Contents

1 Introduction

Stroke is the third leading cause of death and fourth leading cause of disability worldwide according to the latest Global Burden of Disease (GBD) estimates [1], representing 7.25 million deaths and over 160 million disability-adjusted life-years (DALYs) in 2021 [1, 2]. A stroke is characterised by an interruption of blood flow to areas of the brain caused by vessel occlusion (ischaemic stroke) or rupture (haemorrhagic stroke) [3]; the extent of resulting cellular death determines the degree of severity and subsequent disability or death [4, 5]. Ischaemic stroke generally accounts for a higher proportion of stroke incidences with estimates of 62% globally [6], however this has been shown to vary across regions of the world [7].

The age-standardised mortality and DALY rate for stroke has substantially decreased between 1990 and 2021 which contributed to increased life expectancy worldwide [1, 2]. Despite this, during the same period there was evidence of increasing burden as absolute numbers of deaths and DALYs from stroke increased dramatically which has been attributed to population growth, ageing and increasing exposure to risk factors [6]. It is predicted that the number of deaths and DALYs from stroke will continue to increase by as much as 50% and 30% respectively by 2050 [8], primarily driven by low and middle-income countries (LMICs) where already over 80% of global stroke burden is experienced [6].

The aggregate economic burden of stroke (incorporating direct and indirect costs) was estimated at $891 billion in 2017, representing 1.12% of global gross domestic product (GDP) [9, 10]. In high-income countries (HICs) such as the United States, Canada and those within Europe, the mean per patient per year (direct and indirect) cost of stroke was $27,702 in 2020 [11]. This has been found to be much lower but widely variable in LMICs due to availability of rehabilitation services and with a higher proportion of out-of-pocket expenditure [12]. Due to the high morbidity associated with stroke, the long-term costs attributed to disability and lost productivity are comparable to direct costs from healthcare as post-stroke survival improves [13]. It is predicted that by 2050 the global

economic burden could have risen to as much as $2.31 trillion in 2017 prices [8]. The predicted growth in global burden of stroke threatens to further increase the already vast economic impact and presents an an even greater challenge for LMICs which may have limited capacity to sufficiently resource health and social care [14].

Given the substantial and growing health and economic burden, the prevention of stroke is seen as a global public health priority [9, 15]. The United Nations (UN) Sustainable Development Goal (SDG) 3.4 aims to reduce premature mortality from non-communicable diseases (NCD) such as stroke by one third between 2015 and 2030 through investments in further prevention and treatment [16]. However, there has been insufficient progress towards achieving this goal, particularly in LMICs [17], leading to calls for resource-effective scalable approaches for surveillance and primary prevention [8, 17].

There are two predominant strategies for stroke prevention; population-based and high-risk [18]. Population-based approaches often utilise primordial stroke prevention which aims to target the underlying social and environmental determinants of disease before risk factors emerge [19]. This approach often requires engaging health systems and implementing policy to avert a populations exposure to risk factors for stroke completely [18, 19]. High-risk stroke prevention strategies tend to encompass primary prevention whereby risk factors are already present and the focus is identification of individuals at a higher risk of stroke followed by modification of these factors [15, 19, 20]. For example, this could include identification of patients at a higher risk of stroke using early detection and management of risk factors at the patient-level using routine healthcare records [8].

There are a number of established non-modifiable and modifiable risk factors for stroke [21, 22]. Stroke incidence is known to increase with age [23] doubling with each decade beyond the age of 55 for both sexes [21, 24]. There is evidence of an interaction between age and sex [25]; stroke incidence is higher in females at <30 years of age [26] which may be attributed to the higher stroke risks associated with pregnancy [27] whilst in mid-life, males have a greater relative risk of stroke

[28]. A multinational case-control study (INTERSTROKE) found that 90% of the risk of stroke could be attributed to ten modifiable risk factors including hypertension, current smoking, diabetes, heart disease and obesity amongst others [7]. This aligns with more recent GBD estimates with the top five risk factors identified as high systolic blood pressure (hypertension), body-mass index (BMI), fasting blood glucose (diabetes), pollution and smoking [6] and is consistent within LMICs [18]. Identification and management of these risk factors is key to preventing stroke and provides an opportunity for early intervention [1, 6, 29].

Predicting stroke risk based on risk factors has been recognised as a method for high-risk strategy primary prevention [30–32]. Conventional statistical-based scoring methods such as the Framingham stroke risk score [30] and more recently QStroke [33] have been used to stratify and identify risk at the patient-level. However, statistical regression-based model limitations, namely linearity assumptions [34] which can oversimplify complex relationships between predictors [32], have led to the growth of machine learning (ML) methods for predicting stroke and stroke outcomes [35, 36]. Supervised ML generally involves iteratively "training" computational algorithms using a labelled training dataset where the outcome label (classification) or real-value (regression) is known in order to generate a function that maps inputs to outputs [37, 38]. Using feature importance and accumulated local effects plots [39], these methods can also provide insight into which features (health and lifestyle factors) have most contributed to the stroke predictions [40].

There is an extensive literature on the use of machine learning methods for stroke risk prediction using routinely collected patient-level information [41]. Sharma et al[42] used 10 features including demographic (age and sex), lifestyle (smoking status, marital status, urban/rural area and occupation sector) and clinical (hypertension, heart disease, average blood glucose and BMI) attributes to predict the risk of stroke occurrence as a binary classification problem. Comparing five ML classifiers they found that random forest, an ensemble method, provided the best performance for classifying the occurrence of stroke [42], consistent with other findings utilising the same dataset [43–48].

Despite this, there is some evidence to suggest support vector machines may best classify stroke occurrence using the aforementioned dataset, even whilst comparing directly with random forest algorithms [49, 50]. Similarly effective classification performance has been achieved using other ensemble methods aside from random forests including stacking [51], where the output of multiple base classifiers are combined by a meta-classifier for final classification [52]. This method has been shown to outperform many single classifier models in the prediction of stroke occurrence as a binary classification problem [51, 53–55]. Using an ensemble stacking algorithm, Hassan et al[55] also calculated feature importance scores and found age, average blood glucose, BMI, heart disease, hypertension and marital status were the most influential in the prediction of stroke. This is consistent within the established ML literature utilising this dataset [54, 56, 57] and presents an opportunity to highlight modifiable risk factors for preventative action [21].

Therefore, there remains uncertainty in the literature around the most effective ML method for stroke prediction using routinely collected demographic, lifestyle and clinical features [55]. This project will aim to add to the body of evidence by further comparing ML classifiers in order to determine the most effective means of predicting stroke occurrence using readily available patient-level information. The resulting predictive models will be used to corroborate previous findings on the most influential risk factors that contribute to predicted stroke occurrence. Further, accumulated local profiles will be used to explore how changes in these features influence the predicted risk of stroke on average [39]. A successful model may provide accurate decision support for clinicians in identifying patients at higher risk and provide an opportunity for early intervention in managing risk factors, reducing more serious health and economic consequences [8, 29].

2 Methods

The following section will detail the methods used to create a predictive ML model for binary classification of stroke occurrence. This will include a description of the dataset used in the analysis followed by data exploration and preprocessing in order to facilitate modelling using ML classifiers. An overview of each ML algorithm will be provided followed by details of corresponding hyperparameters utilised in model tuning. Model implementation and comparison methods (using appropriate evaluation metrics) amongst candidate models will be described. Finally, model selection and assessment methods will be detailed and model interpretation explored, including feature importance and accumulated local effects.

All data processing and analyses were completed using R version 4.4.0 [58]. The `tidymodels` meta-package and ecosystem [59] was used for all data preprocessing and modelling.

2.1 Data description

The dataset was open-access data sourced from Kaggle [60] and consisted of 4,981 observations of anonymised electronic health care records (EHR) [61]. The dataset had already undergone processing however the extent and type of processing applied was unknown. Therefore as the data source was also unknown a number of assumptions about the dataset were made. For the purposes of this analyses, as there were no unique identifiers present, it was assumed that each observation represented a unique patient. The outcome and focus of this classification problem was a binary categorical variable `stroke` which indicated if a stroke had occurred or not for each patient. The stroke outcome was assumed to have been collected as a follow up to collection of various features therefore the dataset could be used to build a predictive model for stroke occurrence as a future outcome.

The dataset comprised of ten features linked to demographic, lifestyle and clinical information. These included age (years), gender, smoking status, marital status, urban/rural residence, occu-

pation sector, hypertension, heart disease, average blood glucose (mg/dL) and BMI (kg/m^2). All features were defined as categorical nominal data except for age, average blood glucose and BMI which were numerical continuous. The binary features "hypertension" and "heart disease" were assumed to represent whether the patient had these diseases at the point of follow-up. Similarly, "average blood glucose" and "BMI" were assumed to have been collected over a number of appointments.

2.2 Data exploration

Initial exploratory analysis was conducted using descriptive statistics and visualisations to summarise and contextualise the dataset [62] prior to modelling for stroke prediction. Univariate distributions of each variable were assessed for erroneous values, missing data and outliers using summaries and visualising distributions. Density plots were used to inspect the distributions of continuous variables whilst cross-tables were used for examining each categorical variable including the outcome. This assisted in establishing symmetry or skew in continuous features for which transformation may later be required in pre-processing for some ML algorithms through a process of scaling and normalisation [63]. Univariate methods also provided the event rate for the binary outcome of stroke occurrence, providing class distribution information to inform subsequent subsampling methods in the case of class imbalance [64, 65].

Bivariate and multivariate visualisations were also used to assess patterns and relationships between features and the binary outcome. Density plots were used to visualise the relationship between continuous features and the outcome whilst cross-tables described the relationship between categorical variables and the outcome. This aided in identifying the features that appeared to be the most informative for the prediction of stroke occurrence as a binary outcome by examining dependency. Similarly, relationships between numeric features were also explored using correlation matrices and scatter plots to identify multicollinearity amongst the predictors which could indicate redundant information [66], and even introduce instability in some ML methods [67].

2.3 Data preprocessing

The dataset was initially split into a training and test set using the `initial_split` function from the `rsample` package [68], utilising a stratified approach to ensure equal proportions of the binary stroke occurrence outcome in both sets [69]. The training set consisted of 80% of the dataset and was used to tune and fit the parameters of classifiers followed by model selection [70]. The test set consisted of the remaining 20% of the dataset and aimed to provide an unbiased estimate of model performance to indicate generalisability [70, 71].

Preprocessing was implemented using the `tidymodels` framework which allowed all steps to be applied to both training and test datasets via the `recipes` package [72]. The exception to this was subsampling which was only applied to the training set as default, including within cross-validation, so as to avoid bias [73].

Transformations are particularly important for parametric ML methods where the tails of skewed distributions can negatively impact the models ability to classify more typical cases [74]. Tree-based methods are less sensitive to skewed distributions and outliers as these algorithms utilise rank as opposed to the value of the predictor [75, 76]. Continuous features that demonstrated a skewed univariate distribution were transformed by applying an appropriation of the Box-Cox transformation [77] using the `recipes::step_BoxCox` function in order to approximate a more symmetrical distribution.

Many ML methods require all features to be numeric [78, 79], therefore where appropriate, categorical predictors were encoded using the `recipes::step_dummy` function which utilises a dummy encoding method [80]. A full rank parameterisation is used, whereby numeric binary dummy variables are generated representing all but the first factor level which is used as the reference cell [80, 81]. Again, tree-based algorithms are more robust in this way due to splitting methods, therefore can handle categorical data and do not require this step of preprocessing [82, 83].

Feature normalisation was implemented using the `recipes::step_normalize` function. Normalisation involves centering and scaling numeric predictors so that they have a mean of zero and a unit variance [84, 85]. This is utilised when the scale of numeric features vary by orders of magnitude and therefore features with larger values may disproportionately impact predictions, particularly for those ML algorithms that use distance calculations [86]. There is also evidence that gradient descent, the process by which algorithms are optimised [87], convergences to a minima at a faster rate following normalisation of inputs [88]. As before, tree-based algorithms are invariant to monotonic transformations of features due to splits based on order as opposed to absolute value of a feature [89].

In the case of an imbalanced dataset (i.e. a class imbalance in the stroke outcome), with fewer instances of stroke occurrence than non-occurrence, Synthetic Minority Over-sampling Technique (SMOTE) was used [90] to generate new observations of stroke occurrence and create an equal proportion of classes. This was achieved using the `step_smote` function from the `themis` package [91] and applied exclusively to training sets including within resampling of training data using cross-validation so as to never apply it to any validation data. Imbalanced data presents a challenge for classification models and performance evaluation using accuracy metrics as misleadingly high predictive accuracy can occur by only predicting the majority class due to a low event-rate of the minority class [92, 93]. SMOTE employs oversampling of the minority class which has been shown to achieve better classifier performance in binary classification problems than conventional undersampling of the majority class in some instances [90]. New synthetic samples of the minority class (in this case; stroke occurrence) are generated via a k-nearest neighbour method (using $k = 5$ as default) whereby novel examples are created along line segments that join the neighbours of a randomly selected data point in the feature space [90].

2.4 Machine learning models

The primary motivation for this project was to build a predictive model for the classification of stroke occurrence as a binary outcome using readily available patient-level information. According to the 'no free lunch theorem' [94] there is no universal best ML algorithm for predictive modelling problems in supervised ML; consequently, it is typical to compare many ML algorithms in order to evaluate and select the optimal approach for each specific problem [95]. This project will therefore compare four ML algorithms including artificial neural networks, support vector machines, gradient-boosted decision trees and random forest. The following section provides an overview of each ML algorithm selected for constructing classification models to predict stroke occurrence.

2.4.1 Artificial neural networks

Artificial neural networks (ANNs) are computational models that were first inspired by biological neurons in the human brain [96] and offer a flexible mathematical approach to nonlinear statistical modelling [97]. The following section describes a commonly used ANN known as a multi-layer perceptron (MLP), which was implemented as a single hidden layer neural network using the `brulee` package [98]. MLPs require encoding of categorical predictors for numeric input [78] and scaling and normalisation to account for scale factors [63].

An MLP model consists of an input layer, one or more intermediate hidden layers and an output layer, each with nodes that are fully connected to nodes in previous and subsequent layers [99]. This algorithm is also a form of feed-forward network whereby information flows successively through hidden layers initiated from the input layer and terminating at the output layer with predictions of the target features [100]. The input layer simply consists of the features or predictor variables in the dataset [101]. Information moves through the feed-forward topology by using the outputs from previous layers as inputs for the next until the outcome layer is reached [102].

MLPs are highly parameterised models; each connection between nodes has an associated weight

and bias parameter which are linearly combined to outputs from previous nodes before becoming the input to the subsequent node [103]. These inputs are then transformed to finite values at hidden nodes within the hidden layers using activation functions which provide a threshold at which the node to continue to signal to another node [102]. The output of these transformations then feed into subsequent hidden nodes or output nodes to model the output [104]. The nonlinearity of the model and its ability to fit to complex interactions is due to the transformation of inputs via the activation function (see section 2.5.1) [105].

In order to train an MLP model the weight and bias parameters are estimated through an optimisation process called backpropagation which seeks to minimise the loss function by gradient descent [106]. In classification problems, the loss function is typically cross-entropy which is a measure of the difference between two probability distributions [107]. The algorithm involves a forward pass with randomly selected parameters before predicting the output and estimating the error using the loss function [108]. The gradient of the error is propagated back to earlier layers in the network to update the weights in the opposite direction to the gradient [106]. Once all training data has contributed to the updating of the parameters in the forward and backward pass this is known as an epoch (see section 2.5.1) [109].

2.4.2 Support vector machines

Support vector machines (SVMs) are machine learning models that can be used for linear and non-linear binary classification by using a hyperplane to separate classes [110]. For this classification problem, SVM was implemented using the `kernlab` package [111]. SVMs also require encoding of categorical predictors [112] and scaling and normalisation for differing magnitudes of predictors [63].

A hyperplane represents a decision boundary for separating classes however many possible boundaries could be used; SVM finds an optimal hyperplane based on maximising the margin as a constrained optimisation problem [113]. The margin is the distance between the hyperplane and the

closest observations (support vectors) of either class and the margin boundaries can be defined by these support vectors which in turn dictate the hyperplane [114]. This conceptual solution allows an optimal separating hyperplane to be found that generalises well in linearly separable data [115].

However, as data is rarely perfectly linearly separable and maximal margin classifiers are sensitive to outliers, SVM often allows the violation of constraints [116]. For greater generalisability, a soft margin can be created using slack variables to allow misclassification by permitting observations on the incorrect side of the hyperplane whilst still maximising the margin [117]. If the cost (see section 2.5.2) of misclassification is high then there will more of a hard margin (strict), whereas if it is lower the margin will allow more misclassified observations [106].

The advantage of SVM over conventional linear classifiers lies in its ability to classify nonlineary separable data using the kernel trick [118]. Using kernel functions (see section 2.5.2) , the input data can be mapped to a higher dimensional feature space which then allows SVM to create a separating hyperplane [119]. The linear hyperplane implemented in the higher dimensional feature space can then be used in the original feature space where it can act as a nonlinear decision boundary to classify observations [116]. The kernel trick is computationally efficient, even whilst operating in an infinite dimensional space, as it engages in this feature mapping without requiring transformation of features [120]. SVMs are known for their flexibility in nonlinearly separable data and generation of smooth boundaries through this process, making them a favourable machine learning algorithm in many binary classification problems [106].

2.4.3 Gradient-boosted decision trees

Gradient-boosted decision trees (GBDT) are an ensemble ML technique whereby successive decision trees, as weak learners, are combined to build a more accurate predictor [121]. The following section describes a popular GBDT algorithm, XGBoost which uses regularisation to account for model complexity and prevent overfitting [122]. This was was implemented using the `xgboost` package [123]. XGBoost is a tree-based method therefore does not require scaling or normalisation of predictors

[89] however encoding is required for categorical predictors for this specific algorithm [124].

XGBoost uses decision trees as base learners in the ensemble model [122]; decision trees are hierarchical non-parametric algorithms that use recursive partitioning to split the data based on features in a way that maximises homogeneity of subsets after each split [125]. This forms a tree-like structure from the first split of all training data at the root node, to further splits at internal nodes before terminating at the leaf nodes [126]. The leaf nodes hold the product of all previous splits encapsulating a decision rule which can then be used for prediction [127]. Subdivision of nodes is typically based on binary splits where branches represent the outcomes and connections from these root or internal node splits [75]. The aim of each split is to minimise heterogeneity [128] in the resulting partitioned data and this is also used in order to select which feature is optimal for the first split at the root node [129]. As decision trees can be prone to overfitting [130], early stopping rules [131] or pruning [132] can be used to improve generalisability (see section 2.5.3).

Individual decision trees are known to produce relatively unstable predictions in response to small changes in the training data however combining trees in an ensemble approach such as boosting can make predictions more reliable [133]. Boosting begins with a shallow (few splits) decision tree that typically has high bias and low variance making it a weak learner [134, 135]. A loss function is minimised using gradient descent optimisation and provides the residuals which are then used to fit the next tree; this continues sequentially, combining trees that are optimised to predict the residuals from the previous tree [122]. This gradient boosting process results in each iteration of the model providing better prediction performance than the last by combining weak learners into an ensemble [136]. GBDTs such as XGBoost are non-parametric tree-based methods, therefore they can handle mixed data types and do not require feature transformations [137].

