1495 UNIVERSITY OF ABERDEEN

CELEBRATING
525 YEARS
1495 – 2020

ABERDEEN 2040

GitHub, Python 3 and an Introduction to Machine Learning with Linear Regression

Exploratory Data Analysis

California Housing CSV Dataset

Coding, GIS and Remote Sensing Data for Glaciology

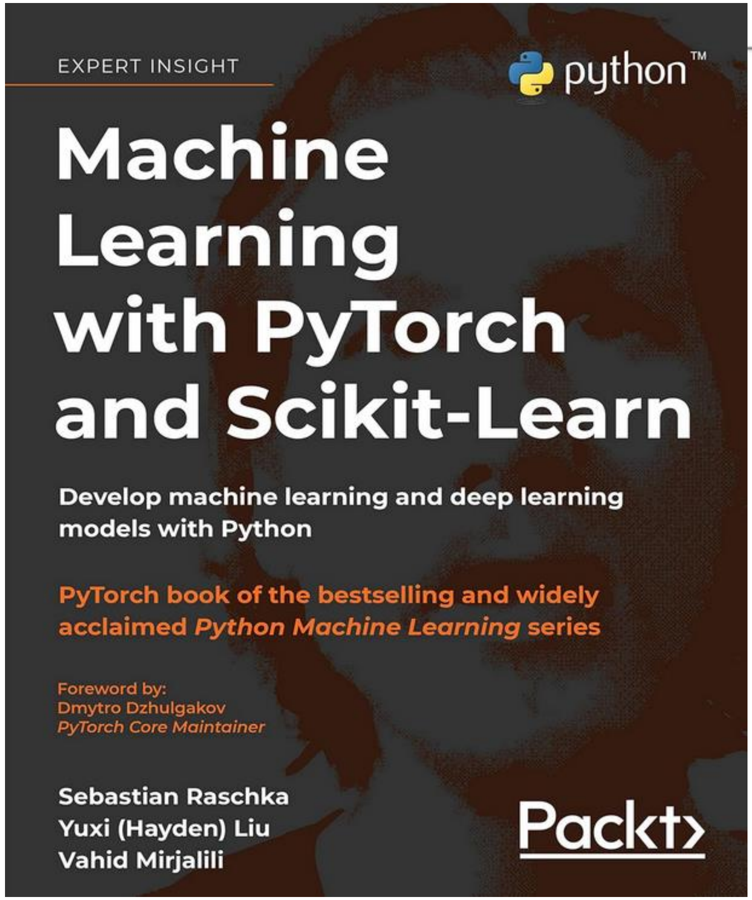
27 February 2025

St Mary’s G15

Overview

1. GitHub
2. Python 3
3. Machine Learning Cross Validation Methodology
4. Simple Linear Regression
5. Multiple Linear Regression

Machine Learning Reading



EXPERT INSIGHT

python™

Machine Learning with PyTorch and Scikit-Learn

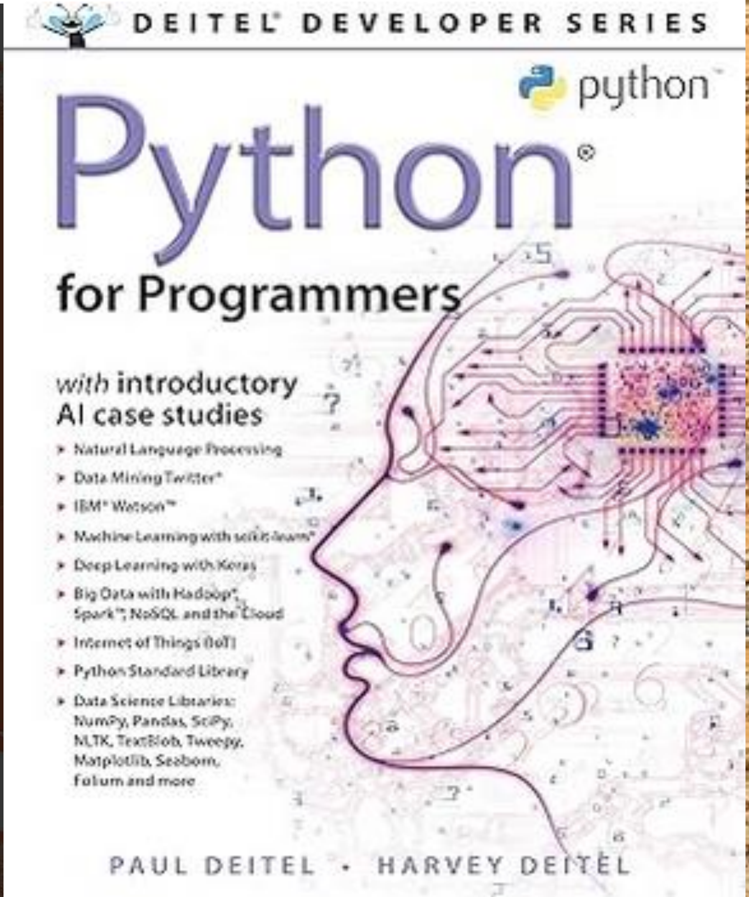
Develop machine learning and deep learning models with Python

