BAN 612 Presentation Plan

YouTube ENGAGEMENT ANALYSIS

MrBeast — 250 Uploads

Ali Raza XY7887

Why Some Videos Go Viral

The challenge: predicting video engagement before upload.

Millions of uploads daily → limited viewer attention.

Goal: Use metadata (title, duration, timing) to forecast engagement.

Case Study: @MrBeast (200M+ subscribers).

YouTube creators constantly ask why some videos explode while others flop. Engagement drives revenue and reach, but creators rarely get actionable insights.

This project focuses on whether we can predict engagement from metadata things a creator controls before upload.

Objectives & Workflow

Objectives

01

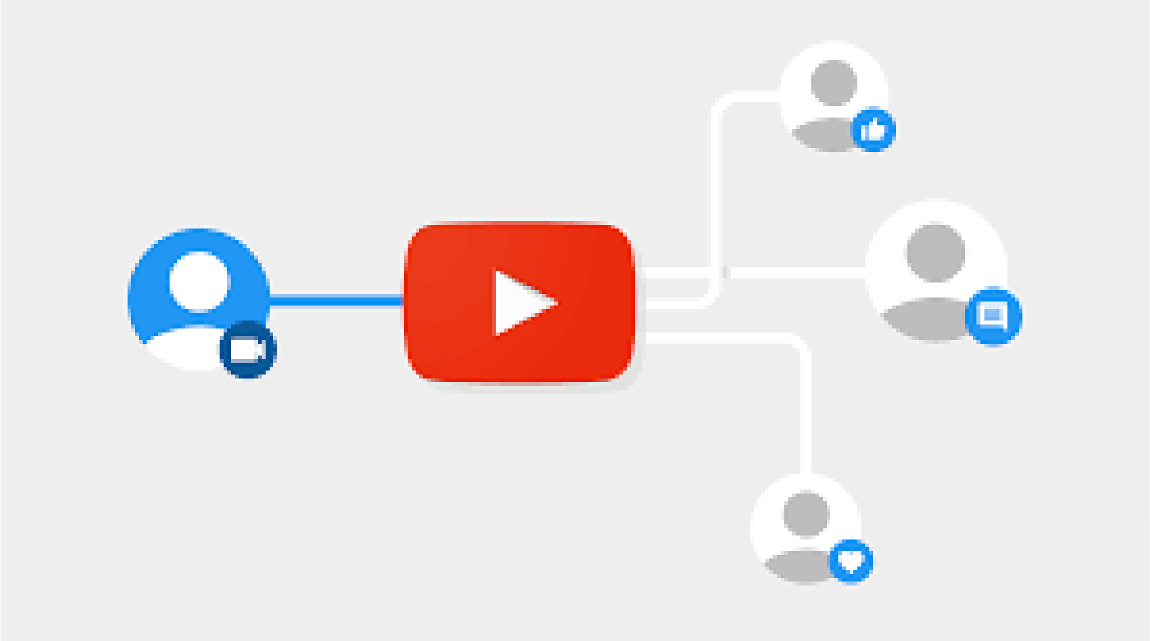
Quantify how metadata affects engagement.
Build predictive models (views, likes, comments).
Explain predictions using SHAP (feature importance).

Workflow

02

API → Cleaning → Feature Engineering → EDA → Modeling → SHAP Interpretation

fetch_youtube.py → fetch_video_stats(video_ids)



```python
BASE = "https://www.googleapis.com/youtube/v3"
def fetch_video_stats(video_ids):
    params = {
        "part": "snippet,contentDetails,statistics",
        "id": ",".join(video_ids),
        "key": os.getenv("YOUTUBE_API_KEY"),
        "maxResults": 50
    }
    r = requests.get(f"{BASE}/videos", params=params, timeout=30)
    r.raise_for_status()
```

Data Collection & Feature Engineering

Source: YouTube Data API v3

Sample: 250 videos from @MrBeast

Variables:

PublishedAt, Duration, Title, Views, Likes, Comments

Transformations:

ISO-8601 Duration → Seconds

PublishedAt → Hour, Day of Week

Added peak_hour flag (18–22 h)

