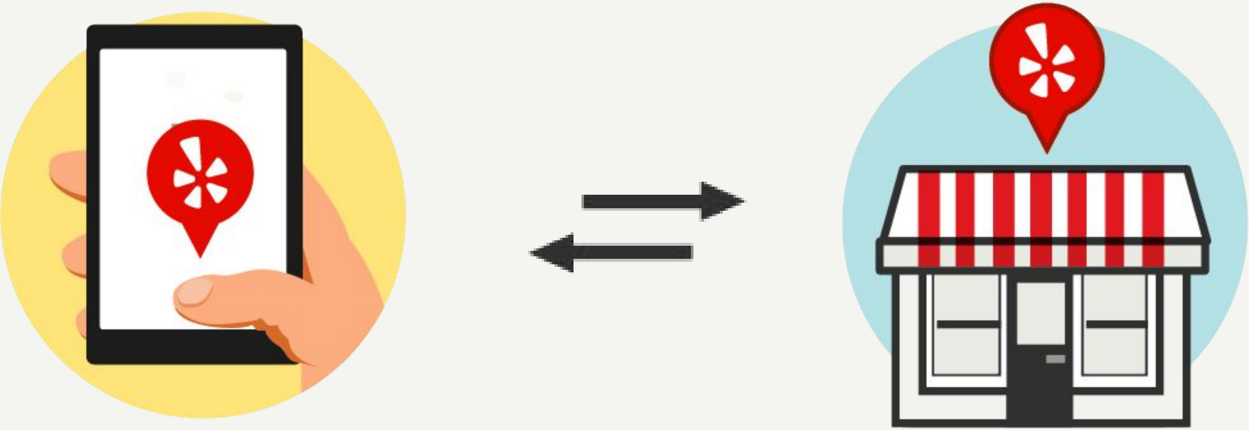
How Likely Would You Give A Five-Star Review on yelp?

Getting Your Hands Dirty with *scikit-learn*

Xun Tang
xun@yelp.com

Yelp’s Mission

Connecting people with great local businesses.



Yelp Stats

As of Q1 2016

90M

102M

70%

32

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

seaborn

scikit learn

matplotlib

jupyter

yelp®

yelp.com/dataset_challenge/

Given user's past reviews on Yelp

When the user writes a review for a business she hasn't reviewed before

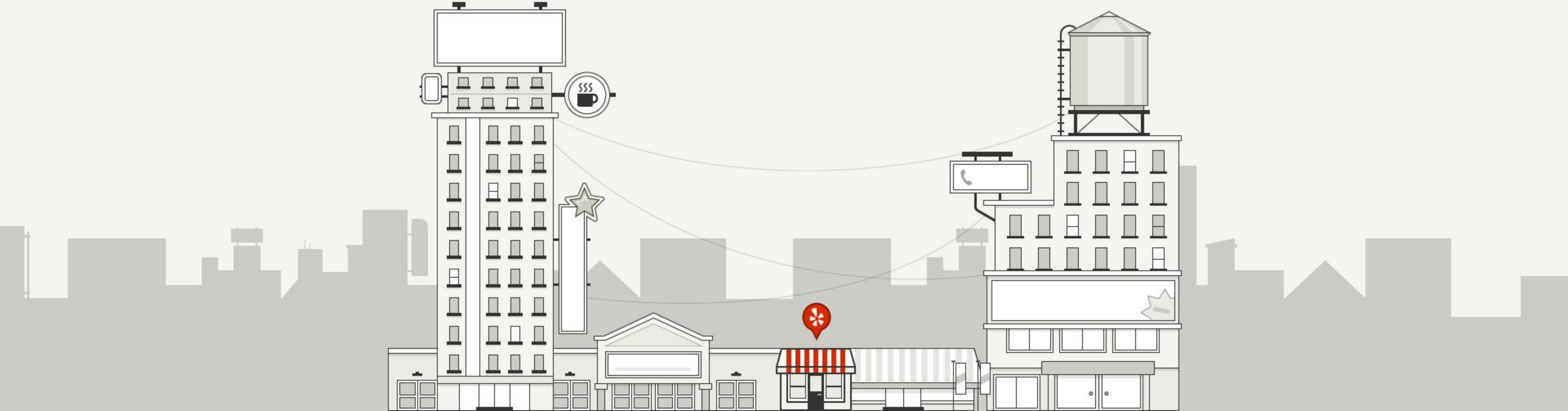
How likely will it be a ★★★★★ review?

