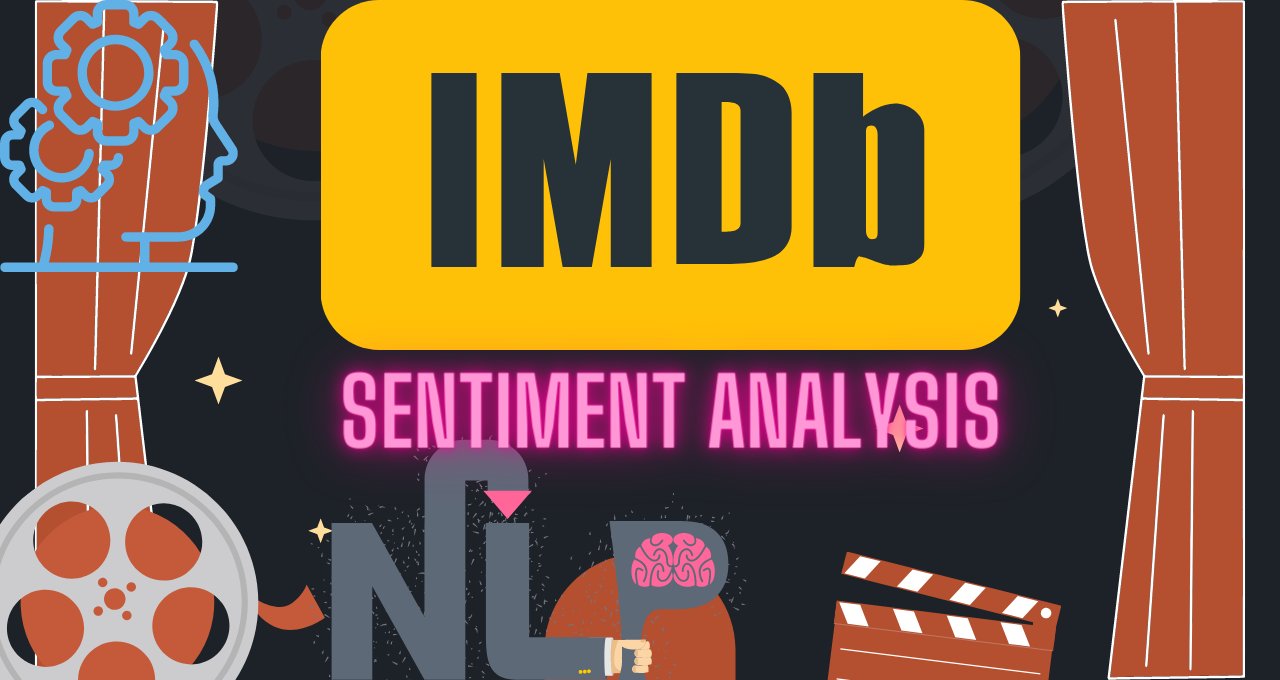
**PREDICTING IMDb SCORES USING MACHINE LEARNING**

**TEAM MEMBER**:SUBIKSHA.V

**PHASE 4** – DOCUMENTATION PART 2

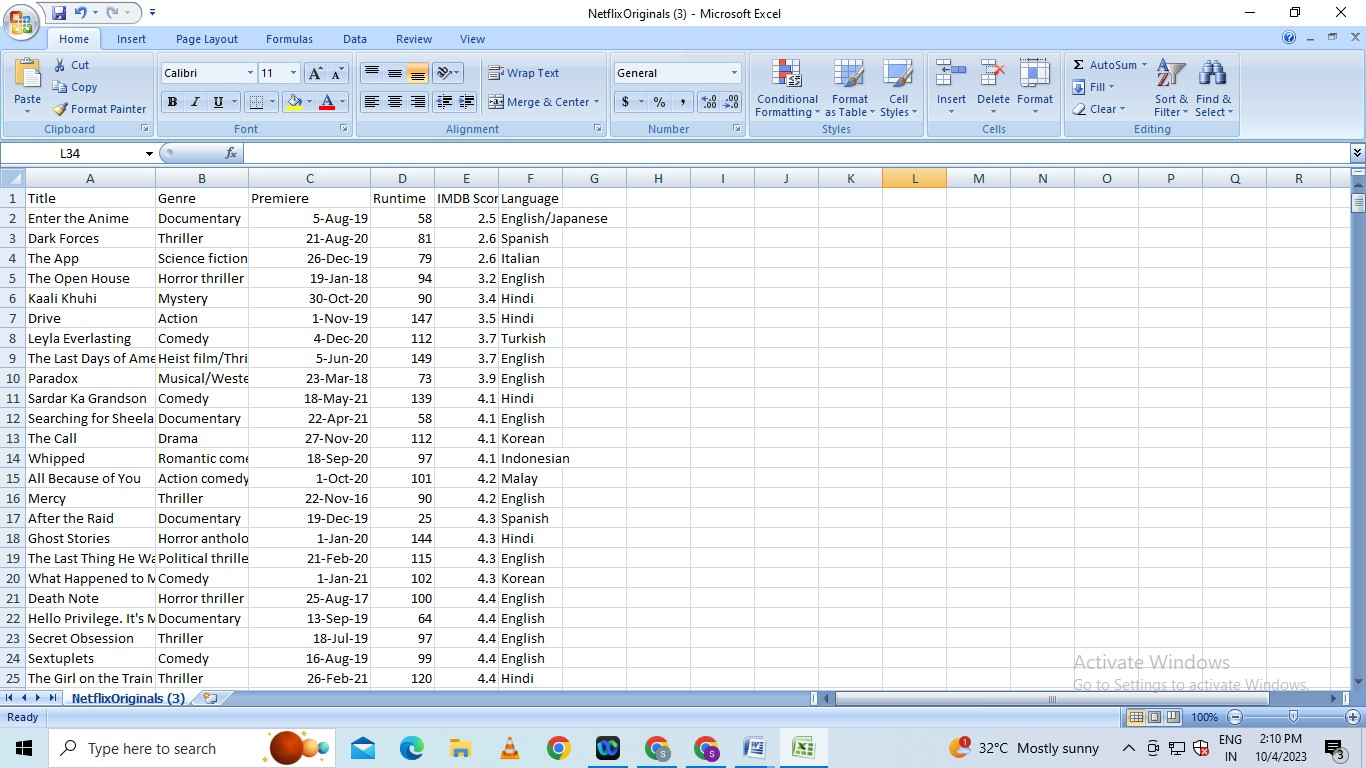


**INTRODUCTION:**

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 :**

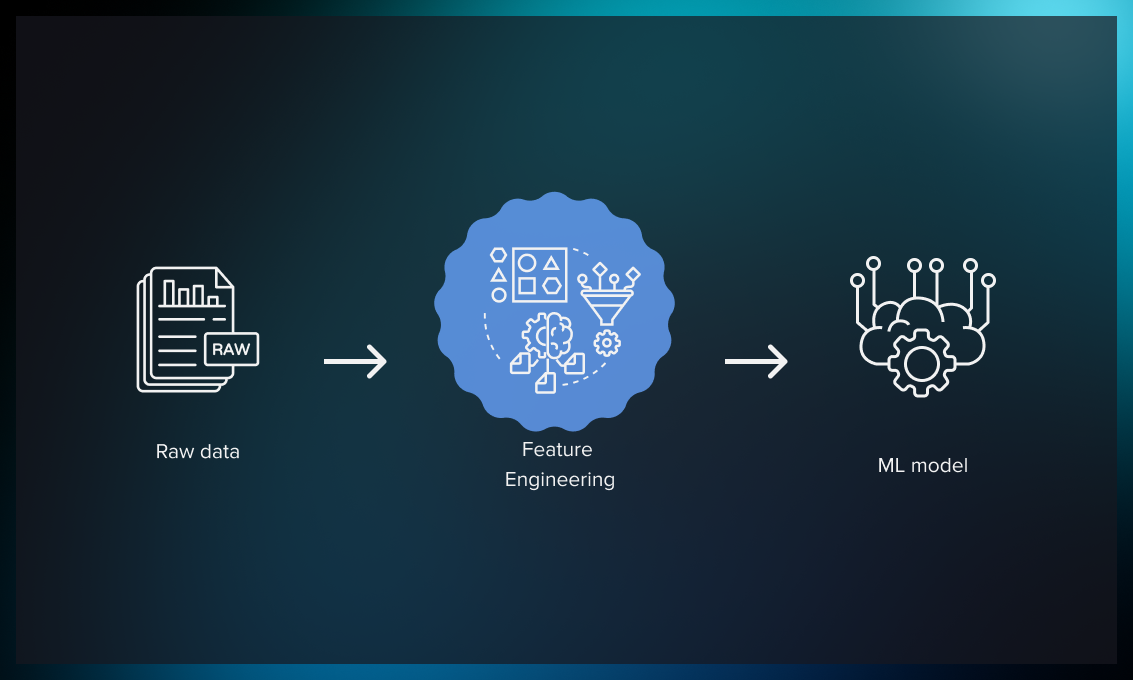
A Good Data for Predicting IMDb Scores using machine learning model should be Accurate , complete , accessible

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

**Features Engineering :**

Feature engineering is the process of **transforming raw data into features that are suitable for machine learning models**. In other words, it is the process of selecting, extracting, and transforming the most relevant features from the available data to build more accurate and efficient machine learning models.

The success of machine learning models heavily depends on the quality of the features used to train them. Feature engineering involves a set of techniques that enable us to create new features by combining or transforming the existing ones. These techniques help to highlight the most important patterns and relationships in the data, which in turn helps the machine learning model to learn from the data more effectively.



