Project 4 Summary:
Audience Reviews of Retold Stories

Project Design:

I used natural language processing and unsupervised learning to generate topics, which I used as features in a supervised classification model to predict whether a review would be positive or negative. Specifically, I analyzed reviews for the following five Disney live-action remakes:

- Cinderella
- The Jungle Book
- Beauty and the Beast
- Aladdin
- The Lion King

This model could be used by Disney to identify features about the films that the audience likes or dislikes. Based on these features, Disney can decide what to continue or adjust in the production of their future remakes.

I tested the following natural language processing steps for this project:

- Preprocessing: Using spaCy, I customized stop words (ex. removing words like "film" or "movie" and adding back words like "not" or "no", performed lemmatization, converted words to lowercase, removed punctuation and numbers, and tokenized emojis
- Vectorization & dimensionality reduction: A combination of CountVectorizer or TF-IDF and LDA or LSA or NMF
  - The combination that produced the best results (i.e., good topic separation into intuitive and distinct groupings for unsupervised learning AND the best ROC AUC score for supervised learning was TF-IDF and NMF)
  - Vectorization: I tuned this by adding more stop words, increasing the ngram range to create bigrams, trigrams, etc., and filtering by minimum document frequency (min_df) to exclude words that only appeared in less than 5% of the documents (reviews) of my dataset
  - Dimensionality reduction: I tuned this by adjusting the number of components (topics) and viewing the quality of separation of the topics and the resulting ROC AUC score for the subtrain and validation set.
    - I tried using a model called HDP, which supposedly can help select the optimal number of topics—but I did not have enough time to try to get it to work; the only model I could get running took over 12 hours and I had to stop running it and didn't see the end result.

For the supervised learning part of my project, I did the following:

- Scraped the ratings from each review (IMDB uses a 10-point rating system while Rotten Tomatoes rates on a scale from 1-5—with the option of half-stars)
  - Dropped reviews that did not have a rating from my dataset
  - Converted ratings with 1-4 from IMDB or 1-2 from Rotten Tomatoes to "negative" sentiment
  - Converted ratings with 7-10 from IMDB or 4-5 from Rotten Tomatoes to "positive" sentiment
  - Dropped reviews that had a "neutral" rating (i.e., 5-6 from IMDB or 3 from Rotten Tomatoes)
- The dataset was imbalanced (87% were positive reviews). I used the following options to balance my dataset:
  - Random oversampling
  - SMOTE oversampling
  - Adasyn oversampling
  - Use the class weight = "balanced" parameter in Logistic Regression, which I believe does some sort of undersampling method. This performed the best; the other options mostly returned overfit results that had lower ROC-AUC scores.
- Prepared the data:
  - Train-test split: I ran my models using a single split of train, validation, and test because I wanted to ensure I followed the correct order of steps of standard-scaling (when necessary), performing oversampling methods, and keep oversampling steps separate from my cross-validation and test data—and I felt CV was making that more confusing. Because my dataset was relatively large enough to do a single train-test split, then split train into "subtrain" and validation sets, I proceeded with this path. However, since this approach is not using k-folds and cross-validation and averaging out the performance metric scores, my measures of model performance may not be as robust.

Kristen Tokunaga

Project 4 Summary:
Audience Reviews of Retold Stories

  - Normalize: On each train and test set, I used vectorization, dimensionality reduction, and normalized on their own separate dataset. Normalization should help reduce the dependence on the length of the review.
  - StandardScale: I fit the standardscaler to my trained set of vectorized/dimensionality-reduced/normalized topics, and transformed the train set and test set
- The classification model I used was logistic regression, which has several benefits for NLP projects:
  - It works well on sparse datasets
  - The coefficients are relatively interpretable:
    - Larger coefficients have bigger correlation with the target variable
    - Positive coefficients are positively correlated with positive reviews; negative coefficients are negatively correlated with positive reviews
- Parameter tuning
  - I tuned my logistic regression parameters using hyperopt (Bayesian optimization)
  - For tuning NLP methods and number of topics, I did this manually as I could not figure out a pipeline for accomplishing this and tuning my logistic parameters while performing all the necessary steps of normalization and standardscaler

My best model was Logistic Regression with the following parameters:

```
LogisticRegression(C=0.001,penalty='l2',solver='saga',class_weight='balanced',
random_state=41,max_iter=500)
```

I used ROC AUC for my metric because it is less sensitive to imbalanced classes and if needed, I can tune the threshold to produce a more desirable confusion matrix of false positives vs. false negatives. Because the outcome of the problem I'm trying to solve does not have a clear advantage or disadvantage of minimizing false negatives or false positives, I used the standard threshold of 0.5. The ROC AUC of my final model was 0.72.

