Loading Datasets

```
import xarray as xr
ds_fc = xr.open_dataset("fourcastnetv2-small_20210601_0000_forecast.nc")
```

```
import xarray as xr
ds_fc2 = "/content/drive/MyDrive/Fmodel/merged_era5_6hourly_june2021.nc"
ds1 = xr.open_dataset(ds_fc2)
```

Subsetting Datasets to India Region

```
# Define India region
ds_fc_india = ds_fc.sel(latitude=slice(35, 5), longitude=slice(65, 100))
ds1_india = ds1.sel(latitude=slice(35, 5), longitude=slice(65, 100))
```

```
# Interpolate ERA5 to ForecastNet times
msl_era5_interp = ds1_india["msl"].interp(valid_time=ds_fc_india.time)
```

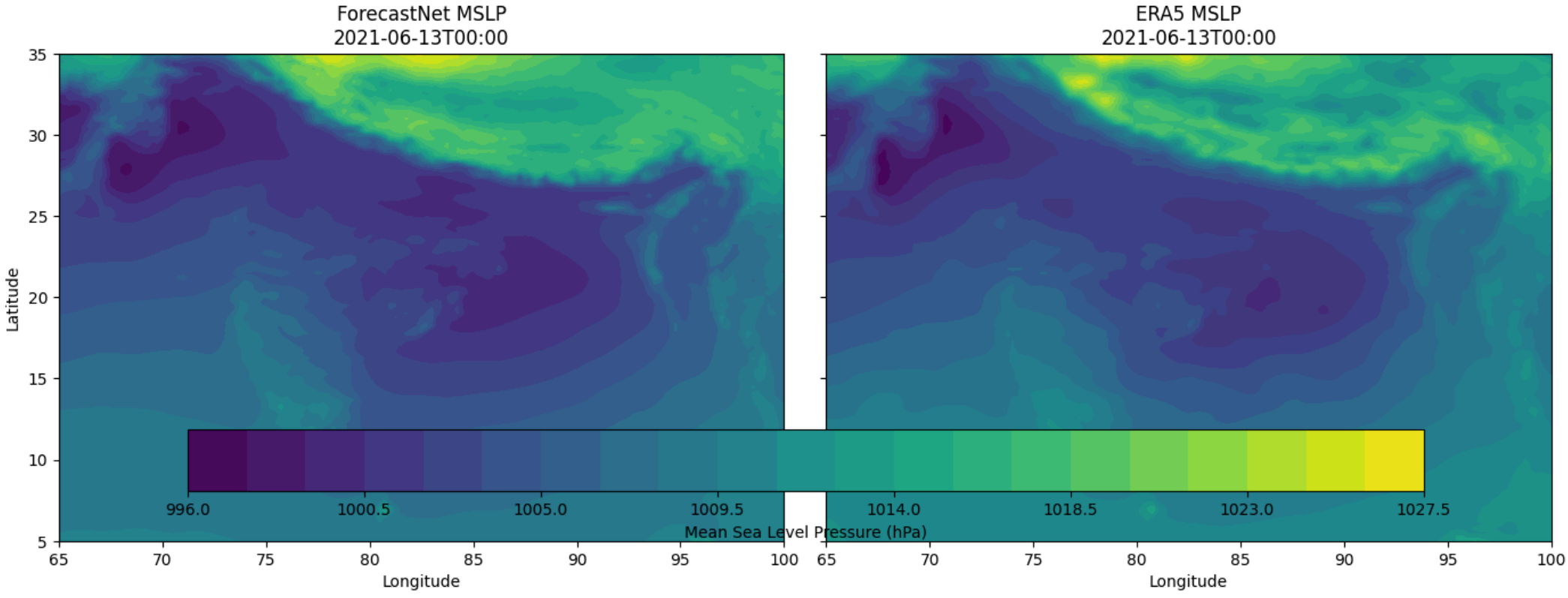
```
# Drop pressure_level if it exists
if "pressure_level" in msl_era5_interp.dims:
    msl_era5_interp = msl_era5_interp.isel(pressure_level=0, drop=True)
```

