ML Project

Alex(ander) Jung
Assistant Professor for Machine Learning
Department of Computer Science
Aalto University

Learning Goals

- modelling real-life as ML problems
- try out concepts taught in lectures
- writing of a scientific report ("paper")
- team work

Timeline

- Fr., 12.08.: each group presents problem formulation
- Sat., 27.08: each group submits project report

Friday 12.08

- each group presents problem formulation
- use the following template slides

Project Title

Project Team

Student 1, Student 2, Student 3

Data

- data points are …
- as features we might use …
- the label of a data point is
- data source:

Loss

- explain the ultimate performance metric that you want to optimize
- benchmark/baseline ?

Candidate Models

- justification
- Python class:
- challenges?

Project Report

to be submitted by 27.08.2022

Required Outline

- 1. Introduction
- 2. Problem Formulation/Setting
- 3. Method
- 4. Results
- 5. Conclusion
- References
- Appendix

1. Introduction

- explain the real-life application/scenario
- discuss the plan of the paper
  - 1-2 sentences explaining content of each section
  - explain how sections relate to each other

2. Problem Formulation

- explain the meaning of data points
- explain what features are used
- explain the label (quantity of interest)
- discuss useful loss function(s)
- discuss benchmark/baseline levels (if available)

3. Method

- discuss data gathering (nr. of datapoints, pre-processing)
- discuss chosen models (e.g., linear maps, dec. trees…)
- discuss hyper-parameters of training algorithms (step-size)
- discuss model validation technique (splitting strategy?)

4. Results

- compare training and validation errors for all models
- which model do you choose finally and why?
- report and discuss the test-set error of the final model

5. Conclusion

- interpret the train, val and test errors
- how close are these to a benchmark/baseline
- discuss limitations of the implemented models
- can you think of possible improvements? which one?

References

…

…

…

Appendix

```python
# Code source: Jaques Grobler
# License: BSD 3 clause

import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score

# Load the diabetes dataset
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)

# Use only one feature
diabetes_X = diabetes_X[:, np.newaxis, 2]

# Split the data into training/testing sets
diabetes_X_train = diabetes_X[:-20]
diabetes_X_test = diabetes_X[-20:]

# Split the targets into training/testing sets
diabetes_y_train = diabetes_y[:-20]
diabetes_y_test = diabetes_y[-20:]

# Create linear regression object
regr = linear_model.LinearRegression()
```

Motivation

- top teams will be posted on school site
- support for developing a conference submission to

https://2023.ieeeicassp.org/

(travelling costs will be covered by Aalto)