**Predicting IMDb Scores Using Machine Learning**

TEAM MEMBER

732521104048 : SUBHASHINI P

**Phase 3 Submission Document**

**Project :** Predicting IMDb Scores



**Introduction:**

* Predicting IMDb scores for movies or TV shows typically involves using machine learning models and features such as cast, crew, genre, user reviews, and more. You can use regression algorithms to build a predictive model.
* The quality of your predictions depends on the quality and quantity of data, as well as the choice of features and model.
* In this project , we will explore advanced regression techniques to enhance the accuracy and robustness of IMDb scores prediction models.
* Highlight the limitations of traditional linear regression models in capturing complex relationships.
* Emphasize the need for advanced regression techniques like Gradient Boosting and Neural Networks to enchance prediction accuracy.

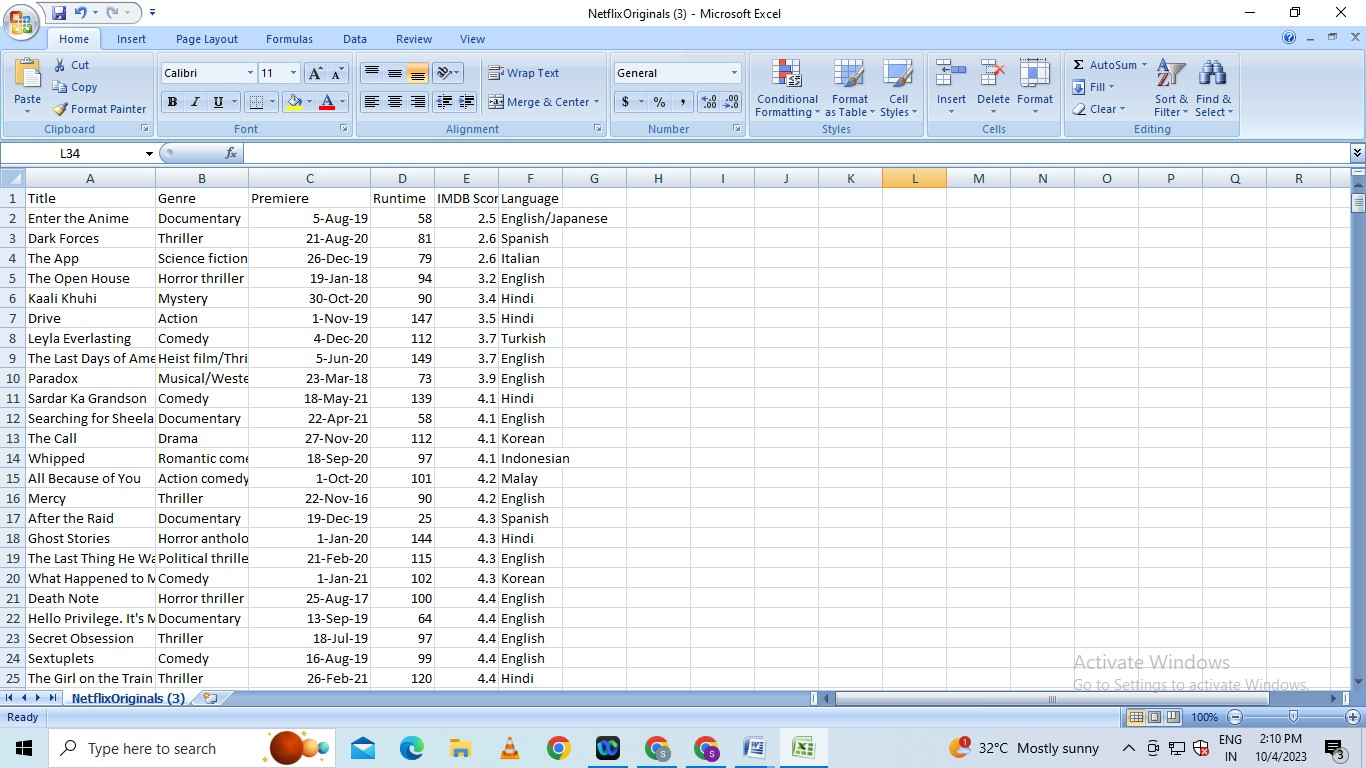
**Content For Project Phase 3 :**

Phase 3: Development Part 1

**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))



# Data loading

These loading utilites can be combined with [preprocessing layers](https://keras.io/guides/preprocessing_layers/) to futher transform your input dataset before training.

Here's a quick example: let's say you have 10 folders, each containing 10,000 images from a different category, and you want to train a classifier that maps an image to its category.

Your training data folder would look like this:

training\_data/

...class\_a/

......a\_image\_1.jpg

......a\_image\_2.jpg

...class\_b/

......b\_image\_1.jpg

......b\_image\_2.jpg

etc.

