Anomaly Detection Preprocessing.ipynb

Notebook Aims:

- We will download data from Bearing Data Center.
- We will write functions to read the data, preprocess the data and examine the data.
- We will not write a model. Model is for the next notebook.

Info on the Data and Our Task

- Data is vibration sensor data from manufacturing machines. A vibration sensor is stuck onto the machine, to measure the vibrations on the machine.
- If the machine vibrates weirdly, there might be a problem and maintenance for the machine should be considered.
- If the maintenance is not provided; machines might break down at an inappropriate time, and there might be pauses in our manufacturing line, costing us money. It costs us money because while the machine is broken, we cannot manufacture products and sell them, and we miss our production deadlines.
- We will detect anomalies in these vibrations in the next notebook (model notebook), to report a “maintenance issue” if the machine is vibrating weirdly. But now, we only read and examine the data.

Notebook Cell Explanations:

Download Data Section

- Make the imports
- Download healthy data with linux system command wget
- Check if the data is there, make a directory, move the data there.
- Do the same things for faulty data

Check .mat files Section

- We write a function to check the “keys” in a .mat file (this file acts like a python dictionary).
- We write another function to check the data files in a folder, using our previous function.

Read Folder Section

- A function to read the data in a folder and combine all data in one numpy array.
- For every .mat file, it reads the file and takes the data values using the relevant keys of the dict. (these keys are `'DE_time' and 'FE_time'`)
- It uses a variable to aggregate the data, the var is called “data”. If “data” is empty, that means we are reading the first file, so we assign the information in the variable “a” to “data”.
- Variable “a” is to store the particular value of a particular file that we read just right now.
- We skip some information if the shapes do not fit between “data” and “a”.
- We put an “id” column to the “data” object, to know which rows come from which file
- We also put the third column values (just to fill, as zeroes) to the relevant rows, if the third column is empty.

Read & Save Section

- We check if our function works
- We use our function, then we save the outputs to our drive as .csv files

Data Examination 1

- We check if our .csv files are saved correctly, by examining id columns.

Data Examination 2

- We only use the relevant columns (meaning, not id column, but value columns) and form a numpy array
- We plot our data using the numpy array

Data Examination 3

- We do other plottings, to:
    - Compare the amplitudes of the signals
    - Compare the lengths of the signals
    - Zoom in and compare the shapes of the signals