Comparing Decision Tree and K-Nearest Neighbor for Cerebral Stroke Prediction

by Stefan Diener

1) Description and motivation of the problem

In 2020, cerebrovascular diseases were the fourth leading cause of death in England [1]. A cerebral stroke is a subtype of these diseases in which the blood supply to part of the brain is interrupted, depriving brain tissue of oxygen and nutrients, which causes the death of brain cells within minutes. Therefore, predicting whether a person will have a cerebral stroke is of great importance to public health, as it can help ensure adequate preparation and quick action in an emergency. The goal of this project is to train and compare the ability of a Decision Tree (DT) and a K-Nearest Neighbor (KNN) classifier to predict whether a person will suffer a stroke. This extends the work of Lui, Fan, and Wu (2019) because it explores the potential of two models that were not analyzed in their original work, and because the DT and KNN models are more interpretable and intuitive, which is an important property for medical applications.

2) Initial analysis of the data set including basic statistics

- The medical data set on cerebral strokes is supplement to Lui, Fan and Wu (2019). It was published on data.mendeley.com [2] and contains 43'000 observations of potential patients.
- It includes one binary target variable indicating whether a patient had a stroke and 11 features on the patient's physiology, including five binary, three continuous and two categorical variables.
- The categorical variables were one-hot encoded, in order to facilitate the calculation of distance for the KNN algorithm.
- The data is characterized by class imbalance in the target variable, with only 738 records of cerebral stroke (figure 1), and by missing values mainly in the 'Smoking Status' feature (figure 4).
- Missing Values: 1'457 observations with missing 'BMI' values were discarded, as they only amount to 3.3% of the data set. The 30.6% of missing records of 'Smoking Status' are treated as a distinct 'Unknown' category. This is done because I suspect the variable to be not completely missing at random, e.g., because smokers might not want to disclose their bad habit. In this case, an unknown category allows the model to capture this information, compared to imputing by the majority category or discarding the column all together.
- Standardization: The data is not standardized, because it is not required for the DT model and because MATLAB's KNN implementation, has built-in standardization.
- The classic summary statistics table is omitted because the figures 1-5 provide deeper insight into variable distribution. Additionally, tables 1-3 give an overview of the possible values.
- Among the binary variables 'Hypertension' and 'Heart Disease' show the largest shift in distribution between the 'stroke' and 'no stroke' patients. Among the continuous variables the 'Age' distribution is most different for the two patient groups. For categorical variables, 'stoke' is more common for 'self-employed' workers and 'former' smokers, counterintuitively the difference is smaller for 'active' smokers. The 'unknown' smoking status is about twice as common among 'no stroke' patients.
- The corelation matrix, shows 'Age' to have the highest correlation with stroke and most other variables. For easier visual representation, the categorical variables are ordinally encoded in the matrix as follows: Smoking: Never=0, Former=1, Active=2 Work Type: Child=0, Govt.=1, Private=2, Self-Emp=3.

Binary Variables	Values	Binary Variables 2	Values
Stroke	Yes/No	Heart Disease	Yes/No
Gender	Male/Female	Ever Married	Yes/No
Hypertension	Yes/No	Urban Home	Yes/No

Table 1: Overview Binary Variables

Cont. Variables	Values
Age	0 – 82
Avg. Glucose	55 – 291
BMI	10.1 - 97.6

Table 2: Overview Continuous Variables

Cat. Variables	Values
Smoker	Never, Former, Active, Unknown
Work Type	Child, Private, Self-Emp., Govt.