PyTorch book of the bestselling and widely acclaimed *Python Machine Learning* series

Foreword by:
Dmytro Dzhulgakov
PyTorch Core Maintainer

Sebastian Raschka
Yuxi (Hayden) Liu
Vahid Mirjalili

Packt›



DEITEL® DEVELOPER SERIES

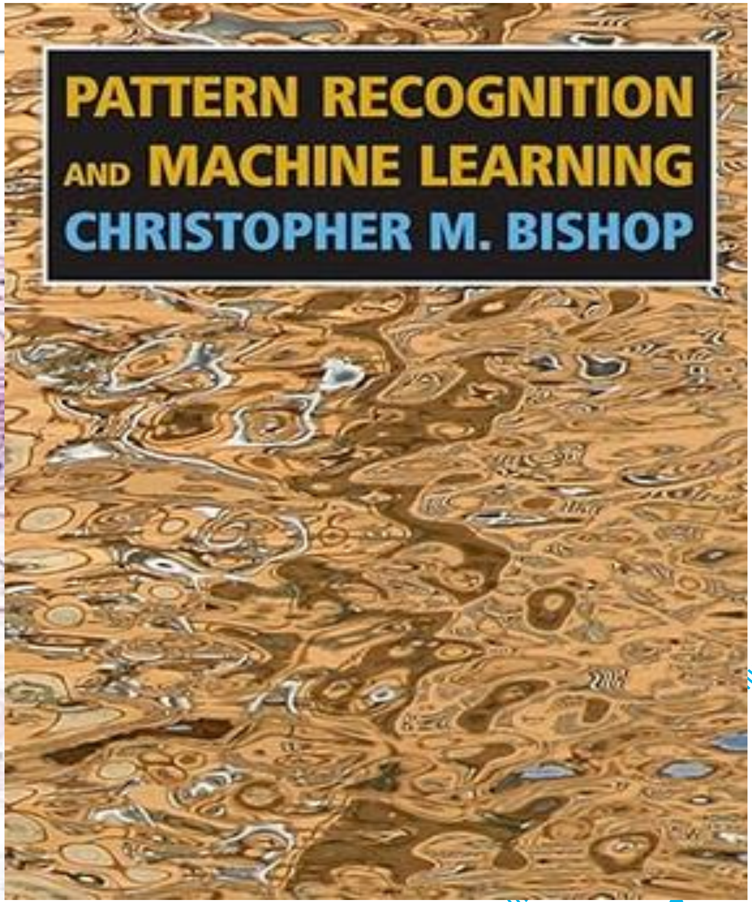
python™

Python® for Programmers

with introductory AI case studies

- Natural Language Processing
- Data Mining Twitter®
- IBM® Watson™
- Machine Learning with scikit-learn®
- Deep Learning with Keras
- Big Data with Hadoop®, Spark™, NoSQL and the Cloud
- Internet of Things (IoT)
- Python Standard Library
- Data Science Libraries: NumPy, Pandas, SciPy, NLTK, TextBlob, Tweepy, Matplotlib, Seaborn, Folium and more