**Why do we Engineer Features?**

We engineer features to improve the performance of machine learning models by providing them with relevant and informative input data. Raw data may contain noise, irrelevant information, or missing values, which can lead to inaccurate or biased model predictions. By engineering features, we can extract meaningful information from the raw data, create new variables that capture important patterns and relationships, and transform the data into a more suitable format for machine learning algorithms.  
 Feature engineering can also help in addressing issues such as overfitting, underfitting, and high dimensionality. For example, by reducing the number of features, we can prevent the model from becoming too complex or overfitting to the training data. By selecting the most relevant features, we can improve the model’s accuracy and interpretability.  
 In addition, feature engineering is a crucial step in preparing data for analysis and decision-making in various fields, such as finance, healthcare, marketing, and social sciences. It can help uncover hidden insights, identify trends and patterns, and support data-driven decision-making.

We engineer features for various reasons, and some of the main reasons include:

* **Improve User Experience:** The primary reason we engineer features is to enhance the user experience of a product or service. By adding new features, we can make the product more intuitive, efficient, and user-friendly, which can increase user satisfaction and engagement.
* **Competitive Advantage:**Another reason we engineer features is to gain a competitive advantage in the marketplace. By offering unique and innovative features, we can differentiate our product from competitors and attract more customers.
* **Meet Customer Needs:**We engineer features to meet the evolving needs of customers. By analyzing user feedback, market trends, and customer behavior, we can identify areas where new features could enhance the product’s value and meet customer needs.
* **Increase Revenue:** Features can also be engineered to generate more revenue. For example, a new feature that streamlines the checkout process can increase sales, or a feature that provides additional functionality could lead to more upsells or cross-sells.
* **Future-Proofing:**Engineering features can also be done to future-proof a product or service. By anticipating future trends and potential customer needs, we can develop features that ensure the product remains relevant and useful in the long term.

