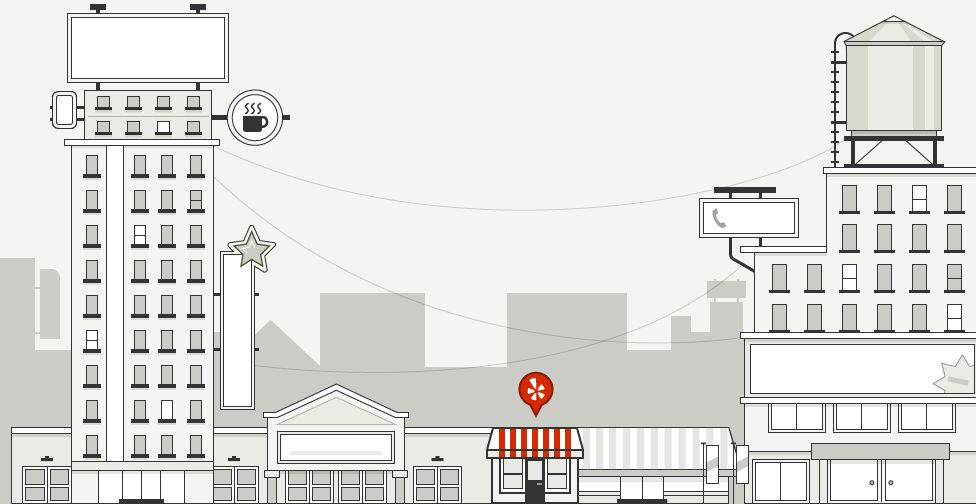


A Hands-on Dive into Making Sense of Real World Data

Xun Tang

Slides: [y/ds-workshop](#)

Code: https://github.com/xun-tang/ds_workshop



Target Audience

If you have done any model training in the past, this workshop is likely too basic for you.



Before The Workshop

1. Download the dataset
 - download from [Yelp Open Dataset](#) and choose the json format
 - unzip to your desired folder: `tar -xzvf yelp_dataset.tar`
2. Install python in your local computer and all the required libraries, via Anaconda
 - download from [Anaconda site](#) and choose the version for python 3.7
 - note all libraries we need in the workshop are pre-packaged in Anaconda so you don't need to `pip install bluh`



How You Can Spend the Next Hour

You can directly run the notebook on your computer if you have the dataset downloaded and environment set up. Don't worry if you don't.

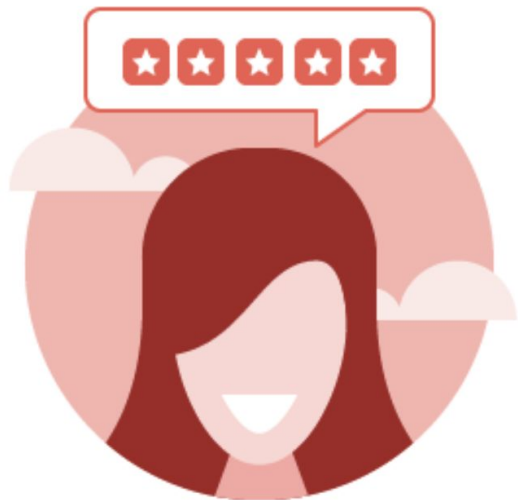
We will make this workshop as interactive as possible by walking through my slides (40+ pages) full of screenshots of the notebook, and explain what each step is about. Feel free to raise your hand to ask questions any time during the workshop.

Github ipynb [renderer](#).



Yelp Open Dataset

[yelp.com/dataset](https://www.yelp.com/dataset)

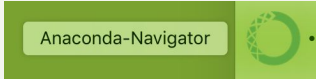


6,685,900 reviews **192,609 businesses**



How to run the notebook in your browser

- clone the repo

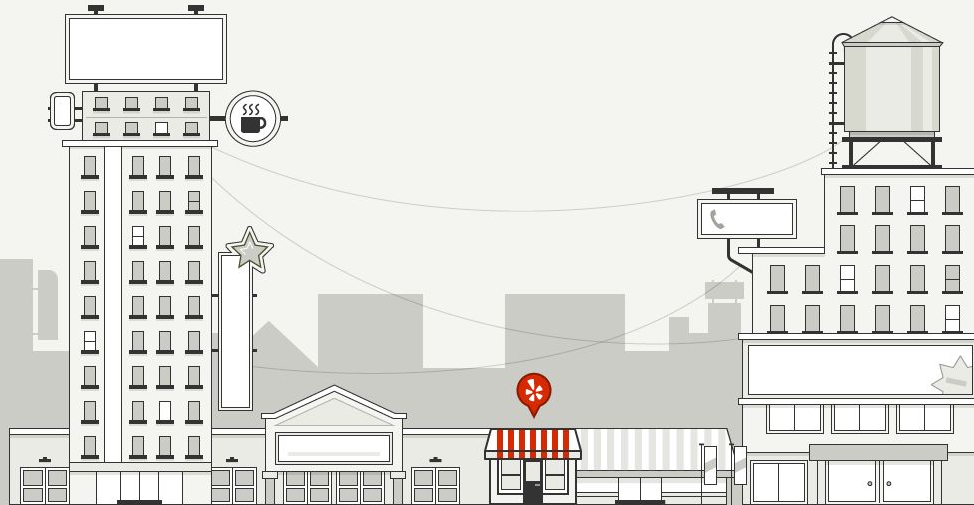
```
git clone https://github.com/xun-tang/ds_workshop.git
```
- run the anaconda-navigator app 
- click to launch jupyter notebook (will open in your default browser)
- navigate to the folder of checked-out repo
- double click to launch the notebook file ending with `.ipynb`

Why not run in y/jupyterhub directly?




