Letterboxd Data Analysis

Evaluating my movie-rating behavior by training predictive models on personal data

Nick Wibert

STA4241

Dec. 8, 2021

Introduction

- Letterboxd
  - Movie diary tool / social
  - log films, rate out of 5 stars, write reviews, etc.
- I have been logging films since May 2016
- Can export data as a .csv
  - Very basic info + star-rating
- Goal: apply classification algorithms covered in the course to evaluate their performance and gain insight on my movie-rating behavior

Data collection

- exported data is barebones
- Using “rvest” package, wrote a web-scraping script to pull additional covariates from the site
- scraped all that was available knowing a lot will be dropped later

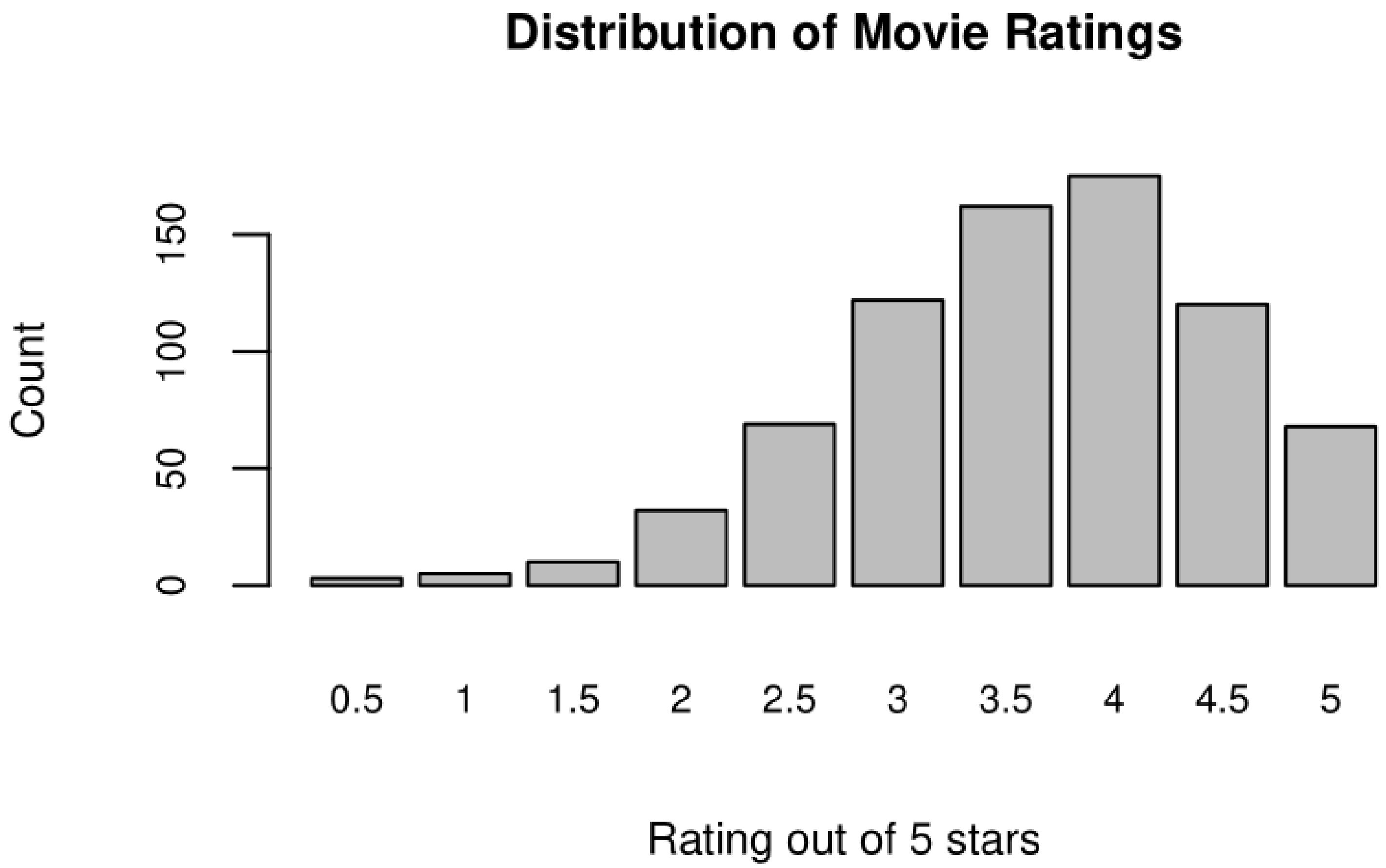
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Name	Year	Rating	Tags	Watched.D	Average.R	Runtime	Genre	Director	Actor	Writer	Editor	Cinemato	Composer	Producer	Studio	Country	Language
753	Training Day	2001	4	streaming	3/25/2021	3.81	122	thriller	Antoine F	Ethan Haw	David Aye	Conrad Bu	Mauro Fio	Mark Man	Bruce Ber	WV Films	Australia	English
754	Lost Highway	1997	4.5	streaming	3/27/2021	3.94	134	drama	David Lyn	Bill Pullm	David Lyn	Mary Swe	Peter Den	Angelo Ba	Tom Stern	CiBy 2000	USA	English
755	Blue Velvet	1986	4	theatre	3/28/2021	4.08	120	thriller	David Lyn	Kyle MacL	David Lyn	Duwayne	Frederick	Angelo Ba	Dino De L	DEG	USA	English

