



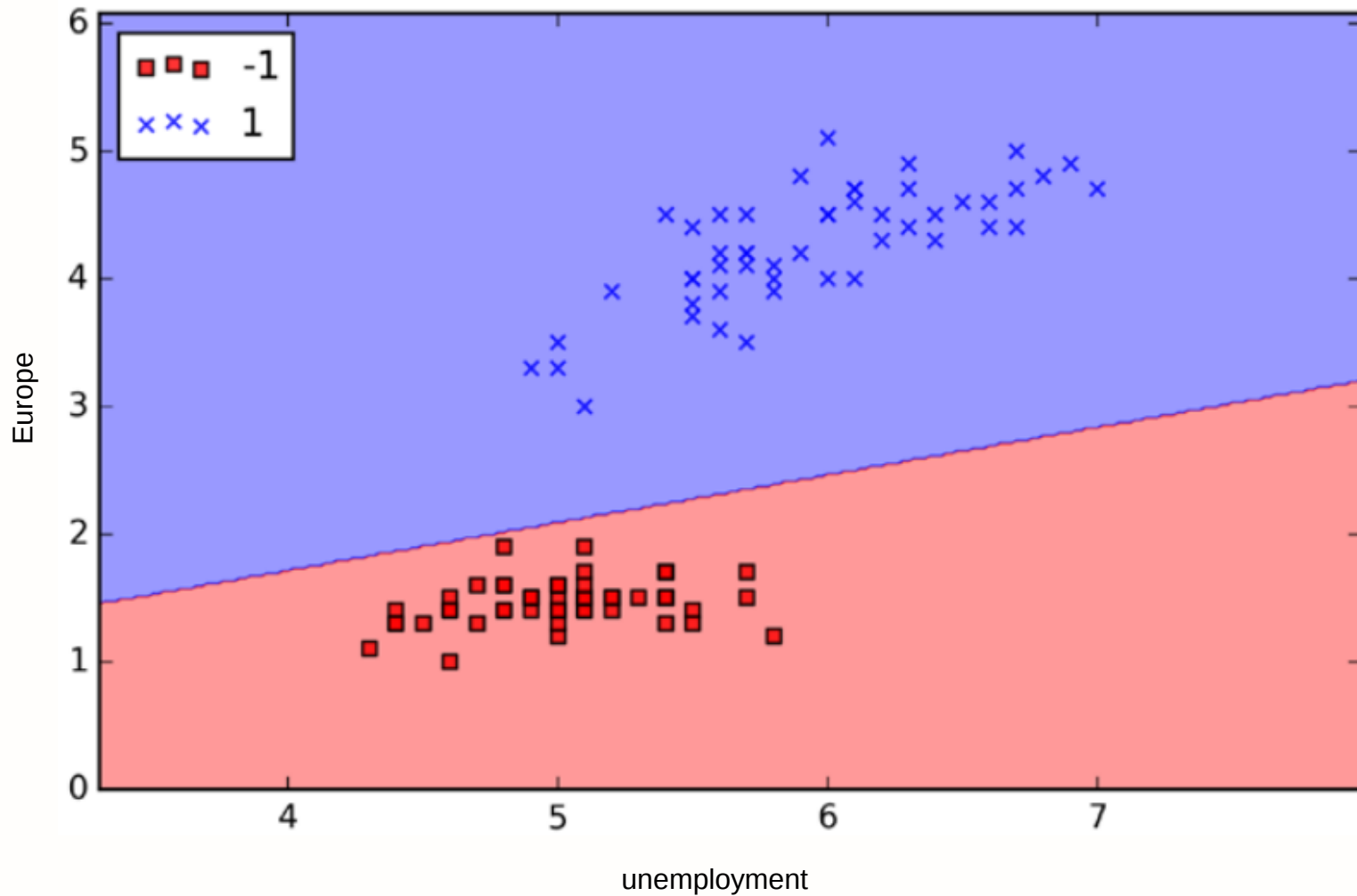
Comparison of Linear SVM Algorithm Implementations in Python for Solving an Author Identification Problem

Miloš Švaňa

Author Identification Problem

- **Goal:** identify a politician making a speech
 - supervised learning problem (multi-class text classification)
 - similar traits as the sentiment analysis problem (opinion mining; message polarity classification)
- Machine-learning algorithms are required to solve this problem
 - resource-intensive task
 - **a most efficient approach should be selected**