2.4.4 Random forest

Random forests (RF) are another ensemble ML approach that utilises bootstrapping and aggregation of decision trees in a process known as bagging [138] whilst also reducing correlation between

base learners to produce a strong predictive model [139]. For this classification problem, RF was implemented using the `ranger` package [140]. RF, as a tree-based method, does not require encoding for categorical predictors [82] and is also invariant to monotonic transformations of predictors therefore no scaling or normalisation of predictors is required [89].

As with XGBoost, the base learners for RF are decision trees which are typically unbiased but high variance models for which ensemble methods such as bagging work especially well [141]. The bagging process involves creating many individual decision trees using random subsets of the training data with replacement through a process of bootstrapping [142]. RF aims to create a diverse array of trees to reduce correlation and improve predictive performance therefore additional to bootstrapped samples, a random subset of features are used to build decision tree nodes at each step in a given tree (see section 2.5.4) [141]. This is repeated many times to create identically distributed trees each of which provides a prediction before being aggregated either by averaging (regression) or by voting as a committee of trees (classification) [139].

RF algorithms can demonstrate impressive predictive performance even with minimal or no hyperparameter tuning [143, 144] and as a tree-based method is invariable to feature transformations [145] and mixed-data types [146].

2.5 Hyperparameter tuning

ML algorithms have hyperparameters which are parameters that cannot be estimated from the learning process itself [147]. Hyperparameters are customisable and specified on configuration of each model allowing model adaptation to to suit the relevant dataset properties and ML problem [148]. It can be challenging to know prior to modelling which hyperparameter values will bring about optimal model performance therefore it is common practice for tuning strategies to be used, where various hyperparameter values and combinations are compared [147].

Hyperparameter tuning was completed within the `tidymodels` ecosystem [59]. The training set was

used throughout the tuning and model selection process with the test set reserved for final model evaluation [70]. Further, ten-fold cross-validation was implemented using the `rsample::vfold_cv` function [68] on the training dataset, stratified on the stroke outcome variable to ensure equal proportions across folds. K-fold cross-validation was used in tuning to characterise and compare model performance when estimating hyperparameters in order to provide a more reliable performance estimate [149]. Resampling methods such as cross-validation can indicate how well models may perform on unseen data and therefore prevent overfitting [80].

Despite existing heuristics and understanding of the non-specific impact of hyperparameters on model performance [150], a more objective approach utilises a defined search space for various values of each hyperparameter in combination [148]. One such method is grid search where this search space is defined by pre-specified finite subsets of hyperparameter values in combination which are all systematically and exhaustively evaluated [151]. Space-filling designs such as those employing Latin hypercube sampling can generate near-random sequences of hyperparameter values which may cover the search space more evenly with less chance of overlap and redundancy [152]. To implement this space-filling design for grid-search, the `dials::grid_latin_hypercube` function [153] was used with 50 candidate parameter sets for each algorithm.

Racing methods can be used to accelerate the grid-tuning process; instead of all models being fit to all folds of training data, racing methods fit and evaluate after an initial subset of folds in order to discard clearly inferior models [154, 155]. From this process, optimal values and combinations of hyperparameter candidates are efficiently identified based on performance metrics. A racing method for hyperparameter tuning was implemented for all models in this project using the `finetune::tune_race_anova` function [156].

The following section outlines the relevant hyperparameters and tuning considerations for each ML method used in the process of hyperparameter optimisation.

2.5.1 Artificial neural networks

Number of hidden nodes As a single hidden layer MLP was used, the number of hidden nodes determines the number of parameters and therefore the capacity of the model [157]. A greater number of hidden nodes may allow the model to learn more complex patterns in the data but also presents a risk of overfitting with increasing numbers of parameters [158]. It is common to prioritise a larger number of hidden units whilst relying more on regularisation to prevent overfitting for MLP models [141, 159]. The optimal number of hidden nodes will depend on context however as a heuristic principle, 75% of the number of input nodes has been widely used [160]. Therefore, a range inclusive of this can provide a starting point for tuning.

Weight decay Weight decay is a regularisation method that prevents overfitting by controlling the magnitude of the coefficient (weight) parameters in the MLP model [161]. The penalty for larger weights is incorporated into the loss function and reasonable values are generally between 0 and 0.1 [147].

Dropout Dropout is another method of regularisation again used to prevent overfitting by effectively combining multiple neural network structures during training [162]. It aims to create independent representations of the data to be learned in order to become less vulnerable to any specific weights within the network. Nodes and associated connections are randomly dropped during training to prevent these co-adaptations or dependencies, which in turn improves generalisability [162]. The range of this hyperparameter is bound between 0 and 1 as it represents the proportion of model parameters that are nullified whilst training the model. Both weight decay and dropout cannot be used within the `mlp` function [98], therefore dropout was used [163].

Epochs Every training example contributes to the updating of model weights and biases within a single epoch via a forward and backward pass [164]. Greater numbers of epochs may be required for increasing numbers of features and complex interactions in the data [106] however a large number of epochs can induce overfitting [165]. The default range of 10 to 1000 epochs was used in tuning with

attention paid to increases in validation error that may indicate overfitting with a greater number of epochs [148].

Activation function The activation function of a node is responsible for transforming inputs and weights to produce an output of a known range often introducing nonlinearity [105]. There are a number of activation functions, each with different mathematical processes; often the same activation function is used across all nodes in an MLP model [166]. For binary classification problems a sigmoid function is generally used in the output layer as it offers a probability interpretation [106] and there is some evidence that this function can also be effective in hidden layers [167]. Therefore, a sigmoid activation function was applied in this project for modelling stroke occurrence.

Learning rate The learning rate determines the extent to which weights update during back-propagation using loss minimisation in gradient descent [168]. If the learning rate is reduced, smaller incremental "steps" will be taken towards minima of the loss function therefore training will take longer to converge however if learning rate is larger it is possible to overshoot minima [169]. Typically, learning rates between 0 and 1 can aid in preventing overshooting [170].

2.5.2 Support vector machines

Kernel function Many kernel functions can be selected to map the input data to a higher dimensional feature space allowing SVM to create a nonlinear separating hyperplane [171]. The radial basis kernel is a popular function that maps input data to an infinite-dimensional feature space [172] and has been used in many binary classification health applications [173–175]. This kernel and associated parameters were used to implement the SVM model.

Cost The cost dictates the extent of the soft-margin, allowing misclassification of observations and making SVMs more robust to outliers [176]. Although there is some evidence the cost parameter demonstrates poor tunability [144], some tuning recommendations comprise of searching using exponentially growing sequences of cost for example between a range of 2^{-5} and 2^{15} [112] which was implemented for the SVM model.

Kernel function bandwidth The bandwidth (σ) determines the degree of non-linearity of the decision boundary [177]. Smaller values of σ will create a strict decision boundary and presents a risk of overfitting, whilst larger values can lead to smoother boundaries and greater misclassification [178]. Exponentially growing sequences of σ between a range of 2^{-15} and 2^3 [112] can be used as a starting point for tuning.

2.5.3 Gradient-boosted decision trees

Number of randomly sampled predictors Feature subsampling, the same technique used in RF algorithms, can be implemented in XGBoost; this tuning hyperparameter specifies the number of predictors that are randomly sampled (*mtry*) at each split [179]. Subsampling in this way decorrelates trees which can make the model more resilient to overfitting and decreases computation time [122]. This was tuned using a range of 1 to the total number of predictors in the dataset.

Number of trees The total number of trees in the boosting ensemble presents more of a risk of overfitting compared to bagging ensembles as they are sequential, using residuals from previous trees as a starting point [106]. It is generally recommended to include as many as 1000 trees in the tuning range [137], therefore the default hyperparameter range of between 1 and 2000 trees was used.

Minimum node size Minimum node size acts as a form of stopping criterion as it defines the minimum number of examples in a node before further splits [180]. In this way it sets the depth of tree as setting larger values leads to fewer splits and therefore less tree depth. Conversely, smaller values can create more complex trees and therefore increase the risk of overfitting [106]. The default range of 2 to 40 was used in tuning.

Maximum tree depth Similar to minimum node size, the tree depth hyperparameter controls tree complexity by defining the number of splits [106, 181]. A greater number of splits provides more flexibility allowing the tree to capture more complex interactions but also makes the model more

prone to overfitting [182]. A tree depth of between 4 and 8 is typically adequate [141], therefore the default range of 1 to 15 was used for this hyperparameter during tuning.

Learning rate Often referred to as the shrinkage parameter, the learning rate is a form of regularisation that scales the contribution of each tree and therefore the rate of gradient descent [183]. Smaller values can prevent overfitting by reducing the impact of any individual tree and subsequently allowing a greater number of trees to be incorporated into the ensemble [122]. A learning rate between 0.001 and 0.3 are often used [106] therefore the upper bound of the default range (10^{-10} to 10^{-1}) was extended to incorporate this.

Loss reduction Loss reduction (γ) is another method of regularisation that defines a minimum reduction in loss function in order for further splits of a leaf node in a tree [122]. Larger values of γ means a higher degree of regularisation whilst smaller values allow greater numbers of splits and increased likelihood of overfitting [106]. The default range of 10^{-10} to $10^{1.5}$ was used for tuning γ.

Sample size The sample size hyperparameter defines the proportion of training examples subsampled wihtout replacement before growing each decision tree in the ensemble at each iteration [122]. Not only can this method make computation time faster, it also introduces randomness that can further improve generalisability and prevent overfitting [184]. Typical values are often between 50% and 80% of the training data [106, 137] which was used as a range for tuning.

Early stopping Early stopping is implemented by specifying the number of iterations or trees where no performance improvement is observed before stopping the algorithm [185]. It can inform the selection of the optimal number of trees before overfitting occurs, improving efficiency of the model [186]. The default range of 3 to 20 iterations was used for tuning.

2.5.4 Random forest

RF has been found to be less sensitive to hyperparameter tuning compared to many other ML methods [143] and can often perform well with default hyperparameter settings [106]. However,

this is likely to vary based on the number of features and size of datasets [187] therefore tuning was conducted using a number of parameters.

Number of randomly sampled predictors As in XGBoost, feature subsampling can be implemented by specifying the number of predictors that are randomly sampled (*mtry*) at each split [188]. Introducing randomness in this way decorrelates trees and provides more stability in the ensemble, preventing overfitting [143]. For classification problems, *mtry* as $\sqrt{p}$ where p is the number of predictors is often default and has been recommended in the literature as a heuristic [189]. Therefore *mtry* was tuned using a range of 1 to the total number of predictors in the dataset.

Number of trees The total number of trees in the RF ensemble presents less of a risk of overfitting compared to boosting ensembles as they are independently grown and aggregated [106]. It is generally recommended to include $p \times 10$ trees, where p is the number of predictors, in the tuning range [106]. Therefore the default hyperparameter range of between 1 and 2000 trees was again used.

Minimum node size Similar to XGBoost, minimum node size defines the minimum number of examples in a node before further splits thereby controlling the complexity of the model [190]. Generally, the default value for classification is 1 to obtain adequate performance [143] therefore, the default range of 2 to 40 was extended to include 1 at the lower bound for tuning.

2.6 Stacking ensemble

Ensemble methods such as those used in XGBoost and RF algorithms can be used to combine base learners and their individual predictions into a meta-learner that provides a single prediction [191]. The predictive performance of the ensemble model often outperforms any of the individual base-learners [192, 193]. Stacking was one of the first established ensemble methods [194], and provides a way to combine many different model types as base learners into a new meta-learning algorithm [195].

The stacking ensemble was implemented using the `stacks` package [196]. In accordance with stacking methodology, the stacking model included four model definitions; MLP, SVM, XGBoost and RF in order to compare the ensemble performance against each individual method. Candidate base learners were models derived from the race tuning process for each method using k-fold cross-validation therefore the stacking model was established using four different types of model and different hyperparameter configurations of each [197].

Out-of-sample predictions from k-fold cross-validation were used for all candidate base learners along with the observed stroke outcome in order to construct the stacking model [197]. As this was a binary classification problem, a regularised generalised linear model (logistic regression) was used as a meta-learner to combine these predictions from the candidates and generate non-negative coefficients for each [195]. A lasso penalty was used for regularisation which helps identify correlation between candidates in order to remove them from the ensemble [198]. The stacking coefficients arise from this training process and provide a method of weighting the predictions from each candidate; only non-zero candidate base learners were included in the stacking model [197].

The resulting base learners were then fit to the entire training dataset generating the final stacked model which could then be used on the test dataset to obtain final performance metrics.

2.7 Model selection

2.7.1 Evaluation metrics

Many metrics can be used to evaluate binary classifiers and often these are based on a 2x2 confusion matrix which is used to establish predictive performance and quantify the types of errors the model produces [199]. In the context of the current project, those patients that have had a stroke occurrence are considered to be the class of interest i.e. the 'positive' class. Therefore, a type I error (false positive (FP)) represents the number of patients who were misclassified as having a stroke occurrence when they in fact did not. A type II error (false negative (FN)) represents the number

of patients who were misclassified as not having had a stroke occurrence but actually did.

In imbalanced classification problems such as the prediction of stroke occurrence where there are many fewer positive than negative observations, certain metrics such as accuracy can be misleading; this is known as the accuracy paradox [200]. Accuracy is the proportion of true positive (TP) and true negative (TN) samples out of all samples. Therefore, where the positive class is a minority a high accuracy can be obtained simply by classifying all samples as the negative class [201]. For this reason, many studies utilise multiple alternative metrics such as precision, recall, F1 and area under the precision-recall curve (AUPRC) for evaluating performance in imbalanced data classification problems [53, 54, 202]. AUPRC is particularly useful as it is threshold-independent and uses the predicted probability of class membership to evaluate the effectiveness of a model in separating classes across all decision thresholds [203]. Therefore, a combination of these performance metrics will be calculated during model evaluation and are defined in the following sections.

Precision Precision is the proportion of positive class predictions that were correctly classified, shown in Equation 1. Precision is defined between 0 and 1.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Recall Recall is calculated in the same way as sensitivity which is commonly used in medical diagnostic testing [204]. It is defined as the proportion of actual positive class samples that were correctly classified, shown in Equation 2. Recall is defined between 0 and 1.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

F1 Score F1 score is defined as a combination (harmonic mean) of precision and recall, with a higher score indicating the classifier performs well both in identifying positive samples and minimising type I and type II errors [205], shown in Equation 3. F1 score is defined between 0 and 1.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{3}$$

Area under precision-recall curve (AUPRC) As opposed to a threshold metric derived from a confusion matrix, AUPRC is a ranking metric that requires probabilities of class membership to which varying thresholds can be applied to produce a curve [206]. It calculates the area under the precision-recall curve (PRC), providing an indication of how well the classifier performs in classifying the minority class and allowing for comparisons across models [203]. AUPRC is defined between 0 and 1 and with recall on the x-axis and precision on the y-axis it is maximised by the curve reaching the top right corner of the plot [203]. A baseline AUPRC for a random (naive) classifier can be given by the proportion of positive cases in the training dataset during model training and test dataset for final model evaluation [207].

2.7.2 Comparing models and final fit

Models were tuned and optimised using the AUPRC metric and 10-fold cross-validation. Model selection within each ML method after tuning was guided by selecting the model that provided the highest AUPRC across the various tuned models. For each ML method, the selected optimal model was re-fit to cross-validation resamples using the training data and its performance evaluated using AUPRC.

Precision, recall and F1 score for the optimal model of each ML method were calculated; as the default decision threshold for binary classification was 0.500 in the `tidymodels` ecosystem, the `threshold_perf` function from the `probably` package [208] was used to identify the decision threshold at which the F1 score was optimised. This could then be used to report threshold metrics and compare optimal F1 performance across models.

Final model selection for use on the test dataset was guided by selecting the individual model that provided the highest AUPRC on the training data across the four different algorithms. All of the

above metrics were again calculated (and optimal F1 score decision thresholds recalculated) for final model evaluation on the test set for both the final individual and stacked model. PRCs derived from the test data were also be compared to illustrate differences across thresholds for the individual and stacked model. The final evaluation of model performance on the test set demonstrated whether the model indicated generalisability if performance remained stable or overfitting if a reduction in performance was observed [130].

2.8 Model interpretation

An important consideration after model selection and evaluation was to explore model interpretation; explaining why a model produces the predictions it does can help establish which features contribute to predictions and the impact changes in values of features alter predictions on average [209]. This is typically more challenging in complex nonlinear models which have historically been said to be "black-box" methods [210] however, there are now a number of ways to derive model explanations [211, 212]. Global model explanations were explored in this project using feature importance to determine influential features [213] and accumulated local effects to decipher how changing values of continuous features impact predictions on average [39].

2.8.1 Feature importance

Feature importance is a model agnostic method that aims to explain which features are most influential in the resulting predictions from a model [106]. Permutation-based feature importance was implemented using the `DALEXtra` package [214], test data and the final model fit to training data [127].

The permutation method establishes the importance of each feature by individually permuting or shuffling the values of the feature to disrupt the relationship to the target variable, before calculating the change in model performance [215]. A feature is considered important the more permutation or shuffling of values decreases the performance of the model (i.e greater error) implying the model was more heavily reliant on this feature in order to make better predictions [215]. Therefore for

this binary classification problem, a measure of error was defined by subtracting AUPRC from 1 to remain consistent with metrics used throughout the modelling process.

2.8.2 Accumulated local effects

Accumulated local effects (ALE) combine local model explanations to form a global explanation of how model predictions change across different intervals of individual features [39]. As opposed to partial dependence, ALE can provide a more realistic representation by using intervals to isolate from correlations between features which typically misrepresent how features influence predictions individually [127]. Accumulated local effects plots were implemented using the `DALEXtra` package [214], test data and the final model fit to training data [127]. In classification, the ALE plot shows the predicted probability of a stroke occurrence under different values for each feature [214].

3 Results

The following section will present the results for each step of analysis and modelling for the prediction of stroke occurrence as a binary classification problem using ML methods. This will include findings from data exploration and pre-processing followed by the results of model tuning, evaluation, selection and interpretation. Numeric results will be reported as rounded to 3 decimal places however proportions will generally be reported as percentages to 1 decimal place.

3.1 Data exploration and preprocessing

The dataset consisted of 4,981 patients of which only 248 (5.0%) had experienced a stroke occurrence whilst 4,733 (95.0%) had not. The low event rate for this binary outcome made it suitable for oversampling in order to address the class imbalance when training the ML classifiers. Oversampling using SMOTE was applied across all ML methods only to the training data; within the 10-fold cross-validation process used to evaluate models, oversampling was also exclusively applied to the training folds and never to the validation folds.

As training data made up 80% of the complete dataset it represented a total of 3,984 patients, of which 195 (4.9%) had experienced a stroke occurrence. Therefore, the remaining 20% of test data represented a total of 997 patients, of which 53 (5.3%) had experienced a stroke occurrence. For the training data, an oversampling ratio of 1 was used in order to generate an additional 3,594 synthetic examples of the stroke minority class to allow for an equal number of examples in each class ahead of model training. Therefore the training dataset contained 7,578 examples comprised of 3,789 (50.0%) patients of each class. The default of 5 nearest-neighbours was used within the SMOTE algorithm in order to generate these new minority class examples [90]. Additional pre-processing steps such as encoding and normalisation were applied to test and train data for MLP and SVM methods only whilst XGBoost required encoding of categorical variables. RF required no additional pre-processing. The following data exploration sections below describe the full dataset before splitting into training and test datasets and SMOTE application.

