Logistic Regression Modeling for Diabetes Prediction

Shabnam Ahmadi

Master of Science in Data Analytics, Western Governors University

D214: Capstone

Dr. Daniel Smith

December 16, 2023

Research Question

Diabetes, a chronic condition that affects the body's ability to regulate blood sugar, has reached an alarming rate of diagnosed individuals globally. In 2021, it was reported that there were 536.6 million people worldwide who have been diagnosed with diabetes, this number is projected to increase to 783.7 million by 2045. Additionally, there were 6.7 million deaths in 2021 attributable to diabetes (International Diabetes Federation, 2021.) Diabetes continues to be an adverse condition that impacts the livelihood of millions. This condition not only impacts the quality of life of those diagnosed, but it is also one of the costliest preventable diseases in the United States. The American Diabetes Association (ADA) reports that the total annual cost of diabetes in 2022 was $412.9 billion. Additionally, the ADA states that diabetes accounts for reduced work productivity and missed workdays that cost $35.8 billion and $5.4 billion, respectively, in indirect costs annually (ADA, 2023.) There is not only a health incentive to maintain preventative diabetes care, but also a financial incentive for organizations across the United States.

In the United States, more than one-third of adults are at risk of diabetes, or pre-diabetic. Diabetes continues to be a threat to the average American's health (Johns Hopkins Medicine, 2023.*)* As such, many organizations have dedicated prevention programs and surveys to monitor and treat those who are at risk. This analysis will be utilizing an annual survey that is conducted by the Center for Disease Control and Prevention (CDC) called the Behavior Risk Factor Surveillance System survey (BRFSS). The BRFSS is a behavioral health survey conducted via telephone that collects data from all 50 states regarding health behaviors and current diagnoses. The research question for this analysis is: What key components, as determined by logistic regression, are most significant in predicting the likelihood of diabetes in a given population using the BRFSS survey data? The null and alternate hypotheses for this analysis are as follows:

Null Hypothesis (H_0) - There is no significant association between diabetes diagnosis (‘Diabetes_binary’) and reported high blood pressure ('HighBP'), reported high cholesterol ('HighChol'), recent cholesterol checks ('CholCheck'), BMI '(BMI_Category',) reported history with strokes ('Stroke'), reported history with heart disease or heat attacks ('HeartDiseaseorAttack'), reported alcohol consumption ('HvyAlcoholConsump'), reported health rating ('GenHlth'), reported difficulty walking ('DiffWalk'), reported sex ('Sex'), and reported age ('Age') variables.

Alternate Hypothesis (H_A) - There is a significant association between diabetes diagnosis (‘Diabetes_binary’) reported high blood pressure ('HighBP'), reported high cholesterol ('HighChol'), reported cholesterol checks ('CholCheck'), BMI '(BMI_Category',) reported history with strokes ('Stroke'), reported history with heart disease or heat attacks ('HeartDiseaseorAttack'), reported alcohol consumption ('HvyAlcoholConsump'), reported health rating ('GenHlth'), reported difficulty walking ('DiffWalk'), reported sex ('Sex'), and reported age ('Age') variables.

Data Collection

The purpose of collecting data is to explore the broader theme of the research question. The purpose of the data analysis is to understand the main components in a survey that can indicate whether an individual is predicted to have diabetes or not. The data being used in this data analysis report is from the CDC’s annual BRFSS survey. The data gathering methodology used in the BRFSS data is a telephone survey. The benefits of a telephone survey are that they are accessible, as most individuals in the United States have access to a personal cellular device, surveys are also easy to conduct and cost-effective (Sincero, 2023.) Another added benefit of using the CDC BRFSS survey data is that the CDC is a reputable and comprehensive source for health-related data. This ensures the integrity of the survey data. One disadvantage of using survey data is that there is a tendency for participants to show a response bias. The risk of response bias was a challenge when searching for potential surveys to use for this analysis (Sincero, 2023.) To alleviate the risk of response bias, we ensured to use of a survey that has objective questions, which leaves little room for the participant to answer the question based on their interpretation or opinion.

The data used in this analysis has 22 variables. The summary of the variables and data are as follows:

Variable Name	**Description**	**Values**
Diabetes_binary	A binary variable that indicates whether the individual has diabetes	0 = no diabetes, 1, 2 = diabetes or prediabetic
HighBP	A binary variable that indicates whether the individual has high blood pressure	0 = no high blood pressure 1 = high blood pressure
HighChol	A binary variable that indicates whether the individual has high cholesterol	0 = no high cholesterol 1 = high cholesterol
CholCheck	A binary variable that indicates whether the individual has checked their cholesterol in the past 5 years	0 = no cholesterol check in 5 years 1 = yes cholesterol check in 5 years
BMI	A quantitative measure of the surveyed individual’s Body Mass Index	Minimum value of 12 and maximum value of 98

Smoker	A binary variable that indicates whether the individual has smoked 100 or more cigarettes in their entire life	0 = no, the individual has not smoked more than 100 cigarettes in their life, 1 = yes, the individual has smoked more than 100 cigarettes in their life
Stroke	A binary variable that indicates whether the individual has had a stroke in their lifetime	0 = no 1 = yes
HeartDiseaseorAttack	A binary variable that indicates whether the individual has had coronary heart disease (CHD) or myocardial infarction in their lifetime	0 = no 1 = yes
PhysActivity	A binary variable that indicates whether the individual has participated in physical activity in the past 30 days (not including their job)	0 = no 1 = yes
Fruits	A binary variable that indicates whether the individual consumes a serving of fruit once or more times per day	0 = no 1 = yes
Veggies	A binary variable that indicates whether the individual consumes a serving of vegetables once or more times per day	0 = no 1 = yes
HvyAlcoholConsump	A binary variable that indicates whether the individual has consumed a heavy amount of alcohol in the last 7 days (adult men >=14 drinks per week and adult women>=7 drinks per week)	0 = no 1 = yes
AnyHealthcare	A binary variable that indicates whether the individual has any kind of health care coverage, including health insurance, prepaid plans such as HMO, etc	0 = no 1 = yes

NoDocbcCost	A binary variable that indicates whether the individual has had a time in the past 12 months when you needed to see a doctor but could not because of cost	0 = no 1 = yes
GenHlth	A variable that indicates an individual’s general health on a scale from 1 to 5, 1 indicating excellent health and 5 indicating poor health	1 = excellent, 2 = good, 3 = ok, 4 = bad, 5 = poor
MentHlth	A variable that indicates the number of poor mental heath days an individual has had in the past 30 days	Integers between 1 – 30 days
PhysHlth	A variable that indicates the number of poor physical heath days an individual has had in the past 30 days (includes illness and injury)	Integers between 1 – 30 days
DiffWalk	A binary variable that indicates whether the individual has difficulty walking or climbing stairs	0 = no 1 = yes
Sex	Sex of the individual	0 = female 1 = male
Age	A 13-bin age category	1 = 18-24 years old 2 = 25-29 years old 3 = 30-34 years old 4 = 35-39 years old 5 = 40-44 years old 6 = 45-49 years old 7 = 50-54 years old 8 = 55-59 years old 9 = 60-64 years old 10 = 65-69 years old

		11 = 70-74 years old 12 = 75-79 years old 13 = 80 years or older
Education	A 6-scale education level	1 = Never attended school or only kindergarten 2 = Grades 1 through 8 (Elementary) 3 = Grades 9 through 11 (Some high school) 4 = Grade 12 or GED (High school graduate) 5 = College 1 year to 3 years (Some college or technical school) 6 = College 4 years or more (College graduate)
Income	An 8-scale income level	1 = Less than $10,000 2 = Less than $15,000 ($10,000 to less than $15,000) 3 = Less than $20,000 ($15,000 to less than $20,000) 4 = Less than $25,000 ($20,000 to less than $25,000) 5 = Less than $35,000 ($25,000 to less than $35,000) 6 = Less than $50,000 ($35,000 to less than $50,000) 7 = Less than $75,000 ($50,000 to less than $75,000) 8 = $75,000 or more

The target variable for this analysis is “Diabetes_binary”. Since the target variable is binary, logistic regression will be used to answer the research question.

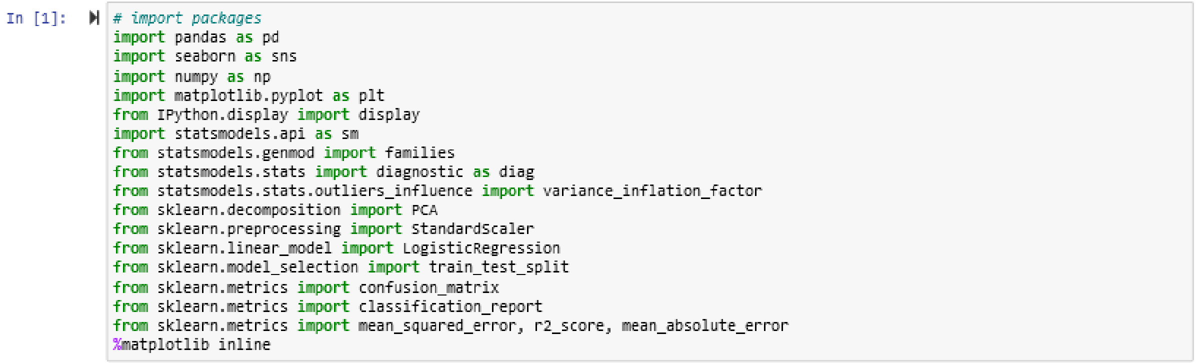
Data Extraction and Preparation

The data extraction process included identifying the data source to be the BRFSS survey data found on Kaggle. The dataset was downloaded as a csv file and uploaded to Jupyter Notebook using Panda's read_csv function. Jupyter Notebook was used for this analysis because we are using Python programming language. Additionally, Jupyter Notebook has an intuitive, easy-to-use interface that is optimal for data analysis. The read_csv function was used since our dataset comes in a csv format. One advantage of using Jupyter Notebook is that the application allows data visualizations through a clear-cut interface, which is beneficial for our analysis. One disadvantage is that there are performance issues when using the application for more demanding computational tasks. To mitigate this, we ensured to use tasks that do not demand a sizeable amount of computation.

