```
#!pip3 install pystan
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/p
Collecting pystan
  Downloading pystan-3.6.0-py3-none-any.whl (13 kB)
Requirement already satisfied: setuptools in /usr/local/lib/python3.9/dist-packages (
Collecting aiohttp<4.0,>=3.6
  Downloading aiohttp-3.8.4-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.0/1.0 MB 13.7 MB/s eta 0:00:00
Collecting httpstan<4.10,>=4.9
  Downloading httpstan-4.9.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.4/43.4 MB 17.9 MB/s eta 0:00:00
Collecting pysimdjson<6.0.0,>=5.0.2
  Downloading pysimdjson-5.0.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.w
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 48.5 MB/s eta 0:00:00
Collecting clikit<0.7,>=0.6
  Downloading clikit-0.6.2-py2.py3-none-any.whl (91 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 91.8/91.8 KB 9.2 MB/s eta 0:00:00
Requirement already satisfied: numpy<2.0,>=1.19 in /usr/local/lib/python3.9/dist-pack
Collecting frozenlist>=1.1.1
  Downloading frozenlist-1.3.3-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64.manyl
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 158.8/158.8 KB 15.3 MB/s eta 0:00:00
Collecting async-timeout<5.0,>=4.0.0a3
  Downloading async_timeout-4.0.2-py3-none-any.whl (5.8 kB)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in /usr/local/lib/python3
Collecting yarl<2.0,>=1.0
  Downloading yarl-1.8.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (26
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 264.6/264.6 KB 21.7 MB/s eta 0:00:00
Collecting aiosignal>=1.1.2
  Downloading aiosignal-1.3.1-py3-none-any.whl (7.6 kB)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.9/dist-package
Collecting multidict<7.0,>=4.5
  Downloading multidict-6.0.4-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.wh
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 114.2/114.2 KB 10.5 MB/s eta 0:00:00
Collecting crashtest<0.4.0,>=0.3.0
  Downloading crashtest-0.3.1-py3-none-any.whl (7.0 kB)
Collecting pastel<0.3.0,>=0.2.0
  Downloading pastel-0.2.1-py2.py3-none-any.whl (6.0 kB)
Collecting pylev<2.0,>=1.3
  Downloading pylev-1.4.0-py2.py3-none-any.whl (6.1 kB)
Collecting webargs<9.0,>=8.0
  Downloading webargs-8.2.0-py3-none-any.whl (30 kB)
Requirement already satisfied: appdirs<2.0,>=1.4 in /usr/local/lib/python3.9/dist-pac
Collecting marshmallow<4.0,>=3.10
  Downloading marshmallow-3.19.0-py3-none-any.whl (49 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.1/49.1 KB 5.2 MB/s eta 0:00:00
Requirement already satisfied: packaging>=17.0 in /usr/local/lib/python3.9/dist-packa
Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.9/dist-packages (f
Installing collected packages: pylev, pysimdjson, pastel, multidict, marshmallow, fro
Successfully installed aiohttp-3.8.4 aiosignal-1.3.1 async-timeout-4.0.2 clikit-0.6.2
```

```
!pip install prophet
```

```
import prophet
```

```
#!pip install convertdate
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/p
Requirement already satisfied: convertdate in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: pymeeus<=1,>=0.3.13 in /usr/local/lib/python3.9/dist-p
```

```
#!pip install lunarcalendar
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/p
Requirement already satisfied: lunarcalendar in /usr/local/lib/python3.9/dist-package
Requirement already satisfied: ephem>=3.7.5.3 in /usr/local/lib/python3.9/dist-packag
Requirement already satisfied: python-dateutil>=2.6.1 in /usr/local/lib/python3.9/dis
Requirement already satisfied: pytz in /usr/local/lib/python3.9/dist-packages (from l
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-packages (fr
```

```
#!pip install pmdarima
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/p
Collecting pmdarima
  Downloading pmdarima-2.0.3-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.mar
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 34.0 MB/s eta 0:00:00
Requirement already satisfied: Cython!=0.29.18,!=0.29.31,>=0.29 in /usr/local/lib/pyt
Requirement already satisfied: urllib3 in /usr/local/lib/python3.9/dist-packages (fro
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: scikit-learn>=0.22 in /usr/local/lib/python3.9/dist-pa
Requirement already satisfied: setuptools!=50.0.0,>=38.6.0 in /usr/local/lib/python3
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.9/dist-package
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: statsmodels>=0.13.2 in /usr/local/lib/python3.9/dist-p
Requirement already satisfied: pandas>=0.19 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.9/dis
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.9/dist-
Requirement already satisfied: patsy>=0.5.2 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.9/dist-packa
Requirement already satisfied: six in /usr/local/lib/python3.9/dist-packages (from pa
Installing collected packages: pmdarima
Successfully installed pmdarima-2.0.3
```

```
#!pip install geehydro
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/p
Collecting geehydro
  Downloading geehydro-0.2.0.tar.gz (15 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: earthengine-api in /usr/local/lib/python3.9/dist-packa
Requirement already satisfied: folium in /usr/local/lib/python3.9/dist-packages (from
Requirement already satisfied: click in /usr/local/lib/python3.9/dist-packages (from
Requirement already satisfied: google-cloud-storage in /usr/local/lib/python3.9/dist-
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (fr
Requirement already satisfied: httplib2<1dev,>=0.9.2 in /usr/local/lib/python3.9/dist
```

```
Requirement already satisfied: google-auth>=1.4.1 in /usr/local/lib/python3.9/dist-pa
Requirement already satisfied: google-api-python-client>=1.12.1 in /usr/local/lib/pyt
Requirement already satisfied: google-auth-httplib2>=0.0.3 in /usr/local/lib/python3
Requirement already satisfied: numpy in /usr/local/lib/python3.9/dist-packages (from
Requirement already satisfied: branca>=0.6.0 in /usr/local/lib/python3.9/dist-package
Requirement already satisfied: jinja2>=2.9 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: uritemplate<5,>=3.0.1 in /usr/local/lib/python3.9/dist
Requirement already satisfied: google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0d
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.9/dist-packages (
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.9/dist-package
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.9/dis
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.9/dist
Requirement already satisfied: pyparsing!=3.0.0,!=3.0.1,!=3.0.2,!=3.0.3,<4,>=2.4.2 in
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packa
Requirement already satisfied: google-cloud-core<3.0dev,>=2.3.0 in /usr/local/lib/pyt
Requirement already satisfied: google-resumable-media>=2.3.2 in /usr/local/lib/python
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-pa
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: googleapis-common-protos<2.0dev,>=1.56.2 in /usr/local
Requirement already satisfied: protobuf!=3.20.0,!=3.20.1,!=4.21.0,!=4.21.1,!=4.21.2,
Requirement already satisfied: google-crc32c<2.0dev,>=1.0 in /usr/local/lib/python3.9
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.9/dist-
Building wheels for collected packages: geehydro
  Building wheel for geehydro (setup.py) ... done
  Created wheel for geehydro: filename=geehydro-0.2.0-py2.py3-none-any.whl size=10138
  Stored in directory: /root/.cache/pip/wheels/ff/aa/13/b6c5b687208b545f735833f12891f
Successfully built geehydro
Installing collected packages: geehydro
Successfully installed geehydro-0.2.0
```

```
#!pip install geemap
```

```
Requirement already satisfied: argon2-cffi-bindings in /usr/local/lib/python3.9/di
Requirement already satisfied: nbclient>=0.5.0 in /usr/local/lib/python3.9/dist-pa
Requirement already satisfied: bleach in /usr/local/lib/python3.9/dist-packages (f
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.9/dist
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.9/di
Requirement already satisfied: defusedxml in /usr/local/lib/python3.9/dist-package
Requirement already satisfied: jupyterlab-pygments in /usr/local/lib/python3.9/dis
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.9/dist-
Requirement already satisfied: lxml in /usr/local/lib/python3.9/dist-packages (fro
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.9/dist-packages
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.9/dist-pac
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.9/dist-pa
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in /u
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.9/dist-pack
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.9/dist-packag
Requirement already satisfied: webencodings in /usr/local/lib/python3.9/dist-packa
Requirement already satisfied: pycparser in /usr/local/lib/python3.9/dist-packages
Building wheels for collected packages: ee-extra, sankee, pycrs, pyperclip
  Building wheel for ee-extra (setup.py) ... done
  Created wheel for ee-extra: filename=ee_extra-0.0.15-py3-none-any.whl size=23677
  Stored in directory: /root/.cache/pip/wheels/66/66/06/98d6dee3f612d84d2b487fc73c
  Building wheel for sankee (setup.py) ... done
  Created wheel for sankee: filename=sankee-0.2.3-py3-none-any.whl size=30511 sha2
  Stored in directory: /root/.cache/pip/wheels/a0/77/76/04746a9b2af1cfe05fbfb90463
  Building wheel for pycrs (setup.py) ... done
  Created wheel for pycrs: filename=PyCRS-1.0.2-py3-none-any.whl size=32702 sha256
  Stored in directory: /root/.cache/pip/wheels/94/01/24/bc7bff66667ef317615144a15e
  Building wheel for pyperclip (setup.py) ... done
  Created wheel for pyperclip: filename=pyperclip-1.8.2-py3-none-any.whl size=1113
  Stored in directory: /root/.cache/pip/wheels/0c/09/9e/49e21a6840ef7955b06d47394a
Successfully built ee-extra sankee pycrs pyperclip
Installing collected packages: pyperclip, pycrs, logzero, colour, xyzservices, whi
```

```
import io, os, sys, setuptools, tokenize
import statsmodels.api as sm
```

```
from prophet import Prophet
```

```
import ee
ee.Authenticate()
ee.Initialize()
```

```
To authorize access needed by Earth Engine, open the following URL in a web browser a

    https://code.earthengine.google.com/client-auth?scopes=https%3A//www.googleapis.c

The authorization workflow will generate a code, which you should paste in the box be
Enter verification code: 4/1AVHEtk6NaaO5lnqgszg10JelCVLbGSsAkl6SSbS0StGxSAemwmiuJ9qY-

Successfully saved authorization token.
```

```
import ee, datetime    # Google Earth Engine
import pandas as pd
import numpy as np
import folium
import geehydro
from datetime import datetime as dt
from IPython.display import Image
from statsmodels.tsa.seasonal import seasonal_decompose
##from pmdarima.arima import auto_arima
from statsmodels.tsa.arima_model import ARIMA
from sklearn.metrics import mean_squared_error, mean_absolute_error
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import prophet
warnings.filterwarnings('ignore')
```

```
srd_map = folium.Map(location=[17.6075, 78.0798], zoom_start=10)
srd_map
```

```
landsat = ee.ImageCollection("LANDSAT/LC08/C01/T1_SR").\
          filter(ee.Filter.lt('CLOUD_COVER', 20)).\
          filterDate('2013-01-01','2021-01-01')


# setteing coordinates to SANGAREDDY DISTRICT
srd_AOI = ee.Geometry.Rectangle([17.6075, 78.0798,17.9075,78.3798 ])
# filter area
landsat_AOI = landsat.filterBounds(srd_AOI)
```

```
print('Total number of images :', landsat_AOI.size().getInfo())
```

```
Total number of images : 88
```

```
# Plot the 'first' image in the collection

# List of images
listOfImages = landsat_AOI.toList(landsat_AOI.size())

# Plot in RGB color composite
palette = ['red', 'green', 'blue']
parameters = {'min': 0,
              'max': 1000,
              'dimensions': 512,
              'bands': ['B4', 'B3', 'B2'],
              'region': srd_AOI}

srd_map.addLayer(ee.Image(listOfImages.get(1)), parameters)
srd_map
```


```
print('Total number of images :', landsat_AOI.size().getInfo())
```

```
Total number of images : 88
```

```
landsat_AOI.first().bandNames().getInfo()
```

```
['B1',
 'B2',
 'B3',
 'B4',
 'B5',
 'B6',
 'B7',
 'B10',
 'B11',
 'sr_aerosol',
 'pixel_qa',
 'radsat_qa']
```

```
#Calculating NDVI For Sangareddy District
```

```
def addNDVI(image):
    ndvi = image.normalizedDifference(['B5', 'B4']).rename('NDVI')
    return image.addBands(ndvi)

with_ndvi = landsat_AOI.map(addNDVI)
```

