Harvardx Data Science: Capstone - US City, sky will be clear!

Mathieu Labelle

MARCH 2020

Contents

1 INTRODUCTION

You will find here a document that is part of my submission for “Choose Your Own!” Project (*“the Project”*) of HarvardX PH125.09x: DATA SCIENCE: Capstone.
This *Project* will show how to apply some of the knowledge base and skills learned throughout the “Data Science” Series.
I wish you a very good read!

1.1 Summarize goal

Using “Historical Hourly Weather Data” dataset, which can be found by clicking here: kaggle.com, we will first create a workable dataset (the *Dataset*), and then we will work on a machine learning algorithm (“*the Algorithm*”) in order to predict in the City Studied if the sky will be clear in 24H from variables given by the *Dataset*. Our *Algorithm* will gave a binary outcome (sky Is clear: Yes or no, respectively 1 or 0) and we will compute Models accuracy to measure performance and select the most accurate ones.

We will train our *Algorithm* with multiples variables and different Models.

Here we choose New-York, but Data provided have 36 different Cities, so just change City_Studied by the one you want (you can find the list of city in 2.1.1).

```
#Define City_Studied as New.York
City_Studied <- "New.York"
```

1.2 Key Steps

To create our *Algorithm*, we will use the methodology we saw in previous course: “Machine Learning”. This methodology is to fit several different classes of Model to the data and select the one that gave the best accuracy.

The Models we chosen to test are:
* “One day looks like the next”
* “Head and Tail”
* Linear Discriminant Analysis
* Quadratic Discriminant Analysis * Generalized Linear Model * k-Nearest Neighbours
* Decision Tree

Find here the keys step of" the *Project*:
a. Extract Data from “.csv” files found in the folder loaded from kaggle.com
a. Explore the *Dataset*, clean and wrangle Data to prepare our analysis
b. Create a *Train* and *Test set* from *Dataset* for Cross validation
c. Train our *Algorithm* with *train_set* and using the *Test set* to calculate accuracy
d. Draw Conclusion based on Results
e. Discuss the report, its potential impact, its limitations, and future work

1.3 Getting Raw Data from Kaggle.com or GitHub

As we said; data could be found by clicking here: kaggle.com, you have to load the folder :“historical-hourly-weather-data” and extract its seven “.csv” files.
You can find the files also directly in github by clicking here: GitHub along with my three reports in “.Rmd”, “.pdf” and “.R”.

The files should be save in your working directory.

2 DATA WRANGLING AND CLEANING

2.1 Preliminary works on Data

Let's look at each file:

2.1.1 city_attributes:

Table 1: Head of city_attributes file

City	Country	Latitude	Longitude
Vancouver	Canada	49.24966	-123.11934
Portland	United States	45.52345	-122.67621
San Francisco	United States	37.77493	-122.41942
Seattle	United States	47.60621	-122.33207
Los Angeles	United States	34.05223	-118.24368
San Diego	United States	32.71533	-117.15726
Las Vegas	United States	36.17497	-115.13722
Phoenix	United States	33.44838	-112.07404
Albuquerque	United States	35.08449	-106.65114
Denver	United States	39.73915	-104.98470
San Antonio	United States	29.42412	-98.49363
Dallas	United States	32.78306	-96.80667
Houston	United States	29.76328	-95.36327
Kansas City	United States	39.09973	-94.57857
Minneapolis	United States	44.97997	-93.26384
Saint Louis	United States	38.62727	-90.19789
Chicago	United States	41.85003	-87.65005
Nashville	United States	36.16589	-86.78444
Indianapolis	United States	39.76838	-86.15804
Atlanta	United States	33.74900	-84.38798
Detroit	United States	42.33143	-83.04575
Jacksonville	United States	30.33218	-81.65565
Charlotte	United States	35.22709	-80.84313
Miami	United States	25.77427	-80.19366
Pittsburgh	United States	40.44062	-79.99589
Toronto	Canada	43.70011	-79.41630
Philadelphia	United States	39.95234	-75.16379
New York	United States	40.71427	-74.00597
Montreal	Canada	45.50884	-73.58781
Boston	United States	42.35843	-71.05977
Beersheba	Israel	31.25181	34.79130
Tel Aviv District	Israel	32.08333	34.80000
Eilat	Israel	29.55805	34.94821
Haifa	Israel	32.81556	34.98917
Nahariyya	Israel	33.00586	35.09409
Jerusalem	Israel	31.76904	35.21633

It shows us Latitude and Longitude and Country of each City. As we will study City_Studied, there is no need for us to include data from this file.

2.1.2 humidity:

Table 2: Head of humidity file

datetime	Vancouver	Portland	San.Francisco	Seattle	Los.Angeles	San.Diego
2012-10-01 12:00:00	NA	NA	NA	NA	NA	NA
2012-10-01 13:00:00	76	81	88	81	88	82
2012-10-01 14:00:00	76	80	87	80	88	81

It shows hourly humidity (in %) level by City, In this file, for our *Algorithm*, we will select only the City_Studied. We will need also to discard the first line with the NAs.

2.1.3 pressure:

Table 3: Head of pressure file

datetime	Vancouver	Portland	San.Francisco	Seattle	Los.Angeles	San.Diego
2012-10-01 12:00:00	NA	NA	NA	NA	NA	NA
2012-10-01 13:00:00	NA	1024	1009	1027	1013	1013
2012-10-01 14:00:00	NA	1024	1009	1027	1013	1013

It shows hourly pressure (in Bar) level by City, In this file, for our *Algorithm*, we will select only the City_Studied. We will need also to discard the first line with the NAs.

