Bank Marketing (Campaign)

•••

The Data Detectives

Problem Description:

ABC Bank wants to predict whether a customer will subscribe to their term deposit product based on past interactions. They aim to develop a machine learning model to identify customers who are more likely to purchase the product. In other words, they want to shortlist customers whose chances of buying the product is more.

Business Understanding:

After the development of a ML predictive model, we can then make assumptions about the class of customers more likely to purchase the product. This will help the bank focus its marketing efforts on those customers during future marketing campaigns. By narrowing the campaign scope, the bank can reduce costs, save resources, and improve profit margins.

Data Understanding:

The dataset we are going to use for the analysis is called "bank-additional-full.csv", which contains 41188 observations and 21 features, encompassing features related to clients' basic information such as age, job, marital status, education, credit in default, housing, and loan; details about contact such as contact communication type, last contact month, last contact day, last contact duration, number of contacts, etc., and information about marketing campaigns like outcome, employment variation rate, consumer price index, consumer confidence index, euribor 3 month rate, and number of employees. We also have the target variable y, which is the answer for the yes-no question "has the client subscribed a term deposit?", and it will be used in future prediction.

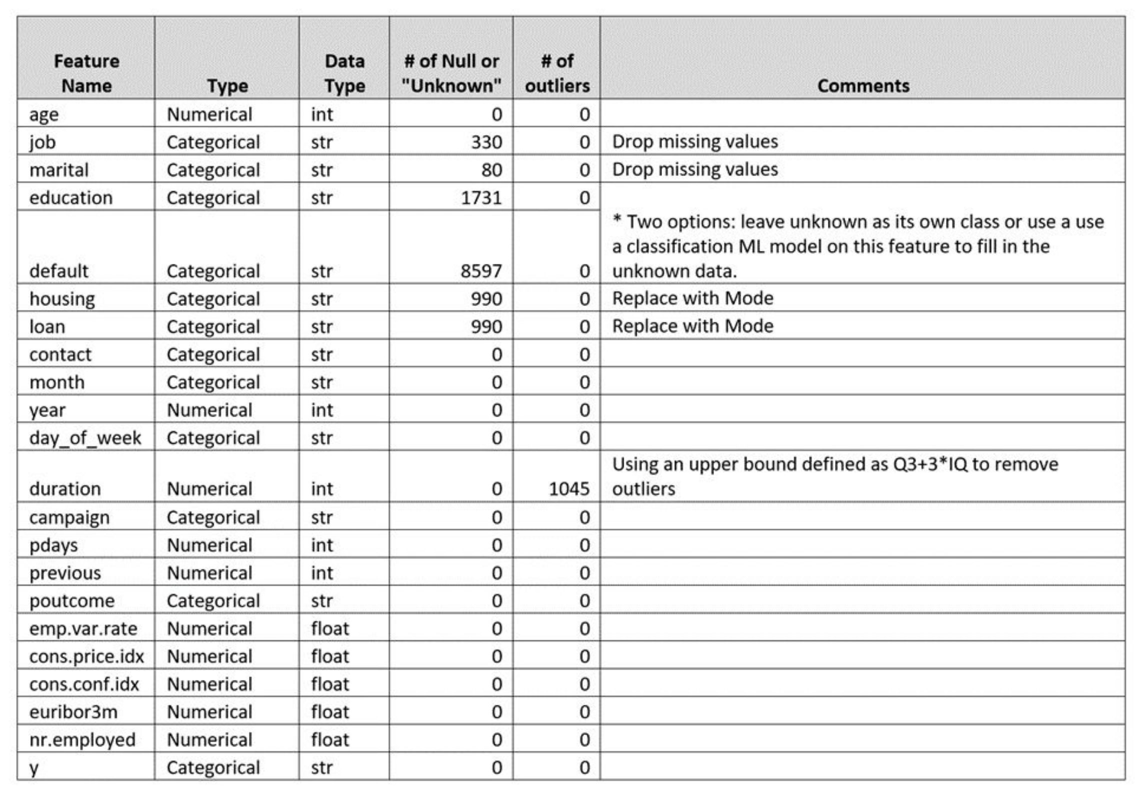
Understanding The Dataset

Name	Type	About
age	Numeric	Age of the customer
job	Categorical	Customer’s occupation
marital	Categorical	Customer’s marital status
education	Categorical	Customer’s education background
default	Categorical	If customer has credit in default
housing	Categorical	If customer has housing loan
loan	Categorical	If customer has personal loan
contact	Categorical	Customer’s contact type
month	Categorical	Customer’s last month of contact
day_of_week	Categorical	Customer’s last weekday of contact

Understanding The Dataset

Name	Type	About
duration	Numeric	Customer’s last contact duration (s)
campaign	Numeric	# of contacts during this campaign
pdays	Numeric	number of days that passed by after the client was last contacted
previous	Numeric	number of contacts performed before this campaign and for this client
poutcome	Categorical	outcome marketing campaign
emp.var.rate	Numeric	employment variation rate quarterly
cons.price.idx	Numeric	consumer price index - monthly
cons.conf.idx	Numeric	consumer confidence index - monthly
euribor3m	Numeric	euribor 3 month rate - daily
nr.employed	Numeric	number of employees - quarterly

Another Visualization



Feature Name	Type	Data Type	# of Null or "Unknown"	# of outliers	Comments
age	Numerical	int	0	0	
job	Categorical	str	330	0	Drop missing values
marital	Categorical	str	80	0	Drop missing values
education	Categorical	str	1731	0	* Two options: leave unknown as its own class or use a use a classification ML model on this feature to fill in the unknown data.
default	Categorical	str	8597	0	
housing	Categorical	str	990	0	Replace with Mode
loan	Categorical	str	990	0	Replace with Mode
contact	Categorical	str	0	0	
month	Categorical	str	0	0	
year	Numerical	int	0	0	
day_of_week	Categorical	str	0	0	
duration	Numerical	int	0	1045	Using an upper bound defined as Q3+3*IQ to remove outliers
campaign	Categorical	str	0	0	
pdays	Numerical	int	0	0	
previous	Numerical	int	0	0	
poutcome	Categorical	str	0	0	
emp.var.rate	Numerical	float	0	0	
cons.price.idx	Numerical	float	0	0	
cons.conf.idx	Numerical	float	0	0	
euribor3m	Numerical	float	0	0	
nr.employed	Numerical	float	0	0	
y	Categorical	str	0	0	

Two main questions:

1. What are the problems in the data (number of NA values, outliers , skewed etc) ?

2. What approaches are you trying to apply on your data set to overcome problems like NA value, outlier etc and why?

What are the problems in the data (number of NA values, outliers , skewed etc)?

There are 6 categorical features with missing data (job, education, marital, default, housing, & loan). There is one numerical feature (“duration”) that contains outlier data. Specifically, we have the mean for “duration” is around 258, but the maximum value is 4918, which indicates the existence of outliers. And in general, the dataset is imbalanced, as the target variable for the predictive classification model skews ~90% to the “N” case.

What approaches are you trying to apply on your data set to overcome problems like NA value, outlier etc and why?

In handling missing (NA) values, we will employ a variety of techniques tailored to the severity of each column and its overall impact on the dataset. Dropping the missing data for those features with lower numbers of “unknown” data points (“marital” & “job”). Replacing the missing data with the most frequent category for “housing” and “loan”. And using a ML classification model to fill the missing values for the “default” and “education” features.

For the outlier numerical data, as mentioned above, we can use an upper outer fence defined at 3IQ (upper fence = Q3 + 3*IQR), where IQR is defined as interquartile range, allowing us to retain 97% of the original data.