Variable : Mean Sea Level Pressure (mslp)

```
import matplotlib.pyplot as plt
import ipywidgets as widgets
from ipywidgets import interact
import numpy as np

# Convert to hPa
msl_fc_hpa = ds_fc_india["msl"] / 100
msl_era5_hpa = msl_era5_interp / 100

# Drop pressure level if mistakenly included
if "pressure_level" in msl_era5_hpa.dims:
    msl_era5_hpa = msl_era5_hpa.isel(pressure_level=0, drop=True)

# Create plotting function
def plot_mslp_interactive(time_idx):
    fig, axs = plt.subplots(1, 2, figsize=(14, 5), sharex=True, sharey=True)

    # ForecastNet
    cf1 = axs[0].contourf(ds_fc_india.longitude, ds_fc_india.latitude,
                          msl_fc_hpa.isel(time=time_idx), levels=20, cmap="viridis")
    axs[0].set_title(f"ForecastNet MSLP\n{str(ds_fc_india.time.values[time_idx])[:16]}")
    axs[0].set_xlabel("Longitude")
    axs[0].set_ylabel("Latitude")

    # ERA5
    cf2 = axs[1].contourf(ds1_india.longitude, ds1_india.latitude,
                          msl_era5_hpa.isel(time=time_idx), levels=20, cmap="viridis")
    axs[1].set_title(f"ERA5 MSLP\n{str(ds_fc_india.time.values[time_idx])[:16]}")
    axs[1].set_xlabel("Longitude")

    cbar = fig.colorbar(cf1, ax=axs, orientation='horizontal', pad=0.08)
    cbar.set_label("Mean Sea Level Pressure (hPa)")

    plt.tight_layout()
    plt.show()

# Create interactive slider
interact(plot_mslp_interactive, time_idx=widgets.IntSlider(min=0, max=len(ds_fc_india.time)-1, step=1, value=0));
```

time_idx 48

```
/tmp/ipython-input-24-2023116686.py:34: UserWarning: This figure includes Axes that are not compatible with tight_layout, so results might be incorrect.
  plt.tight_layout()
```

