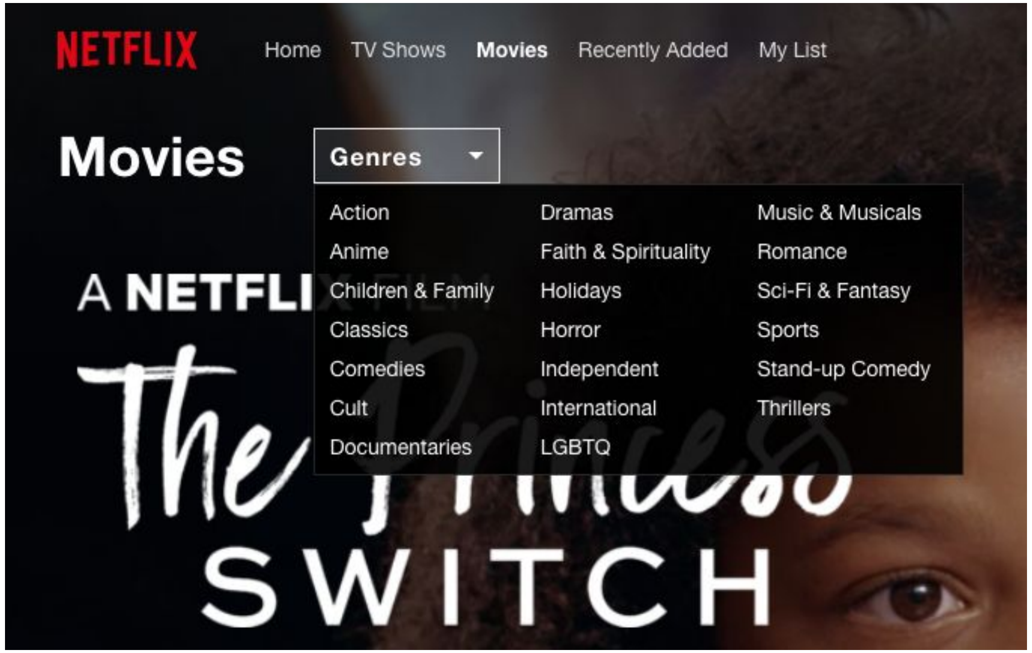
Data Mining Final Project

Predicting Movie Genre from Plot Summaries

Team Members:
Audrey Salerno, Helen Wang, Rich Poole

Business Problem

- Consumers select the genres first based on personal interest before watching any movies on Netflix or Hulu
- Companies want to avoid manually tagging movie genres
- Movies are assigned with one or more genres
- So we aim to automate the process to save human effort and to improve accuracy



Dataset Description

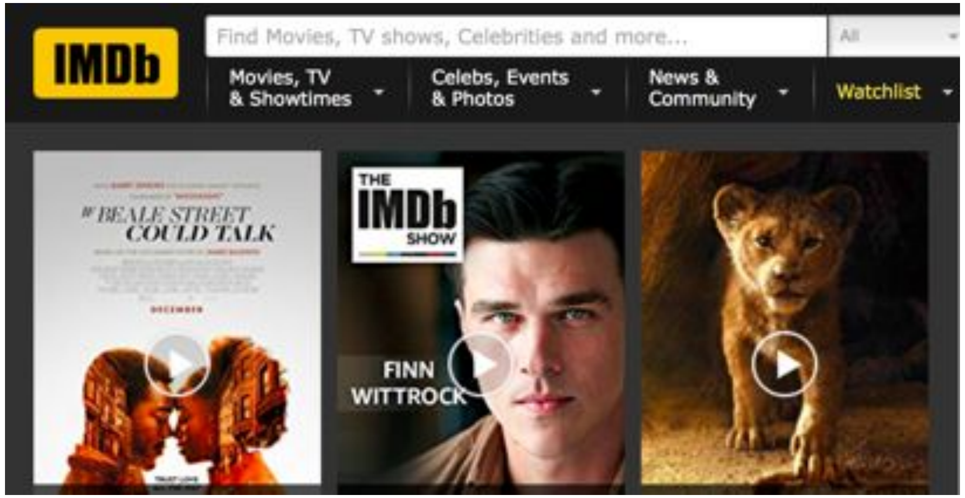
Used omdb API to get IMDB Movie data for 2000-2017

Factors:

- Movie plot (long version)
- Director names
- Movie Rating (G, PG, PG-13, R)

