MATH/STAT 4450/8456 Machine Learning Competition #1

Predicting the repeat purchases for an online retail store

The first contest for this course is to predict the repeat orders during a 4-week period for an online store. You will receive the historical orders and item features information during the period between August 1, 2020 and January 31, 2021, then predict whether the users will repeated order the items during the four weeks of February.

Response variable to predict (the `target` column in `sample_submission.csv`):

- 0: no repeat orders during the 4-week period
- 1: Repeat orders in the first week (2/1 - 2/7)
- 2: Repeat orders in the second week (2/8 - 2/14)
- 3: Repeat orders in the third week (2/15 - 2/21)
- 4: Repeat orders in the fourth week (2/22 - 2/28)

Kaggle website

The data can be downloaded from the Kaggle site and you will have to participate the competition through the link (https://www.kaggle.com/t/5d596c7db4d54fde92c90e2446a339d7) as it is not open to public.

- Please merge to teams before making submissions.
- The maximum daily submission number is 15. So you will need to wait until the next UTC day after submitting 15 results.

Evaluation metrics

Actual evaluation for the final submission

- If the item is correctly identified as a repeated purchase in the 4-week period (1-4) or no repeat order (0), then you receive 0.0001 points.
- If the correct week is predicted, then you receive 0.0003 points.
- The total points you received will be the final accuracy score (between 0 and 3).

	Actual Class				
Predicted Class	**0**	**1**	**2**	**3**	**4**
0	0.0001	0	0	0	0
1	0	0.0003	0.0001	0.0001	0.0001
2	0	0.0001	0.0003	0.0001	0.0001
3	0	0.0001	0.0001	0.0003	0.0001
4	0	0.0001	0.0001	0.0001	0.0003

Kaggle evaluation

- The closest Kaggle evaluation metric is Weighted Categorization Accuracy, which is given as follows. So your Kaggle score will be lower than the actual score.

	Actual Class				
Predicted Class	**0**	**1**	**2**	**3**	**4**
0	0.0001	0	0	0	0
1	0	0.0003	0	0	0
2	0	0	0.0003	0	0
3	0	0	0	0.0003	0
4	0	0	0	0	0.0003

Description of datasets and variables

1. `orders.csv` contains all the online purchase information between August 1, 2020 and January 31, 2021. It has four columns:
    - `date`: The date when the order was placed.
    - `userID`: A non-negative integer that indicates the user unique ID.
    - `itemID`: A non-negative integer that indicates the item unique ID.
    - `count`: The number of items purchased in this order.
2. `items.csv` contains the item description information. It has eight columns:
    - `itemID`: A non-negative integer that indicates the item unique ID.
    - `manufacturerID`: A non-negative integer that indicates the manufacturer unique ID.
    - `f1`: Feature 1. Categorical. -1 represents missing value.
    - `f2`: Feature 2. Categorical. -1 represents missing value.
    - `f3`: Feature 3. Categorical. -1 represents missing value.
    - `f4`: Feature 4. Categorical. -1 represents missing value.
    - `f5`: Feature 5. Categorical. -1 represents missing value.
    - `category`: A list of categories associated with the item. Missing values exist.
3. `categories.csv` contains the tree-shaped hierarchy of the categories. It has two columns:
    - `category`: A non-negative integer that indicates the category unique ID.
    - `parent_category`: The parent category ID.
4. `test.csv` is the data set used to make predictions. It contains two columns:
    - `userID`: User unique ID. All user IDs exist in the `orders.csv` file.
    - `itemID`: Item unique ID. All item IDs exist in the `orders.csv` file.
    - `ID`: A merged column of `userID` and `itemID` for Kaggle submission.

