IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Advance Research Progress on Technical Trends and Developments of Machine Learning in Python

S. Senthil[1] , M. RajeshKumar[2]

[1]Department of Mechanical Engineering, Mailam Engineering College, Mailam.

[2]Department of Master of Computer Applications, Mailam Engineering College, Mailam.

ABSTRACT

More intelligent applications are utilizing the experiences gathered from information, having the sway on each industry and research discipline. At the center of this upheaval lie the tools and the strategies that are driving it, from preparing the monstrous heaps of information created every day to learning from and making helpful move. Deep neural networks, alongside headways in traditional machine learning, have become basic segments of artificial intelligence (AI), empowering a large number of these amazing forward leaps and bringing down the hindrance to selection. Python keeps on being the most favored language for logical computing, natural language processing, data science, deep learning and machine learning, boosting both execution and profitability by empowering the utilization of low-level libraries and clean high level APIs. This overview offers knowledge into the field of machine learning with Python, taking a visit through significant themes to distinguish some of the core hardware and programming ideal models that have empowered it. This paper tells generally utilized libraries and ideas, gathered together for holistic comparison, with the objective of instructing the pursuer and driving the field of Python machine learning forward.

Keywords: Python; machine learning; deep learning; Naïve Bayes; natural language processing (NLP); data science.

1. INTRODUCTION

Artificial intelligence (AI) acts as a vital role in the field of computer science. It focuses mainly on coding computer programs and machines capable of performing tasks that humans are naturally good at, such as understanding the natural language, speech command, and detection of images. In the mid-twentieth century, machine learning emerged as a subset of AI, focuses on analyzing and interpreting patterns and structures in data to enable learning, reasoning, and decision making outside of human interaction. In Recent days, machine learning remains deeply intertwined with AI research. Machine learning user's implementing the huge amount of algorithm based only on raw data. If any corrections are identified, the algorithm can incorporate that information to boost its future higher cognitive process.

Historically, Python is a widely used high-level programming language for general-purpose programming. Apart from being open source programing language , python may be a great object-oriented, interpreted, and interactive programing language . Python combines remarkable power with very clear syntax. Modules, classes, exception handling, effective and dynamic data types are available in Python. There are interfaces to several system calls and libraries, also on various windowing systems. New built-in modules are easily written in C or C++ (or other languages, looking on the chosen implementation). Python is additionally usable as an extension language for applications written in other languages that require easy-to-use scripting or automation interfaces. Python is widely considered because the preferred language for teaching and learning Ml (Machine Learning). Few simple reasons are: It's simple to learn. As compared to C, C++ and Java the syntax is easier and Python also consists of tons of code libraries for simple use, the information handling capacity is great, Open Source, Capability of interacting. As indicated by an ongoing KDnuggets poll that reviewed more than 1800 members for inclinations in investigation, data science, and Machine learning, Python kept up its situation at the highest point of the most generally utilized language in 2019 [1]. Today, Python is one among the foremost popular programming languages for this task and it's replaced many languages within the industry, one among the explanations is its vast collection of libraries.

The main intention for this study is to advance the pursuer with a concise prologue to the most significant points and trends that are pervasive in the present scene of machine learning in Python. Our commitment is a study of the field, summing up a portion of the critical difficulties, scientific classifications and approaches. All through this article, we mean to locate a reasonable harmony between both scholastic research and industry themes, while additionally featuring the most pertinent instruments and programming libraries. Be that as it may, this is neither intended to be a far reaching guidance nor a thorough rundown of the methodologies, inquire about, nor accessible libraries. Just simple information on Python is expected, and some familiarity with processing, statistics, and machine learning will likewise be valuable. At last, we trust that this article gives a beginning stage to additionally research and helps drive the Python machine learning people group forward.

The paper is sorted out to give an review of the significant points that spread the broadness of the field. Despite the fact that every point can be perused in detachment, the intrigued pursuer is urged to tail them all together, as it can give the extra advantage of interfacing the development of specialized difficulties to their subsequent arrangements, alongside the memorable and anticipated settings of patterns certain in the description.

