ML
in Practice:

by Dane Brown

With great power comes great risk...

Tools

- Python OpenCV
- Scikit-learn
- Matplotlib
- Keras for ANN/deep learning...time permitting

Standard Coding Style in Scikit/OpenCV

m : Number of training examples

n : Number of features (Dimensionality of the input)

X : Input variables (features for training)

Y: Output variables (target features -- normally unseen data)

x : Labels corresponding to X

y: Labels corresponding to Y

y_pred (i.e. ŷ): The labels that the classifier “guessed”, given

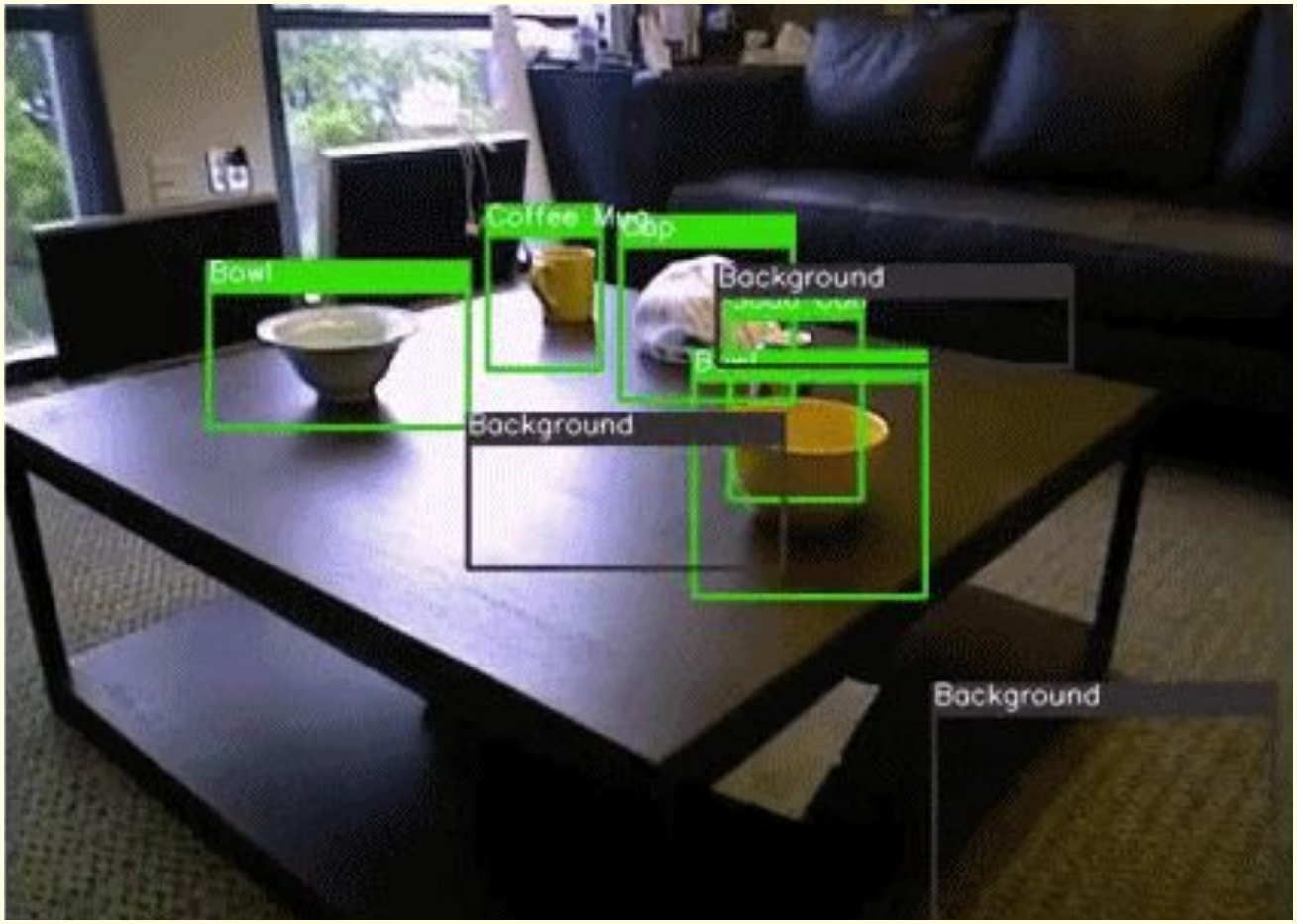
Remember

- Column-wise and Row-wise matrix operations
- In Matplotlib:
  - Column-wise: **axis=0** means aggregating row values
  - Row-wise **axis=1** means aggregating col values
- There are many other things to learn in Matplotlib
  - Read up after class

Tired of Maths/Stats Yet?

