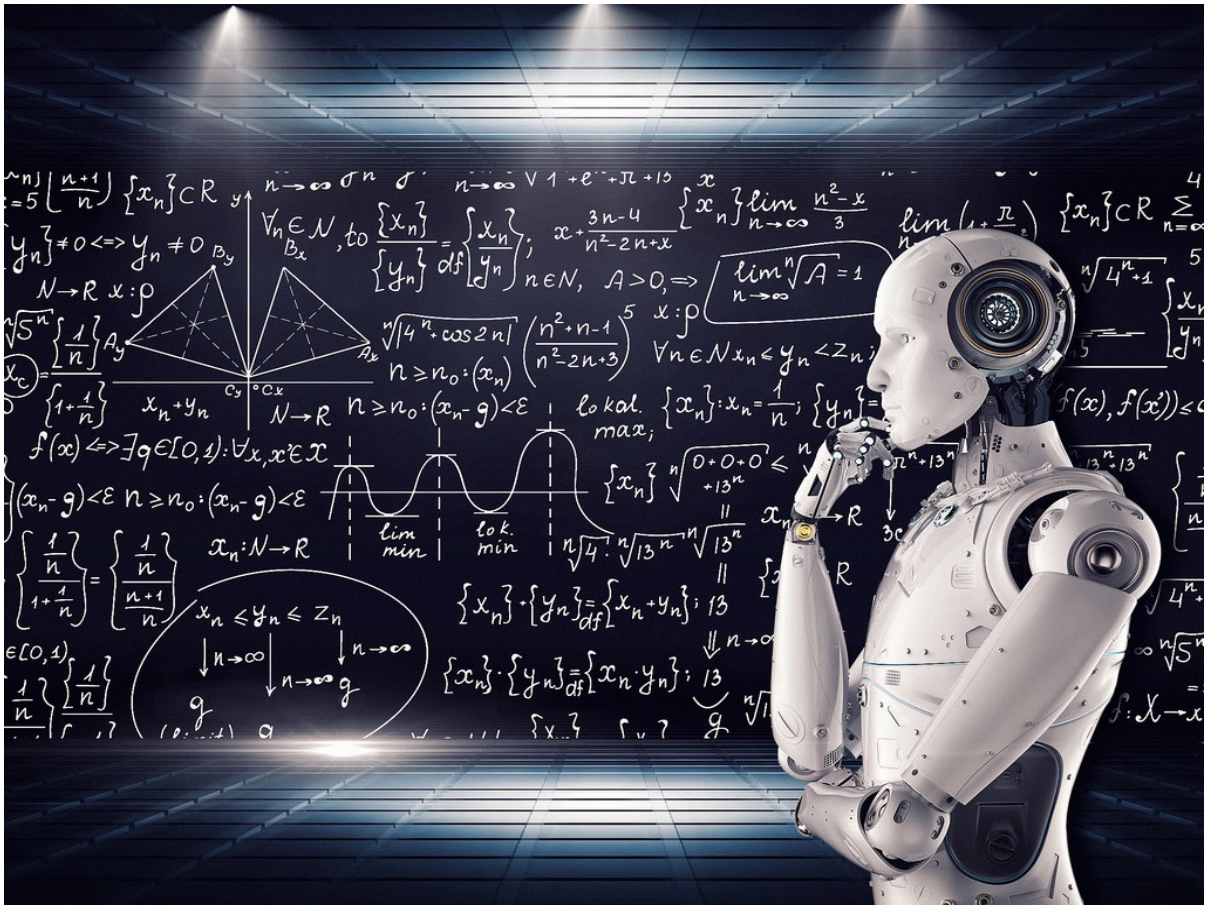
Machine-Learning-with-Python

Sep 14, 2022

Contents

Authored and maintained by Dr. Tirthajyoti Sarkar, Fremont, CA. Please feel free to add me on LinkedIn here.



CHAPTER 1

Requirements

- Python 3.5
- NumPy (`pip install numpy`)
- Pandas (`pip install pandas`)
- Scikit-learn (`pip install scikit-learn`)
- SciPy (`pip install scipy`)
- Statsmodels (`pip install statsmodels`)
- MatplotLib (`pip install matplotlib`)
- Seaborn (`pip install seaborn`)
- Sympy (`pip install sympy`)

You can start with this article that I wrote in Heartbeat magazine (on Medium platform):

"Some Essential Hacks and Tricks for Machine Learning with Python"

CHAPTER 2

Essential tutorial-type notebooks on Pandas, Numpy, and visualizations

Jupyter notebooks covering a wide range of functions and operations on the topics of NumPy, Pandans, Seaborn, matplotlib etc.

