Welcome!
Please copy the content on the usb drive to your laptop, or download from sources

What's in the usb drive:

1. A software platform we'll run ipython notebook on. Please copy & install the version suitable to your OS (Windows, Mac, Linux). https://www.anaconda.com/download/

2. A dataset we'll play with in the second half of the workshop. https://www.yelp.com/dataset/

3. A sample source code we'll cover in the second half of the workshop. https://goo.gl/WgoeUL

GRACE HOPPER CELEBRATION

ANITA B.ORG

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

Jamie Whitacre | @JupyterJW

Xun Tang | @whoisxun | xun@yelp.com

Association for Computing Machinery

#GHC17

Plan for Today

-1-

Install Jupyter

-2-

Learn how to use it

-3-

Complete a machine learning case study using the Jupyter Notebook & Yelp data

Jupyter is great for learning and doing Python!
Basic knowledge of Python is assumed today as is high school level math and statistics

By the End of the Workshop, You Will Be Able To. . .

- Explain the value of using Jupyter Notebooks for data analysis as an individual or on a team.
- Use Pandas and other Python libraries to perform common data analysis and machine learning tasks.
- Solve a simple modeling problem adapted from a real-world problem and data set.

About us

Jamie

Xun

Jupyter Notebook

- **Is the evolution of the *IPython Notebook* into a language agnostic computing environment supporting 60+ programming languages including Python, R, Julia, and many others.**

Jupyter Protocol is Language Agnostic

Jupyter Notebook

- **Is the evolution of the *IPython Notebook* into a language agnostic computing environment supporting 60+ programming languages including Python, R, Julia, and many others.**

- **A digital "document" that gives you a way to record your thoughts, CODE, and visualizations in a single place**

Jupyter Notebook

- Is the evolution of the *IPython Notebook* into a language agnostic computing environment supporting 60+ programming languages including Python, R, Julia, and many others.
- A digital "document" that gives you a way to record your thoughts, CODE, and visualizations in a single place
- A tool to explore, understand, communicate, and tell stories about your data in a systematic and reproducible way

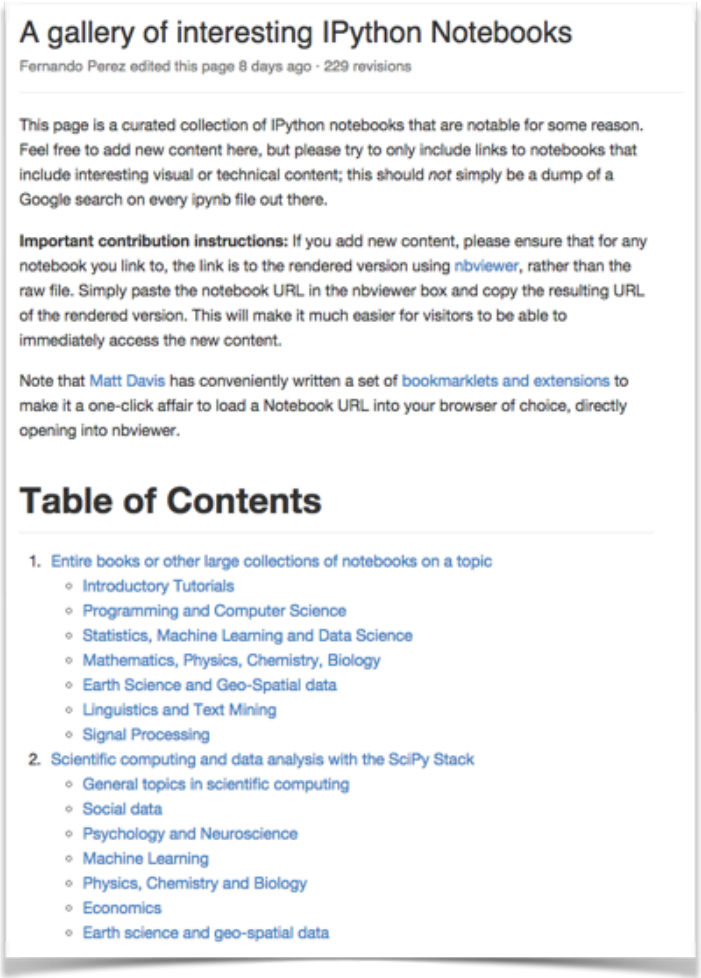
Jupyter Notebook

- Is the evolution of the *IPython Notebook* into a language agnostic computing environment supporting 60+ programming languages including Python, R, Julia, and many others.
- A digital "document" that gives you a way to record your thoughts, CODE, and visualizations in a single place
- A tool to explore, understand, communicate, and tell stories about your data in a systematic and reproducible way
- Provides access to many, many Python libraries

A Tool for Interactive Computing

- A dialogue between the human and the computer.
- Assemble ideas using the computer as playground, as “data microscope.”
- A way to assemble the building blocks of scientific computing and data science.

Learn Through Examples: Gallery of Interesting Jupyter Notebooks



A gallery of interesting IPython Notebooks

Fernando Perez edited this page 8 days ago · 229 revisions

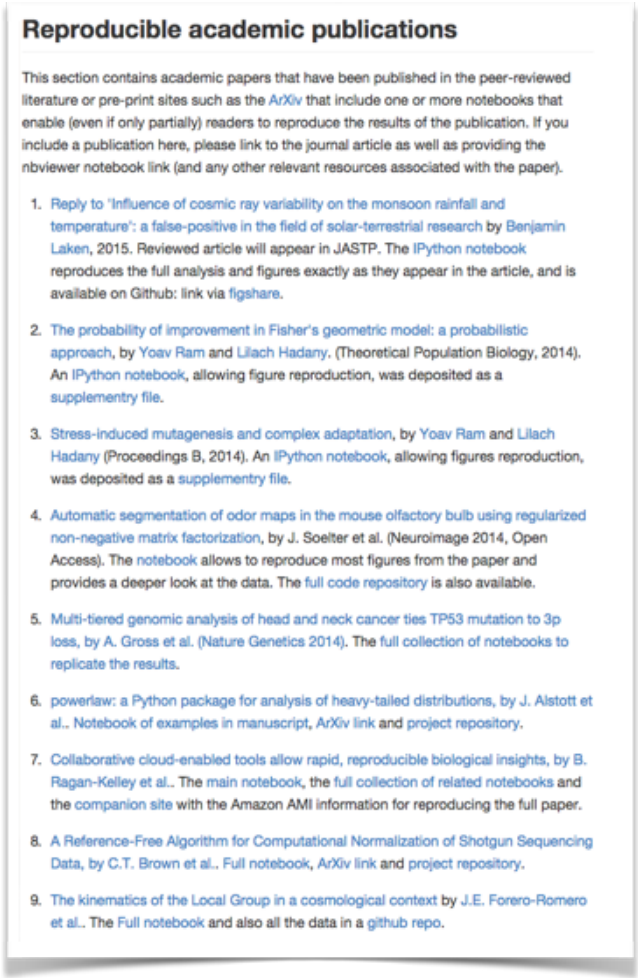
This page is a curated collection of IPython notebooks that are notable for some reason. Feel free to add new content here, but please try to only include links to notebooks that include interesting visual or technical content; this should *not* simply be a dump of a Google search on every ipynb file out there.

Important contribution instructions: If you add new content, please ensure that for any notebook you link to, the link is to the rendered version using nbviewer, rather than the raw file. Simply paste the notebook URL in the nbviewer box and copy the resulting URL of the rendered version. This will make it much easier for visitors to be able to immediately access the new content.

Note that Matt Davis has conveniently written a set of bookmarklets and extensions to make it a one-click affair to load a Notebook URL into your browser of choice, directly opening into nbviewer.

Table of Contents

1. Entire books or other large collections of notebooks on a topic
   - Introductory Tutorials
   - Programming and Computer Science
   - Statistics, Machine Learning and Data Science
   - Mathematics, Physics, Chemistry, Biology
   - Earth Science and Geo-Spatial data
   - Linguistics and Text Mining
   - Signal Processing
2. Scientific computing and data analysis with the SciPy Stack
   - General topics in scientific computing
   - Social data
   - Psychology and Neuroscience
   - Machine Learning
   - Physics, Chemistry and Biology
   - Economics
   - Earth science and geo-spatial data



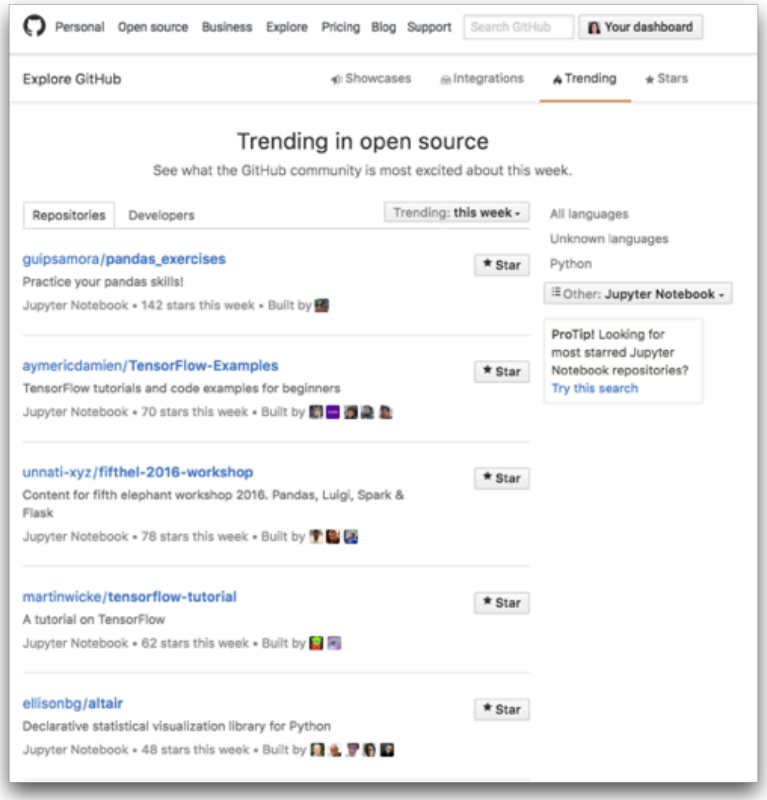
Reproducible academic publications

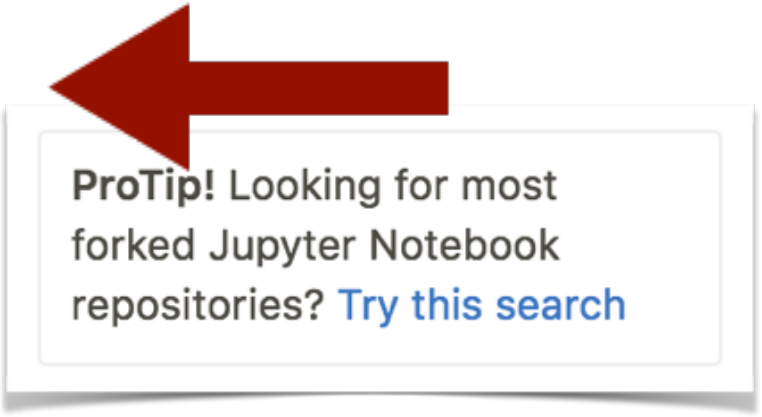
This section contains academic papers that have been published in the peer-reviewed literature or pre-print sites such as the ArXiv that include one or more notebooks that enable (even if only partially) readers to reproduce the results of the publication. If you include a publication here, please link to the journal article as well as providing the nbviewer notebook link (and any other relevant resources associated with the paper).

1. Reply to 'Influence of cosmic ray variability on the monsoon rainfall and temperature': a false-positive in the field of solar-terrestrial research by Benjamin Laken, 2015. Reviewed article will appear in JASTP. The IPython notebook reproduces the full analysis and figures exactly as they appear in the article, and is available on Github: link via figshare.
2. The probability of improvement in Fisher's geometric model: a probabilistic approach, by Yoav Ram and Lilach Hadany. (Theoretical Population Biology, 2014). An IPython notebook, allowing figure reproduction, was deposited as a supplementry file.
3. Stress-induced mutagenesis and complex adaptation, by Yoav Ram and Lilach Hadany (Proceedings B, 2014). An IPython notebook, allowing figures reproduction, was deposited as a supplementry file.
4. Automatic segmentation of odor maps in the mouse olfactory bulb using regularized non-negative matrix factorization, by J. Soelter et al. (Neuroimage 2014, Open Access). The notebook allows to reproduce most figures from the paper and provides a deeper look at the data. The full code repository is also available.
5. Multi-tiered genomic analysis of head and neck cancer ties TP53 mutation to 3p loss, by A. Gross et al. (Nature Genetics 2014). The full collection of notebooks to replicate the results.
6. powerlaw: a Python package for analysis of heavy-tailed distributions, by J. Alstott et al.. Notebook of examples in manuscript, ArXiv link and project repository.
7. Collaborative cloud-enabled tools allow rapid, reproducible biological insights, by B. Ragan-Kelley et al.. The main notebook, the full collection of related notebooks and the companion site with the Amazon AMI information for reproducing the full paper.
8. A Reference-Free Algorithm for Computational Normalization of Shotgun Sequencing Data, by C.T. Brown et al.. Full notebook, ArXiv link and project repository.
9. The kinematics of the Local Group in a cosmological context by J.E. Forero-Romero et al.. The Full notebook and also all the data in a github repo.

