Weather Forecasting with PySpark

Big Data Computing project A.Y. 2020-2021
Prof. Gabriele Tolomei

MSc in Computer Science
La Sapienza, University of Rome

Andrea Gasparini

Addressed task

- It is possible to make valuable predictions of meteorological conditions only based on previously seen meteorological data?
- The goal of the task is to create a Machine Learning model that, given a set of meteorological measurements, predicts which meteorological condition should occur

Dataset

- https://www.kaggle.com/selfishgene/historical-hourly-weather-data
- Hourly weather measurements data of 36 cities, collected from 2012 to 2017
- Approximately **45.000** weather measurement (e.g. temperature, humidity, air pressure, ...) for each city
- Composed by 7 different *csv* files:
    - one *csv* file containing geographical information about the different cities
    - one *csv file* containing the textual description of the weather conditions, where each column refers to a different city and each row refers to a specific datetime in which the weather condition occurred
    - one *csv* file for each weather measurement type, where each column refers to a different city and each row refers to the specific datetime of the measurement

Outline

Dataset preprocessing

- Data analysis and exploration
- Dataset shape and schema refactoring
- Target classes unbalancing

Machine Learning pipeline

- Data encoding
- Training Machine Learning models
- Evaluation and comparisons
- Best model selection

OpenWeather comparison

- Retrieval of actual weather forecasts
- Comparison with the best model results

Creation of a *DataFrame* that includes all the data

- Working with multiple *csv* files with this format is not the best option for Machine Learning purposes
- The best solution would be to have a single *DataFrame* instance which includes all the information about the hourly measurements

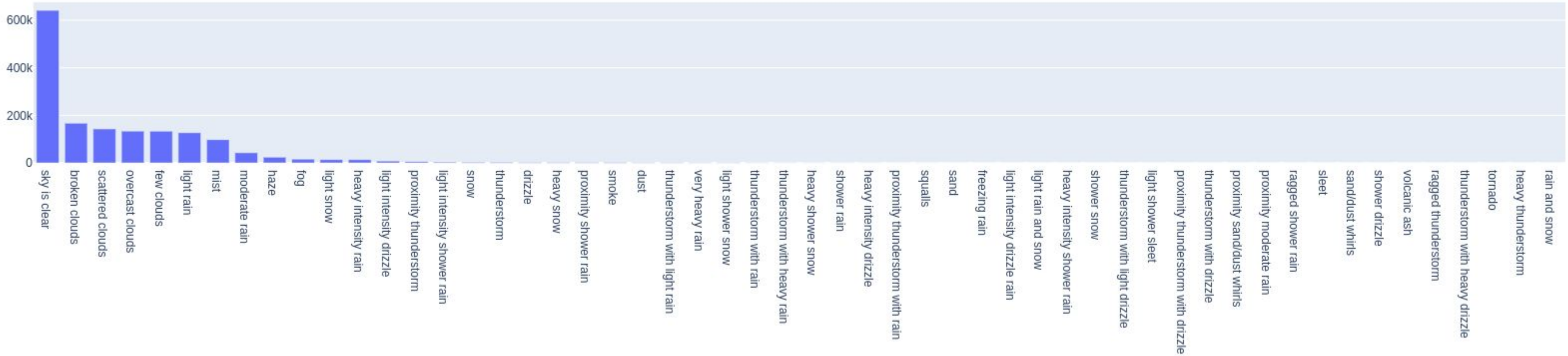
Dataset shape and schema

The new shape of the dataset is **1.629.108** rows by 11 columns

1. **datetime**: string
2. **humidity**: double
3. **pressure**: double
4. **temperature**: double
5. **wind_direction**: double
6. **wind_speed**: double
7. **weather_condition**: string
8. **city**: string
9. **country**: string
10. **latitude**: double
11. **longitude**: double

