Auto ML

Mrs. K. Sri Lakshmi[1], M.Tech , Department of CSE, SRGEC Gudlavalleru, kalapala.srilakshmi@gmail.com
G. Deep Aman[2], Student, Department of CSE, SRGEC Gudlavalleru gantaaman12@gmail.com
CH. Raghava[3], Student, Department of CSE, SRGEC Gudlavalleru, raghavachekuri777@gmail.com
B. Satya Rahul[4], Student, Department of CSE, SRGEC Gudlavalleru, bandelarahul2002@gmail.com
D. Venkat Appaji[5], Student, Department of CSE, SRGEC Gudlavalleru, dabbalaappuvenkat23@gmail.com

Abstract
In today's world, every task has been automated by the power of technology and artificial intelligence[1]. With all this automation being part of human survival, it is even difficult for people to learn things and to implement the code from scratch to develop a machine-learning model for various applications[2]. Even though they want to implement from scratch, an abundant number of algorithms are available, which makes it far more difficult to choose the correct and appropriate algorithmic model for their data.

To solve the above problem, we are proposing a project that will eradicate all the above issues by automating the model training process. In this project, we will provide a user interface via a website where the user can submit his dataset and provide us with his target variable[7]. Once the user has submitted the dataset and target variable with the help of python at the back-end we will perform the required data pre-processing and implement several algorithms. Finally, we will give them the best model with high accuracy which he/she can directly use.

Keywords: automl, model, data pre-processing, algorithm, drive, plots, clustering, visualize

Introduction

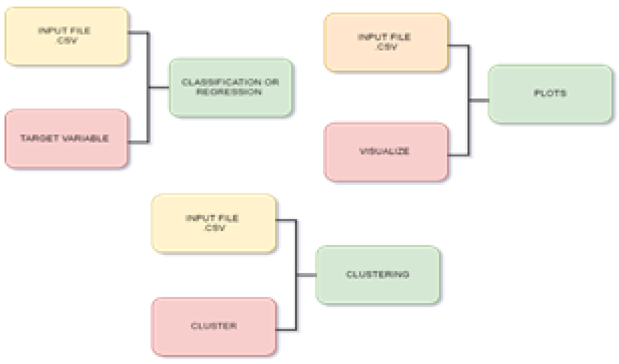
Automation is an important factor for future generations. Bringing automation into the way we code is also important, so that the people at the far end can easily implement the task they want even if they lack knowledge of the task. In today's world machine learning is a very fascinating area of interest for everyone[9]. But, to achieve the desired results, one has to go through the various algorithms and the code for each of the algorithms. Even if he acquired code, then the problem at hand would be finding the best out of many algorithms. Not only this, even for visualizing the data that he has he needs to implement some code or else have to do it manually. By considering all these problems at hand, we have introduced a one-stop solution to all these problems[4]. AUTOML is a platform where you only need to provide data that you have and the target variable or feature that you want to predict for future data. And the rest of the process starting from applying algorithms and finding the best algorithm and even the visualization will be taken care of by our model.

Literature Survey

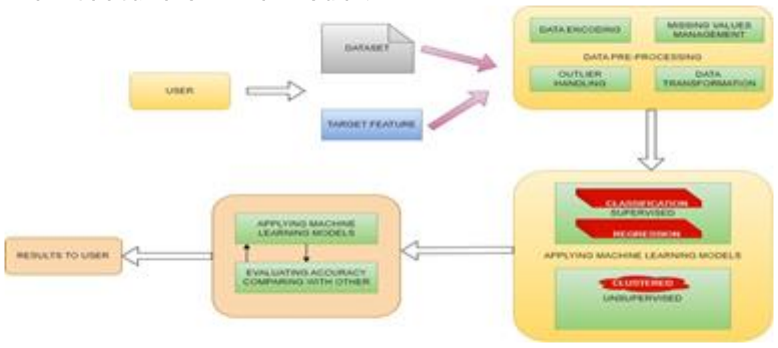
Previously, there was an "Auto-Sklearn" module which has been introduced around the year 2020[5]. This is a module that implements both the classifier and regressor by tuning different parameters and using different algorithms in the stipulated time given by the user. Though it saves the time required for finding the best model and also reduces the stress to the user to code for the best algorithm, it still requires the user to implement the module and code. Which again adds the work to the user. To overcome that issue here we are proposing the best solution which saves time as well as provides good interphase[6]. And use a wide variety of implementation methods.

