Business Analytics - Data Science

Building a Machine Learning Model Using Regression as an Additional Feature Selection Metric

Student: Amanda Kimie Miyano

June 15th 2023

Word Count: 3621

Table of Contents

1. Introduction and Rationale (Business Context)

This research endeavor aims to develop a machine-learning model that incorporates regression as an added feature selection metric. I will utilize the Air Quality dataset from the UCI Machine Learning repository, which comprises 9471 hourly averaged responses from 5 metal oxide chemical sensors present in an Air Quality Chemical Multi-Sensor Device. This device was placed at road level in an Italian city, a heavily polluted area. The data was collected over a year, from March 2004 to February 2005 (UCI, 2016).

The attribute information is as follows:

- PT08.S1(CO)-Tin oxide hourly averaged sensor response (nominally CO targeted).
- C6H6(GT)-True hourly averaged Benzene concentration in microg/m^3 (reference analyzer).
- PT08.S2(NMHC)-Titania hourly averaged sensor response (nominally NMHC targeted).
- NOx(GT)-True hourly averaged NOx concentration in ppb (reference analyzer)
- PT08.S3(NOx)-Tungsten oxide hourly averaged sensor response (nominally NOx targeted).
- NO2(GT)-True hourly averaged NO2 concentration in microg/m^3 (reference analyzer).
- PT08.S4 (tungsten oxide)-hourly averaged sensor response (nominally NO2 targeted).
- PT08.S5 (indium oxide)-hourly averaged sensor response (nominally O3 targeted).
- Temperature-Temperature in Â°C.
- Humidity-Relative Humidity (%).
- AH-Absolute Humidity.

Air quality classification as 'Healthy' or 'Unhealthy' can be determined by considering the role of air temperature in the dispersion of pollutants. Temperature is a crucial factor affecting pollutants' movement, mainly through convection. When the temperature is high, pollutants tend to move upward in a more vertical direction, leading to the efficient entrainment of localized pollutants from lower to higher levels. In contrast, lower temperatures cause thermally induced vertical movements, which cause

pollutants to remain at lower levels (Education, nd). Therefore, analyzing air temperature is essential to accurately determine the air quality classification.

The results of this analysis hold significant potential for businesses seeking to gain valuable insights and make informed decisions based on data-driven research. The study of air quality data is of utmost importance for companies to evaluate their environmental impact by utilizing machine learning models to analyze the relationship between various atmospheric features and identify the pollutants that have the most significant impact on air quality. Companies can use the results to recognize areas where their activities may contribute to poor air quality—enabling them to take the necessary measures to mitigate harmful influences and adhere to health and safety regulations, thus safeguarding the well-being of their employees and society. Additionally, businesses can rely on this data to consider their environmental sustainability efforts and develop strategies to reduce emissions and enhance energy efficiency. Finally, companies operating in the air cleansing, pollution monitoring, or environmental protection sectors can leverage air quality data to discern market demand, prospective customer segments, and optimal marketing approaches.

2. Data Preprocessing

(a) Loading Dataset & Dataset Exploration

In order to begin our analysis, I downloaded a compressed file in the ZIP format containing a data set and unzipped the included CSV file. Then I transformed the data set into a data frame and initially explored the data utilizing the df.head and df.shape commands. Our examination revealed that the data set consists of 9471 rows and 17 columns. I also used the df.info command and identified that the data set contains two distinct data types, namely objects and floats. It was observed that certain columns within the dataset exhibited many missing values, which were primarily bounded to the columns classified as

object data type. Consequently, it is necessary to convert these columns to an appropriate numeric type before proceeding with any further analysis.

First 10 rows of the Air Quality Dataset

index	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH	Unnamed: 15	Unnamed: 16
0	10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	166.0	1056.0	113.0	1692.0	1268.0	13,6	48,9	0,7578	NaN	NaN
1	10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	103.0	1174.0	92.0	1559.0	972.0	13,3	47,7	0,7255	NaN	NaN
2	10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	131.0	1140.0	114.0	1555.0	1074.0	11,9	54,0	0,7502	NaN	NaN
3	10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	172.0	1092.0	122.0	1584.0	1203.0	11,0	60,0	0,7867	NaN	NaN
4	10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	131.0	1205.0	116.0	1490.0	1110.0	11,2	59,6	0,7888	NaN	NaN
5	10/03/2004	23.00.00	1,2	1197.0	38.0	4,7	750.0	89.0	1337.0	96.0	1393.0	949.0	11,2	59,2	0,7848	NaN	NaN
6	11/03/2004	00.00.00	1,2	1185.0	31.0	3,6	690.0	62.0	1462.0	77.0	1333.0	733.0	11,3	56,8	0,7603	NaN	NaN
7	11/03/2004	01.00.00	1	1136.0	31.0	3,3	672.0	62.0	1453.0	76.0	1333.0	730.0	10,7	60,0	0,7702	NaN	NaN
8	11/03/2004	02.00.00	0,9	1094.0	24.0	2,3	609.0	45.0	1579.0	60.0	1276.0	620.0	10,7	59,7	0,7648	NaN	NaN
9	11/03/2004	03.00.00	0,6	1010.0	19.0	1,7	561.0	-200.0	1705.0	-200.0	1235.0	501.0	10,3	60,2	0,7517	NaN	NaN

(b) Data Cleaning

Data Type Conversion

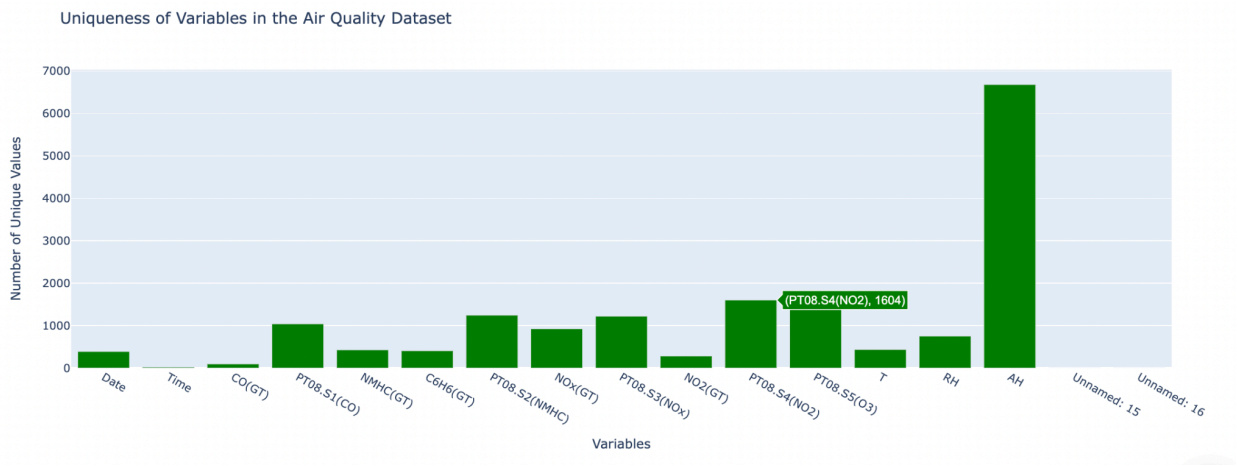
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9471 entries, 0 to 9470
Data columns (total 17 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Date            9357 non-null   object
 1   Time            9357 non-null   object
 2   CO(GT)          9357 non-null   object
 3   PT08.S1(CO)     9357 non-null   float64
 4   NMHC(GT)        9357 non-null   float64
 5   C6H6(GT)        9357 non-null   object
 6   PT08.S2(NMHC)   9357 non-null   float64
 7   NOx(GT)         9357 non-null   float64
 8   PT08.S3(NOx)    9357 non-null   float64
 9   NO2(GT)         9357 non-null   float64
 10  PT08.S4(NO2)    9357 non-null   float64
 11  PT08.S5(O3)     9357 non-null   float64
 12  T               9357 non-null   object
 13  RH              9357 non-null   object
 14  AH              9357 non-null   object
 15  Unnamed: 15     0 non-null      float64
 16  Unnamed: 16     0 non-null      float64
dtypes: float64(10), object(7)
```

