

AUTO ML TOOL FOR SUPERVISED MACHINE LEARNING DATA

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering

by

KADIMISETTI PRASANTH (Reg. No. 38110220)

DESAMSETTI SAI PAVAN SAKETH (Reg. No. 38110124)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **DESAMSETTI SAI PAVAN SAKETH (38110124)**, **KADIMISETTI PRASANTH (38110220)** who carried out the project entitled "**AUTO-ML TOOL FOR SUPERVISED MACHINE LEARNING DATA**" under my supervision from November 2021 to April 2022.

Internal Guide

DR. ASHA.P, M.E., PH.D.,

Head of the Department

Dr. S. Vigneshwari, M.E., Ph.D.

Dr. L. Lakshmanan, M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **KADIMISETTI PRASANTH (Reg. No. 38110220)** and **DESAMSETTI SAI PAVAN SAKETH (Reg. No. 38110124)** hereby declare that the project report entitled “**AUTO ML TOOL FOR SUPERVISED MACHINE LEARNING DATA**” done by me under the guidance of **Dr. ASHA. P, M.E., Ph.D., (Internal)** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE: 20-03-2022

K. Prasanth

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing, Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Head of the Department, Dept. of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. P. Asha, M.E., Ph.D., for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Auto-ML sets out as connection joining the different levels of competence when building Machine learning pipelines or systems and achieve the data science processes more quickly. We present the common Automl tool which works on cleaned datasets using normal Ready-made algorithms provided by sklearn to run against regression and classification datasets and we also use the open source Automl libraries like auto-sklearn,hyperpot,Tpot etc and found that TPOT is best suitable for the regression datasets and auto-sklearn is best suitable for the classification datasets .The auto sklearn library is basically used for running the auto-ml . By using the scikit-learn machine learning library we can train the high level of machine learning algorithms and can find best accuracy easily for the given dataset.

TABLE OF CONTENTS

	ABSTRACT	v
	LIST OF FIGURES	vii
CHAPTER NO.	TITLE	PAGE NO.
1	INTRODUCTION	1
	1.1 OBJECTIVE OF THE PROJECT	
	1.1.1 NECESSITY	1
	1.1.2 SOFTWARE DEVELOPMENT METHOD	1
	1.1.3 LAYOUT OF THE DOCUMENT	2
	1.2 OVERVIEW OF THE PROJECT	2
2	LITERATURE SURVEY	2
	2.1 LITERATURE SURVEY	3
3	METHODOLOGY	3
	3.1 PROJECT PROPOSAL	5
	3.1.1 MISSION	5
	3.1.2 GOAL	5
	3.2 SCOPE OF THE PROJECT	5
	3.3 OVERVIEW OF THE PROJECT	5
	3.4 FLOWCHART	6
4	SYSTEM DESIGN AND IMPLEMENTATION	7
	4.1 SYSTEM STUDY	7
	4.1.1 SYSTEM REQUIREMENT SPECIFICATIONS	7
	4.2 SYSTEM SPECIFICATIONS	7
	4.2.1 MACHINE LEARNING OVERVIEW	7
	4.2.2 FLASK OVERVIEW	8

4.3	PYTHON LIBRARIES NEEDED	8
4.3.1	NUMPY LIBRARY	8
4.3.2	PANDAS LIBRARY	8
4.3.3	MATPLOTLIB LIBRARY	8
4.3.4	SEABORN LIBRARY	9
4.3.5	SCIKIT LEARN LIBRARY	9
4.4	DESCRIPTION ABOUT EACH LIBRARY	9
4.4.1	HOW NUMPY CAN BE USED	9
4.4.2	HOW PANDAS CAN BE USED	10
4.5	MODULES	11
4.5.1	DATA PRE-PROCESSING	12
4.5.2	DATA VALIDATION /CLEANING /PREPARING PROCESS	13
4.5.3	EXPLORATION DATA ANALYSIS OF VISUALIZATION	14
4.5.4	COMPARING ALGORITHM WITH PREDICTION IN THE FORM OF BEST ACCURACY RESULT	14
4.5.5	ALGORITHM AND TECHNIQUES	18
4.6	DEPLOYMENT USING DJANGO	30
4.7	DETAIL EXPLANATION OF AUTO ML	32

5	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	33
6	SUMMARY AND CONCLUSION	34
6.1	SUMMARY	34
6.2	CONCLUSION	34
	REFERENCES	35
	APPENDIX	
A.	SOURCE CODE	37
B.	SCREENSHOTS	42
C.	PUBLICATION	46

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
3.1	System Architecture	6
3.2	Flow chart	6
4.2	Logistic Regression	20
4.3	Linear Regression	21
4.4	Random Forest	22
4.5	Decission Trees	24
4.6	Naïve Bayes	25
4.7	K Nearest Neighbor	26
4.8	Support Vector Classifier	27
4.9	Support Vector Regressor	28
4.10	Gradient Boosting	28
4.11	XG Boosting	29
4.12	Adaptive Boosting	30

CHAPTER-1

INTRODUCTION

1.1 OBJECTIVE OF THE PROJECT:

To build an Auto-ML tool for any cleaned dataset so that it applies machine learning algorithm and predicts the accuracy scores for all applied algorithms and also give top three result.

The recent substantial progress in machine learning has led to a growing demand for hands-free ML systems that can support developers and ML novices in efficiently creating the new ML applications. Since different datasets require different ML pipelines, this demand has given rise to the area of automated machine learning.

1.1.1 Necessity: This website helps in overcoming the time management. This Application is very easy to use. It can work accurately and very smoothly in a different scenario. It reduces the effort workload and increases efficiency in work. In aspects of time value, it is worthy.

In this website the user can easily use our auto-ml tool for choosing the best algorithm for the given supervised machine learning data. Our auto-ml tool provides the accuracy scores of all algorithms for classification/Regression data. Then it displays the best three accuracy models for the uploaded dataset. So, by using our auto-ml tool the user can easily find the best suitable model for the data

Hence even the non-coders also easily done machine learning by using our tool.

1.1.2 Software development method:

In many software applications program different methods and cases are followed such as, Waterfall model, Iterative model, Spiral model, V-model and Big Bang model. I used waterfall model in this application. I tried to use test case and case software approaches.

1.1.3 Layout of the document:

This documentation starts with formal introduction. After introduction analysis and design of the project are described. In analysis and design of the project have many parts such as project proposal, mission, goal, target audience, environment. After that design and table diagram will be found. Use cases and test cases are in chapter 2 and chapter 3 respectively. Finally this documentation finished with result and Conclusion part.

1.2 Overview of the Designed Project:

At first we take the dataset from our resource then we have to perform data-preprocessing, visualization methods for cleaning and visualizing the dataset respectively and we upload the cleaned dataset and can run all algorithms easily by clicking button for getting accuracy scores and then it will give best three accuracy scores algorithms and flask is used for user interface.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE SURVEY:

General

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a re-organization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them.

Review of Literature Survey

Title : Benchmarking Automatic Machine Learning Frameworks

Author: Adithya Balaji, Alexander Allen

Year : 2018

We test auto-sklearn, TPOT, auto_ml, and H2O's AutoML solution against a compiled set of regression and classification datasets sourced from OpenML and found that TPOT is best suitable for the regression type datasets and auto-sklearn is best suitable for the classification type datasets

Title : Efficient and Robust Automated Machine Learning

Author: Matthias Feurer, Aaron Klein, Katharina Eggensperger

Year : 2015

The demand for the machine learning has been increased due to the success of machine learning in various range of applications .For good effective, such systems need to automatically chooses the algorithm and data pre-processing steps for a new dataset and should set their respective hyper-parameters.

Title : Hyper-parameter Optimization of Machine Learning Algorithms

Author: Li Yang, Abdallah Shami

Year : 2020

Currently most of us using machine learning algorithms in different applications and in different areas. For fitting into different problems the machine learning algorithms have to do hyper-parameters must be tuned. There is an direct impact on models performances by selecting the best hyperparameter configuration for ML models. It even also requires more knowledge on ML algorithms and hyper-parameter optimization.

Title : Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-learn

Author: Brent Komer, James Bergstra, Chris Eliasmith

Year : 2014

Hyperopt-sklearn is the project which gives the auto algorithm configuration for the sklearn ML library. Following Auto-Weka, To represent a single large hyper-parameter optimization problem we can take the choice of large pre-processing modules and the choice of classifier together.

CHAPTER-3

METHODOLOGY

3.1 Project Proposal:

The project proposal is the term of documents. A project can describe the project proposal. It is the set of all plans of a project. Like, how the software works, what are the steps to complete the entire projects, and what are the software requirements and analysis for this project. In my project, I am doing all the steps and also risk and reward and other project dependencies in the project proposal.

3.1.1 Mission:

An online Web based machine learning application is very popular and well known to everyone. We can train high level custom machine learning models with minimal effort and machine learning expertise. Hence even non coder also can do machine learning easily by using our Auto-ML tool. This simple Auto-ML tool gives fast and accurate results for choosing the best model for the given dataset.

3.1.2 Goal:

The goal is to build an Auto-ML tool for choosing the best accuracy model for any given cleaned dataset.

3.2 Scope of the Project:

The scope of this paper is to implement and investigate how different supervised binary classification methods impact default prediction. The model evaluation techniques used in this project are limited to precision, sensitivity, F1-score.

3.3 Overview of the Project:

The overview of the project is to build an Auto-ML tool for choosing the best accuracy model for any given cleaned dataset. So we can do any classification and regression type projects can predict the data and displays the best three accuracy model.

3.4 Flow Chart:

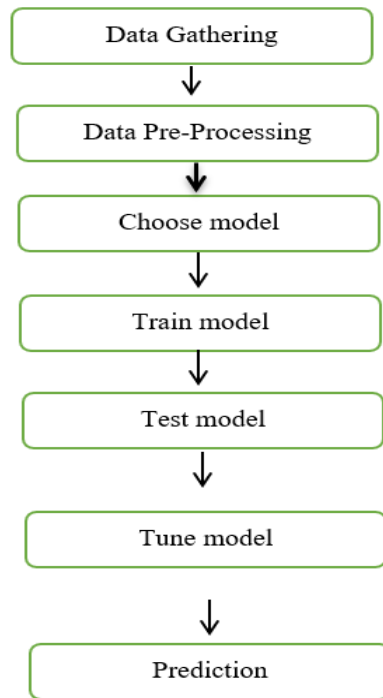


Fig:3.1: Machine Learning workflow diagram

The above flow diagram represents the flow of process from gathering data to predicting result.

