SC1015 MINI PROJECT

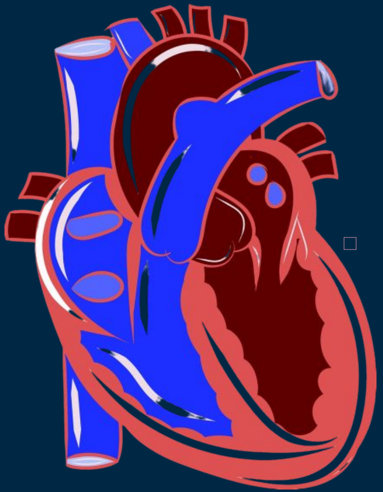
HEART DISEASE RISK PREDICTOR

BCF3 Group7 :
Jai Jindal (U2123034K) , Livia Iriawan (U2121980C) , Shayanthaviy Sivasankar (U2123253J)

Heart Disease

Heart disease is ***the leading cause of death*** for men, women, and people of most racial and ethnic groups in the United States.

One person dies every 36 seconds in the United States from cardiovascular disease.



For heart disease, if caught early enough, lifestyle changes may be enough to **prevent the disease from ever fully developing** and requiring treatment.

Our Motivation

Data Set used:

Behavioral Risk Factor Surveillance System (BRFSS) 2011-2015

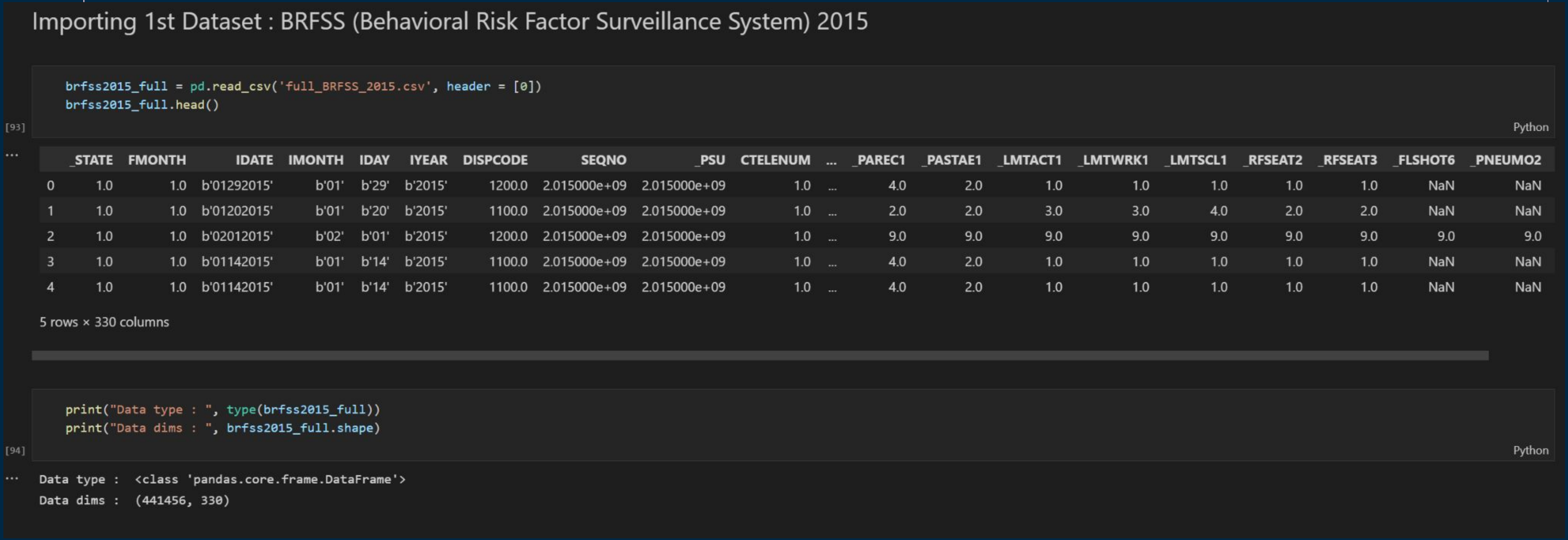
- (Link: https://www.kaggle.com/datasets/cdc/behavioral-risk-factor-surveillance-system/)

Problem Definition:

Identifying key health indicators that significantly affect heart diseases so that we can detect and prevent the factors having the greatest impact.

- What risk factors are most predictive of heart disease risk?
- Can survey questions from the BRFSS provide accurate predictions of whether an individual has heart disease?
- Can we use a subset of the risk factors to accurately build a webapp that can predict whether an individual has heart disease?

Setting the Stage: Cleaning Data



Importing 1st Dataset : BRFSS (Behavioral Risk Factor Surveillance System) 2015

```
brfss2015_full = pd.read_csv('full_BRFSS_2015.csv', header = [0])
brfss2015_full.head()
```

[93] Python

	_STATE	FMONTH	IDATE	IMONTH	IDAY	IYEAR	DISPCODE	SEQNO	_PSU	CTELENUM	...	_PAREC1	_PASTAE1	_LMTACT1	_LMTWRK1	_LMTSCL1	_RFSEAT2	_RFSEAT3	_FLSHOT6	_PNEUMO2
0	1.0	1.0	b'01292015'	b'01'	b'29'	b'2015'	1200.0	2.015000e+09	2.015000e+09	1.0	...	4.0	2.0	1.0	1.0	1.0	1.0	1.0	NaN	NaN
1	1.0	1.0	b'01202015'	b'01'	b'20'	b'2015'	1100.0	2.015000e+09	2.015000e+09	1.0	...	2.0	2.0	3.0	3.0	4.0	2.0	2.0	NaN	NaN
2	1.0	1.0	b'02012015'	b'02'	b'01'	b'2015'	1200.0	2.015000e+09	2.015000e+09	1.0	...	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0
3	1.0	1.0	b'01142015'	b'01'	b'14'	b'2015'	1100.0	2.015000e+09	2.015000e+09	1.0	...	4.0	2.0	1.0	1.0	1.0	1.0	1.0	NaN	NaN
4	1.0	1.0	b'01142015'	b'01'	b'14'	b'2015'	1100.0	2.015000e+09	2.015000e+09	1.0	...	4.0	2.0	1.0	1.0	1.0	1.0	1.0	NaN	NaN

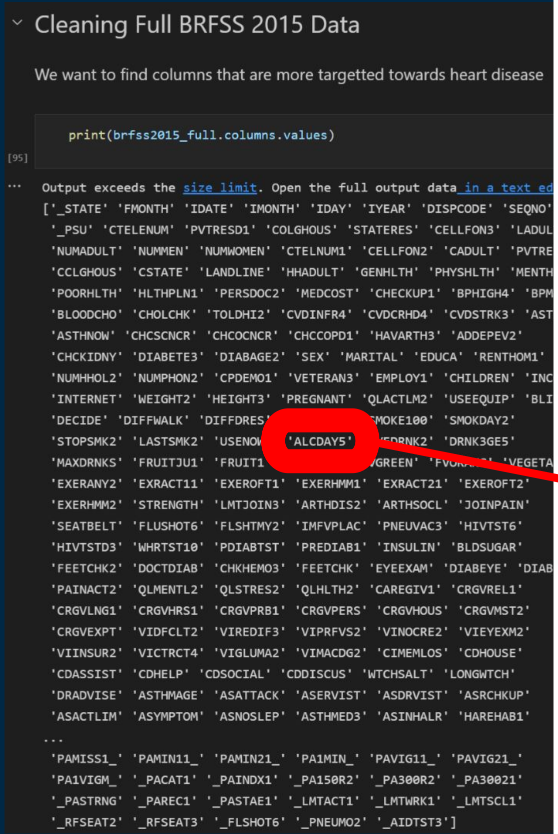
5 rows × 330 columns

```
print("Data type : ", type(brfss2015_full))
print("Data dims : ", brfss2015_full.shape)
```

[94] Python

```
Data type :  <class 'pandas.core.frame.DataFrame'>
Data dims :  (441456, 330)
```

