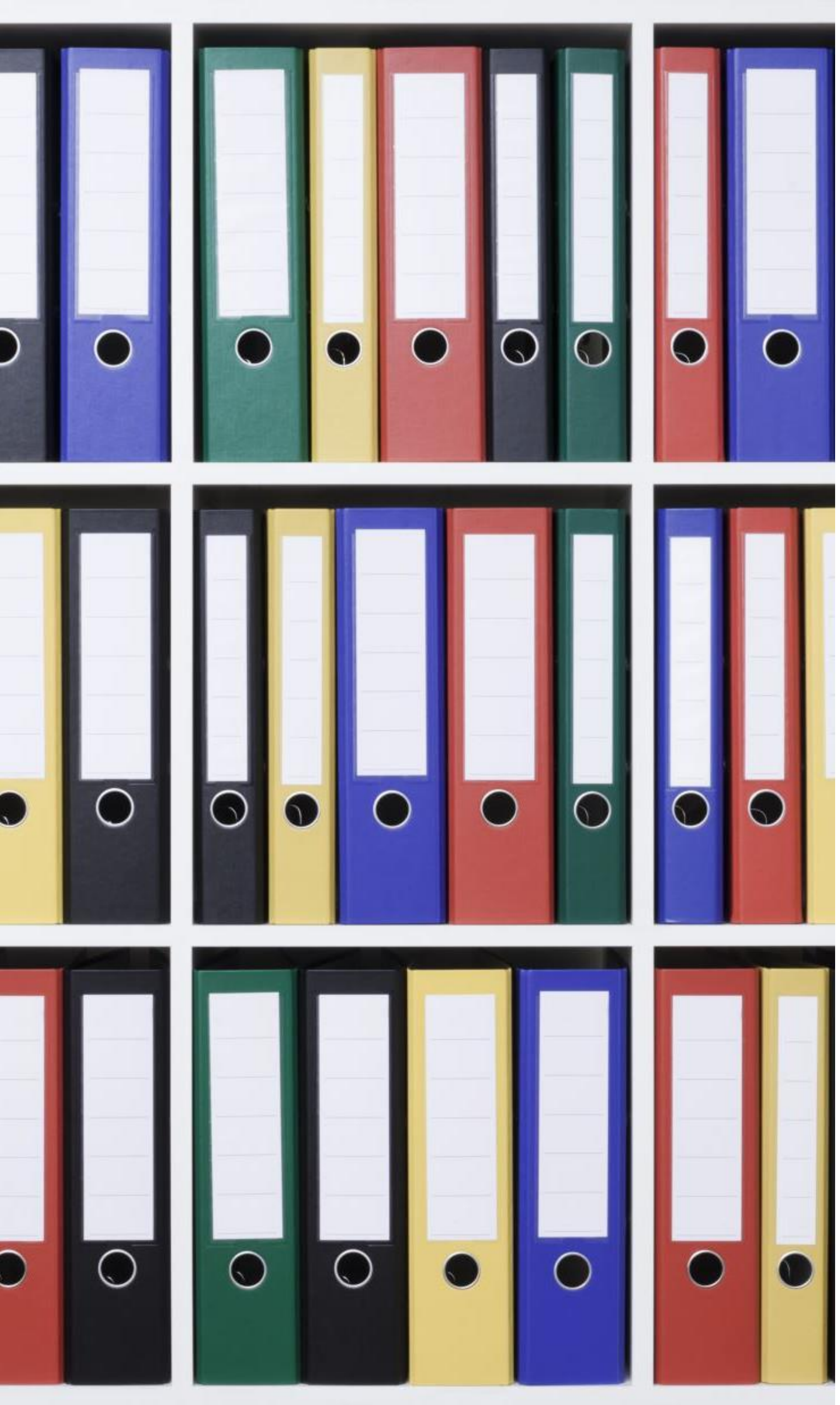
ASSIGNMENT 1: TEXT CLASSIFICATION

GNG5125 Data Science Applications

Spring-Summer 2022

Group 12

Yinruo Jiang (300274815)
Rasheeq Mohammad (6849734)
Shahin Mahmud (300274789)

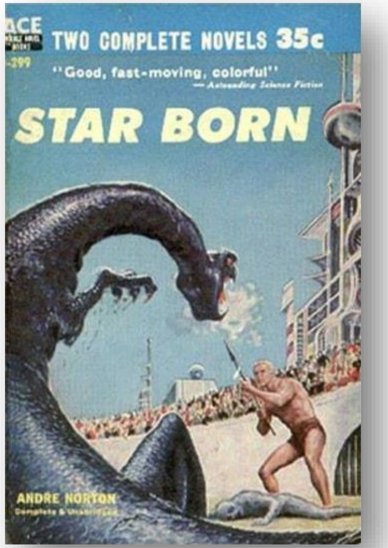


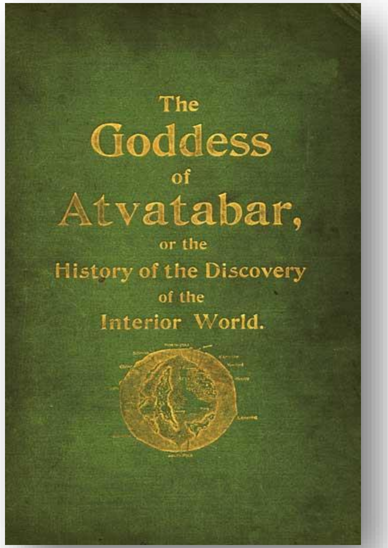
Text Classification

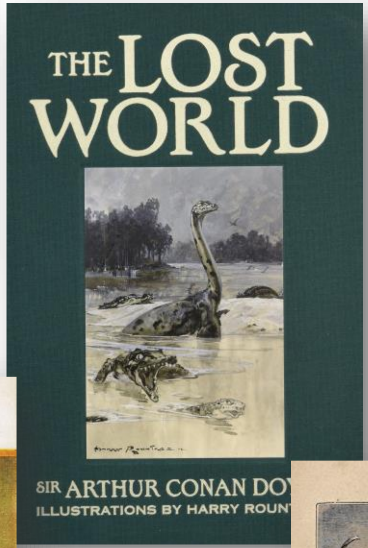
- Assignment of a document to one or more categories
- Supervised machine learning task
- Applications: spam filtering, readability assessments, etc.
- Goal: classify the author given a set of texts belonging to the same genre and language
  - Science fiction
  - English

