Övningstenta maskininlärning AI21

1. Beskriv vad k-folded cross-validation är för något och ge exempel på hur det kan användas.

Answers:

```
### k-fold cross-validation

One strategy to choose the best hyperparameter alpha is to take the training part
of the data and
1. shuffle dataset randomly
2. split into k groups
3. for each group -> take one test, the rest training -> fit the model -> predict
on test -> get evaluation metric
4. take the mean of the evaluation metrics
5. choose the parameters and train on the entire training dataset

Repeat this process for each alpha, to see which yielded lowest RMSE. k-fold
cross-validation:
- good for smaller datasets
- fair evaluation, as a mean of the evaluation metric for all k groups is
calculated
- expensive to compute as it requires k+1 times of training
```

k-Fold Cross-Validation

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
1. Take the group as a hold out or test data set
2. Take the remaining groups as a training data set
3. Fit a model on the training set and evaluate it on the test set
4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores

Importantly, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set 1 time and used to train the model k-1 times.

This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining k − 1 folds.

— Page 181, An Introduction to Statistical Learning, 2013.

It is also important that any preparation of the data prior to fitting the model occur on the CV-assigned training dataset within the loop rather than on the broader data set. This also applies to any tuning of hyperparameters. A failure to perform these operations within the loop may result in data leakage and an optimistic estimate of the model skill.

Despite the best efforts of statistical methodologists, users frequently invalidate their results by inadvertently peeking at the test data.

— Page 708, Artificial Intelligence: A Modern Approach (3rd Edition), 2009.

The results of a k-fold cross-validation run are often summarized with the mean of the model skill scores. It is also good practice to include a measure of the variance of the skill scores, such as the standard deviation or standard error.

Configuration of k

The k value must be chosen carefully for your data sample.

A poorly chosen value for k may result in a mis-representative idea of the skill of the model, such as a score with a high variance (that may change a lot based on the data used to fit the model), or a high bias, (such as an overestimate of the skill of the model).

Three common tactics for choosing a value for k are as follows:

- **Representative**: The value for k is chosen such that each train/test group of data samples is large enough to be statistically representative of the broader dataset.
- **k=10**: The value for k is fixed to 10, a value that has been found through experimentation to generally result in a model skill estimate with low bias a modest variance.
- **k=n**: The value for k is fixed to n, where n is the size of the dataset to give each test sample an opportunity to be used in the hold out dataset. This approach is called leave-one-out cross-validation.

The choice of k is usually 5 or 10, but there is no formal rule. As k gets larger, the difference in size between the training set and the resampling subsets gets smaller. As this difference decreases, the bias of the technique becomes smaller

— Page 70, Applied Predictive Modeling, 2013.

A value of k=10 is very common in the field of applied machine learning, and is recommend if you are struggling to choose a value for your dataset.

To summarize, there is a bias-variance trade-off associated with the choice of k in k-fold cross-validation. Typically, given these considerations, one performs k-fold cross-validation using k = 5 or k = 10, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance.

— Page 184, An Introduction to Statistical Learning, 2013.

If a value for k is chosen that does not evenly split the data sample, then one group will contain a remainder of the examples. It is preferable to split the data sample into k groups with the same number of samples, such that the sample of model skill scores are all equivalent.

2. Datasetet du ser här är inte linjärt separerbart, det har två olika klasser. Beskriv hur du skulle gå tillväga för att klassificera punkterna i datasetet med SVM.

In machine learning, **support-vector machines** (**SVMs**, also **support-vector networks**[1]) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis.

SVM is an extension of the support vector classifier. We use SVM to address the problem of possibly non-linear boundaries between classes, by enlarging the feature spaces using kernels.

We use instead a generalization of the inner product of two observataions in support vector classifer: K(x, xi). K refers to a kernel. A kernel is a function that quantifies the similarity of two observations.

A linear kernel will be used when support vector classifier is linear in the features, the linear kernel essentially quantifies the similarity of a pair of observations using pearson standard correlation.

A polynomial kernel of degree d, where d is a positive integer. Using such a kernel with d>1 in the support vector classifier algorithms leads a much more flexible decision boundary.

K(xi, xi´) =(1+sum xij xi´j)^d

It essentially amounts to fitting a support vector classifier in a higher-dimensional space involving polynomials of degree d, rather than in the original feature space.

When the support vector classifier is combined with a non-linear kernel such as polynomial kernel, the resulting classifier is known as a support vector machine.

