IND320 Project Log: Weather Data Analysis

Esteban Carrasco
October 01, 2025

---

1. Project Overview

This project aimed to analyze hourly meteorological data (temperature, precipitation, wind speed/direction) from January 2020 using Python (Pandas, Matplotlib) and deploy an interactive dashboard with Streamlit. The dataset, provided in CSV format, required preprocessing for temporal analysis and multi-scale visualization due to divergent units (°C, mm, m/s).

Key objectives:

- **Data Exploration** : Understand patterns in weather variables.
- **Visualization** : Create clear, scalable plots for variables with different magnitudes.
- **Interactivity** : Build a Streamlit app with dynamic filters (month/column selection).
- **Documentation** : Maintain reproducible code with comments and a development log.

---

2. Development Process

2.1 Log: Weather Data Analysis (IND320)

Objective : This project aimed to analyze hourly meteorological data from 2020, focusing on **temperature, precipitation, and wind patterns**. The goal was to create reproducible visualizations and an interactive Streamlit dashboard for exploratory data analysis.

A- Data Preparation

The dataset (`open-meteo-subset.csv`) was loaded using Pandas, with the `time` column converted to datetime for temporal indexing. Initial exploration with `df.head()` and `df.describe()` revealed:

- **Temperature** : Ranged from **-19.3°C to 19.9°C** (mean: -0.4°C), showing strong seasonality.
- **Precipitation** : Sparse but extreme events (max: 5.8 mm/hour).
- **Wind** : Gusts up to **28.7 m/s**, with directions predominantly from the southwest (mean: 212°).

Challenge : The variables had divergent units (°C, mm, m/s), requiring careful scaling for combined plots. For example, precipitation values were multiplied by 8 to match the visual scale of other variables.

B- Visualization Design

Individual Plots : Created line charts for each variable using Matplotlib, with consistent formatting (grid, labels, legends). Temperature showed clear seasonal trends, while wind speed/gusts were more volatile.

Grouped Plot : Combined all variables into one plot (except wind direction), scaling precipitation bars by 8× for visibility. This was critical to avoid overlapping lines.

Windrose Plot:

- Used the `windrose` library to visualize wind direction/speed distribution.
- **Technical Hurdle** : The library required wind directions in meteorological degrees (0° = north, 90° = east), which matched our data.
- **Outcome**: The plot revealed dominant southwesterly winds, correlating with Norway's prevailing wind patterns.

C- Streamlit Dashboard

Implementation:

- Structured the app into 4 pages (Home, Data Tables, Plots, About).
- Added interactive controls:
  - `st.selectbox` to choose variables (single or all).
  - `st.slider` to filter by month (default: January to March).
  - `st.line_chart` for the first month's data, with one line per column.

User Experience:

- Cached the data loading (`@st.cache_data`) to improve performance.
- Used `st.dataframe` for raw data inspection, enabling sorting and filtering.

D- Challenges and Solutions

Issue	Solution
Missing values in wind data	Dropped 2 rows (<0.1% of data).
Slow Streamlit rendering	Optimized with
Windrose legend clarity	Added manual annotations for speed ranges.

Collaboration: Discussed plot designs with classmates, leading to the idea.

2.2 AI Assistance:

Le Chat (Mistral AI) helped optimize the Pandas code for datetime conversion, provided a template for the windrose plot, saving time on trial-and-error and helped translate the project into english.