2.1.4 temperature:

Table 4: Head of temperature file

datetime	Vancouver	Portland	San.Francisco	Seattle	Los.Angeles	San.Diego
2012-10-01 12:00:00	NA	NA	NA	NA	NA	NA
2012-10-01 13:00:00	284.630	282.0800	289.480	281.8000	291.8700	291.5300
2012-10-01 14:00:00	284.629	282.0833	289.475	281.7972	291.8682	291.5335

It shows hourly temperature (in Kelvin) level by City, In this file, for our *Algorithm*, we will select only the City_Studied. We will need also to discard the first line with the NAs. We will also round the temperature to the nearest with only one decimal to get more entries for the same temperature.

2.1.5 weather_description:

Table 5: Head of weather_description file

datetime	Vancouver	Portland	San.Francisco	Seattle	Los.Angeles
2012-10-01 12:00:00					
2012-10-01 13:00:00	mist	scattered clouds	light rain	sky is clear	mist
2012-10-01 14:00:00	broken clouds	scattered clouds	sky is clear	sky is clear	sky is clear

It shows hourly Weather Description by City, In this file, for our *Algorithm*, we will select only the City_Studied. We will need also to discard the first line with the missing information. Also we will need to change “sky is clear” with 1 and otherwise 0.

2.1.6 wind_direction:

Table 6: Head of wind_direction file

datetime	Vancouver	Portland	San.Francisco	Seattle	Los.Angeles	San.Diego
2012-10-01 12:00:00	NA	NA	NA	NA	NA	NA
2012-10-01 13:00:00	0	0	150	0	0	0
2012-10-01 14:00:00	6	4	147	2	0	0

It shows wind direction in degree by City, In this file, for our *Algorithm*, we will select only the city :City_Studied. We will need also to discard the first line with the NAs.

2.1.7 wind_speed:

Table 7: Head of wind_speed file

datetime	Vancouver	Portland	San.Francisco	Seattle	Los.Angeles	San.Diego
2012-10-01 12:00:00	NA	NA	NA	NA	NA	NA
2012-10-01 13:00:00	0	0	2	0	0	0
2012-10-01 14:00:00	0	0	2	0	0	0

It shows wind speed by City, In this file, for our *Algorithm*, we will select only the :City_Studied. We will need also to discard the first line with the NAs.

2.1.8 Work on the six DataFrames

We will first keep only City_Studied in each 6 DataFrames.

We will then change the columns names to datetime and name of the observable.

2.2 Create our *Dataset* from the six DataFrames

Create our *Dataset* with the seven variables (datetime, humidity, pressure, temperature, weather_des, wind_Dir, wind_speed). And let's have a look of the head:

Table 8: Dataset Temp

datetime	weather_des	pressure	temperature	humidity	wind_dir	wind_speed
2012-10-01 12:00:00		NA	NA	NA	NA	NA
2012-10-01 13:00:00	few clouds	1012	288.2200	58	260	7
2012-10-01 14:00:00	few clouds	1012	288.2477	57	260	7
2012-10-01 15:00:00	few clouds	1012	288.3269	57	260	7
2012-10-01 16:00:00	few clouds	1012	288.4062	57	260	7
2012-10-01 17:00:00	few clouds	1012	288.4855	57	261	6

2.3 Clean and Wrangle our *Dataset*

We need to check class of each variable:

Table 9: Columns

	Class
datetime	factor
weather_des	factor
pressure	numeric
temperature	numeric
humidity	numeric
wind_dir	numeric
wind_speed	numeric

2.3.1 Populate missing entries

There are missing entries for some variables, we will change the NAs by previous observation for each variable.

2.3.2 Change datetime to the correct Date and Time format

We have to change format and class of datetime to proper date and time.

Let's introduce Month and Hour to study seasonality

2.3.3 Populate each entry with the sky_clear

Let's change string "sky is clear" into a factor "sky_clear" with "1" otherwise by "0".

For each entries add the sky_clear in 24H, so for each entry we know if the sky is clear in 24H.

2.3.4 Rounding temperature

To avoid having too few observations for a given temperature we will round temperature to the nearest with one decimal.

2.3.5 *Dataset* ready for analysis

Let's now have a look at the head of our *Dataset*:

Table 10: Dataset (Not showing 1st Columns: datetime)

pressure	temperature	humidity	wind_dir	wind_speed	Month	Hour	sky_clear	Pred_24H
1012	288	58	260	7	10	13	0	0
1012	288	57	260	7	10	14	0	0
1012	288	57	260	7	10	15	0	0
1012	288	57	260	7	10	16	0	1
1012	288	57	261	6	10	17	0	1
1012	289	56	261	6	10	18	0	0

2.3.6 Clean our workspace

We will remove all useless DataFrames to keep and save our *Dataset*.

