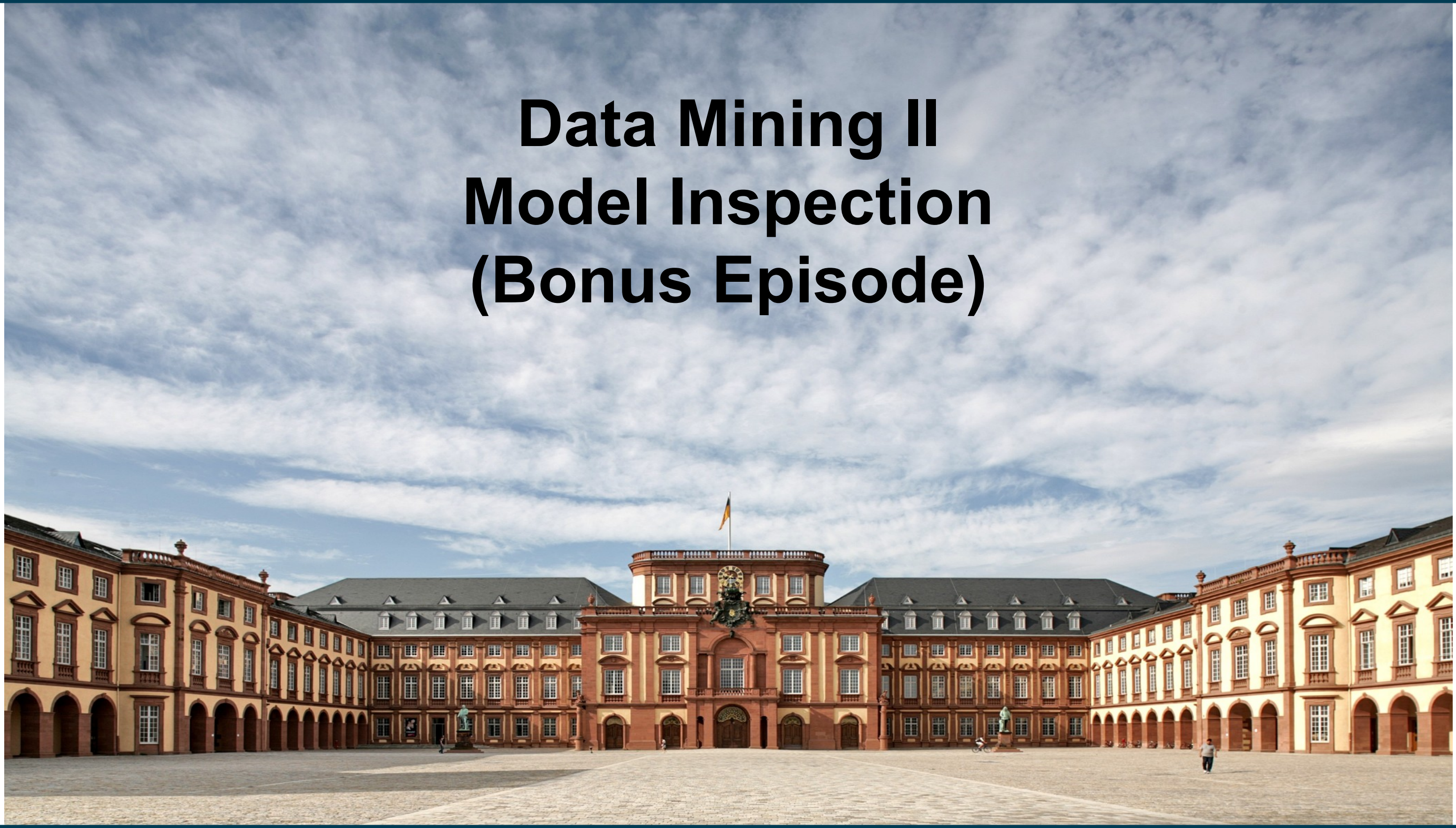
Data Mining II
Model Inspection
(Bonus Episode)



Heiko Paulheim

Occam's Razor Revisited

- Let’s rephrase:
  - if you have two models
  - where none is *significantly* better than the other
  - choose the simpler one

- Indicators for simplicity:
  - number of features used
  - number of variables used, e.g.,
    - hidden neurons in an ANN
    - no. of trees in a Random Forest
    - …

Measuring Model Simplicity

- Idea: the more the models focuses on less features, the simpler
  - Not necessarily: the better

Caveats: identifiers, false predictors, ...

