Assignment #1

This assignment can be completed individually or by a team with up to 5 members.
Total score: 100
Due date: see the class page

Objectives

Conduct predictive modeling using supervised learning methods for regression and classification and write an analysis report based on the results of the modeling.

Only Python programs written using Python 3.0 or higher will be accepted. **NO Jupyter notebook or any Python variant will be accepted** for efficient grading.

Regression analysis

Chemical engineers measured various properties of a gas and created a training data set with four features *temperature (T)*, *pressure (P)*, *thermal conductivity (TC)*, and *sound velocity (SV)*. They also calculated the gas quality using four measured properties and converted it into a quality index (*Idx*). They want to know if any functional relationships exist between the measured properties and the gas quality. The schema of the data set (posted on the class page) is:

GasProperties*(T, P, TC, SV, Idx)* where the type of each attribute is double

(1) Use the **Least Square method** available in the Python package to find predictive models from the above data set. [20]

(a) Write a summary of modeling results for each model by order as shown in the following table format:

Polynomial order	Training RMSE	Training R^2	Training time	Testing RMSE	Testing R^2
Order 1	0.252004	0.671159	0:00:02.0469	0.252004	0.771159
…	…	…	…	…	…

(b) What is the sampling method used to create training and testing data sets? What is the percentage of training and testing data set used?

(c) Choose the best model and write it in the form of a polynomial function with a brief justification of your choice.

(d) Which data preprocessing method(s) is(are) used? Show an example row of data before and after data preprocessing. Did the preprocessing impact the modeling outcomes? If so, why? If you didn't preprocess the data, explain the reason.

(2) Implement the **Gradient Descent method** discussed in class to find predictive models without using any machine learning or statistical package. [30]
   (a) Write a summary of modeling results as shown in the above table format.
   (b) For the training, did you use the batch gradient or stochastic gradient? What is the total number of examples used for training?

   (c) Choose the best model and write it in the form of a polynomial function with a brief justification of your choice.

   (d) What is the initial weight vector and learning rate for the best model? Did the preprocessing impact the modeling outcomes? If you didn't preprocess the data, explain the reason.

(3) Use the LASSO method available in the Python package to find predictive models. [10]
   (a) Write a summary of modeling results as shown in the above table format.

   (b) Choose the best model and write it in the form of a polynomial function with a brief justification of your choice.

   (c) What is λ selected for each method?

   (d) Which features can be removed from the data set and why?

Text data processing and classification

A text data set "**emails.csv**" contains a collection of emails (corpus), each with an indication of spam (1) or non-spam (0). In this email training data set, the collection of entire emails (text data) in the training data set is called "**corpus**" in Natural Language Processing (NLP). Each row (email message) is referred to "**document**". In NLP, a document refers to a piece of text such as a list of words, a collection of sentences or paragraphs in a corpus.

Modeling for text data typically requires several processing tasks such as cleaning the raw data by removing special characters, stop words, whitespaces, lower casing, tokenization, stemming, and lemmatization. The sklearn.feature_extraction.text library, nltk.corpus, and others are available for common text data preprocessing tasks as well as other Natural Language Processing (NLP) tasks.

Especially, **CountVectorizer** from the package **sklearn.feature_extraction.text** allows you to create a **bag of words** (BoW) that represents a document as a set of words and the frequency of each word) and **document term matrix** (DTM) where each row corresponds to a document and each column corresponds to a unique word in the vocabulary from the corpus, and each element (cell) represents the frequency of the word appears in each document.

(4) Use the Logistic Regression and Naïve Bayes to find predictive models that can detect Spam emails (e.g., the sentiment classification example) and assess the performance of each classifier as requested below. [40]

(a) Briefly describe a step-by-step of your classification process and create a confusion matrix for each classifier.

(b) Compare the classification performance in terms of % accuracy and the precision metric.

(c) Create a ROC curve for each classifier.

(d) Based on the results of your analysis, which classifier would you recommend for Spam detection and why?

Warning: Although you can reuse any source codes available on the Internet, you are not allowed to share your codes with any other team or students in class. Any student or team violating this policy will receive a **ZERO** score for this assignment, potentially for all the remaining assignments.

What to submit

- An analysis report with your member names and % contribution made by each member in **PDF** or **Word format**. If every member contributed equally, state "equal contribution." If your team does not reach an agreement on individual contribution, briefly write a task description for each member. Different grades may be assigned based on individual contributions.
- Only two Python program files, one for regression and the other one for classification
- If the data is too big, select only a few examples and include them in the report.
- DO NOT submit any zip file. Instead, upload individual files.
- **Submit only one report and programs for each team**.

Grading criteria

- The overall quality of work based on the modeling results, analysis process, methods used, and programs
- The level of understanding shown in the report
- Effort