I. Introduction

1. Exploring the data set

The data set, *diabetes_5050.csv*, is taken from a survey conducted in the US in 2015 about diabetes among the US citizens. The data consists of 70,692 survey responses. This data consists of 21 features with 1 response variable.

- Response Variable

  The column name of the response variable in this data is a categorical data, **Diabetes_binary** which consists of **0** and **1**. While **0** indicates that the person does not have diabetes, **1** indicates that the person classified as prediabetes or having diabetes. In this data, the proportion of people with having diabetes is the same as those who does not, which is **50:50**.

- Input Variables

  There are some Categorical inputs:

  - **HighBP**: 0 = no high BP; 1 = high BP.
  - **HighChol**: 0 = no high cholesterol; 1 = high cholesterol.
  - **CholCheck**: 0 = no cholesterol checks in 5 years; 1 = has cholesterol check in 5 years.
  - **Smoker**: Have smoked at least 100 cigarettes (= 5 packs) in your life? 0 = no; 1 = yes.
  - **Stroke**: had been diagnosed with a stroke 0 = No; 1 = Yes.
  - **HeartDiseaseorAttac**k: coronary heart disease (CHD) or myocardial infarction (MI): 0 = no; 1 = yes.
  - **PhysActivity**: physical activity in past 30 days, not including jobs: 0 = no; 1 = yes.
  - **Fruits**: consume fruits one or more times per day: 0 = no; 1 = yes.
  - **Veggies**: consume vegetables 1 or more times per day: 0 = no; 1 = yes.
  - **HvyAlcoholConsump**: heavy drinkers (adult men having more than 14 drinks per week and adult women having more than 7 drinks per week): 0 = no; 1 = yes.
  - **AnyHealthcare**: Do you have any kind of health care coverage (including health insurance, prepaid plans or government plans)? 0 = no; 1 = yes.
  - **NoDocbcCost**: was there 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**: general health: 1 = excellent; 2 = very good; 3 = good; 4 = fair; 5 = poor.
  - **DiffWalk**: have serious difficulty walking or climbing stairs? 0 = no, 1 = yes.
  - **Sex**: 0 = female; 1 = male.
  - **Age**: 13 categories: 1 = age from 18 to 24; ....; 9 = age 60 to 64; 13 = age 80 or above.
  - **Education**: Education level scale 1 to 6: 1 = never attended school or only kindergarten; 2 = elementary; ....

- **Income**: Income scale 1 to 8: 1 = less than 10k; ...; 5 = less than 35k; ...; 8 = 75k or more.

There are also Numerical(quantitative) inputs:

- **BMI**: Body mass index
- **MentHlth**: How many days in the past 30 days feel that your mental health is not in good condition.
- **PhysHlth**: For how many days during the past 30 days was your physical health not good.

2. Purpose

Choosing a classification method for predicting diabetes status; and propose the best classifier. Investigate on the goodness of fit of that classifier.

II. Statistical Procedures Used

1. Association between each input variable and response variable.

a. Categorical variables

For categorical variables, especially those with binary response (0 and 1), I check the association using **odd ratio**. I perform the odd ratio calculation for each categorical input and response. I drop the inputs with odd ratio **0.3< x <1** (for the negative association) and **1 < x < 3** (for positive association).

b. Numerical variables

For numerical inputs, I use boxplot to check whether the shape and distribution of the boxplot is difference for each value of response variable (0 and 1). As stated in the instruction, ordinal variables should be treated as numerical instead of categorical.

Based on these methods, I decided to drop a few variables: **Smoker, Physactivity, Fruits, Veggies, HvyAlcoholConsump, AnyHealthcare, NoDocbcCost, PhysHlth, Sex, Education**.