3 DATA EXPLORATION AND VISUALIZATION

3.1 General

```
## [1] "We have: 45228 records (datetime), for the city: New.York"
```

What is the probability that the “sky is clear” in 24H out of all entries:

Table 11: Probability to get a clear sky in 24H

Probability
0.2565004

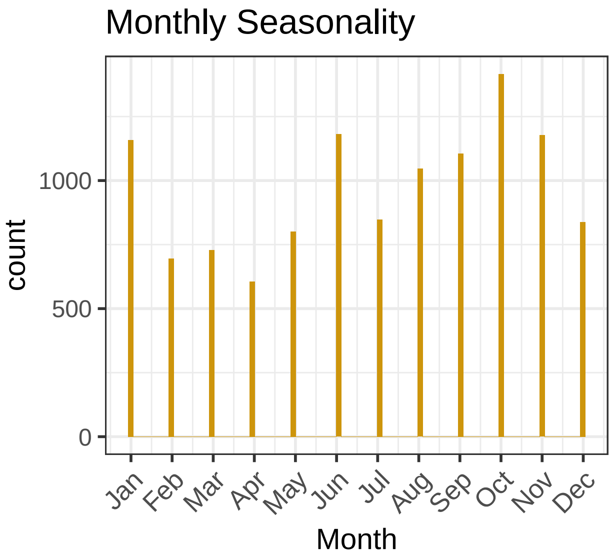
What is the probability that the “sky is clear” in 24H knowing that it is clear now:

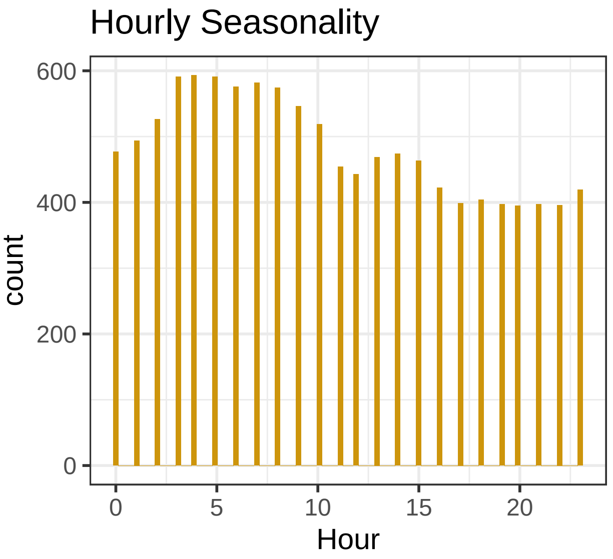
Table 12: Probability to get a clear sky in 24H as it is now

Probability
0.4428067

3.2 Seasonality

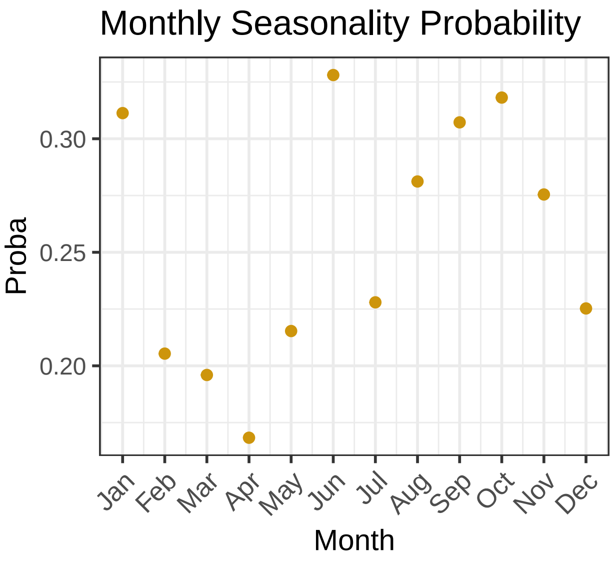
Let’s look at distribution of clear_sky per Month and per Hour.

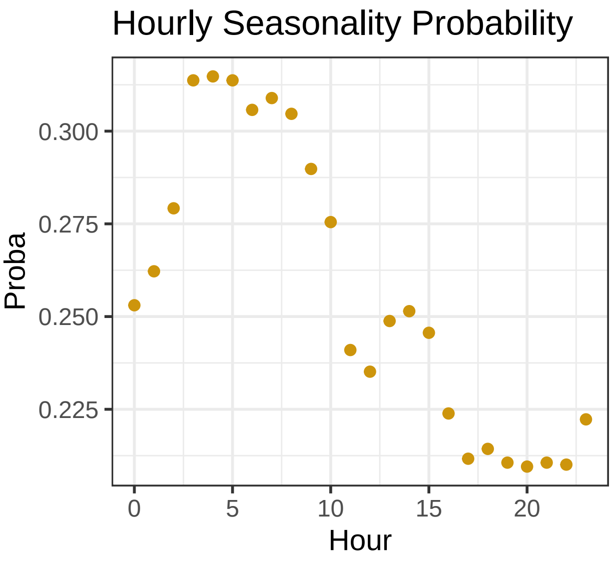




There are obviously and logically: seasonality by Month as by Hour, just to mention because Months are compose by 30 or 31 days (Even 28 or 29 for February), we have little bit more observations for longer months, so likely to have little bit clear sky too.

