**(7153CEM)**

Coursework

Big Data Analytics and Data Visualisation using Pyspark

**Module Leader: Dr. Marwan Fuad**

**Student Name: Guddu Kumar Shah**

**SID: 11402949**

**Analysing and predicting air quality standards in the USA**

I can confirm that all work submitted is my own: Yes

INTRODUCTION

This report demonstrates the big data management and data visualization module work. This report focuses on demonstrating the deep understanding of nature of big data including its advantages and drawbacks. The work also includes the identification and selection of different data analytical techniques for bid data analysis.

This work includes the application of Apache Spark. Apache Spark is basically a framework for data processing that processed very large datasets by distributing over multiple computers (Pointer, 2020). Apache Spark is a very popular open-source framework that performs big data processing with support of various general purpose, high level programming language like python and R. Here, Apache Spark is collaborated with python API to demonstrate the functionalities of Spark that harnesses the simplicity of the python and power of Spark to tackle big data problems (Chand, 2021). This report demonstrates how to set up Apache Spark with Python also called as pyspark for big data processing. It is generally termed as a Spark Python API that connects Resilient Distributed Datasets (RDD) with Apache Spark and Python.

The dataset name used for demonstration in this work is US Air Quality 1980-Present. This dataset is collected from Kaggle. The aim of this work is to analyse the air pollution and predict the quality of air on the scale from perfectly healthy to extremely hazardous in the US from 1980 to present. The work demonstration is structured as follows. First, setting up the Pyspark environment. Second is loading and processing the dataset using Pyspark. After that visualization of the data is done only using Tableau. Then, pre-processing of the data is done on the pyspark environment. Finally, identifying appropriate machine learning techniques to predict the category of air quality.

BACKGROUND AND IMPLEMENTATION

1. **PYSPARK INSTALLATION**

Pyspark installation was done in the jupyter notebook environment. The notebook was launched on the top of anaconda framework. The notebook also provides functionalities of some pre-defined libraries making harnessing its work to work with python interchangeably. The main components of Pyspark installation are Apache Spark, Java and those installation were done in a chronological order as depicted below in figures.

* The spark distribution was downloaded from <https://spark.apache.org> and the corresponding window is in ***Figure 1***.

Figure 1: Spark distribution download

Graphical user interface, text, application, email

Description automatically generated

* A hadoop binary file for windows named winutils.exe is downloaded from [GitHub repo](https://github.com/steveloughran/winutils/) for the corresponding version of spark distribution. The link to the winutils.exe is attached here, <https://github.com/steveloughran/winutils/tree/master/hadoop-3.0.0/bin>. The corresponding image is below in *Figure* 2.

Figure 2: winutils.exe download

Graphical user interface, text, application

Description automatically generated

* After that java jdk installation is done by downloading in the jdk file from Oracle website. The link to the fie is [https://www.oracle.com/java/technologies/downloads/#jdk18-windows](https://www.oracle.com/java/technologies/downloads/%23jdk18-windows). The corresponding image is in *Figure 3*.

Figure 3: Java jdk download

Graphical user interface, text, application, email

Description automatically generated

* The java jdk is installed after downloading and the .tgz file from spark distribution is unzipped using 7-zip and extracted in C drive for installation. The downloaded winutils.exe file is moved to the \bin folder of spark distribution as C:\spark-3.3.0-bin-hadoop3\bin\winutils.exe.
* The Environment variables and path are set for the spark distribution and java jdk to let the windows find the files during the launching of pyspark kernel.

The Environment variables name with their value are as follows:

|  |  |
| --- | --- |
| Variable | Value |
| JAVA\_HOME | C:\Program Files\Java\jdk1.8.0\_341 |
| SPARK\_HOME | C:\spark-3.3.0-bin-hadoop3 |
| HADOOP\_HOME | C:\spark-3.3.0-bin-hadoop3 |
| PATH | C:\Program Files\Java\jdk1.8.0\_341\bin |
| SPARK\_PATH | C:\spark-3.3.0-bin-hadoop3\bin |
| HADOOP\_PATH | C:\spark-3.3.0-bin-hadoop3\bin |
| PYSPARK\_DRIVER\_PYTHON | jupyter |
| PYSPARK\_DRIVER\_PYTHON\_OPTS | notebook |

* Finally the anaconda framework is downloaded from anaconda.com and installed. The anaconda prompt powershell is launched for installing the pyspark package and libraries using pip as shown *Figure 4* below.

Figure 4: Installing findspark library

Text

Description automatically generatedText

Description automatically generated

* Launching the jupyter notebook from anaconda from for checking the successful installation if the pyspark by running the following code as shown in *Figure 5*.

Figure 5: Checking Pyspark setup and installation

Graphical user interface, text, application

Description automatically generated

1. **DATASET DESCRIPTION**

The dataset used in this work is US Air Quality 1980-Present which is publicly accessible dataset at Kaggle platform. The dataset consists of air pollution matters over 14 timezones in the USA from 2015 to 2022. In this dataset series of tasks will be performed from data pre-processing to visualization to application of machine learning techniques for model building.

All in all the dataset contains 1048575 rows and 15 features/columns. The description for each features of the dataset is presented in the ***Table 1*** below.