This data analysis project will use logistic regression to answer the research question. As such, the data cleaning process will consist of general data cleaning and data cleaning to ensure the requirements for logistic regression are met.

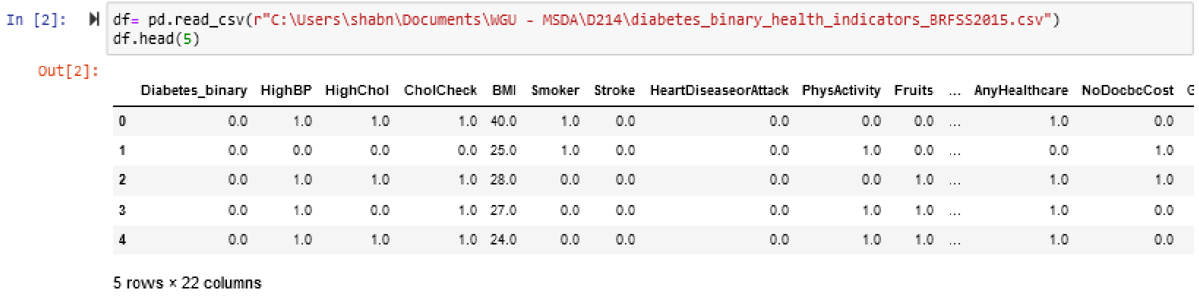
For our general data cleaning process, the following was conducted:

1. Import the packages needed for data cleaning and data analysis



```
In [1]:  # import packages
         import pandas as pd
         import seaborn as sns
         import numpy as np
         import matplotlib.pyplot as plt
         from IPython.display import display
         import statsmodels.api as sm
         from statsmodels.genmod import families
         from statsmodels.stats import diagnostic as diag
         from statsmodels.stats.outliers_influence import variance_inflation_factor
         from sklearn.decomposition import PCA
         from sklearn.preprocessing import StandardScaler
         from sklearn.linear_model import LogisticRegression
         from sklearn.model_selection import train_test_split
         from sklearn.metrics import confusion_matrix
         from sklearn.metrics import classification_report
         from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
         %matplotlib inline
```

2. Load data into Jupyter Notebook using read_csv function from Pandas

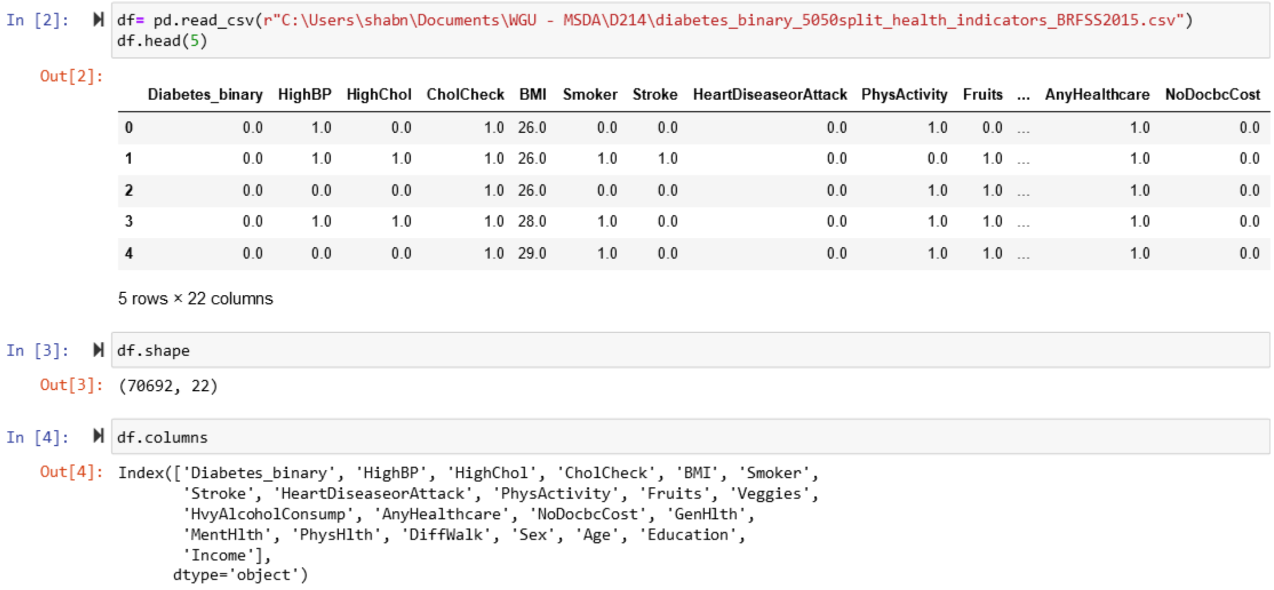


```
In [2]:  df= pd.read_csv(r"C:\Users\shabn\Documents\WGU - MSDA\D214\diabetes_binary_health_indicators_BRFSS2015.csv")
         df.head(5)

Out[2]:
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruits	...	AnyHealthcare	NoDocbcCost	G
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	
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	
2	0.0	1.0	1.0	1.0	28.0	0.0	0.0	0.0	0.0	1.0	...	1.0	1.0	
3	0.0	1.0	0.0	1.0	27.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	
4	0.0	1.0	1.0	1.0	24.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	

5 rows × 22 columns

3. Use .head(), .shape, .columns to get familiar with the data.



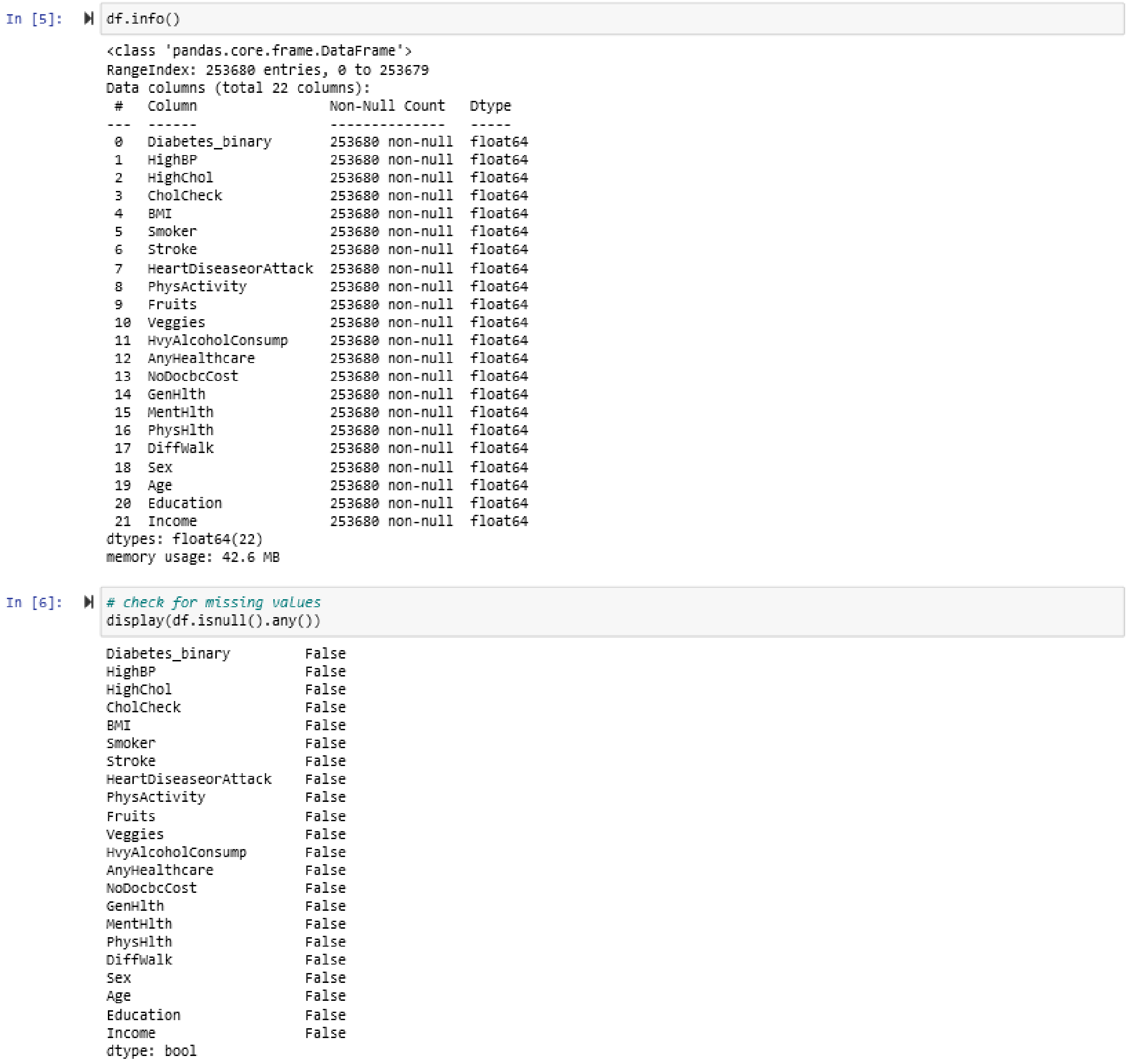
```
In [2]:  df= pd.read_csv(r"C:\Users\shabn\Documents\WGU - MSDA\D214\diabetes_binary_5050split_health_indicators_BRFSS2015.csv")
         df.head(5)

Out[2]:
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruits	...	AnyHealthcare	NoDocbcCost
0	0.0	1.0	0.0	1.0	26.0	0.0	0.0	0.0	1.0	0.0	...	1.0	0.0
1	0.0	1.0	1.0	1.0	26.0	1.0	1.0	0.0	0.0	1.0	...	1.0	0.0
2	0.0	0.0	0.0	1.0	26.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0
3	0.0	1.0	1.0	1.0	28.0	1.0	0.0	0.0	1.0	1.0	...	1.0	0.0
4	0.0	0.0	0.0	1.0	29.0	1.0	0.0	0.0	1.0	1.0	...	1.0	0.0

5 rows × 22 columns

```
In [3]:  df.shape

Out[3]:  (70692, 22)

In [4]:  df.columns

Out[4]:  Index(['Diabetes_binary', 'HighBP', 'HighChol', 'CholCheck', 'BMI', 'Smoker',
                'Stroke', 'HeartDiseaseorAttack', 'PhysActivity', 'Fruits', 'Veggies',
                'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'GenHlth',
                'MentHlth', 'PhysHlth', 'DiffWalk', 'Sex', 'Age', 'Education',
                'Income'],
               dtype='object')
```

4. Look for null values using .info() and .isnull().any(). In this step, it is confirmed that there are no null values in the data set.