Over 500K Notebooks on GitHub

CHECK OUT “TRENDING REPOSITORIES”





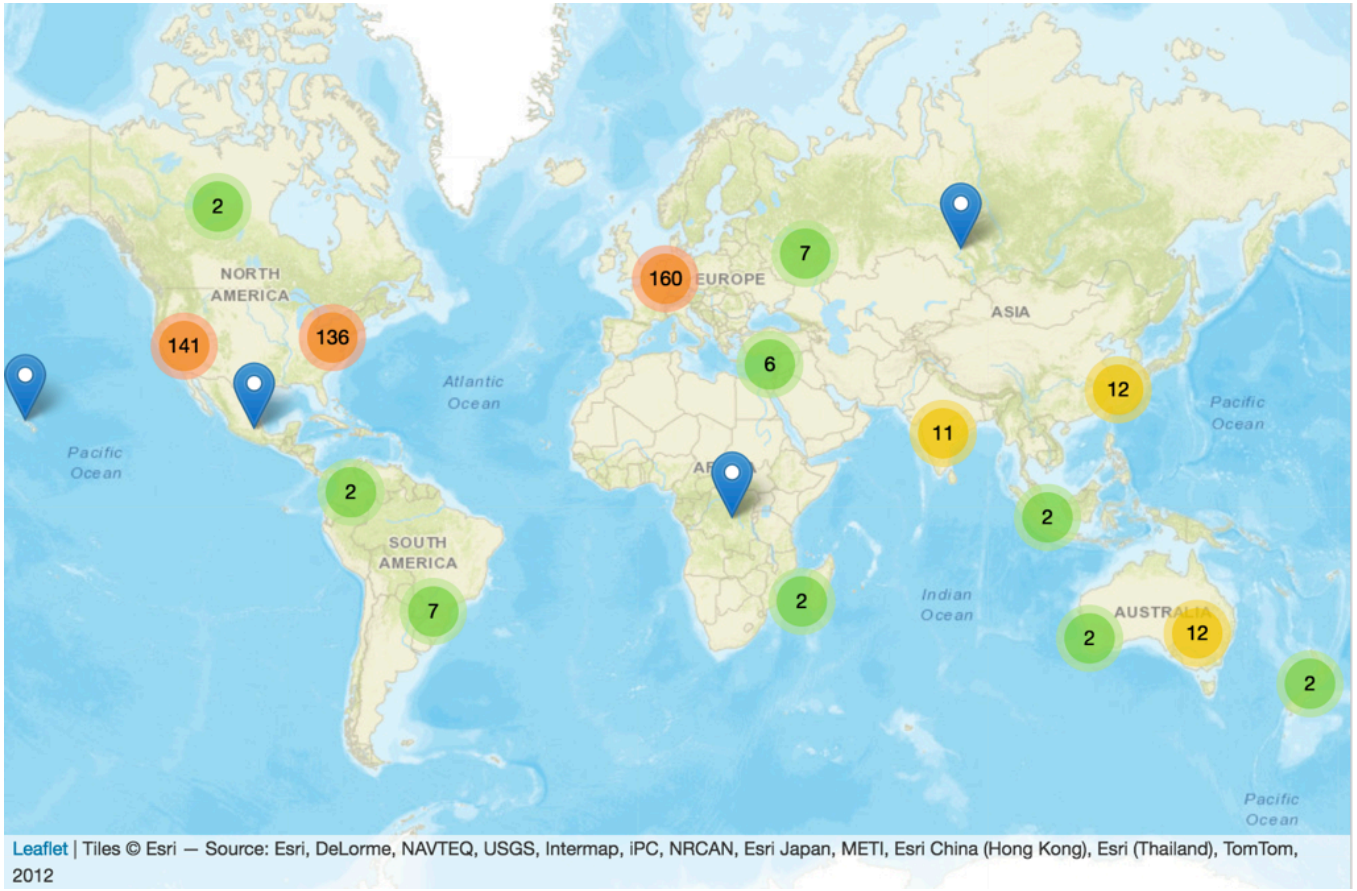
https://github.com/trending/jupyter-notebook?since=weekly

Jupyter in the Wild

- Open source - which means anyone can download it and use it for free.
- Digital Journalism, Scientific research, Commercial products, Classrooms, Coding Bootcamps, Online Learning, Conferences, Data Engineering Pipelines, High Performance Computing (HPC) . . .
- Physics, Astronomy, Biology, Economics & Finance, Social Sciences, Geo Sciences, Digital Humanities . . .
- Commercial big data platforms: Microsoft, IBM, Google, Bloomberg, more . . .

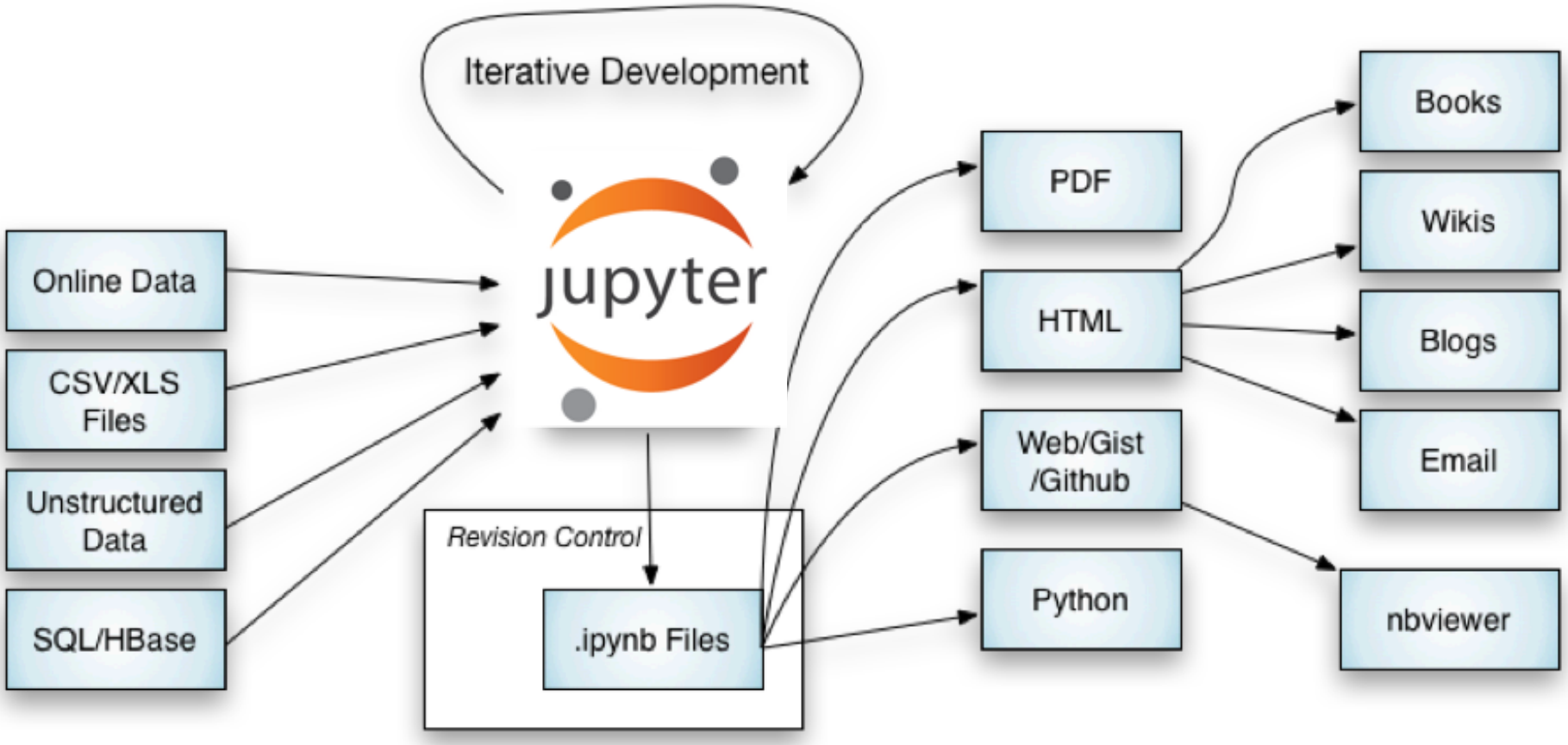
The Jupyter Team

~500 Committers to IPython/Jupyter Repositories



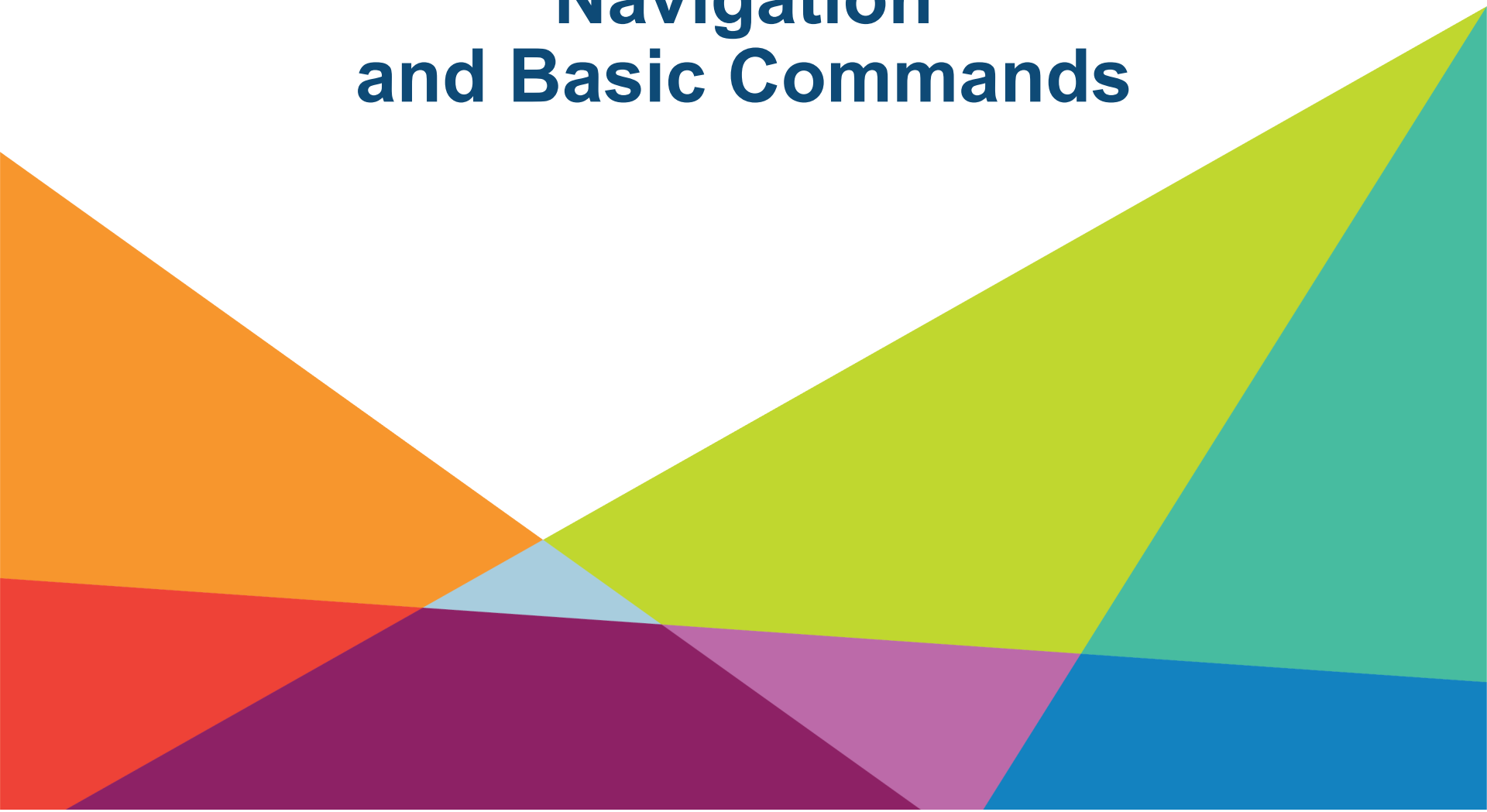
By Stuart Geiger, Jamie Whitacre, Matthias Bussonnier, and Nami Saghaei

