Instagram User Category and Like Count Prediction

CS 412 Term Project

Selim Sıdan 30992

Saner Bilir 30555

Fırat Yurdakul 29170

Mert Polat 31326

Table of Contents

1. Overview

This repository contains the implementation and analysis of a machine learning project with 2 notebooks each containing a different model, classification notebook for predicting the Instagram users' categories and regression notebook for predicting the like count of posts.

a. Classification

The main objective is to combine textual features from Instagram post captions with user-level profile attributes to build a robust classification model. Key sections of the project include:

- **Data Preparation**: Preprocessing captions, creating training corpora, and cleaning user-level features.
- **Feature Engineering**: Implementation of TF-IDF vectorization with enhanced configurations and integration of user-level features like follower counts and engagement metrics.
- **Model Training and Optimization**: Training a Logistic Regression model with L1 regularization for feature selection and performing hyperparameter tuning via grid search.
- **Evaluation and Results**: Assessing model performance using metrics such as accuracy, precision, recall, and classification reports.
- **Baseline Comparison**: Demonstrating significant improvements over the provided baseline Naive Bayes model, which relied solely on TF-IDF features.

Key Files:

- **classification.ipynb**: The classification notebook containing all steps for preprocessing, feature engineering, model training, and evaluation.

b. Like Prediction

The regression component focuses on predicting Instagram post like counts by leveraging temporal patterns, historical engagement, and content characteristics. Key sections include:

Feature Engineering: Development of sophisticated time-weighted engagement metrics, temporal features, and content-based indicators.

Historical Processing: Implementation of exponential decay weighting for historical engagement with a 15-day half-life, capturing recency effects in user interactions.

Model Architecture: XGBoost regression with optimized hyperparameters, including careful tuning of estimators, learning rate, and tree depth parameters.

Evaluation and Results: Evaluation using Log Mean Squared Error metrics on both training and test sets, with extensive feature importance analysis.

Key Files:

regression.ipynb: The regression notebook implementing time-weighted engagement predictions, feature engineering, and model optimization.

2. Methodology

a. Classification

The classification part's methodology consists of 4 key stages:

1. **Text Preprocessing**

Most of the preprocessing steps were adapted from a provided notebook, minor customizations were introduced to better fit the data. This involved removing unnecessary characters, normalizing text, and creating a unified training corpus. Each user's captions were combined into a single document for TF-IDF representation, ensuring alignment with the username-label mapping.

2**. TF-IDF Vectorization**

Textual features are extracted via Term Frequency-Inverse Document Frequency (TF-IDF) vectorization, which encodes the corpus into a high-dimensional sparse matrix. This process involves defining thresholds (min_df=2 and max_df=0.9) to exclude overly frequent or rare terms, ensuring the robustness of the feature space. Additionally, user profile features are preprocessed to handle missing values through imputation strategies (e.g., mode-based for categorical features, constant values for numerical features). Boolean fields are encoded as binary indicators, while categorical features are label-encoded for compatibility with machine learning models. Numerical attributes are standardized using StandardScaler to maintain uniform feature scaling.

3. **User-Level Feature Engineering**

User-level features, such as follower count and engagement metrics, were extracted using the provided notebook. The novel contributions in this step included handling missing values through imputation (e.g., replacing missing numeric values with median or mean), encoding categorical variables, and performing feature selection to identify the most informative user attributes while eliminating unnecessary ones. These processed features were then combined with the TF-IDF features into a unified dataset, capturing both textual and user-level features.

4. **Model Training and Optimization**

Building upon the baseline Naive Bayes classifier, which relied solely on TF-IDF features, the enhanced approach employed a Logistic Regression model with L1-regularization. Designed to address the 10-class classification problem, the model adopted the One-vs-All (OvA) strategy, wherein a separate binary classifier was trained for each class against all others. This approach allowed efficient handling of multi-class classification while leveraging L1-regularization to mitigate overfitting and perform automatic feature selection by shrinking less informative features to zero.

