1 F1e

Your friends Alice and Bob are discussing.

Alice: I assumed that the features and the label are linearly related and therefore I fitted a linear regression model. However, my results are not very good.
Bob: Come on, Alice. It is stupid to make assumptions. The whole point of machine learning is to learn from data. Assumptions just make it more likely that you are wrong. You should never make assumptions.

How would you resolve their disagreement?
Fill in your answer here

Maximum marks: 3

2 F1f

Your friends Alice and Bob are learning a regression model for a small data set. They are discussing.

Alice: I'm using a simple linear model but it does not generalise.

Bob: Interesting. I'm using a deep neural network with 7 layers with 500 neurons each. However, it does not generalise either.

Alice: I tried to take bootstrap samples of my training set and learned a model with each of the new training sets. However, all my models look pretty much the same.

Bob: I tried that too but each of my models is very different. The common thing is that none of them generalises.

Can you help Alice and Bob to figure out what is going on? What could be an explanation for the behaviour of their models.

Fill in your answer here

Maximum marks: 3

3 F1b

Your friends Alice and Bob are discussing.

Alice: I fitted a SVM on the data set that I have. However, I didn't get good results.

Bob: Come on, Alice. Everybody knows that deep neural networks are the best machine learning algorithm. You can always get reasonable results using them and thus there is no need to use other algorithms.

How would you resolve their disagreement?

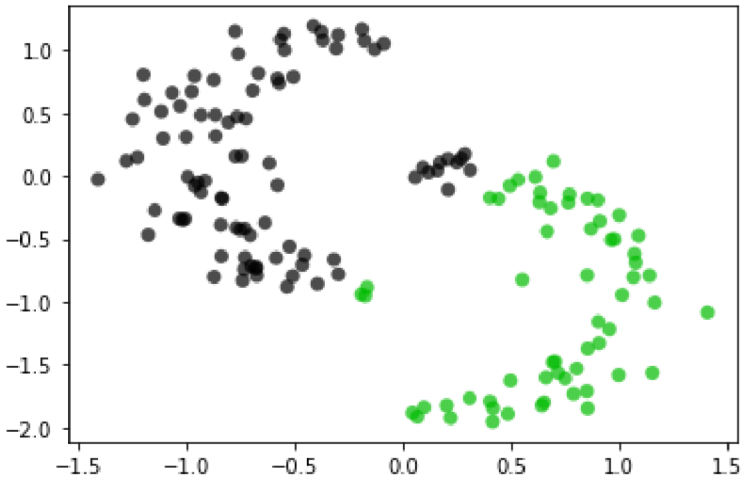
Fill in your answer here

Maximum marks: 3

4 M2c

Your friends Alice and Bob are discussing.

Alice: I implemented the Lloyd's algorithm for k-Means clustering. I ran my software and got the clusters (black and green) shown in the figure below. However, I think that there is a bug in my software because the clusters do not make any sense even though there are two clear clusters in my data.



Bob: I don't think it is about a bug. Lloyd's algorithm is known to converge to a local optimum. You should try another initialisation and your problems will probably go away.

How would you resolve their argument?

Fill in your answer here

Maximum marks: 3

5 M2h

Your friends Alice and Bob are discussing.

Alice: I am trying to predict when a component fails. I have a validation data set with 10 failed components and 9,990 working ones. My classifier has a validation accuracy 99.5%.
Bob: My classifier is better than yours. Its validation accuracy is 99.9%.
Alice: Impressive! How do you do it? I managed to correctly predict 9 of the failed components. However, I also had 49 false positives.
Bob: I recognised all working components correctly.

Which classifier is better? Justify your answer.

Fill in your answer here

Maximum marks: 3

6 E3a

Charlie has recently heard about dimensionality reduction and is excited about it. He tells about his experiments in this field below.

Is this a good answer? What parts are good and what could be improved? If there are any errors in the answer how should they be corrected?

