

Time Series to observe DAILY temperature variations

Daily temperature prediction using Facebook Prophet

For this analysis, to complement our analysis of models we turned to using an open-sourced library by Facebook called Prophet. <https://facebook.github.io/prophet/>
(<https://facebook.github.io/prophet/>)

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

Prophet is open source software released by Facebook's Core Data Science team.

We have take the original series as provided and using their documentation repeated our forecasting analysis.

```
In [3]: import pandas as pd
        from fbprophet import Prophet
```

```
In [4]: import seaborn as sns
        import matplotlib as mpl
        import matplotlib.pyplot as plt

        mpl.style.use('seaborn')

        cities_df = pd.read_csv("../data/temperature.csv", parse_dates=['datetime'])
        cities_df.head()

        temp_df = cities_df.fillna(method = 'bfill', axis=0).dropna()
        temp_df = temp_df.rename(columns={'Los Angeles': 'y', 'datetime': 'ds'})
        temp_df['y'] = temp_df['y'] - 273.15
```

```
In [7]: m = Prophet(changepoint_prior_scale=0.01)
        m.fit(temp_df.loc[:,["ds","y"]])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pystan\misc.py:399: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
    elif np.issubdtype(np.asarray(v).dtype, float):
```

```
Out[7]: <fbprophet.forecaster.Prophet at 0x129cfbef0>
```

```
In [8]: future = m.make_future_dataframe(periods=60, freq='D')

forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
fig1 = m.plot(forecast)
fig2 = m.plot_components(forecast)
```