Will Your Next **yelp** Review

Be a  Review?

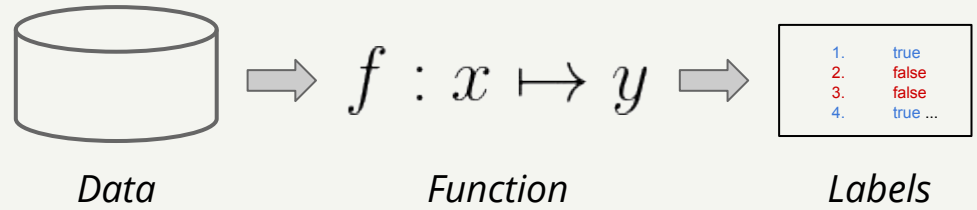


Given a user and a business,
predict whether the user will
write a  review for
the business.

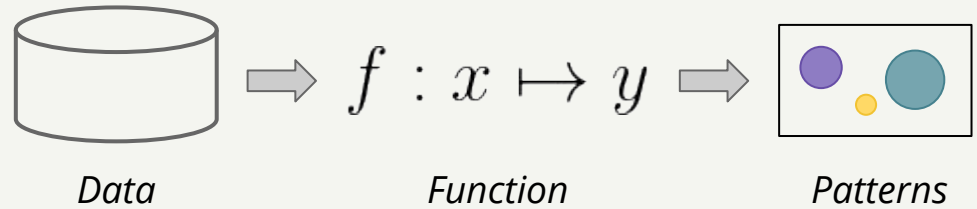
yelp  Open Dataset



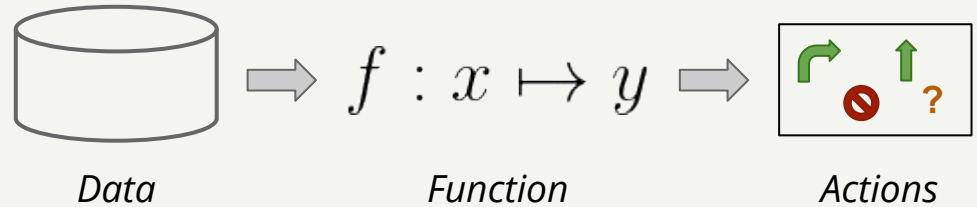
Supervised learning



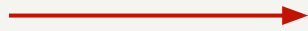
Unsupervised learning



Reinforcement learning



**Supervised
learning**



$$f : x \mapsto y$$

"Learn this function"

**Unsupervised
learning**



$$f : x \mapsto y$$

"Learn this function"

**Reinforcement
learning**



$$f : x \mapsto y$$

"Learn this function"

We call the learning process *training* or *fitting*.



Supervised learning



$$f : x \mapsto y$$

"Learn this function"

Task

Labels (function output)

Classification

Binary (boolean) or categorical



Regression

Continuous (float) value 

Ranking

Ordered items



To Solve a Data Science Problem

Step 1: Load the Data

Step 2: Explore and Visualize the Data

Step 3: Generate the Features

Step 4: Train a Model

Step 5: Evaluate the Model

Step 6 & Beyond: Iterate Through the Process



Step 1

Load the Data



Store Data in Pandas DataFrames



```
In [1]: %%time
import pandas as pd

CHUNK_SIZE = 10000
PATH = '~/Desktop/yelp_dataset/'
```

```
CPU times: user 407 ms, sys: 203 ms, total: 610 ms
Wall time: 1.77 s
```

```
In [2]: %%time
review_df = pd.concat(pd.read_json(PATH + 'review.json', lines=True, chunksize=CHUNK_SIZE))
review_df = review_df.set_index('review_id')
```

```
CPU times: user 1min 26s, sys: 19.1 s, total: 1min 45s
Wall time: 1min 46s
```

```
In [3]: %%time
user_df = pd.concat(pd.read_json(PATH + 'user.json', lines=True, chunksize=CHUNK_SIZE))
user_df = user_df.set_index('user_id')
```

```
CPU times: user 40.9 s, sys: 10.5 s, total: 51.4 s
Wall time: 50.9 s
```

```
In [4]: %%time
biz_df = pd.concat(pd.read_json(PATH + 'business.json', lines=True, chunksize=CHUNK_SIZE))
biz_df = biz_df.set_index('business_id')
```

```
CPU times: user 5.84 s, sys: 1.05 s, total: 6.89 s
Wall time: 6.04 s
```



What's in a Review DataFrame?

`review_df.head()`: Print top rows in the data frame.

`review_df.describe()`: Generate various summary statistics, mean, max, count, etc.

```
review_df.head()
```