- Detailed Numpy operations
- Detailed Pandas operations
- Numpy and Pandas quick basics operations
- Basics of visualization with Matplotlib and Seaborn
- Advanced Pandas operations
- How to read various data sources
- PDF reading and table processing demo
- How fast are Numpy operations compared to pure Python code? (Read my article on Medium related to this topic)
- Fast reading from Numpy using .npy file format (Read my article on Medium on this topic)

CHAPTER 3

Regression related Notebooks

- Simple linear regression with t-statistic generation (Here is the Notebook)
- Linear regression as a statistical estimation problem (Here is the Notebook)
- Multiple ways to perform linear regression in Python and their speed comparison (Here is the Notebook). Also check the article I wrote on freeCodeCamp
- Multi-variate regression with regularization (Here is the Notebook)
- Polynomial regression using **scikit-learn pipeline feature** (Here is the Notebook). Also check the article I wrote on Towards Data Science.
- Decision trees and Random Forest regression (showing how the Random Forest works as a robust/regularized meta-estimator rejecting overfitting) (Here is the Notebook).
- Detailed visual analytics and goodness-of-fit diagnostic tests for a linear regression problem (Here is the Notebook).
- How linear regression and neural network fare in the task of nonlinear function approximation (Here is the Notebook)
- Robust regression fit example (Here is the Notebook)
- Support vector regression example using nonlinear synthetic data (Here is the Notebook)

CHAPTER 4

Classification related Notebooks

- Logistic regression/classification (Here is the Notebook).
- k-nearest neighbor classification (Here is the Notebook).
- Decision trees and Random Forest Classification (Here is the Notebook).
- Support vector machine classification (Here is the Notebook). Also check the article I wrote in Towards Data Science on SVM and sorting algorithm.
- Naive Bayes classification (Here is the Notebook).
- Classification using Stochastic Gradient Descent (SGD) (Here is the Notebook).

CHAPTER 5

Clustering related Notebooks

- *K*-means clustering (Here is the Notebook).
- Affinity propagation (showing its time complexity and the effect of damping factor) (Here is the Notebook).
- Mean-shift technique (showing its time complexity and the effect of noise on cluster discovery) (Here is the Notebook).
- DBSCAN (showing how it can generically detect areas of high density irrespective of cluster shapes, which the k-means fails to do) (Here is the Notebook).
- Hierarchical clustering with Dendograms showing how to choose optimal number of clusters (Here is the Notebook).
- Clustering metrics better than the elbow-method (Here is the Notebook).

CHAPTER 6

Dimensionality reduction related Notebooks

- Principal component analysis (Here is the Notebook)
- Clustering combined with dimensionality reduction technqieus (Here is the Notebook)

CHAPTER 7

Complexity and Learning curve analysis

Complexity and learning curve analyses are essentially are part of the visual analytics that a data scientist must perform using the available dataset for comparing the merits of various ML algorithms.

Learning curve: Graphs that compares the performance of a model on training and testing data over a varying number of training instances. We should generally see performance improve as the number of training points increases.

Complexity curve: Graphs that show the model performance over training and validation set for varying degree of model complexity (e.g. degree of polynomial for linear regression, number of layers or neurons for neural networks, number of estimator trees for a Boosting algorithm or Random Forest).

- Complexity and learning curve with Lending club dataset (Here is the Notebook).
- Complexity and learning curve with a synthetic dataset using the `Hastie function` from Scikit-learn (Here is the Notebook).

CHAPTER 8

Random data generation using symbolic expressions

- Simple script to generate random polynomial expression/function (Here is the Notebook).
- How to use Sympy package to generate random datasets using symbolic mathematical expressions (Here is the Notebook). Also, here is the Python script if anybody wants to use it directly in their project.
- Here is my article on Medium on this topic: Random regression and classification problem generation with symbolic expression

CHAPTER 9

Simple deployment examples (serving ML models on web API)

- Serving a linear regression model through a simple HTTP server interface. User needs to request predictions by executing a Python script. Uses `Flask` and `Gunicorn`.
- Serving a recurrent neural network (RNN) through a HTTP webpage, complete with a web form, where users can input parameters and click a button to generate text based on the pre-trained RNN model. Uses `Flask`, `Jinja`, `Keras`/`TensorFlow`, `WTForms`.

CHAPTER 10

Object-oriented programming with machine learning

Implementing some of the core OOP principles in a machine learning context by building your own Scikit-learn-like estimator, and making it better.

Here is the complete Python script with the linear regression class, which can do fitting, prediction, cpmputation of regression metrics, plot outliers, plot diagnostics (linearity, constant variance, etc.), compute variance inflation factors.

I created a Python package based on this work, which offers simple Scikit-learn style interface API along with deep statistical inference and residual analysis capabilities for linear regression problems. Check it out here.

See my articles on Medium on this topic.

- Object-oriented programming for data scientists: Build your ML estimator
- How a simple mix of object-oriented programming can sharpen your deep learning prototype