Finlatics Case Project

Banking Dataset

Ritwik Raj

Table of Contents

Summary

Finlatics Data Science Experience Program is a live project that helped me gain work experience in the field of Data Science .

It was a 2-month project that :

- helped me build upon my Python foundation , delving into the basics and understanding its pivotal role in data science

- taught essential statistical methods for precise data analysis in dynamic business applications

- concluded with me applying my learnings to an industry project in the Banking sector for a practical data science learning experience .

Introduction to the Project

The aim of this project is to develop a more granular understanding of

- its customer base

- predict customers' response to its marketing campaign

- establish a target customer profile for future marketing plans

By analysing customer features, such as demographics and transaction history, the bank will be able to predict customer saving behaviours and identify which type of customers is more likely to make term deposits. The bank can then focus its marketing efforts on those customers.

This will not only allow the bank to secure deposits more effectively but also increase customer satisfaction by reducing undesirable advertisements for certain customers.

The data is related to direct marketing campaigns (phone calls) of a Portuguese banking institution .

The classification goal is to predict if the client will subscribe to a term deposit .

Objective

- Exploratory Data Analysis
- Data Visualisation
- Classification using Machine Learning
- Conclusion and Recommendation

Exploratory Data Analysis

There are 45216 observations in the dataset , each representing an existing customer that the bank reached out to . Each observation has 19 features , with the last feature (‘y’) revealing whether the clients have subscribed to the term deposit or not .

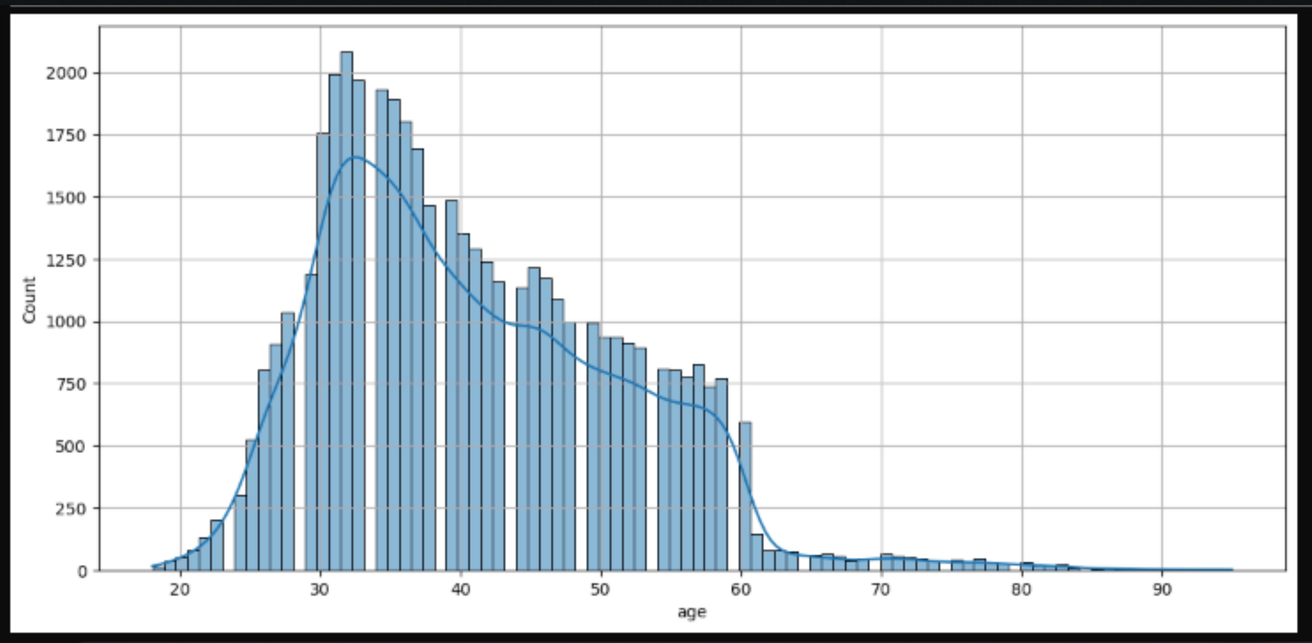
- ‘age’ feature : represents the age of the

Bank client .