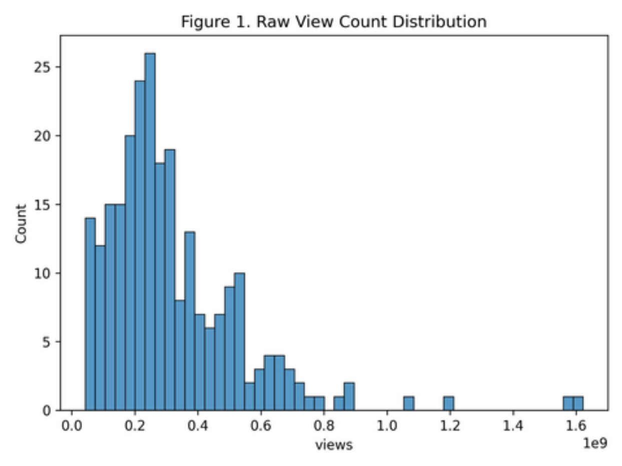
Log(views) transformation

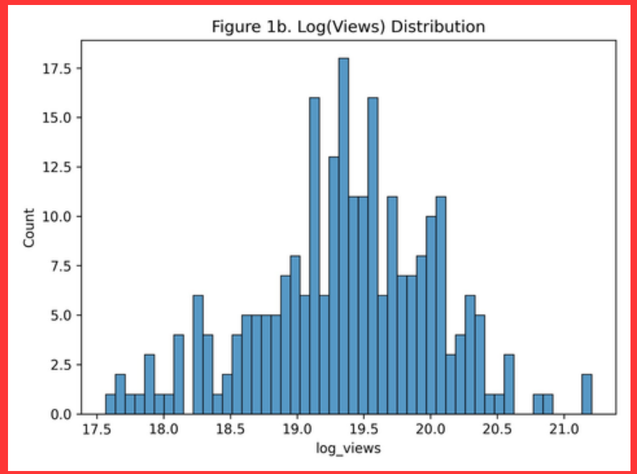
```
# features.py
def rows_to_df(items):
    df = pd.json_normalize(items)
    df["published_at"] = pd.to_datetime(df["snippet.publishedAt"])
    df["publish_hour"] = df["published_at"].dt.hour
    df["duration_seconds"] = df["contentDetails.duration"].apply(parse_iso8601_to_seconds
    df["title_len"] = df["snippet.title"].str.len()
    df["views"] = df["statistics.viewCount"].astype(int)
    df["log_views"] = np.log1p(df["views"])
    return df[["video_id","title","publish_hour","duration_seconds","title_len","views","
```

