Social Popularity Prediction

Data Mining Homework 2

B10704031 Po-Yen Chu

1 Abstract

This paper discusses the prediction of Social Popularity, including preprocessing methods, the chosen model, results, and evaluation. Before proceeding, note the execution details when running the provided notebook (.ipynb):

- **Environment**: The provided code should be run on Kaggle Notebook or Colab (with necessary file path modifications) to utilize CUDA (GPU) for improved efficiency.
- **Files**: There are several .ipynb files in the .zip file. The one with student ID is the final submission, while others are complex models that are discussed in this paper.

2 Preprocessing

2.1 Feature Engineering

Some feature engineering concept in this section refer to *Multimodal Deep Learning Framework for Image Popularity Prediction on Social Media (Fatma A 2021)*. Features can be categorize to **Text Features**, **Time Features**, **Categorical Features** and **Image Features**.

2.1.1 Text Features

In this section, three types of text feature are proposed : pre-trained embeddings, concept vector and manual features.

For pretrained embeddings, *'paraphrase-MiniLM-L6-v2'* are utilized to generate a 384-dimension embedding from text concatenated from *'Title'* and *'Alltags'*. SVD from *scikit-learn* is used to reduce dimension to 30. However, in the final submission, SVD is not utilized for a better performance.

As for concept vectors, in order to reduce dimensions, texts in *Concept* are parsed and handled by *word2vec*.

Manual features include *'tags_count'*, *'title_len'*, which is represent as numerical features.

2.1.2 Time Features

Predicting popularity highly correlates to time features, where popularity may be influenced by cyclical features or duration. Cyclical features are composed of 3 types of one-hot encoded features: *weekday*, *month* and *hour*, where hour is classified to *'morning'*, *'afternoon'*, *'evening'*, *'night'*. Duration feature is calculated by subtracting the minimum *PostDate* from the row's *PostDate*.

2.1.3 Categorical Features

Categorical Features includes *'Category', 'Subcategory', 'Concept', 'Uid'*, in which *'Category'* contains only eleven unique values and both train and test datasets have exact same values. Therefore, one-hot encoded can be properly used without the curse of dimension.

Meanwhile, *'Subcategory*, *Concept* and *Uid* contains a lot of unique values. In most trials, PCA reduction is adapted to handle the situation.

Another feature, which is the frequency of *Uid*, indicates the frequency of a user's posts, which slightly improves the performance.

In the final submission, datasets are trained and predicted individually according to *Category*, and *Subcategory* is one-hot encoded, and *Uid, Concept* is not included due to bad performance. It overall yields a better accuracy.

2.2 Normalization

2.2.1 Numeric Features

All numeric columns are normalized to z-score. Mean and standard deviation are calculated only on train data to prevent data leakage; *X_test* data is normalized by the mean and std of *X_train*.

2.2.2 Images

Images are resized to (3, 128, 128) considering computing power, while some images that contain only one channel are extended to three by duplication.

3 Models

In this section, three different models are discussed: *Multimodal with Linear Fusion Layer*, *Multimodal with LSTM Fusion Layer*, *Catboost with pretrained Resnet50 image features extraction*.

3.1 Multimodal with Linear Fusion Layer

Multimodal is construct with *MLP_branch* and *Image_branch*, where *image branch* utilized CNN implemented by *Pytorch*. Lastly, a fusion layer with multiple linear layers and activation layers dealt with the concatenate features output by *MLP_branch* and *Image_branch*. However, the result is not ideal. One possible reason to the unfavorable result is dimension curses and lack of construction knowledge.

3.2 Multimodal with LSTM Fusion Layer

This model is similar to the one in the last section, but time sequences are created according to *time_steps* and the fusion layer is a LSTM layer. However, this complicated method generates a result that is worse than the previous one. Probably due to that the data is not continuous on the dimension of time, and that the popularity is not highly depending on cyclical or time features.

3.3 Catboost with pre-trained Resnet50 image features extraction

Lastly, the model mainly base on *catboost* is introduced. Instead of back prop-

agating on image features, in this method, a pre-trained *resnet50* is used to generate image features. To avoid out of memory errors, image features are reduced to 50 dimensions by SVD. Next, concatenate numerical features and image features and treat as an input of catboost.

3.4 Performance & Conclusion

The performance of the methods, though very time-consuming, is not ideal. Most of the ideas come from two papers (See References): *Multi-feature Fusion for Predicting Social Media Popularity* and *Multimodal Deep Learning Framework for Image Popularity Prediction on Social Media*. Though the paper generates great performances, but the natural differences of datasets are ignored and it led to the bad performance in my work.

Furthermore, tuning stage are limited to computing power. Though Pytorch based network tuning implementation is already well written, it is useless due to memory and time issues.

References

[1] Abousaleh, F., Cheng, W., Yu, N., Tsao, Y., Senior Member, IEEE. (2021) *Multimodal Deep Learning Framework for Image Popularity Prediction on Social Media*.

[2] Lv, J., Liu, W., Zhang, M., Gong, H., Wu, B., Ma, H. (2017) *Multi-feature Fusion for Predicting Social Media Popularity*.