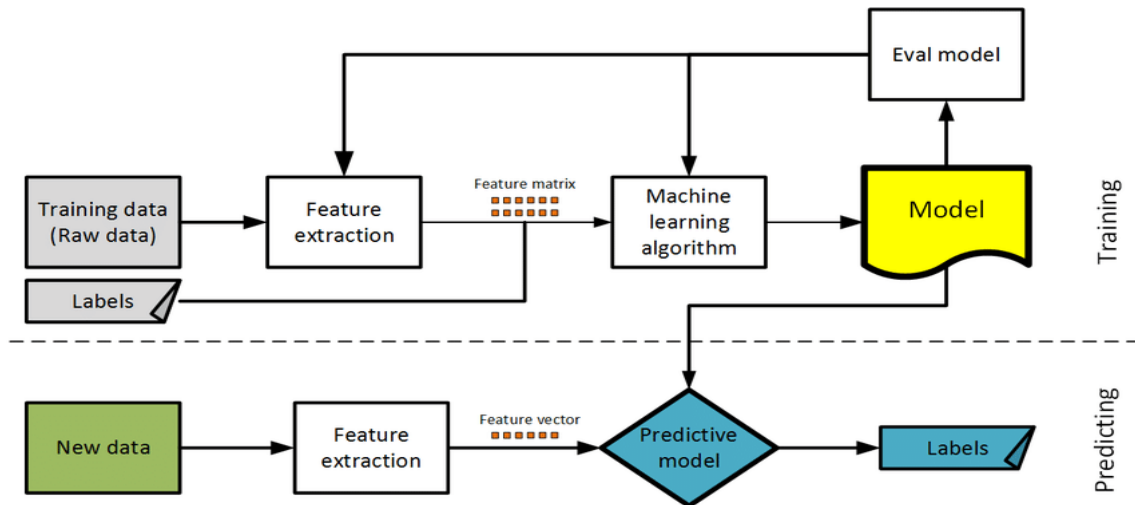


Fig:3.2: System Architecture

Clean the raw data then apply machine learning algorithms and finally predict the results.

CHAPTER-4

SYSTEM DESIGN AND IMPLEMENTATION

4.1 System Study:

To develop this model we use new modern technologies which are Machine Learning using Python for predicting and Flask is for user interface.

4.1.1 System requirement specifications:

a) Hardware requirements:

- Processor : Intel i3 or i5
- RAM : 2GB
- Hard Disk : 80GB

b) Software requirements:

- OS : Windows
- Framework : Flask
- Technology : Machine Learning using Python
- Web Browser : Chrome, Microsoft Edge
- Code editor : Visual Studio Code, Google Colab, Anaconda or Jupyter notebook.

4.2 System Specifications:

4.2.1 Machine Learning Overview:

Machine learning is a field of study that looks at using computational algorithms to turn empirical data into usable models. The machine learning field grew out of traditional statistics and artificial intelligences communities. Through their business processes immense amounts of data have been and will be collected. This has provided an opportunity to re-invigorate the statistical and computational approaches to autogenerate useful models from data. Machine learning algorithms can be used to (a) gather understanding of the cyber phenomenon that produced the data under study, (b) abstract the understanding of underlying phenomena in the form of a model, (c) predict future values of a phenomena using the above-generated model, and (d) detect

anomalous behavior exhibited by a phenomenon under observation.

4.2.2 Flask Overview:

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application.

4.3 PYTHON LIBRARIES NEEDED

There are many libraries in python. In those we only use few main libraries needed

4.3.1 NUMPY LIBRARY

NumPy is an open-source numerical Python library. NumPy contains a multi- dimensional array and matrix data structures. It can be utilized to perform a number of mathematical operations on arrays such as trigonometric, statistical, and algebraic routines like mean, mode, standard deviation etc.

4.3.2 PANDAS LIBRARY

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables. There are several ways to create a DataFrame. Here we use pandas for reading the csv files, for grouping the data, for cleaning the data using some operations

4.3.3 MATPLOTLIB LIBRARY

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Use interactive figures that can zoom, pan, update, visualize etc.,

matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB.

Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

4.3.4 SEABRON LIBRARY

Seaborn package was developed based on the Matplotlib library. It is used to create more attractive and informative statistical graphics. While seaborn is a different package, it can also be used to develop the attractiveness of matplotlib graphics.

4.3.5 SCIKIT-LEARN LIBRARY

Scikit-learn is a free machine learning library for the Python. It features various algorithms like support vector machine, random forests, regression and k-neighbours, and it also supports Py(thon numerical and scientific libraries like NumPy and SciPy.

4.4 [□] DESCRIPTION ABOUT EACH LIBRARY

□

4.4.1 *How Numpy Can Be Used*

The Numpy arrays provides a mechanism of specifying the data types of the contents. The **array** object in numpy is called **ndarray** . We can create a numpy ndarray object by using the **array()** function .While doing the data science or machine learning projects, we would often be required to carry out some statistical operations. Mainly we have numpy statistical functions numpy mean, numpy mode, numpy median and numpy standard deviation. All functions help in better understanding of data and also facilitates in deciding what actions should be taken further on data.

4.4.2 *How Pandas Can Be Used*

Pandas Data Frame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-

dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns. They can read **Csv file**-A CSV is a comma-separated values file, which allows data to be saved with a . **csv** extension. CSV files can be used with most any spreadsheet program, such as Microsoft Excel or Google Spread sheets. It is also useful in data cleaning process.

Data cleaning process:-

1. Look at your data-

- Selecting first n rows: `data.head(n)`
- Select a column: `data['movie_title']`
- Select multiple columns: `data[['budget','gross']]`

2. Deal with missing data-

- Add in a default value for the missing data
- Get rid of (delete) the rows that have missing data
- Get rid of (delete) the columns that have a high incidence of missing data

3. Add default values-

- Add the default mean values to the missing values
- Add the default median values to the missing values

4. Remove incomplete rows-

Example- `data.dropna()` , `data.dropna(how='all')` , `data.dropna(thresh=5)` etc.

4.5 Modules:

A modular design reduces complexity, facilitates change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different parts of the system. Software with effective modularity is easier to develop because functions may be compartmentalized and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements.

Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comprehensibility, Modular Understandability, Modular continuity, Modular Protection.

The following are the modules of the project, which is planned in aid to complete the project with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement

4.5.1 Data Pre-processing

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model.

A number of different data cleaning tasks using Python Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it's important to understand the sources of missing data. Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.

- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

- import libraries for access and functional purpose and read the given dataset
- General Properties of Analyzing the given dataset
- Display the given dataset in the form of data frame
- show columns
- shape of the data frame
- To describe the data frame
- Checking data type and information about dataset
- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

4.5.2 Data Validation/ Cleaning/Preparing Process

Importing the library packages with loading given datasets. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process.

The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

4.5.3 Exploration data analysis of visualization

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

- How to chart time series data with line plots and categorical quantities with bar charts.
- How to summarize data distributions with histograms and box plots.

4.5.4 Comparing Algorithm with prediction in the form of best accuracy result

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different

techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies. In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set. And another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in given dataset.

These below different algorithms are compared for Classification type data:

- Logistic Regression
- Naïve bayes
- K Nearest Neighbor
- Decision Tree
- Support Vector Machine

These below different algorithms are compared for Regression type data:

- Linear Regression
- Decision Tree
- Support Vector Regression

- Lasso Regression
- Random Forest

These below different algorithms are used for boosting techniques:

- AdaBoosting
- Gradient Tree Boosting
- XG Boosting

The K-fold cross validation procedure is used to evaluate each algorithm, importantly configured with the same random seed to ensure that the same splits to the training data are performed and that each algorithm is evaluated in precisely the same way. Before that comparing algorithm, Building a Machine Learning Model using install Scikit-Learn libraries. In this library package have to done preprocessing, linear model with logistic regression method, cross validating by KFold method, ensemble with random forest method and tree with decision tree classifier. Additionally, splitting the train set and test set. To predicting the result by comparing accuracy.

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

Prediction result by accuracy:

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. It needs the output of the algorithm to be classified variable data. Higher accuracy predicting result is logistic regression model by comparing the best accuracy.

True Positive Rate (TPR) = $TP / (TP + FN)$

False Positive Rate (FPR) = $FP / (FP + TN)$

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

Accuracy = $(TP + TN) / (TP + TN + FP + FN)$

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct.

Precision = $TP / (TP + FP)$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = $TP / (TP + FN)$

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the

all observations in actual class - yes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

General Formula:

$$F\text{- Measure} = 2TP / (2TP + FP + FN)$$

F1-Score Formula:

$$F1 \text{ Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

4.5.5 ALGORITHM AND TECHNIQUES

Algorithm Explanation

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

Used Python Packages:

sklearn:

- In python, sklearn is a machine learning package which include a lot of ML algorithms.
- Here, we are using some of its modules like train_test_split,

DecisionTreeClassifier or Logistic Regression and accuracy_score.

NumPy:

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

Pandas:

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

Matplotlib:

- Data visualization is a useful way to help with identify the patterns from given dataset.
- Data manipulation can be done easily with data frames.

Logistic Regression

It is a statistical method for analyzing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

In other words, the logistic regression model predicts $P(Y=1)$ as a function of X . Logistic regression Assumptions:

- Binary logistic regression requires the dependent variable to be binary.
- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.

- Only the meaningful variables should be included.
- The independent variables should be independent of each other. That is, the model should have little.
- The independent variables are linearly related to the log odds.
- Logistic regression requires quite large sample sizes.

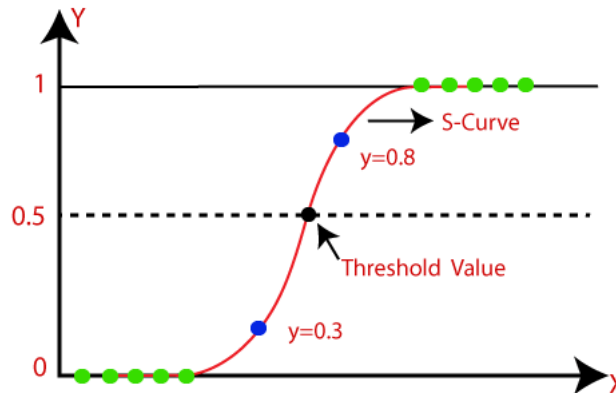


Fig:4.2: Logistic Regression

The above graph represents that the Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.

Linear Regression

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y (output).

Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

While training the model we are given :

x: input training data (univariate – one input variable(parameter))

y: labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ_1 and θ_2 values.

θ_1 : intercept

θ_2 : coefficient of x

Once we find the best θ_1 and θ_2 values, we get the best fit line. So when we are finally using our model for prediction, it will predict value of y for input value of x.

