CMPT 318 | Project | Weather Condition Classifier

Nan Liu

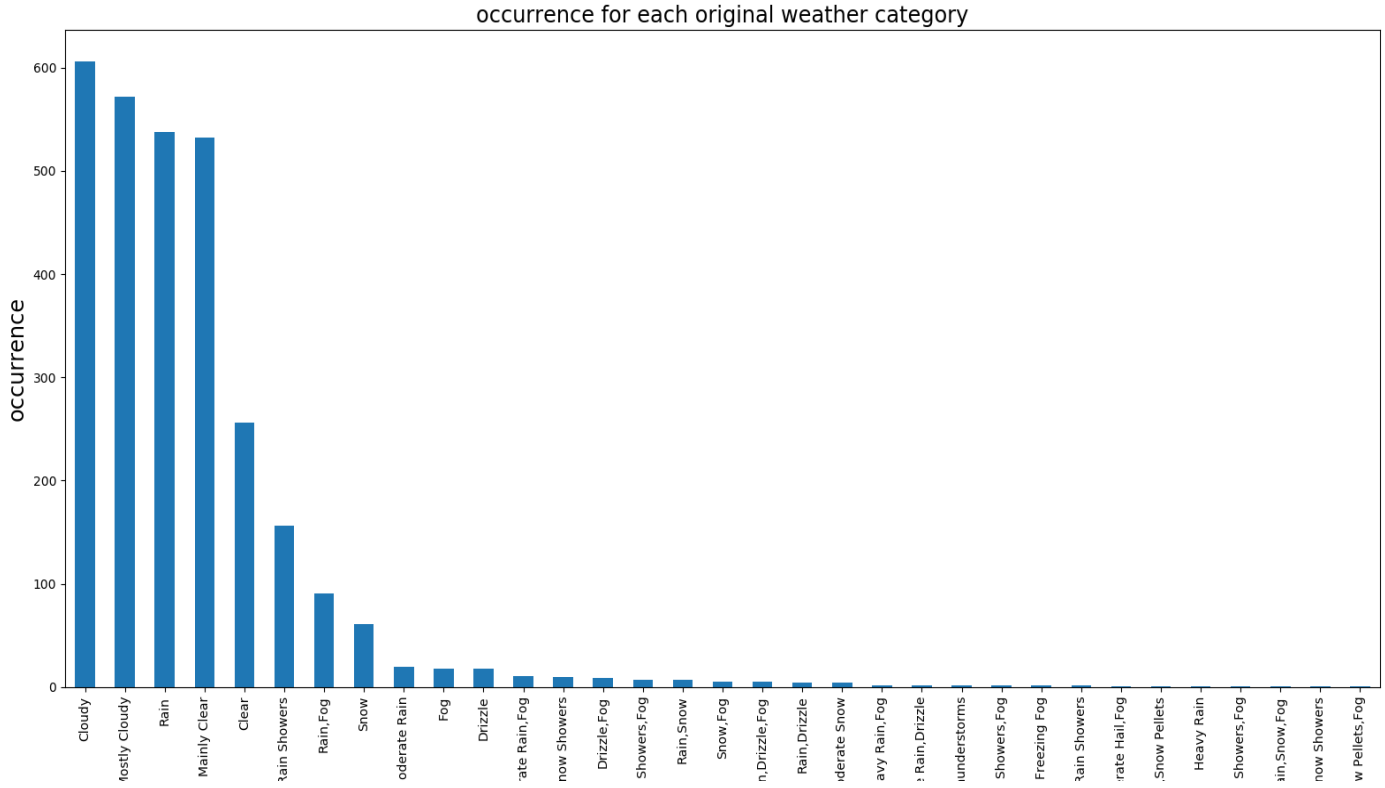
Gathering Data

For this project, I split the images and csv files provided as training and testing data. To quickly access the data, I used the python Glob and OS modules to efficiently read all the csv files in the folder into one dataframe and all the images in another folder into one numpy array.

Data processing and cleaning

1. Some images do not have corresponding weather records, and some weather records do not have corresponding images. To link images to corresponding weather records, I used their shared information – time. I extracted and formatted the time information from image names and join them on the dataframe containing the weather records. And then only rows having both weather descriptions and images are kept. I stored the index of the image in the numpy array into the dataframe to associate the record with the image.

2. After extracting all unique weather descriptions, I found that: **A**. there are some descriptions with preceding adjectives to describe the intensity of weather patterns, but the occurrences of this kind of descriptions are very low. **B**. The occurrences of all descriptions with multilabel are very low. **C**. Some weather categories have less than 10 images associated with them. The bar graph below shows these features.



Considering the low resolution of the images, the subjectivity of the descriptions and to simplify the classification, I did the following cleaning on the dataset:

- Exclude records with weather descriptions whose occurrences are less than 10.
- Discard all adjectives in the description, like “Moderate”, “Mainly”, “Mostly”, etc. Especially, categorize “Drizzle” as “Rain”.
- Since images can only reflect the weather conditions at the time they are taken, “Rain Showers”, “Snow Showers” are equivalent to “Rain” and “Snow” respectively.

After these cleaning on the weather descriptions, the final weather categories and their occurrences are as follow:

Weather	Occurrence
Clear	640
Cloudy	903
Rain	591
Snow	57

3. Some images taken in the early morning and at dusk are very dark, which do not provide useful information for the weather. So I filter out all images taken at 6:00 AM and 7:00 AM, as well as images taken at 19:00 in October, 17:00, 18:00, 19:00 in November, December and January.

4. I tried to only use the 2/3 part of the image containing only the sky by getting rid of the 1/3 bottom part which contains surface, but there was no improvement in accuracy. So I discarded this step.

Analysis

I used three classification algorithms, namely Bayesian Classifier, KNN and SVM.

Before data cleaning and processing, the score for SVC model is only 0.48. After taking the steps above, the score is more than 0.70. And I have individually tested that step 1 and 4 listed above improved the accuracy score independently.

After multiple simulations, I got the following results and findings:

Classifier and Best Parameter	Score
Bayesian	0.29
KNN (neighbor=8)	0.72
SVM (C=0.1)	0.75

- Both MinMaxScaler and StandardScaler do not work, they would significantly lower the score to around 0.3.
- PCA is the transformation method to be used for speed, and the best parameter is 1000.
- The Bayesian Classifier failed to predict weather condition from the image.
- Both KNN and SVM did a satisfactory job on prediction, with SVM (kernel=linear, C=0.1) being the best classifier (the accuracy score ranges roughly between 0.74 and 0.78).
- Printing out the report for SVM, the model did very well on snowy weather (0.92), good on clear weather (0.79), worst on cloudy (0.72) or rainy weather (0.75). The assumption is that to improve the accuracy of the predictions, special measures need to be taken on distinguishing cloudy and rainy weather.

Limitations

The images and records for some weather descriptions are insufficient to train the model, eg. Snow and Drizzle.

If there is more time, I would try to use the image to predict the season and the time of the day, come up with ideas to help the model to better distinguish between rainy and cloudy weather, and try other machine learning models.

Project Experience Summary

Weather Condition Classifier — Nov - Dec 2017

- developed a python program using Pandas, Numpy and Scikit-learn libraries to efficiently predict the weather in an image with an accuracy score of 0.75.
- applied extract-transform-load workflow to gather, clean and process the data, which improved efficiency and increased the accuracy score from 0.48 to 0.75.
- compared and analyzed several machine learning models and processing methods in terms of weather prediction performance.