03 Logistic Regression

- Linear relationship between the output and the input variable
- Mean Squared Error
    - the difference between the actual y values present in the dataset (supervised learning) and the H(x) values predicted by the model
    - [H(x) - y]^2 - cost function
    - small value → good
    - Large value → bad
    - MSE = sum the squared of all error term
- Prediction Equation - H(x) = b0 + b1x -
    - Aim is to find optimal b0, and b1
    - min b [H(x) - y]^2
        - we want to find the minimum by tuning these parameters b
        - y is the value we know from the training data // supervised learning
- Linear Regression - Parameters
    - R^2 statistics
        - Measure the accuracy of the regression models
        - square of the correlation coefficient in R
        - measures how strong a linear relationship between two variables
        - R^2 = 1 - (RSS/TSS)
            - RSS - residual sum of square.
            - RSS measures the variability left unexplained after performing regression
            - RSS = Sum [H(x) - x]^2
            - TSS is the sum of squares
            - measures the total variance in y
            - TSS = 1/n sum(y-mu)^2

04 - Logistic Regression

- Logistic regression solves classification problems
- Usually a method of binary classification
- Outcome of dependent variable is discrete
- Assigns probabilities to given outcomes
- Logistic regression uses sigmoid-function
- probability from sigmoid-function
- logit transformation
- Types of regression
  - simple logistic regression
  - Multinomial logistic regression

05 - K-Nearest Neighbour Classifier

- classify examples by assigning them the class of the most similar labeled examples
- very simple but extremely powerful
- Well suited for classifying tasks where the relationships between features are very complex and hard to understand
- training dataset → classified into several categories
- kNN identifies k elemts in the training dataset that are the "nearest" in similarity
- unlabelled test example is assigned to the nearest k cluster

06 - Naive Bayes Classifier

- naive means there is a strong independence assumptions between the given features
- relies heavily on condition probability

- we can decompose the conditional probability on bayes theorem
- choose class with highest probability
- Great for text classification

07 - Support Vector Machine

- Defines margin / boundary → between the data points in multidimensional space
- Goad: find a flat boundary ('hyperplane') that leads to a homogeneous partition of the data
- A good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class since in general the large the margin the lower the generalisation error of the classifier
- so we have to maximise the margin
- application - classifications or numerical predictions
- pattern recognition - disease classification, text classification and detecting rare events
- Linear separable problem
- Support vectors - the points from each class that are closest to the maximum margin hyperplane // each class have at least 1 support vector
- Convex Hull
- Vector geometry

08 - DecisionTrees

- A type of supervised learning approach
- mainly for classification but can be used for regression
- works fine for both categorical variables and continuous input as well
- split the data/population into two or more homogeneous sets based on significant splitter in input variables
- Root node, decision node, leaf nodes
- How to decide on decision trees

  - Gini index Approach
  - Calculating the information entropy
  - Algorithms based on variance reduction
- ID3 algorithm
  - Used to build the decision tree
  - Top-down greedy search of possible branches
  - Uses Shannon-entropy
    - H(X) = SUM(i in range(1,n) P(xi)log2P(xi)
    - For completely homogeneous data- entropy is 0
    - for equally divided dataset - entropy is 1
  - A branch with entropy more than 1 needs splitting
    - root node has the maximum information gain (entropy reduction)
    - leef nodes have entropy 0
  - Key problems
    - Every split it makes at each node is optimised for the dataset provided
    - Will rarely generalise well to other data set

09 - Random Trees Forest

- Decision Tree tends on over fit
  - every split it makes at each node is optimised for the dataset it is fit to
  - this splitting will process will rarely generalize well to other data
  - Unstable classifiers
- Two solutions
  - Pruning
  - Bagging
- Bias-variance tradeoff
  - bais- error from misclassification in the learning algorithms

        - High bias→ the algorithm misses the relevant relationships between features and target outputs
        - Underfitting
        - Erro due to model mismatch
    - variance - error from sensitivity to small changes in the training set
        - High variance → can cause oevrfitting
        - algorithm models the noise
        - variation due to training sample and randomisation
    - we are not able to optimise both bias and variance at the same time
        - low bias → high variance
        - low variance → high bias
    - Model complexity
- Random Forest Classifier
    - decorrelates the single decision trees that has been constructed
    - reduces variances even more when averaging trees
    - the number of features considered at a given split is approximately equal to the square root of the total number of features (for classification
    - Algorithm searches over a random sqrt(N) features to find the best one
- Pruning
    - grow a large tree and then prune it back to a smaller subtree
    - weak link prunning
    - reduce variance

10. Boosting

- can be used for classification and regression
- helps to reduce variance and bias