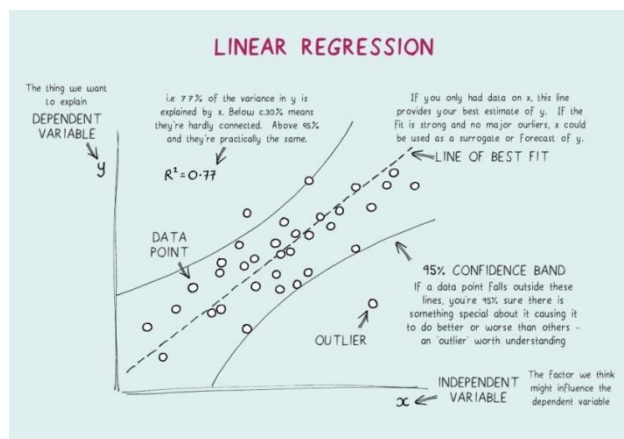


Fig:4.3:Linear Regression

Linear Regression works by using an independent variable to predict the values of dependent variable. In linear regression, a line of best fit is used to obtain an equation from the training dataset which can then be used to predict the values of the testing dataset.

Random Forest Classifier

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision

forests correct for decision trees' habit of over fitting to their training set. Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees*, resulting in a *forest of trees*, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm:

- Pick N random records from the dataset.
- Build a decision tree based on these N records.
- Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

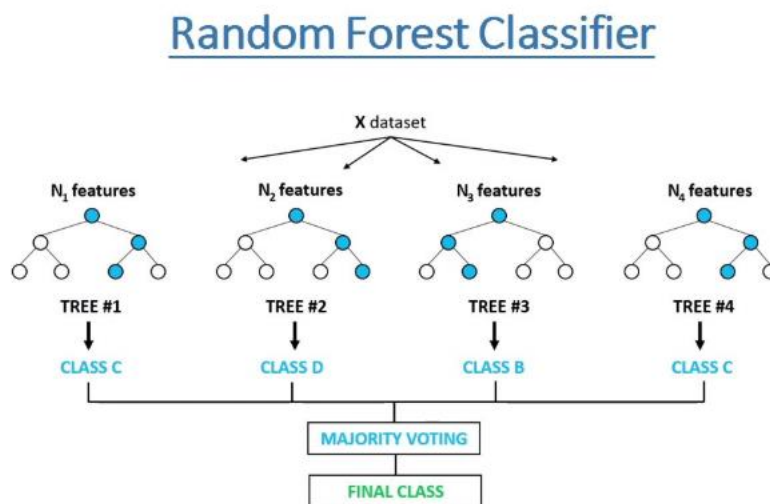


Fig:4.4: Random Forest

The above tree represents that random forest classifier contains a number of decision trees on various subsets of the given dataset and takes the average to improve the

predictive accuracy of that dataset."

Decision Tree Classifier

It is one of the most powerful and popular algorithm. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. Assumptions of Decision tree:

- At the beginning, we consider the whole training set as the root.
- Attributes are assumed to be categorical for information gain, attributes are assumed to be continuous.
- On the basis of attribute values records are distributed recursively.

We use statistical methods for ordering attributes as root or internal node.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. Decision tree builds classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed.

This process is continued on the training set until meeting a termination condition. It is constructed in a top-down recursive divide-and-conquer manner. All the attributes should be categorical. Otherwise, they should be discretized in advance. Attributes in the top of the tree have more impact towards in the classification and they are identified using the information gain concept. A decision tree can be easily over-fitted generating too many branches and may reflect anomalies due to noise or outliers.

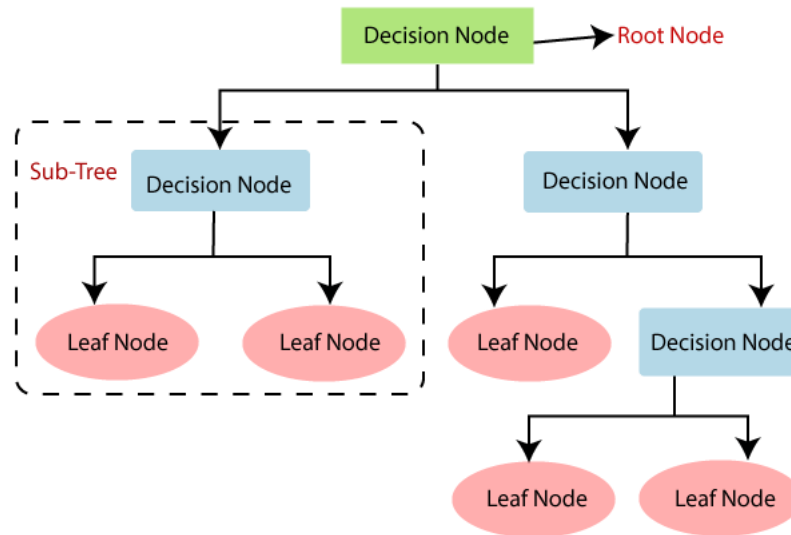


Fig:4.5:Decission Tree

The above decision tree is a graphical representation of all possible solutions to a decision based on certain conditions. On each step or node of a decision tree, used for classification, we try to form a condition on the features to separate all the labels or classes contained in the dataset to the fullest purity.

Naive Bayes algorithm:

The Naive Bayes algorithm is an intuitive method that uses the probabilities of each attribute belonging to each class to make a prediction. It is the supervised learning approach you would come up with if you wanted to model a predictive modeling problem probabilistically.

- Naive bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method.
- The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have a probability of a data instance belonging to that class. To make a prediction we can calculate probabilities of the instance belonging to each class and select the class value with the highest probability.

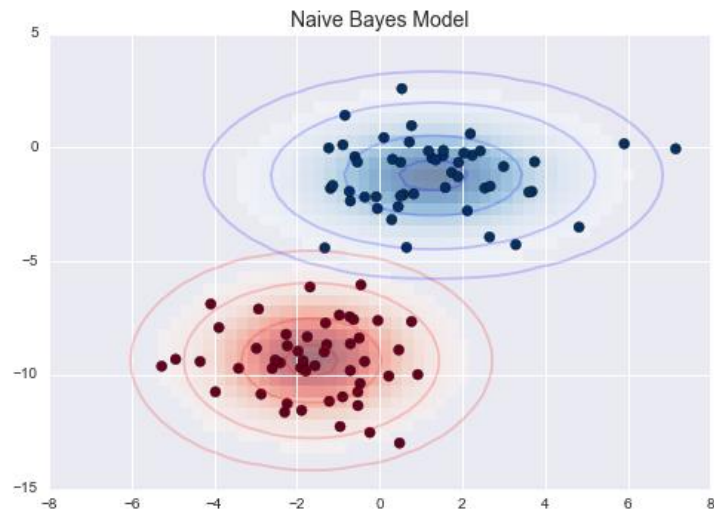


Fig:4.6: Naïve Bayes

The above diagram shows the how it predicts the membership probabilities for each class such as the probability that given every record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class.

K-Nearest Neighbor

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. It stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

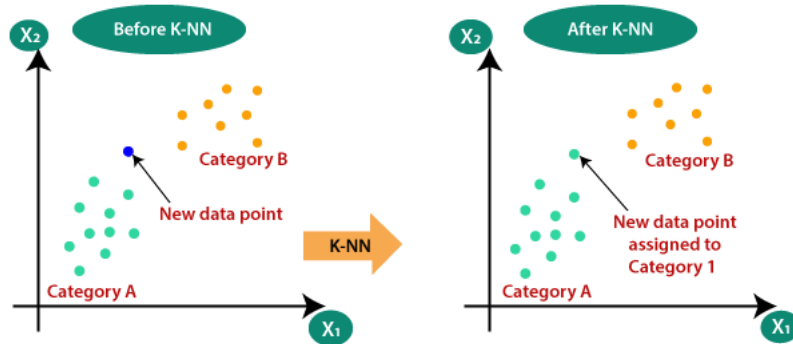


Fig:4.7:K-Nearest Neighbor

In the above graph the KNN tries to predict the class for the test data by calculating the distance between the test data and all the training points. Then it selects the no. of points which are closer to the test data

Support Vector Classifier

Support Vector Classifier or SVC is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. Primarily, it is used for Classification problems in Machine Learning. The goal of the SVC algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes. So that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVC chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

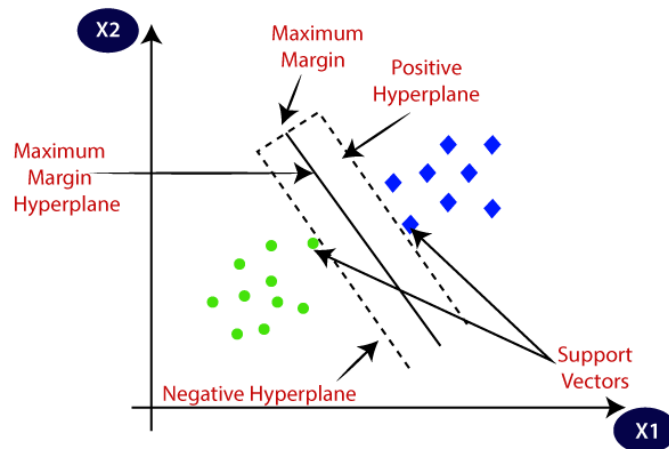


Fig:4.8: Support Vector Classifier

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we will maximize the margin of the classifier and these are the points that help us to build our svc.

Support Vector Regression (SVR)

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already requested from the problem. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated.

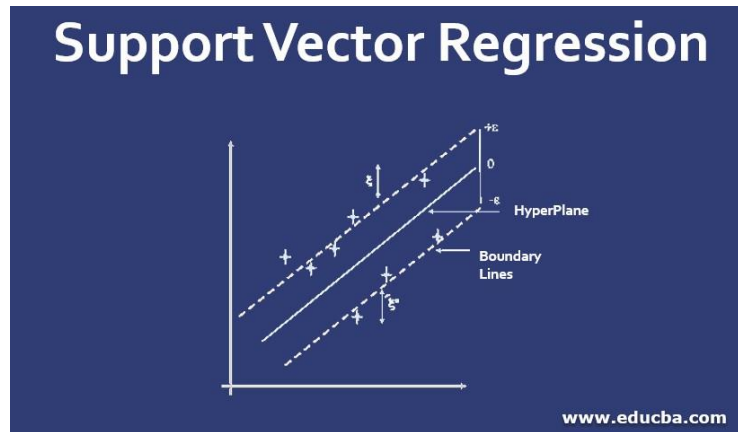


Fig:4.9:Support Vector Regression

The above graph represents the best fit line. In Support Vector Regressor the best fit line is the hyperplane that has the maximum number of points.

