

```

#author: 7_hermit
#mail: uzumaki.sahoo@gmail.com
#dependencies: numpy, matplotlib and Scipy
import matplotlib.pyplot as plt
import numpy as np
from scipy.interpolate import make_interp_spline, BSpline

data = {
    'Line Current':[6, 6.3, 6.6, 6.9, 7.2, 7.5],
    'Output Power':[410.46, 451.65, 577.07, 658.69, 698.18, 726.01],
    'Torque':[2.67, 2.95, 3.79, 4.35, 4.63, 4.91],
    'Slip': [0.021, 0.025, 0.031, 0.036, 0.040, 0.059],
    'Power Factor':[0.5, 0.54, 0.62, 0.67, 0.68, 0.69],
    'Efficiency':[62.19, 59.43, 64.11, 65.8, 64.64, 63.68],
}

def OPvsTELPs():
    x = np.array(data['Output Power'])
    torque = np.array(data['Torque'])
    efficiency = np.array(data['Efficiency'])
    line_current = np.array(data['Line Current'])
    slip = np.array([x*100 for x in data['Slip']])
    power_factor = np.array(data['Power Factor'])

    x_smooth = np.linspace(x.min(), x.max(), 100)

    spl = make_interp_spline(x, torque, k = 3)
    torque_smooth = spl(x_smooth)

    plt.plot(x_smooth, torque_smooth, 'm', x, torque, 'mo', label = 'Torque')
    plt.xlabel('Output Power->')
    plt.legend(loc = 'upper left')
    plt.show()

    spl = make_interp_spline(x, efficiency, k = 3)
    efficiency_smooth = spl(x_smooth)

    plt.plot(x_smooth, efficiency_smooth, 'b', x, efficiency, 'bo', label = 'Efficiency')
    z = np.where(efficiency_smooth == efficiency_smooth.max())
    r = 'Power at Maximum efficiency(' + str(efficiency_smooth.max()[0])[5:] + '%' + ' is ' + str(x_smooth[z[0]])[1:7] + 'W'
    plt.plot(x_smooth[z[0]], efficiency_smooth.max(), 'ro', label = r)
    label = 'Output Power->\nPower Output of ' + str(x_smooth[z[0]])[1:7] + 'W is found at Maximum efficiency of ' + str(efficiency_smooth.max()[0])[5:] + '%.'
    plt.xlabel(label)
    plt.legend(loc = 'upper left')
    plt.show()

    spl = make_interp_spline(x, line_current, k = 3)
    line_current_smooth = spl(x_smooth)

    plt.plot(x_smooth, line_current_smooth, 'k', x, line_current, 'ko', label = 'Line Current')
    plt.xlabel('Output Power->')

```

```
plt.legend(loc = 'upper left')
plt.show()

spl = make_interp_spline(x, power_factor, k = 3)
power_factor_smooth = spl(x_smooth)

plt.plot(x_smooth, power_factor_smooth, 'g', x, power_factor, 'go', label = 'Power Factor')
plt.xlabel('Output Power->')
plt.legend(loc = 'upper left')
plt.show()

spl = make_interp_spline(x, slip, k = 3)
slip_smooth = spl(x_smooth)

plt.plot(x_smooth, slip_smooth, 'r', x, slip, 'ro', label = 'Slip%')
plt.xlabel('Output Power->')
plt.legend(loc = 'upper left')
plt.show()

def TorqueVsSlip():
    x = np.array(data['Slip'])
    y = np.array(data['Torque'])

    x_smooth = np.linspace(x.min(), x.max(), 100)
    spl = make_interp_spline(x, y, k = 3)
    y_smooth = spl(x_smooth)

    plt.plot(x, y, 'mo', label = 'Actual Points')
    plt.plot(x_smooth, y_smooth, 'b-.', label = 'Interpolated Curve');

    plt.legend(loc = 'upper left')
    plt.ylabel('Torque(N-m)->')
    plt.title("Torque Vs Slip.")
    plt.xlabel('Slip->')
    plt.show()

if __name__ == '__main__':
    TorqueVsSlip()
    OPvsTELPs()
```