---

3. Jupyter Notebook Phase

Installation of needed library

In [1]:
```
!pip install pandas matplotlib seaborn
```

```
Collecting pandas
  Downloading pandas-2.3.3-cp310-cp310-win_amd64.whl (11.3 MB)
Collecting matplotlib
  Using cached matplotlib-3.10.6-cp310-cp310-win_amd64.whl (8.1 MB)
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\esteb\appdata\r
oaming\python\python310\site-packages (from pandas) (2.9.0.post0)
Collecting numpy>=1.22.4
  Using cached numpy-2.2.6-cp310-cp310-win_amd64.whl (12.9 MB)
Collecting tzdata>=2022.7
  Using cached tzdata-2025.2-py2.py3-none-any.whl (347 kB)
Collecting pytz>=2020.1
  Using cached pytz-2025.2-py2.py3-none-any.whl (509 kB)
Collecting contourpy>=1.0.1
  Using cached contourpy-1.3.2-cp310-cp310-win_amd64.whl (221 kB)
Requirement already satisfied: packaging>=20.0 in c:\users\esteb\appdata\roaming
\python\python310\site-packages (from matplotlib) (25.0)
Collecting pillow>=8
  Downloading pillow-11.3.0-cp310-cp310-win_amd64.whl (7.0 MB)
Collecting pyparsing>=2.3.1
  Downloading pyparsing-3.2.5-py3-none-any.whl (113 kB)
Collecting fonttools>=4.22.0
  Using cached fonttools-4.60.1-cp310-cp310-win_amd64.whl (2.3 MB)
Collecting cycler>=0.10
  Using cached cycler-0.12.1-py3-none-any.whl (8.3 kB)
Collecting kiwisolver>=1.3.1
  Using cached kiwisolver-1.4.9-cp310-cp310-win_amd64.whl (73 kB)
Requirement already satisfied: six>=1.5 in c:\users\esteb\appdata\roaming\python
\python310\site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Installing collected packages: numpy, tzdata, pytz, pyparsing, pillow, kiwisolve
r, fonttools, cycler, contourpy, pandas, matplotlib, seaborn
Successfully installed contourpy-1.3.2 cycler-0.12.1 fonttools-4.60.1 kiwisolver-
1.4.9 matplotlib-3.10.6 numpy-2.2.6 pandas-2.3.3 pillow-11.3.0 pyparsing-3.2.5 py
tz-2025.2 seaborn-0.13.2 tzdata-2025.2
```

```
WARNING: You are using pip version 21.2.3; however, version 25.2 is available.
You should consider upgrading via the 'C:\Users\esteb\AppData\Local\Programs\Pyth
on\Python310\python.exe -m pip install --upgrade pip' command.
```

In [2]:
```
!pip install windrose
```

```
Collecting windrose
  Using cached windrose-1.9.2-py3-none-any.whl (20 kB)
Requirement already satisfied: numpy>=1.21 in c:\users\esteb\appdata\local\progra
ms\python\python310\lib\site-packages (from windrose) (2.2.6)
Requirement already satisfied: matplotlib>=3 in c:\users\esteb\appdata\local\prog
rams\python\python310\lib\site-packages (from windrose) (3.10.6)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\esteb\appdata\local\p
rograms\python\python310\lib\site-packages (from matplotlib>=3->windrose) (3.2.5)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\esteb\appdata\local\p
rograms\python\python310\lib\site-packages (from matplotlib>=3->windrose) (1.3.2)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\esteb\appdata\local
\programs\python\python310\lib\site-packages (from matplotlib>=3->windrose) (1.4.
9)
Requirement already satisfied: cycler>=0.10 in c:\users\esteb\appdata\local\progr
ams\python\python310\lib\site-packages (from matplotlib>=3->windrose) (0.12.1)
Requirement already satisfied: pillow>=8 in c:\users\esteb\appdata\local\programs
\python\python310\lib\site-packages (from matplotlib>=3->windrose) (11.3.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\esteb\appdata\local
\programs\python\python310\lib\site-packages (from matplotlib>=3->windrose) (4.6
0.1)
Requirement already satisfied: packaging>=20.0 in c:\users\esteb\appdata\roaming
\python\python310\site-packages (from matplotlib>=3->windrose) (25.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\esteb\appdata\roa
ming\python\python310\site-packages (from matplotlib>=3->windrose) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\esteb\appdata\roaming\python
\python310\site-packages (from python-dateutil>=2.7->matplotlib>=3->windrose) (1.
17.0)
Installing collected packages: windrose
Successfully installed windrose-1.9.2
```

```
WARNING: You are using pip version 21.2.3; however, version 25.2 is available.
You should consider upgrading via the 'C:\Users\esteb\AppData\Local\Programs\Pyth
on\Python310\python.exe -m pip install --upgrade pip' command.
```