Gradient Boost

Gradient boosting algorithm is one of the most powerful algorithms in the field of machine learning. As we know that the errors in machine learning algorithms are broadly classified into two categories i.e., Bias Error and Variance Error. As gradient boosting is one of the boosting algorithms it is used to minimize bias error of the model.

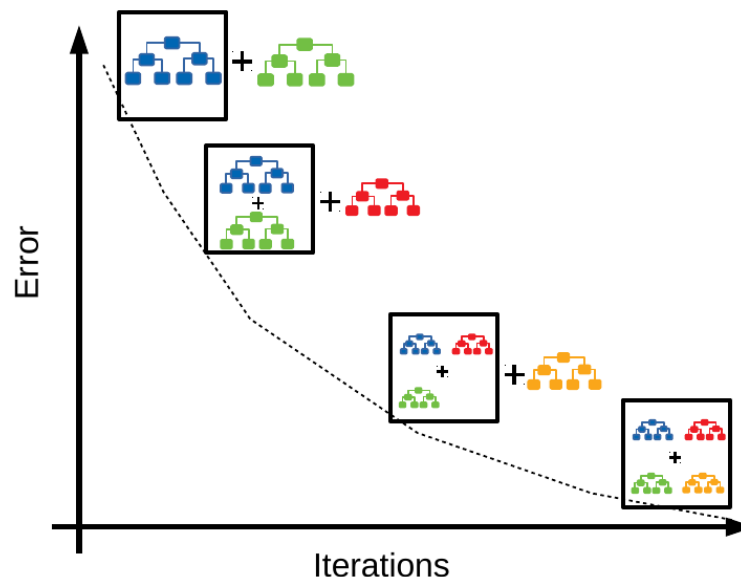


Fig:4.10:Gradient Boosting

The above algorithm can be used for predicting not only continuous target variable (as a Regressor) but also categorical target variable (as a Classifier).

XG Boost

XG Boost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now

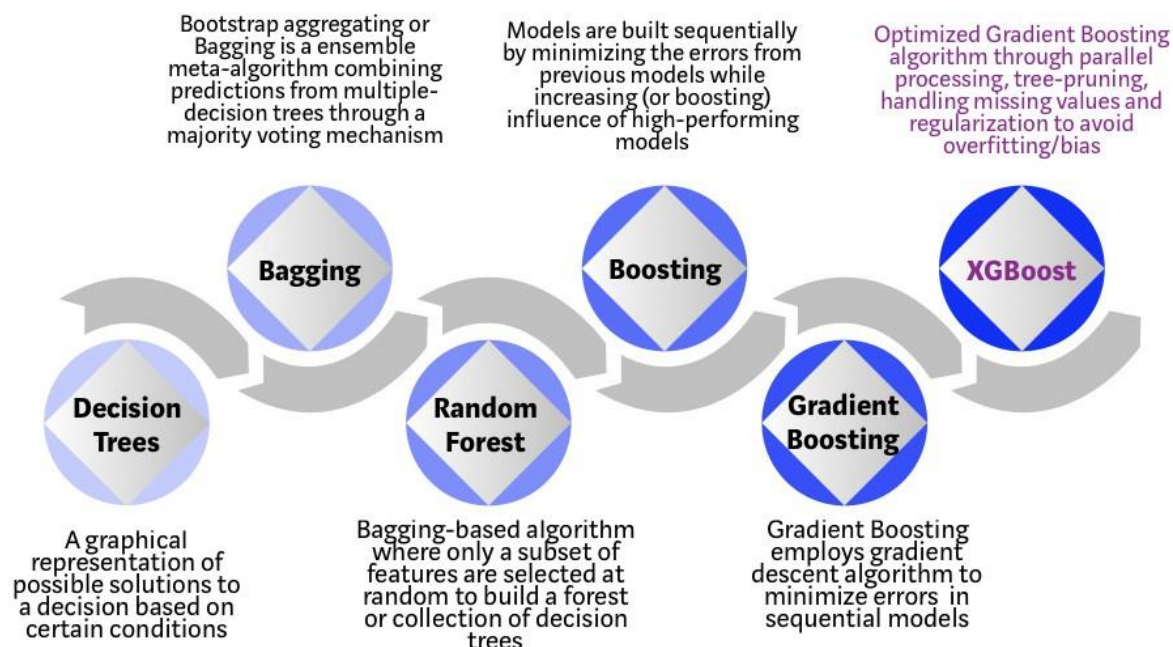


Fig:4.11: XG Boosting

The above chart diagram represents the evolution of tree-based algorithms over the years.

Adaptive Boosting

Ada-Boost also called Adaptive Boosting is a technique in Machine Learning used as an Ensemble Method. The most common algorithm used with Ada-Boost is decision trees with one level that means with Decision trees with only 1 split. These trees are also called Decision Stumps.

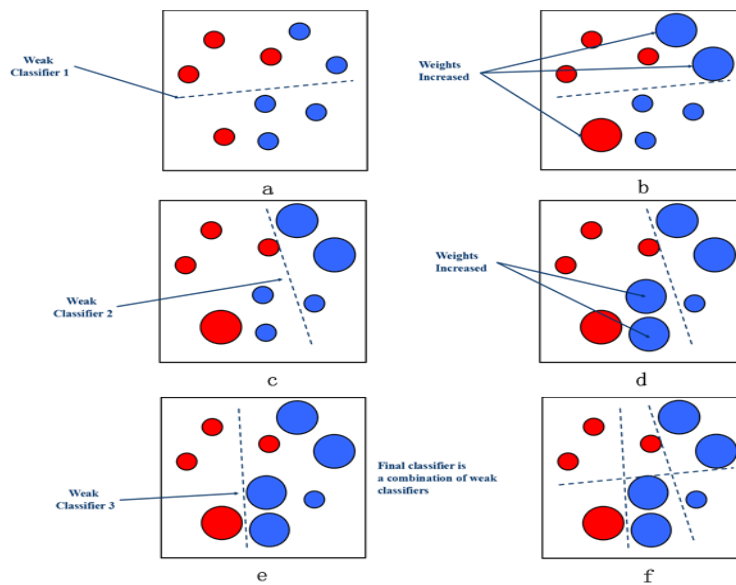


Fig:4.12: Adaptive Boosting

The above figure represents that the weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances.

4.6 Deployment Using DJANGO (Web Framework) :

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

Django helps you write software that is:

Versatile

Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is built with Django.

Internally, while it provides choices for almost any functionality you might want (e.g. several popular databases, templating engines, etc.), it can also be extended to use other components if needed.

Secure

Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.

A password hash is a fixed-length value created by sending the password through a cryptographic hash function. Django can check if an entered password is correct by running it through the hash function and comparing the output to the stored hash value. However due to the "one-way" nature of the function, even if a stored hash value is compromised it is hard for an attacker to work out the original password. Django enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery and clickjacking (see Website security for more details of such attacks).

Scalable

Django uses a component-based "shared-nothing" architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed). Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the busiest sites have successfully scaled Django to meet their demands (e.g. Instagram and Disqus, to name just two).

Maintainable

Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the Model View Controller (MVC) pattern).

Portable

Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavours of Linux, Windows, and Mac OS X. Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

4.7 DETAIL EXPLANATION OF AUTO ML

First our Auto ML Tool will ask the user to choose the machine learning type i.e., Supervised, Unsupervised and Reinforcement from the drop down menu. Currently we have done this Auto ML Tool for Only Supervised Machine learning Data. So, if we select unsupervised or reinforcement it will display a Still under Construction message. After selecting the supervised machine learning it will asks the user to upload a only cleaned dataset. After the dataset is uploaded, then it will show successfully uploaded message. Then we select the Regression or classification type based on the dataset uploaded. Then we have to select the output column and train, test data split ratio. After splitting data we can print the input (x) columns and output (y) column, x train size, y train, and test sizes in the result box by just clicking on the buttons. Then we have to choose the algorithm that we want to train and predict the data. After that it will print all the predicted values in the result box. Finally it will show r2score, mean value error, Absolute error and Root mean square error in the result box.

CHAPTER-5

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

Performance Analysis:

Website performance optimization, the main point of technologically superior website designs is the primary factor dictating success to get the fast and accurate results in predicting the results for any classification and prediction problems easily without any prior knowledge on machine learning. After all, unimpressive website performance kills admission process when the torture of waiting for slow Web pages to load frustrates visitors into seeking alternatives – impatience is a digital virtue! All classification and prediction algorithms find the accuracies easily by just clicking on those buttons and it gives fast and exceptional results to user.

We created the following six chapter in-depth speed optimization guide to show you how important it is to have a fast loading, snappy website! Countless research papers and benchmarks prove that optimizing your sites' speed is one of the most affordable and highest ROI providing investments!

Lightning-fast page load speed amplifies visitor engagement, retention, and boosts sales. Instantaneous website response leads to higher conversion rates, and every 1 second delay in page load decreases customer satisfaction by 16 percent, page views by 11 percent and conversion rates by 7 percent according to a recent Aberdeen Group research.

Discussion:

While discussions provide avenues for exploration and discovery, leading a discussion can be anxiety-producing: discussions are, by their nature, unpredictable, and require us as instructors to surrender a certain degree of control over the flow of information. Fortunately, careful planning can help us ensure that discussions are lively without being chaotic and exploratory without losing focus. When planning a discussion, it is helpful to consider not only cognitive, but also social/emotional, and physical factors that can either foster or inhibit the productive exchange of ideas.

CHAPTER-6

SUMMARY AND CONCLUSION

6.1 Summary:

To build an auto-ml tool for any cleaned dataset so that it applies machine learning algorithm and predicts the accuracy scores for all applied algorithms and also give top three result. The recent substantial progress in machine learning has led to a growing demand for hands-free ML systems that can support developers and ML novices in efficiently creating the new ML applications. Since different datasets require different ML pipelines, this demand has given rise to the area of automated machine learning.

6.2 Conclusion:

Automated Machine Learning (AutoML) supports practitioners and researchers with the tedious task of designing machine learning pipelines and has recently achieved substantial success. In this project we introduce our new AutoML tool that makes the application of few machine learning algorithms easier to cleaned dataset without having any prior knowledge in coding. Our Auto-ML tool benchmark cleaned datasets by applying basic machine learning algorithms and also by comparing our result with other existing auto ml libraries.