The clients have an age range of 18-95 years

old . However , a majority of customers who

called belong to the ages of 30s and 40s (33 to

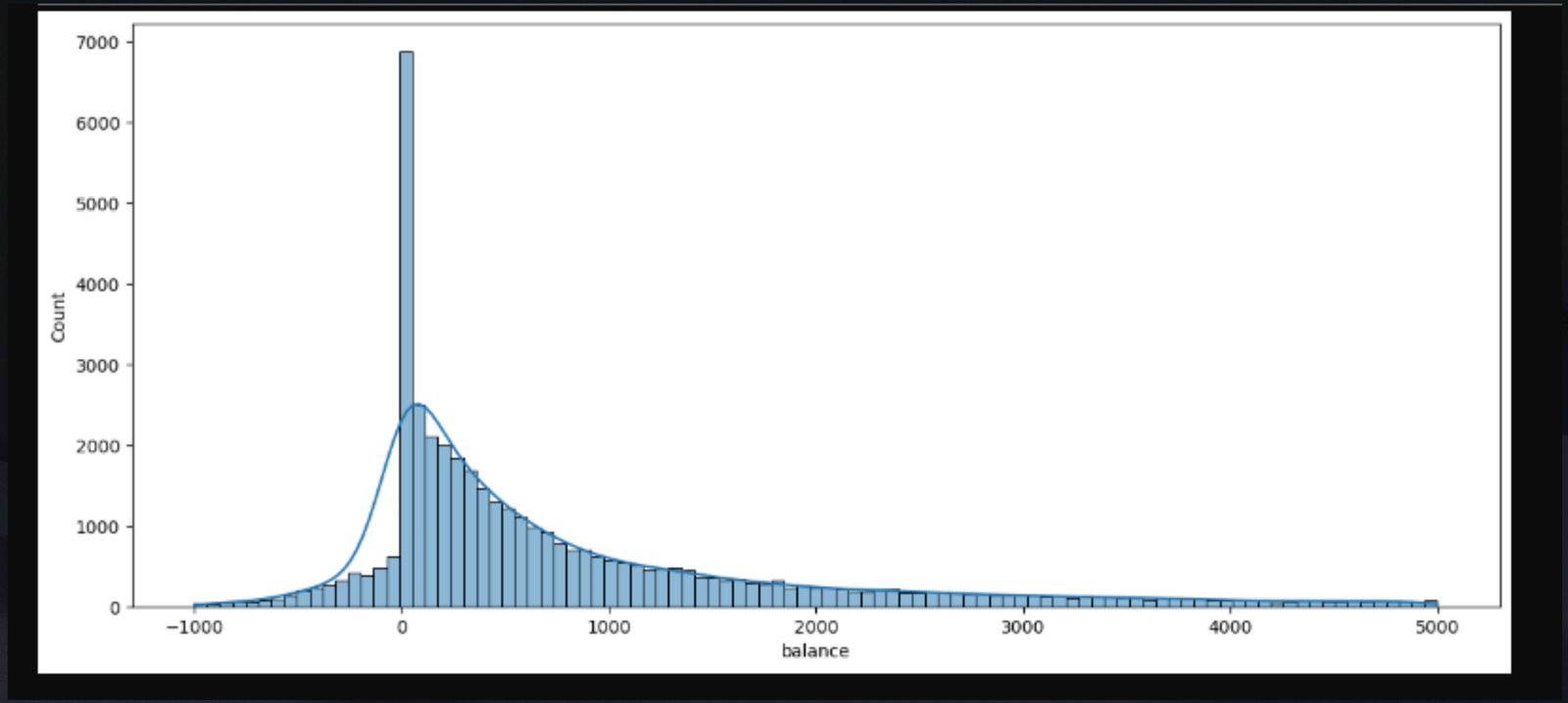


48 years old falls within the 25th and 75th percentile). The distribution is fairly normal with a small standard deviation .

Exploratory Data Analysis

- ‘balance’ feature : represents the average yearly balance in euros for the client .

After dropping outliers in balance , the range of balance is still massive . The distribution of balance has a huge standard deviation relative to the mean , suggesting large variabilities in customers balance levels .



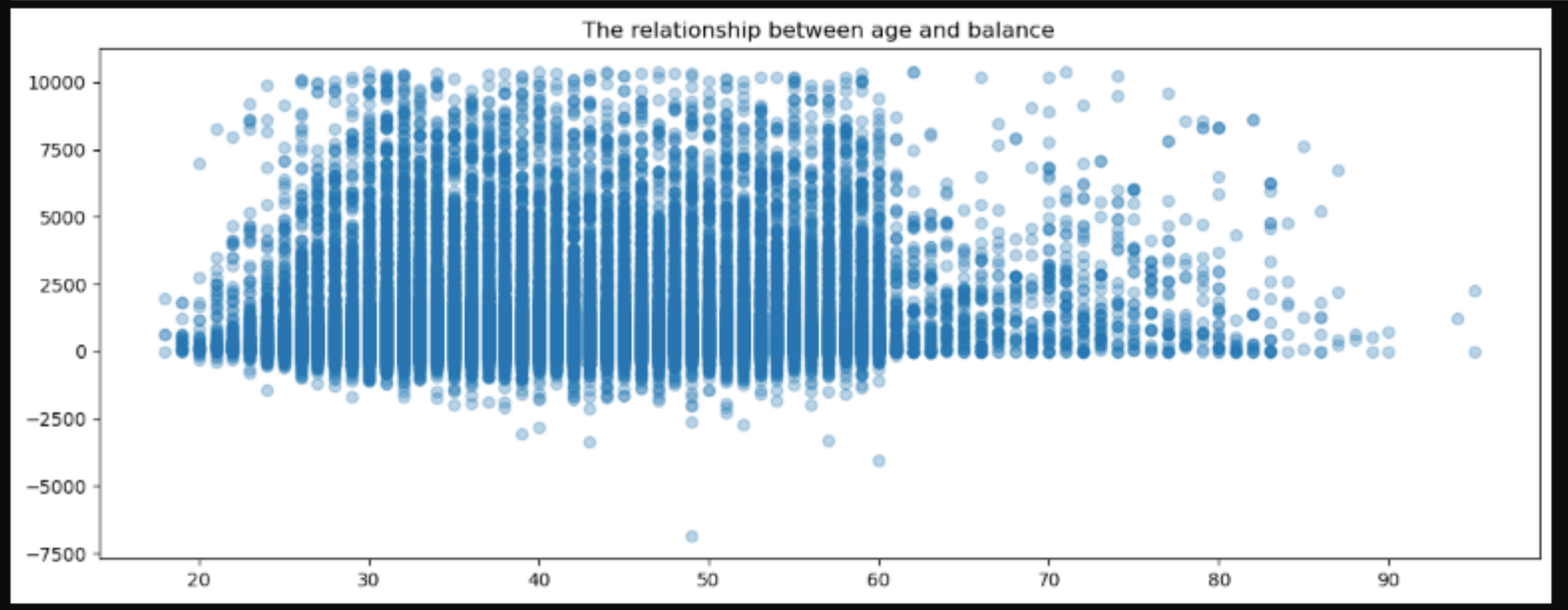
Exploratory Data Analysis

- relationship between ‘age’ and ‘balance’

There is no clear relationship between client’s age and their balance levels. Although , it can be seen that the younger and the older side of the age group have significantly lower balance , attributed to both of them not having a reliable source of income .

```
# Visualising the relationship between 'age' and 'balance'

plt.figure(figsize=[14,5])
plt.scatter(data=dataframe, x='age', y='balance', alpha=0.3)
plt.title("The relationship between age and balance")
plt.show()
```

