Machine Learning Exam Answer Sheet

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

October 31, 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.

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

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.

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

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)

Ditsribution b and e may indicate features with outliers.

Distribution b could contain outliers, as the distribution is heavily skewed to the right. The majority of the values are between $[0 - 10]$, with a small amount of values that reaches $(25 - 40)$. These values could indicate extreme values according as they are far away from the majority of the statistics in the distribution, such as the mean, median and even the third quartile.

Distribution e could also contain outliers, as the majority of the values are clustered around lower ranges, but a few of these values might extend to much higher values.

(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)

Features b, c, d and e will not be sensible to impute missing values with the mean.

Feature b might contain outliers, and imputing values for a distribution that includes outliers could skew the central tendency. This skewing may lead to unreliable and inaccurate imputed values. Additionally, the distribution of Feature B is skewed to the right, with the median around 0. Imputing missing values using the mean would fail to accurately reflect the central tendency of the feature.

Feature c is a bimodal distribution, which indicates that there are two separate groups or processes within the data. Therefore, to impute missing values with the mean of this feature is not ideal, as the mean may not accurately represent either of the underlying distributions.

Feature d is a unique feature, that represents identification information, where mean values have no meaningful interpretation.

Feature e might contain outliers, and imputing values for a distribution that includes outliers could skew the central tendency. This skewing may lead to unreliable and inaccurate imputed values. Additionally, the distribution of Feature E is slightly skewed to the right, with the median around 40000. Imputing missing values using the mean would fail to accurately reflect the central tendency of the feature.

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.

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

then the following can be calculated for each class:

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

the calculation of the entropy of the dataset is as folows:

$$\begin{aligned} H(D) &= -\sum_{m=1}^{7} p(y_m)\log_7\left(p(y_m)\right) \\ H(D) &= -\big[-0.1335776952 + (-0.1335776952) + (-0.1335776952) + (-0.1335776952) \\ &\qquad + (-0.1781035936) + (-0.1335776952) + (-0.1335776952)\big] \\ H(D) &= -(-0.9795697645) \\ H(D) &= 0.9795697645 \\ H(D) &\approx 0.97957 \end{aligned}$$

(b)

Firstly, the equation used to calculate the entropy due to the split is

$$H_x(D) = \sum_{s=1}^{S} p_s H(D_s)$$

where the probability of outcome s is

$$p_s = \frac{|D_s|}{|D|}$$

and $|D_s|$ is the subset containing all patters associated with outcome s. The information gain is then calculated as follow:

$$\text{gain}(x) = H(D) - H_x(D)$$

The calculation of the question is then as follows.

$$\text{odd set} = [3, 5, 7]$$
$$\text{even set} = [2, 4, 6, 6, 8]$$

The probabilities of the observations in the odd set is:

$$p_o(3) = \frac{1}{3}$$
$$p_o(5) = \frac{1}{3}$$
$$p_o(7) = \frac{1}{3}$$

then the following can be calculated for each class in the odd set:

$$p_o(3)log_7\left(p_o(3)\right) = \frac{1}{3}log_7\left(\frac{1}{3}\right) = -0.188191678$$
$$p_o(5)log_7\left(p_o(5)\right) = \frac{1}{3}log_7\left(\frac{1}{3}\right) = -0.188191678$$
$$p_o(7)log_7\left(p_o(7)\right) = \frac{1}{3}log_7\left(\frac{1}{3}\right) = -0.188191678$$

The calculation of the entropy of the odd set is as follows:

$$H(D_{odd}) = -\sum_{m=1}^{3} p_o(y_m)\log_7\left(p_o(y_m)\right)$$
$$H(D_{odd}) = -\big[-0.188191678 + (-0.188191678) + (-0.1881916782)\big]$$
$$H(D_{odd}) = -(-0.5645750341)$$
$$H(D_{odd}) = 0.5645750341$$

The probabilities of the observations in the even set is:

$$p_e(2) = \frac{1}{5}$$
$$p_e(4) = \frac{1}{5}$$
$$p_e(6) = \frac{2}{5}$$
$$p_e(8) = \frac{1}{5}$$

then the following can be calculated for each class in the even set:

$$\begin{aligned}
p_{even}(2) log_7\left(p_e(2)\right) &= \frac{1}{5} log_7\left(\frac{1}{5}\right) = -0.1654174951 \\
p_{even}(4) log_7\left(p_e(4)\right) &= \frac{1}{5} log_7\left(\frac{1}{5}\right) = -0.1654174951 \\
p_{even}(6) log_7\left(p_e(6)\right) &= \frac{2}{5} log_7\left(\frac{2}{5}\right) = -0.1883521153 \\
p_{even}(8) log_7\left(p_e(8)\right) &= \frac{1}{5} log_7\left(\frac{1}{5}\right) = -0.1654174951
\end{aligned}$$