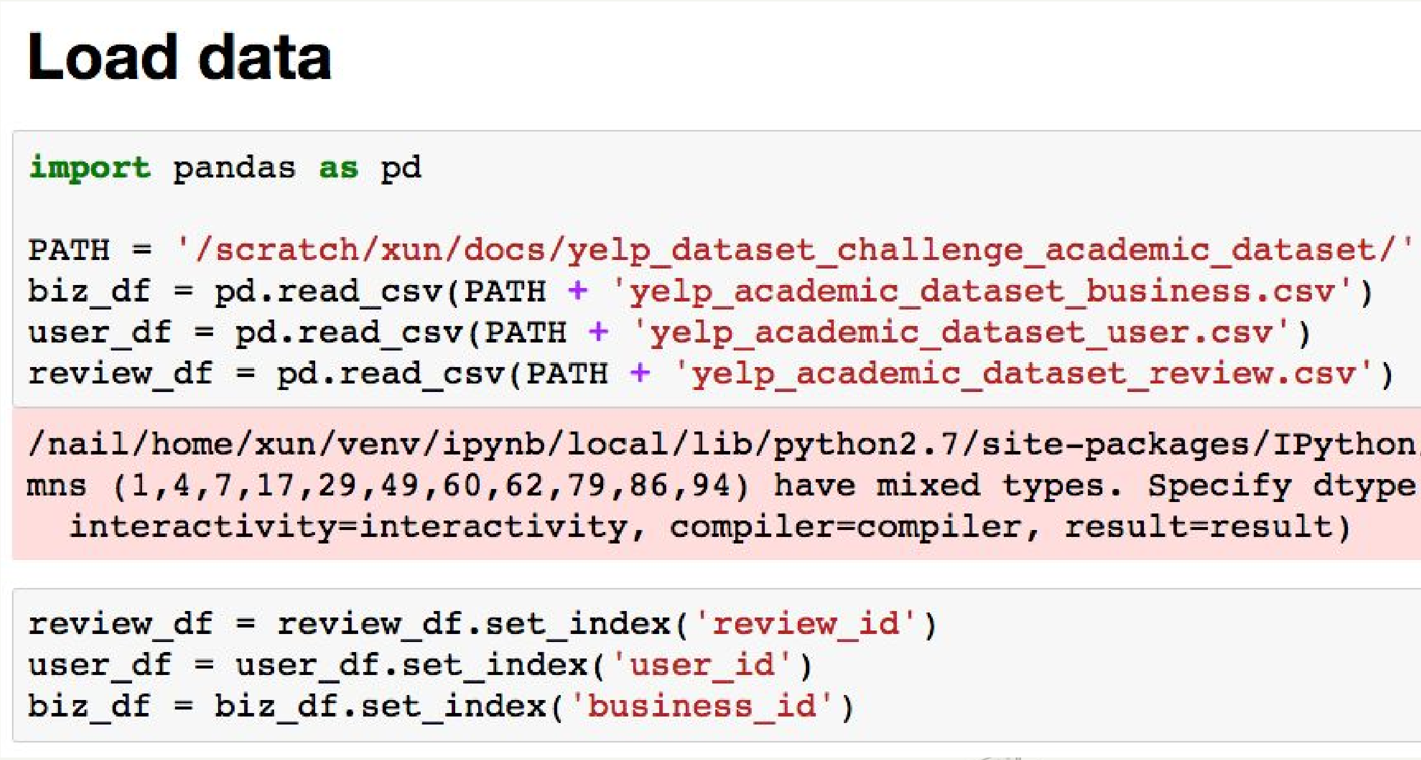
yelp Public Dataset

yelp.com/dataset_challenge/

Demo

github.com/xun-tang/pyladies_jupyter_demo





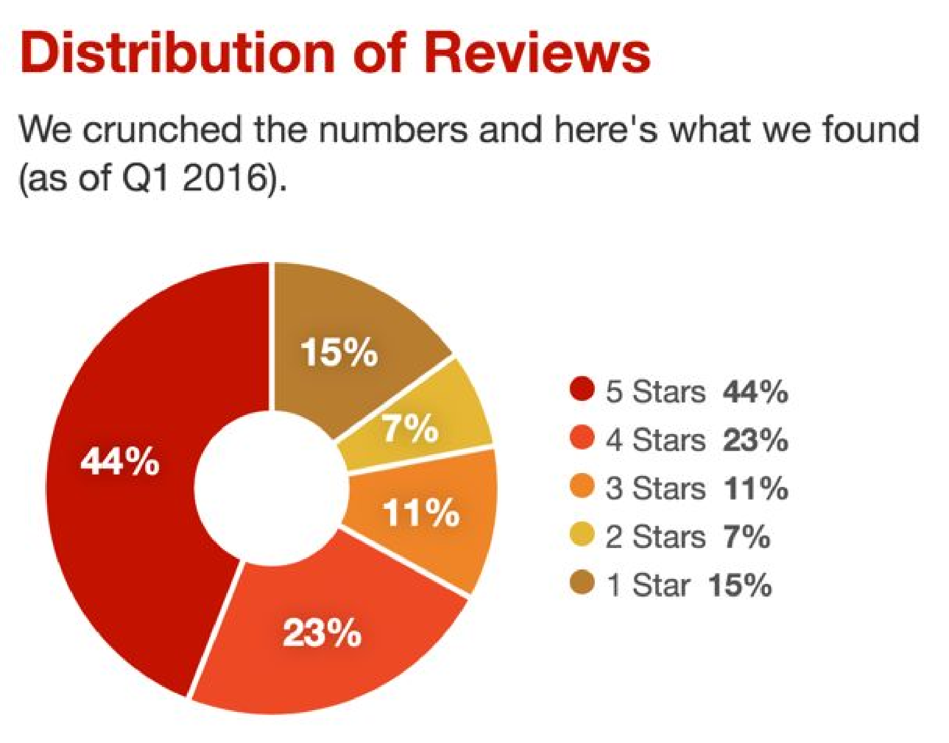
Load data

```
import pandas as pd

PATH = '/scratch/xun/docs/yelp_dataset_challenge_academic_dataset/'
biz_df = pd.read_csv(PATH + 'yelp_academic_dataset_business.csv')
user_df = pd.read_csv(PATH + 'yelp_academic_dataset_user.csv')
review_df = pd.read_csv(PATH + 'yelp_academic_dataset_review.csv')
```

```
/nail/home/xun/venv/ipynb/local/lib/python2.7/site-packages/IPython
mns (1,4,7,17,29,49,60,62,79,86,94) have mixed types. Specify dtype
  interactivity=interactivity, compiler=compiler, result=result)
```

```
review_df = review_df.set_index('review_id')
user_df = user_df.set_index('user_id')
biz_df = biz_df.set_index('business_id')
```

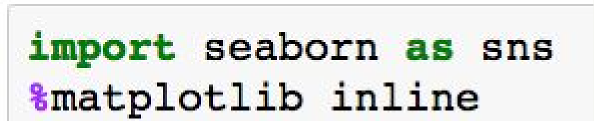
Visualize the data
Plot distribution of review star ratings



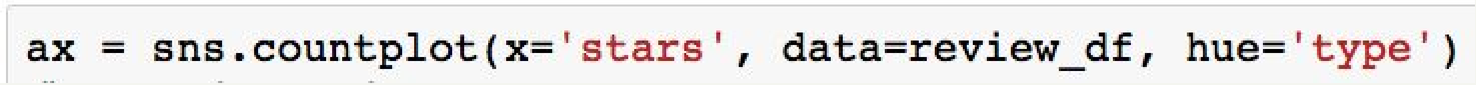
Distribution of Reviews

We crunched the numbers and here's what we found (as of Q1 2016).