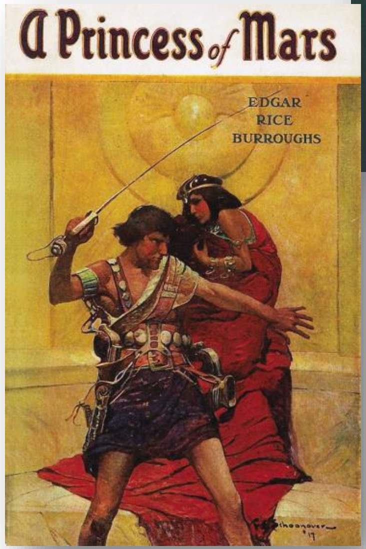
Texts

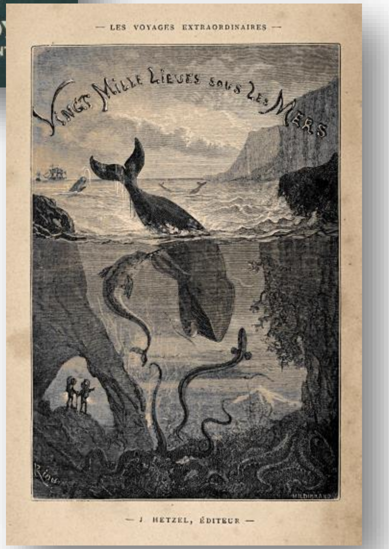
	Titles	Author	Year of Publication
1	Star Born	Andre Norton	1957
2	The Goddess of Atvatabar	William R. Bradshaw	1892
3	Twenty Thousand Leagues Under the Sea (slightly abridged)	Jules Verne	1872
4	A Princess of Mars	Edgar Rice Burroughs	1912
5	The Lost World	Arthur Conan Doyle	1912











Data Preparation

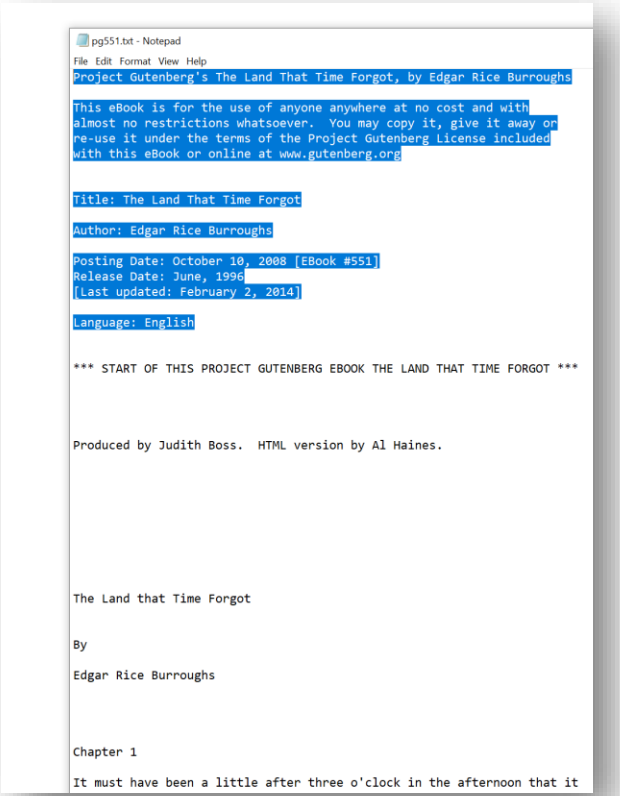
Goal

Prepare data for feature engineering

- Simplify
- Preprocess
- Tokenize
- Clean
- Sample

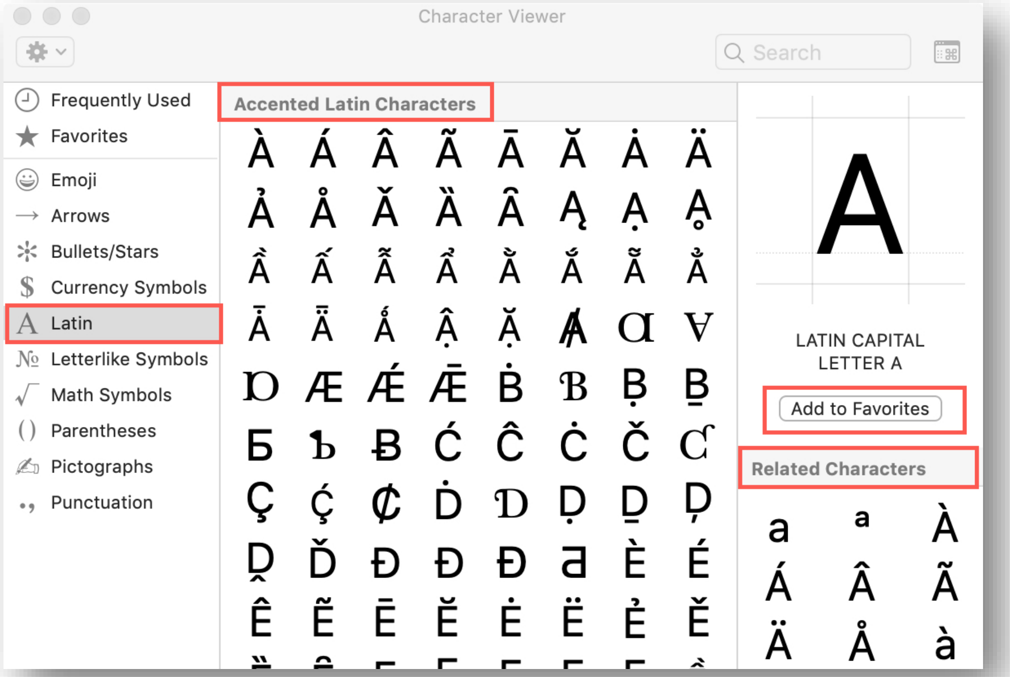
Simplify Text

- Used **regex** to find the start and end of the text to ignore copyright and license sections
  - Decreases computational overhead



Normalize Accented Characters

- Leveraged the unidecode library to transform accented characters into their base forms
  - "Café" becomes "Cafe"
  - Dimensionality reduction