The calculation of the entropy of the even set is as follows:

$$\begin{aligned}
&H(D_{even}) = -\sum_{m=1}^{4} p_e(y_m)\log_7\left(p_e(y_m)\right) \\
&H(D_{even}) = -\big[-0.1654174951 + (-0.1654174951) + (-0.1883521153) + (-0.1654174951)\big] \\
&H(D_{even}) = -(-0.6846046005) \\
&H(D_{even}) = 0.6846046005
\end{aligned}$$

The entropy due to the split is then calculated with the following equation:

$$H_x(D) = p_{odd} \times H(D_{odd}) + p_{even} \times H(D_{even})$$

where:

$$\begin{aligned}
p_{odd} &= \frac{|D_{odd}|}{|D|} = \frac{3}{8} \\
p_{even} &= \frac{|D_{even}|}{|D|} = \frac{5}{8}
\end{aligned}$$

The entropy due to the split is then calculated as:

$$\begin{aligned}
H_x(D) &= \frac{3}{8} \times 0.5645750341 + \frac{5}{8} \times 0.6846046005 \\
H_x(D) &= 0.2117156378 + 0.4278778753 \\
H_x(D) &= 0.6395935131
\end{aligned}$$

The information gain is then calculated as:

$$\begin{aligned}
\text{gain}(x) &= 0.9795697645 - 0.6395935131 \\
\text{gain}(x) &= 0.339762514 \\
\text{gain}(x) &\approx 0.33976
\end{aligned}$$

Question 3

The probabilities of the categories are as follow:

$$\begin{aligned}
p(\text{Cinema}) &= \frac{6}{10}\\
p(\text{Tennis}) &= \frac{2}{10}\\
p(\text{Shopping}) &= \frac{1}{10}\\
p(\text{Stay in}) &= \frac{1}{10}
\end{aligned}$$

and the following is calculated: then the following can be calculated for each observation:

$$\begin{aligned}
p(\text{Cinema}) log_4\left(p(\text{Cinema})\right) &= \frac{6}{10} log_4\left(\frac{6}{10}\right) = -0.221896782\\
p(\text{Tennis}) log_4\left(p(\text{Tennis})\right) &= \frac{2}{10} log_4\left(\frac{2}{10}\right) = -0.2321928095\\
p(\text{Shopping}) log_4\left(p(\text{Shopping})\right) &= \frac{1}{10} log_4\left(\frac{1}{10}\right) = -0.166094047\\
p(\text{Stay in}) log_4\left(p(\text{Stay in})\right) &= \frac{1}{10} log_4\left(\frac{1}{10}\right) = -0.166094047
\end{aligned}$$

the calculation of the entropy of the dataset is as folows:

$$\begin{aligned}
H(D) &= -\sum_{m=1}^{4} p(y_m)\log_4\left(p(y_m)\right)\\
H(D) &= -\big[-0.221896782 + (-0.2321928095) + (-0.166094047) + (-0.166094047)\big]\\
H(D) &= -(-0.7854752972)\\
H(D) &= 0.7854752972\\
H(D) &\approx 0.78548
\end{aligned}$$

The Decision distributions across the Weather feature are represented in the table below. The Decision distributions across the Parents feature are represented in the table below.

Decision	**Sunny**	**Rainy**	**Windy**
Cinema: p(C)	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{3}{4}$
Tennis: p(T)	$\frac{2}{3}$	0	0
Stay in: p(SI)	0	$\frac{1}{3}$	0
Shopping: p(S)	0	0	$\frac{1}{4}$

Table 1: Probability of each class occuring in the Weather feature set

Decision	**Yes**	**No**
Cinema: p(C)	1	$\frac{1}{5}$
Tennis: p(T)	0	$\frac{2}{5}$
Stay in: p(SI)	0	$\frac{1}{5}$
Shopping: p(S)	0	$\frac{1}{5}$

Table 2: Probability of each class occuring in the Parents feature set

The Decision distributions across the Money feature are represented in the table below.

