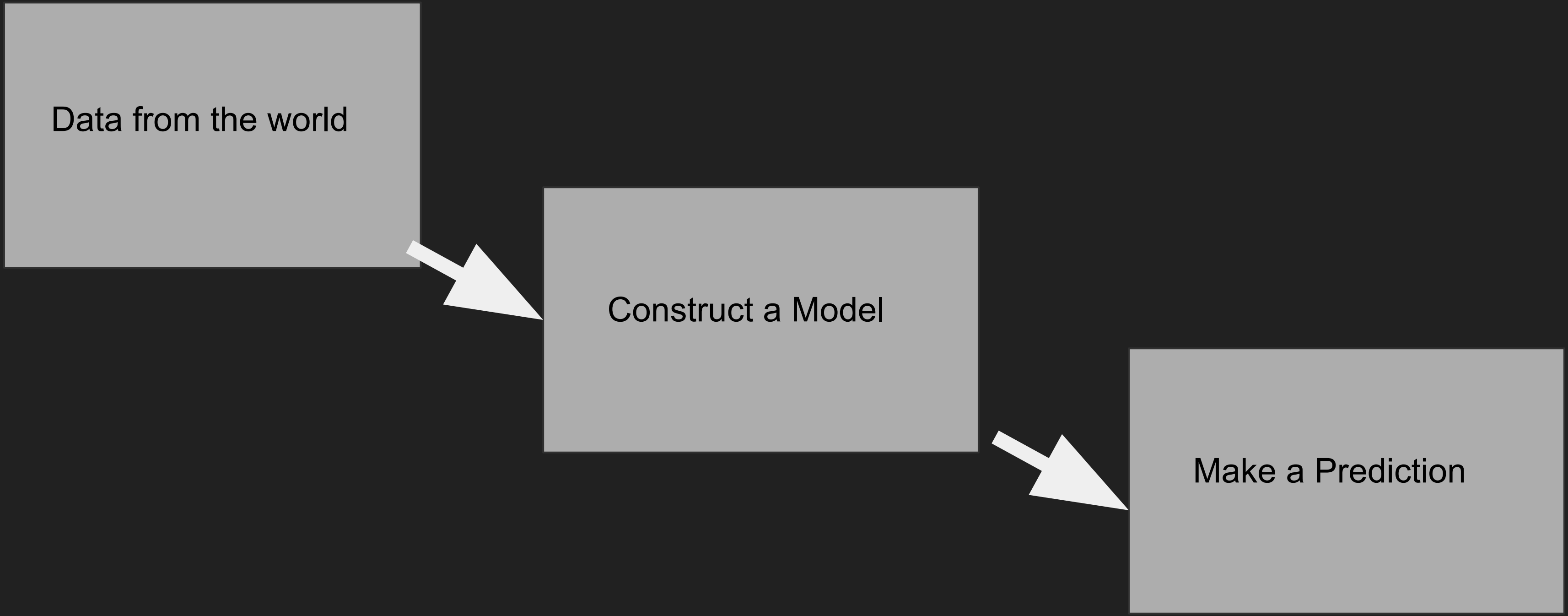
# Data Science and Machine Learning for Non-Data Scientists

Instructor: Camille Avestruz, PhD
EFI/KICP Postdoctoral Fellow @ University of Chicago
co-Instructor: Daniela Huppenkothen, PhD
Assoc Director DIRAC Institute @ U. Washington

Credit: Based on a lesson developed by CA and M. Lightman for a regional SACNAS conference

# What is data science?

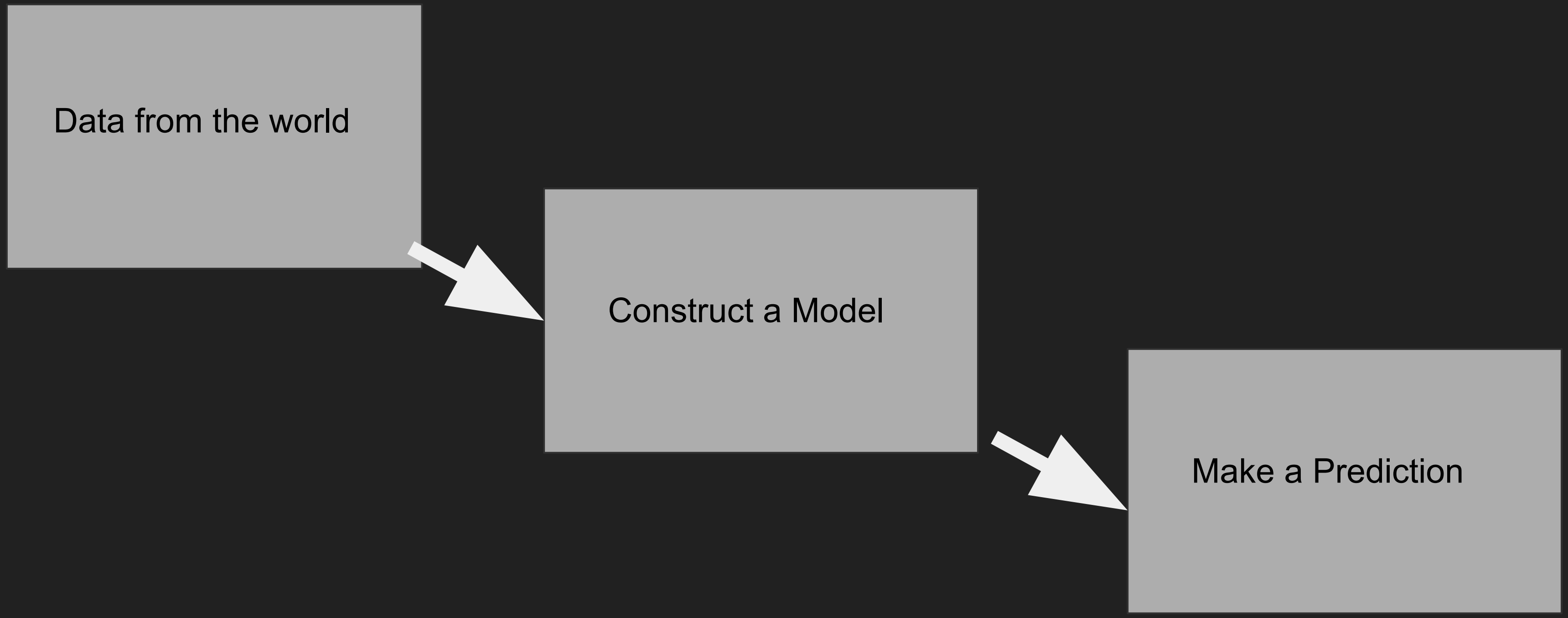


Data from the world

Construct a Model

Make a Prediction

# Who does data science?

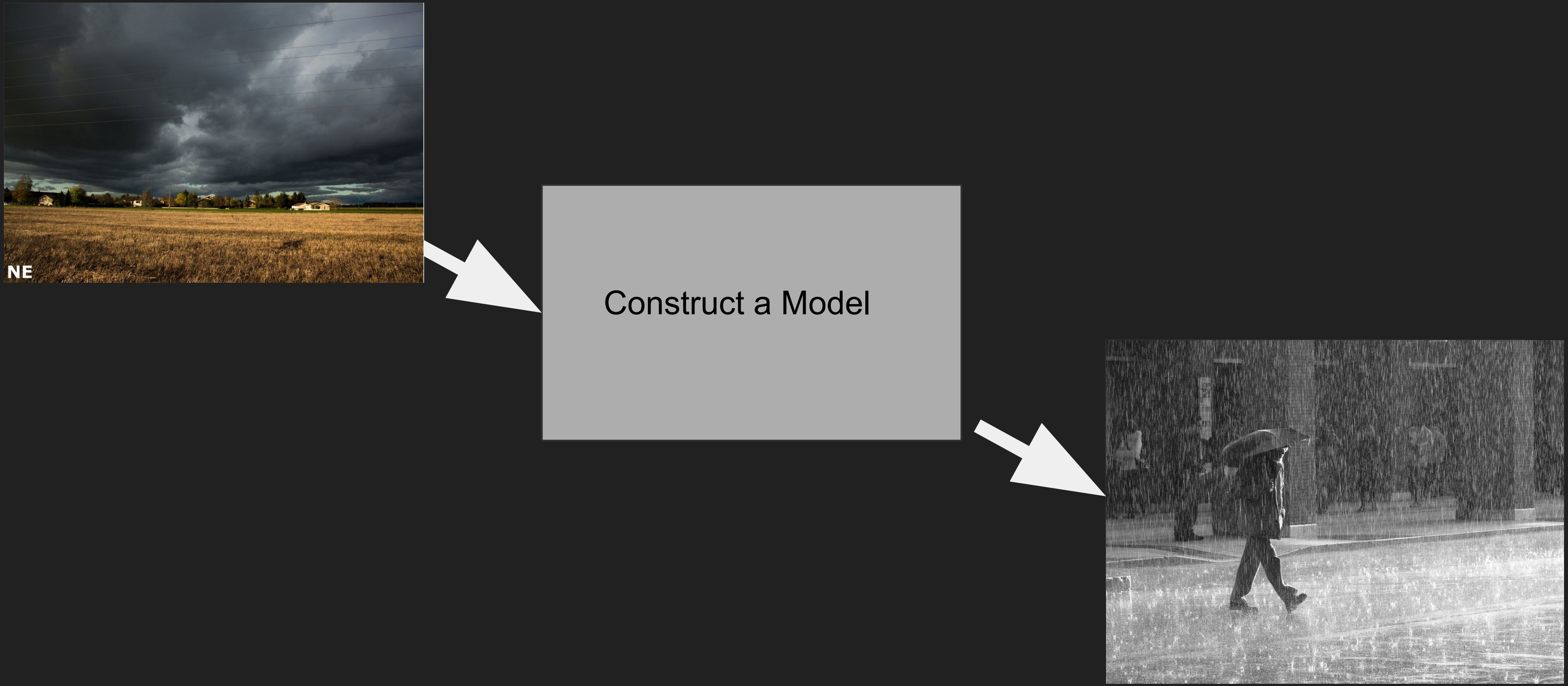


Data from the world

Construct a Model

Make a Prediction

# Who does data science?  Everyone!!!



Construct a Model

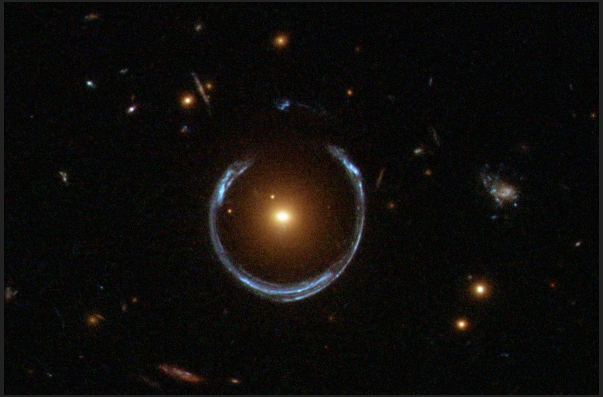
# Who does data science?

Camille Avestruz

KICP Fellow @ UChicago

Areas of research:
Astronomy, Cosmology

Data sets: Simulations "volumes of our universe", galaxies, clusters of galaxies, ...



5min

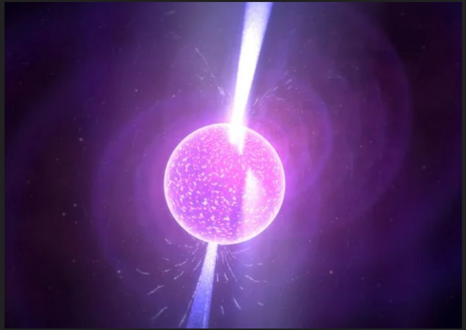
# Who does data science?

