Features & Visualisations

Machine Learning

UNIVERSITY OF SAN FRANCISCO
CHANGE THE WORLD FROM HERE

What is Machine Learning?

- Use *features* (X) to describe a dataset of *instances*
  - Instances can be things, events, categories, etc.
  - Features have many names, including (predictor) variables, stimulus, etc.
- Unsupervised Learning
  - Principal Components Analysis = describe variance in dataset
  - Clustering = group / segment data instances
- Supervised Learning
  - One part of data is a target (Y)
  - We expect that Y is correlated to each X by some function
  - Tasks:
    - We need to choose our features (X) so they correlated with Y
    - We need to choose the function on each X that minimises the error in predicting Y

Machine learning process

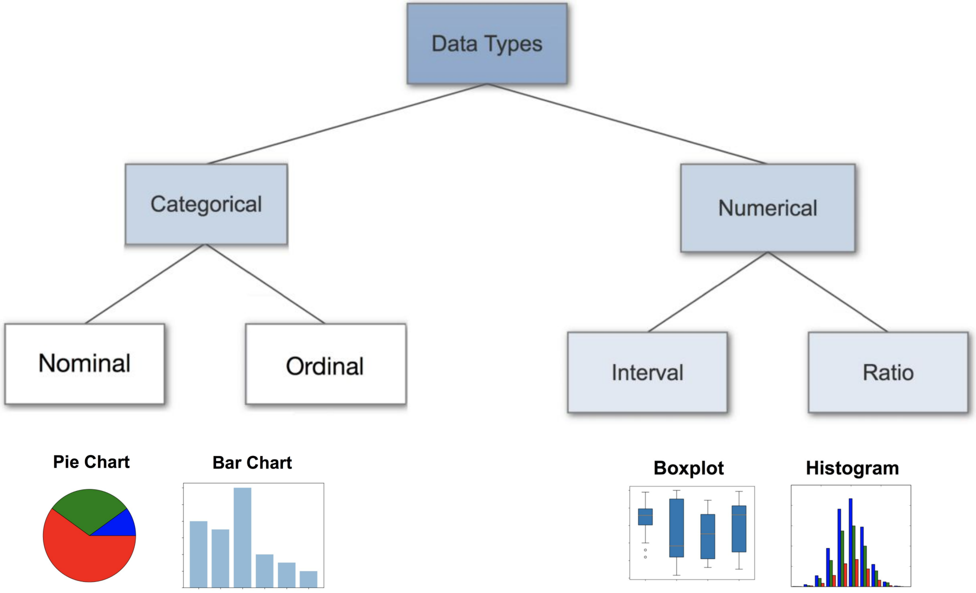
- Goal
    - Unsupervised: describe X
    - Supervised: use X to predict Y
- Process
    - Determine the problem (classification?) and choose one or more performance metrics (accuracy?)
    - Collect data?
    - Visualise the data — identify types of data and plot them, clean outliners, etc.
    - Transform data / engineer features?
    - Implement and tune model(s); compare efficacy of each model using chosen metric(s)... or vs. baseline

ML in production: expectation

1. Collect data
2. Train model
3. Deploy model
4. 

https://twitter.com/chipro/status/1348265019012743169

Types of data



Has / Can	*Nominal*	*Ordinal*	*Interval*	*Ratio*
Count / Frequency / Proportion	✔	✔	✔	✔
Mode, Median		✔	✔	✔
Order of values is known		✔	✔	✔
Mean, stdev of values			✔	✔
Add / subtract values			✔	✔
Multiply / divide values				✔
True zero				✔

https://towardsdatascience.com/data-types-in-statistics-347e152e8bee

Feature Engineering

- The art of creating good feature variables (X) from raw data
- Good feature set
  - Describes or represents the target well with fewest values
  - Leads to good models (better results)
  - Requires less complex algorithms
- Requires a lot of domain knowledge?
  - Listen to Subject Matter Experts (SMEs), knowing:
  - SMEs are often overestimate how predictive some data is
  - SMEs often miss some important predictors (or combinations of predictors)
  - SMEs often lack one or more data transformation steps

(One) Feature Engineering Process

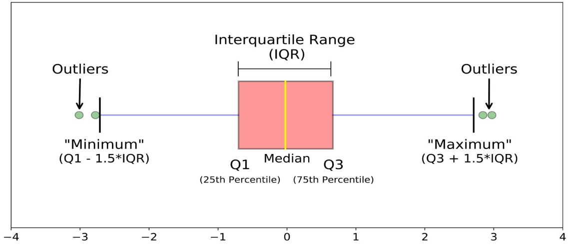
- Get SME input
- Brainstorm features
  - Look at what other people have done*
  - Encourage wild ideas (especially working in teams) — and get a lot of ideas
- Decide which one(s) to use in the model
  - Judge effort vs. expected return in power of model
  - Go for novelty so that the model becomes more powerful
- Implement the features above
- Study the impact of the implemented features
- Repeat, repeat, repeat

