Bank marketing

Norah sulaiman Alogla

Bank marketing analysis using machine learning

In this project I will try to perform exploratory data analysis and build a machine learning model to Predict if a client will subscribe (yes/no) to a term deposit in the bank.

Understanding the problem

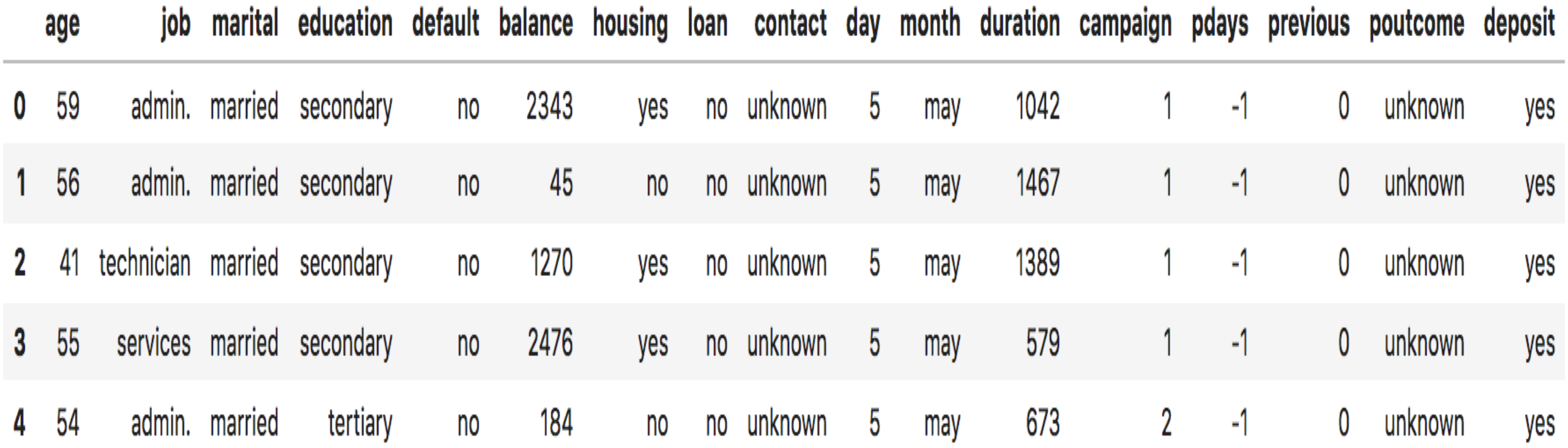
Increasing the number of client who deposit their money in the bank by analyzing their past marketing campaign data and recommending which customer to target. And this is defined as a classification problem.

Dataset contained 17 different features and 11162 clients. Features were both categorical and numerical. Target variable was binary (“yes” or “no”).

Data Exploration

The dataset gives us information about a marketing campaign of a financial institution in which you will have to analyze in order to find ways to look for future strategies in order to improve future marketing campaigns for the bank. So here we will import the dataset that describe.