**Table 1: Features decsriptions**

|  |  |  |
| --- | --- | --- |
| Features | Description | Datatypes |
| Index | Index | Numerical |
| **CBSA Code** | Unique geographical area code | Numerical |
| **Date** | Date of measurement | Date |
| **AQI** | Average Air Quality Index | Numerical |
| **Category** | Air quality from good to hazardous | Categorical |
| **Defining Parameter** | PM2.5,PM10,Ozone,Carbon Monoxide, Sulpher Dioxide, Nitrogen Dioxide concentration | Categorical |
| **Number of Sites Reporting** | Number of stations where measurement was taken | Numerical |
| **city\_ascii** | City names | Categorical |
| **state\_id** | State ID abbreviation | Categorical |
| **state\_name** | State names | Categorical |
| **lat** | Latitude of the station | Numerical |
| **lng** | Longitude of the station | Numerical |
| **population** | Population where measurement was taken | Numerical |
| **density** | Density per square km where measurement was taken | Numerical |
| **timezone** | Timezone where the measurement was taken | Categorical |

Table 2 describes the Air Quality Index Distribution on the scale of 0-500

Table 2: Air Quality Index Distribution

A picture containing text

Description automatically generated

The Air Quality Index is basically used to measure the severity of the air pollution in the environment by developing strategies by several environmental specialists (Monteiro et. al 2017). The Air Quality Index is specifically divided into 6 different varieties (Good, Moderate, Unhealthy for Sensitive Groups, Unhealthy, Very Unhealthy and Hazardous) that measures and defines the pollutant in the air to make people conscious of the arising consequences (Mirabelli et. al, 2020). The major factors contributing the air pollution are the increasing concentration of the pollutants like Nitrogen dioxide, carbon monoxide, sulpher oxide, ozone, PM2.5 and PM 10. The continuous increasing of these factors are in the verge of causing several chronic as well as lethal health issues in among all the age groups. Because of these factors and the corresponding increasing risks regarding the health it is extremely important to categorize the air quality so that further positive actions can be withheld to improve the air quality standards.

1. **IMPLEMENTATION**

In this section of the work, several data analysis tasks like loading the dataset, pre-processing, features engineering, features selection are performed prior to the model building. All the figures are created using Tableau.

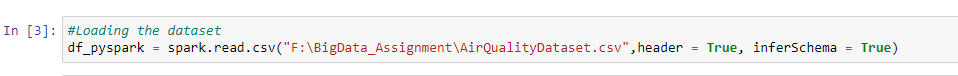
* **Loading the Dataset**

In this work, Spark SQL module helps in increasing the computation. After the SparkSession is created the csv file containing this dataset loaded using read.csv in the pyspark environment. Below Figure 6and Figure 7 represents the sparksession creation and loading the csv file in jupyter notebook.

Figure 6: Creating Sparksession



Figure 7: Loading the dataset



Further in order to remove the messiness of the structure of the pyspark dataframe the following lines of code was written as shown below.

Graphical user interface, text, application

Description automatically generated

Figure 8: Dataset with features

Table

Description automatically generated

Table

Description automatically generated

A picture containing graphical user interface

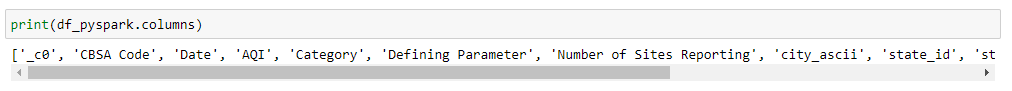
Description automatically generated

In *Figure* 8, the top 5 rows with all the 15 features are shown. The presence of data indicates it as a time series dataset. It includes combinations of categorical and numerical values which is defined shortly in *Table* ***1***. The defining parameters are basically the pollutant PM2.5, PM10, Ozone, CO, NO2, SO2. The category is the class which categorizes the air quality based on the air quality index.

* **Data Pre-processing**

Data pre-processing is basically a technique in which series of operations like addition, deletion and transformation of the training set of data is done. It’s a very important tasks to be performed before applying machine learning technique for model building (Kuhn & Johnson, 2013). During this process the dataset is checked for missing values, skewness and outliers. The names of some of the features are renamed because of its difficulty in reading the data. The feature renaming task is done as steps in *Figure* 9 below.

Figure 9: Steps renaming the features

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

*Figure* 10 represents the total number of records/rows in the dataset, *Figure* 11prints the schema for each of the features and *Figure* 12 describes the dataset. It indicates that there is no missing values as the counts for all the features are same and it also indicates absence of errors or negative readings since there are no negative values . *Figure 13* is checking the missing values for confirmation.

Figure 10: Total records in the dataset

Rectangle

Description automatically generated with medium confidence

Figure 11: Schema of the features

Text

Description automatically generated

Figure : Describe the dataset

Text

Description automatically generated Text

Description automatically generated

Figure : Checking for missing data

Text

Description automatically generated with medium confidenceA picture containing chart

Description automatically generated

The Figure 14 shows the distinct numbers of the timezone where the measurements were taken while and Figure 15 depicts the names along with the number of measurements taken in each timezone.

Figure 14: Number of unique timezone

Graphical user interface, text, application

Description automatically generated

Figure 15:Unique timezone with number of measurements taken

Text, table

Description automatically generated with medium confidence

Figure 16 visualizes the names of and number of measurements taken in each timezone.