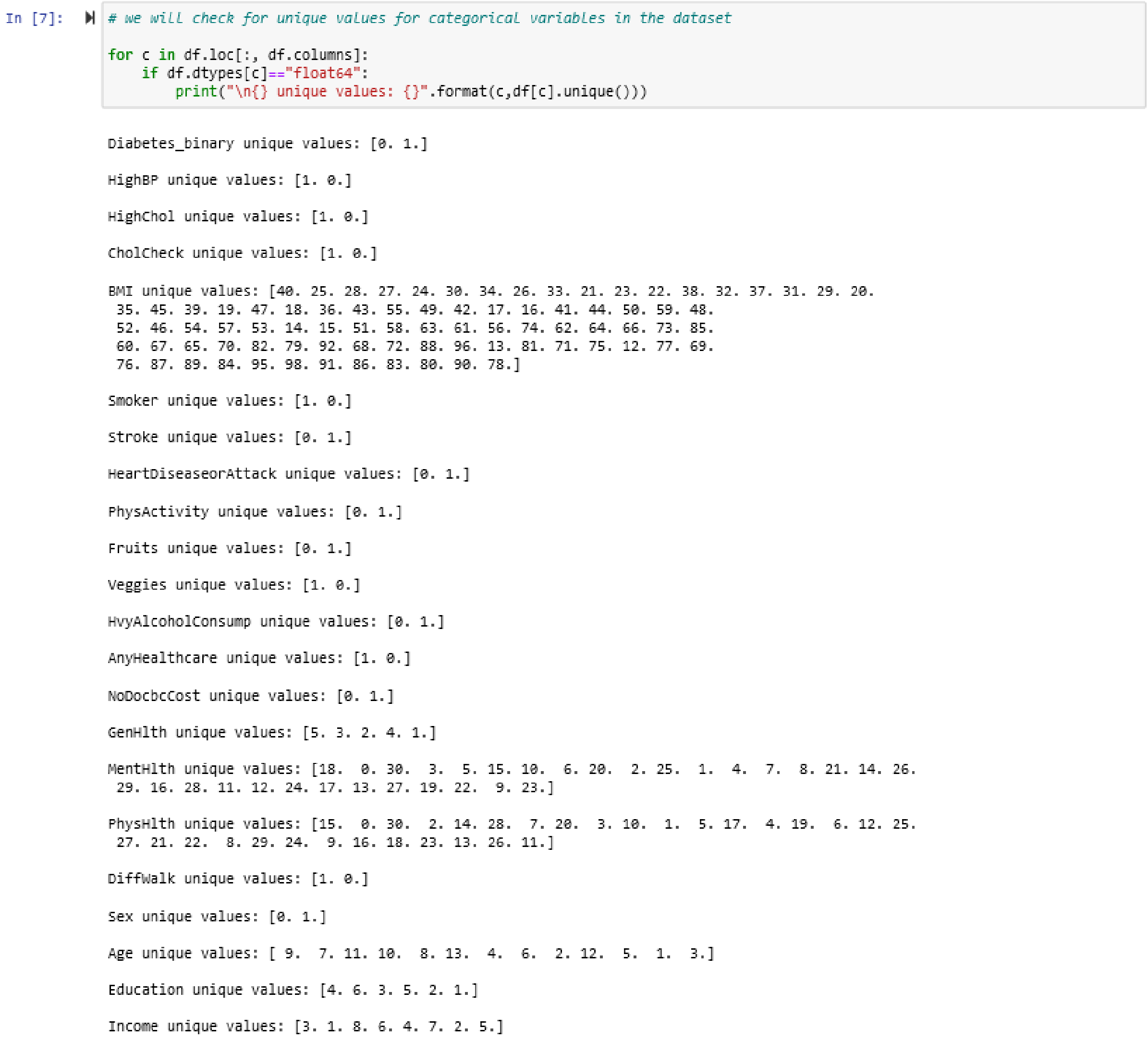


```
In [5]:  df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 253680 entries, 0 to 253679
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   Diabetes_binary       253680 non-null  float64
 1   HighBP                253680 non-null  float64
 2   HighChol              253680 non-null  float64
 3   CholCheck             253680 non-null  float64
 4   BMI                   253680 non-null  float64
 5   Smoker                253680 non-null  float64
 6   Stroke                253680 non-null  float64
 7   HeartDiseaseorAttack  253680 non-null  float64
 8   PhysActivity          253680 non-null  float64
 9   Fruits                253680 non-null  float64
 10  Veggies               253680 non-null  float64
 11  HvyAlcoholConsump     253680 non-null  float64
 12  AnyHealthcare         253680 non-null  float64
 13  NoDocbcCost           253680 non-null  float64
 14  GenHlth               253680 non-null  float64
 15  MentHlth              253680 non-null  float64
 16  PhysHlth              253680 non-null  float64
 17  DiffWalk              253680 non-null  float64
 18  Sex                   253680 non-null  float64
 19  Age                   253680 non-null  float64
 20  Education             253680 non-null  float64
 21  Income                253680 non-null  float64
dtypes: float64(22)
memory usage: 42.6 MB
```

```
In [6]:  # check for missing values
         display(df.isnull().any())

Diabetes_binary         False
HighBP                  False
HighChol                False
CholCheck               False
BMI                     False
Smoker                  False
Stroke                  False
HeartDiseaseorAttack    False
PhysActivity            False
Fruits                  False
Veggies                 False
HvyAlcoholConsump       False
AnyHealthcare           False
NoDocbcCost             False
GenHlth                 False
MentHlth                False
PhysHlth                False
DiffWalk                False
Sex                     False
Age                     False
Education               False
Income                  False
dtype: bool
```

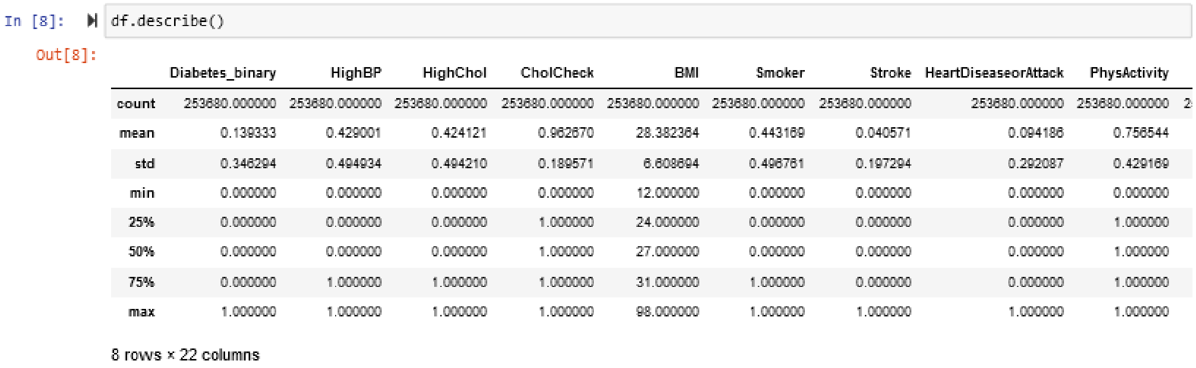
5. Check the values in the dataset. Since we are expecting most of the columns to be binary variables, we will create a for loop to return all the unique values of each column. In this step, it is confirmed that the values in the dataset are as expected. An advantage of using the for loop is that with a few lines of code, we can get all the unique values of our

