Predicting Stroke Risk Using kNN and Random Forest Classifiers

Rachel Butcher

Group Size: 2

Table of individual contribution by each member of my group, based on my subjective opinion:

Student	Effort
Ynes Benotmane	50%
Rachel Butcher	50%

Introduction

In an increasingly aging world, insights into maladies that impact older people are of a greater importance than ever before, The WHO estimates that by the year 2050 the global population above the age of 60 will exceed 2 billion[1], this is in stark comparison to the figure of 900 million in 2015. It is thought that the majority of this aging population will occur in what WHO defines as low and middle-income countries. Given that these countries will often have limited economic resources, determining the most efficient system for their distribution must be conducted with an acute regard for wider social indicators.

It is widely accepted that advancing age is the most 'powerful independent factor' for determining Stroke risk. Stroke continues to rank in the top 3 causes of death across the world and is considered by the WHO as the leading cause of disability[2]. This not only has a significant impact at an individual level; causing distress to sufferer and their family, but also has a wider socio-economic impact.

Understanding the risk factors that lead to instances of Stroke would at the very least help countries across the income spectrum adjust and prepare health systems for this dramatic 'demographic shift' and at best allow them to introduce measures to reduce instances of the disease altogether,

The motivation of this project is to explore various machine-learning methods namely kNN and Random Forest to discover patterns in the population of a dataset, determine those individuals more at risk of developing a Stroke and the associated risk factors. In doing this analysis we hope to further extract indicators of those of greater risk and allow health professionals to use this information to intervene at earlier stages.

Literature Review

A stroke is a subset of cerebrovascular diseases in which blood vessels become unable to supply oxygen and nutrients to brain cells. It presents itself in one of two forms: ischemic and haemorrhagic. The former is more common and occurs when a blood clot obstructs a blood vessel in the brain causing the starvation and subsequent death of brain cells. Haemorrhagic strokes are caused by a brain aneurysm (bleed on the brain). While the two types of strokes have slightly different symptoms, their devastating effects are considered equivalent in the impact that they have on the patient and their family. While our dataset makes no distinction between the types of stroke, given the rarity of the haemorrhagic stroke, we will assume that the stroke instances were of the ischemic type[3].

Symptoms of a stroke commonly present themselves within seconds of the event and while there have been some advances with regard to post stroke treatment to restore neuroplasticity such as hydrogels[17], if the symptoms of stroke persist for longer than 24 hours they frequently leave the patient with severe disability or result in death.

Strokes are often accompanied by a variety of other health complications that prove to have a weighty impact on not only the sufferer but also their close family and wider community. Though the total deaths caused by strokes have continued to decrease in developed economies, with total deaths in the UK have fallen by almost 50% between 1990 and 2010, it continues to rank as one of the leading causes of severe disability within the UK[3].

As such, the rehabilitation often proves to be an expensive and intensive process that only serves to improve independence[4] and seldom returns the patient to their original level of health. Caplan cites increased wear on joints and bronchopulmonary infections as some of the frequent complaints[7] experienced post stroke.

As Caplan states, given that all major bodily organs exist to preserve the normal functioning of the brain, 'any change in the brain's function profoundly impacts living'[7]. The overreaching impact of a stroke is what drives this investigation, working to reduce the instances of stroke through the introduction of preventative measures will allow Governments to get ahead of what is a devastating disease. A theme that is echoed through a number of the texts consulted in this literature review is exemplified by Feigin in that 'The need to scale-up the primary prevention actions is urgent.'[5]

Data definitions

The dataset for this project is sourced from Kaggle. It provides relevant variables for classification of stroke and offers a large sample size (~43k) to mitigate risk of bias or overfitting. We are confident in the variables available as they corresponded with risk factors found in our literature review.
The data source has since been made private on Kaggle, previously available at:
https://www.kaggle.com/asaumya/healthcare-dataset-stroke-data

Dictionary of commonly used terms:

Term	Definition
Hypertension	Chronic high blood pressure that causes damage to blood vessels
Body Max Index (BMI)	A measurement of weight proportional to height, historically used as an indicator for health.
Blood glucose level	Concentration of sugar present in the blood. Normal is considered ~72-140mg/dL and chronic high levels can cause damage to blood vessels.
Heart disease	An umbrella term for several disorders that lead to the narrowing or blocking of blood vessels

Data Visualisation

Before undertaking the machine learning processes, creating a visual representation of our variables will assist in understanding how our data fits within the context of the background research we have conducted.

Overall our data confirms the conclusions made by the WHO with age being the greatest factor in determining stroke risk, with the majority of instances occurring for people above the age of 60 as displayed in the below box plot.