Inputs:
The input for this model is the data file which is in .csv format. A text file with a specific format known as a CSV (comma-separated values) file allows data to be saved in a table-structured fashion.

The second input to this model varies based on the needs of the user. The second input can be a target variable that the user is interested in predicting, but this is only in the case of Regression or Classification tasks. In another case, it can be a thing that acts as an action specifier. By the word action specifier what we intend to say is that it can be used to represent the type of action to be performed, which in turn it comes handy in the case of Visualization and clustering. Although in those cases we don't need a target variable. In both cases, the .csv file is a must. And one more thing that matters is giving correct input to the model that is the target variable much match the column name of anyone the columns. Though with the help of python language tools we are handling the issues with the latter case and unnecessary spaces on both sides of the input.

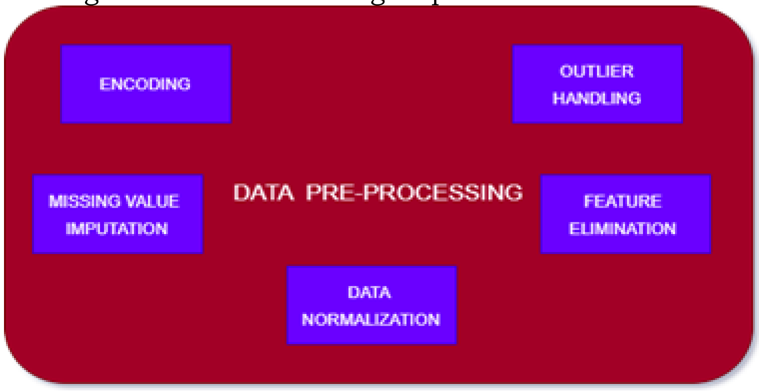


Understanding Architecture of The Model:

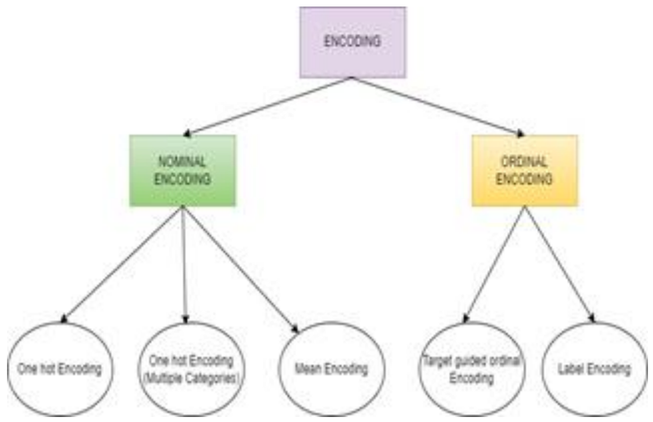


As the user submits the file and target variable in case of Regression and Classification the target variable would be the column name that the user wants to predict. Whereas in the second case clustering, there will be no target variable. The reason for combining these three in one architecture is that they are of similar workflow. The architecture involves 4 major parts. The first one is data accusation and the second major sector is data pre-processing once it is done. The third and major task is identifying the problem at hand. In the case of clustering and visualization, the problem identification is straightforward forward but in the case of Classification and Regression, it involves a much more complex process which we will discuss later in the paper. The final step is sending the accuracy graphs and the best model to the user. Let's now understand each step, in detail.

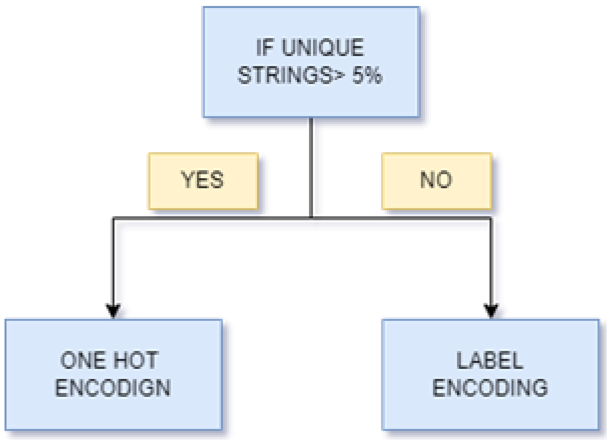
Data pre-processing:
The data pre-processing involves the following steps:



Encoding:
Categorical variables can be easily fitted to a machine learning model by using the encoding approach, which encodes them into numeric data.
There are many encoding techniques like the below that can be seen in the below image out of which we are employing label encoding technique as a base model which can be further made complex encoding processes[3].