```
Date             datetime64[ns]
Time             datetime64[ns]
CO(GT)                  float64
PT08.S1(CO)             float64
NMHC(GT)                float64
C6H6(GT)                float64
PT08.S2(NMHC)           float64
NOx(GT)                 float64
PT08.S3(NOx)            float64
NO2(GT)                 float64
PT08.S4(NO2)            float64
PT08.S5(O3)             float64
T                       float64
RH                      float64
AH                      float64
Unnamed: 15             float64
Unnamed: 16             float64
```

To effectively work with the 'Date' and 'Time' columns present in our dataset, we needed to convert them from their initial string format to the DateTime' data type. This conversion was deemed necessary for any datetime-related operations or analysis that must be performed on these columns.

On the other hand, the additional columns, such as 'CO(GT)', 'C6H6(GT)', 'T', 'RH', and 'AH' consisted of numeric values represented as strings with commas serving as decimal separators. To make possible mathematical calculations or application of machine learning algorithms on these columns, we had to convert them to the numeric data type by eliminating the commas.

(c) Checking for Redundant Data



To assess the presence of redundant variables in the dataset, I conducted an analysis of uniqueness using Plotly. The results were presented in the form of a bar graph that displayed each variable's uniqueness level. I observed that the columns identified as "Unnamed: 15" and "Unnamed: 16" had a count of zero unique values, indicating that they may be devoid of useful information or could be empty. Therefore, I recognized the need to conduct further investigations to identify any missing values in the dataset.

(d) Missing Values

Based on the heatmap, it appears that there are a total of 114 missing values present in the following columns: 'Date', 'Time', 'CO(GT)', 'PT08.S1(CO)', 'NMHC(GT)', 'C6H6(GT)', 'PT08.S2(NMHC)', 'NOx(GT)', 'PT08.S3(NOx)', 'NO2(GT)', 'PT08.S4(NO2)', 'PT08.S5(O3)', 'T', 'RH', and 'AH'. Additionally, it has been observed that the columns 'Unnamed: 15' and 'Unnamed: 16' have 9471 missing values each, indicating that these specific columns contain empty values for all rows present in the dataset. Therefore, to ensure accuracy, I eliminated two unnecessary columns and any instances of missing data from the dataset.

(e) Scaling

To guarantee the precision of our results, I found it necessary to adjust the dataset by scaling the data. This was a crucial step due to the presence of features with varying numerical ranges within the dataset. For instance, the 'CO(GT)', 'PT08.S1(CO)', and 'T'. These features have different scales and units of measure. Specifically, 'CO(GT)' values range from 1.6 to 2.6, 'PT08.S1(CO)' values range from 1272 to 1402, and 'T' values range from 11.0 to 13.6. If I had not scaled these features, our model might have inaccurately assumed that 'PT08.S1(CO)' had a greater impact on the target variable than 'CO(GT)' or 'T' due to its relatively higher numerical values. Therefore, scaling the dataset was necessary to ensure our results' accuracy and reliability.

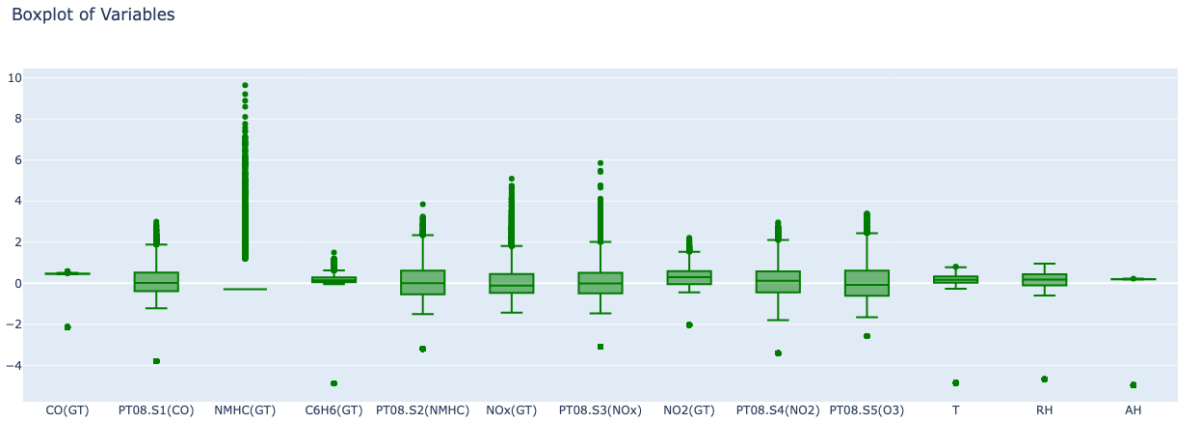
Scaled dataframe (partial) Table

index	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)
0	0.4739998974535257	0.9429827612549412	2.2112355435871613	0.24250369088710735	0.4422965461502871	-0.010166149235445109	0.8106490932282691	0.43212432665559714
1	0.46627321797951937	0.73680662439452	1.9393829266152163	0.18208510393822022	0.17645949228271268	-0.25490227366684787	1.177135643526273	0.26668359145690945
2	0.46884877780418815	1.070326845786378	1.7676865369487245	0.17241813002639825	0.12971891138291938	-0.14613066280844667	1.071537823948882	0.4400024569031537
3	0.46884877780418815	0.9914947934573933	1.710454407059894	0.1772516169823092	0.15601048813905313	0.013142053091355157	0.9224585492513889	0.5030274988836061