Setting the Stage: Cleaning Data



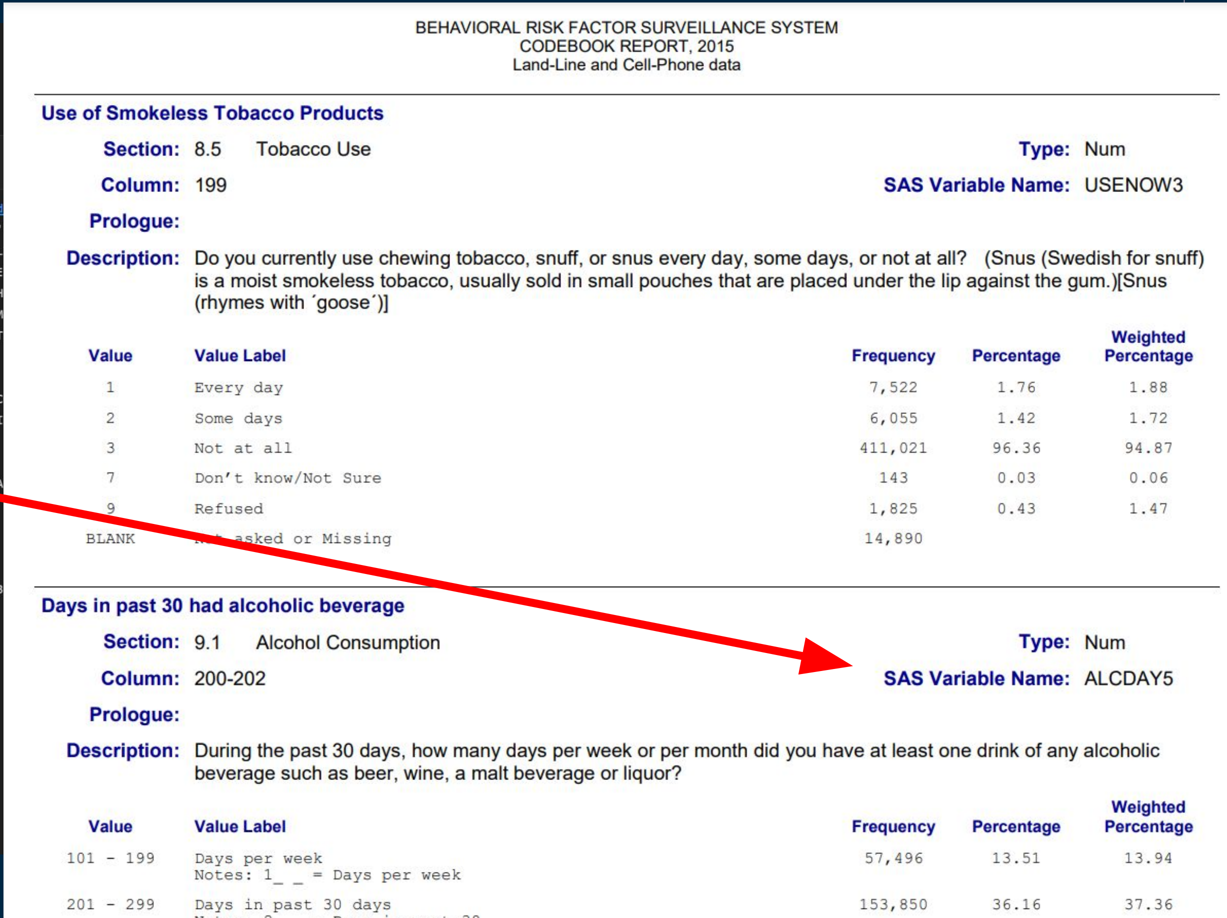
Cleaning Full BRFSS 2015 Data

We want to find columns that are more targetted towards heart disease

```
print(brfss2015_full.columns.values)
```

[95]

```
Output exceeds the size limit. Open the full output data in a text ed
['_STATE' 'FMONTH' 'IDATE' 'IMONTH' 'IDAY' 'IYEAR' 'DISPCODE' 'SEQNO'
 '_PSU' 'CTELENUM' 'PVTRESD1' 'COLGHOUS' 'STATERES' 'CELLFON3' 'LADUL
 'NUMADULT' 'NUMMEN' 'NUMWOMEN' 'CTELNUM1' 'CELLFON2' 'CADULT' 'PVTRE
 'CCLGHOUS' 'CSTATE' 'LANDLINE' 'HHADULT' 'GENHLTH' 'PHYSHLTH' 'MENTH
 'POORHLTH' 'HLTHPLN1' 'PERSDOC2' 'MEDCOST' 'CHECKUP1' 'BPHIGH4' 'BPM
 'BLOODCHO' 'CHOLCHK' 'TOLDHI2' 'CVDINFR4' 'CVDCRHD4' 'CVDSTRK3' 'AST
 'ASTHNOW' 'CHCSCNCR' 'CHCOCNCR' 'CHCCOPD1' 'HAVARTH3' 'ADDEPEV2'
 'CHCKIDNY' 'DIABETE3' 'DIABAGE2' 'SEX' 'MARITAL' 'EDUCA' 'RENTHOM1'
 'NUMHHOL2' 'NUMPHON2' 'CPDEMO1' 'VETERAN3' 'EMPLOY1' 'CHILDREN' 'INC
 'INTERNET' 'WEIGHT2' 'HEIGHT3' 'PREGNANT' 'QLACTLM2' 'USEEQUIP' 'BLI
 'DECIDE' 'DIFFWALK' 'DIFFDRES'            MOKE100' 'SMOKDAY2'
 'STOPSMK2' 'LASTSMK2' 'USENOW    'ALCDAY5'   EDRNK2' 'DRNK3GE5'
 'MAXDRNKS' 'FRUITJU1' 'FRUIT1               VGREEN' 'FVO      'VEGETA
 'EXERANY2' 'EXRACT11' 'EXEROFT1' 'EXERHMM1' 'EXRACT21' 'EXEROFT2'
 'EXERHMM2' 'STRENGTH' 'LMTJOIN3' 'ARTHDIS2' 'ARTHSOCL' 'JOINPAIN'
 'SEATBELT' 'FLUSHOT6' 'FLSHTMY2' 'IMFVPLAC' 'PNEUVAC3' 'HIVTST6'
 'HIVTSTD3' 'WHRTST10' 'PDIABTST' 'PREDIAB1' 'INSULIN' 'BLDSUGAR'
 'FEETCHK2' 'DOCTDIAB' 'CHKHEMO3' 'FEETCHK' 'EYEEXAM' 'DIABEYE' 'DIAB
 'PAINACT2' 'QLMENTL2' 'QLSTRES2' 'QLHLTH2' 'CAREGIV1' 'CRGVREL1'
 'CRGVLNG1' 'CRGVHRS1' 'CRGVPRB1' 'CRGVPERS' 'CRGVHOUS' 'CRGVMST2'
 'CRGVEXPT' 'VIDFCLT2' 'VIREDIF3' 'VIPRFVS2' 'VINOCRE2' 'VIEYEXM2'
 'VIINSUR2' 'VICTRCT4' 'VIGLUMA2' 'VIMACDG2' 'CIMEMLOS' 'CDHOUSE'
 'CDASSIST' 'CDHELP' 'CDSOCIAL' 'CDDISCUS' 'WTCHSALT' 'LONGWTCH'
 'DRADVISE' 'ASTHMAGE' 'ASATTACK' 'ASERVIST' 'ASDRVIST' 'ASRCHKUP'
 'ASACTLIM' 'ASYMPTOM' 'ASNOSLEP' 'ASTHMED3' 'ASINHALR' 'HAREHAB1'
...
 'PAMISS1_' 'PAMIN11_' 'PAMIN21_' 'PA1MIN_' 'PAVIG11_' 'PAVIG21_'
 'PA1VIGM_' '_PACAT1' '_PAINDX1' '_PA150R2' '_PA300R2' '_PA30021'
 '_PASTRNG' '_PAREC1' '_PASTAE1' '_LMTACT1' '_LMTWRK1' '_LMTSCL1'
 '_RFSEAT2' '_RFSEAT3' '_FLSHOT6' '_PNEUMO2' '_AIDTST3']
```



BEHAVIORAL RISK FACTOR SURVEILLANCE SYSTEM
CODEBOOK REPORT, 2015
Land-Line and Cell-Phone data

Use of Smokeless Tobacco Products

Section: 8.5 Tobacco Use — **Type:** Num

Column: 199 — **SAS Variable Name:** USENOW3

Prologue:

Description: Do you currently use chewing tobacco, snuff, or snus every day, some days, or not at all? (Snus (Swedish for snuff) is a moist smokeless tobacco, usually sold in small pouches that are placed under the lip against the gum.)[Snus (rhymes with ´goose´)]

Value	Value Label	Frequency	Percentage	Weighted Percentage
1	Every day	7,522	1.76	1.88
2	Some days	6,055	1.42	1.72
3	Not at all	411,021	96.36	94.87
7	Don’t know/Not Sure	143	0.03	0.06
9	Refused	1,825	0.43	1.47
BLANK	Not asked or Missing	14,890		

Days in past 30 had alcoholic beverage

Section: 9.1 Alcohol Consumption — **Type:** Num

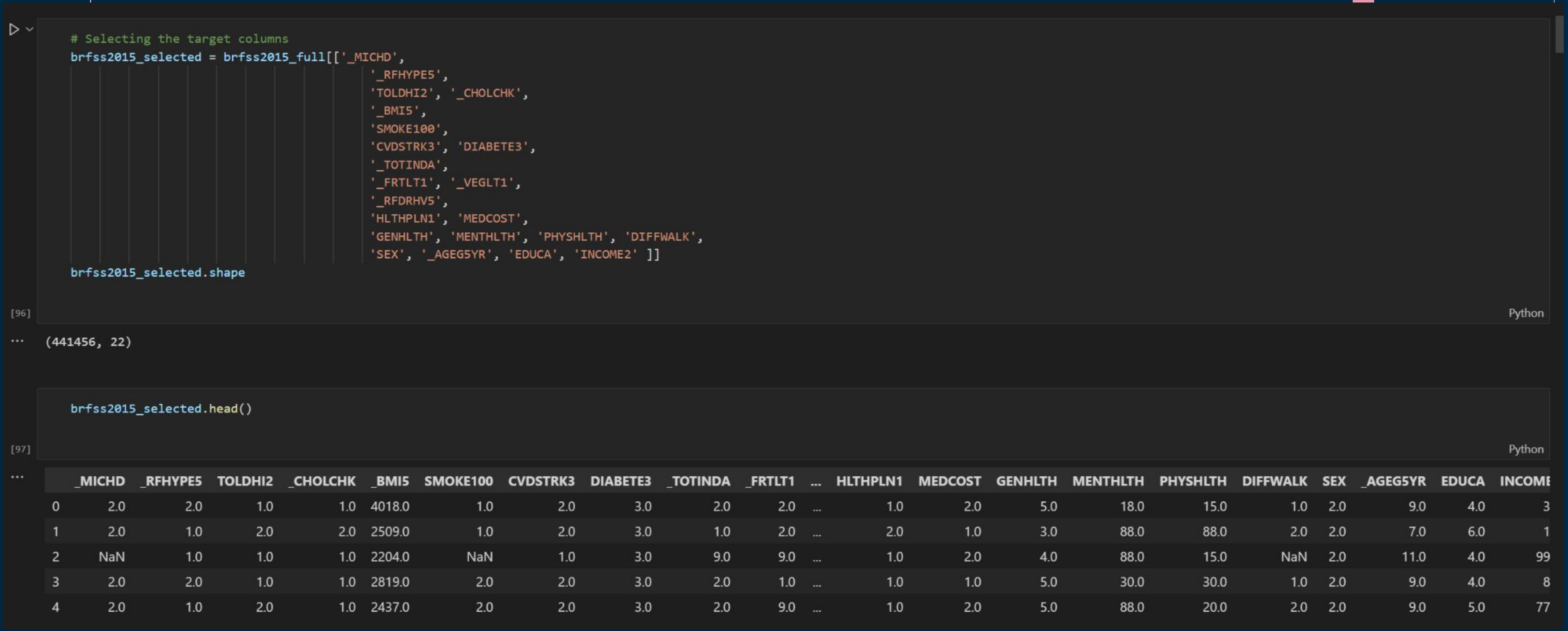
Column: 200-202 — **SAS Variable Name:** ALCDAY5

Prologue:

Description: During the past 30 days, how many days per week or per month did you have at least one drink of any alcoholic beverage such as beer, wine, a malt beverage or liquor?

