

NewYorker Data Science Exercise

Kai Chen

Github repo: https://github.com/ck-unifr/yelp_dataset_challenge

Case 1: predict star ratings with checkin data

- The goal is to predict the star ratings based on the business and checkin data.
- The problem is formulated as a multi-label classification problem.
- Steps [*]
 - Load data into pandas dataframe
 - *business.json* contains the business data
 - *checkin.json* contains the check in data
 - Data preprocessing
 - Merge the business data and checkin data
 - Fill missing value
 - Transfer string features into integer
 - Normalize numeric features
 - Extract text feature with tf-idf
 - Data is splitted into train set and test set
 - Model training
 - Two ensemble learning models, i.e., random forest, gradient boost machine, are used
 - Random search is used for hyperparameter tuning

[*] The details can be found in the jupyter notebook

https://github.com/ck-unifr/yelp_dataset_challenge/blob/master/notebooks/predict_star_ratings_with_checkin.ipynb

Case 1: predict star ratings with checkin data

- Conclusion [*]
 - Gradient boost machine (xgboost) with hyperparameter tuning achieves slightly better performance than random forest.
 - tf-idf 'categories' does not improve the performance.
 - The difficulty is the imbalanced data, i.e., we have only one thousand samples with one star in our train set and we have much more samples with 4 stars.
- Future work
 - Error analysis.
 - One-hot encoding 'categories'.
 - Word Embedding 'categories'
 - Using sampling algorithms to overcome the imbalanced data problem.
 - For example, python imbalanced-learn (<https://imbalanced-learn.org/en/stable/index.html>)

[*] The details can be found in the jupyter notebook

https://github.com/ck-unifr/yelp_dataset_challenge/blob/master/notebooks/predict_star_ratings_with_checkin.ipynb

Case 2: predict star ratings with photo

- The goal is to predict the star ratings based on the photos
- The problem is formulated as a multi-label classification problem
- Steps [*]
 - Load data into pandas dataframe [**]
 - *business.json* contains the business data
 - *photo.json* contains the photo data
 - Data preprocessing
 - Merge the business and photo data
 - Prepare the image data, i.e., transfer image into array in python
 - Model training
 - Convolutional neural network is used for the classification task

[*] The details can be found in the jupyter notebook

https://github.com/ck-unifr/yelp_dataset_challenge/blob/master/notebooks/predict_star_ratings_with_photos.ipynb

[**] Due the the limited computational resource I have, only a part of the photos is used.

Case 2: predict star ratings with photo

- Conclusion [*]
 - The performance is better than using checkin data, however only a small part of data is used in this case. So the comparison is not fair.
 - The difficulty is the imbalanced data as mentioned in case 1.
 - Error analysis shows that even for human it is hard to predict stars by images
- Future work
 - Combine text data, 'caption'
 - Transfer learning with VGG neural network.
 - Try other CNNs, e.g., GoogleNet, ResNet, OctConv.
 - Using sampling algorithms to overcome the imbalanced data problem.
 - GANs
 - In order to predict the stars of a hotel more accurate, we can take several images as the input.

[*] The details can be found in the jupyter notebook

https://github.com/ck-unifr/yelp_dataset_challenge/blob/master/notebooks/predict_star_ratings_with_photos.ipynb

Docker

- Type the command in terminal
 - `docker run -it --rm --name ds-jupyter -p 8888:8888 -v [work directory]:/home/jovyan/work jupyter/datascience-notebook`
- Open a web browser and enter the address which is indicated in the terminal
 - For example
 - `http://127.0.0.1:8888/?token=5b02107f343bc4bab3d87e81a9821c70e20be879a9620983`