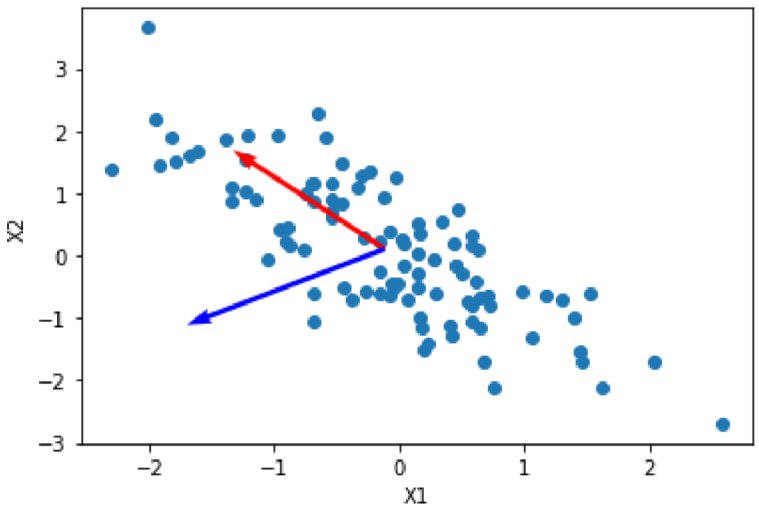
"It is cool to know that you can represent high dimensional data sets in smaller scale and still retain most of the information. There are two types of dimensionality reduction methods: feature selection and feature extraction.

In feature selection, you represent a data set with less data points. Basically, we use different methods to figure out which data points are most important and throw away the rest.

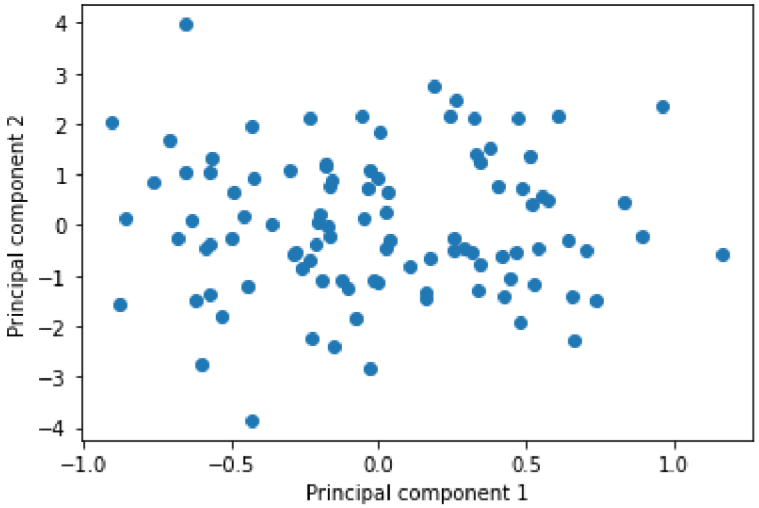
In feature extraction, you create a small number of new features based on the original ones and then throw away the original ones and do your analysis with the new features.

I have recently tried Principal Component Analysis (PCA) which is a linear feature extraction scheme. In other words, new features are linear combinations of the original features.

As an experiment, I implemented PCA. You can see my results below. So my original data is two-dimensional. If you would like to represent it with only one dimension, you would project the data to the first principal component (blue arrow in the figure below) and throw away the second component.



Here is the data after we have projected it to the new space.



The PCA works as follows: First, we take our data matrix X and subtract the column mean from each column. Then we multiply the transpose of X with X and get the covariance matrix. Next, we find eigenvalues and eigenvectors of the covariance matrix. The first principal component is actually an eigenvector that corresponds to the largest eigenvalue."

Fill in your answer here

Maximum marks: 5

7 E3b

Charlie has recently heard about neural networks and he thinks that they are the coolest thing in the world. He explains below what neural networks are.

Is this a good answer? What parts are good and what could be improved? If there are any errors in the answer how should they be corrected?

"An artificial neuron is a simple object. Each neuron has associated a set of weights. Its input is a vector. First, the artificial neuron computes a weighted sum of the elements of the input vector and then it transforms the weighted sum using a non-linear activation function. The output of the activation function is also the output of the neuron. There are lots of different activation functions. For example, rectified linear unit (ReLU) is a commonly used activation function. By definition, $\boldsymbol{relu(z) = \frac{1}{1+e^{-z}}}$.

