Data Science: Capstone - MovieLens Report

Edin Ceman

31st January 2021

Introduction

(Section that describes the dataset and summarizes the goal of the project and key steps that were performed.)

The MovieLens Collection

The MovieLens dataset is a publicly available source of information containing user ratings for movies over a defined period of time. This data was collected and made available by GroupLens Research on the web site (http://movielens.org). There are different kinds of sizes of the MovieLens data. While there is also a full-size version of the data set available, in this course we use the 10m version to facilitate the modeling process while also reducing the necessary computing power.

Dataset

The 10m dataset from MovieLens is a smaller version of the full-size dataset and contains 10000054 ratings and 95580 tags applied to 10681 movies by 71567 users of the online movie recommender service MovieLens. From the dataset, the following fields are extracted in the Data Science: Capstone course:

- **userId** : This is an integer representing a unique user ID. Movielens users were selected at random for inclusion. Their ids have been anonymized.
- **movieId** : This is a numeric representing a unique movie ID.
- **rating** : This is a numeric representating the user rating for a movie. It can display the following discrete values (0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0).
- **timestamp** : This is an integer representing the exact time when the user rating for a certain movie was made.
- **title** : This is a character value displaying the movie's title.
- **genres** : This is a character representing the genre of the movie.

Goal

The goal of the Capstone Project at hand is to build a movie recommendation system based on the MovieLens data set. Hence, the aim of the system of to predict a rating for a certain user and a certain movie at a certain point in time based on his previous ratings and and behavior.

Key Steps

To build the model, the following key steps are conducted:

- Initially, the 10 MovieLens data is downloaded and the above described fields are extraced and put into tidy format. The relevant code has been provided in the edx Course "Data Science: Capstone".
- This data set is then split into a dataset called "edx" and a data set called "validation". The first one contains 90% of the 10m MovieLens data set and shall be used to construct the movie recommendation system. The latter data set contains 10% of the 10m data set and represents the final holdout data set that is used to evaluation the constructed recommendation system. It is not used during the model creation process.
- The edx dataset is then enriched with additional information (date is a feature constructed from the timestamp, and we filter to include only users that have at least 50 ratings). The edx data set is then again split into a train (90%) and test set (10%)
- In the next step, the model is build on the basis of the train set. The model captures certain effects (e.g. user, movie, time and genre effect). Furthermore regularization is used to include a penalty for large deviations.
- The optimal lambda or penalty paramter is found.
- The model is evaluated using the validation set with the optimal parameter.
- Since previous data filtering produces NAs in the validation set, those are replaced by averages.
- Finally, the RMSE is computed for our model on with the results from the validation set.

Methods & Analysi

That explains the process and techniques used, including data cleaning, data exploration and visualization, insights gained, and your modeling approach.

Process

We use the train set to construct our model. First, the average movie rating is calculated. Then, we want to capture certain effect as explained in the Data Science: Machine Learning course. The model contains the following effects:

- Movie Effect: This parameter captures the effect a certain movie has on the rating while adjusting for the overall average rating mu.
- User Effect: This parameter caputres the effect a certain user has on the rating while adjusting for the overall average and the movie effecg.
- Time Effect: This parameter captures the effect the date (time) when a rating was made has on the rating while adjusting for the overall average mu and the movie and user effect.
- Genre Effect: This parameter captures the effect the genre of the movie has on the rating while adjusting for the movie, user, and time effect.

For each of the effect, the concept of regularization is used. This concept includes a penalty parameter lambda and each effect is adjusted to penalize large deviations such that we achieve regularization.

To find the optimal lambda, the modeling process is repeated 100 times. From own experimenting and to avoid you having to let your computer run for several hours, I have narrowed the lambda range to 4.9 to 5.15 (as opposed to 1 to 10 in 0.01 steps). Of course, If you want to verify the optimal lambda, you can run the code with the full lambda range. But be warned that it will take a very long time.

Since our applied data restrictions, especially the restriction to users who have at least 50 ratings, some NAs are produced. Those NAs are replaced by the overall average mu to fill the blanks.

Results section

Present the modeling results and discusses the model performance. To evaluate the model results, we use the root mean squared error (RMSE). The RMSE is a measure that… When evaluating our model against the validation set using the RMSE, we achieve a value of 0.85.