Numerical Computing and Machine Learning in Python

Numerical Computing defines an area of computer science and mathematics dealing with algorithms for numerical approximations of problems from mathematical or numerical analysis. The core of the Google search engine is numerical. Google executes the world's largest matrix computation by using PageRank algorithm.

Pure Python with none numerical modules couldn't be used for numerical tasks Matlab, R and other languages are designed for. If it comes to computational problem solving, it is of greatest importance to consider the performance of algorithms, both concerning speed and data usage.It is as efficient, Python in combination with its modules NumPy, SciPy, Matplotlib and Pandas, it belongs to the top numerical programming languages. The several operations in several data structure can often be parallelized over many processing cores; libraries such as NumPy [2] and SciPy [3] utilize C/C++, Fortran, and third party BLAS implementations. Numpy is a module which gives the fundamental data structures, actualizing multi-dimensional array and matrices. Other than that the module supplies the fundamental functionalities to make and control these data structures. SciPy depends on Numpy, for example it utilizes the data structures gave by NumPy. It expands the abilities of NumPy with further helpful capacities for minimization, regression, Fourier-transformation and many others.

Enhancing Python’s Performance for Numerical Computing

There is a regular requirement for preparing a lot of information in computational science applications. Putting away information in records and crossing records with plain Python for circles prompts moderate code, particularly when contrasted and comparative code in aggregated dialects, for example, Fortran, C, or C++. Luckily, there is an expansion of Python, generally called Numerical Python, or truncated NumPy, which offers productive exhibit calculations. Numerical Python has a fixed-size, homogeneous (fixed-type), multi-dimensional exhibit type and parcels of capacities for different exhibit tasks. The outcome is a powerfully composed condition for exhibit figuring like essential Matlab. As a rule, the speed of NumPy tasks is very near what is gotten in unadulterated Fortran, C, or on the other hand C++.

There are three distinct usage of Numerical Python: Numeric, numarray, and numpy. The last is the most up to date and contains all highlights of the previous two, or more some new upgrades. It is along these lines prescribed to apply numpy. The free documentation of the old Numeric execution can be utilized somewhat for numpy programming, however there are some noteworthy changes, particularly in coding style.

```
from numpy import *
```

Creating array using NumPy

This is common methods to create array using NumPy. Array of Specified Length, Filled with Zeros.

```
>>> from numpy import *
>>> size = 3
>>> arr1 = zeros(size)              # one-dim. array of length n(size)
>>> print(arr1)                      # str(arr1) [ 0. 0. 0. 0.]
>>> arr1                            # repr(arr1)
array([ 0., 0., 0., 0.])
>>> r = c = 2
>>> arr1 = zeros((r,c,3))           # row*column*3 three-dim. array
>>> print arr1
[[[ 0. 0. 0.]
[ 0. 0. 0.]]
[[ 0. 0. 0.]
[ 0. 0. 0.]]]
```

2. DATA PREPROCESSING FOR MACHINE LEARNING IN PYTHON

Pre-processing refers to the changes applied to our information before implementing in the algorithm. Data Preprocessing is a procedure that is utilized to change over the raw data into clean data collection. In other words, at whatever point the data is assembled from various sources it is gathered in raw format which isn't attainable for the examination.

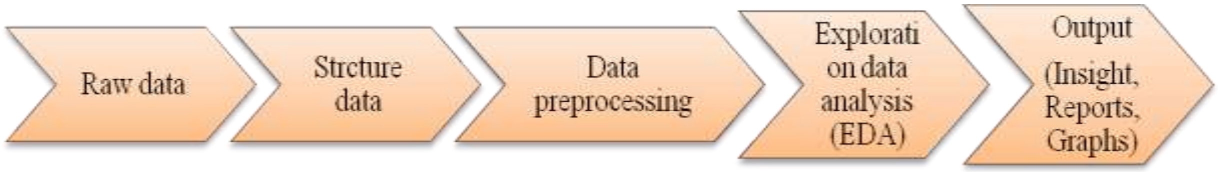
Raw data
Strcture data
Data preprocessing
Explorati on data analysis (EDA)
Output (Insight, Reports, Graphs)

Figure 1: *Methods for data preprocessing in machine learning*

Need of Data Preprocessing

For accomplishing better outcomes from the applied model in Machine Learning ventures the configuration of the data must be in an appropriate way. Some specified Machine Learning model needs data during a predetermined format, as an example, Random Forest algorithm doesn't accept null values; during this process way to execute random forest algorithm null values need to be managed from the first data set. Another viewpoint is that data index should to be designed so that more than one Machine Learning and Deep Learning algorithm are executed in one data set, and best out of them is picked.

