Final Report

Linh Tran

3/30/2021

I - Introduction

Significance and Background

According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths. This dataset used in this report 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 relavant information about the patient.

Data Source: https://www.kaggle.com/fedesoriano/stroke-prediction-dataset./

Variables included in the original data set:

- id: unique identifier
- gender: "Male", "Female" or "Other"
- age: age of the patient
- hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
- heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
- ever_married: "No" or "Yes"
- work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
- Residence_type: "Rural" or "Urban"
- avg_glucose_level: average glucose level in blood
- bmi: body mass index
- smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*
- stroke: 1 if the patient had a stroke or 0 if not *Note: "Unknown" in smoking_status means that the information is unavailable for this patient

Questions of Interest

I am interested in looking at and comparing several predictive models' performance to see which model has the best accuracy. Besides that, it would also be interesting to see variables that are significant and included in the model. Because of the nature of the dataset, ROC will be used as the evaluation metric.

Data Cleaning

The dataset is essentially clean and contains 12 variables (including ID) and 5110 observations. Excluding the id, we have ten predictor variables and one binary outcome variable `stroke` (0:no stroke, 1:stroke). . Categorical variables are converted from character to factor class in order for them to be included in the model and for the purpose of analysis. I also omit the "Other" category which only includes one observation from the `gender` variable, leaving "male" and "female" as the two categories. Binary predictor variables such as `hypertension`, `heart_disease`, and `ever_married` are also recoded so that `0` means no and `1` means yes. The `work_type` variable which has 5 levels (children, govt_job, never_worked, private, and self_employed) are recoded to lower snake case. My main aim is to find out the appropriate models that have a better performance on prediction by comparing several models' performance.

II- Exploratory analysis/visualization

Exploratory analysis

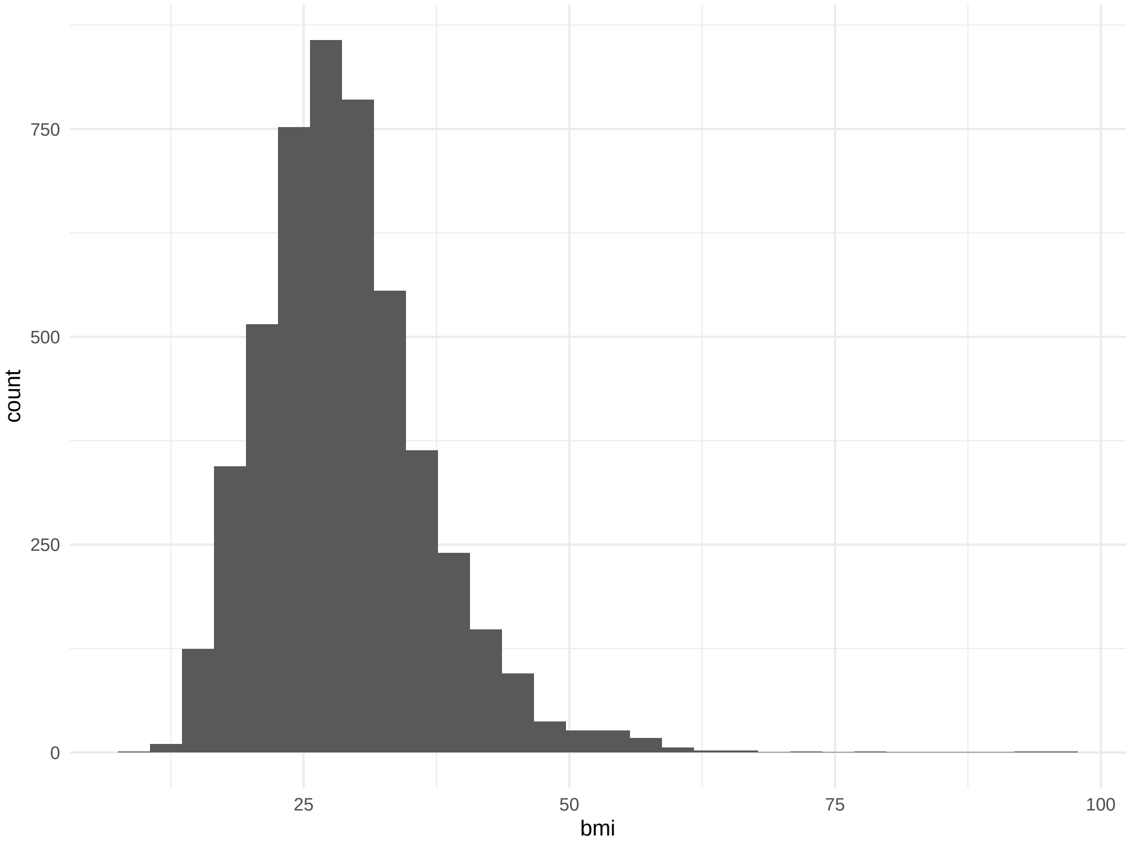
Using `summary` function, we can have a brief look at the summary statistics of the dataset. The imported dataset has 5110 observations in total. Excluding the id, we only gave ten features and one binary outcome variable-stroke (0:no stroke, 1:stroke). We found that the stroke outcome distribution is imbalanced with 4861 observations have no stroke while 249 observations have a stroke. The proportion of people who had a stroke is roughly 5%, which show a highly imbalanced outcome. I tested out oversampling method to balance this dataset.