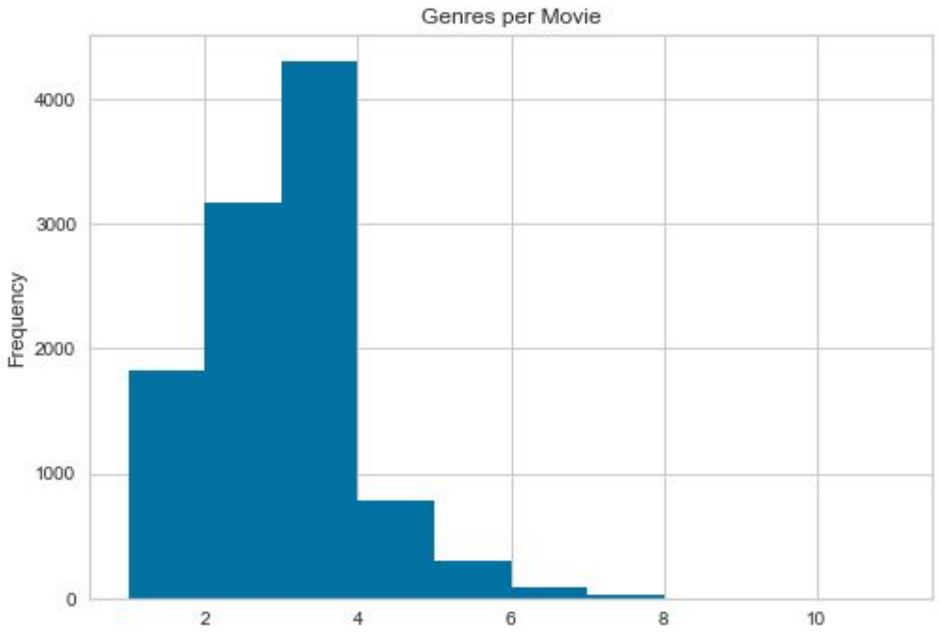
Output:

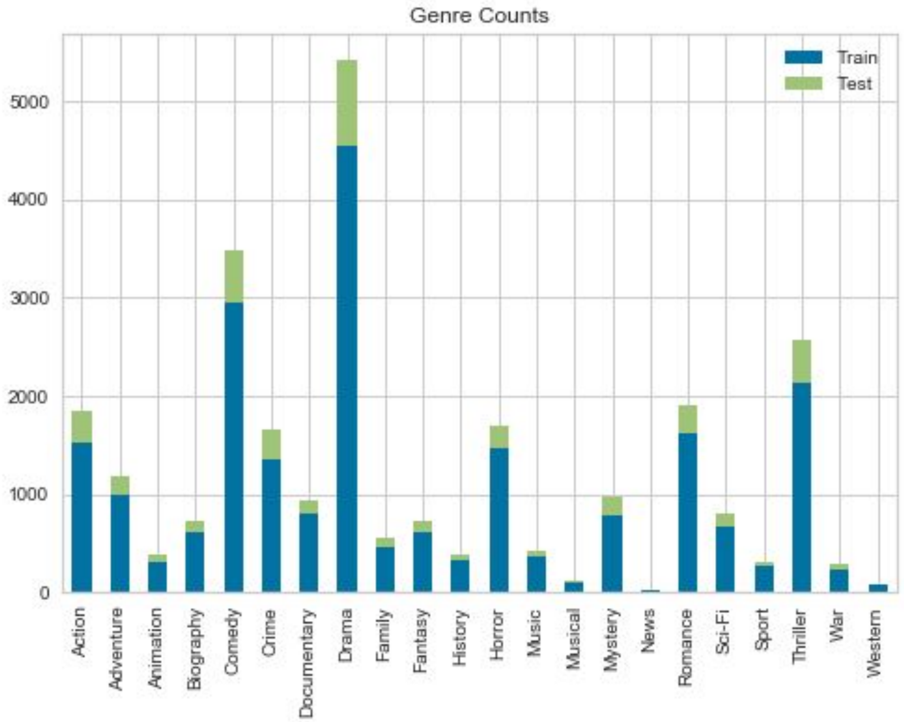
- Multi-label Genre (each movie can have multiple genres)



