**Owen Galvin**

**May 11, 2016**

## Topic: Machine Learning with H2O.ai

## Concentration: Simple and Multiple Linear Regression

**Problem/Purpose:** Illustrate use of both simple and multiple linear regression in H2O.ai using movie data set. For simple, the goal will be to predict professional movie critics rating of a film by using an explanatory variable feature-engineered to encompass “star power” of the actors in the films, along with the general audience rating of given movie. Multiple Linear regression will attempt to predict same value but using a larger number of primarily categorical values. A preliminary assessment of the technology will also be performed using the Small Car dataset used in class.

**Big Data Set:**

**Data files:** movies.dat, movie\_countries.dat, movie\_directors.dat, movie\_genres.dat (hetrec2011-movielens-2k-v2.zip)

**Description of data:** Links the movies of MovieLens dataset with their corresponding web pages at Internet Movie Database (IMDb) and Rotten Tomatoes movie review systems, includes data on actors, country, genre tags, ratings from both critics and audience members.

**URL To get data:** original .zip @ on <http://grouplens.org/datasets/hetrec-2011/>. Data joined through common movie\_id values via SQL Server + feature engineering to create new values. Full massaged dataset available at [FP\_movies.psv](http://oweng.net/csci-e63/FP_movies.psv).

**Size:** Orig zip: 18MB, used .dat files: ~12MB uncompressed, final input data file: 1MB uncompressed.

**Format of data file:** csv orig, psv final export file for ingestion by H2O

**Hardware:** Windows 7 running VMware with Linux CentOS 6.

**Software:** (grayed lines were used only in data prep**)**

|  |  |
| --- | --- |
| **Technology/tools** | **Description** |
| VMWare Workstation | For hosting Linux CentOS system |
| Excel, UltraEdit | Data munging, editing |
| Microsoft SQL Server | Joining, querying, calculating and hosting data |
| Python via Anaconda2 | Programming, usage of H2O library |
| IPython/Jupyter notebook | Running and output of python/H2O |
| H2O & Flow | Python library + web notebook like UI interface (provided by H2O install) |

**Summary:**

Fast (so they say, I believe them) and well documented open source tool for performing machine learning and related Big Data analysis tasks. R, Python, Scala interfaces; my impression is that R is most fully supported/documented, followed closely by Python. Didn’t find too much Scala documentation, though I wasn’t looking for any.

**Overview of steps:**

1) cover installation and setup in a Python 2.7 environment

2) using Jupyter notebook, run through a tutorial using small car data set (100 rows) from Assignment 11

- loading of data, splitting into test/train

- creating simple linear regression model and training it on displacement in order to predict horsepower

- creating multiple regression model, using same features as from Assignment 11 in order to predict horsepower

- accuracy measures, csv output, viz of simple model results via D3, compare vs. Spark/MLlib predictions

3) similar simple and multiple regression examples for the movie dataset

- for simple, predict critic’s rating with the help of feature engineering (movie star power + audience rating)

- visualize results of above in D3

- carry out similar predictions via model using multiple feature variables

**Pro/Con:**

Silicon Valley startup but supported by some big names in the field (Rob Tibshirani & Trevor Hastie from Stanford among others), so likely to continue growing. Codebase is fast moving, which means many continual improvements but also conflicting documentation and videos on the web, i.e. old vs. new. Internal help function very comprehensive. Syntactically friendly, within Python both pandas and numpy supported/expected. Similarly named syntax between R and Python, though overall more oriented toward R in nomenclature. One “pro” is lots of knobs to play with, which is also a con. One other con is that I keep typing H20 when I mean to type H2O.

**YouTube URLs here: long:** [**https://youtu.be/OxjCc99Jer0**](https://youtu.be/OxjCc99Jer0) **short:** [**https://youtu.be/uAxmUhqxOFY**](https://youtu.be/uAxmUhqxOFY)