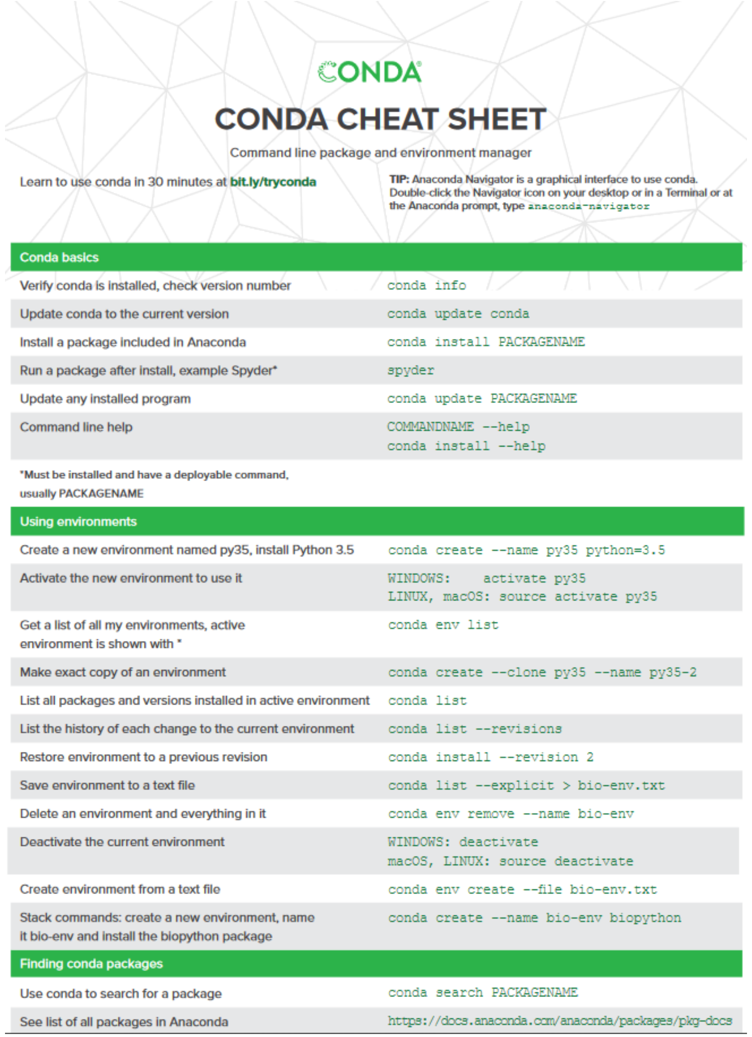
PAUL DEITEL · HARVEY DEITEL



PATTERN RECOGNITION AND MACHINE LEARNING

CHRISTOPHER M. BISHOP

Anaconda Cheat Sheet



CONDA

CONDA CHEAT SHEET

Command line package and environment manager

Learn to use conda in 30 minutes at bit.ly/tryconda

TIP: Anaconda Navigator is a graphical interface to use conda. Double-click the Navigator icon on your desktop or in a Terminal or at the Anaconda prompt, type `anaconda-navigator`

Conda basics

Verify conda is installed, check version number	`conda info`
Update conda to the current version	`conda update conda`
Install a package included in Anaconda	`conda install PACKAGENAME`
Run a package after install, example Spyder*	`spyder`
Update any installed program	`conda update PACKAGENAME`

Command line help	`COMMANDNAME --help`
	`conda install --help`

*Must be installed and have a deployable command, usually PACKAGENAME

Using environments

Create a new environment named py35, install Python 3.5	`conda create --name py35 python=3.5`

Activate the new environment to use it	`WINDOWS:    activate py35`
	`LINUX, macOS: source activate py35`

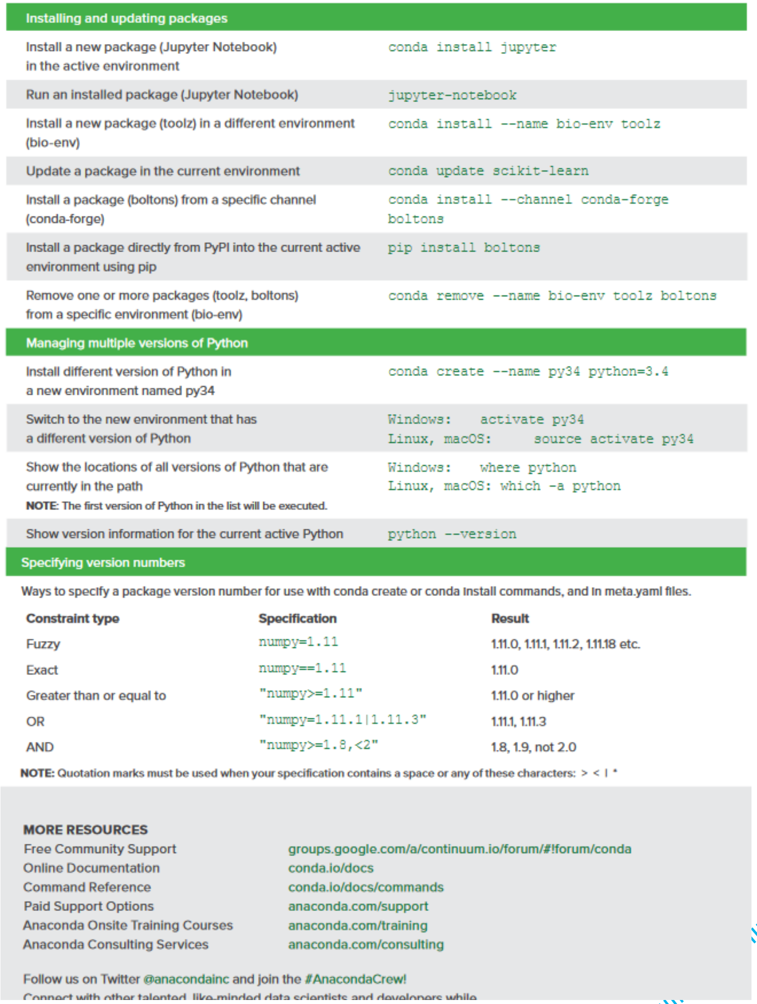
Get a list of all my environments, active environment is shown with *	`conda env list`
Make exact copy of an environment	`conda create --clone py35 --name py35-2`
List all packages and versions installed in active environment	`conda list`
List the history of each change to the current environment	`conda list --revisions`
Restore environment to a previous revision	`conda install --revision 2`
Save environment to a text file	`conda list --explicit > bio-env.txt`
Delete an environment and everything in it	`conda env remove --name bio-env`

