# Data Science and Visualization (DSV, F23)

## 10. Data Science in Practice

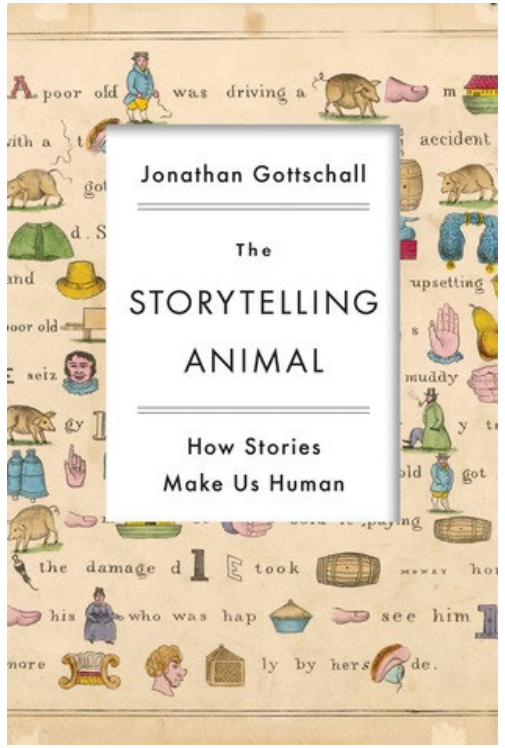
Hua Lu
https://luhua.ruc.dk; luhua@ruc.dk
PLIS, IMT, RUC

# Agenda

- Storytelling with data
  - More than mere visualization
- MLOps
- Finale of the course

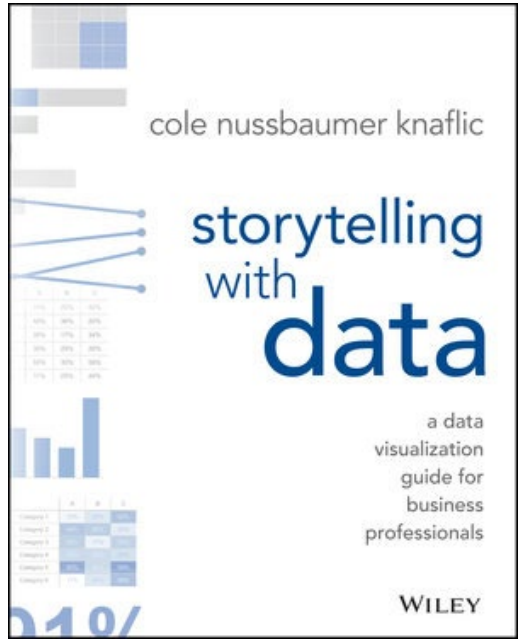
# Storytelling with data

- **What**
  - Communicate something to someone using data
    - Communicate effectively with data
- **Why**
  - Stories could be attractive:
    - *"We are, as a species, addicted to story. Even when the body goes to sleep, the mind stays up all night, telling itself stories."*
      — Jonathan Gottschall,
      The Storytelling Animal: How Stories Make Us Human
  - Data could be convincing
- **How**
  - Data visualization, and
  - Other techniques



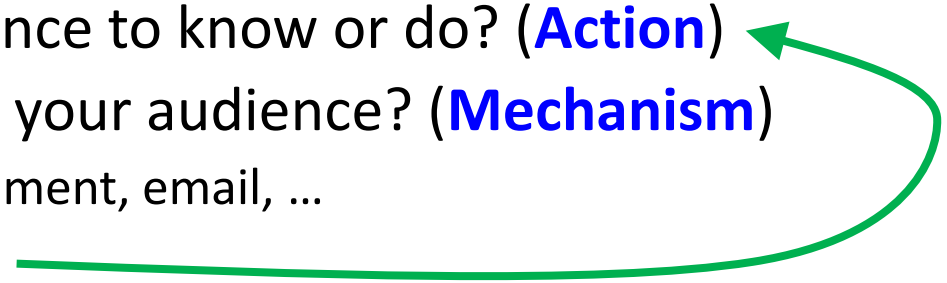
# Storytelling with Data: 6 Key Lessons

1. Understand the context
2. Choose an appropriate visual display
3. Eliminate clutter
4. Focus attention where you want it
5. Think like a designer
6. Tell a story



# 1. Understand the context

- Exploratory vs. explanatory analysis
  - Exploratory data analysis is for *yourself*
    - To understand the data and figure out what might be interesting to others
  - Explanatory data analysis is for *others*
    - Tell them the story with *highlighted* points
- Context: Who, What, How
  - Who is your audience?
  - What do you need your audience to know or do? (**Action**)
  - How will you communicate to your audience? (**Mechanism**)
    - Live presentation, written document, email, …
    - End with a clear '**call to action**'

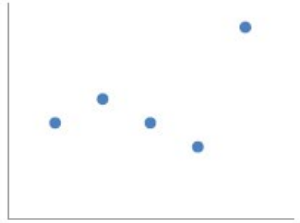


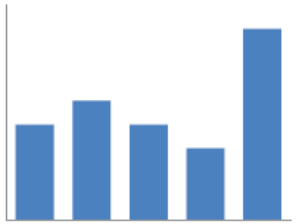
# 2. Choose an appropriate visual display

- Which would be the best, i.e., most effective, to tell your story?

- More graph types at
  - https://datavizproject.com/

Simple text


Scatterplot

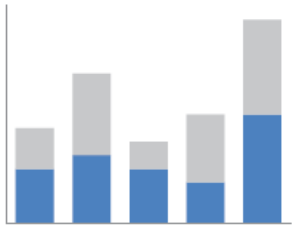

Vertical bar

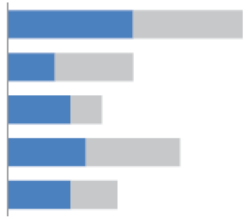

Horizontal bar

| | A | B | C |
|---|---|---|---|
| Category 1 | 15% | 22% | 42% |
| Category 2 | 40% | 36% | 20% |
| Category 3 | 35% | 17% | 34% |
| Category 4 | 30% | 29% | 26% |
| Category 5 | 55% | 30% | 58% |
| Category 6 | 11% | 25% | 49% |

Table

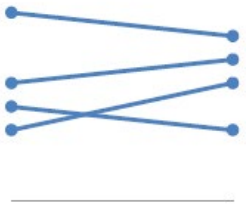

Line

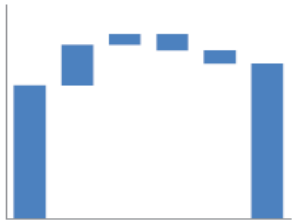

Stacked vertical bar

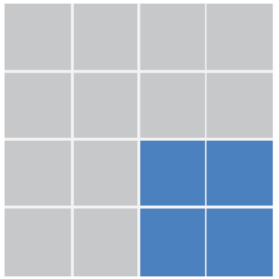

Stacked horizontal bar

| | A | B | C |
|---|---|---|---|
| Category 1 | 15% | 22% | 42% |
| Category 2 | 40% | 36% | 20% |
| Category 3 | 35% | 17% | 34% |
| Category 4 | 30% | 29% | 26% |
| Category 5 | 55% | 30% | 58% |
| Category 6 | 11% | 25% | 49% |

Heatmap

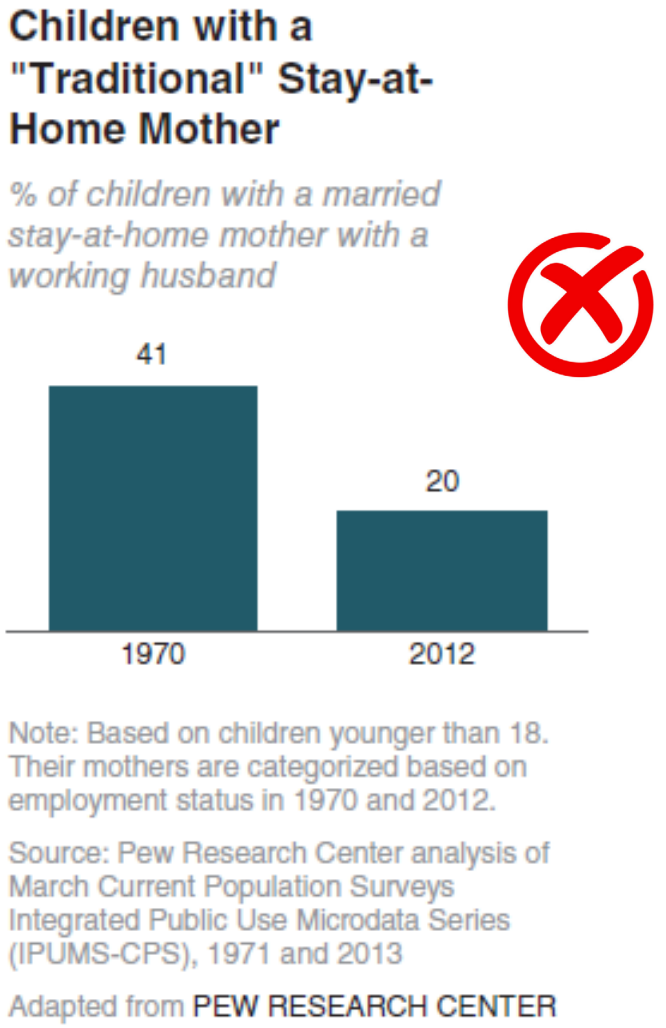

Slopegraph


Waterfall


Square area

# Simple text

- A great way to communicate if your have just a number or two to share
- In comparison of two numbers, you can highlight one




# Tables

- Tables interact with our verbal system
  - We read them
- Tables are great if
  - You need to communicate multiple different (units of) measures
  - Different audience members will look for different rows/columns
- Use light borders or simply white space to set apart elements of the table.