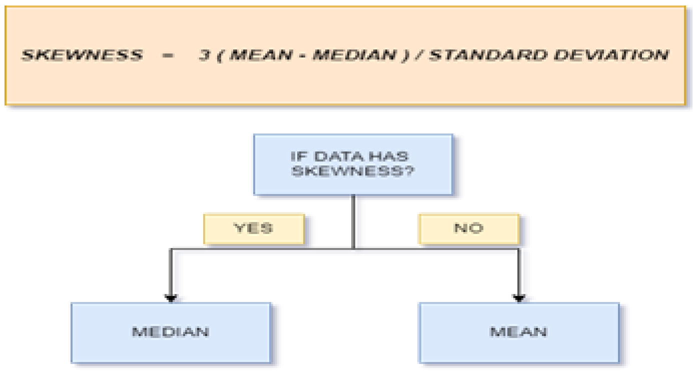


Label encoding works by converting each different string into a unique number. Although this might not be the best-employed method for the columns which has more unique strings, this is implemented as one of the standard methods in our model. Later we can employ a method by using the feature uniqueness that can be understood by the below flowchart.

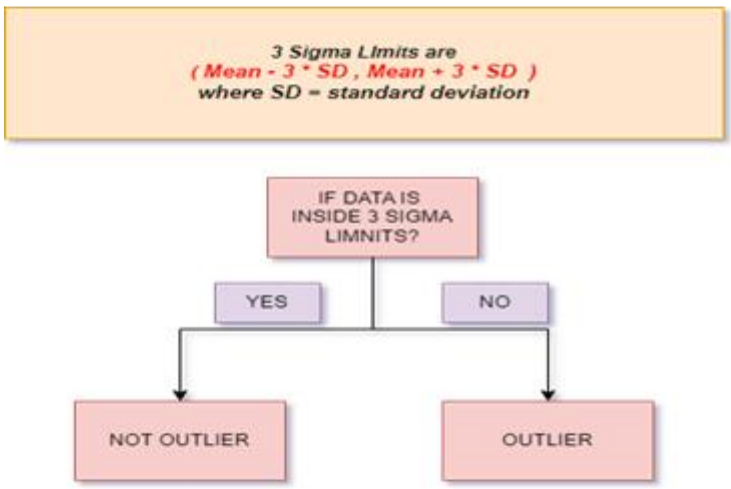


Here we employed both Nominal and Ordinal encoding methods, which provide a strong base for our desired model. The other time-taking but best approach is to apply both these encoding techniques one at a time and implement all the machine learning algorithms. But this increases turnaround time.

Missing Value imputations:
When you don't have information captured for particular columns or variables, it leads to random error, also known as missing values. For a wide range of reasons including erroneous data entry, malfunctions, deleted files, and many others, data may disappear. Void spaces in your spreadsheet symbolize null values or missing values. These things will be managed by our model internally by using various techniques that were available in Pandas and NumPy. The best-employed technique is the median imputation method and means imputation methods. Rather than using 0 as a standard technique, we implemented it using a better idea, that is by understanding the skewness in the data. If the data has skewness, we will use the median else we prefer the mean[8].



Outlier handling:
An observation that deviates drastically from those other values in a feature or variable is referred to as an outlier. Many solutions were readily available for handling outliers but one of the best-employed techniques is to identify the outliers using 3 sigma limits and then we are employing the same logic that we are using for missing values. The 3 sigma limits formulation is defined below.


3 Sigma Limits are
(Mean - 3 * SD , Mean + 3 * SD)
where SD = standard deviation
IF DATA IS INSIDE 3 SIGMA LIMNITS?
YES
NO
NOT OUTLIER
OUTLIER

Feature Elimination:
There will be certain columns in your data that are not relevant to the task that you're trying to perform. That is in the case of predicting the salary of an employee features like the name of the employee are not appropriate, in the same way, there will be certain

columns in our data that are not relevant to the target variable that we are predicting. So, the best solution is to eliminate those features or columns. This is done so that the time required for applying the algorithm is reduced. But how to decide which are important and which are not? The best solution is to use correlation as the method for determining how well a particular feature is related to the target variable. There are many formulations available for calculating correlation one of the best ones is the Spearman correlation coefficient[10].

