**Batch: H – DA3 Roll No.: 16010122096**

**Experiment No. 1**

|  |
| --- |
| **Title: Implement data pre-processing using python on real world dataset** |

# Course Outcome:

# CO1 Understand basic concepts of data analytics to solve real-world problems

# Books/ Journals/ Websites referred:

Kaggle

# Resources used:

Numpy, Pandas

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Theory of Data Preprocessing:**

Data preprocessing is a crucial step in the data analysis pipeline that involves transforming raw data into a clean and organized format suitable for further exploration, analysis, and modeling. It typically includes several key steps:

1. **Data Cleaning**:
   * **Handling Missing Values**: Techniques like mean imputation, median imputation, or deletion of rows/columns with missing data ensure data completeness.
   * **Dealing with Outliers**: Outliers can skew statistical analyses or model predictions. Techniques such as Z-score normalization or IQR (Interquartile Range) method help identify and manage outliers appropriately.
   * **Removing Duplicates**: Identifying and eliminating duplicate records to maintain data integrity and avoid redundancy.
2. **Data Transformation**:
   * **Normalization**: Scaling numerical features to a standard range (e.g., 0-1) to prevent features with larger values from dominating those with smaller values. This aids in improving model performance and convergence in algorithms like gradient descent.
   * **Standardization**: Transforming data to have zero mean and unit variance, which is important for algorithms that assume normally distributed data, such as linear regression and SVMs.
3. **Feature Engineering**:
   * **Encoding Categorical Variables**: Converting categorical variables into numerical representations (e.g., one-hot encoding, label encoding) suitable for machine learning algorithms that require numeric inputs.
   * **Feature Scaling**: Ensuring all features have the same scale to prevent bias towards certain features during model training.
4. **Data Reduction**:
   * **Dimensionality Reduction**: Techniques like Principal Component Analysis (PCA) or Singular Value Decomposition (SVD) to reduce the number of input variables while retaining important information. This helps in reducing computational complexity and improving model efficiency.
5. **Data Integration and Transformation**:
   * **Data Integration**: Combining data from multiple sources into a unified dataset, ensuring consistency and compatibility across different formats.
   * **Data Transformation**: Performing transformations such as logarithmic transformation or polynomial transformation to make data distribution more suitable for modeling assumptions.

**Importance of Data Preprocessing**

* **Improves Data Quality**: Cleaning and preprocessing data ensure that data is accurate, complete, and consistent, reducing errors and biases in subsequent analyses or models.
* **Enhances Model Performance**: Proper preprocessing prepares data in a way that allows machine learning models to learn effectively and make accurate predictions.
* **Facilitates Interpretability**: Clear and well-preprocessed data enhances the interpretability of results and insights derived from machine learning models.
* **Enables Automation**: Automated data preprocessing pipelines streamline the process of preparing data for analysis, making it easier to handle large datasets efficiently.

# Program:

import pandas as pd

import numpy as np

**# Sample data**

data = {

'name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],

'age': [25, np.nan, 30, 22, 35],

'gender': ['F', 'M', 'M', 'M', 'F'],

'income': [50000, 60000, 75000, np.nan, 80000]

}

df = pd.DataFrame(data)

**# Display the original data**

print("Original DataFrame:")

print(df)

**# User-defined function for discretization**

def discretize\_age(age):

if age < 30:

return 'Young'

elif age >= 30 and age < 40:

return 'Middle-aged'

else:

return 'Old'

**# Handling missing values (NaN)**

**# Fill missing values in 'age' with the mean age**

mean\_age = df['age'].mean()

df['age'].fillna(mean\_age, inplace=True)

**# Apply discretization function to 'age' column**

df['age\_category'] = df['age'].apply(discretize\_age)

**# Drop rows with missing values in any column**

df.dropna(inplace=True)

**# Convert categorical variables (gender) to numerical**

df['gender'] = df['gender'].map({'F': 0, 'M': 1})