- bagging create multiple copies of the original data. it consists of several decision trees on the copies and combining all the trees to make predictions. We construct these trees independently.
- **boosting** - the decision trees are grown sequentially so each tree is grown using information from previously grown trees
- these trees are not independent from each other
- boosting is a sequential learning algorithm
- a weak learner is not able to make good predictions
- combining weak learners can prove to be an extremely powerful classifier
- by fitting small trees (decision stumps). we slowly improve the final result in cases when it does not perform well
- Next level is adaptive boosting algorithm

11 - Clustering

11.1 Principal component analysis (PCA)

- Unsurpervised learning
- PCA gives us al low dimensional representation of a dataset
- able to find linear combinations of features / variables that are mutually uncorrelated
- Linearly uncorrelated variables are the principal components
- Good for visualisation
- Can be done
    - eigenvalue decomposition of a data covariance / correlation matrix
    - singular value decomposition of a data matrix usually after mean centering
- To do
    - Read on
        - Eigenvalue decomposition
        - Data covariance

        - Correlation matrix
        - Singular value decomposition
    - More examples on PCA

11.2 K-means algorithms

- Very popular unsupervised learning algorithm in data mining
- Automatically divides the data into clusters / groupings of similar items
- Doesnt need a labelled dataset
- Problem - how could a computer possibly know where one group end and another begins?
- Elements inside Aa cluster should be very similar to each other, but very different from those outside.
- K-means clustering aims to partition **n** observations into **k** clusters in which each observation belongs to the cluster with the nearest mean
- Can be done with graph algorithm - construct the minimum spanning tree… and remove the last k edges
- NP-hard problem
- Lloyd-algorithm is very common nowadays
    - Initialize the centroids at random, these are the centers of a given cluster
    - Decided for every point in the dataset what centriod is the near to them
    - calculate the new means of every distinct clusters
        - run until convergence
- Finding k parameter
    - sometimes we have some a prior knowledge: we know how mand cluster we want to construct
    - without any a priori knowledge: **k** is approximately equal to the square root of n//2 where n is the number of elements in the dataset

  - Elbow method: we monitor the change of homogeneity within the clusters with different k values
  - It looks at the percentage of variance explained as a function of the number of clusters
    - one should choose a number of clusters so that adding another clusr does not give much better modelling of the data
  - we have to find the "elbow point" at a plot
- Advantages
  - Relies on simple principles to identify clusters
  - Flexible
  - Efficient
- Disadvantages
  - Not so sophisticated
  - Because it uses an element of random chance, it s not guaranted to find the optimal set of clusters
  - K parameter → we have to know in advance how many clusters we want to find
- Clustering vs Classification
  - clustering is different from classification or numerical predictions
  - classification / regression: the result is a mode that related features to an outcome
  - ***clustering creates new data***
  - Unlabelled examples are given a cluster label and inferred entirely from the relationship within the data
- Text clustering
  - Measure text similarity
  - Apply a clustering algorithm
    - usually k-means clustering but we can use any other machine learning approach
  - Tokenizing

    - split a given text into a set of words
    - document term matrix
- TF-IDF
    - term frequency - inverse document frequency vectors
    - handles weight of a given word **w** in a document **d**
    - tf(w) = number of times w appears in document d // total number of words in document d
    - idf = log(number of documents / number of documents that contain word w)

11.3 DBSCAN Algorithm

- Overview
    - DBSCAN - Density Based Spatial Clustering of Application with Noise
    - Data clustering algorithms such as K-means
    - Density-based → given a set of points in some space, it groups together points that are closely packed together
    - Very common clustering algorithm
- Algorithm
    - There are given points in the 2 dimensional space
    - Try to find every points → that are separated by a distance no more than a given e_epsilon (the threshold distance)
    - same clusters: we can hop from a given node to another by hopping no more than e_epsilon → the points are in the same cluster
- Advantages
    - Finds non-linearly separable clusters (arbitrarily shaped clusters)!!!
    - For K-means we have to specificy the number of clusters we want to find → here we do not need to do so
    - Very robust to outliers

    - Result does not depend on the starting conditions
    - Parameters: e_epsilon (distance threshold) + minimum number of neighbors
    - O(N logN) running time !!!
- Disadvantages
    - DBSCAN is not entirely deterministic
    - Border points that are reachable from more than one cluster can be part of either cluster depending on the order the data is processed
    - Relies heavily on a distance measure: Euclidean-measure. In higher dimesnions it is very hard to find a good value for e_epsilon
    - curse of dimensionality
    - if the data and scale are not well understood → choosing a meaningful distance threshold e_epsilon can be difficult