In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv('heart.csv')
```

In [3]:
```
df.shape
```

Out[3]:
```
(303, 15)
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  303 non-null    int64  
 1   Age         303 non-null    int64  
 2   Sex         303 non-null    int64  
 3   ChestPain   303 non-null    object 
 4   RestBP      303 non-null    int64  
 5   Chol        303 non-null    int64  
 6   Fbs         303 non-null    int64  
 7   RestECG     303 non-null    int64  
 8   MaxHR       303 non-null    int64  
 9   ExAng       303 non-null    int64  
 10  Oldpeak     303 non-null    float64
 11  Slope       303 non-null    int64  
 12  Ca          299 non-null    float64
 13  Thal        301 non-null    object 
 14  AHD         303 non-null    object 
dtypes: float64(2), int64(10), object(3)
memory usage: 35.6+ KB
```

In [5]:
```
(df==0).sum()
```

Out[5]:
```
Unnamed: 0      0
Age             0
Sex            97
ChestPain       0
RestBP          0
Chol            0
Fbs           258
RestECG       151
MaxHR           0
ExAng         204
Oldpeak        99
Slope           0
Ca            176
Thal            0
AHD             0
dtype: int64
```

In [6]:
```
df['Age'].mean()
```

Out[6]:
```
54.43894389438944
```

In [7]:
```
df.isna().sum()
```

Out[7]:
```
Unnamed: 0    0
Age           0
Sex           0
ChestPain     0
RestBP        0
Chol          0
Fbs           0
RestECG       0
MaxHR         0
ExAng         0
Oldpeak       0
Slope         0
Ca            4
Thal          2
AHD           0
dtype: int64
```

In [8]:

```
df.mean()
```

Out[8]:

```
Unnamed: 0    152.000000
Age            54.438944
Sex             0.679868
RestBP        131.689769
Chol          246.693069
Fbs             0.148515
RestECG         0.990099
MaxHR         149.607261
ExAng           0.326733
Oldpeak         1.039604
Slope           1.600660
Ca              0.672241
dtype: float64
```

In [9]:

```
df[df['Ca'].isnull()].index.tolist()
```

Out[9]:

```
[166, 192, 287, 302]
```

In [10]:

```
df.iloc[[166]]
```

Out[10]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
166	167	52	1	nonanginal	138	223	0	0	169	0	0.0	1	NaN	normal	No

In [11]:

```
df['Ca'].fillna(df['Ca'].mean(), inplace=True)
```

In [12]:

```
df.isna().sum().sum()
```

Out[12]:

```
2
```

In [13]:

```
df.iloc[[166]]
```

Out[13]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal	AHD
166	167	52	1	nonanginal	138	223	0	0	169	0	0.0	1	0.672241	normal	No

In [14]:

```
df = df[df['Thal'].notna()]
```

In [15]:

```
df2 = df.filter(['Age', 'Sex', 'ChestPain', 'RestBP', 'Chol'])
```

In [16]:

```
print(df['ChestPain'].unique())
```

```
['typical' 'asymptomatic' 'nonanginal' 'nontypical']
```

In [17]:

```
X = df2.replace(to_replace=dict(typical=1 ,asymptomatic=2 ,nonanginal=3 ,nontypical=4))
X = X / X.max()
```

In [18]:

```
df3 = df.filter(['AHD'])
```

In [19]:

```
y = df3.replace(dict(No=0, Yes=1))
```

In [20]:

```
print("X size: ", X.shape)
print("y size: ", y.shape)
```

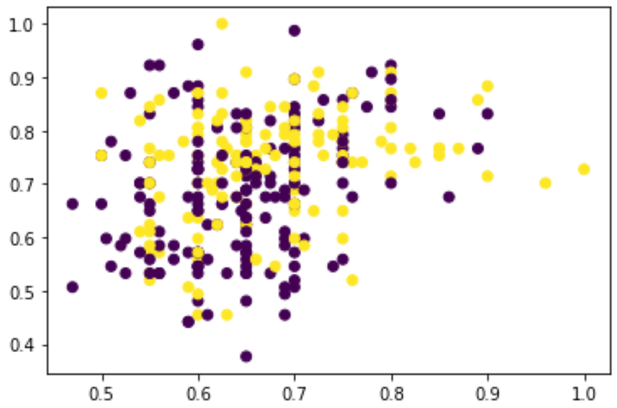
```
X size:  (301, 5)
y size:  (301, 1)
```

In [21]:

```
import matplotlib.pyplot as plt
```

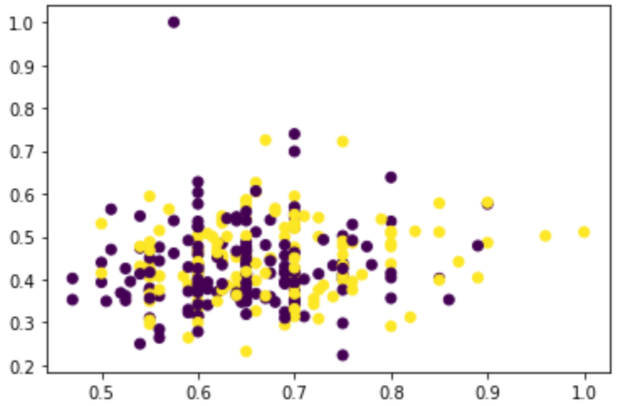
```
In [22]: plt.scatter(X['RestBP'], X['Age'] , c= y.values.tolist())
```