A total of 10 demographic, lifestyle and clinical characteristics of patients were included as features; age (years), gender, smoking status, marital status, urban/rural residence, occupation sector, hypertension, heart disease, average blood glucose (mg/dL) and BMI (kg/m^2). These variables contained no missingness or apparent erroneous values on inspection using descriptive summaries. Right-skewed distributions were evident for average blood glucose and BMI whilst age demonstrated a left-skewed distribution (Figure 1). The median (interquartile range (IQR)) average blood glucose was 91.850 (77.230-113.860) mg/dL, 28.100 (23.700-32.600) kg/m^2 for BMI and 45 (25-61) years for age. As these variables were positive a Box-Cox transformation was used to rescale to a more symmetrical distribution using lambda estimates of -1.060, 0.368 and 0.840 for average blood glucose, BMI and age respectively (Figure 1). Despite this, a bimodal distribution persisted for average blood glucose and age after transformation.

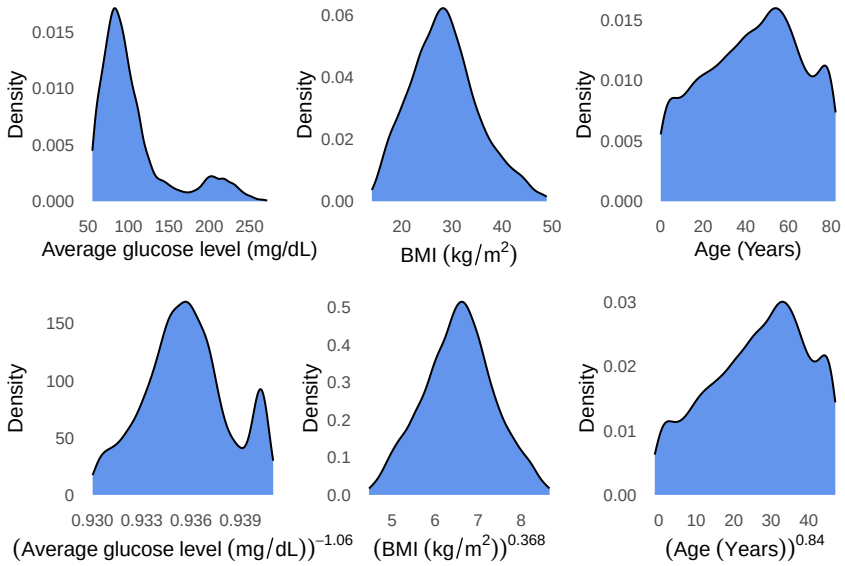

Figure 1: Density plots for the distribution of average blood glucose, BMI and age before (top) and after (bottom) Box-Cox transformation

Table 1: Demographic, lifestyle and clinical characteristics of patients within the sample split by stroke occurrence outcome. Age is displayed in years, average glucose level in milligrams per decilitre and BMI in kilograms per square metre. Summary statistics are provided for continuous variables including minimum (Min), maximum (Max), median (Med) and interquartile range (IQR), mean and standard deviation (std), total number of observations (N) and the number of missings (NA). Categorical variables are presented as 'number of observations (column proportion as percentage)'

Variable	Level	Stroke	No stroke	Total
Gender	Female	140 (56.5%)	2767 (58.5%)	2907 (58.4%)
	Male	108 (43.5%)	1966 (41.5%)	2074 (41.6%)
Age	Min / Max	1.320 / 82.000	0.080 / 82.000	0.080 / 82.000
	Med [IQR]	71.000 [59.000;78.000]	43.000 [24.000;60.000]	45.000 [25.000;61.000]
	Mean (std)	67.820 (12.671)	42.141 (22.345)	43.420 (22.663)
	N (NA)	248 (0)	4733 (0)	4981 (0)
Hypertension	No	182 (73.4%)	4320 (91.3%)	4502 (90.4%)
	Yes	66 (26.6%)	413 (8.7%)	479 (9.6%)
Heart disease	No	201 (81.0%)	4505 (95.2%)	4706 (94.5%)
	Yes	47 (19.0%)	228 (4.8%)	275 (5.5%)
Ever married	No	29 (11.7%)	1672 (35.3%)	1701 (34.1%)
	Yes	219 (88.3%)	3061 (64.7%)	3280 (65.9%)
Work type	Children	2 (0.8%)	671 (14.2%)	673 (13.5%)
	Government	33 (13.3%)	611 (12.9%)	644 (12.9%)
	Private	148 (59.7%)	2712 (57.3%)	2860 (57.4%)
	Self-employed	65 (26.2%)	739 (15.6%)	804 (16.1%)
Residence type	Rural	113 (45.6%)	2336 (49.4%)	2449 (49.2%)
	Urban	135 (54.4%)	2397 (50.6%)	2532 (50.8%)
Avg. glucose level	Min / Max	56.110 / 271.740	55.120 / 267.760	55.120 / 271.740
	Med [IQR]	105.040 [79.573;195.960]	91.450 [77.120;112.620]	91.850 [77.230;113.860]
	Mean (std)	132.176 (61.771)	104.569 (43.602)	105.944 (45.075)
	N (NA)	248 (0)	4733 (0)	4981 (0)
BMI	Min / Max	16.900 / 48.900	14.000 / 48.900	14.000 / 48.900
	Med [IQR]	29.450 [26.975;32.650]	28.000 [23.500;32.600]	28.100 [23.700;32.600]
	Mean (std)	30.187 (5.658)	28.410 (6.834)	28.498 (6.790)
	N (NA)	248 (0)	4733 (0)	4981 (0)
Smoking status	Formerly smoked	70 (28.2%)	797 (16.8%)	867 (17.4%)
	Never smoked	89 (35.9%)	1749 (37.0%)	1838 (36.9%)
	Smokes	42 (16.9%)	734 (15.5%)	776 (15.6%)
	Unknown	47 (19.0%)	1453 (30.7%)	1500 (30.1%)

From the totals shown in Table 1 for the full dataset, most patients in this sample were female (58.4%), did not have hypertension (90.4%) or heart disease (94.5%), have been married (65.9%) and work in the private sector (57.4%). Most patients had also never smoked (36.9%) however there was also a large proportion that had an unknown smoking status (30.1%). Residence type was evenly split between urban and rural. However, the distribution of these features differed for patients that had experienced a stroke occurrence compared to patients who had not (Table 1).

As shown in the density plots in Figure 2, patients who experienced a stroke tended to be over 40 years old with a higher proportion aged 60 to 80 years, and also more likely to have a higher average blood glucose level between 150 and 270 mg/dL. These patients were also more likely to be self-employed and former smokers (Table 1).

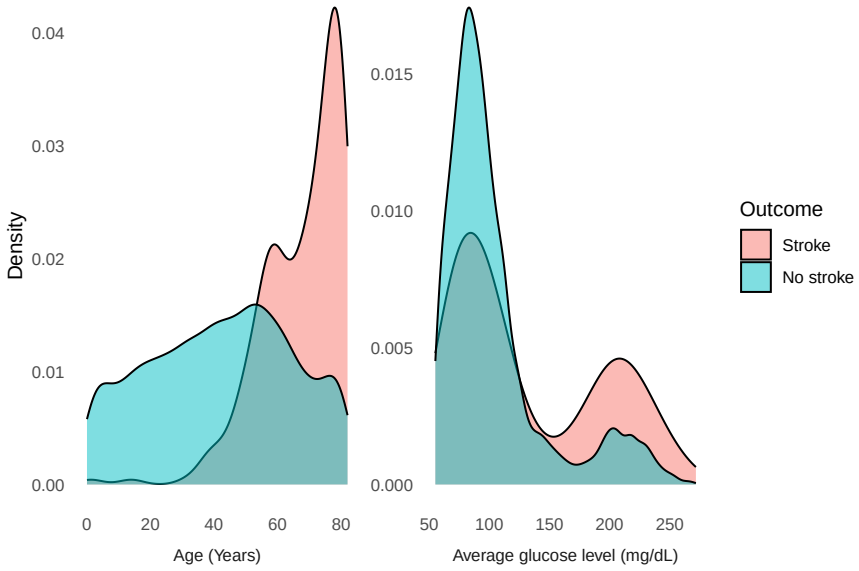

Figure 2: Density plots of age (years) and average glucose level (mg/dL) by stroke outcome

Weak correlations were observed between age, BMI and average blood glucose with the strongest observed between age and BMI (0.374) and age and average glucose level (0.234). This can be inferred from scatter plots shown in Figure 3 where it appeared BMI generally increased with age and higher levels of average blood glucose tended to occur in older patients. These plots also

demonstrated that stroke occurrence is more likely with increasing age across different values of both BMI and average blood glucose. The low degree of correlation provided confidence that these features were not redundant and would not introduce instability in the modelling process [67].

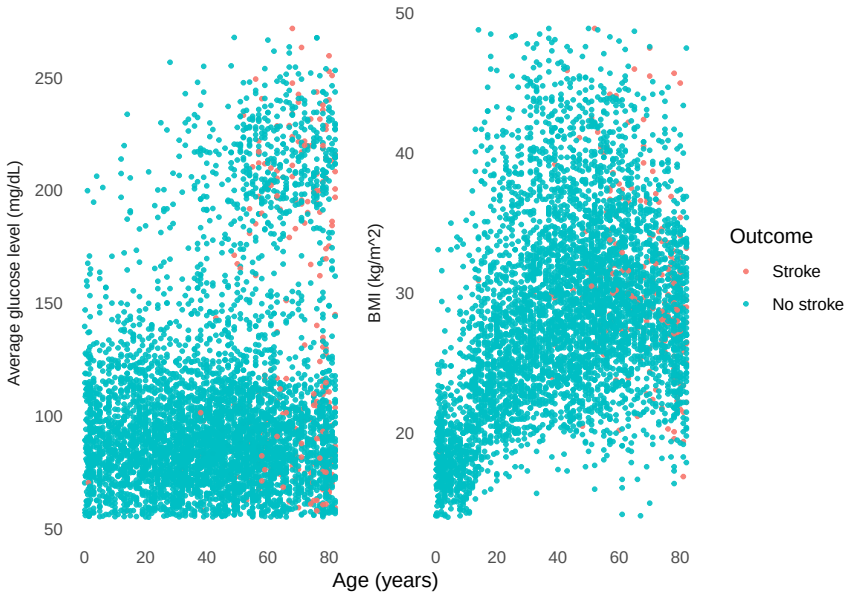

Figure 3: Scatter plots for average glucose level (mg/dL) and BMI (kg/m^2) against age (years) by stroke outcome

3.2 Hyperparameter tuning

The tuning process involved a grid-search using a space-filling design implemented using Latin hypercube sampling which generated 50 candidate hyper parameter combinations for each ML method. Ten-fold cross-validation was used within the racing tuning process with a burn-in of three initial resamples to evaluate every set of candidate hyperparameters before excluding those not significantly different (at a 5% significance level) from the best candidate. For each resample thereafter up to the full ten folds, candidates continued to be excluded until only the most performant combinations remained [155]. The baseline AUPRC if a random classifier was used on the cross-validation training data would be equal to 0.049 as this was the proportion of stroke cases in the training dataset [207], providing an estimate for model performance comparison across ML methods. For final model evaluation, the baseline AUPRC would be equal to 0.053 as this was the proportion of stroke cases

in the test dataset. The values of some hyperparameters are presented in tables and figures on a transformed scale and will be discussed in terms of this transformation where these are referenced; the transformer will be identified in parentheses alongside the corresponding hyperparameter in each figure if applicable. All other references to hyperparameter values can be presumed to be in their original scale.

3.2.1 Multi-layer perceptron

Of the 50 candidate hyperparameter combinations used, 34 were excluded during the racing process for MLP leaving 16 best performing candidates based on mean AUPRC. Candidate model performance across each resample during the racing process is shown as an example in Figure 4, with each line representing a unique hyperparameter combination. Models are compared from the third resample onwards leading to a large number being initially excluded.

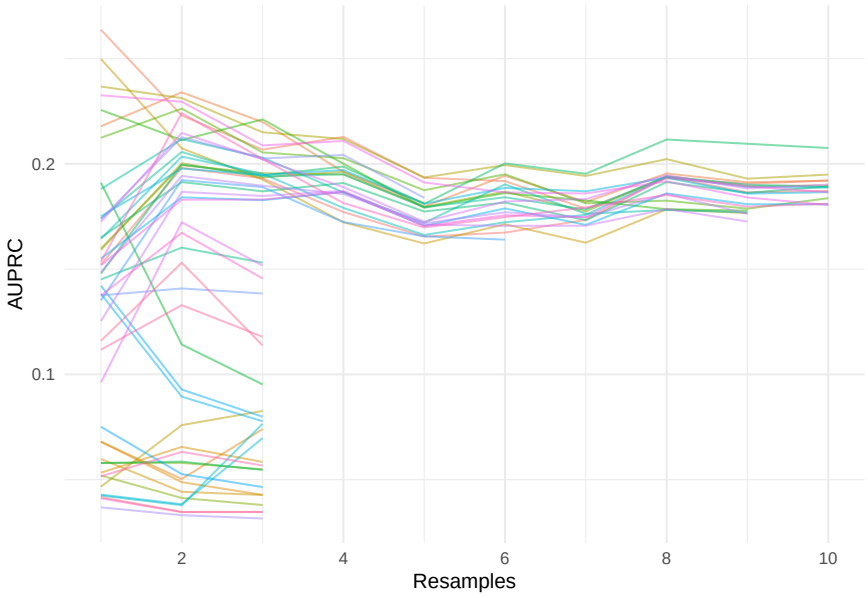

Figure 4: Model performance (AUPRC) for 50 MLP model candidates (hyperparameter combinations) at each subsequent resample during racing tuning

Using mean AUPRC as the performance metric, the tuning results of all 50 candidate models were examined to establish optimal configurations for the four hyperparameters selected for tuning as

shown in Figure 5; lighter points indicated poorer combinations whilst only the darkest points corresponding to resample 10 were retained after the tuning process indicating more successful combinations.

Dropout rates of 0.7 or more tended to perform poorly, often being excluded early in the race tuning process whilst transformed values of learning rate below -2 performed the worst (Figure 5). Classifier performance appeared to vary across the range of 0 to 500 epochs and from 1 to 10 hidden nodes.

The five best performing hyperparameter combinations are displayed in Table 2 and demonstrated generally lower dropout rates, higher learning rates and widely varying numbers of epochs and hidden nodes. The performance of these models were very similar with overlapping AUPRC 95% confidence intervals (CI), therefore the model with the highest mean AUPRC was selected at 0.208 (95% CI: 0.166, 0.249). This also corresponded to the simplest model with only 1 hidden node and the lowest number of epochs out of the five best performing models which was favourable in preventing overfitting [165]. The optimal MLP model was therefore configured with 1 hidden node, trained for 87 epochs, a transformed learning rate of -1.634 and a dropout rate of 0.187 (Table 2).

Table 2: Five best candidate MLP models and hyperparameter combinations based on AUPRC with 95% confidence interval (CI)

Hidden nodes	Dropout	Epochs	Learning rate (log-10)	AUPRC (95% CI)
1	0.187	87	-1.634	0.208 (0.166, 0.249)
9	0.143	123	-1.447	0.195 (0.165, 0.225)
3	0.291	228	-0.878	0.192 (0.155, 0.229)
6	0.100	437	-1.109	0.192 (0.151, 0.233)
2	0.252	270	-1.320	0.190 (0.157, 0.223)

After fitting the optimal configured model to the resampled training data, the mean AUPRC performance on the non-oversampled validation folds was 0.180 (95% CI: 0.149, 0.210) (Table 6). The model therefore outperforms a random classifier with an AUPRC of 0.049 however demonstrates insufficiency in balancing precision and recall across all decision thresholds as perfect performance

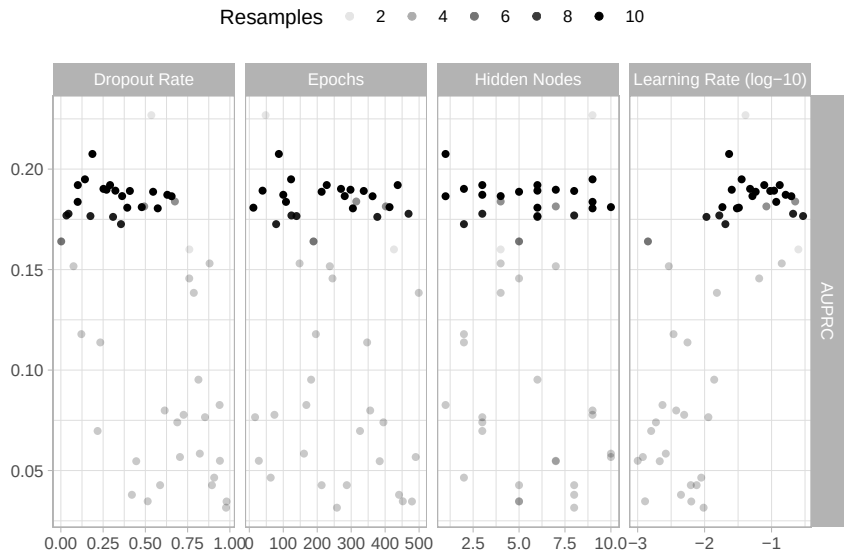

Figure 5: MLP hyperparameter values and corresponding performance given by AUPRC for 50 hyperparameter combinations

is given by a AUPRC of 1.

A decision threshold of 0.650 was found to produce an optimal F1 score of 0.255 for this model as shown in Table 6. At this threshold, recall for this model was higher than precision with 52.3% of actual positives correctly predicted, whilst only 16.9% of all positive predictions were truly positive (Table 6). An example of a confusion matrix is shown in Figure 6 and illustrates class predictions when this threshold was applied using the validation fold predictions from training dataset resamples. Therefore at this decision threshold, the model performance was characterised by almost as many false negatives as true positives and five times as many false positives than true positives at this threshold (Figure 6).

3.2.2 Support vector machine

SVM returned fewer candidate hyperparameter combinations from race tuning than MLP with 10 out of 50 remaining by the last resample. The performance of the two hyperparameters selected for tuning are shown in Figure 7 and provide an insight into the optimal configurations for SVM in

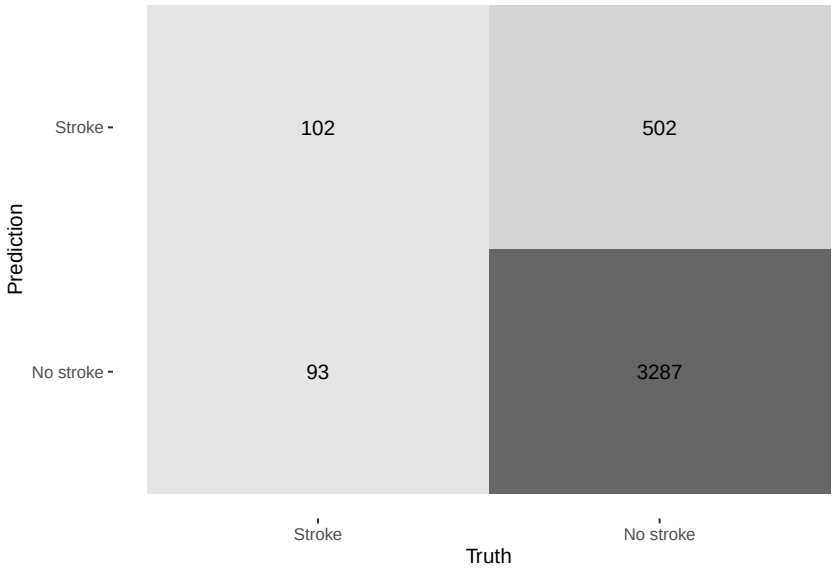

Figure 6: Confusion matrix for the optimal MLP model using decision threshold of 0.650 on cross-validation predictions to optimise F1 score

this context. For the cost hyperparameter, the best performance appeared to occur at transformed values of approximately -5 to -2.5 and 9 to 14 however there was variability with approximately four models with similar values that performed poorly and were excluded during race tuning. A transformed value of sigma between -5 and -2.4 appears to be optimal for bringing about the best performance whilst values greater than approximately -2.4 appear to lead to a large drop in performance (Figure 7).

