Machine learning-CSCE 5215 Section 005

Prediction of trending status for a YouTube video using machine learning method

participants:

Karthik Undi-11666360
Surya Vardhan Kadiyala-11709167
Vinay Kusuma-11706797

Abstract

- In the era of digital media, YouTube has become a vital resource for both users and content producers. Viewers, advertisers, and content producers can all benefit from an understanding of the mechanisms behind YouTube trending patterns and video popularity. In order to meet this demand, our project will create a predictive model that can predict if a YouTube video will become popular within a week of its release.

- The project makes use of a dataset from the "USvideos.csv" file, which contains information on YouTube videos including views, likes, dislikes, the number of comments, the publish date, and the time. The main goal is to forecast the "day-to-trend," a binary classification issue that asks whether a video will become popular a week after it is released. These predictions are generated using a powerful machine learning model called the Random Forest Classifier. To provide a thorough evaluation of the model's predictive power, multiple metrics are used to analyse the model's performance, including the F1 score, Mean Accuracy, and Out of Bag score.

This project concludes by outlining a methodical strategy for identifying and forecasting YouTube video trends. The project offers a platform that helps different stakeholders make educated decisions by utilising machine learning techniques and a rich dataset. In the ever-changing world of digital media, the insights produced by this project can be used by content creators wishing to attract a wider audience, marketers hoping to optimize their campaigns, or consumers hoping to keep up with the latest trends. The initiative highlights how data-driven methods have the power to revolutionize how we perceive online video-sharing sites like YouTube.

ML Problem specification:

The dataset used in this project is ‘USvideos.csv’, which contains data about YouTube videos.

The data includes features like views, likes, dislikes, comment count, publish date, and time.

The target variable is ‘day-to-trend’, which represents the number of days a video takes to get on the trending list. This is a binary classification problem where the aim is to predict whether a video will trend within a week of being published.

The model used for this task is the Random Forest Classifier. The performance of the model is evaluated using metrics like Out of Bag score, Mean Accuracy, and F1 score.

Grid search is used to find the best parameters for the model. The features used for training the model are ‘views’, ‘likes’, ‘dislikes’, ‘publish_wd’, and ‘publish_hr’.

The model is trained on a subset of the data where comments are disabled to reduce training time. The model’s feature importance's are also calculated to understand the contribution of each feature to the prediction. The results are then analysed and interpreted.

Data Specification:

The dataset used in this project is ‘USvideos.csv’, which contains data about YouTube videos. The dataset includes features like views, likes, dislikes, comment count, publish date, and time. The target variable is ‘day-to-trend’, which represents the number of days a video takes to get on the trending list. The features used for training the model are ‘views’, ‘likes’, ‘dislikes’, ‘publish_wd’, and ‘publish_hr’.

The **type of samples in the dataset are YouTube videos** and the number of samples is the length of the data. The **length of the data is 40949**.