Below are the coefficients for the features (topics) of my best model:

Coefficients	Topics (Features)
-0.54	not, original, like, well, make, remake, story, feel, character, new, version, song, time, animate, voice, amaze, scene, look, lion, classic, no, little, animal, only, enjoy, cgi, way, cast, beautiful, real
0.54	love, amaze, kid, smith, awesome, beautiful, old, music, song, genie, story, lion, aladdin, time, new, version, make, real, character, animal, little, fun, actor, loved, way, cast, enjoy, animate, look, well
0.50	great, smith, loved, genie, music, cast, story, fun, awesome, amaze, classic, kid, enjoy, remake, aladdin, act, actor, song, make, new, beautiful, little, old, character, voice, time, cgi, animate, version, well
0.09	good, smith, genie, enjoy, music, aladdin, make, act, story, fun, actor, cast, version, kid, song, little, animate, awesome, cgi, character, classic, old, amaze, scene, feel, time, only, voice, beautiful, look

I was not able to get a lot of clear separation among my topics, which forced me to use fewer topics. I also had to dig deeper into the original reviews to get a better understanding of HOW reviews were talking about the features (e.g., CGI and "not" and "original"), and test my model on one-word "reviews" to get an idea of whether CGI was associated with more positive or negative reviews.

Based on the negative associations with CGI and "not" and "original" and positive correlations with "[will] smith", I tailored my recommendations to focus on these features:

- CAST: Choice in casting helps. Famous actors can draw a big audience and stir a lot of conversation.
- CGI: We may want to re-think how we're using CGI. Among negative reviews of CGI: audience reviewers said it was the only good thing about the movie and/or CGI made it harder to connect emotionally with the character—due to lack of expression.
- NOT ORIGINAL: It appears that the audience wants something a little different with these films. Flip the script a little, and BEE original.

Kristen Tokunaga

Project 4 Summary:
Audience Reviews of Retold Stories

Tools

- Web-scraping
  - BeautifulSoup
  - Selenium
- Python
  - Data Cleaning & Analysis: Pandas
- NLP
  - spaCy, genism
  - SKLearn:
    - Vectorization
      - CountVectorizer, TF-IDF
    - Dimensionality Reduction
      - NMF, LSA, LDA, ~~HDP~~

Data

I obtained my data by web-scraping over 44,000 reviews from Rotten Tomatoes and IMDB. I selected these 5 films because they were most similar to their original animated versions—as opposed to Maleficent (which was told from the villain's point of view), or Tim Burton's versions of Alice in Wonderland or Dumbo. Additionally, less recent films (e.g., 101 Dalmatians) was excluded because it was released in 1996 and audience taste/opinions, film technology, and use of online platforms to post reviews was very different back then.

I used BeautifulSoup and Selenium to scrape the text of the reviews and the ratings from Rotten Tomatoes and IMDB. Using BeautifulSoup was a good challenge for me because 1. for Project 2, I only used Selenium to scrape for my data and 2. This time, I was able to do more advanced functions with Selenium to click between pages (i.e., click the "next" button) and add error exceptions for when I reached the last page to break from the loop instead of erroring out and losing all the data.

What I Would Do Differently

If I had more time, I would have liked to:

- Explore HDP models and other NLP optimization techniques further. I was hoping to get better separation of topics and a much better ROC AUC score. Several things I think could have helped with that are:
  - Setting up a pipeline to test my models more efficiently –with the NLP, normalization, and standardscaler steps included in the pipeline
  - Expanding my stop word list further
  - Testing other classification model—to see if they would have worked better with my topics, beyond logistic regression. Most examples I found online tried logistic regression and perhaps SVM, but I'm not sure whether others would have been appropriate or performed better on this dataset/problem
- Troubleshoot the limitations of my model
  - When testing several unseen reviews (e.g., fake reviews that I wrote to test my logistic regression model), I found that longer reviews (i.e., > 1 word) appeared to be correlated with a positive review while 1-word reviews appeared to result in a negative classification. I'm still unsure why this is the case, since I normalized my train and test datasets—which I believe should have accounted for varying lengths of the documents (reviews).

Future Work

- Cost-benefit analysis of hiring famous actors (e.g. Will Smith)
- Deeper dive into negative reviews, to see what specifically audiences would like to be "original" vs. kept the same in future remakes (e.g., new music, plot changes)
- Explore different data sources:
  - Expand scope to analyze social media comments (e.g., Twitter, Facebook)
  - Compare other Disney live-action remakes (e.g., Maleficent, Tim Burton's films like Alice in Wonderland or Dumbo)
  - Expand scope to analyze sentiments of international audience—according to Box Office Mojo, over 60% of box office revenues came from international audiences for many of the Disney live-action remakes. It would be interesting to learn these audiences think of the films to predict whether future remakes will be as well-received.
- Once optimized, real-time modeling
  - I would like to get to a point of being able to perform real-time modeling of audience sentiment to monitor the company brand and product trends

Kristen Tokunaga