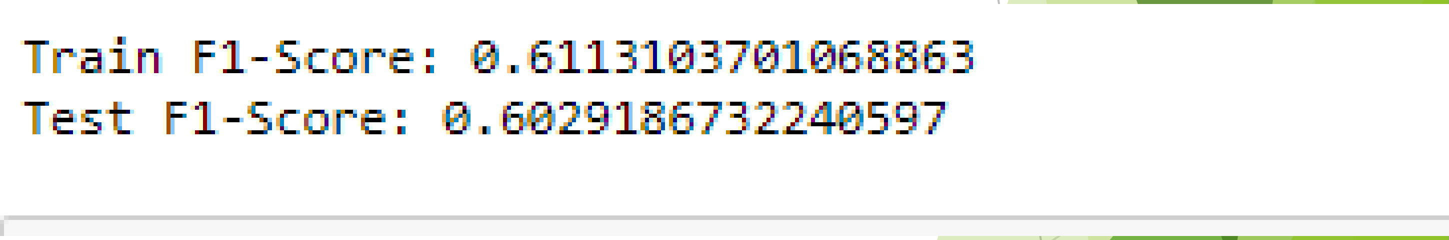


```
Train AUC Score 0.7711413716612388
Test AUC Score 0.752989762463322
```

- We get a **Train F1-score** of **0.6113**

and a **Test F1- score** of **0.6029.**

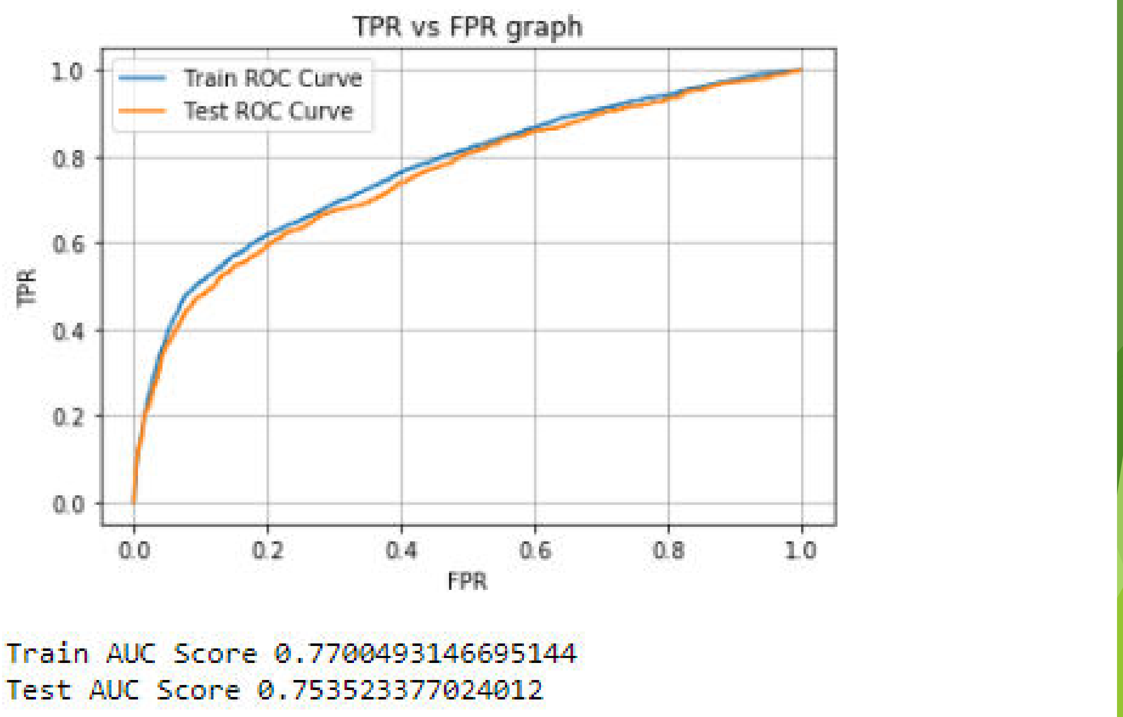
This is a slightly good performance.Now let's work with SGD Classifier with log-loss machine learning model.



```
Train F1-Score: 0.6113103701068863
Test F1-Score: 0.6029186732240597
```

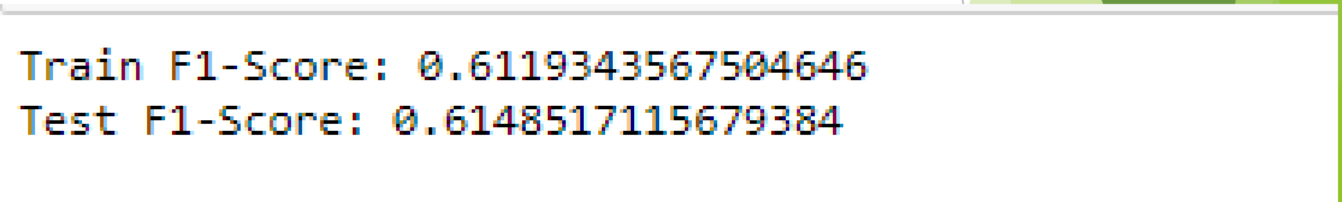
4.SGD Classifier with log-loss

- Using the SGD Classifier with log-loss we get a **Train AUC score** of **0.77** and a **Test AUC score** of **0.7535**.
- This also indicates a good performance. Now let's look at the F1-score.



```
Train AUC Score 0.7700493146695144
Test AUC Score 0.753523377024012
```

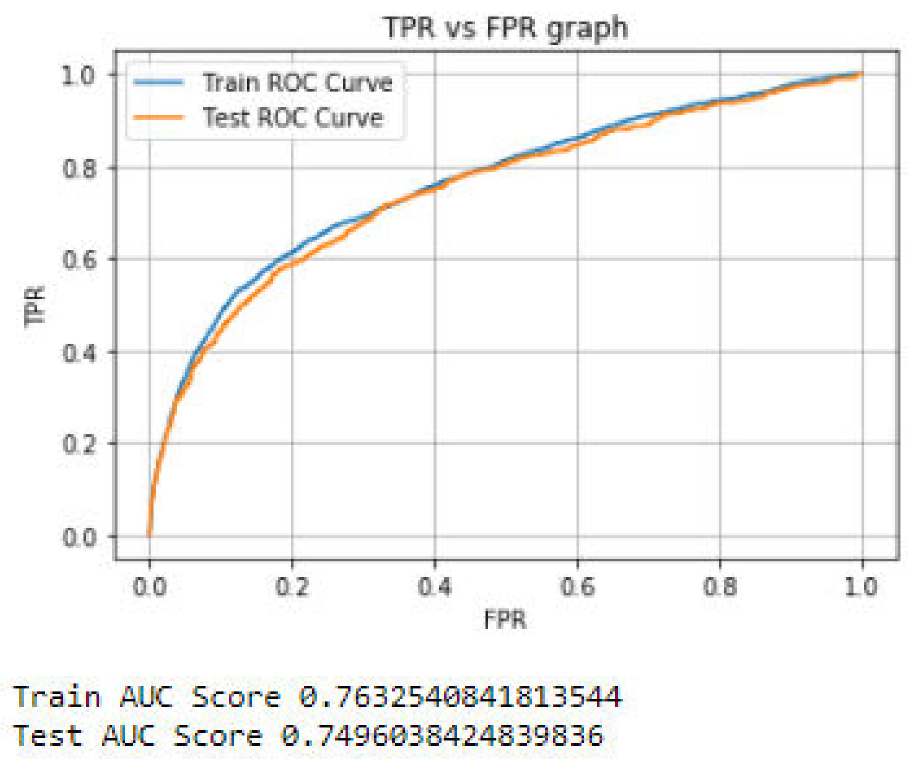
- The **Train F1-score** is **0.6119** and the **Test F1-score** is **0.6149** for the SGD Classifier with log-loss.
- This is also indicating a slightly good performance.Now let's work with the Linear SVM machine learning model.



```
Train F1-Score: 0.6119343567504646
Test F1-Score: 0.6148517115679384
```

