Project Task 4: Analysis

Jack Diaz
111499298

Data Set

- The initial data set was composed of movie ratings on a scale from 1 to 5
  - This included data about the users who made the ratings and time of rating
  - The movie data included the title, release year, and genres
  - The user data included gender, age, occupation, and zip-code
- The data set was split up into a training set and a testing set
- I randomly partitioned the training set in order to run a k-fold cross validation on my models to assess how well my models would generalize to an independent data set
- I also measured F1 scores for decision trees and MCC for association rule mining in order to compare to the grading thresholds

Decision Trees

- Our first task was to build a decision tree based on the training data that would:
  - Reach an F1 score of .40 for classifying movies as 5 stars
  - Reach an F1 score of .65 for classifying movies as 4 or 5 stars.
- I ran into a couple of issues with this:
  - First it would take a long time to build
  - Second my tree kept growing very large and reaching the maximum recursive depth of my system
    - The way I fixed this was by increasing the recursion limit
    - It still took a while to build and gave me low F1 scores (~.25 for 5 star and ~.3 for 4 and 5 stars)

Decision Trees

- To speed up building I limited the maximum depth of the tree to 4 levels
  - This increased my F1 scores for 5 star ratings to ~.41 and my F1 scores for 4 and 5 star ratings to ~.72
  - I tested with many different level limits and it seemed that 4 levels would get me the best scores
    - <4 max levels the scores were ok, but not above the threshold
    - >4 max levels the scores dropped very quickly

Decision Trees

- So why did limiting the height of the tree help increase scores?
  - I noticed a few independent variables had a large effect on the rating of a movie
  - That means any independent variable that would be used as a split condition after the 4^{th} level was likely having little impact on the movie's score and contributed to over fitting
  - Because building time was faster I could see the results quicker and therefore make any necessary changes in a reasonable amount of time

Decision Trees

- The independent variables I tried were the following:
  - The gender, age, occupation, and 1 through 5 digits of the zip code of the user
  - The ID, year, decade, and genres of the movie
  - The year and month the rating was made
- Upon testing with these I quickly found that some were unnecessary
  - I removed them because they were slowing down the building of the tree and contributing to over fitting

Decision Trees

- The independent variables that I used to build the trees were the following:
  - The gender, age, occupation, and the first digit of the zip code of the user
  - The release year and genres of the movie
  - The year the rating was made
- The most salient independent variables were the following:
  - Genres includes Drama
  - Genres includes Film-Noir
  - Year is 1977
  - Genres includes War
- I figured this out because they were the closest to the top of the tree, meaning they split the data the best and were most relevant to determining a movie's rating

Decision Trees

- The threshold for F1 scores was .65 for 4 and 5 star ratings and I got .72 which I think is excellent
- This is good for an algorithm that created the tree very quickly
- This is also good because I had such a small tree, so I had a simple way to classify data accurately

Association Rules

- For the association rules mining task I decided to use the Apriori Algorithm
- I used this algorithm to find rules that say if a set of independent variables have values $A_1 \ldots A_n$ then the dependent variable will be B
- Our rules needed to have class-wise support of 0.01 and confidence of 0.65
- I ended up finding a few rules for the positive case, classifying a movie as having 4 or 5 stars, and no rules for the negative case

Association Rules

- From the first task I found that a lot of the data I had about the users and the movies were irrelevant so I built my association rules using the following independent variables:
  - Gender, age, occupation, and the first digit of the zip code of the users
  - Decade the movies came out, and genres of the movies
  - The year the rating was made

Association Rules

- These independent variables seemed to work rather quickly and well and my algorithm provided me with the following rules:
  - 'year'='194' -> 1 Confidence: 0.753356373787
  - 'year'='195' -> 1 Confidence: 0.72119140625
  - 'year'='196' -> 1 Confidence: 0.701339128392
  - 'year'='197' -> 1 Confidence: 0.664988669192
  - 'genre'='Film-Noir' -> 1 Confidence: 0.765306924467
  - 'genre'='War' -> 1 Confidence: 0.693752347565

Association Rules

- The threshold for MCC scores was 0.1 and I got around .13
- This is good for an algorithm that mined rules very quickly
- This is also good because I had so few rules, so I had a simple way to classify data accurately

Support Vector Machine (SVM)

- For this part of the project we needed to use an SVM to classify a subset of the data.
  - That subset was only movies that came out in or after 2000
- We had to train our SVM to get MCC scores above .17
- I used libSVM in Weka
  - I increased the cache size to around 6gb
  - Raising the cost to 10 increased my MCC score

SVM

- From the first two tasks I found that a lot of the data I had about the users and the movies were irrelevant so I trained my SVM using the following independent variables:
  - Gender, age, occupation, and the first digit of the zip code of the users
  - Genres of the movies

SVM

- Because of the nature of SVMs it's difficult to determine which variables had the most influence
- There is no intuition about the data that you can draw from an SVM
- This makes it a bad technique to use if you would like to describe how you got your results to someone with little mathematical background

SVM

- That being said, I know that the independent variables I chose were good because they yielded me a high MCC score.
- Our required score was .17 and I got around .22
- This is a good score for such a quick technique
- Unfortunately this technique does not count as simple to implement, but it is rather simple to use

So what makes a movie good or bad?

- The genres Film-Noir and War are good indicators of a good movie
  - They appeared as split conditions high in the decision tree
  - They were part of the positive rules in the association rule mining algorithm
  - They contributed to the SVM
- The Drama genre is a good decider
  - It was the split condition of the root of the decision tree
  - It contributed to the SVM
- The year the movie came out is a good decider
  - 1977 was a split condition high in the decision tree
  - The decades ’40 ’50 ’60 and ’70 were all in the association rules