

▼ Machine Learning

▼ Simple Linear Regression

▼ Step 1.Import Libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn
```

▼ Step 2. Import Dataset

```
df = pd.read_csv('salary_data.csv')
df.head()
```

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

▼ Step 3. Selecting input and output variable

```
X = df[["YearsExperience"]]
y = df["Salary"]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=0)
```

▼ Step 4.Making linear regression model

```
from sklearn.linear_model import LinearRegression
model= LinearRegression()
```

▼ Step 5.Fitting the model

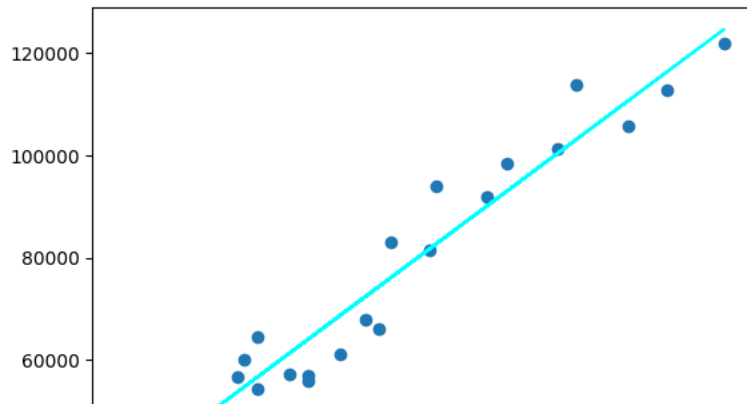
```
model = model.fit(X,y)
model
```

```
LinearRegression
LinearRegression()
```

▼ Step 6. Plot the Graph

```
import matplotlib.pyplot as plt
plt.scatter(X_train,y_train)
plt.plot(X_train.values, model.predict(X_train), color="aqua")
```

[<matplotlib.lines.Line2D at 0x7f6781baf8e0>]



▼ step 7. Predicting the Model

```
model.predict([[10]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was
warnings.warn(
array([120291.82341322])
```

▼ Step 8. Evaluating the Model

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
model = LinearRegression()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = r2_score(y_test, y_pred)
print("Accuracy score: {:.2f}".format(accuracy))
```

```
Accuracy score: 0.99
```

▼ Step 9. Splitting and computing score according to 80/20

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
# Splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
# Creating and fitting the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
# Predicting on the test set
y_pred = model.predict(X_test)
# Computing the accuracy (R-squared score)
accuracy = r2_score(y_test, y_pred)
# Printing the accuracy
print("Accuracy:", accuracy)
```

```
Accuracy: 0.988169515729126
```

✓ 0s completed at 8:55 AM

● ×