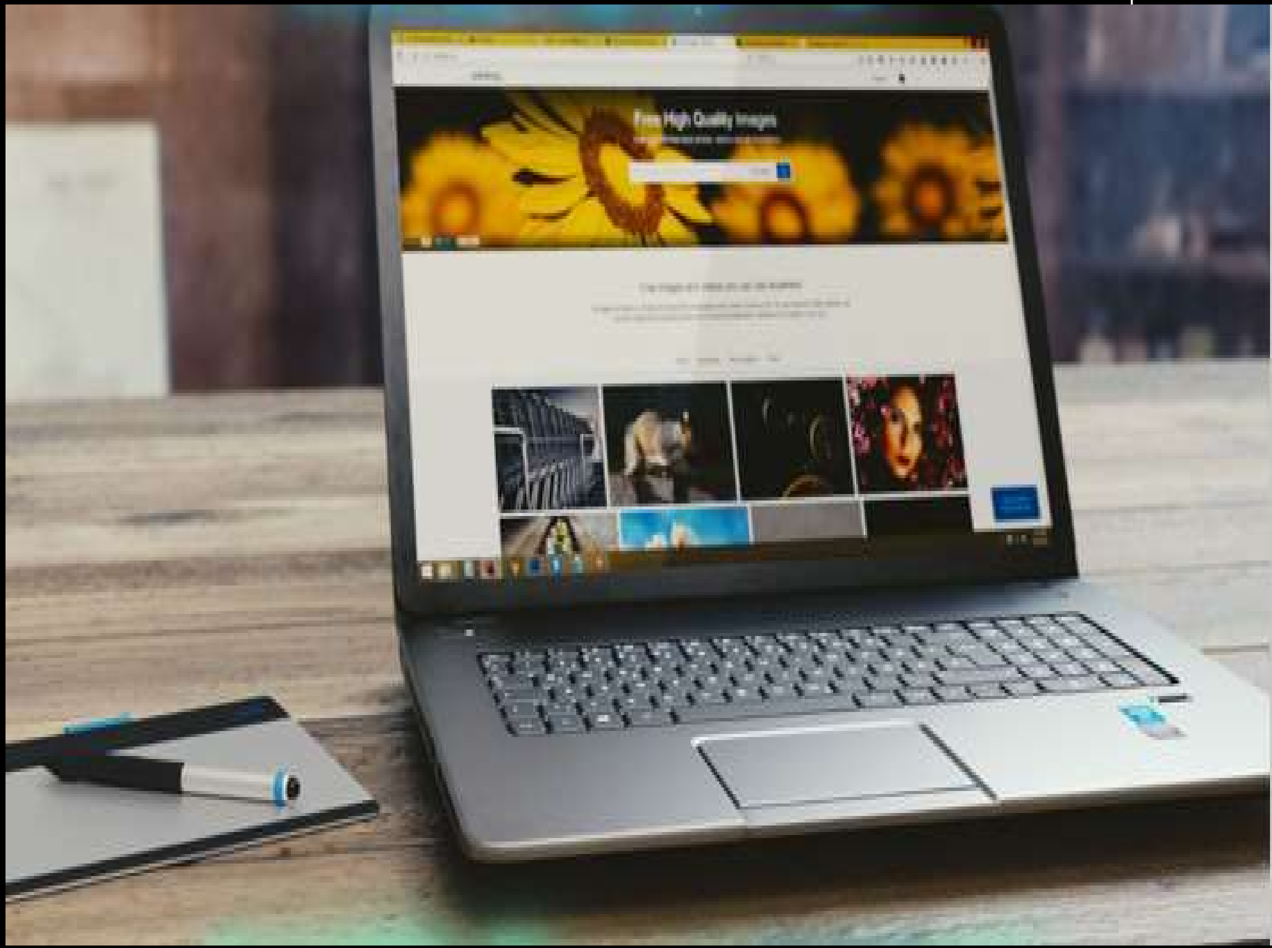
LAB-1

Calculate Prediction Confidence Intervals for a Linear Regression Model



By

Dr. Sumit Kumar Singh

Associate Professor

SR University

WHAT IS REGRESSION?

- Regression helps us understand how one variable changes when another changes.

Example:

Study hours → Exam marks

House size → House price

- Regression draws a best-fitting line through the data to model this relationship.

EXAMPLE: STUDY HOURS VS MARKS

Study Hours (X)	Marks (Y)
1	45
2	50
3	55
4	60
5	65

Observation: As study hours increase, marks increase, the pattern looks like a straight line.

WHAT IS LINEAR REGRESSION?

- Linear Regression assumes a straight-line relationship:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Where:

Y – Dependent variable (output)

X – Independent variable (input)

β_0 – Intercept (value of Y when $X=0$)

β_1 – Slope (change in Y for one unit change in X)

ε – Random error

Multiple Linear Regression (MLR) extends this to **two or more independent variables**:

$$Y = a + b_1X_1 + b_2X_2 + b_3X_3 + \cdots + b_nX_n + \epsilon$$

What is R² (R-squared)?

- **R² (Coefficient of Determination)** measures **how well the regression line fits the data**.
- It tells us: "How much of the variation in Y (Marks) is explained by X (Study Hours)."