Hyperparameter tuning was systematically conducted using GridSearchCV, optimizing critical parameters such as regularization strength (C) and class weighting. Robustness was further enhanced by employing cross-validated feature selection with SelectFromModel, ensuring that the model consistently identified and retained the most predictive features across different validation splits. By leveraging both textual and user-level features, the final Logistic Regression model demonstrated superior generalizability and predictive accuracy, significantly outperforming the baseline Naive Bayes classifier.

b. Like Prediction (Regression)

The regression methodology mainly comprises four steps:

1. **Temporal Feature Engineering and Preprocessing**
   Temporal features regarding the Instagram posts were processed to capture engagement patterns: the timestamp of the post, hour of the day, day of the week, and month. Although a number of basic temporal features could be found in the provided dataset, some extended customizations were made to highlight nuanced temporal patterns. Additionally, data cleaning included identifying missing values with respect to temporal information and standardizing datetime presentation across all rows.

2. **Historical Engagement Processing**
   A significant part of processing historical engagement data involved implementing time-weighted metrics. This new metric uses an exponential decay function with a 7-day half-life, where higher weights are given to recent interactions, while still maintaining the relevance of older posts. Preprocessing in this context aimed to calculate engagement statistics such as the average number of likes, standard deviation of likes, and rolling average of the last 5 posts. Missing values in historical data were handled through forward-filling or zero-imputation strategies, ensuring robust feature computation even when engagement histories were incomplete.

3. Content-Based Feature Engineering

Features were extracted from both post-level and user-level data. Media types were binary encoded as video, image, and carousel. User-level features, such as verification status and number of followers, were preprocessed to handle missing values and normalized where necessary. Novel contributions in this step included creating composite features that combined multiple aspects of content characteristics, enabling the model to capture more complex patterns in user engagement.

4. Model Architecture and Training

The approach incorporated regression methods using XGBoost Regression, with hyperparameter tuning. In designing the model architecture, the nonlinear nature of social media engagement was considered, which led to the use of gradient boosting and tree-based learning. The target variable underwent log transformation (log(like_count + 1)) since social media metrics usually follow a right-skewed distribution. Hyperparameter tuning focused on the most important parameters: the number of estimators (200), learning rate (0.1), and tree depth (6), with column and row sampling (0.8) to prevent overfitting.

Systematic feature selection and importance analysis were performed, revealing that time-weighted engagement metrics and recent post performance were the strongest predictors. The final model architecture effectively combines these various features, with the time-weighted approach showing strong predictive power for engagement patterns. Cross-validation was used to ensure robust performance assessment, and **Log Mean Squared Error** was chosen as the metric because it can handle the wide range of values typically found in social media data.

3. Results

a) Classification

The evaluation of our methodology for classification prediction was carried out by comparing the performance of the baseline model, which utilizes only TF-IDF features with a Naive Bayes classifier, against the proposed final model that integrates additional profile-based features and employs Logistic Regression with optimized hyperparameters.

The baseline **Naive Bayes** model achieved a cross-validation score of **0.59**, indicating moderate performance in predicting Instagram user categories based solely on text features extracted through TF-IDF vectorization. In contrast, our final **Multinomial Logistic Regression** model significantly enhanced predictive accuracy by incorporating user metadata and leveraging a more advanced logistic regression framework. Through feature preprocessing, selection, and scaling, alongside hyperparameter tuning, the final model demonstrated superior generalization capability. The cross-validation score for this improved model rose to **0.74**, reflecting a **15 percentage point improvement** over the baseline.

This substantial increase underscores the importance of combining textual and profile-based features to enrich the representation of user behavior and categories. Furthermore, the feature selection process ensured that only the most informative attributes contributed to the model's predictions, enhancing interpretability and reducing noise.