Decision	**Rich**	**Poor**
Cinema: p(C)	$\frac{3}{7}$	1
Tennis: p(T)	$\frac{2}{7}$	0
Stay in: p(SI)	$\frac{1}{7}$	0
Shopping: p(S)	$\frac{1}{7}$	0

Table 3: Probability of each class occuring in the Money feature set

Calculating the information gain of the Weather feature. then the following can be calculated for each class in the weather set associated with sunny and the probabilities are represented in Table 1:

$$\begin{aligned}
p(\mathrm{C})log_4\left(p(\mathrm{C})\right) &= \frac{1}{3}log_4\left(\frac{1}{3}\right) = -0.2641604168\\
p(\mathrm{T})log_4\left(p(\mathrm{T})\right) &= \frac{2}{3}log_4\left(\frac{2}{3}\right) = -0.1949875002\\
p(\mathrm{S})log_4\left(p(\mathrm{S})\right) &= 0 \times log_4\left(0\right) = 0\\
p(\mathrm{SI})log_4\left(p(\mathrm{SI})\right) &= 0 \times log_4\left(0\right) = 0
\end{aligned}$$

The calculation of the entropy of the sunny set is as follows:

$$\begin{aligned}
H(D_{sunny}) &= -\sum_{m=1}^{4} p(y_m)\log_4\left(p(y_m)\right)\\
H(D_{sunny}) &= -\big[-0.2641604168 + (-0.1949875002) + 0 + 0\big]\\
H(D_{sunny}) &= -(-0.459147917)\\
H(D_{sunny}) &= 0.459147917
\end{aligned}$$

then the following can be calculated for each class in the weather set associated with rainy and the probabilities are represented in Table 1:

$$\begin{aligned}
p(\mathrm{C})log_4\left(p(\mathrm{C})\right) &= \frac{1}{3}log_4\left(\frac{1}{3}\right) = -0.1949875002\\
p(\mathrm{T})log_4\left(p(\mathrm{T})\right) &= \frac{2}{3}log_4\left(\frac{2}{3}\right) = 0\\
p(\mathrm{S})log_4\left(p(\mathrm{S})\right) &= 0 \times log_4\left(0\right) = -0.2641604168\\
p(\mathrm{SI})log_4\left(p(\mathrm{SI})\right) &= 0 \times log_4\left(0\right) = 0
\end{aligned}$$

The calculation of the entropy of the rainy set is as follows:

$$
\begin{aligned}
H(D_{rainy}) &= -\sum_{m=1}^{4} p(y_m)\log_4\left(p(y_m)\right) \\
H(D_{rainy}) &= -\big[-0.1949875002 + 0 + (-0.2641604168) + 0\big] \\
H(D_{rainy}) &= -(-0.459147917) \\
H(D_{rainy}) &= 0.459147917
\end{aligned}
$$

then the following can be calculated for each class in the weather set associated with Windy and the probabilities are represented in Table 1:

$$
\begin{aligned}
p(\mathrm{C})log_4\left(p(\mathrm{C})\right) &= \frac{1}{3}log_4\left(\frac{1}{3}\right) = -0.1556390622 \\
p(\mathrm{T})log_4\left(p(\mathrm{T})\right) &= \frac{2}{3}log_4\left(\frac{2}{3}\right) = 0 \\
p(\mathrm{S})log_4\left(p(\mathrm{S})\right) &= 0 \times log_4\left(0\right) = 0 \\
p(\mathrm{SI})log_4\left(p(\mathrm{SI})\right) &= 0 \times log_4\left(0\right) = -0.25
\end{aligned}
$$

The calculation of the entropy of the windy set is as follows:

$$
\begin{aligned}
H(D_{windy}) &= -\sum_{m=1}^{4} p(y_m)\log_4\left(p(y_m)\right) \\
H(D_{windy}) &= -\big[-0.1556390622 + 0 + 0 + (-0.25)\big] \\
H(D_{windy}) &= -(-0.4056390622) \\
H(D_{windy}) &= 0.4056390622
\end{aligned}
$$

The entropy due to the split is then calculated with the following equation:

$$
H_{weather}(D) = p_{sunny} \times H(D_{sunny}) + p_{rainy} \times H(D_{rainy}) + p_{windy} \times H(D_{windy})
$$

where:

$$
\begin{aligned}
p_{sunny} &= \frac{|D_{sunny}|}{|D|} = \frac{3}{10} = 0.3 \\
p_{rainy} &= \frac{|D_{rainy}|}{|D|} = \frac{3}{10} = 0.3 \\
p_{windy} &= \frac{|D_{windy}|}{|D|} = \frac{4}{10} = 0.4
\end{aligned}
$$

The entropy due to the split is then calculated as:

$$
\begin{aligned}
H_{weather}(D) &= 0.3 \times 0.459147917 + 0.3 \times 0.459147917 + 0.4 \times 0.4056390622 \\
H_{weather}(D) &= 0.1377443751 + 0.1377443751 + 0.1622556249 \\
H_{weather}(D) &= 0.4377443751
\end{aligned}
$$

