Data Mining Final Project Report

News Recommendation Prediction

Group 29, 109350008 張詠哲, 110705013 沈昱宏, 110550014 吳權祐

Q1. Describe how you solve this problem. Details include preprocessing, embeddings, model selection, hyperparameters should be provided.

1. **Preprocessing & Embedding**

   We utilized category, subcategory, title_entity, and abstract_entity as our features. Our preprocessing method can be divided into two parts. First, we counted the occurrences of each category and subcategory in clicked_news and impressions (only the one for prediction) respectively, and concatenated these counts. The second part involved calculating two similarity scores between title_entity and between abstract_entity for clicked_news and impression respectively. We used the embeddings provided by the TA to compute the cosine similarity and used the mean of these similarities as the score.

   The training data dimensions is twice the number of unique categories + twice the number of unique subcategories + 2 (two similarity scores), resulting in a total dimension of 608. We employed supervised learning methods with labels indicating whether an impression was clicked or not.

2. **Hyperparameter**

   This is the figure about hyperparameters of our model.

```
model = CatBoostClassifier(iterations=1000, depth=5, learning_rate=0.1, loss_function='Logloss',
                           eval_metric='AUC')
```

3. **Model selection**

   We have tried several different models, such as XGBoostClassifier, CatBoostClassifier, and some deep learning based models. At the same time, we used sklearn.train_test_split to get 20% of the training data as validation data. We then selected the model and preprocessing method with the best validation result as our final approach.

Q2. Choose a variable excluding hyperparameters and compare their performance. Explain what causes the difference of performance or why.

Except for the final method mentioned in Q1. In this assignment, we experiment 5 attempts, which contain 2 different embedding pretrain (All-MinLM-L6-V2, mxbai-embed-large-v1), 3 models (CatBoost, XgBoost, and Neural Network). Following table shows the experiment and the final method result on Kaggle public score.
First of all, we use the given entity embedding given by TAs, and use CatBoost to predict the values, and it turns out that it performs poorly. Secondly, we use the embedding in

All-MinLM (384 dimensional vector) with a simple nn (1 multihead attention layers + 3 linear blocks and + ReLU). It generates a lower training error (0.3) compared to the other methods (~0.35), but does not perform better on the validation dataset. We tried adding weight decay and dropout layers and still did not help. After that, we tried using a different embedding, which is a larger model called mxbai. It seems that mxbai can extract better features, making the model perform better.

	Kaggle public score (AUC)
Given_Entity_Embedding + CatBoost	0.5001
All-MinLM + NN	0.5652
All-MinLM + XgBoost	0.5482
mxbai + CatBoost	0.6654
Final method (details in Q1)	0.6664

As the above table shows that the performance from high to low is our final method, mxbai + CatBoost, All-MinLM + NN, All-MinLM + XgBoost, Given_Entity_Embedding + CatBoost respectively. It can be discussed in two parts: embedding and model aspect.

Embedding

For the different embedding methods, the mxbai performance the best, second is the All-MinLM, and third is the given embedding. There are some explanation:

The mxbai embedding is derived from a larger model, which is more powerful than All-MinLM and the given embedding on extracting relevant and useful features from the news title or the abstract, and more likely provides a richer and more useful representation of the data. And this embedding combined with CatBoost yielded a better performance (AUC = 0.6654). As for All-MinLM, although it's more resource friendly and captures more detailed features compared to the given entity embedding, it's still not strong enough on this task. Because we don't have the implementation details on the given entity embedding, we can't give some reasonable explanation.

For the final method we use, we add the statistical amount of subcategory and category for each id on mxbai embedding. It might provide additional context, enhancing the model's ability to make accurate predictions.

Model

For the NN we use 1 multihead attention layer + 3 linear blocks + ReLU. Despite achieving a lower training error, it did not generalize well to the test dataset, indicating overfitting. As for the CatBoost, while it performed poorly with the given entity embedding (AUC = 0.5001), it performed well with the mxbai embedding (AUC = 0.6654) and even better with the final method (AUC = 0.6664), which indicates that the model can leverage high-quality embeddings to improve prediction accuracy. As for XgBoost, it's known for robustness and

efficiency, performing better than the given entity embedding but worse than CatBoost with the mxbai embedding. This indicates that while XgBoost is also a good choice, it might not be more suitable than CatBoost on this task.

Q3. Do some error analysis or case study. Is there anything worth mentioning while checking the mispredicted data? Share with us.

We have observed several mispredicted data in our validation dataset and we will discuss the cases using the following behavior data.

behavior: `0 U1349561 11/11/2019 9:31:08 AM`

clicked_news:

news_id	category	subcategory	title_labels	abstract_labels
N410559	sports	foorball_nfl	"New England Patriots" "National Football League"	"New England Patriots" "National Football League" "Tom Brady"
N109405	sports	football_nfl_videos	-	"Good Morning Football" "NFL Network"
N79284	sports	football_nfl	"National Football League"	"Houston Texans" "Kansas City Chiefs"
N812877	lifestyle	lifestyle_pets_animals	"Litter box"	-
N311012	sports	football_nfl	"Sports rating system" "National Football League"	"Sports rating system" "National Football League"
N362483	sports	football_nfl	-	-
N326916	sports	football_nfl	"Yardbarker" "National Football League"	"Yardbarker" "Dallas Cowboys"
N199698	sports	football_nfl	"National Football League"	"National Football League"
N148892	sports	football_nfl	"Vince Lombardi Trophy" "National Football League"	"Vince Lombardi Trophy" "National Football League"
N37277	sports	football_nfl	"Tom Brady"	"Tom Brady" "New England Patriots" "Baltimore Ravens" "WEEI (AM)"

N35289	sports	football_nfl	"Russell Wilson" "Adam Gase"	"Russell Wilson" "New York Jets" "Adam Gase" "Madison Square Garden"
N104848	sports	football_nfl	"Kellen Winslow II" "Chronic traumatic encephalopathy"	"Kellen Winslow II" "National Football League"
N759621	music	cma-awards	"Country music"	-

It shows that most of the clicked news of the user are related to football and NFL

Impressions (we only showed those mispredicted**):**

N881666-1 N409782-1

Case1:

```
news_id: N881666
category: food and drink
subcategory: recipes
title_labels: -
abstract_labels: -
predicted_prob: 0.0603
```

Case2:

```
news_id: N409782
category: sports
subcategory: football_nfl
title_labels: "NFL regular season"
abstract_labels: "2019 NFL season"
predicted_prob: 0.2867
```

For case 1. You can see that the user did click the news, but we predict a relatively low probability. There are two possible causes of this misprediction. Firstly, the user's history behavior might not represent his real interest. Although the clicked news is only mainly about the NFL, we can not tell whether the user will click the news or not. The user might be a chef that likes to watch NFL games, and he accidentally found a recipe he has never seen, so he clicked it. Secondly, we use the titles and abstract labels (with the embedding provided by TA) instead of news titles. However, there is no such label in this case. If the news is something like "food for football players", the user will probably click it, and if the news is "food for cats", the user will probably not click it. The design of our model does not know the content of the news, and only notices that it is related to food, so it is mispredicted.

For case 2. Although this news is about sports or the NFL, which match the preference of this user, it is still mispredicted. We think the reasonable explanation is that the title and the abstract on this news is not shown in the training data, causing the title embedding and the abstract embedding is an unseen case for the model. But the category and the subcategory is still about sports, so the statistical count on category and subcategory make the predicted probability higher than the case 1.