Figure : Names of timezone with total number of measurements. Figure created using Tableau

Chart

Description automatically generated

The total distinct category of air quality along with the number of its corresponding times recorded from 2015 to 2022 is shown in the *Figure 17* below and the equivalent visualization is shown in *Figure 18,*. The data indicated that the quality of air over past 7 years in the US was good.

Figure 17: Category with number of recordings

Graphical user interface, text, application

Description automatically generated

Figure 18: Category distribution per counts. Figure created using Tableau

A picture containing chart

Description automatically generated

*Figure 19* contributes the conceteration of the pollutants over the period in determining the air quality. The ozone shows the highest concentrations towards deteriorating the quality of air. *Figure 20 shows corresponding visualization.*

Figure 19: Concentration of pollutants

Graphical user interface, text, application, email

Description automatically generated

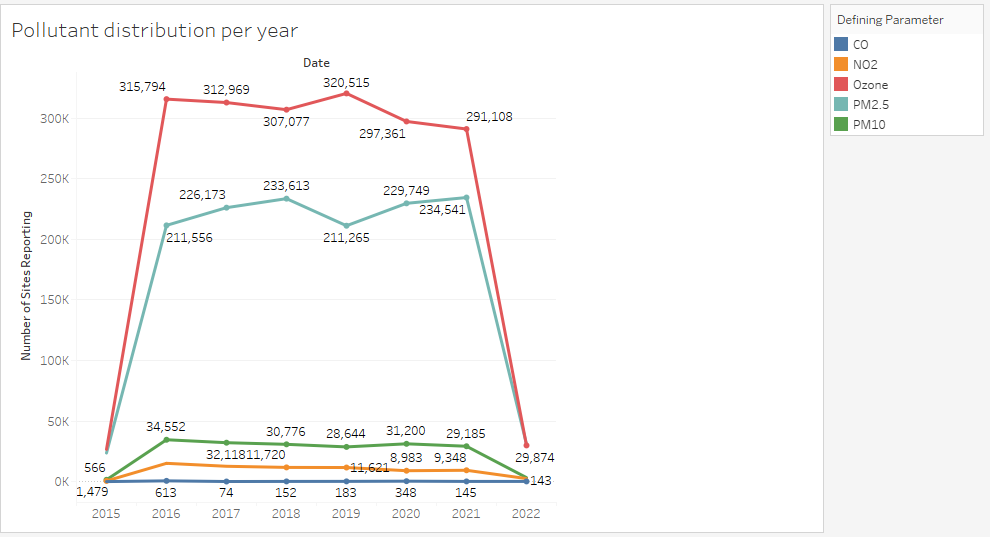
Figure : Pollutants recorded over measurement taken. Figure created using Tableau

Chart, bar chart

Description automatically generated

*Figure 21* shows pollutant distribution per year from 2015 to 2022. This clearly demonstrates that ozone is the skyrocketed during the first one year of investigation, remained stagnant with some fluctuations from 2016 to 2021 and drastically fell down. The second pollutant is PM2.5 which impacts a lot after ozone.

Figure 21: Pollutant distribution per year



The F*igure 22* shows that there are 482 distinct cities in the United States where the measurements were taken which comes under 52 states in F*igure 23*. The city names and the state names along with the number of times the measurements were recorded are shown in Figure 24 and Figure 25.

Figure 22: Number of cities

Graphical user interface, text, application

Description automatically generated

Figure 23: Number of states

Graphical user interface, text, application

Description automatically generated

Figure 24: City names with number of measurements. Figure created using Tableau

Graphical user interface, text, application

Description automatically generated

Figure 25: State names with number of measurements. Figure created using Tableau

Text

Description automatically generated with medium confidence

* **Feature Engineering**

The date feature in the dataset represents date, so the types casting is done here by converting the categorical type to date type using to\_date function in the Pyspark environment. Similarly Index and CBSA\_Code are converted to numerical type to categorical type, this is because these data are used for identification and make the libraries not to apply mathematical or statistical calculations. In this the type casting is done using StringType method. The casting for date is demonstrated in *Figure 26* while the rest in *Figure 26*.

Figure 26: Casting date from categorical to date type

Text

Description automatically generated

Figure 27: Casting index and CBSA\_Code from numerical to categorical type

Graphical user interface, text, application

Description automatically generated

Figure 28 shows checking the schema after type casting

Figure 28: Schema of features after type casting

Text

Description automatically generated

Feature encoding is basically the procedure to convert the categorical data into numerical for the application to the machine learning algorithms, this generally improves the accuracy of the model (Jackson & Agrawal, 2019). Jackson & Agrawal (2019) also says that scientists and data specialists have used several approaches for data transformation from text to number, some of popularly known approaches are label encoding, one hot encoding and binary encoding. In this work, label encoding is done on Defining\_Parameter and Category features, shown in *Figure 29* and *Figure 30* shows the corresponding number of the labels (number starts from 0) in these features.