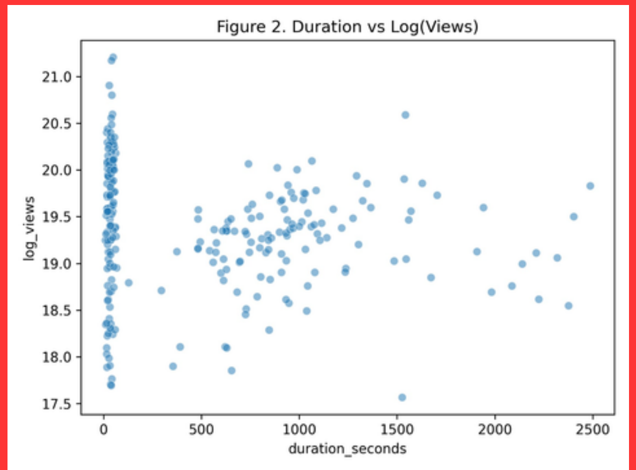
Exploratory Data Analysis

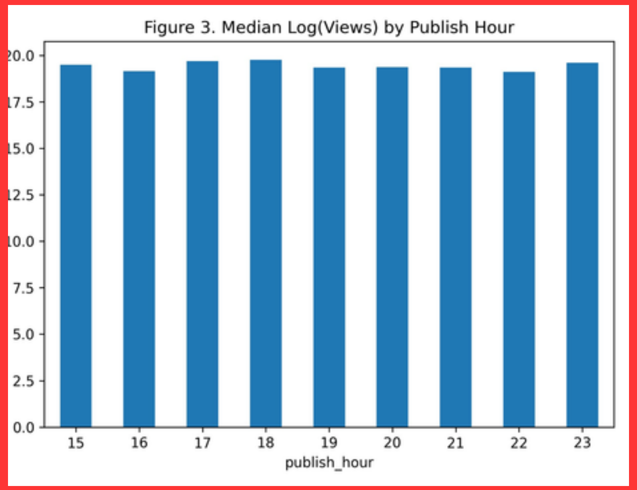
Key Findings:

- Heavy-tailed views → log transform used.
- Duration vs. log(views): non-linear trend.
- Publish Hour: evening (18–22h) videos perform best.
- Title Length: small positive correlation.









Modeling Approach

Baselines

01

- Linear Regression (log views)
- Logistic Regression (high/low engagement)

Advanced Models

02

- Random Forest
- Gradient Boosting

Validation

03

- Adaptive 5-Fold Cross-Validation
- Metrics: R^2 (Regression), ROC-AUC (Classification)

```python
# models.py (skeleton)
X = df[["publish_hour","duration_seconds","title_len"]]
y_reg = df["log_views"]
y_cls = (df["views"] >= df["views"].median()).astype(int)

pre = ColumnTransformer(
    [("num", StandardScaler(), ["publish_hour","duration_seconds","title_len"])]
)
lin_reg = Pipeline([("pre", pre), ("m", LinearRegression())])
log_clf = Pipeline([("pre", pre), ("m", LogisticRegression(max_iter=1000))])

kfold = KFold(n_splits=5, shuffle=True, random_state=42)
r2 = cross_val_score(lin_reg, X, y_reg, cv=kfold, scoring="r2").mean()
auc = cross_val_score(log_clf, X, y_cls, cv=kfold, scoring="roc_auc").mean()
```

Subscribed

Evaluation & Results

Even though the numerical gains are modest, the tree models outperform the linear baselines.

Their real value is interpretability, understanding why engagement changes.

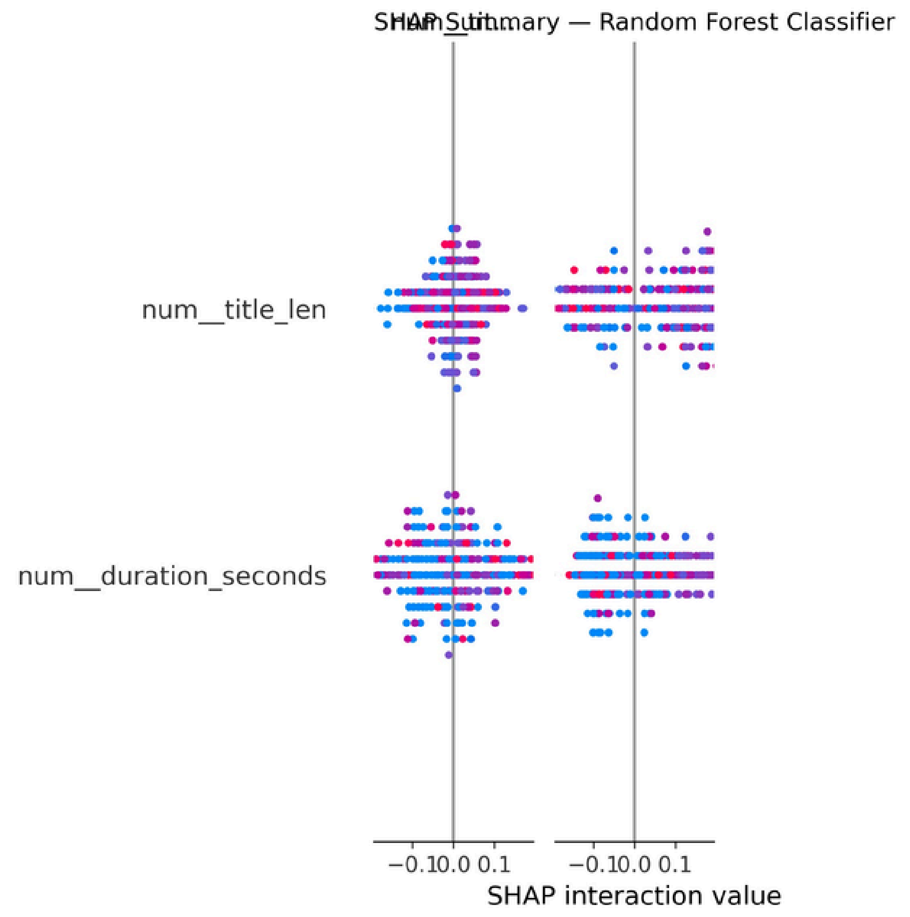
Model	Task	Metric	Score
Linear Regression	log views	R^2	−0.013
Logistic Regression	High/Low	ROC-AUC	0.634
Random Forest	log views	R^2	0.0135
Random Forest	High/Low	ROC-AUC	0.653
Gradient Boosting	log views	R^2	−0.0856
Gradient Boosting	High/Low	ROC-AUC	0.6646