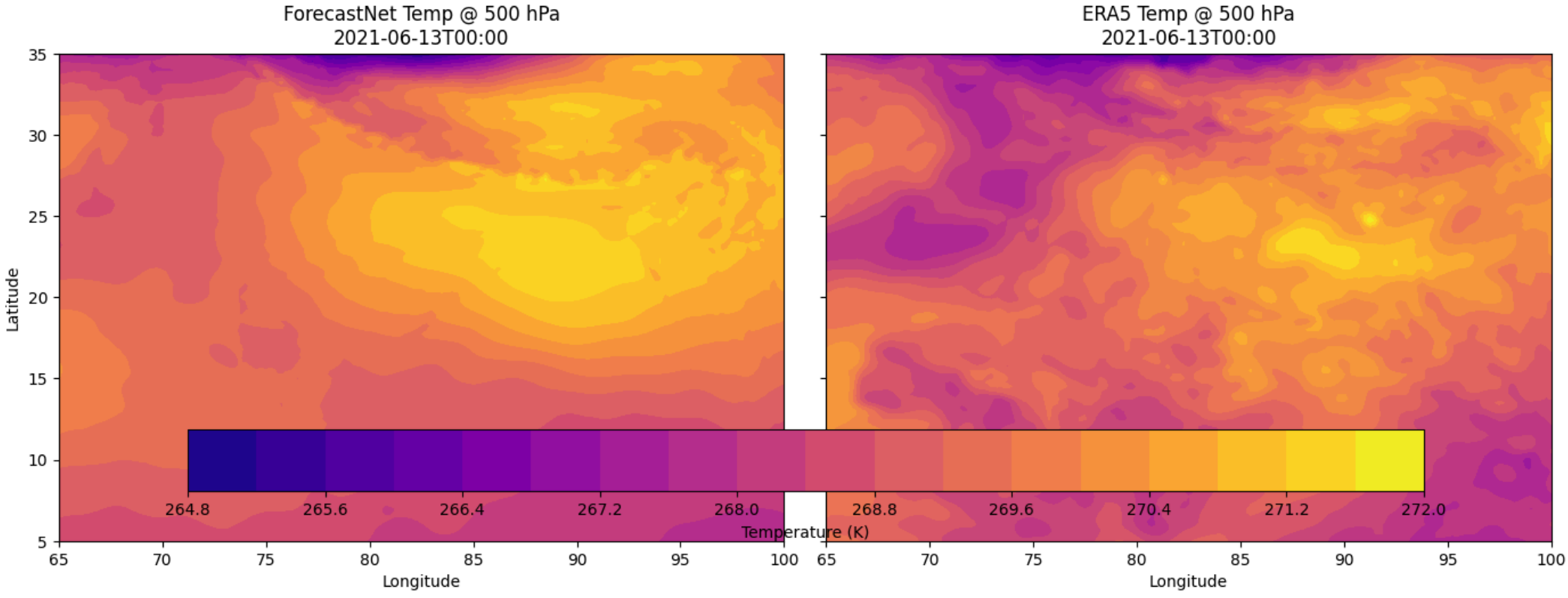


Variable : Temperature (t)

```
import matplotlib.pyplot as plt
import ipywidgets as widgets
from ipywidgets import interact

# Choose pressure level to visualize
level = 500

# Select temperature at 500 hPa
t_fc_500 = ds_fc_india["t"].sel(level=level)
t_era5_500 = ds1_india["t"].sel(pressure_level=level)

# Interpolate ERA5 time to match ForecastNet
t_era5_interp = t_era5_500.interp(valid_time=ds_fc_india.time)

# Interactive plotting function
def plot_temp_interactive(time_idx):
    fig, axs = plt.subplots(1, 2, figsize=(14, 5), sharex=True, sharey=True)

    # ForecastNet temp
    cf1 = axs[0].contourf(ds_fc_india.longitude, ds_fc_india.latitude,
                          t_fc_500.isel(time=time_idx), levels=20, cmap="plasma")
    axs[0].set_title(f"ForecastNet Temp @ {level} hPa\n{str(ds_fc_india.time.values[time_idx])[:16]}")
    axs[0].set_xlabel("Longitude")
    axs[0].set_ylabel("Latitude")

    # ERA5 temp
    cf2 = axs[1].contourf(ds1_india.longitude, ds1_india.latitude,
                          t_era5_interp.isel(time=time_idx), levels=20, cmap="plasma")
    axs[1].set_title(f"ERA5 Temp @ {level} hPa\n{str(ds_fc_india.time.values[time_idx])[:16]}")
    axs[1].set_xlabel("Longitude")

    # Shared colorbar
    cbar = fig.colorbar(cf1, ax=axs, orientation='horizontal', pad=0.08)
    cbar.set_label("Temperature (K)")

    plt.tight_layout()
    plt.show()

# Interactive slider
interact(plot_temp_interactive, time_idx=widgets.IntSlider(min=0, max=len(ds_fc_india.time)-1, step=1, value=0));
```

time_idx 48

```
/tmp/ipython-input-25-3397455138.py:36: UserWarning: This figure includes Axes that are not compatible with tight_layout, so results might be incorrect.
  plt.tight_layout()
```

