CSC 215-01 Artificial Intelligence (Spring 2019)

Mini-Project 1: Yelp Business Rating Prediction

Due at 4 pm, Wednesday, February 13, 2019

Demo Session: class time, Wednesday, February 13, 2019

In this project you will practice supervised learning algorithms with Sklearn.

1. Problem Formulation

In this project, you predict a business's stars rating using all the reviews of that business and review count. Use the following models implemented in Sklearn:

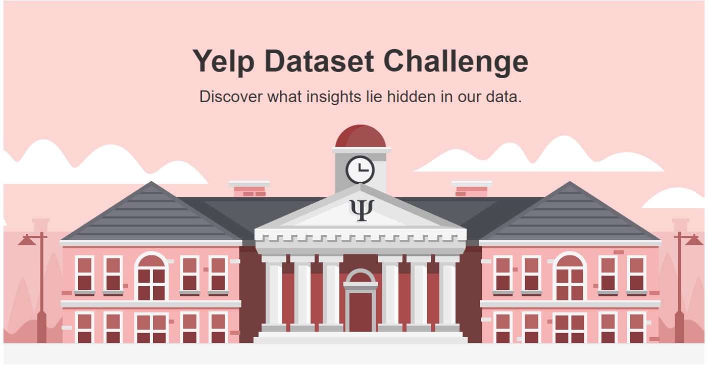
- Linear Regression (Consider the problem as a regression problem)
- Logistic Regression
- Nearest Neighbor
- Support Vector Machine
- Multinomial Naive Bayes

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

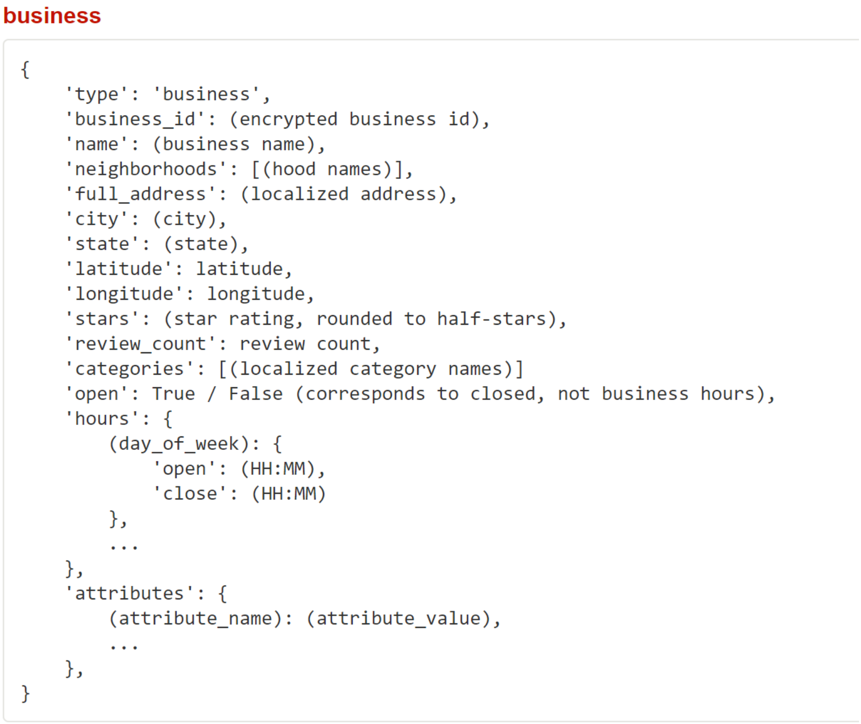
2. Dataset (30 pts)

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

This set includes information about local businesses in 10 metropolitan areas across 2 countries. The dataset contain several json files.

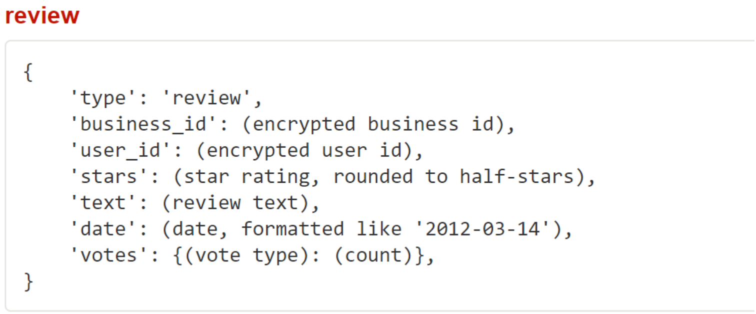


Example file formats are as follows.



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)},
}
```

You can find the meaning of each field here: https://www.yelp.com/dataset/documentation/main

3. Requirements

- You are required to split data for training and testing. Use training data to train your models but you **do not need to** evaluate the model quality using test data. You will evaluate the quality of your models in the next project.

- Use Python/Numpy/Pandas and Sklearn to finish this project. Any other Python libraries are also welcome to use.

- Use TF-IDF to do feature extraction from review contents for your models.

- If you experience low memory error on your machine when you use ***tfidfVectorzier***, set parameters ***max_df, min_df,*** and ***max_features*** appropriately.

- Do feature normalization.

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	**15 pts**
In-class defense	**10 pts**
Additional features (novelty)	**5 pts**

5. Teaming:

Students must work in teams of 2 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):

- **Problem Statement**
- **Methodology**
- **Experimental Results and Analysis**
- **Task Division and Project Reflection**
- **Additional Features.**

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

4 pm, Wednesday, February 13, 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 **five minutes** to demo your work in class. The following is how you should allocate your time:

- Model design (1 minute)
- Findings/results (1 minute)
- Additional features (1 minute)
- Task division (1 minute)
- Challenges encountered and what you have learned from the project (1 minutes)

If you implement **additional features (novelty)**, please do mention them to receive credit for novelty.

Failure to show up in defense session will result in **zero** point for the project.

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})
```

- 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