YouTube

REPORT

Aasti Priya

About Dataset

This dataset provides an in-depth look at YouTube video analytics, capturing key metrics related to video performance, audience engagement, revenue 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).

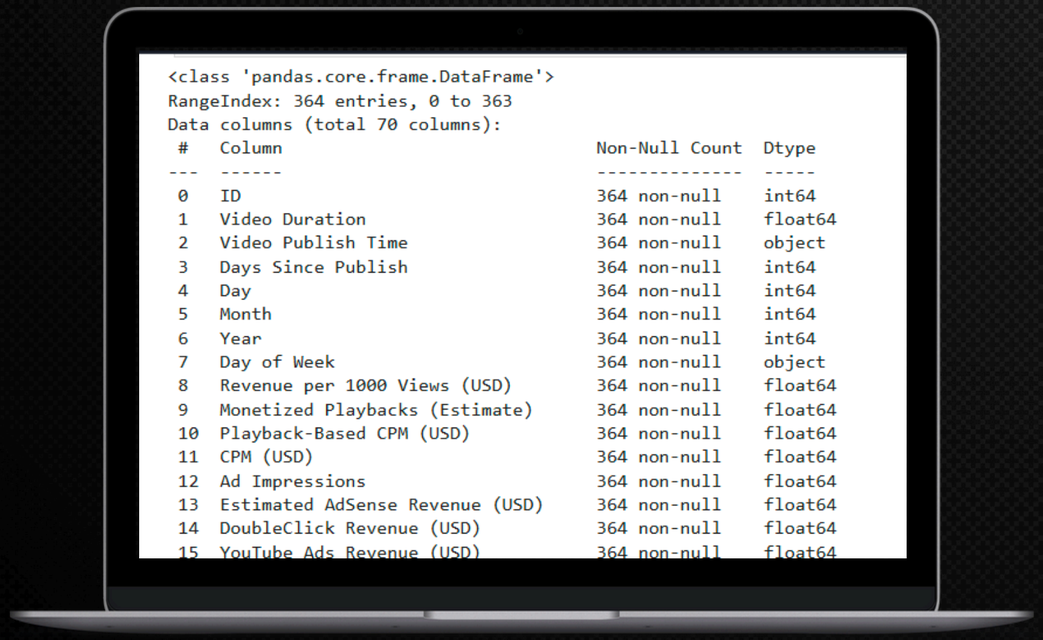
Data Collection & Importing

```
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

