EXAMINATION QUESTIONS

Faculty:	**Science and Technology**	
Examination in:	**DAT200** *Course code*	**Applied Machine Learning** *Course name*
Time for exams:	Friday, 04.01.2019 *Day and date*	9:00 – 12:30 (3.5 hours) *As from – to and duration of examinations (hours)*
Course responsible:	Kristian Hovde Liland (6723 1624) and Oliver Tomic *Name*	

Permissible aids:

A1: no calculator, no other aids

The exams papers includes: _____
Number of pages incl. attachment

If the examination consists of several parts, information must be given as to how much each part will count toward the grade

Course responsible: Kristian Hovde Liland and Oliver Tomic

External examiner: Bjørn-Helge Mevik

Exercise 1 (13 points in total)

Support vector machines (SVM) are built on simple principles, but can yield powerful classifications, especially when extended with various kernels.

a) (7 points)

Explain the basics of SVMs. What are the support vectors? How does changing the width of the margin between classes affect modelling? s 76-88

* Optimisation objective is to maximise the margin. (2p)
* The margin is defined as the distance between the separating hyperplane (decision boundary) and the training samples that are closest to this hyperplane, which are the so-called support vectors. (3p)
* The rationale behind having decision boundaries with large margins is that they tend to have a lower generalization error whereas models with small margins are more prone to overfitting. (2p)

b) (6 points)

How are kernels used with SVM? Explain the kernel trick and illustrate symbolically.

* Using a kernel function, compute distance between all pairs of sample vectors in training step (instead of mapping all samples into a non-linear higher dimensional space, use linear classifier and then map back again). This results in a square kernel matrix that is the input to the SVM (2p)
* Compute distance between all support vectors and test data in test step (2p)
* Write up kernel matrix of pairs. (2p)

Exercise 2 (20 points in total)

Exploring of a data set and validation of models built on it are important aspects of machine learning.

a) (6 points)

Which two/three parts do we usually split our data into? What are the roles of each part? How would you handle a situation with few samples?

* training data, validation data and test data
* training: use Xtrain and ytrain to train model (2p)
* validation: use Xvalid and yvalid to measure performance of model and tune hyperparameters (2p)
* test: use Xtest and ytest to measure performance of model on unseen data (2p)

b) (10 points)

Describe the concept of nested cross validation. What is it used for? Draw/sketch a 5x2 cross validation.

* Nested cross validation is used for selection among different machine learning algorithms (and thus provides a close to unbiased error estimation). (1p)
* outer loop: k-fold cross validation to split data into training and test folds (3p)
* inner loop: k-fold cross validation on training fold to select model (2p)
* after model selection, evaluate performance on test data (2p)
* Sketch of 5x2 cross validation (2p)

c) (4 points)
What are learning curves used for? Sketch various scenarios of how learning curves may look, and recommend actions based on the curves? s 196
* Learning curves are for diagnosing bias and variance. (2p)
* Sketch plot p. 196 (2p)

Exercise 3 (10 points in total)
Bagging and boosting are two basic techniques in ensemble learning.

a) (5 points)
Explain the principles of bagging. (s. 240)
* All individual classifiers are not trained on same training data (as in majority vote classifier), but on bootstrap samples (random samples with replacement) from that training data.
* Classifiers are fit to the bootstrap samples
* Predictions are combined using majority voting

b) (5 points)
Explain the principles of boosting. (s. 246)
* Draw a random subset of training samples d_1 without replacement from training set D to train a weak learner C_1.
* Draw a second random training subset d_2 without replacement from the training set and add 50 percent of the samples that were previously misclassified to train a weak learner C_2
* Find the training samples d_3 in training set D, which C_1 and C_2 disagree upon, to train a third weak learner C_3 .
* Combine the weak learners C_1, C_2, and C_3 via majority voting.

Exercise 4 (12 points in total)
In a regression where we want to predict exam grades based on hours spent working on compulsory assignments there are some large outliers.

a) (8 points)
RANSAC is an algorithm that reduces the influence of outliers. Explain how the iterative RANSAC algorithm works. (s 325)
1. Select a random number of samples (a subset of the data) to be inliers and fit the model. (1p)
2. Test all other data points against the fitted model and add those points that fall within a user-given tolerance to the inliers. (2p)
3. Refit the model using all inliers. (1p)
4. Estimate the error of the fitted model versus the inliers. (2p)
5. Terminate the algorithm if the performance meets a certain user-defined threshold or if a fixed number of iterations were reached; go back to step 1otherwise. (2p)

b) (4 points)
We want to know if the hours spent working on compulsory assignments is enough to get a good model. What tools do we have to search for un-modelled phenomena in our model? And how can we correct for these? (s. 328 + slides?)
* A useful tool would be the residual plot, where residuals are plotted versus predicted values. (1p)
* Residual plots can help detect non-linearity and outliers and whether errors are randomly distributed (3p)

Exercise 5 (5 points in total)
Regularisation is one approach to tackle overfitting by additional information. The most popular approaches to regularised linear regression are Ridge Regression, Least Absolute Shrinkage and Selection Operator (LASSO) and Elastic Net.