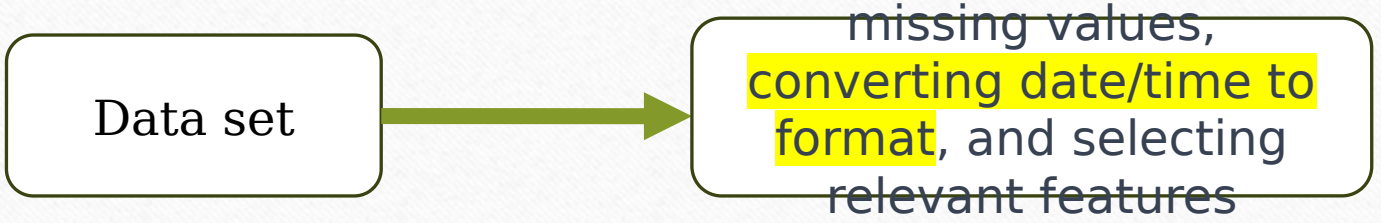
Design and milestones

1. **Data Import and Inspection:** The data is imported from the ‘USvideos.csv’ file using the pandas library in Python. The structure of the data is inspected using methods like head(), info(), and Len() to understand the dataset’s size and the types of variables it contains.

```
#dataset size
len(data)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40949 entries, 0 to 40948
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   video_id                40949 non-null  object
 1   trending_date           40949 non-null  object
 2   title                   40949 non-null  object
 3   channel_title           40949 non-null  object
 4   category_id             40949 non-null  int64
 5   publish_time            40949 non-null  object
 6   tags                    40949 non-null  object
 7   views                   40949 non-null  int64
 8   likes                   40949 non-null  int64
 9   dislikes                40949 non-null  int64
 10  comment_count           40949 non-null  int64
 11  thumbnail_link          40949 non-null  object
 12  comments_disabled       40949 non-null  bool
 13  ratings_disabled        40949 non-null  bool
 14  video_error_or_removed  40949 non-null  bool
 15  description             40379 non-null  object
dtypes: bool(3), int64(5), object(8)
memory usage: 4.2+ MB
```

2. Data Preprocessing: The date and time columns in the dataset are converted to the appropriate format using pandas. New columns for publish date and time are added. Unnecessary columns are dropped, and duplicates are removed from the dataset.

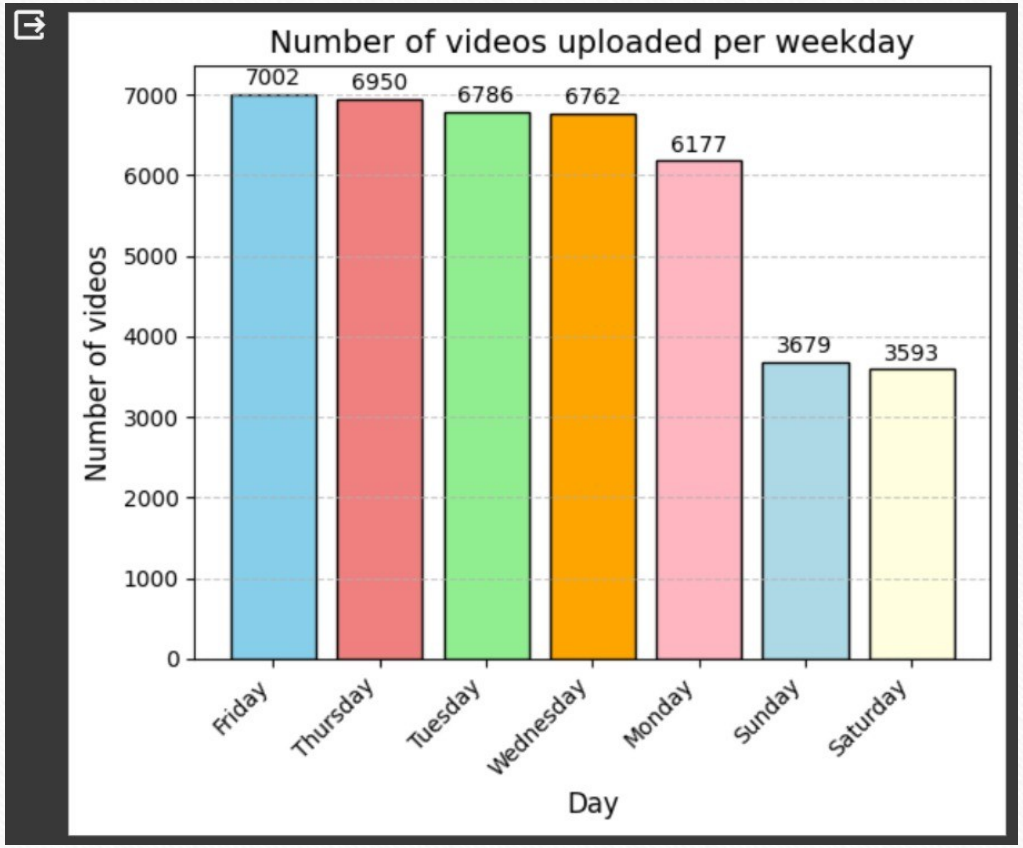


This is the output after preprocessing the data by removing comments, changing date time format

