KDD CUP of Fresh Air

Jie Zhou, Hengxing Cai, and Xiaozhou Liu

East China Normal University
Sun Yat-Sen University, Cortex Labs
Sun Yat-Sen University
{jzhou}@ica.stc.sh.cn
{caihx1288}@qq.com

Abstract. This paper is about building an air quality prediction system for KDD CUP 2018: Fresh Air Task, worked by the team Ready Player One@ICA@CortexLabs. The air quality system is to predict concentration levels of several pollutants over the upcoming 48 hours for two cities: Beijing, China, and London, UK. The prediction framework is mainly composed by five parts: data crawling, data preprocessing, feature engineering, modeling and ensemble. Finally, the results show a good performance of the prediction method.

Keywords: KDD CUP 2018, fresh air

1 Introduction

Over the past years, air pollution has become progressively more severe in many large cities, such as Beijing. In 2011, an article [1] in the Los Angeles Times cited Dane Westerdahl, an air quality expert from Cornell University, describing the air quality of Beijing as 'downwind from a forest fire'. Among different air pollutants, air particles, or Particulate Matters (PM), are one of the deadliest forms. Particles with a diameter of 2.5 μm or less (called PM2.5) can penetrate deeply into human lungs and enter blood vessels, causing DNA mutations, cancer, central neural system damage, and premature death.

Existing biomedical [1] research demonstrates that, once inhaled, PM2.5 can hardly be self-cleaned by the human immune system. Therefore, accurately monitoring and predicting the concentration of PM2.5 and other air particles have become increasingly crucial. With precise predictions of air pollution levels, the public and governments can respond with appropriate decisions, such as closing schools and discouraging outdoor activities, to greatly mitigate the harmful consequences of air pollution.

In this task, we are requested to predict concentration levels of several pollutants, include PM2.5, over the coming 48 hours for two cities: Beijing, China, and London, UK. On each day throughout the competition, air quality data and meteorological data for both cities will be provided on the hourly basis. For

[1] http://articles.latimes.com/2011/oct/29/world/la-fg-china-air-quality-20111030

example, on May 14, the participants will be able to access historical data up to May 14 (including), and will have to predict the pollution level for May 15 and 16. Over a period of 24 hours (by 23:59 UTC), each team will be allowed to make no more than 3 submissions to predict 48 hours of air quality results, starting from 0:00 UTC of the next day. You can see more details on the submission API and the submission file format on the 'data' [2] page or in this tutorial [3].

The reminder of the paper is organized as follows. Section 2 describes our approach. In Section 3, experimental results are presented. Finally, the paper is concluded in Section 4.

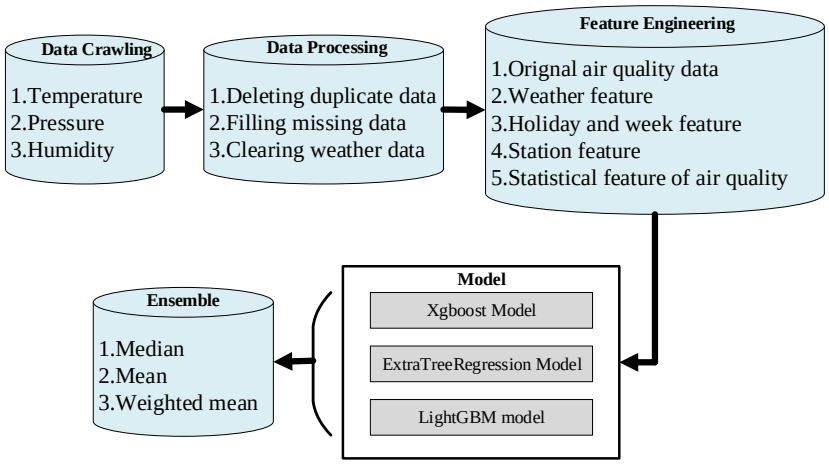

Fig. 1. The architecture of our system.

2 Our Approach

In this section, we demonstrate the architecture of our system, which is shown in Figure 1. It shows that our system mainly consists of five parts, namely data crawling, data preprocessing, feature engineer, model and ensemble. The details of each part are demonstrated in the following sections.

Data Crawling We think the weather forecast data is important for the air quality prediction, so we crawl external data from website underground [4]. We get weather data (e.g. temperature, pressure, humidity, wind_speed and

[2] `https://biendata.com/competition/kdd_2018/data/`
[3] `https://biendata.com/forum/view_post/9`
[4] `https://www.wunderground.com/`

wind_direction) from the API 'https://api.weather.com/v1/geocode/' + str(lat) + '/' + str(lng) + '/forecast/hourly/240hour.json?apiKey= 6532d6454b8aa370768e63d6ba5a832e&units=e'. You can see the detail on the file **crawl_data.py** and **weather_data_processing.py** under the baseline (folder).

