import liberaries

In [ ]:

```
from neuralprophet import NeuralProphet
```

In [98]:

```
!pip install Pyppeteer
!pyppeteer-install
```

```
Collecting Pyppeteer
  Downloading pyppeteer-1.0.2-py3-none-any.whl (83 kB)
     -------------------------------------- 83.4/83.4 kB 585.1 kB/s eta 0:00:00
Requirement already satisfied: appdirs<2.0.0,>=1.4.3 in c:\users\youne\anaconda3\lib\sit
e-packages (from Pyppeteer) (1.4.4)
Requirement already satisfied: certifi>=2021 in c:\users\youne\anaconda3\lib\site-packag
es (from Pyppeteer) (2023.7.22)
Requirement already satisfied: importlib-metadata>=1.4 in c:\users\youne\anaconda3\lib\s
ite-packages (from Pyppeteer) (6.0.0)
Collecting pyee<9.0.0,>=8.1.0 (from Pyppeteer)
  Downloading pyee-8.2.2-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: tqdm<5.0.0,>=4.42.1 in c:\users\youne\anaconda3\lib\site-
packages (from Pyppeteer) (4.65.0)
Requirement already satisfied: urllib3<2.0.0,>=1.25.8 in c:\users\youne\anaconda3\lib\si
te-packages (from Pyppeteer) (1.26.16)
Collecting websockets<11.0,>=10.0 (from Pyppeteer)
  Downloading websockets-10.4-cp311-cp311-win_amd64.whl (101 kB)
     -------------------------------------- 101.4/101.4 kB 1.2 MB/s eta 0:00:00
Requirement already satisfied: zipp>=0.5 in c:\users\youne\anaconda3\lib\site-packages
(from importlib-metadata>=1.4->Pyppeteer) (3.11.0)
Requirement already satisfied: colorama in c:\users\youne\anaconda3\lib\site-packages (f
rom tqdm<5.0.0,>=4.42.1->Pyppeteer) (0.4.6)
Installing collected packages: pyee, websockets, Pyppeteer
Successfully installed Pyppeteer-1.0.2 pyee-8.2.2 websockets-10.4
```

```
[INFO] Starting Chromium download.

  0%|          | 0.00/137M [00:00<?, ?b/s]
  0%|          | 51.2k/137M [00:00<05:51, 389kb/s]
  0%|          | 92.2k/137M [00:00<05:46, 395kb/s]
  0%|          | 215k/137M [00:00<03:11, 714kb/s]
  0%|          | 399k/137M [00:00<02:04, 1.09Mb/s]
  1%|          | 737k/137M [00:00<01:15, 1.80Mb/s]
  1%|          | 1.02M/137M [00:00<01:05, 2.07Mb/s]
  1%|          | 1.27M/137M [00:00<01:02, 2.17Mb/s]
  1%|1         | 1.56M/137M [00:00<00:58, 2.31Mb/s]
  1%|1         | 1.81M/137M [00:00<00:56, 2.38Mb/s]
  2%|1         | 2.14M/137M [00:01<00:51, 2.60Mb/s]
  2%|1         | 2.47M/137M [00:01<00:49, 2.73Mb/s]
  2%|2         | 2.80M/137M [00:01<00:47, 2.83Mb/s]
  2%|2         | 3.08M/137M [00:01<00:48, 2.77Mb/s]
  3%|2         | 3.43M/137M [00:01<00:46, 2.88Mb/s]
  3%|2         | 3.73M/137M [00:01<00:47, 2.82Mb/s]
  3%|2         | 4.01M/137M [00:01<00:50, 2.64Mb/s]
  3%|3         | 4.28M/137M [00:01<00:51, 2.59Mb/s]
  3%|3         | 4.55M/137M [00:01<00:51, 2.55Mb/s]
  4%|3         | 4.85M/137M [00:02<00:49, 2.66Mb/s]
  4%|3         | 5.18M/137M [00:02<00:50, 2.63Mb/s]
  4%|4         | 5.51M/137M [00:02<00:47, 2.77Mb/s]
  4%|4         | 5.80M/137M [00:02<00:47, 2.78Mb/s]
  4%|4         | 6.11M/137M [00:02<00:46, 2.82Mb/s]
  5%|4         | 6.44M/137M [00:02<00:45, 2.86Mb/s]
  5%|4         | 6.77M/137M [00:02<00:45, 2.89Mb/s]
  5%|5         | 7.07M/137M [00:02<00:46, 2.82Mb/s]
  5%|5         | 7.35M/137M [00:03<00:53, 2.41Mb/s]
```

```
 6%|5         | 7.61M/137M [00:03<01:03, 2.03Mb/s]
 6%|5         | 7.92M/137M [00:03<00:58, 2.21Mb/s]
 6%|5         | 8.20M/137M [00:03<00:54, 2.37Mb/s]
 6%|6         | 8.48M/137M [00:03<00:52, 2.46Mb/s]
 6%|6         | 8.82M/137M [00:03<00:48, 2.63Mb/s]
 7%|6         | 9.14M/137M [00:03<00:46, 2.73Mb/s]
 7%|6         | 9.49M/137M [00:03<00:44, 2.84Mb/s]
 7%|7         | 9.84M/137M [00:03<00:43, 2.93Mb/s]
 7%|7         | 10.1M/137M [00:04<00:43, 2.94Mb/s]
 8%|7         | 10.5M/137M [00:04<00:42, 3.01Mb/s]
 8%|7         | 10.8M/137M [00:04<00:42, 2.99Mb/s]
 8%|8         | 11.1M/137M [00:04<00:42, 2.97Mb/s]
 8%|8         | 11.4M/137M [00:04<00:41, 3.01Mb/s]
 9%|8         | 11.7M/137M [00:04<00:41, 3.04Mb/s]
 9%|8         | 12.1M/137M [00:04<00:41, 2.99Mb/s]
 9%|9         | 12.4M/137M [00:04<00:40, 3.07Mb/s]
 9%|9         | 12.7M/137M [00:04<00:41, 3.03Mb/s]
10%|9         | 13.0M/137M [00:05<00:41, 3.02Mb/s]
10%|9         | 13.3M/137M [00:05<00:41, 3.01Mb/s]
10%|9         | 13.6M/137M [00:05<00:49, 2.49Mb/s]
10%|#         | 13.9M/137M [00:05<00:54, 2.24Mb/s]
10%|#         | 14.3M/137M [00:05<00:48, 2.54Mb/s]
11%|#         | 14.7M/137M [00:05<00:43, 2.80Mb/s]
11%|#         | 15.0M/137M [00:05<00:42, 2.89Mb/s]
11%|#1        | 15.4M/137M [00:05<00:38, 3.16Mb/s]
12%|#1        | 15.8M/137M [00:05<00:37, 3.26Mb/s]
12%|#1        | 16.1M/137M [00:06<00:37, 3.20Mb/s]
12%|#2        | 16.4M/137M [00:06<00:38, 3.13Mb/s]
12%|#2        | 16.8M/137M [00:06<00:36, 3.32Mb/s]
13%|#2        | 17.2M/137M [00:06<00:37, 3.23Mb/s]
13%|#2        | 17.6M/137M [00:06<00:35, 3.38Mb/s]
13%|#3        | 18.0M/137M [00:06<00:34, 3.47Mb/s]
13%|#3        | 18.4M/137M [00:06<00:33, 3.54Mb/s]
14%|#3        | 18.7M/137M [00:06<00:33, 3.53Mb/s]
14%|#3        | 19.1M/137M [00:06<00:32, 3.62Mb/s]
14%|#4        | 19.5M/137M [00:07<00:32, 3.58Mb/s]
15%|#4        | 19.9M/137M [00:07<00:34, 3.37Mb/s]
15%|#4        | 20.3M/137M [00:07<00:33, 3.47Mb/s]
15%|#5        | 20.7M/137M [00:07<00:32, 3.54Mb/s]
15%|#5        | 21.1M/137M [00:07<00:32, 3.59Mb/s]
16%|#5        | 21.4M/137M [00:07<00:31, 3.61Mb/s]
16%|#5        | 21.9M/137M [00:07<00:30, 3.75Mb/s]
16%|#6        | 22.3M/137M [00:07<00:30, 3.80Mb/s]
17%|#6        | 22.8M/137M [00:07<00:28, 4.02Mb/s]
17%|#6        | 23.2M/137M [00:08<00:28, 3.93Mb/s]
17%|#7        | 23.7M/137M [00:08<00:28, 4.01Mb/s]
18%|#7        | 24.1M/137M [00:08<00:28, 4.03Mb/s]
18%|#7        | 24.5M/137M [00:08<00:27, 4.04Mb/s]
18%|#8        | 25.0M/137M [00:08<00:27, 4.08Mb/s]
19%|#8        | 25.4M/137M [00:08<00:26, 4.14Mb/s]
19%|#8        | 25.9M/137M [00:08<00:26, 4.12Mb/s]
19%|#9        | 26.3M/137M [00:08<00:26, 4.12Mb/s]
20%|#9        | 26.7M/137M [00:08<00:28, 3.93Mb/s]
20%|#9        | 27.1M/137M [00:09<00:29, 3.69Mb/s]
20%|##        | 27.5M/137M [00:09<00:32, 3.32Mb/s]
20%|##        | 27.9M/137M [00:09<00:33, 3.28Mb/s]
21%|##        | 28.2M/137M [00:09<00:33, 3.29Mb/s]
21%|##        | 28.6M/137M [00:09<00:33, 3.28Mb/s]
21%|##1       | 28.9M/137M [00:09<00:32, 3.29Mb/s]
21%|##1       | 29.4M/137M [00:09<00:30, 3.56Mb/s]
22%|##1       | 29.8M/137M [00:09<00:29, 3.66Mb/s]
22%|##2       | 30.2M/137M [00:09<00:28, 3.78Mb/s]
22%|##2       | 30.6M/137M [00:10<00:28, 3.80Mb/s]
23%|##2       | 31.1M/137M [00:10<00:27, 3.83Mb/s]
23%|##3       | 31.5M/137M [00:10<00:27, 3.89Mb/s]
23%|##3       | 32.0M/137M [00:10<00:26, 4.03Mb/s]
```

```
 24%|##3       | 32.4M/137M [00:10<00:26, 4.00Mb/s]
 24%|##3       | 32.8M/137M [00:10<00:34, 3.02Mb/s]
 24%|##4       | 33.2M/137M [00:10<00:32, 3.19Mb/s]
 25%|##4       | 33.6M/137M [00:10<00:31, 3.33Mb/s]
