Marketing Campaign Prediction

Reddy Charan Pulivendula

Pandya Harsh Maheshkumar

Sheta Rushank Ghanshyam

Abstract

- On average a company spends 4% to 25% of their net revenue into Marketing campaigns and Advertisements
- While some companies like Salesforce and Tableau spent around 50% of their revenue in marketing campaigns
- Targeted marketing is used to get good returns from this investment which aims at identifying groups of customers which are highly likely to become future customers
- Customer segmentation allows us to group customers based on demographics, geography, interests, user activity, etc.

Problem Statement

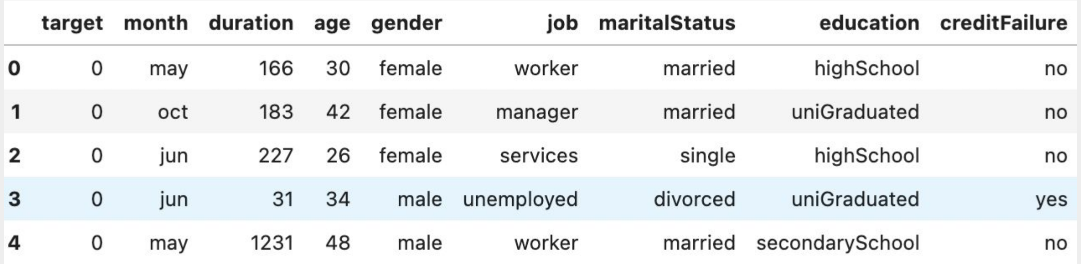
- Marketing campaign analysis allows companies to find customers which are highly likely to become future customer
- Using historical data from previous marketing campaigns to predict the response of a new customer applying predictive modeling
- Use customer demographics like age, income, marital status, etc. to classify if a new customer will respond positively or negatively to the campaigns

Related Work

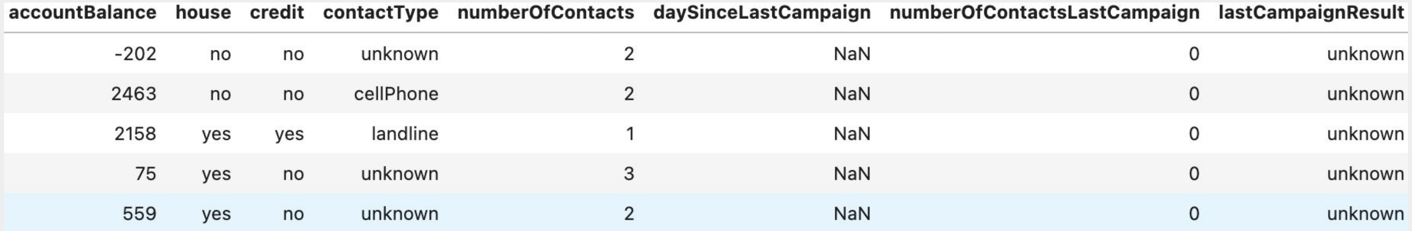
Performed steps like:

- EDA on Data Set
- Correlation Analysis
- Handling Missing Data
- Feature Engineering
- Binary Classification

