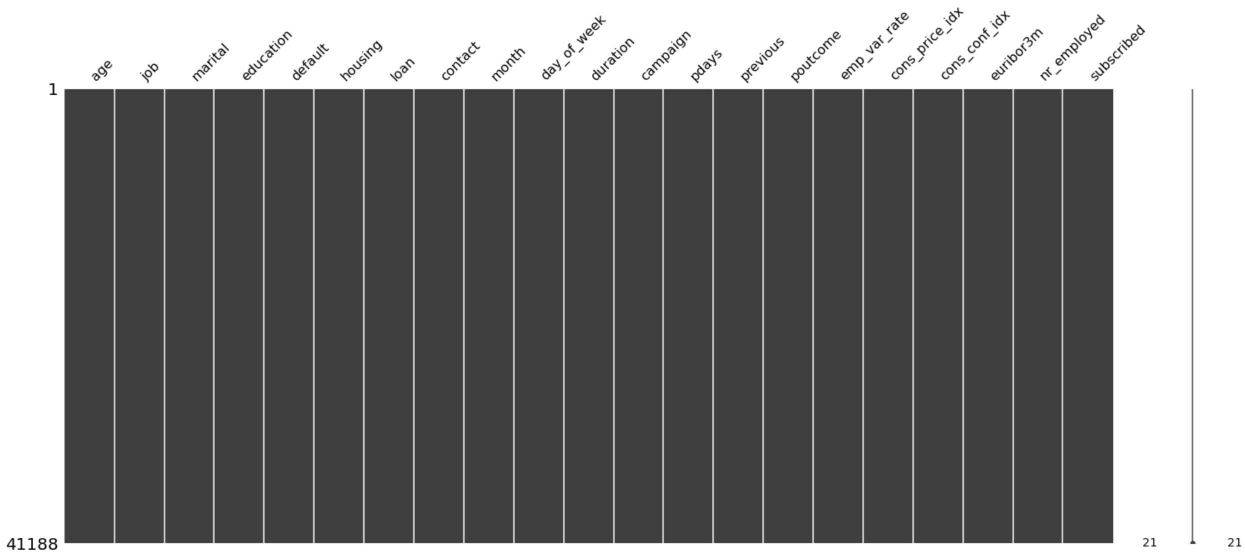
Bank marketing: Subscription prediction with Support Vector Classifier

Table of Contents

Exploratory data analysis

Visualizing missing values

We have noticed that although there are no missing values - for many of the categorical values these missing values were replaced by a value called ‘unknown’ which means that the particular attribute information for that record is unavailable. This informs about the preliminary preprocessing performed on the dataset.



Identifying data types & duplicate records

There are a total of 21 variables. Out of which, 11 are categorical values and 10 are continuous variables. Even though the continuous variables, 5 are integers and 5 are floats with decimal values.

There are around 12 records that are duplicates and we drop these records as they deem to be rudimentary to occur twice in the dataset.

Categorical data descriptive statistics

	job	marital	education	default	housing	loan	contact	month	day_of_week	poutcome	subscribed
count	41188	41188	41188	41188	41188	41188	41188	41188	41188	41188	41188
unique	12	4	8	3	3	3	2	10	5	3	2
top	admin.	married	university.ddegree	no	yes	no	cellular	may	Thu	nonexistentt	no
freq	10422	24928	12168	32588	21576	33950	26144	13769	8623	35563	36548
% of freq	25%	61%	30%	79%	52%	82%	63%	33%	21%	86%	89%

Like observed in the missing value plot above, there seems to be no value that is missing in the dataset. And the stats show the number of unique values for each categorical value and the value appearing highest times along with the count of the value. It gives a brief idea about the popular values in the dataset. A simple calculation of freq/count in the above table can give us insight into highly dominating categorical values in each column. For example, setting a threshold of 50% - we can see that 'married' in marital, 'no' in default, 'yes' in housing, 'no in loan, 'cellular' in contact, 'nonexistent' in poutcome and 'no' in subscribed are values that appear more than 50% of the time in their respective column.

