ENGR 301
Engineering Applications of Data Science

Laboratory 4
Topic: Mechanical Engineering; Subtopic: Mechanical Part Degradation

Introduction: As mechanical parts in applications such as engines and turbines are used, they constantly undergo degradation from vibrations, torque, and natural phenomena, among other things. In this lab, you will use a variety of regression models to predict damage and reasons for failure of parts from ships. The system under study here is a "combined diesel-electric and gas" (CODLAG) propulsion system. This system is used on heavy ships that need a high maximum speed, such as modern warships. In this type of propulsion system, electric motors are powered by a diesel generator and in turn drive the ship's propellors; however, if the ship needs to go faster, a gas turbine is used to power the shafts. An example is shown in Figure 1. You will be provided with two data sets in the form of CSV files to do this analysis, with the details of each explained below.

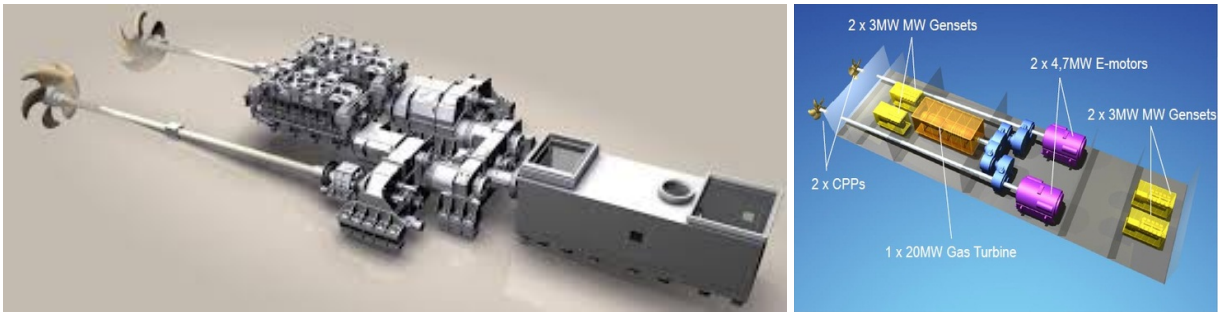

Figure 1. (left) Image of a CODLAG propulsion system. (right) Detailed schematic of a CODLAG propulsion system.

Discipline Specific Information: For this laboratory, there are too many features to list each one individually, so a general description will be given here.

Data Set 1: Condition of a gas turbine on a ship (part_condition.csv)

In this data set, scientists used a simulation to model the degradation of two parts (compressor and turbine) of the gas turbine (GT) in a CODLAG system on a ship. They varied a number of the ship's parameters and recorded the damage done to these parts as each parameter was increased.

The parameters include things such as the ship's speed, the temperature and pressure of the input and output air flow, the rate of the GT's revolution, flow of fuel, and torque produced on the propellor.

The damage to each part of the GT (and thus the desired outputs) is characterized by two decay coefficients: the "GT Compressor decay state coefficient" and "GT Turbine decay state coefficient". (In both cases 1 is perfect condition and it goes towards 0 as it becomes more damaged).

The goal of analyzing data set 1 is to predict the degradation based on the ship's parameters using linear regression.

Data Set 2: Type of defect causing degradation of a part (defects.csv)

Following degradation of these parts in a real ship, engineers took samples from the damaged parts and identified what type of defect was causing the degradation. They measured the shape of each sample and took an image of each one, which were digitized and processed.

In this data set, there are 27 features that describe (1) each image via characteristics such as a pixel's color, brightness, etc., and (2) the shape and details of the part they tested (such as part size, length, thickness, etc.).

There are also 7 types of possible defects responsible for the degradation; each sample/image has 1 of these assigned to it.

The goal of analyzing data set 2 is to predict what type of defect is present in a sample given an image and its shape using logistic regression.

Tasks

Note: When model_name is specified, this means you should replace it with the name of your model.

Part 1 – Linear Regression, Regularization, and the Limits of Linear Regression

1. Import the appropriate packages (numpy, pandas, matplotlib, seaborn) into a Jupyter notebook. The data for this part is provided on Canvas in a CSV file called:

part_condition.csv

2. Oftentimes, you will get data that is unlabeled, as is the case here. Luckily, however, we were provided with what each column represents.

Copy the array of column names from the provided "column_names.ipynb" file and copy it into your data frame.

Using *pandas*, we can both import this data set *and* clean it at the same time. Try doing the following (make sure the path to your CSV file is set correctly).

df = pd.read_csv("part_condition.csv", delim_whitespace = True, header = None, names = columns)

3. Do some exploratory data analysis. Does it look like any features are strongly correlated to either of the decay coefficients?

4a. Build two linear regression models using all of the features as your x values; one using the "GT Compressor decay state coefficient" as the y output, and one using the "GT Turbine decay state coefficient" as the y output. Use 30% of your data as the test set in each case.

Predict y values based on the x test values.

Which decay coefficient is better predicted using a linear regression model (look at R^2 of the training and test for both models)?

4b. Plot the residuals for both linear regression models. Are your linear regression models appropriate for this analysis?

4c. Describe your models in words. What are you predicting and what are you using as input? If you used these models in the real world, what would you be doing?

5. For this part, we will only investigate the "GT Compressor decay state coefficient" output. YOU DON'T NEED TO DO THE OTHER ONE.

We will now test the effect of regularization. Create a new Ridge regression model. We can do this in a similar way to linear regression:

```
from sklearn.linear_model import Ridge
model_name = Ridge(alpha = value)
```

Here alpha is our penalization parameter. Fit and predict a model with an alpha of 0.01; what is the R^2 value compared to simple linear regression? (Don't forget to take the same steps as in linear regression: test/train split, model fit, and model predict).

What happens if you change the penalization parameter to 0.1, 1, and 10? Do you notice a trend in the R^2 value? What is regularization supposed to do? Why might this trend you observe be happening? (Maybe look back at your correlation table)

Part 2 – Logistic Regression

1. The data for this part is provided on Canvas in a CSV file called:

defects.csv

Import this data into a data frame.

The type of defect responsible for the degradation in a given part is labelled as "fault_type".

2. What is the distribution of fault types? (*i.e.*, how many of each type are there in the data frame?)

3. Create a logistic regression model in which your y (output) is the fault type and your x is all of the features. Use 30% of your data as the test set. Use 10,000 iterations to train your logistic regression model (*i.e.*, set max_iter = 10000 when you call the LogisticRegression() function).

NOTE: THE LOGISTIC REGRESSION MODEL WILL LIKELY NOT CONVERGE. It will also take some time to train. That is okay, you can proceed with the remainder of the parts.

Predict new y values based on the x test values using your trained model.

4. Create a confusion matrix and determine the accuracy of your model on the test data. To do this, first import the "metrics" object from sklearn:

from sklearn import metrics

Then call the metrics.confusion_matrix() object and compare the y test data to your predicted y data.

You can do the same with the metrics.accuracy_score() object.

What is the confusion matrix telling you about how well your model is classifying the various defects? Which one is the most improperly classified?

5. Your boss has recently measured and taken images of a new set of 1,000 degraded parts, but he doesn't have time to do the analysis to figure out what type of defect caused the degradation. Your boss's data is provided in the CSV file:

new_defects_data.csv

Using your trained logistic regression model, predict the type of defect that caused each of the part failures.

What is the distribution of the part failures? Are you confident in your response to your boss?