Subjective Questions

Table of Contents

Assignment-based Subjective Questions

Question 1: From your analysis of the categorical variables from the dataset, what could you infer about their effect on the dependent variable?

1. Season: fall (3) has highest demand for rental bikes
2. Demand has grown in next year
3. Demand is continuously growing until June. Sept has the highest demand. After Sept, demand started decreasing
4. Demand is decreased on holidays
5. Weekday and working day is not giving any clear picture about demand
6. Weathersit clear has highest demand

Question 2: Why is it important to use drop_first=True during dummy variable creation?

Using `drop_first=True` when creating dummy variables is important to **avoid multicollinearity** in linear regression models. Here's a breakdown of why this matters:

Dummy Variables and Multicollinearity

When you convert a categorical variable into dummy variables (one-hot encoding), each category is typically transformed into its own binary (0/1) column. For example, if you have a categorical variable for a season with four levels (`spring`, `summer`, `fall`, and `winter`), without `drop_first=True`, you would create four columns: one for each season.

However, this can lead to **perfect multicollinearity**, also known as the **dummy variable trap**. This happens because the dummy variables are perfectly correlated with each other—if you know the values of three of the dummy variables, you can perfectly predict the fourth.

Example:

For the `season` variable with four categories:

- `spring` = (0, 0, 0)
- `summer` = (1, 0, 0)
- `fall` = (0, 1, 0)
- `winter` = (0, 0, 1)

Here, the four dummy variables are perfectly collinear, meaning the sum of the four columns will always equal 1. This introduces **redundancy** in the model, making it impossible for the regression algorithm to compute unique parameter estimates for each variable.

Why `drop_first=True` Solves the Problem

By setting `drop_first=True`, you drop the first category (often the baseline category) from the dummy variables, which:

- **Avoids perfect multicollinearity**: Dropping one dummy variable removes the redundancy. Now, knowing the remaining variables allows you to infer the dropped category.
- **Retains model interpretability**: The dropped category becomes the reference or baseline category against which the remaining categories are compared.

Benefits of Using `drop_first=True`:

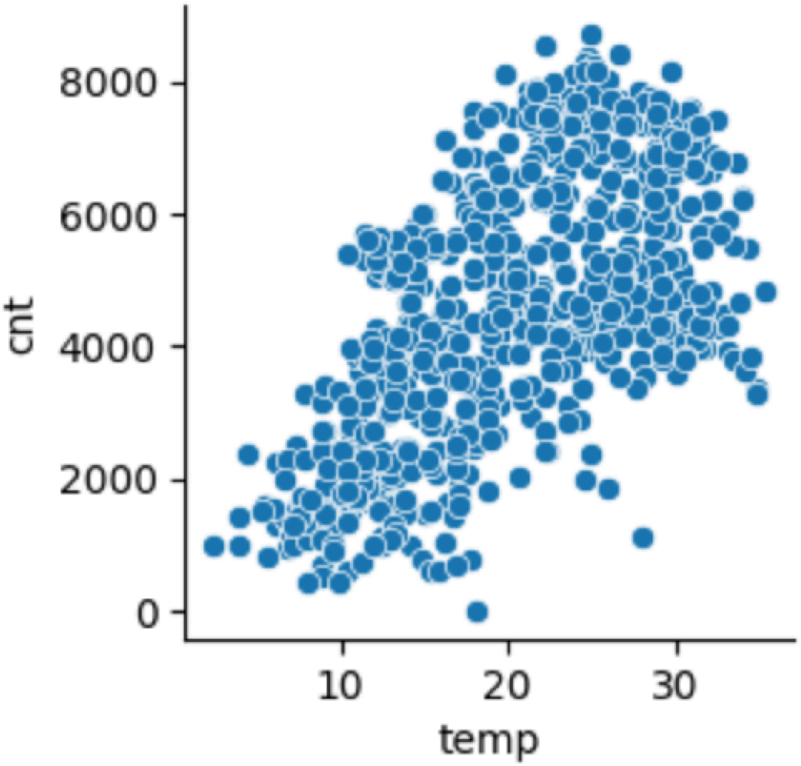
1. **Avoids the dummy variable trap**: Prevents perfect multicollinearity by removing redundancy.
2. **Improves model stability**: By avoiding multicollinearity, you ensure that the model coefficients are interpretable and statistically reliable.
3. **Interpretability**: The dropped category acts as a reference point, making it easier to understand the impact of the remaining categories on the dependent variable.

Key Takeaway:

In regression modeling, using `drop_first=True` when creating dummy variables is crucial to prevent multicollinearity and ensure that the model can make unique coefficient estimates.

