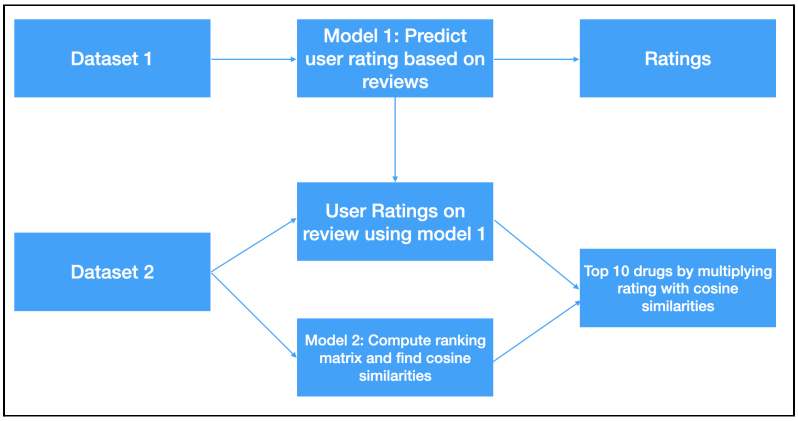
W4995-AML Project Deliverable #3 - Project Report

Team 25: Huaizhi Ge, Shashwat Singh, Yosha Singh Tomar, Alex Kita

Drug Recommendation based on LDA and Sentiment Analysis

Introduction

With the advent of new drugs everyday, it is important to take into account the level of user satisfaction and the effectiveness of the drug. There are also popular websites such as WebMD where users provide feedback regarding the drug usage and effectiveness. We conducted two tasks on this final project. The first task is to build a CNN model predicting user rating based on drug reviews on UCI drug dataset (Gräßer et al., 2019) and predict ratings from reviews in the WebMD dataset (Harode, 2020). On the second task, we implemented a recommendation system that computes ranking matrix and cosine similarities based on their age, sex, conditions by using LDA recommendation system. We also take into account the rating scores we predicted on the first task to recommend top 10 drugs. The following image illustrates the whole procedure:



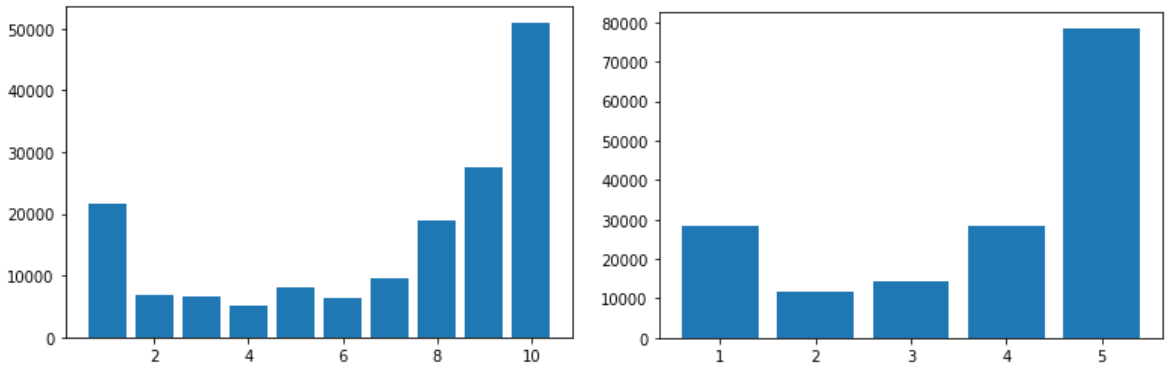
Exploratory Data Analysis and Visualization

https://drive.google.com/file/d/1FA_un2e2NI-ECfKgBIcsinT11zlQAk5Y/view?usp=sharing

Task I: Build a model to predict user ratings from reviews

Data Preprocessing

- Data Cleaning - Converted all the words into lowercase. Then removed hyperlinks using the ‘re’ python package. Removed punctuations, numbers. These steps were followed by tokenization using the TreebankwordTokenizer which primarily uses regular expressions to tokenize text and has been practically determined more feasible in case of sentiment prediction tasks. Further, stopwords are removed for the tokens generated and these tokens are further stemmed to using PorterStemmer.
- Reducing the number of classes - Since, the second dataset (i.e. the WebMD dataset) consists of certain columns which depict terms similar to rating in the scale of 1 to 5, we had to convert the ratings in our first dataset (i.e. the UCI dataset) that was originally on a 1 to 10 scale to a 1 to 5 scale. We just assigned the first two ratings as 1, second two as 2 and so on. In essence original ratings of 1 and 2 were assigned a new rating of 1, 3 and 4 were assigned 2 and so on.



