Regression Analysis

Introduction to Regression

Regression analysis is a statistical technique used to model and analyze the relationship between a dependent variable (target) and one or more independent variables (predictors). In data science, regression is fundamental for prediction, trend analysis, and understanding variable relationships.

There are two main types:

- **Simple Linear Regression:** One independent variable, one dependent variable.
- **Multiple Linear Regression:** Two or more independent variables, one dependent variable.

Simple Linear Regression

Definition

Simple linear regression models the relationship between two continuous variables by fitting a straight line to the data. The goal is to predict the dependent variable y from the independent variable x using the equation:

$$ \hat{y} = b_0 + b_1 x $$

- $\hat{y}$ (y-hat): The predicted or estimated value of the dependent variable (y) given a specific x.
- b_0: Intercept (value of y when $x = 0$)
- b_1: Slope (change in y for a one-unit change in x)

Assumptions

- **Linearity:** The relationship between x and y is linear.
- **Independence:** Observations are independent of each other.
- **Homoscedasticity:** The variance of residuals (errors) is constant across all values of x.
- **Normality:** Residuals are normally distributed.

Use Cases

- Predicting sales based on advertising spend
- Estimating house prices from square footage
- Forecasting temperature from time of year

Fitting the Model (Ordinary Least Squares)

The best-fit line minimizes the sum of squared residuals (vertical distances between observed and predicted values).

Formulas

- **Slope:**

$$ b_1 = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sum (x_i - \overline{x})^2} $$

- **Intercept:**

$$ b_0 = \overline{y} - b_1 \overline{x} $$

Python Example

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Example data: Advertising spend (x) vs. Sales (y)
x = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]).reshape(-1, 1)
y = np.array([3, 4, 2, 5, 6, 7, 8, 8, 9, 10])

model = LinearRegression()
model.fit(x, y)

print(f"Intercept (b0): {model.intercept_:.2f}")
print(f"Slope (b1): {model.coef_[^24_0]:.2f}")

# Predict and plot
y_pred = model.predict(x)
plt.scatter(x, y, color='blue', label='Actual')
plt.plot(x, y_pred, color='red', label='Fitted Line')
plt.xlabel('Advertising Spend')
plt.ylabel('Sales')
plt.legend()
plt.show()
```

Interpretation

- The **slope** tells you how much the dependent variable changes for each unit increase in the independent variable.
- The **intercept** is the expected value of y when $x = 0$.

Practice Problem:

Problem 1: Simple Linear Regression

Scenario

Suppose you want to predict a student's final exam score (y) based on the number of hours they studied (x). You collect data from 8 students:

Hours Studied (x)	Exam Score (y)
2	65

Hours Studied (x)	Exam Score (y)
3	70
5	75
7	80
8	85
10	88
12	95
14	100

Step 1: Fit the Simple Linear Regression Model

The regression equation is:

$$ \hat{y} = b_0 + b_1 x $$

Python Code

```
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# Data
x = np.array([2, 3, 5, 7, 8, 10, 12, 14]).reshape(-1, 1)
y = np.array([65, 70, 75, 80, 85, 88, 95, 100])

# Fit model
model = LinearRegression()
model.fit(x, y)

# Get coefficients
intercept = model.intercept_
slope = model.coef_[^25_0]
print(f"Intercept (b0): {intercept:.2f}")
print(f"Slope (b1): {slope:.2f}")

# Predict and plot
y_pred = model.predict(x)
plt.scatter(x, y, color='blue', label='Actual')
plt.plot(x, y_pred, color='red', label='Fitted Line')
plt.xlabel('Hours Studied')
plt.ylabel('Exam Score')
plt.legend()
plt.show()
```

Step 2: Interpret the Results

- **Slope (b_1)**: For each additional hour studied, the exam score increases by about b_1 points.
- **Intercept (b_0)**: The predicted exam score for 0 hours studied.

Step 3: Make a Prediction

Suppose a student studies 9 hours. What is their predicted score?

$$ \hat{y} = b_0 + b_1 \times 9 $$

Plug in the values from your model to get the answer.

Sales Example: Regression Analysis with R-squared and Metrics

Let's use the sales data from our previous example (sales with and without advertisement) to perform a simple linear regression. We'll calculate the regression coefficients, R-squared value, and other common metrics to evaluate the model's performance.

Step 1: Data Setup

Suppose you have weekly sales and corresponding advertising spend:

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error,
r2_score
import matplotlib.pyplot as plt

# Example data: Advertising spend (X) vs. Sales (Y)
ad_spend = np.array([10, 12, 15, 18, 20, 22, 25, 28, 30, 32]).reshape(-1,
1)
sales = np.array([200, 210, 220, 230, 240, 250, 265, 270, 280, 290])
```

