CSC 180-01 Intelligent Systems (Fall 2019)

Mini-Project 1: Yelp Business Rating Prediction using Tensorflow

Due at 11:00 am, Friday, September 27, 2019

Demo Session: class time, Friday, September 27, 2019

1. Problem Formulation

In this project, we aim to predict a business's stars rating using ALL the reviews of that business based on neural network implementation in Tensorflow. Consider this problem as a regression problem.

(1) Report the RMSE and plot the lift chart of the BEST neural network model you obtained.

(2) Choose 3-5 businesses from your test dataset (preferably from different categories).   Show the true star ratings of those businesses and the predicted ratings output by the best model.

2. Dataset

https://www.yelp.com/dataset/download

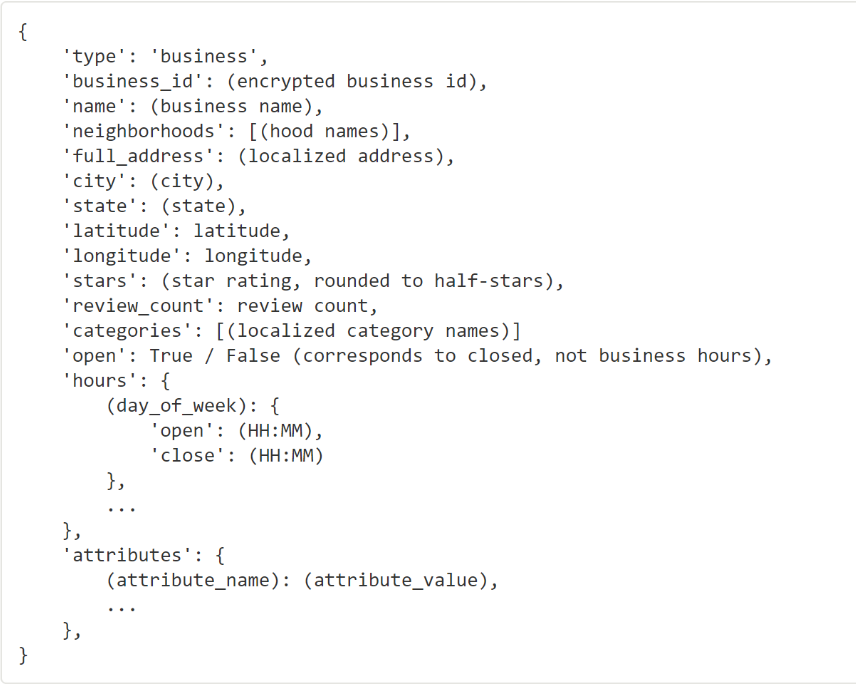
This set includes information about local businesses in 10 metropolitan areas across 2 countries.

- 1,223,094 tips by 1,637,138 users
- Over 1.2 million business attributes like hours, parking, availability, and ambience
- Aggregated check-ins over time for each of the 192,609 businesses

The dataset contain several JSON files. You can find the format of the data here:
https://www.yelp.com/dataset/documentation/main

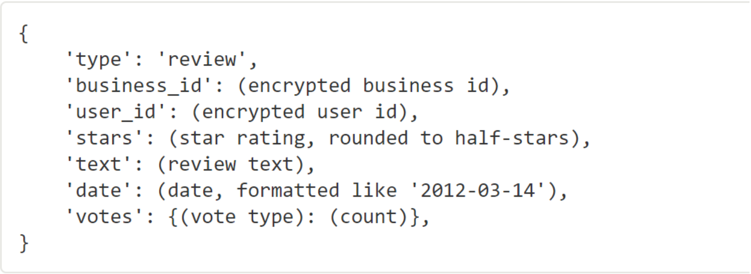
Example file formats are as follows.

business



```
{
    'type': 'business',
    'business_id': (encrypted business id),
    'name': (business name),
    'neighborhoods': [(hood names)],
    'full_address': (localized address),
    'city': (city),
    'state': (state),
    'latitude': latitude,
    'longitude': longitude,
    'stars': (star rating, rounded to half-stars),
    'review_count': review count,
    'categories': [(localized category names)]
    'open': True / False (corresponds to closed, not business hours),
    'hours': {
        (day_of_week): {
            'open': (HH:MM),
            'close': (HH:MM)
        },
        ...
    },
    'attributes': {
        (attribute_name): (attribute_value),
        ...
    },
}
```

review



```
{
    'type': 'review',
    'business_id': (encrypted business id),
    'user_id': (encrypted user id),
    'stars': (star rating, rounded to half-stars),
    'text': (review text),
    'date': (date, formatted like '2012-03-14'),
    'votes': {(vote type): (count)},
}
```

3. Requirements

- You are required to split data to training and test. Use training data to train your models and evaluate the model quality using test data.
- Use TF-IDF to extract features from review contents for your model. If you experience low memory error when using ***tfidfVectorzier***, set parameters ***max_df, min_df,*** and ***max_features*** appropriately.
- You must use EarlyStopping when training neural networks using Tensorflow.
- Tuning the following hyperparameters when training neural networks using Tensorflow and record how they affect performance
  - **Activation:** relu, sigmoid, tanh
  - **Number of layers and neuron count for each layer**
  - **Optimizer:** adam and sgd.

