import libraries

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

import dataset

In [3]:
```
data=pd.read_csv('mental_data2.csv')
data
```

Out[3]:

	Age Group	Sex	Sexual Orientation	Race	Living Situation	Employment Status	Numl W Each
0	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	OTHER	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
1	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	WHITE ONLY	INSTITUTIONAL SETTING	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
2	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	WHITE ONLY	PRIVATE RESIDENCE	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
3	ADULT	FEMALE	STRAIGHT OR HETEROSEXUAL	OTHER	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
4	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	BLACK ONLY	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
...	...	...	...	...	...	...	
179091	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	WHITE ONLY	PRIVATE RESIDENCE	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
179092	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	WHITE ONLY	PRIVATE RESIDENCE	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
179093	ADULT	MALE	UNKNOWN	WHITE ONLY	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
179094	ADULT	FEMALE	STRAIGHT OR HETEROSEXUAL	WHITE ONLY	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
179095	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	BLACK ONLY	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC

179096 rows × 13 columns

pre-processing

handle categorical data

In [4]:
```
data.dtypes
```

Out[4]:
```
Age Group                           object
Sex                                 object
Sexual Orientation                  object
Race                                object
Living Situation                    object
Employment Status                   object
Number Of Hours Worked Each Week    object
Education Status                    object
Obesity                             object
Alzheimer or Dementia               object
Cancer                              object
Smokes                              object
Illness Status                      object
dtype: object
```

In [5]:
```
data.iloc[:,0].value_counts()
```

Out[5]:
```
ADULT      143213
CHILD       35865
UNKNOWN        18
Name: Age Group, dtype: int64
```

In [6]:
```
data[data['Age Group']=='UNKNOWN'].index
```

Out[6]:
```
Int64Index([  3894,   8034,  18215,  19189,  19311,  20031,  20354,  20398,
             20420,  22788,  22828,  22830,  43541, 107082, 125668, 137594,
            145371, 155672],
           dtype='int64')
```

In [7]:
```
data1=data.drop([3894,  8034,  18215,  19189,  19311,  20031,  20354,  20398,
             20420,  22788,  22828,  22830,  43541, 107082, 125668, 137594,
            145371, 155672],axis=0).reset_index().drop('index',axis=1)
data1
```

Out[7]:

	Age Group	Sex	Sexual Orientation	Race	Living Situation	Employment Status	Numl W Each
0	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	OTHER	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
1	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	WHITE ONLY	INSTITUTIONAL SETTING	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
2	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	WHITE ONLY	PRIVATE RESIDENCE	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
3	ADULT	FEMALE	STRAIGHT OR HETEROSEXUAL	OTHER	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
4	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	BLACK ONLY	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
...	...	...	...	...	...	...	
179073	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	WHITE ONLY	PRIVATE RESIDENCE	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
179074	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	WHITE ONLY	PRIVATE RESIDENCE	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
179075	ADULT	MALE	UNKNOWN	WHITE ONLY	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
179076	ADULT	FEMALE	STRAIGHT OR HETEROSEXUAL	WHITE ONLY	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC
179077	ADULT	MALE	STRAIGHT OR HETEROSEXUAL	BLACK ONLY	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	APPLIC

179078 rows × 13 columns

In [8]:
```
data1.iloc[:,0].value_counts()
```

Out[8]:
```
ADULT    143213
CHILD     35865
Name: Age Group, dtype: int64
```

In [9]:
```
data1.iloc[:,1].value_counts()
```

Out[9]:
```
FEMALE     89915
MALE       88787
UNKNOWN      376
Name: Sex, dtype: int64
```

In [10]:
```
data2=data1.replace({'Age Group':'CHILD'},0).replace({'Age Group':'ADULT'},1)
```

In [11]:
```
a=data2[data2['Sex']=='UNKNOWN'].index
data3=data2.drop(axis=0,index=a).reset_index().drop('index',axis=1)
data4=data3.replace({'Sex':'MALE'},1).replace({'Sex':'FEMALE'},0)
```

In [12]:
```
data4.iloc[:,2].value_counts()
```

Out[12]:
```
STRAIGHT OR HETEROSEXUAL    144959
UNKNOWN                      17273
CLIENT DID NOT ANSWER         6181
LESBIAN OR GAY                5048
BISEXUAL                      4157
OTHER                         1084
Name: Sexual Orientation, dtype: int64
```

