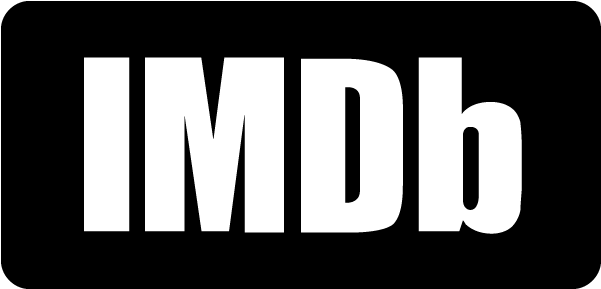
**Predicting IMDb Scores Using Machine Learning**

TEAM MEMBER

732521104305 : POORANA G

**Phase 2 Submission Document**

**Project :** Predicting IMDb Scores



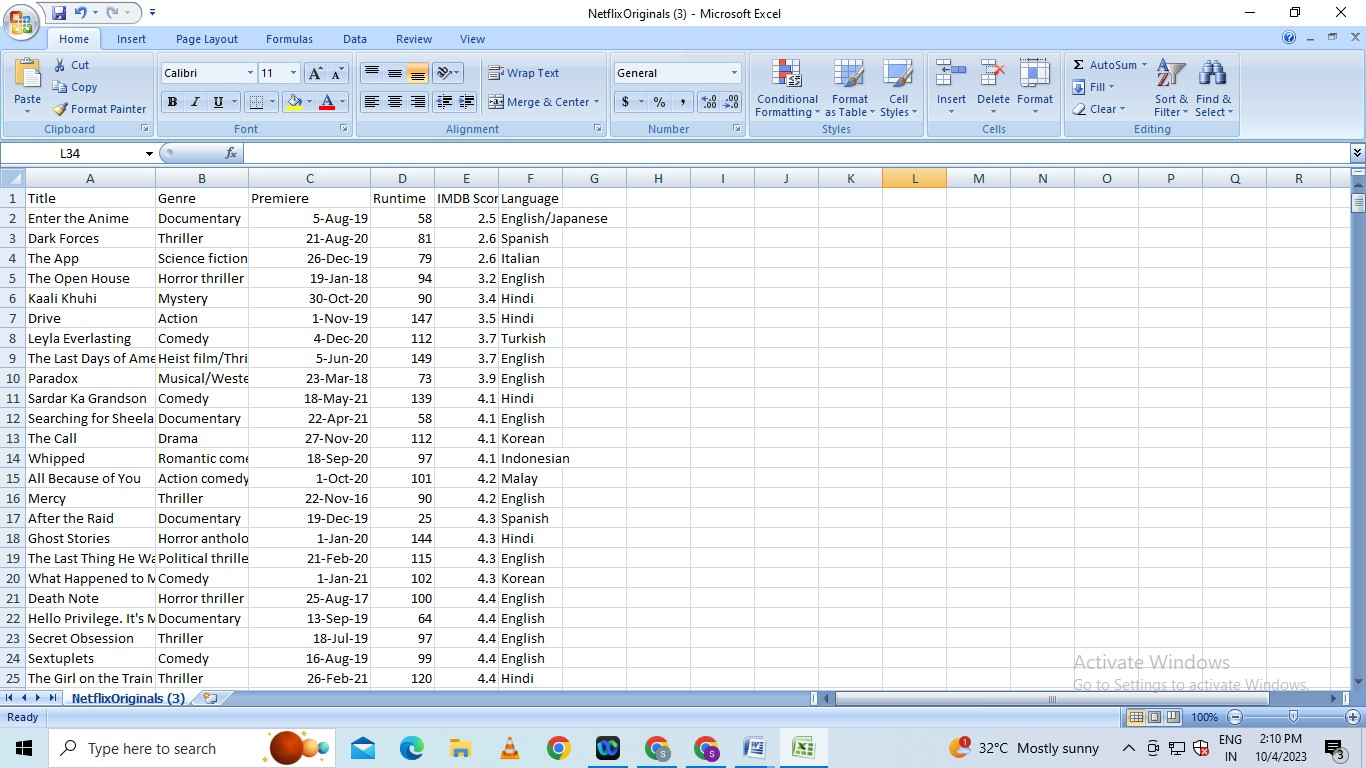
Introduction:

* 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 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))

**Intuition :**

* Gradient Boosting relies on the intuition that the best possible next model , when combined with the previous models, minimizes the overall prediction errors.
* The key idea is to set the target outcomes from the previous models to the next model in order to minimize the errors. This is another boosting algorithm(few others are Adaboost, XGBoost etc.).