A neural network consist of artificial neurons that are organised in layers. The first layers consist of the input. Each hidden layer has one or more artificial neurons whose input is the output from the previous layer. Finally, the last layer is the output layer which is the output of the model. Output can be either continuous or categorical and thus neural networks can be used in both classification and regression tasks. However, they cannot be used in unsupervised learning.

These days we use deep learning which means neural networks that have several layers of hidden neurons. The reason why we use these deep networks is that it is impossible to approximate complex functions using just one layer of hidden neurons. Another reason why deep neural networks are popular is because they have lots of parameters and therefore they seldom overfit.

Neural networks are usually trained using gradient descent with help of back-propagation. Back-propagation is an efficient algorithm for computing the gradient of a loss function. "

Fill in your answer here

Maximum marks: 5

8 G5a

Consider univariate linear regression. Suppose we have n observations (x_i, y_i) where $x_i \in \mathbb{R}$ is one-dimensional feature and $y_i \in \mathbb{R}$ is a continuous label.

Now, we have

$$y_i = wx_i + b + \epsilon_i$$

where w and b are the parameters and ϵ_i is the error for the ith observation.

We want to penalise outliers very strongly and thus we use a special loss function based on fourth degree polynomials: $L(y, \hat{y}) = (y - \hat{y})^4$.

This leads to the following loss over the whole data set:

$$E[w, b] = \sum_{i=1}^{n} (y_i - wx_i - b)^4.$$

Suppose we have observed two data points $(2, 3)$ and $(3, 4)$. The goal is to find parameter values w and b that minimise the loss $E[w, b]$.

Use initial values $w_0 = 1.2$ and $b_0 = 0.8$ and learning rate 0.05. Perform one step of gradient descent, that is, update the parameters w and b once.

Show your work. How does the update affect the loss?

Upload your answers in ONE single PDF file below.

Please make sure that the scan/ the photo/ document is readable.

Mark every page clearly with:

- Your candidate number (do not write your name or student number)
- Course code
- Question number
- Page number

Upload your file here. Maximum one file.

The following file types are allowed: **.pdf** Maximum file size is **2 GB**

Select file to upload

- Permitted file format: .pdf
- Maximum file size: 2 GB
- Only ONE file upload permitted

Maximum marks: 10

9 H4b

Suppose we have 6 data points named as A, B, C, D, E, and F. The matrix below shows pairwise distances between the points; note that distances are non-Euclidean.

	A	B	C	D	E
B	4				
C	5	2			
D	12	1	8		
E	13	10	14	11	
F	3	15	7	9	6

Construct a hierarchical clustering of the data points using agglomerative hierarchical clustering with **complete linkage**.

Show intermediate steps. Answer containing only the final solution without intermediate steps will get 0 points.

Upload your answers in ONE single PDF file below.
Please make sure that the scan/ the photo/ document is readable.

Mark every page clearly with:

- Your candidate number (do not write your name or student number)
- Course code
- Question number
- Page number

Upload your file here. Maximum one file.

The following file types are allowed: **.pdf** Maximum file size is **2 GB**

Select file to upload

C

- Permitted file format: .pdf
- Maximum file size: 2 GB
- Only ONE file upload permitted

Maximum marks: 10

10 S6a

Consider the three models shown below. Answer the following questions:

1. For each model, argue whether it is overfitting or not. Justify your decision. Which indicators suggest overfitting, which not?
2. Which model would you choose? Why?
3. Based on the information that you have below, what is your unbiased estimate on the performance of the selected model on unseen data? Why?

We have a binary classification problem with two continuous features. In the figures below, the positive class is red and the negative is blue. A set of results for each model is shown below.

