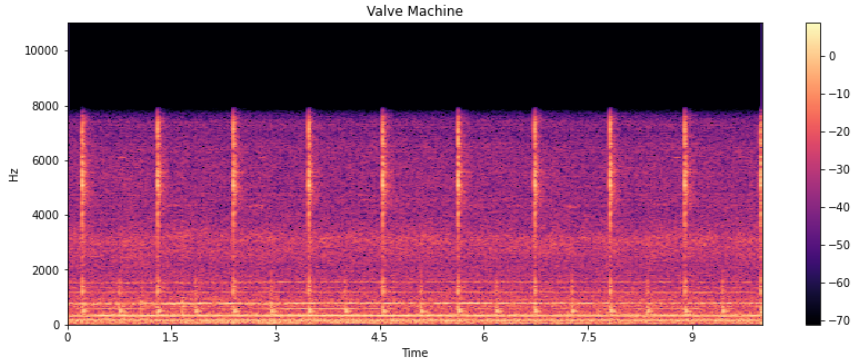
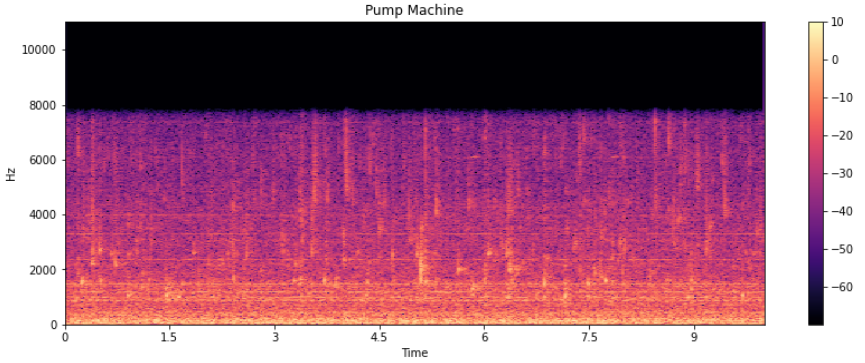
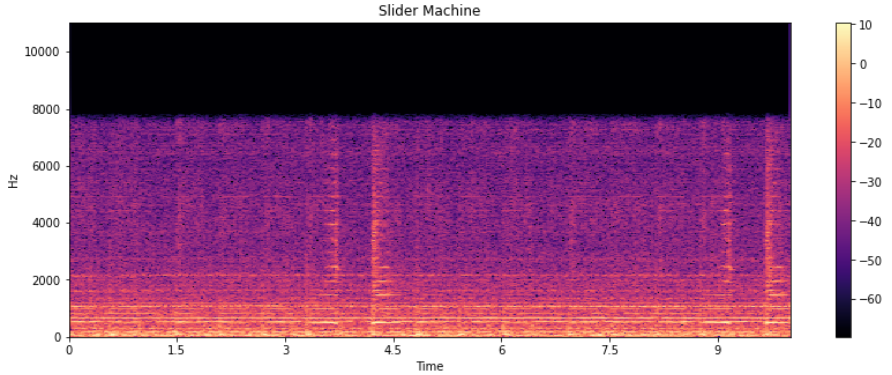
ARTIFICIAL INTELLIGENCE BOOTCAMP - BeCode

