Break Through Tech Virtual Program @ Cornell Tech
December 5, 2024

Google 3E
AI Studio Final Presentation

Team Introductions

Aleksandra Jacewicz
Wesleyan University

Trina Dang
Louisiana State University

Alyssa Rodriguez
University of Central Florida

Kamillah Ismail
George mason university

Our AI Studio TA and Challenge Advisor

Helenna Yin
AI Studio TA

Sanjay Surendranath Girija
Challenge Advisor

Presentation Agenda

1. Overview
2. Goals & Business Impact
3. Actions Taken
    a. Data Understanding and Processing
    b. Modeling & Evaluation
    c. Model Comparisons
4. Reflections on What We Learned & Next Steps

Youtube Viral Video Prediction

Our Goals

1. Pre-process a real world dataset

2. Engineer features that are powerful and useable in the real world

3. (Using 1 & 2) **Build machine learning models that accurately predict how many views a viral/trending video is likely to get**
    - Random Forest
    - Linear
    - Deep Learning
    - Decision Tree

Business Impact

- Resource allocation
- User engagement
- Recommendation algorithm (& monetization)

Our Approach

August

Project and Data Understanding

- Took time to understand the problem we were interested in solving + its limitations
- Looked at dataset to anticipate what complications we would face while processing the data.

September-October

Data Pre-Processing and Feature Engineering

- Pre-processed data to make it usable for ML model (ex. Getting rid of null values and removing outliers)
- Engineered features (one hot encoding, extracting info from posted data)

November

Model Training and Evaluation

- Looked into and trained different model options, with each team member focusing on a specific type of model
- Evaluated models after training, tuning hyperparameters to optimize performance

December

Reflecting and Preparing

- Reflected on our work this semester, as well as on what we would do if we had more time
- Prepared this presentation!

Resources We Leveraged

Data Understanding & Processing

Data Examination

- Dataset gets data by taking a daily snapshot of the day's trending videos information.
- Majority of videos trend for more than one day (47k unique videos vs 268k entries in datatable).
- Realization that the columns we began with will not be usable by ML models without processing!
- Necessity for data normalization (taking log of numeric variables)

```
268787
Unique videos in set: %d 47142
Number of unique columns in data: 16
Index(['video_id', 'title', 'publishedAt', 'channelId', 'channelTitle',
       'categoryId', 'trending_date', 'tags', 'view_count', 'likes',
       'dislikes', 'comment_count', 'thumbnail_link', 'comments_disabled',
       'ratings_disabled', 'description'],
      dtype='object')
```

Data Pre-Processing & Feature Engineering

- Replacing missing data with average values (for ‘dislikes’ and ‘comments’, on a channel basis when possible)
- Combining data tables to have access to full dataset
- Removing exact duplicates in dataset (≈ 160)

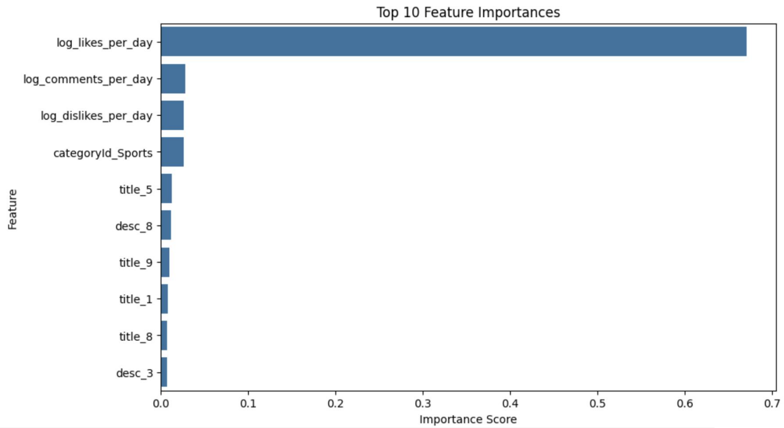
- Extracting information from time based columns (‘publishedAt’, ‘trending_date’) and taking difference between them to be able to do per day based calculations for numeric columns
- One hot encoding relevant columns (ex. categoryID -> Film & Animation, News & Politics, etc)
- Processing text based data
  - Cleaning, tokenization, building vocabulary, train embeddings and get vectorized results
  - Final output included 50 columns of numeric information for each original text based column where meaning matters (‘title’, ‘description’, ‘tags’)

