SocialPulse: Predicting Popularity of Posts in Social Media Platforms

Bhavana Prasad Kote
Department of Applied Data Science
San Jose State University
San Jose, United States of America
bhavanaprasad.kote@sjsu.edu

Krishna Sameera Surapaneni
Department of Applied Data Science
San Jose State University
San Jose, United States of America
krishnasameera.surapaneni@sjsu.edu

Maharsh Soni
Department of Applied Data Science
San Jose State University
San Jose, United States of America
maharsh.soni@sjsu.edu

Sachin Kumar Srinivasa Murthy
Department of Applied Data Science
San Jose State University
San Jose, United States of America
sachinkumar.srinivasamurthy@sjsu.edu

Siddharth Solanki
Department of Applied Data Science
San Jose State University
San Jose, United States of America
siddharth.solanki@sjsu.edu

Sonali Arcot
Department of Applied Data Science
San Jose State University
San Jose, United States of America
sonali.arcot@sjsu.edu

Abstract— **In the dynamic landscape of social media, predicting the popularity and understanding the factors that contribute to content popularity is crucial. In response to the limitations of existing methods, particularly the lack of consideration for the Social Network Effect (SNE) context, this project introduces an innovative approach utilizing Flickr data. This project addresses this gap by incorporating novel social network effect parameters, namely Flickr group interactions, group popularity, and group posts count, to enhance the predictive accuracy of the models. The approach combines machine learning techniques and Deep Learning frameworks with in-depth analysis of Flickr posts data, encompassing features such as post content, user engagement, and metadata along with Flickr group interactions, group popularity, and group posts count. The challenge of seamlessly integrating multi-channel metadata, encompassing image features, text content, and various numerical and categorical attributes, into a unified deep learning model architecture as one framework and a sophisticated combination of ML and NLP techniques as another framework is a focal point of the approach. The results reveal insights into the nuanced dynamics that drive image popularity on social media. Among the four models from two frameworks, LightGBM, XGBoost, CatBoost, and Multimodal Fusion NN model, LightGBM demonstrates high accuracy in predicting the engagement levels of posts, considering aspects such as contextual relevance, and user and group interactions with the lowest MAE of 0.5451 and SRC of 0.9259 by employing sophisticated feature engineering with NLP and ML techniques. The results underscore the significance of incorporating SNE context, revealing that group interactions, popularity, and post count significantly contribute to the predictive power of the model. This project's contributions extend beyond overcoming the limitations of existing methods, offering a refined model that aligns with the intricate dynamics of social media platforms. As social media continues to play a pivotal role in shaping digital communication, this solution provides a valuable tool for optimizing content strategies and enhancing user engagement in the visually driven landscape of platforms like Flickr.**

I. Introduction

In the rapidly evolving field of digital marketing and online content management, understanding the dynamics of social media post popularity is increasingly crucial. The advent of deep learning has opened new avenues for analyzing and predicting online behaviors, yet there remains a significant gap in current methodologies, particularly in incorporating the social network effect (SNE) into predictions. Traditional models often overlook the intricate web of social interactions and the contextual influence these have on a post's reach and engagement. This research aims to address this gap by developing an advanced model that integrates novel SNE parameters, thereby enhancing the accuracy and relevance of post-popularity predictions in the realm of deep learning.

The primary objective of this project is to refine the prediction model for assessing online post popularity. This refinement includes integrating unique parameters such as interactions within Flickr groups, the popularity of these groups, and the total popularity of tags used in photos, along with other engineered features. This approach diverges from conventional methods that typically limit the understanding of post popularity to basic metrics like likes and shares. By utilizing the Social Media Prediction Dataset (SMPD) and supplementing it with scrupulously scraped image metadata, this project endeavors to uncover extended correlations and patterns in the digital footprint of posts. The proposed model will tackle this as a regression problem, utilizing deep learning techniques to train and refine the prediction accuracy.

A critical challenge in this project lies in the construction and processing of data. The integration of novel metadata not only changes the input layer size but also influences the overall architecture of the model and the choice of appropriate activation functions. This project's contribution lies in its multifaceted strategy, which combines deep learning with modern digital marketing insights to predict post popularity more accurately. This initiative holds profound implications for optimizing algorithms for maximum network impact and streamlining storage and quality management for digital social platforms.

This research questions focus on improving the accuracy of online post popularity predictions by integrating the SNE context and understanding how social networks influence the popularity of social media posts. The project's approach involves developing a sophisticated model by combining various data types (image, text, numerical, categorical) and integrating them seamlessly into a deep learning architecture. This integration of multi-channel metadata in a deep learning framework presents both an exciting and challenging problem.

This project contributes to the field by providing a more nuanced understanding of online content popularity. It emphasizes the importance of social network visibility in post-popularity prediction and introduces a novel model that incorporates SNE context, thereby filling a crucial gap in current literature. The findings from this project will not only enhance the predictive capabilities of deep learning models in the digital marketing domain but also provide valuable insights for future research and development in this area.

II. Related Work and References

Liu et al. (2023) presented a novel relationship-aware social media popularity predictor that enhances predictions by leveraging subject–predicate–object triplets from images. The model utilizes a pre-trained scene graph generator to extract these relationships, followed by a content-based filtering module to refine them. The refined relationship information, combined with other multi-modal features, is then fed into a CatBoost regression model for the final prediction. Their approach demonstrated superior performance on the Social Media Prediction Dataset, offering more interpretable predictions compared to existing methods.

