Economics 412 Group Project 1

Alexander Ramos (ID:605657325) Aneri Patel (ID:305642991)
Anshika Sharma (ID:305488635) Cristian Martinez (ID:205642760)

2021/04/30

Contents

I. Introduction

With the Coronavirus pandemic still ongoing throughout most of the world, health research has stopped for many other diseases with the news consistently focused on the Coronavirus. Stress, anxiety, depression, and other mental instabilities began to increase. Based on these behavior triggers, many partook in activities to

pass the time, but also partook in activities involving drug and alcohol abuse. With this in mind, one cannot help, but be observant on what may result from the aftermath of the pandemic. However, focus should transition back to addressing other health issues that are still occurring irrespective of the Coronavirus pandemic. According to the World Health Organization (WHO) strokes are the 2nd leading cause of death globally and accounts for approximately 11% of total deaths. With this information in mind, we will focus our analysis on strokes, particularly on stroke prediction analysis.

The purpose of this project is to use a machine learning classifier model to determine what the probability is of sustaining a stroke based on a number of independent predictors. The classification model that will be used is the Naïve Bayes model to assist us in calculating the probability of sustaining a stroke based on conditioning for other metrics like age, gender, history of smoking, and etc. What follows next is the data description of our dataset, the methodology of the statistical machine learning model that was used, the results derived, and conclusion of the analysis.

II. Data Description

Data Description

For the purpose of this project, a cross-sectional dataset was sourced from Kaggle. The data contains information on 3246 patients. The dependent variable is, as expected, an indicator variable, for stroke predictions. There are 10 independent variables, both categorical and continuous, which cover a patient's demographic information, their medical histories and factors like smoking habits which may contribute to a heart stroke.

Further descriptions of each variable are given below, followed by a summary table that showcases the descriptive statistics of each variable:

Data Directory:

- **id**: unique identifier code for each patient
- **gender**: classifies patients into "Male", "Female" or "Other"
- **age**: age of the patient.
- **hypertension**: This dummy variable takes the value 0 if the patient doesn't have hypertension, 1 if the patient has hypertension.
- **heart_disease**: This dummy variable takes the value 0 if the patient doesn't have any heart diseases, 1 if the - patient has a heart disease.
- **ever_married**: Dummy variable for whether an individual is married or not. "Yes" if the individual is married; "No" if the individual is not married.
- **work_type**: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
- **Residence_type**: This variable states whether the person resides in a "Rural" or "Urban" area.
- **avg_glucose_level**: This variable tells the average glucose level in blood.
- **bmi**: body mass index: Body mass index is a value derived from the mass and height of a person. The BMI is defined as the body mass divided by the square of the body height, and is expressed in units of kg/m^2, resulting from mass in kilograms and height in metres. (Wikipedia)
- **smoking_status**: This variable tells the smoking status of an individual. It takes the values: "formerly smoked", "never smoked", "smokes" or "Unknown". ("Unknown" in smoking_status means that the information is unavailable for this patient).
- **stroke**:This is the dependent variable for our analysis. Dummy variable for whether an individual gets a stroke or not. 1 if the patient had a stroke or 0 if not.

III. Methodology

The aim of this project was to build a supervised machine learning algorithm that predicts whether an individual would have a stroke or not. A Naive Bayes algorithm was used for training the model. Naive Bayes classifiers are collection algorithms based on Bayes' Theorem who share a common principle- every pair of features being classified is independent of each other. The Naive Bayes algorithm assumes that the data has features that are independent of each other. If this assumption holds true, such algorithms are considered to be simple, but also fast, accurate and reliable.

For this purpose, the 11 variables listed in the data description above were chosen. The variables that were already classified as factor variables in the data set include: gender, ever_married, work_type, residence_type, smoking status. Certain variables including age, hypertension, heart_disease, avg_glucose, stroke, bmi were converted into factor variables. The data set was then split into training (60%) and validation (40%) sets. Naive Bayes can also be used with continuous features but is more suited to categorical variables. And hence, the continuous variables were converted into categorical variables.

The naiveBayes function from the e1071 library in R, was used on the independent variable to compute a-posterior probabilities of the categorical variables using Baye's rule. And the predict function was used from the stats library in R to make predictions based on the results of the fitted model.

Accuracy metrics: Confusion matrix and Lift chart were used as accuracy metrics for our analysis and to evaluate the performance of our model over the testing/validation data set. The confusion matrix was calculated for the training and validation data split.

IV. Results & Analysis

1. Read Data

Our data is a 3246x11 matrix, and it is a large enough dataset to be considered big data. The above six entries are the very first six entries in our dataframe.

2. Checking data to determine if NAs exist

```
## [1] 0
```

```
## tibble [3,426 x 11] (S3: tbl_df/tbl/data.frame)
##  $ gender           : chr [1:3426] "Male" "Male" "Female" "Female" ...
##  $ age              : num [1:3426] 67 80 49 79 81 74 69 81 61 54 ...
##  $ hypertension     : num [1:3426] 0 0 0 1 0 1 0 1 0 0 ...
##  $ heart_disease    : num [1:3426] 1 1 0 0 0 1 0 0 1 0 ...
##  $ ever_married     : chr [1:3426] "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr [1:3426] "Private" "Private" "Private" "Self-employed" ...
##  $ Residence_type   : chr [1:3426] "Urban" "Rural" "Urban" "Rural" ...
##  $ avg_glucose_level: num [1:3426] 229 106 171 174 186 ...
##  $ bmi              : num [1:3426] 36.6 32.5 34.4 24 29 27.4 22.8 29.7 36.8 27.3 ...
##  $ smoking_status   : chr [1:3426] "formerly smoked" "never smoked" "smokes" "never smoked" ...
##  $ stroke           : num [1:3426] 1 1 1 1 1 1 1 1 1 1 ...
```

We confirmed that there exists no NAs in our dataset.

