HarvardX: PH125.9x Final Project

By Philip J Brown, pjbMit@pjb3.com

6/16/2019

HarvardX Data Science Capstone Class PH125.9x (2T2018)

- Student: Philip Brown
- email: Phil@pjb3.com
- github: https://github.com/pjbMit

RealML - *A Real Estate Machine Learning Project*

This is RealML, a real estate machine learning project and report created by Philip J Brown (pjbMit@pjb3.com) as the final capstone project for the Data Science Certificate program offered by HarvardX: PH125.9x from edx.org.

Part 1) EXECUTIVE SUMMARY

The goal of this project is to utilize data analysis and modelling skills to a create machine learning engine and this report as the final exercise in completing the 9 course Data Science Certificate program offered by HarvardX through edx.org.

For my project I chose to use machine learning techniques to build a *RealML,a real estate sales price prediction engine.* More specifically, I wanted to answer this question:

> **Can I reasonably predict the resale price of residential condominum and single family real estate within a five mile radius of *Fairlington Villages (link)*, the condominum development in Arlington Virginia that I call home?**

Through this project, I am able to demonstrate examples of data identificaton and acquisition, data wrangling and cleansing, data analysis, modeling and machine learning techniques, data presentation and data visualization and report generation and presentation. The project was built by acquiring and analyzing more than 20,000 reports of current real estate sales within the stated five mile radius for residential properties that sold for at least $5,000 but less than $1,000,000.

After some research I was able to locate and curate live data for this project, so the basis for this report is real and impactful – at least it is to me as home owner in this area. * :-)

The results of this analysis were very encouraging, and are included in the **results** section and the **conclusion** section, which are the last two sections in this report.

The mission was to locate, curate, wrangle and cleanse real data, and use it build a prediction engine that tries to predict sales prices so as to optimize the model for a low Root Mean Square Error (RMSE), defined as

$$RMSE = \sqrt{\Sigma_{i=1}^{n} \frac{(actual_i - predicted_i)^2}{N}}$$

This project is intended to highlight some of the skills acquired throughout the courses in this program. All programming was done in R Code using RStudio on MacBook Pro. After acquiring and processing the data, the real work began!

In addition to this **executive summary**, this report also includes a **methods and analysis section**, a **results section** and a **conclusion section**.

Key files for this project have been uploaded and stored on my git hub page at github.com/pjbMit/real_estate_project. The three main files for this project are listed below, and can be viewed on git hub – The file names are also links:

- real_ml_script.R *(link)*

- real_ml_report.Rmd *(link)*

- real_ml_report.pdf *(link)*

Additionally, a gzip'd version of the data file is on github at:

- realml_data_file.github.json.gz *(link)*

Part 2) METHODS AND ANALYSIS

The project was created in the RStudio environment using Rstudio Version 1.1.442 on a Macintosh; Intel Mac OS X 10_14_5

R version 3.5.1 (2018-07-02)
nickname Feather Spray

See the README.Rmd or README.html files for more information on the environment and setup. For questions, email me: pjbMit@pjb3.com , and I'll gladly respond promptly.

All code was written in R and executed in RStudio.

Here are the methods and techniques used.

TODO * Data was . . . * After these models were evaluated, we looked at variability, and attempted to add genre to models using several standard models available through the **caret package** and applied techniques such as cross-validation. While we examined these models, and made multiple attempts to improve the results, none of the techniques tried improved upon the best results that were previously used.

– General approach:

For many approaches, I first tried working on a very small data set, just to get the code working, then I re-ran the on a medium sized data set, and then when I was satisfied, then I processed the full training set.

Similarly, initially I did NOT do full cross-validation, but once the model was built and the code was working, I enabled cross validation and other ML techniques.

Additionally, being sensitive to computation times, I wrote code and used global variables to enable saving daa and objects containing intermediate results as files on the local file system. By changing the values of these logial variable from TRUE to FALSE, or vice-versa, I was able to re-run code without having to repeat some of the more lengthy processing or repeatedly downloading and cleansing the same data.

–set up

Set up libraries and enable multi-core processing for some of the operations used by the caret package. Because I have an 8 core processor, for calcuations that can utilize the parallel processing features, this script runs ***substantially** faster.

First set up a few logical variables in the global environment, then install packages and load libraries if and as needed.

Part 3) RESULTS

Here's a link back to the Summary section of this script.

The model results were promising, as can be seen by the output from rmse_results.

TODO

Part 4) CONCLUSION

I discovered a library and options to set to enable multi-core parallel processing for some of the algorithms in the **caret** package, and this technique helped tremendously, as I was able to span 8 R-sessions that ran in parallel to process some of the algorithms.

Ultimately, the best results that I obtained were a **RMSE of TODO** which was obtained ffrom the final model. This was deemed satisfactory based on goals and scope of this project. Of course, if you plan to move nearby, please do your own due dilligence before purchasing a home – while I wanted to choose an impactful and relevant project, this project was created primarily for didactic purposes.

(See the output below which shows the best results obtained.)

```
# The estimates that minimize this can be found similarly to what we did above.
# Here we use cross-validation to pick a  lambda
if(runningInScript){
}
```

```
## [1] "Thanks for checking this out!  pjbMit@pjb3.com  :-)"
```