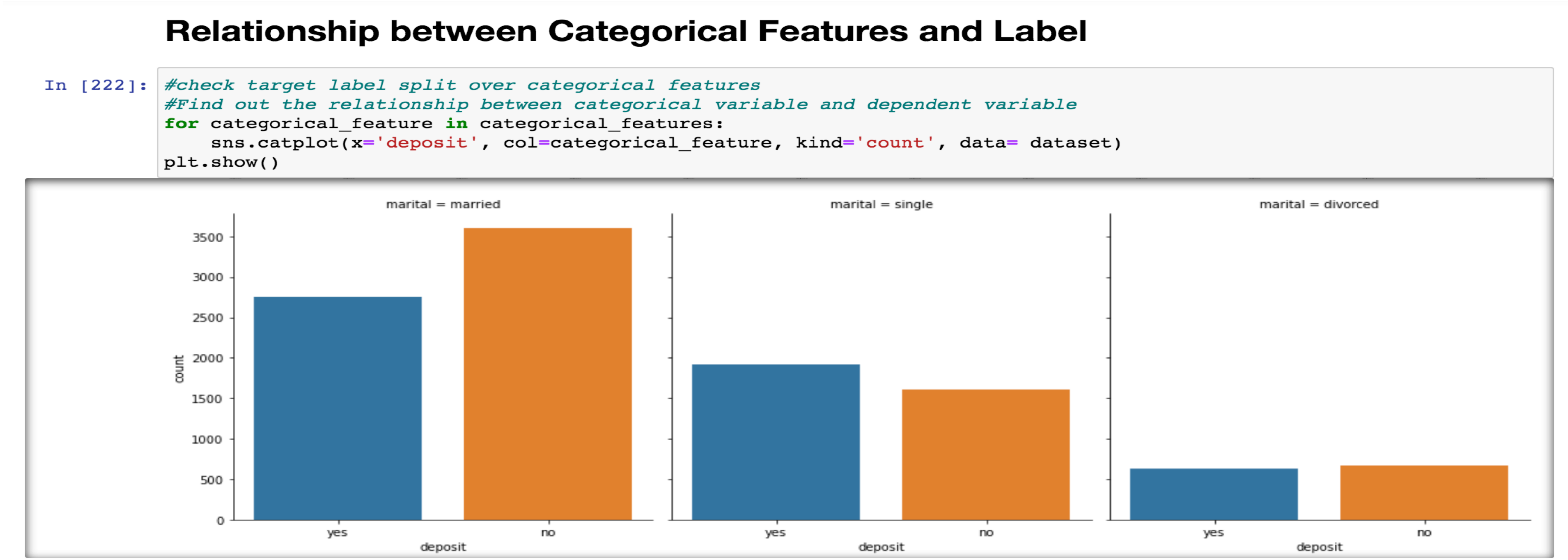
Features are



	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	deposit
0	59	admin.	married	secondary	no	2343	yes	no	unknown	5	may	1042	1	-1	0	unknown	yes
1	56	admin.	married	secondary	no	45	no	no	unknown	5	may	1467	1	-1	0	unknown	yes
2	41	technician	married	secondary	no	1270	yes	no	unknown	5	may	1389	1	-1	0	unknown	yes
3	55	services	married	secondary	no	2476	yes	no	unknown	5	may	579	1	-1	0	unknown	yes
4	54	admin.	married	tertiary	no	184	no	no	unknown	5	may	673	2	-1	0	unknown	yes

My Target or Label:

deposit | object | has the client subscribed a term deposit? (binary: 'yes’, ‘no’).



Relationship between Categorical Features and Label

```
In [222]: #check target label split over categorical features
          #Find out the relationship between categorical variable and dependent variable
          for categorical_feature in categorical_features:
              sns.catplot(x='deposit', col=categorical_feature, kind='count', data= dataset)
          plt.show()
```

