International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512| || Volume 9, Special Issue 2, October 2020 ||

3rd International Conference on Emerging Trends in Science, Technology and Mathematics [ICETSTM 2020]

18th September 2020

Organized by

Department of Computer Science, Parvathy's Arts and Science College, Dindigul, Tamilnadu, India

Machine Learning Algorithms Using Phython-Scikit

I.Jenifer[1], M.Vennila[2]

Assistant Professor, Department of Computer Science, Parvathy's Arts and Science College, Dindigul, Tamilnadu, India[1, 2]

ABSTRACT: Artificial intelligence (AI), as a subdivision of a field of computer science, core on designing computer programs and machines capable of performing tasks that humans are naturally good at, including natural language understanding, speech comprehension, and image recognition. Over the past decade, artificial intelligence (AI) has become a popular subject in scientific community. Today, machine learning remains deeply interlace with AI research. However, ML is usually more broadly considered a scientific field that focuses on the planning of computer models and algorithms which will perform specific tasks, often involving pattern recognition, without the necessity to be explicitly programmed. AI is a field focused on automating intellectual jobs normally performed by humans, and ML and DL are specific methods of achieving this goal. These more complicated tasks are where Machine Learning and Deep Learning methods perform well. AI falls within the realm of knowledge science, and includes classical programming and machine learning (ML). Machine Learning facilitates the computers to program themselves. This very basic idea behind the concept of machine learning describes its importance in today's computing. Machine Learning contains many models and methods. Phython is one such programing language that provides a wealthy library of modules and packages for use in scientific computing and machine learning. This paper aims at compared all the 6 machine learning techniques which we have utilized and exploring the basic concepts related to machine learning and attempts to implement a few of its applications using phython.

KEYWORDS: Machine Learning, AI, Deep Learning, NumPy, Python, Scikit-Learn

I. MACHINE LEARNING TECHNIQUES

1.1 Linear Regression

Start with the thought of 'learning'. In Machine Learning, the method of learning involves finding a function that maps the inputs to the outputs. A linear relationship means you'll represent the connection between two sets of variables with a line. Many phenomena represent a linear relationship.

$$y = mx + b$$

Where: "m" is that the slope of the road, "x" is any point (an input or x-value) on the road, and "b" is where the road crosses the y-axis. In linear relationships, any given change in an experimental variable produces a corresponding change within the variable. Rectilinear regression is employed in predicting many problems like sales forecasting, analyzing customer behavior etc. The rectilinear regression model aims to seek out a relationship between one or more features (independent variables) and endless target variable (dependent variable).

1.2 Linear Discriminant Analysis

Linear Discriminant Analysis is most generally utilized as dimensionality decrease procedure in the pre-handling venture for design grouping and AI applications. The objective is to extend a dataset onto a lower-dimensional space with great class-distinguishable all together abstain from over fitting and furthermore decrease computational expenses. In this way, basically, frequently the objective of a LDA is to extend a component space (a dataset n-dimensional examples) onto a littler subspace kk (where $k \leq n-1 k \leq n-1$) while keeping up the class-based data.

1.3 K-Nearest Neighbors

K-Nearest Neighbor is one of the most straightforward Machine Learning calculations dependent on Supervised Learning strategy. K-NN calculation accept the comparability between the new case/information and accessible cases and put the new case into the class that is generally like the accessible classifications. K-NN calculation stores all the accessible information and groups another information point dependent on the closeness. This implies when new

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512| || Volume 9, Special Issue 2, October 2020 ||

3rd International Conference on Emerging Trends in Science, Technology and Mathematics [ICETSTM 2020]

18th September 2020

Organized by

Department of Computer Science, Parvathy's Arts and Science College, Dindigul, Tamilnadu, India

information shows up then it tends to be handily characterized into a well suite classification by utilizing K-NN calculation. K-NN calculation can be utilized for Regression just as for Classification yet generally it is utilized for the Classification issues. K-NN is a non-parametric calculation, which implies it doesn't make any suspicion on fundamental information. It is likewise called an apathetic student calculation since it doesn't gain from the preparation set quickly rather it stores the dataset and at the hour of characterization, it plays out an activity on the dataset. KNN calculation at the preparation stage just stores the dataset and when it gets new information, at that point it arranges that information into a classification that is a lot of like the new information.

