

# NewYorker Data Science Exercise

Kai Chen

# 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
    - Text feature extraction 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](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

- One-hot encoding '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](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 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
  - 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](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 difficulty is the imbalanced data as I mentioned in case 1
- Future work
  - Transfer learning with VGG neural network
  - Try other CNNs, e.g., GoogleNet, ResNet, OctConv

[\*] 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](https://github.com/ck-unifr/yelp_dataset_challenge/blob/master/notebooks/predict_star_ratings_with_photos.ipynb)