Aufbereitung Prüfungsfragen vor 2019

**General**

* Explain the goals of automatic classification. Detail how classification differs from other machine learning paradigms.
	+ Paradigms: Supervised Learning, Unsupervised Learning, Reinforcement Learning. In automatic classification, both inputs and outputs are necessary. The program learns the mapping of the input to the output (class) and then predicts the output of a given input. The goal is a high accuracy of the mapping of unknown inputs to the output classes. In unsupervised learning, no labels are present and the algorithm creates clusters by similarity. In reinforcement learning, the ideal behavior is determined within a specific context. Only rewards and penalty are available for certain actions and the system finds out automatically how to maximize the reward.

**Preprocessing**

* What is the difference between z-score and min-max, when to use which on which feature types?
	+ for neural networks is recommended normalization Min max for activation functions. To avoid saturation Basheer & Najmeer (2000) recommend the range 0.1 and 0.9. Z-score preserve range (maximum and minimum) and introduce the dispersion of the serie (standard deviation / variance). If you data follow a gaussian distribution, they are converted into a N(0,1) distribution and the comparison between series (probabilities calculation) will be easier, good if many outliers

**Feature Selection**

* PCA supervised feature selection method?
	+ FALSE, PCA combines similar (correlated) attributes and creates new ones. Superior to original attributes. Feature selection doesn't combine attributes. Just evaluates their quality, predictive power and select the best set. Benefit in PCA is that combination of N attributes is better than any individual attribute. Disadvantage is in harder explanation what exactly that PCA component means. You can combine both methods: Create PCA components and then run feature selection. In business consulting I'm not using PCA at all, just FS. Sacrificing a bit of predictive power to get more understandable model is my proven strategy.
* Name methods of feature selection
	+ Lasso regression (L1 regularization), PCA, Decision Trees (information gain), forward selection, backward elimination, X2

**KNN**

* when k-nn is used with 1-n encoding, min-max scaling is needed to perform euclid. Distance
	+ FALSE, although it handles distances, 1-n encoding always encodes in the range [0, 1]
* k-nn is recommended for large datasets
	+ FALSE, as no model is used and all data has to be located in the memory
* for k-nn categorical features should be normalized
	+ FALSE, because they don’t relate to each other. They are classifications
* What is a lazy learner? In which situations can it be beneficial?
	+ A learner without building a model with training data, like knn. Good to use if you are using an online learning system and don’t want to retrain your model after each new example.
* Classification example with kNN. Given a Dataset with 7 entries: Using knn(1), calculate the prediction accuracy using the first 6 entries and 2-fold.



* + Hints: Use the hamming distance to calculate the distance, i.e. binary comparison of variables. If equal, distance=0, else 1. 2-Fold: Split set into two.
* Using k-nn, determine the unknown class of the last item (assume k=3). Define an appropriate distance metric for the categorical variables.

**Naïve Bayes**

* example with 8 instances, perform
	+ naive bayes on 5 test examples
	+ calculate accuracy, precision and recall, zero frequency was optional to handle
* Klassifizierung der letzten beiden Samples mittels Naive Bayes



**1R**

* example with 8 instances, perform
	+ 1R on 5 test examples
	+ calculate accuracy, precision and recall, zero frequency was optional to handle
* Example with 1R – Klassifizierung der letzten Samples mit 1R



**Decision Trees**

* pruning is needed for underfitting
	+ Question not clear. TRUE, decision trees are a nonparametric machine learning algorithm that is very flexible and is subject to overfitting training data. This problem can be addressed by pruning a tree after it has learned in order to remove some of the detail it has picked up.
* Entropy max when all elements from same class?
	+ FALSE, if all elements are of the same class, entropy in min.

**Ensemble Learners**

* Describe the difference between bagging and boosting. Describe one algorithm that makes use of bagging.
	+ Both are ensemble methods, where a set of weak learners are combined to create a strong learner. Ensemble learning minimizes noise, bias and variance. Bagging and boosting consists of a pool of tress. The trees use new data by random sampling with replacement. In bagging, each observation has the same probability to appear in a dataset. In boosting, observations are weighted, i.e. some observations will be used more frequently. In bagging, each training step takes place in parallel, where each model is built independently, in boosting, the learners are built in sequence. The classification success of a learner is used to reinforce misclassified examples to make the algorithm better on them. Random Forests use bagging (<https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/>).
* RF heterogeneous ensemble learner?
	+ FALSE. RF are homogenous ensemble learner
* Describe random forests in detail + compare to  similar algorithms (e.g. 1R and  decision trees)
	+ Random forests consist of several weak learners, here decision trees, which form a homogenous (same type of learner) ensemble of learners. Each decision tree is prone to overfitting. Small changes of the data causes big differences in structure of the learner. Therefore, each decision tree is trained only on a variation of the training data, to get a big variety of trainers. Test data is passed to the RF and by majority voting, the correct classification is decided. Classifiers are independent, i.e. their results do not depend on other trees. Difference to decision trees: A decision tree is one element of the ensemble. 1R: The 1R represents the first split of a decision tree.
* What makes a RF random?
	+ The bagging, where data for each tree is generated with random sampling with replacement.
* Given a Dataset with 7 entries: Build a Decision Tree with your own separation criteria
	+ Hints: Use total class error to decide on which attribute shall be split.