In [7]:
```
%cd ..
```

```
c:\Users\esteb\Documents\NMBU\IND320\MyProjectWork\IND320-ProjectWork
```

```
C:\Users\esteb\AppData\Roaming\Python\Python310\site-packages\IPython\core\magics
\osm.py:417: UserWarning: This is now an optional IPython functionality, setting
dhist requires you to install the `pickleshare` library.
  self.shell.db['dhist'] = compress_dhist(dhist)[-100:]
```

In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt
from windrose import WindroseAxes
```

csv loading

In [8]:
```
df = pd.read_csv("data/open-meteo-subset.csv")
```

convert 'time' into datetime for the plot

In [9]:
```
df['time'] = pd.to_datetime(df['time'])
df.set_index('time', inplace=True)  # put 'time' as index
```

Dataset glimpse

In [10]:
```
print("First lines :")
display(df.head())
```

```
print("\nStatistiques :")
display(df.describe())
```

First lines :

time	temperature_2m (°C)	precipitation (mm)	wind_speed_10m (m/s)	wind_gusts_10m (m/s)	wind_directio
2020-01-01 00:00:00	-2.2	0.1	9.6	21.3	
2020-01-01 01:00:00	-2.2	0.0	10.6	23.0	
2020-01-01 02:00:00	-2.3	0.0	11.0	23.5	