- Class Imbalance - We had some class imbalance in the data. We can observe that most people have rated the drugs on a higher range.
- Word2Vec - Prior to applying Word2Vec, we used the 'text_to_sequences' function from keras text preprocessing module to convert the preprocessed reviews into sequences determined by their corresponding word index. We also padded each of these sequences to a maximum length of 25 to be consistent with the input shape.
  We chose the Google News Word2Vec embedding with embedding size set to 300 and unique vocabulary length set to 150000. We had to make sure that this model also gives predictions on the second WebMD dataset which contained 149k unique vocabulary words after all the preprocessing. Therefore, we chose 150k as the unique vocabulary length. We used the gensim package to load the trained Word2Vec model as an embedding matrix.

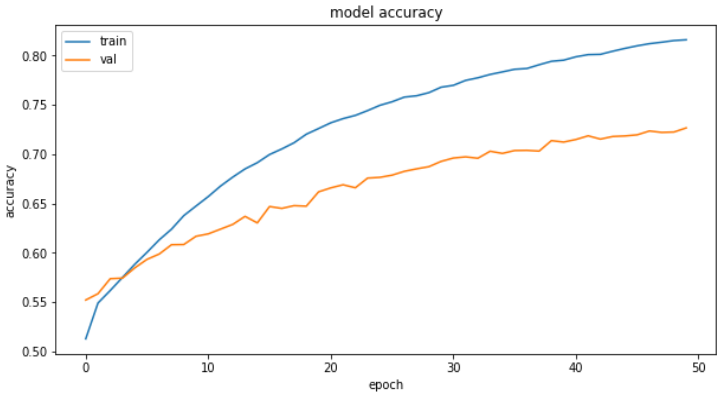
Modeling

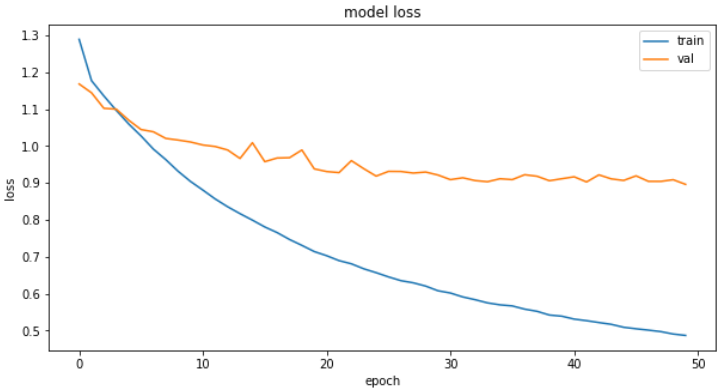
- Different techniques used - We have primarily used CNN to predict the ratings from the reviews. In CNN itself, we tried multiple models from single layered Convolution layers to multiple layer Convolution layers and to using LSTM. But we achieved the best results for two Convolution layers followed by two dense layers.
- Model Architecture

```
model.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 25, 300)           45000300

conv1d (Conv1D)              (None, 25, 300)           450300

max_pooling1d (MaxPooling1D  (None, 12, 300)           0
)

batch_normalization (BatchN  (None, 12, 300)           1200
ormalization)

spatial_dropout1d (SpatialD  (None, 12, 300)           0
ropout1D)

conv1d_1 (Conv1D)            (None, 12, 300)           450300

max_pooling1d_1 (MaxPooling  (None, 6, 300)            0
1D)

batch_normalization_1 (Batc  (None, 6, 300)            1200
hNormalization)

spatial_dropout1d_1 (Spatia  (None, 6, 300)            0
lDropout1D)

flatten (Flatten)            (None, 1800)              0

dense (Dense)                (None, 128)               230528

dense_1 (Dense)              (None, 5)                 645

=================================================================
Total params: 46,134,473
Trainable params: 1,132,973
Non-trainable params: 45,001,500
_________________________________________________________________
```

Model Performance

- Training Results



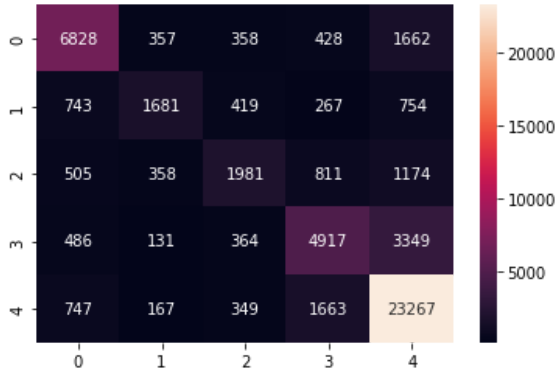


- Validation Results
  - Classification Report

	precision	recall	f1-score	support
0	0.74	0.72	0.73	5710
1	0.64	0.43	0.52	2305
2	0.58	0.43	0.50	2871
3	0.62	0.53	0.58	5670
4	0.77	0.90	0.83	15704
accuracy			0.73	32260
macro avg	0.67	0.60	0.63	32260
weighted avg	0.72	0.73	0.72	32260