Conclusion

Section that gives a brief summary of the report, its limitations and future work. The report at hand has given you the key insights of my Capstone Project. Initially, the MovieLens Project and data set has been explained. Following that, the key steps have been described. Afterwards, the modeling process as well as the approach for construcing the model has been explained and reasoning has been provided. Finally, the results have been evaluated and presented. The limitations of this project are that we have only used a restricted dataset (10m) instead of the full-scope dataset. Also, not all available fields and information of the MovieLens data set have been used such that certain effects could not be evaluated. Furthermore, the limitations of the computing power of a regular notebook or desktop pc did not allow for the application of more advanced modeling techniques, e.g. using ensemble methods with Random Forests, KNN, GamLoess and many more.

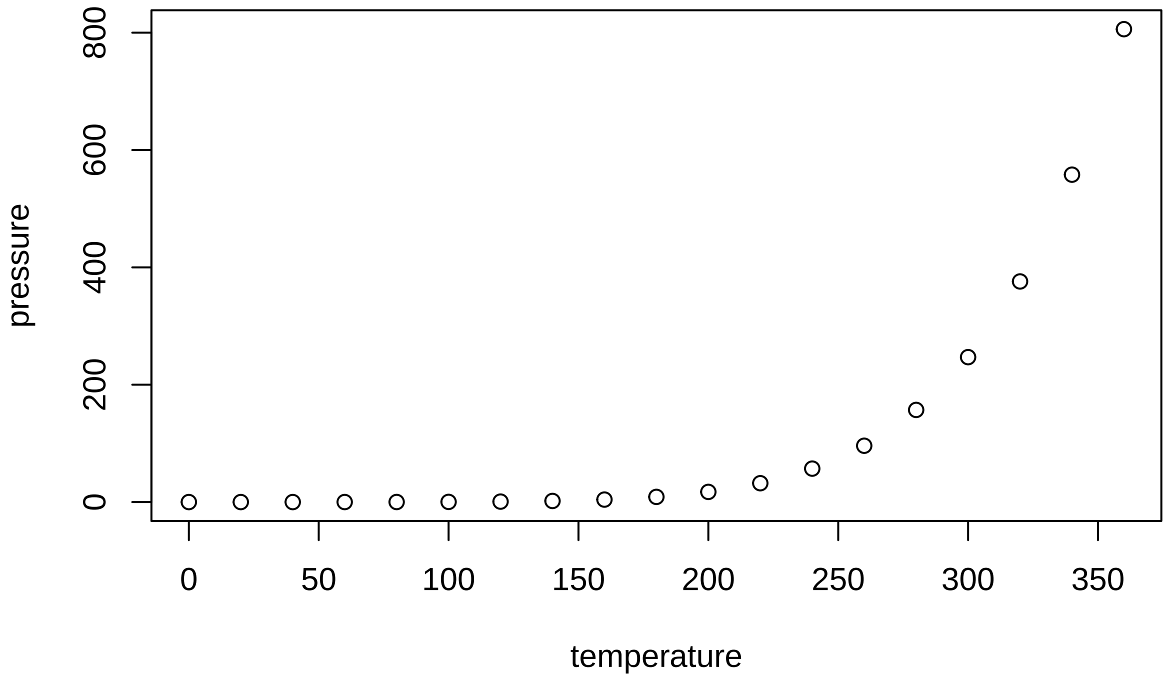
0 points: The report is either not uploaded or contains very minimal information AND/OR the report appears to violate the edX Honor Code. 10 points: Multiple required sections of the report are missing. 15 points: The methods/analysis or the results section of the report is missing or missing significant supporting details. Other sections of the report are present. 20 points: The introduction/overview or the conclusion section of the report is missing, not well-presented or not consistent with the content. 20 points: The report includes all required sections, but the report is significantly difficult to follow or missing supporting detail in multiple sections. 25 points: The report includes all required sections, but the report is difficult to follow or missing supporting detail in one section. 30 points: The report includes all required sections and is well-drafted and easy to follow, but with minor flaws in multiple sections. 35 points: The report includes all required sections and is easy to follow, but with minor flaws in one section. 40 points: The report includes all required sections, is easy to follow with good supporting detail throughout, and is insightful and innovative.

```r
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.