NN Recommendation System of Books - Using User and Book Title/ Description Embeddings

Imola Fodor, Information Retrieval 2023

Developed in R2023a.

Table of Contents

Ingest Data

The data is collected from Books Dataset at Kaggle (https://www.kaggle.com/datasets/saurabhbagchi/books-dataset) and from Goodreads found at https://github.com/sdhilip200/Content-Based-Recommendation---Good-Reads-data . The Kaggle set is taken to get mapping of book titles from user to rankings and from Goodreads to get book descriptions, for richer embeddings.

```
clear all
warning('off','all')

descr = readtable('data\book_decr_goodreads_sdhilip200.csv');
ids_books = readtable('data\kaggle_Books.csv');
descr.Properties.VariableNames
```

```
ans = 1×8 cell
'Var1'       'Desc'       'Unnamed_0'  'author'     'genre'      'image_link' 'r···
```

```
ids_books.Properties.VariableNames
```

```
ans = 1×8 cell
'ISBN'       'title'      'Book_Author''Year_Of_Publication''Publisher'  'Image_···
```

```
% First hand pre-processing
```

```
descr.title = lower(descr.title);
head(descr,3)
```

	Var1
0	{'We know that power is shifting: From West to East and North to South, from presidential palaces to pub
1	{'Following the success of The Accidental Billionaires and Moneyball comes Console Wars—a mesmerizing, b
2	{'How to tap the power of social software and networks to build your business In Trust Agents, two soci

```
ids_books.title = lower(ids_books.title);
head(ids_books,3)
```

	ISBN	title	Book_Author	Year_Of_Publication	Publisher
1	1.9515e+08	{'classical mythology' }	{'Mark P. O. Morford' }	2002	{'Oxford University
2	2.005e+06	{'clara callan' }	{'Richard Bruce Wright'}	2001	{'HarperFlamingo Ca
3	6.0973e+07	{'decision in normandy'}	{'Carlo D'Este' }	1991	{'HarperPerennial'

Basic Cleaning and Joining data

```
% Before joining tables, keeping unique pairs in descr table
[~,uidx] = unique(descr(:,8),'stable');
descr_without_dup = descr(uidx,:);
```

```
% Keeping only books that have descriptions in the Goodreads set
books = innerjoin(ids_books, descr_without_dup);
```

```
users_all = readtable('data\kaggle_Users.csv');
ratings_all = readtable('data\kaggle_Ratings.csv');

users_all.Properties.VariableNames
```

```
ans = 1×5 cell
'User_ID'    'City'       'State'      'Country'    'Age'
```

```
ratings_all.Properties.VariableNames
```

```
ans = 1×3 cell
'User_ID'    'ISBN'       'Book_Rating'
```

Originally the users.csv from Kaggle has one column Location with comma separated text for city, county, country. Text to Columns for Location and Cleaning for Country has been done in Excel manually.

```
% Any User for which the Country was eliminated, gets to be removed from
% the set.
users = rmmissing(users_all);
```

```
ratings_all = innerjoin(users, ratings_all);
ratings = rmmissing(ratings_all);
```

Final Join of 4 datasets

```
data =  innerjoin(ratings, books);
```

EDA and more cleaning

```
size(unique(data.User_ID),1)
```

```
ans = 11885
```

```
size(unique(data.ISBN),1)
```

```
ans = 2455
```

```
size(unique(data.title),1)
```

```
ans = 792
```

```
% Same title has more ISBN, hence a unique BookID column is created to
% substitute ISBN
data.BookID = grp2idx(data.title);
size(unique(data.BookID),1)
```

```
ans = 792
```

```
data.ISBN = [];
data.Unnamed_0 = [];
```

```
columns = data.Properties.VariableNames;
data = renamevars(data,["Book_Rating","rating"],["User_Rating","Overall_Rating"]);

min(data.Overall_Rating)
```

```
ans = 3.0800
```

```
max(data.Overall_Rating)
```

```
ans = 4.5700
```

The Overall Rating is renamed from rating, to get distinguished from the user rating. The range of this rating is small, hence it is considered to exclude from feature set.

```
unique(data.genre)
```

```
ans = 2×1 cell
'Business'
'Non-Fiction'
```