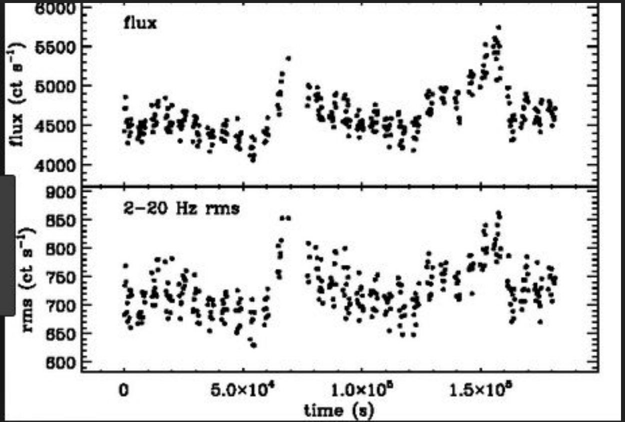
Daniela Huppenkothen

Assoc Director @ DIRAC Institute, U. Washington

Areas of work: Astrostatistics

Data sets: Time series of black holes, neutron stars





# Let’s do data science: (1) Look at Data

**Like**
Dislike

**Zootopia**
**Moonlight**
Deadpool
**Rogue One**

Zootopia
Moonlight
**Deadpool**
**Rogue One**

**Zootopia**
Moonlight
Deadpool
Rogue One

**Zootopia**
**Moonlight**
Deadpool
Rogue One

Zootopia
**Moonlight**
Deadpool
Rogue One

Zootopia
**Moonlight**
Deadpool
**Rogue One**

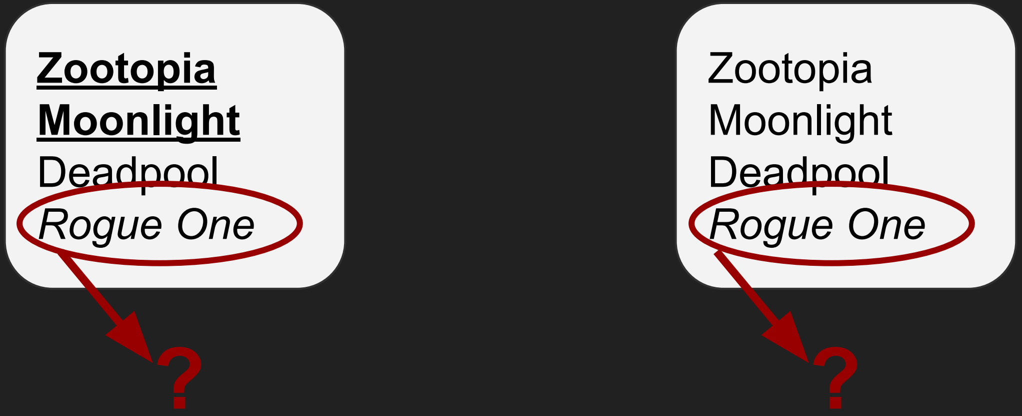
**Zootopia**
**Moonlight**
**Deadpool**
**Rogue One**

**Zootopia**
Moonlight
**Deadpool**
Rogue One

15min

# Let’s do data science: (2) Build a Model

Break into groups of 4 - discuss for 5 minutes: Does the person like Rogue One?



**Zootopia**
**Moonlight**
Deadpool
*Rogue One*

?

Zootopia
Moonlight
Deadpool
*Rogue One*

?

# Let’s do data science: (2) Build a Model

**Like**
Dislike

**Zootopia**
**Moonlight**
Deadpool
**Rogue One**

Zootopia
Moonlight
**Deadpool**
**Rogue One**

**Zootopia**
Moonlight
Deadpool
Rogue One

**Zootopia**
**Moonlight**
Deadpool
Rogue One

Zootopia
**Moonlight**
Deadpool
Rogue One

Zootopia
**Moonlight**
Deadpool
**Rogue One**

**Zootopia**
**Moonlight**
**Deadpool**
**Rogue One**

**Zootopia**
Moonlight
**Deadpool**
Rogue One

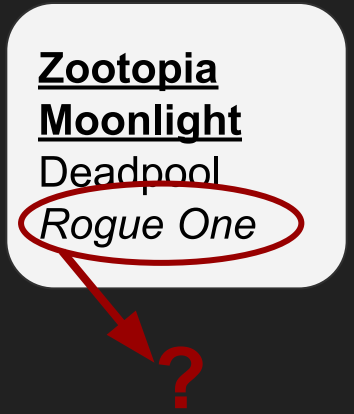
**Zootopia**
**Moonlight**
Deadpool
*Rogue One* ?

Zootopia
Moonlight
Deadpool
*Rogue One* ?

10min

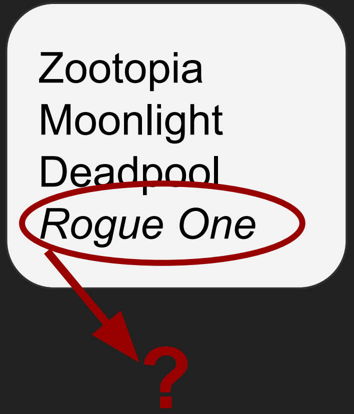
# Let’s do data science: (3) Make a Prediction

What ideas did you come up with?



**<u>Zootopia</u>**
**<u>Moonlight</u>**
Deadpool
*Rogue One*

?



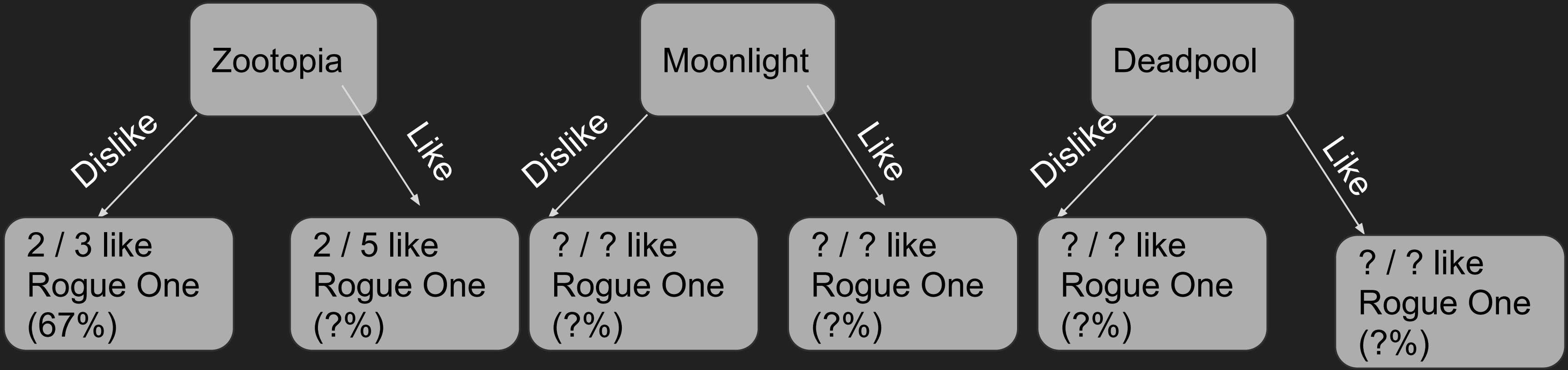
Zootopia
Moonlight
Deadpool
*Rogue One*

?

15min

# Let’s do data science: (2b) Construct a “tree” model

(Fill this out in your groups)



Zootopia

Dislike

Like

2 / 3 like Rogue One (67%)

2 / 5 like Rogue One (?%)

Moonlight

Dislike

Like

? / ? like Rogue One (?%)

? / ? like Rogue One (?%)

Deadpool

Dislike

Like

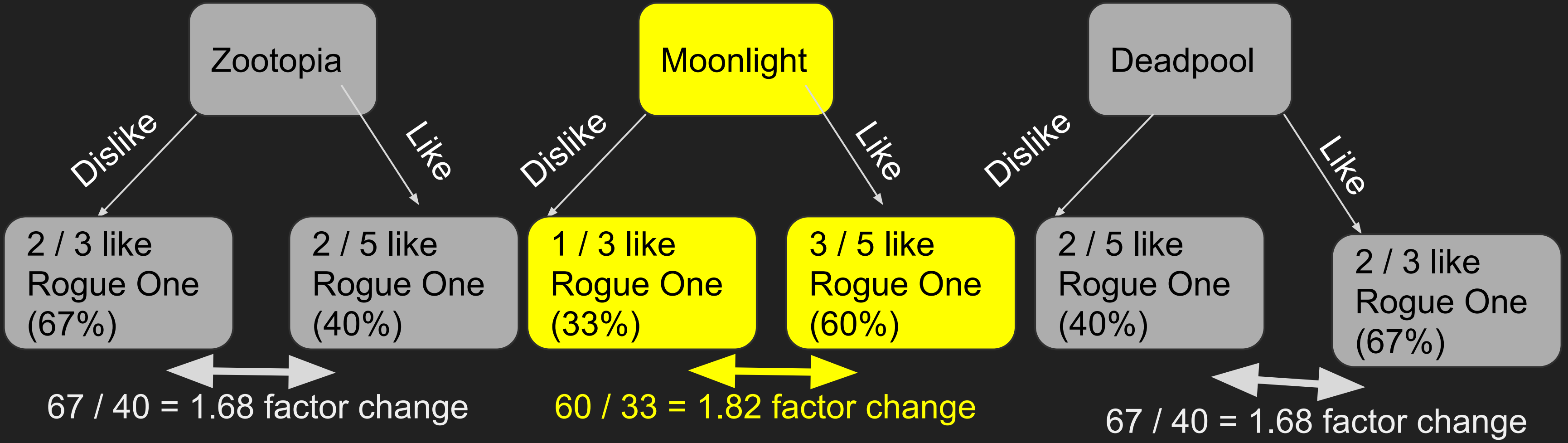
? / ? like Rogue One (?%)

? / ? like Rogue One (?%)

15min

# Let’s do data science: (2b) Construct a “tree” model

(Fill this out in your groups)



Zootopia

Dislike

Like

2 / 3 like Rogue One (67%)

2 / 5 like Rogue One (40%)

67 / 40 = 1.68 factor change

Moonlight

Dislike

Like

1 / 3 like Rogue One (33%)

3 / 5 like Rogue One (60%)

60 / 33 = 1.82 factor change

Deadpool

Dislike

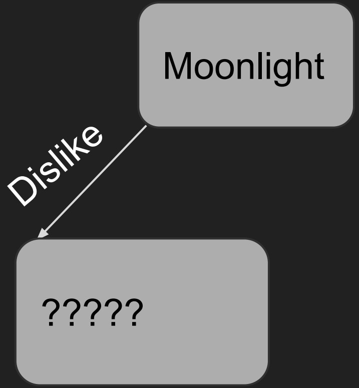
Like

2 / 5 like Rogue One (40%)

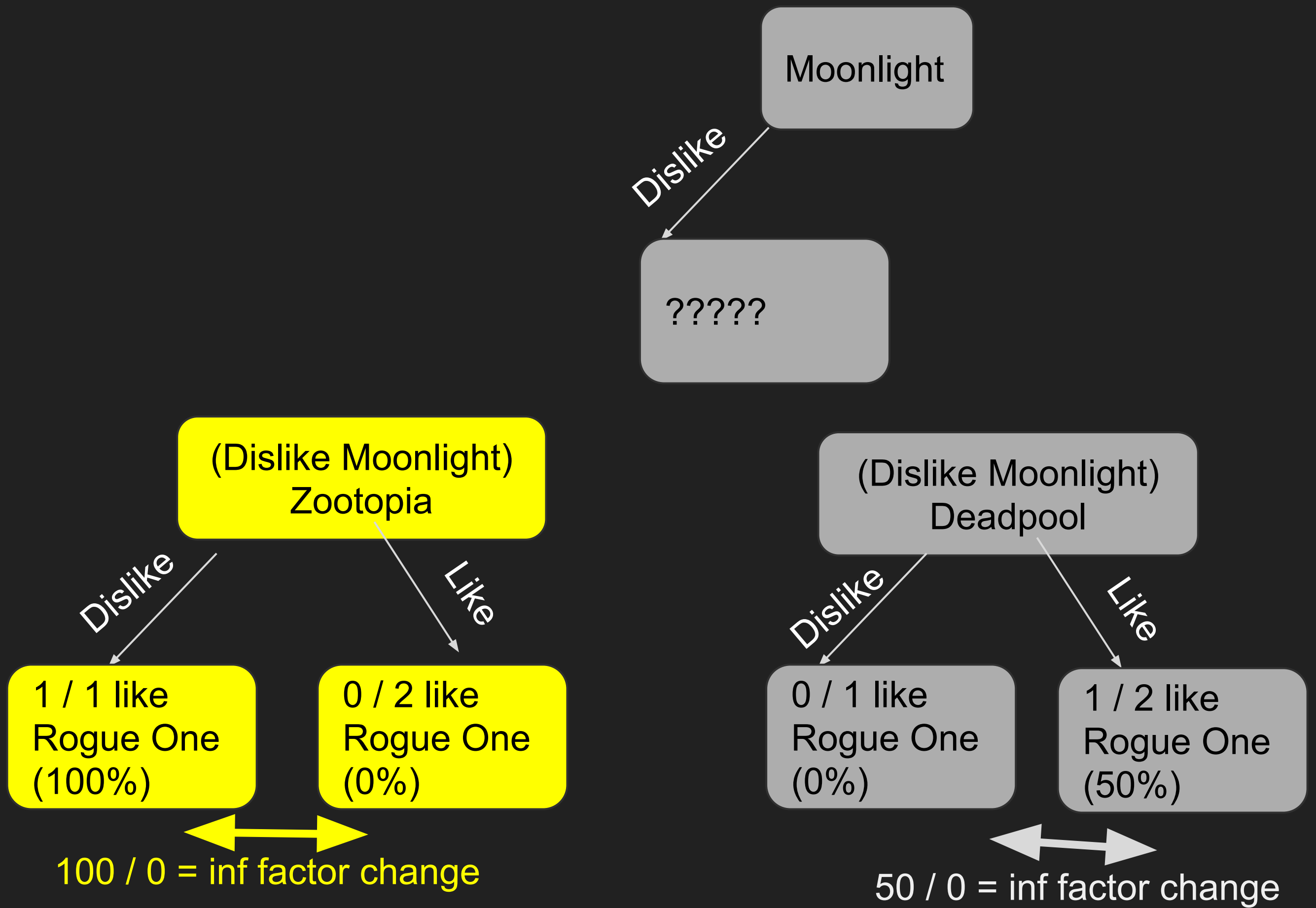
2 / 3 like Rogue One (67%)