In [13]:
```
data4.iloc[:,3].value_counts()
```

Out[13]:
```
WHITE ONLY      92125
BLACK ONLY      46712
OTHER           29543
UNKNOWN RACE     5702
MULTI-RACIAL     4620
Name: Race, dtype: int64
```

In [14]:
```
data5=data4.replace({'Sexual Orientation':'STRAIGHT OR HETEROSEXUAL'},1).replace(
data5.iloc[:,2].value_counts()
```

Out[14]:
```
1    144959
0     33743
Name: Sexual Orientation, dtype: int64
```

In [15]:
```
data5['Race'].replace({'WHITE ONLY':'white','BLACK ONLY':'black','OTHER':'other',
data5
```

Out[15]:

	Age Group	Sex	Sexual Orientation	Race	Living Situation	Employment Status	Number Of Hours Worked Each Week	Edu
0	1	1	1	other	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	M SC TC SC
1	1	1	1	white	INSTITUTIONAL SETTING	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	COL GRAD DE
2	1	1	1	white	PRIVATE RESIDENCE	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	COL GRAD DE
3	1	0	1	other	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	M SC TC SC
4	1	1	1	black	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	M SC TC SC
...	...	...	...	...	...	...	...	
178697	1	1	1	white	PRIVATE RESIDENCE	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	M SC TC SC
178698	1	1	1	white	PRIVATE RESIDENCE	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	COL
178699	1	1	0	white	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	COL GRAD DE
178700	1	0	1	white	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	COL
178701	1	1	1	black	OTHER LIVING SITUATION	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	M SC TC SC

178702 rows × 13 columns

In [16]:
```
data5.iloc[:,4].value_counts()
```

Out[16]:
```
PRIVATE RESIDENCE         141190
OTHER LIVING SITUATION     32233
UNKNOWN                     3400
INSTITUTIONAL SETTING       1879
Name: Living Situation, dtype: int64
```

In [17]:
```
data6=data5[(data5['Living Situation']=='PRIVATE RESIDENCE')| (data5['Living Situ
data6=data6.replace({'Living Situation':'PRIVATE RESIDENCE'},1).replace({'Living
data6
```

Out[17]:

	Age Group	Sex	Sexual Orientation	Race	Living Situation	Employment Status	Number Of Hours Worked Each Week	Education Status
0	1	1	1	other	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	MIDDLE SCHOOL TO HIGH SCHOOL
1	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	COLLEGE OR GRADUATE DEGREE
2	1	0	1	other	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	MIDDLE SCHOOL TO HIGH SCHOOL
3	1	1	1	black	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	MIDDLE SCHOOL TO HIGH SCHOOL
4	1	1	1	black	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	MIDDLE SCHOOL TO HIGH SCHOOL
...	...	...	...	...	...	...	...	...
173418	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	MIDDLE SCHOOL TO HIGH SCHOOL
173419	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	SOME COLLEGE
173420	1	1	0	white	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	COLLEGE OR GRADUATE DEGREE
173421	1	0	1	white	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	SOME COLLEGE
173422	1	1	1	black	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	NOT APPLICABLE	MIDDLE SCHOOL TO HIGH SCHOOL

173423 rows × 13 columns

In [18]:
```
data6.iloc[:,5].value_counts()
```

Out[18]:
```
NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING FOR WORK    124126
EMPLOYED                                                   26320
UNEMPLOYED, LOOKING FOR WORK                               16871
UNKNOWN EMPLOYMENT STATUS                                   4288
NON-PAID/VOLUNTEER                                          1818
Name: Employment Status, dtype: int64
```

In [19]:
```
data6.iloc[:,6].value_counts()
```

Out[19]:
```
NOT APPLICABLE              147103
35 HOURS OR MORE              9437
15-34 HOURS                   9176
01-14 HOURS                   5006
UNKNOWN EMPLOYMENT HOURS      2701
Name: Number Of Hours Worked Each Week, dtype: int64
```

In [20]:
```
data6.iloc[:,7].value_counts()
```

Out[20]:
```
MIDDLE SCHOOL TO HIGH SCHOOL    95109
COLLEGE OR GRADUATE DEGREE      23797
SOME COLLEGE                    23715
PRE-K TO FIFTH GRADE            15264
UNKNOWN                         11903
OTHER                            2895
NO FORMAL EDUCATION               740
Name: Education Status, dtype: int64
```

