Prediction of Air Pollution

Inês Rocha

Introduction

This report details all the steps made to create the model.
First, we started with only one station but in the end, all stations were tested for the best model.

Problem Definition

Each day our planet is getting more polluted and it's getting more and more needed to know how will the pollution be in the next day.
Given a set of data collected from 12 stations in China, we will use their data to choose a model that will help us warn people to be careful when doing any activity outside.

Data Pre-Processing

It was created two functions to import, clean and process the data. The first function performs the following steps for all the 12 excel files from each station:

1. Import the data using the function read.csv and convert it to a dyplyr table.
2. Checks if there are NA values and if they exist the row will be removed.
3. The first column is the number of the row so it will be removed.
4. Import the AQI Breakpoint table provided in the Evaluation of the Chinese New Air Quality Index. TThis table has the breakpoints of each pollutant that will allow us to calculate the AQI value.

After examining the values from the AQI table and the data table, I realize that there were values from the pollutants that were bigger than the maximum possible for that pollutant in the AQI table. So those values were converted to the max present in the AQI table.

And also the CO pollutant needs to be divided by 100 to be at the same scale as the AQI table.

5. Calculate the AQI Value

The AQI value is calculated using the following formula:

$$\frac{maxAQI - minAQI}{maxCon - minCon} * (polluent - minCon) + minAQI$$

MaxCon and minCon are the maximum and the minimum concentration that our pollutant is in. MaxAqi and minAqi are the AQI values that belong to that concentration interval.

The PM10, PM2.5, SO2, NO2, CO pollutants are calculated by their 24H average rounded value, and the O3 polluent is calculated using the 8H and 1H average rounded value and is chosen the bigger one. To make this step easier was created a function that given a data table with the pollutant already calculated by their average, it performs a cut using the AQI breakpoints of that pollutant that gives us the classification of that value. Then it searches the AQI breakpoints Table for that classification and extracts the MaxAqi, MinAqi, MinCon and MaxCon that the is used in the formula above. After having all the AQI values, they will be put in a data table and the bigger value of each row will be chosen and that will be the AQI value

6. Build the data for the model, this is a simplified version of the data that means that we will not add the from the other stations. In this data we will have the following values calculated for each day:

- The minimum and maximum temperature
- The minimum and maximum pressure
- The minimum and maximum DEWP
- The maximum of rain
- The maximum occurrence of the direction of the wind
- The minimum and maximum WSPM
- The AQI value from the day before
- The AQI classification from the day before
- The AQI value for that day
- The AQI classification for that day
- The corresponding weekday
- The corresponding season
- The corresponding month

