Advertising Video Analysis

BA890 Final Research

- **Name:** Simeng Li
- **Email:** simone@bu.edu
- **BU ID:** U01878286

Write in Front

1. Colab Link:

https://colab.research.google.com/drive/1qepZI_rv1MDq3wX_r-59yqQCGR0mLyOl?usp=sharing

2. Google Drive Link:

Running DeepFM is out of memory in the colab chunk, I have trained the model in the local Pycharm IDE and saved the DeepFM model and the environment into Google Drive.

https://drive.google.com/drive/folders/1n_v8mlqWkoogXYiKTCdTAFc1XPQcgLWr?usp=sharing

3. Environment to Local Pycharm IDE:

- Python 3.7.0
- Tensorflow 2.8.0

Initial Proposal

1. Background & Motivation

A good advertisement can attract customers to click, convert and even pay. So what makes a good advertisement? It should be an advertisement with a high click rate and a high ROI. However, making a good ad is pretty hard and unpredictable, and even the people who create these successful ads don't know why. How can we identify good ads, enlarge the budget, and win higher user recovery? At the same time, how can we identify ineffective ads and save the company's budget? Based on this business problem, I want to use the machine learning method to learn the complex logical structure of advertisements, and the final result will be to realize the binary classification of excellent advertisements.

2. Data Introduction

The dataset is from online video advertisements. These videos are dedicated to attracting gamers. It includes 16 columns and 8980 rows.

Link: https://github.com/SimengLi1998/Data/blob/main/ori_data.txt

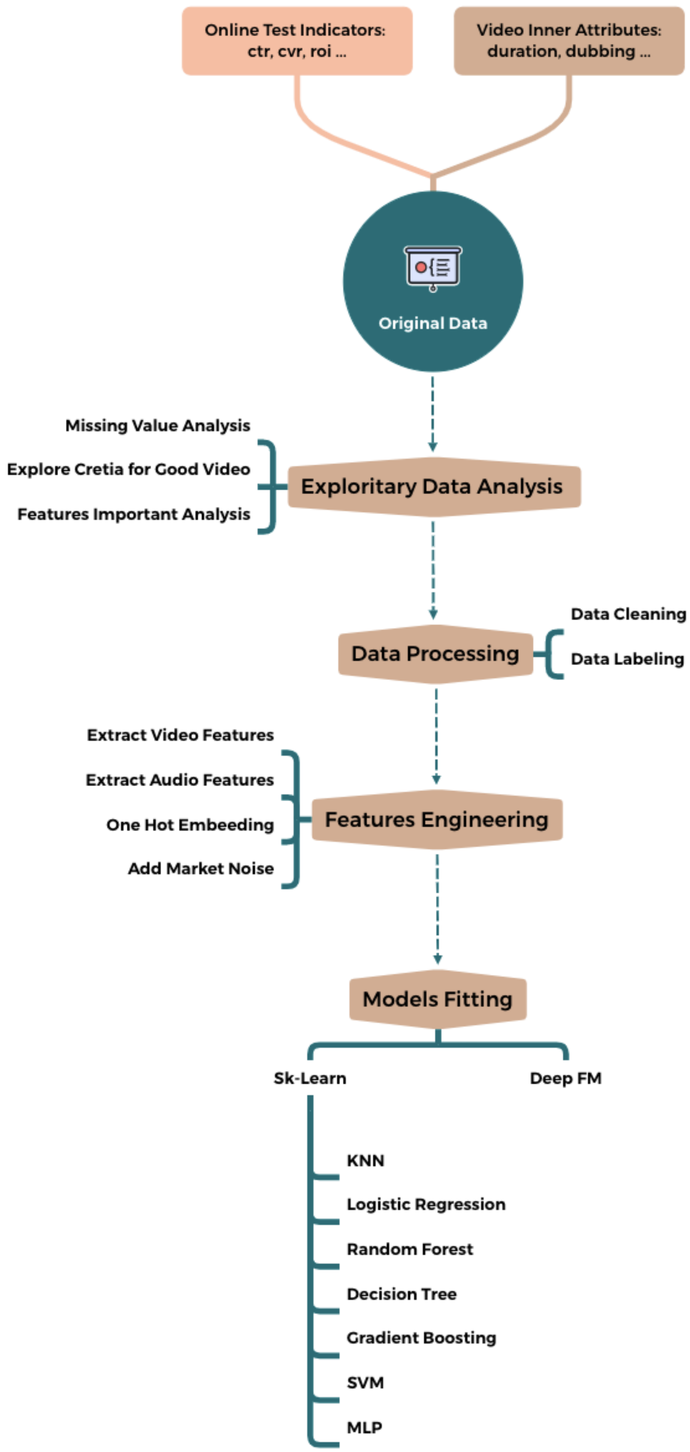
Video intrinsic features:	**Video effects features:**
• Material_id: video id • Label_ids: element ids used in the video • V_resolution_id: resolution of the vedio • Duration: length of the vedio • r_frame_rate: video frame rate • Display_type: view in landscape or portrait • Path: video path	• Avg_cost: the average cost per serving • Ctr: the average click rate • Cvr: the average converted rate • Ecpm: effective cost per mille • Deep_convert_rate: deep converted rate • Pay_rate: the average pay rate • First_day_roi: first day ROI • Valid_play_rate: the rate of successfully played videos • Finish_play_rate: the rate of successfully finished viewing

