****

EE257: Machine Learning for Electrical Engineers

**Design of an E-Nose**

Course Project- Spring 2019

Professor: Birsen Sirkeci

Submitted By:

Mridul Chopra-012451413

(With Vipul Gargav 012425816)

**Table of Contents:**

1. Dataset Gathering Process 3
2. Attribute Information 3
3. Dataset Description 4
4. Data Set Visualization 4
5. Data set Cleaning 7
6. Related Work 8
7. Feature Extraction 8
8. Model Development 8
9. Fine tune model and Optimize feature set 9
10. Performance 9
11. Novel ways to resolve the sensor drift problem 9
12. Results and Conclusion 10
13. Extra Credit 11
14. Reference 11
15. Appendix I (Table of Figures) 11
16. Appendix II (Table of Tables) 11

**Dataset Gathering Process:**

A chemical detection platform was used that comprised of 8 chemo-resistive gas sensors (TGS2611, TGS2612, TGS2610, TGS2600, TGS2602 TGS2620) that were bared to turbulent gas mixtures generated naturally in a wind tunnel. Finally, the acquired time series of the gas sensors was provided. The experimental setup was designed to test these gas sensors in realistic environments. Conventionally, the detection system for chemo-resistive gas sensors include a gas chamber to control the sample air flow and lessen turmoil. In its place, they used a wind tunnel with 2 impartial gas sources that produce two gas plumes. The gases get naturally mixed along a tempestuous flow and imitate the concentration variations witnessed in natural environments. Therefore, the gas sensors capture the spatial temporal information contained in the gas plumes.

The chemical detection platform consisted of 8 MOX gas sensors that generate a time contingent multivariate reaction to the different gas stimuli. The utilized sensors were TGS2611, TGS2612, TGS2610, TGS2600, TGS2602 TGS2620 commercially made available by Figaro. The functioning temperature of the sensors was controlled by the heater built-in with a constant voltage of 5 volts. The detection platform also contains Relative Humidity and Temperature sensors. The responses from generated sensors were learnt at a sampling rate of 20ms for the whole extent of the experiment.

The wind tunnel is a 2.5 m x 1.2 m x 0.4 m facility in an open environment with two autonomous gas sources (source 1 and source 2). Each source was controlled impartially to release selected gases at different flow rates that produced different concentration levels in the position of sensors. The wind generator built a turbulent flow that incessantly displaced the introduced volatiles towards exhaust outlet.

Finally, the detection unit was exposed to Ethylene with Methane or Carbon Monoxide. The mixtures were originated releasing Ethylene at source1 and releasing Methane / Carbon Monoxide at source2. Each volatile was released at four different flows (zero z, low l, medium m, high h) providing up to 15 mixtures of Ethylene with Carbon Monoxide (h+h, h+m, h+l, h+z, z+m, z+l, z+h) and 15 mixtures of Ethylene with Methane likewise. Each was repeated 6 times and therefore the complete dataset is composed of 180 measurements performed in random order.

Each measurement with a total of 300 seconds was performed as: Initially no gas was released, followed by clean air along the tunnel. After 60 seconds both the gas sources started to release at a specified flow rate. The total duration of gas release is 180 seconds. Finally, the system developed the recovery for another 60 seconds.

**Attribute Information:**

The dataset is presented in 180 text files, where each file related to a different measurement. The filenames are represented as following: the first 3 letters are a local identifier, next 5-8 are concentration level of Ethylene (zero, low, medium, high). And the last 4 characters is the gas (Methane or Carbon Monoxide) and its concentration. Each file is the acquired time series: Time (s), Temperature in degree Celsius, Relative Humidity and reading of 8 gas sensors.

The sensor array was exposed to mixtures of ethylene with carbon monoxide or methane. The gases were released at four different rates to induce different concentration levels near the sensor array and each was released 6 times, for a total of 180 measurement.

