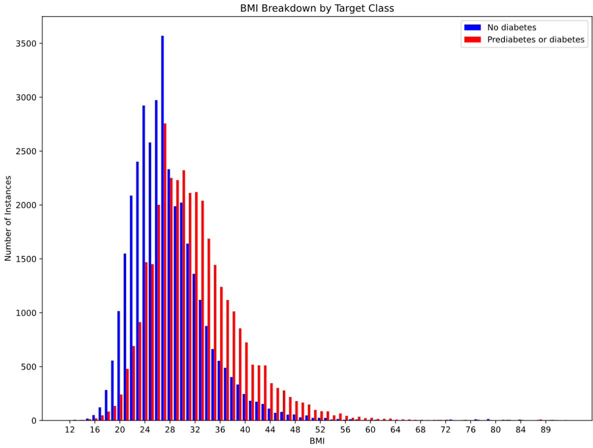
Introduction

Diabetes is a disease that has seen its effects escalate - particularly in the US - as the number of new cases continues to rise, along with extreme increases in the price of insulin. Analyses of data from 1980 to 2012, indicate a doubling of the prevalence of diabetes in the US [REF1] while, in 2019, 11.3% of the population had diabetes - with approximately 22.8% of those cases being undiagnosed [REF2]. Those diagnosed with diabetes, on average, have medical expenditures roughly 2.3 times higher than they would otherwise [REF3], incurring huge personal costs for medical care. On a nationwide scale, the rises in the cost of insulin, as well as the prevalence of diabetes, have resulted in a total estimated cost of $237 billion, in direct medical costs, and $90 billion in reduced productivity [REF3]. These statistics highlight the importance of prediabetes and early-stage diagnosis, which may allow subjects to alter their behaviours - such as exercise and eating habits - in such a way that the effects of the disease are reduced.

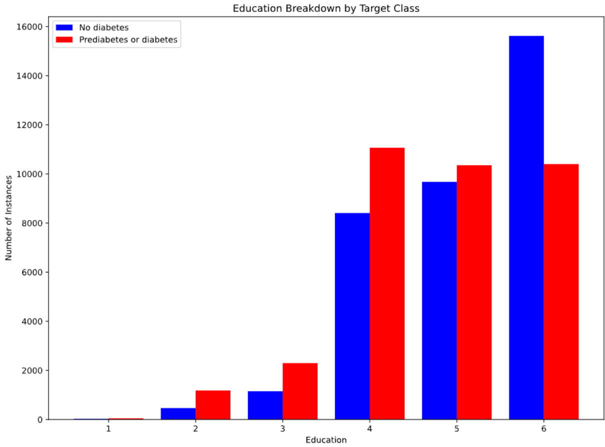
This paper will compare and evaluate two neural network models, each tasked with diagnosing diabetes or prediabetes in patients based on a number of health-related attributes collected from a survey. The models utilised in this report are a Multilayer Perceptron (MLP) and a Support Vector Classifier, which are models that are commonly used in the field of neural networks and can be applied to a wide range of problems.

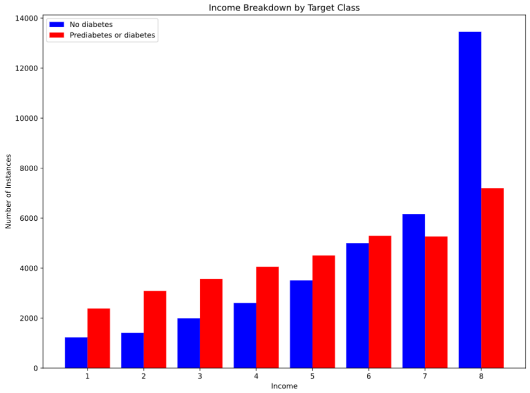
Description of the Dataset



The dataset that was analysed was found on Kaggle {REF4], having been compiled from survey data collected by the Centers for Disease Control and Prevention (CDC). The dataset contains 70692 instances, each with 21, largely categorical, attributes - consisting of 14 binary attributes, 6 discrete attributes. These feature values are numerically encoded in the dataset, so the corresponding values can be found in the appendix. The binary target variable - Class 0 = No Diabetes, Class 1 = Prediabetes or Diabetes - is evenly distributed and combines prediabetes and diabetes into one class due to the conditions' similarities.

Exploratory analysis of the dataset involved plotting bar graphs for each attribute, with the bars split by target class. This enables a search for distinctions in attribute values based on target class. The graphs included in this report are for some of the attributes that demonstrate higher levels of distinction between values for instances in Class 0 and instances in Class 1. BMI (top right) shows clear margins between occurrences of each class for a given value, as one might expect due to the relationship between diabetes and overeating. Education level (bottom right) also displays distinctions between both classes. Instances with low to moderate levels of education contain higher numbers of Class 1 than Class 0, whereas the top level of reported education - corresponding to college graduates - shows a very large margin between these classes, with a higher number of instances of Class 0.