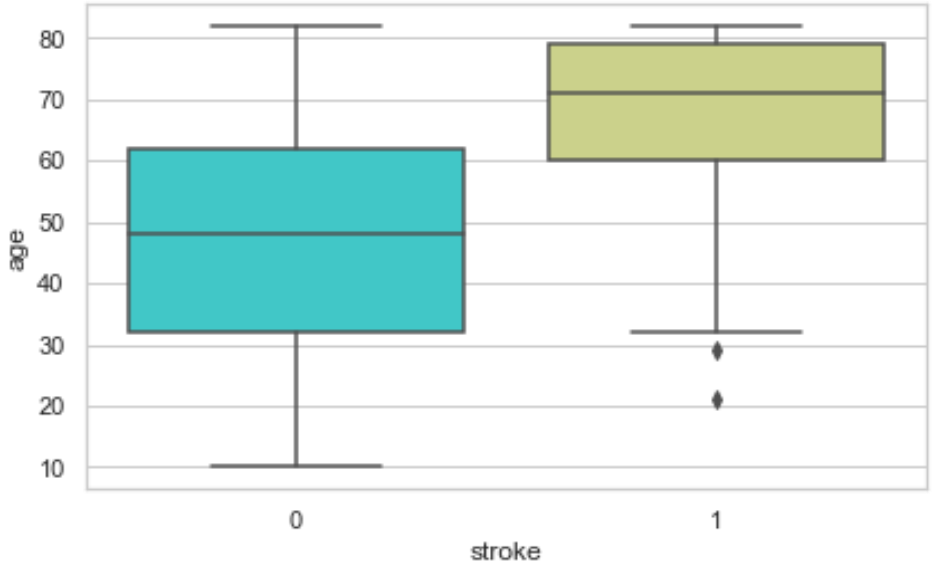


Figure 1>>> Age vs Stroke Boxplot

In the following figures, two of the risk factors identified by the WHO as significant indicators of stroke risk are illustrated using stacked bar graphs. It is clear that for those who were diagnosed with pre-existing health conditions (hypertension and heart disease respectively) there was a greater instance of stroke. The relation is especially obvious for heart disease where 8% of all those who experienced a stroke had been previously diagnosed with heart disease.

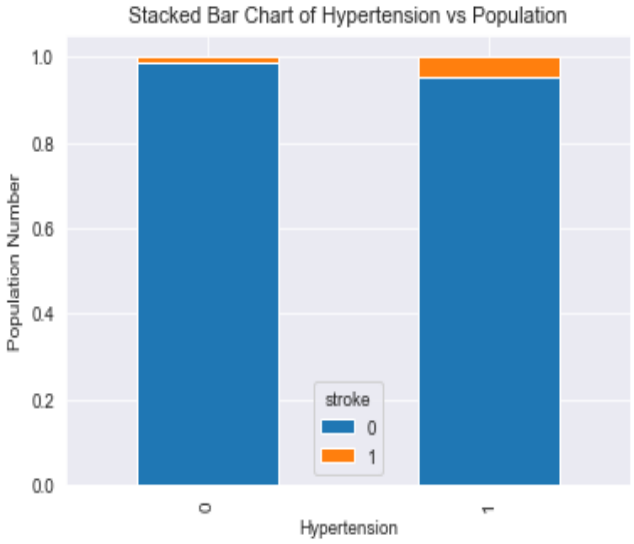


Figure 2>>> Stacked Bar Chart Hypertension vs Population

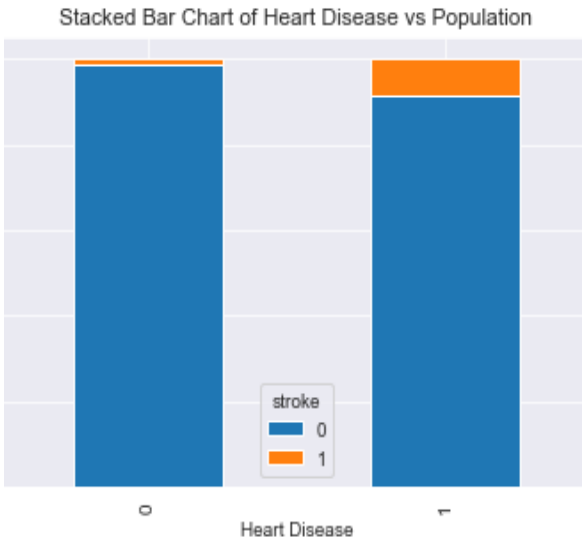


Figure 3>>> Stacked Bar Chart Heart Disease vs Population

There is a similar trend for those who smoke, with a noticeable increase in stroke instances for those who were identified as active smokers.

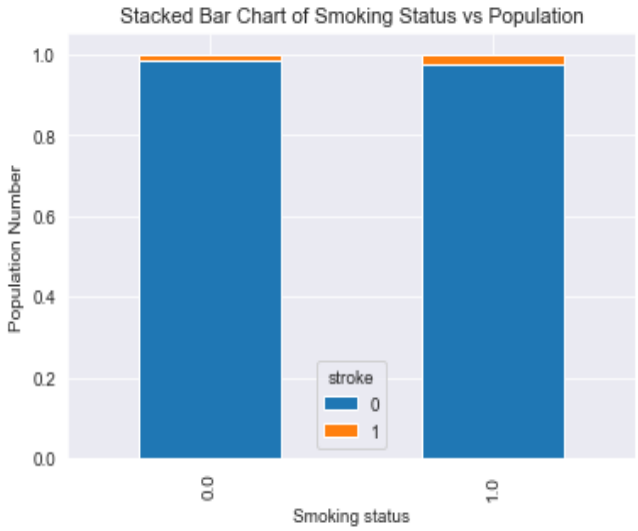


Figure 4>>> Stacked Bar Chart Smoking Status vs Population

Data pre-processing

We performed a number of cleaning functions on the raw Kaggle data. Initially, the ID column was removed as this is not a variable that contributes to stroke risk.

```
train.drop('id', axis=1, inplace=True)
```

The data consisted of a number of non-numeric categorical values, which required some modifications in order to make them usable in our computation.

Original Column	New Column	Values
Male/Female	gender_Male	1 = Male 0 = Female
Ever_married	ever_married_Yes	1 = Yes 0 = No
Work Type	work_type_Never_worked	1 = Yes 0 = No
Work Type	work_type_Private	1 = Yes 0 = No
Work Type	work_type_Self-employed	1 = Yes 0 = No
Work Type	work_type_children	1 = Yes 0 = No
Work Type	work_type_govt	1 = Yes 0 = No
Residence type	Residence_type_Urban	1 = Urban 0 = Rural
Smoking_status	smoking_status_never smoked	1 = Yes 0 = No
Smoking_status	smoking_status_smokes	1 = Yes 0 = No
Smoking_status	smoking_status_formerly_smoked	1 = Yes 0 = No