Expand Contractions

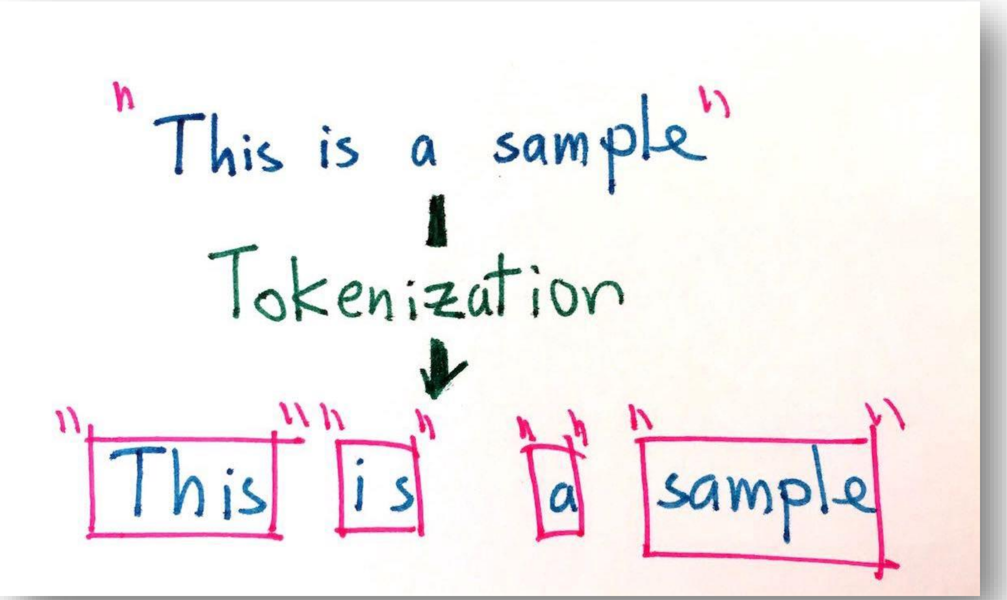
- Used the contractions library expand contractions
  - "You're" becomes "You are"
  - **Dimensionality reduction**
  - Distinct expansions are not always possible
    - Should "I'd" expand to "I had" or "I would"?
  - pycontractions is an alternative library
    - Employs Word Mover's Distance (WMD)

Common Contractions in English

aren't - are not	I'm - I am	that's - that is
can't - cannot	I've - I have	there's - there is
didn't - did not	isn't - is not	we're - we are
don't - do not	let's - let us	what's - what is
he'll - he will	she'll - she will	you'll - you will

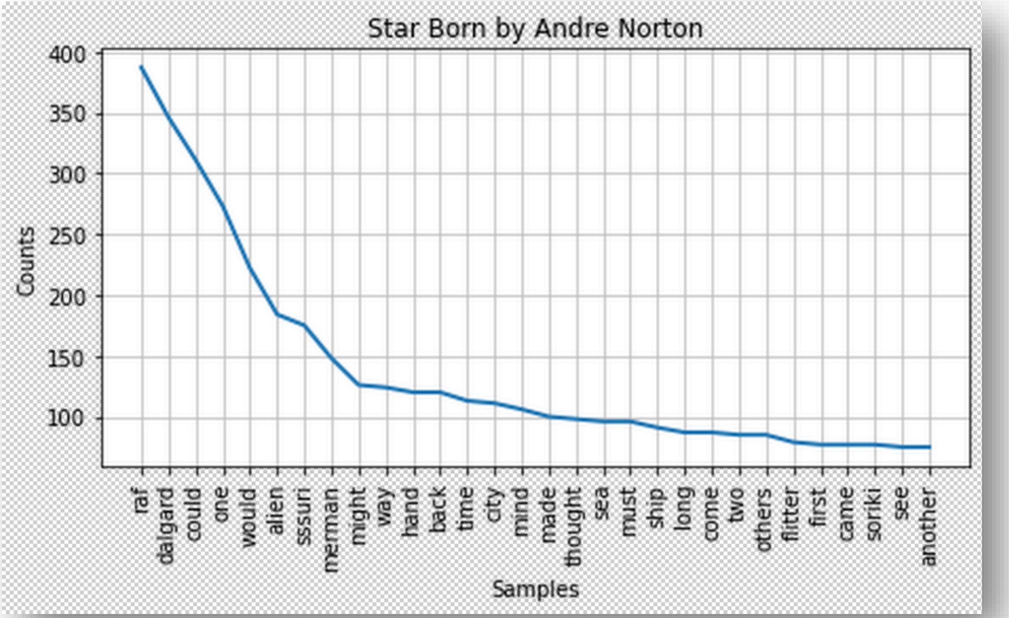
Tokenize Text

- Tokenized the text into words



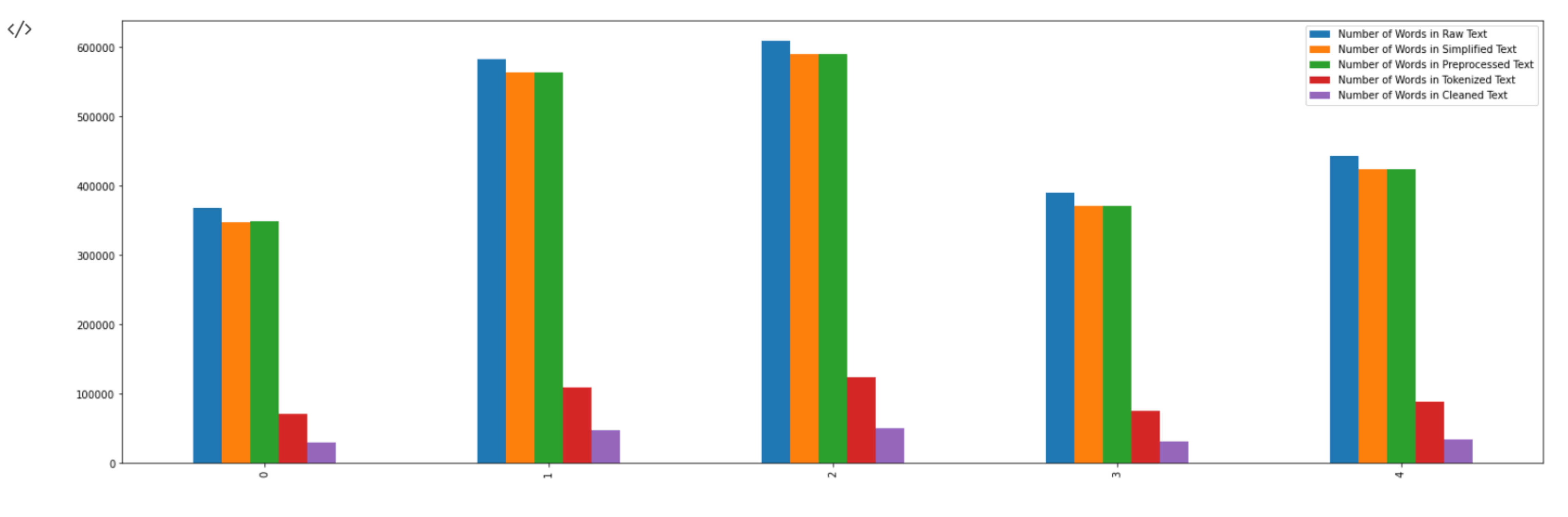
Clean Text

- Removed punctuation, numbers, special characters, etc.
- Converted text to lower case
- Removed stop words
- Performed lemmatization
  - Stemming is less resource-intensive
  - Opted for lemmatization since it is linguistically motivated, and we are dealing with a text classification problem



