Introduction to Machine Learning in Python

G. Parkes, T. Rampat, S. Senior

Download VM at: http://www.southampton.ac.uk/~ngcmbits/virtualmachines/

Overview

1. Introduction to Machine Learning
2. Practical 1 – Decision Trees/K-Means
3. Regression, Preprocessing and Fitting
4. Practical 2 – Regression
5. Image Classification, Deep Learning
6. Practical 3 – TensorFlow

Assumptions

- Solid understanding of Python.
- No/Little understanding of Machine Learning.
- Exposure to Pandas and NumPy beneficial.
- Knowledge of the Jupyter Notebook and a Python IDE like Spyder will be beneficial.

Machine Learning – What is it?

- A subset of Computer Science and derived from Artificial Intelligence (AI)
- At it's heart a probability generator, with strong ties to statistics/mathematical optimisation.
- Improving the probability score taken as 'learning'. Uses this learnt data to 'predict' on new data.
- Examples of use: *Spam Filtering, Character Recognition, Computer Vision, Search Engines, Social Media*

Machine Learning – The Problem

```
def detect_edges('image'):
    # lots of code
def count_colour_pixels('image'):
    # lots of code
def guess_texture('image'):
    # lots of code
def define_fruit('image'):
    # lots of code
```

Machine Learning – The solution

$$f(\boldsymbol{X}) = \boldsymbol{y}$$

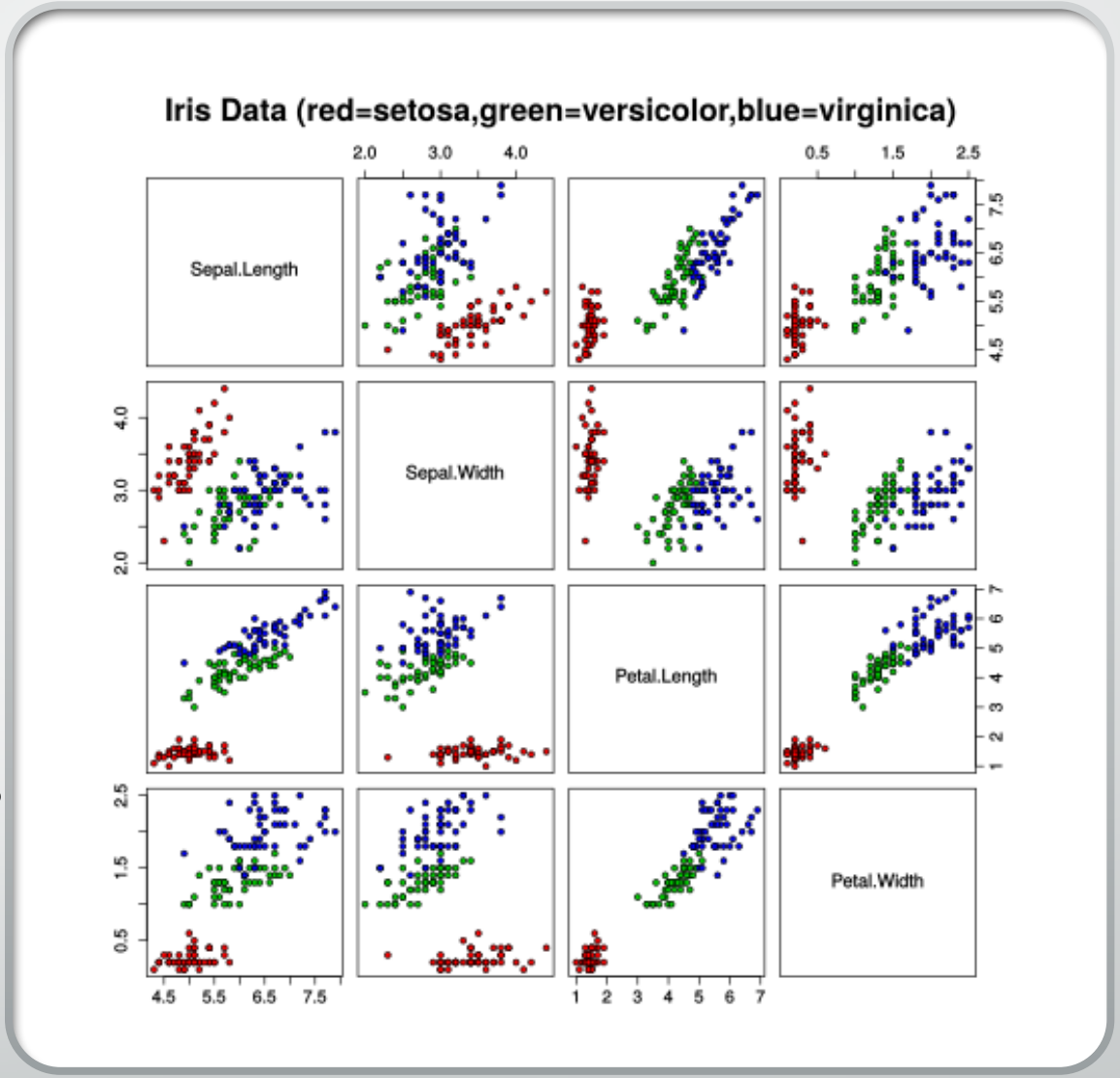
where ***X*** is a **matrix** of input, independent variables, and *y* is a **vector** of output, dependent variables *that classify* ***X***.

