

Predictive Analytics

- 1. Data Analytics
- 2. Predictive Analytics

In [29]:

```
import pandas as pd
import numpy as np
import seaborn as sns
```

In [30]:

```
#Import Dataset
dataset = pd.read_excel('Dataset Predictive Analysis.xlsx')
```

In [31]:

```
dataset.head()
```

Out[31]:

	status	card_present_flag	bpay_bill	code	account	currency	long_lat	txn_description	merchant_id	merchant_c
0	authorized	1.0	NaN		ACC-1598451071	AUD	153.41-27.95	POS	81c48296-73be-44a7-befa-d053f48ce7cd	
1	authorized	0.0	NaN		ACC-1598451071	AUD	153.41-27.95	SALES-POS	830a451c-316e-4a6a-bf25-e37caedca49e	
2	authorized	1.0	NaN		ACC-1222300524	AUD	151.23-33.94	POS	835c231d-8cdf-4e96-859d-e9d571760cf0	
3	authorized	1.0	NaN		ACC-1037050564	AUD	153.10-27.66	SALES-POS	48514682-c78a-4a88-b0da-2d6302e64673	
4	authorized	1.0	NaN		ACC-1598451071	AUD	153.41-27.95	SALES-POS	b4e02c10-0852-4273-b8fd-7b3395e32eb0	

5 rows x 23 columns



In [32]:

```
data = dataset[['age', 'amount', 'balance']]
```

In [33]:

```
data.head()
```

Out[33]:

	age	amount	balance
0	26	16.25	35.39
1	26	14.19	21.20
2	38	6.42	5.71

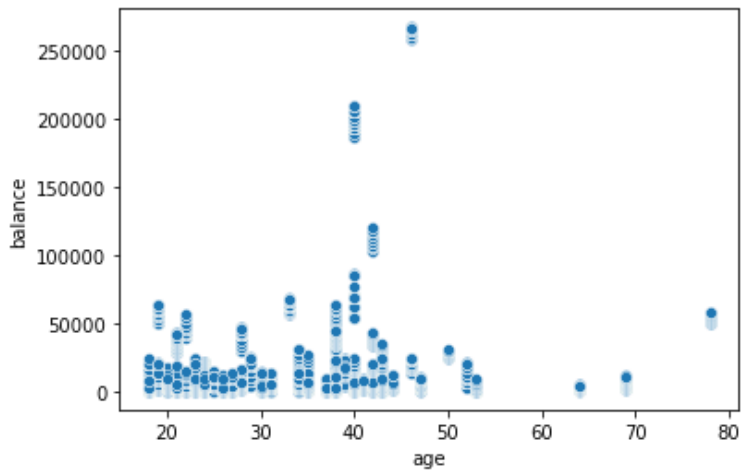
	age	amount	balance
3	40	40.90	2117.22
4	26	3.25	17.95

In [34]:

```
#Scatter Plot - age Vs balance
sns.scatterplot(x=data['age'], y=data['balance'], data= data)
```

Out[34]:

<AxesSubplot:xlabel='age', ylabel='balance'>

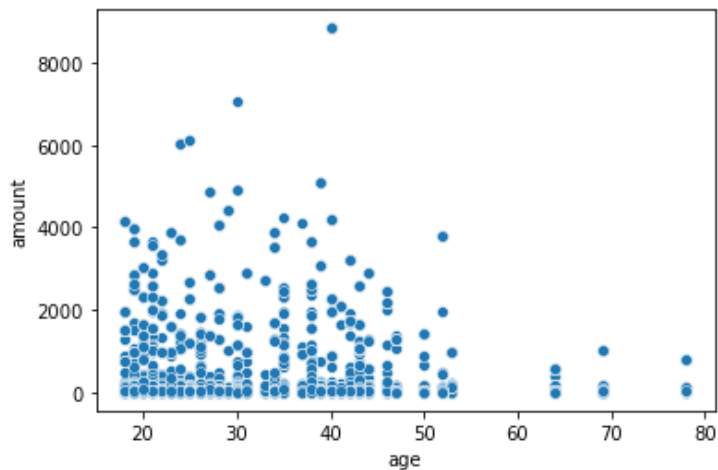


In [35]:

```
#Scatter Plot - age Vs amount
sns.scatterplot(x=data['age'], y=data['amount'], data= data)
```

Out[35]:

<AxesSubplot:xlabel='age', ylabel='amount'>



REGRESSION MODEL

In [36]:

```
x = data[['age', 'balance']]
y = data['balance']
```

In [37]:

```
import sklearn
from sklearn.model_selection import train_test_split
```

In [38]:

```
#Train & Test
```

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.20)
```

```
In [39]:
```

```
from sklearn.linear_model import LinearRegression
```

```
In [40]:
```

```
regr = LinearRegression()
```

```
In [41]:
```

```
regr.fit(X_train, y_train)
```

```
Out[41]:
```

```
LinearRegression()
```

```
In [42]:
```

```
pred = regr.predict(X_train)
```

```
In [43]:
```

```
pred
```

```
Out[43]:
```

```
array([ 1273.98,  1520.95, 15701.73, ...,  9522.21, 10233.88,  3935.48])
```