2022

MINI PROJECT 2

TEAM MEMBERS:

Anand Manivannan
Ayush Mohan
Shalika Siddique
Srishti Srivastav

1. **EXPLORATORY DATA ANALYSIS**

a) **Statistical Exploration**

- First we explore basic statistical information using Pyspark function *describe()* for our features like record counts, mean, standard deviation, min & max values to understand if there are odd or extreme values that require special handling
- **We find that our data is accurately captured except for pdays field that contains a max value of '999'. This value is label-encoded in future steps as a new category since it represents clients who were never contacted previously**

Out[12]:

	0	1	2	3	4
summary	count	mean	stddev	min	max
age	41188	40.02406040594348	10.42124998093404З	17	98
job	41188	None	None	admin.	unknown
marital	41188	None	None	divorced	unknown
education	41188	None	None	basic.4y	unknown
default	41188	None	None	no	yes
housing	41188	None	None	no	yes
loan	41188	None	None	no	yes
contact	41188	None	None	cellular	telephone
month	41188	None	None	apr	sep
day_of_week	41188	None	None	fri	wed
duration	41188	258.2850101971448	259.27924883646455	0	4918
campaign	41188	2.567592502670681	2.770013542902331	1	56
pdays	41188	962.4754540157328	186.910907344741	0	999
previous	41188	0.17296299893172767	0.49490107983928927	0	7
poutcome	41188	None	None	failure	success
emp_var_rate	41188	0.08188550063178966	1.57095974051703	-3.4	1.4
cons_price_idx	41188	93.5756643682899	0.5788400489540823	92.201	94.767
cons_conf_idx	41188	-40.502600271918276	4.628197856174573	-50.8	-26.9
euribor3m	41188	3.621290812858533	1.7344474048512595	0.634	5.045
nr_employed	41188	5167.035910943957	72.25152766826338	4963.6	5228.1
y	41188	None	None	no	yes

b) **Target Variable Distribution**

- Next we look at the distribution of responses in the target variable using Pyspark's *groupby* function– **we find that our dataset is imbalanced and that the majority of responses are of the 'no' category**

y	count
no	36548
yes	4640

c) **Exploring Categorical Features**

- Next we explore the distributions of all the categories in our categorical features if certain rare categories require to be combined – combining these rare categories would serve to reduce bias while training our models
- **Some of these categories(‘*Illiterate’* category in Education feature and ‘*Yes’* category in Default feature) represent a very small proportion of records. To reduce bias in modelling, these categories were combined with their respective ‘unknown’ category.**
- **Additionally the ‘*unknown’* category in *Marital feature was replaced as missing values, since the proportion of records was very small.***

marital	count
unknown	80
divorced	4612
married	24928
single	11568

education	count
high.school	9515
unknown	1731
basic.6y	2292
professional.course	5243
university.degree	12168
illiterate	18
basic.4y	4176
basic.9y	6045

d) **Missing Value Analysis and Handling**

- First, we replace the missing values of Marital feature(from the previous step) with its mode
- Next we find missing values for each feature using Pyspark function *isnan* **– We find that there are no missing values in any of the features**

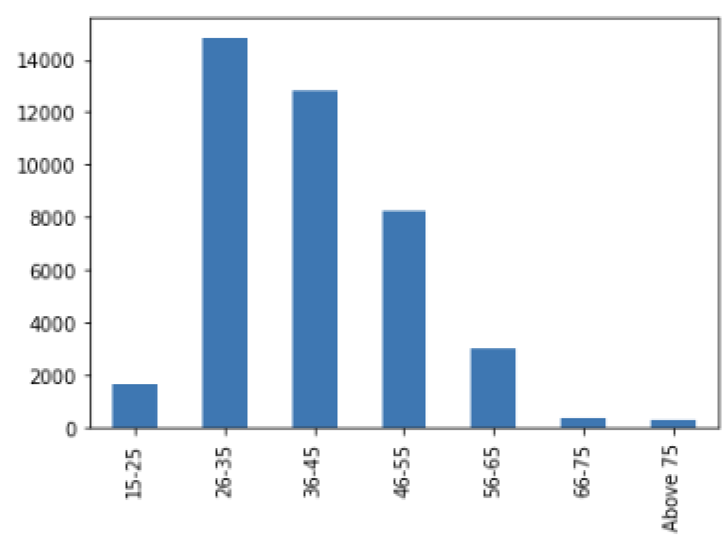
Out[11]:

	0
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

e) Binning, Label-Encoding & Uni-Variate Analysis

- Next we bin features such as Age and pdays so we can convert them into categorical features, to represent the information more accurately
- **Looking at the distribution of our binned 'Age' variable, we find ages '26-35' are the most prominent category. This feature was considered as an ordinal variable in the modeling phase**

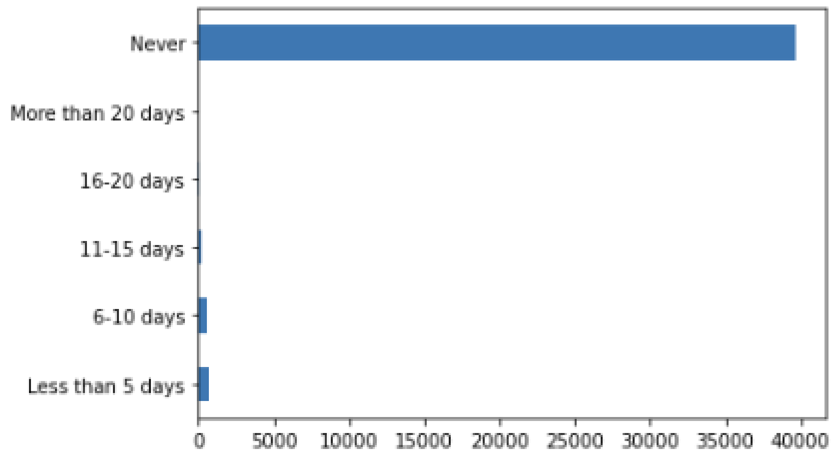
Out[20]: <AxesSubplot:>



- Next looking at the distribution of binned 'pdays' feature, which represents the time period between the last time a client was contacted before, the value '999' was considered to be a new category 'Never' since this value represents customers who were never contacted before.

- **Exploring the distribution of pdays, we find that most customers were never contacted before through any campaign**

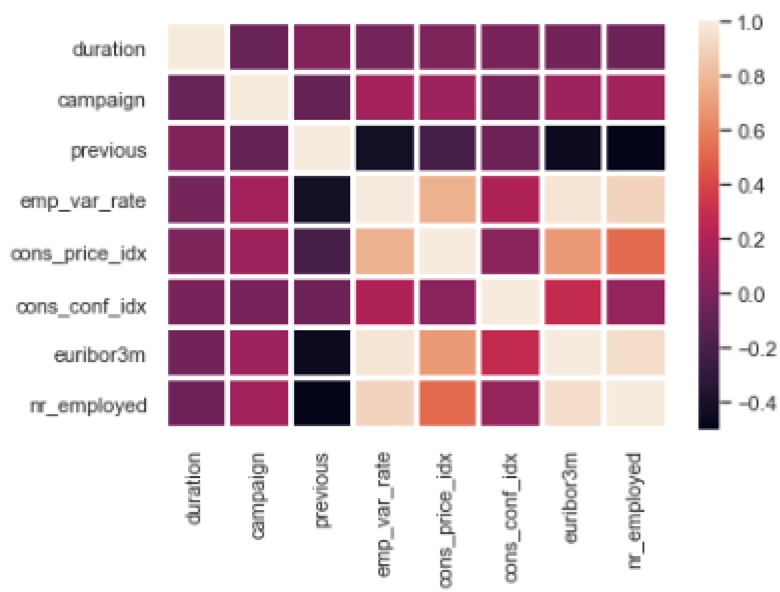
Out[26]: <AxesSubplot:>



f) Correlation Analysis

- Next we construct a correlation matrix for numerical features using Pearson correlation to identify highly correlated features
- We find several features to be highly correlated, for example, emp_var_rate is highly correlated with euribor3m, nr_employed and cons_price_idx with a pearson coefficient higher than 0.7.
- Studying the variable descriptions of these features, we find these are all social and economic indicators representing the same information- which justifies the reason for the correlation. **We remove these correlated features(emp_var_rate and euribor3m) so as to not introduce bias in the modeling phase.**

