OZONE LEVEL DETECTION

using

Machine Learning

Prepared by : 1. Kunal Demla (102003088)
2. Gaurav Pahwa (102003087)

Submitted to : Ms. Suchita Sharma

November 14, 2022

Contents

1 Introduction

Ground ozone level depends on a sophisticated chemical and physical process as a function of many known and unknown factors, and stochastic in nature (i.e., with the same set of currently observable variables, the ozone level can differ from time to time). For many years, it has been an active topic for air quality study, an interdisciplinary field among atmospheric research, geochemistry and geophysics, since an ozone level above some well known threshold is rather harmful to human health, and affects other important parts of our daily life, such as farming, tourism etc. Therefore an accurate ozone alert forecasting system is necessary to issue warnings to the public before the ozone reaches a dangerous level. In air quality study, the estimation of ozone level uses known physical and chemistry reaction theories that attempt to explain the "true" mechanisms. There are several such theories around. As a result of these research, simulation systems, physical formulas and parametric models are created to calculate ozone level.

However, due to the difficulty of the problem and still limited knowledge about the true physical and chemical mechanism, existing approaches can only use a rather small number of parameters (≤ 10), and are still rather inaccurate and can be costly to build. However, it is a common belief among environmental scientists that a significant large number of other features currently never explored yet are very likely useful in building highly accurate ozone prediction model. Yet, little is known on exactly what these features are and how they actually interact in the formation of ozone. Information available is rather speculative in the sense that we roughly know a rather exhaustive set of features out there. Indeed, this candidate list contains over 60 features. This provides a wonderful opportunities for data mining.

2 Technology Used

2.1 Preprocessing

2.1.1 Principle Component Analysis

Principal component analysis (PCA) is a popular technique for analyzing large datasets containing a high number of dimensions/features per observation, increasing the interpretability of data while preserving the maximum amount of information, and enabling the visualization of multidimensional data. Formally, PCA is a statistical technique for reducing the dimensionality of a dataset. This is accomplished by linearly transforming the data into a new coordinate system where (most of) the variation in the data can be described with fewer dimensions than the initial data. Many studies use the first two principal components in order to plot the data in two dimensions and to visually identify clusters of closely related data points. Principal component analysis has applications in many fields such as population genetics, microbiome studies, and atmospheric science.

2.1.2 Random Oversampling

The learning phase and the subsequent prediction of machine learning algorithms can be affected by the problem of imbalanced data set. The balancing issue corresponds to the difference of the number of samples in the different classes. One way to fight this issue is to generate new samples in the classes which are under-represented. The most naive strategy is to generate new samples by randomly sampling with replacement the current available samples.

2.1.3 Standardization

Standardization entails scaling data to fit a standard normal distribution. A standard normal distribution is defined as a distribution with a mean of 0 and a standard deviation of 1.

2.2 Random Forest

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of

over-fitting to their training set. ." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

2.3 Support Vector Machine (with Radial Bias Function Kernel)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

RBF is a kernel function that is used in machine learning to find a non-linear classifier or regression line. Kernel Function is used to transform n-dimensional input to m-dimensional input, where m is much higher than n then find the dot product in higher dimensional efficiently. The main idea to use kernel is: A linear classifier or regression curve in higher dimensions becomes a Non-linear classifier or regression curve in lower dimensions.

2.4 Logistic Regression

It is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds. Types: Binary, Multinomial, Ordinal Logistic Regression.

Regularization is one of the most important concepts of machine learning. It is a technique to prevent the model from overfitting by adding extra information to it. Sometimes the machine learning model performs well with the training data but does not perform well with the test data. It means the model is not able to predict the output when deals with unseen data by introducing noise in the output, and hence the model is called overfitted. This problem can be deal with the help of a regularization technique. This technique can be used in such a way that it will allow to maintain all variables or features in the model by reducing the magnitude of the variables. Hence, it maintains accuracy as well as a generalization of the model. It mainly regularizes or reduces the coefficient

of features toward zero. In simple words, "In regularization technique, we reduce the magnitude of the features by keeping the same number of features."