```python
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
```

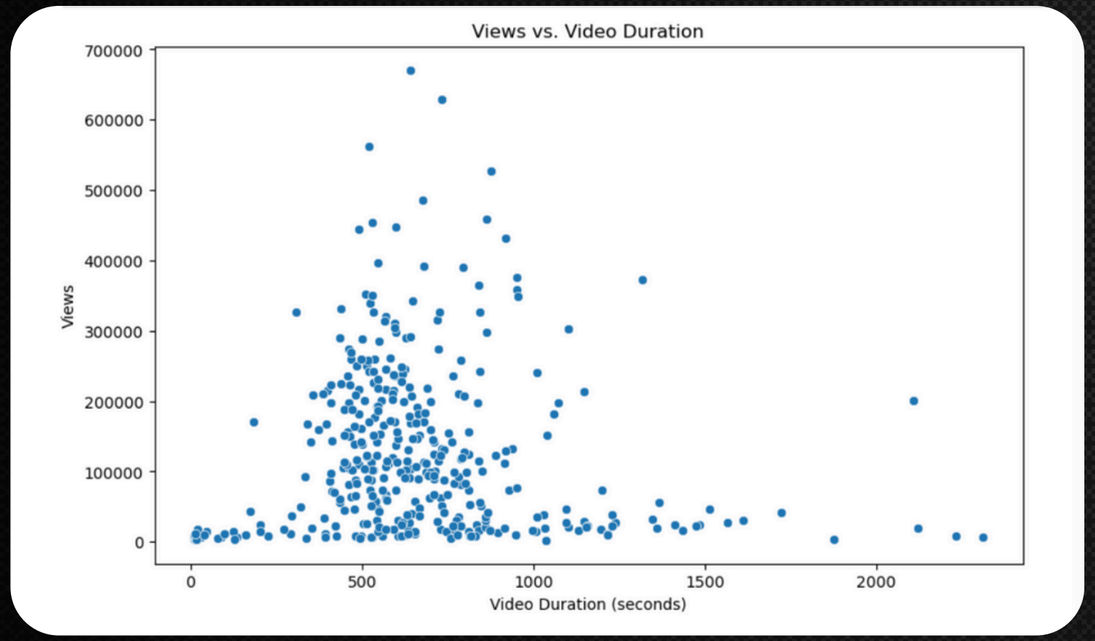
Exploratory Data Analysis

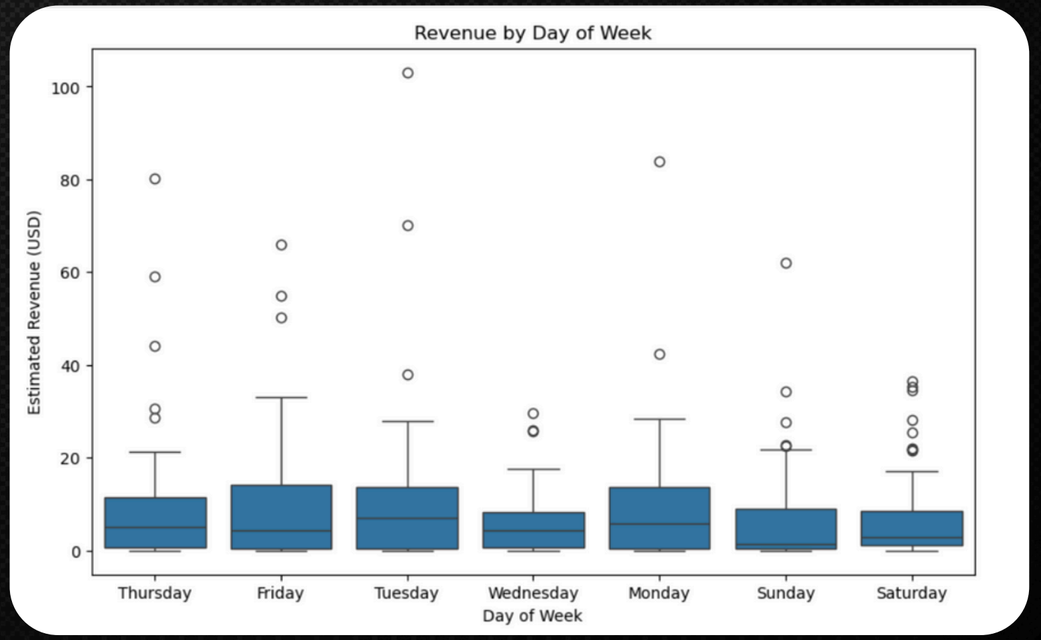
```
# Display summary statistics for numeric columns
print(df.describe())

