Exam Answer Sheet

A.D. van der Merwe
Department of Computer Science
University of Stellenbosch
24923273
24923273@sun.ac.za

October 30, 2024

Section A: Introduction to Machine Learning

Question 1

Ockham's razor is a principle that states if there are multiple ways to explain a phenomenon, the simplest explanation that still effectively explains the phenomenon should be used.

Regarding predictive models, Ockham's razor is crucial in model selection and development. A simple model that is less computationally expensive or a model with fewer parameters and assumptions should be favoured if the prediction accuracy is sufficient enough for the task at hand.

For example, if a classification decision tree model achieves similar results to a more complex random forest classification model, the simple classification tree is preferred to perform predictions for the specific task. Simpler models are also easier to interpret, are less prone to overfitting on the training data, and are much less computationally expensive.

Question 2

Variance in a model refers to the extent to which a prediction varies when different test sets are used to construct the model [1]. It is almost impossible to find real-world data where there is no variability in the dataset. Therefore there will always be variability in each training dataset and the prediction of a model will almost always result in different values. Ideally should the prediction of an instance not vary too much between training sets. High variance indicates that the model may be overfitting, meaning it captures noise and specific patterns from the training data and generalises poorly to unseen data. This sensitivity to small changes in the training data can result in significant fluctuations in predictions. To calculate the variance of a regression model, the following equation is used.

$$Var(\hat{y}) = \mathbb{E}\left[\left(\hat{y} - \mathbb{E}\left[\hat{y}\right]\right)^2\right] \tag{1}$$

where $\hat{y}$ is the predicted value of the regression model and $\mathbb{E}\left[\hat{y}\right]$ is the expected predicted value across different training sets.

Bias in a model refers to the error introduced by approximating a real-world problem, which may be extremely complicated, by a much simpler model. A model with high bias can lead to inaccurate predictions by underfitting the data as a simple model might not be able to capture the complex relationships between the descriptive features and the target feature. A high bias indicates that the model's predictions deviate significantly from the actual values, often due to an overly simplistic model that cannot capture the underlying complexities of the data. To calculate the bias of a regression model, the following equation is used.

$$Bias(\hat{y})^2 = \left(\mathbb{E}[\hat{y}] - y\right)^2 \tag{2}$$

where y is the actual value that the model aims to approximate.

The expected test mean squared error (MSE) can be decomosed into the sum of three fundamental quantities which is given in the following equation.

$$\mathbb{E}\left(y-\hat{y}\right)^2 = Var(\hat{y}) + Bias(\hat{y})^2 + \sigma^2 \tag{3}$$

where σ^2 is the irreducible error, which is the noise in the observations that cannot be reduced by any model.

From equation (3) it is clear that both the variance and the bias of a model has to be small values to minimise the test MSE. The bias-variance dilemma arises from this. The more complex a model becomes to capture relationships between the descriptive features and the target feature, the more susceptible the model becomes to overfitting on the data, therefore the reduction in bias may lead to an increase in variance. Conversely, if a model is kept simple to prevent overfitting, the model may struggle to capture complex relationships between the descriptive features and the target feature, therefore the reduction in variance may lead to an increase in bias.

Question 3

(a)

Underfitting occurs when a predictive model is too simplistic to capture the underlying relationships between the descriptive features and the target feature resulting in poor predictive performance for both the training and test datasets.

(b)

Overfitting occurs when the predictive model is overly complex, leading it to become sensitive to noise and fluctuations in the training dataset. This results in the model fitting too closely to the training data, where the complex underlying relationships between the descriptive features and target features are captured, but the model learns the noise present in the training dataset. This then results in a good predictive performance for the training set, but the generalisation too the unseen test dataset will be poor.

(c)

When a model is underfitting, it fails to learn underlying relationships between the descriptive features and the target feature, thus resulting in a high bias and a low variance. Conversely, when a model is overfitting, it accurately learns the underlying relationships between the descriptive features and the target feature but also captures noise within the dataset. This leads to the model becoming sensitive to small fluctuations in the training dataset, thus resulting in a high variance and a low bias.

Question 4

When a dataset contains outliers, it is generally a bad idea to impute missing numerical values by use of the mean as the mean is influenced by the outliers that leads to an inaccurate representation of the central dendency of the data. It is generally better to use the median of the numerical values of a feature when outliers are present in the feature values. The median is less affected by outliers and provides a better measure of central tendency of the feature values if these values are not normally distributed.

