Machine Learning Classification Methods for Stroke Prediction

by

SN: 18006555

November 2023

A Report submitted in part fulfilment of the

Degree of Master of Science:

Data Science & Machine Learning

COMP0172: AI for Biomedicine and Healthcare, Coursework 1

Department of Computer Science

University College London

Dataset used: https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

CONTENTS

1. INTRODUCTION

We aim to apply machine learning for predicting strokes amongst patients, given various physical and health descriptors. The importance of building and optimising such a model is derived from the World Health Organisation (WHO), who determine stroke as the 2nd leading cause of death globally, accounting for approximately 11% of deaths.

Prevention of stroke fatalities is heavily influenced by the early detection and prediction of stroke. Machine learning can prove instrumental in providing medical diagnosis of stroke, by analysing large and complex medical datasets to identify patterns via binary classification models to predict if a patient may suffer from a stroke in the future.

These predictions can be used by healthcare professionals to make more accurate and precise diagnoses, accelerate and reach decisions earlier, thus improving the chance of effective prevention upon diagnosis. Reaching diagnosis earlier for stroke is particularly beneficial since medications can be prescribed to control blood pressure or cholesterol levels, reducing the likelihood of a stroke. Machine Learning can also take a more unbiased approach, reducing human error and external factors potentially resulting from a healthcare facility. It is crucial to assert that machine learning predictions are not intended to be used without professional interpretation and intervention, but rather as an additional factor when handling situations of medical prevention and diagnosis.

2. DATASET CHARACTERISTICS

2.1. Feature Analysis

We utilise a stroke dataset from Kaggle with 5110 patients, 10 features and 1 target variable. The 10 features are a mix of numerical and categorical data, broken down as follows:

- Binary: Gender, Hypertension, Heart disease, Ever married, Residence type
- Numerical: Age, Average glucose level, BMI
- Categorical: Work type, Smoking status

We aim to perform feature engineering on this data set i.e. univariate feature analysis, feature generation/selection and feature modelling, since upon closer inspection of the data, we observe the following issues which require these data engineering processes.

First analysing the continuous numerical features, the following violin plots display their distribution split by stroke class:

Figure 2.1: Violin Plots of Stroke against BMI, Avg Glucose Level and Age

We observe that BMI appears to have no significant differences between the two plots, whereas Avg Glucose Level has a noticeable bulge towards the higher levels (150+) for positive cases of Stroke. Age is significantly distinct, with the majority of stroke patients being above 40 years of age.

This suggests a significant impact from a patient's age, with some impact from a patient's average glucose level, in determining the likelihood of stroke within a patient.

2.2. Feature Engineering

Anomaly Detection

We then observe the numerical variables via a histogram, and understand that BMI and Avg Glucose Level appear to be normally distributed.

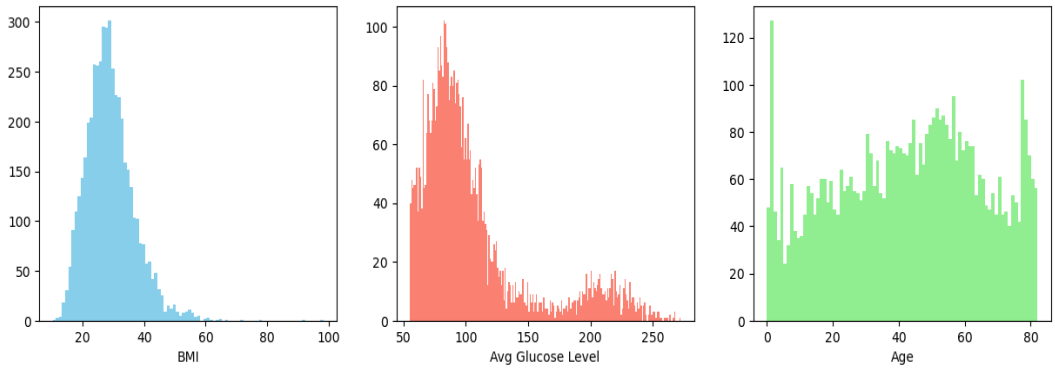

Figure 2.2: Histogram of BMI, Avg Glucose Level and Age, before anomaly detection