The information gain is then calculated as:

$$\begin{aligned}\text{gain}(weather) &= 0.7854752972 - 0.4377443751\\ \text{gain}(weather) &= 0.3477309221\\ \text{gain}(weather) &\approx 0.34773\end{aligned}$$

Calculating the information gain of the Parents feature. then the following can be calculated for each class in the parents set associated with "yes" and the probabilities are represented in Table 2:

$$\begin{aligned}p(\text{C})log_4\left(p(\text{C})\right) &= 1log_4\left(1\right) = 0\\ p(\text{T})log_4\left(p(\text{T})\right) &= 0 \times log_4\left(0\right) = 0\\ p(\text{S})log_4\left(p(\text{S})\right) &= 0 \times log_4\left(0\right) = 0\\ p(\text{SI})log_4\left(p(\text{SI})\right) &= 0 \times log_4\left(0\right) = 0\end{aligned}$$

The calculation of the entropy of the "yes" set is as follows:

$$\begin{aligned}H(D_{yes}) &= -\sum_{m=1}^{4} p(y_m)\log_4\left(p(y_m)\right)\\ H(D_{yes}) &= -\big[0+0+0+0\big]\\ H(D_{yes}) &= 0\end{aligned}$$

then the following can be calculated for each class in the parents set associated with "no" and the probabilities are represented in Table 2:

$$\begin{aligned}p(\text{C})log_4\left(p(\text{C})\right) &= \frac{1}{5}log_4\left(\frac{1}{5}\right) = -0.2321928095\\ p(\text{T})log_4\left(p(\text{T})\right) &= \frac{2}{5}log_4\left(\frac{2}{5}\right) = -0.264385619\\ p(\text{S})log_4\left(p(\text{S})\right) &= \frac{1}{5}log_4\left(\frac{1}{5}\right) = -0.2321928095\\ p(\text{SI})log_4\left(p(\text{SI})\right) &= \frac{1}{5}log_4\left(\frac{1}{5}\right) = -0.2321928095\end{aligned}$$

The calculation of the entropy of the "no" set is as follows:

$$\begin{aligned}H(D_{no}) &= -\sum_{m=1}^{4} p(y_m)\log_4\left(p(y_m)\right)\\ H(D_{no}) &= -\big[-0.2321928095 + (-0.264385619) + (-0.2321928095) + (-0.2321928095)\big]\\ H(D_{no}) &= -(-0.9609640474)\\ H(D_{no}) &= 0.9609640474\end{aligned}$$

The entropy due to the split is then calculated with the following equation:

$$H_{parents}(D) = p_{yes} \times H(D_{yes}) + p_{no} \times H(D_{no})$$

where:

$$p_{yes} = \frac{|D_{yes}|}{|D|} = \frac{5}{10} = 0.5$$
$$p_{no} = \frac{|D_{no}|}{|D|} = \frac{5}{10} = 0.5$$

The entropy due to the split is then calculated as:

$$H_{parents}(D) = 0.5 \times 0 + 0.5 \times 0.9609640474 H_{parents}(D) \qquad = 0 + 0.4804820237$$
$$H_{parents}(D) = 0.4804820237$$

The information gain is then calculated as:

$$\text{gain}(parents) = 0.7854752972 - 0.4804820237$$
$$\text{gain}(parents) = 0.3049932735$$
$$\text{gain}(parents) \approx 0.30499$$

Calculating the information gain of the Money feature. then the following can be calculated for each class in the money set associated with rich and the probabilities are represented in Table 3:

$$p(\text{C})log_4\left(p(\text{C})\right) = \frac{3}{7}log_4\left(\frac{3}{7}\right) = -0.2619412331$$
$$p(\text{T})log_4\left(p(\text{T})\right) = \frac{2}{7} \times log_4\left(\frac{2}{7}\right) = -0.2581935603$$
$$p(\text{S})log_4\left(p(\text{S})\right) = \frac{1}{7} \times log_4\left(\frac{1}{7}\right) = -0.2005253516$$
$$p(\text{SI})log_4\left(p(\text{SI})\right) = \frac{1}{7} \times log_4\left(\frac{1}{7}\right) = -0.2005253516$$