Genres

- Most movies have more than one genre.
- The genres are not balanced





Drama
Comedy
Thriller
Romance
Action
Horror
Crime
Adventure
Documentary
Mystery
Sci-Fi
Fantasy
Biography
Family
Music
History
Animation
Sport
War
Musical
Western
News

Text Data Pre-Processing

1. Add Directors and Movie Rating to plot to include in the tokenized text.
2. Remove special characters and numbers
3. Lemmatization or Stemming
   - Reduce inflections or variant forms to base form
     - am, are, is → be | car, cars, car's, cars' → car
   - Convert words to its root word
     - walking → walk | activate → activ
4. Remove Stop Words:
   - Remove words that are most common, short function words but have no actual meaning.
     - Ex: is, be, the, that, on, which, where, at
5. Vectorization

Approach

Fundamentally, predicting movie genres is a multi-label problem. Each movie can have multiple genres.

We tried both supervised and unsupervised approaches. The supervised approach tried to predict the exact list of genres, while the unsupervised approach tried to group movies into 5 high-level genre groups.

Exact Multi-Label Predictions

- SVC, Logistic Regression, Decision Tree, Random Forest, KNN

Grouped Multi-Label Predictions

- LDA Model (unsupervised)

Latent Dirichlet Allocation (LDA)

- **What is LDA? How does LDA work?**
  LDA generates topics based on word frequency from a set of documents.

- **Motivation of using LDA Topic Modeling**
  Use LDA model to classify plot summaries in the movie dataset to a particular topic (genre).

- **Goal:** Carried out Topic Modeling on each movie plot, obtained the percentage distribution of each movie for each of genre.

- **Build the Model**
  - Create the Document-Word matrix - CountVectorizer
  - Apply LDA model with sklearn - Choose number of topics

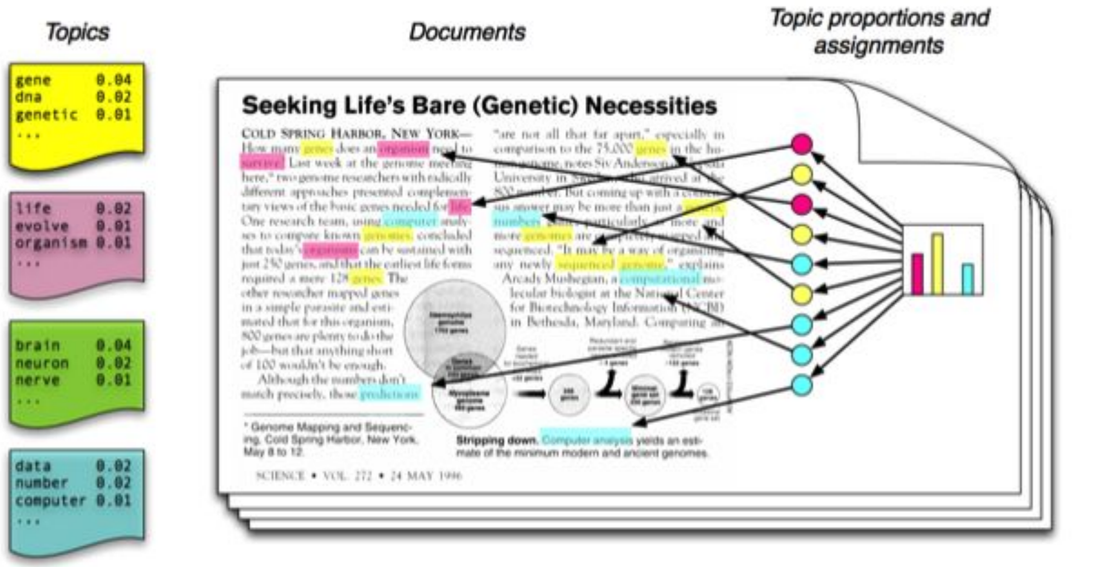
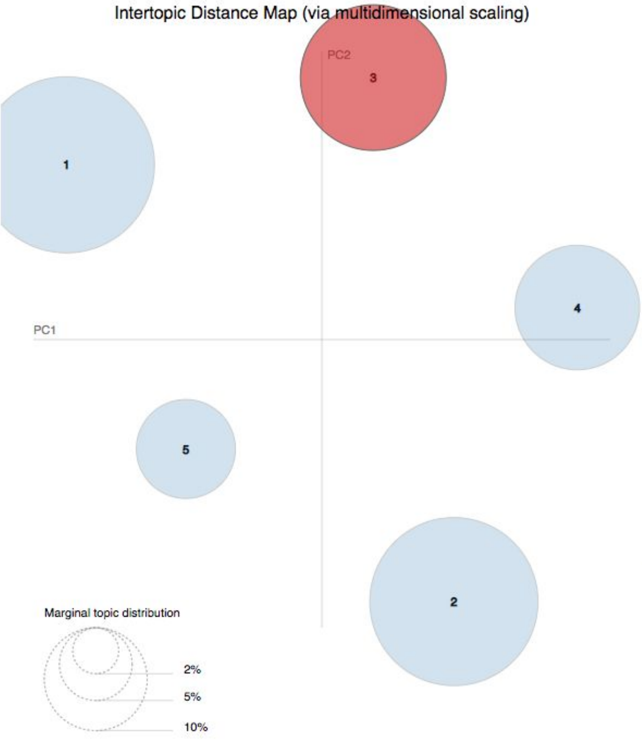
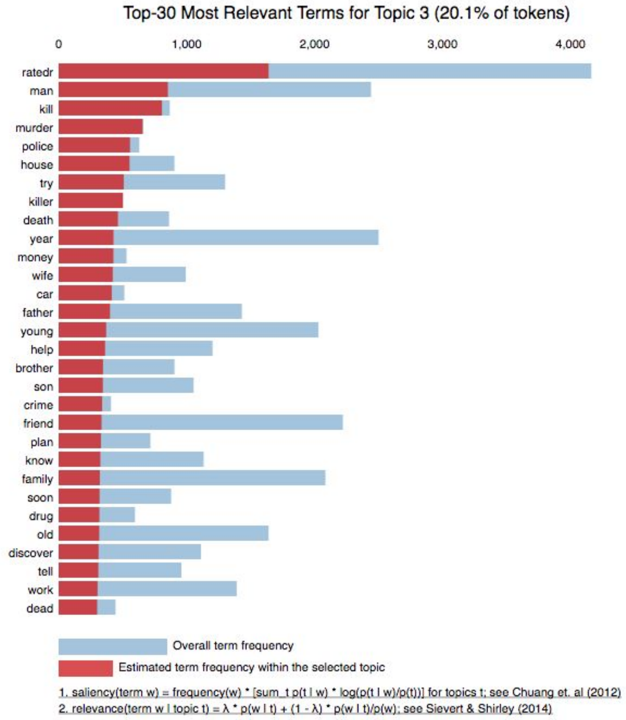