One-hot encoding (OHE) is a common technique for transforming categorical variables into numeric data, as many machine learning techniques only accept numeric data. OHE was used for the following variables: gender, work_type, ever_married, residence_type and smoking_status. For each possible category of the variable, OHE creates a binary dummy variable. Take our variable smoking status: the potential options are ‘smokes’, ‘never

smoked' or 'formerly smoked'. Using Pandas get_dummies function, the single categorical variable is represented by 3 binary indicator variables (one for each possible category). [15]

```
smoking_dummies = pd.get_dummies(dummy_train.smoking_status, prefix='smoking_status').iloc[:, 1:]
```

The indicator variable to which the patient belongs is populated with a 1 and the other indicators populated with 0, i.e. a smoker would have the following:

Smoking status: smokes	Smoking status: never smoked	Smoking status: formerly smoked
1	0	0

When inspecting the data, the get_dummies function has only produced two indicator columns, as it can be inferred that a 0 value in both represents the column not presented. OHE can represent the whole information using 1 less dimension i.e. n-1 where n=possible variable categories.

The get_dummies function creates a separate data frame with which the original data set must be concatenated. Finally, the original categorical column should be dropped from the combined data frame to prevent duplicates. We have achieved this with the following command:

```
pd.concat([dummy_train, smoking_dummies], axis=1)
clean_train = pd.get_dummies(dummy_train, columns=['smoking_status'], drop_first=True)
```

A frequently reported limitation of OHE in Random Forest Classifiers is that it offers few options for splitting. This results in sparse decision trees as for each dummy variable there becomes only two possible values (0 or 1).

Missing Values

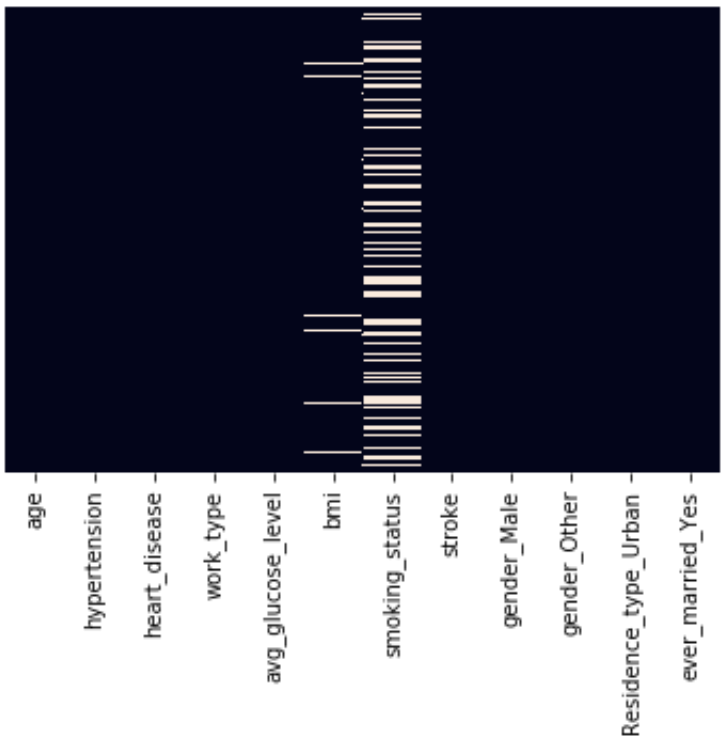


Figure 5>>> Heatmap identifying missing value amount per column

We used the isnull() function to determine the raw data was missing 30.6% of the values for smoking status and 3.4% of values for BMI.

Given the few missing entries for BMI, we decided to replace them with the mean value of the total dataset. We felt that this would be a suitable replacement value for the omitted results given the continuous and limited nature of the BMI; it would be unlikely to have extreme results skewing the mean by a large amount. This would also be a lesser forfeiture than removing those entries completely from our dataset.

For the missing smoking status entries, we assumed that those within the dataset with missing smoking status would have been non-smokers. According to the Office for National Statistics, the proportion of the UK population that were identified as active smokers in 2018 was 14.7%.[7] This is not too dissimilar to our dataset where those marked as smokers made up 15.1% of our total.

Methodology

To perform these machine learning processes, we decided that supervised classification analysis would be best suited for our dataset; it would not make sense to perform unsupervised learning techniques as our dataset is labelled. We have decided to perform the kNN algorithm and Random Forest. Both methods will be evaluated through accuracy score testing and confusion matrix analysis. The confusion matrix is an illustrative table which classifies each instance of predicted and actual points in a count of true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

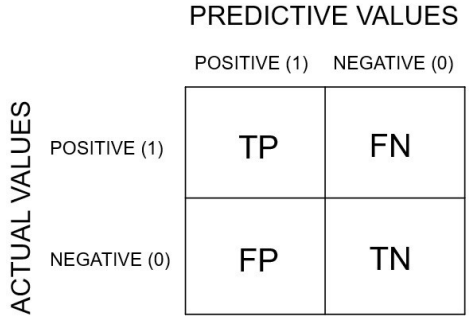


Figure 6>>> Classification Matrix[9]

The accuracy score of a model is essentially the count of true positives and true negatives classified correctly.

kNN Algorithm Background

This is a classification algorithm that performs on a very simple principle, data points that are within close proximity are assumed to be of the same type. The proximity is determined by the Euclidean distance of the points and the 'k' value initialised in order to determine the number of neighbouring points.
Implementation of this algorithm is fairly straightforward and produces results that represent the data in a clear way.

