**Predicting IMDb Scores Using Machine Learning**

TEAM MEMBER

732521104049 : SUBIKSHA V

**Phase 2 Submission Document**

**Project :** Predicting IMDb Scores



This is a project based on Predicting IDMb scores using a dataset.

Introduction:

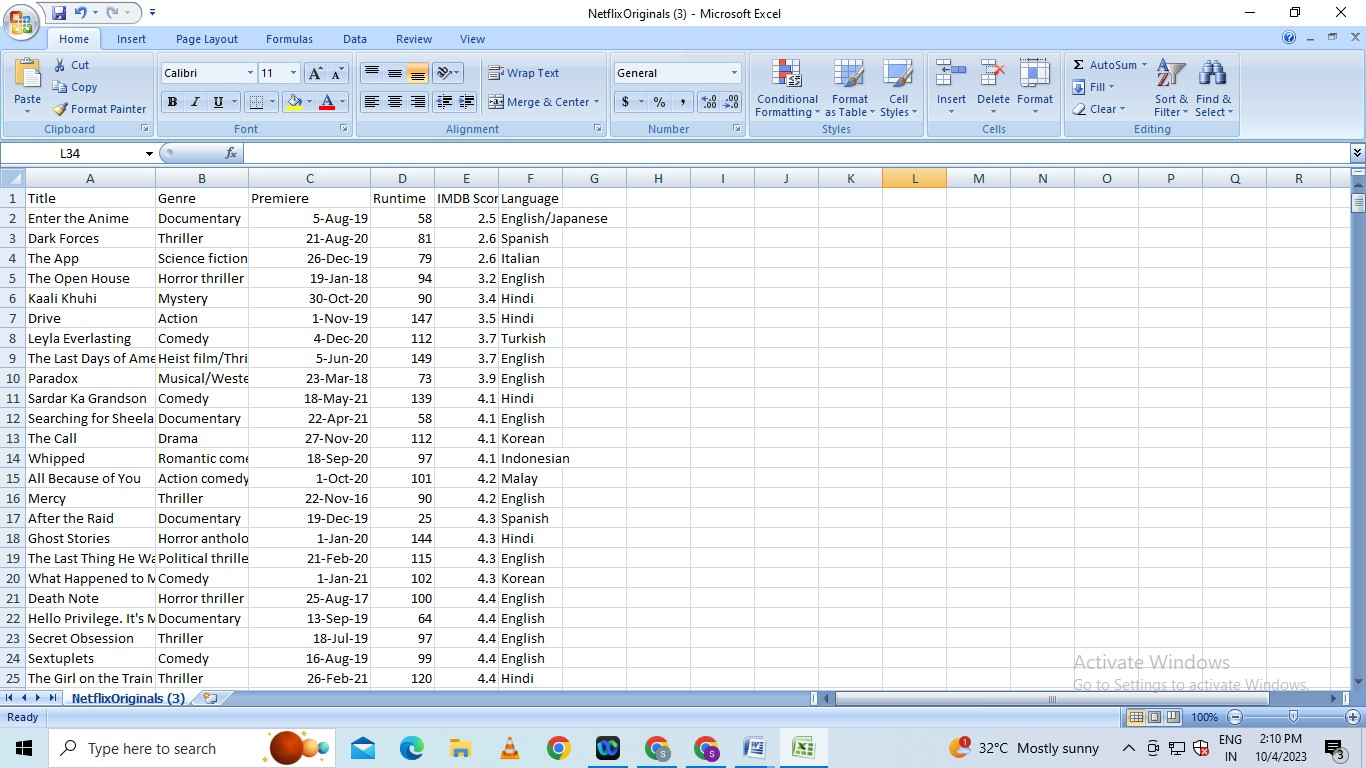
* Predicting IMDb scores for movies or TV shows typically involves using machine learning models and features such as cast, crew, genre, user reviews, and more. You can use regression algorithms to build a predictive model.
* The quality of your predictions depends on the quality and quantity of data, 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 2 :**

Consider exploring advanced regression technique like Gradient Boosting or Neural Networks for improved Prediction accuracy.