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
5.1	3.5	1.4	0.2	I. setosa
4.9	3.0	1.4	0.2	I. setosa
4.7	3.2	1.3	0.2	I. setosa
7.0	3.2	4.7	1.4	I. versicolor
6.4	3.2	4.5	1.5	I. versicolor
6.3	3.3	6.0	2.5	I. virginica

Taken from Fisher's *Iris* data set

Iris Dataset

- Each species is shaded a different colour
- Clear distinct patterns emerge
- ML trained to learn these patterns
- Applies it to new information when available

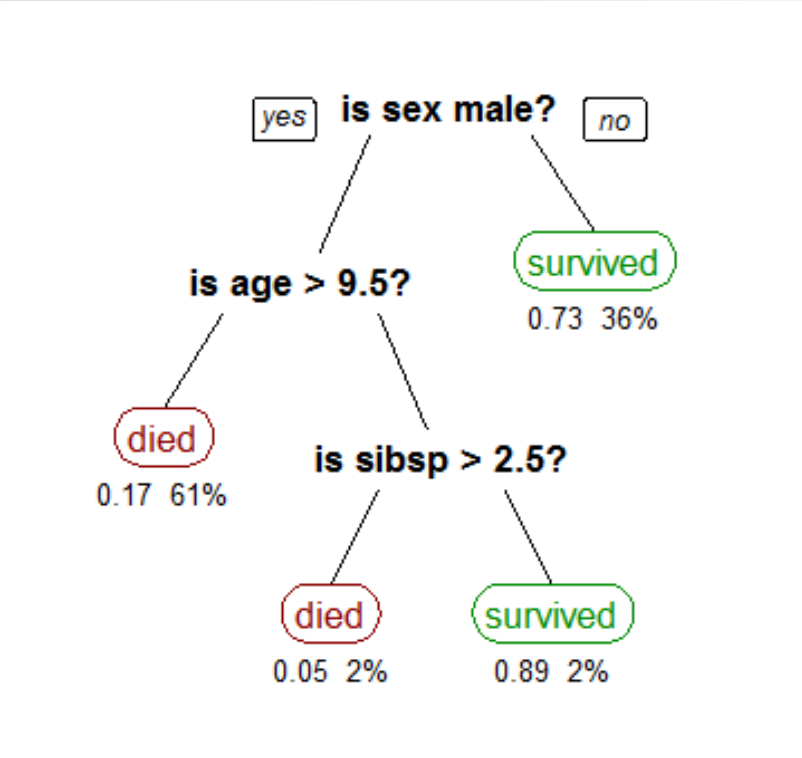


Machine Learning – Going slightly deeper

Three broad categories:

1. **Supervised Learning** – Presented with example inputs and corresponding outputs, given by a 'teacher'.
2. **Unsupervised Learning/Data Mining** – No labels given to inputs, computer infers features.
3. **Reinforcement Learning** – Interacts with a dynamic environment where it must perform a certain goal, feedback via rewards/punishments.

Decision Trees

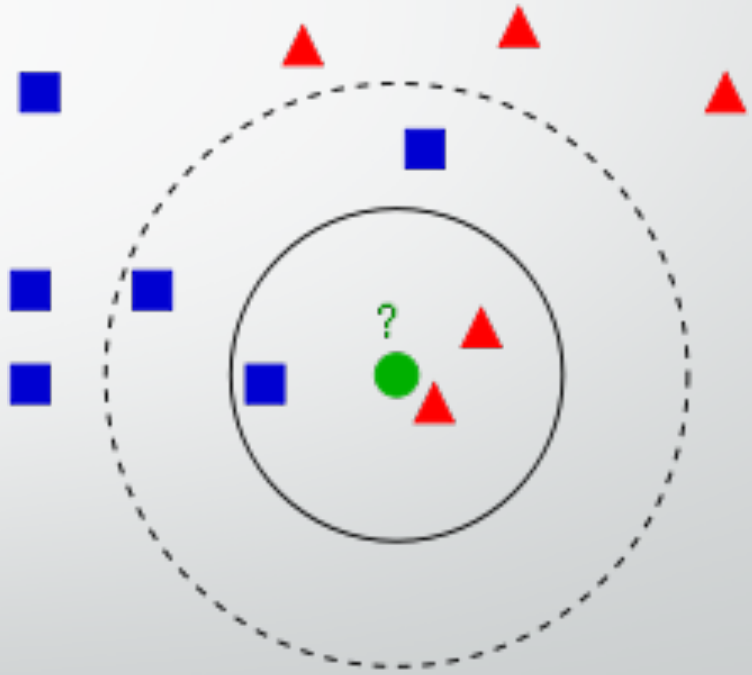


By Stephen Milborrow - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=14143467

