In [18]:
```
import pandas as pd
df=pd.read_csv('Social_Network_Ads.csv')
df.head()
```

Out[18]:

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

In [19]:
```
df.describe()
```

Out[19]:

	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

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

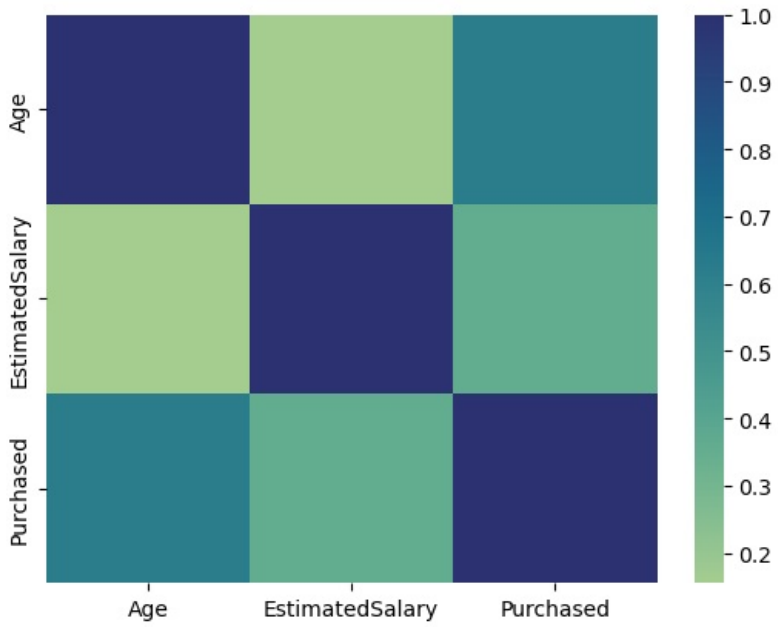
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Age              400 non-null    int64
 1   EstimatedSalary  400 non-null    int64
 2   Purchased  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

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

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

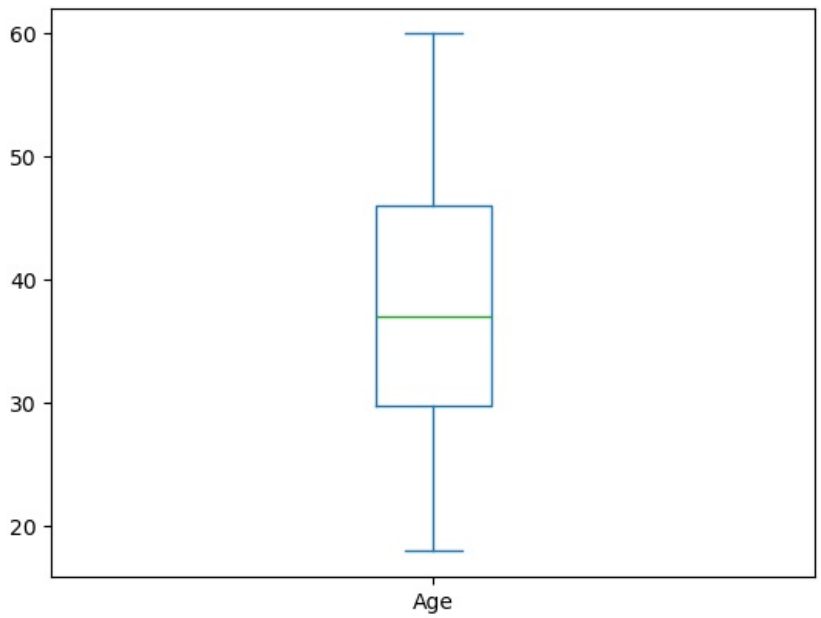
In [59]:
```
import seaborn as sns
correaltion_matrix=df.corr()
sns.heatmap(data=correaltion_matrix,cmap="crest")
```