67 / 40 = 1.68 factor change

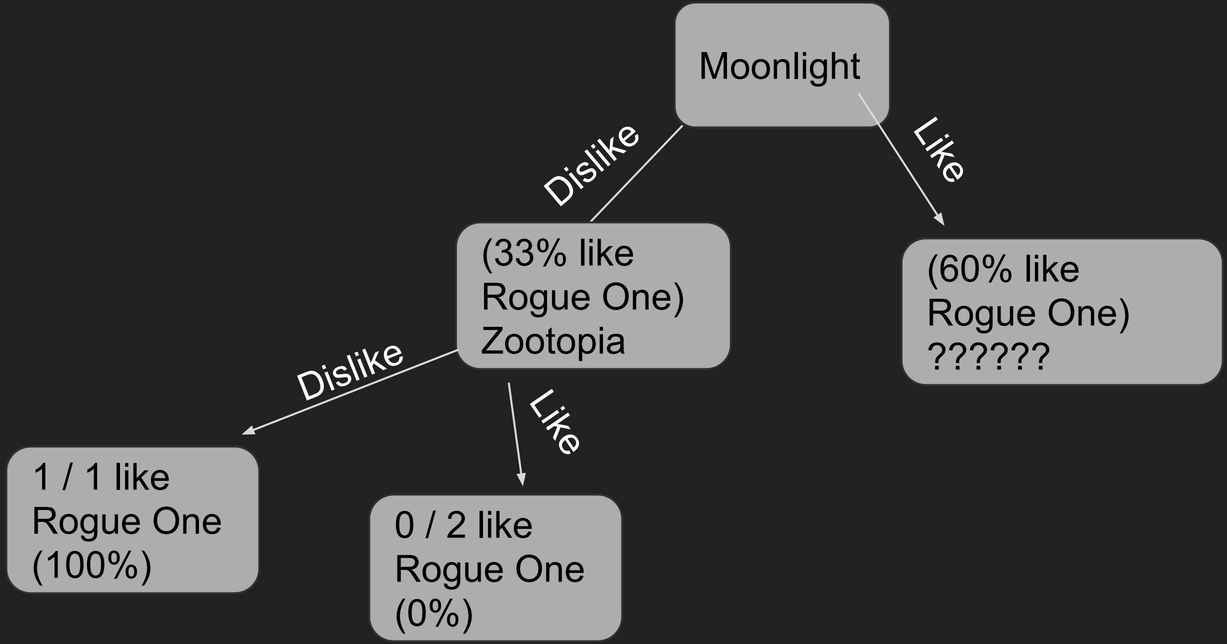
# Let’s do data science: (2b) A visualized decision tree



# Let’s do data science: (2b) A visualized decision tree


Moonlight
Dislike
?????
(Dislike Moonlight) Zootopia
Dislike
Like
1 / 1 like Rogue One (100%)
0 / 2 like Rogue One (0%)
100 / 0 = inf factor change
(Dislike Moonlight) Deadpool
Dislike
Like
0 / 1 like Rogue One (0%)
1 / 2 like Rogue One (50%)
50 / 0 = inf factor change

# Let’s do data science: (2b) A visualized decision tree



Moonlight

Dislike

Like

(33% like Rogue One) Zootopia

(60% like Rogue One) ??????

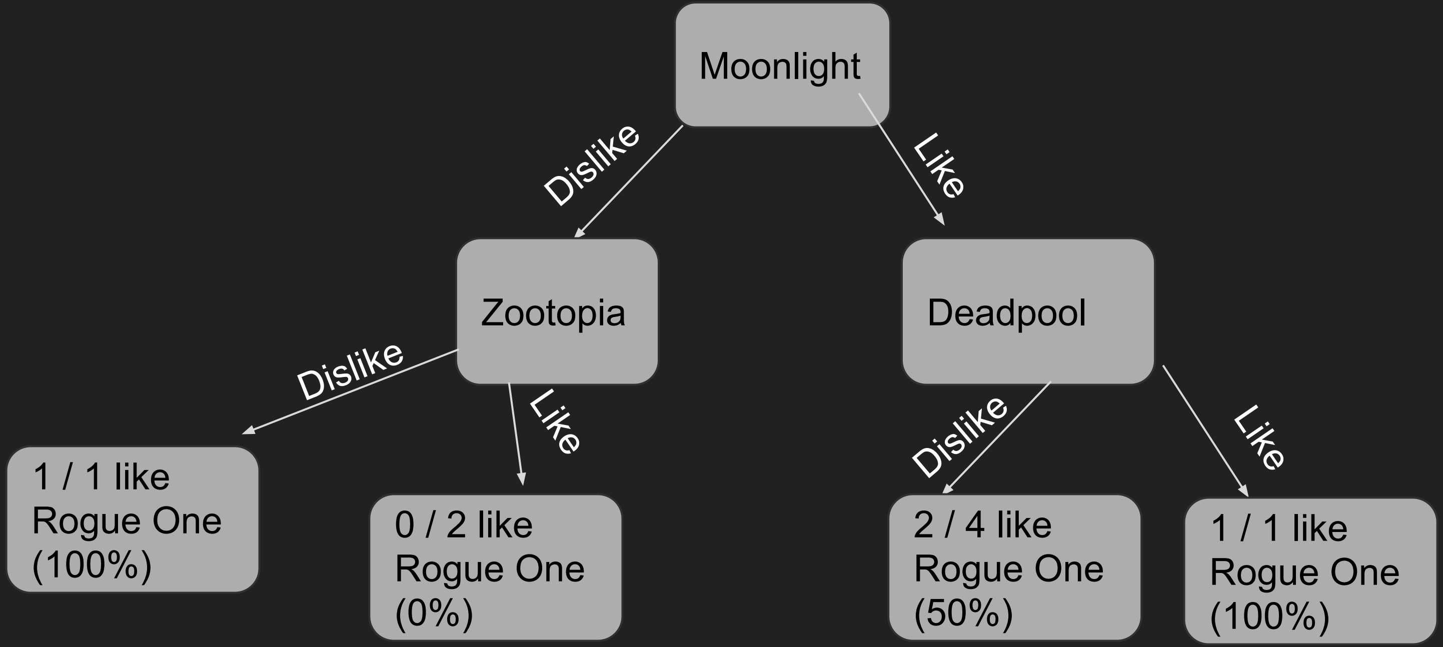
Dislike

Like

1 / 1 like Rogue One (100%)

0 / 2 like Rogue One (0%)

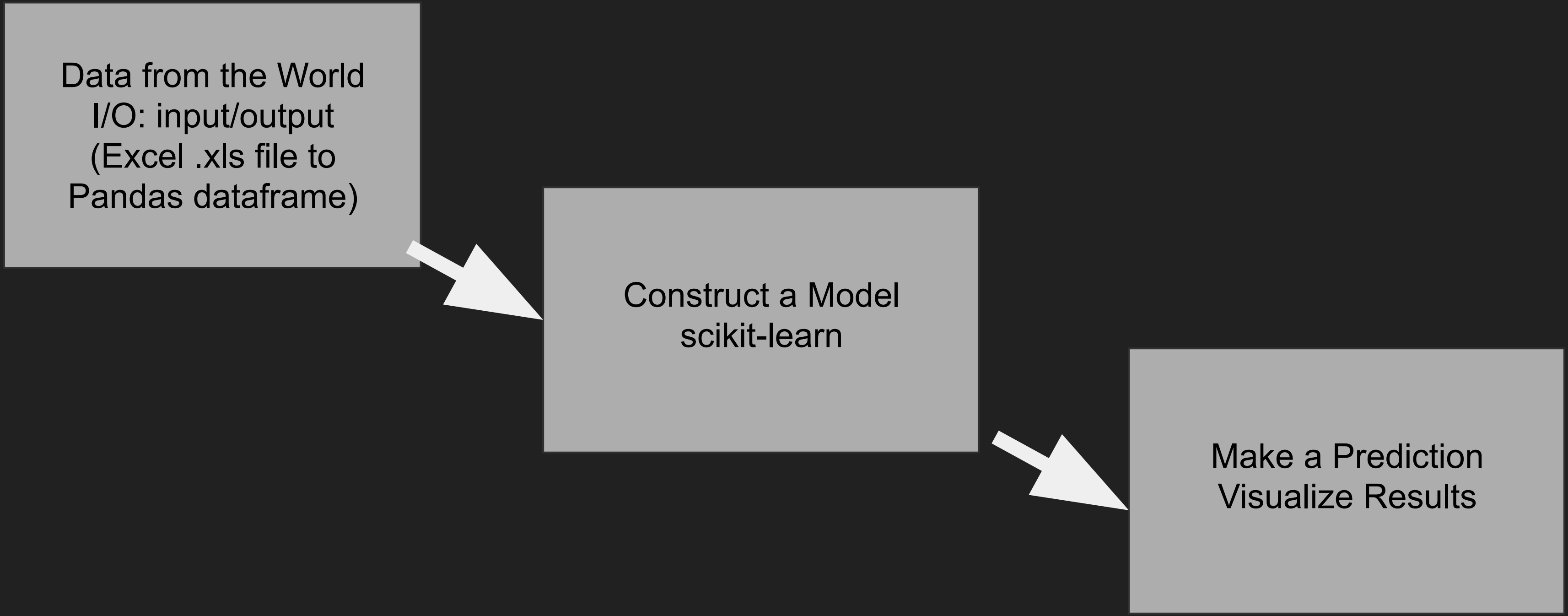
# Let’s do data science: (2b) A visualized decision tree



Moonlight
Dislike
Like
Zootopia
Deadpool
Dislike
Like
Dislike
Like
1 / 1 like Rogue One (100%)
0 / 2 like Rogue One (0%)
2 / 4 like Rogue One (50%)
1 / 1 like Rogue One (100%)

20min

# Let’s do data science: A coding version

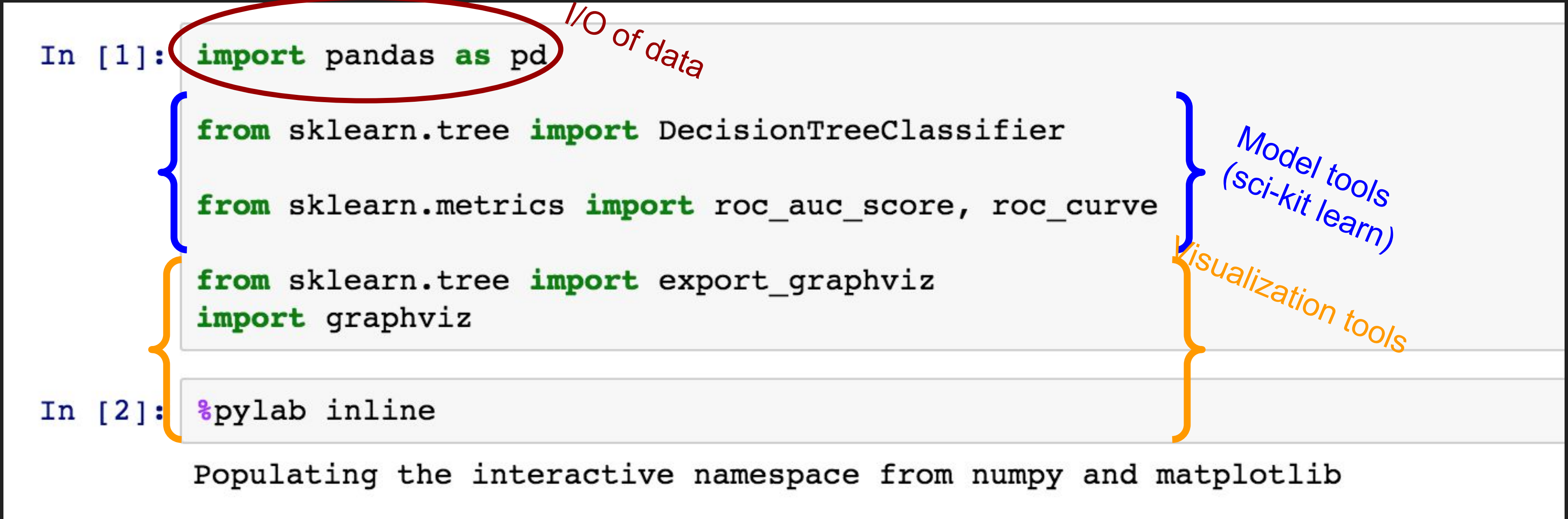


Data from the World
I/O: input/output
(Excel .xls file to
Pandas dataframe)