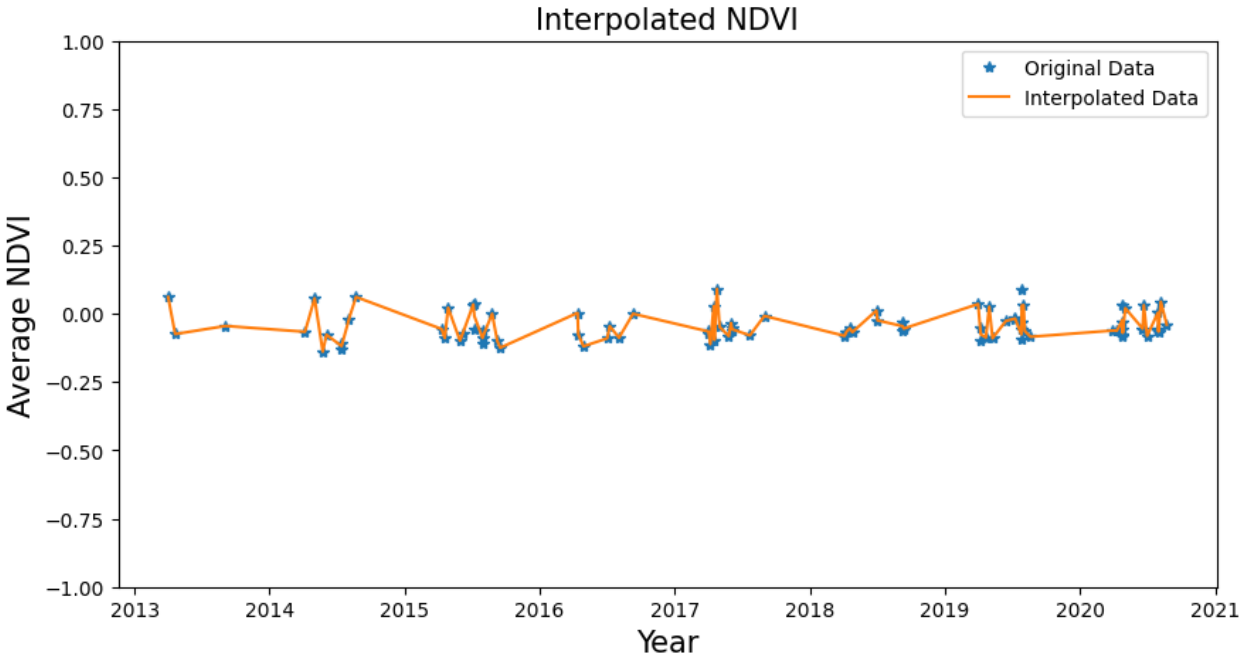
```
def meanNDVI(image):
    image = ee.Image(image)
    meanDict = image.reduceRegion(reducer = ee.Reducer.mean().setOutputs(['NDVI']),
        geometry = srd_AOI,
        scale = image.projection().nominalScale().getInfo(),
                                maxPixels = 100000,
                                bestEffort = True);
    return meanDict.get('NDVI').getInfo()


listOfImages_ndvi = with_ndvi.select('NDVI').toList(with_ndvi.size())

ndvi_coll = []

for i in range(listOfImages_ndvi.length().getInfo()):
    image = ee.Image(listOfImages_ndvi.get(i-1))
    temp_ndvi = meanNDVI(image)
    ndvi_coll.append(temp_ndvi)


dates = np.array(with_ndvi.aggregate_array("system:time_start").getInfo())
day = [datetime.datetime.fromtimestamp(i/1000).strftime('%Y-%m-%d') for i in (dates)]


ndvi_df = pd.DataFrame(ndvi_coll, index = day, columns = ['ndvi'])
ndvi_df.index = pd.to_datetime(ndvi_df.index, format="%Y/%m/%d")
ndvi_df.sort_index(ascending = True, inplace = True)

ndvi_df.head(5)
```

