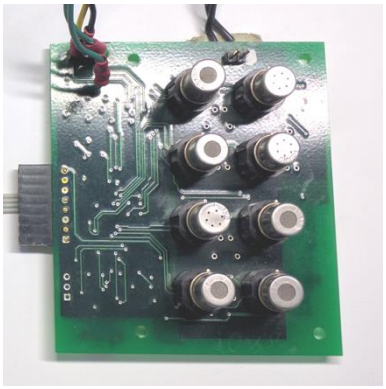
DISTINGUISHING BETWEEN BA-NANA,WINE AND BACKGROUND BASED ON MOS SENSOR OUTPUT

IIITB Internet of Things Final Project Report



AUGUST 6, 2020

AKHILESH KUMAR(SMT2018002), GAURAV JAIN(SMT2018007) , PD MOHGAN(SMT2017008)

TABLE OF CONTENTS

Introduction

A method for computing correlation of chemical sensor readings from the effects of environmental humidity and temperature variations is proposed. The goal is to distinguish between wine, banana, and baseline and improve the accuracy of discriminating gas measurements for continuous monitoring by processing data from simultaneous readings of environmental humidity and temperature.

The goal is to distinguish between Banana, Wine and Baseline from the data given by training the model using part of the data and predict for the remaining part of the data and evaluate the method used using the accuracy score. The challenge here is to choose which classifier to apply and how much percent of data to be used for training.

System Model

Hardware setup

This section is skipped since the data set is used from the experiment conducted and provided in Internet.

Data Set

Here the data set used is taken from Machine Learning repository of Center for Machine Learning and Intelligence Systems. The link is provided in the reference section.

The data set consists of the recordings of gas sensor array of 8MOX gas sensors, temperature and a humidity sensor. The sensor readings are taken by exposing them to Banana, Wine and Background activity without the presence of Banana or Wine. The duration of expose varies from 7 minutes to 2 hours with an average duration of 42 minutes. The data set is a time series data with 36 instances of Wine, 33 instances of Banana and 31 instances of Background Activity. Also, the data contains the values of one hour of background activity prior and after the exposing of sensors to Banana and Wine.

This data set is composed of two files: HT_sensor_dataset.dat (zipped), where the actual time series data are stored, and the HT_Sensor_metadata.dat, where metadata is stored. Each instance is uniquely identified by an id in both files. Thus, metadata for a particular induction can be easily found by matching columns id from each file. The time series data stores in HT_Sensor_dataset.dat is composed of 100 instances of time series, each being exposed to either Banana or Wine or background activity. On total, there are 919438 points. For each induction, the time when the stimulus was presented is set to zero. For the actual time, see column t0 of the metadata file.

Attribute Information:

The metadata stored in file HT_Sensor_metadata.dat is divided in the following columns:

- id: identification of the induction, to be matched with id in file HT_Sensor_dataset.dat
- date: day, month and year when this induction was recorded
- class: what was used to generate this induction (wine, banana or background)
- t0: time in hours in which the induction started (represents the time zero in file HT_Sensor_dataset.dat)
- dt: interval that this induction lasted

The file HT_Sensor_dataset.dat, each column has a title according to the following

- id: identification of the induction, to be matched with id in file HT_Sensor_metadata.dat
- time: time in hours, where zero is the start of the induction
- R1 to R8: value of each of the 8 MOX sensors resistance at that time
- Temp.: measurement of temperature in Celsius at that time
- Humidity: measurement of humidity in percent at that time

Describe the analysis conducted

The metadata and the data set given are combined or joined into a single table using inner join. The 8-sensor data, humidity and temperature sensor are plotted and as shown in Figure 1. Data correlation is computed for data set and also the table for correlation > 0.98 is shown in Table 1.

	time	R1	R2	R3	R4	R5	R6	R7	R8	Temp.	Humidity	id	dt
time	True	False	False	False	False	False	False	False	False	False	False	False	False
R1	False	True	False	False	False	False	False	False	False	False	False	False	False
R2	False	False	True	False	False	False	False	False	False	False	False	False	False
R3	False	False	False	True	False	False	False	False	False	False	False	False	False
R4	False	False	False	False	True	False	False	False	False	False	False	False	False
R5	False	False	False	False	False	True	False	False	False	False	False	False	False
R6	False	False	False	False	False	False	True	False	False	False	False	False	False
R7	False	False	False	False	False	False	False	True	False	False	False	False	False
R8	False	False	False	False	False	False	False	False	True	False	False	False	False
Temp.	False	False	False	False	False	False	False	False	False	True	False	False	False
Humidity	False	False	False	False	False	False	False	False	False	False	True	False	False
id	False	False	False	False	False	False	False	False	False	False	False	True	False
dt	False	False	False	False	False	False	False	False	False	False	False	False	True

