**YOUTUBE VIDEO POPULARITY PREDICTION SYSTEM**

**WITH LINEAR REGRESSION MODELS**

**Initial paper-** [**https://ieeexplore.ieee.org/document/7450136/authors#authors**](https://ieeexplore.ieee.org/document/7450136/authors#authors)

**colab notebook-**[**https://colab.research.google.com/drive/1J2\_vKWgv2PHvPVRak2RCOthdeNZ5A66M**](https://colab.research.google.com/drive/1J2_vKWgv2PHvPVRak2RCOthdeNZ5A66M)

## TEAM MEMBER CONTRIBUTIONS:-

| **CLASS S.NO.** | **ENROL NO.** | **TEAM MEMBER** | **WORK ASSIGNED** | **WORK DONE** | **REMARKS** |
| --- | --- | --- | --- | --- | --- |
| **14** | 01596303121 | Daksh Singh | * Attribute fusion * Documentation * Changed the layers and weights , ran the results | YES | Completed all works assigned. |
| **22** | 02396303121 | Aditya Kumar | * Text attribute extraction * Data collection * Data preprocessing * Changed learning rates and ran the model and computed the results | YES | Completed all works assigned. |
| **52** | 35396303121 | Uday Sangwan | * working on dataset * extraction * converting dataset into form of vectors | YES | Completed all works assigned. |

## 

**ABSTRACT-**

A vast quantity of video content floods the internet, yet user attention tends to gravitate asymmetrically, with only a handful of videos gaining widespread notice while the majority languish in obscurity. Consequently, comprehending the factors that contribute to the popularity of online videos and forecasting the future popularity of individual ones holds considerable significance. This understanding bears direct relevance in various domains, including service design, advertising strategy, and network management.

Addressing this challenge, we propose a model capable of capturing the dynamics of video popularity, leveraging early popularity trends and predicting future bursts in popularity. Through validation on extensive real-world data, our approach demonstrates notable improvements, yielding reductions in relative prediction errors of up to 32.73% and 11.28% when compared to two state-of-the-art baseline models, respectively. Finally, we conduct an analysis of the potential and limitations of model parameters in practical applications.

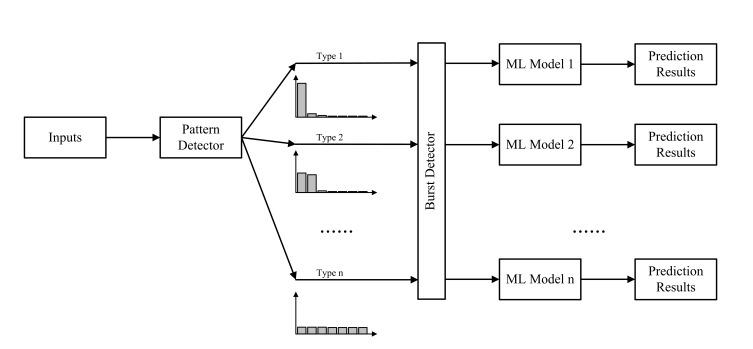
**1. INTRODUCTION-**

The widespread accessibility of the internet and the evolution of Web 2.0 have triggered an explosion of online content, particularly in the form of videos. These videos have become the dominant force in internet traffic, constituting 64% in 2014 and projected to surge to 80% by 2019. However, this immense volume intensifies the competition for user attention, creating a "winner-take-all" scenario where only a small fraction of videos attract the majority of views, leaving the rest unnoticed.

Understanding the dynamics of online video popularity and accurately predicting individual video trends have become paramount for various stakeholders. Service providers can refine content filtering, ranking, and recommendation systems to help users discover valuable videos amidst the sea of content. Advertisers stand to optimize revenue by predicting emerging internet stars for strategic ad placement. Network operators can streamline bandwidth management and proactively deploy cache servers to accommodate the surge in demand for trending videos. Additionally, in resource-constrained environments such as mobile devices, the ability to predict popular videos is indispensable for efficient content delivery, caching, and replication strategies.