	ndvi
2013-04-02	0.061529
2013-04-21	-0.073949
2013-09-03	-0.044161
2014-04-07	-0.065468
2014-05-03	0.056954

```
ndvi_df_daily = ndvi_df.resample('D').median()

# Linear interpolate NDVI data

ndvi_df_daily.interpolate(method='polynomial', order = 1, inplace = True)
ndvi_df_daily.head(5)
```

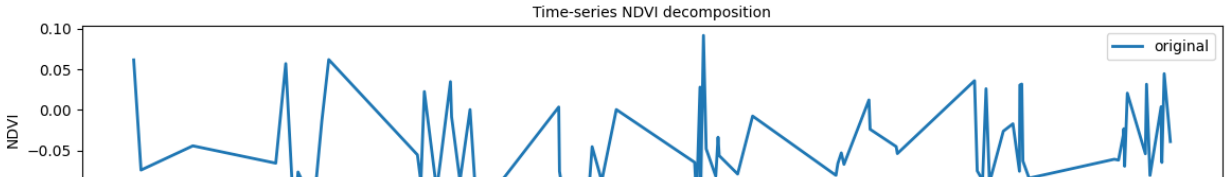
	ndvi
2013-04-02	0.061529

```
plt.figure(figsize=(10,5), dpi=100)
plt.plot(ndvi_df, '*')
plt.plot(ndvi_df_daily)
plt.xlabel('Year', fontsize=15)
plt.ylabel('Average NDVI', fontsize=15)
plt.legend(['Original Data', 'Interpolated Data'])
plt.title("Interpolated NDVI", fontsize=15)
plt.ylim([-1, 1])
plt.show()
```

