Air quality predictor

Project in practical machine learning

Anniina Sallinen

March 16, 2017

Contents

1 Introduction

The subject of the project was predicting air quality based on weather. Many of the current applications are showing the air quality in real-time, but very few actually try predicting it. Those kind of applications do exists, though, and most of them are studies or governmental projects.

Finnish people are used to having very good air quality and it is usually good or excellent [5]. However, when cities grow, the air quality goes down. In Asia there are many cities where air quality is very poor and it affects everyday life of huge amount of people.
In fact, there are examples of this in Europe. For example in Paris they let public use public transportation for free if the air quality gets too low. Now imagine that we could predict air quality when we know the weather: actions could be performed beforehand and not just afterwards.

At first the intention was to use the API provided by Finnish Meteorological Institution, which only provided information for Finland. As said earlier, Finland is not very interesting country in a way that the air quality doesn't vary very much. After another student hinted about API which provides data as JSON and from all around the world, it was taken into use and Paris was selected as city to be studied. The API is called aqicn [1] and it was used to get air quality information. Another API Open Weather Map[2] provided the weather information.
There were two parts in this project: the one that retrieved the data, analysed it and made predictions based on the model formed from data. The other part was web application that retrieved weather predictions and gives prediction of air quality based on that information. It also shows how model is performing: it shows latest weather and air quality observation, and which air quality category the model assigned to it based on weather.

2 Data

As mentioned in introduction, two APIs were used as data sources: aqicn and Open Weather Map.

2.1 Aciqn API

Aciqn API provides information about air quality as JSON, including information about when the data was updated (about every two hours) and different measurements considering air quality. There is example JSON below.


```
{"status":"ok",
"data":
        {"aqi":63,
        "idx":5722,
        "attributions":
                [{"url":"http://www.airparif.asso.fr/",
                "name":"AirParif - Association de surveillance de la
                qualité de l'air en Île-de-France"}],
        "city":
                {"geo":[48.856614,2.3522219],
                "name":"Paris","url":"http://aqicn.org/city/paris/"},
        "dominentpol":"pm25",
        "iaqi":
                {"co":{"v":9.1},
                "h":{"v":93},
                "no2":{"v":26.6},
                "o3":{"v":4.5},
                "p":{"v":1022},
                "pm10":{"v":24},
                "pm25":{"v":63},
                "so2":{"v":1.1},
                "t":{"v":4.74}},
        "time":
                {"s":"2017-02-08 09:00:00",
                "tz":"+01:00","v":1486544400}
        }
}
```


The JSON provides air quality index in the first element of data, attribute called aqi. It reflects the amount of PM2.5 (pm25 in the JSON) which are particulates in the air and affect health [6] [4].

2.2 Open Weather Map API

Open Weather Map provides real-time information about weather in more detail. The information includes for example information about humidity, air pressure, temperature and wind speed. It was used for collecting weather forecasts for web application, too. JSON below is an example of real-time weather data returned from API. Forecast data contains list of similar JSON.


```
{
    "base": "stations",
    "clouds": {
        "all": 90
    },
    "cod": 200,
    "coord": {
        "lat": 48.86,
        "lon": 2.35
    },
    "dt": 1485952200,
    "id": 6455259,
    "main": {
        "humidity": 76,
        "pressure": 1011,
        "temp": 11.5,
        "temp_max": 12,
        "temp_min": 11
    },
    "name": "Paris",
    "sys": {
        "country": "FR",
        "id": 5610,
        "message": 0.7035,
        "sunrise": 1485933567,
        "sunset": 1485967742,
        "type": 1
    },
    "visibility": 10000,
    "weather": [
        {
            "description": "overcast clouds",
            "icon": "04d",
```

```
            "id": 804,
            "main": "Clouds"
        }
    ],
    "wind": {
        "deg": 190,
        "speed": 4.6
    }
}
```


2.3 Air quality data

As mentioned, the air quality is measured as particular matter which has diameter less than 2.5 micrometers (PM2.5). Outdoor they are mainly from traffic and burning of fuels such as wood or oil. The particulates are so small that they can get into lungs and cause shortness of breath and irritation of eyes [4]. In the table below there are the levels of air quality index, which is originally from aciqn page[3].