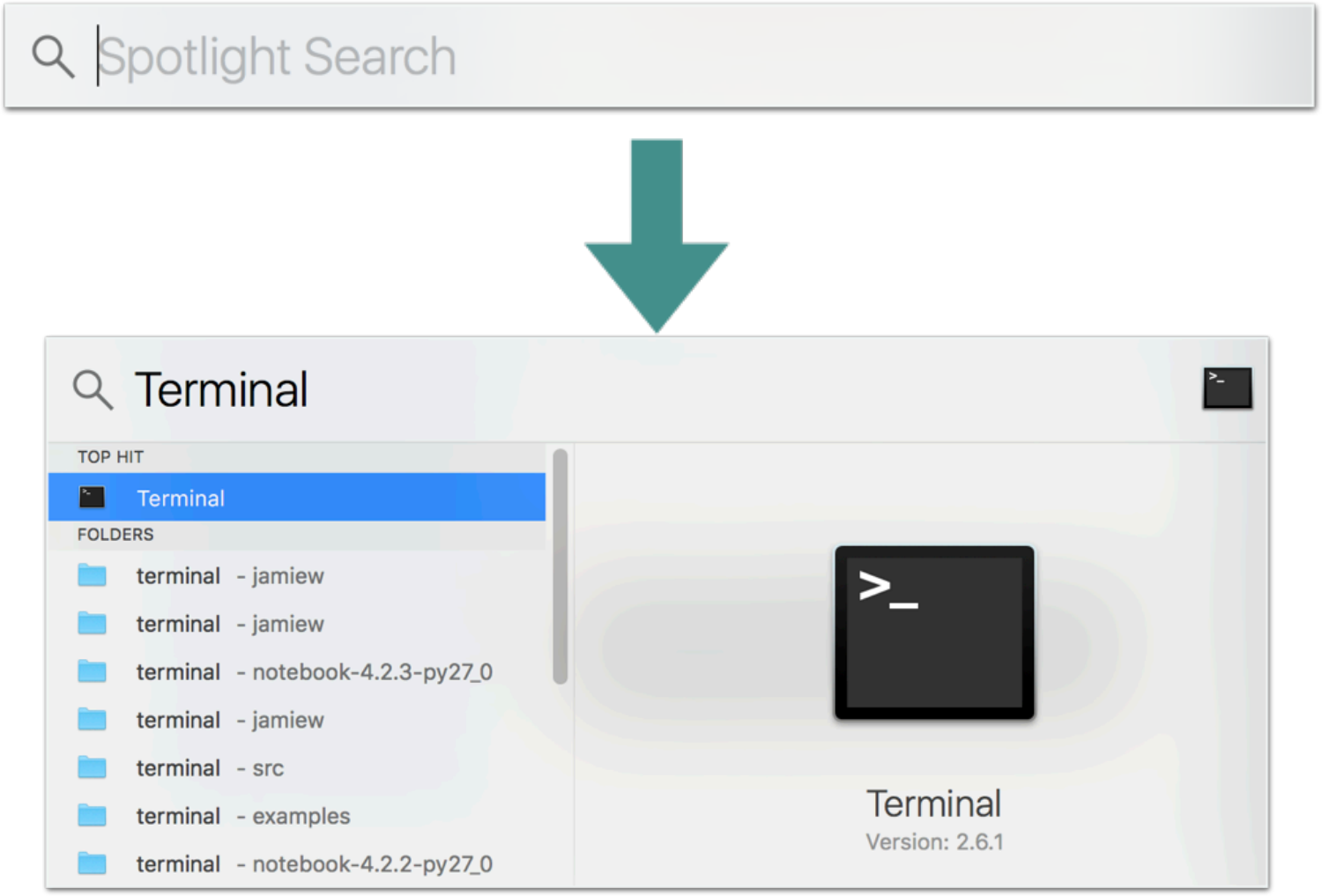
Notebook Workflows: The Big Picture

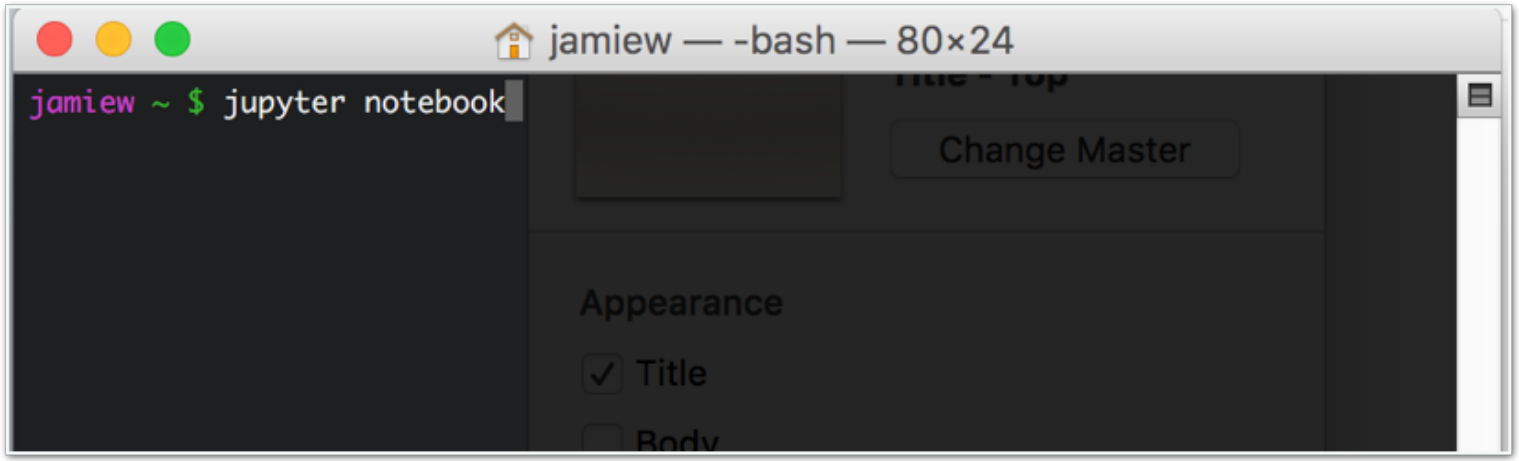


Let's Get Started

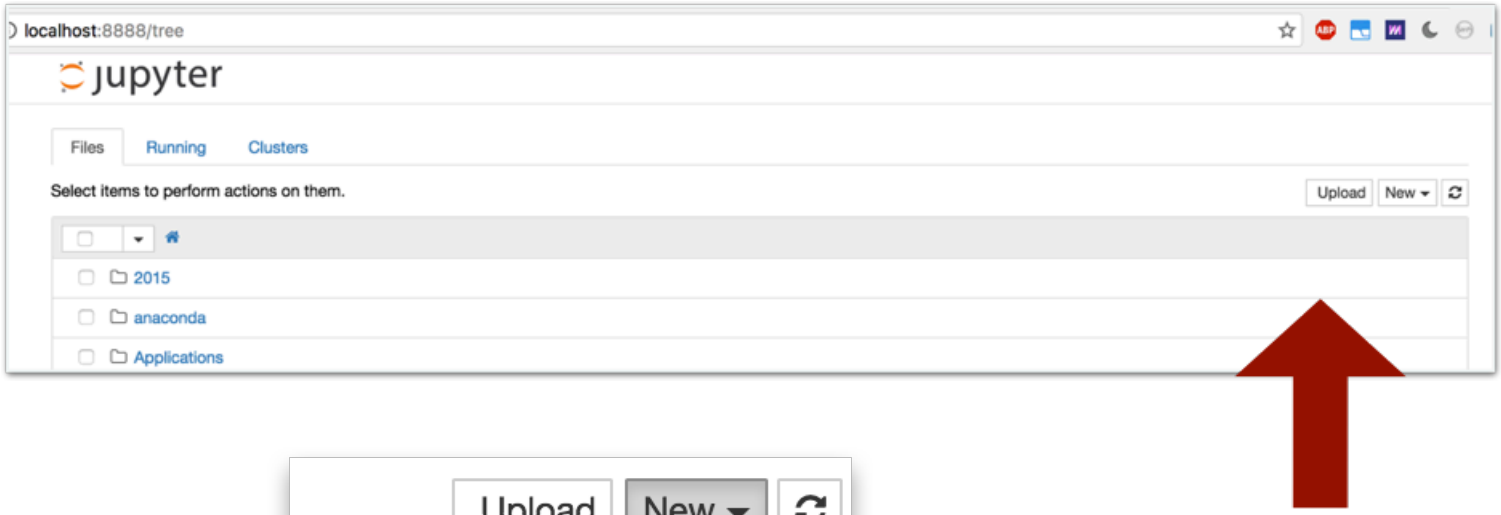
Navigation and Basic Commands

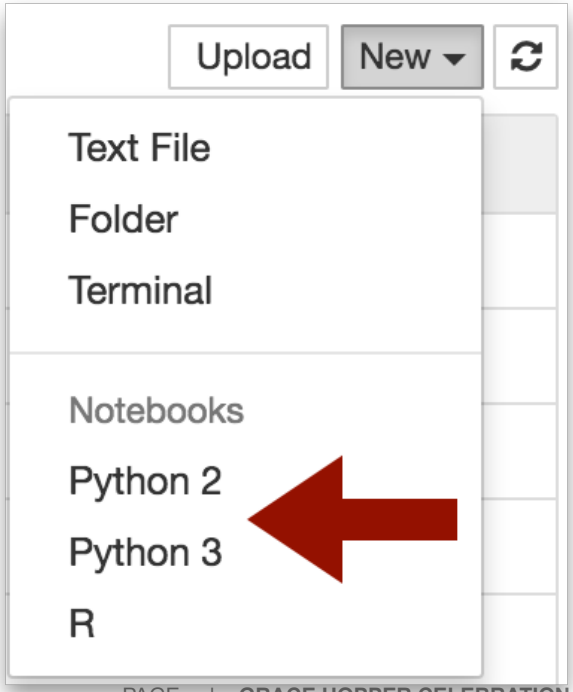


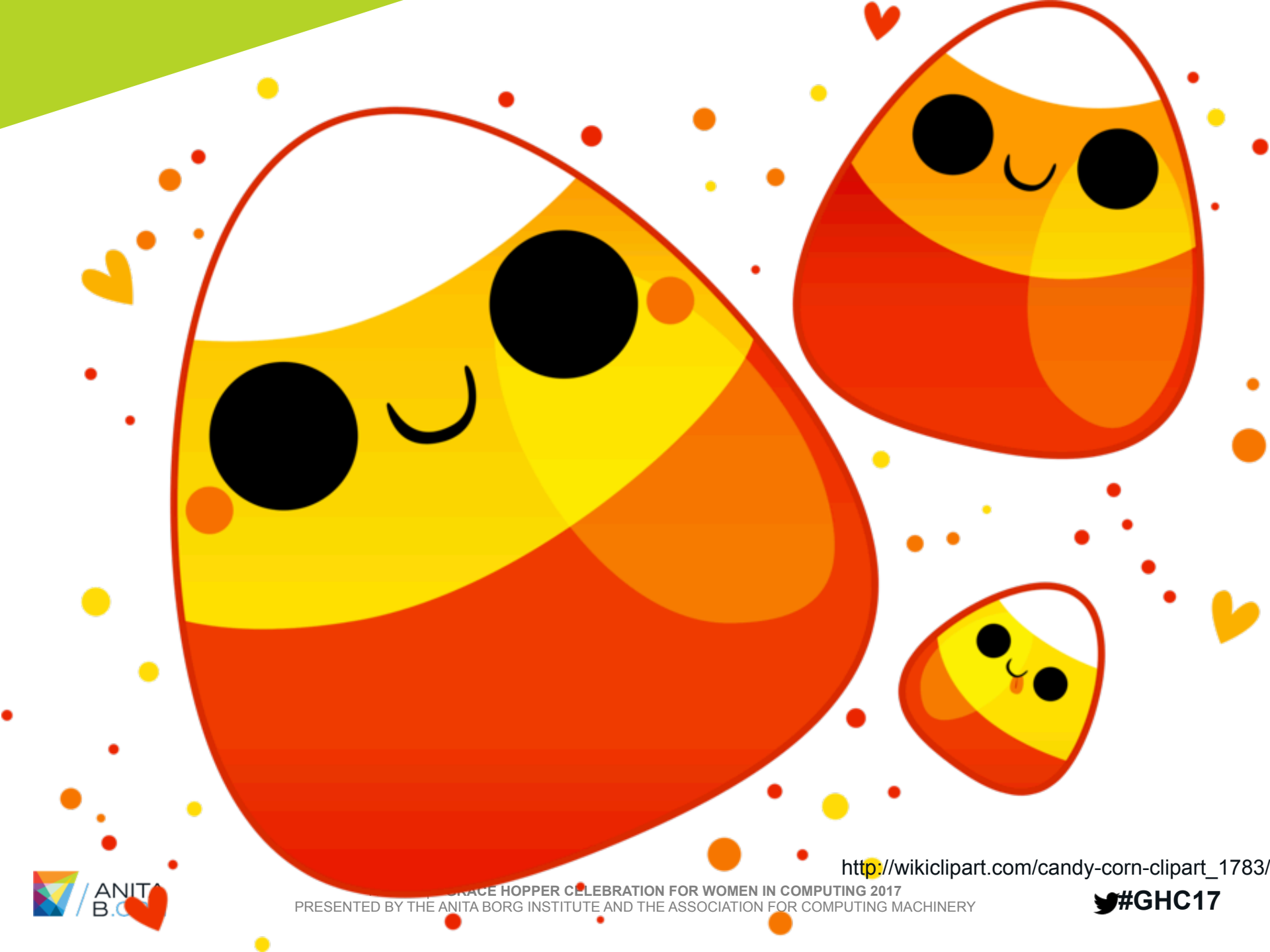

Spotlight Search
Terminal
TOP HIT
Terminal
FOLDERS
terminal - jamiew
terminal - jamiew
terminal - notebook-4.2.3-py27_0
terminal - jamiew
terminal - src
terminal - examples
terminal - notebook-4.2.2-py27_0
>_
Terminal
Version: 2.6.1


jamiew — -bash — 80×24
jamiew ~ $ jupyter notebook
Change Master
Appearance
Title