You may also have a validation data folder validation\_data/ structured in the same way.

You could simply do:

from tensorflow import keras

train\_ds = keras.utils.image\_dataset\_from\_directory(

directory='training\_data/',

labels='inferred',

label\_mode='categorical',

batch\_size=32,

image\_size=(256, 256))

validation\_ds = keras.utils.image\_dataset\_from\_directory(

directory='validation\_data/',

labels='inferred',

label\_mode='categorical',

batch\_size=32,

image\_size=(256, 256))

model = keras.applications.Xception(

weights=None, input\_shape=(256, 256, 3), classes=10)

model.compile(optimizer='rmsprop', loss='categorical\_crossentropy')

model.fit(train\_ds, epochs=10, validation\_data=validation\_ds)

**Data Preprocessing**

Data preprocessing transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks. The techniques are generally used at the earliest stages of the [machine learning](https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML) and AI development pipeline to ensure accurate results.

There are several different tools and methods used for preprocessing data, including the following:

* sampling, which selects a representative subset from a large population of data;
* transformation, which manipulates raw data to produce a single input;
* denoising, which removes [noise](https://www.techtarget.com/whatis/definition/noise) from data;
* imputation, which synthesizes statistically relevant data for missing values;
* [normalization](https://searchsqlserver.techtarget.com/definition/normalization), which organizes data for more efficient access; and
* feature extraction, which pulls out a relevant feature subset that is significant in a particular context.

These tools and methods can be used on a variety of data sources, including data stored in files or databases and streaming data.

### Why is data preprocessing important?

Virtually any type of data analysis, [data science](https://www.techtarget.com/searchenterpriseai/definition/data-science) or AI development requires some type of data preprocessing to provide reliable, precise and robust results for enterprise applications.

Real-world data is messy and is often created, processed and stored by a variety of humans, business processes and applications. As a result, a data set may be missing individual fields, contain manual input errors, or have duplicate data or different names to describe the same thing. Humans can often identify and rectify these problems in the data they use in the line of business, but [data used to train machine learning](https://www.techtarget.com/searchbusinessanalytics/feature/Data-preparation-in-machine-learning-6-key-steps) or deep learning algorithms needs to be automatically preprocessed.

### What are the key steps in data preprocessing?

The steps used in data preprocessing include the following:

**1. Data profiling**. [Data profiling](https://www.techtarget.com/searchdatamanagement/definition/data-profiling) is the process of examining, analyzing and reviewing data to collect statistics about its quality. It starts with a survey of existing data and its characteristics. Data scientists identify data sets that are pertinent to the problem at hand, inventory its significant attributes, and form a hypothesis of features that might be relevant for the proposed analytics or machine learning task. They also relate data sources to the relevant business concepts and consider which preprocessing libraries could be used.

**2. Data cleansing**. The aim here is to find the easiest way to rectify quality issues, such as eliminating bad data, filling in missing data or otherwise ensuring the raw data is suitable for feature engineering.

**3. Data reduction.**Raw data sets often include redundant data that arise from characterizing phenomena in different ways or data that is not relevant to a particular ML, AI or analytics task. Data reduction uses techniques like principal component analysis to transform the raw data into a simpler form suitable for particular use cases.

**4. Data transformation**. Here, data scientists think about how different aspects of the data need to be organized to make the most sense for the goal. This could include things like structuring [unstructured data](https://www.techtarget.com/searchbusinessanalytics/definition/unstructured-data), combining salient variables when it makes sense or identifying important ranges to focus on.

**5. Data enrichment**. In this step, data scientists apply the various feature engineering libraries to the data to effect the desired transformations. The result should be a data set organized to achieve the optimal balance between the training time for a new model and the required compute.

**6. Data validation**. At this stage, the data is split into two sets. The first set is used to train a machine learning or deep learning model. The second set is the testing data that is used to gauge the accuracy and robustness of the resulting model. This second step helps identify any problems in the [hypothesis](https://www.techtarget.com/whatis/definition/hypothesis) used in the cleaning and feature engineering of the data. If the data scientists are satisfied with the results, they can push the preprocessing task to a [data engineer](https://www.techtarget.com/searchdatamanagement/definition/data-engineer) who figures out how to scale it for production. If not, the data scientists can go back and make changes to the way they implemented the data cleansing and feature engineering steps.

### Data preprocessing techniques

There are two main categories of preprocessing -- [data cleansing](https://www.techtarget.com/searchdatamanagement/definition/data-scrubbing) and feature engineering. Each includes a variety of techniques, as detailed below.

### Data cleansing

Techniques for cleaning up messy data include the following:

**Identify and sort out missing data.**There are a variety of reasons a data set might be missing individual fields of data. Data scientists need to decide whether it is better to discard records with missing fields, ignore them or fill them in with a probable value. For example, in an [IoT](https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT) application that records temperature, adding in a missing average temperature between the previous and subsequent record might be a safe fix.