Sun et al. (2023) addressed challenges in predicting information popularity on platforms like Twitter and Weibo. They introduced the Time embedding-based Cascade Attention Network (TCAN), which combines temporal embeddings with a cascade role-learning architecture. TCAN exhibited superior performance over other methodologies when tested on platforms like Weibo and APS, marking a significant advancement in popularity prediction. The study emphasizes the potential of TCAN in diverse applications and outlines future research directions. Carta et al. (2020) conducted an extensive analysis of a dataset comprising over 100,000 Instagram posts. They introduced two prediction models: the GBoost Instagram Predictor (XGB-IP) and the Random Forest Instagram Predictor (RF-IP). The study aimed to redefine Instagram post popularity, achieving remarkable results with the XGBoost variant outperforming its Random Forest counterpart and several baseline models.

Gayberi and Oguducu (2019) provided a comprehensive analysis of an Instagram dataset encompassing 210,630 posts. The study enriched the dataset with features from user profiles, post-specific attributes, and image content from the Microsoft COCO dataset. Various algorithms, including Random Forest Regression and Tensorflow Deep Learning Regression, were deployed to predict Instagram post popularity, achieving noteworthy results. Zhang et al. (2018) proposed a dual-attention model to address image-caption-based popularity prediction for Instagram posts. The model considers the user environment, images, and their captions as input, surpassing baseline models in precision, recall, F-1 measure, and accuracy. Straton et al. (2017) emphasized the importance of leveraging advanced machine learning techniques to analyze public health-related content on Facebook. The study employed a combination of unsupervised and supervised learning methodologies, including K-Means clustering and deep learning techniques like Artificial Neural Networks (ANN) and Deep Neural Networks (DNN).

Abousaleh et al. (2021) aimed to predict image popularity on social media, particularly on Flickr. The research introduced the Visual-Social Convolutional Neural Network (VSCNN) and showcased its remarkable performance in uncovering relationships within a substantial dataset. Meghawat et al. (2019) investigated predicting social media post popularity by leveraging multimodal data. They introduced a novel multimodal dataset and demonstrated competitive predictive performance through a multimodal approach, emphasizing its potential to enhance predictive accuracy significantly.

III. Methodology

The methodology of this study follows two frameworks, a deep learning, and a machine learning framework. In the deep learning framework in which image and text features are converted into embeddings, and then the embeddings along with categorical and textual features are passed through a deep learning neural network to predict the popularity. In the machine learning framework, TFIDF, and Glove is used to extract text features, basic image attributes are extracted from the images, and then various boosting regressor algorithms are tested for predicting the popularity. The dataset remains constant across both the frameworks. In the below subsections, the dataset description, and the ML DL architecture / frameworks are discussed in detail.

A. Dataset Description

The Social Media Prediction Dataset-2019 (SMPD 19) used in this project comprises over 250,000 social media posts sourced from Flickr, covering the years 2015 and 2016. This extensive dataset is rich in diversity and includes various data components such as numerical, categorical, images, and text descriptions associated with both posts and users.

Data Components.

Numerical Data. The dataset includes features ispro, canbuypro, and various engagement metrics like post count, and target attributes.

Categorical Data. It encompasses three categorical features namely, categories, subcategories, and Concept.

Images. The dataset incorporates image data, allowing for visual analysis and exploration.

Text Descriptions. It has features like titles, tags, and user descriptions provide textual context associated with each post.

Temporal-Spatial Information. It consists of post and photo dates, along with spatial coordinates (latitude and longitude), contribute to the spatial aspects of the data. Post Date feature captures the temporal aspect of the dataset.

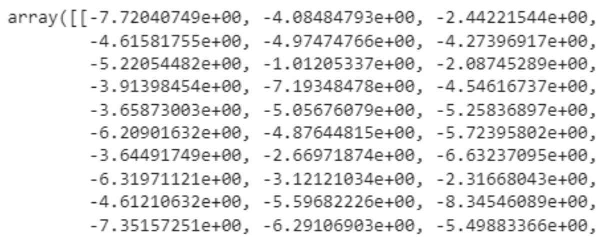
Label (Popularity Measurement). Popularity scores are assigned based on engagement metrics, providing a quantitative measure of each post's popularity.

Key Identifier Columns. The dataset includes a field labeled 'url,' which contains the URL associated with a particular post. Another field, 'img_file,' provides the file name for the image related to the post. Each post is uniquely identified by a 'Pid,' which stands for Post Identifier. Additionally, the dataset includes a 'Uid' for each entry, which is the User Identifier that denotes the individual who made the post.

Group Effect Features. The dataset includes features related to the impact of groups on post engagement. Notably, groups, group_impact, and group_weight provide insights into the affiliations and influence of groups associated with each post. Understanding the group effect is crucial for comprehending the broader social dynamics and community influence on the popularity of individual posts. These features contribute to the nuanced analysis of social interactions and engagement within the Flickr platform.