Description of data set

- Response: **Rating**
  - Rating I assigned to the film out of 5 stars
  - Broken up into increments of 0.5 stars
    - Ordinal response with 10 levels (0.5, 1, 1.5, …, 4.5, 5)
- Numerical covariates
  - Release Year, Watched.Date, Average.Rating, Runtime (minutes)
- Categorical covariates
  - “Tags” (format), Genre, Country, Language, Studio, various crew members
- Goal: using these covariates, train models to classify films into one of the 10 possible classes of **Rating**

Exploratory data analysis

- clear left skew
- Imbalanced data set
  - will need to account for this when resampling
  - Stratified sampling
- "Average.Rating" highly correlated with response
  - r ≈ 0.8
- Expect this to be the most significant covariate in predicting ratings



Exploratory data analysis (cont.)

- Many web-scraped covariates were excluded early on
  - Extremely small coefficients in preliminary models with low significance, linear dependencies
  - Heavy computationally because of many possible values (30+ categories; for crew members, several hundred)
- Variable selection procedures
  - Best subset selection and backward stepwise regression
  - Both procedures wittled down to “Average.Rating”, “Watched.Date”, and “as.factor(Tags)”
  - Best subset also included many levels of “Genre” in the model which maximized the adjusted R-squared

Implementation of classification algorithms

- in class, we dealt a lot with binary classification, and several methods had natural extensions to n-level classifications
- I performed simulation studies using four of these methods which had natural extensions
  - LDA, KNN, SVM with polynomial kernel, and SVM with radial kernel

Implementation of classification algorithms (cont.)

- What about logistic regression?
    - We need a modified logit that can handle ordinal response
    - We discussed baseline-category logit model in class, though this is not often used for classification + I had trouble implementing
- Another option is the *cumulative logit model with proportional odds* (Categorical Data Analysis)
    - Designed to handle ordinal response
- Makes a very strong model assumption of proportional odds
    - Meaning, the effect of each predictor is the same for each logit
- Performed a LR test to test this assumption and ended up rejecting it
    - Not surprising given the imbalance in the data set
- I still implemented the proportional odds model out of curiosity to compare with other methods
    - Keep in mind that this major assumption is violated so we can’t really use it here

Implementation of classification algorithms (cont.)

- Three simulation studies
  - One using Average.Rating, Watched.Date, as.factor(Tags), as.factor(Genre)
    - Chosen by best subset
  - One using the same set, minus Genre
  - One using only Average.Rating
- Loop with 100 iterations
  - Stratified resampling on each loop with 80%-20% training-test data split
- Train each model on training data, run predictions on test data, and store classification error rate
- Average error rates over all 100 iterations

