1. What is an inductive bias?

   (a) Inductive Bias are all the assumptions about the nature of the target function and it's selection. We can have restriction, limiting the hypothesis space, or preference, ordering the hypothesis space.

2. What are Bias and Variance ? What are overfitting and underfitting? How are they related?

   (a) Bias refers to how well the model can represent any training set. Variance is the sensitivity to changes in the training set. When we have high variance, the model is overfitting, the model fits too well with the training set but is no able to generalise well to the test set. When we have high bias, the model is underfitting, the model is unable to fit the training set and also the test set.

3. Describe the Linear Regression, discuss its hypothesis space, its inductive bias, how to make predictions on new examples, and how to train it.

   (a) Is a supervised learning model used for regression problems. The main goal of this model is to find a linear function by choosing the θ parameters such that the hypothesis $h_\theta(x)$ is close to y for our training example. The hypothesis function is defined as $h_\theta(x) = \theta_0 + \theta_1 X$ (if we are in a multidimensional space we just compute the product of the theta vector transpose with the input vector). The inductive bias assumes a linear relation between input features and the target, which is not always true. To make a new prediction, we compute weighted sum of thee feature inputs and model parameters, then directly use the result as our prediction. The training process aims to minimize the least squared (residuals), the sum of the difference between the predicted label and the actual label, squared, defined by the cost function J(θ) = $\frac{1}{2m}$ the sum of the least squared with respect to each theta parameter. To find the optimal parameters theta, we use the Gradient Descent algorithm, where we compute the derivative of the cost function and update each theta parameter (one derivative on the parameter to update each parameter) by subtracting a fraction (learning rate) of the gradient. The algorithm stops when convergence is reached.

4. Apply the Gradient Descent on this example: (1,2),(2,2.8),(3,3.6) and $\theta_0 = 0, \theta_1 = 0, \alpha = 0.1$.

   (a) Derivate the loss function for θ_0

   (b) Derivate the loss function for θ_1

   (c) Keep recursively until convergence

5. Classify with Linear Regression this example:θ =[1,1], predict x=4.

   (a) $1 + 1 * 4 = 5$

6. Describe the Perceptron, discuss its hypothesis space, its inductive bias, how to make predictions on new examples, and how to train it.

   (a) The perceptron is a learning algorithm for binary classification problems which separates data with a panel called hyperplane. Given x in R^d, its hypothesis space is composed of all linear functions in the input space $\Theta^T x$, where $\Theta \in R^d$ are parameters we want to learn from the data. Its function takes an input value x and maps it into {1,-1} and the decision rule is the sign of the hypothesis on the example. The inductive bias is the assumption that the data is linearly separable and that a linear hyperplane is sufficient for classification, but not always is possible. So, the inductive bias of the perceptron are all the assumptions made on the hypothesis space to generalize from training to unseen data. The perceptron is trained by adjusting the weights iteratively whenever a misclassification occurs, using the update rule. To predict a new example you compute the product with the example $X =< 1, 2, 3 >$ and the weights $\Theta =< 2, 3, -6 >$, then do the sign, if is < 0 the classification value is -1, if is $\geqslant 0$ the classification value is 1. The perceptron will find a hyperplane in k¡R^2/γ^2 steps, where R is the maximum norm of training examples and γ is the margin.

7. Train the perceptron on this examples: $X = \{(1,1)(2,1)(3,3)\}, y = \{1, -1, 1\}, \theta$ =[0,0].

   (a) $< 1, 1 > * < 0, 0 >= 2 \rightarrow 1$ Ok

(b) $< 2,1 >< 0,0 >= 0 \rightarrow 1$ No, $\theta =< 0,0 > + < 2,1 > *(-1) =< -2,-1 >$

(c) $< 3,3 >< -2,-1 >= -9 \rightarrow -1$ No, $\theta =< -2,-1 > + < 3,3 > *1 =< 1,2 >$

(d) Keep going until convergence

8. Classify with Perceptron this example:θ =[1,-1],x=[2,3].

   (a) $1 * 2 + -1 * 3 = 2 - 3 = -1 \rightarrow sign(-1) = -1 \rightarrow -1$

9. Describe the Logistic Regression, discuss its hypothesis space, its inductive bias, how to make predictions on new examples, and how to train it.

