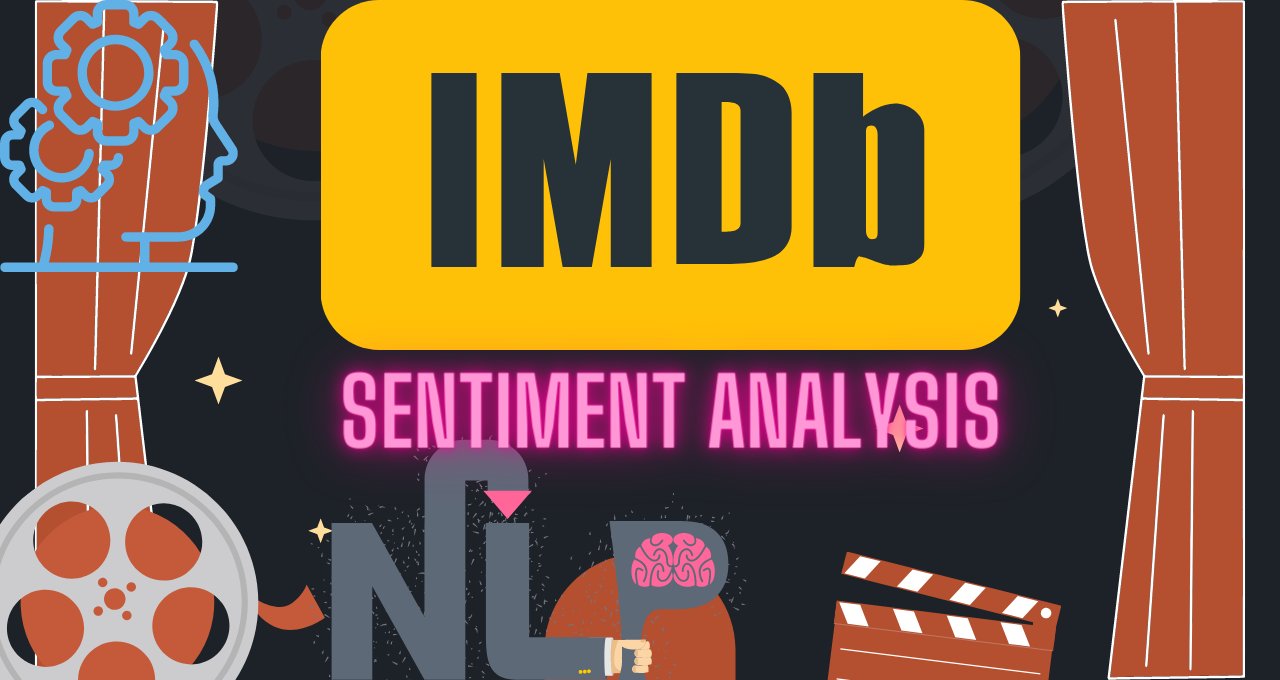
**Predicting IMDb Scores using Machine Learning**

