* **Salary\_hike -> Build a prediction model for Salary\_hike**

0 3155.841248

1 8127.848783

2 -2236.143681

3 -1167.124842

4 -6691.117306

5 3444.909069

6 6007.912837

7 -1587.079627

8 8412.920373

9 -3568.060788

10 570.946748

11 -7798.049484

12 -6635.049484

13 -7456.045717

14 -7206.030645

15 -4159.015574

16 -7958.008038

17 7210.999498

18 -183.977895

19 11448.025873

20 1686.056015

21 5386.067319

22 855.097462

23 10530.108765

24 1424.127605

25 -5259.861092

26 1402.157748

27 -3876.838485

28 -735.812110

29 -3144.804574

dtype: float64

plt.scatter(sal\_hike.YearsExperience,sal\_hike.Salary,c='r');plt.plot(sal\_hike.YearsExperience,lin\_predict,c='b');plt.xlabel('years of experience');plt.ylabel('salary')

Out[16]: Text(0, 0.5, 'salary')

lin\_model.conf\_int()

Out[17]:

0 1

Intercept 21136.061314 30448.339084

sal\_hike.YearsExperience 8674.118747 10225.805896

np.corrcoef(sal\_hike.Salary,lin\_predict)

Out[18]:

array([[1. , 0.97824162],

[0.97824162, 1. ]])

#r=0.978

from sklearn.metrics import mean\_squared\_error

from math import sqrt

lin\_rmse=sqrt(mean\_squared\_error(sal\_hike.Salary,lin\_predict))

lin\_rmse

Out[23]: 5592.043608760662

#rmse=5592, R\_sq=0.957

###################Log Model##################

import statsmodels.formula.api as smf

log\_model=smf.ols('sal\_hike.Salary~np.log(sal\_hike.YearsExperience)',data=sal\_hike).fit()

log\_model.params

Out[28]:

Intercept 14927.97177

np.log(sal\_hike.YearsExperience) 40581.98796

dtype: float64

log\_model.summary()

Out[29]:

<class 'statsmodels.iolib.summary.Summary'>

"""

OLS Regression Results

==============================================================================

Dep. Variable: sal\_hike.Salary R-squared: 0.854

Model: OLS Adj. R-squared: 0.849

Method: Least Squares F-statistic: 163.6

Date: Sun, 01 Nov 2020 Prob (F-statistic): 3.25e-13

Time: 12:42:46 Log-Likelihood: -319.77

No. Observations: 30 AIC: 643.5

Df Residuals: 28 BIC: 646.3

Df Model: 1

Covariance Type: nonrobust

====================================================================================================

coef std err t P>|t| [0.025 0.975]

----------------------------------------------------------------------------------------------------

Intercept 1.493e+04 5156.226 2.895 0.007 4365.921 2.55e+04

np.log(sal\_hike.YearsExperience) 4.058e+04 3172.453 12.792 0.000 3.41e+04 4.71e+04

==============================================================================

Omnibus: 1.094 Durbin-Watson: 0.512

Prob(Omnibus): 0.579 Jarque-Bera (JB): 0.908

Skew: 0.156 Prob(JB): 0.635

Kurtosis: 2.207 Cond. No. 5.76

==============================================================================

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

"""

#will check for p-value<0.05 and high R-squared value(0.854) to be a good model,if not go for transformation

log\_predict=log\_model.predict(sal\_hike)

log\_predict

Out[32]:

0 18795.848339

1 25575.235192

2 31382.551905

3 43057.262306

4 46925.138875

5 58136.050079

6 59511.842441

7 62130.943929

8 62130.943929

9 68022.718504

10 70159.105863

11 71186.552842

12 71186.552842

13 72188.628149

14 75966.422577

15 79422.295729

16 81045.791737

17 82606.829882

18 86959.066704

19 87641.132977

20 92720.502137

21 94472.514696

22 98805.371390

23 100317.918684

24 102719.920751

25 104095.713112

26 106289.868435

27 106714.814600

28 109571.007247

29 110351.454145

dtype: float64

log\_Error=sal\_hike.Salary-log\_predict

log\_Error

Out[34]:

0 20547.151661

1 20629.764808

2 6348.448095

3 467.737694

4 -7034.138875

5 -1494.050079

6 638.157559

7 -7685.943929

8 2314.056071

9 -10833.718504

10 -6941.105863

11 -15392.552842

12 -14229.552842

13 -15107.628149

14 -14855.422577

15 -11484.295729

16 -15016.791737

17 481.170118

18 -5596.066704

19 6298.867023

20 -982.502137

21 3800.485304

22 2496.628610

23 13494.081316

24 6711.079249

25 1486.286888

26 10679.131565

27 5920.185400

28 12819.992753

29 11520.545855

dtype: float64

plt.scatter(sal\_hike.YearsExperience,sal\_hike.Salary,c='r');plt.plot(sal\_hike.YearsExperience,log\_predict,c='b');plt.xlabel('years of experience');plt.ylabel('salary')

