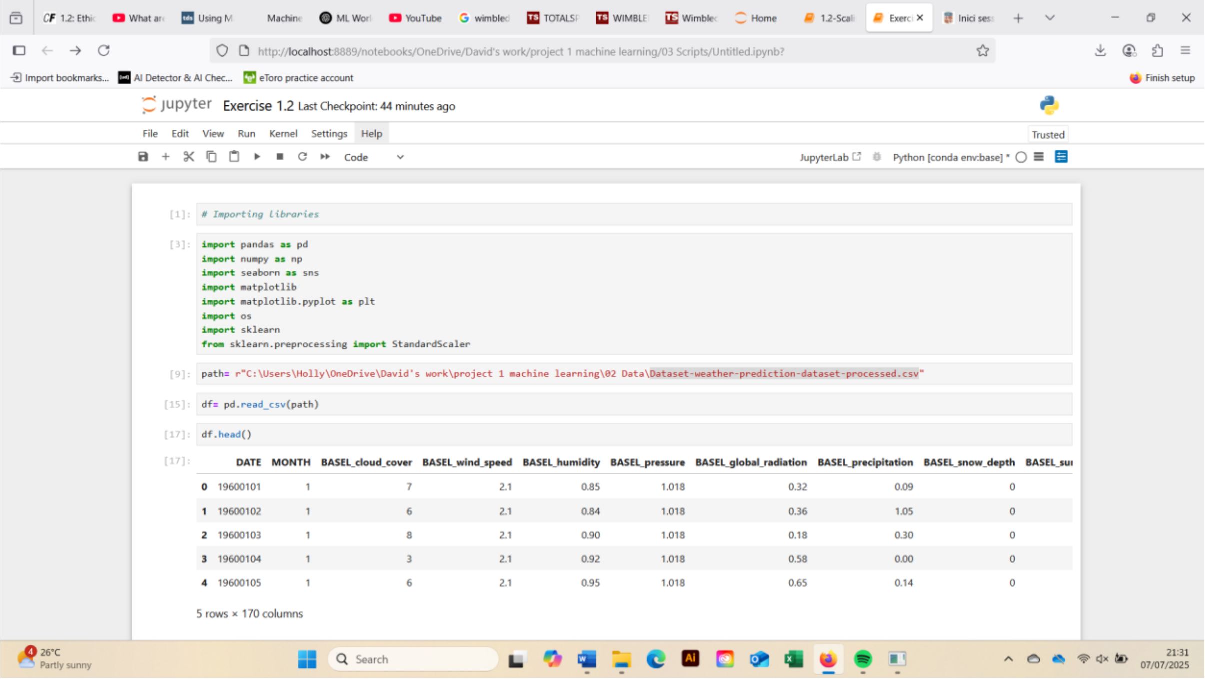
Part 1

When working with climate change data, ClimateWins needs to be careful about several ethical issues, especially when applying machine learning. First, while most weather data isn’t personal, there could still be indirect risks. For example, one major concern is regional and cultural bias. Climate models are often trained on data from wealthier countries with more consistent reporting, like those in Western Europe or the U.S. This bias can lead to inaccurate predictions or blind spots for places with less data which is often where climate change hits hardest. Additionally, human biases in how past climate events were reported or categorized could be “learned” by the model and repeated in predictions. If the data reflects historical inequality in response or attention to climate events, the model might carry that bias forward. Finally, predictions made by machine learning could misidentify risk zones, either by overlooking areas that are in danger or by over-prioritizing others. This bias can cause harm if used in policymaking.

