In [23]:
```
import pandas as pd
import numpy as np
import imblearn
from imblearn.under_sampling import NearMiss
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LassoCV
```

Creating The Data Set

In [5]:

```
df = pd.read_csv('diabetes.csv')
undersample = NearMiss(version=1)
X = df.loc[:, df.columns != 'Diabetes_binary']
y = df.loc[:, df.columns == 'Diabetes_binary']
X, y = undersample.fit_resample(X, y)

print(X.info())
print(y.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70692 entries, 0 to 70691
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   HighBP                70692 non-null  float64
 1   HighChol              70692 non-null  float64
 2   CholCheck             70692 non-null  float64
 3   BMI                   70692 non-null  float64
 4   Smoker                70692 non-null  float64
 5   Stroke                70692 non-null  float64
 6   HeartDiseaseorAttack  70692 non-null  float64
 7   PhysActivity          70692 non-null  float64
 8   Fruits                70692 non-null  float64
 9   Veggies               70692 non-null  float64
 10  HvyAlcoholConsump     70692 non-null  float64
 11  AnyHealthcare         70692 non-null  float64
 12  NoDocbcCost           70692 non-null  float64
 13  GenHlth               70692 non-null  float64
 14  MentHlth              70692 non-null  float64
 15  PhysHlth              70692 non-null  float64
 16  DiffWalk              70692 non-null  float64
 17  Sex                   70692 non-null  float64
 18  Age                   70692 non-null  float64
 19  Education             70692 non-null  float64
 20  Income                70692 non-null  float64
dtypes: float64(21)
memory usage: 11.3 MB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70692 entries, 0 to 70691
Data columns (total 1 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Diabetes_binary  70692 non-null  float64
dtypes: float64(1)
memory usage: 552.4 KB
None
```

Splitting The Dataset

In [16]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, ran
scaler = StandardScaler()
scaler.fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)

df_undersampled_train = pd.DataFrame(X_train_scaled, columns = X.columns)
df_undersampled_train['Diabetes_binary'] = y_train
df_undersampled_train.head()

df_undersampled_test = pd.DataFrame(X_test_scaled, columns = X.columns)
df_undersampled_test['Diabetes_binary'] = y_test
df_undersampled_test.head()
```

Out[16]:

	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysA
0	-1.212894	0.876922	0.074482	-1.061978	1.158253	-0.225623	-0.384172	0.5
1	-1.212894	-1.140353	0.074482	0.377975	1.158253	-0.225623	-0.384172	0.5
2	0.824475	0.876922	0.074482	1.017954	-0.863369	-0.225623	-0.384172	0.5
3	-1.212894	0.876922	0.074482	0.377975	-0.863369	-0.225623	-0.384172	0.5
4	0.824475	-1.140353	0.074482	2.777896	-0.863369	-0.225623	-0.384172	0.5

5 rows × 22 columns

Looking At The Results That LassoCV Yields

In [25]:

```
lasso = LassoCV(cv=5, random_state=0).fit(X_train_scaled, y_train)

coef = lasso.coef_
col = X.columns
for index in range(len(coef)):
    if coef[index] > 0.015:
        print(f'{col[index]}: {np.round(coef[index], 3)}')
```

```
C:\Users\Felipe\anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_
descent.py:1571: DataConversionWarning: A column-vector y was passed when a 1
d array was expected. Please change the shape of y to (n_samples, ), for exam
ple using ravel().
  y = column_or_1d(y, warn=True)

HighBP: 0.031
BMI: 0.065
Smoker: 0.016
HeartDiseaseorAttack: 0.029
HvyAlcoholConsump: 0.017
GenHlth: 0.11
MentHlth: 0.016
PhysHlth: 0.024
DiffWalk: 0.03
```

Looking At The Results That Correlation Yields

In [18]:

```
corr = df_undersampled_train.corr()
corr
```

Out[18]:

	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDise
HighBP	1.000000	0.290281	0.019190	0.252404	0.132512	0.119859	
HighChol	0.290281	1.000000	0.012837	0.129130	0.125240	0.081647	
CholCheck	0.019190	0.012837	1.000000	-0.004101	-0.005190	-0.000986	
BMI	0.252404	0.129130	-0.004101	1.000000	0.063508	0.057133	
Smoker	0.132512	0.125240	-0.005190	0.063508	1.000000	0.072942	
Stroke	0.119859	0.081647	-0.000986	0.057133	0.072942	1.000000	
HeartDiseaseorAttack	0.192153	0.160192	-0.003985	0.098065	0.144389	0.233298	
PhysActivity	-0.185341	-0.123335	0.019316	-0.255187	-0.102905	-0.126957	
Fruits	-0.103985	-0.089436	0.015673	-0.158303	-0.102695	-0.046131	
Veggies	-0.123597	-0.084961	0.010999	-0.123689	-0.060063	-0.084637	
HvyAlcoholConsump	0.022629	0.028740	-0.005638	0.000244	0.066169	-0.008613	
AnyHealthcare	-0.032929	-0.023433	0.079857	-0.064963	-0.028095	-0.028131	
NoDocbcCost	0.077367	0.065678	-0.059807	0.129655	0.042161	0.077127	
GenHlth	0.322079	0.223396	-0.020087	0.345232	0.181521	0.223366	
MentHlth	0.124688	0.117957	-0.027875	0.201497	0.108281	0.141691	
PhysHlth	0.188203	0.139995	-0.014818	0.255436	0.140699	0.210303	
DiffWalk	0.225602	0.147227	-0.009001	0.316306	0.140045	0.236295	
Sex	-0.011667	-0.007690	-0.010735	-0.044020	0.115278	-0.019964	
Age	0.275838	0.163192	0.022321	-0.112717	0.145365	0.083517	
Education	-0.227568	-0.134820	0.006402	-0.201685	-0.171960	-0.120345	
Income	-0.282530	-0.163382	0.031126	-0.242094	-0.152527	-0.194099	
Diabetes_binary	-0.012546	-0.011030	-0.002274	-0.004097	-0.006953	-0.002408	

22 rows × 22 columns

These Are The Correlation Results

In [19]:

```
corr_target = abs(corr["Diabetes_binary"])
relevant_features = corr_target[corr_target>0.006]
relevant_features
```

Out[19]:

```
HighBP             0.012546
HighChol           0.011030
Smoker             0.006953
Fruits             0.006887
AnyHealthcare      0.007229
MentHlth           0.011795
DiffWalk           0.012193
Sex                0.012702
Income             0.008735
Diabetes_binary    1.000000
Name: Diabetes_binary, dtype: float64
```

Creating A Variable Containing The Training and Testing Splits Of The Correlation Variables

- The Features that were selected for correlation:
    - Sex
    - HighBP
    - DiffWalk
    - MentHlth
    - HighChol
    - AnyHealthCare
    - Smoker
    - Fruits
    - Income
- Below is going to be the creation and presentation of the dataframe to see its details

In [20]:

```
X_selected_train = df_undersampled_train.loc[:, ['Sex', 'HighBP', 'DiffWalk',
                       'Fruits','Income']]
print(X_selected_train.info())

X_selected_test = df_undersampled_test.loc[:, ['Sex', 'HighBP', 'DiffWalk', 'Me
                       'Fruits','Income']]
print(X_selected_test.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49484 entries, 0 to 49483
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Sex            49484 non-null  float64
 1   HighBP         49484 non-null  float64
 2   DiffWalk       49484 non-null  float64
 3   MentHlth       49484 non-null  float64
 4   HighChol       49484 non-null  float64
 5   AnyHealthcare  49484 non-null  float64
 6   Smoker         49484 non-null  float64
 7   Fruits         49484 non-null  float64
 8   Income         49484 non-null  float64
dtypes: float64(9)
memory usage: 3.4 MB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21208 entries, 0 to 21207
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Sex            21208 non-null  float64
 1   HighBP         21208 non-null  float64
 2   DiffWalk       21208 non-null  float64
 3   MentHlth       21208 non-null  float64
 4   HighChol       21208 non-null  float64
 5   AnyHealthcare  21208 non-null  float64
 6   Smoker         21208 non-null  float64
 7   Fruits         21208 non-null  float64
 8   Income         21208 non-null  float64
dtypes: float64(9)
memory usage: 1.5 MB
None
```

Creating A Variable Containing The Training and Testing Splits Of The Lasso Variables

- The Featurs that were selected for Lasso:
    - HighBP
    - BMI
    - Smoker
    - HeartDiseaseorAttack
    - HvyAlcoholConsump
    - GenHlth

    - MentHlth
    - PhysHlth
    - DiffWalk

- Below is the creating of the datadrame containing the features and presentation of details concerning them

In [29]:
```
X_selected_train = df_undersampled_train.loc[:, ['HighBP', 'BMI', 'Smoker','Hea
                                                 'GenHlth','MentHlth', 'PhysHlt
print(X_selected_train.info())

X_selected_test = df_undersampled_test.loc[:, ['HighBP', 'BMI', 'Smoker','Heart
                                               'GenHlth','MentHlth', 'PhysHlth
print(X_selected_test.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49484 entries, 0 to 49483
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   HighBP                49484 non-null  float64
 1   BMI                   49484 non-null  float64
 2   Smoker                49484 non-null  float64
 3   HeartDiseaseorAttack  49484 non-null  float64
 4   HvyAlcoholConsump     49484 non-null  float64
 5   GenHlth               49484 non-null  float64
 6   MentHlth              49484 non-null  float64
 7   PhysHlth              49484 non-null  float64
 8   DiffWalk              49484 non-null  float64
dtypes: float64(9)
memory usage: 3.4 MB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21208 entries, 0 to 21207
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   HighBP                21208 non-null  float64
 1   BMI                   21208 non-null  float64
 2   Smoker                21208 non-null  float64
 3   HeartDiseaseorAttack  21208 non-null  float64
 4   HvyAlcoholConsump     21208 non-null  float64
 5   GenHlth               21208 non-null  float64
 6   MentHlth              21208 non-null  float64
 7   PhysHlth              21208 non-null  float64
 8   DiffWalk              21208 non-null  float64
dtypes: float64(9)
memory usage: 1.5 MB
None
```

Both Methods Are Going To Be Tested On Models From Step 1 And The Method With The Best Results Will Be Choosen

- In the future tests you will see that Lasso performs the best having a significantly greater accuracy than the correlation set