 25%|##4       | 34.0M/137M [00:11<00:29, 3.52Mb/s]
 25%|##5       | 34.5M/137M [00:11<00:28, 3.61Mb/s]
 25%|##5       | 34.9M/137M [00:11<00:27, 3.69Mb/s]
 26%|##5       | 35.3M/137M [00:11<00:27, 3.71Mb/s]
 26%|##6       | 35.7M/137M [00:11<00:27, 3.63Mb/s]
 26%|##6       | 36.1M/137M [00:11<00:26, 3.75Mb/s]
 27%|##6       | 36.5M/137M [00:11<00:26, 3.85Mb/s]
 27%|##6       | 37.0M/137M [00:11<00:25, 3.90Mb/s]
 27%|##7       | 37.4M/137M [00:11<00:25, 3.92Mb/s]
 28%|##7       | 37.8M/137M [00:11<00:24, 4.00Mb/s]
 28%|##7       | 38.2M/137M [00:12<00:24, 3.97Mb/s]
 28%|##8       | 38.6M/137M [00:12<00:25, 3.86Mb/s]
 29%|##8       | 39.0M/137M [00:12<00:26, 3.67Mb/s]
 29%|##8       | 39.4M/137M [00:12<00:26, 3.67Mb/s]
 29%|##9       | 39.8M/137M [00:12<00:28, 3.42Mb/s]
 29%|##9       | 40.2M/137M [00:12<00:27, 3.46Mb/s]
 30%|##9       | 40.6M/137M [00:12<00:26, 3.67Mb/s]
 30%|##9       | 41.1M/137M [00:12<00:25, 3.76Mb/s]
 30%|###       | 41.5M/137M [00:12<00:24, 3.89Mb/s]
 31%|###       | 42.0M/137M [00:13<00:23, 3.98Mb/s]
 31%|###       | 42.4M/137M [00:13<00:23, 4.00Mb/s]
 31%|###1      | 42.8M/137M [00:13<00:23, 3.92Mb/s]
 32%|###1      | 43.2M/137M [00:13<00:23, 3.97Mb/s]
 32%|###1      | 43.6M/137M [00:13<00:23, 3.89Mb/s]
 32%|###2      | 44.0M/137M [00:13<00:23, 3.89Mb/s]
 32%|###2      | 44.4M/137M [00:13<00:26, 3.49Mb/s]
 33%|###2      | 44.8M/137M [00:13<00:26, 3.53Mb/s]
 33%|###3      | 45.3M/137M [00:13<00:24, 3.71Mb/s]
 33%|###3      | 45.7M/137M [00:14<00:23, 3.83Mb/s]
 34%|###3      | 46.2M/137M [00:14<00:22, 3.96Mb/s]
 34%|###4      | 46.6M/137M [00:14<00:22, 3.98Mb/s]
 34%|###4      | 47.0M/137M [00:14<00:22, 3.99Mb/s]
 35%|###4      | 47.5M/137M [00:14<00:22, 4.00Mb/s]
 35%|###4      | 47.9M/137M [00:14<00:22, 3.97Mb/s]
 35%|###5      | 48.3M/137M [00:14<00:22, 3.93Mb/s]
 36%|###5      | 48.8M/137M [00:14<00:22, 3.98Mb/s]
 36%|###5      | 49.2M/137M [00:14<00:22, 3.94Mb/s]
 36%|###6      | 49.6M/137M [00:15<00:21, 4.02Mb/s]
 37%|###6      | 50.1M/137M [00:15<00:21, 4.02Mb/s]
 37%|###6      | 50.5M/137M [00:15<00:21, 3.94Mb/s]
 37%|###7      | 50.9M/137M [00:15<00:23, 3.69Mb/s]
 37%|###7      | 51.3M/137M [00:15<00:24, 3.55Mb/s]
 38%|###7      | 51.6M/137M [00:15<00:23, 3.57Mb/s]
 38%|###7      | 52.0M/137M [00:15<00:24, 3.51Mb/s]
 38%|###8      | 52.4M/137M [00:15<00:23, 3.64Mb/s]
 39%|###8      | 52.8M/137M [00:15<00:23, 3.62Mb/s]
 39%|###8      | 53.1M/137M [00:16<00:23, 3.52Mb/s]
 39%|###9      | 53.5M/137M [00:16<00:22, 3.70Mb/s]
 39%|###9      | 54.0M/137M [00:16<00:21, 3.90Mb/s]
 40%|###9      | 54.4M/137M [00:16<00:20, 3.96Mb/s]
 40%|####      | 54.8M/137M [00:16<00:21, 3.86Mb/s]
 40%|####      | 55.2M/137M [00:16<00:21, 3.87Mb/s]
 41%|####      | 55.6M/137M [00:16<00:20, 3.94Mb/s]
 41%|####      | 56.1M/137M [00:16<00:20, 4.00Mb/s]
 41%|####1     | 56.5M/137M [00:16<00:19, 4.07Mb/s]
 42%|####1     | 57.0M/137M [00:17<00:19, 4.06Mb/s]
 42%|####1     | 57.4M/137M [00:17<00:19, 4.10Mb/s]
 42%|####2     | 57.9M/137M [00:17<00:18, 4.19Mb/s]
 43%|####2     | 58.4M/137M [00:17<00:18, 4.17Mb/s]
 43%|####2     | 58.8M/137M [00:17<00:21, 3.66Mb/s]
 43%|####3     | 59.3M/137M [00:17<00:19, 3.90Mb/s]
 44%|####3     | 59.7M/137M [00:17<00:24, 3.10Mb/s]
```

```
44%|####3     | 60.1M/137M [00:17<00:23, 3.28Mb/s]
44%|####4     | 60.5M/137M [00:18<00:22, 3.33Mb/s]
44%|####4     | 60.9M/137M [00:18<00:22, 3.31Mb/s]
45%|####4     | 61.3M/137M [00:18<00:22, 3.41Mb/s]
45%|####5     | 61.6M/137M [00:18<00:22, 3.37Mb/s]
45%|####5     | 62.0M/137M [00:18<00:22, 3.34Mb/s]
46%|####5     | 62.4M/137M [00:18<00:21, 3.51Mb/s]
46%|####5     | 62.8M/137M [00:18<00:23, 3.17Mb/s]
46%|####6     | 63.2M/137M [00:18<00:21, 3.36Mb/s]
46%|####6     | 63.6M/137M [00:18<00:22, 3.33Mb/s]
47%|####6     | 64.1M/137M [00:19<00:20, 3.57Mb/s]
47%|####7     | 64.5M/137M [00:19<00:20, 3.62Mb/s]
47%|####7     | 64.8M/137M [00:19<00:19, 3.64Mb/s]
48%|####7     | 65.2M/137M [00:19<00:19, 3.74Mb/s]
48%|####7     | 65.7M/137M [00:19<00:18, 3.84Mb/s]
48%|####8     | 66.1M/137M [00:19<00:18, 3.89Mb/s]
49%|####8     | 66.5M/137M [00:19<00:17, 3.91Mb/s]
49%|####8     | 67.0M/137M [00:19<00:17, 3.95Mb/s]
49%|####9     | 67.4M/137M [00:19<00:17, 4.00Mb/s]
50%|####9     | 67.8M/137M [00:19<00:17, 4.00Mb/s]
50%|####9     | 68.3M/137M [00:20<00:16, 4.11Mb/s]
50%|#####     | 68.7M/137M [00:20<00:16, 4.13Mb/s]
51%|#####     | 69.2M/137M [00:20<00:16, 4.06Mb/s]
51%|#####     | 69.6M/137M [00:20<00:16, 4.01Mb/s]
51%|#####1    | 70.0M/137M [00:20<00:16, 3.98Mb/s]
51%|#####1    | 70.5M/137M [00:20<00:16, 4.03Mb/s]
52%|#####1    | 70.9M/137M [00:20<00:17, 3.83Mb/s]
52%|#####2    | 71.3M/137M [00:20<00:17, 3.84Mb/s]
52%|#####2    | 71.7M/137M [00:20<00:16, 3.86Mb/s]
53%|#####2    | 72.1M/137M [00:21<00:16, 3.93Mb/s]
53%|#####2    | 72.6M/137M [00:21<00:16, 3.87Mb/s]
53%|#####3    | 73.0M/137M [00:21<00:16, 3.97Mb/s]
54%|#####3    | 73.4M/137M [00:21<00:16, 3.95Mb/s]
54%|#####3    | 73.9M/137M [00:21<00:15, 4.02Mb/s]
54%|#####4    | 74.3M/137M [00:21<00:15, 4.01Mb/s]
55%|#####4    | 74.8M/137M [00:21<00:15, 4.04Mb/s]
55%|#####4    | 75.2M/137M [00:21<00:17, 3.50Mb/s]
55%|#####5    | 75.6M/137M [00:22<00:17, 3.45Mb/s]
55%|#####5    | 76.0M/137M [00:22<00:17, 3.47Mb/s]
56%|#####5    | 76.4M/137M [00:22<00:17, 3.52Mb/s]
56%|#####6    | 76.7M/137M [00:22<00:18, 3.32Mb/s]
56%|#####6    | 77.1M/137M [00:22<00:18, 3.27Mb/s]
57%|#####6    | 77.4M/137M [00:22<00:17, 3.36Mb/s]
57%|#####6    | 77.9M/137M [00:22<00:16, 3.58Mb/s]
57%|#####7    | 78.3M/137M [00:22<00:16, 3.64Mb/s]
57%|#####7    | 78.7M/137M [00:22<00:18, 3.16Mb/s]
58%|#####7    | 79.0M/137M [00:23<00:19, 3.02Mb/s]
58%|#####7    | 79.4M/137M [00:23<00:17, 3.24Mb/s]
58%|#####8    | 79.8M/137M [00:23<00:16, 3.42Mb/s]
59%|#####8    | 80.2M/137M [00:23<00:15, 3.68Mb/s]
59%|#####8    | 80.7M/137M [00:23<00:15, 3.73Mb/s]
59%|#####9    | 81.1M/137M [00:23<00:14, 3.88Mb/s]
60%|#####9    | 81.5M/137M [00:23<00:14, 3.87Mb/s]
60%|#####9    | 82.0M/137M [00:23<00:14, 3.89Mb/s]
60%|######    | 82.4M/137M [00:23<00:13, 3.92Mb/s]
60%|######    | 82.8M/137M [00:24<00:14, 3.67Mb/s]
61%|######    | 83.2M/137M [00:24<00:14, 3.63Mb/s]
61%|######1   | 83.6M/137M [00:24<00:14, 3.67Mb/s]
61%|######1   | 84.1M/137M [00:24<00:13, 3.89Mb/s]
62%|######1   | 84.5M/137M [00:24<00:13, 3.84Mb/s]
62%|######2   | 84.9M/137M [00:24<00:13, 3.76Mb/s]
62%|######2   | 85.4M/137M [00:24<00:12, 3.97Mb/s]
63%|######2   | 85.8M/137M [00:24<00:12, 4.04Mb/s]
63%|######2   | 86.2M/137M [00:24<00:12, 3.98Mb/s]
63%|######3   | 86.6M/137M [00:25<00:13, 3.75Mb/s]
64%|######3   | 87.0M/137M [00:25<00:12, 3.86Mb/s]
```

```
64%|######3   | 87.4M/137M [00:25<00:13, 3.62Mb/s]