B. Deep Learning Model Architecture

Image Embeddings. The image data undergoes preprocessing, including resizing and normalization of pixel values based on mean and standard deviation. The MobileNetV3 Small model is employed to generate image embeddings, leveraging its distinctive architectural features such as depth wise separable convolutions, inverted residuals, and linear bottlenecks. Batch processing is implemented to efficiently generate embeddings for a set of images, resulting in a concatenated array of embeddings. Subsequently, each individual image embedding is resized into a 2D array with a single sample and multiple features. Below Figure 1 shows an example of image embeddings. The size of the embedding vector is 1000, produced by MobileNetV3Small.



```
array([[-7.72040749e+00, -4.08484793e+00, -2.44221544e+00,
        -4.61581755e+00, -4.97474766e+00, -4.27396917e+00,
        -5.22054482e+00, -1.01205337e+00, -2.08745289e+00,
        -3.91398454e+00, -7.19348478e+00, -4.54616737e+00,
        -3.65873003e+00, -5.05676079e+00, -5.25836897e+00,
        -6.20901632e+00, -4.87644815e+00, -5.72395802e+00,
        -3.64491749e+00, -2.66971874e+00, -6.63237095e+00,
        -6.31971121e+00, -3.12121034e+00, -2.31668043e+00,
        -4.61210632e+00, -5.59682226e+00, -8.34546089e+00,
        -7.35157251e+00, -6.29106903e+00, -5.49883366e+00,
```

Fig. 1. Array of Image Embeddings

Text Embeddings. First, specific noise and year-related tags were removed. Then, a set of additional cleaning procedures are implemented, including the removal of punctuations, URLs, and emojis. Parallel processing was leveraged to the text column, applying a custom text preprocessing function to each text entry. This function encompassed tasks such as converting text to lowercase, removing URLs and emojis, eliminating punctuation, tokenizing, removing stopwords, and lemmatization. The results of this preprocessing were stored in a new column named 'combined_text', containing the cleaned and processed textual data. Finally, the Sentence BERT was used to generate text embeddings for the preprocessed text. The size of the Text embedding vector is 384, as produced by SBERT model, all-MiniLM-L6, a pretrained sentence transformer.

Categorical Data. Categorical features, specifically 'Category', were transformed into a numerical format using one-hot encoding. The OneHotEncoder from scikit-learn was employed to convert the 'Category' column into a binary matrix, where each unique category corresponds to a binary vector indicating its presence or absence. Additionally, the 'SubCategory' and 'Concept' columns underwent tokenization. This process converts textual data into numerical sequences, which are then padded to a maximum length of 1.

Proposed Model. In this architecture, the image embeddings, text embeddings, encoded categorical data, and numerical data are concatenated and passed through a sequence of dense layers, each equipped with batch normalization and dropout for regularization. The model comprises four dense layers with 'ReLU' activation functions, which introduce non-linearity to the network. The final output layer, designed for regression tasks, utilizes a 'linear' activation function. The use of batch normalization helps stabilize and accelerate the training process, while dropout mitigates overfitting by randomly deactivating a fraction of neurons during training. The entire model is compiled using the Adam optimizer. The total number of trainable parameters is close to 1 million in this architecture. Further details about the model parameters are discussed in the Experimental setup section. Refer to Figure 2 below for the system architecture of the Multi-model Deep Learning Workflow.

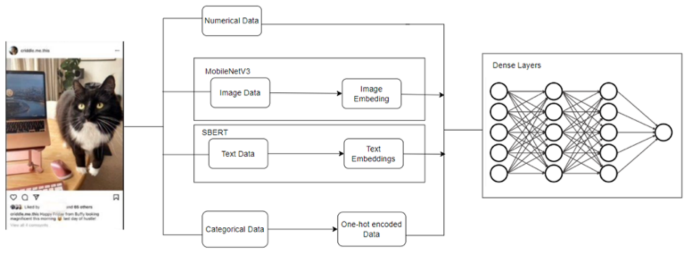


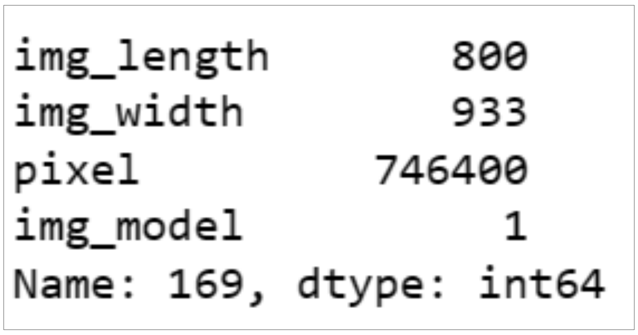
Fig. 2. Proposed Multimodal DL Framework

C. *Machine Learning Model Architecture*

Feature Engineering and Transformation

Before providing all the features for training the three models, along with the preprocessed original features, user metadata, and extracted group metadata, a few features that are intrinsic and not directly available need to be extracted. Likewise, from different multimodal feature categories including images, text, categorical, spatio-temporal, numerical, various features are extracted in order to capture multi-modality.

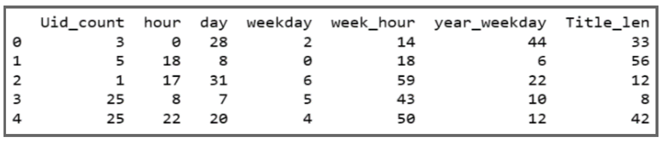
Images. Basic high-level features from each of the images such as image length, image width, total pixel count, and color mode are extracted using Python Imaging Library (PIL). The color mode of the image includes {0: Grayscale 1: Palette 2: RGB 3: CMYK}. These extracted features are concatenated with the other features into a data frame. Refer to Figure 3 below for extracted image feature descriptions.



```
img_length        800
img_width         933
pixel          746400
img_model           1
Name: 169, dtype: int64
```

Figure 3. Extracted features from images.