AQI	**Level**
0-50	Good
51 -100	Moderate
101-150	Unhealthy for Sensitive Groups
151-200	Unhealthy
201-300	Very Unhealthy
300+	Hazardous

These are the same categories that are in this project's predictions. For the first level there are no health effects or cautionary statements. Moderate level of air quality has no effect on most people either, but it might affect some people that are unusually sensitive.
Even in the next level most of the population is not affected, but risk groups such as children and elderlies are advised to limit their stay outdoors. Also people with for example asthma may experience some impact. From the level on other people start to feel some effects, too, and sensitive groups should avoid long stay outdoors.

3 Technology

Two programming languages were used in the project: Python and Javascript. Python code made API calls for retrieving weather and air quality data, inserted those into PostgreSQL database and analysed the data. Javascript was used in backend and frontend of web application: it retrieved weather prediction data and sent it to frontend so that it could be displayed. Expressjs was used for creating web server and serving requests from frontend.
Python is a good choice for data processing because it has good C-based libraries for processing data (such as NumPy and scikit-learn). It also provides simple ways to make API calls and insert data to database. Javascript is good for manipulating DOM tree and is therefore good choice for frontend. It also provides easy way to create web server and make API calls both to external and local resources.
The application is hosted on virtual Ubuntu server hosted by UpCloud. Collecting weather and air quality data is done periodically every hour by using crontab on the server. The web application part, which shows latest entry and our prediction for that and weather and air quality forecasts for the future, is available at `http://83.136.250.43:3000/`.

4 Methodology

In the section "Data sources" it was described which data sources are used and what information they provide. From all available information following is used: air quality index, when the information was updated, humidity, pressure, temperature and wind speed. The air quality data is not updated real-time, new row is added only when new information is updated and retrieved, and then matched to the weather data which were measured at the same time. The hypothesis is that in the days that wind speed and humidity is high, the air quality is good. High humidity means in this case that humidity is 100%, which occurs when it is raining or there is foam.

4.1 Processing data

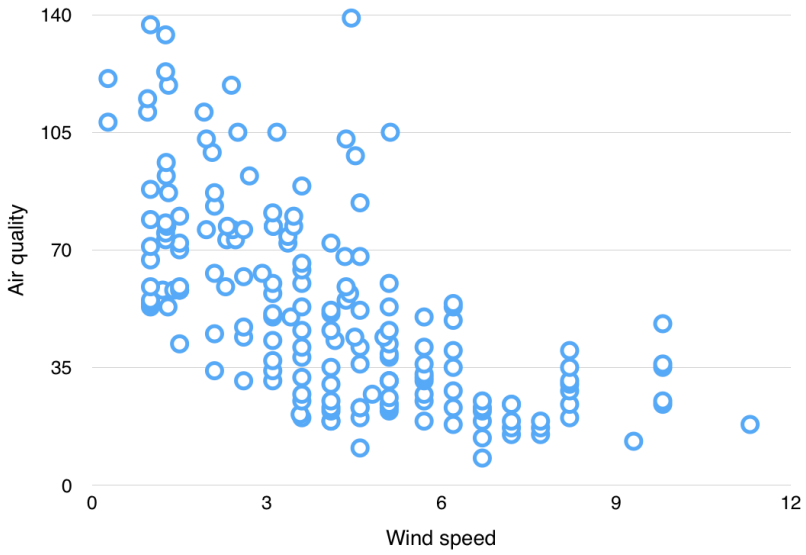
As there were very clear categories for air quality, the very obvious method was classification. For each level of air quality "typical weather conditions" were formed by taking the mean of each weather feature: humidity, pressure, temperature and wind speed. Predicting air quality was then based on those categories: the difference between weather data row values and model values was calculated for each attribute and the category with most similar values was

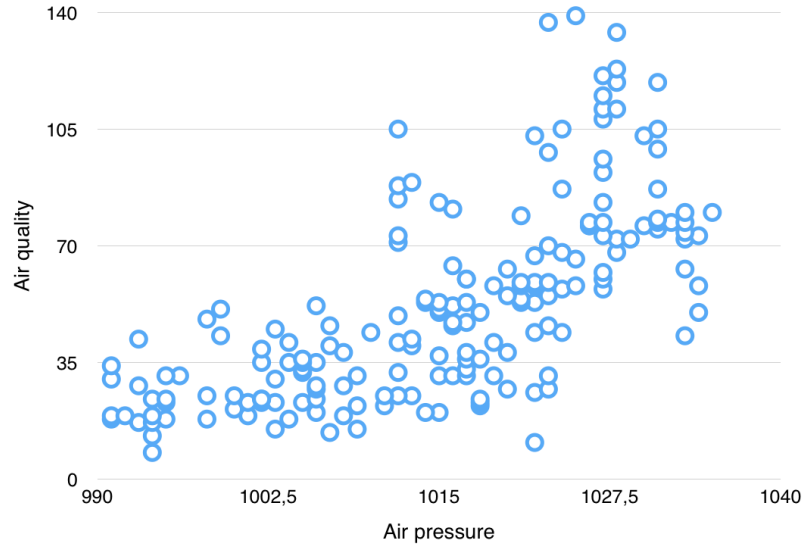
selected. If there was a tie, the better quality was selected.