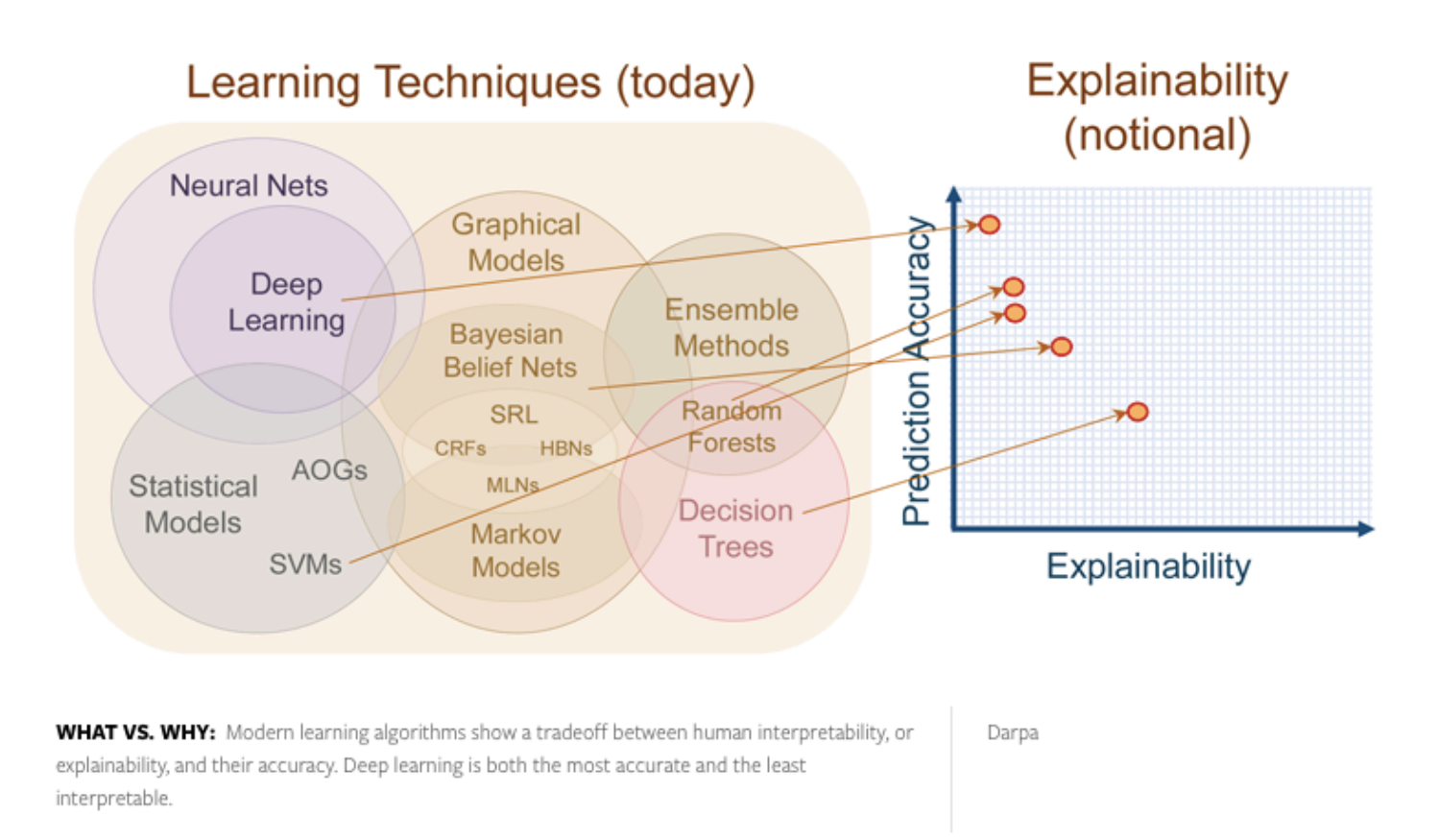
- Decision Trees convert the column data into a tree hierarchy where each leaf node will refer to one of the labels that can be identified with the feature data.
- Each branch tries to classify with a condition based on one of the feature columns.
- More columns/data leads to more complex tree structure.

K-Nearest Neighbours

- With Nearest Neighbour classifiers, we can use the Euclidean distance as a weight between points as a means of clustering features and identifying them.
- Not to be confused with *k-means clustering*; this is a classifier and not a clustering algorithm.
- The unidentified points are classified according to the label of the majority of nearest neighbours.



Understandability of Machine Learning



Taken from Nautil.us

Scikit-Learn

- A package in Python programming language. Part of the Anaconda Distribution.
- Built on NumPy, SciPy and MatPlotLib libraries.
- Open source, commercially usable.
- Simple to set up for most applications, very powerful functionality.
- Highly accessible.

Scikit-Learn

Name of Method	Example use	Algorithm
Classification	Spam Detection, Image Recognition	SVM, random forest, nearest neighbours
Regression	Drug response, stock prices	SVR, Ridge regression, LASSO
Clustering	Customer segmentation, grouping experiment outcomes	K-Means, Spectral Clustering, mean-shift
Dimensionality Reduction	Visualization, increased efficiency	PCA, Feature Selection, non-negative matrix factorisation
Model Selection	Improved accuracy via parameter tuning	Grid search, cross validation, metrics
Preprocessing	Transforming data (i.e text) for use in ML algorithms	Feature Extraction

Taken from scikit-learn.org

Practical – Getting Started

1. If the files aren't on the VM, clone them from GitHub using 'git clone https://github.com/gregparkes/ML_Intro.git'
2. Alternatively use your own machine, install Anaconda through Homebrew or follow instructions from https://www.continuum.io/downloads for your OS. Once you have full Anaconda installed, in Terminal type 'pip install sklearn', 'pip install tensorflow' and 'pip install graphviz' to ensure the correct packages are installed. NumPy, Pandas and Matplotlib should come with the distribution.