Additionally considering more advanced imputation methods that accounts for the relationships between the descriptive features can yield better results. This is especially the case for when the data structure is complex. A k-nearest neighbours (KNN) classifier technique can be used to identify the k-nearest instances to the observation with a missing value based on the other features similarity. The missing value is then imputed using the mean or median of the feature values from the k-nearest instances, which often results in a more robust and representative imputed value.

When a classification task is considered, it is generally better to impute the missing value of a feature with only the mean or median features reletad to the class which contains the missing value. This approach leverages the conditional relationships between the descriptive features and the target feature, which could lead to more accurate imputations.

Question 5

Feature selection is a crucial step when preprocessing data to construct a predictive machine learning. Reducing the number of features in the dataset will improve the performance of the model, reduce overfitting, result in faster training times, create a simpler model which is more interpretable and improve the data quality.

The model's performance improves, because only the most relevant descriptive features are chosen to perform predictions of the target feature, which often leads to improved accuracy and predictive power. Additionally, overfitting is reduced as the reduction in dimensionality of the original dataset leads to models that are less complex and reduces the chances of capturing noise in the dataset and increases the chance of the model learning the underlying relationships between the remaining descriptive features and the target feature. The dimensionality reduction also leads to faster training times as the model is constructed on less data, which is very beneficial when large datasets are used and the models are easier to interpret, which is beneficial when a model is used where the decision-making process is important to understand. Lastly, the quality of the data is improved as features with noise or irrelevant features are removed.

Question 6

(a)

Stationary data is data that has statistical properties which stays consistent over time, such as the mean and variance. This means that the behavior of the dataset does not significantly change over time and stays more or less constent. Conversely non-stationary data has statistical properties which changes over time. Non-stationary data may show increasing or decreasing means and changing variances, due to trends or even seasonal effects, making it less predictable.

(b)

A real world problem that exhibits non-stationary data is stock market data. This is because stock market data exhibit upward and downward trends over time, which is influenced by different factors such as economic growth and some stocks may exhibit seasonal patterns, such as retail stocks that perform better during the holiday.

(c)

The main implications are that the models can become unstable over time, the model needs to consantly learn and the models might overfit past historical patterns.

The first implications is that the model may become unstable overtime and lead to invalid predictions, as the relationships between the descriptive features and target feature may change over time. This makes previously learned patterns unreliable and justifies the need for the second implication, which is that the model needs to consantly be updated. The model has to incorporate some form of continuous learning or regular retraining to ensure that the models remain relevant and can adapt as soon as the underlying patterns between the descriptive features and the target feature changes over time. The models may also overfit to past historical, which leads to poor generalisation. This may lead to high error rates when the model is applied to current or future data.

Question 7

The consequences of using too few bins is that important information may be lost with respect to the distribution of values in the original continuous feature. This loss of information may lead to reduced predictive power of the model, as the small number of bins reduces the feature's ability to differentiate between classes or predict outcomes accurately. A small number of bins can also potentially hide variations or patterns that could be useful for modeling.

Conversely, the consequences of using too many bins, is that each bin will have a small number of instances contained within them, where some bins may end up having no instances. Having too many bins may lead to a more computationally expensive model. Additionally, too many bins may increase the complexity of the model, such as a classification tree that has to evaluate a greater number of categories.

Question 8

(a)

TODO

(b)

The distribution of ID (d) indicates a feature that should be removed, as the feature contains unique values. This feature has no predictive information relevant to the target feature and therefore it should be removed.

(c)

TODO

Section B: Information-based Learning

Question 1

e

Question 2

(a)

Firstly, the equation used to calculate the entropy of the dataset D is defined as following:

$$H(D) = -\sum_{m=1}^{M} p(y_m) log_M \left(p(y_m)\right)$$

where the probability of a class y_m occuring in D is given as

$$p(y_m) = \frac{\text{freq}(y_m, D)}{|D|}$$

where M is the number of classes, and $\text{freq}(y_m, D)$ is the number of times that class y_m occurs in D.