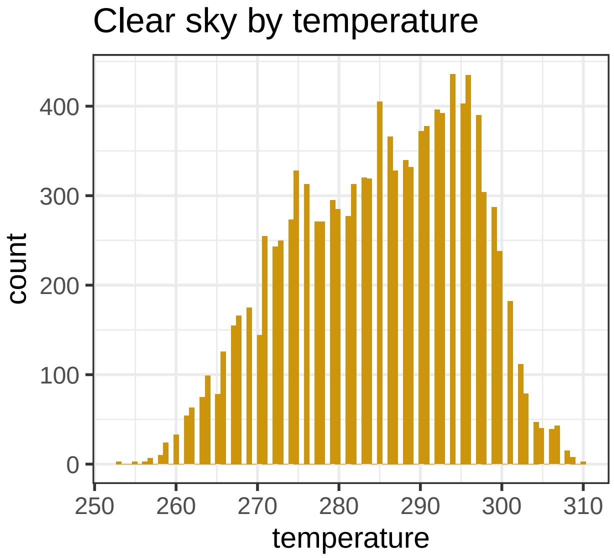
Let’s look at the Probability to get a clear sky by Month and by Hour.

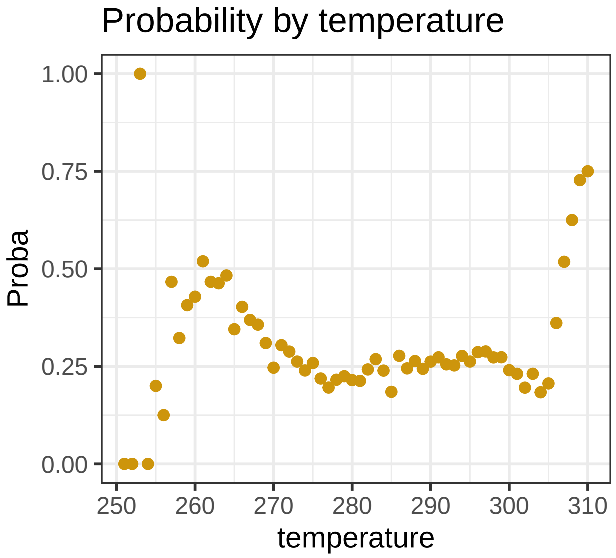

Monthly Seasonality Probability


Hourly Seasonality Probability

3.3 Temperature, Pressure and Humidity

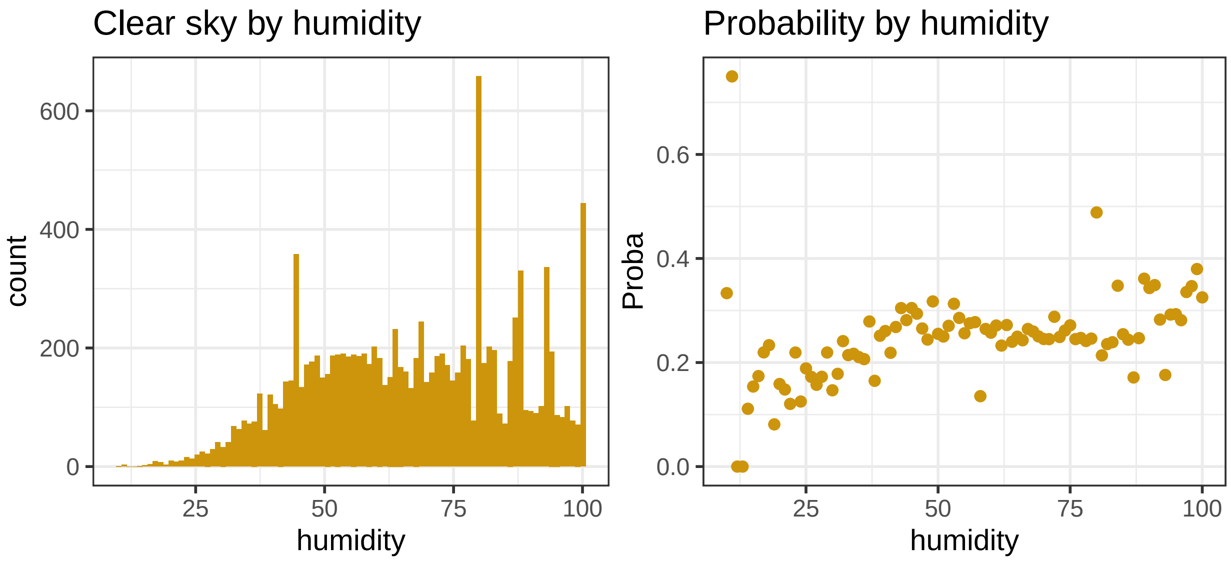
First we can study the temperature effect to get a clear sky in 24H. Let's have a look at the distribution of clear sky by temperature and for each temperature have a look at the probability to get a clear sky in 24H.


Clear sky by temperature


Probability by temperature

Except extreme and singular point, "Normal temperature" get all the same probability. Likely that a decision tree Model will fit as <270 and >305 probability are higher to get a clear sky.