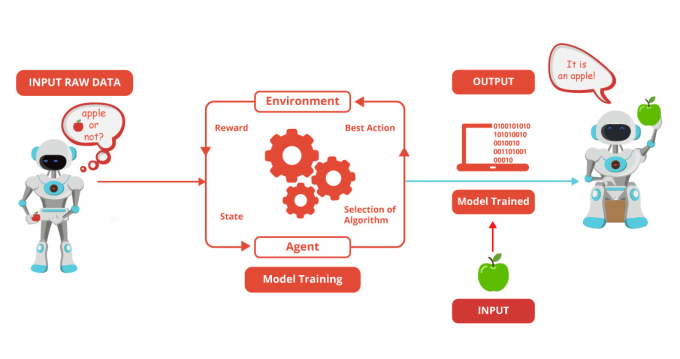
**Model Training :**

A training model is a dataset that is used to train an ML algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model.

This iterative process is called “model fitting”. The accuracy of the training dataset or the validation dataset is critical for the precision of the model.

Model training in machine language is the process of feeding an ML algorithm with data to help identify and learn good values for all attributes involved. There are several types of machine learning models, of which the most common ones are supervised and unsupervised learning.

Supervised learning is possible when the training data contains both the input and output values. Each set of data that has the inputs and the expected output is called a supervisory signal. The training is done based on the deviation of the processed result from the documented result when the inputs are fed into the model.

 Unsupervised learning involves determining patterns in the data. Additional data is then used to fit patterns or clusters. This is also an iterative process that improves the accuracy based on the correlation to the expected patterns or clusters. There is no reference output dataset in this method.

**Evaluation :**

**Crea**Model evaluation in machine learning is the process of determining a model’s performance via a metrics-driven analysis. It can be performed in two ways:

* Offline: The model is evaluated after training during experimentation or [continuous retraining](https://www.iguazio.com/glossary/model-retraining/).
* Online: The model is evaluated in production as part of model monitoring.

The metrics selection for the analysis varies depending on the data, algorithm, and use case.

For supervised learning, the metrics are categorized with respect to classification and regression. Classification metrics are based on the confusion matrix, such as [accuracy](https://www.iguazio.com/glossary/model-accuracy-in-ml/), precision, recall, and f1-score; regression metrics are based on errors, such as mean absolute error (MAE) and root mean squared errors (RMSE).