Only Business and Non-Fuction categories, even though by personal inspection of title, Fiction is indeed present.

```
unique(data.author)
```

```
ans = 485×1 cell
'A.A. Milne'
'A.S. Byatt'
'Aesop'
'Agatha Christie'
'Akio Morita'
'Al Ries'
'Alan Lightman'
'Alan Moore'
'Albert Camus'
'Aldous Huxley'
      ⋮
```

```
size(unique(data.User_ID),1)
```

```
ans = 11885
```

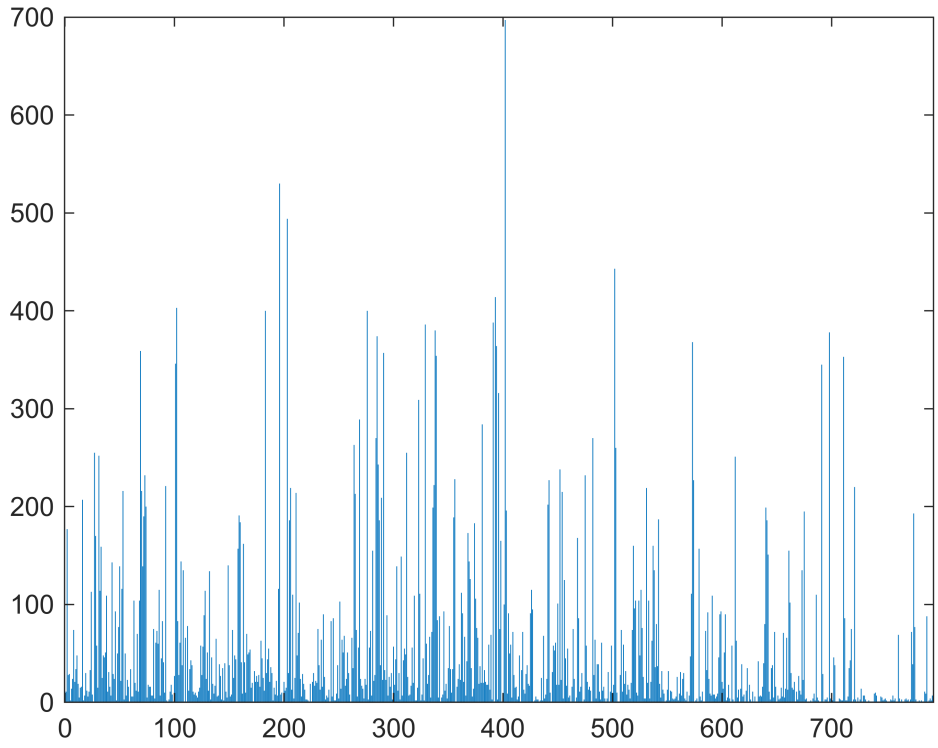
```
min(data.User_Rating)
```

```
ans = 0
```

```
max(data.User_Rating)
```

```
ans = 10
```

```
G = groupsummary(data,"BookID");
b = bar(G.BookID, G.GroupCount)
```




```
b =
  Bar with properties:
```

```
    BarLayout: 'grouped'
     BarWidth: 0.8000
    FaceColor: [0 0.4470 0.7410]
    EdgeColor: 'none'
    BaseValue: 0
        XData: [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 3
        YData: [11 177 28 29 2 14 24 74 21 34 48 19 5 15 16 207 2 7 30 12 6 11 33 113 8 1 255 170 58 15 252 114 159

  Show all properties
```

The User Rating is more promising as a feature. For the use-case, only higher ratings are considered, since we are modelling what the user likes.

