International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 06 June 2024
Page No: 1755- 1761
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0241

Smart Spam Detection and Correction for Temperature in Heater

Ms. Dhanashri Ashok Patil[1], Dr. Jayashree S Awati[2]
[1,2]Department of Electronics Engineering, Rajarambapu Institute of Technology, Rajaramnagar, (Affiliated to Shivaji University, Kolhapur), Islampur, India
Emails: *dhanupatil810@gmail.com[1], jayashree.awati@ritindia.edu[2]*

Abstract

The efficient operation of heating systems relies heavily on accurate temperature monitoring. However, such systems are vulnerable to data corruption, which can lead to erroneous temperature readings and potentially hazardous conditions. We propose a novel approach to address this challenge by employing machine learning techniques for spam detection and correction in heater temperature data for the detection phase.

We explore various machine learning algorithms including spam detection and classification models, to identify spam temperature readings. These models are trained on the preprocessed dataset and evaluated using appropriate metrics to assess their performance.

Keywords: *Spam Detection, Temperature Control, Heater Systems, Python, Machine Learning, Linear Regression.*

1. Introduction

In modern heating systems, maintaining precise temperature control is essential for efficiency and comfort. In this research paper, we propose a novel approach to smart spam detection and correction for temperature monitoring in heaters using python machine learning techniques. Machine learning offers a powerful toolkit for identifying patterns and spam in large datasets, making it well-suited for this task. Traditional methods of spam detection often rely on predefined thresholds or simple statistical measures, which can be insufficient in handling complex and dynamic environments typical of heating systems. Machine learning algorithms, on the other hand, can learn from historical data, adapt to new patterns, and provide more accurate and reliable detection and correction mechanisms. In the context of heating systems, temperature spam can have severe consequences, such as increased energy consumption, equipment damage, and reduced comfort levels. Moreover, incorrect temperature readings can also lead to inaccurate diagnoses and inefficient maintenance procedures. Therefore, it is essential to develop effective methods to detect and correct temperature spam in real-time. Existing methods for detecting spam data in smart heaters typically rely manual inspection or rulebased filtering, which can be time-consuming and prone to errors. These methods may not be effective in detecting complex patterns of spam data. The significance of this research in its potential to revolutionize temperature monitoring systems in various domains, including residental and industrial heating applications. By mitigating the impact of spam data, our proposed solution not only improves temperature control but also reduces energy consumption and maintenance costs. The contributions of this paper:

1. We propose a machine learning-based approach for detecting spam data in smart heaters, which can improve the accuracy and reliability of temperature readings.
2. We develop a correction algorithm that can adjust the temperature readings based on the detected spam data, ensuring that the heating system operates efficiently and safely.

2. Literature Survey

2.1. Spam

The heater adjusts its power output based on the temperature of the room. When the room temperature is lower than the set point, the heater will increase power output to heat the room faster. As the room temperature approaches the set point, the heater will gradually decrease its power output to maintain the set temperature.

2.2. The Benefits of Spam Include

1. **Faster Heating:** By increasing power output when the room is cold, spam helps to heat the room quickly.
2. **Energy Efficiency:** By adjusting power output based on temperature, spam can help reduce energy consumption and minimize standby losses.
3. **Improved Temperature Control:** Spam helps to maintain a consistent temperature by adjusting power output to match changing room conditions.

2.3. Spam Detection and Correction

The involves looking into existing methods, algorithms and techniques used for spam detection and correction in various domains such as email, social media, heater, or other communication channel.

2.4. Temperature Control in Heaters

Exploring literature related to temperature control systems, especially in heaters or similar devices, will provide insights into different control strategies, feedback mechanisms and algorithms used to maintain or regulate temperature effectively and efficiently.

2.5. Python Machine Learning

Reviewing literature on machine learning techniques and algorithms implemented using python will provide a foundation for understanding the tools and methodologies available for building machine learning tools. This includes exploring topics such as classification, regression, clustering and NLP, among others.

2.6. Integration of Machine Learning in Temperature Control

Investigating studies that have applied machine learning techniques to temperature control systems or similar domains can offer valuable insights into the challenges, opportunities and best practices for integrating machine learning into such systems.