Algorithms vs. outliers

Class	Algorithm	Sensitive to outliers?
Unsupervised	K-Means	Sensitive
	Hierarchical Clustering	Sensitive
	PCA	Sensitive
Regression	Linear Regression	Sensitive
Classification	Logistic Regression	Sensitive
	K-Nearest Neighbours	Not sensitive
	Naive Bayes, SVM	Not sensitive
	Decision Trees, Random Forest, Boosted Trees	Not sensitive
	Neural Networks	Sensitive

Data —> Features (1)

- Imputation
  - Numerical: default value (eg. 0), mean, median, etc.
  - Categorical: most frequent value
- Outlier detection
  - Standard deviation is often used for detecting outliers
  - Common to project data to normal distribution and flagging values +/-3x std. Dev
- Binning
  - Combining range of numerical values or multiple categorical values
  - Use carefully since it loses (important?) information
- Mathematical transformation
  - *Log transform* can make small differences more significant
  - Sine / cosine to cyclical data (eg. dates) can show that Jan & Dec are closer than May & Jul



https://www.simplypsychology.org/boxplots.html

Canada	—> North America
Iran	—> Asia
Thailand	—> Asia
USA	—> North America

Data —> Features (2)

- One-hot encoding
  - Encoded categorical data may look related — eg. 1 = Alabama, 2 = Alaska
  - 1-hot encoding transforms data to multiple unrelated binary columns
- Splitting
  - Taking a string (eg. date: 2021-01-24) and extracting important parts
  - Also typical for NLP
- Lookup / External
  - Dates —> holidays, weekday, etc.
  - External sources are often useful
- Feature combinations
  - Add, subtract, multiply or divide two features to form a third
  - Remove features which are not useful

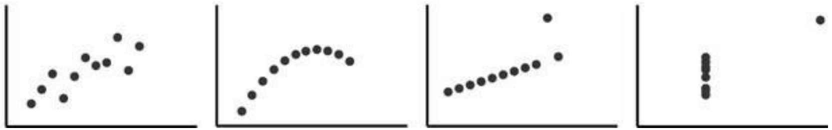
Why use visualisations?

- Difficult to see data patterns, trends, etc. in “wall of numbers” charts
- We can sometimes determine functions relating X to Y
- We can see outliers, possible errors in data, etc.

a

I		II		III		IV	
x	*y*	*x*	*y*	*x*	*y*	*x*	*y*
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.10	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.10	4	5.39	19	12.5
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89

b



https://towardsdatascience.com/9-data-visualization-tools-that-you-cannot-miss-in-2019-3ff23222a927

What do visualisations need?

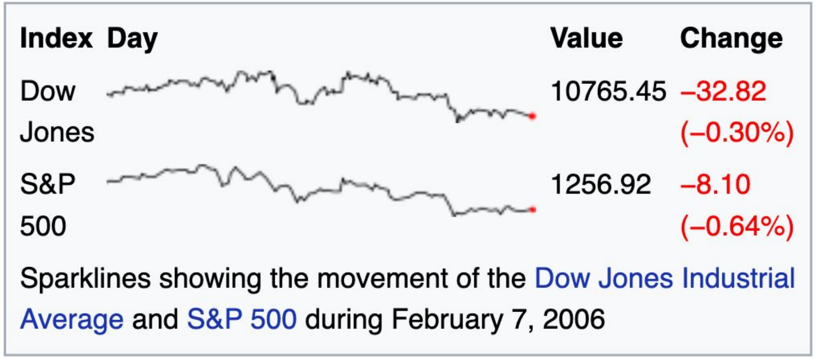
Title — Count of unique speakers outside the USA

Labels for each axis (or scale) — 12 speakers

Tufte

- Known for…
  - Data visualisation pioneer
  - “Formally documented” the sparkline chart
- Principles for graphic design
  - Graphical Integrity
    - Chart axes must be labelled
    - Scale must be consistent
  - Minimalism
    - Use as little “ink” to show the data
    - Use few (no?) graphical effects — instead, show how the data varies

Example sparklines in small multiple



Index	Day	Value	Change
Dow Jones		10765.45	–32.82 (–0.30%)
S&P 500		1256.92	–8.10 (–0.64%)

Sparklines showing the movement of the Dow Jones Industrial Average and S&P 500 during February 7, 2006

https://en.wikipedia.org/wiki/Sparkline

Why matplotlib / seaborn?

- matplotlib
  - Basic plotting library, default for python
  - Widely used
- seaborn
  - A layer on top of matplotlib
  - Provides additional functionality and improved aesthetics
- Not ggplot
  - A copy (port?) of package in R (ggplot2)
  - Not pythonic, therefore not intuitive for data people who are stronger in python
  - … but preferred by many for its OO-treatment of charts