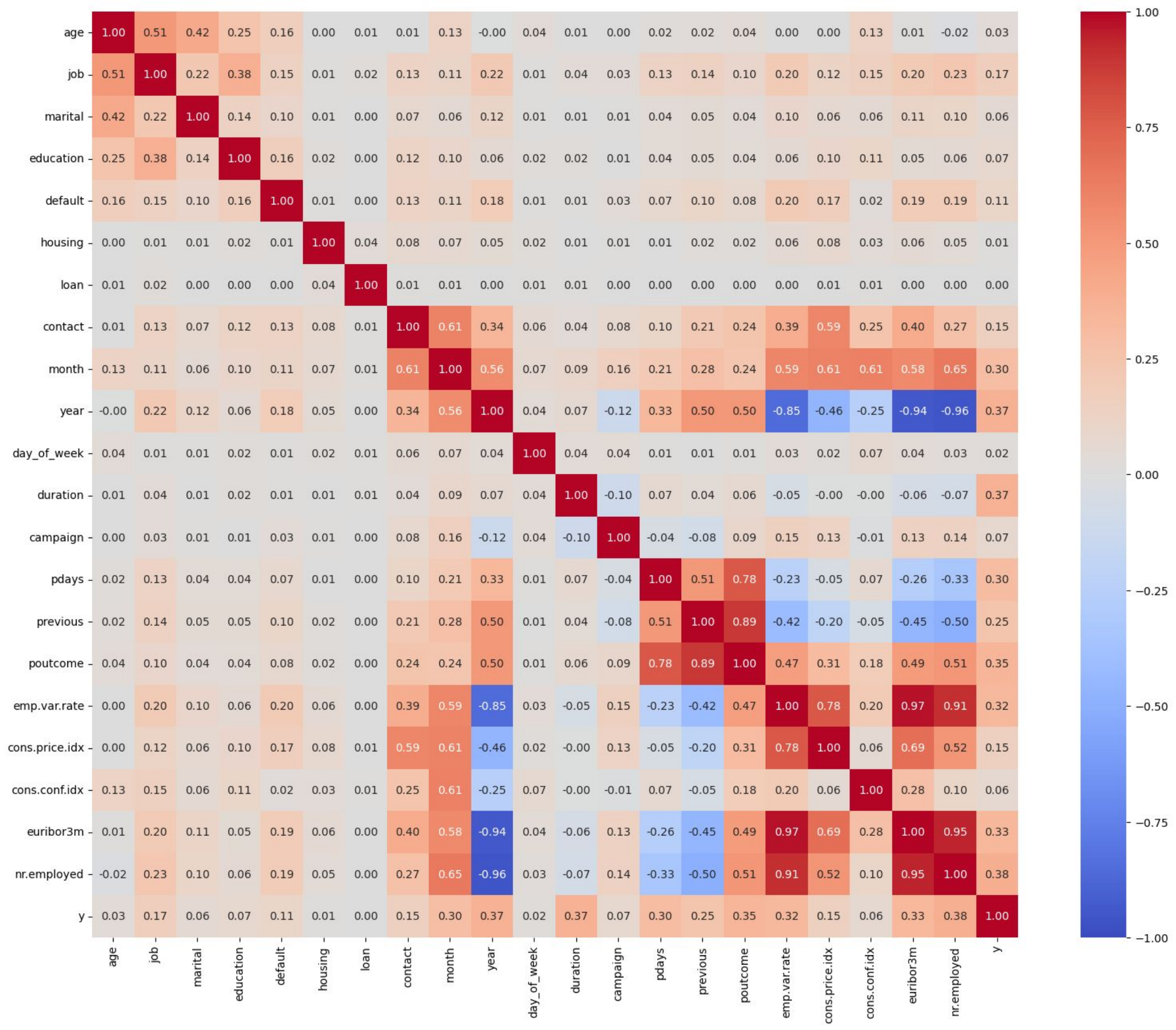
For the imbalance related to the target variable, we can help account for this imbalance in the model by choosing the correct evaluation metric. For this data set, that most likely will mean using the AUROC curve to help identify which models provide the best results for True Positive and False Negative predictions. Additionally, since the size of the dataset is large enough, we could consider under-sampling from the majority case. Or, when splitting the data during training, instead of randomizing the folds, we can ensure that the rare cases are kept each time and only randomly split from the majority case. We can also manipulate the ratio of rare:majority cases in the training data to over-represent the rare case for the model.

Data Exploration

Uncovering patterns, trends, and correlations that allow us to select those features that are most relevant when training the ML model.

Correlation Heat Map

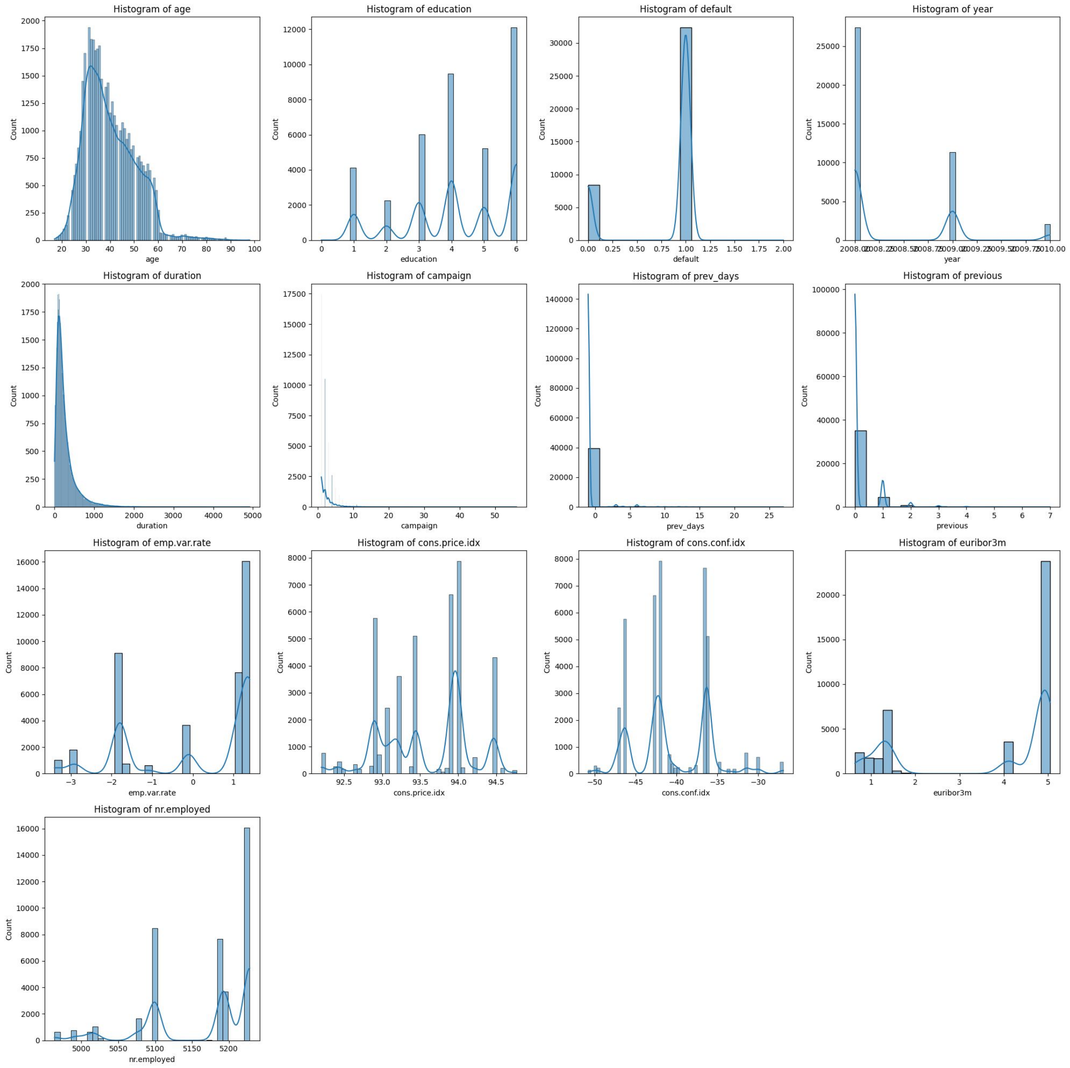
The figure to the right shows the strength of correlation of each feature to one another amongst the dataset. For our problem statement, we are most interested in those features that most strongly correlate with the target variable ‘y.’