Value	Value Label	Frequency	Percentage	Weighted Percentage
101 - 199	Days per week Notes: 1_ _ = Days per week	57,496	13.51	13.94
201 - 299	Days in past 30 days	153,850	36.16	37.36

Setting the Stage: Cleaning Data



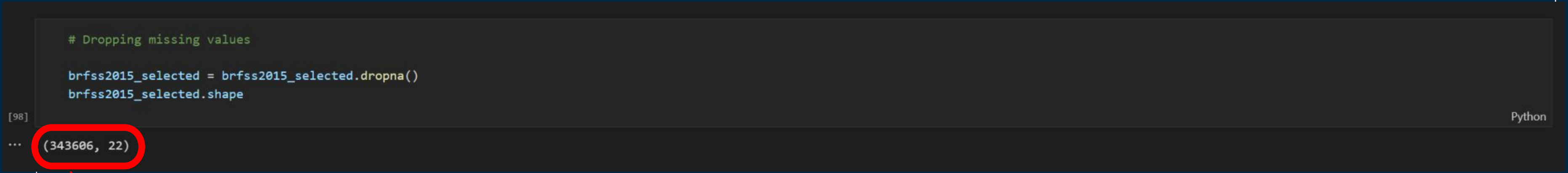
```python
# Selecting the target columns
brfss2015_selected = brfss2015_full[['_MICHD',
                                         '_RFHYPE5',
                                         'TOLDHI2', '_CHOLCHK',
                                         '_BMI5',
                                         'SMOKE100',
                                         'CVDSTRK3', 'DIABETE3',
                                         '_TOTINDA',
                                         '_FRTLT1', '_VEGLT1',
                                         '_RFDRHV5',
                                         'HLTHPLN1', 'MEDCOST',
                                         'GENHLTH', 'MENTHLTH', 'PHYSHLTH', 'DIFFWALK',
                                         'SEX', '_AGEG5YR', 'EDUCA', 'INCOME2' ]]
brfss2015_selected.shape
```

```
(441456, 22)
```

```python
brfss2015_selected.head()
```

	_MICHD	_RFHYPE5	TOLDHI2	_CHOLCHK	_BMI5	SMOKE100	CVDSTRK3	DIABETE3	_TOTINDA	_FRTLT1	...	HLTHPLN1	MEDCOST	GENHLTH	MENTHLTH	PHYSHLTH	DIFFWALK	SEX	_AGEG5YR	EDUCA	INCOME
0	2.0	2.0	1.0	1.0	4018.0	1.0	2.0	3.0	2.0	2.0	...	1.0	2.0	5.0	18.0	15.0	1.0	2.0	9.0	4.0	3
1	2.0	1.0	2.0	2.0	2509.0	1.0	2.0	3.0	1.0	2.0	...	2.0	1.0	3.0	88.0	88.0	2.0	2.0	7.0	6.0	1
2	NaN	1.0	1.0	1.0	2204.0	NaN	1.0	3.0	9.0	9.0	...	1.0	2.0	4.0	88.0	15.0	NaN	2.0	11.0	4.0	99
3	2.0	2.0	1.0	1.0	2819.0	2.0	2.0	3.0	2.0	1.0	...	1.0	1.0	5.0	30.0	30.0	1.0	2.0	9.0	4.0	8
4	2.0	1.0	2.0	1.0	2437.0	2.0	2.0	3.0	2.0	9.0	...	1.0	2.0	5.0	88.0	20.0	2.0	2.0	9.0	5.0	77

Setting the Stage: Cleaning Data



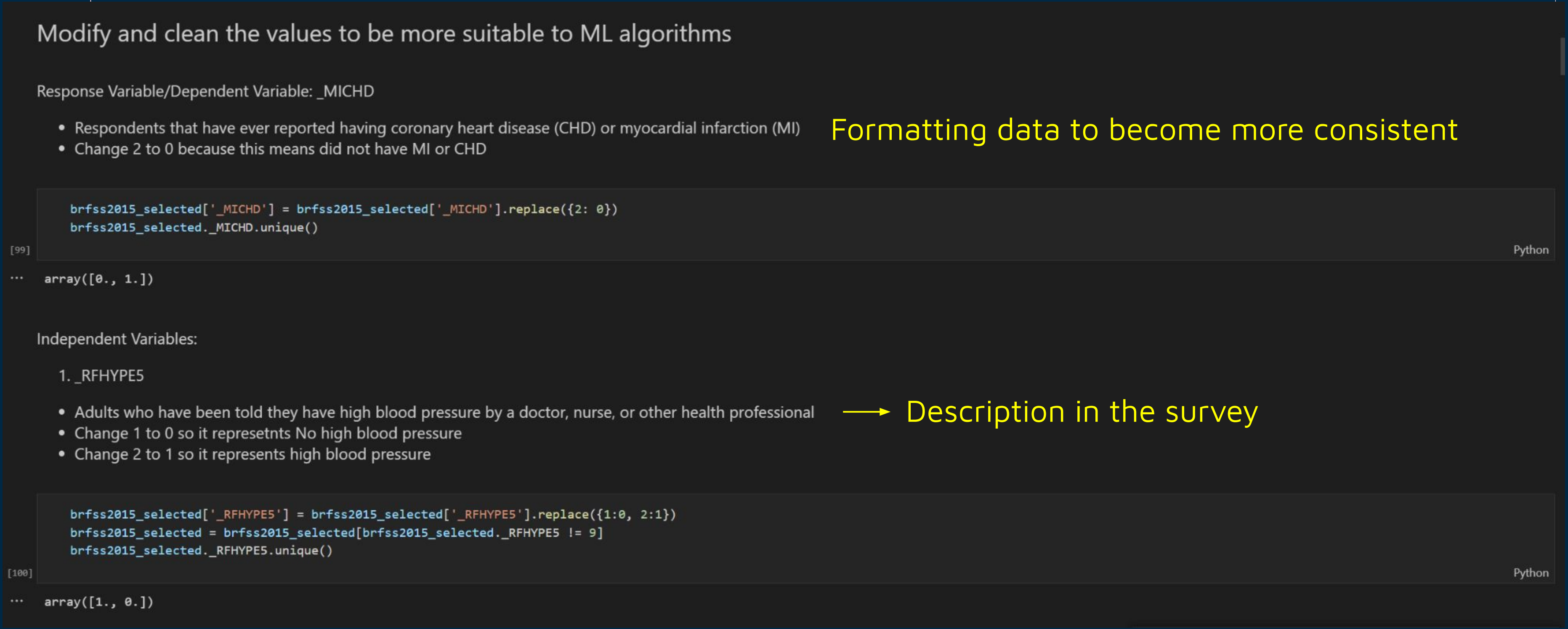
```
# Dropping missing values

brfss2015_selected = brfss2015_selected.dropna()
brfss2015_selected.shape
```

```
(343606, 22)
```

-97850

Setting the Stage: Cleaning Data



Modify and clean the values to be more suitable to ML algorithms

Response Variable/Dependent Variable: _MICHD

- Respondents that have ever reported having coronary heart disease (CHD) or myocardial infarction (MI)
- Change 2 to 0 because this means did not have MI or CHD

Formatting data to become more consistent

```
[99]
brfss2015_selected['_MICHD'] = brfss2015_selected['_MICHD'].replace({2: 0})
brfss2015_selected._MICHD.unique()
```

Python

```
array([0., 1.])
```

Independent Variables:

1. _RFHYPE5

- Adults who have been told they have high blood pressure by a doctor, nurse, or other health professional
- Change 1 to 0 so it represetnts No high blood pressure
- Change 2 to 1 so it represents high blood pressure

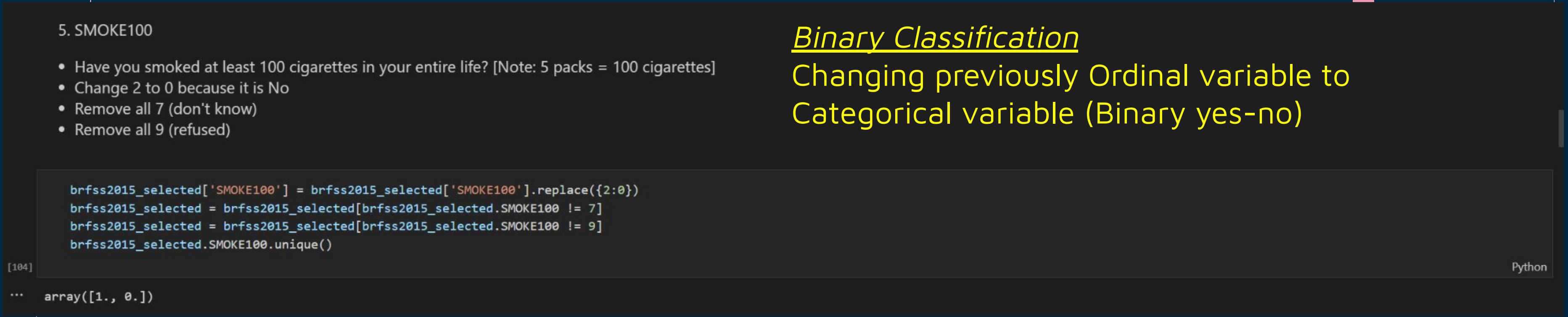
→ Description in the survey

```
[100]
brfss2015_selected['_RFHYPE5'] = brfss2015_selected['_RFHYPE5'].replace({1:0, 2:1})
brfss2015_selected = brfss2015_selected[brfss2015_selected._RFHYPE5 != 9]
brfss2015_selected._RFHYPE5.unique()
```