The income attribute (left) shows that, for those that earn less than $50000 a year, the number of instances that belong to Class 0 is lower than the number that belong to Class 1. For those who earn above $50000, the target class balance is reversed, with an extremely severe margin for those who earn $75000 or more. This is likely due to the types of food-making processes that are employed in making cheap food, as well as the accessibility of free time and exercise for individuals.

Other attributes showing clear distinctions are the binary attributes representing difficulty walking, the presence of heart disease or attack, the presence of high blood pressure and the presence of high cholesterol. The last three of these attributes are symptoms of diabetes, while the first is intuitively linked due to the health complications presented by the disease. The graphs for these four binary attributes can be seen in the appendix.

Model Summaries

MLPs are fully-connected feedforward neural networks. The network consists of layers of nodes that feed linear combinations of their inputs through connections to the next layer. Connections in the model are linear, each having a unique weight as well as a bias that's shared across all connections between two consecutive layers. The models begin with an input layer, where the number of neurons is equal to the number of input values. Values from the input layer are fed forward through one or more hidden layers that each contain a specified number of neurons.

The activation of neurons in the layers after the input layer are calculated as the bias plus the weighted sum of the activations of connected neurons in the previous layer, where each weight corresponds to a connection, with an activation function applied. The neurons in the hidden layers employ non-linear activation functions, such as ReLU or sigmoid, which limit the amplitude of the output of a neuron [REF5] as well as adding non-linearity to the model. This allows MLPs to learn complex mappings between inputs and outputs [REF6] despite their relative simplicity, a key advantage of MLPs. Once the values have been fed through the hidden layers, they will reach an output layer with a size and activation function that are both dictated by the type of problem the model is designed to solve.

MLPs are initialised with random weights and biases that are optimised iteratively. Training data is forward passed through the model, with outputs then being compared with the true target values using a loss function, such as mean squared error for regression or cross entropy loss for classification. The loss function is minimised via gradient descent, which is performed using an approximation of the gradient of the loss function with respect to the weights and biases. The gradient obtained indicates the direction of steepest ascent for the error with respect to the weights. Thus, the weights and biases are adjusted by the negative of the gradient, multiplied by the learning rate, through backpropagation.

Advantages of MLPs that haven't been mentioned include their fault tolerance, the ability to produce outputs from incomplete inputs after training, and having structure that allows for parallel processing [REF7]. These models can also handle large amounts of input data, making them ideal for processing images and audio. The limitations of MLPs can mainly be attributed to the lack of methodology for determining network structure, resulting in lots of trial and error, as well as the need for the inputs to be numerical values, often resulting in the need for encoding [REF7]. MLPs are also not guaranteed to find global minima of the loss function, although momentum can be applied during the training process to counteract this.

SVCs reduce computation time via the kernel trick - employing kernel functions to calculate distance between points in high-dimensional space using dot products, as opposed to explicitly calculating their coordinates. The most common kernel used to solve complex problems is the radial kernel, which calculates distance in infinite-dimensional space. Another commonly used kernel in SVCs is the polynomial kernel, which computes distance in a dimensional space where the number of dimensions acts as a hyperparameter.

The kernel trick is utilised to fit a hyperplane decision boundary, to split the data into classes, by minimising the hinge loss function, which has two components. The first has the effect of maximising the soft margin - the distance between the decision boundary and a pair of parallel hyperplanes, which pass through a single observation from each class. The observations on, as well as between, the soft margin boundaries are referred to as support vectors. The soft margin allows misclassifications in order to avoid overfitting, hence the misclassified samples are the second component of the loss function. The regularising parameter, C, dictates the tradeoff between the two components of the loss function. Larger values of C will make the model prioritise minimising the number of misclassifications, corresponding to high bias and low variance, whereas smaller values of C result in more tolerance of misclassifications.

The main advantage of SVCs is that they are convex optimisation problems, meaning they have unique global minima. Other advantages include their ability to fit decision boundaries in a high number of dimensions without having to calculate their coordinates. In contrast, SVCs face issues in the computation time required for their training as well as the cross-validation required to find the best model hyperparameters. This model is also less suitable for large datasets in comparison to MLPs.

Hypothesis Statement

Due to the convex nature of the SVC optimisation problem, it is predicted that the SVC will slightly outperform the MLP. SVCs generate near optimal classifications as they obtain the optimum separating surface which has good performance on previously unseen data points [REF8]. In contrast, the training time for SVCs are longer than training times for MLPs, which means that SVC hyperparameter tuning must be run on a smaller number of trials so that computation time is manageable. The lower number of hyperparameters that need to be tuned for SVCs, in comparison to MLPs, mean that the lower number of trials used for optimisation shouldn't prevent acquisition of a near-optimal set of hyperparameters.