Data

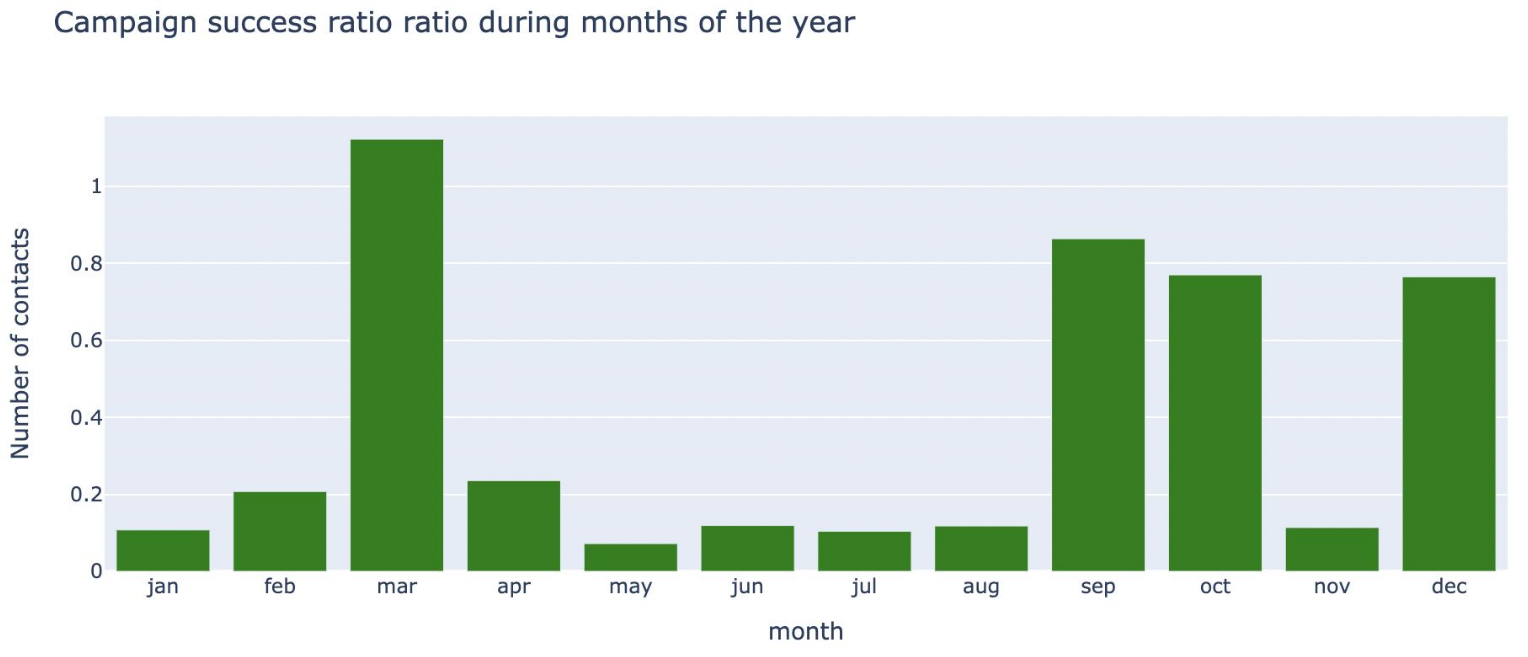


	target	month	duration	age	gender	job	maritalStatus	education	creditFailure
0	0	may	166	30	female	worker	married	highSchool	no
1	0	oct	183	42	female	manager	married	uniGraduated	no
2	0	jun	227	26	female	services	single	highSchool	no
3	0	jun	31	34	male	unemployed	divorced	uniGraduated	yes
4	0	may	1231	48	male	worker	married	secondarySchool	no

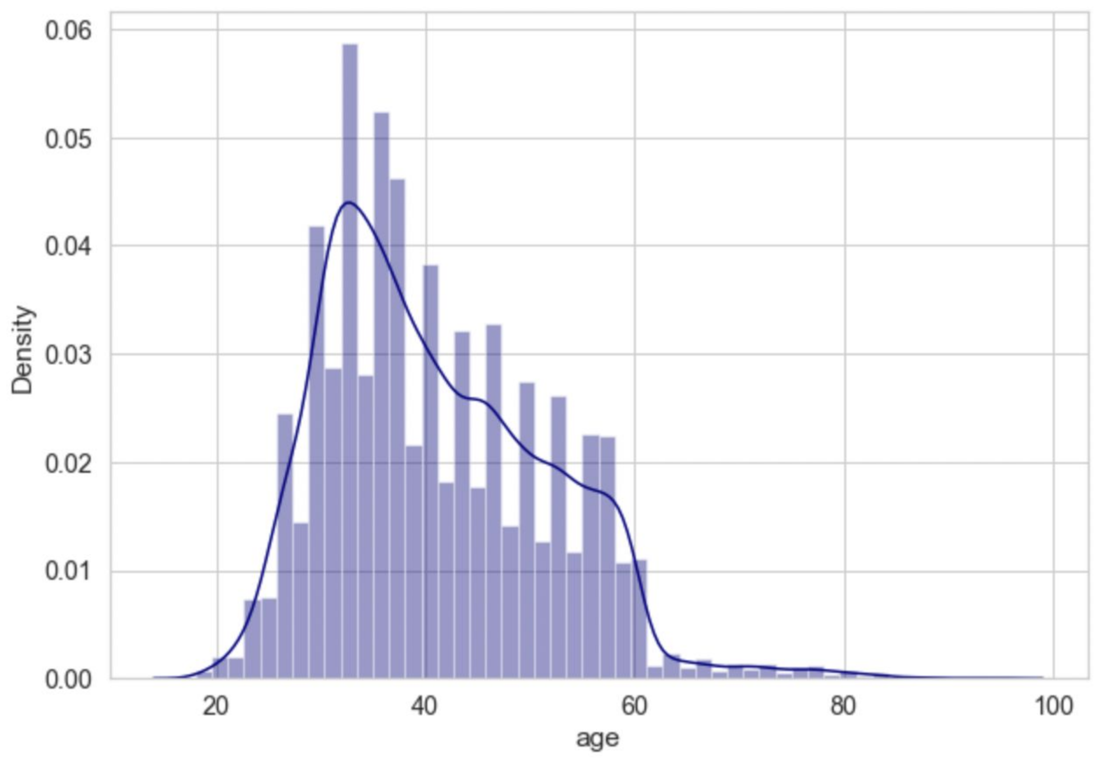


accountBalance	house	credit	contactType	numberOfContacts	daySinceLastCampaign	numberOfContactsLastCampaign	lastCampaignResult
-202	no	no	unknown	2	NaN	0	unknown
2463	no	no	cellPhone	2	NaN	0	unknown
2158	yes	yes	landline	1	NaN	0	unknown
75	yes	no	unknown	3	NaN	0	unknown
559	yes	no	unknown	2	NaN	0	unknown