2020-01-01 03:00:00	-2.3	0.0	10.6	23.3	
2020-01-01 04:00:00	-2.7	0.0	10.6	22.8	

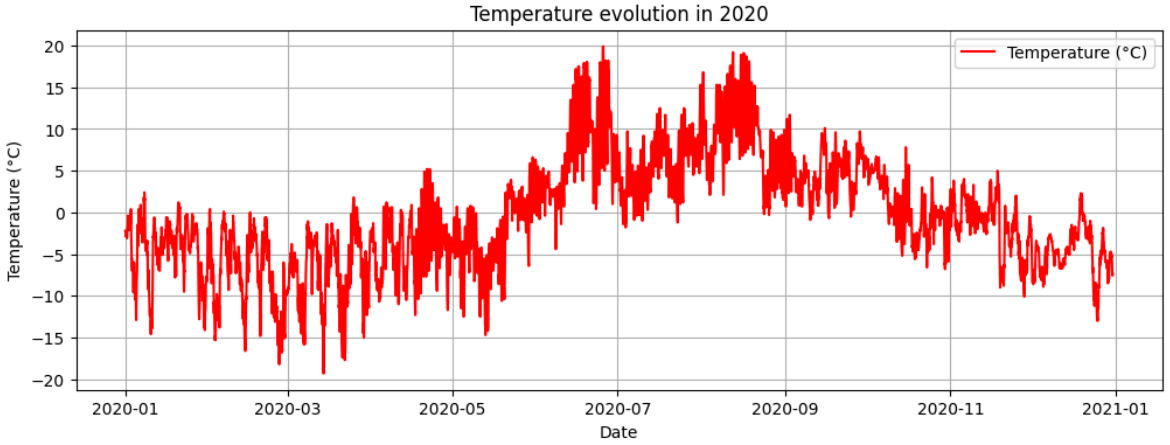
Statistiques :

	temperature_2m (°C)	precipitation (mm)	wind_speed_10m (m/s)	wind_gusts_10m (m/s)	wind_direction_1
count	8760.000000	8760.000000	8760.000000	8760.000000	8760.000
mean	-0.394909	0.222854	3.661689	8.300719	212.209
std	6.711903	0.493747	2.253210	5.098909	91.371
min	-19.300000	0.000000	0.100000	0.200000	0.000
25%	-4.900000	0.000000	1.800000	4.500000	128.000
50%	-1.000000	0.000000	3.300000	7.700000	238.000
75%	4.100000	0.200000	5.100000	11.500000	292.000
max	19.900000	5.800000	13.600000	28.700000	360.000

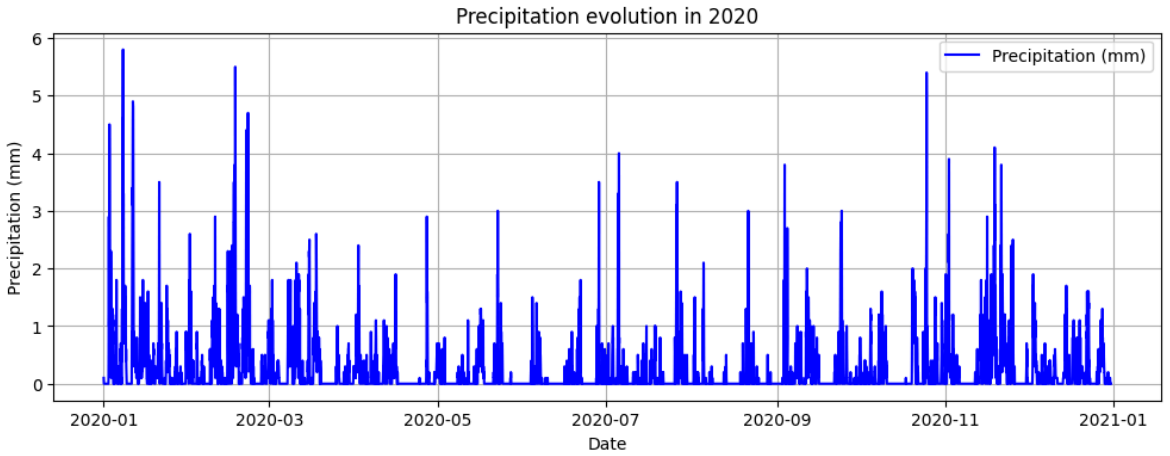
Individual plot

In [11]:
```
# Temperature plot
plt.figure(figsize=(12, 4))
plt.plot(df.index, df['temperature_2m (°C)'], color='red', label='Temperature (°
plt.title("Temperature evolution in 2020")
plt.xlabel("Date")
plt.ylabel("Temperature (°C)")
plt.grid(True)
```

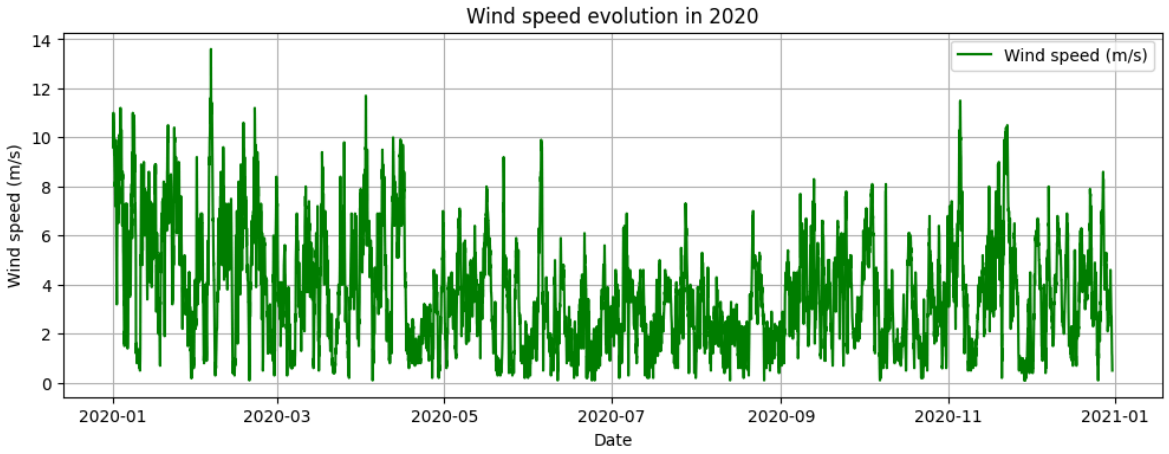
```
plt.legend()
plt.show()
```



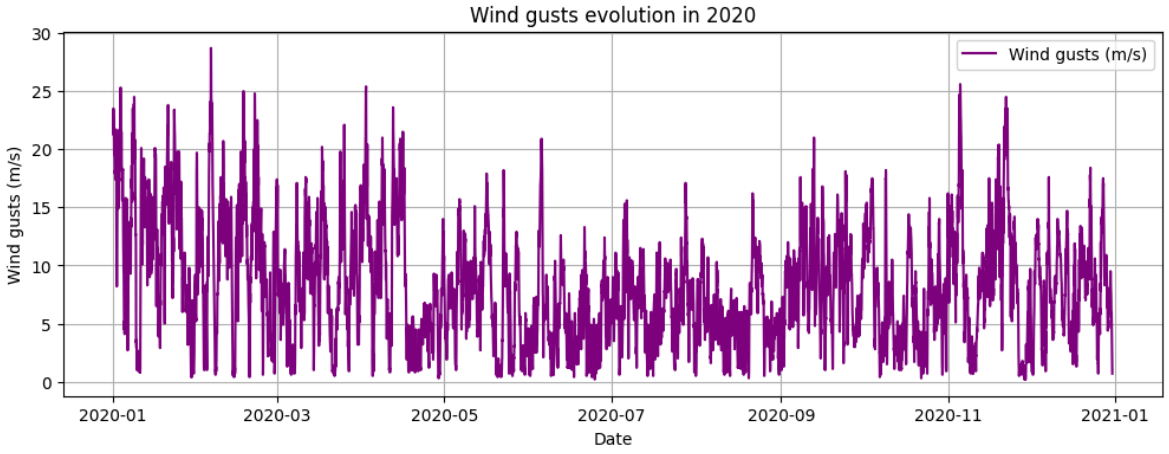
In [12]:
```
# Precipitation plot
plt.figure(figsize=(12, 4))