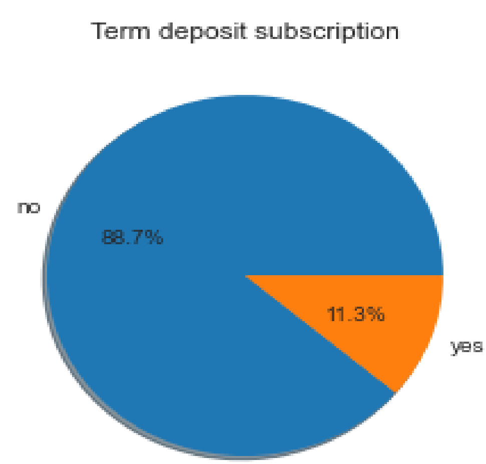
Continuous data descriptive statistics

	age	duration	campaign	pdays	previous	emp_var_rate	cons_price_idx	cons_conf_idx	euribor3m	nr_employed
count	41188	41188	41188	41188	41188	41188	41188	41188	41188	41188
mean	40.02	258.29	2.57	962.48	0.17	0.08	93.58	-40.50	3.62	5167.04
std	10.42	259.28	2.77	186.91	0.49	1.57	0.58	4.63	1.73	72.25
min	17	0	1	0	0	-3.4	92.201	-50.8	0.634	4963.6
25%	32	102	1	999	0	-1.8	93.075	-42.7	1.344	5099.1
50%	38	180	2	999	0	1.1	93.749	-41.8	4.857	5191
75%	47	319	3	999	0	1.4	93.994	-36.4	4.961	5228.1
max	98	4918	56	999	7	1.4	94.767	-26.9	5.045	5228.1

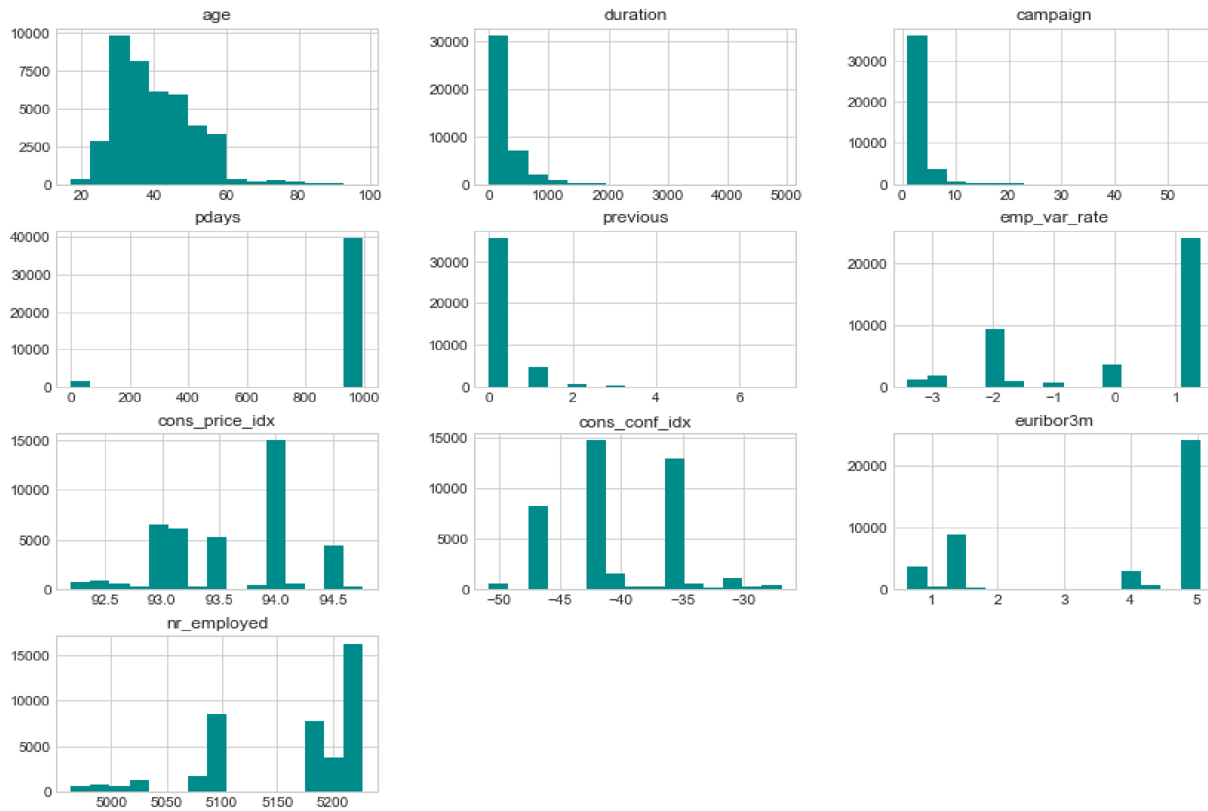
The above statistics for the continuous values give us a fair idea on their data distribution. And it's quite evident that only age has a distribution close to gaussian. We will further very this insight through histogram plots further.