Practical 1.1 and 1.2 – Classification

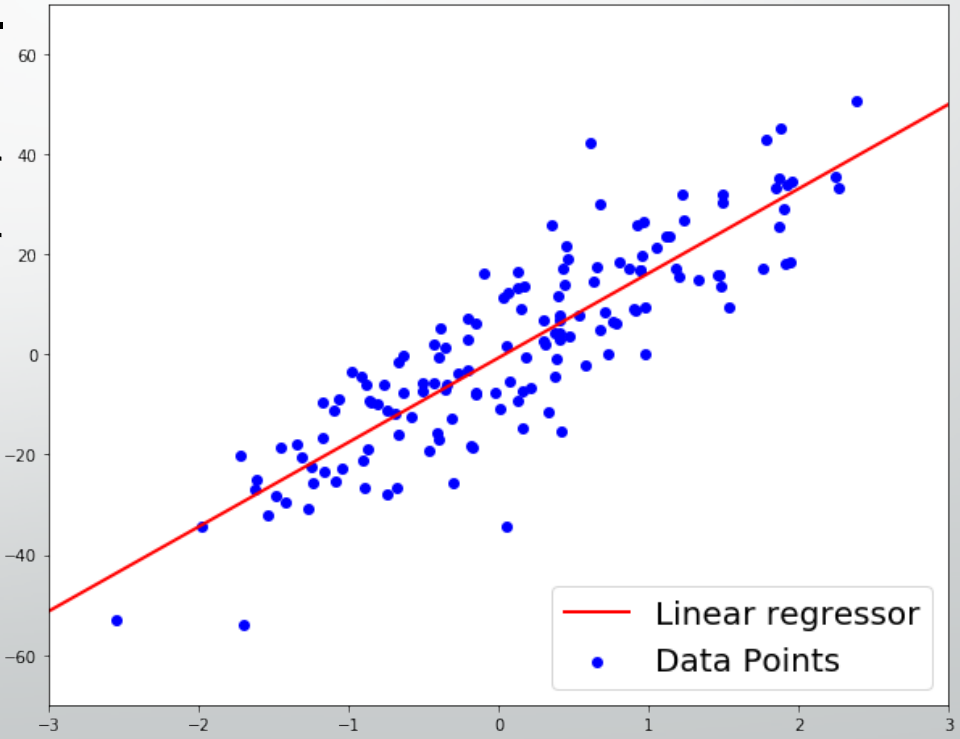
- Open Terminal (Ctrl-Alt-T), cd to Practical_1 directory, then call 'jupyter notebook'. Open up apples.ipynb. Look at the code and understand it.
- Open up iris.ipynb and complete the tasks. Try and complete at least the first 3 examples before looking at the solutions!
- Look at the solutions in iris-solutions.ipynb.

Introduction Summary

- Machine Learning is a subset of AI and is concerned with learning patterns in data to identify and predict useful features.
- Ski-Kit Learn is the primary comprehensive package available in Python.
- Classification is primarily concerned with identifying to which category an object belongs to.
- Decision Trees and K-Nearest Neighbours are examples of classifiers.

Regression

- Regression is a measure of the relationship between some input values (features) and an output value (target).
- Regression can be used for prediction of a target variable, the modeling of relationships, and the testing of hypotheses.
- There are many different regression techniques.
- Ski-Kit learn contains useful modules for regression.

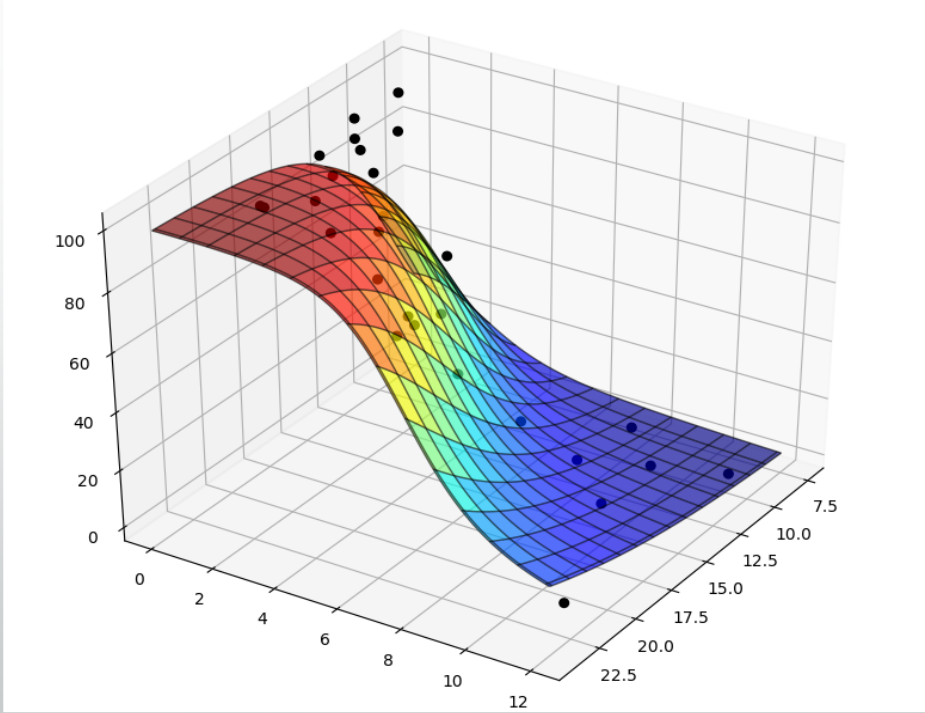


Regression

- Regression becomes very useful when the dataset being dealt with is large or there are many features to it.
- In a linear regression model, n variables satisfy a linear relationship,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon_i$$

- where the coefficients β are unknown and ε_i are random error terms.
- By fitting a dataset, the β coefficients can be found and a model made.