Hence we use a z-score to determine outliers for both features. For BMI, the decision to remove outliers is straight forward, however for Avg glucose level, it becomes more complicated. Retroactively, the final machine learning model improves (with regards to all performance metrics) upon anomaly removal, as the Gaussian distribution is better understood by the learning algorithms. However, heuristically this would suggest that the model may not perform as well for patients with high glucose levels. Overall, we choose to remove these patients as these patients represent a small subsample of the total sample (9.2%) and we prioritise the model to predict all types of patients, rather than a

<10% subset. Finally, age appears to be more uniformly distributed, and we choose to not remove outliers here.

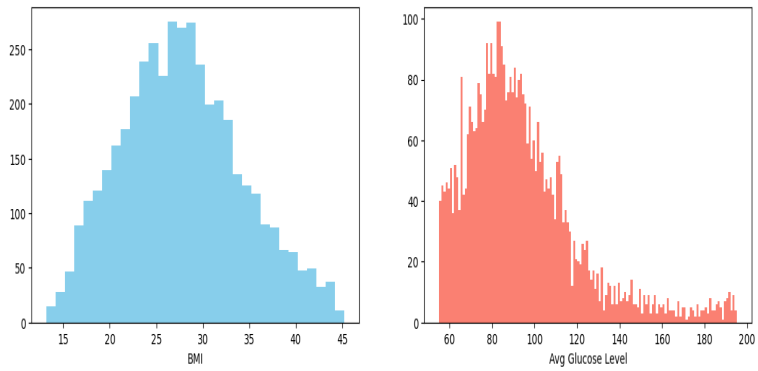

Figure 2.3: Histogram of BMI and Avg Glucose Level, after anomaly detection

Categorical Encoding

We then transform the categorical variables, with simple transformations of binary classes to 0 and 1 for Gender, Ever Married and Residence Type, noting that we removed a single row as there was only 1 anomalous entry for Gender (Other).

For Work Type and Smoking Status, we choose to apply one-hot-encoding for Work Type, and ordinal encoding for Smoking Status as this can be interpreted on a linear scale (never smoked, formerly smoked, smokes). We make an arbitrary choice for 'Unknown' smoking in between never smoked and formerly smoked, to avoid a heavy dependence on Machine Learning for missing values where possible.

Random Forest Regressor for missing values

We then assess the remaining missing values (153 NaN values within BMI), and choose to fill these in via a Random Forest Regressor by using the filled values as training outputs, and the other complete features as training data. The issue of missing values is common within industry (especially healthcare) data, as records are often lost or damaged.

Feature Correlation

We now study the Pearson's correlation coefficients across all features within the dataset, to understand any potential linear correlations between the stroke feature and other features. Stroke does not appear to have any sort of correlation with any variable, with a notable yet still weakly rogue positive correlation with Age ($r = 0.23$).