3. Convert all Numerical variables to categorical

We converted six variables into categorical variables. The converted variables are as follows:

- age
- avg_glcouse_level
- hypertension
- heart_disease
- stroke
- bmi

However, for the variables age, ave_glucose_level, and bmi, we identified too many levels that make our analysis hardly readable. Hence, we set appropriate bins for the variables to depict a clear trend in each class. For the variable age, we grouped the level by every ten years. This helps to identify in what age group shows the high probability of getting a stroke as well as the gradual increase in the risk of getting a stroke in the age group. Similarly, we applied bins to bmi and ave_glucose_level. To illustrate the risk of having a stroke at different levels of bmi and ave_glucose_level, we have set the bins according to the grouping definition from the CDC and WHO.

4. Confirms that all variables are now categorical

```
## tibble [3,426 x 11] (S3: tbl_df/tbl/data.frame)
##  $ gender           : chr [1:3426] "Male" "Male" "Female" "Female" ...
##  $ age              : chr [1:3426] "61-70" "71 and above" "41-50" "71 and above" ...
##  $ hypertension     : Factor w/ 2 levels "0","1": 1 1 1 2 1 2 1 2 1 1 ...
##  $ heart_disease    : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 1 1 2 1 ...
##  $ ever_married     : chr [1:3426] "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr [1:3426] "Private" "Private" "Private" "Self-employed" ...
##  $ Residence_type   : chr [1:3426] "Urban" "Rural" "Urban" "Rural" ...
##  $ avg_glucose_level: chr [1:3426] "Abnormal" "Normal" "Abnormal" "Abnormal" ...
##  $ bmi              : chr [1:3426] "Obese" "Obese" "Obese" "Healthy" ...
##  $ smoking_status   : chr [1:3426] "formerly smoked" "never smoked" "smokes" "never smoked" ...
##  $ stroke           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

Of the above variables, hypertension, heart_disease, and stroke are converted into an indicator variable that takes a value of either 0 or 1. Other variables are sorted by bins to represent a specific group, as explained in the previous section.

By using the data that we wrangled, we will expand our analysis.

A. Use all the features to construct a classification model

5. Creating Testing/Validating dataset

We split the dataset into two different sets, the training set, and the testing set. The training test will be used for building our model components that predict whether a person with specific characteristics and conditions will suffer from a stroke.

6. Run naive Bayes

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.94549878 0.05450122
##
## Conditional probabilities:
##    gender
## Y        Female        Male       Other
##   0 0.601132270 0.398353062 0.000514668
##   1 0.544642857 0.455357143 0.000000000
##
##    age
## Y        10-20      21-30      31-40      41-50      51-60      61-70
##   0 0.08337622 0.13021101 0.16057643 0.17807514 0.18528049 0.13329902
##   1 0.00000000 0.00000000 0.03571429 0.06250000 0.19642857 0.19642857
##    age
## Y   71 and above
##   0   0.12918168
##   1   0.50892857
##
##    hypertension
## Y           0         1
##   0 0.9016984 0.0983016
##   1 0.6517857 0.3482143
##
##    heart_disease
## Y           0          1
##   0 0.95110654 0.04889346
##   1 0.80357143 0.19642857
##
##    ever_married
## Y           No        Yes
##   0 0.24909933 0.75090067
##   1 0.09821429 0.90178571
##
##    work_type
## Y      children    Govt_job Never_worked     Private Self-employed
##   0 0.020586722 0.145136387  0.004632012 0.652599074   0.177045805
##   1 0.000000000 0.151785714  0.000000000 0.607142857   0.241071429
##
##    Residence_type
## Y       Rural     Urban
##   0 0.5033453 0.4966547
##   1 0.5000000 0.5000000
##
##    avg_glucose_level
```

```
## Y    Abnormal    Normal
##   0 0.1955739 0.8044261
##   1 0.4642857 0.5357143
##
##    bmi
## Y      Healthy      Obese Overweight Underweight
##   0 0.21413829 0.44014448 0.33126935  0.01444788
##   1 0.15178571 0.46428571 0.38392857  0.00000000
##
##    smoking_status
## Y   formerly smoked never smoked    smokes
##   0       0.2269686    0.5517241 0.2213073
##   1       0.3571429    0.4821429 0.1607143
```

Our learning algorithm, Naive Bayes, is on the training set from part 5. The result exhibits the relationship between each variable and the probability of having a stroke. Hence, we will discuss the noteworthy findings for each feature against our dependent variable, stroke, in our training set.

General facts about the training sample:

- Only 5.45% of the people in our training set had a stroke.
- Of the ones who had a stroke, 45.5% are male, whereas 39.8% are male in the group of people who did not have a stroke.

