TABLE OF CONTENTS

1. INTRODUCTION

According to WHO (2020), stroke is the **second leading cause of death** and **third leading cause of disability** globally, responsible for approximately **11% of total deaths.** It occurs due to an interruption of blood supply to the brain, preventing brain tissues from receiving oxygen and nutrients. This condition requires **immediate** medical treatment as it can reduce brain damage and prevent death. Potential stroke risks include hypertension, cardiovascular diseases, diabetes, smoking and obesity.

1.1. Business Problem and Objective (Hospital and Healthcare Institute)

Evidence from studies in developed and developing countries shows that patients' **awareness** of established stroke risk factors is **less than 50%** on average. Lack of early recognition of potential stroke factors or warning signs can be extremely detrimental (Eric, 2018). Thus, the **goal of our project is to apply machine learning (ML) techniques to predict how likely an individual is to encounter a stroke based on certain health and lifestyle attributes in the chosen dataset.** With this prediction, patients who have been predicted to have a high risk of suffering from stroke could be flagged and sent for further medical tests to verify if they are truly at high risk.

1.2. About the Stroke Prediction Dataset

The team obtained the stroke dataset from Kaggle. The description of each column can be found in Appendix A. This is a binary classification problem, where the inputs are patients' historical medical information and the outcome are two classes, 1 if the patient had a stroke and 0 if not.

1.3. Target Audience & Value Proposition of the model developed

In this project, our target audience who would be using this prediction model would be people from the healthcare industry who can use such information to identify patients who are likely to suffer from stroke based on health data. Given that the first hour is crucial to anyone who suffers from stroke (INTEGRIS, 2019), it is important that more attentive care is given to these groups of patients who have been identified positively by the model. For example, such high-risk patients should avoid being alone for a long period of time, so that in the event of an onset of stroke, there will be someone with them to call an ambulance and provide them with immediate treatment. By doing so, this would increase their chances of surviving from stroke and avoid long term brain damage (INTEGRIS, 2019).

It is **critical to have an interpretable decision support system** especially in the medical industry for a medical practitioner to know why a patient is classified as high risk for having a stroke. Apart from having a model that predicts the class, providing explanations alongside predictions is crucial for decision making and prevention of future relapses of the disease. Therefore, interpretability of our proposed prediction model is an important aspect as users (e.g. medical practitioners) are not only able to offer an accurate prediction of risk but also able to provide explanations to the contributing health factors.

2. EXPLORATORY DATA ANALYSIS

During initial exploration, we identified a **significant class imbalance** on the target variable ('stroke'). Further, all the observations are stored in a single csv file. To ensure a truly blind test set and prevent information leakage, we performed a **randomised 80%-20% stratified split**, so that both the train and test sets have the same proportion of '1's and '0's before conducting further EDA. The new **train** and **test** dataset contain **4,088** and **1,022 rows** respectively. Out of 4,088 rows, the train dataset contains 3,899 instances of class 1 (positive cases of stroke) and 199 instances of class 0 (negative cases of stroke) (Appendix B). Due to a high class imbalance ratio of approximately **20:1**, stratified cross-validation is ideal and is applied on the 'stroke' feature (in this case our target variable for classification)..

The only column that contains **missing value** is the *'bmi'* variable with **165 null observations**, making up 4% of the entire training dataset (Appendix C). Since this is a small dataset, we decided to retain the information by filling the null values instead of dropping them to prevent significant data losses.

Outliers in both **categorical and continuous variables** were detected in the dataset. For *'gender'*, there is a single observation for the category *'Other'* (Appendix D1) which is an anomaly that should be handled. From the boxplots of *'age'*, *'avg_glucose_level'* and *'bmi'* (Appendix D2), we observe that there are no significant outliers for *'age'*, but outliers are present in *'avg_glucose_level'* and *'bmi'*.

In addition, the distribution for *'avg_glucose_level'* and *'bmi'* (Appendix E1 & E2) are **skewed**. Hence preprocessing steps are needed to transform these variables closer to a normal distribution. Furthermore, as shown in the **correlation matrix** (Appendix F), we can conclude that there is **no significant correlation** between variables thus concerns about **multicollinearity are low or negligible**.