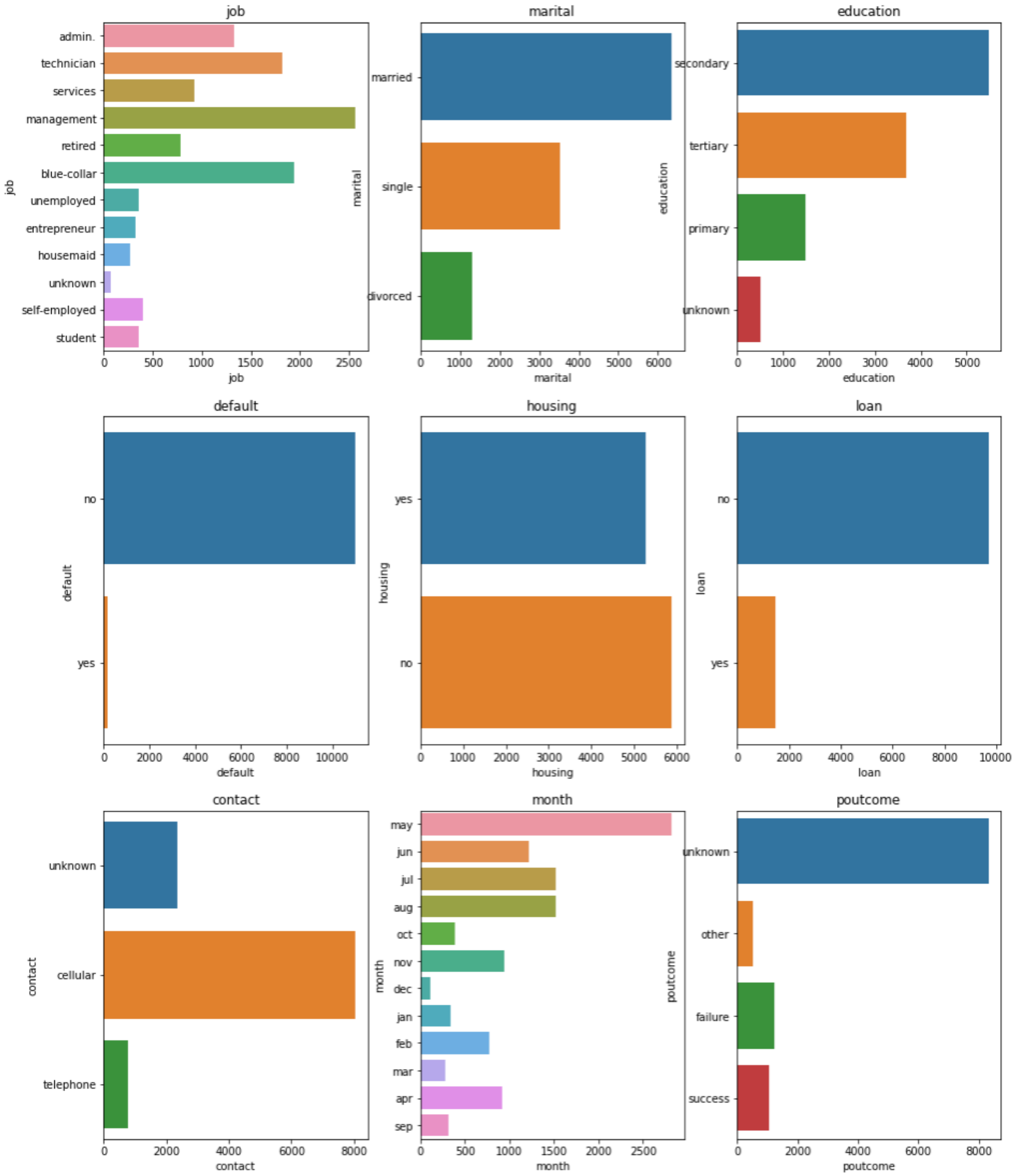
Data preparation

Clean the data

I will perfume these step to make sure everything is ready on my data analysis:

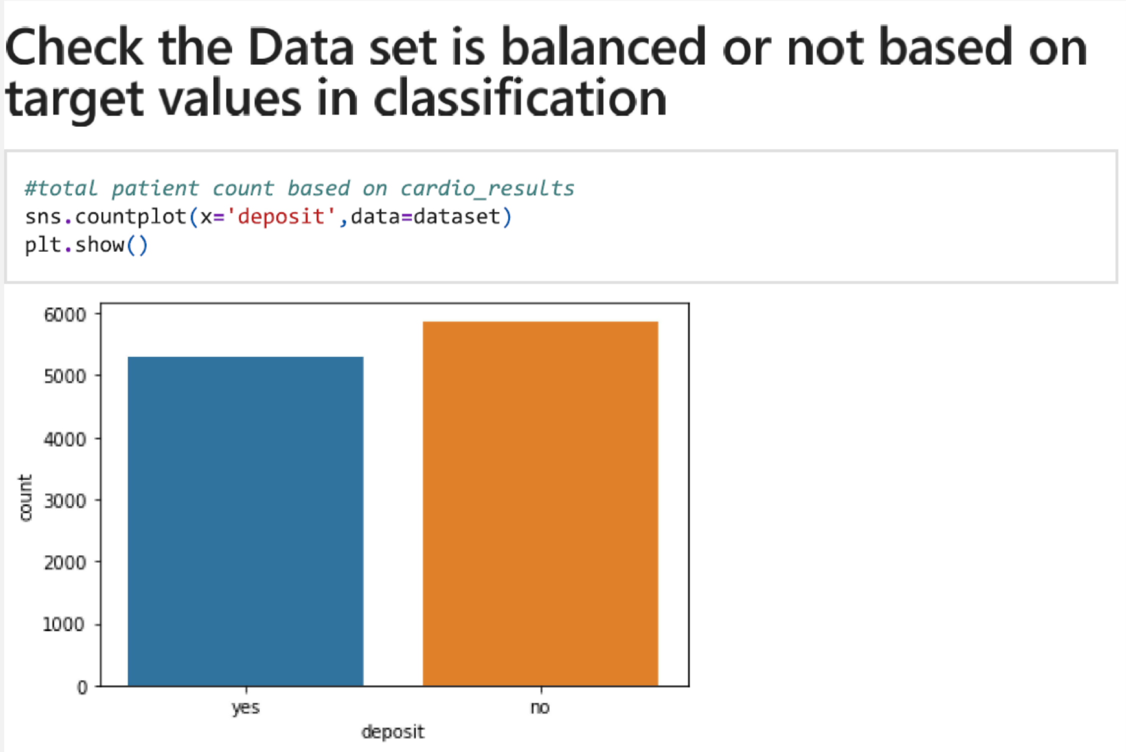
- Feature Engineering
- Drop unwanted Features
- Handle Missing Values
- Handle Categorical Features
- Handle Feature Scaling
- Remove Outliers