4	0.4611220983301818	0.6761665841414549	1.5029879362128833	0.11199954307751112	-0.1711735781595	-0.14613066280844667	1.2734160084350705	0.4557587173982668
5	0.45597097868084424	0.4487664331924609	1.4099857251435337	0.06849816047431237	-0.42240420049588895	-0.3092880790960485	1.6833840138531766	0.2981961124471357
6	0.45597097868084424	0.41238240904062184	1.359907611490807	0.041913982216802016	-0.5976813788701139	-0.4141749895666497	2.0716112917112315	0.14851163774356108
7	0.4533954188561754	0.26381431042061243	1.359907611490807	0.03466375178293555	-0.6502645323823814	-0.4141749895666497	2.0436589277054518	0.14063350749600453
8	0.452107638943841	0.1364702258891758	1.3098294978380802	0.01049631700338069	-0.8343055696753174	-0.48021489615925045	2.434992023786371	0.014583423535099614
9	0.44824429920683784	-0.1182179431736975	1.2740594166575612	-0.0040041438643522225	-0.9745273123746974	-1.4319664911702614	2.8263251198672905	-2.033730440829605

(f) Outlier Analysis

I utilized Plotly's boxplot function to detect the presence of any outliers across all columns in our data. This provided us with insight into the variables that exhibited the greatest number of outliers. I subsequently conducted an individual inspection of each column, removing any outliers I identified. This involved the computation of the 25th and 75th percentiles and the interquartile range (IQR). Then I established upper and lower boundaries for outliers, and any data that exceeded these limits were removed.

Before removing outliers



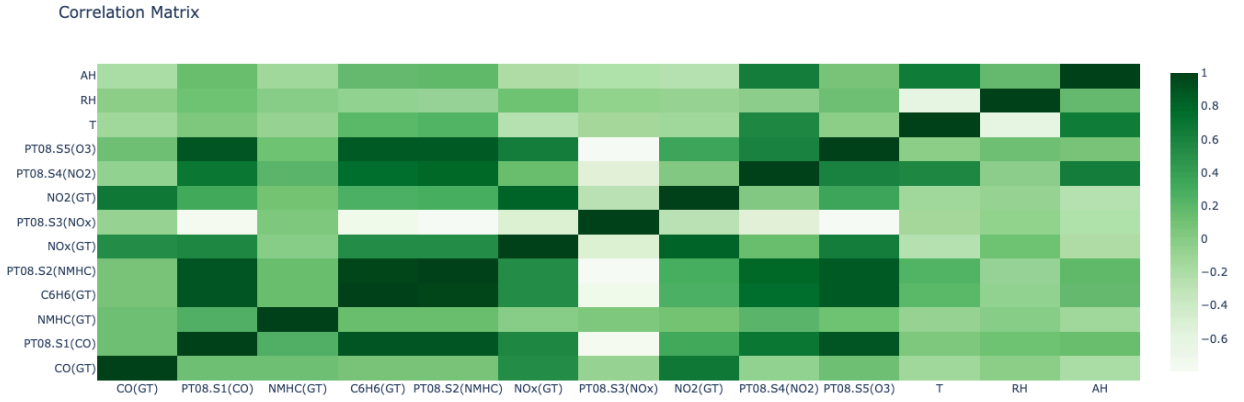
Following the removal of outliers from our dataset, I conducted a thorough check to ensure that the resulting data would not negatively impact our analysis. Specifically, I calculated the percentage decrease between the initial dataset size of 9471 and the final size of 8990, using the formula:

Percentage Decrease = ((Initial Value - Final Value) / Initial Value) * 100

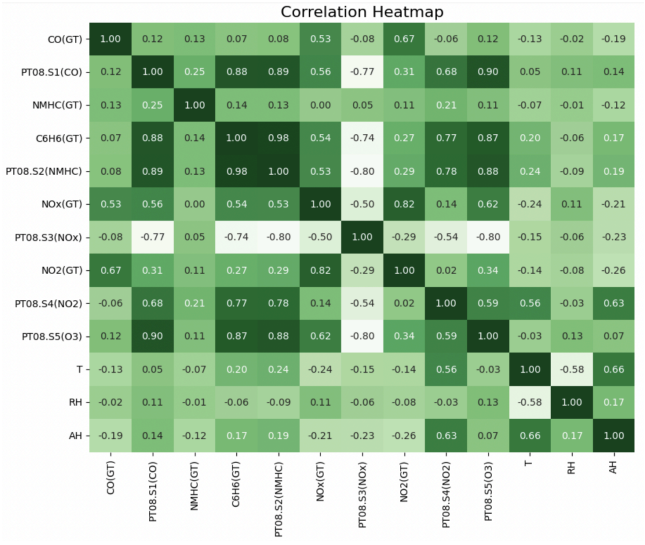
Substituting the relevant values into the formula, I determined that the percentage decrease from 9471 to 8990 was approximately 5.08%. This decrease is deemed insignificant and does not compromise the integrity of our results.

3. Exploratory data analysis

(a) Bivariate Analysis - Correlation Matrix

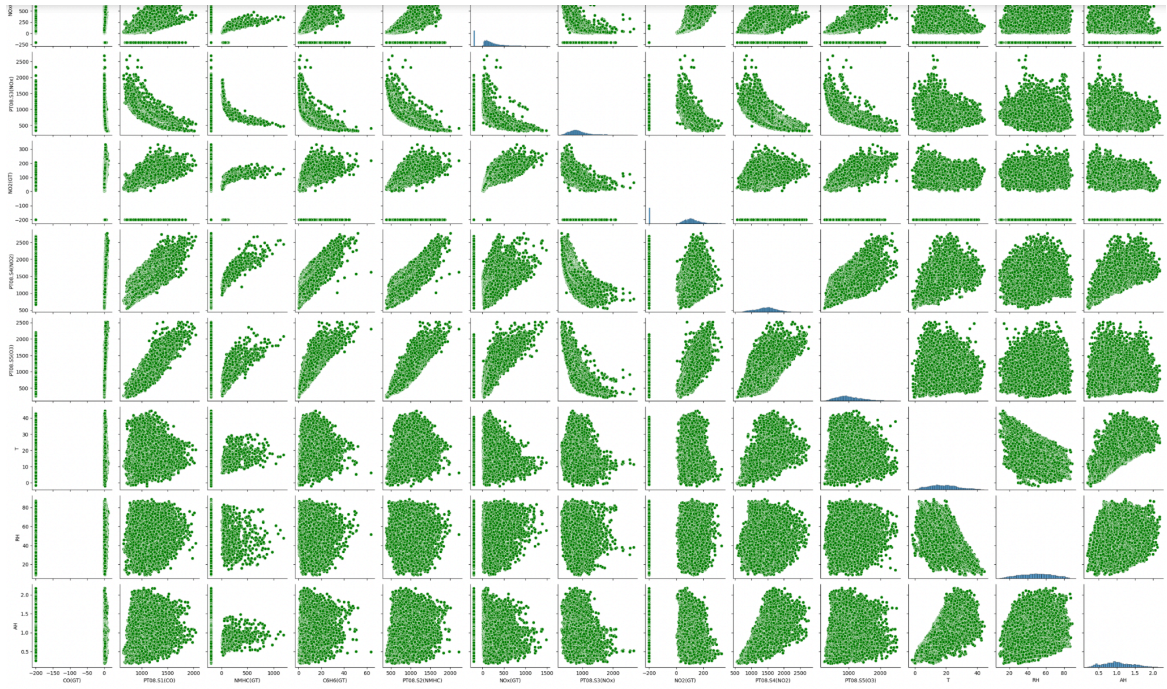


*The graph is interactive, however, it does not work on the google document



Using a correlation matrix and pyplot, I could effectively select features and gain valuable insight into the interaction between variables. Our findings indicate a significant positive correlation between 'PT08.S2(NMHC)' (Titania hourly averaged sensor response) and 'C6H6(GT)' (True hourly averaged Benzene concentration). Additionally, I observed weaker correlations between absolute Humidity and carbon monoxide, as well as a negative correlation between temperature and relative Humidity.

Pair Plot Graph



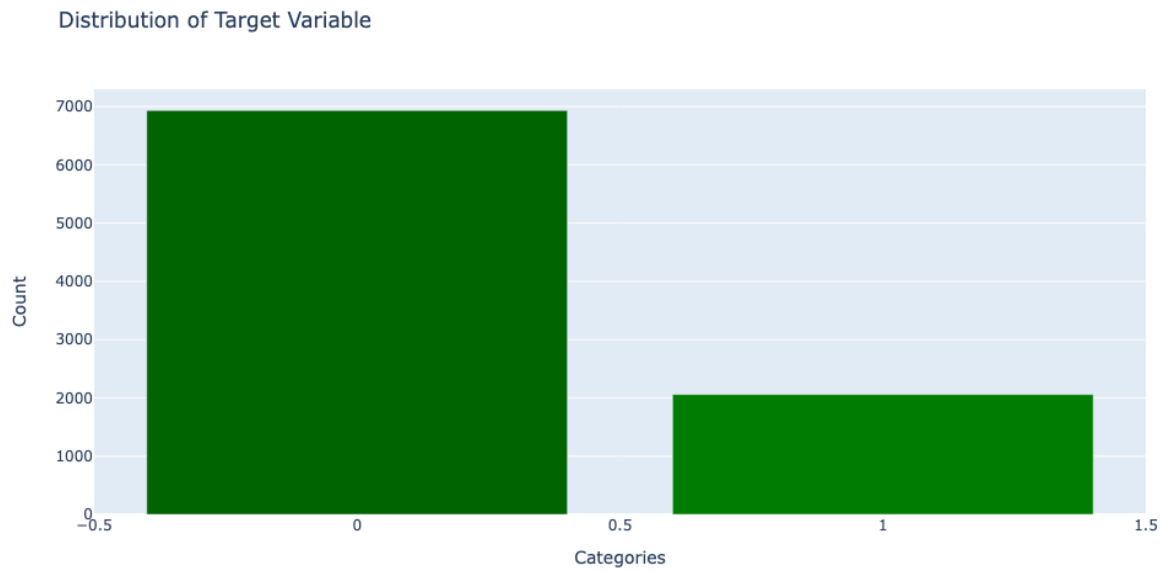
(b) Creating Y-Variable

In this portion of our data analysis, I encountered some challenges when creating the y variable. Our initial attempts at utilizing carbon monoxide and nitrogen dioxide in conditional formatting were unsuccessful due to the potential for an imbalanced target distribution. Instead, I turned to temperature as a viable alternative, given its significance in determining air quality, as previously established in our introduction. Upon examining the data's minimum (-1.9), maximum (44.6), average (18.3), and mode (20.8) values, I decided that a temperature of 25 degrees Celsius would be an appropriate threshold for high-temperature values. This decision was also informed by Italy's typical climate range, which usually spans from 0 degrees to 31 degrees Celsius. Although I considered using 30 degrees as the high-temperature threshold, I ultimately decided against it because the resulting target variable became too imbalanced. Specifically, the count of 'T' values above 30 degrees Celsius was only 938, which would have further exacerbated the existing imbalance.

By using the following code:

```
df_filtered['Air Quality'] = df_filtered['T'].apply(lambda x: 1 if x
> 25 else 0)
```

The lambda function systematically evaluates each component listed under the 'T' column. Whenever a value exceeds 25, it marks the corresponding row under the 'Air Quality' column with a 1, indicating an unhealthy reading. Conversely, if the value is below or equal to 25, a 0 indicates a healthy reading. This process is carried out for all 'T' variables, and the resulting values are then stored in the 'Air Quality column. After adding the new target variable to the dataframe, I created a bar graph to visualize its distribution. The healthy category (0) had 6932 entries, while the unhealthy category (1) had 2058 entries.



(c) Interaction Terms

I created interaction terms by selecting the variables 'CO(GT)' and 'NOx(GT)' based on their correlation in the matrix. Using PolynomialFeatures, I generated the interaction terms and converted them into a DataFrame. Then, I concatenated the interaction terms DataFrame with the original DataFrame. The reason for this correlation was that both NOx and CO are produced through the oxidation of nitrogen (N)

and carbon (C). However, when combined with oxygen (O), nitrogen oxides become the final product, while Carbon Monoxide is an intermediate product that can further oxidized into Carbon Dioxide (Cardu, 2005). Gaining a comprehensive understanding of the reasons behind certain gasses having a stronger correlation, as well as how they can potentially combine to pose a threat to the environment, can offer invaluable insights.

4. Feature Selection

(a) From Correlation Matrix

To initiate the process of selecting features, I started by determining the features that would be chosen if I utilized the correlation matrix as a selection method. I accomplished this by identifying the feature names with the highest correlation but not equal to one. Some of the features I found were 'CO(GT)', 'NOx(GT)', 'CO(GT)', 'NO2(GT)', 'PT08.S1(CO)', and 'C6H6(GT)'.

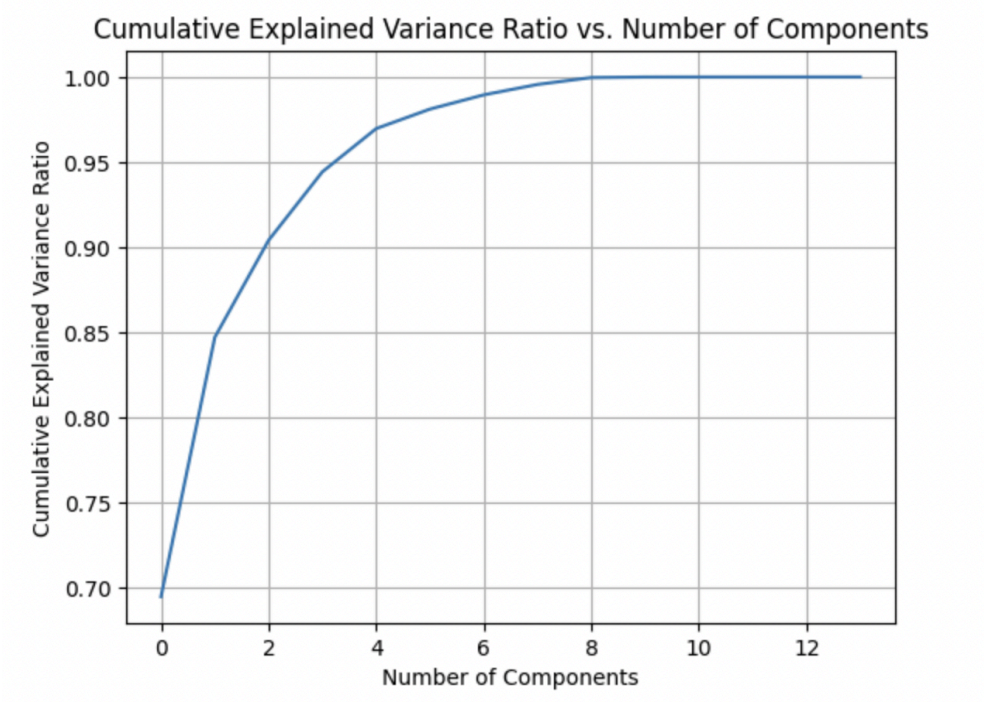
I conducted a correlation analysis between the target and x variables to determine their relationship. After sorting the correlation values in descending order, I found that the most correlated features were T(0.7573598117626836), RH(0.5499872228578574), AH(0.399839789093359), and PT08.S4(NO2)(0.32322475370639053). These results are logical because Temperature was previously used in conditional formatting to establish the y-variable, and humidity is known to trap contaminants near the ground, stopping them from spreading into the atmosphere. Additionally, NO2 is a gas that significantly impacts air quality, although I expected it to rank higher. If temperature hadn't been used to create the target variable, its ranking might have been different.