Data Preprocessing Before we start to run the system, we preprocess the dataset. We first solve the tweets in the following steps:

- Deleting the duplicate data.
- Filling missing data. First, if the missing data is less than three, we will fill them with a linear function. Second, we use the continuous data to train a model, and predict the missing data by this model. You can see the details on the file **data_processing.py** and **pre_train.py** under the baseline.
- Clearing the weather data and we only keep the temperature, pressure, humidity in our model, see file **weather_data_processing.py**.

In particular, we use the first three days to train a Xgboost model to predict the value of next hour. Considering the distribution between different air pollutants and different cities, we train five models to predict the missing data (bj_PM2.5.model, bj_PM10, bj_O3.model, ld_PM2.5, ld_PM10.model). In file **data_processing.py**, the function **pre_precessing()** deletes the duplicate data, the function **loss_data _process_main()** prepares the training data for xgboost model and fills the missing data, and the function **pre_main()** trains the xgboost model.

Feature Engineering We extract the features in this step, see the function **get_all_ statistic_feature()** in file **lightgbm_with_weather.py**. Before extracting the features, we first split the data into samples in lines through sliding-window, see function **get_train_test_data()** in file **extraTree_with_weather.py**. We list the features as follows:

- Original air quality features. We use the data of the first twenty-one day to predict the next two days.
- Statistical feature of air quality. We statistic the max, min, mean, median, standard deviation, variance on different units, including day, week, all day and so on.
- Weather features. We add the temperature, pressure, humidity of history weather data and next two days' forecast weather data.
- Station features. The id and the type of stations will be used in our model.
- Holiday and week features. We arrange all the holidays, the first day of work, the first day of holiday, weekdays and so on as the features.

Model We select xgboost, extra-tree regression and lightgbm as our baselines.

- Xgboost model [2]. In this model, we add the number from 1 to 48 as features to predict.

- ExtraTreeRegression model [4]. We predict 48 values in a time with this model.
- LightGBM model [3]. This model is really fast, and this model get the best results in all the models.

Ensemble Finally, we ensemble all the model results(see file **ensemble.py**) according to the following methods:

- Mean. We select the mean of the results as the final result.
- Median. We select the median of the results as the final result.
- Weighted mean. We search the best weight through the valid data.

For each model, we train with different parameters and different data data resolution and the missing rate of data to improve the stability of the model.

3 Experiments

3.1 Data

We use historical air quality and meteorological (weather) data to forecast air quality in the future. for two cities, Beijing, China and London, Britain. Between April 1 to April 30, 2018, we can use live API to acquire data and submit our result for practice.

From May 1st to May 31st, we are required to submit on daily basis. The results of this phase will decide the final leaderboard of this year's KDD Cup.

Noting that we need to submit our result before 23:59 (UTC time) every day, to predict the air pollution from 0:00 (UTC time) of next day to 23:00 (UTC time) of the day after next. Therefore, we need predict 48 hours' values in each submission.

3.2 Evaluation

On each day, the submitted values will be compared to the ground truth (i.e. pollution concentration in the real world) via symmetric mean absolute percentage error [5]:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{(A_t + F_t)/2}, \tag{1}$$

If the actual value and the forecast value are both 0, we will set SMAPE score as 0, too.

Participants are required to predict the PM2.5, PM10, and O3 concentration levels over the coming 48 hours for every measurement station in Beijing (another city will be disclosed on 3/31). Predictions can be made on the daily basis, over the course of one month. To evaluate the predictions, on each day, SMAPE scores will be calculated for each station, each hour of the day (48 hours overall), and each pollutant (PM2.5, PM10, and O3). The daily SMAPE score will then be the average of all the individual SMAPE scores.

[5] `https://en.wikipedia.org/wiki/Symmetric_mean_absolute_percentage_error`

Team	SMAPE
First floor to eat Latiao	0.3681
getmax	0.3696
Ready Player One@ICA@CortexLabs	0.3767
deepx	0.3847
Late team	0.3854
oneday	0.3863
ToBeDone	0.3907
613papa team	0.3910
Zugzug	0.3926
Tony2018	0.3927

Table 1. Performance of our submitted runs.

3.3 Experiment Results and Analysis

The experiment results are shown in Table 1, which proves the good performance of the proposed model, And our model gets the second prize on the second-day prediction.

4 Conclusions

In this paper, we present our work in KDD CUP 2018 of fresh air. We build a air quality prediction system. It mainly performs five steps to predict. Noting that the our strategy works very well, we will extract more useful features and focus on the learning to regression approaches in the future.

References

1. Becker, S., Fenton, M.J., Soukup, J.M.: Involvement of microbial components and toll-like receptors 2 and 4 in cytokine responses to air pollution particles. American journal of respiratory cell and molecular biology 27(5), 611–618 (2002)
2. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794. ACM (2016)
3. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems. pp. 3149–3157 (2017)
4. Liaw, A., Wiener, M., et al.: Classification and regression by randomforest. R news 2(3), 18–22 (2002)