SFU SCHOOL OF MECHATRONIC SYSTEMS ENGINEERING

Simon Fraser University
School of Mechatronic Systems Engineering
MSE491 - Application of Machine Learning in Mechatronic Systems

Lab 1 – Regression
Due Date: Feb 21, 2021 23:59

Background

In this lab, we will become familiar with how to train a regression model using Python. Generally, regression models are sets of supervised machine learning (ML) algorithms to predict continuous outcomes using one or multiple independent variables.

- **Simple Linear Regression**

Linear regression is the oldest, simple, and widely used algorithm for predictive analysis.

Linear regression is used for finding linear relationship between target and one or more predictors. There are two types of linear regression- Simple and Multiple. In Simple linear regression we find the relationship between a dependent Y and independent variable X, the mathematical equation that approximates linear relationship between X and Y is

$$Y \approx \theta_0 + \theta_1 X$$

where θ_0 and θ_1 are two unknown constants that represent the intercept and slope terms in the linear model. Together, θ_0 and θ_1 are known as the model coefficients or parameters.

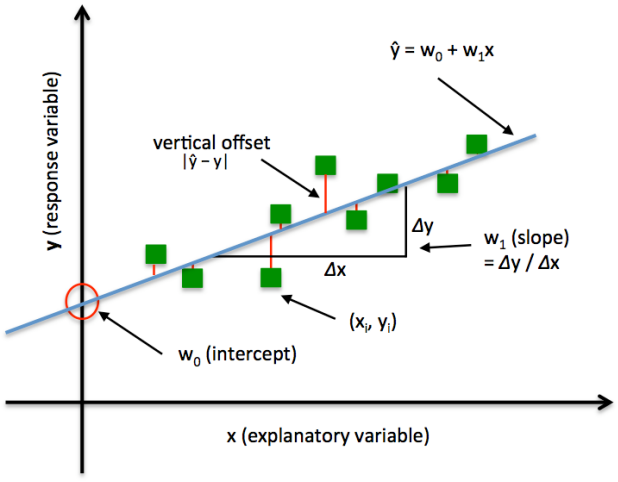

Figure 1. Simple linear regression. The goal of a linear regression model is to find the best fitted linear line (blue line) on the observed data (green points).

- **Multiple Linear Regression**

Multiple linear Regression is the most common form of linear regression analysis. As a predictive analysis, the multiple linear regression is used to explain the relationship between one continuous dependent variable (Y) and two or more independent variables (X_i).

$$Y \approx \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_n X_n$$

where θ_i are slope coefficients for each explanatory variable.

- **Polynomial Regression**

Polynomial regression can be regarded as a generalized case of linear regression. You assume the polynomial dependence between the output and inputs and, consequently, the polynomial estimated regression function. In polynomial regression, we have a polynomial equation of degree n represented as:

$$Y \approx \theta_0 + \theta_1 X + \theta_2 X^2 + \cdots + \theta_n X^n$$

where θ_i are the weights in the equation of the polynomial regression, and n is the degree of the polynomial.

- **Non-linear Regression**

Simple linear regression relates two variables (X and Y) with a straight line, while non-linear regression relates the two variables in a non-linear (curved) relationship. Non-linear models are more complicated than linear models to develop because the function is created through a series of approximations (iterations) that may stem from trial-and-error.

Machine Learning in Python

The ***Scikit-Learn***[1] Python package provides various algorithms for implementing supervised and unsupervised learning algorithms for machine learning tasks. Scikit-Learn is built upon standard libraries such as NumPy, pandas, and Matplotlib. This library is focused on modeling data. It is not aimed to offer solutions for loading, manipulating, and summarizing data. For these features, refer to NumPy and Pandas. It is worth noting that for implementing more complex ML algorithms on a high number of data, other frameworks (e.g., ***TensorFlow***[2], ***PyTorch***[3], etc.) are developed to take advantage of Graphics Processing Units (GPUs) for more efficient training.