Deactivate the current environment	`WINDOWS: deactivate`
	`macOS, LINUX: source deactivate`

Create environment from a text file	`conda env create --file bio-env.txt`
Stack commands: create a new environment, name it bio-env and install the biopython package	`conda create --name bio-env biopython`

Finding conda packages

Use conda to search for a package	`conda search PACKAGENAME`
See list of all packages in Anaconda	https://docs.anaconda.com/anaconda/packages/pkg-docs



Installing and updating packages

Install a new package (Jupyter Notebook) in the active environment	`conda install jupyter`
Run an installed package (Jupyter Notebook)	`jupyter-notebook`
Install a new package (toolz) in a different environment (bio-env)	`conda install --name bio-env toolz`
Update a package in the current environment	`conda update scikit-learn`
Install a package (boltons) from a specific channel (conda-forge)	`conda install --channel conda-forge boltons`
Install a package directly from PyPI into the current active environment using pip	`pip install boltons`
Remove one or more packages (toolz, boltons) from a specific environment (bio-env)	`conda remove --name bio-env toolz boltons`

Managing multiple versions of Python

Install different version of Python in a new environment named py34	`conda create --name py34 python=3.4`

Switch to the new environment that has a different version of Python	`Windows:   activate py34`
	`Linux, macOS:    source activate py34`

Show the locations of all versions of Python that are currently in the path
NOTE: The first version of Python in the list will be executed.
	`Windows:   where python`
	`Linux, macOS: which -a python`

Show version information for the current active Python	`python --version`

Specifying version numbers

Ways to specify a package version number for use with conda create or conda install commands, and in meta.yaml files.

Constraint type	Specification	Result
Fuzzy	`numpy=1.11`	1.11.0, 1.11.1, 1.11.2, 1.11.18 etc.
Exact	`numpy==1.11`	1.11.0
Greater than or equal to	`"numpy>=1.11"`	1.11.0 or higher
OR	`"numpy=1.11.1|1.11.3"`	1.11.1, 1.11.3
AND	`"numpy>=1.8,<2"`	1.8, 1.9, not 2.0

NOTE: Quotation marks must be used when your specification contains a space or any of these characters: > < | *

MORE RESOURCES

Free Community Support	groups.google.com/a/continuum.io/forum/#!forum/conda
Online Documentation	conda.io/docs
Command Reference	conda.io/docs/commands
Paid Support Options	anaconda.com/support
Anaconda Onsite Training Courses	anaconda.com/training
Anaconda Consulting Services	anaconda.com/consulting

Follow us on Twitter @anacondainc and join the #AnacondaCrew!
Connect with other talented, like-minded data scientists and developers while

https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf

Python Code for the Session

The slides and the code for the session can be found implemented at:

GitHub ID: Stevieee83

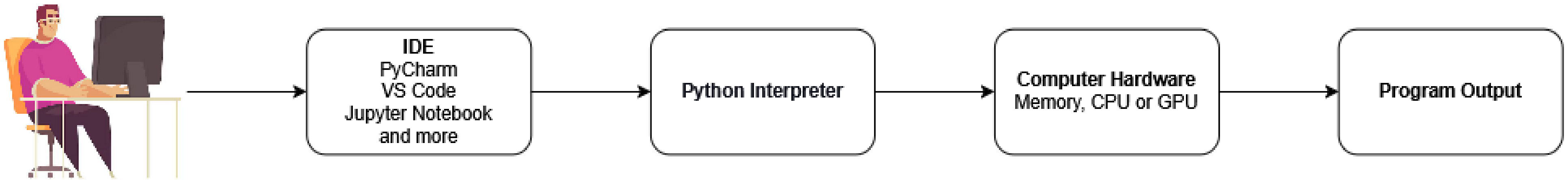
https://github.com/Stevieee83

Machine Learning Library: ScikitLearn

GitHub Web Browser Demonstration

Python 3

Python 3 ipython Interpreter



IDE
PyCharm
VS Code
Jupyter Notebook
and more

Python Interpreter

Computer Hardware
Memory, CPU or GPU

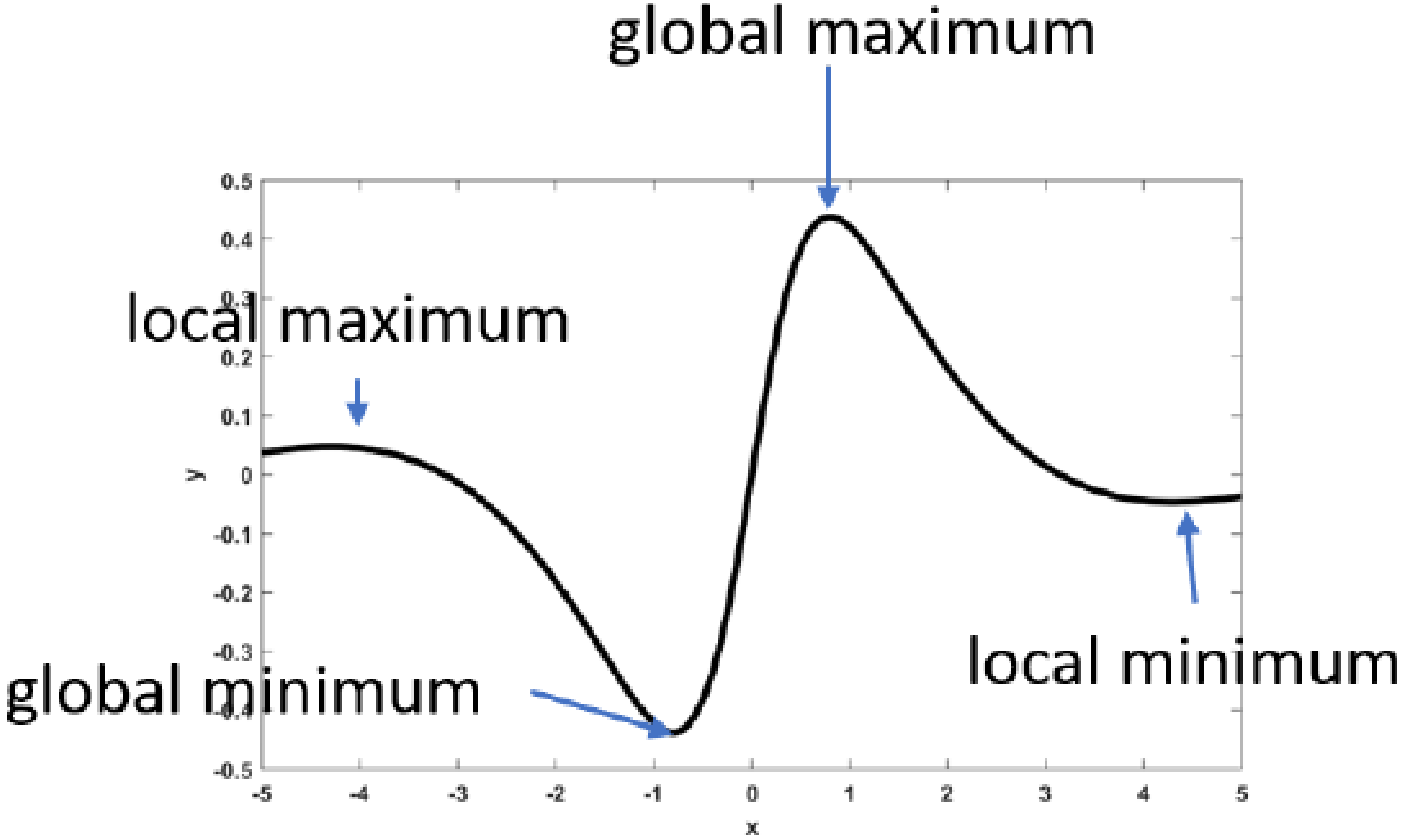
Program Output

Machine Learning with Linear Regression

Machine Learning Libraries

- ScikitLearn (well recognised machine learning library)
- Pandas (data processing Python library)
- NumPy (numerical Python library)
- Matplotlib (data visualisation Python library)
- Seaborn (data visualisation Python library)
- SciPy (statistics Python library)

