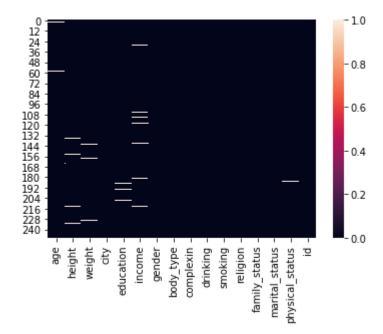
```
In [347]:
            import pandas as pd
            import numpy as np
            from sklearn import pipeline,preprocessing,metrics,model selection,ensemble
            from sklearn_pandas import DataFrameMapper
            import seaborn as sb
In [348]:
            data=pd.read_excel('jugulbondi.xlsx')
            data
Out[348]:
                        height weight
                                            city
                                                 education
                                                              income
                                                                      gender
                                                                              body_type
                                                                                         complexin drinki
                  23.0
               0
                          66.0
                                 70.0
                                                       BSC
                                                              60000.0
                                                                      Female
                                                                                    Slim
                                           pabna
                                                                                                Fair
               1
                  25.0
                          70.0
                                  50.0
                                           dhaka
                                                       MBA
                                                              30000.0
                                                                      Female
                                                                                    Fatty
                                                                                            Lite Fair
               2
                  NaN
                          58.0
                                 58.0
                                           pabna
                                                       HSC
                                                             0.00008
                                                                        Male
                                                                                    Slim
                                                                                              Black
               3
                  35.0
                          62.0
                                 80.0
                                          khulna
                                                       LLB
                                                            100000.0
                                                                        Male
                                                                                 Medium
                                                                                          Lite Brown
                                                                                                         ١
                  30.0
                          60.0
                                       chittagong
                                                       MBA
                                                              40000.0
                                                                      Female
                                                                                    Slim
                                 55.0
                                                                                              Brown
                    ...
                           ...
                                   ...
                                                                   ...
                                                                                      ...
             245
                  30.0
                          62.0
                                  58.0
                                           sylhet
                                                    Honor's
                                                                  0.0
                                                                      Female
                                                                                    Slim
                                                                                          Lite Brown
                                                                                                         1
             246
                  25.0
                          65.0
                                 75.0
                                           sylhet
                                                       BSC
                                                              20000.0
                                                                        Male
                                                                                 Medium
                                                                                          Lite Brown
             247
                          67.0
                                                       BSC
                  24.0
                                 90.0
                                          khulna
                                                                  0.0
                                                                        Male
                                                                                    Fatty
                                                                                          Lite Brown
                                                                                                         `
             248
                  34.0
                          66.0
                                 85.0
                                           dhaka
                                                    Honor's
                                                              55000.0
                                                                        Male
                                                                                 Medium
                                                                                                Fair
             249
                  29.0
                          60.0
                                                                                    Slim
                                                                                                         ١
                                 55.0
                                         rangpur
                                                    Honor's
                                                             60000.0 Female
                                                                                              Black
            250 rows × 16 columns
In [349]:
            data.isnull().sum()
Out[349]: age
                                   3
                                   6
            height
            weight
                                   4
            city
                                   0
                                   3
            education
                                   8
            income
            gender
                                   0
                                   0
            body type
            complexin
                                   0
            drinking
                                   0
            smoking
                                   0
                                   0
            religion
            family_status
                                   0
            marital_status
                                   0
                                   2
            physical_status
            id
                                   0
            dtype: int64
```

```
In [350]: sb.heatmap(data.isnull())
```

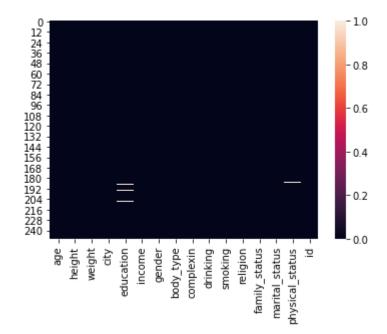
Out[350]: <matplotlib.axes.\_subplots.AxesSubplot at 0x16db815dc70>



```
In [351]: data["age"] = data["age"].fillna(data["age"].mode()[0])
    data["height"] = data["height"].fillna(data["height"].mean())
    data["weight"] = data["weight"].fillna(data["weight"].mode()[0])
    data["income"] = data["income"].fillna(data["income"].mean())
```