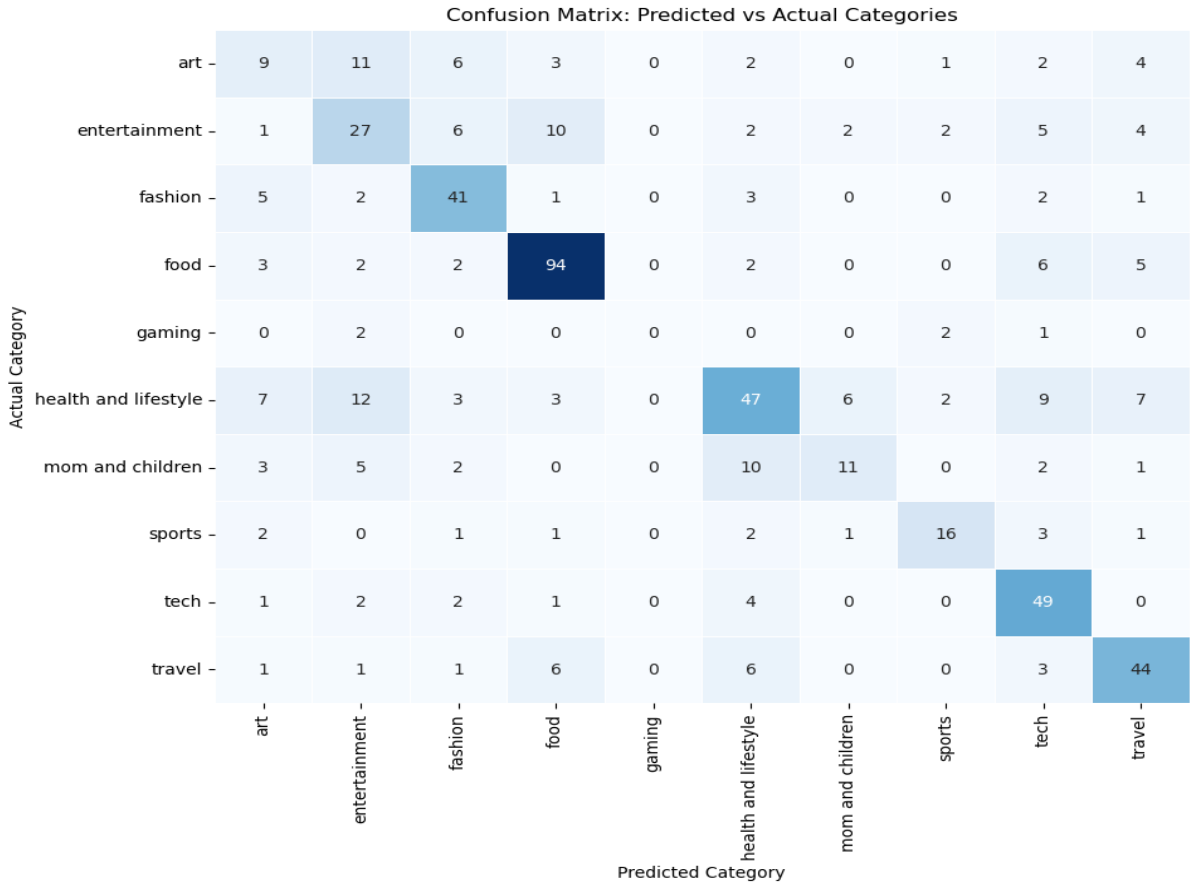
In addition to these performance enhancements, detailed results of our final model and its confusion matrix for the 10 predicted categories could be found below.

Logistic Regression Model Evaluation

```
Best Model - Validation Accuracy with Feature Selection: 0.6156648451730419
Best Model - Validation Classification Report with Feature Selection:
                     precision    recall  f1-score   support

                art       0.28      0.24      0.26        38
      entertainment       0.42      0.46      0.44        59
            fashion       0.65      0.75      0.69        55
               food       0.79      0.82      0.81       114
             gaming       0.00      0.00      0.00         5
health and lifestyle       0.61      0.50      0.55        96
    mom and children       0.50      0.29      0.37        34
             sports       0.70      0.59      0.64        27
               tech       0.60      0.83      0.70        59
             travel       0.66      0.71      0.68        62

           accuracy                           0.62       549
          macro avg       0.52      0.52      0.51       549
       weighted avg       0.60      0.62      0.60       549
```



