**Predicting IMDb Scores Using Machine Learning**

TEAM MEMBER

732521104053 : THAMARAISELVI A

**Phase 2 Submission Document**

**Project :** Predicting IMDb Scores



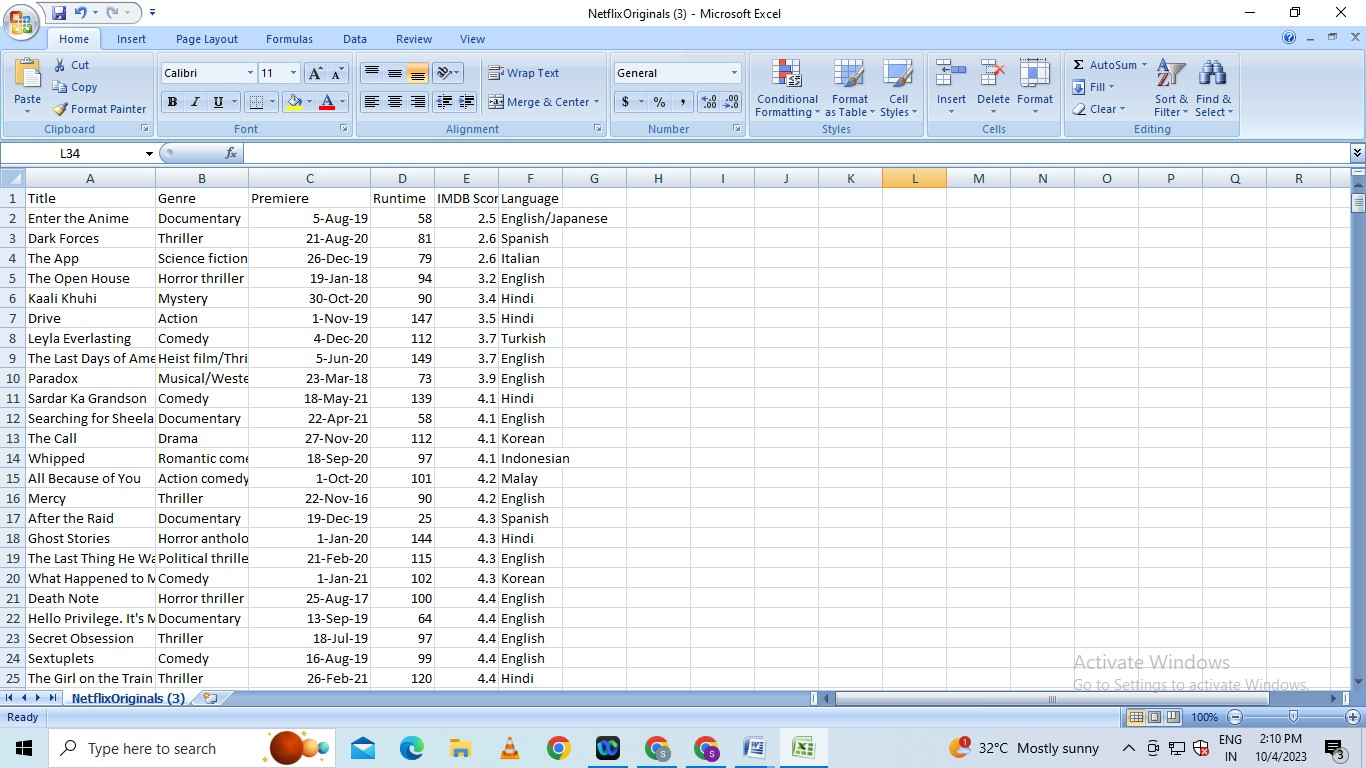
Introduction:

* 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 :**

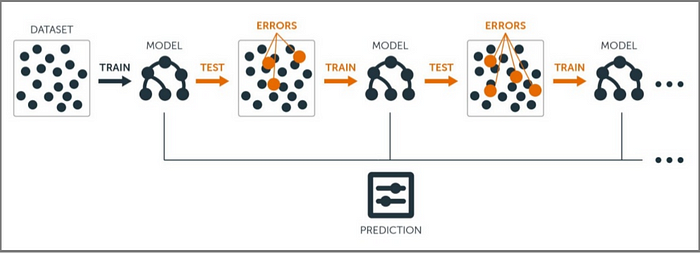
* Consider exploring advanced regression technique like Gradient Boosting or Neural Networks for improved Prediction accuracy.

**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.).



**Input requirement for Gradient Boosting:**

* A Loss Function to optimize.
* A weak learner to make prediction(Generally Decision tree).
* An additive model to add weak learners to minimize the loss function.

Loss function :

The loss function basically tells how my algorithm, models the data set.In simple terms it is difference between actual values and predicted values.

**Regression Loss functions:**

* L1 loss or Mean Absolute Errors (MAE)
* L2 Loss or Mean Square Error(MSE)
* Quadratic Loss

**Binary Classification Loss Functions:**

* Binary Cross Entropy Loss
* Hinge Loss
* A gradient descent procedure is used to minimize the loss when adding trees.

