# **Lab 8: Feature Engineering – Creating and Transforming Variables**

**Prelab Questions**

1. **What is feature engineering, and why is it important in data analysis?**

Feature engineering is the process of transforming raw data into meaningful features to improve machine learning model performance. It helps models capture patterns, reduces overfitting, enhances interpretability, and optimizes computational efficiency.

***Importance:***

* **Boosts Model Performance** – Helps models capture patterns for better accuracy.
* **Prevents Overfitting** – Reduces noise and improves generalization.
* **Handles Data Complexity** – Cleans and preprocesses messy data.
* **Enhances Interpretability** – Makes model decisions more understandable.
* **Optimizes Efficiency** – Reduces redundant features, speeding up training.

1. **What are derived features? Give an example.**

Derived features are new features created from existing data to improve model performance. These features are generated through mathematical transformations, aggregations, or domain-specific knowledge.

***Example:***

In a dataset with a "Date of Birth" column, you can derive a new feature **"Age"** by calculating:

Age = Current Year – Year of Birth

This transformation makes the data more useful for modeling.

1. **How can categorical variables be transformed for machine learning models?**

Categorical variables can be transformed into numerical format for machine learning models using the following techniques:

* **Label Encoding** – Assigns unique numbers (e.g., Red → 0, Blue → 1).
* **One-Hot Encoding** – Creates binary columns for each category.
* **Target Encoding** – Replaces categories with the mean of the target variable.
* **Frequency Encoding** – Encodes categories based on their occurrence count.
* **Embeddings** – Converts categories into continuous vectors (used in deep learning)

1. **Explain the difference between normalization and standardization.**

|  |  |  |
| --- | --- | --- |
| FEATURE | NORMALIZATION | STANDARDIZATION |
| DEFINITION | Rescales data to a fixed range (e.g., 0 to 1). | Centers data with a mean of 0 and standard deviation of |
| FORMULA | X’ = X-Xmin/Xmax-Xmin | X’ =X - μ / σ |
| EFFECT ON DATA | Compresses values within a specific range. | Preserves distribution but shifts and scales data. |
| SENSITIVITY TO OUTLIERS | Yes, because extreme values affect the range. | Less sensitive, as it considers mean and standard deviation. |
| WHEN TO USE | When you need a bounded scale (e.g., neural networks). | When data has varying scales but needs normal distribution (e.g., PCA, regression). |
| EXAMPLE | Image processing, deep learning. | Clustering, regression, principal component analysis (PCA). |

1. **Why is it necessary to handle skewed data before modelling?**

* **Improves Accuracy** – Many models perform better with normally distributed data.
* **Prevents Bias** – Skewness can distort predictions.
* **Balances Feature Importance** – Avoids dominance of certain values.
* **Helps Parametric Models** – Linear/logistic regression work best with symmetric data.
* **Reduces Outlier Impact** – Minimizes extreme value influence.

**Steps to be taken:**

* Log Transformation – log (x + 1)
* Square Root / Cube Root Transformation
* Box-Cox Transformation
* Binning (Discretization)

