Georgia State University

CSC4780/6780&DSCI4780 – Fundamentals of Data Science

Spring 2022

Project Progress Report

Bank Data Analysis

[AJKFDS]
Jagan Gedela
Sridutt Kalidindi
Aryan Shrestha
Caleb Tsai
Christopher Yoon

Table of Contents

1 Business Understanding

This project assists banks in determining the factors that influence whether a consumer will accept a bank's term deposit offer. The banks can use this information to identify consumers who are likely to accept the term deposit and lower the amount of effort required because they are likely to accept it. They can also provide extra incentives to less likely consumers to accept. The project's purpose is to find crucial attributes in the dataset that can help categorize clients based on whether or not they accept or reject the term deposit that has been provided to them.

1.1 Business Problem

Banks routinely contact many customers, either by phone or email, to offer term plans. The majority of these calls, it is thought, do not result in the anticipated outcome. As a result, banks' time and resources are spent in vain. We present a model that can predict which users are more likely to accept term policies based on some basic user information to address this problem. The model may also determine the optimal marketing plan for the next campaign.

1.2 Dataset

Data is gathered through a phone call campaign for term deposits. The campaign aims to give clients a term deposit while also recording other customer attributes. The information also includes the call's conclusion, such as whether the customer accepted or rejected the term deposit. Using this property, we can turn a real-world problem into a Supervised Machine Learning challenge.

The dataset has 21 attributes and 41,189 instances. This dataset has both continuous and categorical (ordinal and nominal) attributes, making it an excellent candidate for applying CRISP-Data Mining techniques. This dataset also allows for using several algorithms for addressing missing values and detecting outliers. Using the cleaned dataset, various data modeling techniques can be used to identify the link between characteristics and goal variables.

1.3 Proposed Analytics Solution

For this dataset, we will run through various classification techniques such as Logistic Regression, Random Forest, Support Vector Machine, and KNN. Prior to classification, however, the data will be cleaned and analyzed through the use of statistical analysis such as Exploratory Data Analysis, Principal Component Analysis, Factor Analysis, and Correlations. Relating to this is the subject of identifying missing or outlier values, which may skew data or disrupt the accuracy of our analysis, so to pre-process our data we should get rid of these values and replace them accordingly with methods such as mean and mode replacement. After the initial cleaning of the data, normalization and transformation occur to ensure that the data is finally ready for our classification.

2 Data Exploration and Preprocessing

For data visualization techniques, exploratory data analysis (EDA) has been used to analyze and examine data sets and summarize their key features. It also helps determine if the statistical approaches considered for data analysis are appropriate.

2.1 Data Quality Report

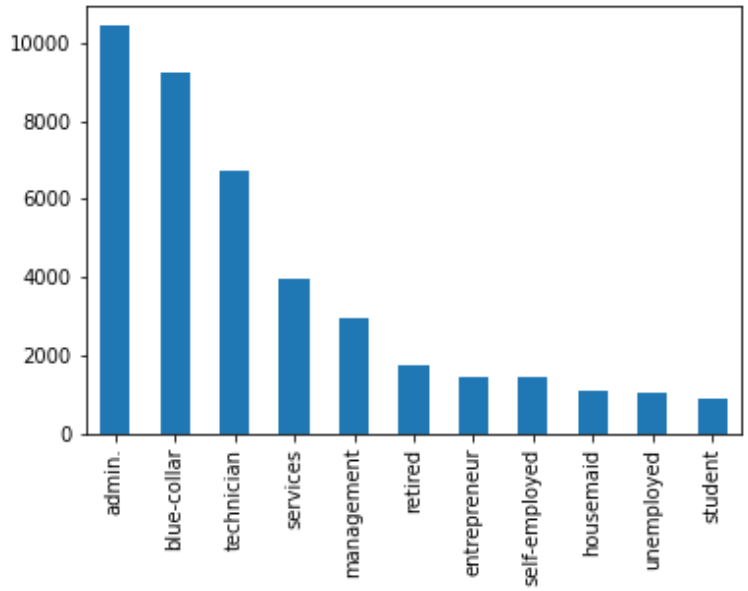
From the Data Quality Reports we can describe the state of the features given by the dataset and determine whether or not they are fine as-is or if they should be split, transformed, and/or normalized. Generally the effect that missing values have on the data is inconsequential, as most features do not have more than 10% missing values. However, the specific feature pdays includes more than 96% of its values as missing values, so in this case we might drop it as a feature entirely or change the feature from continuous to categorical.