Table 3: Overview Categorical Variables

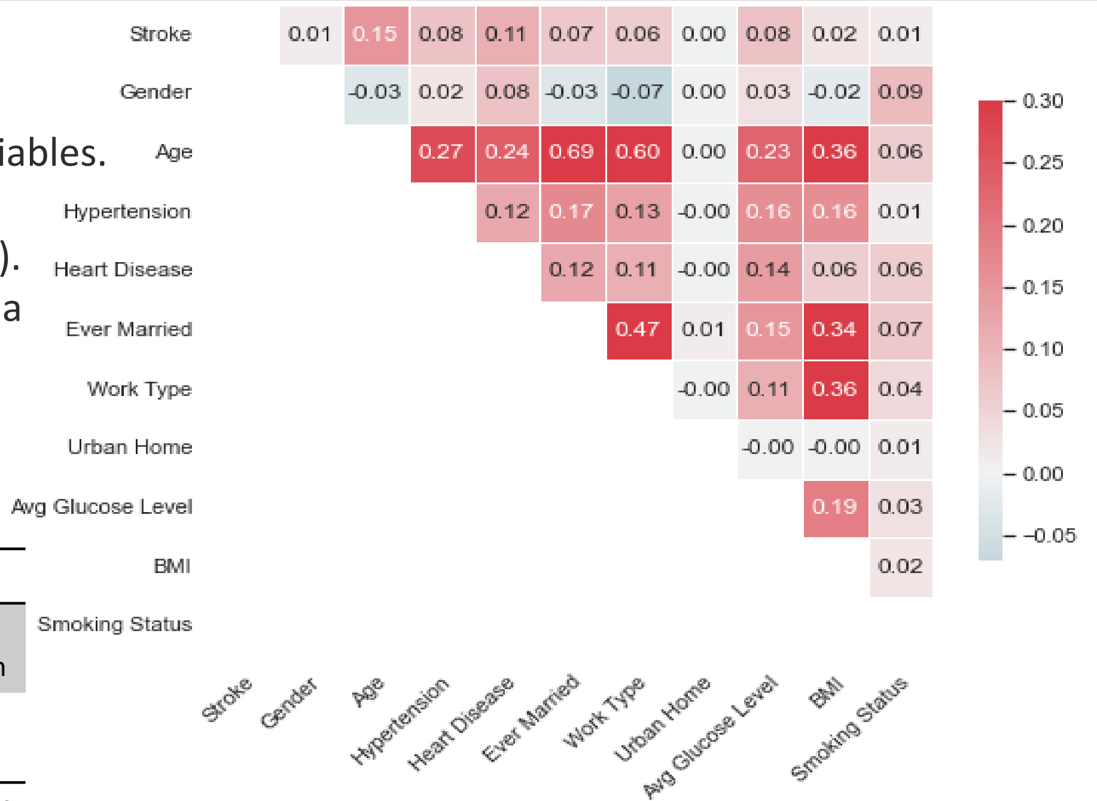


Figure 5: Correlation Matrix

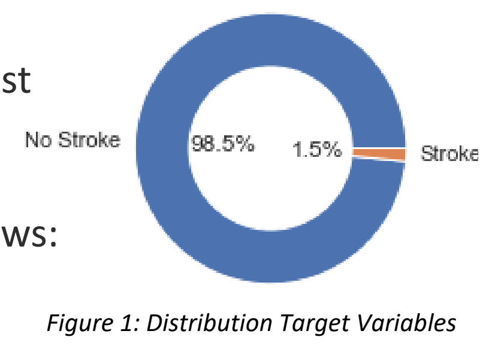


Figure 1: Distribution Target Variables

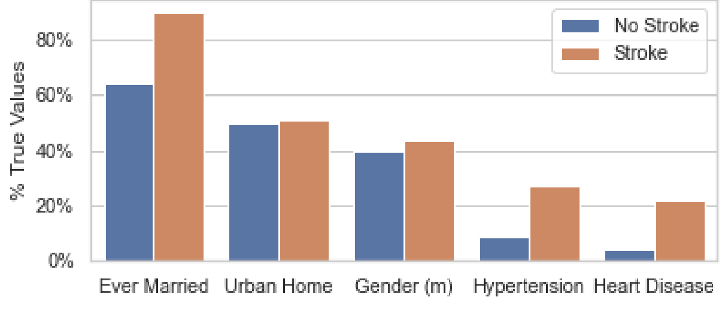


Figure 2: Distribution Binary Variables

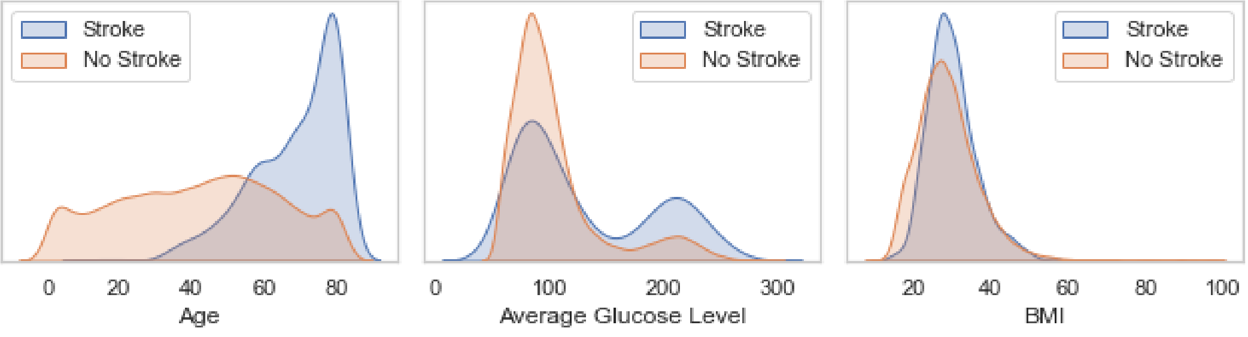


Figure 3: Distribution Continuous Variables

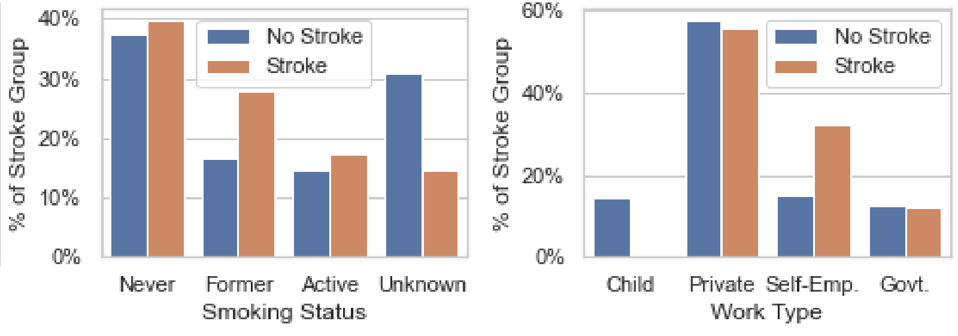


Figure 4: Distribution Categorical Variables

3) a) Model description - Decision Tree (DT)

- A DT is a non-parametric supervised learning model for classification and regression. [3]
- DTs aim to predict the value of the target variable by learning simple decision rules based on the input variables that recursively divide the data into mutually exclusive sup-groups, called leaves. [3]
- Starting from the entire sample (root), a top-down greedy algorithm, called recursive partitioning, considers all possible features and cut off points, and chooses the one that maximizes a specified optimization criterion. [3]
- Additional splits are successively added in that same fashion, until some stopping criterion is reached. [3]

Pros

- White box model with high interpretability. [4]
- Can easily be displayed graphically. [5]
- Handles numerical, categorical, outliers, missing values, and irrelevant features well. [6]
- Insensitive to monotone transformations of inputs. [6]
- Few assumptions about the data (non-parametric). [5]

Cons

- Tends to overfit the training data (high variance). [5]
- Works best in case of few highly important features. [5]
- Struggles to capture complex feature interactions. [5]
- Predictions are not continuous (bad extrapolation). [6]
- Bad at capturing linear relationships. [6]

3) b) Model description - K-Nearest Neighbors (KNN)

- KNN is an instance-based non-parametric supervised learning model for classification and regression. [7]
- Instead of estimating an underlying model, KNN classifies unseen data by comparing it to instances of the training set. [7]
- For classification, the assigned class is determined by a popularity vote of the closest k instances, as defined by a distance function. For regression, the assigned value is the average value of the neighbors. [7]
- In general, a higher k can suppress overfitting to noise, but can cause the majority class to dominate in every vote. [7]

Pros

- Simple and intuitive algorithm. [8]
- Instance-base learning does not require training. [7]
- Consequently, new data can be added seamlessly. [8]
- Few assumptions about the data (non-parametric). [7]

