**Batch: C3 Roll No.: 16010120193**

**Experiment 01**

**Grade: AA / AB / BB / BC / CC / CD /DD**

|  |
| --- |
| **Title:**  Dataset preparing/ pre-processing |

# Objective:

# 1. To learn how to prepare the dataset

# 2. To learn various steps in Data -Preprocessing

# Course Outcome:

# CO1 : Learn how to locate and download datasets, extract insights from that data and present their findings in a variety of different formats.

# Books/ Journals/ Websites referred:

# <https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset>

<https://pandas.pydata.org/>

# Resources used:

1. Kaggle

# Theory (About Data Preprocessing):

# Different steps in Data Preprocessing:

# Finding missing, null values:

# df. isnull() : This will return boolean value for every column in the data frame, i.e. if the vale is null it returns True, and False values are other than null.

# df. isnull(). sum() : This code will give you total number of null values in each features in the data frame.

# df. isnull(). any() : This will return Boolean value for every column, True if column has null values, False if column doesn’t have null values.

# isnull(). values. any() : This will check if missing values are present or not, will give single line Boolean answer

# isnull(). sum(). sum() : This will return the total count of missing values

# Replacing missing, null values with statistical parameters.

# df[‘language’].fillna(0,inplace=True)

# Replace missing values with ‘0’

# Encoding categorical data

# The categorical data must be encoded to numbers before we can use it to fit and evaluate a model.

# Normalization

* Normalization is the process of organizing the data in the database.
* Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
* Normalization divides the larger table into the smaller table and links them using relationship.
* The normal form is used to reduce redundancy from the database table.

# Platform used by the student: Python

# Working (Put the code and Output for each Data Preprocessing task):

# Loading dataset

import numpy as np

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

# loading dataset

df = pd.read\_csv('Datasets/update.csv', delimiter=',' )

# Check for missing values in each individual column

# Code:

#Check for missing values in each individual columns

print (df['title'].isnull())

print (df['year'].isnull())

print (df['genre'].isnull())

print (df['duration'].isnull())

print (df['country'].isnull())

print (df['language'].isnull())

print (df['director'].isnull())

print (df['actors'].isnull())

print (df['description'].isnull())

print (df['avg\_vote'].isnull())

print (df['reviews\_from\_users'].isnull())

print (df['reviews\_from\_critics'].isnull())

# Output:

# Text Description automatically generated

# Text Description automatically generated

# Text Description automatically generated

# Text Description automatically generated

# Check for missing values in columns

# Code:

print(df.isnull().any())

print(df.isnull().sum())

# Output:

# Text Description automatically generated with medium confidence

# Code:

print(df.isnull().values.any()) #check if missing values are present or not, single line answer

print(df.isnull().sum().sum()) #total count of missing values

# Output:

# 

# 

# Replacing missing values in a column

# Code:

df['language'].fillna(0,inplace=True)

print(df['language'])

# Output:

# Text Description automatically generated

# Encoding the dataset

# Code:

col\_list = ["genre"]

# Encoding

genre = {'Romance':1,

            'Drama':2,

            'Crime':3,

            'Action':4,

            'Adventure':5,

            'Comedy':6,

            'Horror':7,

            'Thriller':8,

            'Fantasy':9,

            'Mystery' : 10,

            'Animation' : 11,

            'Family' : 12,

            'Sci-Fi' : 13,

            'Biography' : 14,

            'Sport' : 15,

            'Musical' : 16,

            'History' : 17,

            'War' : 18,

            }

df['genre'] = df.genre.map(genre)

print(df['genre'])

# Output:

# Text Description automatically generated

# Normalization

# Code:

# normalization of columns

df['avg\_vote'].plot(kind = 'bar')

df\_min\_max\_scaled = df.copy()

# apply normalization techniques to the Rating column

column = 'avg\_vote'

df\_min\_max\_scaled[column] = (df\_min\_max\_scaled[column] - df\_min\_max\_scaled[column].min()) / (df\_min\_max\_scaled[column].max() - df\_min\_max\_scaled[column].min())

# view normalized data

print(df\_min\_max\_scaled)

# Output:

# Graphical user interface Description automatically generated

# Platform used by the student: R

# Working (Put the code and Output for each Data Preprocessing task):

# Loading and finding the summary of the dataset

# Code:

dataset = read.csv("update.csv")

# finding the summary of the dataset

summary(dataset)

# Output:

# Text Description automatically generated with medium confidence

# Encoding the dataset

# Code:

dataset = read.csv("update.csv")

#Encoding the data

dataset$genre = factor(dataset$genre, levels = c('Romance',

            'Drama',

            'Crime',

            'Action',

            'Adventure',

            'Comedy',

            'Horror',

            'Thriller',

            'Fantasy',

            'Mystery',

            'Animation',

            'Family',

            'Sci-Fi',

            'Biography',

            'Sport',

            'Musical',

            'History',

            'War'

            ),labels=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18))

print(dataset$genre)

# 

# Output:

# Text Description automatically generated

# Conclusion

# Thus, we learnt how to locate and download datasets, and process the data, extract insights from that data and present their findings in a variety of different formats.

# Post Lab Question:

# Write the importance of Data Preprocessing in Software System Designing

# The importance of Data Preprocessing in Software System Designing is:

# It reduces overall development cycle

# Makes the data process easier to maintain (no matter which programming language or data preparation tool is used)

# Make the system more open and easier to operate

# Ensure data quality from the beginning