Different data preprocessing techniques for machine learning

Rescale Data

Rescale data using scikit-learn using the MinMaxScaler class.

Required Packages

```
import pandas
import scipy
import numpy
from sklearn.preprocessing import MinMaxScaler
```

Binarize Data (Make Binary)

scikit-learn with the Binarizer class is act as vital role to Create new binary attributes in Python

To transform the data using a binary threshold.

Required Packages

```
from sklearn.preprocessing import Binarizer
import pandas
import numpy
```

Standardize Data

Standardize data using scikit-learn with the StandardScaler class.

The values for each attribute now have a mean value of 0 and a standard deviation of 1.

Required Packages

```
from sklearn.preprocessing import StandardScaler
import pandas
import numpy
```

3. NAIVE BAYES CLASSIFIER FROM SCRATCH IN PYTHON

This algorithm will works without libraries with use of implementing from scratch in Python. It is used to get probability to make predictions in machine learning. Maybe the most generally utilized model is known as the Naive Bayes algorithm. It provides a way that we can calculate the probability of a piece of data. Remembering its utilization for a framework for fitting a model to a preparation dataset, alluded to as greatest a posteriori or MAP for short, and in creating models for grouping prescient displaying issues.

For example, the Bayes Optimal Classifier and Naive Bayes. Bayes' Theorem is stated as:

P(class|data) = (P(data|class) * P(class)) / P(data)

4. NATURAL LANGUAGE PROCESSING (NLP) WITH PYTHON

The research on NLP started in early 1950s after Booth & Richens' investigation and Weaver's memorandum on machine translation in 1949. The first international conference on Machine Translation (MT) was held in 1952 and second was held in 1956. Ambiguity, by and large utilized in natural language preparing, can be refereed as the capacity of being comprehended in more than one way. In basic terms, we can say that ambiguity is the capacity of being comprehended in more than one way. Natural language is ambiguous. NLP has the some kinds of ambiguities such as Lexical ambiguity, Semantic ambiguity, Anaphoric ambiguity and Pragmatic ambiguity.