Python

```
array([1., 0.])
```

Setting the Stage: Cleaning Data



5. SMOKE100

- Have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes]
- Change 2 to 0 because it is No
- Remove all 7 (don't know)
- Remove all 9 (refused)

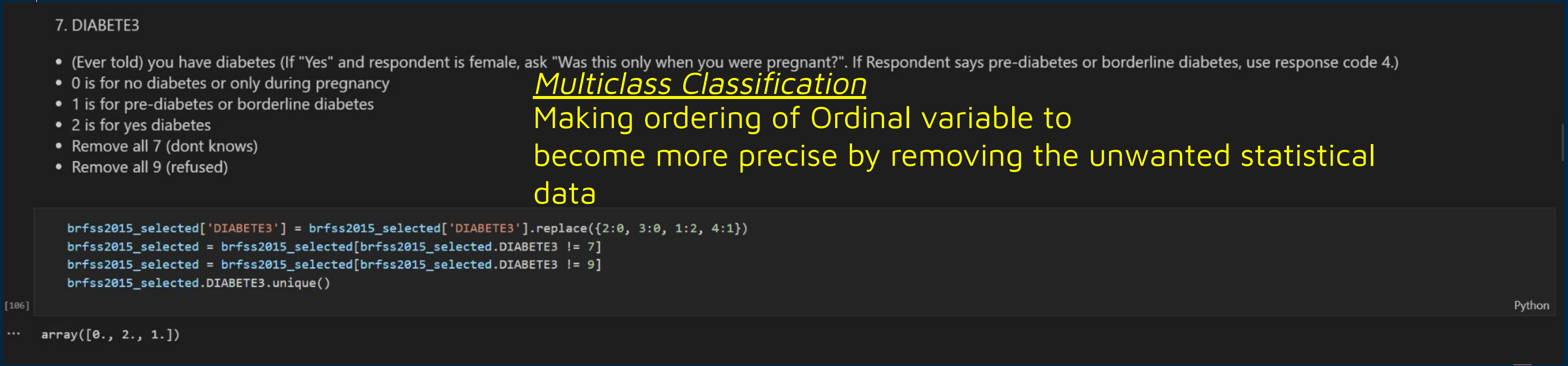
```
brfss2015_selected['SMOKE100'] = brfss2015_selected['SMOKE100'].replace({2:0})
brfss2015_selected = brfss2015_selected[brfss2015_selected.SMOKE100 != 7]
brfss2015_selected = brfss2015_selected[brfss2015_selected.SMOKE100 != 9]
brfss2015_selected.SMOKE100.unique()
```

[104] Python

```
array([1., 0.])
```

Binary Classification

Changing previously Ordinal variable to Categorical variable (Binary yes-no)



7. DIABETE3

- (Ever told) you have diabetes (If "Yes" and respondent is female, ask "Was this only when you were pregnant?". If Respondent says pre-diabetes or borderline diabetes, use response code 4.)
- 0 is for no diabetes or only during pregnancy
- 1 is for pre-diabetes or borderline diabetes
- 2 is for yes diabetes
- Remove all 7 (dont knows)
- Remove all 9 (refused)

```
brfss2015_selected['DIABETE3'] = brfss2015_selected['DIABETE3'].replace({2:0, 3:0, 1:2, 4:1})
brfss2015_selected = brfss2015_selected[brfss2015_selected.DIABETE3 != 7]
brfss2015_selected = brfss2015_selected[brfss2015_selected.DIABETE3 != 9]
brfss2015_selected.DIABETE3.unique()
```

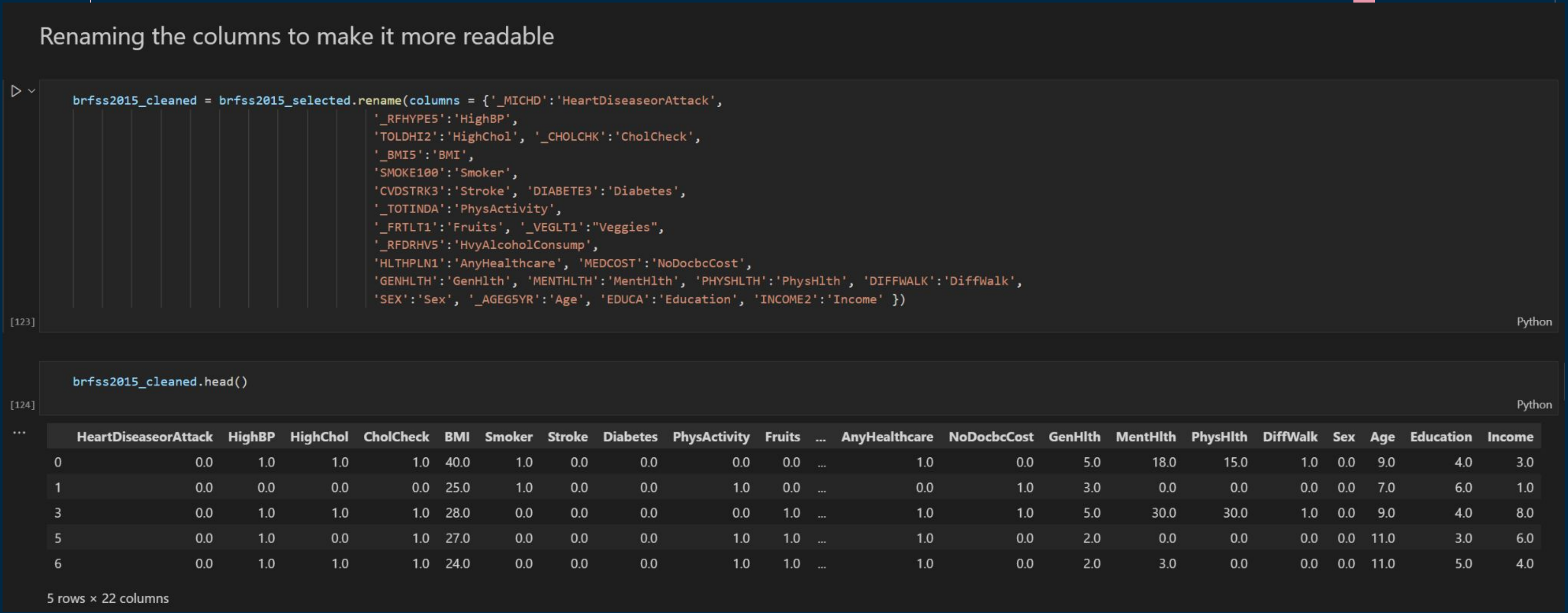
[106] Python

```
array([0., 2., 1.])
```

Multiclass Classification

Making ordering of Ordinal variable to become more precise by removing the unwanted statistical data

Setting the Stage: Cleaning Data



Renaming the columns to make it more readable

```python
brfss2015_cleaned = brfss2015_selected.rename(columns = {'_MICHD':'HeartDiseaseorAttack',
                                          '_RFHYPE5':'HighBP',
                                          'TOLDHI2':'HighChol', '_CHOLCHK':'CholCheck',
                                          '_BMI5':'BMI',
                                          'SMOKE100':'Smoker',
                                          'CVDSTRK3':'Stroke', 'DIABETE3':'Diabetes',
                                          '_TOTINDA':'PhysActivity',
                                          '_FRTLT1':'Fruits', '_VEGLT1':"Veggies",
                                          '_RFDRHV5':'HvyAlcoholConsump',
                                          'HLTHPLN1':'AnyHealthcare', 'MEDCOST':'NoDocbcCost',
                                          'GENHLTH':'GenHlth', 'MENTHLTH':'MentHlth', 'PHYSHLTH':'PhysHlth', 'DIFFWALK':'DiffWalk',
                                          'SEX':'Sex', '_AGEG5YR':'Age', 'EDUCA':'Education', 'INCOME2':'Income' })
```