In [352]: sb.heatmap(data.isnull())

Out[352]: <matplotlib.axes.\_subplots.AxesSubplot at 0x16db74e8760>



```
In [353]: data = data.dropna(axis=0)
```

In [354]: data

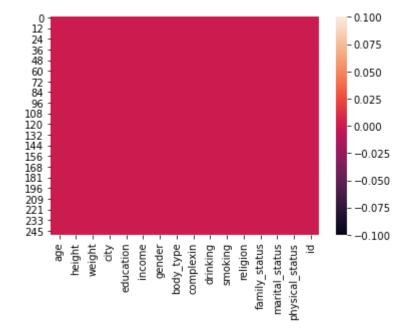
Out[354]:

	age	height	weight	city	education	income	gender	body_type	complexin	drinki
0	23.0	66.0	70.0	pabna	BSC	60000.0	Female	Slim	Fair	_
1	25.0	70.0	50.0	dhaka	MBA	30000.0	Female	Fatty	Lite Fair	
2	25.0	58.0	58.0	pabna	HSC	80000.0	Male	Slim	Black	
3	35.0	62.0	80.0	khulna	LLB	100000.0	Male	Medium	Lite Brown	Υ
4	30.0	60.0	55.0	chittagong	MBA	40000.0	Female	Slim	Brown	
245	30.0	62.0	58.0	sylhet	Honor's	0.0	Female	Slim	Lite Brown	Υ
246	25.0	65.0	75.0	sylhet	BSC	20000.0	Male	Medium	Lite Brown	
247	24.0	67.0	90.0	khulna	BSC	0.0	Male	Fatty	Lite Brown	Υ
248	34.0	66.0	85.0	dhaka	Honor's	55000.0	Male	Medium	Fair	
249	29.0	60.0	55.0	rangpur	Honor's	60000.0	Female	Slim	Black	Υ

245 rows × 16 columns

In [355]: sb.heatmap(data.isnull())