Table 1. Data Quality Report for Categorical Features

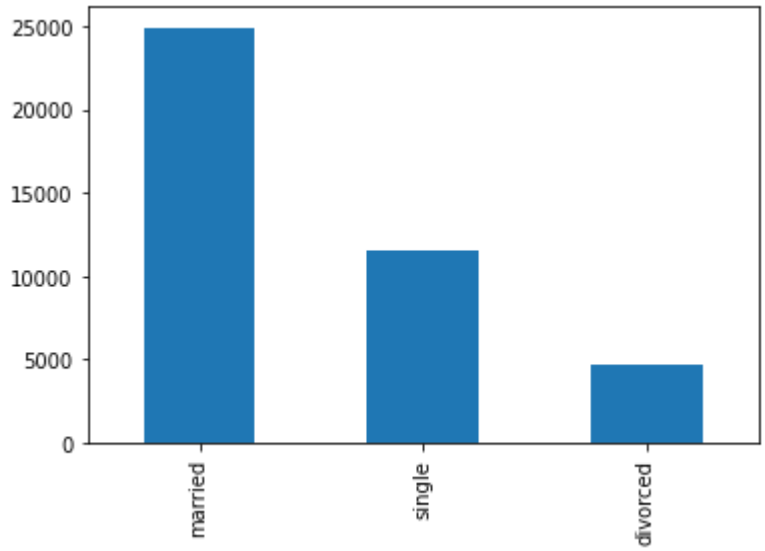
Feature	% of Missing	2nd Mode	2nd Mode Freq	2nd Mode Perc	Card	Count	Mode	Mode %	Mode Freq
job	0.80	blue-collar	9253.0	22.47	12.0	41176.0	admin.	25.30	10419.0
marital	0.19	single	11564.0	28.08	4.0	41176.0	married	60.52	24921.0
education	4.20	high.school	9512.0	23.10	8.0	41176.0	university.degree	29.54	12164.0
default	20.88	yes	3.0	0.01	3.0	41176.0	no	79.12	32577.0
housing	2.40	no	18615.0	45.21	3.0	41176.0	yes	52.39	21571.0
loan	2.40	yes	6248.0	15.17	3.0	41176.0	no	82.42	33938.0
contact	0.00	telephone	15041.0	36.53	2.0	41176.0	cellular	63.47	26135.0
month	0.00	jul	7169.0	17.41	10.0	41176.0	may	33.43	13767.0
day_of_week	0.00	mon	8512.0	20.67	5.0	41176.0	thu	20.93	8618.0
poutcome	0.00	failure	4252.0	10.33	3.0	41176.0	nonexistent	86.34	35551.0
y	0.00	yes	4639.0	11.27	2.0	41176.0	no	88.73	36537.0