	age	job	marital	education	default	housing	loan	contact	month	year	day_of_week	duration	campaign	pdays	previous	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
age	1.00	0.51	0.42	0.25	0.16	0.00	0.01	0.01	0.13	-0.00	0.04	0.01	0.00	0.02	0.02	0.04	0.00	0.00	0.13	0.01	-0.02	0.03
job	0.51	1.00	0.22	0.38	0.15	0.01	0.02	0.13	0.11	0.22	0.01	0.04	0.03	0.13	0.14	0.10	0.20	0.12	0.15	0.20	0.23	0.17
marital	0.42	0.22	1.00	0.14	0.10	0.01	0.00	0.07	0.06	0.12	0.01	0.01	0.01	0.04	0.05	0.04	0.10	0.06	0.06	0.11	0.10	0.06
education	0.25	0.38	0.14	1.00	0.16	0.02	0.00	0.12	0.10	0.06	0.02	0.02	0.01	0.04	0.05	0.04	0.06	0.10	0.11	0.05	0.06	0.07
default	0.16	0.15	0.10	0.16	1.00	0.01	0.00	0.13	0.11	0.18	0.01	0.01	0.03	0.07	0.10	0.08	0.20	0.17	0.02	0.19	0.19	0.11
housing	0.00	0.01	0.01	0.02	0.01	1.00	0.04	0.08	0.07	0.05	0.02	0.01	0.01	0.01	0.02	0.02	0.06	0.08	0.03	0.06	0.05	0.01
loan	0.01	0.02	0.00	0.00	0.00	0.04	1.00	0.01	0.01	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00
contact	0.01	0.13	0.07	0.12	0.13	0.08	0.01	1.00	0.61	0.34	0.06	0.04	0.08	0.10	0.21	0.24	0.39	0.59	0.25	0.40	0.27	0.15
month	0.13	0.11	0.06	0.10	0.11	0.07	0.01	0.61	1.00	0.56	0.07	0.09	0.16	0.21	0.28	0.24	0.59	0.61	0.61	0.58	0.65	0.30
year	-0.00	0.22	0.12	0.06	0.18	0.05	0.00	0.34	0.56	1.00	0.04	0.07	-0.12	0.33	0.50	0.50	-0.85	-0.46	-0.25	-0.94	-0.96	0.37
day_of_week	0.04	0.01	0.01	0.02	0.01	0.02	0.01	0.06	0.07	0.04	1.00	0.04	0.04	0.01	0.01	0.01	0.03	0.02	0.07	0.04	0.03	0.02
duration	0.01	0.04	0.01	0.02	0.01	0.01	0.01	0.04	0.09	0.07	0.04	1.00	-0.10	0.07	0.04	0.06	-0.05	-0.00	-0.00	-0.06	-0.07	0.37
campaign	0.00	0.03	0.01	0.01	0.03	0.01	0.00	0.08	0.16	-0.12	0.04	-0.10	1.00	-0.04	-0.08	0.09	0.15	0.13	-0.01	0.13	0.14	0.07
pdays	0.02	0.13	0.04	0.04	0.07	0.01	0.00	0.10	0.21	0.33	0.01	0.07	-0.04	1.00	0.51	0.78	-0.23	-0.05	0.07	-0.26	-0.33	0.30
previous	0.02	0.14	0.05	0.05	0.10	0.02	0.00	0.21	0.28	0.50	0.01	0.04	-0.08	0.51	1.00	0.89	-0.42	-0.20	-0.05	-0.45	-0.50	0.25
poutcome	0.04	0.10	0.04	0.04	0.08	0.02	0.00	0.24	0.24	0.50	0.01	0.06	0.09	0.78	0.89	1.00	0.47	0.31	0.18	0.49	0.51	0.35
emp.var.rate	0.00	0.20	0.10	0.06	0.20	0.06	0.00	0.39	0.59	-0.85	0.03	-0.05	0.15	-0.23	-0.42	0.47	1.00	0.78	0.20	0.97	0.91	0.32
cons.price.idx	0.00	0.12	0.06	0.10	0.17	0.08	0.01	0.59	0.61	-0.46	0.02	-0.00	0.13	-0.05	-0.20	0.31	0.78	1.00	0.06	0.69	0.52	0.15
cons.conf.idx	0.13	0.15	0.06	0.11	0.02	0.03	0.01	0.25	0.61	-0.25	0.07	-0.00	-0.01	0.07	-0.05	0.18	0.20	0.06	1.00	0.28	0.10	0.06
euribor3m	0.01	0.20	0.11	0.05	0.19	0.06	0.00	0.40	0.58	-0.94	0.04	-0.06	0.13	-0.26	-0.45	0.49	0.97	0.69	0.28	1.00	0.95	0.33
nr.employed	-0.02	0.23	0.10	0.06	0.19	0.05	0.00	0.27	0.65	-0.96	0.03	-0.07	0.14	-0.33	-0.50	0.51	0.91	0.52	0.10	0.95	1.00	0.38
y	0.03	0.17	0.06	0.07	0.11	0.01	0.00	0.15	0.30	0.37	0.02	0.37	0.07	0.30	0.25	0.35	0.32	0.15	0.06	0.33	0.38	1.00

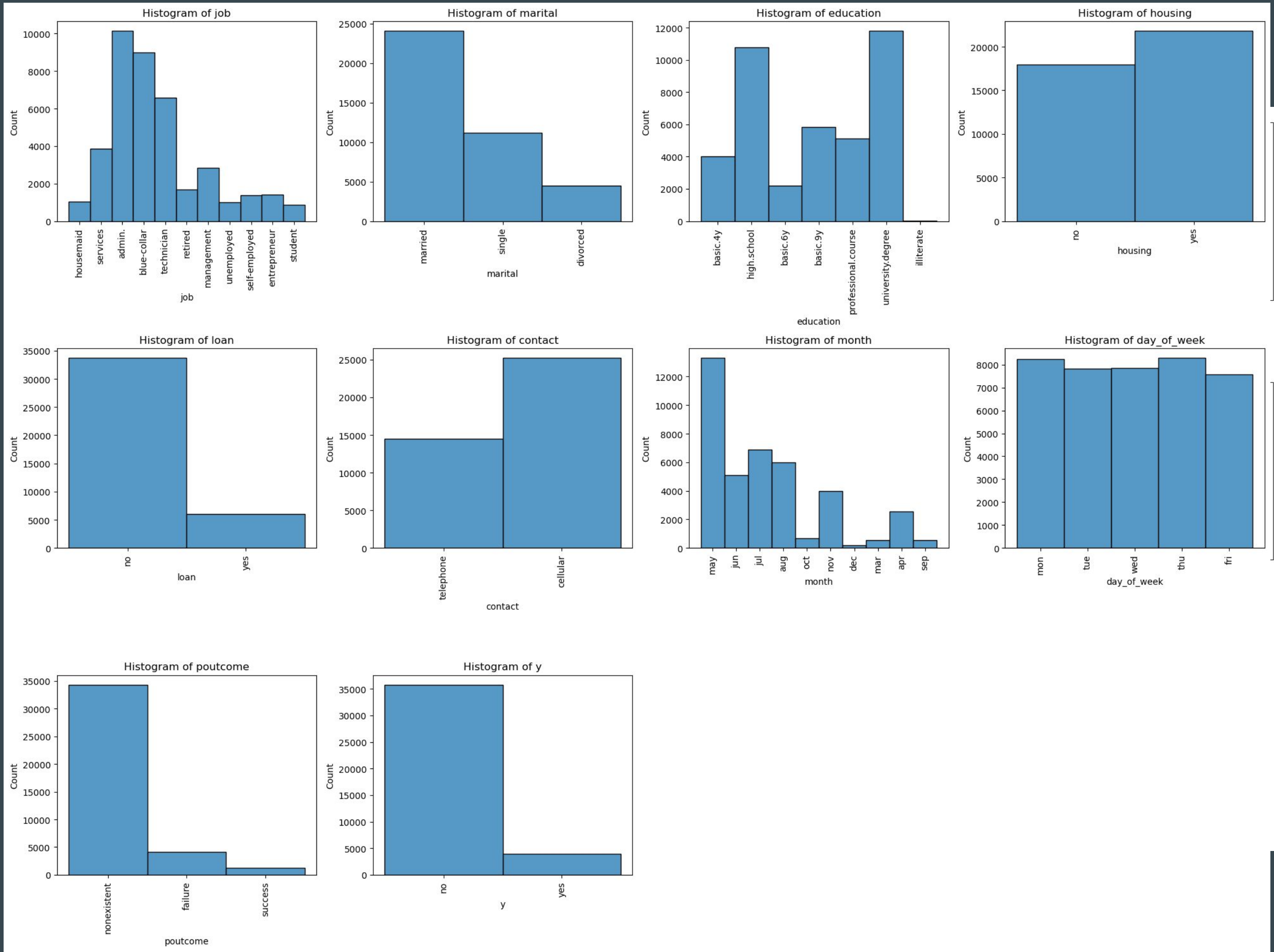
Numerical Features

To better visualize the range, spread, biases, etc in each of our features, we plotted histograms for the entire data set. The figures to the left capture the numerical data.