Cons

- Distance calculation is computationally expensive for large and high-dimensional data sets. [7]
- Requires feature scaling for distance calculation. [6]
- Assumes equal importance of all features. [6]
- Only limited forms of distributions can be represented. [7]

4) Hypothesis Statement

- It is expected that without addressing the class imbalance both models with be heavily biased towards the majority class.
- The metrics achieved by the deep neural network approach of Lui, Fan and Wu (2019), are expected to be an upper benchmark that cannot be exceeded by neither DT nor KNN without any significant tradeoff in at least one metric. These are a 71.6% accuracy, 32.6% specificity, and 67.4% sensitivity/recall.
- DT is expected to outperform KNN across most classification metrics. This is because, Lui, Fan and Wu (2019) showed that the feature importance in their Random Forest model is heavily dominated by only a few variables, which caters to the DT, as it tends to perform best in an environment with few important predictors [5], as opposed to KNN, which is negatively impacted by many irrelevant features [6] like one-hot encoded categorial variables with low correlation.
- KNN is expected to have a longer cross validation runtime, because performing exhaustive distance calculations on 43'000 is likely to outweigh the computational benefits of not requiring a training period.

5) Description of the choice of training and evaluation methodology

- Split data set into train (80%) and test set (20%), resulting in 33'404 training and 8'351 testing observations.
- Hyperparameter tuning is done using Bayesian optimization and 10-fold cross validation, to prevent overfitting.
- To deal with the strong class imbalance three approaches were compared. Namely, SMOTE oversampling, stratified cross validation and regular cross validation as a baseline. The oversampled training data contains 65'822 observations.
- Additionally, for each approach, the effect of using the F2-measure as the objective function is tested. Unlike accuracy, F2 is a classification metric based on precision and sensitivity. Compared to F1, F2 gives higher importance to sensitivity, which is a desired characteristic for models in the medical domain. Thus, six models are tuned in total.
- After extracting the best parameters for each approach, the six models are trained on the entire training set.
- Each model is then evaluated by its ability to predict the observations in the test set, based on false positive and false negative rates, as well as accuracy, specificity, and sensitivity.

6) a) Choice of Parameters and Experimental Results – Decision Tree

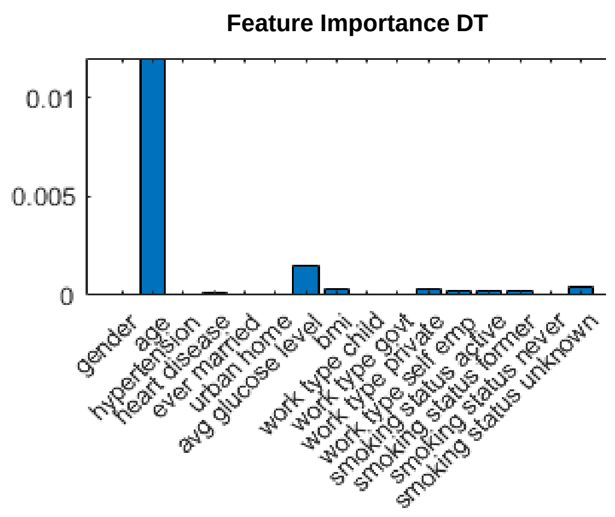
- To prevent overfitting, two stopping criterions, i.e., the maximum number of splits and minimum size of leaf nodes, were tuned in a range of 1-30, which limits the depth of the final tree.
- To help deal with the highly imbalanced data, the effects of implementing stratified cross-validation and SMOTE oversampling, as well as choosing F2-score instead of accuracy as the cross-validation loss function were tested.

