Decision trees (<https://towardsdatascience.com/a-guide-to-decision-trees-for-machine-learning-and-data-science-fe2607241956>)

Data preparation

* Categorical data
  + mean encoding
  + categorical features into numeric attributes (only works if ordinal)
  + one-hot encoding (expensive)
* Ways to reduce overfitting
  + Pruning: When you remove sub-nodes of a decision node, this process is called **pruning**. The opposite of pruning is **splitting**.
  + min\_samples\_leaf (bike sharing hatte bei ~100 bei cross validation ~128, ohne min\_samples\_leaf aber jedoch sogar weniger)
  + max depth (
    - bike sharing max depth 5-10: 130-140   
       max depth ~30: 125-130  
      -> braucht nicht so hohe max depth
    - student performance war mit max\_depth ~ 3,4 bei [numeric\_attributes] schon sehr gut bei cross validation

-> bei online Abgabe nicht unbedingt!!! 5.6 statt 5.0 bei voller max\_depth

* + max\_features: use all for better fitting, but may take longer
* Classification: gini & entropy waren ~ gleich, wobei gini ein bisschen besser war
* Pros (<https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/>)
  + easy to interpret since it is visual

fastest way to explore data as well as identifying the most significant variables

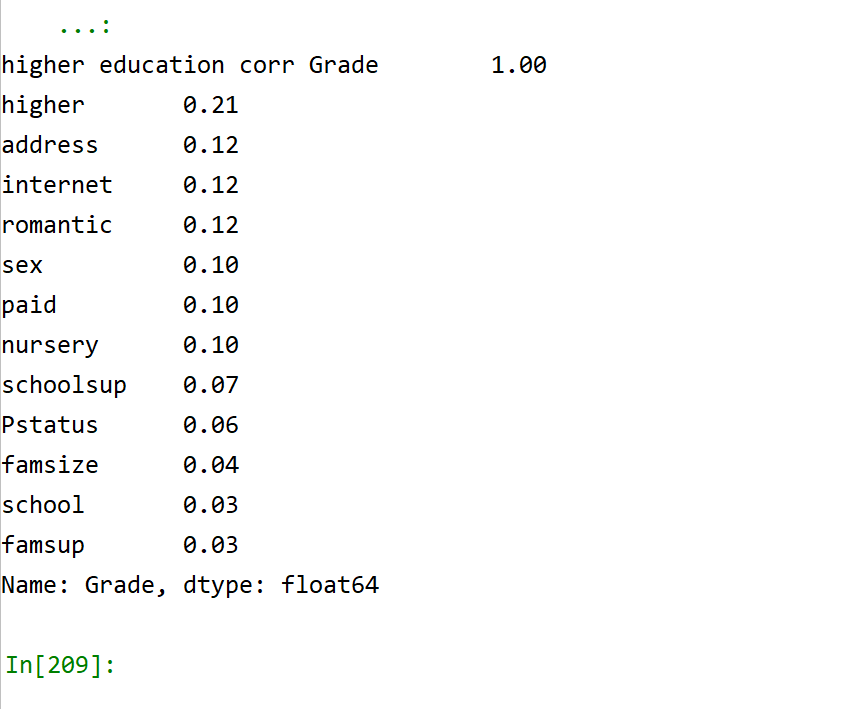
* + little data preparation, less influenced by outliers and missing values
  + data type is not a constraint
  + As each feature is processed separately, and the possible splits of the data don’t depend on scaling, no pre-processing like normalization or standardization of features is needed for decision tree algorithms. In particular, decision trees work well when we have features that are on completely different scales, or a mix of binary and continuous features.
* Cons (<https://medium.com/cracking-the-data-science-interview/decision-trees-how-to-optimize-my-decision-making-process-e1f327999c7a>)
  + overfitting -> dimensionality reduction (PCA)
  + vulnerable to becoming biased to classes in the majority -> class balancing (class weight, sampling, specialized loss function)
  + not fit for continuous variable
  + decision trees generally do not have the same level of predictive accuracy as other approaches, since they aren’t quite robust. A small change in the data can cause a large change in the final estimated tree. Even with the use of pre-pruning, they tend to overfit and provide poor generalization performance.

<https://pbpython.com/categorical-encoding.html> :

* One Hot Encoding: viele neue columns
* category encoding: Kategorien beinhalten nicht unbedingt ratios (Fjob, MJob) aber bei Student Performance bei manchen schon (Medu, Fedu)

Student performance

* some data is already encoded -> traveltime, studytime -> hours amount to [1:4]
* nominal (Kategorie:
  + sex, school, paid
* ordinal (klein, mittel, groß)
  + Medu, Fedu (0:none, 4:higher education)
  + famsize
  + famrel, freetime, goout, Dalc, Walc,
* interval (Einheiten)
* ratio (mit Nullpunkt)
  + age,
* age and id correlated -> younger ones asked first
  + thus since age and Grade are a bit correlated, Grade is also correlated with id
* Mother education more important than father (second highest correlation)
* Take attributes
* Used Decision Tree Classifier just to see the difference
  + -> actually was better with cross validation for a while (~ 4.0-4.3 mit online 5.0)
* <https://www.dremio.com/tutorials/analyzing-student-performance-with-dremio-python/>

Daten/Werteanalyse  
binomial data

Reihenfolge:

1. Kategorische Variablen Korrelation herausfinden
2. Methode schreiben um für verschiedene Attribute, und verschieden Parameter die beste Möglichkeit zu finden