Figure 1. Visualizations of Categorical Features in Dataset

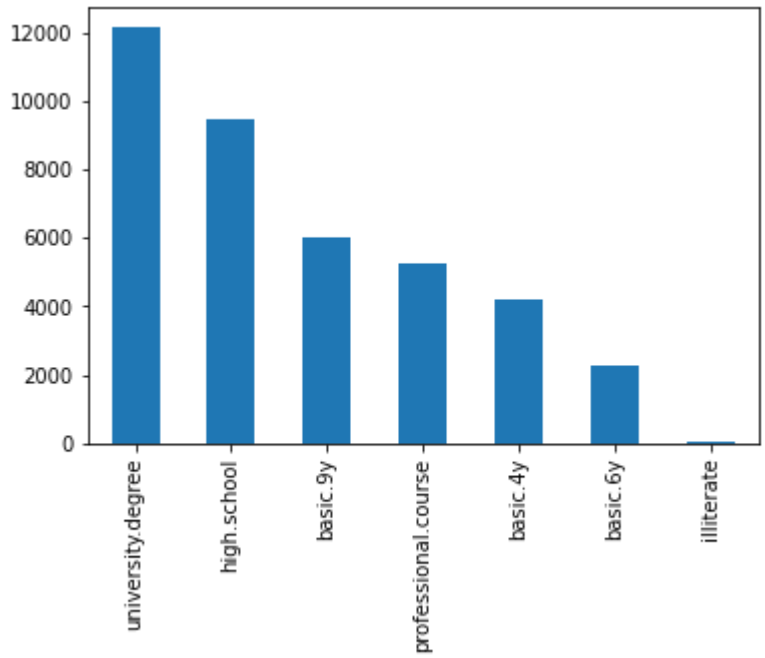
job



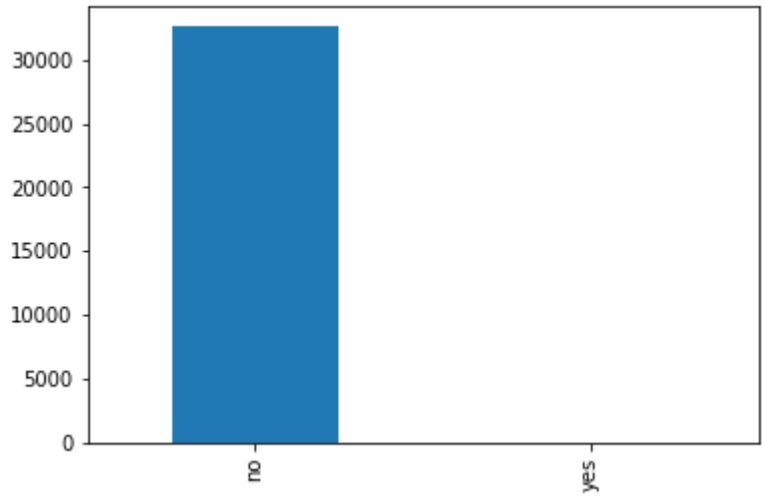
marital



