Lesson 21: Linear Regression

Data Science with Python – BSc Course

45 Minutes

Learning Objectives

The Problem: A portfolio manager needs to understand how stocks respond to market movements. How do we quantify systematic risk?

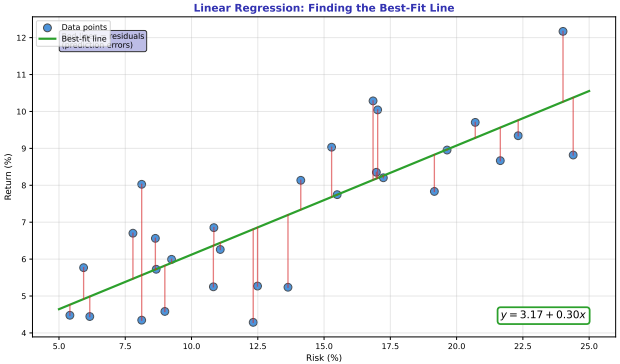
After this lesson, you will be able to:

- Understand OLS estimation and the least squares principle
- Fit linear models using sklearn's LinearRegression
- Interpret coefficients (slope as beta, intercept as alpha)
- Estimate CAPM beta to classify stocks by risk profile

Finance Application: Stock classification for portfolio construction

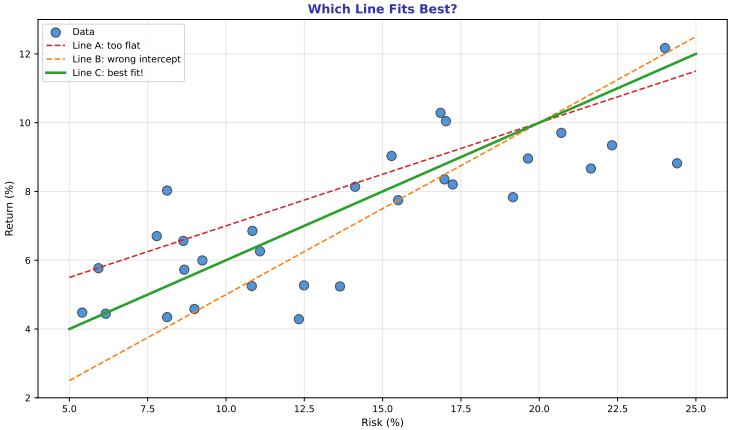
Finding the Best-Fit Line

- Linear regression finds the line that best describes the relationship
- In finance: How does stock return respond to market return?



The "best" line minimizes the sum of squared vertical distances (residuals)

Which Line Fits Best?

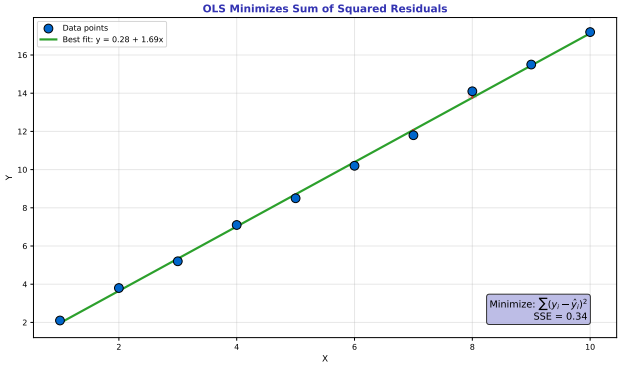


OLS finds the unique line that minimizes total squared error

OLS Formula

Ordinary Least Squares: The Math

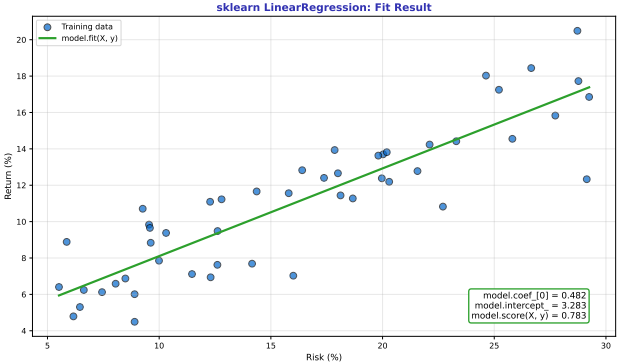
- Goal: Find β_0, β_1 that minimize $\sum(y_i - \hat{y}_i)^2$
- Solution: $\beta_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$



