Computer Science Department
CS677 – Machine Learning (CRN: 22150) – Spring 2024
Professor Enze Bai
Project #2 | Due: March 19, 2024

This problem is a typical anomaly detection task. You must build various Unsupervised Learning models to detect whether a dataset contains anomalies or not.

The dataset for study is one that contains the temperatures (in Fahrenheit degrees) of a device through time. Namely, on specific dates and specific times throughout a day. It contains only two columns, the date/time portion and the corresponding temperature of the device:

	timestamp	value
0	2013-07-04 00:00:00	69.880835
1	2013-07-04 01:00:00	71.220227
2	2013-07-04 02:00:00	70.877805
3	2013-07-04 03:00:00	68.959400
4	2013-07-04 04:00:00	69.283551

The dataset **(temperature_device_failure.csv)** is provided, uploaded together with this document.

Perform the following tasks:

1) **Plot / Visualize** the 'original' dataset (hint: this is a Time Series object)

2) Perform **Feature Engineering** on the dataset such that new features to be added. Specifically, you need to create a feature that will indicate the day of the week and time of the day. Namely, there should be four (4) categories (clusters?) for the feature, name it 'dtcat' (date-time-category):
   - Weekday Day
   - Weekday Night
   - Weekend Day
   - Weekend Night

Note: Some features such as 'dayofweek', 'hours', 'day', etc. may remain in the dataset.

We define the duration of 'Day' and Night' as follows:
Duration of **'Day'** should be defined: 7:00am - 7:00pm
Duration of **'Night'** should be defined: 7:01pm - 6:59am

Ultimately, we would like to figure out when (weekday, weekend, day or night) the device fails!

3) Apply the **K-Means** algorithm to the revised dataset and determine the **best value for K.** I would suggest to test K in the range of [1, 20]. Plot a graph showing the number of clusters (K) in relation to score of each K-Means model. (Look at slide #20 from lecture for setting up the plot...)

4) After determining the best value of K, plot (scatter plot) all these K clusters by choosing 2 features from the dataset. Should the dataset has more than 2 features (which most likely will be the case), apply **PCA** to derive those 2 features **(2 Principal Components)** [pca = PCA(n_components=2), then 'fit' pca into the dataset]

All following steps should be executed twice:
- for **outliers_fraction = 0.01 (1%),** assume that someone gave us this figure
- **calculate outliers_fraction** by finding the total number of outliers utilizing the IQR Method; specifically use the **1.5 x IQR rule**

Identifying outliers with the 1.5 x IQR rule
https://www.khanacademy.org/math/statistics-probability/summarizing-quantitativedata/box-whisker-plots/a/identifying-outliers-iqr-rule

5) Apply the **Gaussian** distribution (EllipticEnvelope) algorithm, as defined at step 2. (Use this command: **from sklearn.covariance import EllipticEnvelope)**
List anomalies (if any) in each category and show them graphically.

6) Apply the **Isolation Forest** algorithm at each category, as defined step 2. (Use this command: **from sklearn.ensemble import IsolationForest)**
List anomalies (if any) in each category and show them graphically.

7) Which of the two (2) **models** performs **better** on detecting anomalies?

Write **Python** scripts in order to complete the above tasks along with their output. All work should be done and submitted in a single **Jupyter (or Colab) Notebook.**