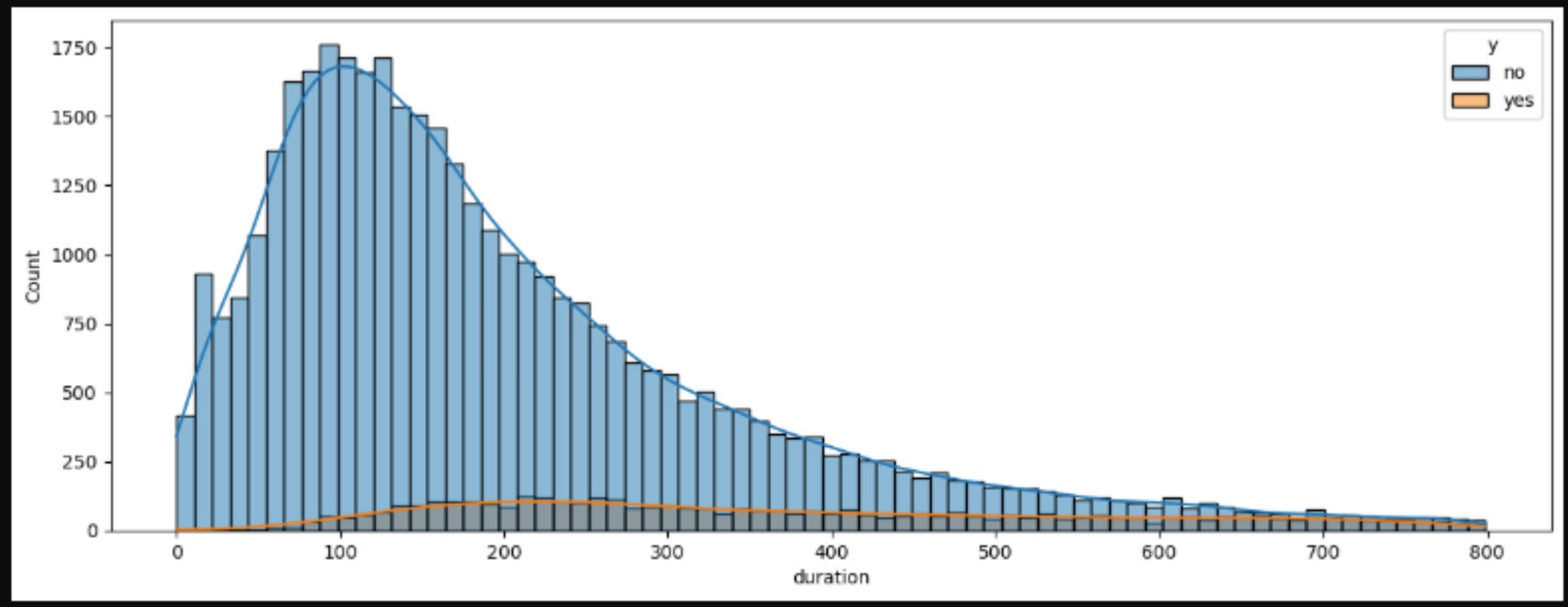


Exploratory Data Analysis

- ‘duration’ feature : represents the duration of the last contact in seconds .

The duration for most calls are relatively short . Although , not much can be said about the positive subscription rate w.r.t duration (as it occurs across all durations) , for the unsuccessful subscriptions , the duration of calls is short , mainly around 50 to 300 seconds .

```
# Magnifying
plt.figure(figsize=[14, 5])
sns.histplot(data=dataframe[dataframe['duration']<800], x='duration', kde=True, hue='y');
```

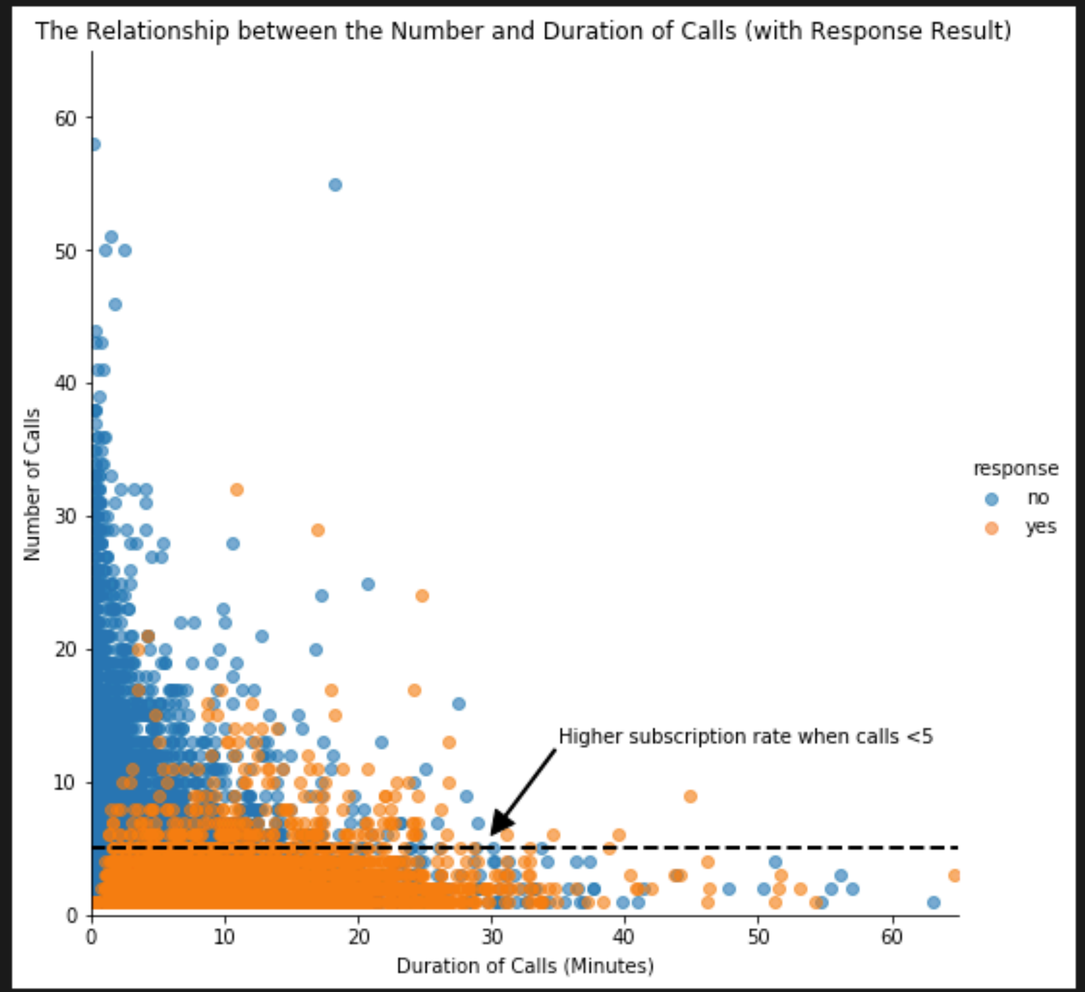


Exploratory Data Analysis

- The relation between the Number and Duration of Calls w.r.t 'y'

As can be seen , successful and unsuccessful subscriptions are forming two relatively separate clusters .

- 'yes' clients were contacted fewer times but for more longer durations .
- clients are more likely to not subscribe after 5-7 campaign calls .
- Banks should resist calling a client for more than 6 times , which can increase dissatisfaction .



