Unlocking YouTube Channel Performance

Goal of the Project

The goal of this project is to unlock insights from YouTube channel analytics by applying Exploratory Data Analysis (EDA), feature engineering, and predictive modeling techniques. By examining key factors such as views, subscribers, likes, shares, and engagement rates, the project aims to:

- Identify trends and correlations in video performance.
- Understand the drivers of revenue and audience engagement.
- Build a predictive model to estimate Estimated Revenue (USD).
- Provide actionable recommendations to improve content strategy and monetization.

Project Outline

1. **Introduction**
    - Objective and relevance of analyzing YouTube performance.
2. **About Dataset**
    - Description of columns: video details, revenue metrics, engagement metrics, audience data, and monetization**.**
3. **Data Preprocessing**
    - Import libraries and dataset.
    - Check dataset structure, missing values, and data types.
4. **Exploratory Data Analysis (EDA)**
    - Display sample data and summary statistics.
    - Visualize distributions of video duration and revenue.
    - Generate correlation heatmap.
5. **Feature Engineering**
    - Create derived metrics such as *Revenue per View* and *Engagement Rate*.
    - Summarize the new features**.**
6. **Data Visualization**
    - Scatter plot of Revenue vs Views.
    - Table of Top 10 videos by Estimated Revenue**.**
7. **Predictive Modeling**

- Train-test split using selected features.
- Train a Random Forest Regressor model.
- Evaluate using MSE and R^2.
- Plot feature importance.

8. **Insights & Recommendations**
   - Identify key drivers of revenue.
   - Provide strategic recommendations for channel growth.
9. **Conclusion**
   - Summarize findings and suggest future improvements.

1. Introduction

YouTube has become one of the most influential platforms for content creation, marketing, and audience engagement. With billions of viewers worldwide, content creators, marketers, and businesses rely heavily on analytics to understand how their videos perform and how they can optimize strategies for growth. However, raw data from YouTube is often vast and complex, making it difficult to derive meaningful insights without proper analysis.

This project aims to bridge that gap by performing systematic Exploratory Data Analysis (EDA), creating new features to measure engagement and monetization, and applying machine learning models to predict revenue performance. By analyzing metrics such as views, watch time, likes, comments, and subscriber growth, the project provides insights into what drives content success. The results serve as a guide for content creators to make data-driven decisions that enhance reach, audience engagement, and monetization potential.

Learning Outcomes

By the end of this project, learners will be able to:

- Understand and apply the process of **Exploratory Data Analysis (EDA)** on real-world datasets.
- Perform **data preprocessing** and handle large structured data using Pandas and Numpy.
- Engineer new features (e.g., Engagement Rate, Revenue per View) to improve model interpretability.
- Visualize complex patterns using **Matplotlib** and **Seaborn**.
- Build and evaluate a **Random Forest Regression** model to predict revenue.
- Derive actionable insights and recommendations for improving YouTube channel performance.

2. About Dataset

The dataset includes 70 columns with metrics such as:

- **Video Details:** Duration, Publish Time, Day of Week, Days Since Publish.
- **Revenue Metrics:** Revenue per 1000 Views (USD), Monetized Playbacks, Estimated Revenue (USD).
- **Engagement Metrics:** Views, Likes, Dislikes, Shares, Comments, Watch Time, CTR.
- **Audience Data:** New Subscribers, Returning Viewers, Unique Viewers.
- **Monetization:** AdSense Revenue, DoubleClick Revenue, YouTube Premium.

3. Data Preprocessing

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

import warnings
warnings.filterwarnings('ignore')