The five best performing combinations are displayed in Table 3 and demonstrated that lower values of sigma were often in combination with larger values of cost and vice versa; there also did not appear to be large differences in performance across these combinations suggesting multiple ranges, particularly of cost, could be used. The performance of these models were again very similar with overlapping mean AUPRC 95% confidence intervals. Therefore the model with the highest mean AUPRC was selected at 0.195 (95% CI: 0.167, 0.222) resulting in the optimal SVM model configured with a transformed cost of -2.743 and a transformed sigma value of -2.448 (Table 3).

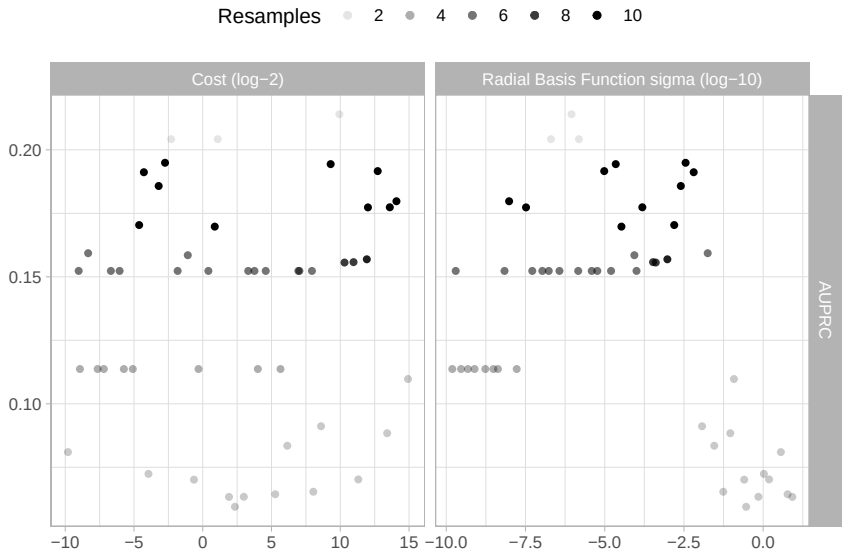

Figure 7: SVM hyperparameter values and corresponding performance metrics for 50 hyperparameter combinations

Table 3: Five best candidate SVM models and hyperparameter combinations based on AUPRC with 95% confidence interval (CI)

Cost (log-2)	RBF Sigma (log-10)	AUPRC (95% CI)
-2.743	-2.448	0.195 (0.167, 0.222)
9.302	-4.652	0.194 (0.156, 0.233)
12.721	-5.015	0.192 (0.153, 0.230)
-4.283	-2.192	0.191 (0.163, 0.219)
-3.207	-2.599	0.186 (0.159, 0.212)

The optimal SVM model performance was similar to that of MLP after being fit to the resampled training data with a mean AUPRC of 0.195 (95% CI: 0.167, 0.222), again indicating the model outperforms a random classifier however is not effective in balancing precision and recall across all thresholds (Table 6).

A decision threshold of 0.770 was found to produce an optimal F1 score of 0.276 for this model as shown in Table 6, higher than the point estimate obtained for the MLP model at its corresponding

threshold. At a threshold of 0.770, recall was again higher than precision with 48.2% of actual positives correctly predicted, whilst 19.4% of all positive predictions were truly positive (Table 6). Therefore, as recall was lower than for MLP, the SVM model demonstrated a greater optimal F1 score due to greater precision but remains characterised by poor prediction of actual stroke occurrences and a large number of false positives at this threshold.

3.2.3 XGBoost

XGBoost returned the greatest number of candidate hyperparameter combinations from race tuning out of all ML methods with 20 remaining models by the last resample out of 50. XGBoost had a total of eight hyperparameters selected for tuning, the most out of all ML methods included in the analysis. The performance of each hyperparameter is shown in Figure 8. Initial tuning was implemented using a log10 transformed learning rate with a default lower bound of -10 however the resulting models produced predicted probabilities of 0.5 for all observations indicating the learning rate may have been too small given the number of boosting rounds to learn patterns from the data. This was improved by increasing the lower bound of the learning rate range to -4 ahead of subsequent tuning.

Increasing values of learning rate tended to improve performance (Figure 8). A weak negative relationship was observed between performance and an increasing number of randomly sampled predictors (mtry) however there was greater variability between 7 to 9 predictors with the highest end-of-race performance observed at a value of 8 (Figure 8). A more subtle negative relationship between performance and increasing numbers of stopping iterations was observed with the optimal range between 5 and 10 which could prevent the algorithm being stopped too early [185]. Model performance was relatively stable across the full tuning range of 1 to 2000 trees except poorer performance observed for 750 to 1000 trees.

A weak positive relationship was observed in performance as loss reduction increased; a higher degree of regularisation tended to perform best with transformed values greater than -5 demonstrating

the optimum (Figure 8). A large amount of variability in performance was observed with increasing minimum node size however values greater than 30 appeared to perform best. The proportion of training examples subsampled demonstrated relatively stable performance between approximately 50.0% to 75.0% after which there was an evident decline. A tree depth of 5 performed best and generally depths of 6 splits or more were associated with inferior performance.

The five best performing combinations are displayed in Table 4 and demonstrated generally higher mtry, greater than 1000 trees, a minimum node size of greater than 30, a tree depth of 5 or less, higher learning rates, higher loss reduction, lower sample size proportion and varying numbers of stopping iterations. The performance of these models were again very similar with overlapping mean AUPRC 95% confidence intervals therefore the simplest model given by the one with the fewest trees was selected demonstrating a mean AUPRC of 0.186 (95% CI: 0.143, 0.228). The optimal XGBoost model was therefore configured with an mtry of 7, 322 trees, minimum node size of 38, tree depth of 3 splits, transformed learning rate of -0.724, transformed loss reduction of -7.600, sample size proportion of 78.2% and 20 stopping iterations (Table 4).

Table 4: Five best candidate XGBoost models and hyperparameter combinations based on AUPRC with 95% confidence interval (CI). mtry = the number of randomly selected predictors, Min n = minimum node size, Depth = tree depth, L.rate = learning rate, L.red = loss reduction, N = sample size, Iter = stop iterations

mtry	Trees	Min n	Depth	L.rate (log-10)	L.red (log-10)	N	Iter	AUPRC (95% CI)
8	1091	35	5	-1.321	-4.818	0.592	7	0.192 (0.151, 0.233)
8	1667	36	7	-1.379	0.562	0.691	10	0.188 (0.147, 0.230)
3	1614	32	3	-1.611	-1.460	0.744	9	0.186 (0.150, 0.222)
7	322	38	3	-0.724	-7.599	0.782	20	0.186 (0.143, 0.228)
8	1241	40	5	-2.169	-2.103	0.635	17	0.185 (0.146, 0.224)

The optimal XGBoost model performed simlarly to MLP and SVM; after being fit to the resampled training data it demonstrated a lower mean AUPRC of 0.176 (95% CI: 0.130, 0.221) but still had overlapping performance with the aforementioned optimal models given by the wide 95% confidence

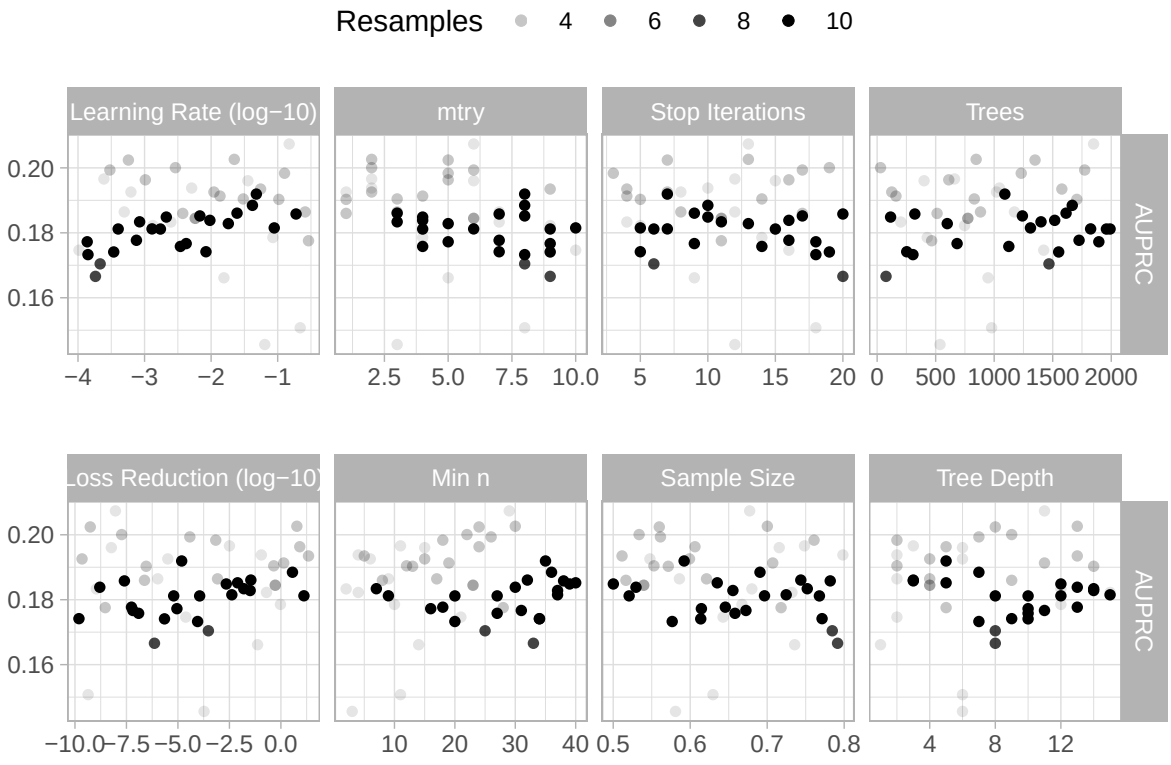

Figure 8: XGBoost hyperparameter values and corresponding performance metrics for 50 hyperparameter combinations

interval (Table 6).

The optimal F1 score was 0.240 (Table 6) and found to occur at a much lower decision threshold of 0.180 compared to both MLP and SVM. The optimal F1 score was lower than the point estimates obtained from MLP and SVM corresponding to a lower recall with 41.5% of actual positives correctly predicted, whilst precision was similar to MLP with 16.9% of all positive predictions truly positive (Table 6). Therefore, the XGBoost model demonstrated a lower F1 score due to lower recall at its corresponding optimal F1 threshold and therefore poorer prediction of actual stroke occurrences at this threshold.

3.2.4 Random forest

RF returned the fewest candidate hyperparameter combinations from race tuning out of all ML methods with only 1 remaining model out of 50; most models were excluded immediately at the third resample followed by two more at resample four leaving a single candidate model (Figure 9).

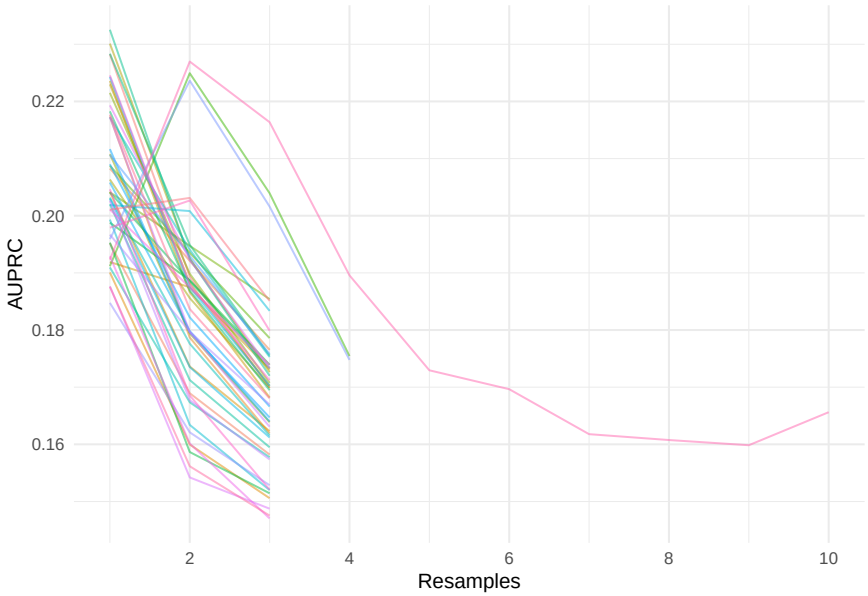

Figure 9: Model performance (AUPRC) for 50 RF model candidates (hyperparameter combinations) at each subsequent resample during racing tuning

The performance of the three hyperparameters selected for tuning are shown in Figure 10. As only a single candidate model remained, discussion of optimal hyperparameter ranges will include omitted configurations where trends of patterns in performance are evident. For example, there appeared to be a strong positive relationship between increasing minimum node size and performance therefore fewer splits and complexity appeared to improve model performance. A weak negative relationship was observed between performance and an increasing number of randomly sampled predictors (mtry) with the highest performance between 1 and 3 predictors and greater variability in performance between 5 to 10 predictors (Figure 10). Model performance was very varied across the full tuning range of 1 to 2000 trees however better performing configurations tended to have fewer than 750 trees.

The best and only remaining hyperparameter combination is displayed in Table 5 demonstrating an AUPRC of 0.166 (95% CI: 0.134, 0.197). Therefore the optimal RF model was configured with 1 randomly selected predictor (mtry), 243 trees and a minimum node size of 36 (Table 5).

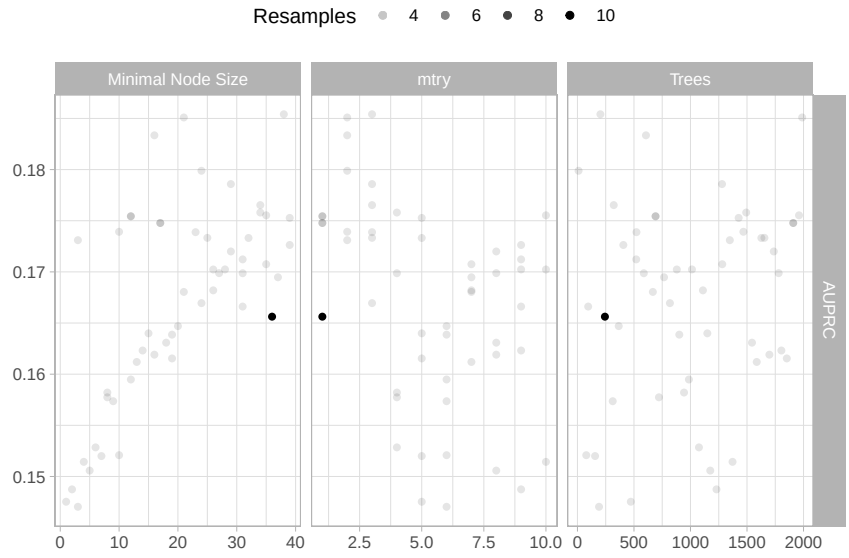

Figure 10: RF hyperparameter values and corresponding performance metrics for 50 hyperparameter combinations

Table 5: Five best candidate RF models and hyperparameter combinations based on AUPRC with 95% confidence interval (CI)

mtry	Trees	Min n	AUPRC (95% CI)
1	243	36	0.166 (0.134, 0.197)

After being fit to the resampled training data, the optimal RF model had the lowest mean AUPRC of all ML methods at 0.164 (95% CI: 0.132, 0.195) however still demonstrated overlap given by the wide 95% confidence interval (Table 6).

Similar to XGBoost, a lower threshold of 0.370 was found to optimise F1 score obtaining a higher point estimate of 0.248 (Table 6). The optimal F1 score was lower than the point estimates obtained from MLP and SVM and the model demonstrated the lowest precision of all models at its optimal F1 threshold with 16.0% of all positive predictions truly positive (Table 6). Despite this, recall was higher than for all other models with 54.4% of actual positives correctly predicted (Table 6). Therefore, although the RF model demonstrated greater recall, as with the all the models selected, it remains characterised by poor prediction of actual stroke occurrences and a large number of false positives at this threshold.

3.3 Stacking ensemble

A stacking model was constructed using all candidate models that remained after the race tuning process for all four model definitions; MLP, SVM, XGBoost and RF. The ensemble was therefore initialised with 47 candidate learners; 16 MLP configurations, 10 SVM configurations, 20 XGBoost configurations and 1 RF configuration. The out-of-sample predictions from 10-fold cross-validation for all candidates were passed to a regularised generalised linear model (logistic regression) as a meta-learner.

A lasso penalty of 0.01 was selected from a range of 0.0001 to 0.1 by the stacking algorithm to maximise the AUPRC performance and minimise the number of candidate members. The stacking algorithm then combined predictions and generated coefficients for each candidate learner. Out of a possible 47 candidates only 4 were retained by the stacking model including two SVM models, one XGBoost model and one MLP model, each with a non-zero coefficient as shown in Figure 11. The SVM models had the largest coefficients at 1.570 and 1.260 and therefore made the greatest contribution to stacking predictions, followed by the XGBoost model (1.130) and the MLP model which had a much lower weighting (0.286).

The stacked model was evaluated using the resulting predictions from aggregating the base learner

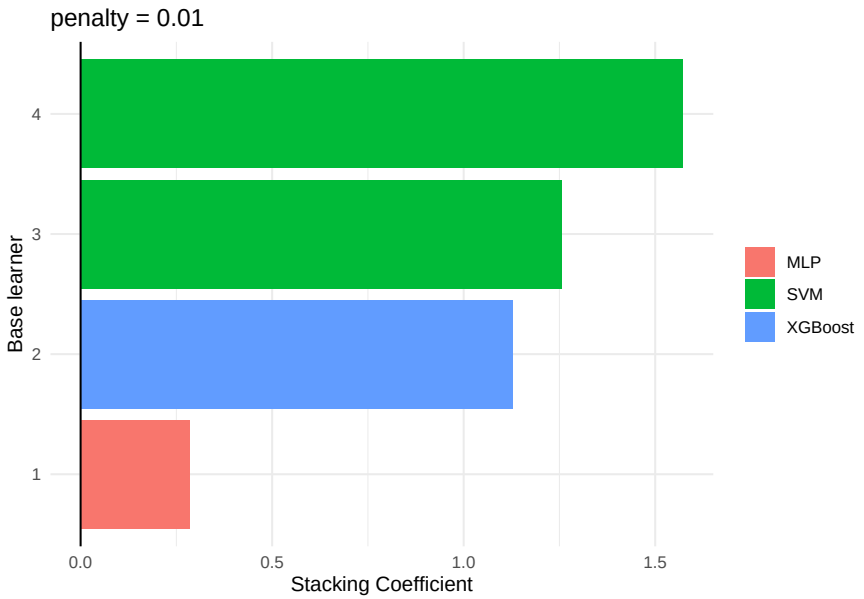

Figure 11: Stacked model coefficients for four base learner models using a lasso penalty of 0.01

out-of-sample predictions in the training set, therefore only a point estimate for performance was obtained. The ensemble demonstrated very similar performance to the optimal individual ML models with an AUPRC of 0.184 indicating it was better than a random classifier but also not effective in balancing precision and recall across decision thresholds (Table 6).