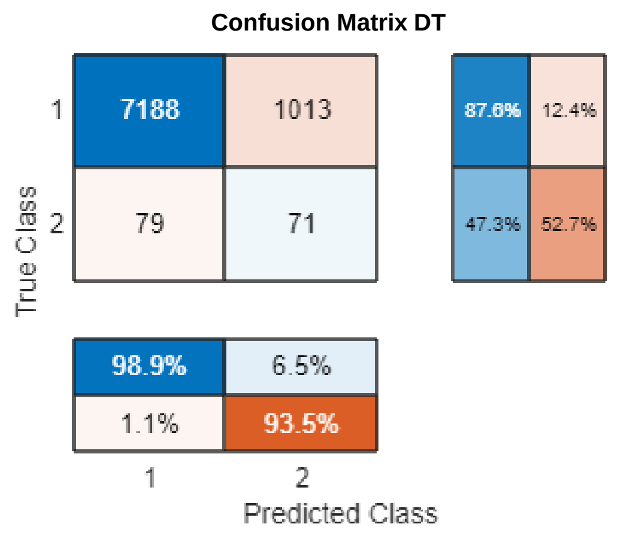
Results:

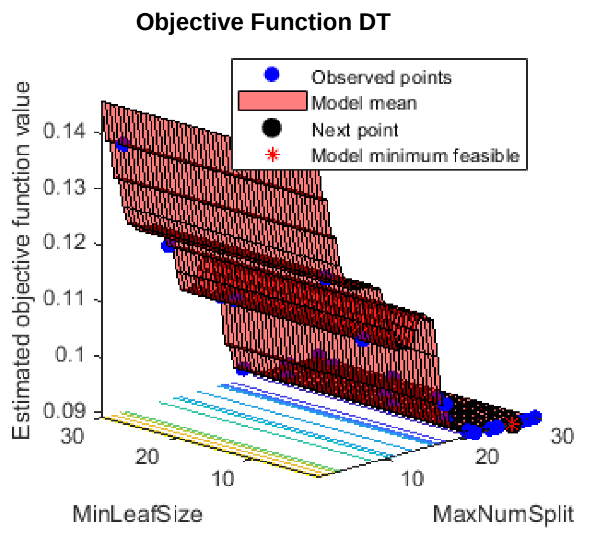
- SMOTE was the most effective technique that led to significant improvements in sensitivity.
- Using F2 instead of accuracy on oversampled data had no effect on evaluation metrics.
- The best model used SMOTE oversampling, accuracy loss, 26 maximum splits and 1 minimum observations per leaf.
- Minimum leaf size had only a minor effect on model performance.
- 'Age' is by far the most important feature.

Classification Metrics

Metric	Score
Accuracy	86.9%
Precision	6.6%
Recall	47.3%
Specificity	87.7%
F2-Score	21.1%
Runtime	29s







6) b) Choice of Parameters and Experimental Results – K-Nearest Neighbors

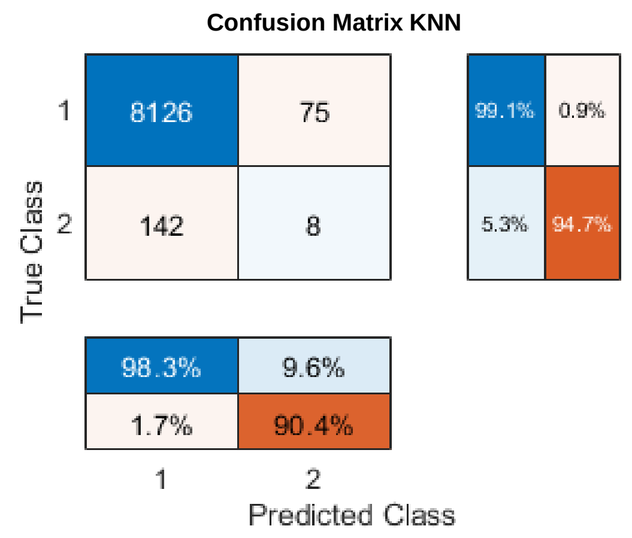
- The main hyperparameter, number of neighbors, was tuned in the range of 1-16, as 16 is the number of features in the data and thus constitutes the upper bound. Additionally, three different distance measures, i.e., 'Euclidean distance', 'cosine similarity' and 'correlation', were tested, as they belong to three different categories of distance measures [9].
- Again, SMOTE, stratified cross validation and F2 cross-validation loss were tested to address class imbalance.