b) Like Prediction

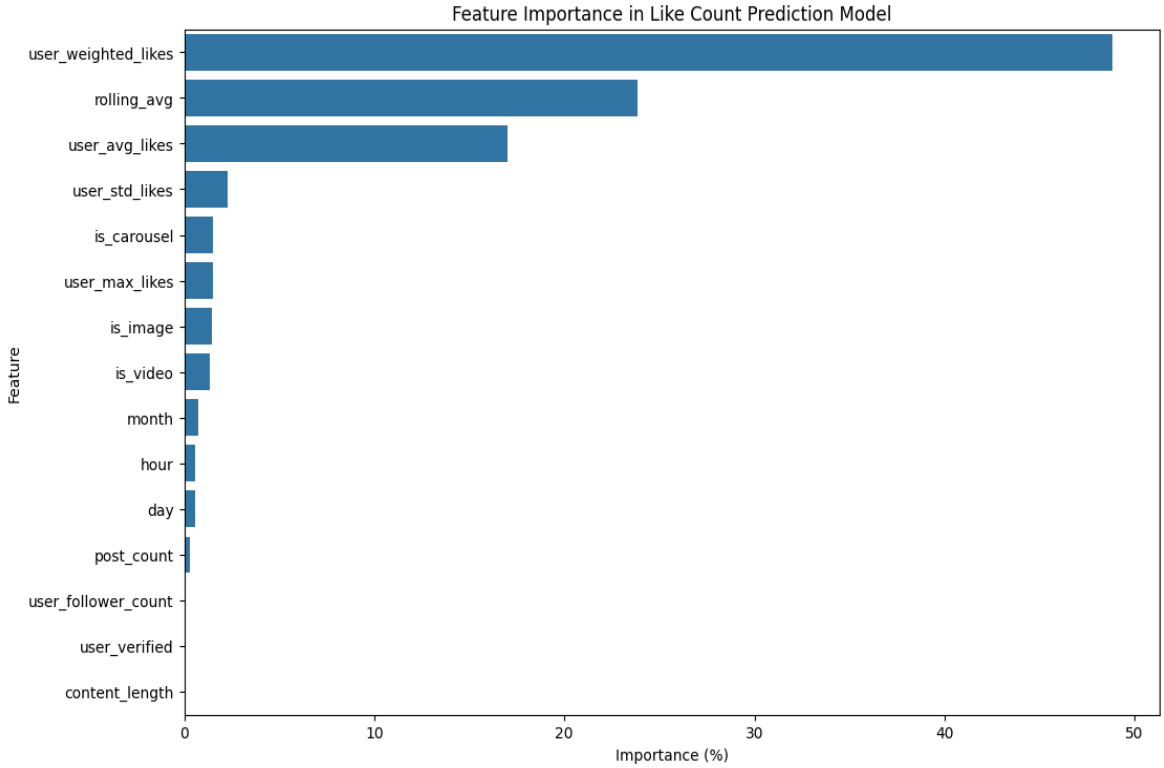
We evaluated our methodology for like count prediction by comparing the performance of a baseline model, which relies on simple historical averages, with our advanced XGBoost model that incorporates temporal patterns and sophisticated feature engineering. The baseline model, which predicts the average number of likes a user receives based on their past activity, achieved a Log MSE of 1.227, reflecting moderate performance.

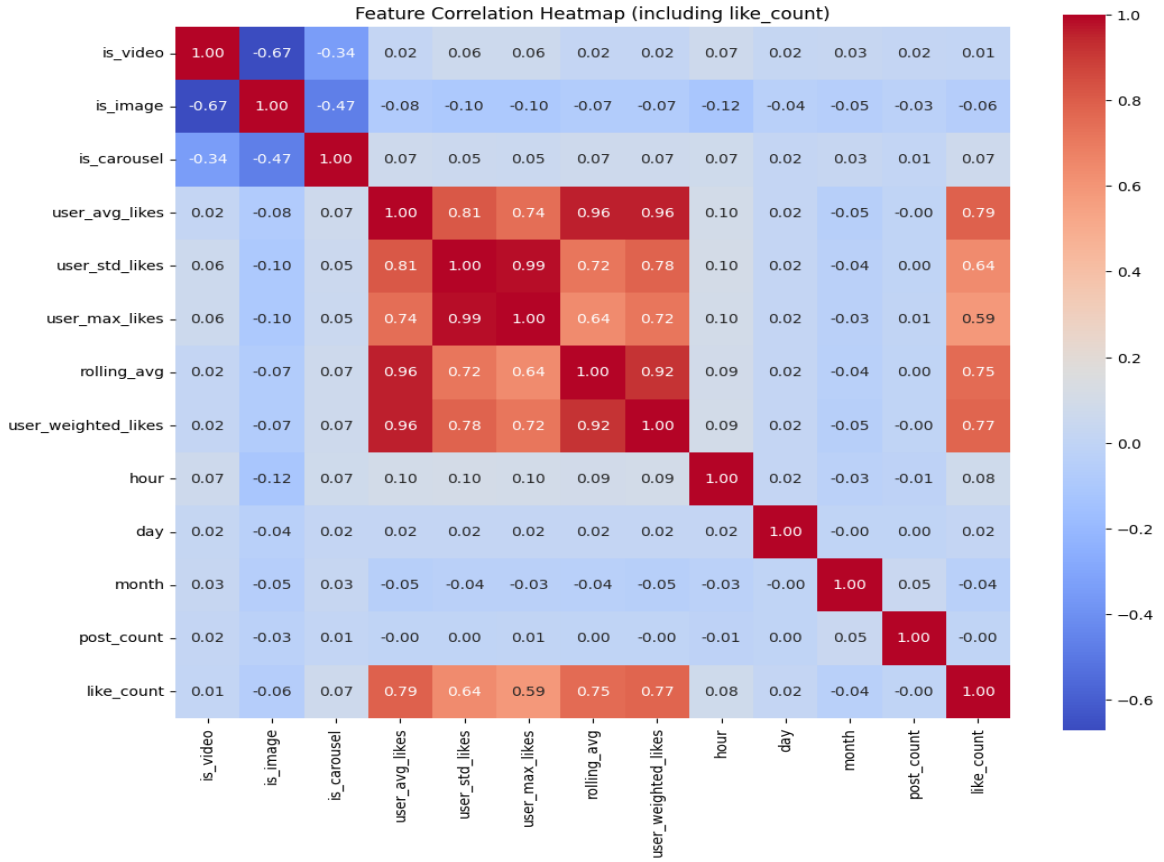
In comparison, our XGBoost Regression model significantly improved prediction accuracy by leveraging time-weighted features and content characteristics. It achieved a Log MSE of 0.522 on the training set and 0.584 on the test set, marking a 52.4% improvement over the baseline.