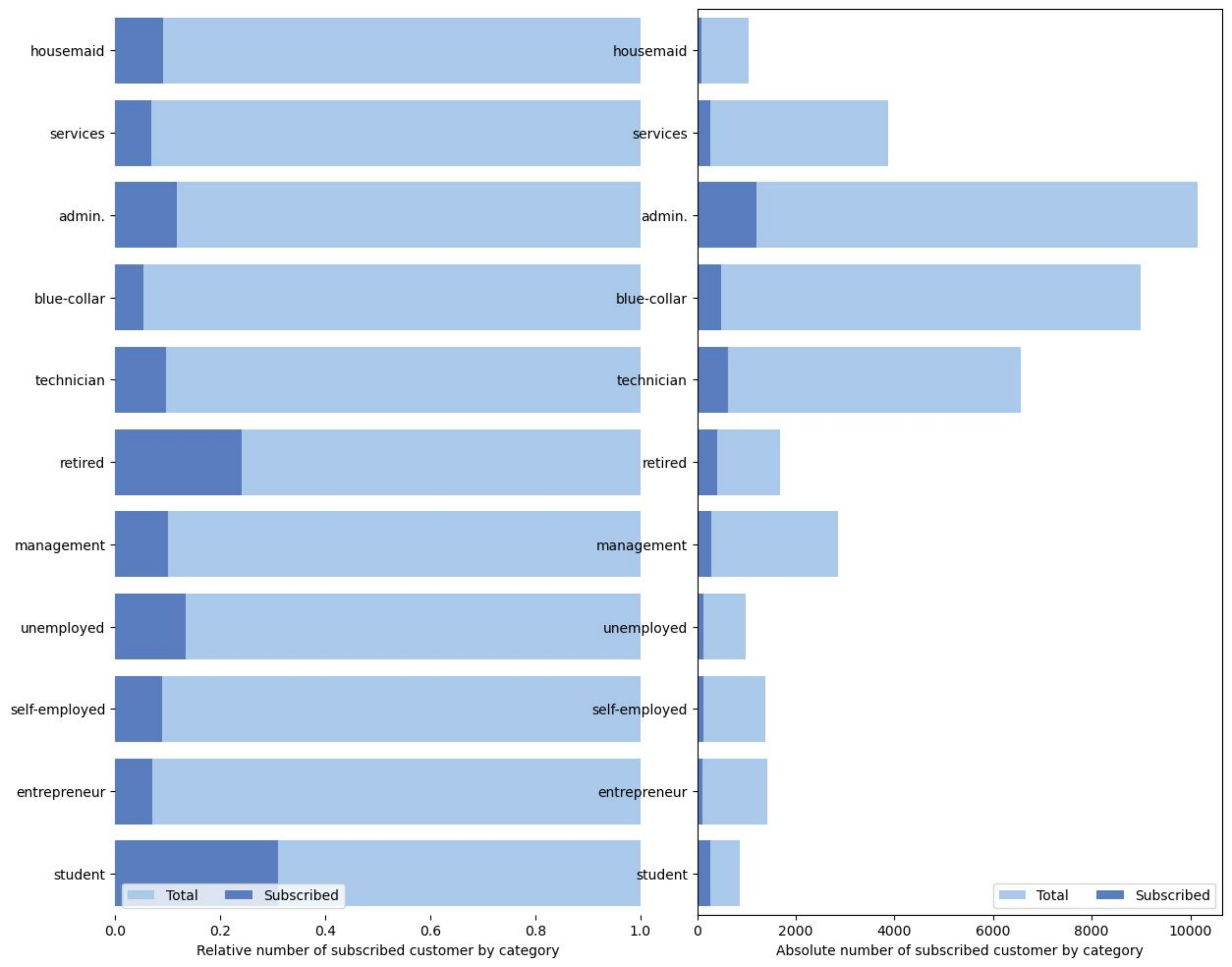
Categorical Features

Similar to the previous slide, these charts are histograms of the categorical features in the data set. Of particular note is the strong bias in the target variable ‘y’. The ‘yes’ cases only account for ~%10 of that feature. This information is important when it comes time to train the predictive model.



Relative vs Absolute Sales per Category

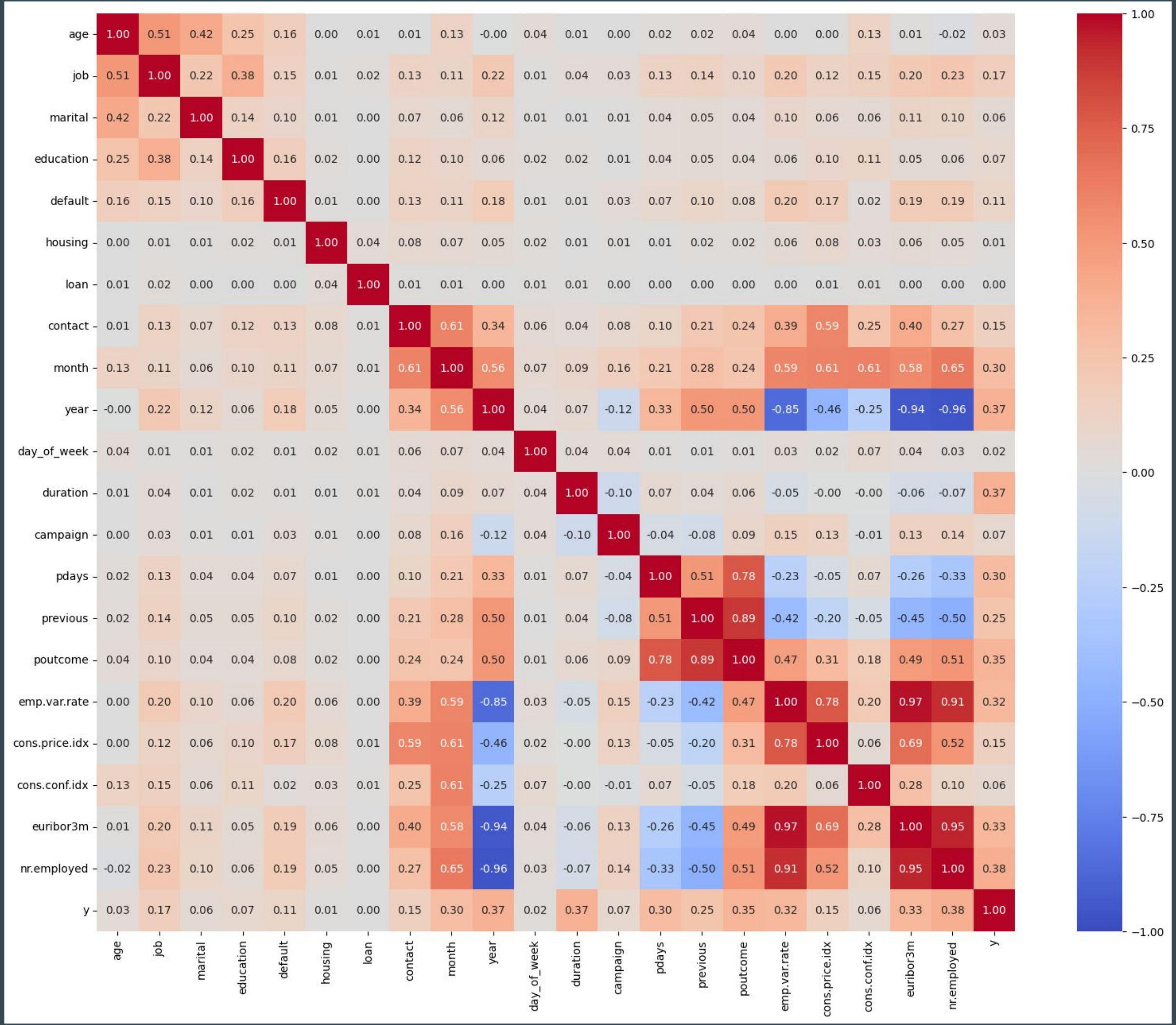
As we dig deeper into the categorical features, we can see those categories that maybe underrepresented in the data (for example, ‘student’), and yet yield high sales conversions relatively.



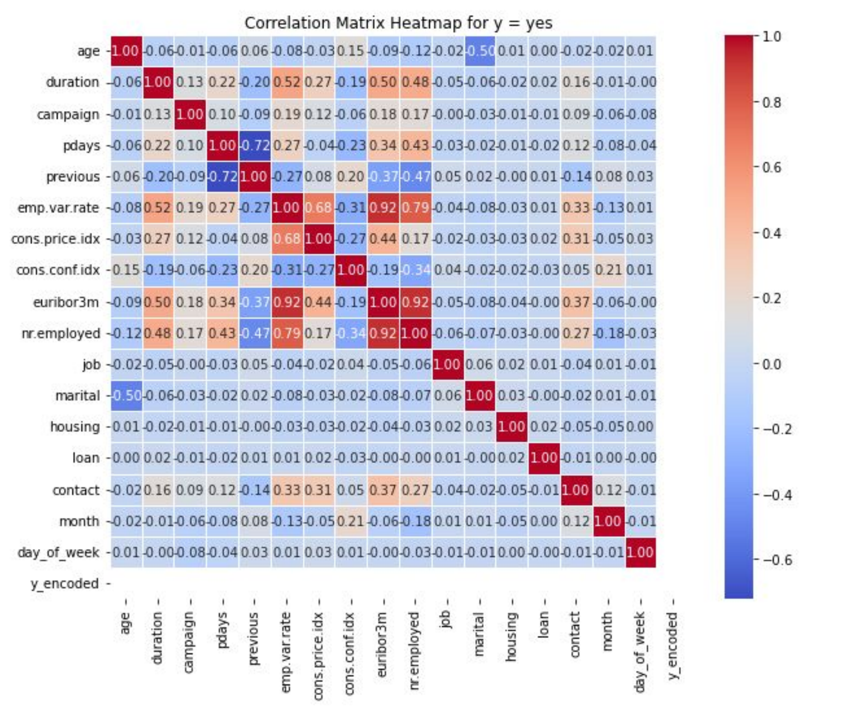
Recommendations

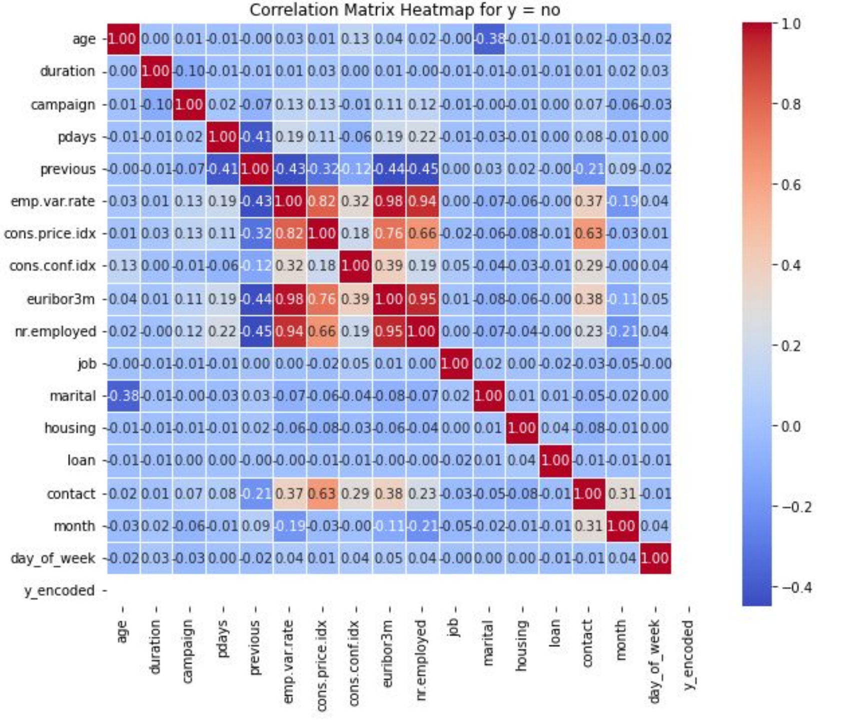
In the following section, we are going to choose the variables that are most correlated with target y and make recommendations of them based on the visualizations.

