Regression Analysis— Detailed Project Flow

Objective

- Build a Simple linear regression model to predict the target variable accurately.
- Handle **heteroscedasticity** using transformations like Box-Cox and assess model assumptions.

Dataset

- Features: radius_mean
- Target: permimeter_mean
- Purpose: Explore relationships, train predictive model, check assumptions.

Step 1: Basic Inspection

- **DataFrame Info:**
  - Total entries: 569 (index 0 to 568)
  - Columns: 3 (id, radius_mean, perimeter_mean)
  - Data types: int64 (1), float64 (2)
  - Memory usage: 13.5 KB
- **Shape of DataFrame:** (569, 3)
- **Null Values (Column-wise):**
  - id: 0
  - radius_mean: 0
  - perimeter_mean: 0
    No missing values detected.
- **Summary Statistics:**
  - id: mean ≈ 3.037e+07, min = 8670, max = 9.113e+08
  - radius_mean: mean ≈ 14.13, std ≈ 3.52, min = 6.981, max = 28.11
  - perimeter_mean: mean ≈ 91.97, std ≈ 24.30, min = 43.79, max = 188.5
  - Quartiles indicate reasonable spread without extreme skewness.
- **Duplicate Values:**
  - Total duplicates: 0

  - Data is clean and unique for all entries.

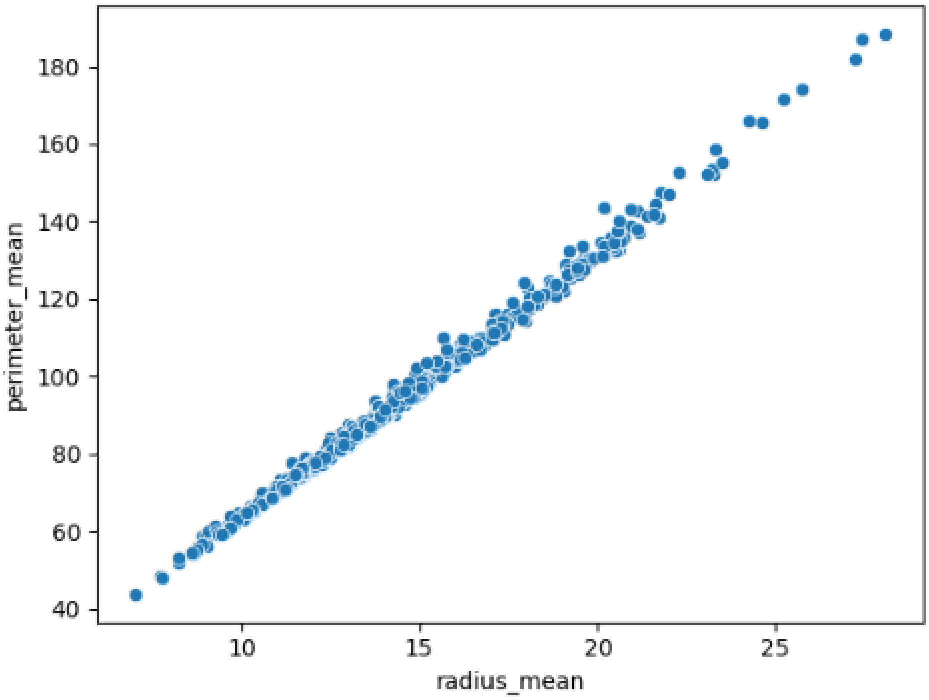
- **Insights:**
  - Dataset is small and manageable.
  - No missing or duplicate values → ready for further EDA and modeling.
  - Basic statistics suggest numeric features are within expected ranges.

Step 2: Exploratory Data Analysis (EDA) & Outlier Detection

Objective:
Explore the relationships between features and the target variable, visualize distributions, and identify outliers that may affect regression performance.