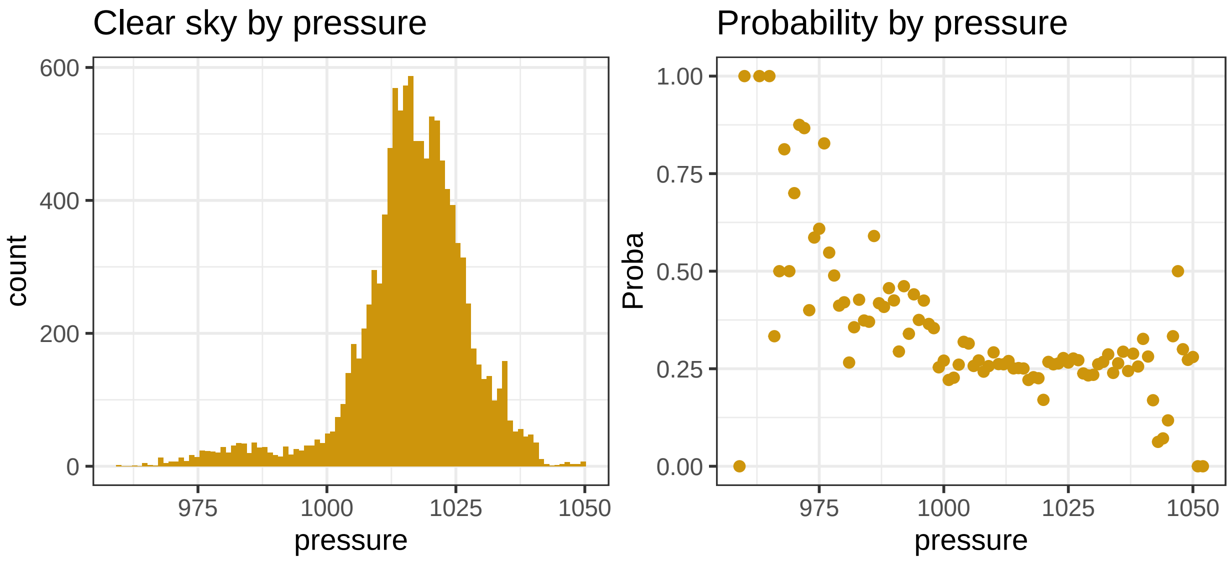
Let's look at distribution of clear_sky per usual given measures by meteorologist: humidity and pressure.



Clear sky by humidity

Probability by humidity

Here also probability are lower or higher when humidity is lower than ~30 or higher than ~80 respectively, so a decision tree Model should fit.
Now it is time to look at probability to get a clear sky in 24H by temperature and pressure



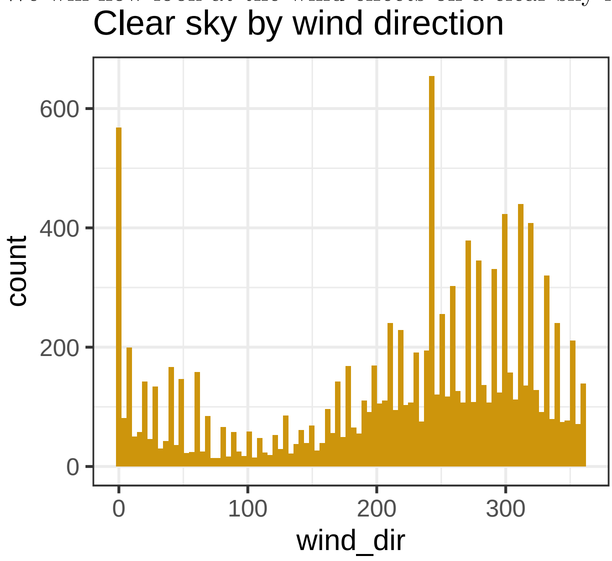
Clear sky by pressure

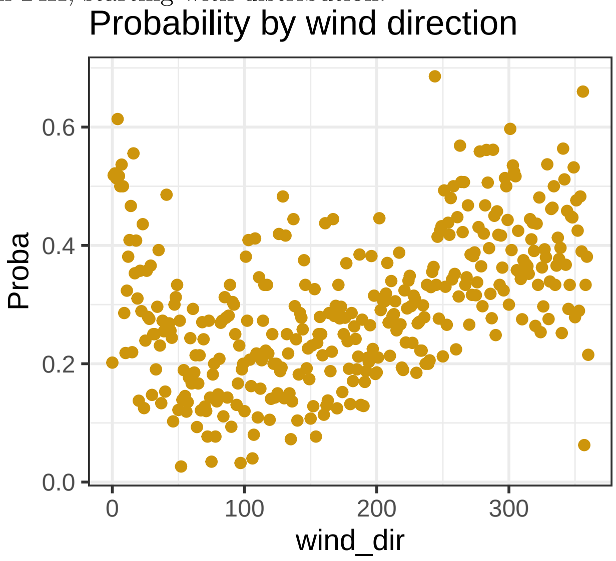
Probability by pressure

Here probability are much higher when pressure is lower than 1000, so a decision tree Model should fit.