IF your data does not have tied ranks

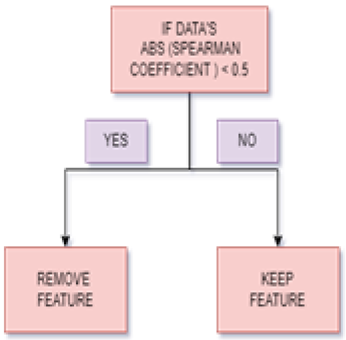
$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

where d_i = difference in paired ranks and n = number of cases.

IF your data has tied ranks

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

where i = paired score.

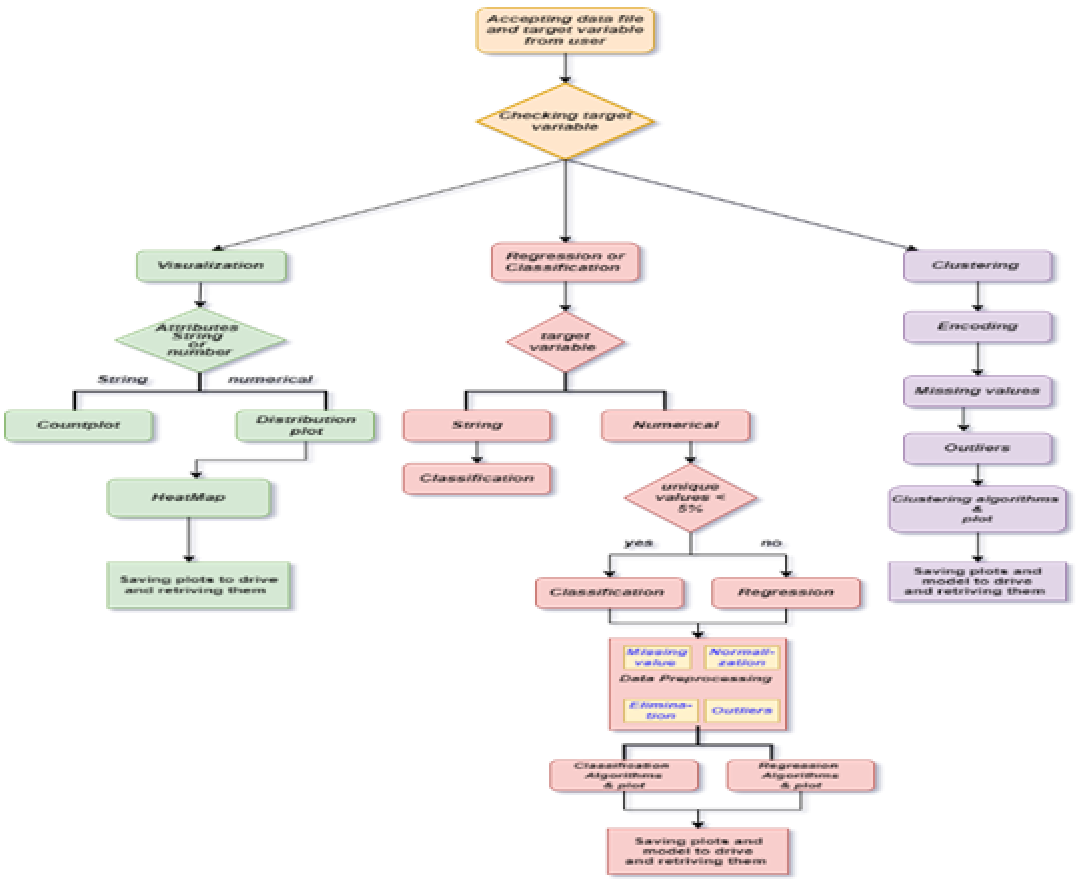


Data Normalization:
Data normalization is performed to bring all the data columns into the same scale. Usually, the main idea behind bringing all the data to the same scale is to reduce the difference between measurement employed by each feature. The best way to perform data normalization is as follows

$$Z = \frac{x - \mu}{\sigma}$$

There are several other methods for performing normalization which include min-max normalization and decimal level normalization etc. It is not always advisable to perform normalization. So then how to decide whether to normalize or not? The best solution is to compare how the values in one column are differing from other values in other columns that is, here we will check the proportion of the values are differing from each other. If the values are different in the ratio of around 10-100% then we choose can choose to perform the normalization.