However, a frequently reported limitation of the kNN algorithm is the slow processing when the datasets increases. This would be a significant disadvantage of using this technique if we were ever to extend this project as it would prove computationally expensive. Another difficulty is found within the selection of the 'k' value; a small value of k would produce a large amount of noise within the dataset results and impact their integrity. A large value of k would offset the original aim of the algorithm in grouping many points within the dataset together. In order to mitigate for this, we implemented the 'Elbow method' which allows you to iterate through various values of k and determine which value produces the smallest error rate. It is so called for the shape that is created when the graph is plotted, small values of k create a high level of error which stabilises as the values get larger creating a sort of L or elbow shape.

kNN Algorithm implementation

The kNN algorithm is found within the Sci-Kit Library in python 3 and is imported and implemented using the following commands,

```
from sklearn.neighbors import KNeighborsClassifier
X = clean_trains.drop("stroke",axis=1)
y = clean_trains["stroke"]
X_train,X_test,y_train,y_test= train_test_split(X,y,test_size=0.25,random_state=101)
```

The X value would behave as our independent variables, this was made up of all of our variables except that of stroke which would be predicted for in our dependent variable y. The parameter test_size outlines how much of our data is retained for the final testing of the model. We have set this to be 25% of the total and have kept this split consistent amongst our various models in order to allow for fair comparison.

When initially fitting the model, we overwrote the default of n_neighbours to 1 with a view to increase this value post evaluation of the confusion matrix and cross validation. All other values within the KNeighborsClassifer were kept as their default.

```
knn_model = KNeighborsClassifier(n_neighbors=1)
knn_model.fit(X_train,y_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=1, p=2,
                     weights='uniform')
```

The knn_model.fit(X_train,y_train) fit function in the sklearn adjusts the weights of each of the variables in preparation for the training of the data and the later testing.

Results

	precision	recall	f1-score	support
0	0.98	0.98	0.98	7350
1	0.08	0.06	0.06	177
accuracy			0.96	7527
macro avg	0.53	0.52	0.52	7527
weighted avg	0.96	0.96	0.96	7527

```
Accuracy : 96.14720340108941
```

Figure 7>>> Confusion Matrix for kNN

Initially the model performed quite well with an initial accuracy score of 96.15%. The confusion matrix illustrates how many of each instance are predicted correctly. This stays

consistent after conducting cross validation testing which produces values that remain around the 96% mark.

```
X1,X2,y1,y2 = train_test_split(X,y,random_state=0,train_size=0.2)
y2_model = knn_model.fit(X1,y1).predict(X2)
y1_model = knn_model.fit(X2,y2).predict(X1)

print(accuracy_score(y1,y1_model))
print(accuracy_score(y2,y2_model))
```

```
0.9599734263411394
0.9638809316228671
```

Improvement to Accuracy

In order to further increase the accuracy of the model, we will perform the elbow method to coax the optimum value of k for our data set. It was conducted by implementing a loop that iterates through values of k ranging from 1 to 20 and producing the error rate for each. Plotting this information produces the following graph.

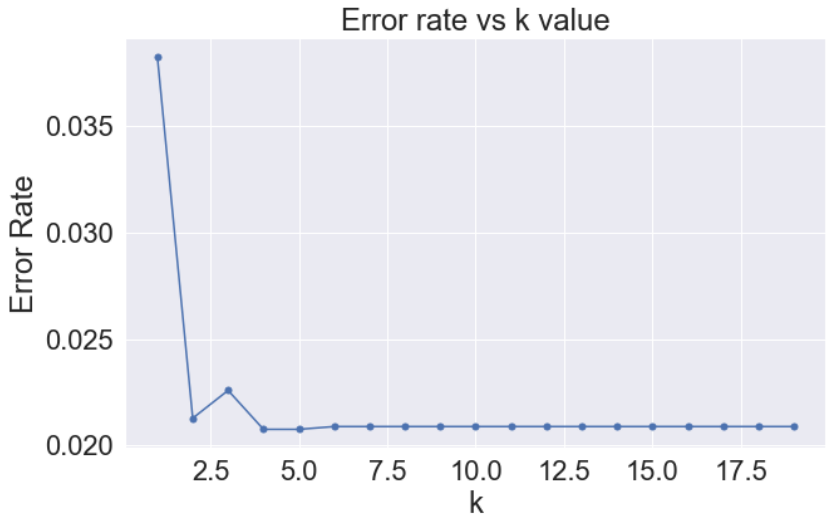
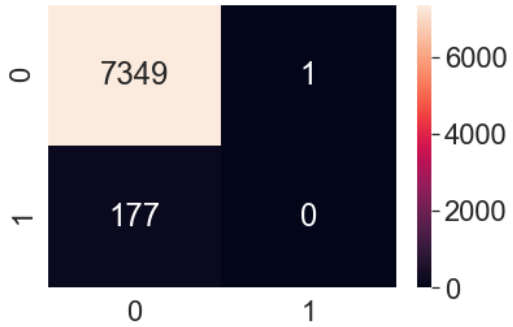


Figure 8>>> Elbow graph for Stroke kNN

It is clear from this plot that for any values of k above 3.75, the error rate stabilises and does not reach the heights that were present in our original model. As previously stated, selecting a value of k that is too high would negate the benefit of the model in grouping too many values within the same 'neighbourhood'. So, we take the value of k for which the error rate remains consistent, 5 and re-run the model.

```
Accuracy : 97.63518001859971
[[7349    1]
 [ 177    0]]
```



This adjustment in the n_neighbors parameter has a significant impact on the accuracy of the model, raising it by almost 2% in comparison to the initial reading. Performing the Confusion Matrix once more illustrates that the though the accuracy has increased and we have more people identified as negative for stroke, we have lost the population of True Negatives i.e. the population of those who were indeed positive for Stroke.

Random Forest Classification Background

Random Forest Classifier is an ensemble algorithm consisting of a 'forest' of decision trees. This technique will function as a binary classifier to predict whether a patient will suffer from a stroke. The algorithm forms decision trees from randomly selected subsets of data with replacement. The results of individual trees are aggregated to produce the final decision output of the forest.

Whilst ensemble algorithms often benefit from a lower variance, they suffer from increased complexity. By default, Python sklearn library creates 100 decision trees, requiring relatively more computational resources and a longer training period.