Heavy borders

| Group | Metric A | Metric B | Metric C |
|---|---|---|---|
| Group 1 | $X.X | Y% | Z,ZZZ |
| Group 2 | $X.X | Y% | Z,ZZZ |
| Group 3 | $X.X | Y% | Z,ZZZ |
| Group 4 | $X.X | Y% | Z,ZZZ |
| Group 5 | $X.X | Y% | Z,ZZZ |

Light borders

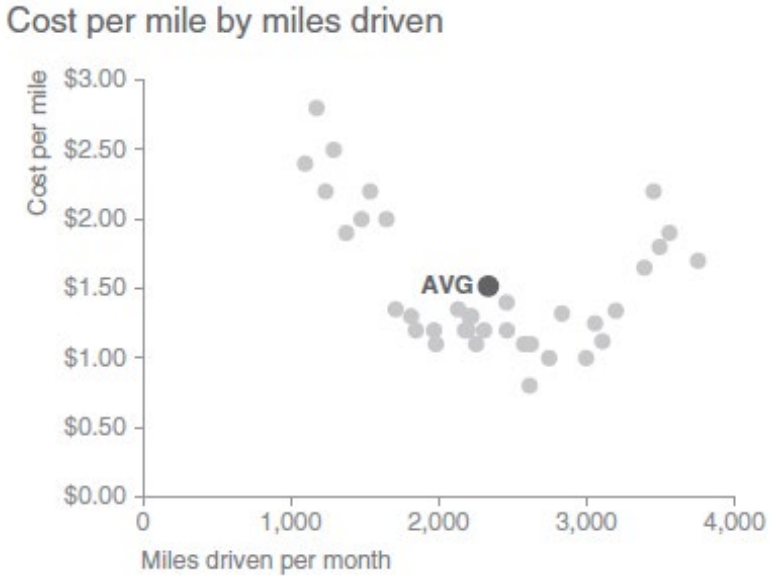
| Group | Metric A | Metric B | Metric C |
|---|---|---|---|
| Group 1 | $X.X | Y% | Z,ZZZ |
| Group 2 | $X.X | Y% | Z,ZZZ |
| Group 3 | $X.X | Y% | Z,ZZZ |
| Group 4 | $X.X | Y% | Z,ZZZ |
| Group 5 | $X.X | Y% | Z,ZZZ |

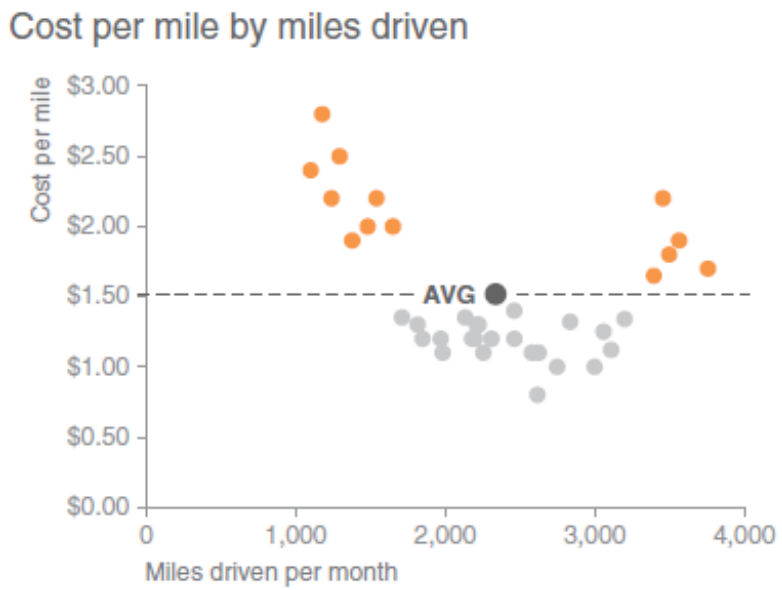
Minimal borders

| Group | Metric A | Metric B | Metric C |
|---|---|---|---|
| Group 1 | $X.X | Y% | Z,ZZZ |
| Group 2 | $X.X | Y% | Z,ZZZ |
| Group 3 | $X.X | Y% | Z,ZZZ |
| Group 4 | $X.X | Y% | Z,ZZZ |
| Group 5 | $X.X | Y% | Z,ZZZ |

# Graphs

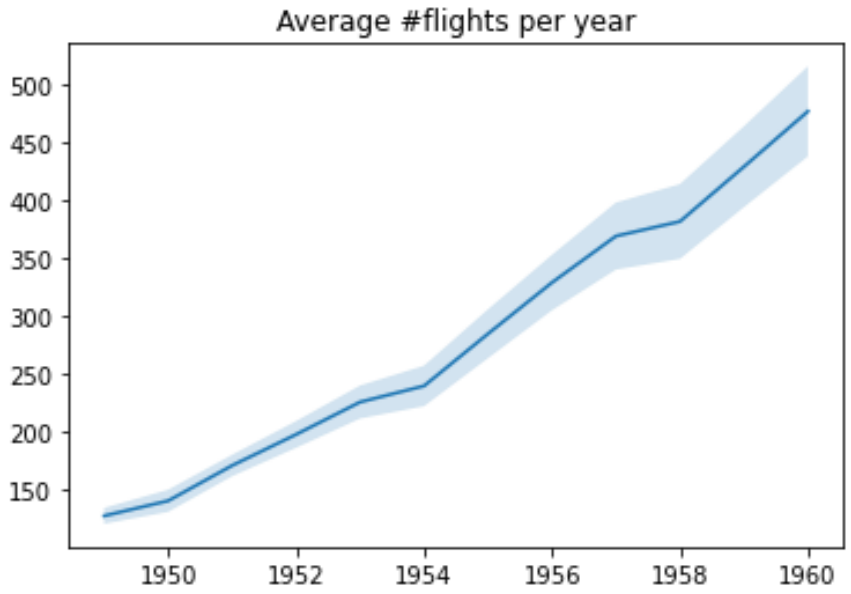
- Graphs interact with our visual system, which is faster at processing information.
  - Points
  - Lines
  - Bars
  - Area
- Scatterplot
  - Useful for showing the relationship between two numeric variables



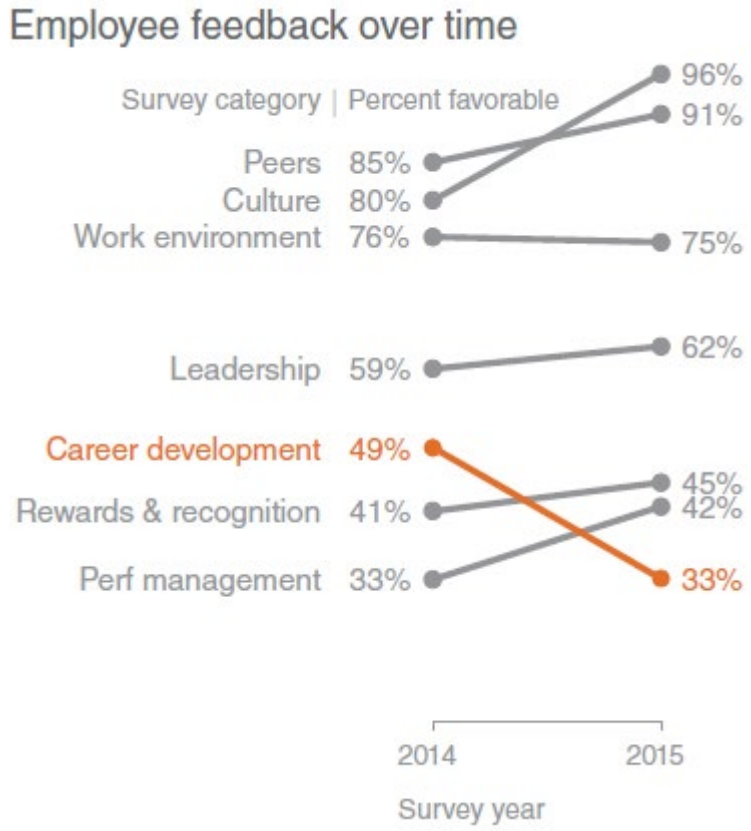
# Line Graph

- Good for a number of series
- A range can be used to illustrate the confidence or range of a variable



