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

All code was written in R and executed in RStudio.

Here are the methods and techniques used.

Data was downloaded from AttomData.com, a commercial data provider, using their RESTful API and an apikey that is needed in order to get data. Sales data was queried from their API, and results were downloaded 10,000 rows at a time.

The data was then filtered to remove property types that aren't residential condos or homes.

Part 3) RESULTS

After downloaing the raw data, we had sales information covering the following sales date range:

```
##   newest_sale oldest_sale
## 1    2019-5-6   2012-1-10
```

We found additional filter criteria to help us cleanse the data. For example, shown below is the data after filtering to show just the property types and subtypes of interest, which had the effect of removing commercial sales, industrial sales, and other data that is out of scope for this project.

```
##              propsubtype HOUSE RESIDENTIAL
## proptype
## CONDOMINIUM                864       11511
## SFR                      10516        5134
```

```
##   num_sales
## 1     28025
```

After examining the data, we saw that a little data-cleansing house keeping was in order. We found that about 10% of the data had zero listed for bedrooms, yet those units had about 1400 sqft on average (mean), thus the zero bedrooms was clearly an error. We removed these rows along with two unneeded colums.

```
## # A tibble: 1 x 2
##   transtype   num
##   <chr>     <int>
## 1 Resale    28025
```

```
##    num mean(sqft)
## 1 3923   1455.371
```

After cleansing the data, here's a grouped summary showing the data by year and property type:

```
##              year_sold 2012 2013 2014 2015 2016 2017 2018 2019
## proptype
## CONDOMINIUM              718  976  989 1797 1741 1412 1426  204
## SFR                     1076 1259 1118 2012 1865 1268 1188  235
```

```
## [1] 19284
```

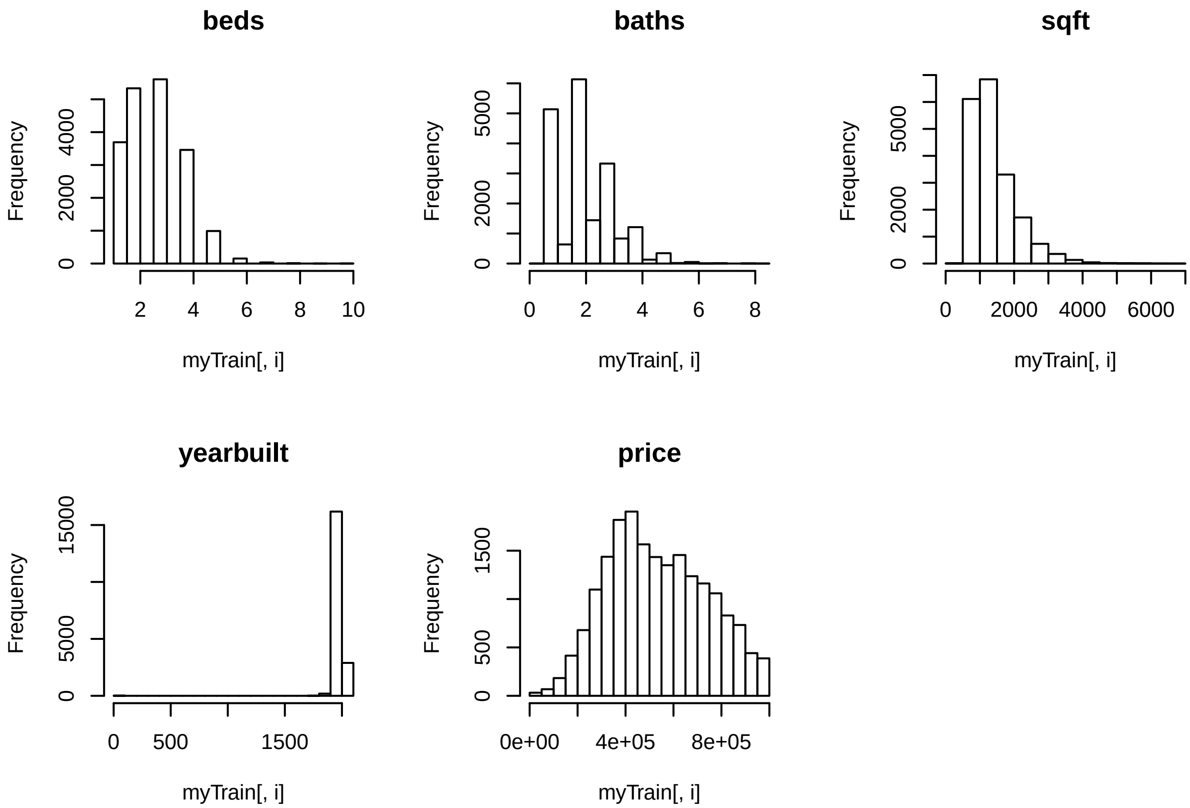
Now it's time to look at some the individual data attributes.

```
##    beds baths sqft yearbuilt    proptype                             addr
## 1     3   2.0 1152      1947         SFR                  3831 1ST ST SE
## 2     2   3.0 1174      1900 CONDOMINIUM     4623 MACARTHUR BLVD NW UNIT B
## 4     2   3.0 1174      1900 CONDOMINIUM 4617 1/2 MACARTHUR BLVD NW UNIT B
## 5     1   1.0  670      1962 CONDOMINIUM           922 24TH ST NW APT 505
## 6     1   2.0 1147      1981 CONDOMINIUM    1080 WISCONSIN AVE NW APT 2013
## 7     5   4.5 5259      1780         SFR                     224 S LEE ST
## 8     2   3.0 1752      1958         SFR                6422 WILLOWOOD LN
## 10    4   2.0 1440      1950         SFR                 6402 CAVALIER DR
## 11    3   1.0 1081      1955         SFR                    3603 KEOTA ST
## 12    3   2.5 2012      1954         SFR                1513 CRESTWOOD DR
##          city state   zip       lat        lon  saledate  price
## 1  Washington    DC 20032 38.833980 -77.006050  2019-5-6 225000
## 2  Washington    DC 20007 38.911207 -77.089150 2019-4-30 665000
## 4  Washington    DC 20007 38.911050 -77.089072 2019-4-30 672000
## 5  Washington    DC 20037 38.901422 -77.051584 2019-4-29 340500
## 6  Washington    DC 20007 38.904825 -77.062959 2019-4-29 570000
## 7  Alexandria    VA 22314 38.802959 -77.041569 2019-4-26 385000
## 8  Alexandria    VA 22310 38.776706 -77.109278 2019-4-26 495000
## 10 Alexandria    VA 22307 38.778635 -77.069521 2019-4-26 595000
## 11 Alexandria    VA 22303 38.800732 -77.094047 2019-4-25 340000
## 12 Alexandria    VA 22302 38.831096 -77.081177 2019-4-25 977800
```