* 1. **CHALLENGES AND PROBLEM FACED-**
* Data Quality and Availability: Obtaining reliable and consistent data on soil nutrients and weather conditions can be problematic due to limitations in monitoring infrastructure and data collection methods. Incomplete or inaccurate data can lead to unreliable predictions.
* Feature Engineering: Extracting meaningful features from video metadata, such as views, likes, comments, duration, upload time, and engagement metrics, requires careful consideration. Choosing the right features and representing them effectively for the machine learning model is essential for accurate predictions.
* Model Selection and Tuning: Selecting the appropriate machine learning algorithms and architectures for video popularity prediction is non-trivial. Experimenting with various models, such as regression, classification, or deep learning, and optimizing hyperparameters to achieve optimal performance is time-consuming and resource-intensive.
* Handling Imbalanced Data: Dealing with imbalanced datasets where popular videos are significantly outnumbered by less popular ones can bias the model towards predicting majority classes. Employing techniques like oversampling, undersampling, or using algorithms designed for imbalanced data is necessary to mitigate this issue.
* Temporal Dynamics: Capturing the temporal dynamics of video popularity, including trends, seasonality, and sudden bursts in popularity, requires specialized modeling techniques. Incorporating time-series analysis or recurrent neural networks (RNNs) to account for temporal dependencies can enhance prediction accuracy.

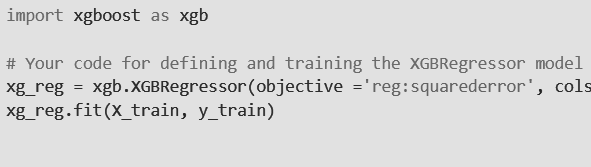
**2. FLOW DIAGRAM-**



**2.1 MODEL USED**

XGBOOST REGRESSOR

We utilized the powerful XGBoost regressor to enhance our predictive capabilities. XGBoost, known for its efficiency and accuracy, proved to be instrumental in handling the complexity of our dataset and optimizing our model's performance. By leveraging its ensemble learning techniques and gradient boosting framework. With XGBoost at the core of our project, we achieved remarkable results, providing valuable insights into factors influencing video popularity trends.



**3. HARDWARE REQUIREMENTS-**

RAM (Memory): At least 16GB of RAM is recommended for handling large datasets during training and inference.

Storage Space: Ensure sufficient storage space for storing datasets, model checkpoints, and intermediate results. 500GB or more is advisable.

CPU: A multi-core CPU (e.g., Intel Core i7 or AMD Ryzen) is essential for data loading, preprocessing, and model training.

GPU (Graphics Processing Unit): For deep learning tasks (e.g., ResNet models), a powerful GPU with at least 4GB VRAM is beneficial. NVIDIA GPUs (such as RTX 30 series) are commonly used. If using XGBoost, a GPU is not strictly necessary, but it can accelerate training.

Internet Connection: Necessary for downloading datasets, pre-trained models, and libraries. Operating System (OS): Any major OS (Windows, macOS, Linux) is suitable.

**4. SOFTWARE REQUIREMENT-**

Required libraries and setup

Python version 3

Jupyter Notebook

pandas

seaborn

numpy

**5. EXPERIMENTS** -

**A. Aim**-Changing the layers and weights of the XGBoost and studying the results and effects.

**CODE**-

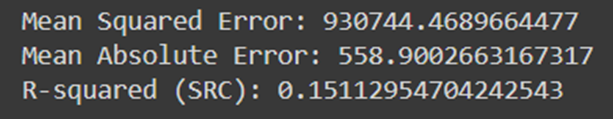
1. Changing The depth to 6 - <https://colab.research.google.com/drive/1J2_vKWgv2PHvPVRak2RCOthdeNZ5A66M>

2. Changing The weights

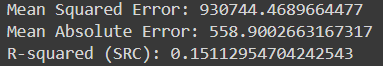
1. <https://colab.research.google.com/drive/1vJl6O-51kSy977JuaX29YWolY9VDif9i>
2. <https://colab.research.google.com/drive/1iLGTfUUhvpOyipb59eNms8F7cl-BILiq>
3. <https://colab.research.google.com/drive/1CK2fYMwAgjoTribE-tyw392SmvVnVNQI>

**OUTPUT**-

* All parameters standard



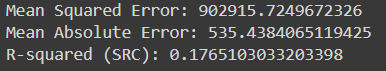
* After changing Layers





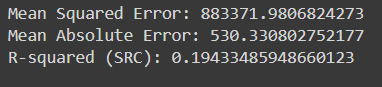
* After changing weights

minchildweight\_1:



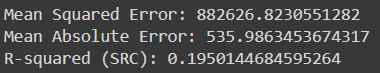


minchildweight\_3:





minchildweight\_5:





**CONCLUSION**-

* Decreasing the weights increased the accuracy whereas increasing the layers decreased the accuracy .