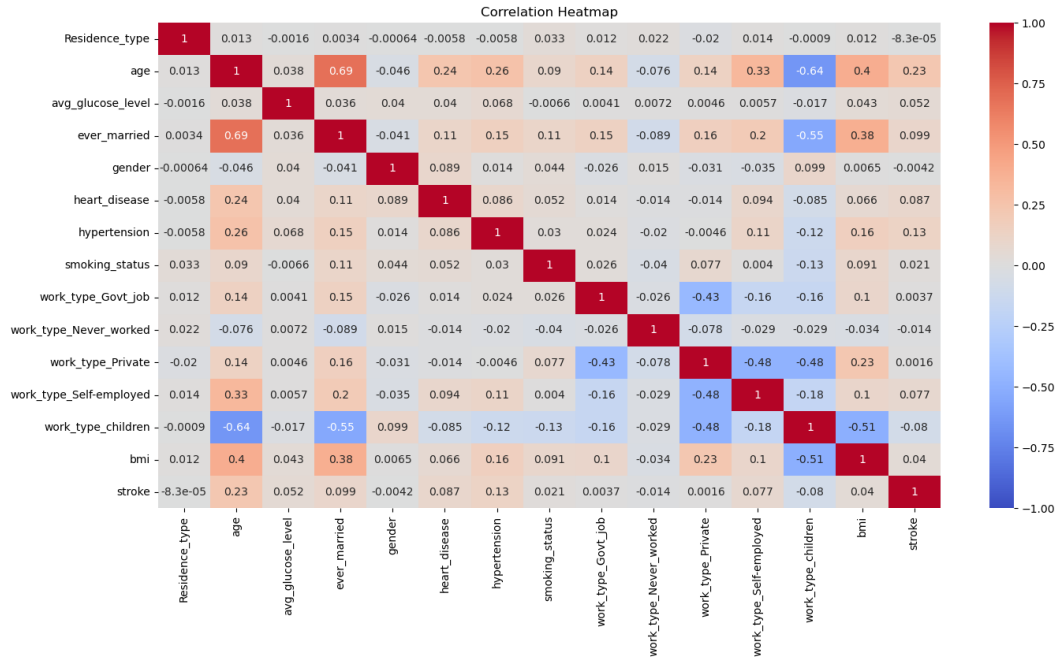

Figure 2.4: Correlation Matrix across all features

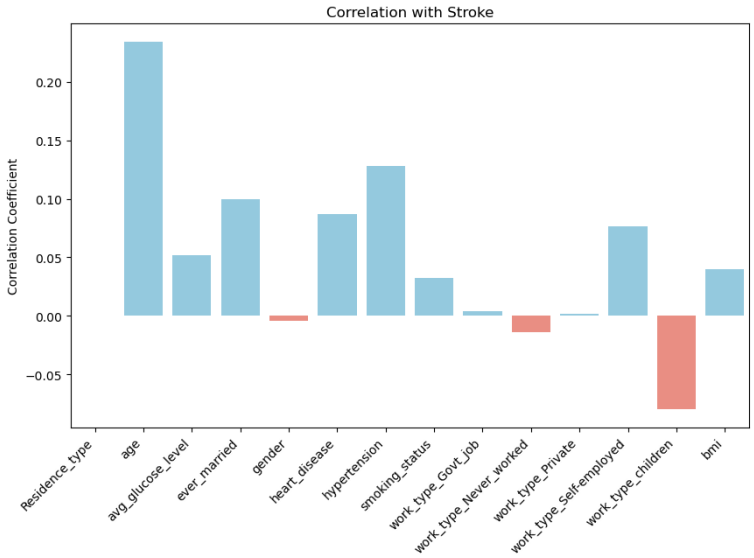

Figure 2.5: Correlations of features with Stroke

This does not imply that the features are not important for predicting strokes, however it suggests the model is not likely to be linear and instead motivates the necessity for machine learning in this scenario to uncover a potential non-linear relationship.

2.3. Feature Importance & Selection

We now conduct Feature Importance to determine the contribution of each feature to a model's predictive performance. We use a Random Forest Classifier and XGBoost to evaluate this:

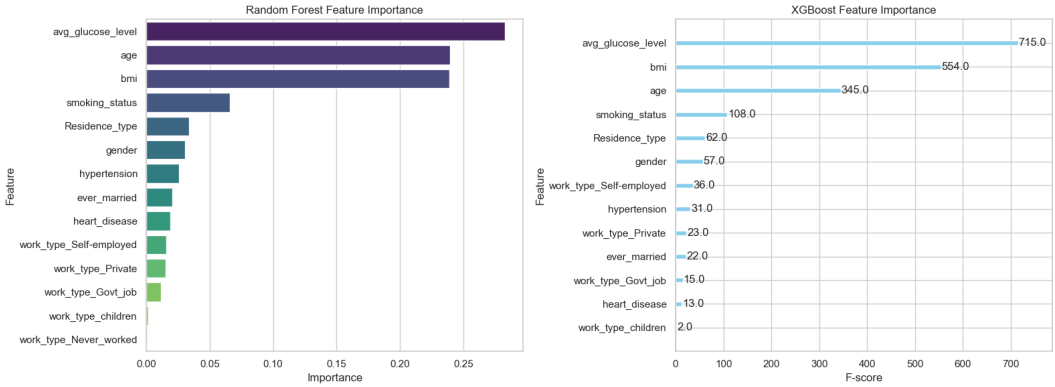

Figure 2.6: Feature Importances via Random Forest and XGBoost

We observe that the rankings and relative scales between importances are similar, giving more ground to suggest that avg glucose level, age, bmi and smoking status are the most important features. This co-incides with previous data exploration, hence we drop the other columns to reduce noise and dimensionality within our dataset. In addition, retroactively we understand that choosing the top 4 features provided the best model performances, compared to any other choice of top N features.