Out[59]:
```
<Axes: >
```



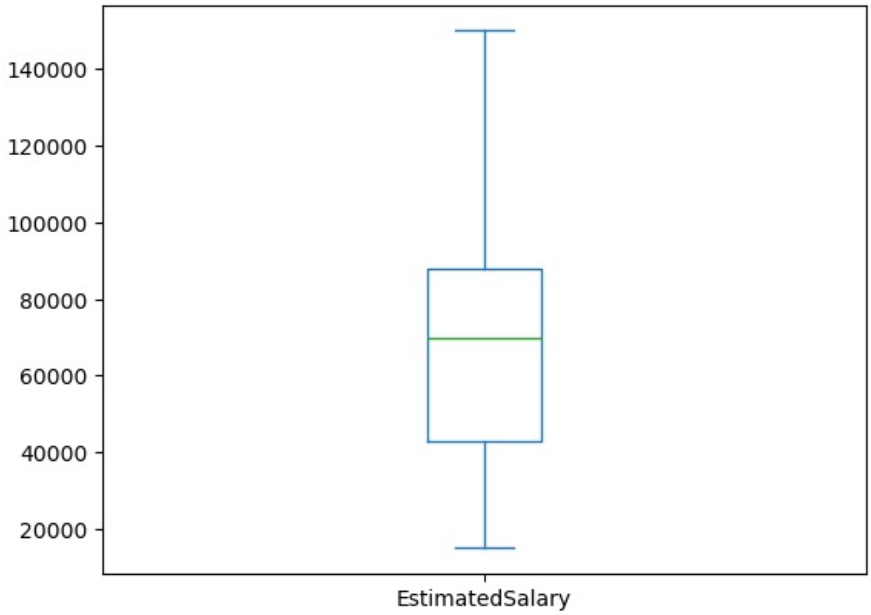
In [23]:
```
df['Age'].plot.box()
```

Out[23]: <Axes: >



In [24]:
```
df['EstimatedSalary'].plot.box()
```

Out[24]: <Axes: >



In [25]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
```

In [26]:
```
x=df.loc[:,['Age','EstimatedSalary']].values
y=df['Purchased'].values
```

Splitting the data

In [32]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

In [33]:
```
#Normalization using scalar transform
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