(b) K-Best

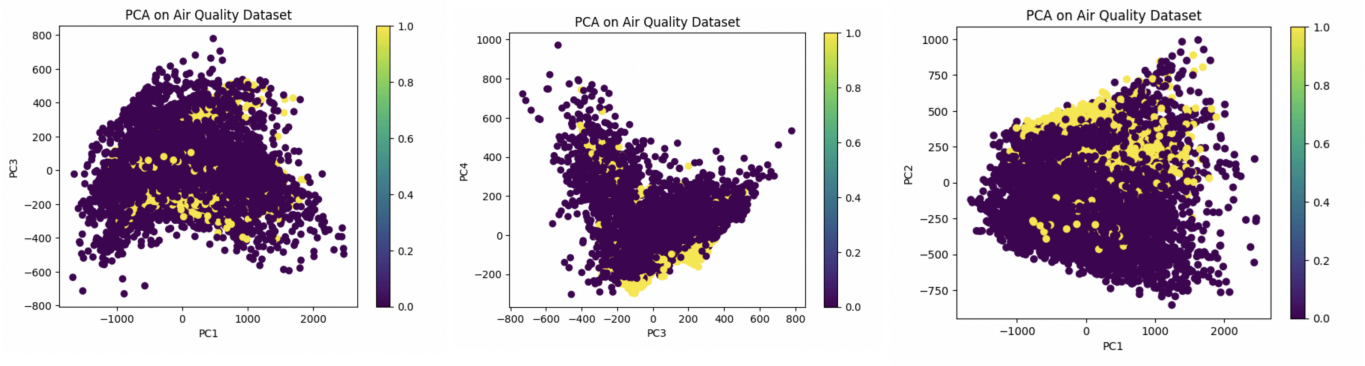
After conducting a K-Best analysis, the selected features were found to be similar to those in the correlation matrix. Here are their scores: NMHC(GT) (11.401146317413405), PT08.S4(NO2) (0.017035282354075816), T (42.42716427188705), RH (10.419628648055237), and AH (14.237013460524569). These will be used in the Logistic Regression model.

(c) PCA

I also attempted to employ Principle Component Analysis (PCA) as a feature selection model and subsequently generated a printout of the explained variance of the principal components. I then utilized matplot lib to create a cumulative explained variance ratio graph to understand this information better. This graph provides a visual representation of the total amount of variance in the data that is explained by a particular number of principal components. By analyzing the Elbow Point, which is the juncture at which the increase in explained variance ratio begins to level off, I determined that approximately four features account for the majority of the variance. This visualization helps in selecting the optimal number of components needed to capture a desired amount of variance within the data.



Explained Variance of Principal Components	PCA Weights (from first)
Principal Component 1: 395100.39246878 Principal Component 2: 86761.61766892544 Principal Component 3: 32447.588681702382 Principal Component 4: 22912.58698092342	1. CO(GT): 0.016917227111850345 2. PT08.S1(CO): 0.3249274553360494 3. NMHC(GT): 0.03251931544979937 4. C6H6(GT): 0.0111419077342393



If the scatter plot of the data transformed by PCA shows complete overlap, it could mean that the principal components are not effectively distinguishing or separating the different categories. This is illustrated in the PCA graphs above. In logistic regression, where the goal is to find linear decision limits, an overlapping PCA scatter plot reveals that the selected principal components may not accurately forecast the 'Air Quality' target variable.

(d) LASSO

Lastly, I used LASSO and the features chosen were, CO(GT), PT08.S1(CO), NMHC(GT), PT08.S2(NMHC), and NOx(GT)

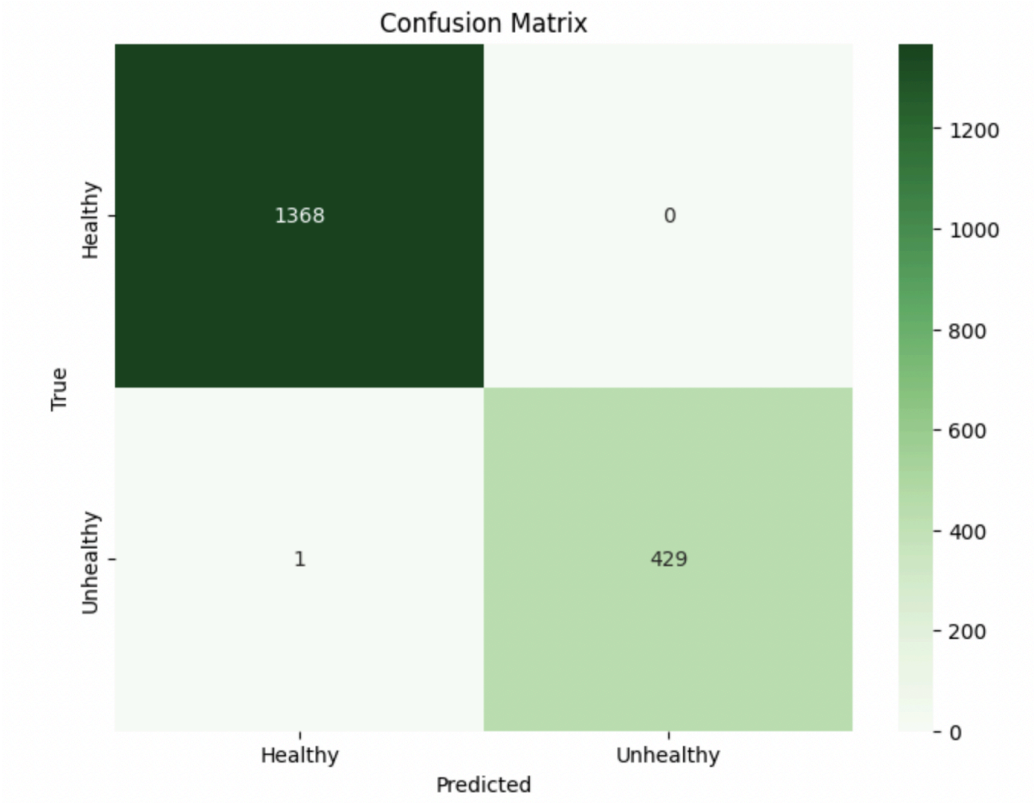
5. Building Machine Learning Models

(a) BASE Logistic Regression - using top 4 features

I followed a process to create a logistic regression model. First, I split the data into training and testing sets. Next, I removed the DateTime column from both X_train and X_test. Then, I created the logistic regression model and fitted it to the scaled training data. After making predictions, I evaluated the model's accuracy, mean squared error, R-squared, and out-of-sample mean squared error.

- Accuracy: 0.9961067853170189
- Mean Squared Error (MSE): 0.00389321468298109