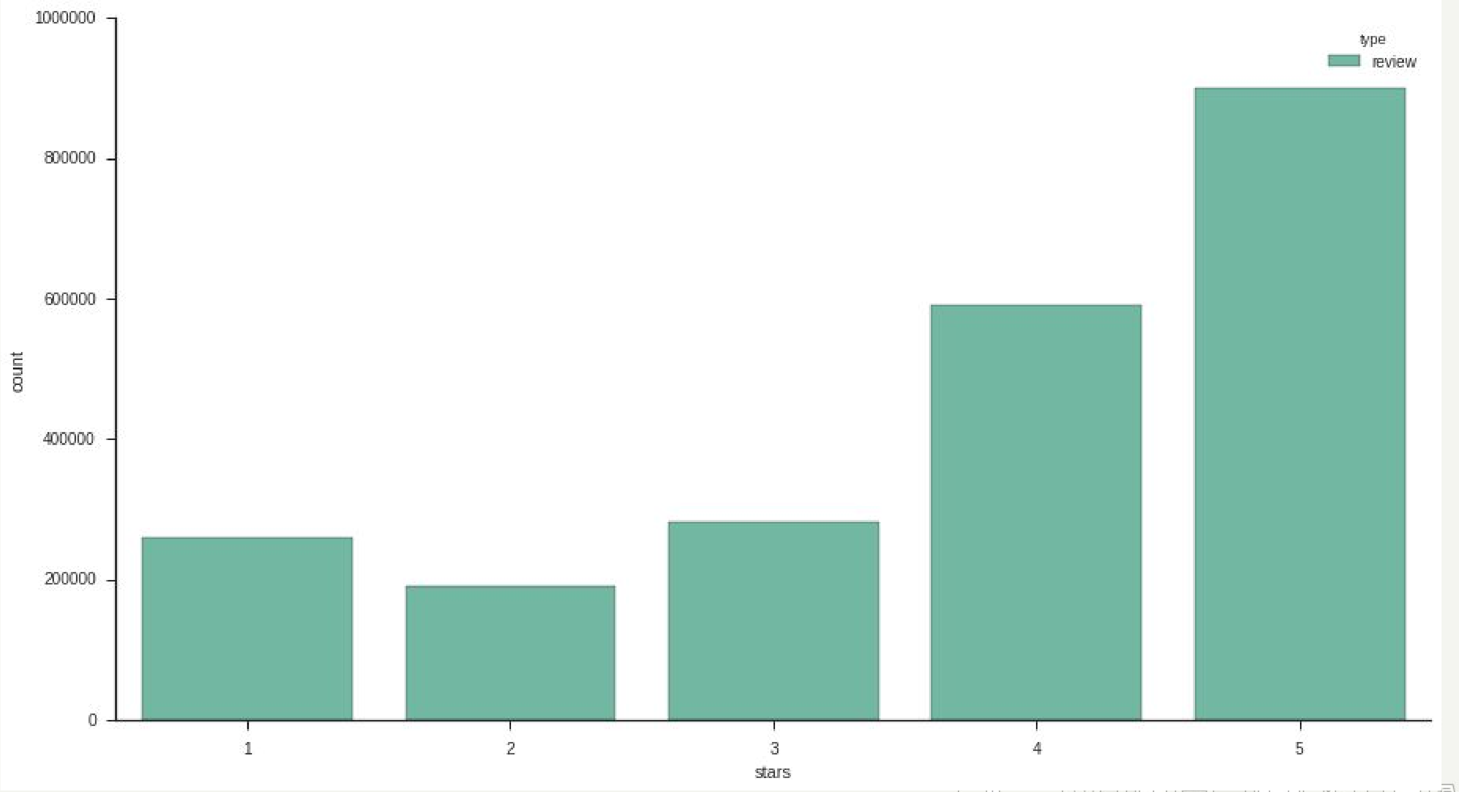
Source: https://www.yelp.com/press



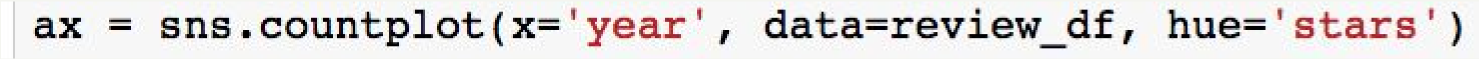
```python
import seaborn as sns
%matplotlib inline
```



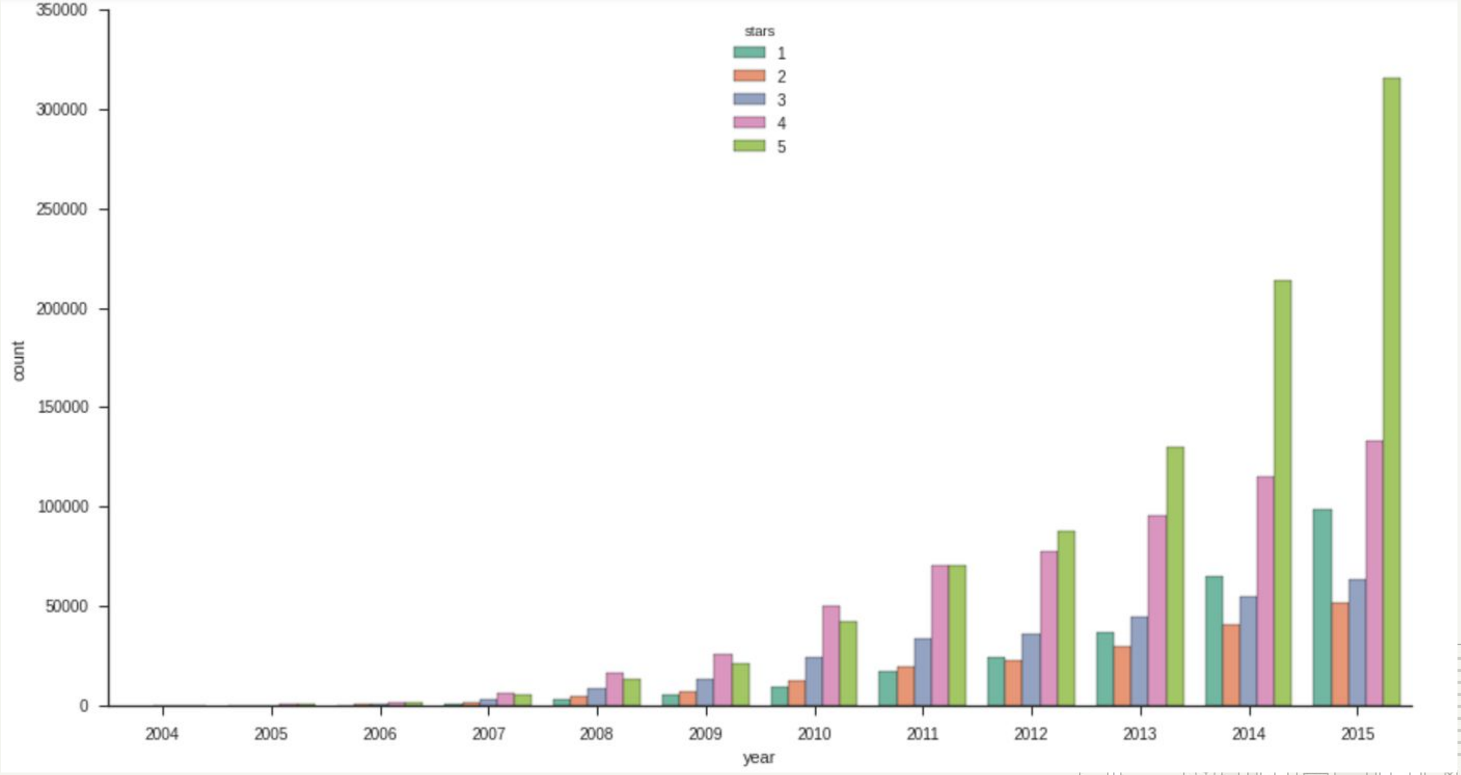
```python
ax = sns.countplot(x='stars', data=review_df, hue='type')
```



Plot review star ratings by year



```
ax = sns.countplot(x='year', data=review_df, hue='stars')
```