In [34]:
```
x_train
```

Out[34]:
```
array([[ 0.40019617, -0.12806124],
       [ 1.55218513,  1.11813582],
       [-0.94379096, -0.41787451],
       [ 2.12817961,  0.39360265],
       [-0.46379556, -1.20037033],
       [ 0.11219893,  0.27767734],
       [-1.32778728, -1.08444502],
       [-0.2717974 , -0.64972512],
       [ 0.88019157, -1.34527697],
       [-0.7517928 ,  0.27767734],
       [ 0.20819801, -0.27296787],
       [-0.46379556, -0.76565043],
       [ 1.45618605,  2.13248226],
       [ 0.11219893,  0.16175203],
       [-1.03979004,  0.53850928],
       [ 0.20819801, -0.64972512],
       [-0.55979464,  1.90063164],
       [ 1.07218973,  0.53850928],
       [-0.17579832,  0.16175203],
       [-0.2717974 , -0.91055706],
       [-0.2717974 , -1.37425829],
       [ 0.88019157, -0.53379981],
       [-0.07979924, -0.41787451],
       [ 0.97619065, -1.0554637 ],
       [ 0.20819801,  0.04582672],
       [-0.55979464,  1.37896776],
       [ 2.03218053,  0.39360265],
       [-0.55979464, -1.4901836 ],
       [-0.65579372,  0.04582672],
       [-0.2717974 , -1.28731431],
       [ 0.97619065, -1.0554637 ],
       [-0.2717974 , -0.24398655],
       [-0.65579372, -1.02648237],
       [-0.65579372,  0.1327707 ],
       [ 1.16818881,  0.53850928],
```

```
[ 0.20819801,  0.24869601],
[-0.65579372,  0.19073336],
[ 0.97619065, -1.171389  ],
[-0.07979924,  0.27767734],
[ 0.97619065,  1.43693041],
[ 0.11219893,  0.77035989],
[-0.55979464,  1.46591174],
[ 0.30419709,  0.27767734],
[ 0.20819801, -0.3019492 ],
[ 0.40019617,  0.59647193],
[-0.2717974 ,  0.21971468],
[ 1.93618145, -0.91055706],
[ 0.88019157,  1.08915449],
[ 1.16818881, -0.7366691 ],
[ 1.07218973, -1.20037033],
[-0.2717974 , -0.56278114],
[ 0.78419249,  0.27767734],
[-0.84779188,  0.39360265],
[ 0.68819341, -1.37425829],
[-1.13578912,  1.40794909],
[ 0.78419249, -0.82361308],
[ 0.40019617,  2.30637022],
[ 0.01619984,  1.26304245],
[-0.94379096, -0.3019492 ],
[ 0.78419249, -1.34527697],
[ 0.30419709,  0.04582672],
[ 0.01619984, -0.12806124],
[ 1.45618605, -1.02648237],
[-1.32778728,  0.42258397],
[-1.32778728, -1.22935166],
[-0.84779188, -0.64972512],
[-0.65579372,  1.40794909],
[-0.07979924, -1.0554637 ],
[-0.55979464,  0.48054663],
[ 0.20819801,  0.16175203],
[ 0.01619984, -0.3019492 ],
[ 0.40019617,  1.00221051],
[-0.94379096, -0.93953839],
[-1.80778268, -1.28731431],
[ 0.20819801,  1.08915449],
[ 0.40019617, -0.44685583],
[-0.55979464,  2.33535154],
[ 1.36018697, -1.40323962],
[-0.7517928 , -1.51916493],
[ 1.74418329,  1.84266899],
[ 0.40019617, -0.47583716],
[ 0.68819341,  0.27767734],
[ 0.88019157,  2.16146358],
[ 2.12817961, -0.79463176],
[-1.90378176,  0.48054663],
[-1.13578912, -1.54814625],
[-0.7517928 ,  0.50952795],
[-1.42378636, -0.18602389],
[ 0.59219433, -0.88157574],
[ 0.97619065,  0.1327707 ],
[ 0.20819801, -0.24398655],
[-1.61578452, -0.04111726],
[ 1.93618145, -1.34527697],
[ 0.68819341, -1.25833299],
[ 2.03218053, -1.171389  ],
[-1.42378636, -1.43222095],
[ 0.78419249, -0.3019492 ],
[-1.13578912,  0.07480805],
[ 0.97619065, -0.82361308],
[ 1.74418329,  1.00221051],
[ 0.88019157, -1.02648237],
[ 2.03218053,  1.75572501],
[ 1.64818421,  1.61081837],
[-1.80778268, -1.40323962],
[ 0.01619984,  1.23406112],
[-1.03979004,  0.42258397],
[ 0.01619984, -0.41787451],
[-1.7117836 , -0.59176247],
[-0.94379096, -1.08444502],
[-0.17579832,  0.85730387],
[ 0.88019157, -0.59176247],
[ 0.20819801, -0.12806124],
[-0.84779188, -1.20037033],
[-0.2717974 , -0.35991185],
[-0.2717974 , -0.3019492 ],
[ 1.55218513,  1.00221051],
[ 0.30419709,  0.07480805],
[ 2.03218053, -0.79463176],
```

```
[ 1.55218513, -1.25833299],
[-1.32778728, -0.33093053],
[-0.2717974 ,  0.27767734],
[-1.03979004, -0.33093053],
[-0.07979924,  0.04582672],
[ 0.97619065,  1.87165031],
[-1.32778728, -1.34527697],
[ 0.20819801,  0.07480805],
[-1.13578912, -1.08444502],
[-0.84779188, -0.24398655],
[-0.2717974 ,  0.04582672],
[ 0.20819801, -0.35991185],
[-1.03979004, -0.44685583],
[-1.32778728, -0.41787451],
[-0.7517928 ,  0.30665866],
[ 1.84018237,  0.1327707 ],
[ 0.11219893,  1.87165031],
[-0.2717974 , -1.22935166],
[ 0.01619984, -0.24398655],
[-1.2317882 ,  0.50952795],
[-1.13578912, -1.51916493],
[ 1.07218973,  0.56749061],
[ 0.78419249, -1.37425829],
[-1.03979004,  0.77035989],
[ 0.40019617,  0.30665866],
[ 0.88019157, -1.14240768],
[ 0.01619984, -0.56278114],
[ 1.84018237, -1.25833299],
[-1.13578912, -0.50481849],
[-1.80778268, -1.25833299],
[ 0.40019617,  0.30665866],
[-0.94379096, -0.3019492 ],
[-1.2317882 ,  0.59647193],
[-0.07979924,  0.30665866],
[-1.13578912, -0.99750104],
[ 0.11219893, -0.3019492 ],
[-0.65579372, -1.4901836 ],
[ 1.36018697, -0.91055706],
[ 1.16818881, -0.96851972],
[-0.2717974 , -0.7366691 ],
[-0.17579832,  1.6397997 ],
[-1.13578912, -1.57712758],
