**Stats 101C IMDB Sentiment Analysis Project**

Refer to: [finalProjInfo.pdf](https://drive.google.com/file/d/1FoxCHt501C9DaLh-OQ4Kw1ViMGlmr5pP/view?usp=drive_link)

| **Questions to answer**   * ~~R or Python (my vote is python)~~   + ~~Depending on what we have to submit, we can do diff stages in diff languages (e.g. preprocessing in Python, experiment in R)~~ * Github repo? or too much work lol   + If we split up as I discuss below, not really necessary as long as we are timely & organized * ~~Dividing tasks~~   + ~~Preprocessing, Experiment, Writing Intro+conclusion, writing paper/report~~   + ~~See chart below~~   + ~~Six people → two for each stage~~ * Timeline   + Due Friday 12/8 → have it ready by Thursday midnight so we can check in discussion Friday before submitting??   + I think final final deadline is sunday 12/10 for prof corrections * What files do we have to submit??   + Obv the report paper   + Safe to assume also the code file(s)   **Misc notes**   * make sure to cite sources   + e.g. opinion lexicon from kaggle * Include graphs/visualization   + e.g. show some comparison between accuracies of each model |
| --- |

| Task | Method | Assignment or notes idk |
| --- | --- | --- |
| **Preprocessing (Leigh Anne, Mabel in Python)**  **Deadline: Monday night (or maybe Sunday)**   * Export data to csv and add to folder | | |
| Eliminate unnecessary words aka Tokenization | Python nltk ~~or spaCy~~ |  |
| Text vectorization: convert tokenized words into numerical vectors | sklearn ~~CountVectorizer~~ or **TfidfVectorizer (TF-IDF as mentioned above)** | <https://medium.com/analytics-vidhya/sentiment-analysis-on-amazon-reviews-using-tf-idf-approach-c5ab4c36e7a1> |
| Dimensionality reduction | **PCA** or ~~SVD~~ |  |
| Word embeddings | ~~Word2Vec~~ |  |
| Exploratory data analysis at some point in the above tasks | Word cloud/frequency?? Maybe length vs sentiment | * Word cloud!!!! |
| **Experiment (Jamie, Steven in R)**  **Deadline: Tuesday due date**   * Split into training and testing for training the model * Decide on split:   + Decided on **80/20** split for all training and testing sets.   + K tbd. * And then do CV with testing * Add lots of notes/comments so easy for write-up. * Take note of references (ask Prof if need to reference for HW packages???) | | |
| Logistic Regression- Steven |  | See HW |
| K-Nearest Neighbors- Steven |  | See HW |
| LDA, QDA - Steven |  | See HW x2 |
| Random Forest - Jamie |  | Use randomForest package. |
| Cross-Validation (to decide which model is best) | K-fold CV (K = 5 or 10), entropy?? | See HW do for every model, **use as gauge of accuracy.** |
| **Report write up (Tyler, Kiran)**  **Deadline: Friday due date** | | |
| Clean up code file(s) |  | Clean up your own part so that write up can be done clearly |
| Write the paper | LaTeX |  |

1. **Transforming Text into Numerical Data**

* Challenges in dealing with text data
* Numericalize textual data
* Data size can be big for both dataset and dimension of features
* Eliminate unnecessary things in textual documents
  + Personal pronouns
    - e.g. He, She, I, We…
  + Determiners – Determiners tend to mark nouns where a determiners usually will be followed by a noun
    - e.g. the, a, an, another
  + Coordinating conjunction
    - e.g. for, an, nor, but, or, yet, so
  + Prepositions
    - in, under, towards, before
* Transform adverbs into adjectives
  + e.g. Happily to Happy
* Numericalization
  + Create sentiment dictionary
  + Bag of words
* Opinion Lexicon
  + Use this: <https://www.kaggle.com/datasets/nltkdata/opinion-lexicon>
* Term Frequency and Inverse document frequency (TF-IDF)
  + Large TF = important feature
  + Large IDF = important feature
* Word Embeddings
  + Represent word as a numerical vector
  + Word2Vec (tensorflow??)

1. **High-Dimensional Feature**

* PCA

**FROM EXAMPLE PAPER** [exampleReport.pdf](https://drive.google.com/file/d/1LDOg026wiRM2VcNLj-je78siCfwJzMn4/view?usp=drive_link)**:**

1. Abstract, Introduction
2. Preprocessing:
   * 1. Shrunk dataset to only reviews from CA
        1. Not rly relevant for us bc we don’t have as much info
     2. Created own binary column
        1. Not relevant for us
   1. Descriptive statistics
      1. Bar graph of distribution of star ratings
         1. We only have two ratings lol… should still do something somewhat similar though? Exploratory or descriptive
         2. **Want to show if our dataset is balanced or imbalanced**
      2. Frequency of Words in review
         1. We should do this: **word frequency plot**
   2. Clustering of users
      1. K-means model on number of reviews, average stars, average review length, etc for each users (user behavior)
         1. We don’t have user data so not relevant
         2. However I think **we can cluster based on length of review**
   3. Feature engineering: sentiment lexicon and PCA
      1. removed stop words
      2. used TF-IDF to show words between a minimum (0.1) and maximum (0.7) document frequency (how often a document contains a certain word)
      3. Constructed sentiment lexicon by using a pre-made text doc with negative and positive words to vectorize all pos and neg words in each review and converted the array into a dataframe
      4. Then use PCA on word columns to reduce dimensionality of dataset
   4. Balancing dataset: SMOTE
      1. combat issue of imbalanced data for training
3. Experiment:
   1. Logistic regression
   2. K-Nearest Neighbors
   3. LDA & QDA
   4. Random Forests
4. Conclusion blah blah blah
   1. Results and Analysis
   2. Final remarks