2.5 K-Nearest Neighbour

k-NN is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, if the features represent different physical units or come in vastly different scales then normalizing the training data can improve its accuracy dramatically. Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor. The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

3 Dataset Description

The dataset is sparse (72 features, and 2 or 5the criteria of "ozone days"), evolving over time from year to year, limited in collected data size (7 years or around 2,500 data entries), contains a large number of irrelevant features, is biased in terms of "sample selection bias", and the true model is stochastic as a function of measurable factors. Besides solving a difficult application problem, this dataset offers a unique opportunity to explore new and existing data mining techniques, and to provide experience, guidance and solution for similar problems

- Number of Instances: 2536
- Number of Attributes: 73

3.1 Relevant Information

The following are specifications for several most important attributes that are highly valued by Texas Commission on Environmental Quality (TCEQ). More details can be found in the two relevant papers.

- O_3 - Local ozone peak prediction
- Upwind - Upwind ozone background level
- EmFactor - Precursor emissions related factor
- Tmax - Maximum temperature in degrees F
- Tb - Base temperature where net ozone production begins (50 F)
- SRd - Solar radiation total for the day
- WSa - Wind speed near sunrise (using 09-12 UTC forecast mode)
- WSp - Wind speed mid-day (using 15-21 UTC forecast mode)

3.2 Attribute Information

- Date: ignore.
- WSR0: continuous.
- WSR1: continuous.
- WSR2: continuous.
- WSR3: continuous.
- WSR4: continuous.

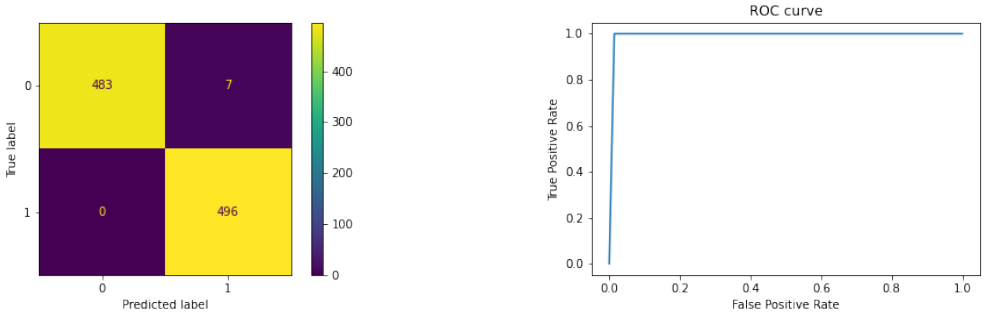
- WSR5: continuous.
- WSR6: continuous.
- WSR7: continuous.
- WSR8: continuous.
- WSR9: continuous.
- WSR10: continuous.
- WSR11: continuous.
- WSR12: continuous.
- WSR13: continuous.
- WSR14: continuous.
- WSR15: continuous.
- WSR16: continuous.
- WSR17: continuous.
- WSR18: continuous.
- WSR19: continuous.
- WSR20: continuous.
- WSR21: continuous.
- WSR22: continuous.
- WSR23: continuous.
- WSR_PK: continuous.
- WSR_AV: continuous.
- T0: continuous.
- T1: continuous.
- T2: continuous.
- T3: continuous.
- T4: continuous.
- T5: continuous.
- T6: continuous.
- T7: continuous.
- T8: continuous.
- T9: continuous.
- T10: continuous.
- T11: continuous.
- T12: continuous.
- T13: continuous.
- T14: continuous.
- T15: continuous.
- T16: continuous.
- T17: continuous.
- T18: continuous.
- T19: continuous.
- T20: continuous.
- T21: continuous.
- T22: continuous.
- T23: continuous.
- T_PK: continuous.
- T_AV: continuous.
- T85: continuous.
- RH85: continuous.
- U85: continuous.
- V85: continuous.
- HT85: continuous.
- T70: continuous.
- RH70: continuous.
- U70: continuous.
- V70: continuous.
- HT70: continuous.
- T50: continuous.
- RH50: continuous.
- U50: continuous.
- V50: continuous.
- HT50: continuous.
- KI: continuous.
- TT: continuous.
- SLP: continuous.
- SLP_: continuous.
- Precp: continuous.

