International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET231035

Web Based Machine Learning Automated Pipeline

Prof. Sachin Sambhaji Patil, Mahesh Manohar Sirsat, Ajitkumar Vishwakarma Sharma, Aashish Shahi, Omkar Maruti Halgi

Computer Engineering Department, Zeal College of Engineering and Research, Pune, Maharashtra, India

ARTICLEINFO

Article History:

Accepted: 20 April 2023
Published: 08 May 2023

Publication Issue
Volume 10, Issue 3
May-June-2023

Page Number
43-51

ABSTRACT

With the increasing volume, velocity, veracity, and variety of data, it has become critical to have efficient techniques and tools for managing and analyzing data in machine learning. Abstraction is a powerful concept that allows users to interact with machine learning algorithms without understanding their technical implementation details. In this project the user will provide the dataset in .csv format the dataset is then processed further to different machine learning preprocessing steps like removing unwanted columns, handling missing values, label encoding, outlier detection and removal, normalization, model building, model prediction, and the result can be downloaded as pdf, tracable pdf and CSV, this all processes gives a result of different model and their respective accuracy so that we can choose the best model for that particular dataset. tracable pdf will be containing all the timestamp of the processes done with their respective result, Apart from client-server model user is also provided a api so that all processes can be implemented in different platforms like c++, java, ruby etc. Overall, this paper highlights the critical role of abstraction in managing the complexity of data and machine learning algorithms, enabling more efficient and effective analysis of large and complex datasets.

Keywords : Dataset, Dataset Filtering, Client Server, Pdf Generation, Data Preprocessing.

I. INTRODUCTION

In today's world, information sharing needs to be fast and efficient. We need tools to take effectively collected data sets from various sources and present and present these visuals in the form of charts, patterns, etc. The tools created process datasets and automate the task of finding various patterns and decoding their semantic structure. The main purpose of integrating tools with datasets is to focus on how the functionality is used rather than how it is implemented to perform further analysis.

According to IDC's AI predictions for 2020 and beyond, IT must invest heavily in data integration, management, and cleansing to effectively use intelligent automation. Data professionals continue to

be plagued by the tedious task of data cleansing. Organizations cannot achieve their digital transformation goals without an efficient way to automate data cleansing. [1] IDC Future Scape report finds solving historical data problems in legacy systems can be a significant barrier to entry, especially for large organizations, highlighting the challenges associated with adopting digital initiatives . According to Morningstar, businesses have spent an estimated $1.3 trillion (USD) on digital transformation initiatives in the past year alone. McKinsey later reported that 70% of his programs were inadequate. Tracking it down at home, outages like this cost businesses over $900 billion. Businesses cannot afford repeated failures in their digital transformation, regardless of the size of the investment lost. You need clean, standardized data to unlock the benefits of your digital transformation projects, but collecting the data you need in the way you need it can be tedious, expensive, and time consuming.

II. METHODOLOGY

This proposed framework is implemented using the machine learning preprocessing tool and Django as the backend using the client-server model, following are the steps which comprise the architecture of the model.

1) CLIENT - Our server will provide you with a user interface that contains various buttons. When you first load the HTML page, you will need to upload your dataset in CSV format. From there, you will go through several processes such as removing unwanted columns, handling missing values, label encoding, and removing outliers. After completing these steps, you will move on to the model building phase where you can build a linear regression, logistic regression, support vector machine, and more with just a few clicks. We hope that our platform will assist you in your research and enable you to conduct data analysis in a streamlined and efficient manner.

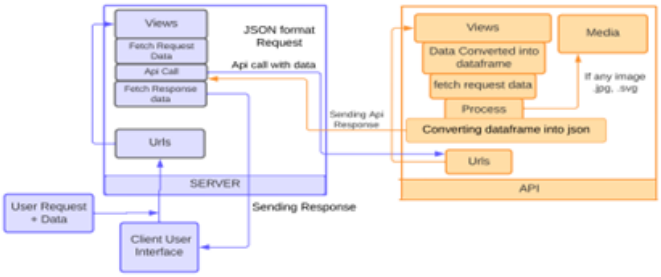


Figure 1. The architecture of the proposed system

2) API- api will handle all request which are forwarded by client to server or processor to perform analysis. Rendering data in form of html is handled by same to provide functionality of dynamic rendering. through the api call all the request is send through the server to the api's urls.py file to manage all the links.
view.py convert that data into the dataframe as it is in the json format. all the pre-processing steps is performed in the api like deleting column from the data or table, deleting missing values, encoding with label encoder, performing normalization, removing outliers, building models with different regression and classifiers in the applications, after that we perform the prediction and compare the results of the model result and compute our results. the pre-processed data is converted into the .json file which is sent to the media file which is the database of the application for temporary purpose. after that all the response is sent to the api call of the server and the response is sent to the client as the output.
3) SERVER- In a typical client-server architecture, the server acts as a bridge between the front-end (i.e., client) and the back-end (i.e., API). The server is comprised of various files such as urls.py, views.py, models.py, etc. When a user sends a request, it first goes to the urls.py file present on the server-side. This file maps the URL to a particular function present in the views.py file.
The views.py file first fetches the data associated with the request. This can involve querying a database or making a request to an external API. Once the data is retrieved, the views.py file calls the respective API

URLs. The API then returns the result in the form of JSON data.

The views.py file then takes this JSON data and passes it on to the template, along with any additional data required to render the template. The template then generates the HTML code required to display the result to the user, which is sent back to the client.