2. Splitting the data

Firstly, I split the data into two (**data_0** and **data_1**) based on the response variable, Diabetes binary, to make sure that the proportion of response variable in each subsequent data set unchanged. Then, I will split each data into five data sets (**data1**, **data2**, **data3**, **data4**, and **data5**) to be the test data while performing n fold cross-validation. Then, I create other 5 data sets (**testdata1**, **testdata2**, **testdata3**, **testdata4**, and **testdata5**) to become the train data.

3. Building classifiers

I use four classifiers for analysing the data.

a. KNN

For KNN, I use the filtered data, which I obtain from the odd ratio and boxplot. I perform the KNN five times for five different test data and train data. For the value of K, I choose from **one to twenty** (inclusive) and also **237** and **238**, which is the approximate square root of number of total observations. Before building the KNN model and predicting the response for test data, I standardize all the input variables using function **scale** since KNN rely on distance on its classification. I store accuracy value of each fold in different vector. Then, I take the average from the vectors for each different value of k and I store it in **accuracy_knn** vector. The KNN code take a lot of time to be finally executed.

b. Decision Tree

For Decision Tree, I use all the input variables in the data set since Decision Tree can filter which variable is most significant for the model. I perform the Decision Tree on 5 different train data and test data and calculate the average accuracy. To obtain the highest accuracy, I try this model on different value of **cp** (0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000, 100000). I use **"class"** method in this classifier since the response is a binary. I also specified that **split = 'information'** for the classifier. Similar to KNN, I also store the accuracy of each fold in different vectors. Then, I take the average from the vectors for each different value of cp and I store it in **accuracy_DT** vector.

c. Naïve Bayes

For Naïve Bayes, I also use the filtered data (the same variables I use for KNN). For this classifier, I set the threshold to be a sequence of **21** number with equal differences from **0.00** to **1**. Then, using the predicted test data, I will classify the response to **1** if the predicted value is greater or equal to subsequent threshold value and **0** otherwise. I perform the classification five times and calculate the average accuracy given certain value of threshold. I also store the accuracy of each fold in different vectors. Then, I take the average from the vectors for each different value of threshold and I store it in **accuracy_NB** vector.

d. Logistic Regression

For Logistic Regression, I use all the input variables. Then, in each iteration, I will drop a variable with the highest p-value (the most unsignificant). I do this steps eleven times, leaving ten input variables (**HighBP, HighChol, CholCheck, BMI, HeartDiseaseorAttack, HvyAlcoholConsump, GenHlth, Sex, Age,** and **Income)** to form the model. For this classifier, I set the threshold to be a sequence of **21** number with equal differences from **0.00** to **1**. Then, using the predicted test data, I will classify the response to **1** if the predicted value is greater or equal to subsequent threshold value and **0** otherwise. I perform the classification five times and calculate the average accuracy given certain value of threshold. I store the accuracy of each fold in different vectors. Then, I take the average from the vectors for each different value of threshold and I store it in **acc_logit** vector.

4. Examining the goodness of fit

I choose accuracy to examine the goodness of fit for each classifier. This is the accuracy I get from each classifier after using n fold cross-validation to calculate the average.

a. KNN

Table 1. The Accuracy of KNN for each value of K.

K	1	2	3	4	5	6	7	8
Accuracy	0.688	0.693	0.707	0.707	0.715	0.716	0.721	0.722
K	9	10	11	12	13	14	15	16

Accuracy	0.725	0.725	0.728	0.728	0.730	0.729	0.731	0.731
K	17	18	19	20	237	238		
Accuracy	0.732	0.731	0.732	0.732	0.736	0.736		

The highest accuracy (**0.736**) is obtained when K equal to **237** and **238**. However, choosing the KNN as a classifier for large data set will require a long running time, which might not be efficient for this case.

b. Decision Tree

Table 2. The accuracy of Decision Tree for different value of cp.

cp	0.00001	0.0001	0.001	0.01	0.1	1
Accuracy	0.707	0.742	0.737	0.722	0.686	0.500
cp	10	100	1000	10,000	100,000	
Accuracy	0.500	0.500	0.500	0.500	0.500	

The highest accuracy is obtained when cp is equal to **0.0001** with accuracy around **0.742**. However, the picture is Tree is quite small and can be barely seen. Then, choosing cp equal to **0.001** might be better for this case.

c. Naïve Bayes

Table 3. The accuracy of Naïve Bayes model for different threshold.

