Graded Assignment 2
(Due Tue 11/12)

ISOM 674, Fall 2019

The data we will use for this assignment is the same data we used in Graded Assignment 1, the data on the sale price of houses in a Midwestern city in the U.S.

As a reminder, a brief description of the data set follows:

- The dataset has 2930 observations.
- The data came from home sales during the period 2006 to 2010.
- Because I do not want to add the complication of time series into this assignment, I have "de-trended" the data and adjusted all of the prices so that you can treat them as all having occurred in May of 2010. To help make sure that there is no confusion about using time, I have also deleted the date of sale variable from the dataset.
- The data dictionary for the dataset is in the file: GradedHW1-DataDocumentation.txt
- For Assignment 1, I split the data into Training, Validation, and Test samples. For this assignment, we will use the same three datasets:

  GradedHW1-Train-Data.csv
  GradedHW1-Validation-Data.csv
  GradedHW1-Test-Data.csv

- We will only use the following 6 variables in the Assignment:

  Lot.Area
  Total.Bsmt.SF
  Gr.Liv.Area
  Full.Bath
  Bedroom.AbvGr
  Building Age

- Note that you will have to compute the Building Age (in years) as of 2010.

The goal of this assignment is to make sure that you understand and can compute regression trees and basic neural networks. Because both regression trees and neural networks are techniques that are intended to work for non-linear relationships, I want the example that we will use to be as non-linear as possible. Thus, I am going to have you work with variables that are not transformed. **But please understand that in practice you should transform your variables as this generally makes the problem simpler, easier, and more stable.**

Note: When I say I do not want you to transform you variables, this does not mean that I do not want you to rescale or standardize them. For example, most neural net algorithms require that you rescale variables to the interval [0,1].

1. Without transforming, standardizing, or rescaling the variables, fit a series of regression trees (using the "tree" package not the "rpart" package) to the data using the six x-variables indicated above. Fit trees with sizes k=20 to 60 end nodes. Evaluate the performance of these trees using the validation data. Make of plot of the $\sqrt{\text{MSE}}$ calculated from the validation data against k . Answer the questions on the Google form. Turn in the plot of the $\sqrt{\text{MSE}}$ vs. k as instructed below.

2. Determine the best k (in question 1) and then determine the (final) $\sqrt{\text{MSE}}$ estimate using the test data. For questions 1 and 2, answer the questions on the Google form.

3. Now repeat questions 1 and 2 using random forests. Fit the random forest with the maximum number of nodes ranging from 20 to 60. Determine the best model and then determine the (final) $\sqrt{\text{MSE}}$ estimate using the test data.

In Questions 4–8, I want you to experiment with fitting neural-net models using Keras and then visualize the fitted surfaces using R. This is very similar to what I did in class when I fit a neural net model to the auto data. So in order for you to be able to make the kind of 3D plots that we have been making (and rotating), we will only use the two x-variables:

Gr.Liv.Area
Building Age

Also, I would like the results of everyone's computation to be the same and not vary between runs as a result of the random starting weights used in neural nets such as those we have created in Keras. Thus the follow code should be at the top of your python file:

```
import numpy as np
import pandas as pd
import random as rn
import tensorflow as tf
np.random.seed(1234)
rn.seed(1234)
session_conf =
tf.ConfigProto(intra_op_parallelism_threads=1,inter_op_parallelism_t
hreads=1)

from keras import backend as K

tf.set_random_seed(1234)
```

```
sess = tf.Session(graph=tf.get_default_graph(),config=session_conf)
K.set_session(sess)
```

What this code does is set the random seeds for all of the random number streams that are used in the computation. Thus, the random number generators start in the same place each time and, therefore, the results of the computations should be the same each time. All of this is quite complicated, so just use the code I have given you above and don't sweat the details.

Finally, use 10,000 epochs when making the neural net fits unless that is just too slow on your computer. If you cannot reasonably use 10,000 epochs, then go ahead and use 5,000. For other parameters (like batch size) do the same thing I did when I fit the neural net to the Auto data.

4. Using only the two x-variables indicated above, fit the least squares linear regression using Keras. Make a 3D plots of the SalePrice against the two variables and add the regression plane. This is the exactly the same process I showed in class using the Auto dataset. Once you have fit the regression plane, find a good view that shows the data and the fit using the 3D plot. Save the view to turn in as indicated below. Note: Selecting the window and then using Cntl-Alt-PrtScr copies the window into the clipboard on Windows as a picture. You can then paste the picture into Word. Remember, you will need to label these plots, so keep track of what they are.
5. Next, fit a neural net model with a layer of 4 ReLU nodes (units). Recall that this will requires the first layer of 4 ReLU nodes to be followed by a layer with one node. Please use a linear activation function for the final node. Make a 3D plot of the fitted surface for this model. Once again, find a good view in the 3D plot that shows the data and the fitted surface and save the plot.
6. Repeat the previous question (Q4) using a layer of 10 ReLU units to see if making the model "wider" changes the fit significantly. Save a good view.
7. Next let's try a neural net model that is "deeper." Try 3 layers of 4 ReLU units. Make and save the plot as before.
8. Finally, try a neural net model that is both "wide" and "deep" by using 4 layers of 10 ReLU units. Save a good view as before.

Turn in the plots you saved as instructed below.

Instructions for Submitting the Plots

In addition to answering the questions on the Google form, you need to turn in the plots requested above. Turn these plots in as labeled figures in a pdf file named:

HW2-Plots-EmoryNetID.pdf

Please substitute your Emory Net ID for "EmoryNetID" in the file name above.

This pdf file should begin with your name and have no more than two plots per page. Please make sure that the plots are large enough to be easily legible (i.e., fill or nearly fill the width of the page) and are labeled with both the question number and a descriptive title.

One way to create the pdf file is to first create a MS Word document (with the file name format as above), add your name and then copy the graphs from R to the Word document. You can add labels to the figures in the Word document. Then export the Word document as a pdf.