**PREDICTING IMDb SCORES USING MACHINE LEARNING**

**TEAM MEMBER:** SRI LAKSHMI T

**Phase 3 submission document**

**Project Title:** IMDb Scores Prediction

**Phase 3:** Development Part 1

**TOPIC :** In this part you will begin building your project by loading and preprocessing the dataset. Begin building the IMDb score prediction model by loading and preprocessing the dataset. Load the movie dataset and preprocess the data for analysis.

**DATA LOADING IN MACHINE LEARNING :**

To load data for machine learning, you typically need to follow these steps:

1. Import the necessary libraries: Start by importing the required libraries such as NumPy, Pandas, or TensorFlow, depending on your specific needs.

2. Obtain the data: Get the dataset that you want to use for machine learning. This data can come from various sources, such as CSV files, databases, or APIs.

3. Load the data into memory: Use the appropriate functions provided by the libraries to read and load the data into memory. For example, you can use Pandas' read\_csv() function to load data from a CSV file into a DataFrame.

4. Explore and preprocess the data: Explore the loaded dataset to get a better understanding of its structure and contents. Perform data preprocessing steps like handling missing values, normalizing or standardizing features, and encoding categorical variables.

5. Split the data: Split the dataset into training and test sets. The training set is used to train the machine learning model, while the test set is used to evaluate its performance.

6. Further split the data (optional): Additionally, you can split the training set into training and validation sets. This allows you to monitor the model's performance during training and tune the hyperparameters accordingly.

7. Convert the data: Convert the data into a suitable format for model training. This might involve transforming text data into numerical representations using techniques like tokenization or one-hot encoding.

8. Finally, feed the data into your machine learning model for training or evaluation.

Here's an example Python program that outlines the steps mentioned earlier for loading data for machine learning using the Pandas library:

import pandas as pd

# Step 1: Import necessary libraries

# Step 2: Obtain the data

# Assuming the data is in a CSV file named 'data.csv' in the same directory

data = pd.read\_csv('data.csv')

# Step 3: Load the data into memory

# Step 4: Explore and preprocess the data

# Example: Handling missing values

data = data.dropna() # Remove rows with missing values

# Example: Splitting the data into features (X) and target variable (y)

X = data.drop('target', axis=1) # Assumes 'target' is the column containing the target values

y = data['target']

# Step 5: Split the data into training and test sets

# Step 6: Further split the data (optional)

# Step 7: Convert the data

# Step 8: Feed the data into your machine learning model for training or evaluation

**Program:**

To predict IMDb scores using Python, you can use various machine learning algorithms. Here's a basic example using linear regression:

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

# Step 1: Load the dataset

data\_path = "path/to/dataset.csv"

# Replace with the actual path to your dataset

data = pd.read\_csv(data\_path)

# Step 2: Split the data into features (X) and target variable (y)

X = data.drop('IMDb\_Score', axis=1) # Assuming 'IMDb\_Score' is the column to predict

y = data['IMDb\_Score']

# Step 3: Split the data into training and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=123)

# Step 4: Create and train the model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Step 5: Predict using the trained model

y\_pred = model.predict(X\_test)

# Step 6: Evaluate the model

mse = mean\_squared\_error(y\_test, y\_pred)

print("Mean squared error:", mse)

# You can further assess the model's performance using other metrics or perform hyperparameter tuning as needed.

This is a basic example using linear regression. Depending on the nature of your dataset and the specific problem, these might want to use different algorithms or apply additional preprocessing and feature engineering techniques.

**PREPROCESSING OF DATASET :**

* Preprocessing of a dataset refers to the steps taken to transform and prepare the data for further analysis or modeling.
* It involves handling missing values, removing irrelevant or noisy data, transforming categorical variables, and scaling numeric data, among other tasks.
* The preprocessing steps vary depending on the nature of the dataset and the specific requirements of the problem at hand.

**SYNTAX FOR PREPROCESSING THE DATASET :**

The Syntax for preprocessing data depends on programming language and libraries you are using.Here we are using more libraries such as pandas etc..