columns in the data set. A disadvantage is that the for loop can be hard to create and requires some knowledge of for loops. One small note to make is that the BMI unique values go up to 98, which is very high for a BMI value. We will investigate this in detail in the second portion of the data cleaning.



```
In [7]:  # we will check for unique values for categorical variables in the dataset

         for c in df.loc[:, df.columns]:
             if df.dtypes[c]=="float64":
                 print("\n{} unique values: {}".format(c,df[c].unique()))
```

```
Diabetes_binary unique values: [0. 1.]

HighBP unique values: [1. 0.]

HighChol unique values: [1. 0.]

CholCheck unique values: [1. 0.]

BMI unique values: [40. 25. 28. 27. 24. 30. 34. 26. 33. 21. 23. 22. 38. 32. 37. 31. 29. 20.
 35. 45. 39. 19. 47. 18. 36. 43. 55. 49. 42. 17. 16. 41. 44. 50. 59. 48.
 52. 46. 54. 57. 53. 14. 15. 51. 58. 63. 61. 56. 74. 62. 64. 66. 73. 85.
 60. 67. 65. 70. 82. 79. 92. 68. 72. 88. 96. 13. 81. 71. 75. 12. 77. 69.
 76. 87. 89. 84. 95. 98. 91. 86. 83. 80. 90. 78.]

Smoker unique values: [1. 0.]

Stroke unique values: [0. 1.]

HeartDiseaseorAttack unique values: [0. 1.]

PhysActivity unique values: [0. 1.]

Fruits unique values: [0. 1.]

Veggies unique values: [1. 0.]

HvyAlcoholConsump unique values: [0. 1.]

AnyHealthcare unique values: [1. 0.]

NoDocbcCost unique values: [0. 1.]

GenHlth unique values: [5. 3. 2. 4. 1.]

MentHlth unique values: [18.  0. 30.  3.  5. 15. 10.  6. 20.  2. 25.  1.  4.  7.  8. 21. 14. 26.
 29. 16. 28. 11. 12. 24. 17. 13. 27. 19. 22.  9. 23.]

PhysHlth unique values: [15.  0. 30.  2. 14. 28.  7. 20.  3. 10.  1.  5. 17.  4. 19.  6. 12. 25.
 27. 21. 22.  8. 29. 24.  9. 16. 18. 23. 13. 26. 11.]

DiffWalk unique values: [1. 0.]

Sex unique values: [0. 1.]

Age unique values: [ 9.  7. 11. 10.  8. 13.  4.  6.  2. 12.  5.  1.  3.]

Education unique values: [4. 6. 3. 5. 2. 1.]

Income unique values: [3. 1. 8. 6. 4. 7. 2. 5.]
```



```
In [8]:  df.describe()
Out[8]:
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity
count	253680.000000	253680.000000	253680.000000	253680.000000	253680.000000	253680.000000	253680.000000	253680.000000	253680.000000
mean	0.139333	0.429001	0.424121	0.962670	28.382364	0.443169	0.040571	0.094186	0.756544
std	0.346294	0.494934	0.494210	0.189571	6.608694	0.496761	0.197294	0.292087	0.429169
min	0.000000	0.000000	0.000000	0.000000	12.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	1.000000	24.000000	0.000000	0.000000	0.000000	1.000000
50%	0.000000	0.000000	0.000000	1.000000	27.000000	0.000000	0.000000	0.000000	1.000000
75%	0.000000	1.000000	1.000000	1.000000	31.000000	1.000000	0.000000	0.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	98.000000	1.000000	1.000000	1.000000	1.000000

8 rows × 22 columns

All the steps above were to get an assessment of the data and to check for null values. An advantage to using read_csv() is that it is included in the highly used pandas package. Similarly, an advantage to using .head(), .shape, .columns, .describe() .info() and .isnull().any() is that it is included in the default Python features. A disadvantage is that the output is text that requires interpretation from the user.

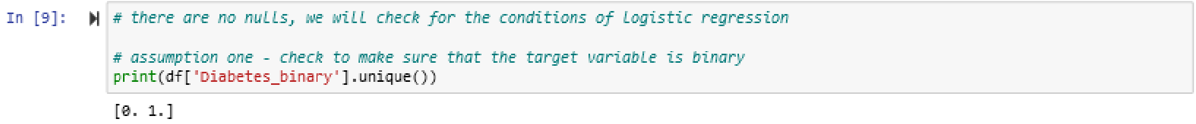
To use logistic regression, we need to ensure that our data meets the following data conditions:

1. The target variable must be binary.
2. There needs to be linearity between the target variable and all continuous variables.
3. There are no influential outliers in the dataset.
4. There is no multicollinearity.
5. All observations are independent.

(Leung, 2022.)

The following will go through the tools and techniques used to prepare and clean the data to check for the data requirements of logistic regression.

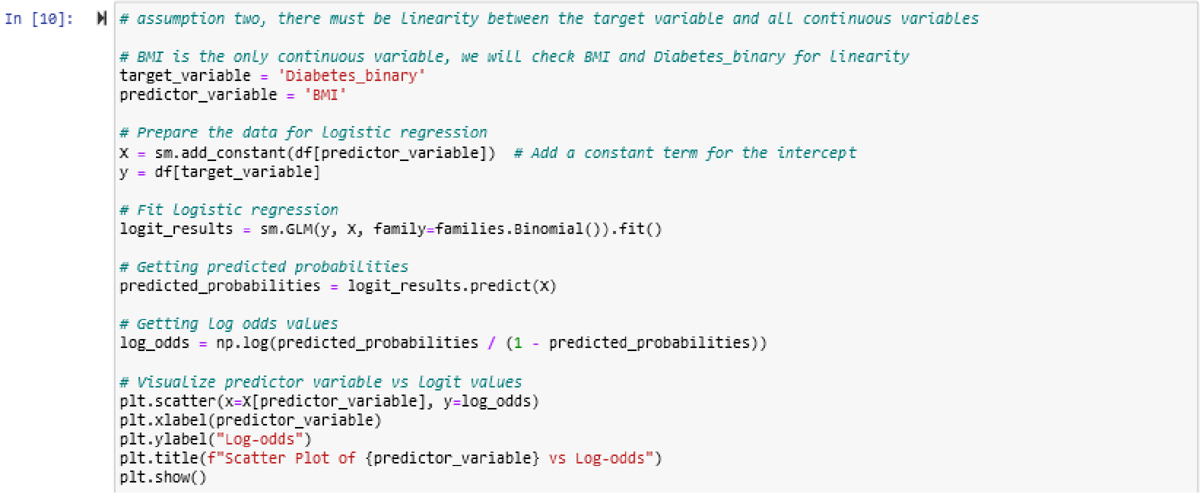
The first assumption of logistic regression is that the target variable needs to be binary. The below shows the use of .unique() to confirm that there are two outputs of the target variable. The .unique() function will display the distinct values of the “Diabetes_binary” variable. The below shows that the target variable is binary. One advantage of using the .unique() function is that it is included in Python’s default functions. One disadvantage can be that the output is not visually pleasing.



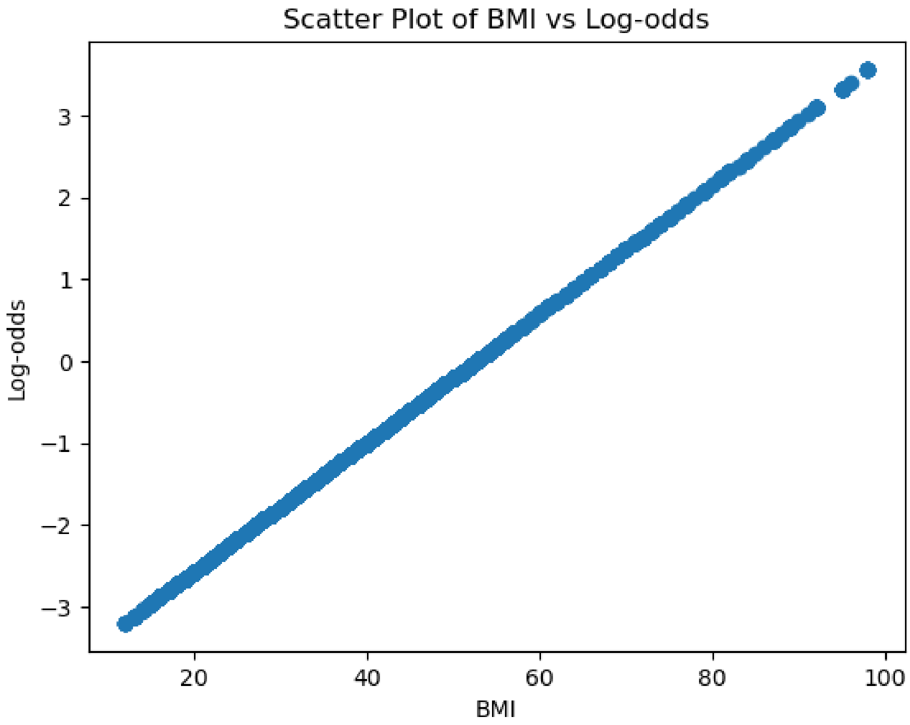
```
# there are no nulls, we will check for the conditions of logistic regression

# assumption one - check to make sure that the target variable is binary
print(df['Diabetes_binary'].unique())
```

```
[0. 1.]
```

The second assumption of logistic regression is that there is linearity between the target variable and continuous variables. For this analysis, we will be using data visualization to confirm that there is linearity between the target variable and continuous variables. The only continuous variable in our dataset is the “BMI” variable. We will be using a scatterplot to assess the linearity between the two variables. One advantage of a scatterplot is that it will explicitly display whether there is linearity. A disadvantage is that outliers are not as clear in a scatterplot. As such, the below is a data visualization that establishes that there is a linear relationship between BMI and Diabetes_binary.

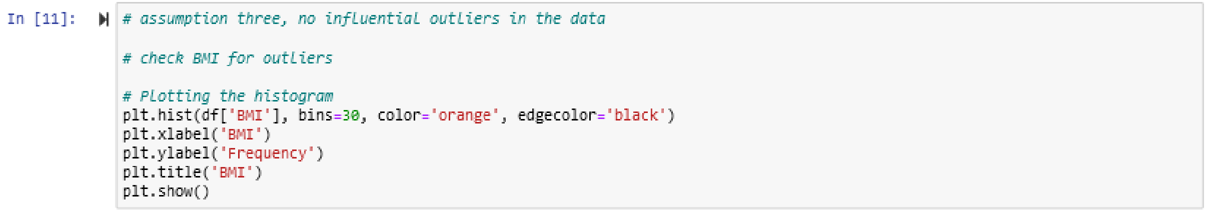


```
In [10]:  # assumption two, there must be linearity between the target variable and all continuous variables

          # BMI is the only continuous variable, we will check BMI and Diabetes_binary for linearity
          target_variable = 'Diabetes_binary'
          predictor_variable = 'BMI'

          # Prepare the data for logistic regression
          X = sm.add_constant(df[predictor_variable])  # Add a constant term for the intercept
          y = df[target_variable]

