ANLY 533 - Regression Module Assignment - Data Acquisition and Model Building

Predicting Home Prices Using The Zillow API and Multiple Regression

Due Date: November 1, 2017 @ 11:55 PM

Assignment Description

This assignment requires you to demonstrate two distinct skill sets. For the first part of this assignment, you will need to flex your R muscles to create two functions that will import and transform data from the Zillow API into clean dataframes for model building. Once you have acquired, cleaned, and organized the data using the functions you have written, you will then need to use the skills you have learned throughout the regression module to build a model that will predict the price of a given home. To succeed on the model building portion of the assignment, you will need to demonstrate competence at all stages of the CRISP-DM process discussed in class. Namely, you will need to demonstrate understanding of:

1. The business problem (Why would we want to predict home values?)
2. The data (What data is available and how can it be leveraged?)
3. Data preparation (Manipulating data from the API to create clean data frames)
4. Modeling (Variable selection, model evaluation/interpretation, diagnostic methods for checking assumptions, cross valiation)
5. Evaluation (Have you solved the business problem sufficiently? Why or why not?)
6. Deployment (Can the model accurately predict new cases as they arise? Does your code work well enough to handle test cases?)

Deliverables

You are required to submit 3 items:

1. A script containing the two functions you have written to acquire data from the API, with clear commenting so that the code can be easily understood
2. A script that clearly documents your model building process and includes a reproducible model object from which predictions can be generated from test data sets
3. A write up that clearly and concisely describes the business problem, the data, your modeling process, an interpretation of the final model (what do the coefficients represent in plain words), and a self-reflective evaluation of your solution to the business problem (Did you succeed? What are the problems that remain? How can this model be improved further?).

Details

Framing the Business Problem

Assume that you are working for a real estate startup and your boss has asked you to build a very specific model that will predict the value of a given home. You need to first choose a target property, which is the property whose value you wish to predict. This target can be your home address, the home address of a relative, or a property that you would consider buying. Your goal is to find a fair market price for your target address given the data you can obtain. To accomplish this goal, you will use the Zillow API to obtain data about properties that are similar to the target property. After you have acquired the data, you will need to wrangle it into a usable data frame.

Zillow API:

In order to interact with the Zillow API you will need to set up an account and obtain Zillow Web Services credentials.

You can obtain credentials via: https://www.zillow.com/webservice/Registration.htm

The ZillowR package contains a set of functions that can be used to interact with the API.

After you install and load the library, use the following to establish a connection:

```r
library(ZillowR)

set_zillow_web_service_id("insert your zws id here")
```

Deliverable 1: Function Script

After you get set up with the API, you will need to write two functions. These functions should be sumbitted in a single script.

Function 1: z_extract()

Build a function called z_extract() that can take the results of a call to

```r
ZillowR::GetDeepComps()
```

and transform the results into a clean, usable data frame.

Remember from class that the result of a call to the GetDeepComps() function results in XML output that must be converted into a complex list using the XML library, or another library of your choice.

As a hint, you can subset specific elements of the list into a more manageable form using map() like so:

```r
comps <- GetDeepComps(zpid = zpid, count = 25)

properties <- xmlToList(comps$response[["properties"]])

zpids <- map(properties$comparables, "zpid")
```

To be more specific about what you are expected to accomplish with z_extract(), assume that your function will need to perform variable extraction on the properties object in the above R code chunk. If you are interested in building a model from the Zillow API data you will need to extract a target variable, and all of the predictors you wish to examine. The z_extract() function should be able to take the list that results from the xmlToList() call in the code chunk above, and extract all of the variables that you need to build your model. *The function should take only one argument (called properties, which is the list object from the xmlToList() conversion) and return a dataframe (or tibble) object that could be used to build a model with no further data manipulation needed.*

Function 2: comp_finder()

Build a function called comp_finder() that takes two arguments (address, zip). This function should directly take the address and zip code for any real address in the Zillow API and deliver a large list of comps for the target. A comp is a comparable property that results from a call to the GetDeepComps() function from the ZillowR library.

The purpose of this function is to leverage the z_extract function and a loop to get a list of comps, and then a list of comps for those comps, and convert all of the results into a data frame that can be used for modeling.

An example list of steps for the function are as follows:

1. Get the zpid for the target property to search for comps
2. Get 25 comps from the API
3. Extract target and predictor variables from the comp data by converting from XML to a list and using z_extract()
4. Get the zpids from the comps and loop over them, employing the same process: converting from XML to a list and using z_extract()
5. Bind the data from the loop process to the original comp data from step 4
6. Clean up: Convert columns in the data frame to the appropriate types (character, numeric, etc.), remove duplicate listings, etc.
7. Return a large data frame with comp info that can be used for modeling

Obviously feel free to implement the steps differently to suit your style. However, steps 1 and 7 are universal. For example, if I were to test your comp_finder() function with the following:

```
comps <- comp_finder("2325 Newburg Rd.", 40205)
```

Then at minimum, the comps object should be a data frame containing data on 25 comps for the target address, and data on 25 comps for all 25 of the original comps. If every comp had 25 unique comps then there would be 625 rows in the data frame. However, many will likely overlap, so after eliminating duplicates, the function will likely return far fewer.

As you will likely need a sufficient number of rows to build a successful model, it is up to you to implement an appropriate number of loops to return the data you need. Feel free to implement loops over the dataframe at step 7 above to achieve an even larger number of possible comps (25 X 625 = 15,625).

For the modeling process you may also wish to expand your training data beyond the basic comps for the target. To do so, you may choose other similar houses in the neighborhood that are not strictly "comps". If you decide to do this, be sure to make it clear in your code at the modeling stage. **The data acquisition process should be reproducible.**

Deliverable 2: Model Building Script

Build a model from the data you acquired. This should be accomplished and submitted in a script that is separate from the function script.

Use the comp_finder() function you built to obtain the data you will use to build your model. As stated above, clearly document this process. If you use more than one target address to build your data frame, be sure to state your rationale.

Once the data is acquired, document the steps you take through the modeling process.

You are required to demonstrate knowledge of variable selection, model evaluation/interpretation, diagnostic methods for checking assumptions, and cross valiation.

Fully document all steps in the code.

Deliverable 3: Write up

This should be a Microsoft Word or Markdown document (rendered to PDF) that clearly and concisely describes the business problem, the data, your modeling process, an interpretation of the final model (what do the coefficients represent in plain words), and a self-reflective evaluation of your solution to the business problem (Did you succeed? What are the problems that remain? How can this model be improved further?)

Please include data visualizations as necessary. This document should be a stand-alone deliverable. Assume that you will be submitting this document to your boss at the real estate startup, who knows very little about data science. Given this context, your write up should not include code chunks or technical jargon. It should be an easily understandable walkthrough of the entire project. At a minimum, the key takeaway should be made obvious (If someone skimmed the document for 2 minutes, would they understand your model and how it solves the problem?)