Target Variable distribution

Our dependent variable here is 'subscribed' - which explains given the independent variables such as demographic & financial information of a lead if the person has subscribed to a term deposit or not. Through the pie chart, we plotted we can see that around 88.7% of the people contacted as a part of marketing haven't subscribed to the term deposit and only 11.3% have gone ahead to make a subscription.

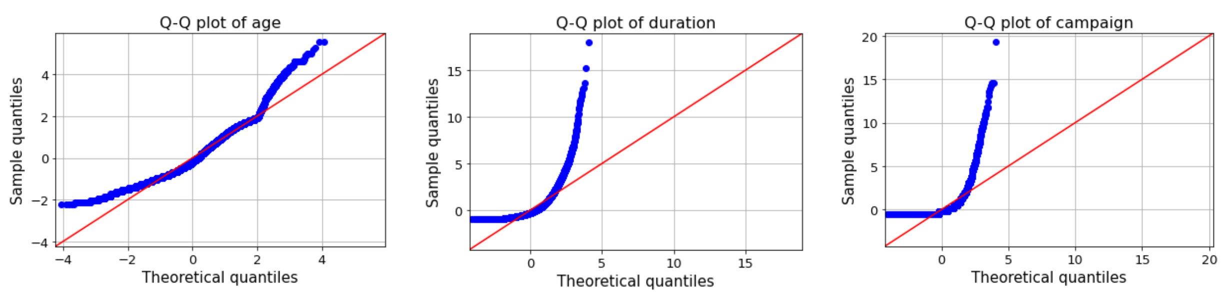


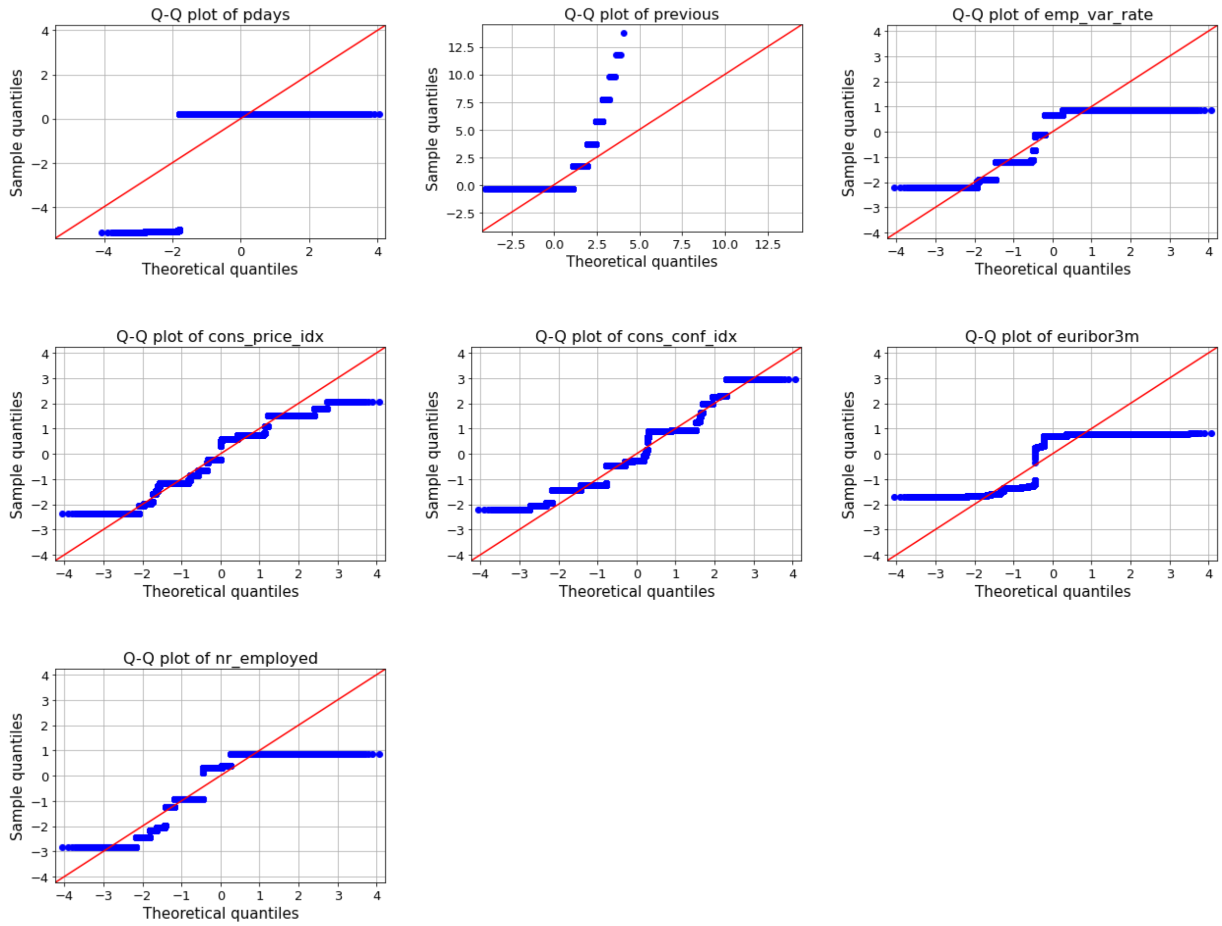
Continuous variables distribution plots



Aligning with our inference from the statistics above, the *age* variable has a close-to-normal distribution. While the other variables are either skewed