Object tracking and recognition using OpenCV+SVM

Let’s aim to get here by the end of the course



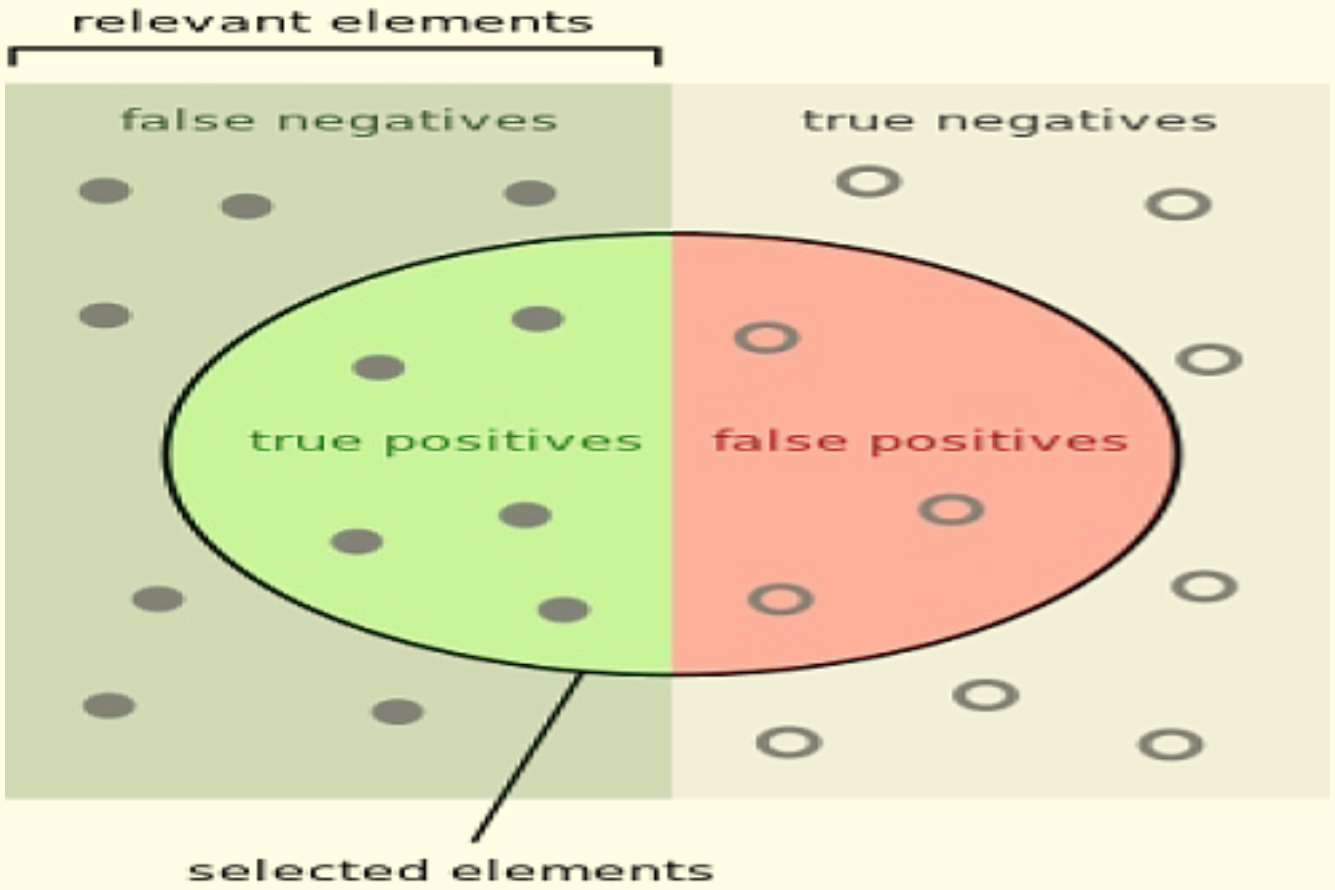
Objectives for this Week

- **Classification vs. regression**
  - which one to use and when?
- What is a k **-nearest neighbor (** k **-NN)** classifier
  - how to implement in OpenCV?
- Building a **linear regression** model + Lasso and ridge regression
- A **logistic regression** model for classification
  - why is it named so confusingly?
- Tips on data processing (time permitting)

cvML Methodology

- **Initialization:** Call the `cv` or `scikit` model by name to create an empty instance of the model
- **Set parameters:** can be default, e.g. k -NN: specify k for more than one neighbour
- **Train the model:** `train` or `fit` is used to fit the model to some data.
- **Predict new labels:** use `predict`, to guess the labels of new (**unseen**) data.
- **Score the model:** refer to slides 10+: works for both `cv` and `scikit`