Figure source: Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.





Topic 0

Drama	3064
Comedy	1843
Thriller	1175
Romance	1045
Action	794
Crime	790
Horror	696

Topic 1

Drama	4014
Comedy	2822
Romance	1751
Thriller	1479
Horror	1078
Crime	900
Action	654

Topic 2

Drama	3081
Thriller	2111
Comedy	1831
Crime	1520
Action	1345
Horror	1317
Romance	865

Topic 3

Drama	3897
Comedy	2372
Thriller	1937
Horror	1430
Action	1492
Adventure	1102
Crime	1010

Topic 4

Drama	1748
Thriller	1043
Comedy	1050
Action	991
Crime	802
Horror	466
Adventure	400

- What information can we extract from these most relevant terms?
- Originally, we want to manually encode Topic # 0 - 4 into different genre based on most relevant words
- Use the model to predict the genre(s) of each movie by creating probabilities associated with particular label.

title	Topic0	Topic1	Topic2	Topic3	Topic4	threshold	dominant_topic
Kate & Leopold	0.00	0.59	0.00	0.40	0.00	0.1	Topic1, Topic3
Glitter	0.76	0.14	0.07	0.03	0.00	0.1	Topic0, Topic1
The Attic Expeditions	0.01	0.01	0.39	0.59	0.01	0.1	Topic2, Topic3
Chinese Coffee	0.00	0.24	0.05	0.62	0.08	0.1	Topic1, Topic3