Also, we exclude users who read less than 15 books.

```
% Keep only good ratings
data(find(data.User_Rating <5), :) = [];

% Keep only users who read at least 15 books
data = groupfilter(data,"User_ID",@(x) numel(x) > 14);
```

Feature Engineering

```
data.Age_Bin = zeros(size(data,1),1);
data.Age_Bin(find(data.Age<=18),:) = 1;
data.Age_Bin(find(data.Age>18 & data.Age<=36),:) = 2;
data.Age_Bin(find(data.Age>36 & data.Age<=55),:) = 3;
data.Age_Bin(find(data.Age>55),:) = 4;
```

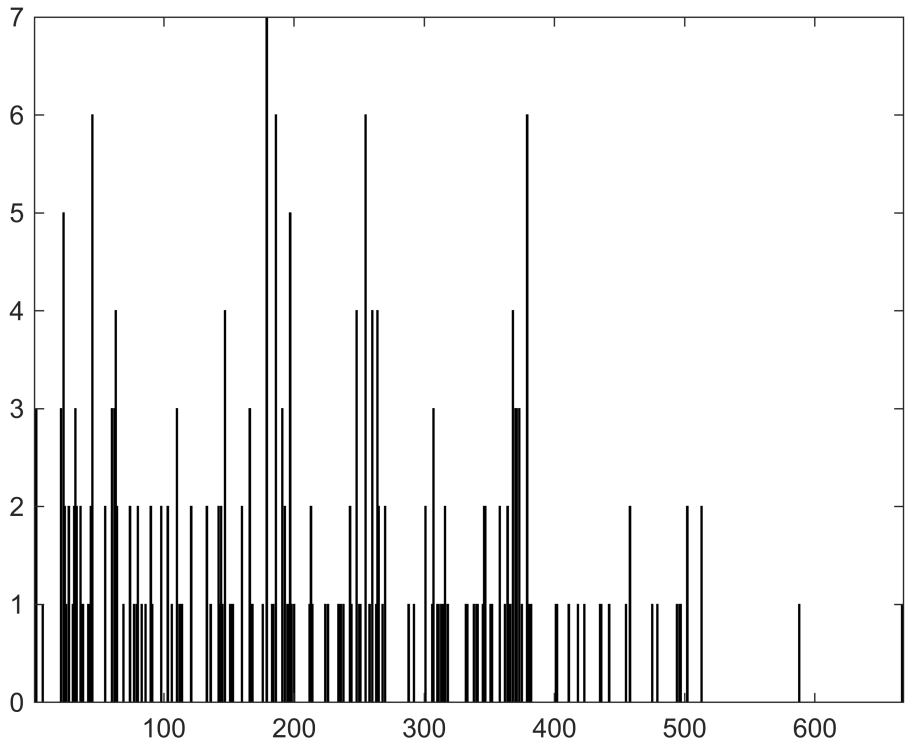
Feature Selection

```
nndata = data(:, ["User_ID", "Age_Bin","Country", "User_Rating", "BookID",
"title","Year_Of_Publication","Desc"]);
```

Separate Train/Test Data - 3 books per User set aside

```
% Manual separation in Excel choosing 3 ratings per user for test

nndataXRaw = readtable('nndataTrainRawText.csv');
nndataYRaw = readtable('nndataTestRawText.csv');
```

```
G = groupsummary(nndataYRaw,"BookID");
b = bar(G.BookID, G.GroupCount)
```




```
b =
  Bar with properties:

    BarLayout: 'grouped'
     BarWidth: 0.8000
    FaceColor: [0 0.4470 0.7410]
    EdgeColor: [0 0 0]
    BaseValue: 0
        XData: [2 7 21 23 24 25 27 30 31 32 33 36 37 38 42 44 45 55 60 62 63 64 69 74 77 79 80 83 86 90 91 98 103 10
        YData: [3 1 3 5 2 1 2 1 2 3 2 2 1 1 1 2 6 2 3 3 4 2 1 2 1 1 2 1 1 2 1 2 2 1 3 1 1 1 2 2 1 2 1 2 1 4 1 1 1 2

  Show all properties
```