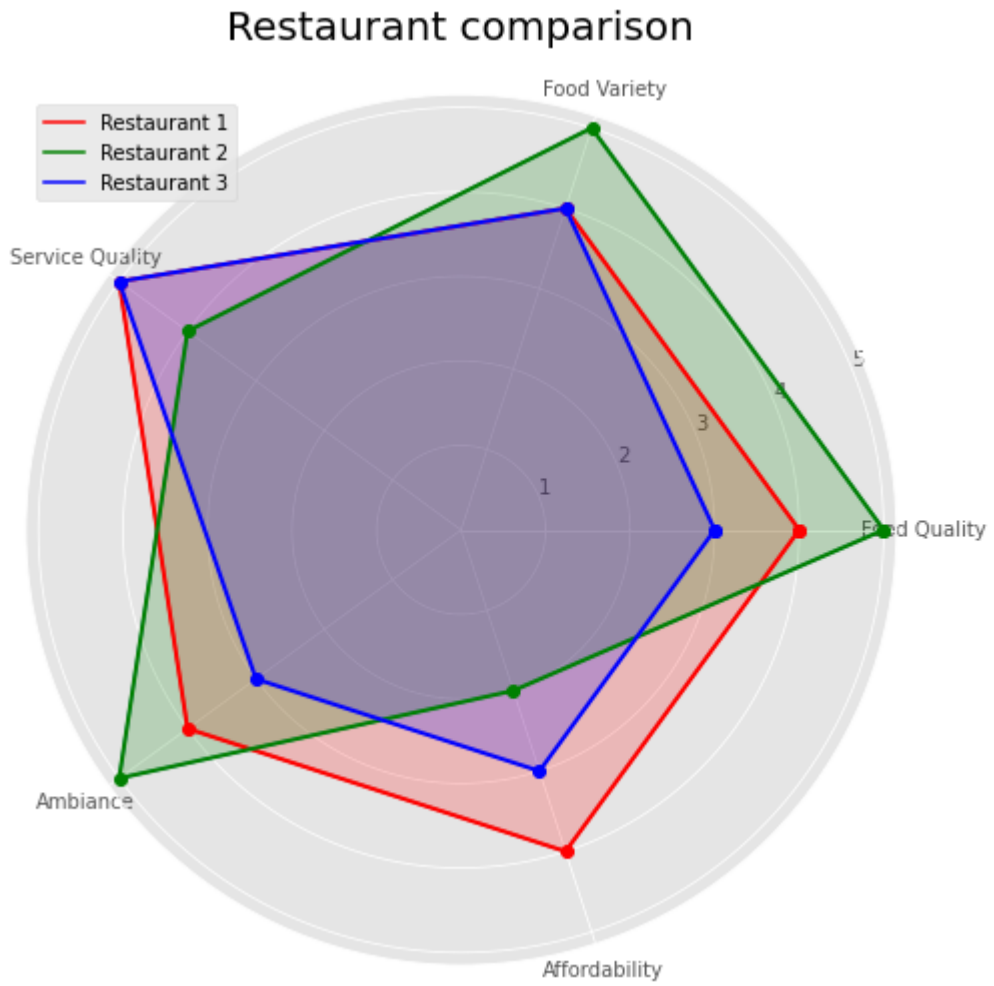

# Slope Graph

- It’s useful when you
  - have two time periods or points of comparison, and
  - want to quickly show relative increases and decreases, or differences across various categories, between the two parties.
- However,
  - no library for that: You need to code yourself
    - Sample code available
  - it can be ugly if there are may lines across each other.



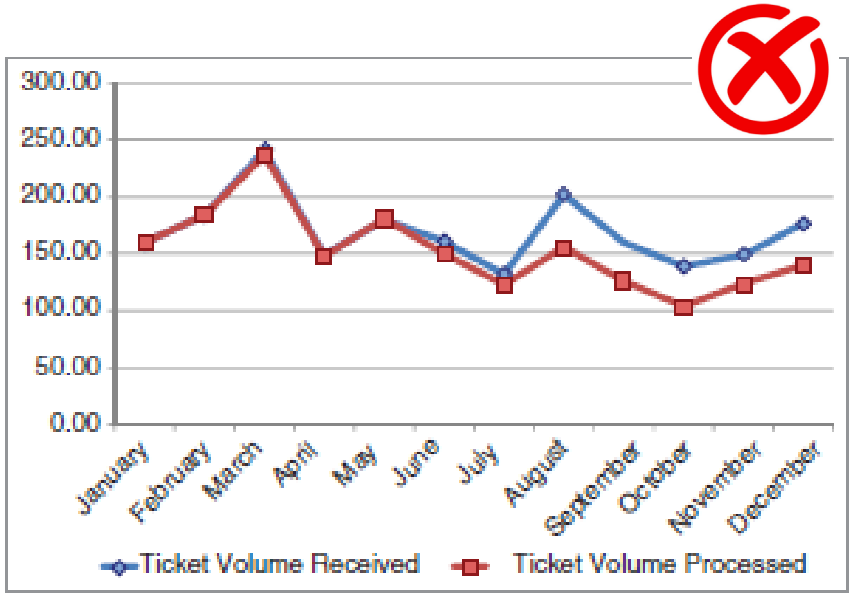
# Radar Charts

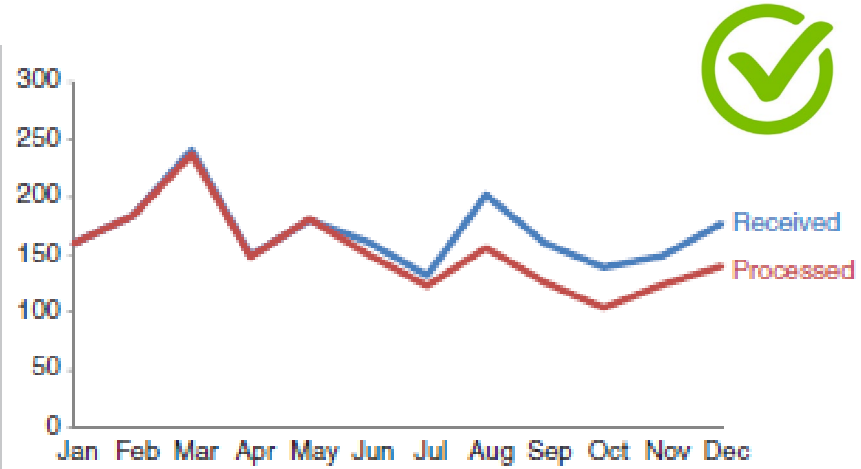
- Useful when you need to compare data points with multiple attributes.
- Dominating or dominated points are visible easily.
  - In this example, Restaurant 3 is dominated by Restaurant 1
    - Restaurant 1 is better on three attributes, and the same on the other two



# 3. Eliminate clutter

- **Clutter**: Visual elements that take up space but don't increase understanding.
- Why should we eliminate a clutter?
  - It makes our visuals appear more complicated than necessary
  - It is also distractive



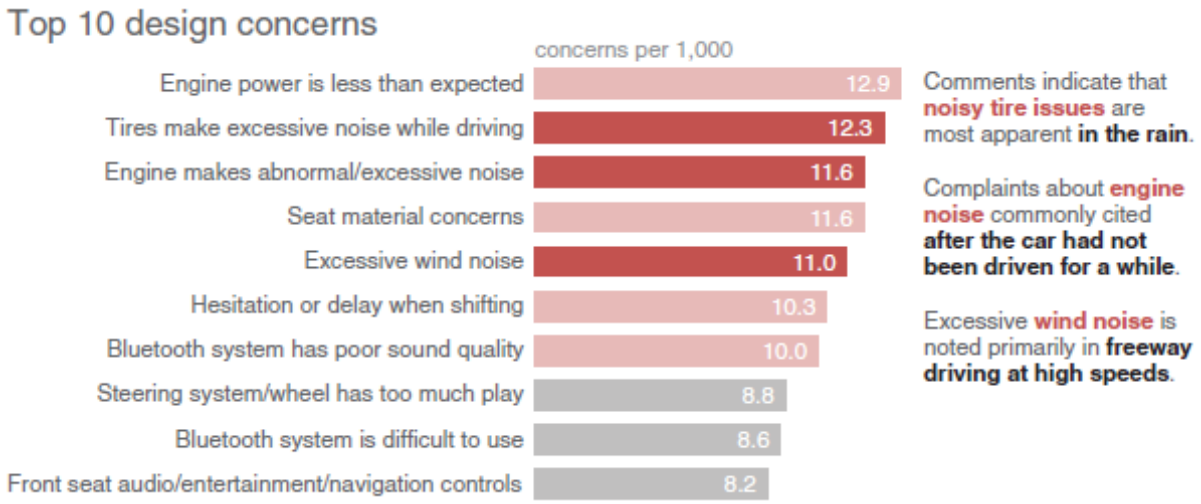
# 4. Focus attention where you want it

- **Preattentive attributes**: font, size, color, position, etc.
  - They can be used to help direct your audience's attention to what you want them to focus on.
  - They can be used to create a visual *hierarchy* of elements to lead your audience through the information you want to communicate in the way you want them to process it.