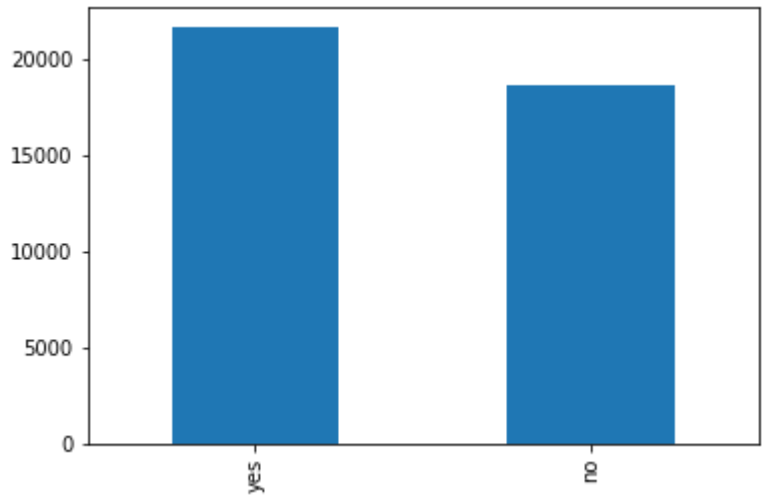
education



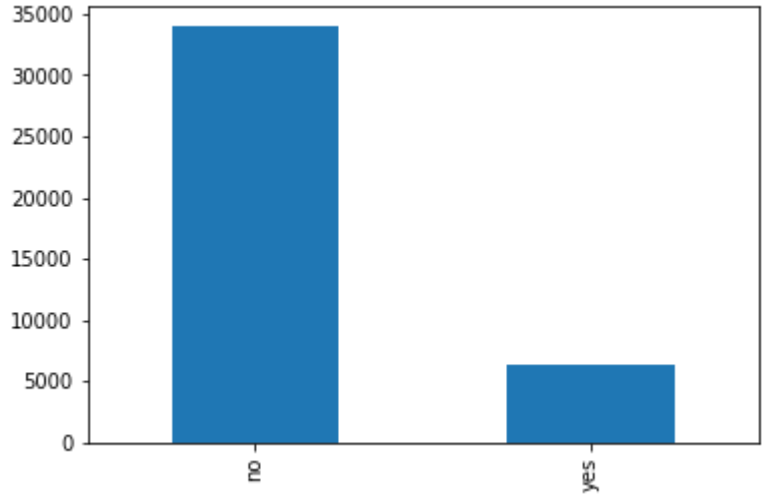
default



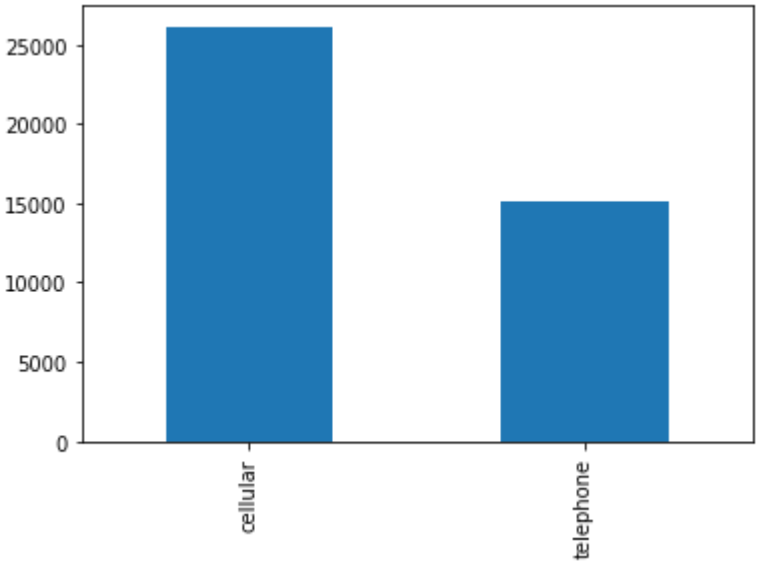