Understanding Architecture off Exact System:

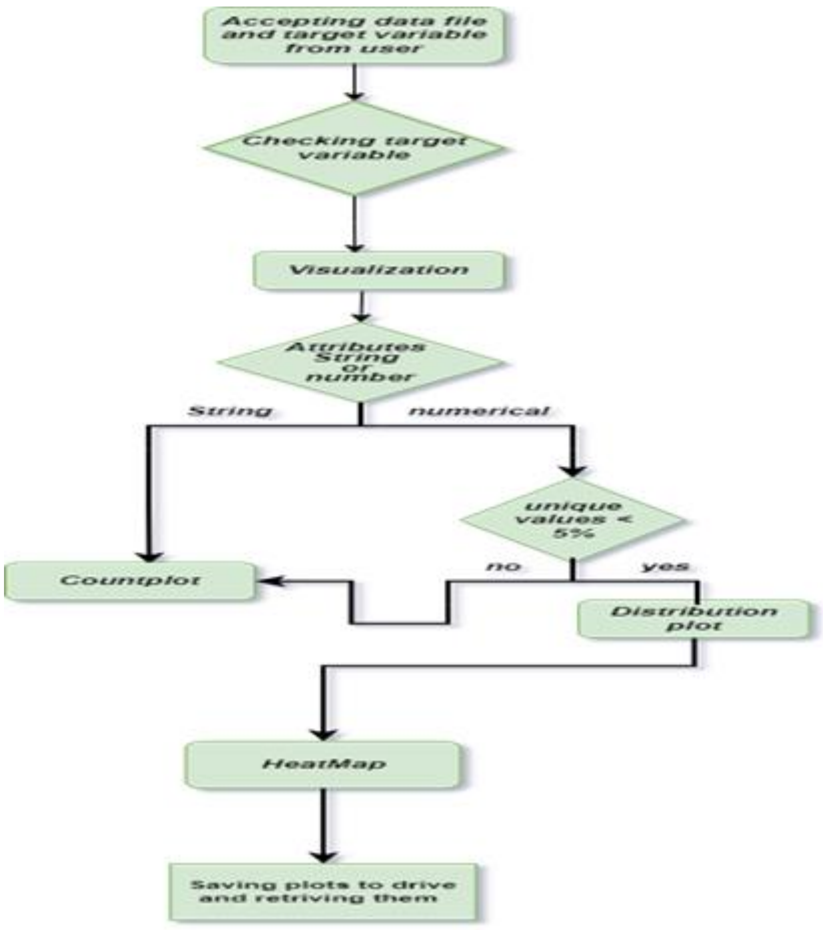

Accepting data file and target variable from user
Checking target variable
Visualization
Regression or Classification
Clustering
Attributes String or number
String
numerical
Countplot
Distribution plot
HeatMap
Saving plots to drive and retriving them
target variable
String
Numerical
Classification
unique values < 5%
yes
no
Classification
Regression
Missing value
Normalization
Data Preprocessing
Elimination
Outliers
Classification Algorithms & plot
Regression Algorithms & plot
Saving plots and model to drive and retriving them
Encoding
Missing values
Outliers
Clustering algorithms & plot
Saving plots and model to drive and retriving them

There are majorly 3 parts to it. Let's understand each part in detail in the below section.

Visualization:
Whenever the user chooses to visualize the data, he will type "visualize" instead of typing the variable name in the target variable place. Once he chooses that internally we will perform the following operations for each column in the dataset.

- Identifying data type
- Deciding which plot to draw
- Uploading the plots to drive
- Retrieving plots from the drive using a flask

Let's understand each step in-detail. The first step is to identify the data type of the column that we are going to plot. Because as a base model we decided to plot a "Distribution plot", if the data in the column is numeric, else we will plot "Bar plot" or "Count plot". And finally, we will plot "HeatMap". This identification of data type is done as follows


Accepting data file and target variable from user
Checking target variable
Visualization
Attributes String or number
String
numerical
unique values < 5%
no
yes
Countplot
Distribution plot
HeatMap
Saving plots to drive and retriving them

Clustering:
The process for clustering is pretty straightforward as above, that is whenever the user enters “cluster” in place of the target variable, we will perform clustering. In simple words, clustering means grouping the data that are similar to each other