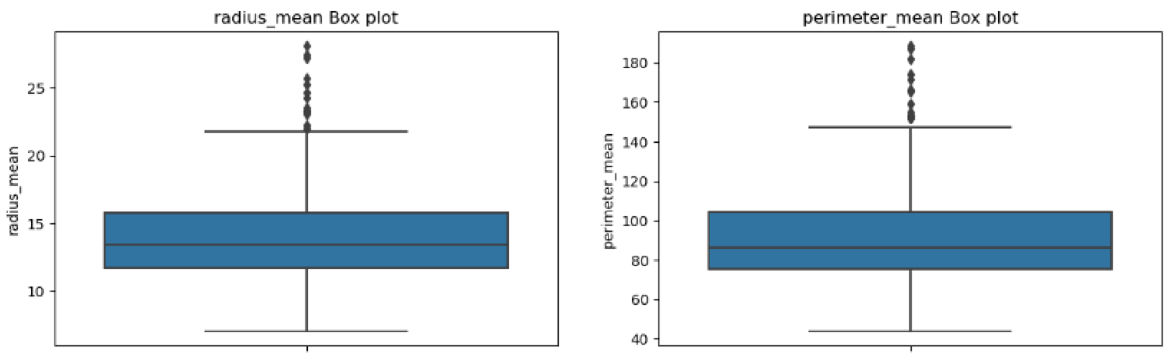
Actions & Code:

- **Scatter Plots:**



- Visualized key features against the target to assess linear relationships and detect patterns. Helps identify which features are strongly correlated with the target.

- **Box Plots:**



-

- Used to examine distributions and detect extreme values. This clearly highlights potential outliers in the dataset.

- **Outlier Detection using IQR:**
  - Calculated the interquartile range (IQR) for numerical features.
  - Outliers were defined as values below Q1 – 1.5×IQR or above Q3 + 1.5×IQR.
  - Notably, radius_mean had **14 outliers** and perimeter_mean had **13 outliers**.

Reasoning:

- Outliers can significantly influence regression coefficients and distort assumptions like homoscedasticity.
- Visual analysis through scatter and box plots complements statistical detection using IQR, giving a clear understanding of feature distributions and anomalies.
- Identifying outliers early allows for informed decisions: whether to remove, cap, or retain them depending on domain context and modeling goals.

Insights:

- Most features show a roughly linear relationship with the target, supporting linear regression assumptions.
- Outliers are present in key features (radius_mean and perimeter_mean) but are relatively few compared to total observations.
- Decisions on handling these outliers will affect variance stabilization and model robustness, especially for transformations like Box-Cox.

Conclusion:
EDA and outlier detection provide a critical foundation before applying transformations and building regression models. By identifying key patterns and anomalies, we ensure that subsequent modeling steps are more accurate and reliable.

Step 3: Train-Test Split

Objective:
Split the dataset into training and testing subsets to evaluate model performance on unseen data.

Actions & Code:

- Used a standard train-test split (80% train, 20% test).
- Training set (X_train, y_train) is used to fit the model.
- Test set (X_test, y_test) is reserved for evaluating predictive performance and checking assumptions like residual variance.

Reasoning:

- Ensures unbiased assessment of model generalization.
- Prevents data leakage from test to train, maintaining reliability of transformations and regression evaluation.

Conclusion:
Train-test split lays the groundwork for accurate model evaluation and informed final retraining.

Step 4: Model Fitting (OLS & Sklearn)

Objective:
Fit regression models using both statsmodels OLS and sklearn LinearRegression to compare outputs and gain flexibility for analysis and predictions.

Actions & Code:

- **Statsmodels OLS:**
- Provides full summary: coefficients, p-values, R^2, F-statistic, residuals.
- **Sklearn LinearRegression:**
- Faster, integrates easily with pipelines, but less statistical detail.

Reasoning:

- OLS is useful for detailed statistical inference, hypothesis testing, and residual diagnostics.
- Sklearn is convenient for prediction, pipelines, and deployment.