Tree-based models slightly improve performance and are fully interpretable via SHAP.

Global SHAP Feature Importance

Top Global Drivers:

1. **Duration** (seconds)
2. **Is Short** (≤60s flag)
3. **Publish Hour**
4. **Peak Hour** (18–22)
5. **Title Length**



Local Explanation – Example Video (#10)

Prediction: High Engagement

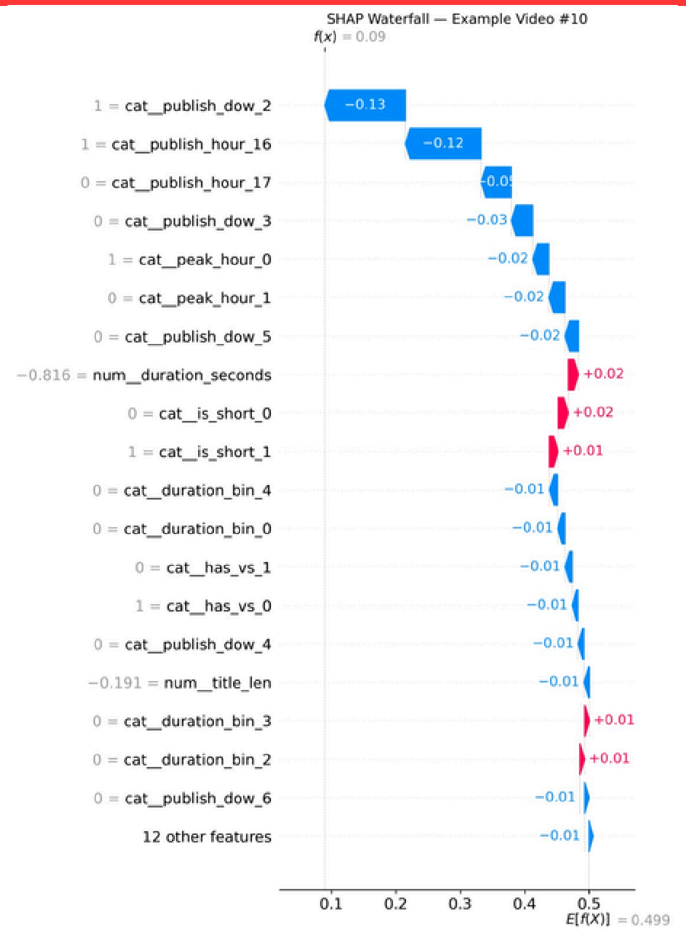
Key Drivers:

Duration = 45s (Short)

Published at 19:00 (Peak Hour)

Balanced title length

Net positive SHAP contribution from timing + duration.



Insights & Recommendations

Key Takeaways

01 Timing

Upload between 18–22h local time.

02 Content

Separate strategy for Shorts vs. Long-form.

03 Keywords

Keep titles concise and keyword-rich.

THANK YOU!

I hope you learned something new!