Question 3: Looking at the pair-plot among the numerical variables, which one has the highest correlation with the target variable?

Variable `temp` has highest correlation with target variable cnt. Below is cropped screenshot from pairplot



Question 4: How did you validate the assumptions of Linear Regression after building the model on the training set?

I have validated the assumption of Linear Regression Model based on below 5 assumptions -

- Normality of error terms:
    - Error terms should be normally distributed
- Multicollinearity check:
    - There should be no multicollinearity among variables.
- Linear relationship validation:
    - Linearity should be visible among variables
- Homoscedasticity:
    - There should be no visible pattern in residual values.
- Independence of residuals:
    - No auto-correlation

Question 5: Based on the final model, which are the top 3 features contributing significantly towards explaining the demand of the shared bikes?

1. temp
2. season_spring
3. weathersit_lightsnow

General Subjective Questions

Question 1: Explain the linear regression algorithm in detail.

Linear Regression Algorithm:

Linear Regression is a fundamental and widely used algorithm in machine learning and statistics for predicting a continuous target variable based on one or more input features. It models the relationship between the dependent (target) variable and the independent (predictor) variables using a straight line.

Linear regression comes in two primary types:

- **Simple Linear Regression**: Involves one independent variable.
- **Multiple Linear Regression**: Involves multiple independent variables.

Goal:

The objective of linear regression is to find the best-fitting linear relationship between the input features (independent variables) and the output (dependent variable) such that the difference between the actual and predicted values (residuals) is minimized.

Equation of Linear Regression:

For **simple linear regression**, the model can be expressed as:

$y=b0+b1\cdot x$

Where:

- y = predicted value (dependent variable)
- b0 = intercept (the value of y when x=0)
- b1 = slope of the line (the rate of change of y with respect to x)
- x = input feature (independent variable)

For **multiple linear regression**, where there are multiple features:

$y=b0+b1\cdot x1+b2\cdot x2+\cdots+bn\cdot xn$ Where:

- y = predicted value (dependent variable)
- b0 = intercept
- b1,b2,…,bn = coefficients of the respective independent variables x1,x2,…,xn

How Linear Regression Works:

1. **Hypothesis**: Linear regression hypothesizes that there is a linear relationship between the dependent variable y and the independent variable(s) x. The model aims to find the values of b0 (intercept) and b1,b2,…,bn (slopes/coefficients) that define the best-fitting line.
2. **Cost Function (Mean Squared Error - MSE)**: The goal is to minimize the cost function, which measures the difference between the actual values and the predicted values. The most common cost function for linear regression is the **Mean Squared Error (MSE)**:

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Where:

- yi= actual value of the dependent variable
- y^i = predicted value
- n = number of data points

The cost function squares the differences between actual and predicted values to penalize large errors.

3. **Optimization (Gradient Descent or Normal Equation)**: To minimize the cost function and find the best values of the parameters b0,b1,…,bn we can use:

- **Gradient Descent**: An iterative optimization algorithm that adjusts the coefficients to minimize the cost function. The parameters are updated in the direction of the negative gradient of the cost function.
  The update rule for each coefficient bj is:

  $$b_j = b_j - \alpha \frac{\partial}{\partial b_j} \mathrm{MSE}$$

  -
  - where α is the learning rate that controls the step size of each update.
- **Normal Equation**: An analytical approach to directly compute the optimal values of the coefficients by solving the following equation:

  $$\mathbf{b} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

  -
  - Where:
    - X is the matrix of independent variables.
    - y is the vector of the dependent variable.

  The normal equation is computationally efficient for small datasets but can become inefficient for large datasets due to matrix inversion.

4. **Prediction**: Once the parameters are learned, the model can predict values of the target variable y for given values of the input variables x using the learned equation y=b0+b1⋅x1+⋯+bn⋅xn

5. **Evaluation**: To evaluate the model’s performance, the following metrics are commonly used:

- **R-squared (Coefficient of Determination)**: Indicates how well the independent variables explain the variance in the dependent variable. It ranges from 0 to 1, where 1 indicates a perfect fit.

  $$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

  - Where:
    - Sres = sum of squared residuals (errors)
    - Stot = total sum of squares
- **Adjusted R-squared**: Adjusts R-squared for the number of predictors in the model, providing a more reliable measure of model performance, especially in multiple regression.
- **Mean Absolute Error (MAE)**: The average absolute difference between actual and predicted values.
- **Root Mean Squared Error (RMSE)**: The square root of the mean squared error, giving more weight to large errors.