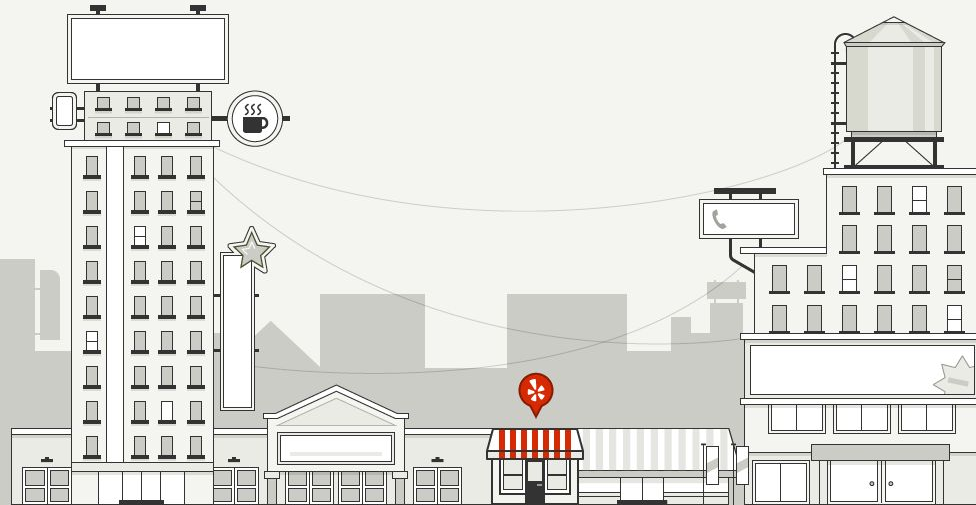
review_id	funny	user_id	text	business_id	stars	useful	type	cool	datetime	year
NxL8SIC5yqOdnIXCg18lBg	0	KpkOkG6Rlf4Ra25Lhxf1A	If you enjoy service by someone who is as comp...	2aFiy99vNLkICx3T_tGS9A	5	0	review	0	2011-10-10	2011
pXbbigOXvLuTi_SPs1hQEQ	0	bQ7fQq1otr9hKX-gXRsrG	After being on the phone with Verizon Wireless...	2aFiy99vNLkICx3T_tGS9A	5	1	review	0	2010-12-29	2010
wsIW2Lu4NYyIb1jEapAGsw	0	r1NUhdNmL6yU9Bn-Yx6FTw	Great service! Corey is very service oriented....	2aFiy99vNLkICx3T_tGS9A	5	0	review	0	2011-04-29	2011
GP6YEearUWrzPtQYSF1vVg	0	aW3ix1KNZAvoM8q-WghA3Q	Highly recommended. Went in yesterday looking ...	2LfluF3_sX6uwe-IR-P0JQ	5	0	review	1	2014-07-14	2014
25RIYGq2s5qShi-pn3ufVA	0	YOo-Cip8HqvKp_p9nEGphw	I walked in here looking for a specific piece ...	2LfluF3_sX6uwe-IR-P0JQ	4	0	review	0	2014-01-15	2014

```
biz_df.describe()
```