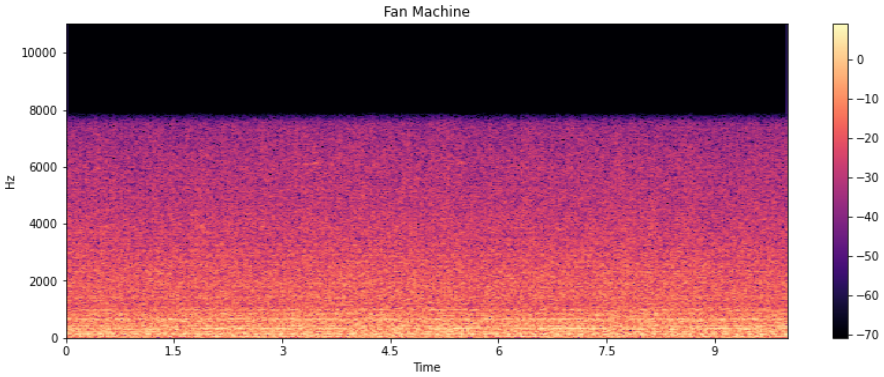
Sound Anomaly Detection in Machines

CLassification of Sounds



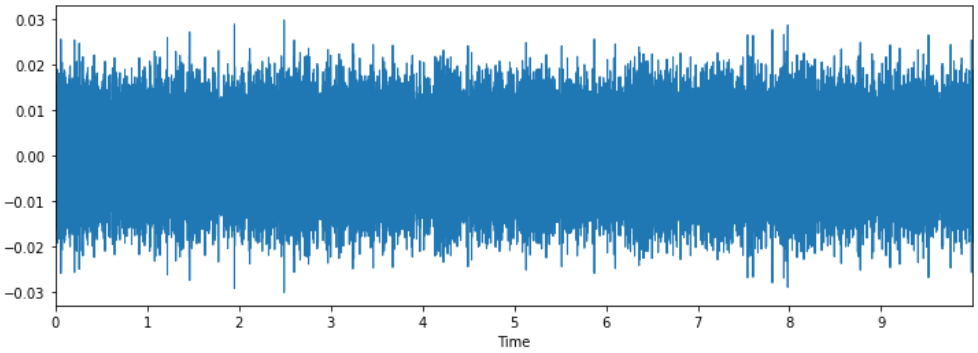






Introduction

As you can see in the above images all sound waves when computed look different. Therefore creating a model for each machine would be advisable. Thus, by creating a classification model for each machine we can classify the sound waves in a better way as normal or abnormal sound.



This is how a sound wave looks like, and we have extracted the features of a sound wave and used that for our analysis of classification models. You can also access the dataset that we have included in our repository. (https://github.com/UjjwalKandel2000/Machine-conditions-monitoring)

Each sound wave has 8 channels which means that each sound was recorded using 8 microphones.

Classification

Classification of machines - classify the sounds as normal or abnormal based on the extracted features.

We started by creating different models for each machine and checking the accuracy for each machine. Based on our analysis we found out that Random FOrest classifier - a classification model proved to show good results when classifying the data as normal or abnormal sounds.

- **Fan machine**

For the fan machine we computed a model and got an accuracy score of 93.34% .

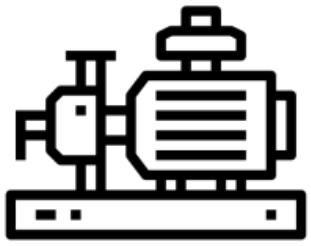
Accuracy score - Accuracy is a pretty simple metric. It is the ratio of the number of correct predictions to the total number of input samples. So based on 0db data we were able to get this result which is pretty impressive to be able to classify sound waves as normal or abnormal .

This below score is calculated when we pass the input samples randomly in different batches. The score varies between 89 - 96% which means the average score is somewhere at 93%.

96.31% , 94.29% , 91.80% , 89.77% , 91.80%

Using the random forest classifier model we can predict if the fan machine is giving normal or abnormal sounds.

- **Pump machine**



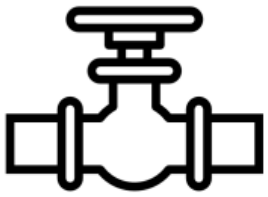
For the pump machine we computed a model and got an accuracy score of 93.34% .

Accuracy score - Accuracy is a pretty simple metric. It is the ratio of the number of correct predictions to the total number of input samples. So based on the 0db dataset we were able to get this result which is pretty impressive to be able to classify sound waves as normal or abnormal. This below score is calculated when we pass the input samples randomly in different batches. The score varies between 89 - 96% which means the average score is somewhere at 93%.

96.31% , 94.29% , 91.80% , 89.77% , 91.80%

Using the random forest classifier model we can predict if the pump machine is giving normal or abnormal sounds.

- **Valve machine**



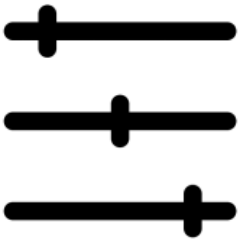
The valve machine when computed got an accuracy score of **93.68%**.

This result is computed for the **0db** dataset. This below score is calculated when we pass the input samples randomly in different batches. The score varies between 87 - 96% which means the average score is somewhere at 91.36%.

95.68% , 94.36% , 95.08% , 91.85% , 87.05%

Using the random forest classifier model we can predict if the valve machine is giving normal or abnormal sounds.

- **Slider machine**



The slider machine when computed got an accuracy score of **94.06%**.

This result is computed for the **0db** dataset. This below score is calculated when we pass the input samples randomly in different batches. The score varies between 89 - 98% which means the average score is somewhere at 94%.

97.19% , 98.17% , 92.06% , 89.57% , 94.05%

Using the random forest classifier model we can predict if the slider machine is giving normal or abnormal sounds.