The calculation of the question is then as follows.

$$\begin{aligned}
p(2) &= \frac{1}{8} \\
p(3) &= \frac{1}{8} \\
p(4) &= \frac{1}{8} \\
p(5) &= \frac{1}{8} \\
p(6) &= \frac{2}{8} \\
p(7) &= \frac{1}{8} \\
p(8) &= \frac{1}{8}
\end{aligned}$$

$$
\begin{aligned}
p(2)log_7\left(p(2)\right) = \frac{1}{8}log_7\left(\frac{1}{8}\right) &= -0.1335776952 \\
p(3)log_7\left(p(3)\right) = \frac{1}{8}log_7\left(\frac{1}{8}\right) &= -0.1335776952 \\
p(4)log_7\left(p(4)\right) = \frac{1}{8}log_7\left(\frac{1}{8}\right) &= -0.1335776952 \\
p(5)log_7\left(p(5)\right) = \frac{1}{8}log_7\left(\frac{1}{8}\right) &= -0.1335776952 \\
p(6)log_7\left(p(6)\right) = \frac{2}{8}log_7\left(\frac{2}{8}\right) &= -0.1781035936 \\
p(7)log_7\left(p(7)\right) = \frac{1}{8}log_7\left(\frac{1}{8}\right) &= -0.1335776952 \\
p(8)log_7\left(p(8)\right) = \frac{1}{8}log_7\left(\frac{1}{8}\right) &= -0.1335776952
\end{aligned}
$$

$$
\begin{aligned}
H(D) &= -\sum_{m=1}^{7} p(y_m)log_7\left(p(y_m)\right) \\
H(D) &= -\left(6 \times (-0.1335776952) + (-0.1781035936)\right) \\
H(D) &= -(-0.9795697645) \\
H(D) &= 0.9795697645
\end{aligned}
$$

(b)

TODO

Question 3

TODO

Question 4

The first inductive bias is that the decision boundaries can only be parallel to the axis. The consequence of this is that more complex trees are needed if the true decision boundaries are not parallel to the axes. This inductive bias can be addressed by using oblique trees.

The second inductive bias is that the tree is induced to overfit the training data. Meaning that the tree creates decision boundaries until all of the classes are separated and achieves a perfect score on the training data. The consequence of this inductive bias is that the tree generalises poorly to unseen data. This inductive bias can be addressed by either pruning the tree while it is being induced or by pruning the induced tree.

The third inductive bias is that if the information gain is maximised, descriptive features with many outcomes are favoured higher up in the tree. This inductive bias results into an induced tree with many decision rules and trees with high branching factors high up in the tree. This inductive bias can be addressed by using the gain ratio criterion.

The last inductive bias is that decision tree algorithm prefers to induce shorter trees [2]. The consequence of this inductive bias is that shoter trees might note always capture the complex relationships between descriptive features and the target feature. This inductive bias can be addressed by using ensemble techniques such as boosting.

Question 5

An oblique tree algorithm would result in the smallest tree structure for this clasification problem, as the classification problem represents the classification trees, such as C4.5 and ID3's, inductive bias of only creating axis-parallel decision boundaries.

Oblique trees, by contrast, allow splits that are not parallel to any axis, enabling the model to create boundaries at arbitrary angles. Fewer nodes are then required to split fully induce the tree to overfit on this particular classification task and therefore the tree resulting from the oblique tree algorithm would result in the smallest tree.

Question 6

(a)

TODO

(b)

TODO

Question 7

TODO

Question 8

TODO

Section C: Similarity-based Learning

Question 1

b

Question 2

b, c

Question 3

b, c, g

Question 4

The inductive bias of the k-nearest neighbour algorithm is instances that have similar descriptive feature values also have the same target feature values. Meaning that if two instances are close in terms of their descriptive feature values, the k-nearest neighbours algorithm assumes that these instances belongs to the same class for classification tasks or similar outputs for regression tasks.

Question 5

For algorithms like k-nearest neighbours that makes use of similarity based learning it is important to normalise the input descriptive features so that each descriptive feature has exactly the same range. The features should be normalised to the same range as as input features with larger differences in the range has a stronger impact on the distance calculations used in the k-nearest neighbours. A descriptive feature with a range of [1000,100000] will have a much stronger impact on the distance calculation than a feature with a range of [10,100]. Therefore these features should be normalised to a specific range such as [0,1].