Assumptions of Linear Regression:

Linear regression relies on several key assumptions for it to work properly:

1. **Linearity**: The relationship between the independent variables and the dependent variable should be linear.
2. **Independence of Errors**: The residuals (errors) should be independent of each other. This means that the outcome of one observation should not influence the outcome of another.
3. **Homoscedasticity**: The variance of residuals should be constant across all levels of the independent variables.
4. **Normality of Residuals**: The residuals should follow a normal distribution.
5. **No Multicollinearity (for multiple regression)**: The independent variables should not be highly correlated with each other. Multicollinearity can cause problems in estimating the coefficients.

Overfitting and Underfitting in Linear Regression:

- **Overfitting**: Occurs when the model is too complex, such as having too many features or noise. The model fits the training data very well but fails to generalize to new, unseen data. This results in low training error but high test error.
- **Underfitting**: Occurs when the model is too simple and cannot capture the underlying patterns in the data. This leads to both high training and test errors.

Advantages of Linear Regression:

- **Simplicity**: Linear regression is easy to understand and interpret.
- **Efficiency**: It can be computed quickly, even for large datasets.
- **Linearity**: Works well when the relationship between variables is linear.

Disadvantages of Linear Regression:

- **Sensitive to outliers**: Outliers can have a large influence on the estimated coefficients.
- **Assumption of linearity**: Linear regression fails when the true relationship is non-linear.
- **Multicollinearity**: In multiple regression, if the independent variables are highly correlated, the model may become unstable.

Conclusion:

Linear regression is a fundamental technique for regression analysis, widely used for its simplicity and effectiveness. However, it has limitations, especially when assumptions are violated or when the relationships are non-linear, which can affect its predictive performance. Proper model evaluation and validation are essential to avoid overfitting or underfitting and ensure good generalization to unseen data.

Question 2: Explain the Anscombe's quartet in detail.

Anscombe's Quartet is a set of four datasets that have nearly identical summary statistics but reveal very different relationships when graphed. It was introduced by the statistician Francis Anscombe in 1973 to demonstrate the importance of data visualization and exploring datasets before making statistical inferences.

Despite having similar statistical properties, these datasets differ significantly when plotted, emphasizing the need for visual inspection of data rather than relying solely on summary statistics.

Key Takeaways from Anscombe’s Quartet:

Each dataset in Anscombe's Quartet shares the following properties:

- **Mean** of the x-values.
- **Mean** of the y-values.
- **Variance** of the x-values.
- **Variance** of the y-values.
- **Correlation** between x and y.
- **Linear regression line** (y = mx + c).
- **Coefficient of determination (R^2)** of the regression line.

However, the relationship between the x and y variables is very different in each dataset.

The summary statistics of the datasets give the impression that the relationships between x and y are similar in all four datasets. However, once these datasets are visualized, the differences become clear.

Why is Anscombe’s Quartet Important?

1. **Highlighting the Importance of Data Visualization**:
   - Anscombe’s Quartet shows that statistical summaries like means, variances, correlations, and regression coefficients don’t always capture the true nature of data. Visual inspection can reveal patterns, outliers, or non-linear relationships that statistics alone may miss.
2. **The Risks of Relying Solely on Summary Statistics**:
   - Just because datasets share similar statistics doesn’t mean they have the same data structure or relationship. Summary statistics can be manipulated or misunderstood if the underlying data patterns are not explored properly.
3. **Importance of Outliers**:
   - Outliers can heavily influence statistics like correlation and regression coefficients. Dataset III illustrates how a single outlier can affect the overall interpretation of the data.
4. **The Need for Model Validation**:
   - Simply running a regression analysis is not enough. It is essential to validate the assumptions of the model (e.g., linearity, independence, etc.) through visualization and diagnostic techniques to avoid misleading conclusions.

Conclusion:

Anscombe's Quartet is a classic example of why data visualization is a crucial step in any data analysis. While summary statistics like the mean, variance, correlation, and R-squared provide valuable insights, they don’t tell the full story. Data visualization helps detect underlying patterns, relationships, and anomalies that can otherwise go unnoticed, ensuring more accurate and meaningful interpretations of the data.

Question 3: What is Pearson’s R?

Pearson’s R (also known as Pearson’s correlation coefficient) is a statistical measure of the strength and direction of a linear relationship between two continuous variables. It is a widely used correlation metric, and its value ranges between -1 and 1.