3. Methodology

The spam detection and correction for temperature in heater using machine learning algorithm

3.1. Software

For implementing this project using python machine learning, I have used jupyter notebook in python software and various libraries. Python is the primary programming language used for machine learning project due to its simplicity, readability, and the availability of numerous libraries for data manipulation, visualization, and machine learning model development.

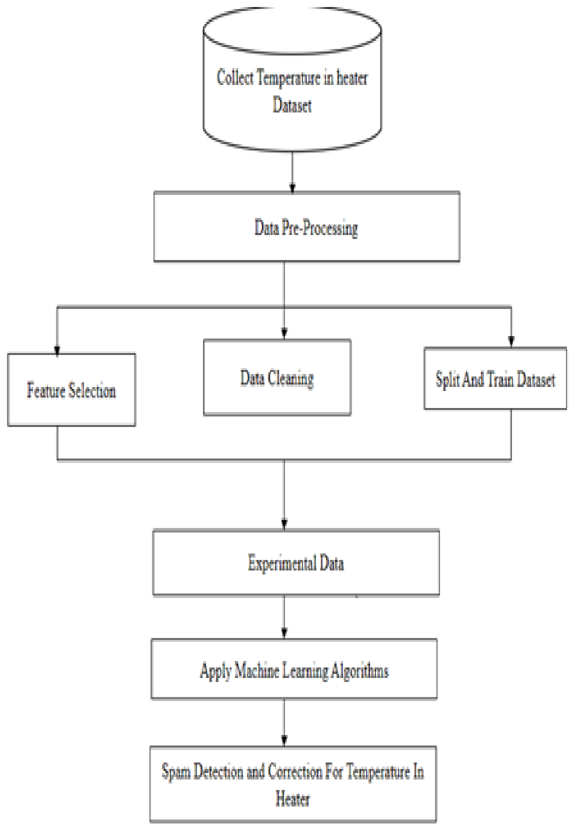


Figure 1 Flowchart of Spam Detection in Temperature in Heater

We start with spam detection and correction for temperature in heater data This is shown in figure.1 according to the above flow. After the data has been collected, we have train our machine learning model using the data they are not directly applicable to our project. To accomplish this our data needs to be preprocesed. Following that, I have split our data into training and test data, which will be used in training and evaluating our model. Once I have done that, I have feed data into our Linear Regression model. [1]

3.2. Importing Libraries

I have started by first imported libraries I have been used shown in figure.2. Using matplotlib you can plot graphs, histogram and bar plot. Pandas for data manipulation and analysis, Numpy is to do the mathematical and scientific operation. [2] Seaborn is used for making statistical graphics more attractive and informative. Statsmodels is used for statistical modeling and hypothesis testing.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
```

Figure 2 Packages

4. Data Collection

Collect a dataset of temperature readings from the heater, along with corresponding labels indicating whether the reading is normal or spam.

4.1. Dataset

The dataset used in this research paper is available on Kaggle this dataset shown in figure 3.

```
[4]: # Import Dataset

[1]: import pandas as pd
     df = pd.read_csv("Temperature_heater_dataset.csv")

[6]: df.head(5)
```

[6]:

	Time (s)	CO (ppm)	Humidity (%r.h.)	Temperature (C)	Flow rate (mL/min)	Heater voltage (V)	R1 (MOhm)	R2 (MOhm)	R3 (MOhm)	R4 (MOhm)	R5 (MOhm)	R6 (MOhm)	R7 (MOhm)	R8 (MOhm)	R9 (MOhm)	R10 (MOhm)	R1 (MOhm
0	0.000	0.0	49.7534	23.7184	233.2737	0.8993	0.2231	0.6365	1.1493	0.8483	1.2534	1.4449	1.9906	1.3303	1.4480	1.9148	3.465
1	0.309	0.0	55.8400	26.6200	241.6323	0.2112	2.1314	5.3552	9.7569	6.3188	9.4472	10.5769	13.6317	21.9829	16.1902	24.2780	31.101
2	0.618	0.0	55.8400	26.6200	241.3888	0.2070	10.5318	22.5612	37.2635	17.7848	33.0704	36.3160	42.5746	49.7495	31.7533	57.7289	53.627
3	0.926	0.0	55.8400	26.6200	241.1461	0.2042	29.5749	49.5111	65.6318	26.1447	58.3847	67.5130	68.0064	59.2824	36.7821	66.0832	66.834
4	1.234	0.0	55.8400	26.6200	240.9121	0.2030	49.5111	67.0368	77.8317	27.9625	71.7732	79.9474	79.8631	62.5385	39.6271	68.1441	62.094

Figure 3 Dataset

5. Data Preprocessing

Clean and preprocess the data for analysis. This might involve handling missing values, normalizing the data and labeling the data if necessary.

5.1. Data Cleaning

5.1.1. Handling Missing Values

Handling missing values this is shown in figure 4 for demonstration and filled them using the rolling mean of temperature column.