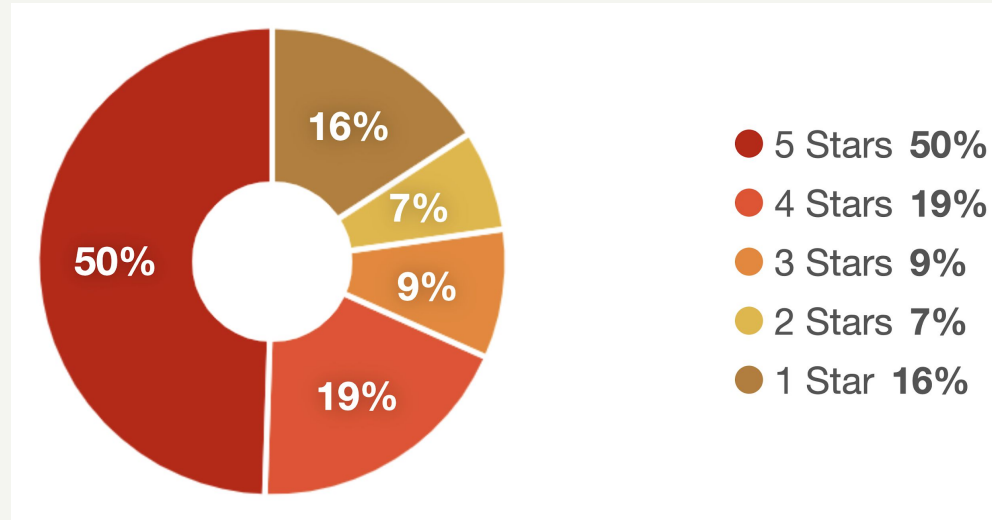


Step 2

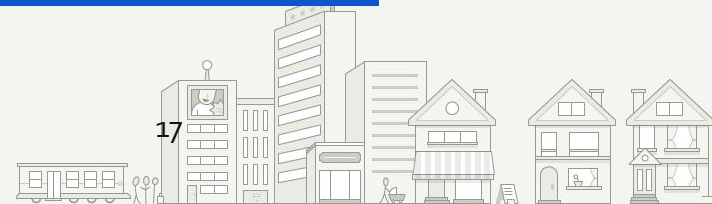
Explore and Visualize the Data



Review Star Rating Distribution Published on Yelp's Factsheet



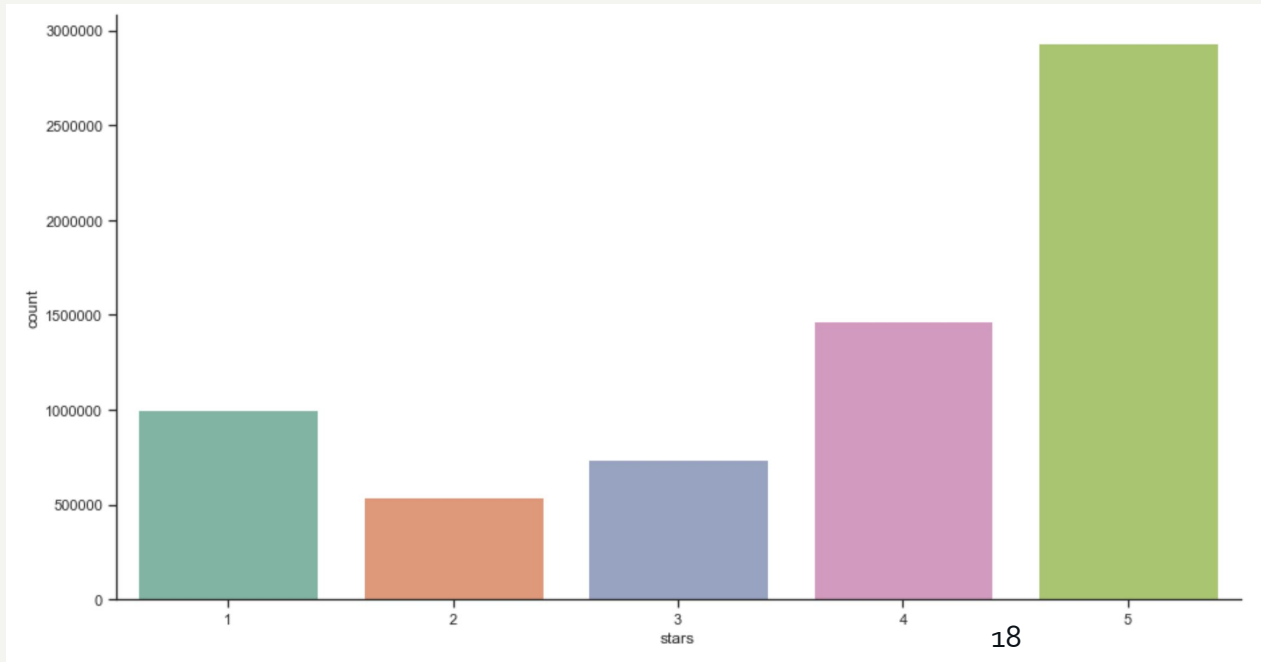
Source: <https://www.yelp.com/factsheet>



Plot Review Star Rating Distribution from Open Dataset

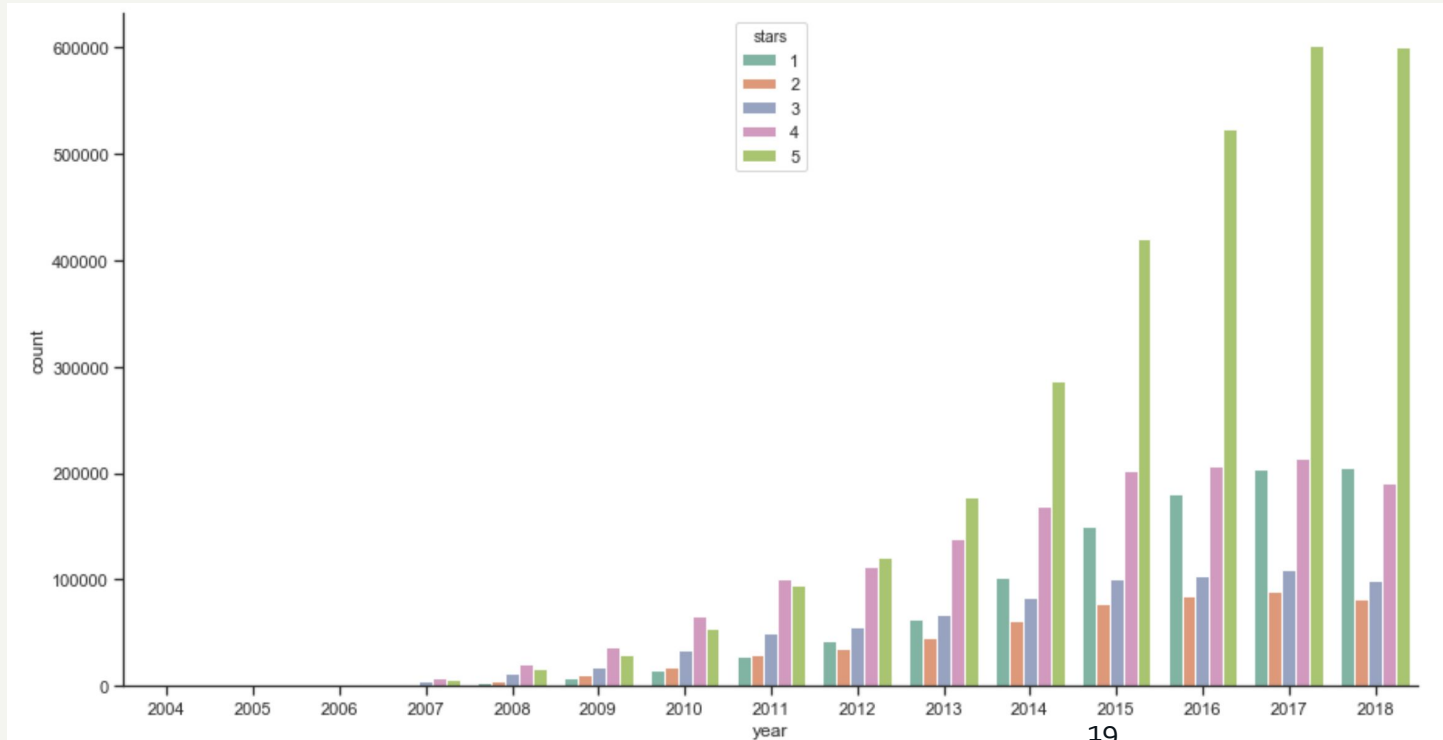
```
import seaborn as sns
%matplotlib inline
```