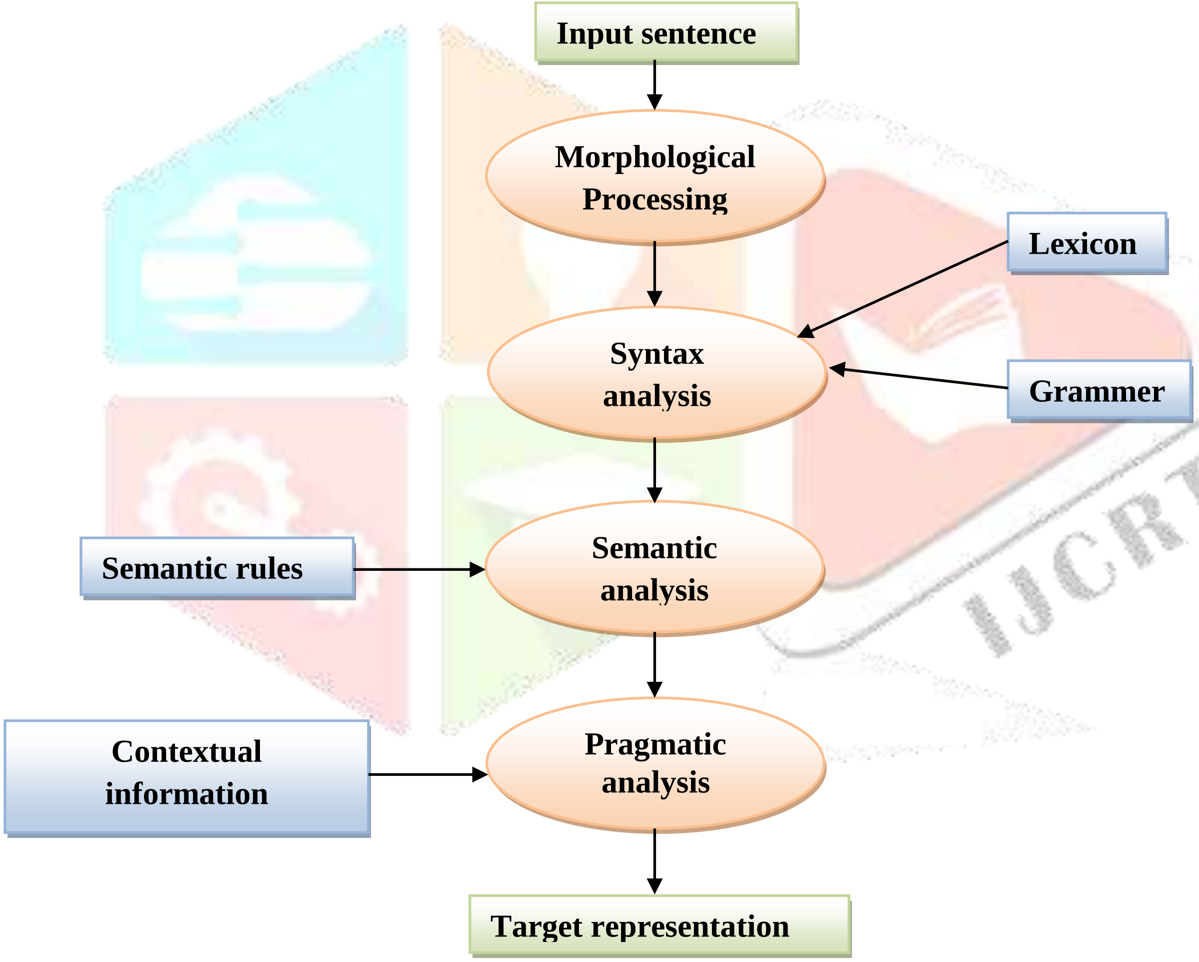

Figure 2: *The phases or logical steps in natural language processing*

Implementation of NLP in Python

Natural language processing (NLP) is a zone of computer science and artificial intelligence concerned about the collaborations among PCs and human (natural characteristic) language, specifically how to program PCs to process and break down a lot of natural language information. It is the part of AI which is tied in with examining any content and taking care of prescient analysis. The process of implementing Natural Language Toolkit (NLTK) package with Python code act as vital role in machine learning.

Steps involved in text processing and the flow of NLP.

a) Loading the Data
b) Cleaning the Data (Preprocessing)
c) Forming the Lists of Keywords
d) Splitting the keywords
e) Matching the Keywords
f) Visualizing the Results

a) Loading the Data

Import dataset with setting delimiter as '\n' or '\t' as columns of rows are separated.

```
import numpy as np              # Importing Libraries
import pandas as pd             # Importing Libraries

dataset = pd.read_csv('sampledata.tsv', delimiter = '\t')        # Import dataset
```

b) Cleaning the Data (Preprocessing)

Removing punctuations and numbers because it will simply improve the size of pack of words that we will make as last advance and decrease the efficiency of algorithm.

Stemming the keywords: *the root keyword is Learn*

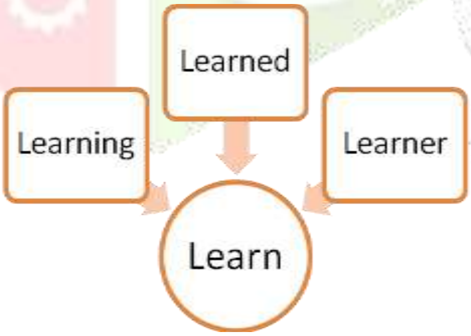

Figure 3: *Example for stemming the keywords*

Convert each word into its lower case: ('SUCCESS' into 'success')

c) Forming the Lists of Keywords

The individual keywords need to be converted into a single-word list and a multi-word list. Need to match the lists of keywords into different ways. The single-word keyword, such as "c" is referring to C programming language in imported data. But "c" is also a common letter that is used in many words including "combine", "clean". The main process is to match only a single letter "c" in the imported data.