2.4. Imbalance within target variable

One major issue with the dataset in use, involves the strong imbalance within the stroke variable. This is common within datasets where the target is an uncommon event, in applications across fraud detection, medical diagnosis or spam email detection.

For stroke specifically, we observe that only 249 patients are marked with stroke out of 5110 patients (4.9% of the total sample), which will be a large challenge as our machine learning model may become biased towards the majority class (0 = No stroke). We must apply various techniques on the training data set to ensure we can reduce the impact of the imbalance on our final model.

3. MACHINE LEARNING APPROACH

We aim to split our dataset into training and test data, with an 80/20 split. We use stratified sampling to ensure our test set has an equivalent ratio of positive cases to the training set.

3.1. Model Choices

We aim to test 5 different classifier algorithms to compare approaches and performance metrics to understand the most suitable model, out of the following:

- Logistic Regression (l_2 Regularisation)
- Decision Tree

- Random Forest
- XGBoost
- SVM (RBF Kernel)

We will also compare our models versus an absolute baseline model of logistic regression with no extra feature engineering or model design (only feature encoding and missing value handling).

3.2. Sampling Methods for Class Imbalance

In order to minimise the consequence of an imbalanced dataset, we apply a mixture of undersampling and oversampling to our training set. We first apply a random undersampler, to increase the proportion of positive cases from the original 4.9% up to 23.1% (30% ratio from positive to negative cases).

We then perform SMOTE (Synthetic Minority Oversampling Technique) to oversample the minority class (positive cases) via interpolation, ensuring this only occurs on the training set to prevent data leakages. This takes our training set imbalance ratio to 1:1, with 966 total samples from an original 3580.

3.3. Hyperparameter Tuning

We also perform hyperparameter tuning on each model, via a randomised parameter search and 5-fold cross validation, taking the recall as the metric to optimise, in order to capture both recall and precision. Due to the class imbalance, accuracy was generally quite high hence not necessary to focus on. In the medical context, we prefer recall over precision due to the understood higher cost of a false negative over a false positive, hence we choose recall to optimise.

The probability threshold was also considered to adjust, to either lean towards precision or recall as the priority performance metric. Retroactively, we analyse the threshold against recall and F1 score (again noting that accuracy is consistently high and precision is consistently low):

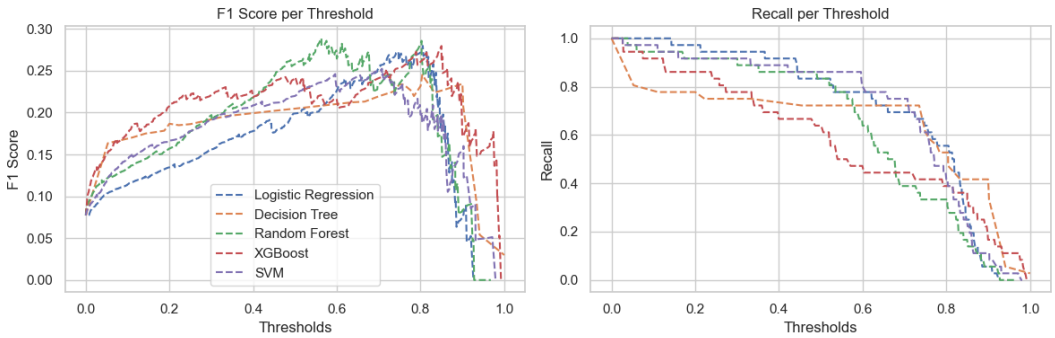

Figure 3.1: Threshold against F1 score and Recall

We observe that the optimal thresholds for F1 score for all models in general is approximately 0.5-0.8; of course recall will decrease as threshold increases, but still seems

to hold above 80% until a threshold of 0.5 for a few models (Logistic Regression, Random Forest and SVM). We opt to keep the threshold at the default 0.5 to balance between both metrics.