Algorithm implementation

The Random Forest Classification algorithm is found within the Sci-Kit Library in Python 3 and is imported and implemented using the following commands:

```
from sklearn import model_selection
from sklearn.model_selection import train_test_split
from sklearn.ensemble.forest import RandomForestClassifier
X = clean_train.drop('stroke', axis=1)
y = clean_train['stroke']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=66)
```

X is the collection of predictor variables which we will be using to predict our target variable, y (stroke). We have chosen a 1:3 train-test split, which has been kept a constant across both machine learning algorithms.

Initially all values within RandomForestClassifier were kept as default. Namely, the n_estimators parameter has not been specified and so the number of trees will take the default value of 100. Similarly, the bootstrap parameter is defaulted to True; therefore, trees are created with replacement rather than using the whole dataset to build each tree. This helps to reduce variance and avoid overfitting.

```
rfc = RandomForestClassifier()
rfc.fit(X_train, y_train)
rfc_predict = rfc.predict(X_test)
```

Cross-validation was implemented to estimate the skill of the classifier. It is a popular function for assessing the model's prediction performance on data not used during training. We set the k parameter to 10, as this number of splits is generally accepted to optimize the bias-variance trade-off. [12]

```
rfc_cv_score = cross_val_score(rfc, X, y, cv=10, scoring='roc_auc')
```

Results

Initially the model performed extremely well, with an f1-score of 0.99 and high true positive value in the Confusion Matrix. The f1-score can be considered a measure of the model's accuracy as it is derived from the harmonic mean of the recall and precision values. The AUC score, however, did not share the same success. Initially the mean AUC score achieved 0.64. This is only marginally greater than a score of 0.5, which would mean the model had no capacity to discriminate between stroke classifications. [13]

```
=== Confusion Matrix ===
[[10608     7]
 [  234     1]]

=== Classification Report ===
              precision    recall  f1-score   support

           0       0.98      1.00      0.99     10615
           1       0.12      0.00      0.01       235

    accuracy                           0.98     10850
   macro avg       0.55      0.50      0.50     10850
weighted avg       0.96      0.98      0.97     10850

 === All AUC Scores ===
 [0.64393462 0.64951232 0.64305698 0.62098569 0.65562244 0.64652745
  0.60242573 0.61530338 0.67917878 0.6697206 ]


 === Mean AUC Score ===
 Mean AUC Score - Random Forest:  0.642626799765438
```

Improvement to AUC score

It is widely accepted that increasing the number of decision trees will improve the robustness of the algorithm. We tested this by doubling the number of trees from the default 100, to 200. The mean AUC score was brought closer to 1 from 0.63 to 0.79, demonstrating an increase of ~25%. The use of multiple trees reduces our risk of overfitting, however computational time was noticed to significantly increase.

```
 === All AUC Scores ===
 [0.80308021 0.7807902  0.75058509 0.80024727 0.7879261  0.79343543
  0.79382798 0.8076216  0.8226235  0.79878775]


 === Mean AUC Score ===
 Mean AUC Score - Random Forest:  0.7938925132041116
```

Feature importance

Feature importance is a Pandas function that allows for evaluating the relevance of each variable to the model output. Plotting a histogram of these calculated values immediately emphasizes the importance of age, BMI and average glucose level. From this we may extrapolate that these variables are the greatest indicators for stroke risk.

As previously mentioned, OHE displays n-1 indicator variables. A limitation we have experienced when combining this with the feature importance function is that the inferred variables seem to have been discounted. The histogram only presents importance scores for n-1 indicator variables rather than n. Another limitation is that we are unable to take advantage of this functions ability to identify variables that do not contribute to the model. The variables are no longer fully independent of each other. We could not remove the lowest scoring variable (work_type_never_worked) as it dependent on several higher scoring ones.
[14]

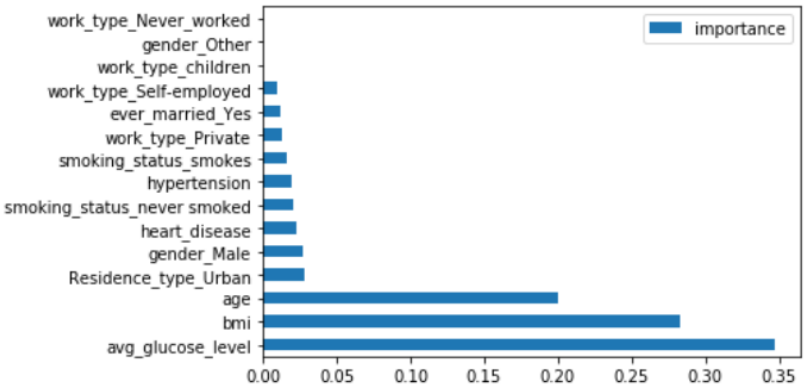


Figure 8>>> Feature Importance for Stroke Dataset

Conclusion and Evaluation

Overall both of the algorithms that were implemented on our Stroke dataset performed well with accuracy levels >90% and helped us to achieve our objective of classifying individuals at Stroke risk based on both medical indicators and socio-economic factors. As a simpler machine learning technique, kNN offered a suitable baseline from which to compare our more complex Random Forest algorithm. However, repeating this analysis with a larger dataset, the Random Forest algorithm should be considered superior to the kNNeighbors given that the latter does not scale well; requiring significant runtime and memory to be completed. It would also be useful to complement the Random Forest Analysis with a more sophisticated technique to allow for cross evaluation of the results; support vector machine analysis would be a good candidate by virtue of its strong accuracy.

The fundamental limitation of our project has certainly been the fact that we have not had expert guidance/knowledge. While this has been somewhat mitigated through our extensive background reading, and by the adjustments made to the missing values by comparing to the recorded ONS values, this is no replacement for expert advice and should be considered as an improvement when conducting this project again in the future.