Exploratory Data Analysis

- 'y' (subscription) vs 'contact' w.r.t age

Clients above 60 have the highest subscription rate , whilst clients in between 30-39 years of age show a higher inclination to subscribing to the to the term deposit plan .

- main investment objective of older

people is saving for retirement

- middle aged people , due to different

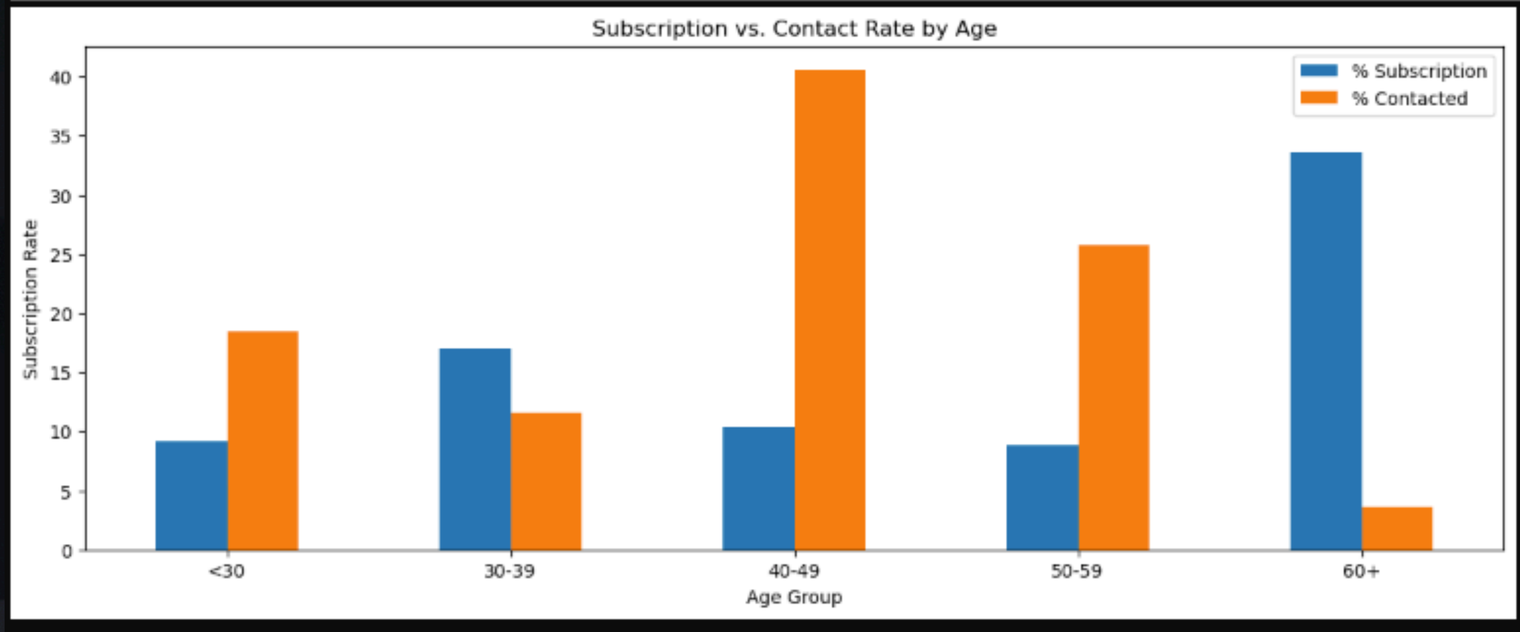
current liabilities feel the burden a lot more

- youngest group , with the main mindset of

generating high investment income do not find term

plans quite appealing .

```
plot_age = age[['% Subscription','% Contacted']].plot(kind = 'bar',figsize=(14,5),)
plt.xlabel('Age Group')
plt.ylabel('Subscription Rate')
plt.xticks(np.arange(5), ('<30', '30-39', '40-49', '50-59', '60+'),rotation = 'horizontal')
plt.title('Subscription vs. Contact Rate by Age')
plt.show()
```



Recommendations : The bank focused more of its effort on 40-49 age group , which returned lower subscription rates than the 30-39 and 60+ crowd . Thus to make the marketing campaigns more effective , the bank should target younger and older clients in the future .

Exploratory Data Analysis

- 'y' (subscription) vs 'contact' w.r.t 'balance'

To identify the trend more easily, clients are categorized into four groups based on their levels of balance:

- No Balance: clients with a negative balance.