**# Data normalization Min -Max**

**# Normalize 'income' column to range [0, 1]**

min\_income = df['income'].min()

max\_income = df['income'].max()

df['income\_normalized'] = (df['income'] - min\_income) / (max\_income - min\_income)

**# Display cleaned, preprocessed, and discretized data**

print("\nCleaned, Preprocessed, and Discretized DataFrame:")

print(df)

—-----------------------------------------------------------------------------------------------------------------------

**Task: Download the real time data set and implement data preprocessing techniques on the real time data set**

# Source of the dataset (URL): [food data cleaning (kaggle.com)](https://www.kaggle.com/datasets/abdelrahman16/food-n/data)

# Platform used by the student: VS Code

# Following points should be written by students

# Different steps in Data Preprocessing:

# Finding missing, null values

# Replacing missing, null values with statistical parameters

# Encoding categorical data if needed (Write user defined function)

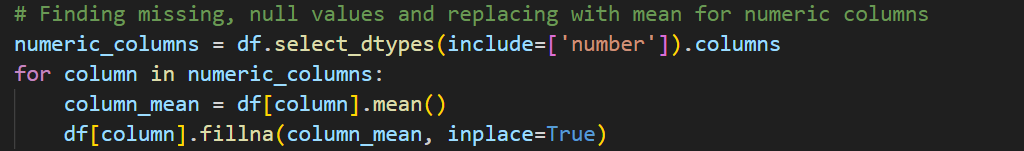
# Normalization (Write user defined function)

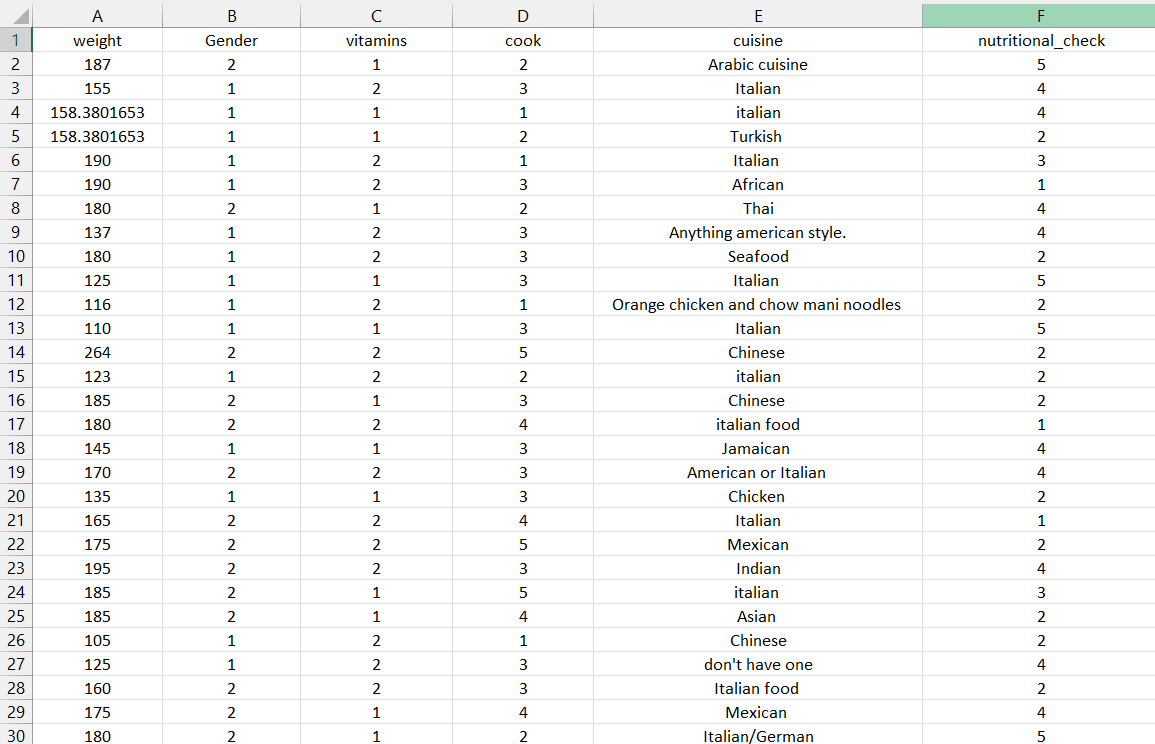
# Discretization (Write user defined function)