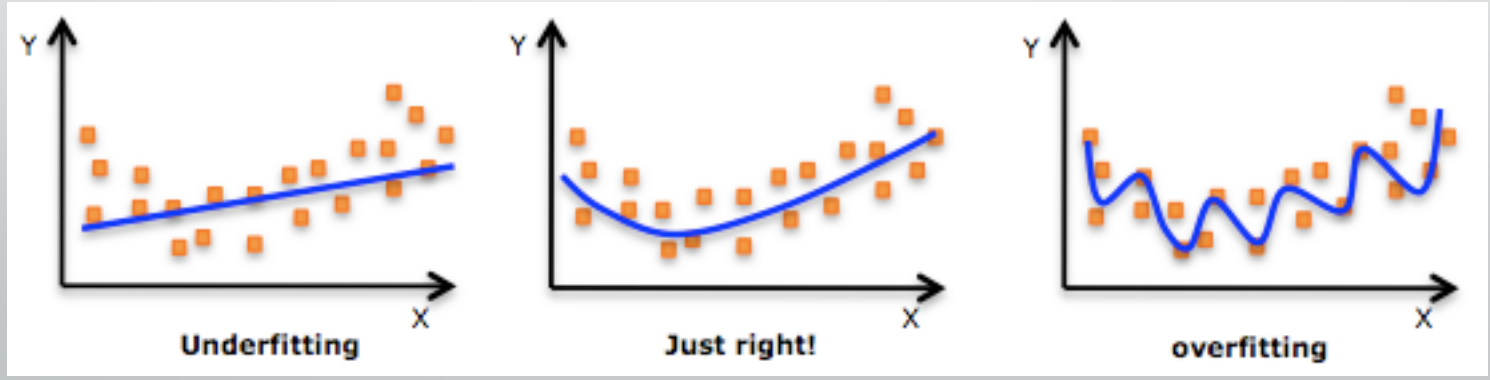


Preprocessing

- Preprocessing is the processing of data before modeling or analysis such that it is gotten into a ready state to be worked on.
- An example of preprocessing was seen earlier in the first classification exercise, where the texture of the fruit was given as a string and had to be converted to a binary number.
- Other preprocessing includes the splitting of data into training and testing sets and normalisation of the data.
- More about preprocessing can be found on the scikit-learn website: http://scikit-learn.org/stable/modules/preprocessing.html#preprocessing

Underfiitting and Overfitting

- If underfitted a model will not be good at making predictions.
- If overfitted a model has been trained too much on one dataset and while it will good accuracy for that set, for other sets it will not have good accuracy.
- One way to reduce overfitting is to only include features in a dataset that are believed to have a significant impact on the target.



https://stats.stackexchange.com/questions/192007/what-measures-you-look-at-the-determine-over-fitting-in-linear-regression/192021

Scikit-Learn and Regression

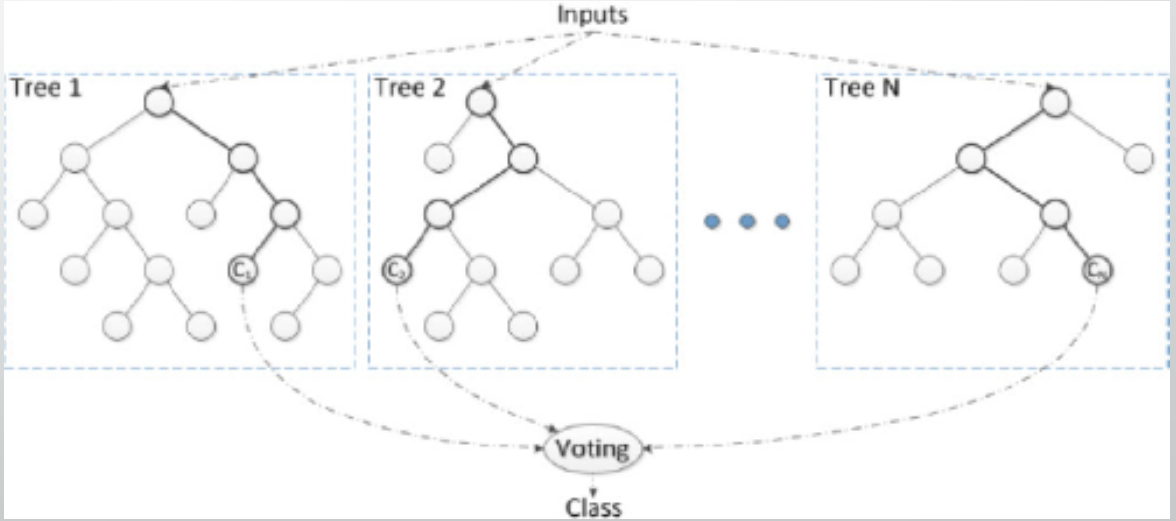
- Scikit-learn makes linear regression very simple.
- Import the linear_model module from sklearn, choose which linear model to use, fit the data to it, analyse and test the results.
- In scikit-learn there are a number of regression techniques available, including linear, lasso, and ridge regression.
- The linear regression techniques can be found under sklearn.linear_model
- Scikit-learn comes with some built in dataset which can be found under sklearn.datasets

Regression and Decision Trees

- When the target in a dataset is discrete the problem is one of classification.
- When the target in a dataset is continuous then the problem is one of prediction.
- It was seen how decision trees were used to classify the type of iris species.
- Decision trees can be used continuous targets to predict values and the importance of each feature.

