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?

b) (6 points)
How are kernels used with SVM? Explain the kernel trick and illustrate symbolically.

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?

b) (10 points)
Describe the concept of nested cross validation. What is it used for? Draw/sketch a 5x2 cross validation.

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?

Exercise 3 (10 points in total)

Bagging and boosting are two basic techniques in ensemble learning.

a) (5 points)
Explain the principles of bagging

b) (5 points)
Explain the principles of boosting.

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.

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?

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.

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.

b) (8 points)
What is cluster inertia, and how would you calculate it?

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?

b) (4 points)
Explain the principle of "gini impurity" with regard to the above formulae. What do the various symbols in the formula signify?

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 ...
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 ... using ...
df = pd.read_csv('https://wineresearch.com/redwine.data')

# Step 3:
# ...
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 ..., stratify on ... with ...
X_train, X_test, y_train, y_test = \
    train_test_split(X, y,
                     test_size=0.30,
                     stratify=y,
                     random_state=1)

# Step 5:
# Make Pipeline ... with ...
pipe_lr = make_pipeline(StandardScaler(),
                        LDA(n_components=2)
                        LogisticRegression(random_state=1))

# Step 6:
# Set up . . .
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:
# Fit . . . using . . .
gs = GridSearchCV(estimator=pipe_lr,
                  param_grid=param_grid,
                  cv=10,
                  n_jobs=1)
gs = gs.fit(X_train, y_train)

#Step 8:
# . . .
result = gs.best_estimator_.score(X_test, y_test)
```