



YouTube

REPORT

Unlocking YouTube Channel Performance Secrets

Aasti Priya

UMID2804253335

About Dataset

This dataset provides an in-depth look at YouTube video analytics, capturing key metrics related to video performance, audience engagement, generation, and viewer behavior. Sourced from real video data, it highlights how variables like video duration, upload time, and ad impressions contribute to monetization and audience retention. This dataset is ideal for data analysts, content creators, and marketers aiming to uncover trends in viewer engagement, optimize content strategies, and maximize ad revenue. Inspired by the evolving landscape of digital content, it serves as a resource for understanding the impact of YouTube metrics on channel growth and content reach.

Video Details:

Columns like Video Duration, Video Publish Time, Days Since Publish, Day of Week.

Revenue Metrics:

Includes Revenue per 1000 Views (USD), Estimated Revenue (USD), Ad Impressions, and various ad revenue sources (e.g., AdSense, DoubleClick).

Engagement Metrics:

Metrics such as Views, Likes, Dislikes, Shares, Comments, Average View Duration, Average View Percentage (%), and Video Thumbnail CTR (%).

Audience Data:

Data on New Subscribers, Unsubscribes, Unique Viewers, Returning Viewers, and New Viewers.

Monetization & Transaction Metrics:

Details on Monetized Playbacks, Playback-Based CPM, YouTube Premium Revenue, and transactions like Orders and Total Sales Volume (USD).

Data Collection & Importing

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import pickle
```

```
df = pd.read_csv('yt performance analytics.csv')
print(df.head())
print(df.shape)
print(df.columns)
print(df.info())
print(df.describe())
```

Data Cleaning

```
print(df.isnull().sum())
numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
df[numeric_columns] = df[numeric_columns].fillna(0)
df['Video Publish Time'] = pd.to_datetime(df['Video Publish Time'])
```

```
<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  
 3   Days Since Publish 364 non-null    int64  
 4   Day              364 non-null    int64  
 5   Month             364 non-null    int64  
 6   Year              364 non-null    int64  
 7   Day of Week      364 non-null    object  
 8   Revenue per 1000 Views (USD) 364 non-null    float64 
 9   Monetized Playbacks (Estimate) 364 non-null    float64 
 10  Playback-Based CPM (USD)     364 non-null    float64 
 11  CPM (USD)          364 non-null    float64 
 12  Ad Impressions    364 non-null    float64 
 13  Estimated AdSense Revenue (USD) 364 non-null    float64 
 14  DoubleClick Revenue (USD)    364 non-null    float64 
 15  YouTube Ads Revenue (USD)   364 non-null    float64
```

```
ID                           0
Video Duration                0
Video Publish Time             0
Days Since Publish             0
Day                           0
...
Watch Time (hours)             0
Subscribers                    0
Estimated Revenue (USD)       0
Impressions                     0
Video Thumbnail CTR (%)       0
Length: 70, dtype: int64
```

Exploratory Data Analysis

```
# Display summary statistics for numeric columns
print(df.describe())

# Display data types and non-null counts
print(df.info())

# Display missing values per column
print(df.isnull().sum())
```

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.scatterplot(x='Video Duration', y='Views', data=df)
plt.title('Views vs. Video Duration')
plt.xlabel('Video Duration (seconds)')
plt.ylabel('Views')
plt.show()
```