Extra Tree Regressor

- To help improve accuracy for decision trees with regression and extra tree regressor can be used.
- An extra tree regressor fits a number of randomised trees on different subsamples of the dataset and averages them to improve the accuracy and to control overfitting.
- When an optimal selection of features is chosen the Extra Tree performs as well as the Random Forest, though in general is computationally faster.



https://wwrchgate.net/301243118_fig4_Figure-6-Random-Forest-Modelw.resea

Feature Importance and Metrics

- Extra Trees are hard to visualise as there's typically a large number of trees.
- Instead, useful information such as feature importance can be extracted.
- Feature importance gives a score to each feature that is a measure of how much they affect the target.
- To gauge the accuracy metrics such as the R-squared value can be used.
- Feature importance and the R-squared values are easy to extract in scikit-learn.

Regression Practical

- An example of linear regression analysis using scikit-learn and Python can be found in the 'RegressionPractical' Jupyter Notebook, located in the 'Practical_2' directory.
- Follow through the regression analysis in this Notebook and perform the same regression analysis for the Boston dataset found in scikit-learn.

Regression Summary

- Regression can be used to find the relationship between a target variable and feature variables.
- It can also be used to predict values of the target variable for given feature values.
- Decision trees can be used to make a regression model.
- Feature importances can be found for a given model.

Image Classifier With TensorFlow

Overview

1. Basic Introduction to Deep-learning
2. TensorFlow for Poets
3. Practical 1
4. Practical 2

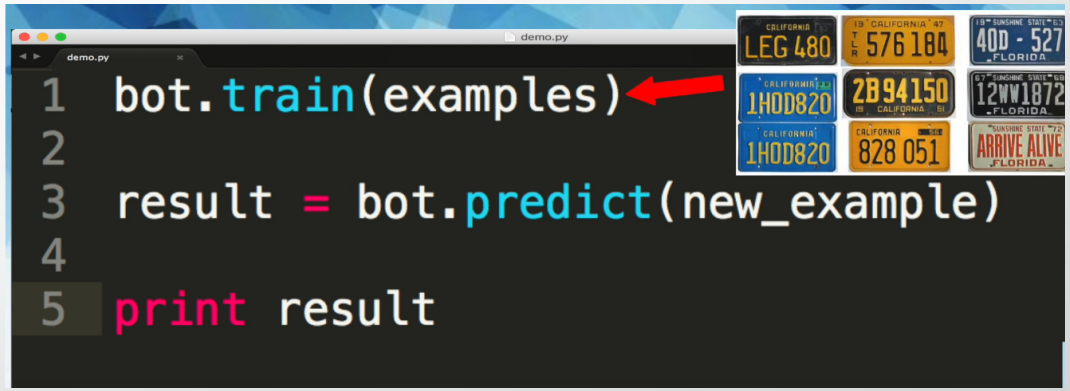
Deep-Learning

Hard Coding

```
1  def detect_letters
2
3  def detect_shapes
4
5  def detect_colors
6
7  def detect_state
```