plt.plot(df.index, df['precipitation (mm)'], color='blue', label='Precipitation
plt.title("Precipitation evolution in 2020")
plt.xlabel("Date")
plt.ylabel("Precipitation (mm)")
plt.grid(True)
plt.legend()
plt.show()
```



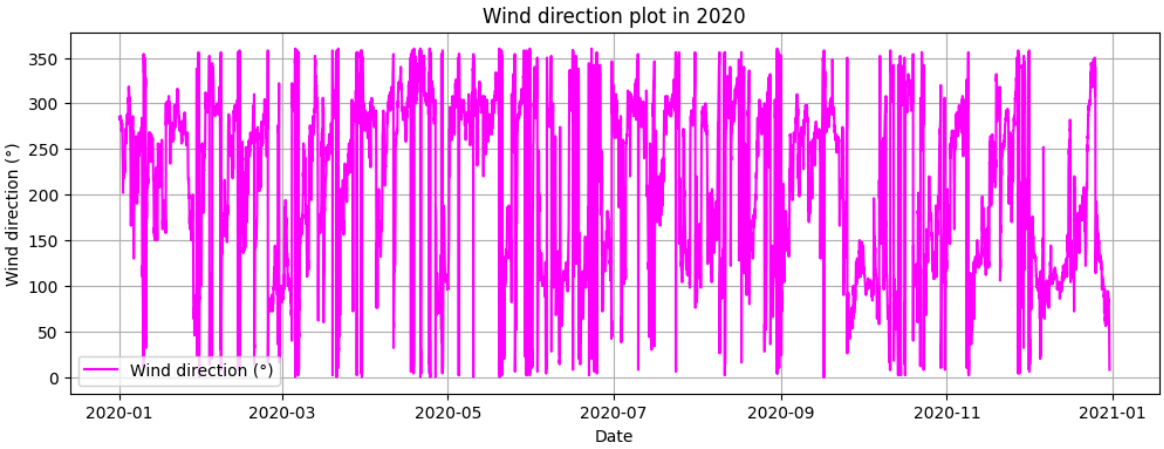
In [13]:
```
# Wind speed plot
plt.figure(figsize=(12, 4))
plt.plot(df.index, df['wind_speed_10m (m/s)'], color='green', label='Wind speed
plt.title("Wind speed evolution in 2020")
plt.xlabel("Date")
plt.ylabel("Wind speed (m/s)")
plt.grid(True)
plt.legend()
plt.show()
```



In [14]:
```
# Wind gusts plot
plt.figure(figsize=(12, 4))
plt.plot(df.index, df['wind_gusts_10m (m/s)'], color='purple', label='Wind gusts
plt.title("Wind gusts evolution in 2020")
plt.xlabel("Date")
plt.ylabel("Wind gusts (m/s)")
plt.grid(True)
plt.legend()
plt.show()
```



In [15]:
```
# Wind direction plot
plt.figure(figsize=(12, 4))
plt.plot(df.index, df['wind_direction_10m (°)'], color='magenta', label='Wind di
plt.title("Wind direction plot in 2020")
plt.xlabel("Date")
plt.ylabel("Wind direction (°)")
plt.grid(True)