[123] Python

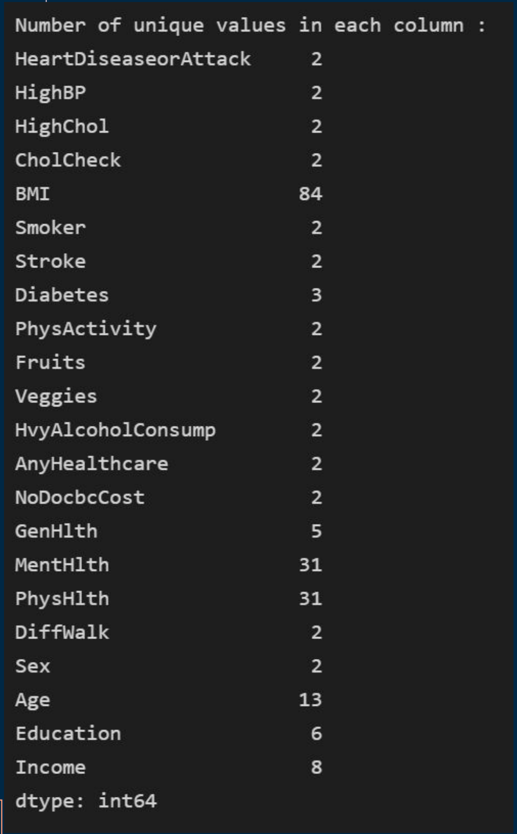
```python
brfss2015_cleaned.head()
```

[124] Python

	HeartDiseaseorAttack	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	Diabetes	PhysActivity	Fruits	...	AnyHealthcare	NoDocbcCost	GenHlth	MentHlth	PhysHlth	DiffWalk	Sex	Age	Education	Income
0	0.0	1.0	1.0	1.0	40.0	1.0	0.0	0.0	0.0	0.0	...	1.0	0.0	5.0	18.0	15.0	1.0	0.0	9.0	4.0	3.0
1	0.0	0.0	0.0	0.0	25.0	1.0	0.0	0.0	1.0	0.0	...	0.0	1.0	3.0	0.0	0.0	0.0	0.0	7.0	6.0	1.0
3	0.0	1.0	1.0	1.0	28.0	0.0	0.0	0.0	0.0	1.0	...	1.0	1.0	5.0	30.0	30.0	1.0	0.0	9.0	4.0	8.0
5	0.0	1.0	0.0	1.0	27.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	2.0	0.0	0.0	0.0	0.0	11.0	3.0	6.0
6	0.0	1.0	1.0	1.0	24.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	2.0	3.0	0.0	0.0	0.0	11.0	5.0	4.0

5 rows × 22 columns

Setting the Stage: What are the Variables?



```
Number of unique values in each column :
HeartDiseaseorAttack     2
HighBP                   2
HighChol                 2
CholCheck                2
BMI                     84
Smoker                   2
Stroke                   2
Diabetes                 3
PhysActivity             2
Fruits                   2
Veggies                  2
HvyAlcoholConsump        2
AnyHealthcare            2
NoDocbcCost              2
GenHlth                  5
MentHlth                31
PhysHlth                31
DiffWalk                 2
Sex                      2
Age                     13
Education                6
Income                   8
dtype: int64
```

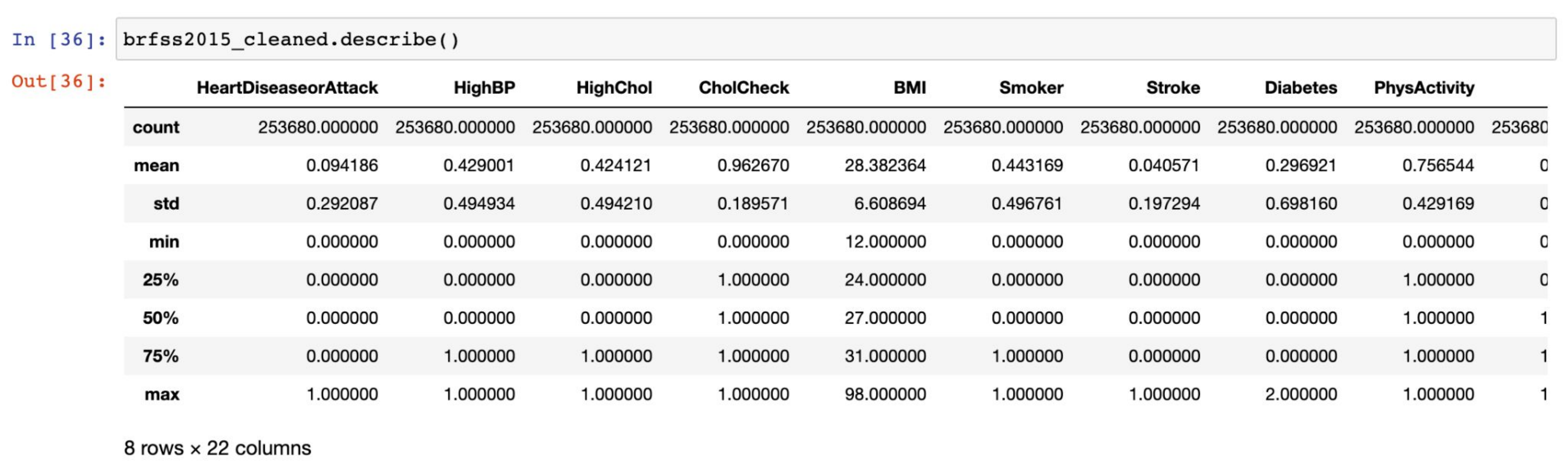
Independent Variable (1):

Heart disease/attack?

Dependent Variables (21):

Blood pressure, cholesterol, how recent is cholesterol check, BMI, smoke activity, stroke, diabetes, physical activity frequency, eat fruits, eat veggies, amount of alcohol consumption, registered healthcare insurance, financial problems for medical visits, general health, mental health, physical health, difficulty walking/climbing stairs, sex, age, education level, income level

Exploratory Data Analysis: Descriptive Statistics



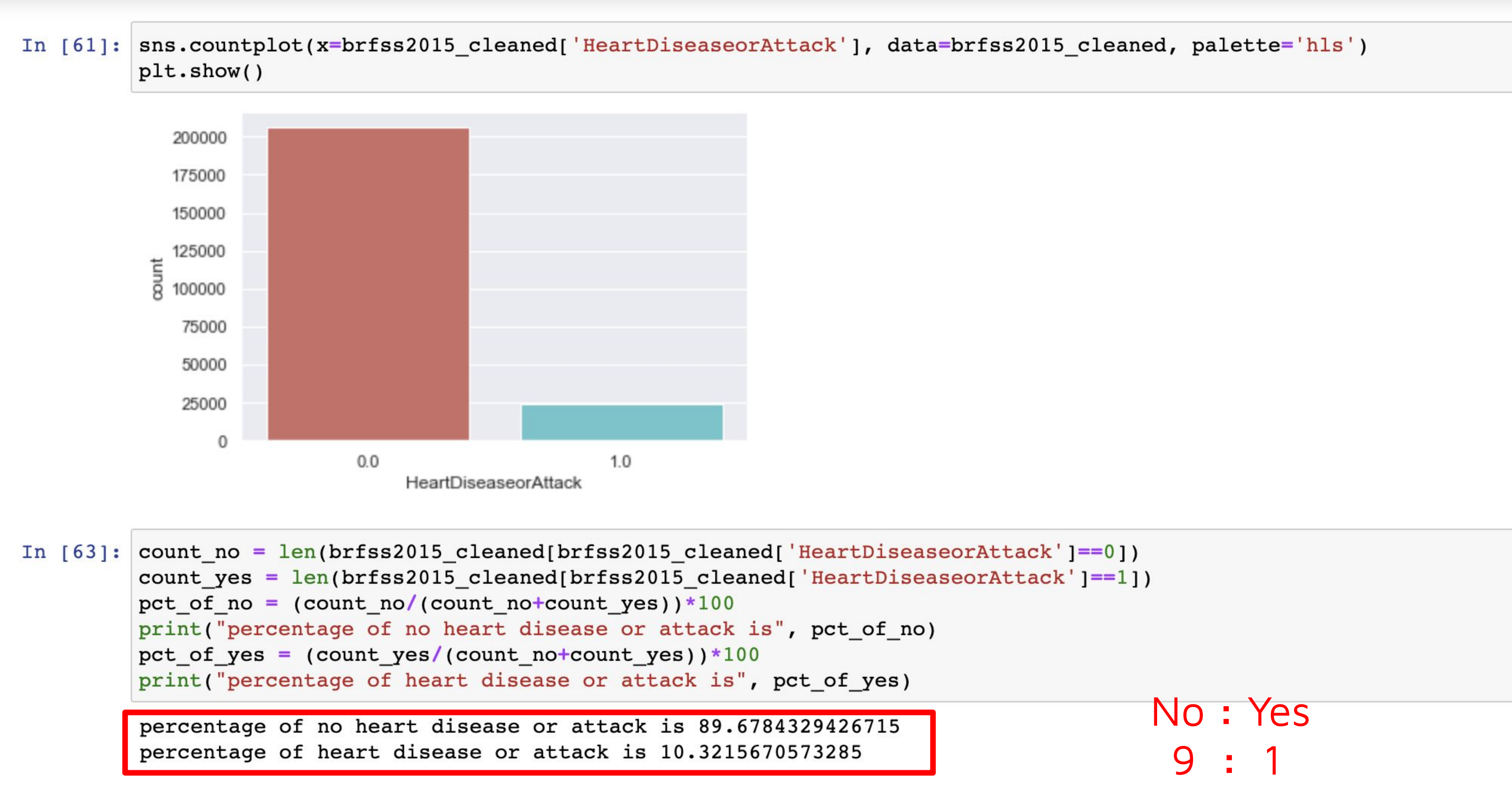
```
In [36]: brfss2015_cleaned.describe()
```

Out[36]:

	HeartDiseaseorAttack	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	Diabetes	PhysActivity	
count	253680.000000	253680.000000	253680.000000	253680.000000	253680.000000	253680.000000	253680.000000	253680.000000	253680.000000	253680
mean	0.094186	0.429001	0.424121	0.962670	28.382364	0.443169	0.040571	0.296921	0.756544	0
std	0.292087	0.494934	0.494210	0.189571	6.608694	0.496761	0.197294	0.698160	0.429169	0
min	0.000000	0.000000	0.000000	0.000000	12.000000	0.000000	0.000000	0.000000	0.000000	0
25%	0.000000	0.000000	0.000000	1.000000	24.000000	0.000000	0.000000	0.000000	1.000000	0
50%	0.000000	0.000000	0.000000	1.000000	27.000000	0.000000	0.000000	0.000000	1.000000	1
75%	0.000000	1.000000	1.000000	1.000000	31.000000	1.000000	0.000000	0.000000	1.000000	1
max	1.000000	1.000000	1.000000	1.000000	98.000000	1.000000	1.000000	2.000000	1.000000	1