Figure 29: Label Encoding

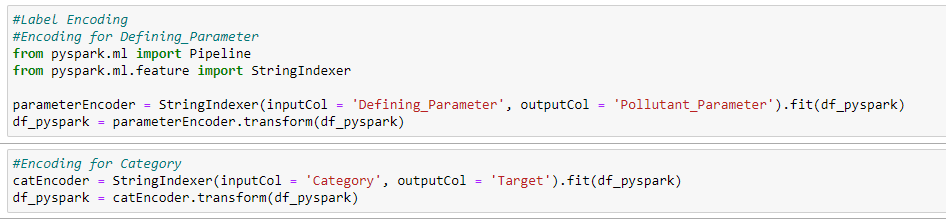


Figure 30: Getting labels

Graphical user interface, text, application, email

Description automatically generated

After encoding the date feature is break down into year, month and day for analysis since this is a time series dataset. *Figure 31* shows code for breaking the date and *Figure 32* shows the dataset after encoding and date manipulation. The schema is shown as in *Figure 33*.

Figure 31: Breaking date

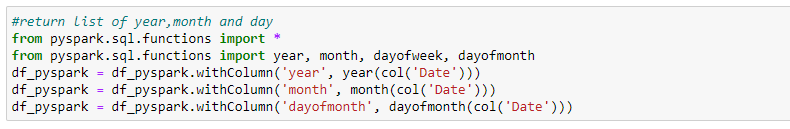


Figure 32: showing the dataset

Graphical user interface, application, table

Description automatically generatedGraphical user interface, application, table

Description automatically generated

Figure 33: Schema after data manipulation

Text

Description automatically generated

* **Feature Selection**

In this section of the work, feature selection of some of the most prominent features are done prior to building the model. This can be achieved by importing the VectorAssembler as shown in Figure 34 and result in Figure 35.

Figure 34: Feature Selection

Graphical user interface, text, application

Description automatically generated

Figure 35: Output of selected features

Graphical user interface, application

Description automatically generated

Fig shows the number of years from which the data collection is done in this dataset and figures shows the year, month and dayofmonth with the total number of the times the measurements were taken.

Figure 36: Years of data collected

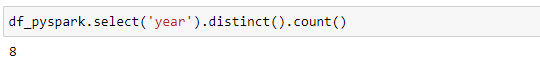


Figure 37: Year, Month and Day with their counts

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

EXPERIMENTAL SECTION

In this section of the work, dataset splitting is done followed by application of three machine learning classification techniques popularly known as Decision Tree classifier, Random Forest Classifier and Logistic Regression are used for classification of the air quality. The 3 classification techniques are used for training the date and the validation check is done on the testing set for checking the accuracy of the model. The final result of this experiment is to find out the most efficient model for predicting the category(class ) of the air quality based on the features selected by comparing the accuracies of all the classifiers.

Since this is a time series problem, this can be solved by simply flattening the time series, the dataset is split into train set and test set based on date, 2015 to 2019 are set for training which is used to predict the category of air quality from 2020 to 2022. Fig shows splitting of the dataset based on date and fig shows the total number of train-test counts.

Figure 38: Splitting the dataset

Text

Description automatically generated with medium confidence

Figure 39: Train-Test Counts

A picture containing rectangle

Description automatically generated Graphical user interface, application

Description automatically generated with medium confidence

**Machine Learning**

After data splitting the machine learning classification techniques are applied on the train set a set of libraries are imported for machine learning modelling are shown in *Figure 40.*

Figure 40: Set of libraries

Text

Description automatically generated

* Decision Tree Classifier

Decision tree classifier is a very popularly known method for classifying data, this method has the ability to change complicated problems into a simple task by traversing through the set of roots and nodes making the solution easy to understand and interpret (Priyanka & Kumar, 2020). Fig shows applying the classifier to the features extracted during feature selection and after that the model is trained by using fit method.

Figure 41: Applyign DTC and training the model

Graphical user interface, text, application, email

Description automatically generated

Figure 42 shows DTC model testing on test set and fig prints the output after transformation of test set.

Figure 42: model testing



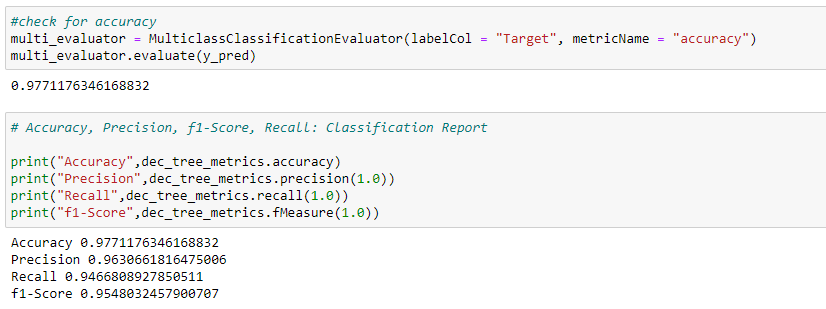
Figure 43: Output

Table

Description automatically generated

Model Evaluation is done using MulticlassClassificationEvaluator and checked for accuracy, precision, f1score and recall in Figure 44. The confusion-matrix is at Appendix C.

Figure 44: Accuracy, precision, f1score and recall

.

* Random Forest Classifier

The model building task is explained in Appendix D. Figure 45 shows the accuracy, precision, f1score and recall obtained by the model.

Figure 45: Accuracy, precision, f1score and recall

Text

Description automatically generated

* Logistic Regression

The classification operations are presented in the Appendix E. Figure 46 shows the accuracy, precision, f1score and recall obtained by the model.