Construct a Model
scikit-learn

Make a Prediction
Visualize Results

# Let’s do data science: Use built-in tools



```
In [1]: import pandas as pd
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.metrics import roc_auc_score, roc_curve
        from sklearn.tree import export_graphviz
        import graphviz

In [2]: %pylab inline

        Populating the interactive namespace from numpy and matplotlib
```

I/O of data

Model tools (sci-kit learn)

Visualization tools

**Like**
Dislike

# Data science we did: Data Input

**Zootopia**
**Moonlight**
Deadpool
**Rogue One**

Zootopia
Moonlight
**Deadpool**
**Rogue One**

**Zootopia**
Moonlight
Deadpool
Rogue One

**Zootopia**
**Moonlight**
Deadpool
Rogue One

Zootopia
**Moonlight**
Deadpool
Rogue One

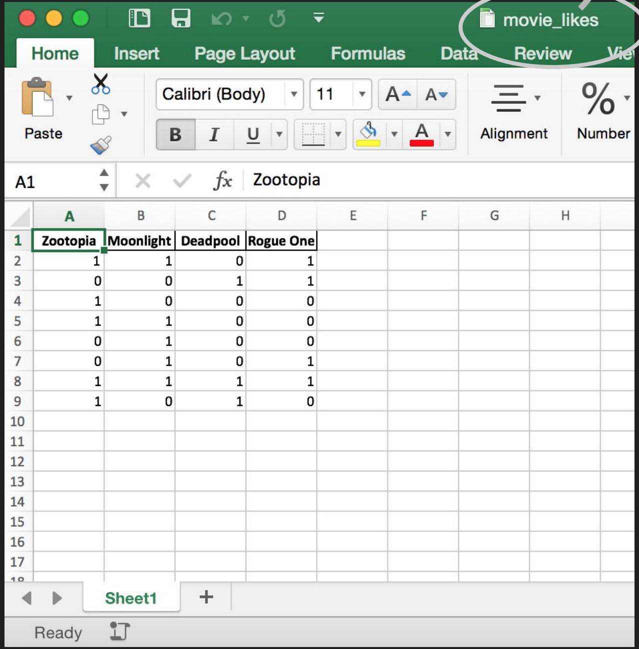
Zootopia
**Moonlight**
Deadpool
**Rogue One**

**Zootopia**
**Moonlight**
**Deadpool**
**Rogue One**

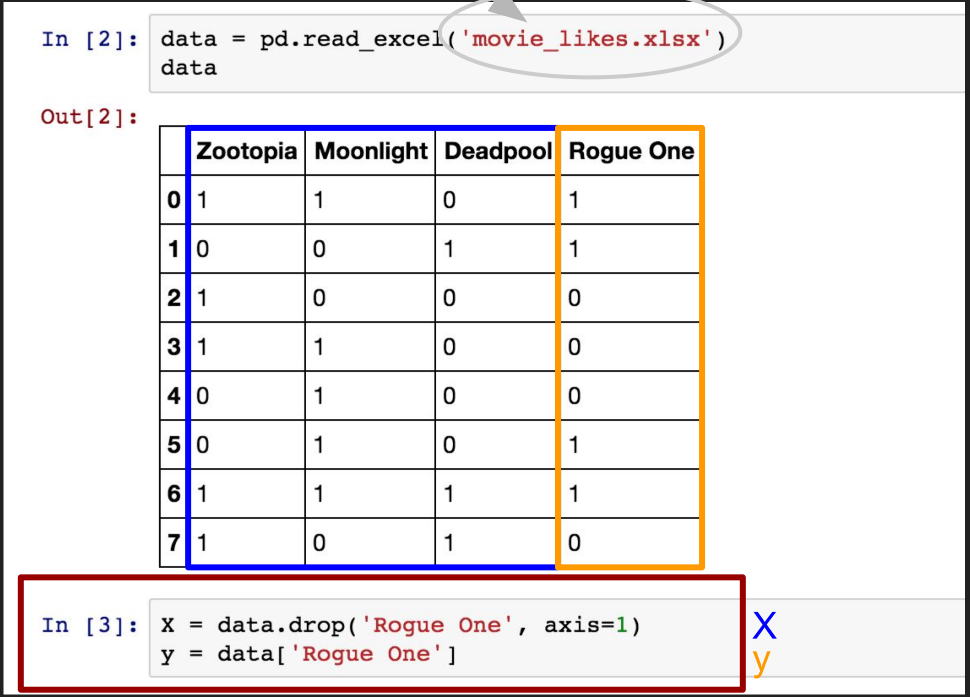
**Zootopia**
Moonlight
**Deadpool**
Rogue One

# Let’s do data science: A coding version (I/O)

Read in the data from Excel



| | A | B | C | D |
|---|---|---|---|---|
| 1 | Zootopia | Moonlight | Deadpool | Rogue One |
| 2 | 1 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 1 | 0 | 0 |
| 6 | 0 | 1 | 0 | 0 |
| 7 | 0 | 1 | 0 | 1 |
| 8 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 1 | 0 |



```
In [2]: data = pd.read_excel('movie_likes.xlsx')
        data
```

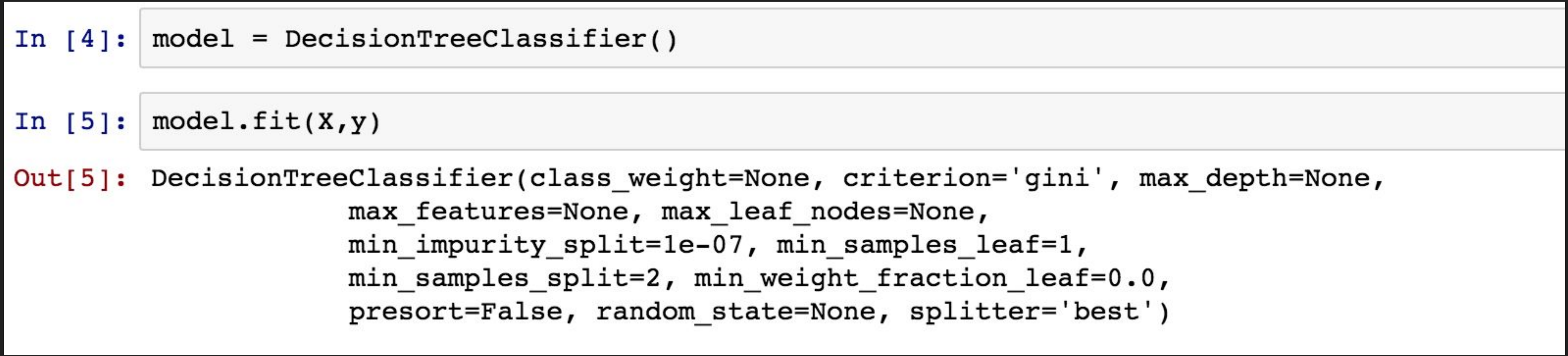
Out[2]:

| | Zootopia | Moonlight | Deadpool | Rogue One |
|---|---|---|---|---|
| **0** | 1 | 1 | 0 | 1 |
| **1** | 0 | 0 | 1 | 1 |
| **2** | 1 | 0 | 0 | 0 |
| **3** | 1 | 1 | 0 | 0 |
| **4** | 0 | 1 | 0 | 0 |
| **5** | 0 | 1 | 0 | 1 |
| **6** | 1 | 1 | 1 | 1 |
| **7** | 1 | 0 | 1 | 0 |

```
In [3]: X = data.drop('Rogue One', axis=1)
        y = data['Rogue One']
```

# Let’s do data science: Build a Model

Build a decision tree (using tools from python sci-kit learn library)

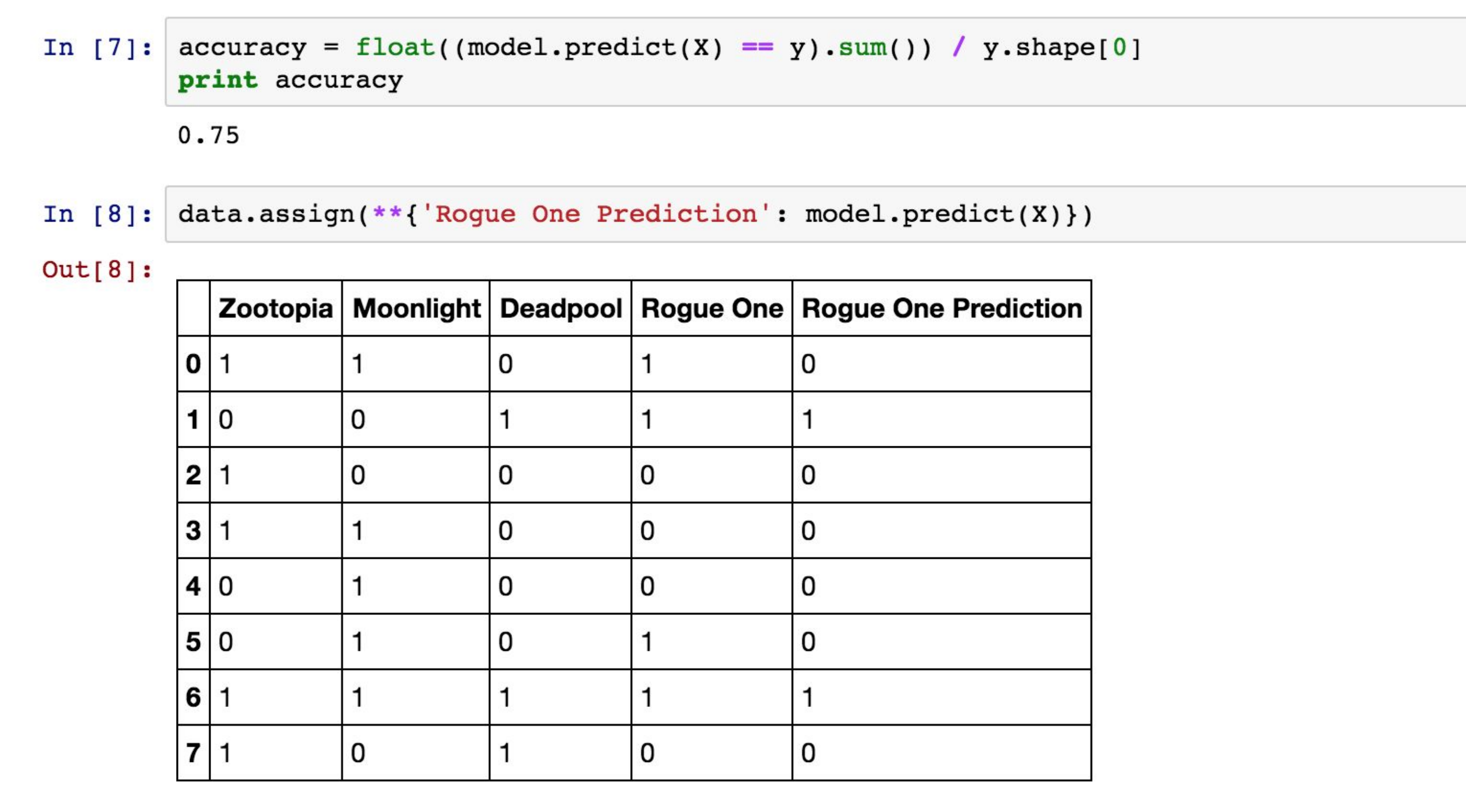


```
In [4]: model = DecisionTreeClassifier()

In [5]: model.fit(X,y)

Out[5]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                    max_features=None, max_leaf_nodes=None,
                    min_impurity_split=1e-07, min_samples_leaf=1,
                    min_samples_split=2, min_weight_fraction_leaf=0.0,
                    presort=False, random_state=None, splitter='best')
```

# Let’s do data science: Visualize your Predictive Power



In [7]:

```
accuracy = float((model.predict(X) == y).sum()) / y.shape[0]
print accuracy
```

```
0.75
```