distribution or bi-modal distribution. Further investigation into these variables will determine how we deal with these variables.

Q-Q plots for continuous variables





'age' has a close-to-normal plot with data at a peak in the middle. '*duration*', *'campaign'*, and '*previous*' seem to be rightly skewed. '*pdays*' is left skewed. *'emp_var_rate'*, *'cons_price_idx'*, *'cons_conf_idx'*, *'euribor3m'*, *and 'nr_employed'* have a bimodal distribution.

Categorical variables: values & distributions

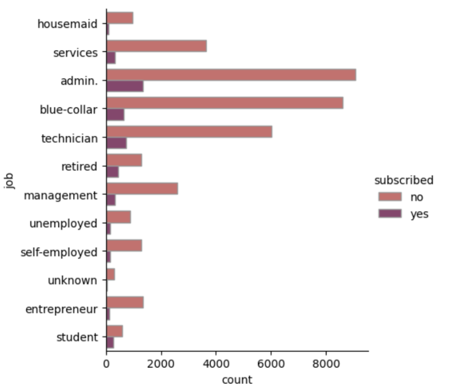
Categorical variable	Values	Count	% of Count
JOB	admin.	10422	25.30%
	blue-collar	9254	22.47%
	technician	6743	16.37%
	services	3969	9.64%
	management	2924	7.10%
	retired	1720	4.18%