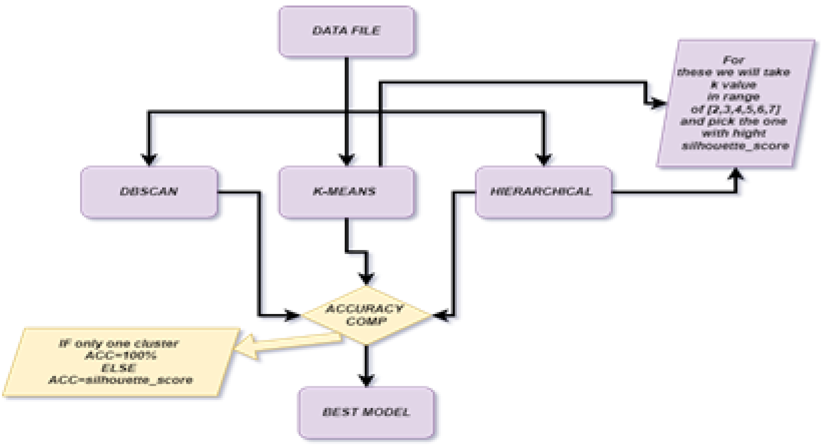
There are several techniques to perform clustering out of which we are using the following algorithms

- DBSCAN
- K-MEANS
- HIERARCHICAL

After applying these algorithms, we will compare the accuracy of these models using the special method called “silhouette_score”. A statistic used to assess the efficacy of a clustering method is the silhouette coefficient, often known as the silhouette score. Its value is between -1 and 1.

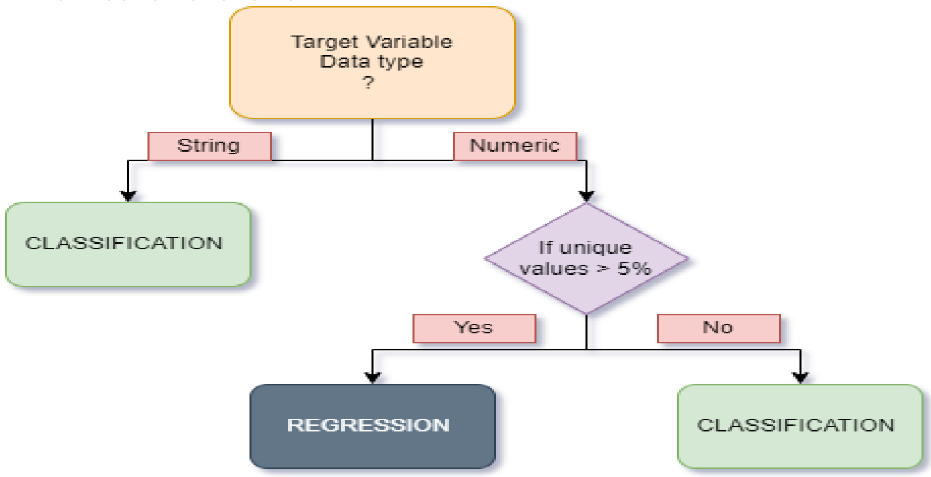
The main problem with the clustering approach is how to decide the number of clusters that we can form. This is a process where we cannot perform the clustering without having domain knowledge in some cases. So, the only possible approach to this problem is just by performing clustering with a different number of clusters. That is, in this model, we are taking a range of cluster numbers in an array and then we are building the cluster models for each number. That is, for example, we will consider the cluster numbers including 2,5,12,20. And once we are done forming clusters, we will calculate the accuracy of these models by using “Silhouette_score”. And now we will pick a value with high accuracy and then we will try picking values around the value with high accuracy. And we will again build models.

And then we will finally consider the model with the highest accuracy. And return the model to the user.Finally, we will upload the best model to the drive we will retrieve the pickle file that is the model file from the drive and we will allow the user to download it with one click.


DATA FILE
For these we will take k value in range of [2,3,4,5,6,7] and pick the one with hight silhouette_score
DBSCAN
K-MEANS
HIERARCHICAL
ACCURACY COMP
IF only one cluster ACC=100% ELSE ACC=silhouette_score
BEST MODEL

Regression and Classification:
In the case of regression and classification, the only difference is that the user submits the column name that he/she wants to predict in place of the target variable. Once the user submits the target variable and dataset, the next main task is to identify whether the task is classification or regression, since for both there will be a target column to predict.