From the heatmap, we can see that most variables are not very closely correlated to target variable ‘y’ with values close to 0. The ones that are more closely correlated to y are ‘year’, ‘duration’, ‘pdays’, ‘poutcome’, etc. Such result makes somewhat sense because variables like ‘duration’ are highly influential to the subscription of the terms.



	age	job	marital	education	default	housing	loan	contact	month	year	day_of_week	duration	campaign	pdays	previous	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
age	1.00	0.51	0.42	0.25	0.16	0.00	0.01	0.01	0.13	-0.00	0.04	0.01	0.00	0.02	0.02	0.04	0.00	0.00	0.13	0.01	-0.02	0.03
job	0.51	1.00	0.22	0.38	0.15	0.01	0.02	0.13	0.11	0.22	0.01	0.04	0.03	0.13	0.14	0.10	0.20	0.12	0.15	0.20	0.23	0.17
marital	0.42	0.22	1.00	0.14	0.10	0.01	0.00	0.07	0.06	0.12	0.01	0.01	0.01	0.04	0.05	0.04	0.10	0.06	0.06	0.11	0.10	0.06
education	0.25	0.38	0.14	1.00	0.16	0.02	0.00	0.12	0.10	0.06	0.02	0.02	0.01	0.04	0.05	0.04	0.06	0.10	0.11	0.05	0.06	0.07
default	0.16	0.15	0.10	0.16	1.00	0.01	0.00	0.13	0.11	0.18	0.01	0.01	0.03	0.07	0.10	0.08	0.20	0.17	0.02	0.19	0.19	0.11
housing	0.00	0.01	0.01	0.02	0.01	1.00	0.04	0.08	0.07	0.05	0.02	0.01	0.01	0.01	0.02	0.02	0.06	0.08	0.03	0.06	0.05	0.01
loan	0.01	0.02	0.00	0.00	0.00	0.04	1.00	0.01	0.01	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00
contact	0.01	0.13	0.07	0.12	0.13	0.08	0.01	1.00	0.61	0.34	0.06	0.04	0.08	0.10	0.21	0.24	0.39	0.59	0.25	0.40	0.27	0.15
month	0.13	0.11	0.06	0.10	0.11	0.07	0.01	0.61	1.00	0.56	0.07	0.09	0.16	0.21	0.28	0.24	0.59	0.61	0.61	0.58	0.65	0.30
year	-0.00	0.22	0.12	0.06	0.18	0.05	0.00	0.34	0.56	1.00	0.04	0.07	-0.12	0.33	0.50	0.50	-0.85	-0.46	-0.25	-0.94	-0.96	0.37
day_of_week	0.04	0.01	0.01	0.02	0.01	0.02	0.01	0.06	0.07	0.04	1.00	0.04	0.04	0.01	0.01	0.01	0.03	0.02	0.07	0.04	0.03	0.02
duration	0.01	0.04	0.01	0.02	0.01	0.01	0.01	0.04	0.09	0.07	0.04	1.00	-0.10	0.07	0.04	0.06	-0.05	-0.00	-0.00	-0.06	-0.07	0.37
campaign	0.00	0.03	0.01	0.01	0.03	0.01	0.00	0.08	0.16	-0.12	0.04	-0.10	1.00	-0.04	-0.08	0.09	0.15	0.13	-0.01	0.13	0.14	0.07
pdays	0.02	0.13	0.04	0.04	0.07	0.01	0.00	0.10	0.21	0.33	0.01	0.07	-0.04	1.00	0.51	0.78	-0.23	-0.05	0.07	-0.26	-0.33	0.30
previous	0.02	0.14	0.05	0.05	0.10	0.02	0.00	0.21	0.28	0.50	0.01	0.04	-0.08	0.51	1.00	0.89	-0.42	-0.20	-0.05	-0.45	-0.50	0.25
poutcome	0.04	0.10	0.04	0.04	0.08	0.02	0.00	0.24	0.24	0.50	0.01	0.06	0.09	0.78	0.89	1.00	0.47	0.31	0.18	0.49	0.51	0.35
emp.var.rate	0.00	0.20	0.10	0.06	0.20	0.06	0.00	0.39	0.59	-0.85	0.03	-0.05	0.15	-0.23	-0.42	0.47	1.00	0.78	0.20	0.97	0.91	0.32
cons.price.idx	0.00	0.12	0.06	0.10	0.17	0.08	0.01	0.59	0.61	-0.46	0.02	-0.00	0.13	-0.05	-0.20	0.31	0.78	1.00	0.06	0.69	0.52	0.15
cons.conf.idx	0.13	0.15	0.06	0.11	0.02	0.03	0.01	0.25	0.61	-0.25	0.07	-0.00	-0.01	0.07	-0.05	0.18	0.20	0.06	1.00	0.28	0.10	0.06
euribor3m	0.01	0.20	0.11	0.05	0.19	0.06	0.00	0.40	0.58	-0.94	0.04	-0.06	0.13	-0.26	-0.45	0.49	0.97	0.69	0.28	1.00	0.95	0.33
nr.employed	-0.02	0.23	0.10	0.06	0.19	0.05	0.00	0.27	0.65	-0.96	0.03	-0.07	0.14	-0.33	-0.50	0.51	0.91	0.52	0.10	0.95	1.00	0.38
y	0.03	0.17	0.06	0.07	0.11	0.01	0.00	0.15	0.30	0.37	0.02	0.37	0.07	0.30	0.25	0.35	0.32	0.15	0.06	0.33	0.38	1.00