- Good models have *both*...
  - ...low test error
  - ...low complexity

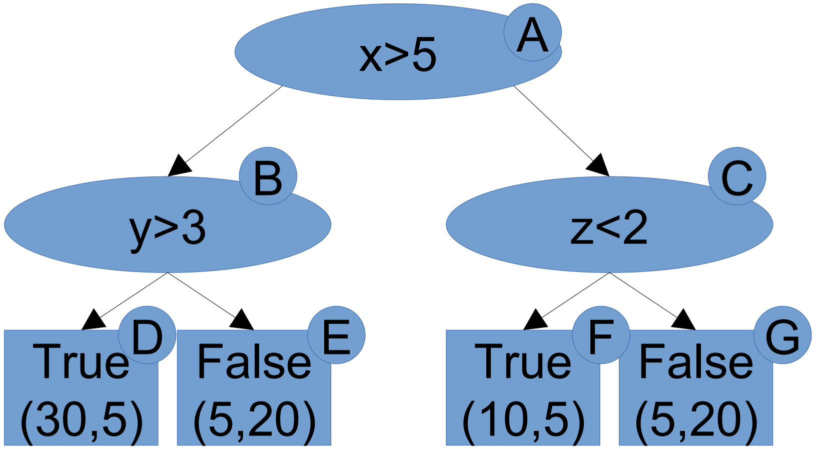
Measuring Feature Importance

- Example: random forests
- A feature is more important if…
  - ...it is used in many trees
    Rationale:
    - weighted prediction across trees
    - the more trees it is used in, the higher the influence
  - ...it is used to classify many examples
    Rationale:
    - more predictions are influenced by that attribute
    - i.e., for a single example: higher likelihood of influence
  - ...it leads to a high increase of purity on average
    Rationale:
    - if the purity is *not* increased, the split is rather a toin coss

Measuring Feature Importance

- A feature is more important if…
  - ...it is used in many trees
  - First take:

$$\text{Importance}(F) = \frac{\text{no. of trees containing F}}{\text{no. of trees}}$$



Measuring Feature Importance

- A feature is more important if…
  - ...it is used to classify many examples
  - First take:

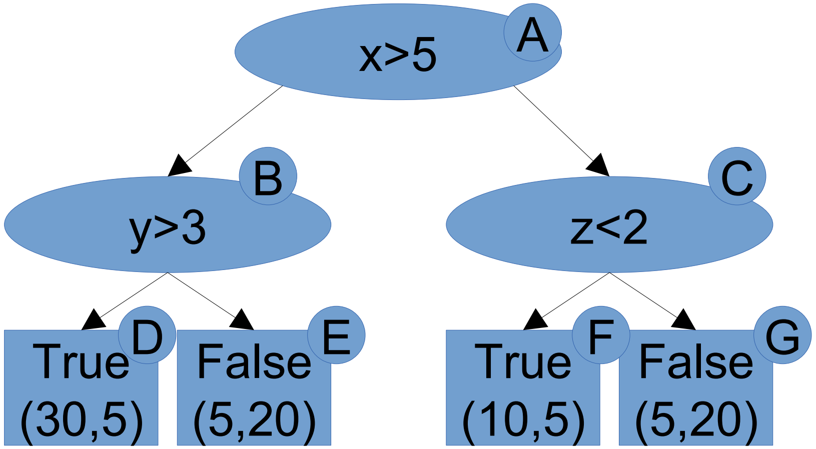
$$\text{Importance}(F) = \frac{\text{no. of examples classified using F}}{\text{no. examples}}$$

  - In this example tree:

    Importance(x) = 1.0

    Importance(y) = 0.6

    Importance(z) = 0.4



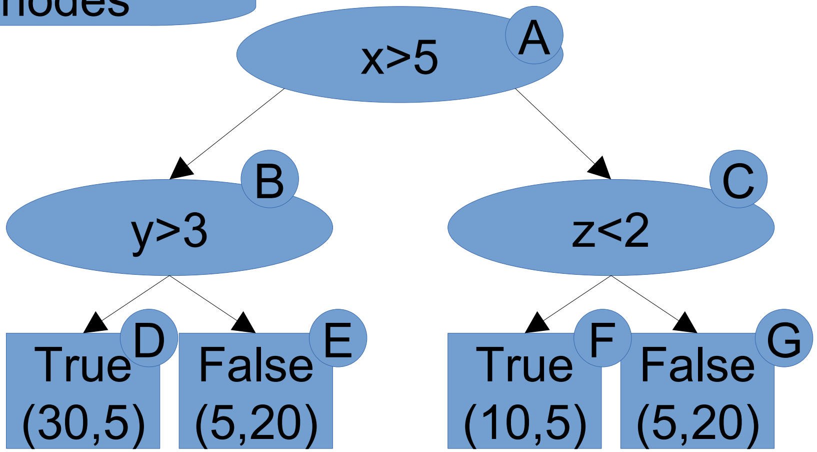
Measuring Feature Importance

- A feature is more important if…
  - ...it leads to a high increase of purity on average
  - First take:

Change of impurity of node and its split nodes

$$\text{Importance}(F) = \Delta I(t, t_s)$$

  - In this example tree:
    - Importance(x) = 0.104
    - Importance(y) = 0.246
    - Importance(z) = 0.109



- gini(A) = 0
- gini(B) = 0.083
- gini(C) = 0.125
- gini(D) = 0.357
- gini(E) = 0.3
- gini(F) = 0.167
- gini(G) = 0.3

Measuring Feature Importance

- For example, random forests
- Putting the pieces together:

Grows with no. of trees using F

$$\text{Importance}(F)=\frac{1}{\text{no. of trees}}\sum_{m=1}^{\text{no. of trees containing F}}\ \sum_{\text{nodes n in tree m containing F}} p(n)\,\Delta I(s_n,n)$$

Probability of single example passing this inner node

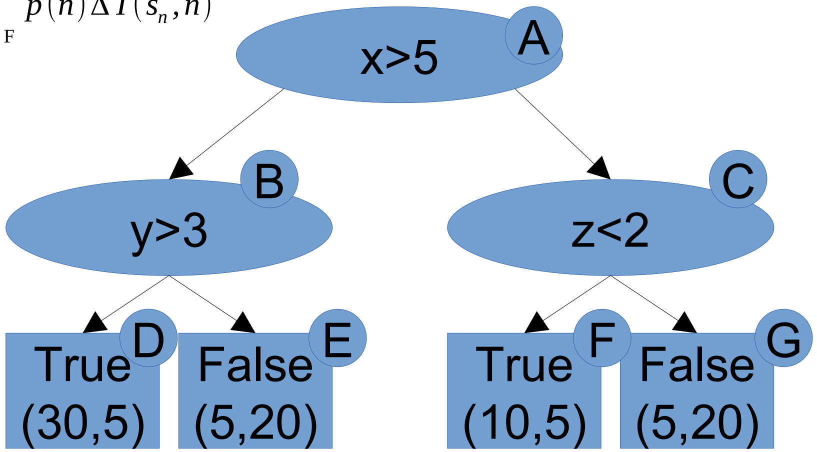
Growth in impurity (e.g. Gini, Entropy)

Measuring Feature Importance

- For example, random forests
- Putting the pieces together:

$$\text{Importance}(F) = \frac{1}{\text{no. of trees}} \sum_{m=1}^{\text{no. of trees containing F}} \sum_{\text{nodes n in tree m containing F}} p(n) \Delta I(s_n, n)$$