Out[30]: <AxesSubplot:>

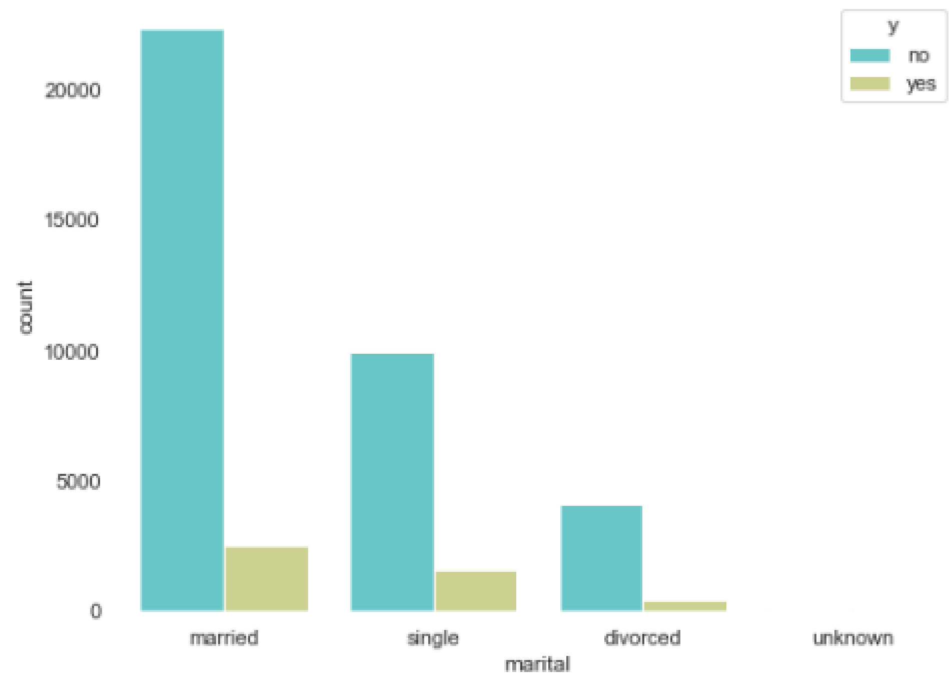


g) Bi-Variate Analysis

- Next we perform a bivariate analysis of our input features against our target variable, to uncover any trends or patterns. NOTE – A few snippets are posted below for explanation purposes and a comprehensive list can be found in our Notebook file.

- **Looking at target responses across the 'marital' variable we find married clients are most probable to subscribe to a term deposit**

