In [208…
```
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, GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error, r2_score
```

In [209…
```
df = pd.read_csv('youtube_channel_real_performance_analytics.csv')
```

In [210…
```
print(df.head())
```

```
   ID  Video Duration   Video Publish Time  Days Since Publish  Day  Month  \
0   0           201.0  2016-06-02 00:00:00                   0    2      6
1   1           391.0  2016-06-10 00:00:00                   8   10      6
2   2           133.0  2016-06-14 00:00:00                   4   14      6
3   3            14.0  2016-06-29 00:00:00                  15   29      6
4   4            45.0  2016-07-01 00:00:00                   2    1      7

   Year Day of Week  Revenue per 1000 Views (USD)  \
0  2016    Thursday                         0.024
1  2016      Friday                         0.056
2  2016     Tuesday                         0.014
3  2016   Wednesday                         0.004
4  2016      Friday                         0.000

   Monetized Playbacks (Estimate)  ...  Watched (Not Skipped) (%)  \
0                           723.0  ...                        0.0
1                           727.0  ...                        0.0
2                            76.0  ...                        0.0
3                            18.0  ...                        0.0
4                             0.0  ...                        0.0

   Feed Impressions  Average View Percentage (%)  Average View Duration  \
0               0.0                        40.38                   81.0
1               0.0                        39.85                  156.0
2               0.0                        30.88                   41.0
3               0.0                       103.05                   14.0
4               0.0                        55.70                   25.0

     Views  Watch Time (hours)  Subscribers  Estimated Revenue (USD)  \
0  23531.0            533.1636         51.0                    0.561
1  11478.0            500.5628         33.0                    0.648
2   6153.0             70.7287          8.0                    0.089
3   4398.0             17.6251          2.0                    0.017
4  14659.0            104.3341         28.0                    0.000

   Impressions  Video Thumbnail CTR (%)
0      41118.0                    27.66
1      41627.0                     5.85
2      38713.0                     7.07
3      35245.0                     5.60
4      46218.0                     8.62

[5 rows x 70 columns]
```

In [211…

```
print(df.info())
```

```
<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
 16  Watch Page Ads Revenue (USD)          364 non-null    float64
 17  YouTube Premium (USD)                 364 non-null    float64
 18  Transaction Revenue (USD)             364 non-null    float64
 19  Transactions                          364 non-null    float64
 20  Revenue from Transactions (USD)       364 non-null    float64
 21  Reactions                             364 non-null    float64
 22  Chat Messages Count                   364 non-null    float64
 23  Reminders Set                         364 non-null    float64
 24  Stream Hours                          364 non-null    float64
 25  Remix Views                           364 non-null    float64
 26  Remix Count                           364 non-null    float64
 27  Subscribers from Posts                364 non-null    float64
 28  New Comments                          364 non-null    float64
 29  Shares                                364 non-null    float64
 30  Like Rate (%)                         364 non-null    float64
 31  Dislikes                              364 non-null    float64
 32  Likes                                 364 non-null    float64
 33  Unsubscribes                          364 non-null    float64
 34  New Subscribers                       364 non-null    float64
 35  Returned Items (USD)                  364 non-null    float64
 36  Unconfirmed Commissions (USD)         364 non-null    float64
 37  Approved Commissions (USD)            364 non-null    float64
 38  Orders                                364 non-null    float64
 39  Total Sales Volume (USD)              364 non-null    float64
 40  End Screen Click-Through Rate (%)     364 non-null    float64
 41  End Screen Impressions                364 non-null    float64
 42  End Screen Clicks                     364 non-null    float64
 43  Teaser Click-Through Rate (%)         364 non-null    float64
 44  Teaser Impressions                    364 non-null    float64
 45  Teaser Clicks                         364 non-null    float64
 46  Card Click-Through Rate (%)           364 non-null    float64
 47  Card Impressions                      364 non-null    float64
 48  Card Clicks                           364 non-null    float64
 49  Views per Playlist Start              364 non-null    float64
 50  Playlist Views                        364 non-null    float64
```

```
 51  Playlist Watch Time (hours)         364 non-null    float64
 52  Clip Watch Time (hours)             364 non-null    float64
 53  Clip Views                          364 non-null    float64
 54  YouTube Premium Watch Time (hours)  364 non-null    float64
 55  YouTube Premium Views               364 non-null    float64
 56  Returning Viewers                   364 non-null    float64
 57  New Viewers                         364 non-null    float64
 58  Average Views per User              364 non-null    float64
 59  Unique Viewers                      364 non-null    float64
 60  Watched (Not Skipped) (%)           364 non-null    float64
 61  Feed Impressions                    364 non-null    float64
 62  Average View Percentage (%)         364 non-null    float64
 63  Average View Duration               364 non-null    float64
 64  Views                               364 non-null    float64
 65  Watch Time (hours)                  364 non-null    float64
 66  Subscribers                         364 non-null    float64
 67  Estimated Revenue (USD)             364 non-null    float64
 68  Impressions                         364 non-null    float64
 69  Video Thumbnail CTR (%)             364 non-null    float64
dtypes: float64(63), int64(5), object(2)
memory usage: 199.2+ KB
None
```