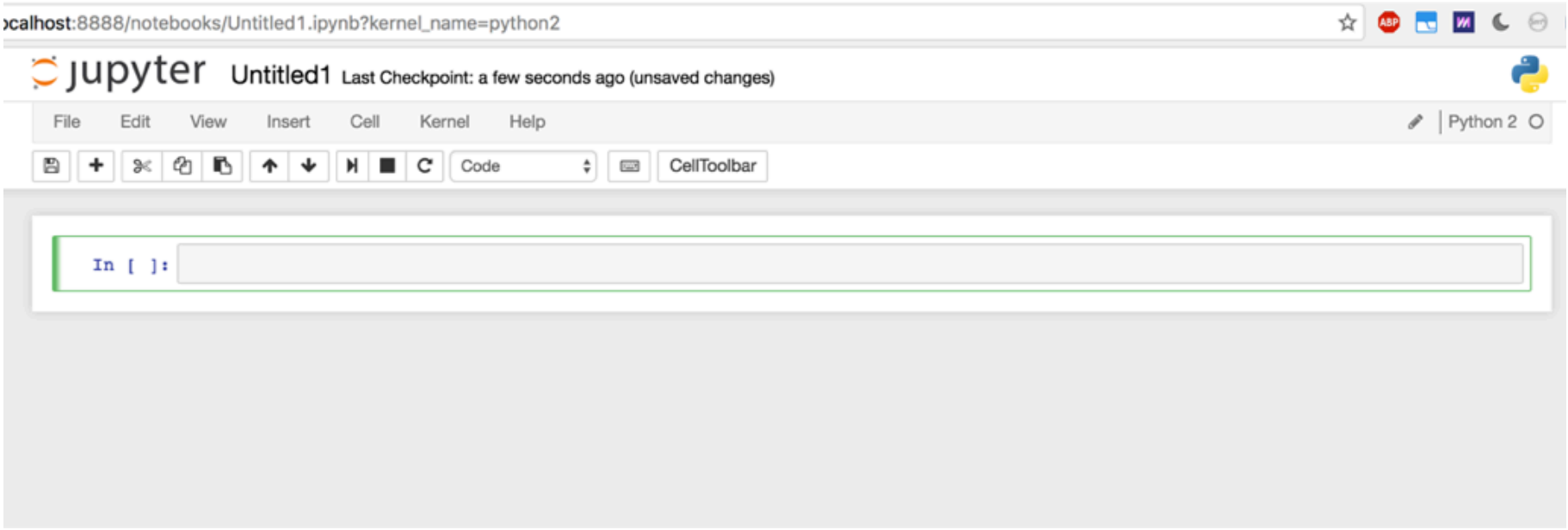

localhost:8888/tree
jupyter
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
2015
anaconda
Applications
Biorepository Legacy Migration
bokeh-notebooks-master
book-exercises
Coursera-HowToUseGitandGitHub
datasciencecoursera
Desktop
development
Documents

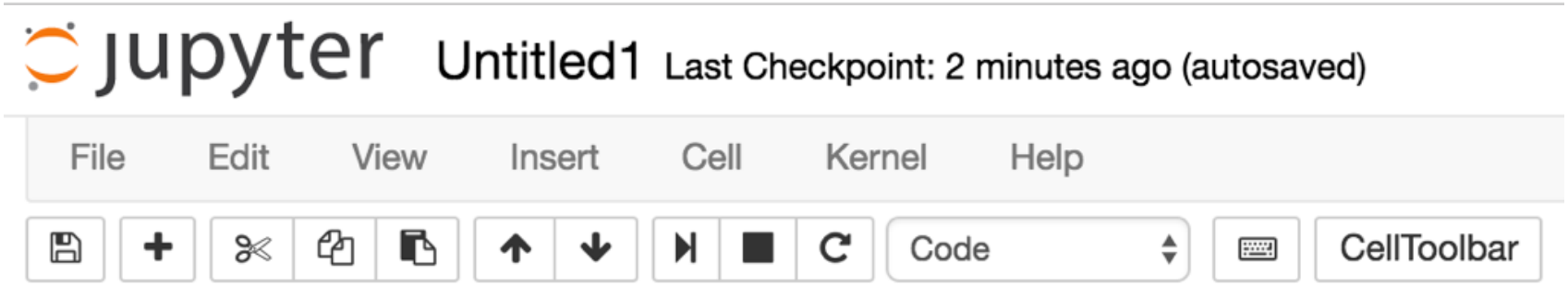

localhost:8888/tree
jupyter
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
2015
anaconda
Applications


Upload
New
Text File
Folder
Terminal
Notebooks
Python 2
Python 3
R



http://wikiclipart.com/candy-corn-clipart_1783/


calhost:8888/notebooks/Untitled1.ipynb?kernel_name=python2
jupyter
Untitled1
Last Checkpoint: a few seconds ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
CellToolbar
In [ ]:


jupyter
Untitled1
Last Checkpoint: 2 minutes ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Help
Code
CellToolbar

Warm Up

```
#Define a variable
x = 5
print(x)
```

Importing Libraries

Importing Libraries

import ____ as _____

CONVENTIONS

import matplotlib.pyplot as plt

%matplotlib inline

import pandas as pd

import seaborn as sns

from sklearn import _____

Pandas

Reading in Data

- Reading data in / querying data sources / writing data out

```
import pandas as pd

PATH = '/Users/jamiew/Desktop/GraceHopper2017/GHC 2017 A - Yelp Jupyter Workshop/dataset/'
biz_df = pd.read_csv(PATH + 'business.csv')
user_df = pd.read_csv(PATH + 'user.csv')
review_df = pd.read_csv(PATH + 'review.csv')

%time
```

Did it work?