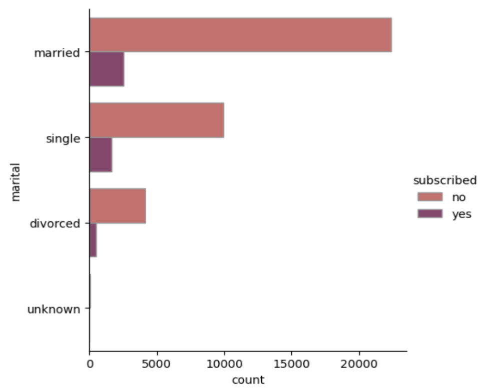
	entrepreneur	1456	3.54%
	self-employed	1421	3.45%
	housemaid	1060	2.57%
	unemployed	1014	2.46%
	student	875	2.12%
	unknown	330	0.80%
MARITAL	married	24928	60.52%
	single	11568	28.09%
	divorced	4612	11.20%
	unknown	80	0.19%
EDUCATION	university.degree	12168	29.54%
	high.school	9515	23.10%
	basic.9y	6045	14.68%
	professional.course	5243	12.73%
	basic.4y	4176	10.14%
	basic.6y	2292	5.56%
	unknown	1731	4.20%
	illiterate	18	0.04%
DEFAULT	no	32588	79.12%
	unknown	8597	20.87%
	yes	3	0.01%
HOUSING	yes	21576	52.38%
	no	18622	45.21%
	unknown	990	2.40%
LOAN	no	33950	82.43%
	yes	6248	15.17%
	unknown	990	2.40%
CONTACT	cellular	26144	63.47%
	telephone	15044	36.53%
MONTH	may	13769	33.43%

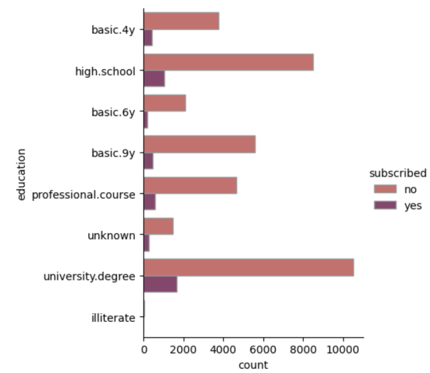
	Jul	7174	17.42%
	Aug	6178	15.00%
	Jun	5318	12.91%
	Nov	4101	9.96%
	Apr	2632	6.39%
	oct	718	1.74%
	sep	570	1.38%
	mar	546	1.33%
	dec	182	0.44%
DAY_OF_WEEK	thu	8623	20.94%
	mon	8514	20.67%
	wed	8134	19.75%
	tue	8090	19.64%
	fri	7827	19.00%
POUTCOME	nonexistent	35563	86.34%
	failure	4252	10.32%
	success	1373	3.33%
SUBSCRIBED	no	36548	88.73%
	yes	4640	11.27%

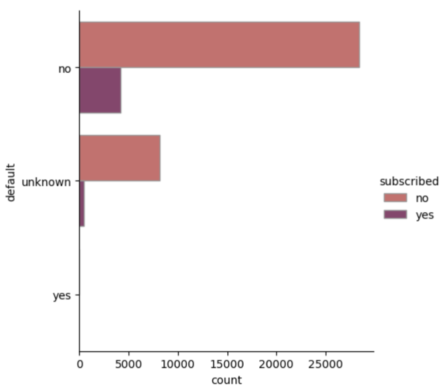
Marital variable has 60.52% of married leads and the others being single, divorced and unknown. *Education* has three values as basic.4y, basic.6y & basic.9y which could be combined into mid-school, hence we would be making this change during the feature cleaning. *Default* has 20.87% of values that are ‘unknown’, we would be making a decision on the column after looking at the correlation and t-test. Similarly, poutcome has 86.34% of values which says ‘nonexistent’, we would be making a decision on the column after looking at the correlation and t-test.