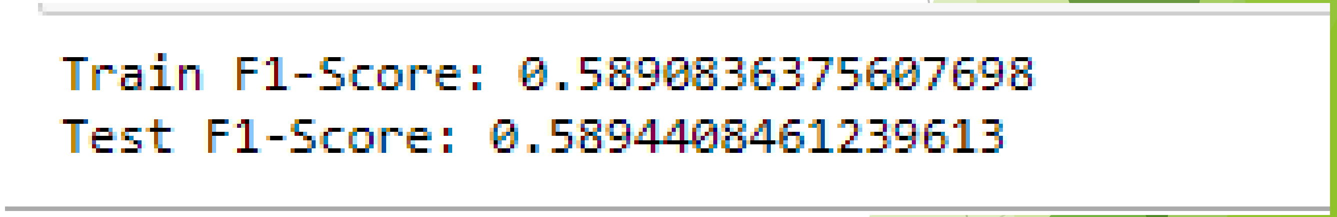
5.Linear SVM

- We get a **Train AUC score** of **0.7632** and a **Test AUC score** of **0.7496.**
- This indicates a good performance.Now let's look at the F1-score.



```
Train AUC Score 0.7632540841813544
Test AUC Score 0.7496038424839836
```

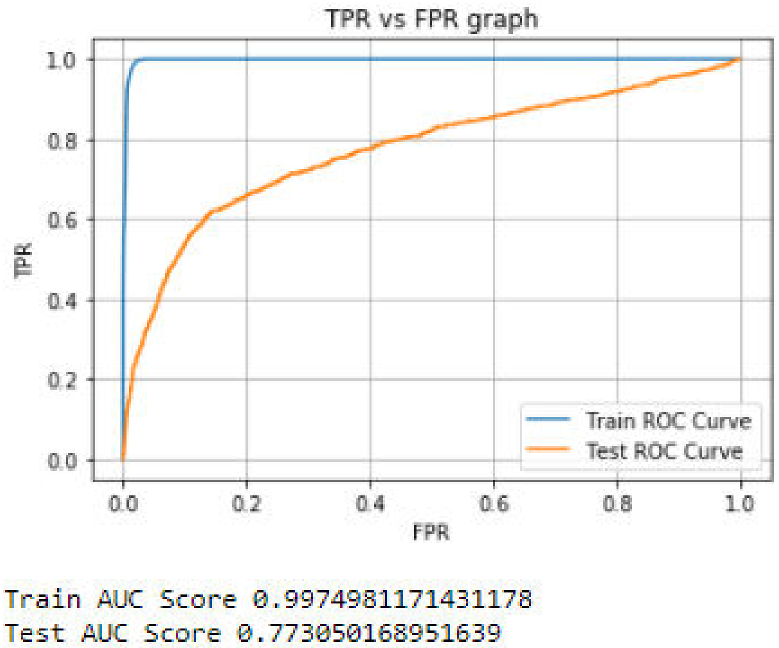
- We get a **Train F1-score** of **0.5891** and a **Test F1-score** of **0.5894.**
- This is not a bad.Now let's work with **Random Forest model.**



```
Train F1-Score: 0.5890836375607698
Test F1-Score: 0.589440846123961З
```

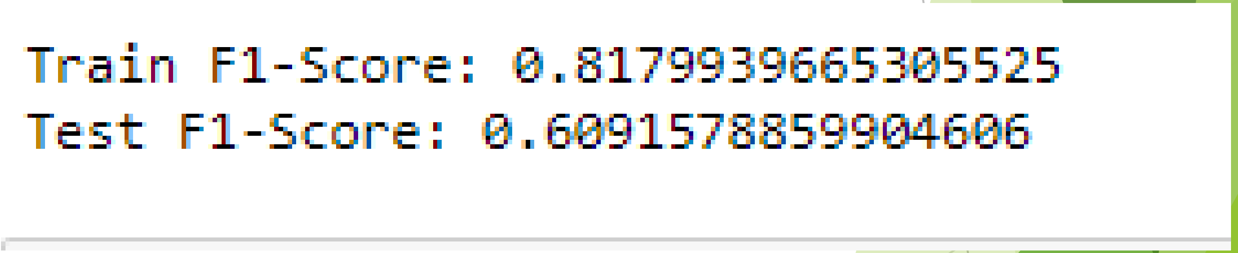
6.Random Forest

- Using the Random Forest model, we get a Train AUC score of 0.9975 and a Test AUC score of 0.7730.
- This indicates a very good performance.Now let's look at the F1-score.



```
Train AUC Score 0.9974981171431178
Test AUC Score 0.773050168951639
```

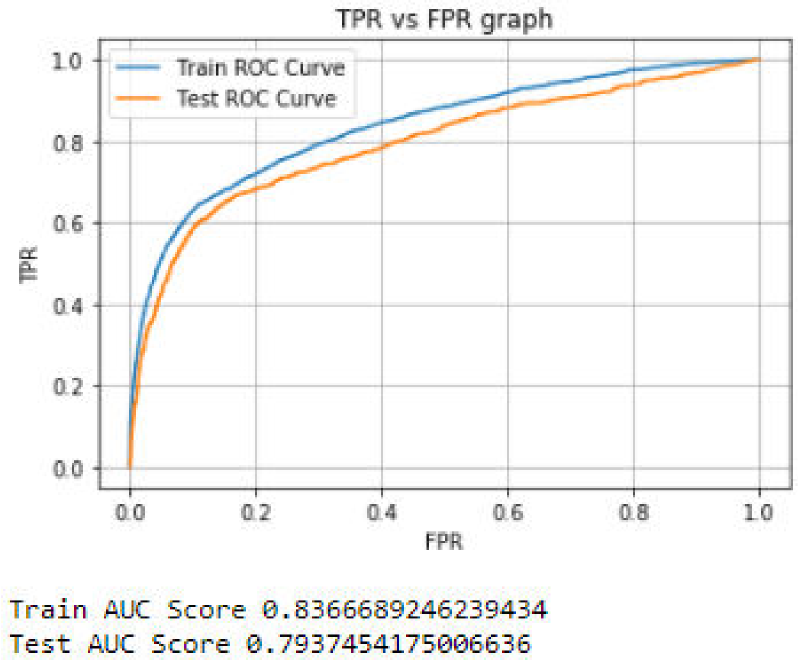
- We get a **Train F1-score** of **0.8180** and the **Test F1-score** of **0.6092.**
- This indicates a good performance for the Train F1-score and a slight good performance for the Test F1-score.Now let's work with the Xgboost machine learning model.



```
Train F1-Score: 0.8179939665305525
Test F1-Score: 0.6091578859904606
```