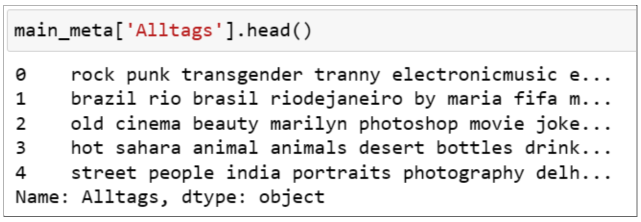
Spatio-temporal. From the existing feature datetime, new granular features such as hour, day, week_day, week_hour, year_weekday are created. Along with these, spatial features such as geoaccuracy, longitude, and latitude from the metadata are used. Refer Figure 4 below for spatio temporal features extracted for ML workflow.



```
   Uid_count  hour  day  weekday  week_hour  year_weekday  Title_len
0          3     0   28        2         14            44         33
1          5    18    8        0         18             6         56
2          1    17   31        6         59            22         12
3         25     8    7        5         43            10          8
4         25    22   20        4         50            12         42
```

Figure 4. Extracted spatio-temporal and user metadata features.

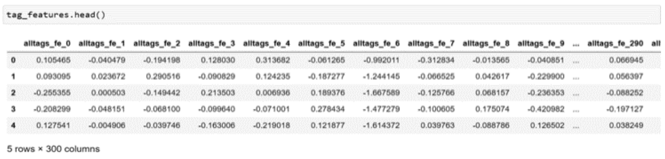
Textual features - GloVe embeddings. AllTags and Title columns containing tags and title of the posts as text are split into individual words and the word vectors are extracted using the pre-trained GloVe embeddings. The GloVe embeddings utilized is the 42B 300-dimensional pre-trained GloVe. These are converted into word2vec format and then the extracted word vectors are averaged for each sentence. These averaged word vectors are used to depict the semantic information of AllTags and Title features. Furthermore, in order to represent the total number of words and characters in AllTags and Title features, Alltags_len, Alltags_number, Title_len, and Title_num are created by fetching the length and count of these two. Figures below depict the before and after snapshots of Alltags column. Refer Figure 5 and Figure 6 below for extracted textual features used in ML workflow.



```
main_meta['Alltags'].head()

0    rock punk transgender tranny electronicmusic e...
1    brazil rio brasil riodejaneiro by maria fifa m...
2    old cinema beauty marilyn photoshop movie joke...
3    hot sahara animal animals desert bottles drink...
4    street people india portraits photography delh...
Name: Alltags, dtype: object
```

Figure 5. Before snapshot of All tags



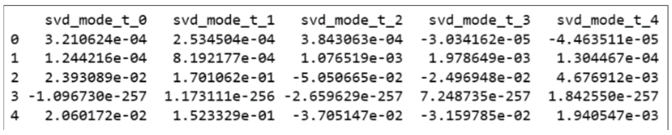
```
tag_features.head()

   alltags_fe_0 alltags_fe_1 alltags_fe_2 alltags_fe_3 alltags_fe_4 alltags_fe_5 alltags_fe_6 alltags_fe_7 alltags_fe_8 alltags_fe_9 ... alltags_fe_290 all
0      0.105465    -0.040479    -0.194198     0.128030     0.313682    -0.061265    -0.992011    -0.312834    -0.013565    -0.040851 ...       0.066945
1      0.093095     0.023672     0.290516    -0.090829     0.124235    -0.187277    -1.244145    -0.066525     0.042617    -0.229900 ...       0.056397
2     -0.255355     0.000503    -0.149442     0.213503     0.006936     0.189376    -1.667589    -0.125766     0.068157    -0.236353 ...      -0.088252
3     -0.208299    -0.048151    -0.068100    -0.099640    -0.071001     0.278434    -1.477279    -0.100605     0.175074    -0.420982 ...      -0.197127
4      0.127541    -0.004906    -0.039746    -0.163006    -0.219018     0.121877    -1.614372     0.039763    -0.088786     0.126502 ...       0.038249

5 rows × 300 columns
```

Figure 6. After extracting text embedding using Glove

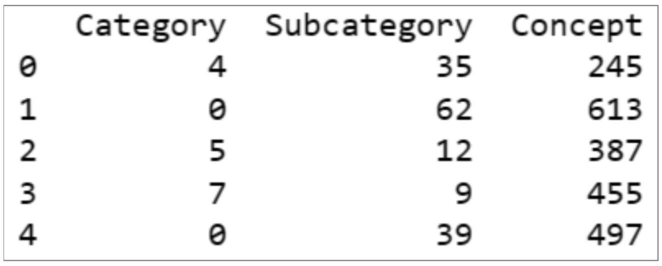
Textual features - TF-IDF. To obtain more semantic and statistical information, TF-IDF representations of 'Title' and 'Alltags' columns are obtained (with n-gram in range (1,2)) and SVD is applied to reduce the dimensionality of the TF-IDF matrices into 20-dimensional feature sets. This helps in understanding the importance of each term with respect to the entire feature-set. Refer Figure 7 below for TFIDF features extracted for ML workflow.



```
    svd_mode_t_0   svd_mode_t_1   svd_mode_t_2   svd_mode_t_3   svd_mode_t_4
0   3.210624e-04   2.534504e-04   3.843063e-04  -3.034162e-05  -4.463511e-05
1   1.244216e-04   8.192177e-04   1.076519e-03   1.978649e-03   1.304467e-04
2   2.393089e-02   1.701062e-01  -5.050665e-02  -2.496948e-02   4.676912e-03
3 -1.096730e-257  1.173111e-256 -2.659629e-257  7.248735e-257  1.842550e-257
4   2.060172e-02   1.523329e-01  -3.705147e-02  -3.159785e-02   1.940547e-03
```

