INTERNSHIP PROJECT REPORT

ON

“YOUTUBE CHANNEL PERFORMANCE DATA ANALYSIS”

SUBMITTED BY:

PRAYAS SAMAL

EXECUTIVE SUMMARY

This analysis uses historical data from 2016 to 2024 to pinpoint the YouTube channel's main growth, content efficiency, and monetization strategies.

Key Findings

1. Performance Drivers and Historical Context: It is confirmed that the two most important and highly associated success measures are Views and Watch Time (ρ = 0.93$), which are also the best indicators of subscriber growth ($\rho \geq 0.73$). Examining past data reveals a huge performance peak in total views in 2017, suggesting a substantial change in audience engagement, platform patterns, or content strategy that requires more research to guide current content creation.

2. Monetization Efficiency of Longer Content: The better efficiency of longer videos is the most important discovery for revenue strategy. The highest Revenue per View (RPV) is produced by videos categorized as "Long" (10–15 min) and "Very Long" (> 15 min), which can be up to five times higher than the RPV of "Short" (< 5 min) videos. This implies that the volume produced by shorter clips is greatly outweighed by the ad load or ad value associated with longer material.

3. The CTR-Revenue Paradox: Surprisingly, the Second CTR Quintile (Q2) yields the highest Average Estimated Revenue, despite the Highest CTR Quintile (Q5) producing the most average views. This raises the possibility of a CTR-Revenue Paradox: highly optimized or provocative thumbnails or titles may draw clicks (high CTR) but fail to hold on to an audience that is valuable to advertisers. This shows that thumbnail approach should be in line with quality viewer intent rather than merely click volume.

4. Duration vs. Engagement Trade-off: All video duration categories continue to have high engagement metrics, with the Like Rate continuously exceeding 93%. But there is an obvious trade-off when it comes to consumption: For videos under five minutes, the average view percentage is 62.53%; for those above fifteen minutes, it is 36.27%. The strategy needs to strike a compromise between the lower audience retention rate and the higher revenue efficiency of lengthier content.

Important Suggestions

We suggest the following strategic pivots in light of these findings:

Prioritize Longer Content: To take advantage of higher RPV while upholding current high production standards, reallocate content production resources to films in the 10–15 minute time range.

Optimize Q2 CTR Strategy: To reproduce the Q2 videos' strong monetization-focused engagement throughout the whole content library, examine the unique thumbnail and title features.

Examine 2017 Performance: To comprehend the enormous increase in view volume and ascertain whether any components are replicable in the present platform environment, thoroughly examine the content and publishing tactics employed in 2017.

TABLE OF CONTENTS

OBJECTIVES

This project's main goals were to provide a thorough study of the channel's video performance data in order to produce data-driven insights for strategic decision-making. In particular, the analysis sought to:

Create a Key Performance Indicator (KPI). Relationships: To identify the most powerful forces behind channel expansion, quantify the relationships between primary engagement and growth KPIs (Views, Watch Time, Subscribers).

Determine Revenue Efficiency by Duration: Compute and compare the Revenue per View (RPV) across various duration cohorts to examine how video length (Duration Group) affects monetization.

Evaluate Front-End Optimization Effectiveness: To find high-converting and high-value content patterns, examine the correlation between pre-click metrics (such as Video Thumbnail CTR) and post-click results (Views and Estimated Revenue).

Determine Historical Performance Anomalies: To find past effective tactics that might be used now, identify and contextualize notable changes or spikes in historical performance (such as Total Views by Year).

