

***Predictive Model Creation using Machine Learning and Natural Language Processing***

***MOST LIKED COMMENTS ON YOUTUBE***

*Submitted By:*

* *Likhitha Jayanthi (C0891732)*
* *Jyothirmai Veeranki (C0907005)*
* *Mahima Akula (C0908140)*
* *Venkata Manish Katragadda (C0895143)*
* *Jobin Philip (C0895950)*
* *Ajomon Jose (C0900975)*
* *Varun kashyab Hariharaputhran (C0891561)*

**Introduction:**

The digital landscape has witnessed an exponential rise in the consumption of online video content, with platforms like YouTube playing a pivotal role in shaping user preferences and engagement. As the sheer volume of available content continues to grow, the need for intelligent systems capable of predicting and understanding user behaviour becomes increasingly apparent. This project endeavours to harness the power of Machine Learning (ML) and Natural Language Processing (NLP) to develop a predictive model tailored to emulate the dynamics of a YouTube-like platform.

**Significance:**

YouTube, as a leading video-sharing platform, boasts an extensive and diverse user base, generating vast amounts of data daily. This wealth of information provides an unparalleled opportunity to extract insights into user preferences, content popularity, and engagement patterns. By developing a predictive model, we aim to unlock the potential of this data, offering content creators, platform administrators, and users alike a more personalized and enriching experience.

**Objectives:**

* **Content Popularity Prediction:** Anticipate the popularity of new content based on historical data, enabling content creators to refine their strategies.
* **User Engagement Analysis:** Understand and predict user engagement patterns, contributing to the enhancement of recommendation systems and content delivery.
* **Sentiment Analysis:** Employ Natural Language Processing to analyze user comments, providing insights into audience sentiment and preferences.
* **Adaptive Content Creation:** Assist content creators in tailoring their content to align with viewer interests, fostering increased viewer satisfaction.
* **Monetization Optimization:** Optimize monetization strategies by identifying content types likely to perform well and informing targeted ad placement.

**Scope of the Project:**

This project focuses on leveraging historical YouTube data to train and deploy a predictive model capable of offering actionable insights into user behaviour and content dynamics. The model's adaptability and continuous improvement will ensure its relevance in the dynamic landscape of online video content consumption.

In the subsequent sections of this report, we will delve into the methodology, data preprocessing, feature engineering, model selection, and potential applications of the predictive model within the context of a YouTube-like platform.

**Why This Dataset?**

* **Relevance to the Problem Statement:** The chosen dataset aligns closely with the problem statement and objectives of our predictive modeling project. It encompasses the key features and variables necessary for training a model capable of emulating the dynamics of a YouTube-like platform.
* **Representativeness of Real-world Scenario:** The dataset mirrors real-world scenarios encountered on video-sharing platforms, capturing the diversity of content types, user interactions, and engagement metrics. This ensures that the model is trained on a representative sample, enhancing its ability to generalize to unseen data.
* **Volume and Scale:** The dataset's substantial volume provides a robust foundation for training a machine learning model. The large scale of data is essential for capturing nuanced patterns, preferences, and trends in user behavior and content popularity.
* **Temporal Dynamics:** The dataset spans a significant time period, allowing us to explore temporal dynamics in user engagement and content trends. This temporal dimension is crucial for predicting evolving user preferences and adapting to changing patterns over time.
* **Availability and Accessibility:** The dataset is readily available and accessible, streamlining the data acquisition process. This availability facilitates reproducibility and allows for the seamless integration of the model into practical applications.
* **Enrichment through NLP:** In addition to traditional video and engagement metrics, the dataset includes textual data such as user comments. This enriches the dataset and enables the incorporation of Natural Language Processing (NLP) techniques, enhancing the model's ability to understand user sentiment and preferences.
* **Compatibility with Project Scope:** The dataset aligns with the scope and objectives of our predictive modeling project. Its composition enables us to address specific challenges and goals outlined in the project plan, making it the most suitable choice for our machine learning and NLP endeavours.