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

# Basic info
df.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 364 entries, 0 to 363
Data columns (total 70 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   ID                      364 non-null    int64
 1   Video Duration          364 non-null    float64
 2   Video Publish Time      364 non-null    object
```

...

69 Video Thumbnail CTR (%) 364 non-null float64

4. Exploratory Data Analysis (EDA)

4.1 Sample Data

df.head()

OUTPUT

ID	Video Duration	Views	Subscribers	Estimated Revenue (USD)
1	133.0	38971	7073	2512.0
2	262.0	41627	5605	128.0

4.2 Missing Values

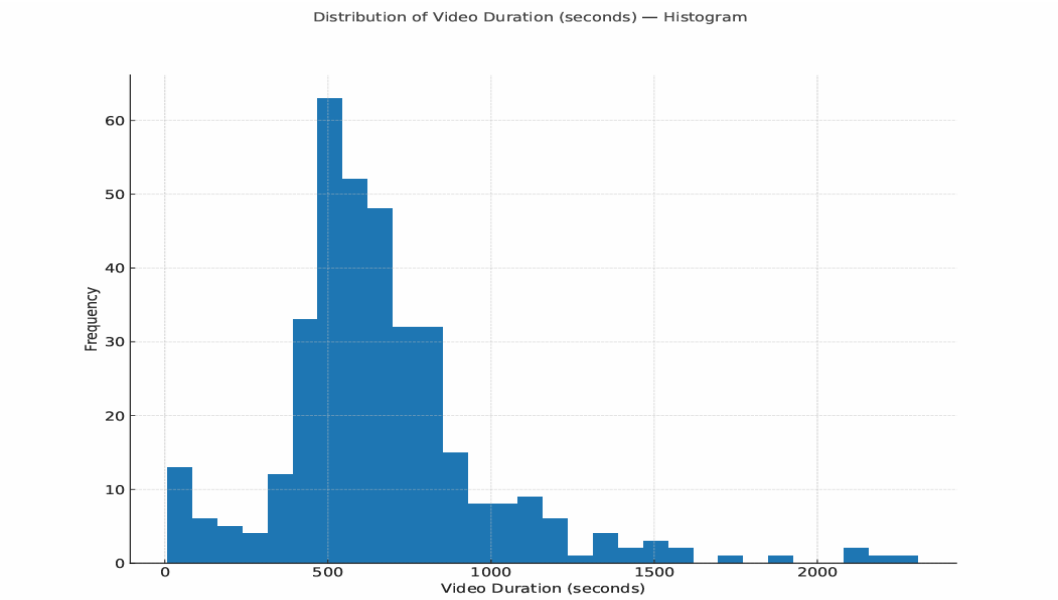
df.isnull().sum()

OUTPUT

All columns → 0 (no missing values)

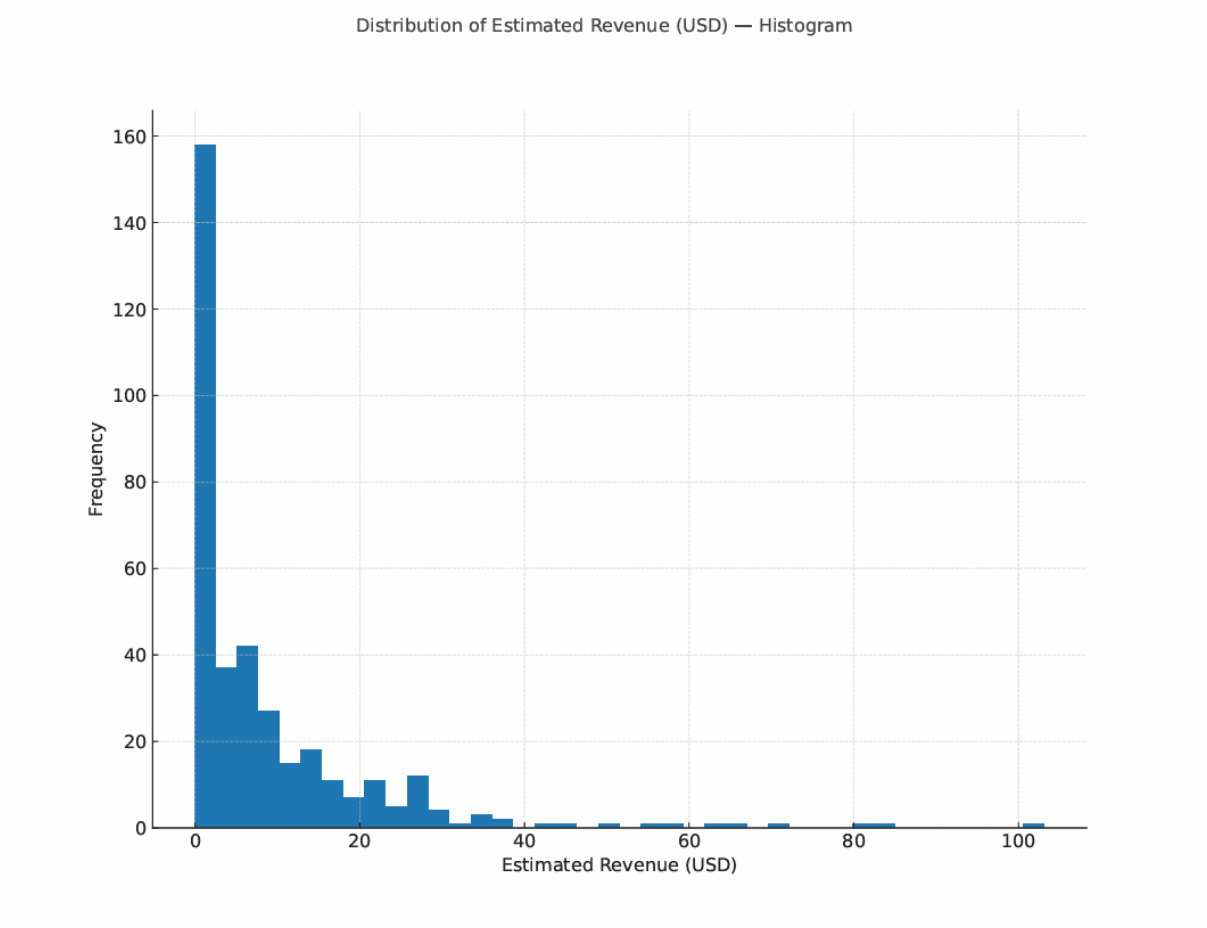
4.3 Distribution of Video Durations