          # Fit logistic regression
          logit_results = sm.GLM(y, X, family=families.Binomial()).fit()

          # Getting predicted probabilities
          predicted_probabilities = logit_results.predict(X)

          # Getting log odds values
          log_odds = np.log(predicted_probabilities / (1 - predicted_probabilities))

          # Visualize predictor variable vs logit values
          plt.scatter(x=X[predictor_variable], y=log_odds)
          plt.xlabel(predictor_variable)
          plt.ylabel("Log-odds")
          plt.title(f"Scatter Plot of {predictor_variable} vs Log-odds")
          plt.show()
```



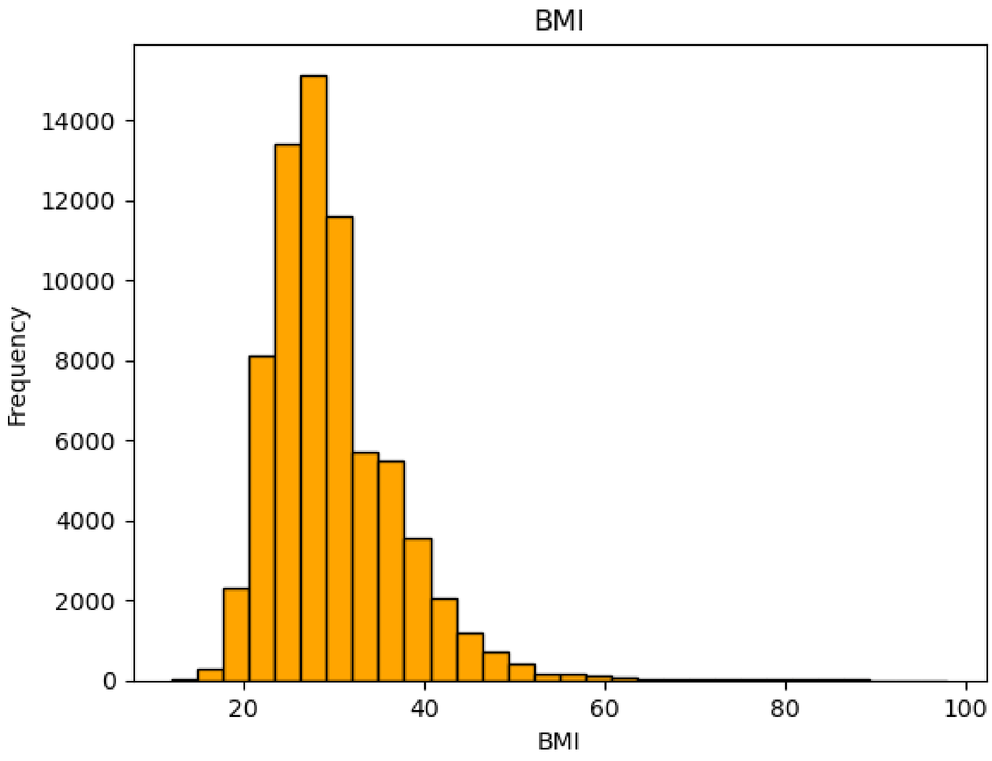
The third assumption is that there are no strong outliers. There are no outliers for variables that are binary or binned in groups. That leaves BMI as the only variable that needs to be checked for outliers. For this step, we will use data visualizations to assess for outliers in the data. We will use .hist() and .boxplot() from the matplotlib.pyplot package to create a histogram and boxplot.

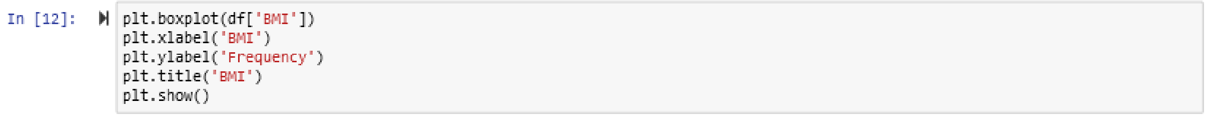
The histogram is used in this step to visualize any skewness or symmetry. The boxplot will explicitly show whether the data has any outliers.



In [11]:

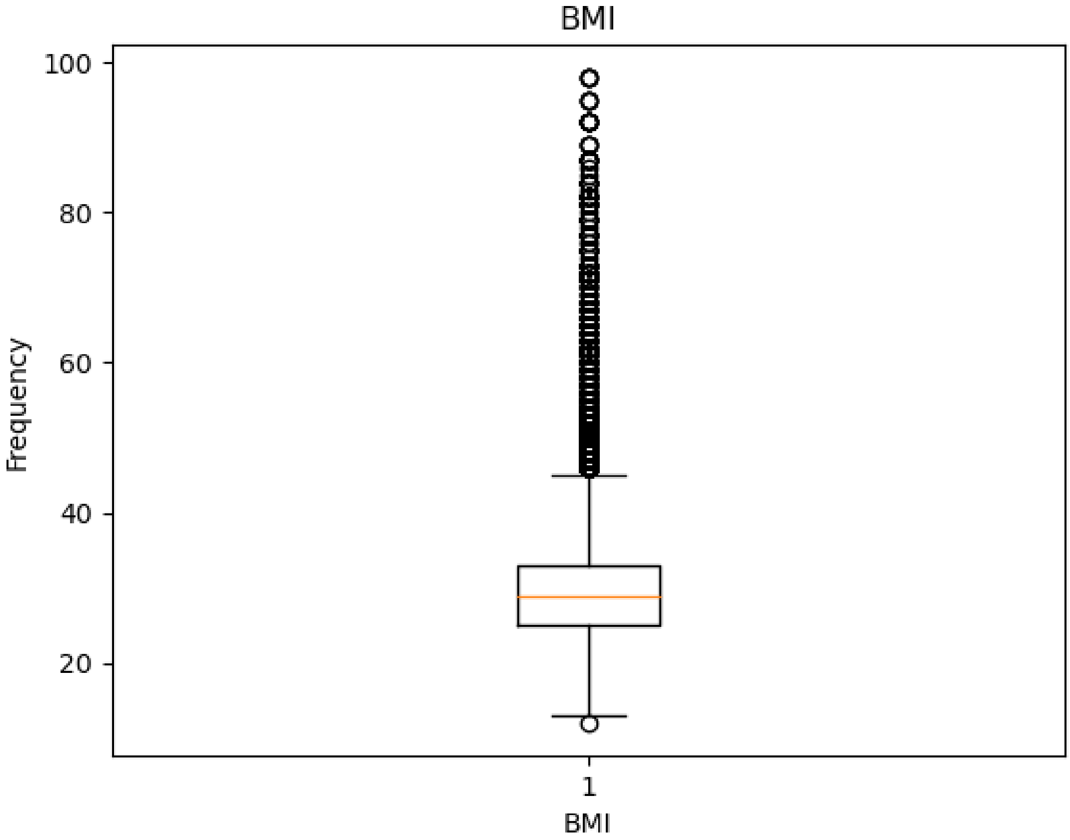
```
# assumption three, no influential outliers in the data

