

social-media-ads

September 22, 2023

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[2]: Ads = pd.read_csv("C:/Users/ypriy/Downloads/Social_Network_Ads.csv")
print(Ads.head())
```

	Age	EstimatedSalary	Purchased
0	19	19000	0
1	35	20000	0
2	26	43000	0
3	27	57000	0
4	19	76000	0

```
[3]: Ads.shape
```

```
[3]: (400, 3)
```

```
[4]: Ads.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Age             400 non-null   int64
1   EstimatedSalary 400 non-null   int64
2   Purchased       400 non-null   int64
dtypes: int64(3)
memory usage: 9.5 KB
```

```
[5]: Ads.describe()
```

```
[5]:
```

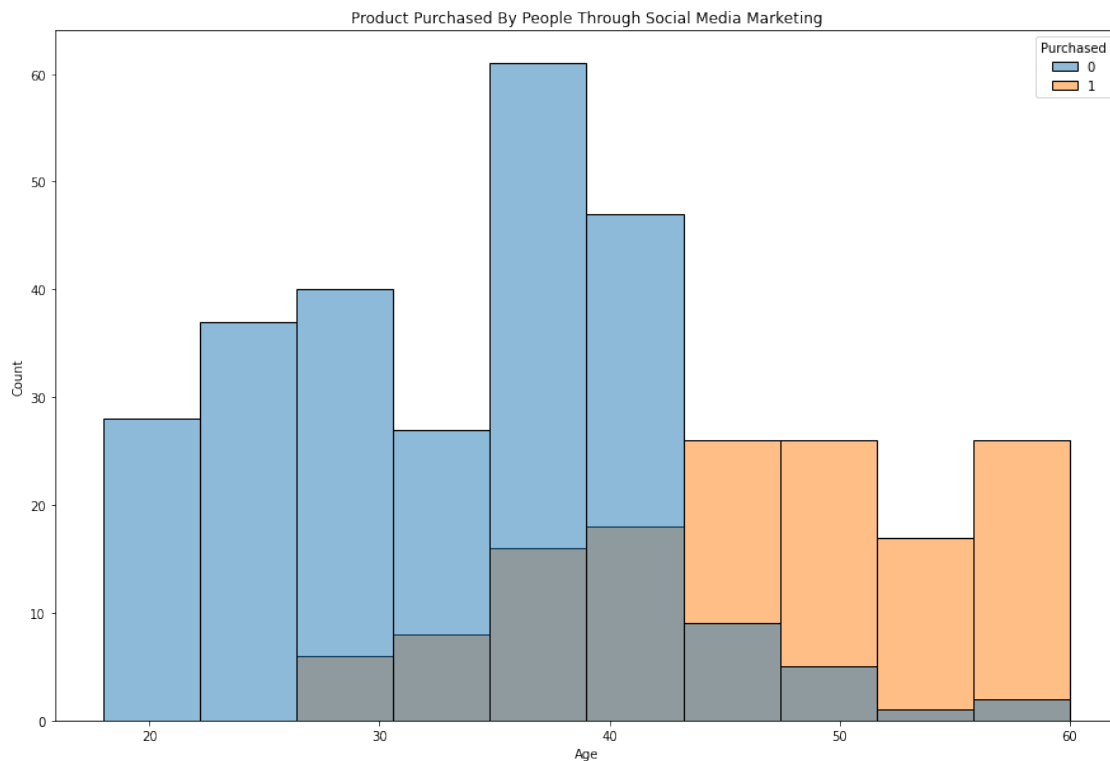
	Age	EstimatedSalary	Purchased
count	400.000000	400.000000	400.000000
mean	37.655000	69742.500000	0.357500
std	10.482877	34096.960282	0.479864

min	18.000000	15000.000000	0.000000
25%	29.750000	43000.000000	0.000000
50%	37.000000	70000.000000	0.000000
75%	46.000000	88000.000000	1.000000
max	60.000000	150000.000000	1.000000