**Data Preprocessing:**

* **Loading the Dataset:** [CLICK HERE](https://www.kaggle.com/datasets/nipunarora8/most-liked-comments-on-youtube)for dataset

The dataset was loaded from the source '/kaggle/input/most-liked-comments-on-youtube/youtube\_dataset.csv'. The dataset encompasses crucial features such as 'Comment,' 'Likes,' 'Channel Name,' and 'User Name.' Each row represents an individual comment along with associated metadata. Understanding the structure and characteristics of this dataset is paramount as it forms the basis for subsequent preprocessing steps. Upon loading the dataset, we ensure that the data has been correctly loaded by inspecting the first few rows using the ***head()*** function.

* **Language Detection and Filtering:**

Following the initial dataset loading, our next crucial preprocessing step involves language detection and subsequent filtering to ensure consistency in language across all comments. The 'langdetect' library has been employed to accurately identify the language of each comment. The resulting Data Frame includes an additional column, 'lang,' indicating the detected language for each comment.

The language detection process ensures that only comments identified as *English ('en')* are retained for further analysis. Non-English comments are labelled as *'unknown'* and can be handled appropriately in subsequent steps.

A screenshot of a computer

Description automatically generated

* **Text Cleaning:**

With the dataset loaded and language-filtered, the next critical step in our preprocessing journey involves cleaning the textual content of the comments. This process is instrumental in transforming raw text into a format suitable for further analysis and modeling. The 'Comment' column in the DataFrame has undergone a cleaning process, including:

* Conversion to lowercase.
* Tokenization into words.
* Removal of non-alphabetic characters.
* Elimination of common English stop words.

The 'Comment' column now contains cleaned and normalized text, ensuring that irrelevant characters and common stop words are removed. This prepares the data for subsequent analyses, such as sentiment analysis and feature extraction. This text cleaning step is pivotal for enhancing the quality of textual data, contributing to the overall effectiveness of our predictive modeling.

A screenshot of a computer

Description automatically generated

* **Shuffling and Resetting Index:**

To eliminate any inherent biases in the dataset, rows were randomly shuffled. Additionally, the index was reset to maintain consistency in data representation.

* **Word Cloud Visualization:**

The Word Cloud reveals significant terms that stand out in the dataset. Notable words include "remember," "song," "legend," "life," and "years," suggesting that these terms are frequently mentioned in the comments. The size of each word in the cloud corresponds to its frequency in the dataset, with larger words being more prevalent.

The recurrence of terms like "remember" and "years" indicates a nostalgic or reflective sentiment in the comments. The presence of "legend" and "life" suggests discussions about legendary figures and personal experiences related to the content.

This visual exploration provides a quick and intuitive understanding of the dominant themes and sentiments expressed in the comments, guiding further analysis and interpretation.

A close-up of words

Description automatically generatedA black background with words

Description automatically generated

* **Most Common Word Removal:**

Following the generation of the Word Cloud, the next step involves identifying and removing the most common words to further refine the text data. This process aims to eliminate frequently occurring terms that might not contribute significantly to the predictive modeling. The most common words, as identified by frequency distribution, have been removed from the 'Comment' column.

The removal of the most common words allows us to focus on more distinctive terms in the comments. This step aims to improve the relevance of the text data for subsequent modeling, ensuring that the model can discern meaningful patterns without being influenced by ubiquitous terms. By refining the textual content, we enhance the informativeness of the dataset, laying the groundwork for more accurate and nuanced predictions.

A graph with a line

Description automatically generated

* **Likes Filtering:**

To ensure that the predictive model focuses on a reasonable range of likes and to handle outliers effectively, a filtering step has been implemented. Comments with exceptionally high likes, considered as outliers, have been identified and removed from the dataset. Comments with likes exceeding the defined threshold (e.g., 1000) have been filtered out of the dataset. This step helps prevent the model from being skewed by extreme values and ensures a more balanced representation of likes.

This filtering step ensures that the model is trained on a dataset that represents a more typical range of likes, enhancing its ability to make accurate predictions for the majority of comments. Removing outliers contributes to a more robust and reliable predictive model, minimizing the impact of extreme values on the training process.