Figure 46: Accuracy, precision, f1score and recall

Graphical user interface, text, application, chat or text message

Description automatically generated

**Results**

After building the machine learning models it was found that all the 3 classification algorithms performed great, but logistic regression holds an upper hand giving 99.97% accuracy and 100% recall. This score confirms that the logistic regression classifier is best for air quality prediction. Table 3 represents score summary.

Table 3: Score Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ML Classifiers | Accuracy | Precision | Recall | F1-score |
| Decision Tree | 0.9771 | 0.9630 | 0.9466 | 0.9548 |
| Random Forest | 0.9807 | 1.0 | 0.9078 | 0.9517 |
| Logistic Regression | 0.9997 | 1.0 | 1.0 | 1.0 |

DISCUSSION

The dataset used in this work was already cleaned and non-presence of any null or missing data, this made the data analysis task much easier. Also, there were no any negative readings obtained during the measurements taken. During analysis it was found that although ozone had the maximum concentration in air quality amongst all pollutants the air quality was still good over the years. Moreover, the measurements taken shown 63.11% of good quality air data. Through visualization it was found that New York had the highest population as well density per square kilometer. On average May is the month where ozone concentration was on peak. It was also observed that maximum states and cities air had PM2.5 concentration from 2015-22. California is the city where highest number of AQI data was recorded.

Overall, all three models were great in analyzing and prediction making showing the satisfactory result. Logistic Regression took more duration in building the model than the rest. Hence, the output obtained was excellent.

CONCLUSION

In conclusion, data analysis of the air quality dataset were performed in pyspark providing great fusion for Python with spark easily. Later, the simplicity of splitting the data and choosing machine learning model were easy to implement. The visualizations are done using tableau. This further provide a transparent image of the dataset into graphical format. Logistic Regression is the best technique for analyzing for this type of time series dataset because of its accuracy.

The data analysis source code is attached in Appendix.

REFERENCES

Chand, s. (2021). PySpark Programming – Integrating Speed With Simplicity. *Edureka!*

Jackson, E., & Agrawal, R. (2019, April). Performance evaluation of different feature encoding schemes on cybersecurity logs. In *2019 SoutheastCon* (pp. 1-9). IEEE.

Kuhn, M., & Johnson, K. (2013). Data pre-processing. In *Applied predictive modeling* (pp. 27-59). Springer, New York, NY.

Mirabelli, M. C., Ebelt, S., & Damon, S. A. (2020). Air Quality Index and air quality awareness among adults in the United States. *Environmental research*, *183*, 109185. <https://doi.org/10.1016/j.envres.2020.109185>

Monteiro, A., Vieira, M., Gama, C., & Miranda, A. I. (2017). Towards an improved air quality index. *Air Quality, Atmosphere & Health*, *10*(4), 447-455.

Pointer, I. (2020). What is Apache Spark? The big data platform that crushed Hadoop. *InfoWorld*, *16*.

Priyanka, & Kumar, D. (2020). Decision tree classifier: A detailed survey. *International Journal of Information and Decision Sciences*, *12*(3), 246-269.

APPENDIX

1. Dataset Link:

[https://www.kaggle.com/datasets/calebreigada/us-air-quality-1980present](%20%20https:/www.kaggle.com/datasets/calebreigada/us-air-quality-1980present)

1. Source code Link:

<https://drive.google.com/file/d/1Q4iGtEKe-8OK481b2EZeKHT1RLR0QU70/view?usp=sharing>

1. Confusion-matrix for Decision Tree Classifier.

Text

Description automatically generatedGraphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. Random Forest Classifier