R^2=1 → Perfect linear relationship
R^2=0 → No relationship
R^2=0.8 → 80% of variation explained by model

FORMULA

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

$$SS_{res} = \sum \left(Y - \hat{Y}\right)^2$$

$$SS_{tot} = \sum (Y - \bar{Y})^2$$

SS_{tot} : How much the actual data vary around their mean (total variation).

SS_{res} : How much error remains after fitting the regression model (unexplained variation).

WHAT IS A CONFIDENCE INTERVAL?

- A confidence interval (CI) gives a range within which we expect the true value to fall.

Example:

If predicted marks = 70, CI = [68, 72],

then we are 95% confident the true mean lies between 68 and 72.

CONFIDENCE INTERVAL FORMULA

$\hat{Y} \pm t_{(\alpha/2,\, n-2)} \times SE(\hat{Y})$

Where:

$\hat{Y}$ – Predicted value

$SE(\hat{Y})$ – Standard error of prediction

$t_{(\alpha/2,\, n-2)}$ – t-critical value

For a 95% confidence interval,
we're 95% confident the true value is inside the range.
so $\alpha = 1 - 0.95 = 0.05$.
So, α=0.05 represents the remaining 5% risk

Common choice: $\alpha = 0.05 \rightarrow 95\%$ confidence level.

$$\hat{Y} \pm t_{(\alpha/2,\, n-2)} \times SE(\hat{Y})$$

Why $\alpha/2$?

- When we make a **two-sided confidence interval** (both upper and lower limits), we divide that 5% risk equally into **two tails** of the t-distribution:

2.5% in the **left tail**
2.5% in the **right tail**
So, we use $\alpha/2 = 0.025$ for each side

Why $n - 2$?

if there are n total observations, and we estimated 2 parameters:

$$\text{Degrees } of\ Freedom = n - 2$$

The **t-distribution** depends on the number of degrees of freedom.
Fewer data points (small n) → wider tails → larger t-critical value (more uncertainty).
So, $t_{\alpha/2,n-2}$automatically adjusts based on sample size.

STEPS IN PYTHON

- Import libraries and dataset
- Build the LR model
- Make prediction and confidence intervals
- Visualize Regression line with Confidence band

STEP 1: IMPORT LIBRARIES AND DATASET

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm

data = {
    'Hours': [1, 2, 3, 4, 5, 6, 7, 8],
    'Marks': [45, 48, 52, 56, 61, 65, 68, 72]
}
df = pd.DataFrame(data)
print(df.head())
```

Python library for **statistical analysis**, **data exploration**, and **regression modeling**.

STEP 2: FIT LINEAR REGRESSION MODEL

```
X = sm.add_constant(df['Hours'])        # adds intercept term
y = df['Marks']

model = sm.OLS(y, X).fit()
print(model.summary())
```

sm.OLS() = Ordinary Least Squares regression

model.summary() shows slope, intercept, and model accuracy (R^2)

STEP 3: MAKE PREDICTIONS & COMPUTE CONFIDENCE INTERVALS

```
pred = model.get_prediction(X)
conf_int = pred.conf_int(alpha=0.05)
predicted = model.predict(X)

result = df.copy()
result['Predicted'] = predicted
result['Lower_CI'] = conf_int[:,0]
result['Upper_CI'] = conf_int[:,1]

print(result.head())
```

- model.get_prediction(X) → This line asks the fitted regression model to generate predictions and statistical details (like confidence intervals) for each input in X.

It doesn't just give predicted values , it returns a **PredictionResults object**, which contains:

Mean predicted values $\hat{Y}$

Standard errors of predictions

Confidence intervals

- conf_int = pred.conf_int(alpha=0.05) → This extracts the **confidence intervals** (CIs) from the prediction results.

alpha=0.05 means a 95% confidence level (since $1-\alpha=0.95$).

The result is an array with two columns:

Column 0 → Lower bound of the 95% CI

Column 1 → Upper bound of the 95% CI

- predicted = model.predict(X) → This line generates the **predicted values** $\hat{Y}_i$ from your regression model, the values that lie exactly on your best-fit line.

Line	Command	What It Does
1	model.get_prediction(X)	Generates predicted means + errors + intervals
2	pred.conf_int(alpha=0.05)	Extracts lower & upper 95% confidence bounds
3	model.predict(X)	Calculates predicted (fitted) values $\hat{Y}$
4	Add new columns to df	Combines original data with predicted + CI
5	print(result)	Displays the result summary table

STEP 4: VISUALIZE REGRESSION LINE WITH CONFIDENCE BAND

```
plt.figure(figsize=(8,5))
plt.scatter(df['Hours'], df['Marks'], color='blue', label='Actual')
plt.plot(df['Hours'], predicted, color='red', label='Predicted line')
plt.fill_between(df['Hours'], result['Lower_CI'], result['Upper_CI'], color='gray', alpha=0.3,
label='95% Confidence Interval')
plt.xlabel('Study Hours')
plt.ylabel('Marks')
plt.title('Linear Regression with 95% Confidence Interval')
plt.legend()
plt.show()
```

INTERPRETING THE GRAPH

Red line: Model's predicted values.

Gray band: 95% confidence interval around predictions.

Narrow band → more certainty.

Wide band → less certainty.