**Student Name – Sachin Shrinivas Kadam**

**Assignment Name – Data Transformation**

* Implement data transformation techniques for encoding categorical variables.
* Implement data transformation techniques for scaling (normalisation, standardisation).
* Discuss how data transformation ensures the data meets model requirements without altering its inherent meaning.

My Analysis & Response as below

I will upload/attach my Python Code with this document.

In Machine Learning, handling categorical variables and scaling numerical features properly is essential for building an effective predictive model. Below I will explain **encoding techniques** for categorical features and **scaling methods** for numerical features, by providing justifications for each method used.

**Dataset Overview and Categorical Variables** **-**

* File Name – train.csv
* Number of Records – 891 rows of passenger
* Number of Columns – 12
* Sex – Male or Female
* Embarked - The port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)
* **Pclass**: The class of the passenger (1st, 2nd, 3rd)

Additionally, **Age**, **Fare**, **SibSp**, and **Parch** are Numerical Variables that may require scaling.

**Encoding Categorical Variables** **-** Categorical variables like **Sex** and **Embarked** cannot be used directly by ML models since most algorithms only work with numerical values. To address this, we need to apply encoding techniques.

* **One-Hot Encoding for 'Embarked' and 'Pclass'** - One-hot encoding is a method of converting categorical variables into a form that could be provided to ML algorithms to do a better job in prediction. It involves creating binary columns for each category.

Example - The **Embarked** column has 3 unique categories (C, Q, S). After applying one-hot encoding, it will create 3 new binary columns:

|  |  |  |
| --- | --- | --- |
| Embarked\_C | Embarked\_Q | Embarked\_S |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

Example - The **Pclass** column is already numeric (1, 2, 3), but sometimes it is treated as a categorical variable (ordinal) because the values represent distinct groups (classes). While **Pclass** doesn't require one-hot encoding, it can be kept as is, or if necessary, it can be transformed as a numerical feature.

* **Label Encoding for 'Sex' -** The **Sex** column contains two categories: **Male** and **Female**. Label encoding will convert these to numerical values (0 and 1)
  + Male – 0
  + Female – 1

Label encoding is suitable for binary categorical variables like **Sex** because it introduces a natural ordering, which doesn't harm the model performance in this case.

**Scaling Numerical Features -** Scaling numerical features is crucial for many machine learning models, especially those that are sensitive to the scale of input data (e.g., logistic regression, k-NN, SVMs). Common scaling techniques are **normalization** and **standardization**.

* **Standardization (Z-score scaling)** - Standardization transforms the features to have a mean of 0 and a standard deviation of 1. This is particularly important when features are on different scales (e.g., **Age** ranges from 0 to 100, but **Fare** can range from 0 to 500). Standardization is typically used when:
  + Features have different units or scales
  + The model is sensitive to the scale (e.g., linear models, SVM, k-NN).
  + Formula for Standardization
    - Standardized value =
      * x is a value
      * m is Mean
      * sd is the Standard Deviation of the feature
* **Normalization (Min-Max Scaling) -** Normalization scales the features to a range between 0 and 1. This is useful when the feature distribution is known to be bounded (e.g., a percentage between 0 and 100). It can be helpful when using models that rely on distance metrics (e.g., k-NN). Normalization is generally used when:
  + We want all features on the same scale for distance-based models (e.g., k-NN).
  + The features are known to have a specific range.
  + Formula for normalization:
    - Normalized value =
      * X is a value

**Recommended Scaling for this assignment -**

* In the train.csv, **Age**, **Fare**, **SibSp**, and **Parch** should be **Scaled**. Among these, **Age** and **Fare** have larger numeric ranges, so **Standardization** is preferable. **SibSp** and **Parch**, which are small integers, can be left as-is or **Normalized** if necessary.

**Justification for Encoding and Scaling Choices -**

* One-Hot Encoding for **Embarked** and **Pclass** -
  + **Justification**: The **Embarked** variable represents categorical data, and one-hot encoding is appropriate because it allows the model to treat each category as a separate feature without implying any ordinality. **Pclass** can be left as a numerical feature since it represents an ordinal relationship (higher class passengers had a higher chance of survival). If treating it as a categorical variable is preferred, one-hot encoding could also be applied.
* Label Encoding for **Sex**
  + **Justification**: The **Sex** variable is binary, so label encoding is the simplest and most efficient approach. It avoids unnecessary creation of new columns and works well for models that can handle ordinal relationships.
* Standardization for Numerical Variables (**Age**, **Fare**, **SibSp**, **Parch**)
  + **Justification**: **Standardization** is chosen for features with a wide range of values, like **Age** and **Fare**, which can have a large impact on models sensitive to scale. Standardization ensures that the models don't become biased towards higher-magnitude features and helps in converging faster.

**Impact on Model Performance -**

* **Without Scaling**: If we don’t scale the features, models like **logistic regression**, **k-NN**, or **SVMs** could become biased by features with large numerical ranges (e.g., **Fare**) leading to poor performance.
* **Without Encoding**: If categorical variables like **Sex** or **Embarked** aren’t properly encoded, machine learning models would not understand them and thus wouldn’t be able to make accurate predictions.
* **With Proper Scaling**: Scaling helps models that are sensitive to input magnitudes, especially those based on distance or gradient descent methods. It also reduces the risk of one feature dominating over others.

**Conclusion -**

* **Encoding Categorical Features**: One-hot encoding for multi-class categorical variables and label encoding for binary variables allows the model to understand categorical features.
* **Scaling Numerical Features**: Standardization is applied to numerical features like **Age** and **Fare** to ensure they are on a similar scale, preventing larger values from dominating the model.

Both encoding and scaling improve model performance by ensuring that categorical and numerical features are correctly represented for machine learning algorithms, enhancing convergence rates and predictive accuracy without altering the inherent meaning of the data.