* Model building

Graphical user interface, text, application, email

Description automatically generated

* Output from the y\_pred\_rf

Table

Description automatically generated

* Confusion-matrix Random forest Classifier

A picture containing calendar

Description automatically generated

1. Logistic Regression

* Model building

Graphical user interface, text, application, email

Description automatically generated

* Output from the y\_pred

Table

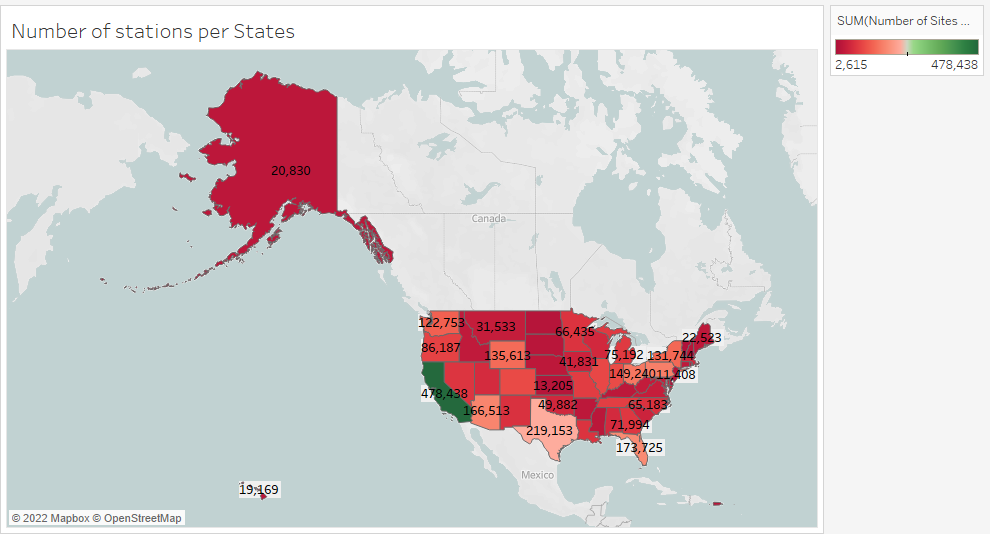
Description automatically generated

* Confusion-matrix Logistic Regression Classifier

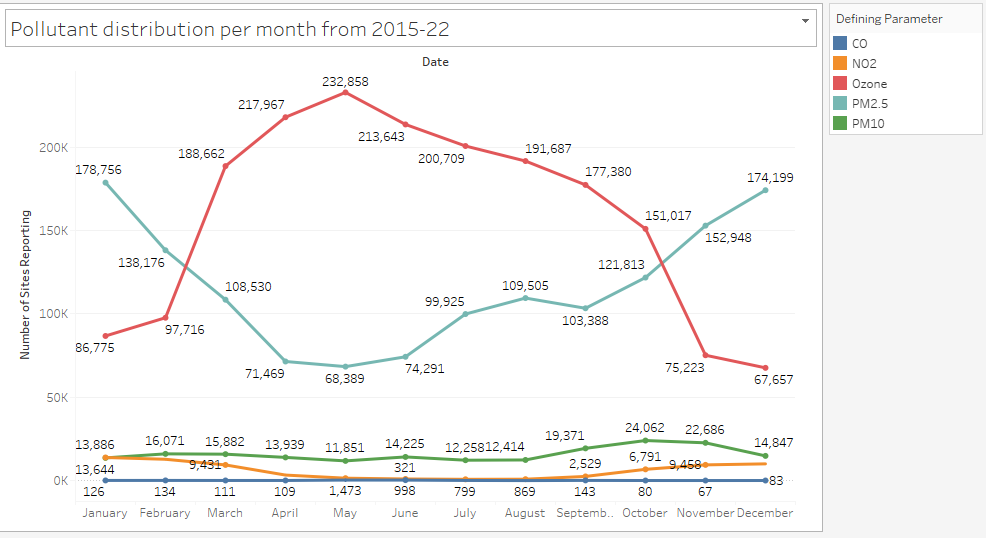
Calendar

Description automatically generated

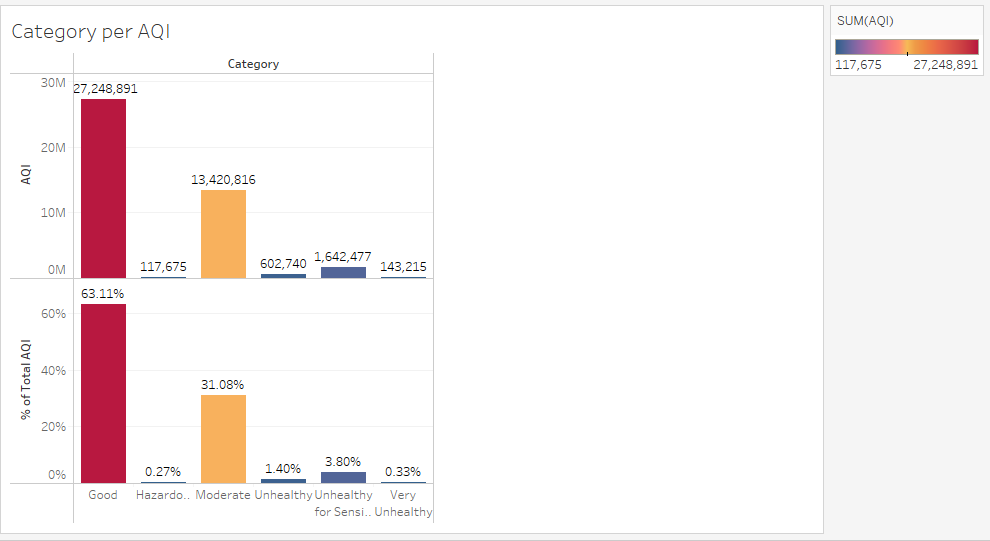
1. Number of stations where the measurements were taken.