## 2. Weak Learner :

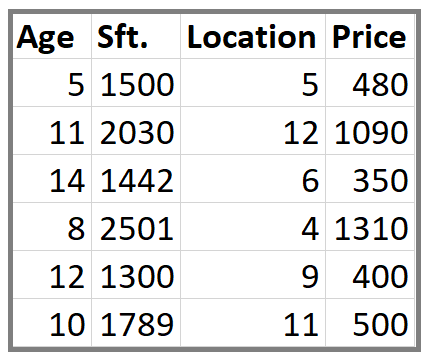
* Weak learners are the models which is used sequentially to reduce the error generated from the previous models and to return a strong model on the end.
* Decision trees are used as weak learner in gradient boosting algorithm.

## 3. Additive Model :

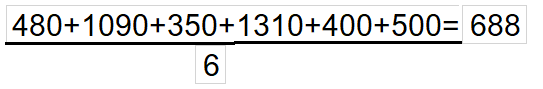
* In gradient boosting, decision trees are added one at a time (in sequence), and existing trees in the model are not changed.

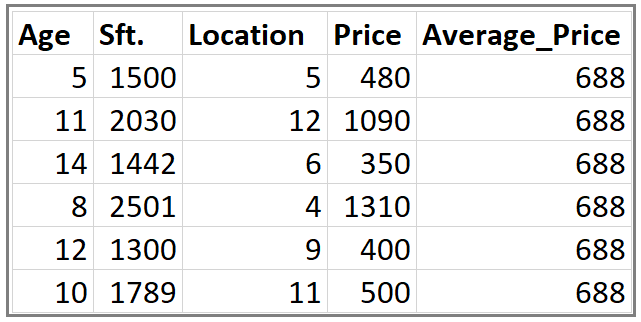
# Understanding Gradient Boosting Step by Step :

* This is our data set. Here Age, Sft., Location is independent variables and Price is dependent variable or Target variable.



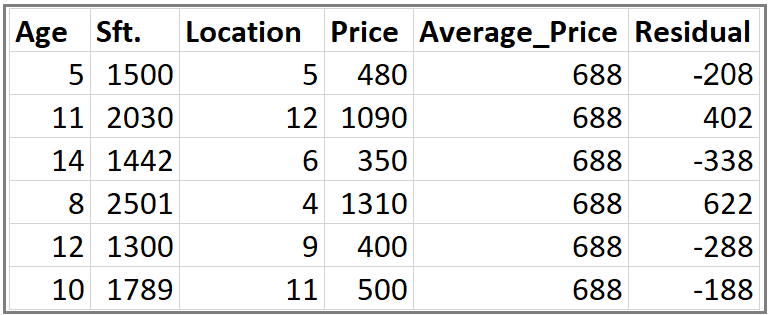
**Step 1**: Calculate the average/mean of the target variable.



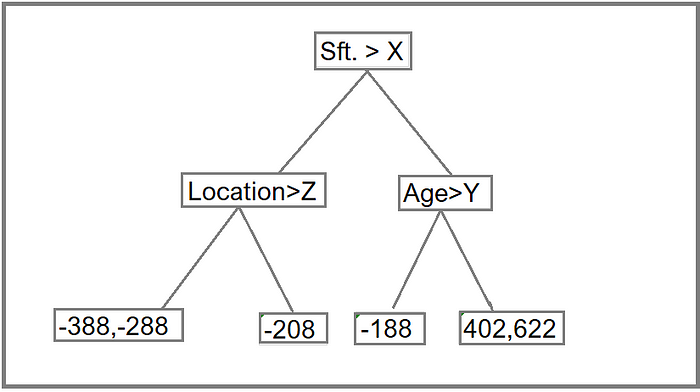


**Step 2**: Calculate the residuals for each sample.

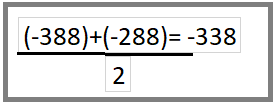
https://miro.medium.com/v2/resize:fit:558/1*zAPsBPDiS82nLBFOOsL9gQ.png

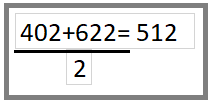


**Step 3:**Construct a decision tree. We build a tree with the goal of predicting the Residuals.

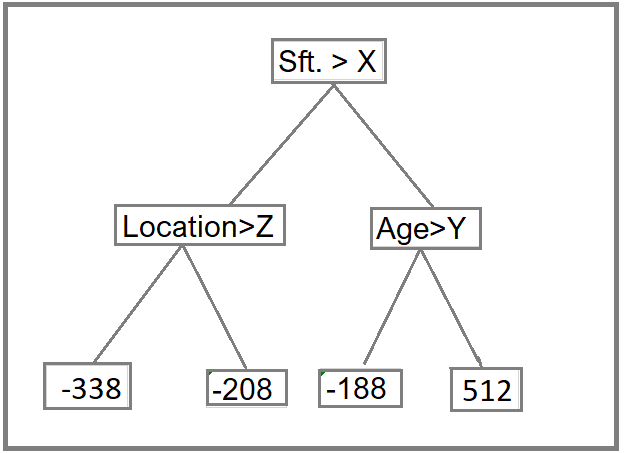


* In the event if there are more residuals then leaf nodes(here its 6 residuals),some residuals will end up inside the same leaf. When this happens, we compute their average and place that inside the leaf.