Featurize the data

Convert date string to date delta

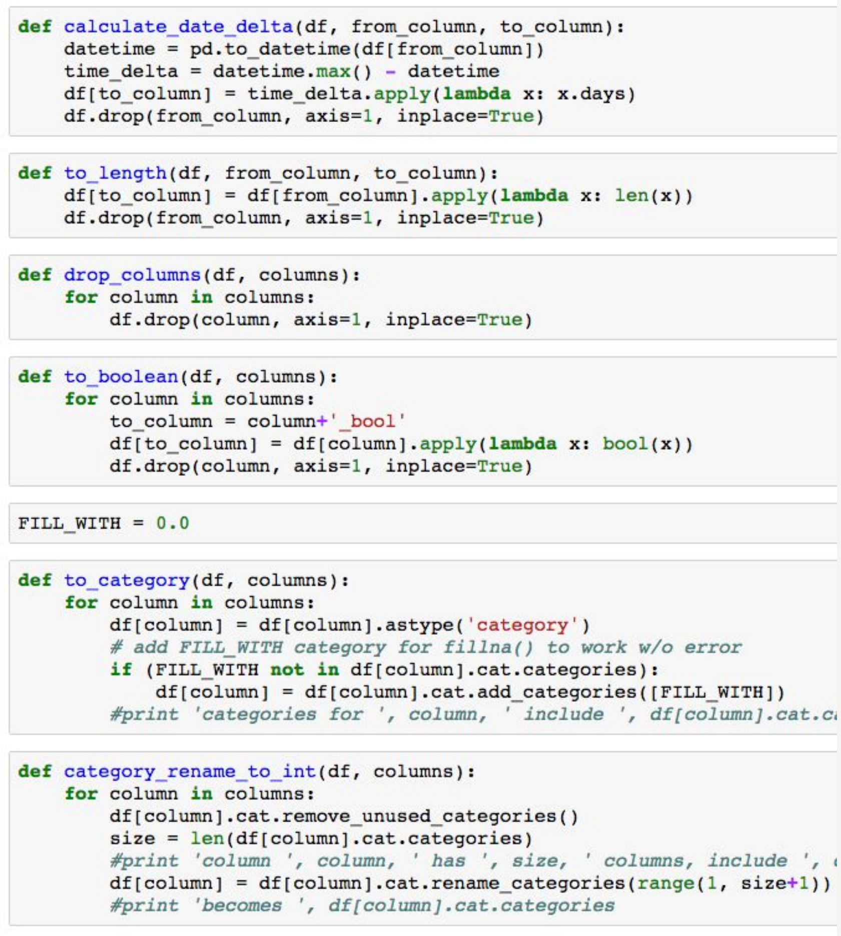
- e.g. business_age

Convert strings to categorical features

- e.g. noise level: {quiet, loud, very loud}.

Drop unused features

- e.g. business_name



```
def calculate_date_delta(df, from_column, to_column):
    datetime = pd.to_datetime(df[from_column])
    time_delta = datetime.max() - datetime
    df[to_column] = time_delta.apply(lambda x: x.days)
    df.drop(from_column, axis=1, inplace=True)
```

```
def to_length(df, from_column, to_column):
    df[to_column] = df[from_column].apply(lambda x: len(x))
    df.drop(from_column, axis=1, inplace=True)
```

```
def drop_columns(df, columns):
    for column in columns:
        df.drop(column, axis=1, inplace=True)
```

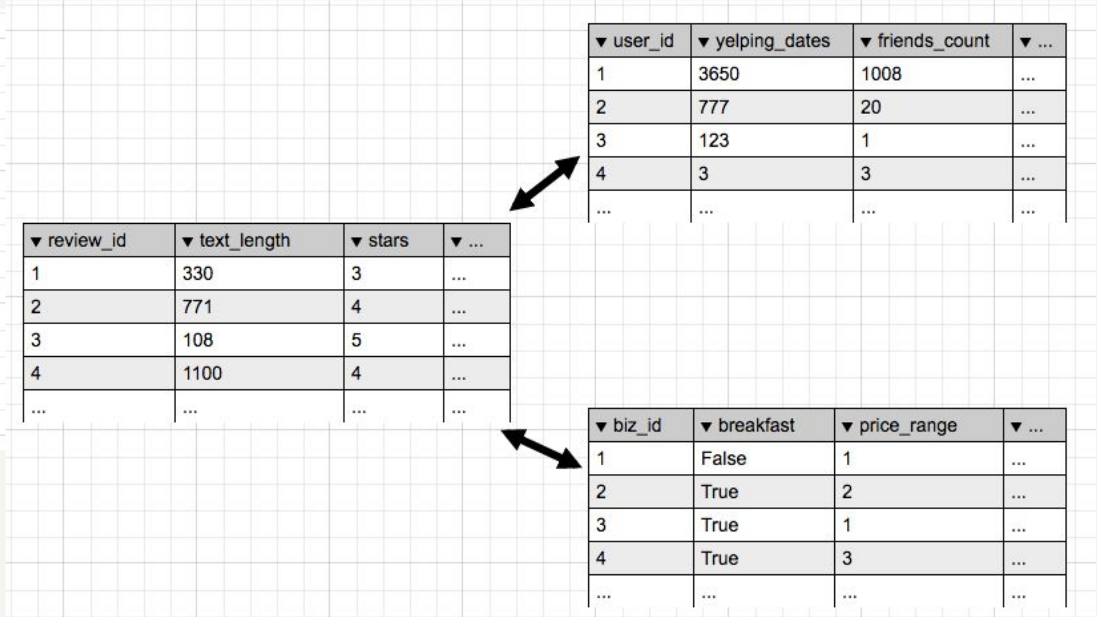
```
def to_boolean(df, columns):
    for column in columns:
        to_column = column+'_bool'
        df[to_column] = df[column].apply(lambda x: bool(x))
        df.drop(column, axis=1, inplace=True)
```

```
FILL_WITH = 0.0
```

```
def to_category(df, columns):
    for column in columns:
        df[column] = df[column].astype('category')
        # add FILL_WITH category for fillna() to work w/o error
        if (FILL_WITH not in df[column].cat.categories):
            df[column] = df[column].cat.add_categories([FILL_WITH])
        #print 'categories for ', column, ' include ', df[column].cat.c
```

```
def category_rename_to_int(df, columns):
    for column in columns:
        df[column].cat.remove_unused_categories()
        size = len(df[column].cat.categories)
        #print 'column ', column, ' has ', size, ' columns, include ',
        df[column] = df[column].cat.rename_categories(range(1, size+1))
        #print 'becomes ', df[column].cat.categories
```

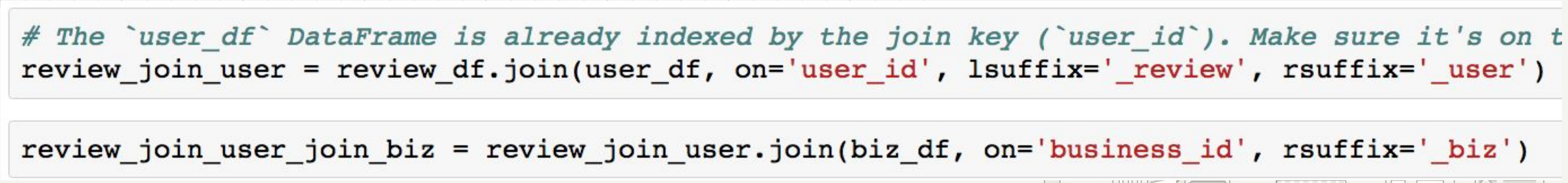
Join tables to populate the features



▼ user_id	▼ yelping_dates	▼ friends_count	▼ ...
1	3650	1008	...
2	777	20	...
3	123	1	...
4	3	3	...
...	...	...	...

▼ review_id	▼ text_length	▼ stars	▼ ...
1	330	3	...
2	771	4	...
3	108	5	...
4	1100	4	...
...	...	...	...