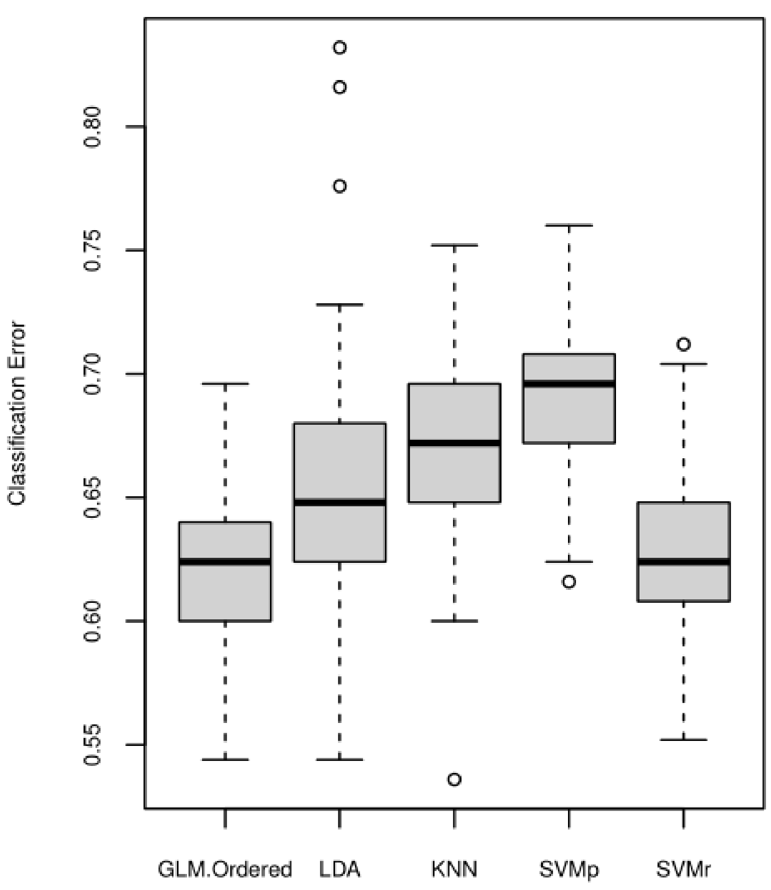
Results

- Across all three simulations and all models, error rates ranged from 60% to 69% on average
- Simulation including Genre had generally higher error rates
- Average error rates for models with *only* Average.Rating were lower than or very similar to error rates for more complex models across the board
  - Average.Rating weighed very heavily across models, with coefficients in the 4-6 range while absolute values of coefficients for all other predictors was generally less than 1
- Models performed relatively similarly across all three simulations

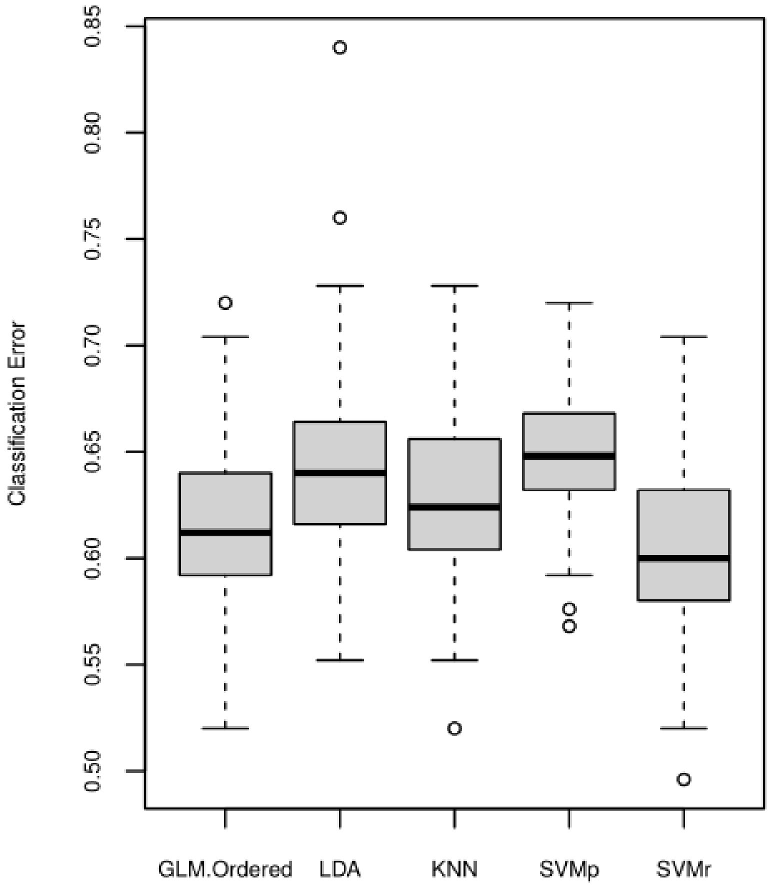
Results (cont.)

