Machine Condition Monitoring with Clustering

Acme Corporation is a worldwide supplier of technological equipment. The factory is facing significant problems with their manufacturing line, the machines are constantly facing failures due to a lack of maintenance and the production is stopped every time an unexpected failure is presented. As a result, Acme is losing millions of U.S. Dollars and important clients like Wile E. Coyote are experiencing delays in deliveries.

conditions.Their objective is to develop a machine learning model able to monitor the operations and identify anomalies in the sound pattern.

The implementation of this model can allow Acme to operate the manufacturing equipment at full capacity and detect signs of failure before the damage is so critical that the production line has to be stopped.

Mission is to build a machine learning model for Acme, so they can continue their manufacturing activities and help the Coyote to catch the roadrunner. For the model in this part I used unsupervised ML model which is clustering. In this term I tried to make a classification to normal, abnormal and transition sounds.

First of all, the data given by the company was examined before starting the project. As a result of the examinations, it was observed that the data set was 100.2 GB in total. Instead of downloading the data set automatically from the link via the required function, all data sets are downloaded manually. Thus, it was estimated that the data in the data set could be accessed quickly at any time.

The table you see below shows us the content of the data set.Also you can download the data set with a link which is exist at references part.

Table 1: MIMII dataset content details

Machine type/	model ID	Segments for normal condition	Segments for anomalous condition
Valve	00	991	119
	01	869	120
	02	708	120
	03	963	120
	04	1000	120
	05	999	400
	06	992	120
Pump	00	1006	143
	01	1003	116
	02	1005	111
	03	706	113
	04	702	100
	05	1008	248
	06	1036	102
Fan	00	1011	407
	01	1034	407
	02	1016	359
	03	1012	358
	04	1033	348
	05	1109	349
	06	1015	361
Slide rail	00	1068	356
	01	1068	178
	02	1068	267
	03	1068	178
	04	534	178
	05	534	178
	06	534	89
Total		26092	6065

When we look at the data after entering the set, we see that the data is recorded as an audio file in .wav format for 10 seconds. This shows us that we need to perform audio feature extraction in order to model the data.

While categorizing the properties of audio files, we know that there are various levels and abstraction.

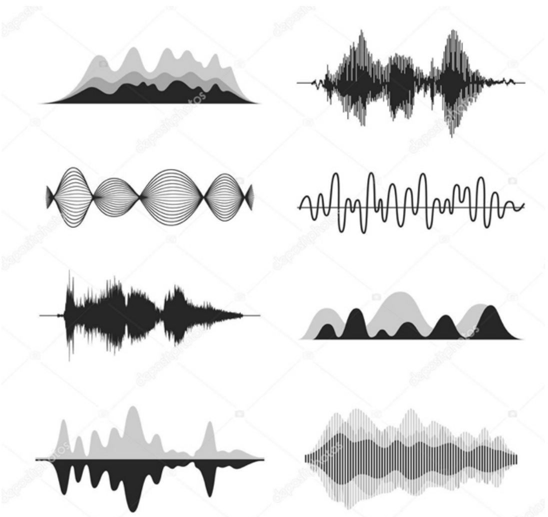
High level: These are the level that can be heard and not disturbed by humans. These can be instrumentation, key, chords, melody, harmony, rhythm.

Intermediate: These we may have trouble perceiving. These include pitch, beat-related identifiers, note beginnings, ripple patterns, MFCCs, etc.

Low level: These contain features that humans cannot understand but can only be achieved with the help of machinery. Examples include amplitude envelope, energy, spectral center, spectral flux, zero crossing ratio, etc.

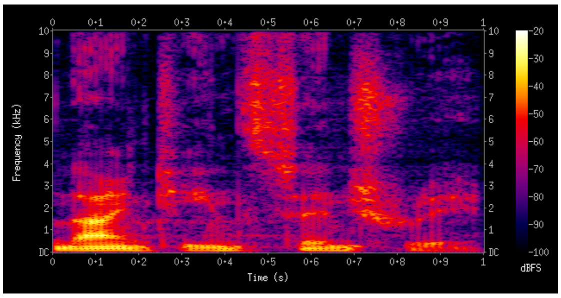
In fact, when we look at it, all the information we need is hidden inside the audio files, and all we need to do is to discover the features we need and extract them from the audio file. Waveforms based on time domain contain raw sound features. These can be exemplified as **Zero crossing rate, amplitude envelope, and RMS(Remote Mean Square)**.

When we look at waveforms with frequency domain, we can extract **band energy ratio, spectral centroid, and spectral flux properties** with the help of Fourier Transform.