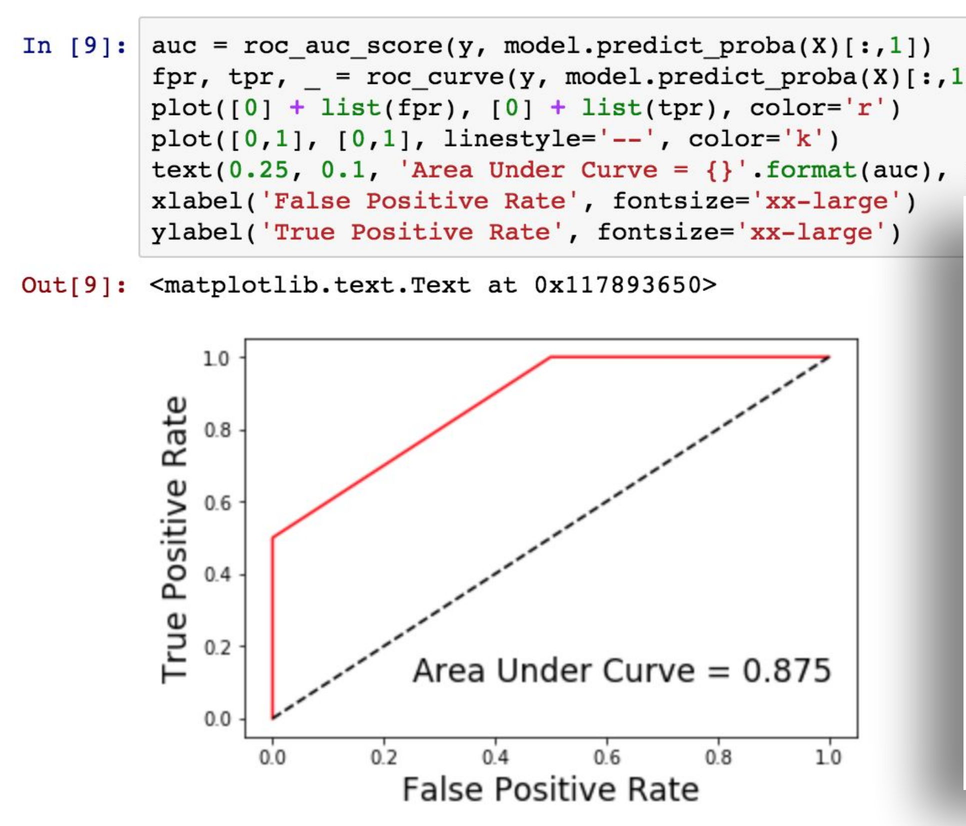
In [8]:

```
data.assign(**{'Rogue One Prediction': model.predict(X)})
```

Out[8]:

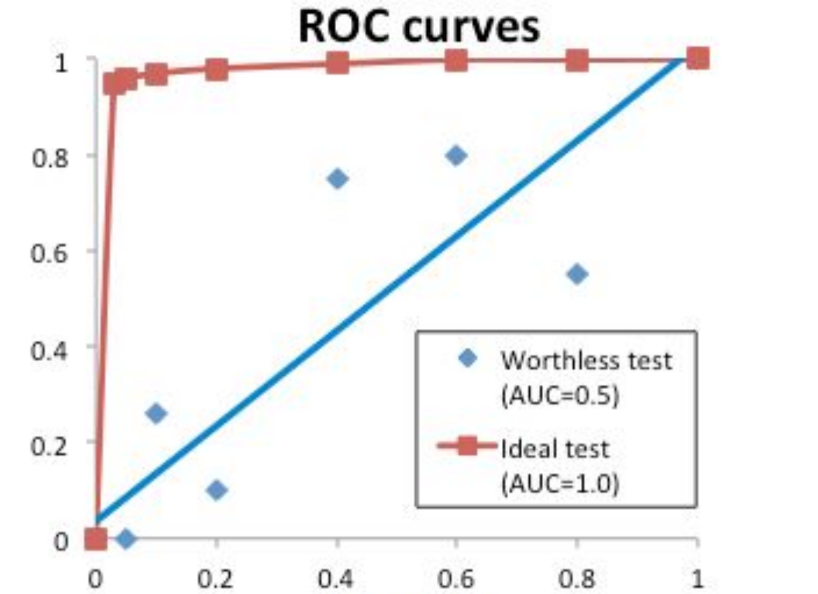
|   | Zootopia | Moonlight | Deadpool | Rogue One | Rogue One Prediction |
|---|---|---|---|---|---|
| **0** | 1 | 1 | 0 | 1 | 0 |
| **1** | 0 | 0 | 1 | 1 | 1 |
| **2** | 1 | 0 | 0 | 0 | 0 |
| **3** | 1 | 1 | 0 | 0 | 0 |
| **4** | 0 | 1 | 0 | 0 | 0 |
| **5** | 0 | 1 | 0 | 1 | 0 |
| **6** | 1 | 1 | 1 | 1 | 1 |
| **7** | 1 | 0 | 1 | 0 | 0 |

# Let’s do data science: Visualize your Predictive Power

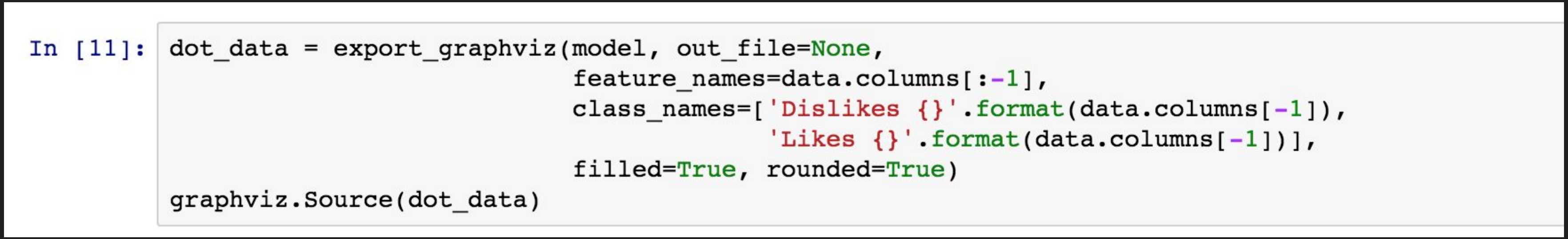


```
In [9]: auc = roc_auc_score(y, model.predict_proba(X)[:,1])
        fpr, tpr, _ = roc_curve(y, model.predict_proba(X)[:,1])
        plot([0] + list(fpr), [0] + list(tpr), color='r')
        plot([0,1], [0,1], linestyle='--', color='k')
        text(0.25, 0.1, 'Area Under Curve = {}'.format(auc), fontsize='xx-large')
        xlabel('False Positive Rate', fontsize='xx-large')
        ylabel('True Positive Rate', fontsize='xx-large')

Out[9]: <matplotlib.text.Text at 0x117893650>
```



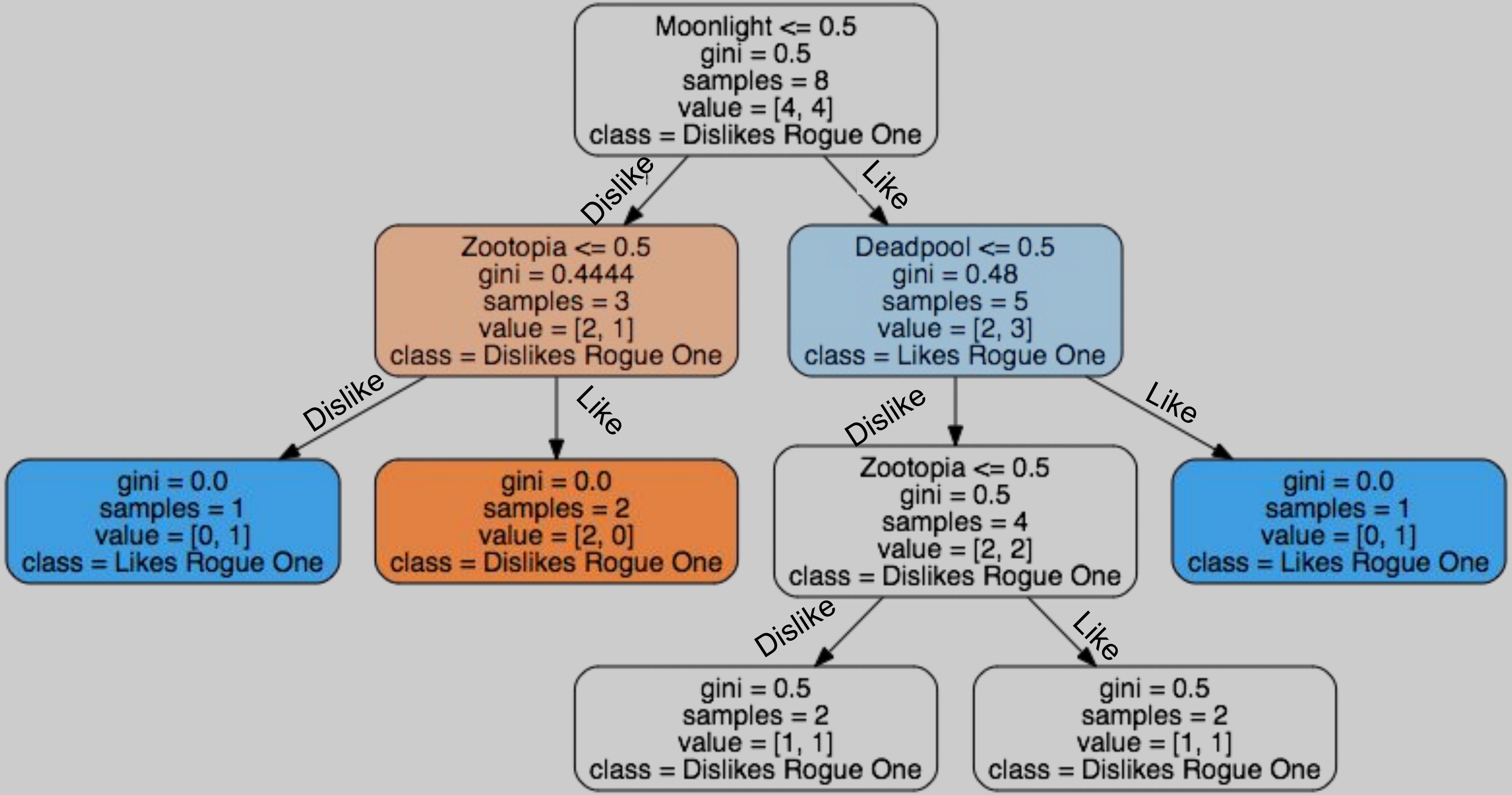
# Let’s do data science: Visualize your Model



In [11]:

```
dot_data = export_graphviz(model, out_file=None,
                           feature_names=data.columns[:-1],
                           class_names=['Dislikes {}'.format(data.columns[-1]),
                                        'Likes {}'.format(data.columns[-1])],
                           filled=True, rounded=True)
graphviz.Source(dot_data)
```

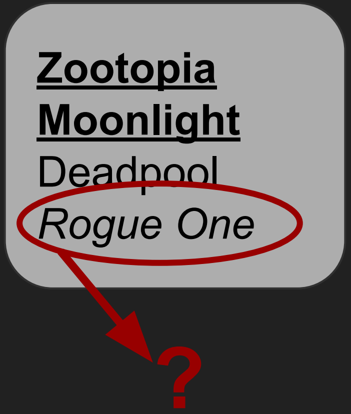
# Data science we did: (2) The “Machine’s” Model



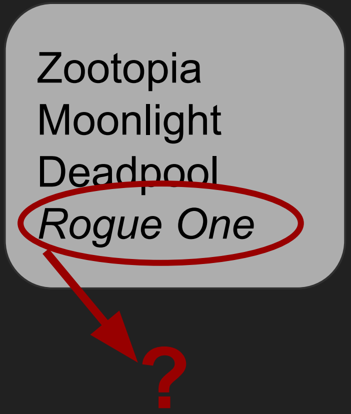
10min

# Data science we did: (3) The “Machine’s” Prediction

What did the machine predict?