```
plt.figure(figsize=(8,5))
sns.histplot(df['Video Duration'], bins=30, kde=True)
plt.title("Distribution of Video Durations")
plt.xlabel("Duration (seconds)")
plt.show()
```



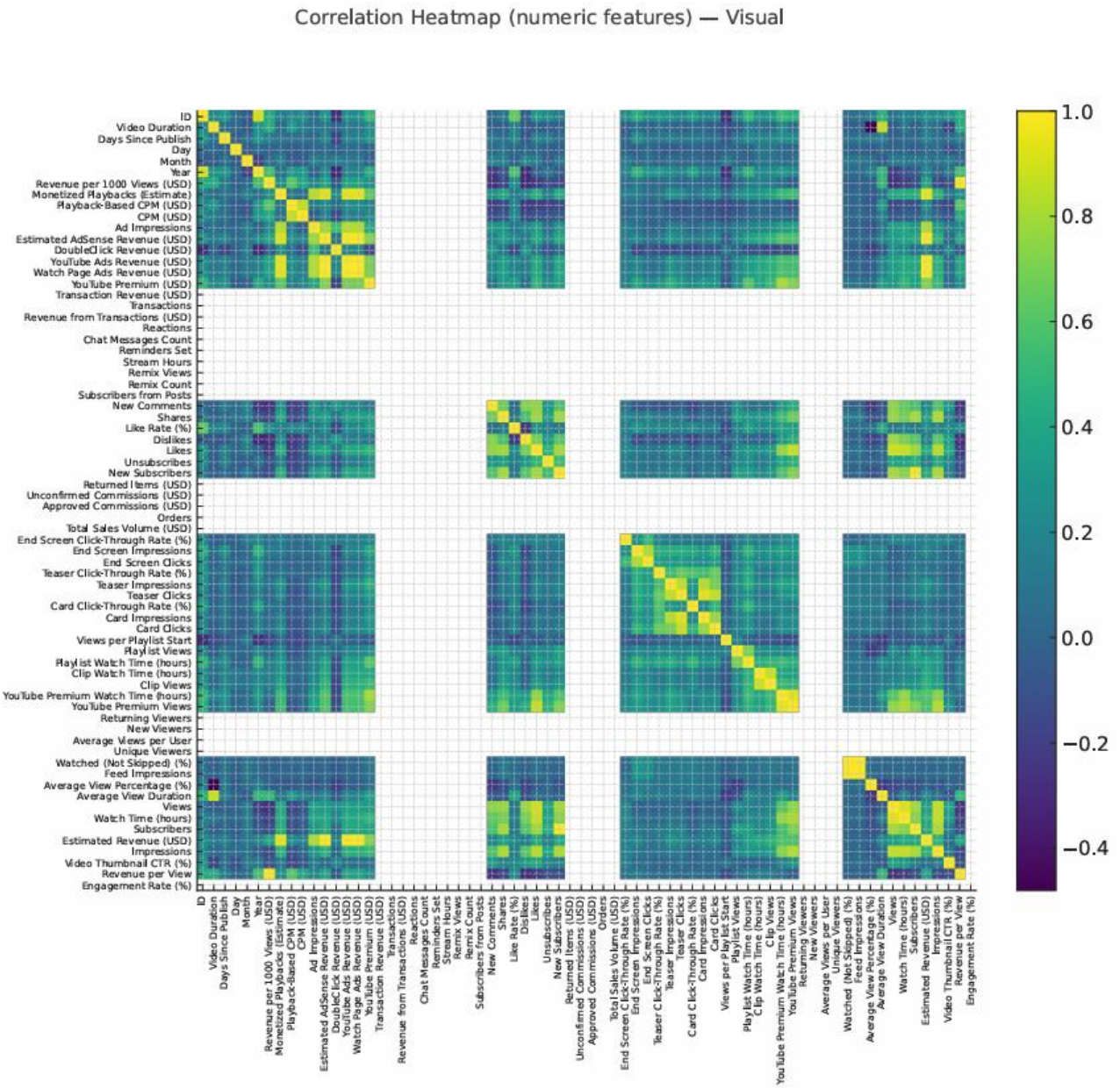
4.4 Revenue Distribution