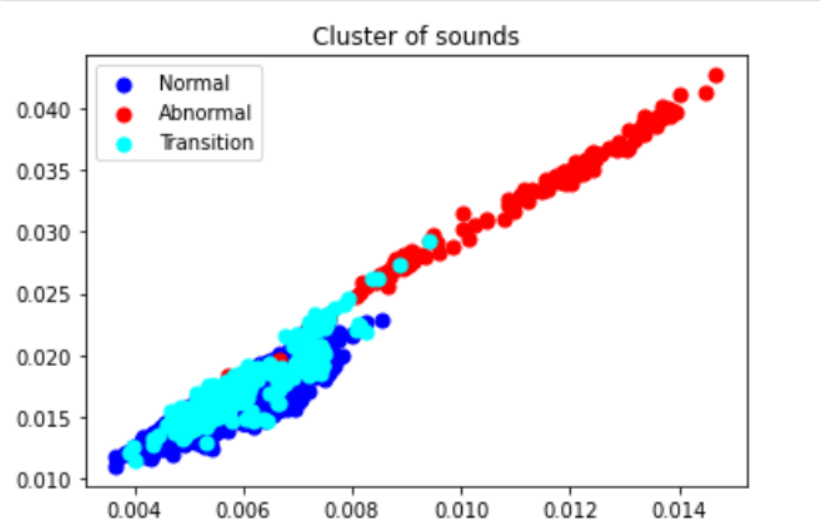
Conclusion

Based on our classification models we can see that our models perform better under the Random Forest Classifier algorithm. The slider machine seems to have performed the best among all the machines and relying on the model for classification of sound signals for the slider machine would give good results.

Each machine shows an average accuracy of 91-93% which can be helpful in classification of the sound waves as normal or abnormal. If we worked on the features and models, we would be able to give better results in the upcoming collaboration with your company.

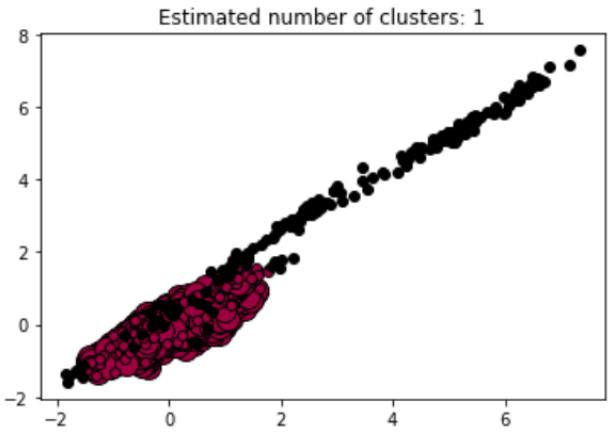
Clustering

- **Pump machine**

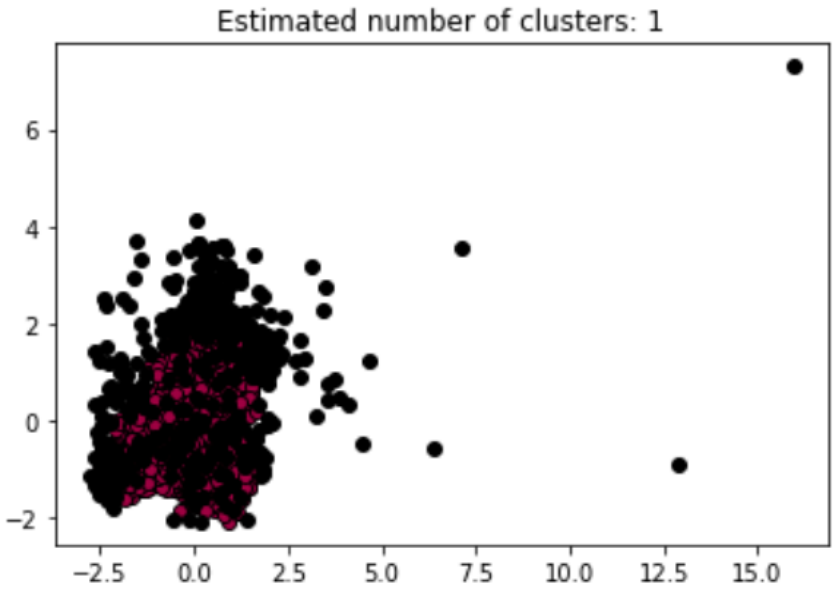


In the above image you can see how the sounds are being clustered as normal abnormal and in transition state.