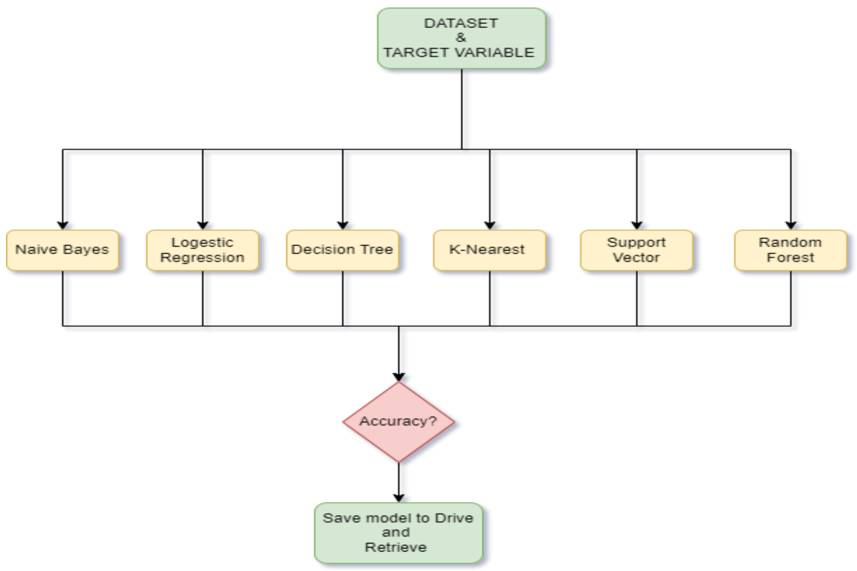
The optimal solution for this task will be identifying the datatype of the target column that we are predicting since for classification the target column datatype would be "String". But for all cases we cannot depend on that assumption, so we found an optimal solution to the problem which looks as follows.



Once the distribution of tasks between classification and regression is done the next task is to apply the algorithms to the data. Since these are supervised learning techniques, we will find the accuracy of all the algorithms and plot those accuracies. And the algorithm which has given the best accuracy is saved to drive as a pickle file. And retrieve them using a flask.

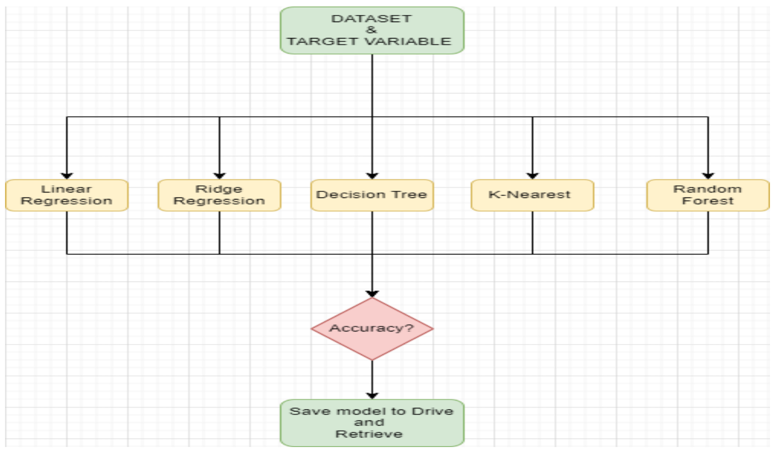
Now let's see the structure of algorithms that are used for classification. The classification algorithms that are used in our model are

- Naïve Bayes
- Logistic Regression
- Decision Tree Classifier
- K-Nearest neighbour
- Support Vector Machine
- Random Forrest



In the same way, we use several algorithms for regression including

- Linear Regression
- Ridge Regression
- Decision Tree Regressor
- Random Forrest Regressor
- Support Vector Machine



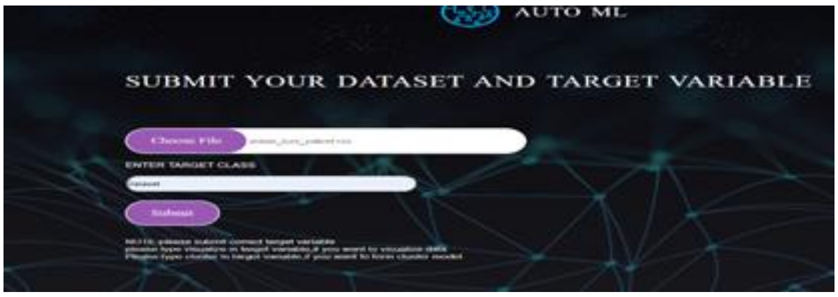
There is one more important concept that we have to discuss here which is called tuning the algorithms. Algorithms like a decision tree and random forest require the tuning of parameters to get the best accuracy of the model.

To find the best parameters for these algorithms the one and only approach is to randomly apply different parameters to the model and pick the one with the best accuracy.
The best tuning method available is "GridSearchCV". This method works as follows. It will take certain possible parameters as a dictionary and then apply all the different combinations of parameters on the data available and result in the best parameters that produced the best accuracy.