Missing values

Firstly, I look at how many missing values are in the dataset per column.

Variable	n
id	0
gender	0
age	0
hypertension	0
heart_disease	0
ever_married	0
work_type	0
Residence_type	0
avg_glucose_level	0
bmi	201
smoking_status	1544
stroke	0

Base on the table generated above, there are 1544 “Unknown” values for `smoking_status` variable and 201 missing values for `bmi` variable. Among these 201 missing values in BMI, 40 observations have a stroke while 161 observations without stroke. We can also look at the distribution of BMI, which is right-skewed. Thus I decided to impute `bmi` values with the median.



Imputation

I impute missing values in `bmi` using `preProcess` function.

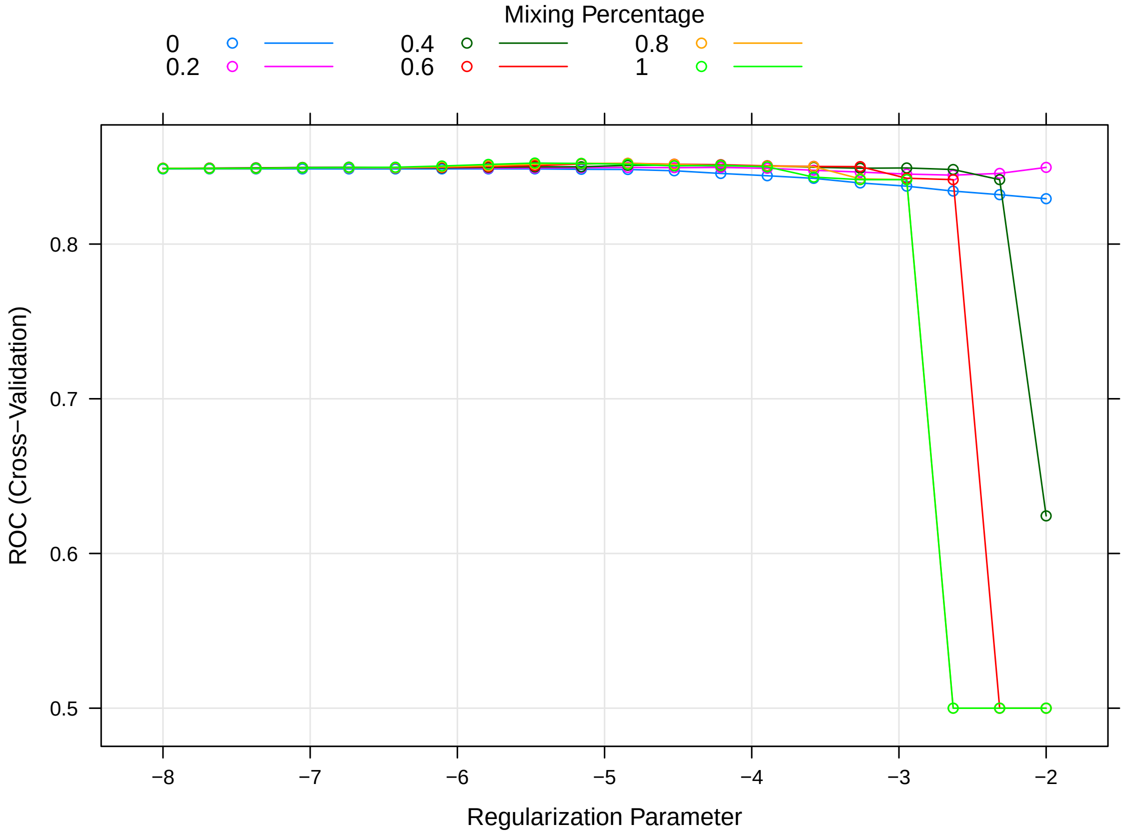
```
set.seed(2021)
trRow_imp = createDataPartition(y = stroke_df$stroke, p = 0.7, list = F)
train_imp = stroke_df[trRow_imp, ]
test_imp = stroke_df[-trRow_imp, ]

knnImp = preProcess(train_imp, method = "knnImpute", k = 3)
train_imp = predict(knnImp, train_imp)
#vis_miss(train_imp)
train_imp = predict(knnImp, train_imp)
#vis_miss(train_imp)
test_imp = predict(knnImp, test_imp)
#vis_miss(test_imp)
```