[9] ✓ 0.8s Python

	Authors	Titles	Number of Words in Raw Text	Number of Words in Simplified Text	Number of Words in Preprocessed Text	Number of Words in Tokenized Text	Number of Words in Cleaned Text
0	Andre Norton	Star Born	367551	348402	348660	71828	29333
1	William R. Bradshaw	The Goddess of Atvatabar	582554	563280	563659	109510	47696
2	Jules Verne	Twenty Thousand Leagues under the Sea (slightl...	609064	589919	590361	123803	50111
3	Edgar Rice Burroughs	A Princess of Mars	390249	371188	371227	75137	31870
4	Arthur Conan Doyle	The Lost World	443413	424311	424852	89354	34862



Number of Words in Raw Text
Number of Words in Simplified Text
Number of Words in Preprocessed Text
Number of Words in Tokenized Text
Number of Words in Cleaned Text
600000
500000
400000
300000
200000
100000
0
0
1
2
3
4

Sample Text

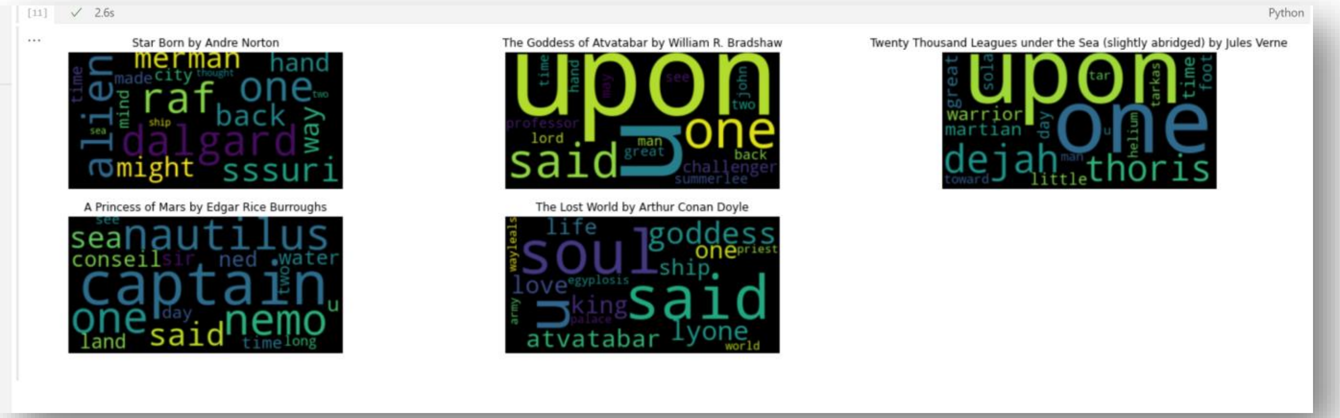
- Create samples of 100 words from the start to the end of the text
- Select 200 random samples from the set of samples for each text

	Cleaned Samples	Author
0	delicately ready flee first hint suspected bel...	Andre Norton
1	explain could one make plain feeling sensible ...	Andre Norton
2	stubbornly gray murmur wonstead went drone ach...	Andre Norton
3	seeming unconcern sssuri first intimation hunt...	Andre Norton
4	raf first reaction must still merman young str...	Andre Norton
...	...	...
995	must difficult one otherwise creature would co...	Arthur Conan Doyle
996	face flashed back went south america solitary ...	Arthur Conan Doyle
997	page disappointing however contained nothing p...	Arthur Conan Doyle
998	one indian group dragged forward edge cliff ki...	Arthur Conan Doyle
999	day sat late mcardle news editor explaining wh...	Arthur Conan Doyle

1000 rows × 2 columns

Clean Text (Advanced)