Overview of data

Here is a quick look at the data.

```r
orders = read.csv("orders.csv")
orders$date = as.Date(orders$date)
str(orders)
```

```
## 'data.frame':    816574 obs. of  4 variables:
##  $ date  : Date, format: "2020-08-01" "2020-08-01" ...
##  $ userID: int  14477 14477 33883 33883 33883 33883 19087 42266 4286 4286 ...
##  $ itemID: int  2375 14961 8927 27830 29700 27432 9798 18630 31949 10468 ...
##  $ count : int  1 2 1 2 1 4 1 3 1 1 ...
```

```r
items = read.csv("items.csv")
str(items)
```

```
## 'data.frame':    32776 obs. of  8 variables:
##  $ itemID        : int  22665 28640 13526 21399 8504 32122 31956 6237 16971 18385 ...
##  $ manufacturerID: int  861 1366 1090 1090 768 5 1388 1492 288 288 ...
##  $ f1            : int  4 10 10 10 4 4 4 4 6 6 ...
##  $ f2            : int  0 1 0 1 1 1 0 1 0 0 ...
##  $ f3            : int  490 537 511 511 484 491 491 491 314 314 ...
##  $ f4            : int  2 0 0 0 0 0 0 3 0 0 ...
##  $ f5            : int  66 101 0 0 66 66 66 66 45 45 ...
##  $ category      : chr  "[2890, 855, 3908, 3909]" "" "[3270, 163, 284, 1694, 12, 3837, 2422, 3595, 35
```

```r
categories = read.csv("categories.csv")
str(categories)
```

```
## 'data.frame':    4332 obs. of  2 variables:
##  $ category       : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ parent_category: int  75 1499 1082 3498 1623 2478 1582 3027 2364 3590 ...
```

```r
test = read.csv("test.csv")
str(test)
```

```
## 'data.frame':    10000 obs. of  3 variables:
##  $ userID: int  0 0 13 15 15 20 24 34 34 46 ...
##  $ itemID: int  20664 28231 2690 1299 20968 8272 11340 21146 31244 31083 ...
##  $ ID    : chr  "0-20664" "0-28231" "13-2690" "15-1299" ...
```

Task

1. Create the most accurate classifier that you can for the data, as measured by the test data.
2. Write a 6-12 page slides summarizing your approach to
    (a) cleaning and preparing the data for modeling,
    (b) formulating the model design matrix (input matrix / predictor space),
    (c) building the model and tuning parameters,
    (d) validating the model by training & validation sets, or other approaches,
    (e) comparing results from all attempts,
    (f) findings from the data and challenges from this contest.

Kaggle submission

Your Kaggle submission file should be in the csv format with two columns: `ID`, and `target`. Note that `ID` is a merged column for `userID` and `itemID`, separated by the dash sign (`-`). Your columns must be exactly same as the `sample_submission.csv` file. The `target` column should be replaced by your prediction. Example of the submission:

```
ID,target
0-20664,4
0-28231,1
13-2690,1
...
46130-395,1
```

Canvas submission

1. Code. The code you used for the BEST Kaggle submission (not for all submissions). Should be `.R`, `.Rmd`, `.py`, or `.ipynb` files. Make sure you include detailed explanation of the steps.
2. Slides for the presentation. Can be a `.pptx`, `.pdf`, or `.html` file.
3. Peer evaluation form.

Presentation

Two undergrad teams will be in a battle group. Four graduate teams will be paired based on their final submission results – the battling teams may have very similar error rates. All the audience will vote for the better presenter and one of the two teams who gets higher votes will receive extra points.

Peer evaluation

The within-team adjustment grade will be calculated based on the average evaluation from the team members. Everyone will rate each team member (including yourself) regarding her or his contribution to the team effort on the peer evaluation form. The total team effort includes: leadership, arranging and/or attending meetings, contributing creative ideas, coding, writing, presenting the results, and any other activities you feel are important for the success of the contest.

Deadlines:

- March 27 (11:59 pm): Final prediction submission on Kaggle.
- March 28 (in class): Presentation. 8 minutes per team.
- March 29 (11:59 pm): Slides, code, and peer evaluation submission.

Grading:

- Total points: 20 (+0.4)
  - Accuracy of classifier: 8
    - Score will be curved based on your total prediction error.
  - Progress made from multiple submissions: 3
    - Number of good submissions (highest accuracy in all previous submissions): 2
    - Amount of accuracy increase: 1
  - Presentation: 6 (+0.4)
    - Data exploration and visualization: 1
    - Data preparation and feature engineering: 1
    - Design matrix: 0.5
    - Validation approach: 1
    - Model selection and parameter tuning: 1

    - Result table for all models & parameters: 0.5
    - Summary and interesting findings: 1
    - Team battle: (+0.4)
  - Met the deadlines: 1
  - Within-team adjustment: 2

Teams (you can create your own team name)

- Undergraduate student teams

  1. Seh Na Mellick, Connor Moorhous
  2. Steven Horn, Wesley Phillipson

- Graduate student teams

  1. Jason Driscoll, Ghaith Al Saifi, Fred Hinsley
  2. Collin Dougherty, Veenamadhuri Dronadula, Michael Neman, Erica Pribil
  3. Akbota Aitbayeva, Sujith Manavalan, Ben Nolan
  4. Devashri Gandhi, Md Tahidul Islam, Timothy Reznicek