**Hidden Markov Models**

* Describe HMM structure
* What is HMM?
* What are the three problems of HMM?
	+ The Evaluation Problem: what is the probability that the observations are generated by the model?
	+ The Decoding Problem: what is the most likely state sequence in the model that produced the observations?
	+ The Learning Problem: how should we adjust the model parameters A, B, pi in order to maximize p(observation|model)?

**Bayes Network**

* difference naive bayesian network to normal (bayesian) network
	+ In naïve bayes, there is no dependey between the features. All features are conditionally independent given the class. It is a special case of a Bayesian network.
* difference naive bayesian network to normal (neuronal) network
	+ Neural nets are highly structured networks bases, and have three kinds of layers - an input, an output, and so called hidden layers, which refer to any layers between the input and the output layers. Bayesian networks, is a probabilistic directed acyclic graphical model, a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies via a directed acyclic graph, e.g. probabilistic relationships between diseases and symptoms.
	+ In Bayesian networks the visual representation of graph that is vertices and edges have meaning- The network structure itself gives you valuable information about conditional dependence between the variables. With Neural Networks the network structure does not tell you anything.
	+ Bayesian networks represent independence (and dependence) relationships between variables. Thus, the links represent conditional relationships in the probabilistic sense. Neural networks, generally speaking, have no such direct interpretation, and in fact the intermediate nodes of most neural networks are discovered features, instead of having any predicate associated with them in their own right.
	+ Bayesian networks are generally simpler in comparison to Neural networks, with many decisions about hidden layers, and topology and variants.
	+ Bayesian networks like Naive bayes assumes that all input variables are independent. If that assumption is not correct, then it can impact the accuracy of the Naive Bayes classifier. An ANN with appropriate network structure can handle the correlation/dependence between input variables.
* Draw an Bayes Net with 5-7 Nodes. Include serial, diverging, and converging connection. D-separation?
	+ Serial: A->B->C, D-separation if B is instantiated
	+ Diverging: A<-B->C, D-separation if B is instantiated
	+ Converging: A->B<-C: D-separation if B is not instantiated or if the descendent of B is not instantiated
* what can be used to create a Bayesian networks? compare it
	+ Human experts can create the network or the network can learn from data. A combination can be used, e.g. structure by human and probabilities. Human experts are good at structures,. Computers are better at probabilities. Computer use local searches like tabu search, hill climbing or simulated annealing to find structures. For tabu search and hill climbing, all neighbors are created and the best one is selected. It can be to remove an edge, add an edge or reverse its direction.
* Propose a Bayesian Network, which you think will be appropriate for this domain (the network should not be a simple Naïve Bayes). Explain shortly why you selected exactly this network



* + Calculate some of the conditional probabilities in your network
	+ Based on your network, classify the last data sample from the table above
	+ Suppose that node Calories is instantiated. Which nodes of your network would be d-separated in this case?
* Bayesian Network -> Explain a) Inference b) Variable Elimination c) d-separation
	+ Inference: The process of querying the network based on some evidence to get conditional probabilites of variables of interest. What’s the whole probability

distribution over variable X given evidence e, P(X | e)? (<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-825-techniques-in-artificial-intelligence-sma-5504-fall-2002/lecture-notes/Lecture16FinalPart1.pdf>)

* + Variable Elimination: Replace variables that are no part of the query with a function and in that way eliminate irrelevant variables. The contribution of the variable is expressed as a function
	+ D-Separation: Serial: A->B->C, D-separation if B is instantiated,  Diverging: A<-B->C, D-separation if B is instantiated,  Converging: A->B<-C: D-separation if B is not instantiated, or if the descendent of B is not instantiated. If D-separated, the nodes are conditionally independent.