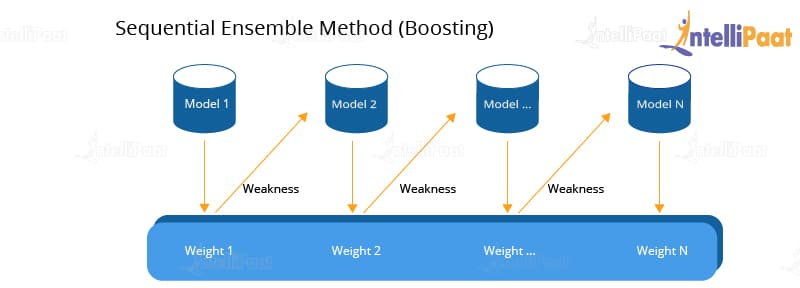
**Why do we need Boosting?**

Before learning gradient boosting technique lets understand the need for boosting with the help of a scenario. Suppose, we have separately built six Machine Learning models for predicting whether it will rain or not. Each of these models has been built on top of the 6 distinct parameters given below to analyze and predict the weather condition:

1. Air temperature
2. Atmospheric (barometric) pressure
3. Humidity
4. Precipitation
5. Solar radiation
6. Wind

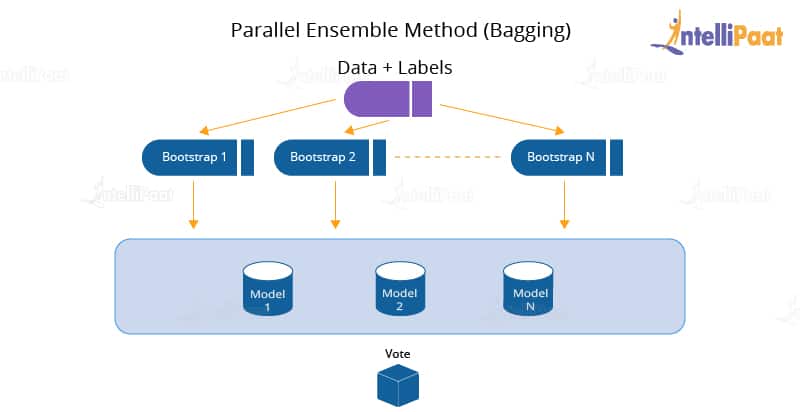
### ****1. Sequential Ensemble Learning****

It is a boosting technique where the outputs from individual weak learners associate sequentially during the training phase. The performance of the model is boosted by assigning higher weights to the samples that are incorrectly classified. AdaBoost algorithm is an example of sequential learning that we will learn later in this blog.



### ****2. Parallel Ensemble Learning****

It is a bagging technique where the outputs from the weak learners are generated parallelly. It reduces errors by averaging the outputs from all weak learners. The [random forest algorithm](https://intellipaat.com/blog/what-is-random-forest-algorithm-in-python/) is an example of parallel ensemble learning.



***Go through this***[***Machine Learning Course***](https://intellipaat.com/machine-learning-certification-training-course/)***to get a clear understanding of Machine Learning!***

### ****Mechanism of Boosting Algorithms****

Boosting is creating a generic [**algorithm**](https://intellipaat.com/blog/what-is-an-algorithm/) by considering the prediction of the majority of weak learners. It helps in increasing the prediction power of the Machine Learning model. This is done by training a series of weak models.

Below are the steps that show the mechanism of the boosting algorithm:

1. Reading data

2. Assigning weights to observations

3. Identification of misinterpretation (false prediction)

4. Assigning the false prediction, along with a higher weightage, to the next learner

5. Finally, iterating Step 2 until we get the correctly classified output

## ****Types of Boosting Algorithms****

Basically, there are three types of boosting algorithms discussed as below:

### ****1. Adaptive Boosting (AdaBoost)****

Adaptive boosting is a technique used for binary classification. For implementing AdaBoost, we use short decision trees as weak learners.

**Steps for implementing AdaBoost**:

1. Train the base model using the weighted training data

2. Then, add weak learners sequentially to make it a strong learner

3. Each weak learner consists of a decision tree; analyze the output of each decision tree and assign higher weights to the misclassified results. This gives more significance to the prediction with higher weights.

4. Continue the process until the model becomes capable of predicting the accurate result

Next, we will see “What is Gradient Boosting?.”