Key Points about Pearson’s R:

1. **Value Range:**
    - +1: Perfect positive linear relationship (as one variable increases, the other increases in a perfectly linear manner).
    - -1: Perfect negative linear relationship (as one variable increases, the other decreases in a perfectly linear manner).
    - 0: No linear relationship between the variables.
2. **Formula:**

$$r = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum (X - \bar{X})^2 \sum (Y - \bar{Y})^2}}$$

    -
    - X and Y are the variables.
    - Xˉ and Yˉ are the means of variables X and Y, respectively.
3. **Interpretation**:
    - **Positive values** (0 < R ≤ 1): As one variable increases, the other tends to increase (positive linear relationship).
    - **Negative values** (-1 ≤ R < 0): As one variable increases, the other tends to decrease (negative linear relationship).
    - **Closer to 0**: Indicates a weaker linear relationship. When RRR is close to 0, it suggests little or no linear correlation between the variables.
    - **Closer to ±1**: Indicates a stronger linear relationship.
4. **Assumptions**:
    - Pearson’s R assumes that the relationship between the two variables is **linear**.

   - It also assumes that the variables are normally distributed (or at least not extremely skewed).
   - It is sensitive to **outliers**, which can significantly affect the value.
5. **Use Case**:
   - Pearson’s R is commonly used to measure the **strength** of the relationship between two variables. For example, it’s used in **correlation analysis** to see how changes in one variable are associated with changes in another.
   - Example: The relationship between **height and weight**, or between **study hours and exam performance**.

Difference from R-squared:

- **Pearson’s R** measures the **strength** and **direction** of a linear relationship between two variables.

Question 4: What is scaling? Why is scaling performed? What is the difference between normalized scaling and standardized scaling?

Scaling is the process of transforming the features of a dataset to a specific range or distribution. It is typically performed when working with machine learning algorithms to ensure that all features contribute equally and no single feature dominates due to its scale. Scaling improves the efficiency, performance, and convergence rate of various models, especially those that rely on distance measurements or gradient descent optimization.

Why is Scaling Performed?

1. **Models Sensitive to Feature Scale:**
   - Algorithms like linear regression, logistic regression, k-nearest neighbors (KNN), support vector machines (SVM), and neural networks are sensitive to the scale of features because they use distance-based metrics or gradients.
   - For example, in KNN or SVM, features with larger scales (like age in years or income in thousands) can dominate features with smaller scales, leading to biased predictions**.**
2. **Gradient Descent Convergence:**
   - In optimization algorithms like gradient descent, unscaled features can cause the algorithm to oscillate or converge more slowly, making it less efficient.

3. **Improves Interpretability and Comparability:**
    - Scaling makes it easier to compare coefficients and feature importance in models like linear regression by putting them on a comparable scale.

Difference Between Standardized Scaling and Normalized Scaling:

1. Standardized Scaling (Z-score normalization):

- **Definition**: Standardization transforms the data to have a mean of 0 and a standard deviation of 1.
- **Formula**:

$$z = \frac{x - \mu}{\sigma}$$

    -
    - x: Original data point
    - μ\muμ: Mean of the data
    - σ\sigmaσ: Standard deviation of the data
- **Result**: The transformed data will have a mean of 0 and a standard deviation of 1.
- **Use Cases**:
    - Works well with models that assume normality in the data or are sensitive to extreme values.
    - Suitable for algorithms like **linear regression**, **logistic regression**, **SVM**, and **K-Means clustering**.
- **Example**: If you have a feature like height with mean 160 cm and standard deviation 10 cm, standardizing would convert it to values that are centered around 0 and scaled relative to the variability of the data.

2. Normalized Scaling (Min-Max scaling):

- **Definition**: Normalization transforms the data to fit within a specified range, usually between 0 and 1.
- **Formula**:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

    -
    - x: Original data point

    - xmin: Minimum value in the dataset
    - xmax: Maximum value in the dataset
- **Result**: The transformed data will range between the specified range (commonly 0 and 1, but it could be [−1, 1] or another range).
- **Use Cases**:
    - Best for algorithms that do not assume normality and need data on a common scale.
    - Algorithms like **KNN**, **neural networks**, and **principal component analysis (PCA)** perform better with normalized data, especially when the dataset contains features with varying units (like price and weight).
- **Example**: If a feature (e.g., age) has values ranging from 20 to 60, normalization would map all ages between 0 and 1, with 20 mapped to 0 and 60 mapped to 1.

When to Use:

- **Standardization** is typically used when the algorithm expects data to follow a normal distribution or when feature variance is important (e.g., in **PCA**, **linear models**, **SVM**).
- **Normalization** is preferred when the range of data is important, or the algorithm relies on distances (e.g., **KNN**, **neural networks**, **clustering**).