Removed the most common words

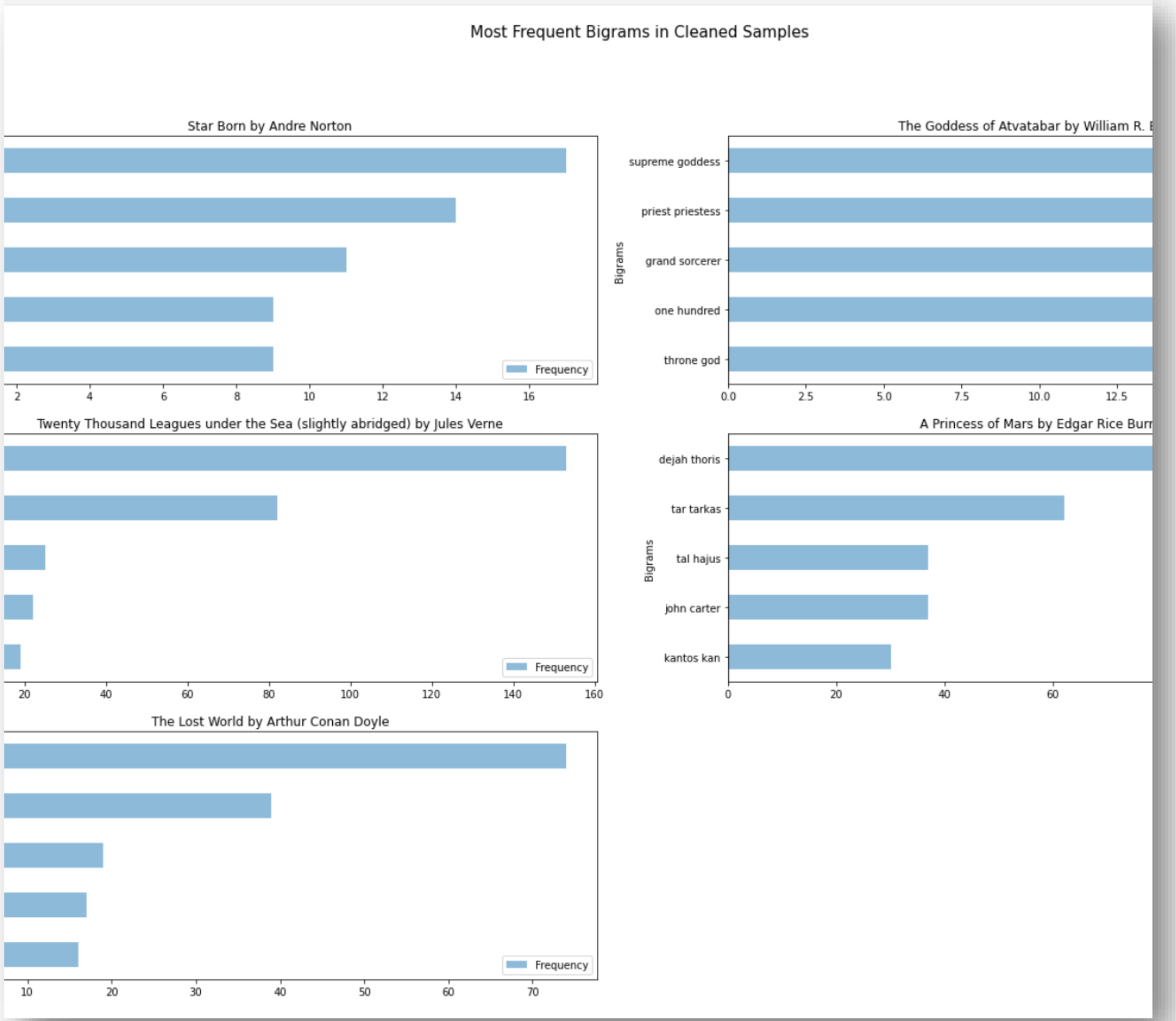


Feature Engineering

Goal

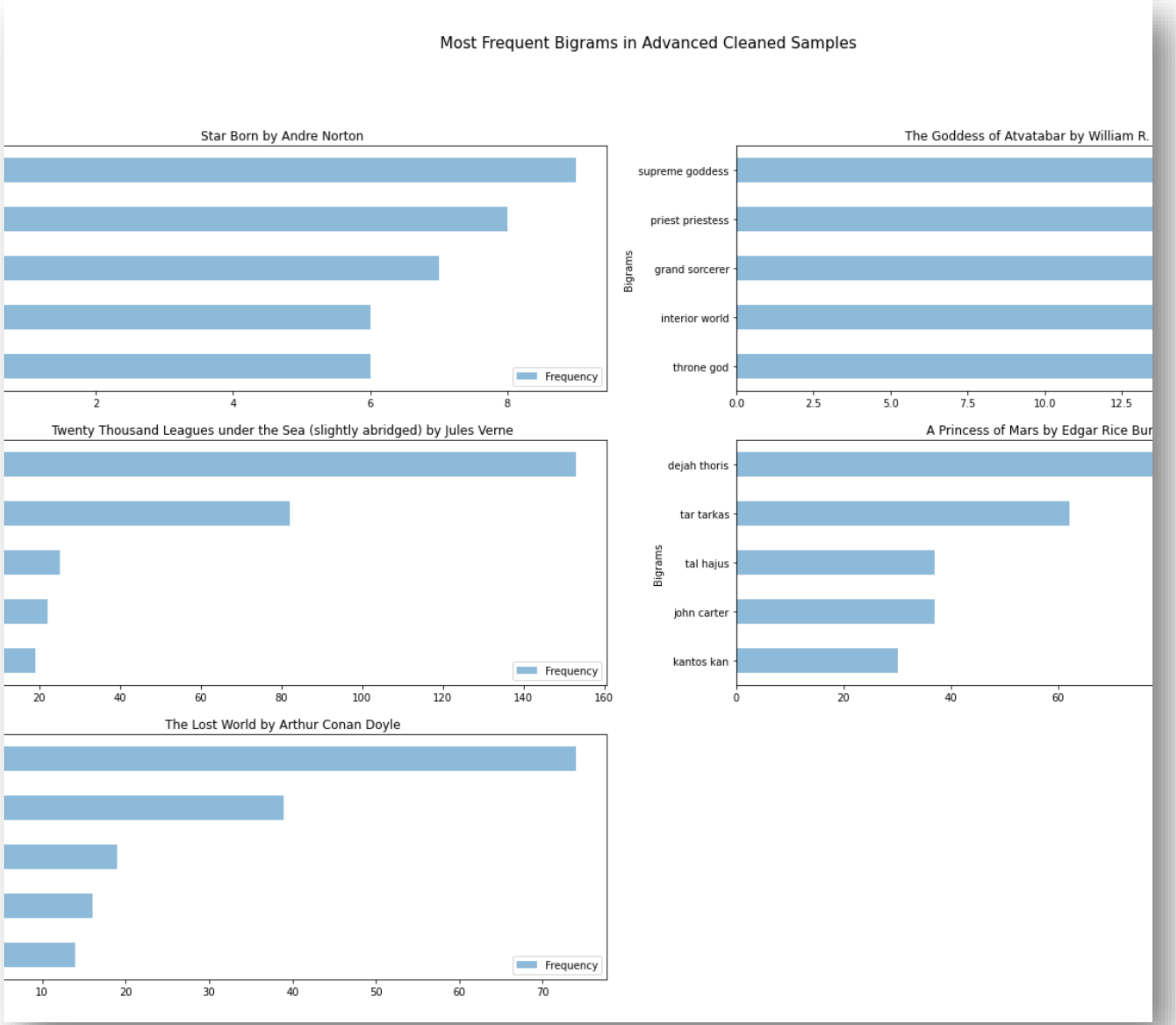
Create features for modeling

- ❑ n-grams
  - Predict the occurrence of a word based on the occurrence of its n – 1 words
- ❑ Bag-of-words (BOW)
  - Describe the occurrence of words using fixed-length vectors
- ❑ Term frequency-inverse document frequency (TFIDF)
  - Reflect how relevant a word is



n-grams

Most frequent bigrams in cleaned samples



n-grams

Most frequent bigrams in cleaned (advanced) samples

BOW & TFIDF

- BOW
  - CountVectorizer
  - Fit
  - Transform
- TFIDF
  - TfidfTransformer (use vector generated using CountVectorizer)
  - Fit
  - Transform

Number of splits = 10

Test size = 0.1

Random state = 0

Modeling & Analysis

Goal

Train and evaluate models for prediction

- Use pipelines to simplify process

Support Vector Machine

- Sets the best decision boundary between vectors that belong to the given text and those that do not