Training and Evaluation Methodology

Training and test sets are acquired through a stratified train test split, with 30% of the instances in the data being used for testing. Stratification preserves the class proportions in the original dataset, meaning the classes are evenly distributed within the training and test sets. The training and test sets are the same for both models to allow for comparison, although the input data for the SVC is normalised as it is required for the model to run.

Hyperparameter tuning is achieved using optuna, which attempts to maximise a 5-fold stratified cross-validation metric using the training set. Cross validation is used to improve estimates of the generalisation error of the model. Splitting the training set into 5 folds, training the model on 4 folds before testing on the remaining fold. This is implemented with each fold taking a turn as the validation set, before the test results are averaged to give a 5-fold cross-validation score.

Evaluation of the models is largely achieved using accuracy, as is the case in the hyperparameter tuning and final model testing sections, due to the task being a binary classification problem.

Other metrics utilised in evaluation - precision, recall and f1 scores - are used to individually evaluate the performance of the models for separate classes. Precision corresponds to the proportion of positive identifications that were correct for a given class. Recall is the proportion of positive instances of a class that were correctly predicted, and often has a trade-off with precision in a model. In this particular problem, for Class 1, recall is considered to be of higher importance than precision due to the health concerns that incur a heavy cost of a false negative classification in the context of diabetes and prediabetes. F1 scores are calculated by taking the harmonic mean of the precision and recall scores and allow for a summary of a model's performance in classifying instances belonging to a particular class. Micro and macro averages can be taken of these three scores to obtain singular score values for a model.

The final metrics used for evaluation are receiver operating characteristic (ROC) curves and area under the ROC curve (AUC). ROC curves plot the true positive rate against the false positive rate whilst varying the decision threshold. The AUC acts as a summary of the ROC curve, providing a singular metric value, used to evaluate the model's ability to distinguish between classes.

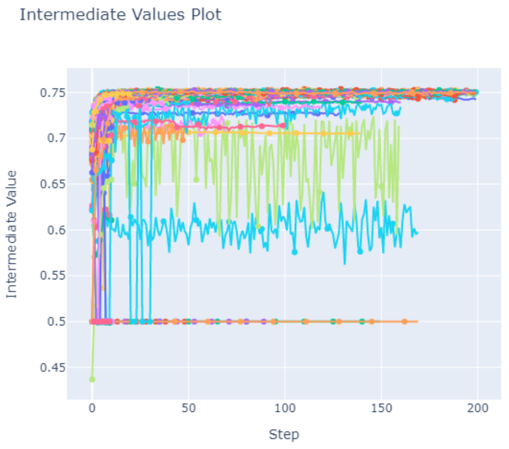
Parameter Selection and Experimental Results

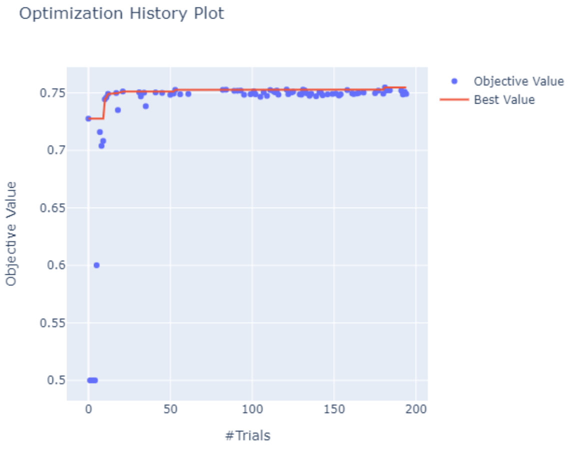
Hyperparameter tuning for both models involved maximising 5-fold cross-validation accuracy across many different combinations of hyperparameter values. The Optuna package was used due to its utilisation of a searching method that tests hyperparameter values in regions similar to values that achieved high validation accuracy in earlier trials.

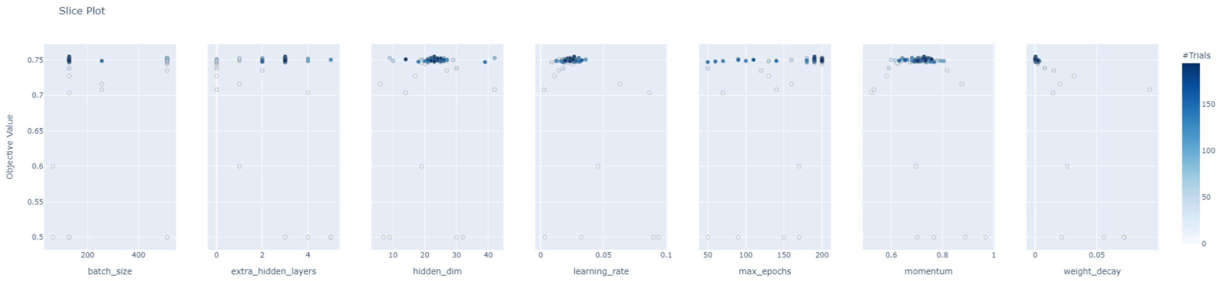
Computation time was reduced via pruning in the hyperparameter tuning for the MLP, using the median stopping rule, as the high number of hyperparameters meant that a higher number of trials were run.

