DS 538 – Assignment-1
Due 06.11.2024 11:59 pm

Motivation:

In this assignment you are expected to implement multiple linear regression model and its extensions on a kaggle data set. The motivation is to learn one of the most used supervised learning models, namely regression, with

- bias/variance tradeoff
- optimization

perspectives.

You will:

- upload a public data set from kaggle.com
- practice matrix & array operations in NumPy
- use built in functions for multiple linear regression and Lasso models in sklearn
- write your own regression and lasso optimizers using SciPy.Optimize
- experience bias/variance tradeoff both for multiple linear regression and Lasso models

You will be dealing with three major tasks:

- model complexity
- optimization (minimization of *loss*) using sci.py
- regularization using variants of Lasso

Here is your task list:

- Load data set from https://www.kaggle.com/aungpyaeap/fish-market
  - Read the data explanation and understand what is there to be predicted!
- Append two random columns to the data frame. Note: You will need these columns in "model complexity task".
  - Populate first random column with data randomly generated between 10 and 100, name the column "Rand1"
  - Populate second random column with data generated from a six-faced (and unbiased) dice experiments, name the column "Rand2"
- Google "sklearn. linear_model" and learn how to run a multiple linear regression
- "Model complexity" task: Run the following linear regression models and for each model report: i) regressor coefficients (i.e. beta values) (do not forget the intercept!), ii) what percent of the variability is explained by the model
  - Model-1: Fit a regression model with all variables. Note: Once you successfully write your script, you will get the following error when you first run your regression model: "could not convert string to float". Google the error message and resolve the issue
  - Model-2: Drop column Rand1 and re-run regression,
  - Model-3: Drop column Rand1 and Rand2, re-run regression,
  - Model-4: Keep only the top two records (i.e. rows) and drop the rest of the records, re-run regression

  Compare your models and write a brief report about your observations.

- “Optimization” task:
    - Write your own “regression optimizer” using SciPy.Optimize. Make sure that you get the same results (coefficients etc.) with the regression models fitted in the previous task. Note: You may report only one version for the optimizer, say the one using all variables (i.e. Model-1 counterpart)
- “Regularization” task: Note: For all Lasso variants, use Model-1 as the benchmark model.
    - Implement Lasso with upper bound constraint. Start with a very large bound, i.e. 1000. Run the model and comment on your observations. Is regularization working? If not, what could be done? Comment and execute necessary steps.
    - Implement Lasso with Lagrange (penalty) multiplier. Start with a small penalty, i.e. 0.0001. Run the model and comment on your observations. Is regularization working? If not, what could be done? Comment and execute necessary steps.
    - Compare the two Lasso versions above once you were able to make regularization work. Are the two solutions the same? Why or why? Can you make the two solutions be exactly the same? If so, how?
    - Run Lasso using built in Lasso fitter in sklearn linear_model. For this sub-task, google sklearn. linear_model.Lasso() and learn how to run built-in Lasso function.

  Compare your models and write a brief report about your observations.

Submit you work through LMS website. If you prefer using Python, please submit your work as a Jupyter notebook. Make sure to include your comments and report as markdown within the notebook. For those of you who prefer other programming languages, please include your source code (properly commented) and report document in your submission.