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

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

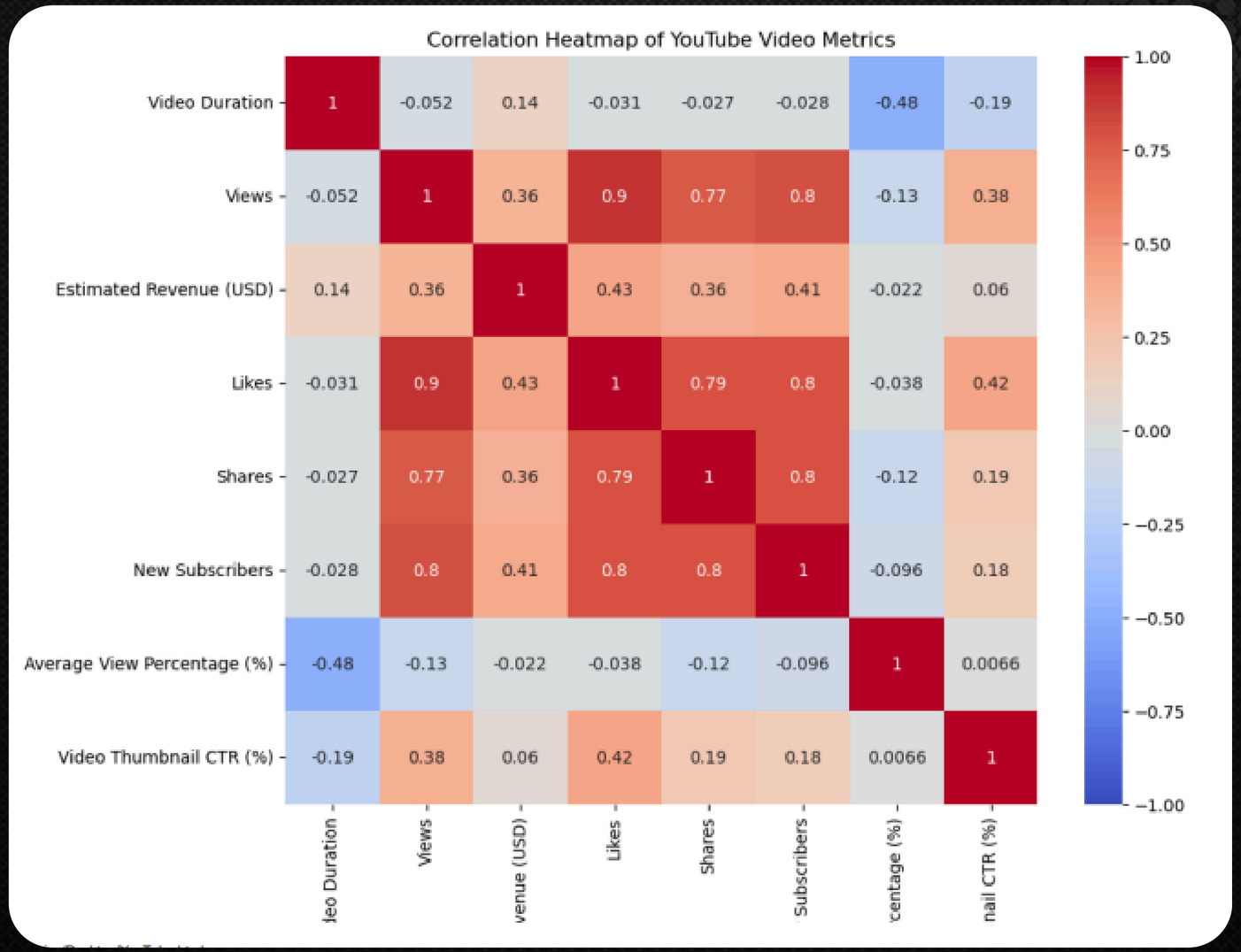
```
import matplotlib.pyplot as plt
import seaborn as sns

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()
```