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



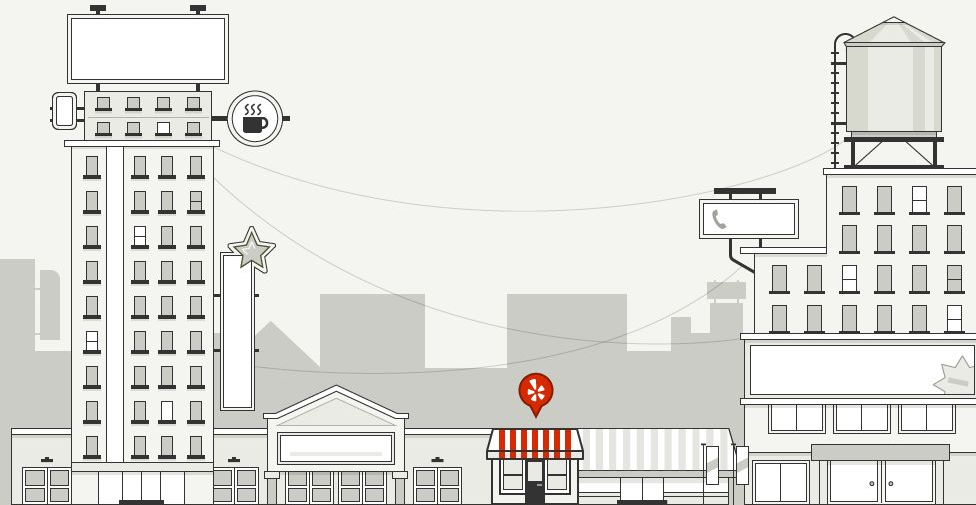
Plot Star Ratings by Year

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



Step 3

Generate the Features



For Example..

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

```
# compute days in between date and max value in date
def calculate_date_delta(df, column):
    to_column = column + '_delta'
    datetime = pd.to_datetime(df[column])
    time_delta = datetime.max() - datetime
    df[to_column] = time_delta.apply(lambda x: x.days)
    df.drop(column, axis=1, inplace=True)
```

```
# compute length of string
def to_length(df, column):
    to_column = column + '_len'
    df[to_column] = df[column].apply(lambda x: len(x))
    df.drop(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)
```

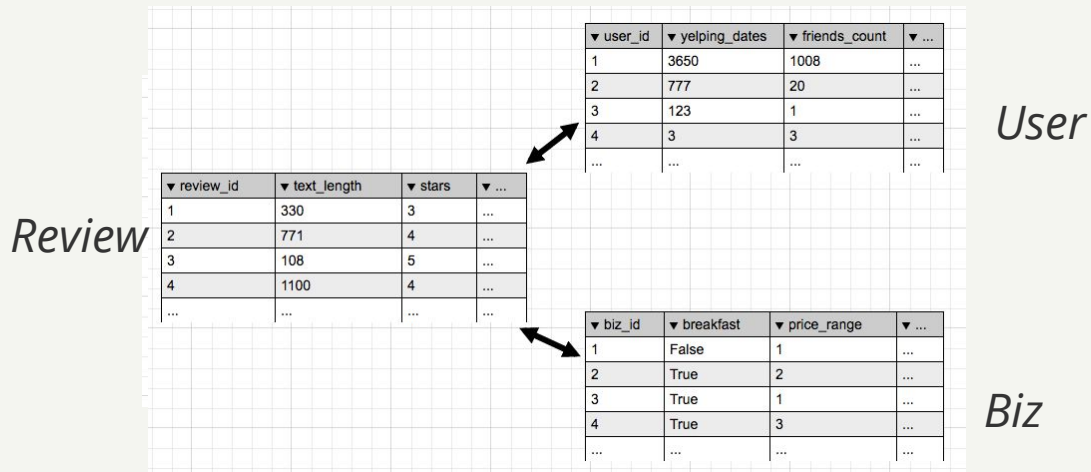
```
FILL_WITH = 0.0
```

```
def to_category(df, columns):
    for column in columns:
        df[column] = df[column].astype('category')
        # add FILL WITH category for fillna()
        if (FILL_WITH not in df[column].cat.categories):
            df[column] = df[column].cat.add_categories([FILL_WITH])
        print(f'categories for {key} include {df[key].cat.categories}')
```

```
def category_rename_to_int(df, columns):
    for column in columns:
        df[column].cat.remove_unused_categories()
        size = len(df[column].cat.categories)
        print(f'column {column} has {size} columns, including {df[column].cat.categories}')
        df[column] = df[column].cat.rename_categories(range(1, size+1))
        print(f'=> {df[column].cat.categories}')
```



Join DataFrames to Populate the Features



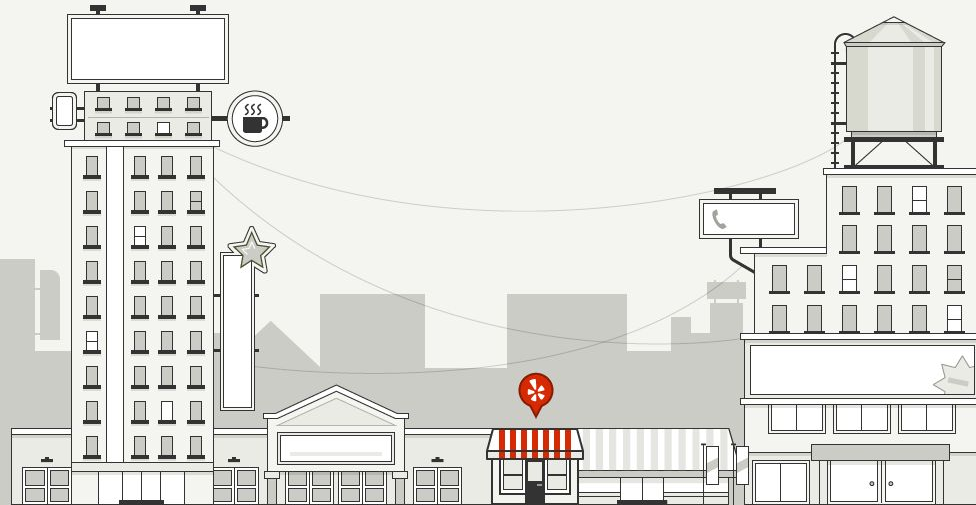
```
# The `user_df` DataFrame is already indexed by the join key (`user_id`). Make sure it's on the right side of join.  
review_join_user = review_df.join(user_df, on='user_id')
```