REFERENCES:

- [1] G. Cawley, I. Guyon, S. Escalera, K. Bennett, H. Escalante, M. Saeed, A. Statnikov, E. Viegas T. Ho, N. Macia, and B. Ray. Design of the 2015 ChaLearn AutoML Challenge. In Proc. of IJCNN'15, 2015.
- [2] M.-A. Zller and M. F. Huber, Survey and benchmark of Automated Machine Learning Frameworks, arXiv preprint arXiv:1904.12054, (2019). <https://arxiv.org/abs/1904.12054>.
- [3] M. Kuhn and K. Johnson, Springer, Applied Predictive Modeling (2013) ISBN: 9781461468493.
- [4] L. Kotthoff, J. Vanschoren, and F. Hutter, Eds., Automatic Machine Learning: Systems, Methods, Springer, Challenges (2019) ISBN: 9783030053185.
- [5] G. Luo, A review of automatic selection methods for Hyper-parameter values and machine learning algorithms Netw. Model. Anal. Heal. Informatics Bioinforma. 5 (2016) 116. <https://doi.org/10.1007/s13721-016-0125-6>.
- [6] Q. Yao et al., Taking Human out of Learning Applications: A Survey on Automated Machine Learning, arXiv preprint arXiv:1810.13306, (2018). <http://arxiv.org/abs/1810.13306>.
- [7] Nalepa, P. R. Lorenzo, L. S. Ramos, J. R. Paster and JM. Kawulok and Particle swarm optimization for hyper-parameter selection Proc. ACM Int. Conf. Gent. Evoll. Compute., (2017) 481488.
- [8] W. Hosmer Jr, S. Lemeshow, Applied logistic regression, Technometrics, 34 (1) (2013), 358-359.
- [9] R. Martinez-Cantin, BayesOpt: A Bayesian optimization library for nonlinear optimization, and bandits ,experimental design, J. Mach. Learn. Res. 15 (2015) 37353739.
- [10] M.A. Gardner, C. Gage, F.A. Fortin, F.M. De Rainville, M. Parizeau, DEAP: Evolutionary algorithms done easily, J. Mach. Learn. Res. 13 (2012) 21712175.
- [11] N.L.C. Talbot, G.C. Cawley On over-fitting in model selection and selection bias in performance evaluation. J. Mach. Learn. Res. 11 (2010) 20792107.

- [12] S. Tatikonda, M. Boehm, A. Surve, et al., SystemML: declarative machine learning, *Pro. VLDB-Endo.* 9 (2016) 14251436.
- [13] B. Yavuz, X. Meng, J. Bradley, et al., ML library: ML in apache spark, *J. Mach. Learn. Res.* 17 (2016) 1235-1241.
- [14] P. Frasconi, L. Franceschi, M. Donini, and M. Pontil, Reverse and Forward gradient-based hyperparameter optimization, *34th Int. Conf. Mach. Learn. ICML 2017*, 70 (2017) 1165-1173.
- [15] J. Bergstra, B. Komer, and C. Eliasmith, Hyperoptz-sklearn: Automatic hyperparameter configuration for scikit-learn, *Proc. ICML Workshop AutoML*, (2014) 3440.
- [16] T. Springenberg, F. Hutter T. Domhan. Extrapolating Learning Curves of Deep Neural Networks, *ICML AutoML Workshop*. 2014.
- [17] H. Hoos, K. Leyton-Brown and F. Hutter. Sequential modelbased optimization for the general algorithm configuration, *LION-5*, 2011. Extended version UBC Tech-report TR-2010-10.
- [18] G. Varoquaux, F. Pedregosa, V. Michel, A. Gramfort, O. Grisel, B. Thirion, V. Dubourg, M. Blondel, P. Prettenhofer, R. Weiss, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: ML in Python, *The Journal of ML Research*, 12:2825–2830. 2011.

APPENDIX:

A.SOURCE CODE

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X, y, test_size=0.20, random_state=20)
def heck(sa,pr,sr,pk,si):
    a=sa
    b=pr
    mod=sr
    def fit(a,b,mod):
        model=mod
        f=model.fit(a,b)
        return f
    def predict(c):
        pred=mod.predict(c)
        return pred
    def score(x,y):
        return r2_score(x,y)
    mod=fit(a,b,mod)
    y_pred=predict(pk)
    return score(si,y_pred)
while(n!=3):
    if n==1:
        print("you had selected regression");algo=input("enter your required algo name or code\n")
        if algo=="linear":
            print(heck(X_train,y_train,LinearRegression(),X_test,y_test))
        elif algo=="Random":
            print(heck(X_train,y_train,RandomForestRegressor(),X_test,y_test))
        elif algo=="all":
            print("linear",heck(X_train,y_train,LinearRegression(),X_test,y_test));print("decision",heck(X_train,y_train,DecisionTreeRegressor(),X_test,y_test));print("Random",heck(X_train,y_train,RandomForestRegressor(),X_test,y_test))
        elif algo=="exit":
            break
    elif n==2:
        print("you had selected classification");algo=input("enter your required algo name or code\n")
        elif algo=="Random":
            print(heck(X_train,y_train,RandomForestRegressor(),X_test,y_test))
```

```

from django.shortcuts import render, redirect
import pandas as pd
from sklearn.model_selection import train_test_split
from django.contrib import messages
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
def uploadfile(request):
    if "GET" == request.method:
        return render(request, 'index.html', {})
    else:
        x_axis=[]
        upload_file = request.FILES["upload_file"]
        try:
            df = pd.read_excel(upload_file)
        except:
            df = pd.read_csv(upload_file)
        df = df.to_json()
        request.session['df']=df
        problemType= True
        messages.success(request,"Successfully Uploaded file")
        messages.success(request,"Successfully Converted to DataFrame")
        return render(request,
'index.html',context={'problemType':problemType})
def xtrain(request):
    if 'df' in request.session and 'y-axis' in request.session:
        data=request.session['df']
        y_axis=request.session['y-axis']
        testsize=request.session['test_size']
        df = pd.read_json(data)
        X=df.drop(y_axis,axis='columns')
        y=df[y_axis]
        X_train,X_test,y_train,y_test = train_test_split(X, y,
test_size=testsize, random_state=20)
        showbuttons=True
        return
    render(request, 'index.html',{"xtrain":X_train.size, 'showbuttons':showbuttons})
def algoTrain(request):
    predictval=None
    skip=True
    if 'train' in request.POST:
        data=request.session['df']
        y_axis=request.session['y-axis']
        testsize=request.session['test_size']
        df = pd.read_json(data)

```

```

X=df.drop(y_axis,axis='columns')
y=df[y_axis]
X_train,X_test,y_train,y_test = train_test_split(X, y,
test_size=testsize, random_state=20)
algo=request.POST.get('algoval')
messages.info(request,"Algorithm is Getting Trained")
if algo=="linear":
    linear=LinearRegression()
    linear.fit(X_train,y_train)
    pred=linear.predict(X_test)
elif algo=="random":
    random=RandomForestRegressor()
    random.fit(X_train,y_train)
    pred=random.predict(X_test)
elif algo=="decision":
    decision=DecisionTreeRegressor()
    decision.fit(X_train,y_train)
    pred=decision.predict(X_test)
if "predict" in request.POST:
    checkMetrics=True
    skip=False
    data=request.session['df']
    y_axis=request.session['y-axis']
    testsize=request.session['test_size']
    df = pd.read_json(data)
    X=df.drop(y_axis,axis='columns')
    y=df[y_axis]
    X_train,X_test,y_train,y_test = train_test_split(X, y,
test_size=testsize, random_state=20)
    algo=request.POST.get('algoval')
    request.session['algo']=algo
    if algo=="linear":
        linear=LinearRegression()
        linear.fit(X_train,y_train)
        pred=linear.predict(X_test)
    elif algo=="random":
        random=RandomForestRegressor()
        random.fit(X_train,y_train)
        pred=random.predict(X_test)
    if "rmse" in request.POST:
        rmse=mean_squared_error(y_test, predictval, squared=False)
        return
render(request,'index.html',{'checkMetrics':checkMetrics,'rmse':rmse})

if "mse" in request.POST:

```

```

        mse=mean_squared_error(y_test, predictval)
        return
render(request,'index.html',{'checkMetrics':checkMetrics,'mse':mse})

    if "mae" in request.POST:
        mae=mean_absolute_error(y_test, predictval)
        return
render(request,'index.html',{'checkMetrics':checkMetrics,'mae':mae})

    if "r2score" in request.POST:
        r2score=r2_score(y_test, predictval)
        return
render(request,'index.html',{'checkMetrics':checkMetrics,'r2score':r2score})

```

urls.py