1.4 Classification and Regression Trees

A Classification and Regression Tree(CART) is a prescient calculation utilized in AI. It clarifies how an objective variable's qualities can be anticipated dependent on different qualities. It is a choice tree where each fork is a part in an indicator variable and every hub toward the end has an expectation for the objective variable. The CART calculation is a significant choice tree calculation that lies at the establishment of AI. Besides, it is additionally the reason for other incredible AI calculations like sacked choice trees, irregular woodland and helped choice trees.

1.5 Naive Bayes

Naive Bayes algorithm is managed learning problems dependent on applying Bayes' hypothesis with the "guileless" supposition of tending autonomy between each pair of highlights given the computation of the class variable. Bayes' hypothesis expresses the accompanying relationship, given class variable y and ward include vector x1 through xn:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n \mid y)}{P(x_1, \ldots, x_n)}$$

1.6 Support Vector Machine:

A Support Vector Machine performs classification by making an N dimensional hyper plane that optimally separates the info into two categories. In fact, a SVM model employing a sigma kernel function is kind of a two layer, perception neural network. Within the parlance of SVM literature, a variable is understood as an attribute, and a transformed attribute that's wont to define the hyper plane is understood as a feature. The task of selecting the foremost suitable representation is understood as feature selection. A group of features that describes one case (i.e., a row of predictor values) is named a vector. Therefore the goal of SVM modeling is to seek out the optimal hyper plane that separates clusters of vector in such how that case with one category of the target variable are on one side of the plane and cases with the opposite category are on the opposite size of the plane.

II. PYTHON TECHNOLOGIES

- **NumPy** - Numerical Python register a quick, conservative, multidimensional cluster productivity to Python. NumPy is the replacement to both Numeric and Num-cluster.
  - Deprecated: Numeric - Numerical Python figures a quick, smaller, multidimensional cluster language office to Python.
  - Deprecated: NumArray – Num-cluster is an implantation of Numeric which adds the capacity to proficiently performs gigantic numeric exhibits in manners coordinating to Matlab and IDL.
- **SciPy** - It is an open source library of logical apparatuses for Python. It informative supplement the well known NumPy module, assembling an assortment of elevated level science and building modules all together bundle. SciPy incorporates modules for straight variable based math, streamlining, reconciliation, exceptional capacities, sign and picture preparing, insights, hereditary calculations, ODE solvers, and others.

- **Numba** - Numba is an open source, NumPy-mindful Python compiler explicitly fit to logical codes.
- **Ad-** - promotion is an open-source Python bundle for straightforwardly performing first-and second-request programmed separation computations with any of the base numeric sorts (int, glide, complex, and so on.). Utility capacities intended for working with SciPy streamlining schedules.

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512| || Volume 9, Special Issue 2, October 2020 ||

3rd International Conference on Emerging Trends in Science, Technology and Mathematics [ICETSTM 2020]

18th September 2020

Organized by

Department of Computer Science, Parvathy's Arts and Science College, Dindigul, Tamilnadu, India

- **APM Python** - APMonitor is a nonlinear programming and improvement condition with an interface to Python. The product is accessible as a web-administration through Python libraries for the arrangement of enormous scope numerical programming issues.
- **SymPy** - SymPy is a representative control bundle, written in unadulterated Python. Its point is to turn into a full included CAS in Python, while keeping the code as straightforward as conceivable so as to be intelligible and effectively extensible.
- **ALGLIB** - mathematical examination library in C++ and C#, with Python and IronPython interfaces.
- **Python Data Analysis Library** - pandas is a library giving elite, simple to-utilize information structures and information examination devices for the Python.
- **PyGSL** - This task gives a python interface to the GNU logical library (gsl).
- **FuncDesigner** - FuncDesigner is Python module to quickly manufacture works and get their subordinates by means of programmed separation. Additionally you can perform joining, introduction, span examination, vulnerability investigation, take care of eigenvalue issues, frameworks of direct/non-straight/ODE conditions and mathematical advancement issues coded in FuncDesigner by OpenOpt.
- **escript** - escript is a Python module to characterize and explain coupled, non-straight, time-subordinate fractional differential conditions (PDEs). The client needs to actualize significant level time incorporation plans and cycle plans to lessen the issue to the arrangement of consistent, direct frameworks of PDEs which are comprehended by a reasonable PDE solver library. The current rendition utilizes the FEM solver library finley yet the plan is open and different libraries can be utilized. escript is parallelized for OpenMP and MPI. It is viable with NumPy and VTK for perception.
- **scikit-learn** - AI and information mining in Python, utilizing NumPy and SciPy
- **mlpy** - Machine Learning PYthon - elite Python module for Predictive Modeling.
- **graph-apparatus** - A python module for productive investigation of diagrams (otherwise known as. networks), with calculations actualized in C++ with the Boost Graph Library.
- **sppy** - A scanty framework bundle dependent on Eigen.