housing



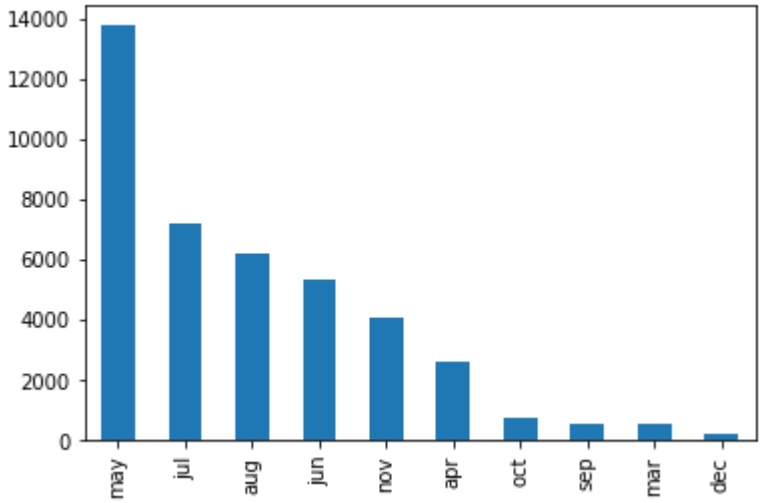
loan



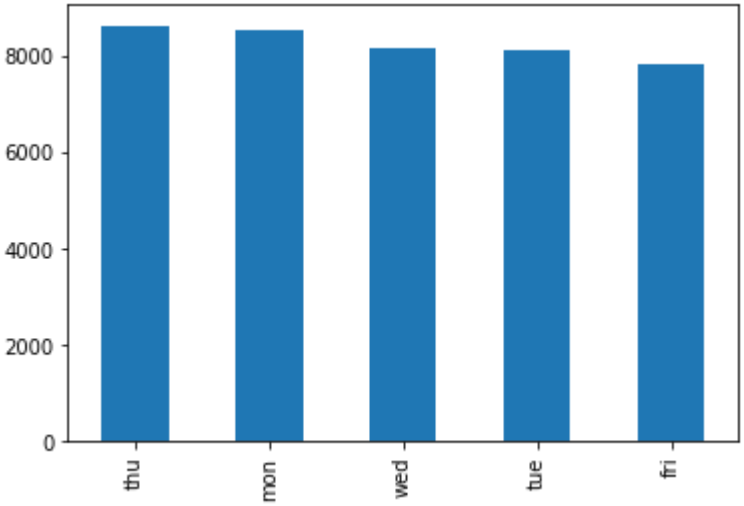
contact



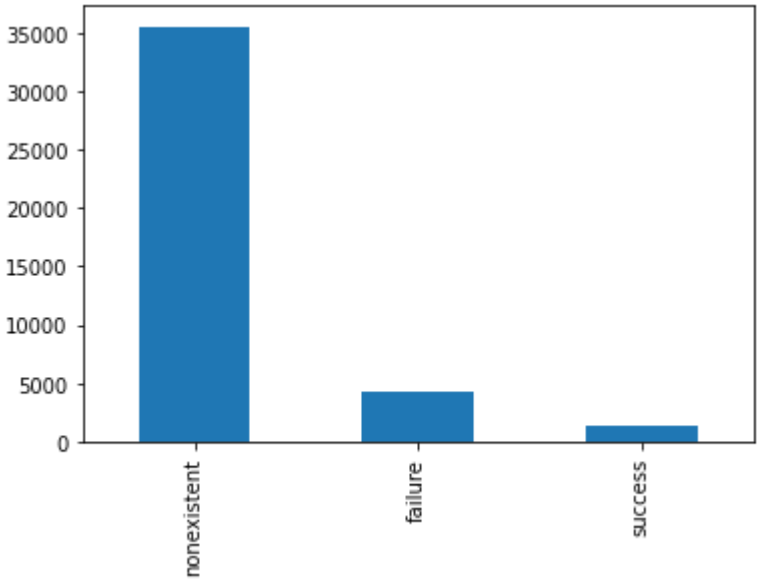
month