New Approach: Map each topic with original genres and check how the genres correspond to different topic

title	Comedy	Drama	Crime/Thriller	Action	Romance
Kate & Leopold	0.432621	58.566830	0.429924	40.142111	0.428513
Glitter	75.545718	13.796856	7.181798	3.168724	0.306904
The Attic Expeditions	0.718987	0.733275	38.649658	59.179923	0.718156
Chinese Coffee	0.484723	24.396041	5.107510	62.255054	7.756672
The Dancer Upstairs	66.970900	0.314591	10.070942	22.329038	0.314529

title	Genre	genre
Kate & Leopold	Comedy/Romance, Action/Adventure	[Comedy, Fantasy, Romance]
Glitter	Drama, Comedy/Romance	[Drama, Music, Romance]
The Attic Expeditions	Crime, Action/Adventure	[Comedy, Horror, Mystery]
Chinese Coffee	Comedy/Romance, Action/Adventure	[Drama]
The Dancer Upstairs	Drama, Crime, Action/Adventure	[Crime, Drama, Thriller]
Don's Plum	Comedy/Romance	[Comedy, Drama]
Heavy Metal 2000	Comedy/Romance, Crime, Action/Adventure	[Animation, Action, Adventure]
State and Main	Drama	[Comedy, Drama]
Vulgar	Drama, Comedy/Romance	[Crime, Drama, Thriller]
Chicken Run	Action/Adventure	[Animation, Adventure, Comedy, Drama, Family]

- **Label each topic into different genres**
- **Classify the movie genre(s):** See which topic has the highest contribution to that document and assign it.
- **Limitations:** Sacrificing total number of genres by manually encode topic number into genres.
- Can we use LDA generated output into supervised machine learning model?

Exact Multi-Label Predictions Approach

Inputs:

- TF-IDF matrix of the processed plot data mentioned earlier. 1 & 2 gram terms. ~11K terms.

Dimensionality Reduction:

- Stemming, Stopword Removal, Regex Processing (during TF-IDF)
- Tried SVD: 5500 terms represented 95% of the variance, but the models took hours to run on the SVD.

Multi-Label Classifiers

- Some classifiers require the OneVsRestClassifier wrapper, while others are inherently multi-label

Model Evaluation Criteria

- Accuracy, Precision, Recall, F1 Score, % genres correct per movie, % null predictions

Model Comparison

- Models compared using holdout Test set since CV required fit/transforming the TF-IDF at each fold (expensive)

Classification Reports

The classification reports analyze both the overall average performance, as well as the individual label classification performance.

The models over-fit to the training data, even with parameter tuning, so the test evaluation metrics are often disappointing.

The precision is often good, but the recall is bad, meaning that it has a lot of Type II errors (false negatives).

We used the micro average metrics since we have unbalanced classes.

Train Classification Report

	precision	recall	f1-score	support
Action	0.96	0.74	0.84	1518
Adventure	0.98	0.60	0.75	996
Animation	1.00	0.39	0.56	314
Biography	0.99	0.44	0.61	602
Comedy	0.94	0.82	0.88	2953
Crime	0.95	0.70	0.81	1355
Documentary	0.98	0.85	0.91	808
Drama	0.89	0.91	0.90	4549
Family	1.00	0.36	0.53	464
Fantasy	0.99	0.38	0.55	614
History	1.00	0.32	0.49	323
Horror	0.97	0.85	0.90	1459
Music	0.98	0.75	0.85	365
Musical	1.00	0.08	0.15	102
Mystery	0.99	0.34	0.51	789
News	0.00	0.00	0.00	21
Romance	0.95	0.68	0.80	1619
Sci-Fi	0.97	0.63	0.77	669
Sport	0.96	0.76	0.85	260
Thriller	0.92	0.69	0.79	2135
War	0.99	0.64	0.78	229
Western	1.00	0.32	0.49	68
micro avg	0.94	0.72	0.81	22212
macro avg	0.93	0.56	0.67	22212
weighted avg	0.95	0.72	0.80	22212
samples avg	0.92	0.75	0.80	22212

% Accuracy: 43.3% | % Null Predict: 3.85%

Test Classification Report

	precision	recall	f1-score	support
Action	0.77	0.46	0.57	320
Adventure	0.72	0.25	0.37	186
Animation	1.00	0.07	0.12	60
Biography	0.62	0.04	0.08	119
Comedy	0.72	0.54	0.62	540
Crime	0.73	0.36	0.48	299
Documentary	0.92	0.53	0.67	131
Drama	0.71	0.72	0.71	870
Family	0.75	0.07	0.13	85
Fantasy	0.86	0.17	0.28	106
History	0.00	0.00	0.00	64
Horror	0.80	0.59	0.68	227
Music	0.72	0.33	0.46	54
Musical	0.00	0.00	0.00	16
Mystery	0.53	0.05	0.08	174
News	0.00	0.00	0.00	1
Romance	0.65	0.38	0.48	288
Sci-Fi	0.75	0.26	0.39	126
Sport	0.70	0.18	0.29	39
Thriller	0.61	0.34	0.44	443
War	0.92	0.21	0.34	53
Western	0.00	0.00	0.00	15
micro avg	0.72	0.43	0.53	4216
macro avg	0.61	0.25	0.33	4216
weighted avg	0.70	0.43	0.50	4216
samples avg	0.66	0.47	0.51	4216