4. MODEL RESULTS & EVALUATION

4.1. Performance Metrics

We now analyse our final results for our 5 models, viewed by 3 performance metrics. We choose to focus on accuracy as it represents the model's overall ability to predict correctly.

However accuracy can be heavily influenced by the class imbalance - accuracy may be high even with a poor model, e.g. in this scenario, we could assert a "model" which predicts 'no stroke' 100% of the time, which would have an accuracy of 95.1% automatically. Hence why we also consider recall as an important metric, to provide significance on false negatives i.e. diagnosing a patient as not expected to suffer from stroke, when they will actually suffer from stroke. False negatives have an extremely high cost in this setting, where the consequences of a patient left untreated far outweighs a false positive where a patient is predicted to suffer from stroke when they actually will not. This is why we do not focus on the model's precision (which scores quite poorly), since there always a trade-off between precision and recall. In a practical setting, those diagnosed with stroke should instead be subject to further examination and care via healthcare professionals, rather than be told automatically that they have stroke, in order to minimise the cost further of a false positive.

Model	**Accuracy**	**Recall**	**ROC AUC**
Logistic Regression	0.7310	0.8333	0.7800
Decision Tree	0.7857	0.6111	0.7021
Random Forest	0.8192	0.8611	0.8393
XGBoost	0.8281	0.5833	0.7109
SVM	0.7645	0.8611	0.8108

Table 4.1: Performance Metrics for 5 Different Models

ROC AUC is also important to demonstrate the model's overall ability to distinguish between positive and negative stroke instances. This can be viewed as a secondary metric to Recall i.e. once reaching a satisfactory recall, we look to optimise for ROC AUC as this is represents the trade-off between sensitivity and specificity.

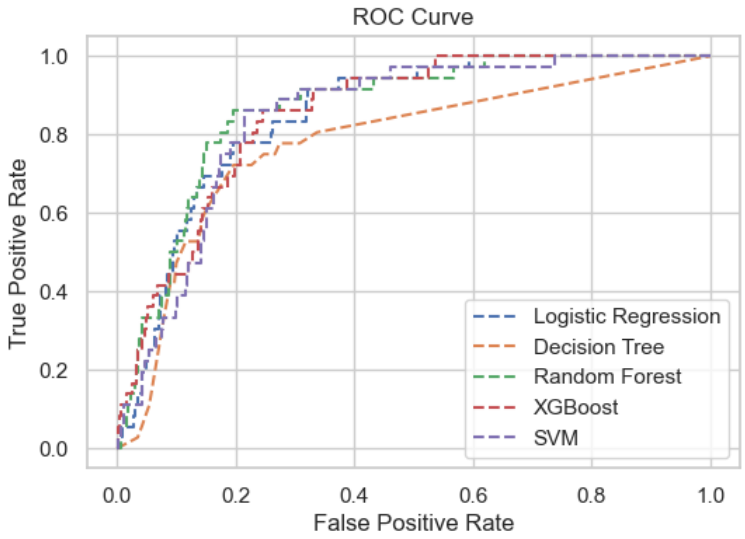

Figure 4.1: ROC Curve for 5 Different Models

4.2. Model Selection

We observe that a Random Forest Classifier has the best performance in terms of accuracy (81.92%), recall (86.11%) and ROC AUC (81.08%). Random Forests are typically effective at handling imbalanced datasets, as it is comprised of an ensemble of decision trees, which uses bootstrap sampling to counteract the class imbalance and improve model generalisation. Following hyperparameter tuning, we discover the best parameters are max depth (25), min samples leaf (9), min samples split (4) and n estimators (55).

	Predicted No Stroke	**Predicted Stroke**
Actual No Stroke	689 = TN	171 = FP
Actual Stroke	5 = FN	31 = TP

Table 4.2: Confusion Matrix for Random Forest Classifier

Viewing the confusion matrix for the Random Forest model's test set above, we observe 5 False Negatives which suggests a strong model, however with only 36 positive cases in the test set, one may suggest a high variance to this. However, with appropriate sampling (via undersampling and TOMEK oversampling) and 5-fold cross validation whilst hyperparameter tuning, this has improved the robustness of the test significantly such that there will be little variance among other test splits or new data.

