Build a prediction model for Salary_hike

```
import pandas as pd
import numpy as np
import seaborn as sns
import statsmodels.formula.api as smf
```

```
df=pd.read_csv('C:/Users/RIG1/Desktop/DS ASSIGNMENTS/QUESTIONS -all
assignments/ASS 4/Salary_Data.csv')
df.head(10)
```

```
   YearsExperience   Salary
0              1.1  39343.0
1              1.3  46205.0
2              1.5  37731.0
3              2.0  43525.0
4              2.2  39891.0
5              2.9  56642.0
6              3.0  60150.0
7              3.2  54445.0
8              3.2  64445.0
9              3.7  57189.0
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     float64
dtypes: float64(2)
memory usage: 608.0 bytes
```

```
df.describe()
```

```
       YearsExperience         Salary
count        30.000000      30.000000
mean          5.313333   76003.000000
std           2.837888   27414.429785
min           1.100000   37731.000000
25%           3.200000   56720.750000
50%           4.700000   65237.000000
75%           7.700000  100544.750000
max          10.500000  122391.000000
```

```
df.corr()
```

```
                 YearsExperience    Salary
YearsExperience         1.000000  0.978242
Salary                  0.978242  1.000000
```

```
df[df.duplicated()].shape
```

```
(0, 2)
```

```
df.dtypes
```

```
YearsExperience    float64
Salary             float64
dtype: object
```

```
# RENAMING THE COLUMNS
```

```
df=df.rename({'YearsExperience':'yr_exp','Salary':'sal'},axis=1)
```

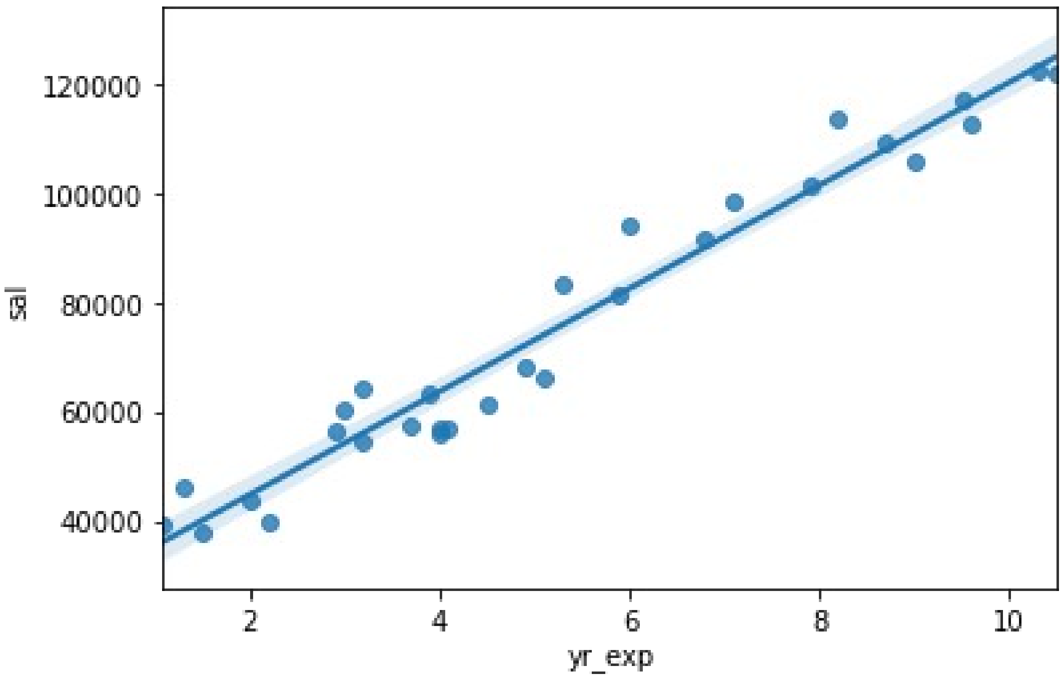
```
df.head()
```

```
   yr_exp      sal
0     1.1  39343.0
1     1.3  46205.0
2     1.5  37731.0
3     2.0  43525.0
4     2.2  39891.0
```

```
# PLOTTING
```

```
sns.regplot(x='yr_exp', y='sal',data=df)
```

```
<AxesSubplot:xlabel='yr_exp', ylabel='sal'>
```



```
# Y=B0+B1*X
# SAL=B0+B1*YR_EXP

## CREATING MODEL
```

```
model=smf.ols('sal~yr_exp',data=df).fit()
```

```
model
```

```
<statsmodels.regression.linear_model.RegressionResultsWrapper at
0x13399fa54f0>
```

```
model.params
```

```
Intercept    25792.200199
yr_exp        9449.962321
dtype: float64
```

```
# B0= Intercept    25792.200199
# B1= yr_exp        9449.962321
```

```
print(model.pvalues,'\n',model.tvalues)
```

```
Intercept    5.511950e-12
yr_exp       1.143068e-20
dtype: float64
 Intercept    11.346940
yr_exp       24.950094
dtype: float64
```

```
# VALIDATION
```

```
model.rsquared
```

```
0.9569566641435086
```

```
# H0: B1=0 (no relation bet. x and y) (means X is not imp. for Y
prediction) (no slope)
# H1: B1≠0 (there is a relation bet x and y) (means X is imp. for Y
prediction) (there is slope, +/-)
# B1 ------ is the slope

# IF P-VAL < ALPHA -------- reject H0
# IF P-VAL > ALPHA -------- reject H1

# p-val (yr_exp       1.143068e-20)
```

prediction

automatic prediction

```
sal=pd.Series([1,12])
```

```
sal
```

```
0     1
1    12
dtype: int64
```

```
prediction=pd.DataFrame(sal,columns=['yr_exp'])
```

```
## BECOMES A DataFrame now
prediction
```

```
   yr_exp
0       1
1      12
```

```
model.predict(prediction)
```

```
0     35242.162520
1    139191.748056
dtype: float64
```

manualy predict

```
# sal=B0+B1*yr_exp
```

```
sal=25792.200199+9449.962321*12
sal
```

```
139191.748051
```