An interesting outcome of our analysis is the conclusion drawn from the feature analysis which suggests that the average glucose level plays a more significant role in determining stroke risk than age, an idea seldom presented in scientific literature on Stroke. This would be an area of interest to build on in future analysis with larger datasets.

Stroke Dataset Sample: https://github.com/RachelButcher/Data-Analytics-ECS748/blob/master/StrokeDatasetSample.xlsx
Stroke Data Analysis: https://github.com/Yb-crypto/Data-Analytics-ECS748/blob/master/Stroke%20Data%20Analysis.ipynb
Random Forest Python Code: https://github.com/RachelButcher/Data-Analytics-ECS748/blob/master/Stroke_Random_Forest.ipynb
KNN Python Code: https://github.com/Yb-crypto/Data-Analytics-ECS748/blob/master/KNN_Model_DA.ipynb
Structural Learning Analysis: https://github.com/RachelButcher/Data-Analytics-ECS748

Bayesian Network Structure Learning Analysis
Missing Values

We assumed 'Non-Smoker' for missing values in smoking status, as per our earlier justification, and 'Normal' for missing BMI values.

Discretisation

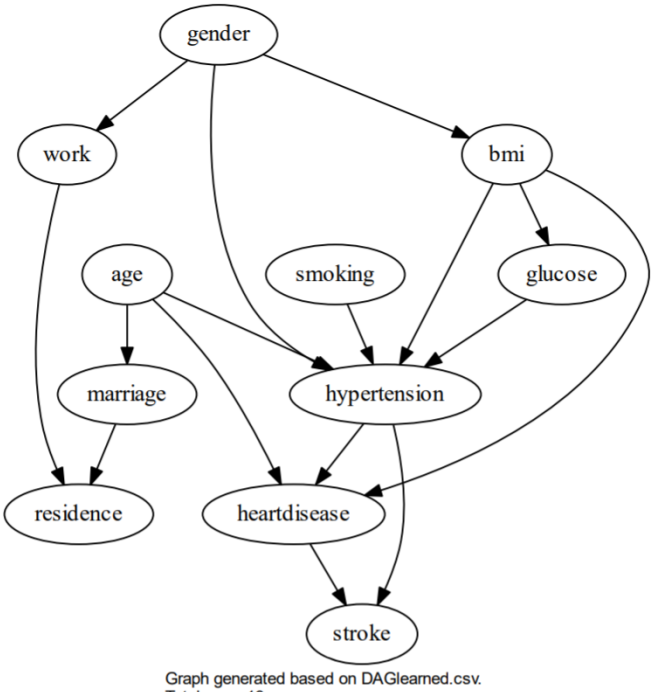
Continuous variables were discretized as follows:

Age	Avg Glucose Level (per diabetes.co.uk)	BMI (per NHS guidelines)
1-20	Low	Low = <18.5
21-40	Normal	Normal = 18.5 - 24.99
41-60	High	Overweight = 25 – 29.99
61-80		Obese = >30
81-100		

Questions

_____ Evaluation _____

Nodes: 11
Sample size: 43400
TrueDAG arcs: 16
TrueDAG independencies: 39
LearnedDAG arcs: 16
LearnedDAG independencies: 39

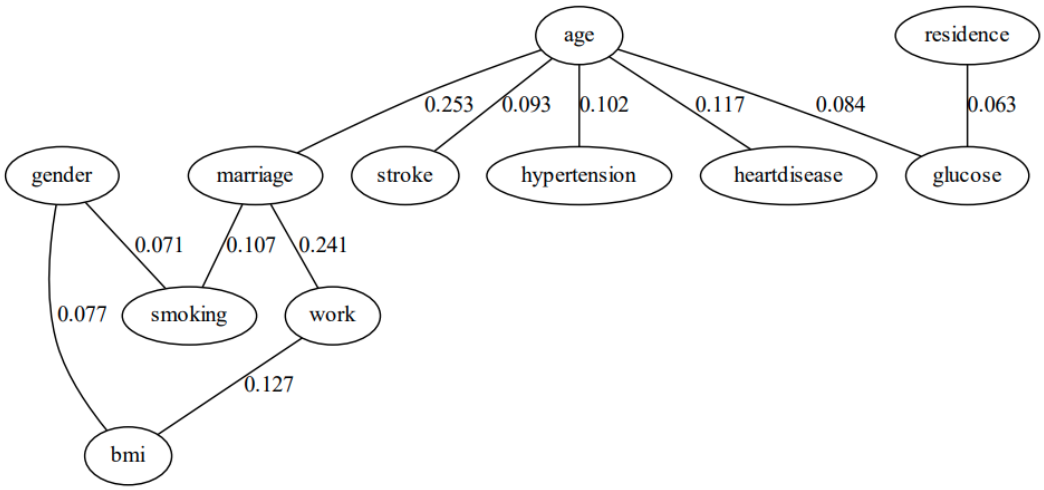


1.Producing a Knowledge Based Graph

An initial structure for the graph was elicited using primary intuition and experience. We then reinforced or updated our causal structure with conclusions from scientific literature[18, 19, 20, 21, 22]. Disputes or counterintuitive decisions were resolved and supported by this research. For example, risk of heart disease is generally accepted to be lower for women, but this is only as they tend to live longer. When age is accounted for, heart disease risk converges to equal in males and females. Therefore, we made the decision to remove the edge between gender and heart disease[23]. It was also found that a number of variables caused hypertension, which in turn increased risk of stroke, rather than having direct causality. Edges to stroke from these