Did it work?

```
review_df.head()
```

Did it work?

```
review_df.head()
```

	funny	user_id	text	business_id	stars	useful	type	cool	datetime	year
review_id										
NxL8SIC5yqOdnlXCg18IBg	0	KpkOkG6RIf4Ra25Lhhxf1A	If you enjoy service by someone who is as comp...	2aFiy99vNLklCx3T_tGS9A	5	0	review	0	2011-10-10	2011
pXbbIgOXvLuTi_SPs1hQEQ	0	bQ7fQq1otn9hKX-gXRsrgA	After being on the phone with Verizon Wireless...	2aFiy99vNLklCx3T_tGS9A	5	1	review	0	2010-12-29	2010
wslW2Lu4NYylb1jEapAGsw	0	r1NUhdNmL6yU9Bn-Yx6FTw	Great service! Corey is very service oriented....	2aFiy99vNLklCx3T_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
25RlYGq2s5qShi-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()
```

Pandas

http://pandas.pydata.org/pandas-docs/stable/10min.html

github.com/jehuston/pandas_tutorial/blob/master/pandas_tutorial.ipynb

https://github.com/brandon-rhodes/pycon-pandas-tutorial

Making Charts

Making Charts with Matplotlib

http://matplotlib.org/users/screenshots.html#simple-plot

- Simple plot
- Histograms
- Bar Charts
- Scatter plots

matplotlib

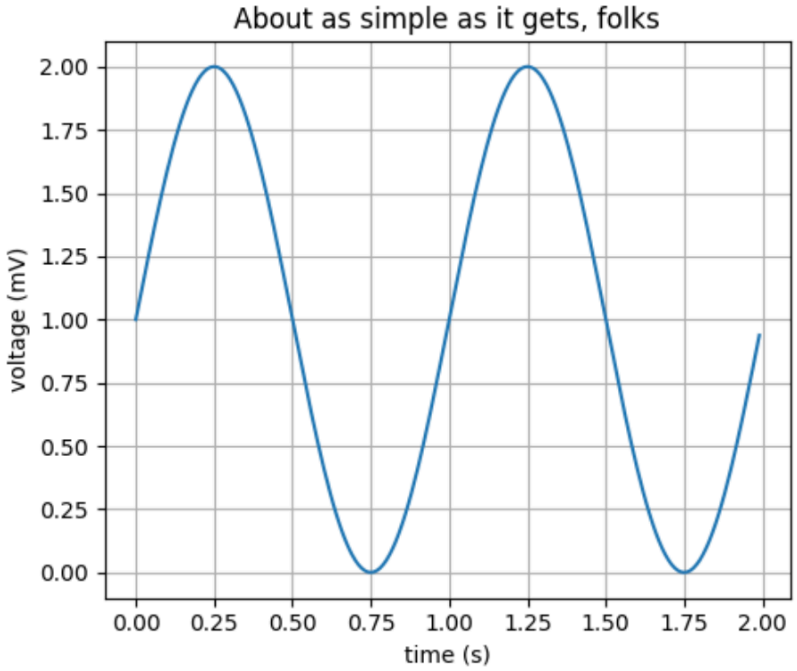
Screenshots

Here you'll find a host of example plots with the code that generated them.

Simple Plot

Here's a very basic `plot()` with text labels:

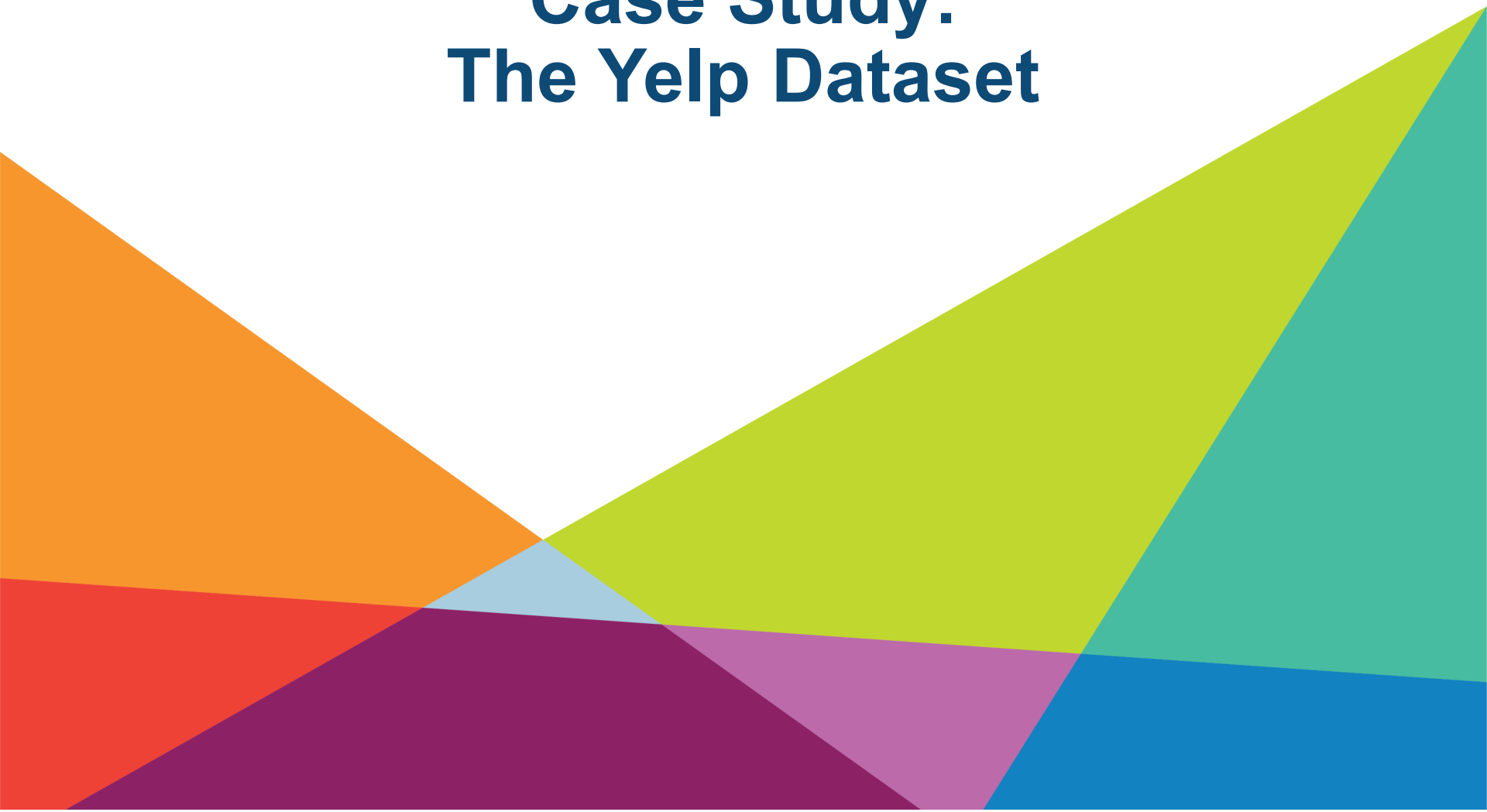
(Source code, png, pdf)



Making Charts with Matplotlib

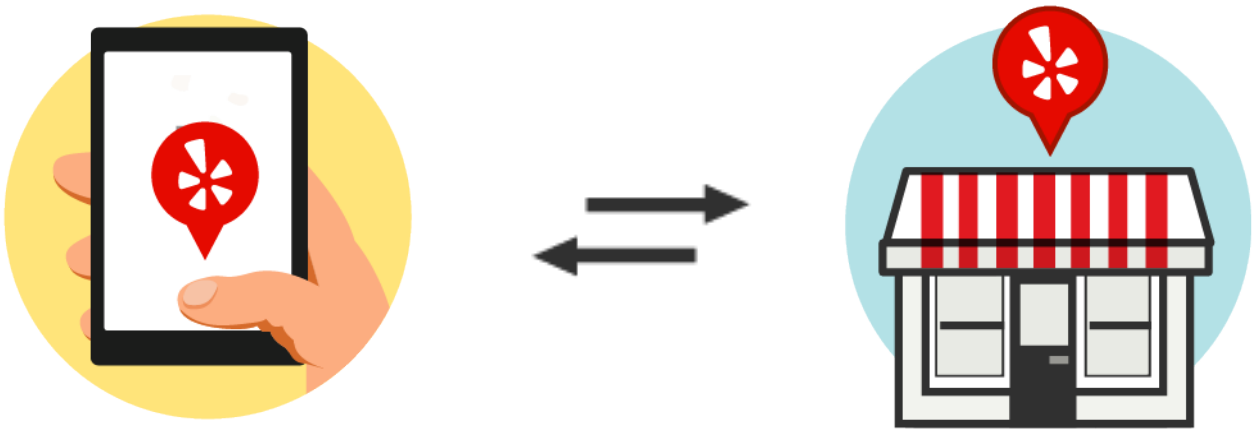
- Other great plotting libraries: plot.ly, Seaborn, Altair, Bokeh
- Remember to use '%matplotlib inline' to display your plot in your notebook

Case Study:
The Yelp Dataset



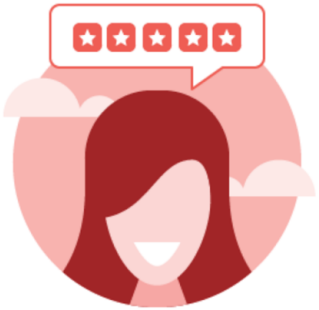
Yelp's Mission

Connecting people with great local businesses.



Yelp Open Dataset

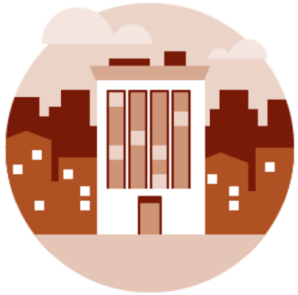
yelp.com/dataset



4,700,000 reviews

156,000 businesses

200,000 pictures



12 metropolitan areas

1,000,000 tips by 1,100,000 users
Over 1.2 million business attributes like hours, parking, availability, and ambience
Aggregated check-ins over time for each of the 156,000 businesses

Given users' past reviews on Yelp

yelp Open Dataset

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

Will it be a  review? (True / False)

To Solve a Data Science Problem

Step 1: Load the Data

Step 2: Explore and Visualize the Data

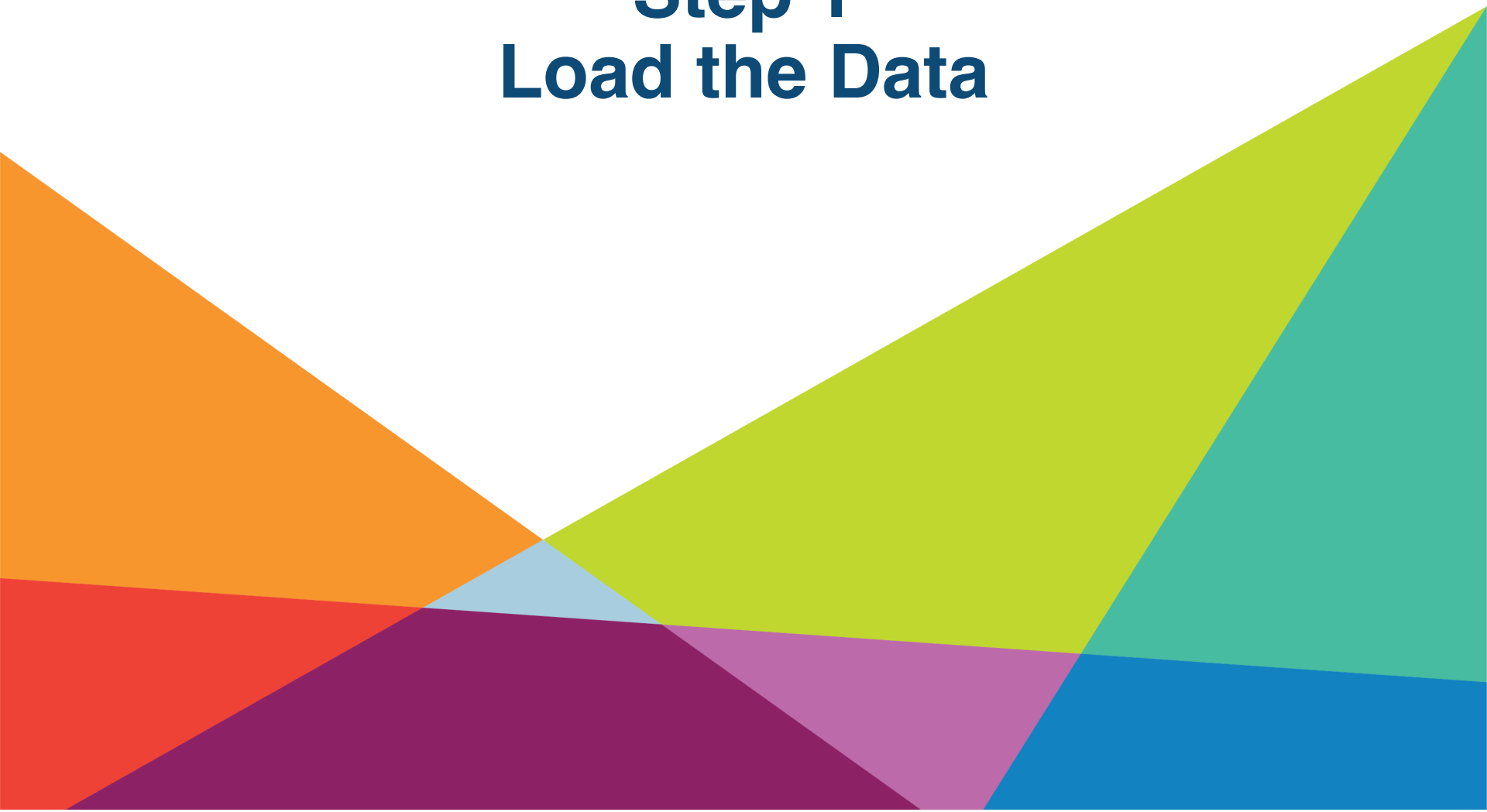
Step 3: Generate the Features

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

```
import pandas as pd

PATH = '/scratch/xun/docs/yelp_dataset_challenge_round10/'
biz_df = pd.read_csv(PATH + 'business.csv')
user_df = pd.read_csv(PATH + 'user.csv')
review_df = pd.read_csv(PATH + 'review.csv')
```

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

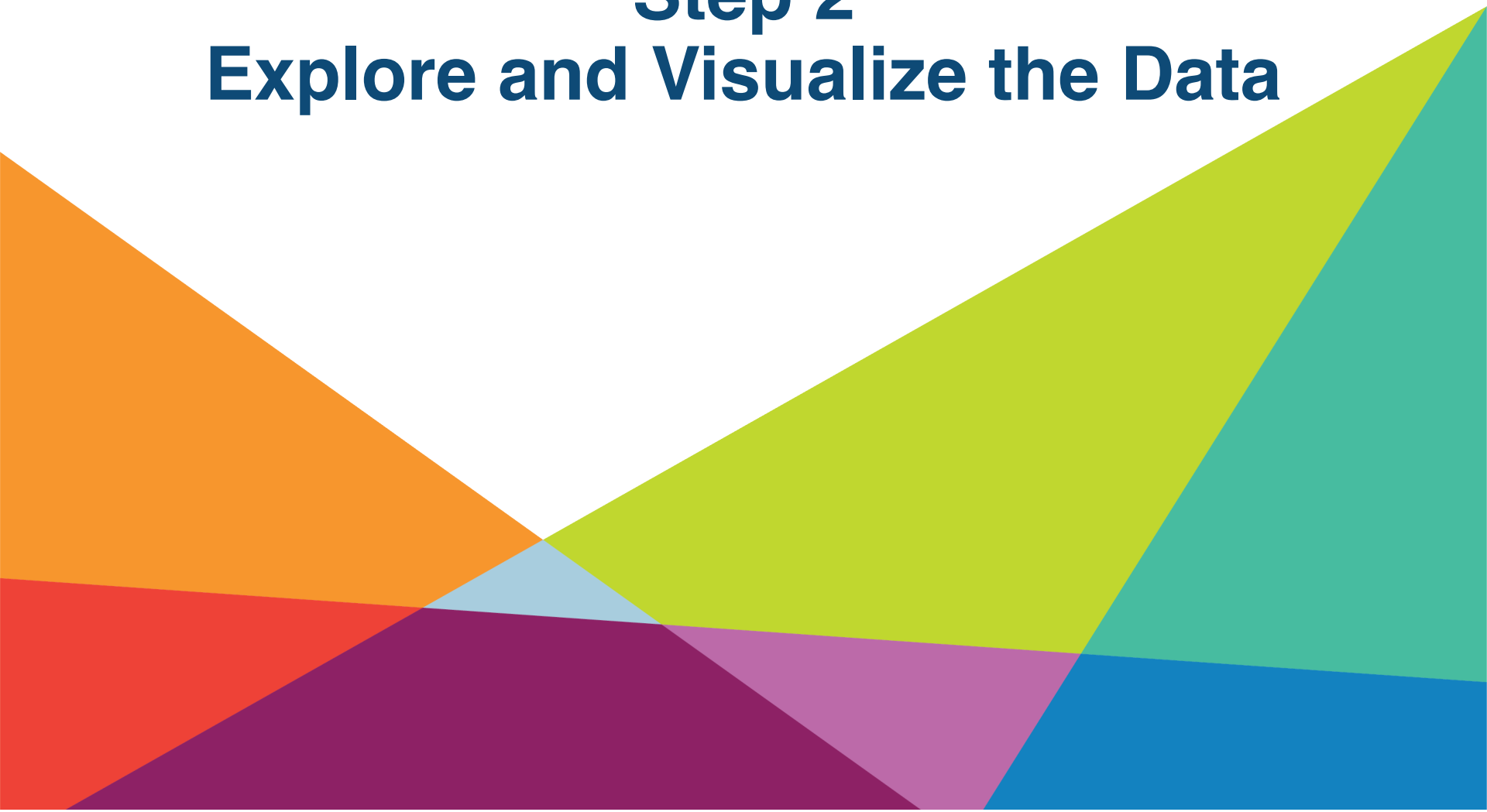
What's in a Review DataFrame?