7563950684733
6586630375760
8603726586020
8465891078300

Color

What are we doing well? Great Products. These products are clearly the best in their class.
Replacement parts are shipped when needed. You sent me gaskets without me having to ask. Problems are resolved promptly. Bev in the billing office was quick to resolve a billing issue I had. General customer service exceeds expectations. The account manager even called to check in after normal business hours.
You have a great company – keep up the good work!



# 5. Think like a designer

- '*Form follows function*' holds in storytelling with data:
  - **Function**: What do we want our audience to be able to *do* with the data?
  - **Form**: A visualization that will allow for this with ease.
- Traditional design concepts that can be applied here:
  - Affordance
    - Highlight the important stuff
    - Eliminate distractions
    - Create a clear visual hierarchy of information
  - Accessibility
    - Don't overcomplicate
    - Thoughtful use of text
  - Aesthetics
    - Make it pretty: color, alignment, white space
  - Acceptance
    - Articulation of the benefits, side-by-side comparison, multiple options for input

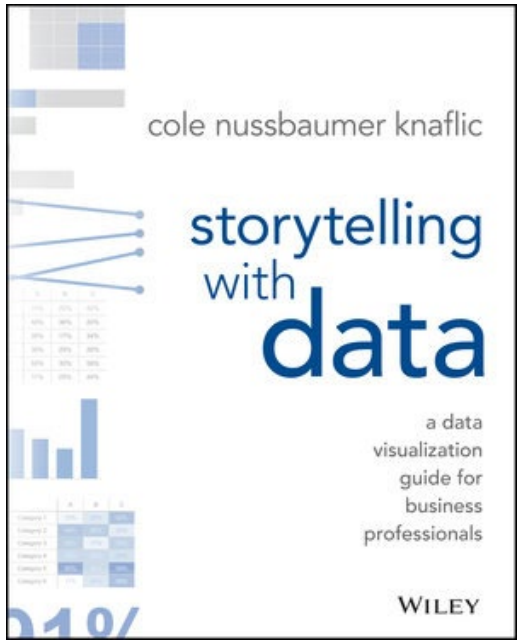
# 6. Tell a story

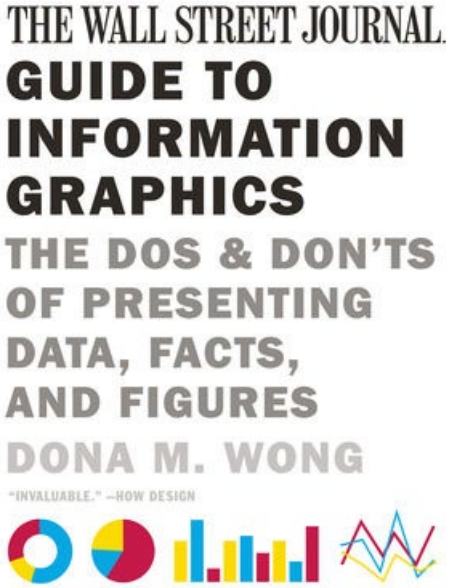
- Story structure of 3 parts

- The beginning
  - What is the context for your audience to understand your data?
- The middle
  - What does the data show and what is interesting about it?
- The end
  - What do you want your audience to do after understanding the data?
  - A clear “call to action”.

# References

- Dona M. Wong: The Wall Street Journal Guide to Information Graphics: The Dos and Don’ts of Presenting Data, Facts, and Figures. Wiley, 2013.
- Cole Nussbaumer Knaflic: Storytelling with data. Wiley, 2015.
  - Python + matplotlib **sample code**: https://github.com/empathy87/storytelling-with-data
- https://matplotlib.org/3.5.0/gallery/index.html

# Example in Jupyter Notebook

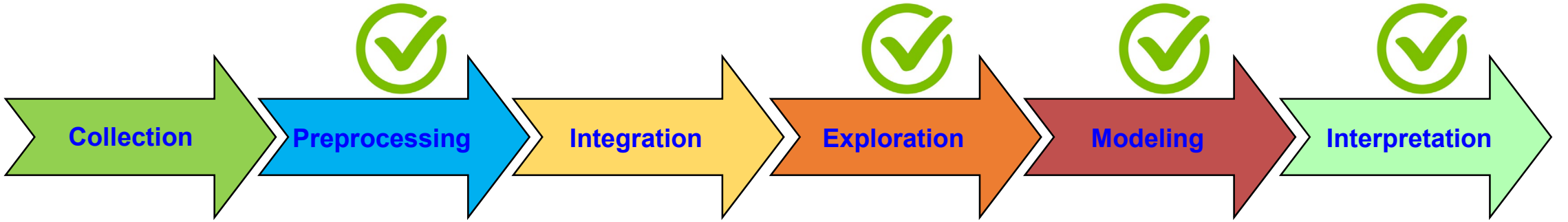
- Lecture10_AdvVisualization.ipynb
  - Line charts with ranges
  - Radar charts
  - Pies
  - Scatterplot with varied point size

# Agenda

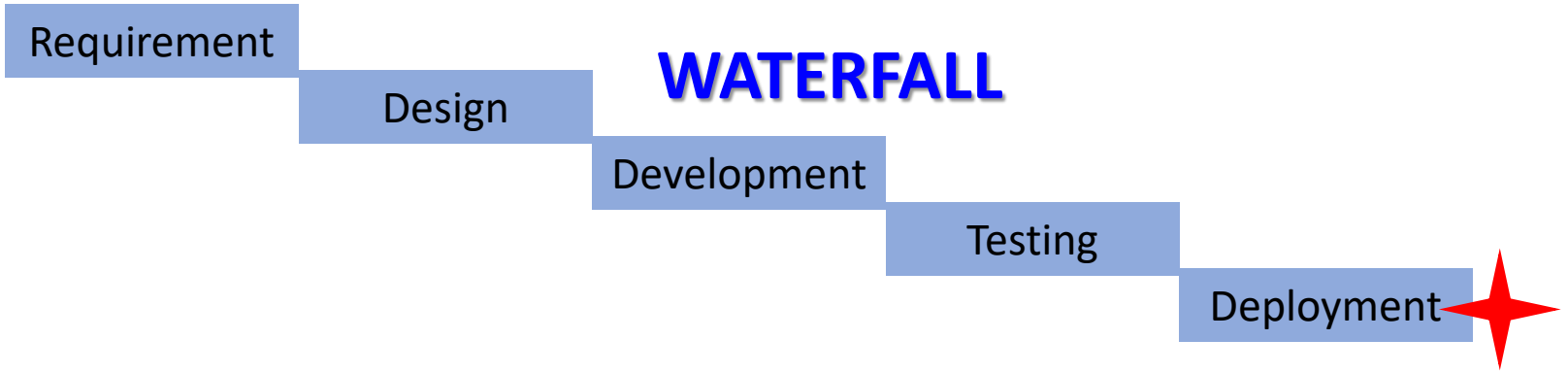
- Storytelling with data
- MLOps
- Finale of the course