III - Model building

I decided to fit 4 models: penalized logistic regression, GAM, LDA and KNN.

Penalized logistic regression



```
##     alpha      lambda
## 109     1 0.004195746
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1458   74
##        Yes    0    0
##
##                 Accuracy : 0.9517
##                   95% CI : (0.9397, 0.9619)
##     No Information Rate : 0.9517
##     P-Value [Acc > NIR] : 0.5309
##
##                    Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.9517
##              Prevalence : 0.0483
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : Yes
##
```

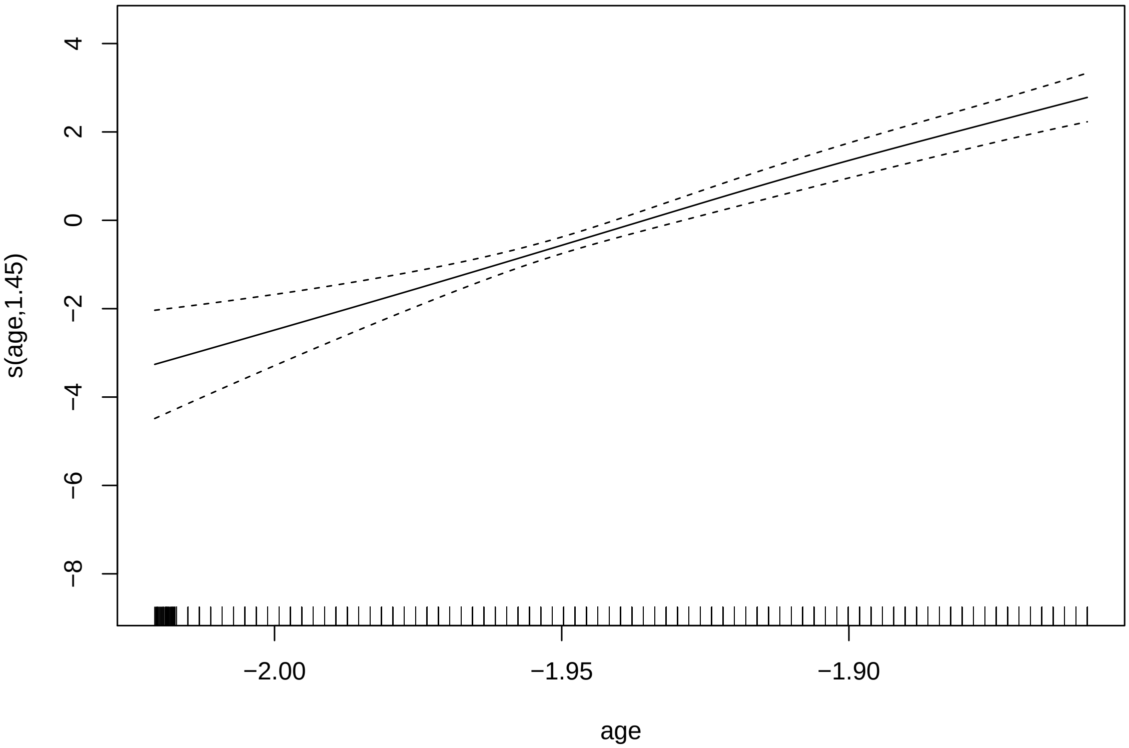
```
## [1] 0.5611955
```

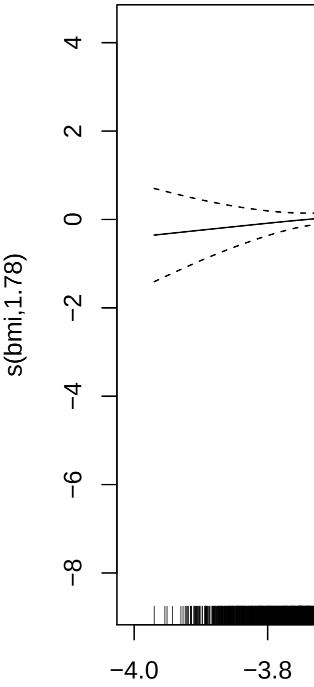
LDA

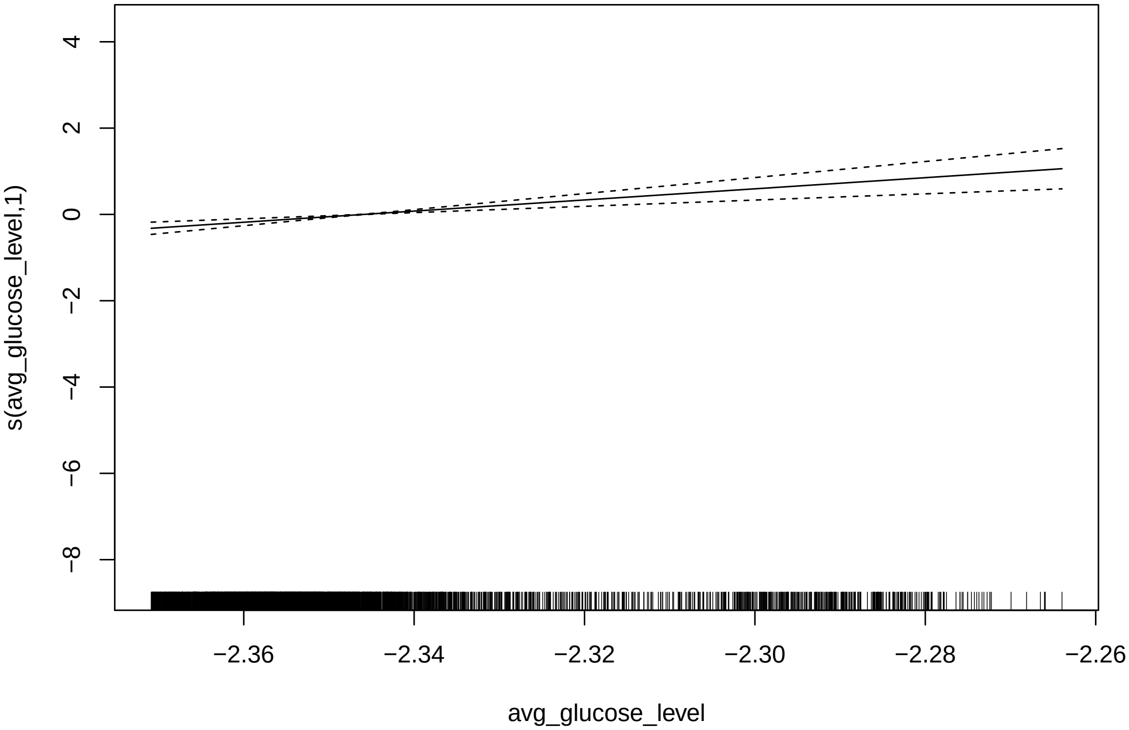
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1458   74
##        Yes    0    0
##
##                Accuracy : 0.9517
##                  95% CI : (0.9397, 0.9619)
##     No Information Rate : 0.9517
##     P-Value [Acc > NIR] : 0.5309
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.9517
##              Prevalence : 0.0483
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : Yes
##
```

GAM

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## .outcome ~ gender + hypertension + heart_disease + ever_married +
##     Residence_type + smoking_status + work_type + s(age) + s(bmi) +
##     s(avg_glucose_level)
##
## Estimated degrees of freedom:
## 1.45 1.78 1.00  total = 12.23
##
## UBRE score: -0.6892917
```







```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1458   74
##        Yes    0    0
##
```

