Linear Regression:

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. It assumes that there is a linear relationship between the independent variables (predictors) and the dependent variable (outcome). In simple terms, it tries to fit a straight line to the data points that best represents the relationship between the variables.

How it works:

Data Collection: First, you collect your data. You need two types of variables: the independent variable(s) (often denoted as X) and the dependent variable (often denoted as Y).
Data Exploration: You explore your data to understand the relationship between the independent and dependent variables. This can involve visualizations like scatter plots.
Model Building: Then, you build the linear regression model. Mathematically, it can be represented as:$Y = \beta0 + \beta1*X + \varepsilon$Where:
Y is the dependent variable
X is the independent variable
β0 is the intercept
β1 is the coefficient for the independent variable
ε is the error term
The goal is to find the best-fitting line (or hyperplane in the case of multiple independent variables) that minimizes the sum of squared differences between the actual and predicted values.
Model Evaluation: Once the model is built, you evaluate its performance using metrics like R-squared, Mean Squared Error (MSE), etc., to see how well the model fits the data.
Practical Example:

Let's say we want to predict the price of houses based on their size. We have a dataset containing the sizes of houses and their corresponding prices.

```
# Importing necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Generating example data
np.random.seed(0)
house_sizes = np.random.randint(1000, 3000, 100)
house_prices = 50000 + 300 * house_sizes + np.random.normal(0, 10000, 100)

# Creating a DataFrame
data = pd.DataFrame({'Size': house_sizes, 'Price': house_prices})

# Visualizing the data
plt.scatter(data['Size'], data['Price'])
```

```
plt.title('House Price vs Size')
plt.xlabel('Size (sqft)')
plt.ylabel('Price ($)')
plt.show()

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data['Size'], data['Price'], test_size=0.2,
random_state=42)

# Reshaping the data (required by scikit-learn)
X_train = X_train.values.reshape(-1, 1)
X_test = X_test.values.reshape(-1, 1)

# Building and training the model
model = LinearRegression()
model.fit(X_train, y_train)

# Making predictions
y_pred = model.predict(X_test)

# Evaluating the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

In this example, we first generate some example data for house sizes and prices. Then, we visualize the data using a scatter plot. After splitting the data into training and testing sets, we build a linear regression model using scikit-learn's LinearRegression class. Finally, we make predictions on the test set and evaluate the model using Mean Squared Error (MSE) and R-squared.