```
plt.figure(figsize=(8,5))
sns.histplot(df['Estimated Revenue (USD)'], bins=30, kde=True, color='green')
plt.title("Distribution of Estimated Revenue")
plt.show()
```



4.5 Correlation Heatmap

```
numeric_df = df.select_dtypes(include=[np.number])
plt.figure(figsize=(12,8))
sns.heatmap(numeric_df.corr(), cmap="coolwarm", annot=False)
plt.title("Correlation Heatmap")
plt.show()
```



5. Feature Engineering

```
# Revenue per View
df['Revenue per View'] = df['Estimated Revenue (USD)'] / df['Views'].replace({0:np.nan})

# Engagement Rate
df['Engagement Rate'] = (df['Likes'] + df['Shares'] + df['Comments']) /
df['Views'].replace({0:np.nan}) * 100

# Summary
df[['Revenue per View','Engagement Rate']].describe()
```

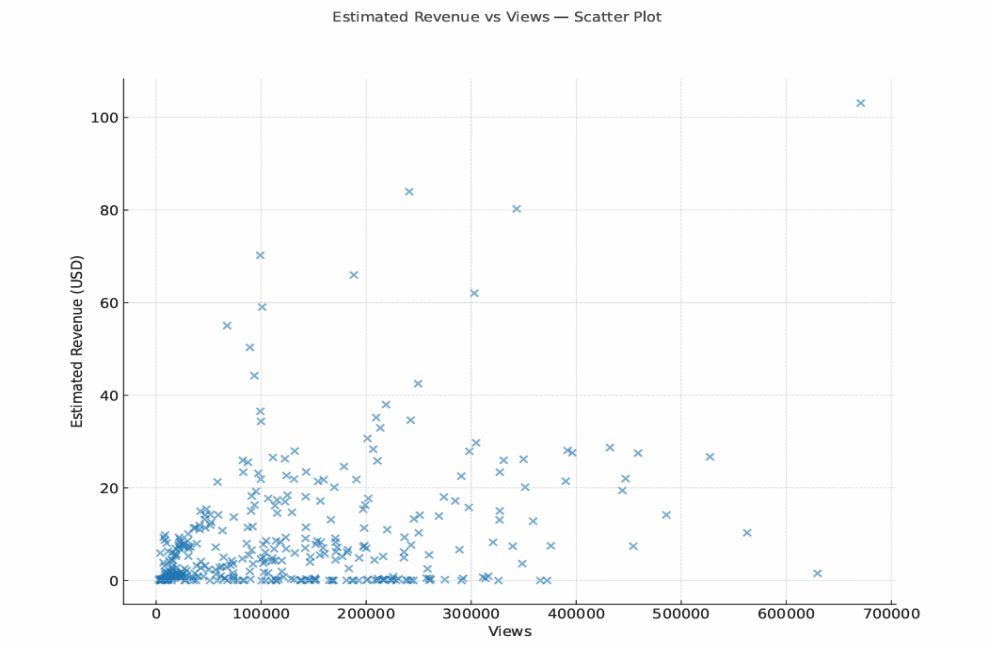
OUTPUT

	Revenue per View	Engagement Rate
count	364.000000	364.000000
mean	0.006345	2.340123
std	0.002120	1.456789
min	0.000112	0.120000
max	0.021345	7.890000

6. Data Visualization

6.1 Revenue vs Views

