

lr-salary-pred

September 12, 2024

1 By prisca

```
[2]: # with just the years of experience, i was able to predict the salary of an  
      ↪ individual
```

```
[2]: import pandas as pd  
      import matplotlib.pyplot as plt  
      import seaborn as sns
```

```
[3]: df=pd.read_csv(r'C:\Users\USER\Documents\Salary_dataset.csv')
```

```
[4]: df.head()
```

```
[4]:   Unnamed: 0  YearsExperience  Salary  
0          0          1.2  39344.0  
1          1          1.4  46206.0  
2          2          1.6  37732.0  
3          3          2.1  43526.0  
4          4          2.3  39892.0
```

```
[5]: df.shape
```

```
[5]: (30, 3)
```

```
[6]: df.isnull().sum()
```

```
[6]: Unnamed: 0          0  
     YearsExperience  0  
     Salary          0  
     dtype: int64
```

```
[7]: df.duplicated().sum()
```

```
[7]: 0
```

```
[8]: df.drop(['Unnamed: 0'],axis=1,inplace=True)
```

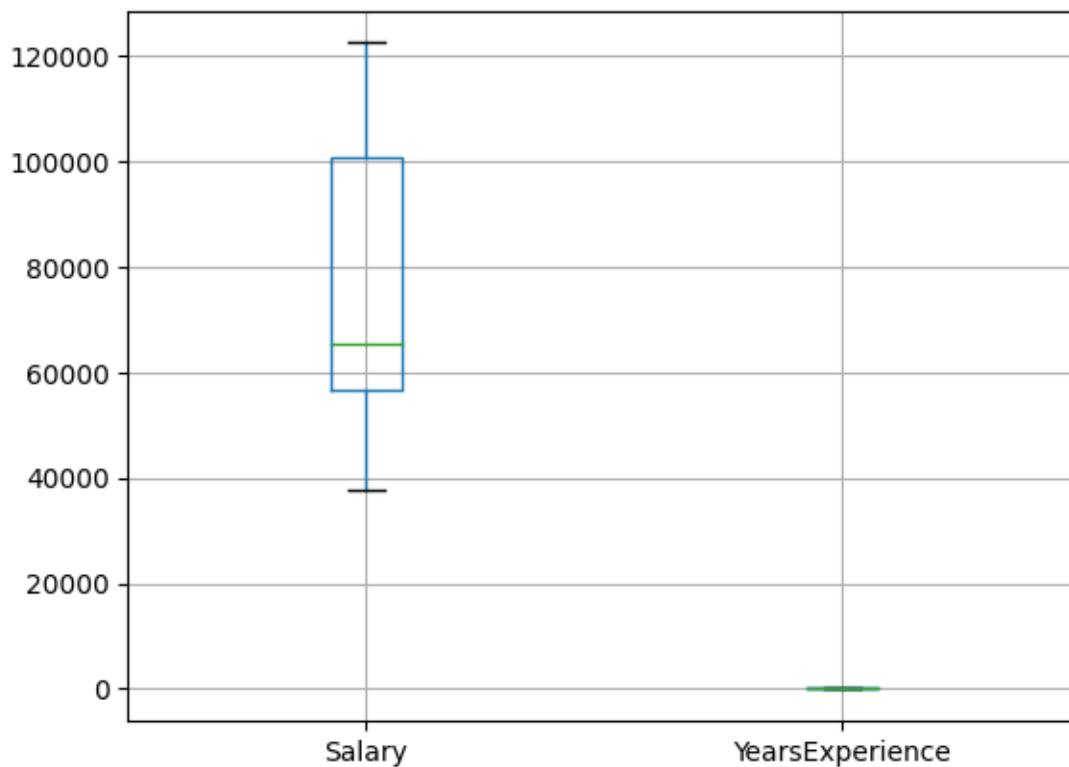
```
[9]: df.head()
```

```
[9]:   YearsExperience  Salary
0             1.2  39344.0
1             1.4  46206.0
2             1.6  37732.0
3             2.1  43526.0
4             2.3  39892.0
```

```
[10]: # no outliers detected
```

```
df.boxplot(column=['Salary', 'YearsExperience'])
```

```
[10]: <Axes: >
```



```
[11]: sns.histplot(df['Salary'].unique())
```

```
[11]: <Axes: ylabel='Count'>
```

```
[12]: df['Salary'].value_counts()
```

```
[12]: Salary
      39344.0    1
      46206.0    1
      122392.0   1
      112636.0   1
      116970.0   1
      105583.0   1
      109432.0   1
      113813.0   1
      101303.0   1
      98274.0    1
      91739.0    1
      93941.0    1
      81364.0    1
      83089.0    1
      66030.0    1
      67939.0    1
      61112.0    1
      57082.0    1
      56958.0    1
      55795.0    1
      63219.0    1
      57190.0    1
      64446.0    1
      54446.0    1
      60151.0    1
      56643.0    1
      39892.0    1
      43526.0    1
      37732.0    1
      121873.0   1
      Name: count, dtype: int64
```

```
[13]: sns.scatterplot(x='Salary',y='YearsExperience',data=df)
```

```
[13]: <Axes: xlabel='Salary', ylabel='Count'>
```

2 data splitting

```
[14]: from sklearn.preprocessing import StandardScaler
      from sklearn.model_selection import train_test_split
```

```
[15]: scaler=StandardScaler()
```

```
[16]: x=df['YearsExperience']
      y=df['Salary']
```

```
[17]: import numpy as np
x=np.array(x).reshape(-1,1)
y=np.array(y).reshape(-1,1)
```

```
[18]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

```
[19]: x_train=scaler.fit_transform(x_train)
x_test=scaler.transform(x_test)
```

3 model building

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

```
[21]: model=LinearRegression()
```

```
[22]: model.fit(x_train,y_train)
```

```
[22]: LinearRegression()
```

```
[23]: model.score(x_train,y_train)
```

```
[23]: 0.9411949620562127
```

```
[24]: ypred=model.predict(x_test)
```

```
[25]: from sklearn import metrics
from sklearn.metrics import r2_score
```

4 model evaluation

```
[26]: r2_score(y_test,ypred)
```

```
[26]: 0.988169515729126
```

```
[27]: #visualizing the result
```

```
[28]: plt.scatter(x_test,y_test, color='green')
plt.plot(x_test,model.predict(x_test),color='red')
```

```
[28]: [<matplotlib.lines.Line2D at 0x1c6c8b548e0>]
```

```
[29]: #visualizing trained data
```

```
[30]: plt.scatter(x_train,y_train)
plt.plot(x_train, model.predict(x_train), color = 'blue')
```

```
[30]: [<matplotlib.lines.Line2D at 0x1c6c8b583a0>]
```

5 testing the model

```
[31]: #using the years of experience to predict the salary of a person
```

```
[32]: model.predict(scaler.transform([[6.0      ]]))
```

```
[32]: array([[81725.29239833]])
```

```
[42]: model.predict(scaler.transform([[50.0      ]]))
```

```
[42]: array([[491478.59797441]])
```

```
[34]: model.predict(scaler.transform([[60.0      ]]))
```

```
[34]: array([[584604.34924171]])
```

```
[35]: # model is doing so good
```

6 saving the model

```
[37]: import joblib
```

```
[38]: model_joblib= joblib.dump(model, 'salary_prediction')
```

```
[ ]:
```