day_of_week



poutcome



y

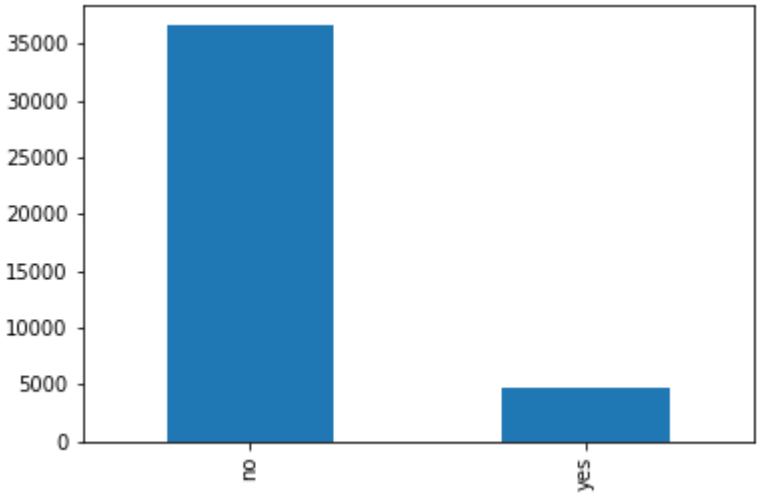
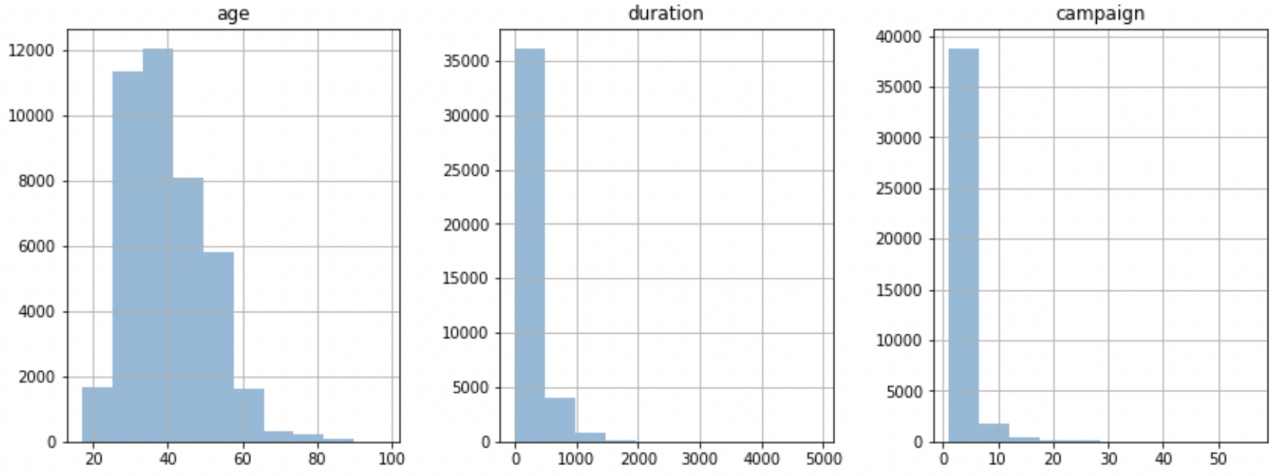
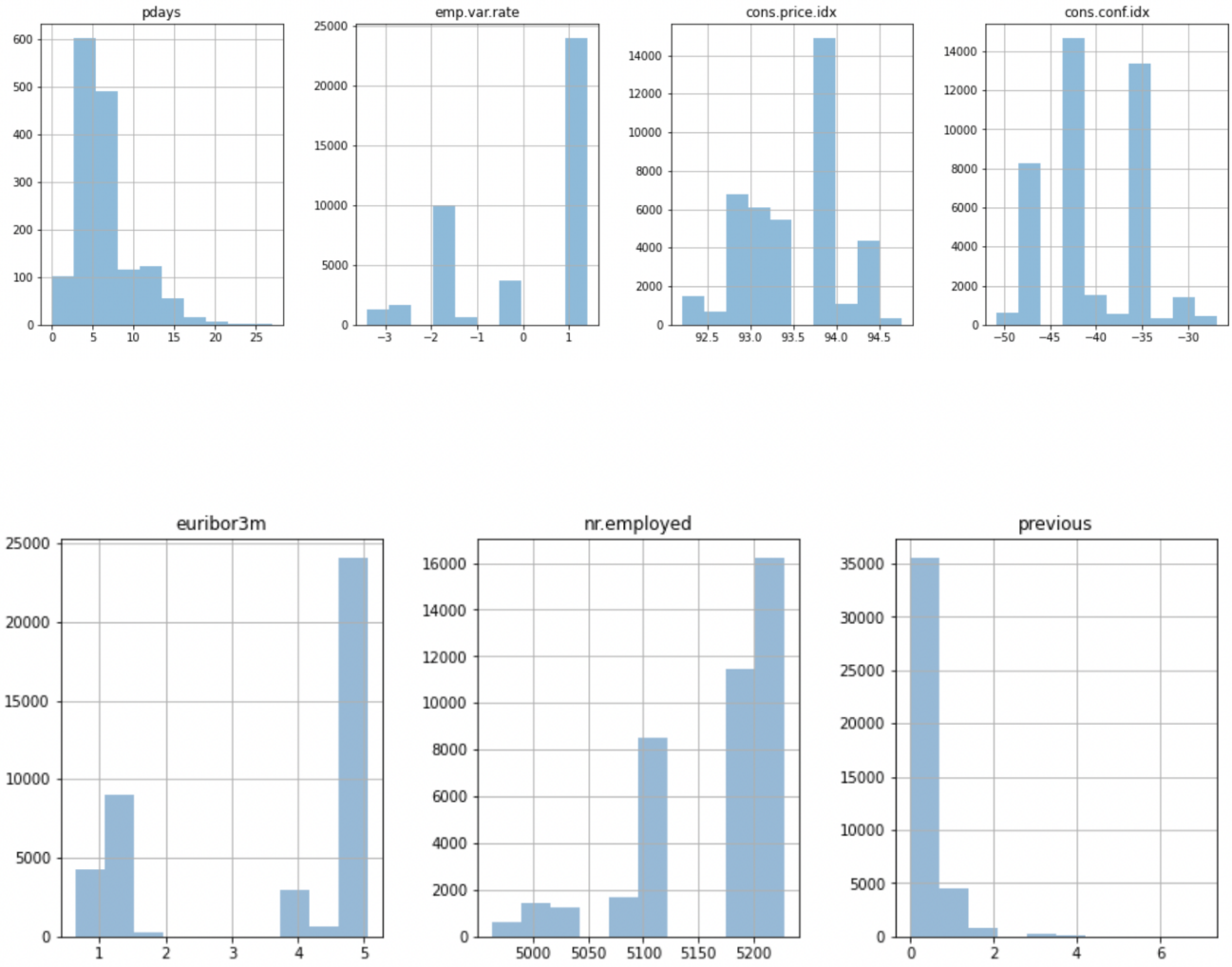