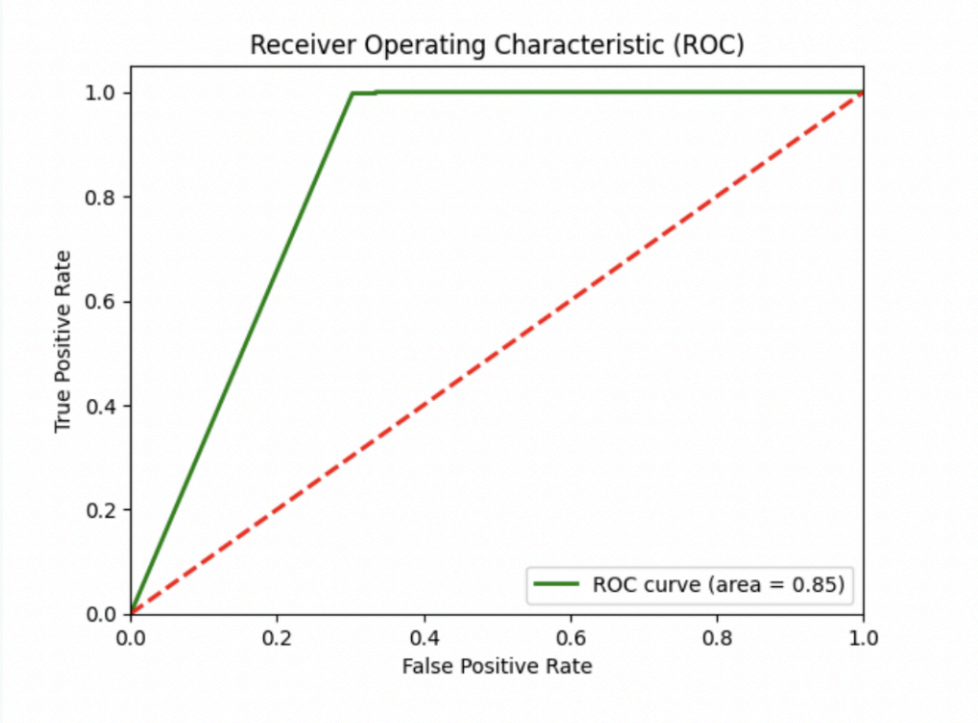
- R-squared (R2): 0.9786718453446126
- Root Mean Squared Error (RMSE): 0.06239563031960724
- Top 4 features: T, AH, PT08.S2(NMHC), & PT08.S4(NO2)



The accuracy of the base logistic regression model is surprisingly high. However, it is important to note that the dataset used for this model was obtained from UCI, which is known for providing reliable datasets with good results. The confusion matrix also had a very high true positive rate with only 1 misclassification. Let us take a look at some other scores:

- F1 Score:0.9879227053140097
- Precision: 0.9951338199513382
- Recall: 0.9951338199513382

ROC



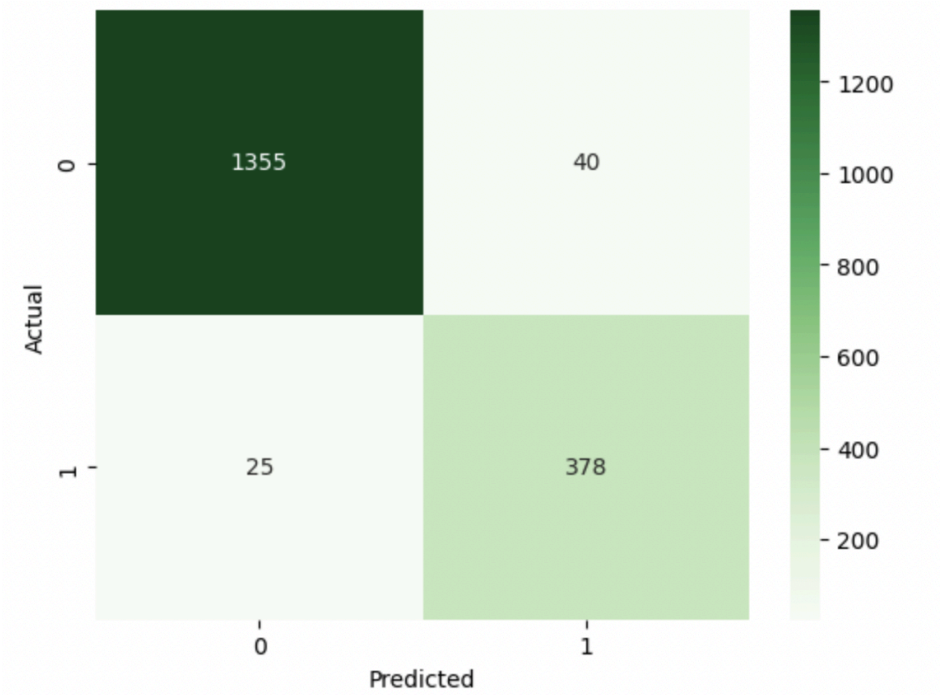
An AUC-ROC score exceeding 0.5 means the classifier performs better than random chance. As the score increases, the classifier's performance enhances. In this example, the score of 85 points indicates that the classifier is likely to rank a positive instance higher than a negative one if chosen randomly. Additionally, the curve closely approaches the top left corner, suggesting a high true positive rate.

(b) Using LASSO: Logistic Regression

Using the features selected from LASSO I created a new logistic regression model.

- Accuracy: 0.9638487208008899
- Mean Squared Error (MSE): 0.3681868743047831
- R-squared (R2): -1.1172318720705814
- Root Mean Squared Error (RMSE): 0.6067840425594456

Confusion Matrix



Accuracy: 0.9638487208008899

The Lasso Model for Logistic Regression performed worse than the base model. The confusion matrix evaluation showed that 65 values were misclassified, with a model accuracy of 96% compared to 99%. There were 40 instances where healthy air quality was interpreted as unhealthy and 25 instances where unhealthy air quality was predicted as healthy. This could be dangerous in terms of health and safety.

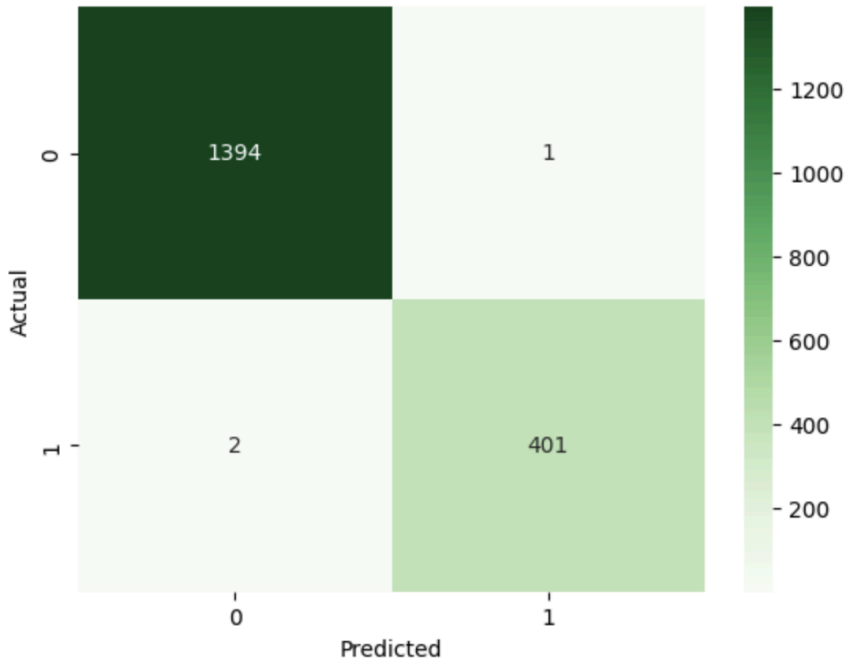
(c) Using KBEST - Logistic Regression

The results obtained from the K-Best method were similar to those of the base regression as both models used three identical features out of the four utilized in the base logistic regression. Additionally, the accuracy of the model was high, reaching 99%.

- Accuracy: 0.9983314794215795
- Mean Squared Error (MSE): 0.26974416017797553

- R-squared (R2): -0.5511441963054864
- Root Mean Squared Error (RMSE): 0.5193690019417558

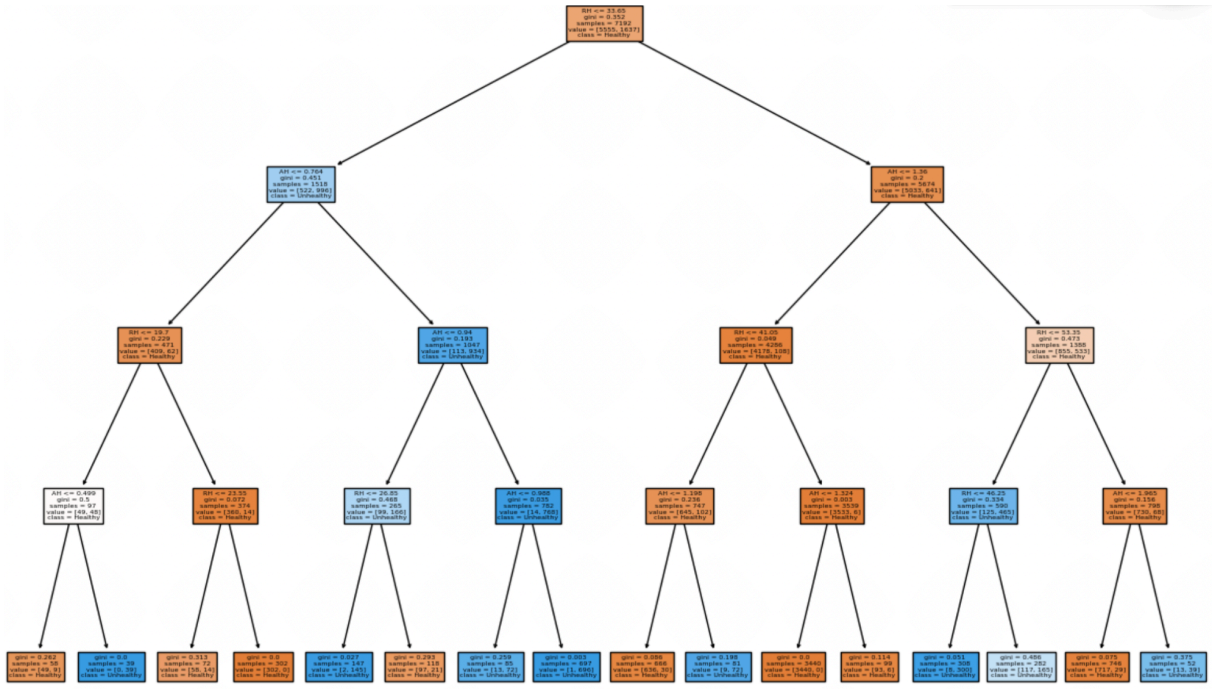
Confusion Matrix



The evaluation using the confusion matrix revealed that the model accuracy was 99%, with only 3 misclassified values. Specifically, there were 2 instances where unhealthy(1) air quality was wrongly predicted as healthy(0) and 1 instance where healthy air quality was wrongly identified as unhealthy. Overall, the model performed well, but when it comes to critical misclassifications, the Base logistic regression model still outperforms it with only 1 misclassification.