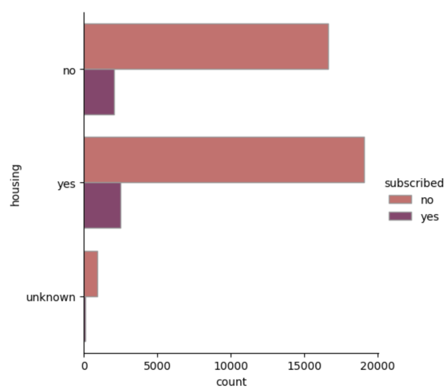
Plots

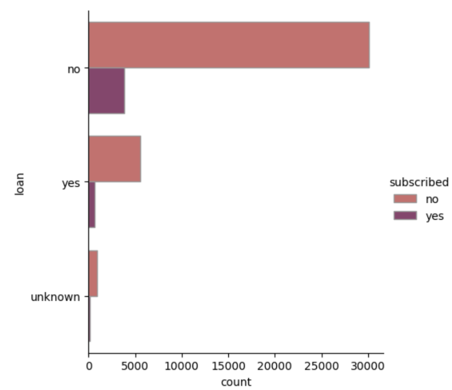

job
housemaid
services
admin.
blue-collar
technician
retired
management
unemployed
self-employed
unknown
entrepreneur
student
0
2000
4000
6000
8000
count
subscribed
no
yes

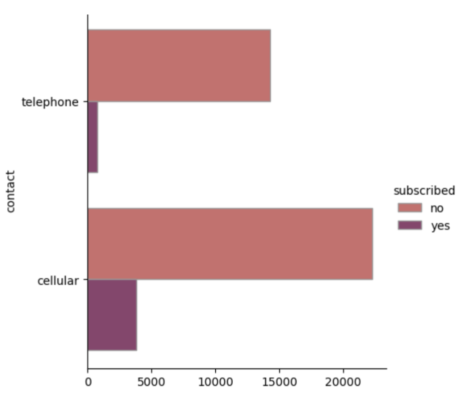

marital
married
single
divorced
unknown
0
5000
10000
15000
20000
count
subscribed
no
yes

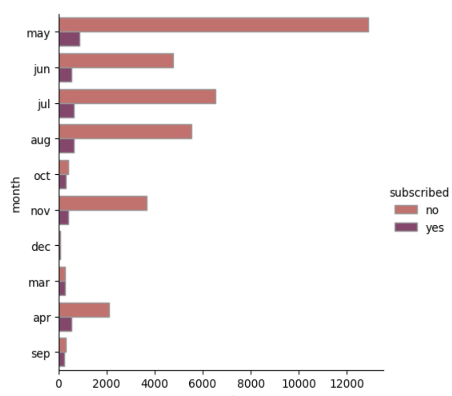

education
basic.4y
high.school
basic.6y
basic.9y
professional.course
unknown
university.degree
illiterate
0
2000
4000
6000
8000
10000
count
subscribed
no
yes

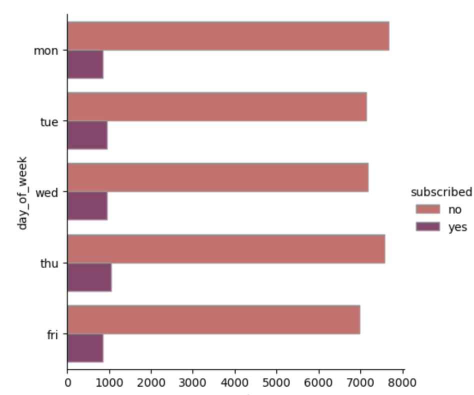

default
no
unknown
yes
0
5000
10000
15000
20000
25000
count
subscribed
no
yes


housing
no
yes
unknown
0
5000
10000
15000
20000
count
subscribed
no
yes


loan
no
yes
unknown
0
5000
10000
15000
20000
25000
30000
count
subscribed
no
yes


contact
telephone
cellular
0
5000
10000
15000
20000
count
subscribed
no
yes


month
may
jun
jul
aug
oct
nov
dec
mar
apr
sep
0
2000
4000
6000
8000
10000
12000
count
subscribed
no
yes


day_of_week
mon
tue
wed
thu
fri
0
1000
2000
3000
4000
5000
6000
7000
8000
count
subscribed
no
yes

Data cleaning & feature selection

Keeping the granularity of the field in mind, *Education* has three values as basic.4y, basic.6y & basic.9y which are combined into one single value called *mid-school*.