3. Steps & Milestones

- **Manual labeling**:
  - According to the Video effects features, find out the reasonable evaluation criteria for good video, above the standard is "1", below the standard is "0"
- **Extract video features**:
  - Download the video, and extract the key frame information and audio features in the video;
- **Sklearn models:**
  - Run the baseline by using Randomforest / KNN / LogisticRgression / SVM / MLP classifiers ;
- **Explore better classifiers:**
  - Try to fit the data with Wide&Deep model and DeepFM model to improve the classification.

Final Research

1. Work Flow Overview



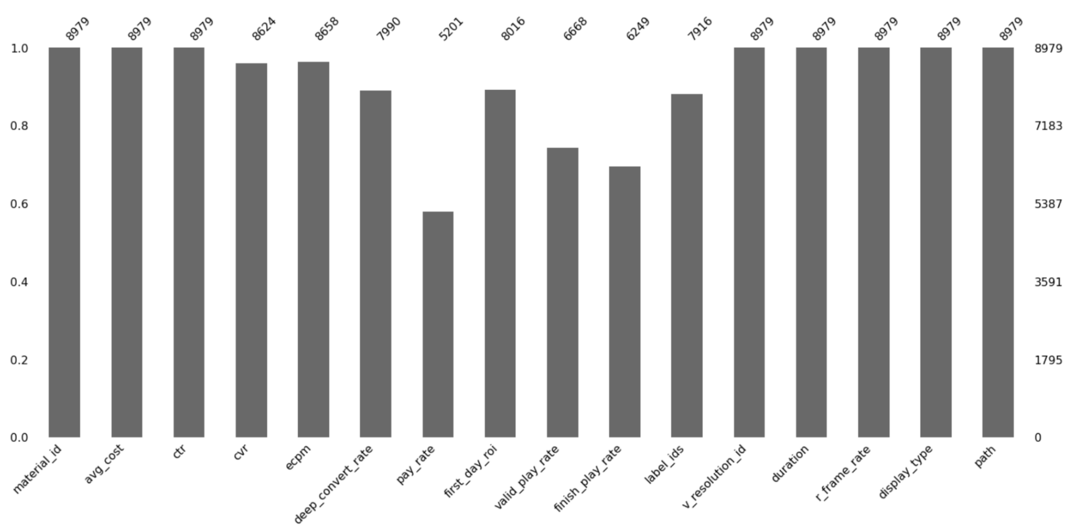
2. Exploratory Data Analysis (EDA)

2.1 Missing Values Analysis

The original data is including 16 columns and 8980 rows. However, not all of those 16 features need to be put into the model. Here I divided those 16 variables into Online test indicators and Video inner attributes. Missing values should be handled differently for different types of features.

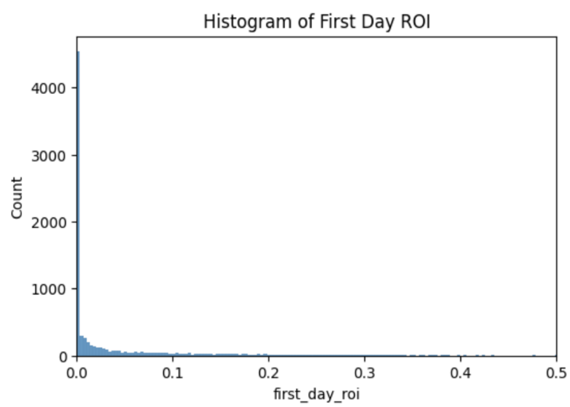
- **Online test indicators:**
  The goal of this research is to find a way to evaluate the video material without exposure. Therefore, the online test indicators like ctr, cvr, ecpm and etc should not be considered predictors. Later, in the data processing part, I will put them into ignoring columns.
  IGNORE_COLS = ['index', 'avg_cost', 'ctr', 'cvr', 'ecpm', 'deep_convert_rate', 'pay_rate', 'first_day_roi', 'valid_play_rate', 'finish_play_rate', 'label']