Custom Word Embedding using Titles

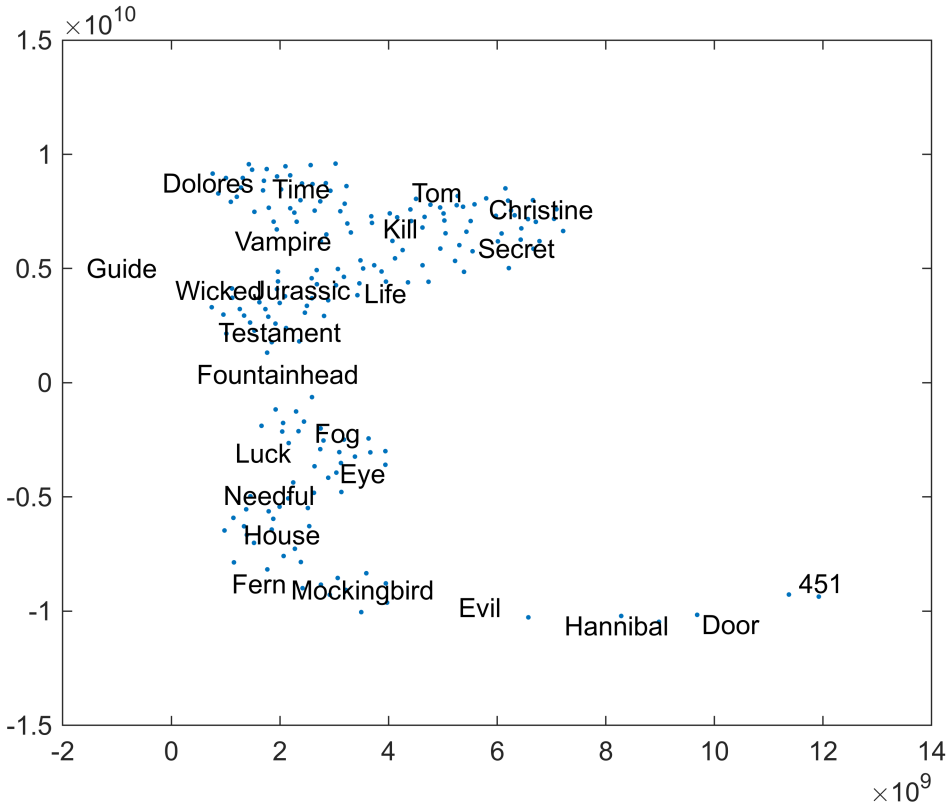
```
% Prepare dataset for custom word2vec
desc_joined_str = strjoin(string(nndataXRaw.title));

textData = split(desc_joined_str,newline);
documents = tokenizedDocument(textData);

% Some house-keeping
documents = erasePunctuation(documents);
documents= removeStopWords(documents);

% No custom hyperparameter chosen, skipgram approach by default
emb = trainWordEmbedding(documents)
```

```
Training: 100% Loss: 3.73587  Remaining time: 0 hours 0 minutes.
```

```
emb =
  wordEmbedding with properties:

     Dimension: 100
    Vocabulary: ["House"    "Life"    "Kill"    "Time"    "Good"    "Red"    "Girl"    "Guide"    "Things"    "Green
```

TSNE Visual

```
% Visualize embedding
words = emb.Vocabulary;
V = word2vec(emb, words);
XY = tsne(V);
textscatter(XY,words)
```



X : Use trained embedding for first round encoding + Age feature

```
[G,user] = findgroups(nndataXRaw.User_ID);
anonFunc = @(optimValues, ~) avg_embedding(emb, optimValues);
[title_emb, age_bin] = splitapply(anonFunc,[nndataXRaw.title,
num2cell(nndataXRaw.Age_Bin)],G);
mean_vec_title = double(title_emb);
nndataX = [user age_bin mean_vec_title];
```

```
[G,user] = findgroups(nndataYRaw.User_ID);
[bookID1, bookID2, bookID3] = splitapply(@getBooks,nndataYRaw.BookID,G);
```

```
nndataY = [user bookID1 bookID2 bookID3];
```

Y: One hot encoding of output (1/3)