Gradient Descent



global maximum

local maximum

global minimum

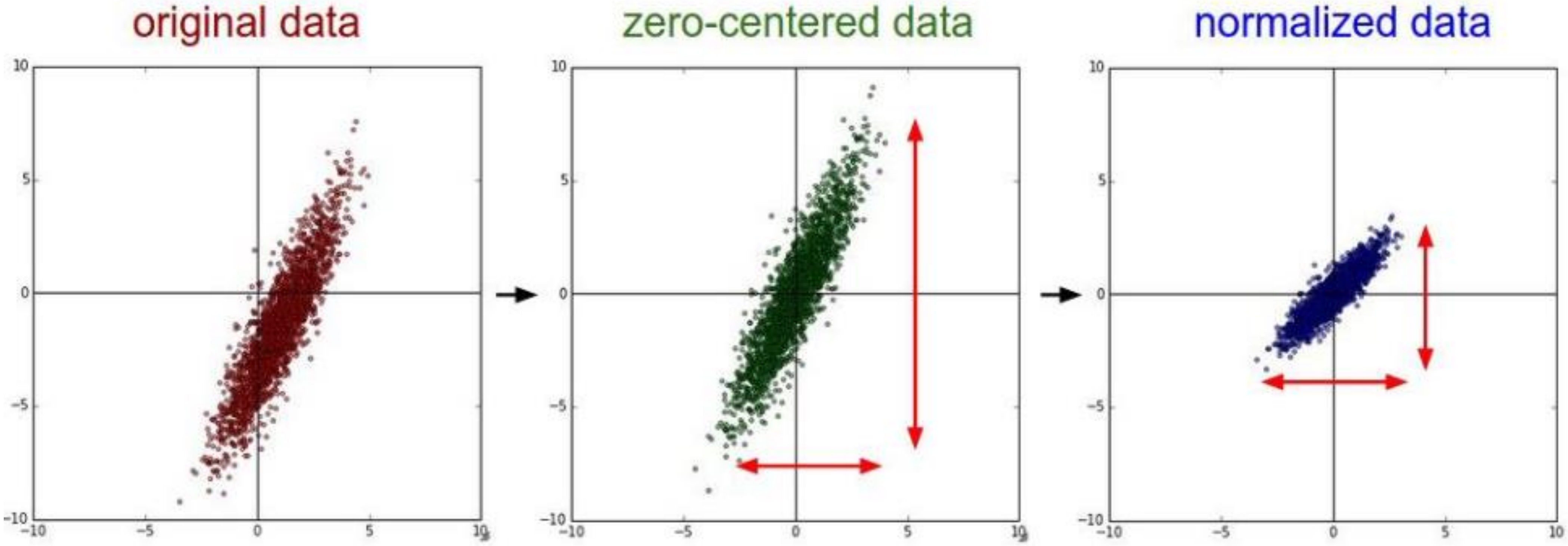
local minimum

$$X = X - learning\_rate * gradient$$

Exploratory Data Analysis Code Demonstration

Data Pre-Processing

Data Normalisation

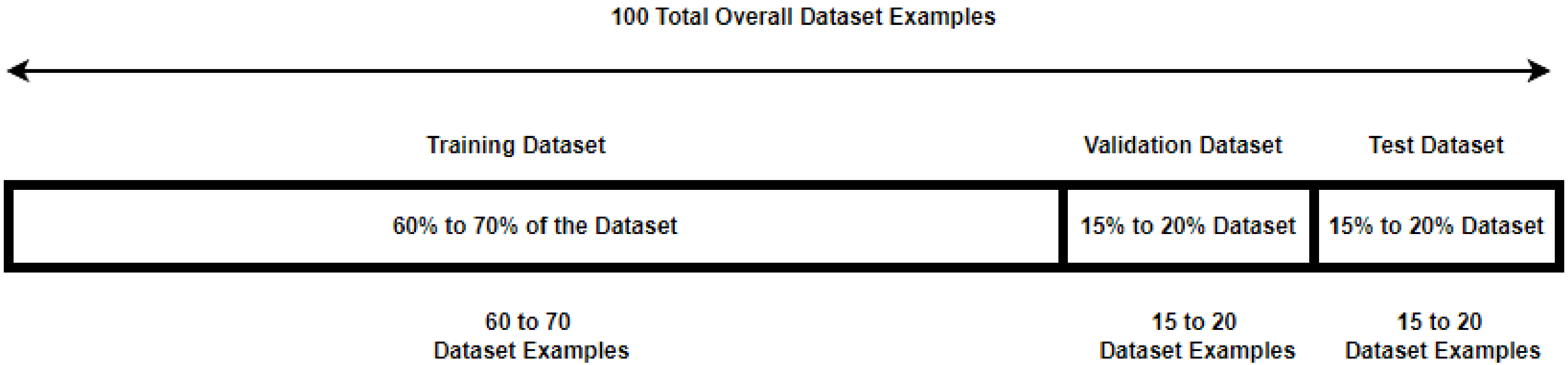


https://cs231n.stanford.edu/

Hold Out Cross Validation

- Split the dataset to 80% training and 20% test data
- Split the training dataset again to generate a 20% validation dataset (20% over all of the examples is recommended, not just the training dataset)
- Only use the test dataset after training and validating the model with the training and validation datasets.