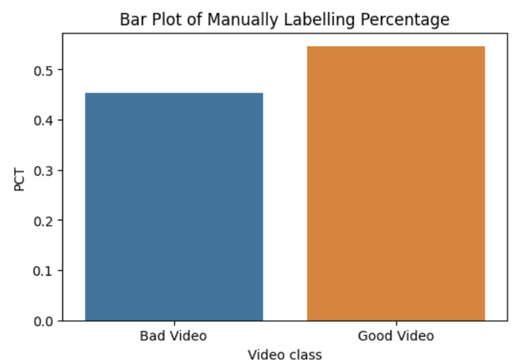
- **Video inner attributes:**
  When a video material is designed, it contains the inherent properties of the video, such as which game character elements are used, whether the screen is horizontal or vertical, video duration, dubbing, etc. These important factors are the first impression that attracts the audience and also lead to whether users convert and pay. Therefore, these missing values need to be taken seriously and dealt with one by one in the subsequent data processing.



2.2 Explore Criteria for Good Video

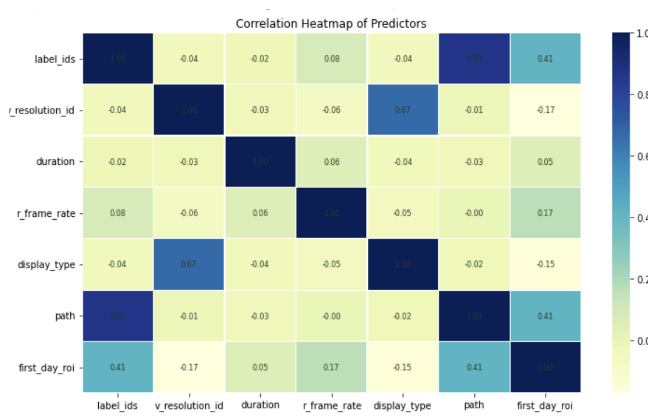
Intuitively, videos that generate value (first_day_roi is not zero) are good materials. Therefore, I drew the distribution map of first_day_roi, and found that there were 4075 observations with zero or null first_day_roi, accounting for about 50% of the total 8980 observations. In the later data processing process, I define the ROI of zero on the first day as bad video material, and label it as "0", otherwise it is a good material, and label it as "1".



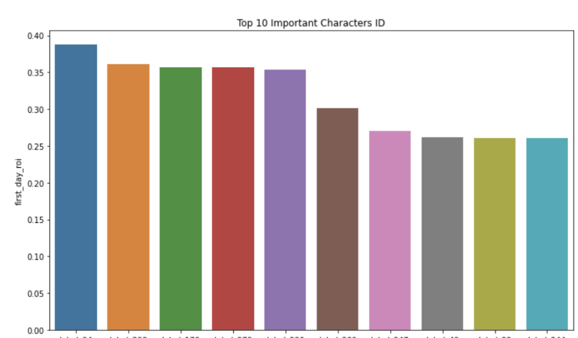


2.3 Features Importance Analysis

By plotting the correlation heatmap, we can see the most important feature of the ROI on the first day are label_ids and path. label_ids is the IDs of key characters, key props, and weapons in the game world. \`path\` is the http link directory to the video. Later, I will further extract the video features and audio features from the path. On the other hand, the least correlation with the first day roi is duration which is the length of the video.



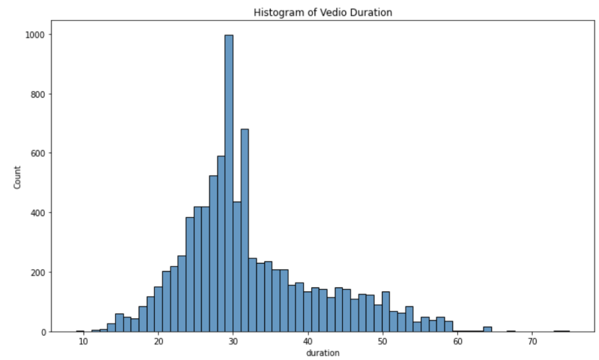
In order to interpret the data and discover the direction of data processing, I further analyzed the label_ids and duration.



