Sold! How do home features add up to its price tag?

Final Project
Fundamentals of Data Science 2016/2017

As a final project, you are taking part to the Kaggle competition "Sold! How do home features add up to its price tag?", whose goal is to predict house prices using a set of 79 features. See:

`https://www.kaggle.com/c/house-prices-advanced-regression-techniques`

You must create a Kaggle account, participate in the competition, and submit your predictions. Kaggle will assign you a score, visible in the public leaderboard:

`https://www.kaggle.com/c/house-prices-advanced-regression-techniques/leaderboard`

You will then send me the code you used to generate the predictions you submitted, and a short description of what you have done. More precise instructions follow.

Project submission

Dend to `fds2016lab@gmail.com` a mail with subject "`ID` final project", containing:

1. **the name of your Kaggle account, and your leaderboard score**
2. **a Python script**, named `ID.py`, that accepts on the command line (`sys.argv`) the name of the training set file and the test set file,

   ```
   $ python 123456.py train.csv test.csv
   ```

   and writes to disk a file named `pred.csv` (the same file you submit to Kaggle).
3. **a Python module**, named `libID.py`, where you can put all functions/classes/... needed by the script.
4. **a PDF file**, named `ID.pdf`, describing concisely your project. The file must consist in **at most one page of text in 11 points font**, plus (optionally) **at most one page containing only plots and figures**. Describe the main preprocessing steps (cleaning, normalization, ...), feature engineering steps (if you created new features from original ones), regression/learning steps, and anything interesting you may have found. Please be concise and quantitative.

Grading

Your grading will be based on three factors:

1. your score on the public leaderboard; a minimal "decent" score is required.
2. the soundness of your approach: for example, whether you have normalized the data, interpreted correctly the features, etc.).
3. the quality of your code: how well and clearly it is written, organised and commented, whether it uses the features of Python/NumPy/... that we learnt during the course, etc.

Emphasis is not on doing complicated things, but on using appropriately your coding and data analysis skills; so, do complicated things only if you think you know what you are doing.

Rules

1. project submission is individual, i.e., you cannot submit as a team
2. collaboration is encouraged through the official mailing list of the course
3. there are no restrictions on the libraries, algorithms, techniques you can use; notably, you can use scikit-learn (`http://scikit-learn.org/`)
4. copying is not allowed (this means: copying large pieces of code from other participants or from online sources is forbidden; copying small pieces of code that overcome technical problems is ok)

Important Dates

There are two deadlines for submission, one for each exam call. Deadlines are **strict**, so make sure to submit before expiration!

1. January 18, 2017, 23:59, Rome time (first call)
2. February 10, 2017, 23:59, Rome time (second call)

Remember to check that your script works correctly before sending it!!

GOOD LUCK!