% Accuracy: 13.4% | % Null Predict: 22.4%

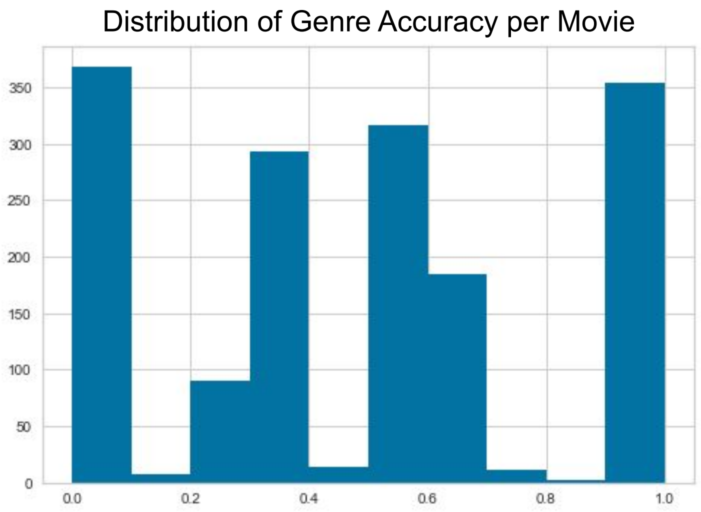
SVC Multi-Label Classification

SVC historically works well for text data. The goal is to find a hyperplane to separate the classes in the data by maximizing the margin between the support vectors, and thus minimizing the hinge loss

SVC requires the OneVsRestClassifier() wrapper to perform multi-label classification

Multiple combinations of C values were tried with the Linear kernel, as well as RBF (Gaussian).

Kernel = Linear, C= 1	Score
Micro Average Precision	0.72
Micro Average Recall	0.43
Micro Average F1-Score:	0.53
% Accuracy (Full Genre List Correct):	13.8%
% One Genre Predicted Correct	77.6%
% Null Predictions	22.4%

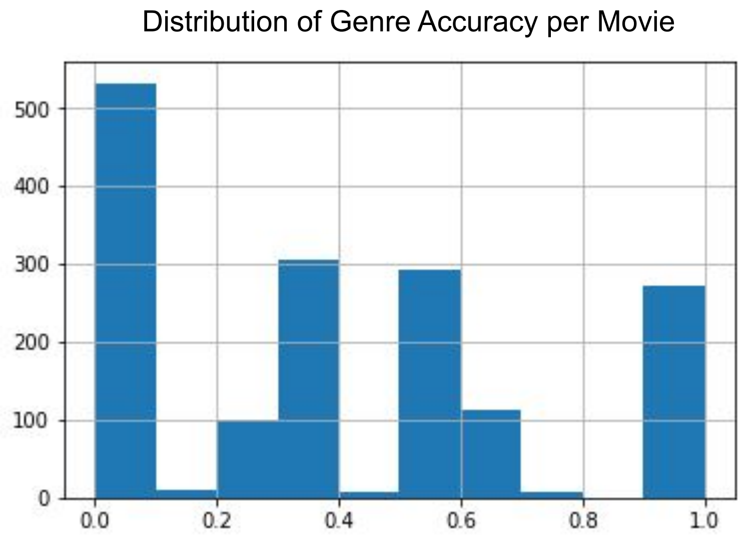


Logistic Regression Multi-Label Classification

Logistic regression also tries to find a plane to separate the data, but uses the probability distribution of the results to form the boundary and tries to minimize the logistic loss.

