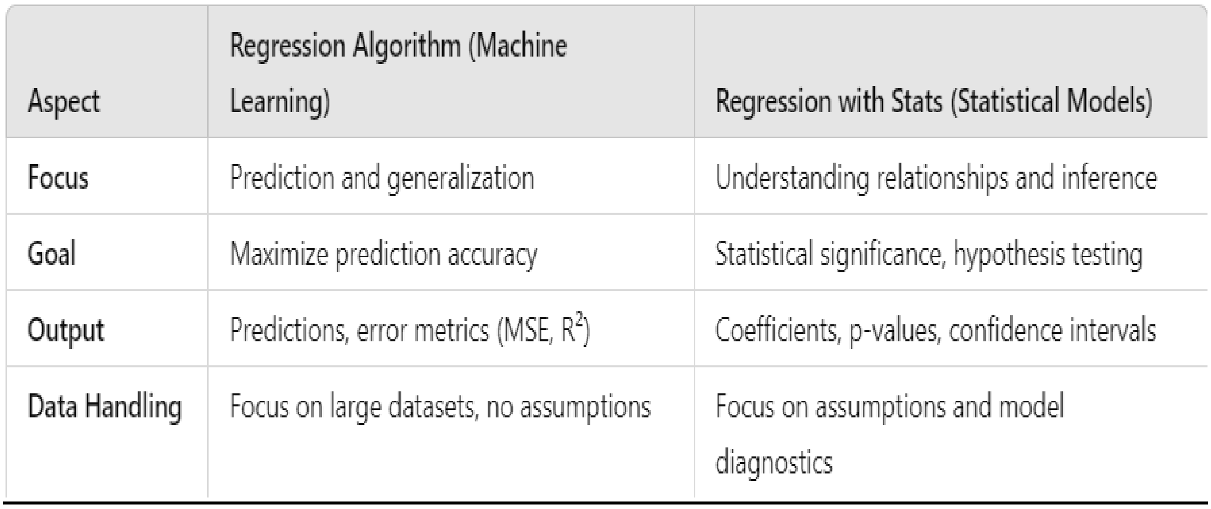
Supervised Learning-Regression

Measure of linear relationship, Regression with stats models

Regression algorithm (uisng machine learning) and **regression with statistical models** lies in their primary goals, approaches, and outputs. Both aim to model relationships between variables, but their use cases and how they handle data can be quite distinct.

- **Regression Algorithm (Machine Learning)**: The focus is on **prediction** and **generalization** to unseen data. It aims to create a model that can predict the target (dependent variable) based on the features (independent variables.
- **Regression with Stats (Statistical Models)**: The focus is on **understanding relationships** between variables. It's often used for explaining the nature of the relationship between independent variables (predictors) and the dependent variable (outcome).



Aspect	Regression Algorithm (Machine Learning)	Regression with Stats (Statistical Models)
Focus	Prediction and generalization	Understanding relationships and inference
Goal	Maximize prediction accuracy	Statistical significance, hypothesis testing
Output	Predictions, error metrics (MSE, R^2)	Coefficients, p-values, confidence intervals
Data Handling	Focus on large datasets, no assumptions	Focus on assumptions and model diagnostics

To measure the linear relationship between two variables and perform regression analysis using **statsmodels** in Python, you can follow these steps:

1. Measure Linear Relationship (Correlation)

Before performing regression, it's common to measure the linear relationship between two variables using **correlation**. You can use **Pearson's correlation coefficient** to measure this.

```python
import pandas as pd
import numpy as np

# Example data
data = {'X': [1, 2, 3, 4, 5], 'Y': [2, 4, 5, 4, 5]}
df = pd.DataFrame(data)

# Pearson correlation coefficient
correlation = df['X'].corr(df['Y'])
print("Pearson Correlation Coefficient:", correlation)
```

Pearson Correlation Coefficient: 0.7745966692414834

The correlation coefficient ranges from -1 to 1:

- 1 indicates a perfect positive linear relationship.
- -1 indicates a perfect negative linear relationship.
- 0 indicates no linear relationship.