Question 6

The k-nearest neighbours algorithm can be used to impute a missing value of a feature by calculating the similarity between instances. For an instance that contains a missing value in one of the descriptive features, the k-nearest neighbours algorithm can be used to examine the k most similar instances in the dataset, where the similarity is calculated based on the features that does not contain missing values. After the k most similar neighbours of this instance has been found, the missing value is imputed by use of the mean or median of the descriptive feature that contains the missing value of these k neighbours for a numerical descriptive feature. Conversely for a categorical descriptive feature, the mode of the descriptive feature that contains the missing value of the k neighbours are used to impute the missing value.

Question 7

The k-nearest neighbour algorithm can be applied to problems with categorical descriptive features. The categorical descriptive features should either be one-hot encoded or ordinally encoded to ensure the categories are represented in numerical form if they are not already. If the categorical features have been one-hot encoded, a variety of distance calculations can be used to calculate the similarity between observations such as the Jaccard, Hamming, Manhattan, Euclidean and cosine distance metrics. For ordinally encoded features, the similarity between observations can be calculated by use of the Jaccard similarity, which calculates the number of features wit the same value.

Question 8

If the number of neighbours, k, in the algorithm is small, the algorithm will become sensitive to noise in the target feature. The predicted value of an instance can be negatively influenced if k is small and there are a few neighbours around a particular instance with an abnormal target value due to noise or outliers. Therefore, the outcome may be skewed as there are fewer neigbours to average out the noise.

Conversely, if the number of neighbours, k, in the algorithm is large, the predictions of the algorithm will be bad as the average of the target features of the k-nearest neighbours is calculated and used as the predicted value of the instance. Therefore, a larger value of k could lead to an algorithm that underfits the data, failing to capture local patterns effectively.

Section D: Error-based Learning

Question 1

c

Question 2

d

Question 3

c

Question 4

a, b, g

Question 5

(a)

Yes, normalising or scaling of the target feature is required. This is because the output of the sigmoid activation function returns a value in the range $(0, 1)$. In the case of a regression problem, the target features should be scaled to the range of $(0, 1)$ to match the output of the sigmoid activation function. In the case of binary classification, the two target classes should be ordinally encoded to 0 and 1, which represent the probability of the observation belonging to class 1. If the target features are not scaled, the neuron will always produce a large error signal, which leads to a continuous adjustment of the weights, meaning the model would never converge.

(b)

Yes, it is prudent to normalise or scale the input features in this scenario. It is not necessary to scale the input feature values, but the performance of the model can be improved if the inputs are scaled to the active domain of the activation function. For the sigmoid activation function, the active domain is $[-\sqrt{3}, \sqrt{3}]$. This corresponds to the parts of the sigmoid function for which weight changes in the input features has a relatively large change in output. The values beyond these points would have a very small influence on the weight updates.

(c)

Large weights and biases used in the gradient descent optimisation algorithm can lead to premature convergence. This occurs because the large weights and biases move to the asymptoic ends of the sigmoid activation function too quickly, which leads to extreme output values with associated derivatives being close to zero, meaning the weight updates are also close to zero. Absolute values of the weights and biases is also a poor strategy as the active domain of the sigmoid activation function is $[-\sqrt{3}, \sqrt{3}]$. If the weights and biases are initialised as only large positive values, the activations of the sigmoid activation function will be biased towards the positive end of the sigmoid's active domain.

Question 6

The momentum term is essential to the stochastic gradient descent (SGD) optimisation algorithm, as it smooths out the search trajectories and prevents the oscillation of the search trajectories. The idea of the momentum term is to average out the weigth changes, thereby ensuring that the search path is in the average downhill direction, meaning the search direction does not prematurely change between epochs, ensuring that oscillation between search directions does not occur. The momentum term is then simply the previous weight change weighted by a scalar value α, which is defined as any value between the range of $\alpha \in (0, 1)$. For larger values of α, the more strict the optimisation algorithm becomes of staying on the current search path, meaning that more significant evidence is needed to jump over to the other side.