4.3. Evaluation

We now compare our final Random Forest Classifier model versus our baseline Logistic Regression model, where we conducted no feature engineering in the form of outlier detection, feature reduction or feature normalisation. We conducted no sampling techniques, hyperparameter tuning or threshold evaluation.

We can quantify the impact of these procedures alone by using our previous logistic regression model as a middle ground, understanding that the baseline achieved a 95.21% accuracy yet 0.02% recall. This model would be impractical, and similar to the trivial

'model' of predicting no stroke in all cases. The vast improvement in recall within the logistic regression family, implies a significant impact of such feature engineering and ML design from the final model. With a decrease in accuracy, due to logistic regression's difficulty in handling imbalanced data set, comes the Random Forest Classifier to attempt to maintain accuracy with recall.

4.4. Improvements

This Random Forest Classifier performs well in terms of accuracy (81.92%) and recall (86.11%), however this is still far from hopes of 90-95% in both areas. More importantly, precision (and hence F1 Score) suffers heavily i.e. there are still a large number of false positive cases. In the above test set, we observe 171 false positives. Even with the aforementioned assertions to not allow these predictions to solely dictate the diagnosis of a patient, but rather inform and influence a healthcare professional to conduct further investigations on potential positive cases, this is still a significant proportion of cases to warrant a manual check for effectively all positive predicted cases. This counteracts the purpose of this machine learning model to improve overall efficiency of stroke diagnosis.

Model	**Accuracy**	**Recall**	**ROC AUC**
Logistic Regression (baseline)	0.9521	0.0200	0.5100
Logistic Regression	0.7310	0.8333	0.7800
Random Forest	0.8192	0.8611	0.8393

Table 4.3: Performance Metrics for Models versus baseline

To improve the model, we seek more data to specifically reduce the class imbalance within the data set. With only 4.9% of positive cases, combined with only 5110 patient records, we are limited with data to make robust predictions. The class imbalance further weakens the already small sample size, as we resort to undersampling of the majority class and hence a loss of data, outside of the existing anomaly detection. Larger datasets would also allow us to retain more features as we can allow for higher model complexity, rather than reducing to the final 4 features via feature importance.

In addition, more features (e.g. prior strokes, ethnicity, physical activity) could be beneficial even to have more choice to reduce features - in particular continuous numerical features as it was understood that these had the largest feature importance. Creating feature interactions and transformations as new features could also be beneficial (e.g. age * hypertension), to understand the interaction between multiple features even if the single variable has little feature importance.

Furthermore, a k-nearest neighbours algorithm or a neural network could've been explored as another family of model instead, to further understand how they handle imbalanced datasets after hyperparameter tuning. Finally, threshold tuning could've been conducted on a model by model basis, where different thresholds may have yielded varied results per model. A threshold of 0.5 proved overall satisfactory across all models, yet this could've been further explored and optimised.

5. CONCLUSION

Challenges of deploying such a model into clinical settings involve the high dependence on the quality and reliability of input data. Clinical data typically contains missing values,

which cause challenges if there are too many gaps as removing data or even machine learning algorithms will impact the model's performance or robustness.

This leads to ethical and legal considerations where private patient data is being used and also liability for malpractice (consequences of incorrect predictions) is potentially vague between the healthcare professional, the ML engineer or even the model itself.

Of course, clinicians and healthcare professionals should use the model's predictions as part of their diagnosis, alongside their own experience. However, in order for this to happen, they may require interpretable models which can explain their results. This is important for the suggested preventive measures e.g. knowing which features had the largest influence for a particular patient (e.g. high BMI, hence the patient should work to lose weight to prevent a stroke). ML models are typically 'black boxes' which make it difficult to understand and trust the predictions.

Overall, there are many practical challenges for implementing ML models within healthcare settings, even outside of the model's performance i.e. the ethical and subjective elements of trusting a model to make important decisions about an individual's health. However with the correct attention and willingness to implement ML predictions within existing clinical workflows, ML can be used to improve healthcare's operational efficiency.