A much lower decision threshold of 0.100 was used to optimise F1 score compared to any of the individual ML models, obtaining a slightly higher F1 score of 0.282 (Table 6). At its optimal F1 score threshold, recall was higher than for any individual model with 62.6% of actual positives correctly predicted whilst precision was lower than that for SVM with 18.2% of all positive predictions truly positive (Table 6). Therefore, the stacked models performance was not dissimilar to the individual base learner models and similarly is characterised by relatively poor prediction of actual stroke occurrences and a high proportion of false positives at the optimal F1 threshold.

3.4 Final model selection and evaluation

After optimally tuned models were selected for each algorithm they were compared using their respective mean AUPRC performance on resamples. A summary is shown in Table 6 including the performance of the stacked model alongside all individual ML models with point estimate metrics at the F1 optimal threshold for each classifier. All classifiers exceeded the baseline random classifier performance of 0.049 and there was limited variability in performance across models with overlap across all 95% confidence intervals for mean AUPRC. Of the individual models, RF had the lowest mean AUPRC estimate at 0.164 (95% CI: 0.132, 0.195) whilst SVM demonstrated the highest mean estimate at 0.195 (95% CI: 0.167, 0.222).

The stacked model demonstrated very similar performance with an AUPRC of 0.184 based on out-of-sample predictions from cross-validation. For the purposes of this analysis, a single base learner model and the stacked model were evaluated on the test dataset to compare performance. SVM was selected as the individual model as it obtained the highest mean AUPRC on the training dataset, however as there was overlap in performance of all models justification could be made for selecting other models for the final evaluation step.

Table 6: Final performance metrics on the training dataset for each single model from 10 cross-validation resamples and stacked model from out-of-sample predictions from cross-validation. Mean AUPRC is shown along with a 95% confidence interval (CI) whilst F1, Recall and Precision are based on a corresponding decision thresholds found to optimise F1 score. No confidence interval is provided for the stacked model as it was calculated on aggregated out of sample predictions

Model	AUPRC (95% CI)	F1	Precision	Recall
MLP	0.180 (0.149, 0.210)	0.255	0.169	0.523
SVM	0.195 (0.167, 0.222)	0.276	0.194	0.482
XGBoost	0.176 (0.130, 0.221)	0.240	0.169	0.415
RF	0.164 (0.132, 0.195)	0.248	0.160	0.544
Stacked	0.184	0.282	0.182	0.626

Therefore the SVM model and the stacked model were fit to the test dataset to obtain estimates of performance on unseen data. The results are shown in Table 7 and highlight almost identical performance between both models with an AUPRC of 0.180 and 0.176 for SVM and the stacked

model respectively. The optimal F1 scores were based on a threshold of 0.740 and 0.130 for the SVM and stacked model respectively (Table 7), and again highlight the similarity between models; F1 score is very similar however SVM presented slightly greater recall and reduced precision compared to the stacked model at their respective thresholds. This is also illustrated in the confusion matrices shown in Figure 12 where SVM demonstrates marginally more true positives and fewer false negatives at the cost of more false positives for patients in the test set at each models respective decision threshold to optimise F1 score.

Table 7: SVM and stacked model performance on the test dataset. AUPRC is shown whilst F1, Recall and Precision are based on decision thresholds of 0.740 and 0.130 for the SVM and stacked model respectively, found to optimise F1 score

Model	AUPRC	F1 score	Precision	Recall
SVM	0.180	0.297	0.204	0.547
Stacked	0.176	0.304	0.220	0.491

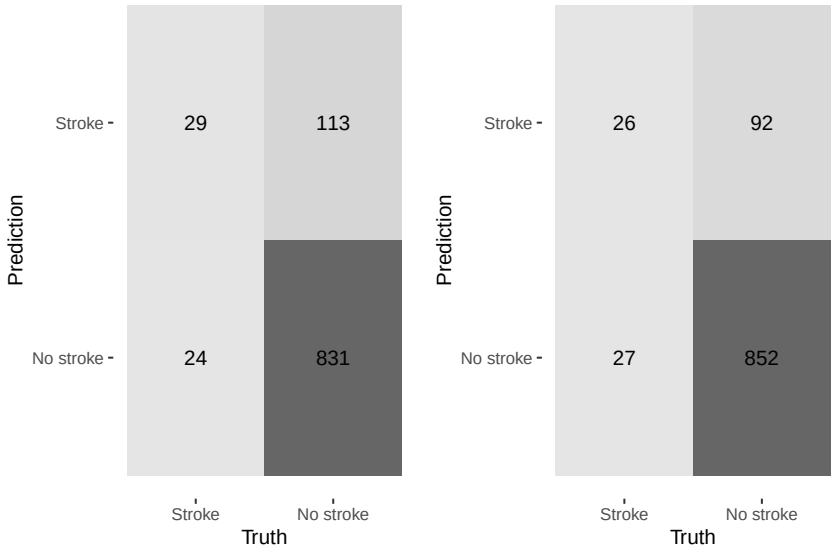

Figure 12: (Left) Confusion matrix for the final SVM model using decision threshold of 0.740 and (Right) final stacked model using decision threshold of 0.130 on test dataset predictions to optimise F1 score

The results obtained confirmed the models performed better than random in predicting stroke occurrence with AUPRC exceeding 0.053 obtained from the proportion of patients that had expe-

rienced a stroke in the test dataset. However, low AUPRC for both models indicated insufficiency in balancing precision and recall across all decision thresholds as illustrated in the PRCs with both models demonstrating a similar pattern (Figure 13). A well-performing model would demonstrate curves close to the top right corner of the plot, indicating a AUPRC of close to 1 (Figure 13).

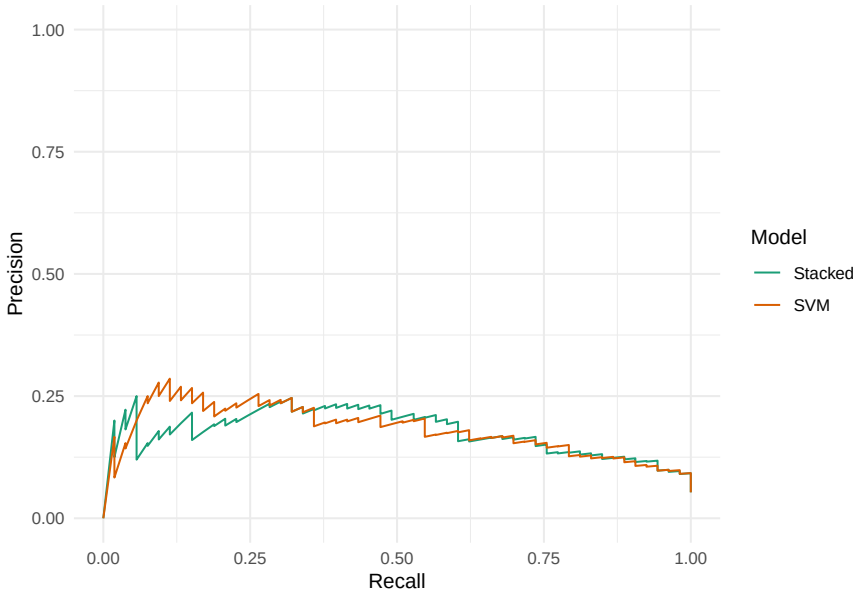

Figure 13: Precision recall curves (PRC) for SVM and stacked model on the test dataset

The stacked model demonstrated a slight decrease in AUPRC performance on the test set which could suggest potential overfitting to the training data leading to a decline in performance on unseen data, however the performance of SVM appeared more robust [216]. Therefore, the final model selected for model interpretation was the SVM model as it demonstrated no evidence of overfitting on the test dataset.

3.5 Model interpretation plots

3.5.1 Feature importance

After selecting SVM as the best performing model, permutation-based feature importance was applied using the test dataset across 100 permutations for each feature. The resulting feature importance plot is shown in Figure 14 and shows the mean loss in 1-AUPRC after permuting each feature given by the length of each bar and a box plot for its associated variability over 100 permutations. The vertical dashed line corresponds to the baseline loss function value of 0.820 1-AUPRC for the full model on the test data (Figure 14).

The plot indicates the most important variable in the models predictions for stroke occurrence was patient age with a mean 1-AUPRC loss of 0.925 (Figure 14). Age, average glucose level, hypertension and heart disease demonstrated most variability in mean loss across the 100 permutations, however all except age cross the baseline loss performance line indicating they are not considered important features for predictions in this model. Similarly, all other variables mean loss distributions cross the baselines loss value indicating that the final SVM model relies on age as the only important feature for predicting stroke occurrence.

3.5.2 Accumulated local effects

The effect of different values of each feature without the influence of other correlated features on the predicted probability of stroke occurrence was demonstrated using ALE plots. The effect of age is shown in Figure 15; the average predicted stroke probability followed a positive curvilinear relationship with increasing age. The predicted stroke probability was generally low on average up to the age of 20 before increasing rapidly up to the age 40 beyond which it increases linearly up to the age of 80. The ALE plot for age demonstrated the highest predicted stroke probabilities observed across all features predominantly for those aged approximately 55 and over on average.

Increasing values of average blood glucose between 50 and 125 mg/dL were associated with sharply

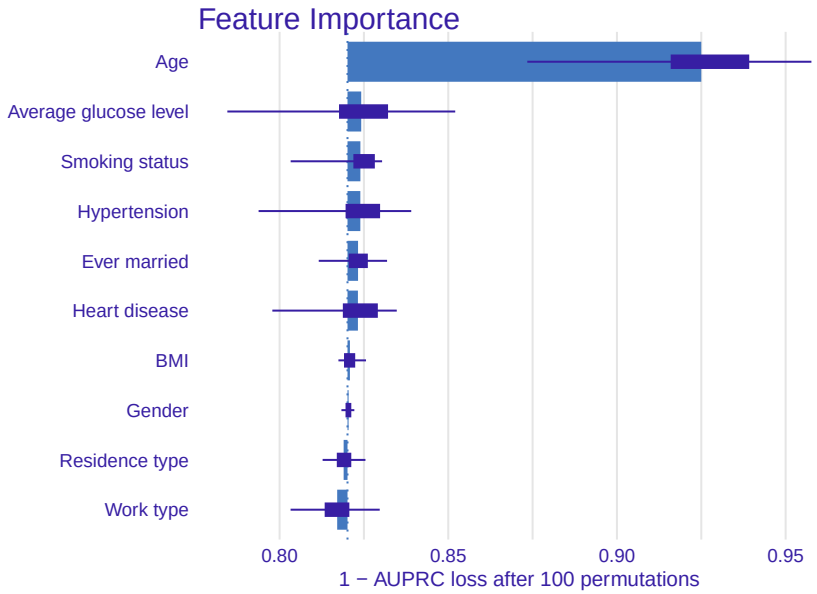

Figure 14: Feature importance plot for the final SVM model based on 100 permutations of each feature against the mean 1-AUPRC loss

increasing predicted stroke probability of 10% on average (Figure 15). Beyond this, predicted stroke risk continued to increase more gradually over 125 mg/dL to 275 mg/dL by a further 5% on average. The predicted stroke probability appeared to remain relatively level with increasing BMI on average although there appears to be a slight peak at a BMI of 30 kg/m^2 (Figure 15).

ALE plots for the categorical features are presented in Figure 16. The average predicted probabilities of stroke for each isolated feature were generally low with the highest occurring for the presence of heart disease and hypertension. The presence of heart disease and hypertension compared to no disease demonstrated the greatest increases in average predicted probability of stroke with smaller effects observed for having been married and urban residential types (Figure 16). Lower predicted probabilities occurred on average for patients with government jobs whilst children and the self-employed had slightly higher predicted stroke probability. Similarly, having never smoked and those with unknown smoking status had lower predicted stroke probability on average whilst former and current smokers had a higher predicted probability (Figure 16).

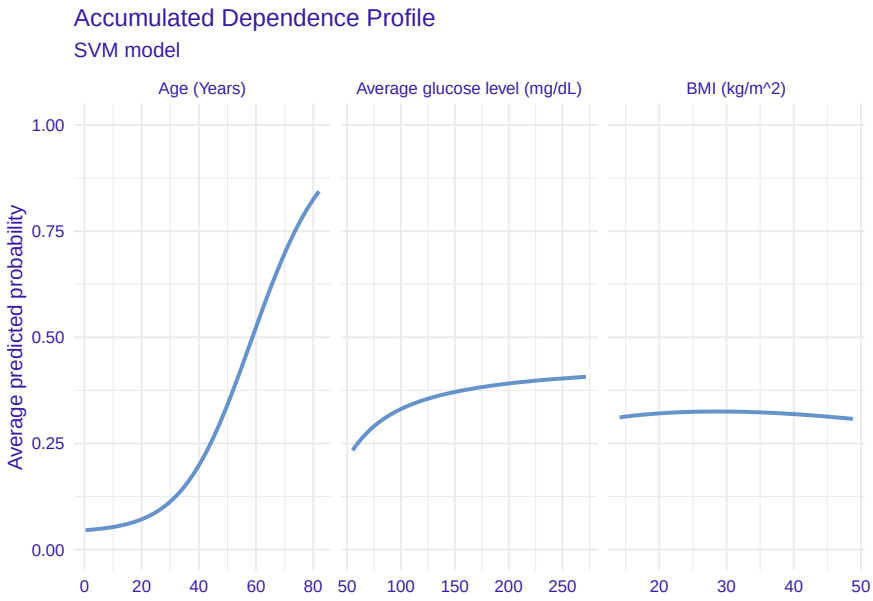

Figure 15: Accumulated local effects (ALE) plot for continuous features using the test dataset and the final SVM model

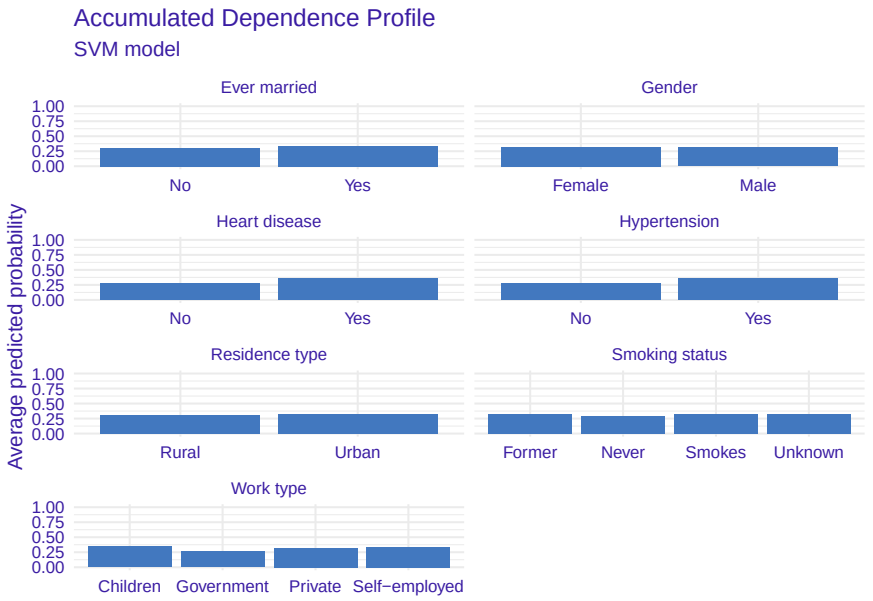

Figure 16: Accumulated local effects (ALE) plot for categorical features using the test dataset and the final SVM model

4 Discussion

The increasing health and economic burden of stroke [8], particularly in LMICs [6], is a global public health priority [9] and poses a significant challenge for high-risk strategy primary prevention [31, 32] via early identification of those at risk. Predicting stroke from risk factors using readily available patient-level information has already demonstrated promising results [30, 41] with recent attempts to build on conventional statistical regression-based models using machine learning methods [36].

Using a highly imbalanced open-access dataset [60], this project aimed to compare four ML algorithms and a stacking model in the prediction of stroke as a binary classification problem from ten demographic, lifestyle and clinical features. Multilayer perceptron (MLP), support vector machine (SVM), gradient boosted decision trees (XGBoost) and random forest (RF) models were optimised using hyperparameter tuning before comparing performance individually and with a stacking ensemble that enlisted MLP, SVM and XGBoost as base learners. From this training process the final single model was selected alongside the stacking model and both were applied to the test data for final model evaluation.

There was overlap in predictive performance on training resamples across all models (Table 6) therefore model selection was guided by the single model that obtained the greatest mean AUPRC estimate. Therefore, SVM was selected however it is likely other models could have been used with comparable success on the final test dataset. The overlap in performance included the stacked model which is contrary to the established literature using these methods on the same dataset [53].

Stacking has previously been shown to outperform its component base learners; Dritsas and Trigka[54] found a stacking model with a logistic regression meta-learner comprised of four different base learners demonstrated superior performance in stroke prediction to single models across seven different algorithms including MLP and RF. Similarly, Hassan et al[55] compared 10 single ML algorithms (including MLP, XGBoost and RF) to a stacking ensemble with a RF meta-learner and the remain-

ing 9 models as base learners and found it outperformed all individual classifiers.

Stacking is typically optimised when base-learners are diverse in their structure and perform well individually [52]. In this project a combination of model structures were utilised including artificial neural network, maximal-margin and tree-based methods however very few candidate models were retained during training indicating many had correlated predictions and therefore presented a lack of diversity in the ensemble [198]. This is also reflected in the dominance of SVM amongst the base-learners with both models demonstrating the greatest weightings out of the four models included in the stack (Figure 11). It is perhaps then not surprising that the stacked model demonstrated a performance similar to SVM both in training and final evaluation using the test dataset (Table 7).

Previous studies comparing single ML model performance in stroke prediction on the same dataset have typically found superior predictive ability for RF or SVM compared to MLP and XGBoost [42–44, 48, 50]. This is inconsistent with the present findings; all single models demonstrated overlapping 95% confidence intervals for mean AUPRC estimates from training resamples indicating performance was relatively similar across all four ML methods. This may be due to methodological differences or varying evaluation metrics to quantify and compare performance.

The metrics used for comparison within the established literature often include accuracy, precision, recall and F1 score without a specified decision threshold [42–45] alongside area under the receiver operating curve (AUROC) [46, 48, 50, 53–55]. In highly imbalanced binary classification problems where there is great importance in detecting rare events AUROC can be misleading as it incorporates the larger number of true negatives (i.e. the majority class) in its calculation whilst AUPRC does not [203, 207]. To the authors knowledge, there were no studies found that enlisted the same dataset for stroke prediction and used AUPRC for model evaluation therefore comparison to the literature is limited to threshold-sensitive metrics such as F1, precision and recall. According to the accuracy paradox, using accuracy for the evaluation of highly imbalanced datasets can also be misleading as high values can be obtained just by classifying all examples as the majority class [200].

