Ozone Day Detection using Logistic Regression

Samir Khanal
Department of Computer Science
Lambton College
Toronto, Canada
samirkhanal35@gmail.com

0000-0002-1778-2374

***Abstract*—As the title of the paper, the research is conducted to detect whether the day is Ozone Day or a Normal day. The input data about the different factors are collected from a given UCI dataset. The dataset is then processed through data cleaning, handling of missing values, feature selection, data normalization, and then splitting of data into a training set and testing set. The training set is then fitted into the Logistic Regression model from the scikit learn library. The model is tested with the testing dataset. It gives a perfect model for detecting Ozone Day or Normal day.**

Keywords—Ozone, UCI dataset, Missing Values, Feature Selection, Normalization, Logistic Regression

I. INTRODUCTION

Ozone level generally means the level of ozone gas in the atmosphere. Ozone gas generally forms by the chemical reactions between the natural and man-made emissions of volatile organic compounds and nitrogen oxides in the presence of Sunlight. Ozone gas is made up of three oxygen atoms. High levels of Ozone in the air cause several health difficulties like difficulty breathing, coughing, sore throat, and damage to the respiratory system. We can detect Ozone Day through different metrics and the Ozone level in the atmosphere. We can also predict the day on the basis of different factors correlated with it.

II. METHODOLOGY

A. Data Acquisition

The Ozone Level Detection dataset from UCI Machine Learning Repository[1] was used as the data set. It was prepared by Kun Zhang, Wei Fan, and XiaoJing Yuan. It was also used for the research on stochastic ozone days[2]. It has 2536 instances, 73 Attributes or features, and a target or dependent variable with binary data of 0 and 1.

B. Data Preprocessing

While checking the data, all the data were object types rather than string types or numeric types. So, the data was converted into a numeric type of float type data. And the float type data of the target, dependent variable 'day', is converted into the integer type.

1. Missing Value Handling

   The null or missing values are represented as '?' by the dataset collector. So, the '?' values are replaced with 'NaN' values with the pandas replace function. The conversion of object data type to float data type was not possible before the replacement of '?'. Now, all the feature columns are converted into a floating data type. The duplication of data was also checked but found none.

2. Feature Selection

   The correlation between the features was calculated and rounded to the decimal point of two decimals. Heatmaps were created using the correlation data of all 73 features. Eight feature columns were selected by keeping the threshold of 16% correlation percentage. They are: 'T_PK', 'T18', 'T17', 'T16', 'T15', 'T14', 'T13', and 'T12'.

3. Data Normalization and Splitting of the dataset

   To normalize the data; the scaling data normalization method was applied to the feature columns of the dataset using the formula in equation (1).

$$\text{Normalized value(x)} = \frac{x - \textit{minimum value}}{\textit{maximum value} - \textit{minimum value}} \quad (1)$$

   Then the data was split into 70% of the training set and the rest into the testing set using the python array splitting method.

C. Data Modelling

The target variable or dependent variable only had two categories of data 0 and 1. So, the binary classification technique, Logistic regression algorithm was used to train the model. The basic Logistic regression algorithm is calculated by using the formula in the equation (2).

$$y = \frac{e^{(b0+b1X)}}{1 + e^{(b0+b1X)}} \quad (2)$$

Here, x = input value, y =predicted output, b0 = bias or intercept term and b1 = coefficient for input(x).

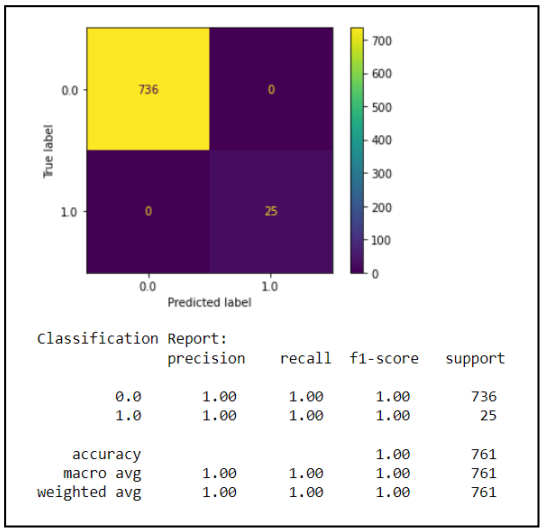
III. PERFORMANCE AND RESULTS

True Positive 736	False Positive 0
False Negative 0	True Negative 25

Fig 1: Confusion Matrix

A Logistic regression model from scikit-learn was used to fit the data. The model gave 100% accuracy to the training dataset. Also, for the test dataset, the model gave 100% accuracy. The confusion matrix of the model for the test

dataset is shown below. It has 736 True Positives and 25 True Negatives, None of the False Negatives and False Positives.



Classification Report:

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	736
1.0	1.00	1.00	1.00	25
accuracy			1.00	761
macro avg	1.00	1.00	1.00	761
weighted avg	1.00	1.00	1.00	761

Fig 2: Confusion Matrix and Classification Report for Test Dataset

IV. CONCLUSION

The method presented in this paper can process the data in Ozone Level Detection Dataset, clean data, deal with missing values, feature selection, normalization, modelling, training, and predicting the Ozone Day or Normal Day.

There is still some further research to do. For example, it has an accuracy of 100% on both training and testing datasets. So, this needs to be verified whether this is the accuracy for general or live environmental data.

ACKNOWLEDGMENT

This research article would not have been possible without the exceptional support of my mentor Prof. Mohammad Islam and the UCI dataset [1]. I am also grateful for the insightful comments offered by the anonymous peer reviewers from my batchmates. The generosity and insights of Prof. Mohammad Islam have improved this study in innumerable ways. Thank you all for your support.

REFERENCES

[1] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[2] Zhang, Kun and Fan Wei (2008). Forecasting skewed biased stochastic ozone days: analyses, solutions and beyond, Knowledge and Information Systems, Vol. 14, No. 3.