Logistic regression also requires the OneVsRestClassifier() wrapper to perform multi-label classification

Threshold = 0.5	Score
Micro Average Precision	0.76
Micro Average Recall	0.34
Micro Average F1-Score:	0.47
% Accuracy (Full Genre List Correct):	12.5%
% One Genre Predicted Correct	67.6%
% Null Predictions	32.43%

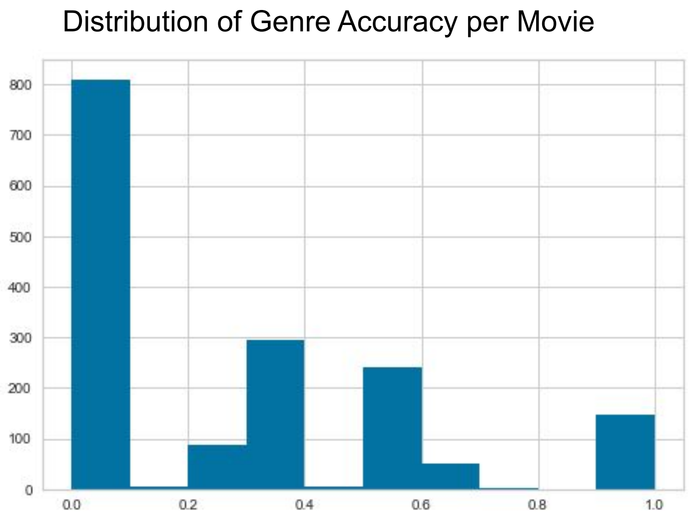


K Nearest Neighbors (KNN) Classification

Tried k=5-25, weights = ‘distance’ & ‘uniform’) and settled on (k=18 & ‘uniform’) providing best accuracy & categorical report scores.

- KNN doesn’t require training before making predictions, so new data could be added seamlessly.
- KNN did pretty well, but was not the best model partly because KNN doesn’t work optimally with large datasets (calculating distances) and categorical values (again calculating distances).

n_neighbors = 18, weights = ‘uniform’	Score
Micro Average Precision	0.72
Micro Average Recall	0.22
Micro Average F1-Score:	0.34
% Accuracy (Full Genre List Correct):	7.6%
% One Genre Predicted Correct	50.7%
% Null Predictions	49.3%

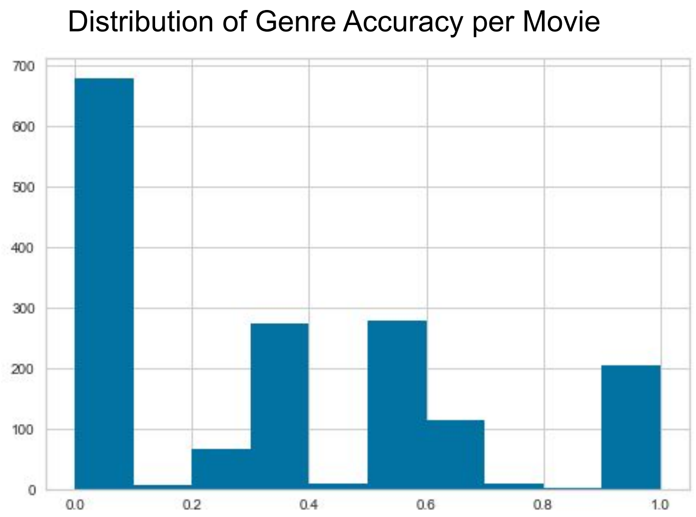


Decision Tree Classification

Tried multiple values for parameters (criterion='entropy'/'gini', max_depth=10-100, min_sample_leaf=2-50k, …) and settled on ('gini', depth=22, leaf=18) using accuracy & categorical report scores as guide, creating 411 nodes.

DT didn't predict as well as KNN.

criterion = 'gini', max_depth = 22, min_sample_leaf = 18	Score
Micro Average Precision	0.57
Micro Average Recall	0.28
Micro Average F1-Score:	0.37
% Accuracy (Full Genre List Correct):	8.4%
% One Genre Predicted Correct	59.6%
% Null Predictions	40.4%

