# TEAM 4:

Sharvari Karnik

Jayesh Patil

Kunal Jaiswal

Srividya Balaji

# **Movie Success Revenue Prediction**

### Background

A commercial success movie not only entertains the audience, but also enables film companies to gain tremendous profit. A lot of factors such as good directors, experienced actors are considerable for creating good movies. However, famous directors and actors can always bring an expected box-office income but cannot guarantee a highly rated imdb score. In this project, we will try to consider all these factors and predict the chances of a movie’s success.

### Data Description

The dataset is from the Kaggle website. It contains 28 variables for 5043 movies, spanning across 100 years in 66 countries. There are 2399 unique director names, and thousands of actors/actresses. “imdb\_score” is the response variable while the other 27 variables are possible predictors.

In this project, we take IMDB scores as response variables and focus on operating predictions by analyzing the rest of variables in the IMDB 5000 movie data. The results can help film companies to understand the secret of generating a commercial success movie.

https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset

### **Algorithms Implemented:**

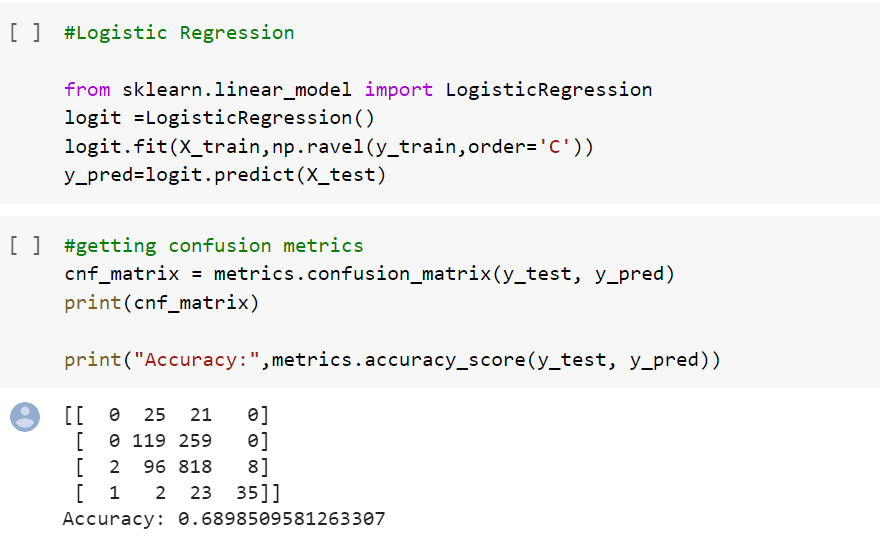
**Logistic Regression:**

Logistic Regression was used in the biological sciences in the early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.

For example,

* To predict whether an email is spam (1) or (0)
* Whether the tumor is malignant (1) or not (0)

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequences in real time.



**KNN:**

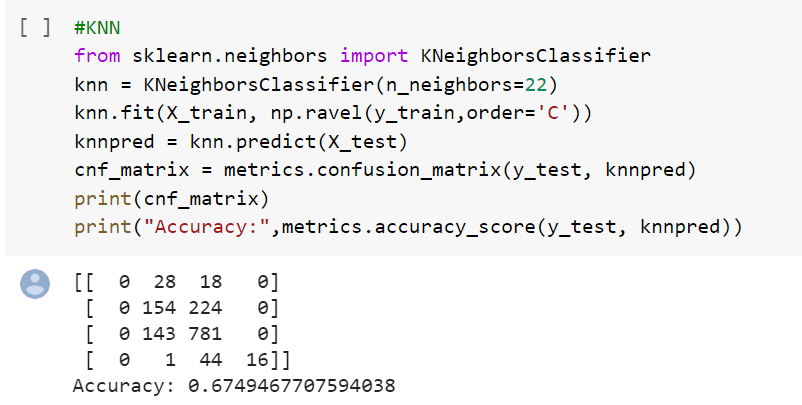
***K*-nearest neighbors algorithm** is a non-parametric method used for classification and regression. In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

In *k-NN classification*, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.

In *k-NN regression*, the output is the property value for the object. This value is the average of the values of *k* nearest neighbors.

*k*-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation.