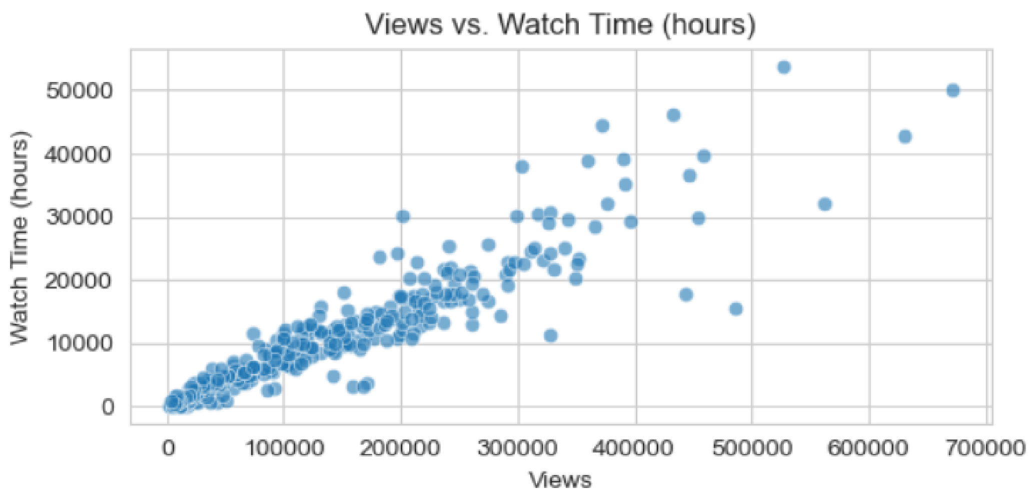
Create Actionable Recommendations: Convert all analytical results into specific, top-priority plans for maximizing next efforts at content production, targeting, and monetization.

DATA ANALYSIS

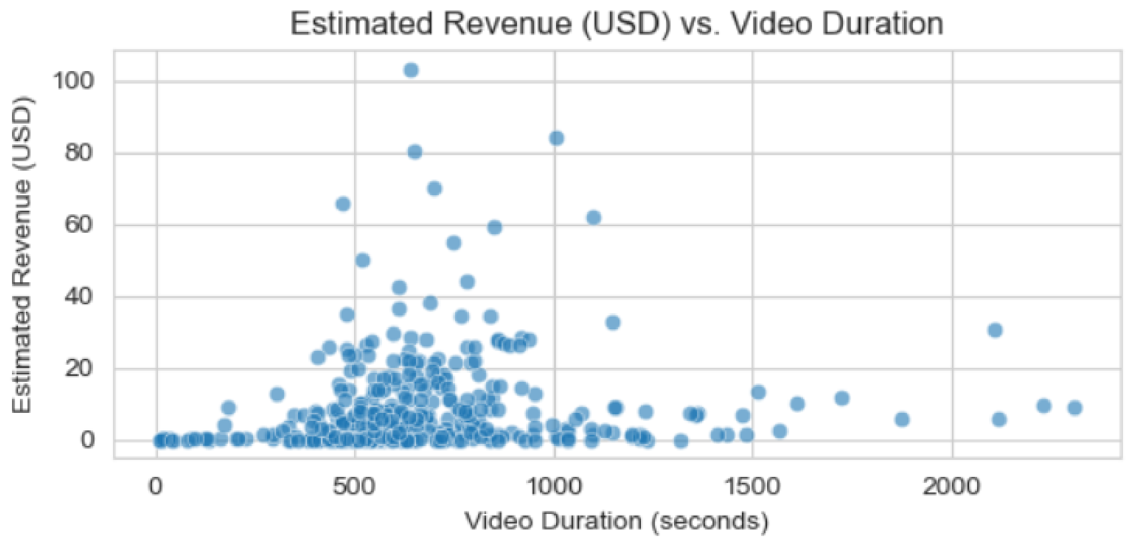
```python
import pandas as pd

# Load the dataset
df = pd.read_csv("youtube_channel_real_performance_analytics.csv")

# Convert 'Video Publish Time' to datetime and extract date
df['Video Publish Time'] = pd.to_datetime(df['Video Publish Time'])
df['Publish Date'] = df['Video Publish Time'].dt.date
```

```python
# Define key performance indicator (KPI) columns for descriptive statistics
kpi_cols = [
    'Views',
    'Watch Time (hours)',
    'Subscribers',
    'Estimated Revenue (USD)',
    'Video Duration',
    'Like Rate (%)',
    'Video Thumbnail CTR (%)'
]

# Calculate descriptive statistics
descriptive_stats = df[kpi_cols].describe().transpose()

# Calculate total/sum for aggregated metrics
total_metrics = df[['Views', 'Watch Time (hours)', 'Subscribers', 'Estimated Revenue (USD)']].sum().to_frame(name='Total')
```