Hold Out Cross Validation



100 Total Overall Dataset Examples

Training Dataset

60% to 70% of the Dataset

60 to 70
Dataset Examples

Validation Dataset

15% to 20% Dataset

15 to 20
Dataset Examples

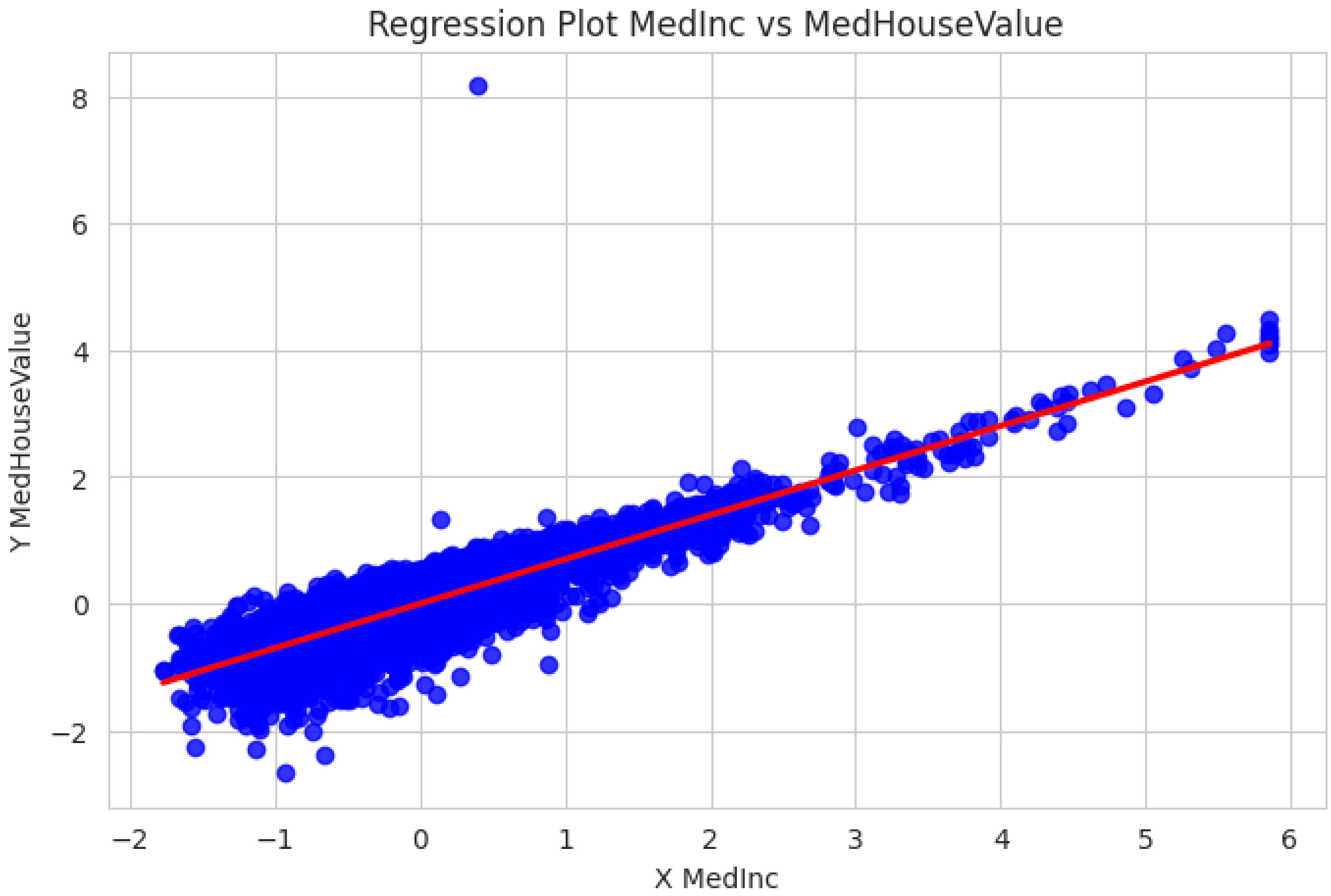
Test Dataset

15% to 20% Dataset

15 to 20
Dataset Examples

Simple Linear Regression

Simple Linear Regression



$$y(X, W) = W_o + (W1X1)$$

Training Linear Regression

- Set the Learning Rate
- Adapt the training iterations if required
- Tune any regularisation parameters

Loss in Linear Regression Models

- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)

MAE does not penalise Linear Regression models as much when there are outliers in the data compared to MSE and RMSE

Evaluating the Linear Regression

- R2 Score
- Pearson Linear Correlation Coefficient (PLCC)
- Spearman Rank Correlation Coefficient (SRCC)
- Kendal Rank Correlation Coefficient (KRCC)

Simple Linear Regression Evaluation (ScikitLearn)

Dataset Split	MSE Loss
Training	0.52
Validation	0.55
Test	0.53

Simple Linear Regression Evaluation (Python)

Dataset Split	MSE Loss
Training	0.52
Validation	0.52
Test	0.53

Simple Linear Regression Evaluation (ScikitLearn)

Dataset Split	R2 Score	PLCC	SRCC	KRCC
Test	0.46	0.68	0.67	0.49

Simple Linear Regression Evaluation (Python)

Dataset Split	R2 Score	PLCC	SRCC	KRCC
Test	0.46	0.68	0.67	0.49

Multiple Linear Regression

Multiple Linear Regression

- Exactly the same concept as Simple Linear Regression except there is more than one feature for the model to use for prediction.
- 8 features are in the example notebooks for the session.