3. DATE CLEANING & PREPARATION

To ensure high-quality data before modelling & tuning, we preprocessed the data in the following steps:

1. **Dropping** the ***'id'*** column since it is just a primary key for the relational dataset.
2. **Categorical covariates handling** ***(No dropping of encoded variables >= 3 for unique cuts in tree):***
   Dummy Encoding: 'gender', 'hypertension', 'heart_disease' and 'ever_married',
   One-Hot Encoding: 'work_type', 'Residence_type' and 'smoking_status'.
3. **Converting** the 1 instance of 'Other' in the 'gender' column to ***'female'*** as it has the highest count.
4. **Filling missing values** in **'bmi'** column. Due to the small dataset, it is not advisable to drop null values. Hence we experimented with 3 methods to fill the missing values - using Mean, Median or **Multivariate Imputation by Chained Equation (MICE)**. Comparing the different distributions after filling in the missing values, we found that the MICE imputation method gave a distribution that is closest to the original distribution of 'bmi' values (Appendix G). **MICE** is the **preferred imputation method** as it uses other variables in the dataset to predict the value in 'bmi' based on regression. Filling the data using MICE, it can better measure the uncertainty of the missing value.
5. **Features Scaling** of continuous variables. This prevents the features with higher value ranges from dominating the model training process.
   a. **Standard Scaling**: ***'age'*** - original distribution follows a normal distribution
   b. **Log Transformation:** ***'bmi'*** and ***'avg_glucose_level'*** - presence of outliers and skewed distribution, forces the distribution to be closer to a normal distribution (Appendix H).

4. MODEL TRAINING AND METHODOLOGY

4.1. Choice of models

The team has decided to approach this classification problem with the following 5 models:

1. **Logistic Regression** - the go-to, easy to comprehend method for binary classification problems
2. **Decision Tree** - simple classification algorithm with high interpretability and feature importance
3. **Random Forest** - adds randomness to the model, moderate interpretability and feature importance
4. **XGBoost** - highly efficient model, works well with imbalance datasets
5. **Convolutional Neural Network** - modern way of classifying, tends to have high accuracy

A mix of simple and complex models allow the team to analyse and determine which model would be appropriate for this business problem. Moreover, this allows the team to eventually choose a model that can **balance the trade-off between interpretability, performance and real world applicability.**

4.2. Cross-validation

Tuning the model hyperparameters, we use **GridSearchCV** with **5-fold** cross validation, with stratified sampling. This ensures that the **proportion of *'stroke' =0* and *'stroke' =1* is the same** across all folds.

4.3. Handing of imbalanced dataset

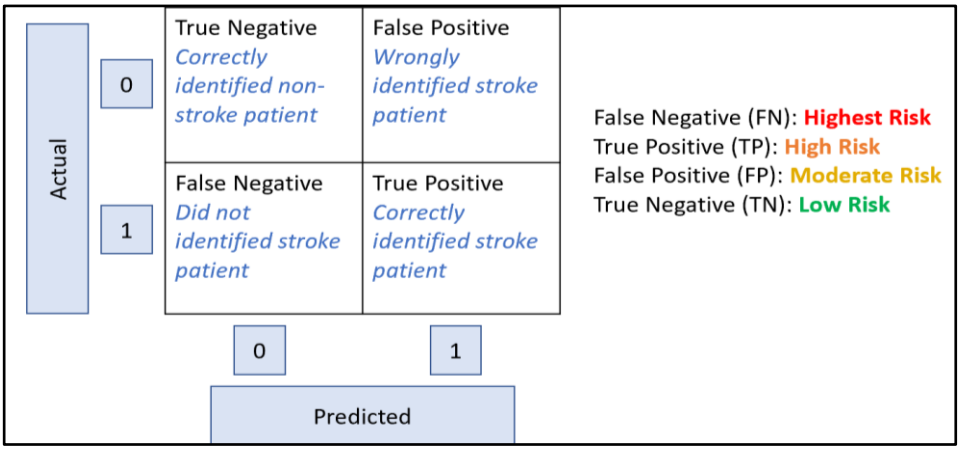
The team has decided to address this issue of imbalanced dataset with the use of Class Weights in the models. This is because the **imbalanced dataset could cause our model to be biased towards the majority class**, classifying the majority as negative so that it can achieve a higher level of performance.

With the use of a class weights parameter, a different weight amount will be assigned to each class. The difference in weights will then influence the classification of the classes during the training phase. This enables the penalisation of misclassification made by the minority class by setting a higher class weight and at the same time reducing weight for the majority class.

There are two ways to determine the appropriate class weights for each class. Firstly, we can **set the class weight as "balanced"** to adjust the weights to be inversely proportional to the class frequencies. The second way would be to **use the GridSearchCV approach** to find the most ideal class weights that maximise performance. Since the model's performance in this business problem is crucial (causing life or death), the team has decided to **proceed with the second approach of using GridSearchCV**.

4.4. Choice of Performance Measure

Several performance measures are available for classification problems. However, it is vital to choose the one that most suits our needs. We classify the risks of predicting each quadrant as shown below.



Taking our business problem into consideration, it is critical for our prediction model to be able to **minimise False Negative (FN)** and **maximise True Positive (TP)**. It would be the worst-case scenario for our model to predict someone as negative and the person does get a stroke subsequently. This may lead to a life-threatening outcome which should be avoided. Hence, we would want the **Recall score to be as high as possible**, correctly predicting more positives and falsely predicting less negatives.