Why squares? (1) Makes errors positive, (2) Penalizes large errors more

Implementation in Python

- `from sklearn.linear_model import LinearRegression`
- `model = LinearRegression().fit(X, y)`
- Access: `model.coef_` (slope), `model.intercept_`

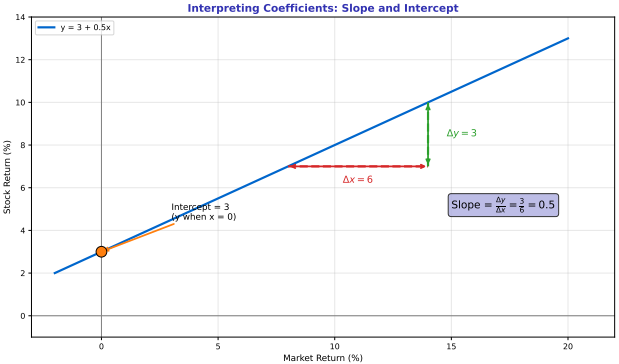


Pattern: model.fit(X, y) then model.predict(X_new) – works for all sklearn models

Coefficient Interpretation

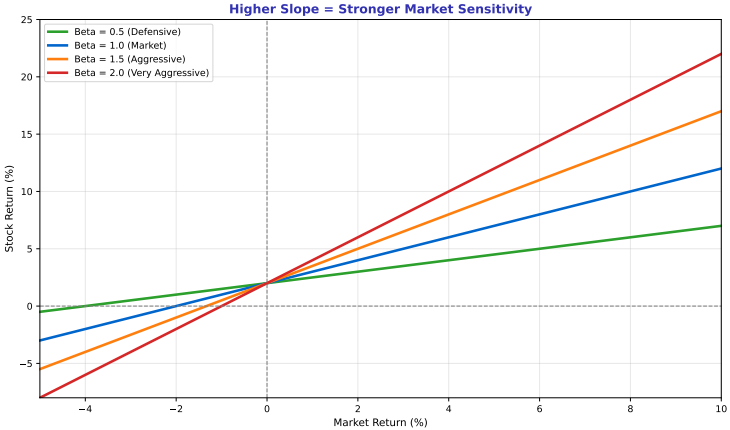
What Do the Numbers Mean?

- **Slope (β_1):** For each 1% market move, stock moves β_1%
- **Intercept (β_0):** Stock's return when market return is zero



Finance translation: Slope = beta (systematic risk), Intercept = alpha (skill)