In [21]:
```
data7=data6[(data6['Employment Status']=='NOT IN LABOR FORCE:UNEMPLOYED AND NOT L
            (data6['Employment Status']=='EMPLOYED')|
             (data6['Employment Status']=='UNEMPLOYED, LOOKING FOR WORK')].reset_
```

In [22]:
```
data8=data7[(data7['Education Status']=='MIDDLE SCHOOL TO HIGH SCHOOL')|
           (data7['Education Status']=='COLLEGE OR GRADUATE DEGREE')|
           (data7['Education Status']=='SOME COLLEGE')|
           (data7['Education Status']=='PRE-K TO FIFTH GRADE')].reset_index().dro
data9=data8.replace({'Education Status':'MIDDLE SCHOOL TO HIGH SCHOOL'},2).replac
data10=data9.drop('Number Of Hours Worked Each Week',axis=1)
data10
```

Out[22]:

	Age Group	Sex	Sexual Orientation	Race	Living Situation	Employment Status	Education Status	Obesity	
0	1	1	1	other	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	NO	
1	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	4	NO	
2	1	0	1	other	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	NO	
3	1	1	1	black	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	NO	
4	1	1	1	black	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	YES	
...	...	...	...	...	...	...	...	...	
153679	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	NO	
153680	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	3	NO	
153681	1	1	0	white	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	4	NO	
153682	1	0	1	white	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	3	NO	
153683	1	1	1	black	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	UNKNOWN	U

153684 rows × 12 columns

In [23]:
```
data10.iloc[:,8].value_counts()
```

Out[23]:
```
NO          142031
UNKNOWN      11042
YES            611
Name: Alzheimer or Dementia, dtype: int64
```

In [24]:
```
data11=data10[(data10['Alzheimer or Dementia']=='YES') | (data10['Alzheimer or De
data12=data11.replace({'Alzheimer or Dementia':'YES'},0).replace({'Alzheimer or D
data12
```

Out[24]:

	Age Group	Sex	Sexual Orientation	Race	Living Situation	Employment Status	Education Status	Obesity	Alzh Dem
0	1	1	1	other	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	NO	
1	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	4	NO	
2	1	0	1	other	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	NO	
3	1	1	1	black	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	NO	
4	1	1	1	black	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	YES	
...	...	...	...	...	...	...	...	...	
142637	1	0	1	black	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	1	NO	
142638	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	NO	
142639	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	3	NO	
142640	1	1	0	white	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	4	NO	
142641	1	0	1	white	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	3	NO	

142642 rows × 12 columns

In [25]:
```
data12.iloc[:,7].value_counts()
```

Out[25]:
```
NO     119856
YES     22786
Name: Obesity, dtype: int64
```

In [26]:
```
data13=data12.replace({'Obesity':'YES'},0).replace({'Obesity':'NO'},1)
data13
```

Out[26]:

	Age Group	Sex	Sexual Orientation	Race	Living Situation	Employment Status	Education Status	Obesity	Alzh Dem
0	1	1	1	other	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	1	
1	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	4	1	
2	1	0	1	other	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	1	
3	1	1	1	black	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	1	
4	1	1	1	black	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	0	
...	...	...	...	...	...	...	...	...	
142637	1	0	1	black	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	1	1	
142638	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	1	
142639	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	3	1	
142640	1	1	0	white	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	4	1	
142641	1	0	1	white	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	3	1	

142642 rows × 12 columns

In [27]:
```
data13.iloc[:,9].value_counts()
```

Out[27]:
```
NO     139494
YES      3148
Name: Cancer, dtype: int64
```

In [28]:
```
data13.iloc[:,10].value_counts()
```

Out[28]:
```
NO         97049
YES        40367
UNKNOWN     5226
Name: Smokes, dtype: int64
```

In [47]:
```
data14=data13.replace({'Cancer':"NO"},1).replace({'Cancer':'YES'},0)
data15=data14[(data14['Smokes']=='YES') | (data14['Smokes']=='NO')].reset_index()
data16=data15.replace({'Smokes':'YES'},0).replace({'Smokes':'NO'},1)
data16
```

Out[47]:

	Age Group	Sex	Sexual Orientation	Race	Living Situation	Employment Status	Education Status	Obesity	Alzh Dem
0	1	1	1	other	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	1	
1	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	4	1	
2	1	0	1	other	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	1	
3	1	1	1	black	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	1	
4	1	1	1	black	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	0	
...	...	...	...	...	...	...	...	...	
137411	1	0	1	black	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	1	1	
137412	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	2	1	
137413	1	1	1	white	1	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	3	1	
137414	1	1	0	white	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	4	1	
137415	1	0	1	white	0	NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING ...	3	1	