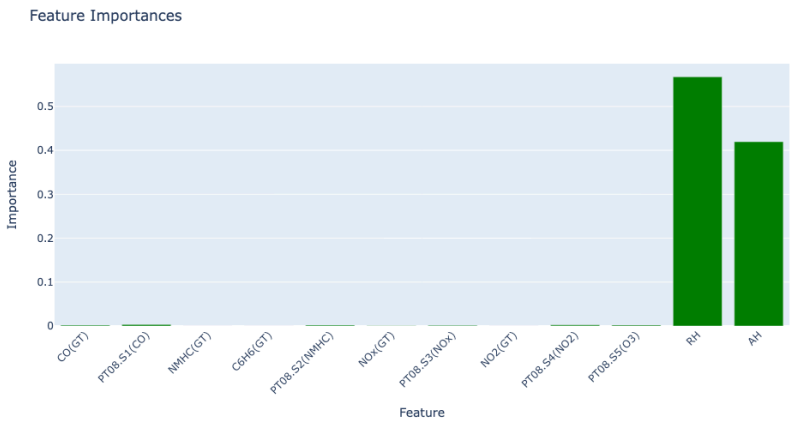
(d) Decision Tree

I utilized the features chosen from K-Best to construct the decision tree model since the regression model produced favorable results. The tree was graphed with the desired hyperparameters. The accuracy, again, was quite high at 96%.



*It is hard to read on the document but on the python notebook you can see the values

The decision tree algorithm effectively divides and predicts outcomes until it arrives at the leaf nodes, where the ultimate prediction is made. The expected result is denoted by class labels, with "0" representing healthy air quality and "1" indicating poor air quality. The Gini index evaluates the division's quality at each tree node, estimating the probability of mistakenly classifying a randomly selected element from the dataset. From the graph, it is noticeable that as the tree divides, the Gini index decreases.



CO(GT): 0.0017181728740302505

PT08.S1(CO): 0.003260532993979198

RH: 0.5674102665277662

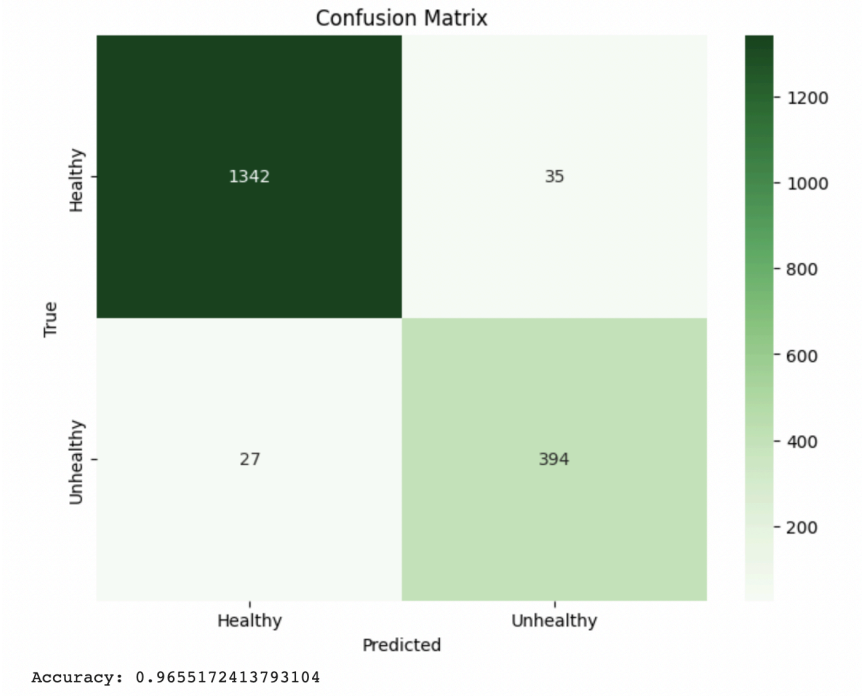
AH: 0.41961894608422873

PT08.S2(NMHC): 0.001964521115995076

NOx(GT): 0.0003468096218771968

PT08.S3(NOx): 0.001395295679504292

Confusion Matrix

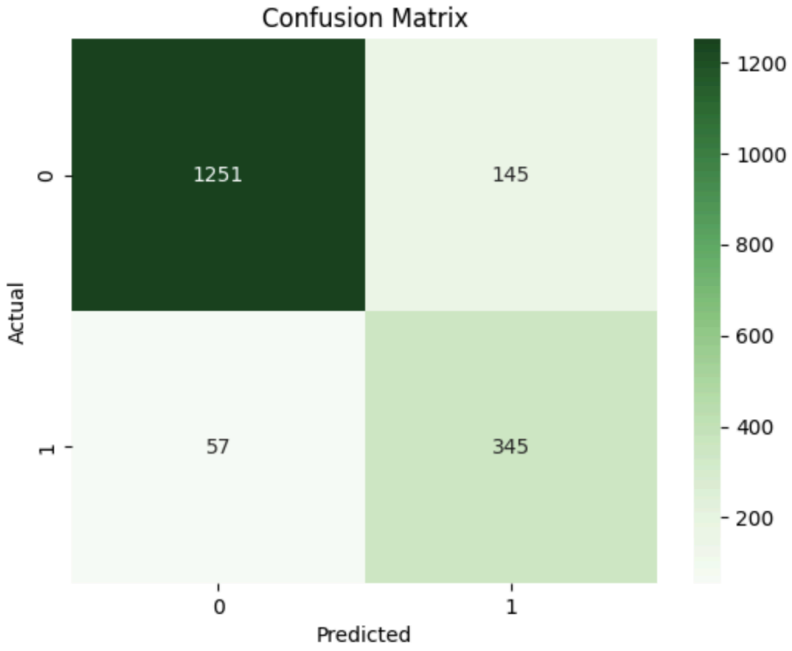


Based on the evaluation using the confusion matrix, it was found that the model accuracy was approximately 97%. However, there were 62 misclassifications, with 35 instances where healthy air quality was incorrectly predicted as unhealthy and 27 instances where unhealthy air quality was incorrectly identified as healthy. Although the model performed well overall, the Base logistic regression model performed better in critical misclassifications, with only one misclassification.

- F1 Score: 0.9270588235294117
- Precision: 0.9184149184149184
- Recall: 0.9358669833729216

(e) Naive Bayes

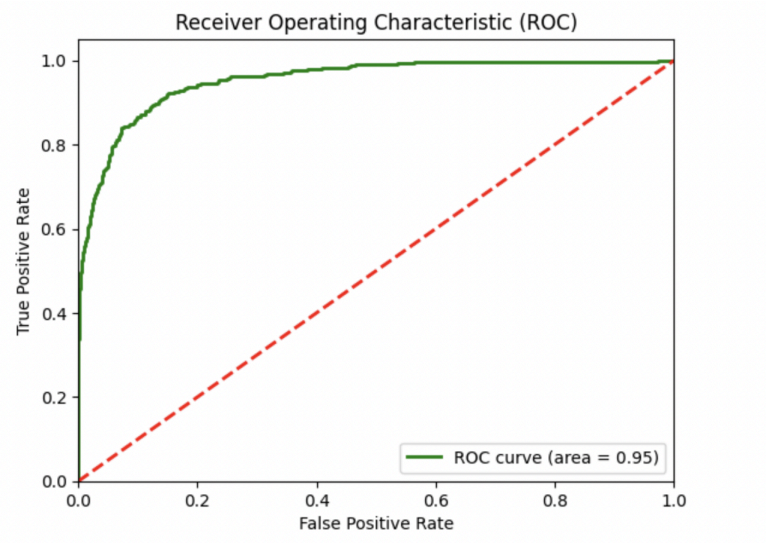
I developed a Gaussian Naive Bayes classifier to create a Naive Bayes model. I then trained the classifier using the training data and evaluated the model. Unfortunately, the accuracy of the model was only 88%, which is the lowest I have achieved so far.



In the Naive Bayes confusion matrix, there were 202 instances of misclassification, which was the highest so far. There were also 57 instances where unhealthy air quality was presented as healthy.

- F1 Score: 0.7735426008968611
- Precision: 0.7040816326530612
- Recall: 0.8582089552238806

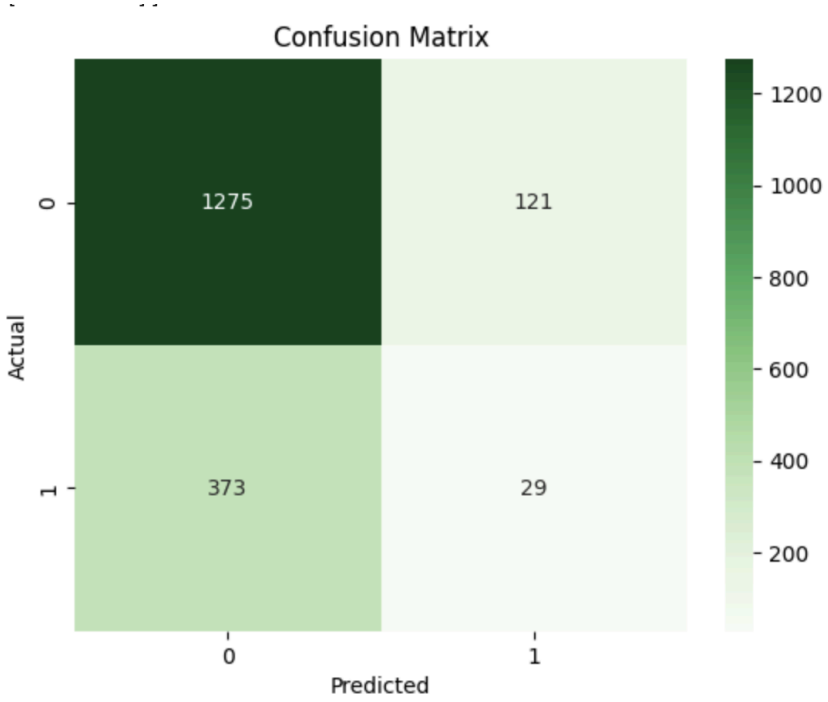
ROC



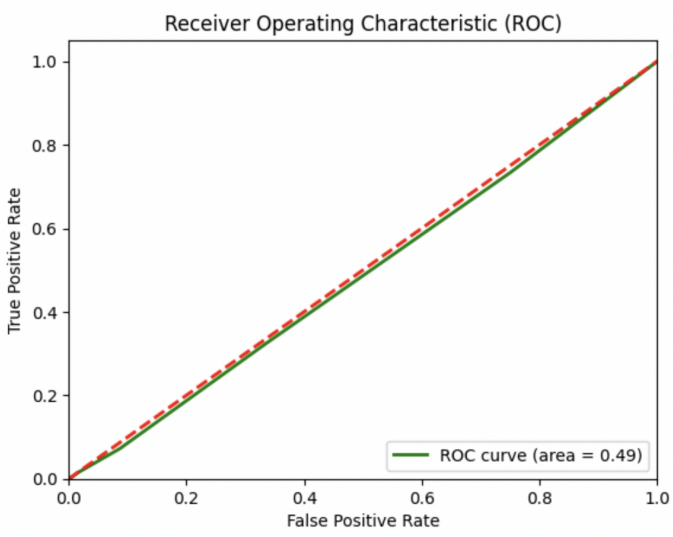
It was surprising that the ROC curve for Naive Bayes performed the best, even though its accuracy, F1 score, precision, and recall were lower than the other models. The curve approached the top left corner closely, indicating a high true positive rate.