Question 7

This statement is false, because if the net input signal is calculated by use of product units, the single neuron can separate non-linearly separable classes. The mathmatical equation used to calculate the product units is as follows.

$$net = \prod_{i=1}^{I} x_i^{w_i} \tag{4}$$

where net is the net input signal, I is the number of input signals, x_i is the i-th signal and w_i is the weight corresponding to the i-th input signal. Product units allow for higher-order combinations of inputs, having the advantage of increased information capacity. This increased information capacity and the product units allows the single neuron to separate non-linearly separable data.

Question 8

The main advantages of using the scaled conjugate gradient optimisation algorithm instead of the gradient descent optimisation algorithm is that the model converges faster due to the fast quadratic convergence of Newton's method. Additionally, the scaled conjugate gradient optimisation algorithm is less susceptible to the local minima as the algorithm restarts every n_w steps if there is no reduction in the test error, where n_w is the total number of weights and biases. Lastly, the scaled conjugate gradient optimisation algorithm performs automatic step size adjustment, eliminating the need for the learning rate used in the gradient descent optimisation algorithm, meaning there is less manual control parameters of which the optimal values should be found for.

Question 9

The first main advantages of using population-based meta-heuristics to train neural networks is that the algorithm is less susceptible to getting trapped in the local minima, meaning that the population-based approach can better explore the entire search space [3].

Another main advantage is that the population-based meta-heuristics does not require the caluclation of any gradients or derivatives, which makes the construction or fine-tuning of the neural network model less computationally intensive per iteration compared to backpropagation.

The last main advantage is that the population-based meta-heuristics algorithms is more robust to noise and missing values in the training data, than the classical optimisation techniques such as gradient descent.

Question 10

Active learning in neural networks is driven by the need to enhance training efficiency and improve model generalisation. Active learning is any form of learning in which the learning algorithm has some control over what part of the input space it receives information from, to ensure that the most optimal information is used when training the neural network model.

The first rationale behind active learning is to efficiently use the data. This rationale is based on active learning that allows the neural network to select the most informative examples in the dataset for training. Less data is then used to construct the model, which leads to faster convergence and training time.

The second rationale behind active learning is to improve the model's generalisation ability to unseen data. This is achieved by by avoiding redundant information and selecting examples that contain enough information to learn a task. By concentrating on diverse and informative samples, the model becomes more robust, enhancing its performance on new, unseen inputs.

Question 11

A linear logistic regression model can not separate the two classes. However a non-linear logistic regression model will be able to separate the two classes.

The non-linear logistic regression model can capture the underlying relationship between the two descriptive features x_1 and x_2 if they are transformed by use of sine and cosine as basis functions. Therefore the non-linear logistic regression model would become the following.

$$P(y=1|\boldsymbol{x}_i) = \frac{1}{1+e^{-(w_0+w_1\cdot\sin(x_{i,1})+w_2\cdot\cos(x_{i,1})+w_3\cdot\sin(x_{i,2})+w_4\cdot\cos(x_{i,2}))}} \tag{5}$$

where $\boldsymbol{x}_i$ is a vector containing all of the descriptive features of the i-th observation, w_0 is the bias and $P(y=1|\boldsymbol{x}_i)$ is the probability of the i-th observation bolonging to class 1.

Therefore by utilising these basis functions to capture the relationship between the two descriptive features, will the non-linear logistic regression be able to separate these classes.

Question 12

Consider the classification problem depicted in the figure below. Explain in detail how logistic regression can be used to separate the classes.

Logistic regression can separate these classes, by making use of a multinomial logistic regression model, which is designed as an extension of the normal logistic regression model to handle multiclass classification problems.

The multinomial logistic regression model is constructed by making use of 5 one versus all logistic regression models for this scenario. These one-versus-all models distinguishes between one class of the target feature and all the other classes of the target feature. The 5 one-versus-all models would then be the following:

- Logistic regression model 1: classify class 0 against the other classes
- Logistic regression model 2: classify class 1 against the other classes
- Logistic regression model 3: classify class 2 against the other classes
- Logistic regression model 4: classify class 3 against the other classes
- Logistic regression model 5: classify class 4 against the other classes

This multinomial logistic regression model will then construct 4 decision boundaries combined from each models decision boundary to separated these classes into 5 segments, where each segment contains all the observations of one class.

Question 13

Neural networks that use bounded activation functions show limited robustness to variations in input feature scale. While scaling input features to the same range is not necessary, it is recommended to do so to improve model performance and generalisation. By scaling inputs to align with the activation function's active domain, the network can avoid regions where gradients are near zero, leading to vanishing gradients, which helps facilitate learning and reduces the risk of poor model performance due to the input features having the same range.