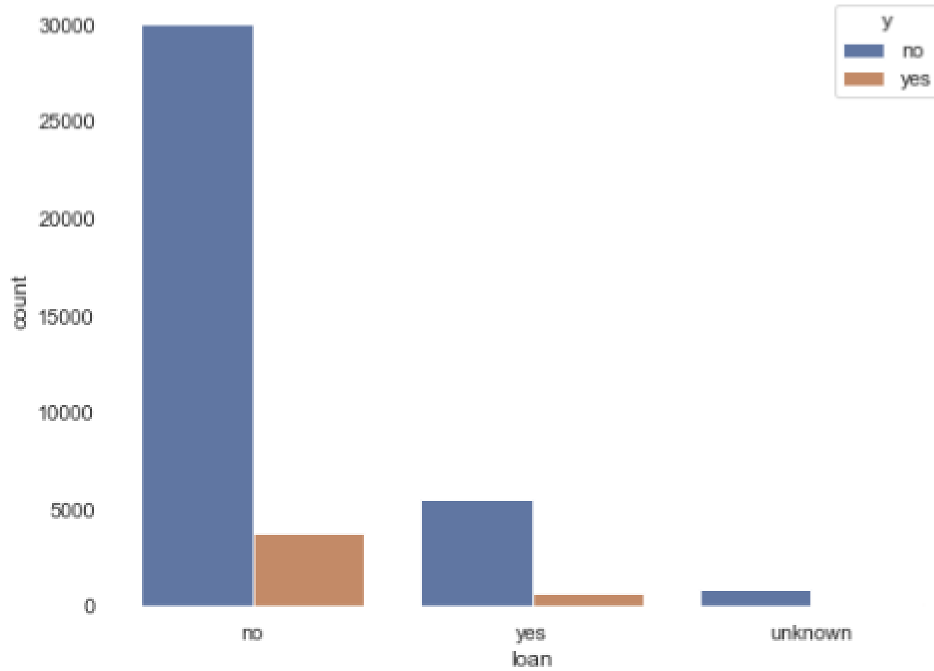
```
Out[31]: <AxesSubplot:xlabel='marital', ylabel='count'>
```



- **Looking at target responses across the 'loan variable we find clients without a personal loan are most probable to subscribe to a term**

deposit

Out[34]: `<AxesSubplot:xlabel='loan', ylabel='count'>`



h) Cross-Tab Analysis

- For categorical variables with higher than 2 categories, we do a cross tab analysis to look at the distribution of responses across categories. NOTE – A few snippets are posted below for explanation purposes and a comprehensive list can be found in our Notebook file.
- **Looking at our 'job' feature, we find clients working as 'admin' to have the highest responses**

Out[40]:

job_y	housemaid	services	self-employed	student	retired	unknown	admin.	blue-collar	technician	entrepreneur	management	unemployed
no	954	3646	1272	600	1286	293	9070	8616	6013	1332	2596	870
yes	106	323	149	275	434	37	1352	638	730	124	328	144

- **Similarly, looking at our 'education' feature, we find clients who have completed high school to have the highest responses**

Out[41]:

education_y	basic.9y	basic.4y	high.school	unknown	basic.6y	professional.course	university.degree	illiterate
no	5572	3748	8484	1480	2104	4648	10498	14
yes	473	428	1031	251	188	595	1670	4

2. MODELLING

- **Data Preparation**

  The following steps were undertaken to prepare the dataset for modelling –

  1. **Numerical and categorical columns are defined –** This step divides the categorical and numerical features to be fed into the predictive models
  2. **Categorical features were indexed and one-hot encoded –** This step would convert a categorical column into multiple columns for each class and thus capture the information more accurately
  3. **Data is assembled as a vector -** This step transforms all the numerical data along with the encoded categorical data into a series of vectors using the Pyspark's `VectorAssembler` function.

4. **Features were scaled(transformed) –** This would normalize features with different ranges of values to appear consistently
5. **A transformation pipeline is created, saved and loaded** to prepare for modelling phase
6. **Train/Validation Split –** We create a 70:30 training and validation split to evaluate the performance of our model on unseen data

- **Predictive Modelling**
  A total of 5 models were built using Pyspark. The evaluation metrics of these models are presented below –