- Low Balance: clients with a balance

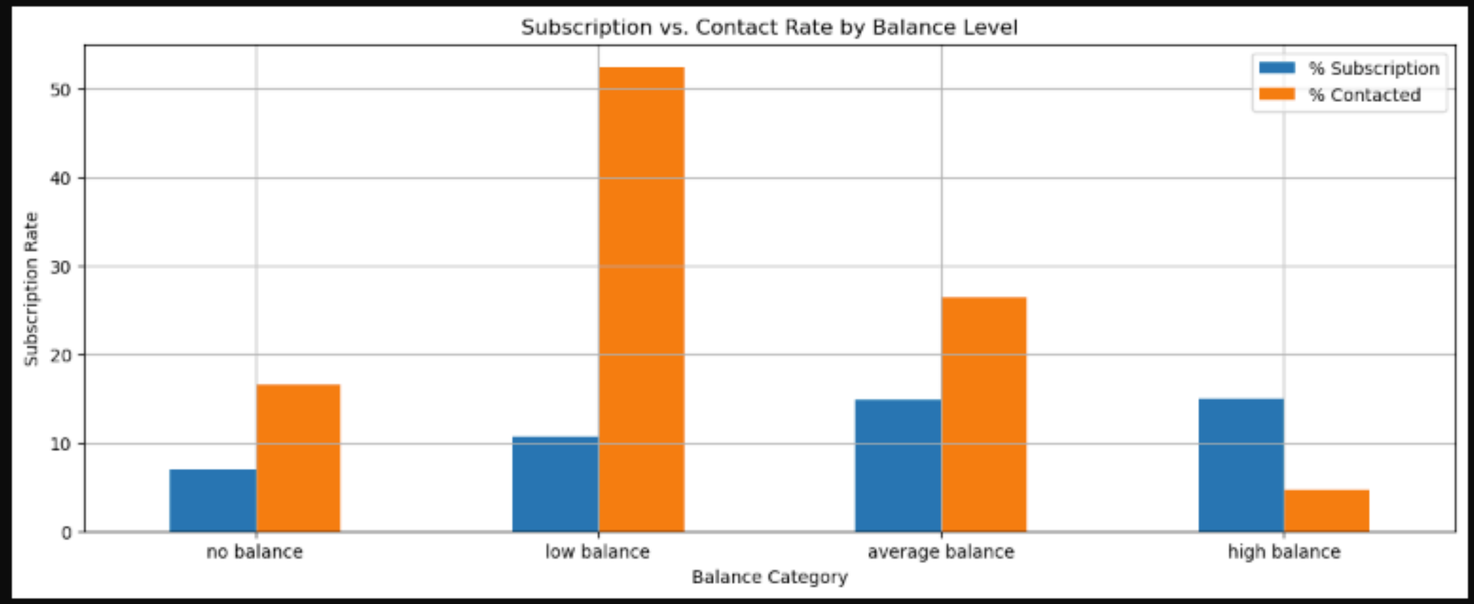
between 0 and 1000 euros

- Average Balance: clients with a

balance between 1000 and 5000 euros.

- High Balance : clients with a balance greater

than 5000 euros



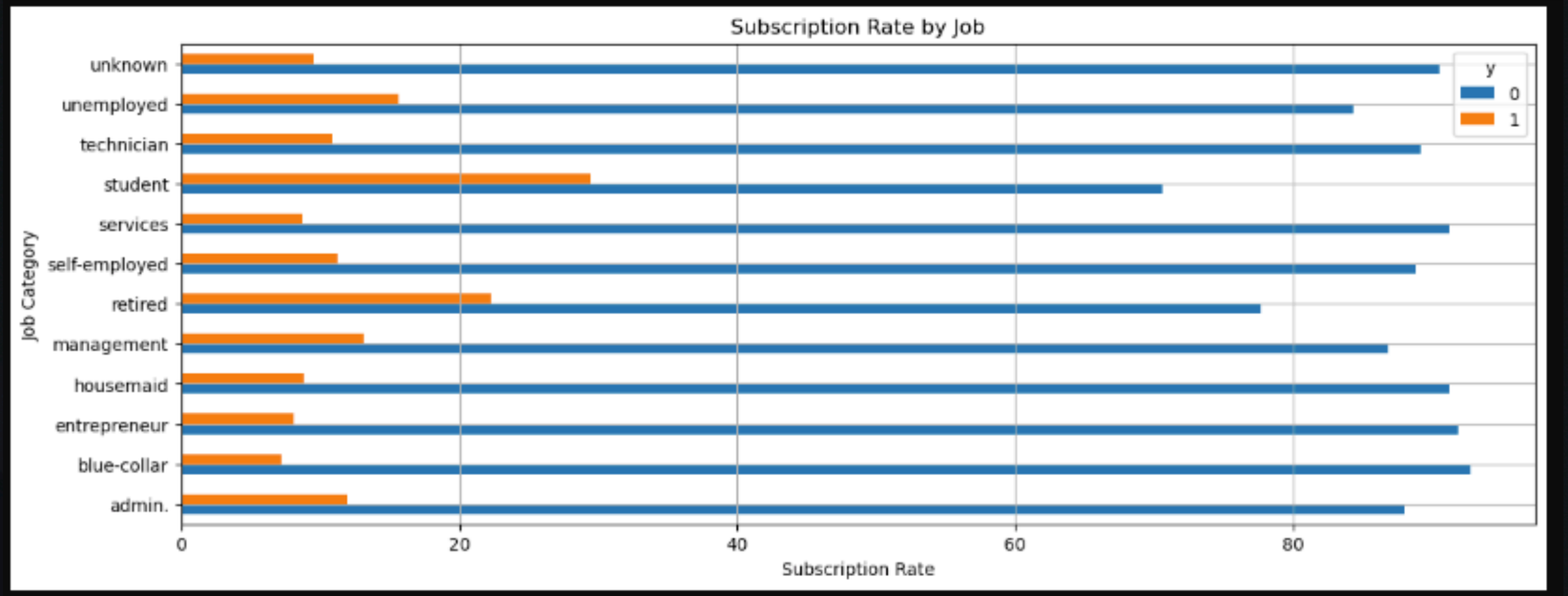
Observation : Unsurprisingly, this bar chart indicates a positive correlation between clients' balance levels and subscription rate. Clients with negative balances only returned a subscription rate of 6.9% while clients with average or high balances had significantly higher subscription rates, nearly 15%.

Recommendation : The bank should shift its marketing focus to high-balance customers to secure more term deposits.

Exploratory Data Analysis

- ‘y’ (subscription) w.r.t job

```
count_job_response_act = pd.crosstab(dataframe['y'], dataframe['job']).apply(lambda x : x/x.sum()*100)
count_job_response_act = count_job_response_act.transpose()
count_job_response_act.plot(kind='barh', figsize=[14, 5])
plt.grid()
plt.title('Subscription Rate by Job')
plt.xlabel('Subscription Rate')
plt.ylabel('Job Category');
```



Observation : Students and Retired clients account for the maximum subscription rates , and hence attention should be put more towards these two client base .

Exploratory Data Analysis

- ‘y’ (subscription) vs ‘contact’ w.r.t ‘month’