- **\`label_ids\`:**
  It shows that the 94th index of the character ID has the greatest positive

effect on the ROI. The top 10 most important character ID are plotted in the chart. It could interpret as there are specific characters that attract players to click on video ads. This finding inspires suggestions to add more of these important roles in the video to attract players.

- **`duration`:**
  I plotted the distribution of duration, where the mean is located in 30 seconds. Considering that the effective play rate and completion rate are very low, in the following data processing, it is not necessary to extract the information of the entire video. Combined with the video duration distribution plot, it may be a more reasonable choice to analyze 30 seconds (if less than 30 seconds, then analyze the entire video).



3. Data Processing & Engineering

3.1 Process 'path'

- Download videos
- Extract video fatures
  - Only capture 30 seconds of the video (if the duration is less then 30 seconds, then process the whole video).
  - In order to keep the amount of columns consistent, ten frames of videos are taken on average.
  - Use Image modual to extract image information from the 10 frames.
- Remove videos
- Extract audio information
- Combine the video features and audio features together

3.2 One Hot Embedding

- ‘label_ids’ is categorical data which saved in a list for each video. First, find the number of unique charactor IDs is 325. Second, turn the categorical data into numarical one by using one hot embedding.

3.3 Sampling

- For DeepFM model: Split the processed data into 80% tranning and 20% testing.
- For Sklearn models: Split the processed data into 70% tranning and 30% testing.

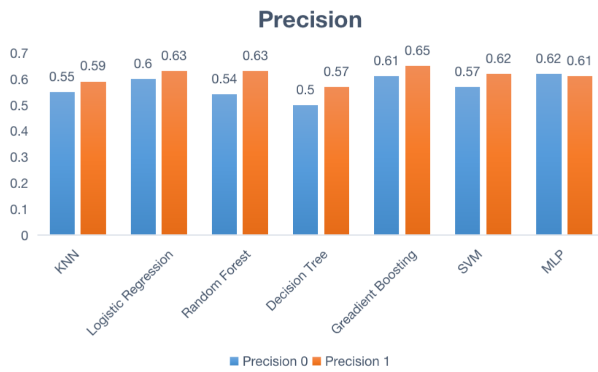
4. Models Fitting

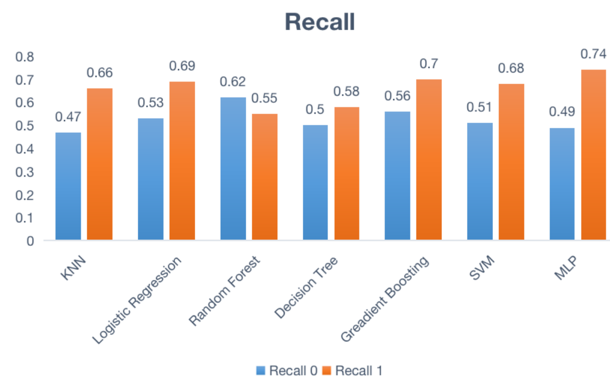
4.1 Sklearn Models

- **Models Introduction:**
  Use sklearn modual to quickly fit processed data into 7 binarial classifiers which are 'KNN', 'LogisticRegression', 'RandomForest', 'DecisionTree', 'GradientBoosting', 'SVM' and 'MLP'. They can predict the effectiveness of the advertisement videos based on the observed predictors. If the model predict that the video is 1, then it means the first day ROI of this advertisement is not zero, vices versa. I have colleced the performance of the models as the baseline. In the next step, I also tried to fit the data into DeepFM model to see if there is a significant improvement in videos classification.

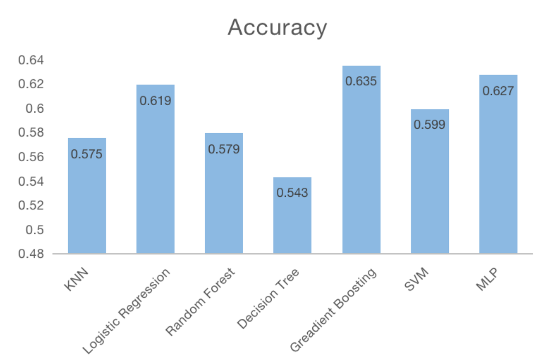
- **Models Evaluation & Comparasion:**
  - Models Evaluation:
    I evaluated the performance of the model by using confusion matrix. Additionally, I comparaed accuracy, precision and recall of the 7 models. Precision captures the number of instances that are relevant, out of the total instances the model retrieved. So it expects a hign precision on good videos. Recall captures the number of instances which the model correctly identified as relevant out of the total relevant instances. In the business scenario, we do not want the bad videos waste money, in other words, we do not wanna miss any videos that are really bad. Therefore, it expects a hign recall on bad videos.
    - Precision = TruePositives / (TruePositives + FalsePositives)

