ML Project Part 2:
Due Thur 5/5
No Late Submissions

Part 1: Data Preparation

1. If you are looking at a regression problem, plot the distribution of target values in your dataset. If classification, plot the distribution of classes. Give a high level summary of any other pertinent statistics / distributions relevant to your dataset.
2. Look for NA or missing values, impute with a value that makes sense, drop the row as a last resort.
3. Looking at the non-numeric columns in your dataset, decide which columns should be kept for model training. (hint: any unique identifiers should be dropped)
4. Of the set of non-numeric columns, decide which should be label-encoded, and which should be one-hot encoded, and then carry this out on the dataset.
5. Create at least 3 aggregation columns. This could be mean, std deviation, min/max, counts, and so on of other likely important columns. It is your choice which columns and which functions. This can easily be done with transforms(as we have done numerous times before) or pd.agg
6. Provide a brief summary of your logic / justifications for steps 3 & 4.
7. Split your data into features (x_data) and labels (y_data)
8. Using a scaling/normalization library of your choice, scale / normalize your data
9. Split your data into a test and training set randomly, with 10% of the data for the test set.
10. Steps 6,7, and 8 may be done slightly out of order based on your implementation.

Part 2: Training:

1. If you are aiming for classification, import the supervised learning models, linear regression and decision tree classifier from sklearn. If regression, import linear regression and logistic regression.
2. Fit / train both models on the training dataset / labels

Part 3: Post Model Analysis

1. Use model.score on your test set / labels to get an idea of accuracy
2. Print a comparison of the two models' accuracy
3. Change the distribution of training-test data to 20% test, 30% test.. All the way to 90% test 10% train. Store this data and plot the relationship between accuracy and percent of training data.
4. Using a residuals library of your choice, plot the residuals from the linear regression model. If you chose a classification problem, this plot will look slightly strange.
    a. Write a few sentences analyzing this plot.

5. If you chose classification, find a way to plot or visualize the performance of the decision tree classifier, if regression, do this for the logistic regression model. I expect you to do your own research on the model and check out a few articles showing how people typically plot this, and explore the performance on your own. I am giving flexibility here as long as I see you did some research.
6. Make an argument for which of the two models for the problem type you chose works better in this case.

Part 4: Submission

- Continue with submission criteria from Part 1. Github repo only, reminder of guidelines below. I expect all analysis as readme, formatted with markdown.
- Put a link to your dataset, all your plots, and your analysis code in the github.
- In the readme put your all of your findings with your plots included as inline attachments
- Turn in your link via canvas, ensure read access is allowed / open to public

Part 5: You're done

1. If you did all of these steps, you're looking pretty good in terms of starting a career in data science. Good luck with what comes next!