▼ biz_id	▼ breakfast	▼ price_range	▼ ...
1	False	1	...
2	True	2	...
3	True	1	...
4	True	3	...
...	...	...	...



```python
# The `user_df` DataFrame is already indexed by the join key (`user_id`). Make sure it's on t
review_join_user = review_df.join(user_df, on='user_id', lsuffix='_review', rsuffix='_user')
```

```python
review_join_user_join_biz = review_join_user.join(biz_df, on='business_id', rsuffix='_biz')
```

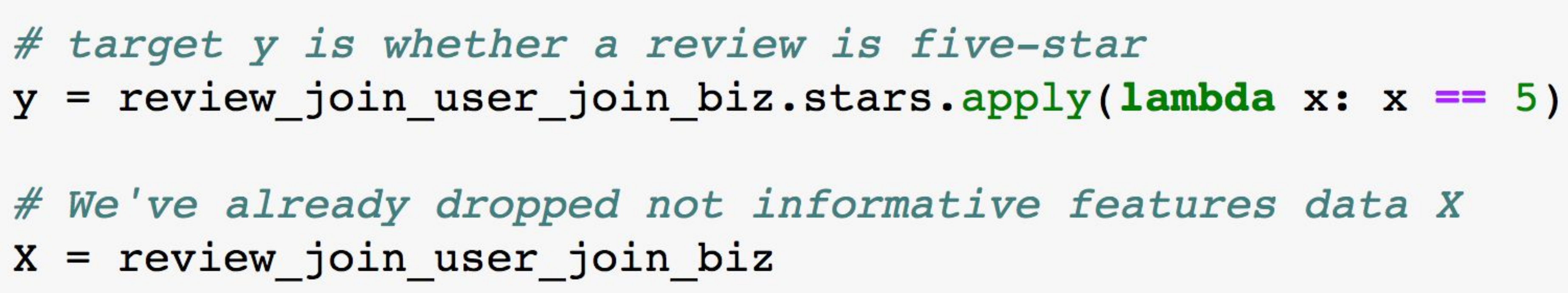
Identify data X and target y

Data X

- All features we gathered from biz, user, review tables

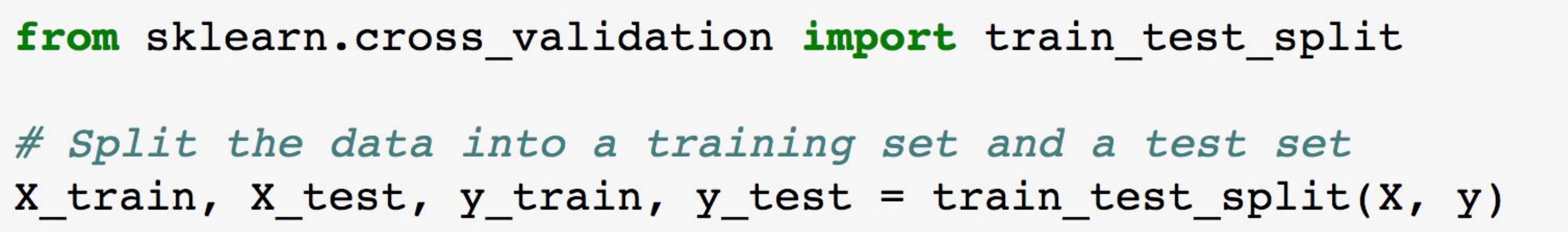
Target y

- What we predict: Whether the review is Five-star or not

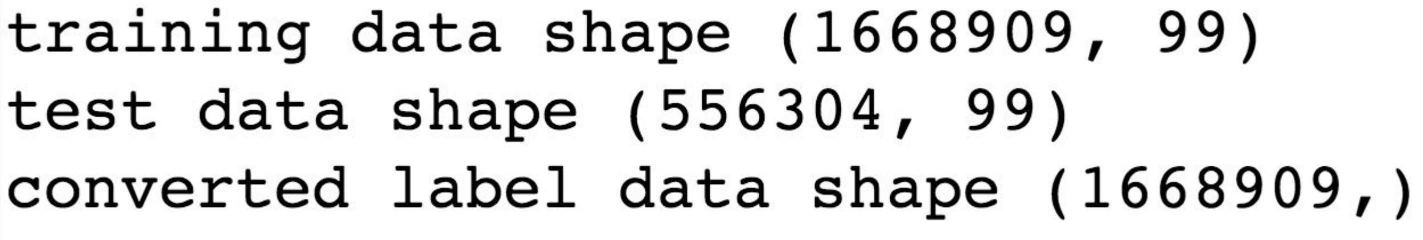


```
# target y is whether a review is five-star
y = review_join_user_join_biz.stars.apply(lambda x: x == 5)

# We've already dropped not informative features data X
X = review_join_user_join_biz
```

Split training set and testing set



```python
from sklearn.cross_validation import train_test_split

# Split the data into a training set and a test set
X_train, X_test, y_train, y_test = train_test_split(X, y)
```



```
training data shape (1668909, 99)
test data shape (556304, 99)
converted label data shape (1668909,)
```

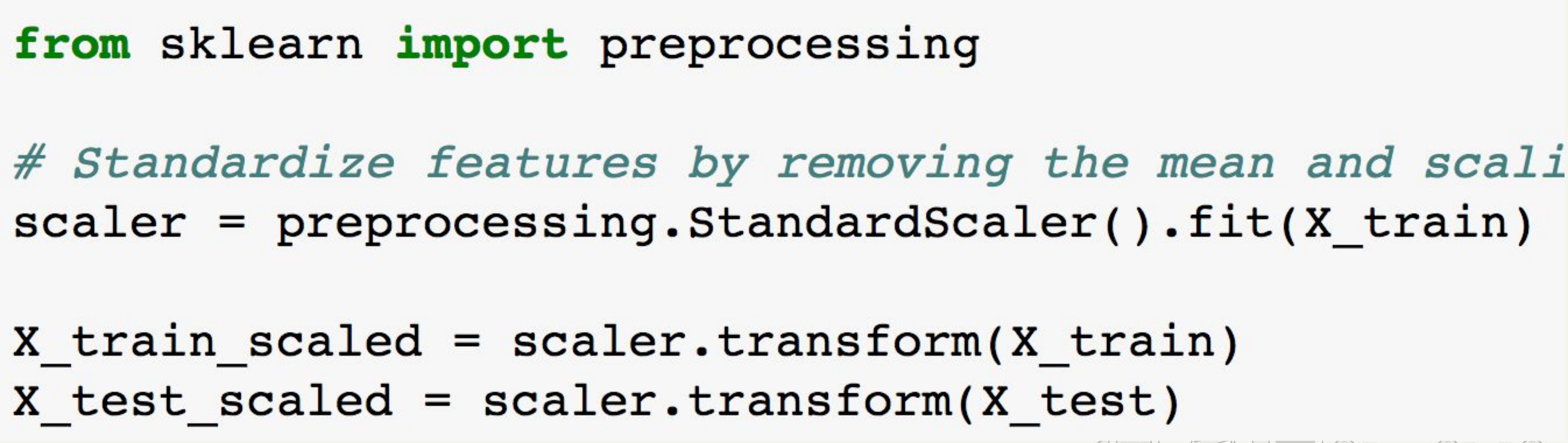
Model the data: Logistic regression

Logistic regression (LR)

- Estimates the probability of a **binary** response
- Here we estimate the probability of a review being five-star

