# PYTHON LIBRARIES

Some of the basic python libraries are:

## NUMPY

NumPy is a library for the Python programming language, adding support for large multidimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

## TIMEDATE

The datetime module supplies classes for manipulating dates and times in both simple and complex ways. While date and time arithmetic are supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation.

## RANDOM

This module implements pseudo-random number generators for various distributions.

For integers, there is uniform selection from a range. For sequences, there is uniform selection of a random element, a function to generate a random permutation of a list in-place, and a function for random sampling without replacement.

## COLLECTIONS

This module implements specialized container datatypes providing alternatives to Python’s general-purpose built-in containers, dict, list, set, and tuple.

## RE

This module provides regular expression matching operations similar to those found in Perl.

## SCIPY

SciPy is a free and open-source Python library used for scientific computing and technical computing.

SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering.

## WARNINGS

Warning messages are typically issued in situations where it is useful to alert the user of some condition in a program, where that condition (normally) doesn’t warrant raising an exception and terminating the program. In Python we can handle warning messages using this module.

# DATA MANUPULATION

We have following libraries for reading data, manipulating data and writing data for python:

## PANDAS

PANDAS is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

Library features

* DataFrame object for data manipulation with integrated indexing.
* Tools for reading and writing data between in-memory data structures and different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of data sets.
* Label-based slicing, fancy indexing, and subsetting of large data sets.
* Data structure column insertion and deletion.
* Group by engine allowing split-apply-combine operations on data sets.
* Data set merging and joining.
* Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
* Time series-functionality: Date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.
* Provides data filtration.

The library is highly optimized for performance, with critical code paths written in Cython or C.

# VISULIZATION LIBRARIES

Below are some very important visualization libraries available for python:

## MATPLOTLIB

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits.

## SEABORN

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures.

Here is some of the functionality that seaborn offers:

* A dataset-oriented API for examining relationships between multiple variables
* Specialized support for using categorical variables to show observations or aggregate statistics
* Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data
* Automatic estimation and plotting of linear regression models for different kinds dependent variables
* Convenient views onto the overall structure of complex datasets
* High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations
* Concise control over matplotlib figure styling with several built-in themes
* Tools for choosing color palettes that faithfully reveal patterns in your data

## PLOTLY

Plotly's Python graphing library makes interactive, publication-quality graphs online.

# DATA SCIENCE LIBRARIES

Below are some of the important libraries one should know to work on machine learning tasks:

## STATSMODELS

statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.

## SKLEARN

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms.

The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

* NumPy: Base n-dimensional array package
* SciPy: Fundamental library for scientific computing
* Matplotlib: Comprehensive 2D/3D plotting
* IPython: Enhanced interactive console
* Sympy: Symbolic mathematics
* Pandas: Data structures and analysis.

It is a Simple and efficient tool for data mining and data analysis.

## YELLOWBRICK

Yellowbrick is an open source, pure Python project that extends the scikit-learn [API](http://scikit-learn.org/stable/modules/classes.html) with visual analysis and diagnostic tools.

Yellowbrick combines scikit-learn with matplotlib in the best tradition of the scikit-learn documentation, but to produce visualizations for *your* models!

For users, Yellowbrick can help evaluate the performance, stability, and predictive value of machine learning models and assist in diagnosing problems throughout the machine learning workflow.

## PYOD

PyOD is a comprehensive and scalable **Python toolkit** for **detecting outlying objects** in multivariate data. This exciting yet challenging field is commonly referred as [Outlier Detection](https://en.wikipedia.org/wiki/Anomaly_detection) or [Anomaly Detection](https://en.wikipedia.org/wiki/Anomaly_detection).

## IMBLEARN

imbalanced-learn is a python package offering a number of re-sampling techniques commonly used in datasets showing strong between-class imbalance. It is compatible with [scikit-learn](http://scikit-learn.org/stable/) and is part of [scikit-learn-contrib](https://github.com/scikit-learn-contrib) projects.

## PRINCE

Prince is a library for doing factor analysis. This includes a variety of methods including principal component analysis (PCA) and correspondence analysis (CA). The goal is to provide an efficient implementation for each algorithm along with a scikit-learn API

You are supposed to use each method depending on your situation:

* All your variables are numeric: use principal component analysis (prince.PCA)
* You have a contingency table: use correspondence analysis (prince.CA)
* You have more than 2 variables and they are all categorical: use multiple correspondence analysis (prince.MCA)
* You have groups of categorical or numerical variables: use multiple factor analysis (prince.MFA)
* You have both categorical and numerical variables: use factor analysis of mixed data (prince.FAMD)

## CATBOOST

“CatBoost” name comes from two words “**Cat**egory” and “**Boost**ing”.

* **Performance:**CatBoost provides state of the art results and it is competitive with any leading machine learning algorithm on the performance front.
* **Handling Categorical features automatically:**We can use CatBoost without any explicit pre-processing to convert categories into numbers. CatBoost converts categorical values into numbers using various statistics on combinations of categorical features and combinations of categorical and numerical features. You can read more about it [here](https://tech.yandex.com/catboost/doc/dg/concepts/algorithm-main-stages_cat-to-numberic-docpage/).
* **Robust:**It reduces the need for extensive hyper-parameter tuning and lower the chances of overfitting also which leads to more generalized models. Although, CatBoost has multiple parameters to tune and it contains parameters like the number of trees, learning rate, regularization, tree depth, fold size, bagging temperature and others. You can read about all these parameters [here](https://tech.yandex.com/catboost/doc/dg/concepts/parameter-tuning-docpage/).
* Easy-to-use: You can use CatBoost from the command line, using an user-friendly API for both Python and

HYPEROPT

PENDULUM

CATBOOST

DASK

SWIFTER

SHAP

LIME

ELi5

<https://gdcoder.com/speed-up-pandas-apply-function-using-dask-or-swifter-tutorial/>