$$y(X, W) = W_o + (W_1 X_1) + \ldots\ldots + (W_D X_D)$$

$$y(X, W) = W_o + (W_1 X_1) + \ldots\ldots + (W_8 X_8)$$

Multiple Linear Regression Evaluation (ScikitLearn)

Dataset Split	MSE Loss
Training	0.39
Validation	0.40
Test	0.42

Multiple Linear Regression Evaluation (Python)

Dataset Split	MSE Loss
Training	0.39
Validation	0.39
Test	0.42

Multiple Linear Regression Evaluation (ScikitLearn)

Dataset Split	R2 Score	PLCC	SRCC	KRCC
Test	0.58	0.76	0.81	0.62

Multiple Linear Regression Evaluation (Python)

Dataset Split	R2 Score	PLCC	SRCC	KRCC
Test	0.58	0.76	0.81	0.62

Summary

- Exploratory data analysis and data pre-processing are essential before training any machine learning model with any dataset.
- Understanding gradient descent and how the learning rate effects the weight parameters are essential for training any machine learning model that uses gradient descent .
- Linear Regression models can only be used on data where the features have linear relationships to the output predictions made by the model over the complete range.

References

[1] Christopher M. Bishopl. Pattern Recognition and Machine Learning. Basingstoke, UK: Springer, 2007. isbn: 978-0387310732.

[2] Paul Deitel and Harvey Deitel. Python For Programmers. London, UK: Pearson, 2019. isbn: 978-0-13-522433-5.

[3] Li, F.-F. CS231N: Deep Learning for Computer Vision, Stanford University CS231n: Deep Learning for Computer Vision. Available at: https://cs231n.stanford.edu/ (Accessed: 24 April 2024).

[4] Sebastian Raschka, Yuxi (Hayden) Liu, and Vahid Mirjalili. Machine Learning with PyTorch and Scikit-Learn. Birmingham, UK: Packt Publishing, 2022. isbn: 978-1801819312.