   (a) Is a supervised learning model used for binary classification problems. The main goal of this model is to find a linear function by choosing the theta parameters so that the hypothesis space $h_\theta(x)$ is close to y for our training examples. The hypothesis space will be defined as $\frac{1}{1+e^{g(z)}}$ where $g(z) = \theta^T X$. The main inductive bias of this algorithm is that it assumes the relationship between input features and the target variable follows a linear trend in the log space, which is not always true. To make a new prediction, we need to compute the sigmoid of the computed hypothesis, based on this result, we select the predicted label using a threshold, typically 0.5 (1 if is equal or over, 0 if is under). During training we aim to minimize the loss function $J(\theta)$ as one over m the sum of the $Cost(h_\theta(x), y)$, defined as -y*log($h_\theta(x)$) -(1-y)log(1-$h_\theta(x)$). To find the optimal parameters theta, we use the Gradient Descent algorithm, where we compute the derivative of the cost function and update each theta parameter (one derivative on the parameter to update each parameter) by subtracting a fraction (learning rate) of the gradient. The algorithm stops when convergence is reached.

10. Apply the Gradient Descent on this example: X={1,2,3},y={0,1,1},θ =[0,0], $\alpha = 0.1$.

    (a) Derivate the loss function for θ_0

    (b) Derivate the loss function for θ_1

    (c) Keep recursively until convergence

11. Classify with Logistic Regression this example: θ =[-3,2], x=2.

    (a) $\frac{1}{1+e^{-3+2*2}} = 0.27 \rightarrow 0.27 \leqslant 0.5 \rightarrow 0$

12. What is the role of a regularization hyperparameter? List three examples of learning algorithms, where their regularization depend on, which hyperparameter is used to control it and how.

    (a) Overfitting is a big problem when we train a model and to address it we can apply regularization. Consists in keeping all the features, but reduce the influence of each parameter theta. We can learn our parameters with gradient descent by adding the regularized part, consisting in adding to the cost function a regularization term lambda:

    - Linear Regression: lambda(sum of theta parameters squared).
    - Logistic Regression: lambda/2m(sum of theta parameters squared).
    - Neural Networks: lambda/2m(sum of theta parameters of each neuron in each layer squared).

    When we have this hyperparameter lambda, we need to split the training set into training and test set to optimize the hypothesis parameters.

13. What is hold-out method?

    (a) We keep a subset of v samples from the training set to evaluate our hyperparameters. In this way hyperparameters lambda are optimized on the training/validation sets. The best lambda is selected on the validation set and after you can retrain the model and evaluate its performance on the test set.

14. What is the Leave One Out method?

    (a) Suppose we have m examples in the training set, we train our model on m-1 examples and then check the error on the example we leave out. We repeat the process for m times and we average the predictions. This method is unbiased but very expensive.

15. What is the K-Fold cross validation?

    (a) Is like the leave one out method, but we leave out a number of examples instead only one for the test. The thumb rule states that k=10 when we can afford it.

16. How we can search the best hyperparameter?

    (a) If we try too many values for the hyperparameters, the error on the validation set is no longer a good estimator for the true risk. We need to perform a telescopic search: we try a number of values of different order of magnitude and select the best lambda (lambda*). If lambda* is at the border of the range try bigger values until you get a lambda* which is between two values with lower performance.

17. What is the VC dimension?

    (a) Measures the complexity of the hypothesis space. The VC dimension of an hypothesis space is the size of the largest S subset of X shattered by H. A set S is shattered by H if and only if for every dichotomy of S, there exists a h in H consistent with the dichotomy. Every h in H partitions a set S into two classes is a dichotomy (2^n possible combinations to check the panel). h is consistent with S if and only if for each (x,y) in S.h(x)=y (h makes no errors on S).

18. Calculate the VC dimension: Consider the hypothesis class of vertical lines in 2D space.

    (a) Find a hyperplane that separates all the possible classifications of a combination of 1 point $\rightarrow VD(H) + 1$.