All four algorithms and the stacked ensemble within this project produced models that demonstrated superior performance to a random classifier based on an AUPRC of 0.049 given by the proportion of patients that experienced stroke in the training dataset [207]. Final model performance for SVM and the stacked model on the test dataset demonstrated similar AUPRC to that observed from training at 0.180 and 0.176 respectively (Table 7) compared to a random classifier baseline of 0.053 on the test dataset. A slight decrease in performance for the stacked model may indicate a degree of overfitting occurred on the training data translating to poorer generalisability to unseen data [216].

As ideal model performance would yield an AUPRC close to 1 and present a PRC close to the upper right corner of the plot [203], the AUPRC and PRCs derived (Figure 13) suggest the final models fail to adequately balance precision and recall across all decision thresholds. This is reflected in the test set class predictions at a threshold that optimised F1 score where both SVM and the stacked model, with F1 scores of 0.297 and 0.304 respectively, led to high numbers of false positives and false negatives (Figure 12). This indicates these models may have severe limitations in their practical application at these thresholds as approximately half of those patients that went on to experience a stroke occurrence failed to be predicted and almost four times as many were incorrectly predicted to have a stroke as were correctly predicted in both models.

The appropriate decision threshold for use in deployment settings is context-dependent and decided upon consideration of false-positive and false-negative associated costs [217]. However, the clinical utility of these models for use in a predictive capacity is likely compromised due to the persistently low precision observed across all values of recall as indicated by the relatively flat PRCs (Figure 13). Therefore whilst increased recall may lead to fewer patients at risk of stroke being missed, the large number of false positives presents the issue of unnecessary follow-up medical examination or treatment for most patients predicted as at risk [218]. This trade-off would likely impact the utility of these models particularly in LMICs where a low precision model may represent an ineffective

allocation of often restricted resources [10].

Based on threshold metrics such as F1, precision and recall, final SVM and stacked model performance was markedly poor compared to that found in the literature using this dataset for stroke prediction; Hassan et al[55] found an F1 score of 0.949 for a stacking ensemble following on from previous studies that demonstrated exceptional F1 performance of 0.974 [54] and even 1.000 [53] using similar stacking methods. Even for single models, Rehman et al[53] achieved an F1 score, precision and recall of 0.630 for an SVM model, far exceeding that found for the final SVM model within the current project. Although decision thresholds were not specified in any of the aforementioned studies, these performance estimates were much higher than the optimal F1 score obtained for both SVM and stacked models on final evaluation (Table 7) indicating superior performance.

The stark difference in predictive performance may be attributed to a fundamental methodological oversampling flaw in many previous studies using this dataset that has been discovered in other clinical domains [219]. Oversampling of the minority stroke class using SMOTE was employed within this project and has been extensively used throughout the stroke prediction literature to overcome limitations of highly imbalanced data and improve predictive performance [35]. For the purposes of this analysis SMOTE was exclusively applied to training data after partitioning into mutually exclusive training and test datasets. Within k-fold cross-validation using the training dataset, SMOTE was also only applied to the data used for training and the sampling routine was never applied to the data used for predictions (i.e. the validation fold) for each resample [91].

This is inconsistent with the implementation of SMOTE within the literature; Rehman et al[53] applied oversampling of the minority stroke class with a non-specified ratio to the entire dataset before partitioning into a training and test dataset. Similarly, Dritsas and Trigka[54] specified that final model assessment was completed using ten-fold cross-validation on the SMOTE-balanced dataset. Ghosh, Dasgupta and Swetapadma[115] initially performed the partitioning of data before applying SMOTE to both the training and test datsets with an oversampling ratio of 1 resulting in

an equal number of patients within each binary stroke class. This methodological step of applying oversampling to validation datasets is described within many other studies using this open-access dataset for stroke prediction [44–46, 48–50, 202, 220, 221].

Hassan et al[55] also applied SMOTE before partitioning the dataset into training and test datasets however additionally explored the modelling and evaluation process in the absence of oversampling. Logistic regression performed best according to accuracy on the imbalanced (non-oversampled) test data alongside an F1 score of 0.230 corresponding to a precision of 0.870 and recall of 0.132. This is also consistent with the findings of Srinivasu et al[222] who found using a SVM model without oversampling the validation data resulted in an F1 score of 0.192 corresponding to a precision of 0.140 and a recall of 0.302. Although a decision threshold is not specified in these studies and therefore may not have been optimised for F1 score, these results are comparable to those observed in this present project.

This highlights a pervasive methodological flaw throughout much of the literature on stroke prediction using this dataset; previous studies using supervised ML approaches in a clinical context have found the incorrect application of oversampling in this way can lead to artificially inflated performance estimates [71, 219]. Whilst training data is used for model tuning and evaluation for final model selection, the test dataset is 'unseen' data created in order to provide an indication of model generalizability to real-world scenarios and evaluate final model performance [70]. The class distribution should therefore closely represent the real-world distribution in which the model will be applied in order to obtain unbiased estimates of future performance [147]. Oversampling the validation datasets therefore contaminates the evaluation process resulting potentially over-optimistic results as the underlying distribution of stroke occurrences is not represented [223].

The disparity in model performance observed compared to that in the literature is also reflected in model interpretation. As the stacked model indicated a degree of overfitting due to a decline in performance on the test dataset, only SVM was considered for model interpretation. Permutation-based

feature importance suggested the age of the patient was the only important feature for predictions made by the final SVM model on the test dataset [127] using mean loss in AUPRC as the importance metric (Figure 14). This is contrary to the permutation feature importance findings of Islam and Ghosh[44] who found all ten features showed importance with average glucose level, BMI and age as the top three using a random forest model. However, permutation feature importance was conducted on oversampled validation data, importance was measured by mean accuracy loss and confidence intervals were not provided for importance estimates making direct comparisons challenging.

Hassan et al[55] compared feature importance for a stacking ensemble between imbalanced and balanced datasets however the loss metric was not specified. Consistent with previous studies [44], age, average glucose level and BMI were the three most important features whilst all other features had minimal contribution to model predictions. In the balanced dataset age had the greatest influence however this was superseded by average blood glucose in the imbalanced dataset. Importance of all features was attenuated in the imbalanced dataset [55] suggesting oversampling in this instance may artificially inflate importance estimates.

Despite this, these findings are consistent with that found using an SVM model and correct implementation of oversampling via SMOTE; Srinivasu et al[222] found average glucose level, age and to a lesser degree BMI most contributed to model predictions using mean AUROC loss whilst all other features were unimportant. Therefore, it is likely the contribution of features to predictions may depend on the metric used to evaluate the loss from permutation, explaining the differences observed.

Nonetheless, the importance of age in the predicted risk of stroke is consistent with the literature and is widely accepted as a primary non-modifiable risk factor [224]. As stroke incidence is known to increase with age [23] it may not be surprising that the predicted stroke risk from the SVM model tended to increase as patient age increased on average (Figure 15). This is in line with the findings

of Kokkotis et al[218] who implemented subsampling exclusively to the training data and used partial dependence plots (PDPs) with Shapley values for an MLP model. They found a similar strong positive monotonic relationship between age and the predicted probability of stroke, reaffirming the importance of this feature in predicting stroke within this dataset. Longitudinal studies suggest stroke incidence doubles with each decade beyond the age of 55 [21, 24]; the predicted risk of stroke in the SVM model appears to increase at its highest rate linearly from approximately 50 years to 80 years of age on average.

A slight increase in predicted risk of stroke was observed with increasing average blood glucose specifically at the low end of the range within this dataset from 50 mg/dL to 125 mg/dL on average, however predicted risk quickly levels off beyond this concentration (Figure 15). This is an interesting finding as a fasting (8-hour) blood glucose greater than 126 mg/dL has been recommended for the diagnosis of diabetes [225–227], which greatly increases the risk of stroke [228].

In contrast, increasing values of BMI appeared to demonstrate no real effect on predicted stroke risk for the SVM model. This is contrary to the literature where PDPs demonstrated a strong negative monotonic relationship between BMI and stroke risk [218]. However, PDPs do not provide a realistic explanation as they are influenced by correlated features [127]; a limitation that ALE overcomes [39]. As the strongest correlation observed during exploratory analysis was between age and BMI (Figure 3) the PDP may misrepresent how BMI influences predicted stroke risk whilst the ALE plot demonstrates little impact. It is also worth noting Kokkotis et al[218] used a dataset of 43,000 patients which is no longer publicly available and from which the present dataset was derived, therefore the differences between ALE and PDP plots may also be attributed to varying distributional properties of the dataset.

Categorical features showed little to no effect on predicted probability of stroke when isolated, reflected in the ALE plots (Figure 16) with the greatest increases for those with heart disease and hypertension compared to no disease on average. The small effect of these features is somewhat

surprising as stroke incidence is known to be 2-4 times greater for those with heart disease [229] and hypertension has been identified as the top risk factor according to GBD estimates [6] and the INTERSTROKE study [7]. However, the ALE plots are specific to the SVM model and dataset therefore the lack of utility in categorical features for stroke prediction is likely due to the models inability to reflect the data-generating process as opposed to those factors intrinsic contribution to real stroke risk [230].

4.1 Strengths and limitations

There are further limitations to this analysis that are specific to the pre-processed open-access dataset used [60]. Very little information is available pertaining to the source of the data, how it was collected and for its intended purpose. The outcome for the purposes of this analysis was a binary stroke variable however it is not clear whether this was the motivation for initial data collection. The study design was not specified however it is common for longitudinal approaches to be used within medical applications for the prediction of an event, such as stroke, at a specific point in time or prediction window [231]. In a systematic review that included 16 studies using the same open-access stroke dataset, the authors concluded information was collected according to a retrospective cohort design however recognised the absence of time-to-event data [36]. Therefore, this project assumed stroke occurrence was a static outcome at an undefined timepoint, and it is recommended future studies include follow-up time information for real-world application [36, 232].

Limitations in generalisability of the present findings may also arise as the sampling method for patients is also not specified; some evidence suggests using electronic health record data in retrospective cohort studies can lead to bias in the selection of patients with generally poorer health [233]. Similarly, as eligibility criteria was not specified for inclusion in the study and subsequent dataset it is unclear the specific population these results are applicable to [234], limiting generalisability.

Precise definitions were not provided for most features within the present dataset. For example age was not defined at a specific time point and was therefore assumed to be collected at the undefined

end time point [235]. Similarly average blood glucose was not defined over a time point or specified as fasting or non-fasting therefore it was assumed to represent average fasting blood glucose [236] over the undefined study period. The model did not include other biochemical features such as serum folate and neutrophils which have also been shown to predict stroke risk [237, 238]. Previous stroke occurrence is an established predictor of secondary stroke [239] therefore this should be considered as an additional feature when predicting stroke risk using electronic health records. The presence of lifestyle factors such as smoking status, occupation, residential area and martial status are assumed to be self-reported which can make them vulnerable to recall and response biases that can render the data unreliable [240]. For example, whilst smoking status is an important predictor of stroke risk [241] the high number of patients with an unknown smoking status in the present dataset may reflect response bias to avoid social judgement in a clinical context [242].

Strengths of this project include the methodologically sound application of oversampling via SMOTE as to avoid overoptimistic performance estimates [219]. Evidence was provided to highlight that this experimental flaw was widespread throughout much of the literature utilising the present dataset for the prediction of stroke occurrence, even in prestigious peer-reviewed journals [55]. To the authors knowledge, this project presents the first correct application of SMOTE to this specific open-access dataset and therefore may represent a more realistic assessment of model performance [71, 223]. Future work using this dataset should include a direct comparison of oversampling applied before and after partitioning into training and test sets which may provide insight into the extent this flawed evaluation methodology impacts model generalisability and real-world application [219].

Contrary to the established literature, the models within the present project were optimised and evaluated primarily using AUPRC which has been recommended as a metric for rare outcomes such as stroke in a recent systematic review on stroke prediction using ML methods [36]. Simulation studies have demonstrated that the commonly used AUROC can lead to overoptimistic performance estimates compared to AUPRC for low prevalence disease prediction using ML methods [243]. Although oversampling was applied in training the models, validation sets remained highly imbalanced

and therefore the use of AURPC more directly captured the ability of models in identifying the stroke group and therefore better reflects the discriminant performance of models in this context [207].

Very few studies in the literature using this dataset for the prediction of stroke investigated model interpretability [44, 55, 222]. To the authors knowledge, this project was the first to include ALE plots for investigating the impact of varying values of features on predicted stroke risk. Building on previous studies that utilised LDPs [222], ALE plots provided a more realistic interpretation of the individual contribution of features independent from correlated features [39, 230].

5 Conclusion

With a globally increasing stroke burden, successfully predicting stroke risk from routinely collected data presents an opportunity for early intervention particularly in resource-limited countries. This project employed MLP, SVM, XGBoost and RF machine learning algorithms and a stacking model to predict stroke occurrence as a binary classification problem. Oversampling techniques were implemented on a highly imbalanced open-access dataset containing demographic, lifestyle and medical features. All models demonstrated similar performance including the stacking ensemble which presented a lack of structural diversity. Final SVM model performance remained superior to a random classifier however illustrated consistently high numbers of false positives, with age as the primary contributor to increasing predicted stroke risk. A pervasive methodological flaw in oversampling implementation was identified within the established literature and discussed as an explanation for discordant findings. ML approaches have promising applications in predicting stroke occurrence using routinely collected data however model development guided by robust methodological practices is of critical importance to ensure generalisability and prevent harm.

References

[1] Alize J Ferrari, Damian Francesco Santomauro, Amirali Aali, Yohannes Habtegiorgis Abate, Cristiana Abbafati, Hedayat Abbastabar, Samar Abd ElHafeez, Michael Abdelmasseh, Sherief Abd-Elsalam, Arash Abdollahi, et al. Global incidence, prevalence, years lived with disability (ylds), disability-adjusted life-years (dalys), and healthy life expectancy (hale) for 371 diseases and injuries in 204 countries and territories and 811 subnational locations, 1990–2021: a systematic analysis for the global burden of disease study 2021. *The Lancet*, 403(10440):2133–2161, 2024.

[2] Mohsen Naghavi, Kanyin Liane Ong, Amirali Aali, Hazim S Ababneh, Yohannes Habtegiorgis Abate, Cristiana Abbafati, Rouzbeh Abbasgholizadeh, Mohammadreza Abbasian, Mohsen Abbasi-Kangevari, Hedayat Abbastabar, et al. Global burden of 288 causes of death and life expectancy decomposition in 204 countries and territories and 811 subnational locations, 1990–2021: a systematic analysis for the global burden of disease study 2021. *The Lancet*, 403(10440):2100–2132, 2024.

[3] Trent M Woodruff, John Thundyil, Sung-Chun Tang, Christopher G Sobey, Stephen M Taylor, and Thiruma V Arumugam. Pathophysiology, treatment, and animal and cellular models of human ischemic stroke. *Molecular neurodegeneration*, 6:1–19, 2011.

[4] Rui Mao, Ningning Zong, Yujie Hu, Ying Chen, and Yun Xu. Neuronal death mechanisms and therapeutic strategy in ischemic stroke. *Neuroscience bulletin*, 38(10):1229–1247, 2022.

[5] Emine Sekerdag, Ihsan Solaroglu, and Yasemin Gursoy-Ozdemir. Cell death mechanisms in stroke and novel molecular and cellular treatment options. *Current neuropharmacology*, 16(9):1396–1415, 2018.

[6] Valery L Feigin, Benjamin A Stark, Catherine Owens Johnson, Gregory A Roth, Catherine Bisignano, Gdiom Gebreheat Abady, Mitra Abbasifard, Mohsen Abbasi-Kangevari, Foad Abd-Allah, Vida Abedi, et al. Global, regional, and national burden of stroke and its risk factors,

1990–2019: a systematic analysis for the global burden of disease study 2019. *The Lancet Neurology*, 20(10):795–820, 2021.

[7] Martin J O'donnell, Denis Xavier, Lisheng Liu, Hongye Zhang, Siu Lim Chin, Purnima Rao-Melacini, Sumathy Rangarajan, Shofiqul Islam, Prem Pais, Matthew J McQueen, et al. Risk factors for ischaemic and intracerebral haemorrhagic stroke in 22 countries (the interstroke study): a case-control study. *The Lancet*, 376(9735):112–123, 2010.

[8] Valery L Feigin, Mayowa O Owolabi, Foad Abd-Allah, Rufus O Akinyemi, Natalia V Bhattacharjee, Michael Brainin, Jackie Cao, Valeria Caso, Bronte Dalton, Alan Davis, et al. Pragmatic solutions to reduce the global burden of stroke: a world stroke organization–lancet neurology commission. *The Lancet Neurology*, 22(12):1160–1206, 2023.

[9] Mayowa O Owolabi, Amanda G Thrift, Ajay Mahal, Marie Ishida, Sheila Martins, Walter D Johnson, Jeyaraj Pandian, Foad Abd-Allah, Joseph Yaria, Hoang T Phan, et al. Primary stroke prevention worldwide: translating evidence into action. *The Lancet Public Health*, 7(1):e74–e85, 2022.

[10] Jakob VE Gerstl, Sarah E Blitz, Qing Rui Qu, Alexander G Yearley, Philipp Lassarén, Rebecca Lindberg, Saksham Gupta, Ari D Kappel, Juan C Vicenty-Padilla, Edoardo Gaude, et al. Global, regional, and national economic consequences of stroke. *Stroke*, 54(9):2380–2389, 2023.

[11] Stefan Strilciuc, Diana Alecsandra Grad, Constantin Radu, Diana Chira, Adina Stan, Marius Ungureanu, Adrian Gheorghe, and Fior-Dafin Muresanu. The economic burden of stroke: a systematic review of cost of illness studies. *Journal of medicine and life*, 14(5):606, 2021.

[12] Paramdeep Kaur, Gagandeep Kwatra, Raminder Kaur, and Jeyaraj D Pandian. Cost of stroke in low and middle income countries: a systematic review. *International Journal of Stroke*, 9(6):678–682, 2014.

[13] Centers for Disease Control, Prevention (CDC, et al. Prevalence of stroke-united states, 2006-2010. *MMWR: Morbidity & Mortality Weekly Report*, 61(20), 2012.

[14] Lijing L Yan, Chaoyun Li, Jie Chen, J Jaime Miranda, Rong Luo, Janet Bettger, Yishan Zhu, Valery Feigin, Martin O'Donnell, Dong Zhao, et al. Prevention, management, and rehabilitation of stroke in low-and middle-income countries. *Eneurologicalsci*, 2:21–30, 2016.

[15] Valery L Feigin, Bo Norrving, Mary G George, Jennifer L Foltz, Gregory A Roth, and George A Mensah. Prevention of stroke: a strategic global imperative. *Nature Reviews Neurology*, 12(9):501–512, 2016.

[16] United Nations. Goal 3: Ensure healthy lives and promote well-being for all at all ages. Available at `https://www.un.org/sustainabledevelopment/health/` (2024/06/22), 2024.

