

prophet model from facebook to be used in time series analysis

importing libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from scipy.stats import pearsonr
```

import the dataset

```
# df = pd.read_csv("./datasets/DailyDelhiClimateTrain.csv",
index_col="date", parse_dates=True)
df = pd.read_csv("./datasets/DailyDelhiClimateTrain.csv")
```

getting general information

```
print(df.head())
print(df.info())
print(df.describe())
```

	date	meantemp	humidity	wind_speed	meanpressure
0	2013-01-01	10.000000	84.500000	0.000000	1015.666667
1	2013-01-02	7.400000	92.000000	2.980000	1017.800000
2	2013-01-03	7.166667	87.000000	4.633333	1018.666667
3	2013-01-04	8.666667	71.333333	1.233333	1017.166667
4	2013-01-05	6.000000	86.833333	3.700000	1016.500000

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 1462 entries, 0 to 1461
```

```
Data columns (total 5 columns):
```

#	Column	Non-Null Count	Dtype
0	date	1462 non-null	object
1	meantemp	1462 non-null	float64
2	humidity	1462 non-null	float64
3	wind_speed	1462 non-null	float64
4	meanpressure	1462 non-null	float64

```
dtypes: float64(4), object(1)
```

```
memory usage: 57.2+ KB
```

```
None
```

	meantemp	humidity	wind_speed	meanpressure
count	1462.000000	1462.000000	1462.000000	1462.000000
mean	25.495521	60.771702	6.802209	1011.104548
std	7.348103	16.769652	4.561602	180.231668
min	6.000000	13.428571	0.000000	-3.041667
25%	18.857143	50.375000	3.475000	1001.580357
50%	27.714286	62.625000	6.221667	1008.563492
75%	31.305804	72.218750	9.238235	1014.944901
max	38.714286	100.000000	42.220000	7679.333333

```

# line plot
sns.set(rc={'figure.figsize':(11,4)}, font_scale = 0.9,
style='darkgrid')
sns.lineplot(data=df, x="date", y = "meantemp")
sns.lineplot(data=df, x="date", y = "humidity")
sns.lineplot(data=df, x="date", y = "wind_speed")
sns.lineplot(data=df, x="date", y = "meanpressure")

<AxesSubplot:xlabel='date', ylabel='meantemp'>

# same plot on plotly
fig = px.line(df,x=df.index, y="meantemp", title="Mean Temperature")
fig.show()

# meanpressure on plotly
fig = px.line(df,x=df.index, y="meanpressure", title="Mean Pressure")
fig.show()

# interpolate the outliers from mean pressure column
df.loc['2016-03-28', 'meanpressure'] = np.nan
df['meanpressure'] = df['meanpressure'].interpolate(method='time')
df.loc['2016-03-28']

# plotting after interpolation
# meanpressure on plotly
fig = px.line(df,x=df.index, y="meanpressure", title="Mean Pressure")
fig.show()

# scatter plot between meantemp and humidity
fig = px.scatter(df, x="meantemp", y="humidity", title="Mean Temp vs
Humidity",trendline="ols", trendline_color_override="red")
fig.show()

corr,_ =pearsonr(df['meantemp'],df['humidity'])
print('Pearsons correlation: %.3f' % corr)

sns.pairplot(df, kind= "reg")

# heatmap of correlation
sns.heatmap(df.corr(), annot = True, cmap = "coolwarm")

# scaling the data
from sklearn.preprocessing import MinMaxScaler, StandardScaler
scaler = StandardScaler()
df_scaled = pd.DataFrame(scaler.fit_transform(df), columns =
df.columns, index = df.index)
df_scaled.describe()

# drawing line plot on scaled data
sns.set(rc={'figure.figsize':(11,4)}, font_scale = 0.8,
style='darkgrid')
sns.lineplot(data=df_scaled, x="date", y = "meantemp")

```

```

sns.lineplot(data=df_scaled, x="date", y = "humidity")
sns.lineplot(data=df_scaled, x="date", y = "wind_speed")
# sns.lineplot(data=df_scaled, x="date", y = "meanpressure")

# added two columns in scaled dataset for month and year
df_scaled['month'] = df_scaled.index.month
df_scaled['year'] = df_scaled.index.year
df_scaled.head()

sns.lineplot(data=df_scaled, x="month", y="meantemp", hue="year",
palette="deep")

# same in plotly
fig = px.line(df_scaled, x = df_scaled.index, y="meantemp", color =
"year", title = "Mean Temperature by year")
fig.show()

# added two columns in dataset for month and year
df['month'] = df.index.month
df['year'] = df.index.year
df.head()

# same in plotly
fig = px.line(df, x = df.index, y="meantemp", color = "year", title =
"Mean Temperature by year")
fig.show()

```

using the prophet model

```

df.head() # bringing date back as a column from index

# renaming the columns
df = df.rename(columns={'date':'ds', 'meantemp': 'y'})
df.head()

from prophet import Prophet
from prophet.plot import plot_plotly, plot_components_plotly

# calling the model
model = Prophet()
model.fit(df)
future = model.make_future_dataframe(periods=365)
forecast = model.predict(future)

plot_plotly(model, forecast)

forecast.head()

```