    (b) Find a hyperplane that separates all the possible classifications of a combination of 2 points $\rightarrow VD(H) + 1$.

    (c) Keep going until you can't find that hyperplane, the VC dimension is the previous number of points.

19. What is the difference between parametric and non parametric models?

    (a) A parametric model is a model with a prefixed number of parameters such as the Linear Regression, the Logistic Regression ecc. On the other hand, non parametric models have a variable number of parameters, which normally depends on the size of the training set.

20. Describe the 1-NN, discuss its hypothesis space, its inductive bias, how to make predictions on new examples, and how to train it.

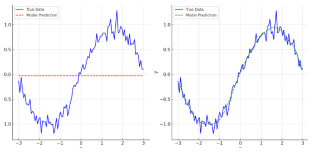
    (a) Is a non parametric supervised learning model typically used for classification tasks, and is also a lazy learner, where all the work is done at prediction time. This algorithm compute the distance between the new sample and all the training samples and select the closest example. The decision rule is to assign the label of the nearest class among the nearest neighbours and that distance is usually an euclidean distance (in alternative it can be used a custom distance). It assumes that the most of the cases in a small neighbourhood of a point in feature space belong to the same class. It is sensitive to outliers and to avoid this problem we can apply feature scaling, by linearly scaling all the values between [0,1], scaling each dimension to have 0 mean and 1 variance or divide each vector by its norm. With large vectors is subject to the course of dimensionality: in high dimensional spaces all points are the same distance.

21. Describe the K-NN, discuss its hypothesis space, its inductive bias, how to make predictions on new examples, and how to train it.

    (a) Is a non parametric supervised learning model typically used for classification tasks, and is also a lazy learner, where all the work is done at prediction time. This algorithm compute the distance between the new sample and all the training samples and select the k closest examples (usually odd and selected through cross validation). The decision rule is to assign the label of the majority class among the k nearest neighbours and that distance is usually an euclidean distance (in alternative it can be used a custom distance). It assumes that the most of the cases in a small neighbourhood of a point in feature space belong to the same class. It is sensitive to outliers and to avoid this problem we can apply feature scaling, by linearly

scaling all the values between [0,1], scaling each dimension to have 0 mean and 1 variance or divide each vector by its norm. With large vectors is subject to the course of dimensionality: in high dimensional spaces all points are the same distance.

22. Recognize if this are situations of high variance or high bias:



    (a) On the left we have a case of high bias (underfitting) and on the right we have a case of high variance (overfitting)

23. What can we do if we have high variance? And if we have high bias?

    (a) If our learning model doesn't work as expected we might have high bias or high variance problems. If we have problems with variance we can: get more training data, try smaller features and try to increase the hyperparameters. If we have problems with bias we can: try to get more features, try to add complexity to the model or try decreasing the hyperparameters.

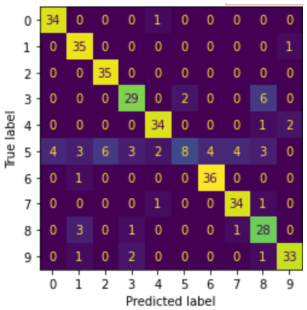
24. What are accuracy, precision, recall and F1? How are they computed?

    (a) Accuracy is the ratio of correctly classified instances over all (TP+TN/ TP+TN+FP+FN). The precision is the ratio between the right positive classifications over all the positive classifications (TP/TP+FP). The recall is the ratio between the right positive classifications and the ground truth (true positive plus false negative, TP/TP+FN). The F1 score is the harmonic mean between precision and recall (2*(Pr*Rc)/(Pr+Rc)).

25. What are the precision-recall curves? How are they computed?

    (a) Precision-Recall curves are obtained by computing precision and recall plots as function of hyperparameters (x=Rc, y=Pr). ROC curves are the same plots but using the FPR and the TPR and than calculating the area under the curve (x= False Positive Rate, y= True Positive Rate).

26. How can we compute the accuracy from the following confusion matrix?



    (a) The sum of the diagonal elements over the sum of all elements in the matrix: all correctly classified / all elements.