The calculation of the entropy of the rich set is as follows:

$$H(D_{rich}) = -\sum_{m=1}^{4} p(y_m)\log_4\left(p(y_m)\right)$$
$$H(D_{rich}) = -\big[-0.2619412331 + (-0.2581935603) + (-0.2005253516) + (-0.2005253516)\big]$$
$$H(D_{rich}) = -(-0.9211854966)$$
$$H(D_{rich}) = 0.9211854966$$

then the following can be calculated for each class in the money set associated with poor and the probabilities are represented in Table 3:

$$p(\text{C})log_4\left(p(\text{C})\right) = \frac{5}{5}log_4\left(\frac{5}{5}\right) = 0$$
$$p(\text{T})log_4\left(p(\text{T})\right) = 0 \times log_4\left(0\right) = 0$$
$$p(\text{S})log_4\left(p(\text{S})\right) = 0 \times log_4\left(0\right) = 0$$
$$p(\text{SI})log_4\left(p(\text{SI})\right) = 0 \times log_4\left(0\right) = 0$$

The calculation of the entropy of the poor set is as follows:

$$\begin{aligned}
H(D_{poor}) &= -\sum_{m=1}^{4} p(y_m)\log_4\left(p(y_m)\right) \\
H(D_{poor}) &= -\big[0+0+0+0\big] \\
H(D_{poor}) &= 0
\end{aligned}$$

The entropy due to the split is then calculated with the following equation:

$$H_{money}(D) = p_{rich} \times H(D_{rich}) + p_{poor} \times H(D_{poor})$$

where:

$$\begin{aligned}
p_{rich} &= \frac{|D_{rich}|}{|D|} = \frac{7}{10} = 0.7 \\
p_{poor} &= \frac{|D_{poor}|}{|D|} = \frac{3}{10} = 0.3
\end{aligned}$$

The entropy due to the split is then calculated as:

$$\begin{aligned}
H_{money}(D) = 0.7 \times 0.9211854966 + 0.3 \times 0 H_{money}(D) && = 0.6448298476 + 0 \\
H_{money}(D) = 0.6448298476
\end{aligned}$$

The information gain is then calculated as:

$$\begin{aligned}
\text{gain}(\mathit{money}) &= 0.7854752972 - 0.6448298476 \\
\text{gain}(\mathit{money}) &= 0.1406454496 \\
\text{gain}(\mathit{money}) &\approx 0.14065
\end{aligned}$$

Therefore, when considering all of the calculated information gain values of gain(*weather*) $\approx$ 0.34773, gain(*parents*) $\approx$ 0.30499 and gain(*money*) $\approx$ 0.14065, the highest information gain is from the weather feature. Therefore, the best split at the root of the tree is the weather feature.

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)

The robustness of the regression tree in this scenario is influenced by the value of the threshold. If the value of the threshold is small, then the tree will not be robust to outliers. The small threshold leads to larger trees that overfits on the training data with less instances in the leave nodes at the bottom of the tree. Consequently, outliers can heavily affect the calculation of the predicted value in the leaf nodes that contain a limited number of instances. Therefore, making the tree sensitive to outliers. If a technique of pruning is used to ensure that the tree is less complex, can enhance the models's robustness to outliers as more instances are used to average out the effect of the outlier. Furthermore, if the median is used as the statistic for predictions, the pruned regression tree would be even more robust to outliers, as the median is less affected by extreme values and would not skew the predicted value.

In contrast, if a large value is used for the threshold, then the regression tree would be more robust to outliers. This is because the leaf nodes at the bottom of the tree would have more instances that can be used in the averaging process, which helps mitigate the influence of outliers over the mean calculation. If the median is used as the statistic, then the pruned regression tree would be even more robust to the outlier as the outlier would have no effect on the predicted value.

However, if the threshold is set too high, the tree may underfit the data. In this case, there would be fewer splits in the tree, resulting in leaf nodes with a large number of instances. While this could reduce sensitivity to outliers, it may also lead to unreliable predictions, as the model might fail to capture important trends and patterns in the data.

(b)

The robustness of the regression tree in this scenario is influenced by the value of the threshold. In the scenario where the threshold is small, the regression tree is sensitive to outliers. This is because the tree would result in an overfitted tree, where the mean squared error is largly affected by the outliers in the target feature. As a result, the leaf nodes at the bottom of the tree may contain only a small subset of observations, allowing outliers to heavily influence the predicted values in those nodes.

In contrast, if a large value is used for the threshold, then the regression tree would be more robust to outliers. Larger values of the mean squared error caused by the outliers would not always result in a split, allowing for the leaf nodes at the bottom of the tree to conatin more instances. This increase in instances helps mitigate the influence of outliers on the mean calculation, leading to more stable predictions.