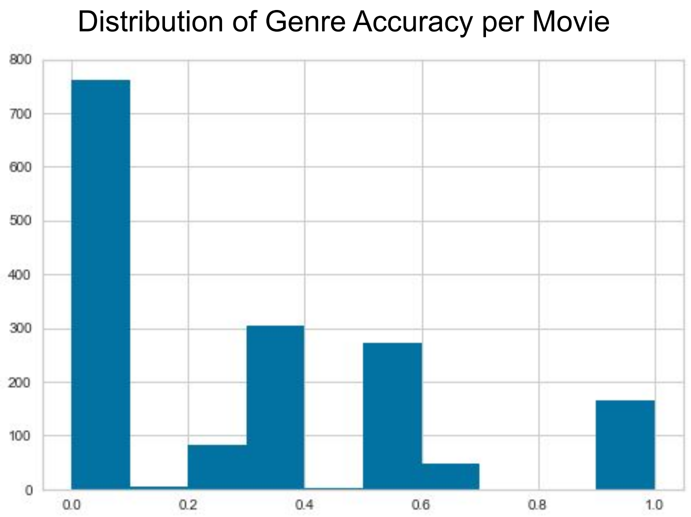


Random Forest Classification (RF)

Tried multiple values for parameters (n_estimators=50-1,000, min_samples=1-4) and settled on (n=125, samples=4) using accuracy & categorical report scores as guide. Let RF select max_features with 13,493 features

RF did much better than DT, and slightly better than KNN.

n_estimators = 125, min_samples_leaf = 4, criterion = 'gini'	Score
Micro Average Precision	0.70
Micro Average Recall	0.18
Micro Average F1-Score:	0.29
% Accuracy (Full Genre List Correct):	6.5%
% One Genre Predicted Correct	45.0%
% Null Predictions	55.0%



Model Predictions

	title	truth	svc_l1_prediction	logreg_prediction	dt_prediction	rf_prediction	knn_prediction
5	The Dancer Upstairs	[Crime, Drama, Thriller]	[Drama, Romance]	[Drama]	[Drama, Romance]	[Drama]	[Drama]
14	Frida	[Biography, Drama, Romance]	[Drama, Romance]	[Drama, Romance]	[Drama]	[Drama]	[Drama]
19	Resident Evil	[Action, Horror, Sci-Fi]	[Action, Horror, Sci-Fi]	[Action, Horror, Sci-Fi]	[Comedy, Horror]	[]	[Sci-Fi]
22	Men in Black II	[Action, Adventure, Comedy, Mystery, Sci-Fi]	[Action, Sci-Fi, Thriller]	[Action]	[Sci-Fi]	[]	[]
26	Star Wars: Episode II - Attack of the Clones	[Action, Adventure, Fantasy, Sci-Fi]	[Action, Sci-Fi]	[Action]	[Drama]	[Drama]	[Sci-Fi]
38	Treasure Planet	[Animation, Adventure, Family]	[Adventure, Sci-Fi]	[]	[Drama]	[]	[]
58	Spider-Man	[Action, Adventure, Sci-Fi]	[Comedy, Drama, Sci-Fi]	[Drama]	[Crime, Drama, Thriller]	[Drama]	[]
59	Naqoyqatsi	[Documentary, Music]	[Documentary]	[Documentary]	[Drama]	[]	[]
77	Clockstoppers	[Action, Adventure, Comedy, Sci-Fi, Thriller]	[Adventure]	[]	[Drama]	[]	[Drama]

Model Selection

The Linear SVC had the highest accuracy and f1 score.

F1 helps balance precision and recall. If this was being used to curate lists of movies to watch, we would want to minimize our Type 1 errors (False Positives), since an improperly classified movie in a list is worse than a movie that doesn't get classified onto a list

Model Metrics	Decision Tree	Random Forest	KNN	Logistic Regression	Linear SVC
Micro Average Precision	0.57	0.70	0.72	0.76	**0.72**
Micro Average Recall	0.28	0.18	0.22	0.34	**0.43**
Micro Average F1-Score:	0.37	0.29	0.34	0.47	**0.53**
% Accuracy (Full Genre List Correct):	8.4%	6.5%	7.6%	12.5%	**13.4%**
% One Genre Predicted Correct	59.6%	45.0%	50.7%	67.6%	**77.6%**
% Null Predictions	40.4%	55.0%	49.3%	32.4%	**22.4%**

Future Work

(With more computing power)

One vs Rest Models

- Use SVD components
- Do a true Grid Search CV with re-fit/transform of TF-IDF for parameter tuning and Nested CV for model selection
- Improve stemming and text cleaning

LDA Model

- Explore more recent techniques such as LDA2Vec, Interactive Topic Modeling
- Try using the terms from the top LDA topics as the vocabulary for the OvR models
- Tune the model parameters (ie, number of topics, learning decay) and do a Grid Search to find the best performance model