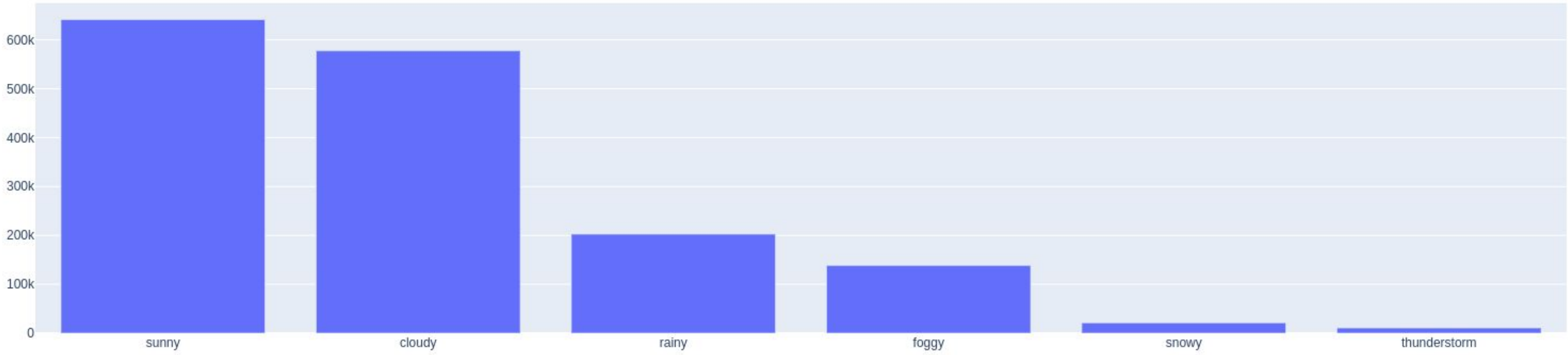
Aggregation of weather conditions classes

- The target classes are too sparse in the original dataset
- **54** different weather conditions
- Some occur few times and a lot of them are really similar between each other



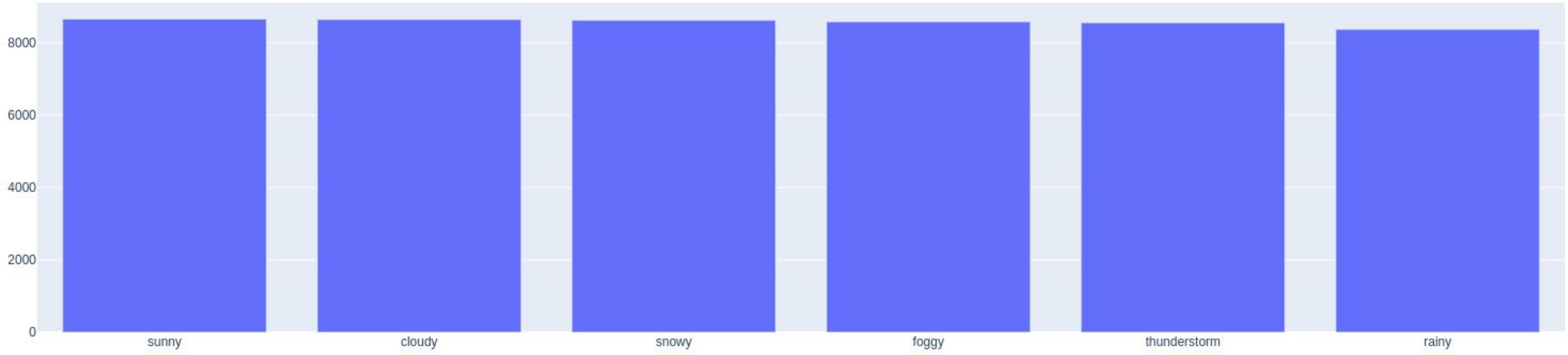
Aggregation of weather conditions classes

- Aggregate in **6** common weather condition classes:
  - thunderstorm
  - rainy
  - snowy
  - cloudy
  - foggy
  - sunny



Dataset undersampling

- The classes aggregation led to a huge class imbalance
- Avoid bias in the classification output, i.e. always predicting majority classes
- Undersampling instead of Oversampling, to have only real measured values