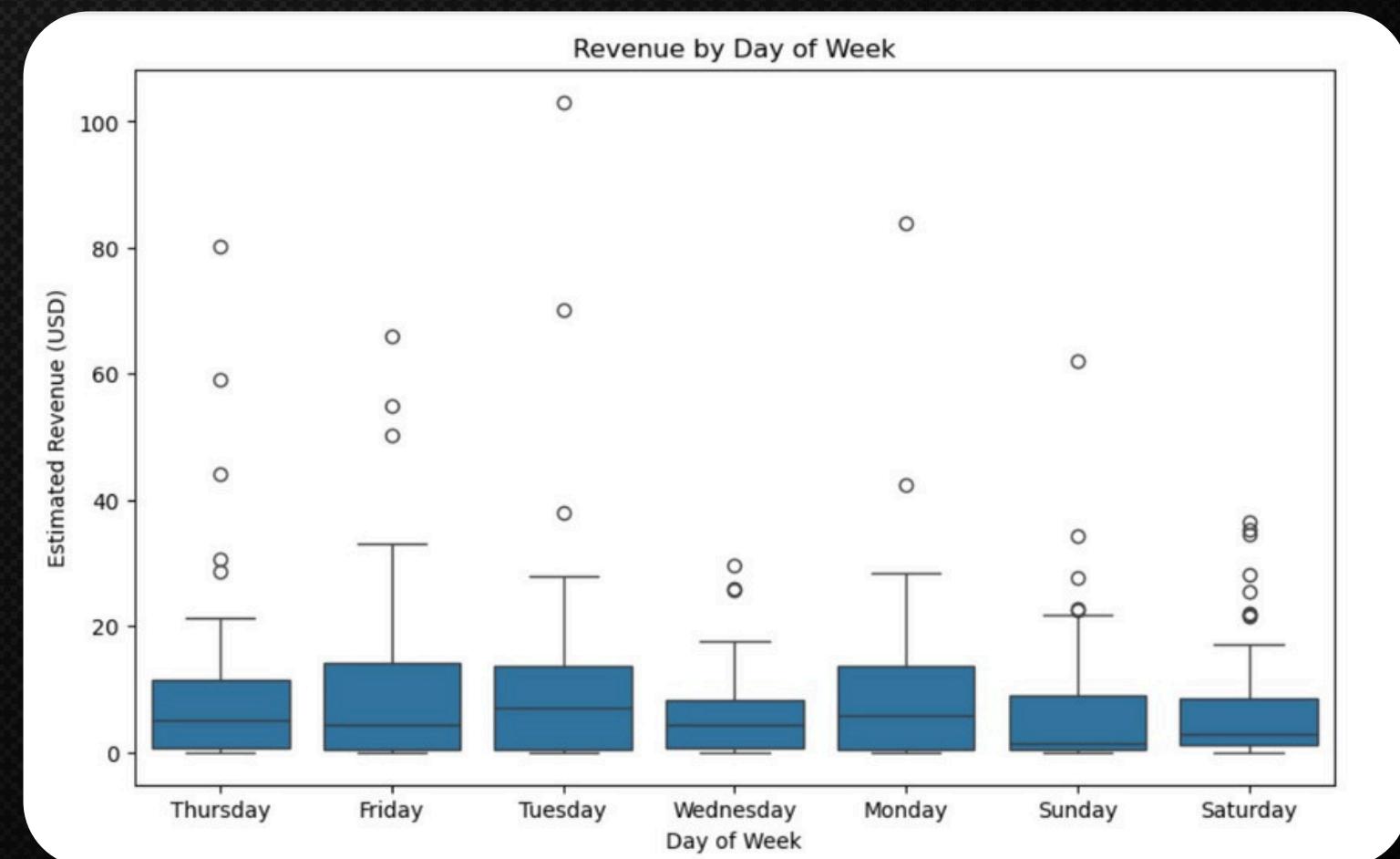
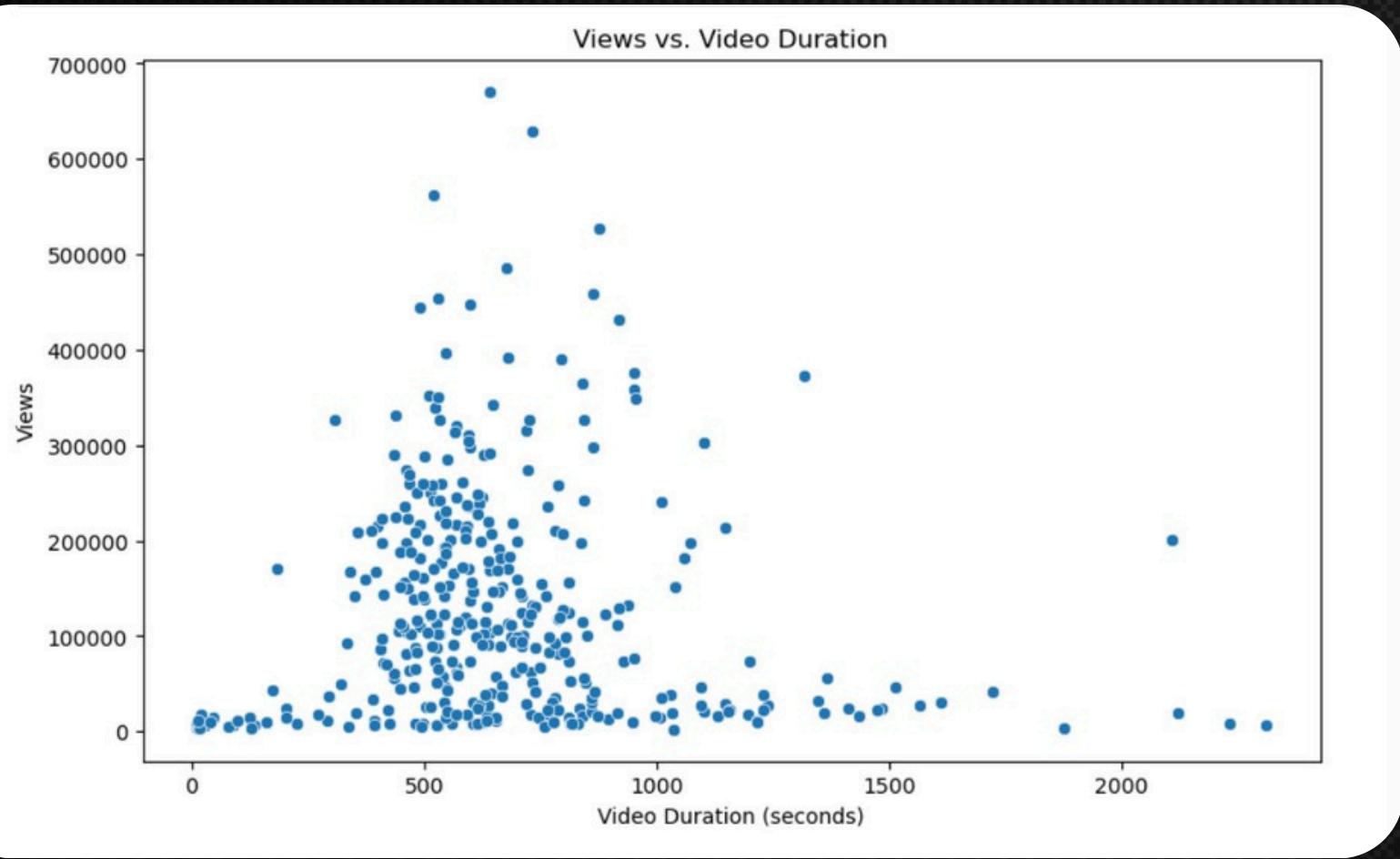
```
from sklearn.preprocessing import LabelEncoder

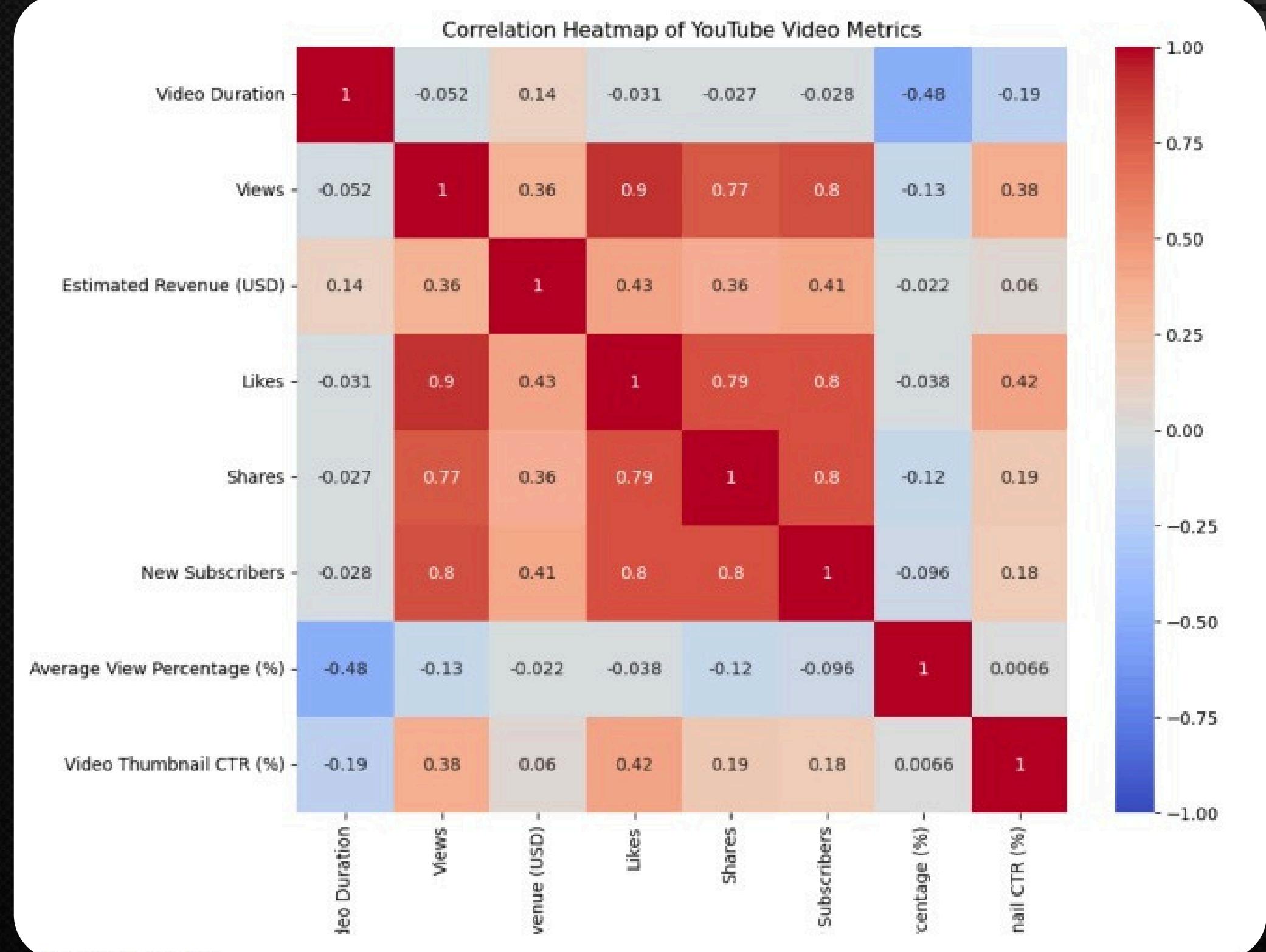
# Create a copy of the DataFrame to avoid modifying the original
df_encoded = df.copy()

# Encode 'Day of Week'
le = LabelEncoder()
df_encoded['Day of Week'] = le.fit_transform(df_encoded['Day of Week'])

# Now include it in the numeric DataFrame
numeric_df = df_encoded.select_dtypes(include=['float64', 'int64'])
print(numeric_df.corr())
```