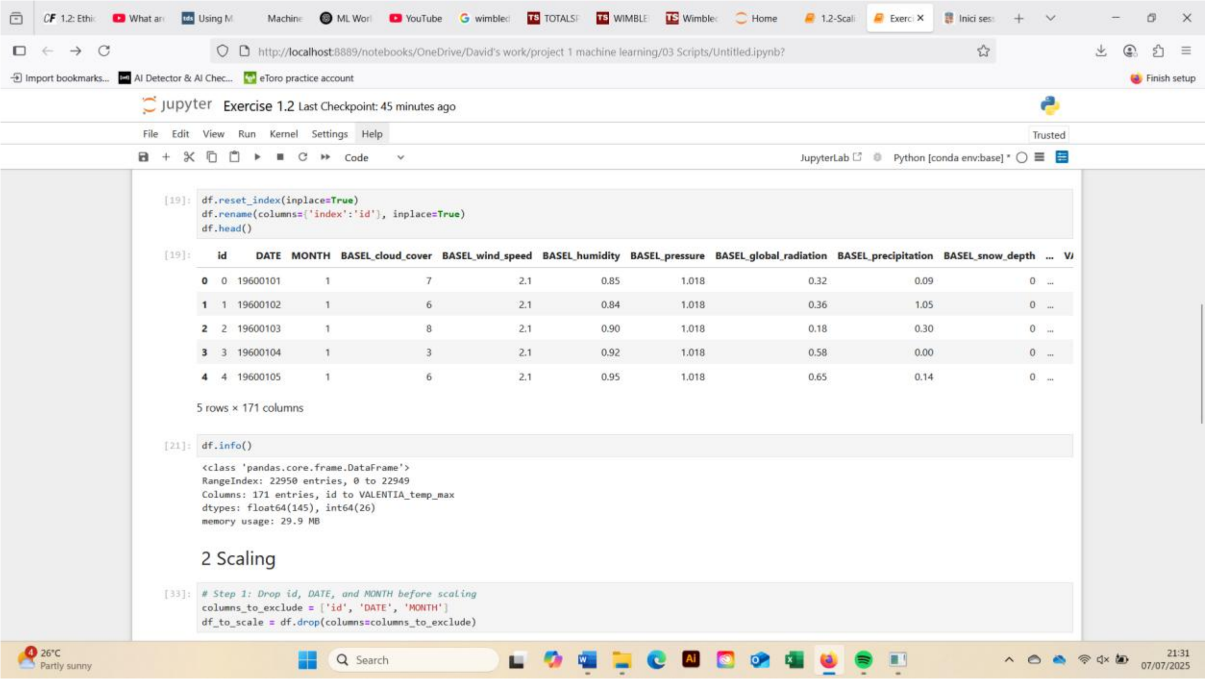
Part 2


http://localhost:8889/notebooks/OneDrive/David's work/project 1 machine learning/03 Scripts/Untitled.ipynb?
Jupyter Exercise 1.2 Last Checkpoint: 44 minutes ago
File
Edit
View
Run
Kernel
Settings
Help
Trusted
Code
JupyterLab
Python [conda env:base] *
Search
26°C
Partly sunny
21:31
07/07/2025

```
[1]: # Importing Libraries

[3]: import pandas as pd
     import numpy as np
     import seaborn as sns
     import matplotlib
     import matplotlib.pyplot as plt
     import os
     import sklearn
     from sklearn.preprocessing import StandardScaler

[9]: path= r"C:\Users\Holly\OneDrive\David's work\project 1 machine learning\02 Data\Dataset-weather-prediction-dataset-processed.csv"

[15]: df= pd.read_csv(path)

[17]: df.head()

[17]:
```

	DATE	MONTH	BASEL_cloud_cover	BASEL_wind_speed	BASEL_humidity	BASEL_pressure	BASEL_global_radiation	BASEL_precipitation	BASEL_snow_depth	BASEL_su
0	19600101	1	7	2.1	0.85	1.018	0.32	0.09	0	
1	19600102	1	6	2.1	0.84	1.018	0.36	1.05	0	
2	19600103	1	8	2.1	0.90	1.018	0.18	0.30	0	
3	19600104	1	3	2.1	0.92	1.018	0.58	0.00	0	
4	19600105	1	6	2.1	0.95	1.018	0.65	0.14	0	

5 rows × 170 columns


http://localhost:8889/notebooks/OneDrive/David's work/project 1 machine learning/03 Scripts/Untitled.ipynb?
jupyter Exercise 1.2 Last Checkpoint: 45 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *
26°C
Partly sunny
Search
21:31
07/07/2025

```
[19]: df.reset_index(inplace=True)
      df.rename(columns={'index':'id'}, inplace=True)
      df.head()
```

[19]:

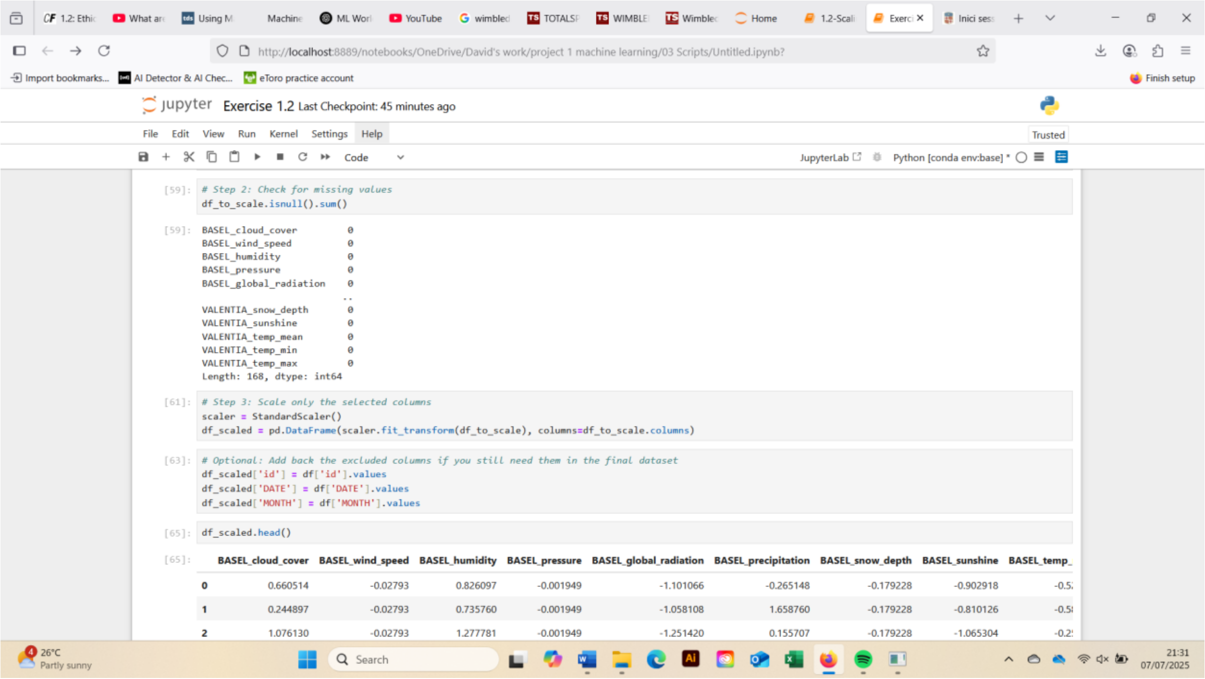
	id	DATE	MONTH	BASEL_cloud_cover	BASEL_wind_speed	BASEL_humidity	BASEL_pressure	BASEL_global_radiation	BASEL_precipitation	BASEL_snow_depth	...	VA
0	0	19600101	1	7	2.1	0.85	1.018	0.32	0.09	0	...	
1	1	19600102	1	6	2.1	0.84	1.018	0.36	1.05	0	...	
2	2	19600103	1	8	2.1	0.90	1.018	0.18	0.30	0	...	
3	3	19600104	1	3	2.1	0.92	1.018	0.58	0.00	0	...	
4	4	19600105	1	6	2.1	0.95	1.018	0.65	0.14	0	...	

5 rows × 171 columns

```
[21]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22950 entries, 0 to 22949
Columns: 171 entries, id to VALENTIA_temp_max
dtypes: float64(145), int64(26)
memory usage: 29.9 MB
```

2 Scaling

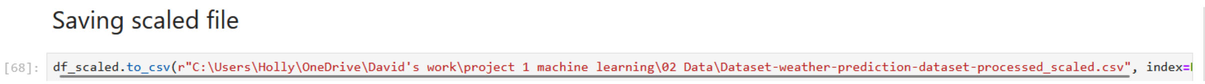
```
[33]: # Step 1: Drop id, DATE, and MONTH before scaling
      columns_to_exclude = ['id', 'DATE', 'MONTH']
      df_to_scale = df.drop(columns=columns_to_exclude)
```


http://localhost:8889/notebooks/OneDrive/David's work/project 1 machine learning/03 Scripts/Untitled.ipynb?
jupyter Exercise 1.2 Last Checkpoint: 45 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *
26°C
Partly sunny
Search
21:31
07/07/2025

```
[59]: # Step 2: Check for missing values
      df_to_scale.isnull().sum()

[59]: BASEL_cloud_cover         0
      BASEL_wind_speed          0
      BASEL_humidity            0
      BASEL_pressure            0
      BASEL_global_radiation    0
                               ..
      VALENTIA_snow_depth       0
      VALENTIA_sunshine         0
      VALENTIA_temp_mean        0
      VALENTIA_temp_min         0
      VALENTIA_temp_max         0
      Length: 168, dtype: int64

[61]: # Step 3: Scale only the selected columns
      scaler = StandardScaler()
      df_scaled = pd.DataFrame(scaler.fit_transform(df_to_scale), columns=df_to_scale.columns)

[63]: # Optional: Add back the excluded columns if you still need them in the final dataset
      df_scaled['id'] = df['id'].values
      df_scaled['DATE'] = df['DATE'].values
      df_scaled['MONTH'] = df['MONTH'].values

[65]: df_scaled.head()
```

[65]:

	BASEL_cloud_cover	BASEL_wind_speed	BASEL_humidity	BASEL_pressure	BASEL_global_radiation	BASEL_precipitation	BASEL_snow_depth	BASEL_sunshine	BASEL_temp_
0	0.660514	-0.02793	0.826097	-0.001949	-1.101066	-0.265148	-0.179228	-0.902918	-0.5
1	0.244897	-0.02793	0.735760	-0.001949	-1.058108	1.658760	-0.179228	-0.810126	-0.5
2	1.076130	-0.02793	1.277781	-0.001949	-1.251420	0.155707	-0.179228	-1.065304	-0.2



Saving scaled file

```
[68]: df_scaled.to_csv(r"C:\Users\Holly\OneDrive\David's work\project 1 machine learning\02 Data\Dataset-weather-prediction-dataset-processed_scaled.csv", index=
```