64%|######4   | 87.8M/137M [00:25<00:13, 3.59Mb/s]
64%|######4   | 88.2M/137M [00:25<00:13, 3.59Mb/s]
65%|######4   | 88.5M/137M [00:25<00:13, 3.62Mb/s]
65%|######4   | 88.9M/137M [00:25<00:13, 3.63Mb/s]
65%|######5   | 89.3M/137M [00:25<00:12, 3.68Mb/s]
66%|######5   | 89.8M/137M [00:25<00:12, 3.81Mb/s]
66%|######5   | 90.2M/137M [00:26<00:12, 3.87Mb/s]
66%|######6   | 90.7M/137M [00:26<00:11, 4.02Mb/s]
67%|######6   | 91.1M/137M [00:26<00:11, 4.01Mb/s]
67%|######6   | 91.6M/137M [00:26<00:11, 4.06Mb/s]
67%|######7   | 92.1M/137M [00:26<00:10, 4.13Mb/s]
68%|######7   | 92.5M/137M [00:26<00:10, 4.05Mb/s]
68%|######7   | 92.9M/137M [00:26<00:10, 4.03Mb/s]
68%|######8   | 93.3M/137M [00:26<00:10, 3.97Mb/s]
69%|######8   | 93.8M/137M [00:26<00:10, 4.04Mb/s]
69%|######8   | 94.2M/137M [00:26<00:10, 4.00Mb/s]
69%|######9   | 94.6M/137M [00:27<00:10, 3.94Mb/s]
69%|######9   | 95.0M/137M [00:27<00:11, 3.80Mb/s]
70%|######9   | 95.5M/137M [00:27<00:10, 3.86Mb/s]
70%|#######   | 95.9M/137M [00:27<00:10, 3.96Mb/s]
70%|#######   | 96.4M/137M [00:27<00:10, 3.97Mb/s]
71%|#######   | 96.8M/137M [00:27<00:09, 4.03Mb/s]
71%|#######1  | 97.2M/137M [00:27<00:09, 4.04Mb/s]
71%|#######1  | 97.7M/137M [00:27<00:09, 4.07Mb/s]
72%|#######1  | 98.1M/137M [00:27<00:09, 4.02Mb/s]
72%|#######1  | 98.5M/137M [00:28<00:09, 3.88Mb/s]
72%|#######2  | 98.9M/137M [00:28<00:12, 3.07Mb/s]
73%|#######2  | 99.3M/137M [00:28<00:12, 3.02Mb/s]
73%|#######2  | 99.6M/137M [00:28<00:12, 3.09Mb/s]
73%|#######3  | 100M/137M [00:28<00:11, 3.15Mb/s]
73%|#######3  | 100M/137M [00:28<00:11, 3.21Mb/s]
74%|#######3  | 101M/137M [00:28<00:11, 3.25Mb/s]
74%|#######3  | 101M/137M [00:28<00:11, 3.20Mb/s]
74%|#######4  | 101M/137M [00:29<00:10, 3.44Mb/s]
74%|#######4  | 102M/137M [00:29<00:09, 3.58Mb/s]
75%|#######4  | 102M/137M [00:29<00:09, 3.67Mb/s]
75%|#######4  | 103M/137M [00:29<00:09, 3.65Mb/s]
75%|#######5  | 103M/137M [00:29<00:10, 3.31Mb/s]
76%|#######5  | 104M/137M [00:29<00:08, 3.80Mb/s]
76%|#######5  | 104M/137M [00:29<00:08, 3.89Mb/s]
76%|#######6  | 104M/137M [00:29<00:08, 3.92Mb/s]
77%|#######6  | 105M/137M [00:29<00:07, 4.04Mb/s]
77%|#######6  | 105M/137M [00:30<00:07, 4.02Mb/s]
77%|#######7  | 106M/137M [00:30<00:07, 4.06Mb/s]
78%|#######7  | 106M/137M [00:30<00:07, 4.03Mb/s]
78%|#######7  | 107M/137M [00:30<00:07, 3.95Mb/s]
78%|#######8  | 107M/137M [00:30<00:07, 3.83Mb/s]
78%|#######8  | 107M/137M [00:30<00:07, 3.85Mb/s]
79%|#######8  | 108M/137M [00:30<00:07, 3.86Mb/s]
79%|#######9  | 108M/137M [00:30<00:07, 3.98Mb/s]
79%|#######9  | 109M/137M [00:30<00:06, 4.02Mb/s]
80%|#######9  | 109M/137M [00:31<00:06, 4.04Mb/s]
80%|########  | 110M/137M [00:31<00:06, 4.12Mb/s]
80%|########  | 110M/137M [00:31<00:06, 4.15Mb/s]
81%|########  | 111M/137M [00:31<00:06, 4.09Mb/s]
81%|########1 | 111M/137M [00:31<00:06, 3.98Mb/s]
81%|########1 | 111M/137M [00:31<00:06, 3.87Mb/s]
82%|########1 | 112M/137M [00:31<00:06, 3.93Mb/s]
82%|########2 | 112M/137M [00:31<00:06, 3.84Mb/s]
82%|########2 | 113M/137M [00:31<00:06, 3.75Mb/s]
83%|########2 | 113M/137M [00:32<00:06, 3.69Mb/s]
83%|########2 | 113M/137M [00:32<00:06, 3.60Mb/s]
83%|########3 | 114M/137M [00:32<00:06, 3.81Mb/s]
84%|########3 | 114M/137M [00:32<00:05, 3.84Mb/s]
84%|########3 | 115M/137M [00:32<00:05, 3.85Mb/s]
```

```
 84%|########4 | 115M/137M [00:32<00:05, 3.93Mb/s]
 84%|########4 | 116M/137M [00:32<00:05, 4.00Mb/s]
 85%|########4 | 116M/137M [00:32<00:05, 3.98Mb/s]
 85%|########5 | 117M/137M [00:32<00:04, 4.07Mb/s]
 85%|########5 | 117M/137M [00:33<00:05, 3.80Mb/s]
 86%|########5 | 117M/137M [00:33<00:05, 3.79Mb/s]
 86%|########6 | 118M/137M [00:33<00:05, 3.72Mb/s]
 86%|########6 | 118M/137M [00:33<00:05, 3.30Mb/s]
 87%|########6 | 119M/137M [00:33<00:06, 2.89Mb/s]
 87%|########6 | 119M/137M [00:33<00:05, 3.14Mb/s]
 87%|########7 | 119M/137M [00:33<00:05, 3.41Mb/s]
 87%|########7 | 120M/137M [00:33<00:04, 3.57Mb/s]
 88%|########7 | 120M/137M [00:33<00:04, 3.72Mb/s]
 88%|########8 | 121M/137M [00:34<00:04, 3.84Mb/s]
 88%|########8 | 121M/137M [00:34<00:04, 3.87Mb/s]
 89%|########8 | 122M/137M [00:34<00:03, 3.89Mb/s]
 89%|########9 | 122M/137M [00:34<00:03, 3.99Mb/s]
 89%|########9 | 122M/137M [00:34<00:03, 3.91Mb/s]
 90%|########9 | 123M/137M [00:34<00:03, 3.79Mb/s]
 90%|########9 | 123M/137M [00:34<00:03, 3.69Mb/s]
 90%|######### | 124M/137M [00:34<00:03, 3.74Mb/s]
 91%|######### | 124M/137M [00:34<00:03, 3.64Mb/s]
 91%|######### | 124M/137M [00:35<00:03, 3.33Mb/s]
 91%|#########1| 125M/137M [00:35<00:04, 2.90Mb/s]
 91%|#########1| 125M/137M [00:35<00:04, 2.76Mb/s]
 92%|#########1| 125M/137M [00:35<00:04, 2.82Mb/s]
 92%|#########1| 126M/137M [00:35<00:03, 2.86Mb/s]
 92%|#########1| 126M/137M [00:35<00:03, 2.76Mb/s]
 92%|#########2| 126M/137M [00:35<00:03, 2.83Mb/s]
 92%|#########2| 127M/137M [00:35<00:03, 2.87Mb/s]
 93%|#########2| 127M/137M [00:36<00:03, 2.85Mb/s]
 93%|#########2| 127M/137M [00:36<00:03, 2.98Mb/s]
 93%|#########3| 128M/137M [00:36<00:03, 2.92Mb/s]
 93%|#########3| 128M/137M [00:36<00:03, 2.90Mb/s]
 94%|#########3| 128M/137M [00:36<00:03, 2.82Mb/s]
 94%|#########3| 128M/137M [00:36<00:02, 2.83Mb/s]
 94%|#########4| 129M/137M [00:36<00:02, 2.87Mb/s]
 94%|#########4| 129M/137M [00:36<00:02, 2.88Mb/s]
 94%|#########4| 129M/137M [00:36<00:02, 2.81Mb/s]
 95%|#########4| 130M/137M [00:37<00:02, 2.87Mb/s]
 95%|#########4| 130M/137M [00:37<00:02, 2.93Mb/s]
 95%|#########5| 130M/137M [00:37<00:02, 2.89Mb/s]
 95%|#########5| 131M/137M [00:37<00:02, 2.96Mb/s]
 96%|#########5| 131M/137M [00:37<00:01, 2.98Mb/s]
 96%|#########5| 131M/137M [00:37<00:01, 3.06Mb/s]
 96%|#########6| 132M/137M [00:37<00:01, 3.04Mb/s]
 96%|#########6| 132M/137M [00:37<00:01, 3.05Mb/s]
 97%|#########6| 132M/137M [00:37<00:01, 2.95Mb/s]
 97%|#########6| 133M/137M [00:37<00:01, 2.95Mb/s]
 97%|#########7| 133M/137M [00:38<00:01, 2.96Mb/s]
 97%|#########7| 133M/137M [00:38<00:01, 2.91Mb/s]
 97%|#########7| 133M/137M [00:38<00:01, 2.62Mb/s]
 98%|#########7| 134M/137M [00:38<00:01, 2.59Mb/s]
 98%|#########7| 134M/137M [00:38<00:01, 2.75Mb/s]
 98%|#########8| 134M/137M [00:38<00:00, 2.60Mb/s]
 98%|#########8| 135M/137M [00:38<00:01, 2.22Mb/s]
 99%|#########8| 135M/137M [00:38<00:00, 2.40Mb/s]
 99%|#########8| 135M/137M [00:39<00:00, 2.31Mb/s]
 99%|#########8| 135M/137M [00:39<00:00, 2.39Mb/s]
 99%|#########9| 136M/137M [00:39<00:00, 2.52Mb/s]
 99%|#########9| 136M/137M [00:39<00:00, 2.50Mb/s]
100%|#########9| 136M/137M [00:39<00:00, 2.64Mb/s]
100%|#########9| 137M/137M [00:39<00:00, 2.69Mb/s]
100%|##########| 137M/137M [00:39<00:00, 3.45Mb/s]
[INFO] Beginning extraction
```

```
[INFO] Chromium extracted to: C:\Users\youne\AppData\Local\pyppeteer\pyppeteer\local-chr
omium\588429
```

