1. Data Set:
   1. Articles and Posts in German
   2. Posts are (partially) labeled
   3. Articles and Posts give rise to a tree structure, as each post replies to either an article or a post
   4. Posts and articles aren’t completely saved, but only the first few words. They end with three dots in case there was more text which was cut off.
   5. As our documents are user comments (mostly), they may contain things such as:
      1. Typos
      2. Auto-correct fails (or, maybe just somewhat incoherent speech in general)
      3. Creative ways of putting emphasis, e.g. Capslock or repeated vocals (looooooooooooooong)
      4. At least in the samples I’ve skimmed through, there were no smileys (in general, no exotic non-ASCII symbols)
   6. 9 Labels:
      1. Arguments used
      2. Personal stories
      3. Discriminating
      4. Inappropriate
      5. Offtopic
      6. Sentiment (negative/neutral/positive)
      8. Some of the labels are very unbalanced
         1. Example:  
            ToDo rausfiltern
      9. SentimentPositive has so few positive samples, it’s basically untrainable.
   7. Some posts have all 9 labels set. Most have only a subset of the labels set, possibly none(??stimmt das?).
   8. Neither of these labels is disjunctive to any other, i.e. a sample can have any combination of labels  
      (even SentimentNegative & SentimentNeutral(??? Stimmt das???), showing that there were multiple people with different evaluation-approaches labeling the dataset (well, common thing, also the fact that the ones labeling aren’t 100% accurate, especially for such vague concepts, isn’t anything new either)
   9. There are some additional labels, which are given by the forum, most notably:
      1. Positive & negative votes
      2. Whether a post was deleted
2. Short statistic analysis of the features:
   1. Covariance matrix (between those samples which had all 9 labels set)
      1. The covariance between most features is negligible;   
         Only exception: SentimentNegative and SentimentNeutral, which has positive correlation (??? Negative, richtig?? Nur positive Kovarianz, was ja klar ist)  
         Je nach Korrelation: Negativ klar, da mehr oder weniger disjunkt (aber nicht annähernd so ausgeprägt wie sein sollte, nämlich 1 für disjunkt!)  
         Positiv: Dafuq.  
         In jedem Fall ist diese Korrelation die Erklärung, warum wir alle Labels als Binärlabels ansehen, und keinen Regressor für Sentiment bauen.
3. Used features:
   1. Word embeddings, for each sample (i.e. labeled post) one: (… we tried both i. and ii.)
      1. Word-Embedding as given by Spacy (dense embedding) (we tried both md and lg [latter has many more word-embeddings], but lg was in fact not consistently better in results)
      2. Word-Embedding yielded by calculating the tf-idf, after lemmatizing and removing punctuation (sparse embedding; lemmatization ‘necessary’ because of way too few samples for the many forms of German words)
4. Trained classifiers:
   1. SVC with different kernels, most notably the RBF-Kernel and polynomial kernels of different degrees. (was is RBF???)
   2. Boosted Decision Trees
   3. MLP (Feed-Forward NN)
   4. k-Nearest Neighbor
   5. Naive Gaussian Bayes
5. Results:
   1. [Show graphs of best models; With meta-labels and without]
      1. Especially with sparse high dimensional embeddings, primal/dual SVC did give best results
      2. With dense embeddings, SVC with RBF-Kernel and other classifiers gave best results (some that, some that)
   2. Difficult labels and somewhat simple labels:
      1. Difficult: Discriminating, Inappropriate, Offtopic
         1. The classifiers for these all might benefit from knowing the full parent comment thread as well as the parent post.
         2. Especially offtopic intuitively should need the original post to be labeled – however it wasn’t the one the classifiers struggled with the most, which was (?) Discriminating(?)
         3. Unsure how to use the tree structure to help labels ‘Discriminating’ and ‘Inappropriate’.
         4. For label ‘Offtopic’, we can give the classifier both the embedding for the post as well as the embedding of its parent article.
            1. ToDo: Results