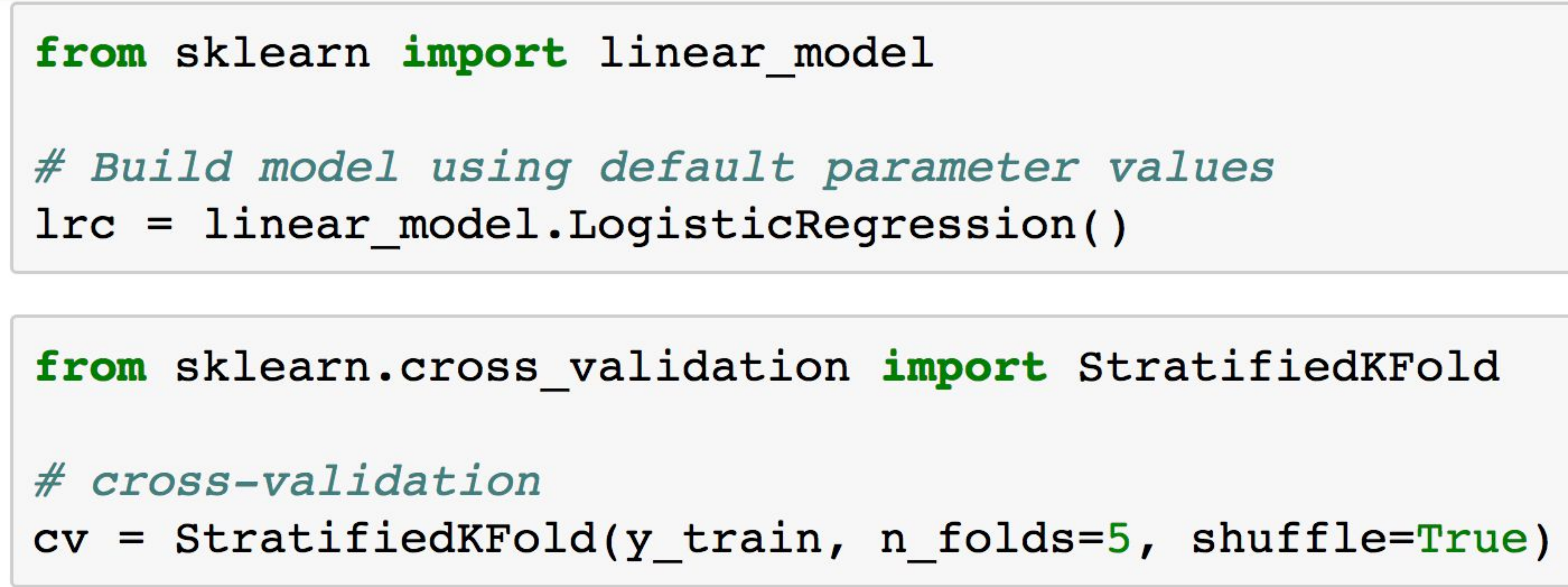
LR: Standardize features

Standardize features by removing the mean and scaling to unit variance



```
from sklearn import preprocessing

# Standardize features by removing the mean and scali
scaler = preprocessing.StandardScaler().fit(X_train)

X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

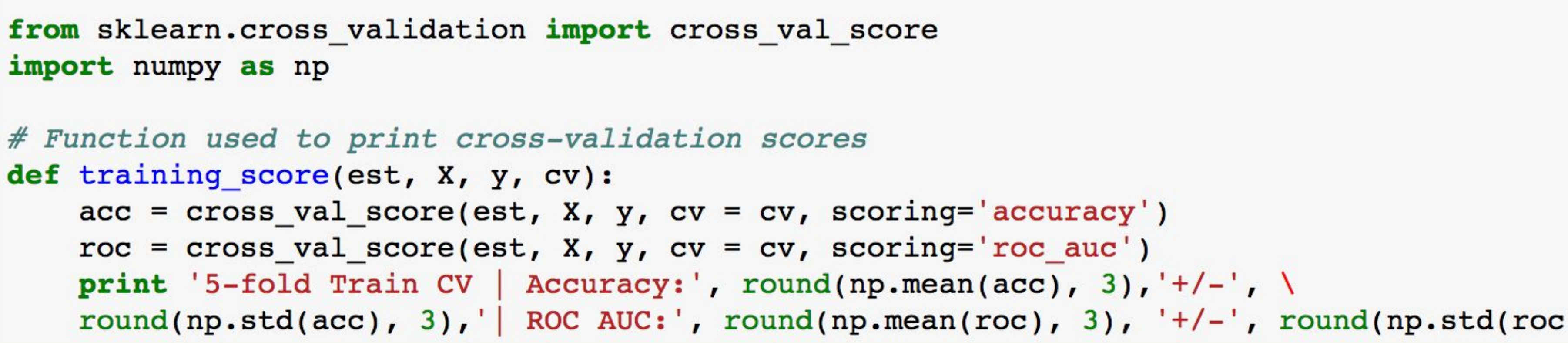
LR: Build model & Cross validation



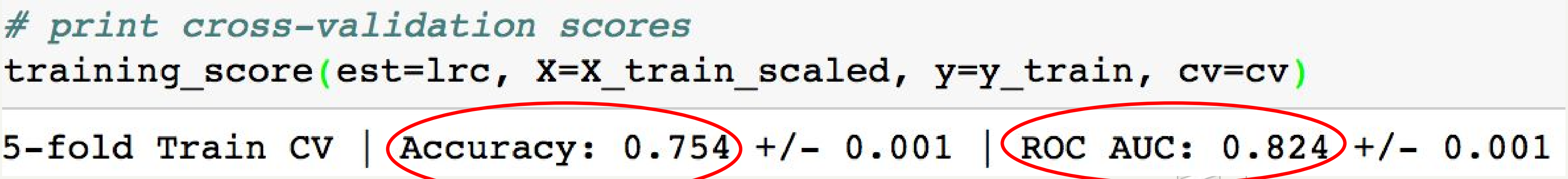
```python
from sklearn import linear_model

# Build model using default parameter values
lrc = linear_model.LogisticRegression()
```

```python
from sklearn.cross_validation import StratifiedKFold

# cross-validation
cv = StratifiedKFold(y_train, n_folds=5, shuffle=True)
```

LR: Build model & Cross validation



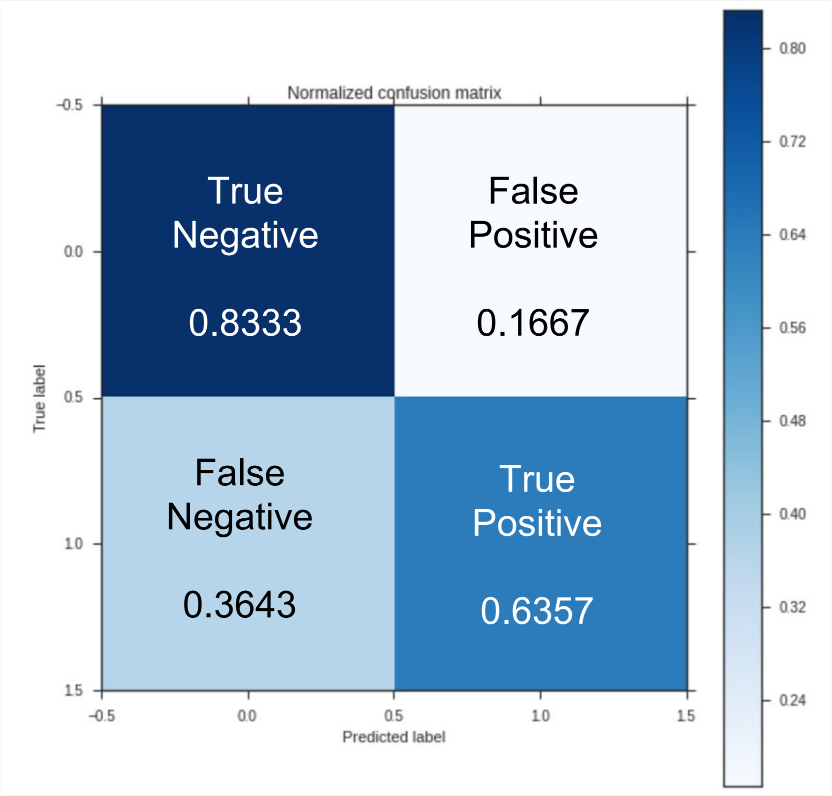
```
from sklearn.cross_validation import cross_val_score
import numpy as np