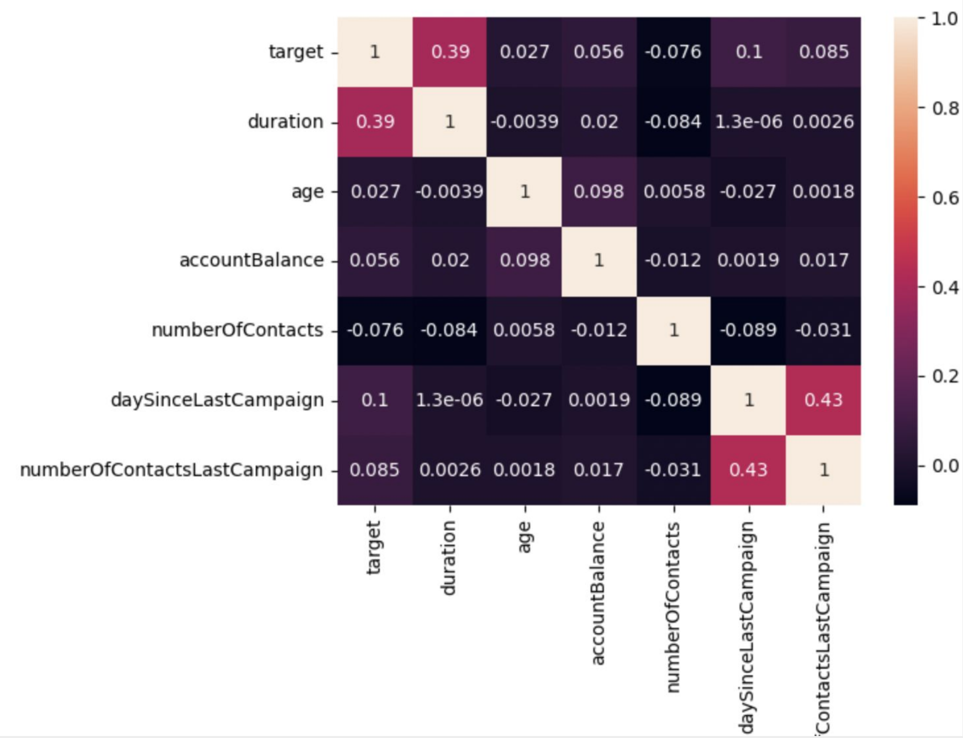
Univariate Analysis



Univariate Analysis

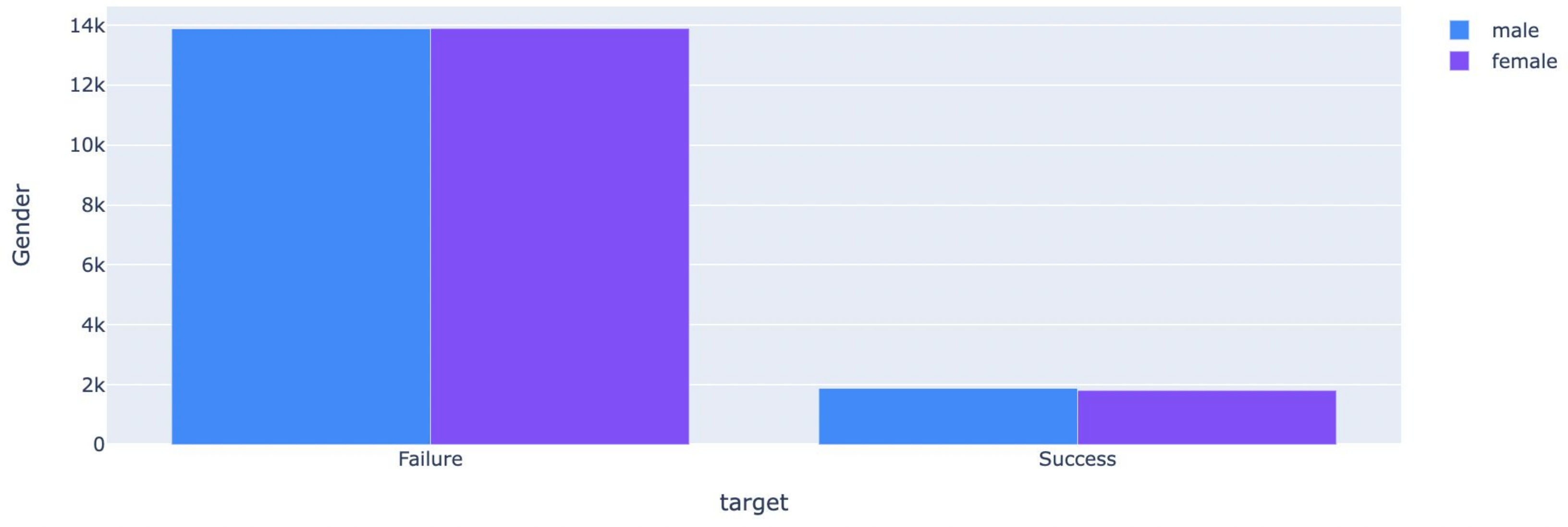


Correlation Analysis

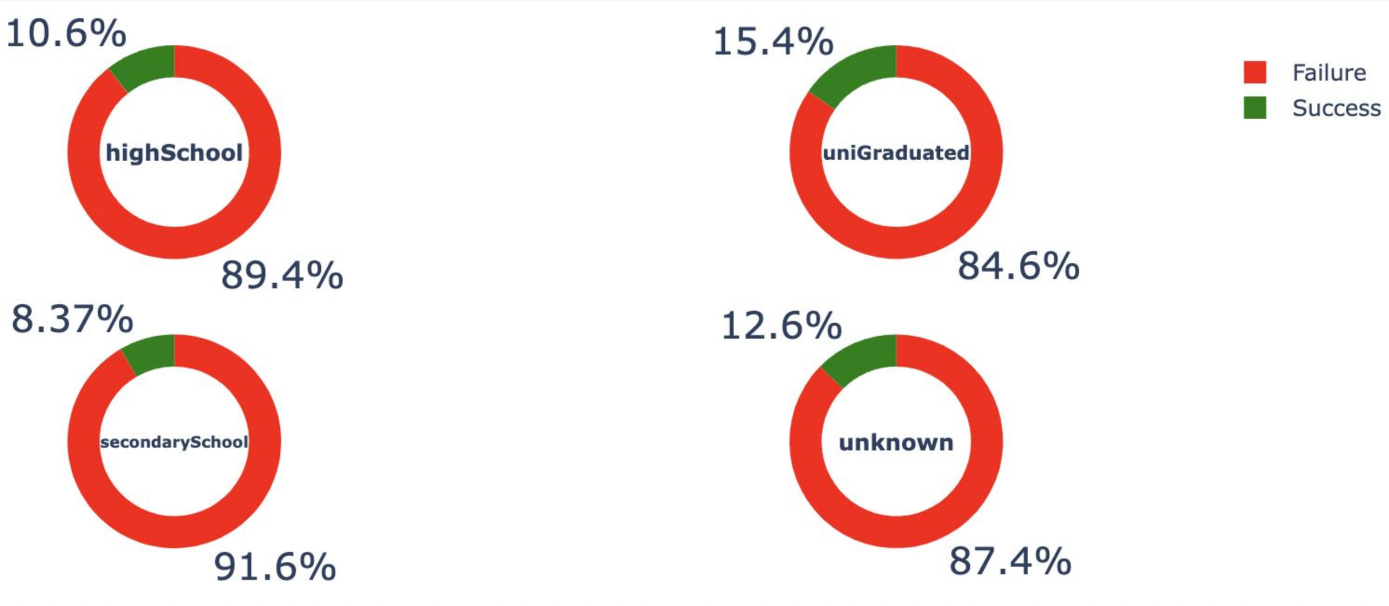


	target	duration	age	accountBalance	numberOfContacts	daySinceLastCampaign	ContactsLastCampaign
target	1	0.39	0.027	0.056	-0.076	0.1	0.085
duration	0.39	1	-0.0039	0.02	-0.084	1.3e-06	0.0026
age	0.027	-0.0039	1	0.098	0.0058	-0.027	0.0018
accountBalance	0.056	0.02	0.098	1	-0.012	0.0019	0.017
numberOfContacts	-0.076	-0.084	0.0058	-0.012	1	-0.089	-0.031
daySinceLastCampaign	0.1	1.3e-06	-0.027	0.0019	-0.089	1	0.43
numberOfContactsLastCampaign	0.085	0.0026	0.0018	0.017	-0.031	0.43	1