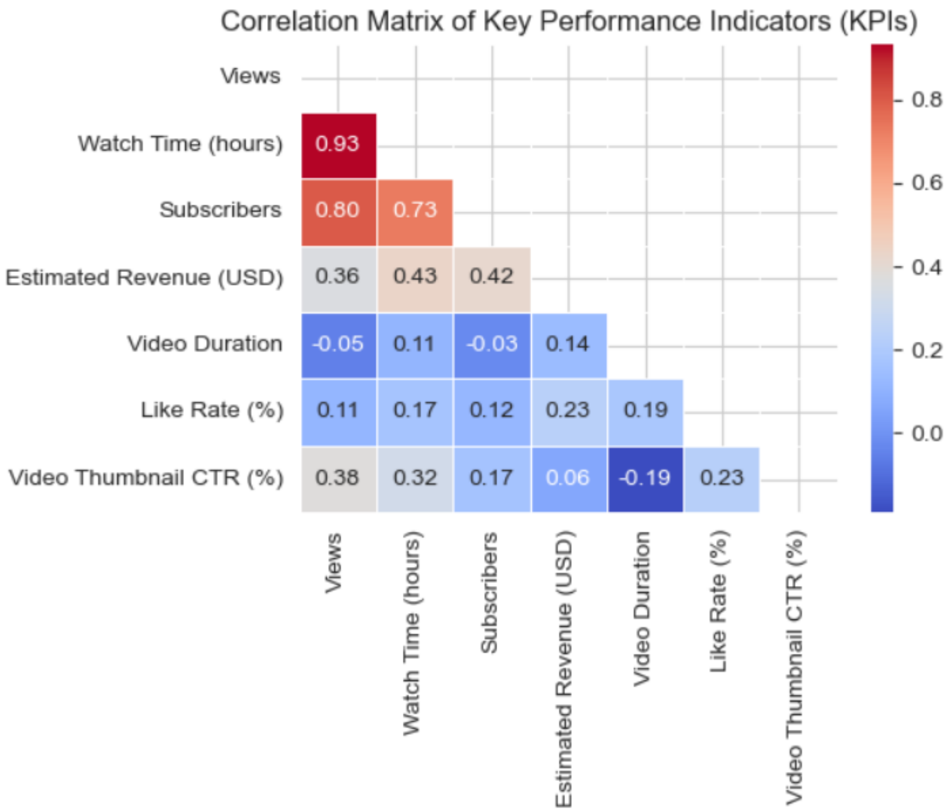
```python
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

sns.set_style("whitegrid")

#Views vs. Watch Time (hours)
plt.figure(figsize=(6, 3))
sns.scatterplot(x='Views', y='Watch Time (hours)', data=df, alpha=0.6)
plt.title('Views vs. Watch Time (hours)')
plt.xlabel('Views')
plt.ylabel('Watch Time (hours)')
plt.ticklabel_format(style='plain', axis='x')
plt.ticklabel_format(style='plain', axis='y')
plt.grid(True)
plt.tight_layout()
plt.show()
plt.close()
```