The second function

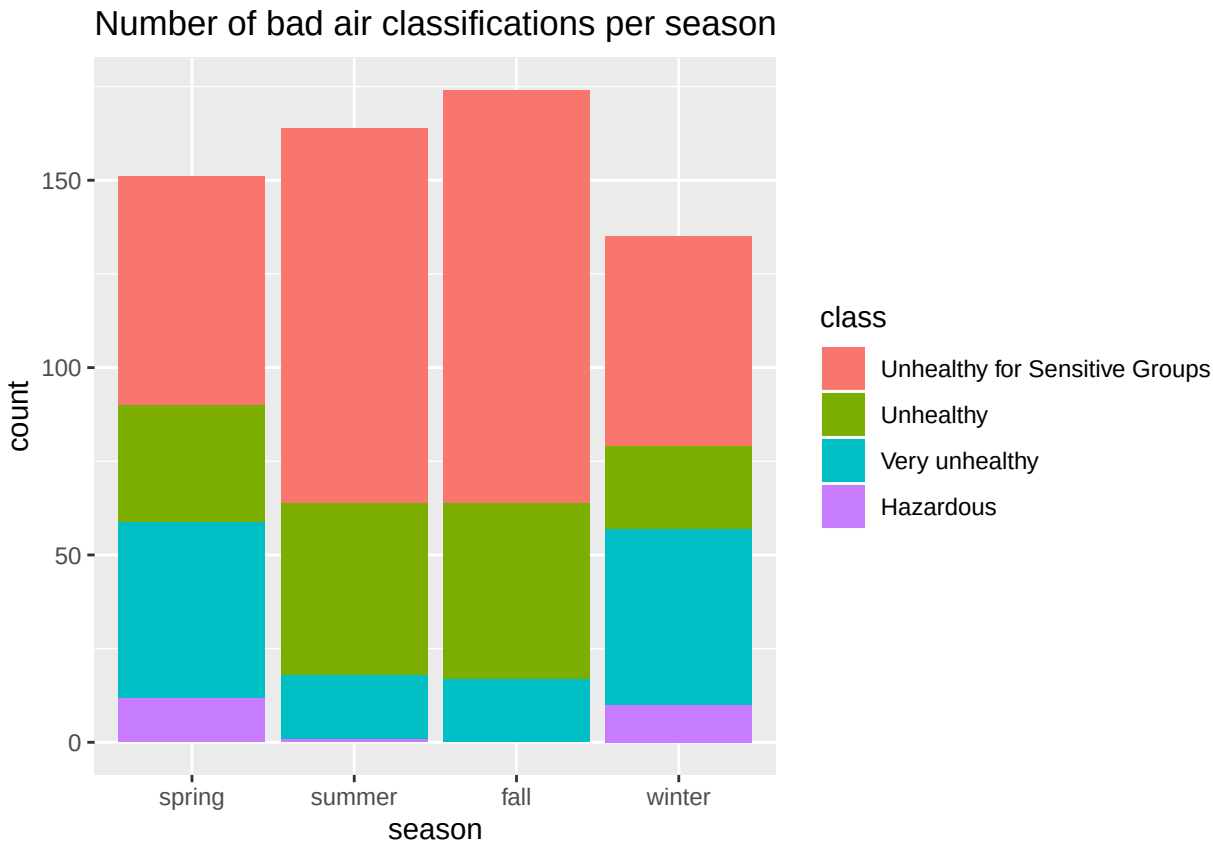
1. Receives a data table and adds the AQI classification values from the day before of all the other stations.
2. Cleans the data of all the columns that were created during the adding of the new columns and converts the char columns in factors.

Exploratory Data Analysis

Our data exploration consisted of seeing how air pollution changed in terms of certain parameters. By counting the number of AQI bigger then 100.

```
## # A tibble: 4 x 2
## # Groups:   season [4]
##   season     n
##   <fct>  <int>
## 1 spring   151
## 2 summer   164
## 3 fall     174
## 4 winter   135
```

We get that the fall is the season that has less polution and if we put this information into to a graph.



we see that besides the fall being better in terms of air quality it also is the season that doens´t reach the maximun of the AQI classification.

By doing the same as before but instead of the season we use the weekdays, we don´t see a significant different between them.

```
## # A tibble: 7 x 2
## # Groups:   weekdays [7]
##   weekdays          n
##   <chr>         <int>
## 1 domingo          94
## 2 quarta-feira     88
## 3 quinta-feira     96
## 4 sábado           97
## 5 segunda-feira    80
## 6 sexta-feira      85
## 7 terça-feira      84
```

Predictive Modelling: experimental setup and obtained results

Because the object of the objective was to predict the level of air pollution it was decided that we were going to predict the AQI classification that is a nominal variable. To do that we will make classification predicts using the following methods:

- Naive Bayes
- Decision Trees

- k-Nearest Neighbours
- Support Vector Machines And to compare them we used the library Performance Estimation.

To start with the predictions we first load the data and remove the column that as the AQI values because we only want the model to predict the classification of the AQI.
Next, we separate data into 70/30. 70% of the data will be used for training and the other 30% will be used to test our model.

To use these methods we need to pass them a few parameters:

- Using the decision tree we will add the parameter of max depth form 1 to 8 to see which value of depth is the best to predict the data.
- Using the k-Nearest Neighbours we need to give it a k. To tried to get the best one we run a function that iterates k from 2 to 200 and predicts the data. We then select the k that gives us better accuracy.
- Using the Support Vector Machines we will predict using different types of kernels (linear, polynomial, radial, sigmoid)

This is the result of this estimation for the data from the Huairou station:

```
## [1] "Workflow: svm.v1"
```

```
## [1] "Estimate: 0.5145833"
```