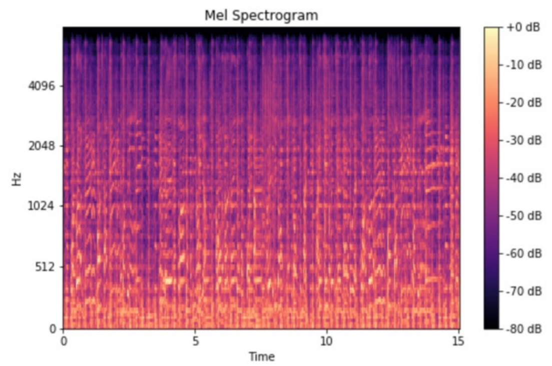
As a result of our research, the features that are important to us are included in the domains I have listed above, and it has been observed that all of them can be accessed at the same time with the Time-frequency representation that includes each of these domains. **Mel spectogram** will be created with the help of **the Short-Time Fourier Transform (STFT)** and the desired properties will be obtained from here.

What are spectrograms?



A spectrogram is not a visual rendering of the frequency spectrum of a sound form that changes in direct proportion to time. While it shows the time in the X line, it shows the frequency in the y line. It is obtained by applying STFF after reading the audio file with the help of python librosa library.

What is a mel-spectrogram?



Humans perceive sounds logarithmically. When the sound frequency range is high, it is easy for us to notice these sounds. For example, if we give an example, we can easily detect the difference between 300 hz and 700 hz, but it will not be easy to make this distinction when this range increases from 10000 to 20000. While making this distinction, the mel spectogram is used. The sounds we will be working with are machine sounds and in some places it will be almost difficult to tell the difference. Therefore, while designing the model, using a system separated according to their characteristics, not a human being who distinguishes sounds, indicates that it will be very effective in solving the sample problem. Mel-spectrogram is a spectrogram in which frequencies are converted to mel scale.

While designing the project, it was requested to separate the desired normal abnormal and transition sounds from each other and to use clustering, which is one of the machine learning models, while doing this.

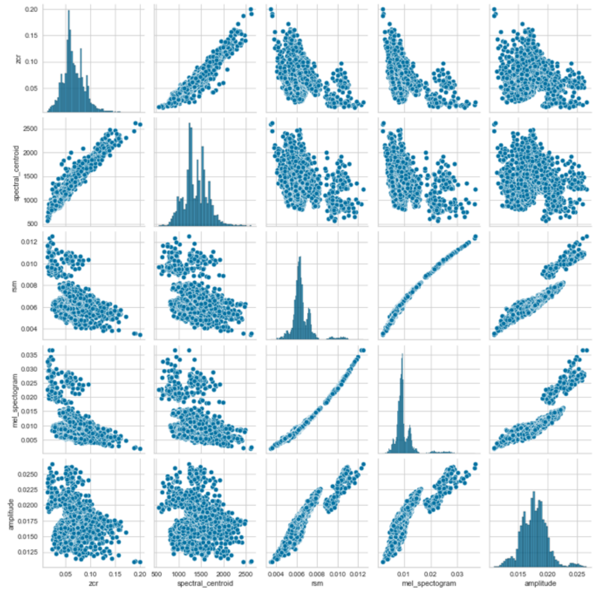
We have listed above the features that we can use before starting to create the model. According to these inferences, it has been determined that the most important features that we will use for voice discrimination are **Zero crossing rate, amplitude envelope, RMS(Remote Mean Square), spectral centroid and mel-spectrogram**

We have obtained the ready-made version of these data from the repository whose address is in the reference section below. Thus, we have saved time to be spent. We thank the owner of the repository for sharing the datasets with us.

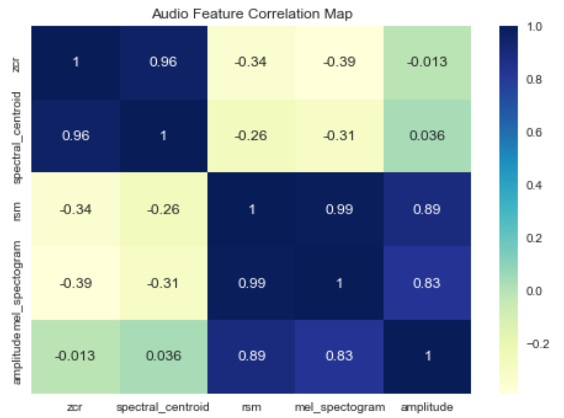
After the data sets are cleaned and checked, we can see the first five rows below. The values taken show the means.