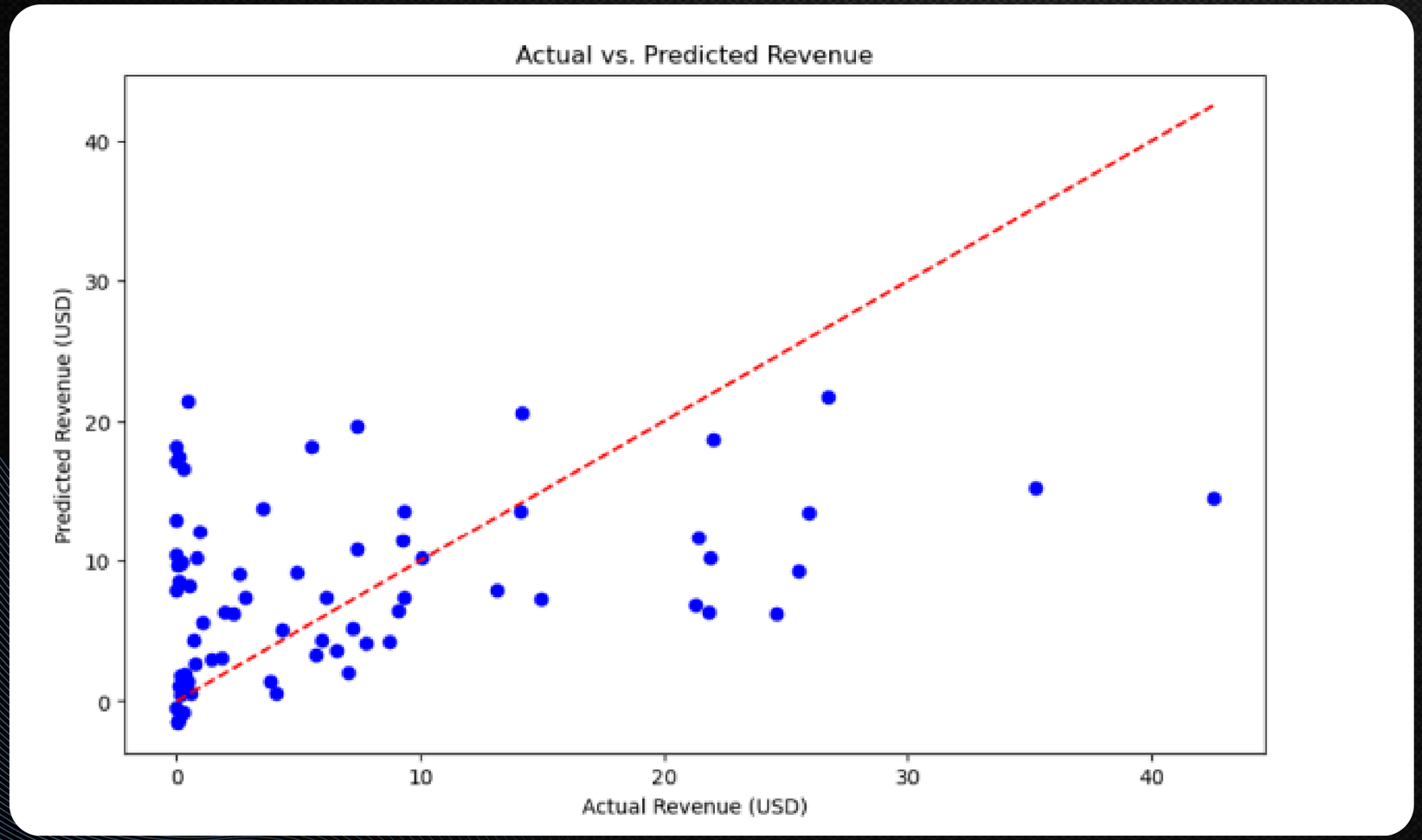

Views vs. Video Duration
Views
Video Duration (seconds)
700000
600000
500000
400000
300000
200000
100000
0
0
500
1000
1500
2000


Revenue by Day of Week
Estimated Revenue (USD)
Day of Week
100
80
60
40
20
0
Thursday
Friday
Tuesday
Wednesday
Monday
Sunday
Saturday