**Reduce**[**noisy data**](https://www.techtarget.com/searchbusinessanalytics/definition/noisy-data)**.** Real-world data is often noisy, which can distort an analytic or AI model. For example, a temperature sensor that consistently reported a temperature of 75 degrees Fahrenheit might erroneously report a temperature as 250 degrees. A variety of statistical approaches can be used to reduce the noise, including binning, regression and clustering.

**Identify and remove duplicates.** When two records seem to repeat, an algorithm needs to determine if the same measurement was recorded twice, or the records represent different events. In some cases, there may be slight differences in a record because one field was recorded incorrectly. In other cases, records that seem to be duplicates might indeed be different, as in a father and son with the same name who are living in the same house but should be represented as separate individuals. Techniques for identifying and removing or joining duplicates can help to automatically address these types of problems.

### Feature engineering

Feature engineering, as noted, involves techniques used by data scientists to organize the data in ways that make it more efficient to train [data models](https://www.techtarget.com/searchdatamanagement/definition/data-modeling) and run inferences against them. These techniques include the following:

**Feature scaling or normalization.**Often, multiple variables change over different scales, or one will change linearly while another will change exponentially. For example, salary might be measured in thousands of dollars, while age is represented in double digits. Scaling helps to transform the data in a way that makes it easier for algorithms to tease apart a meaningful relationship between variables.

**Data reduction.**Data scientists often need to combine a variety of [data sources to create a new AI or analytics model](https://www.techtarget.com/searchbusinessanalytics/feature/6-data-preparation-best-practices-for-analytics-applications). Some of the variables may not be correlated with a given outcome and can be safely discarded. Other variables might be relevant, but only in terms of relationship -- such as the ratio of debt to credit in the case of a model predicting the likelihood of a loan repayment; they may be combined into a single variable. Techniques like principal component analysis play a key role in reducing the number of dimensions in the training data set into a more efficient representation.

**Discretization.**It's often useful to lump raw numbers into discrete intervals. For example, income might be broken into five ranges that are representative of people who typically apply for a given type of loan. This can reduce the overhead of training a model or running inferences against it.

**Feature encoding**. Another aspect of feature engineering involves organizing unstructured data into a structured format. Unstructured data formats can include text, audio and video. For example, the process of developing natural language processing algorithms typically starts by using data transformation algorithms like Word2vec to translate words into numerical vectors. This makes it easy to represent to the algorithm that words like "mail" and "parcel" are similar, while a word like "house" is completely different. Similarly, a [facial recognition](https://www.techtarget.com/searchenterpriseai/definition/facial-recognition) algorithm might reencode raw pixel data into vectors representing the distances between parts of the face.

### How is data preprocessing used?

Data preprocessing plays a key role in earlier stages of machine learning and AI application development, as noted earlier. In an [AI](https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence) context, data preprocessing is used to improve the way data is cleansed, transformed and structured to improve the accuracy of a new model, while reducing the amount of compute required.

A good data preprocessing pipeline can create reusable components that make it easier to test out various ideas for streamlining business processes or improving customer satisfaction. For example, preprocessing can improve the way data is organized for a recommendation engine by improving the age ranges used for categorizing customers.

Preprocessing can also simplify the work of creating and modifying data for more accurate and targeted business intelligence insights. For example, customers of different sizes, categories or regions may exhibit different behaviors across regions. Preprocessing the data into the appropriate forms could help BI teams weave these insights into BI dashboards.

In a customer relationship management ([CRM](https://www.techtarget.com/searchcustomerexperience/definition/CRM-customer-relationship-management)) context, data preprocessing is a component of web mining. Web usage logs may be preprocessed to extract meaningful sets of data called user transactions, which consist of groups of URL references. User sessions may be tracked to identify the user, the websites requested and their order, and the length of time spent on each one. Once these have been pulled out of the raw data, they yield more useful information that can be applied, for example, to consumer research, marketing or [personalization](https://www.techtarget.com/searchcustomerexperience/definition/personalization).

**Conclusion**

The data preprocessing phase is crucial for determining the correct input data for the machine learning algorithms. As we saw previously, without applying the proper techniques, you can have a worse model result. For example, the k-nearest neighbors algorithm is affected by noisy and redundant data, is sensitive to different scales, and doesn’t handle a high number of attributes well. If you use this algorithm, you must clean the data, avoid high dimensionality and normalize the attributes to the same scale.

However, if you use a Decision Tree algorithm, you don’t need to worry about normalizing the attributes to the same scale. Thus, each model has its own peculiarity, and you need to know beforehand to give a proper data input to the model. With that said, now you can move forward to the model exploration phase and know those peculiarities of the algorithms.