```
review_df.head()
```

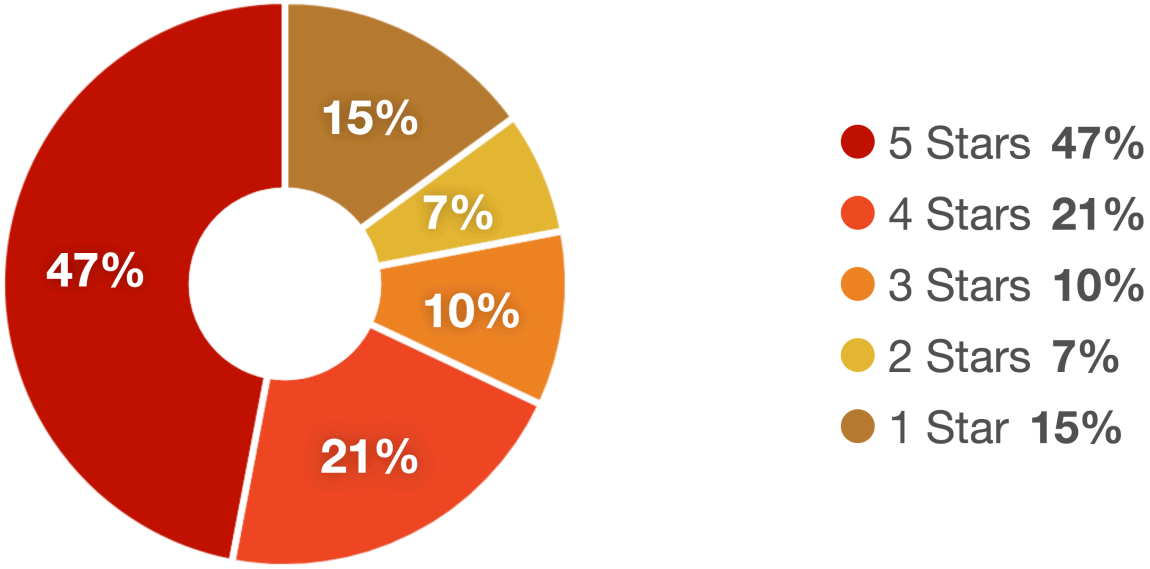
	funny	user_id	text	business_id	stars	useful	type	cool	datetime	year
review_id										
NxL8SIC5yqOdnlXCg18IBg	0	KpkOkG6RIf4Ra25Lhhxf1A	If you enjoy service by someone who is as comp...	2aFiy99vNLklCx3T_tGS9A	5	0	review	0	2011-10-10	2011
pXbbIgOXvLuTi_SPs1hQEQ	0	bQ7fQq1otn9hKX-gXRsrgA	After being on the phone with Verizon Wireless...	2aFiy99vNLklCx3T_tGS9A	5	1	review	0	2010-12-29	2010
wslW2Lu4NYylb1jEapAGsw	0	r1NUhdNmL6yU9Bn-Yx6FTw	Great service! Corey is very service oriented....	2aFiy99vNLklCx3T_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
25RlYGq2s5qShi-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

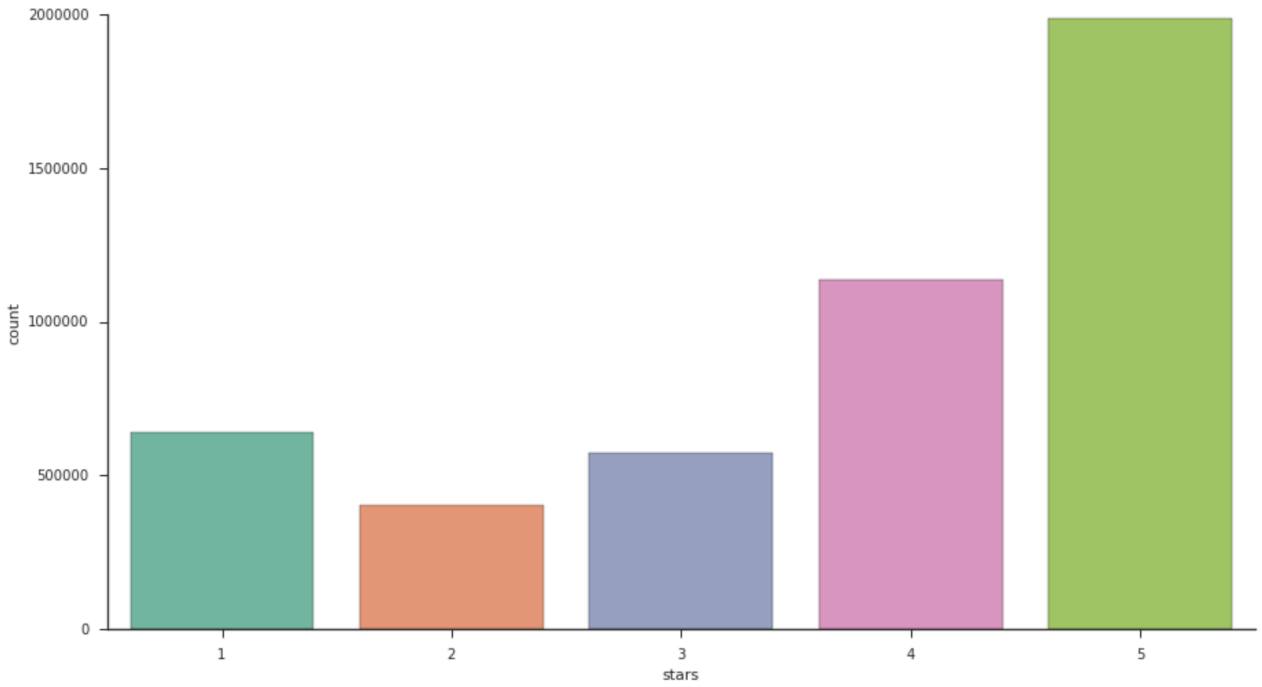


Source: https://www.yelp.com/factsheet

Plot Review Star Rating Distribution from Open Dataset

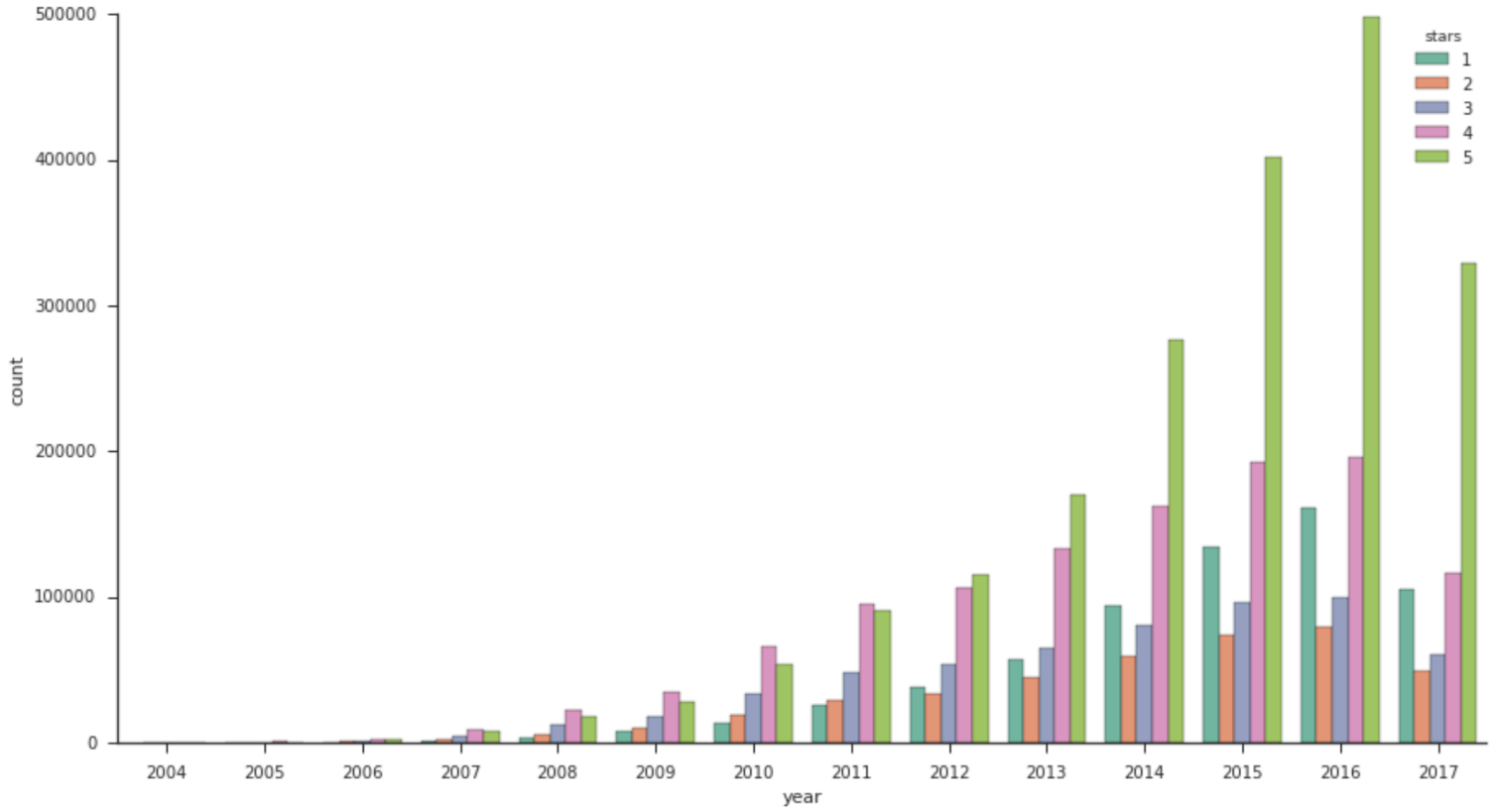
```
import seaborn as sns
%matplotlib inline
```

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

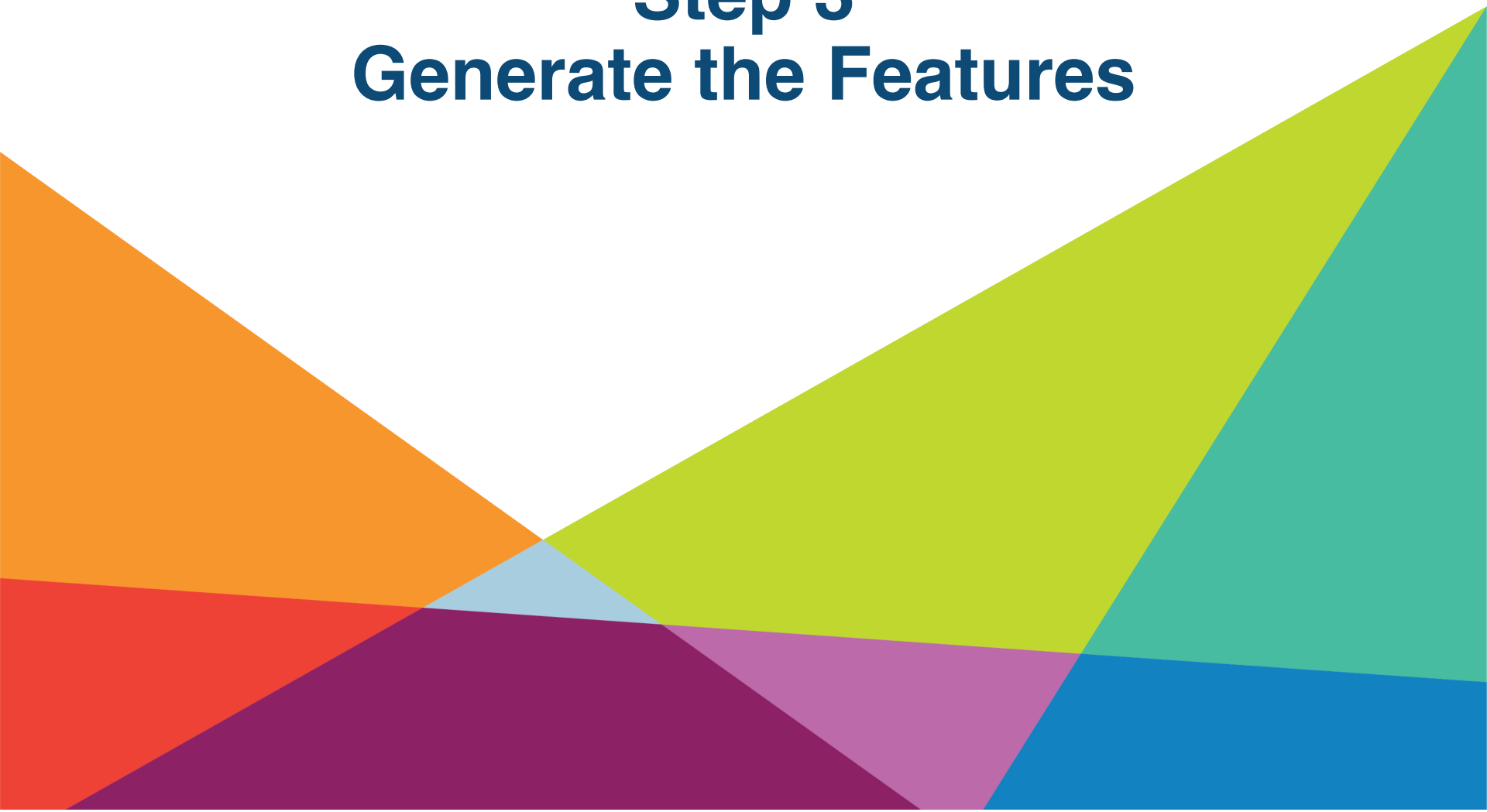


Plot Star Ratings by Year