# check BMI for outliers

# Plotting the histogram
plt.hist(df['BMI'], bins=30, color='orange', edgecolor='black')
plt.xlabel('BMI')
plt.ylabel('Frequency')
plt.title('BMI')
plt.show()
```





In [12]:

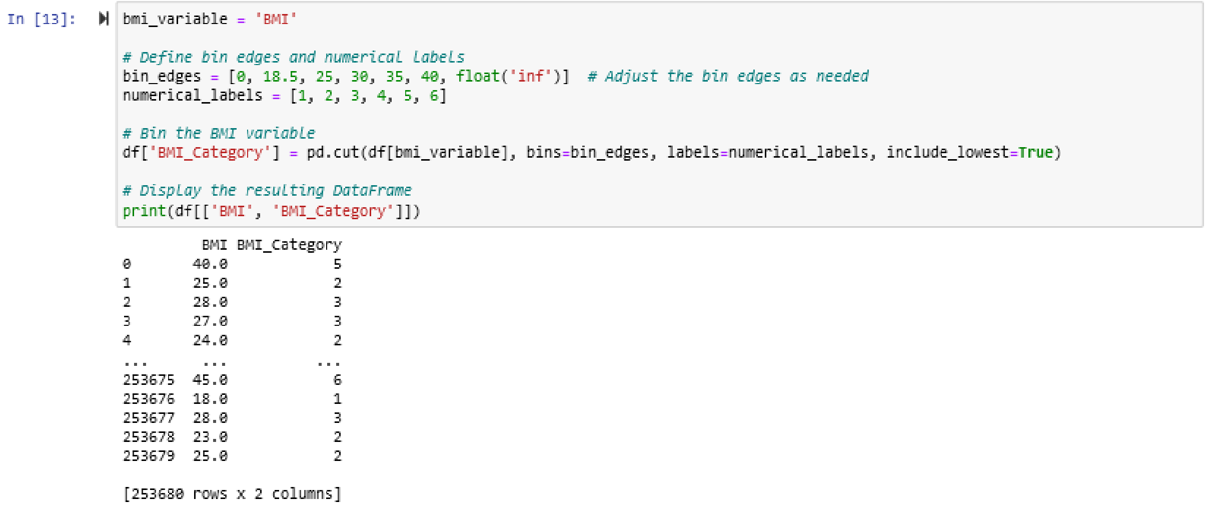
```
plt.boxplot(df['BMI'])
plt.xlabel('BMI')
plt.ylabel('Frequency')
plt.title('BMI')
plt.show()
```



The boxplot above shows that there are outliers in the data. The largest BMI in the data set is 98, this is a case of an outlier. An individual who is 5’7 would need to weigh about 625 lbs to have a BMI of 98. However, this would be considered a plausible outlier, since an individual can weigh 625 lbs. To mitigate this, we will create classes of BMI categories based on the CDC’s BMI subdivisions. The following is how the BMI classes will be delegated:

Dataset Value	**Description**	**CDC Class Name**
1	< 18.5	Underweight
2	18.5 - < 25	Healthy
3	25 - < 30	Overweight
4	30 - < 35	Obese: Class I

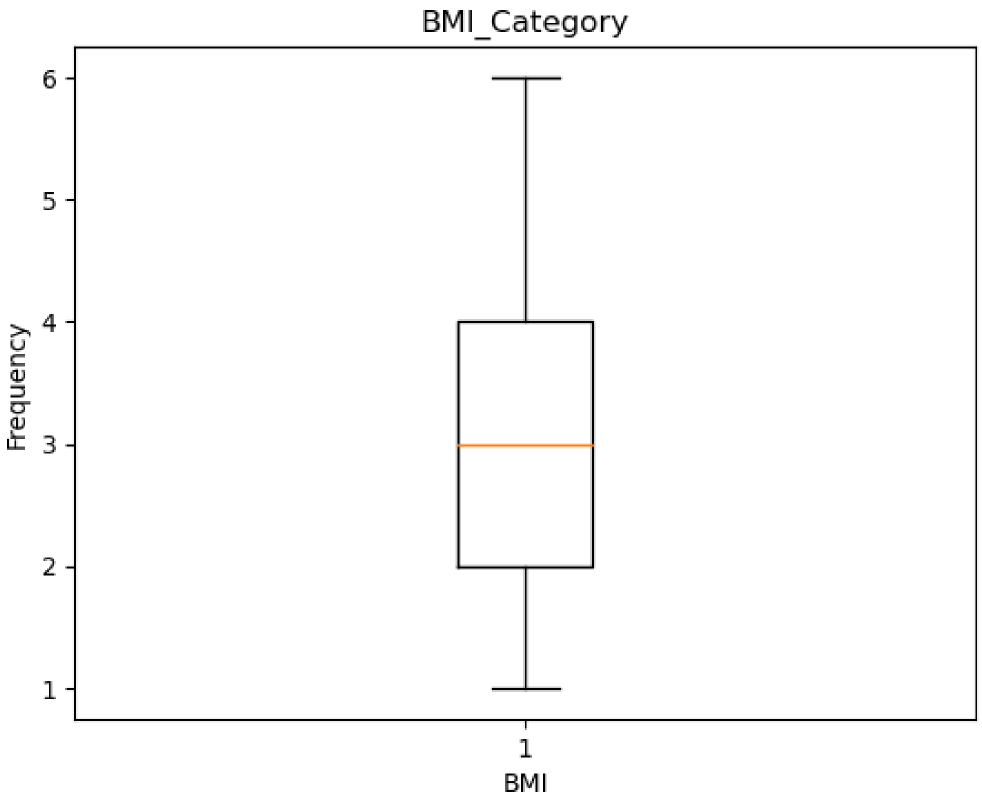
5	35 - < 40	Obese: Class II
6	>= 40	Obese: Class III (Severe)



```
In [13]:   bmi_variable = 'BMI'

           # Define bin edges and numerical labels
           bin_edges = [0, 18.5, 25, 30, 35, 40, float('inf')]  # Adjust the bin edges as needed
           numerical_labels = [1, 2, 3, 4, 5, 6]

           # Bin the BMI variable
           df['BMI_Category'] = pd.cut(df[bmi_variable], bins=bin_edges, labels=numerical_labels, include_lowest=True)

           # Display the resulting DataFrame
           print(df[['BMI', 'BMI_Category']])
```

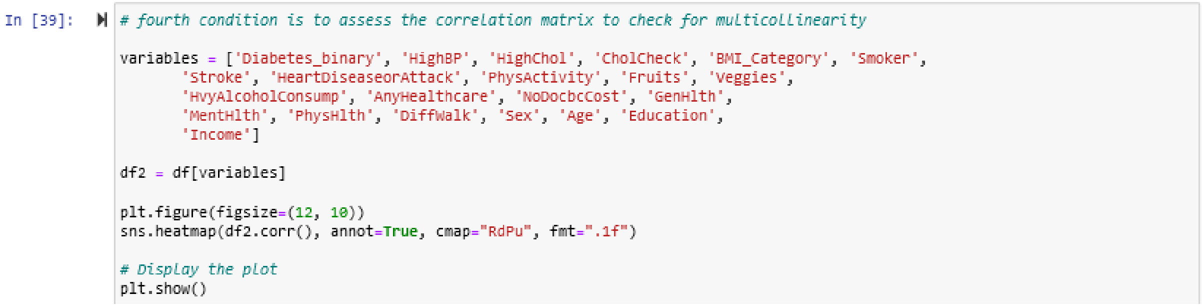
```
         BMI BMI_Category
0       40.0            5
1       25.0            2
2       28.0            3
3       27.0            3
4       24.0            2
...      ...          ...
253675  45.0            6
253676  18.0            1
253677  28.0            3
253678  23.0            2
253679  25.0            2

[253680 rows x 2 columns]
```

With the BMI data now being categorized, the boxplot is as follows:

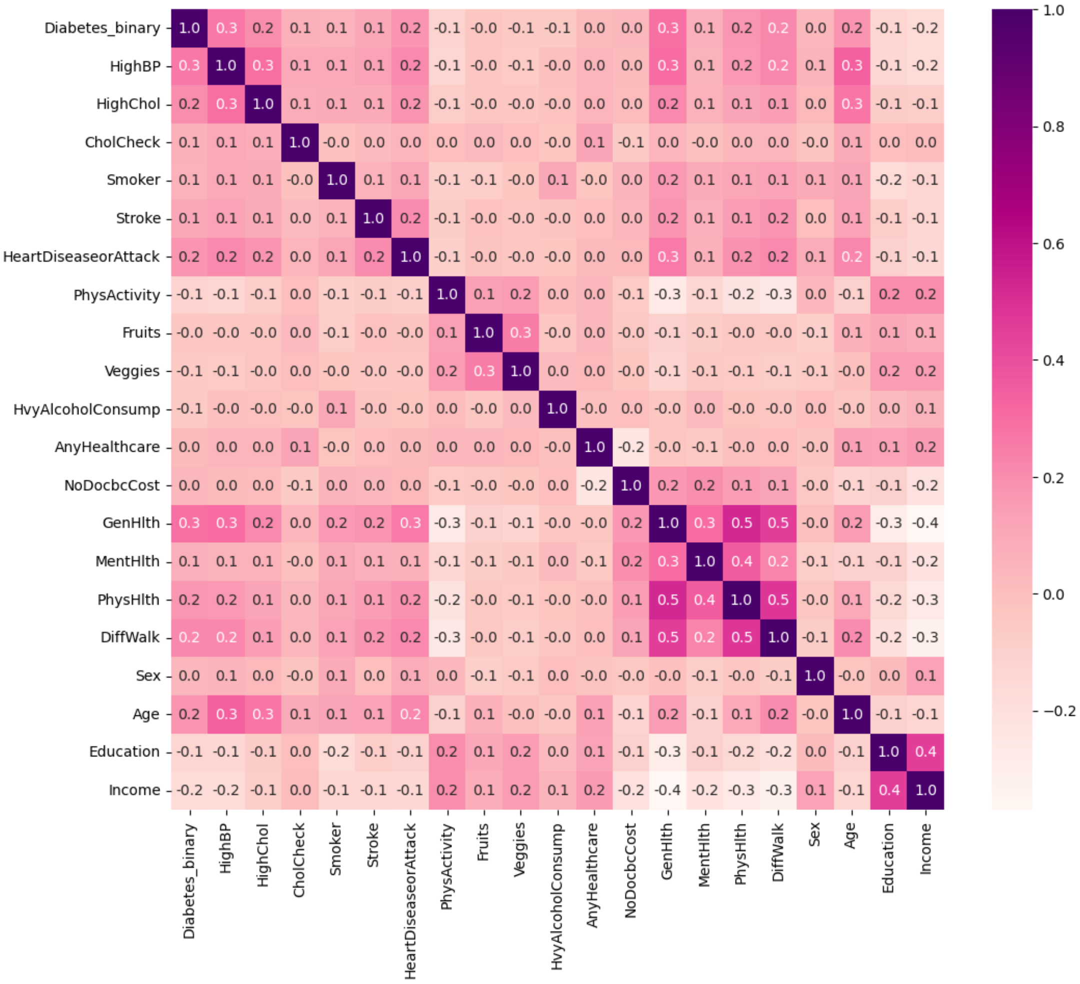


There are now no outliers in our dataset.

The fourth assumption is that there is an absence of multicollinearity. Multicollinearity refers to data that contains highly correlated independent variables. For this analysis, we will be assessing the correlation coefficient matrix. We will use the .corr() function from the pandas library to create a correlation matrix. We will be using the correlation matrix because it offers an easy-to-interpret data visualization to evaluate for multicollinearity. Any correlation coefficient with an absolute value above 0.7 is considered a strong correlation and will be remedied (Akoglu, 2018).