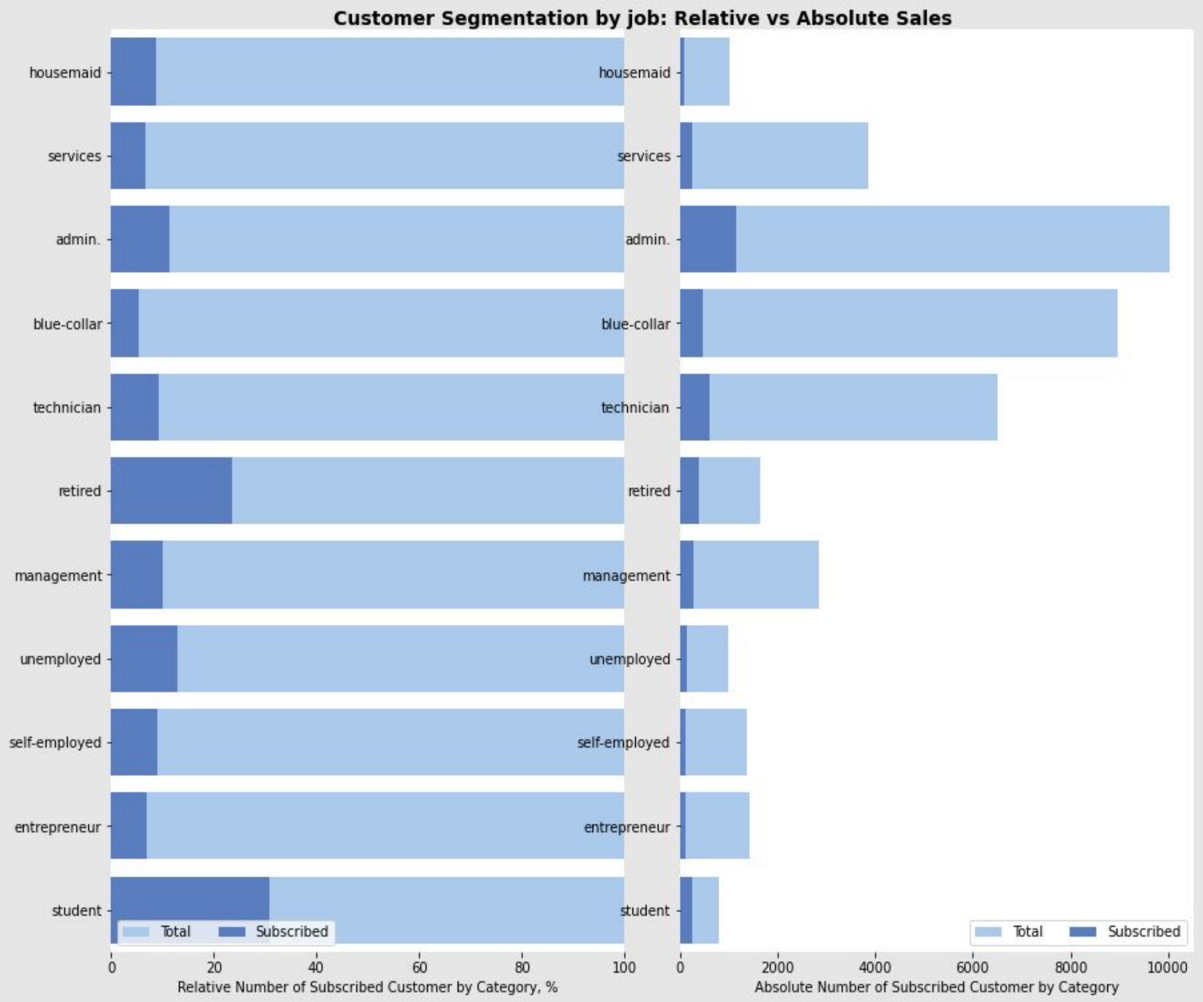




If we split into yes and no cases, we can see that the values close to 0 for both cases, indicating weak correlation to separated y.

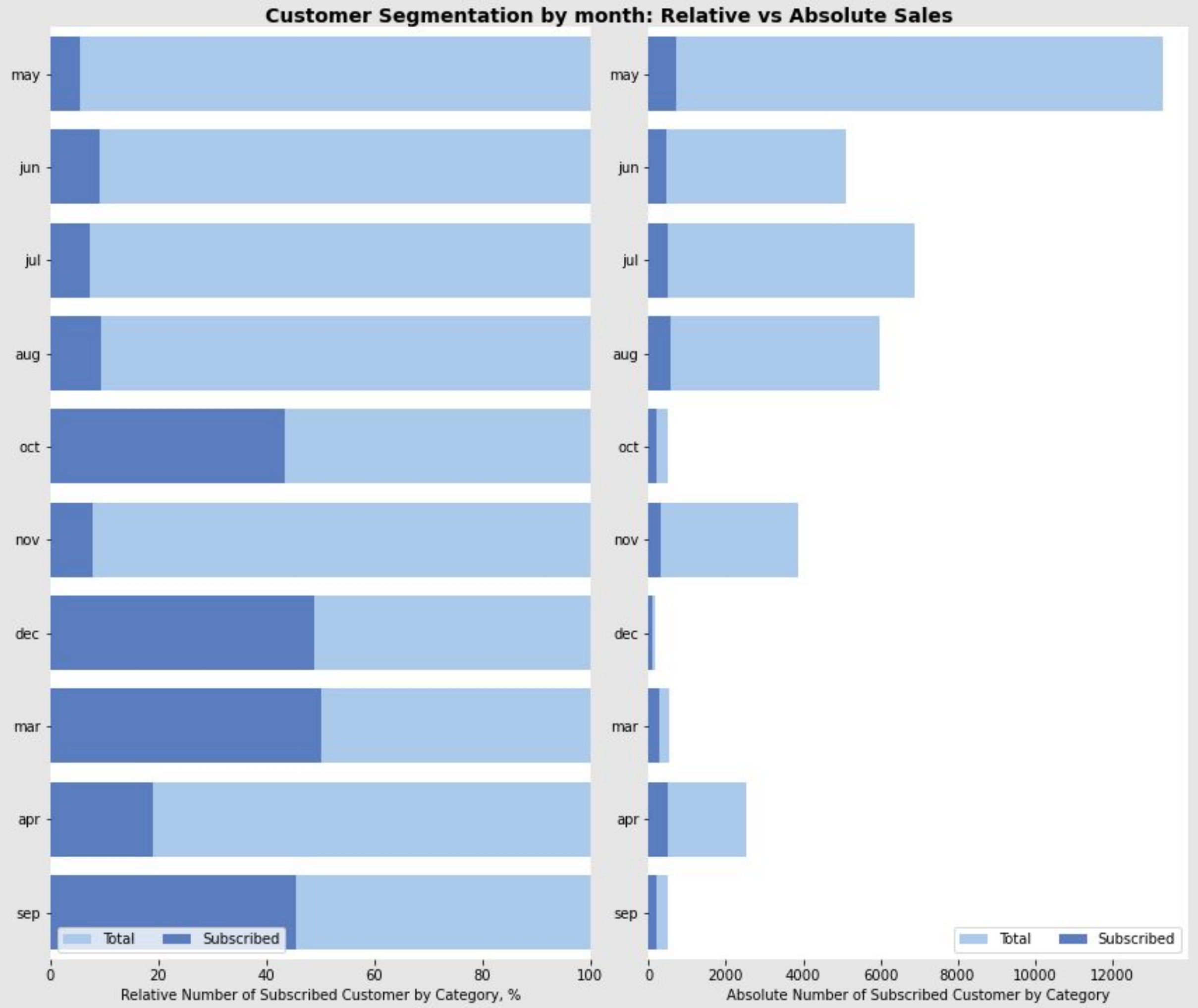
Improvements - Jobs

Based on distribution of categorical variables, we can find out that there are many improvements can be done. Specifically, students and retirees subscribe the terms more often than others relatively, but their absolute number of subscriptions are much fewer due to small number of phone calls; thus making more phone calls to them might lead to an increase of subscriptions



Improvements - Months

Another thing to improve would be calling months. As we can see, the relative number of subscriptions in March, October, December, etc., but their absolute number of subscriptions are much fewer due to small number of phone calls among these months; thus making continuous phone calls throughout the year will also lead to an increase of subscriptions



Recommended Models

As our problem is to predict whether a customer will purchase the term deposit or not, an ideal solution would be to employ a binary classification model that excels in making accurate predictions.

Below are some models that we believe will be best suited for this problem. We will also outline how we will deal with imbalance and testing our models.

Logistic Regression: This model is straightforward, quick to train, and its output is highly interpretable. However, it assumes a linear relationship between features and the log-odds of the target variable, which might not always hold true.

Decision Trees: These models are interpretable and capable of handling non-linear relationships. They also implicitly perform feature selection. But they are prone to overfitting, especially when dealing with a high number of features.

Random Forest: An ensemble method that enhances decision tree performance, Random Forests are less likely to overfit and are adept at managing non-linear relationships. However, they might be slightly more challenging to interpret and could require longer training time.

Gradient Boosting Machines (XGBoost, LightGBM): These models are highly accurate, can tackle non-linear relationships efficiently, and handle missing data. However, they require longer training time, are more complex to interpret, and need careful parameter tuning.

Handling Imbalance

Imbalance are a normal problem of binary classification models so we need to handles these are seemed fit.

Resampling: Adjust the class distribution by oversampling the minority class, undersampling the majority class, or using a combination of both. This helps create a more balanced dataset, but may lead to overfitting (oversampling) or loss of information (undersampling).

Evaluation of Model

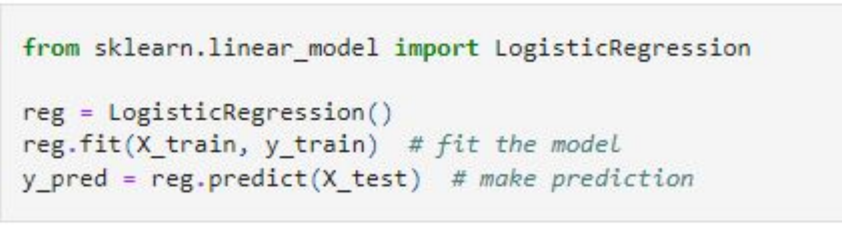
In order to assess the performance of our models, we will employ precision, recall, and F1-score as our evaluation metrics. Utilizing accuracy as a measure would not yield reliable results in the context of our current models, particularly when dealing with imbalanced datasets.

Logistic Regression Model

Next we are going to apply logistic regression to make prediction of variable ‘y’

Code

The code for logistic regression is straightforward; we first split the dataset into X and y, which y is the variable ‘y’ and X is the remaining, then divide each of them into 80-20 training-testing sets. For the model part, we can directly import logistic regression model from sklearn package and apply it on training sets; finally make prediction with the trained model on test set of X.