```

from django.urls import path
from . import views

urlpatterns = [
    path('', views.selectML, name='selectML'),
    path('uploadfile/',views.uploadfile,name='uploadfile'),
    path('problemtype/',views.problemType, name="problemType"),
    path('regression/',views.regression, name="regression"),
    path('classification/',views.classification, name="classification"),
    path('xcolumns/',views.xcolumns,name="xcolumns"),
    path('ycolumns/',views.ycolumns,name="ycolumns"),
    path('xtest/',views.xtest,name="xtest"),
    path('xtrain/',views.xtrain,name="xtrain"),
    path('ytest/',views.ytest,name="ytest"),
    path('ytrain/',views.ytrain,name="ytrain"),
    path('skip/',views.skip,name="skip"),
    path('trainalgo/',views.algoTrain,name="algoTrain"),
    path('checkmetrics/',views.checkMetrics,name="checkMetrics"),
]

```

Html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>TechSnap Auto ML</title>

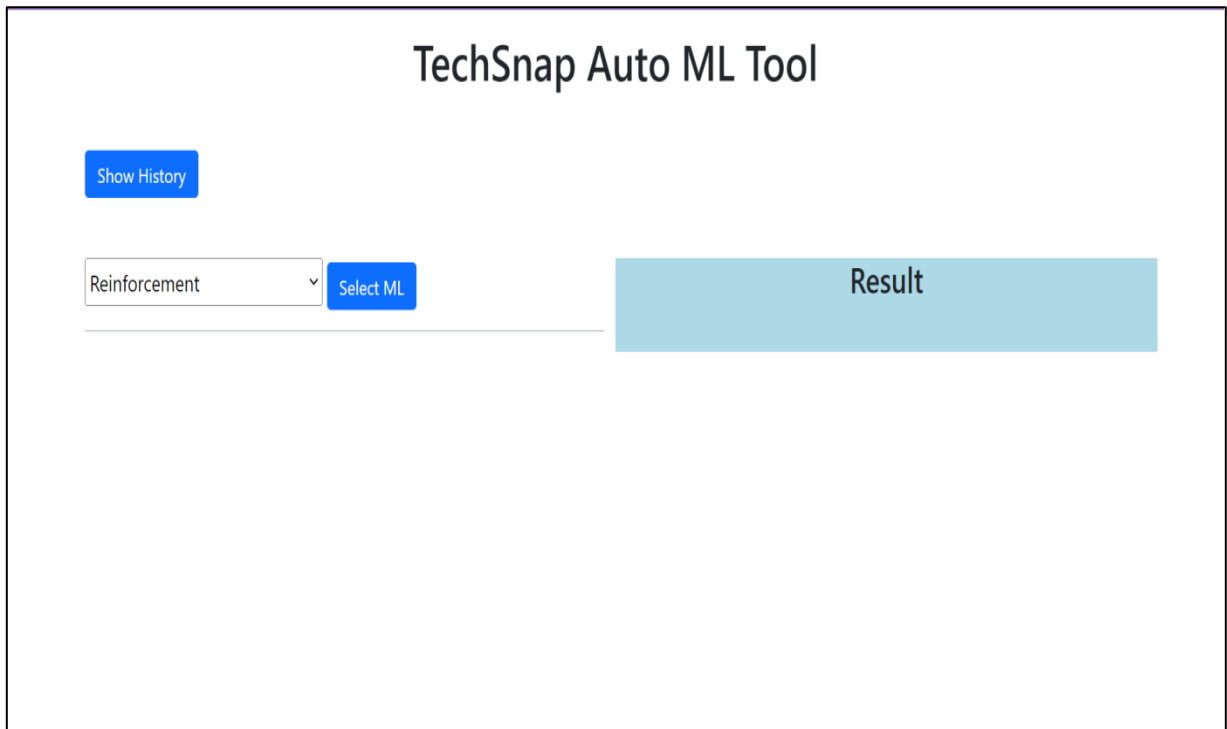
```

```

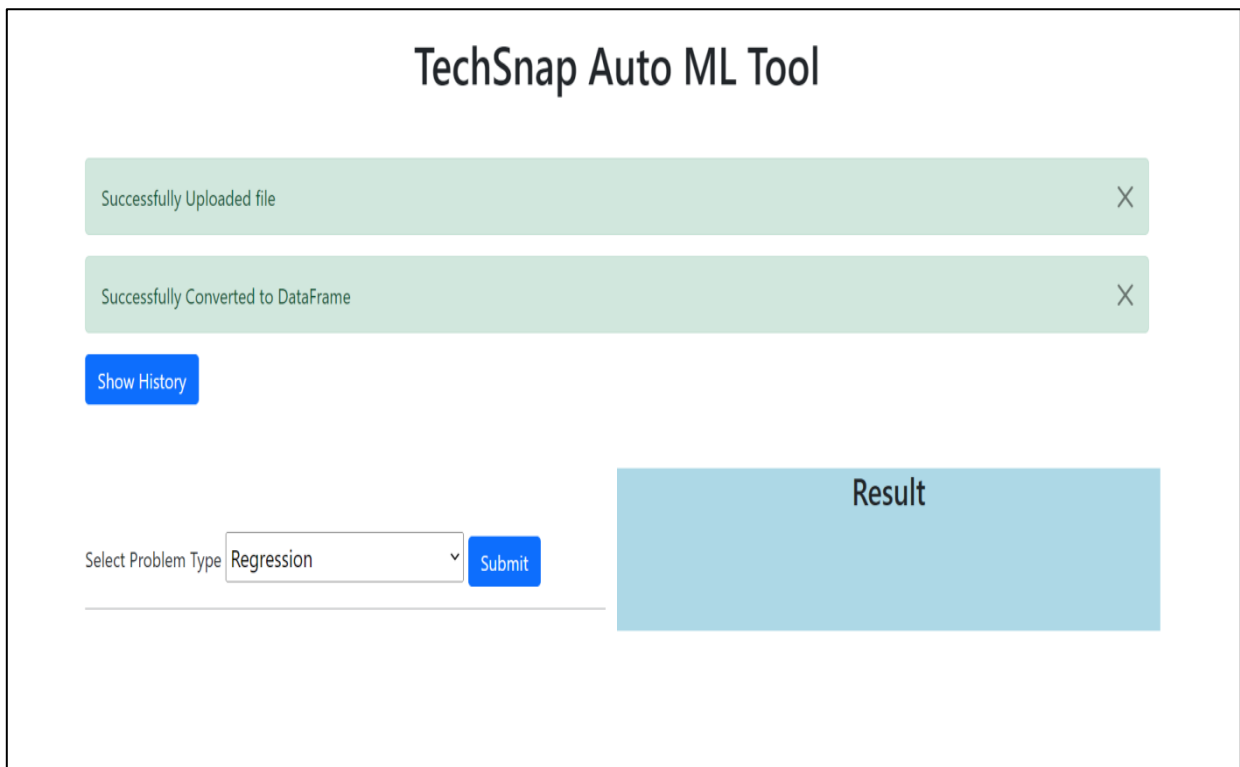
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFtdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js
" integrity="sha384-
7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31e0z1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-
QJHtvGhmr9X0IpI6YVutG+2Q0K9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"
crossorigin="anonymous"></script>
<style>
{% elif uploadfile %}
        <form class="mt-5" action="{% url 'uploadfile' %}"
method="post" enctype="multipart/form-data">
            {% csrf_token %}
            <div class="mb-3">
                <label for="formFile" class="form-label">Upload
Dataset</label><input class="form-control" type="file" name="upload_file"
id="formFile" required></div><input type="submit" class="btn btn-primary"
value="Upload"></form>
<form class="mt-5" action="{% url 'algoTrain' %}" method="post"
enctype="multipart/form-data">
    {% csrf_token %}
    <label for="custom-select">Select Algorithm Type to
Implement</label>
        <select class="custom-select" name="algoval"
            style="width:250px;font-size:18px;height:38px;">
            <option value="linear">linear</option>
            <option value="decision">decision</option>
            <option value="Random">Random</option>
            <option value="logistic">logistic</option>
            <option value="svm">svm</option>
            <option value="knn">knn</option>
            <option value="ridge">ridge</option>
        </select>
        <input type="submit" class="btn btn-primary" name="train"
value="Train">
        <input type="submit" class="btn btn-primary"
value="Predict" name="predict" >
    </form>

```

B.SCREENSHOTS



The above Screenshot represents the UI of the AutoML Tool



After uploading cleaned dataset it will ask us to choose the Regression or Classification

Show History

Selected : Regression

new_total_sqft

Alandi Road

Ambegaon Budruk

Anandnagar

Aundh

Aundh Road

Balaji Nagar

Baner

Baner road

Bhandarkar Road

Bhavani Peth

Unnamed: 0

Enter "to be predicted" column (y value)

Enter Test Size :