Categorical features



Numerical features = 7
Categorical features = 10

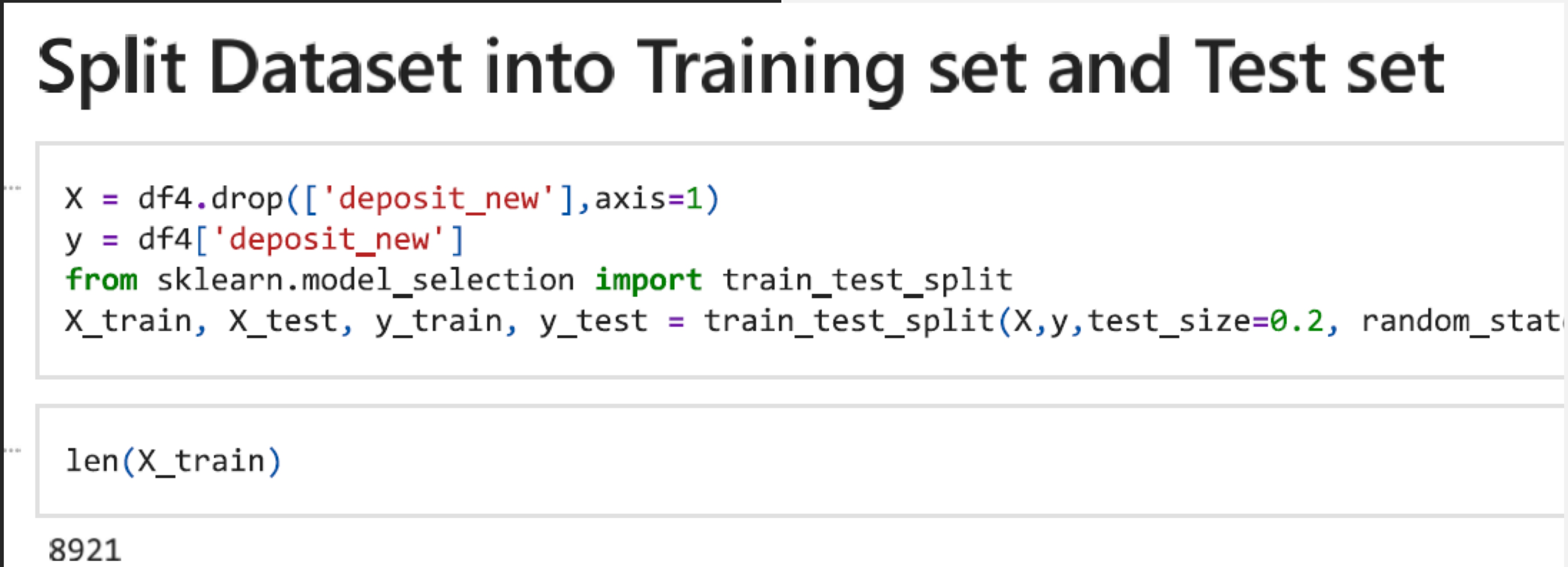
As data analysis I need to follow the step to prepare the data for machine learning to build a model that predict if the client will deposit in the bank or not.



