

(Bank full.csv)

-----Import Important Libraries-----

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
```

-----Read Datasets-----

```
data = pd.read_csv('Downloads/bank-full.csv', sep = ';')
data
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
...
45206	51	technician	married	tertiary	no	825	no	no	cellular	17	nov	977	3	-1	0	unknown	yes
45207	71	retired	divorced	primary	no	1729	no	no	cellular	17	nov	456	2	-1	0	unknown	yes
45208	72	retired	married	secondary	no	5715	no	no	cellular	17	nov	1127	5	184	3	success	yes
45209	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	unknown	no
45210	37	entrepreneur	married	secondary	no	2971	no	no	cellular	17	nov	361	2	188	11	other	no

45211 rows × 17 columns

-----Read last 10 columns-----

```
data.tail(10)
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
45201	53	management	married	tertiary	no	583	no	no	cellular	17	nov	226	1	184	4	success	yes
45202	34	admin.	single	secondary	no	557	no	no	cellular	17	nov	224	1	-1	0	unknown	yes
45203	23	student	single	tertiary	no	113	no	no	cellular	17	nov	266	1	-1	0	unknown	yes
45204	73	retired	married	secondary	no	2850	no	no	cellular	17	nov	300	1	40	8	failure	yes
45205	25	technician	single	secondary	no	505	no	yes	cellular	17	nov	386	2	-1	0	unknown	yes
45206	51	technician	married	tertiary	no	825	no	no	cellular	17	nov	977	3	-1	0	unknown	yes
45207	71	retired	divorced	primary	no	1729	no	no	cellular	17	nov	456	2	-1	0	unknown	yes
45208	72	retired	married	secondary	no	5715	no	no	cellular	17	nov	1127	5	184	3	success	yes
45209	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	unknown	no
45210	37	entrepreneur	married	secondary	no	2971	no	no	cellular	17	nov	361	2	188	11	other	no

-----Read column names-----

```
data.columns.values
```

```
array(['age', 'job', 'marital', 'education', 'default', 'balance',  
      'housing', 'loan', 'contact', 'day', 'month', 'duration',  
      'campaign', 'pdays', 'previous', 'poutcome', 'y'], dtype=object)
```

-----Select specific columns-----

```
columns = ['age', 'balance', 'campaign', 'duration', 'y']
data = data[columns]
data
```

	age	balance	campaign	duration	y
0	58	2143	1	261	no
1	44	29	1	151	no
2	33	2	1	76	no
3	47	1506	1	92	no
4	33	1	1	198	no
...
45206	51	825	3	977	yes
45207	71	1729	2	456	yes
45208	72	5715	5	1127	yes
45209	57	668	4	508	no
45210	37	2971	2	361	no

45211 rows × 5 columns

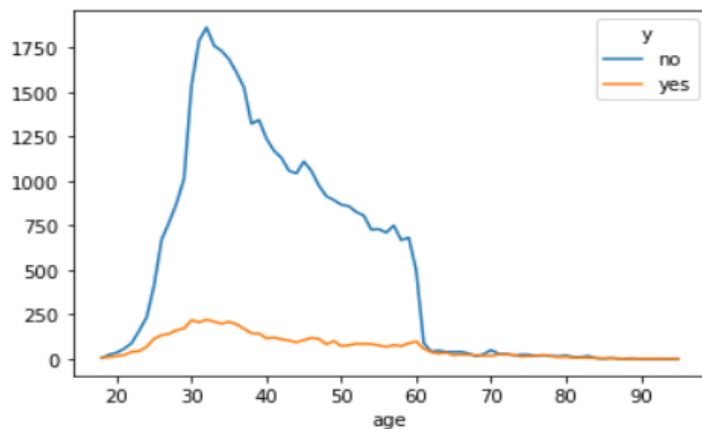
-----Get information-----

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   age         45211 non-null  int64
1   balance     45211 non-null  int64
2   campaign    45211 non-null  int64
3   duration    45211 non-null  int64
4   y           45211 non-null  object
dtypes: int64(4), object(1)
memory usage: 1.7+ MB
```

-----Line Plot-----

```
pd.crosstab(data.age, data.y).plot(kind = 'line')
```



-----Adding new column named 'outcome'-----

```
data['outcome'] = data.y.map({'no':0, 'yes':1})
data.tail(10)
```

	age	balance	campaign	duration	y	outcome
45201	53	583	1	226	yes	1
45202	34	557	1	224	yes	1
45203	23	113	1	266	yes	1
45204	73	2850	1	300	yes	1
45205	25	505	2	386	yes	1
45206	51	825	3	977	yes	1
45207	71	1729	2	456	yes	1
45208	72	5715	5	1127	yes	1
45209	57	668	4	508	no	0
45210	37	2971	2	361	no	0

-----Dividing the Data-----

```
features = ['age', 'balance', 'campaign', 'duration']
feature = ['outcome']
x = data[features]
y = data[feature]
```

-----Define classifier-----

```
classifier = LogisticRegression()  
classifier.fit(x,y)
```

```
LogisticRegression()
```

-----Prediction-----

```
ypredict = classifier.predict (x)  
ypredict
```

```
array([0, 0, 0, ..., 1, 0, 0], dtype=int64)
```

-----Define Confusion matrix-----

```
from sklearn.metrics import confusion_matrix  
confusion_matrix = confusion_matrix (y,ypredict)  
print (confusion_matrix)
```

```
[[39342  580]  
 [ 4435   854]]
```

-----Define Classification Report-----

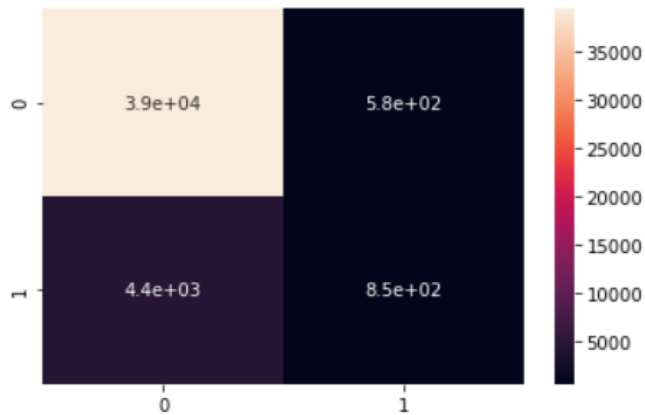
```
from sklearn.metrics import classification_report  
print(classification_report(y,ypredict))
```

	precision	recall	f1-score	support
0	0.90	0.99	0.94	39922
1	0.60	0.16	0.25	5289
accuracy			0.89	45211
macro avg	0.75	0.57	0.60	45211
weighted avg	0.86	0.89	0.86	45211

-----Plot heatmap-----

```
sns.heatmap (confusion_matrix, annot = True)
```

<AxesSubplot:>



-----ROC Curve-----

```
from sklearn.metrics import roc_curve, roc_auc_score
fpr, tpr, thresholds = roc_curve(y, classifier.predict_prob(x)[: , 1])
auc = roc_auc_score(ypredict, y)
plt.plot(fpr, tpr, color='red', label='logit model (area = %0.2f)%auc)
plt.plot([0, 1], [0, 1], 'k--')
plt.title('Receiver operating characteristic')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend(loc='lower right')
```

