**Project Proposal**

Team Members: Dan Houli, Andrew Charland, Jay Jayakumar, Jason Popowski

The problem or need: Spend a lot of time looking for the right movie to watch? Sure, Netflix and Amazon make recommendations based on your viewing history or rating history. But what if you could have more control over the results by rating genres or actors? Then have an algorithm more accurately return, with the highest probability, the movies that you may want to see.

[original]:

But, what if you could interact with the data and rate genres or actors and then let an algorithm return the movies with the highest probability that you may like?

Customer/value proposition: Content streaming service looking to differentiate on product capabilities.

Users: Content streaming service viewers

Topic type: Comparative study of existing algorithms applied in an interesting way.

The Data Set: IMDB 5000 Movie Dataset

Freely available at <https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset>.

Description: 5000+ movie data scraped from IMDB website

Attributes: The 28 variables include

"movie\_title" "color" "num\_critic\_for\_reviews" "movie\_facebook\_likes" "duration" "director\_name" "director\_facebook\_likes" "actor\_3\_name" "actor\_3\_facebook\_likes" "actor\_2\_name" "actor\_2\_facebook\_likes" "actor\_1\_name" "actor\_1\_facebook\_likes" "gross" "genres" "num\_voted\_users" "cast\_total\_facebook\_likes" "facenumber\_in\_poster" "plot\_keywords" "movie\_imdb\_link" "num\_user\_for\_reviews" "language" "country" "content\_rating" "budget" "title\_year" "imdb\_score" "aspect\_ratio"

Hypothesis:

1) We can dynamically change the observed evidence of a few select variables with prior probabilities to produce an effective “movie like” probability.

2) We can dynamically build these probabilities into Bayesian networks through visual graphs.

Technical Approach:

1) Math / Equations

We’ll predict whether a movie viewer will like a movie, based on his/her choices/rating for genre, actor, and other variables. Prior knowledge will be used from dataset including general population Facebook likes – (IMDB score converted to a percentage (i.e. imdb score \* 0.01)), overall IMDB movie rating (again converted to a percentage) and other variables. For new observations in the Bayesian process, viewers will select multiple variables (genre, director and/or actor), rate each one on scale of 1 to 10. The scale will be assumed to equal probability of liking the variable (i.e., director or actor). For example, if a viewer rates “Tom Hanks” as an 7, the P(Actor) = .7. To determine P(Actor | Like) or P(Director | Like), we’ll apply a “to be determined” normalization and aggregation of all ratings and input. Next, the we’ll apply the Bayesian approach of probability and inference to obtain a probability of liking the film (posterior probability). For example, P(Like | Actor) or P(Like | Director). Finally, we’ll have to conduct a limited test of the algorithm to evaluate its effectiveness. The probabilities learned from the data will also be applied to constructing Bayesian networks.

2) Technology: The principal analysis will be conducted using R programming language. Key packages include:

1. Shiny: provides easy to use, interactive web applications from R. A user with zero of knowledge of R may intuitively interact with the data and algorithms. Additionally, the applications present a neat and visual stimulating graph, table, or chart.
2. LaplacesDemon: Package to conduct Bayesian inference.
3. bnlearn: Package to conduct Bayesian network
4. igraph: Package to conduct network analysis and visualization

Project execution risks and risk mitigation strategies:

1) Adjusting Facebooks likes to probabilities: The Facebook “likes” contain a high degree of variance across the movies. Adjusting the “likes” counts judiciously is important to avoid skewing the results proportionately.

2) Building shiny web app: Some learning of the shiny software to learn its capabilities and flexibility manage the interactive data drill downs / selections, interoperability with other packages, and ability to produce some of the graphs is required by the team. Some of the analysis may need to be presented separately from the shiny-produced web application.

Goals and Deliverables:

1) Prove both hypothesis. Specifically, provide alternate method for movie viewers to select movies possessing a high probability of being liked. Further, the offering alternative selections of movies may be aided through network analysis and visualization.

2) Tool: Build shiny web app and deploy to shinyapps.io, and publish shiny web app on GitHub.