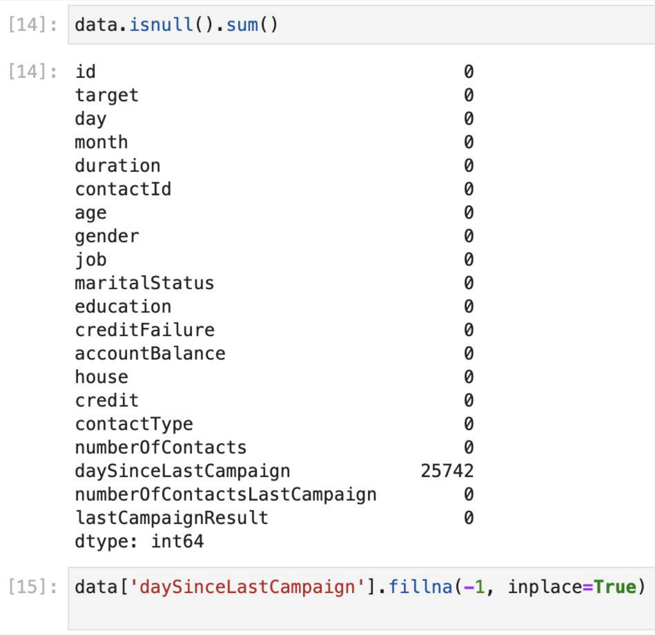
Bivariate Anaysis



Bivariate Analysis



Handling null values



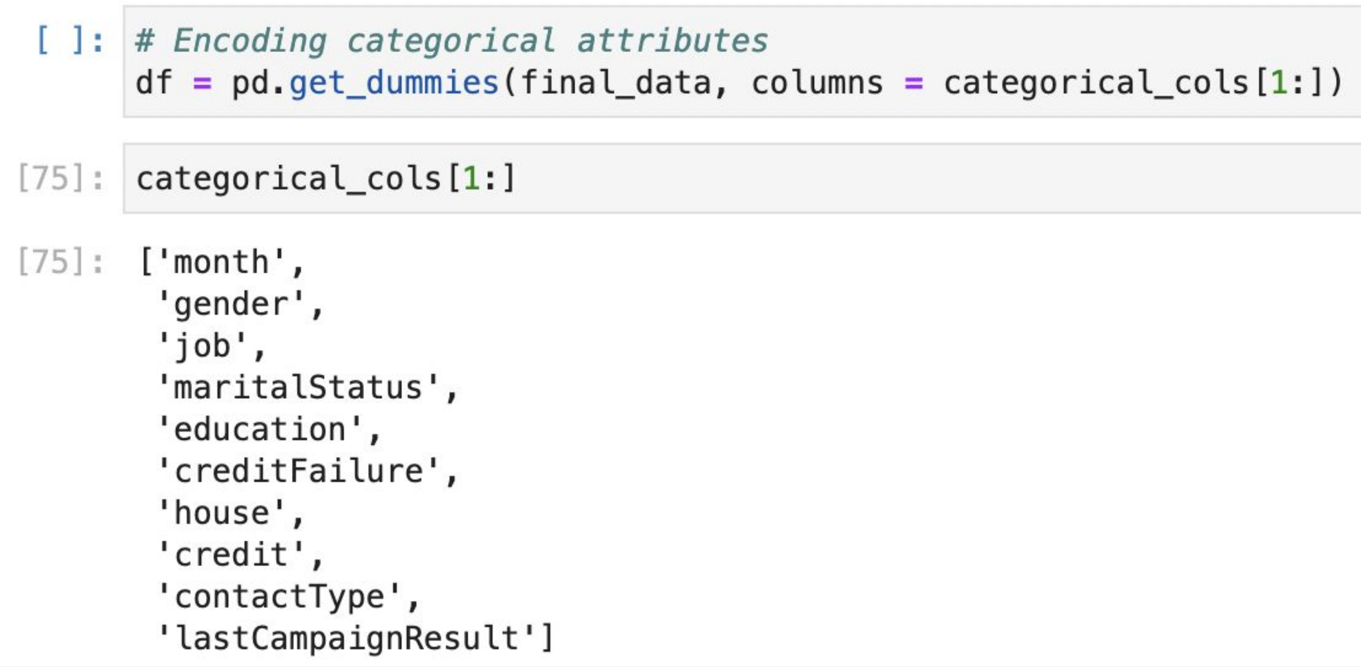
```
[14]: data.isnull().sum()
```

```
[14]: id                                 0
      target                             0
      day                                0
      month                              0
      duration                           0
      contactId                          0
      age                                0
      gender                             0
      job                                0
      maritalStatus                      0
      education                          0
      creditFailure                      0
      accountBalance                     0
      house                              0
      credit                             0
      contactType                        0
      numberOfContacts                   0
      daySinceLastCampaign           25742
      numberOfContactsLastCampaign       0
      lastCampaignResult                 0
      dtype: int64
```

```
[15]: data['daySinceLastCampaign'].fillna(-1, inplace=True)
```

Replacing null values with -1, which implies that the customer did not respond positively in previous campaign

Data pre-processing

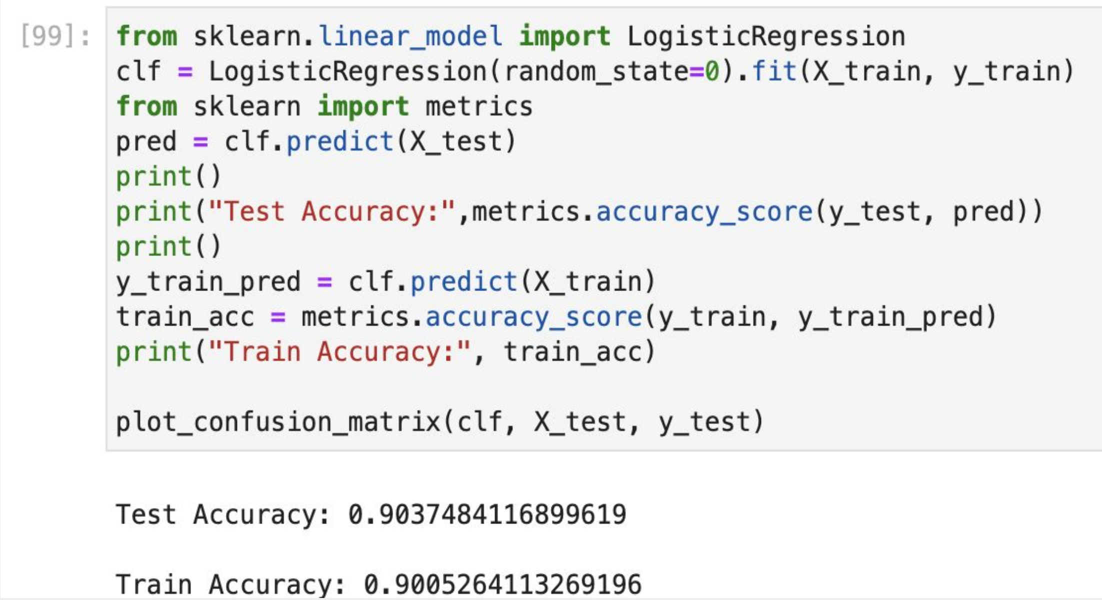


```
[ ]: # Encoding categorical attributes
     df = pd.get_dummies(final_data, columns = categorical_cols[1:])
```

```
[75]: categorical_cols[1:]
```