  1. Logistic Regression

  2. Decision Tree

  3. Random Forest Classifier

  4. Gradient Boost Classifier

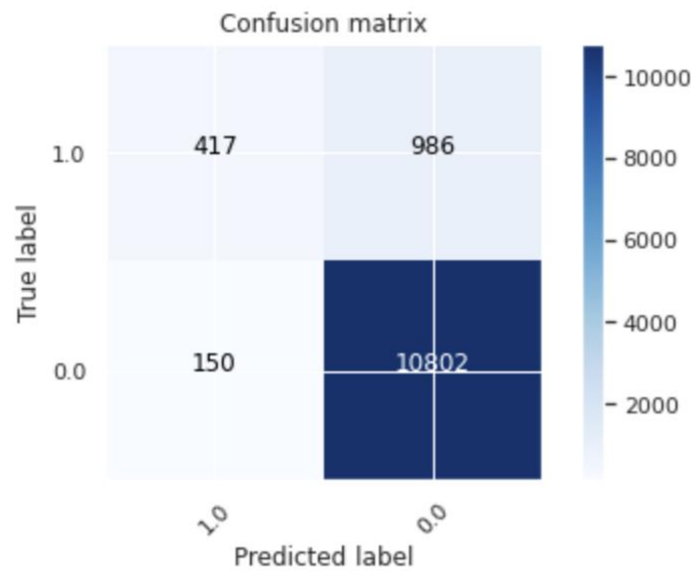
  5. Linear Support Vector Classifier

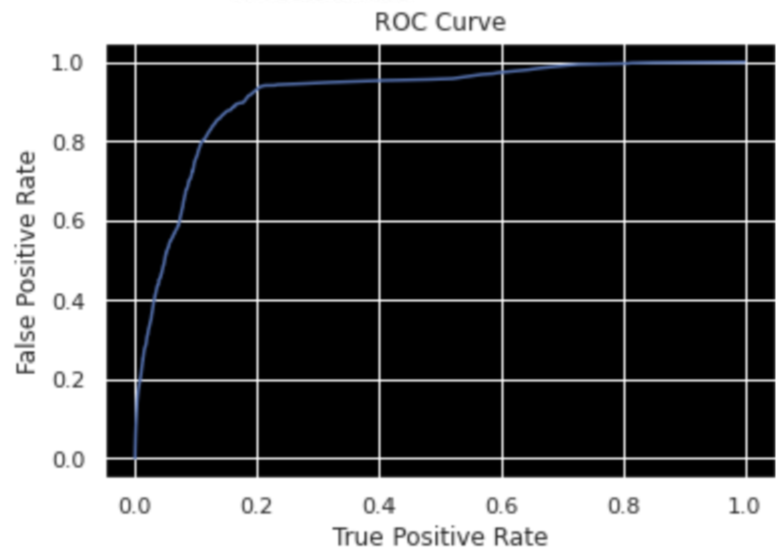
  ParamGridBuilder was used to include various parameters for **hyperparameter tuning**. Additionally, **5-fold cross validation** was performed with roc as the evaluation metric.

- **Model Evaluation**
  - Accuracy and AUC were chosen as the evaluation metrics to determine the best performing model, since we are concerned with accurate classification of clients who will respond to a term deposit offer.
  - Once these metrics were calculated (found below), we found our Random Forest to have the highest AUC(0.825) and Gradient Boosting to have the highest Accuracy(94.5%)

Model/Metric	**Accuracy**	**AUC**
Logistic Regression	0.938	0.806
Decision Tree	0.822	0.802
Random Forest	0.941	0.825
Gradient Boosting	0.945	0.805
Linear Support Vector Classifier	0.934	0.778

  - Since classification True Positives are more important than just model Accuracy, AUC was the deciding factor in choosing our **Random Forest model** as the champion model
  - The presented output is the classification matrix and ROC chart for the Random Forest model:





- **Benefits of Random Forest Classifier**

  A random forest classifier, is constructed by building several heavily unpruned trees and making classifications by averaging the decisions from all the trees. This poses several advantages in classifications –

  1. They solve the over-fitting issue from decision trees by averaging predictions from multiple trees. Hence they have low bias during training and moderate variance with predictions
  2. They use a random subset of variables to build each tree, thus de-correlating subsequently built trees
  3. They handle imbalanced data
  4. They don't make assumptions about the distributions of the data, hence it is quite robust against non-linearity, extreme values and outliers
  5. Work well with datasets that have a large number of features, which is often common in real-world settings

- **Feature Importance**

To interpret the most significant features that contribute to clients subscribing to a term deposit, a Gini-Based Method was used to extract feature importance from our Random Forest model

From the list below, we can see **duration** and **nr_employed** most significantly affect the probability of client responding to a term deposit subscription