From the above facts, we can say that males tend to yield a higher chance of getting a stroke than females.

age:

Observing stroke from people under the age of 40 is extremely rare as the probability of getting a stroke for those particular groups is nearly 0% in our training set. Of those who had a stroke, only 3.5% of them are from the age group of the '30s. For the '40s, 6.3% of the people who had a stroke are from the age group of 41 years old to 50 years old. The age group of 51-60 and 61 to 70 accounts for nearly 40% of those who had a stroke. Lastly, we found out that 50% of the stroke cases are from 71 and above.

Therefore, it is reasonable to say that aging is one of the critical factors that increase the risk of having a stroke as age goes up.

hypertension:

In our training set, the proportion of the people who have hypertension is 9.8% among people who did not have a stroke in life. Meanwhile, 34.8 % of the people who had a stroke in life have hypertension. Therefore, hypertension increases the risk of having a stroke drastically.

heart_disease:

In our training set, the proportion of the people who have heart disease is 4.9% among people who did not have a stroke in life. Meanwhile, 19.6% of the people who had a stroke in life have heart disease. Therefore, heart disease increases the risk of having stroke drastically.

ever_married:

In our training set, the proportion of the people who have ever married is 75.1% among people who did not have a stroke in life. Meanwhile, 90.2% of the people who had a stroke in life have married at least once in life. Therefore, indirectly we can see that getting married increases the risk of having a stroke.

work_type:

In our training set, roughly 65.3% of the people are working in the private sector, and we also see about 17.7% of the people are working in the self-employed sector. While those two sectors account for nearly 83%

of all workforce, the proportion of people with stroke is also 85% from those sectors. Except for those who are working in childcare and never worked, we see that the similar proportion to the risk of having a stroke and the proportion to the corresponding job occupations among the people in our training set. Interestingly, the number of stroke cases observed from those who are in childcare and never worked account for almost 0%

Residence_type:

In our training set, the proportion of the people who live in an urban area is 49.7% among the group of people who did not have a stroke in life. Meanwhile, 50.0% of the people who had a stroke in life live in an urban area. Therefore, living in an urban area increases the risk of having a stroke by a little in our training set.

avg_glcouse_level:

WHO defines the normal average glucose level to be in the range of 55.12 to 126 mg/dL. While we divided the group of people into two groups, abnormal and normal, we identified that about 80% of the people have the average glucose level in the normal range for those who did not have a stroke. However, for those who had a stroke, only 53% of them are in the range of the normal average glucose level. It indicates that the high average glucose level is associated with a higher probability of having a stroke in life than those in the normal range.

bmi:

We identified four groups of people according to the definition of bmi from CDC. The underweight people yields about 0% in the group of people who had a stroke. In contrast, the people under obese accounts for nearly 46.2% of the entire stroke cases. Following the obese group, overweight people occupy about 38.4%, and healthy people yield 15.2% of the stroke cases in our training set. Hence, the increase in weight has a positive effect on the risk of having a stroke in life.

smoking_status:

From our training set, the proportion of the people who do not smoke is 55.2% in the group of people who did not have a stroke, whereas it is 48.2% for the people who had a stroke. Also, the data shows that the person who formerly smoked and smoke does indicate a higher proportion in the group of the people who had a stroke than did not have a stroke.

7. Examine the relationship between heart disease and stroke

```
##
##               0          1
##   0 0.95110654 0.04889346
##   1 0.80357143 0.19642857
```

The above result indicates that about 19.7% of the people who have had a stroke also had heart disease. Of those who did not have a stroke, only about 4.9% of people have heart disease. Hence, having heart disease increases the chance of getting a stroke by far. We wanted to observe how conditioning for heart disease would change the probability of developing a stroke because we collectively felt that heart disease could be a strong indicator of sustaining a stroke.

