**CHAPTERS**

* 1. **INTRODUCTION**
     1. **MACHINE LEARNING**

**Machine learning** algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics. Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

* **Supervised learning:** The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.
* **Unsupervised learning:** No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).
* **Reinforcement learning:** A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.

Other approaches have been developed which don't fit neatly into this three-fold categorization, and sometimes more than one is used by the same machine learning system.

MACHINE LEARNING LIBRARIES:

To implement this Machine learning network, we have the following options.

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Scikit-learn
* Pickle

MACHINE LEARNING FRAMEWORK:

* Flask
* **NUMPY**
  + 1. **PANDAS**

Pandas is an opensource Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named [Numpy](https://www.activestate.com/products/python/python-packages/), which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other [data science](https://www.activestate.com/products/python/python-data-science/) modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState’s[ActivePython](https://platform.activestate.com/featured-projects).

Pandas makes it simple to do many of the time consuming, repetitive tasks associated with working with data, including:

* Data cleansing
* Data fill
* Data normalization
* Merges and joins
* Data visualization
* Statistical analysis
* Data inspection
* Loading and saving data
* And much more

In fact, with Pandas, you can do everything that makes world-leading datascientists vote Pandas as the best data analysis and manipulation tool available.

* + 1. **MATPLOTLIB**

**Matplotlib** is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable opensource alternative to MATLAB. Developers can also use matplotlib’s APIs (Application Programming Interfaces) to embed plots in GUI applications.

A Python matplotlib script is structured so that a few lines of code are all that is required in most instances to generate a visual data plot. The matplotlib scripting layer overlays two APIs:

* The pyplot API is a hierarchy of Python code objects topped by *matplotlib.pyplot*
* An OO (Object-Oriented) API collection of objects that can be assembled with greater flexibility than pyplot. This API provides direct access to Matplotlib’s backend layers.
  + 1. **SCIKIT-LEARN**

**Scikit-learn** is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

**Components of scikit-learn:**

Scikit-learn comes loaded with a lot of features. Here are a few of them to help you understand the spread:

* **Supervised learning algorithms:** Think of any supervised machine learning algorithm you might have heard about and there is a very high chance that it is part of scikit-learn. Starting from Generalized linear models (e.g Linear Regression), Support Vector Machines (SVM), Decision Trees to Bayesian methods – all of them are part of scikit-learn toolbox. The spread of machine learning algorithms is one of the big reasons for the high usage of scikit-learn. I started using scikit to solve supervised learning problems and would recommend that to people new to scikit / machine learning as well.
* **Cross-validation:** There are various methods to check the accuracy of supervised models on unseen data using sklearn.
* **Unsupervised learning algorithms:**Again, there is a large spread of machine learning algorithms in the offering – starting from clustering, factor analysis, principal component analysis to unsupervised neural networks.
* **Various toy datasets:** This came in handy while learning scikit-learn. I had learned SAS using various academic datasets (e.g IRIS dataset, Boston House prices dataset). Having them handy while learning a new library helped a lot.
* **Feature extraction:** Scikit-learn for extracting features from images and text (e.gBag of words)