	GLM.Ordered	LDA	KNN	SVMp	SVMr
w/ Genre	0.6208800	0.6525600	0.6671200	0.69088	0.6288000
wo/ Genre	0.6138400	0.6432800	0.6284800	0.64816	0.6038400
Average.Rating only	0.6262667	0.6505333	0.6221333	0.64120	0.6498667

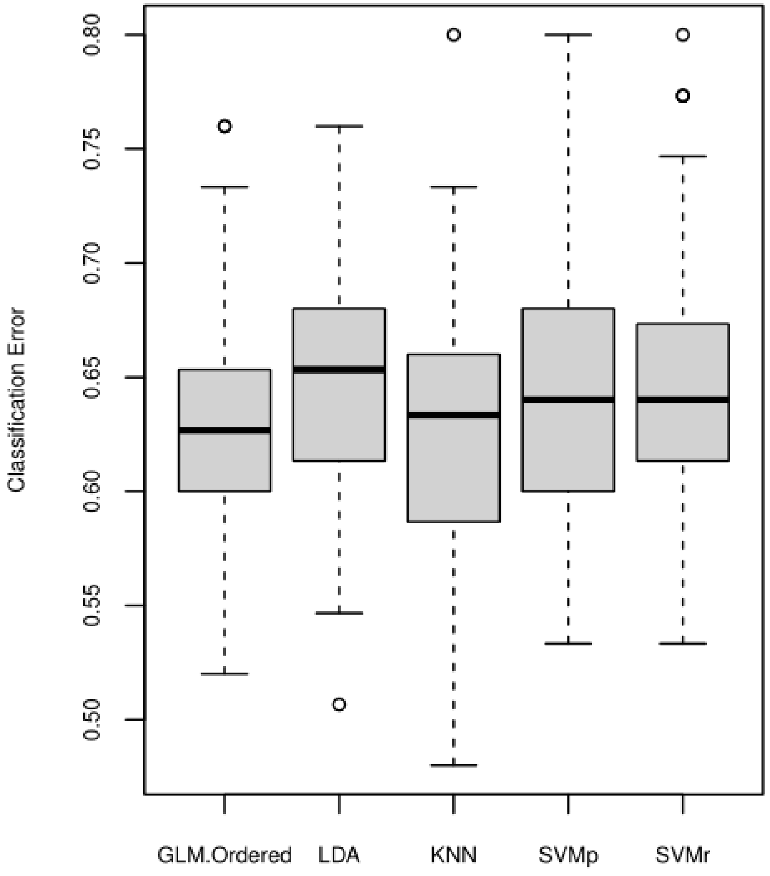
Classification Error Rates for Models with Average.Rating, Watched.Date, as.factor(Tags), as.factor(Genre)



Classification Error Rates for Models with Average.Rating, Watched.Date, as.factor(Tags)



Classification Error Rates for Models with Average.Rating Only



Discussion

- Predicting human behavior is difficult, especially something as subjective as rating films
  - While models did not perform spectacularly, this is about what we would expect given the nature of the data set
- We need a benchmark to evaluate the performance of the models
  - Simplest benchmark is random chance: 10% success rate for 10-level classif.
  - Zero rule: choose majority class (4 stars, ~ 23% of final dataset)
- Average success rate was 30-40% on average
  - While not great, still better than these benchmarks
  - There's *some* level of predictive power, mostly due to Average.Rating

Discussion (cont.)

- Average.Rating is the most significant covariate considered by far
  - Possible explanations
    - Average rating is one of the last things you see when you pull up a film to rate it
    - Easy to be slightly / subconsciously influenced by what general audiences think
    - Or, my taste just tends to align with the population of Letterboxd users
    - Even when classification failed, it was very often in one of the adjacent classes
      - Considering Rating as a continuous response and calculating MSE, consistently got MSE ~20%
- Other main covariates considered, Watched.Date and Tags, cannot be extrapolated to films I have not seen yet
  - I do not know when/how I will watch them

Discussion (cont.)

- Weaknesses of data set
  - The imbalance had clear effects on predictive power
    - Models tended to fail more often when Average.Rating was in the lower half of ratings
    - Original data set has far fewer observations in those classes, so it has less to train on
  - Since exporting the original data set for this project, I have seen 6 movies
    - True Ratings: 3.5, 5.0, 5.0, 2.0, 3.0, 4.0
    - None of the models got more than 2 correct
      - The 2 that were correct were always one of the three higher ratings (4 or 5 stars)
      - Shows that models are weakest when Average.Rating OR my personal rating are lower, as the models are trained primarily on data in the 3.5 to 5 star range
      - Stratified resampling helps, but cannot make up for lack of data