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



Step 4

Train a Model



Arrange Data into a Feature Matrix and a Target Array

Feature matrix (X)

All features generated from biz, user, review dataframes

Target array (y)

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

```
# Target y is whether a review is five-star (True / False)
y = review_join_user_join_biz.review_stars.apply(lambda x: x == 5)

# Exclude the `stars` columns from the feature matrix, since it is the target
X = review_join_user_join_biz
review_join_user_join_biz.drop('review_stars', axis=1, inplace=True)
```



Split Training and Testing Set

Training set: used for an machine learning algorithm to train from

Testing set: used to to estimate / evaluate how well the model has been trained

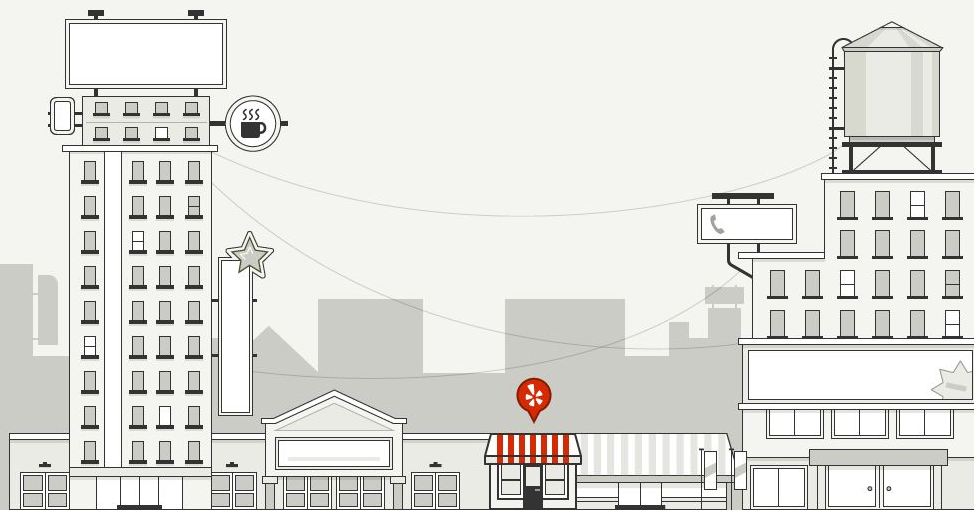
Split them s.t. we don't evaluate on the same dataset we train from

```
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 (3552672, 109)  
test data shape (1184225, 109)  
converted label data shape (3552672,)
```



What Model to Use?



Choose the model to use

Logistic Regression (LR)

Estimates the prob. of a **binary** response based on the features

Here we estimate the prob. of a review being five-star

```
from sklearn import linear_model

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



Normalize the Features

Standardize features by removing the mean and scaling to unit variance

Logistic Regression requires all features normalized

```
from sklearn import preprocessing

scaler = preprocessing.StandardScaler().fit(X_train)

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



Modeling

Key methods:

fit: Fit the model according to the given training data

predict: Predict class labels for samples in data

score: Returns the mean accuracy on the given test data and labels



```
from sklearn import linear_model

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

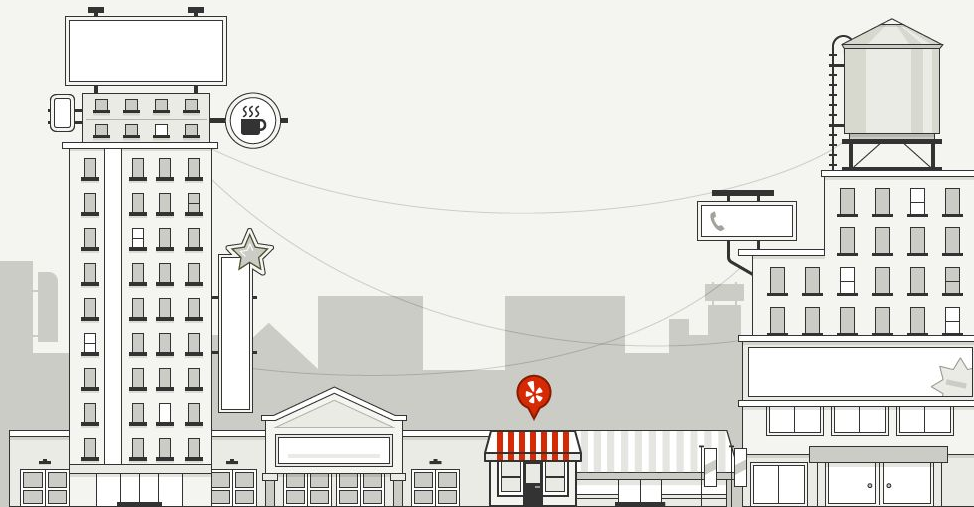
```
%%time
lrc_fit = lrc.fit(X_train_scaled, y_train)
y_pred = lrc_fit.predict(X_test_scaled)
```