Table 2. Data Quality Report for Continuous Features

Feature	% of Missing	Card.	Count	Max	Mean	Median	Min.	Q1	Q3	Std. Dev.
age	0.00	78.0	41176.0	98.000	40.023800	38.000	17.000	32.000	47.000	10.420680
duration	0.00	1544.0	41176.0	4918.000	258.279837	180.000	0.000	102.000	319.000	259.260227
campaign	0.00	42.0	41176.0	56.000	2.567879	2.000	1.000	1.000	3.000	2.770318
pdays	96.32	27.0	41176.0	27.000	6.014521	6.000	0.000	3.000	7.000	3.824906
emp.var.rate	0.00	10.0	41176.0	1.400	0.081922	1.100	-3.400	-1.800	1.400	1.570883
cons.price.idx	0.00	26.0	41176.0	94.767	93.575720	93.749	92.201	93.075	93.994	0.578839
cons.conf.idx	0.00	26.0	41176.0	-26.900	-40.502863	-41.800	-50.800	-42.700	-36.400	4.627860
euribor3m	0.00	316.0	41176.0	5.045	3.621293	4.857	0.634	1.344	4.961	1.734437
nr.employed	0.00	11.0	41176.0	5228.100	5167.034870	5191.000	4963.600	5099.100	5228.100	72.251364
previous	0.00	8.0	41176.0	7.000	0.173013	0.000	0.000	0.000	0.000	0.494964

Figure 1. Visualizations of Continuous Features in Dataset





2.2 Missing Values and Outliers

To increase the performance of our classifiers, we need to replace or remove all missing values and identify outliers to see if they should be replaced/removed as well. For continuous features, only pdays has missing values, perhaps due to there being no calls on record between the customer and the bank. For these missing values we simply replace the NaN values with 0 for now, though a separate solution may be used to improve upon this feature in case the accuracy of this replacement affects classification. For categorical features, there are quite a few that have missing values- though since a couple have only less than 1 percent missing values, the values are dropped using dropna instead of simply filling the values. For those categorical features with more than 1 percent missing values, the missing values are replaced by the mode to prevent any conflicts with our classifier's outcomes.