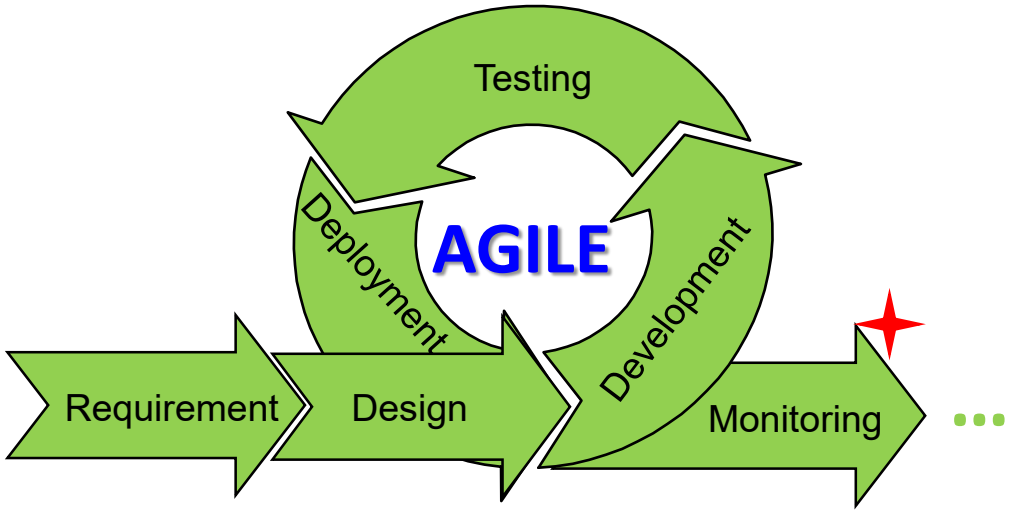
# Data Science Process

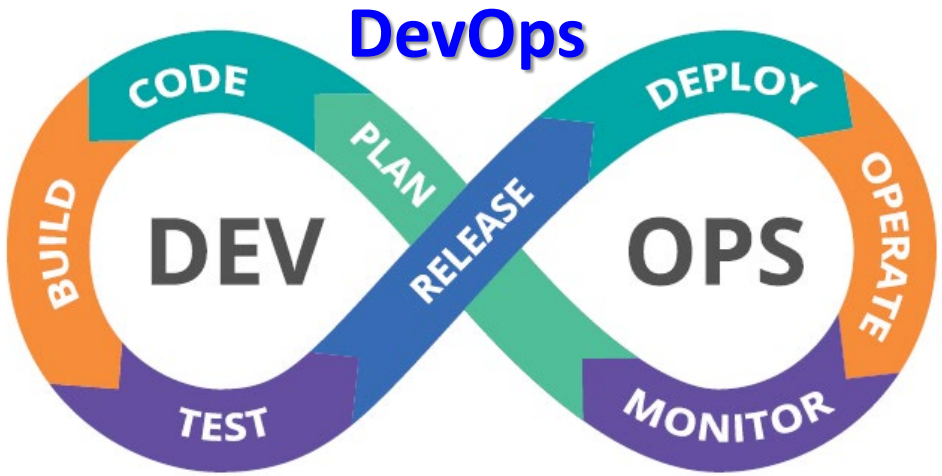
- We only cover some of them in this course
- In practice, a pipeline is needed for the whole process
  - A software based system