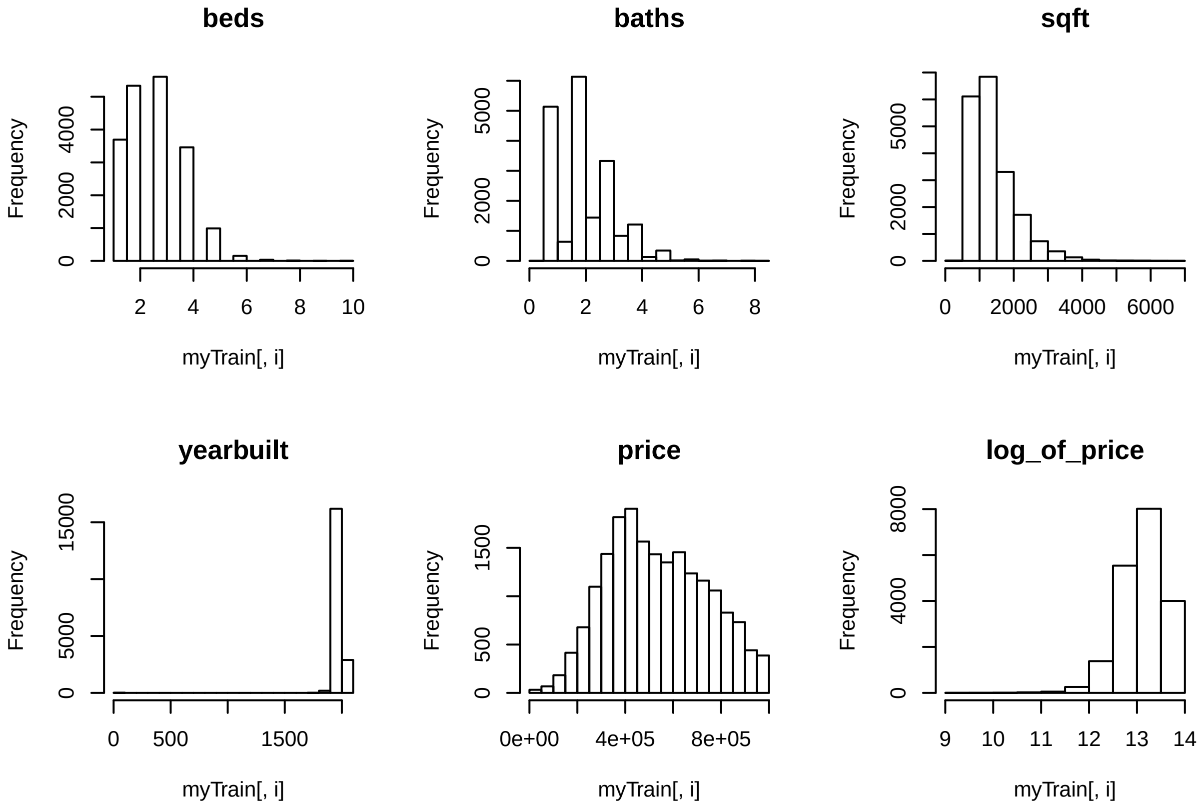
```
##        beds       baths        sqft   yearbuilt    proptype        addr
##   "integer"   "numeric"   "integer"   "integer" "character" "character"
##        city       state         zip         lat         lon    saledate
## "character" "character" "character" "character" "character" "character"
##       price
##   "integer"
```

```
##       beds            baths             sqft          yearbuilt
##  Min.   : 1.000   Min.   :0.000   Min.   :   2   Min.   :   0
##  1st Qu.: 2.000   1st Qu.:1.000   1st Qu.: 925   1st Qu.:1946
##  Median : 3.000   Median :2.000   Median :1218   Median :1960
##  Mean   : 2.656   Mean   :2.206   Mean   :1392   Mean   :1964
##  3rd Qu.: 3.000   3rd Qu.:3.000   3rd Qu.:1680   3rd Qu.:1986
##  Max.   :10.000   Max.   :8.500   Max.   :6642   Max.   :2018
##    proptype             addr               city
##  Length:19284       Length:19284       Length:19284
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
```