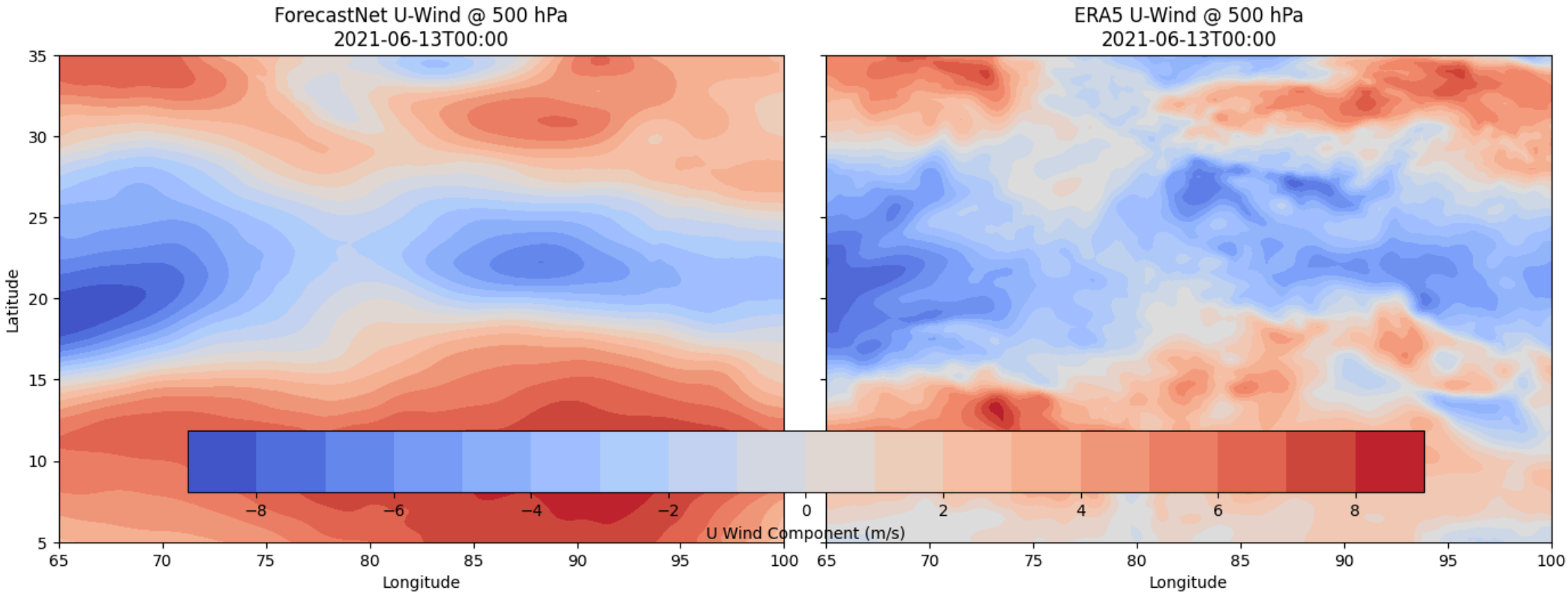


Variable : U-Wind

```
import matplotlib.pyplot as plt
import ipywidgets as widgets
from ipywidgets import interact

# Select pressure level
level = 500

# Slice U wind at 500 hPa
u_fc_500 = ds_fc_india["u"].sel(level=level)
u_era5_500 = ds1_india["u"].sel(pressure_level=level)

# Interpolate ERA5 U-wind to ForecastNet time steps
u_era5_interp = u_era5_500.interp(valid_time=ds_fc_india.time)

# Define interactive plotting function
def plot_u_interactive(time_idx):
    fig, axs = plt.subplots(1, 2, figsize=(14, 5), sharex=True, sharey=True)

    # ForecastNet U-wind
    cf1 = axs[0].contourf(ds_fc_india.longitude, ds_fc_india.latitude,
                          u_fc_500.isel(time=time_idx), levels=20, cmap="coolwarm")
    axs[0].set_title(f"ForecastNet U-Wind @ {level} hPa\n{str(ds_fc_india.time.values[time_idx])[:16]}")
    axs[0].set_xlabel("Longitude")
    axs[0].set_ylabel("Latitude")

    # ERA5 U-wind
    cf2 = axs[1].contourf(ds1_india.longitude, ds1_india.latitude,
                          u_era5_interp.isel(time=time_idx), levels=20, cmap="coolwarm")
    axs[1].set_title(f"ERA5 U-Wind @ {level} hPa\n{str(ds_fc_india.time.values[time_idx])[:16]}")
    axs[1].set_xlabel("Longitude")

    # Shared colorbar
    cbar = fig.colorbar(cf1, ax=axs, orientation='horizontal', pad=0.08)
    cbar.set_label("U Wind Component (m/s)")

    plt.tight_layout()
    plt.show()

# Interactive slider
interact(plot_u_interactive, time_idx=widgets.IntSlider(min=0, max=len(ds_fc_india.time)-1, step=1, value=0));
```