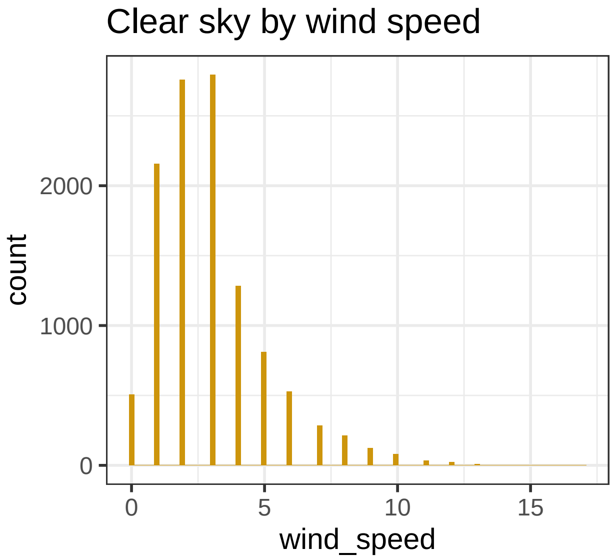
3.4 Wind effect

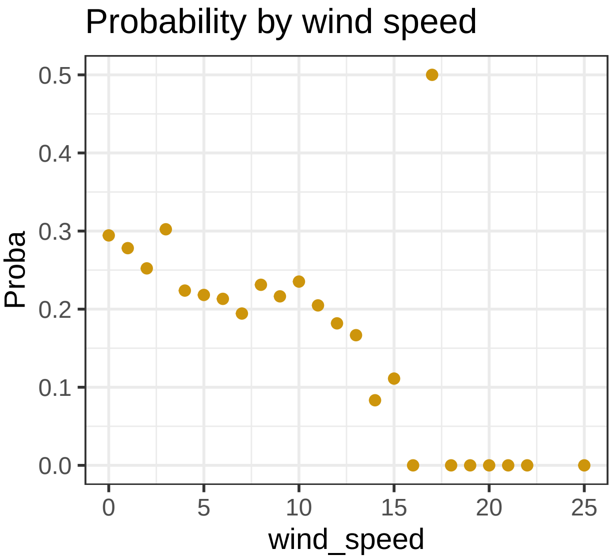
We will now look at the wind effects on a clear sky in 24H, starting with distribution.


Clear sky by wind direction


Probability by wind direction

There is also a higher probability for extreme case of pressure.


Clear sky by wind speed


Probability by wind speed

Wind speed play a role when values are >10, and chance to get a clear sky are lower.

4 MACHINE LEARNING MODEL

4.1 Creating a train and a test set from our ***Dataset***

We will use cross-validation to built our *Algorithm*, we will need to create two sets within *Dataset*, one to train our Models, another one to evaluate its predictions, respectively train_set and test_set. For the partition of our *Dataset* will set the seed to 1, in order to constantly get the same partition. The partition will be 80% and 20% for train set and test set respectively.

4.2 “One day looks like the next”

Let’s start our machine Learning *Algorithm* with our first Model. It will be a really simple one, as we will just consider that the state of the sky at datetime will be the same in 24H.
The accuracy of the Model is given by the probability to get both state of the sky clear.

Table 13: Models Accuracy

Model	Accuracy
One day looks like the next	0.714

That’s give a relatively good accuracy above 2 chances out of 3.

4.3 Simple and Biased “Head and Tail”

4.3.1 Simple Coin

Our second Model, will be like flipping a coin, Head = 1 (sky clear), Tail = 0 (any other weather description).

Table 14: Models Accuracy

Model	Accuracy
One day looks like the next	0.7140000
Head and Tail	0.4966844

This Model gave an obvious result of 1 chance on 2, not really useful, but we will improve this Model just below.

4.3.1 Biased Coin

We can introduce a bias in our sample, if we use the global probability to get a clear sky (compute in 3.1) instead of 0.5 (as a normal coin)

Table 15: Models Accuracy

Model	Accuracy
One day looks like the next	0.7140000
Head and Tail	0.4966844
Biased: Head and Tail	0.6123549

We are a bit luckier than flipping a coin, but not as good as our very first simple Model.

4.4 Linear Discriminant Analysis

Let’s now try a Linear regression on the different parameters we have now to predict the state of the sky in 24H. Starting with a simple “Linear Discriminant Analysis” on temperature, pressure, humidity, wind direction, wind speed.

Table 16: Models Accuracy

Model	Accuracy
One day looks like the next	0.7140000
Head and Tail	0.4966844
Biased: Head and Tail	0.6123549
Linear Discriminant Analysis	0.7469054

This is a jump in accuracy.

4.5 Quadratic Discriminant Analysis

Let’s now try Quadratic regression on the different parameters we have now to predict the state of the sky in 24H. Starting with a simple "Linear Discriminant Analysis on temperature, pressure, humidity, wind direction, wind speed.

Table 17: Models Accuracy

Model	Accuracy
One day looks like the next	0.7140000
Head and Tail	0.4966844
Biased: Head and Tail	0.6123549
Linear Discriminant Analysis	0.7469054
Quadratic Discriminant Analysis	0.7449160

Not a surprise to see result, close to the previous Model.

4.6 Generalized Linear Model

Let’s now try General Linear Model on easy observable variables as temperature, Month, Hour and the state of the sky now.

Table 18: Models Accuracy

Model	Accuracy
One day looks like the next	0.7140000
Head and Tail	0.4966844
Biased: Head and Tail	0.6123549
Linear Discriminant Analysis	0.7469054
Quadratic Discriminant Analysis	0.7449160
General Linear Model with 4 observations	0.7484527

Better because we introduce the state of sky at datetime as variable in the Model.

4.7 k-Nearest Neighbours

We will try a k-Nearest Neighbours Model on our four simpliest parameters: temperature, Month, Hour and the state of the sky at datetime.

Table 19: Models Accuracy

Model	Accuracy
One day looks like the next	0.7140000
Head and Tail	0.4966844
Biased: Head and Tail	0.6123549
Linear Discriminant Analysis	0.7469054
Quadratic Discriminant Analysis	0.7449160
General Linear Model with 4 observations	0.7484527
k-Nearest Neighbours Model	0.7500000

Once again so improvement for the accuracy we are now at 3 chances on 4 to have a good prediction.