However, we should not only use Recall as the evaluation metric as we would want to also **minimise False Positives (FP)**. Although the consequence of FP has a smaller impact than FN and TP, we would not want the model to predict the majority of the people as positive in order to minimise FN. This would lead to a huge wastage of medical resources, and reduce our model's reliability. In summary, we would also want our **Precision to be as high as possible**. Therefore, in order to maximise Recall and Precision, our team would choose **Precision-Recall Area Under Curve (PR AUC) as the main evaluation metric.**

One reason why the team did not use ROC AUC, which is a more common measure to understand the different thresholds with regards to **True Positive Rate (TPR)** and **False Positive Rate (FPR)**, is because in a hugely

imbalanced dataset where we have a large number of **True Negatives (TN)**, it will tend to cause our FPR to be very small [FP / (FP + TN)] (Huilgol, 2020). This will cause the x-axis of the ROC plot to be "pushed" to the left, resulting in a high ROC AUC value which can be misleading. On the other hand, PR Curves **do not make use of TN**. Hence, PR Curves are more appropriate for situations where the number of TN is very large or when TN is less significant in the business context. Therefore, the team decided to use **PR AUC, also known as average precision score in scikit-learn.**

Alternatively, we could also consider using **F2-Measure** as our evaluation metric (Brownlee, 2020). F2-Measure is a modification of F1-Measure (harmonic mean of precision and recall), with the effect of lowering the importance of precision and increasing the importance of recall vice versa.

5. MODEL EVALUATION, SELECTION & INSIGHTS

5.1. Preliminary Comparison of Models Performance (Cross Validated)

Model	ROC AUC		Recall		Precision		F2*		PR AUC	
	Train	Val	Train	Val	Train	Val	Train	Val	Train	Val
Log Reg	0.842	0.83	0.828	0.809	0.137	0.134	0.412	0.403	0.122	0.118
Classification Tree	0.840	0.807	0.778	0.744	0.149	0.144	0.421	0.404	0.127	0.119
Random Forest	0.916	0.819	0.866	0.659	0.193	0.15	0.509	0.392	0.173	0.116
XGB	0.913	0.821	0.871	0.699	0.179	0.145	0.491	0.395	0.162	0.116
C-NN (not tuned)**	0.500	0.500	0.039	0.000	0.800	0.000	-	-	0.0782	0.048***

**F2-Measure = ((1 + 2^2) * Precision * Recall) / (2^2 * Precision + Recall)*
***Unable to use Keras-tuner to maximise PR AUC, only ROC_AUC optimization.*
****C-NN Val is the baseline for PR-AUC where baseline, y = sample (P/P+N) [i.e sample = validation dataset]*

5.2. Model of Choice

Based on PR AUC, both Logistic Regression (0.118) and Classification Tree (0.119) have surprisingly achieved a better performance as compared to the more complex models. As such, the team has decided to proceed on with the simple models since they are **not only performing better but also easily interpretable**, which meets both our objectives. Although the value of PR AUC for Classification Tree is slightly higher than Logistic Regression, we chose to use **Logistic Regression** as our final model as the difference between the train and validation PR AUC is significantly smaller **(0.004)** for Logistic Regression than the Classification Tree **(0.008)**, which shows that there is less risk of overfitting. Moreover, Classification Tree is known to be **susceptible to overfitting**, especially when scaling up.

5.3. Model Optimisation

Most machine learning algorithms are not good at handling skewed class data in the case of CNN - a highly complex model, it performed the worst at PR AUC with only 0.048 (Singh, 2020).

Hence, the chosen Logistic Regression model will be tuned further to overcome the issue of class imbalance and achieve a better PR AUC score. GridsearchCV results show that best class weights returned using stratified cross-validation are **0.075 for class 0** and **0.925 for class 1**, and the ratio of weights between class 0 and class 1 would be approximately 1:12, with a higher weightage on the minority class. With the new weights, we run the model again and we can see that the PR AUC has improved from **0.118 to 0.121** (**+2.5%**).

Validation Result	F2	Recall	Precision	ROC AUC	PR AUC
Before (Δ Weights)	0.403	0.809	0.134	0.83	0.118
After (Δ Weights)	0.404	0.694	0.152	0.83	0.121

5.4. Check for Possibility of Overfitting before fitting ideal model on Test Data

After optimising our model, it is important to ensure that there is **no sign of overfitting** taking place in the model. Based on the table below, we can observe that the validation score for all the CV folds are very close to the training score and this suggests that this model is unlikely to overfit on the test set.

	split	train_average_precision	val_average_precision
0	0	0.128921	0.105143
1	1	0.127240	0.131975
2	2	0.135137	0.110814
3	3	0.116796	0.123395
4	4	0.117958	0.131921
5	mean	0.125210	0.120650

This result is not surprising because this model is **consistent with the rule of thumb** where it states that the number of observations (4088) should be more than or equal to 10 times the number of features (18).