# Software Engineering Approaches

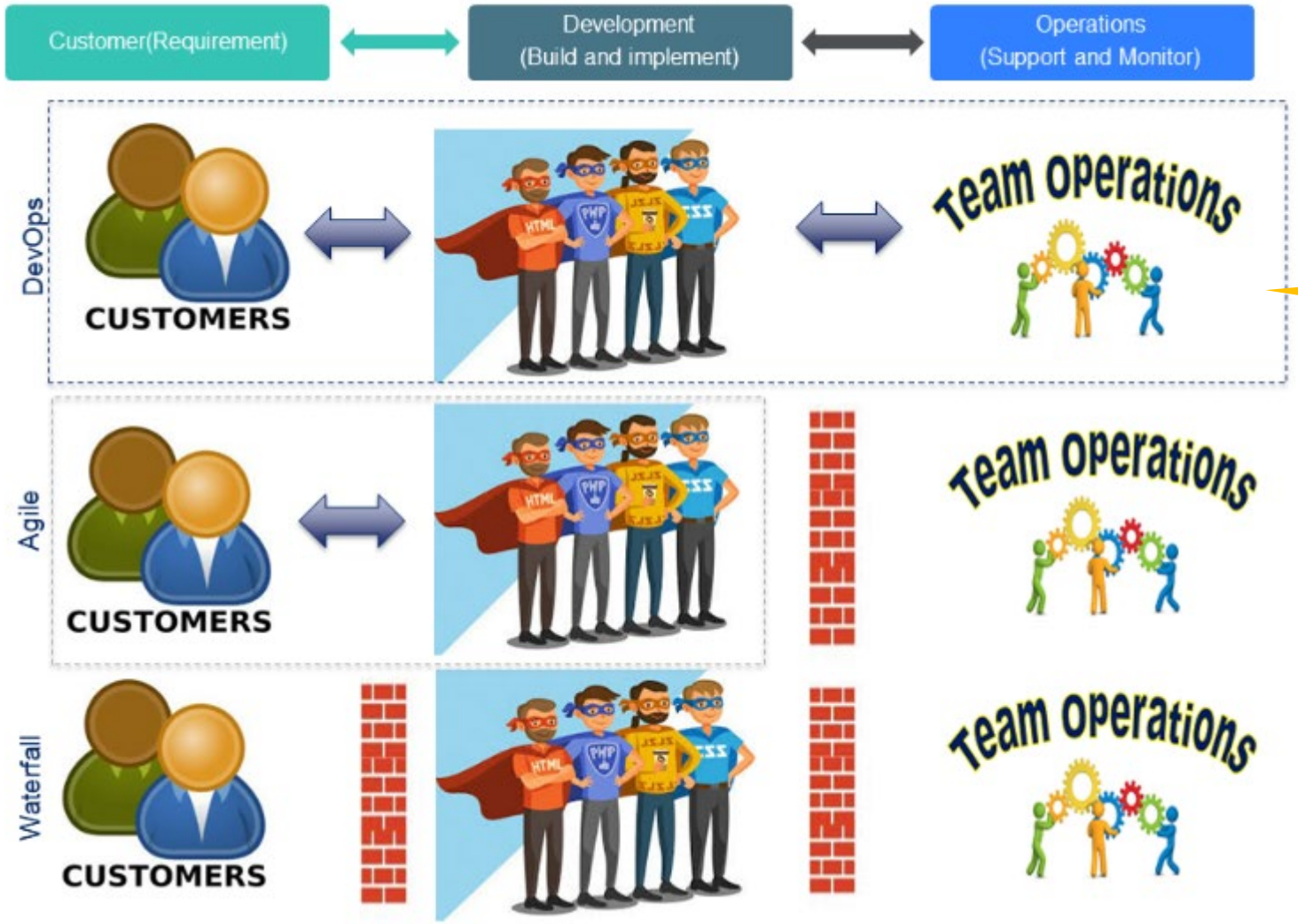



https://software.af.mil/training/devops/

# Team Dynamics

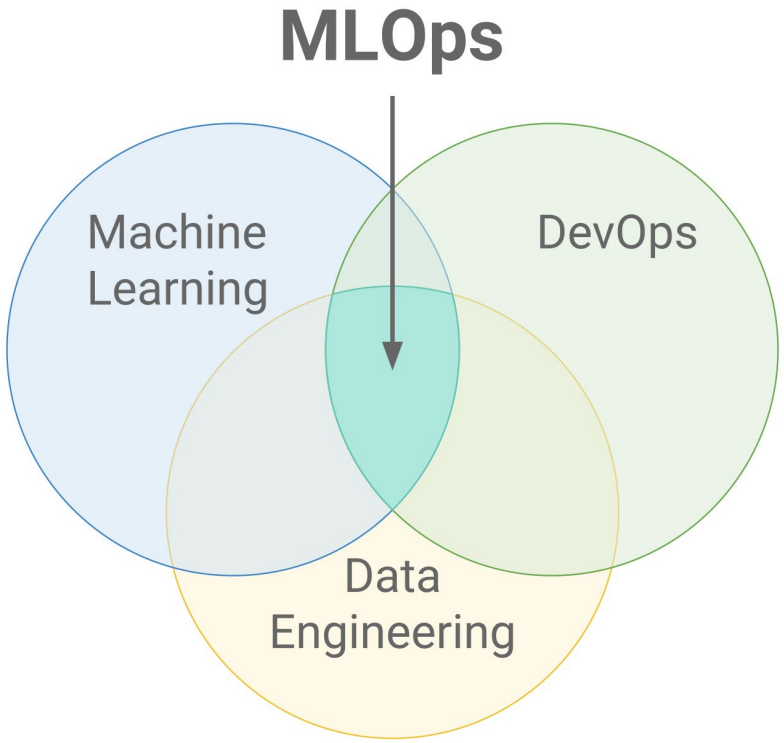



Data Science in industry or production line

https://www.linkedin.com/pulse/waterfallagiledevops-cloud-nitin-srivastava

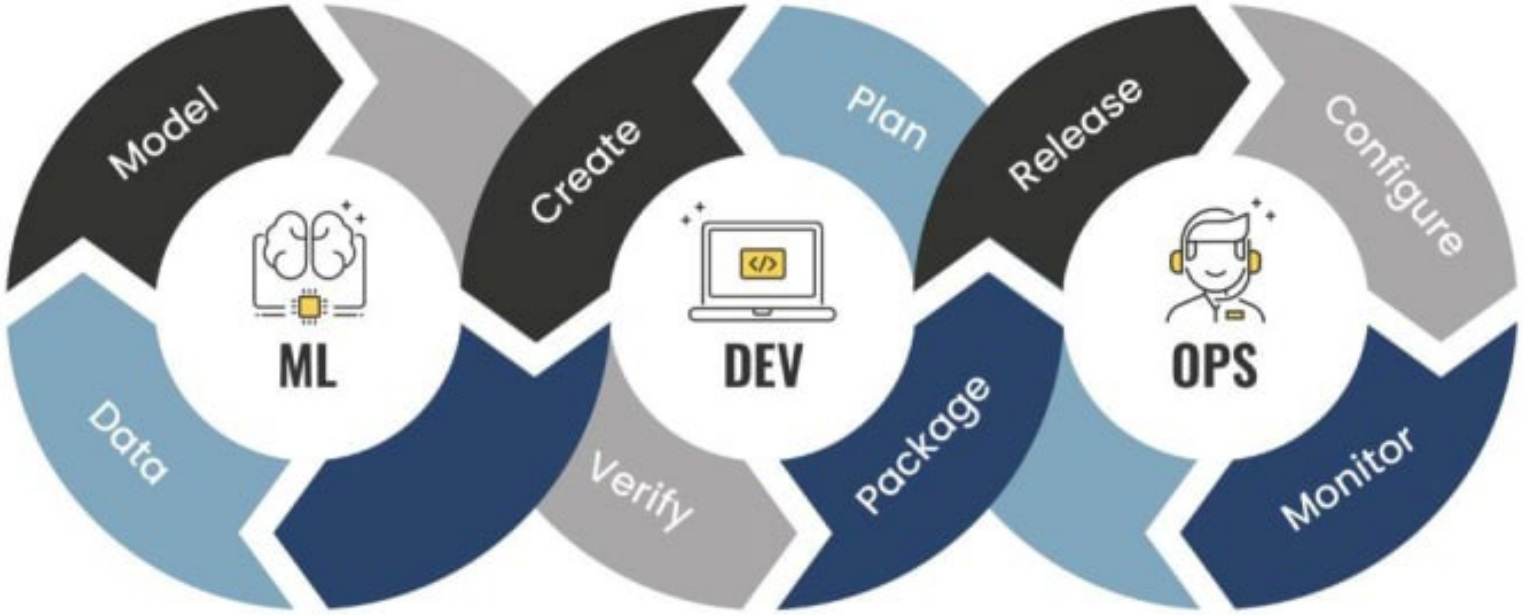
# MLOps

- DevOps = Development + Operations
- **MLOps**: a paradigm that aims to deploy and maintain ML models in production *reliably* and *efficiently*.
- MLOps = ML + DevOps
  - Production models (through lifecycles)
    - Automation
    - Quality
  - Business and regulatory requirements



# MLOps: 3-Cycle View

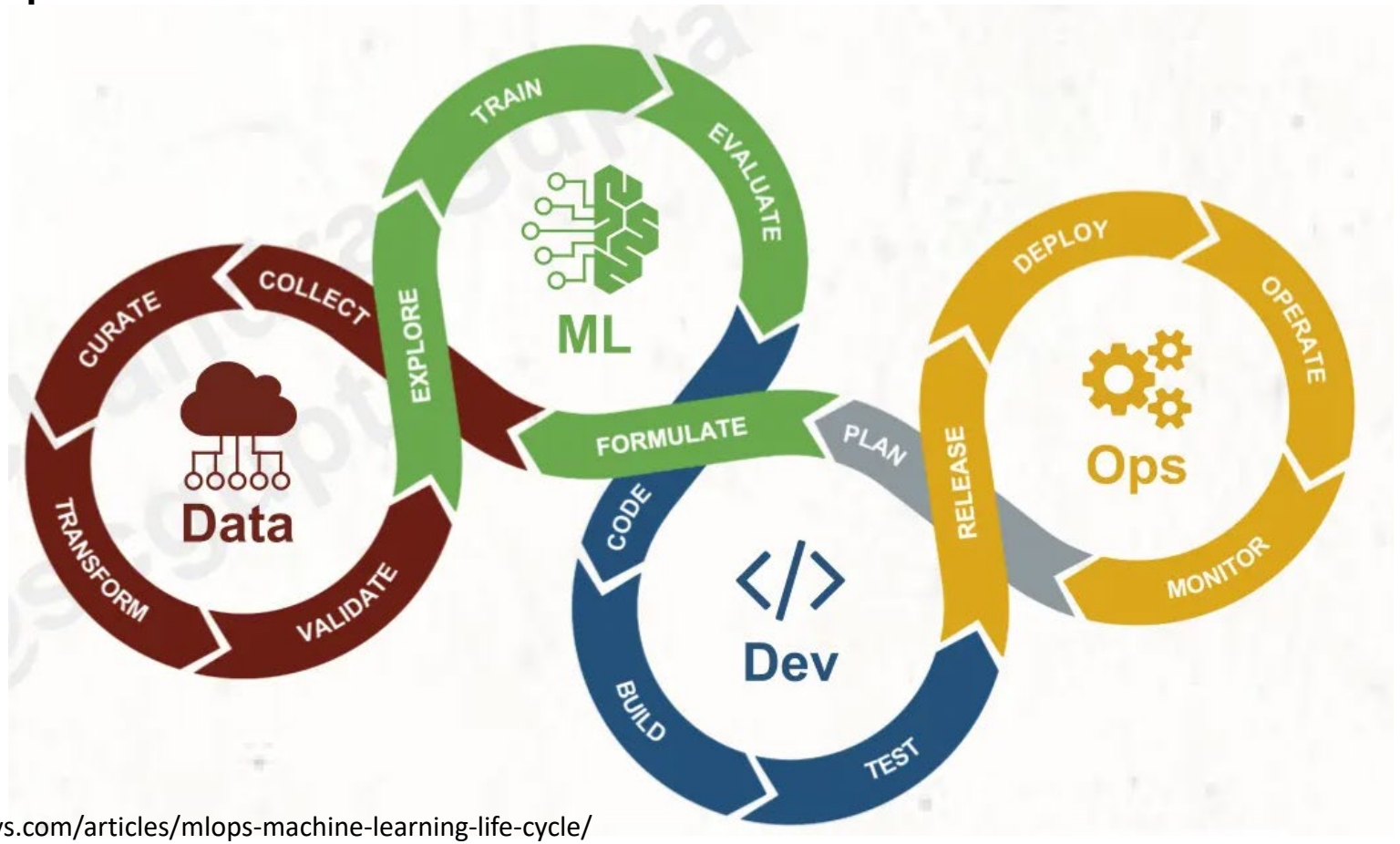
- Lifecycle management




https://ubuntu.com/blog/what-is-mlops

# MLOps: 4-Cycle View

- Data is emphasized for ‘Data Science’