Check the Data set is balanced or not based on target values in classification

```
#total patient count based on cardio_results
sns.countplot(x='deposit',data=dataset)
plt.show()
```

Check the dataset



Split Dataset into Training set and Test set

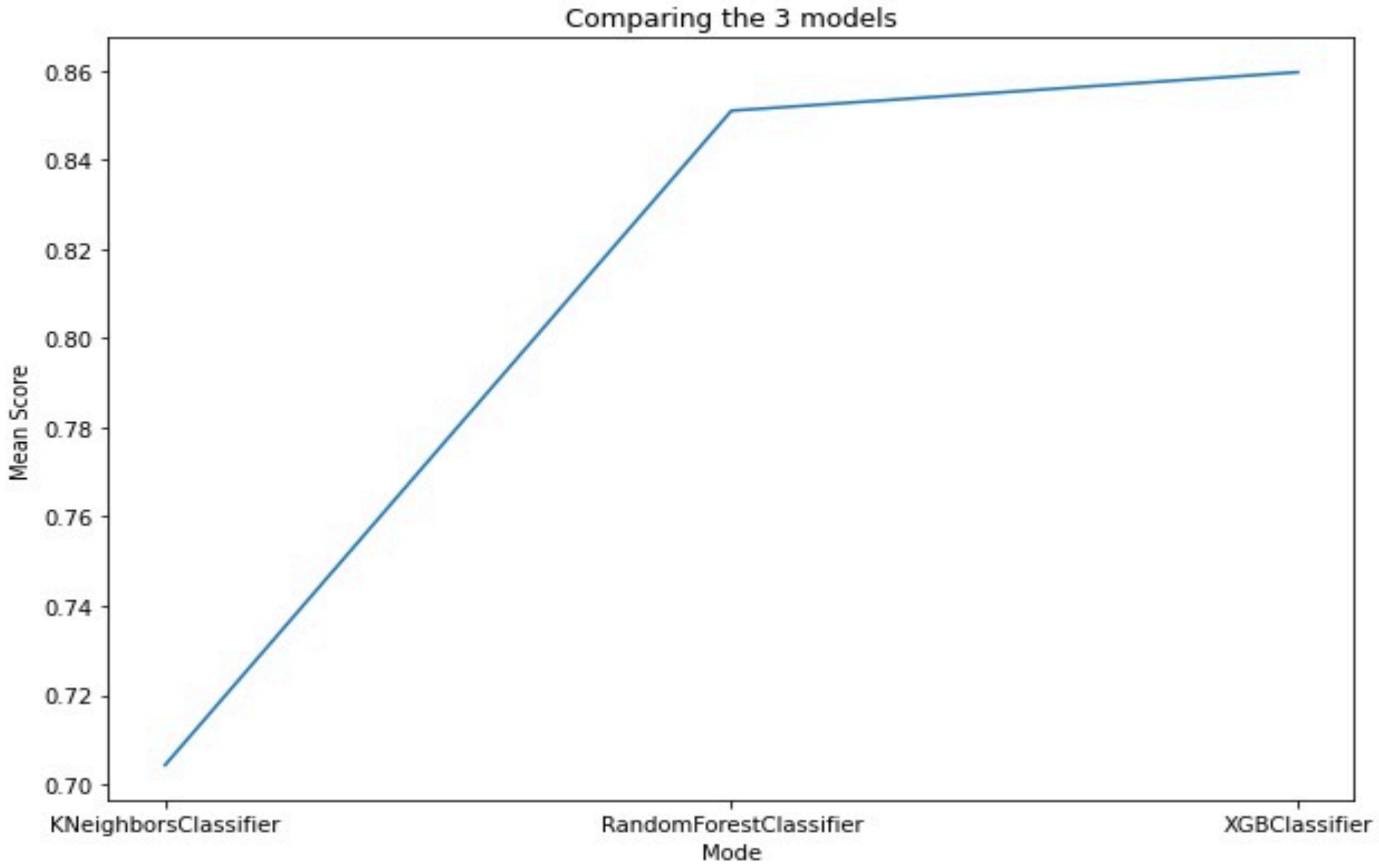
```
X = df4.drop(['deposit_new'],axis=1)
y = df4['deposit_new']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2, random_stat
```

```
len(X_train)
```

```
8921
```