In [212…
```
print(df.isnull().sum())
```

```
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
```

In [213…
```
df['Video Publish Time'] = pd.to_datetime(df['Video Publish Time'])
```

In [214…
```
le = LabelEncoder()
df['Day of Week'] = le.fit_transform(df['Day of Week'])
```

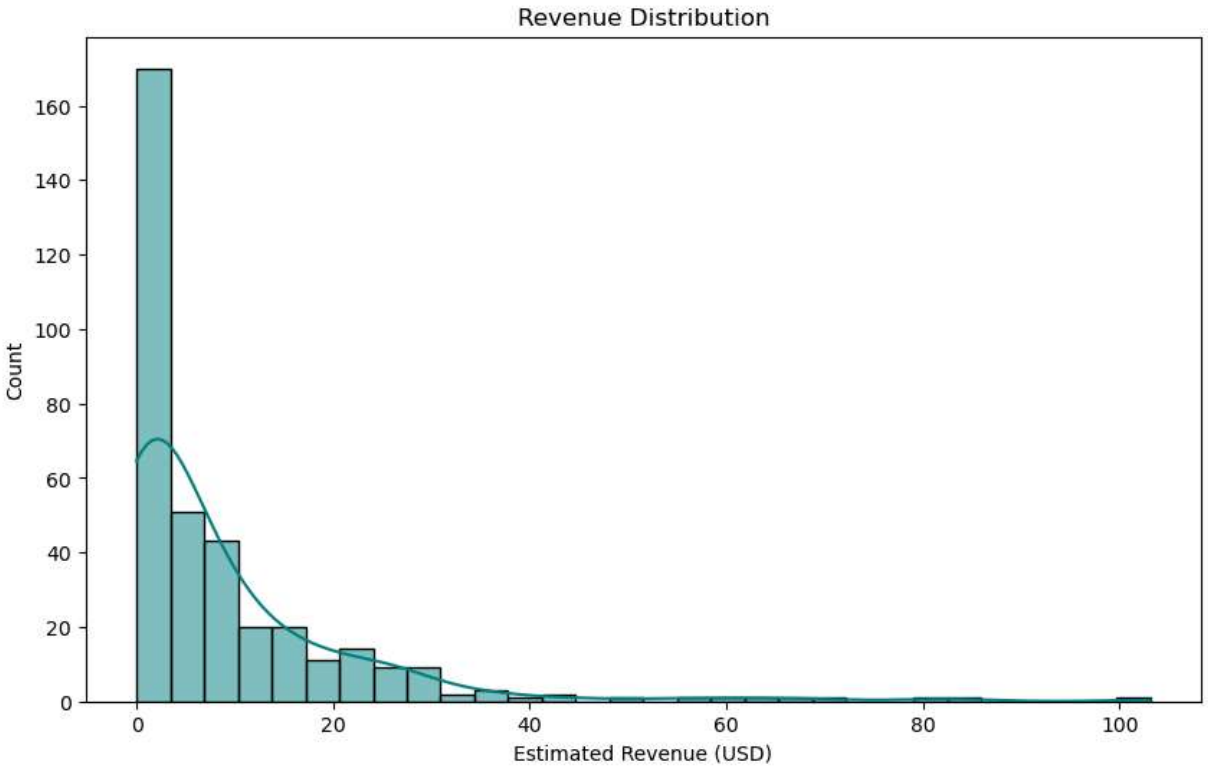
In [215…
```
# Feature Engineering
## Revenue per View
df['Revenue per View'] = df['Estimated Revenue (USD)'] / df['Views'].replace(0, np.

## Engagement Rate (%)
df['Engagement Rate (%)'] = ((df['Likes'] + df['Shares'] + df['New Comments']) / df

## Log Transform Revenue
df['Log Revenue'] = np.log1p(df['Estimated Revenue (USD)'])
```