However, if the threshold is set too high, the tree may underfit the data. In this case, there would be fewer splits in the tree, resulting in leaf nodes with a large number of instances. While this could reduce sensitivity to outliers, it may also lead to unreliable predictions, as the model might fail to capture important trends and patterns in the data.

Question 7

Model trees are not robust to outliers in the target features. If an outlier falls into a leaf node where the linear or non-linear model is being fitted, it can significantly skew the parameters of that model. For instance the presence of an outlier can disproportionately affect the slope and intercept in a linear regression model, which leads to predictions that may not reflect the underlying relationships of the target features for the other instances. This makes a model tree sensitive to outliers in the target feature.

Question 8

The advantages of this approach is that the tree is less prone to overfitting, as the threshold ensures that that the tree does not induce until each instance in the training data is correctly classified. This then leads to a clasification tree that is less complex and a tree that does not overfit on the training data. The computational efficiency is also reduced, as the tree performs less calculations needed to obtain optimal splits, which speeds up the inducing process of the tree.

A disadvantage to this approach is that the tree might underfit the data. This can occur if the threshold is too large and the induction process stops too early with a small number of splits. Information could also be lopst with this approach, as distinct differences between classes may be missed if the threshold is set too high, as the split may never occur. Another disadvantage is that the minority class may never be classified in a leaf node, as the splits may stop with the minority class encapsulated in the leaf node which contains more instances of another class.

Therefore, the advantages of this approach is that the tree becomes less prone to overfitting, the tree becomes less complex, which makes it easier to interpreted, and the tree induces faster. The disadvantages of this approach is that information is lossed, the model might underfit the data and the minority class may never be classified in a leaf node.

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

b

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, c, d

Question 2

A SOM used to profile customers of a medical aid company is created by training the SOM algorithm on a high-dimensional dataset without labels. The weights of the SOM are initialised by choosing a set number of random customers in the dataset, where there are $K \times J$ weights to be assigned and where K is the number of rows and J is the number of columns of the map. A 2D map is then constructed by the SOM algorithm, during the training fase of the model, that organises the customers into clusters based on similarity. The nearby nodes on the 2D map created by the SOM algorithm represents customers with similar descriptive features.

The clusters between customers is created, where each neuron can represent a different type of customer based on common patterns in the data. Since there are no labels in the training data, an expert in the field of medical aid should be asked to examine the customers and produce labels for each distinct cluster produced by the SOM. Once the clusters have received labels from the expert, the newly trained SOM model can be used to label a new customer to a specific cluster by collecting the information from the customer and adding the data to the trained SOM model. The winning neuron of the cluster would then assign a label of the corresponding cluster label of the winning neuron to the new instance. Therefore, the SOM can be used to classify new customers, so that the medical aid company can have a good idea of the potential risk of the new customer.

The newly trained SOM can also be used for regression by the medical company, to receive an exact amount of money the customer has to pay each month to be covered by the medical aid company. This is achieved by the SOM after receiving the input features of a new potential customer and finding the winning neuron. The average of the target, which is in this case the amount payed by customers to be covered by the medical aid company, over the neighbourhood of the winning neuron is calculated.

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

A SOM can be used as a classifier. This is possible by first training the SOM on the data without labels to create clusters of classes. A label is then assigned to each cluster in the SOM, where the label should be assigned by an expert in the field related to the data. New instances of the data is then given to the model, where a winning neuron from the SOM is found. The new instance is then assigned the label of the winning neuron, classifying it based on its similarity to the labeled clusters.

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)

Feed the descriptive features of the watched movie into the SOM recommender system to retrieve a winning neuron that best represents the watched movie's features. After the winning neuron has been identified for the winning neuron, the movies represented by neurons in the neighbourhood of this winning neuron are retrieved as these neurons represents movies with similar features of the watched movie. From these neighbourhood of neurons, a few other neurons that is most similar to the winning neuron is chosen as the next movie to recommend to the user, as well as the winning neurons associated movie. However, if only the best movie should be recommended, then should only the movie linked to the winning neuron is then chosen by the recommender system SOM model as the next movie that the user should watch.

(b)

The neuron activations in the SOM can be tracked to create a profile for a user after watching a movie. The winnig neuron that is achieved by feeding the SOM the watched movie's features can be tracked and saved. The more movies the user watches, the more specific neurons would be activated, which should be noted and saved by the SOM.

After the user has watched a lot of movies, the frequency of the activated neurons should be analysed, as these previously winning neurons would represent clusters of movies that the user frequently watched. The key descriptive features from the winning neuron should be extracted to identify what types of movies the user prefers.

