

# SENTIMENT ANALYSIS

---

90 MIN WORKSHOP





## Robin Rojowiec

*Master Student and Consultant for AI/NLP*



<https://www.linkedin.com/in/rojowiec/>



[robin.rojowiec@de.ibm.com](mailto:robin.rojowiec@de.ibm.com)



[https://twitter.com/Robin\\_it\\_is](https://twitter.com/Robin_it_is)



<https://www.instagram.com/robinlphood/>

# AGENDA

---

- Introduction 10 min
- Exercise 70 min
  - Task
  - Implementation
  - Testing and Improvements
- Conclusion 10 min



Quelle: <https://www.kdnuggets.com/2018/03/5-things-sentiment-analysis-classification.html>

# SENTIMENT BASED ON WORDS

---

Document Level = Neutral

I watched a bad movie last night. Luckily, the next movie was much better.

Sentence 1 = Negative

Sentence 2 = Negative



# SENTIMENT BASED ON WORDS (ASPECTS)

---

I watched a bad movie last night. Luckily, the next movie was much better.



Movie: Negative



Movie: Positive



# EXERCISE

---

Task





# WORD BASED BINARY SENTIMENT CLASSIFICATION

---

- Predict sentiment of unseen movie reviews
  - 25 000 reviews for training, equal split positive/negative  
*(You can reduce the amount for training/test to speed up development)*
- Calculate class probability, feature probability and the probability of each word belonging to each class
- Use logarithmus naturalis ( $\ln$ ) to avoid underflow
- discard unknown or zero-frequent tokens!

# WORD BASED BINARY SENTIMENT CLASSIFICATION

---

Class Probability

$$P(\textit{Class} \mid \textit{Document}) = \log P(\textit{Class}) + \sum_{i=0}^n \log p_{\textit{Word}_i}$$

Word probability

$$P(\textit{Class} \mid \textit{Word}) = P(\textit{Word}) + P(\textit{Word} \mid \textit{Class})$$



# WORD BASED BINARY SENTIMENT CLASSIFICATION STEPS

---

- 1) Load text files
  - 2) Tokenize and remove stopwords
  - 3) Calculate for each token:
    - 1) Number of occurrences in the documents
    - 2) Number of occurrences per class (positive/negative)
  - 4) Write prediction function which normalizes the probabilities
  - 5) Run test script and modify your code to improve results
- (You should end up with around 70% accuracy)

# WORD BASED BINARY SENTIMENT CLASSIFICATION

## BONUS

---

Bonus 1)

Calculate integrate the TF/IDF value (document frequency ratio to word frequency) to improve accuracy

Bonus 2)

Change granularity to sentence level or aspect-level (rule-based)

Bonus 3)

Try to improve performance with word bi- and trigrams as well as character bi- and trigrams





# EXERCISE

---

Implementation





# CONCLUSION

---



# CONCLUSION

---

- ~ 70 % Accuracy => 7/10 Classifications are correct
- Simple Algorithm which uses probabilistic Properties
- Nice Exercise, in practice you would use one of these:



And many more!



# FURTHER QUESTIONS ?

---



# FEEDBACK

---

1. „I liked...”

2. „For future workshops it would be nice...”

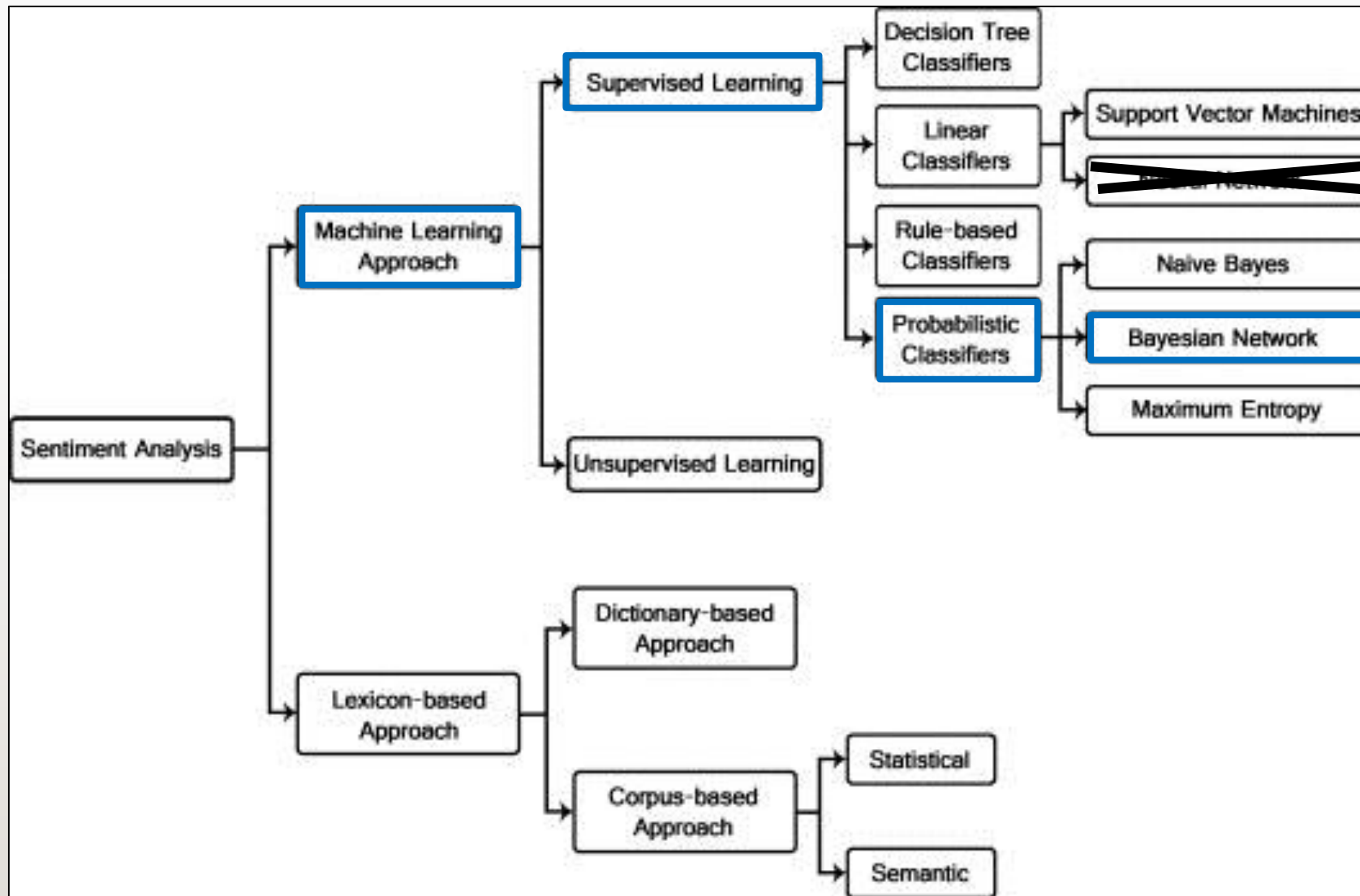
3. „Overall the ... was very good”



# BACKUP

---







# PROBABILISTIC SENTIMENT CLASSIFICATION

---

- Word-based (Unigram, 1 Word = 1 Feature)
- Probability  $\rightarrow$  Frequency for  $x_i$  / Frequency  $x$
- Granularity:
  - Word-Level (Aspects)
  - Sentence-Level
  - Document-Level
- Unseen words:
  - Backoff / Laplace Smoothing
  - Discard words