[-0.07979924,  2.16146358],
[ 1.93618145, -0.64972512],
[ 0.68819341, -1.08444502],
[-1.61578452,  0.07480805],
[-0.46379556, -0.27296787],
[-0.65579372, -0.04111726],
[-0.46379556, -1.11342635],
[ 1.84018237, -1.0554637 ],
[-0.2717974 , -0.33093053],
[-0.94379096,  0.50952795],
[ 2.12817961, -0.79463176],
[-1.80778268,  0.4515653 ],
[-0.65579372,  0.56749061],
[-0.36779648,  1.23406112],
[-1.32778728, -0.41787451],
[-1.80778268,  0.19073336],
[-1.7117836 ,  0.48054663],
[-0.36779648,  0.07480805],
[-0.46379556,  2.30637022],
[-0.07979924,  0.16175203],
[ 0.97619065,  0.59647193],
[-0.7517928 , -0.59176247],
[-0.46379556, -0.01213593],
[ 1.36018697,  1.29202378],
[-0.2717974 , -0.88157574],
[-0.17579832, -0.44685583],
[-1.42378636, -0.62074379],
[ 0.20819801,  2.10350093],
[ 0.40019617,  0.10378938],
[-0.46379556, -0.53379981],
[-1.32778728, -1.46120227],
[-1.80778268, -1.46120227],
[-1.03979004,  1.9585943 ],
[ 0.78419249, -1.08444502],
[-0.65579372, -1.57712758],
[-1.90378176, -0.04111726],
[ 0.88019157, -0.64972512],
[ 0.40019617,  1.11813582],
[ 0.30419709, -0.50481849],
[-1.03979004, -1.43222095],
[ 0.97619065,  2.0745196 ],
```

```
[ 0.30419709,  0.30665866],
[ 0.01619984, -0.56278114],
[ 0.20819801,  0.16175203],
[ 0.49619525,  1.23406112],
[ 0.11219893,  0.21971468],
[-0.2717974 ,  0.53850928],
[ 0.78419249,  0.77035989],
[-0.2717974 , -0.27296787],
[-1.13578912, -1.14240768],
[ 0.49619525,  1.72674368],
[ 0.01619984,  0.30665866],
[-0.07979924,  0.24869601],
[-0.17579832,  1.61081837],
[-1.7117836 ,  0.36462132],
[-0.2717974 , -0.41787451],
[ 0.01619984, -0.24398655],
[ 0.11219893,  0.04582672],
[ 0.97619065, -0.99750104],
[ 0.78419249,  1.37896776],
[-1.90378176, -0.7366691 ],
[ 0.30419709,  0.07480805],
[ 1.07218973,  0.1327707 ],
[ 0.97619065,  1.98757562],
[ 0.30419709, -0.3019492 ],
[-1.13578912,  0.30665866],
[-0.17579832, -0.50481849],
[-0.2717974 , -0.56278114],
[ 1.07218973,  2.0745196 ],
[ 0.11219893,  0.10378938],
[-0.2717974 , -0.12806124],
[ 2.03218053,  0.19073336],
[-0.46379556,  1.26304245],
[-0.2717974 ,  1.11813582],
[ 1.36018697,  1.98757562],
[-0.2717974 ,  0.10378938],
[ 1.64818421,  1.75572501],
[ 0.01619984, -0.53379981],
[-1.90378176,  0.36462132],
[-1.13578912,  0.33563999],
[-0.07979924,  0.30665866],
[ 0.30419709,  0.07480805],
[ 0.88019157, -0.76565043],
[-0.7517928 ,  1.08915449],
[ 1.93618145,  2.16146358],
[ 1.45618605,  0.07480805],
[ 1.93618145,  0.91526653],
[ 0.11219893,  0.04582672],
[-0.36779648, -0.76565043],
[ 2.12817961,  0.94424786],
[ 0.11219893, -0.24398655],
[-0.2717974 ,  0.62545326],
[-0.2717974 ,  0.16175203],
[-0.7517928 ,  1.34998643],
[-0.94379096,  0.27767734],
[ 1.36018697,  2.33535154],
[ 1.07218973, -1.20037033],
[-1.61578452,  0.53850928],
[-1.51978544, -0.41787451],
[ 0.11219893, -0.79463176],
[ 2.12817961, -1.02648237],
[-0.84779188,  2.27738889],
[-0.07979924,  0.21971468],
[-1.03979004, -1.51916493],
[-0.2717974 ,  0.07480805],
[-1.2317882 , -1.0554637 ],
[-1.03979004, -0.35991185],
[ 0.78419249,  0.53850928],
[-1.03979004,  0.56749061],
[-0.17579832, -0.27296787],
[ 0.30419709, -1.14240768],
[ 0.88019157,  1.26304245],
[ 0.40019617,  0.0168454 ],
[-1.2317882 , -1.37425829],
[ 0.88019157, -1.43222095],
[-0.65579372, -0.33093053],
[-0.2717974 , -1.43222095],
[-0.2717974 ,  2.24840756],
[-1.03979004,  0.59647193],
[-0.17579832,  2.16146358],
[-0.2717974 , -1.34527697],
[ 0.11219893,  1.52387439],
[-1.42378636, -0.09907991],
[ 0.78419249,  0.1327707 ],
```

```
       [ 1.26418789, -1.34527697],
       [-1.51978544, -1.22935166],
       [ 0.30419709, -0.70768778],
       [ 0.49619525,  1.84266899],
       [ 1.45618605,  1.00221051],
       [ 0.88019157,  1.03119184],
       [-0.55979464,  1.37896776],
       [-1.80778268,  0.0168454 ],
       [-1.32778728,  0.56749061],
       [ 2.12817961, -0.67870645],
       [-0.84779188, -0.76565043],
       [ 0.78419249,  0.36462132],
       [ 0.30419709, -0.18602389],
       [-0.94379096,  0.4515653 ],
       [ 0.30419709,  0.50952795],
       [ 1.84018237, -0.27296787],
       [ 1.93618145,  0.74137857],
       [ 0.30419709, -0.50481849],
       [-0.36779648, -1.28731431],
       [ 0.11219893,  1.87165031],
       [-0.2717974 , -0.47583716],
       [ 1.55218513,  0.0168454 ],
       [-0.94379096,  0.42258397],
       [ 1.16818881, -1.43222095],
       [-0.7517928 ,  0.56749061],
       [ 1.26418789,  2.21942624],
       [-0.94379096,  0.56749061],
       [-0.17579832, -0.56278114],
       [-1.03979004, -1.11342635],
       [ 0.40019617,  0.16175203],
       [ 1.07218973, -0.88157574],
       [-0.55979464, -1.4901836 ],
       [-0.84779188,  0.30665866],
       [-0.17579832, -0.18602389],
       [-0.07979924,  1.9585943 ],
       [-1.13578912, -1.57712758]])
```

