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

In [4]:
```
df=pd.read_csv("C:/Users/USER/Desktop/Datasets/Social_Network_Ads.csv")
df
```

Out[4]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

In [6]:
```
del df["User ID"]
df
```

Out[6]:

	Gender	Age	EstimatedSalary	Purchased
0	Male	19	19000	0
1	Male	35	20000	0
2	Female	26	43000	0
3	Female	27	57000	0
4	Male	19	76000	0
...	...	...	...	...
395	Female	46	41000	1
396	Male	51	23000	1
397	Female	50	20000	1
398	Male	36	33000	0
399	Female	49	36000	1

400 rows × 4 columns

In [7]:
```
df.dtypes
```

Out[7]:
```
Gender             object
Age                 int64
EstimatedSalary     int64
Purchased           int64
dtype: object
```

In [11]:
```
x=pd.DataFrame(df.iloc[:,0:3])
y=pd.DataFrame(df.iloc[:,3])
```

In [12]:
```
x
```

Out[12]:

	Gender	Age	EstimatedSalary
0	Male	19	19000
1	Male	35	20000
2	Female	26	43000
3	Female	27	57000
4	Male	19	76000
...	...	...	...
395	Female	46	41000
396	Male	51	23000
397	Female	50	20000
398	Male	36	33000
399	Female	49	36000

400 rows × 3 columns

In [13]:
```
y
```

Out[13]:

	Purchased
0	0
1	0
2	0
3	0
4	0
...	...
395	1
396	1
397	1
398	0
399	1

400 rows × 1 columns

In [14]:
```
x.describe()
```

Out[14]:

	Age	EstimatedSalary
count	400.000000	400.000000
mean	37.655000	69742.500000
std	10.482877	34096.960282
min	18.000000	15000.000000
25%	29.750000	43000.000000
50%	37.000000	70000.000000
75%	46.000000	88000.000000
max	60.000000	150000.000000

In [19]:
```
x["Gender"].value_counts()
```

Out[19]:
```
Female    204
Male      196
Name: Gender, dtype: int64
```

In [20]:
```
x.info()
```

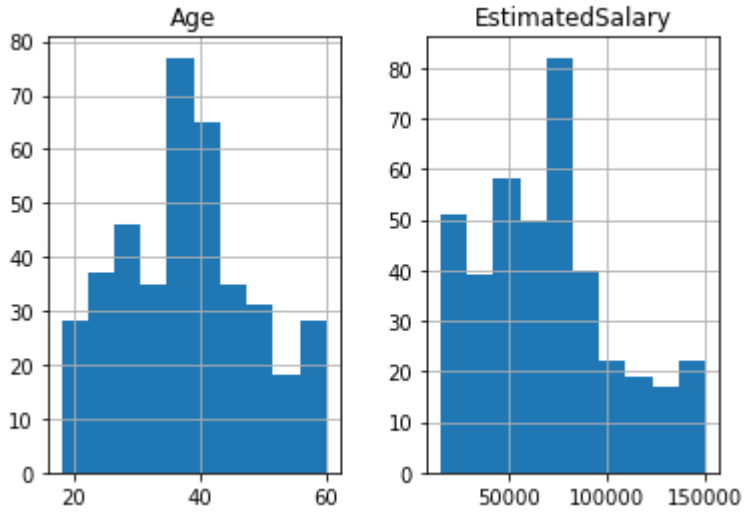
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Gender           400 non-null    object
 1   Age              400 non-null    int64
 2   EstimatedSalary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

In [21]:
```
x.isnull().sum()
```

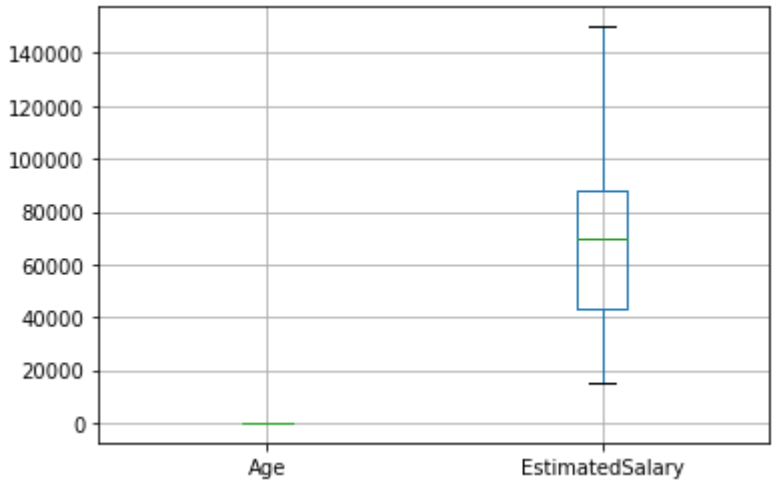
Out[21]:
```
Gender             0
Age                0
EstimatedSalary    0
dtype: int64
```

In [23]:
```
x.hist()
```

Out[23]:
```
array([[<AxesSubplot:title={'center':'Age'}>,
        <AxesSubplot:title={'center':'EstimatedSalary'}>]], dtype=object)
```



In [25]:
```
boxplot=x.boxplot()
```



In [29]:
```
newx=pd.get_dummies(x)
```

In [30]:
```
newx
```

Out[30]:

	Age	EstimatedSalary	Gender_Female	Gender_Male
0	19	19000	0	1
1	35	20000	0	1
2	26	43000	1	0
3	27	57000	1	0
4	19	76000	0	1
...	...	...	...	...
395	46	41000	1	0
396	51	23000	0	1
397	50	20000	1	0
398	36	33000	0	1
399	49	36000	1	0

400 rows × 4 columns

In [31]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(newx, y, test_size = 0.2)
```

In [33]:
```
x_train
```

Out[33]:

	Age	EstimatedSalary	Gender_Female	Gender_Male
378	41	87000	0	1
233	49	86000	0	1
24	46	23000	0	1
151	41	45000	0	1
377	42	53000	1	0
...	...	...	...	...
10	26	80000	1	0
120	36	75000	1	0
165	18	86000	1	0
118	40	59000	0	1
31	27	137000	1	0

320 rows × 4 columns

In [34]:
```
y_test
```

Out[34]:

	Purchased
123	0
86	0
181	0
271	1
193	0
...	...
235	1
111	0
394	0
302	1
351	0

80 rows × 1 columns

In [36]:
```
from sklearn.preprocessing import StandardScaler
scale = StandardScaler()
x_train = scale.fit_transform(x_train)
x_test = scale.transform(x_test)
```

In [37]:
```
from sklearn.naive_bayes import GaussianNB
clf= GaussianNB()
clf.fit(x_train, y_train)
```

```
C:\Users\USER\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataC
onversionWarning: A column-vector y was passed when a 1d array was expected. Pl
ease change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
```

Out[37]:
```
GaussianNB()
```

In [38]:
```
y_pred = clf.predict(x_test)
y_pred
```

Out[38]:
```
array([0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0,
       0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1,
       0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0], dtype=int64)
```

In [40]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
CM=confusion_matrix(y_test, y_pred)
```

In [42]:
```
AS=accuracy_score(y_test, y_pred)
```

In [43]:
```
CM
```

Out[43]:
```
array([[49,  1],
       [ 6, 24]], dtype=int64)
```

In [44]:
```
AS
```

Out[44]:
```
0.9125
```

In [ ]:
```

```