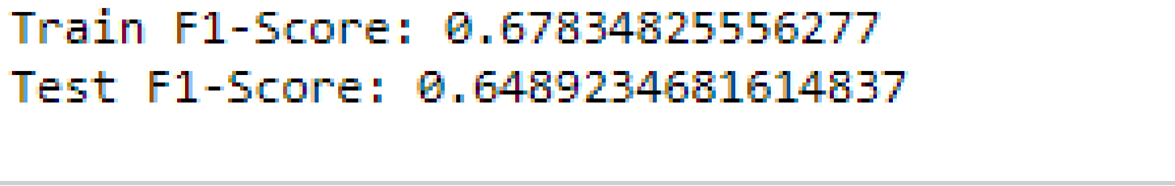
7.XGBoost Classifier

- The **Train AUC score** is **0.8367** and the **Test AUC score** is **0.7937.**
- This indicates a very good performance.Now let's look at the F1-score.



```
Train AUC Score 0.8366689246239434
Test AUC Score 0.7937454175006636
```

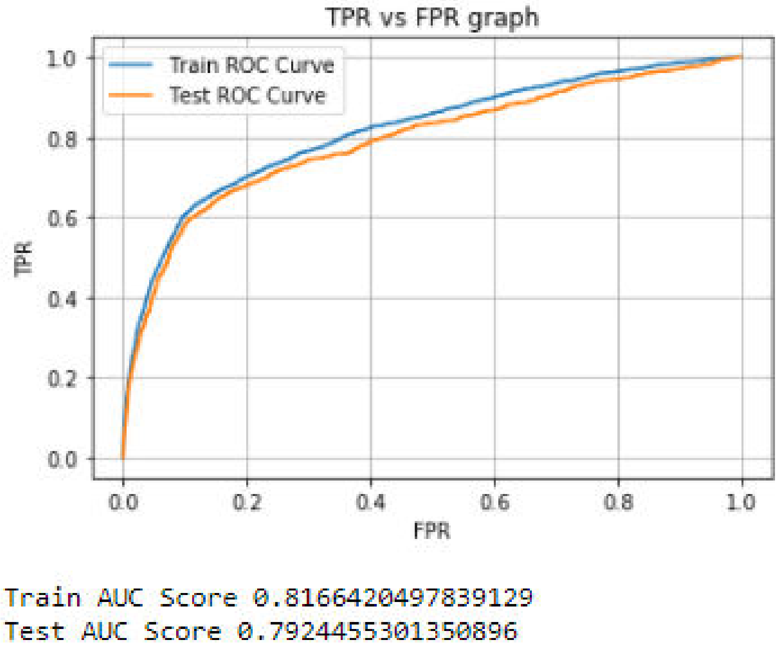
- We get a **Train F1-score** of **0.6783** and a **Test F1-score** of **0.6489.**
- This indicates a slight good performance.Now let's look at the Adaboost machine learning model.



```
Train F1-Score: 0.67834825556277
Test F1-Score: 0.6489234681614837
```

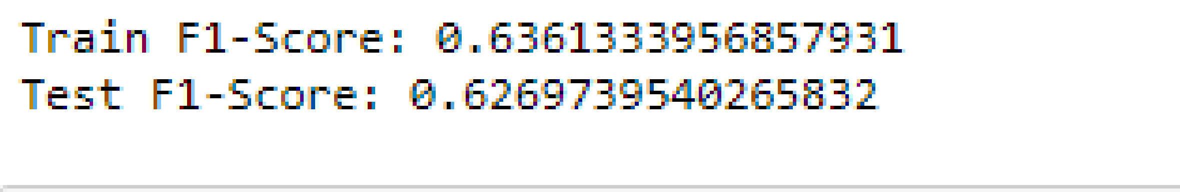
8.Adaboost

- We get a **Train AUC score** of **0.8166** and a **Test AUC score** of **0.7924.**
- This indicates a good performance.Now let's look at the F1-score.



```
Train AUC Score 0.8166420497839129
Test AUC Score 0.7924455301350896
```

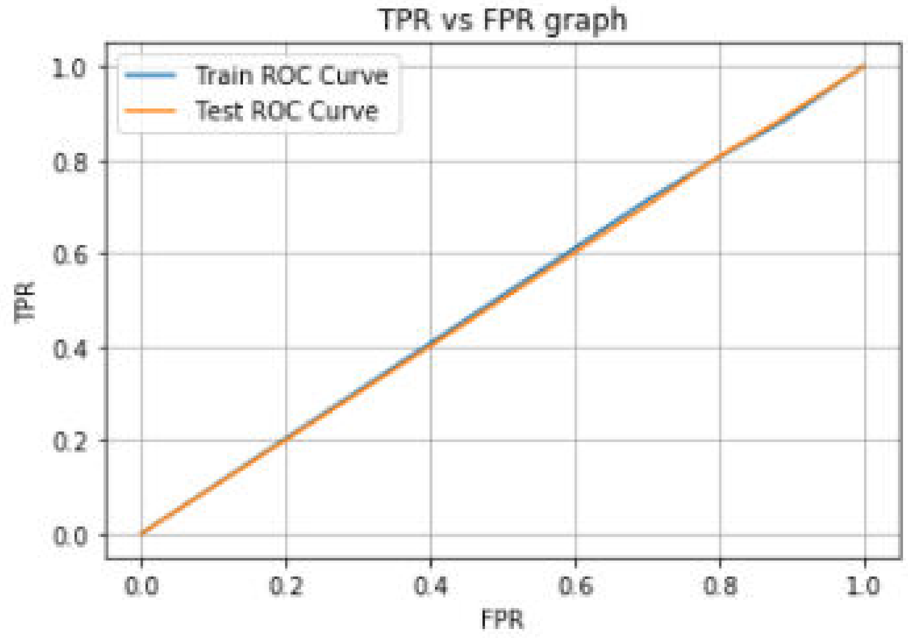
- The **Train F1-score** is **0.6361** and the **Test F1-score is 0.6270**.
- This is a slightly good performance.I’ve finished working with all the machine learning models encoded using One Hot Encoding method,now I will be working with the machine learning models encoded using Response Encoding method. I will be working with the same models in the preceding examples.



```
Train F1-Score: 0.6361333956857931
Test F1-Score: 0.6269739540265832
```