III. IMPLEMENTATION USING PYTHON

In this paper we compared six algorithms such as LR, LDA, KNN, CART, NB and SVM. The goal of this paper is to predict the high accuracy, besides high precession and recall metrics. Although this metrics are used more often in the field of health care industry like diseases identification, diagnosis in medical imaging. We used heart diseases sample dataset for our comparison of machine learning algorithm.

CODING:

```
import pandas
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
data=pd.read_csv("../input/heart.csv")
n = ['age','sex','cp','trestbps','chol','fbs','restecg','thalach']
df = pandas.read_csv(data, n=n)
arr = df.values
X = arr[:,0:9]
Y = arr[:,9]
s = 8
mod = []
mod.append(('LR', LogisticRegression()))
mod.append(('LDA', LinearDiscriminantAnalysis()))
mod.append(('KNN', KNeighborsClassifier()))
```

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512| || Volume 9, Special Issue 2, October 2020 ||

3rd International Conference on Emerging Trends in Science, Technology and Mathematics [ICETSTM 2020]

18th September 2020

Organized by

Department of Computer Science, Parvathy's Arts and Science College, Dindigul, Tamilnadu, India

```
mod.append(('CART', DecisionTreeClassifier()))
mod.append(('NB', GaussianNB()))
mod.append(('SVM', SVC()))
r = []
n = []
score = 'accuracy'
for n, mod in mods:
        kfd = model_selection.KFold(n_splits=10, random_state=seed)
        cv_results = model_selection.cross_val_score(model, X, Y, cv=kfold, score=score)
        results.append(cv_results)
        n.append(n)
        msg1 = "%s: %f (%f)" % (n, cv_results.mean(), cv_results.std())
        print(msg1)
fig1 = plt.figure()
fig1.suptitle('MLAlgorithm Comparison')
ax1 = fig1.add_subplot(111)
plt1.boxplot(r)
ax1.set_xticklabels(n)
plt1.show()
```

IV. MACHINE LEARNING ALGORITHM COMPARISON

ALGORITHMS	Accuracy	Time
LR	0.76	0.03 Secs
LDA	0.77	0.02 Secs
KNN	0.72	0.06 Secs
CART	0.69	0.08 Secs
NB	0.75	0.05 Secs
SVM	0.65	0.10 Secs

