**Mathematic Functions**

1. **NumPy**

Numerical computing library for handling large, multi-dimensional arrays and matrices, along with mathematical functions.

**Reference**

<https://www.prepbytes.com/blog/python/numpy-functions-in-python/>

1. **Theano**

* Theano is a numerical computation library specifically designed for deep learning and machine learning tasks
* It automatically optimizes CPU or GPU computations, making it well-suited for training complex neural networks.

**Reference**

<https://jupyter.brynmawr.edu/services/public/dblank/Experiments/Introduction%20to%20Theano.ipynb>

**Data Manipulation and Analysis**

Cleaning, transforming, and restructuring of raw data.

1. **Pandas**

Data manipulation and analysis library, providing data structures like DataFrame for easy data handling.

**Features**

* Handling Missing Data (methods for filling in or dropping missing values)
* Data Merging and Joining
* Grouping and Aggregation
* Time Series Data (date range generation and frequency conversion)
* I/O Functions (can read data CSV, Excel, SQL, HDF5, etc.)
* Statistical Functions

**Reference**

<https://pandas.pydata.org/docs/user_guide/>

1. **Matplotlib**

A 2D plotting library for creating static, interactive, and animated visualizations in Python.

**Features**

* Plotting Functions (plot(), scatter(), bar(), hist(), pie())
* Customization (colors, line styles, markers, fonts, and more)
* Saving and Exporting (PNG, PDF, SVG, etc.)
* Matplotlib gallery (<https://matplotlib.org/stable/gallery/index.html>)

1. **Seaborn**

A statistical data visualization library based on Matplotlib.

**Features**

* High-Level Interface
* Customization
* Saving and Exporting
* More type of graph compared to Matplotlib
* Seaborn gallery (<https://seaborn.pydata.org/examples/index.html>)

**Natural Language Processing (NLP)**

1. **NLTK (Natural Language Toolkit)**

A library for working with human language data.

**Features**

* Tokenization (breaking down a text into individual words or sentences)
* Stopwords (words that are often removed in text processing because they are considered irrelevant)
* Stemming and Lemmatization (reducing words to their root or base form or dictionary form)
* Named Entity Recognition (NER) (identifying named entities such as names of people, organizations, locations in text)
* Concordance and Similarity (measuring similarity between words or documents)
* Corpora (large bodies of text)

1. **TextBlob**

A simple library for processing textual data, providing tools for part-of-speech tagging, noun phrase extraction, sentiment analysis, and more.

**Features**

* Tokenization
* Sentiment Analysis [polarity score (ranging from -1 to 1) to indicate the sentiment (negative to positive)]
* Noun Phrase Extraction
* Word Inflection and Lemmatization
* Spell Checking (but not accurate)

**Pros**

Simplicity and Ease of Use. Just need short code to get the features.

**Cons**

The pre-trained models may not be as accurate or flexible as those in NLTK, especially for more specialized tasks.

**Installation**

<https://textblob.readthedocs.io/en/dev/install.html>

1. **Jieba**

* Chinese text segmentation built to be the best Python Chinese word segmentation module.
* Supports Traditional Chinese
* Supports customized dictionaries
* Accurate Mode attempts to cut the sentence into the most accurate segmentations.
* Full Mode gets all the possible words from the sentence but not accurate.

**Reference**

<https://github.com/fxsjy/jieba>

**Machine Learning**

1. **PyTorch**

**Features**

* Dynamic Computational Graph
* Tensors (similar to NumPy arrays and are used for representing multi-dimensional data)
* Neural Network Module

1. **Scikit-learn**

Provides simple and efficient tools for data analysis and modeling, including various algorithms for classification, regression, clustering, and dimensionality reduction.

* Model Selection and Evaluation
  + Model Selection
  + Hyperparameter Tuning Tools

Finding the best values for these hyperparameters to optimize the model's performance.

* + Performance Evaluation Tools

Can access the accuracy, loss, recall, and other relevant measures.

* Provides tools for building predictive models and performing data analysis.
* Provides various algorithms for building predictive models from labeled data, including linear regression, logistic regression, decision trees, random forests, support vector machines (SVMs), and neural networks.

**Reference**

<https://scikit-learn.org/stable/>

1. **TensorFlow**

* Static Computational Graph
* High-level APIs

Provides high-level APIs like Keras for easy model building and training, as well as lower-level APIs for more flexibility. Keras was integrated as the official high-level API of TensorFlow.

1. XGBoost (eXtreme Gradient Boosting)

* Gradient boosting is an ensemble learning technique that combines the predictions of multiple weak learners (typically decision trees) to create a strong predictive model.

**Deep Learning**

1. **Keras**

* Keras covers every step of the machine learning workflow, from data processing to hyperparameter tuning to deployment. It was developed with a focus on enabling fast experimentation
* Allow to serve Keras models via a web API
* High-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano.

1. **Fastai**

**Features**

* Data Preprocessing
* Data Augmentation
* Data Manipulation
* Training
* Integration with PyTorch