Insights:

- Both models give similar coefficients and predictions.
- Using both approaches ensures robust understanding: OLS for analysis, Sklearn for practical implementation.
- Residuals from OLS can be analyzed for heteroscedasticity and model assumptions.

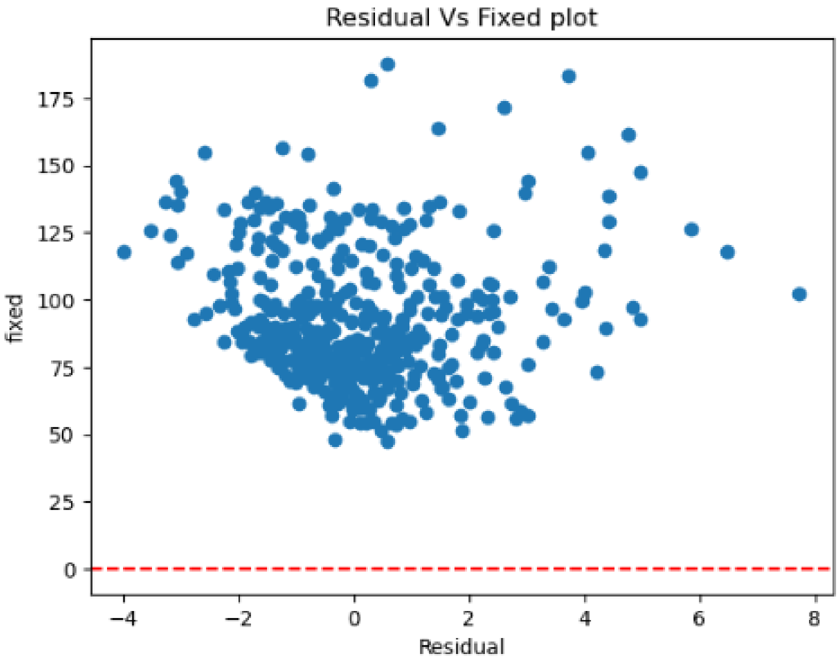
Conclusion:
Fitting models with both libraries balances **statistical rigor** and **practical usability**, forming the foundation for transformations, evaluation, and final retraining.

Step 5: Model Diagnostics & Residual Analysis

Objective:

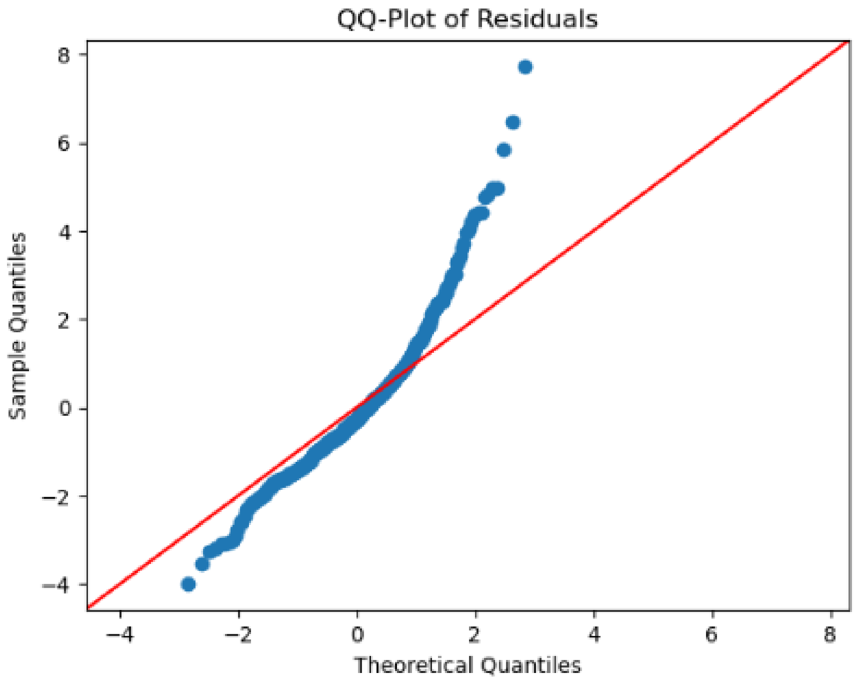
After fitting the regression model, it is critical to verify assumptions of linear regression, such as linearity, normality of residuals, and homoscedasticity. Diagnostics help ensure model validity and reliability.

