

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

```
In [2]: data = pd.read_csv("C:/Users/shara/OneDrive/Desktop/ML LAB/salary_data.csv")
```

```
In [7]: x = np.array(data.iloc[:,["YearsExperience"]])
```

```
In [20]: y = np.array(data.iloc[:,1])
```

```
In [6]: from sklearn.model_selection import train_test_split
```

```
In [41]: xtrain,xtest,ytrain,ytest = train_test_split(x,y,train_size=.80,random_state=4697)
```

```
In [42]: from sklearn.linear_model import LinearRegression
```

```
In [43]: model = LinearRegression()
```

```
In [44]: model.fit(xtrain,ytrain)
```

```
Out[44]: LinearRegression()
```

```
In [45]: ypred = model.predict(xtest)
```

```
In [46]: ypred
```

```
Out[46]: array([ 44788.50460346, 107700.91806713, 100188.98810131,  81409.16318679,
 122724.77799875,  89860.08439832])
```

```
In [47]: ytest
```

```
Out[47]: array([ 43525, 109431, 101302,  81363, 122391,  91738], dtype=int64)
```

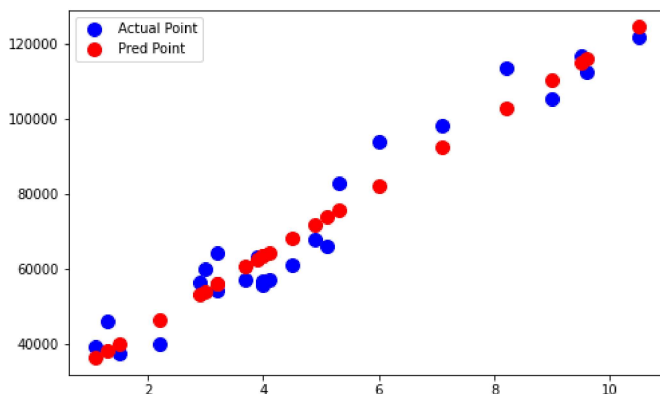
```
In [48]: from sklearn.metrics import r2_score
```

```
In [49]: r2 = r2_score(ytest,ypred)
```

```
In [50]: r2
```

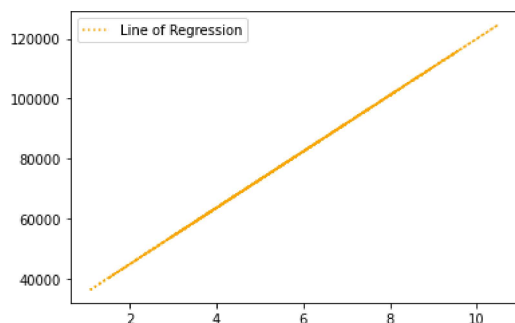
```
Out[50]: 0.9974925617006956
```

```
In [51]: plt.figure(figsize=(8,5))
plt.scatter(xtrain,ytrain,color='blue',s=100,label="Actual Point")
plt.scatter(xtrain,model.predict(xtrain),color='red',s=100,label="Pred Point")
plt.legend()
plt.show()
```

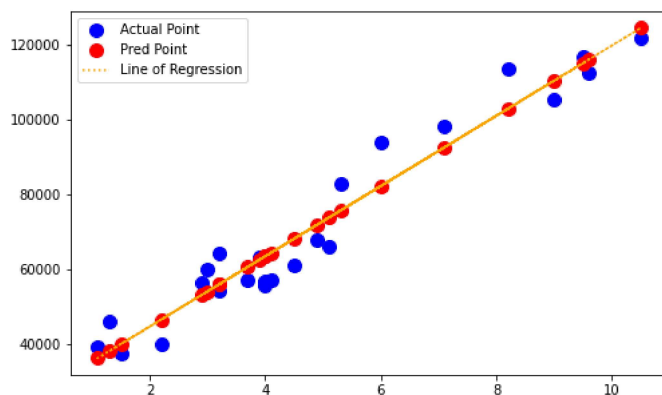


```
In [55]: plt.plot(xtrain, model.predict(xtrain), linestyle='dotted', color='orange', label="Line of Regression")
plt.legend()
```

```
Out[55]: <matplotlib.legend.Legend at 0x1fa7e557370>
```



```
In [56]: plt.figure(figsize=(8,5))
plt.scatter(xtrain,ytrain,color='blue',s=100,label="Actual Point")
plt.scatter(xtrain,model.predict(xtrain),color='red',s=100,label="Pred Point")
plt.plot(xtrain, model.predict(xtrain), linestyle='dotted', color='orange', label="Line of Regression")
plt.legend()
plt.show()
```



```
In [52]: scores = []
for i in range(5000):
    xtrain1,xtest1,ytrain1,ytest1 = train_test_split(x,y,train_size =.80,random_state=i)
    model1 = LinearRegression()
    model1.fit(xtrain1,ytrain1)
    ypred1 = model1.predict(xtest1)
    scores.append(r2_score(ytest1,ypred1))
```

```
In [53]: np.max(scores)
```

```
Out[53]: 0.9974925617006956
```

```
In [54]: np.argmax(scores)
```

```
Out[54]: 4697
```

```
In [ ]:
```