time_idx 48

```
/tmp/ipython-input-26-2440011192.py:36: UserWarning: This figure includes Axes that are not compatible with tight_layout, so results might be incorrect.
  plt.tight_layout()
```

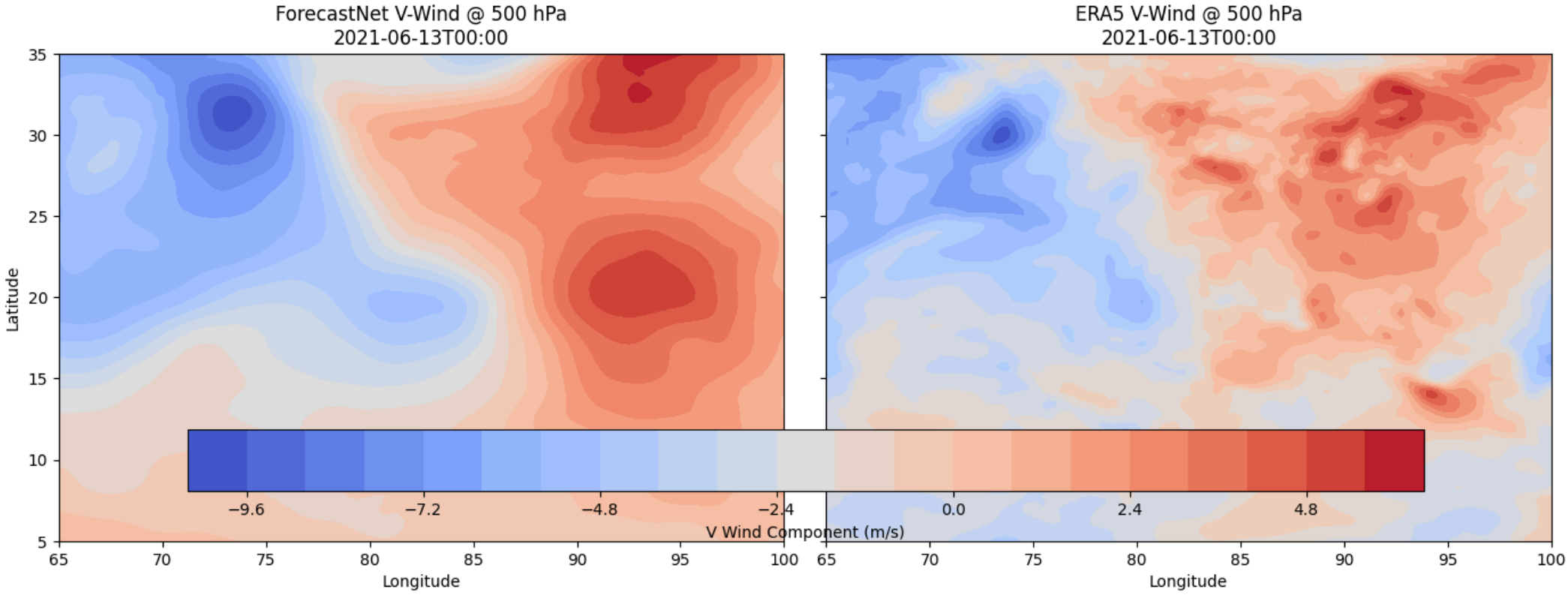


Variable : V-Wind

```
import matplotlib.pyplot as plt
import ipywidgets as widgets
from ipywidgets import interact

# Define pressure level
level = 500

# Slice the V wind data
v_fc_500 = ds_fc_india["v"].sel(level=level)
v_era5_500 = ds1_india["v"].sel(pressure_level=level)

# Interpolate ERA5 V wind to ForecastNet time
v_era5_interp = v_era5_500.interp(valid_time=ds_fc_india.time)

# Define plotting function
def plot_v_interactive(time_idx):
    fig, axs = plt.subplots(1, 2, figsize=(14, 5), sharex=True, sharey=True)

    # ForecastNet V wind
    cf1 = axs[0].contourf(ds_fc_india.longitude, ds_fc_india.latitude,
                          v_fc_500.isel(time=time_idx), levels=20, cmap="coolwarm")
    axs[0].set_title(f"ForecastNet V-Wind @ {level} hPa\n{str(ds_fc_india.time.values[time_idx])[:16]}")
    axs[0].set_xlabel("Longitude")
    axs[0].set_ylabel("Latitude")

    # ERA5 V wind
    cf2 = axs[1].contourf(ds1_india.longitude, ds1_india.latitude,
                          v_era5_interp.isel(time=time_idx), levels=20, cmap="coolwarm")
    axs[1].set_title(f"ERA5 V-Wind @ {level} hPa\n{str(ds_fc_india.time.values[time_idx])[:16]}")
    axs[1].set_xlabel("Longitude")

    # Shared colorbar
    cbar = fig.colorbar(cf1, ax=axs, orientation='horizontal', pad=0.08)
    cbar.set_label("V Wind Component (m/s)")

    plt.tight_layout()
    plt.show()

# Create interactive slider
interact(plot_v_interactive, time_idx=widgets.IntSlider(min=0, max=len(ds_fc_india.time)-1, step=1, value=0));
```

time_idx 48

```
/tmp/ipython-input-27-3518229210.py:36: UserWarning: This figure includes Axes that are not compatible with tight_layout, so results might be incorrect.
  plt.tight_layout()
```



