Exam for Data and Things (F2023)

General info:

- 20 minutes for each student, including time for assessment and feedback
- You draw a number from 1 to 9, corresponding to one of the exercises handed in (Part One below). You then present it and we ask questions about it. This part takes approximately 10 min.
- For the last 5 min we will ask questions from other topics in the entire course. A list of possible questions is given in Part Two below. However, it does not mean that each question in the exam will be the same as one of those below.

Hand-in info:

- The hand-in must happen through eksamen.ruc.dk before March 23 at 10:00.
- You should hand in answers to the exercises in part one below in a Jupyter Notebook. (A template is provided in Moodle)
- You may hand in in groups, which Eksamen.ruc.dk should be set up for.
- You should hand in one Jupyter Notebook with answers to each question. The answers should be clearly marked using sections with the number and title as below. Each section should be able to run separately, including importing of modules used.

Part One: Hand-in exercises for the exam

1. EDA and data cleaning (Lecture 2 & 5)

Make an Exploratory Data Analysis (EDA) and data cleaning of the "titanic_survival_data.csv" dataset from Lectures 5 and 6, including dealing with outliers and missing values.

2. Classification (Lecture 3 & 4)

Combine the exercise from Lecture 3 with exercise 2 from Lecture 4 into one, and construct some classification models to predict if a passenger would survive or not in the Titanic dataset.

a) You should have (1) decision tree, (2) random forest, and (3) KNN. You may also vary the configuration of each model type.
b) You should do necessary data preprocessing (e.g., missing value fill-in, and data scaling if needed for a classifier).
c) You should also do cross-validation of your models.
d) Plot the ROC with AUC for each model you implement.

3. Regression (Lecture 6)

Train a multiple linear regression, a random forest model, and an AdaBoost model on the "boston_housing_data.csv" dataset from Lectures 5 and 6 and remember to do train-test split as well as other necessary pre-processing dataset.

4. Clustering (Lecture 7 & 8)

Exercise 2 (both 2.1 and 2.2) from Lecture 7 and exercise 1 from Lecture 8.

5. Key-value stores (Lecture 9)

Exercise 1 from Lecture 9.

6. Deep learning (Lecture 10)

Train a deep neural network to predict if a passenger would survive or not in the Titanic dataset and remember to do train-test split as well as other necessary pre-processing dataset.

7. MapReduce (Lecture 13)

All exercises from Lecture 13.

8. Time Series Analysis (Lecture 14 & 15)

Do a time series analysis of the Copenhagen ice cream dataset ("cph_ice_cream_searches.csv") from Lectures 14 and 15.

9. IoT (Lecture 17)

All exercises from Lecture 17.

Part Two: Possible exam questions for other topics

1. Parallel databases (Lecture 12)

- What is IO parallelism? Explain at least one of Round-Robin partitioning, Hashing partitioning and Range partitioning.
- Explain how Range-Partitioning Sort or Parallel External Sort-Merge works.
- What is parallel join? Explain at least one parallel join algorithm from Partitioned Parallel Join, Fragment-and-Replicate Join, Partitioned Parallel Hash Join and Parallel Nested Loop Join.

2. Visualization and storytelling with data (Lecture 14 & 15)

- Explain the 6 key lessons from storytelling with data

3. Distributed databases (Lecture 16)

- Explain the two phase commit protocol for distributed transactions.
- Explain the Semijion strategy for a join of two relations that are stored in two sites respectively.

4. MLOps (Lecture 18)

- What is MLOps and why do it?
- What are some of the things we want to monitor for our machine learning models?
- What are the 6 principles of Infrastructure as Code?

5. Ethics (Lecture 19)

- What ethical issues or discussions can be sparked by the Amazon case?
- What is the COMPAS case about and to what does it illustrate with respect to fairness of machine learning algorithms?
- What can make machine learning algorithm biased?