Both techniques improve model performance by ensuring that no one feature dominates due to its scale or magnitude.

Question 5: You might have observed that sometimes the value of VIF is infinite. Why does this happen?

An infinite Variance Inflation Factor (VIF) indicates perfect multicollinearity, which occurs when one predictor variable in a regression model is a perfect linear combination of other predictor variables. This situation makes it impossible to estimate unique regression coefficients because the model can't distinguish between the perfectly correlated variables.

Reasons for Infinite VIF:

1. **Perfect Multicollinearity:**
    - VIF quantifies how much a variable's variance is inflated due to multicollinearity. If a predictor variable can be exactly explained by a linear combination of one or more other predictors, the VIF becomes infinite.
    - Example: If X1=2×X2X_1 = 2 \times X_2X1=2×X2, then the two variables are perfectly collinear. Including both in the model results in an infinite VIF.

2. **Dummy Variable Trap:**
    - The dummy variable trap occurs when you include all categories of a categorical variable in the model without dropping one as a reference category. For example, if you have three categories (`A`, `B`, `C`), including dummy variables for all three will result in perfect multicollinearity.
    - Solution: Drop one category to avoid this trap.
3. **Highly Correlated Variables:**
    - Although high correlation among predictors doesn't always lead to an infinite VIF, when the correlation is exactly 1 (or −1), VIF becomes infinite. In practice, near-perfect multicollinearity will cause very high VIF values.

How to Handle Infinite VIF:

1. **Remove one of the collinear variables:** If two or more variables are perfectly correlated, remove one to avoid the multicollinearity issue.
2. **Combine or transform variables:** If the variables are highly related, consider combining them into a single variable (e.g., by summing or averaging) or using a dimensionality reduction technique like **Principal Component Analysis (PCA).**
3. **Check for dummy variable traps:** Always ensure that you've dropped one dummy variable to prevent perfect multicollinearity in models with categorical variables.

When VIF is infinite, it indicates that the model can't uniquely estimate regression coefficients for some variables because they convey redundant information.

Question 6: What is a Q-Q plot? Explain the use and importance of a Q-Q plot in linear regression.

A **Q-Q plot** (Quantile-Quantile plot) is a graphical tool used to compare the distribution of a dataset to a theoretical distribution, usually the **normal distribution**. It plots the quantiles of the sample data against the quantiles of a theoretical distribution, helping to assess whether the data follows that distribution.

In Linear Regression, Q-Q plots are often used to check for:

1. **Normality of Residuals**: In linear regression, one of the key assumptions is that the residuals (errors) are normally distributed. A Q-Q plot allows you to visually assess this assumption.
    - **X-axis**: The theoretical quantiles from the normal distribution.
    - **Y-axis**: The quantiles of the residuals from the regression model.

Interpretation of a Q-Q Plot:

- **If the residuals are normally distributed**, the points on the Q-Q plot will lie roughly along a **45-degree line** (the line of perfect normality).
- **Deviations from normality** will show as:
  - **Heavy tails** (data with more extreme values than expected): Points at the ends of the plot will deviate upward or downward from the line.
  - **Skewness**: If the data is skewed, the points will form a curve, deviating systematically from the 45-degree line.

Importance of a Q-Q Plot in Linear Regression:

1. **Testing Normality Assumption**: A key assumption of linear regression is that the residuals are normally distributed. Violations of this assumption can affect statistical tests like t-tests or F-tests, making them unreliable. A Q-Q plot helps to visually diagnose this assumption.
2. **Detecting Outliers**: Points that deviate significantly from the 45-degree line in a Q-Q plot indicate outliers or heavy-tailed distributions. Identifying these can guide corrective actions, such as transformation or handling of outliers.
3. **Model Adequacy**: A well-fitted linear regression model should have residuals that are approximately normal. If the Q-Q plot shows major deviations, it may suggest model issues, such as the need for a transformation of variables or a different modeling approach.

Example:

- **Ideal Q-Q plot**: If your residuals are normally distributed, the points on the Q-Q plot will align closely with the diagonal reference line.
- **Heavy-tailed residuals**: If there are more extreme values than expected, the points at the tails of the Q-Q plot will deviate significantly from the line.
- **Skewed residuals**: A curved pattern in the Q-Q plot indicates that the residuals are skewed, meaning the distribution is not symmetric.

Conclusion:

In linear regression, a Q-Q plot is a vital diagnostic tool for assessing the normality of residuals, a key assumption that affects the validity of statistical inferences in the model.