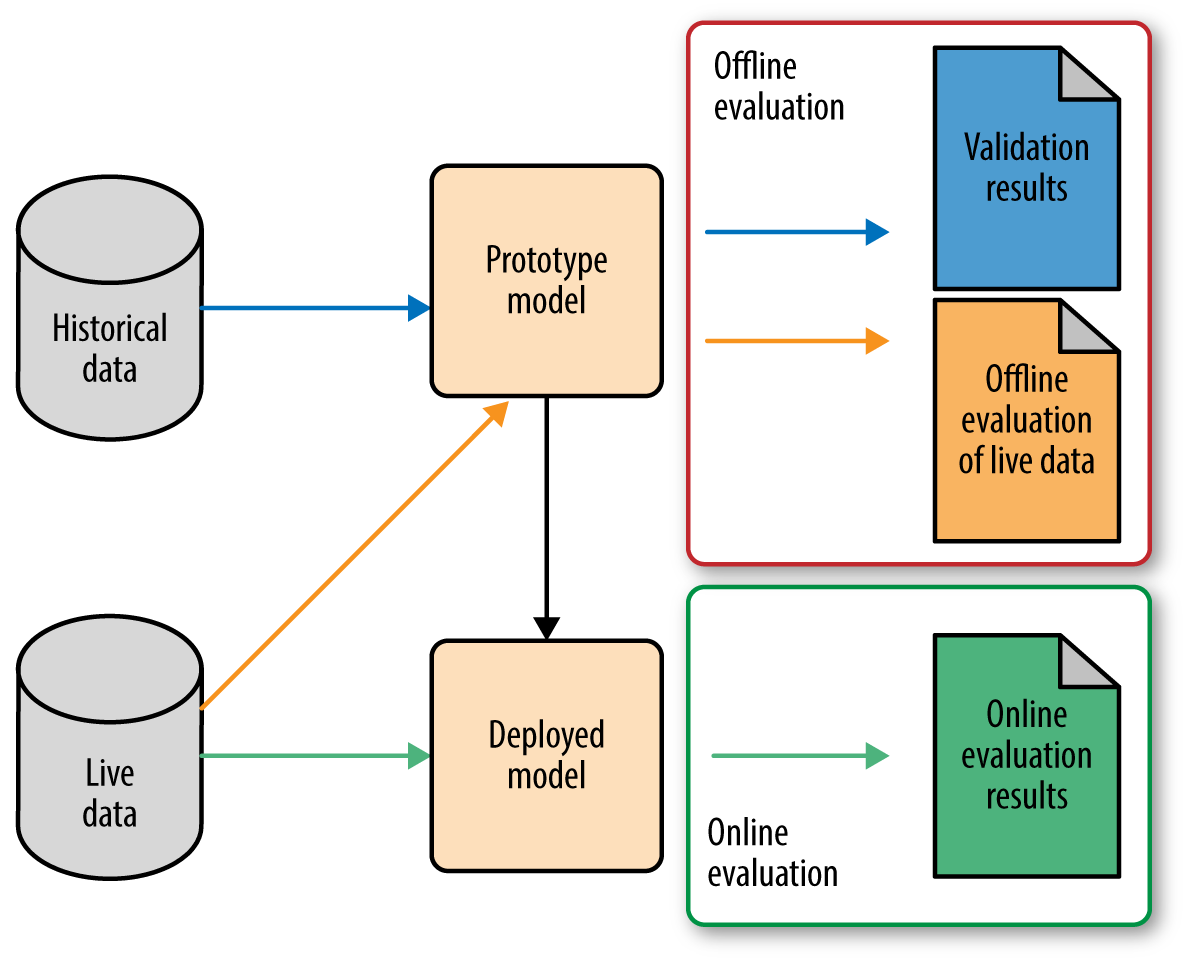
For [unsupervised learning](https://www.iguazio.com/glossary/unsupervised-ml/), the metrics aim to define the cohesion, separation, confidence, and error in the output. For example, the silhouette measure is used for clustering in order to measure how similar a data point is to its own cluster relative to its similarity to other clusters.

For both learning approaches, and necessarily for the latter, model evaluation metrics are extended during experimentation with visualizations and manual analysis of (groups of) data points. Domain experts are often required to support this evaluation.

Beyond technical metrics and analysis, business metrics such as incremental revenue and reduced costs should also be defined and reported. This allows an understanding of the impact of putting the model into production.

Domain experts are often required to support this evaluation. This allows an understanding of the impact of putting the model into production.

* Optimal: The productionized model(s) performs as well as is currently achievable, typically in comparison to multiple other trained models.
* Reliable: The productionized model(s) behaves as expected. The behavioral profile of the model is an in-depth review of how the model maps inputs to outputs—overall and with respect to specific data slices—as defined by feature contribution, counterfactual analysis, and fairness tests.

**Conclusion:**

**T** Machine Learning can be a Supervised or Unsupervised. If you have lesser amount of data and clearly labelled data for training, opt for Supervised Learning. Unsupervised Learning would generally give better performance and results for large data sets. If you have a huge data set easily available, go for deep learning techniques. You also have learned Reinforcement Learning and Deep Reinforcement Learning. You now know what Neural Networks are, their applications and limitations.

Finally, when it comes to the development of machine learning models of your own, you looked at the choices of various development languages, IDEs and Platforms. Next thing that you need to do is start learning and practicing each machine learning technique. The subject is vast, it means that there is width, but if you consider the depth, each topic can be learned in a few hours. Each topic is independent of each other. You need to take into consideration one topic at a time, learn it, practice it and implement the algorithm/s in it using a language choice of yours. This is the best way to start studying Machine Learning. Practicing one topic at a time, very soon you would acquire the width that is eventually required of a Machine Learning expert.

## g A Model In Machine Learnin