**Here are some commonly performed preprocessing steps:**

1. Handling missing data: This involves identifying and dealing with missing values in the dataset. Options include removing rows or columns with missing values, imputing missing values with the mean or median, or using advanced imputation techniques.

2. Removing irrelevant data: If the dataset contains unnecessary columns or attributes that won't contribute to the analysis or modeling, those columns can be dropped or removed from the dataset.

3. Handling categorical variables: Categorical variables, such as text labels or nominal values, need to be converted into a numerical representation for machine learning algorithms to process them. This process is known as encoding or one-hot encoding and can be done using techniques like label encoding or creating dummy variables.

4. Scaling numeric data: Numeric variables that have different scales or units can be standardized or normalized to ensure they are on a similar scale. Common techniques include min-max scaling or standardization using mean and standard deviation.

5. Handling outliers: Outliers, which are extreme or unusual values in the dataset, can impact the modeling process. They can be treated by either removing them, transforming them, or using robust algorithms that are less sensitive to outliers.

6. Feature engineering: This involves creating new features from existing ones that may improve the performance of the model. It could include operations like polynomial expansion, log transforms, or creating interaction terms.

7. Splitting into training and testing sets: The dataset is typically split into a training set and a testing set to assess the model's performance. The training set is used to train the model, while the testing set is used to evaluate its performance on unseen data.

These are general steps in preprocessing a dataset, and the specific preprocessing required may vary depending on the dataset and the machine learning task being undertaken.

**Steps Involved in Data Preprocessing:**

**1. Data Cleaning:**

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

**(a). Missing Data:**

This situation arises when some data is missing in the data. It can be handled in various ways.

Some of them are:

Ignore the tuples:

This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

Fill the Missing values:

There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

**(b). Noisy Data:**

Noisy data is a meaningless data that can’t be interpreted by machines.It can be generated due to faulty data collection, data entry errors etc.

It can be handled in following ways :

Binning Method:

This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.

Regression:

Here data can be made smooth by fitting it to a regression function.The regression used may be linear (having one independent variable) or multiple (having multiple independent variables). Clustering:

This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

**2. Data Transformation:**

This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

Normalization:

It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)

Attribute Selection:

In this strategy, new attributes are constructed from the given set of attributes to help the mining process.

Discretization:

This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.

Concept Hierarchy Generation:

Here attributes are converted from lower level to higher level in hierarchy. For Example-The attribute “city” can be converted to “country”.

**3. Data Reduction:**

Data reduction is a crucial step in the data mining process that involves reducing the size of the dataset while preserving the important information. This is done to improve the efficiency of data analysis and to avoid overfitting of the model. Some common steps involved in data reduction are:

Feature Selection:

This involves selecting a subset of relevant features from the dataset. Feature selection is often performed to remove irrelevant or redundant features from the dataset. It can be done using various techniques such as correlation analysis, mutual information, and principal component analysis (PCA).

Feature Extraction:

This involves transforming the data into a lower-dimensional space while preserving the important information. Feature extraction is often used when the original features are high-dimensional and complex. It can be done using techniques such as PCA, linear discriminant analysis (LDA), and non-negative matrix factorization (NMF).

Sampling:

This involves selecting a subset of data points from the dataset. Sampling is often used to reduce the size of the dataset while preserving the important information. It can be done using techniques such as random sampling, stratified sampling, and systematic sampling.

Clustering:

This involves grouping similar data points together into clusters. Clustering is often used to reduce the size of the dataset by replacing similar data points with a representative centroid. It can be done using techniques such as k-means, hierarchical clustering, and density-based clustering.

Compression:

This involves compressing the dataset while preserving the important information. Compression is often used to reduce the size of the dataset for storage and transmission purposes. It can be done using techniques such as wavelet compression, JPEG compression, and gzip compression.

**CONCLUSION:**

In conclusion, loading and preprocessing a dataset is a crucial step in preparing data for machine learning tasks. The process typically involves handling missing data, removing irrelevant features, encoding categorical variables, scaling numeric data, handling outliers, and potentially performing feature engineering**.**