[1] https://scikit-learn.org/stable/index.html
[2] https://www.tensorflow.org/
[3] https://pytorch.org/

MSE 491- Assignment 1 (Spring 2021)

Gas Turbine Data Set

This dataset is composed of hourly average sensor measurements of eleven variables (eight input and three target variables). There are a total of 36,733 instances collected over 5 years. The eight input measurements (independent variables) can be grouped into two as ambient variables (e.g., temperature, humidity, pressure) and process parameters (e.g., air filter difference pressure).

The names, abbreviations and basic statistics of the variables used in the study are summarized in Table1. In Figure1, the locations of sensors and sources of turbine parameters are shown on the illustration of the gas turbine. See the attribute information and relevant paper for details[4].

Table 1. Basic statistical information of data used in the study

Index	Variable	Unit	Min	Max
F1	Ambient temperature (AT)	°C	6.23	37.1
F2	Ambient pressure (AP)	mbar	985.85	1036.56
F3	Ambient humidity (AH)	%	24.08	100.2
F4	Air filter difference pressure (AFDP)	mbar	2.09	7.61
F5	Gas turbine exhaust pressure (GTEP)	mbar	17.7	40.72
F6	Turbine inlet temperature (TIT)	°C	1000.85	1100.89
F7	Turbine after temperature (TAT)	°C	511.04	550.61
F8	Compressor discharge pressure (CDP)	mbar	9.85	15.16
T1	Turbine energy yield (TEY)	MWH	100.02	179.5
T2	Carbon monoxide (CO)	mg/m^3	0	44.1
T3	Nitrogen oxides (NOx)	mg/m^3	25.9	119.91

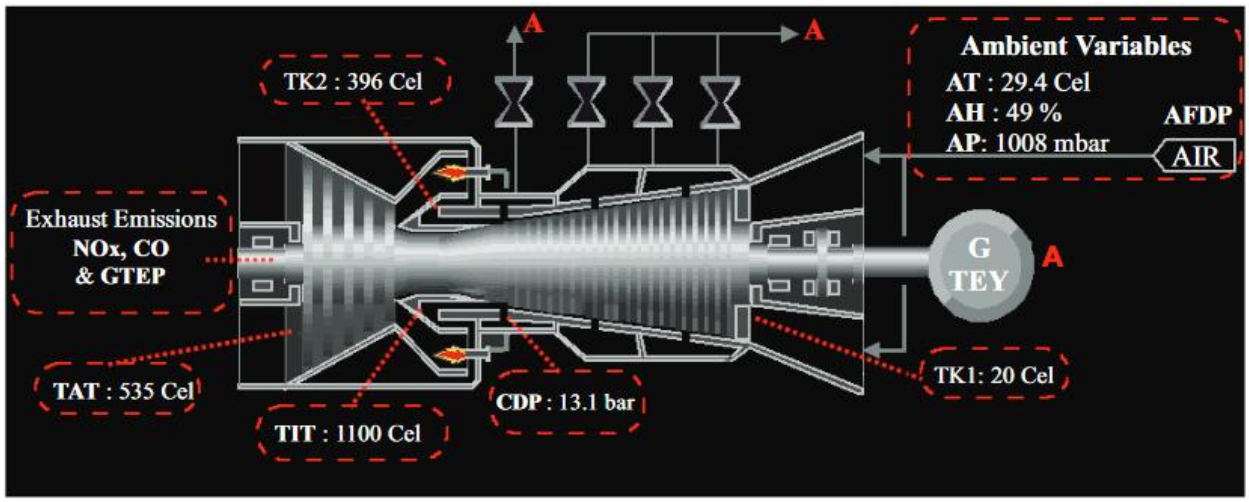

Figure 1. The sensor locations/parameter sources for all input and output variables used in the study.

[4] https://journals.tubitak.gov.tr/elektrik/issues/elk-19-27-6/elk-27-6-54-1807-87.pdf

Getting Started

Usually, once the data is collected, it must be explored to assess its conditions, including looking for trends, outliers, exceptions, incorrect, inconsistent, missing, or skewed information. This step is essential because your source data will inform your findings, so it is critical to be sure it does not contain unseen biases.