137416 rows × 12 columns

In [48]:
```
data16.iloc[:,11].value_counts()
```

Out[48]:
```
NO         121887
YES         12004
UNKNOWN      3525
Name: Illness Status, dtype: int64
```

In [49]:
```
data17=data16[(data16['Illness Status']=='NO') | (data16['Illness Status']=='YES'
data18=data17.replace({'Illness Status':'NO'},1).replace({'Illness Status':'YES']
data19=pd.get_dummies(data18)
data19
```

Out[49]:

	Age Group	Sex	Sexual Orientation	Living Situation	Education Status	Obesity	Alzheimer or Dementia	Cancer	Smokes	Illnes Statu
0	1	1	1	0	2	1	1	1	1	
1	1	1	1	1	4	1	1	1	1	
2	1	0	1	0	2	1	1	1	1	
3	1	1	1	0	2	1	0	1	1	
4	1	1	1	0	2	0	1	1	1	
...	...	...	...	...	...	...	...	...	...	
133886	1	0	1	1	2	1	1	1	0	
133887	1	0	1	1	2	1	1	1	0	
133888	1	1	1	1	2	1	1	1	1	
133889	1	1	1	1	3	1	1	1	0	
133890	1	0	1	0	3	1	1	1	0	

133891 rows × 16 columns

handle missing value

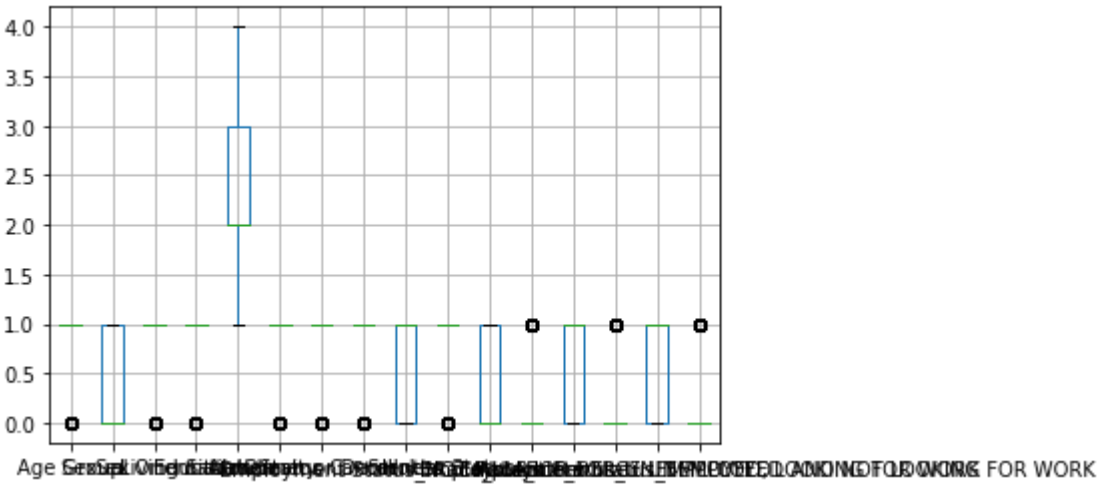
In [50]:
```
data19.isna().sum()
```

Out[50]:
```
Age Group                                                          0
Sex                                                                0
Sexual Orientation                                                 0
Living Situation                                                   0
Education Status                                                   0
Obesity                                                            0
Alzheimer or Dementia                                              0
Cancer                                                             0
Smokes                                                             0
Illness Status                                                     0
Race_black                                                         0
Race_other                                                         0
Race_white                                                         0
Employment Status_EMPLOYED                                         0
Employment Status_NOT IN LABOR FORCE:UNEMPLOYED AND NOT LOOKING FOR WORK    0
Employment Status_UNEMPLOYED, LOOKING FOR WORK                     0
dtype: int64
```

handle outlier data

In [51]:
```
data19.boxplot()
```

Out[51]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ec47fda880>
```



handle duplicated data

In [37]:
```
data19.duplicated().sum()
```

Out[37]:
```
131722
```

In [80]:
```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
from sklearn.feature_selection import f_classif
```

In [82]:
```
x=data19.iloc[:,[0,1,2,3,4,5,6,7,8,10,11,12,13,14,15]]
y=data19.iloc[:,9]
bestfeatures=SelectKBest(chi2,k=15)
fit=bestfeatures.fit(x,y)
dfscores=pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(x.columns)
featurescores = pd.concat([dfcolumns, dfscores], axis = 1)
featurescores.columns = ["feature_name", "feature_score"]
print(featurescores.nlargest(8, "feature_score"))
```