- Testing Results
  - The model is tested on the data from the test file “drugComTest_raw.tsv”
  - Classification Report and Confusion Matrix

	precision	recall	f1-score	support
0.0	0.20	0.13	0.15	9633
1.0	0.08	0.03	0.04	3864
2.0	0.10	0.06	0.07	4829
3.0	0.17	0.15	0.16	9247
4.0	0.49	0.66	0.56	26193
accuracy			0.38	53766
macro avg	0.21	0.20	0.20	53766
weighted avg	0.32	0.38	0.34	53766



The intermediate classes (1, 2 and 3) have low recall scores as compared to the extreme classes (0 and 4). From the confusion matrix, we can see that across all classes, the majority of misclassified predictions are either in class 0 or class 4. This skewed distribution is probably due to the imbalance in the training data. However, precision, recall and accuracy of the extreme classes 0 and 4 are considerably much better.

Task II: Drug Recommendation

Preprocessing

- Followed the same preprocessing steps as in the first dataset on the ‘reviews’ column of the WebMD dataset.
- Filtered only those rows for which their corresponding drug count is greater than 50.

Prediction Using First Dataset

- Predicted equivalent ratings using the model generated from the first dataset (UCI Drug Review) on the preprocessed reviews columns of the WebMD dataset.

Creating User Embedding

- Selected input features as ‘Age’, ‘Condition’ and ‘Sex’ from the WebMD dataset and applied one hot encoding to get a feature matrix of shape 316629 x 1169. 316629 corresponds to the number of users and 1169 corresponds to one hot encoded feature.
- Applied PCA to reduce the dimension of the user matrix to 100 dimensions.
- Applied LDA over the PCA to regularize the parameters and further reduce the dimension to just 10 features. This is our final user embedding.

Creating Item (Drug) Embedding

- One hot encoded the ‘Drug’ column from the WebMD dataset to get a feature matrix of shape 316629 x 1081. 1081 indicates the number of uniquely one-hot encoded ‘Drug’ columns.
- Applied PCA for dimensionality reduction upto 100 features which is followed by LDA to further reduce dimensionality to 10. This is our final item (drug) embedding.

Drug Recommendation

- Generated the Ranking Matrix by multiplying User Embedding with the transpose of the Item (Drug) Embedding. This ranking matrix has a shape of 316629 (i.e. the number of users) x 1081 (i.e. number of different Drugs).
- Ex: As an example we have recommended drugs to User 0.
    1. We found the cosine similarity for the 0th index vector (Representing User 0) of the ranking matrix with every other vector (all other Users) in the ranking matrix.
    2. Multiply the predicted ratings with the cosine similarity to change the ordering based on other Users reviews of each drug.
    3. ArgSort the ranking by the net score in descending order to get the index of the drugs.
    4. Select top 10 Drugs based on the index retrieved from the previous step.
    5. Final recommendations obtained for User 0.

```
{'capsaicin cream',
 'capzasin-hp cream',
 'claritin tablet',
 'diclofenac sodium',
 'diclofenac sodium er',
 'ferrous sulfate tablet, delayed release (enteric coated)',
 'medrol',
 'phenylephrine hcl tablet, chewable',
 'venofer vial'}
```

The original drug used by the person was ‘25dph-7.5peh’ which has a ‘DrugID’ of 146724. If we see the conditions which are cured by this drug, we can see that they are used to cure allergies, coughs and itchiness. Now the recommended drugs mostly contain the drugs that are used to cure itchiness such as ‘capsaicin cream’ or used to cure allergies such as ‘claritin tablet’ which is highly recommended by the recommender engine.

Limitations/Next Steps/Conclusion

We successfully predicted rating score with 72% test accuracy and generated drug recommendation based on rating, age, sex, and conditions in WebMD dataset. For the first dataset, we observed imbalance data for rating in which users tend to put a polarized score such as 0 and 10 for rating. Though we dealt with the data imbalance by reducing the number of categories from 10 to 5, our model predicts rating 0 or 4 for the significant amount of data. For the second task, as a future work, we could improve our recommendation model by taking into account other user scores in the dataset such as satisfaction, ease of use and effectiveness. Furthermore, we could try neural networks instead of PCA and LDA to find embeddings.

References

1. Gräßer, F., Kallumadi, S., Malberg, H., Zaunseder, S.: UCI Machine Learning Repository (2019). https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29#Irvine. Accessed 1 Nov 2021.

2. Harode, R.. (2020). WebMD Drug Reviews Dataset, Version 1. Retrieved 1 Nov 2021 from https://www.kaggle.com/rohanharode07/webmd-drug-reviews-dataset/.

3. Kung-Hsiang, Huang (Steeve) (2020), A Deep Dive into Latent Dirichlet Allocation (LDA) and Its Applications on Recommender System, https://blog.rosetta.ai/a-deep-dive-into-latent-dirichlet-allocation-lda-and-its-applications-on-recommender-system-e2e8ea5e661c.