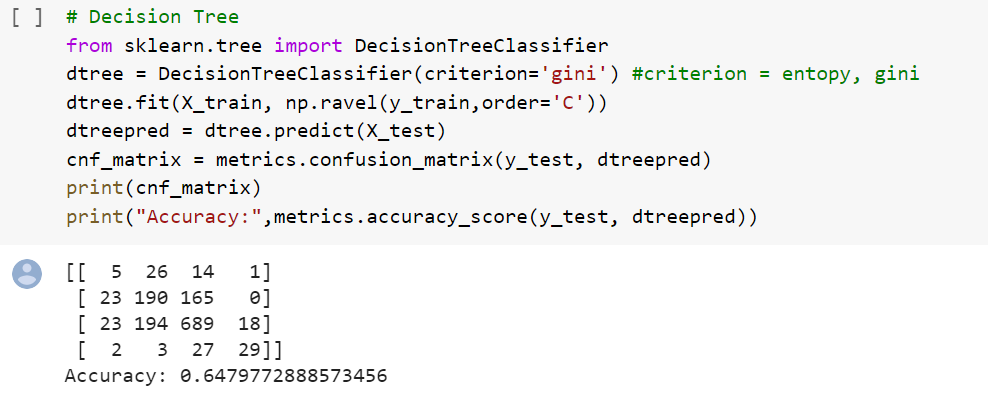
Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/*d*, where *d* is the distance to the neighbor.



**Decision Tree:**

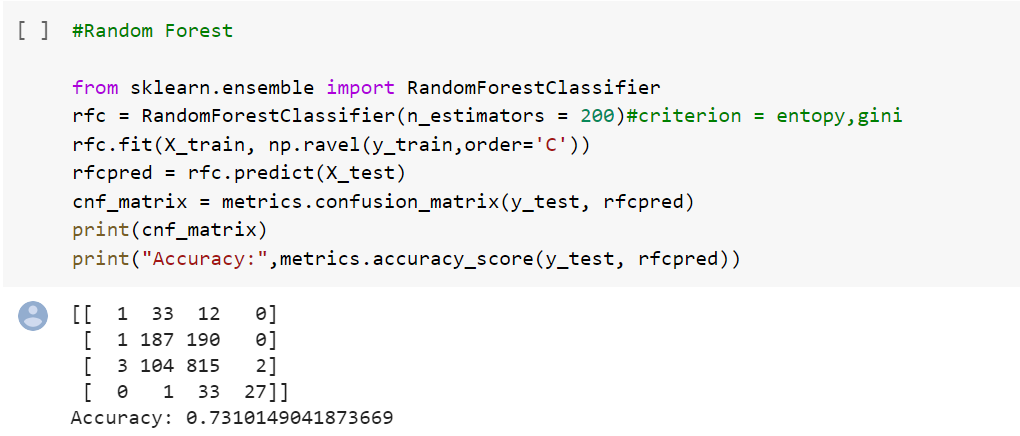
A **decision tree** is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

A decision tree is a flowchart-like structure in which each internal node represents a “test” on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.



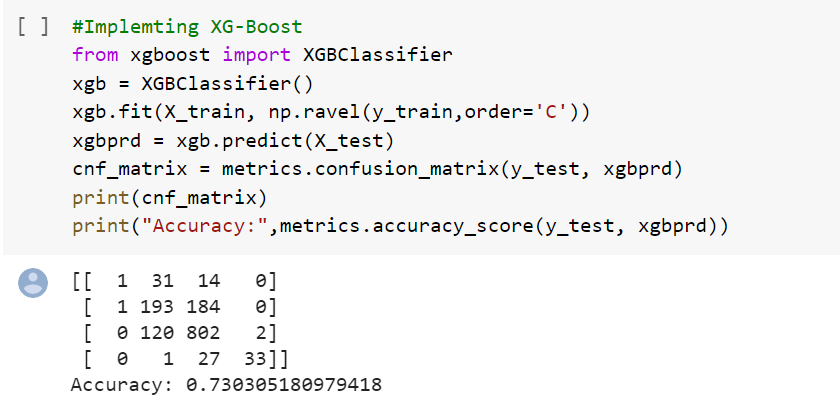
**Random Forest:**

Random Forest algorithm is a supervised classification algorithm. We can see it from its name, which is to create a forest by some way and make it random. There is a direct relationship between the number of trees in the forest and the results it can get: the larger the number of trees, the more accurate the result. But one thing to note is that creating the forest is not the same as constructing the decision with information gain or gain index approach.



**XG-Boost:**

XGBoostis a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now.



