Exercise 2: Feature engineering: Data encoding, scaling an Bias mitigation methods

Objective

In this exercise, you will explore the *House Prices: Advanced Regression Techniques* dataset from Kaggle and perform systematic feature engineering, including encoding, transformation, scaling, and bias mitigation. The goal is to understand how data characteristics influence model behaviour and to justify preprocessing choices using both statistical evidence and visual analysis.

Step 1: Environment Setup & Data Loading

1. Load the House Prices dataset using pd.read_csv().
2. Inspect the dataset structure using .info() and .head().

Step 2: Feature Type Separation (numerical and categorical)

1. Separate the features into: Numerical features and Categorical features
2. Create two Python lists: numerical_features and categorical_features

Deliverables:

- Code showing feature separation
- Number of numerical vs categorical features
- Brief justification of why this separation is important

Step 3: Numerical Correlation Analysis

1. Compute the correlation matrix between numerical features and the target variable 'SalePrice'.
2. Visualize correlations using a heatmap.
3. Focus on the relationship between numerical features and 'SalePrice'.

Deliverables:

- Correlation heatmap
- 3 key observations about highly correlated features

Step 4: Identify Top 10 Numerical Drivers

1. Compute the absolute correlation of each numerical feature with 'SalePrice'.
2. Exclude 'SalePrice' itself.
3. Identify the top 10 numerical features most correlated with the target.

Deliverables:

- Code to extract top 10 features
- Table listing feature name and correlation value

- Short explanation of why absolute correlation is used

Step 5: The "Hidden" Categorical Analysis

1. Examine the Top 10 numerical features identified earlier.
2. Identify any feature that was originally categorical but encoded numerically.
3. For that feature:
    a. Explain what type of variable it represents (ordinal / nominal)
    b. Plot the feature against 'SalePrice'.
    c. Comment on the implicit encoding used

Deliverables:

- Plot (boxplot / stripplot / barplot)
- Explanation of encoding type and its implications

Step 6: Inspect Pure Categorical Variables

1. Examine the remaining Categorical columns.
2. Check for Cardinality and think about the data encoding methods you have learned for the same

Step 7: Random Forest Feature Importance

1. Use a non-linear model (Random-forest) to find the "true" influencers.
2. Find top 10 features (from the categorical values) and plot them against target column. (you may use box plot, strip plots , violin plots etc). You can use the importance score to find which features are good.

Deliverables**:**

- Feature importance plot
- Short explanation of why Random Forest helps uncover non-linear effects

Step 8: Encoding Strategy and Rationale

1. Select the top 5 categorical features identified earlier.
2. For each feature:
    - Decide whether it is ordinal or nominal
    - Choose an appropriate encoding strategy
3. Apply the encoding and show the transformed data.

Step 9: Data Transformation and scaling

1. Find the Skew of numerical features (e.g., SalePrice, GrLivArea, TotalBsmtSF).
2. Apply appropriate transformations: Log, Box–Cox, Yeo–Johnson (where applicable)
3. Compare skewness before and after transformation.
4. Apply different scaling methods:

    - StandardScaler
    - MinMaxScaler
    - RobustScaler
    - MaxAbsScaler
5. Visually compare the effect of scaling and comment on which scaler to go for.

Deliverables**:**

- Skewness comparison table
- Distribution plots (before & after transformation)
- Scaled feature histograms
- Final recommendation of the most suitable scaler with justification

Step 10: Model Comparison – Impact of Feature Engineering on Linear Regression

1. Train a baseline Linear Regression model using:
    - Original numerical features
    - Basic encoding (if any)
    - No skew correction or advanced feature engineering
2. Train an improved Linear Regression model using:
    - Transformed features (Log / Box–Cox / Yeo–Johnson)
    - Appropriate scaling (e.g., StandardScaler)
3. Evaluate both models using suitable regression metrics:
    - R^2, RMSE or MAE
4. Compare:
    - Model performance
    - Sensitivity to outliers

Deliverables**:**

- Table comparing performance metrics of:
    - Baseline model
    - Feature-engineered model
- Brief commentary (5–6 sentences) explaining:
    - Which preprocessing steps contributed most to performance improvement
    - Why linear models benefit strongly from skew correction and scaling