# Working:

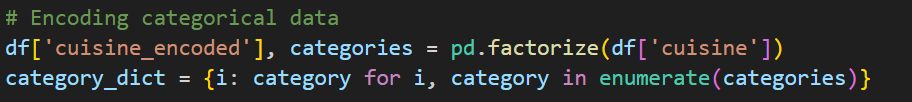
# 1] Removing redundant columns:

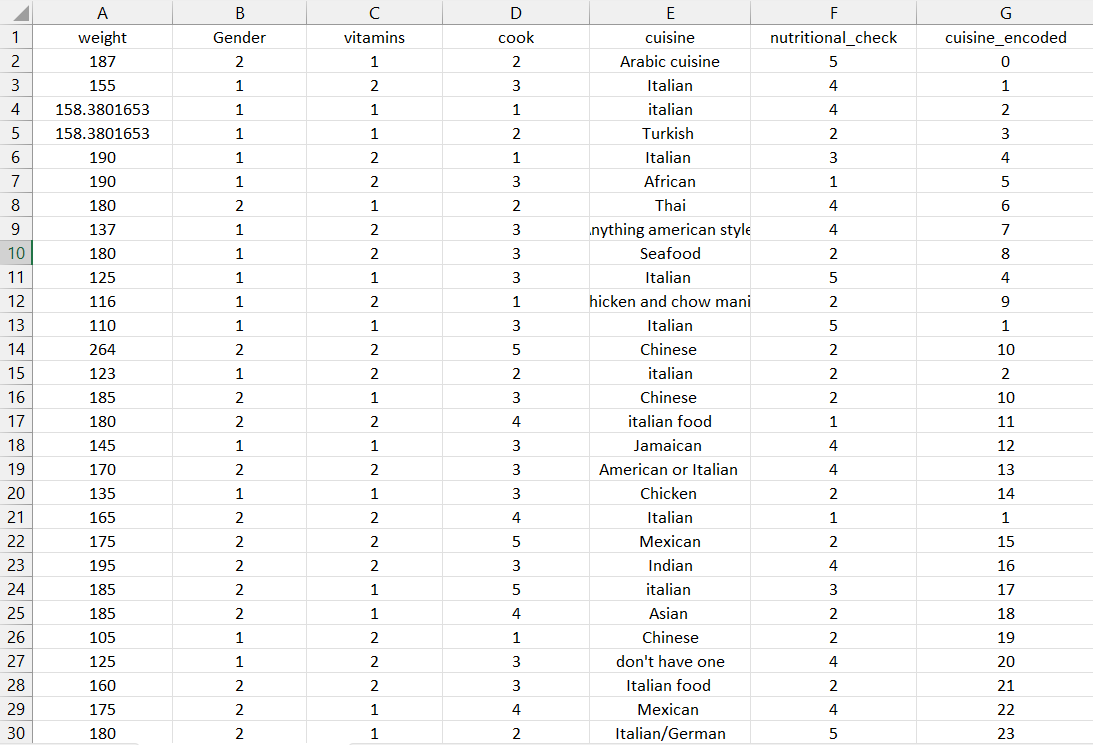
# 

2] Finding missing & null values and replacing them with the mean of the column: 