Another choice is the radial kernel

K(xi, xi´) =exp(-v*sum(xij-xi´j)^2)

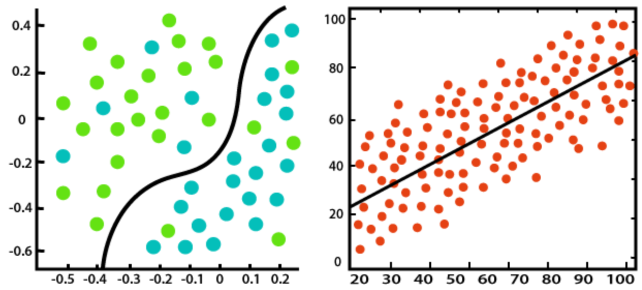
f(xi) = beta0+sum betai* K(xi, xi´)

Training observations xi´ that are far away from xi will play essentially no role in the predicted class label for xi. This means that the radial kernel has very local behavior, in the sense that only nearby training observations have an effect on the class label of a test observation.

3. Vad är stora skillnaden mellan regression och klassificering?

Regression and Classification algorithms are Supervised Learning algorithms. Both the algorithms are used for prediction in Machine learning and work with the labeled datasets. But the difference between both is how they are used for different machine learning problems.

The main difference between Regression and Classification algorithms that Regression algorithms are used to **predict the continuous** values such as price, salary, age, etc. and Classification algorithms are used to **predict/Classify the discrete values** such as Male or Female, True or False, Spam or Not Spam, etc.



Classification Regression

Classification:

Classification is a process of finding a function which helps in dividing the dataset into classes based on different parameters. In Classification, a computer program is trained on the training dataset and based on that training, it categorizes the data into different classes.

The task of the classification algorithm is to find the mapping function to map the input(x) to the discrete output(y).

Example: The best example to understand the Classification problem is Email Spam Detection. The model is trained on the basis of millions of emails on different parameters, and whenever it receives a new email, it identifies whether the email is spam or not. If the email is spam, then it is moved to the Spam folder.

Types of ML Classification Algorithms:

Classification Algorithms can be further divided into the following types:

- Logistic Regression
- K-Nearest Neighbours
- Support Vector Machines
- Kernel SVM
- Naïve Bayes
- Decision Tree Classification
- Random Forest Classification

Regression:

Regression is a process of finding the correlations between dependent and independent variables. It helps in predicting the continuous variables such as prediction of **Market Trends**, prediction of House prices, etc.

The task of the Regression algorithm is to find the mapping function to map the input variable(x) to the continuous output variable(y).

Example: Suppose we want to do weather forecasting, so for this, we will use the Regression algorithm. In weather prediction, the model is trained on the past data, and once the training is completed, it can easily predict the weather for future days.

- Simple Linear Regression
- Multiple Linear Regression

- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression

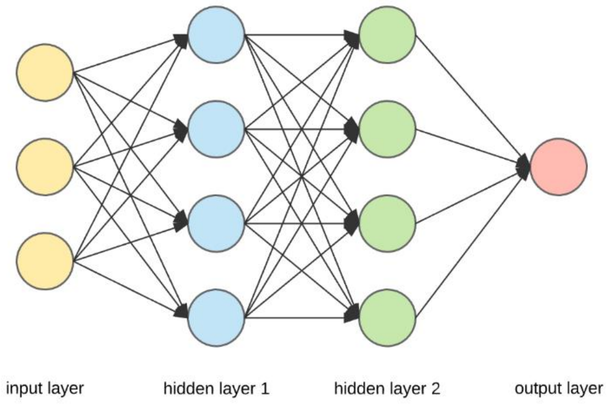
Difference between Regression and Classification

Regression Algorithm	**Classification Algorithm**
In Regression, the output variable must be of continuous nature or real value.	In Classification, the output variable must be a discrete value.
The task of the regression algorithm is to map the input value (x) with the continuous output variable(y).	The task of the classification algorithm is to map the input value(x) with the discrete output variable(y).
Regression Algorithms are used with continuous data.	Classification Algorithms are used with discrete data.
In Regression, we try to find the best fit line, which can predict the output more accurately.	In Classification, we try to find the decision boundary, which can divide the dataset into different classes.
Regression algorithms can be used to solve the regression problems such as Weather Prediction, House price prediction, etc.	Classification Algorithms can be used to solve classification problems such as Identification of spam emails, Speech Recognition, Identification of cancer cells, etc.
The regression Algorithm can be further divided into Linear and Non-linear Regression.	The Classification algorithms can be divided into Binary Classifier and Multi-class Classifier.

4. Beskriv begreppen input layer, hidden layer, outputlayer i ett multilayered perceptronnätverk.

A **multilayer perceptron** (**MLP**) is a class of feedforward artificial neural network (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training.[2][3] Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.[4]

Artificial Neural Network is computing system inspired by biological neural network that constitute animal brain. Such systems “learn” to perform tasks by considering examples, generally without being programmed with any task-specific rules.



The Neural Network is constructed from 3 type of layers:

Input layer — initial data for the neural network.