4. Grading breakdown

You may feel this project is described with some certain degree of vagueness, which is left on purpose. In other words, **creativity is strongly encouraged**. Your grade for this project will be based on the soundness of your design, the novelty of your work, and the effort you put into the project.

Use the evaluation form on Canvas as a checklist to make sure your work meet all the requirements.

Implementation	**70 pts**
Your report	**20 pts**
In-class defense	**10 pts**

5. Teaming:

Students must work in teams with no more than 3 people. Think clearly about who will do what on the project. Normally people in the same group will receive the same grade. However, the instructor reserve the right to assign different grades to team members depending on their contributions. So you should choose partner carefully!

6. Deliverables:

(1) All your source code in Python Jupyter notebook.

(2) Your report in PDF format, with your name, your id, course title, assignment id, and due date on the first page. As for length, I would expect a report with more than one page. Your report should include the following sections (but not limited to):

(1) **Problem Statement**
(2) **Methodology**
(3) **Experimental Results and Analysis**
(4) **Task Division and Project Reflection**

In the section "**Task Division and Project Reflection**", describe the following:

- who is responsible for which part,
- challenges your group encountered and how you solved them
- and what you have learned from the project as a team.

10 pts will be deducted for missing the section of task division and project reflection.

All the files must be submitted **by team leader** on Canvas before

11:00 am, Friday, September 27, 2019

NO late submissions will be accepted.

7. In-class Demo:

Each team member must demo your work during the scheduled demo session. Each team have **three minutes** to demo your work in class. Failure to show up in defense session will result in **zero** point for the project. The following is how you should allocate your time:

- Model/code design (1 minute)
- Findings/results (1 minute)
- Task division, challenges encountered, and what you learned from the project (1 minutes)

8. Hints

- You may use the following code to convert JSON data into a tabular format Pandas can read.

```
import json
import csv
import pandas as pd

outfile = open("review_stars.tsv", 'w')
sfile = csv.writer(outfile, delimiter ="\t", quoting=csv.QUOTE_MINIMAL)
sfile.writerow(['business_id','stars', 'text'])

with open('yelp_academic_dataset_review.json') as f:
  for line in f:
    row = json.loads(line)
    # some special char must be encoded in 'utf-8'
    sfile.writerow([row['business_id'], row['stars'], (row['text']).encode('utf-8')])

outfile.close()

df= pd.read_csv('review_stars.tsv', delimiter ="\t", encoding="utf-8")
```

- You may use the following sample code to group ALL the reviews by each business and create a new dataframe, where each line is a business with all its reviews. Write your code to use ***tfidfVectorzier*** to obtain TFIDF representation for each business.

```
df_review_agg = df.groupby('business_id')['text'].sum()

df_ready_for_sklearn = pd.DataFrame({'business_id': df_review_agg.index, 'all_reviews':
df_review_agg.values})
```

- To align all the reviews of a business with its business star rating, you may want to join the review table with the business table on the business_id column. Pandas supports high performance SQL join operations. Use Pandas function ***pd.merge()*** to **merge (or to say, join) two dataframes** based on values in one particular column. See an example here:

https://chrisalbon.com/python/data_wrangling/pandas_join_merge_dataframe/

- If you want to **merge two numpy arrays**, check out Numpy function ***np.concatenate()***

  https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.concatenate.html

- Convert a Pandas Dataframe to its corresponding Numpy array representation, use the *DataFrame.values* attribute

  http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.values.html#pandas.DataFrame.values

- For one-hot coding, you may use Pandas ***pd.get_dummies().***

  https://pandas.pydata.org/pandas-docs/stable/generated/pandas.get_dummies.html

9. Think beyond the Project

- If a business have too few reviews, predicting star rating from the review contents only may not be a good idea.
  - Can you filter out the businesses with too few reviews (e.g., less than 10) to build a more accurate model?
  - Can you build a more accurate model by also taking the number of reviews (review count) into account?
  - What other information can be used to train a more accurate model? Business categories? Check-in count?