27. What is a decision tree? Describe them. How are they related to random forests?

    (a) A decision tree is a structure in which internal nodes represent attributes, leaf nodes represent class labels and branching is determined by the attribute value. They provide interpretable results and present the hypothesis function h(x). A tree is built by splitting the training set into subsets which constitute the successor children and splitting is based on decision boundaries based on feature values. You may end up having a classification tree if you have discrete output, or a regression tree if your output is continuous. The goal is to select the splitting such that the resulting areas have as many examples of the same class as possible. To

select this split we can base on the Entropy, on the Gini Index or non the Information Gain. Decision trees are related to the random forests since this is an algorithm consisting of many decision trees that uses bagging and feature randomness to combine trees and then making an ensemble out of it, with majority voting for the assignment of the final class.

28. Classify this example with the decision tree: (Rainy, Cool)

Weather	Temperature	Play Tennis
Sunny	Hot	No
Rainy	Mild	Yes
Overcast	Cool	Yes

    (a) Sunny→No and Rainy/Overcast → Mild/Cold → Yes, so (Rainy,Cool) → Yes.

29. What is the difference between Entropy and Gini?

    (a) Entropy H is a measure of the amount of uncertainty (or randomness) in the dataset S. Is defined as $H(S) = -\sum_{c \in C} p(c) log_2(p(c))$, where C is the set of classes in S and p(c) is the proportion of elements in C to the elements in S (for boolean r.v. becomes $H(S) = -(q log_2(q) + (1-q) log_2(1-q))$). The Gini Index (Gini) measures the probability of incorrectly classifying a random chosen element if it is labelled according to the distribution of labels in the dataset $Gini(S) = 1 - \sum_{c \in C} p(c)^2$.

30. Calculate Entropy and Gini on this example:

Class	Count
A	3
B	2

    (a) $H = -(\frac{3}{5} log_2(\frac{3}{5}) + \frac{2}{5} log_2(\frac{2}{5}))$

    (b) $Gini = 1 - (\frac{3}{5}^2 + \frac{2}{5}^2)$

31. Describe the ID3 algorithm.

    (a) Is an algorithm used to build a decision tree by finding the best split to maximize the Information Gain (IG) from that split.

32. Apply the ID3 algorithm on this example:

Outlook	Temperature	Wind	Play
Sunny	Hot	Weak	No
Sunny	Mild	Strong	No
Overcast	Hot	Weak	Yes
Rainy	Cool	Weak	Yes

    (a) Calculate entropy for the entire dataset

    (b) Choose the attribute with the highest information gain

    (c) Split data based on that attribute and repeat recursively

33. What is the Information Gain (IG)?

    (a) Measures the difference in entropy from before to and after the set S is split on an attribute a, so IG(S,a) measures how much uncertainty in S was reduced after splitting set S on attribute a $IG(S, a) = H(S) - \sum_{t \in T} p(t) H(t)$ where p(t) is the proportion in class T and H(t) is the entropy on pos/neg classification of such class.

34. What is pruning?

    (a) Pre-pruning stops the tree's growth before it comes fully developed. This leads to have max depth, minimum number of samples required to split and minimum improvement in the impurity measures.

    (b) Post-pruning starts with a fully grown tree and removes branches that provide little predictive power. Removes branches based on a trade-off between complexity and performance; eliminates nodes if their removal does not increase the tree's error on validation set.

35. What is the difference between multiclass and multilabel classification?

    (a) Multiclass classification consists of classifying an instance through multiple classes but belonging to ONLY one.

    (b) Multilabel classification consists of assigning to an instance multiple different labels.

36. What is the One VS One approach?

    (a) If there are k classes in our problem, we train $k(k-1)/2$ binary classifiers. Each binary classifier is trained to discriminate between two classes.

37. What is the One VS All approach?

    (a) We train a classifier per class. Each binary classifier is trained using its positive samples and negative samples belonging to all the other classes as negative. This strategy requires each classifier to produce a real valued confidence score for its decisions. At prediction time we select the classifier with the highest confidence score.

38. Describe the SVM, discuss its hypothesis space, its inductive bias, how to make predictions on new examples, and how to train it.