```
CPU times: user 2min 20s, sys: 4.82 s, total: 2min 25s
Wall time: 27.4 s
```



Step 5

Evaluate the Model



Does It Work?

Given users' past reviews on Yelp

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

Will it be a  review?
(True / False)



Given users' past reviews on Yelp

```
user1 = user_df[user_df.index == 'kEtR1ZVL3Xr-tEX7lg16dQ']  
#print user1.review_count  
print user1.average_stars
```

```
user_id  
kEtR1ZVL3Xr-tEX7lg16dQ    4.96  
Name: average_stars, dtype: float64
```

```
user2 = user_df[user_df.index == 'Hj20fg3vyzKnJwnLn_rMqw']  
#print user2.review_count  
print user2.average_stars
```

```
user_id  
Hj20fg3vyzKnJwnLn_rMqw    4.55  
Name: average_stars, dtype: float64
```

```
user3 = user_df[user_df.index == 'om5ZiponkpRqUNa3pVPiRg']  
#print user2.review_count  
print user3.average_stars
```

```
user_id  
om5ZiponkpRqUNa3pVPiRg    3.94  
Name: average_stars, dtype: float64
```



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

Postino Arcadia Claimed



1169 reviews



★ Write a Review

📷 Add Photo

🔗 Share

🔖 Bookmark

\$\$ · Wine Bars, Italian, Breakfast & Brunch

3939 E Campbell Ave
Phoenix, AZ 85018

Get Directions
(602) 852-3939
postinowinecafe.com
Message the business
Send to your Phone

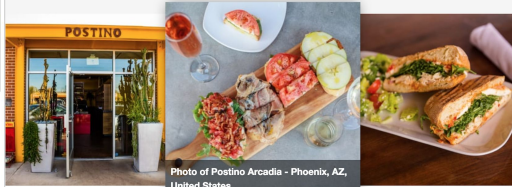


Photo of Postino Arcadia - Phoenix, AZ, United States

See all 520

biz1

biz2

Port Authority of Allegheny County Unclaimed



53 reviews



★ Write a Review

📷 Add Photo

🔗 Share

🔖 Bookmark

Public Transportation Edit

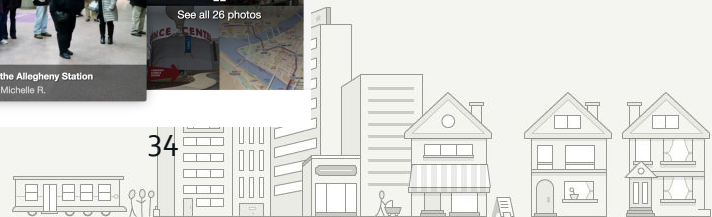
345 6th Ave
Pittsburgh, PA 15222
Shadyside

Get Directions
(412) 442-2000
portauthority.org
Send to your Phone



At the Allegheny Station by Michelle R.

34



Will it be a review?

```
predict_given_user_biz(user=user1, biz=biz1, review_df=review_df)
predict_given_user_biz(user=user2, biz=biz1, review_df=review_df)
predict_given_user_biz(user=user3, biz=biz1, review_df=review_df)
```

```
True, with probability [False, True] == [0.08 0.92]
True, with probability [False, True] == [0.22 0.78]
False, with probability [False, True] == [0.63 0.37]
```

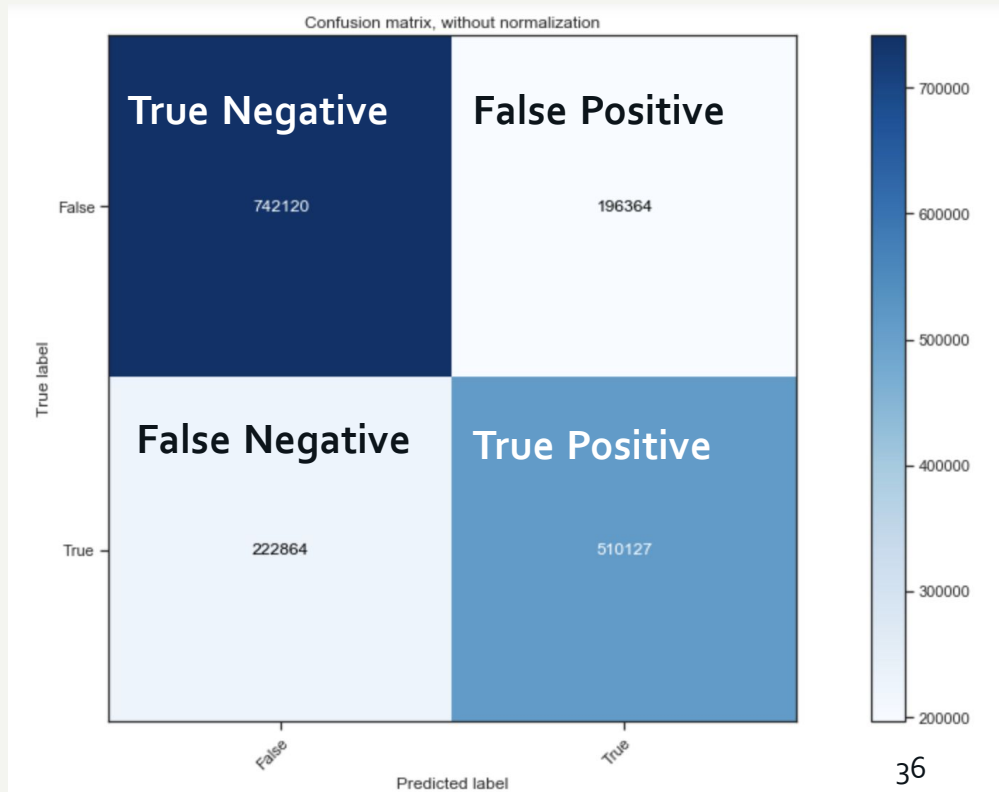
Make predictions for user[1,2,3]'s review on biz2