8. Training and confusion matrix

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
```

```
##          0 1882   87
##          1   61   25
##
##                Accuracy : 0.928
##                  95% CI : (0.9159, 0.9388)
##     No Information Rate : 0.9455
##     P-Value [Acc > NIR] : 0.99966
##
##                   Kappa : 0.2154
##
##  Mcnemar's Test P-Value : 0.03988
##
##             Sensitivity : 0.9686
##             Specificity : 0.2232
##          Pos Pred Value : 0.9558
##          Neg Pred Value : 0.2907
##              Prevalence : 0.9455
##          Detection Rate : 0.9158
##    Detection Prevalence : 0.9582
##       Balanced Accuracy : 0.5959
##
##        'Positive' Class : 0
##
```

The statistics from the confusion matrix show that the model accuracy is 92.8% for the training dataset. Kappa is around 0.22, and it indicates that the samples used in the model are substantially representative of the variables measured. The low kappa is the result of applying bins to some of our variables.

The value of sensitivity, 0.97, tells us how much of the actual stroke was detected by the model's prediction over the training sample set. The value of specificity, 0.22, shows that how much the model correctly predicted the true negative. The probability of detecting a true positive is around 97%. Hence, our model's predictive accuracy is very satisfactory for the training set to correctly classifying whether a person had a stroke or not.

9. Validation and confusion matrix

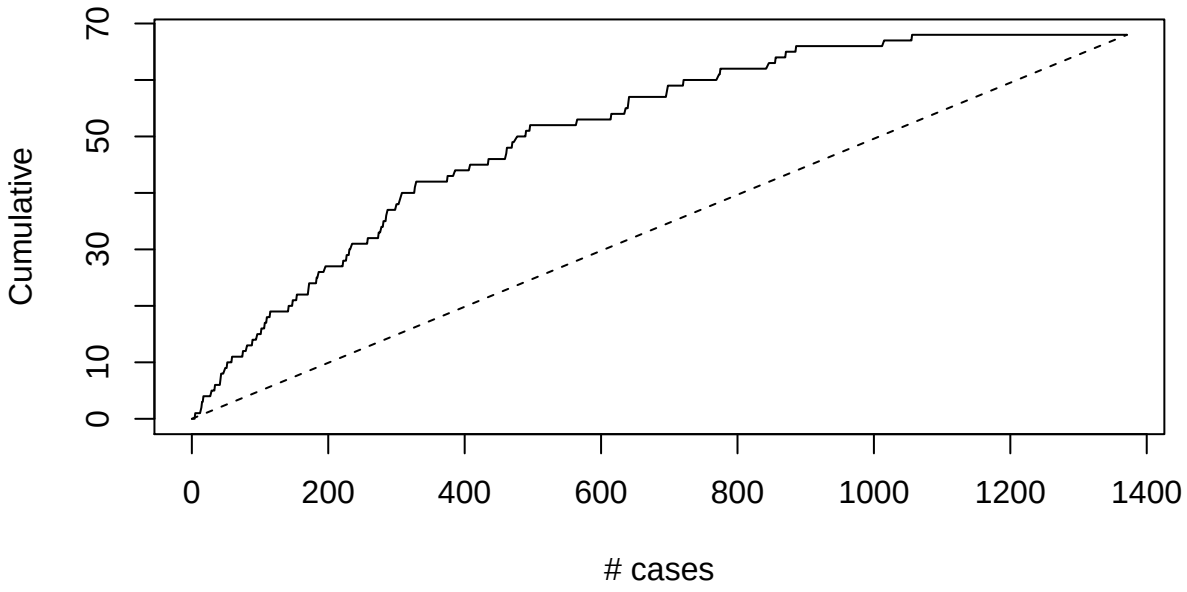
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1244   57
##          1   59   11
##
##                Accuracy : 0.9154
##                  95% CI : (0.8994, 0.9296)
##     No Information Rate : 0.9504
##     P-Value [Acc > NIR] : 1.000
##
##                   Kappa : 0.1149
##
##  Mcnemar's Test P-Value : 0.926
##
##             Sensitivity : 0.9547
##             Specificity : 0.1618
```

```
##          Pos Pred Value : 0.9562
##          Neg Pred Value : 0.1571
##              Prevalence : 0.9504
##          Detection Rate : 0.9074
##    Detection Prevalence : 0.9489
##       Balanced Accuracy : 0.5582
##
##        'Positive' Class : 0
##
```

The statistics from the confusion matrix show that the model accuracy is 91.5% for the testing dataset. Kappa is around 0.11, and it indicates that the samples used in the model are substantially representative of the variables measured. The low kappa is the result of applying bins to some of our variables.

The value of sensitivity, 0.95, tells us how much of the actual stroke was detected by the model's prediction over the testing sample set. The value of specificity, 0.16, shows that how much the model correctly predicted the true negative. The probability of detecting a true positive is around 95%. Hence, our model's predictive accuracy is very satisfactory for the testing set to correctly classifying whether a person had a stroke or not.

10. Plot the Lift Chart



The plot exhibits a lift curve. Lift represents the amount of information gained by using a machine learning model instead of randomly guessing. As our gain curve is above the baseline, we can say that our machine learning algorithm is more efficient than a random model. In other words, there exists a positive relationship between the dependent variable and the features.

B. Case study: dropping gender, ever married, and work type in our model

We conducted exactly the same analysis process we did in part a. Therefore, we will briefly explain what is significant in this case study.

1. Run Native Bayes

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.95231144 0.04768856
##
## Conditional probabilities:
##    age
## Y        10-20      21-30      31-40      41-50      51-60      61-70
##   0 0.07971385 0.13438937 0.15074093 0.16760347 0.19621870 0.13132345
##   1 0.00000000 0.00000000 0.02040816 0.07142857 0.22448980 0.23469388
##    age
## Y   71 and above
##   0   0.14001022
##   1   0.44897959
##
##    hypertension
## Y           0         1
##   0 0.9075115 0.0924885
##   1 0.7857143 0.2142857
##
##    heart_disease
## Y            0          1
##   0 0.94890138 0.05109862
##   1 0.76530612 0.23469388
##
##    Residence_type
## Y       Rural     Urban
##   0 0.4941237 0.5058763
##   1 0.4795918 0.5204082
##
##    avg_glucose_level
## Y    Abnormal    Normal
##   0 0.1982626 0.8017374
##   1 0.4183673 0.5816327
##
##    bmi
## Y      Healthy      Obese Overweight Underweight
##   0 0.21333333 0.45435897 0.32153846  0.01076923
##   1 0.12244898 0.53061224 0.34693878  0.00000000
##
```