RMSE Results

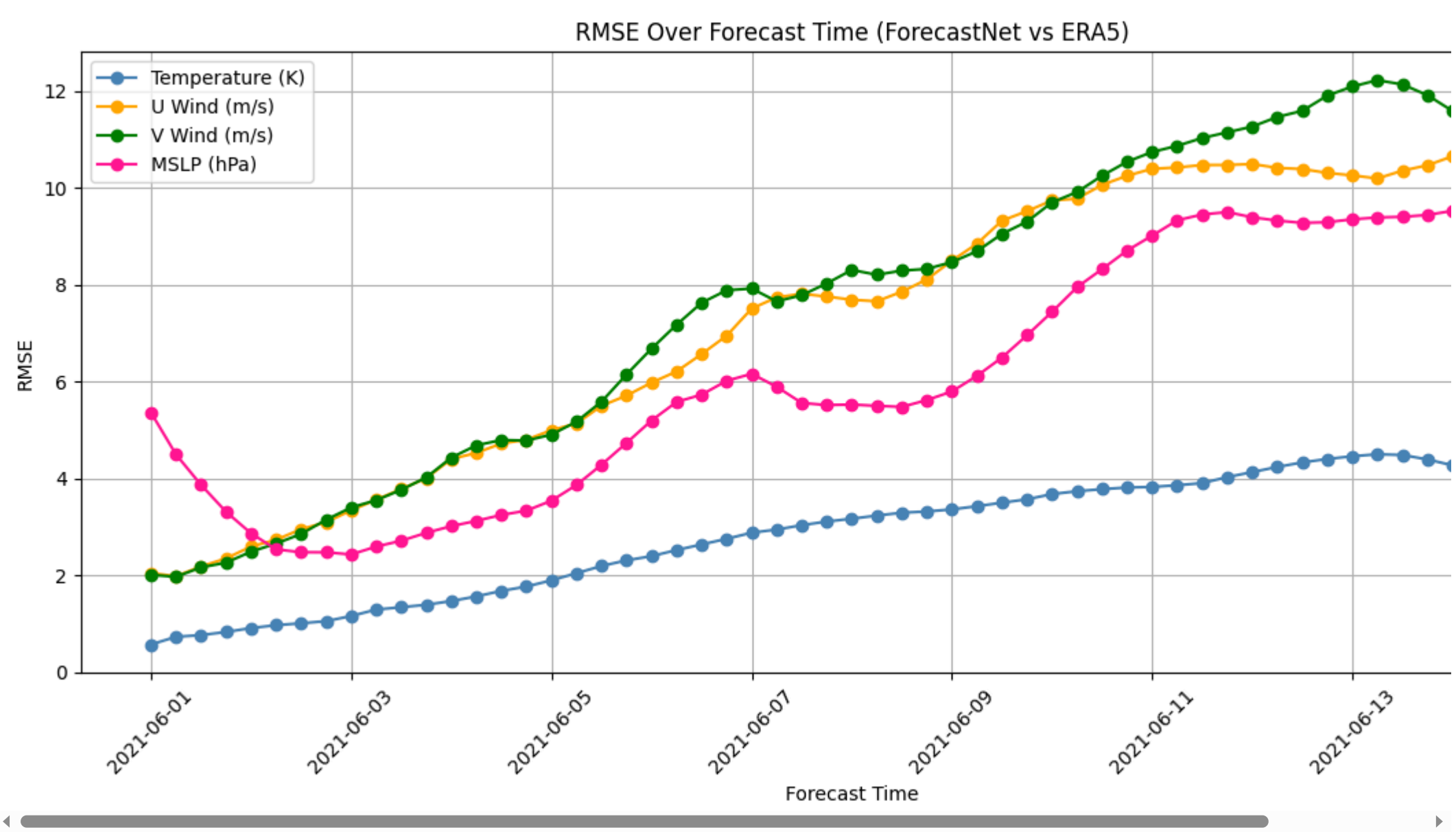
```
# Forecast time
forecast_time = ds_fc["time"].values[:ref_len]
import pandas as pd
# Now build dataframe
rmse_df = pd.DataFrame(rmse_results)
rmse_df["Forecast Time"] = forecast_time
rmse_df = rmse_df.set_index("Forecast Time")
rmse_df
```