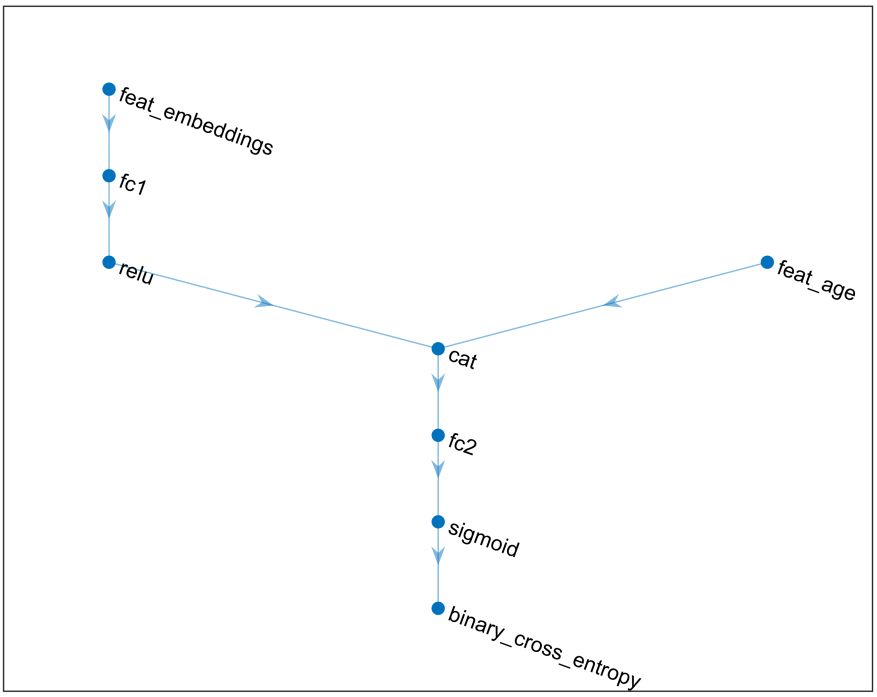
```
% One hot encoding of output
response = categorical(nndataY(:, 2));
response = onehotencode(response,2);
```

```
dsX2Train = arrayDatastore(nndataX(1:74, 2));
dsX1Train = arrayDatastore(nndataX(1:74, 3:end)', "IterationDimension",2);
dsTTrain = arrayDatastore(response(1:74,:));
dsTrain = combine(dsX1Train,dsX2Train,dsTTrain);

dsX2Val = arrayDatastore(nndataX(75:82, 2));
dsX1Val = arrayDatastore(nndataX(75:82, 3:end)', "IterationDimension",2);
dsTVal = arrayDatastore(response(75:82,:));
dsVal = combine(dsX1Val,dsX2Val,dsTVal);
```