TEAM MEMBER:SUBIKSHA.V

**PHASE 3** – DOCUMENTATION PART 1

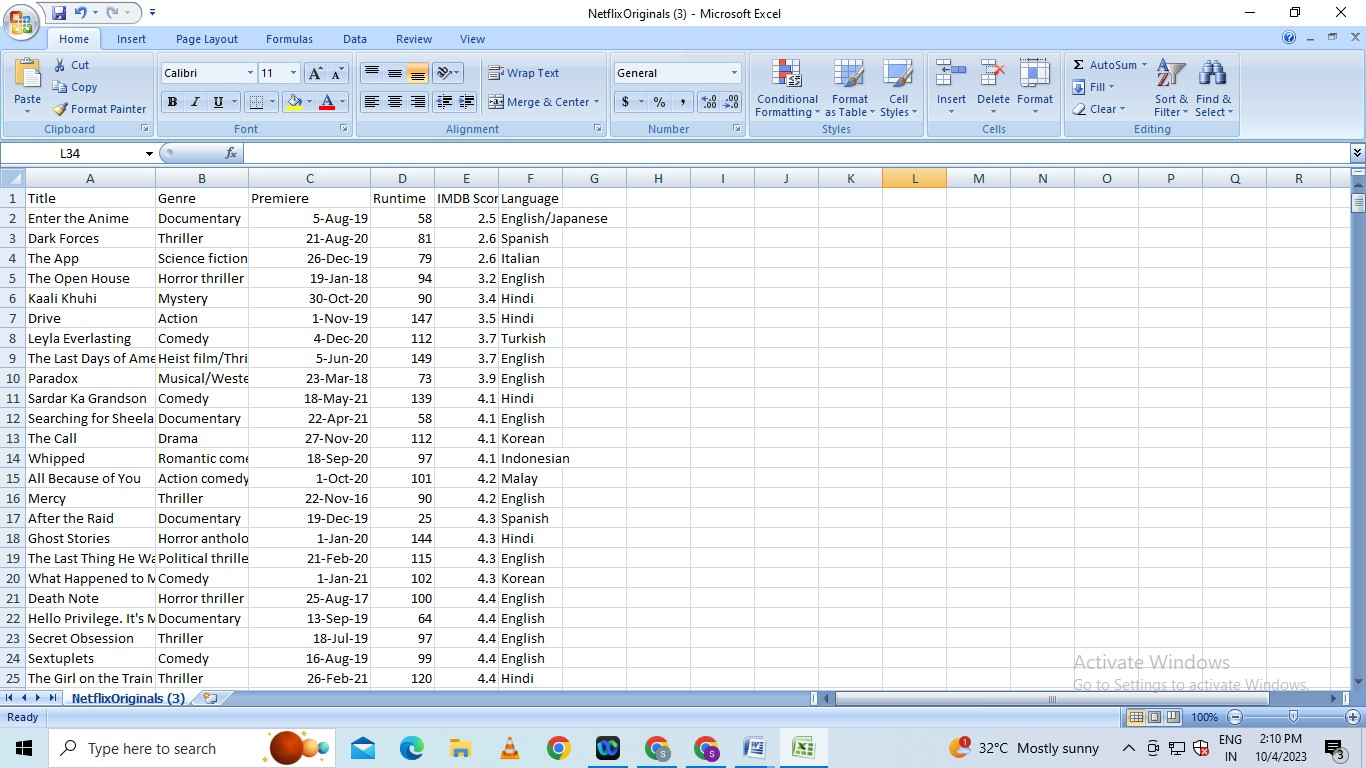


**INTRODUCTION:**

IMDb is the world’s most popular and authoritative source for movie, TV, and celebrity content. IMDb users often look at ratings to get an idea of how good movies are, so that they can decide what movies to watch or which ones to prioritize. However, movies that are not yet released don’t have ratings, and even the ones with few votes often change as more users vote. Therefore, I wrote code to predict IMDb ratings of new movies based on various features, such as budget, actors, directors, writers, release year, genres, and plot. While others have used linear regressions to predict ratings of movies in general, those predictions rely on features like movie earnings or number of votes, which would not be available for new movies. I instead combined cosine similarities and normalized Euclidean distances with a modified kNN algorithm, which still produced mostly very accurate predictions. This will provide a way to obtain an estimated rating that’s not yet provided by IMDb.

**Data Source :**

A Good Data for Predicting IMDb Scores using machine learning model should be Accurate , complete , accessible

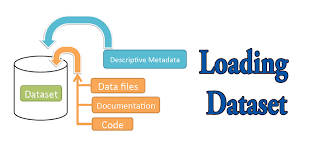
**Dataset Link : (**[**https://www.kaggle.com/datasets/luiscorter/netflix-original-films-imdb-scores**](https://www.kaggle.com/datasets/luiscorter/netflix-original-films-imdb-scores))

# Loading Datasets :

Data preparation is the first step of the 7 step Rapid Process Troubleshooting methodology. Data is prepared to create a meaningful and effective dataset that can be used to model the process in step 3 of the methodology. At any stage during data preparation, you can load any dataset in your data recipe into the Troubleshooters, provided the datasets contain double fields. Simply select the dataset required, and click [Load dataset] from the panel at the bottom of the canvas.

From this panel, it is possible to perform the following actions:

* [Load dataset](https://www.ge.com/digital/documentation/csense/version85/Data%20Preparation/#Load_dataset)
* [Unload dataset](https://www.ge.com/digital/documentation/csense/version85/Data%20Preparation/#Unload_dataset)
* [Switch datasets](https://www.ge.com/digital/documentation/csense/version85/Data%20Preparation/#Switch_datasets)
* [Change categories](https://www.ge.com/digital/documentation/csense/version85/Data%20Preparation/#Categories)
* [Remove models](https://www.ge.com/digital/documentation/csense/version85/Data%20Preparation/#Remove_models)



## How do you load a dataset?

Load Data With Built-In Python Functions  
  
To both read from and write to a file, you can use the built-in function open() , which takes in two parameters: file name and mode. File name: the directory path to the file that you want to read or write to. Mode: the mode you want to use for the file.