2. Linear Regression Using Statsmodels

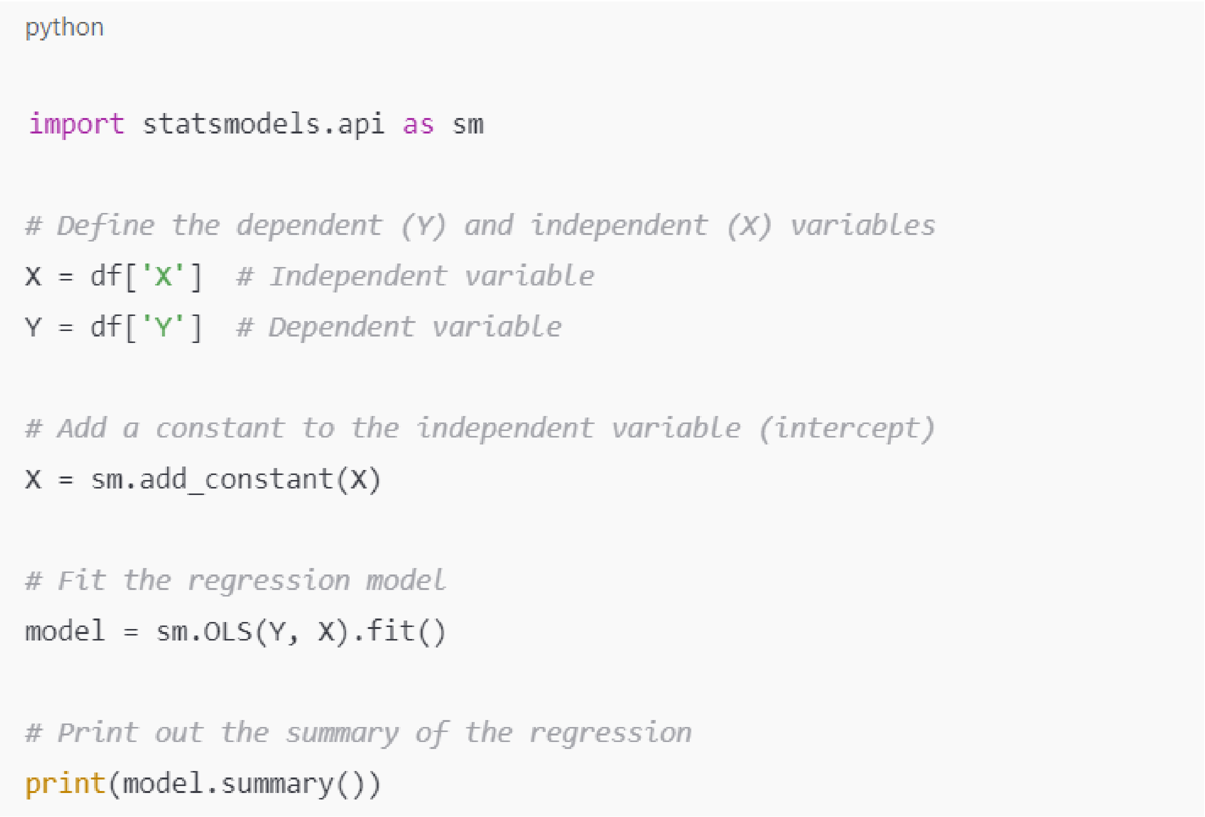
You can use the **statsmodels** library to perform linear regression. Here's an example:

Step-by-step guide:

1. Install **statsmodels** if not already installed.

```
pip install statsmodels
```

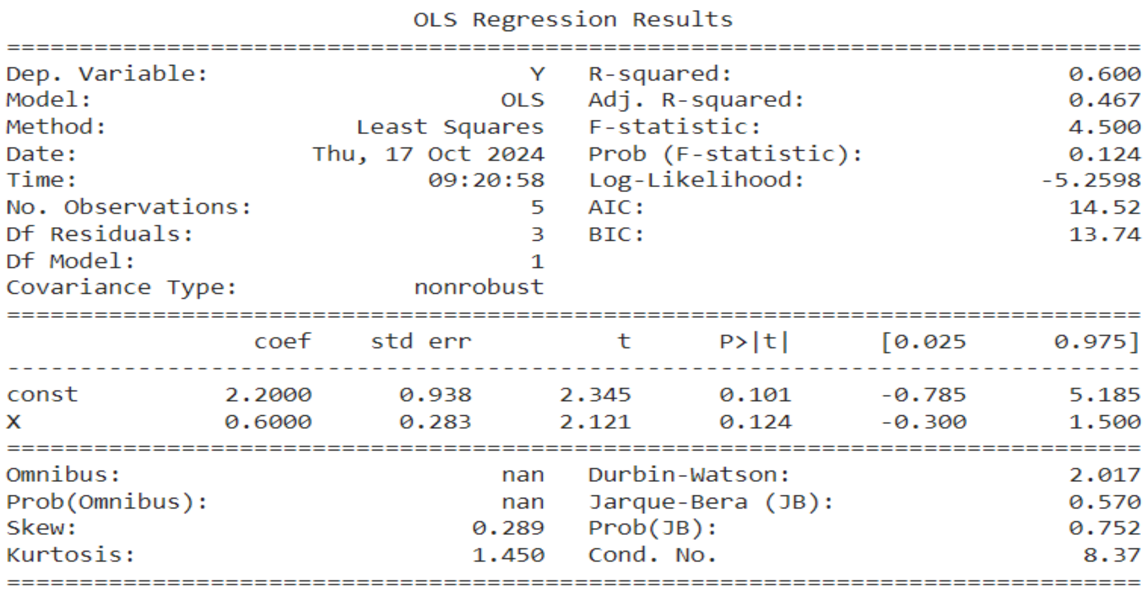
2. **Perform the regression:**



python

```python
import statsmodels.api as sm

# Define the dependent (Y) and independent (X) variables
X = df['X']  # Independent variable
Y = df['Y']  # Dependent variable

# Add a constant to the independent variable (intercept)
X = sm.add_constant(X)

# Fit the regression model
model = sm.OLS(Y, X).fit()

# Print out the summary of the regression
print(model.summary())
```

- **sm.add_constant(X)** adds a constant term to the predictor variable for the intercept.
- **sm.OLS(Y, X)** creates the OLS (Ordinary Least Squares) model.
- **model.fit()** fits the model to the data.
- **model.summary()** provides a detailed statistical summary, including coefficients, R-squared value, p-values, and confidence intervals.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.600
Model:                            OLS   Adj. R-squared:                  0.467
Method:                 Least Squares   F-statistic:                     4.500
Date:                Thu, 17 Oct 2024   Prob (F-statistic):              0.124
Time:                        09:20:58   Log-Likelihood:                -5.2598
No. Observations:                   5   AIC:                             14.52
Df Residuals:                       3   BIC:                             13.74
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.2000      0.938      2.345      0.101      -0.785       5.185
X              0.6000      0.283      2.121      0.124      -0.300       1.500
==============================================================================
Omnibus:                          nan   Durbin-Watson:                   2.017
Prob(Omnibus):                    nan   Jarque-Bera (JB):                0.570
Skew:                           0.289   Prob(JB):                        0.752
Kurtosis:                       1.450   Cond. No.                         8.37
==============================================================================
```

3. Interpreting the Results

- **R-squared**: Indicates how well the independent variable explains the variability in the dependent variable.
- **p-value**: Tests the null hypothesis that the coefficient of a variable is zero (no relationship). A p-value less than 0.05 usually indicates statistical significance.
- **Coefficients**: The slope of the regression line (relationship strength) and the intercept.

Studying regression with **statsmodels** offers several advantages, especially for those who want a deeper understanding of regression analysis and statistical modeling. Here are some reasons why **statsmodels** is beneficial for studying regression:

Detailed Statistical Output

Unlike other libraries like **scikit-learn**, which focus primarily on prediction, **statsmodels** provides comprehensive statistical information about the regression model. The output includes:

- **R-squared and Adjusted R-squared**: Measures of model fit.
- **P-values**: Help determine the statistical significance of predictors.
- **Confidence Intervals**: Shows the range of values for the coefficients with a certain level of confidence.