Content Monetization Modeler

Project Overview

The Content Monetization Modeler is a friendly Streamlit web app that can be used to predicts YouTube revenue based on user inputs such as views, likes,comments, watch time,subscribers, vedio length, type of vedio category, country. It is to estimate earnings and understand how different factors influence monetization.

We started with basic input data such as:

- views
- likes
- comments
- watch_time_minutes
- video_length_minutes
- subscribers
- category
- device
- country

To improve the model, we added a few extra features:

- Views_Outlier_Flag → to mark extreme cases like very low or very high views
- engagement_rate → calculated from likes, comments, and views
- watch_per_view → calculated from watch time divided by views

We also used **one-hot encoding** to convert category, device, and country into numeric columns so the model could understand them better.

These extra features helped the model learn more patterns and make better predictions.

Since our goal was to predict a **numeric value** (YouTube revenue), we used different types of **regression models** during training. These included:

- **Linear Regression**

- **Ridge Regression**
- **Decision Tree Regressor**
- **Random Forest Regressor**
- **XGBoost Regressor**

We trained and tested each model to see which one gave the good results. To measure their performance, we used three common evaluation metrics:

- **R^2 Score** (how well the model fits the data)
- **RMSE** (Root Mean Squared Error — shows average prediction error)
- **MAE** (Mean Absolute Error — shows how far predictions are from actual values)

We evaluated for train result and test result as well

We found some outliers in the dataset using IQR statistical analysis. In the Streamlit app, we introduced a basic outlier check using simple business rules — for example, if views are less than 100 or more than 500,000. This helps the app accept inputs that match how the model was trained, since the model already includes an outlier flag feature. This way, the prediction stays realistic and consistent with the training logic.

Installation Instructions

To run the app locally:

```
pip install streamlit pandas numpy matplotlib seaborn scikit-learn
```

```
streamlit run app.py
```

Insights and Observations

One of the most observations is the strong influence of view count and watch time on ad revenue. These two features consistently demonstrated high correlation with the target variable, confirming their importance in monetization prediction.

Future Improvements

We can focus to analyse the prediction to support other content platforms like **Instagram Reels**, **TikTok**, **Telegram Channels**, and **Facebook Video**. This would allow creators to estimate revenue across multiple ecosystems, not just YouTube, because nowadays vedioes views happens in different platform as well