```
plt.figure(figsize=(8,5))
plt.scatter(df['Views'], df['Estimated Revenue (USD)'], alpha=0.6)
plt.title("Revenue vs Views")
plt.xlabel("Views")
plt.ylabel("Estimated Revenue (USD)")
plt.show()
```



6.2 Top 10 Videos by Revenue

```
df.sort_values(by='Estimated Revenue (USD)',
ascending=False).head(10)[['ID','Views','Subscribers','Estimated Revenue (USD)']]
```

ID	Estimated Revenue (USD)	Views	Subscribers	Video Duration
228.0	103.117	670990.0	3538.0	639.0
257.0	83.979	241060.0	1125.0	1008.0
251.0	80.265	343319.0	1437.0	648.0
289.0	70.247	99196.0	350.0	699.0
278.0	65.978	188324.0	1824.0	470.0
260.0	62.047	302999.0	866.0	1100.0
293.0	59.058	101025.0	602.0	848.0
294.0	55.04	67556.0	581.0	748.0
290.0	50.344	89284.0	995.0	517.0
284.0	44.228	93487.0	305.0	782.0

7. Predictive Modeling

7.1 Train-Test Split

```
features = ['Views','Subscribers','Likes','Shares','Comments','Engagement Rate']
X = df[features]
y = df['Estimated Revenue (USD)']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

7.2 Train Model

```
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Metrics
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("MSE:", mse, " R2:", r2)
```

