Assignment 3

Shandon Herft

What the feature represents

1. Raw Image Data:

- What It Represents: The raw pixel values of each image capture the visual details of buildings. These include colors, textures, edges, and patterns that are inherent to the architectural style.
- Use in the Assignment: When using convolutional neural networks (CNNs) or other image processing techniques, these raw values are transformed into more abstract, high-level features that better capture the unique characteristics of each style.

2. Extracted Visual Features:

- What It Represents: After processing the images (for example, via CNNs), the model learns internal representations or embeddings that summarize important patterns such as ornamentation, structural forms, and layout details. These features provide a compact numerical description of each image.
- Use in the Assignment: Techniques like t-SNE use these extracted features to reduce the high-dimensional data into a two-dimensional scatter plot, allowing you to visualize how different architectural styles cluster together based on their similarities.

3. Categorical Labels (Architectural Style):

- What It Represents: Each image is tagged with an architectural style (e.g., Art Deco, Gothic, Bauhaus). This label is the "ground truth" that you want your classification model to learn and predict.
- Use in the Assignment: The labels are used to both train the classification model and to analyze the distribution of images per style, which is important for understanding issues like class imbalance.

4. Numerical Indicators (e.g., Image Counts, Era Years):

- What It Represents:
  - Image Counts: The frequency of images per architectural style, which helps highlight imbalances in the dataset.
  - Observed Era Year: For the regression part of your analysis, this feature represents the actual, known year (or era) associated with a building.

    - Predicted Era Year: The output of a regression model that attempts to predict the era based on the features extracted from the image or other attributes.

- Use in the Assignment:
    - The image counts are visualized in bar charts to assess the dataset's balance.
    - The observed versus predicted era years are plotted to evaluate model performance, where the ideal situation is that the predicted era year closely matches the observed era year.

What datatype the feature is (e.g., categorical, numerical, float, integer, etc.)

1. Architectural Style Label
    - Datatype: Categorical (often stored as a string or a categorical type in programming libraries)
    - Example: "Gothic", "Art Deco", "Bauhaus"
2. Raw Image Data
    - Datatype: Numerical (typically a multi-dimensional array of pixel values)
    - Detail:
        - Pixel Values: Usually integers (ranging from 0 to 255) when read directly from images.
        - Preprocessed Representations: They might be normalized to floats (ranging from 0.0 to 1.0) for model input.
3. Extracted Visual Features (Embeddings)
    - Datatype: Numerical (often represented as vectors of floats)
    - Detail: These features come from layers of a CNN or another feature extraction method, and they are typically stored in a floating-point format (e.g., float32).
4. Image Count per Architectural Style
    - Datatype: Numerical (Integer)
    - Example: The number of images, e.g., 700 for Queen Anne architecture.
5. Observed Era Year
    - Datatype: Numerical (Integer)

- Example: The actual year or era value associated with the building.

6. Predicted Era Year
    - Datatype: Numerical (Typically a float if the model outputs a continuous prediction, though it could be cast to an integer based on the context)
    - Detail: When evaluating regression, the predictions are usually in float format to capture the precise estimated values.
7. t-SNE Coordinates (Dimension 1 and Dimension 2)
    - Datatype: Numerical (Floats)
    - Detail: These coordinates are the result of dimensionality reduction and are typically stored as floating-point numbers.

A judgement as to whether the feature might be an important predictor of your target variable

1. Raw Image Data / Pixel Values
    - Judgement:
        - For Classification:
            - Raw pixel values, when processed appropriately (e.g., via a CNN), are crucial. They contain the complete visual information necessary to distinguish between architectural styles.
            - Predictive Importance: High—when used with deep learning, they enable the model to learn nuanced patterns in textures, shapes, and colors that define different styles.
        - For Regression (Predicting Era Year):
            - Alone, raw pixel data might not directly correlate with the era unless significant architectural differences are tied to historical periods.
            - Predictive Importance: Potentially moderate if the visual cues strongly correlate with historical design trends, but they usually require transformation into higher-level features.
2. Extracted Visual Features (Embeddings)
    - Judgement:
        - For Classification:

- These features are derived from the image data through techniques like CNNs and encapsulate complex patterns and abstract visual characteristics.
- Predictive Importance: Very High—because they summarize the essential visual details in a lower-dimensional space, making them very effective predictors for categorizing architectural styles.