A user profile can then be created from these extracted features, as these features indicates the users preferred movies to watch. From these created profile, new movies can be recommended to the user that closely matches the users profile.

(c)

i The node that achieves the highest quantisation error should be selected to split via the addition of an additional row an column next to the node to reduce the error of that node. The quantisation error is used as the map accuracy and is defined as the Euclidean distance of all the patterns to the codebook vector of the winning neuron. By targeting the node with the highest quantisation error, the map grows where representation of input features are the weakest, effectively reducing the quantisation error in that region and leading to a more optimal SOM structure.

ii The furthest immediate neighbour in the row and column dimension of the node with the highest quantisation error is chosen. A new node is then inserted in the column between the node with the highest quantisation error and its immediate furthest neighbour in the column dimension and the same is done for in the row dimension. This strategy of adding new nodes into the SOM ensures that the square structure of the map is retained.

iii When the new row or column is added and weights initialized for the nodes in the row or column, should interpolation between the two existing neurons bias towards that neuron with the lowest quantization error or the highest quantization error. Justify your answer.

The interpolation step is to assign a weight vector to a new neuron such that it removes patterns from the neuron that obtained the highest quantisation error. Therefore, the interpolation is problem depended when choosing to what neuron, between the neuron with the highest quantisation error and the furthest immediate neighbour of the neuron in both the column and row dimension, should bias to. If the neuron with the highest quantisation error has too much patterns assigned to it, then the new interpolation should bias towards this neuron, but if the neuron with the quantisation error has too few patterns, the interpolation should bias towards the furthest immediate neighbour in the row and column dimesnions.

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.

Additionally, if there is too much noise contained in the data, the SVM would never converge. This is because the SVM algorithm is created to make no errors in the final model. There would be no optimal hyperplane to separate classes in the case where noise is contained in the middle of a different class or there is too much noise. Therefore, the algorithm would never converge, as the SVM can not create an optimal hyperplane that splits all of the classes.

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

SVMs does not scale well to large datasets. This is because of computational complexity, memory requirements and training time.

For computational complexity SVM algorithms relies on solving quadratic or cubic optimisation problems, with a complexity of $O(n^2)$ or $O(n^3)$ respectively, where n is the number observations in the training data [4]. This complexity arises from the need to compute pairwise relationships among data points to determine the optimal decision boundary. As the dataset size increases, the number of pairwise computations grows rapidly, leading to significant increases in training time and memory usage. Consequently, training an SVM on large datasets becomes impractical due to these high computational demands.

Question 7

Kernels are applied to map a dataset into a higher dimensional space, that which creates more complex representations of the data to be able to explore the non-linear relationships between features. The SVM is then constructed on this higher dimensional feature space to ensure that input features with non-linear relationships can be separated by a hyperplane in a higher dimension that maximises the margin betwen different classes.

Question 8

Both a Gaussian mixture model and a radial basis function neural network needs to decide on how many clusters there are going to be in the model. The number of clusters is denoted by J.

A radial basis function neural network usually employs a Gaussian function as the kernel function used as the activation function in the hidden layer of the neural network. A single unit in the hidden layer of the neural network makes use of a vector of initial cluster centroids, denoted by $\boldsymbol{\mu}_j$ and a scalar deviation, denoted by σ_j controls the width of the Gaussian and determines how quickly the influence of the neuron diminishes with distance from the centroid. This vector of cluster centroids are used as the weight vectors of the neural network, where the deviation is used as the bias. In a single neuron, the kernel function is then calculated and if the Gaussian function is used as the kernel function, then will the output of the neuron be a probability that represents the similarity of an instance to the associated cluster represented by the neuron.

A Gaussian mixture model also makes use of the Gaussian functions to where each Gaussian represents a probability that the observation is associated to that cluster. Therefore, both models are similar as they make use of the Gaussian mixture model and effectively, the output of all the hidden neurons after applying the kernel function and applying softmax to these outputs can be used to represent a Gaussian mixture model.

Question 9

Support vector regression (SVR) algorithms are not robust to target outliers. The goal of the SVR algorithm is to minimise the mean squared error within a specific margin of tolerance, known as the ϵ margin. Outliers in the target values can fall outside of the ϵ margins, which results into the SVR assigning them large errors. These large errors increase the value of the mean squared error. The increased error then influences the position of the support vectors and potentially skewing the regression results toward the outliers. Although if a soft margin with slack variables are used, the SVR becomes more robust to target outliers, as some of the errors outside the ϵ margin is ignored.

