Chapter 4 - Polynomial Regression

March 30, 2020

1 Polynomial Regression

1.1 Three basic regression Model

- Simple Linear Regression : $y = b_0 + b_1 x$
- Multivariate Linear Regression : $y = \sum_{i=0}^{n} b_i x_i$
- Polynomial Regression : $y = \sum_{i=0}^{n} b_i x^i$

1.1.1 When to use polynomian Regression

1. Dataset looks like a non-linear association
2. It is still called Polynomial **Linear** regression, as the coefficients are linear. the funtion can be expressed as a linear combination of co-efficients.
3. This is the simplest form of linear regression model, more complex alternatives are SVR, DT & RF

2 Dataset

source : https://www.superdatascience.com/machine-learning

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

[7]:
```
dataset = pd.read_csv('ds/Position_Salaries.csv')
dataset
```

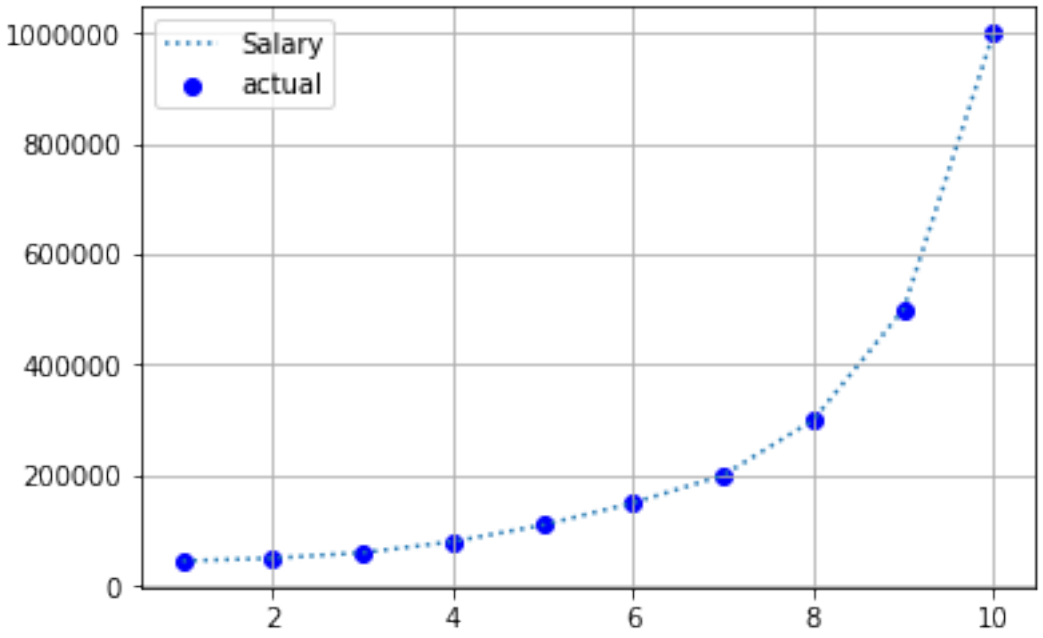
[7]:

	Position	Level	Salary
0	Business Analyst	1	45000
1	Junior Consultant	2	50000
2	Senior Consultant	3	60000
3	Manager	4	80000
4	Country Manager	5	110000
5	Region Manager	6	150000

```
6            Partner      7   200000
7     Senior Partner      8   300000
8            C-level      9   500000
9                CEO     10  1000000
```

[96]:
```
def plot_me(x,y,col,lab,style):
    plt.xlabel='Level'
    plt.ylabel='Salary'
    plt.title='Level - Salary '
    plt.scatter(x, y, color=col, label=lab)
    plt.plot(x, y, style)
    plt.grid(True)
    plt.legend()
```

[97]:
```
plot_me(dataset['Level'],dataset['Salary'],'blue','actual',':')
plt.show()
```



2.1 Dataset Details

2.1.1 size

[5]:
```
dataset.shape
```

[5]: (10, 3)

[ ]:

2.2 Data Preprocessing

[20]:
```
# separating dependent and independent variables
X = dataset.iloc[:,1:2].values  # col 1 is level
                                # 1:2 is to make the output as matrix
Y = dataset.iloc[:,-1].values   # last col is salary
```

3 Perform Regression

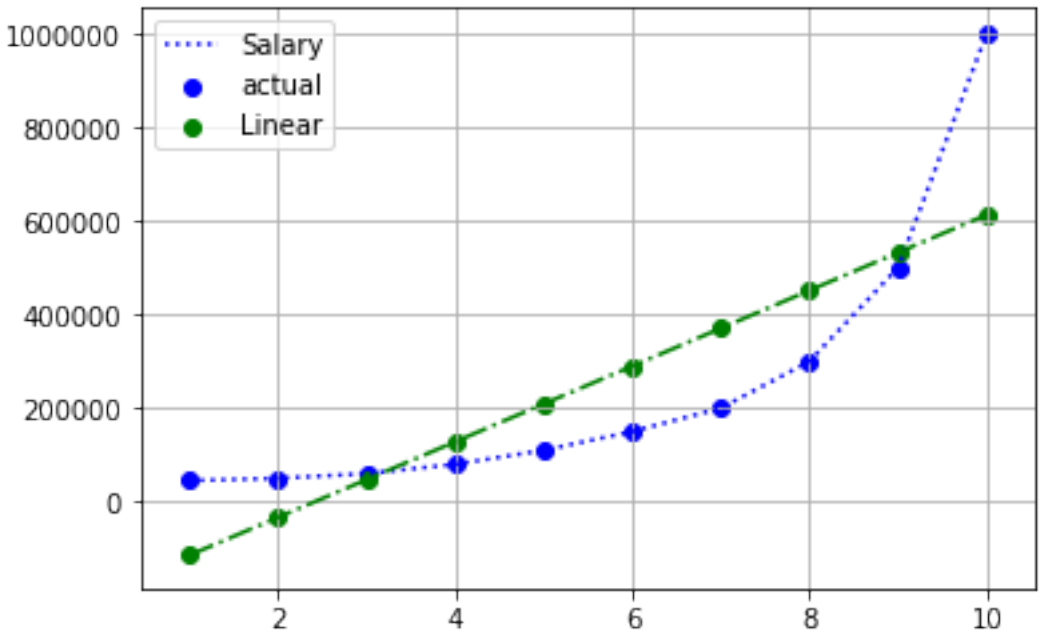
3.1 Linear regression (Just for ref.)

[30]:
```
from sklearn.linear_model import LinearRegression

Lin_reg = LinearRegression()
Lin_reg.fit(X,Y)
```

[30]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

[101]:
```
plot_me(dataset['Level'],dataset['Salary'],'blue','actual','b:')
plot_me(X,Lin_reg.predict(X),'green','Linear','g-.')
plt.show()
```



3.2 Polynomial Regression

[116]:
```
from sklearn.preprocessing import PolynomialFeatures

poly_reg = PolynomialFeatures(degree=3)
X_poly = poly_reg.fit_transform(X)
Lin_reg2 = LinearRegression()
Lin_reg2.fit(X_poly, Y)
```

[116]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

[119]:
```
plot_me(dataset['Level'],dataset['Salary'],'blue','actual','b:')
plot_me(X,Lin_reg.predict(X),'green','linear','g:')
plot_me(X,Lin_reg2.predict(poly_reg.fit_transform(X)),'red','Polynomial','r-.')
plt.show()
```