https://www.ml4devs.com/articles/mlops-machine-learning-life-cycle/

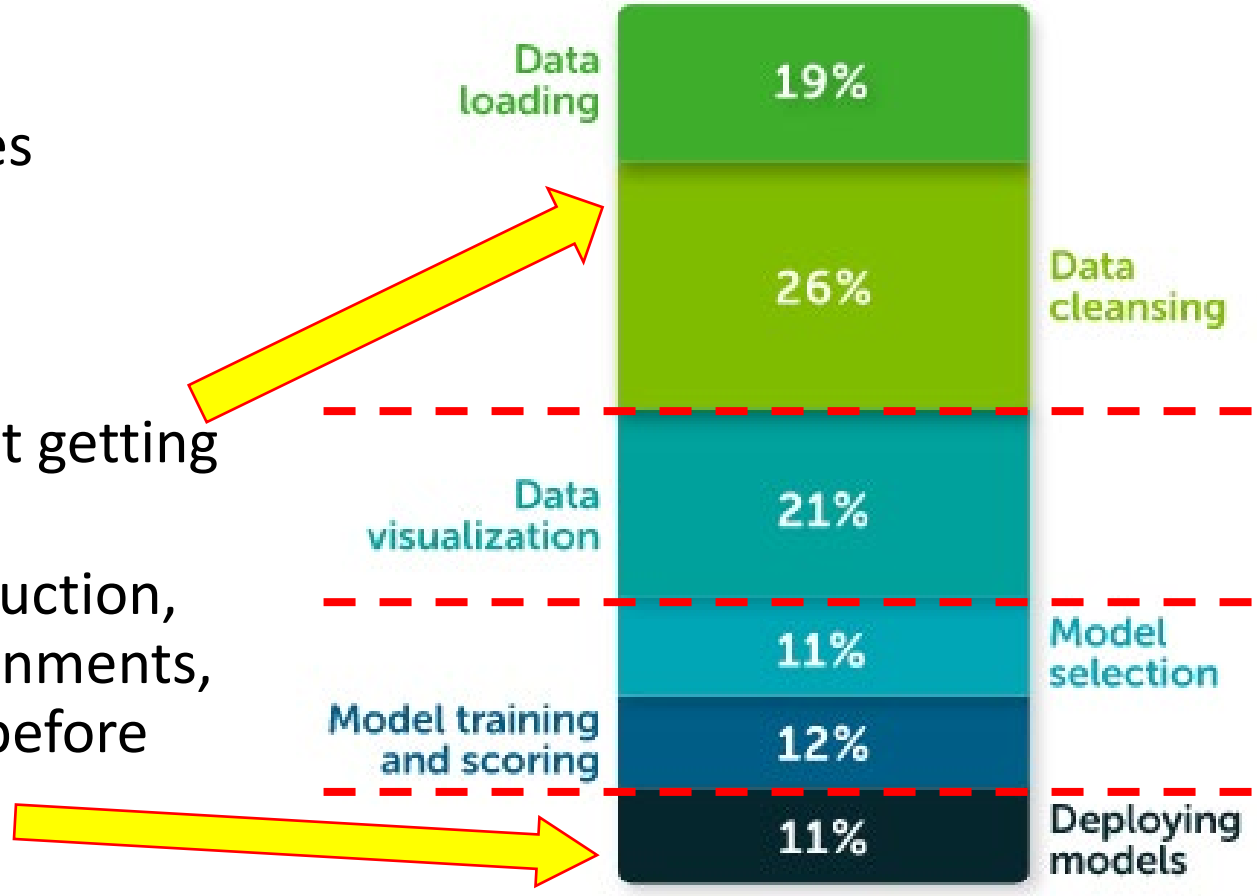
# Readings on MLOps

- https://ubuntu.com/blog/what-is-mlops
- https://blogs.nvidia.com/blog/2020/09/03/what-is-mlops/
- https://www.ml4devs.com/articles/mlops-machine-learning-life-cycle/

# Efficiency Gap

- An international survey
  - 2,360 responses from 100+ countries

- Time spending
  - On average, **45%** of the time is spent getting data ready (loading and cleansing).
  - Once the models are ready for production, they contend with numerous environments, dependencies, and even skill gaps, before the models see the light of day.




https://www.anaconda.com/state-of-data-science-2020

# Agenda

- Storytelling with data
- MLOps
- Finale of the course
  - Review
  - Mini-project
  - Exam
  - Evaluation

# Course Content

| | Title | Topics |
|---|---|---|
| 1 | Data science and data | Data science, data science process, data types, Jupyter Notebook |
| 2 | Exploratory data analysis | Series, DataFrame, missing value handling |
| 3 | Visualization | Histogram, box plot, bar chart, line chart, scatter plot, pairplot, correlation heatmaps |
| 4 | Classification I | Supervised vs. unsupervised learning, data scaling<br>classification problem and general steps, decision tree, random forest, KNN,<br>classification result evaluation, model evaluation, cross-validation, ROC and AUC |
| 5 | Classification II | |
| 6 | Regression | Regression problem and evaluation, linear regression, polynomial regression,<br>decision tree regression, logistic regression |
| 7 | Clustering I | Clustering problem, k-means, hierarchical clustering, DBSCAN,<br>clustering result evaluation<br>One-hot-encoding, feature engineering |
| 8 | Clustering II | |
| 9 | Association rules | Association rule definition, support, confidence, lift, frequent itemsets, Apriori |
| 10 | Data science in practice | Storytelling with data, MLOps |

# From Another Perspective

- Data Science process (Lecture 1)
- Jupyter Notebook (Lecture 1)
- Preprocessing
  - Missing data handling (Lecture 2)
  - Data scaling (Lecture 5)
  - One-hot-encoding (Lecture 8)
- Data modelling
  - Feature engineering (Lecture 8)
  - Classification (Lectures 4 & 5)
  - Regression (Lecture 6)
  - Clustering (Lectures 7 & 8)
  - Association rules (Lecture 9)
- Visualization (Lectures 3, 10)

# PYTHON FOR DATA SCIENCE CHEAT SHEET

## Python Scikit-Learn

## Introduction

**Scikit-learn:**"sklearn"is a machine learning library for the Python programming language. Simple and efficient tool for data mining, Data analysis and Machine Learning.

Importing Convention - import sklearn

## Preprocessing

### Data Loading

- Using NumPy:

```
>>>import numpy as np
>>>a=np.array([(1,2,3,4),(7,8,9,10)],dtype=int)
>>>data = np.loadtxt('file_name.csv',
   delimiter=',')
```

- Using Pandas:

```
>>>import pandas as pd
>>>df=pd.read_csv('file_name.csv',header=0)
```

### Train-Test Data

```
>>>from sklearn.model_selection
import train_test_split

>>> X_train, X_test, y_train, y_test =
train_test_split(X,y,random_state=0)
```

### Data Preparation

- Standardization

```
>>>from sklearn.preprocessing import
StandardScaler
>>>get_names = df.columns
>>>scaler =
preprocessing.StandardScaler()
>>>scaled_df = scaler.fit_transform(df)
>>>scaled_df =
pd.DataFrame(scaled_df,
columns=get_names)m
```

- Normalization

```
>>>from sklearn.preprocessing import
   Normalizer
>>>pd.read_csv("File_name.csv")
 >>>x_array = np.array(df['Column1'])
#Normalize  Column1
 >>>normalized_X =
preprocessing.normalize([x_array])
```

## Working On Model

### Model Choosing

**Supervised Learning Estimator:**

- Linear Regression:

```
>>> from sklearn.linear_model import
LinearRegression
>>> new_lr=
LinearRegression(normalize=True)
```

- Support Vector Machine:

```
>>> from sklearn.svm import SVC
>>> new_svc = SVC(kernel='linear')
```

- Naive Bayes:

```
>>> from sklearn.naive_bayes import
GaussianNB
>>> new_gnb = GaussianNB()
```

- KNN:

```
>>> from sklearn import neighbors
>>>
knn=neighbors.KNeighborsClassifier(n_ne
ighbors=1)
```

**Unsupervised Learning Estimator:**

- Principal Component Analysis (PCA):

```
>>> from sklearn.decomposition import
PCA
>>> new_pca= PCA(n_components=0.95)
```

- K Means:

```
>>> from sklearn.cluster import KMeans
>>> k_means = KMeans(n_clusters=5,
random_state=0)
```

### Train-Test Data

**Supervised:**

```
>>>new_ lr.fit(X, y)
>>> knn.fit(X_train, y_train)
>>>new_svc.fit(X_train, y_train)
```

**Unsupervised :**

```
>>> k_means.fit(X_train)
>>> pca_model_fit =
new_pca.fit_transform(X_train)
```

## Post-Processing

### Prediction

**Supervised:**

```
>>> y_predict =
new_svc.predict(np.random.random((3,5)))
>>> y_predict = new_lr.predict(X_test)
>>> y_predict = knn.predict_proba(X_test)
```

**Unsupervised:**

```
>>> y_pred = k_means.predict(X_test)
```

### Model Tuning

**Grid Search:**

```
>>> from sklearn.grid_search import GridSearchCV
>>> params = {"n_neighbors": np.arange(1,3), "metric":
          ["euclidean", "cityblock"]}
>>> grid = GridSearchCV(estimator=knn,
param_grid=params)
>>> grid.fit(X_train, y_train)
>>> print(grid.best_score_)
>>> print(grid.best_estimator_.n_neighbors)
```

**Randomized Parameter Optimization:**

```
>>> from sklearn.grid_search import RandomizedSearchCV
>>> params = {"n_neighbors": range(1,5), "weights":
["uniform", "distance"]}
>>> rsearch = RandomizedSearchCV(estimator=knn,
param_distributions=params, cv=4, n_iter=8, random_state=5)
>>> rsearch.fit(X_train, y_train)
>>> print(rsearch.best_score_)
```

### Evaluate Performance

**Classification:**

1. Confusion Matrix:

```
>>> from sklearn.metrics import
      confusion_matrix
>>> print(confusion_matrix(y_test,
      y_pred))
```

