**Week 7 Exercise Model Building (Linear Regression)**

**Do not forget to answer the Additional Questions for LR1 (below) AND include a brief 1-2 sentence interpretation of the results (R-squared) for LR2. Also note you must submit your assignment either in a Word document with code and visualizations included OR as a Jupyter Notebook file (.ipynb). Two documents/files, one for LR1 and one for LR2.**

You will complete two simple linear regressions for this exercise, for each dataset and problem statement, you should complete and include the following:

**General Required Elements**

1. **Model building and linear regression process:** Use scikit learn, train\_test\_split to fit the model, run the linear regression and predict the target variable (y).
2. **Model Results:** Print the intercept and coefficient of the line.
3. **Model Accuracy:** Print accuracy metrics from scikit learn, including r-squared
4. **Visualizations:** Create a least 3 visualizations:
   1. Before running the linear regression:
      1. (1) A scatterplot of the raw data, x vs y. See **Note: Scientific Notation in matplotlib** below.
   2. After running the linear regression:
      1. (2) Plot actual vs predicted values
      2. (3) Plot the error

* Finally be sure to **comment input and output of each code cell or cells that are associated with the 4 numbered elements above (model building/linear regression process, results, accuracy and 3 visualizations)**, with explanations geared to a non-data science colleague.
  + Do not simply copy the comments in the sample python code, since they are written to an audience of data science students, not ‘non-data science’ colleagues.
  + Use plain language and explain inputs and outputs for non-specialists.
  + HINT: you will need at least 2-3 sentences to explain what linear regression is, what it means to train\_test\_split and how we arrive at a result. You will need another 1-2 sentences explaining what r-squared is and how to interpret the result.

**LR1: Dataset: kc\_house\_data.csv (from the multiple regression tutorial)**

LR1 specific instructions:

* Run a simple linear regression using sqft\_living as the only feature.
* Be sure to include all of the General Required Elements above and include all python code and output.
* Use a training size of 0.8 so it is comparable to the multiple regression.
* **Additional Questions:** 
  + How does the simple linear regression model results (using sqft\_living) compare to the multiple regression we ran in the tutorial.
  + Which model (which features) better capture the variation in y based on the variation in the feature(s)? Justify your response based on output from your model(s).

**LR2: Dataset: student\_scores.csv.**

LR2 specific instructions:

* This simple dataset contains number of hours studied and corresponding results on an assessment.
* Create a simple linear regression model, using 0.8 as the training size.
* Include all the General Required Elements above and include all python code and output.
* While there are no additional questions for LR2, do not forget the comments on input and output. **Your analysis of the results is required.**

**Submission (two options, for both, two files maximum should be submitted, one for LR1 and one for LR2):**

1. You can submit your **Jupyter Notebook** (one per dataset, LR1 and LR2), with code, comments and output. Responses to the Questions in LR1 can be added to a Text Cell at the end of the Notebook. From Colab, use File - Download – Download .ipynb to download your finished assignment.

A picture containing graphical user interface

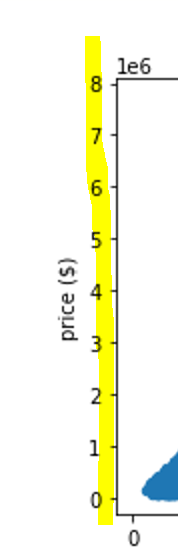
Description automatically generated

1. You can create a **Word document**, and cut and paste your python code, comments and output into the document. Images/graphics should be cut and pasted from Colab into the document (or the development environment where you are running Python). An easy way to start this is to download your python code as ‘py’ (which is basically a text file), open that file in NotePad, then cut and paste the code into word and add images/graphics as needed. Responses to the Questions in LR1 can be added to a Text Cell at the end of the Word file.

Graphical user interface, application, table

Description automatically generated

**Note: Scientific Notation in matplotlib**

If you are using matplotlib to create the scatterplot for your KC House data, you might see the y-axis for house price looks like this.

Matplotlib is converting our house prices to scientific notation (note the 1e6 at the top of the y-axis).  To remedy this, add the following line AFTER the final parentheses of your data.plot code OR just before plt.show(), depending on whether you created your data.plot in one 'go' or with several lines:

plt.ticklabel\_format(axis="y", style="plain")

Alternatively you can drop matplotlib completely and use seaborn's pairplot functionality:

<https://seaborn.pydata.org/generated/seaborn.pairplot.html>