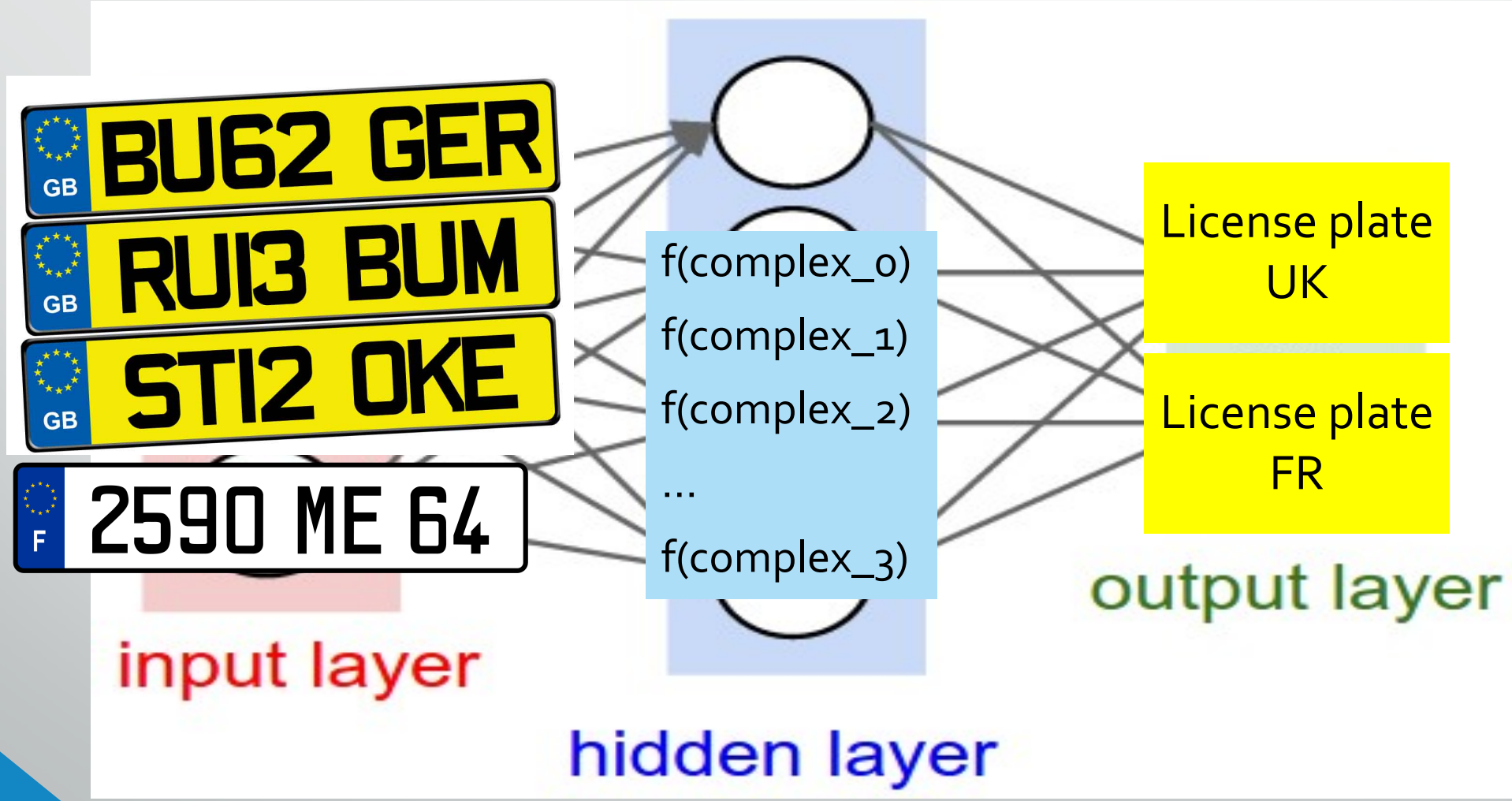
E.G License plate

Machine-Learning



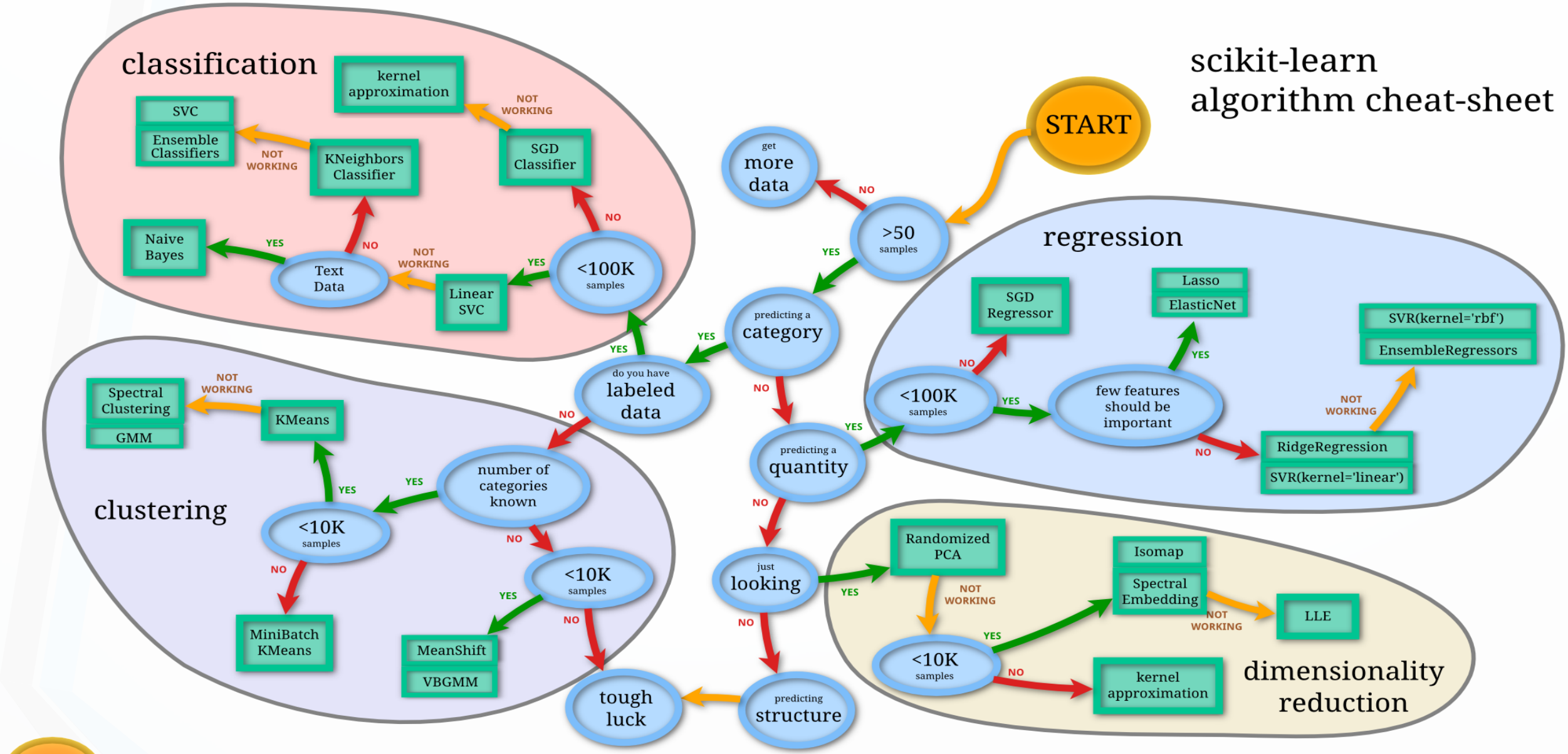
Recap:
Supervised: Feedback every time – (Practical 1)
Unsupervised: No feedback, learns by itself
Reinforced learning: Feedback if positive(trial and error) – Practical 2

Neural Network



http://playground.tensorflow.org/

scikit-learn algorithm cheat-sheet



Back

scikit learn

What is TensorFlow?

- TensorFlow is a deep learning library recently open-sourced by Google.
- But what does it actually do?

