YouTube Shorts Performance Prediction: Approach Document

This document provides a structured approach for navigating the **YouTube Shorts Performance Prediction Case Study**. The approach is designed to guide the process from data understanding and feature engineering through to model training, evaluation, and the extraction of critical business insights.

Case Study Overview

The core challenge of this case study is to leverage **Supervised Machine Learning** to predict the potential performance (specifically, the Engagement Rate tertile: Low, Medium, or High) of a YouTube Short based on its intrinsic features (title, duration, category) and its publishing behavior (upload hour). The final goal is to develop a reliable predictive model and deliver **actionable content strategy recommendations** to maximize viral potential and channel growth.

Datasets

You will be primarily working with one integrated dataset:

- **Shorts_Performance.csv**: Contains video metadata and engagement metrics.

General Approach & Data Preparation

Before training any model, a thorough understanding and preparation of the data is essential, with a specific focus on transforming raw features into model-ready numerical data.

Data Loading & Cleaning

- **Load and Inspect Data:** Load the provided CSV file and inspect its shape, data types (dtypes), and look for missing values.
- **Standardize Columns:** Ensure consistency in column names (e.g., snake_case).
- **Missing Values:** Address any missing values, potentially using mean imputation for numerical columns or mode imputation for categorical columns, if necessary.
- Target Creation (Critical Step): Follow the precise formula provided:

  Engagement_Rate = (likes + comments + shares) / views

  - Hint: Calculate the Engagement Rate for all videos. Then, use the quantile function (0.33 and 0.66) to establish the thresholds and create the 3-class target column, performance_engagement_tertile (Low, Medium, High).

Feature Engineering & Transformation

- **Derived Textual Features:** Create the required features: title_len_chars, title_word_count, and the boolean feature title_has_question_mark.
- **Rate Features:** Calculate per-second engagement rates: likes_per_sec, comments_per_sec, and shares_per_sec.
- **Logarithmic Transformation:** Apply log transformation to heavily skewed features like views, likes, comments, and shares (e.g., log_views) to meet model assumptions and reduce the impact of outliers.
- **Time-Based Feature:** Create the binary feature is_peak_hour based on a commonly observed evening window (e.g., hours 17-21, or as defined by EDA).

Exploratory Data Analysis (EDA) & Feature Insights

EDA is crucial for understanding the data distribution and for informing feature engineering choices.

Analysis/Plot	**Purpose & Hint**
Engagement Rate Histogram	Visualize the distribution of the newly created target variable's underlying metric. This visually confirms the need for the tertile splits.
Boxplot of Engagement Rate vs. Category	Determine which content categories inherently drive higher or lower engagement. **Hint:** Use the performance_engagement_tertile for visualization against category.
Correlation Heatmap	Identify multicollinearity among numerical features and the correlation strength between each feature and the underlying engagement_rate. **Hint:** Focus on correlation between duration_sec, hashtags_count, and the engagement metrics.
Upload Hour vs. Average Engagement Rate	Identify optimal posting times. **Hint:** Group data by upload_hour (0-23) and plot the mean engagement_rate to look for peaks.

Scatter Plot: Duration vs. Engagement Rate	Assess if there's an optimal video length for maximizing performance. **Hint:** Look for a non-linear relationship or clusters.

ML Approach & Model Training

This phase focuses on building robust, comparable models using structured pipelines.

Data Split & Pipelines

- **Train/Test Split:** Perform a **Stratified 80/20 split** using the performance_engagement_tertile to ensure the target classes are equally represented in both sets.
- **Pipelines:** Use sklearn.compose.ColumnTransformer to manage different preprocessing steps for different feature types (numerical vs. categorical).
  - **Numerical Features:** Apply standard or min-max **scaling**.
  - **Categorical Features:** Apply **One-Hot Encoding** (especially to category and potentially upload_hour).

Model Training & Evaluation

- **Model Selection:** Train the mandatory classification models (Logistic Regression, Random Forest, XGBoost/LightGBM, KNN, SVM).
- Cross-Validation: Utilize K-fold Cross-Validation (e.g., k=5 or k=10) for robust training and reporting of mean (±) standard deviation for Accuracy and F1-macro.
  - **Hint:** F1-macro is essential because it treats all three classes (Low, Medium, High) equally, providing a better measure of performance than simple accuracy on an imbalanced problem.

Hyperparameter Tuning

- **Tuning:** Use **GridSearch** or **RandomizedSearch** on at least two of the specified models to optimize performance.

Final Evaluation

- **Test Set Metrics:** Evaluate the best-performing models on the held-out **Test Set**, providing a comprehensive comparison table including: Confusion Matrix, Classification Report, Accuracy, F1-macro, and **ROC-AUC (One-vs-Rest)**.

Task 9 & 10: Model Explainability & Business Insights

The final and most crucial step is translating model results back into business value.

Task	**Explanation & Business Focus**
Model Explainability	**Feature Importance:** Analyze **Feature Importances** from tree-based models (Random Forest, XGBoost) to identify which inputs (e.g., duration_sec, hashtags_count, or a specific category) the model relies on most. **Coefficient Analysis** for Logistic Regression serves a similar purpose.
Business Insights Summary	**Actionable Advice:** Based on Feature Importances and EDA plots, provide a concise summary that answers the core business questions: *What is the optimal video duration?*, *What are the best categories to focus on?*, and *What is the ideal upload time?* This advice must be directly supported by your model and analysis findings.