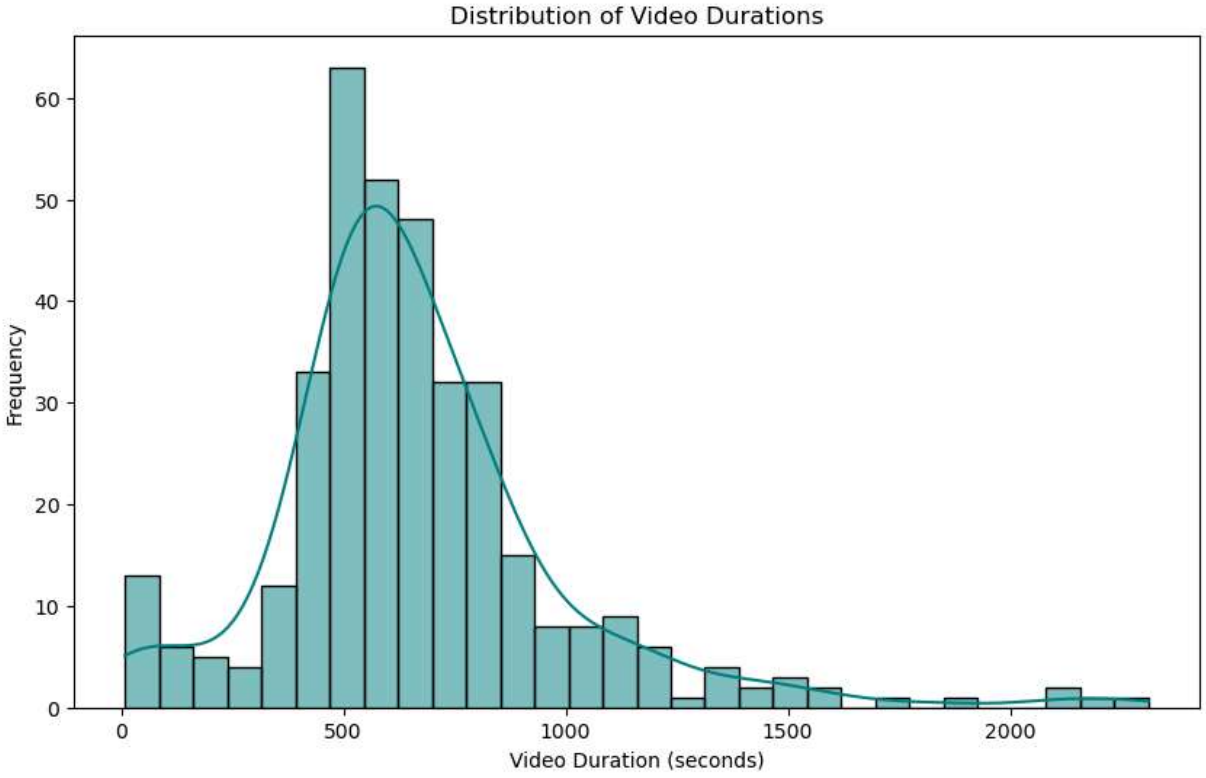
In [216…
```
# Exploratory Data Analysis
plt.figure(figsize=(10, 6))
sns.histplot(df['Estimated Revenue (USD)'], bins=30, kde=True, color='teal')
```

```
plt.title('Revenue Distribution')
plt.show()
```