```
##    smoking_status
## Y   formerly smoked never smoked    smokes
##   0       0.2324987    0.5436893 0.2238120
##   1       0.3979592    0.3877551 0.2142857
```

The tendency identified in part a holds for each feature from the native Bayes probabilistic outcome.

2. Training and confusion matrix

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1926   84
##          1   31   14
##
##                Accuracy : 0.944
##                  95% CI : (0.9332, 0.9536)
##     No Information Rate : 0.9523
##     P-Value [Acc > NIR] : 0.9624
##
##                   Kappa : 0.1709
##
##  Mcnemar's Test P-Value : 1.241e-06
##
##             Sensitivity : 0.9842
##             Specificity : 0.1429
##          Pos Pred Value : 0.9582
##          Neg Pred Value : 0.3111
##              Prevalence : 0.9523
##          Detection Rate : 0.9372
##    Detection Prevalence : 0.9781
##       Balanced Accuracy : 0.5635
##
##        'Positive' Class : 0
##
```

For the testing dataset, while we dropped three features than the model in part a, the statistics from the confusion matrix show that the model accuracy is 94.4% for the training dataset. Kappa is around 0.17, and it indicates that the samples used in the model are substantially representative of the variables measured. The low kappa is the result of applying bins to some of our variables.

The value of sensitivity, 0.98, tells us how much of the actual stroke was detected by the model's prediction over the training sample set. The value of specificity, 0.14, shows that how much the model correctly predicted the true negative. The probability of detecting a true positive is around 98%. Hence, our model's predictive accuracy is very satisfactory for the training set to correctly classifying whether a person had a stroke or not. Also, we can say that the mode is performing better than the one from part a.

3. Validation and confusion matrix

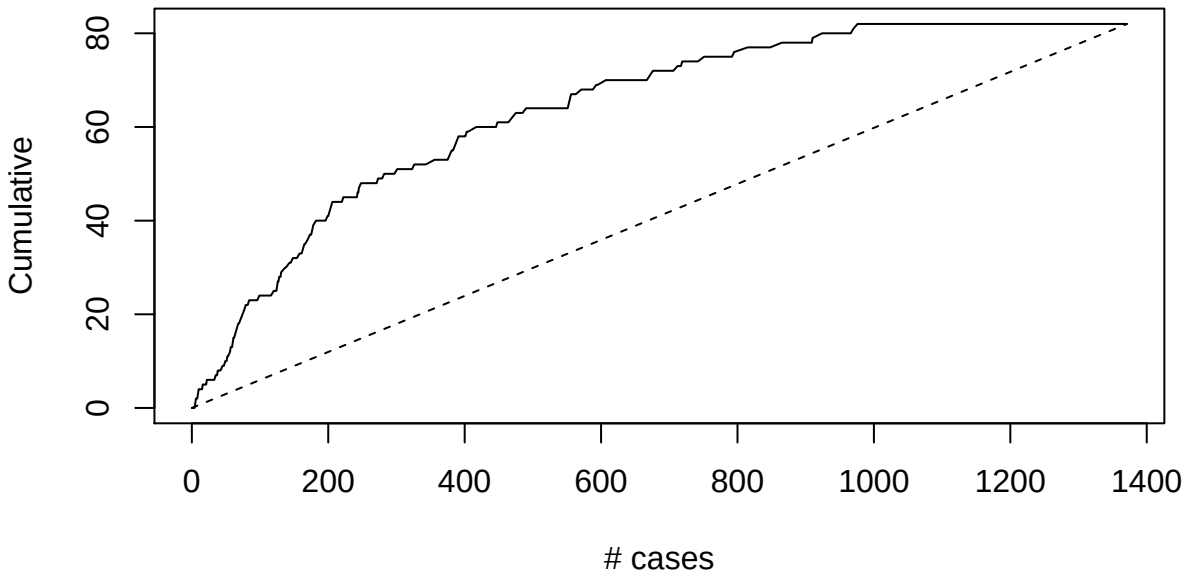
```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction    0    1
##          0 1256   74
##          1   33    8
##
##                Accuracy : 0.922
##                  95% CI : (0.9065, 0.9356)
##     No Information Rate : 0.9402
##     P-Value [Acc > NIR] : 0.9974051
##
##                   Kappa : 0.094
##
##  Mcnemar's Test P-Value : 0.0001102
##
##             Sensitivity : 0.97440
##             Specificity : 0.09756
##          Pos Pred Value : 0.94436
##          Neg Pred Value : 0.19512
##              Prevalence : 0.94019
##          Detection Rate : 0.91612
##    Detection Prevalence : 0.97009
##       Balanced Accuracy : 0.53598
##
##        'Positive' Class : 0
##
```

For the testing dataset, while we dropped three features than the model in part a, the statistics from the confusion matrix show that the model accuracy is 92.2% for the testing dataset. Kappa is around 0.09, and it indicates that the samples used in the model are substantially representative of the variables measured. The low kappa is the result of applying bins to some of our variables.

The value of sensitivity, 0.97, tells us how much of the actual stroke was detected by the model's prediction over the testing sample set. The value of specificity, 0.09, shows that how much the model correctly predicted the true negative. The probability of detecting a true positive is around 97%. Hence, our model's predictive accuracy is very satisfactory for the testing set to correctly classifying whether a person had a stroke or not. This result is similar to what we found out in part a.

4. Plot the Lift Chart



As our gain curve is above the baseline, we can say that our machine learning algorithm is more efficient than a random model. In other words, there exists a positive relationship between the dependent variable and the features. However, the gain chart is less bending outwards, and it tells that the model from part a is more efficient in gaining information. Namely, we can say that the model from part a is superior to the model from part b in terms of learning efficiency.