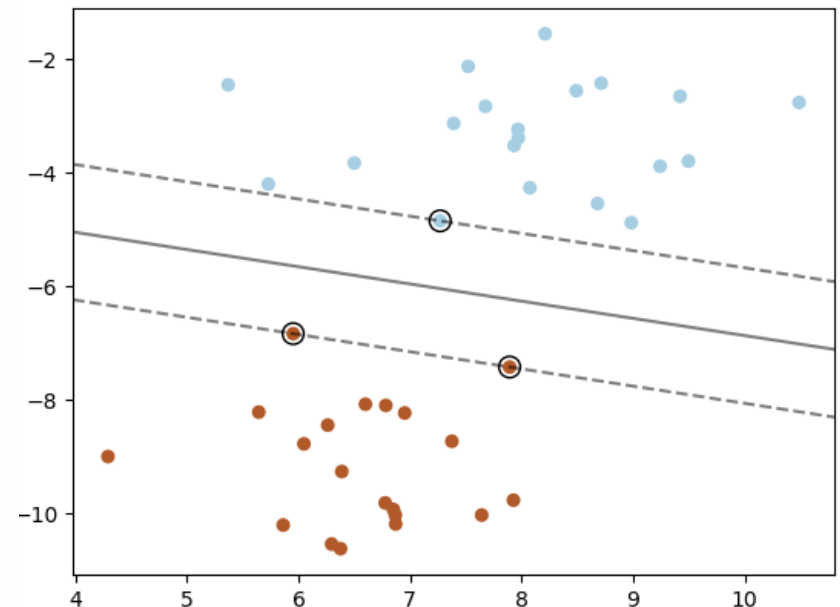
Linear classifiers



Multi-class classification using SVM

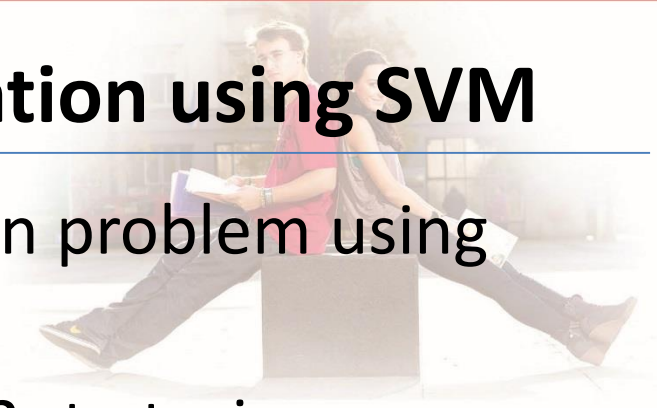
- **Support Vector Machine (SVM) algorithm** is recommended for a text classification task
 - find the “best” hyperplane (model) separating data points, i.e. with equal distance to closest data points (max. margin)
 - originally for binary classification problems

Best separating hyperplane for a binary classification problem



Multi-class classification using SVM

- Solving of a multi-class classification problem using SVM:
 - training a classifier follows one of 2 strategies:
 - One vs. One
 - One vs. Rest
- **Training a SVM classifier:** done through solving of a loss (cost) function minimization problem



Author Identification Problem - methodology

1. Dataset description
2. Feature extraction
3. Class prediction using Python language



Author Identification Problem - methodology

1. Dataset description

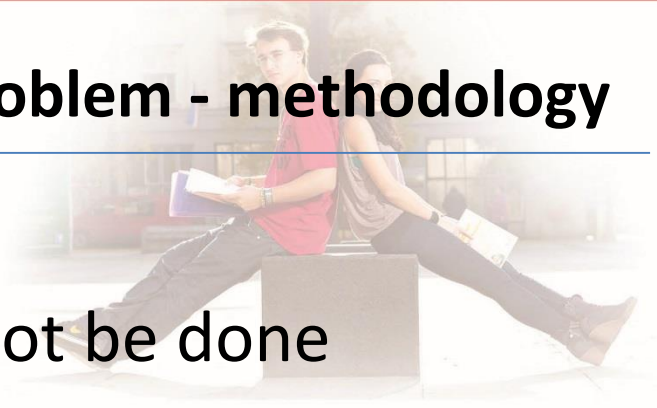
- Corpus (set) of documents (i.e. transcripts of politicians' speeches)
- Only politicians with more than 50 speeches
- 158 different politicians
- 39347 speeches



Author Identification Problem - methodology

2. Feature extraction

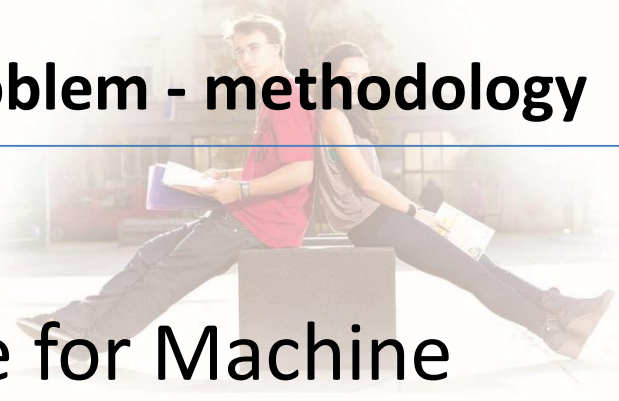
- Speech author determination cannot be done directly from a whole document
 - **features must be extracted from the document**
 - **TF-IDF approach is used** (with dimensionality reduction)
- **TF-IDF** (term frequency–inverse document frequency) **approach**
 - document transformed into feature vectors (bag of words)
 - *each dimension* = one term (word) in the dataset
 - *each value* = term importance in the document