Transforming categorical data into numeric values

Most machine learning models accept only numeric values. Thus we identify and transform certain features into numeric values. The data attribute is categorical if it represents a discrete value that belongs to a specific finite set of categories or classes using a Label Encoder.

Normalization of features

For the next step, we standardize the features to bring coefficients to a similar scale which enables us to compare attributes with each other. This is done by scaling the mean of features to 0 and the standard deviation to 1. We use maximum absolute scaling to normalize the data.

Correlation matrix for feature selection

A correlation matrix is computed to check the relationship between the numerical features. We would be setting a threshold of 90% and drop the features that we think wouldn't affect the predictive performance of the model.

	age	**duration**	**campaign**	**pdays**	**previous**	**emp_var_rate**	**cons_price_idx**	**cons_conf_idx**	**euribor3m**	**nr_employed**
age	100.00%	0.09%	0.46%	3.44%	2.44%	0.04%	0.09%	12.94%	1.08%	1.77%
duration	0.09%	100.00%	7.17%	4.76%	2.06%	2.80%	0.53%	0.82%	3.29%	4.47%
campaign	0.46%	7.17%	100.00%	5.26%	7.91%	15.08%	12.78%	1.37%	13.51%	14.41%
pdays	3.44%	4.76%	5.26%	100.00%	58.75%	27.10%	7.89%	9.13%	29.69%	37.26%
previous	2.44%	2.06%	7.91%	58.75%	100.00%	42.05%	20.31%	5.09%	45.45%	50.13%
emp_var_rate	0.04%	2.80%	15.08%	27.10%	42.05%	100.00%	77.53%	19.60%	97.22%	90.70%
cons_price_idx	0.09%	0.53%	12.78%	7.89%	20.31%	77.53%	100.00%	5.90%	68.82%	52.20%
cons_conf_idx	12.94%	0.82%	1.37%	9.13%	5.09%	19.60%	5.90%	100.00%	27.77%	10.05%
euribor3m	1.08%	3.29%	13.51%	29.69%	45.45%	97.22%	68.82%	27.77%	100.00%	94.52%
nr_employed	1.77%	4.47%	14.41%	37.26%	50.13%	90.70%	52.20%	10.05%	94.52%	100.00%

We can notice a high correlation among euribor3m: emp_var_rate, nr_employed: emp_var_rate & euribor3m: nr_employed. Hence, we can drop two columns i.e euribor3m and nr_employed which have a translative high correlation.

Balancing the dataset

We had was highly unbalanced on the target variable split, with an 89:11 ratio. Hence, we balance it to have an equal split of categories in the target variable using a random state. After the balancing, we get a total number of records of 9280.

Performing a t-test on features to check their significance

p-values helps us identify each column's impact on the prediction of the target variable. Hence, we have performed a t-test to compute the p-values for each variable. Based on the results and by setting a *threshold* of *00.5*, we have dropped the variables with a p-value less than *0.05*. Hence dropping *housing, loan, month,* and *day_of_week* from the dataset. This forms to be the final dataset after cleaning that we would further use for data modeling.

Data modeling

Based on the size of the dataset and distribution of values, we have decided to go with a support vector classifier and fit a grid search approach on top of the model to identify the best values as parameters that give a better score.

Below are the results of an SVC fit with ‘poly’ as kernel:

	precision	*recall*	*f1-score*	*support*
0	0.87	0.82	0.84	931
1	0.83	0.87	0.85	925
accuracy			**0.85**	1856
macro avg	0.85	0.85	0.85	1856
weighted avg	0.85	0.85	0.85	1856

A grid search on the above model has resulted in a slight boost of a score of 2.8% (87.8%) with parameters as follows:

C=1000, gamma=0.1, kernel=poly;