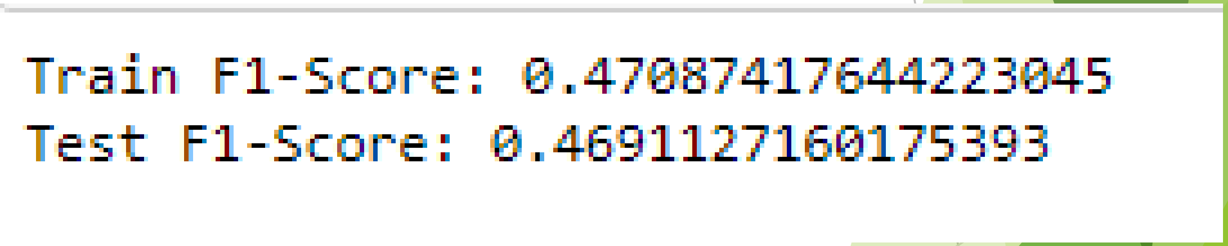
1. Dummy Classifier(Stratified) Response Encoding

- As we can see, the **Train AUC score** is **0.5050** and the **Test AUC score** is **0.5013.**
- This is not bad.Now let's look at the F1-score.



```
Train AUC Score 0.5049619456972307
Test AUC Score 0.5013427171441076
```

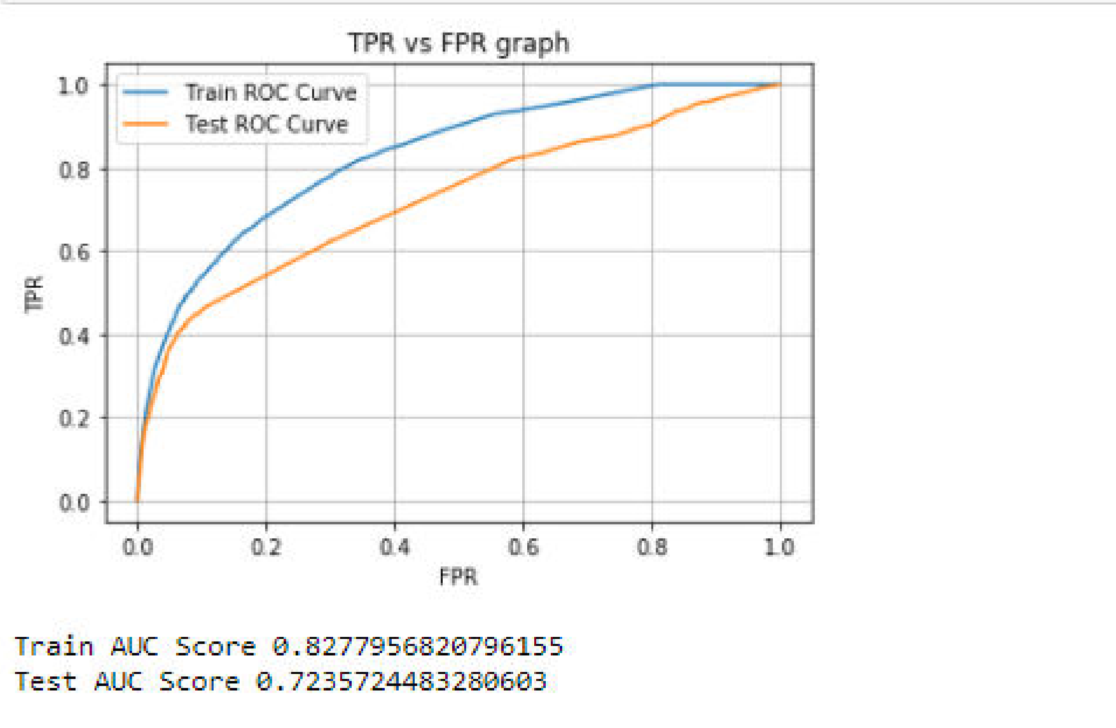
- We get a **Train F1-score** of **0.4709** and a **Test F1-score** of **0.4691.**
- This is a bad performance.Now let's work with K-NN model.



```
Train F1-Score: 0.47087417644223045
Test F1-Score: 0.4691127160175393
```

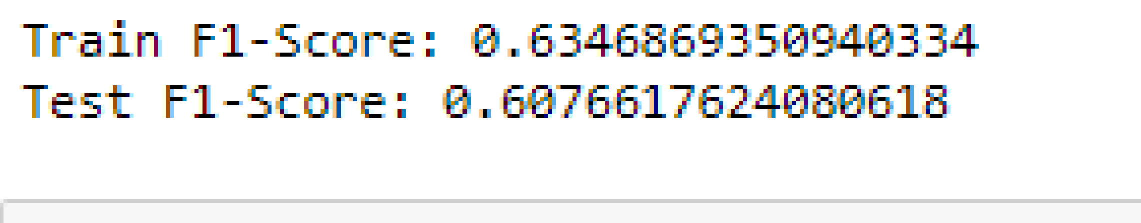
2.K-Nearest Neighbour

- Using the K-Nearest Neighbour algorithm,we get a **Train AUC score** of **0.8278** and a **Test AUC score** of **0.7236.**
- This indicates a good performance.Now let's look at the F1-score.



```
Train AUC Score 0.8277956820796155
Test AUC Score 0.7235724483280603
```

