Logo

Description automatically generated with low confidence

**A Project Report**

**Ozone Level Detection Using Machine Learning Algorithms**

**Name : Prateek Pandey**

**Registration no: 11905876**

**Section : K19HV**

**Date : 10/04/2022**

**Github Link:** https://github.com/Prateek03-tech/Project-Ozone-level-detection-

**INTRODUCTION:**

Ground ozone pollution has been a serious air quality problem over the years and can be extremely harmful to people’s health if no advanced forecasts are provided. However, the occurrence of an ozone polluted day depends on a lot of sophisticated chemical, physical, and geological factors, so it is too complicated and indirect to use simple math formula to calculate the ozone level.

Because of the increasing attention on environmental issues, especially air pollution, predicting whether a day is polluted or not is necessary to people’s health. In order to solve this problem, this project is classifying ground ozone level based on big data and machine learning models, where polluted ozone day has class 1 and non-ozone day has class 0. The dataset used in this project was derived from the UCI Website, containing various environmental factors in Houston, Galveston and Brazoria area that could possibly affect the occurrence of ozone pollution . This dataset is first filled up for further process, next standardized to ensure every feature has the same weight, and then split into training set and testing set. After this, three different machine learning models are used in the prediction of ground ozone level and their final accuracy scores are compared and at last an ensembled model is created with these three classifiers and that helped in booting the accuracy of the model. The Algorithms used are SVM, K nearest neighbors and Logistic Regression.

**Literature Review:**

1. Sarkar, A., Ray, S. S., Prasad, A., & Pradhan, C. (2021, November). A Novel Detection Approach of Ground Level Ozone using Machine Learning Classifiers. In *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 428-432). IEEE.
2. Harrou, F., Dairi, A., Sun, Y., & Kadri, F. (2018). Detecting abnormal ozone measurements with a deep learning-based strategy. *IEEE Sensors Journal*, *18*(17), 7222-7232.
3. Al-Alawi, S. M., Abdul-Wahab, S. A., & Bakheit, C. S. (2008). Combining principal component regression and artificial neural networks for more accurate predictions of ground-level ozone. *Environmental Modelling & Software*, *23*(4), 396-403.
4. Yılmaz, A. (2021). Ozone Level Prediction with Machine Learning Algorithms. *Journal of Aeronautics and Space Technologies*, *14*(2), 177-183.
5. Somula, R., & CH, E. (2018). Ozone Layer Concentration Prediction Using Machine Learning Techniques.

**Proposed Methodology:**

Diagram

Description automatically generated

**Dataset and Preprocessing:**

This dataset is downloaded from the UCI website, called Ozone Level Detection Data Set. It includes 2536 instances with 73 attributes, containing only 1 date data and 72 numeric data. Features like T (temperature), WSR (wind speed resultant), RH (relative humidity), U (u wind: east-west direction wind), V (v wind: north-south direction wind), HT (geopotential height), KI (k index: thunderstorm potential), TT (t totals: assess storm strength), SLP (sea level pressure), SLP\_ (SLP changes from yesterday), Prep (precipitation) are measured each day for 24 hours although a great number of missing values are present. The maximum number of missing values of one feature is 299 in WSR0, and most data is continuously missing on the first 53 attributes . The final prediction is binary, 0 means that day is non-ozone day and 1 means that day is polluted. In this dataset, about 2300 instances are classified as non-ozone days and 200 as ozone-days. Missing values were handled using Imputer. Data is splitted into feature(x) and target(y).

**Data Visualization:**

Peak Temperature Plots

Chart, histogram

Description automatically generated

Peak Wind Speed Plots

Chart, histogram

Description automatically generated

Correlation Matrix

Chart, treemap chart

Description automatically generated

**Algorithms :**

1.Support Vector Machine :

SVM is a machine learning model using kernel method (non-linear regression) to make binary predictions. In linear SVM, data is first transferred to an eigenspace by non-linear mapping, then a linear machine learning method is used to classify data. Linear SVM’s decision rule can be shown using the inner product of train and test sets. Kernel method, on the other hand, is able to directly calculate the inner product inside eigenspace, thus can handle non-linear division problems.94.87% accuracy was there with this algorithm

2. Logistic Regression is a statistical model using a logistic function to predict binary outcomes. Since this dataset contains too many attributes, feature attraction is needed before building the actual machine learning model. Principal Component Analysis (PCA) is a statistical procedure that is used to convert a set of possible related variables into a set of linearly uncorrelated variables called principal components, which means to reduce the dimensions of this dataset and select some principal features.

3.KNN:

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It’s easy to implement and understand, but has a major drawback of becoming significantly slows as the size of that data in use grows. KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

**Ensembling and Result:**

Ensembling is a powerful technique to improve the performance of the model by combining various base models in order to produce an optimal and robust model.

Technique used here is Voting Classifier with accuracy of 95.06% which is more than any individual algorithms used .