Machine Learning pipeline

1. Train and test split
   - **80%** for the train set
   - **20%** for the test set
2. Data encoding pipeline
   - *StringIndexer*
   - *OneHotEncoder*
   - *VectorAssembler*
   - *StandardScaler*
3. Machine Learning model
   - Decision Tree
   - Random Forest
   - Linear SVM (with One-vs-Rest)
   - Logistic Regression
4. Evaluation
   - Accuracy, Precision, Recall, F1-Score
   - Confusion matrix

Pipeline summarization + cross validation

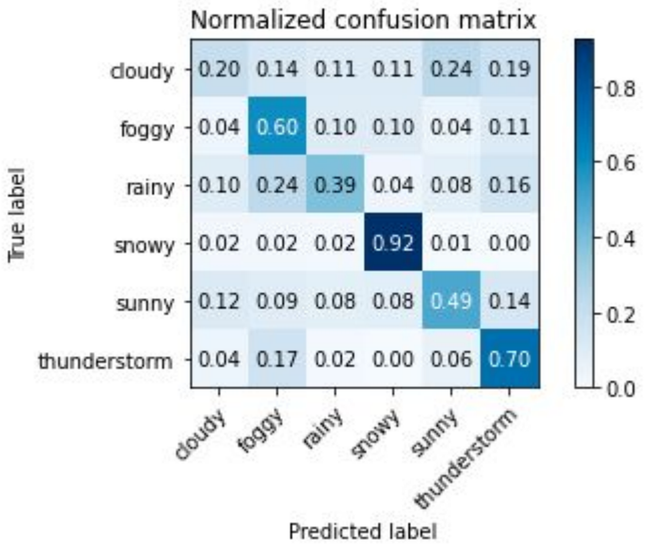
- Summarization of the training process and hyperparameters tuning
  - Creation of a new pipeline that includes both the data encoding and the models training
  - Addition of a new **K-Fold Cross Validation** step
  - Addition of an *IndexToString* that converts the numerical predictions in their label version

Machine Learning models

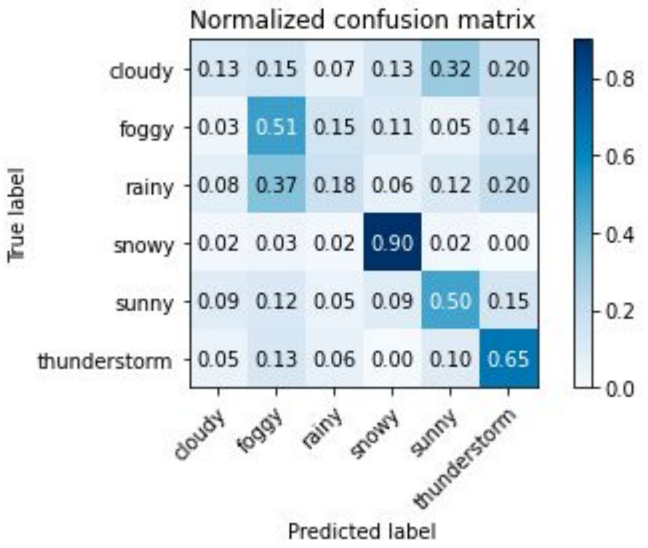
Random Forest	Logistic Regression
numTrees = 8 maxDepth = 50	maxIter = 1000 regParam = 0.0 elasticNetParam = 0.0
Accuracy = **0.550** Precision = **0.533** Recall = **0.551** F1-score = **0.542**	Accuracy = 0.477 Precision = 0.443 Recall = 0.477 F1-score = 0.459

Machine Learning models

Random Forest



Logistic Regression



OpenWeather comparison

- OpenWeather offers APIs for weather forecasting and historical data
- Comparison of the model predictions with actual weather forecasts by:
  - Create a *DataFrame* from an API response
  - Fit the Machine Learning pipeline with the new data

5 day forecast API

- 5 day forecast is available at any location or city
- It includes weather forecast data with 3-hour step
- The response is available in JSON or XML format