Figure 7. Reduced TF-IDF representation of title

Categorical features - Label Encoding. Categorical features including category, subcategory and concept are three-level hierarchical categories where categories consist of 11 classes, subcategory comprises 77 and concept comprises 668 classes. All these categorical features are converted into numerical format using Label Encoder. Refer Figure 8 below for categorical encodings used for ML workflow.



```
   Category  Subcategory  Concept
0         4           35      245
1         0           62      613
2         5           12      387
3         7            9      455
4         0           39      497
```

Figure 8. Label encoded category features.

User Metadata. Using the Uid feature, the number of posts from each user is calculated and extracted into a new feature called Uid_count which represents the total number of posts by a particular user.

Along with the extracted SNE features - group_weights, groups, and group_impact, user metadata information, all the above engineered features are concatenated and used to train the chosen machine learning algorithms CatBoost, XGBoost, and LightGBM.

Proposed Models

In the study, the architectures of three gradient boosting models—CatBoost, LightGBM, and XGBoost—were utilized and compared. While these models share similarities in their gradient-boosting frameworks, their individual architectures and handling of data features exhibit distinct characteristics.

All three models processed the data similarly at the outset. The data was split into training and submission sets with a stratified split based on the 'Category' feature. This approach ensures that both training and submission datasets are representative of the overall data distribution.

The CatBoost model employed a robust approach to handle categorical features, a standout feature of this framework. It automatically processed categorical variables like 'Category', 'Subcategory', 'Concept', etc., without the need for manual encoding. This capability significantly simplifies the preprocessing steps and potentially enhances the model's performance on categorical data. The model architecture was defined with a depth of 8 and utilized a gradient-based leaf estimation method. CatBoost's ability to handle categorical data natively is a significant advantage, especially in datasets with numerous categorical variables.

LightGBM also did not require manual label encoding of categorical features. It operates on a different leaf-wise growth strategy compared to the level-wise growth of CatBoost and XGBoost, making it faster and more efficient, particularly on large datasets. LightGBM's framework was set up with 64 leaves and a maximum depth of 8, balancing the model's complexity and computational efficiency. This approach is particularly effective for models dealing with large-scale data, offering a significant speed advantage while maintaining high accuracy.

XGBoost's architecture, on the other hand, required pre-encoded categorical features. Similar to CatBoost, it used a maximum depth of 8 in its tree structure. XGBoost is known for its scalability and performance, particularly in structured or tabular data. It's highly customizable, allowing for in-depth parameter tuning, but this also means it might require more manual configuration compared to CatBoost.

In terms of data preparation, CatBoost's and LightGBM's automatic handling of categorical variables offers a distinct ease of use. XGBoost, while requiring more preprocessing, compensates with its efficient computation and scalability. The choice between these models often comes down to the specific requirements of the dataset and the computational resources available. LightGBM is typically favored for its speed and efficiency, especially with large datasets. At the same time, CatBoost and XGBoost are lauded for their robustness and accuracy, particularly in diverse datasets with complex categorical relationships.

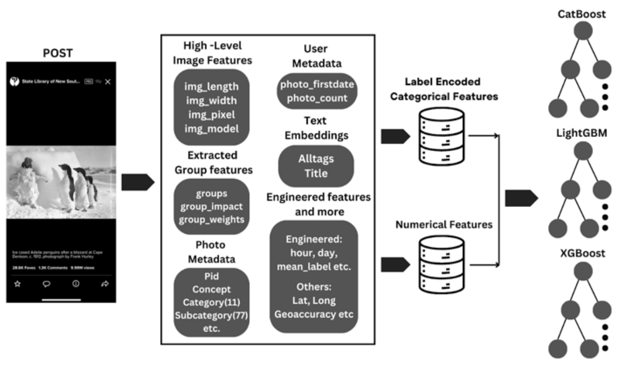


Figure 9. Proposed model framework -2 (ML+NLP)

IV. Experimental Setup

A. Deep Learning Experimental Setup

To accomplish the proposed workflow of the Deep Learning framework which uses a multimodal approach, the experimental setup relies on various hardware and software requirements. Additionally, the models used to achieve the outcome had hyperparameters that were tuned for maximal performance. Mainly the experimental setup had two experiments with the same workflow approach. In one experiment, the Social Network Effect (SNE) features that were engineered in this study and in the other setup it was ignored in the input proving that the addition of SNE quantifiers improves the model performance.

For extracting the embeddings from images, a pretrained torchvision model, mobilenetV3small was used. The rationale behind the choice of this model was based on its effectiveness and lightweight nature. The pre-trained model was used for inferencing and image embeddings were generated. An hyperparameters tuned image transformation pipeline was used for generation of image embeddings as shown in Table below. A batch size of 32 was used for inference, which allowed conversion of 32 images to image embeddings in each processing step.