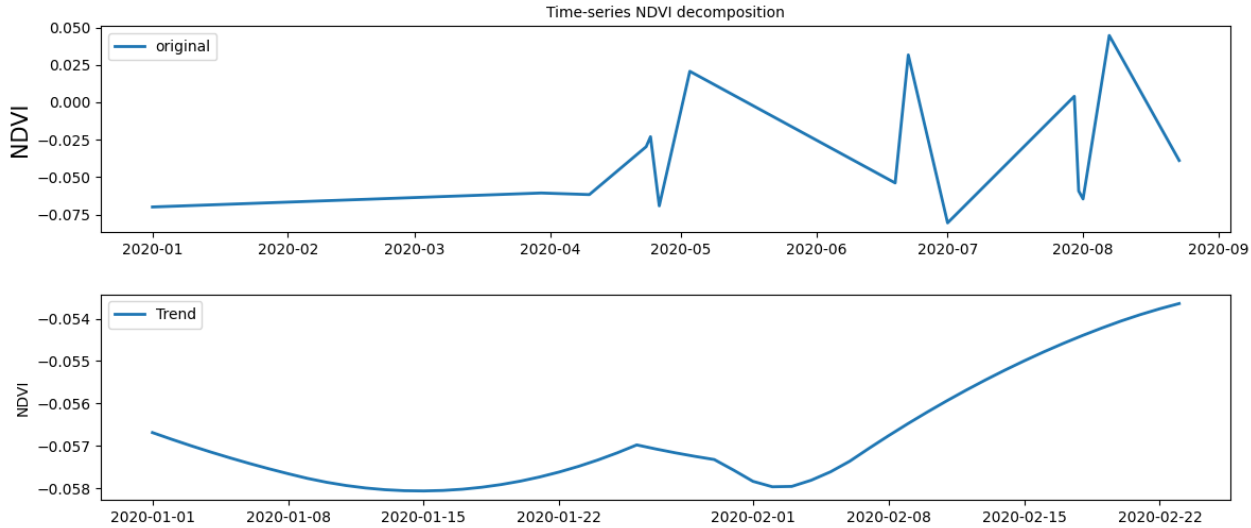



```
decomposition = seasonal_decompose(ndvi_df_daily, model= 'additive', period = 365)    # ad
                                                                                # compared to multiplicative

# assign trend, seasonal components from decomposed data

trend = decomposition.trend
seasonal = decomposition.seasonal


# Plot the original data, the trend, the seasonality, and the residual

plt.figure(figsize=(12,10))
plt.subplot(411)
```

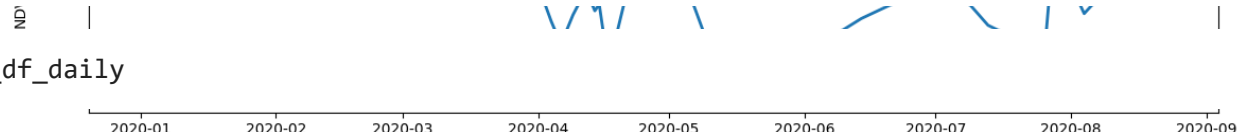
```
plt.plot(ndvi_df_daily, label = 'original', linewidth=2)

plt.legend(loc = 'best', fontsize=10)
plt.ylabel('NDVI', fontsize=10)
plt.title('Time-series NDVI decomposition', fontsize=10)
plt.subplot(412)
plt.plot(trend, label = 'Trend', linewidth=2)

plt.legend(loc = 'best', fontsize=15)
plt.ylabel('NDVI', fontsize=15)
plt.subplot(413)
plt.plot(seasonal, label = 'seasonal', linewidth=2)

plt.legend(loc = 'best', fontsize=10)
plt.ylabel('NDVI', fontsize=10)
plt.xlabel('Year', fontsize=10)
plt.tight_layout()
```




```
two_year = (ndvi_df_daily.index>='2020-01-01') & (ndvi_df_daily.index<='2022-01-01')

plt.figure(figsize=(12,10))
plt.subplot(411)
plt.plot(ndvi_df_daily[two_year], label = 'original', linewidth=2)

plt.legend(loc = 'best', fontsize=10)
plt.ylabel('NDVI', fontsize=15)
plt.title('Time-series NDVI decomposition', fontsize=10)
plt.subplot(412)
plt.plot(trend[two_year], label = 'Trend', linewidth=2)

plt.legend(loc = 'best', fontsize=10)
plt.ylabel('NDVI', fontsize=10)
plt.subplot(413)
plt.plot(seasonal[two_year], label = 'seasonal', linewidth=2)

plt.legend(loc = 'best', fontsize=10)
plt.ylabel('NDVI', fontsize=10)
plt.xlabel('Year', fontsize=10)
plt.tight_layout()
```




```
low_index = day.index('2014-05-25')
high_index = day.index('2017-04-25')
```