```
                                    feature_name  feature_score
1                                            Sex    1315.372643
8                                         Smokes     820.431506
3                                Living Situation     683.874237
9                                      Race_black     408.044249
0                                       Age Group     378.869340
14  Employment Status_UNEMPLOYED, LOOKING FOR WORK     293.995647
10                                     Race_other     162.136430
11                                     Race_white      33.597076
```

In [83]:
```
x=data19.iloc[:,[0,1,2,3,4,5,6,7,8,10,11,12,13,14,15]]
y=data19.iloc[:,9]
bestfeatures=SelectKBest(f_classif,k=15)
fit=bestfeatures.fit(x,y)
dfscores=pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(x.columns)
featurescores = pd.concat([dfcolumns, dfscores], axis = 1)
featurescores.columns = ["feature_name", "feature_score"]
print(featurescores.nlargest(8, "feature_score"))
```

```
                                        feature_name  feature_score
3                                    Living Situation    4346.055209
8                                              Smokes    2894.240469
1                                                 Sex    2609.264254
0                                           Age Group    1771.104557
9                                          Race_black     548.674936
14     Employment Status_UNEMPLOYED, LOOKING FOR WORK     328.636637
10                                         Race_other     207.820565
13  Employment Status_NOT IN LABOR FORCE:UNEMPLOYE...      96.019852
```

feature selection

In [ ]:
```
#embeded
```

handle imbalance data

In [59]:
```
data19.iloc[:,9].value_counts()
```

Out[59]:
```
1    121887
0     12004
Name: Illness Status, dtype: int64
```

In [85]:
```
class_0=data19[data19['Illness Status']==0]
class_1=data19[data19['Illness Status']==1]
```

```
(12004, 16)
```

In [65]:
```
class_count_0 , class_count_1 = data19.iloc[:,9].value_counts()
```

In [66]:
```
class0over=class_0.sample(class_count_1,replace=True)
```

In [69]:
```
dataover=pd.concat([class_0,class_1],axis=0).reset_index().drop('index',axis=1)
dataover
```

Out[69]:

	Age Group	Sex	Sexual Orientation	Living Situation	Education Status	Obesity	Alzheimer or Dementia	Cancer	Smokes	Illnes Statu
0	1	1	1	0	2	1	1	1	1	
1	1	1	1	1	4	1	1	1	1	
2	1	0	1	0	2	1	1	1	1	
3	1	1	1	0	2	0	1	1	1	
4	1	1	1	0	2	1	1	1	1	
...	...	...	...	...	...	...	...	...	...	
133886	1	0	1	1	2	1	1	1	0	
133887	1	0	1	1	2	1	1	1	0	
133888	1	1	1	1	2	1	1	1	1	
133889	1	1	1	1	3	1	1	1	0	
133890	1	0	1	0	3	1	1	1	0	

133891 rows × 16 columns

splitting dataset into training set and test set

In [90]:
```
x=data19.iloc[:,[0,1,2,3,8,4,6,5,10,11,7,12,13,14,15]]
y=data19.iloc[:,9]
```

In [91]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=0
```

In [92]:
```
x_train
```

Out[92]:

	Age Group	Sex	Sexual Orientation	Living Situation	Smokes	Education Status	Alzheimer or Dementia	Obesity	Race_black
22049	0	0	0	1	1	2	1	1	0
89983	0	1	0	1	1	1	1	1	0
5392	1	0	0	1	0	4	1	0	0
131884	1	1	1	1	1	4	1	1	1
83841	0	1	0	1	1	1	1	1	0
...	...	...	...	...	...	...	...	...	...
41993	1	0	1	1	1	2	1	0	1
97639	1	1	1	1	1	2	1	1	0
95939	1	1	1	1	1	2	1	1	0
117952	1	0	0	0	1	2	1	0	1
43567	0	1	1	1	1	1	1	1	1