4.8 Decision Tree (easy observable parameters)

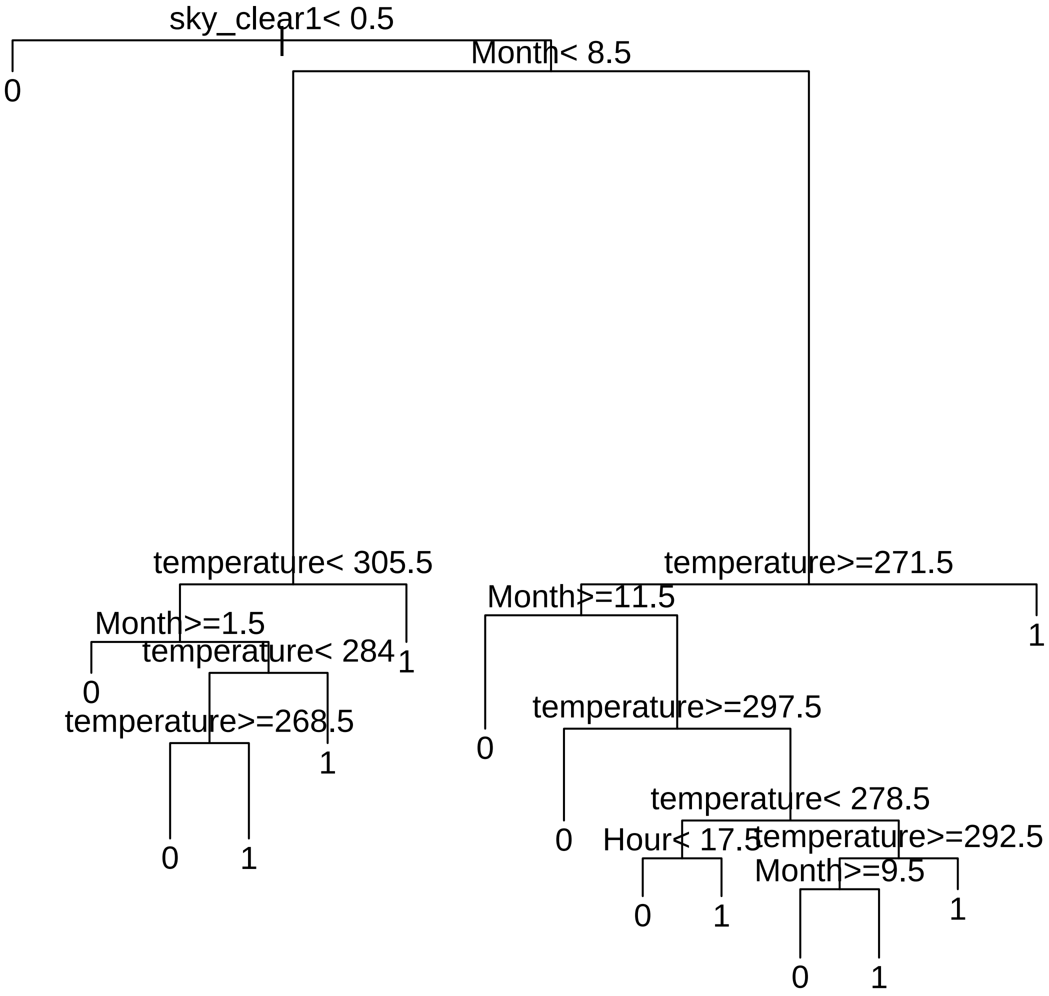
Let's now try Decision Tree Model on parameters that are easy to observe on a daily basis without complicate instruments. I propose to study a Decision Tree with only the following variables: temperature, Month, Hour and if the sky is clear at datetime.

Table 20: Models Accuracy

Model	Accuracy
One day looks like the next	0.7140000
Head and Tail	0.4966844
Biased: Head and Tail	0.6123549
Linear Discriminant Analysis	0.7469054
Quadratic Discriminant Analysis	0.7449160
General Linear Model with 4 observations	0.7484527
k-Nearest Neighbours Model	0.7500000
Easy to observe, Decision Tree Model	0.7568523

Good performance, and easy to use, you just need to measures the temperature, and you get a good accuracy. A Model you can use every day (even without computer!).

You can find the Decision Tree here:



4.9 Decision Tree with all variables

Let's now try a Decision Tree Model all parameters.

Table 21: Models Accuracy

Model	Accuracy
One day looks like the next	0.7140000
Head and Tail	0.4966844
Biased: Head and Tail	0.6123549
Linear Discriminant Analysis	0.7469054
Quadratic Discriminant Analysis	0.7449160
General Linear Model with 4 observations	0.7484527
k-Nearest Neighbours Model	0.7500000
Easy to observe, Decision Tree Model	0.7568523
Decision Tree Model all Variables	0.7582891