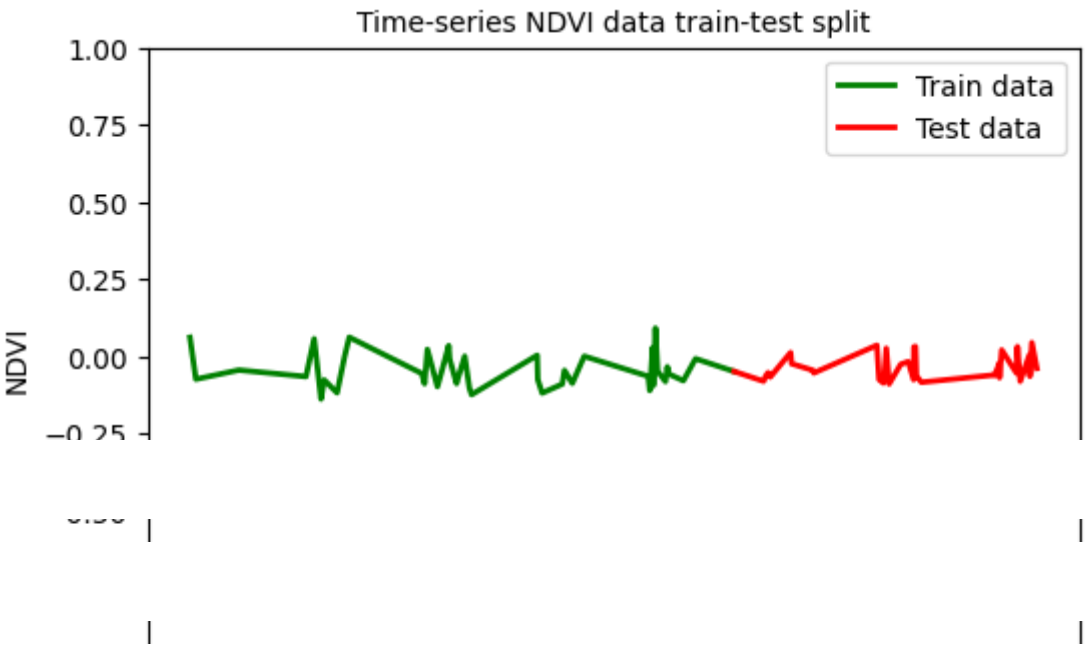


```
#ndvi_df_daily
```



```
ndvi_low = ee.Image(listOfImages_ndvi.get(low_index))
ndvi_high = ee.Image(listOfImages_ndvi.get(high_index))


palette = ['red', 'yellow', 'green']
ndvi_parameters = {'min': -1,
                   'max': 1,
                   'dimensions': 512,
                   'palette': palette,
                   'region': srd_AOI}
srd_map.addLayer(ndvi_low, ndvi_parameters)
srd_map

srd_map.addLayer(ndvi_high, ndvi_parameters)
srd_map
```


```
train_data, test_data = ndvi_df_daily[ndvi_df_daily.index <= '2018-01-01'],\
                        ndvi_df_daily[ndvi_df_daily.index >= '2018-01-01']


# plot the training and testing data

plt.figure(figsize=(6,4))
#plt.grid(True)
plt.xlabel('Year', fontsize=10)
plt.ylabel('NDVI', fontsize=10)
plt.plot(train_data, 'green', label='Train data', linewidth=2)
plt.plot(test_data, 'red', label='Test data', linewidth=2)
plt.title('Time-series NDVI data train-test split', fontsize=10)
plt.ylim([-1,1])
plt.legend(fontsize=10)
plt.show()
```



Prophet Algorithm for NDVI Prediction

```
train_data_fb = train_data.reset_index()
train_data_fb.rename(columns={"index": "ds", "ndvi": "y"},inplace=True)
train_data_fb.head(5)
```

	ds	y
0	2013-04-02	0.061529
1	2013-04-03	0.054399
2	2013-04-04	0.047268
3	2013-04-05	0.040138
4	2013-04-06	0.033007

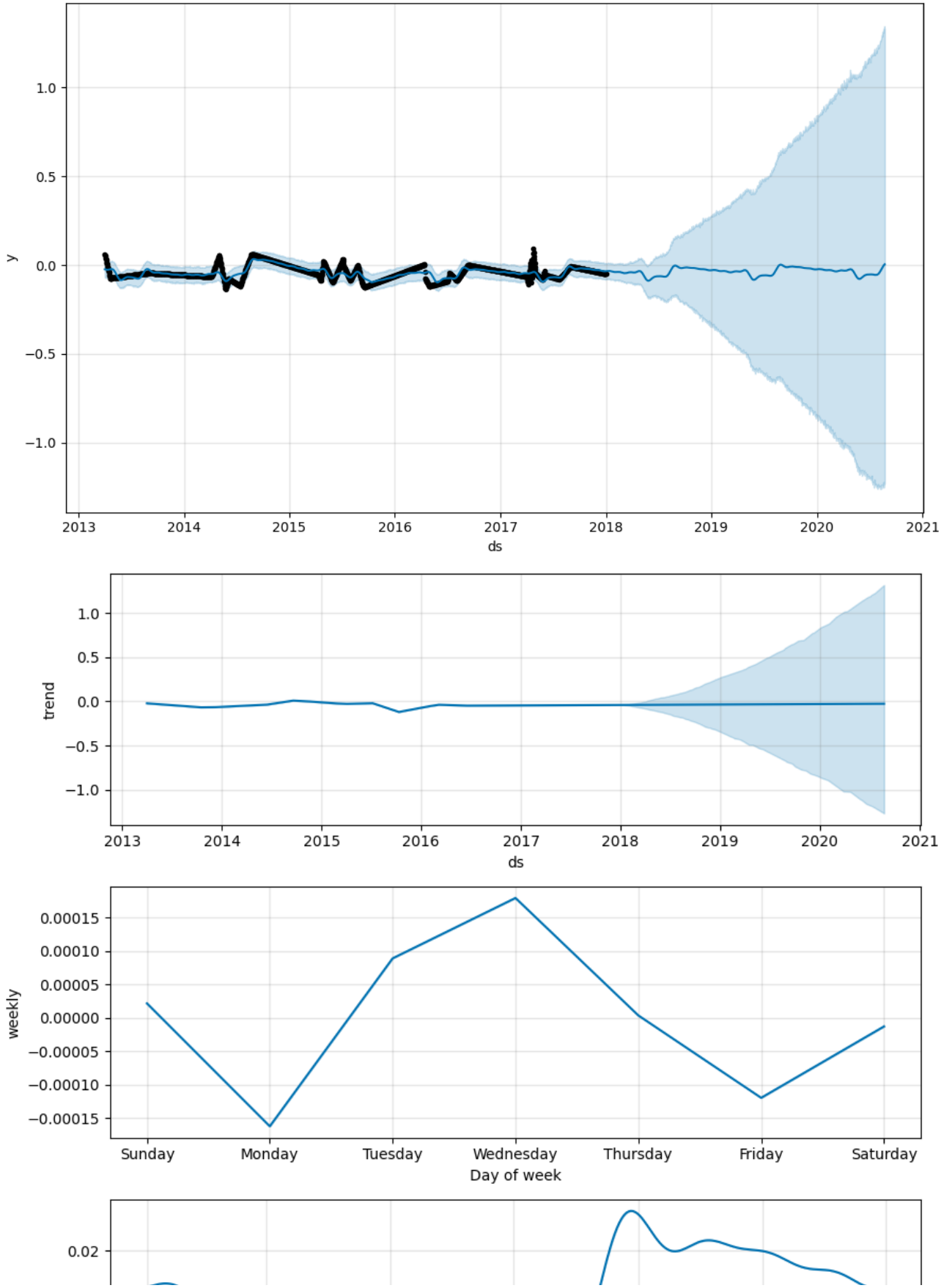
```
m1= Prophet(interval_width=0.95, daily_seasonality=False, # interval_width = confidence in
           changepoint_range=0.7,                         # % of train data to look for cha
                                                   # (default value is 0.8) 0.7 produced better mode
           changepoint_prior_scale=0.3)   # determines trend flexibility. tuned around the
                                           # 3 produced best performance

m1.fit(train_data_fb)

# number of days to forecast, based on test_data

forecast_days = (test_data.index[-1]-test_data.index[0]).days

# Create dataframe with the dates we want to predict

future = m1.make_future_dataframe(periods = forecast_days, freq = 'D')
```

```
# Predict the price

forecast = m1.predict(future)
```

```
DEBUG:cmdstanpy:input tempfile: /tmp/tmpc1ygqrzp/0vpiyo5p.json
DEBUG:cmdstanpy:input tempfile: /tmp/tmpc1ygqrzp/z0cqrbid.json
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.9/dist-packages/prophet/stan_m
06:41:55 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
06:41:56 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
```