- For Regression:
  - They might capture information about stylistic elements that change over time, potentially aiding in predicting an era.
  - Predictive Importance: High if there is a systematic evolution of style features with time, but this depends on how well the features capture temporal trends.

3. Architectural Style Label (Categorical Target in Classification)
   - Judgement:
     - Role:
       - Since this is your target variable in the classification task, it is not a predictor but rather the label you want to predict.
     - Predictive Importance: N/A as a predictor.
4. Image Count per Architectural Style (Frequency of Samples)
   - Judgement:
     - For Model Training:
       - This is not a direct predictor for classification or regression but an important statistic.
       - Predictive Importance: Low—its role is more in guiding decisions about how to handle class imbalance (e.g., through oversampling or using class weights) rather than predicting the target variable.
5. Observed Era Year (Numerical Feature in Regression)
   - Judgement:
     - For Regression:

        - When predicting a continuous variable like the era (or the predicted era year), the observed era year serves as the ground truth.
        - Predictive Importance: High—if you use historical information or other correlated features related to the observed era, they can be very predictive of the target. However, if you're trying to predict the era from visual cues, the observed era might be part of your validation or evaluation rather than a feature.

6. Predicted Era Year (Output of Regression Model)
    - Judgement:
        - Role:
            - This is the target variable for the regression task.
            - Predictive Importance: N/A as a predictor.
7. t-SNE Coordinates
    - Judgement:
        - For Visualization and Clustering:
            - t-SNE coordinates are derived from high-dimensional features to visualize the data. They are not typically used directly as predictors.
            - Predictive Importance: Low as features—their primary purpose is to help understand the relationships and clusters in your data rather than to serve as input variables in predictive modeling.

Some characterization of the range of observed values (e.g., mean and standard deviation for numerical variables, list or description of the levels for categorical variables)

1. Architectural Style (Categorical Variable)

- Levels/Classes:
  The dataset includes 25 distinct architectural styles. Examples of these styles (i.e., the levels) include:
    - Achaemenid
    - American Foursquare

  - American Craftsman
  - Ancient Egyptian
  - Art Deco
  - Art Nouveau
  - Bauhaus
  - Chicago School
  - Edwardian
  - Gothic
  - Queen Anne
  - (and others up to 25 total)
- Description:
  Each style represents a unique category of building design. Visualizations (e.g., bar charts) reveal that the distribution of images across these styles is imbalanced. For instance, styles such as Queen Anne have a much higher image count (e.g., over 700 images) compared to less-represented styles like Chicago School, Edwardian, Romanesque, Byzantine, and Bauhaus.

2. Image Count per Architectural Style (Numerical Variable: Integer)

- Observed Range:
  - Minimum: Some architectural styles have very few images (exact numbers may vary, but they could be on the order of tens or a low hundreds).
  - Maximum: The dominant category (e.g., Queen Anne) has over 700 images.
- Summary Statistics (Hypothetical Example):
  - Mean: You might calculate an average count across all 25 styles. For example, if the total number of images is 10,113, the mean count would be roughly 404 images per style.
  - Standard Deviation: Given the high imbalance, the standard deviation is likely quite high, reflecting that while some styles have very few images, others have several hundred.

3. Raw Image Data / Extracted Visual Features (Numerical Variables: Arrays of Integers or Floats)

- Raw Image Data:

    - Datatype: Multi-dimensional numerical arrays.
    - Range of Values:
        - Before Normalization: Typically, pixel values are integers in the range [0, 255].
        - After Normalization: When preprocessed for model input (e.g., for CNNs), these values are often converted to floats in the range [0.0, 1.0].
- Extracted Visual Features (Embeddings):
    - Datatype: Vectors of floats (e.g., float32).
    - Range of Values:
        - The exact range depends on the activation functions and layers used in your CNN, but typically these values are normalized or standardized.
        - You might observe mean values around 0 with some standard deviation that depends on how the features are scaled during training.