8 rows × 22 columns

Exploratory Data Analysis



In [61]:

```
sns.countplot(x=brfss2015_cleaned['HeartDiseaseorAttack'], data=brfss2015_cleaned, palette='hls')
plt.show()
```

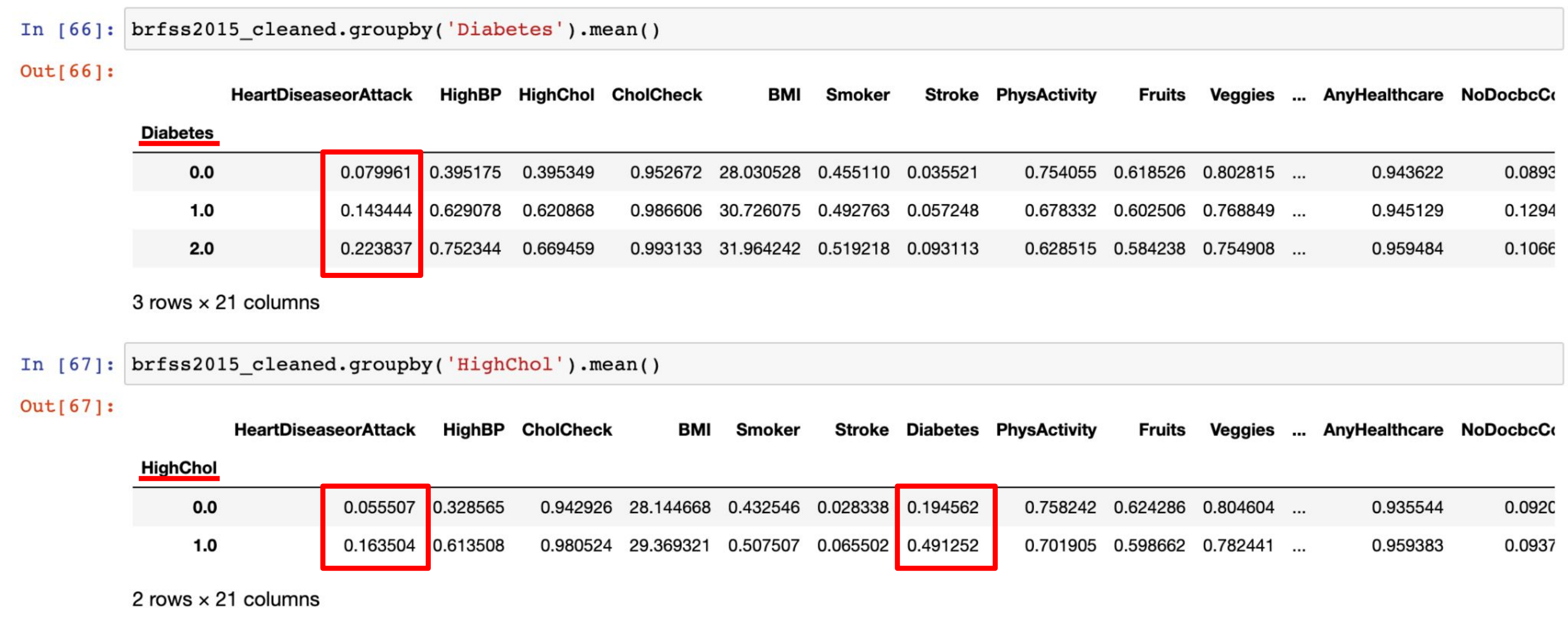
In [63]:

```
count_no = len(brfss2015_cleaned[brfss2015_cleaned['HeartDiseaseorAttack']==0])
count_yes = len(brfss2015_cleaned[brfss2015_cleaned['HeartDiseaseorAttack']==1])
pct_of_no = (count_no/(count_no+count_yes))*100
print("percentage of no heart disease or attack is", pct_of_no)
pct_of_yes = (count_yes/(count_no+count_yes))*100
print("percentage of heart disease or attack is", pct_of_yes)
```

```
percentage of no heart disease or attack is 89.6784329426715
percentage of heart disease or attack is 10.3215670573285
```

No : Yes
9 : 1

Exploratory Data Analysis



In [66]:
```
brfss2015_cleaned.groupby('Diabetes').mean()
```

Out[66]:

	HeartDiseaseorAttack	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	PhysActivity	Fruits	Veggies	...	AnyHealthcare	NoDocbcC
Diabetes													
0.0	0.079961	0.395175	0.395349	0.952672	28.030528	0.455110	0.035521	0.754055	0.618526	0.802815	...	0.943622	0.0893
1.0	0.143444	0.629078	0.620868	0.986606	30.726075	0.492763	0.057248	0.678332	0.602506	0.768849	...	0.945129	0.1294
2.0	0.223837	0.752344	0.669459	0.993133	31.964242	0.519218	0.093113	0.628515	0.584238	0.754908	...	0.959484	0.1066

3 rows × 21 columns

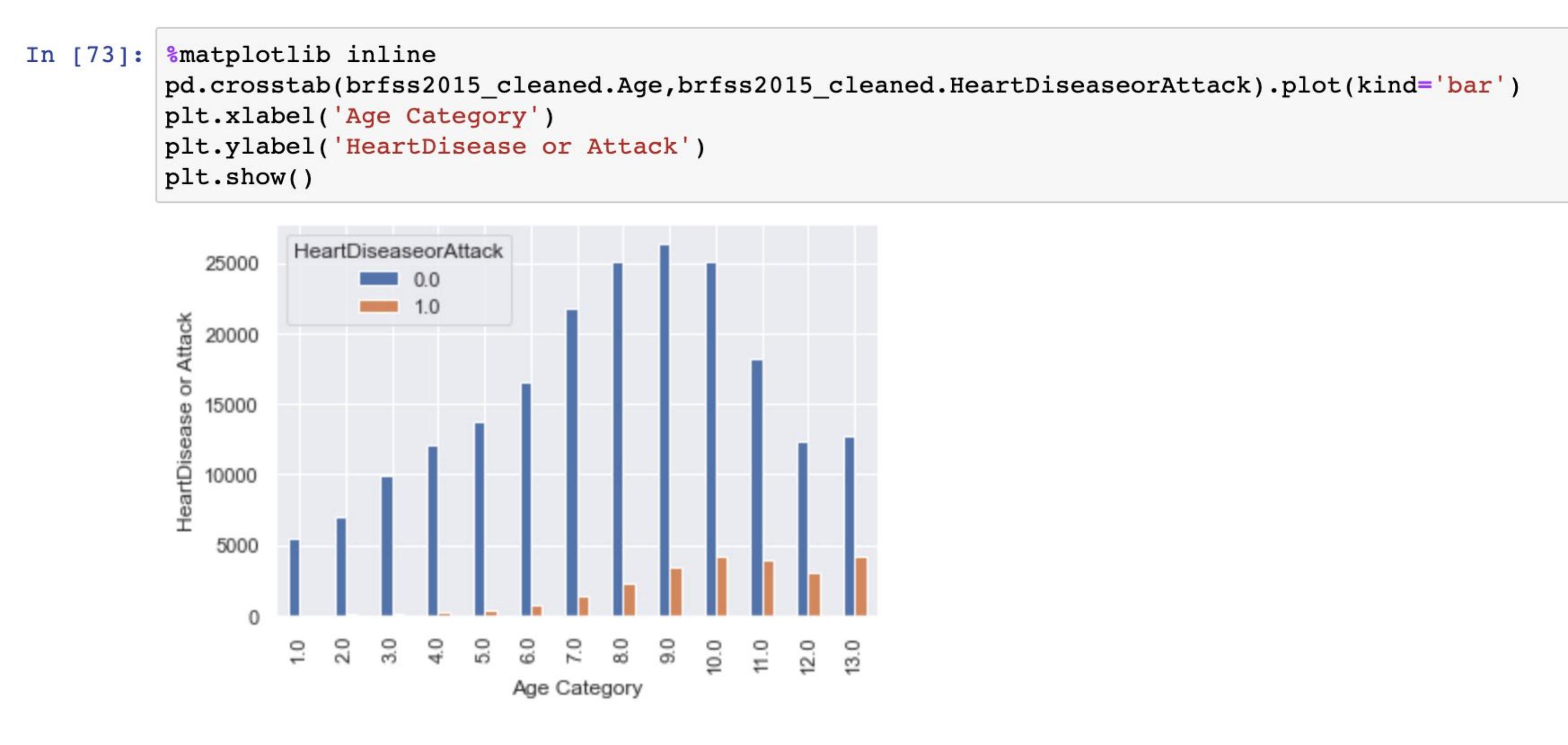
In [67]:
```
brfss2015_cleaned.groupby('HighChol').mean()
```

Out[67]:

	HeartDiseaseorAttack	HighBP	CholCheck	BMI	Smoker	Stroke	Diabetes	PhysActivity	Fruits	Veggies	...	AnyHealthcare	NoDocbcC
HighChol													
0.0	0.055507	0.328565	0.942926	28.144668	0.432546	0.028338	0.194562	0.758242	0.624286	0.804604	...	0.935544	0.0920
1.0	0.163504	0.613508	0.980524	29.369321	0.507507	0.065502	0.491252	0.701905	0.598662	0.782441	...	0.959383	0.0937