Different Slopes = Different Betas

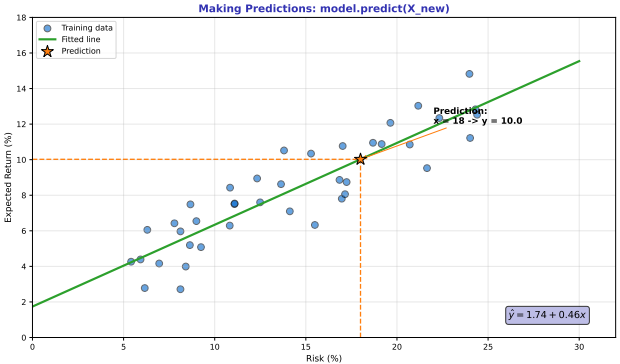


Higher beta = stock amplifies market moves more

Making Predictions

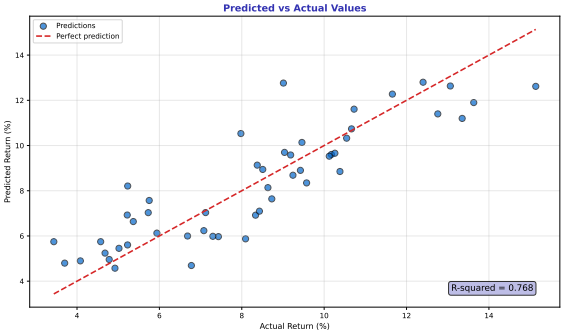
Using the Model for Forecasting

- Once fitted, predict stock return for any market scenario
- `predicted = model.predict([[market_return]])`



Caution: Predictions assume the relationship stays stable

Predicted vs Actual



Points on the diagonal = perfect predictions. R-squared measures fit quality.

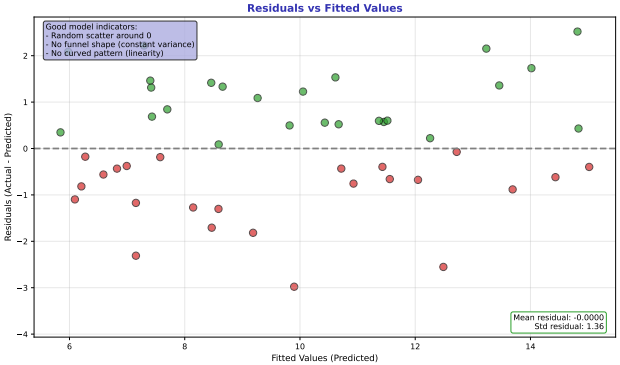
Extrapolation Warning

Never predict outside your training data range!

Residuals

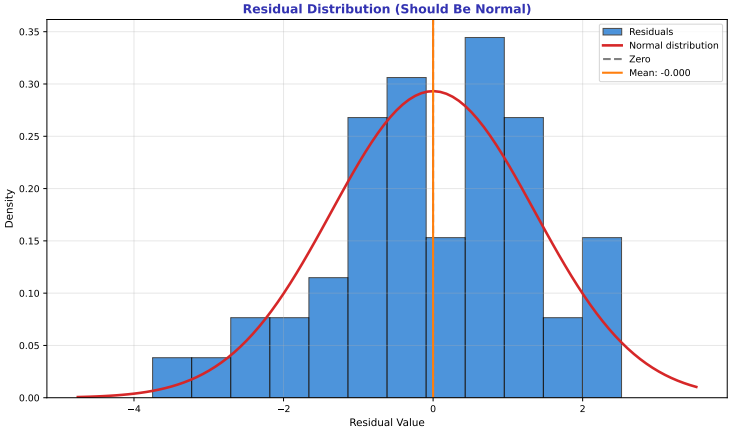
Checking Prediction Quality

- Residual = Actual - Predicted ($e_i = y_i - \hat{y}_i$)
- Good model: residuals should be random (no pattern)



Plot residuals vs predicted: if you see a pattern, the model is missing something

Residual Distribution

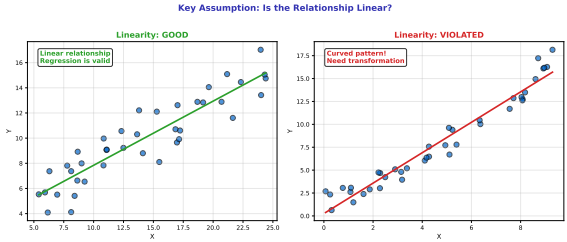


Normality assumption: residuals should follow a bell curve centered at zero

Assumptions

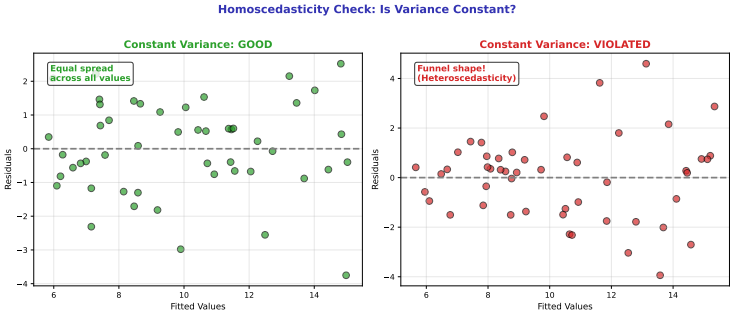
When Does Linear Regression Work?