Out[355]: <matplotlib.axes.\_subplots.AxesSubplot at 0x16db75ab070>



```
In [356]: data.info()
          <class 'pandas.core.frame.DataFrame'>
          Int64Index: 245 entries, 0 to 249
          Data columns (total 16 columns):
               Column
                                 Non-Null Count Dtype
           - - -
           0
                                 245 non-null
                                                 float64
               age
           1
                                 245 non-null
                                                 float64
               height
           2
               weight
                                 245 non-null
                                                 float64
           3
               city
                                 245 non-null
                                                 object
           4
               education
                                 245 non-null
                                                 object
           5
                                 245 non-null
                                                 float64
               income
           6
               gender
                                 245 non-null
                                                 object
           7
                                 245 non-null
               body type
                                                 object
           8
               complexin
                                 245 non-null
                                                 object
           9
               drinking
                                 245 non-null
                                                 object
           10 smoking
                                 245 non-null
                                                 object
           11 religion
                                 245 non-null
                                                 object
           12 family status
                                 245 non-null
                                                 object
           13 marital_status
                                 245 non-null
                                                 object
           14 physical status 245 non-null
                                                 object
           15
                                 245 non-null
                                                 int64
               id
          dtypes: float64(4), int64(1), object(11)
          memory usage: 32.5+ KB
In [357]:
          mapper = DataFrameMapper([
                                   (['age', 'height', 'weight', 'income'], preprocessing.Sta
          ndardScaler()),
                                   (['city','education','gender','body_type','complexin',
           'drinking', 'smoking', 'religion', 'family_status', 'marital_status', 'physical_sta
           tus'], preprocessing.OneHotEncoder()),
                                   1)
In [358]: data.columns
Out[358]: Index(['age', 'height', 'weight', 'city', 'education', 'income', 'gender',
                  'body_type', 'complexin', 'drinking', 'smoking', 'religion',
                  'family status', 'marital status', 'physical status', 'id'],
                 dtype='object')
In [359]: X=['age', 'height', 'weight', 'city', 'education', 'income', 'gender',
                  'body_type', 'complexin', 'drinking', 'smoking', 'religion',
                  'family status', 'marital status', 'physical status']
          Y=['id']
In [392]:
          pipeline obj = pipeline.Pipeline([
               ('mapper', mapper),
               ("model", ensemble.RandomForestRegressor(n estimators=70))
          ])
In [393]: | ob=pipeline_obj.fit(data[X],data[Y].values.ravel())
```

```
In [394]: kn=pipeline obj.score(data[X],data[Y].values.ravel())
In [395]: print("Random Forest Regressor Accuracy =",kn)
          Random Forest Regressor Accuracy = 0.6838107404888036
          from sklearn.neighbors import KNeighborsClassifier
In [396]:
          pipeline_Kn = pipeline.Pipeline([
               ('mapper', mapper),
               ("model", KNeighborsClassifier(n neighbors=1, metric='minkowski', p=2))
          ])
In [397]:
          ngb=pipeline Kn.fit(data[X],data[Y].values.ravel())
In [398]: | ngba=pipeline Kn.score(data[X],data[Y].values.ravel())
In [399]:
          print("KNeighborsClassifier Accuracy =",ngba)
          KNeighborsClassifier Accuracy = 0.9551020408163265
In [400]: p=pipeline Kn.predict(data[X])
In [401]: | pipeline_rfcla = pipeline.Pipeline([
               ('mapper', mapper),
               ("model", ensemble.RandomForestClassifier(n estimators=70))
          ])
In [402]:
          rtcla=pipeline rfcla.fit(data[X],data[Y].values.ravel())
In [403]:
          rtclaa=pipeline rfcla.score(data[X],data[Y].values.ravel())
          print("RandomForestClassifierr Accuracy =",rtclaa)
In [404]:
          RandomForestClassifierr Accuracy = 0.9551020408163265
In [405]:
          from sklearn import naive_bayes
          pipeline nv = pipeline.Pipeline([
               ('mapper',mapper),
               ("model", naive bayes.BernoulliNB())
          1)
In [406]:
          nvfit=pipeline_nv.fit(data[X],data[Y].values.ravel())
In [407]: | nvacc=pipeline_nv.score(data[X],data[Y].values.ravel())
In [408]:
          print("Naive Bayes Accuracy =",nvacc)
          Naive Bayes Accuracy = 0.9387755102040817
```

```
In [409]:
          temp={}
           temp['age']=20
           temp['height']=60
           temp['weight']=70
           temp['city']='dhaka'
           temp['education']='LLB'
           temp['income']=700000
           temp['gender']='Female'
           temp['body type']='Slim'
           temp['complexin']='Fair'
           temp['drinking']='No'
           temp['smoking']='No'
           temp['religion']='Muslim'
           temp['family_status']='High'
           temp['marital status']='Unmarried'
           temp['physical_status']='Good'
In [410]:
          testData=pd.DataFrame({'i':temp}).transpose()
In [411]:
          testData
Out[411]:
                             city complexin drinking education family_status gender height income
                  body_type
             age
              20
                       Slim dhaka
                                       Fair
                                                No
                                                        LLB
                                                                    High Female
                                                                                   60
                                                                                       700000
In [412]:
          res = ob.predict(testData)[0]
           print("Using Random Forest Regressor =",res,"==",round(res))
          Using Random Forest Regressor = 133.02857142857144 == 133.0
In [413]:
          res = ngb.predict(testData)[0]
           print("Using KNeighborsClassifier =",res,"==",round(res))
          Using KNeighborsClassifier = 131 == 131
In [414]: res = rtcla.predict(testData)[0]
           print("Using RandomForestClassifierr =",res,"==",round(res))
          Using RandomForestClassifierr = 210 == 210
In [415]:
          res = nvfit.predict(testData)[0]
           print("Using Naive Bayes =",res,"==",round(res))
          Using Naive Bayes = 165 == 165
  In [ ]:
```