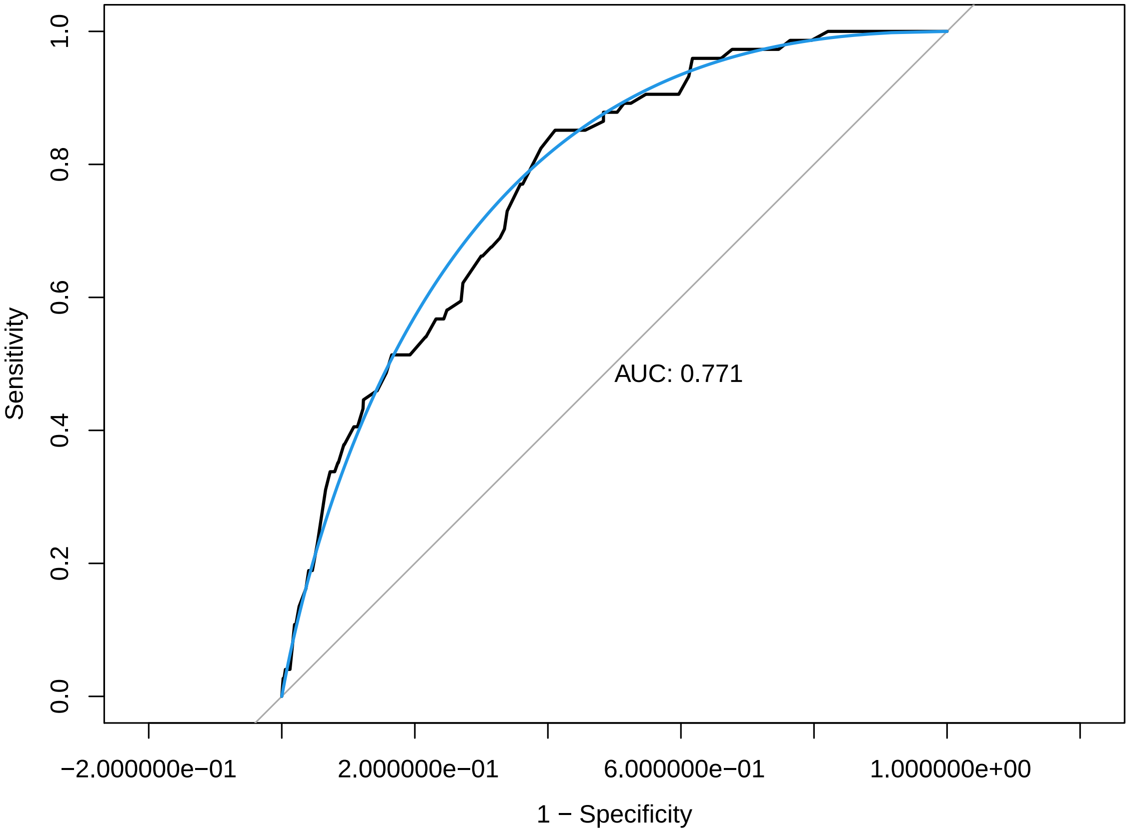
```
##                Accuracy : 0.9517
##                  95% CI : (0.9397, 0.9619)
##     No Information Rate : 0.9517
##     P-Value [Acc > NIR] : 0.5309
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.9517
##              Prevalence : 0.0483
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : Yes
##
```

KNN

```
## 131-nearest neighbor model
## Training set outcome distribution:
##
##   No  Yes
## 3402  175
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1458   74
##        Yes    0    0
##
##                Accuracy : 0.9517
##                  95% CI : (0.9397, 0.9619)
##     No Information Rate : 0.9517
##     P-Value [Acc > NIR] : 0.5309
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.9517
##              Prevalence : 0.0483
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : Yes
```

```
##
## [1] 0.7714798
```



IV - Conclusion

```r
auc <- c(roc.glmn$auc[1], roc.gam$auc[1], roc.lda$auc[1], roc.knn$auc[1])
auc
```

```
## [1] 0.5611955 0.5462314 0.5106310 0.7714798
```

After evaluating the performance of the penalized logistic regression, GAM, LDA and KNN model, it seems that the KNN model performs the best with the highest AUC. All the models have pretty high accuracy but low Kappa, which is the agreement between the predictive value and the true value. The sensitivity is also very low, this is understandable given the highly imbalanced data. We could fix this by ovesampling, however, for the purpose of our analysis, I opted to evaluate normal sampling data to avoid biased prediction results. The linear discriminant model would be more stable than the logistic regresion model if the distribution of the predictors is approximately normal, which is not the case in this example. LDA is also more popular when we have more than two response classes.