We tested the model using a large dataset, consisting of 94,824 training samples and 92,478 test samples. Feature importance analysis showed that time-weighted engagement metrics were the most influential predictors, contributing 49.62% to the model's performance. Rolling average engagement features followed with 23.67%, while user average likes accounted for 16.42%. This confirms the effectiveness of our approach, which emphasizes recent engagement patterns through time-weighted features.

Overall, the substantial improvement over the baseline demonstrates the value of our feature engineering strategy and the effectiveness of XGBoost in capturing complex patterns in user engagement. The similar Log MSE values for the training and test sets indicate that the model generalizes well and does not suffer from overfitting.

Detailed Final Model





4. Contributions

Selim Sıdan: Responsible for the entire classification part of the project, including feature engineering, model training, and optimization. For feature engineering, enhanced the provided TF-IDF implementation by incorporating English stopwords and tuning parameters like max_df and min_df to reduce overfitting and improve feature quality. Additionally, preprocessed user-level features by imputing missing values, encoding categorical variables, and standardizing numerical attributes, ensuring a seamless integration of user-level and TF-IDF features. In terms of model development, replaced the baseline Naive Bayes model, which used only TF-IDF features, with a Logistic Regression model employing L1 regularization. This change enabled feature selection and improved handling of the multi-class classification task. Also conducted hyperparameter tuning using GridSearchCV to optimize parameters such as regularization strength and class weighting, leading to a significantly more robust and accurate classification model.

Saner Bilir: Enhanced the regression model through feature engineering and hyperparameter optimization. Developed key derived features including time-weighted engagement metrics using exponential decay, temporal features from timestamps (hour, day, month), and historical engagement patterns (rolling averages, standard deviations, maximum likes). Implemented user-level features that capture engagement history while emphasizing recent activities through time-weighted calculations. For model optimization, conducted systematic hyperparameter tuning of XGBoost parameters including number of estimators (200), learning rate (0.1), maximum tree depth (6), and sampling ratios (0.8 for both column and row sampling).

Fırat Yurdakul: Built a dataframe for the most correlated features on the given dataset and found 5 best most correlated features. After finding these features, used linear regression for the like prediction part in order to improve MSE yet got worse results than xgboost. Also tried Random forest for classification part but compared to Logistic Regression it didn't improve the model. Wrote explanations for the code in regression notebook.

Mert Polat: Worked on feature engineering on the regression model. Extracted rich features for the XGBoost algorithm like recent like counts and their rolling average, standart deviation. Since an account's recent activity reflect the current engagement of that account these features helped get lower MSE for the regression model.