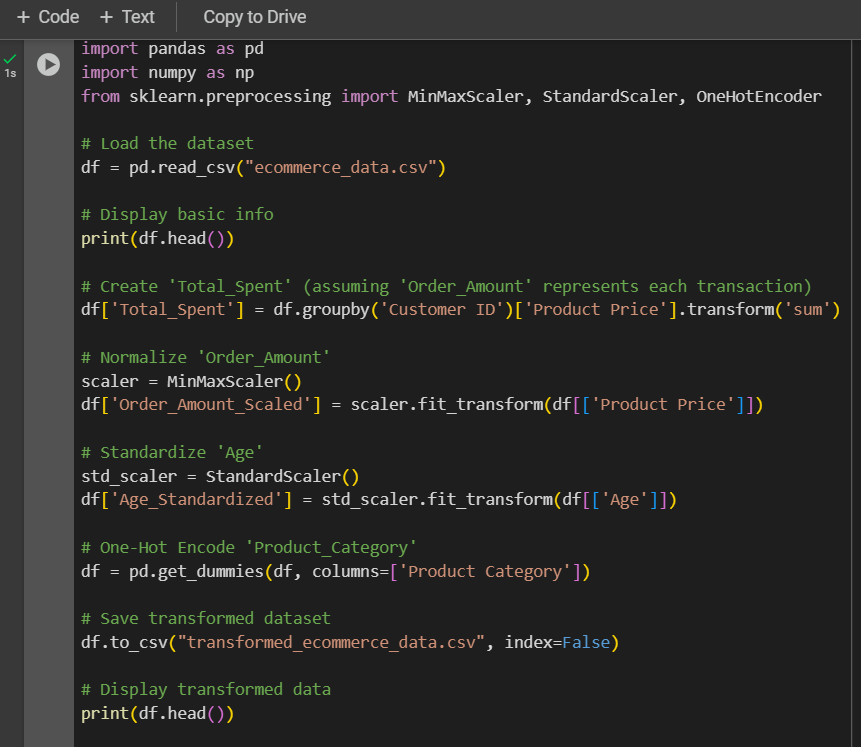
**In-Lab Details**

**Objective**:

* Learn how to create new features and transform existing ones to improve data representation.

**Resources**:

* Python (Jupyter Notebook).e
* Libraries: Pandas, NumPy, Scikit-learn.
* Dataset: ecommerce\_data.csv containing order amount, customer age, and product category.



Source Code :

import pandas as pd

import numpy as np

from sklearn.preprocessing import MinMaxScaler, StandardScaler, OneHotEncoder

# Load the dataset

df = pd.read\_csv("ecommerce\_data.csv")

# Display basic info

print(df.head())

# Create 'Total\_Spent' (assuming 'Order\_Amount' represents each transaction)

df['Total\_Spent'] = df.groupby('Customer ID')['Product Price'].transform('sum')

# Normalize 'Order\_Amount'

scaler = MinMaxScaler()

df['Order\_Amount\_Scaled'] = scaler.fit\_transform(df[['Product Price']])

# Standardize 'Age'

std\_scaler = StandardScaler()

df['Age\_Standardized'] = std\_scaler.fit\_transform(df[['Age']])

# One-Hot Encode 'Product\_Category'

df = pd.get\_dummies(df, columns=['Product Category'])

# Save transformed dataset

df.to\_csv("transformed\_ecommerce\_data.csv", index=False)

# Display transformed data

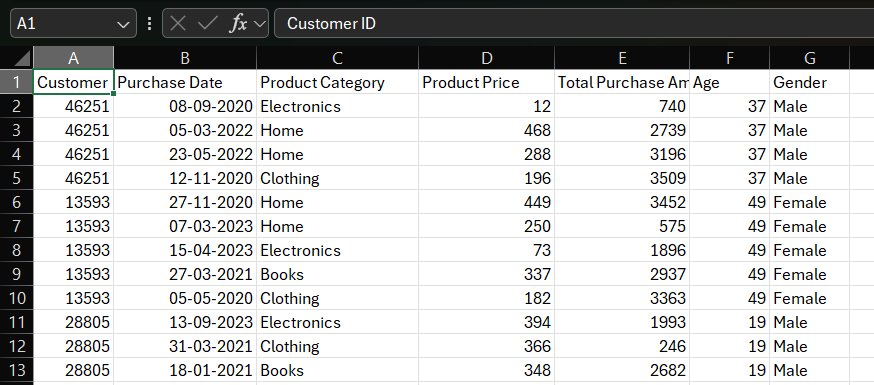
print(df.head())

OUTPUT:

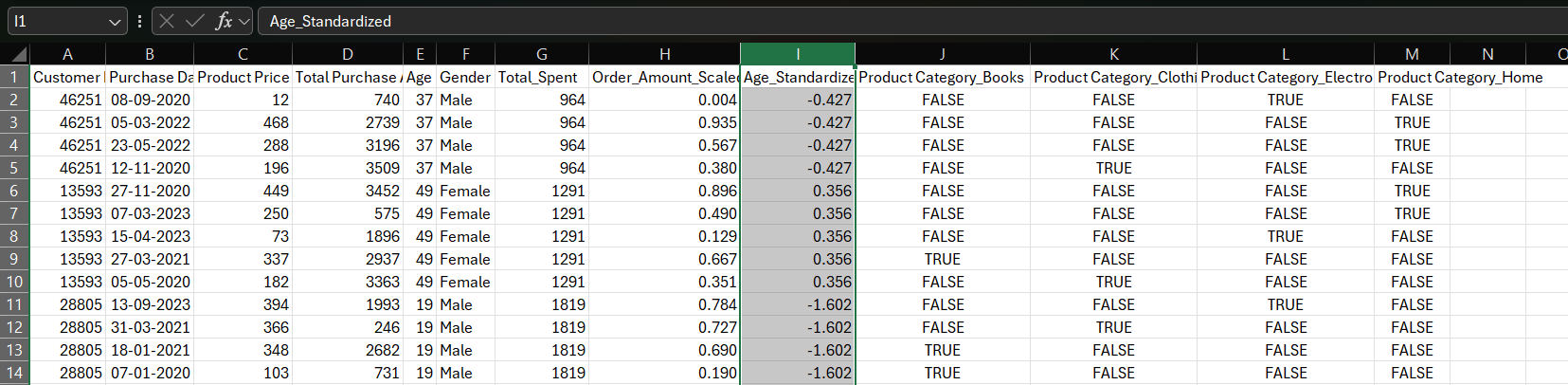
**Expected Output**

1. **New Feature: Total\_Spent**: Captures spending patterns.
2. **Normalized Order\_Amount\_Scaled**: Scaled between 0 and 1 for better model compatibility.
3. **Standardized Age\_Standardized**: Mean-centered age variable with unit variance.
4. **One-Hot Encoded Product\_Category**: Converts categorical data into numerical format.

Before transformation :



After transformation :



**Post lab Questions**

1. **What is the purpose of feature engineering in predictive modelling?**

**Purpose of Feature Engineering in Predictive Modelling:**

* Enhances model accuracy by creating more informative features.
* Helps in transforming raw data into a format suitable for ML models.
* Captures complex relationships that standard features may not represent.
* Reduces the risk of overfitting by removing redundant features.
* Improves model generalization to unseen data.

1. **Why is normalization essential when using distance-based models like KNN?**

**Importance of Normalization in Distance-Based Models (KNN):**

* Ensures fair comparison between features of different scales.
* Avoids bias where features with larger magnitude dominate distance calculations.
* Improves the stability of algorithms like KNN, K-Means, and SVM.
* Accelerates convergence in optimization algorithms.
* Reduces computational cost and improves distance-based decision boundaries.

1. **Compare one-hot encoding and label encoding for categorical variables.**

**Comparison of One-Hot Encoding and Label Encoding:**

* **One-Hot Encoding:**
  + Suitable for non-ordinal categorical variables.
  + Avoids introducing artificial ordinal relationships.
  + Increases feature space, which may lead to sparsity.
  + Can cause the curse of dimensionality with high cardinality.
* **Label Encoding:**
  + Useful for ordinal categorical variables.
  + More memory-efficient than one-hot encoding.
  + Can introduce unintended relationships between categories.
  + May lead to biased models if categorical values are misinterpreted as numerical order.

1. **How can feature selection improve model efficiency?**

**Feature Selection for Model Efficiency:**

* Helps reduce overfitting by removing irrelevant features.
* Improves training speed and reduces computational overhead.
* Enhances model interpretability by focusing on significant features.
* Reduces noise in the dataset, improving prediction accuracy.
* Prevents the curse of dimensionality, especially in high-dimensional data.
* Can improve performance for linear models where irrelevant features add noise.

1. **What are some common techniques for handling highly skewed features?**

**Techniques for Handling Highly Skewed Features:**

* **Transformation Methods:**
  + Log transformation (for right-skewed data).
  + Square root or cube root transformation.
  + Box-Cox or Yeo-Johnson transformation for flexible normalizations.
* **Binning and Bucketing:**
  + Converts continuous features into categorical bins.
  + Helps smooth out the effect of outliers.
* **Winsorization:**
  + Capping extreme values to reduce the impact of outliers.
* **Use of Robust Models:**
  + Tree-based models (e.g., Random Forest, XGBoost) handle skewed data better than linear models.
* **Outlier Detection and Removal:**
  + Identifying and treating extreme values to normalize distribution.
* **Using Non-Linear Algorithms:**
  + Some models (e.g., neural networks, decision trees) are less sensitive to skewed distributions.

DONE BY

Durga S – 22BCS026

Nandhini Priya K K -22BCS073

Nithiyaa T – 22BCS082

Priya Dharshini P – 22BCS094

Rithika S – 22BCS103

Sandhiya G – 22BCS109

Santhiya P – 22BCS111