Regression Week 2: Multiple Linear Regression Assignment 1

Predicting House Prices (Multiple Variables)

In this notebook you will use data on house sales in King County to predict prices using multiple regression. The first assignment will be about exploring multiple regression in particular exploring the impact of adding features to a regression and measuring error. In the second assignment you will implement a gradient descent algorithm. In this assignment you will:

- Use SFrames to do some feature engineering
- Use built-in GraphLab Create (or otherwise) functions to compute the regression weights (coefficients)
- Given the regression weights, predictors and outcome write a function to compute the Residual Sum of Squares
- Look at coefficients and interpret their meanings
- Evaluate multiple models via RSS

If you are doing the assignment with IPython Notebook

An IPython Notebook has been provided below to you for this assignment. This notebook contains the instructions, quiz questions and partially-completed code for you to use as well as some cells to test your code.

What you need to download

If you are using GraphLab Create:

- Download the King County House Sales data In SFrame format: kc_house_data.gl.zip (https://s3.amazonaws.com/static.dato.com/files/coursera/course-2/kc_house_data.gl.zip)
- Download the companion IPython Notebook: week-2-multiple-regression-assignment-1-blank.ipynb (https://s3.amazonaws.com/static.dato.com/files/coursera/course-2/week-2-multiple-regression-assignment-1-blank.ipynb)
- Save both of these files in the same directory (where you are calling IPython notebook from) and unzip the data file.

If you are not using GraphLab Create:

- Download the King County House Sales data csv file: kc_house_data.csv

(https://s3.amazonaws.com/static.dato.com/files/coursera/course-2/kc_house_data.csv.zip)

- Download the King County House Sales training data csv file: kc_house_train_data.csv (https://s3.amazonaws.com/static.dato.com/files/coursera/course-2/kc_house_train_data.csv.zip)
- Download the King County House Sales testing data csv file: kc_house_test_data.csv (https://s3.amazonaws.com/static.dato.com/files/coursera/course-2/kc_house_test_data.csv.zip)
- **IMPORTANT: use the following types for columns when importing the csv files. Otherwise, they may not be imported correctly: [str, str, float, float, float, float, int, str, int, int, int, int, int, int, int, int, str, float, float, float, float]. If your tool of choice requires a dictionary of types for importing csv files (e.g. Pandas), use:**

```
dtype_dict = {'bathrooms':float, 'waterfront':int, 'sqft_above':int, 'sqft_living15':float,
'grade':int, 'yr_renovated':int, 'price':float, 'bedrooms':float, 'zipcode':str, 'long':floa
t, 'sqft_lot15':float, 'sqft_living':float, 'floors':str, 'condition':int, 'lat':float, 'dat
e':str, 'sqft_basement':int, 'yr_built':int, 'id':str, 'sqft_lot':int, 'view':int}
```

Useful resources

You may need to install the software tools or use the free Amazon EC2 machine. Instructions for both options are provided in the reading for Module 1.

If instead you are using other tools to do your homework

You are welcome, however, to write your own code and use any other libraries, like Pandas or R, to help you in the process. If you would like to take this path, follow the instructions below.

1. If you are using SFrame, import graphlab and load in the house data, otherwise you can also download the csv. (Note that we will be using the training and testing csv files provided) e.g in python with SFrames:

```
sales = graphlab.SFrame('kc_house_data.gl/')
```

2. Split into training and test data Use this command to set the same seed for everyone: e.g. in python with SFrames:

```
train_data,test_data = sales.random_split(.8,seed=0)
```

For those students not using SFrames please download the training and testing data csv files.

From now on we will train the models using train_data. It will be important that we use the same split here to ensure the results are the same.

3. Although we often think of multiple regression as including multiple different features (e.g. # of bedrooms, square feet, and # of bathrooms) but we can also consider transformations of existing variables e.g. the log of the square feet or even "interaction" variables such as the product of bedrooms and bathrooms. Add 4 new variables

in both your train_data and test_data.

- ‘bedrooms_squared’ = ‘bedrooms’*‘bedrooms’
- ‘bed_bath_rooms’ = ‘bedrooms’*‘bathrooms’
- ‘log_sqft_living’ = log(‘sqft_living’)
- ‘lat_plus_long’ = ‘lat’ + ‘long’

Before we continue let’s explain these new variables:

- Squaring bedrooms will increase the separation between not many bedrooms (e.g. 1) and lots of bedrooms (e.g. 4) since 1^2 = 1 but 4^2 = 16. Consequently this variable will mostly affect houses with many bedrooms.
- Bedrooms times bathrooms is what's called an "interaction" variable. It is large when both of them are large.
- Taking the log of square feet has the effect of bringing large values closer together and spreading out small values.
- Adding latitude to longitude is non-sensical but we will do it anyway (you'll see why)

For those students not using SFrames you should first download and import the training and testing data sets provided and then add the four new variables each to both data sets (training and testing)

4. Quiz Question: what are the mean (arithmetic average) values of your 4 new variables on TEST data? (round to 2 digits)

5. Use graphlab.linear_regression.create (or any other regression library/function) to estimate the regression coefficients/weights for predicting ‘price’ for the following three models:(In all 3 models include an intercept -- most software does this by default).

- Model 1: ‘sqft_living’, ‘bedrooms’, ‘bathrooms’, ‘lat’, and ‘long’
- Model 2: ‘sqft_living’, ‘bedrooms’, ‘bathrooms’, ‘lat’,‘long’, and ‘bed_bath_rooms’
- Model 3: ‘sqft_living’, ‘bedrooms’, ‘bathrooms’, ‘lat’,‘long’, ‘bed_bath_rooms’, ‘bedrooms_squared’, ‘log_sqft_living’, and ‘lat_plus_long’

You’ll note that the three models here are “nested” in that all of the features of the Model 1 are in Model 2 and all of the features of Model 2 are in Model 3.

If you use graphlab.linear_regression.create() to estimate these models please ensure that you set validation_set = None. This way you will get the same answer every time you run the code.

Learn all three models on the TRAINING data set. Save your model results for quiz questions later.

6. Quiz Question: What is the sign (positive or negative) for the coefficient/weight for ‘bathrooms’ in Model 1?

7. Quiz Question: What is the sign (positive or negative) for the coefficient/weight for 'bathrooms' in Model 2?

8. Is the sign for the coefficient the same in both models? Think about why this might be the case.

9. Now using your three estimated models compute the RSS (Residual Sum of Squares) on the Training data.

10. Quiz Question: Which model (1, 2 or 3) had the lowest RSS on TRAINING data?

11. Now using your three estimated models compute the RSS on the Testing data

12. Quiz Question: Which model (1, 2, or 3) had the lowest RSS on TESTING data?

13. Did you get the same answer for 9 and 11? Think about why this might be the case.