5 Results

Measuring success rate was done continuously by using 70% of data for training and remaining 30% for testing. After creating model it was tested with test data and the results varied as amount of data increased. Success rates and models were saved in database and for different models success rate varied from 40% to 70%.

The initial hypotheses are not properly tested, but predicting with different combinations of attributes, always leaving something out, was performed and the success rate seemed to get worse every time. Especially wind speed and air pressure seemed to correlate pretty well with air quality. There are example figures below which show the correlation between the attributes and air quality: based on the figures we can say that when the air speed goes higher, the air quality gets better and when the air pressure increases, so does the air quality index.





6 Problems encountered and lessons learned

The APIs used provided very good attributes to predict the air quality, but the problem was that the air quality information was not updated frequently enough and amount of data is low. The air quality information was updated every hour or two so it resulted to only 313 rows between 12.02.-14.03. There were also problems with availability of the other data source, Open Weather Map API, because it didn't guarantee 100% availability and it was indeed quite often unavailable.

As weather and air quality data were merged based on weather data retrieval time and air quality update time, there were some points of time that we had air quality data but not weather data. Due to this, and some other problems encountered during project, the amount of usable data was reduced to 190 rows. Neither of those would have been a problem if there was history data available. Unfortunately, free sources of weather history data in Paris were not available, so only choice was to gather data as much as possible.

Open Weather Map also limited the amount of requests made, but that did not turn out to be a problem, because weather condition do not usually change very often and since in this case the air quality was not real-time, we did not have to update the weather data that often, either.

Although there are six levels of air quality, only three of them were present

in the data: good, moderate and unhealthy for sensitive groups. This means that typical weather for rest of the levels could not be formed and if in the future the air quality gets worse, the application would not be able to predict that air quality based on weather conditions. The amount of data rows where air quality is unhealthy for sensitive groups is also very low. Out of 190 rows only 16 are in that category. This made forming stereotypical weather for that category very hard and should have been taken into account better.

Another way to predict the air quality could have been a good idea. Linear regression would be a pretty obvious choice and then we could predict exact values instead of categories. This would have had some effect on how we measure success rate. Neural network could also be used, and they were used in the study of Ordieres et al. [7]. The study was conducted on Mexican-USA border and it was shown that neural networks performed better than linear models. It used some same attributes, such as wind speed and temperature for modeling and predicting.

Measuring success could also be improved. Now if we have for example weather row which real air quality is 51 and thus in moderate category and based on weather we predict it is in good category, we punish too much for it. We punish the application as much when the case is that the real air quality is 99 and we thought it would be good. However, if we used for example linear regression and predicted the exact number in the first case we would punish less than in the second case, because the numerical values are compared, not just the categories. Measuring success rate could be improved also by performing cross validation and ensuring that different amount of data in categories is mirrored in the training data set.

References

[1] Api for air quality. http://aqicn.org/api/. Accessed: 2017-02-08.

[2] Api for weather. http://openweathermap.org/api. Accessed: 2017-02-08.

[3] Aqicn page listing air quality levels. http://aqicn.org/scale/. Accessed: 2017-02-08.

[4] Department of health, q&a about pm2.5. https://www.health.ny.gov/environmental/indoors/air/pmq_a.htm. Accessed: 2017-03-11.

[5] Finnish meteorological institution, information about affects of weather in air quality. http://en.ilmatieteenlaitos.fi/weather-and-air-quality. Accessed: 2017-03-11.

[6] Wikipedia article about particulates. https://en.wikipedia.org/wiki/Particulates. Accessed: 2017-02-08.

[7] Ordieres, J., Vergara, E., Capuz, R., and Salazar, R. Neural network prediction model for fine particulate matter (pm 2.5) on the us–mexico border in el paso (texas) and ciudad juárez (chihuahua). *Environmental Modelling & Software 20*, 5 (2005), 547–559.