Score the Model



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

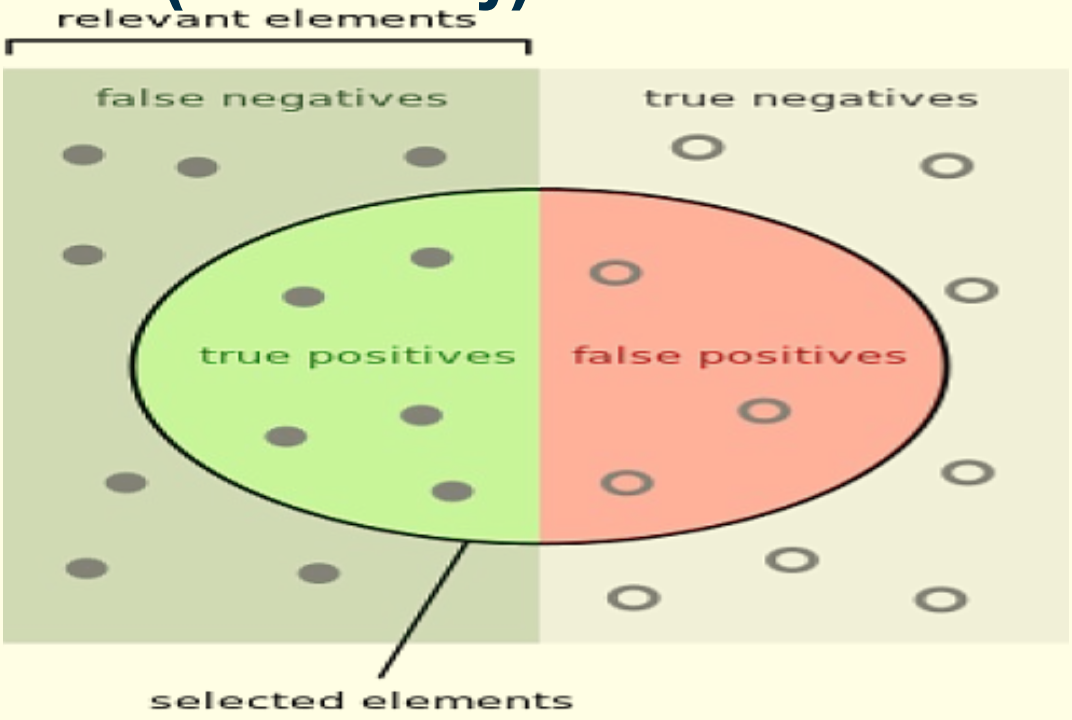
Scoring a Classification Model

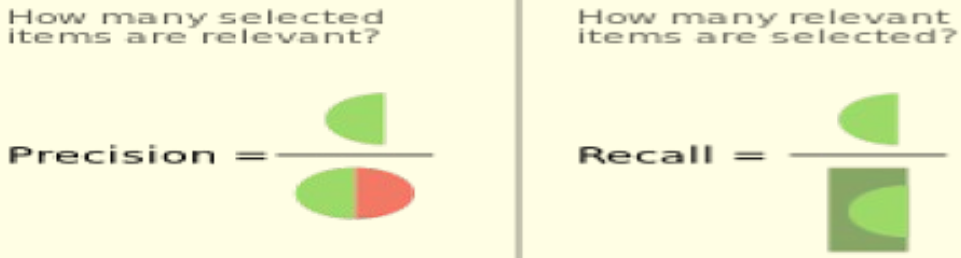
- **Consider cats as positive data points (== 1)**
- **Consider dogs as negative data points (== 0)**

- **Training labels:** the true or target value of a data point that the classifier aims to predict
- **Predicted labels:** the classifier predicts a data point
  - guesses the label of the class it thinks it belongs to

Scoring a Classification Model

- *accuracy_score*: Correctly predicted all data points
- *precision_score*: Not predicting a cat as a dog.
- *recall_score* **(sensitivity):** scores all cats





Scoring a Classification Model

- *accuracy_score*: Correctly predicted all data points
  - returns the fraction of pictures that have been correctly classified as containing a cat or a dog
  - accuracy = (tp + fp) / (tp + fp + tn + fn)

- *precision_score*: Not predicting a cat as a dog.
  - returns the fraction of pictures that actually contain a cat over the total number of (positive) predictions.
  - precision = tp / (tp + fp)

Check it out -> CV_ML

- *recall_score*: also called **sensitivity,** scores all the pictures that contain cat.
  - returns the fraction of pictures that have been correctly identified as pictures of cats over the total number of cats.
  - recall = tp / (tp + fn)

Scoring Regression:

by Dane Brown

“Life is ten percent what you experience and ninety percent how you respond to it.”

