MGT 6203 Group #30 Project Progress Report

**1. Abstract**

According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths. In this project, the “ [Stroke Prediction Dataset](https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/code?datasetId=1120859&searchQuery=R)” is used to predict whether a patient is likely to get stroke based on the input parameters like gender, age, various diseases, and smoking status. Each row in the data provides relevant information about the patient. The project is delivered following the below five main sections:

* Data Exploration and Analysis
* Classification and prediction using Support Vector Machine
* Classification and prediction using Logistic Regression
* Classification and prediction using Random Forest
* Classification and prediction using K-Nearest Neighbor

The objective questions that are aimed to be answered at the end of this project are:

1. Which predictor variables are most correlated with the rate of stroke?
2. Will marriage increase the risk of getting a stroke?
3. Does stroke have a strong correlation with other health indicator factors (such as smoking?)

The business justifications for the project are as follows:

The outcome of our analysis is expected to indicate that stroke is correlated with the health status of individuals, and the following markets will be impacted:

1. Health Insurance: Increasing the insurance premium for individuals with a high probability of stroke.
2. Advertisement: Strategy changes in targeting the individuals by the various markets such as health care providers, dietary supplement industry, etc.
3. Medical institutions: Improving more routine care for individuals who have a high chance of stroke and are susceptible

**2. Data Analysis**

**2.0. Data Exploration and Analysis**

The source of this dataset is a clean/refined subset of the original dataset which is based on the Electronic Health Record (EHR) controlled by McKinsey & Company.

Dataset given for stroke prediction contains target variable which is a binary categorical variable and the predictor categorical variables are gender, ever married, work type, residence type, heart disease, Hypertension, smoking status and the numerical predictor variables are age, average glucose level and BMI.

BMI variable has 201 missing or N/A values. Since BMI is an important variable, the missing values are replaced by the mean.

The dataset contains 5110 observations of which 2994 are male participants and 2115 are female participants and 1 other. The observation with other can be treated as an outlier from analysis as there is very minimal data or only one data for the category to include in analysis as this could affect the model performance.

Approximately 30% of the "smoking\_status" variable is recorded as "unknown" which is a significant portion. In order to reduce this value, the "smoking\_status" for the ages equal or below 7 were converted to "never smoked". This reduced the 30% "unknown"smoking\_status to 22%.

Table 1 shows a simple summary of the data exploration for each variable after the above changes. Based on Table 1, the following bullets should be considered during the analysis.

* The "stroke" == 1 comprises only 5% of the data and cost analysis is required for each model.
* The "smoking\_status" has 22% of "uknown" category data points that may or may not be used in each model.
* There is an imbalance between the Female and Male population.
* There is heteroscedasticity for "bmi" and "avg\_glucose\_level".
* Scaling is required when not using regression.

Table 1. Summary of the variables data insight

| Variable | Observation |
| --- | --- |
| "gender" | Imbalanced with 59% "Female" and 41% "Male". |
| “age” | There is enough data with an acceptable balance before 40 and after 40. |
| "hypertension" | - Approximately 90% "No" and 10% "Yes" - Significantly imbalanced |
| “heart\_disease” | 95% "No" and 5% "Yes" - Significantly imbalanced |
| "ever\_married" | Imbalanced with approximately 65% "Yes" and 35% "No". |
| "work\_type" | Children Govt\_job Never\_worked Private Self-employed  14% 13% <1% 57% 16% |
| “residence\_type” | 50% balanced between "rural" and "urban" |
| “avg\_glucose\_level” | Major heteroscedasticity can be observed from the histogram – Right skewed |
| “bmi” | Minor heteroscedasticity can be observed from the histogram – Right skewed |
| "smoking\_status" | formerly smoked never smoked smokes Unknown  25% 38% 15% 22% |
| “stroke” | 4.3% indicating stroke and 95.7% with no stroke occurance |

**[Note to TA]: The next step in “Data Exploration and Analysis” is to provide graphs and examine the relation between the variables. We will also be addressing heteroscedasticity for "bmi" and "avg\_glucose\_level" such as by keeping only 1 of the 2 predictor variables**

**2.1. Cross Validation Procedure**

We divided the complete data into:

| Percentage of Total Data | Data Purpose |
| --- | --- |
| 70% | Training Set |
| 15% | Validation Set |
| 15% | Test Set |

We set a seed number to ensure that all of the models are working with the same training, validation, and test set.

**[Note to TA]: Currently we only trained our data on the training set. We will attempt k-fold cross validation to train our data in the future.**

**2.2. Support Vector Machine**

1. Transformed variables from numeric(Stroke) and strings to factor.
2. Since the dataset is imbalanced and we need to use SMOTE method to prepare the training dataset
3. Using SMOTE to create 6000 rows of data and fit the SVM model

**summary of the model:**

The type of SVM is: Classification

The kernel used in this model is radial

Cost is 1

The support vector in this model is 2930 and it has two levels - 0 and 1

1. Predict the result using test dataset and build confusion matrix (not using validation dataset since it will be conducted when using cross-validation to create the model, however, result in less training dataset)
2. The result is

| Accuracy | 0.7453 |
| --- | --- |
| Sensitivity | 0.7479 |
| Specificity | 0.6981 |
| Balanced Accuracy | 0.7230 |
| Pos Pred Value | 0.9784 |
| Neg Pred Value | 0.1317 |
| Detection Rate | 0.7091 |

1. Tuning model and find the best value of gamma and cost combination (gamma = 1, cost =10)
2. Refine the model use new gamma and cost and predict again

| Accuracy | 0.904 |
| --- | --- |
| Sensitivity | 0.95145 |
| Specificity | 0.03774 |
| Balanced Accuracy | 0.49459 |
| Pos Pred Value | 0.94753 |
| Neg Pred Value | 0.04082 |
| Detection Rate | 0.90206 |

From above, we can see that after tuning parameters, the accuracy of model has been improved from 74.5% to 90.4%.