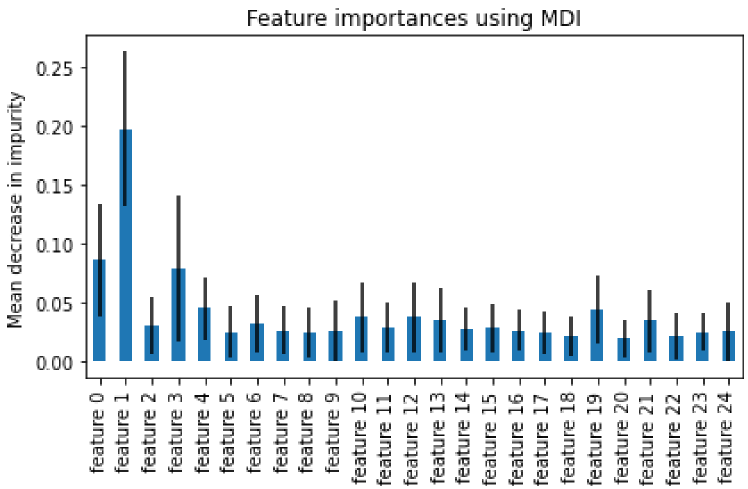
- In this example:
  - Importance(x) = 1.0 * 0.104 = 0.104
  - Importance(y) = 0.6 * 0.246 = **0.148**
  - Importance(z) = 0.4 * 0.109 = 0.044

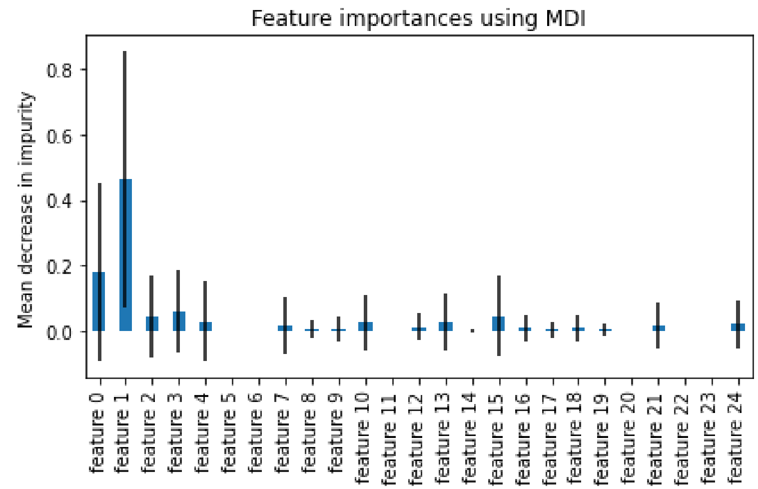


Back to Model Simplicity

- Left hand side:
  - Accuracy on test set: 0.72
- Right hand side:
  - Accuracy on test set: 0.66

Fewer influential features





Feature Weights and Model Simplicity

- Idea of feature shuffling:
  - If a feature is relevant, assigning random values to it should make the predictions worse
  - Simulation of random, but realistic values: shuffling a column
- This can be applied to *any* model

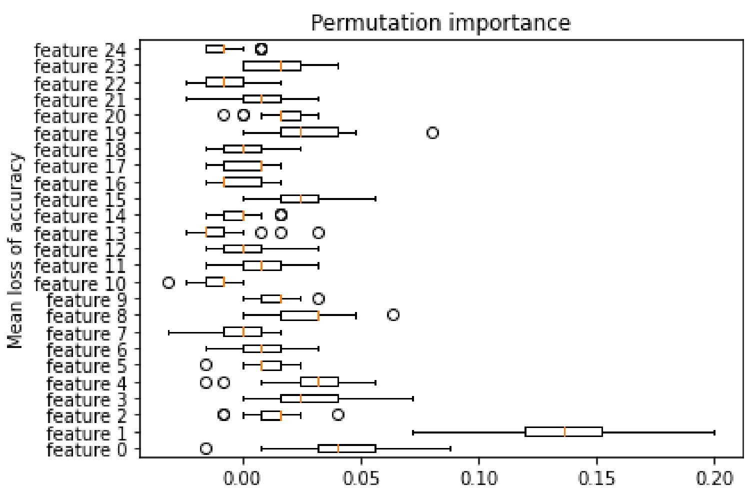
X_A	X_B	X_C	Y
xa1	*xb1*	*xc1*	*y1*
xa2	*xb2*	*xc2*	*y2*
xa3	*xb3*	*xc3*	*y3*
xa4	*xb4*	*xc4*	*y4*
xa5	*xb5*	*xc5*	*y5*
xa6	*xb6*	*xc6*	*y6*