Decision Tree

- Builds a decision tree based on answers to yes-no questions

KNeighbor

- Implements classification based on voting by nearest k-neighbors

Random Forest

- Uses ensemble learning and decision trees

Multinomial Naïve Bayes

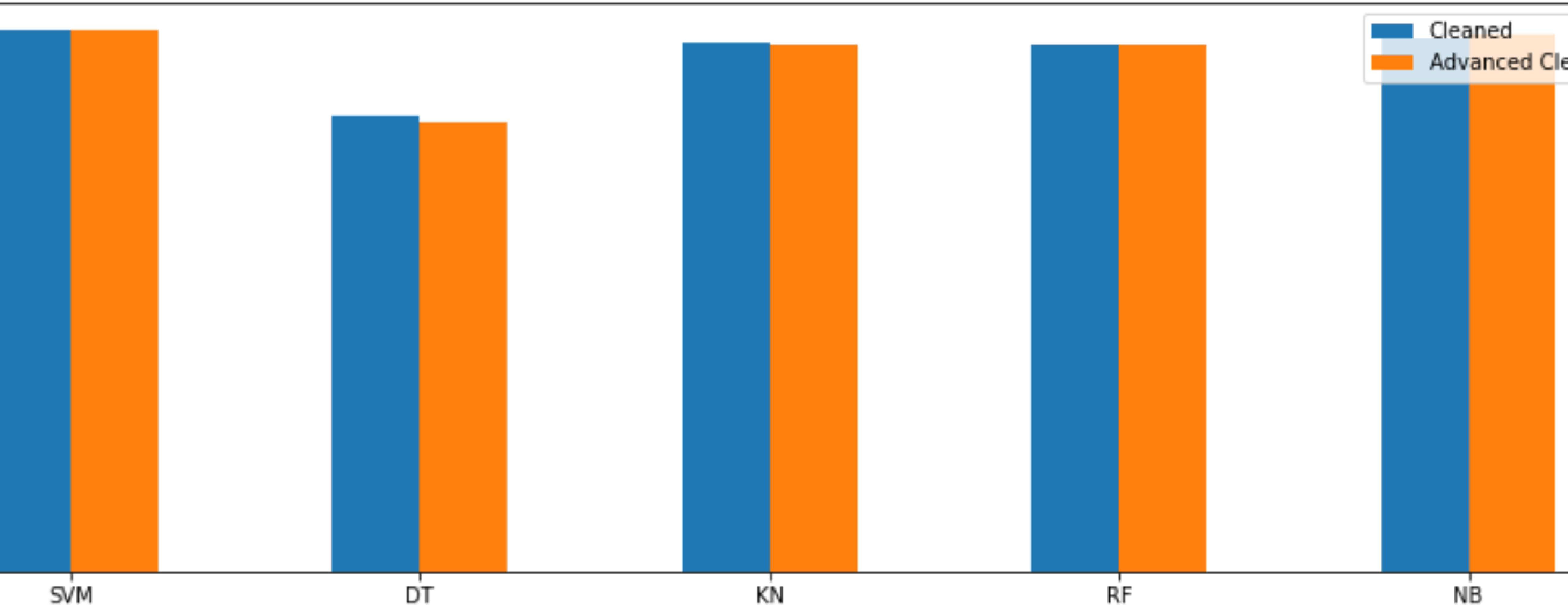
- Assumes the effect of a certain feature is independent from other ones

```
 Vector Machine (SVM)


leaned Samples
= Pipeline([("bow", bow_cln_tr),
                ("tfidf", tfidf_cln_tr),
                ("clf", SGDClassifier(loss="hinge", penalty=
fleSplit(n_splits=n_splits, test_size=test_size, random_
cross_val_score(pipeline, labeled_texts_df["Cleaned Samp
amp_type_to_avg_acc["SVM + Cleaned Samples"] = scores.me

dvanced Cleaned Samples
= Pipeline([("bow", bow_adv_cln_tr),
                ("tfidf", tfidf_adv_cln_tr),
                ("clf", SGDClassifier(loss="hinge", penalty=
fleSplit(n_splits=n_splits, test_size=test_size, random_
cross_val_score(pipeline, labeled_texts_df["Advanced Cle
amp_type_to_avg_acc["SVM + Advanced Cleaned Samples"] =
```