	video_id	trending_date	title	channel_title	category_id	publish_time	views	likes	dislikes	comment_count	thumbnail_link	comments_disabled	ratings_disabled	publish_date	publish_wd	publish_hr
0	2kyS6SvSYSE	2017-11-14	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat	22	17:13:01	748374	57527	2966	15954	https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg	False	False	2017-11-13	0	17
1	1ZAPwfrtAFY	2017-11-14	The Trump Presidency: Last Week Tonight with J...	LastWeekTonight	24	07:30:00	2418783	97185	6146	12703	https://i.ytimg.com/vi/1ZAPwfrtAFY/default.jpg	False	False	2017-11-13	0	7
2	5qpjK5DgCt4	2017-11-14	Racist Superman | Rudy Mancuso, King Bach & Le...	Rudy Mancuso	23	19:05:24	3191434	146033	5339	8181	https://i.ytimg.com/vi/5qpjK5DgCt4/default.jpg	False	False	2017-11-12	6	19
3	puqaWrEC7tY	2017-11-14	Nickelback Lyrics: Real or Fake?	Good Mythical Morning	24	11:00:04	343168	10172	666	2146	https://i.ytimg.com/vi/puqaWrEC7tY/default.jpg	False	False	2017-11-13	0	11
4	d380meD0W0M	2017-11-14	I Dare You: GOING BALD!?	nigahiga	24	18:01:41	2095731	132235	1989	17518	https://i.ytimg.com/vi/d380meD0W0M/default.jpg	False	False	2017-11-12	6	18

3. Exploratory Data Analysis The number of videos uploaded per weekday is visualized using a bar chart created with matplotlib. This provides an initial understanding of the data distribution.

- This visualization is done based on the publish day uing a bar plot we have observed that most videos are uploaded in YouTube on Friday.



4. Feature Engineering: A new column for the number of days a video takes to get on the trending list is created. The dataset is then filtered to only include videos with disabled comments to reduce training time. This process is to filer the features based on the model selection by eliminating comments

This is the output screen shot after feature engineering

```
[ ]  #number of days it takes for a video to get to trending page on youtube

     #considering only videos with disabled comments to reduce training time and also because comment count is currently not useful
     new_data = data.loc[(data.comments_disabled) &  (~data.ratings_disabled)].copy()

     #Create a new column for the number of days a video takes to get on the trending list
     new_data['day_to_trend'] = abs(np.subtract(new_data.trending_date.dt.date,new_data.publish_date).apply(lambda x: x.days))
     left_vars = ['views','likes','dislikes','comment_count','publish_wd','publish_hr','day_to_trend','title']

     new_data = new_data[left_vars]
     new_data.reset_index(inplace=True)
     new_data.head()
```

	index	views	likes	dislikes	comment_count	publish_wd	publish_hr	day_to_trend	title
0	31	26000	119	69	0	0	17	8	Amazon Christmas Advert 2017 - Toys & Games
1	103	264793	3283	853	0	3	8	5	H&M Holiday 2017 starring Nicki Minaj – offici...
2	290	94229	217	177	0	0	17	9	Amazon Christmas Advert 2017 - Toys & Games
3	372	271685	3330	854	0	3	8	6	H&M Holiday 2017 starring Nicki Minaj – offici...
4	483	11769	127	13	0	1	17	2	Amazon CEO Jeff Bezos and brother Mark give a ...

