Instagram Likes Prediction

Yi Sun

1 Introduction

This project aims to predict Instagram image likes. The dataset includes follower counts, comments, time of posting, likes, and a set of 3,785 images. We explore the dataset, examine the distribution of likes, preprocess the data, categorize likes into low, medium, and high, apply various models (CNN, CLIP, Random Forest, Logistic Regression, XGBoost), and compare their performance.

2 Data Explore and Cleaning

The distribution of likes is highly skewed, thus we apply log transform to the data. In addition, we found that there weren't any missing values that we need to handle, so we can focus on dealing with the outliers.

2.1 Outlier Removal

We split the training and testing first to leave test set untouched. Then, Z-scores of training set were used to detect and remove outliers. Any points with Z-scores greater than 3 standard deviations were considered outliers and excluded from the training set.

2.2 Likes Categorization

Based on the calculated Z-scores for the training set, likes are classified into one of three categories:

- **Low**: Z-score < -1 (likes significantly below the mean)
- **Medium**: $-1 \leq$ Z-score ≤ 1 (likes close to the mean)
- **High**: Z-score > 1 (likes significantly above the mean)

These categories are then mapped to numeric labels (e.g., 0 for low, 1 for medium, and 2 for high) for classification model training.

For the test set, we use the mean and standard deviation of the training set to calculate the Z-scores. The test likes are then categorized into the same three classes.

3 Models Used

3.1 CNN (Convolutional Neural Network)

The CNN model is designed to classify Instagram images into the three likes categories based on the image content. It consists of three convolutional layers with increasing filter sizes (32, 64, 128), each followed by max-pooling to reduce spatial dimensions. After flattening the extracted features, two fully connected layers are used to predict the like categories via a softmax output layer. The model uses an image data generator, which preprocesses the images in batches (resizing, normalizing, and feeding them) during training and testing, ensuring efficient main memory usage.

3.2 CLIP Model

CLIP was used to extract image embeddings by preprocessing and encoding images into feature vectors. These image embeddings were then combined with log-transformed and scaled numerical features, including the number of comments, followers, and time, to create a comprehensive feature set. A fully connected neural network was built on top of these combined features, consisting of dense layers with dropout for regularization and a softmax layer for classification. The model was trained for 10 epochs and evaluated on test sets.

3.3 Traditional Machine Learning Methods

We also explored three different traditional methods (Random Forest, Logistic Regression, and XGBoost), using only the metadata features.

4 Accuracy Results

Model	Accuracy
CNN	0.504624
CLIP	0.832232
Random Forest	0.829590
Logistic Regression	0.763540
XGBoost	0.833554

Table 1: Model Accuracy Comparison

5 Findings

The findings show that XGBoost and CLIP achieved the highest accuracy (0.833554 and 832232), demonstrating that combining image and textual features or complex ensemble methods are more effective. Traditional methods also performed well, but CNN, using only image data, performed the worst, indicating that image data alone is insufficient for accurate predictions. However, to ensure a fairer comparison and potentially improve results, hyperparameter tuning with cross-validation should be applied across all models if given enough resource and time.