After this tree become like this.



**Step 4**: Predict the target label using all the trees within the ensemble.

Each sample passes through the decision nodes of the newly formed tree until it reaches a given lead. The residual in the said leaf is used to predict the house price.

https://miro.medium.com/v2/resize:fit:700/1*jgFL4ajDbulZ3ON098boSg.png

https://miro.medium.com/v2/resize:fit:302/1*Bp01OWZAkGr67bqGVNUgdg.png

https://miro.medium.com/v2/resize:fit:360/1*81Rcj45XL8xFUm_p0ym5nQ.png

https://miro.medium.com/v2/resize:fit:279/1*6iN4MRQb1YQItPF-jqFeWw.png

https://miro.medium.com/v2/resize:fit:354/1*I8NnPSldKzcy7H8OUZpQFQ.png

**Calculation above for Residual value (-338) and (-208) in Step 2**

* Same way we will calculate the **Predicted Price** for other values

**Note:** We have initially taken 0.1 as learning rate.

**Step 5** : Compute the new residuals

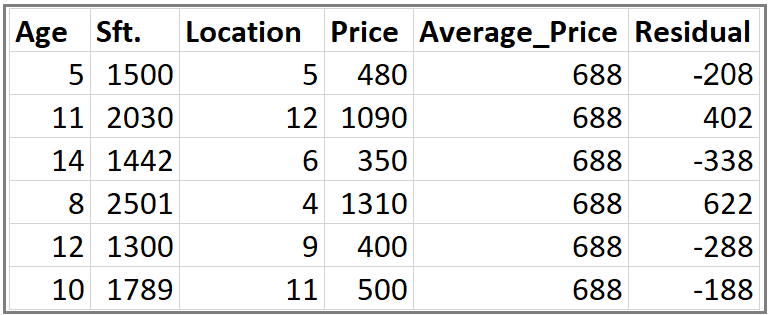
https://miro.medium.com/v2/resize:fit:545/1*HwKAAuhdbIR6ihQQ2C9vRw.png

https://miro.medium.com/v2/resize:fit:325/1*1CCravSOh8PMdV3VIzrNBw.png

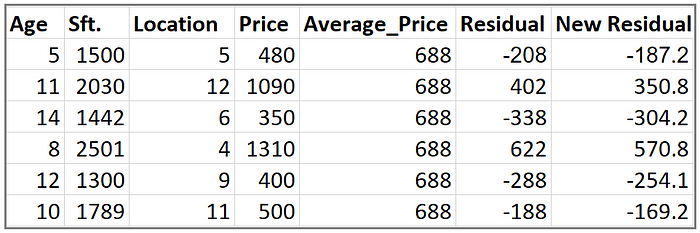
https://miro.medium.com/v2/resize:fit:308/1*RjYwwnFyFJY4AVN0vWCrKw.png

**When Price is 350 and 480 Respectively.**

* With our Single leaf with average value**(688)**we have the below column of Residual.



With our decision tree ,we ended up the below new residuals.



**Step 6**: Repeat steps 3 to 5 until the number of iterations matches the number specified by the hyper parameter(numbers of estimators)

**Step 7**: Once trained, use all of the trees in the ensemble to make a final prediction as to value of the target variable. The final prediction will be equal to the mean we computed in Step 1 plus all the residuals predicted by the trees that make up the forest multiplied by the learning rate.

Here,

**LR** : Learning Rate

**DT**: Decision Tree

https://miro.medium.com/v2/resize:fit:700/1*hOOvQbw9Jd_4yTc44wuPoA.png

https://miro.medium.com/v2/resize:fit:539/1*aKXj5kI4Yy2i6pXKysDS5g.png

# ****Gradient Boosting Code Implementation in Python :****

# Advantages :

Most of the time predictive accuracy of gradient boosting algorithm on higher side.

* It provides lots of flexibility and can optimize on different loss functions and provides several hyper parameter tuning options that make the function fit very flexible.
* Most of the time no data pre-processing required.
* Gradient Boosting algorithm works great with categorical and numerical data.
* Handles missing data — missing value imputation not required
* Gradient Boosting algorithm works great with categorical and numerical data.
* Handles missing data — missing value imputation not required

disadvantages :

# Gradient Boosting Models will continue improving to minimize all errors. This can overemphasize outliers and cause over fitting. Must use cross-validation to neutralize.

* It is computationally very expensive — GBMs often require many trees (>1000) which can be time and memory exhaustive.
* The high flexibility results in many parameters that interact and influence heavily the behavior of the approach (number of iterations, tree depth, regularization parameters, etc.). This requires a large grid search during tuning.

# ****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.