Step 2: Fit the Regression Model

```
# Fit linear regression model
model = LinearRegression()
model.fit(ad_spend, sales)

# Get coefficients
intercept = model.intercept_
slope = model.coef_[^26_0]
print(f"Intercept: {intercept:.2f}")
print(f"Slope: {slope:.2f}")
```

Step 3: Make Predictions and Calculate Metrics

```
# Predict sales
sales_pred = model.predict(ad_spend)

# R-squared value
r2 = r2_score(sales, sales_pred)
print(f"R-squared: {r2:.3f}")

# Mean Squared Error (MSE)
mse = mean_squared_error(sales, sales_pred)
print(f"Mean Squared Error: {mse:.2f}")

# Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)
print(f"Root Mean Squared Error: {rmse:.2f}")

# Mean Absolute Error (MAE)
mae = mean_absolute_error(sales, sales_pred)
print(f"Mean Absolute Error: {mae:.2f}")
```

Step 4: Plot the Results

```
plt.scatter(ad_spend, sales, color='blue', label='Actual Sales')
plt.plot(ad_spend, sales_pred, color='red', label='Regression Line')
plt.xlabel('Advertising Spend')
plt.ylabel('Sales')
plt.legend()
plt.title('Sales vs. Advertising Spend')
plt.show()
```

Step 5: Interpretation

- **Intercept:** Expected sales when advertising spend is zero.
- **Slope:** Increase in sales for each additional unit of advertising spend.
- **R-squared:** Proportion of variance in sales explained by advertising spend (closer to 1 means a better fit).
- **MSE, RMSE, MAE:** Lower values indicate better model performance.

Would you like to try this with your own data, or see how to interpret these metrics in a business context?

Interpreting Regression Metrics in Business Context

Understanding regression metrics is crucial for translating model results into actionable business decisions. Here's how to interpret the most common metrics and their real-world significance, especially in a sales

scenario:

Step 1: R-squared (R^2)

- **Definition:** Proportion of variance in the dependent variable (e.g., sales) explained by the independent variable(s) (e.g., advertising spend).
- **Business Meaning:** If $R^2 = 0.80$, then 80% of the variation in sales is explained by your model. A higher R^2 means your predictions are more reliable for decision-making.[^27_2][^27_3][^27_8]
- **Quantification:** Use R^2 to judge if your model is good enough for forecasting or strategic planning. For example, if R^2 is low, you may need more features or a different model.

Step 2: Mean Absolute Error (MAE)

- **Definition:** The average absolute difference between predicted and actual sales.
- **Business Meaning:** If MAE is $5,000, your sales predictions are off by $5,000 on average.[^27_3][^27_2]
- **Quantification:** Directly tells you the typical error in your forecasts. Lower MAE means more accurate predictions, which helps in budgeting and inventory planning.

Step 3: Mean Squared Error (MSE) & Root Mean Squared Error (RMSE)

- **Definition:** MSE is the average squared error; RMSE is its square root, giving error in original units (e.g., dollars).
- **Business Meaning:** RMSE of $7,000 means your predictions typically deviate from actual sales by $7,000.[^27_2][^27_3]
- **Quantification:** RMSE is sensitive to large errors, so it highlights big mistakes. Use it to assess risk in high-stakes decisions.