1. Load the datasets into Python.
2. Concatenate data from all five years into a single array.
3. Generate a histogram with 100 bins for the Nitrogen oxides (NOx) emission. Does it look like a normal distribution? **(5 points)**
4. Generate a scatter plot for each of the following cases and explain whether there is an apparent linear or non-linear association between them or not:
    a. Turbine inlet temperature (TIT) versus Turbine after temperature (TAT) **(5 points)**
    b. Turbine inlet temperature (TIT) versus Turbine energy yield (TEY) **(5 points)**
    c. Compute the Pearson's correlation coefficient (*scipy.stats.pearsonr*) for parts a and b and explain what these coefficients tell about the association between our variables of interest? **(5 points)**
5. Split the data into training (80%) and test set (20%).

Simple Linear Regression

One way to evaluate the performance of a regression model is to get a measure of the spread of the real values around the predicted line/curve. To do this, we usually use the root-mean-square error (RMSE). To construct the RMSE, you first need to determine the residuals. Residuals are the difference between the actual values (y_i) and the predicted values ($\hat{y}_i$). They can be positive or negative as the predicted value under or above estimates the actual value. Squaring the residuals, averaging the squares, and taking the square root gives us the RMSE.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

Based on the linear correlation between a single feature (from F1 to F8) and one of the targets (T1 to T3), select one feature with the highest correlation for each target.

1. Build a simple linear regression model for each case and save the trained models. **(5 points)**
2. Evaluate each model's performance on the test set and report the R2 and RMSE metrics. (you can use built-in functions of sklearn; *sklearn.metrics*) **(5 points)**
3. Write your own function for calculating the RMSE for each sample point and save the result vector in an array. Use this array to generate a bar chart (*matplotlib.pyplot.bar*) to understand the difference between each real and predicted value. Discuss about the performance of each model by comparing the bar charts. **(10 points)**

Multiple Linear Regression

1. Using all features (F1 to F8) build multiple linear regression models to predict the following targets:
   a. Turbine energy yield (TEY)
   b. Carbon monoxide (CO)

   Evaluate these models on the test set and report the results by providing the previous metrics (R2, RMSE, and Bar Charts). **(10 points)**

2. Compare the performance of each model with the simple linear regression task and explain which model could predict the target better? Does employing more features to train the ML model always improve the performance? **(5 points)**

Feature Selection

Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in. Having redundant features in your data can decrease the models' accuracy and make your model learn based on irrelevant features. One of the easiest ways of feature selection is to rank features based on their correlation coefficients, but there are multiple available approaches to accomplish this task.

Repeat the multiple linear regression task but this time, only use four features with highest correlation coefficients with targets. Explain the rationale of choosing these four features. Evaluate your new results and compare them with the previous task. Use bar charts and evaluation metrics to make comparisons. **(15 points)**

Polynomial Regression

a. Using a polynomial regression, train a model based on F1 to F8 to predict the Turbine energy yield for each following case and report the evaluation results: **(10 points)**
   1. Use a polynomial regression with the degree of 2 for training.
   2. Use a polynomial regression with the degree of 5 for training.
   3. Use a polynomial regression with the degree of 11 for training.

b. Compare the performance of your models using different polynomial degrees. Check for underfitting/overfitting for each case and explain the potential solutions to overcome each case. (you may need to do a brief research on these issues and provide a summary of your findings) **(10 points)**

Raspberry Pi

Implement one of your trained models on the Raspberry Pi kit and record a short video showing you can connect it to your device and getting results. **(10 points)**

Submitting your Assignment

The assignment must be submitted online at Canvas with the following structure:

For each group, you must submit **one zipped folder** that contains:

1. **A report**- The assignment report must be in **PDF format**, called **report.pdf**. This report must contain all the figures and discussions/explanations for the questions.
2. **A video**- The recorded video from Raspberry Pi function in the **MPEG4 format**, called **video.mp4**. This video should not exceed 3 minutes. Make sure to introduce all the members of your group at the beginning. This video must clearly indicate you could successfully connect the Raspberry Pi kits to your system, and you are able to run the code and produce results. You can also discuss the performance of the model in the video.
3. **A zipped file that contains all your codes**- This zipped folder must be called **code.zip** and must have a single directory called code. No sub-directories and leading path names are accepted. For each question, a separate script must be included in this folder.