Author Identification Problem - methodology

3. Class prediction using Python

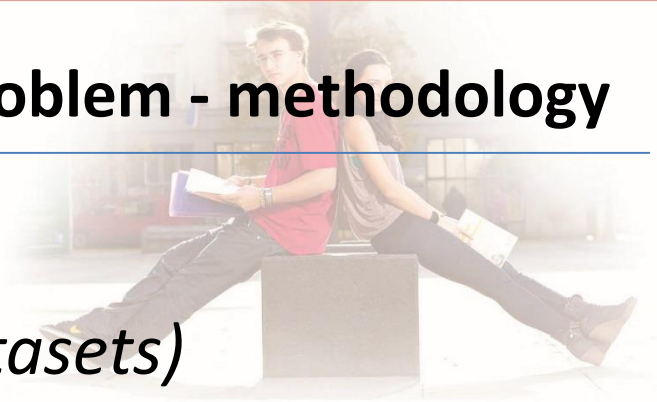
- Popular programming language for Machine Learning
 - Many tools for science and engineering in general
 - Easy to learn
 - Supported by big companies
- Multiple implementations of the same algorithm



Author Identification Problem - methodology

Two popular libraries selected

- **Scikit-learn** (*best for smaller datasets*)
- **Theano** (*best for large datasets; also used for solving of deep-learning problems*)
- **3 options (i.e. classification models):**
 - two ready-to-use SVM-implementing Scikit-learn library classes (*SVC* and *LinearSVC*)
 - one custom SVM implementation using Theano (*hinge loss optimization*)