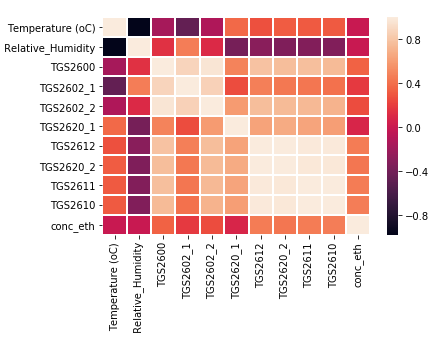
**Dataset Description:** We used the Gas sensor array exposed to turbulent gas mixtures dataset from the UCI repository. The project is a classification problem, so we classified the data into NULL, LOW, MEDIUM, HIGH corresponding to the particular gas. Since, our data had 180 different files with different concentration of Ethylene, Methane and Carbon Monoxide, we concatenated a few files with all different concentration of different gases. The data set contained 11 features and we added 5 more columns to the dataset. All the features were float type.

1. Ethane [0-Null, 1-Low, 2-Medium and 3-High]
2. Carbon Monoxide [-1- Absent, 0-Null, 1-Low, 2-Medium and 3-High]
3. Methane [-1- Absent, 0-Null, 1-Low, 2-Medium and 3-High]
4. Mean Concentration of Ethylene [31ppm-Low, 46ppm-Medium, 96ppm-High]
5. Mean Concentration of Carbon Monoxide [270ppm-Low, 397ppm-Medium, 460ppm-High]
6. Mean Concentration of Methane [51ppm-Low, 115ppm-Medium, 131ppm-High]

Then we split the data into Training and Test in 80-20 ratio. The training data has 71256 data points and test has 17814 data points.

**Data Set Visualization:** Data visualization is a technique of using an array of visuals in a specific context to help understand the large amount of data and draw conclusions. So, here we used libraries like matplotlib and seaborn to do simple visualizations. We also performed Exploratory Data Analysis using boxplot, bar plot, pair plot, count plot and heatmap for correlation. While visualizing the data, we got to know what variables are dependent on each other with having a positive or negative correlation.

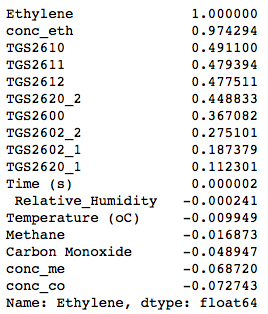
1. **Heat Map:** It is a color coded 2-D graph that gives negative and positive values. It is used to represent the dependencies and behavior of all the attributes with each other.



**Figure 1: Heatmap for correlation**

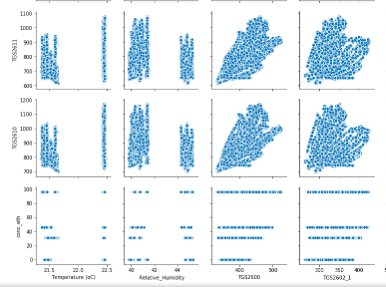
In a heatmap, -1 is being negatively correlated and +1 being positively related. 0 states that there is very less correlation with the corresponding variables. The above graph shows that all the sensors are highly correlated. We can see from the Figure as well as the table above that Concentration of Ethylene is not correlated with Temperature in degree Celsius and Relative Humidity.

1. **Correlation Matrix:** The tabular format specifies the correlation of one attribute with all the other attributes in the sensor array dataset. The value is a float between -1 and +1, and 0 in case of no correlation between the attributes. The correlation matrix gives a clear picture of the least important attributes which can be eventually dropped, and the performance of the machine learning algorithm can be improved. All the 8 sensors are very correlated to Ethylene. We can drop a few attributes to perform data cleaning and enhance accuracy and precision.

****

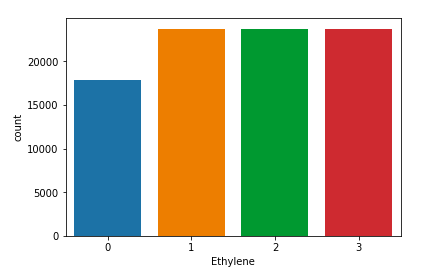
**Table 1: Correlation of ‘Concentration of Ethylene’ with all the features**

1. **Pair Plot:** This is a graphical representation of specific relations in pairs. We have 11 input variables and the small part of the pair plot shown below depicts the relationship in pairs. Example: in the figure below, we can see the relationship of TGS2611, TGS2610 sensor and Temperature (s), Relative Humidity and concentration of ethylene.