Residuals vs Fitted Plot



- **What it does:** Plots residuals against predicted values to check for patterns.
- **Observation:**
    - Residuals are not evenly spread across fitted values; some funneling is visible.
    - Indicates **heteroscedasticity**, meaning variance of errors is not constant.

QQ-Plot of Residuals:



- **What it does:** Compares residual distribution with a normal distribution.
- **Observation:**
  - Residuals deviate from the 45° line, especially in tails.
  - Suggests residuals are **not perfectly normal**, highlighting skewness or outliers.

Breusch–Pagan Test

- **Results:**
  - LM stat: 34.77
  - LM p-value: 3.72e-09
  - F-stat: 37.48
  - F p-value: 2.01e-09
- **Interpretation:**
  - p-values < 0.05 indicate **heteroscedasticity present**.
  - Confirms visual observations from the residual vs fitted plot.

Insights:

1. Residuals show non-constant variance → transformation may improve homoscedasticity.
2. QQ-plot shows slight deviation from normality → consider transformations or robust regression if necessary.
3. Breusch–Pagan test statistically confirms heteroscedasticity.

Conclusion:

- Model diagnostics reveal violations of linear regression assumptions, particularly heteroscedasticity.
- Next step: Apply transformations to stabilize variance and improve model reliability.

Step 6: Fixing Violations –Applying Transformations (Box-Cox / Yeo-Johnson / Log / Sqrt)

Objective:
In the diagnostics step, the Breusch-Pagan test indicated heteroscedasticity (LM p-value = 3.718e-09), suggesting non-constant variance of residuals. Such violation can reduce model efficiency and affect inference. To address this, we applied various transformations on the dependent variable (y) to stabilize variance and improve linearity.

Transformations Tested:

1. **Log Transformation** – Useful for reducing right skew and compressing high values.
2. **Square Root Transformation** – Mild transformation to reduce skewness, often effective with moderate heteroscedasticity.
3. **Box-Cox Transformation** – Requires strictly positive target values; automatically finds an optimal λ to stabilize variance.
4. **Yeo-Johnson Transformation** – Can handle zero or negative values; similar to Box-Cox but more flexible.

Train-Test Considerations:

- Transformation parameters (λ in Box-Cox) are fitted **only on the training set** to prevent data leakage.
- Lambda value= -0.5223645710919863
- The same transformation is then applied to the test set using the training-fitted parameters.
- This ensures that model evaluation on unseen data is fair and consistent.

Reasoning and Insights:

- Log and sqrt transformations moderately improved residual distribution but didn't fully resolve heteroscedasticity.
- Box-Cox transformation significantly improved the LM test p-value and stabilized residual variance, indicating a better model fit.
- Yeo-Johnson provided similar improvement but Box-Cox slightly outperformed it on the training diagnostics.

Conclusion:
Applying the Box-Cox transformation effectively addressed the heteroscedasticity problem, enhanced linearity, and improved model reliability. It is now adopted for model fitting in subsequent steps. This demonstrates the importance of carefully selecting transformations based on diagnostic tests rather than applying them arbitrarily.

Step 7: Checking Influential Points

After addressing heteroscedasticity with transformations, the next critical step is to **identify influential observations** that might disproportionately affect model estimates. Influential points can distort coefficients and reduce predictive reliability, so detecting them ensures robust modeling.