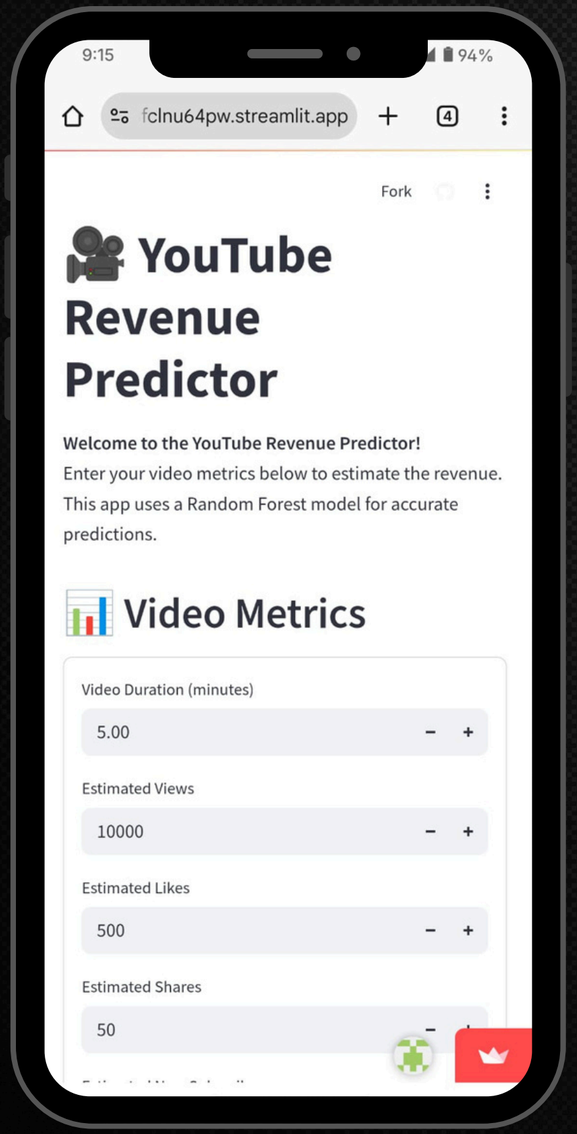

Correlation Heatmap of YouTube Video Metrics
Video Duration
Views
Estimated Revenue (USD)
Likes
Shares
New Subscribers
Average View Percentage (%)
Video Thumbnail CTR (%)
1 -0.052 0.14 -0.031 -0.027 -0.028 -0.48 -0.19
-0.052 1 0.36 0.9 0.77 0.8 -0.13 0.38
0.14 0.36 1 0.43 0.36 0.41 -0.022 0.06
-0.031 0.9 0.43 1 0.79 0.8 -0.038 0.42
-0.027 0.77 0.36 0.79 1 0.8 -0.12 0.19
-0.028 0.8 0.41 0.8 0.8 1 -0.096 0.18
-0.48 -0.13 -0.022 -0.038 -0.12 -0.096 1 0.0066
-0.19 0.38 0.06 0.42 0.19 0.18 0.0066 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