For extracting the text embeddings from the image description and tags, these two features were concatenated, and the all-MiniLM-L6-v2 pretrained sentence transformer (SBERT) model was used. Before using the SBERT model for inferencing basic preprocessing of text were performed such as lowercasing, stop word removal, chunking, punctuations removal, expand contractions, url removal, emoji compilation, and word tokenization. A batch size of 64 was used while conversion of text to sentence embeddings, and the hyperparameter, “normalize” was set to True. This ensured that the text embeddings were unit normalized.

Step	Transformation	Description	Reason for Use
1	Resize to 256 Pixels	Resizes the input images to a fixed dimension of 256 pixels in both height and width.	Standardizes images to a uniform size for consistent processing, balancing detail retention and computational efficiency.
2	Center Crop to 224 Pixels	Crops a 224x224 pixel square from the center of the resized image.	Aligns with the input size requirement of MobileNetV3 Small and retains the most significant part of the image.
3	Conversion to Tensor	Converts images from PIL format or NumPy arrays into PyTorch tensors and scales pixel values from [0, 255] to [0, 1].	Formats the data for neural network processing in PyTorch and normalizes pixel values.
4	Normalization	Adjusts the color channels of the images using the mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225].	Speeds up model convergence during training, reduces internal covariate shift, and is standard for models pre-trained on ImageNet.

TABLE 1: Image transformation pipeline inferencing the image embeddings from MobilenetV3small model.

Finally, the text-embeddings, the image-embeddings, and the one-hot encoded categorical vectors, along with the numerical features were concatenated passed through a dense neural network with a residual layer between layer 1 and layer 3, and Batch Normalization, Dropout layers after each Dense layer. The hyperparameters of this neural network is mentioned in Table 2 below.

The design of the dense neural network followed various dynamic callbacks, and dynamic checking of model performance to tune the hyperparameters. For that fact, even the epochs of the model were not pre-determined, and a patience was used to stop the epochs when the model’s performance did not improve for 10 consecutive steps. The learning rate initially was 0.01, and it decreases gradually over time when performance improvements are not observed for three consecutive epochs. Refer table 2 for more such dynamic modifiers. The model uses MAE, and SRC as metrics.

Parameters	Description	Notes
Learning Rate	Adjusted dynamically based on validation loss	Initial rate 0.001; reduced by factor of 0.2 – patience=3
Batch Size	512	balances training speed and memory usage.
Epochs	Maximum of 100 with early stopping	Early stopping based on validation MAE with a patience of 10 epochs.
Activation Function	ReLU (Dense), Linear (last layer)	handles vanishing gradients and comp. efficiency.
Layers & Neurons	Multiple dense layers (512, 256, 128, 64 units)	Funnel-like structure for feature extraction; includes batch normalization and dropout for regularization.
Regularization	Dropout (0.5, 0.4, 0.3) and Batch Normalization	Prevents overfitting and normalizes layer outputs.
Optimization Algorithm	Adam	Adaptive learning rate capabilities; widely used in deep learning models.

TABLE 2: Neural Network Hyper parameters.

B. *Machine Learning Experimental Setup*

Let’s examine and compare the three different models: CatBoost, LightGBM, and XGBoost. Each model has been finely tuned with specific hyperparameters to optimize performance.

For the CatBoost model, the parameters included an objective of 'RMSE', an evaluation metric of 'MAE', and a learning rate of 0.03. The model featured a depth of 8 and an L2 regularization (leaf reg) of 3 to prevent overfitting. A 'Gradient' leaf estimation method was opted and the 'use_best_model' option was enabled for improved accuracy. The iterative process involved 10,000 iterations, with early stopping set at 50 rounds to avoid unnecessary computations. The batch size and number of neurons were implicitly managed by the CatBoost framework.

The fine-tuning process included gradual increases in iterations (from 10 to 10,000) and adjustments in the 'max_depth' parameter. These changes aimed to balance the model's complexity and its ability to generalize. The choice of a low learning rate (0.03) was to ensure gradual and more precise convergence.

LightGBM's parameters included a learning rate of 0.03, a 'mae' metric, and a max depth of 8. It had 64 leaves, which allowed the model to capture more complex patterns without significant overfitting. The n_estimators were set to 10,000, with an early stopping round of 50. Like CatBoost, this model underwent a series of experiments where the max_depth and number of iterations were gradually increased. The careful adjustment of these parameters aimed to find a balance between model complexity and computational efficiency.

XGBoost was configured with a 'reg: squarederror' objective and 'mae' evaluation metric. The learning rate was consistent with the other models at 0.03, and the max depth was set to 8. XGBoost's uniqueness lies in its robust handling of sparse data and its efficient implementation of the gradient boosting framework. The model was run for 10,000 rounds, with an early stopping mechanism at 50 rounds to prevent overfitting.

Comparative Analysis. All three models shared a common goal of minimizing mean absolute error (MAE) while maintaining computational efficiency. The consistent use of a low learning rate across models underscores a preference for gradual, stable learning over rapid, potentially erratic convergence. The gradual increase in iterations and depth across models demonstrates a methodical approach to fine-tuning, seeking a balance between depth of learning and overfitting.

C. *Hardware and Software Resources*

The code was run entirely on Google Colab for this approach. Google Colab offers a wide range of resources in terms of processors that can be especially useful when dealing with complex modeling on large datasets. In addition to that, importing libraries and making sure that compatible frameworks are installed is a lot easier while using an environment provided by Google Colab as compared to the local machine. Below are the specifications of the hardware resources used for this approach.