- Recall = TruePositives / (TruePositives + FalseNegatives)





- Models Comparasion:
  Consider the 2 main business scenarios, one is to identify the good videos which could make profits, another is to identify the none profit vifeos which only cost money but no return. Therefore, I will focus on the overall accuracy of the model and the precision on targeting the good videos(target 1), and the recall on targeting the bad videos(target 0).



  - Gradient Boosting has the highest accuracy at the level of 63.5%.
  - Gradient Boosting also has the highest precision on predicting good videos which at the level of 65%.
  - Random Forest has the highest recall on predicting bad videos which at the level of 62%.

- **Models Conclusion:**
  - Gradient Boosting has the best performance on capturing good videos.
  - Random Forest has the best performance on capturing bad videos

4.2 DeepFM

- **Model Introduction:**
  DeepFM is known as a Deep CTR model which is improved from Wide&Deep

model. It is based on Google's paper Wide&Deep learning, by replacing the wide part-LR part of the original paper with FM , thereby improving the shortcomings of the original model still requiring artificial feature engineering, and obtaining an end-to-end deep learning model.

- Wide for Memorization: the wide side remembers those common, high-frequency patterns in historical data. Input valuable and obvious features and feature combinations into the wide side based on human experience and business background.
- Deep for Generation: the deep side transforms the tag into a vector through embedding, and changes the exact match of the tag into a fuzzy query of the tag vector, so the model has a good "expansion" ability.

- **Models Evaluation:**
  - The overall accuracy is 61.7%.
  - Precision on predicting good videos is at the level of 67%.
  - Recall on predicting bad videos is at the level of 64%.
- **Model Conclusion:**
  - According to the evaluation indicators of accuracy, precision and recall, we can drwa the conclusion that DeepFM is slightly better than Gradient Bosst and Random Forest.
  - Model is apparently overfitting, for the next steps, it is needed to adjust the parameters of the model. For example, I ca try to reduce the number of hidden layers and the units in each layer.

```
deepfm
epoch: 24, train loss: 0.53536594, train acc: 0.8457469024084644, test loss: 2.7412233, test acc: 0.56681514476147 *
epoch: 25, train loss: 1.1732486, train acc: 0.7437004037310316, test loss: 3.0126407, test acc: 0.573496659242761 7 *
epoch: 26, train loss: 1.2490131, train acc: 0.7531672003341222, test loss: 2.357466, test acc: 0.585746102449887 *
epoch: 27, train loss: 0.6456675, train acc: 0.8425448976750661, test loss: 1.8440367, test acc: 0.609688195991 0914 *
epoch: 28, train loss: 0.27641314, train acc: 0.9018515940414868, test loss: 2.2381976, test acc: 0.568485523385 3007
epoch: 29, train loss: 0.4595307, train acc: 0.8461645551997773, test loss: 2.7826922, test acc: 0.542316258351 8931
epoch: 30, train loss: 0.6960801, train acc: 0.7929834331059445, test loss: 2.708365, test acc: 0.547884187082 4054
epoch: 31, train loss: 0.57185364, train acc: 0.8180426005847139, test loss: 2.2252054, test acc: 0.574610244988 8641
epoch: 32, train loss: 0.27830684, train acc: 0.8957260197688988, test loss: 1.8778979, test acc: 0.605790645879 7327
epoch: 33, train loss: 0.113279894, train acc: 0.9604622024223862, test loss: 1.8729346, test acc: 0.612472160356 3474 *
epoch: 34, train loss: 0.08723585, train acc: 0.9713211749965196, test loss: 2.0880046, test acc: 0.59409799554 5657
epoch: 35, train loss: 0.12243229, train acc: 0.9614367256021161, test loss: 2.3036017, test acc: 0.591870824053 4521
epoch: 36, train loss: 0.15356313, train acc: 0.9498816650424614, test loss: 2.3757284, test acc: 0.590200445434 2984
epoch: 37, train loss: 0.13574731, train acc: 0.9551719337324238, test loss: 2.2763655, test acc: 0.587416481069 0423
epoch: 38, train loss: 0.0847929, train acc: 0.9728525685646666, test loss: 2.0792258, test acc: 0.591314031180 4009
epoch: 39, train loss: 0.047279585, train acc: 0.9859390226924684, test loss: 1.8689014, test acc: 0.585189309576 8375
epoch: 40, train loss: 0.029877093, train acc: 0.9901155506055965, test loss: 1.6884748, test acc: 0.598552338530 0668
epoch: 41, train loss: 0.021468306, train acc: 0.9926214673534735, test loss: 1.5593244, test acc: 0.595211581291 7594
epoch: 42, train loss: 0.018829653, train acc: 0.9938744257274119, test loss: 1.4855702, test acc: 0.599665924276 1693
epoch: 43, train loss: 0.020466499, train acc: 0.9934567729360991, test loss: 1.4597198, test acc: 0.594654788418 7082
epoch: 44, train loss: 0.024159312, train acc: 0.9920645969650563, test loss: 1.4661441, test acc: 0.587416481069 0423
epoch: 45, train loss: 0.028061703, train acc: 0.9912292913824308, test loss: 1.4840909, test acc: 0.586302895322 9399
epoch: 46, train loss: 0.030448772, train acc: 0.989837115411388, test loss: 1.4975023, test acc: 0.584075723830 7349
epoch: 47, train loss: 0.030023621, train acc: 0.9894194626200752, test loss: 1.4990778, test acc: 0.583518930957 6837
No optimization for a long time, auto early stopping...
exporting...
```

```
accuracy is 0.617483

              precision    recall  f1-score   support

           0       0.57      0.64      0.60       806
           1       0.67      0.60      0.63       990

    accuracy                           0.62      1796
   macro avg       0.62      0.62      0.62      1796
weighted avg       0.62      0.62      0.62      1796

Confusion Matrix...
[[516 290]
 [397 593]]
```

