1. **Decision Trees as Interpretable Models**

   (a) Download the Accute Inflamations data from `https://archive.ics.uci.edu/ml/datasets/Acute+Inflammations`.

   (b) Build a decision tree on the whole data set and plot it.[1]

   (c) Convert the decision rules into a set of IF-THEN rules.[2]

   (d) Use cost-complexity pruning to find a minimal decision tree and a set of decision rules with high interpretability.

2. **The LASSO and Boosting for Regression**

   (a) Download the Communities and Crime data[3] from `https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime`. Use the first 1495 rows of data as the training set and the rest as the test set.

   (b) The data set has missing values. Use a data imputation technique to deal with the missing values in the data set. The data description mentions some features are nonpredictive. Ignore those features.

   (c) Plot a correlation matrix for the features in the data set.

   (d) Calculate the Coefficient of Variation CV for each feature, where $CV = \frac{s}{m}$, in which s is sample standard deviation and m is sample mean..

   (e) Pick $\lfloor\sqrt{128}\rfloor$ features with highest CV, and make scatter plots and box plots for them. Can you draw conclusions about significance of those features, just by the scatter plots?

   (f) Fit a linear model using least squares to the training set and report the test error.

   (g) Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.

   (h) Fit a LASSO model on the training set, with λ chosen by cross-validation. Report the test error obtained, along with a list of the variables selected by the model. Repeat with standardized[4] features. Report the test error for both cases and compare them.

   (i) Fit a PCR model on the training set, with M (the number of principal components) chosen by cross-validation. Report the test error obtained.

[1] This data set is a multi-label data set. Sk-Learn seems to support building multi-label decision trees. Alternatively, you can use the label powerset method to convert it to a multiclass data set. Also, you can use the binary relevance method and build one decision tree for each label. It seems that the label powerset approach is more relevant here. Is that right?

[2] You can use the code in `https://www.kdnuggets.com/2017/05/simplifying-decision-tree-interpretation-decision-rules-python.html`.

[3] Question you may encounter: I tried opening the dataset and download it but the file is not readable. How to download the file? Just change .data to .csv. .

[4] In this data set, features are already normalized.

(j) In this section, we would like to fit a boosting tree to the data. As in classification trees, one can use any type of regression at each node to build a multivariate regression tree. Because the number of variables is large in this problem, one can use $\mathscr{L}_1$-penalized regression at each node. Such a tree is called $\mathscr{L}_1$ penalized gradient boosting tree. You can use XGBoost[5] to fit the model tree. Determine α (the regularization term) using cross-validation.

[5] Some hints on installing XGBoost on Windows: `http://www.picnet.com.au/blogs/guido/2016/09/22/xgboost-windows-x64-binaries-for-download/`.