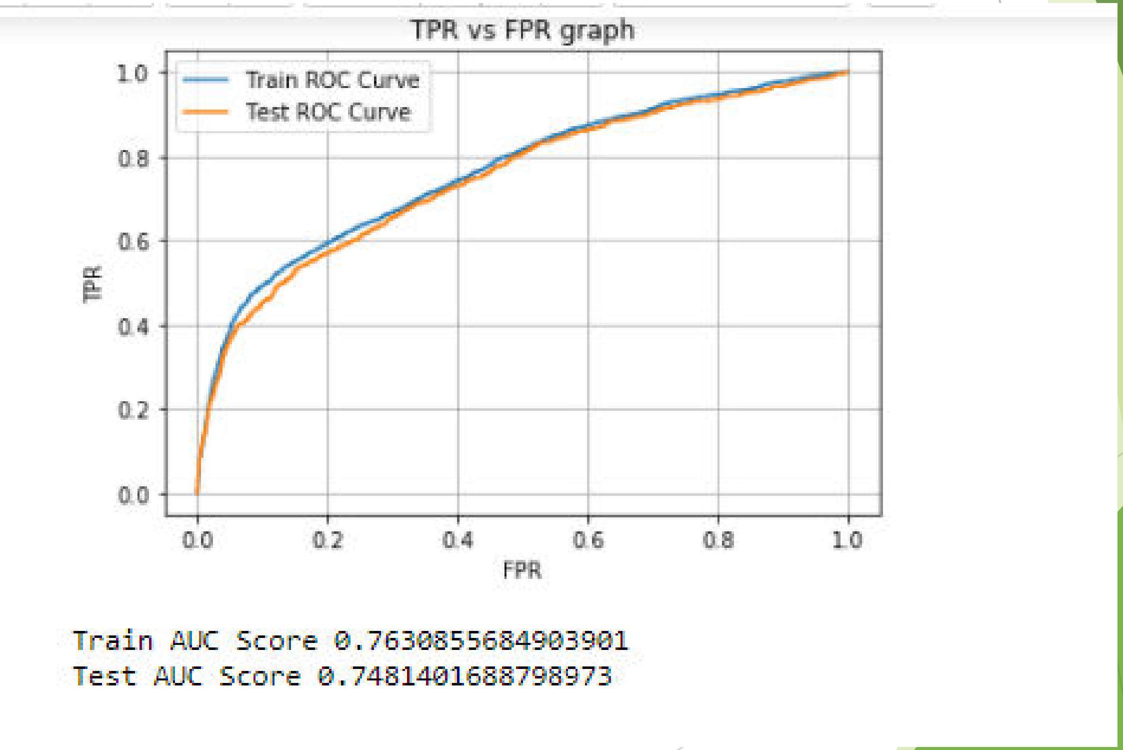
- The **Train F1-score** is **0.6347** and the **Test F1-score** is **0.6077**.
- This indicates a slightly good performance.Now let's work with Logistic Regression model.



```
Train F1-Score: 0.6346869350940334
Test F1-Score: 0.6076617624080618
```

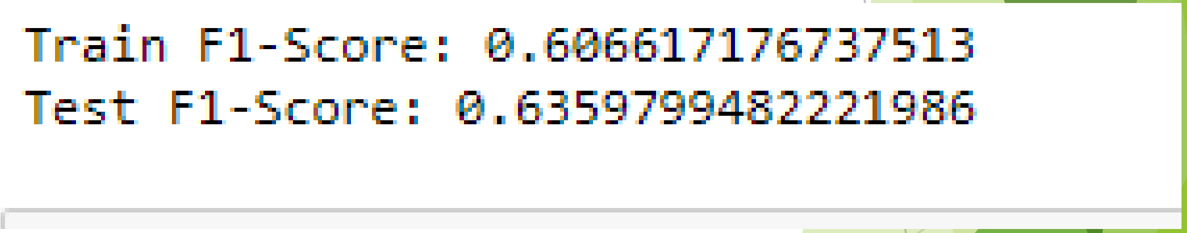
3.Logistic Regression

- We get a **Train AUC score** of **0.7630** and a **Test AUC score** of **0.7481.**
- This indicates a good performance.Now let's look at the F1-score.



```
Train AUC Score 0.7630855684903901
Test AUC Score 0.7481401688798973
```

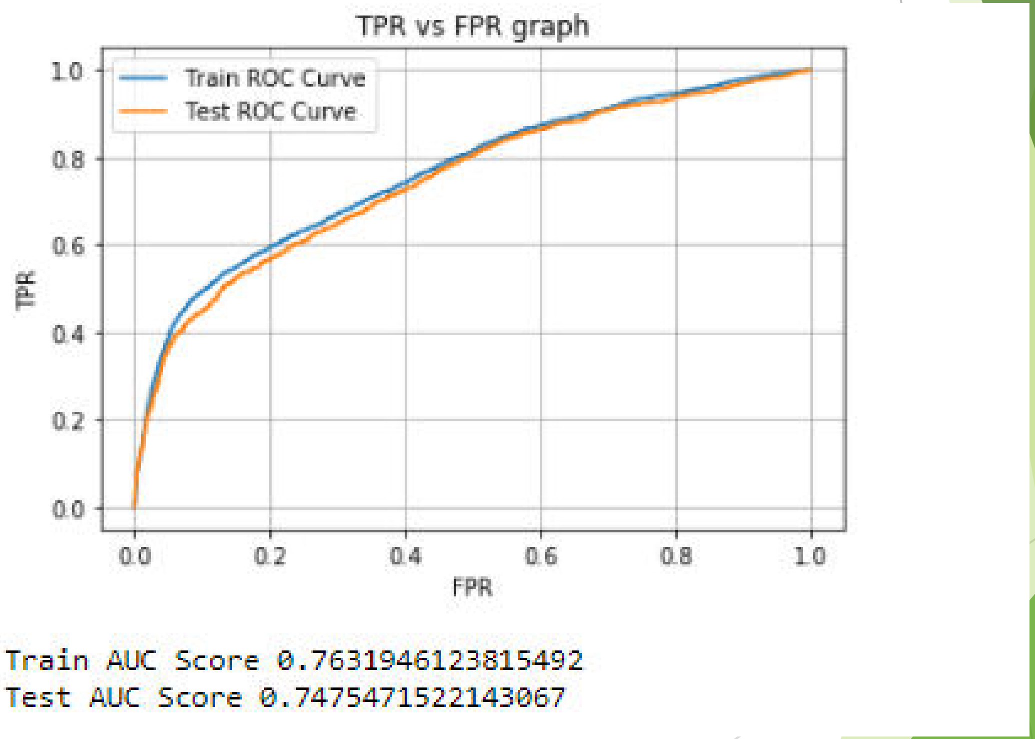
- The **Train F1-score** is **0.6066** and the **Test F1-score** is **0.6360.**Now let's work with the SGD with log-loss.



```
Train F1-Score: 0.606617176737513
Test F1-Score: 0.6359799482221986
```

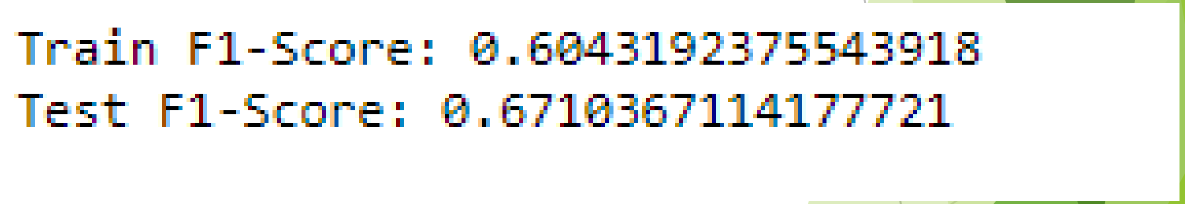
4.SGD with log-loss

- We get a **Train AUC score** of **0.7631** and a **Test AUC score** of **0.7475.**
- This indicates a good performance.
- Now let's look at the F1-score.



```
Train AUC Score 0.7631946123815492
Test AUC Score 0.7475471522143067
```