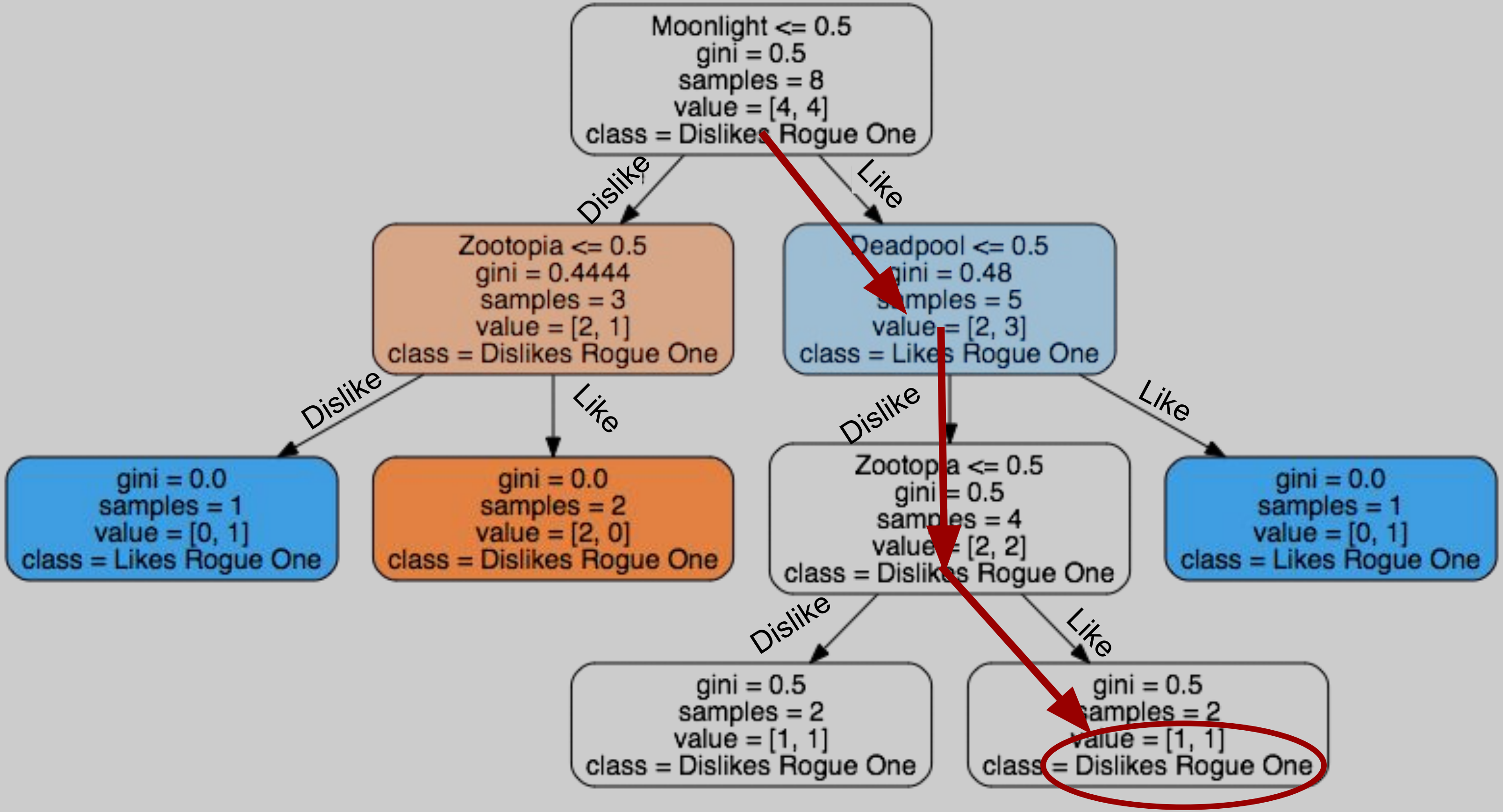


**<u>Zootopia</u>**
**<u>Moonlight</u>**
Deadpool
*Rogue One*



Zootopia
Moonlight
Deadpool
*Rogue One*

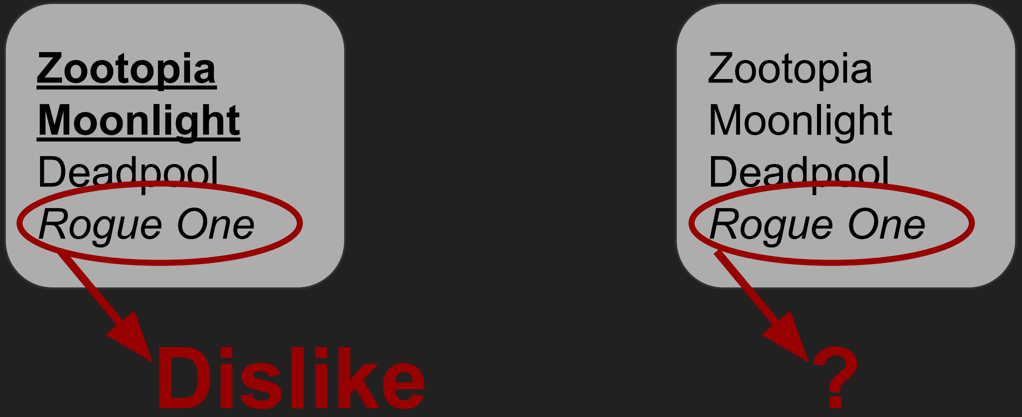
# Data science we did: (2) The “Machine’s” Model



10min

# Data science we did: (3) The “machine’s” Prediction

What did the machine predict?



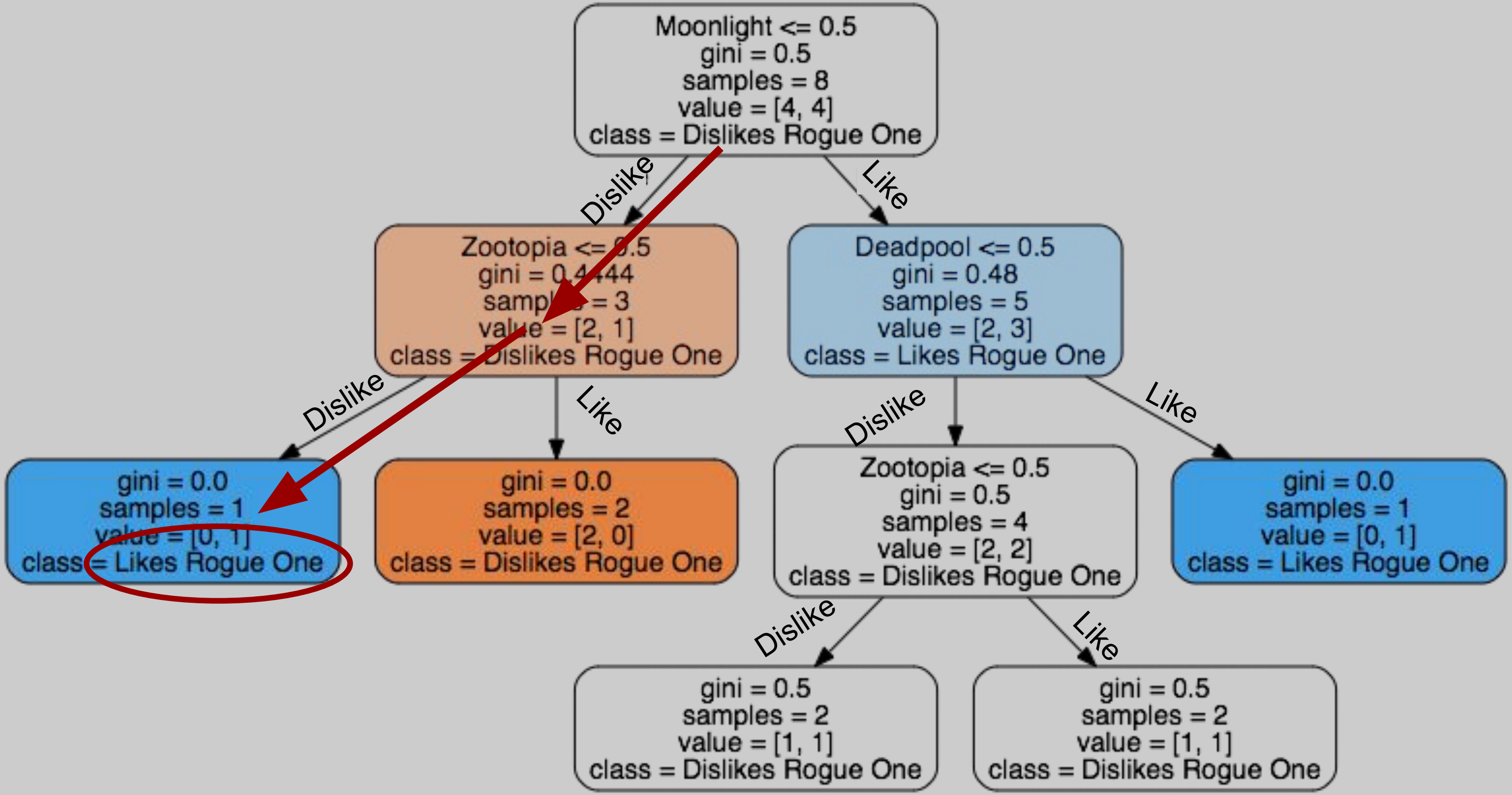
**Zootopia**
**Moonlight**
Deadpool
*Rogue One*

Dislike

Zootopia
Moonlight
Deadpool
*Rogue One*

?

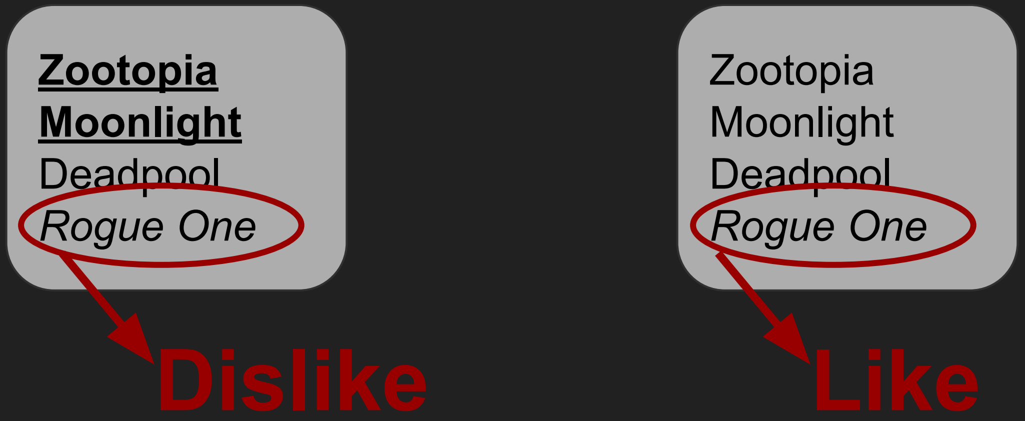
# Data science we did: (2) The “Machine’s” Model



10min

# Data science we did: (3) The “Machine’s” Prediction

What did the machine predict?



**Zootopia**
**Moonlight**
Deadpool
*Rogue One*

Dislike

Zootopia
Moonlight
Deadpool
*Rogue One*

Like

40min

# Do **your** coding version of data science

(1) Identify Skills you Wish to Craft

(2) Join or Develop a Community

(3) Find Resources

(4) Practice, practice, practice

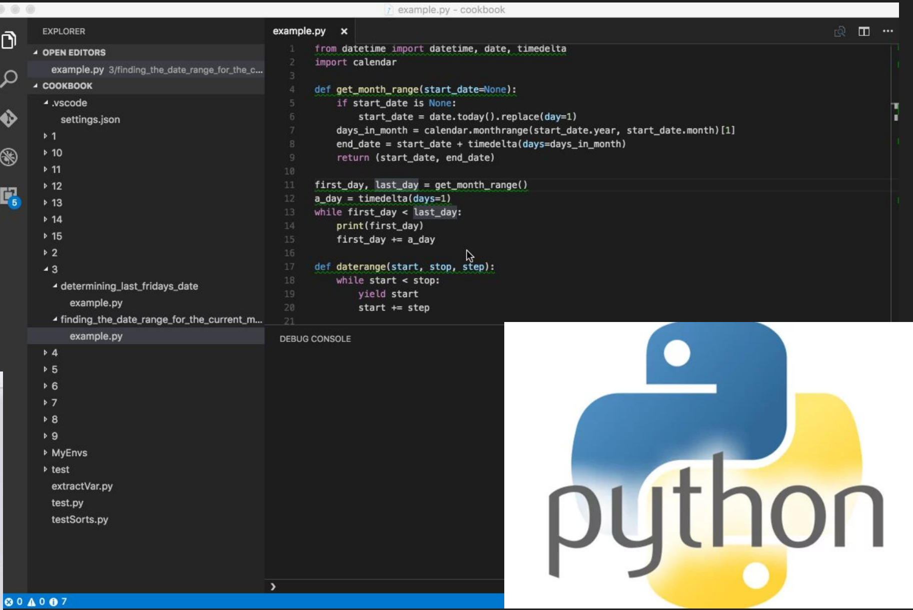
# Do your coding version of data science