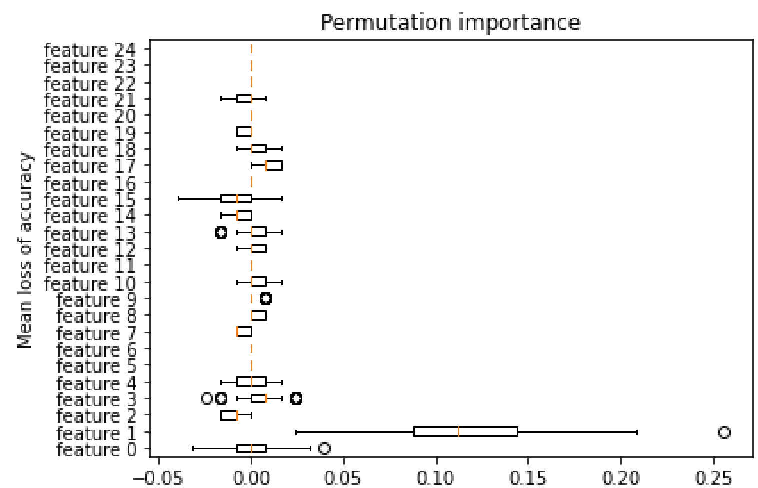
https://towardsdatascience.com/feature-importance-with-neural-network-346eb6205743

Back to Model Simplicity

- Left hand side:
  - Accuracy on test set: 0.66
- Right hand side:
  - Accuracy on test set: 0.64

Fewer features with importance >0





Feature Weights and Model Simplicity

- Let's rephrase:
  - if you have two models
  - where none is *significantly* better than the other
  - choose the simpler one

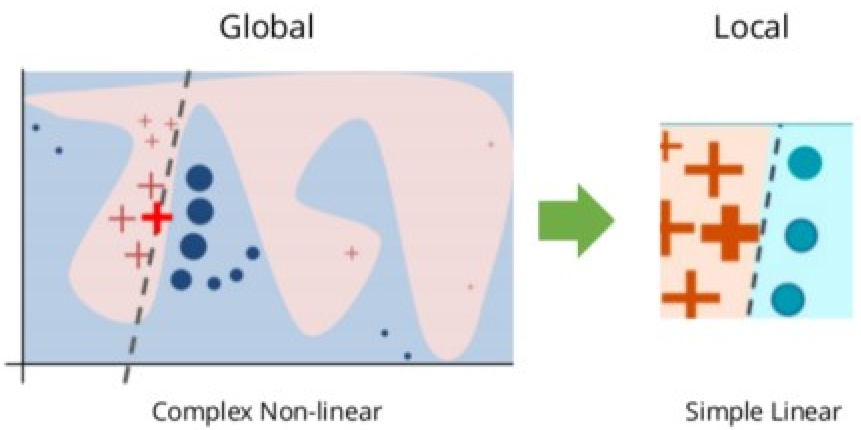
- Feature weights
  - Can indicate model simplicity (few high weighted features)

- Examples for computation
  - Random Forest, XGBoost: Mean Decrease in Impurity (MDI)
  - General: feature shuffling

LIME Model Explanation

- Idea: in a local area, models are simpler
  - They do not need to account for all the patterns of the data
  - Concentrate on patterns relevant in that area
- Motivation:
  - Try to extract the relevant model for a given data point
  - Hopefully, this is simple enough to interpret

$y>5x \longrightarrow$ class = +

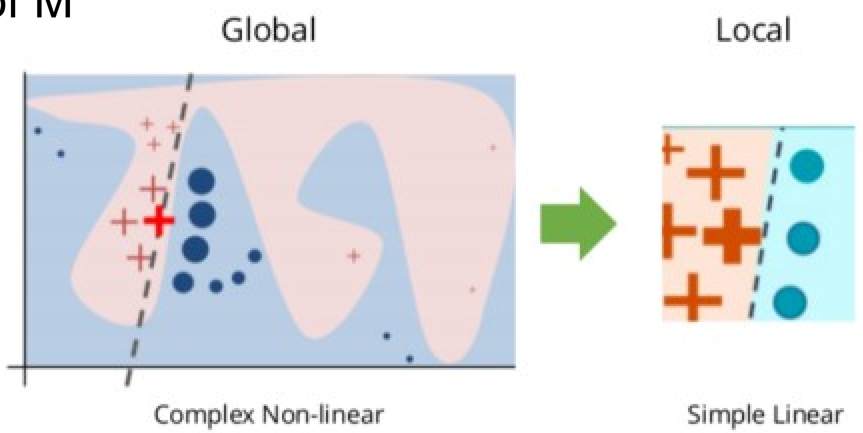


https://c3.ai/glossary/data-science/lime-local-interpretable-model-agnostic-explanations/