- **Linearity:** Relationship is actually linear (not curved)
- **Homoscedasticity:** Variance of errors is constant
- **Normality:** Residuals are normally distributed



Finance reality: Stock returns often violate these – check residuals

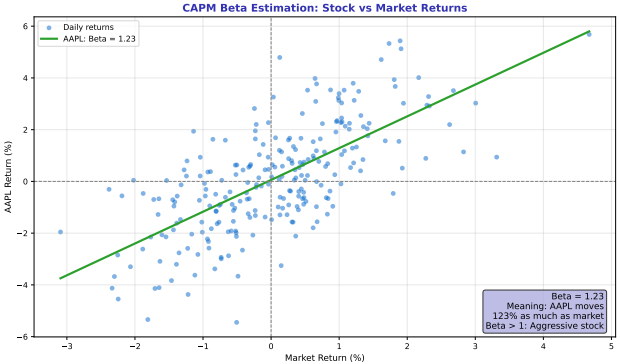
Homoscedasticity Check



Funnel shape = heteroscedasticity. Fix: weighted least squares or log transform

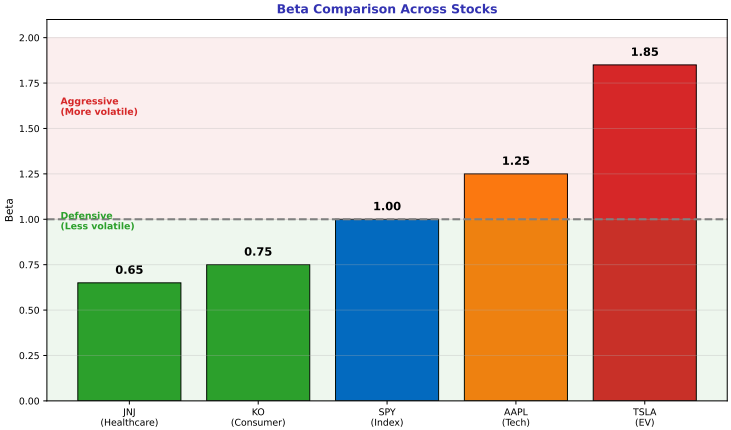
The Solution: Stock Classification by Systematic Risk

- **Beta > 1:** Aggressive stock – amplifies market moves
- **Beta < 1:** Defensive stock – dampens volatility



Alpha (β_0): Outperformance after risk adjustment

Beta Comparison



Mix defensive (low beta) and aggressive (high beta) stocks based on risk tolerance

Hands-On Exercise (25 min)

Task: Estimate Beta for Your Favorite Stock

1. Download 1 year of daily returns for a stock (e.g., MSFT) and SPY
2. Fit: `model.fit(spy_returns, stock_returns)`
3. Extract and interpret: What is the beta? What is the alpha?
4. Plot the regression line with actual data points

Deliverable: Scatter plot with regression line, annotated with beta value.

Extension: Compare beta estimates using different time periods (1yr vs 5yr)

Lesson Summary

Problem Solved: We can now quantify systematic risk using CAPM beta via linear regression.

Key Takeaways:

- OLS finds the line that minimizes squared errors
- sklearn: `LinearRegression().fit(X, y)` – three lines of code
- Slope = beta (market sensitivity), Intercept = alpha (skill)

Next Lesson: Regularization (L22) – what happens with too many features?

Memory: Beta = slope of stock vs market regression. High beta = high volatility.