(1) Identify Skills you Wish to Craft

(2) Join or Develop a Community

(3) Find Resources

(4) Practice, practice, practice

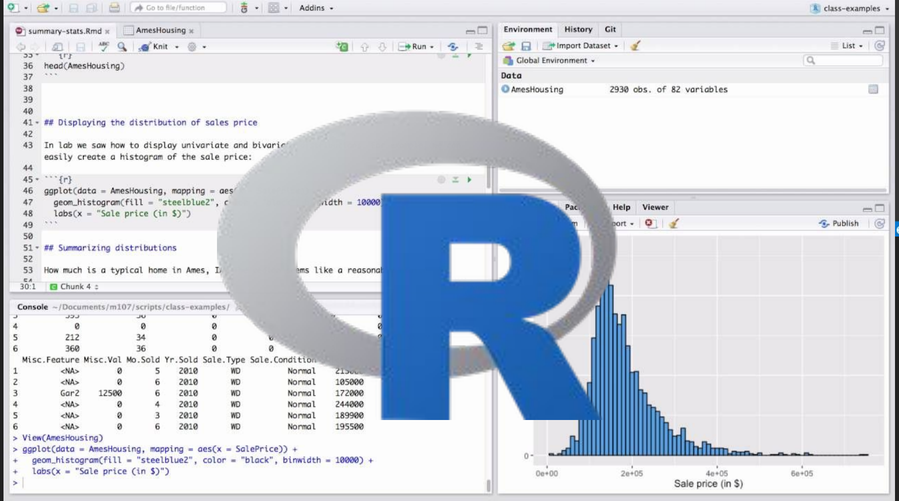


```python
from datetime import datetime, date, timedelta
import calendar

def get_month_range(start_date=None):
    if start_date is None:
        start_date = date.today().replace(day=1)
    days_in_month = calendar.monthrange(start_date.year, start_date.month)[1]
    end_date = start_date + timedelta(days=days_in_month)
    return (start_date, end_date)

first_day, last_day = get_month_range()
a_day = timedelta(days=1)
while first_day < last_day:
    print(first_day)
    first_day += a_day

def daterange(start, stop, step):
    while start < stop:
        yield start
        start += step
```



AmesHousing 2930 obs. of 82 variables

```
head(AmesHousing)
```

## Displaying the distribution of sales price

In lab we saw how to display univariate and bivaria
easily create a histogram of the sale price:

```{r}
ggplot(data = AmesHousing, mapping = aes
  geom_histogram(fill = "steelblue2", c                    idth = 10000
  labs(x = "Sale price (in $)")
```

## Summarizing distributions

How much is a typical home in Ames, I                  ems like a reasonal

```
4         0          0          0
5       212         34          0          0
6       360         36          0          0
  Misc.Feature Misc.Val Mo.Sold Yr.Sold Sale.Type Sale.Condition
1         <NA>        0       5    2010       WD          Normal  215
2         <NA>        0       6    2010       WD          Normal  105000
3         Gar2    12500       6    2010       WD          Normal  172000
4         <NA>        0       4    2010       WD          Normal  244000
5         <NA>        0       3    2010       WD          Normal  189900
6         <NA>        0       6    2010       WD          Normal  195500
> View(AmesHousing)
> ggplot(data = AmesHousing, mapping = aes(x = SalePrice)) +
+   geom_histogram(fill = "steelblue2", color = "black", binwidth = 10000) +
+   labs(x = "Sale price (in $)")
>
```

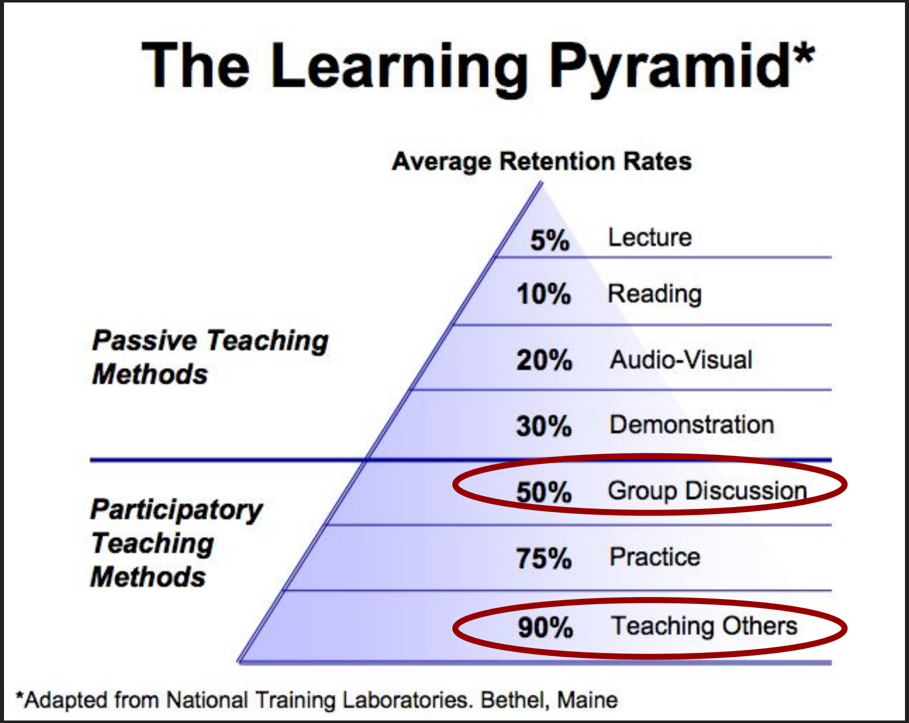
# Do your coding version of data science

(1) Identify Skills you Wish to Craft

(2) Join or Develop a Community

(3) Find Resources

(4) Practice, practice, practice



The Learning Pyramid*

*Adapted from National Training Laboratories. Bethel, Maine

# Do your coding version of data science

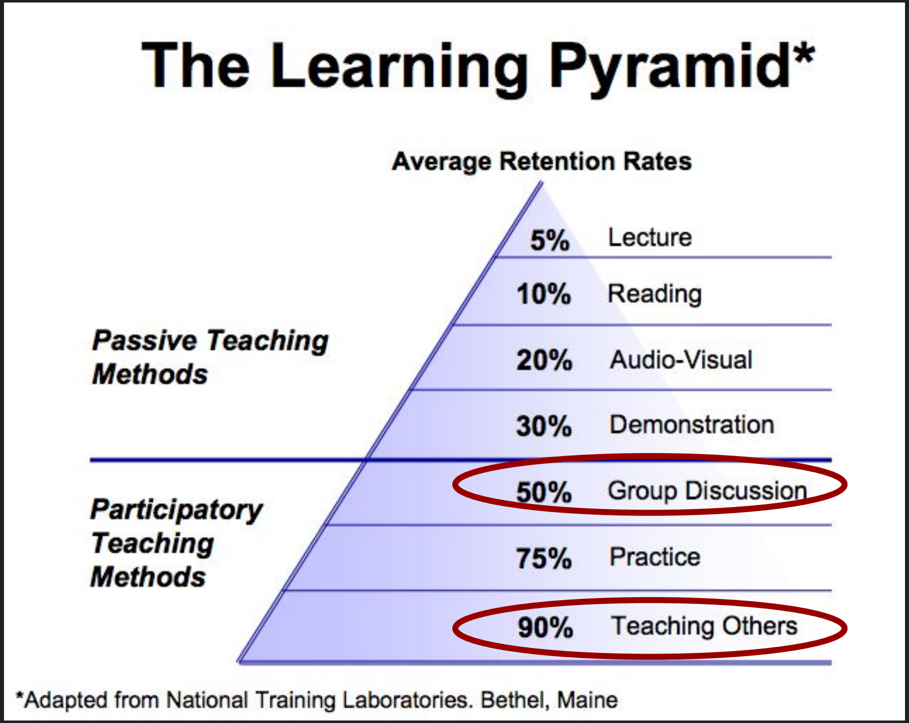
(1) Identify Skills you Wish to Craft

(2) Join or Develop a Community

(3) Find Resources

(4) Practice, practice, practice

Study buddies,
online communities,
lab co-workers,
fellow Amsterdam
summer students!



The Learning Pyramid*

Average Retention Rates

*Passive Teaching Methods*

5% Lecture

10% Reading

20% Audio-Visual

30% Demonstration

*Participatory Teaching Methods*

50% Group Discussion

75% Practice

90% Teaching Others

*Adapted from National Training Laboratories. Bethel, Maine

# Do your coding version of data science

(1) Identify Skills you Wish to Craft

(2) Join or Develop a Community

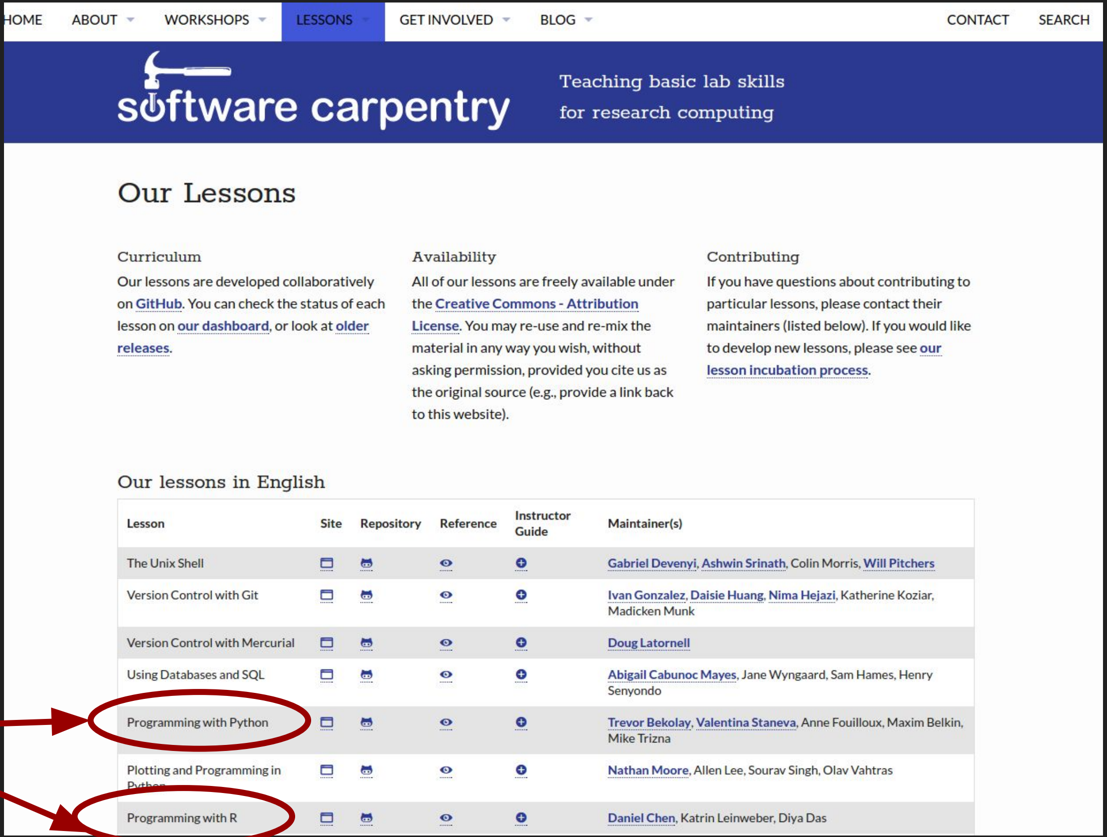
(3) Find Resources

(4) Practice, practice, practice

software
carpentry

DATA
CARPENTRY

# Online Lesson Material: Software and Data Carpentry



Teaching basic lab skills
for research computing

## Our Lessons

### Curriculum

Our lessons are developed collaboratively on GitHub. You can check the status of each lesson on our dashboard, or look at older releases.

### Availability

All of our lessons are freely available under the Creative Commons - Attribution License. You may re-use and re-mix the material in any way you wish, without asking permission, provided you cite us as the original source (e.g., provide a link back to this website).

### Contributing

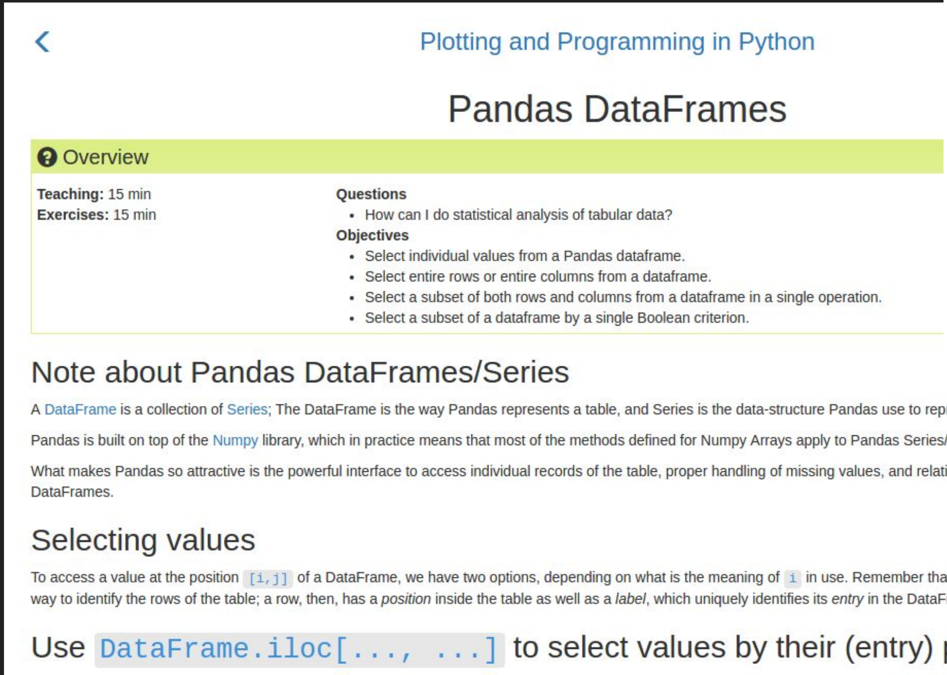
If you have questions about contributing to particular lessons, please contact their maintainers (listed below). If you would like to develop new lessons, please see our lesson incubation process.