```
[9]: #check for missing values in dataset

[10]: df.isnull().sum()

[10]: Time (s)              0
      CO (ppm)              0
      Humidity (%r.h.)      0
      Temperature (C)       0
      Flow rate (mL/min)    0
      Heater voltage (V)    0
      R1 (MOhm)             0
      R2 (MOhm)             0
      R3 (MOhm)             0
      R4 (MOhm)             0
      R5 (MOhm)             0
      R6 (MOhm)             0
      R7 (MOhm)             0
      R8 (MOhm)             0
      R9 (MOhm)             0
      R10 (MOhm)            0
      R11 (MOhm)            0
      R12 (MOhm)            0
      R13 (MOhm)            0
      R14 (MOhm)            0
      dtype: int64
```

Figure 4 Handling Missing Values

5.1.2. View Summary of Dataset

A typical dataset for temperature monitoring in heaters might include the following columns belown shown in figure 5.

```
[7]: # view summary of dataset

[8]: df.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 295719 entries, 0 to 295718
     Data columns (total 20 columns):
      #   Column              Non-Null Count   Dtype
     ---  ------              --------------   -----
      0   Time (s)            295719 non-null  float64
      1   CO (ppm)            295719 non-null  float64
      2   Humidity (%r.h.)    295719 non-null  float64
      3   Temperature (C)     295719 non-null  float64
      4   Flow rate (mL/min)  295719 non-null  float64
      5   Heater voltage (V)  295719 non-null  float64
      6   R1 (MOhm)           295719 non-null  float64
      7   R2 (MOhm)           295719 non-null  float64
      8   R3 (MOhm)           295719 non-null  float64
      9   R4 (MOhm)           295719 non-null  float64
      10  R5 (MOhm)           295719 non-null  float64
      11  R6 (MOhm)           295719 non-null  float64
      12  R7 (MOhm)           295719 non-null  float64
      13  R8 (MOhm)           295719 non-null  float64
      14  R9 (MOhm)           295719 non-null  float64
      15  R10 (MOhm)          295719 non-null  float64
      16  R11 (MOhm)          295719 non-null  float64
      17  R12 (MOhm)          295719 non-null  float64
      18  R13 (MOhm)          295719 non-null  float64
      19  R14 (MOhm)          295719 non-null  float64
     dtypes: float64(20)
     memory usage: 45.1 MB
```

Figure 5 Summary of Dataset

5.1.3. Columns

The pandas dataframe below shown in figure.6 lists the names of the columns in the dataframe.