Set up matplotlib & seaborn

- Basic syntax

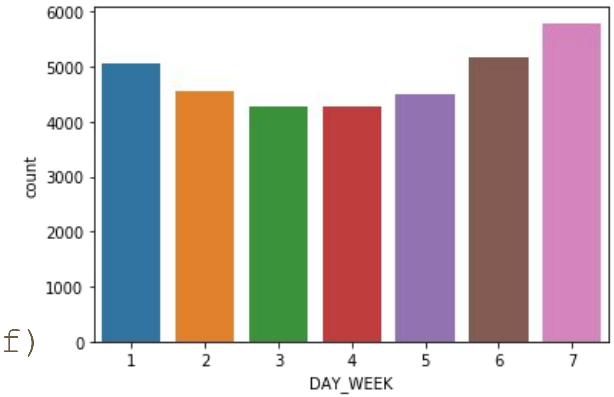
```
import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline
```

- Common plot types:
  - Bar plot (“countplot”)
  - Box plot
  - Line plot
  - Scatterplot

Example

- Countplot (bar plot)
    - x-axis = category labels
    - y-axis is frequency count of category
- Basic syntax

```
sns.countplot(x = df['LABEL'], data = df)
```



- Example
    - 2018 FARS ACCIDENT.CSV
    - Best day of week to be on the road is Wednesday?

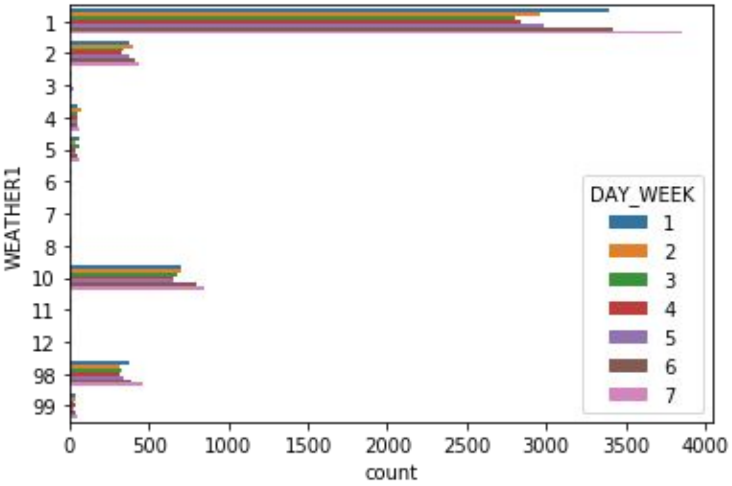
Some parameters to countplot

- Add additional dimensions (as “hue”)

```
sns.countplot(x = df['LABEL'], hue=df['WEATHER1'], data = df)
```

- Change orientation

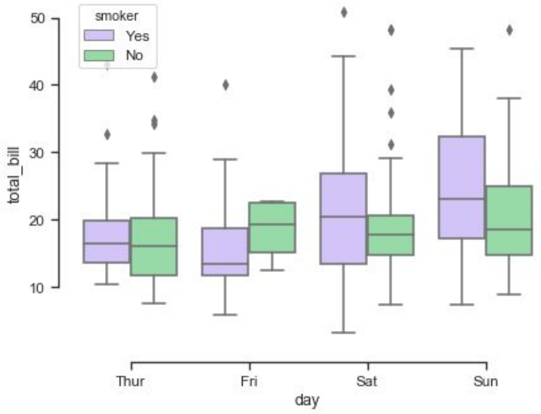
```
sns.countplot(y = df['LABEL'],\
  hue=df['WEATHER1'], data = df)
```



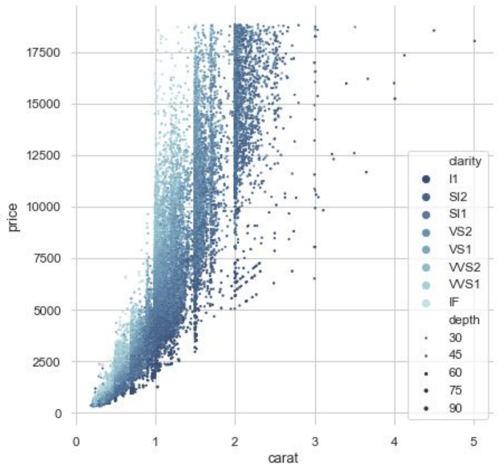
- Save

```
myplot = sns.countplot(...
myplot.get_figure().savefig('plot.pdf')
```

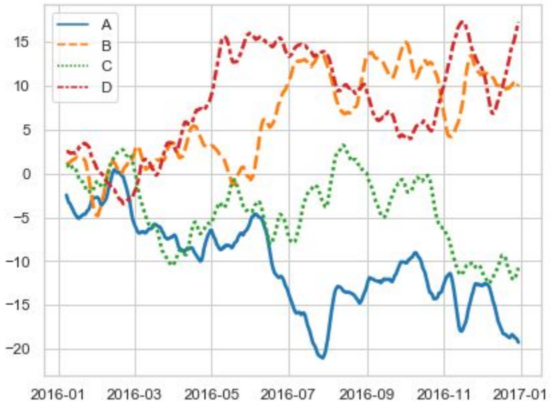
Other common plot types



Box Plot
(1-dimensional)
`sns.boxplot`



Scatter Plot
(2-dimensional)
`sns.scatterplot`



Line Plot
(2-dimensional)
`sns.lineplot`