LIME Model Explanation

- How to interpret a “black box” (i.e., uninterpretable) model M?
- Local: for a datapoint p
- Basic idea:
  1) create artificial datapoints P(p) in vicinity of p
  2) score each p’ in P with black box model
  3) learn interpretable model M’

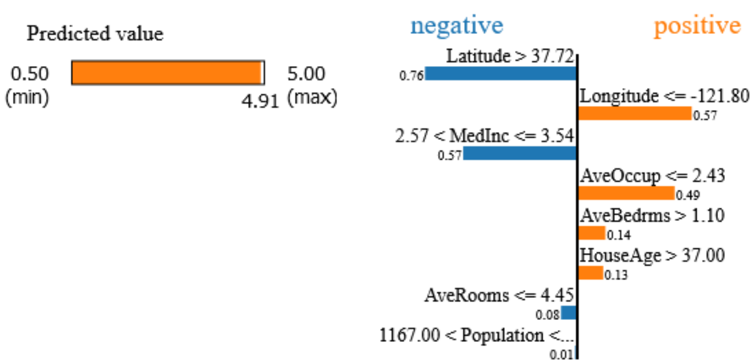
     → values: P, labels: scores of M
  4) create prediction for p using M’
     or analyze M’ directly

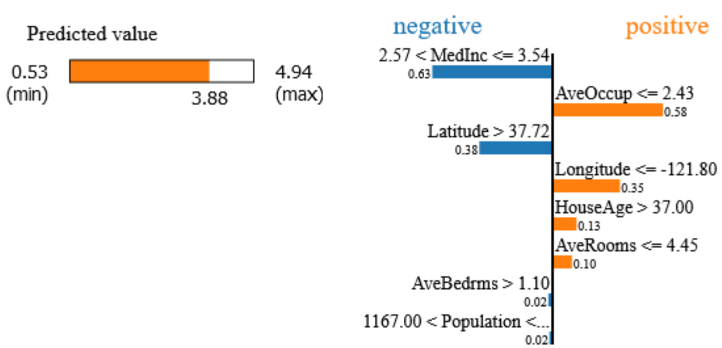


https://c3.ai/glossary/data-science/lime-local-interpretable-model-agnostic-explanations/

LIME Model Explanation (example)

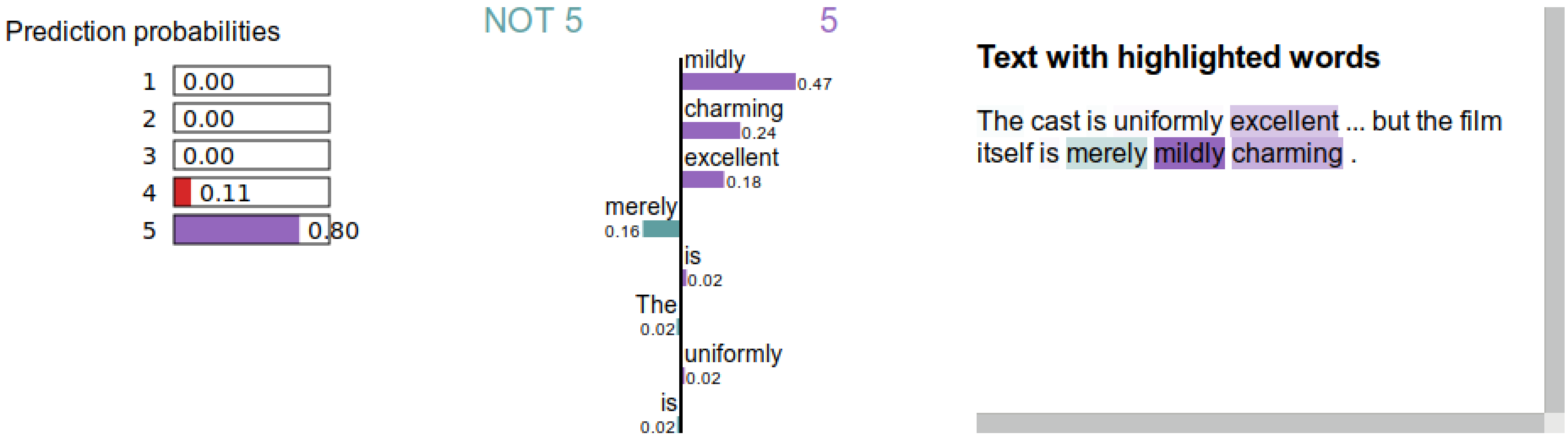
- Left hand side:
  - Model score on test set: 0.80
- Right hand side:
  - Model score on test set: 0.74