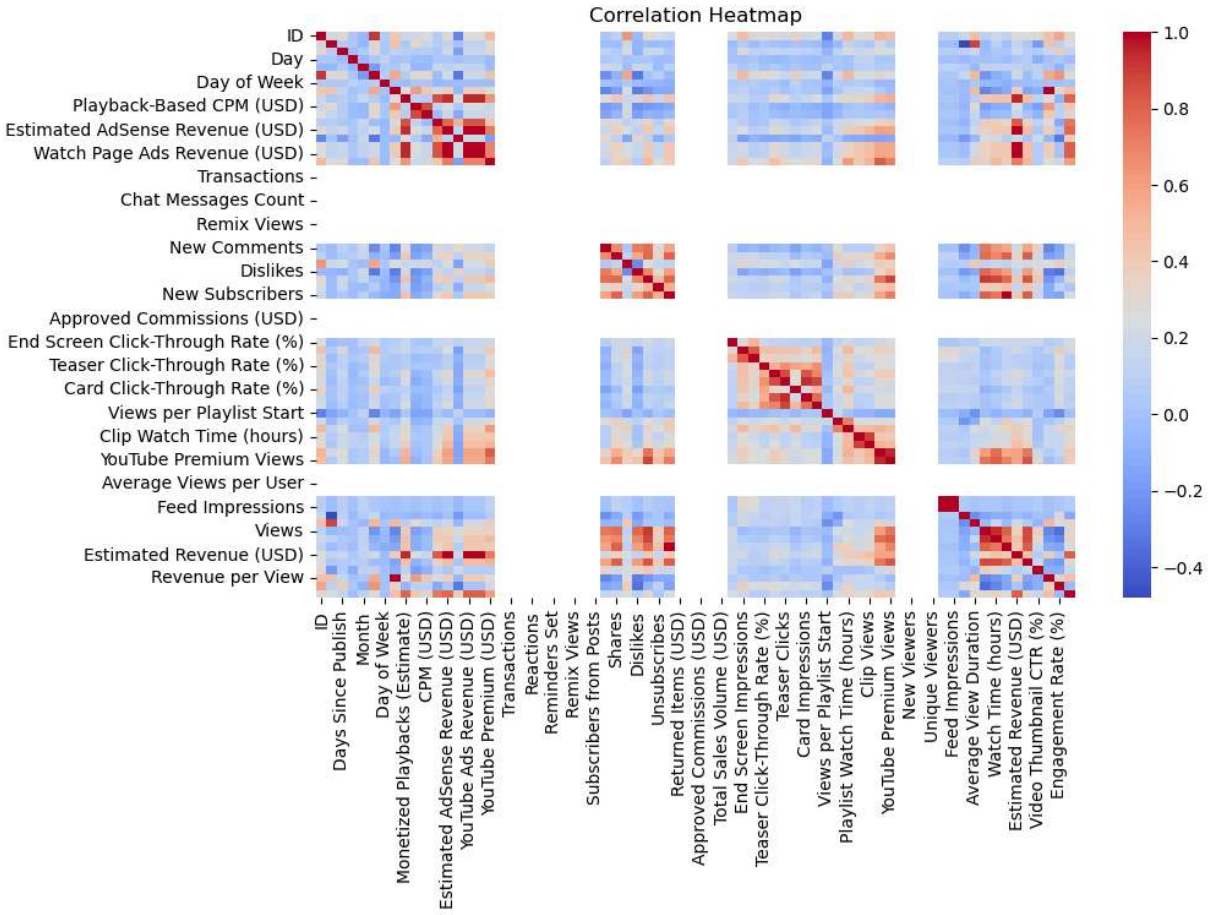
In [217…

```
# Distribution of video durations
plt.figure(figsize=(10, 6))
sns.histplot(df['Video Duration'], bins=30, kde=True, color = 'teal')
plt.title('Distribution of Video Durations')
plt.xlabel('Video Duration (seconds)')
plt.ylabel('Frequency')
plt.show()
```



In [218…

```
plt.figure(figsize=(10, 6))
sns.heatmap(df.select_dtypes(include=np.number).corr(), cmap='coolwarm', annot=Fals
plt.title('Correlation Heatmap')
plt.show()
```