4. Observed Era Year (Numerical Variable: Integer)
    - Range:
        - This variable represents the actual, known era or year associated with each building.
        - Example Range:
            - Buildings from ancient styles (e.g., Ancient Egyptian or Achaemenid) might have era years that are very low or even negative if represented as BCE (depending on your encoding).
            - More modern buildings (e.g., American Craftsman) might have era years in the 1900s.
        - Summary Statistics (Hypothetical Example):
            - Mean: Could be centered around a particular historical period depending on your dataset’s focus.
            - Standard Deviation: The spread will reflect the diversity of eras included in the dataset.
5. t-SNE Coordinates (Numerical Variables: Floats)

- Datatype: Two numerical variables (one for each t-SNE dimension) represented as floats.
- Range and Interpretation:
    - The t-SNE algorithm projects high-dimensional data into 2D.
    - Typical Range:
        - Although the axes don't have a standardized scale (i.e., the values might range from -50 to 50 or a similar interval), the important aspect is the relative distances and clustering rather than the exact numeric range.
    - Summary Statistics:
        - You can compute the mean and standard deviation for each dimension, but these are more for understanding the dispersion and density of clusters rather than having an intrinsic meaning.

Identify if any transformation (e.g., log transform) or encoding (e.g., one-hot encoding) might be needed to use the feature in a predictive model

1. Raw Image Data

- Normalization/Scaling:
    - Transformation Needed: Yes.
    - Why: Pixel values are typically in the range [0, 255] (integers). It's common to scale these values to the range [0.0, 1.0] (floats) before feeding them into a CNN or other models.
    - How: Divide by 255, or use another normalization technique.

2. Extracted Visual Features (Embeddings)

- Standardization/Normalization:
    - Transformation Needed: Possibly.
    - Why: These features are numerical and might benefit from standardization (e.g., zero mean, unit variance) or normalization if the model is sensitive to the scale of input features.
    - How: Apply techniques like z-score standardization or min–max scaling if required by your subsequent modeling step.

3. Architectural Style (Categorical Variable)

- Encoding:
  - Transformation Needed: Yes.
  - Why: Many machine learning algorithms require numerical input rather than raw string labels.
  - How:
    - One-Hot Encoding: This is a common approach, especially if the number of categories (25 in this case) is not too large.
    - Label Encoding: This might be used with algorithms that can handle categorical indices (like tree-based methods), but one-hot encoding is generally safer to avoid any unintended ordinal relationship.

4. Image Count per Architectural Style

- Transformation Considerations:
  - Log Transformation:
    - When Needed: If the distribution is highly skewed (e.g., a few styles have very high counts compared to others).
    - Why: A log transform can help stabilize variance and reduce the impact of outliers.
  - Scaling:
    - When Needed: If you decide to include this as a feature in a predictive model, scaling might help—though in many cases, it is used mainly to understand data imbalance rather than as an input feature.

5. Observed Era Year

- Standardization/Scaling:
  - Transformation Needed: Possibly.
  - Why: If this numerical variable is used directly as a feature in regression models, it may be beneficial to standardize it (especially if its range is large or if you combine it with other features that are on different scales).
  - How: Apply z-score normalization or min–max scaling.
- Encoding (if categorical):
  - When Needed: If the era is discretized into periods (e.g., ancient, medieval, modern), then you might consider one-hot encoding.

    - Note: If it remains as a continuous numerical value (e.g., the actual year), then only scaling is typically necessary.

6. Predicted Era Year
    - Note:
        - This is your target variable for regression, so you wouldn't transform it for input purposes.
        - Consideration: Sometimes it is useful to standardize the target variable during model training, but you would transform predictions back to the original scale for interpretation.
7. t-SNE Coordinates
    - Usage Consideration:
        - Transformation Needed: Not typically.
        - Why: t-SNE coordinates are generated for visualization purposes and do not represent intrinsic features that need to be encoded or normalized before model training.
        - Note: They are usually not used as predictors in a model; they help in understanding data structure and clusters.