Hidden layers — intermediate layer between input and output layer and place where all the computation is done.

Output layer — produce the result for given inputs.

There are 3 yellow circles on the image above. They represent the input layer and usually are noted as vector X. There are 4 blue and 4 green circles that represent the hidden layers. These circles represent the “activation” nodes and usually are noted as W or θ. The red circle is the output layer or the predicted value (or values in case of multiple output classes/types).

Each node is connected with each node from the next layer and each connection (black arrow) has particular weight. Weight can be seen as impact that that node has on the node from the next layer.

5. Beskriv fördelarna att använda regulariseringsmetoder som och regularisering för linjär regression. Beskriv även hur man kan gå tillväga för att välja penalty parametern .

```
## Overfitting

Model too complicated, and fitted too much to the data. Complicated model (high
variance) risk to fit to noise in training data, which make them generalize
worse. Overfitting usually occurs when there is too small training set, and/or it
is not representative for testing data.


## Regularization technique:
```

Problem with overfitting was discussed in previous lecture. When model is too complex, data noisy and dataset is too small the model picks up patterns in the noise. The output of a linear regression is the weighted sum:
y = theta_0 + theta_1*x_1 + theta_2*x_2 + ... + theta_n*x_n , where the weights theta_i represents the importance of the i-th feature. Want to constrain the weights associated with noise, through regularization. We do this by adding a regularization term to the cost function used in training the model. Note that the cost function for evaluation now will differ from the training.
most regularization model requires scaling of data.

6. Använd KNN med och Euklidisk distansmått för att klassificera testpunkten .

## KNN

KNN or k-nearest neighbours is a supervised machine learning algorithm that can be used for both regression and classification. It calculates the distance between a test data point and all training data, find k training points nearest to the test data. Then it does majority voting to classify that test point to majority of the class of the training data points that are closest. For regression instead it takes an average of those k points that are closest.

In KNN it is absolute necessity to do feature scaling as the distance calculated using a distance metric can be very wrong if the features are in different scales.

7. Använd min-max normalization för att skala och
8. Företaget Xhopper driver en webshop som säljer saxar av olika slag och har samlat in data på kunders shoppinghistorik, features som datum, typer av produkter och en del andra beteenderelaterade features. De har hört att AI är coolt, och att k-means skulle kunna hjälpa dem att segmentera kunderna. Med några paket och typexempel har de lyckats skala datan, använda kmeans på datasetet och fått fram följande plots: Du gör praktik på företaget och de frågar dig som expert vad det är de fått fram, vad k-means är och vad de kan göra med resultatet.

## k-means clustering

- k-means clustering is an unsupervised learning algorithm, which means that there are no labels:

1. $k$ number of clusters are chosen
2. $k$ points are randomly selected as cluster centers
3. the nearest points to each cluster center are classified as that cluster
4. the center of the cluster is recalculated
5. repeat 3 and 4 until convergence

note that nearest points are defined by some distance metric

### Choose k

```
- plot an elbow plot of sum of squared distances (inertia in sklearn) and find
the an inflexion point to choose $k$, i.e. the point with significant lower rate
of change than before (note that this might be hard to find exact)
- domain skills, it's important to understand your dataset to find an adequate
$k$ and also equally important to be able to know what the clusters represent
- note that it is hard to find correct number of clusters, and it is here the art
and domain skills become more important


---
## Silhouette score

Note that it's usually not possible to plot the clusters, instead the silhouette
score in combination with elbow plot can help in determining clusters.

- silhouette score is a measure of cluster tightness

The silhoutte coefficient $S_i$ is calculated as
S_i = {b_i-a_i}/max{a_i, b_i},where
- a_i is mean distance between i and other points in the cluster it belongs to
- b_i is the mean distance from i to clusters it doesn't belong to

Calculate average silhouette score for different $k$ clusters in the clustering
algorithm, in this case KMeans.

- silhouette coefficient is between -1 and 1
- value 1 -> very compact clusters
- value 0 -> overlapping clusters
- value -1 -> worst value
```

1. Decision tree classification

In this algorithm, a classification model is created by building a decision tree where every node of the tree is a test case for an attribute and each branch coming from the node is a possible value for that attribute.

2. Random forest classification

This tree-based algorithm includes a set of decision trees which are randomly selected from a subset of the main training set. The random forest classification algorithm aggregates outputs from all the different decision trees to decide on the final output prediction, which is more accurate than any of the individual trees.

3. K-nearest neighbor

The K-nearest neighbor algorithm assumes that similar things exist in close proximity to each other. It uses feature similarity for predicting values of new data points. The algorithm helps grouping similar data points together according to their proximity. The main goal of the algorithm is to determine how likely it is for a data point to be a part of the specific group.