2. Accuracy Score:

```
>>> knn.score(X_test, y_test)
>>> from sklearn.metrics import
      accuracy_score
>>> accuracy_score(y_test, y_pred)
```

**Regression:**

1. Mean Absolute Error:

```
>>> from sklearn.metrics import mean_absolute_error

>>> y_true = [3, -0.5, 2]
>>> mean_absolute_error(y_true, y_predict)
```

2. Mean Squared Error:

```
>>> from sklearn.metrics import mean_squared_error
>>> mean_squared_error(y_test, y_predict)
```

3. $R^2$ Score :

```
>>> from sklearn.metrics import r2_score
>>> r2_score(y_true, y_predict)
```

**Clustering:**

1. Homogeneity:

```
>>> from sklearn.metrics import
homogeneity_score
>>> homogeneity_score(y_true,
y_predict)
```

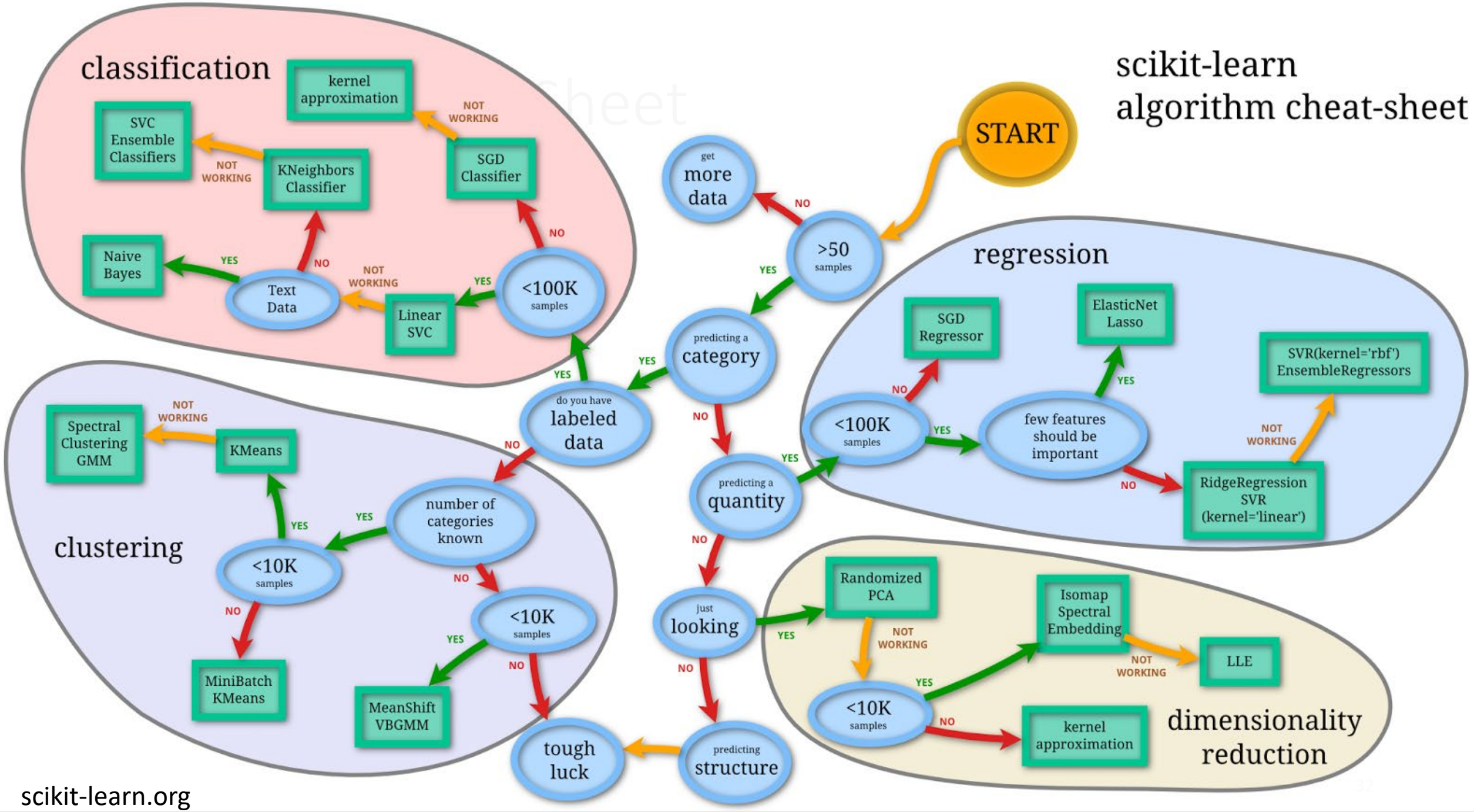
2. V-measure:

```
>>> from sklearn.metrics import
v_measure_score
>>> metrics.v_measure_score(y_true,
y_predict)
```

**Cross-validation:**

```
>>> from
sklearn.cross_validation
import cross_val_score
>>>
print(cross_val_score(knn,
X_train, y_train, cv=4))
>>>
print(cross_val_score(new_
lr, X, y, cv=2))
```

**FURTHERMORE:**
Python for Data Science Certification Training Course




scikit-learn.org

# What we didn’t cover in this course?

- More ML models
  - Bayes
  - SVM
  - Neural networks and deep models
  - …
- Data Science Ethics
  - Data Ownership
  - Privacy and Anonymity
  - Data Validity
  - Algorithmic Fairness
  - Societal Consequences
  - Code of Ethics

**https://www.coursera.org/learn/data-science-ethics**

# What's next?

- Mini-projects
  - Report (code and data) deadline: **23:59 May 2, 2023** (to Digital Exam)
    - Each group should make only one submission
  - Formatting: Use the template provided in Moodle!
  - What is the current status of your mini-projects?

- Exams
  - June 6-7, 2023
  - Reexam: August 07, 2023

# Code and Data for Mini-Projects

- You must make sure that the examiner and censor can run your code with the data you use.
- Code
  - Upload your Jupyter Notebook (using the provided template) to Digital Exam
    - Group based, included in a group's submission
  - Remember to use clear Markdowns and comments in your notebook
- Data
  - Option 1: Upload it together with your notebook to Digital Exam.
    - Still group based
  - Option 2: Provide a URL in your notebook to upload and make sure the URL works properly.

# Exam Format

- Oral, 20 minutes in total for each student.
- Internal censor (Jialiang and Masoumeh)
- It will start with a short presentation of your mini-project. (**~5m**)
  - Highlight the most important things
  - Powerpoint or the like is recommended but *not* mandatory
- Then, it'll be a dialog (Q&A) between the examiner/censor and you. (**~10m**)
- After that, the examiner and censor will decide the grade without your presence. (**~2.5m**)
- Finally, you will receive your grade and feedback. (**~2.5m**)

- **NB**: We may refer to your mini-project report (incl. code and data) during the *whole* period of the exam

# Exam questions

- Introduction, data science in general
- Data preprocessing
- Supervised vs. unsupervised learning
- Classification
- Regression
- Clustering
- Association rule mining

Refer to the document in Moodle
- Totally 30 questions
- You don’t have to grasp all of them.
- But if you aim at 10 or 12, you’d better know most of them well, as questions will be asked *randomly*.

# Tips for exam preparation

- Study all the slides and read the mandatory materials
  - Make sure you understand all concepts and methods
- Do all the exercises
  - Make sure you're able to apply all techniques
  - All solutions are in Moodle
- Go through the exam question list
  - Make sure you're able to talk about each topic
- Finish your mini-project and submit your notebook on time
  - Make sure you get your hands dirty and understand every part of the Jupyter Notebook

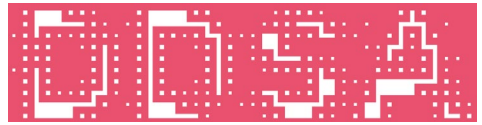
- **NB**: The final grading will take into account
  - Mini-project (notebook)
  - Your short presentation in the exam
  - Question answering in the exam

# This morning until 12:00

- **Evaluation**
  - Refer to the email sent to you from study administration
  - ~10 minutes needed
- Consultancy (until 12:00)
  - Exercises
  - Mini-project
  - Exam
  - Course in general
  - Any other relevant issues
- You’re welcome to contact me, Jialiang or Mousemeh before the exam.

# Danish Data Science Academy

- Fellowships (annually)
  - PhD
  - Postdoc
- Other funding
  - Course and event
  - Travel and visit
- Mentoring Programme
  - If you work on data science and need mentoring from a senior person
  - https://ddsa.dk/events/ddsa-mentoring-programme/
- Pre-Graduate Retreat
  - Information about PhD programs for bachelor and master students



ddsa.dk