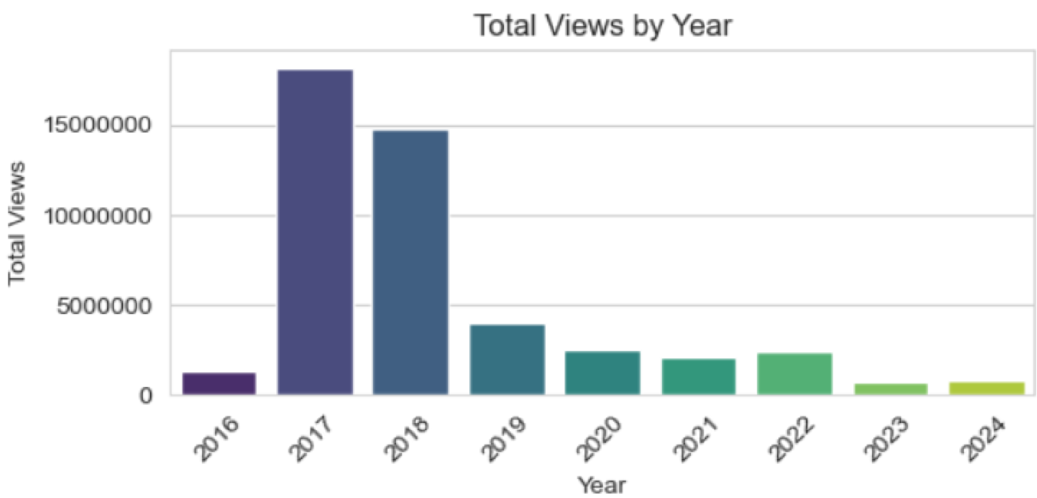
```
#Estimated Revenue (USD) vs. Video Duration
plt.figure(figsize=(6, 3))
sns.scatterplot(x='Video Duration', y='Estimated Revenue (USD)', data=df, alpha=0.6)
plt.title('Estimated Revenue (USD) vs. Video Duration')
plt.xlabel('Video Duration (seconds)')
plt.ylabel('Estimated Revenue (USD)')
plt.grid(True)
plt.tight_layout()
plt.show()
plt.close()
```




```
#Correlation Matrix (select key metrics)
correlation_cols = [
    'Views',
    'Watch Time (hours)',
    'Subscribers',
    'Estimated Revenue (USD)',
    'Video Duration',
    'Like Rate (%)',
    'Video Thumbnail CTR (%)'
]
corr_matrix = df[correlation_cols].corr()

plt.figure(figsize=(6, 5))
# Create a mask for the upper triangle
mask = np.triu(corr_matrix)
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5, mask=mask)
plt.title('Correlation Matrix of Key Performance Indicators (KPIs)')
plt.tight_layout()
plt.show()
plt.close()
```




```python
# Total Views by Year
views_by_year = df.groupby('Year')['Views'].sum().reset_index()

plt.figure(figsize=(6, 3))
sns.barplot(x='Year', y='Views', data=views_by_year, palette='viridis')
plt.title('Total Views by Year')
plt.xlabel('Year')
plt.ylabel('Total Views')
plt.ticklabel_format(style='plain', axis='y')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
plt.close()
```



```python
#Preprocessing & Feature Engineering
df['Video Publish Time'] = pd.to_datetime(df['Video Publish Time'])

# Define duration bins (in seconds)
bins = [0, 300, 600, 900, df['Video Duration'].max() + 1]
labels = ['Short (< 5 min)', 'Medium (5-10 min)', 'Long (10-15 min)', 'Very Long (> 15 min)']
df['Duration Group'] = pd.cut(df['Video Duration'], bins=bins, labels=labels, right=False)

# Define day order for categorical grouping
day_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
df['Day of Week'] = pd.Categorical(df['Day of Week'], categories=day_order, ordered=True)
```

```python
#Performance by Video Thumbnail CTR (%) Quintiles
df['CTR Quintile'] = pd.qcut(df['Video Thumbnail CTR (%)'], q=5, labels=False, duplicates='drop')
df['CTR Quintile'] = df['CTR Quintile'].astype(str)
quintile_map = {'0': 'Q1 (Lowest)', '1': 'Q2', '2': 'Q3', '3': 'Q4', '4': 'Q5 (Highest)'}
df['CTR Quintile'] = df['CTR Quintile'].map(quintile_map).fillna('Q5 (Highest)')

ctr_quintile_performance = df.groupby('CTR Quintile', observed=False)[['Views', 'New Subscribers', 'Estimated Revenue (USD)', 'Video Thumbnail CTR (%)']]

print("Analysis 1: Average Performance by Video Thumbnail CTR (%) Quintile")
print(ctr_quintile_performance.sort_values(by='Video Thumbnail CTR (%)').to_markdown(index=False, numalign="left", stralign="left", floatfmt=".2f"))
```

```
Analysis 1: Average Performance by Video Thumbnail CTR (%) Quintile
| CTR Quintile   | Views     | New Subscribers   | Estimated Revenue (USD)   | Video Thumbnail CTR (%)   |
|:---------------|:----------|:------------------|:--------------------------|:--------------------------|
| Q1 (Lowest)    | 46769.91  | 140.84            | 6.31                      | 3.49                      |
| Q2             | 104462.07 | 365.07            | 11.54                     | 6.92                      |
| Q3             | 154525.10 | 514.29            | 10.89                     | 8.39                      |
| Q4             | 164073.51 | 352.78            | 7.02                      | 9.61                      |
| Q5 (Highest)   | 175312.63 | 382.00            | 8.60                      | 11.22                     |
```

```python
#Revenue Efficiency (Revenue per View)
df['Revenue per View (USD)'] = df['Estimated Revenue (USD)'].div(df['Views']).replace(np.inf, 0).fillna(0)

rpv_by_day = df.groupby('Day of Week', observed=True)['Revenue per View (USD)'].mean().reset_index()
rpv_by_duration = df.groupby('Duration Group', observed=True)['Revenue per View (USD)'].mean().reset_index()

print("\nAnalysis 2: Average Revenue per View (USD) by Day of Week")
print(rpv_by_day.to_markdown(index=False, numalign="left", stralign="left", floatfmt=".6f"))

print("\nAnalysis 2: Average Revenue per View (USD) by Duration Group")
print(rpv_by_duration.to_markdown(index=False, numalign="left", stralign="left", floatfmt=".6f"))
```

