Homework II

You can submit in groups of 2!

All assignments need to be submitted via github classroom, assignment https://classroom.github.com/assignment-invitations/72d2784a4ce3f81acea1f73769ff7762

The repository should contain code to download the data and the code to process it, as well build and validate the model. Use travis to run your experiment and write a test that ensures the outcome is the actual error you report (Say, if you claim your algorithm is 90% accurate, write a test that checks that the score returned by your model is at least 90%).

We recommend that you fork the homework repository and run Travis on your own repository - this way you don't have to wait for other students submissions to finish on travis.

Task 1

A real estate agent wants to estimate the market rate for some apartments in NYC that just went into the market again. They want to use the data posted by the Census in 2014:
https://www.census.gov/housing/nychvs/data/2014/userinfo2.html
(The data is at https://www.census.gov/housing/nychvs/data/2014/uf_14_occ_web_b.txt)
You can find a parsed version on figshare: https://ndownloader.figshare.com/files/7586326

Create and validate a machine learning approach to predict the monthly rent of an apartment. Make sure to only use features that apply to pricing an apartment that is not currently rented. You can make the simplifying assumption that the market doesn't increase, so the rent for a new tenant would be the same as for the current tenant.

Explain how you validated your model and why.

Report the test error using R^2 and any other metric you deem appropriate.

Limit yourself to linear models for the prediction (though feature engineering, feature selection and imputation methods are allowed). Ensembles[1] of models are not allowed.

There should be a file "homework2_rent.py" with a function "score_rent" that returns the R^2 and a function "predict_rent" that returns your test data, the true labels and your predicted labels (all as numpy arrays).

[1] Ensembles are models that are created by averaging the result of multiple models. The goal here is to build a single model.

The tests should be in a separate file called ``test_rent.py" with a function called "test_rent" that checks the R^2 returned by score_rent to be as least as good as your expected outcome.