Question 14

Larger gradients will be created when the control parameter that controls the steepness of the activation function is increased. This can help to mitigate the vanishing gradient problem and more complex relationships between descriptive features can be learned, as neurons become more sensitive to important distinctions. Additionally, the model also converges and trains faster, as the neurons in the hidden layer better identify the most informative features for the classification task at hand.

Question 15

A neural network that deploys a sigmoid activation function as the activation function in the hidden layer, as opposed to using the linear activation function as the activation function in the hidden layer, will result in less hidden units for a highly non-linear mapping. This is because the linear activation function only perform linear transformations, which limits the model's capacity to capture non-linear patterns and more units in the hidden layer would have to be added for the model to capture these non-linear patterns. Conversly, the sigmoid activation function allows the hidden layer to capture non-linear patterns, which will result in less units that should be used in the hidden layer to achieve the same generalisation performance as the linear activation function deployed in the hidden layer with more hidden units.

Section E: Unsupervised Learning

Question 1

a, d

Question 2

TODO

Question 3

The first approach that can be used is by making use of batch maps. A batch map aims to reduce the number of weight updates by only updating weight values after all patterns have been presented. This helps to negate the effect of the stochastic training of the SOM, ensuring faster convergence speed and faster training.

The second approach that can be used is by modifying the neighbourhood function. Applying the Gaussian function ensures that every codebook vector will be updates in each iteration. This can be computationally expensive and use a lot of memory. By using a clipped Gaussian function as the neighbourhood function, the number of updates of the codebook vectors is reduced and ensures faster convergence speed and that the SOM uses less memory. A dynamic width is then also applied to the neighbourhood function with the equation of

$$\sigma(t) = \sigma(0)e^{-t \div \tau_1}$$

where τ_1 is a positive constant and $\sigma(0)$ is the initial large variance. This will ensure faster convergence and training speeds.

The last approach that can be used is by making use of a shortcut technique to determine the winner. Once each instance in the trainig data is assigned to a winning neuron, in the next iteration of the SOM, we search for the winning neuron of a data instance in the neighbourhood of the previous winning neuron and if the new winning neuron is not on the edge of the neighbourhood in which searching was performed, the search stops. This leads to faster convergence and training times.

Question 4

TODO

Question 5

The SOM is robust to missing values as it simply ignores the features that contains the missing values in the distance calculation. Therefore, a winning neuron can be found for the instance that contains the missing values. The missing value can then be replaced by either the value of the corresponding value of the codebook vector of the winning neuron, the average of the neighbouring values, the average of the entire cluster, or the neighbourhood function weighted average.

Question 6

The goal of unsupervised learning is to discover patterns, relationships or structures in the input data without the need for labeled outputs or target values. Instead of aiming to predict a specific outcome, unsupervised learning algorithms seek to organise or transform the data to uncover hidden structures in the data.

Question 7

(a)

TODO

(b)

TODO

(c)

i TODO

ii TODO

iii TODO

Section F: Kernel-based Learning

Question 1

The inductive bias of linear support vector machines is that the alogorithm assumes the data can be linear separable by using a straight hyperplane.

Question 2

The maximum margin approach of support vector machine is based on the principle that the best decision boundary between classes is the one that maximises the distance from the boundary to the nearest points in each class. The main rationale behind this is that the maximum margin approach of support vector machine improves generalisation of the algorithm on unseen data, the SVM becomes more robust to outliers.

Therefore, the main rationale behind the maximum margin approach of support vector machine is to vind a set of support vectors that are the most influencial instances, to construct an optimal hyperplane between the two classes to ensure for the best possible generalisation performance and creates a SVM algorithm that is more robust to outliers.

Question 3

The consequence of noise in the linear SVM algorithm is that the maximum margin between the two classes will become more narrow. It is not possible for a linear SVM to separate the classes with one linear hyperplane. As the linear SVMs try to find the maximum distance between the two classes, if there is noise present in the data, this maximum distance would be very small, which could then lead to poor generalisation and outliers having a higher influence on the data.