(f) KNN

Out of all the models, the KNN model had the lowest accuracy rate, which was approximately 73%. The model's confusion matrix showed poor results, with a total of 494 misclassifications. Of particular concern are the 373 instances where unhealthy air quality was predicted as healthy, potentially impacting people's health.



- F1 Score: 0.10507246376811595
- Precision: 0.19333333333333333
- Recall: 0.07213930348258707



ROC

If an AUC-ROC score is below 0.5, it means the classifier performs worse than random chance. In this particular case, the score of 0.49 indicates that the classifier is likely to rank a negative instance higher than a positive

one if chosen randomly. Furthermore, the curve does not reach the top left corner, indicating a low true positive rate.

6. Conclusion

Through machine-learning models that use regression as an additional feature selection metric, this research aimed to predict if the air quality of an Italian city would be classified as healthy or unhealthy. The Air Quality dataset from the UCI Machine Learning repository was used to create various models, including Base Logistic Regression, Regression using K-Best, Regression using LASSO, Decision Tree, Naive Bayes, and KNN. To determine the best model, accuracy, confusion matrix, F1 score, precision, and recall were calculated for each model. Mean Squared Error (MSE), R-squared, and Root Mean Squared Error were also considered for logistic regression models. Based on the evaluation, the Base Logistic Regression model performed the best, with a lower MSE than K-Best, indicating a minimal difference between predicted and actual values. The R-squared value of 0.99 suggests a good fit, explaining approximately 99% of the variance in the target variable. Finally, the Root Mean Squared Error of 0.04 units indicates minimal deviation from the actual value.

Once again, companies can analyze air quality data using machine learning to identify pollutants and take measures to reduce harmful influences. This data can also help develop strategies for sustainability efforts and understand market demand in the environmental sector.

7. Bibliography

Cardu, M., & Baica, M. (2005). *Regarding the relation between the NOx content and CO content in thermo power plants flue gases. Energy Conversion and Management*, 46(1), 47–59. https://doi.org/10.1016/j.enconman.2004.02.009

Education, U. C. F. S. (n.d.). *How Weather Affects Air Quality | Center for Science Education*. UCAR. https://scied.ucar.edu/learning-zone/air-quality/how-weather-affects-air-quality

Saverio, Vitor. (2016). Air Quality. UCI Machine Learning Repository. https://doi.org/10.24432/C59K5F.