****

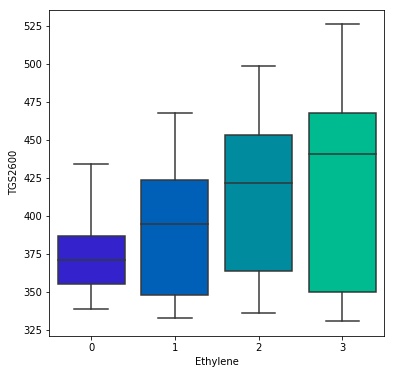
**Figure 2: Pair Plot SNS**

1. **Count Plot:** This plot gives the information about the total count of a categorical attribute and one numerical variable in a bar graph. The figure below shows the total count of level of concentration of Ethylene. (Null-0, Low-1, Medium-2, High-3). The dependent variable is the number of instances of each instance of the independent variable.

****

**Figure3: Count Plot SNS**

1. **Box Plot:** Python’s pandas and seaborn have plotting capabilities. Box plot gives a representation of how the values are distributed in a dataset, shows the distribution and summary of quantitative data. A box plot contains minimum, maximum, median, first quartile and third quartile. The figure below shows a box plot of level of concentration in Ethylene and a sensor TGS2600 distribution. The median of Ethylene concentration is greater in this sensor.

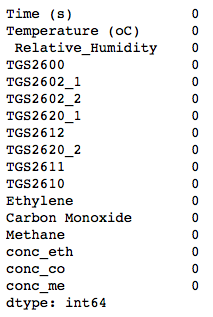


**Figure 4: Box Plot SNS**

Therefore, by visualizing the data we found out that features like {Temperature and Relative Humidity} are not important for our project objective. So, we can perform some Dataset Cleaning techniques for not considering data that is not required.

**Data set Cleaning:** Machine Learning algorithms predict high precision and accuracy because the data used is full of garbage and missing values.This process is used to get rid of garbage value, removing features that are not useful in predicting and fill up the missing values with either of mean or median of the particular feature. We can use two steps for the process of Data Cleaning:

1. Checking missing values: We checked for missing values or NAN values in the dataset and found nothing. Had there been missing values, we could have replaced them by the mean of the particular feature to fill up the dataset.



**Table 2: Missing values found in Dataset – None**

1. Feature Selection: In this process, we manually select those features which do not contribute to the prediction output and omit them. So, we removed features like Temperature and Relative Humidity which had very low relation to the dataset. Having irrelevant features in the dataset can decrease the accuracy and make the model learn irrelevant features.
2. Outliers: The boxplot could also be used for visualizing outliers but most features in the gas sensor array dataset did not have outliers. By looking at the box plot we can make out if any points lie outside of the quartile region, they can be considered as outliers.

Another method of data cleaning can be converting text and categorical values into numeri values, but there were no categorical values in the data set which could be converted.

**Related Work:** In the paper, Chemical discrimination in turbulent gas mixtures with MOX sensors validated by gas chromatography mass spectrometry, the authors included a gas chambers to control the air flow and minimize turbulence, such kind of gas concentration fluctuations cannot be reproduced in natural environments and it destroys the spatio temporal information too. The main problem of the natural environment is that when a volatile is emitted, the molecules travel in the direction of fluid flow forming a patchy plume in downstream direction. Therefore, the air direction and intensity change in time along with the irregular structures of the plumes. This paper sought to develop a chemical detection system that reduces fluctuations, variations in temperature, humidity and flow. Binary mixtures of three gas at different concentration levels, a standalone Metal Oxide gas sensor array and a Gas chromatography mass spectrometry system is used to identify the compound by photo ionization detector. The authors built an Support Vector Machine classifier to detect ethylene in a turbulent environment composed of Methane and Carbon Monoxide. The aim of the experiment was 2-fold: to determine whether it is possible to build a classifier to discriminate ethylene in dynamic mixtures and to analyze the power of different concentration levels. The proposed classification method allowed effective discrimination of ethylene on dynamic mixtures and showed the importance of including low concentration measurements in the dataset.

**Feature Extraction:** This is a process of combing existing features to produce a more useful feature also known as Dimensionality Reduction Algorithm. The figure below shows the feature extraction method used to identify the features that did not provide comparatively better accuracy.