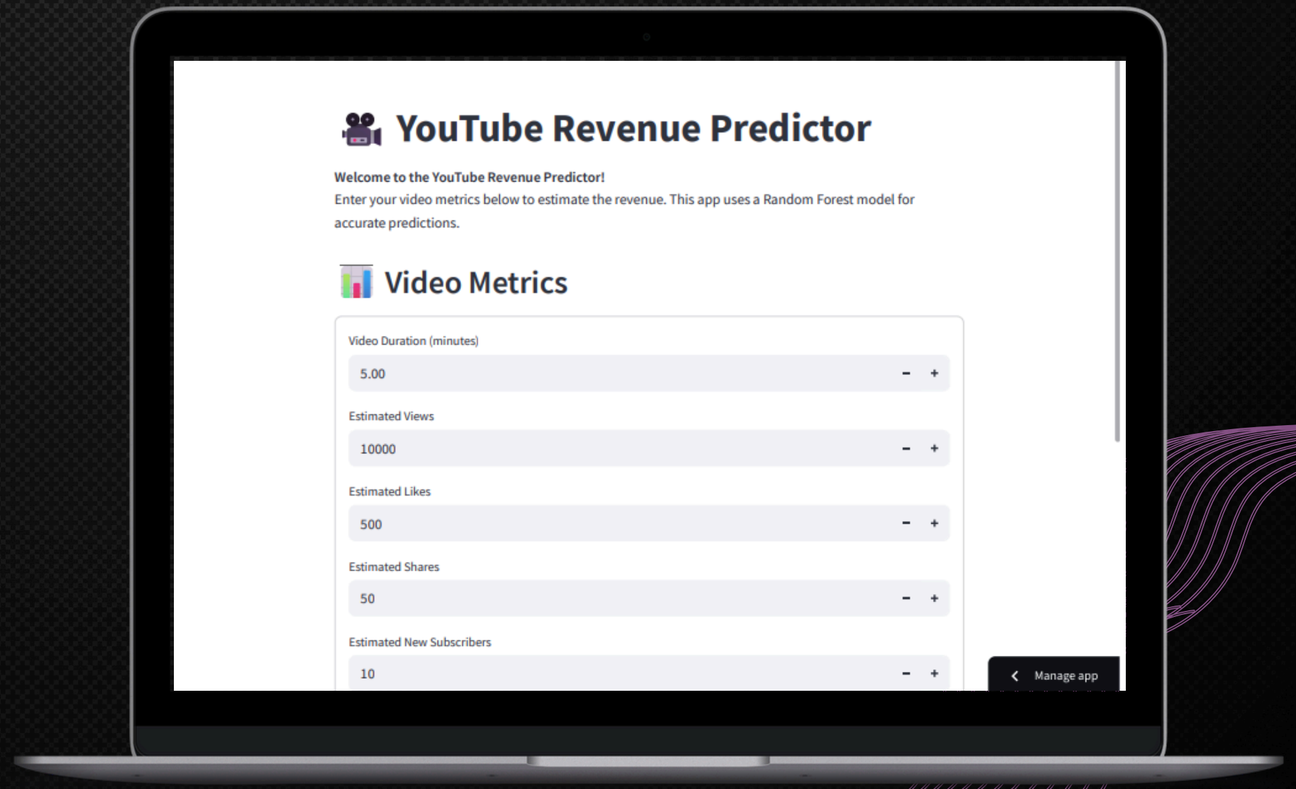
Data Visualization

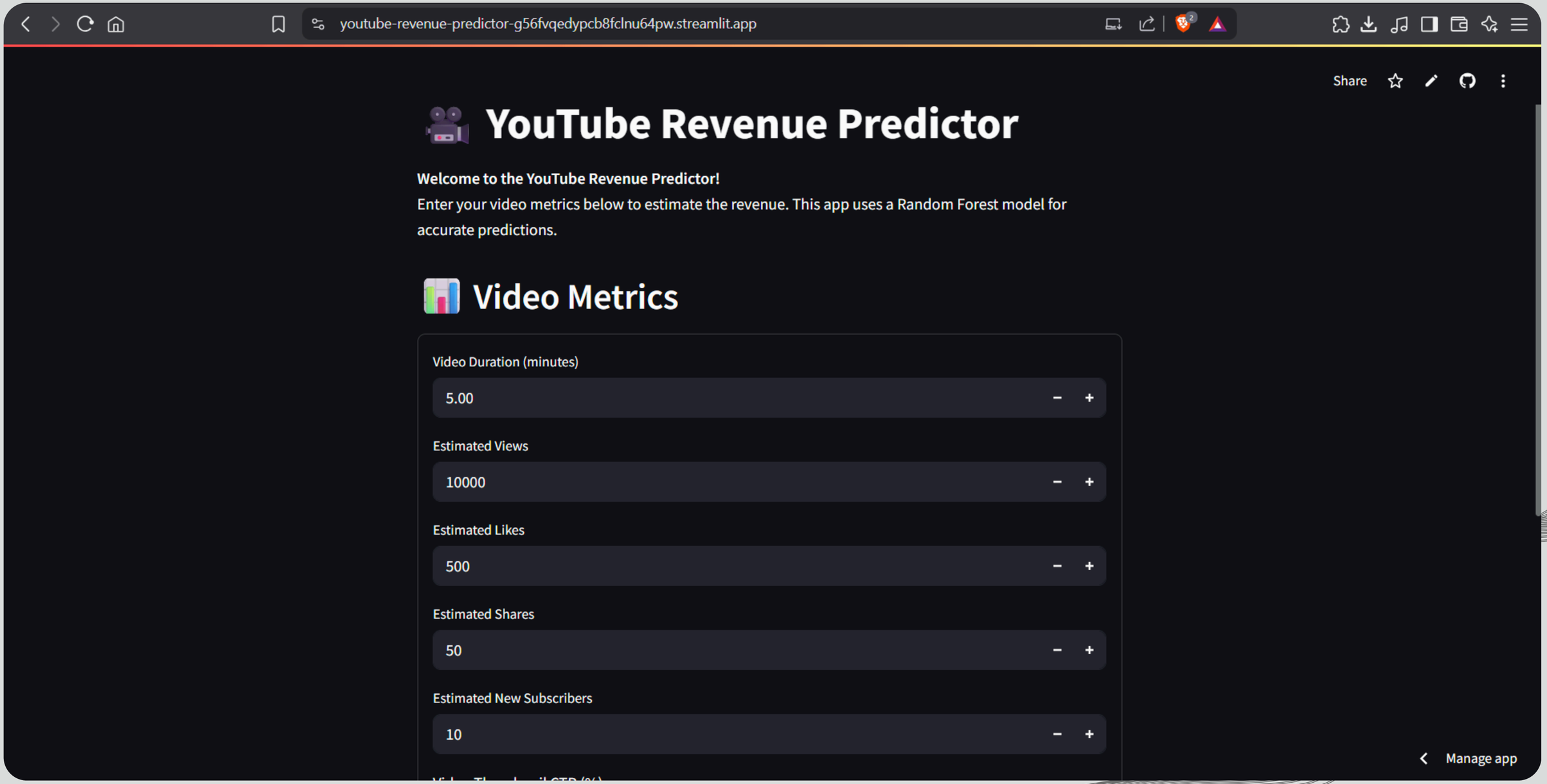
```
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='blue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.title('Actual vs. Predicted Revenue')
plt.xlabel('Actual Revenue (USD)')
plt.ylabel('Predicted Revenue (USD)')
plt.show()
```


Actual vs. Predicted Revenue
Predicted Revenue (USD)
Actual Revenue (USD)
0
10
20
30
40

Dashboard


9:15
94%
fclnu64pw.streamlit.app
Fork
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


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
Manage app


youtube-revenue-predictor-g56fvqedypcb8fclnu64pw.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
Manage app

youtube-revenue-predictor-g56fvqedypcb8fclnu64pw.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