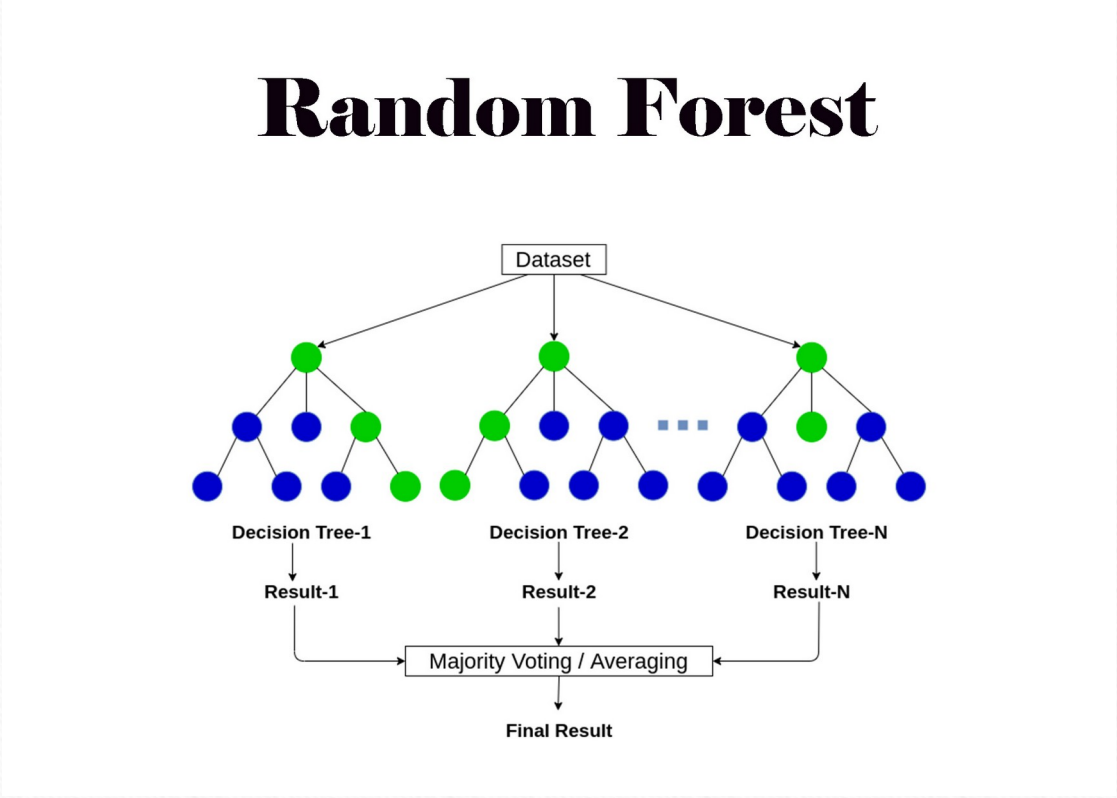
4. Model selection: we have considered various classification models such as gradient boosting and random forest for this project.
We have decided to use these model because this are a ensemble methods which are responsible for better accuracy.

gradient boosting: Gradient Boosting is an ensemble learning method that sequentially builds a series of weak models, typically decision trees, and combines their predictions to form a strong predictive model. It belongs to the family of boosting algorithms, where each subseq evious ones.

```
Gradient Boosting Classifier:
Accuracy: 0.8207547169811321

Confusion Matrix:
[[18 12]
 [ 7 69]]

Classification Report:
              precision    recall  f1-score   support

       False       0.72      0.60      0.65        30
        True       0.85      0.91      0.88        76

    accuracy                           0.82       106
   macro avg       0.79      0.75      0.77       106
weighted avg       0.81      0.82      0.82       106
```

Random forest classifier: Random Forest is an ensemble learning technique that operates by constructing multiple decision trees during training and outputs the aggregated result of these trees. It belongs to the bagging family of ensemble methods, which aims to reduce overfitting and improve the model's generalization.