**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))

**Model Evaluation and Selection:**

* Split the dataset into training and testing sets.
* Evaluate models using appropriate metrics (e.g., Mean Absolute Error, Mean Squared Error, R-squared) to assess their performance.
* Use cross-validation techniques to tune hyperparameters and ensure model stability. Compare the results with traditional linear regression models to highlight improvements.
* Select the best-performing model for further analysis.

**Model Interpretability:**

* Explain how to interpret feature importance from Gradient Boosting and Neural Networks.
* Discuss the insights gained from feature importance analysis and their relevance to IDMb scores prediction.
* Interpret feature importance from ensemble models like Random Forest and Gradient
* Boosting to understand the factors influencing IDMb.

**Deployment and Prediction:**

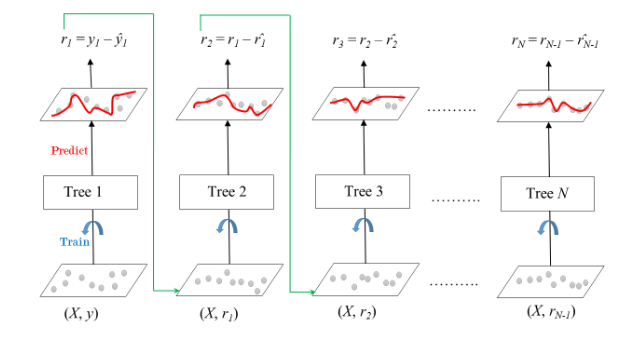
* Deploy the chosen regression model to predict IDMb.
* Develop a user-friendly interface for users to input property features and receive IDMb scores predictions.

****

**Steps to Gradient Boosting**

Gradient boosting classifier requires these steps:

* Fit the model
* Adapt the model's Hyperparameters and Parameters.
* Make forecasts
* Interpret the findings

****

**Program:**

**IDMb Score Prediction**

import pandas as pd

import numpy as np

From sklearn.metrics import classification\_report

from sklearn.datasets import load\_breast\_cancer

from sklearn.ensemble import GradientBoostingClassifier

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

df = pd.DataFrame(load\_breast\_cancer()['data'],

columns=load\_breast\_cancer()['feature\_names'])

df['y'] = load\_breast\_cancer()['target']

df.head(5)

X,y = df.drop('y',axis=1),df.y

test\_size = 0.30 # taking 70:30 training and test set

seed = 7 # Random number seeding for repeatability of the code

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

gradient\_booster = GradientBoostingClassifier(learning\_rate=0.1)

Gradient boosting classifiers are required to implement gradient boosting.

gradient\_booster.fit(X\_train,y\_train)

The training dataset must now be used to fit the model; if the data is appropriately fitted, it will result in good accuracy.

print(classification\_report(y\_val,gradient\_booster.predict(X\_val)))

## Advantages and Disadvantages of Gradient Boost

### Advantages:

* Frequently has remarkable forecasting accuracy.
* Numerous choices for hyperparameter adjustment and the ability to optimize various loss functions.
* It frequently works well with numerical and categorical values without pre-processing the input.
* Deals with missing data; imputation is not necessary.

### Disadvantages:

* Gradient Boosting classifier will keep getting better to reduce all inaccuracies. This may lead to overfitting and an overemphasis on outliers.
* Costly to compute since it frequently requires a large number of trees (>1000), which can be memory and time-consuming.
* Due to the high degree of flexibility, numerous variables interact and significantly affect how the technique behaves.
* Less interpretative, even though this can be easily corrected with several tools.

**Conclusion and Future Work (Phase 2):**

**Project Conclusion:**

It is possible to use a gradient boosting classifier, which is a strong algorithm, for classification and regression problems. On extremely complicated datasets, gradient boosting models can perform remarkably well, but they are also prone to overfitting, which can be avoided using several techniques.