Model A

	Training data	Validation data
Accuracy	0.975	0.87
Precision	0.98	0.81
Recall	0.94	0.76
F1	0.96	0.79

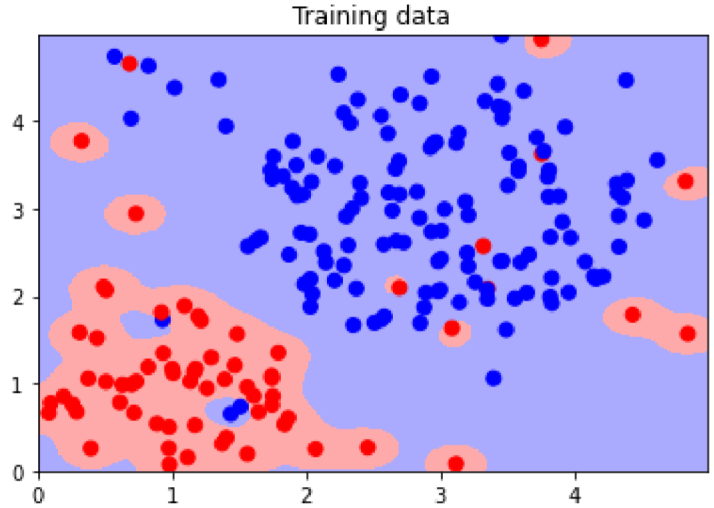
Confusion matrix on training data

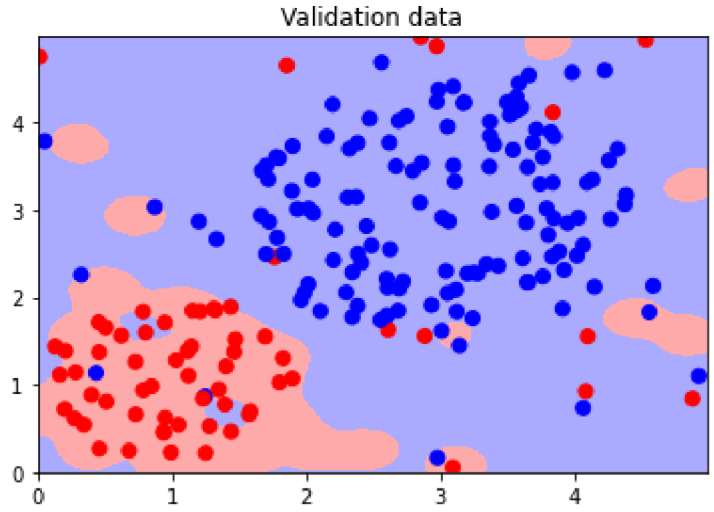
		Predicted label	
		blue	red
True label	blue	132	1
	red	4	63

Confusion matrix on validation data

		Predicted label	
		blue	red
True label	blue	125	11
	red	15	49

Decision boundaries:





Model B

	Training data	Validation data
Accuracy	0.935	0.91
Precision	0.95	0.90
Recall	0.85	0.81
F1	0.90	0.85