In [ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from neuralprophet import NeuralProphet
from matplotlib import pyplot as plt
import seaborn as sns
import pickle
import datetime as dt
from datetime import timedelta
import re
```

viewing data

In [86]:

```
df = pd.read_csv('weatherHistory.csv')
df.head()
```

Out[86]:

	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover	P (m
0	2006-04-01 00:00:00.000 +0200	Partly Cloudy	rain	9.472222	7.388889	0.89	14.1197	251.0	15.8263	0.0	
1	2006-04-01 01:00:00.000 +0200	Partly Cloudy	rain	9.355556	7.227778	0.86	14.2646	259.0	15.8263	0.0	
2	2006-04-01 02:00:00.000 +0200	Mostly Cloudy	rain	9.377778	9.377778	0.89	3.9284	204.0	14.9569	0.0	
3	2006-04-01 03:00:00.000 +0200	Partly Cloudy	rain	8.288889	5.944444	0.83	14.1036	269.0	15.8263	0.0	
4	2006-04-01 04:00:00.000 +0200	Mostly Cloudy	rain	8.755556	6.977778	0.83	11.0446	259.0	15.8263	0.0	

Cleaning data

In [ ]:

```
df['Year']= df['Formatted Date'].astype(str).apply(lambda x: re.search(r'\d{4}-\d{2}-\d{
print(df['Year'])
```

In [ ]:

```
data = data.drop_duplicates(subset='ds', keep='last')
data
```

data visualisation

In [ ]:

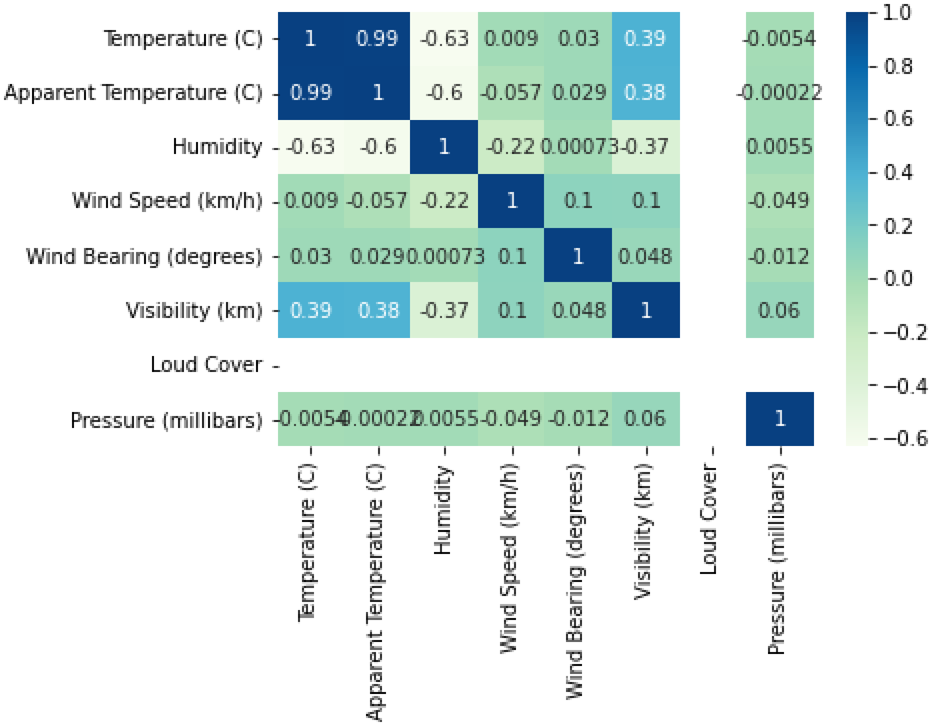
```python
# Convert the 'Year' column to datetime for
df['Year'] = pd.to_datetime(df['Year'],format='%Y-%m-%d')
# Select only the 'Year' column
year_data = df['Year']

# Plot the temperature vs. year
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], df['Temperature (C)'])

# Set the labels to the full dates the axis
plt.title('Temperature over Time')
plt.xlabel('Date')
plt.ylabel('Temperature (C)')
plt.show()
```

In [55]:

```python
#see the corelation beween the data set
sns.heatmap(df.corr(), cmap = 'GnBu', annot = True)
plt.show()
```

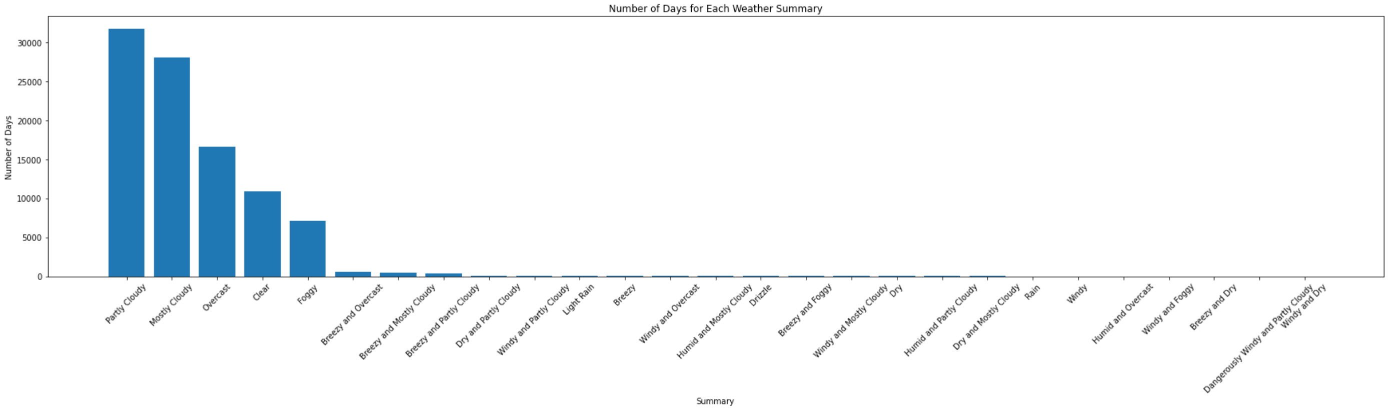


In [56]:

```python
# Get the data for the 'Summary' and 'Formatted Date' columns
summary_data = df[['Summary', 'Formatted Date']]

# Count the number of days for each summary
summary_counts = summary_data['Summary'].value_counts()

# Create a bar chart of the data
plt.figure(figsize=(30, 6))
plt.bar(summary_counts.index, summary_counts.values)
plt.xlabel('Summary')
plt.ylabel('Number of Days')
plt.title('Number of Days for Each Weather Summary')
plt.xticks(rotation=45)
plt.show()
```



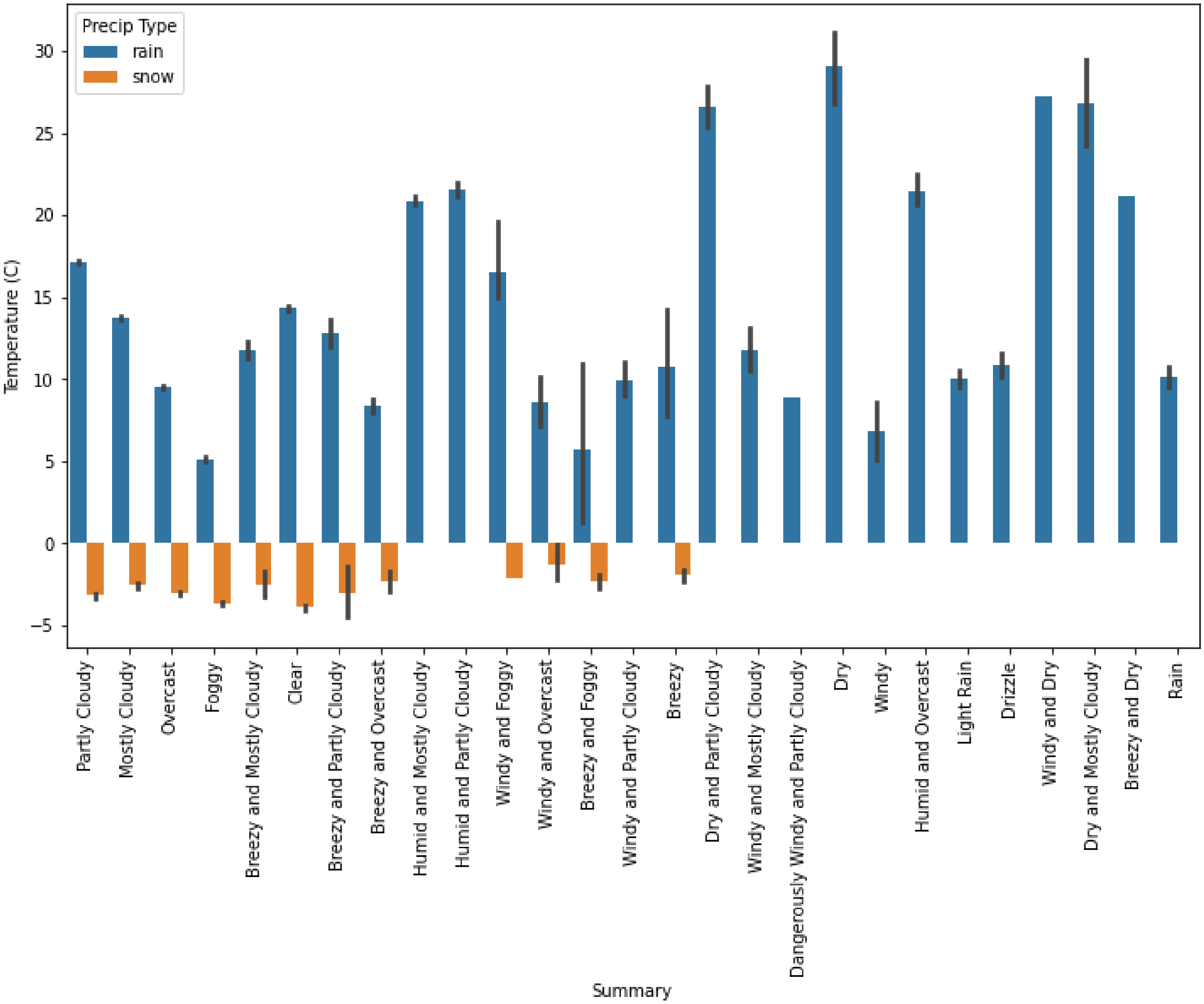
In [73]:

```
# Convert the datetime values to datetime
df["Formatted Date"] = pd.to_datetime(df["Formatted Date"], format="%Y-%m-%d",utc=True)


#plot and show the tepurature vs rain and snow days
plt.figure(figsize=(12,7))
plt.xticks(rotation=90)
sns.barplot(data=df, x="Summary", y="Temperature (C)",hue="Precip Type")
```

Out[73]:

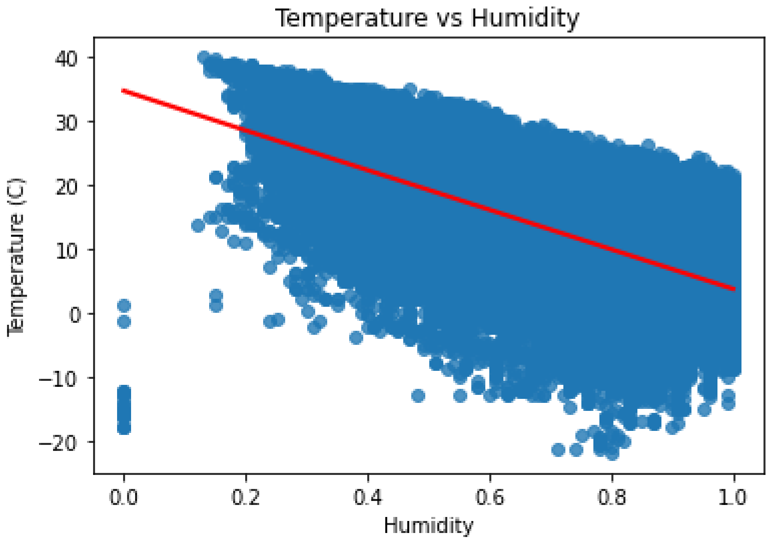
```
<AxesSubplot:xlabel='Summary', ylabel='Temperature (C)'>
```



In [57]:

```
#define our plot
temperature = df['Temperature (C)']
humidity = df['Humidity']
```

```python
# Create a scatter plot and fit a regression line
sns.regplot(x=humidity, y=temperature, line_kws={"color": "red"})
plt.xlabel('Humidity')
plt.ylabel('Temperature (C)')
plt.title('Temperature vs Humidity')
plt.show()
```



Training the model

In [88]:
```python
model_0 = NeuralProphet(epochs=1000)
model_0.fit(data, freq='D')
```