In [219…
```
# Prepare Data for Modeling
feature_cols = ['Views', 'Subscribers', 'Likes', 'Shares', 'New Comments', 'Engagem
X = df[feature_cols].fillna(0)
y = df['Log Revenue']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [220…
```
# Model Training with Hyperparameter Tuning
param_grid = {
    'n_estimators': [100, 200],
    'max_depth': [None, 10, 20]
}
rf = RandomForestRegressor(random_state=42)
grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=3, scoring='r2')
grid_search.fit(X_train, y_train)
best_model = grid_search.best_estimator_
```

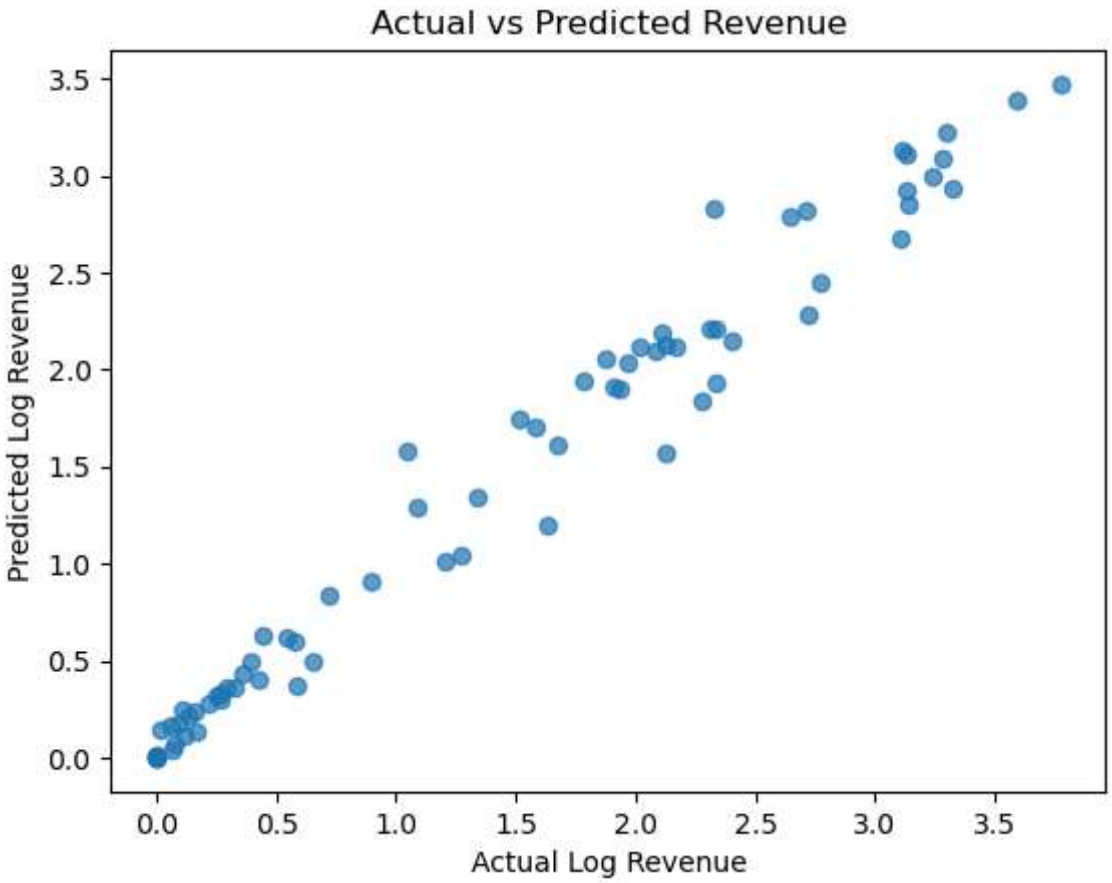
In [221…
```
# Model Evaluation
y_pred = best_model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f'Best Model Parameters: {grid_search.best_params_}')
print(f'Mean Squared Error: {mse:.4f}')
print(f'R2 Score: {r2:.4f}')
```

```
Best Model Parameters: {'max_depth': None, 'n_estimators': 200}
Mean Squared Error: 0.0411
R2 Score: 0.9700
```

In [222…
```
# Feature Importance

importances = best_model.feature_importances_
feat_importance_df = pd.DataFrame({'Feature': feature_cols, 'Importance': importanc
```

In [223…
```
plt.scatter(y_test, y_pred, alpha=0.7)
plt.xlabel('Actual Log Revenue')
plt.ylabel('Predicted Log Revenue')
plt.title('Actual vs Predicted Revenue')
plt.show()
```



In [224…
```
from sklearn.model_selection import cross_val_score
cv_scores = cross_val_score(best_model, X, y, cv=5, scoring='r2')
print('Cross-Validation R2 Scores:', cv_scores)
print('Average CV R2:', np.mean(cv_scores))
```

```
Cross-Validation R2 Scores: [0.69451047 0.91442931 0.96238737 0.93948684 0.91287163]
Average CV R2: 0.8847371220023257
```

In [225…
```
X = numeric_df.drop(columns=['Estimated Revenue (USD)'])
y = numeric_df['Estimated Revenue (USD)']
```

In [226…
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [227…
```
# Initialize and train the model
model = RandomForestRegressor(n_estimators=100,
random_state=42)
model.fit(X_train, y_train)
```

Out[227…
```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```

In [228…
```
# Make predictions
y_pred = model.predict(X_test)
```

In [229…
```
# Calculate the prediction accuracy
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
rmse
```

Out[229…
```
0.4828740106603764
```