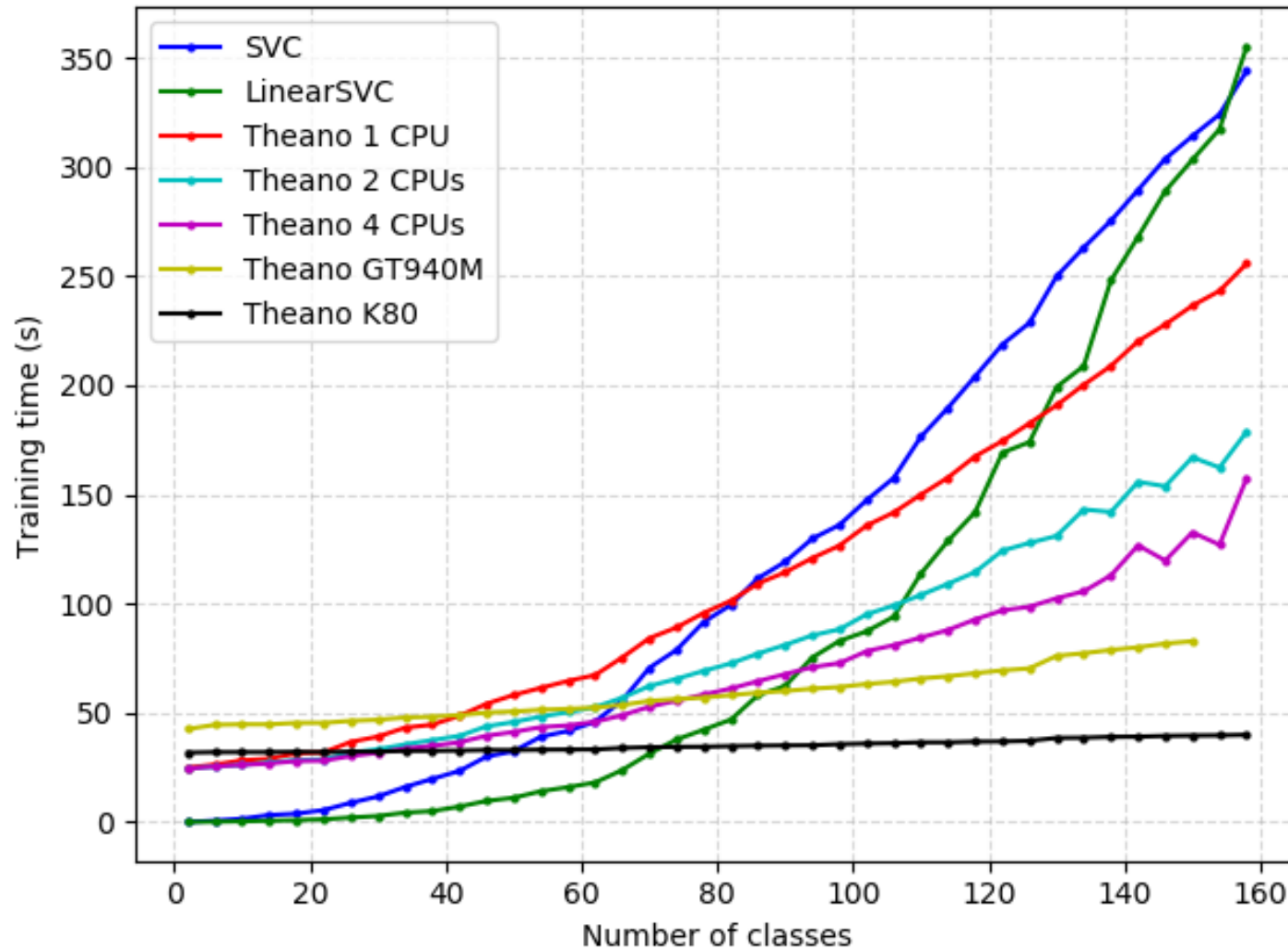


Comparison of SVM implementations

- **Performance comparison of 2 computation environment setups:**
 - *CPU-only computation*
 - Intel Xeon 2.6GHz (Sandy Bridge) on Google Cloud
 - 1 to 4 cores (2-8 vCPUs)
 - *GPU computation*
 - GPUs are very useful for ML tasks (designed for parallel computing operations)
 - K80 on Google Cloud and GT940M on consumer laptop



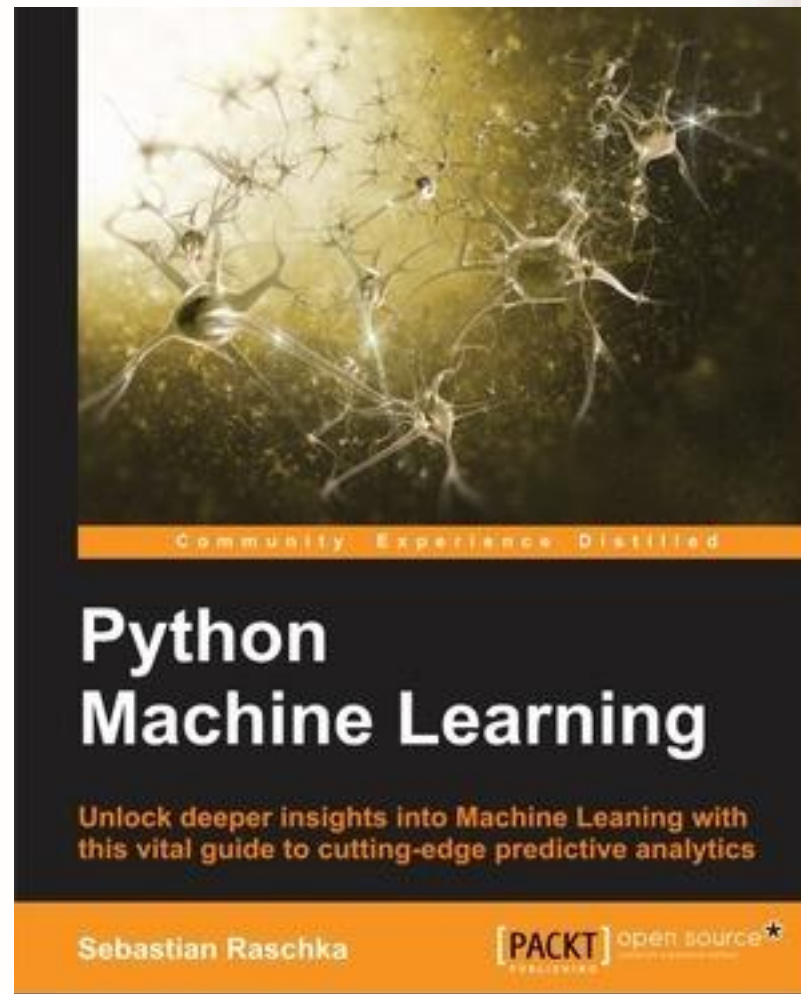
Performance results



Conclusion

- **Comparably best performance:** custom SVM impl. using the Theano library
 - Faster even on 1-CPU setup
 - GPU usage proved to be beneficial
 - 4-CPU setup was close to both GPU-related setups





THANK YOU FOR YOUR ATTENTION



Q&A



milos.svana.st@vsb.cz



MilosSvana