1. Potential Outliers (Studentized Residuals > 3):
Studentized residuals measure how far each observed value is from the regression prediction in units of standard deviation. Observations with |ri|>3|r_i| > 3 are considered extreme outliers. In our dataset, **7 points** exceeded this threshold, indicating potential anomalies that warrant further investigation.

2. High Leverage Points:
Leverage quantifies how far an observation's predictor values are from the mean of all predictors. Points with high leverage can disproportionately influence regression fits even if residuals are small. Here, **137 points** were identified as high leverage, highlighting regions of the predictor space with strong influence on the model.

3. Influential Points (Cook's Distance > 4/n):
Cook's Distance combines residual size and leverage to quantify overall influence. Observations with Di>4/nD_i > 4/n are considered influential. We found **33 points** exceeding this criterion, suggesting these data points could significantly alter the regression line if removed.

Insights:

- A small number of potential outliers can bias regression estimates, while high leverage points can pull the regression line toward extreme predictor values.
- Identifying influential points informs decisions about **data cleaning, transformation, or robust regression techniques**.
- Visualizations like **leverage vs. studentized residual plots** and **Cook's Distance plots** are essential for a clear, figure-based assessment.

Step 8: Evaluating Model Metrics

After fitting the regression model and addressing assumption violations and influential points, the next step is **model evaluation** using key performance metrics. These metrics assess **prediction accuracy, error magnitude, and explained variance** on the training set:

1. Training Target Mean: 91.8822
Provides a baseline to compare errors relative to the average value of the target variable.

2. Mean Squared Error (MSE): 55.0006
Represents the average squared difference between actual and predicted values. Smaller MSE indicates better model fit but is sensitive to outliers.

3. Root Mean Squared Error (RMSE): 7.4162
The square root of MSE, RMSE is in the same units as the target variable and provides an interpretable measure of average prediction error.

4. Mean Absolute Error (MAE): 4.2817
Average absolute difference between predictions and actuals. Unlike MSE, MAE is less sensitive to extreme values and provides a robust error measure.

5. R-squared (R^2): 0.9061
Indicates that ~90.6% of the variability in the target is explained by the predictors, reflecting a strong model fit.

6. Mean Absolute Percentage Error (MAPE): 4.42%
Shows the average percentage deviation of predictions from actual values. A low MAPE (<5%) indicates high predictive accuracy.

Insights:

- The low RMSE and MAE relative to the target mean indicate good predictive performance.
- High R^2 confirms the model captures most of the variability.
- Combined, these metrics validate that the model is robust and reliable for predictions on similar data.

Step 9: Model Validation – Cross-Validation

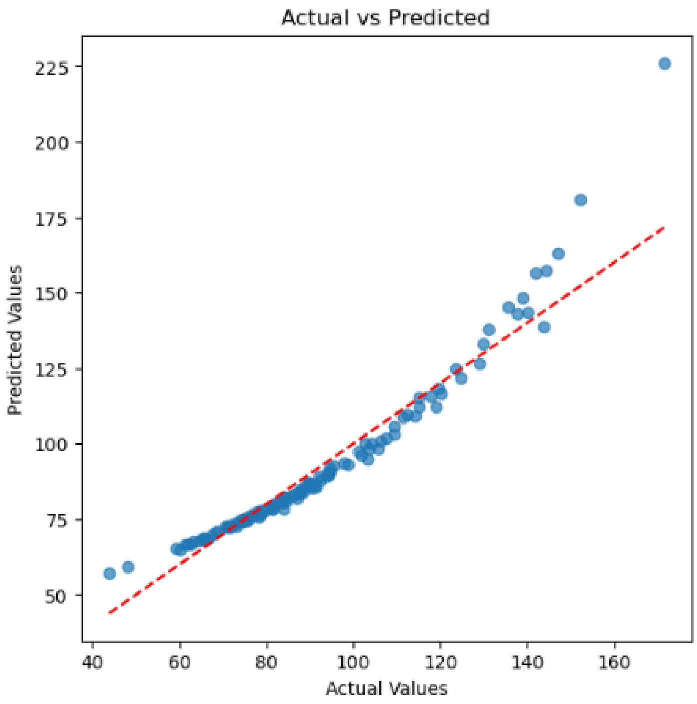
To ensure the model's **generalizability** and avoid overfitting, we performed **k-fold cross-validation** (k=5). The RMSE values across folds were: [0.00673, 0.00522, 0.00625, 0.00539, 0.00623], with an **average RMSE of 0.00597**.