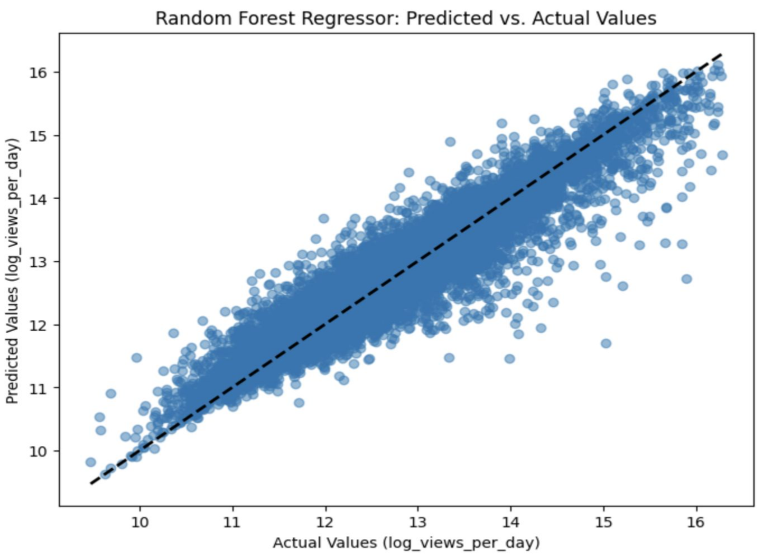
Modeling & Evaluation

Data Split & Model Selection

- We used a 90/5/5 split for training/validation/testing

- We decided to have each team member work on a different kind of model and compare what results we were able to get. We decided on the following models:
  - Random Forest
  - Linear
  - Deep Learning
  - Decision Tree

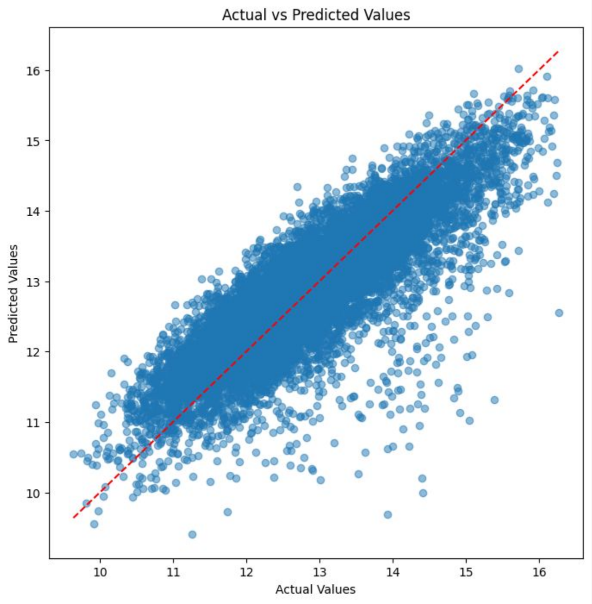




Random Forest

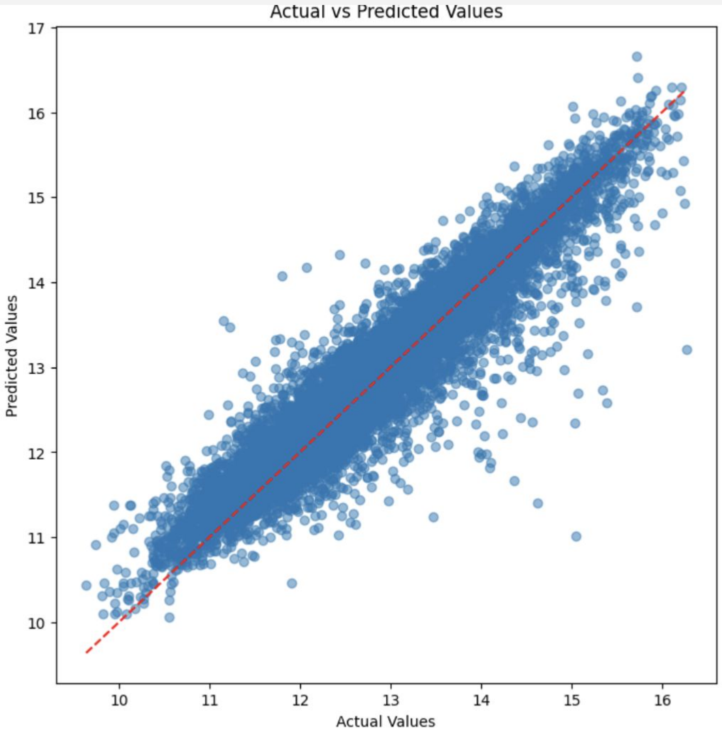
- Predicts video view counts by learning patterns from the provided features and making predictions based on the collective knowledge of multiple decision trees.
- Target Variable : The target variable in this model is the log_views_per_day, which represents the logarithm of the video's view count per day it was trending

- MSE : 0.1276 - lower is better, indicating good predictive accuracy.
- R-squared - 0.8891 - higher is better, showing the model explains most of the variability.