5.5. Benefit Structure to adjust the probability threshold

	Correctly identified stroke patient (TP)	Did not identified stroke patient (FN)	Wrongly identified stroke patient (FP)	Cost of each test	Optimal Threshold
Option 1	+10	-10	-2	-1	60%
Option 2	+100	-100	-2	-1	20%

Since healthcare models should take a more conservative approach in predicting stroke, the most optimal threshold to be chosen should be **20%**. The recommended threshold is usually below 0.5 (Arjun, 2019) , i.e. roughly around 0.2 in the case of a binary classification to maximise recall, hence our team has decided to assign the threshold of 20% to our model. Moreover, this can be understood easily because with a higher benefit value and cost assigned to TP and FN respectively, this would mean that the model would prioritize recall, resulting in a more conservative threshold value by classifying more patients as positive. Ultimately, the team wants to achieve the best recall score, that is to predict more positive cases correctly and **avoid missing patients with high risk of stroke**. Moreover, this benefit structure provides flexibility for the medical professionals to exercise judgment and customise their own benefit structure to their needs. This would then translate into an optimal threshold that would help achieve their objectives.

5.6. Final Model Results fitted on Test Set

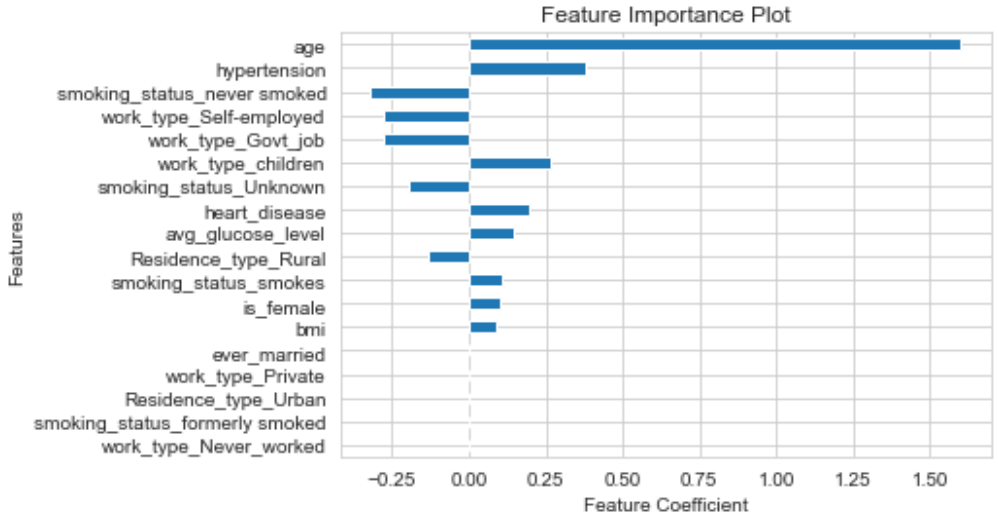
	PR AUC	Recall	Precision	F2	ROC AUC
Train*	0.192	0.945	0.092	0.33	0.840

Test	0.221	0.96	0.094	0.338	0.867

**Based on full training data to maximise information gain by algorithms.*

With the 20% threshold, our model obtained a 0.221 PR AUC score on the test dataset, with a high recall of 0.96. A recall value of 96% means that 4 of every 100 stroke patients in reality are not identified by our model and 96 predicted correctly. We observe that the test results slightly outperforms the train results, and it could be due to a lucky train-test split, which can be minimised by increasing the size of the dataset.

5.7. Feature Importance



From the feature importance plot, we observe that the top 3 important features are ***'age'***, ***'hypertension'***, ***'smoking_status_never_smoked'***. For features with positive coefficients such as ***'age'*** and ***'hypertension'***, we can conclude that the higher the value of these features, the more likely a patient is diagnosed with stroke. Similarly, for features with negative coefficients such as ***'smoking_status_never_smoked'*** (which is a binary feature), a patient is less likely to have a stroke for higher values of these features (e.g. if the patient has never smoked, class 1 for binary feature *'smoking_status_never_smoked'*)

5.8. Possible Improvements & Insights from our methods

From the test results, we observe that the PR AUC and Recall are higher than the corresponding train results. This could imply possible underfitting of the chosen model, or a lucky split during the train-test split. As the dataset used consists of only 5,110 observations, getting more training data could help mitigate the risk of model underfitting and also reduce the chances of a lucky train-test split.

A possible way to improve the complexity of our model is to introduce new variables into the model through feature cross. Feature cross is a synthetic feature created by combining two or more features. (Sharma, 2021) By introducing new features into the model training process, it can increase the model complexity and reduce the risk of underfitting while potentially improving the model performance.

6. CONCLUSION

In summary, the team believes that it is important to contextualise the model based on the requirement of the business problem. By **choosing the right evaluation metric that fits the context of our business problem,** it enables us to improve the model's performance and have a greater understanding of what the model is prioritising. Finally, we believe that this Logistic Regression model would be able to provide **a right balance**

between interpretability and performance for the medical professionals that would aid their work of identifying patients with high risk of stroke.