In the case of binary classification, the MLP output layer consists of one neuron, with a sigmoid activation function that converts the neuron's value to represent the probability of the target class being Class 1. The loss function used is binary cross entropy loss (BCE), which is minimised using gradient descent with momentum.

The network-structure hyperparameters tuned for the MLP model included the number of hidden layers, ranging from 1 to 6, and the hidden layer size, which ranged from 4 to double the input size. Optimiser hyperparameters that were tuned were the learning rate, momentum and weight decay. Learning rate and values were tested in a commonly-accepted range, between 0.0001 and 0.1. Momentum is used to try to help the model to find global minima in the loss minimisation problem, the most common value used is 0.9 so values were tested in the range 0.5 up to, and not including, 1. The weight decay parameter acts as a coefficient to the L2 regularisation term in the loss function, determining the level of regularisation employed to avoid overfitting caused by overly complex models. L2 regularisation reduces the complexity of the models by decaying weights towards zero. Common values for weight decay lie in a similar range to that of the learning rate and, as such, the value is optimised over the same range as the learning rate. The two remaining hyperparameters that were tuned are the size of the batches used for mini-batch gradient descent and the maximum number of training epochs. Higher batch sizes lead to fewer parameter updates, as the weights and biases are updated at the end of each batch. Batch sizes are tested on powers of 2, from 2^6 to 2^9, due to the number of processors in a GPU being a power of 2 - this means that the batches can be split evenly across the GPU processors. The maximum number of training epochs was tuned over multiples of 10 in the range 50 to 190. The hyperparameter tuning plots shown below indicate that a similar validation accuracy could have still been achieved if the number of trials had been heavily reduced, with the optimal hyperparameter combination being found just after 50 trials.

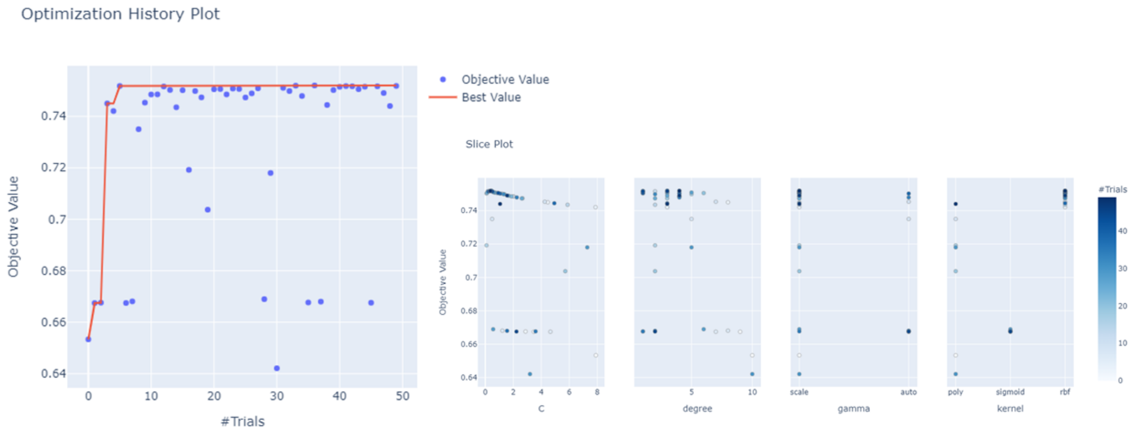






Hyperparameters for the SVC model are less numerous, while the training time for the model is much longer, so only 50 combinations of hyperparameters were tested in contrast to the 200 combinations tested for the MLP model. The first hyperparameter is the kernel function, which is tested over three different types of kernel - the polynomial kernel, the radial kernel and the sigmoid kernel. The second parameter, degree, only applies to the polynomial kernel, which requires the degree of the kernel to be specified. This value is tested over the range 1 to 9, where a degree of 1 would result in a linear kernel. The regularisation parameter, C, is tested over the range 2^{-5} to 2^3. The final hyperparameter is gamma, which determines how far away from the decision boundary an observation can be, whilst influencing the placement on the decision boundary. Higher values of gamma decrease the distance of influence, while lower values increase the distance of influence. Gamma is tested over two values - auto, which sets gamma to the reciprocal of the number of features, and scale, which uses the reciprocal of the number of features multiplied by the variation of the attribute values.