5. Conclusions & Limitations

5.1 Conclusions

- All milestones mentioned in the proposal are completed in this final project.
- I have used 8 models to fit the processed advertisement video data. DeepFM is slightly better than others. Those advertising video materials whose first-day online return is not zero can be identified at the best level of 67% precision.
- Two main application business scenarios for the model: one is to identify the good videos which could make profits; another is to identify the none profit vifeos which only cost money but no return.
- According to EDA, we can find the the role(the label_ids column) in the game is highly correlate to the first day ROI. It could interpret as there are specific charactors more attract players to click on video ads. This finding inspires suggestions to add more of these important roles in video to attract players.

5.2 Limitations and Future Steps

- I only considered the first_day_roi as the cretia of the good vedio. However, first_day_roi is not zero could not because of the video is designed well but could be interpreted as luck when ctr , cvr and costis low. Therefore, it is not rigorous to only look at one indicator here. It is necessary to combine multiple indicators to evaluate the quality of the material, and at the same time, it needs to be combined with the commercial intuition of advertising.
- In the process of labeling the quality of the video, the completion play rate is not considered. The observation data for training might be not valid, because some people might do not watch it because they cannot open it valid. This kind of video cannot be considered a bad video. In the next step, before data processing, it is better to remove these ambiguous data and make the results of the model more reliable. This is also one of the possible reasons why DeepFM has not improved significantly.
- DeepFM is apparently overfitting, for the next steps, it is needed to adjust the parameters of the model. For example, I can try to reduce the number of hidden layers and the units in each layer.
- When extracting information from the video, I only extract ten frames evenly. If I extract more video frames, will it be more conducive to classifier training? The next step can continue to verify this view.

6. References

1. Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (*IJCAI'17*). AAAI Press, 1725–1731.

2. Princewen. (2019, June 24). deepfm-stepbystep. GitHub. Retrieved May 1, 2022, from https://github.com/princewen/tensorflow_practice/blob/master/recommendation/Basic-DeepFM-model/DeepFM-StepByStep.ipynb

3. Shafi, A. (2022, April 19). How to learn the definitions of precision and recall (for good). Medium. Retrieved May 1, 2022, from https://towardsdatascience.com/precision-and-recall-88a3776c8007#:~:text=Precision%20is%20calculated%20by%20dividing,was%20predicted%20as%20a%20positive.&text=Recall%20(or%20True%20Positive%20Rate,have%20been%20predicted%20as%20positive.