2.3 Normalization

During the normalization of the data, the values for continuous data in the dataset is scaled between values of 0 and 1 based on the minimum and maximum values for each column. During normalization it can be more clear to the viewer that there are certain columns that have values that generally reach either 0 or 1 after the process, which makes it more simpler when it comes to calculating the weight of each specific value. Initial feature ranges such as the range for duration [0, 4918] and even age [17, 98] are the features that benefit the most from this process, as the data that can be visualized from normalizing them is more concise and easier to read on graphs. Most normalized/continuous features in this dataset are usually spread out across only a few specific values, however, so they may benefit from the normalization less- but the process will create more concise and readable data regardless.

```
In [24]: scaler = preprocessing.MinMaxScaler()

         d = scaler.fit_transform(transformed_data)

         scaled_df = pd.DataFrame(d, columns=transformed_data.columns)


         scaled_df
```

Out[24]:

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	...	day_of_week_fri	day_of_week_mo
0	0.481481	0.053070	0.000000	0.0	0.000000	0.937500	0.698753	0.60251	0.957379	0.859735	...	0.0	1
1	0.493827	0.030297	0.000000	0.0	0.000000	0.937500	0.698753	0.60251	0.957379	0.859735	...	0.0	1
2	0.246914	0.045954	0.000000	0.0	0.000000	0.937500	0.698753	0.60251	0.957379	0.859735	...	0.0	1
3	0.283951	0.030704	0.000000	0.0	0.000000	0.937500	0.698753	0.60251	0.957379	0.859735	...	0.0	1
4	0.481481	0.062424	0.000000	0.0	0.000000	0.937500	0.698753	0.60251	0.957379	0.859735	...	0.0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	
40770	0.691358	0.067914	0.000000	0.0	0.000000	0.479167	1.000000	0.00000	0.089322	0.000000	...	1.0	0
40771	0.358025	0.077877	0.000000	0.0	0.000000	0.479167	1.000000	0.00000	0.089322	0.000000	...	1.0	0
40772	0.481481	0.038430	0.018182	0.0	0.000000	0.479167	1.000000	0.00000	0.089322	0.000000	...	1.0	0
40773	0.333333	0.089874	0.000000	0.0	0.000000	0.479167	1.000000	0.00000	0.089322	0.000000	...	1.0	0
40774	0.703704	0.048597	0.036364	0.0	0.142857	0.479167	1.000000	0.00000	0.089322	0.000000	...	1.0	0

40775 rows × 59 columns

2.4 Transformations

We encoded the categorical features to numerical values used in the model to get better-predicted values. For this, we are going to use one-hot encoding. To binarize these categorical variables, we apply hot encoding. One-hot encoding is done since the ML models can not use text-based features to train models.

We use pd.get_dummies as an encoding method since we want a fair and quantifiable distribution. General encoding gives [0, infinite] values that we do not want.

In [10]:
```
transformed_data = pd.get_dummies(df, columns=cat_columns)

transformed_data.head()
```

Out[10]:

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	...	day_of_week_fri	day_of_week_mon	day_of
0	56	261.0	1	0.0	0	1.1	93.994	-36.4	4.857	5191.0	...	0	1	
1	57	149.0	1	0.0	0	1.1	93.994	-36.4	4.857	5191.0	...	0	1	
2	37	226.0	1	0.0	0	1.1	93.994	-36.4	4.857	5191.0	...	0	1	
3	40	151.0	1	0.0	0	1.1	93.994	-36.4	4.857	5191.0	...	0	1	
4	56	307.0	1	0.0	0	1.1	93.994	-36.4	4.857	5191.0	...	0	1	

5 rows × 59 columns