**Figure 5: Feature Extraction**

After dropping the uncorrelated features, the Ethylene concentration level performance improved significantly. And there was no more need of dropping any more features.

**Model Development:** The project is focused on Classification problem, so we used four different classification models to measure the precision and accuracy. The fine tuning and optimization process improved the accuracy a great deal. The models used in this project are:

1. Logistic Regression
2. Decision Tree
3. LDA and QDA
4. Random Forest

The table below gives the accuracy for classification methods used before fine tuning and optimization using K-Fold cross validation using K=10, while the table 3 provided the test performance over trained data.

|  |  |
| --- | --- |
| Model | Accuracy(%) |
| Logistic Regression | 84 |
| Decission tree | 80 (max\_depth=2) |
| lda | 54 |
| qda | 20 |
| Random Forest | 95 (max\_depth=3) |

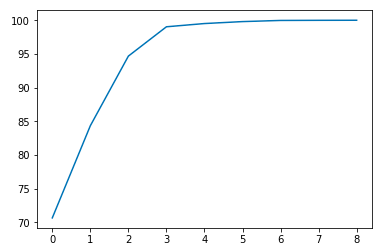
**Table 3: Model Development**

|  |  |
| --- | --- |
| Model | Precision(%) |
| Logistic Regression | 84 |
| Decission tree | 69 |
| lda | 56 |
| qda | 20 |
| Random Forest | 99 |

**Table 4: Model Precision**

The reason for specifying Random forest at last is that Random forest is not best suited for our dataset as it is overfitting and so it is not a good Machine learning algorithm for gas sensor dataset.

**Fine tune model and Optimize feature set:** After obtaining the results from all the machine learning models, we observed that Logistic Regression and Decision Tree are the best algorithms suited for this dataset. Fine tuning the parameters might result in a better and efficient algorithm. So, we used Principal Component Analysis that compared the ROC curve and gives the best fit where the curve starts. There were total 9 input variables and since the curve bent at 3, the principal component(n) can be chosen as 3.



**Figure 6: Principal Component Analysis Graph**

**Performance:** The performance analysis before and after the fine tuning of the data was done. Prior to fine tuning the data random forest shows 89% accuracy but after fine tuning it showed 99% accuracy. Sometimes, the dimensionality reduction doesn’t work really well as the precision and recall score are not good as compared to the precision before fine tuning. The figure 7 displays the performance of each model.

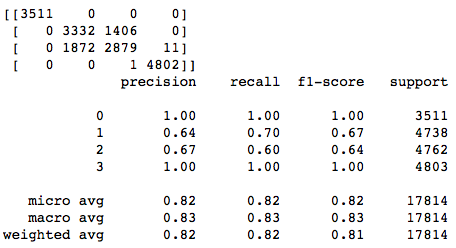
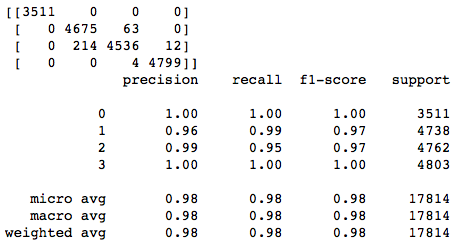
**Novel ways to resolve the sensor drift problem:** The sensor drift problem arises when due to passage of time the sensors behave abnormally and do not give out the correct measured value. As in this case, the first 60 seconds turbines send out clean air, then for the next 180 seconds it gives out Ethylene & Carbon Monoxide or Ethylene & Methane. After 240 seconds (60+180), the turbines stop giving out gases and the sensors may get worn out and possibly give false values of the gases. This mainly happens due to the ageing of the gas sensors when they are exposed to a high temperature, pressure or humidity for a long time. The accuracy for Logistic Regression came out to be 80% and the accuracy for Random forest with a maximum depth of 4 came out to be 99%.

**Results & Conclusions:** The objective of the project was to classify the level of concentration of Ethylene. We used several machine learning algorithms and techniques to evaluate the accuracy and performance of the models and provide a solution for sensor drift problem. The test accuracy increased from the training accuracy after fine tuning and optimization for Random Forest, Decision Tree, QDA and LDA significantly.

