Problem Statement/Readme

- Ideal to have some quantitative description, particularly stating metrics for success early on.
- Rubric asked for the type of model being developed to be stated.

Data Cleaning and EDA

- Be clear about the difference between “Discrete”, “Continuous”, “Ordinal Categorical” and “Nominal Categorical” Variables/Features.
- Imputation for missing values are highly dependent on their variable type.
- Some methods demonstrated include dropping of rows, dropping of columns, replacing NaNs with median, mode, mean, “None” or 0.
- Provide justification for imputation.
- Always show a final NaN check on all cleaned data to act as proof and show confidence in your cleaning process.
- Consider the use of violin plots to combine distribution visualization with boxplots.
  https://mode.com/blog/violin-plot-examples/
  https://seaborn.pydata.org/generated/seaborn.violinplot.html
- Alternatively, combine scatter plots with distribution via rotated pdf.
  https://matplotlib.org/3.2.1/gallery/lines_bars_and_markers/scatter_hist.html
- Always good to identify and display outliers, and explain treatment (keep/modify/impute/drop).
- Rubric asked for the likelihood of answering problem statement with EDA discoveries.

Preprocessing and Modelling

- drop_first = False is the default in the .get_dummies() method and returns all values of the dummied column into its component columns..
- drop_first = True causes it to drop the first column of all dummied values instead. This is applicable when we have a single column we are breaking down for multi-label classification but not a good idea for regression.
- When defending model and using model metrics, good not to assume audience knows the quantitative basis for good scoring, eg. best score for r2 is 1.0, RMSE should be as small as possible.
- Briefly, but explicitly state the above. This should apply for all future projects, even beyond GA.
- Tie in explanation of limitations of model, and some discussion on actual coefficients of predictor variables vs their expected importance. This also helps showcase external research done, and domain expertise applied.

Project Organization

- Ensure no run-time error by restarting kernel fully and ensuring full functionality before saving for submission.
- As files can be too large for doing the above, this is further rationale for breaking up code into sub-notebooks.
- Always sequence sub-notebooks via their name, could be done by pre-fixing a number before the rest of the filename.

Visualization

- Choice of colours helps a lot for visualisation.
- Where possible, always add trend lines for scatterplots, and kernel density estimators (kde) for distributions.

Presentation

- Be explicit with your problem statement
- Level of technicality is appropriate
- Have a script
- Rehearse!
- Prefer visuals over words and numbers (picture speaks a thousand words)
- Keeping only the key points on slides, and elaborating on the rest

Overall

- Copy and paste the rubric in your notebook
- Delete everything else you don’t need
- Priortizing and time management