Even that the best accuracy was 0.5145833 it was still just a little above 50% of accuracy and that was not a satisfatory value for a model so we tried to repeat the same estimation, with the same parameters but including the AQI classification from the other stations.

```
## [1] "Workflow: svm.v3"
```

```
## [1] "Estimate: 0.4460870"
```

The accuracy went down. To see if this just happened is this data or if adding the AQI classification made our model worse was tried to run the same performance estimation with the data form Nongzhanguan.

```
## [1] "Without other stations"
```

```
## [1] "Workflow: svm.v1"
```

```
## [1] "Estimate: 0.5145833"
```

```
## [1] "With other stations"
```

```
## [1] "Workflow: svm.v3"
```

```
## [1] "Estimate: 0.4947826"
```

The results were the same as before. The model is better at predicting the AQI classification if the data has only data from those stations.

Not being happy with these predictions we decided to try a different method. Instead of having the model predict 6 classes, we would be putting the model predicting only two classes: Safe or Not Safe. This classification would let people know if the air quality of the air was safe for them to go outside.
A safe classification corresponds to the AQI value been between 0-100 (classification good and moderate). This new tactic was tested on the data from the Huairou station, and this was the result:

```
## [1] "Without other stations"

## [1] "Workflow: svm.v1"

## [1] "Estimate: 0.7591304"

## [1] "With other stations"

## [1] "Workflow: svm.v1"

## [1] "Estimate: 0.7704348"
```

Beside having a much better accuracy, the accuracy increases if we add the data from other stations. We made a table showing each station with their model values.

Table 1: Model for each Station

stations	Class	ClassAll	TwoClass	TwoClassAll
Data_Aotizhongxin	svm.v1 -> 0.5145833	svm.v1 -> 0.4591304	svm.v1 -> 0.7521739	svm.v1 -> 0.7486957
Data_Changping	svm.v3 -> 0.4729167	svm.v1 -> 0.4886957	svm.v3 -> 0.7721739	svm.v1 -> 0.7582609
Data_Dingling	rpart.v6 -> 0.4920863	svm.v3 -> 0.4678261	rpart.v6 -> 0.4920863	svm.v1 -> 0.7704348
Data_Dongsi	rpart.v6 -> 0.4920863	rpart.v5 -> 0.4878261	svm.v3 -> 0.7713043	svm.v1 -> 0.7704348
Data_Guanyuan	svm.v1 -> 0.4921986	svm.v1 -> 0.4626087	svm.v1 -> 0.7495652	svm.v1 -> 0.7330435
Data_Gucheng	svm.v1 -> 0.4619718	svm.v3 -> 0.4947826	svm.v3 -> 0.7443478	svm.v1 -> 0.7391304
Data_Huairou	svm.v1 -> 0.5145833	svm.v3 -> 0.4460870	svm.v1 -> 0.7591304	svm.v1 -> 0.7704348
Data_Nongzhanguan	svm.v1 -> 0.5145833	svm.v3 -> 0.4947826	svm.v1 -> 0.7713043	svm.v1 -> 0.7678261
Data_Shunyi	svm.v3 -> 0.4664234	svm.v1 -> 0.4260870	svm.v3 -> 0.7747826	svm.v3 -> 0.7286957
Data_Tiantan	svm.v1 -> 0.5104895	svm.v1 -> 0.4904348	svm.v1 -> 0.7660870	svm.v1 -> 0.7739130
Data_Wanliu	svm.v1 -> 0.4781022	svm.v1 -> 0.4756522	svm.v3 -> 0.7513043	svm.v3 -> 0.7373913
Data_Wanshouxigong	svm.v1 -> 0.4916667	svm.v1 -> 0.4800000	svm.v1 -> 0.7582609	svm.v1 -> 0.7547826

After examining this table we can see that the value between the models when using the data form the other stations sometimes gives us a better prediction and sometimes don't. The model that was chosen was the Support Vector Machines with a linear kernel (svm.v1). This model gives the best accuracy when predicting the data. But we think it would be worthed to also use the model Support Vector Machine with a radial kernel because these two appear a lot in the table.

Conclusions, Shortcomings and Future Work

The average final data table that was used had only 1400 rows and if we had more data the predictions could get more precise. Our future work will involve getting more data and getting a prediction of at least 90% so that people can feel secure when seeing our predictions.