B. **AIM**-Changing the Learning rate of the XGBoost and studying the results and effects.

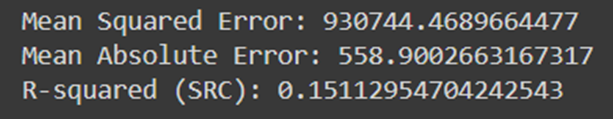
**CODE**-

**1. Changing Learning Rate -**

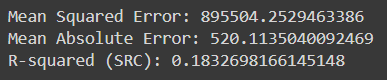
<https://colab.research.google.com/drive/1pmJD-8P13jFxlkAm5UDxdx_AXNptjMbs>

**OUTPUT**-

* **All parameters standard**



* **After changing Learning Rate-**





**CONCLUSION**-

* increasing learning rate from .1 to .3 increases accuracy from 964 to 946

C. **AIM**-Changing multiple parameters to obtain the most optimized model of the XGBoost and studying the results and effects.

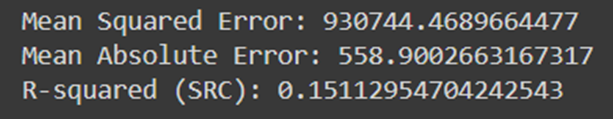
**CODE**-

1. Optimized Model -

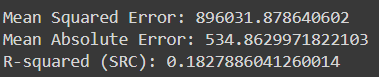
<https://colab.research.google.com/drive/1woeOEHsHFsMRHz0qBC7tlV-YyjA2Vq2C>

**OUTPUT**-

* All parameters standard



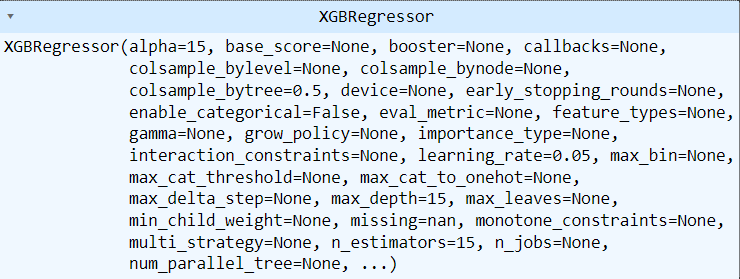
* All parameters changed-





**CONCLUSION**-

* The optimized model changes increased number of nodes per layer to 15, colsample\_bytree 0.5, learning\_rate 0.05, max depth 15, alpha 15, n\_estimators 15



**6. RESULTS-**

Linear Regression and XGBoost stand out as practical options for predicting video popularity, offering simplicity and clarity compared to more intricate models like XGBoost or Random Forest. With Linear Regression, you get a straightforward equation that shows how features like views and likes affect a video's popularity. Even though it's simple, Linear Regression gives content creators and marketers useful insights to make decisions. But it's important to remember that Linear Regression works best with linear relationships and can be thrown off by outliers. Still, overall, Linear Regression is a valuable tool for grasping and forecasting video engagement online.