I need to split the data in order to test and train the model

model selection



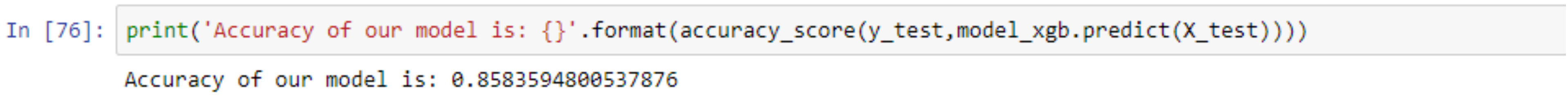
For this classification project, We have many different models like a Logistic Regression, Decision Tree, Random Forest classifier, XGBclassifier and many other models.

I used 3 models which are KNN, Random Forest classifier, and XGB Classifier.

From the results I found that: XGB Classifier has the best performance on the data and achieve the best accuracy.

model building

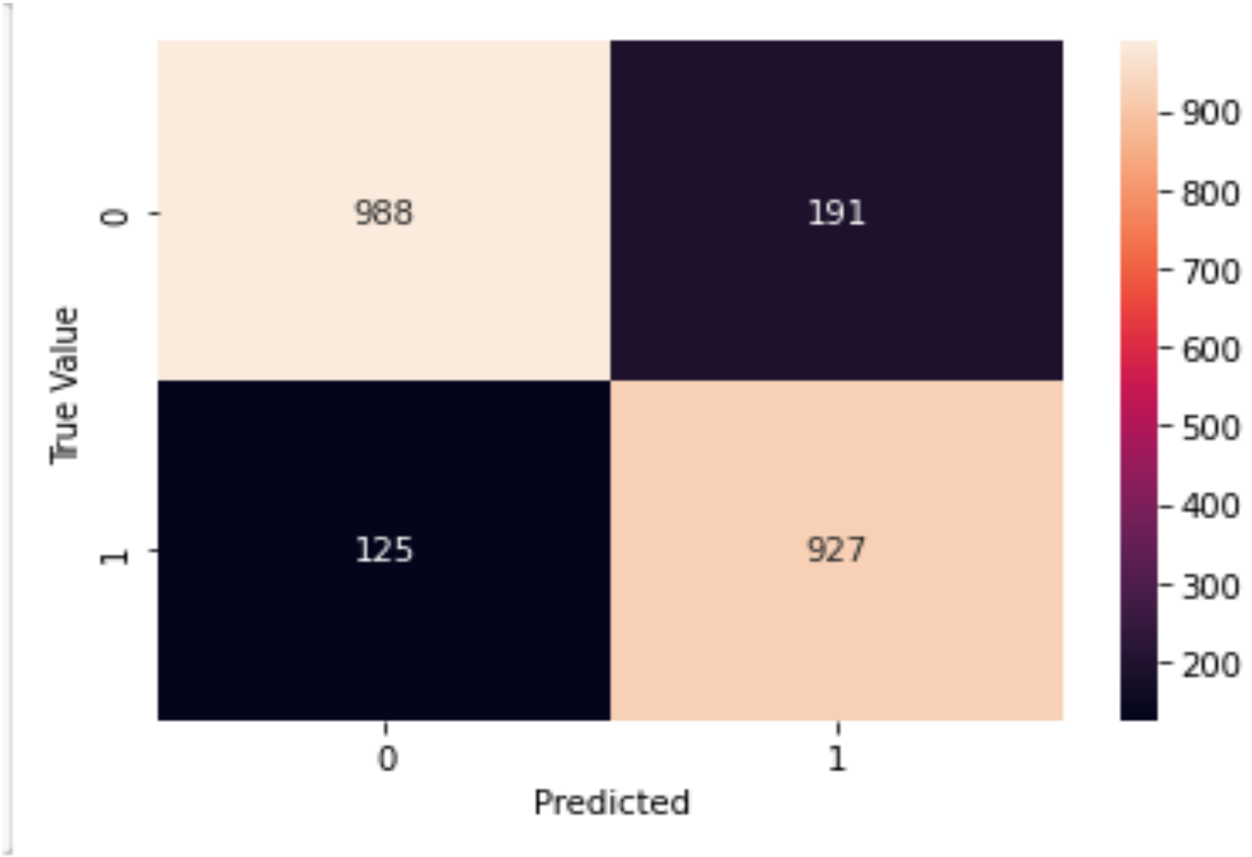
- <u>For the Evaluation</u>:
- I used many different methods to be sure that our model performs well on the data.
- I used the Accuracy matrix, Confusion Matrix, and the ROC Curve.
- The model achieved 0.858 accuracy which is acceptable.



