Data Collection and Preprocessing Phase

Date	15 July 2024
Team ID	Team-740680
Project Title	View count visionary:data driven approach to forecasting youtube videos views project
Maximum Marks	6 Marks

Preprocessing Template

The preprocessing template for "View Count Visionary: A Data-Driven Approach to Forecasting YouTube Video Views" outlines a systematic approach to preparing data for predictive modeling.Standardizing or normalizing numerical features, encoding categorical variables, and splitting data into training and testing sets to prepare for model training.

Section	Description
Data Overview	Assess the dataset containing YouTube video metadata and statistics. This includes variables such as video ID, title, upload date, view count, likes, dislikes, and comments.
Resizing	Resize any thumbnail images associated with the YouTube videos to a standard size suitable for analysis. This ensures uniformity in image dimensions and facilitates efficient processing.
Normalization	Normalize numerical features such as view count, likes, dislikes, and comments to a consistent scale, such as [0, 1]. This standardization helps in reducing the impact of varying scales on predictive models.
Data Augmentation	Augment the dataset by extracting additional features that could influence video views, such as video duration, upload time (hour of the day, day of the week), and categorical features like video category or uploader statistics. This expands the dataset to capture diverse factors affecting view counts.
Denoising	Apply denoising techniques to handle outliers or anomalies in the data, ensuring that extreme values or errors do not disproportionately influence forecasting models. Techniques may include statistical methods or domain-specific filters.
Edge Detection	In the context of YouTube video analysis, edge detection may not directly apply. However, analogous techniques could involve identifying sudden spikes or drops in view counts over time, which may indicate viral trends or content saturation.
Color Space Conversion	While color space conversion is specific to image processing and may not directly apply to YouTube video data, a related concept could involve sentiment analysis or categorization based on video

	content themes (e.g., educational,entertainment).
Image Cropping	select relevant segments of the dataset for focused analysis, such as videos within specific categories or those uploaded by influential creators. This targeted approach helps in understanding trends within particular subsets of YouTube content
Batch Normalization	Implement batch normalization techniques when training machine learning models to forecast view counts. This ensures stable model training by normalizing activations and accelerating convergence during iterative processes.
Data Preprocessing Code Screenshots	

Loading Data	This involves reading data into your program, commonly from files or databases.
	#generating birds eye view
	data.info()
	<class 'pandas.core.frame.DataFrame'>
	RangeIndex: 14999 entries, 0 to 14998
	Data columns (total 9 columns):
	 # Column Non-Null Count Dtype
	--- ------ -------------- -----
	 0 vidid 14999 non-null object
	 1 adview 14999 non-null int64
	 2 views 14999 non-null object
	 3 likes 14999 non-null object
	 4 dislikes 14999 non-null object
	 5 comment 14999 non-null object
	 6 published 14999 non-null object
	 7 duration 14999 non-null object
	 8 category 14999 non-null object
	dtypes: int64(1), object(8)
	memory usage: 1.0+ MB

Resizing	Adjusting the dimensions of images or data points to a specified size, which is often necessary for standardization in machine learning tasks.
	#to disply the no.of missing values
	data.isna().sum()
	vidid 0
	adview 0
	views 0
	likes 0
	dislikes 0
	comment 0
	published 0

	duration 0 category 0 dtype: int64
Normalization	Scaling data to a standardized range, typically between 0 and 1 or -1 and 1, to ensure that different features contribute equally to the analysis. data.describe() adview count 1.499900e+04 mean 2.107791e+03 std 5.237711e+04 min 1.000000e+00 25% 1.000000e+00 50% 2.000000e+00 75% 6.000000e+00 max 5.429665e+06 data.fillna(0,inplace=True)
Data Augmentation	Techniques used to artificially increase the diversity of your training dataset by applying transformations such as rotation, flipping, or cropping to existing data. data.dropna() vidid adview views likes dislikes comment published duration category 0 VID_18655 40 1031602 8523 363 1095 2016- 09-14 PT7M37S F 1 VID_14135 2 1707 56 2 6 2016-10-01 PT9M30S D 2 VID_2187 1 2023 25 0 2 2016-07-02 PT2M16S C 3 VID_23096 6 620860 777 161 153 2016- 07-27 PT4M22S H 4 VID_10175 1 666 1 0 0 2016-06-29 PT31S D ... ... 14994 VID_31 2 525949 1137 83 86 2015- 05-18 PT6M10S A 14995 VID_5861 1 665673 3849 156 569 2015- 10-20 PT3M56S D 14996 VID_805 4 3479 16 1 1 2013-08-23 PT3M13S B 14997 VID_19843 1 963 0 0 0 2010-10-02 PT26S G 14998 VID_8534 1 15212 22 5 4 2016-02-19 PT1M1S D 14999 rows × 9 columns

Denoising	Removing noise from data, which is especially common in image processing tasks to improve the quality of images. data.isnull().sum() vidid 0 adview 0 views 0 likes 0 dislikes 0 comment 0 published 0 duration 0 category 0 dtype: int64 [ ]
Edge Detection	Identifying and highlighting boundaries within an image, which is crucial for tasks like object detection and segmentation. data.info() <class 'pandas.core.frame.DataFrame'> RangeIndex: 14999 entries, 0 to 14998 Data columns (total 9 columns): # Column Non-Null Count Dtype --- ------ -------------- ----- 0 vidid 14999 non-null object 1 adview 14999 non-null int64 2 views 14999 non-null object 3 likes 14999 non-null object 4 dislikes 14999 non-null object 5 comment 14999 non-null object 6 published 14999 non-null object 7 duration 14999 non-null object 8 category 14999 non-null object dtypes: int64(1), object(8) memory usage: 1.0+ MB

Color Space Conversion	Changing the representation of colors in an image from one color space to another (e.g., RGB to HSV), which can help in certain types of image analysis.
	import pandas as pd
	
	# Load the dataset
	file_path = '/content/train.csv'
	df = pd.read_csv(file_path)
	
	# Remove all rows with NaN values
	df_cleaned = df.dropna()
	
	# Save the cleaned dataset
	df_cleaned.to_csv('/content/train.csv', index=False)
	
	print("NaN values removed and cleaned dataset saved.")