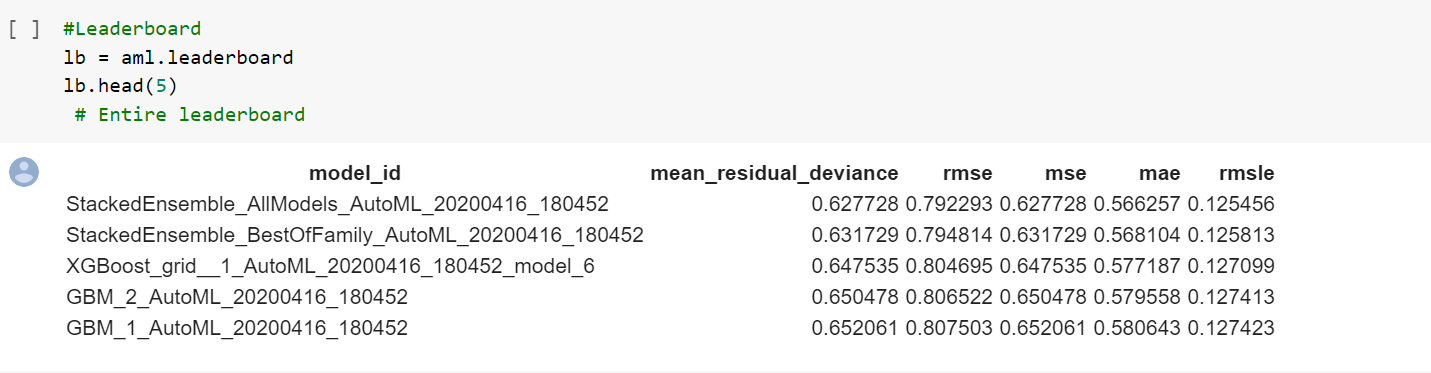
### 

### **h2o AutoML:**

**H2O** is a fully open-source, distributed in-memory machine learning platform with linear scalability. H2O supports the most widely used statistical & machine learning algorithms, including gradient boosted machines, generalized linear models, deep learning, and many more.

H2O also has an industry-leading AutoML functionality (available in H2O ≥3.14) that automates the process of building a large number of models, to find the “**best**” model without any prior knowledge or effort by the Data Scientist. H2O AutoML can be used for automating the machine learning workflow, which includes automatic training and tuning of many models within a user-specified time-limit.

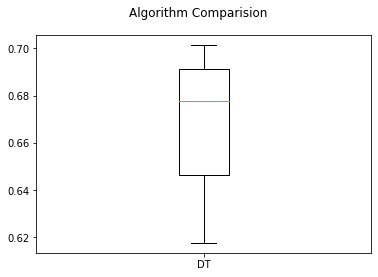
H2O’s AutoML can also be a helpful tool for the novice as well as advanced users. It provides a simple wrapper function that performs a large number of modeling-related tasks that would typically require many lines of code. This essentially frees up the time to focus on other aspects of the data science pipeline, such as data preprocessing, feature engineering, and model deployment.



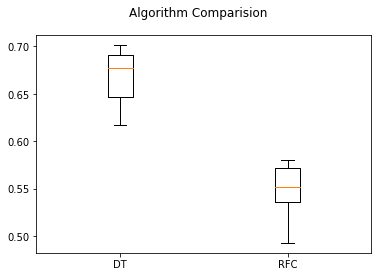
**Final Results:**

**Cross-validation Score of models:**

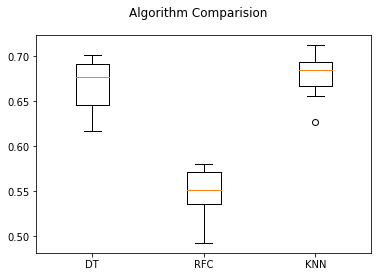
DT: 0.667371 (0.000990)



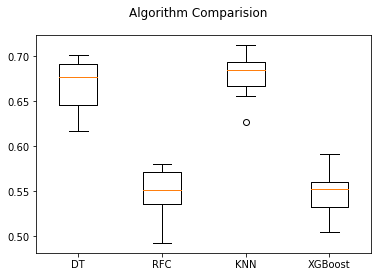
RFC: 0.548825 (0.000953)



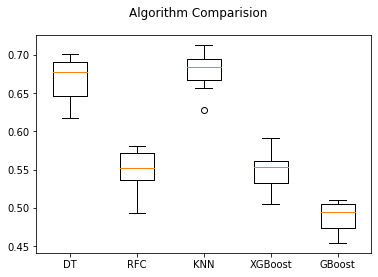
KNN: 0.678094 (0.000805)



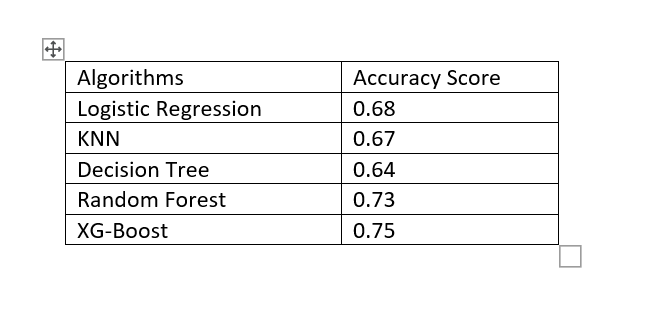
XGBoost: 0.548184 (0.000780)



GBoost: 0.488326 (0.000458)



Accuracy-



Citations:

1. <https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset>
2. <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html>
3. <https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb>
4. <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>
5. <https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d>

## 