LIME Models for Non-Tabular Data

- Example: text classification
  - Datapoints P(p) are created by changing single *words* in training example



https://towardsdatascience.com/fine-grained-sentiment-analysis-in-python-part-2-2a92fdc0160d

LIME Models for Non-Tabular Data

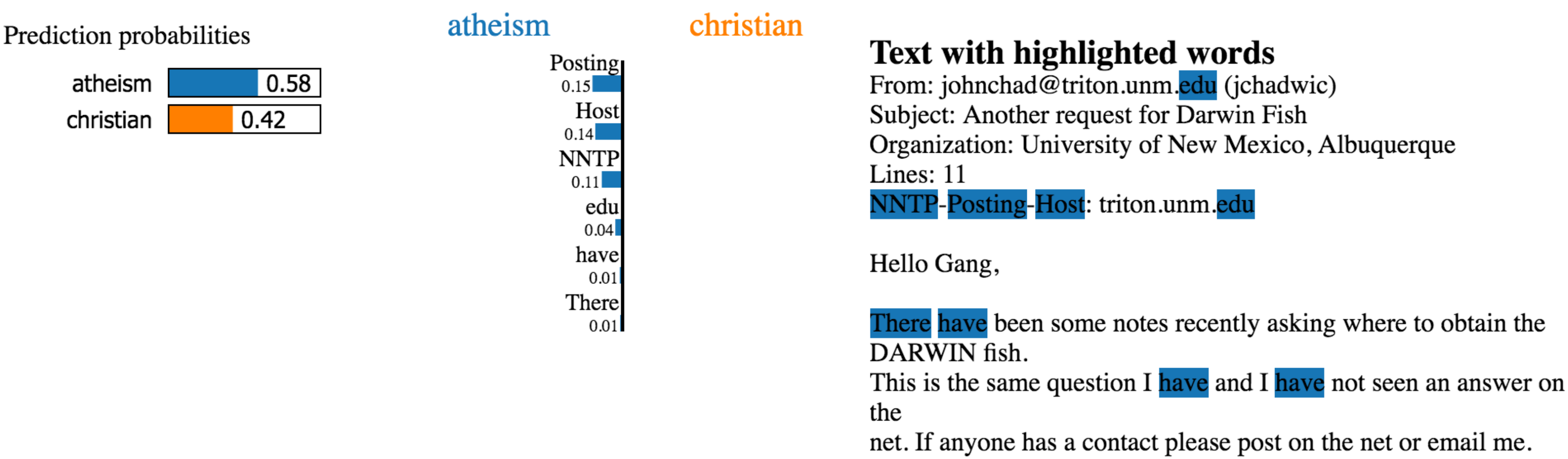
- Example: image classification
  - Datapoints P(p) are created by changing single *pixels* in training example

```
336 fox squirrel, eastern fox squirrel, Sciurus niger 0.9377041
844 swing 0.001819109
337 marmot 0.00076952425
```