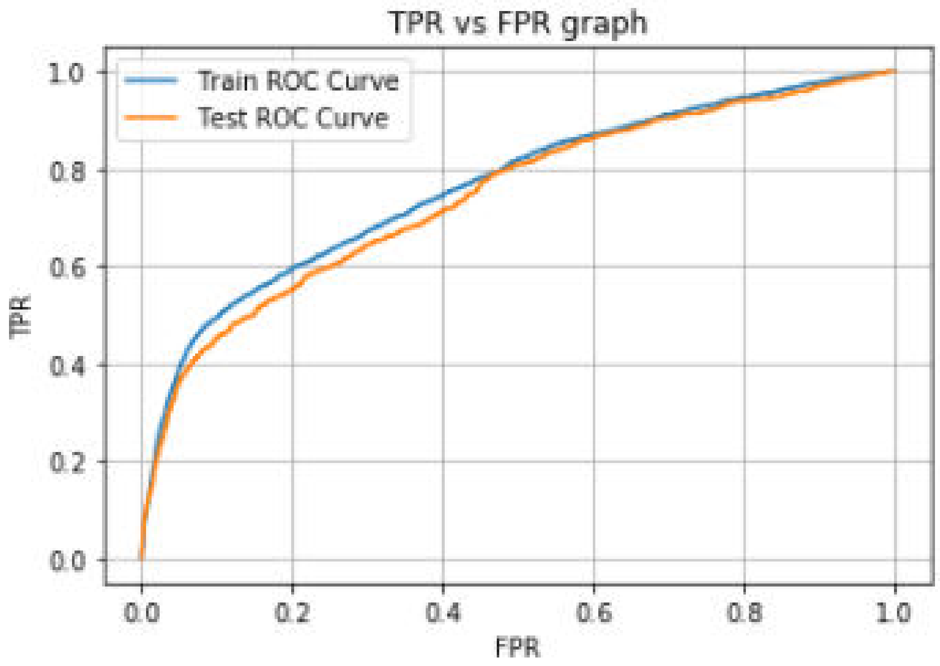
- The **Train F1-score** is **0.6043** and the **Test F1-score** is **0.6710**.Now let's work with Linear SVM model.



```
Train F1-Score: 0.6043192375543918
Test F1-Score: 0.6710367114177721
```

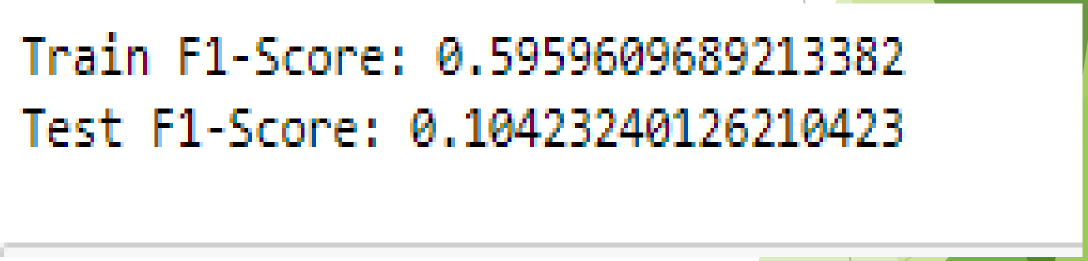
5.Linear SVM

- Using the Linear SVM model we get a **Train AUC score** of **0.7633** and a **Test AUC score** of **0.7436.**
- This indicates a good performance.Now let’s look at the F1-score.



```
Train AUC Score 0.7633221612255043
Test AUC Score 0.7435761071533622
```

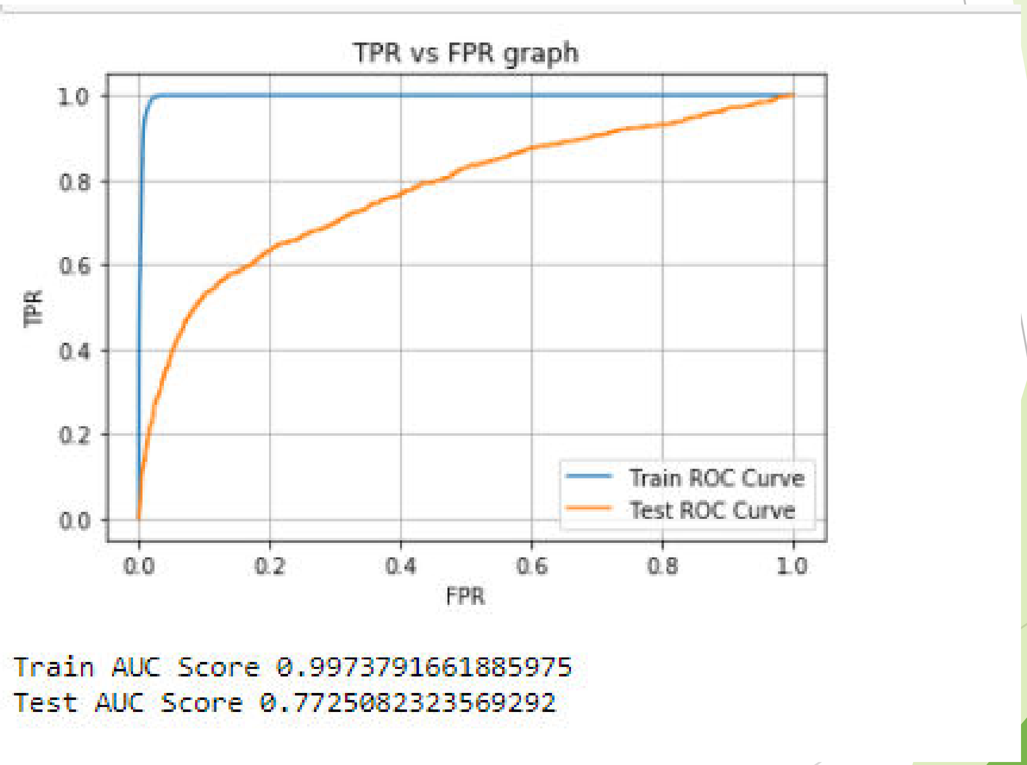
- The **Train F1-score** is **0.5960** and the **Test F1-score** is **0.1042.**
- The Test F1-score is very bad compared with the Test F1-score of all the other models.Now let’s work with Random Forest model.



```
Train F1-Score: 0.5959609689213382
Test F1-Score: 0.10423240126210423
```