C. Case study: dropping gender, ever married, work type, bmi, and smoking status in our model

We conducted exactly the same analysis process we did in parts A and B, but also drop additional predictors. Therefore, we will briefly explain what is significant in this case study.

1. Run Naive Bayes

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.94209246 0.05790754
##
```

```
## Conditional probabilities:
##    age
## Y        10-20      21-30      31-40      41-50      51-60      61-70
##   0 0.08626033 0.13171488 0.14669421 0.17768595 0.19421488 0.13481405
##   1 0.00000000 0.00000000 0.02521008 0.08403361 0.17647059 0.21008403
##    age
## Y   71 and above
##   0   0.12861570
##   1   0.50420168
##
##    hypertension
## Y           0         1
##   0 0.8899793 0.1100207
##   1 0.6890756 0.3109244
##
##    heart_disease
## Y            0          1
##   0 0.94886364 0.05113636
##   1 0.80672269 0.19327731
##
##    Residence_type
## Y       Rural     Urban
##   0 0.5030992 0.4969008
##   1 0.4537815 0.5462185
##
##    avg_glucose_level
## Y    Abnormal    Normal
##   0 0.1905992 0.8094008
##   1 0.4453782 0.5546218
```

The tendency identified in parts A and B holds for each feature from the Native Bayes probabilistic outcome.

2. Training and confusion matrix

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1884   97
##          1   52   22
##
##                Accuracy : 0.9275
##                  95% CI : (0.9154, 0.9383)
##     No Information Rate : 0.9421
##     P-Value [Acc > NIR] : 0.9973599
##
##                   Kappa : 0.1921
##
##  Mcnemar's Test P-Value : 0.0003126
##
##             Sensitivity : 0.9731
##             Specificity : 0.1849
##          Pos Pred Value : 0.9510
```

```
##          Neg Pred Value : 0.2973
##              Prevalence : 0.9421
##          Detection Rate : 0.9168
##    Detection Prevalence : 0.9640
##       Balanced Accuracy : 0.5790
##
##        'Positive' Class : 0
##
```

For the training dataset, while we dropped five features than the model in part a, the statistics from the confusion matrix show that the model accuracy is 92.8% for the training dataset. Kappa is around 0.19, and it indicates that the samples used in the model are substantially representative of the variables measured. The low kappa is the result of applying bins to some of our variables.

The value of sensitivity, 0.97, tells us how much of the actual stroke was detected by the model's prediction over the training sample set. The value of specificity, 0.18, shows that how much the model correctly predicted the true negative. The probability of detecting a true positive is around 97%. Hence, our model's predictive accuracy is very satisfactory for the training set to correctly classifying whether a person had a stroke or not.