https://www.inovex.de/de/blog/lime-machine-learning-interpretability/

Model Inspection for Improving Model Quality

- Example: Text Classification
  - Observation: focus on metadata and stop words



https://homes.cs.washington.edu/~marcotcr/blog/lime/

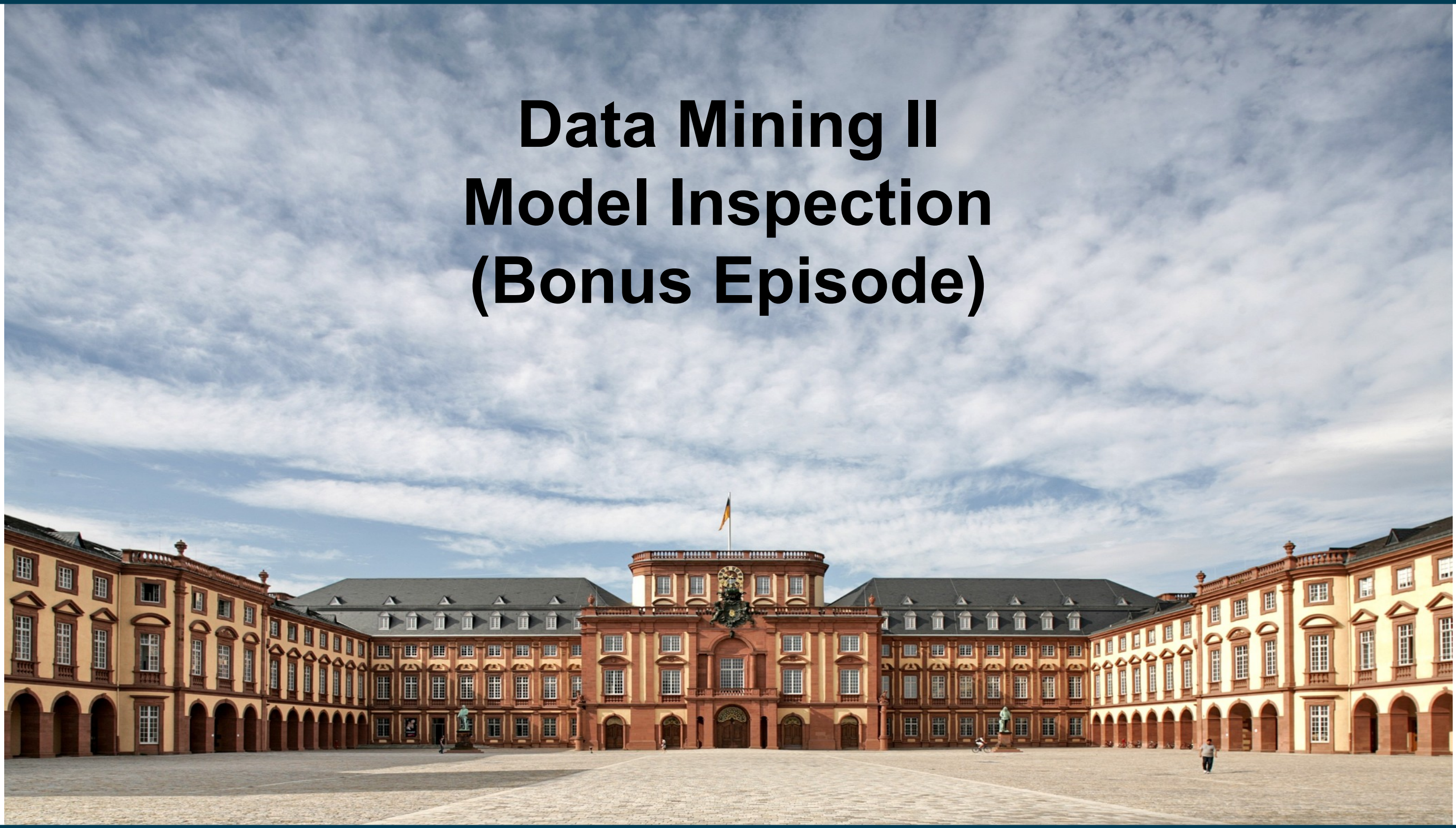
Take Aways

- Model inspection on global level
  - Model complexity
  - Proxy: feature importance
  - Less complex model → more likely to generalize

- Model inspection on local level
  - Generating explanations for test instances
  - Do they look plausible?

UNIVERSITÄT MANNHEIM

Data Mining II
Model Inspection
(Bonus Episode)



Heiko Paulheim