Table 1: Correlation of Data Set

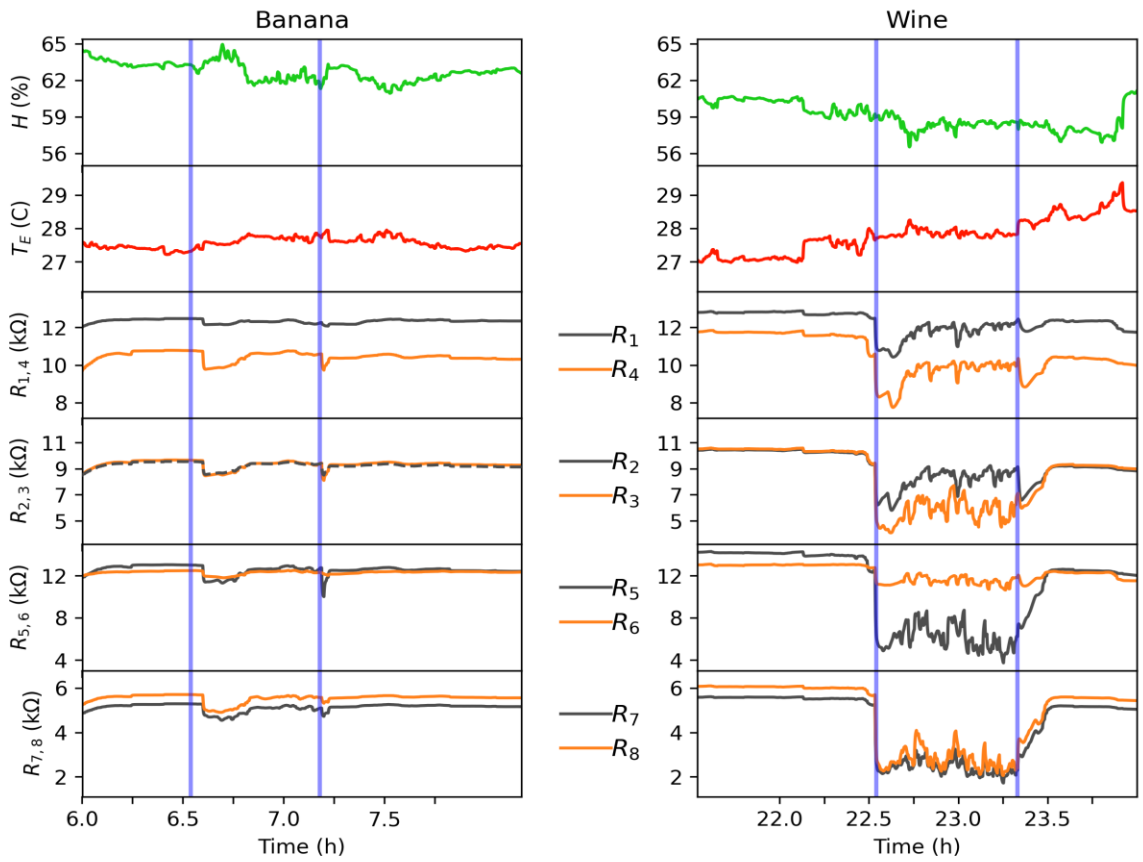


Figure 1: Plot Representation of R1~R8, Temperature & Humidity for Banana & Wine

We have tried to correlate data from all sensors with each other based on one such induction instance from banana and wine respectively. The correlation between sensor output based on the induction from wine or banana can be seen in the Figure 1.

Approach

The data collected from UCI website is huge and consists of around 9 lakhs data points for over 100 inductions. If we do a training and prediction based on the total data points it is consuming too much time. So, in order to get good prediction results with less data we have pruned the data such that each induction is having at least 3000 points. This makes around 3 Lakhs data points over all. We have used 70% of the pruned data for training the model classifiers and the remaining 30% for the testing. We trained the prediction model on the following Classifiers.

Logistic Regression Classifier

Logistic Regression is first applied to train the data and predict as well using the test data. Data need to be regularized for applying Logistic regression to prevent over fitting of the model.

Learning the parameters of a prediction function and testing it on the same data is a methodological mistake: a model that would just repeat the labels of the samples that it has just seen would have a perfect score but would fail to predict anything useful on yet-unseen data. This situation is called over fitting. To avoid it, it is common practice when performing a (supervised) machine learning experiment to hold out part of the available data as a test set X_test, y_test. Note that the word “experiment” is not intended to denote academic use only, because even in commercial settings machine learning usually starts out experimentally. We did 11-fold cross validation.

SVM Classifier

SVM Linear Classifier is used to train the data and predict based on the trained model on the test data points.

Results Conclusion

After training the model using the train data, prediction is performed for both Logistic Regression and SVM models and their Accuracy Scores are computed as shown below.

Data Categorization	Logistic Regression Accuracy Score	SVM Accuracy Score
Train Data	39.64%	95.82%
Test Data	38.56%	95.88%

We can see with SVM Classifier we have got very satisfying results and we are able to distinguish the banana, wine and background with good accuracy.

References

[1] https://archive.ics.uci.edu/ml/datasets/Gas+sensors+for+home+activity+monitoring

[2] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/HEWNOU

[3] https://scikit-learn.org/stable/modules/svm.html