0.20

Train Test Split

Auto ML Tool

Result

We have to select the predicted column and the test size ratio

TechSnap Auto ML Tool

Successfully Splitted into four datasets

Show History

Print X columns

Print Y columns

X Train Size

X Test Size

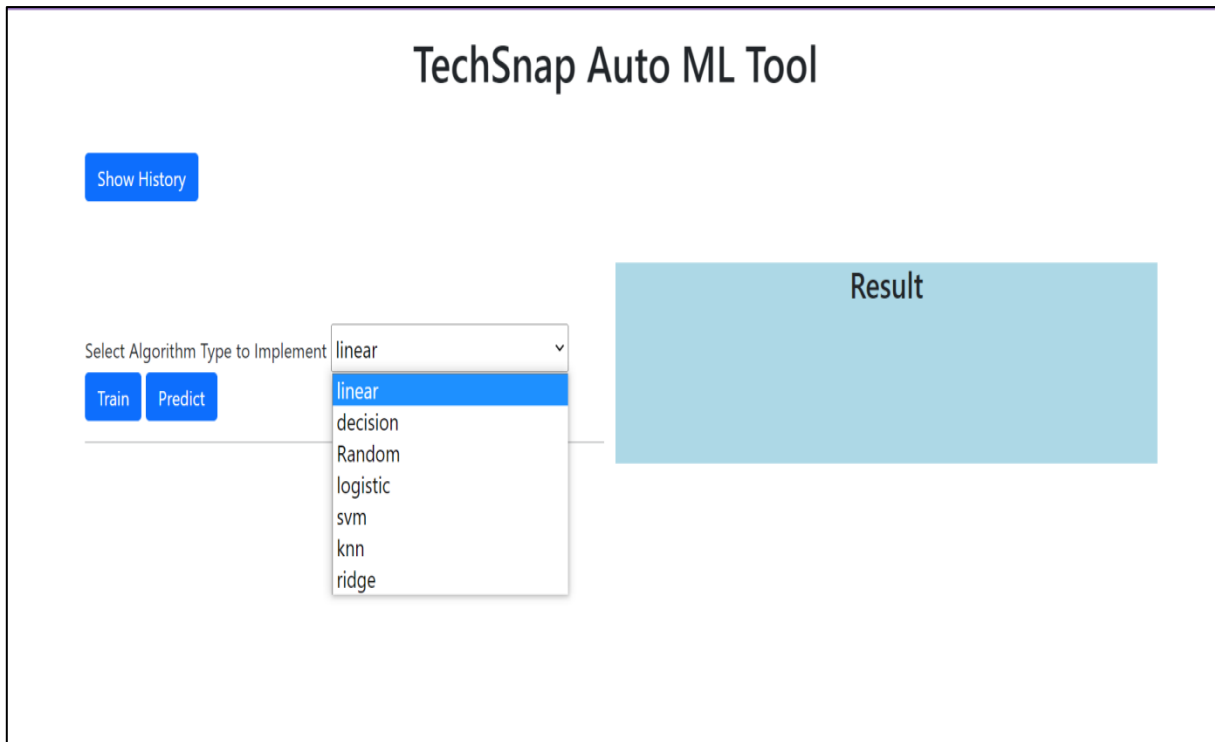
Y Train Size

Y Test Size

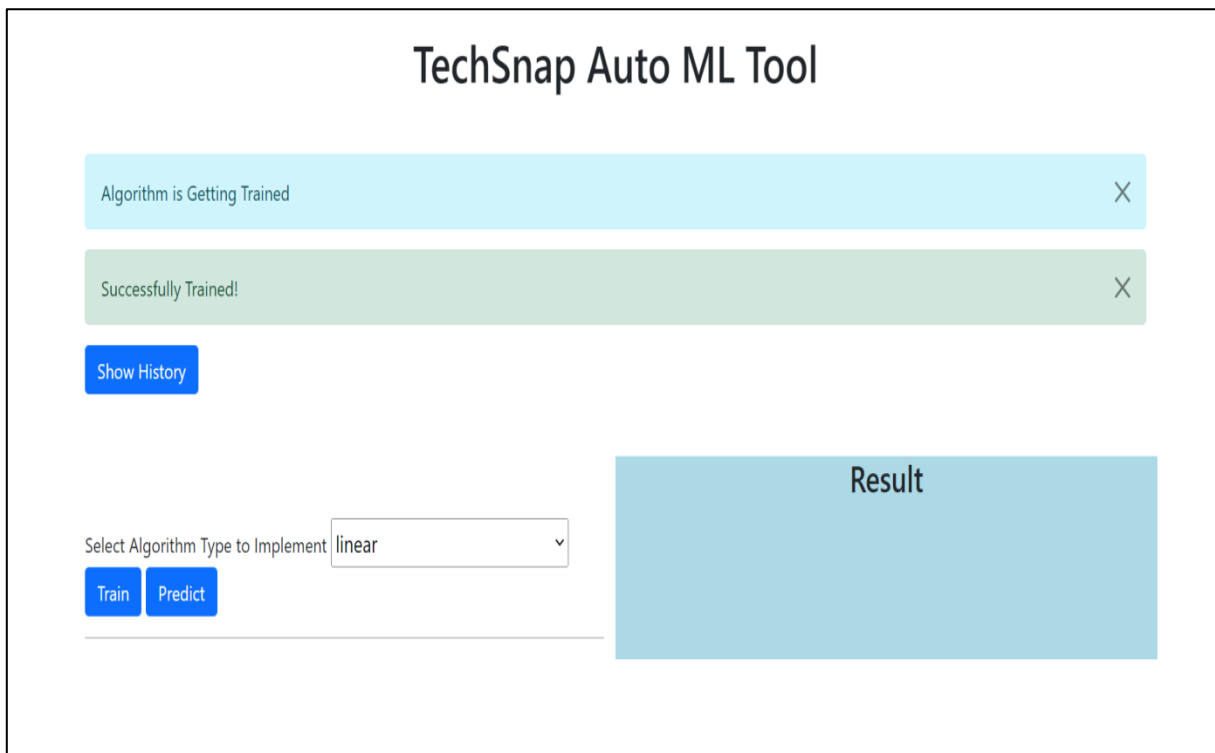
Skip to Algorithm Implementation

Result

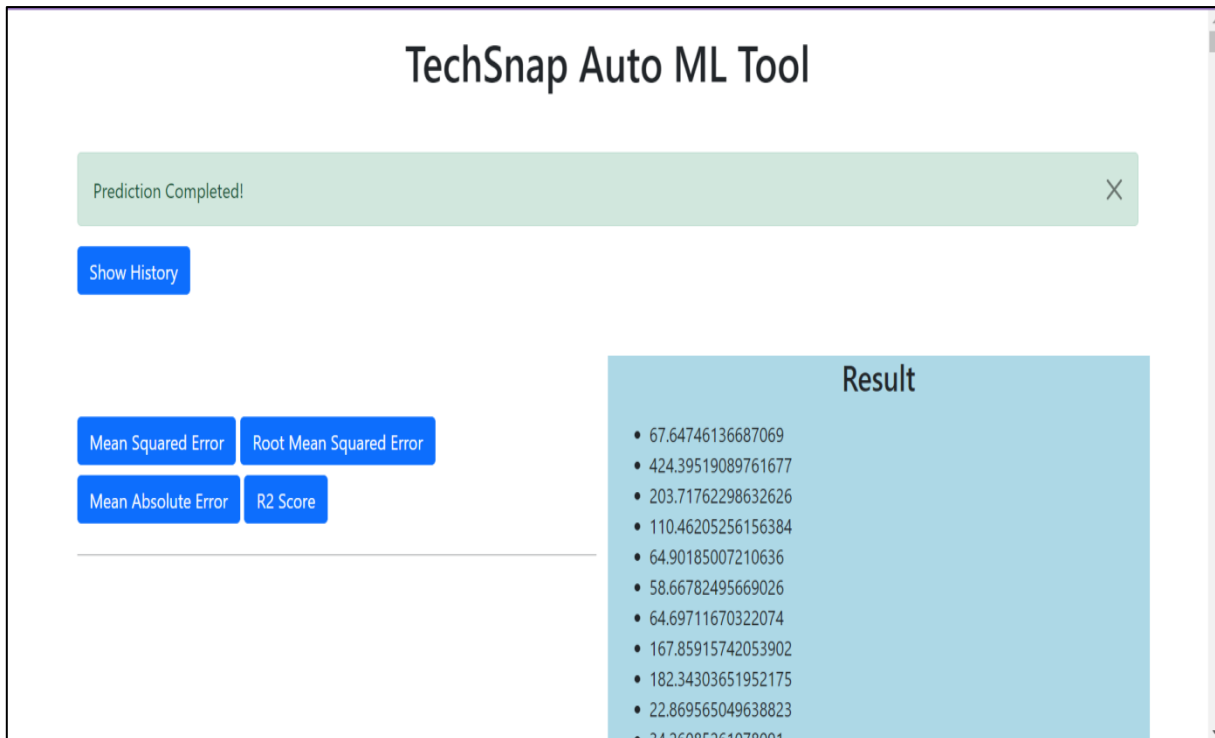
The above screenshot shows like after doing test train split. While clicking on those buttons then the resultant output displays in the result box.



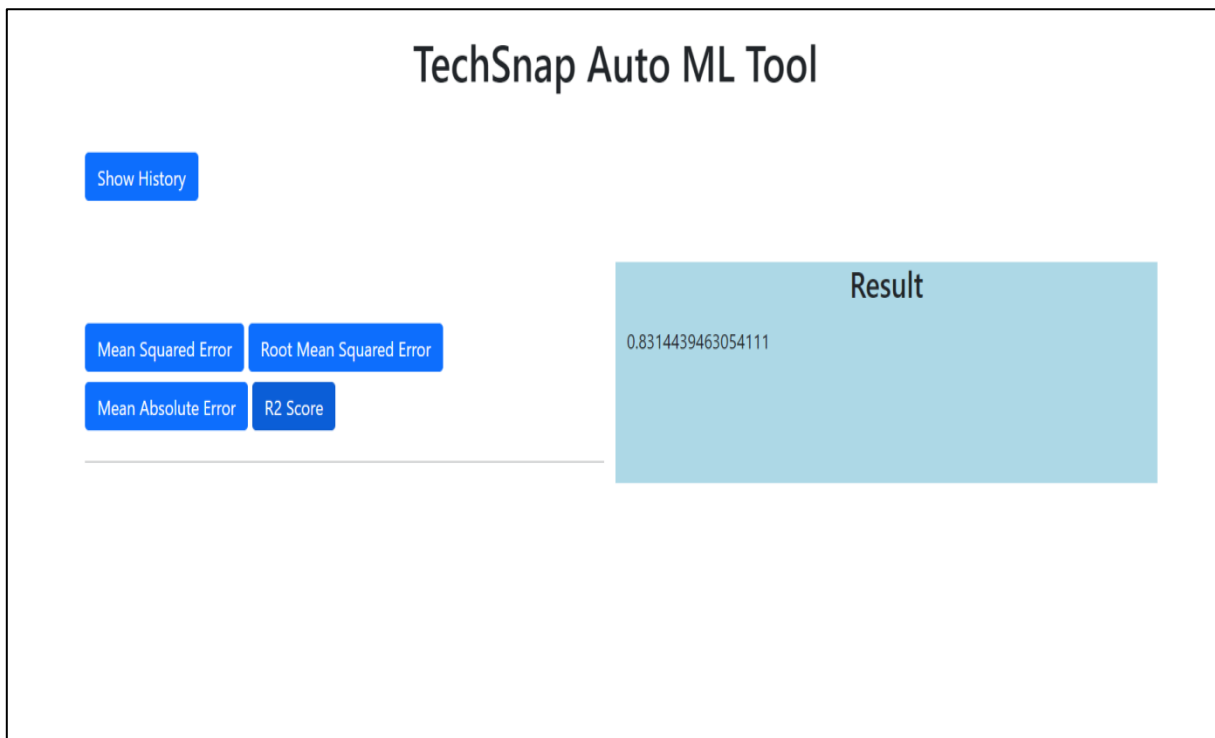
After clicking on Algorithm implementation button it will ask us to choose the algorithms in the drop down menu bar.



Then train and predict the values by clicking the buttons train and predict.



After Prediction completes it will display completed message and displays predicted values in the result box.



Finally, We can even find mean square error, root mean square error, mean absolute error and r2 score easily by just clicking on those buttons.

C. PUBLICATION

automl research paper (1).docx

by Automl Research Paper (1).docx Automl Research Paper (1).docx

Submission date: 26-Jan-2022 04:09AM (UTC+1000)

Submission ID: 1747993775

File name: automl_research_paper_1.docx (261.01K)

Word count: 1966

Character count: 10996

AutoML Tool for Supervised Machine Learning Data

Desamsetti Sai Pavan Saketh¹, Kadimisetti Prasanth², Dr.P.Asha³

^{1,2} Student, ³Associate professor, CSE, Sathyabama Institute of Science and Technology, Chennai, India

ABSTRACT: Automl sets out as connection joining the different levels of competence when building Machine learning pipelines or systems and achieve the data science processes more quickly. We present the common Automl tool which works on cleaned datasets using normal Ready-made algorithms provided by sklearn to run against regression and classification datasets and we also use the open source Automl libraries like auto-sklearn,hyperpot,TPot etc and found that TPOT is best suitable for the regression datasets and auto-sklearn is best suitable for the classification datasets .The auto sklearn library is basically used for running the auto-ml . By using the scikit-learn machine learning library we can train the high level of machine learning algorithms and can find best accuracy easily for the given dataset.

Keywords: Automated machine learning, Machine learning, automl tool.

1. INTRODUCTION

Automated Machine Learning (AutoML) supports users and scientists with the difficult task of creating machine learning pipelines and has achieved essential success.

In this project we introduce our new AutoML tool that makes the application of few machine learning algorithms easier to cleaned dataset without having any prior knowledge in coding.Our AutoML tool benchmark cleaned datasets by applying basic machine learning algorithms and also by comparing our result with other existing auto ml libraries. So we can train high level ml algorithms easily and it suggests the best accuracy model to predict the given the dataset.

2. RELATED WORK

Benchmarking Automatic Machine Learning Frameworks by Adithyan Balaji, Alexander Allen in the year of 2018. We test auto-sklearn, TPOT, auto_ml, and H2O's AutoML solution runs for a cleaned dataset of regression and classification datasets and found that TPOT is best suitable for the regression type datasets and auto-sklearn is best suitable for the classification type datasets

Efficient and Robust Automated Machine Learning by Matthias Feurer, Aaron Klein, Katharina Eggensperger in the year of 2015. The demand for the machine learning has been increased due to the success of machine learning in

various range of applications .For good effective, such systems need to automatically chooses the algorithm and data pre-processing steps for a new dataset and should set their respective hyper-parameters.