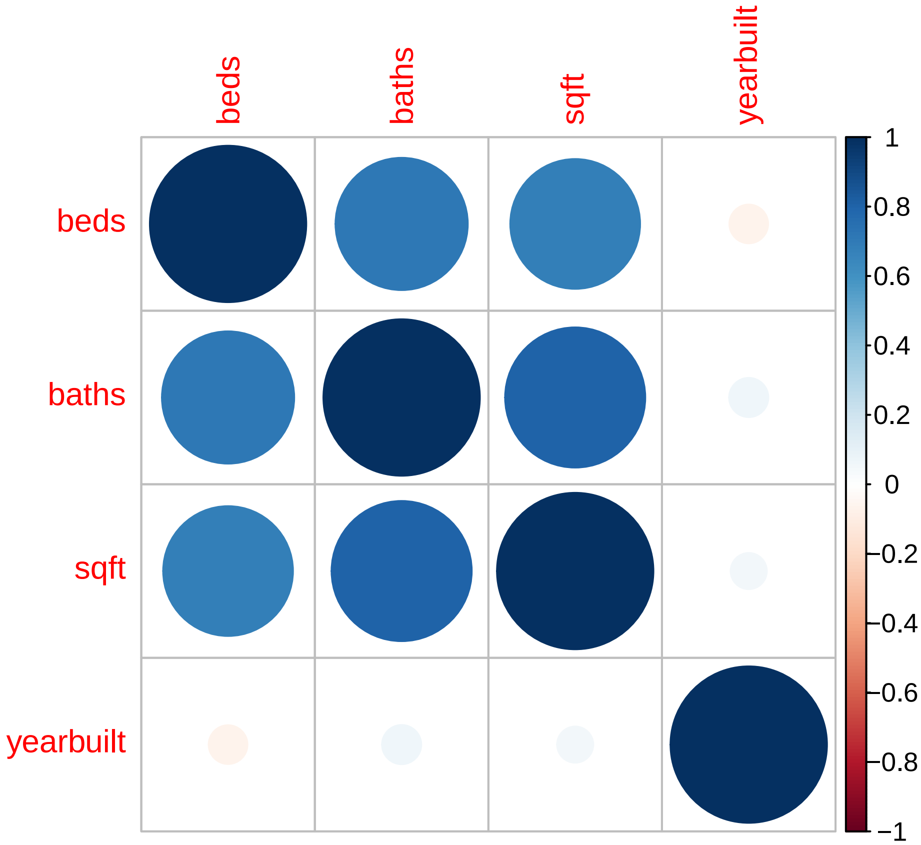
```
##
##      state               zip                 lat
##   Length:19284       Length:19284       Length:19284
##   Class :character   Class :character   Class :character
##   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       lon              saledate              price
##   Length:19284       Length:19284       Min.   :   9000
##   Class :character   Class :character   1st Qu.: 380000
##   Mode  :character   Mode  :character   Median : 519000
##                                         Mean   : 537423
##                                         3rd Qu.: 695250
##                                         Max.   :1000000
```



The bar plots are a little jagged. Lets get a different perspective by smoothing out the graphs using a density plot. This helps us better visualize the data to see if we have a binomial distribution, or other anomally.



Next, we looked the remaining columns to see how they correlated, to see if we can remove any columns that are highly correlated.



The data is highly correlated. We use the calculation below to determine which attribute, if any, we should remove, and the answer turns out to be baths. But for now, we'll leave them in the data.

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.

TODO

Part 4) CONCLUSION

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

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.)

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