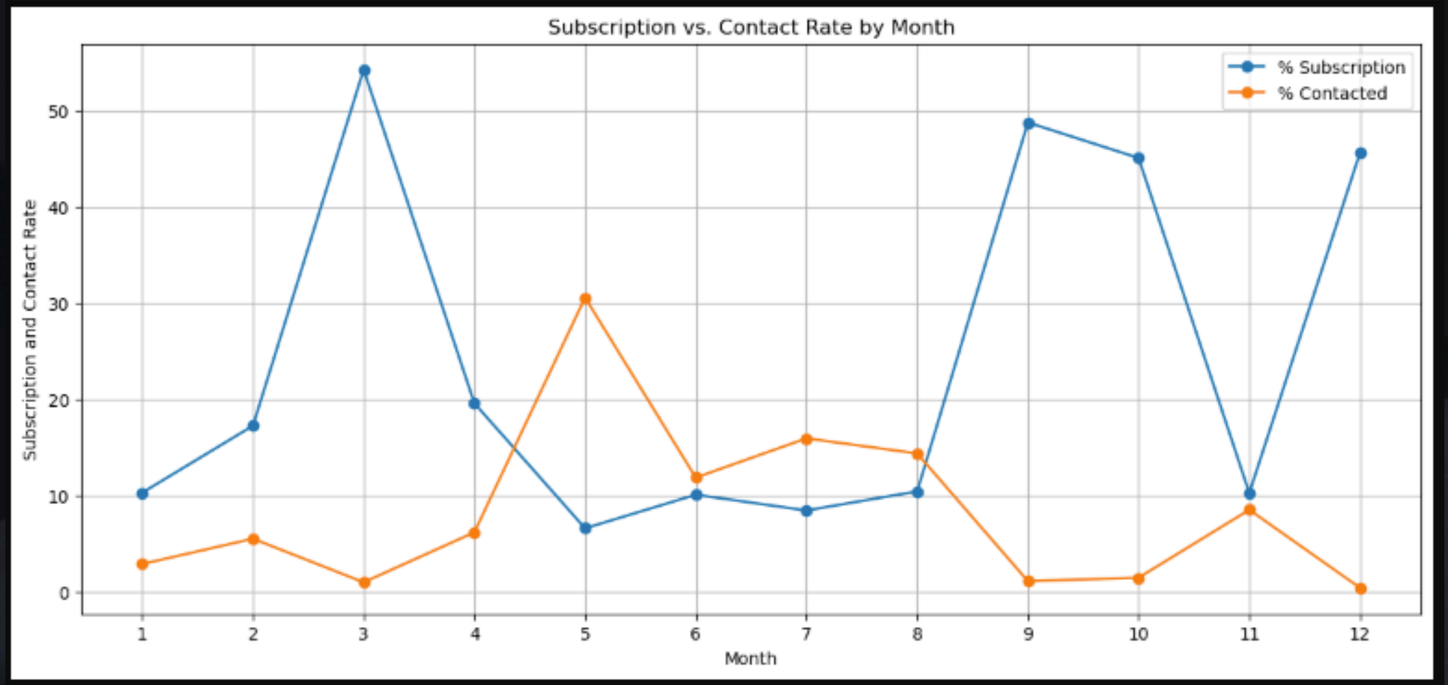
Visualising the subscription and contact rate by month

```
count_month_response_pct = pd.crosstab(dataframe['y'],dataframe['month_int']).apply(lambda x: x/x.sum() * 100)
count_month_response_pct = count_month_response_pct.transpose()
month = pd.DataFrame(dataframe['month_int'].value_counts())
month.rename(columns={'count':'month_int'}, inplace=True)
month['% Contacted'] = month['month_int']*100/month['month_int'].sum()
month['% Subscription'] = count_month_response_pct[1]

month.drop('month_int',axis = 1,inplace = True)
month['Month'] = [5,7,8,6,11,4,2,1,10,9,3,12]

month = month.sort_values('Month',ascending = True)
# display(month)

month[['% Subscription', '% Contacted']].plot(kind='line', figsize=[14,6], marker='o');
plt.xticks(np.arange(1,13,1))
plt.grid()
plt.title('Subscription vs. Contact Rate by Month')
plt.ylabel('Subscription and Contact Rate')
plt.xlabel('Month');
```



(Inference and recommendation in the next slide)

Exploratory Data Analysis

Observation : The line chart displays the bank's contact rate in each month as well as clients' response rate in each month. One way to evaluate the effectiveness of the bank's marketing plan is to see whether these two lines have a similar trend over the same time horizon.

The bank contacted most clients between April and June . The highest contact rate is around 30%, which happened in May, while the contact rate is closer to 0 in March, September, October, and December.

However, the subscription rate showed a different trend. The highest subscription rate occurred in March, which is over 50%, and all subscription rates in September, October, and December are over 40%.

Recommendation : Clearly, these two lines move in different directions which strongly indicates the inappropriate timing of the bank's marketing campaign. To improve the marketing campaign, the bank should consider initiating the telemarketing campaign in fall and spring when the subscription rate tends to be higher.

Note : The bank should be cautious when analyzing external factors. More data from previous marketing campaign should be collected and analyzed to make sure that this seasonal effect is constant over time and applicable to the future.

Classification using Machine Learning

The main objective of this project is to identify the most responsive customers before the marketing campaign so that the bank will be able to efficiently reach out to them, saving time and marketing resources. To achieve this objective, classification algorithms will be employed.

By analysing customer statistics, a classification model will be built to classify all clients into two groups: "yes" to term deposits and "no" to term deposits .

Classification using Machine Learning

```
X = df.drop(columns=['y'], axis=1)
y = df['y']
```

```
X_cat = X.select_dtypes(include=['object'])
X_num = X.select_dtypes(exclude=['object'])
```

```
from sklearn.preprocessing import OneHotEncoder, StandardScaler
# encoder = OneHotEncoder(sparse_output=False)
scaler = StandardScaler()

dummies_X_cat = pd.get_dummies(X_cat)
dummies_X_cat = dummies_X_cat.astype(int)
scaled_X_num = pd.DataFrame(scaler.fit_transform(X_num), columns=X_num.columns)
```

```
final_df = pd.concat([scaled_X_num, dummies_X_cat], axis=1)
final_df
```

	age	balance	day	duration	campaign	pdays	previous	job_admin.	job_blue-collar	job_entrepreneur	job_housemaid	job_management	job_retired	job_self-employed	job_serv
0	1.606403	0.256431	-1.298558	0.011004	-0.569318	-0.411501	-0.252048	0	0	0	0	1	0	0	
1	0.288276	-0.437919	-1.298558	-0.416159	-0.569318	-0.411501	-0.252048	0	0	0	0	0	0	0	
2	-0.747395	-0.446787	-1.298558	-0.707407	-0.569318	-0.411501	-0.252048	0	0	1	0	0	0	0	
3	0.570732	0.047206	-1.298558	-0.645274	-0.569318	-0.411501	-0.252048	0	1	0	0	0	0	0	
4	-0.747395	-0.447116	-1.298558	-0.233644	-0.569318	-0.411501	-0.252048	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
45211	-1.124003	-0.196178	0.023251	-0.078312	-0.569318	-0.411501	-0.252048	0	0	0	0	1	0	0	
45212	2.547922	-0.071037	0.023251	-0.179277	-0.569318	1.466112	2.352398	0	0	0	0	0	1	0	
45213	1.135643	-0.255956	0.143415	-0.124911	-0.569318	1.436150	1.484249	0	0	0	0	1	0	0	
45214	3.018682	0.488647	0.143415	0.162453	-0.569318	-0.002022	3.220546	0	0	0	0	0	1	0	
45215	2.830378	0.120451	0.143415	0.768249	-0.246514	-0.411501	-0.252048	0	0	0	0	0	1	0	

45216 rows x 372 columns

Separating data into features and target , separating features into categorical and numerical features , perform

- one-hot encoding for categorical variables

- standard scaler for numerical variables

Concat the two datasets

Classification using Machine Learning

- Separating training and testing data into X_train , X_test, y_train and y_test.