```
[11]: df.columns

[11]: Index(['Time (s)', 'CO (ppm)', 'Humidity (%r.h.)', 'Temperature (C)',
             'Flow rate (mL/min)', 'Heater voltage (V)', 'R1 (MOhm)', 'R2 (MOhm)',
             'R3 (MOhm)', 'R4 (MOhm)', 'R5 (MOhm)', 'R6 (MOhm)', 'R7 (MOhm)',
             'R8 (MOhm)', 'R9 (MOhm)', 'R10 (MOhm)', 'R11 (MOhm)', 'R12 (MOhm)',
             'R13 (MOhm)', 'R14 (MOhm)'],
            dtype='object')
```

Figure 6 List of Columns

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 06 June 2024
Page No: 1755- 1761
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0241

5.2. Separate Feature and Target Variable

5.2.1.Feature Variable

The feature variable this shown in figure.7 or independent variable is the input data used to make classifications. In this case, the feature variable would be the temperature readings from the heater.

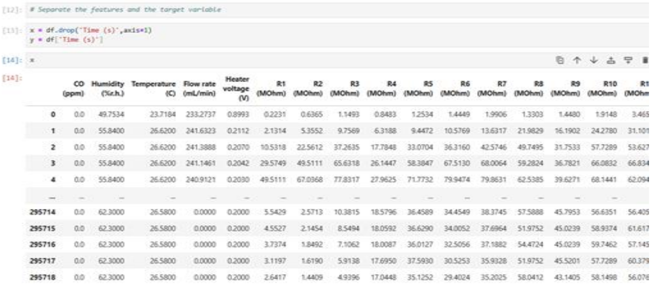


Figure 7 Separate Feature Variable

5.2.2.Target Variable

The target variable this shown in figure.8 or dependent variable is the variable we want to classify based on the feature variables. In this case, the target variable could be whether each temperature reading is normal or spam.

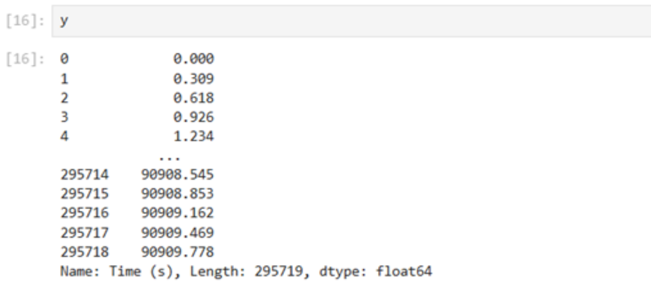


Figure 8 Separate Target Variable

6. Splitting Training and Testing Data

The next part is the most important since we used one set of data to test our model and another set to evaluate it. In othere words, part of the X will be our training data, and the other part will be our test data. The same applies to Y. [3]

6.1. Training Dataset

This training dataset shown in figure.9 will be used to train the machine learning model to detect and correct spam temperature readings. It should include a sufficient number of normal and spam temperature readings. A common split is around 70-80% of the data for training.

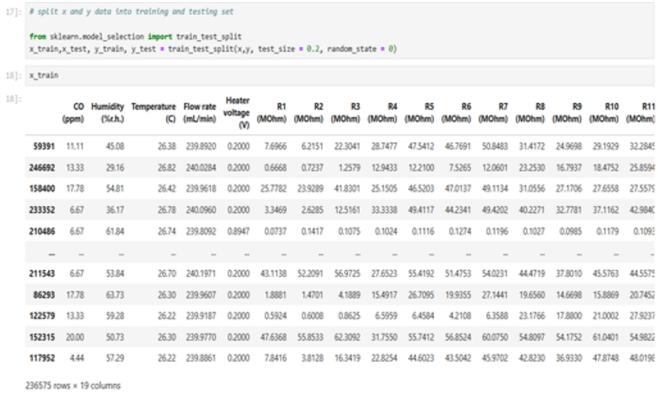


Figure 9 Training Dataset

6.2. Testing Dataset

This testing dataset shown in figure.10 will be used to evaluate the performance of the trained model. It should also include a mix of normal and spam temperature readings, but it should be distinct from the training dataset to ensure an unbiased evaluation. The remaining 20-30% of the data is typically used for testing.

Figure 10 Testing Dataset

6.3. Model Training

Using linear regression to train the model based on the preprocessed data to learn patterns and relationships between a dependent variable and one or more independent variables this shown in figure 11.

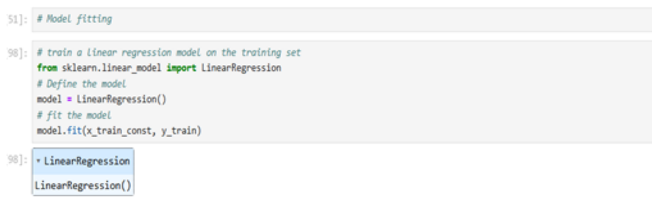


Figure 11 Model Training

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 06 June 2024
Page No: 1755- 1761
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0241

6.4. Compare Train and Test Set Accuracy

It's essential to evaluate the performance of your model on both the training data and the testing data. This is because the training data is used to train the model, while the testing data is used to evaluate its performance on unseen data.

Accuracy Formula:

Accuracy is a measure of how well your model predicts the correct output this shown in figure.12.

Accuracy = (TP+TN)/(TP+TN+FP+FN)

Where:

TP = True Positives (correct predictions)
TN = True Negatives (correct predictions)
FP = False Positives (incorrect predictions)
FN = False Negatives (incorrect predictions)