**[Note to TA]: The next step in “Support Vector Machine” is to 1) conduct feature engineering to remove some variables 2) use 10-fold cross-validation to train the svm model using train set and validation set 3)try different kernels and tune parameters to refine the model 3)conduct model visualization 4) finish the model evaluation, draw ROC and calculate AUC.**

**2.3. Logistic Regression**

1.Fit the training dataset to the logistic model

2.Use the logistic regression model (from training dataset) to predict results on the validation set and add the predicted probability into the validation dataframe; the prediction shows up as the probability of getting a stroke based on the predictor variables.

3.If probability of getting a stroke is > cutoff probability (i.e. 0.4), then classify as 1 (stroke), else 0 (no stroke). Add the prediction classification to the validation dataset

4.Compare the actual stroke to the predicted results to evaluate the logistic regression model by using a confusion matrix

|  | Reference | |
| --- | --- | --- |
| Prediction | No | Yes |
| No | 969 | 52 |
| Yes | 0 | 0 |

5. We tried a different cutoff probability (i.e. 0.3) to see if accuracy will increase

|  | Reference | |
| --- | --- | --- |
| Prediction | No | Yes |
| No | 959 | 51 |
| Yes | 10 | 1 |

6.Analysis of the result

When the cutoff is 0.4, all of the predicted classification was “no stroke,” leading to 52 cases of type 2 error false negative. As a result, the sensitivity (= true positive/ (true positive + false negative)) is 0 and specificity (=true negative/(true negative + )) is 1. Even though accuracy is high at 0.949, the high accuracy is due to the fact that most of the data are “no stroke.”

Our hypothesis was that the cutoff might be contributing to the model’s wrong prediction and we decreased the cutoff to 0.3 and reevaluated the model.

When the cutoff is 0.3, we see that the model prediction result includes both stroke and no-stroke cases. However, the accuracy actually decreased from 0.949 to 0.940, as now there are 51 cases of type 2 error false negative and 10 cases of type 1 error false positive, leading to 61 cases of incorrect prediction.

**[Note to TA]: Next step in logistic regression is to refine the logistic regression by a) removing variables and b) trying k-fold cross validation to train the model (instead of only using the training dataset)**

**2.4. Random Forest**

1. Added the column with stroke categorical variable
2. Create a random forest model with the training data set.
3. Predict stroke result using the model with the test data set.
4. The **Model Summary**:

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 3

**Confusion matrix:**

|  | Reference | |
| --- | --- | --- |
| Prediction | No | Yes |
| No | 2921 | 0 |
| Yes | 0 | 144 |

1. The **prediction** **result**

Confusion Matrix

|  | Reference | |
| --- | --- | --- |
| Prediction | No | Yes |
| No | 967 | 0 |
| Yes | 0 | 55 |

Accuracy : 1

95% CI : (0.9964, 1)

No Information Rate : 0.9462

P-Value [Acc > NIR] : < 2.2e-16

**[Note to TA]: The next step in “Random Forest” is**

1. **Current accuracy rate is 1. It seems to be too ideal. Will need to double check the result.**
2. **Will vary the number of trees and compare the accuracy.**
3. **Run with the validation data set.**

**2.5. K-Nearest Neighbor**

[To add high level KNN backgrounds and steps for model creation and comparison…etc as an overview.]

To run the KNN model, we need to create dummy variables for categorical variables on “gender”, “hypertension”, “heart\_disease”,”ever\_married”,”Residence\_type”,”work\_type” and “smoking\_status”.

Then, create a function “check\_accuracy” including *kknn* function in order to check each row of data points under different K values. So firstly to set K between 1 and 20 then the results shown in below table2.

As the table 2 shows, the key K value we can find is to use K>=5 which has significant increase on Accuracy, although there are some values such as 17-18 having higher Accuracy than others, however it’s not statistically significant difference.

Although this runs on scaled data in the training dataset, the Accuracy supposedly should be lower if we use unscaled data.

Table 2. % Accuracy by K Value using KNN

| **Value of K (Scaled data)** | **Accuracy (% Correct Predictions )** |
| --- | --- |
| 1-4 | 81.23% |
| 5-6 | 83.02% |
| 7-10 | 83.09% |
| 11 | 83.06% |
| 12 | 83.12% |
| 13,15 | 83.19% |
| 14,16 | 83.22% |
| 17-18 | 83.35% |
| 19-20 | 83.32% |

**[Note to TA]: The next steps in “K-Nearest Neighbor” are 1) to reduce variables and see if accuracy can be increased; 2) to run the results in test and validation datasets for getting the recommended model to use.**

**3. Model Comparison**

**[Note to TA]: This part will be completed as part of the Final Submission. Here, we will utilize the model to specifically answer the questions posed on our objectives**

**4. Conclusion**

**[Note to TA]: This part will be completed as part of the Final Submission.**

**[Note to TA]: The R code will be provided as part of the Final Submission.**

**5. Timeline**

**July 9, 2022: Model fine-tuning discussion**

**July 16, 2022: Final project preparation**

**July 20, 2022: Final project submission**

**6. R-Code**

**Please find our R-Code on our team’s repository on Github**