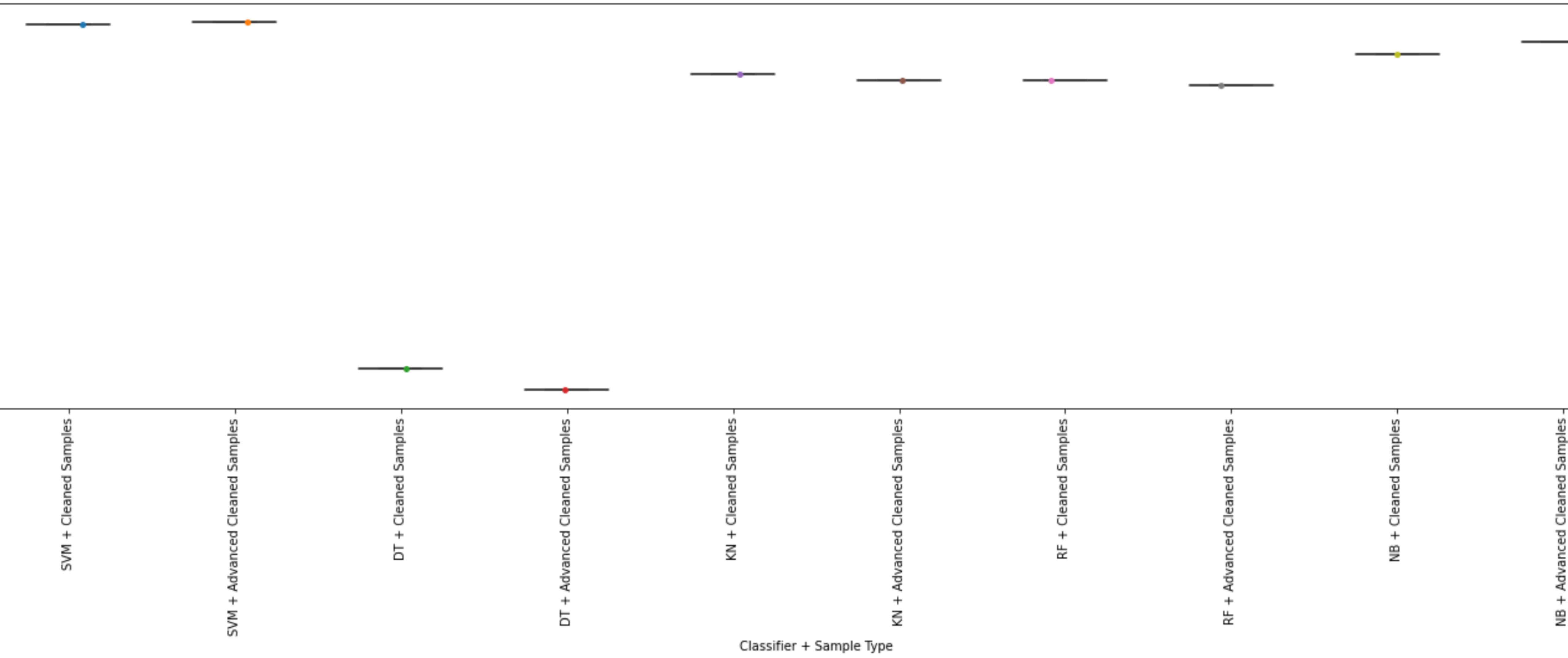
Pipelines

- BOW + TFIDF + classifier
- ShuffleSplit
- cross_val_score

	Classifier + Sample Type	Average Accuracy
0	SVM + Cleaned Samples	0.997
1	SVM + Advanced Cleaned Samples	0.998
2	DT + Cleaned Samples	0.838
3	DT + Advanced Cleaned Samples	0.828
4	KN + Cleaned Samples	0.974
5	KN + Advanced Cleaned Samples	0.971
6	RF + Cleaned Samples	0.971
7	RF + Advanced Cleaned Samples	0.969
8	NB + Cleaned Samples	0.983
9	NB + Advanced Cleaned Samples	0.989



Cleaned
Advanced Cle
SVM
DT
KN
RF
NB


SVM + Cleaned Samples
SVM + Advanced Cleaned Samples
DT + Cleaned Samples
DT + Advanced Cleaned Samples
KN + Cleaned Samples
KN + Advanced Cleaned Samples
RF + Cleaned Samples
RF + Advanced Cleaned Samples
NB + Cleaned Samples
NB + Advanced Cleaned Samples
Classifier + Sample Type