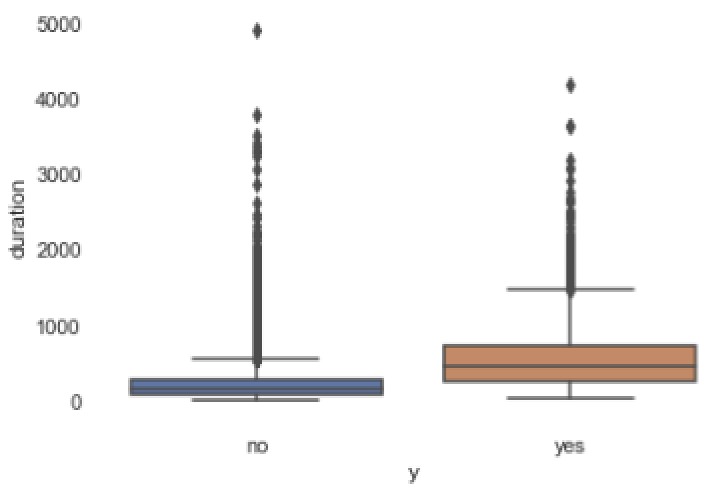
	idx	name	score
0	49	duration	0.438633
1	55	nr_employed	0.228233
2	51	pdays	0.053575
3	54	cons_conf_idx	0.047398
4	27	contactclassVec_cellular	0.034464
5	53	cons_price_idx	0.026323
6	50	campaign	0.020138
7	5	jobclassVec_retired	0.014451
8	33	monthclassVec_apr	0.013596
9	28	monthclassVec_may	0.013213

3. RECOMMENDATIONS

Based on our 3 most significant variables, we can make the following recommendations –

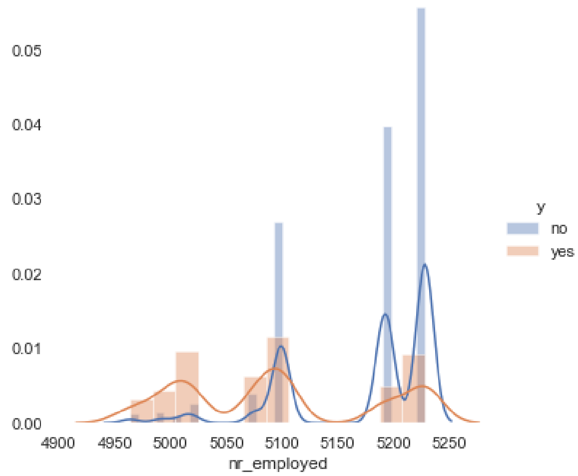
- **Duration**

  This feature represents the time elapsed in seconds from when the customer was most recently contacted. Looking at the trend chart below, we see that the majority of the customers subscribe for term deposits when the contact duration is 500-1000 seconds. We recommend the ideal contact duration to be within this range to improve responses.



- **Nr_employed**

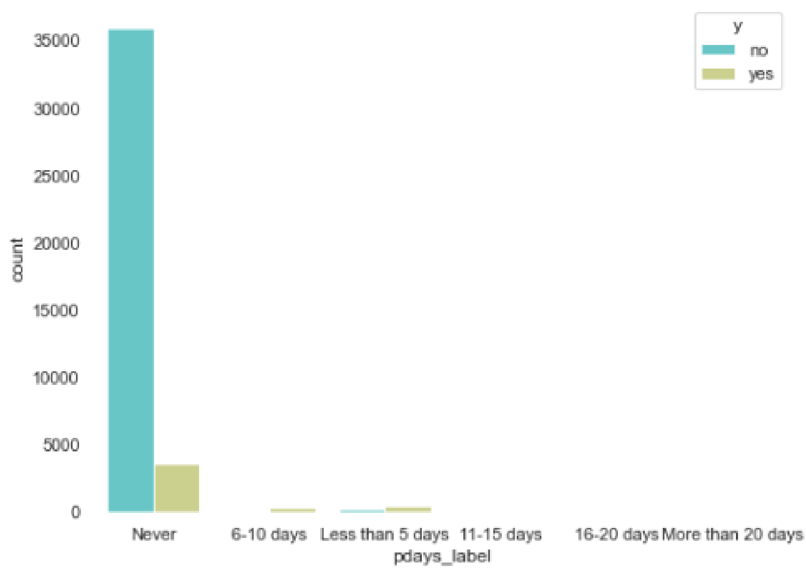
  This feature measures the number of employees on a quarterly basis. Looking at the trend chart below, we can see the number of customers not responding to offers increases exponentially when the employees exceeded 5150. We would recommend employing 5050-5150 employees to maximize profits