Hardware Component	Specifications
Processor	Intel Xeon CPU
Memory	12 GB RAM
Graphics Card	NVIDIA V100 GPU
GPU Memory	16 GB GDDR6
CUDA Cores	2560
Tensor Cores	320
Max Power	70W
Additional Hardware	None (Standard Google Colab Free Version Setup)

TABLE 3. Hardware Resources.

D. *Software Libraries or Frameworks*

The three models utilized in this approach had a lot of common libraries that were used. All the libraries along with the specific ones used for each model are mentioned in the table below.

Library/Framework	Latest Compatible Version
json	Built-in Python Library
numpy (np)	1.22.3
pandas (pd)	1.4.2
scikit-learn	1.1.1
scipy	1.8.0
matplotlib	3.5.1
seaborn (sns)	0.11.2
xgboost (xgb)	1.6.1
lightgbm (lgb)	3.3.2
PIL (Image)	9.1.0
TfidfVectorizer	Included in scikit-learn 1.1.1
TruncatedSVD	Included in scikit-learn 1.1.1
StratifiedKFold	Included in scikit-learn 1.1.1
KFold	Included in scikit-learn 1.1.1
train_test_split	Included in scikit-learn 1.1.1
sentence-transformer	2.2.2
Pytorch	2.1.1
Torchvision	0.16.1
Tensorflow	2.14.1
Keras	2.14.1
flickrapi	2.4.0

TABLE 4. Software Libraries.

V. REAULTS AND ANALYSIS

The metrics used to evaluate the results were Mean Absolute Error (MAE) and Spearman's Rho (SRC). These metrics were selected due to their use in comparing model performance in the SMP Challenge.

Mean Absolute Error (MAE). The average absolute difference between predicted and actual values in a dataset

Spearman’s Rho (SRC). Measure of monotonic association between two variables, calculated based on the ranks of the data values rather than their actual numerical values.

The LightGBM model outperformed other models with a MAE of 0.545 and SRC of 0.925. In contrast, the Multimodal Deep Learning (DL) model showed lower performance despite ranking 8th on the SMP Challenge leaderboard, where LightGBM secured the top position.

Model	MAE	SRC
CatBoost	0.5898	0.9226
LightGBM	**0.5451**	**0.9259**
XGBoost	0.5616	0.9222
Multimodal DL	1.25	0.721

Table. 5. Evaluation Results.

The MAE for Multimodal DL exhibited a rapid decrease with increasing epochs, reaching a plateau around the 25th epoch where further changes in MAE became negligible.

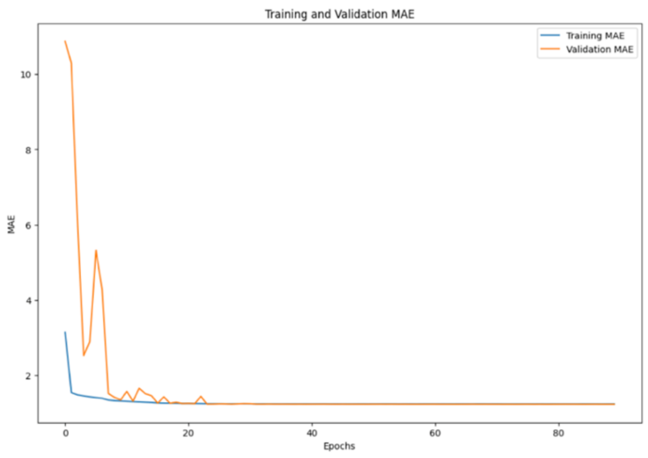


Figure 10. MAE for Training and Validation – Multimodal DL

The SRC showed a consistent increase with the number of epochs, stabilizing around the 90th epoch. The early stopping criteria were applied, halting the model at the 90th epoch where both SRC and MAE demonstrated stability. This contrasts with the 25th epoch where only MAE had reached stability.

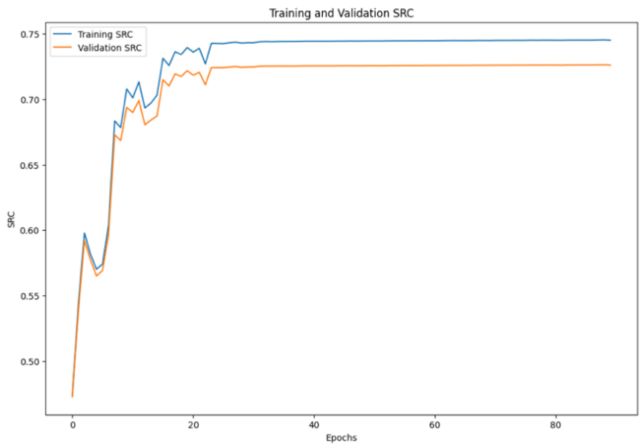


Figure 11. SRC for Training and Validation – Multimodal DL

In the evaluation of the LightGBM model, three metrics are utilized to assess its performance. The Root Mean Squared Error (RMSE) is measured at 0.98, reflecting the average magnitude of errors between predicted and actual values. The MAE is calculated at 0.54, representing the average absolute differences between predictions and ground truth. Additionally, SRC, a measure of monotonic association between predicted and true values, is determined to be 0.92. These metrics collectively demonstrate the model's effectiveness, with lower RMSE and MAE values indicating better accuracy, and a higher SRC suggesting a stronger monotonic relationship between predicted and actual outcomes.