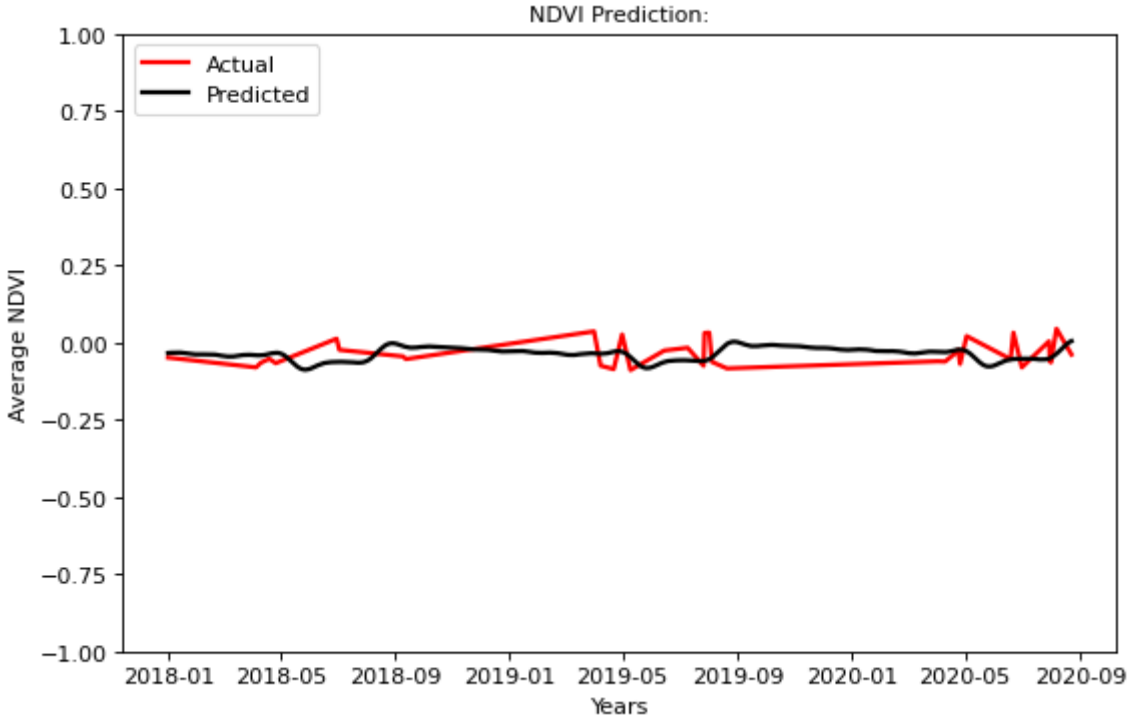
```
plt.figure(figsize=(6,3), dpi=80)

fig = m1.plot(forecast)
fig = m1.plot_components(forecast)


plt.show()
```

```
<Figure size 480x240 with 0 Axes>
```




```
fc_test = forecast[forecast.ds.isin(test_data.index)]

# take forecast data for 2018-2020

plt.figure(figsize=(8,5), dpi=80)
ax=plt.plot(test_data, color = 'red', label='Actual', linewidth=2)
```

```
plt.plot(fc_test.ds,fc_test.yhat, color = 'black',label='Predicted', linewidth=2)
plt.title('NDVI Prediction:', fontsize=10)
plt.xlabel('Years', fontsize=10)
plt.ylabel('Average NDVI', fontsize=10)
plt.ylim((-1,1))
plt.legend(loc='upper left', fontsize=10)


# save the plot
plt.show()
```




```
def performance_measure(model, yhat, y):
    # mean squared error
    mse = mean_squared_error(y, yhat)
    #mean absolute error
    mae = mean_absolute_error(y, yhat)
    # root mean squared error
    rmse=np.sqrt(mse)
    #average score
    average=np.mean((mse, mae, rmse))
    # save model performance as dataframe
    metrics=pd.DataFrame({'model': model, 'mse': [mse], 'mae': [mae], 'rmse': [rmse], 'ave
    return metrics


FBProphet = performance_measure('Prophet', fc_test.yhat.values.flatten(), test_data.values
FBProphet
```