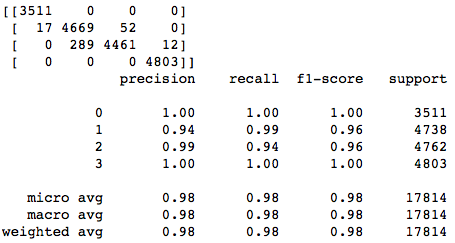
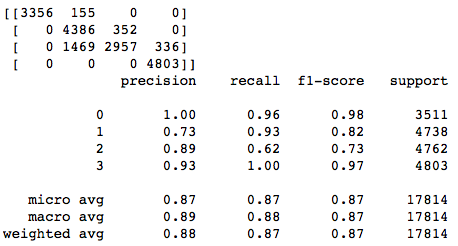
|  |  |  |
| --- | --- | --- |
| Machine Learning Models: | Training Accuracy: | Test Accuracy After Fine Tuning: |
| Logistic Regression | 84% | 81% |
| Decision Tree | 80% | 99% |
| Random Forest | 89% | 99% |
| LDA | 54% | 87% |
| QDA | 20% | 87% |

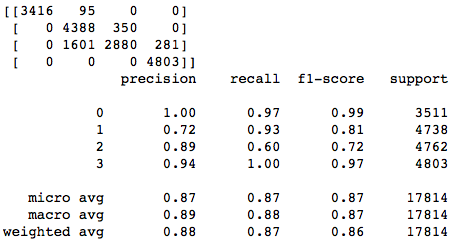
**Table 5: Training and Test accuracy after Fine Tuning**

**Logistic Regression Random Forest**

**Decision Tree LDA**

**QDA  
**

**Figure 7: Precision, Recall and F1-score of all Machine Learning models used.**

**Extra Credit:** In the extra credit problem, we sought to predict if a gas exists or not. The dataset composed of 180 files but to save computational power, we used 30 files each with a different concentration of different gas. We used 0 for a gas present and 1 for gas absent (Carbon Monoxide and Methane). Since, this problem is a binary classifier, the decision tree classifier algorithm performed the best. The presence of gas is highly correlated with the gas sensors only and the input features contained 8 sensors. The decision tree eventually over-fitted but after changing the maximum depth level to 8 the accuracy came out to be 99%.

**References:**

[1] [Jordi Fonollosa](https://www.ncbi.nlm.nih.gov/pubmed/?term=Fonollosa%20J%5BAuthor%5D&cauthor=true&cauthor_uid=25325339), [Irene Rodríguez-Luján](https://www.ncbi.nlm.nih.gov/pubmed/?term=Rodr%26%23x000ed%3Bguez-Luj%26%23x000e1%3Bn%20I%5BAuthor%5D&cauthor=true&cauthor_uid=25325339),[Marco Trincavelli](https://www.ncbi.nlm.nih.gov/pubmed/?term=Trincavelli%20M%5BAuthor%5D&cauthor=true&cauthor_uid=25325339), [Alexander Vergara](https://www.ncbi.nlm.nih.gov/pubmed/?term=Vergara%20A%5BAuthor%5D&cauthor=true&cauthor_uid=25325339) and [Ramón Huerta](https://www.ncbi.nlm.nih.gov/pubmed/?term=Huerta%20R%5BAuthor%5D&cauthor=true&cauthor_uid=25325339), “<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4239943/>”, October 2014

[2]<https://archive.ics.uci.edu/ml/datasets/Gas+sensor+array+exposed+to+turbulent+gas+mixtures>

**APPENDIX I**

**Table of Figures:**

|  |  |
| --- | --- |
| Figure 1: | Heatmap for correlation |
| Figure 2: | Pair Plot SNS |
| Figure 3: | Count Plot SNS |
| Figure 4: | Box Plot SNS |
| Figure 5: | Feature Extraction |
| Figure 6: | Principal Component Analysis Graph |
| Figure 7: | Precision Recall and F1-score of all Machine Learning models used. |

**APPENDIX II**

**Table of Tables:**

|  |  |
| --- | --- |
| Table 1: | Correlation of ‘Concentration of Ethylene’ with all the features |
| Table 2: | Missing values found in Dataset – None |
| Table 3: | Model Development |
| Table 4: | Model Precision |
| Table 5: | Training and Test accuracy after Fine Tuning |