```
[ ]: # Compare the train-set and test-set accuracy

[48]: y_pred_train = model.predict(x_train)
      y_pred_train

[48]: array([34959.89814237, 73880.23720063, 60778.80737922, ...,
             27926.43061417, 67402.45054057, 23661.85604638])
```

Figure 12 Train and Test Set Accuracy

6.5. Training and Test Set Accuracy Score

The accuracy score is a measure of how well your machine learning model performs on the test set. This is shown in figure.13. It's the proportion of correctly classified instances (correct/incorrect temperature readings) out of the total number of instances in the test set.

Formula:

Accuracy = (TP+ TN)/(Total Samples)

Where:

1. True Positives (TP) = Correctly classified correct temperature readings.
2. True Negatives (TN) = Correctly classified incorrect temperature readings.
3. True Samples = Total number of instances in the test set.

```
[ ]: # print the scores on training and test set

[25]: print('Training set score: {:.4f}'.format(model.score(x_train, y_train)))
      print('Test set score: {:.4f}'.format(model.score(x_test, y_test)))

Training set score: 0.4411
Test set score: 0.4382
```

Figure 13 Accuracy Score

6.6. Confusion Matrix

A confusion matrix is a matrix the summarizes the performance of a machine learning model on a set of test data. It is means of displaying the number of accurate and inaccurate instances based on the model's predictions. It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance. The specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning model. It is used for classification problems where the output can be two or more classes. The matrix itself is a square table with dimensions equal to the number of classes in the classification problem. this confusion matrix shown in figure.14. Confusion matrix is a very good way to understand results like true positive, false positive, true negative and so on [4].

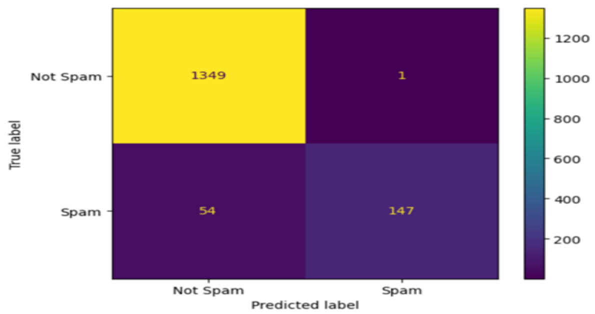


Figure 14 Confusion Matrix

6.7. Precision & Recall

Precision it is ratio of true positive predictions to the total predicted positives.

Precision = TP/(TP+FP)

Recall It is the ration of true positive predictions to the total actual positives.

Recall = TP/(TP+FN)

- TP = True Positive
- FP = False Positive
- TN = True Negative
- FN = False Negative
- Precision = 99%
- Recall = 91.3%

7. Machine Learning Algorithm

I have used linear regression algorithm in this project.

7.1. Linear Regression

Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation to observed data [5]. When there is only one independent variable, it is known as Simple Linear Regression and when there are multiple independent variables, it is known as Multiple Linear Regression. A supervised machine learning algorithm that learns from the labelled dataset and maps the data points to the most optimized linear functions. Which can be used for predictions on new dataset [6].

Classification: It predicts the class of the dataset based on the independent input variable, class is the categorical or discrete values [7].