	model	mse	mae	rmse	average_score
0	Prophet	0.001827	0.037646	0.042743	0.027405

```
ndvi_df_daily.to_csv('ndvi.csv')
```

```
from google.colab import drive
drive.mount('/drive')
path = '/drive/My Drive/ndvi.csv'
```

```
Drive already mounted at /drive; to attempt to forcibly remount, call drive.mount("/c
```

```
ndvi_df_daily
```

1 to 25 of 2701 entries Filter

index	ndvi
2013-04-02 00:00:00	0.06152915108971359
2013-04-03 00:00:00	0.05439871939818949
2013-04-04 00:00:00	0.047268287706665385
2013-04-05 00:00:00	0.04013785601514129
2013-04-06 00:00:00	0.03300742432361719
2013-04-07 00:00:00	0.025876992632093088
2013-04-08 00:00:00	0.018746560940568986
2013-04-09 00:00:00	0.011616129249044892
2013-04-10 00:00:00	0.004485697557520787
2013-04-11 00:00:00	-0.002644734134003311
2013-04-12 00:00:00	-0.009775165825527423
2013-04-13 00:00:00	-0.016905597517051514
2013-04-14 00:00:00	-0.02403602920857562
2013-04-15 00:00:00	-0.031166460900099717
2013-04-16 00:00:00	-0.03829689259162382
2013-04-17 00:00:00	-0.04542732428314791
2013-04-18 00:00:00	-0.05255775597467202
2013-04-19 00:00:00	-0.05968818766619612
2013-04-20 00:00:00	-0.06681861935772022
2013-04-21 00:00:00	-0.07394905104924432
2013-04-22 00:00:00	-0.07372839967265328
2013-04-23 00:00:00	-0.07350774829606224
2013-04-24 00:00:00	-0.0732870969194712
2013-04-25 00:00:00	-0.07306644554288017
2013-04-26 00:00:00	-0.07284579416628913

Show 25 per page 1 2 10 100 109

Like what you see? Visit the data table notebook to learn more about interactive tables.

```
"""from google.colab import files
ndvi_df_daily.to_csv('ndvi_df_daily.csv')
files.download('ndvi_df_daily.csv')"""
```

Land Surface Temperature and Growing Degree Days ANALYSIS

```
from __future__ import print_function
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn import tree
import warnings
warnings.filterwarnings('ignore')
```

GEE SCRIPT

var dataset = var L8 = ee.ImageCollection("LANDSAT/LC08/C02/T2_L2")

.filterBounds(ROI)

.filterDate('2013-01-01','2018-01-01')

.filterMetadata('CLOUD_COVER','less_than',1)

.mean()

.clip(ROI);

var indiaBorder = dataset.filter(ee.Filter.eq('country_na', 'India')); print(indiaBorder);

Map.centerObject(indiaBorder, 6); Map.addLayer(indiaBorder);

Importing LST image collection.

var modis = ee.ImageCollection('LANDSAT/LC08/CO2/T2_L2');

var start = ee.Date('2013-01-01');

var dateRange = ee.DateRange(start, start.advance(5, 'year'));

var mod11a2 = modis.filterDate(dateRange);

// Select only the 1km day LST data band. var modLSTday = mod11a2.select('LST_Day_1km','LST_Night_1km');

var inCelsius = modLSTday.map(function(img) {

return img

```
  .multiply(0.02)
  .subtract(273.15)
  .clip(geometry)
  .copyProperties(img, ['system:time_start']);
```

});

Chart time series of LST for India 2017.

var ts1 = ui.Chart.image.series({

imageCollection: inCelsius,

region: indiaBorder,

reducer: ee.Reducer.median(),

scale: 1000,

xProperty: 'system:time_start'})

.setOptions({

```
  title: 'LST 2013 Time Series',
  vAxis: {title: 'LST Celsius'}});
```

print(ts1);

Calculating 8-day Mean Temp

var clippedDay=inCelsius.select('LST_Day_1km').median().clip(indiaBorder);

var clippedNight=inCelsius.select('LST_Night_1km').median().clip(indiaBorder);

Map.addLayer(clippedDay,{

min:3,max:30,

palette:['blue','limegreen','yellow','darkorange','red']},

'Mean Day Temperature,2013');