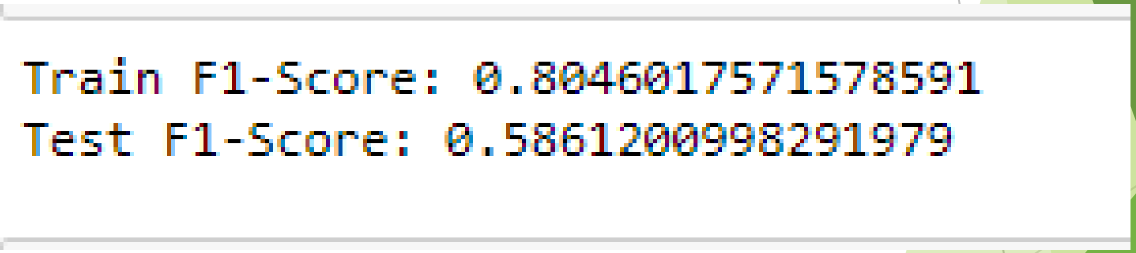
6.Random Forest

- The **Train AUC score** is **0.9974** and the **Test AUC score** is **0.7725.**
- This indicates a very good performance.Now let’s look at the F1-score.



```
Train AUC Score 0.9973791661885975
Test AUC Score 0.7725082323569292
```

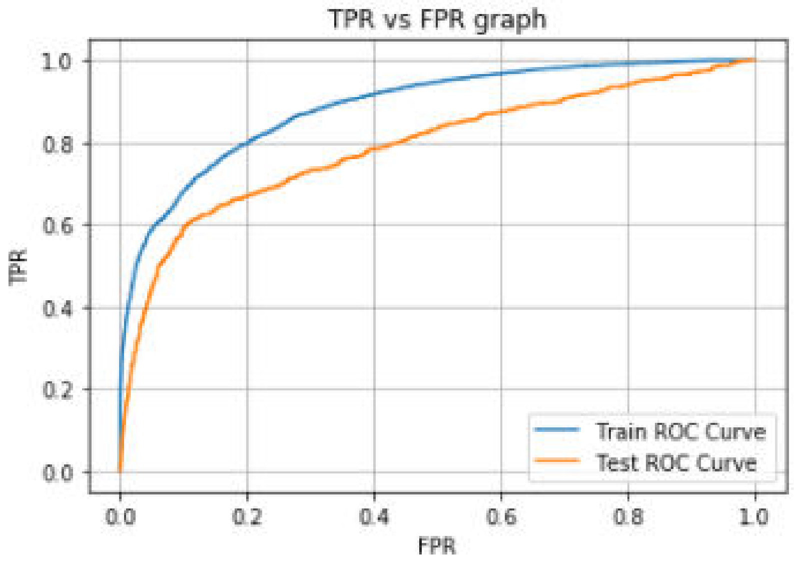
- We get a **Train F1-score** of **0.8046** and a **Test F1-score** of **0.5861.**
- Now let's work with the Xgboost model.



```
Train F1-Score: 0.8046017571578591
Test F1-Score: 0.5861200998291979
```

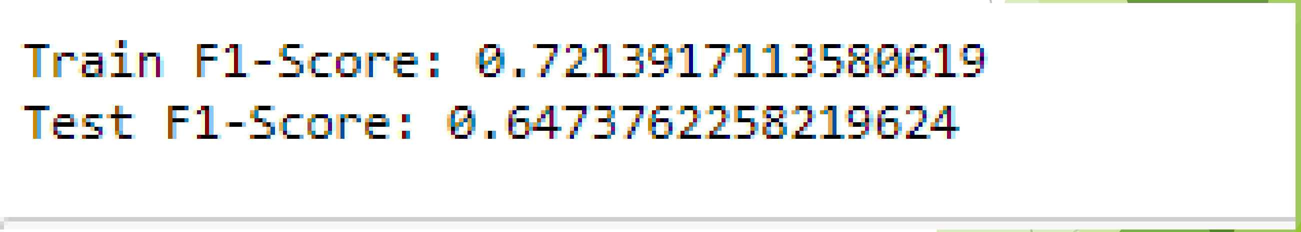
7.Xgboost

- We get a **Train AUC score** of **0.8877** and a **Test AUC score** of **0.7902.**
- This indicates a very good performance.Now let’s look at the F1-score.



```
Train AUC Score 0.8877379313286792
Test AUC Score 0.7902341648193187
```

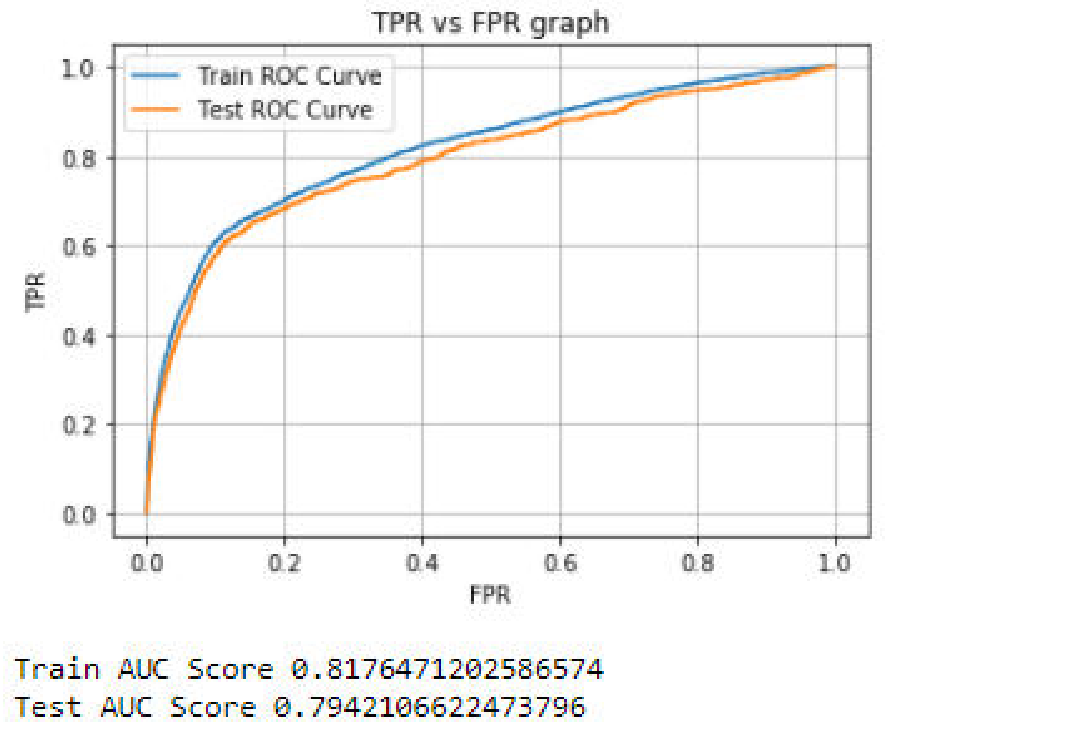
- The **Train F1-score** is **0.7213** and the **Test F1-score** is **0.6473.**
- This indicates a slight good performance.Now let's work with the last machine learning model,Adaboost.



```
Train F1-Score: 0.7213917113580619
Test F1-Score: 0.6473762258219624
```

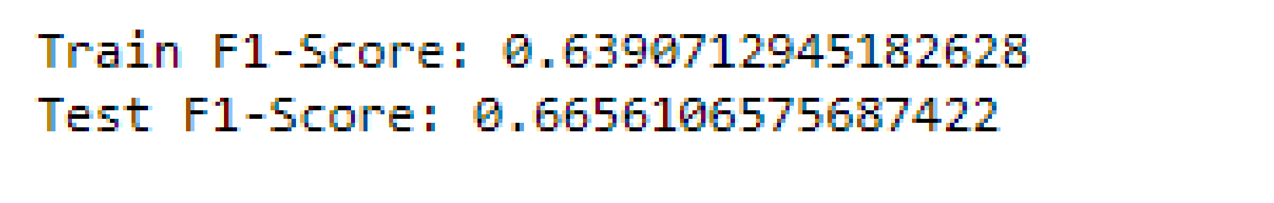
8. Adaboost

- Using the Adaboost model, we get a **Train AUC score** of **0.8176** and a **Test AUC score of 0.7942.**
- This indicates a very good performance.Now let's look at the F1-score.