Regression: It predicts the continuous output variables based on the independent input variable, like the predictions of spam mail [8-10].

7.2. Types of Linear Regression

7.2.1. Simple Linear Regression

This is the simplest form of linear regression, and it involves only one independent variable and one dependent variable. The equation for simple linear regression is:

$$Y=\beta 0+\beta 1X$$

Where:

- Y is the dependent variable
- X is the independent variable
- β0 is the intercept
- β1 is the slope

7.2.2. Multiple Linear Regression

This involves more than one independent variable and one dependent variable. The equation for multiple linear regression is [11]:

$$Y=\beta 0+\beta 1X+\beta 2X+\dots\dots\dots\beta nX$$

Where:

- Y is the dependent variable
- X1, X2, …., X_p are the independent variables
- β0 is the intercept
- β1, β2, …., βn are the slopes

8. Calculate Spam in Temperature

Calculate the spam in temperature based on the provided variables we'll denote the spam in temperature as T which is the difference between the maximum and minimum temperature.

8.1. Identify Variables

- CO: carbon monoxide concentration
- H: Humidity
- T: Temperature
- F: Flowrate
- V: Heater Voltage
- R: Resistance
- Ri: Values of resistors
- β0: Intercept
- β1, β2..., β5+nβ1, β2….., β5+n: Coefficients for each independent variable

ϵϵ: Error term

8.2. General Linear Model for Temperature

Assuming a linear relationship (which can be modified if the relationship is known to be non-linear):

$$T= \beta 0 + \beta 1\cdot CO + \beta 2\cdot H + \beta 3\cdot F + \beta 4\cdot V + \beta 5\cdot R + \sum I =1n \; \beta 5+i\cdot Ri+\epsilon$$

8.3. Spam Calculation

The spam of temperature T is given by:

$$T= 〖T〗\_max - 〖T〗\_min$$

8.3.1. Determining 〖T〗_max and T_min

To find 〖T〗_max and T_min calculates the temperature using the maximum and minimum values of each variable.

Then,

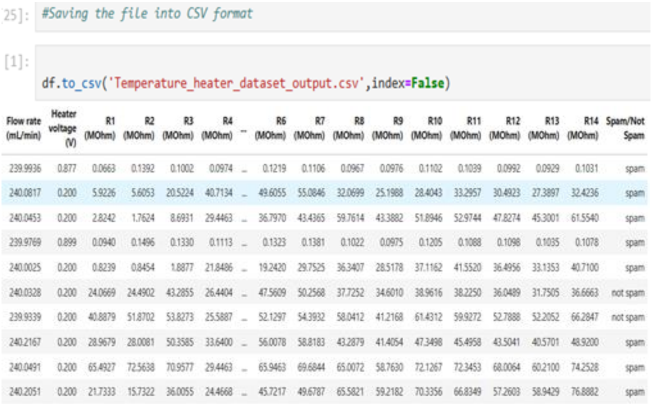
$$T\_max = \beta 0 + \beta 1\cdot CO\_max+ \beta 2\cdot H\_max+\beta 3\cdot F\_max + \beta 4\cdot V\_max + \beta 5\cdot R\_max+\sum i=1n\beta 5 + i\cdot 〖Ri〗\_max + \epsilon$$

$$T\_min = \beta 0 + \beta 1\cdot CO\_min+ \beta 2\cdot H\_min+ \beta 3\cdot F\_min + \beta 4\cdot V\_min + \beta 5\cdot R\_min+\sum i=1n\beta 5 + i\cdot 〖Ri〗\_min + \epsilon$$

9. Result and Discussion

In this session, all the results are presented in graph, tables. All the results are discussed in detail. This code that the dataset is stored in csv files named dataset.csv. This CSV file shown in figure.15. The code loads the data, defines the features and target variable, create a linear regression model, fits the model to each dataset, predicts the temperature for each dataset, calculate the error term and spam or not spam classification for each dataset, and saves the results to a CSV file. Finally, it evaluates the accuracy of the model using the accuracy_score function from scikit_learn [12].

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 06 June 2024
Page No: 1755- 1761
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0241