Fitting LOGISTIC REGRESSION model

- accuracy score : 0.91

- f1-score : 0.51 (yes) , 0.95(no)

Note : Disparity in successful vs unsuccessful Subscription because of the imbalance in the dataset

```
# Split models
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, ConfusionMatrixDisplay, confusion_matrix

X_train, X_test, y_train, y_test = train_test_split(final_df, y, test_size=0.2, random_state=42)
```

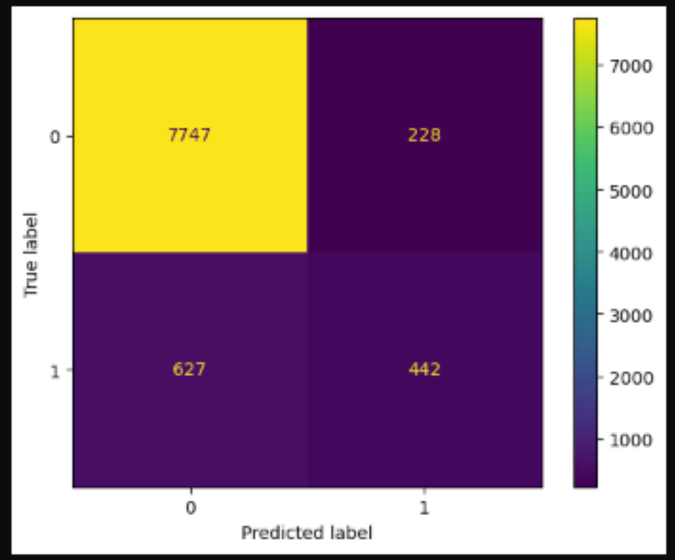
```
# Testing models - Logistic Regression
from sklearn.linear_model import LogisticRegression
lr_model = LogisticRegression(max_iter=100000)
lr_model.fit(X_train, y_train)
pred = lr_model.predict(X_test)

print(accuracy_score(y_test, pred))
print(classification_report(y_test, pred))
ConfusionMatrixDisplay(confusion_matrix=confusion_matrix(y_test, pred)).plot();
```

```
0.9054621848739496
              precision    recall  f1-score   support

          no       0.93      0.97      0.95      7975
         yes       0.66      0.41      0.51      1069

    accuracy                           0.91      9044
   macro avg       0.79      0.69      0.73      9044
weighted avg       0.89      0.91      0.90      9044
```



Classification using Machine Learning

Fitting K NEAREST NEIGHBOR model

- accuracy score : 0.90

- f1-score : 0.47 (yes) , 0.94(no)

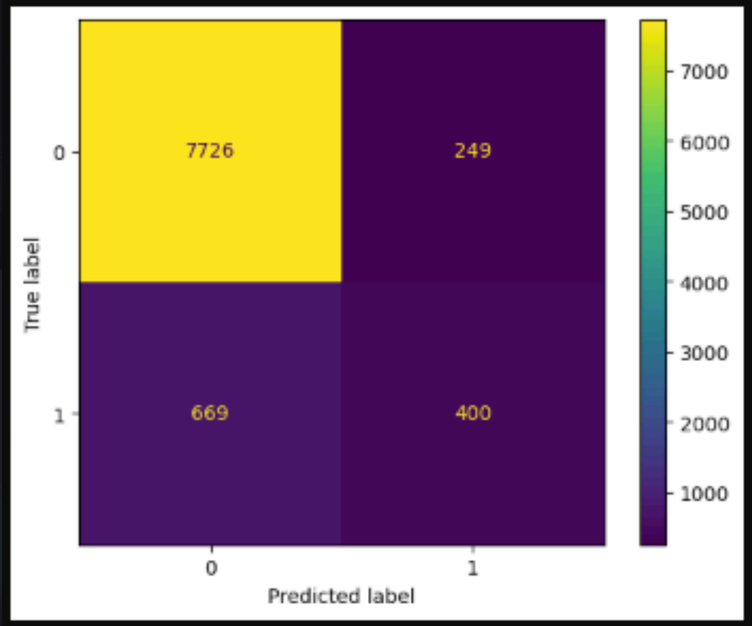
```
# Testing models - KNN
from sklearn.neighbors import KNeighborsClassifier
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model.fit(X_train, y_train)
pred = knn_model.predict(X_test.values)

print(accuracy_score(y_test, pred))
print(classification_report(y_test, pred))
ConfusionMatrixDisplay(confusion_matrix=confusion_matrix(y_test, pred)).plot();
```

```
/Users/ritwik/anaconda3/lib/python3.10/site-packages/sklearn/base.py:464: UserWarn
ure names
  warnings.warn(
0.8984962406015038
              precision    recall  f1-score   support

          no       0.92      0.97      0.94      7975
         yes       0.62      0.37      0.47      1069

    accuracy                           0.90      9044
   macro avg       0.77      0.67      0.70      9044
weighted avg       0.88      0.90      0.89      9044
```



Classification using Machine Learning

Fitting DECISION TREE model

- accuracy score : 0.88

- f1-score : 0.49 (yes) , 0.93 (no)