Question 10

Support vector regression (SVR) is somewhat robust to noise. If the noisy instances fall within the error ϵ margin, then the SVR will be robust to noise as the noise flass within the bounds and can be used to calculate the linear regressor used to predict values. Conversely, if the noise falls outside of the ϵ error margin, the SVR algorithm becomes sensitive to noise, as the noisy instances would be assigned errors. These errors would increase the value of the mean squared error. Therefore, since the mean squared error should be minimised, the SVR would adjust the positioning of the ϵ margin and therefore, will the regression be skewed towards the outlier.

Therefore, a SVR is very dependend on the value of ϵ. If the value of ϵ is large enough to ensure noise is contained within the ϵ margin, then will the SVR algorithm be robust to noise. Conversely if ϵ is too small such that noise falls outside of the ϵ margin, then will the SVR algorithm not be robust to noise.

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 skew class distributions. 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

Q-learning is a form of model-free reinforcement learning that can also be considered a method of asynchronous dynamic programming (ADP) [5]. Q-learning enables agents to learn how to act optimally in Markovian environments by experiencing consequences of their actions, without requiring them to build maps of the domains. Q-learning is also capable of learning the optimal policy.

Table 4: Optimal Policy of Grid (a)

State	$(1,1)$	$(1,2)$	$(1,3)$	$(2,1)$	$(2,2)$	$(2,3)$
Optimal Policy	North	East	None	East	East	None

Question 2

(a)

(b)

Iteration 1:

$$V(1,1) = 0 \quad V(1,2) = 0 \quad V(1,3) = -5$$
$$V(2,1) = 0 \quad V(2,2) = 0 \quad V(2,3) = 5$$

Iteration 2:

$$V(1,1) = 0 \quad V(1,2) = 0 \quad V(1,3) = -5$$
$$V(2,1) = 0 \quad V(2,2) = 0 + 0.9(0.8 \times 5) = 3.6 \quad V(2,3) = 5$$

(c)

To learn the optimal policy, the agent must explore all possible actions to find actions that are beneficial. This is known as the exploration component. As a result of this exploration the agent starts to learn what actions are the best and what actions leads to penalties that would result in a negative rewards. This process of the agent using previously learned knowledge to select an action is referred to as exploration. As learning continuous, the amount of exploration reduces as the agent recognises which action is the best to take.

This exploration can be either performed by making use of model-based reinforcement learning by collecting data on state transitions and rewards, or by using model-free reinforcement learning like Q-learning or SARSA.

(d)

Estimates for state (1,1):

$$\begin{aligned}
\text{Trace 1} &= -5 \\
\text{Trace 2} &= 5 \\
\text{Trace 3} &= 5 \\
\text{Average of the traces} &= \frac{-5+5+5}{3} \\
\text{Average of the traces} &= \frac{10}{3} \\
\text{Average of the traces} &= 1.66666\ldots \\
\text{Average of the traces} &\approx 1.66667
\end{aligned}$$

Estimates for state (2,2):

$$\begin{aligned}
\text{Trace 2} &= 5\\
\text{Trace 3} &= 5\\
\text{Average of the traces} &= \frac{5+5}{2}\\
\text{Average of the traces} &= \frac{10}{2}\\
\text{Average of the traces} &= 5
\end{aligned}$$

(e)

The equation used to calculate the TD-learning update is as follows [6]:

$$V(s) = V(s) + \eta(r + \gamma V(s') - V(s))$$

where s is the current state, s' is a future state, η is the learning rate, r is the immediate reward, and γ is the discount factor.

Trial 1:

$$\begin{aligned}
V(1,1) &= 0 + 0.1(0 + 0.9(0) - 0) = 0\\
V(1,2) &= 0 + 0.1(-5 + 0.9(0) - 0) = 0.1(-5) = -0.5
\end{aligned}$$

Trial 2:

$$\begin{aligned}
V(1,1) &= 0 + 0.1(0 + 0.9(-0.5) - 0) = 0.1(-0.45) = -0.045\\
V(1,2) &= -0.5 + 0.1(0 + 0.9(0) + 0.5) = -0.5 + 0.1(0.5) = -0.45\\
V(2,2) &= 0 + 0.1(5 + 0.9(0) - 0) = 0.1(5) = 0.5
\end{aligned}$$

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

[4] A. Ben-Hur, J. Weston, G. Maira, and B. Schölkopf, "Support vector machines and kernels for computational biology," *Journal of Machine Learning Research*, vol. 13, pp. 1–19, 2013.

[5] C. J. C. H. Watkins, "Learning from delayed rewards," *PhD Thesis*, 1989.

[6] A. P. Engelbrecht, *Computational intelligence: an introduction.* John Wiley & Sons, 2007.