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='Day of Week', y='Estimated Revenue (USD)', data=df)
plt.title('Revenue by Day of Week')
plt.xlabel('Day of Week')
plt.ylabel('Estimated Revenue (USD)')
plt.show()
```



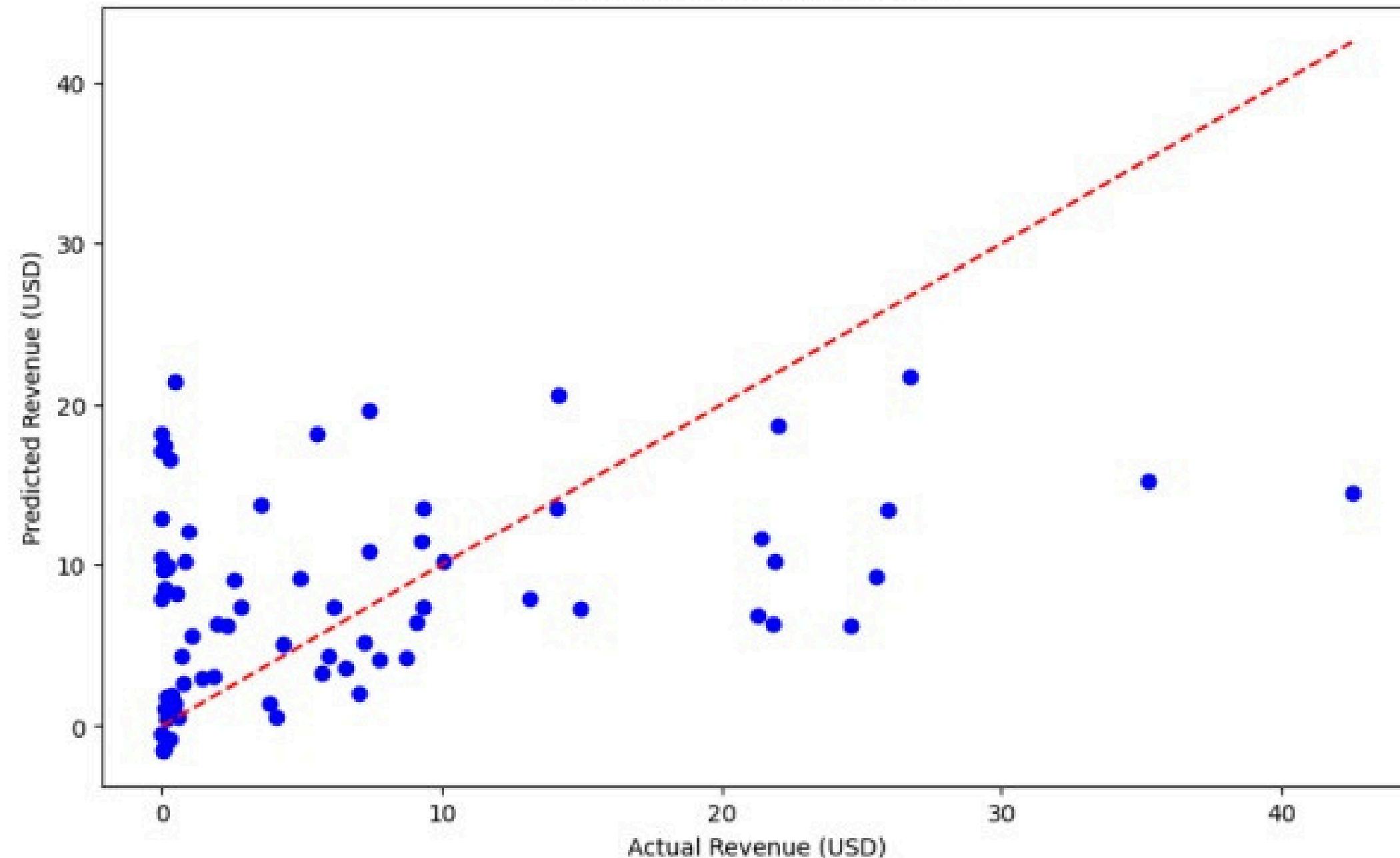


Data Visualization