### Our lessons in English

| Lesson | Site | Repository | Reference | Instructor Guide | Maintainer(s) |
|---|---|---|---|---|---|
| The Unix Shell | ▭ | 🐱 | 👁 | ⊕ | Gabriel Devenyi, Ashwin Srinath, Colin Morris, Will Pitchers |
| Version Control with Git | ▭ | 🐱 | 👁 | ⊕ | Ivan Gonzalez, Daisie Huang, Nima Hejazi, Katherine Koziar, Madicken Munk |
| Version Control with Mercurial | ▭ | 🐱 | 👁 | ⊕ | Doug Latornell |
| Using Databases and SQL | ▭ | 🐱 | 👁 | ⊕ | Abigail Cabunoc Mayes, Jane Wyngaard, Sam Hames, Henry Senyondo |
| Programming with Python | ▭ | 🐱 | 👁 | ⊕ | Trevor Bekolay, Valentina Staneva, Anne Fouilloux, Maxim Belkin, Mike Trizna |
| Plotting and Programming in Python | ▭ | 🐱 | 👁 | ⊕ | Nathan Moore, Allen Lee, Sourav Singh, Olav Vahtras |
| Programming with R | ▭ | 🐱 | 👁 | ⊕ | Daniel Chen, Katrin Leinweber, Diya Das |

Python,
R, .....

# Online Lesson Material



Plotting and Programming in Python

## Pandas DataFrames

**Overview**

**Teaching:** 15 min
**Exercises:** 15 min

**Questions**

- How can I do statistical analysis of tabular data?

**Objectives**

- Select individual values from a Pandas dataframe.
- Select entire rows or entire columns from a dataframe.
- Select a subset of both rows and columns from a dataframe in a single operation.
- Select a subset of a dataframe by a single Boolean criterion.

### Note about Pandas DataFrames/Series

A DataFrame is a collection of Series; The DataFrame is the way Pandas represents a table, and Series is the data-structure Pandas use to represent a column.

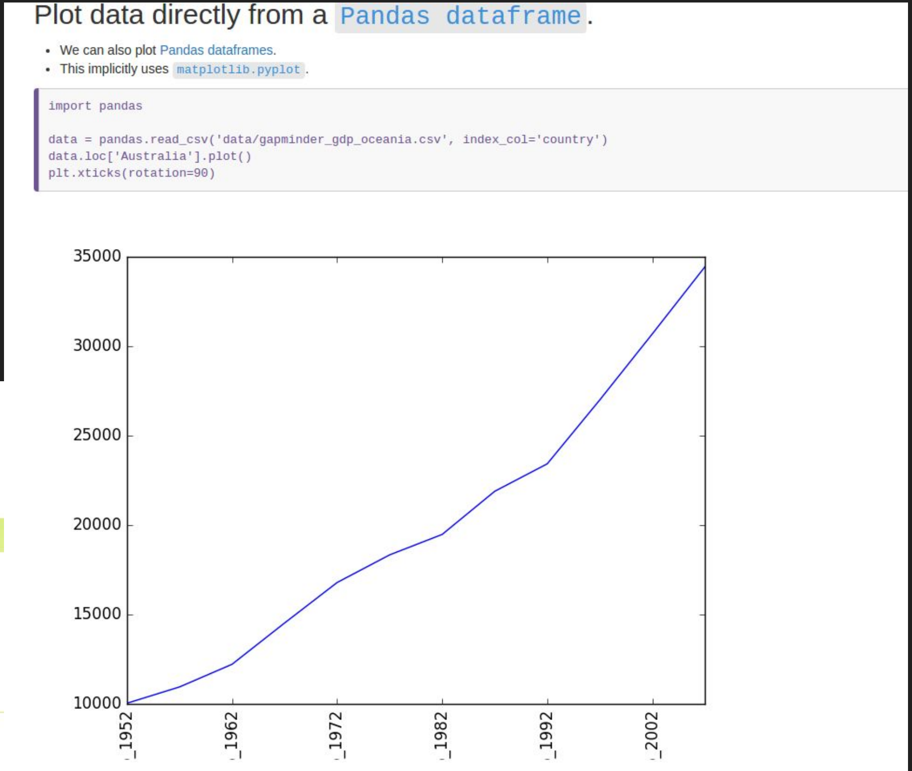
Pandas is built on top of the Numpy library, which in practice means that most of the methods defined for Numpy Arrays apply to Pandas Series/DataFrames.

What makes Pandas so attractive is the powerful interface to access individual records of the table, proper handling of missing values, and relational-databases operations between DataFrames.

### Selecting values

To access a value at the position `[i,j]` of a DataFrame, we have two options, depending on what is the meaning of `i` in use. Remember that a DataFrame provides a *index* as a way to identify the rows of the table; a row, then, has a *position* inside the table as well as a *label*, which uniquely identifies its *entry* in the DataFrame.

### Use `DataFrame.iloc[..., ...]` to select values by their (entry) position



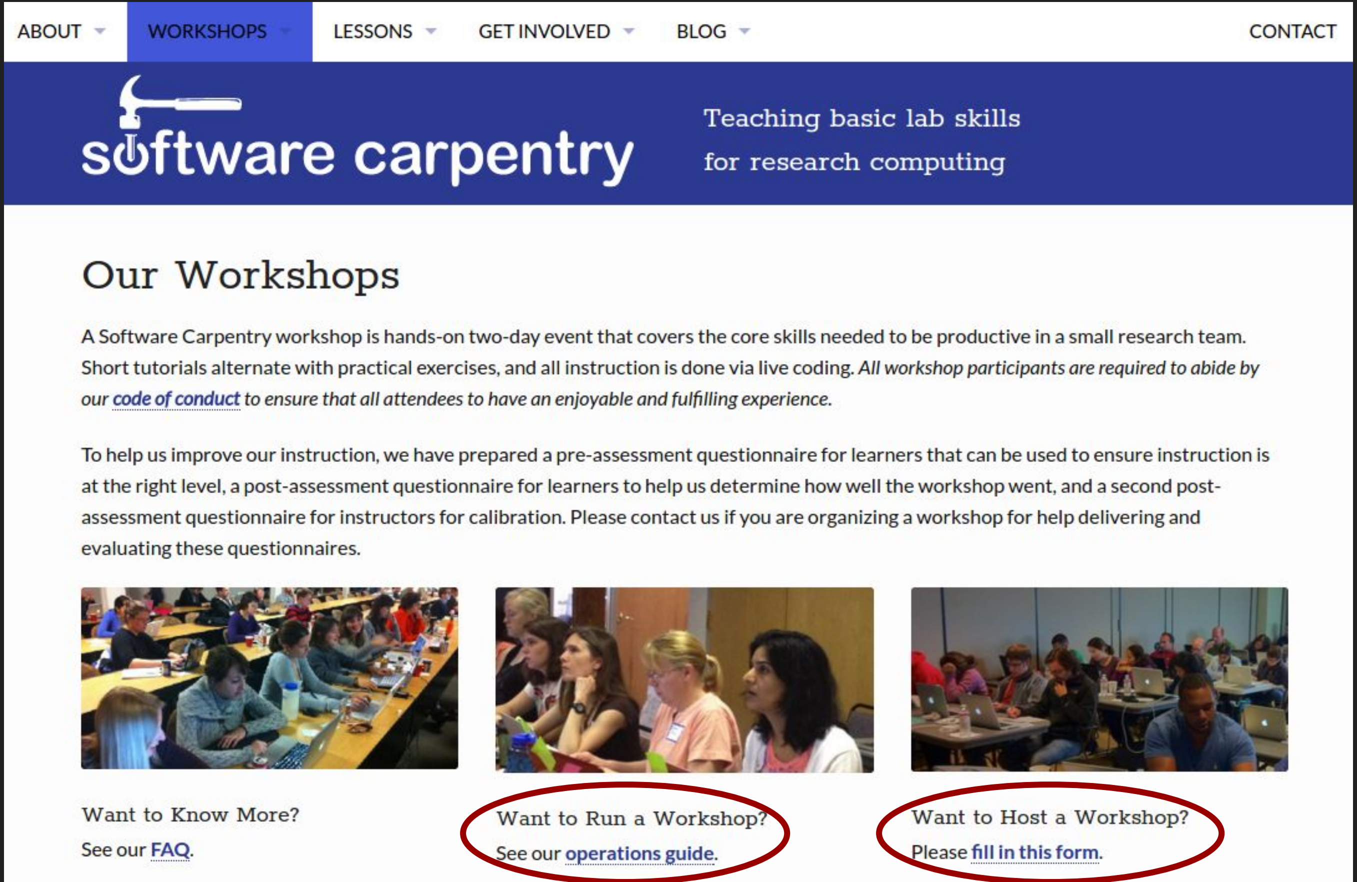
### Plot data directly from a `Pandas dataframe`.

- We can also plot Pandas dataframes.
- This implicitly uses `matplotlib.pyplot`.

```
import pandas

data = pandas.read_csv('data/gapminder_gdp_oceania.csv', index_col='country')
data.loc['Australia'].plot()
plt.xticks(rotation=90)
```

# Online Lesson Material: Host/Attend a Workshop



## Our Workshops

A Software Carpentry workshop is hands-on two-day event that covers the core skills needed to be productive in a small research team. Short tutorials alternate with practical exercises, and all instruction is done via live coding. *All workshop participants are required to abide by our code of conduct to ensure that all attendees to have an enjoyable and fulfilling experience.*

To help us improve our instruction, we have prepared a pre-assessment questionnaire for learners that can be used to ensure instruction is at the right level, a post-assessment questionnaire for learners to help us determine how well the workshop went, and a second post-assessment questionnaire for instructors for calibration. Please contact us if you are organizing a workshop for help delivering and evaluating these questionnaires.

Want to Know More?

See our FAQ.

Want to Run a Workshop?

See our operations guide.

Want to Host a Workshop?

Please fill in this form.

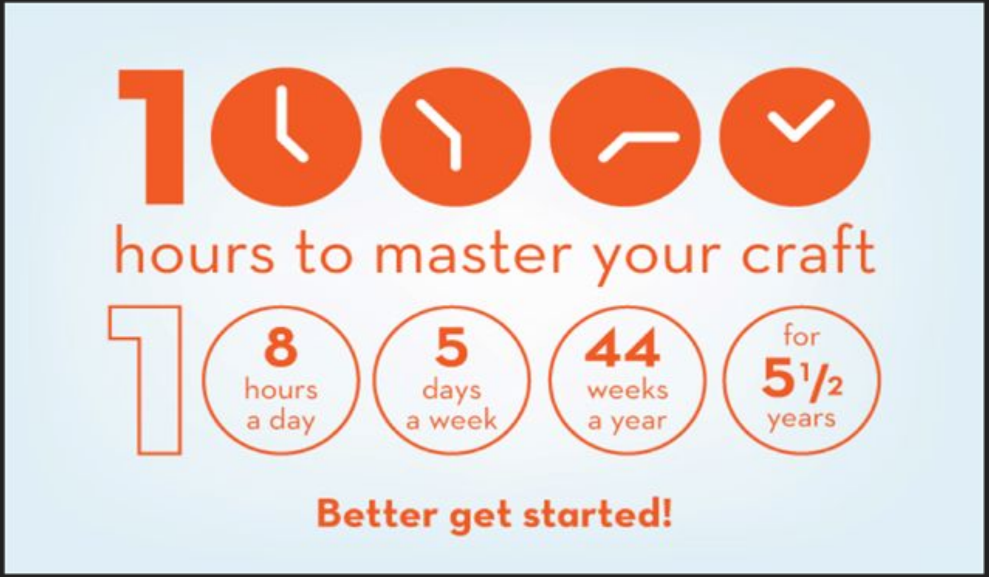
# Do your coding version of data science

(1) Identify Skills you Wish to Craft

(2) Join or Develop a Community

(3) Find Resources

(4) Practice, practice, practice



# Do your coding version of data science

(1) Identify Skills you Wish to Craft

(2) Join or Develop a Community

(3) Find Resources

(4) Practice, practice, practice

1
hours to master your craft
1
8 hours a day
5 days a week
44 weeks a year
for 5½ years
Better get started!

[LINK TO PRACTICE HERE]