Forecast Time	Temperature (K)	U Wind (m/s)	V Wind (m/s)	MSLP (hPa)
2021-06-01 00:00:00	0.567611	2.040490	2.006033	5.357902
2021-06-01 06:00:00	0.726083	1.978263	1.965675	4.491093
2021-06-01 12:00:00	0.761755	2.177729	2.159331	3.863192
2021-06-01 18:00:00	0.825834	2.344900	2.257135	3.307790
2021-06-02 00:00:00	0.906893	2.590028	2.483946	2.862675
2021-06-02 06:00:00	0.966273	2.729698	2.651341	2.531948
2021-06-02 12:00:00	1.005784	2.948237	2.849806	2.472473
2021-06-02 18:00:00	1.049587	3.091767	3.142971	2.470992
2021-06-03 00:00:00	1.155090	3.337343	3.396127	2.426482
2021-06-03 06:00:00	1.289347	3.557489	3.546922	2.589313
2021-06-03 12:00:00	1.337159	3.777090	3.760499	2.709373
2021-06-03 18:00:00	1.387937	4.000646	4.017128	2.873369
2021-06-04 00:00:00	1.464857	4.396349	4.425773	3.011152
2021-06-04 06:00:00	1.559978	4.522841	4.679055	3.119281
2021-06-04 12:00:00	1.674673	4.716729	4.789184	3.242214
2021-06-04 18:00:00	1.763141	4.794450	4.777907	3.330666
2021-06-05 00:00:00	1.892642	4.992514	4.900721	3.536559
2021-06-05 06:00:00	2.039128	5.130029	5.167902	3.860447
2021-06-05 12:00:00	2.188214	5.494728	5.580122	4.279283
2021-06-05 18:00:00	2.304711	5.706841	6.141911	4.729352
2021-06-06 00:00:00	2.391596	5.973625	6.676851	5.192604
2021-06-06 06:00:00	2.517098	6.205179	7.169339	5.575457
2021-06-06 12:00:00	2.633392	6.563279	7.622087	5.725490
2021-06-06 18:00:00	2.748735	6.942316	7.880948	6.011263
2021-06-07 00:00:00	2.875625	7.503169	7.916451	6.156361
2021-06-07 06:00:00	2.940624	7.733841	7.652524	5.888730
2021-06-07 12:00:00	3.030372	7.808882	7.778702	5.554485
2021-06-07 18:00:00	3.106707	7.751067	8.022545	5.511023
2021-06-08 00:00:00	3.164327	7.685421	8.300303	5.520690
2021-06-08 06:00:00	3.229426	7.653758	8.201265	5.495213
2021-06-08 12:00:00	3.288278	7.851122	8.286611	5.472866
2021-06-08 18:00:00	3.313845	8.104570	8.321245	5.613756
2021-06-09 00:00:00	3.356892	8.493175	8.463041	5.791589
2021-06-09 06:00:00	3.422353	8.840855	8.685683	6.113654