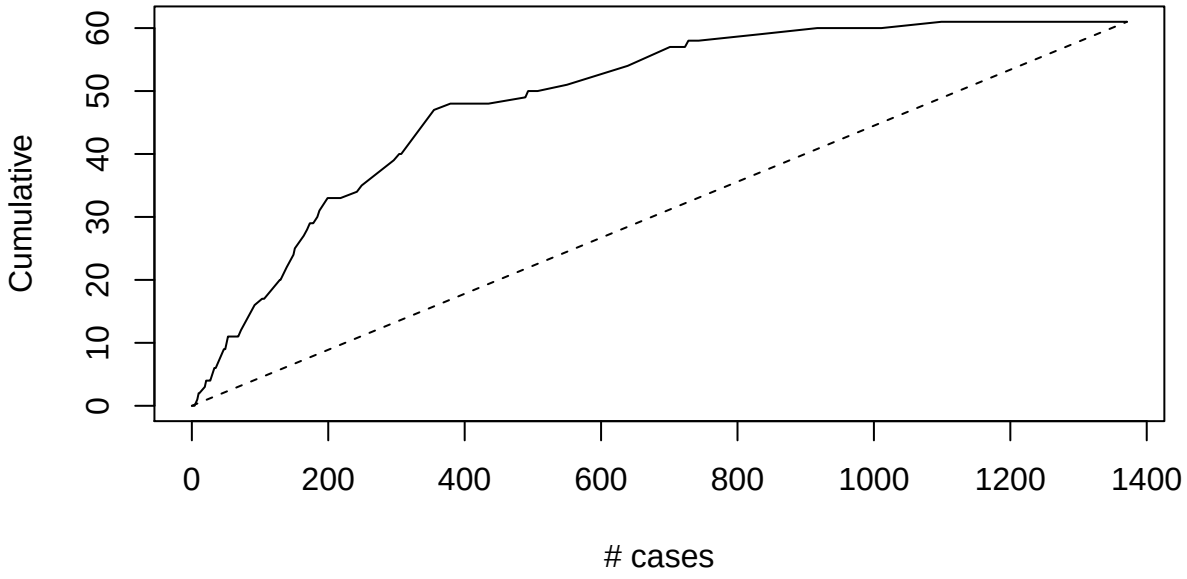
3. Validation and confusion matrix

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1272   52
##          1   38    9
##
##                Accuracy : 0.9344
##                  95% CI : (0.9199, 0.9469)
##     No Information Rate : 0.9555
##     P-Value [Acc > NIR] : 0.9999
##
##                   Kappa : 0.1331
##
##  Mcnemar's Test P-Value : 0.1706
##
##             Sensitivity : 0.9710
##             Specificity : 0.1475
##          Pos Pred Value : 0.9607
##          Neg Pred Value : 0.1915
##              Prevalence : 0.9555
##          Detection Rate : 0.9278
##    Detection Prevalence : 0.9657
##       Balanced Accuracy : 0.5593
##
##        'Positive' Class : 0
##
```

For the testing dataset, while we dropped five features than the model in part a, the statistics from the confusion matrix show that the model accuracy is 93.4% for the testing dataset. Kappa is around 0.13, and it indicates that the samples used in the model are substantially representative of the variables measured. The low kappa is the result of applying bins to some of our variables.

The value of sensitivity, 0.97, tells us how much of the actual stroke was detected by the model's prediction over the testing sample set. The value of specificity, 0.14, shows that how much the model correctly predicted the true negative. The probability of detecting a true positive is around 97%. Hence, our model's predictive accuracy is very satisfactory for the testing set to correctly classifying whether a person had a stroke or not. This result is similar to what we found out in part a.

4. Plot the Lift Chart



As our gain curve is above the baseline, we can say that our machine learning algorithm is more efficient than a random model. In other words, there exists a positive relationship between the dependent variable and the features. However, the gain chart is less bending outwards, and it tells that the model from part a is more efficient in gaining information. Namely, we can say that part c is superior to the model from parts a and b in terms of learning efficiency.

V. Conclusion

In conclusion, after performing the NaiveBayes classification model on our data and segmenting age, BMI, and average glucose levels based on what the CDC classifies as healthy and unhealthy ranges, we derived results from a model that we collectively agree is robust and highly accurate. To determine the accuracy of our model, we created three different scenarios where we removed independent indicators to determine if the model could still efficiently predict the probability based on the remaining independent predictors. We first use every independent predictor in our dataset to observe how the model performs in terms of accuracy and report both a training and validation accuracy metric. We derived a training and validation accuracy of 92.80% and 91.54% respectively for our first case study. We then alter the model by removing gender, if the individual was ever married, and their work type and find that the training and validation accuracies increase for both with the training set deriving an accuracy of 94.40% and validation accuracy of 92.20%. We performed one last study where we removed gender, if the individual was ever married, work type,

BMI, and their smoking status and found that the training accuracy was reported at 92.75% and validation was 93.44%. Collectively, we see that there is a trade-off in terms of how many independent predictors to use and believe that the more predictors that are used, the possibility of overfitting is present. We find that the last case derived the highest validation accuracy rate and the second case derived the highest training accuracy rate. Thus, we conclude that our model using the NaiveBayes algorithm is incredibly easy to perform yet generates strong results.

VI. Future Work

The human body is a complex system that till this day is still the subject of heavy research, with strokes being one of the more complex diseases to understand. Our analysis is assuming that our predictors are independent of each other, but it can be argued that some of the predictors may be correlated with one another. Perhaps data on genetics can be the key to bridge the lack of understanding on why individuals who are healthy sustain a stroke and for those who we classify as "unhealthy" and should likely sustain a stroke don't. By collecting data on individual genetics, perhaps we could get closer to understanding the reasons why strokes occur.

VII. R Code Source

1. Read Data

```
Data = read_excel("C:/Users/alexd/Desktop/412 GP1/Stroke_Updated.xlsx")
Data = na.omit(Data)
head(Data)
```

2. Checking data to determine if NAs exist

```
sum(is.na(Data)) #there are no NAs in data

describe(Data)
str(Data)
```

3. Convert all Numerical variables to categorical (Will change Age, BMI & Average glucose level)

```
Data$age <- as.numeric(Data$age)
Data$hypertension<- factor(Data$hypertension)
Data$heart_disease <- factor(Data$heart_disease)
Data$avg_glucose_level <- as.numeric(Data$avg_glucose_level)
Data$stroke <- factor(Data$stroke)
Data$bmi = as.numeric(Data$bmi)

# Creating group bins based on health standards (Recommended from CDC)
Data$avg_glucose_level = ifelse(Data$avg_glucose_level>=55.12 & Data$avg_glucose_level<=126,"Normal","Al
```

```
Data<- Data %>% mutate(bmi=case_when(
  bmi>=11.5 & bmi<=18.0 ~ "Underweight",
  bmi>=18.5 & bmi<=24.9 ~ "Healthy",
  bmi>=25.0 & bmi<=29.9 ~ "Overweight",
  bmi>=30 ~ "Obese"))

Data<- Data %>% mutate(age=case_when(
  age>=10 & age<=20 ~ "10-20",
  age>=21 & age<=30 ~ "21-30",
  age>=31 & age<=40 ~ "31-40",
  age>=41 & age<=50 ~ "41-50",
  age>=51 & age<=60 ~ "51-60",
  age>=61 & age<=70 ~ "61-70",
  age>=71 ~ "71 and above"))
```

4. Confirms that all variables are now categorical

```
str(Data) # Confirms that all variables are now categorical
```

a. Use all the features to construct a classification model

5. Creating Testing/Validating dataset

```
Selected.Var <- c(1,2,3,4,5,6,7,8,9,10,11) # Use every indepedent categorical variable
train.index <- sample(c(1:dim(Data)[1]), dim(Data)[1]*0.6)  # Splitting the dataset into 6:4
train.df <- Data[train.index, Selected.Var]
valid.df <- Data[-train.index, Selected.Var]
```

6. Run naive Bayes