REFERENCES

Badr, W. (2019, January 12). *6 different ways to compensate for missing data (data imputation with examples)*. Medium. Retrieved November 14, 2021, from https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputationa-with-examples-6022d9ca0779

Brownlee, J. (2020, January 14). *A gentle introduction to the fbeta-measure for machine learning*. Machine Learning Mastery. Retrieved November 16, 2021, from https://machinelearningmastery.com/fbeta-measure-for-machine-learning/

Brownlee, J (2021, January). *How to use ROC Curves and Precision-Recall Curves for Classification in Python. Retrieved November 16, 2021, from* https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

Chou, S.-Y. (2020, April 25). Compute the AUC of precision-recall curve. Sin. Retrieved November 14, 2021, from https://sinyi-chou.github.io/python-sklearn-precision-recall/.

Classifier evaluation with imbalance datasets. Classeval. (n.d) Retrieved November 16, 2021, from https://classeval.wordpress.com/introduction/introduction-to-the-precision-recall-plot/

Eric S, D. *Stroke in the 21st Century: A Snapshot of the Burden, Epidemiology, and Quality of Life.* Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6288566/

Fedesoriano. (2021, January 26). *Stroke prediction dataset*. Kaggle. Retrieved November 14, 2021, from https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

Ghoneim, S. (2019, April 8). *Accuracy, recall, precision, F-Score & Specificity, which to optimize on?* Medium. Retrieved November 14, 2021, from https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124

How to dealing with imbalanced classes in machine learning. Analytics Vidhya. (2021, January 6). Retrieved November 14, 2021, from https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/

INTEGRIS. (2019). *What is the Golden Hour in Strokes? Why is it Important?*. Retrieved from https://integrisok.com/resources/on-your-health/2019/may/why-is-the-golden-hour-so-important-when-it-comes-to-stroke

Lador, S. M. (2017, October 22). What metrics should be used for evaluating a model on an imbalanced data set? Medium. Retrieved November 16, 2021, from https://towardsdatascience.com/what-metrics-should-we-use-on-imbalanced-data-set-precision-recall-roc-e2e79252aeba

Precision vs recall: Precision and recall machine learning. Analytics Vidhya. (2021, March 9). Retrieved November 14, 2021, from https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/

Radečić, D. (2019, September 17). *Stop using mean to fill missing data*. Medium. Retrieved November 14, 2021, from https://towardsdatascience.com/stop-using-mean-to-fill-missing-data-678c0d396e22

Sharma, R. (2021, August 20). Why feature crosses are still important in machine learning. LinkedIn. Retrieved November 16, 2021, from https://www.linkedin.com/pulse/why-feature-crosses-still-important-machine-learning-rakesh-sharma/.

Singh, K. (2020, October 6). *How to Improve Class Imbalance using Class Weights in Machine Learning.* Retrieved November 16, 2021, from https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/

Solutions, E., & Name, *. (2016, November 11). *Accuracy, precision, Recall & F1 score: Interpretation of performance measures*. Exsilio Blog. Retrieved November 16, 2021, from https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

WHO(2020). *The top 10 causes of death*. Retrieved from https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

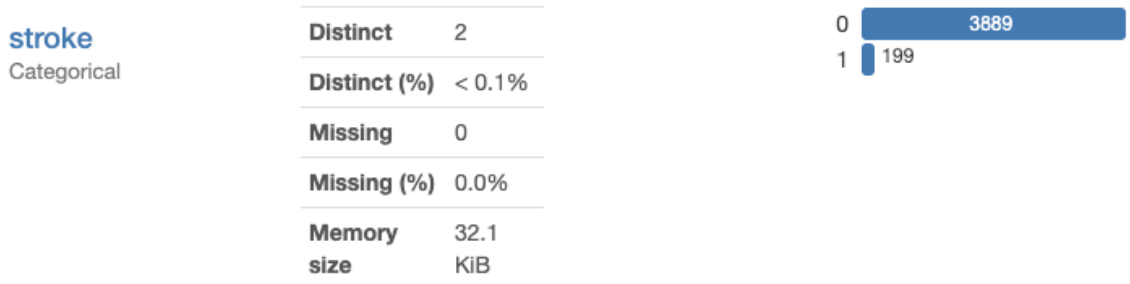
APPENDICES

Appendix A - Description of Dataset

1) **id**: unique identifier
2) **gender**: "Male", "Female" or "Other"
3) **age**: age of the patient
4) **hypertension**: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5) **heart_disease**: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
6) **ever_married**: "No" or "Yes"
7) **work_type**: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
8) **Residence_type**: "Rural" or "Urban"
9) **avg_glucose_level**: average glucose level in blood
10) **bmi**: body mass index
11) **smoking_status**: "formerly smoked", "never smoked", "smokes" or "Unknown"*
12) **stroke**: 1 if the patient had a stroke or 0 if not