```
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 99.975% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.config.init_data_params) - Setting normalization to global as only one datafr
ame provided for training.
INFO - (NP.utils.set_auto_seasonalities) - Disabling daily seasonality. Run NeuralProphe
t with daily_seasonality=True to override this.
INFO - (NP.config.set_auto_batch_epoch) - Auto-set batch_size to 32
  0%|          | 0/140 [00:00<?, ?it/s]
INFO - (NP.utils_torch.lr_range_test) - lr-range-test results: steep: 7.80E-02, min: 2.0
7E+00
  0%|          | 0/140 [00:00<?, ?it/s]
INFO - (NP.utils_torch.lr_range_test) - lr-range-test results: steep: 7.80E-02, min: 1.1
4E+00
INFO - (NP.forecaster._init_train_loader) - lr-range-test selected learning rate: 8.47E-
02
Epoch[1000/1000]: 100%|██████████| 1000/1000 [03:35<00:00,  4.65it/s, SmoothL1Loss=0.003
87, MAE=2.95, RMSE=3.71, Loss=0.00288, RegLoss=0]
```

Out[88]:

	SmoothL1Loss	MAE	RMSE	Loss	RegLoss
0	0.911916	56.313144	72.406693	0.686427	0.0
1	0.720254	47.386564	61.754921	0.539153	0.0
2	0.551351	39.194261	51.566978	0.409800	0.0
3	0.402388	31.676476	41.951134	0.297077	0.0
4	0.275806	25.096955	33.374571	0.202172	0.0
...	...	...	...	...	...

995	0.003876	2.952610	3.704199	0.002878	0.0
996	0.003875	2.952469	3.697933	0.002877	0.0
997	0.003875	2.952407	3.701122	0.002877	0.0
998	0.003875	2.952310	3.700233	0.002877	0.0
999	0.003875	2.952279	3.705640	0.002877	0.0

1000 rows × 5 columns

Forcasting Tempurature

In [ ]:

```
copy = df.copy() # create a copy of df
copy['Year'] = pd.to_datetime(copy['Year'], format='%Y-%m-%d') # convert to datetime
copy['Year'] = copy['Year'].dt.strftime('%Y-%m-%d') # modify the copy
# Now you can extract the data
data = copy[['Year', 'Temperature (C)']]
data.dropna(inplace=True)
data.columns = ['ds', 'y']
data.head()
```

In [89]:

```
# Create a future dataframe for 2 years into the future
future = model_0.make_future_dataframe(data, periods=1095)

# Predict the future
forecast1 = model_0.predict(future)

# Display the first few rows of the forecast
forecast1.head()
```

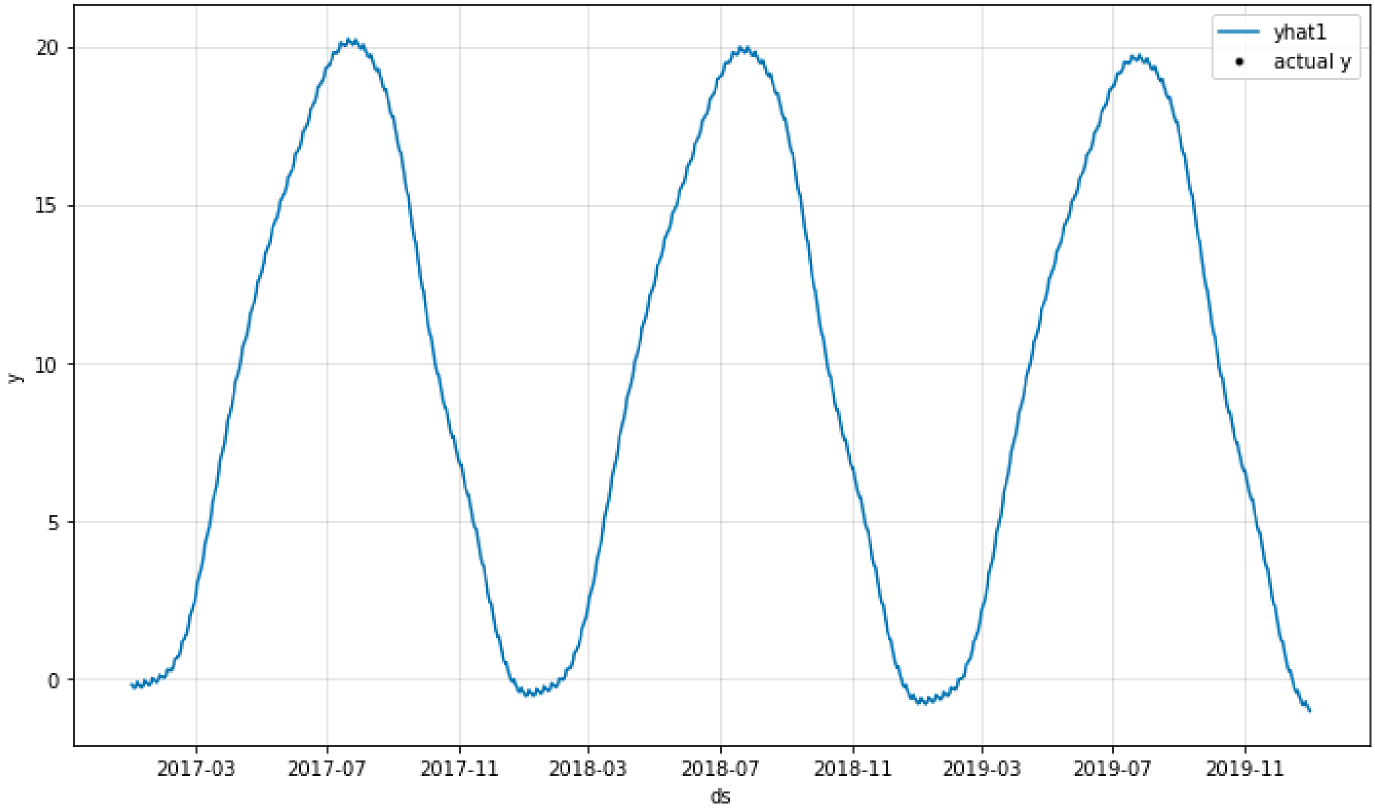
```
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 77.551% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.df_utils.return_df_in_original_format) - Returning df with no ID column
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 99.909% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 99.909% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.df_utils.return_df_in_original_format) - Returning df with no ID column
```

Out[89]:

	ds	y	residual1	yhat1	trend	season_yearly	season_weekly
0	2017-01-01	None	NaN	-0.200571	9.968204	-10.170050	0.001275
1	2017-01-02	None	NaN	-0.213371	9.967516	-10.177994	-0.002893
2	2017-01-03	None	NaN	-0.300238	9.966825	-10.183084	-0.083979
3	2017-01-04	None	NaN	-0.278555	9.966135	-10.185597	-0.059092
4	2017-01-05	None	NaN	-0.249355	9.965445	-10.185810	-0.028991

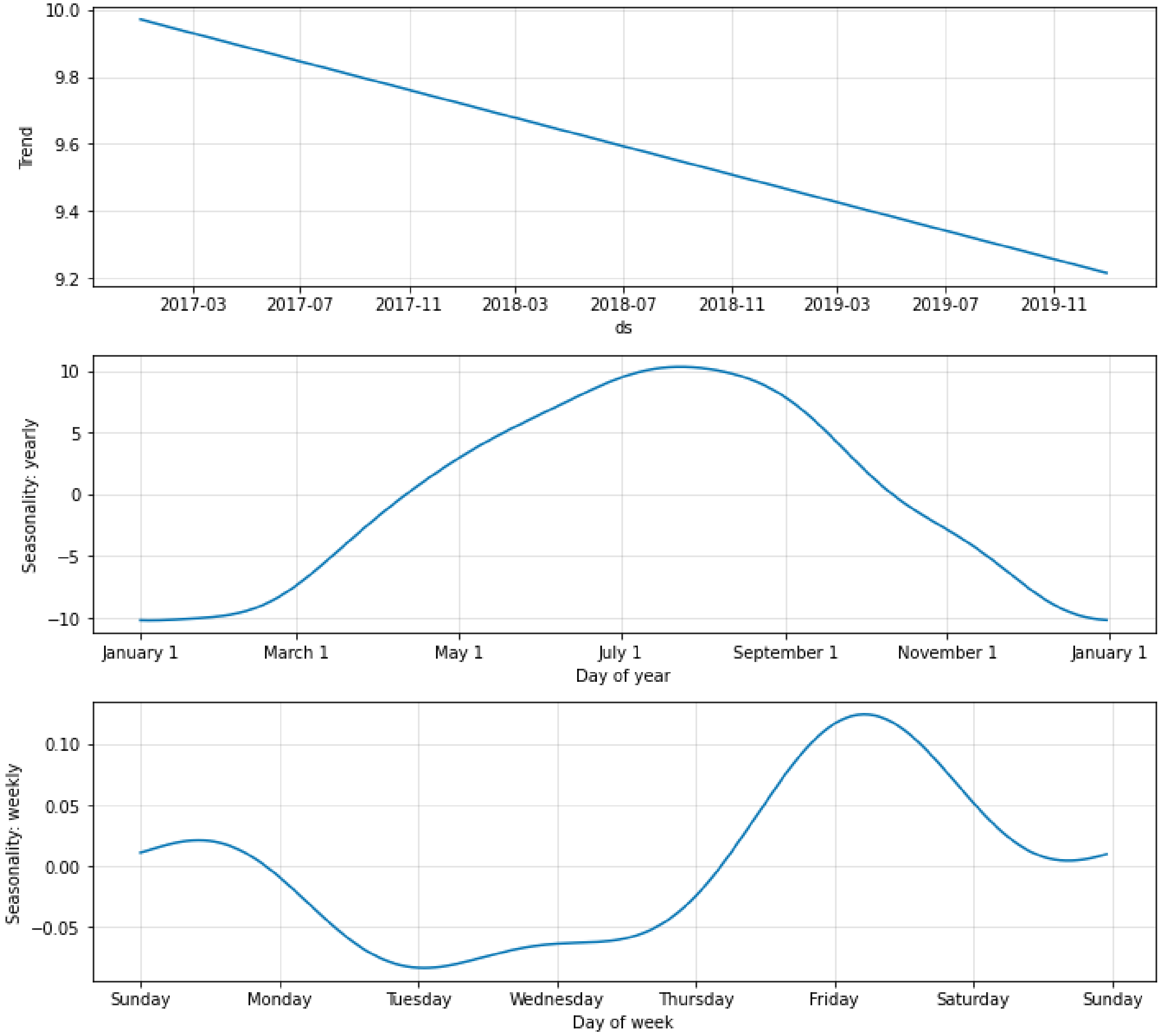
In [74]:

```
#see the plot forcasting of tow years
plot1 = m.plot(forecast1)
```



we can see that the period of havving the tempurature under 0 C is too small its less than 1 month during 2 years while the temourature upper than 5 % is more than than 6 months

```
In [75]: #show the plot of all the composents trend , seasonality,yearly,seasonality weekly
         plt2 = model_0.plot_components(forecast1)
```



In [77]:

```
# Create a future DataFrame for 50 years
future2 = model_0.make_future_dataframe(data, periods=18250)

# Predict the future
forecast2 = model_0.predict(future2)

# Display the first few rows of the forecast
forecast2.head()
```

```
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 77.551% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.df_utils.return_df_in_original_format) - Returning df with no ID column
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 99.995% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 99.995% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.df_utils.return_df_in_original_format) - Returning df with no ID column
```

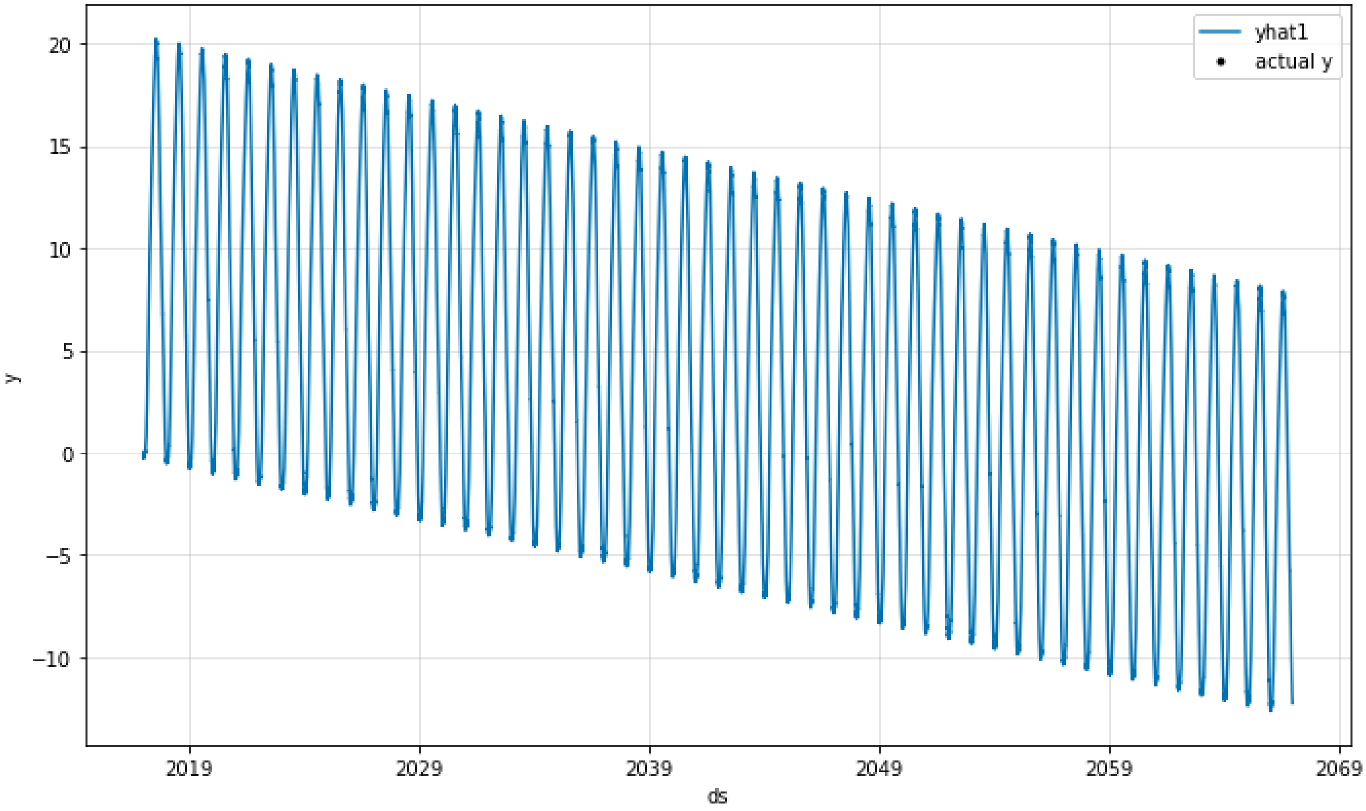
Out[77]:

	ds	y	residual1	yhat1	trend	season_yearly	season_weekly

0	2017-01-01	None	NaN	-0.191397	9.970308	-10.172876	0.011170
1	2017-01-02	None	NaN	-0.219831	9.969618	-10.180647	-0.008803
2	2017-01-03	None	NaN	-0.299591	9.968927	-10.185558	-0.082960
3	2017-01-04	None	NaN	-0.283104	9.968239	-10.187889	-0.063454
4	2017-01-05	None	NaN	-0.244764	9.967550	-10.187909	-0.024404

In [78]:

```
#show the forcast of tempurature in the next 50 years
plot2 = model_0.plot(forecast2)
```



Forcasting precipitation

In [80]:

```
#creat a copy of df
df_precip = df.copy()

# Create the 'y' column
df_precip['y'] = df_precip['Precip Type'].replace(['rain', 'snow'], [1, 0])

# Drop any rows with missing values
df_precip_dropped = df_precip.dropna(subset=['y'])

# Convert the 'Year' column to datetime format
df_precip_dropped['ds'] = pd.to_datetime(df_precip_dropped['Year'])
```

```
WARNING - (py.warnings._showwarnmsg) - C:\Users\youne\anaconda3\envs\myenv\lib\site-pack
ages\ipykernel_launcher.py:10: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
```

In [81]:
```python
df_precip = df.copy()

# Create the 'y' column
df_precip['y'] = df_precip['Precip Type'].replace(['rain', 'snow'], [1, 0])

# Drop any rows with missing values
df_precip_dropped = df_precip.dropna(subset=['y'])

# Convert the 'Year' column to datetime format
df_precip_dropped['ds'] = pd.to_datetime(df_precip_dropped['Year'])

# Drop duplicates in the 'ds' column
df_precip_dropped.drop_duplicates(subset=['ds'], inplace=True)

# Create the NeuralProphet model
model = NeuralProphet(epochs=500)

# Fit the model to the data
model.fit(df_precip_dropped[['ds','y']], freq='D')
```

```
WARNING - (py.warnings._showwarnmsg) - C:\Users\youne\anaconda3\envs\myenv\lib\site-pack
ages\ipykernel_launcher.py:10: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy


WARNING - (py.warnings._showwarnmsg) - C:\Users\youne\anaconda3\envs\myenv\lib\site-pack
ages\ipykernel_launcher.py:13: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy


INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 99.85% of the d
ata.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.config.init_data_params) - Setting normalization to global as only one datafr
ame provided for training.
INFO - (NP.utils.set_auto_seasonalities) - Disabling daily seasonality. Run NeuralProphe
t with daily_seasonality=True to override this.
INFO - (NP.config.set_auto_batch_epoch) - Auto-set batch_size to 32
  0%|          | 0/140 [00:00<?, ?it/s]
INFO - (NP.utils_torch.lr_range_test) - lr-range-test results: steep: 1.64E-01, min: 3.4
6E-01
  0%|          | 0/140 [00:00<?, ?it/s]
INFO - (NP.utils_torch.lr_range_test) - lr-range-test results: steep: 1.64E-01, min: 3.4
6E-01
INFO - (NP.forecaster._init_train_loader) - lr-range-test selected learning rate: 1.39E-
01
Epoch[500/500]: 100%|██████████| 500/500 [01:40<00:00,  4.96it/s, SmoothL1Loss=0.0391, M
AE=0.173, RMSE=0.276, Loss=0.0289, RegLoss=0]
```

Out[81]:

	SmoothL1Loss	MAE	RMSE	Loss	RegLoss

0	2.647092	3.142042	3.418085	2.000410	0.0
1	2.302409	2.794845	3.089022	1.724173	0.0
2	1.953063	2.438768	2.745778	1.445091	0.0
3	1.604627	2.076968	2.397204	1.167608	0.0
4	1.263476	1.717243	2.028927	0.898092	0.0
...	...	...	...	...	...
495	0.039099	0.172568	0.275808	0.028888	0.0
496	0.039098	0.172686	0.274148	0.028887	0.0
497	0.039093	0.172654	0.275486	0.028883	0.0
498	0.039090	0.172638	0.276090	0.028880	0.0
499	0.039088	0.172641	0.276135	0.028879	0.0

500 rows × 5 columns

In [82]:

```
#make aprediction of 50 years
future3 = model.make_future_dataframe(df_precip_dropped[['ds','y']],periods=18250)

# Make predictions on the future dataframe
forecast3 = model.predict(future3)

# Print the forecast
print(forecast3[['ds', 'y']])
```

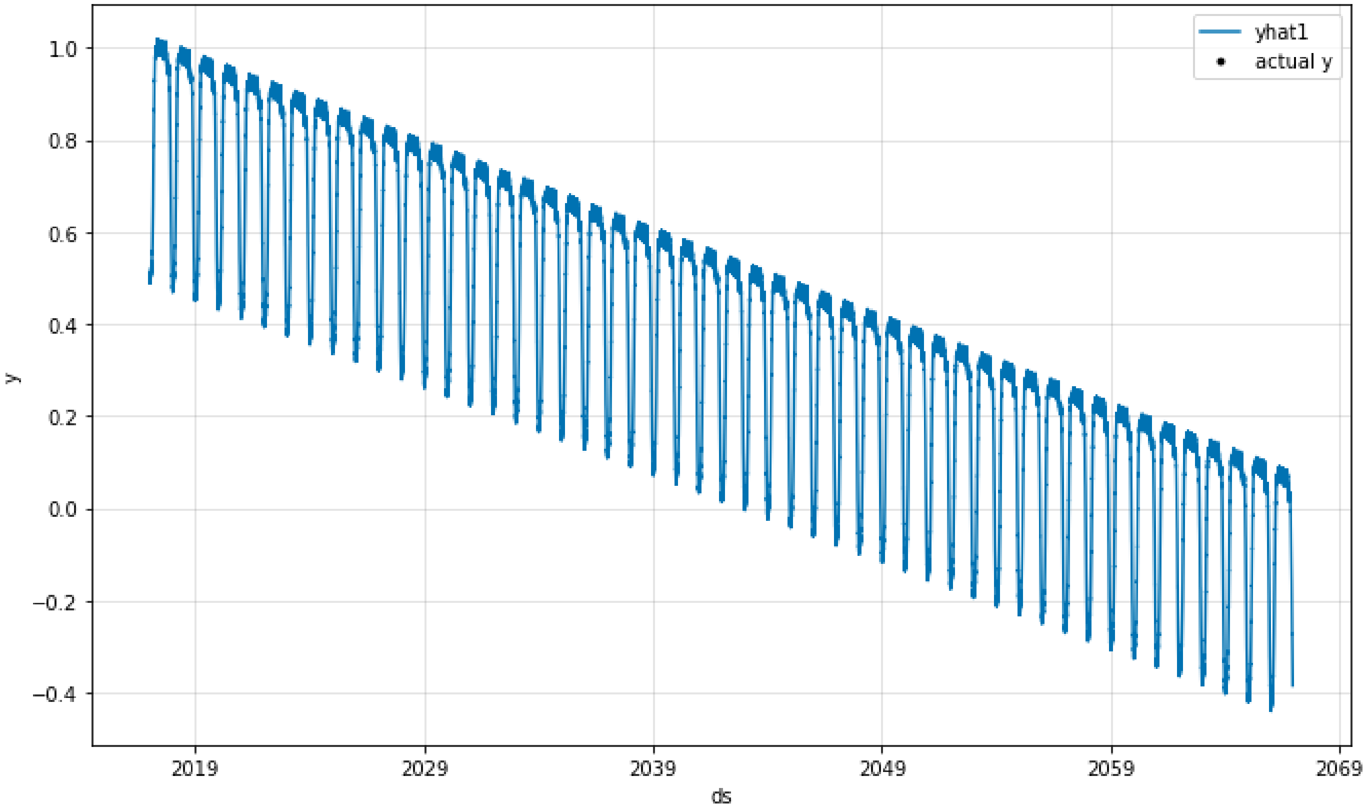
```
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 77.501% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.df_utils.return_df_in_original_format) - Returning df with no ID column
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 99.995% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 99.995% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.df_utils.return_df_in_original_format) - Returning df with no ID column
```

```
              ds     y
0     2017-01-01  None
1     2017-01-02  None
2     2017-01-03  None
3     2017-01-04  None
4     2017-01-05  None
...          ...   ...
18245 2066-12-15  None
18246 2066-12-16  None
18247 2066-12-17  None
18248 2066-12-18  None
18249 2066-12-19  None

[18250 rows x 2 columns]
```

In [83]:

```
plot3 = model.plot(forecast3)
```



In [84]:

```
# #make aprediction of 3 years
future4 = m.make_future_dataframe(df_precip_dropped[['ds','y']], periods=1095)

# Predict the future
forecast4 = model.predict(future4)

# Display the first few rows of the forecast
forecast4.head()
```

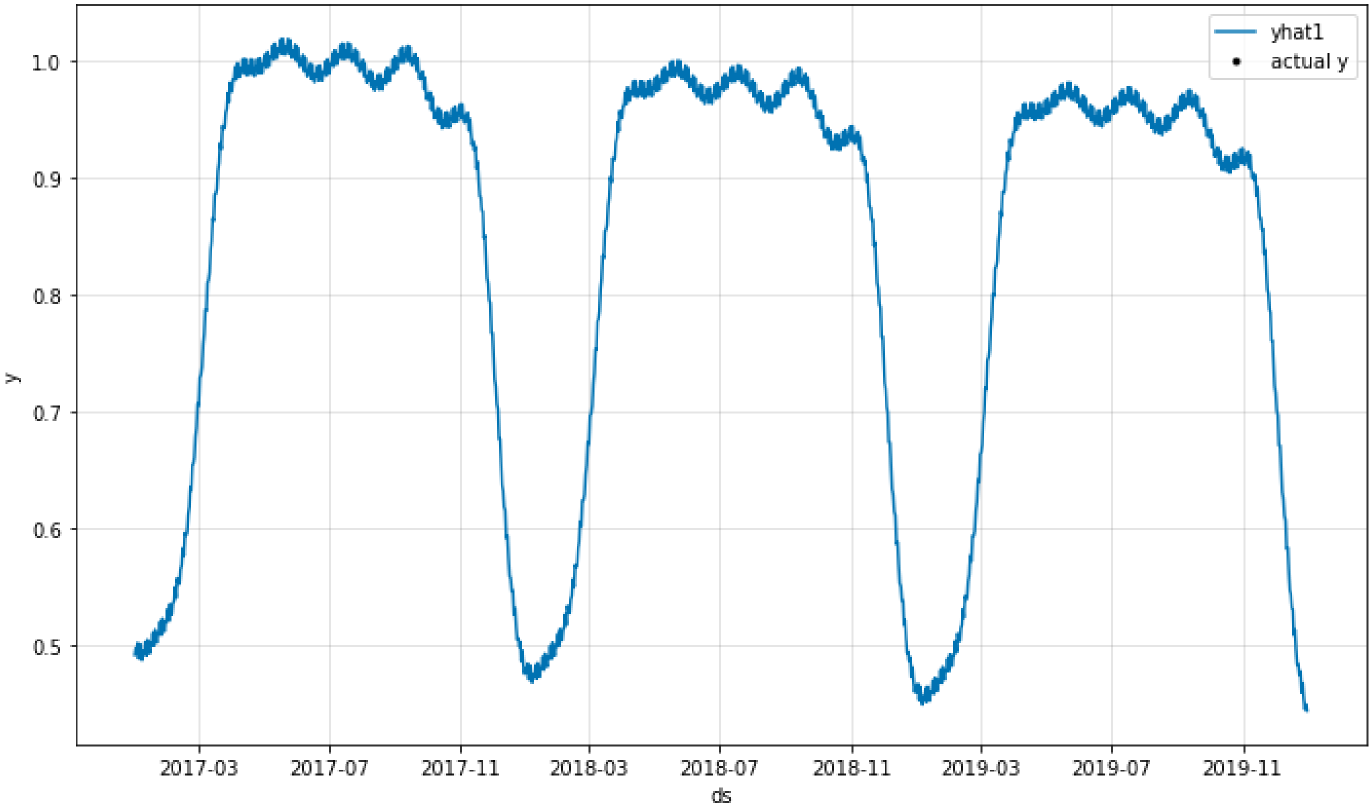
```
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 77.501% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.df_utils.return_df_in_original_format) - Returning df with no ID column
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 99.909% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.df_utils._infer_frequency) - Major frequency D corresponds to 99.909% of the
data.
INFO - (NP.df_utils._infer_frequency) - Defined frequency is equal to major frequency -
D
INFO - (NP.df_utils.return_df_in_original_format) - Returning df with no ID column
```

Out[84]:

	ds	y	residual1	yhat1	trend	season_yearly	season_weekly
0	2017-01-01	None	NaN	0.492182	0.874703	-0.375863	-0.006659
1	2017-01-02	None	NaN	0.497760	0.874651	-0.377505	0.000613
2	2017-01-03	None	NaN	0.494278	0.874600	-0.378764	-0.001558
3	2017-01-04	None	NaN	0.502063	0.874548	-0.379665	0.007180
4	2017-01-05	None	NaN	0.488972	0.874496	-0.380237	-0.005286

In [85]:

```
plot4 = model.plot(forecast4)
```


yhat1
actual y
y
1.0
0.9
0.8
0.7
0.6
0.5
ds
2017-03
2017-07
2017-11
2018-03
2018-07
2018-11
2019-03
2019-07
2019-11