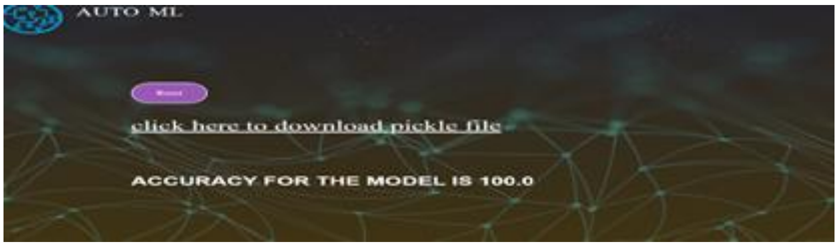
But the only constraint here in this tuning approach is the time required to find the best parameters. Since there will be a different number of parameters for each algorithm it would take a huge amount of time to find the best parameters. The only solution to this problem is by applying the algorithm to the portion of the dataset. That is we will take around 20% of the data in a uniform manner. And once we have taken the data we will apply tuning to the portion of data. And based on the best parameters obtained we will again apply those parameters to the entire dataset and consider that as the best model.
Now let's understand how the user journey would be as you can see below initially the user will submit the dataset and the target variable in the corresponding positions. Once the user entered the details, he can choose to submit those details to our model.

Once the user submitted the data file and target variable it would be redirected to our model through the flask server. Once the file and target variable is reached, the model will be applied to the given file.

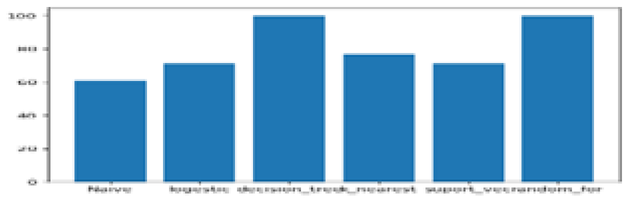
And the results will be updated by the user in some time frame. The time frame completely depends on the size of the dataset that is used.



NOTE: please submit correct target variable
please type visualize in target variable,if you want to visualize data
Please type cluster in target variable,if you want to form cluster model



ACCURACY FOR THE MODEL IS 100.0



References:

1. Feurer, Matthias & Eggensperger, Katharina & Falkner, Stefan & Lindauer, Marius & Hutter, Frank. (2020). Auto-Sklearn 2.0: The Next Generation.
2. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J.T., Blum, M., Hutter, F. (2019). Auto-sklearn: Efficient and Robust Automated Machine Learning. In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-05318-5_6
3. F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., Automatic Machine Learning: Methods, Systems, Challenges, ser. SSCML. Springer, 2019.
4. H. Jin, Q. Song, and X. Hu, "Auto-Keras: An efficient neural architecture search system," in Proc. of KDD'19, 2019, pp. 1946–1956.
5. I. Guyon, L. Sun-Hosoya, M. Boull´e, H. Escalante, S. Escalera, Z. Liu, D. Jajetic, B. Ray, M. Saeed, M. Sebag, A. Statnikov,W. Tu, and E. Viegas, "Analysis of the AutoML Challenge Series 2015-2018," in Automatic Machine Learning:Methods, Systems, Chal-lenges, ser. SSCML, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds.Springer, 2019, ch. 10, pp. 177–219.
6. M. Feurer, K. Eggensperger, S. Falkner, M.Lindauer, and F. Hutter,"Practical automated machine learning for the automl challenge 2018," in ICML 2018 AutoML Workshop, 2018
7. P. Probst, M. N. Wright, and A.-L. Boulesteix, "Hyperparameters and tuning strategies for random forest," WIREs Data Mining and Knowledge Discovery, vol. 9, no. 3, p. e1301, 2019
8. NeuroQuantology|November2022| Volume 20 | Issue 15 | PAGE 5932-5944| DOI: 10.48047/NQ.2022.20.15.NQ88597 G.Deep Aman / Performance Analysis of Visual Geometry Group Algorithms on Pneumonia Classification
9. NeuroQuantology|November2022| Volume 20 | Issue 15 | PAGE 5945-5951-5962| DOI: 10.48047/NQ.2022.20.15.NQ88599 Ashok Reddy Kandula / Machine Learning Algorithms for Predictive Analysis in the Identification of Chronic Liver Disease
10. NeuroQuantology |November 2022 | Volume 20 | Issue 16 | Page 2107-2113 | doi: 10.48047/NQ.2022.20.16.NQ880301 M.N.Satish Kumar et al / Lung Opacity Detection Using Transfer Learning.