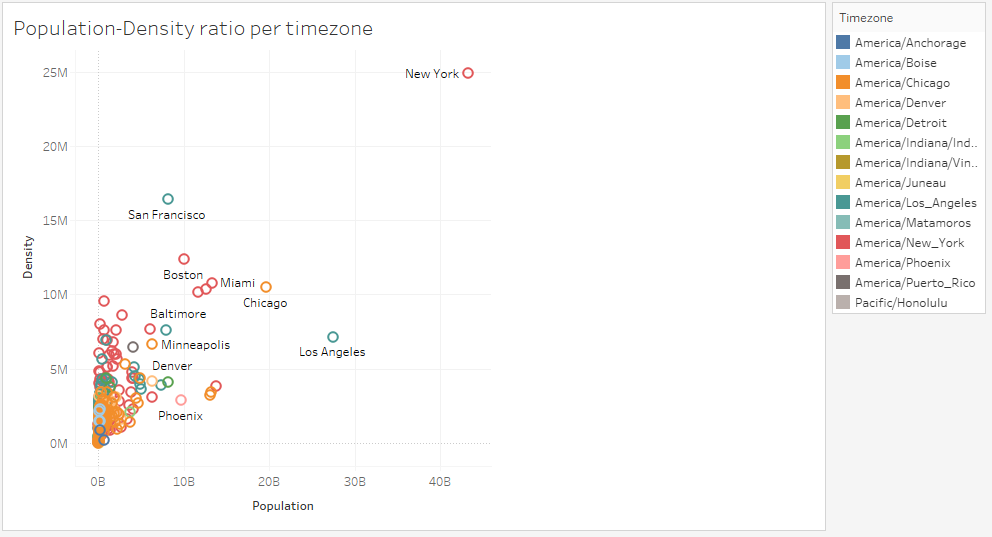
1. Pollutant(CO, SO2, Ozone, PM2.5, PM10) concenteration in air per month from 2015-2022



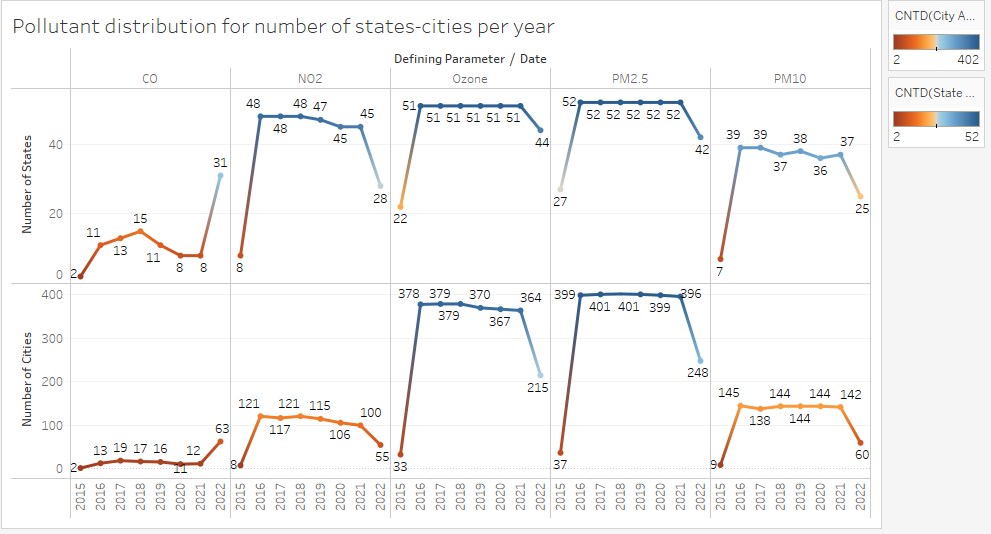
1. Category/class translation from the AQI measurements taken.



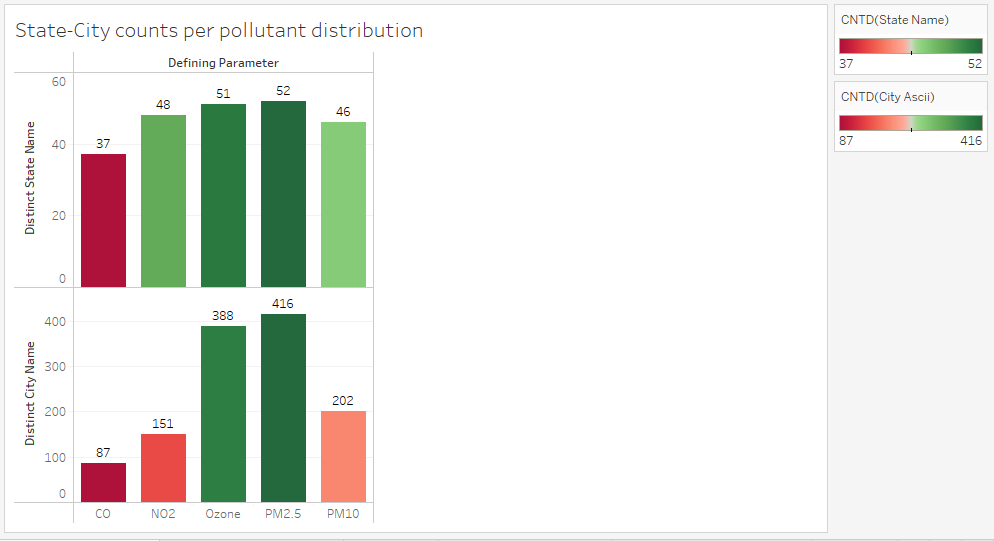
1. Population and density per square kilometer ratio according to the 14 timezone.