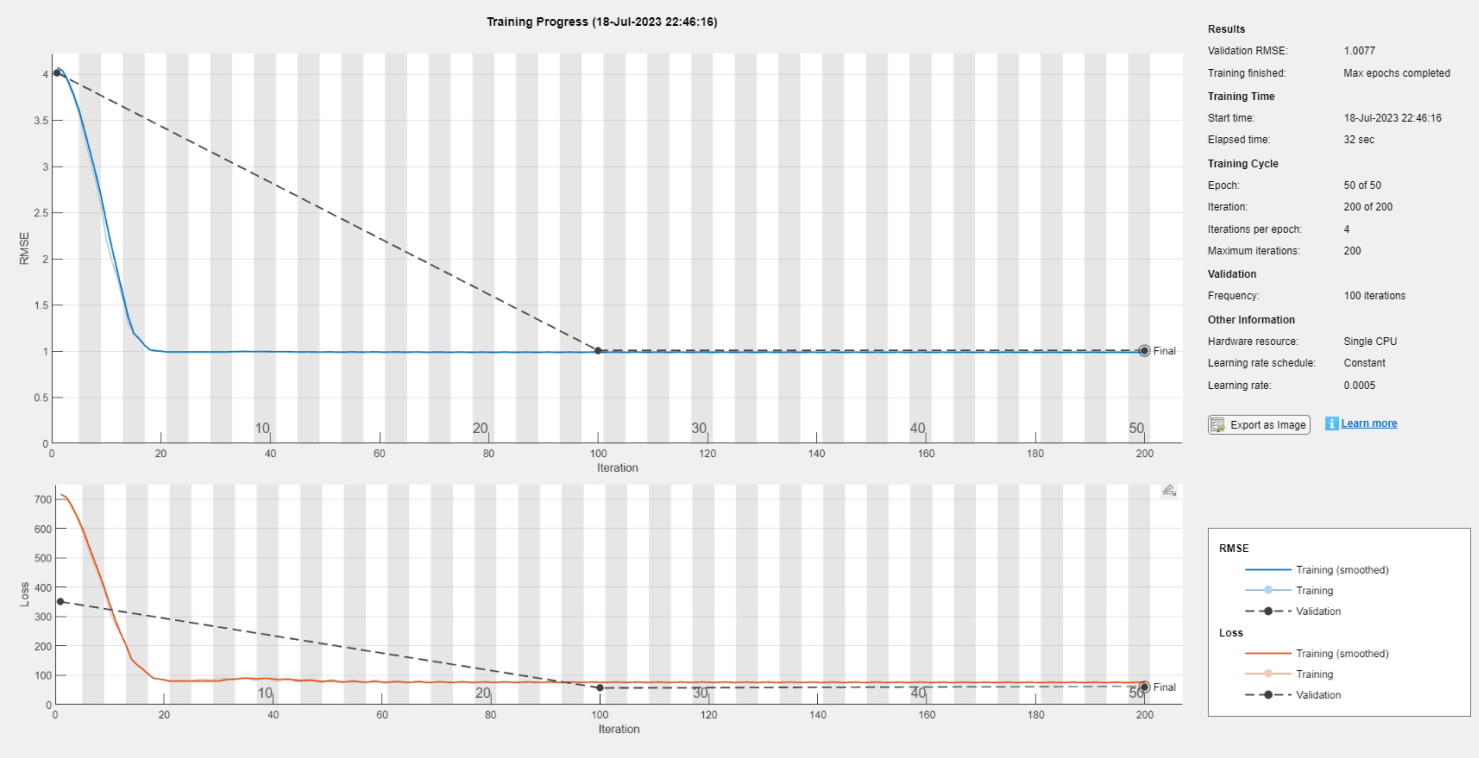
"End-to-End" Modelling

Note: Would have been a reasonable alternative, to not embed before-hand the input, but use traditional encoding. And leave the fc1 layer to act as a learnt embedding.

```
numFeatures = size(nndataX(:, 3:end),2);
% 60 different books are in the test set
numClasses = size(response, 2);

numHiddenUnits = 100;

layers = [
    featureInputLayer(numFeatures, Name='feat_embeddings')
    fullyConnectedLayer(100, Name = 'fc1')
    reluLayer
    concatenationLayer(1,2,Name="cat")
    fullyConnectedLayer(numClasses, Name = 'fc2')
    sigmoidLayer(Name = 'sigmoid')
    CustomBinaryCrossEntropyLossLayer("binary_cross_entropy")];

lgraph = layerGraph(layers);

featInput2 = featureInputLayer(1, Name = 'feat_age');
lgraph = addLayers(lgraph,featInput2);
lgraph = connectLayers(lgraph,"feat_age","cat/in2");
plot(lgraph)
```