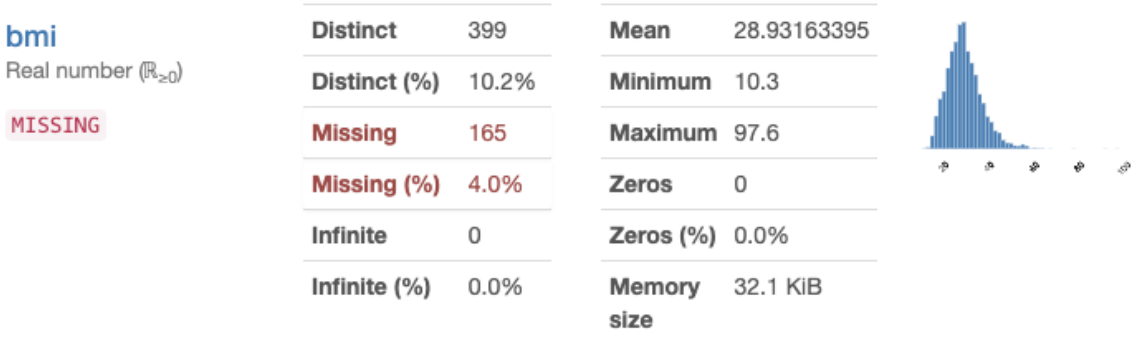
**Note: "Unknown" in smoking_status means that the information is unavailable for this patient*

Appendix B - Class Imbalance



Appendix C - Missing Values

C1) 165 missing observations in 'bmi'

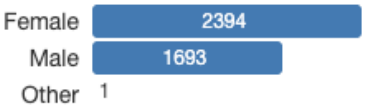


Appendix D - Outlier Detection

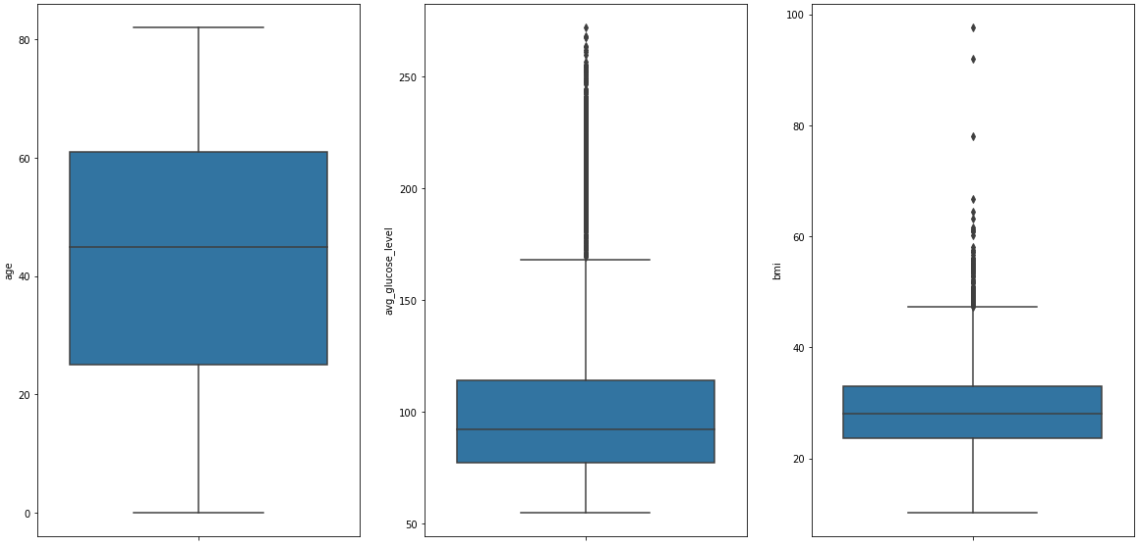
D1) 1 observation of 'Other' in 'gender'