```
Out[22]: <matplotlib.collections.PathCollection at 0x7fbba1e038d0>
```



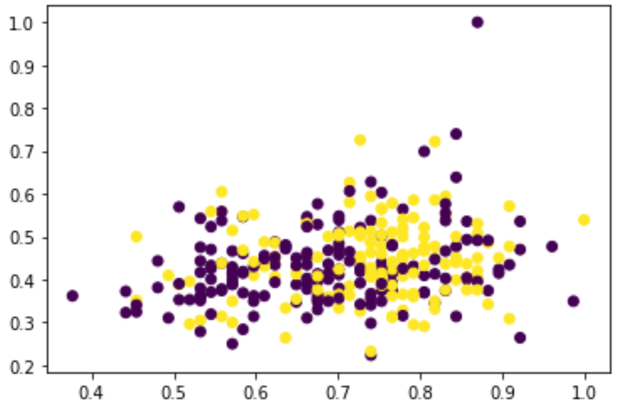
```
In [23]: plt.scatter(X['RestBP'], X['Chol'] , c= y.values.tolist())
```

```
Out[23]: <matplotlib.collections.PathCollection at 0x7fbb99d02828>
```



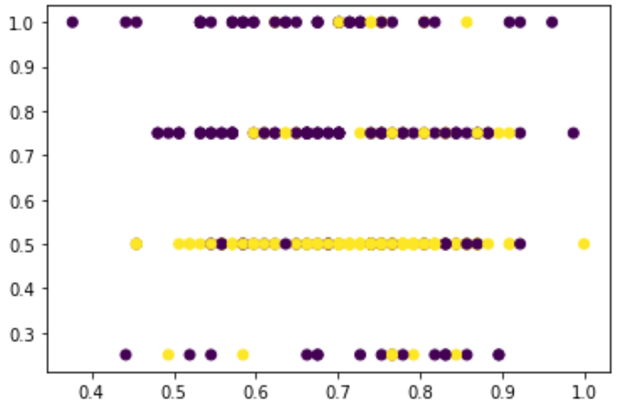
```
In [24]: plt.scatter(X['Age'], X['Chol'] , c= y.values.tolist())
```

```
Out[24]: <matplotlib.collections.PathCollection at 0x7fbb99c7a550>
```



```
In [25]: plt.scatter(X['Age'], X['ChestPain'] , c= y.values.tolist())
```

```
Out[25]: <matplotlib.collections.PathCollection at 0x7fbb99ccd748>
```



In [26]:
```
actual = pd.Series([1 for i in range(50)] + [0 for i in range(50)], name='Actual')
```

In [27]:
```
prediction = pd.Series([1 for i in range(45)] + [0 for i in range(55)], name='Prediction')
```

In [28]:
```
cm = pd.crosstab(actual, prediction)
```

In [29]:
```
cm
```

Out[29]:

Prediction	0	1
Actual		
0	50	0
1	5	45

In [30]:
```
tp = cm[0][0]
tn = cm[1][1]
fp = cm[1][0]
fn = cm[0][1]
print("TP : ", tp)
print("TN : ", tn)
print("FP : ", fp)
print("FN : ", fn)
```

```
TP :  50
TN :  45
FP :  0
FN :  5
```

In [31]:
```
accuracy = (tp + tn) / (tp + tn + fp + fn)
precision = tp / (tp + fp)
recall = tp / (tp + fn)
f1 = (2 * precision * recall) / (precision + recall)
```

In [32]:
```
print("Accuracy : ", accuracy)
print("Precision : ", precision)
print("Recall : ", recall)
print("F1 Score : ", f1)
```

```
Accuracy :  0.95
Precision :  1.0
Recall :  0.9090909090909091
F1 Score :  0.9523809523809523
```

In [33]:
```
from sklearn.metrics import accuracy_score, precision_recall_fscore_support,classification_repo
rt
```

In [34]:
```
accuracy_score(actual, prediction)
```

Out[34]:
```
0.95
```

In [35]:
```
precision_recall_fscore = precision_recall_fscore_support(actual, prediction, average="weighte
d")
precision_recall_fscore
```

Out[35]:
```
(0.9545454545454546, 0.95, 0.949874686716792, None)
```

In [36]:
```
print(classification_report(actual, prediction))
```

```
              precision    recall  f1-score   support

           0       0.91      1.00      0.95        50
           1       1.00      0.90      0.95        50

    accuracy                           0.95       100
   macro avg       0.95      0.95      0.95       100
weighted avg       0.95      0.95      0.95       100
```