**6.1 PARAMETERS USE FOR EVALUATION:-**

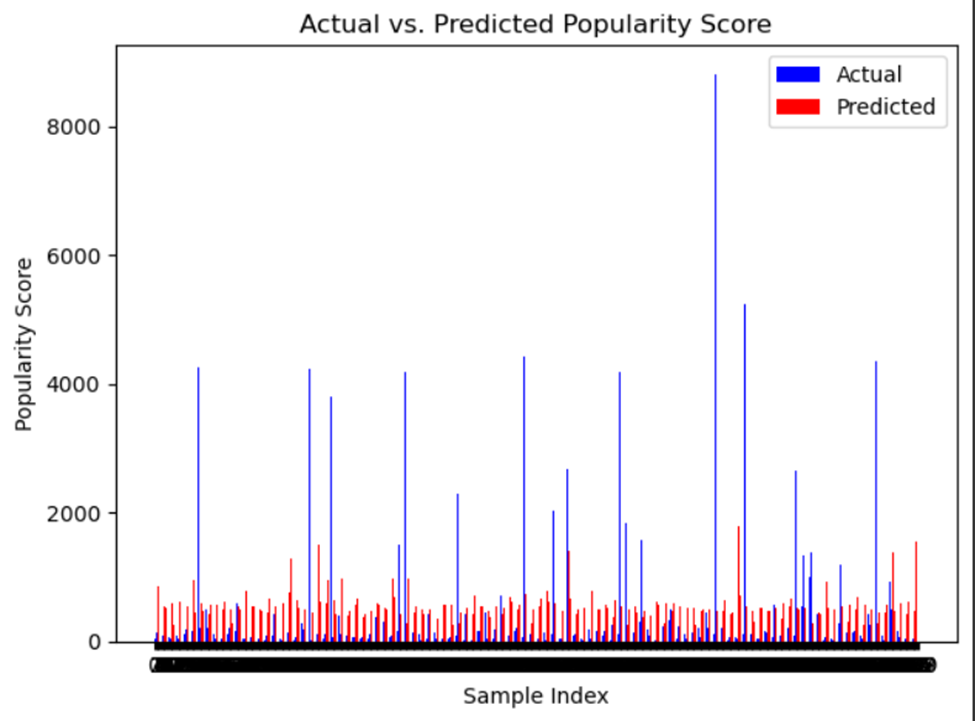
In contrast to neural networks such as Tensorflow, XGBoost operates as an ensemble learning technique rooted in decision trees, devoid of layers or weights akin to neural networks. Additionally, XGBoost offers a range of other hyperparameters that can be adjusted to enhance the model's performance:-

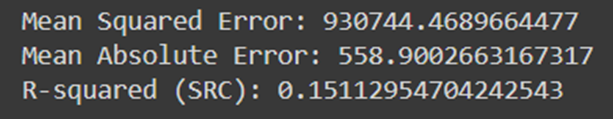
Learning rate, Maximum Depth of Trees, Number of Estimators, Column Subsampling, Alpha.