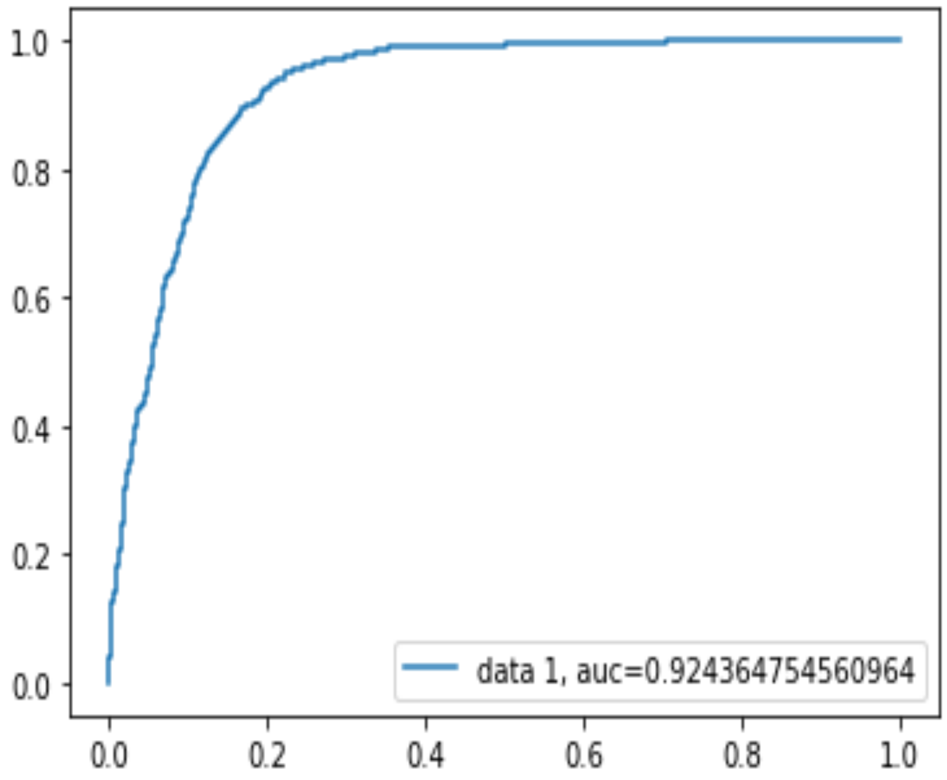
```
In [76]: print('Accuracy of our model is: {}'.format(accuracy_score(y_test,model_xgb.predict(X_test))))

         Accuracy of our model is: 0.8583594800537876
```

- For the Confusion Matrix
- It describes the number of True Positives and True Negatives comparing to the number of False Positives and False Negatives.
- Our goal is to maximize the number of trues and minimize the number of false.
- 0 means no deposit and
- 1 means deposit



- For the ROC Curve:
- People usually called it as the Area Under Curve(AUC).
- It describes the relation between the number of true positives on the y axis and the number of false positive on the x axis.
- Our goal is to maximize the number of true positive and minimize the number of false positives.
- Or we can say we want to maximize the area under the curve.
- The max value for the (AUC) equals 1.
- Our model has successfully achieved AUC equals 0.92.



CONCLUSION

- For Exploring the dataset: I found that our dataset is clean and doesn’t contain any missing values.
- For data preprocessing we apply many changes to the data to make it ready and pass it to the model ex: drop the past days that have negative values.
- The third step was splitting the data into training and testing data.
- The final step was building the model I tried many models and I choose the XGB classifier as our final model.
- From the results we found the number of the clients who not deposit is more than the clients who made deposits.
- I suggest to make more campaigns to encourage clients to make more deposits in our bank to increase the income.

Thank you for listening