Step 4: Regression Coefficients (Intercept & Slope)

- **Intercept:** Predicted sales when advertising spend is zero. Useful for baseline planning.
- **Slope:** Change in sales for each additional unit of advertising spend. Helps quantify ROI of marketing campaigns.[^27_5][^27_6][^27_8]

Step 5: P-value (for coefficients)

- **Definition:** Tests if the relationship between advertising and sales is statistically significant.
- **Business Meaning:** If p-value < 0.05, you can be confident that advertising truly impacts sales, not just by chance.[^27_8]
- **Quantification:** Use p-values to decide which factors to invest in or drop from your strategy.

Step 6: Physical Significance & Quantification in Real World

- **R-squared:** Tells you how much of your sales variability is explained by your model. High R^2 means you can trust the model for planning; low R^2 means you need to improve it.
- **MAE/RMSE:** Directly measure the average error in your predictions. If your business can tolerate a $5,000 error, and your MAE is $4,000, your model is acceptable. If not, you need to refine it.
- **Coefficients:** Quantify the impact of each variable. For example, if the slope is 2,000, every $1,000 increase in ad spend is expected to boost sales by $2,000.

- **P-value:** Ensures your decisions are based on statistically significant relationships, not random noise.

Step 7: Summary Table

Metric	What It Means in Business	How to Use It
R-squared	% of sales explained by model	Trustworthiness of forecasts
MAE	Avg. prediction error ($)	Budgeting, risk assessment
RMSE	Typical error ($)	Spot large mistakes
Slope	Sales increase per ad spend unit	ROI calculation
Intercept	Baseline sales (no ads)	Minimum expected sales
P-value	Statistical significance	Decision confidence

In summary: These metrics help you quantify model accuracy, understand business impact, and make informed decisions. Always relate the numbers back to your business goals and risk tolerance.

If you want, we can walk through a real sales dataset and interpret these metrics together, or discuss how to improve them for your use case.

Interpreting Regression Metrics for Business Impact

Let's address your two contexts for using regression metrics:

Step 1: Assuming the Model is Very Good: Real-World Business Impact

When your regression model is strong (high R-squared, low error metrics, significant coefficients), here's how to interpret the metrics for business decisions:

- **R-squared (R^2):**
  - *Business meaning:* If R^2 is 0.85, your model explains 85% of the variation in sales based on predictors like advertising spend. This means you can confidently use the model for forecasting, budgeting, and strategic planning.[^28_2][^28_3][^28_5]
  - *Impact:* High R^2 supports data-driven decisions, such as allocating marketing budgets or predicting inventory needs.
- **Regression Coefficients (Slope, Intercept):**
  - *Business meaning:* The slope quantifies how much sales increase for each unit of ad spend. If the slope is 2,000, every $1,000 increase in ad spend is expected to boost sales by $2,000. [^28_3][^28_5][^28_2]
  - *Impact:* Use this to estimate ROI and justify marketing investments.
- **Error Metrics (MAE, RMSE):**
  - *Business meaning:* If RMSE is $5,000, your sales predictions are typically off by $5,000. If this is within your business's risk tolerance, the model is actionable.[^28_4][^28_6]
  - *Impact:* Helps set realistic expectations for forecast accuracy and manage risk.
- **P-values:**

- *Business meaning:* Low p-values (<0.05) for coefficients mean those predictors have a statistically significant impact on sales.[^28_2][^28_3]
- *Impact:* Focus resources on significant drivers; ignore or remove non-significant ones.

Step 2: How Metrics Indicate Model Quality & Inform Improvements

- **R-squared (R^2):**
  - *High R^2 (e.g., >0.7):* Model fits data well; most variance is explained.
  - *Low R^2 (e.g., <0.5):* Model misses key patterns; consider adding more features, using non-linear models, or improving data quality.[^28_8][^28_2]
- **Error Metrics (MAE, RMSE):**
  - *High errors:* Model predictions are far from actuals. Try feature engineering, outlier handling, or more complex models.[^28_6][^28_4][^28_8]
  - *Low errors:* Model is precise; predictions are close to reality.
- **P-values:**
  - *High p-values (>0.05):* Predictor may not be useful; consider removing it to simplify the model.[^28_3][^28_2]
  - *Low p-values:* Predictor is important; keep it.
- **Bias (Mean Bias Error):**
  - *Consistent over- or under-prediction:* Indicates systematic error; may need to adjust model or include more relevant variables.[^28_1]
- **Adjusted R-squared:**
  - *Use when multiple predictors are present.* It penalizes for adding irrelevant features, giving a more honest assessment of model quality.[^28_9][^28_2]
- **Model Accuracy (Overall):**
  - *For regression, accuracy is not a direct metric like in classification.* Instead, use R^2, RMSE, and MAE to judge how well predictions match actuals.[^28_4][^28_8]