### ****2. Gradient Boosting****

In Machine Learning, we use gradient boosting to solve [**classification**](https://intellipaat.com/blog/tutorial/machine-learning-tutorial/classification-machine-learning/) and regression problems. It is a sequential ensemble learning technique where the performance of the model improves over iterations. This method creates the model in a stage-wise fashion. It infers the model by enabling the optimization of an absolute differentiable loss function. As we add each weak learner, a new model is created that gives a more precise estimation of the response variable.

The gradient boosting algorithm requires the below components to function:

1. **Loss function**: To reduce errors in prediction, we need to optimize the loss function. Unlike in AdaBoost, the incorrect result is not given a higher weightage in gradient boosting. It tries to reduce the loss function by averaging the outputs from weak learners.

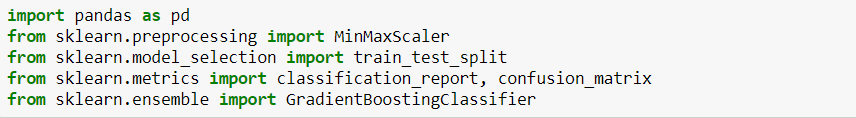
2. **Weak learner**: In gradient boosting, we require weak learners to make predictions. To get real values as output, we use regression trees. To get the most suitable split point, we create trees in a greedy manner, due to this the model overfits the dataset

## ****Implementation of Gradient Boosting :****

In this section, we will look into the implementation of the gradient boosting algorithm. For this, we will use the Titanic dataset.

Here are the steps of implementation:

1. Importing the required libraries



2. Loading the dataset

https://intellipaat.com/blog/wp-content/uploads/2020/01/t2.png

3. Performing data preprocessing

https://intellipaat.com/blog/wp-content/uploads/2020/01/t3.png

4. Concatenating a new dataset

https://intellipaat.com/blog/wp-content/uploads/2020/01/t4.png

5. Dropping the columns that are not required

https://intellipaat.com/blog/wp-content/uploads/2020/01/t5.png

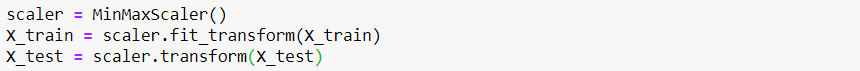
6. Assigning empty sets a value of 0

https://intellipaat.com/blog/wp-content/uploads/2020/01/t6.png

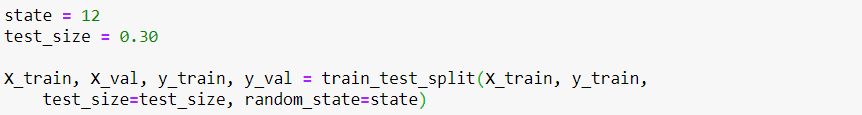
7. Splitting the data into train and test sets