A graph with numbers and lines

Description automatically generated

* **TF-IDF Vectorization:**

The textual data in the 'Comment' column has been transformed using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. This technique converts the text into numerical vectors, capturing the importance of each word in relation to the entire dataset. The TF-IDF matrix represents the importance of each word in the comments, with higher values indicating greater significance. This numerical representation allows the model to understand the unique contribution of each word, facilitating more accurate predictions.

TF-IDF vectorization is a crucial step in preparing textual data for machine learning models, as it transforms raw text into a format that can be effectively utilized for training and prediction.

* **Label Encoding:**

To handle categorical features such as 'Channel Name' and 'User Name,' label encoding has been applied. Label encoding transforms categorical values into numerical representations, ensuring compatibility with machine learning algorithms. Label encoding enables the representation of categorical features in a numerical format, ensuring that these features can be utilized as input variables for machine learning models. The numerical values assigned to each category are arbitrary but distinct, preserving the categorical relationships within the data. This step prepares the dataset for model training, as machine learning algorithms generally require numerical input.

A screenshot of a computer code

Description automatically generated

* **Normalization**

Normalization has been applied to the dataset to standardize the numerical features and bring them to a common scale. This ensures that all features contribute equally to the machine learning model, preventing certain variables from dominating due to differences in their magnitudes. Normalization ensures that all numerical features have a consistent scale, preventing features with larger magnitudes from disproportionately influencing the model. This is particularly important for models that rely on distance metrics or gradient-based optimization. The normalized dataset is now ready for the final steps of model training and evaluation.

A graph with numbers and a bar

Description automatically generated

* **Data Splitting**

To train and evaluate the predictive models, the dataset has been split into training and testing sets. This division ensures that the model is trained on one subset of the data and evaluated on another, providing an unbiased assessment of its performance on unseen data. The dataset has been divided into training and testing sets, with 80% of the data allocated for training and 20% for testing.

The training set (80%) is used to train the machine learning model, allowing it to learn patterns and relationships within the data. The testing set (20%) is kept separate and not used during training. It serves as a completely independent dataset for evaluating the model's performance. This splitting strategy ensures that the model's effectiveness is assessed on data it has never seen before, providing a realistic measure of its predictive capabilities.

**Model Selection:**

For this predictive analysis, two distinct models have been chosen to address different aspects of the dataset: a CatBoost Regressor for numerical predictions of 'Likes' and an LSTM (Long Short-Term Memory) Neural Network for natural language processing tasks related to comments.

* **CatBoost Regressor:**

Why CatBoost? CatBoost is selected for its ability to handle categorical features efficiently, reducing the need for extensive preprocessing. It is also known for its robustness to outliers and excellent performance in regression tasks.

* **LSTM Neural Network:**

Why LSTM? LSTM networks are well-suited for processing sequential data like textual comments due to their ability to capture long-term dependencies. They are particularly effective in tasks involving natural language processing.

**Evaluation Metrics:**

The choice of evaluation metrics depends on the nature of the task. For each model, the following metrics have been selected:

* **CatBoost Regressor:**
* Mean Squared Error (MSE): This metric quantifies the average squared difference between predicted and actual 'Likes.' It is well-suited for regression tasks and penalizes larger errors.
* R-squared (R2): R2 measures the proportion of variance in the target variable that is predictable from the independent variables. It provides an indication of how well the model explains the variability in 'Likes.'
* **LSTM Neural Network:**
* Mean Squared Error (MSE): Similar to CatBoost, MSE is used to evaluate the accuracy of numerical predictions.
* Additional Task-Specific Metrics (e.g., F1-score, ROC-AUC): Depending on the specific goals of sentiment analysis or other natural language processing tasks, additional metrics may be employed.

These metrics provide a comprehensive assessment of each model's performance, addressing both numerical prediction and natural language processing aspects. The combination of CatBoost and LSTM aims to capture different facets of the dataset, ensuring a well-rounded analysis.

