bank_market.py 3 ●

C: > Users > Deeksha > bank_market.py > ...

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score
data = pd.read_csv('https://drive.google.com/uc?export=download&id=1MysWWjMWLXF-c4EbAmDk3t80V0BE6sLT')
print(data.head())
print(data.describe())
X = data.drop('y', axis=1)
y = data['y']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')

```

```
>>    age         job  marital  ...  campaign  pdays  y
>> 0   59  blue-collar  married  ...         2    999  0
>> 1   56   housemaid  married  ...         1    999  0
>> 2   41  technician   single  ...         1    999  0
>> 3   55  blue-collar  married  ...         2    999  0
>> 4   54     retired  married  ...         2    999  0
>>
>> [5 rows x 17 columns]
>>                age          campaign          pdays        y
>> count  41188.000000  41188.000000  41188.000000  41188.0
>> mean      40.936207      2.763841    962.475454      0.1
>> std       10.618762      3.098021    186.910907      0.3
>> min       17.000000      1.000000      0.000000      0.0
>> 25%       33.000000      1.000000    999.000000      0.0
>> 50%       39.000000      2.000000    999.000000      0.0
>> 75%       48.000000      3.000000    999.000000      0.0
>> max       98.000000     63.000000    999.000000      1.0
>> Accuracy: 0.9116485355648535
>> Precision: 0.4285714285714285
>> Recall: 0.6666666666666666
>>
```