

The Complete Machine Learning & Data Science Bootcamp - Course Resources List

Section 2: Predict Movie Box Office Revenue with Linear Regression

- the-numbers.com - Movie budgets and revenue data
- [Try Jupyter Notebook Online](#) (no installation)
- pandas.pydata.org - the Pandas data analysis library
- matplotlib.org - the Matplotlib graphing and plotting library

Section 3: Python Programming for Data Science and Machine Learning

- [Install Jupyter with the Python Anaconda distribution](#)
- [The original research paper on test scores and LSD tissue concentration](#) (for the more curious)
- [Raw Experiment Data](#)
- [Exercise 1 Solution - Python Variables](#)
- [Exercise 2 Solution - Python Lists](#)
- [Exercise 3 Solution - Python Functions Part 1](#)
- [Exercise 4 Solution - Python Functions Part 2](#)
- [Exercise 5 Solution - Python Functions Part 3](#)

Section 4: Introduction to Optimisation and the Gradient Descent Algorithm

- [Symbolab.com](https://symbolab.com) - an online derivative calculator
- [SymPy Homepage](#) - a Python library for symbolic mathematics
- [Exercise 6 Solution - Python Loops](#)

Section 5: Predict House Prices with Multivariable Linear Regression

- [load_boston\(\) documentation](#) - available through scikit learn's website
- [scikit learn regression metrics documentation](#)
- [US inflation calculator](#)
- [Exercise 7 Solution - Conditional Statements](#)

Section 6: Pre-Process Text Data for a Naive Bayes Classifier to Filter Spam Emails (Part 1)

- [Spam Assassin data description](#)
- [VS Code](#) - optional Text Editor (my current favourite)
- [Atom](#) - optional Text Editor
- [Spam Assassin Public Corpus](#) - the original email dataset source
- [Detailed description of the email data](#)
- [XKCD on Unicode](#)
- [The Timewaster Letters](#) by Robin Cooper
- [jsonmate.com](#) - visualise JSONs
- [flatuicolors.com](#) - colours that make pie charts look pretty
- [www.nltk.org](#) - Natural Language Toolkit (NLTK)
- [Documentation on Python sets](#)
- [Word stemmers on nltk.org](#)
- [Martin Porter's \(rather humorous\) homepage](#) - creator of the Porter Stemmer
- [example.com](#) - clear and beautiful HTML on the web. Right-click to view source
- [BeautifulSoup](#) - a tool to pull data out of HTML and XML files
- [github.com/amueller/word_cloud](#) - Andreas Mueller's WordCloud Github Repo
- [WordCloud documentation](#)
- [Pillow documentation](#) - a module for image manipulation module
- [Colormap reference](#) (Matplotlib)
- [Font Awesome](#) - free icons for masks
- [Google Fonts](#) - free fonts for your projects
- [www.wordclouds.com](#) - online word cloud generator
- [What is mqjibake?](#)

Section 7: Train a Naive Bayes Classifier to Create a Spam Filter (Part 2)

- [Laplace Smoothing](#) (for the more curious)
- [Numpy savetxt\(\) documentation](#)
- [Numpy loadtxt\(\) documentation](#)

Section 8: Test and Evaluate a Naive Bayes Classifier (Part 3)

- [Numpy .dot\(\) product documentation](#)
- [Matplotlib marker types](#) for data points
- [Seaborn colour palettes](#) for graphs and charts
- [Color Lisa](#) - curated colours from famous artists
- [webmd symptom checker](#) - the perfect tool for every hypochondriac?

Section 9: Introduction to Neural Networks and How to Use Pre-Trained Models

- [Google Translate singing](#)
- [Google Colab Notebooks](#) - getting started
- [ModelZoo.co](#) - list of pre-trained models
- [TensorFlow Hub](#) - list of pre-trained models
- [Sample Images for Image Classification](#)
- [Unsplash.com](#) - high quality, royalty free images
- [Pre-trained Models available through Keras](#)
- [Keras InceptionResNetV2 documentation](#)
- [Keras models and their methods](#)
- [Keras VGG19 documentation](#)
- [Google Blog Post on NasNet and Image Classification](#)

Section 10: Build an Artificial Neural Network to Recognise Images using Keras and Tensorflow

- [Canadian Institute for Advanced Research \(CIFAR\)](#)
- [CIFAR 10 Dataset and Description by Alex Krizhevsky](#)
- [Keras Activation Functions](#)
- [Keras Optimizers](#)
- [Python strftime\(\) documentation](#)
- [Fitting a Keras Model](#)
- [Keras Dropout Layer documentation](#)
- [predict\(\) method documentation](#)
- [evaluate\(\) method documentation](#)
- [matplotlib colormaps](#) - examples
- [Python itertools](#) - efficient looping

Section 11: Use Tensorflow to Classify Handwritten Digits

- [The MNIST Database](#)
- [Numpy eye\(\) documentation](#)
- [Tensorflow Placeholder documentation](#)
- [Tensorflow matmul documentation](#)
- [Tensorflow relu documentation](#)
- [Tensorflow softmax documentation](#)
- [Softmax Cross Entropy with Logits](#)
- [Adam Optimizer](#)
- [MNIST handwritten digit database model comparison](#)

Section 12: Serving a Tensorflow Model through a Website

- [TF Session Object documentation](#)
- [Saved Model documentation](#)
- [Saved Model load\(\) documentation](#)
- [Session run\(\) documentation](#)
- [Model conversion](#)
- [Tensorflow.js converter](#)
- [Managing environments with Anaconda](#)
- [Atom text editor](#)
- [VS Code](#)
- [MDN - Web Development Documentation](#)
- [W3 Schools](#)
- [FlatUI Colors](#)
- [Getting started with Tensorflow.js](#)
- [Favicon Generator](#)
- [MDN events](#)
- [W3 Schools HTML Canvas reference](#)
- [Download the OpenCV.js file](#)
- [Geometric transformations for images](#)
- [Contour Features & the Centre of Mass](#)
- [Javascript map\(\) documentation](#)
- [Tensorflow.js operations](#)
- [Tensorflow.js dataSync](#)
- [Window setTimeout\(\) documentation](#)
- [Github Pages](#)