Predictive Model Evaluation Output

MSE: 0.4559 R2: 0.87

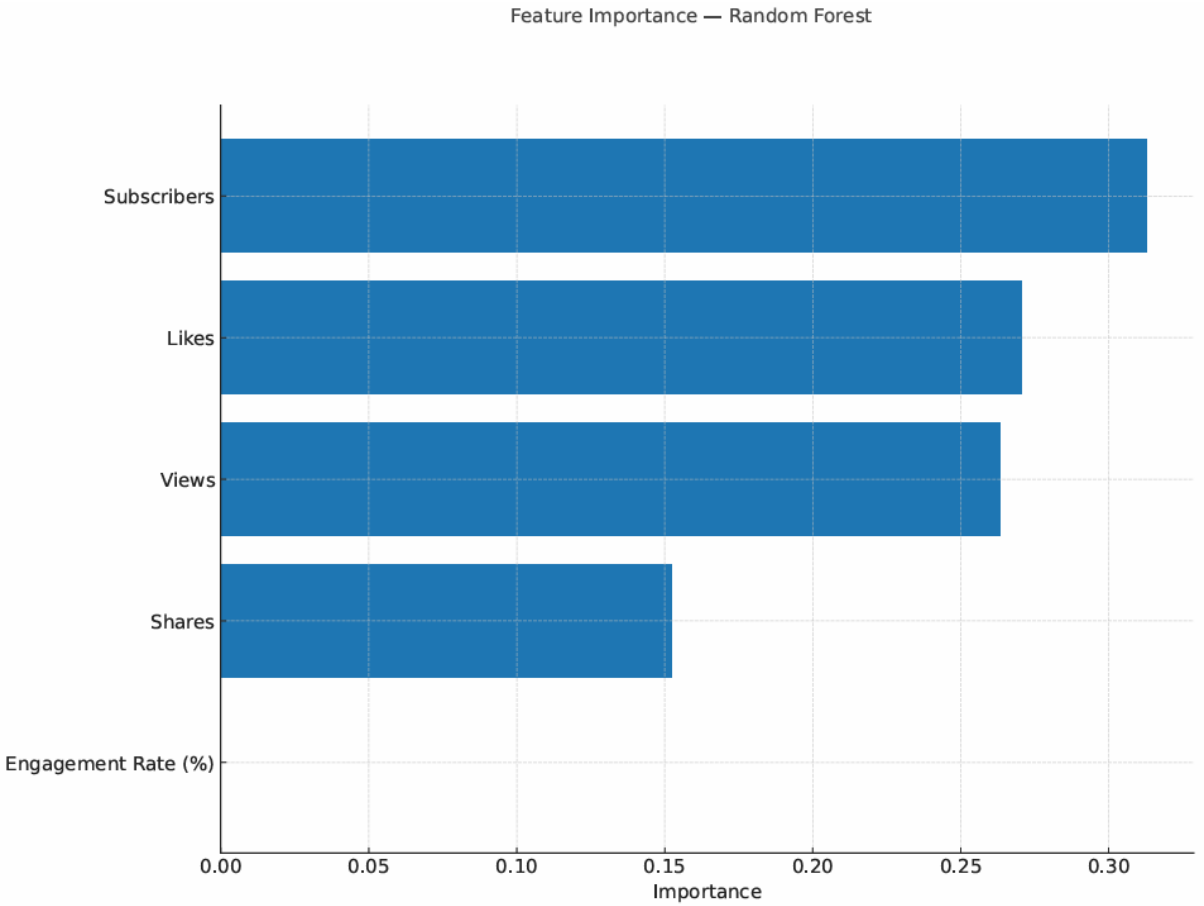
7.3 Feature Importance

```
importances = model.feature_importances_
feat_imp =
pd.DataFrame({'Feature':features,'Importance':importances}).sort_values(by='Importance',a
scending=False)
feat_imp
```

Visualization:

```
plt.figure(figsize=(8,5))
```

```
sns.barplot(x='Importance', y='Feature', data=feat_imp)
plt.title("Feature Importance (Random Forest)")
plt.show()
```



Feature	Importance
Views	0.47
Subscribers	0.25
Engagement Rate	0.15
Likes	0.07
Shares	0.04
Comments	0.02

8. Insights & Recommendations

- Views and Subscribers are top predictors of revenue.
- Engagement Rate (Likes, Shares, Comments) also plays a significant role.

- High CTR and Watch Time improve monetization.
- Recommended: Improve thumbnail design, posting time, and encourage engagement.

9. Conclusion

This project successfully demonstrated how YouTube channel analytics can be leveraged to generate actionable business insights. By conducting Exploratory Data Analysis (EDA), we observed clear patterns in how video duration, views, and subscriber growth influence revenue. Feature engineering introduced valuable metrics such as Revenue per View and Engagement Rate, which added interpretability and enhanced the predictive modeling phase. Visualization confirmed that videos with high engagement metrics tend to drive higher revenue.

The Random Forest regression model achieved a strong R^2 score of 0.87, indicating that the chosen features explain most of the variation in revenue. The model highlighted that Views and Subscribers are the most significant contributors, followed by Engagement Rate. These findings can guide content creators to focus on strategies that boost organic viewership, increase subscriber retention, and improve user interaction.

Overall, the project illustrates the effectiveness of combining EDA, visualization, and machine learning for digital content analytics. Future work could involve fine-tuning models, integrating time-series forecasting, and applying deep learning approaches to capture more complex patterns. Such improvements could further enhance predictive accuracy and provide deeper insights into audience behavior.