**Data Mining – Lecture Notes:**

Lecture 1,2)

Supervised Learning

EDA – Exploratory Data Analysis

EDA – Feature Selection

* Pool of attributes (columns) and instances (rows), but which ones are the most relevant for my purpose. Eliminate redundant and insignificant variables (avoid noise and increase transparency and performance).
* Identify core attributes and choose top-down or bottom up approach
* Top-down (backward): Remove attribute by attribute and discuss impact on model performance
* Bottom-up (forward): Add attribute by attribute and see performance increase
* What about impact of the combination of unimportant attributes

EDA – Feature Transformation

* Normalization/Standardization
* Aggregation of attributes (PCA, first principal component, select number based on elbow of the graph)
* No missing values

Classification algorithm:

Paperpile

**Tutorial 1 🡪 temporal data mining**

Brimag\_Exam (canvas channel)

Google Scholar

Time data processing in Pandas

Test, Validate, Test

Export figures as pdf

Make GitLab ready

Task1: Deal with preparing data for time-series usage

Data has been used, temporal data mining, aggregation methods, target window

Instance based dataset

Historical data can be used up to from different length

Baseline model, mood of the next day is mood of yesterday

Carefully read the instruction for academic writing

Explain rationales how solutions are developed

No right or wrong, just explain

<https://github.com/Dantesean/dmt-1>

<https://github.com/Dantesean/dmt-1/blob/master/DMT%20A1%20Preprocess.ipynb>

<https://github.com/Dantesean/dmt-1/blob/master/cleaning.ipynb>

<https://github.com/Dantesean/dmt-1/blob/master/DMT%20Assignment%201_Dante.ipynb>

<https://medium.com/data-science-at-microsoft/introduction-to-feature-engineering-for-time-series-forecasting-620aa55fcab0>

<https://www.bi4all.pt/en/news/en-blog/supervised-machine-learning-in-time-series-forecasting/>

**Lecture 3**

Instance based learning

* See explanation for thesis

Recurrent networks

* Allow for some form of memory
* Time-series based

Neural network

* Learn by adjusting weights
* In case of wrong classification, move & turn the ‘line’ a bit towards that point

Back-propagation

* Propagate change backwards from the output in case we made a classification error on the training data
* Make updates to the weights based on how big the mistake was & how much the error contributed to the output

Gradient descent

* How to find the minimum of the error function
* Graphically: Gradient say where we should go in order for this error to go down

Association rules

* Does the combination of spaghetti & sauce also make people want to buy cheese
* Support & confidence
* Finding frequent item sets