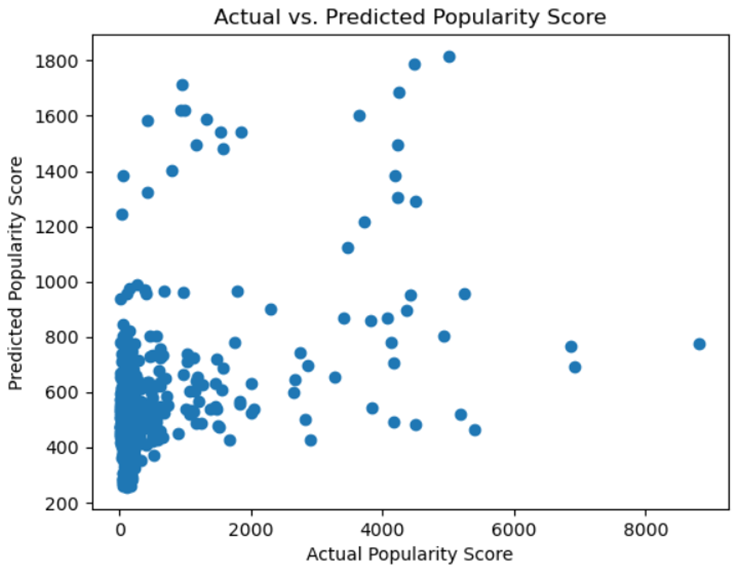
**6.2 RESULTS OF EACH PARAMETER:-**

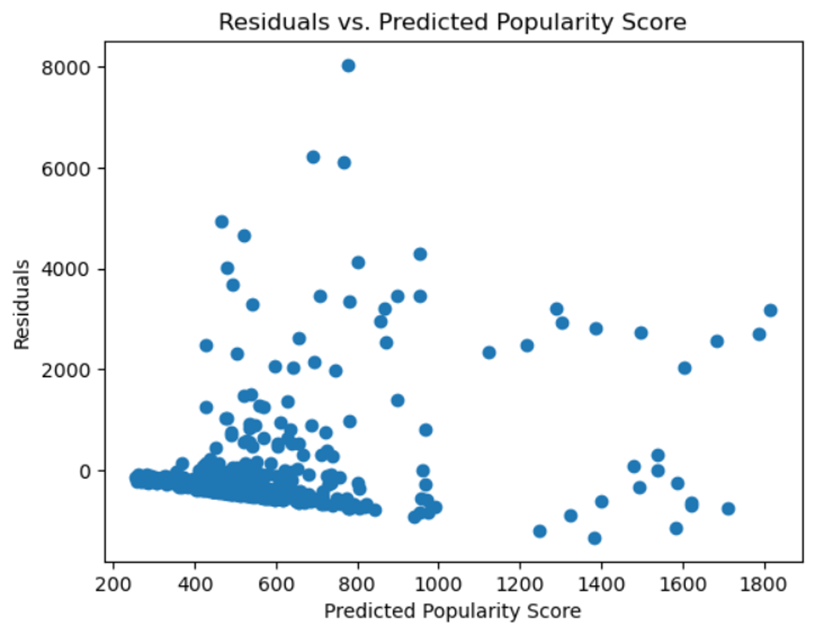
1. **Learning Rate (learning\_rate):-** Controls the step size at each iteration while moving towards a minimum of the loss function. Lower values make the model more robust but require more boosting rounds.
2. **Maximum Depth of Trees (max\_depth):-** Controls the maximum depth of the individual trees. Deeper trees can model more complex relationships but may overfit.
3. **Number of Estimators (n\_estimators):-** The number of boosting rounds or decision trees to build. More trees can lead to better performance but increase computational cost.
4. **Column Subsampling (colsample\_bytree):-** Specifies the fraction of features to consider when building each tree. Lower values can help reduce overfitting.
5. **Regularization Parameters (alpha):** L1 regularization term on weights. It helps to prevent overfitting by penalizing large weights.

**7. OUTPUT-**









**7. DATASET-**

<https://drive.google.com/drive/folders/1jT3CGME4u2EFKSniXYCN9YlB5RTaSZ34?usp=sharing>

## 8. CONCLUSION-

Through experimentation with various characteristics such as learning rate, maximum depth of trees, number of estimators, column subsampling, and regularization parameters (alpha), I gleaned valuable insights into the performance of my video popularity prediction model using XGBoost regressor.

Firstly, tinkering with the learning rate shed light on its pivotal role in determining model robustness and convergence speed. Lower rates bolstered the model's resilience but necessitated more boosting rounds, accentuating the trade-off between performance and computational overhead.

Moreover, adjusting the maximum depth of trees unearthed a delicate balance between complexity and overfitting. Deeper trees proved adept at capturing intricate relationships within the data but posed a heightened risk of overfitting, underscoring the importance of prudent depth selection.

The number of estimators emerged as a critical factor in enhancing model performance, albeit at the expense of computational resources. Incrementing the number of boosting rounds exhibited a discernible improvement in predictive accuracy, albeit with diminishing returns beyond a certain threshold.

Column subsampling surfaced as a potent tool in mitigating overfitting tendencies by constraining the feature space considered during tree construction. Opting for lower subsampling fractions facilitated a more generalized model, albeit with a potential sacrifice in predictive capacity.

Finally, regularization parameters, particularly alpha, furnished a means of curbing overfitting by penalizing excessive weight magnitudes. Fine-tuning alpha values enabled striking a delicate equilibrium between model complexity and generalization prowess.

In essence, my experimentation journey underscored the nuanced interplay between these characteristics, elucidating the multifaceted dynamics governing the efficacy of the XGBoost regressor in predicting video popularity.

**9. REFERENCES-**

* <https://ieeexplore.ieee.org/document/7450136/authors#authors>
* <https://github.com/aaryan2134/Video-Popularity-Prediction/blob/main/requirements.txt>
* <https://research.google.com/youtube8m/>