

• IMPROVING THE SMOKE • DETECTOR

By: Angus Leung

Design

- Client: A small start-up looking to shake up and revolutionize the smoke detector market.
- Objective: Create a smoke alarm that keeps all the great safety considerations of having a smoke detector while avoiding the annoyances of traditional smoke detectors
- Goal: Produce a classification model that can precisely classify when smoke means fire vs when smoke means dinner is on the way.

Data

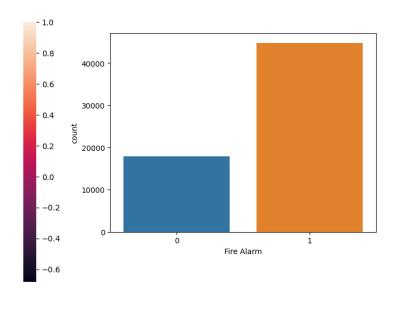
- This dataset was found on Kaggle. The data was created from a DIY smoke detector made of an Arduino board and some sensors (around \$150 USD worth of hardware)
- The dataset contains 62630 readings from a sensor in various different environments, with 15 features measured.
- Tools: Numpy, Pandas, Scikit-learn, Matplotlib, and Seaborn

	итс	Temperature[C]	Humidity[%]	TVOC[ppb]	eCO2[ppm]	Raw H2	Raw Ethanol	Pressure[hPa]	PM1.0	PM2.5	NC0.5	NC1.0	NC2.5	CNT	Fire Alarm
0	1654733331	20.000	57.36	0	400	12306	18520	939.735	0.0	0.0	0.0	0.0	0.0	0	0
1	1654733332	20.015	56.67	0	400	12345	18651	939.744	0.0	0.0	0.0	0.0	0.0	1	0
2	1654733333	20.029	55.96	0	400	12374	18764	939.738	0.0	0.0	0.0	0.0	0.0	2	0
3	1654733334	20.044	55.28	0	400	12390	18849	939.736	0.0	0.0	0.0	0.0	0.0	3	0
4	1654733335	20.059	54.69	0	400	12403	18921	939.744	0.0	0.0	0.0	0.0	0.0	4	0

• One row of data shows UTC(seconds), temperature (in degrees Celsius), humidity, TVOC (Total Volatile Organic Compounds), eCO2 (co2 equivalent concentration), Raw H2, Raw Ethanol, Pressure [hPa] (Air Pressure), PM 1.0, PM 2.5 (particulate matter size), NC0.5, NC1.0, NC2.5 (Number concentration of particulate matter), CNT (Sample counter) and Fire Alarm.

Exploratory Data Analysis

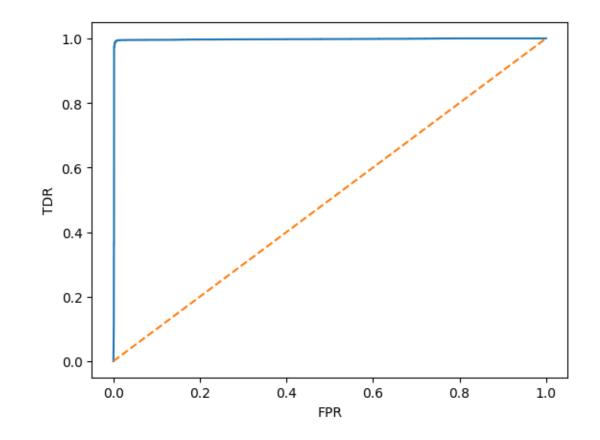




- High correlations for PM1.0, PM2.5, NC0.5, NC1.0 & NC2.5. Dropping PM1.0, NC0.5, and NC1.0.
- Imbalance in Fire Alarm count (a little more than 1:2)

Modeling

- Logistic Regression
 - Initial logistic model had a training score of 98.87%, precision score of 1, and F1 score of 0.99
- Additional Logistic Regression with the 'liblinear' solver for convergence



Additional Tuning

- Oversampling
 - Very similar results on training data
- Tuning C
 - Grid search to tune C resulted in 1 being the best configuration

Gradient Boosting

• MSE: 0

• RMSE: 0

• Training score: 100%

Complexity

- Computational complexity
- Space complexity
- Time complexity
 - Logistic Regression: 0.002 seconds
 - Gradient Boosting: 0.102 seconds