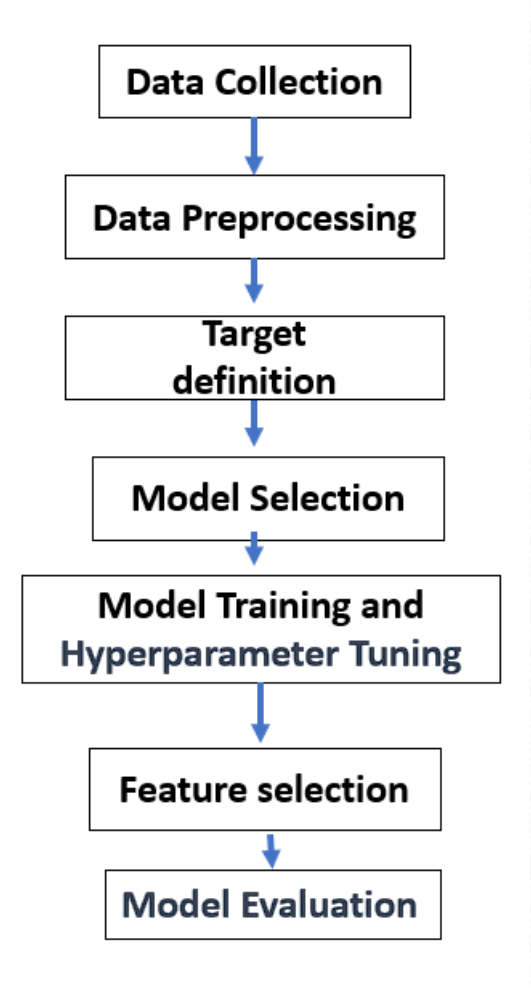
We have implemented both the classifier and observed the accuracy then finally we decided to use random forest classifier



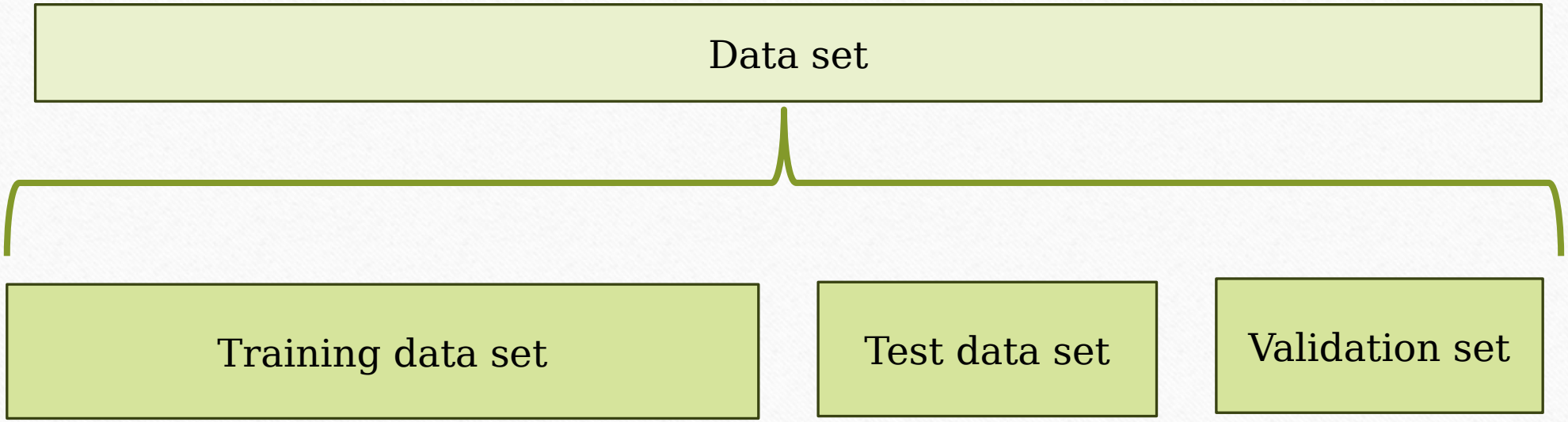
Ensemble methods:

- In this project we use gradient boosting and random forest classification because these methods are useful for better accuracy and we are not separately implementing ensemble methods but the classifiers we use is it self a ensemble method.