```
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

```
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)
```

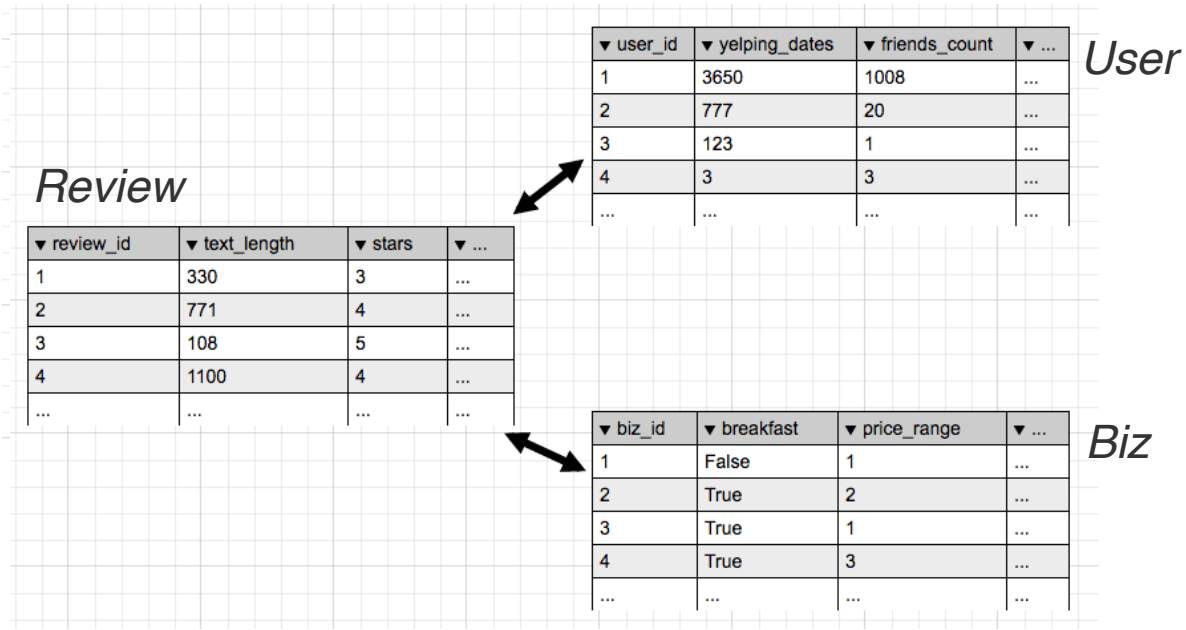
```
def to_boolean(df, columns):
    for column in columns:
        to_column = column+'_bool'
        df[to_column] = df[column].apply(lambda x: bool(x))
        df.drop(column, axis=1, inplace=True)
```

```
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 DataFrames to Populate the Features



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

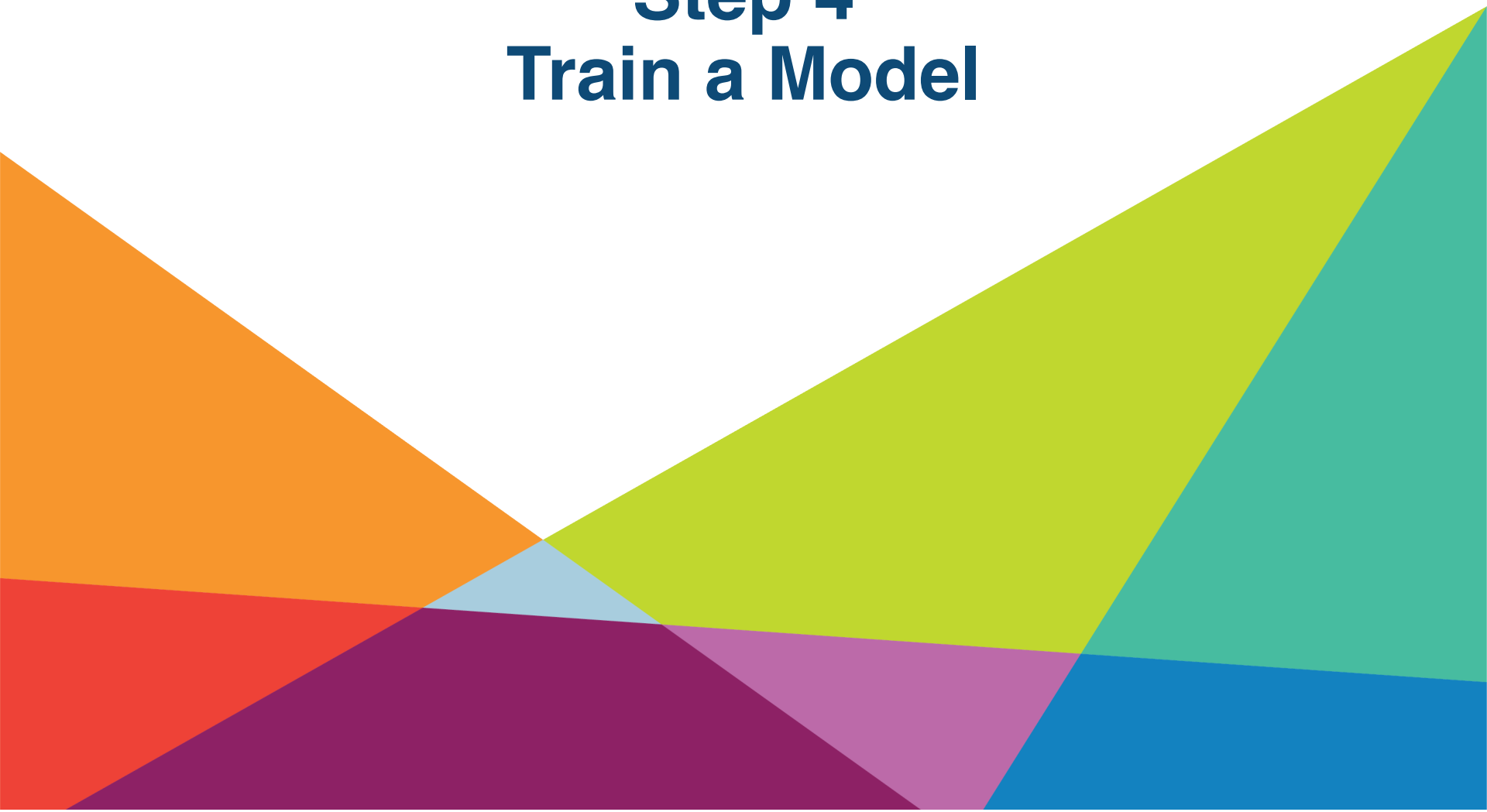
user_id	yelping_dates	friends_count	...
1	3650	1008	...
2	777	20	...
3	123	1	...
4	3	3	...
...	...	...	...

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

```
# 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')
```

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

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.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('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,)
```

Model: Logistic Regression (LR)

Estimates the probability of a **binary** response based on the features

Here we estimate the probability of a review being five-star

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)
```

Cross Validation

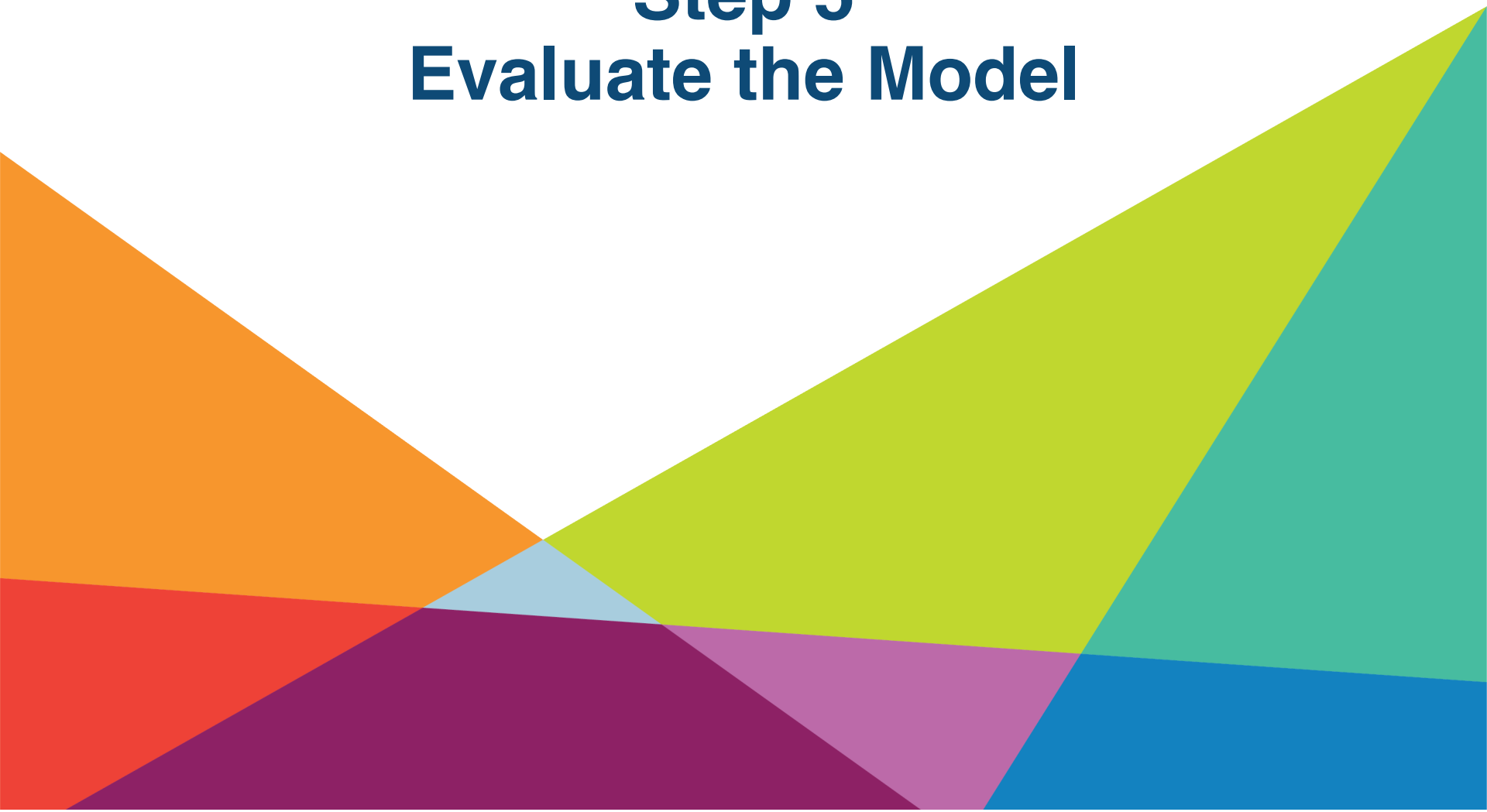
Holding out a portion of the training data for model validation, and do this for `n_folds`

- Ensure that the model does not overfit the training data
- Select optimal model parameters

```
from sklearn.cross_validation import StratifiedKFold

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

Step 5
Evaluate the Model



Metrics

Accuracy:
Percentage of labels correctly predicted. The higher the better.