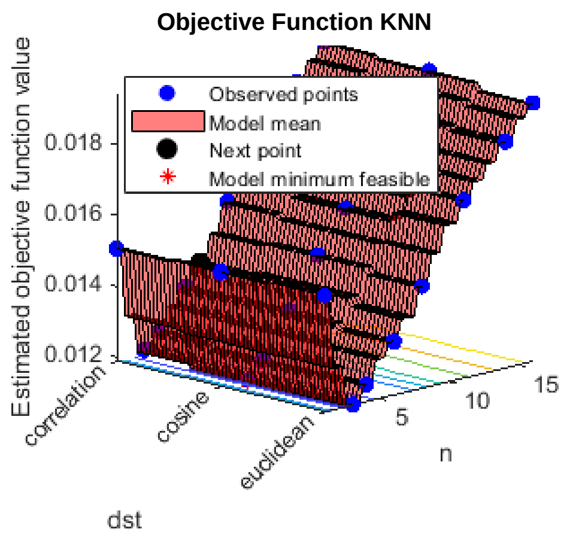
Results:

- SMOTE was the most effective technique that led to significant improvements in sensitivity.
- Using F2 instead of accuracy on oversampled data had no effect on evaluation metrics.
- The best model used SMOTE oversampling, accuracy loss, 3 neighbors and 'cosine' distance.
- The different distance functions had only a minor effect on model performance.

Classification Metrics

Metric	Score
Accuracy	97.4%
Precision	9.6%
Recall	5.3%
Specificity	99.1%
F2-Score	5.9%
Runtime	~10 min





7) Analysis and Critical Evaluation of Results

- The above analysis shows that the DT model is better suited for the application in cerebral stroke prediction compared to KNN. The main advantage of DT is the recall of 47.3% compared to only 5.3% for KNN. This is especially important in the medical context, where a false negatives, i.e., an undetected stroke, can cause serious harm to the patient, whereas a false positive, i.e., predicting a stroke in a healthy patient, will only result in an unnecessary trip to the hospital. In addition, this almost 10-fold increase in sensitivity, only increases the false-positive rate of the algorithm by 3.1%, from 90.4% to 93.5%. This overall improvement is also reflected in the F2-score of 21.1% for DT compared to only 5.9% for KNN.
- While the accuracy in the KNN model is 10.5% higher compared to the DT, this seems to stem mainly from the fact that KNN is more reluctant to predict the minority class. As shown in the supplementary materials, a classifier that only predicts the majority class has an accuracy of 98.2%, while not having any useful application.
- One possible reason for the superior performance of the DT, which is in line with the initial hypothesis, stems from the fact that the distribution of feature importance is similar to that of Lui, Fan and Wu (2019), which is characterized by 'Age' being by far the most important feature, while most others have very little importance. Thus, the analysis constitutes further evidence that DTs excel in an environment with one or few dominant features.
- In line with the initial hypothesis, the runtime of KNN is also significantly longer than that of the DT. Especially because training a single decision tree is fast, the KNN model cannot make up for its long prediction time by omitting the training period.
- Also in line with the initial hypothesis, the 32.6% specificity, and 67.4% sensitivity from the deep learning model by Lui, Fan and Wu (2019) were note exceeded, most likely because the simpler DT and KNN algorithms are less well suited to pick up on more nuanced feature interactions. However, unlike the black box deep learning model, DT and KNN provide much higher interpretability and easier intuition on how decisions for a given patient were formed, which is a highly desirable characteristic for the application in the medical domain. This observation suggests that there is a trade-off between high performance and high interpretability.
- From the DT objective function plot, one can clearly see that the maximum number of splits is the most important parameter, as the plane is almost completely flat along the 'MinLeafSize' axis. A possible explanation could be that the partitioning algorithm is usually stopped by reaching the maximum number of splits, before the size of the leaf nodes starts to restrict the tree depth.
- Similarly, for KNN, the objective function plot is mainly influenced by the number of neighbors, whereas the distance functions does not seem to have a big impact.
- In accordance with the initial hypothesis, without specifically addressing the class imbalance the DT and KNN models drastically overfit to the majority class while hyperparameter tuning. This results in low variance, at the cost of high bias. While they have a seemingly high accuracy of 98.2% (see supplementary materials) this just represents the share of the majority class in the data. The confusion matrix confirms that both models in fact only predict 'no stroke' for every observation.
- Stratified cross-validation did not improve models' ability to predict the majority class. While it eliminates the chance of having zero observations of 'stroke' in any given fold, the imbalance remains too high for the algorithms correctly model it.
- The results presented in section 6) show that SMOTE oversampling is indeed an effective technique for addressing the class imbalances. By having an equal amount of 'stroke' and 'no stroke' observations in the training data, the models are able to learn an effective decision boundary to discriminate between the two classes, which improves its ability to generalize to the unseen test data.
- The effect of choosing F2-score over accuracy as the objective function in Bayesian hyperparameter optimization is ambiguous. When SMOTE is applied to the training data, the final models that result from hyperparameter tuning have identical evaluation metrics on the test set, independent on the objective function. This suggests that in a balanced data setting, the model that achieves the highest accuracy is very similar to the one that maximizes F2 score. However, in an imbalanced setting, i.e., for regular and stratified cross-validation without SMOTE, the objective function had some effect on the final model. With regular cross-validation, the final DT and KNN model tune on F2 made at least some minority class predictions, albeit with a very small recall under 1%. With stratified cross-validation, only the KNN made minority class predictions, whereas the DT only predicted the majority class independent of the objective function used. These apparent inconsistencies in the improvements by an F2 objective function may be caused by the random CV splitting, especially since the overall influence of the objective function seems to be very small in general.