Prediction

Goal

Test model and perform error analysis

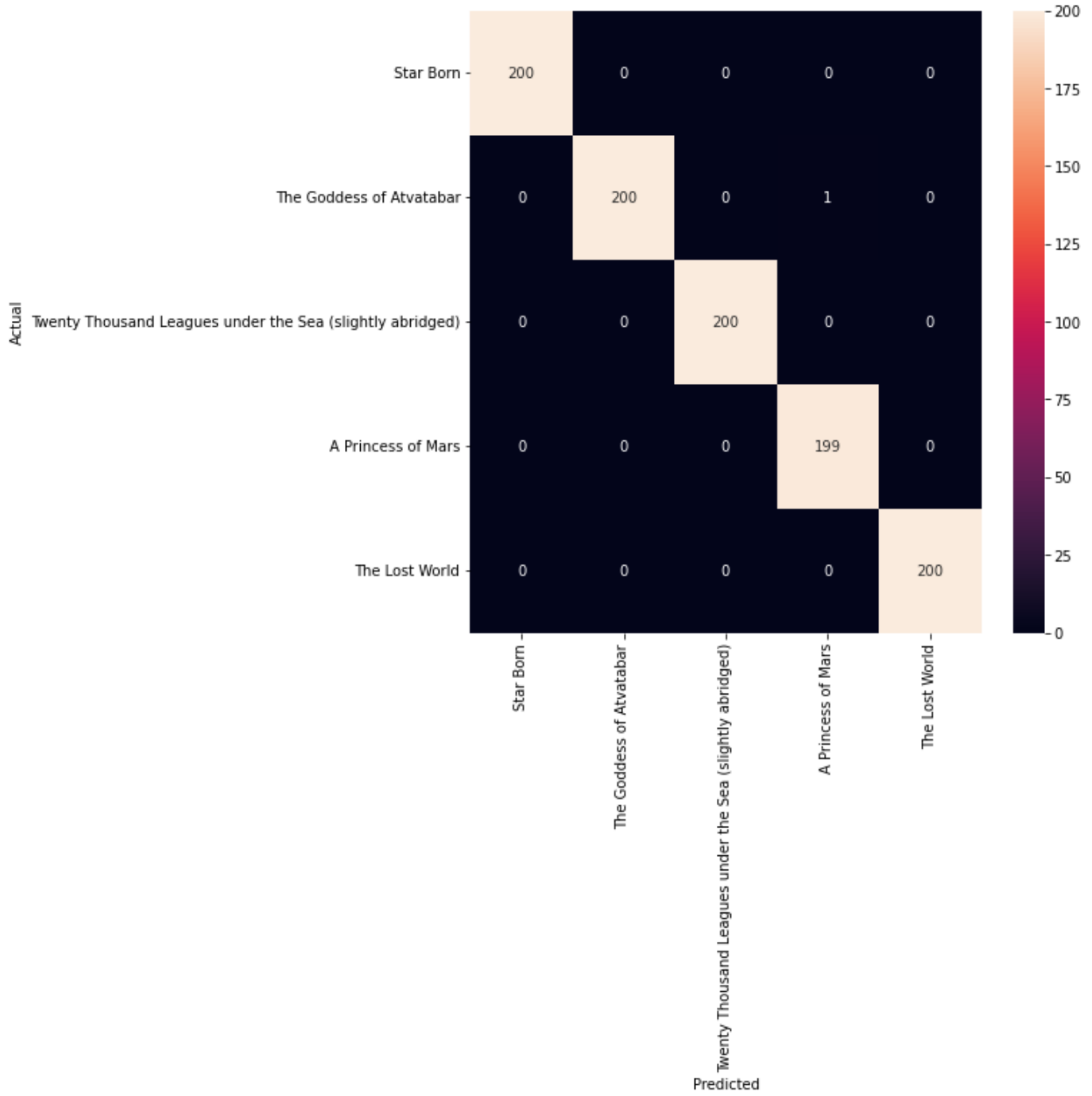
- Champion models: Naïve Bayes and SVM

- ❖ Classification Report
- ❖ Confusion Matrix

Champion 1: Naïve Bayes

Report

```
NB + Cleaned Samples
             Prediction       Actual  Predicted Wrong
877  Arthur Conan Doyle  Jules Verne            False
NB + Advanced Cleaned Samples
     Prediction             Actual  Predicted Wrong
70  Jules Verne  William R. Bradshaw            False
NB + Cleaned Samples
                     precision    recall  f1-score   support

      Andre Norton       1.00      1.00      1.00       200
 Arthur Conan Doyle       1.00      1.00      1.00       201
Edgar Rice Burroughs       1.00      1.00      1.00       200
       Jules Verne       0.99      1.00      1.00       199
 William R. Bradshaw       1.00      1.00      1.00       200

          accuracy                           1.00      1000
         macro avg       1.00      1.00      1.00      1000
      weighted avg       1.00      1.00      1.00      1000

NB + Advanced Cleaned Samples
                     precision    recall  f1-score   support

      Andre Norton       1.00      1.00      1.00       200
 Arthur Conan Doyle       1.00      1.00      1.00       200
Edgar Rice Burroughs       1.00      1.00      1.00       200
...
          accuracy                           1.00      1000
         macro avg       1.00      1.00      1.00      1000
      weighted avg       1.00      1.00      1.00      1000
```



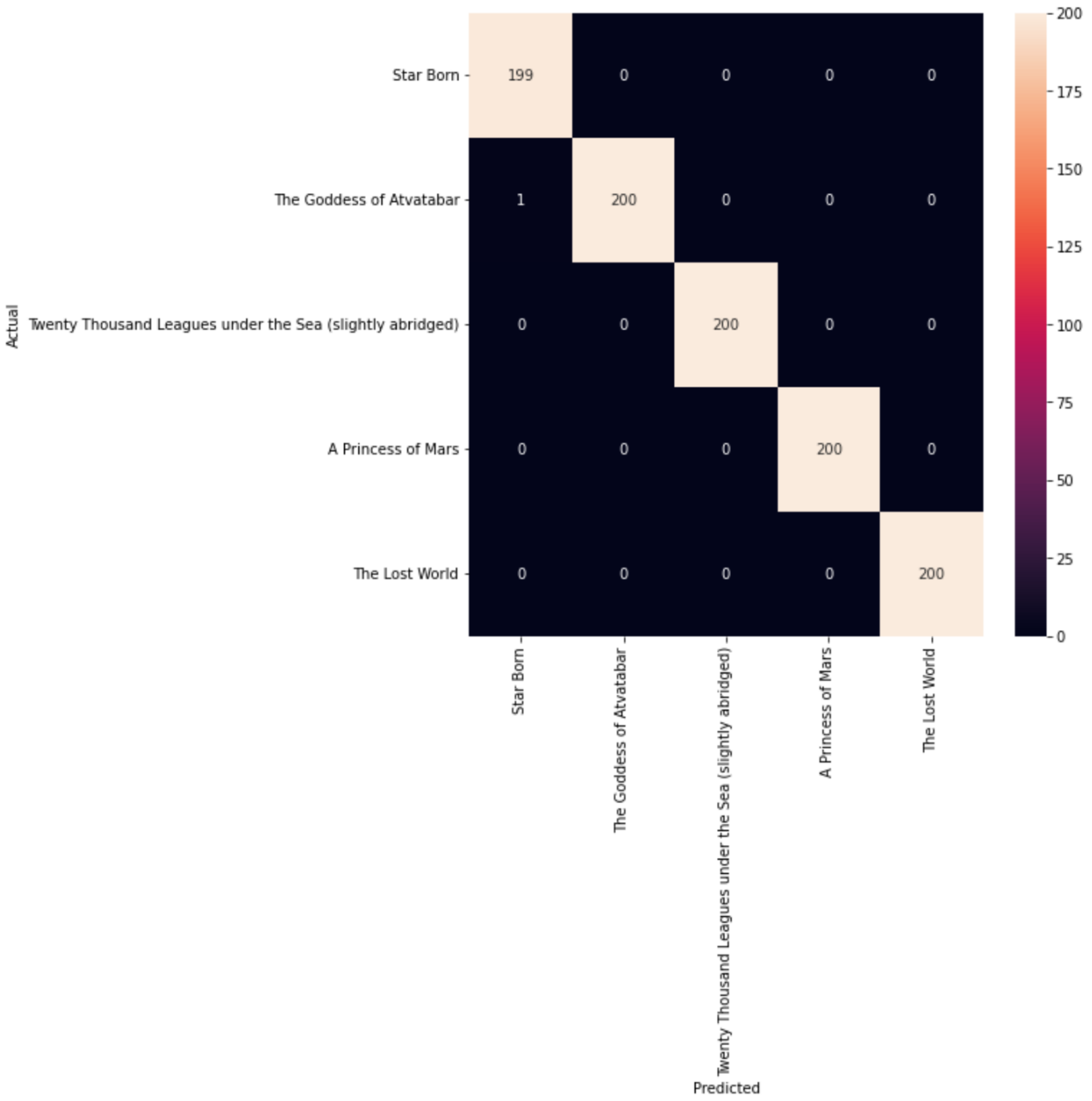
Champion 1: Naïve Bayes

Confusion Matrix

```
SVM + Cleaned Samples
Empty DataFrame
Columns: [Prediction, Actual, Predicted Wrong]
Index: []
SVM + Advanced Cleaned Samples
             Prediction        Actual  Predicted Wrong
308  Arthur Conan Doyle  Andre Norton            False
SVM + Cleaned Samples
                      precision    recall  f1-score   support

        Andre Norton       1.00      1.00      1.00       200
  Arthur Conan Doyle       1.00      1.00      1.00       200
Edgar Rice Burroughs       1.00      1.00      1.00       200
         Jules Verne       1.00      1.00      1.00       200
 William R. Bradshaw       1.00      1.00      1.00       200

            accuracy                           1.00      1000
           macro avg       1.00      1.00      1.00      1000
        weighted avg       1.00      1.00      1.00      1000

SVM + Advanced Cleaned Samples
                      precision    recall  f1-score   support

        Andre Norton       0.99      1.00      1.00       199
  Arthur Conan Doyle       1.00      1.00      1.00       201
...
            accuracy                           1.00      1000
           macro avg       1.00      1.00      1.00      1000
        weighted avg       1.00      1.00      1.00      1000
```

Champion 2: SVM

Report



Champion 2: SVM

Confusion Matrix

Conclusion

- Champion classifiers: Naïve Bayes & SVM
- Advanced cleaning performed as well as "normal" cleaning
- Decision Tree classifier performed badly potentially due to a high-dimensional feature space
- Naïve Bayes classifier performed the best and was simple to use as well
- Will explore ways to analyze and account for bias in samples