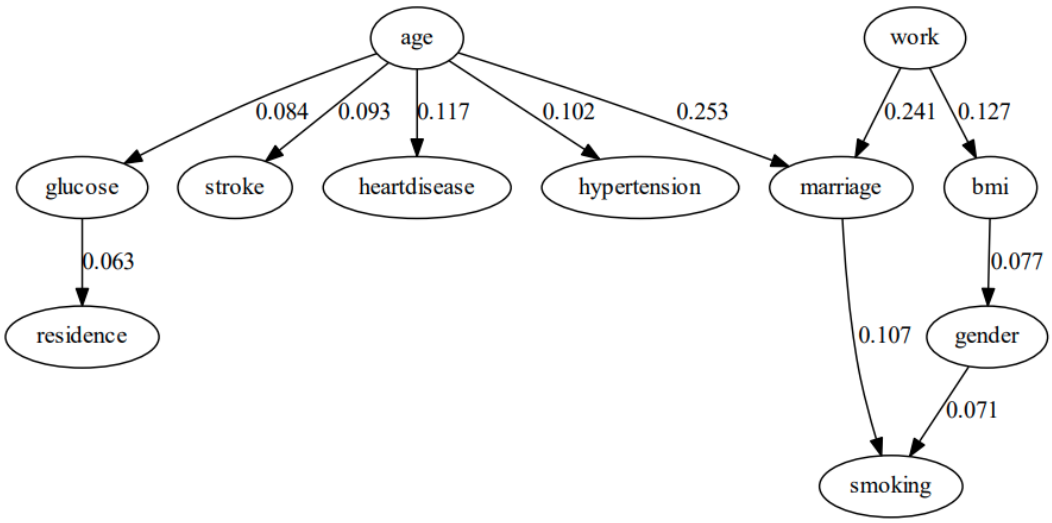
variables, i.e. smoking and BMI, were removed. The final structure was agreed by all members of the group.

Phase 1



SaiyanH_Phase_1 EMSG graph.
Total edges: 11

Phase 2



SaiyanH_Phase_2 EMSG directed graph.
Total arcs: 11

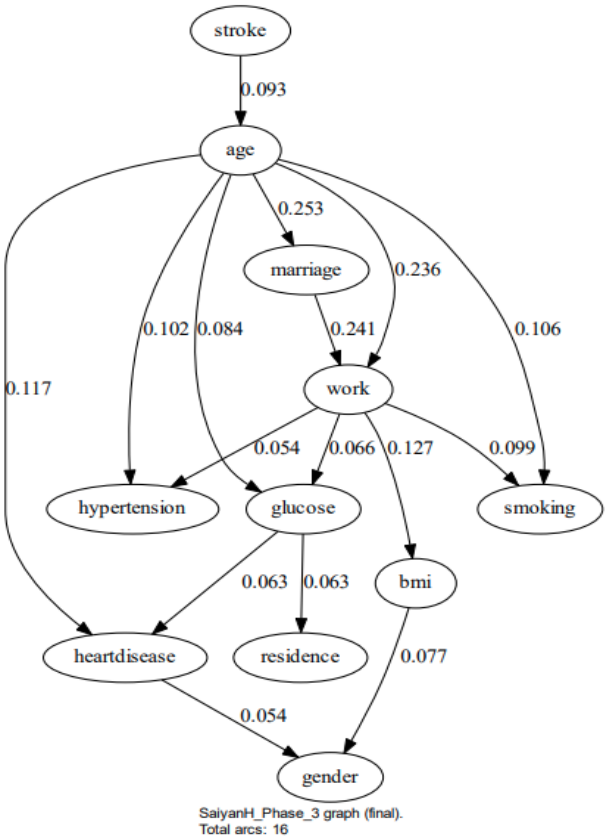
---

2. Phase_1 to Phase_2

The Phase_1 is the best guess for the associations between each of the variables. In this stage the associations are undirected; the causal relation is known to exist but the orientation of the edge is yet unknown to the network. The figures presented at each of the nodes are the MMD scores which indicate the dependency between the two variables, the higher the value, the greater the dependency. The Phase_2 graph classifies three variables (a pair of nodes and conditional on an independent node) through the application of conditional

dependence/independence tests. The principal difference between the two phase graphs is the orientation of the edges which is determined through this constraint-based testing and classification.

3. Phase_2 to Phase_3



The Phase_3 graph employs the classic machine learning technique of score-based learning in which the algorithm searches for the graphs which best align the fitting distribution with the empirical distribution. It uses the Phase_2 graph as its starting point in the Hill Climbing search method to find graphs that improve the BIC score of the respective graph through the addition, removal and inversing the orientation of arcs. In our dataset, the Phase_3 graph identifies Stroke as the fundamental causal variable. This is in contrast with what was reflected in the Phase_2 graph and is somewhat contrary to what we, using our knowledge would consider as factual.

4. Scores Total

conditionalDependency = 35 Scores
marginalDependency = 55 Scores
conditionalIndependence = 20 Scores
conditionalInsignificance = 440 Scores

The likely reason that the conditional insignificance score exceeds that of both the conditional independence and dependence scores is as a result of the 50% threshold in the conditional testing. The majority of the relations between each of the MMD scores in the triple testing are not extreme enough to merit falling into the independence/dependence classification. The levels across the other scores are fairly evenly distributed.

5. Score Comparisons

_____ Stats from metrics and scoring functions _____
Precision score: 0.219
Recall score: 0.219
F1 score: 0.219
SHD score: 24.500
DDM score: -1.313
BSF score: -0.089
# of independent graphical fragments: 1

In general, our F1, SHD and BSF scores are lower than those of Fig 2 at roughly 43k sample size. In relation to nodes and arcs, our data is closest to the Sports case study (11 nodes, 16 arcs and 9 nodes, 15 arcs respectively) yet the latter has an F1 and BSF score almost 4x closer to 1. However, this accuracy may be explained by the large number of free parameters in the Sports study (1049). A BSF score of -0.089 is close to the most inaccurate learned graph possible. This seems surprising compared to the case studies as it is the only negative value for this score type, but we can see the poor dependency discovery (i.e. stroke causing age) when compared to the knowledge graph and therefore this score was to be expected.