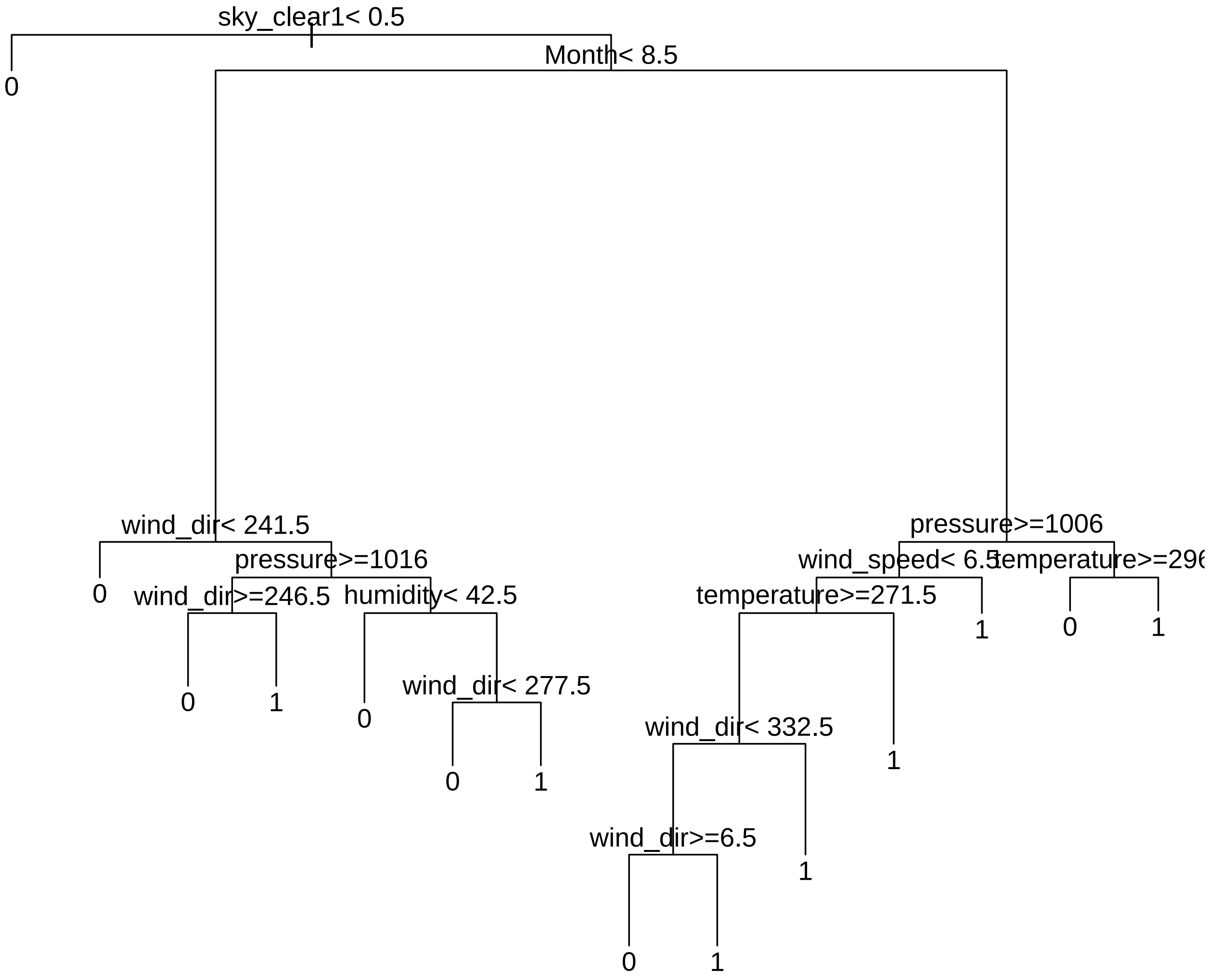
Our best Model so far, let's look at the most important variables:

Table 22: Most important Variables

Variable	Importance
sky_clear1	865.58675
wind_dir	574.15731
pressure	274.72306
temperature	195.17646
Month	156.06510
humidity	99.27813
Hour	90.02644
wind_speed	84.75108

It is not a surprise to find the state of the sky at datetime as the most important one. Surprisingly temperature as humidity do not play a massive role.

You can find the Decision Tree here:



4 CONCLUSION

Decision Tree give here the best result to predict if the sky will be clear in 24H, it is a bit better than “Regression Models”. If you look at the most important variable for the Tree, it is by far the state of the sky at datetime, that’s why the simple Model: if sky is clear it will be clear in 24H, is working well too. By far my prefer one, despite its lower accuracy, is the Decision Tree with only 4 parameters, with only one measure (temperature), it gave a good result, decision tree figure can be printed and easy to use!

Table 23: Models Accuracy

Model	Accuracy
One day looks like the next	0.7140000
Head and Tail	0.4966844
Biased: Head and Tail	0.6123549
Linear Discriminant Analysis	0.7469054
Quadratic Discriminant Analysis	0.7449160
General Linear Model with 4 observations	0.7484527
k-Nearest Neighbours Model	0.7500000
Easy to observe, Decision Tree Model	0.7568523
Decision Tree Model all Variables	0.7582891

I would have be very happy to try other Model, like Random Forest but computation time make it difficult to run on a personal computer. Internet shows so many different Models in R, it is impressive, there is obviously ways to improve prediction, but here the aim was to show code (R, R-MD) and knowledge rather than accuracy in term of weather modelling.

Another way to improve our Model is take into account other cities state of sky knowing the longitude, latitude, wind direction and speed, we may improve our prediction.

The Models we choose are from different category, and show what we learned during theses fabulous eight courses on Edx. Going from absolutely no knowledge of R before starting, I feel relatively confident now, and it is very useful to perform analysis at home and at work. I replace Excel for all data analysis by R/R-Studio, as it is much more flexible and much more powerful.

As my first own project in data science, it is a small one, but I learn a lot from it. I use all learnings obtain during the previous eight courses of Havardx “Data Science”, honestly it was great fun. And despite its modesty, I am relatively proud of my first long R code, my first use of R Markdown and also my first publish in LaTeX (in fact my second use of them, as I did MovieLens project before).

Thank you for your time reading this document, I hope you find some useful informations.