1. With K-means clustering algorithm the rand_score is 85.29%
2. DBSCAN algorithm the rand score is 85.41% for the pump machine model. (image below) - The abnormal sounds are marked as black.



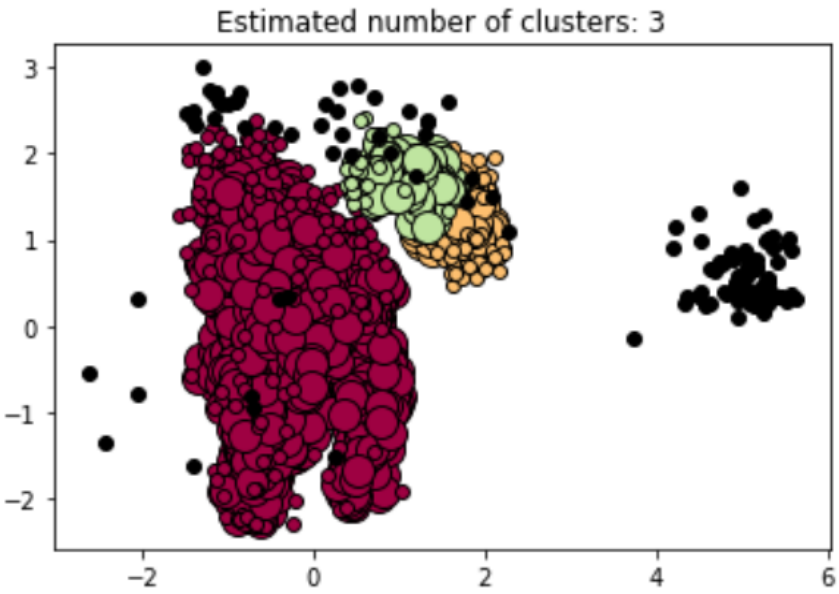
- **Valve machine**



In the above image you can see that I have used the DBSCAN algorithm which plots the abnormal sounds as black dots and normal are clustered into one. The k-means algorithms seems to give really low results for this machine.

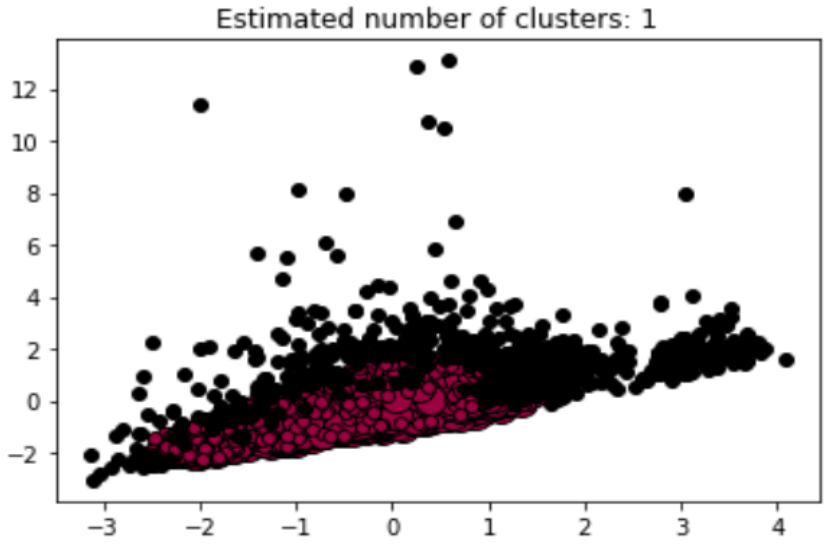
1. With the K-means clustering algorithm the rand_score is around 50%.
2. DBSCAN algorithm the rand score is 79.32% for the valve machine model.

- **Slider machine**



1. With the K-means clustering algorithm the rand_score is 57.11%.
2. DBSCAN algorithm the rand score is 82.40% for the pump machine model.

- **Fan machine**



The fan machine was difficult to cluster. In the above image you can see the noise(abnormal sound) being placed everywhere.

With DBSCAN the score that I achieved for this machine was around 65%.

Conclusion

Based on our clustering models we can see that our models perform better under the DBSCAN algorithm. The pump and slider machines seem to have performed the best among all the machines and relying on the model for clustering of sound signals for the machines would give good results.

Each machine shows an average score of 82-85% which can be helpful in clustering the sound waves as normal or abnormal. If we worked on the features and models, we would be able to give better results in the upcoming collaboration with your company.

You can refer to my github repositories for the code.

https://github.com/Arfameher/machine_conditions-unsupervised

https://github.com/UjjwalKandel2000/Machine-conditions-monitoring