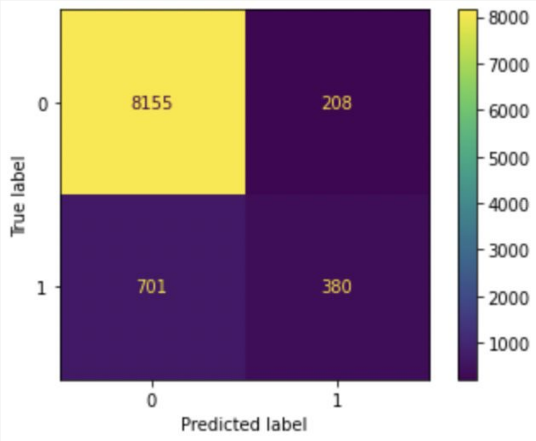
```
[75]: ['month',
       'gender',
       'job',
       'maritalStatus',
       'education',
       'creditFailure',
       'house',
       'credit',
       'contactType',
       'lastCampaignResult']
```

Logistic Regression

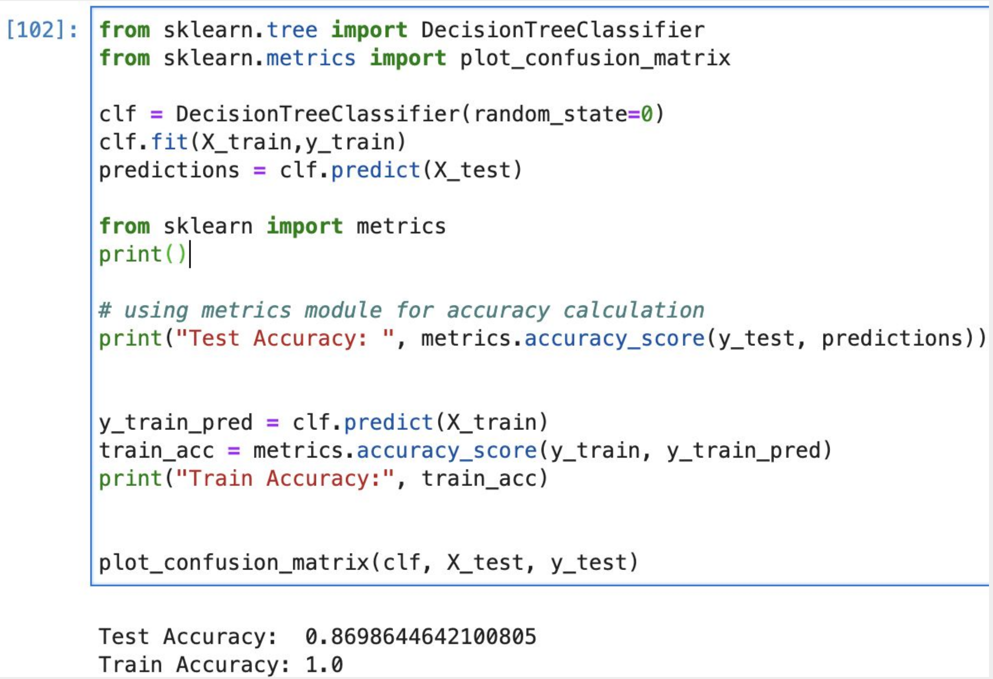


```
[99]: from sklearn.linear_model import LogisticRegression
      clf = LogisticRegression(random_state=0).fit(X_train, y_train)
      from sklearn import metrics
      pred = clf.predict(X_test)
      print()
      print("Test Accuracy:",metrics.accuracy_score(y_test, pred))
      print()
      y_train_pred = clf.predict(X_train)
      train_acc = metrics.accuracy_score(y_train, y_train_pred)
      print("Train Accuracy:", train_acc)

      plot_confusion_matrix(clf, X_test, y_test)
```

```
Test Accuracy: 0.9037484116899619

Train Accuracy: 0.9005264113269196
```