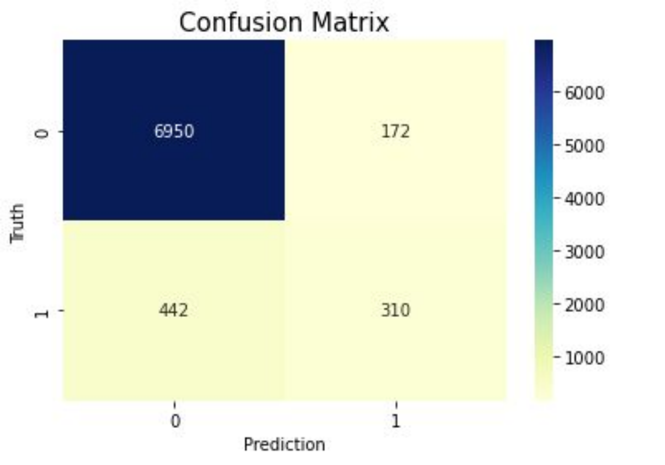


```python
from sklearn.linear_model import LogisticRegression

reg = LogisticRegression()
reg.fit(X_train, y_train)  # fit the model
y_pred = reg.predict(X_test)  # make prediction
```

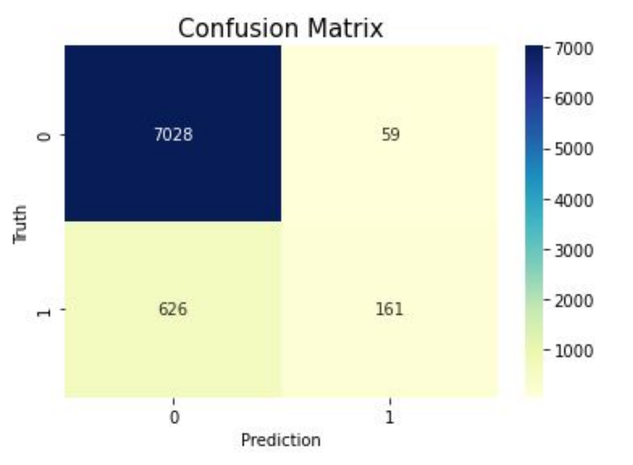
Confusion Matrices with and without ‘duration’

Since variable ‘y’ would cause impacts on the prediction, we are going to test the score twice with and without ‘duration’

With Duration



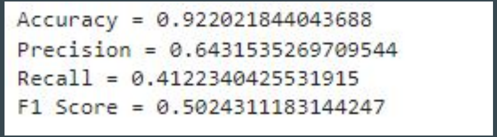
Without Duration



Scores

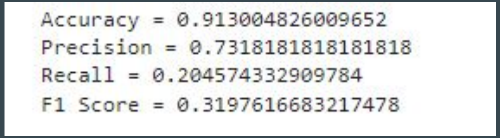
We can see that the accuracy scores for both cases are relatively high, which around 92% for duration included and 91% for duration excluded. But the other scores are very low. Specifically, for the included case, the precision is about 64%, which indicates that 64% of positive identifications are accurate. The recall is around 41%, which indicates the ratio of the number of samples correctly predicted as yes and the number of samples predicted as yes is 41%. The reason for such low score is that the number of yes subscription is very low, and the type II error occurs more frequently due to such imbalance. The f1 score is the mean between recall and precision, which is around 50%. The recall score is much lower in excluded case, which leads to lower f1 score.

With Duration



```
Accuracy = 0.922021844043688
Precision = 0.6431535269709544
Recall = 0.4122340425531915
F1 Score = 0.5024311183144247
```

Without Duration



```
Accuracy = 0.913004826009652
Precision = 0.7318181818181818
Recall = 0.204574332909784
F1 Score = 0.3197616683217478
```

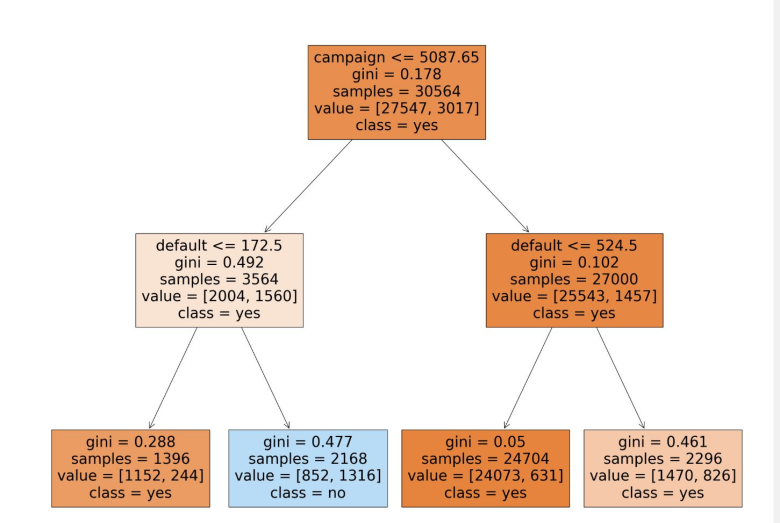
Decision Tree Model

Results:

Accuracy: 0.9203088196807119

F-1 Score: 0.5340474368783473

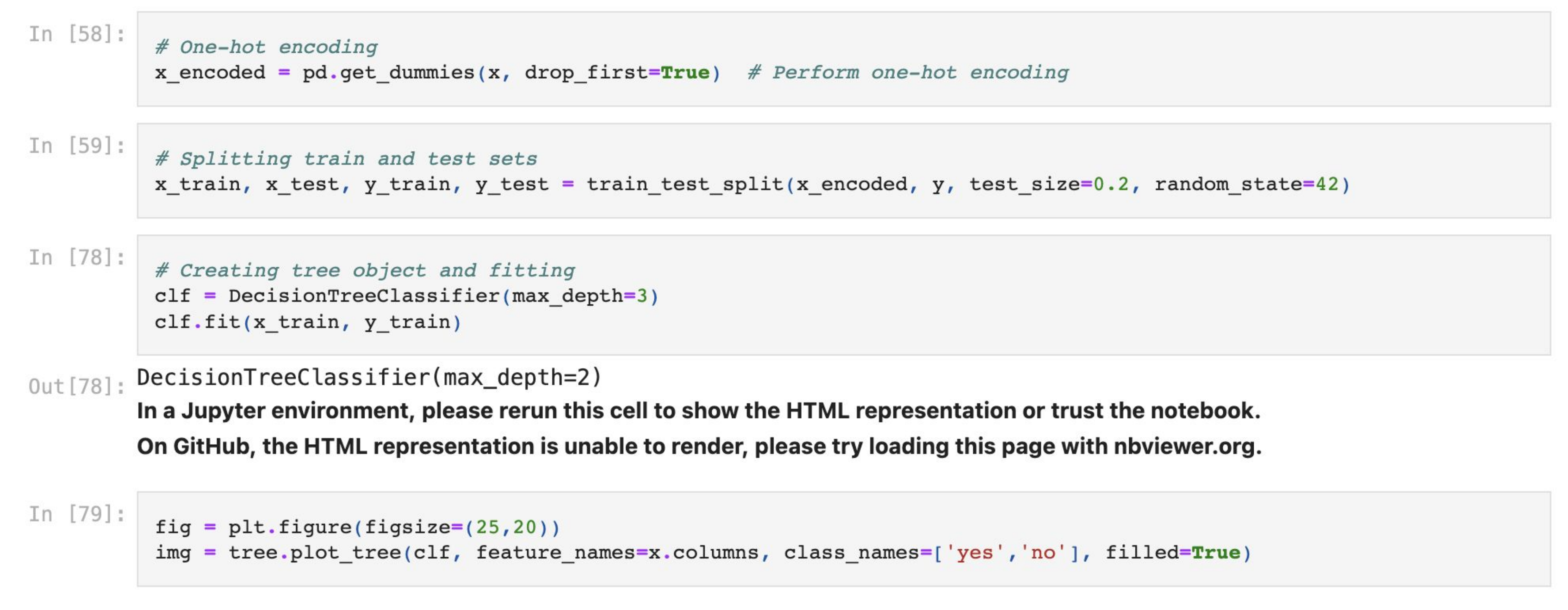
Precision: 0.6380255941499086



Context:

This result is to be expected due to the nature of decision trees. They are highly prone to noise and data with outliers. Which means that miniscule changes to the dataset can alter the entire decision tree and its node parameters. Thus, the accuracy score was decent, but the F-1 score and the Precision score was incredibly low.