Question 4

Soft margin hyperplanes with slack variables are introduced to ensure that linear SVMs can cope with noise. Soft margin hyperplanes allows for errors to be made by the linear SVM algorithm to ensure the linear SVM is more robust to noise. The number of errors which the linear SVM algorithm can make is specified by the slack variable. The slack variable is used in conjunction with a penalty function to ensure that the linear SVM can make mistakes. The higher the value of the slack variable, the more errors can be made by the linear SVM, which leads to a linear SVM algorithm that is more robust to noise. The new objective of the linear SVM then becomes the minimisation of the sum of squared errors with the addition of the penalty term and then to maximise the margin.

Question 5

The advantages to this approach is that the linear SVM algorithm becomes more robust to noise, improves generalisation and allows for a larger margin between classes. This is due to the penalty term that allows the algorithm to make a few mistakes to ensure that noise has little to no affect when the margin is constructed. When this margin is then constructed, the noisy instances are ignored, allowing for a larger distance margin, as the distance between the two nearest points of the two classes becomes larger. Therefore, the generalisation of the algorithm improves for unseen data.

The disadvantages to this approach is that the algorithm adds an additional penalty term to objective function, which increases complexity and creates longer training times. The performance of the model can be sensitive to the choice of the slack variable. Therefore, another disadvantage is that the slack variable is problem depended, which means an optimal value of the slack value should be found for each problem, which could be computationally and resource intensive. If the slack variable is too large of a value, then too much errors can be made, which could lead to a SVM model that has too large of a margin, and if the slack variable is too small, the model becomes sensitive to any noise in the dataset.

Question 6

TODO

Question 7

TODO

Question 8

TODO

Question 9

TODO

Question 10

TODO

Section G: Ensemble Learning

Question 1

b, d

Question 2

The main rationale behind ensemble learning is to combine multiple learners or models that each provide slightly different results on the same dataset. Since different models can perform differently on various data patterns, aggregating the knowledge of all of the learners will result in a better overall performance compared to a single model. Although, this is only true if the models used in the ensemble are all different in their predictive behavior.

Question 3

No single machine learning algorithm is universally the most accurate due to the influence of its inductive bias. While combining experts of the same type, known as a mixture of homogeneous experts, can improve accuracy by reducing the adverse effects of this bias, the overall predictive capability remains constrained by the inductive bias of the individual algorithm used in the mixture.

Conversely, a combination of different machine learning algorithms, which is referred to as a mixture of heterogeneous experts, is used to take advantage of each of the strenghts, and to reduce the adverse effects of the inductive biases of each algorithm used in the ensemble.

As a result, heterogeneous mixtures of experts are expected to perform better than homogeneous mixtures of experts, as the heterogeneous model has the ability to combine different inductive biases that allows the ensemble to make more informed predictions.

Question 4

AdaBoost is an ensemble that consists of many weak learners all induced to underfit the data. All of these weak learners are then assigned a voting weight in order to create an accurate ensemble model. This weight is calculated by making use of the training error-rate of the model. The voting weight is then calculated by use of the following equation.

$$\alpha_t = 0.5 \times \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{6}$$

where ϵ_t and α_t is the error-rate and the voting weight of the t-th model in the ensemble respectively.

As seen in equation 6, the higher the error-rate, the lower the voting weight of the model will be and the lower the error-rate, the higher the voting weight of the model will be. The final decision or prediction of the ensemble is then the sign of the following equation.

$$\hat{y} = sign\left(\sum_{t1}^{T} \alpha_t h_t(x)\right) \tag{7}$$

where $\hat{y}$ is the final prediction of the ensemble, T is the number of models in the ensemble and $h_t(x)$ is the t-th predicted label of the t-th model in the ensemble. As seen in equation 7, the lower the voting weight is, the smaller the influence the model has on the final prediction. Therefore the weighted vote is impacted by how well the model performs.

Since each model is designed to underfit the data, some may focus on features with minimal predictive power or those containing significant noise. By incorporating voting weights, AdaBoost reduces the impact of these less effective models on the overall ensemble, allowing the more accurate models to dominate the decision-making process. This weighted voting mechanism enhances the model's robustness and accuracy, ultimately leading to better generalisation on unseen data.