The optimisation history for the SVC also indicates a similar validation accuracy could have been achieved in fewer trials. The slice plots show that rbf kernels consistently outperform other types of kernels, while the other hyperparameters show a wider range of values over which the validation accuracy is similar.

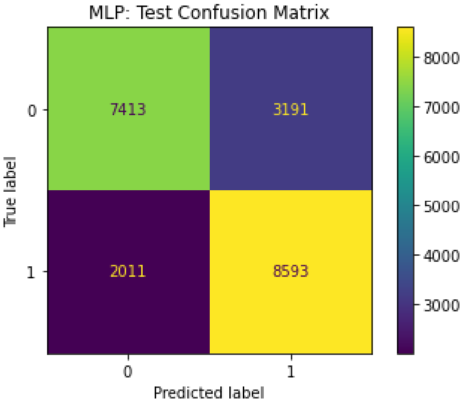


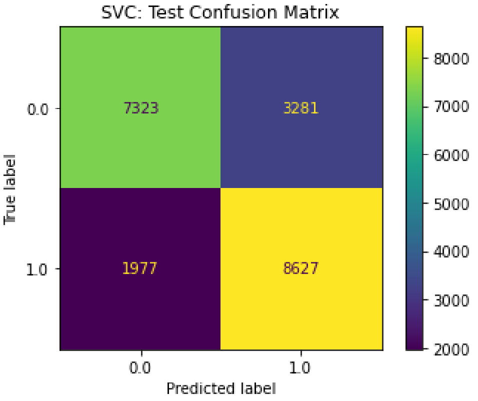
Analysis of Results

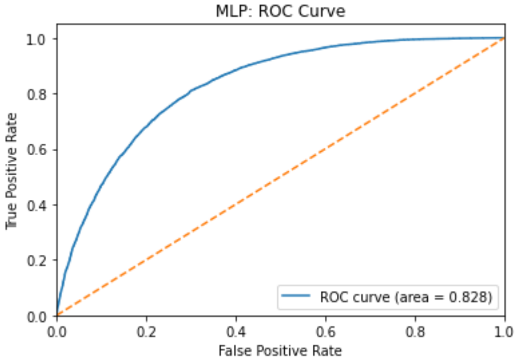
The MLP was trained with the tuned hyperparameters - 4 hidden layers, each with 23 neurons, a batch size of 128, learning rate equal to 0.026645655560194527, momentum set to 0.7281967530014857, a weight decay of 0.0003253167822144232 and a maximum of 190 training epochs. After training, the tuned MLP achieved an accuracy score of 75.47% (to 2 s.f.), when tested on the test set.

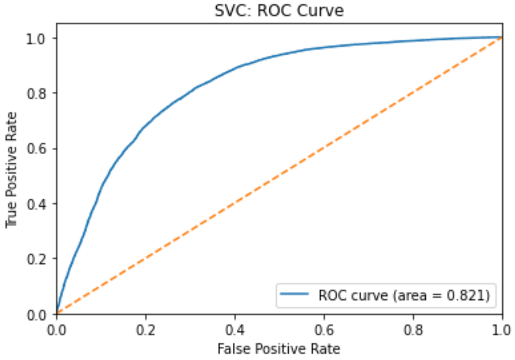
The final SVC was trained on the hyperparameters found during optimisation - a radial kernel, which removes the significance of the degree parameter, with the C regularisation parameter set to 0.4055376523017632 and the gamma parameter set to the reciprocal of the number of features, multiplied by the variation of the attribute values. Testing on the normalised version of the test set, the final SVC model performed with an accuracy score, similar to the MLP model, of 75.20% (to 2 s.f.).

The confusion matrices show little distinction between the two models, with the values in corresponding quadrants being in similar ranges. Despite having a slightly lower accuracy score, analysing the confusion matrices for both models, the SVC model is shown to correctly classify 34 more instances belonging to the diabetes and prediabetes class, in comparison to the MLP. A false negative classification for someone with diabetes or prediabetes could mean that they do not receive the medical attention and information that they need in order to maintain their health. As a result, the cost of a false negative classification is much greater than the cost of a false positive, suggesting that the SVC could be considered the best of the two models in the given context.









ROC curves were plotted to provide a visual comparison of the relationship between false positives and false negatives across different classification thresholds. The initial slope of the MLP's curve is steeper than that of the SVC's, however the gradient of the MLP's curve begins decreasing earlier. This results in the both ROC curves having very similar AUC values: 0.828 for MLP and 0.821 for SVC. While both AUC values being above 0.8 suggests that both models have good distinguishing power, the difference in values indicates that the MLP is slightly better at distinguishing between classes.