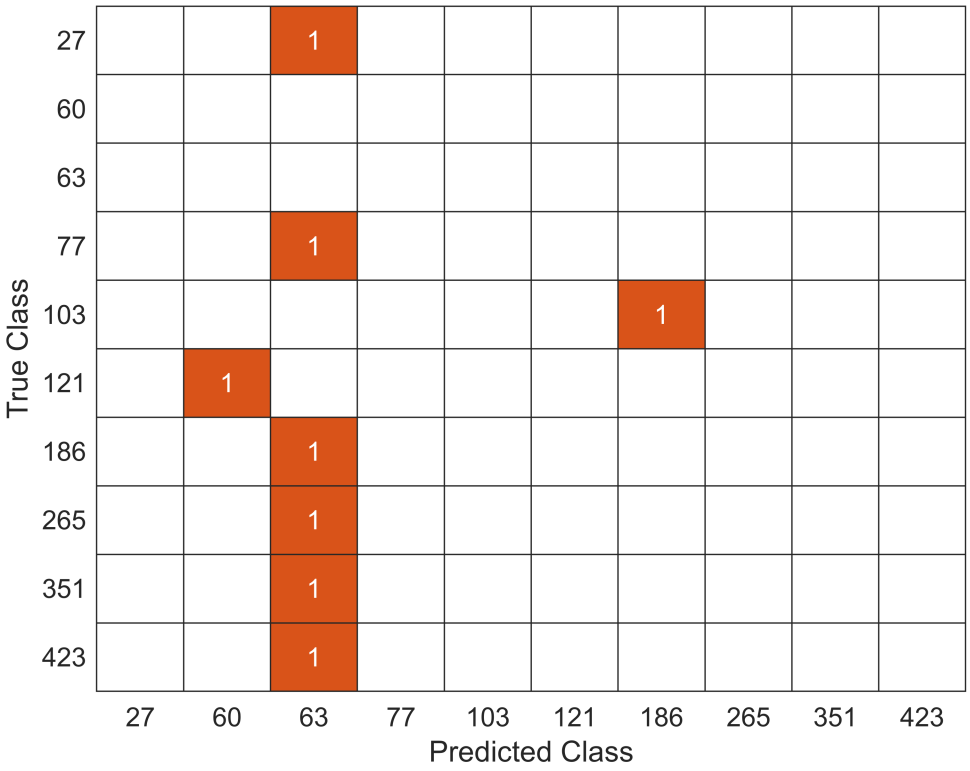



```
% No hyperparam tuning performed
options = trainingOptions("sgdm", ...
    InitialLearnRate=0.0005, ...
    MiniBatchSize=16, ...
    MaxEpochs=50, ...
    Verbose= false, ...
    ValidationData=dsVal, ...
    ValidationFrequency=100, ...
    ValidationPatience=5, ...
    Plots="training-progress");

trainedNet = trainNetwork(dsTrain,lgraph,options);
```



(Bad) Results

```
YPred = predict(trainedNet,dsVal);
bookTestID = unique(nndataY(:, 2));

[YPredTop, i] = maxk(YPred,3,2);

YPredresult = bookTestID(i);

figure
confusionchart(nndataY(75:82,2),YPredresult(:,1))
```




```
% Quite a Fictional book
title = nndata(find(nndata.BookID==255),6); title(1,1)
```

```
ans = 1×1 table
```

	title
1	'the pilot's wife'

Factors for improvement

1. Due to lack of time, only the title and age are considered for this modelling, even though a rich set (rankings, even images) are provided in the dataset.
2. Only one out of three books that were set aside per user are used in training. Much more of them can be used, creating encodings of output with multiple "1s".

Annex Data for Python nn-recommendation-system-word2vec.ipynb on github

```
nndata_train = [nndataX(1:75, 2:end) nndataY(1:75, 2:end)];
nndata_test = [nndataX(76:end, 2:end) nndataY(76:end, 2:end)];

writematrix(nndataX(1:75, 2:end), 'nndataTrainXCustomEmbTitle.csv')
writematrix(nndataY(1:75, 2:end), 'nndataTrainYCustomEmbTitle.csv')
writematrix(nndataX(76:end, 2:end), 'nndataTestXCustomEmbTitle.csv')
```

```
writematrix(nndataY(76:end, 2:end), 'nndataTestYCustomEmbTitle.csv')
```

Helper Functions

As a separate file, provided by Mathworks: CustomBinaryCrossEntropyLossLayer.m

```
function [avg_title_emb, age_bin] = avg_embedding(emb, x, optimValues)

title_embs = [];

for i=1:size(x,1)

    title = string(x(i,1));

    title = strrep(title,',','');
    title = strrep(title,'.','');

    splits = split(title, ' ');
    title_emb = word2vec(emb, splits);
    title_emb = rmmissing(title_emb);
    title_embs = [title_embs;title_emb];

end

avg_title_emb = mean(title_embs);
age_bin = cell2mat(x(1,2));

end

function [id1, id2, id3] = getBooks(x)

id1 = x(1);
id2 = x(2);
id3 = x(3);
end
```

Inspired by https://ceur-ws.org/Vol-2871/paper2.pdf Embedding-based Neural Network Models for Book Recommendation in University Libraries, Choi et al. 2021