gender
Categorical

Distinct	3
Distinct (%)	0.1%
Missing	0
Missing (%)	0.0%
Memory size	32.1 KiB

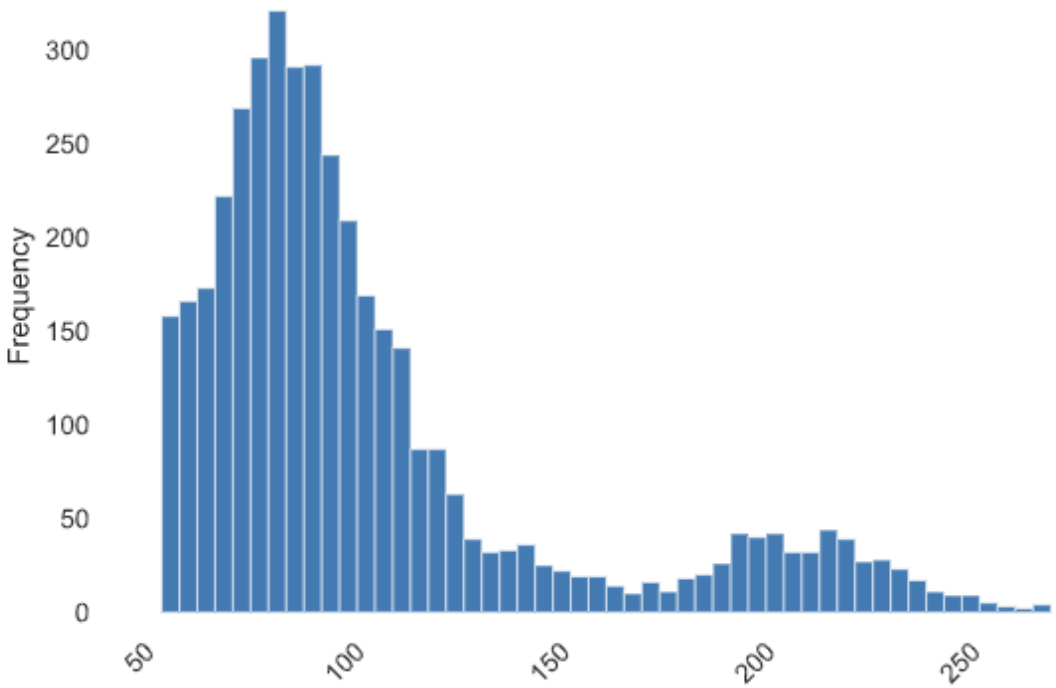


D2) Boxplots for 'age', 'avg_glucose_level' and 'bmi'