```
[6]: Ads.isnull().sum()
```

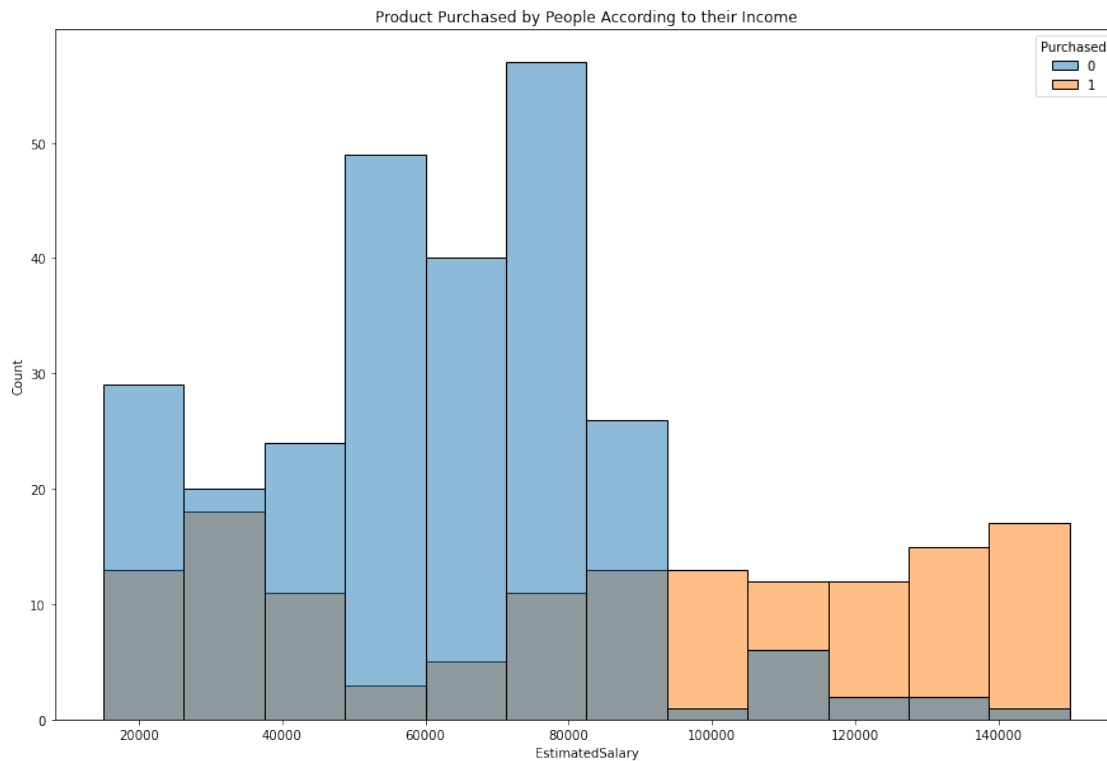
```
[6]: Age          0
     EstimatedSalary  0
     Purchased      0
     dtype: int64
```

```
[7]: #Let's find the ages of people who responded to the social media ads & bought
      ↳ the product
plt.figure(figsize=(15, 10))
plt.title("Product Purchased By People Through Social Media Marketing")
sns.histplot(x="Age", hue="Purchased", data=Ads)
plt.show()
```



The above visualization shows that people over 45 among the target audience are more interested in purchasing the product

```
[13]: #To look at the income group of people who responded to social media ads and
      ↪purchased the product
plt.figure(figsize=(15,10))
plt.title("Product Purchased by People According to their Income")
sns.histplot(x="EstimatedSalary",hue="Purchased", data=Ads)
plt.show()
```



The visualization above shows that people with a monthly income of over 90,000 among the target audience are more interested in purchasing the product. 0 indicates = not interested , 1 indicates = interested

```
[14]: #To Train a model
X = Ads[["Age","EstimatedSalary"]].values
y = Ads["Purchased"].values
```

```
[36]: #let's split the data and train a social media ads classification model using
      ↪the random forest classifier
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,
      ↪random_state=100)
```

```
[37]: from sklearn.ensemble import RandomForestClassifier
      rf = RandomForestClassifier()
```

```
rf.fit(X_train,y_train) #to fit the model
y_pred = rf.predict(X_test) #to predict
```

```
[38]: print(y_pred) #printing the predicted values
```

```
[0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 1 1 1 1 1
 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0
 0 0 0 0 0 0]
```

```
[39]: from sklearn.metrics import accuracy_score, classification_report
accuracy = (accuracy_score(y_pred,y_test)) #to print accuracy score
print(classification_report(y_pred,y_test)) #classification report
```

	precision	recall	f1-score	support
0	0.94	0.85	0.89	54
1	0.74	0.88	0.81	26
accuracy			0.86	80
macro avg	0.84	0.87	0.85	80
weighted avg	0.87	0.86	0.87	80