Explain the differences between Ridge Regression, LASSO and Elastic Net.
The difference between Ridge Regression, LASSO and Elastic Net is that they use different regularisation terms that are added to the cost function. Ridge Regression adds the L2 penalty term, which are simply the squared sums of the weights. This leads to less extreme weight values as compared to a model with an unregularised cost function. LASSO adds the L1 penalty term, which is the absolute values of the terms. The L1 term can lead to sparse models and as such works as a feature selection technique. Elastic Net adds both L1 and L2 penalty terms and in this way finds a compromise between Ridge Regression and LASSO.

Exercise 6 (13 points in total)
We want to search for customer groups in a data where we have access to wages and reported interest in electronic gadgets.

a) (5 points)
Explain the process of hierarchical clustering with complete linkage, assuming the distance measure: 1-correlation. Make a simple sketch of your explanations.
1. Compute the distance matrix of all samples.
2. Represent each data point as a singleton cluster.
3. Merge the two closest clusters based on the distance between the most dissimilar (distant) members.
4. Update the similarity matrix.
5. Repeat steps 2-4 until one single cluster remains.

b) (8 points)
What is cluster inertia, and how would you calculate it? (s. 350)
Cluster inertia is the within-cluster Sum of Squared Errors (SSE). (1p)

$$SSE = \sum_{i=1}^{n}\sum_{j=1}^{k} w^{(i,j)} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(j)} \right\|_2^2$$

Computation:
1. For one cluster, compute difference between each sample belonging to this cluster and its

cluster centroid. (3p)
2. Square all these differences and compute sum. (2p)
3. Repeat 1. and 2. for each cluster. (1p)
4. Compute sum across sums of each cluster. (2p)

Exercise 7 (12 points in total)

When working with decision trees the following formulae are important: (s. 90)

$$I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j) \qquad \sum_{i=1}^{c} p(i|t)\left(1 - p(i|t)\right) = 1 - \sum_{i=1}^{c} p(i|t)^2$$

a) (8 points)

Explain the principle of "information gain" with regard to the above formulae. What do the various symbols in the formula signify?

In order to split the nodes at the most informative features, we need to define an objective function that we want to optimise via the tree learning algorithm. Here, our objective function is to maximize the information gain at each split. The information gain is simply the difference between the impurity of the parent node and the sum of the child node impurities — the lower the impurity of the child nodes, the larger the information gain. There is no impurity in single class nodes. (5p)

D_p: Data in parent node
D_j: Data in *j*th child node
I: Impurity measure
N_p: total number of samples in parent node
N_j: number of samples in *j*th child node
(3p)

b) (4 points)

Explain the principle of "gini impurity" with regard to the above formulae. What do the various symbols in the formula signify?

Gini impurity can be understood as a criterion to minimise the probability of misclassification. Gini impurity is maximal if the classes are perfectly mixed. (2p)

p(i|t): proportion of samples that belong to a class c for a particular node t
(2p)

Exercise 8 (15 points in total)

Using the code below, describe briefly the role of each step of the analysis being performed and precisely what the choice of parameters in each line means (1 point for step 1, 2 points for each of the other steps).

```
# Step 1:
# Import all relevant libraries
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV

# Step 2:
# Read standard CSV-data using Pandas
df = pd.read_csv('https://wineresearch.com/redwine.data')

# Step 3:
# Subset df, OneHot encoding (non-collinear) of 'wine_berry', concatenate
horizontally
X = pd.concat([df.iloc[:, 2:20], \
               pd.get_dummies(df['wine_berry'],\
               drop_first=True)], axis=1).values
y = df.iloc[:, 0].values

# Step 4:
# Split train test 70:30, stratify on the classes in y (equal proportions in
segments) with fixed random state
X_train, X_test, y_train, y_test = \
    train_test_split(X, y,
                     test_size=0.30,
                     stratify=y,
                     random_state=1)

# Step 5:
# Pipe with centering and standardisation, plus LDA transformation, plus Logistic
regression (fixed random state)
pipe_lr = make_pipeline(StandardScaler(),
                        LDA(n_components=2)
                        LogisticRegression(random_state=1))

# Step 6:
# Set up optimisation grid for C
param_range_C = \
    [0.0001, 0.001, 0.01, 0.1, 1.0, 10.0, 100.0, 1000.0]
param_grid = [{'logisticregression__C': param_range_C]

# Step 7:
# Tune hyperparameters using cross-validated gridsearch with 10 segments, single
core processing
gs = GridSearchCV(estimator=pipe_lr,
                  param_grid=param_grid,
                  cv=10,
                  n_jobs=1)
gs = gs.fit(X_train, y_train)

#Step 8:
# Predict test set using best_estimator and compute accuracy
result = gs.best_estimator_.score(X_test, y_test)
```