plt.legend()
plt.show()
```



Group plot

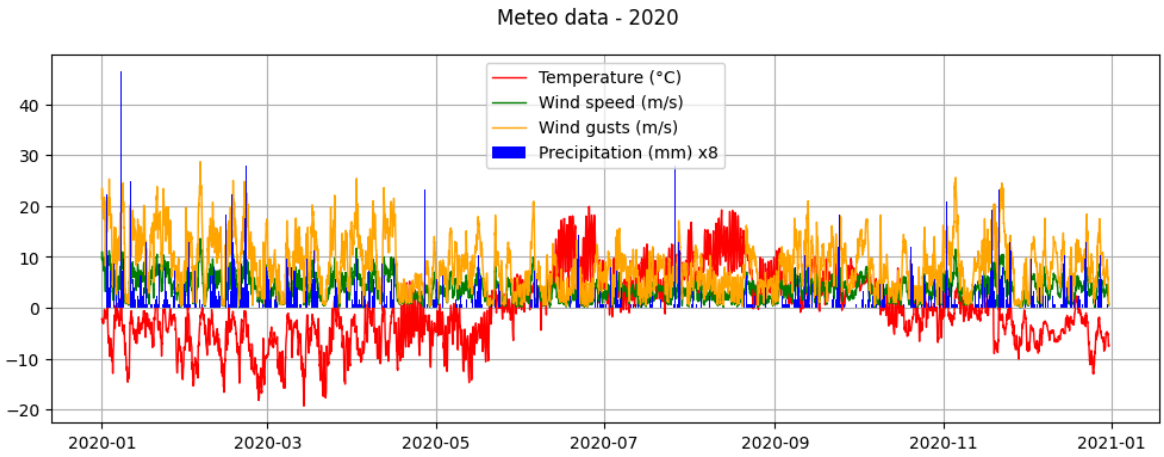
In [16]:
```
plt.figure(figsize=(12, 4))

# Temperatures
plt.plot(df.index, df['temperature_2m (°C)'], color='red', label='Temperature (°

# Wind speed and gusts
plt.plot(df.index, df['wind_speed_10m (m/s)'], color='green', label='Wind speed
plt.plot(df.index, df['wind_gusts_10m (m/s)'], color='orange', label='Wind gusts

# Precipitations (bars brought to the foreground)
plt.bar(df.index, df['precipitation (mm)'] * 8, color='blue', width=0.05, label=

plt.suptitle("Meteo data - 2020")
plt.legend()
plt.grid(True, zorder=1)  # Ensure the grid is in the background
plt.show()
```



Windrose

In [17]:
```
# Colones extraction
wind_dir = df['wind_direction_10m (°)']  # Directions in degres (0-360)
wind_speed = df['wind_speed_10m (m/s)']

# Windrose plot
fig = plt.figure(figsize=(10, 8))
ax = WindroseAxes.from_ax(fig=fig)
ax.bar(wind_dir, wind_speed, normed=True, opening=0.8, edgecolor='white')
```

```
# Personnalisation
ax.set_legend(title="Wind speed (m/s)")
plt.title("Windrose - 2020", y=1.1)
plt.show()
```