Scoring a Regression Model

- *mean_squared_error*: squared error between **predicted** and **true value** for every data point in the training set, averaged across all data points.

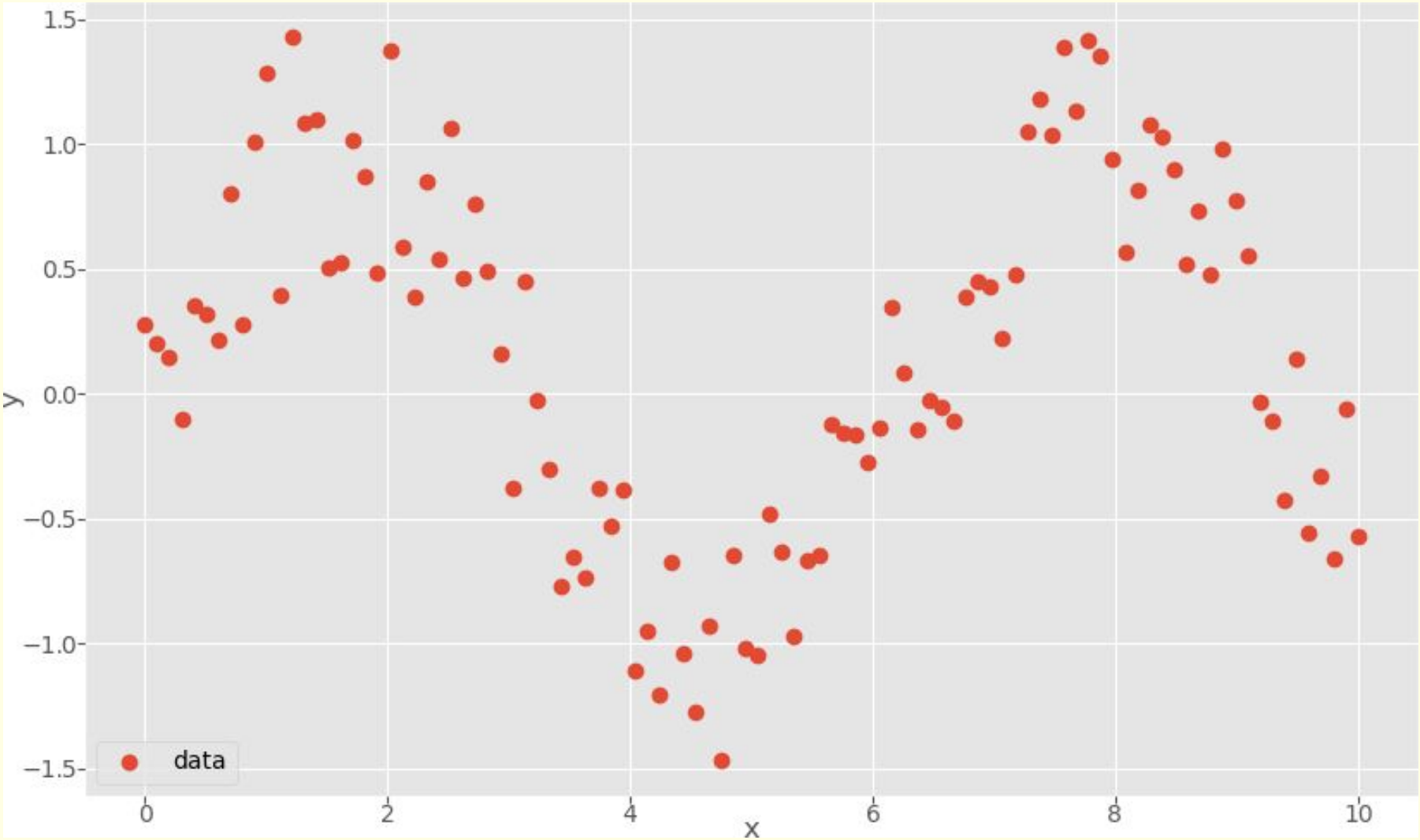
- *explained_variance_score*: the degree a model can explain the variation or dispersion of the test data. Measured using the correlation coefficient.

- *r2_score*: The R^2 score is closely related to the explained variance score, but uses an unbiased variance estimation. It is also known as the coefficient of determination.

Guess what this Data Represents?

Hint: Draw a “best fit”

Check it out -> CV_ML



Learn the basics of Matplotlib

I recommend getting clued up on matplotlib's basic functions.

https://pythonprogramming.net/matplotlib-python-3-basics-tutorial/