```
In [39]:    # fourth condition is to assess the correlation matrix to check for multicollinearity

            variables = ['Diabetes_binary', 'HighBP', 'HighChol', 'CholCheck', 'BMI_Category', 'Smoker',
                   'Stroke', 'HeartDiseaseorAttack', 'PhysActivity', 'Fruits', 'Veggies',
                   'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'GenHlth',
                   'MentHlth', 'PhysHlth', 'DiffWalk', 'Sex', 'Age', 'Education',
                   'Income']

            df2 = df[variables]

            plt.figure(figsize=(12, 10))
            sns.heatmap(df2.corr(), annot=True, cmap="RdPu", fmt=".1f")

            # Display the plot
            plt.show()
```



There are no variables that correlate with an absolute above the value of 0.7, the multicollinearity assumption is met.

Finally, the fifth assumption is that all observations need to be independent of one another. The independence assumption is met since each record on the dataset consists of an individual's survey results.

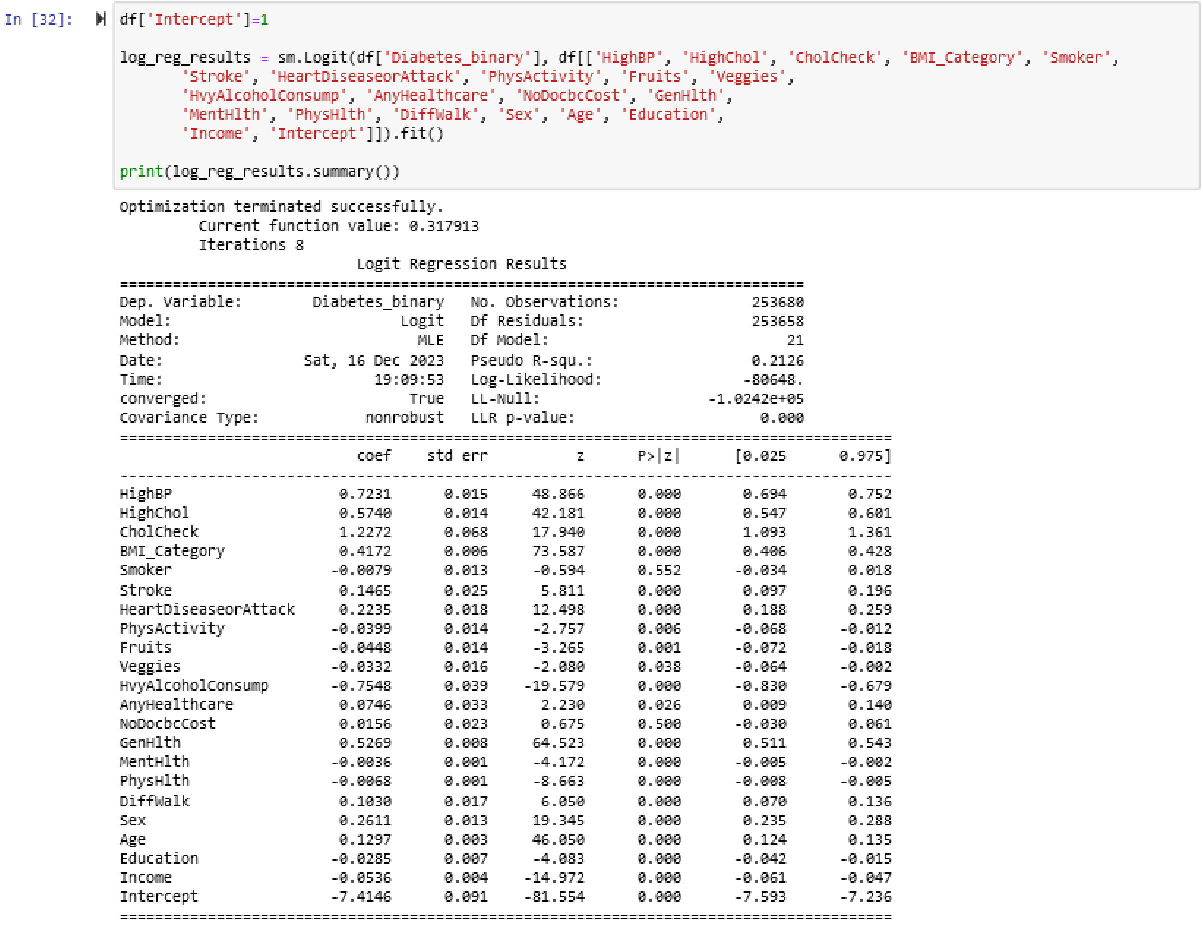
In the above, we went through the general data cleaning steps along with checking for the assumptions for logistic regression. The tools and techniques were described for each step along with a justification for why the specific tool or technique was used. Additionally, for each step, there was an assessment of an advantage and disadvantages of the tools and techniques used for the data preparation process.

Analysis

Logistic regression is used to predict the probability of an event (the target variable) occurring. For logistic regression, we will be looking at the key outputs to determine the efficiency of the regression analysis:

1. **Coefficient of the variables**: The coefficient of the variables determines the weight of the predictor variable. The larger the coefficient, the larger the impact it has on predicting whether one has diabetes or not.
2. **Z-Score and P-Score**: For each variable, there is a p and z score that measure the statistical significance of the variable. In this analysis, we will drop any values that have a p-value above 0.05.
3. **Pseudo R-Squared**: The pseudo R-squared value indicates how well the model will explain variation in the target variable. This value assesses the degree of how well the model can accurately predict the target variable. Ideally, the pseudo R-squared value is as high as possible.

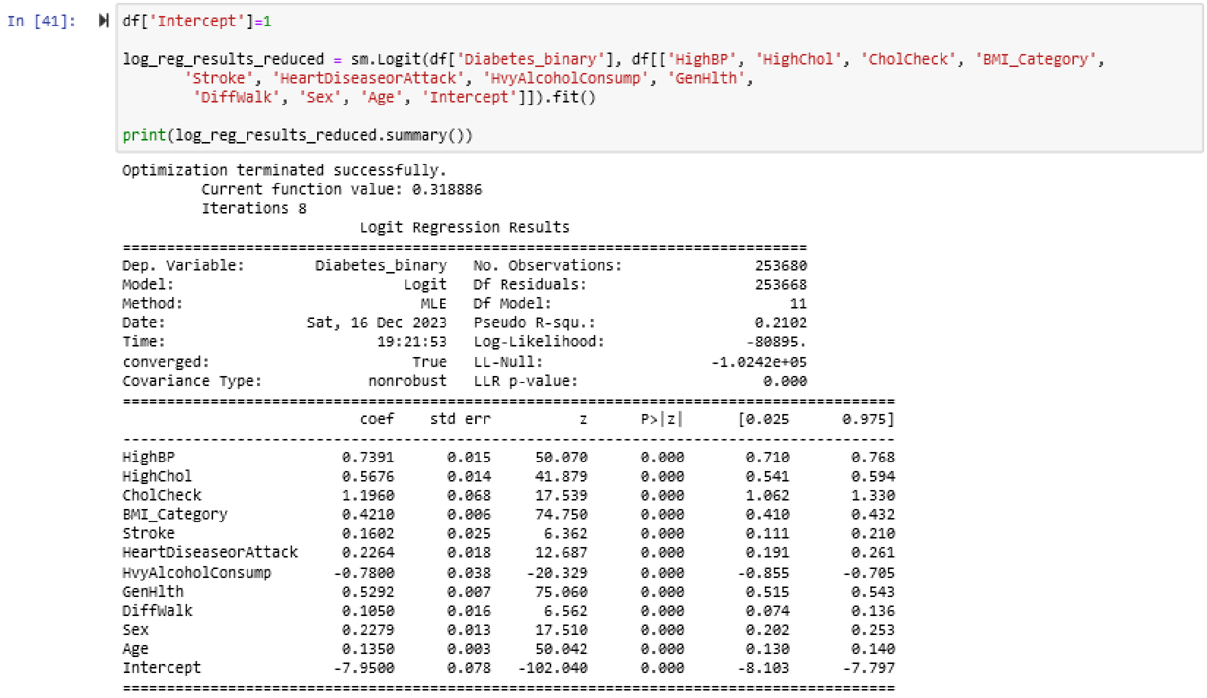
To conduct the logistic regression, we will need to add an intercept. The intercept represents the log odds of the target variable occurring when the predictor variables are equal to zero. Secondly, we will need to use the Logit() function from the statsmodel library. In the function, we will include the target variable and the predictor variables to be considered for the logistic regression.



```
In [32]:  df['Intercept']=1

          log_reg_results = sm.Logit(df['Diabetes_binary'], df[['HighBP', 'HighChol', 'CholCheck', 'BMI_Category', 'Smoker',
                 'Stroke', 'HeartDiseaseorAttack', 'PhysActivity', 'Fruits', 'Veggies',
                 'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'GenHlth',
                 'MentHlth', 'PhysHlth', 'DiffWalk', 'Sex', 'Age', 'Education',
                 'Income', 'Intercept']]).fit()

          print(log_reg_results.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.317913
         Iterations 8
                           Logit Regression Results
==============================================================================
Dep. Variable:        Diabetes_binary   No. Observations:               253680
Model:                          Logit   Df Residuals:                   253658
Method:                           MLE   Df Model:                           21
Date:                Sat, 16 Dec 2023   Pseudo R-squ.:                  0.2126
Time:                        19:09:53   Log-Likelihood:                -80648.
converged:                       True   LL-Null:                   -1.0242e+05
Covariance Type:            nonrobust   LLR p-value:                     0.000
========================================================================================
                           coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------
HighBP                   0.7231      0.015     48.866      0.000       0.694       0.752
HighChol                 0.5740      0.014     42.181      0.000       0.547       0.601
CholCheck                1.2272      0.068     17.940      0.000       1.093       1.361
BMI_Category             0.4172      0.006     73.587      0.000       0.406       0.428
Smoker                  -0.0079      0.013     -0.594      0.552      -0.034       0.018
Stroke                   0.1465      0.025      5.811      0.000       0.097       0.196
HeartDiseaseorAttack     0.2235      0.018     12.498      0.000       0.188       0.259
PhysActivity            -0.0399      0.014     -2.757      0.006      -0.068      -0.012
Fruits                  -0.0448      0.014     -3.265      0.001      -0.072      -0.018
Veggies                 -0.0332      0.016     -2.080      0.038      -0.064      -0.002
HvyAlcoholConsump       -0.7548      0.039    -19.579      0.000      -0.830      -0.679
AnyHealthcare            0.0746      0.033      2.230      0.026       0.009       0.140
NoDocbcCost              0.0156      0.023      0.675      0.500      -0.030       0.061
GenHlth                  0.5269      0.008     64.523      0.000       0.511       0.543
MentHlth                -0.0036      0.001     -4.172      0.000      -0.005      -0.002
PhysHlth                -0.0068      0.001     -8.663      0.000      -0.008      -0.005
DiffWalk                 0.1030      0.017      6.050      0.000       0.070       0.136
Sex                      0.2611      0.013     19.345      0.000       0.235       0.288
Age                      0.1297      0.003     46.050      0.000       0.124       0.135
Education               -0.0285      0.007     -4.083      0.000      -0.042      -0.015
Income                  -0.0536      0.004    -14.972      0.000      -0.061      -0.047
Intercept               -7.4146      0.091    -81.554      0.000      -7.593      -7.236
========================================================================================
```

The analysis returned a pseudo-R squared value of 0.2126. Predictor variables “Smoker” and “NoDocbcCost” will be dropped since both variables have a p-value above 0.05. Additionally, we will reduce the model by dropping all variables that have a coefficient below an absolute value of 0.10. The reduced model will include predictor variables 'HighBP', 'HighChol', 'CholCheck', 'BMI_Category', 'Stroke', 'HeartDiseaseorAttack', 'HvyAlcoholConsump', 'GenHlth', 'DiffWalk', 'Sex', and 'Age'.



In [41]:

```
df['Intercept']=1

log_reg_results_reduced = sm.Logit(df['Diabetes_binary'], df[['HighBP', 'HighChol', 'CholCheck', 'BMI_Category',
       'Stroke', 'HeartDiseaseorAttack', 'HvyAlcoholConsump', 'GenHlth',
        'DiffWalk', 'Sex', 'Age', 'Intercept']]).fit()

print(log_reg_results_reduced.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.318886
         Iterations 8
                           Logit Regression Results
==============================================================================
Dep. Variable:        Diabetes_binary   No. Observations:               253680
Model:                          Logit   Df Residuals:                   253668
Method:                           MLE   Df Model:                           11
Date:                Sat, 16 Dec 2023   Pseudo R-squ.:                  0.2102
Time:                        19:21:53   Log-Likelihood:                -80895.
converged:                       True   LL-Null:                   -1.0242e+05
Covariance Type:            nonrobust   LLR p-value:                     0.000
========================================================================================
                          coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------
HighBP                  0.7391      0.015     50.070      0.000       0.710       0.768
HighChol                0.5676      0.014     41.879      0.000       0.541       0.594
CholCheck               1.1960      0.068     17.539      0.000       1.062       1.330
BMI_Category            0.4210      0.006     74.750      0.000       0.410       0.432
Stroke                  0.1602      0.025      6.362      0.000       0.111       0.210
HeartDiseaseorAttack    0.2264      0.018     12.687      0.000       0.191       0.261
HvyAlcoholConsump      -0.7800      0.038    -20.329      0.000      -0.855      -0.705
GenHlth                 0.5292      0.007     75.060      0.000       0.515       0.543
DiffWalk                0.1050      0.016      6.562      0.000       0.074       0.136
Sex                     0.2279      0.013     17.510      0.000       0.202       0.253
Age                     0.1350      0.003     50.042      0.000       0.130       0.140
Intercept              -7.9500      0.078   -102.040      0.000      -8.103      -7.797
========================================================================================
```

The reduced regression model returns a pseudo-R squared value of 0.2102, which is a reduction of 0.0024 from the original model. The model was reduced from 21 variables to 11 variables. One advantage of using the technique above is that we can dissect the summary of the Logit() function to reduce the variables in the model. It is beneficial to reduce the model to increase computational efficiency by defining the predictor values that have statistical significance. One disadvantage of the technique used above is that it is not the most efficient method for dimensionality reduction. This method would not be optimal for a dataset with many predictor variables since it would be required to analyze each predictor variable's p-score and coefficient.

Data Summary and Implications

The final reduced regression equation is as follows:

$$L = 0.74 * HighBP + 0.57 * HighChol + 1.20 * CholCheck + 0.42 * BMICategory + 0.16 * Stroke + 0.23 * HeartDiseaseorAttack - 0.78 * HvyAlcoholConsump + 0.53 * GenHlth + 0.11 * DiffWalk + 0.23 * Sex + 0.14 * Age$$

The final logistic regression model yielded a pseudo-R squared value of 0.2102 with less than 0.01 difference from the initial model. The pseudo R-squared value indicates that the model and the predictor variables explain 21.02% of the variance of the target variable, “Diabetes_binary”. This indicates that we are able to use the regression model with 21% confidence in predicting whether an individual has diabetes using the reduced predictor variables. The reduced model also has nearly half of the initial model’s dimensions (reducing the dimensions from 21 to 11 variables). Each coefficient describes the estimated change in the log-odds of the response variable when the coefficient’s predictor variable increases by one. For example, BMI_Category has a regression coefficient of 0.42. We calculate the log-odds by taking eˆ(0.42) = 1.52. This indicates that an individual is 1.52 times as likely to have diabetes for every unit of increase of BMI_Category.

A limitation of the analysis above is that the fit of the model is a smaller value of 0.2102. The low pseudo-R squared value indicates that the predictor variables do not explain the variance of “Diabetes_binary” significantly. However, in the context of the research question and survey, having 11 variables that explain 21% of the variance of the target event is productive given that a telephone survey is low-cost and generally does not require much preparation from the surveyors. Additionally, the majority of the questions of the survey are simple “Yes” or “No” questions which make the survey easy to conduct. Based on our regression analysis, the main predictor variables are 'HighBP', 'HighChol', 'CholCheck', 'BMI_Category', 'Stroke', 'HeartDiseaseorAttack', 'HvyAlcoholConsump', 'GenHlth', 'DiffWalk', 'Sex', and 'Age'. 5 of the 11 variables are related to heart or cholesterol ('HighBP', 'HighChol', 'CholCheck', 'Stroke', 'HeartDiseaseorAttack'). Additionally, the questions that remain in the reduced model are all objective questions (with the exception of ‘GenHlth’) that require no opinion or interpretation from the one being surveyed. With these observations in mind, I would suggest to the CDC and any parties that are interested in surveying and monitoring diabetes to consider using logistic regression models when assessing the effectiveness of the survey questions being asked. A model fitness of 0.21 is standard, I would suggest the interested parties to continue to ask objective questions and include additional survey questions that are related the heart health and cholesterol. One final suggestion would be to include other statistical analyses such as classification models (such as Decision Trees and Naive Bayes) to determine the characteristics of individuals who have been diagnosed with diabetes. The suggestions above will give potentially more insight into the significant components of the BRFSS survey data and direction on how to optimize the survey questions.

For future suggestions on the future study of the dataset, I would recommend that the researcher attempts a classification model on the dataset. Using the classification model, the researcher can get an insight into whether the data fits into different classes based on the BRFSS survey data features. This classification model could potentially be a better predictor into whether an individual has diabetes based on the survey. Additionally, I would suggest the researcher to consider if there are any other integrations that could be considered in addition to the BRFSS survey data. The additional data can give a more complete view of the research topic and could give more insights on future survey questions to add to the annual survey.

Conclusion

Our research question was: What key components, as determined by logistic regression, are most significant in predicting the likelihood of diabetes in a given population using the BRFSS survey data? This analysis found that the key components in the BRFSS survey are the 'HighBP', 'HighChol', 'CholCheck', 'BMI_Category', 'Stroke', 'HeartDiseaseorAttack', 'HvyAlcoholConsump', 'GenHlth', 'DiffWalk', 'Sex', and 'Age' variables in the survey. Additionally, with a fit of 0.21, we can reject the null hypothesis and state that there is a significant association between diabetes diagnosis (‘Diabetes_binary’) reported high blood pressure ('HighBP'), reported high cholesterol ('HighChol'), reported cholesterol checks ('CholCheck'), BMI '(BMI_Category',) reported history with strokes ('Stroke'), reported history with heart disease or heat attacks ('HeartDiseaseorAttack'), reported alcohol consumption ('HvyAlcoholConsump'), reported health rating ('GenHlth'), reported difficulty walking ('DiffWalk'), reported sex ('Sex'), and reported age ('Age') variables.

Citations

Akoglu, Haldun. “User’s Guide to Correlation Coefficients.” *Turkish Journal of Emergency Medicine*, U.S. National Library of Medicine, 7 Aug. 2018, www.ncbi.nlm.nih.gov/pmc/articles/PMC6107969/#bib5.

“Diabetes Statistics.” *DRIF*, 10 Oct. 2023, diabetesresearch.org/diabetes-statistics/.

“Facts & Figures.” *International Diabetes Federation*, 14 Sept. 2023, idf.org/about-diabetes/diabetes-facts-figures/.

“Facts about Diabetes.” *Johns Hopkins Medicine*, www.hopkinsmedicine.org/health/conditions-and-diseases/diabetes. Accessed 16 Dec. 2023.

Leung, Kenneth. “Assumptions of Logistic Regression, Clearly Explained.” *Medium*, Towards Data Science, 13 Sept. 2022, towardsdatascience.com/assumptions-of-logistic-regression-clearly-explained-44d85a22b290.

“Life Doesn’t End with Type 2 Diabetes.” *Type 2 Diabetes* | *ADA*, diabetes.org/living-with-diabetes/type-2. Accessed 16 Dec. 2023.

Sincero, Sarah Mae (Mar 18, 2012). Advantages and Disadvantages of Surveys. Retrieved Dec 05, 2023 from Explorable.com: https://explorable.com/advantages-and-disadvantages-of-surveys

Teboul, Alex. “Diabetes Health Indicators Dataset.” *Kaggle*, 8 Nov. 2021, www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?select=diabetes_binary_5050split_health_indicators_BRFSS2015.csv.