```
Data.nb <- naiveBayes(train.df$stroke ~ ., data = train.df)
Data.nb
```

Our learning algorithm, Naive Bayes, is on the training set from part 5. The result exhibits the relationship between each variable and the probability of having a stroke. Hence, we will discuss the no

7. Examine the relationship between heart disease and stroke

```
prop.table(table(train.df$stroke, train.df$heart_disease), margin = 1)


pred.prob <- predict(Data.nb, newdata = valid.df, type = "raw")
```

```
pred.class <- predict(Data.nb, newdata = valid.df)

df <- data.frame(actual = valid.df$stroke, predicted = pred.class, pred.prob)
```

8. Training and confusion matrix

```
df[valid.df$work_type == "Private" & valid.df$bmi == 35 & valid.df$Residence_type == "Rural" &
     valid.df$heart_disease == 1 & valid.df$age == 35 & valid.df$gender == "Male" &
valid.df$ever_married == "No" & valid.df$avg_glucose_level == 200 & valid.df$smoking_status == "Yes"
& valid.df$hypertension == 1,]

# Training
pred.class <- predict(Data.nb, newdata = train.df)
confusionMatrix(pred.class, train.df$stroke)
```

9. Validation and confusion matrix

```
pred.class <- predict(Data.nb, newdata = valid.df)
confusionMatrix(pred.class, valid.df$stroke)
```

10. Plot the Lift Chart

```
gain <- gains(ifelse(valid.df$stroke=="1",1,0), pred.prob[,1], groups=2000)

# Plot the Lift Chart
plot(c(0,gain$cume.pct.of.total*sum(valid.df$stroke==1))~c(0,gain$cume.obs),
     xlab="# cases", ylab="Cumulative", main="", type="l")
lines(c(0,sum(valid.df$stroke== 1))~c(0, dim(valid.df)[1]), lty=2)
```

b. Case study: dropping gender, ever married, and work type in our model

1. Run Native Bayes

```
# Creating Testing/Validating dataset
Selected.Var <- c(2,3,4,7,8,9,10,11) # Use every variable expect for gender, ever married, work type
train.index <- sample(c(1:dim(Data)[1]), dim(Data)[1]*0.6)
train.df <- Data[train.index, Selected.Var]
valid.df <- Data[-train.index, Selected.Var]


# Run Naive Bayes
Data.nb <- naiveBayes(stroke ~ ., data = train.df)
Data.nb
```

2. Training and confusion matrix

```
pred.prob <- predict(Data.nb, newdata = valid.df, type = "raw")


pred.class <- predict(Data.nb, newdata = valid.df)

df <- data.frame(actual = valid.df$stroke, predicted = pred.class, pred.prob)


# Training
pred.class <- predict(Data.nb, newdata = train.df)
confusionMatrix(pred.class, train.df$stroke)
```

3. Validation and confusion matrix

```
# Validation
pred.class <- predict(Data.nb, newdata = valid.df)
confusionMatrix(pred.class, valid.df$stroke)
```

4. Plot the Lift Chart

```
gain <- gains(ifelse(valid.df$stroke=="1",1,0), pred.prob[,2], groups=1000)



# Plot the Lift Chart
plot(c(0,gain$cume.pct.of.total*sum(valid.df$stroke==1))~c(0,gain$cume.obs),
     xlab="# cases", ylab="Cumulative", main="", type="l")
lines(c(0,sum(valid.df$stroke== 1))~c(0, dim(valid.df)[1]), lty=2)
```

c. Case study: dropping gender, ever married, work type, bmi, and smoking status

1. Run Naive Bayes

```
# Creating Testing/Validating dataset
Selected.Var <- c(2,3,4,7,8,11) # Use every variable expect for gender, ever married, work type, bmi, a
train.index <- sample(c(1:dim(Data)[1]), dim(Data)[1]*0.6)
train.df <- Data[train.index, Selected.Var]
valid.df <- Data[-train.index, Selected.Var]

# Run Naive Bayes
Data.nb <- naiveBayes(stroke ~ ., data = train.df)
Data.nb
```

2. Training and confusion matrix

```
pred.prob <- predict(Data.nb, newdata = valid.df, type = "raw")
table(pred.prob)

pred.class <- predict(Data.nb, newdata = valid.df)

df <- data.frame(actual = valid.df$stroke, predicted = pred.class, pred.prob)


# Training
pred.class <- predict(Data.nb, newdata = train.df)
confusionMatrix(pred.class, train.df$stroke)
```

3. Validation and confusion matrix

```
# Validation
pred.class <- predict(Data.nb, newdata = valid.df)
confusionMatrix(pred.class, valid.df$stroke)
```

4. Plot the Lift Chart

```
gain <- gains(ifelse(valid.df$stroke=="1",1,0), pred.prob[,2], groups=1000)



# Plot the Lift Chart
plot(c(0,gain$cume.pct.of.total*sum(valid.df$stroke==1))~c(0,gain$cume.obs),
     xlab="# cases", ylab="Cumulative", main="", type="l")
lines(c(0,sum(valid.df$stroke== 1))~c(0, dim(valid.df)[1]), lty=2)
```

VIII. Reference

- Data source: https://www.kaggle.com/fedesoriano/stroke-prediction-dataset
- Center for Disease control (https://www.cdc.gov/)
- World Health Organization (https://www.who.int/)