Hyper-parameter Optimization of Machine Learning Algorithms: Theory and Practice by Li Yang, Abdallah Shami in the year of 2020 Currently most of us using machine learning algorithms in different applications and in different areas. For fitting into different problems the machine learning algorithms have to do hyper-parameters must be tuned. There is an direct impact on models performances by selecting the best hyper-parameter configuration for ML models. It even also requires more knowledge on ML algorithms and hyper-parameter optimization .

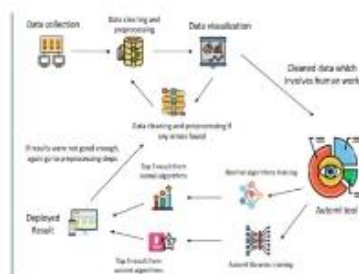
Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn by Brent Komer, James Bergstra, Chris Eliasmith in the year of 2014. Hyperopt-sklearn is the project which gives the auto algorithm configuration for the sklearn ML library. Following Auto-Weka,To represent a single large hyper-parameter optimization problem we can take the choice of large pre-processing modules and the choice of classifier together .

AutoML is an important tool to train any number of algorithms easily and accurately. Basically there are many built-in libraries in the machine learning to simplify the steps of coding. Now the autoML tool is basically to train the high level custom machine learning models with minimal effort and machine level expertise. Even the non-coders can easily use this autoML tool to train machine learning models easily.

We are using different ML Algorithms for obtaining the best from that to use. We are using a dataset from Kaggle, the dataset taken by Kaggle is thoroughly Pre-processing and cleaning has been done. Convert all the categorical data into the numerical data in the dataset. After this data we split the dataset into test data set and we trained that test data set with train data set and finally, we take the Top three best accuracy Algorithm by comparing different Algorithms. So then we suggest the user the three best algorithms to train their model.

This auto-ml tool helps in predicting the values of both classification and regression projects with the best accuracy models. The steps involved in this implementation are Data Gathering which is to collect the required dataset and then perform data pre-processing . The data has to be clean completely and delete irrelevant and unnecessary data from the dataset. The Data cleaning helps in removing the unwanted data like missing values, Noisy data and removing outliers.

Splitting the dataset for train and testing then implemented all the classification and regression algorithms. Now the user can find the accuracy score by just clicking the algorithm button. So like that the accuracy scores will get and for getting accuracy scores of all algorithms the user just clicks the all button then all algorithms will run and displays the accuracy scores of all algorithms respectively.



48

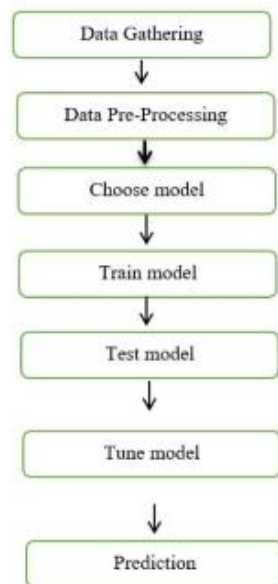


Fig: 1.2 PROJECT WORKFLOW

For more info related to the project are followed the steps involved in doing the project is Data preprocessing, Data Analysis of visualization, compare the algorithms and get valuable accuracy score for better result.

5. MODULES DESCRIPTION

5.1 PUNE HOUSE DATASET

The automl tool is performed on any classification and regression datasets. For sample We have taken Pune house dataset for predicting the house price in the pune by taking the parameters like area_type,availability size,society,total_sqft,bath,balcony,price,site_location. By taking all these parameters we will predict the price of the house in pune.

Fig: 1.3 DataSet

5.2 DATA VALIDATION/ CLEANING/PREPARING PROCESS

After taking the dataset, the first and most important step is data cleaning or validation. In this step by using some pandas libraries like shape, isnull, describe, info etc., for analysing and cleaning the dataset. Here we remove the unwanted data like missing values, noisy data and null values removing outliers from the dataset. For removing missing values here, we followed by removing or ignoring the unwanted tuples and also by filling the missing values in the dataset. After this we check for the noisy data if there is any noisy data identified in the dataset after checking the missing values. We have used some methods like binning, regression and clustering for removing the noisy data in the dataset. After removing missing values and noisy data from the dataset if there any outliers are present we remove them by using clustering technique.

Algorithms

We have used all classification and regression algorithms for the model building in this automl project.

The **classification algorithms** which used are:

- Support vector Machine
- Naïve Bayes classifier
- K-nearest neighbors algorithm
- Logistic regression
- Random Forest

The **Regression Algorithms** which used are:

- Linear Regression
- Lasso Regression
- Decision Tree Algorithm
- Random Forest
- Knn Model
- Support Vector Machine

We have taken Pune House Dataset and have done data pre-processing then applied above all Regression algorithms on the model .Therefore these are the accuracy scores of all those algorithms will displayed in the below table.

Linear Regression	82.6
Decision Trees	76.2
Random Forest	85.1
Support Vector Machine	69.0
KNN	82.3
Ridge	82.6

So we can see the accuracy scores of all Regression algorithms .Now it will suggests to take the top three best accuracy algorithms from it, which are:

Random Forest	85.1
Linear Regression	82.6
Ridge	82.6

Our tool will display the above three best accuracy algorithms for our given dataset.

5.3 DEPLOYMENT USING FLASK

The project is deployed using Flask for making an interface. The interface contains the Dataset upload button for uploading cleaned dataset and display bar is used to display the accuracy scores of the all models.

6. RESULT

The result of this project is first we need to upload the cleaned dataset in the file upload button. Then it will ask the user to choose the classification or regression category in the display bar. Then all corresponding algorithms will display if we click any algorithm then automatically codes executes and displays the corresponding algorithm and if we want to display the accuracy scores of all algorithms then we can use all button then it will display the accuracy scores of all algorithms.

7. CONCLUSION

By using this auto-ML. Tool the people can easily train all the classification and all Regression algorithms easily. By clicking the algorithm name then the code will run and displays the accuracy score of that model quickly and easily so we can check any number of algorithms easily with minimal effort. Hence the non-Coders also easily do the machine learning with this auto-ML tool.

REFERENCES:

- [1] G. Cawley, , I. Guyon, S. Escalera, K. Bennett, H. Escalante, M. Saeed, A. Statnikov, E. Viegas T. Ho, N.Macia, and B. Ray. Design of the 2015 ChaLearn AutoML Challenge. In Proc. of IJCNN'15, 2015.
- [2] M.-A. Ziler and M. F. Huber, Survey and benchmark of Automated Machine Learning Frameworks, arXiv preprint arXiv:1904.12054, [2019]. <https://arxiv.org/abs/1904.12054>.
- [3] M. Kuhn and K. Johnson, Springer, Applied Predictive Modeling (2013) ISBN: 9781461468493.
- [4] L. Kotthoff, J. Vanschoren, and F. Hutter, Eds., Automatic Machine Learning: Systems, Methods, Springer, Challenges (2019) ISBN: 9783030053185.
- [5] G. Luo, A review of automatic selection methods for Hyper-parameter values and machine learning algorithms Netw. Model. Anal. Heal. Informatics Bioinforma. 5 (2016) 116. <https://doi.org/10.1007/s13721-016-0125-6>.
- [6] Q. Yao et al., Taking Human out of Learning Applications: A Survey on Automated Machine Learning, arXiv preprint arXiv:1810.13306, (2018). <http://arxiv.org/abs/1810.13306>.
- [7] . Nalepa, P, R. Lorenzo, L. S. Ramos, J. R. Paster and JM. Kawulok and Particle swarm optimization for hyper-parameter selection Proc. ACM Int. Conf. Gent. Evoll. Compute., (2017) 481488.
- [8] W. Hosmer Jr, S. Lemeshow, Applied logistic regression, Technometrics, 34 (1) (2013), 358-359.
- [9] R. Martinez-Cantin, BayesOpt: A Bayesian optimization library for nonlinear optimization, and bandits ,experimental design, J. Mach. Learn. Res. 15 (2015) 37353739.
- [10] M.A. Gardner, C. Gage ,F.A. Fortin, F.M. De Rainville, M. Parizeau, DEAP: Evolutionary algorithms done easily, J. Mach. Learn. Res.13 (2012) 21712175.

- [11] N.L.C. Talbot, G.C. Cawley On over-fitting in model selection and selection bias in performance evaluation , J. Mach. Learn. Res. 11 (2010) 2079-2107
- [12] S. Tatikonda, M. Boehm, A. Surve, et al., SystemML: declarative machine learning, Pro. VLDB-Endo. 9 (2016) 1425-1436.
- [13] B. Yavuz, X. Meng, J. Bradley, et al., ML library: ML in apache spark, J. Mach. Learn. Res. 17 (2016) 1235-1241.
- [14] P. Frasconi, L. Franceschi, M. Donini, and M. Pontil, Reverse and Forward gradient-based hyper-parameter optimization, 34th Int. Conf. Mach. Learn. ICML 2017, 70 (2017) 1165-1173.
- [15] J. Bergstra, B. Komer, and C. Eliasmith, Hyperoptz-sklearn: Automatic hyper-parameter configuration for scikit-learn, Proc. ICML Workshop AutoML, (2014) 3440.
- [16] T. Springenberg, F. Hutter T. Domhan, Extrapolating Learning Curves of Deep Neural Networks, ICML AutoML Workshop, 2014.
- [17] H. Hoos, K. Leyton-Brown and F. Hutter, Sequential modelbased optimization for the general algorithm configuration, LION-5, 2011. Extended version UBC Tech-report TR-2010-10.
- [18] G. Varoquaux, F. Pedregosa, V. Michel, A. Gramfort, O. Grisel, B. Thirion, V. Dubourg, M. Blondel, P. Prettenhofer, R. Weiss, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, Scikit-learn: ML in Python, The Journal of ML Research, 12:2825–2830, 2011.

automl research paper (1).docx

ORIGINALITY REPORT

9%

SIMILARITY INDEX

9%

INTERNET SOURCES

6%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

arxiv.org

Internet Source

4%

2

www.automl.org

Internet Source

2%

3

www.semanticscholar.org

Internet Source

1%

4

compneuro.uwaterloo.ca

Internet Source

1%

5

originalstatic.aminer.cn

Internet Source

1%

Exclude quotes ☒ On

Exclude matches ☐ Off

Exclude bibliography ☒ On