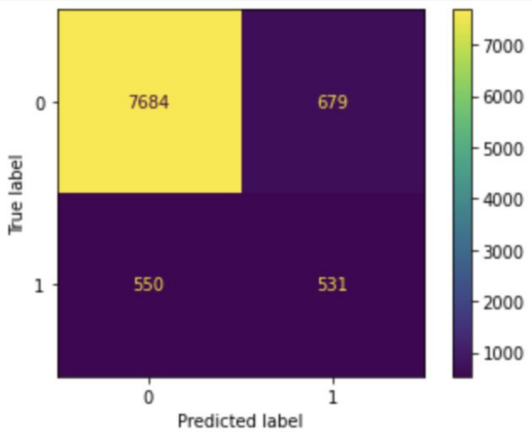


Decision Tree

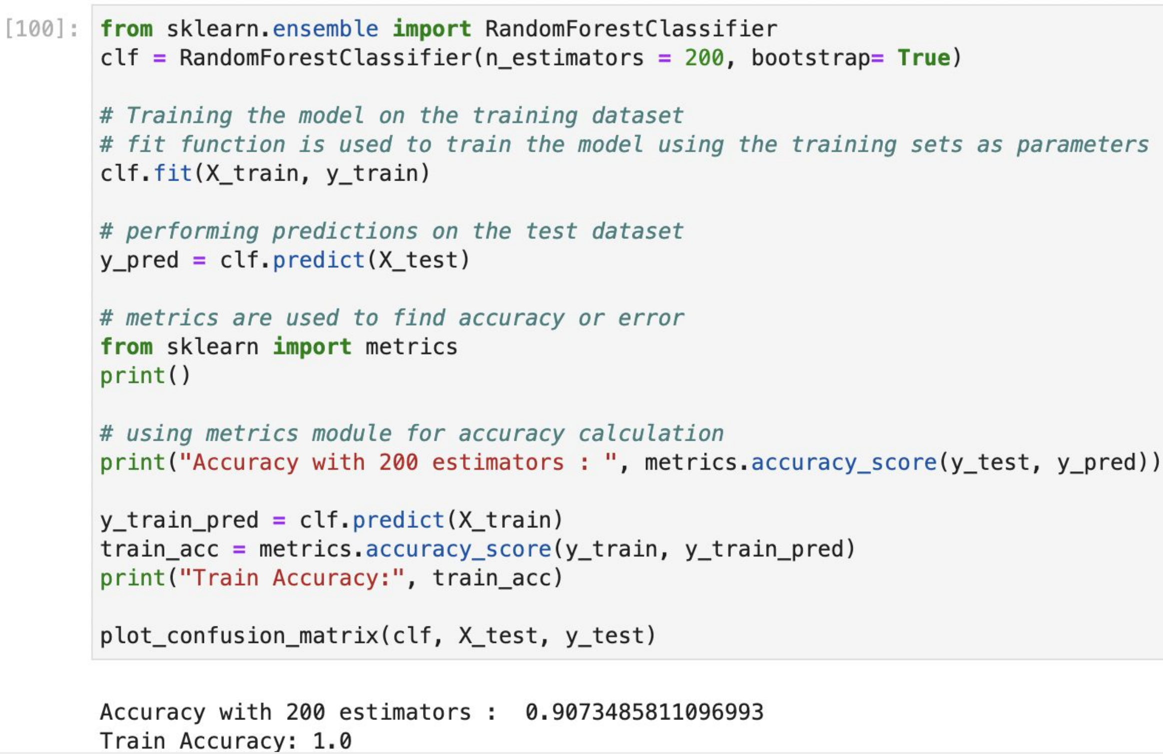


```
[102]: from sklearn.tree import DecisionTreeClassifier
       from sklearn.metrics import plot_confusion_matrix

       clf = DecisionTreeClassifier(random_state=0)
       clf.fit(X_train,y_train)
       predictions = clf.predict(X_test)

       from sklearn import metrics
       print()

       # using metrics module for accuracy calculation
       print("Test Accuracy: ", metrics.accuracy_score(y_test, predictions))


       y_train_pred = clf.predict(X_train)
       train_acc = metrics.accuracy_score(y_train, y_train_pred)
       print("Train Accuracy:", train_acc)


       plot_confusion_matrix(clf, X_test, y_test)
```

```
Test Accuracy:  0.8698644642100805
Train Accuracy: 1.0
```