Threshold	0	0.05	0.1	0.15	0.2	0.25	0.3
Accuracy	NA	0.676	0.706	0.712	0.721	0.725	0.726
Threshold	0.35	0.4	0.45	0.5	0.55	0.6	0.65
Accuracy	0.726	0.725	0.722	0.718	0.714	0.710	0.704
Threshold	0.7	0.75	0.8	0.85	0.9	0.95	1
Accuracy	0.698	0.691	0.685	0.677	0.668	0.654	NA

The highest accuracy (**0.726**) is obtained when Threshold is equal to **0.3 between 0.35**. However, choosing Naïve Bayes as classifier will give lower accuracy compared to other classifiers.

d. Logistic Regression

Table 4. The accuracy of Linear Regression model for different threshold.

Threshold	0	0.05	0.1	0.15	0.2	0.25	0.3
Accuracy	NA	0.535	0.596	0.643	0.678	0.705	0.724
Threshold	0.35	0.4	0.45	0.5	0.55	0.6	0.65
Accuracy	0.737	0.745	0.748	0.748	0.742	0.732	0.715
Threshold	0.7	0.75	0.8	0.85	0.9	0.95	1
Accuracy	0.692	0.663	0.630	0.591	0.549	0.513	NA

The highest accuracy (**0.748**) is obtained when Threshold is around **0.45** and **0.5** which also the highest accuracy among all the classifiers.

III. Summary of Statistical Findings

After comparing the accuracy of each classifier, it can be concluded that **Logistic Regression** with threshold value between **0.45** and **0.5** will give the highest accuracy (**0.748**) compared to other classifiers. By using **summary** function, the coefficient of each variable and intercept value can be obtained from the logistic regression model. **Decision Tree** can also be considered as an alternative due to its high accuracy **0.742** when cp is equal to **0.0001**. However, the picture of the tree can not be generated in my computer. Hence **Decision Tree** with cp **0.001** can be considered. It still has a high accuracy **0.737** and the tree picture can be generated. For **Naïve Bayes** and **KNN**, the accuracy is below **Logistic Regression** and **Decision Tree**. Moreover, **KNN** requires a lot of time for the code to be fully executed.

IV. Scope of Inference

1. Population Characteristics:

   The study encompasses individuals with a balanced distribution of havibg diabetes and not. A number of health indicators, including blood pressure, cholesterol, BMI, smoking habits, history of stroke, heart disease, or attack, physical activity, dietary choices, alcohol intake, access to healthcare, and sociodemographic characteristics, may vary among participants.

2. Health-Related Inference:

   Conclusions about the relationship between diabetes status and health indicators—such as high blood pressure (HighBP), high cholesterol (HighChol), frequency of cholesterol checks (CholCheck), BMI, smoking habits, history of stroke, and history of heart disease or heart attack—can be drawn from the dataset.

3. Lifestyle and Behavioural Inference:

   It is possible to investigate the connections between diabetes and lifestyle elements like exercise, eating a lot of fruits and vegetables, drinking a lot of alcohol, and having access to healthcare.