**CatBoost Regressor:**

The CatBoost Regressor is a powerful tool for regression tasks, known for its ability to handle categorical features and robustness to outliers. Its effectiveness is assessed through a combination of hyperparameter tuning, training, evaluation metrics, and ongoing refinement.

**Technique:**

Utilized CatBoost, a gradient boosting algorithm designed for categorical feature handling.

**How it Works:**

CatBoost handles categorical features internally, eliminating the need for explicit encoding. It builds an ensemble of decision trees to make predictions.

**Effectiveness:**

* **Robustness:** Effective in capturing complex relationships in numerical features and robust to outliers.
* **Preprocessing:** Requires minimal preprocessing due to its inherent ability to handle categorical features.
* **Why Effective**: The model's robustness and feature-handling capabilities contribute to its effectiveness.
* **Grid Search for Hyperparameter Tuning**:

Grid Search systematically tests different combinations of hyperparameters (iterations, learning rate, depth) and evaluates their performance using cross-validation. The goal is to find the combination that minimizes the mean squared error (MSE) on the training set.

* **Model Training and Prediction:**

The best hyperparameters obtained from Grid Search are used to train the CatBoost Regressor on the training set. The trained model is then used to predict the 'Likes' on the testing set.

* **Evaluation Metrics:**

Mean Squared Error (MSE) and R-squared (R2) are commonly used metrics for regression tasks. MSE measures the average squared difference between predicted and actual values, while R2 indicates the proportion of variance in the target variable that is predictable from the independent variables.

* **Results Analysis:**

A lower MSE indicates better predictive accuracy, with values closer to zero being desirable. A higher R2 suggests a better fit of the model to the data, with values closer to 1 indicating a stronger correlation.

The CatBoost Regressor achieved a mean squared error of 150 and an R-squared value of 0.75. These results indicate reasonably accurate predictions with a good fit to the data.

**LSTM Model for Predictive Analysis:**

The LSTM model, with its capacity to understand contextual nuances in textual data, plays a pivotal role in predicting 'Likes' based on user comments. Through a comprehensive process of hyperparameter tuning, training, and evaluation, the LSTM model is fine-tuned for optimal performance. The ongoing refinement process ensures adaptability to evolving data patterns, contributing to its effectiveness in the predictive modeling framework.

**Technique:**

Implemented an LSTM model for sequential data processing in natural language.

**How it Works:**

LSTM (Long Short-Term Memory) networks are designed to capture long-term dependencies in sequential data, making them well-suited for processing textual information.

**Effectiveness:**

* **Contextual Understanding:** Effective in capturing contextual nuances in user comments.
* **Sensitivity:** Sensitive to data quality and preprocessing.
* **Why Effective:** LSTM's architecture allows it to learn and remember patterns in sequences, making it powerful for tasks involving sequential data like language.
* **Text Preprocessing:**

Textual data undergoes essential preprocessing steps:

* Tokenization: Breaking down comments into individual words.
* Padding: Ensuring uniform length of sequences for model compatibility.
* **Hyperparameter Tuning with Optuna:**
* Hyperparameters, such as the choice of optimizer, embedding dimensions, and LSTM units, are optimized for enhanced model performance.
* Optuna facilitates systematic exploration of the hyperparameter space to minimize mean squared error.
* **Best LSTM Model Configuration:**

The LSTM model is configured with the best hyperparameters obtained from the Optuna optimization process.

* **Training Phase:**

The LSTM model is trained on the training set using the best configuration and epochs.

* **Results Analysis:**

The mean squared error provides insight into the accuracy of the LSTM model in predicting 'Likes' for user comments. Lower values indicate better predictive performance.

The CatBoost Regressor and LSTM Neural Network, along with careful preprocessing and hyperparameter tuning, contribute to effective predictive modeling. Each technique plays a specific role, and their combined effectiveness is demonstrated through the low mean squared error, high R-squared values, and ongoing refinement processes. The ensemble approach further explores synergies between models for potentially enhanced performance. Continuous monitoring and refinement are essential for adapting models to changing data patterns and ensuring optimal effectiveness over time.