    (a) SVM is a non parametric supervised learning model, without a prefixed number of parameters, a batch algorithm, usually used for classification tasks. They key idea is to find the optimal separating hyperplane that maximizes the margin between different classes in the feature space. The hypothesis space consists of all the hyperplanes that can separate the data $w^T x + b$, where w is the weights vector, x is the example and b is the bias of the classification. For non-linearly separable data, SVM uses kernel functions, which implicitly map data into a higher-dimensional space where it becomes linearly separable. The inductive bias are related to the boundaries that maximize the margin, assuming that a larger margin leads to better generalization. The model is also assuming that data are linearly separable in the original dimensional space or in a higher dimensional space and that only the support vectors are crucial in defining the decision boundary. The prediction is based on the formula $sign(w^t x + b)$ or a kernel based decision function and the training is based on solve the convex optimization problem using slack variables.

39. Demonstrate the SVM margins.

    (a) Given a dataset $\{x_i, y_i\}_{i=1}^m$, where $x_i \in R^d$ are feature vectors and $y_i \in \{-1, +1\}$ are the corresponding class labels, the decision boundary of a linear SVM is defined as $w^T x + b = 0$ where w is the weight vector and b is the bias term. For a given training example (x_i, y_i), the functional margin is defined as $\gamma_i = y_(w^T x_i + b)$. The margin depends on the scale of w, so it is normalized to obtain $\hat{\gamma} = \frac{y_i(w^T x_i + b}{||w||}$ and the margin of the hyperplane is therefore $\hat{\gamma} = \frac{1}{||w||}$. Maximizing the margin is equivalent to minimizing $||w||$, which leads to better generalization. To find the optimal hyperplane, SVM solves $min_{w,b} \frac{1}{2}||w||^2$ subject to the constraint $y_i(w^T x_i + b) \geqslant 1, \forall i$. This ensures that all data points are correctly classified with at least a margin of 1.

40. Describe the Kernelised Perceptron, discuss its hypothesis space, its inductive bias, how to make predictions on new examples, and how to train it.

    (a) In the base perceptron, when the problem is non linearly separable we can create new features by modifying the input examples via a function $\phi(x)$. For any function $\phi(x)$ for which a product can be computed, there exists a kernel function $K()$ such that the phi function applied to the values is equal to the kernel function, where K() is a similarity function.

41. How can be changed the perceptron in such a way that we might have a chance to make zero mistakes during training?

    (a) By using a Kernelised Perceptron we might have a chance to have 0 error during training.

42. Describe the SVM dual, discuss its hypothesis space, its inductive bias, how to make predictions on new examples, and how to train it.

(a) Is a SVM version used in high dimensional spaces or kernelized problems. The hypothesis space consists of linear combinations of training examples (defined with the choice of the kernel), weighted by a coefficient α_i. Instead of optimizing weights directly, the SVM dual optimizes these coefficients, representing the importance of each training point. The inductive bias are still the same as the primal SVM, as the maximization of the margin, the decision boundary influenced only by the support vectors, and in addition we have the assumption of linearly separability for the choice of the kernel. The predictions are made by evaluating the sign of the decision function previously obtained with a weighted sum using support vectors and kernels and the training optimized the primal one with the lagrange multipliers.

43. What is a Kernel function?

    (a) Is a function that the shape of the decision boundary and significantly impact the classification accuracy. There are linear, polynomial kernels and RBF kernel = $exp(-\frac{||x-x'||^2}{2\sigma^2})$.

44. What are the properties of Kernel Matrices? What is a valid kernel?

    (a) A matrix is symmetric positive semidefinite if for each eigenvalue $\lambda, \lambda \geqslant 0$. A kernel function is positive semidefinite if every possible dataset is positive semidefinite (can have a global optimum).

45. Which methods can be used for Structured Datas?

    (a) Subtree Kernel: counts all matching proper subtrees of the two inputs.

    (b) Subset Tree Kernel: counts the number of matching subset trees.