4. Demographic Inference:

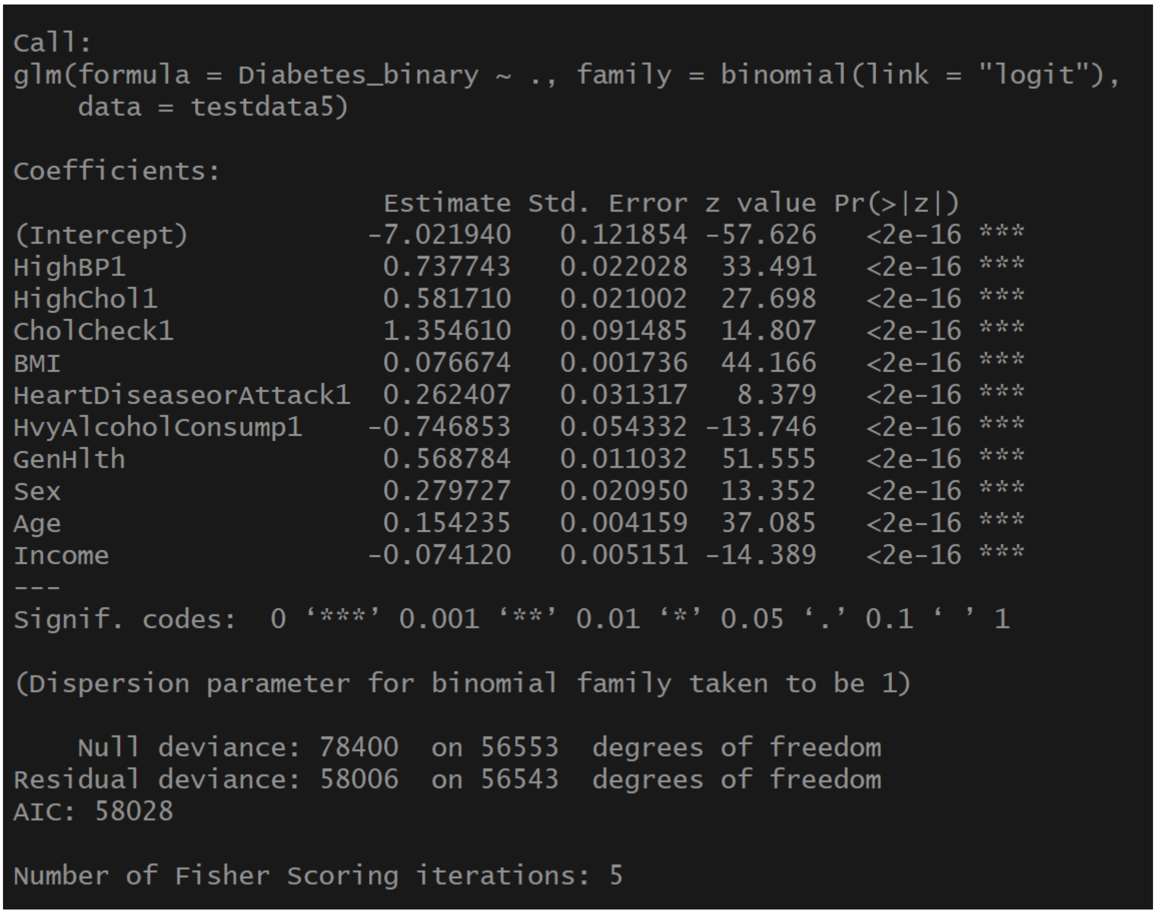
   Demographic variables such as sex, age, education, and income are included, providing an opportunity to examine how these factors relate to diabetes prevalence.

5. Limitations:

   There might be other variables that contribute toward diabetes. If these variables were added to the data, the choice of classifiers might change. Also, the proportion of number of people having diabetes and not should not be 50:50. The proportion of Diabetes people should be much lower than that.