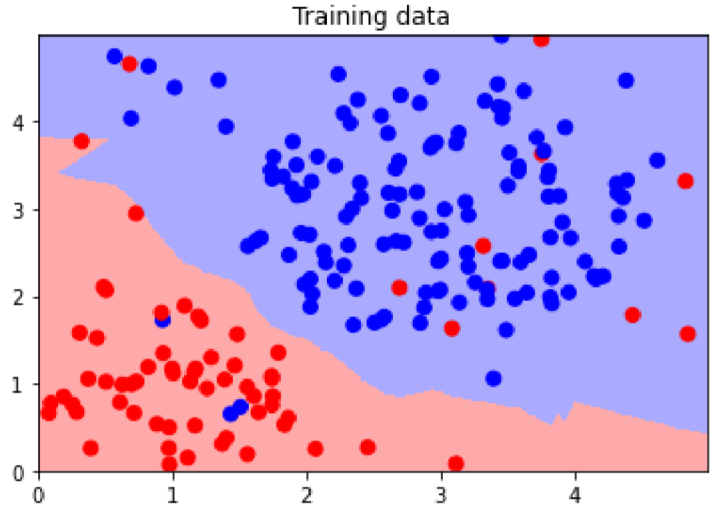
Confusion matrix on training data

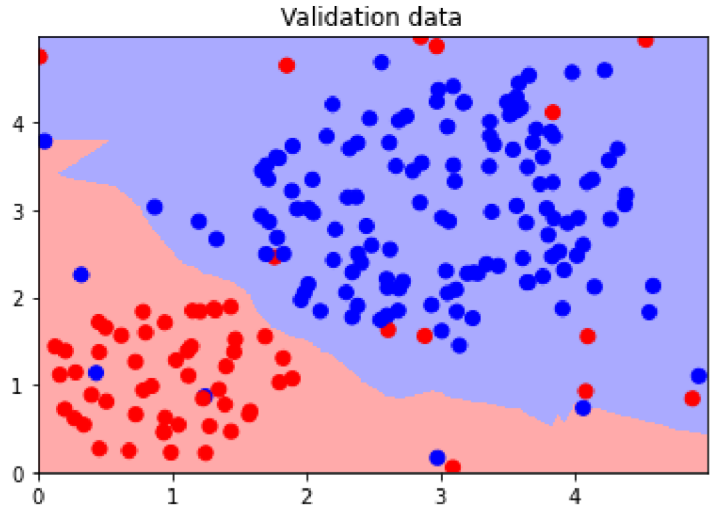
		Predicted label	
		blue	red
True label	blue	130	3
	red	10	57

Confusion matrix on validation data

		Predicted label	
		blue	red
True label	blue	130	6
	red	12	52

Decision boundaries:





Model C

	Training data	Validation data
Accuracy	0.93	0.915
Precision	0.93	0.87
Recall	0.85	0.86
F1	0.89	0.87

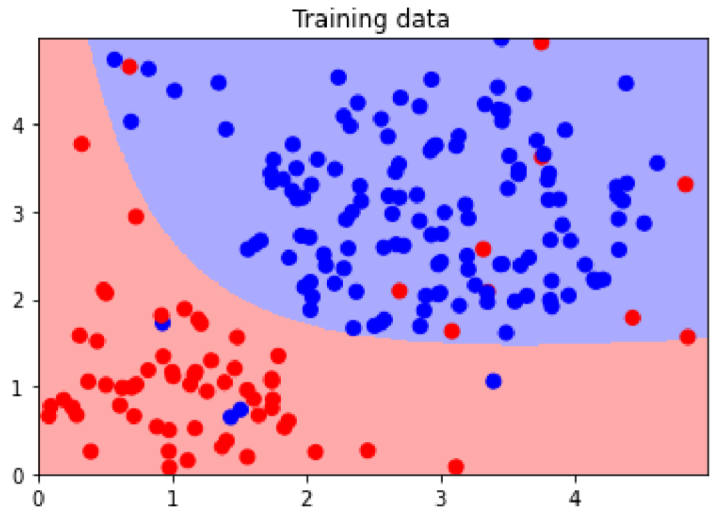
Confusion matrix on training data

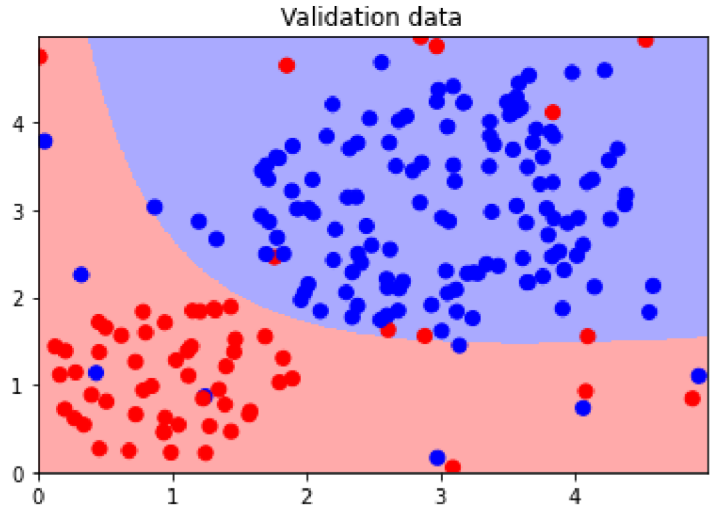
		Predicted label	
		blue	red
True label	blue	129	4
	red	10	57

Confusion matrix on validation data

		Predicted label	
		blue	red
True label	blue	128	8
	red	9	55

Decision boundaries:





Fill in your answer here

Maximum marks: 10