```
predict_given_user_biz(user=user1, biz=biz2, review_df=review_df)
predict_given_user_biz(user=user2, biz=biz2, review_df=review_df)
predict_given_user_biz(user=user3, biz=biz2, review_df=review_df)
```

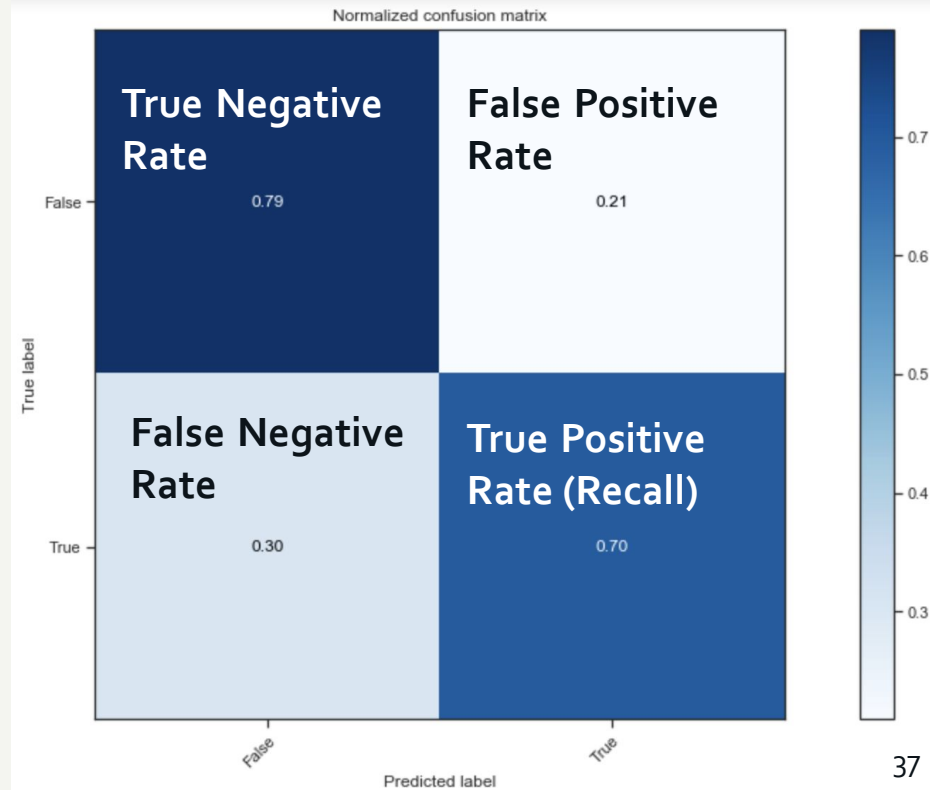
```
True, with probability [False, True] == [0.46 0.54]
False, with probability [False, True] == [0.75 0.25]
False, with probability [False, True] == [0.95 0.05]
```



Confusion Matrix



Normalized Confusion Matrix



Cross Validation

Holding out a portion of the training data for model validation, and do this for K_FOLDS

Ensure that the model does not overfit the training data

Select optimal model parameters

```
from sklearn.model_selection import cross_validate
import numpy as np

# Function used to calculate and print cross-validation scores
def training_score(est, X, y, cv):
    scores = cross_validate(est, X, y, cv=cv, scoring=['accuracy', 'roc_auc'])
```



Accuracy

Percentage of labels correctly predicted. The higher the better.

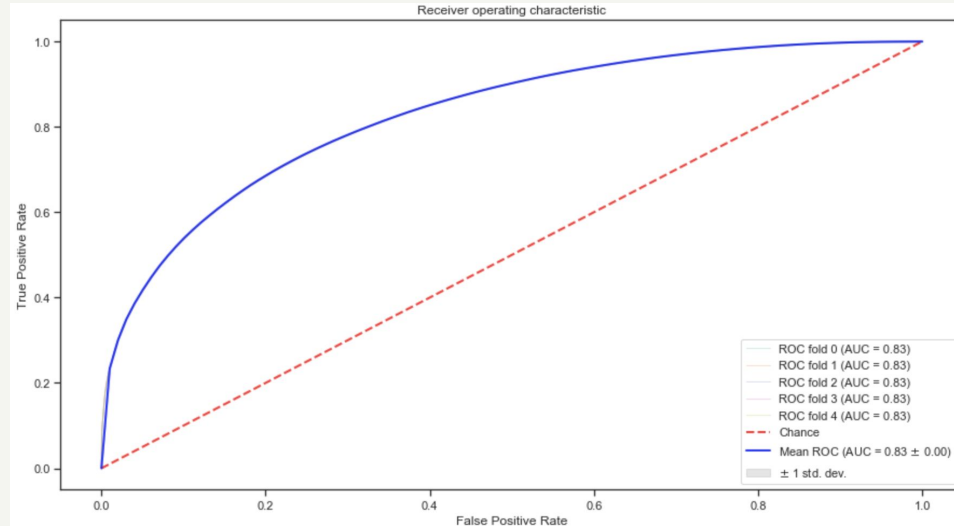
```
training_score(est=lrc, X=X_train_scaled, y=y_train, cv=K_FOLD)
```

```
5-fold Train Cross Validation | Accuracy: 0.749 +/- 0.0 | ROC A
```



ROC AUC

ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. ROC AUC ("Area Under Curve"). The higher the better.



Step 6 & Beyond Iterate Through the Process