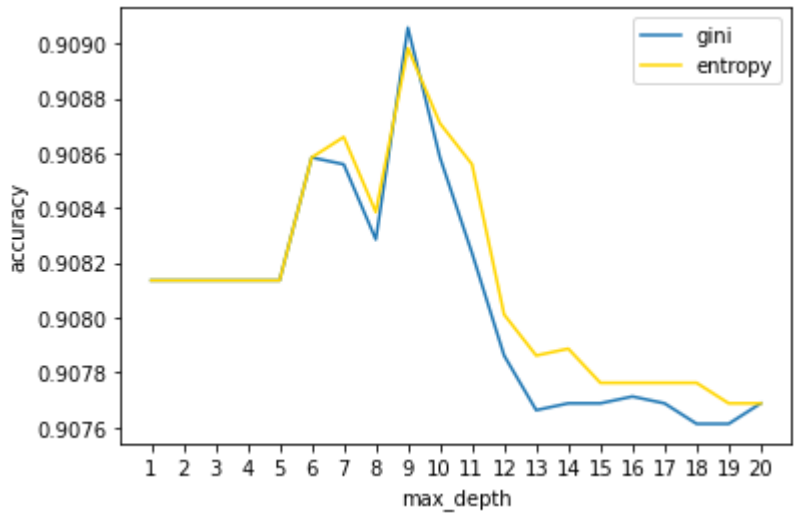
93723 rows × 15 columns

training the decision tree model on training set

In [93]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
max_depth=[]
acc_gini=[]
acc_entropy=[]
for i in range(1,21):
    dtree=DecisionTreeClassifier(criterion='gini',max_depth=i)
    dtree.fit(x_train,y_train)
    y_pred=dtree.predict(x_test)
    acc_gini.append(accuracy_score(y_test,y_pred))

    dtree=DecisionTreeClassifier(criterion='entropy',max_depth=i)
    dtree.fit(x_train,y_train)
    y_pred=dtree.predict(x_test)
    acc_entropy.append(accuracy_score(y_test,y_pred))

    max_depth.append(i)
df=pd.DataFrame({'acc_gini':pd.Series(acc_gini),'acc_entropy':pd.Series(acc_entro
plt.plot('max_depth','acc_gini',data=df,label='gini')
plt.plot('max_depth','acc_entropy',data=df,label='entropy',color='gold')
plt.xlabel('max_depth')
plt.ylabel('accuracy')
plt.xticks([j for j in range(1,21)])
plt.legend()
plt.show()
```



building model

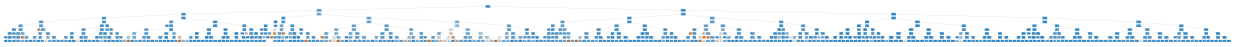
In [94]:
```
classifier=DecisionTreeClassifier(criterion='gini',max_depth=9)
classifier.fit(x_train,y_train)
```

Out[94]:
```
DecisionTreeClassifier(max_depth=9)
```

In [95]:
```
from six import StringIO
from sklearn.tree import export_graphviz
import pydotplus
from IPython.display import Image
dot_data = StringIO()
export_graphviz(classifier, out_file = dot_data, filled = True, precision = 2)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

```
dot: graph is too large for cairo-renderer bitmaps. Scaling by 0.687732 to fit
```

Out[95]:



validation

confusion matrix

In [96]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

Out[96]:
```
array([[  132,  3558],
       [  150, 36328]], dtype=int64)
```

accuracy score

In [97]:
```
accuracy_score(y_test,y_pred)
```

Out[97]:
```
0.9076877116112329
```

precision and recall

In [99]:
```
from sklearn.metrics import precision_score
precision_score(y_test, y_pred)
```

Out[99]:
```
0.910795767938625
```

recall

In [100]:
```
from sklearn.metrics import f1_score
f1_score(y_test, y_pred)
```

Out[100]:
```
0.9514430883662458
```

k-fold cross validation

k_fold cross validation

In [101]:
```
from sklearn.model_selection import cross_val_score
estimator = cross_val_score(estimator = classifier, X = x_train, y = y_train, cv
estimator.mean()
```

Out[101]:
```
0.9112597422108408
```

RANDOM FOREST MODEL

In [102]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [104]:
```
classifier1=RandomForestClassifier(n_estimators=100,criterion='gini',max_depth=9)
classifier1.fit(x_train,y_train)
```

Out[104]:
```
RandomForestClassifier(max_depth=9)
```

In [105]:
```
y_pred1=classifier1.predict(x_test)
```

In [107]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred1)
```

Out[107]:
```
array([[   20,  3670],
       [    6, 36472]], dtype=int64)
```

In [108]:
```
accuracy_score(y_test,y_pred1)
```

Out[108]:
```
0.9084843656642103
```