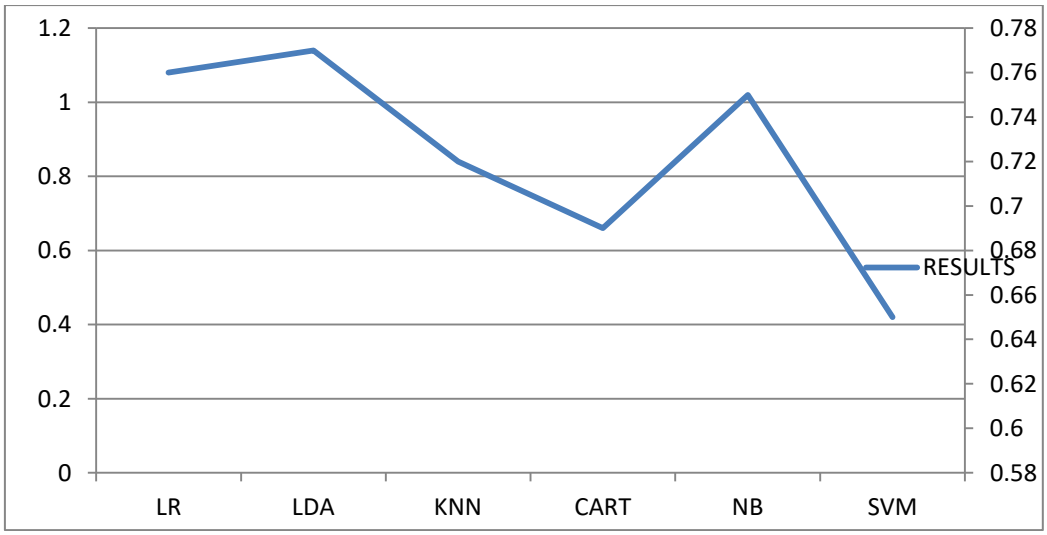

Fig.1 COMPARISION OF ML ALGORITHMS

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512| || Volume 9, Special Issue 2, October 2020 ||

3rd International Conference on Emerging Trends in Science, Technology and Mathematics [ICETSTM 2020]

18th September 2020

Organized by

Department of Computer Science, Parvathy's Arts and Science College, Dindigul, Tamilnadu, India

In this comparision of each ML algorithms, Linear Regression takes 0.03secs and accuracy rate 0.76. LDA takes 0.02secs and accuracy rate 0.77. Decision Tree Classifier 0.08 secs and accuracy rate 0.69. K-NN takes 0,06secs and accuracy rate 0.72. Naive Bayes0.05secs and accuracy rate 0.75. Support Vector Machine 0.10secs and accuracy rate 0.65.

V. CONCLUSION

In this paper we discovered the way to evaluate multiple machine learning algorithms on a heart diseases dataset using Python with scikit-learn. And also we've discussed a number of effective techniques of machine learning which will be used for classification and therefore the accuracy of classification techniques is evaluated supported the chosen algorithm. a crucial challenge in machine learning areas is to create precise and computationally efficient result. therein result we predict LDA is that the best algorithm among these six. The performance of LDA shows high level compare with other algorithms. we discover that better differentiation of algorithms are often obtained by examining computational performance metrics like accuracy and time period. In comparing the accuracy, we discover that LDA is in a position to spot network flows faster than the remaining algorithms. We found SVM to possess the slowest accuracy rate followed by LR, LDA, KNN, CART, NB. As this paper represents a preliminary investigation, there are variety of potential avenues for further work, like an in-depth evaluation of on why different algorithms exhibit different classification accuracy and computational performance. during a wider context, investigating the robustness of ML classification (for instance training on a knowledge from one location and classifying data from other locations) and a comparison between ML and non-ML techniques on a uniform dataset would even be valuable. we might also wish to explore different methods for sampling and constructing training datasets.

REFERENCES

[1]. Pinky Sodhi, Naman Awasthi, Vishal Sharma," Introduction to Machine Learning and Its Basic Application in Python".
[2]. Sebastian Raschka, Joshua Patterson , and Corey Nolet , **"**Machine Learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence".
[3]. Rene Y. Choi1,Aaron S. Coyner2,Jayashree Kalpathy-Cramer3, Michael F. Chiang and J. Peter Campbel "Introduction to Machine Learning, Neural Networks, and Deep Learning".
[4]. Witten H.I., Frank E., Data Mining: Practical Machine Learning Tools and Techniques, Second edition, Morgan Kaufmann Publishers, 2005.
[5]. Ian H. Witten and Eibe Frank. Data Mining: Practical machine learning tools and techniques, 2nd Edition. San Fransisco:Morgan Kaufmann; 2005
[6]. Luengo J, Garca S, Herrera F (2012) On the choice of the best imputation methods for missing values considering three groups of classification methods.Knowledge and Information Systems 32: 77-108.
[7]. Li D, Deogun J, Spaulding W, Shuart B (2004) Towards missing data imputation: A study of fuzzy k-means clustering method. In: Tsumoto S, Sowiski R, Komorowski J, Grzymaa-Busse J (eds.) Rough Sets and Current Trends in Computing. Springer Berlin Heidelberg 3066: 573-579.