# Function used to print cross-validation scores
def training_score(est, X, y, cv):
    acc = cross_val_score(est, X, y, cv = cv, scoring='accuracy')
    roc = cross_val_score(est, X, y, cv = cv, scoring='roc_auc')
    print '5-fold Train CV | Accuracy:', round(np.mean(acc), 3),'+/-', \
    round(np.std(acc), 3),'| ROC AUC:', round(np.mean(roc), 3), '+/-', round(np.std(roc
```



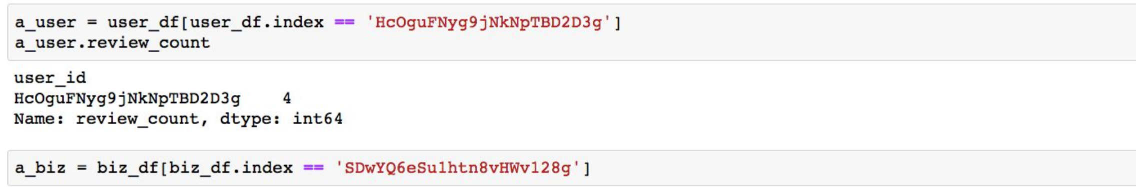
```
# print cross-validation scores
training_score(est=lrc, X=X_train_scaled, y=y_train, cv=cv)
```

```
5-fold Train CV | Accuracy: 0.754 +/- 0.001 | ROC AUC: 0.824 +/- 0.001
```

LR: Evaluation via Confusion Matrix



Make prediction with the model

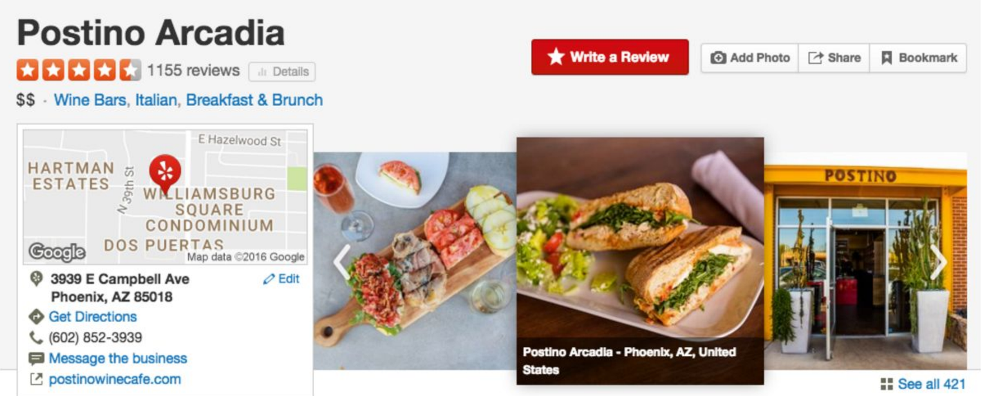


```
a_user = user_df[user_df.index == 'HcOguFNyg9jNkNpTBD2D3g']
a_user.review_count
```

```
user_id
HcOguFNyg9jNkNpTBD2D3g     4
Name: review_count, dtype: int64
```

```
a_biz = biz_df[biz_df.index == 'SDwYQ6eSu1htn8vHWv128g']
```

https://www.yelp.com/biz/postino-arcadia-phoenix



Postino Arcadia

1155 reviews

$$ · Wine Bars, Italian, Breakfast & Brunch

3939 E Campbell Ave
Phoenix, AZ 85018

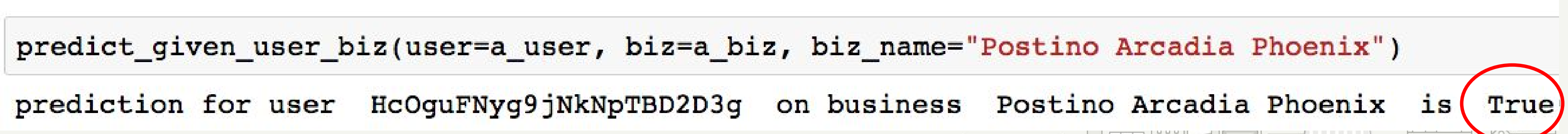
Get Directions

(602) 852-3939

Message the business

postinowinecafe.com

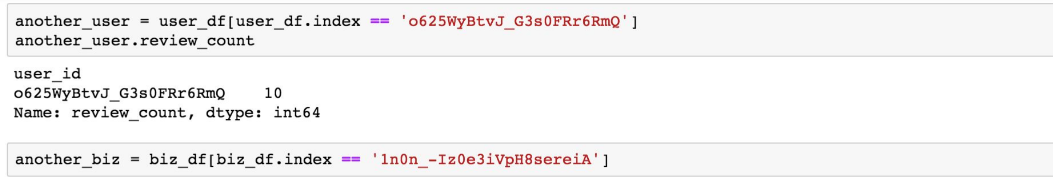
Postino Arcadia - Phoenix, AZ, United States



```
predict_given_user_biz(user=a_user, biz=a_biz, biz_name="Postino Arcadia Phoenix")
```

```
prediction for user  HcOguFNyg9jNkNpTBD2D3g  on business  Postino Arcadia Phoenix  is  True
```

Make prediction with the model

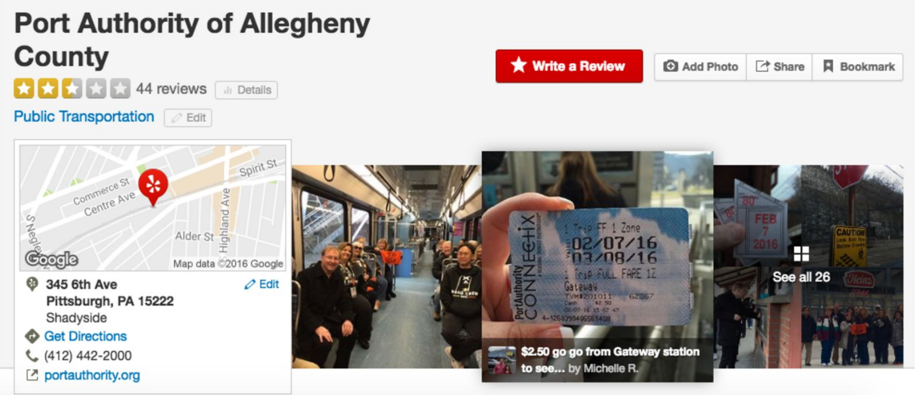