```
[25]: #Saving the file into CSV format

[1]: df.to_csv('Temperature_heater_dataset_output.csv',index=False)
```

Flow rate (mL/min)	Heater voltage (V)	R1 (MOhm)	R2 (MOhm)	R3 (MOhm)	R4 (MOhm)	...	R6 (MOhm)	R7 (MOhm)	R8 (MOhm)	R9 (MOhm)	R10 (MOhm)	R11 (MOhm)	R12 (MOhm)	R13 (MOhm)	R14 (MOhm)	Spam/Not Spam
239.9936	0.877	0.0663	0.1392	0.1002	0.0974	...	0.1219	0.1106	0.0967	0.0976	0.1102	0.1039	0.0992	0.0929	0.1031	spam
240.0817	0.200	5.9226	5.6053	20.5224	40.7134	...	49.6055	55.0846	32.0699	25.1988	28.4043	33.2957	30.4923	27.3897	32.4236	spam
240.0453	0.200	2.8242	1.7624	8.6931	29.4463	...	36.7970	43.4365	59.7614	43.3882	51.8946	52.9744	47.8274	45.3001	61.5540	spam
239.9769	0.899	0.0940	0.1496	0.1330	0.1113	...	0.1323	0.1381	0.1022	0.0975	0.1205	0.1088	0.1098	0.1035	0.1078	spam
240.0025	0.200	0.8239	0.8454	1.8877	21.8486	...	19.2420	29.7525	36.3407	28.5178	37.1162	41.5520	36.4956	33.1353	40.7100	spam
240.0328	0.200	24.0669	24.4902	43.2855	26.4404	...	47.5609	50.2568	37.7252	34.6010	38.9616	38.2250	36.0489	31.7505	36.6663	not spam
239.9339	0.200	40.8879	51.8702	53.8273	25.5887	...	52.1297	54.3932	58.0412	41.2168	61.4312	59.9272	52.7888	52.2052	66.2847	not spam
240.2167	0.200	28.9679	28.0081	50.3585	33.6400	...	56.0078	58.8183	43.2879	41.4054	47.3498	45.4958	43.5041	40.5701	48.9200	spam
240.0491	0.200	65.4927	72.5638	70.9577	29.4463	...	65.9463	69.6844	65.0072	58.7630	72.1267	72.3453	68.0064	60.2100	74.2528	spam
240.2051	0.200	21.7333	15.7322	36.0055	24.4668	...	45.7217	49.6787	65.5821	59.2182	70.3356	66.8349	57.2603	58.9429	76.8882	spam

Figure 15 **CSV File Spam Not Spam**

Conclusion

The conclusion is that algorithm can be used to classify whether a given temperature reading is spam or not spam based on a linear regression model that takes into account several factors. The algorithm can be trained on dataset and then used to predict whether new temperature readings are spam or not spam. This script automates the process of calculating the temperature, classifying the data, and saving the results to new csv files. You can adjust the coefficients and threshold as needed for your specific application.

Reference

[1]. Vinodhini. M, Prithvi. D, Balaji. S "Spam Detection Framework using ML Algorithm" in IJRTE ISSN: 2277-3878, Vol.8 Issue.6, March 2020.

[2]. Javatpoint, "Machine Learning Tutorial" 2017 https://www.javatpoint.com/machine- learning.

[3]. A. Alghoul, S. Al Ajrami, G. Al Jarousha, G. Harb, and S. S. AbuNaser, "Email classification using artificial neural network," International Journal for Academic Development, vol. 2, 2018.

[4]. Yusksel, A. S., CANKAYA, S. F., & Usncus, It. S. (2017). "Design of a Machine Learning Based Predictive Analytics System for Spam Problem." Acta Physica Polonica, A.,132(3). [26] Goodman, J. (2004, July). "IP Addresses in

[5]. Deepika Mallampati, Nagaratna P. Hegde "A MachineLearning Based Email Spam Classification Framework Model" in IJITEE, ISSN: 2278-3075, Vol.9 Issue.4, February 2020.

[6]. G. V. Cormack, "Email Spam Filtering: A Systematic Review," Foundations and Trends® in Information Retrieval, vol. 1, no. 4, pp. 335-455,

[7]. Guzella, T. S. and Caminhas, W. M."A review of machine learning approaches to Spam filtering." Expert Syst. Appl., 2009.

[8]. Linda Huang, Julia Jia, Emma Ingram, Wuxu Peng, "Enhancing the Naive Bayes Spam Filter through Intelligent Text Modification Detection", 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications.

[9]. Jason Brownlee, "Logistic Regression for Machine Learning" The Machine Learning Mastery, April 1, 2016.https://machinelearningmastery.com/logisti c-regression-for-machine-learning/.

[10]. M. Siponen and C. Stucke, "Effective Anti-Spam Strategies in Companies: An International Study," Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), 2006.

[11]. W.A. Awad, S.M. Elseuofi, Machine learning methods for spam E-mail classification, Int. J. Comput. Sci. Inf. Technol. 3 (1) (2011) 173–184.

[12]. Jianying Zhou, Wee-Yung Chin, Rodrigo Roman, and Javier Lopez, (2007) "An Effective MultiLayered Defense Framework against Spam", Information Security Technical Report 01/2007.

[13]. H. Faris, I. Aljarah, and B. Al-Shboul, "A hybrid approach based on particle swarm optimization and random forests for e-mail spam filtering," in Proceedings of the International Conference on Computational Collective Intelligence, Springer, Halkidiki, Greece, September 2016.