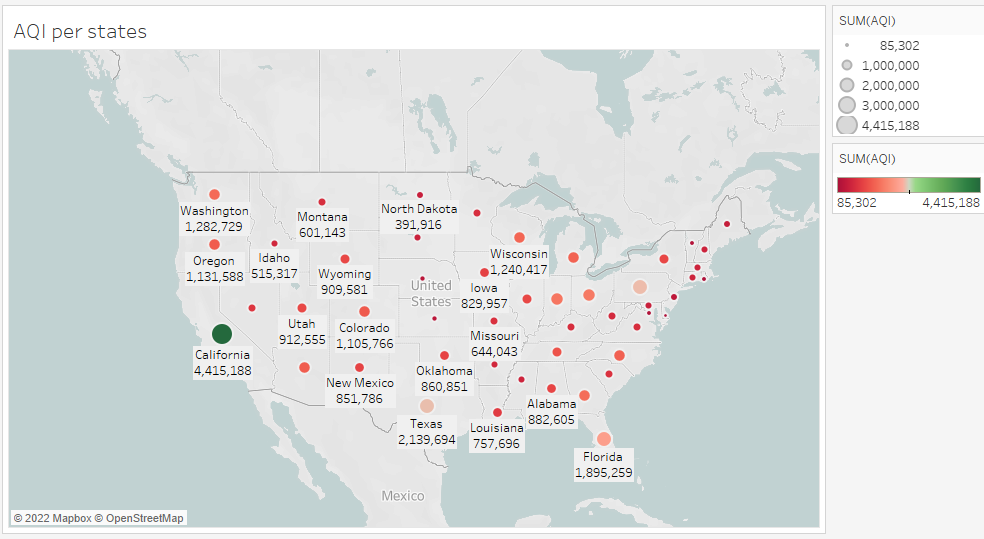
1. Pollutant distribution for every year over every states and cities in the US.



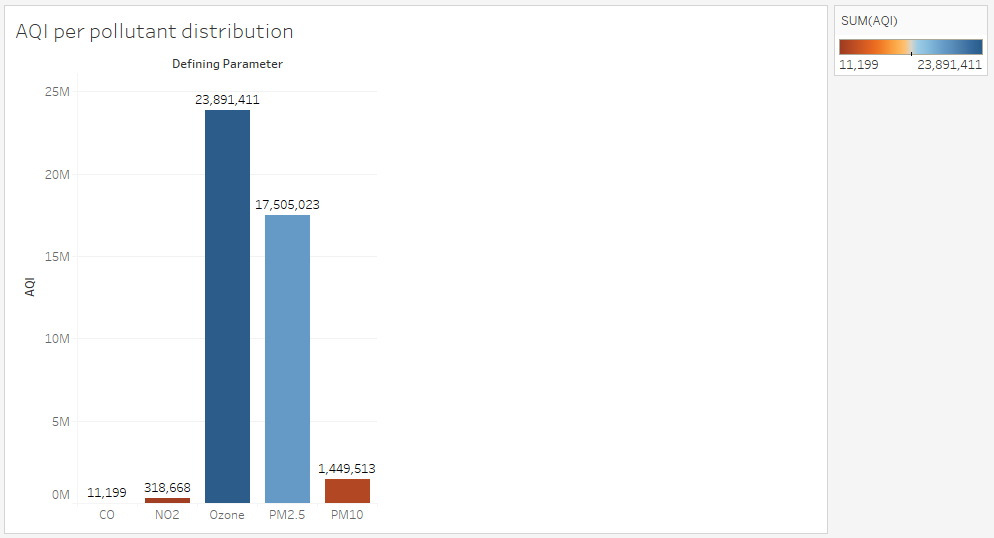
1. Number of cities and states per pollutant distribution



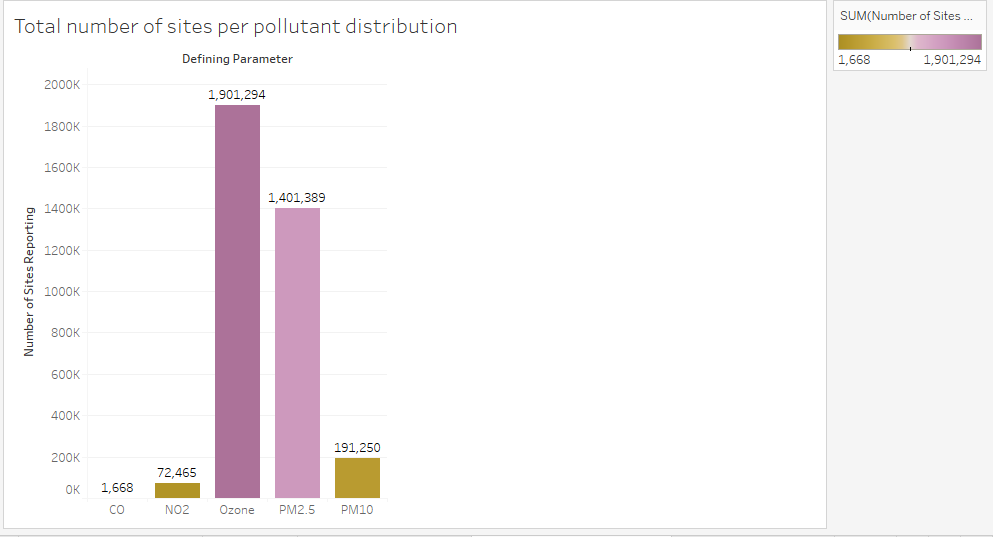
1. Air Quality Index distribution over states in the US.



1. Number of AQI measurements considering particular pollutant.



1. Finding out the number of sites from where measurements were taken considering particular pollutant.



1. Finding the total sum of AQI considering number of sites per states

Chart, treemap chart

Description automatically generated