```
another_user = user_df[user_df.index == 'o625WyBtvJ_G3s0FRr6RmQ']
another_user.review_count
```

```
user_id
o625WyBtvJ_G3s0FRr6RmQ    10
Name: review_count, dtype: int64
```

```
another_biz = biz_df[biz_df.index == '1n0n_-Iz0e3iVpH8sereiA']
```

https://www.yelp.com/biz/port-authority-of-allegheny-county-pittsburgh



Port Authority of Allegheny County

44 reviews

Public Transportation

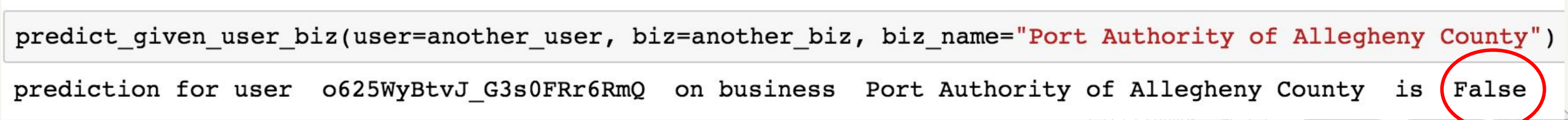
345 6th Ave
Pittsburgh, PA 15222
Shadyside

Get Directions

(412) 442-2000

portauthority.org

$2.50 go go from Gateway station to see... by Michelle R.



```
predict_given_user_biz(user=another_user, biz=another_biz, biz_name="Port Authority of Allegheny County")
```

```
prediction for user  o625WyBtvJ_G3s0FRr6RmQ  on business  Port Authority of Allegheny County  is  False
```

3 Things...

Jupyter Notebook

Scikit-learn

Yelp public dataset

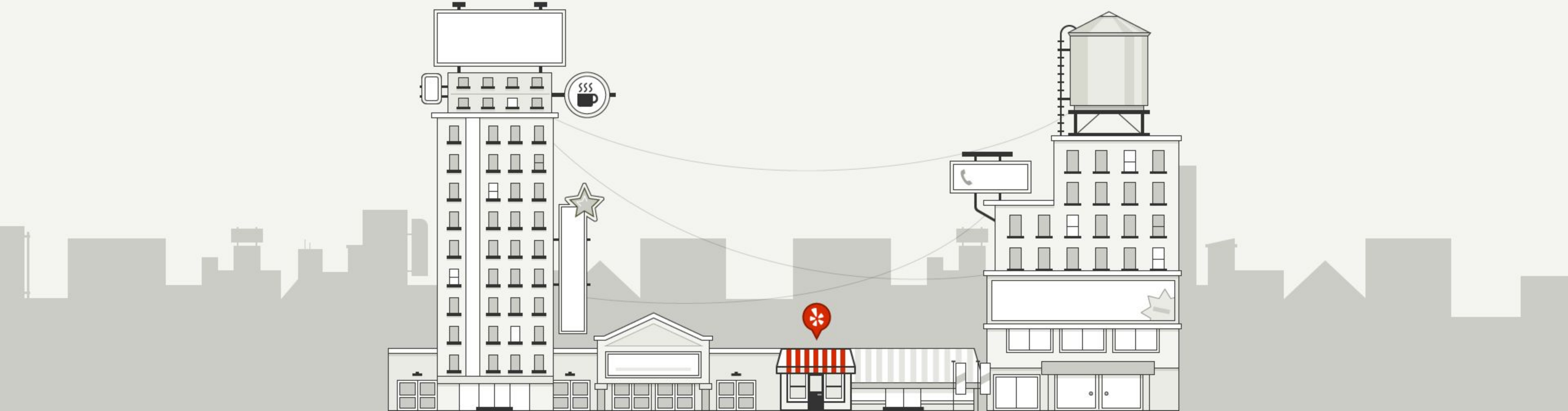
Questions?

Repo: github.com/xun-tang/pyladies_jupyter_demo

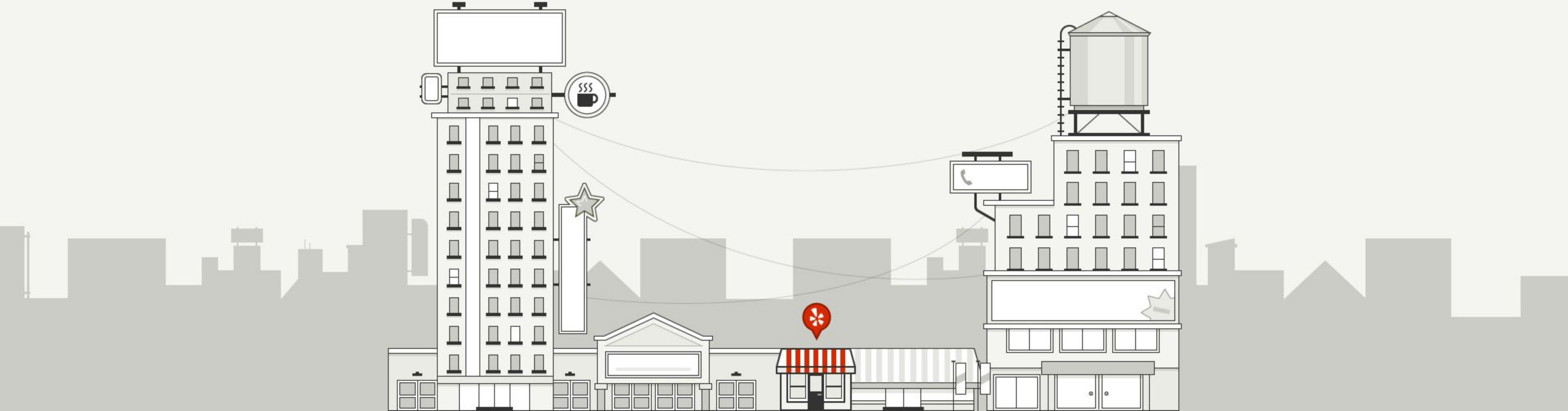
linkedin.com/in/xuntang xun@yelp.com

Yelp Careers: yelp.com/careers/teams/engineering

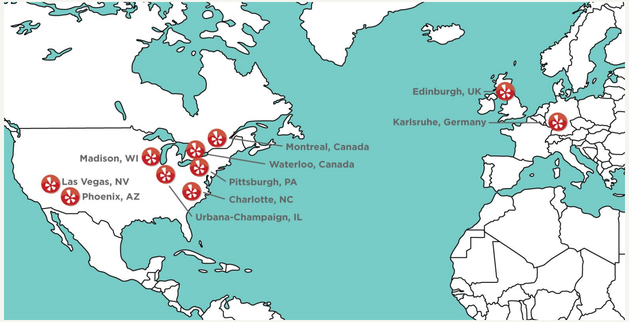
Yelp Dataset Challenge: yelp.com/dataset_challenge/



Backup Slides



yelp.com / dataset_challenge



Academic dataset from 10 cities in 4 countries!

- 77K businesses
- 55K checkin-sets
- 566K business attributes
- 200k photos
- 2.2M reviews
- 552K users
- 3.5M edge social-graph
- 591K tips

Your academic project, research or visualizations, submitted by June 30, 2016

=

$5,000 prize + $1,000 for publication + $500 for presenting*

*See full terms on website