	zcr_mean	spec_cent_mean	rms_mean	melspec_mean	amp_mean
0	0.109654	2105.182951	0.005728	0.007248	0.015913
1	0.075940	1524.399576	0.005831	0.007658	0.016215
2	0.037438	1081.963248	0.006127	0.008834	0.015449
3	0.055022	1220.102213	0.005838	0.008052	0.015978
4	0.051522	1262.663528	0.006116	0.008755	0.016088

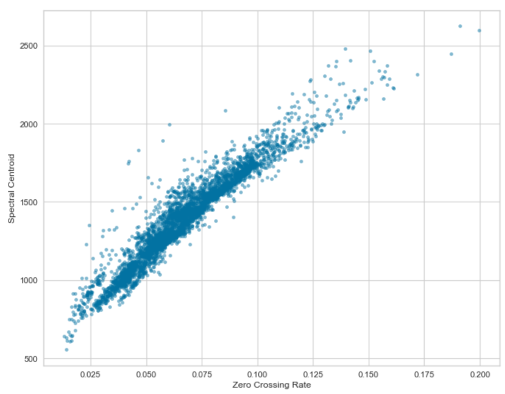
In order to understand the connection in the data, the pairplot() function of the seaborn library is used and the result is as follows. When we look at the relationships between them, we observe the most obvious result between the zcr and spectral centroids. Therefore, it was decided to go through these two values while grouping.



In order to see the accuracy of the graphs, again with the help of the seaborn library's hetmap() function, the values that emerged with another graph were observed, and as a result, we saw the harmony between the two with a value of 0.96.



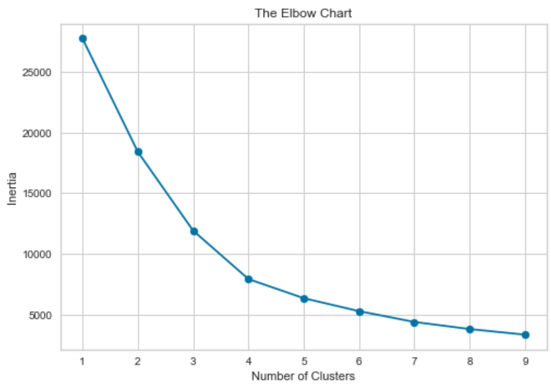
In order to see the harmony between these two, a new graph was created with the help of the scatter() function. The graph is below.



K-Means of the Clustering model was chosen as the model. When we look at clustering, clustering is the separation of data showing similar characteristics in a data set into groups. Within the same cluster, the similarities are high, and the similarities between the clusters are low. It is a form of Unsupervised Learning.

While scaling the data set, the StandardScaler() function is used and it is fitted with the fit_transform() function.

In k-means, it is one of the most popular to use the elbow method when choosing k, that is, the number of clustering. While creating this model, the elbow method was used.



It is observed in the graph that the number of clusters is 3 when installing KMeans, which is the most appropriate. Thus, we understand that this model will make a correct grouping for the desired format. The normal, abnormal and transition data requested from us must be clustered correctly.

After clustering, the data is marked between [0,1,2].

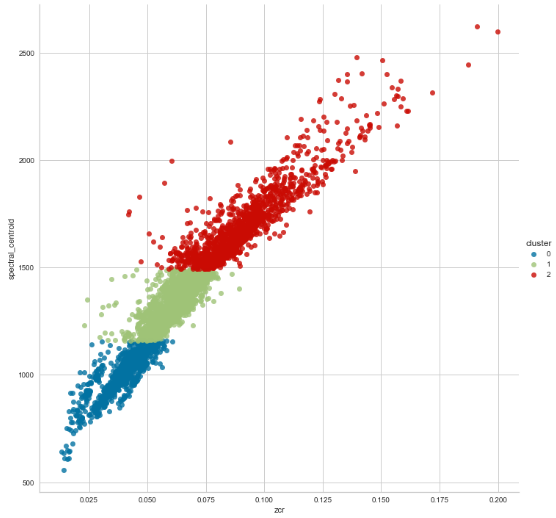
As a result:

Total number of data 5550

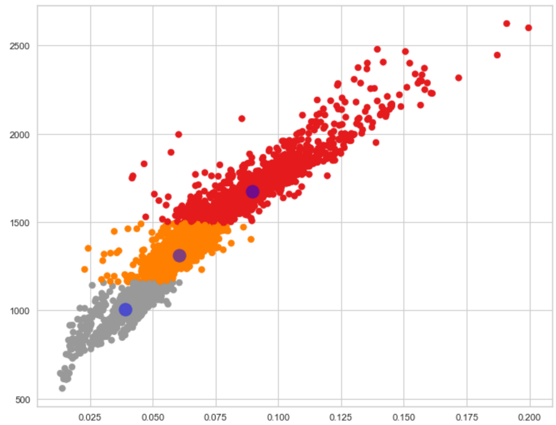
Cluster 0 : 0.17909909909909991

Cluster 1 : 0.4772972972972973

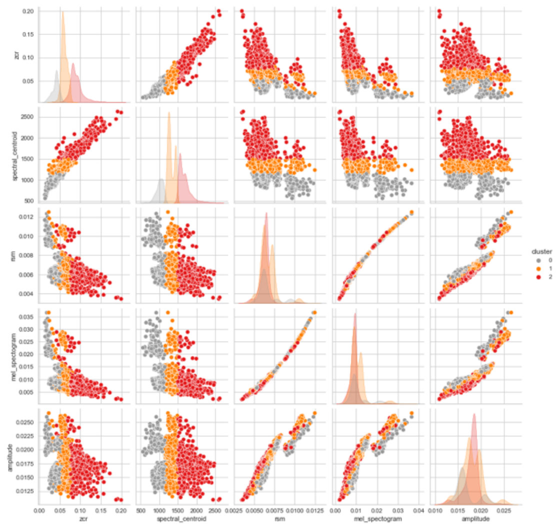
Cluster 2 : 0.3436036036036036



The graph that comes out as a result of the clustering process looks like the above.

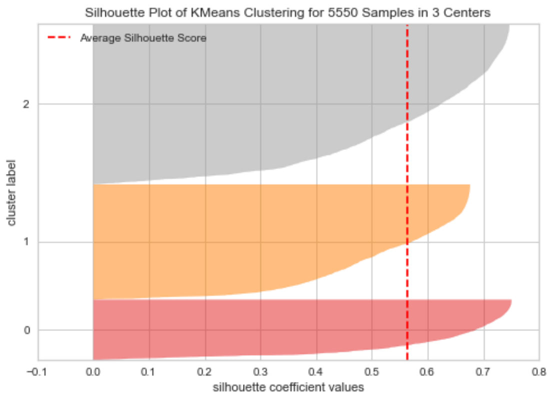


While clustering, the centroids are marked in blue to ensure that the clustering is done correctly.



The graph is as above as a result of the clustering process based on the relationships of all the data with each other.

When we look at the metrics that measure the accuracy of the model, an accuracy of approximately 60% is observed.



In order to choose the most suitable value for 3 clusters, we have handled the Silhouette analysis made in the graphic above.

A value of 3 clusters seems to be optimal. The silhouette score for each cluster is above the average silhouette scores. Also, there is not much difference between fluctuations in size. Only cluster 0 is smaller than the other two, which tells us that it is a transition selection. Therefore, the optimal number of clusters can be chosen as 3.

As a result;

For this research it is taken fan machine as a basement.The silhouette score considers the intracluster distance (a) between the given sample and other data points in the same cluster, and the intercluster distance (b) between the sample and the next closest cluster.The silhouette score is within the range of [-1, 1]. A silhouette score of 1 means the clusters are very densely and beautifully separated. A score of 0 means the clusters overlap. A score less than 0 means that the data for the clusters may be false/false. We see that the Silhouette score that our graph has shown us is approximately 60%, and this shows us that the model is not fully compatible.

other models can be used for this classification.

The similar model of the project has been prepared according to other machine types and is included in the Machine Condition Monitoring with Clustering repository.

```
Machine-condition-monitoring-with-clustering
|
|
|__
|   Datasets                   : include .csv files
|
|__ __ valve.csv
|
|__ __ fan.csv
|
|__ __ pump.csv
|
|__ __ slider.csv
|
|
|__ README.md                  : explains the project
|
|
|__ fan_clustering.ipynb       : k-mean clustering for machine of fan
|
|__ pump_clustering.ipynb      : k-mean clustering for machine of pump
|
|__ slider_clustering.ipynb    : k-mean clustering for machine of slider
|
|__ valve_clustering.ipynb     : k-mean clustering for machine of valve
|
```

References :

1. https://zenodo.org/record/3384388

2. Purohit, Harsh, et al. "MIMII Dataset: Sound dataset for malfunctioning industrial machine investigation and inspection." *arXiv preprint arXiv:1909.09347* (2019).

3. https://devopedia.org/audio-feature-extraction

4.https://github.com/UjjwalKandel2000/Machine-conditions-monitoring/tree/main/Dataset

5.https://github.com/yusufakcakaya/Machine-condition-monitoring-with-clustering.git