[17] NCD Countdown et al. Ncd countdown 2030: efficient pathways and strategic investments to accelerate progress towards the sustainable development goal target 3.4 in low-income and middle-income countries. *Lancet (London, England)*, 399(10331):1266, 2022.

[18] Yogeshwar V Kalkonde, Suvarna Alladi, Subhash Kaul, and Vladimir Hachinski. Stroke prevention strategies in the developing world. *Stroke*, 49(12):3092–3097, 2018.

[19] Masaraf Hussain. Primordial prevention: The missing link in neurological care. *Journal of Family Medicine and Primary Care*, 10(1):31–34, 2021.

[20] Valery L Feigin, Michael Brainin, Bo Norrving, Philip B Gorelick, Martin Dichgans, Wenzhi Wang, Jeyaraj Durai Pandian, Sheila Cristina Ouriques Martins, Mayowa O Owolabi, David A Wood, et al. What is the best mix of population-wide and high-risk targeted strategies of primary stroke and cardiovascular disease prevention? *Journal of the American Heart Association*, 9(3):e014494, 2020.

[21] Amelia K Boehme, Charles Esenwa, and Mitchell SV Elkind. Stroke risk factors, genetics, and prevention. *Circulation research*, 120(3):472–495, 2017.

[22] Diji Kuriakose and Zhicheng Xiao. Pathophysiology and treatment of stroke: present status and future perspectives. *International journal of molecular sciences*, 21(20):7609, 2020.

[23] Sudha Seshadri, Alexa Beiser, Margaret Kelly-Hayes, Carlos S Kase, Rhoda Au, William B Kannel, and Philip A Wolf. The lifetime risk of stroke: estimates from the framingham study. *Stroke*, 37(2):345–350, 2006.

[24] Margaret Kelly-Hayes. Influence of age and health behaviors on stroke risk: lessons from longitudinal studies. *Journal of the American Geriatrics Society*, 58:S325–S328, 2010.

[25] Moira K Kapral, Jiming Fang, Michael D Hill, Frank Silver, Janice Richards, Cheryl Jaigobin, and Angela M Cheung. Sex differences in stroke care and outcomes: results from the registry of the canadian stroke network. *Stroke*, 36(4):809–814, 2005.

[26] Manav V Vyas, Frank L Silver, Peter C Austin, Amy YX Yu, Priscila Pequeno, Jiming Fang, Andreas Laupacis, and Moira K Kapral. Stroke incidence by sex across the lifespan. *Stroke*, 52(2):447–451, 2021.

[27] Kathryn M Rexrode, Tracy E Madsen, Amy YX Yu, Cheryl Carcel, Judith H Lichtman, and Eliza C Miller. The impact of sex and gender on stroke. *Circulation research*, 130(4):512–528, 2022.

[28] Virginia J Howard, Tracy E Madsen, Dawn O Kleindorfer, Suzanne E Judd, J David Rhodes, Elsayed Z Soliman, Brett M Kissela, Monika M Safford, Claudia S Moy, Leslie A McClure, et al. Sex and race differences in the association of incident ischemic stroke with risk factors. *JAMA neurology*, 76(2):179–186, 2019.

[29] Liyuan Pu, Li Wang, Ruijie Zhang, Tian Zhao, Yannan Jiang, and Liyuan Han. Projected global trends in ischemic stroke incidence, deaths and disability-adjusted life years from 2020 to 2030. *Stroke*, 54(5):1330–1339, 2023.

[30] Philip A Wolf, Ralph B D'Agostino, Albert J Belanger, and William B Kannel. Probability of stroke: a risk profile from the framingham study. *Stroke*, 22(3):312–318, 1991.

[31] Elizabeth Hunter and John D Kelleher. A review of risk concepts and models for predicting the risk of primary stroke. *Frontiers in Neuroinformatics*, 16:883762, 2022.

[32] Yaacoub Chahine, Matthew J Magoon, Bahetihazi Maidu, Juan C Del Álamo, Patrick M Boyle, and Nazem Akoum. Machine learning and the conundrum of stroke risk prediction. *Arrhythmia & Electrophysiology Review*, 12, 2023.

[33] Julia Hippisley-Cox, Carol Coupland, and Peter Brindle. Derivation and validation of qstroke score for predicting risk of ischaemic stroke in primary care and comparison with other risk scores: a prospective open cohort study. *Bmj*, 346, 2013.

[34] Agni Orfanoudaki, Emma Chesley, Christian Cadisch, Barry Stein, Amre Nouh, Mark J Alberts, and Dimitris Bertsimas. Machine learning provides evidence that stroke risk is not linear: The non-linear framingham stroke risk score. *PloS one*, 15(5):e0232414, 2020.

[35] Julia Amann. Machine learning in stroke medicine: Opportunities and challenges for risk prediction and prevention. *Artificial Intelligence in Brain and Mental Health: Philosophical, Ethical & Policy Issues*, pages 57–71, 2022.

[36] Sermkiat Lolak, Chaiyawat Suppasilp, Napaphat Poprom, Tunlanut Sapankaew, Myat Su Yin, Ratchainant Thammasudjarit, Gareth J McKay, John Attia, and Ammarin Thakkinstian. Machine learning prediction of stroke occurrence: A systematic review. *medRxiv*, pages 2024–03, 2024.

[37] Iqbal H Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3):160, 2021.

[38] James A Nichols, Hsien W Herbert Chan, and Matthew AB Baker. Machine learning: applications of artificial intelligence to imaging and diagnosis. *Biophysical reviews*, 11:111–118, 2019.

[39] Daniel W Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(4):1059–1086, 2020.

[40] Junyao Li, Yuxiang Luo, Meina Dong, Yating Liang, Xuejing Zhao, Yafeng Zhang, and Zhaoming Ge. Tree-based risk factor identification and stroke level prediction in stroke cohort study. *BioMed Research International*, 2023(1):7352191, 2023.

[41] Yuexin Qiu, Shiqi Cheng, Yuhang Wu, Wei Yan, Songbo Hu, Yiying Chen, Yan Xu, Xiaona Chen, Junsai Yang, Xiaoyun Chen, et al. Development of rapid and effective risk prediction models for stroke in the chinese population: a cross-sectional study. *BMJ open*, 13(3):e068045, 2023.

[42] Chetan Sharma, Shamneesh Sharma, Mukesh Kumar, and Ankur Sodhi. Early stroke prediction using machine learning. In *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, pages 890–894. IEEE, 2022.

[43] Archana Saini, Kalpna Guleria, and Shagun Sharma. Performance analysis of machine learning approaches for stroke prediction in healthcare. In *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 903–907. IEEE, 2023.

[44] F Islam and M Ghosh. An enhanced stroke prediction scheme using smote and machine learning techniques. In *2021 12th International conference on computing communication and networking technologies (ICCCNT)*, pages 1–6. IEEE, 2021.

[45] Tahia Tazin, Md Nur Alam, Nahian Nakiba Dola, Mohammad Sajibul Bari, Sami Bourouis, and Mohammad Monirujjaman Khan. Stroke disease detection and prediction using robust learning approaches. *Journal of healthcare engineering*, 2021(1):7633381, 2021.

[46] Okpe Anthony Okwori, Moses Adah Agana, Ofem Ajah Ofem, and Obono I Ofem. Stroke prediction with random forest machine learning model. *Asian Research Journal of Current Science*, pages 122–131, 2024.

[47] Nugroho Sinung Adi, Richas Farhany, Rafidah Ghina, and Herlina Napitupulu. Stroke risk prediction model using machine learning. In *2021 International Conference on Artificial Intelligence and Big Data Analytics*, pages 56–60. IEEE, 2021.

[48] Hamza Al-Zubaidi, Mohammed Dweik, and Amjed Al-Mousa. Stroke prediction using machine learning classification methods. In *2022 International Arab Conference on Information Technology (ACIT)*, pages 1–8. IEEE, 2022.

[49] Nitish Biswas, Khandaker Mohammad Mohi Uddin, Sarreha Tasmin Rikta, and Samrat Kumar Dey. A comparative analysis of machine learning classifiers for stroke prediction: A predictive analytics approach. *Healthcare Analytics*, 2:100116, 2022.

[50] Ivan G Ivanov, Yordan Kumchev, and Vincent James Hooper. An optimization precise model of stroke data to improve stroke prediction. *Algorithms*, 16(9):417, 2023.

[51] Minhaz Uddin Emon, Maria Sultana Keya, Tamara Islam Meghla, Md Mahfujur Rahman, M Shamim Al Mamun, and M Shamim Kaiser. Performance analysis of machine learning approaches in stroke prediction. In *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1464–1469. IEEE, 2020.

[52] Saso Džeroski and Bernard Ženko. Is combining classifiers with stacking better than selecting the best one? *Machine learning*, 54:255–273, 2004.

[53] Amjad Rehman, Teg Alam, Muhammad Mujahid, Faten S Alamri, Bayan Al Ghofaily, and Tanzila Saba. Rdet stacking classifier: a novel machine learning based approach for stroke prediction using imbalance data. *PeerJ Computer Science*, 9, 2023.

[54] Elias Dritsas and Maria Trigka. Stroke risk prediction with machine learning techniques. *Sensors*, 22(13):4670, 2022.

[55] Ahmad Hassan, Saima Gulzar Ahmad, Ehsan Ullah Munir, Imtiaz Ali Khan, and Naeem Ramzan. Predictive modelling and identification of key risk factors for stroke using machine learning. *Scientific Reports*, 14(1):11498, 2024.

[56] Soumyabrata Dev, Hewei Wang, Chidozie Shamrock Nwosu, Nishtha Jain, Bharadwaj Veeravalli, and Deepu John. A predictive analytics approach for stroke prediction using machine learning and neural networks. *Healthcare Analytics*, 2:100032, 2022.

[57] Tianyu Liu, Wenhui Fan, and Cheng Wu. A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical dataset. *Artificial intelligence in medicine*, 101:101723, 2019.

[58] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2024.

[59] Max Kuhn and Hadley Wickham. *Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles.*, 2020.

[60] Kaggle. Brain stroke prediction dataset. Available at `https://www.kaggle.com/datasets/zzettrkalpakbal/full-filled-brain-stroke-dataset` (2024/06/22), 2024.

[61] Muhammad Salman Pathan, Zhang Jianbiao, Deepu John, Avishek Nag, and Soumyabrata Dev. Identifying stroke indicators using rough sets. *IEEE Access*, 8:210318–210327, 2020.

[62] John Wilder Tukey et al. *Exploratory data analysis*, volume 2. Springer, 1977.

[63] Dalwinder Singh and Birmohan Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97:105524, 2020.

[64] Xinjian Guo, Yilong Yin, Cailing Dong, Gongping Yang, and Guangtong Zhou. On the class imbalance problem. In *2008 Fourth international conference on natural computation*, volume 4, pages 192–201. IEEE, 2008.

[65] Camelia Lemnaru and Rodica Potolea. Imbalanced classification problems: systematic study, issues and best practices. In *Enterprise Information Systems: 13th International Conference, ICEIS 2011, Beijing, China, June 8-11, 2011, Revised Selected Papers 13*, pages 35–50. Springer, 2012.

[66] Lianxi Wang, Shengyi Jiang, and Siyu Jiang. A feature selection method via analysis of relevance, redundancy, and interaction. *Expert Systems with Applications*, 183:115365, 2021.

[67] Jireh Yi-Le Chan, Steven Mun Hong Leow, Khean Thye Bea, Wai Khuen Cheng, Seuk Wai Phoong, Zeng-Wei Hong, and Yen-Lin Chen. Mitigating the multicollinearity problem and its machine learning approach: a review. *Mathematics*, 10(8):1283, 2022.

[68] Hannah Frick, Fanny Chow, Max Kuhn, Michael Mahoney, Julia Silge, and Hadley Wickham. *rsample: General Resampling Infrastructure*, 2024. R package version 1.2.1.

[69] Cristiano Cervellera and Danilo Macciò. Distribution-preserving stratified sampling for learning problems. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):2886–2895, 2017.

[70] Yun Xu and Royston Goodacre. On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of analysis and testing*, 2(3):249–262, 2018.

[71] Farhad Maleki, Katie Ovens, Rajiv Gupta, Caroline Reinhold, Alan Spatz, and Reza Forghani. Generalizability of machine learning models: quantitative evaluation of three methodological pitfalls. *Radiology: Artificial Intelligence*, 5(1):e220028, 2022.

[72] Max Kuhn, Hadley Wickham, and Emil Hvitfeldt. *recipes: Preprocessing and Feature Engineering Steps for Modeling*, 2024. R package version 1.0.10.

[73] Aydin Demircioğlu. Applying oversampling before cross-validation will lead to high bias in radiomics. *Scientific Reports*, 14(1):11563, 2024.

[74] Halis Altun, A Bilgil, and BC Fidan. Treatment of skewed multi-dimensional training data to facilitate the task of engineering neural models. *Expert Systems with Applications*, 33(4):978–983, 2007.

[75] Yan-Yan Song and LU Ying. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2):130, 2015.

[76] Shovan Chowdhury, Yuxiao Lin, Boryann Liaw, and Leslie Kerby. Evaluation of tree based regression over multiple linear regression for non-normally distributed data in battery per-

formance. In *2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pages 17–25. IEEE, 2022.

[77] George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 26(2):211–243, 1964.

[78] Kedar Potdar, Taher S Pardawala, and Chinmay D Pai. A comparative study of categorical variable encoding techniques for neural network classifiers. *International journal of computer applications*, 175(4):7–9, 2017.

[79] Florian Pargent, Florian Pfisterer, Janek Thomas, and Bernd Bischl. Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Computational Statistics*, 37(5):2671–2692, 2022.

[80] Max Kuhn and Kjell Johnson. *Feature engineering and selection: A practical approach for predictive models*. Chapman and Hall/CRC, 2019.

[81] Neil H Timm and James E Carlson. Analysis of variance through full rank models. *Multivariate behavioral research monographs*, 1975.

[82] Timothy C Au. Random forests, decision trees, and categorical predictors: the" absent levels" problem. *Journal of Machine Learning Research*, 19(45):1–30, 2018.

[83] Brian Lucena. Exploiting categorical structure using tree-based methods. In *International Conference on Artificial Intelligence and Statistics*, pages 2949–2958. PMLR, 2020.

[84] Kelsy Cabello-Solorzano, Isabela Ortigosa de Araujo, Marco Peña, Luís Correia, and Antonio J. Tallón-Ballesteros. The impact of data normalization on the accuracy of machine learning algorithms: a comparative analysis. In *International Conference on Soft Computing Models in Industrial and Environmental Applications*, pages 344–353. Springer, 2023.

[85] Dilber Uzun Ozsahin, Mubarak Taiwo Mustapha, Auwalu Saleh Mubarak, Zubaida Said Ameen, and Berna Uzun. Impact of feature scaling on machine learning models for the di-

agnosis of diabetes. In *2022 International Conference on Artificial Intelligence in Everything (AIE)*, pages 87–94. IEEE, 2022.

[86] Amit Pandey and Achin Jain. Comparative analysis of knn algorithm using various normalization techniques. *International Journal of Computer Network and Information Security*, 10(11):36, 2017.

[87] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[88] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[89] Roman Timofeev. Classification and regression trees (cart) theory and applications. *Humboldt University, Berlin*, 54:48, 2004.

[90] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[91] Emil Hvitfeldt. *themis: Extra Recipes Steps for Dealing with Unbalanced Data*, 2023. R package version 1.0.2.

[92] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[93] Philipp Thölke, Yorguin-Jose Mantilla-Ramos, Hamza Abdelhedi, Charlotte Maschke, Arthur Dehgan, Yann Harel, Anirudha Kemtur, Loubna Mekki Berrada, Myriam Sahraoui, Tammy Young, et al. Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data. *NeuroImage*, 277:120253, 2023.

[94] David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996.

[95] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168, 2006.

[96] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[97] Chao Ma, Stephan Wojtowytsch, Lei Wu, et al. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. *arXiv preprint arXiv:2009.10713*, 2020.

[98] Max Kuhn and Daniel Falbel. *brulee: High-Level Modeling Functions with 'torch'*, 2024. R package version 0.3.0.

[99] PG Benardos and G-C Vosniakos. Optimizing feedforward artificial neural network architecture. *Engineering applications of artificial intelligence*, 20(3):365–382, 2007.

[100] Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588, 2009.

[101] George Bebis and Michael Georgiopoulos. Feed-forward neural networks. *Ieee Potentials*, 13(4):27–31, 1994.

[102] Andrej Krenker, Janez Bešter, and Andrej Kos. Introduction to the artificial neural networks. *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech*, pages 1–18, 2011.

[103] AD Dongare, RR Kharde, Amit D Kachare, et al. Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1):189–194, 2012.

[104] Rudolf Kruse, Sanaz Mostaghim, Christian Borgelt, Christian Braune, and Matthias Steinbrecher. Multi-layer perceptrons. In *Computational intelligence: a methodological introduction*, pages 53–124. Springer, 2022.

[105] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316, 2017.

[106] Brad Boehmke and Brandon M Greenwell. *Hands-on machine learning with R*. Chapman and Hall/CRC, 2019.

[107] Maikel Kerkhof, Lichao Wu, Guilherme Perin, and Stjepan Picek. No (good) loss no gain: systematic evaluation of loss functions in deep learning-based side-channel analysis. *Journal of Cryptographic Engineering*, 13(3):311–324, 2023.

[108] Laura Burke and James P Ignizio. A practical overview of neural networks. *Journal of Intelligent Manufacturing*, 8:157–165, 1997.

[109] Sagar V Kamarthi and Stefan Pittner. Accelerating neural network training using weight extrapolations. *Neural networks*, 12(9):1285–1299, 1999.

[110] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[111] Alexandros Karatzoglou, Alex Smola, Kurt Hornik, and Achim Zeileis. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20, 2004.

[112] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003.

[113] Johan AK Suykens. Nonlinear modelling and support vector machines. In *IMTC 2001. proceedings of the 18th IEEE instrumentation and measurement technology conference. Rediscovering measurement in the age of informatics (Cat. No. 01CH 37188)*, volume 1, pages 287–294. IEEE, 2001.

[114] Mariette Awad, Rahul Khanna, Mariette Awad, and Rahul Khanna. Support vector machines for classification. *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*, pages 39–66, 2015.

[115] Sourish Ghosh, Anasuya Dasgupta, and Aleena Swetapadma. A study on support vector machine based linear and non-linear pattern classification. In *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, pages 24–28. IEEE, 2019.

[116] Derek A Pisner and David M Schnyer. Support vector machine. In *Machine learning*, pages 101–121. Elsevier, 2020.

[117] Shan Suthaharan and Shan Suthaharan. Support vector machine. *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pages 207–235, 2016.

[118] MN Murty, Rashmi Raghava, MN Murty, and Rashmi Raghava. Kernel-based svm. *Support vector machines and perceptrons: Learning, optimization, classification, and application to social networks*, pages 57–67, 2016.

[119] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[120] Vojislav Kecman. Support vector machines–an introduction. In *Support vector machines: theory and applications*, pages 1–47. Springer, 2005.

[121] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[122] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[123] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan Xie, Min Lin, Yifeng Geng, Yutian Li, and Jiaming Yuan. *xgboost: Extreme Gradient Boosting*, 2024. R package version 1.7.7.1.