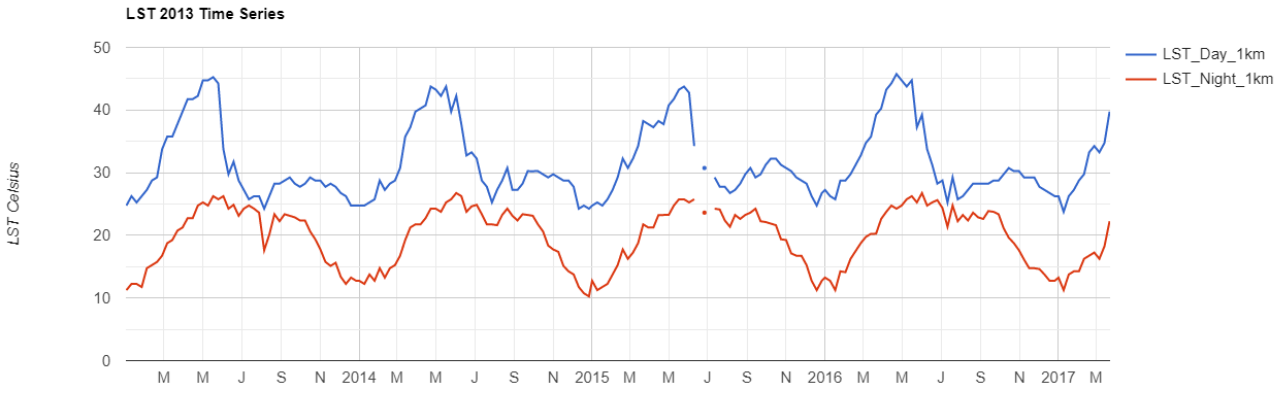
Map.addLayer(clippedNight,{

min:3,max:30,

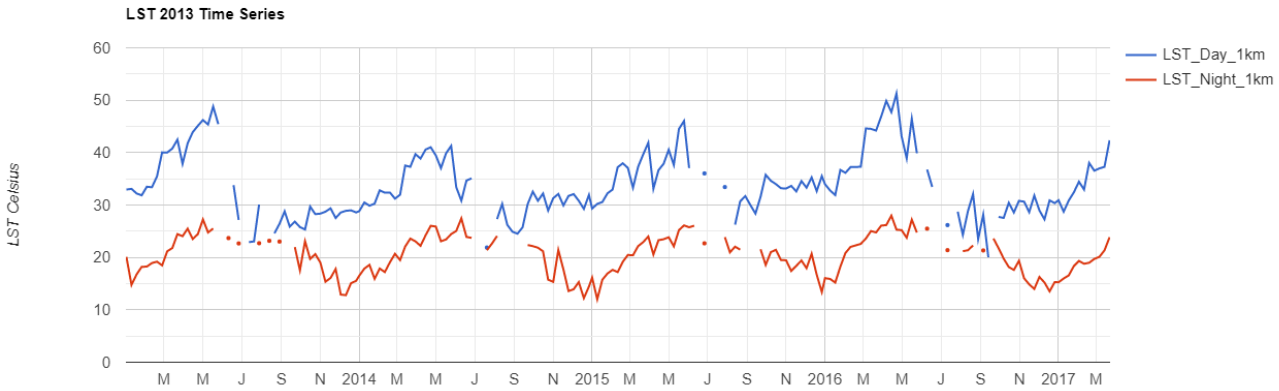
palette:['blue','limegreen','yellow','darkorange','red']},

'Mean Day Temperature,2013');

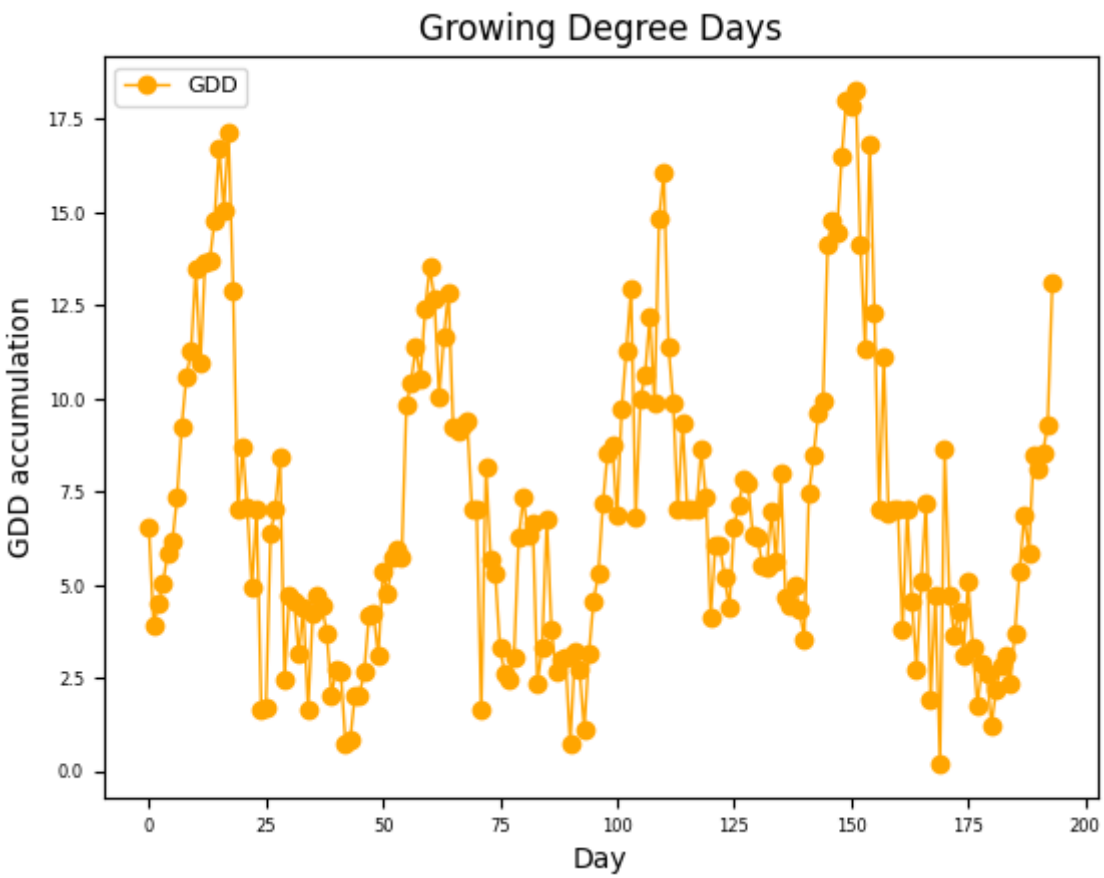
LST- INDIA (2013-2017)



LST-SANGAREDDY DISTRICT (2013-2017)



GDD Result



Double-click (or enter) to edit

✓ 0s completed at 1:07 PM