4 Results

This section compares the results provided by different models. These performance parameters were compared to determine the best performing algorithm for predicting Ozone day occurrences. Table 4.1 shows the performance outcome parameters of the classification algorithms employed, namely Accuracy, Precision, Recall, F1 score, Sensitivity, Specificity and AUC. These all show good outcomes, with RF providing maximal accuracy, sensitivity and specificity, followed by SVM which shows better performance than LR and KNN.

Table 4.1: Results and Comparisons of different models

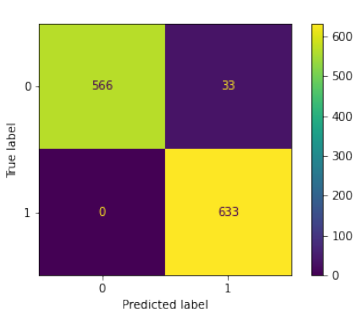
Sr No.	Model	Accuracy	Precision	Recall	F1 score	Sensitivity	Specificity	AUC
1	Random Forest	0.9929	0.9860	1.0	0.9929	1.0	0.9857	0.9928
2	SVM	0.9732	0.9504	1.0	0.9745	1.0	0.9449	0.9724
3	Logistic Regression (With L2 Regularization)	0.9026	0.8787	0.9354	0.9062	0.9354	0.8693	0.9024
4	KNN ($k = \sqrt{N}$)	0.8691	0.8147	0.9576	0.8804	0.9576	0.7795	0.8686



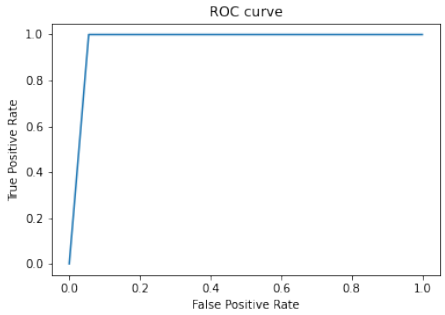
(a) Confusion Matrix

(b) Area under Curve

Figure 4.1: Results with Random Forest

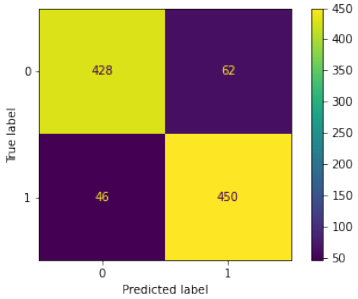


(a) Confusion Matrix

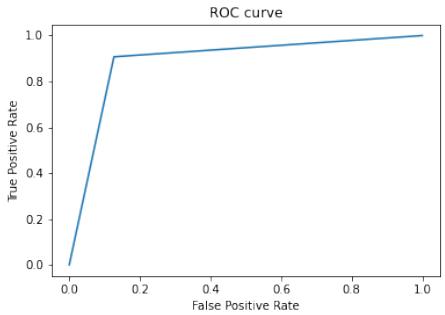


(b) Area under Curve

Figure 4.2: Results with SVM

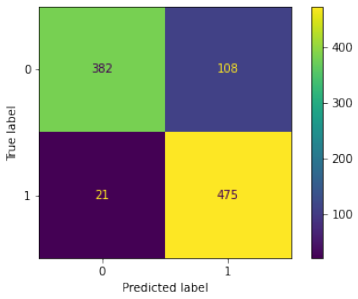


(a) Confusion Matrix

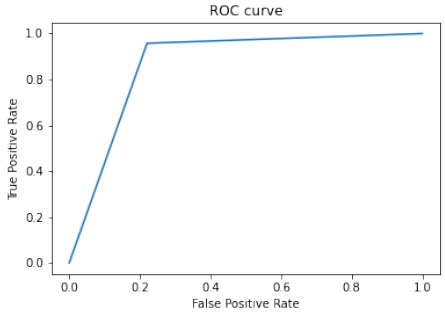


(b) Area under Curve

Figure 4.3: Results with Logistic Regression



(a) Confusion Matrix



(b) Area under Curve

Figure 4.4: Results with K-Nearest Neighbour

Bibliography

[1] K. Zhang, W. Fan, Forecasting skewed biased stochastic ozone days: analyses, solutions and beyond, Knowledge and Information Systems 14 (3) (2008) 299–326.

[2] P. BANERJEE, Kaggle dataset for ozone level prediction.
URL https://www.kaggle.com/datasets/prashant111/ozone-level-detection