- **Pdays**

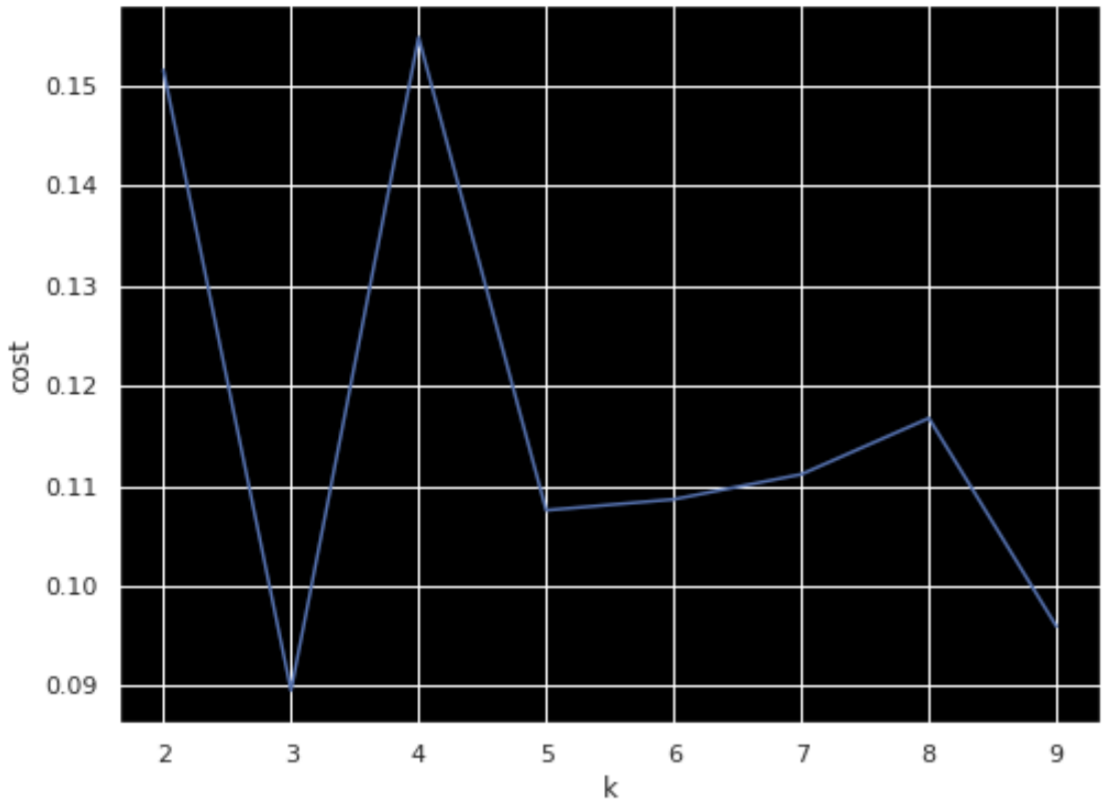
  This feature represents **the** number of days that passed by after the client was most recently contacted from a previous campaign. Looking at the trend chart below, we can see that customers who were never contacted from a previous campaign were most likely to respond. We would recommend prioritizing these customers.

```
Out[140]: <AxesSubplot:xlabel='pdays_label', ylabel='count'>
```