The final metrics that were examined for each model are the precision, recall and F1 scores for both classes, as well as the macro-average of each of these scores. Micro averages are excluded from this report because it is a weighted average that is only applicable for datasets with an unbalanced target class. The target class for the data used in this report is balanced and, thus, macro and micro averages of these metrics are equal. Values for precision and recall are very similar for both models, as is reflected by the confusion matrices. The values for recall for Class 1, which is the priority in this context, are the highest metric values in both models - although the minute difference between these values, demonstrated in the confusion

Model	Class or Average	Precision	Recall	F1
MLP	Class 0	0.79	0.70	0.74
	Class 1	0.73	0.81	0.77
	Macro Average	0.76	0.75	0.75

matrices, is lost due to rounding. Macro-averaged precision is higher than the macro-averaged recall in both models, which indicates that the precision recall trade-off is in favour of the quality of positive predictions as opposed to the quantity of positively predicted instances that belong to a class. Values for F1 score, as well as corresponding macro-averages are identical to two decimal places in both models, providing little distinction in performance.

Conclusions

This report evaluates and compares two neural network models, an MLP and an SVC, designed to classify patients into the one of the classes - non-diabetic (Class 0), and prediabetic or diabetic (Class 1) - based on 21 health-related attributes collected from a survey.

The final models are shown to perform with very similar metric values, from accuracy and AUC to precision, recall and F1 score. Although the MLP model is shown to achieve slightly higher accuracy and AUC, the SVC model correctly classifies a higher proportion of data instances with a target class value corresponding to prediabetes or diabetes. It is thus arguable as to which model has better performance, as the MLP performs better in the more general metrics, whereas the SVC performs better if the context-specific objectives of the model are taken into account.

Improvements could be made to the models via techniques similar to the bootstrap method, which can be applied to determine ideal network architectures [REF9]. Creating ensemble classifiers of either of these types of models could increase accuracy, while allowing for simplicity in the individual models within the ensemble. Further improvements could involve implementing early stopping, used to avoid overfitting that occurs at a certain point in training, when the training error decreases while the test error increases [REF10]. Our models could also be improved by using a larger number of trials in the hyperparameter tuning, allowing for more combinations to be tested. Optimisation, or consideration, of different metrics during hyperparameter training could yield results that are more tailored to the context of the data. Computation time could also be interpreted as a metric when searching for ideal model structures, as practicality is also important. Finally, experimentation with different gradient descent methods, such as the Adam optimiser, as well as experimentation with dropout regularisation, could have generated interesting results for the MLP model.

References

[1] L. S. Geiss et al., “Prevalence and incidence trends for diagnosed diabetes among adults aged 20 to 79 years, United States, 1980-2012,” JAMA, vol. 312, no. 12, pp. 1218–1226, 2014.

[2] “Statistics about diabetes | American Diabetes Association,” Diabetes.org. [Online]. Available: https://www.diabetes.org/about-us/statistics/about-diabetes. [Accessed: 08-May-2022].

[3] American Diabetes Association, “Economic costs of diabetes in the U.s. in 2017,” Diabetes Care, vol. 41, no. 5, pp. 917–928, 2018.

[4] A. Teboul, “Diabetes Health Indicators Dataset, Kaggle.” .

[5] S. O. Haykin, Neural networks and learning machines: International edition, 3rd ed. Upper Saddle River, NJ: Pearson, 2010.

[6] S. Sharma, S. Sharma, and A. Athaiya, “ACTIVATION FUNCTIONS IN NEURAL NETWORKS,” Ijeast.com. [Online]. Available: https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf. [Accessed: 08-May-2022].

[7] M. Mijwil, “Artificial Neural Networks Advantages and Disadvantages,” Jan. 2018.

[8] E. A. Zanaty, “Support Vector Machines (SVMs) versus Multilayer Perceptron (MLP) in data classification,” Egypt. Inform. J., vol. 13, no. 3, pp. 177–183, 2012.

[9] R. Kallel, M. Cottrell, and V. Vigneron, “Bootstrap for neural model selection,” Neurocomputing, vol. 48, no. 1–4, pp. 175–183, 2002.

[10] L. Prechelt, “Early stopping - but when?,” in Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 55–69.

Appendix:

Glossary:

- Accuracy: The fraction of predictions that a classification model got right.
- Activation function: A function that takes in the weighted sum of all of the inputs from the previous layer and then generates and passes an output value (typically nonlinear) to the next layer.
- Attribute: Synonym for feature.
- AUC (Area under the ROC curve): An evaluation metric that considers all possible classification thresholds.
- Backpropagation: The primary algorithm for performing gradient descent on neural networks. First, the output values of each node are calculated (and cached) in a forward pass. Then, the partial derivative of the error with respect to each parameter is calculated in a backward pass through the graph.
- Batch: The set of examples used in one iteration of model training.
- Batch size: The number of examples in a batch.
- Bias: An intercept or offset from an origin.
- Binary classification: A type of classification task that outputs one of two mutually exclusive classes.
- Categorical data: Features having a discrete set of possible values.
- Class: One of a set of enumerated target values for a label.
- Classification model: A type of model that distinguishes among two or more discrete classes.
- Classification threshold: A scalar-value criterion that is compared to a model's predicted score in order to separate the positive class from the negative class.
- Confusion matrix: An NxN table that aggregates a classification model's correct and incorrect guesses.
- Convex function: A function in which the region above the graph of the function is a convex set.
- Cross-entropy: A generalisation of Log Loss to multi-class classification problems.
- Cross-validation: A mechanism for estimating how well a model would generalise to new data by testing the model against one or more non-overlapping data subsets withheld from the training set.
- Decision boundary: The separator between classes learned by a model in a binary class or multi-class classification problems.
- Discrete feature: A feature with a finite set of possible values.
- Dropout regularisation: A form of regularisation useful in training neural networks.
- Early stopping: A method for regularisation that involves ending model training before training loss finishes decreasing.
- Ensemble: A collection of models trained independently whose predictions are averaged or aggregated.
- Epoch: A full training pass over the entire dataset such that each example has been seen once.
- False negative: An example in which the model mistakenly predicted the negative class.
- False negative rate: The proportion of actual positive examples for which the negative class is predicted.
- False positive: An example in which the model mistakenly predicted the positive class.
- False positive rate: The x-axis in an ROC curve.
- Feature: An input variable used in making predictions.
- Feedforward neural network: A neural network without cyclic or recursive connections.
- Fully connected layer: A hidden layer in which each node is connected to every node in the subsequent hidden layer.
- Generalisation: Refers to your model's ability to make correct predictions on new, previously unseen data.
- Gradient: The vector of partial derivatives with respect to all of the independent variables.
- Gradient descent: A technique to minimise loss by computing the gradients of loss with respect to the model's parameters, conditioned on training data.
- Hidden layer: A synthetic layer in a neural network between the input layer and the output layer.
- Hinge loss: A family of loss functions for classification designed to find the decision boundary as distant as possible from each training example, thus maximising the margin between examples and the boundary.
- Hyperplane: A boundary that separates a space into two subspaces.
- Input layer: The first layer in a neural network.
- Instance: Synonym for example.
- Iteration: A single update of a model's weights during training.
- L2 regularisation: A type of regularisation that penalises weights in proportion to the sum of the squares of the weights.
- Label: In supervised learning, the "answer" or "result" portion of an example.
- Learning rate: A scalar used to train a model via gradient descent.
- Loss: A measure of how far a model's predictions are from its label.
- Mean Squared Error (MSE): The average squared loss per example.
- Metric: A number that you care about.
- Mini-batch gradient descent: A gradient descent algorithm that uses small, randomly selected subsets of the entire batch of examples run together in a single iteration of training or inference.
- Momentum: A sophisticated gradient descent algorithm in which a learning step depends not only on the derivative in the current step, but also on the derivatives of the step(s) that immediately preceded it.
- Neural network: A model that, taking inspiration from the brain, is composed of layers consisting of simple connected units or neurons followed by nonlinearities.
- Neuron: A node in a neural network, typically taking in multiple input values and generating one output value. Also referred to as a node
- Normalisation: The process of converting an actual range of values into a standard range of values, typically -1 to +1 or 0 to 1.
- Objective function: The mathematical formula or metric that a model aims to optimise.
- Output layer: The "final" layer of a neural network.
- Overfitting: Creating a model that matches the training data so closely that the model fails to make correct predictions on new data.
- Parameter: A variable of a model that the machine learning system trains on its own.
- Perceptron: A system (either hardware or software) that takes in one or more input values, runs a function on the weighted sum of the inputs, and computes a single output value.
- Precision: A metric that identifies the frequency with which a model was correct when predicting the positive class.
- Recall: A metric for classification models that answers the following question: Out of all the possible positive labels, how many did the model correctly identify?
- Rectified Linear Unit (ReLU): A type of activation function.
- Regularisation: The penalty on a model's complexity.
- Receiver operating characteristic (ROC) curve: A curve of true positive rate vs. false positive rate at different classification thresholds.
- Sigmoid: Another type of activation function.
- Target: Synonym for label.
- Test set: The subset of the dataset that you use to test your model on after training.
- Training: The process of determining the ideal parameters comprising a model.
- Training set: The subset of the dataset used to train a model.
- True negative: An example in which the model correctly predicted the negative class.
- True positive: An example in which the model correctly predicted the positive class.
- True positive rate: Synonym for recall.
- Validation: A process used, as part of training, to evaluate the quality of a machine learning model using the validation set.