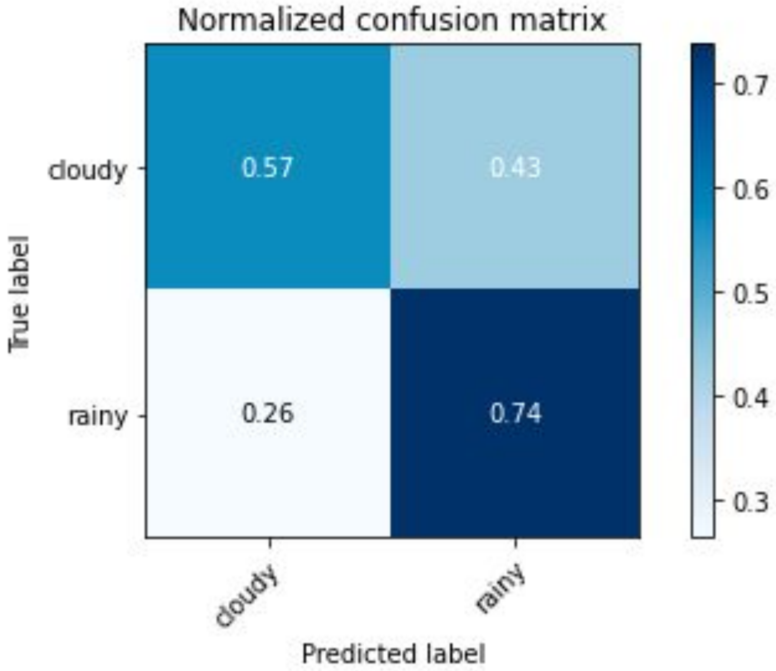

```
{
  "cod": "200",
  "message": 0,
  "cnt": 40,
  "list": [
    {
      "dt": 1596564000,
      "main": {
        "temp": 293.55,
        "feels_like": 293.13,
        "temp_min": 293.55,
        "temp_max": 294.05,
        "pressure": 1013,
        "sea_level": 1013,
        "grnd_level": 976,
        "humidity": 84,
        "temp_kf": -0.5
      },
      "weather": [
        {
          "id": 500,
          "main": "Rain",
          "description": "light rain",
          "icon": "10d"
        }
      ],
      "clouds": {
        "all": 38
      },
      "wind": {
        "speed": 4.35,
        "deg": 309,
        "gust": 7.87
      },
      "visibility": 10000,
      "pop": 0.49,
      "rain": {
        "3h": 0.53
      },
      "sys": {
        "pod": "d"
      },
      "dt_txt": "2020-08-04 18:00:00"
    },

    ...

  "city": {
    "id": 2643743,
    "name": "London",
    "coord": {
      "lat": 51.5073,
      "lon": -0.1277
    },
    "country": "GB",
    "timezone": 0,
    "sunrise": 1578384285,
    "sunset": 1578413272
  }
}
```

Predictions and forecasts comparison

	datetime	city	openweather_forecast	predicted_weather_condition
0	2021-06-15 06:00:00	Vancouver	rainy	rainy
1	2021-06-15 15:00:00	Vancouver	rainy	rainy
2	2021-06-18 15:00:00	Vancouver	cloudy	cloudy
3	2021-06-18 12:00:00	Vancouver	cloudy	rainy
4	2021-06-16 03:00:00	Vancouver	rainy	cloudy
5	2021-06-13 21:00:00	Vancouver	rainy	rainy
6	2021-06-17 09:00:00	Vancouver	cloudy	rainy
7	2021-06-15 00:00:00	Vancouver	cloudy	cloudy
8	2021-06-15 03:00:00	Vancouver	cloudy	cloudy
9	2021-06-16 12:00:00	Vancouver	cloudy	rainy
10	2021-06-14 00:00:00	Vancouver	rainy	rainy
11	2021-06-14 12:00:00	Vancouver	rainy	rainy
12	2021-06-16 06:00:00	Vancouver	rainy	rainy
13	2021-06-16 18:00:00	Vancouver	cloudy	cloudy
14	2021-06-17 00:00:00	Vancouver	cloudy	cloudy



Conclusions and future work

- Handling of huge raw datasets common problems
- Training and comparison of different models
- The best model was the Random Forest with 8 trees and a max depth of 50
- Obtained results similar to the ones provided from OpenWeather

- Collect more balanced data
- Exploit time series nature of the data with an RNN approach