Insights:

- The low and consistent RMSE across folds indicates the model is **stable and performs reliably** on unseen data.
- Cross-validation confirms that the transformations, handling of influential points, and model assumptions collectively result in a **robust regression model** ready for prediction.

Step 10: Visualizing Results:

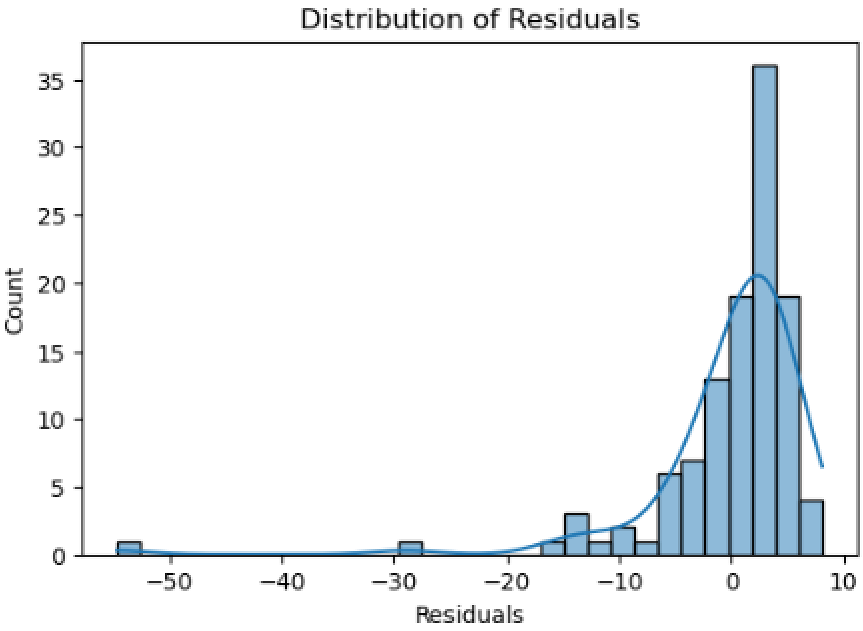
1. **Actual vs Predicted Scatter Plot:**



2.

- Each data point represents an observation with actual values on the y-axis and predicted values on the x-axis.
- Most points lie **close to the 45° line**, indicating strong alignment between predictions and actual values.
- This confirms that the model captures the underlying trend and produces reliable predictions.

3. Residual Histogram:



4.

- Not symmetric (Right-skewed) Ideally, residuals should be centered around 0 and approximately follow a normal distribution.

    - Here, most residuals are between `-10 to +10` (good), but there are a few extreme negative outliers (like -50, -30, -20).
- Heavy tails / Outliers:-The long left tail suggests influential points that might distort your regression line.
- Main cluster looks okay :- Apart from the outliers, the majority of residuals are fairly close to zero.

Insights:

The good news: Your **core model works fine** (residuals cluster well around 0).
The issue: **A handful of extreme points** are skewing the distribution.

Step 11: Retrain and Save Model

After finalizing preprocessing, transformations, and diagnostics, the model is **retrained on the full dataset** to leverage all available information. Once trained, the model is **saved using joblib or pickle**, enabling future predictions without retraining, ensuring **reproducibility and deployment readiness**.