Question 5

The k-nearest neighbours algorithm is very dependend on the value of k and there is no universally correct value of what the value of k should be. Observations in the data may be classified correctly for one value of k and incorrectly for a different value of k. This problem is then addressed by creating an ensemble that uses different values of k, which makes the overall performance of the ensemble model more robust and allows for the model to generalise better to unseen data.

Different values of k can also address the bias-variance dilemma. Small values of k leads to a low bias, but a high variance, which makes the model sensitive to any noise or outliers in the dataset of any small changes in the dataset. Conversely, a larger value for k leads to low variance, but an increase in bias, which results in smoother decision boundaries. The ensemble of k-NN models, will achieve a greater balance between bias and variance if the k-NN models in the ensemble contains different values of k. This improves the model's overall performance and reliability.

Question 6

Each tree in the random forest model is constructed from a random bootstrap sample, also known as sampling with replacement, from the original dataset. Within each of these bootstrap samples, a random subset of features, with a fixed size, is selected to account for more variability between each dataset. For each of these randomly sampled bootstrap samples and subset of features from the original training data, a decision tree is indeuced.

This process ensures that each decision tree in the random forest is trained on a unique combination of observations and features, resulting in diverse trees that capture different patterns.

Question 7

Regression trees can be either sensitive or robust to outliers depending on which statistic is used to form a final prediction of the regression tree.

If the mean of all output features from each tree in the random forests is used as the statistic to calculate the prediction of the ensemble, the random forest is sensitive to outliers. The reasoning behind this is because the mean is directly influenced by extreme values. An outlier can skew the mean pulling it away from the central dendency of the true distribution, which makes the regression random forest sensitive to outliers.

Conversely, if the median is used as the statistic to calculate the final output of the regression random forest model, then will the regression random forest model becomes robust to outliers. This is because the median is less affected by extreme values, whicch offers a more stable and robust prediction that better reflects the true central tendency of the data in the presence of outliers.

Question 8

The adaptive boosting algorithm is robust to outliers. The algorithm is robust to imbalanced classes as the model focuses more on incorrect predictions by lowering the weight of the associated correct predicted observations and increasing the weight of the associated incorrectly predicted observations. Therefore, if instances from the minority class are incorrectly classified as the majority class, the associated weights of these observations would be increased, placing more emphasis on these observations for the next decision stump to investigate. The weights of all incorrect predicted observations keeps on increasing if the predictions are incorrect, meaning that for even extreme inbalanced classes, the model would keep on increasing the weights of the incorrectly classsified minority class, until these observations are correctly classsified.

Question 9

Bagging result in reduced variance as the models in the ensmble are trained on a smaller subset of the original dataset. If random observations are sampled through the bootstrap sampling technique, then each model in the ensemble is trained on different subsets of the data, which leads to variations in the predictions of each model.

If subsets of randomly selected features are used to fit the models in the ensemble, the models also becomes less prone to overfitting, thus reducing the variance of the model. The variation between each dataset allows models to focus more on a subset of features from the original dataset, allowing for more variation in the predictions of the model.

By using the average or median in regression ensembles and majority voting in classification ensembles to create a final prediction of the ensemble, fluctuations and errors made by each individual mdoel is smoothed out, leading to more stable and robust predictions.

Question 10

By combining a number of weak learners, boosting can incrementally improve the performance of the ensemble, by targeting incorrect predictions made by previous learners. This iterative process allows that the ensemble to build a strong model through aggregation of multiple weak learners. The weak learners are also easy to interpret and has a very fast training speed. Overfitting is also reduced as the weak learners are trained to underfit the data, by only capturing basic patterns in the data. The ensemble maintains a level of generalisation ability by training these weak models to underfit the data, which might have been lost if more complex models were used.

Section H: Reinforcement Learning

Question 1

TODO

Question 2

(a)

TODO

(b)

TODO

(c)

TODO

(d)

TODO

(e)

TODO

References

[1] G. James, D. Witten, T. Hastie, R. Tibshirani *et al.*, *An introduction to statistical learning.* Springer, 2013, vol. 112.

[2] D. Kim. (2023) What is the inductive bias in machine learning? Accessed: 2024-10-29. [Online]. Available: https://medium.com/@chkim345/what-is-the-inductive-bias-in-machine-learning-212a5f53e9aa

[3] “Metaheuristics,” accessed: 2024-10-29. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/metaheuristics