**Program:**

def load\_csv(filepath):

data = []

col = []

checkcol = False

with open(filepath) as f:

for val in f.readlines():

val = val.replace("\n","")

val = val.split(',')

if checkcol is False:

col = val

checkcol = True

else:

data.append(val)

df = pd.DataFrame(data=data, columns=col)

return df

**Preprocessing Datasets :**

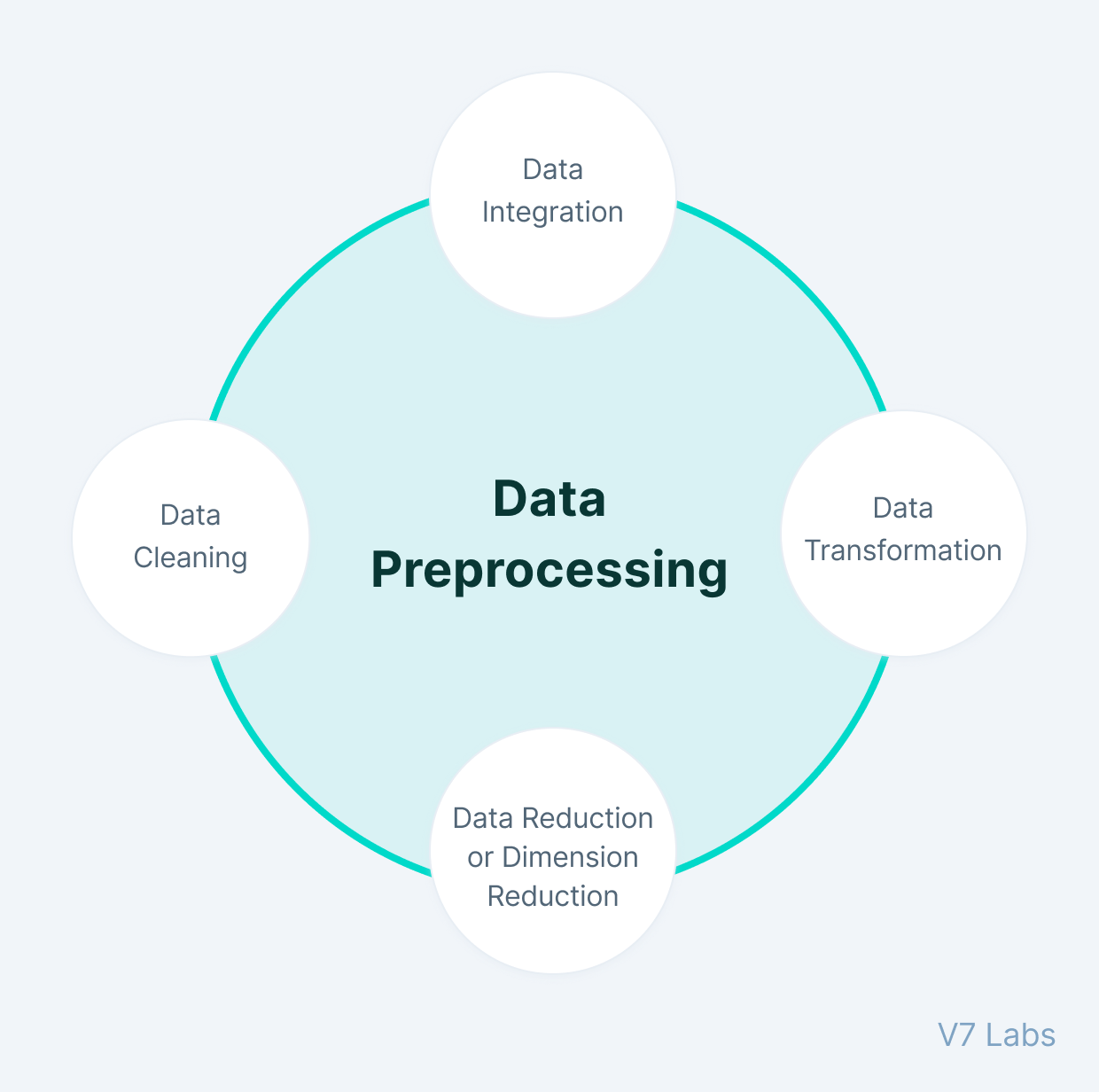
Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

### ****Need of Data Preprocessing**** :

* For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example,Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.
* Another aspect is that the data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithm are executed in one data set, and best out of them is chosen.



## Why do we need Data Preprocessing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

* Getting the dataset
* Importing libraries
* Importing datasets
* Finding Missing Data
* Encoding Categorical Data
* Splitting dataset into training and test set
* Feature scaling

**Program:**

import pandas as pd

import scipy

import numpy as np

from sklearn.preprocessing import MinMaxScaler

import seaborn as sns

import matplotlib.pyplot as plt

**Conclusion:**

This article gives a comprehensive overview of the Data Loading component of the ETL process. It also gave loads of tools that are cloud-based and can ease the process of ETL.

To make things easier, Hevo comes into the picture. Hevo Data is a No-code Data Pipeline and has awesome 100+ pre-built Integrations that you can choose from.

Preprocessing data before applying it to a machine learning algorithm is a crucial step in the ML workflow. It helps to improve the accuracy, reduce the time and resources required to train the model, prevent overfitting, and improve the interpretability of the model.