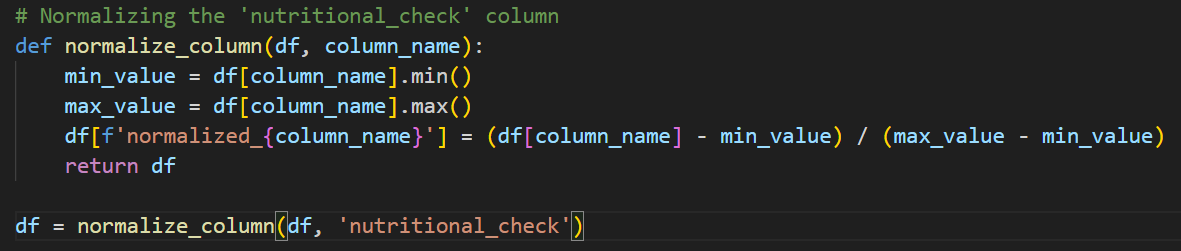


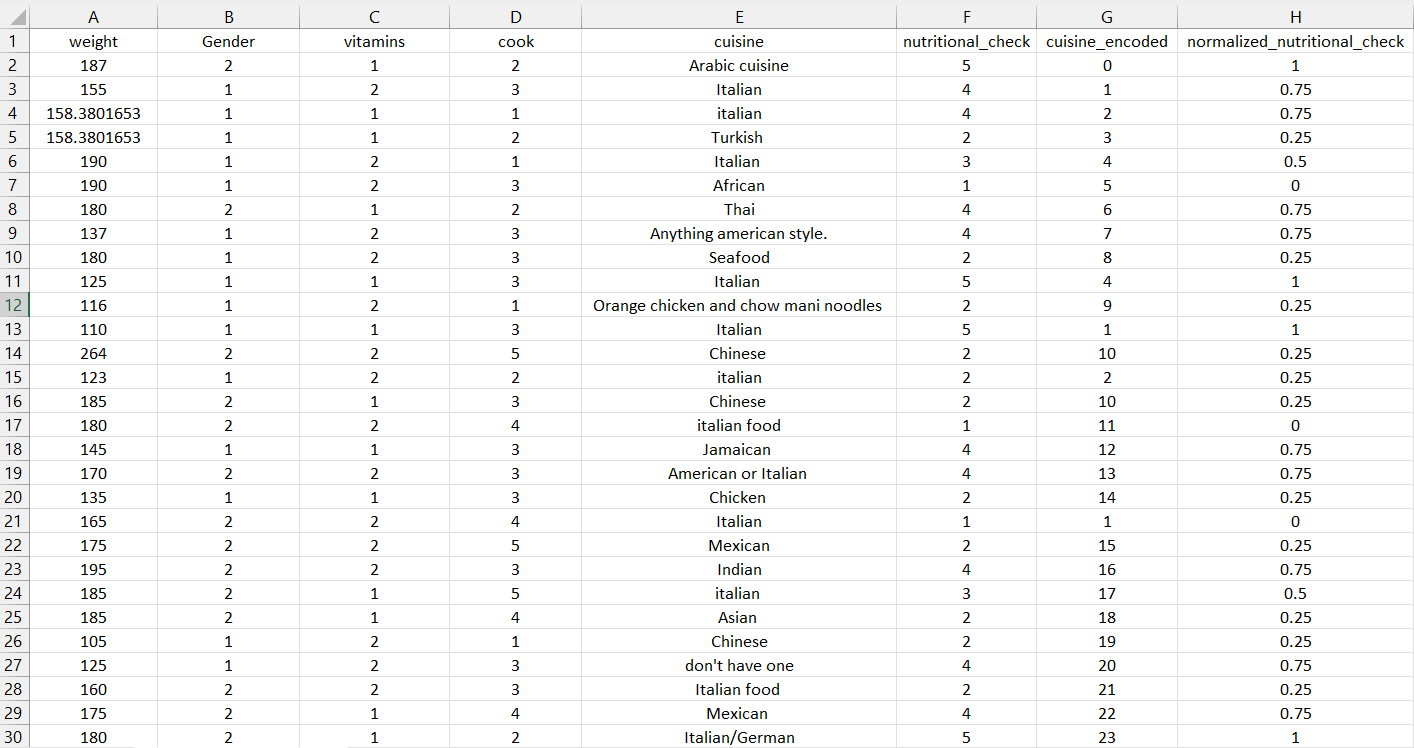
3] Encoding Cuisine:



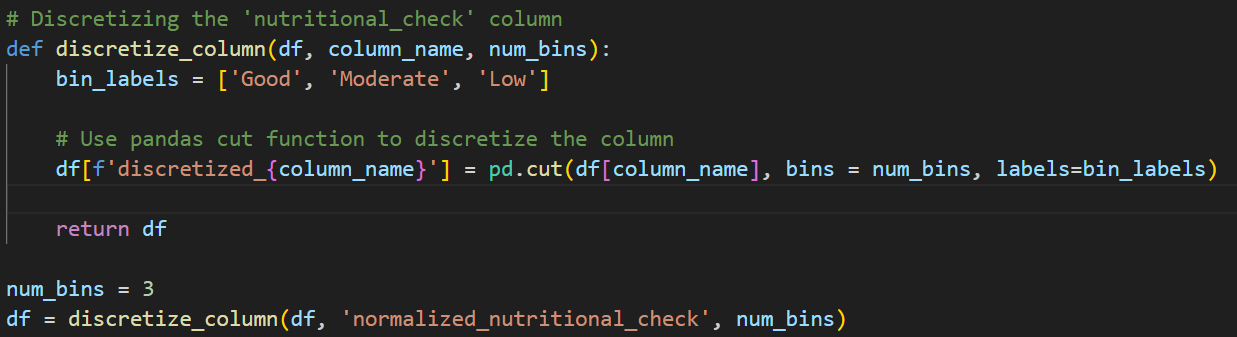


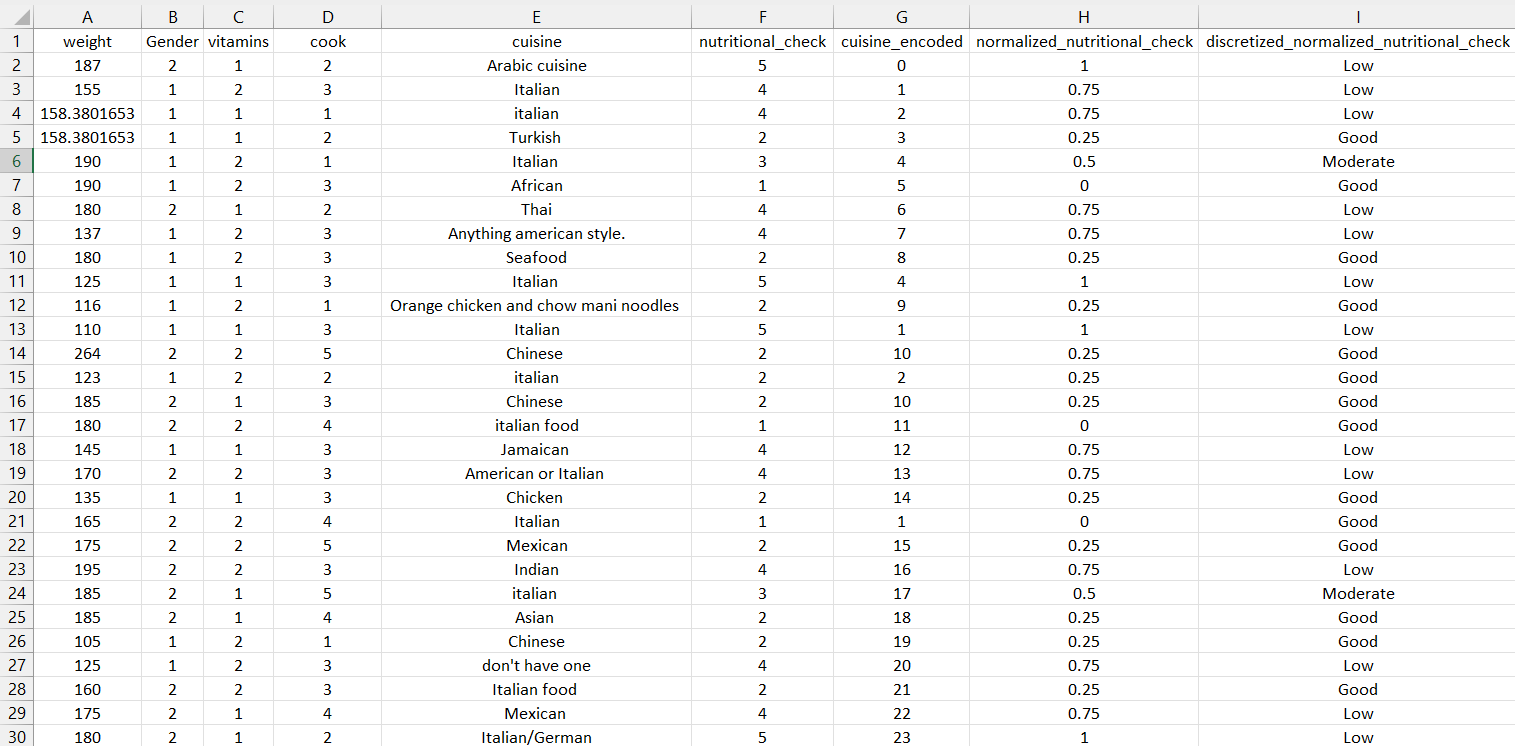
4] Min – Max Normalization of nutrition check:





5] Discretization of normalized nutrition check:





# Conclusion:

# Implementing Python-based data preprocessing on real-world datasets enhances understanding of data analytics fundamentals, crucial for solving practical problems in various domains effectively.

**Post lab questions:**

**Q.1 What are some common challenges encountered during data cleaning? How did you handle missing values in the provided dataset?**

* **Missing Values:** Data may have missing values due to various reasons such as data entry errors, sensor malfunctions, or incomplete surveys. Handling missing values is crucial to prevent biased analysis or modeling results.
* **Outliers:** Extreme values that deviate significantly from other observations can skew analysis or model training. Identifying and handling outliers requires domain knowledge and statistical methods like Z-score or IQR.
* **Inconsistent Formats:** Data may be stored in inconsistent formats (e.g., dates, currencies), requiring standardization for accurate analysis.
* **Data Integrity:** Ensuring data consistency and correctness throughout the dataset, especially when merging data from multiple sources.
* **Duplicates:** Identifying and removing duplicate records to avoid redundancy and ensure data integrity.

**Q.2 Explain the importance of data normalization in the context of machine learning models. How does normalizing benefit the analysis?**

Data normalization is crucial in machine learning to preprocess numeric data into a standardized format. Key reasons and benefits include:

* **Range Standardization:** Normalization scales numeric features to a consistent range, typically between 0 and 1 or -1 and 1. This prevents features with larger numeric ranges from dominating those with smaller ranges.
* **Improved Convergence:** Normalizing data aids in faster convergence during model training, especially in algorithms sensitive to feature scales like gradient descent-based methods.
* **Enhanced Model Performance:** Normalization helps improve the performance and accuracy of machine learning models by making the optimization process more efficient and effective.
* **Facilitates Comparison:** Normalized data allows for meaningful comparisons between different features, enabling interpretable model outputs and insights.

**Q.3 Discuss why it's essential to convert categorical variables like 'gender' into numerical representations.**

In machine learning, most algorithms expect numerical inputs. Therefore, converting categorical variables into numerical representations is essential for effective data processing and model training. Here's why:

* **Mathematical Processing**: Machine learning algorithms operate on mathematical equations and computations. Categorical variables like 'gender' (e.g., male, female) need to be encoded numerically (e.g., 0, 1) for algorithms to perform calculations.
* **Model Compatibility**: Numerical encoding enables categorical data to be included in models such as regression, SVMs, and neural networks, which require numerical inputs.
* **Feature Engineering**: Encoding categorical variables allows for the creation of meaningful features that capture relationships between categories, enhancing the model's ability to learn patterns and make predictions.
* **Interpretability**: Numeric representations of categorical variables make it easier to interpret model outputs and understand the impact of different categories on predictions.