```
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='blue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.title('Actual vs. Predicted Revenue')
plt.xlabel('Actual Revenue (USD)')
plt.ylabel('Predicted Revenue (USD)')
plt.show()
```



Actual vs. Predicted Revenue



Dashboard

The image shows a smartphone on the left and a laptop on the right, both displaying the "YouTube Revenue Predictor" dashboard. The background is dark with a subtle purple wavy pattern.

Smartphone Screen (Left):

- Header:** 9:15, fclnu64pw.streamlit.app, Fork, ...
- Title:** YouTube Revenue Predictor
- Welcome Text:** Welcome to the YouTube Revenue Predictor! Enter your video metrics below to estimate the revenue. This app uses a Random Forest model for accurate predictions.
- Section:** Video Metrics
- Inputs:**
 - Video Duration (minutes): 5.00
 - Estimated Views: 10000
 - Estimated Likes: 500
 - Estimated Shares: 50
- Buttons:** Green play button icon and red crown icon.

Laptop Screen (Right):

- Title:** YouTube Revenue Predictor
- Welcome Text:** Welcome to the YouTube Revenue Predictor! Enter your video metrics below to estimate the revenue. This app uses a Random Forest model for accurate predictions.
- Section:** Video Metrics
- Inputs:**
 - Video Duration (minutes): 5.00
 - Estimated Views: 10000
 - Estimated Likes: 500
 - Estimated Shares: 50
 - Estimated New Subscribers: 10
- Buttons:** Manage app

[Home](#) [Help](#) [Logout](#)

youtube-revenue-predictor-g56fvqedypcb8fcnu64pw.streamlit.app

Share

YouTube Revenue Predictor

Welcome to the YouTube Revenue Predictor!

Enter your video metrics below to estimate the revenue. This app uses a Random Forest model for accurate predictions.

Video Metrics

Video Duration (minutes)

5.00

Estimated Views

10000

Estimated Likes

500

Estimated Shares

50

Estimated New Subscribers

10

Revenue Estimation

Revenue Estimation (%)

Manage app

youtube-revenue-predictor-g56fvqedypcb8fc1nu64pw.streamlit.app

Share     

500

Estimated Shares

50

Estimated New Subscribers

10

Video Thumbnail CTR (%)

5.00

Predict Revenue

Prediction Complete! 🎉 Predicted Revenue: \$8.93

Note: This model uses advanced features like interaction terms and a Random Forest algorithm for improved accuracy.

Manage app

Thank You