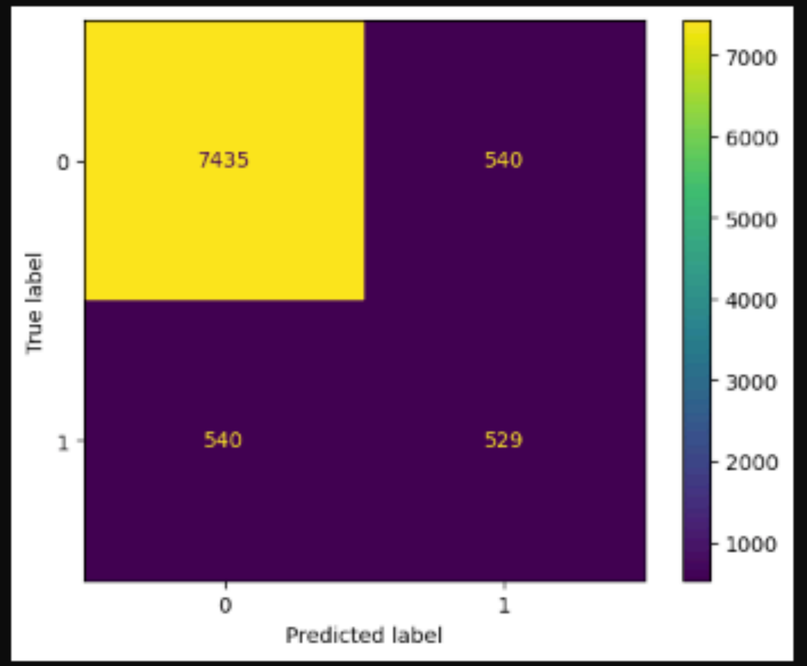
```
# Testing models - Decision Tree
from sklearn.tree import DecisionTreeClassifier
dtree_model = DecisionTreeClassifier()
dtree_model.fit(X_train, y_train)
pred = dtree_model.predict(X_test.values)

print(accuracy_score(y_test, pred))
print(classification_report(y_test, pred))
ConfusionMatrixDisplay(confusion_matrix=confusion_matrix(y_test, pred)).plot();
```

```
/Users/ritwik/anaconda3/lib/python3.10/site-packages/sklearn/base.py:464: UserWar
ature names
  warnings.warn(
0.8805838124723574
              precision    recall  f1-score   support

          no       0.93      0.93      0.93      7975
         yes       0.49      0.49      0.49      1069

    accuracy                           0.88      9044
   macro avg       0.71      0.71      0.71      9044
weighted avg       0.88      0.88      0.88      9044
```



Classification using Machine Learning

Fitting RANDOM FOREST CLASSIFIER model

- accuracy score : 0.90

- f1-score : 0.45 (yes) , 0.95 (no)

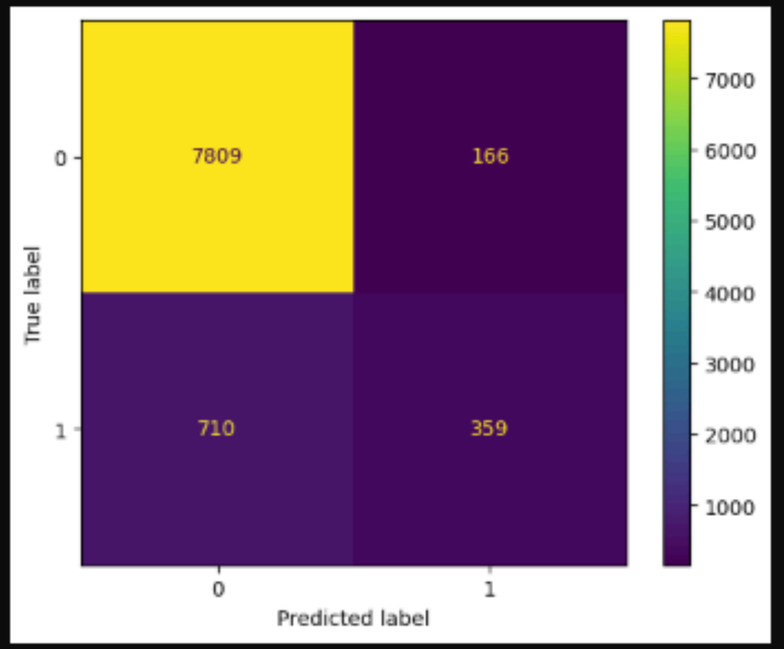
```
: # Testing models -
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier()
rf_model.fit(X_train, y_train)
pred = rf_model.predict(X_test.values)

print(accuracy_score(y_test, pred))
print(classification_report(y_test, pred))
ConfusionMatrixDisplay(confusion_matrix=confusion_matrix(y_test, pred)).plot();
```

```
/Users/ritwik/anaconda3/lib/python3.10/site-packages/sklearn/base.py:464: UserWa
ature names
  warnings.warn(
0.903140203449801
              precision    recall  f1-score   support

          no       0.92      0.98      0.95      7975
         yes       0.68      0.34      0.45      1069

    accuracy                           0.90      9044
   macro avg       0.80      0.66      0.70      9044
weighted avg       0.89      0.90      0.89      9044
```



Conclusion and Recommendation

Conclusion

According to previous analysis, a target customer profile can be established. The most responsive customers possess these features:

- Feature 1: age (30,39) or age > 60

- Feature 2: students or retired people

- Feature 3: a balance of more than 5000 euros

By applying logistic regression , KNN , decision trees and random forest algorithms, classification models were successfully built. With this model , the bank will be able to predict a customer's response to its telemarketing campaign before calling this customer.

In this way, the bank can allocate more marketing efforts to the clients who are classified as highly likely to accept term deposits, and call less to those who are unlikely to make term deposits.

In addition, predicting duration before calling and adjusting marketing plan benefit both the bank and its clients.

- On the one hand, it will increase the efficiency of the bank's telemarketing campaign, saving time and efforts.

- On the other hand, it prevents some clients from receiving undesirable advertisements, raising customer satisfaction.

With the aid of classification models , the bank can enter a virtuous cycle of effective marketing, more investments and happier customers.

Conclusion and Recommendation

Recommendations

1. More appropriate timing

When implementing a marketing strategy, external factors, such as the time of calling, should also be carefully considered. The previous analysis points out that March, September, October and December had the highest success rates. Nevertheless, more data should be collected and analyzed to make sure that this seasonal effect is constant over time. If the trend has the potential to continue in the future, the bank should consider initiating its telemarketing campaign during those times .

2. Smarter marketing design

By targeting the right customers, the bank will have more and more positive responses, and the classification algorithms would ultimately eliminate the imbalance in the original dataset. Hence, more accurate information will be presented to the bank for improving the subscriptions. Meanwhile, to increase the likelihood of subscription, the bank should re-evaluate the content and design of its current campaign, making it more appealing to its target customers.

3. Better services provision

With a more granular understanding of its customer base, the bank has the ability to provide better banking services. For example, marital status and occupation reveals a customer's life stage while loan status indicates his/her overall risk profile. With this information, the bank can estimate when a customer might need to make an investment. In this way, the bank can better satisfy its customer demand by providing banking services for the right customer at the right time.