Step 3: How to Improve Model Performance

- Add more relevant features (e.g., seasonality, promotions).
- Remove or transform features with high p-values or low impact.
- Try non-linear models if relationships are not linear.
- Address data quality issues (missing values, outliers).
- Use cross-validation to ensure model generalizes well.

Summary Table: Business Interpretation of Metrics

Metric	High Value Means	Low Value Means	Business Action
R-squared	Good fit, reliable model	Poor fit, missing factors	Use for planning, or improve
MAE/RMSE	Precise predictions	Large errors	Trust forecasts, or refine
Slope	Strong impact per unit	Weak impact	Invest in driver, or reassess
P-value	Statistically significant	Not significant	Focus on key drivers

Metric	High Value Means	Low Value Means	Business Action
Adjusted R^2	Honest fit with many features	Overfitting risk	Feature selection

In summary:

- High R^2, low errors, and significant p-values mean your model is ready for business impact—use it for forecasting, budgeting, and strategic decisions.
- Low R^2, high errors, or non-significant predictors mean you should refine your model by adding features, improving data, or trying new algorithms.

If you want, we can walk through a real dataset and interpret these metrics together, or discuss how to communicate these results to stakeholders.

Using p-Values for Feature Selection in Regression

Can p-values be used for feature selection?

Yes, p-values from regression analysis can be used to help select features. The p-value for each feature tests the null hypothesis that the feature's coefficient is zero (i.e., the feature has no relationship with the target variable). A low p-value (typically < 0.05) suggests the feature is statistically significant and likely contributes to the model.[^29_1][^29_5][^29_10]

How is this done in practice?

- **Stepwise Selection:** Features are added or removed from the model based on their p-values. Common thresholds are 0.05 or 0.1. Features with high p-values are considered for removal, as their relationship with the target may be due to chance.[^29_5]
- **Manual Filtering:** After fitting a regression model, you can examine the p-values for each feature and drop those with p-values above your chosen threshold (e.g., 0.05 or 0.1).[^29_1]

Physical Significance

- **Low p-value (< 0.05):** Strong evidence that the feature is related to the target variable. Keep these features.
- **High p-value (> 0.05):** Weak evidence; the feature may be noise. Consider removing these features to simplify the model and reduce overfitting.[^29_10][^29_1]

Limitations and Cautions

- **Multicollinearity:** If features are highly correlated, p-values can be misleading. A feature may have a high p-value not because it's unimportant, but because its effect is shared with another feature.
- **Model Type:** p-values are most meaningful in linear regression and generalized linear models. For tree-based models, other feature importance metrics are preferred.
- **Business Context:** Sometimes, features with high p-values are kept for interpretability or domain reasons, even if not statistically significant.

Best Practices

- Use p-values as one tool among many for feature selection.
- Combine with domain knowledge, regularization methods (Lasso, Ridge), and cross-validation for robust feature selection.[^29_3][^29_4]
- Always check for multicollinearity and consider the business impact of removing features.

Python Example

```
import statsmodels.api as sm
import pandas as pd

# Example data
X = pd.DataFrame({'ad_spend': [10, 12, 15, 18, 20, 22, 25, 28, 30, 32],
                  'seasonality': [1, 2, 1, 2, 1, 2, 1, 2, 1, 2]})
y = [200, 210, 220, 230, 240, 250, 265, 270, 280, 290]

X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
print(model.summary())
# Look at p-values in the output to decide which features to keep
```

In summary: p-values are a useful guide for feature selection in regression, but should be used alongside other methods and domain expertise to build robust, interpretable models.[^29_5][^29_10][^29_1]

-