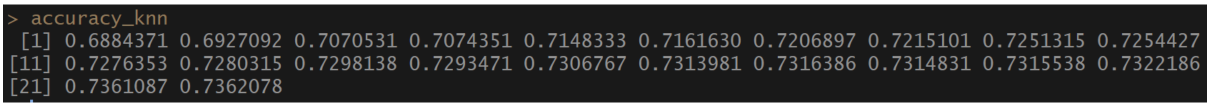
APPENDIX

1. **Summary of one of the logistic regression models.**



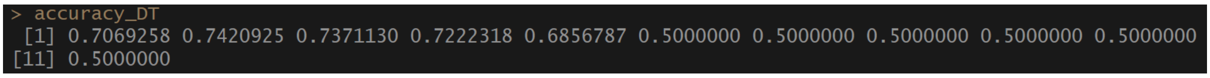
```
Call:
glm(formula = Diabetes_binary ~ ., family = binomial(link = "logit"), 
    data = testdata5)

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)    
(Intercept)           -7.021940   0.121854 -57.626   <2e-16 ***
HighBP1                0.737743   0.022028  33.491   <2e-16 ***
HighChol1              0.581710   0.021002  27.698   <2e-16 ***
CholCheck1             1.354610   0.091485  14.807   <2e-16 ***
BMI                    0.076674   0.001736  44.166   <2e-16 ***
HeartDiseaseorAttack1  0.262407   0.031317   8.379   <2e-16 ***
HvyAlcoholConsump1    -0.746853   0.054332 -13.746   <2e-16 ***
GenHlth                0.568784   0.011032  51.555   <2e-16 ***
Sex                    0.279727   0.020950  13.352   <2e-16 ***
Age                    0.154235   0.004159  37.085   <2e-16 ***
Income                -0.074120   0.005151 -14.389   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 78400  on 56553  degrees of freedom
Residual deviance: 58006  on 56543  degrees of freedom
AIC: 58028

Number of Fisher Scoring iterations: 5
```

2. **The accuracy obtained by using KNN.**



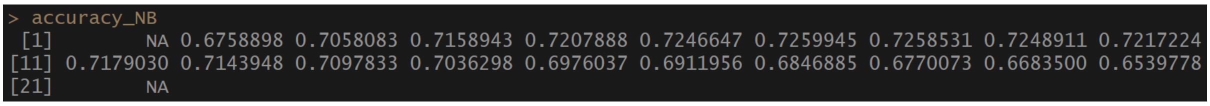
```
> accuracy_knn
 [1] 0.6884371 0.6927092 0.7070531 0.7074351 0.7148333 0.7161630 0.7206897 0.7215101 0.7251315 0.7254427
[11] 0.7276353 0.7280315 0.7298138 0.7293471 0.7306767 0.7313981 0.7316386 0.7314831 0.7315538 0.7322186
[21] 0.7361087 0.7362078
```

3. **The accuracy obtained using Decision Tree.**



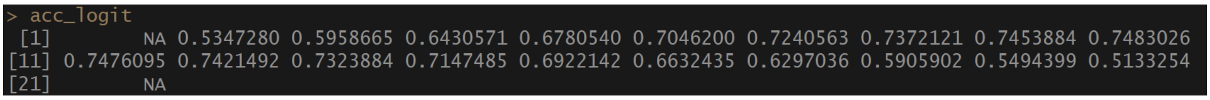
```
> accuracy_DT
 [1] 0.7069258 0.7420925 0.7371130 0.7222318 0.6856787 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000
[11] 0.5000000
```

4. **The accuracy obtained using Naïve Bayes.**



```
> accuracy_NB
 [1]        NA 0.6758898 0.7058083 0.7158943 0.7207888 0.7246647 0.7259945 0.7258531 0.7248911 0.7217224
[11] 0.7179030 0.7143948 0.7097833 0.7036298 0.6976037 0.6911956 0.6846885 0.6770073 0.6683500 0.6539778
[21]        NA
```

5. **The accuracy obtained using Logistic Regression.**



```
> acc_logit
 [1]        NA 0.5347280 0.5958665 0.6430571 0.6780540 0.7046200 0.7240563 0.7372121 0.7453884 0.7483026
[11] 0.7476095 0.7421492 0.7323884 0.7147485 0.6922142 0.6632435 0.6297036 0.5905902 0.5494399 0.5133254
[21]        NA
```