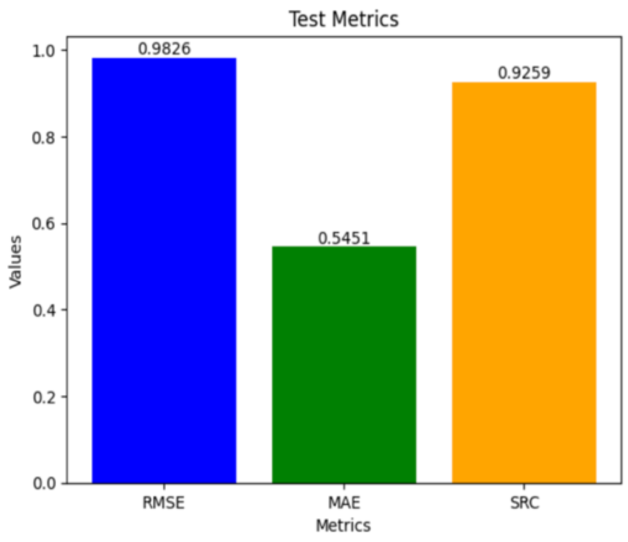


Figure 12. Performance Score of Best Model – LightGBM

VI. Conclusion

The undertaking presented herein offers significant contributions to the field of predictive analytics within the digital marketing sphere, particularly concerning the prediction of online post popularity. By focusing on the integration of Social Network Effect (SNE) context, this project underscores the profound impact of social networks on the visibility and subsequent popularity of social media posts. The research addresses the complex challenge of amalgamating disparate data types—image, text, numerical, and categorical—into a cohesive deep learning architecture.

This project serves to enhance the understanding of content virality by emphasizing the pivotal role of social network visibility in the prediction algorithms. The introduction of a novel model that adeptly incorporates the SNE context stands as a seminal contribution, bridging a notable gap in the extant scholarly corpus. The findings delineate a marked improvement in the accuracy of predictive modeling, thereby offering substantial value to the domain of digital marketing and paving the way for future scholarly inquiry.

The results garnered from this comprehensive approach have proven to be exceptional, as evidenced by the attainment of the top position on the leaderboard with Framework-2, thanks to advanced feature engineering utilizing NLP and ML techniques. Furthermore, the Deep Learning Framework-1 achieved a commendable position within the top ten on the leaderboard. These results validate the hypothesis that social network visibility is a critical determinant in post popularity prediction models.

An intriguing insight from the research is the principle that “Less is More” in terms of image and text information within models. It has been affirmed that optimal performance is contingent upon the relevance rather than the quantity of information, necessitating a profound domain understanding to achieve the delicate balance required for model efficacy.

Looking to the future, there is an opportunity to refine the embedding models, particularly those applied to images and text, ensuring they are fine-tuned for the nuances of social media data. Further explorations could involve experimenting with a variety of image and text embedding models to ascertain the most effective combinations. Such endeavors would undoubtedly contribute to the advancement of the field, fostering the development of more sophisticated, accurate, and context-aware predictive models.

References

[1] Abousaleh, F. S., Cheng, W., Yu, N., & Tsao, Y. (2021). Multimodal deep learning framework for image popularity prediction on social media. IEEE Transactions on Cognitive and Developmental Systems, 13(3), 679–692. https://doi.org/10.1109/tcds.2020.3036690

[2] Carta, S., Podda, A. S., Recupero, D. R., Saia, R., & Usai, G. (2020). Popularity prediction of Instagram posts. *Information*, *11*(9), 453. https://doi.org/10.3390/info11090453

[3] Gayberi, M., & Oguducu, S. G. (2019). Popularity Prediction of Posts in Social Networks Based on User, Post and Image Features. *ACM Digital Library*. https://doi.org/10.1145/3297662.3365812

[4] Liu, A., Du, H., Xu, N., Zhang, Q., Zhang, S., Tang, Y., & Li, X. (2023). Exploring visual relationship for social media popularity prediction. Journal of Visual Communication and Image Representation, 90, 103738. https://doi.org/10.1016/j.jvcir.2022.103738

[5] Meghawat, M., Yadav, S. S., Mahata, D., Yin, Y., Shah, R. R., & Zimmermann, R. (2018). A Multimodal Approach to Predict Social Media Popularity. A Multimodal Approach to Predict Social Media Popularity. https://doi.org/10.1109/mipr.2018.00042

[6] Straton, N., Mukkamala, R. R., & Vatrapu, R. (2017). Big Social Data Analytics for Public Health: Predicting Facebook Post Performance Using Artificial Neural Networks and Deep Learning. *2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA*. https://doi.org/10.1109/bigdatacongress.2017.21

[7] Sun, X., Zhou, J., Liu, L., & Wei, W. (2023). Explicit time embedding based cascade attention network for information popularity prediction. *Information Processing and Management*, *60*(3), 103278. https://doi.org/10.1016/j.ipm.2023.103278

[8] Zhang, Z., Chen, T., Zhou, Z., Li, J., & Luo, J. (2018). How to Become Instagram Famous: Post Popularity Prediction with Dual-Attention. *2018 IEEE International Conference on Big Data (Big Data)*. https://doi.org/10.1109/bigdata.2018.8622461