2 rows × 21 columns

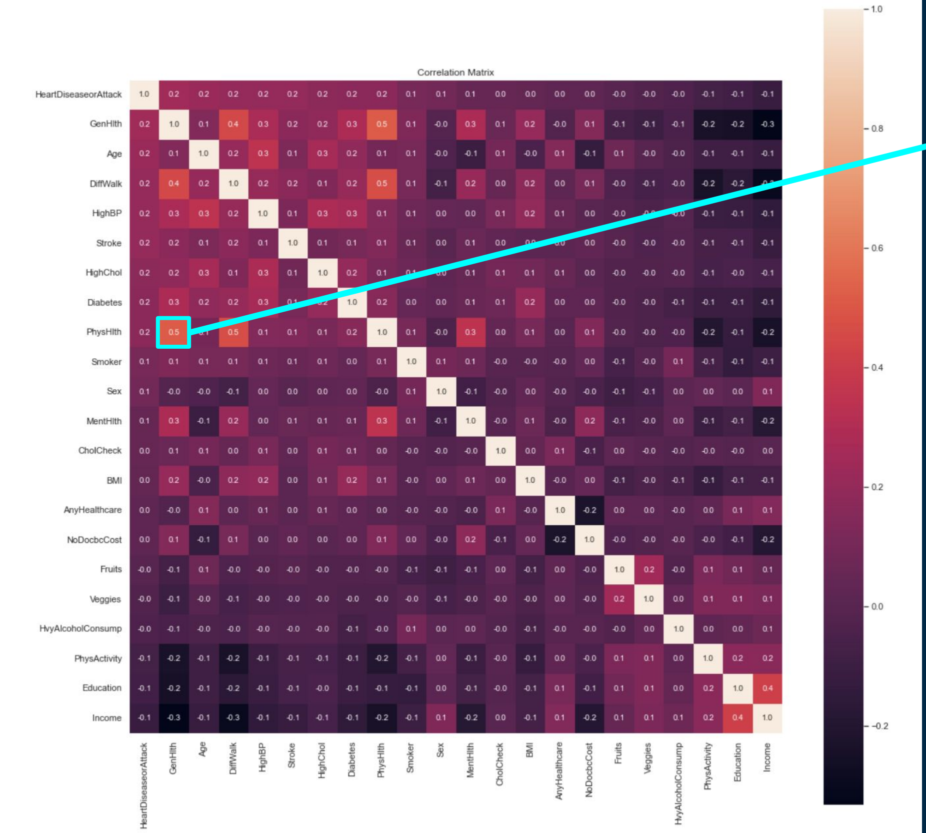
Data Exploration: Bar chart



In [73]:

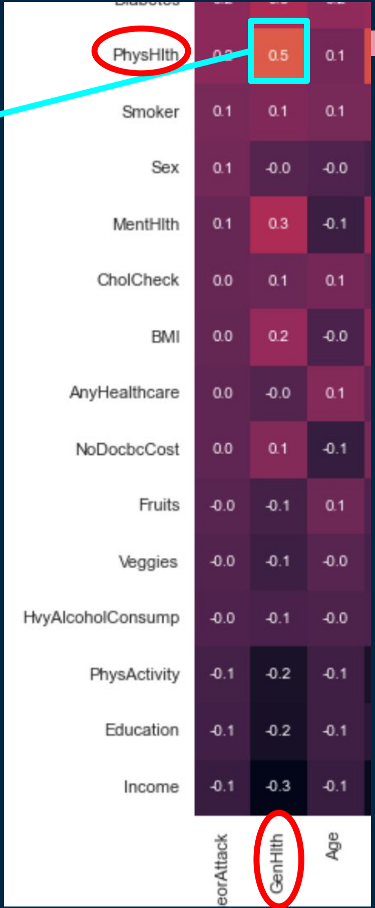
```
%matplotlib inline
pd.crosstab(brfss2015_cleaned.Age,brfss2015_cleaned.HeartDiseaseorAttack).plot(kind='bar')
plt.xlabel('Age Category')
plt.ylabel('HeartDisease or Attack')
plt.show()
```

Exploratory Data Analysis: Correlation Matrix



Correlation Matrix

	HeartDiseaseorAttack	GenHlth	Age	DiffWalk	HighBP	Stroke	HighChol	Diabetes	PhysHlth	Smoker	Sex	MentHlth	CholCheck	BMI	AnyHealthcare	NoDocbcCost	Fruits	Veggies	HvyAlcoholConsump	PhysActivity	Education	Income
HeartDiseaseorAttack	1.0	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.1	0.1	0.1	0.0	0.0	0.0	0.0	-0.0	-0.0	-0.0	-0.1	-0.1	-0.1
GenHlth	0.2	1.0	0.1	0.4	0.3	0.2	0.2	0.3	0.5	0.1	-0.0	0.3	0.1	0.2	-0.0	0.1	-0.1	-0.1	-0.1	-0.2	-0.2	-0.3
Age	0.2	0.1	1.0	0.2	0.3	0.1	0.3	0.2	0.1	0.1	-0.0	-0.1	0.1	-0.0	0.1	-0.1	0.1	-0.0	-0.0	-0.1	-0.1	-0.1
DiffWalk	0.2	0.4	0.2	1.0	0.2	0.2	0.1	0.2	0.5	0.1	-0.1	0.2	0.0	0.2	0.0	0.1	-0.0	-0.1	-0.0	-0.2	-0.2	-0.3
HighBP	0.2	0.3	0.3	0.2	1.0	0.1	0.3	0.3	0.1	0.1	0.0	0.0	0.1	0.2	0.1	0.0	-0.0	-0.0	-0.0	-0.1	-0.1	-0.1
Stroke	0.2	0.2	0.1	0.2	0.1	1.0	0.1	0.1	0.1	0.1	0.0	0.1	0.0	0.0	0.0	0.0	-0.0	-0.0	-0.0	-0.1	-0.1	-0.1
HighChol	0.2	0.2	0.3	0.1	0.3	0.1	1.0	0.2	0.1	0.1	0.0	0.1	0.1	0.1	0.1	0.0	-0.0	-0.0	-0.0	-0.1	-0.0	-0.1
Diabetes	0.2	0.3	0.2	0.2	0.3	0.1	0.2	1.0	0.2	0.0	0.0	0.1	0.1	0.2	0.0	0.0	-0.0	-0.0	-0.1	-0.1	-0.1	-0.1
PhysHlth	0.2	0.5	0.1	0.5	0.1	0.1	0.1	0.2	1.0	0.1	-0.0	0.3	0.0	0.1	0.0	0.1	-0.0	-0.0	-0.0	-0.2	-0.1	-0.2
Smoker	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.1	1.0	0.1	0.1	-0.0	-0.0	-0.0	0.0	-0.1	-0.0	0.1	-0.1	-0.1	-0.1
Sex	0.1	-0.0	-0.0	-0.1	0.0	0.0	0.0	0.0	-0.0	0.1	1.0	-0.1	-0.0	0.0	-0.0	-0.0	-0.1	-0.1	0.0	0.0	0.0	0.1
MentHlth	0.1	0.3	-0.1	0.2	0.0	0.1	0.1	0.1	0.3	0.1	-0.1	1.0	-0.0	0.1	-0.0	0.2	-0.1	-0.0	0.0	-0.1	-0.1	-0.2
CholCheck	0.0	0.1	0.1	0.0	0.1	0.0	0.1	0.1	0.0	-0.0	-0.0	-0.0	1.0	0.0	0.1	-0.1	0.0	-0.0	-0.0	-0.0	-0.0	0.0
BMI	0.0	0.2	-0.0	0.2	0.2	0.0	0.1	0.2	0.1	-0.0	0.0	0.1	0.0	1.0	-0.0	0.0	-0.1	-0.0	-0.1	-0.1	-0.1	-0.1
AnyHealthcare	0.0	-0.0	0.1	0.0	0.1	0.0	0.1	0.0	0.0	-0.0	-0.0	-0.0	0.1	-0.0	1.0	-0.2	0.0	0.0	-0.0	0.0	0.1	0.1
NoDocbcCost	0.0	0.1	-0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.0	-0.0	0.2	-0.1	0.0	-0.2	1.0	-0.0	-0.0	-0.0	-0.0	-0.1	-0.2
Fruits	-0.0	-0.1	0.1	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.1	-0.1	-0.1	0.0	-0.1	0.0	-0.0	1.0	0.2	-0.0	0.1	0.1	0.1
Veggies	-0.0	-0.1	-0.0	-0.1	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.1	-0.0	-0.0	-0.0	0.0	-0.0	0.2	1.0	0.0	0.1	0.1	0.1
HvyAlcoholConsump	-0.0	-0.1	-0.0	-0.0	-0.0	-0.0	-0.0	-0.1	-0.0	0.1	0.0	0.0	-0.0	-0.1	-0.0	-0.0	-0.0	0.0	1.0	0.0	0.0	0.1
PhysActivity	-0.1	-0.2	-0.1	-0.2	-0.1	-0.1	-0.1	-0.1	-0.2	-0.1	0.0	-0.1	-0.0	-0.1	0.0	-0.0	0.1	0.1	0.0	1.0	0.2	0.2
Education	-0.1	-0.2	-0.1	-0.2	-0.1	-0.1	-0.0	-0.1	-0.1	-0.1	0.0	-0.1	-0.0	-0.1	0.1	-0.1	0.1	0.1	0.0	0.2	1.0	0.4
Income	-0.1	-0.3	-0.1	-0.3	-0.1	-0.1	-0.1	-0.1	-0.2	-0.1	0.1	-0.2	0.0	-0.1	0.1	-0.2	0.1	0.1	0.1	0.2	0.4	1.0