[124] Deandra Aulia Rusdah and Hendri Murfi. Xgboost in handling missing values for life insurance risk prediction. *SN Applied Sciences*, 2(8):1336, 2020.

[125] Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(6):275–285, 2004.

[126] Wei-Yin Loh. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(1):14–23, 2011.

[127] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[128] Vinícius G Costa and Carlos E Pedreira. Recent advances in decision trees: An updated survey. *Artificial Intelligence Review*, 56(5):4765–4800, 2023.

[129] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41:77–93, 2004.

[130] Xue Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing, 2019.

[131] Lior Rokach and Oded Maimon. Decision trees. *Data mining and knowledge discovery handbook*, pages 165–192, 2005.

[132] Floriana Esposito, Donato Malerba, Giovanni Semeraro, and J Kay. A comparative analysis of methods for pruning decision trees. *IEEE transactions on pattern analysis and machine intelligence*, 19(5):476–491, 1997.

[133] Antonella Plaia, Simona Buscemi, Johannes Fürnkranz, and Eneldo Loza Mencía. Comparing boosting and bagging for decision trees of rankings. *Journal of Classification*, 39(1):78–99, 2022.

[134] Jerry Ye, Jyh-Herng Chow, Jiang Chen, and Zhaohui Zheng. Stochastic gradient boosted distributed decision trees. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 2061–2064, 2009.

[135] Thomas G Dietterich and Eun Bae Kong. Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. 1995.

[136] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[137] Jane Elith, John R Leathwick, and Trevor Hastie. A working guide to boosted regression trees. *Journal of animal ecology*, 77(4):802–813, 2008.

[138] Leo Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

[139] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[140] Marvin N. Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1):1–17, 2017.

[141] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[142] Tae-Hwy Lee, Aman Ullah, and Ran Wang. Bootstrap aggregating and random forest. *Macroeconomic forecasting in the era of big data: Theory and practice*, pages 389–429, 2020.

[143] Philipp Probst, Marvin N Wright, and Anne-Laure Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, 9(3):e1301, 2019.

[144] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, 20(53):1–32, 2019.

[145] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[146] Tao Shi and Steve Horvath. Unsupervised learning with random forest predictors. *Journal of Computational and Graphical Statistics*, 15(1):118–138, 2006.

[147] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013.

[148] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 2020.

[149] Drew Wilimitis and Colin G Walsh. Practical considerations and applied examples of cross-validation for model development and evaluation in health care: tutorial. *JMIR AI*, 2:e49023, 2023.

[150] Eva Bartz, Thomas Bartz-Beielstein, Martin Zaefferer, and Olaf Mersmann. *Hyperparameter tuning for machine and deep learning with R: A practical guide*. Springer Nature, 2023.

[151] Matthias Feurer and Frank Hutter. Hyperparameter optimization. *Automated machine learning: Methods, systems, challenges*, pages 3–33, 2019.

[152] Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

[153] Max Kuhn and Hannah Frick. *dials: Tools for Creating Tuning Parameter Values*, 2024. R package version 1.2.1.

[154] Oden Maron and Andrew W Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11:193–225, 1997.

[155] Max Kuhn. Futility analysis in the cross-validation of machine learning models. *arXiv preprint arXiv:1405.6974*, 2014.

[156] Max Kuhn. *finetune: Additional Functions for Model Tuning*, 2024. R package version 1.2.0.

[157] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[158] L Fletcher, V Katkovnik, FE Steffens, and AP Engelbrecht. Optimizing the number of hidden nodes of a feedforward artificial neural network. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*, volume 2, pages 1608–1612. IEEE, 1998.

[159] Bradley Efron and Trevor Hastie. *Computer age statistical inference, student edition: algorithms, evidence, and data science*, volume 6. Cambridge University Press, 2021.

[160] Steven Walczak and Narciso Cerpa. Heuristic principles for the design of artificial neural networks. *Information and software technology*, 41(2):107–117, 1999.

[161] Prasenjit Dey, Kaustuv Nag, Tandra Pal, and Nikhil R Pal. Regularizing multilayer perceptron for robustness. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(8):1255–1266, 2017.

[162] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[163] Stefan Wager, Sida Wang, and Percy S Liang. Dropout training as adaptive regularization. *Advances in neural information processing systems*, 26, 2013.

[164] George D Magoulas, Michael N Vrahatis, and George S Androulakis. Effective backpropagation training with variable stepsize. *Neural networks*, 10(1):69–82, 1997.

[165] Yinyin Liu, Janusz A Starzyk, and Zhen Zhu. Optimized approximation algorithm in neural networks without overfitting. *IEEE transactions on neural networks*, 19(6):983–995, 2008.

[166] Sridhar Narayan. The generalized sigmoid activation function: Competitive supervised learning. *Information sciences*, 99(1-2):69–82, 1997.

[167] Emad AM Andrews Shenouda. A quantitative comparison of different mlp activation functions in classification. In *International Symposium on Neural Networks*, pages 849–857. Springer, 2006.

[168] Robert A Jacobs. Increased rates of convergence through learning rate adaptation. *Neural networks*, 1(4):295–307, 1988.

[169] D Randall Wilson and Tony R Martinez. The need for small learning rates on large problems. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, volume 1, pages 115–119. IEEE, 2001.

[170] George D. Magoulas, Michael N. Vrahatis, and George S Androulakis. Improving the convergence of the backpropagation algorithm using learning rate adaptation methods. *Neural Computation*, 11(7):1769–1796, 1999.

[171] Nello Cristianini and Bernhard Scholkopf. Support vector machines and kernel methods: the new generation of learning machines. *Ai Magazine*, 23(3):31–31, 2002.

[172] Arti Patle and Deepak Singh Chouhan. Svm kernel functions for classification. In *2013 International conference on advances in technology and engineering (ICATE)*, pages 1–9. IEEE, 2013.

[173] Abdul Azis Abdillah et al. Diagnosis of diabetes using support vector machines with radial basis function kernels. *International Journal of Technology*, 7(5), 2016.

[174] V Anuja Kumari, R Chitra, et al. Classification of diabetes disease using support vector machine. *International Journal of Engineering Research and Applications*, 3(2):1797–1801, 2013.

[175] Yan Zhang, Fugui Liu, Zhigang Zhao, Dandan Li, Xiaoyan Zhou, and Jingyuan Wang. Studies on application of support vector machine in diagnose of coronary heart disease. In *2012 Sixth*

International Conference on Electromagnetic Field Problems and Applications, pages 1–4. IEEE, 2012.

[176] Yun Liu, Jie Lian, Michael R Bartolacci, and Qing-An Zeng. Density-based penalty parameter optimization on c-svm. *The Scientific World Journal*, 2014(1):851814, 2014.

[177] Feiping Nie, Wei Zhu, and Xuelong Li. Decision tree svm: An extension of linear svm for non-linear classification. *Neurocomputing*, 401:153–159, 2020.

[178] Henry Han and Xiaoqian Jiang. Overcome support vector machine diagnosis overfitting. *Cancer informatics*, 13:CIN–S13875, 2014.

[179] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54:1937–1967, 2021.

[180] Maria D Guillen, Juan Aparicio, and Miriam Esteve. Gradient tree boosting and the estimation of production frontiers. *Expert Systems with Applications*, 214:119134, 2023.

[181] Omer Sagi and Lior Rokach. Approximating xgboost with an interpretable decision tree. *Information sciences*, 572:522–542, 2021.

[182] Atefeh Mansoori, Masoomeh Zeinalnezhad, and Leila Nazarimanesh. Optimization of tree-based machine learning models to predict the length of hospital stay using genetic algorithm. *Journal of healthcare engineering*, 2023(1):9673395, 2023.

[183] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[184] Bulat Ibragimov and Gleb Gusev. Minimal variance sampling in stochastic gradient boosting. *Advances in Neural Information Processing Systems*, 32, 2019.

[185] Tong Zhang and Bin Yu. Boosting with early stopping: Convergence and consistency. 2005.

[186] Zhen Chen and Wei Fan. A freeway travel time prediction method based on an xgboost model. *Sustainability*, 13(15):8577, 2021.

[187] Barbara FF Huang and Paul C Boutros. The parameter sensitivity of random forests. *BMC bioinformatics*, 17:1–13, 2016.

[188] Eric W Fox, Ryan A Hill, Scott G Leibowitz, Anthony R Olsen, Darren J Thornbrugh, and Marc H Weber. Assessing the accuracy and stability of variable selection methods for random forest modeling in ecology. *Environmental monitoring and assessment*, 189:1–20, 2017.

[189] Simon Bernard, Laurent Heutte, and Sébastien Adam. Influence of hyperparameters on random forest accuracy. In *Multiple Classifier Systems: 8th International Workshop, MCS 2009, Reykjavik, Iceland, June 10-12, 2009. Proceedings 8*, pages 171–180. Springer, 2009.

[190] Mark R Segal. Machine learning benchmarks and random forest regression. 2004.

[191] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198, 1999.

[192] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[193] Mark J Van der Laan, Eric C Polley, and Alan E Hubbard. Super learner. *Statistical applications in genetics and molecular biology*, 6(1), 2007.

[194] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[195] Leo Breiman. Stacked regressions. *Machine learning*, 24:49–64, 1996.

[196] Simon Couch and Max Kuhn. *stacks: Tidy Model Stacking*, 2024. R package version 1.0.4.

[197] Simon P Couch and Max Kuhn. stacks: Stacked ensemble modeling with tidy data principles. *Journal of Open Source Software*, 7(75):4471, 2022.

[198] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[199] Alaa Tharwat. Classification assessment methods. *Applied computing and informatics*, 17(1):168–192, 2021.

[200] Francisco J Valverde-Albacete and Carmen Peláez-Moreno. 100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox. *PloS one*, 9(1):e84217, 2014.

[201] Amalia Luque, Alejandro Carrasco, Alejandro Martín, and Ana de Las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231, 2019.

[202] Meshrif Alruily, Sameh Abd El-Ghany, Ayman Mohamed Mostafa, Mohamed Ezz, and AA Abd El-Aziz. A-tuning ensemble machine learning technique for cerebral stroke prediction. *Applied Sciences*, 13(8):5047, 2023.

[203] Helen R Sofaer, Jennifer A Hoeting, and Catherine S Jarnevich. The area under the precision-recall curve as a performance metric for rare binary events. *Methods in Ecology and Evolution*, 10(4):565–577, 2019.

[204] William Cullerne Bown. Sensitivity and specificity versus precision and recall, and related dilemmas. *Journal of Classification*, pages 1–25, 2024.

[205] Zachary C Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize f1 measure. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14*, pages 225–239. Springer, 2014.

[206] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from imbalanced data sets*, volume 10. Springer, 2018.

[207] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

[208] Max Kuhn, Davis Vaughan, and Edgar Ruiz. *probably: Tools for Post-Processing Predicted Values*, 2024. R package version 1.0.3.

[209] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning–a brief history, state-of-the-art and challenges. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 417–431. Springer, 2020.

[210] Vikas Hassija, Vinay Chamola, Atmesh Mahapatra, Abhinandan Singal, Divyansh Goel, Kaizhu Huang, Simone Scardapane, Indro Spinelli, Mufti Mahmud, and Amir Hussain. Interpreting black-box models: a review on explainable artificial intelligence. *Cognitive Computation*, 16(1):45–74, 2024.

[211] Mark Ibrahim, Melissa Louie, Ceena Modarres, and John Paisley. Global explanations of neural networks: Mapping the landscape of predictions. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 279–287, 2019.

[212] Mirka Saarela and Susanne Jauhiainen. Comparison of feature importance measures as explanations for classification models. *SN Applied Sciences*, 3(2):272, 2021.

[213] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.

[214] Szymon Maksymiuk, Alicja Gosiewska, and Przemyslaw Biecek. Landscape of r packages for explainable artificial intelligence. *arXiv*, 2020. Pages 6, 7, 11, 15.

[215] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

[216] Osval Antonio Montesinos López, Abelardo Montesinos López, and Jose Crossa. Overfitting, model tuning, and evaluation of prediction performance. In *Multivariate statistical machine learning methods for genomic prediction*, pages 109–139. Springer, 2022.

[217] Jonathan Birch, Kathleen A Creel, Abhinav K Jha, and Anya Plutynski. Clinical decisions using ai must consider patient values. *Nature medicine*, 28(2):229–232, 2022.

[218] Christos Kokkotis, Georgios Giarmatzis, Erasmia Giannakou, Serafeim Moustakidis, Themistoklis Tsatalas, Dimitrios Tsiptsios, Konstantinos Vadikolias, and Nikolaos Aggelousis. An explainable machine learning pipeline for stroke prediction on imbalanced data. *Diagnostics*, 12(10):2392, 2022.

[219] Gilles Vandewiele, Isabelle Dehaene, György Kovács, Lucas Sterckx, Olivier Janssens, Femke Ongenae, Femke De Backere, Filip De Turck, Kristien Roelens, Johan Decruyenaere, et al. Overly optimistic prediction results on imbalanced data: a case study of flaws and benefits when applying over-sampling. *Artificial Intelligence in Medicine*, 111:101987, 2021.

[220] Gangavarapu Sailasya and Gorli L Aruna Kumari. Analyzing the performance of stroke prediction using ml classification algorithms. *International Journal of Advanced Computer Science and Applications*, 12(6), 2021.

[221] Saad Sahriar, Sanjida Akther, Jannatul Mauya, Ruhul Amin, Md Shahajada Mia, Sabba Ruhi, and Md Shamim Reza. Unlocking stroke prediction: Harnessing projection-based statistical feature extraction with ml algorithms. *Heliyon*, 10(5), 2024.

[222] Parvathaneni Naga Srinivasu, Uddagiri Sirisha, Kotte Sandeep, S Phani Praveen, Lakshmana Phaneendra Maguluri, and Thulasi Bikku. An interpretable approach with explainable ai for heart stroke prediction. *Diagnostics*, 14(2):128, 2024.

[223] Miriam Seoane Santos, Jastin Pompeu Soares, Pedro Henrigues Abreu, Helder Araujo, and Joao Santos. Cross-validation for imbalanced datasets: avoiding overoptimistic and overfitting approaches [research frontier]. *ieee ComputatioNal iNtelligeNCe magaziNe*, 13(4):59–76, 2018.

[224] MS Jahirul Hoque Choudhury, Md Tauhidul Islam Chowdhury, Abu Nayeem, and Waseka Akter Jahan. Modifiable and non-modifiable risk factors of stroke: A review update. *Journal of National Institute of Neurosciences Bangladesh*, 1(1):22–26, 2015.

[225] World Health Organization et al. Definition and diagnosis of diabetes mellitus and intermediate hyperglycaemia: report of a who/idf consultation. 2006.

[226] American Diabetes Association. Diagnosis and classification of diabetes mellitus. *Diabetes care*, 37(Supplement_1):S81–S90, 2014.

[227] Khanh NC Duong, Chia Jie Tan, Sasivimol Rattanasiri, Ammarin Thakkinstian, Thunyarat Anothaisintawee, and Nathorn Chaiyakunapruk. Comparison of diagnostic accuracy for diabetes diagnosis: A systematic review and network meta-analysis. *Frontiers in Medicine*, 10:1016381, 2023.

[228] Ofri Mosenzon, Alice YY Cheng, Alejandro A Rabinstein, and Simona Sacco. Diabetes and stroke: what are the connections? *Journal of Stroke*, 25(1):26–38, 2023.

[229] Keira Robinson, Judith M Katzenellenbogen, Timothy J Kleinig, Joosup Kim, Charley A Budgeon, Amanda G Thrift, and Lee Nedkoff. Large burden of stroke incidence in people with cardiac disease: a linked data cohort study. *Clinical Epidemiology*, pages 203–211, 2023.

[230] Christoph Molnar, Gunnar König, Julia Herbinger, Timo Freiesleben, Susanne Dandl, Christian A Scholbeck, Giuseppe Casalicchio, Moritz Grosse-Wentrup, and Bernd Bischl. General pitfalls of model-agnostic interpretation methods for machine learning models. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pages 39–68. Springer, 2020.

[231] Anna Cascarano, Jordi Mur-Petit, Jeronimo Hernandez-Gonzalez, Marina Camacho, Nina de Toro Eadie, Polyxeni Gkontra, Marc Chadeau-Hyam, Jordi Vitria, and Karim Lekadir. Machine and deep learning for longitudinal biomedical data: a review of methods and applications. *Artificial Intelligence Review*, 56(Suppl 2):1711–1771, 2023.

[232] Farah Shamout, Tingting Zhu, and David A Clifton. Machine learning for clinical outcome prediction. *IEEE reviews in Biomedical Engineering*, 14:116–126, 2020.

[233] Alexander Rusanov, Nicole G Weiskopf, Shuang Wang, and Chunhua Weng. Hidden in plain sight: bias towards sick patients when sampling patients with sufficient electronic health record data for research. *BMC medical informatics and decision making*, 14:1–9, 2014.

[234] Muriel Ramirez-Santana. Limitations and biases in cohort studies, 2018.

[235] Adel Sadeq, Mohamed A Baraka, Amar Hamrouni, and Asim Ahmed Elnour. Retrospective cohort study on risk factors for developing ischemic stroke. *Pharmacy Practice*, 20(3):1–7, 2022.

[236] Joohon Sung, Yun-Mi Song, Shah Ebrahim, and Debbie A Lawlor. Fasting blood glucose and the risk of stroke and myocardial infarction. *Circulation*, 119(6):812–819, 2009.

[237] Eman M Alanazi, Aalaa Abdou, and Jake Luo. Predicting risk of stroke from lab tests using machine learning algorithms: Development and evaluation of prediction models. *JMIR formative research*, 5(12):e23440, 2021.

[238] Si-Ying Song, Xiao-Xi Zhao, Gary Rajah, Chang Hua, Rui-jun Kang, Yi-peng Han, Yu-chuan Ding, and Ran Meng. Clinical significance of baseline neutrophil-to-lymphocyte ratio in patients with ischemic stroke or hemorrhagic stroke: an updated meta-analysis. *Frontiers in neurology*, 10:1032, 2019.

[239] Giorgio Colangelo, Marc Ribo, Estefanía Montiel, Didier Dominguez, Marta Olivé-Gadea, Marian Muchada, Álvaro Garcia-Tornel, Manuel Requena, Jorge Pagola, Jesús Juega, et al. Prerisk: A personalized, artificial intelligence–based and statistically–based stroke recurrence predictor for recurrent stroke. *Stroke*, 55(5):1200–1209, 2024.

[240] Alaa Althubaiti. Information bias in health research: definition, pitfalls, and adjustment methods. *Journal of multidisciplinary healthcare*, pages 211–217, 2016.

[241] Biqi Pan, Xiao Jin, Liu Jun, Shaohong Qiu, Qiuping Zheng, and Mingwo Pan. The relationship between smoking and stroke: a meta-analysis. *Medicine*, 98(12):e14872, 2019.

[242] Julianne Williams, Ivo Rakovac, Enrique Loyola, Lela Sturua, Nino Maglakelidze, Amiran Gamkrelidze, Kristina Mauer-Stender, Bente Mikkelsen, and João Breda. A comparison of self-reported to cotinine-detected smoking status among adults in georgia. *European journal of public health*, 30(5):1007–1012, 2020.

[243] Brice Ozenne, Fabien Subtil, and Delphine Maucort-Boulch. The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases. *Journal of clinical epidemiology*, 68(8):855–859, 2015.