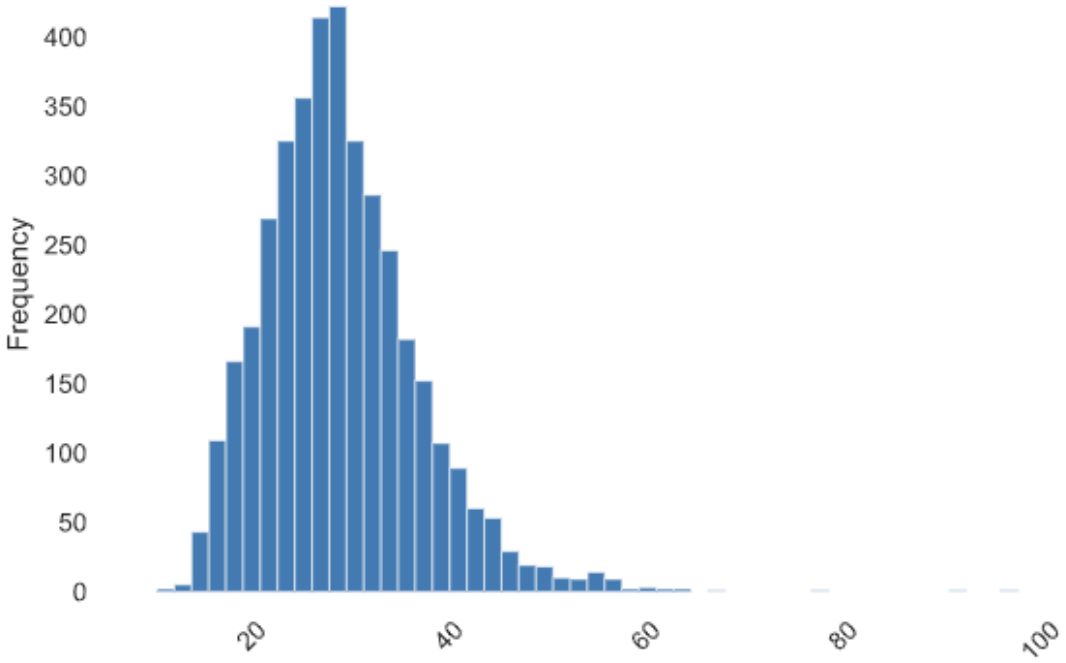


Appendix E - Skewness of Distribution

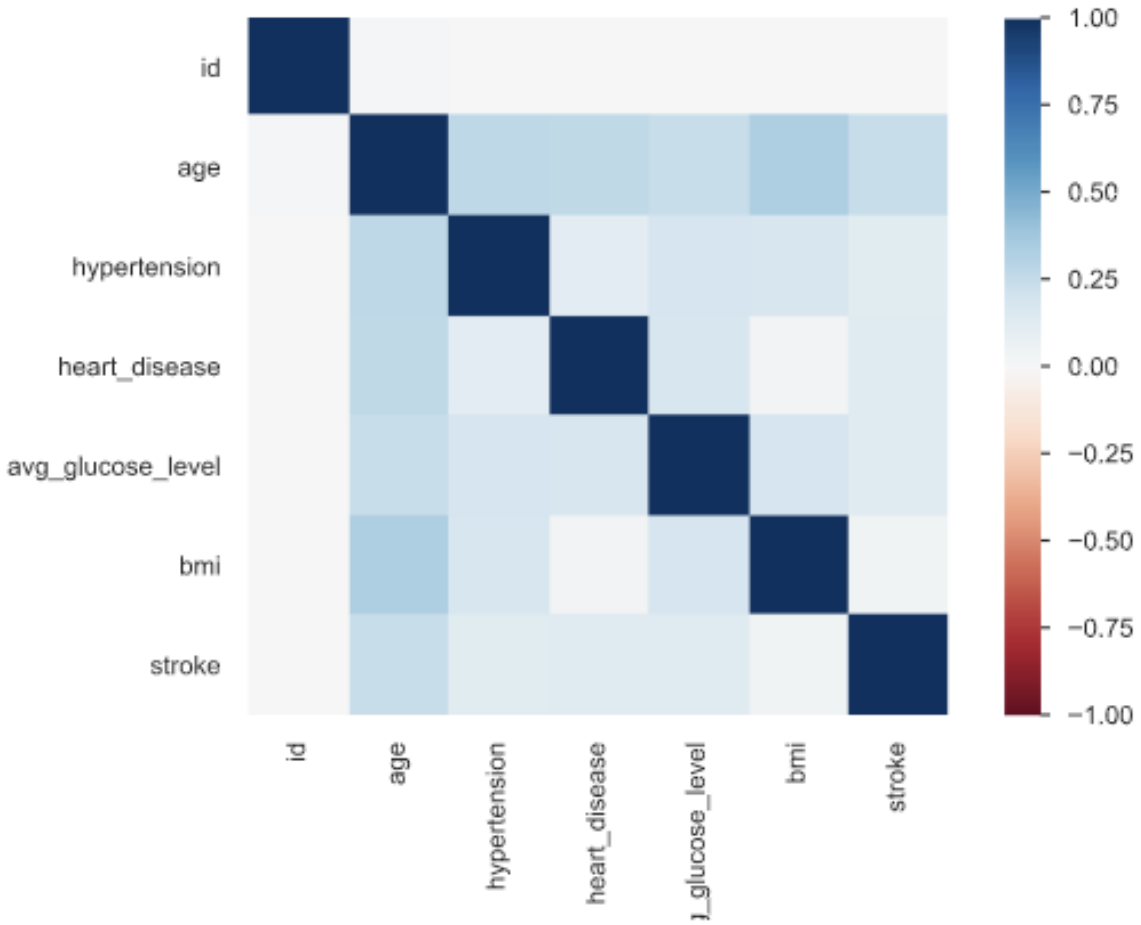
E1) Distribution of 'avg_glucose_level'



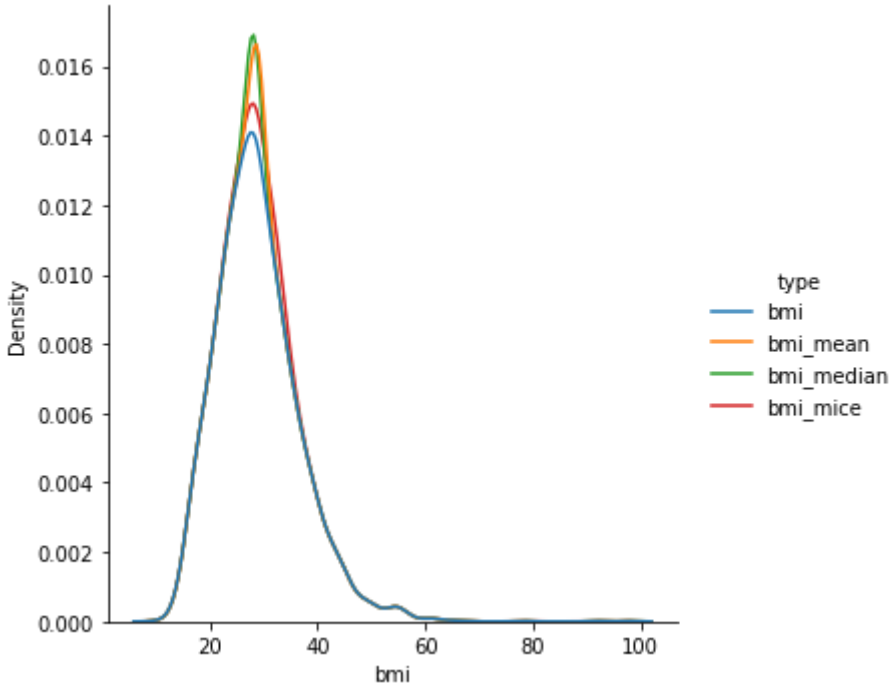
E2) Distribution of 'bmi'



Appendix F - Correlation Matrix



Appendix G - Comparison of Distribution using Different Imputation Methods for 'bmi'



Appendix H - Distribution after Log Transformation