Linear

- Less performative than other models because of the sophisticated
- MSE of 0.33
- Strongest weights were likes and amount of days between the day published and day it trended.
- Lowest weights were columns associated with dates



Neural Network

Four layers:

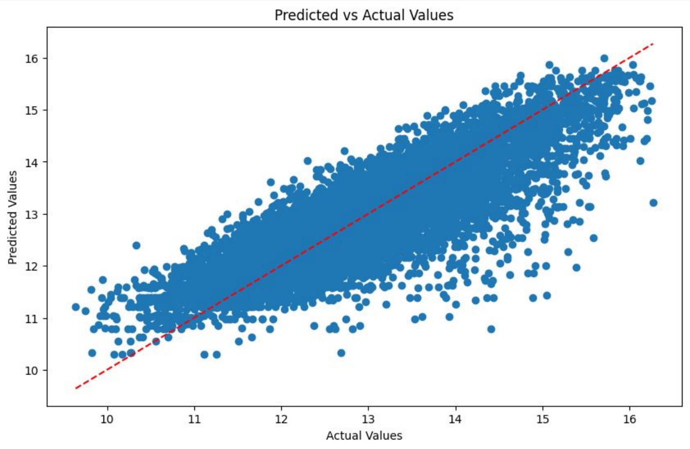
- First dense layer with 64 units and ReLU activation
- Second dense layer with 32 units ad ReLU activation
- Third dense layer with 16 units and ReLu activation
- Output dense layer with 1 unit and linear activation

Choice of scaler for scaling data: MinMax scaler versus Standard Scaler. Also decision of which parts of data are scaled (x-vales, y-values, or all)

Learning rate of .001

Has a MSE of .1069 on testing data when data is converted back to being unscaled (so that it is in correct units)

Decision Tree



Hyperparameter Tuning:

- Used GridSearchCV to find the best combination of hyperparameters for optimal model performance
    - criterion: ‘squared_error’
    - max_depth: 12
    - min_samples_leaf: 41
    - min_samples_split: 149

Model achieved a MSE 0.3107 on the validation set.

Does not perform as well due to large, complex data.

Model Name	Description	Results (MSE)	Pros	Cons
Linear Regression Model	Model which tries to find linear relationship between each feature and the y value	0.33	- Lightweight model to run - Easy to understand where it got results	- Doesn’t’ capture sophisticated nonlinear relationships
Random Forest Model	Ensemble learning model that builds multiple decision trees during training and combines their predictions using averaging	0.127	- Robust to overfitting	- Computationally expensive

Model Name	Description	Results (MSE)	Pros	Cons
Neural Network	Model which learns through data propagated using layers of nodes (loosely based on human brain)	.1069	Capable of capturing complexity of data	- Costly computation - Poor human interpretability
Decision Tree	Model which recursively splits data into subsets based on feature values to form decisions	0.3107	- Easy to interpret and visualize	- Prone to overfitting

Final Thoughts

Insights and Key Findings

- Importance of soft skills while working on technical challenges (team collaboration, planning & hosting meetings, time management, etc)
- Importance of understanding the problem you're trying to solve
- Importance of data pre-processing in the ML building process (& many rounds of it that are necessary to get to model building stage)
- Importance of picking a label ('log_views_per_day')

- Neural network performed the best as it could capture nonlinear relationships the best

What We Learned

- Practical experience pre-processing real world data and the multitude of things that have to be done to it before it can be used in a ML model
- How to get around Google Colab's RAM related issues using batching and caches
- Using unfamiliar Python libraries such as Gensim and Tensorflow

Next Steps

- Try an XGBoost model
- Test current models on data from other countries and evaluate performance
- Try out time based split instead of random split of testing/training/validation data and evaluate performance
- Make models more applicable to real world by getting rid of columns related to number of likes/number of comments (data leakage) and evaluate performance with that accounted for
- Find way to derive more features while avoiding data leakage
- Test more variations of parameter values on the models we worked on

Questions?

Thank you for listening to our presentation!