TensorFlow provides primitives for defining functions on tensors and automatically computing their derivatives.

http://playground.tensorflow.org/

Practical 1 - Supervised

https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/?utm_campaign=chrome_series_machinelearning_063016&utm_source=gdev&utm_medium=yt-desc#0

Practical 2

What about a game using reinforcement learning?

OpenAI

Input

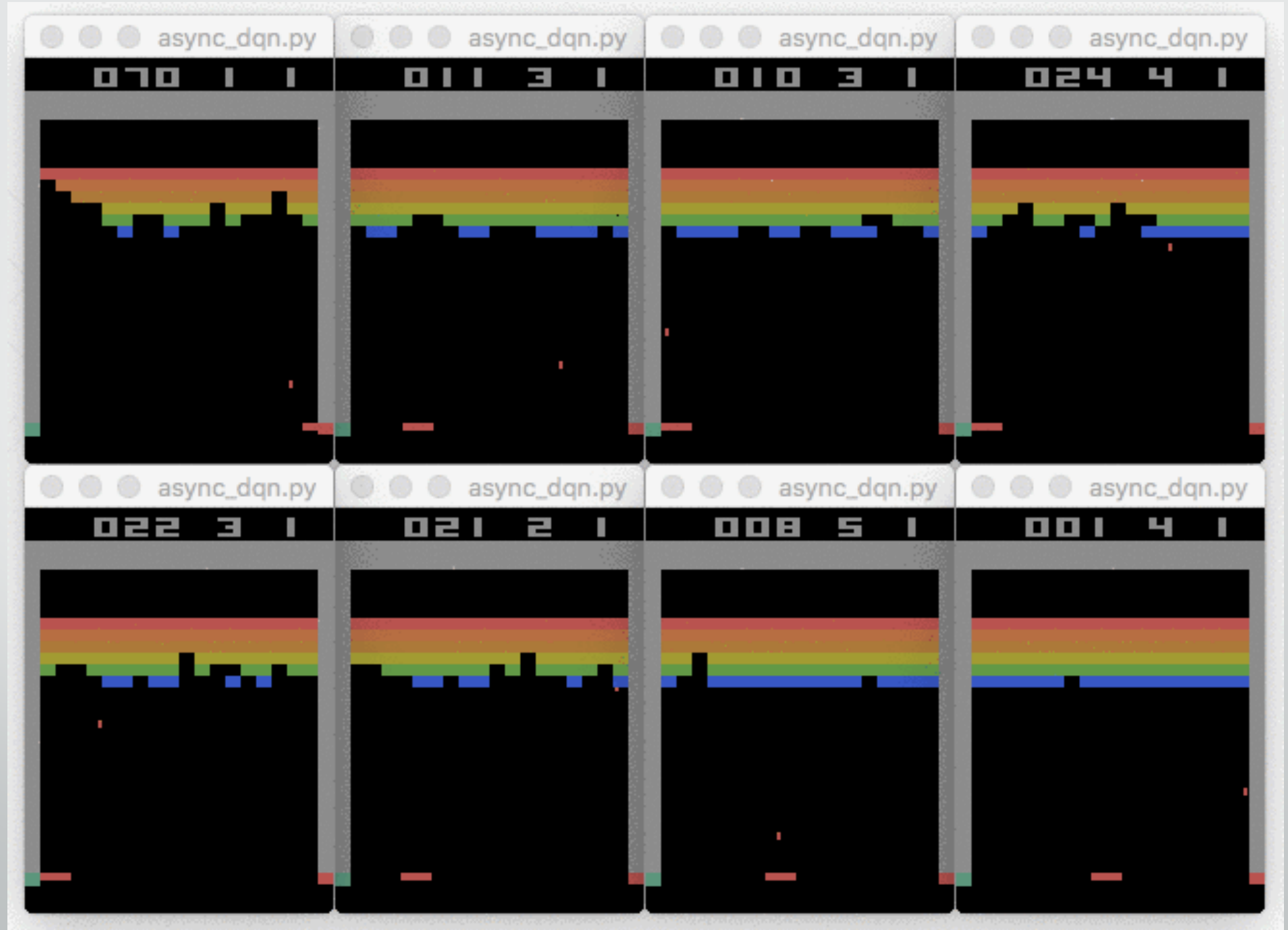
Output

Raw Pixel

Maximise the score

https://github.com/HackerHouseYT/OpenAI-NEAT

OpenAI


async_dqn.py
070 1 1
async_dqn.py
011 3 1
async_dqn.py
010 3 1
async_dqn.py
024 4 1
async_dqn.py
022 3 1
async_dqn.py
021 2 1
async_dqn.py
008 5 1
async_dqn.py
001 4 1

Deep-Learning Summary

- A **deep** neural network (DNN) is an artificial neural network (ANN) with multiple hidden layers of units between the input and output layers.
- Computational deep learning is closely related to a class of theories of brain development.
- Computer A.I use deep learning to understand how users interact.

Summary

- Machine learning is a wide ranging topic that focuses on learning from large datasets.
- Data from discrete types of input can be classified using decision trees.
- Regression can be used to find relationships in datasets and to make predictions.
- TensorFlow can be used for image classification.

This work was supported by an EPSRC Doctoral Training Centre grant (EP/L015382/1)

Further Resources

- The scikit-learn website: www.scikit-learn.org
- The tensorflow website: www.tensorflow.org
- A YouTube series on making a basic neural network: https://goo.gl/uvhELR
- A YouTube series on Machine Learning (inspiration): https://www.youtube.com/watch?v=cKxRvEZd3Mw