https://intellipaat.com/blog/wp-content/uploads/2020/01/t7.png

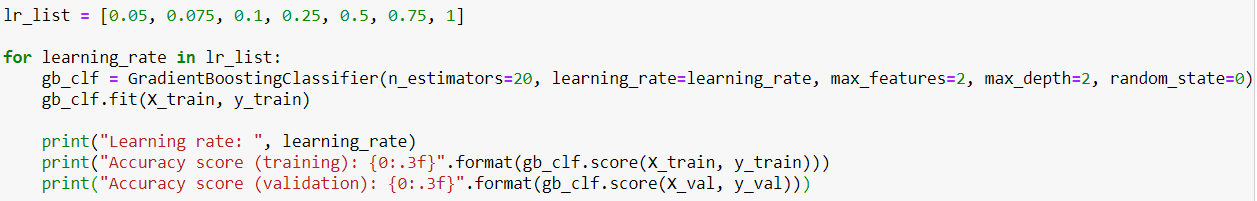
8. Scaling the data using MinMaxScaler



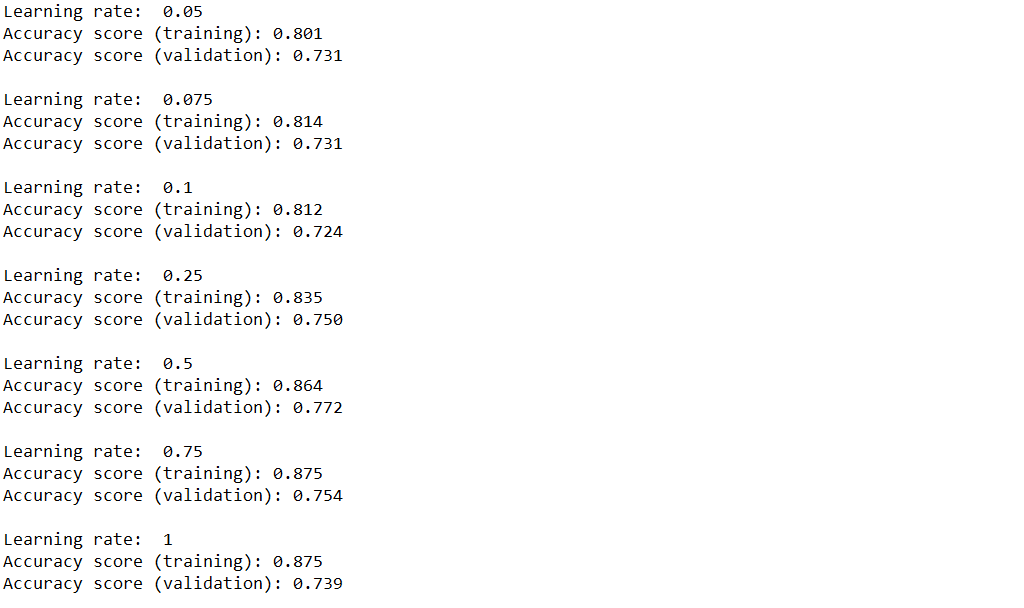
9. Selecting the size of the dataset for testing



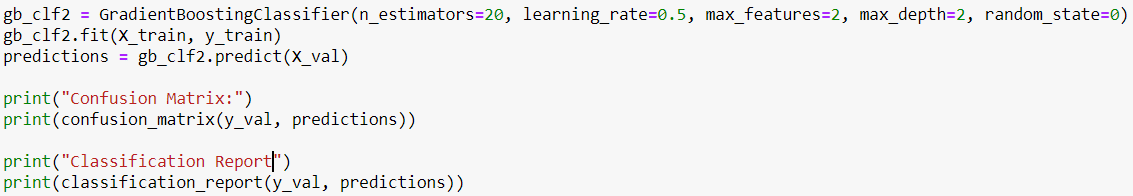
10. Assigning the learning rate to evaluate the classifier’s performance



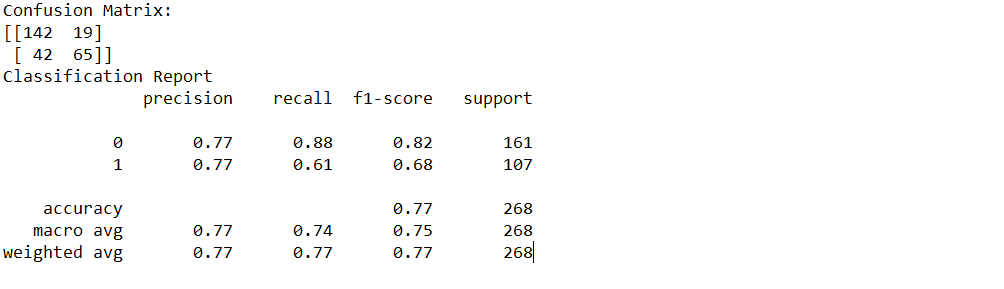
Performance of different learning rates:



11. Creating a new gradient boosting classifier and building a [**confusion matrix**](https://intellipaat.com/blog/confusion-matrix-python/) for checking accuracy



Output:



**Advantages of gradient boosting trees**

There are several reasons as to why you would consider using gradient boosting tree algorithms:

* generally more accurate compare to other modes,
* train faster especially on larger datasets,
* most of them provide support handling categorical features,
* some of them handle missing values natively.

**Disadvantages of gradient boosting trees :**

Let’s now address some of the challenges faced when using gradient boosted trees:

* prone to overfitting: this can be solved by applying L1 and L2 regularization penalties. You can try a low learning rate as well;
* models can be computationally expensive and take a long time to train, especially on CPUs;
* hard to interpret the final models

# ****Conclusion :****

* Gradient Boosting algorithm is very widely used machine learning and predictive modeling technique (Preferred in Kaggle and other code competitions).

Linked In : <https://www.linkedin.com/in/anjani-kumar-9b969a39/>

In the Phase 2 conclusion, we will summarize the key findings and insights from the advanced regression techniques. We will reiterate the impact of these techniques on improving the accuracy and robustness of IDMb score predictions.

Future Work: We will discuss potential avenues for future work, such as incorporating additional data sources (e.g., real-time economic indicators), exploring deep learning models for prediction, or expanding the project into a web application with more features and interactivity.