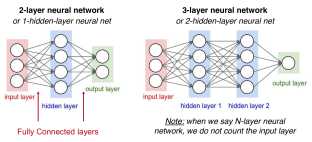
46. What is a NN? What is his architecture? What are his inductive bias?

    (a) Is a group of different neurons stacked up together with a non linear activation function. The architecture refers to how the different neurons are connected to each other. The two metrics used are the number of neurons and the number of parameters. In a NN there is no representation bias and a NN with one hidden layer is able to approximate any function (with an undefined number of neurons).

47. What is the difference between feed-forwarding and back-propagation?

    (a) In a feed-forward NN the outputs of each layer are going to be the input of the next layer. With back-propagation each unit j is responsible for a fraction of the error in each of the output units to which it connects. The score is propagated back to provide the error values for the hidden layers.

48. Calculate the number of parameters of this NNs:



    (a) First: $3 * 4 + 4 * 2$

    (b) Second: $3 * 4 + 4 * 4 + 4 * 1$

49. What is the dropout method?

    (a) Consists in randomly deactivating some of the neurons during training to avoid for a neuron to become predominant.

50. What is clustering? What is a cluster?

    (a) Is used in unsupervised learning. Given a dataset we want to partition it into k clusters (group of data points whose intra class distances are small compared with distances to points outside)

51. Describe the K-Means.

(a) Is used in unsupervised learning and is a clustering algorithm. Given a dataset we want to partition it into k clusters. K is find through cross validation (number of clusters) and we want to minimize the distance between the elements in the cluster with the cost function $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2$. Initially we random generate k cluster centroids and we assign each example to the nearest centroid (distance calculated with euclidean distance) and then we recalculate the position of the centroids by averaging all the cluster elements with that label. Then we repeat this algorithm from step 2 until convergence of the centroids. With this algorithm the convergence is assured but the choice of the initial centroids can lead to find a local minimum rather than a global minimum, so we might try different initializations. This algorithm suffers of the course of dimensionality, where in high dimensional spaces data points become sparse making clustering difficult; and suffers of overfitting and noisy features since in high dimensional datasets, irrelevant features can distort clustering results.

52. What are the Rand Index and the Adjusted RI?

    (a) The Rand Index (RI) evaluates the agreement between the clustering results (as accuracy) and the Adjusted RI takes into account the random probability of assignment.

53. What are ensemble methods? Which type can be?

    (a) The general idea is to get predictions from multiple models and aggregate the predictions. An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way to classify new examples. An ensemble can be:
        i. Parallel: pick the prediction with the highest number of votes (voting) or multiple base learners are built from different samples of the training set to make predictions (bagging).
        ii. Sequential: the overall performance can be boosted incrementally.

54. What is Bootstrapping? What is a common algorithm used? Explain it.

    (a) Consists in sampling with replacement M overlapping groups of instances of the same size. The main purpose is that each model sees a random subset of features, called feature randomization. A common algorithm used is the Bootstrap Aggregating Approach:
        i. Create k bootstrap samples
        ii. Train a distinct classifier on each sample
        iii. Classify new instances by majority vote/average

        One big problem with bootstrapping is that assumes the independence of weak learners, but they are not and so bagging tends to reduce variance but increase bias.

55. What is a typical bagging algorithm used? Explain it.

    (a) The Random Forest Algorithm is used for bagging:
        i. Use k bootstrap replicates to train k different decision trees
        ii. At each node, pick a subset of features at random
        iii. Aggregate the predictions of each tree to make classification decision

56. What is Boosting? What is a typical algorithm used? Explain it.

    (a) Through sequential training with example re-weighting we can take a weak classifier and boost it.
        i. Use the training set to train a simple predictor
        ii. Re-weight the training examples, putting more weight on examples that were poorly classified in the previous predictor
        iii. Repeat n times
        iv. Combine the simple hypothesis into a single accurate predictor

    (b) With Adaboost (Adaptive boosting) instances weights are updated using an exponential rule, so harder examples weight exponentially more than easy ones.

57. What is stacking? What are the main principles?

    (a) Is a technique for combining an arbitrary set of learning models using a meta-model (meta-learner). Stacking works at its best when the base models have complementary strengths and weaknesses.