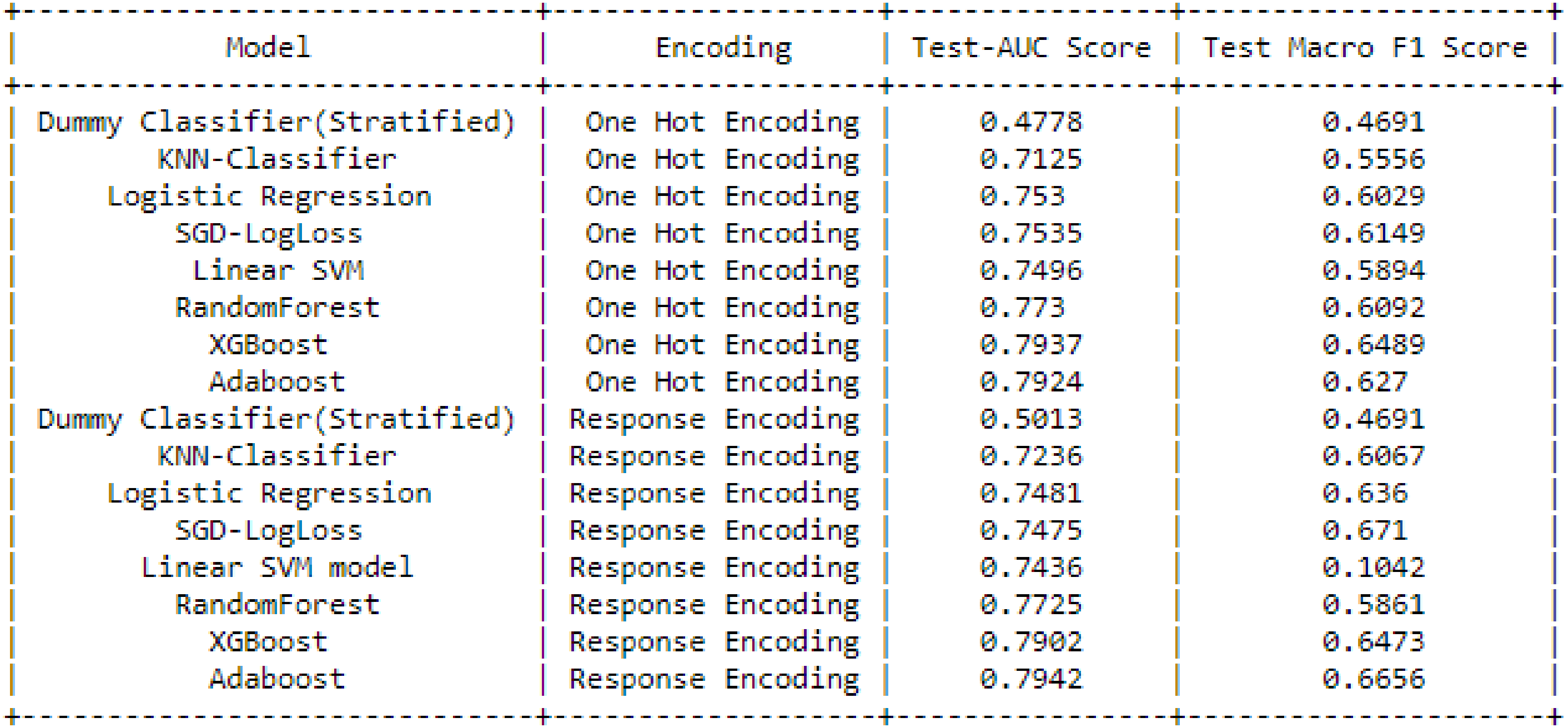
```
Train AUC Score 0.8176471202586574
Test AUC Score 0.7942106622473796
```

- The **Train F1-score** is **0.6391** and the **Test F1-score** is **0.6656.**
- This indicates a slightly good performance.Now let's select the best or champion model.



```
Train F1-Score: 0.6390712945182628
Test F1-Score: 0.6656106575687422
```

• Model Selection



```
+------------------------------+-------------------+----------------+---------------------+
|            Model             |     Encoding      | Test-AUC Score | Test Macro F1 Score |
+------------------------------+-------------------+----------------+---------------------+
| Dummy Classifier(Stratified) |  One Hot Encoding |     0.4778     |        0.4691       |
|        KNN-Classifier        |  One Hot Encoding |     0.7125     |        0.5556       |
|     Logistic Regression      |  One Hot Encoding |     0.753      |        0.6029       |
|         SGD-LogLoss          |  One Hot Encoding |     0.7535     |        0.6149       |
|          Linear SVM          |  One Hot Encoding |     0.7496     |        0.5894       |
|         RandomForest         |  One Hot Encoding |     0.773      |        0.6092       |
|           XGBoost            |  One Hot Encoding |     0.7937     |        0.6489       |
|           Adaboost           |  One Hot Encoding |     0.7924     |        0.627        |
| Dummy Classifier(Stratified) | Response Encoding |     0.5013     |        0.4691       |
|        KNN-Classifier        | Response Encoding |     0.7236     |        0.6067       |
|     Logistic Regression      | Response Encoding |     0.7481     |        0.636        |
|         SGD-LogLoss          | Response Encoding |     0.7475     |        0.671        |
|       Linear SVM model       | Response Encoding |     0.7436     |        0.1042       |
|         RandomForest         | Response Encoding |     0.7725     |        0.5861       |
|           XGBoost            | Response Encoding |     0.7902     |        0.6473       |
|           Adaboost           | Response Encoding |     0.7942     |        0.6656       |
+------------------------------+-------------------+----------------+---------------------+
```

Let’s compare our Models and check which model is performing the best.

- From the above we can see that model with **Adaboost** with response encoding has **best Test-AUC score** of **0.7942**

and **best F1-score of 0.6656** among other models.

- We would be saving xgboost using joblib library to predict the data.
- Therefore, we would require this model for deployment.

Thank You

Data Glacier

Your Deep Learning Partner