4. K-MEANS CLUSTERING (BONUS POINTS)

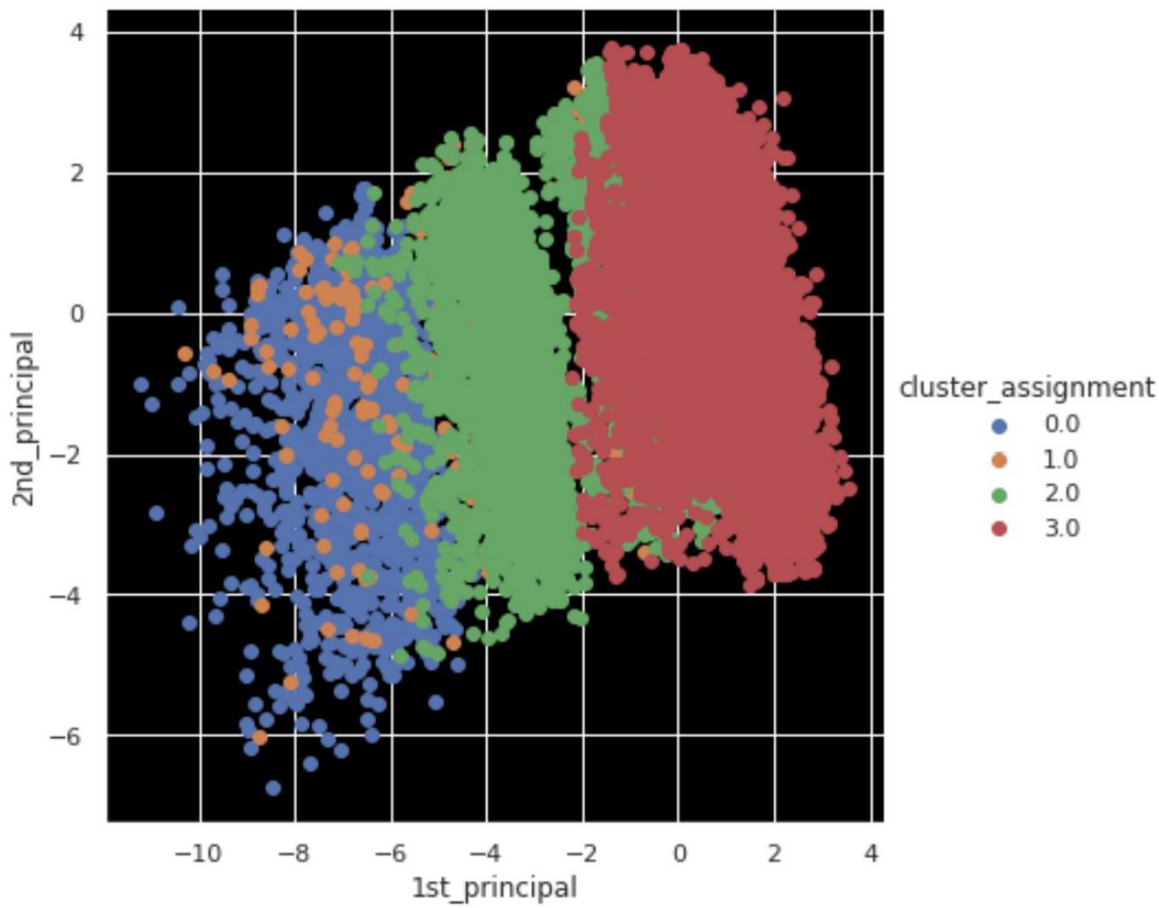
- First, the optimal number of clusters was found using Silhouette scores
- Since local maxima was 4, we selected 4 clusters



- Following represents the count of observations in each cluster

```
+----------+-----+
|prediction|count|
+----------+-----+
|         1|  845|
|         3|24695|
|         2|11726|
|         0| 3922|
+----------+-----+
```

- To visualize the distributions of the clusters, Principal Component Analysis(PCA) was performed and 2 PCA variable were created. Following is the output:


4
2
0
−2
−4
−6
2nd_principal
−10
−8
−6
−4
−2
0
2
4
1st_principal
cluster_assignment
0.0
1.0
2.0
3.0