8) Lessons Learned and Future Work

- The comparison with Lui, Fan and Wu (2019) has demonstrated a trade-off between interpretability and performance.
- Adequate preprocessing techniques like SMOTE oversampling are critical to enable the model to learn a relevant decision boundary from imbalanced data.
- The effect of F2 objective function has only limited impact, on finding the optimal hyperparameters for generalizability.
- The long prediction time of KNN outweighs the benefits of not having a training period, when applied to large data sets.
- For KNN, the number of neighbors, and for DT, the maximum number of splits are the most important hyperparameters, that when tuned correctly find the optimal tradeoff between bias and variance of the final model.
- Future work could explore the impact of additional preprocessing techniques like PCA dimensionality reduction or the use of combining over- and undersampling to create balanced data.
- Additionally, feature selection techniques could be explored to remove the number of irrelevant features with the goal of improving the KNN classifier.
- Also, future work could explore the effect of other hyperparameters for DT and KNN that were not tuned in this analysis.

References:

[1] Office for National Statistics. (2021). Total deaths in the UK in 2020 and deaths from heart attacks, heart disease, cancer, and Alzheimer's and dementia, 2016 to 2020. Retrieved from: https://www.ons.gov.uk/aboutus/transparencyandgovernance/freedomofinformationfoi/totaldeathsintheukin2020anddeathsfromheartattacksheartdiseasecancerandalzheimersanddementia2016to2020

[2] Liu, Tianyu; Fan, Wenhui; Wu, Cheng (2019), "Data for: A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical-datasets", Mendeley Data, V1, doi: 10.17632/x8ygrw87jw.1

[3] Lewis, R. J. (2000, May). An introduction to classification and regression tree (CART) analysis. In Annual meeting of the society for academic emergency medicine in San Francisco, California (Vol. 14).

[4] ElShawi, R., Sherif, Y., Al-Mallah, M., & Sakr, S. (2020). Interpretability in healthcare: A comparative study of local machine learning interpretability techniques. Computational Intelligence.

[5] Kohavi, R., & Quinlan, J. R. (2002). Data mining tasks and methods: Classification: decision-tree discovery. In Handbook of data mining and knowledge discovery (pp. 267-276)

[6] Friedman, J. H. (2017). The elements of statistical learning: Data mining, inference, and prediction.

[7] Bishop, C. (2006). Pattern Recognition and Machine Learning. Springer

[8] Sun, J., Du, W., & Shi, N. (2018). A Survey of KNN Algorithm. Information Engineering and Applied Computing. doi:10.18063/ieac.v1i1.770

[9] Abu Alfeilat, H. A., Hassanat, A. B., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., & Prasath, V. S. (2019). Effects of distance measure choice on k-nearest neighbor classifier performance: a review. Big data, 7(4), 221-248.