In [36]:
```
model=LogisticRegression()
model.fit(x_train,y_train)
```

Out[36]:
```
▾ LogisticRegression
LogisticRegression()
```

In [37]:
```
y_pred=model.predict(x_test)
print(y_pred)
```

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

In [38]:
```
import numpy as np
#comparing acutal and predicted value side by side
y_comp=np.vstack((y_test,y_pred))
y_comp
```

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

In [45]:
```
mse=mean_squared_error(y_test,y_pred)
mse
```

Out[45]: 0.175

In [46]:
```
import numpy as np
rmse=np.sqrt(mse)
rmse
```

Out[46]: 0.4183300132670378

In [47]:
```
r2_score=model.score(x_test,y_test)
r2_score
```

Out[47]: 0.825

Confusion matrix

In [39]:
```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
cm
```

Out[39]:
```
array([[48,  3],
       [11, 18]], dtype=int64)
```

In [40]:
```
print('TN',cm[0,0])
print('FP',cm[0,1])
print('FN',cm[1,0])
print('TP',cm[1,1])
```

```
TN 48
FP 3
FN 11
TP 18
```

In [61]:
```
from sklearn import metrics
fpr, tpr, threshold = metrics.roc_curve(y_test, y_pred)
print('The accuracy value : ', metrics.auc(fpr, tpr))
```

```
The accuracy value :  0.7809330628803245
```

In [62]:
```
from sklearn.metrics import average_precision_score
print('The Average Precision score: ',average_precision_score(y_test, y_pred) )
```

```
The Average Precision score:  0.6695197044334975
```

In [63]:
```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_pred)
print('The mean square error :',mse)
```

```
The mean square error : 0.175
```