Out[35]: Text(0, 0.5, 'salary')

log\_model.conf\_int()

Out[36]:

0 1

Intercept 4365.921305 25490.022235

np.log(sal\_hike.YearsExperience) 34083.511736 47080.464184

np.corrcoef(sal\_hike.Salary,log\_predict)

Out[37]:

array([[1. , 0.92406108],

[0.92406108, 1. ]])

#r=0.924

from sklearn.metrics import mean\_squared\_error

from math import sqrt

log\_rmse=sqrt(mean\_squared\_error(sal\_hike.Salary,log\_predict))

log\_rmse

Out[42]: 10302.893706228304

#rmse=10302.89, R\_sq=0.854

###################Exponential Model##################

import statsmodels.formula.api as smf

Exp\_model=smf.ols('np.log(sal\_hike.Salary)~sal\_hike.YearsExperience',data=sal\_hike).fit()

Exp\_model.params

Out[47]:

Intercept 10.507402

sal\_hike.YearsExperience 0.125453

dtype: float64

Exp\_model.summary()

Out[48]:

<class 'statsmodels.iolib.summary.Summary'>

"""

OLS Regression Results

===================================================================================

Dep. Variable: np.log(sal\_hike.Salary) R-squared: 0.932

Model: OLS Adj. R-squared: 0.930

Method: Least Squares F-statistic: 383.6

Date: Sun, 01 Nov 2020 Prob (F-statistic): 7.03e-18

Time: 12:43:35 Log-Likelihood: 28.183

No. Observations: 30 AIC: -52.37

Df Residuals: 28 BIC: -49.56

Df Model: 1

Covariance Type: nonrobust

============================================================================================

coef std err t P>|t| [0.025 0.975]

--------------------------------------------------------------------------------------------

Intercept 10.5074 0.038 273.327 0.000 10.429 10.586

sal\_hike.YearsExperience 0.1255 0.006 19.585 0.000 0.112 0.139

==============================================================================

Omnibus: 0.826 Durbin-Watson: 1.438

Prob(Omnibus): 0.661 Jarque-Bera (JB): 0.812

Skew: 0.187 Prob(JB): 0.666

Kurtosis: 2.286 Cond. No. 13.2

==============================================================================

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

"""

#will check for p-value<0.05 and high R-squared value(0.932) to be a good model,if not go for transformation

pred=Exp\_model.predict(sal\_hike)

Exp\_predict=np.exp(pred)

Exp\_predict

Out[52]:

0 41998.957468

1 43066.066794

2 44160.289228

3 47019.029189

4 48213.688474

5 52639.142472

6 53303.675386

7 54658.014935

8 54658.014935

9 58196.330788

10 59674.982903

11 60428.338456

12 60428.338456

13 61191.204604

14 64340.199291

15 67651.246149

16 69370.128712

17 71132.684635

18 76693.630780

19 77661.835040

20 85860.704208

21 89153.725393

22 98565.809591

23 102346.110512

24 108971.540758

25 113150.933364

26 120475.819601

27 121996.743836

28 133194.621344

29 136578.829694

dtype: float64

Exp\_Error=sal\_hike.Salary-Exp\_predict

Exp\_Error

Out[54]:

0 -2655.957468

1 3138.933206

2 -6429.289228

3 -3494.029189

4 -8322.688474

5 4002.857528

6 6846.324614

7 -213.014935

8 9786.985065

9 -1007.330788

10 3543.017097

11 -4634.338456

12 -3471.338456

13 -4110.204604

14 -3229.199291

15 286.753851

16 -3341.128712

17 11955.315365

18 4669.369220

19 16278.164960

20 5877.295792

21 9119.274607

22 2736.190409

23 11465.889488

24 459.459242

25 -7568.933364

26 -3506.819601

27 -9361.743836

28 -10803.621344

29 -14706.829694

dtype: float64

plt.scatter(sal\_hike.YearsExperience,sal\_hike.Salary,c='r');plt.plot(sal\_hike.YearsExperience,Exp\_predict,c='b');plt.xlabel('years of experience');plt.ylabel('salary')

Out[55]: Text(0, 0.5, 'salary')

Exp\_model.conf\_int()

Out[56]:

0 1

Intercept 10.428656 10.586148

sal\_hike.YearsExperience 0.112332 0.138574

np.corrcoef(sal\_hike.Salary,Exp\_predict)

Out[57]:

array([[1. , 0.96604697],

[0.96604697, 1. ]])