Machine Learning: Logistic Regression

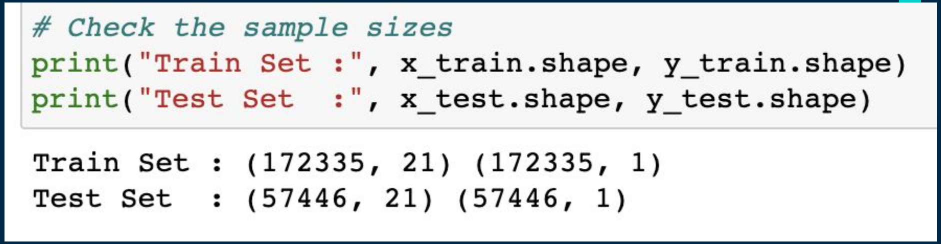
- Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable

- The dependent variable is a binary variable that contains data coded as 1 (yes) or 0 (no)

- Dataset Split

  Train Set - 75%

  Test Set - 25%

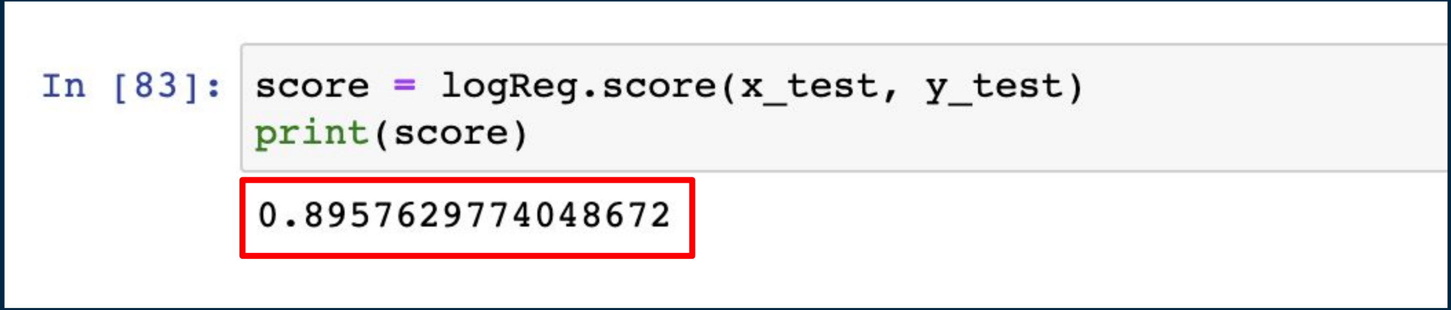


```
# Check the sample sizes
print("Train Set :", x_train.shape, y_train.shape)
print("Test Set  :", x_test.shape, y_test.shape)
```

```
Train Set : (172335, 21) (172335, 1)
Test Set  : (57446, 21) (57446, 1)
```

Model Performance

- Score: Accuracy of model on given test data



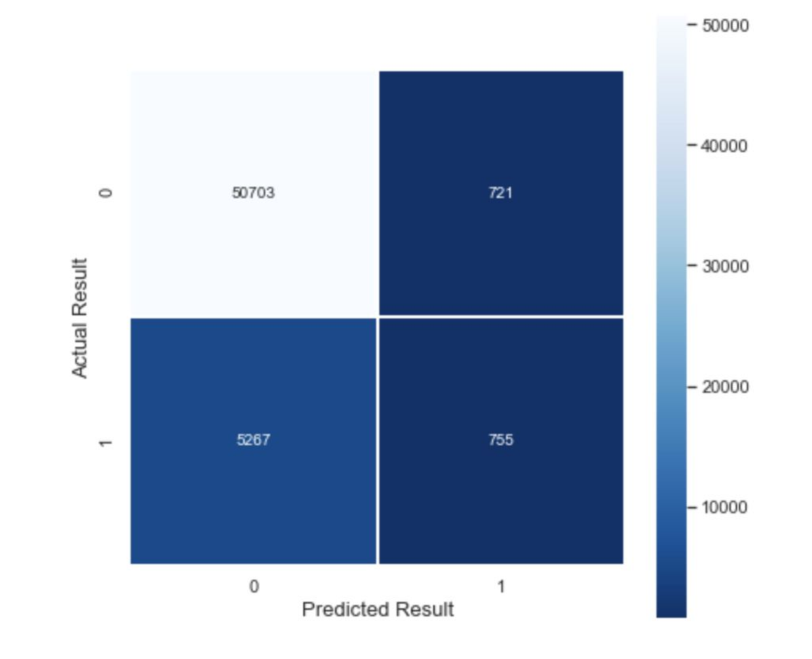
```
In [83]: score = logReg.score(x_test, y_test)
         print(score)

         0.8957629774048672
```

The score 0.8957629774048672 ≃ 0.9

Model Performance

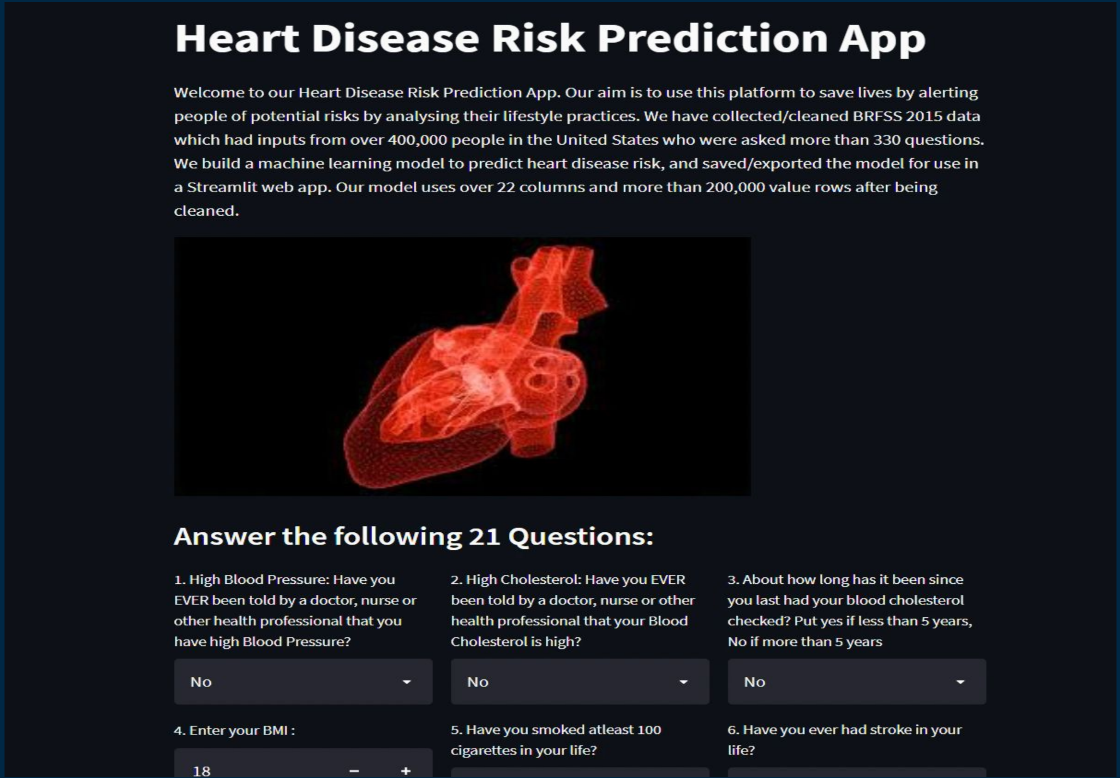
- Confusion Matrix: Tabular summary of the number of correct and incorrect predictions made by a classifier



- Misclassification Rate
  0.104

- False Positive Rate
  0.014

Final Outcome



Heart Disease Risk Prediction App

Welcome to our Heart Disease Risk Prediction App. Our aim is to use this platform to save lives by alerting people of potential risks by analysing their lifestyle practices. We have collected/cleaned BRFSS 2015 data which had inputs from over 400,000 people in the United States who were asked more than 330 questions. We build a machine learning model to predict heart disease risk, and saved/exported the model for use in a Streamlit web app. Our model uses over 22 columns and more than 200,000 value rows after being cleaned.

Answer the following 21 Questions:

1. High Blood Pressure: Have you EVER been told by a doctor, nurse or other health professional that you have high Blood Pressure?

No

2. High Cholesterol: Have you EVER been told by a doctor, nurse or other health professional that your Blood Cholesterol is high?

No

3. About how long has it been since you last had your blood cholesterol checked? Put yes if less than 5 years, No if more than 5 years

No

4. Enter your BMI :

18

5. Have you smoked atleast 100 cigarettes in your life?

6. Have you ever had stroke in your life?

What we learned (Something new)

- Use of Streamlit for deploying webapp and incorporating a beautiful interface for our ML model pickled using joblib
- Use of methods such as predict and predict_proba in logistic regression
- Exploring Sklearn modules such as Pipeline , Standard scaler for scaling and further exploration of seaborn as a tool for visualization

Usability (Application in real life)

Using our ML Function through our Streamlit interface, users can:

- Check for risk of heart diseases based on lifestyle practices
- Look for recommendations to improve their overall health
- Understand the major factors affecting their health and consult their doctors before it is too late.

Our Data Driven Insights

- Increase risk of Heart Diseases if:
  - High Cholesterol levels
  - High Blood Pressure
  - Smoked more than 100 cigarettes in your life
  - Older age
  - Worsening Physical Health and Mental Health
  - No Physical Exercise

Our Data Driven Insights

- Decreases risk of Heart Diseases if:

  - More money and education
  - If you are a female

Factors such as number of fruits and vegetables eaten in a day, health insurance , and medical cost did not significantly affect the risk of heart diseases as compared to independent variables mentioned above .

CONCLUSION

Future Vision For Our Project

- Using Datasets from all around the world and not only the United States to access heart disease risk worldwide
- Include more relevant factors that affect heart health
- Use different machine learning models to check for better accuracy
- Deploy the Streamlit application online
- Expand the scope of project by also predicting other health problems such as diabetes , cancer

THANK YOU