Decision Tree Model Code (Reference)



In [58]:

```
# One-hot encoding
x_encoded = pd.get_dummies(x, drop_first=True)  # Perform one-hot encoding
```

In [59]:

```
# Splitting train and test sets
x_train, x_test, y_train, y_test = train_test_split(x_encoded, y, test_size=0.2, random_state=42)
```

In [78]:

```
# Creating tree object and fitting
clf = DecisionTreeClassifier(max_depth=3)
clf.fit(x_train, y_train)
```

Out[78]:

```
DecisionTreeClassifier(max_depth=2)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [79]:

```
fig = plt.figure(figsize=(25,20))
img = tree.plot_tree(clf, feature_names=x.columns, class_names=['yes','no'], filled=True)
```

Random Forest Classifier

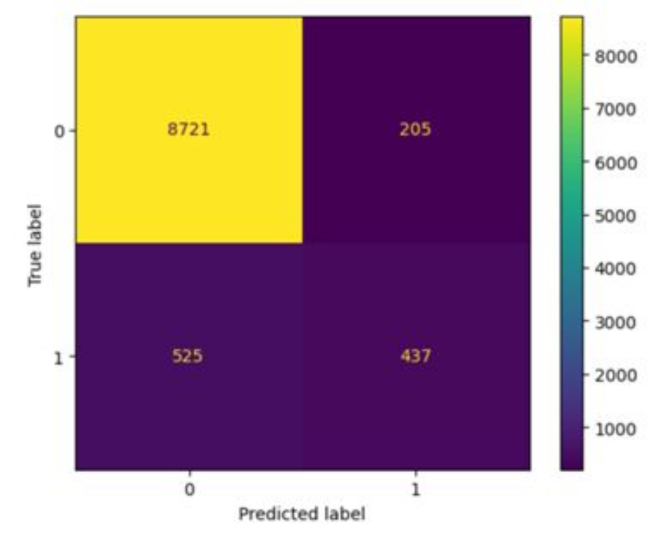
Results:

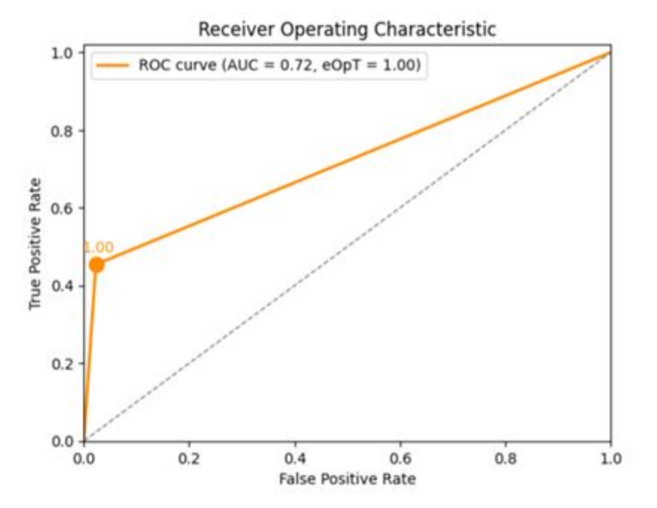
Accuracy: 0.926173139158576

f1 score: 0.5448877805486284

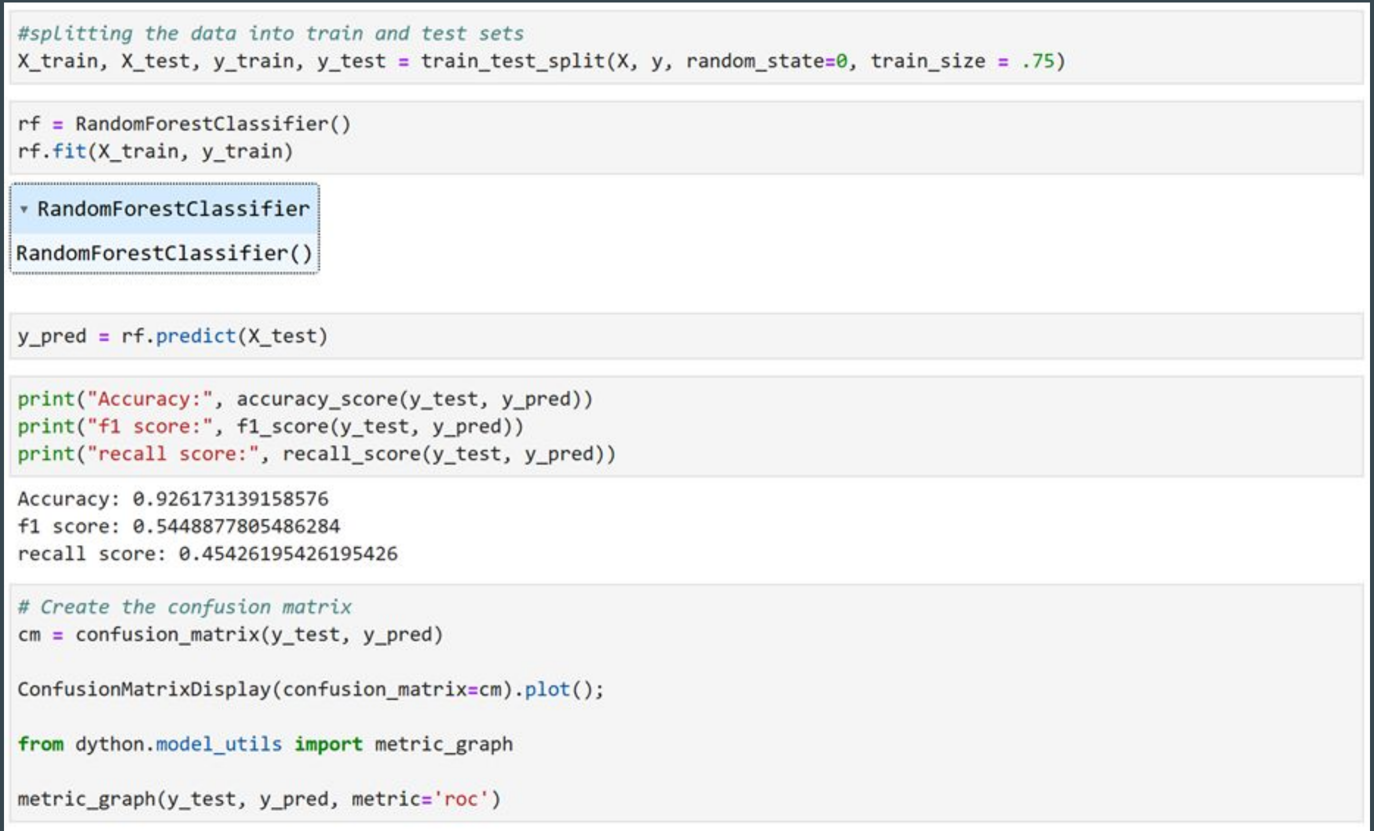
Recall score: 0.45426195426195426

As might be expected, the Random Forest Classifier produced similar results as the decision tree model.





Random Forest Classifier Model Code (Reference)



```python
#splitting the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, train_size = .75)
```

```python
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```python
y_pred = rf.predict(X_test)
```

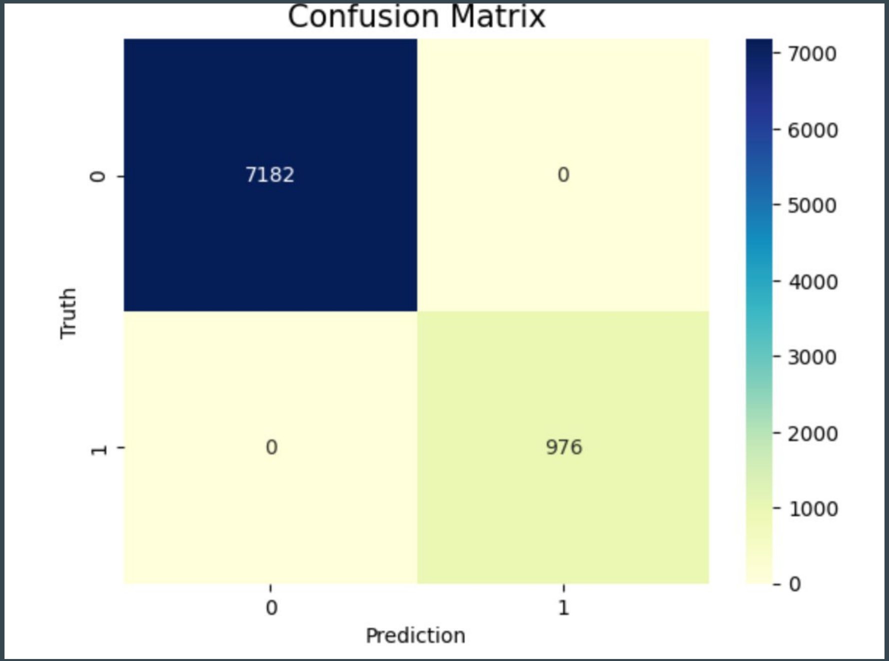
```python
print("Accuracy:", accuracy_score(y_test, y_pred))
print("f1 score:", f1_score(y_test, y_pred))
print("recall score:", recall_score(y_test, y_pred))
```

```
Accuracy: 0.926173139158576
f1 score: 0.5448877805486284
recall score: 0.45426195426195426
```

```python
# Create the confusion matrix
cm = confusion_matrix(y_test, y_pred)

ConfusionMatrixDisplay(confusion_matrix=cm).plot();

from dython.model_utils import metric_graph

metric_graph(y_test, y_pred, metric='roc')
```

Neural Network

Results:
Precision 1.0
Recall 1.0
F1-Score 1.0



The model has achieved a very high accuracy of 1.0 on the test data, which indicates that the model is performing extremely well and making correct predictions for all the samples in the test set

Conclusions

Through our analysis of the data and the four separate predictive models, we have developed a Neural Network Model that we are confident can make accurate predictions of future potential customers.

We have also made a few observations of the previous marketing campaigns effectiveness and have the following recommendations for future campaigns. These recommendations are based on the influence these factors have towards the over all sales conversion rates:

- Target more retirees and students
- Increase call volumes during the months of Sept - April instead of only targeting the Summer months.