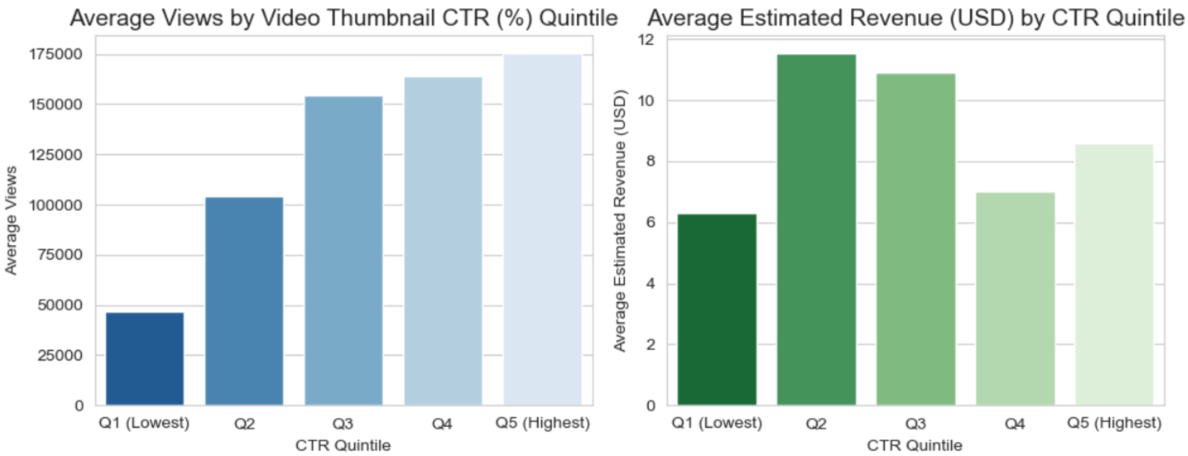
```
Analysis 2: Average Revenue per View (USD) by Day of Week
| Day of Week   | Revenue per View (USD)   |
|:--------------|:-------------------------|
| Monday        | 0.000108                 |
| Tuesday       | 0.000134                 |
| Wednesday     | 0.000083                 |
| Thursday      | 0.000149                 |
| Friday        | 0.000121                 |
| Saturday      | 0.000089                 |
| Sunday        | 0.000110                 |
```

```
Analysis 2: Average Revenue per View (USD) by Duration Group
| Duration Group       | Revenue per View (USD)   |
|:---------------------|:-------------------------|
| Short (< 5 min)      | 0.000035                 |
| Medium (5-10 min)    | 0.000063                 |
| Long (10-15 min)     | 0.000147                 |
| Very Long (> 15 min) | 0.000198                 |
```