Attribute Numerical Encoding Guide:

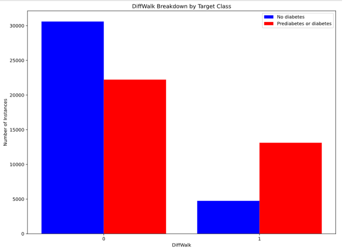
The values below are taken from the Kaggle dataset page, as well as the BRFSS's 2015 codebook. The websites can be found in the two links below:

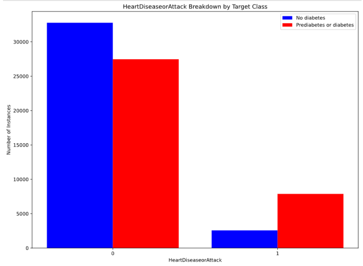
- https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?select=diabetes_012_health_indicators_BRFSS2015.csv
- https://www.cdc.gov/brfss/annual_data/2015/pdf/codebook15_llcp.pdf

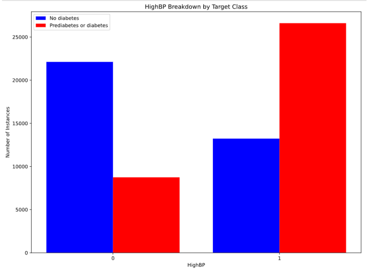
HighBP: 0 = no high blood pressure, 1 = high blood pressure
HighChol: 0 = no high cholesterol, 1 = high cholesterol
CholCheck: 0 = no cholesterol check in 5 years, 1 = yes cholesterol check in 5 years
Smoker (Have you smoked at least 100 cigarettes in your entire life?): 0 = no, 1 = yes
Stroke: 0 = not had a stroke, 1 = had a stroke
HeartDiseaseorAttack (coronary heart disease or myocardial infarction): 0 = no, 1 = yes
PhysActivity (physical activity in past 30 days - not including job): 0 = no, 1 = yes
Fruit (consume fruit 1 or more times a day): 0 = no, 1 = yes
Veggies (consume vegetables 1 or more times a day): 0 = no, 1 = yes
HvyAlcoholConsumption (adult men having more than 14 drinks per week and adult women having more than 7 drinks per week): 0 = no, 1 = yes
AnyHealthcare (Have any kind of health care coverage, including health insurance, prepaid plans such as HMO, etc.): 0 = no, 1 = yes
NoDocbcCost (Was there a time in the past 12 months when you needed to see a doctor but could not because of cost?): 0 = no, 1 = yes
GenHlth (Would you say that in general your health is): 1 = excellent, 2 = very good, 3 = good, 4 = fair, 5 = poor
MentHlth (Now thinking about your mental health, which includes stress, depression, and problems with emotions, for how many days during the past 30 days was your mental health not good?): number of days
PhysHlth (Now thinking about your physical health, which includes physical illness and injury, for how many days during the past 30 days was your physical health not good?): number of days
DiffWalk (Do you have serious difficulty walking or climbing stairs?): 0 = no, 1 = yes
Sex: 0 = female, 1 = male
Age (13-level age category): 1 = 18-24, 2 = 25-29, 3 = 30-34, 4 = 35-39, 5 = 40-44, 6 = 45-49, 7 = 50-54, 8 = 55-59, 9 = 60-64, 10 = 65-69, 11 = 70-74, 12 = 75-79, 13= 80+
Education (What is the highest grade or year of school you completed?): 1 = Never attended school or only kindergarten, 2 = Grades 1 through 8 (Elementary), 3 = Grades 9 through 11 (Some high school), 4 = Grade 12 or GED (High school graduate), 5 = College 1 year to 3 years (Some college or technical school), College 4 years or more (College graduate)
Income (Annual household income from all sources): 1 = Less than $10,000, 2 = $10000-$15000, 3 = $15000-$20000, 4 = $20000-$25000, 5 = $25000-$35000, 6 = $35000-$50000, 7 = $50000-$75000, 8 = $75000+

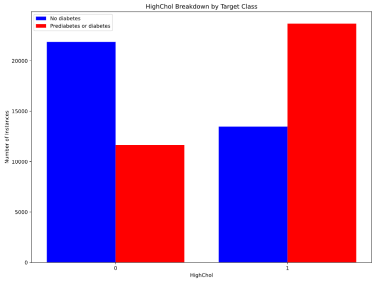
EDA Graphs:

The graphs mentioned, but not shown, in the report are on the right hand side of this page in the order (from top to bottom): DiffWalk, HeartDiseaseorAttack, HighBP, HighChol.









Implementation details:

The MLP model was constructed from a created nn.Module class, wrapped inside a skorch NeuralNetBinaryClassifier. The SVM model is constructed using sklearn's SVC function. Hyperparameter tuning for both models was carried out using optuna, with relevant graphs produced by plotly. Once hyperparameters have been tuned, each model has been trained and saved to the final_models folder. The final_model_tests notebook is used to import the trained final models and test them using the test set, producing confusion matrices and ROC curves using sklearn functions.