6. Time

The total runtime for the Phase_4 of the Stroke dataset was a total build time of 29 seconds. For a sample size of 43,400 and 250 free parameters, the processing took longer when compared to other similar sized datasets. The nearest example in Table 2 is the 13th entry of the table with 9 nodes, 15 true edges and a sample size of 100k. This example took only 10 seconds to complete even though it had 4x the number of free parameters. This disparity in runtime can almost certainly be attributed to the dimensionality of the variables that we have in the Stroke dataset; 3 for ‘Glucose’, 4 for ‘BMI’ and 5 for the ‘age’ variable. This would likely increase the time spent in Phase_2 of the process and the constraint-based analysis where each of these ‘states’ would have to be accounted for.

7. Phase BIC/MDL Score

The BIC score in Step 4 was -414452.377 in comparison to Step 3’s score of -462768.89, a percentage increase of 10%. The BIC is a scoring function which helps to establish the orientation of an edge, if the BIC is increased by the reversal of an edge, then the orientation is set. The aim is to improve the BIC score as much as possible, a step which is conducted in Phase_3. It would make sense that the BIC score is improved from Step 3 to Step 4 by virtue of the fact that the number of free parameters is reduced with the introduction with more arcs, and as such the penalty term is reduced.

8. Free Parameters

The number of free parameters demonstrated ~60% percent decrease from Step 3 (624) to Step 4 (250). Free parameters are those not pre-defined by the model and that must be estimated. Increasing the number of free parameters can appear to improve the fit of a model, but the model is less likely to reflect reality due to the greater proportion of estimates. Usually it is desirable to keep the number of free parameters to a minimum. A reduction in free parameters of the Step 4 model is understandable considering the inclusion of more constraints.

References:

[1] WHO, Who.int. 2020. Ageing And Health. [online] Available at: <https://www.who.int/news-room/fact-sheets/detail/ageing-and-health> [Accessed 12 April 2020].

[2] Walter Johnson, Oyere Onuma, Mayowa Owolabi & Sonal Sachdev,2016, Stroke: a global response is needed. Bulletin of the World Health Organization 2016;94:634-634A.

[3] Donnan GA, Fisher M, Madeod M, Davis SM. Stroke [Seminar]. Lancet,2008, 371 :p.1612-23.

[4] Adamson, J., Beswick, A. and Ebrahim, S., 2004. Is stroke the most common cause of disability?. Journal of stroke and cerebrovascular diseases, 13(4), pp.171-177.

[5] Dombovy, M.L., Sandok, B.A. and Basford, J.R., 1986. Rehabilitation for stroke: a review. Stroke, 17(3), pp.363-369.

[7] Feigin, V.L., Roth, G.A., Naghavi, M., Parmar, P., Krishnamurthi, R., Chugh, S., Mensah, G.A., Norrving, B., Shiue, I., Ng, M. and Estep, K., 2016. Global burden of stroke and risk factors in 188 countries, during 1990–2013. The Lancet Neurology, 15(9), pp.913-924.

[8] Strong, K., Mathers, C. and Bonita, R., 2007. Preventing stroke: saving lives around the world. The Lancet Neurology, 6(2), pp.182-187.

[9] Caplan, L.R. ed., 2016. Caplan's stroke. Cambridge University Press.

[10] Ons.gov.uk. 2018. Adult Smoking Habits In The UK - Office For National Statistics. [online] <https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/healthandlifeexpectancies/bulletins/adultsmokinghabitsingreatbritain/2018> [Accessed 12 April 2020].

[11] Medium. 2020. Decoding The Confusion Matrix. [online] <https://towardsdatascience.com/decoding-the-confusion-matrix-bb4801decbb> [Accessed 18 April 2020].

[12] Machine Learning Mastery. 2018. A Gentle Introduction to k-fold Cross-Validation. [online] <https://machinelearningmastery.com/k-fold-cross-validation/> [Accessed 20 April 2020].

[13] Medium. 2018. Understanding AUC - ROC Curve. [online] <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5> [Accessed 20 April 2020].

[14] Medium. 2018. Feature Selection Techniques in Machine Learning with Python. [online] <https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e> [Accessed 20 April 2020].

[15] Machine Learning Mastery. 2017. Why One-Hot Encode Data in Machine Learning? [online] <https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/> [Accessed 20 April 2020].

[16] Obese, H.W.O., 1998. Body Mass Index (BMI). Obesity Research, 6(2), pp.51S-209S.

[17] Nih, Lina Ratiba et al. “Hydrogels for brain repair after stroke” Current opinion in biotechnology vol. 40 (2016): 155-163. doi:10.1016/j.copbio.2016.04.021

[18] P Primatesta, E Falaschetti, S Gupta, M G. Marmot, and N R. Poulter (2001) 'Association Between Smoking and Blood Pressure', *AHA Journals,* 37(), pp. 187-193.

[19] I S. Ockene, N Houston Miller (1997) 'Cigarette Smoking, Cardiovascular Disease, and Stroke', *AHA Journals,* 96(9), pp. 3243–3247.

[20] J Dubow & M E Fink (2011) 'Impact of Hypertension on Stroke', *Current Atherosclerosis Reports,* 13(), pp. 298–305.

[21] N Agrawall , M Kumar Agrawal, T Kumari, S Kumar (2017) 'Correlation between Body Mass Index and Blood Glucose Levels in ', *International Journal of Contemporary Medical Research,* 4(8), pp. 1633-1636.

[22] W B Kannel, W F Wilson, T Zhang (1991) 'The epidemiology of impaired glucose tolerance and hypertension', *American Heart Journal,* 121(4), pp. 1268-1273.

[23] A.H.E.M. Maas and Y.E.A. Appelman (2010) 'Gender differences in coronary heart disease', *Netherlands Heart Journal,* 18(12), pp. 598-602.