d) Splitting the keywords

To split the keywords in imported data by using the packages of ***train_test_split***

e) Matching the Keywords

Fitting the predictive model to match the relevant keywords by using the package of ***RandomForestClassifier***

f) Final results

The final result in the form of confusion matrix or in the form of visual effect (Example: Bar chart)

5. CONCLUSIONS

This article reviewed some of the most outstanding advance trends in machine learning in Python, and also implementation of data science, scientific computing and natural language processing in Python. It gave a short foundation into significant themes, while examining the different difficulties and current condition of answers for each. There are a few increasingly specific application and research zones that are outside the extent of this study. For instance NAIVE BAYES classifier from scratch in python and implementation of NLP in python. Python is simple when contrasted with different dialects. The innovations in machine learning can be increasingly evolved and advanced with the assistance of the python. Python patterns has been giving strong difficulties to its contenders and will accomplish more noteworthy predominance in the year of 2020.

6. REFERENCES:

[1] Piatetsky, G. Python Leads the 11 Top Data Science, Machine Learning Platforms: Trends and Analysis. 2019. Available online: https://www.kdnuggets.com/2019/05/ poll-top-data-science-machine-learningplatforms.html (accessed on 1 February 2020).

[2] Oliphant, T.E. Python for scientific computing. Comput. Sci. Eng. 2007, 9, 10–20.

[3] Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.;Burovski, E.; Peterson, P.; Weckesser,W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nat. Methods 2020, 17, 261–272.

[4] Mckinney, W. pandas: A Foundational Python Library for Data Analysis and Statistics. Python High Perform.Sci. Comput. 2011, 14, 1–9.

[5] Mananmongia, shashank_v_ray, Python | NLP analysis of Restaurant reviews. Available

online:https://www.geeksforgeeks.org/python-nlp-analysis-of-restaurant-reviews/ (accessed on 1 June 2020).

[6] Mihir Mistry, Top ten python development trends for 2020. Available online : https://kodytechnolab.com/top-10-python-development-trends (accessed on 1 June 2020)

[7] Raschka, S.; Mirjalili, V. Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-learn, and TensorFlow 2; Packt Publishing Ltd.: Birmingham, UK, 2019.

[8] Wolpert, D.H. Stacked generalization. Neural Netw. 1992, 5, 241–259.

[9] Deshai, N.; Sekhar, B.V.; Venkataramana, S. MLlib: Machine learning in Apache Spark. Int. J. Recent Technol. Eng. 2019, 8, 45–49.

[10] Feurer,M.; Klein, A.; Eggensperger, K.; Springenberg, J.T.; Blum,M.; Hutter, F. Auto-sklearn: Efficient and robust automated machine learning. In AutomatedMachine Learning; Springer: Switzerland, Cham, 2019; pp. 113–134.

[11] Team, T.T.D.; Al-Rfou, R.; Alain, G.; Almahairi, A.; Angermueller, C.; Bahdanau, D.; Ballas, N.; Bastien, F.; Bayer, J.; Belikov, A.; et al. Theano: A Python framework for fast computation of mathematical expressions. arXiv 2016, arXiv:1605.02688.

[12] Raschka, S.; Kaufman, B. Machine learning and AI-based approaches for bioactive ligand discovery and GPCR-ligand recognition. arXiv 2020, arXiv:2001.06545.

[13] Berlin, Heidelberg, Python Scripting for Computational Science; Springer pp 131-188: Available online: | https://doi.org/10.1007/978-3-540-73916-6_4 (accessed on 2 June 2020)

[14] Sebastian Raschka, Joshua Patterson, Corey Nolet. Machine Learning in Python: MainDevelopments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence, Information 2020, 11, 193.

[15] Ricardo A. Calix , Sumendra B. Singh, Tingyu Chen, Dingkai Zhang and Michael Tu. Cyber Security Tool Kit (CyberSecTK): A Python Library for Machine Learning and Cyber Security. Information 2020, 11, 100.

[16] Hiroshi Kuwajima, Hirotoshi Yasuoka, Toshihiro Nakae. Engineering problems in machine learning systems. Machine learning, Springer: Available online: https://doi.org/10.1007/s10994-020-05872-w.