**SVM**

* Compare SVM with perceptron  what  is common, what differs
	+ Perception is an online learner, where the model can be updated after each sample. SVM learns all samples at once
	+ SVM maximizes the margin, the perception finds a sufficient separation
	+ SVM with an RBF kernel creates as many dimensions as samples, having an O(n^2) scaling. Perceptron does not grow in size
	+ SVM project the problem space into a high dimensional space to make a linear separation there. The perception makes a non-linear separation
	+ Both use something equivalent to an L2 regularization term.
	+ The linear kernel of the SVM works almost equal as the linear logistic regression.
	+ In both systems, the data have to be scaled/normalized
	+ SVM uses 1 vs all or 1 vs 1 classification, i.e. only binary classification, while the perceptron can classify all on the same time
* Describe in detail the ideas and concepts of Support Vector Machines; describe all algorithm variations discussed in the lecture.
	+ The basic idea of the SVM is to project the data into a higher feature space, as they are not linear separable in the current feature space. This is done by using a kernel, e.g. the radial base kernel e^(-gamma\*||x-xcenter||^2). The gamma is a meta parameter. The higher the gamma, the more influence does each sample has on the margin, i.e. large gamma leads to high bias and low variance, underfitting. In the higher dimensional space, the data is separated linearly. This is done by maximizing the margin, i.e. finding the margin with the best separation between classes. The parameter C is used to consider misclassifications. It is used as an 1/C regularization term. The higher the C, the less misclassifications are allowed, which may lead to overfitting. After classification, the data is transformed back into the original feature space and the decision line is not linear anymore. Algorithm variations: Linear, poly, rdf, sigmoid kernel. Gamma and C as parameters to tune the model. High C->overfit, high gamma->underfit
* What re the advantages and disadvantages of SVMs?
	+ + High accuracy
	+ + Any complex problem solvable with the kernel trick
	+ + Good at unstructured data
	+ + Unlike NN, problem not solved for a local minimum
	+ + Good generalization
	+ – Does not cope well with many features
	+ – Does not scale well, O(n^2), takes long time on big datasets
	+ – Hard to tune the parameters
	+ – Model hard to visualize and to interpret
* How do they differ from other classifiers using the basic principle of SVMs?
	+ Higher dimensional space projection, max margin classifier, kernel trick, good generalization capability (compared to DT, robust to outliers)

**Linear Regression**

* difference gradient descent and normal equation? which would you use on a lin. reg. example?
	+ Gradient descent heuristic, normal equation analytical solution, normal equation computational intensive, for smaller datasets, gradient descent on bigger datasets
* Draw Gradient Descent pseudo code
	+ 1. Initialize weights w randomly, z=w0\*1 + w1\*x1 + w2\*x2
	+ 2. Feed forward, i.e. calculate z(x) for a sample.
	+ 3. Calculate the error through the loss function, L=1/(2m)\*sum((y-z(x))^2) + 1/(2m)\*sum((W)^2)
	+ 4. Calculate the gradients to find out how much does a certain weight wi affect the result, dL/dw for each weight, dL/dw1=1/m\*sum(y-h(x))\*x1
	+ 5. Calculate new weights: w=w-alpha\*dL(w).
	+ 6. Repeat until there is no improvement of L or max iterations reached
	+ (<https://www.freecodecamp.org/news/understanding-gradient-descent-the-most-popular-ml-algorithm-a66c0d97307f/>)

**Perceptrons, MLP, neural networks**

* backpropagation is a technique used with perceptrons (single layer)
	+ TRUE?, because it applies on multilayers
* Compare to SVM
	+ Already covered

**Deep neural networks**

* What is Deep Learning?
	+ It is a subset of machine learning, usually neural networks. The network consists of stacked layers, where each layer has a certain function, e.g. to extract features (convolutional layer) from the data, or compress the input features (sub-sampling layer) and finally a fully connected layer. Through functions by neural layers, hardcoded tasks are minimized.
* How does it compare to "traditional" Machine Learning?
	+ Traditional machine learning: Data->Handcrafted features->Mapping features to Output->Output
	+ Deep Learning: Data->Simple features generated->Complex features generated->Mapping features to output->Output

**Meta Model Optimization**

* Name some model based attributes in meta learning
	+ Number of attributes, Number of classes, Ratio examples/attributes, average class entropy, signal to noise ratio

**Evaluation**

* paired t-tests are used when dealing with train/test splits
	+ TRUE, paired t-tests are used to see if two samples came from the same distribution. Are the errors similarly distributions, then there is no significant difference in performance of algorithms. t=(median1-median2)/(sigma1^2+sigma2^2)
* McNemar's Test is used for significance testing?
	+ TRUE, I used this test for the comparison of two classification algorithms. Based on McNemar’s test results you can decide which algorithm performs (statistically) better regarding accuracies, which are used in the formula. For the value X, calculated by McNemar’s test, holds: if X=3.84 (or lower), when algorithms have the same error (with probability ~95%). Otherwise, if X>3.84, when the performance of one algorithm is higher. McNemar’s test is used for testing generalisation ability, e.g. of a new designed classifier. Further, it is well-suitable for the situation, where training set and validation set are predefined. <https://www.researchgate.net/post/What_is_McNemars_Test_Telling_Us>
* Describe 3 metrics to evaluate classification algorithms and explain how these metrics relate to each other.
	+ Accuracy=(TP+TN)/number of samples, Precision=TP/(TP+FP), Recall=TP/(TP+FN). Accuracy is how close the whole classification is to the true classification. Precision (exactness) is how well a certain class is correctly predicted. Recall (sensitivity) is how well all instances of a class are classified.
* Explain the importance of statistical significance tests.
	+ Important to assure that “better” has a statistical base, which adds more power to the claim.