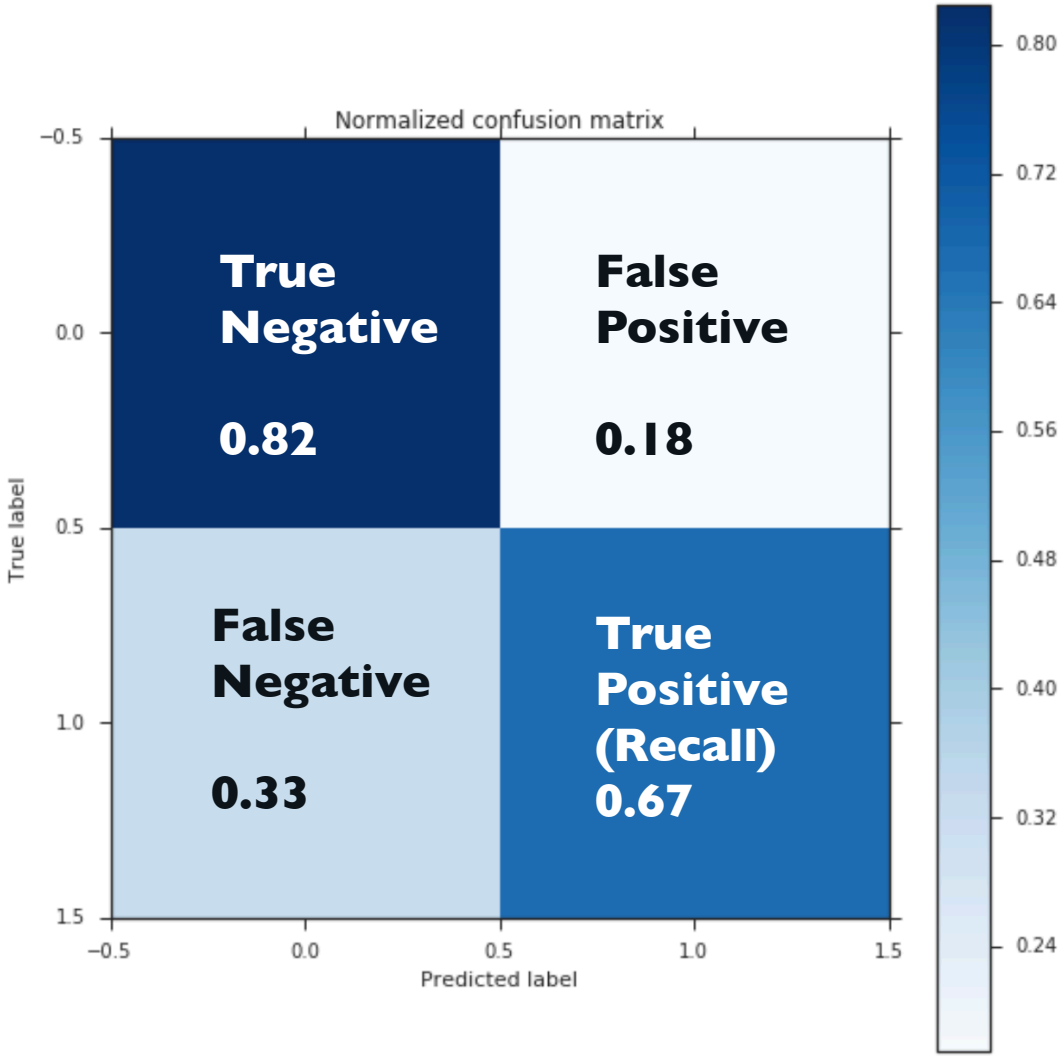
```
from sklearn.cross_validation import cross_val_score
import numpy as np

# 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), 3)
```

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

```
5-fold Train CV | Accuracy: 0.76 +/- 0.001 | ROC AUC: 0.836 +/- 0.001
```

Evaluation via Confusion Matrix



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?

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
```

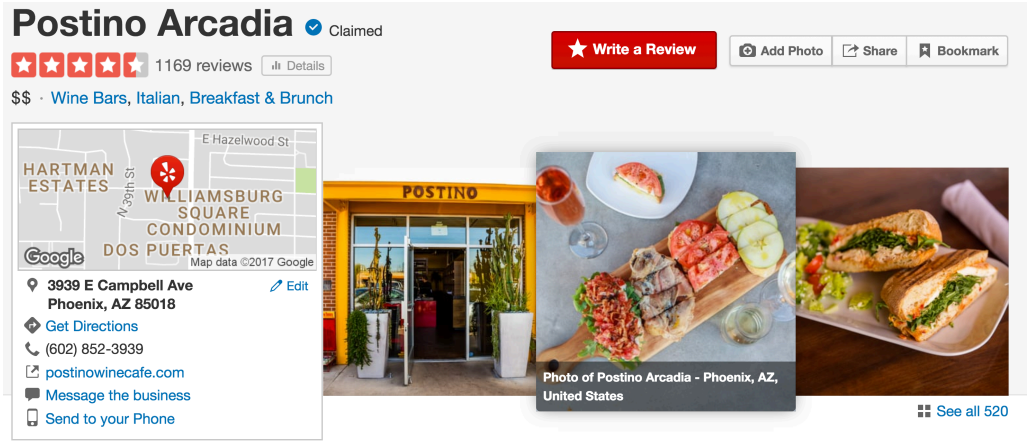
```
user_id
Hj20fg3vyzKnJwnLn_rMqw    4.55
Name: average_stars, dtype: float64
```

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

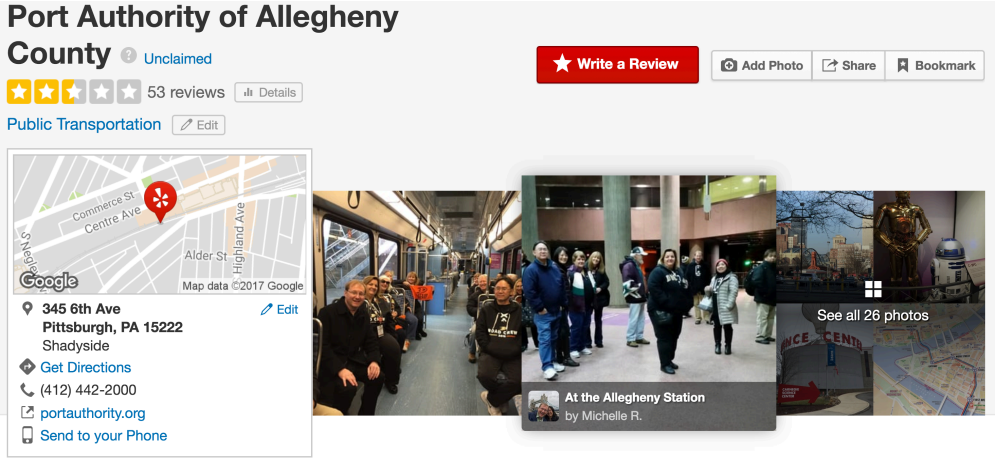
```
user_id
om5ZiponkpRqUNa3pVPiRg    3.94
Name: average_stars, dtype: float64
```

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

biz1



biz2



Will it be a ★★★★★ review?

Make predictions for user[1,2,3]'s review on biz1

```
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)
```

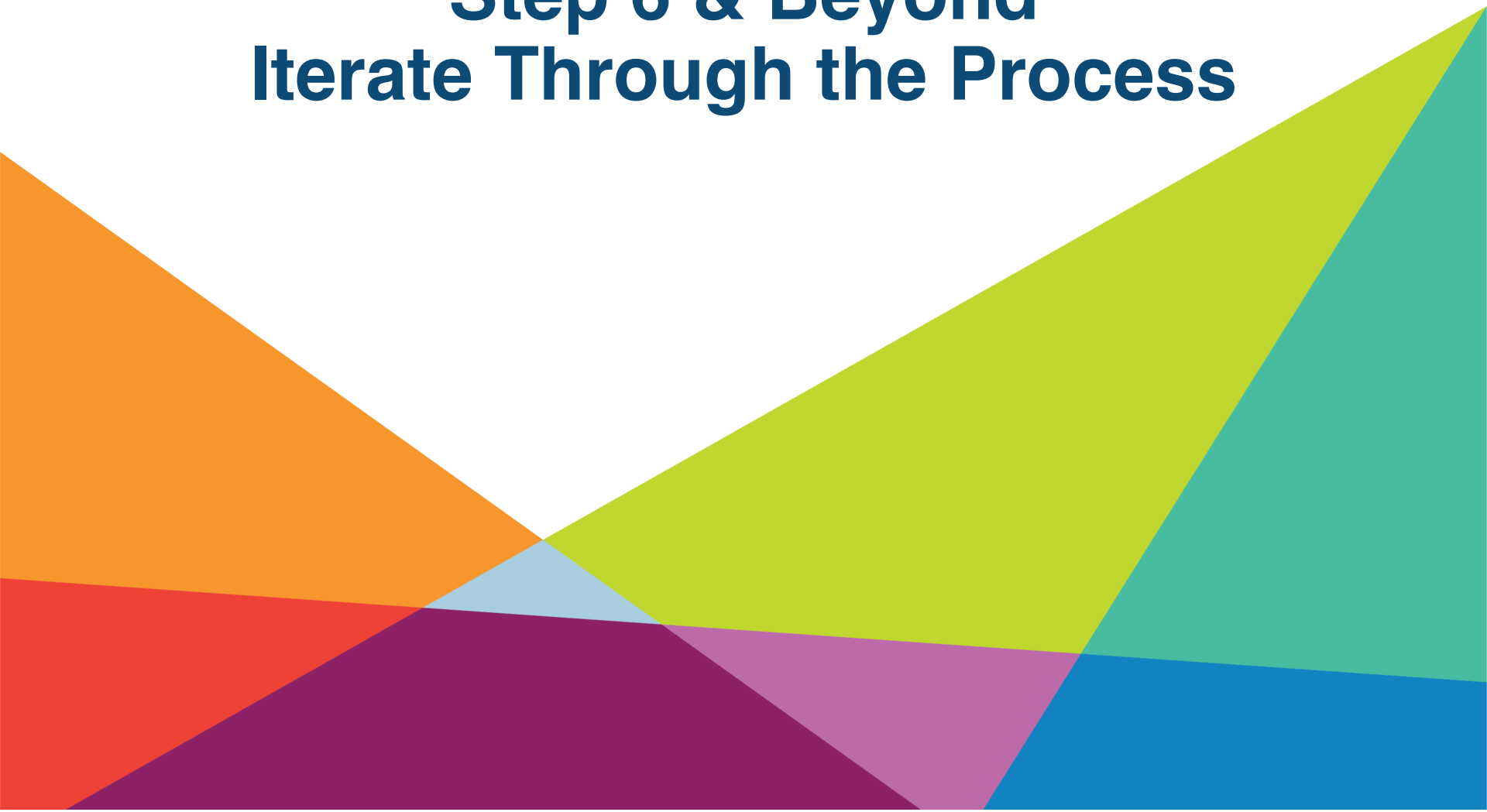
```
True , with probability [False, True] ==  [ 0.0871309  0.9128691]
True , with probability [False, True] ==  [ 0.21115779  0.78884221]
False , with probability [False, True] ==  [ 0.8328338  0.1671662]
```

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] ==  [ 0.35877821  0.64122179]
False , with probability [False, True] ==  [ 0.61076983  0.38923017]
False , with probability [False, True] ==  [ 0.9668934  0.0331066]
```

Step 6 & Beyond
Iterate Through the Process



Yelp Dataset Challenge Round 10

yelp.com/dataset/challenge

August 30, 2017 - December 31st, 2017

GRACE HOPPER CELEBRATION

ANITA B.ORG

Thank you

FEEDBACK? RATE AND REVIEW THE SESSION ON OUR MOBILE APP

Download the GHC 17 app at http://bit.ly/ghc17app or search GHC 2017 in the app store

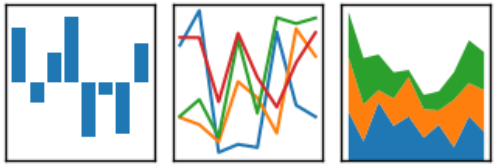
Association for Computing Machinery

Backup Slides

Tools

pandas

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



seaborn

matplotlib

scikit learn

jupyter

Will Your Next yelp Review Be a ★★★★★ Review?

GRACE HOPPER CELEBRATION

ANITA B.ORG

Notebook & Slides Available At

https://github.com/xun-tang/ghc_ds_workshop

Short URL: https://goo.gl/WgoeUL

Association for Computing Machinery

#GHC17

json -> csv

https://www.yelp.com/dataset/documentation/json

https://github.com/Yelp/dataset-examples

json_to_csv_converter:

Convert the dataset from json format to csv format.

Pandas DataFrame

Pandas is great for ETL and exploration of tabular data (tsv, csv, psv format) and dictionary (or json) data

In memory data warehouse

Seaborn: Statistical Data Visualization

Makes visualization a central part of exploring and understanding data

Operates on dataframes

Internally performs the necessary aggregation and statistical model-fitting to produce informative plots

Questions?

Repo: github.com/xun-tang/ghc_ds_workshop

Linkedin: linkedin.com/in/xuntang

Email: xun@yelp.com

Twitter: @whoisxun

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

GRACE HOPPER CELEBRATION

ANITA B.ORG

What Model to Use?

Association for Computing Machinery

#GHC17

For Example..

Decision Tree

predicts the value of a target variable by learning simple decision rules inferred from the data features

Random Forest

combines de-correlated trees, where each tree is built from a bootstrap sample and node splits are calculated from random feature subsets

Ensemble Model

combine predictions of several models in order to improve the accuracy (decrease bias) and robustness (decrease variance) over a single model

Build model

```
from sklearn import linear_model

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