- We have choosed Random Forest algorithm which is well suited for non-Gaussian/non-normalized data due to its ensemble nature.it builds numerous decision trees that independently train on random subsets of data and features the model is able to accurately represent intricate, nonlinear interactions. Because taking the average of numerous trees lessens the influence of outliers Random Forest is adaptable to different types of data, managing both linear and non-linear correlations which our data contains. It is more successful in situations where Gaussian assumptions would not hold true because of its decreased overfitting and flexibility to various patterns.


Data Collection
Data Preprocessing
Target definition
Model Selection
Model Training and Hyperparameter Tuning
Feature selection
Model Evaluation

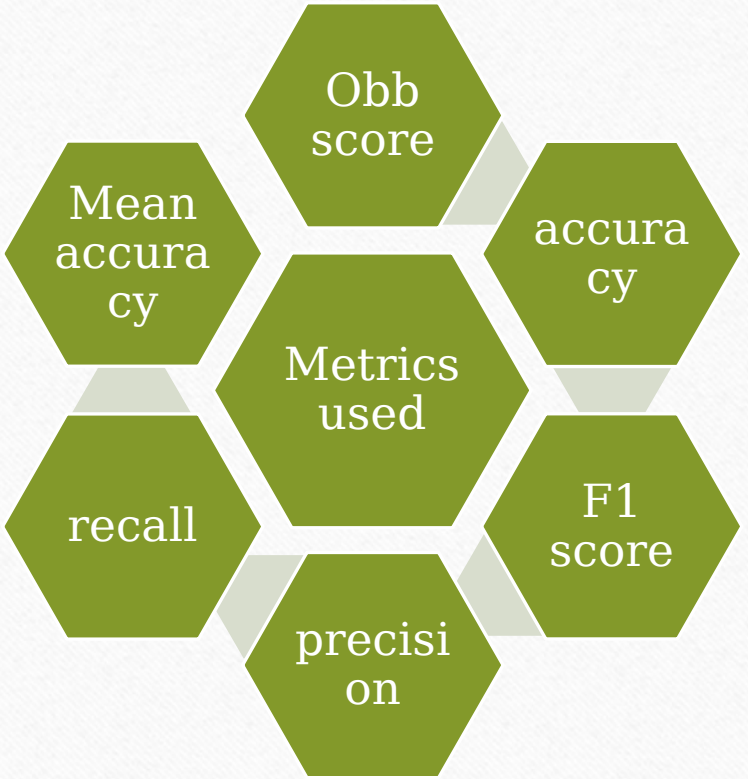
5.Model training and evaluation:



Hyper parameter tuning:

- After training the model , we have done grid search which is helpful for hyper tuning the parameters.
- After performing the grid search we obtained maximum depth and n_estimators and also give best score .

The Random Forest Classifier model from the sklearn library is used for prediction. The features used for training the model are ‘views’, ‘likes’, ‘dislikes’, ‘publish_wd’, and ‘publish_hr’. The data is split into training and testing sets using the train_test_split method from sklearn. Grid search is performed to find the best parameters for the model. The model’s performance is evaluated using metrics like Out of Bag score, Mean Accuracy, and F1 score. The feature importances are also calculated to understand the contribution of each feature to the prediction.



References and related projects:

https://www.kaggle.com/code/westoon/exercise-summary-functions-and-maps

https://github.com/ashutoshkrris/YouTube-Trending-Videos-Analysis

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://scikit-learn.org/stable/modules/grid_search.html

https://www.kdnuggets.com/2022/10/hyperparameter-tuning-grid-search-random-search-python.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

Thank you