#r=0.966

from sklearn.metrics import mean\_squared\_error

from math import sqrt

Exp\_rmse=sqrt(mean\_squared\_error(sal\_hike.Salary,Exp\_predict))

Exp\_rmse

Out[62]: 7213.2350766202335

#rmse=7213.23, R\_sq=0.932

###################Quad Model##################

import statsmodels.formula.api as smf

#sal\_hike['sq\_exp']=sal\_hike.YearsExperience\*sal\_hike.YearsExperience

#sal\_hike.drop('sq\_exp',axis=1,inplace=True)

#sal\_hike

Quad\_model=smf.ols('sal\_hike.Salary~(sal\_hike.YearsExperience\*sal\_hike.YearsExperience+sal\_hike.YearsExperience)',data=sal\_hike).fit()

#Quad\_model=smf.ols('sal\_hike.Salary~sal\_hike.sq\_exp+sal\_hike.YearsExperience',data=sal\_hike).fit()

Quad\_model.params

Out[71]:

Intercept 25792.200199

sal\_hike.YearsExperience 9449.962321

dtype: float64

Quad\_model.summary()

Out[72]:

<class 'statsmodels.iolib.summary.Summary'>

"""

OLS Regression Results

==============================================================================

Dep. Variable: sal\_hike.Salary R-squared: 0.957

Model: OLS Adj. R-squared: 0.955

Method: Least Squares F-statistic: 622.5

Date: Sun, 01 Nov 2020 Prob (F-statistic): 1.14e-20

Time: 12:44:23 Log-Likelihood: -301.44

No. Observations: 30 AIC: 606.9

Df Residuals: 28 BIC: 609.7

Df Model: 1

Covariance Type: nonrobust

============================================================================================

coef std err t P>|t| [0.025 0.975]

--------------------------------------------------------------------------------------------

Intercept 2.579e+04 2273.053 11.347 0.000 2.11e+04 3.04e+04

sal\_hike.YearsExperience 9449.9623 378.755 24.950 0.000 8674.119 1.02e+04

==============================================================================

Omnibus: 2.140 Durbin-Watson: 1.648

Prob(Omnibus): 0.343 Jarque-Bera (JB): 1.569

Skew: 0.363 Prob(JB): 0.456

Kurtosis: 2.147 Cond. No. 13.2

==============================================================================

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

"""

#will check for p-value<0.05 and high R-squared value(0.957) to be a good model,if not go for transformation

Quad\_predict=Quad\_model.predict(sal\_hike)

Quad\_predict

Out[75]:

0 36187.158752

1 38077.151217

2 39967.143681

3 44692.124842

4 46582.117306

5 53197.090931

6 54142.087163

7 56032.079627

8 56032.079627

9 60757.060788

10 62647.053252

11 63592.049484

12 63592.049484

13 64537.045717

14 68317.030645

15 72097.015574

16 73987.008038

17 75877.000502

18 81546.977895

19 82491.974127

20 90051.943985

21 92886.932681

22 100446.902538

23 103281.891235

24 108006.872395

25 110841.861092

26 115566.842252

27 116511.838485

28 123126.812110

29 125016.804574

dtype: float64

Quad\_Error=sal\_hike.Salary-Quad\_predict

Quad\_Error

Out[77]:

0 3155.841248

1 8127.848783

2 -2236.143681

3 -1167.124842

4 -6691.117306

5 3444.909069

6 6007.912837

7 -1587.079627

8 8412.920373

9 -3568.060788

10 570.946748

11 -7798.049484

12 -6635.049484

13 -7456.045717

14 -7206.030645

15 -4159.015574

16 -7958.008038

17 7210.999498

18 -183.977895

19 11448.025873

20 1686.056015

21 5386.067319

22 855.097462

23 10530.108765

24 1424.127605

25 -5259.861092

26 1402.157748

27 -3876.838485

28 -735.812110

29 -3144.804574

dtype: float64

plt.scatter(sal\_hike.YearsExperience,sal\_hike.Salary,c='r');plt.plot(sal\_hike.YearsExperience,Quad\_predict,c='b');plt.xlabel('years of experience');plt.ylabel('salary')

Out[78]: Text(0, 0.5, 'salary')

Quad\_model.conf\_int()

Out[79]:

0 1

Intercept 21136.061314 30448.339084

sal\_hike.YearsExperience 8674.118747 10225.805896

np.corrcoef(sal\_hike.Salary,Quad\_predict)

Out[80]:

array([[1. , 0.97824162],

[0.97824162, 1. ]])

#r=0.978

from sklearn.metrics import mean\_squared\_error

from math import sqrt

Quad\_rmse=sqrt(mean\_squared\_error(sal\_hike.Salary,Quad\_predict))

Quad\_rmse

Out[85]: 5592.043608760662