2021-06-09 12:00:00	3.495062	9.318202	9.045417	6.491855
2021-06-09 18:00:00	3.564876	9.514299	9.297268	6.952192
2021-06-10 00:00:00	3.672852	9.735283	9.695796	7.434284
2021-06-10 06:00:00	3.731934	9.771060	9.909012	7.944892
2021-06-10 12:00:00	3.774706	10.056285	10.246139	8.317457
2021-06-10 18:00:00	3.808549	10.245178	10.534641	8.701567
2021-06-11 00:00:00	3.820959	10.389107	10.736007	9.008335
2021-06-11 06:00:00	3.855478	10.415798	10.862500	9.325672
2021-06-11 12:00:00	3.899860	10.466855	11.024010	9.445374
2021-06-11 18:00:00	4.023037	10.468289	11.141550	9.495036
2021-06-12 00:00:00	4.127351	10.487207	11.260357	9.387617
2021-06-12 06:00:00	4.233347	10.408162	11.458743	9.319599
2021-06-12 12:00:00	4.326609	10.383472	11.585682	9.271946

2021-06-12 18:00:00	4.397672	10.303564	11.900157	9.286025
2021-06-13 00:00:00	4.454244	10.252449	12.087485	9.345904
2021-06-13 06:00:00	4.495209	10.189449	12.217885	9.384074
2021-06-13 12:00:00	4.480389	10.349914	12.130525	9.397340
2021-06-13 18:00:00	4.387974	10.461790	11.907650	9.435796
2021-06-14 00:00:00	4.267035	10.647014	11.582363	9.521371
2021-06-14 06:00:00	4.155679	10.748658	11.194910	9.606395
2021-06-14 12:00:00	4.100302	10.834351	10.901707	9.600846
2021-06-14 18:00:00	4.074737	10.772572	10.607791	9.580409
2021-06-15 00:00:00	4.022849	10.679074	10.438162	9.496664

RMSE Results Visualisation

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))

# Plot each variable once
colors = {
    "Temperature (K)": "steelblue",
    "U Wind (m/s)": "orange",
    "V Wind (m/s)": "green",
    "MSLP (hPa)": "deeppink"  # Set custom color for MSLP
}

for var, rmse in rmse_results.items():
    plt.plot(ds_fc.time, rmse, label=var, marker='o', color=colors.get(var, None))

plt.title("RMSE Over Forecast Time (ForecastNet vs ERA5)")
plt.xlabel("Forecast Time")
plt.ylabel("RMSE")
plt.grid(True)
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()
```




```
import matplotlib.pyplot as plt
```