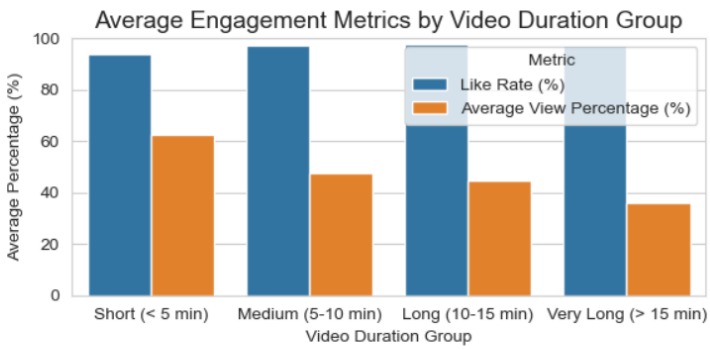
```python
#Engagement Metrics by Video Duration Group
engagement_by_duration = df.groupby('Duration Group', observed=True)[['Like Rate (%)', 'Average View Percentage (%)']].mean().reset_index()

print("\nAnalysis 3: Average Engagement Metrics by Video Duration Group")
print(engagement_by_duration.to_markdown(index=False, numalign="left", stralign="left", floatfmt=".2f"))
```

```
Analysis 3: Average Engagement Metrics by Video Duration Group
| Duration Group       | Like Rate (%)   | Average View Percentage (%)   |
|:---------------------|:----------------|:------------------------------|
| Short (< 5 min)      | 93.97           | 62.53                         |
| Medium (5-10 min)    | 97.49           | 47.46                         |
| Long (10-15 min)     | 97.93           | 44.91                         |
| Very Long (> 15 min) | 97.14           | 36.27                         |
```

```python
#CTR Quintile Performance
plot_data = ctr_quintile_performance.sort_values(by='Video Thumbnail CTR (%)')
plot_data_melted = plot_data.melt(id_vars='CTR Quintile', value_vars=['Views', 'Estimated Revenue (USD)'],
                                  var_name='Metric', value_name='Value')
fig, axes = plt.subplots(1, 2, figsize=(10, 4))

# Plot 1: Views by CTR Quintile
sns.barplot(ax=axes[0], x='CTR Quintile', y='Value', data=plot_data_melted[plot_data_melted['Metric'] == 'Views'], palette='Blues_r')
axes[0].set_title('Average Views by Video Thumbnail CTR (%) Quintile', fontsize=14)
axes[0].set_xlabel('CTR Quintile')
axes[0].set_ylabel('Average Views')
axes[0].ticklabel_format(style='plain', axis='y')

# Plot 2: Estimated Revenue (USD) by CTR Quintile
sns.barplot(ax=axes[1], x='CTR Quintile', y='Value', data=plot_data_melted[plot_data_melted['Metric'] == 'Estimated Revenue (USD)'], palette='Greens_r')
axes[1].set_title('Average Estimated Revenue (USD) by CTR Quintile', fontsize=14)
axes[1].set_xlabel('CTR Quintile')
axes[1].set_ylabel('Average Estimated Revenue (USD)')
axes[1].ticklabel_format(style='plain', axis='y')

plt.tight_layout()
plt.show()
plt.close()
```




```python
#Engagement Metrics by Duration Group
engagement_melted = engagement_by_duration.melt(id_vars='Duration Group', value_vars=['Like Rate (%)', 'Average View Percentage (%)'],
                                                var_name='Metric', value_name='Percentage')

plt.figure(figsize=(6, 3))
sns.barplot(x='Duration Group', y='Percentage', hue='Metric', data=engagement_melted, palette=['#1f77b4', '#ff7f0e'])
plt.title('Average Engagement Metrics by Video Duration Group', fontsize=14)
plt.xlabel('Video Duration Group')
plt.ylabel('Average Percentage (%)')
plt.ylim(0, 100) # Ensure y-axis is sensible for percentages
plt.legend(title='Metric')
plt.tight_layout()
plt.show()
plt.close()
```



CONCLUSION

A clear route to optimizing channel development and income is revealed by the thorough examination of key performance indicators. Despite lower view completion rates, the data clearly supports a strategy shift toward longer-form content, with a focus on the 10–15 minute duration band. These films generate much greater Revenue per View (RPV). This increased effectiveness implies that the audience that is retained in these longer formats has a far higher value to marketers.

Additionally, we found a critical mismatch in front-end optimization: the highest click-through rates do not result in the highest revenue. This suggests that the thumbnail and title strategy should be changed to drive qualified, high-value interaction rather than merely clickbait. The 2017 peak's historical abnormality also offers a chance; examining the particular content and platform environment of that year may reveal useful, replicable growth tactics.

We strongly advise putting the three main recommendations in this research into practice in order to take immediate advantage of these findings. Based on the successful Q2 CTR characteristics, the next immediate step is to start the 2017 performance deep-dive and start A/B testing different thumbnail techniques. To guarantee the success of this optimized strategy, RPV and Average View Percentage for fresh content must be continuously monitored.