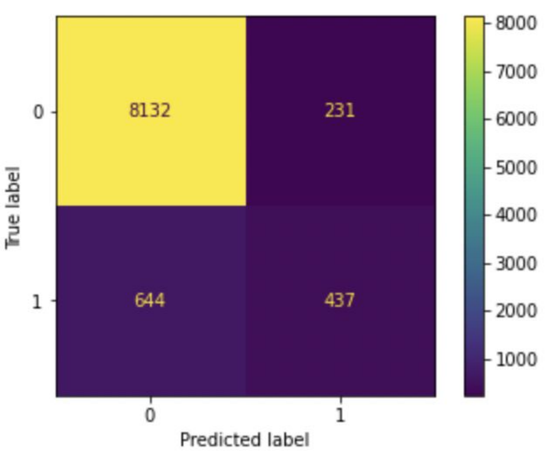


Random Forest

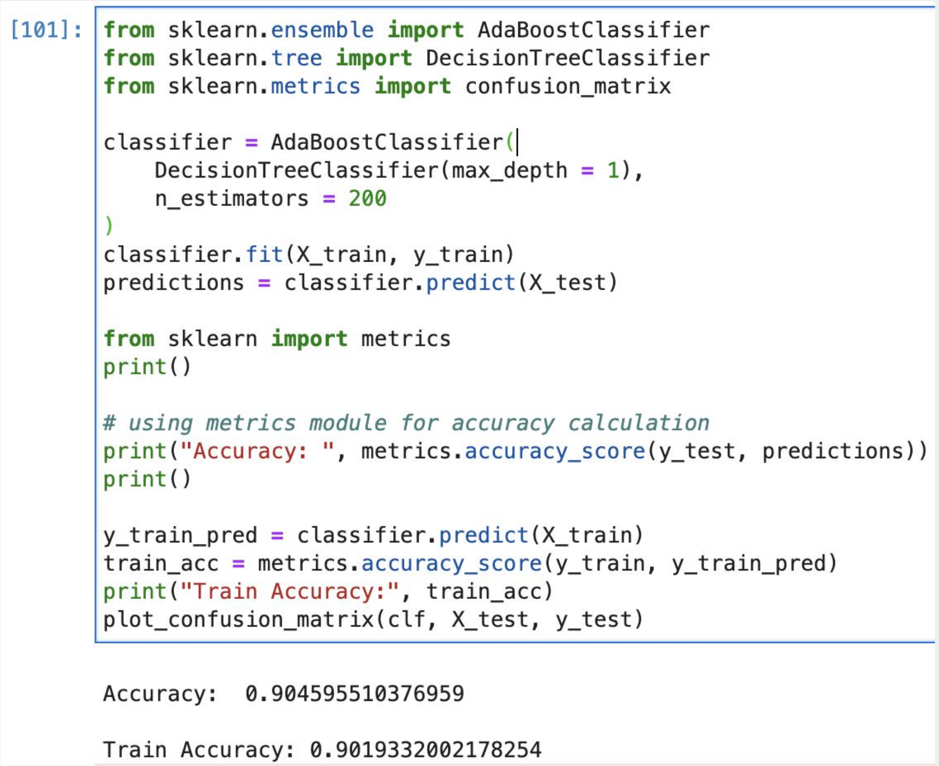


```
[100]: from sklearn.ensemble import RandomForestClassifier
       clf = RandomForestClassifier(n_estimators = 200, bootstrap= True)

       # Training the model on the training dataset
       # fit function is used to train the model using the training sets as parameters
       clf.fit(X_train, y_train)

       # performing predictions on the test dataset
       y_pred = clf.predict(X_test)

       # metrics are used to find accuracy or error
       from sklearn import metrics
       print()

       # using metrics module for accuracy calculation
       print("Accuracy with 200 estimators : ", metrics.accuracy_score(y_test, y_pred))

       y_train_pred = clf.predict(X_train)
       train_acc = metrics.accuracy_score(y_train, y_train_pred)
       print("Train Accuracy:", train_acc)

       plot_confusion_matrix(clf, X_test, y_test)
```

```
Accuracy with 200 estimators :  0.9073485811096993
Train Accuracy: 1.0
```

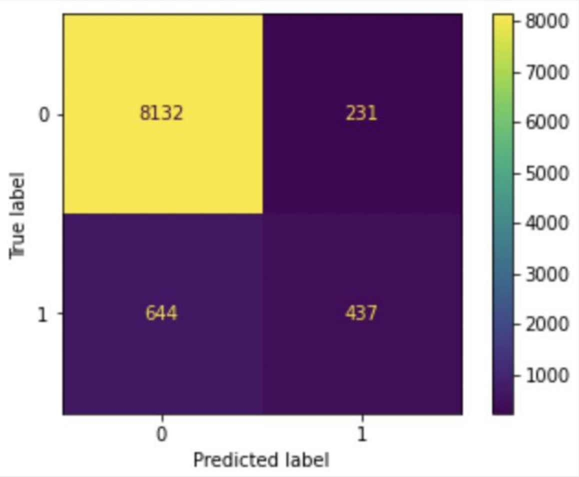


Boosting



```
[101]: from sklearn.ensemble import AdaBoostClassifier
       from sklearn.tree import DecisionTreeClassifier
       from sklearn.metrics import confusion_matrix

       classifier = AdaBoostClassifier(
           DecisionTreeClassifier(max_depth = 1),
           n_estimators = 200
       )
       classifier.fit(X_train, y_train)
       predictions = classifier.predict(X_test)

       from sklearn import metrics
       print()

       # using metrics module for accuracy calculation
       print("Accuracy: ", metrics.accuracy_score(y_test, predictions))
       print()

       y_train_pred = classifier.predict(X_train)
       train_acc = metrics.accuracy_score(y_train, y_train_pred)
       print("Train Accuracy:", train_acc)
       plot_confusion_matrix(clf, X_test, y_test)
```

```
Accuracy:  0.904595510376959

Train Accuracy: 0.9019332002178254
```



Results and Challenges

- Random Forest classifier results in best test accuracy
- Models are highly training data specific
- Data authenticity and Quality issues
- Many irrelevant features
- Limited availability of open source data
- Class Imbalance

Future Scope

- Apply methods to reduce impact of class imbalance on the classifier
- Experiment with different combination of features
- Optimize current models
- Utilize more classification metrics to assess model’s performance

References

- https://www.kaggle.com/code/khanimar/bi-marketing-campaign-eda-analysis-prediction/data
- https://towardsdatascience.com/how-to-predict-the-success-of-your-marketing-campaign-579fbb153a97
- https://www.kaggle.com/code/seananguyen/marketing-campaign-analysis-python#3.-Data-Visualizations
- https://waypointmc.com/blog/analyzing-marketing-results#:~:text=What%20is%20Marketing%20Analysis%3F,improve%20future%20conversions%20or%20sales.