Overall,[2] this client-server architecture allows for a clear separation of concerns between the front-end and the back-end. The server-side code is responsible for handling all of the business logic, while the front-end code is responsible for displaying the data in a user-friendly manner.

By structuring the code in this way, it becomes easier to maintain and scale the application over time. The different components of the system can be developed and tested independently, and changes can be made without affecting the rest of the system.

4) PDF- At last after processing is finished (if user performing all this operations using UI interface) pdf report is send back to client which will record all analysis made and logs of command and also provide visualization to it. There is the csv as we get as the output which is fully traceable that means we can compute the result of the first prediction with the another one so that the result can be varied and hence we can achieve the better results. Further this pdf is static which is beginner friendly and show all the time stamp of the work done in the application with its result.

One of the essential features of an HTML-based front-end is the ability to download a PDF. This feature is particularly useful as it allows users to easily save and share the contents of the web page in a standardized format. PDFs can be of two types: static PDFs and traceable PDFs.

4.1 Static PDFs typically contain static data, such as the theoretical aspects of the various processes involved in a project, such as loading datasets, removing unwanted columns, handling missing values, removing outliers, model building, and normalization. Additionally, static PDFs provide answers to common questions, such as why certain actions are required and how to perform them. These PDFs also typically include screenshots or images of the various steps taken during the project to provide a visual representation of the process.

4.2 traceable PDFs have two columns: time with date and description. These PDFs save every moment a user performs within the project, along with the corresponding timestamp and a brief description of the action taken. This type of PDF is particularly useful when multiple users are working on a project and need to keep track of the changes made over time.

The final step of a typical project involves prediction, where users are provided with the option to download a CSV file containing data in different sheets. This file is an essential tool for users as it helps them understand the entire data processing pipeline that was utilized while building the model.

The CSV file is structured in seven different sheets. The first sheet contains the raw data that was loaded during the first step of the project. The second sheet contains feature data, while the third sheet contains target data. These two sheets are crucial as they provide insight into the data used to build the model.

The fourth and fifth sheets contain feature train and feature test data, respectively. These sheets represent how the data was split into training and testing sets to evaluate model performance. The sixth and seventh sheets contain target train and target test data, respectively. These sheets provide a similar split of the target data used for evaluation.

5) The structure of the CSV file provides a comprehensive understanding of the data processing pipeline involved in building the model. It helps users to verify that the data was pre-processed correctly and that the correct split was used for training and testing.

Additionally, it allows users to reproduce the results obtained during the model building For users: Runs smoothly without requiring a powerful processor or graphics card, large amounts of RAM or hard disk space. A system with at least 4 GB of RAM and a multi-core processor is required so that performance is not impacted or slowed down.phase.For users: Runs smoothly without requiring a powerful processor or graphics card, large amounts of RAM or hard disk space. A system with at least 4 GB of RAM and a multi-core processor is required so that performance is not impacted or slowed down.

6) Workflow of the application

i. Our project methodology aims to implement an application in a step-by-step manner to achieve an optimized machine learning model building process.
ii. The trained model will then be utilized for dataset prediction and classification, with the flexibility to use any dataset for the model building process.
iii. Once completed, the user will receive the output in the form of a traceable PDF and CSV file.
iv. The following flowchart illustrates the detailed process of our project application. It outlines the sequential steps that need to be followed during the processing of the dataset.
v. Firstly, the dataset is uploaded and the unwanted columns are removed. Then, we handle missing values by using various techniques such as mean, mode, median, and deleting rows.
vi. Additionally, we check for any missing values in the table. Next, we convert categorical values to numerical values. Following this, we detect and remove outliers from the dataset. These are just a few of the steps involved in the process, and there are other crucial steps that need to be executed as well.

III. IMPLEMENTATION

Uploading Csv – Initially we upload the dataset, that can be any kind of dataset, we will be processing, user has to upload the dataset in the given .csv format and if not uploaded it will give error of format, then after we will be processed for handling missing values. The below diagram shows the interface for uploading the dataset in the UI, which will be send to the api request handler, and the to the server which will do all the preprocessing and make the dataset clean without any human intervention, just we have to follow up with the process

Figure 3. Dataset Uploading UI

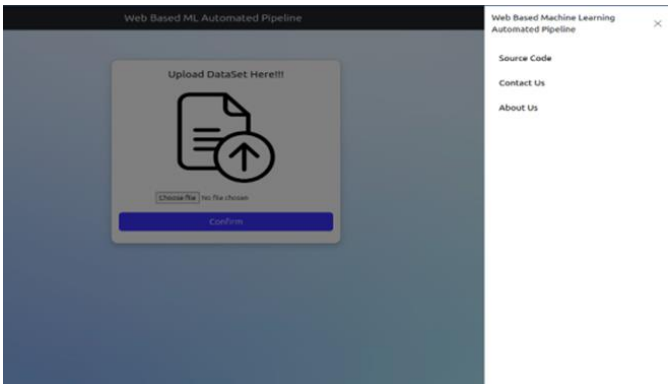


Figure 4. Source code and About us page

Handling missing data in a dataset is the next step in data preprocessing. Missing data in datasets can cause serious problems for machine learning models. As a result, the records contain missing values that need to be addressed.How to handle missing data. apps from several vendors, speaking different languages on various platforms, communicating with clients, and exchanging business processes throughout the network.Apps built in languages accessible by web browsers and accessed via a network are referred to as web applications (e.g., HTML, JavaScript). Web applications, which include many well-known ones like webmail, online retail sales, and online auctions, rely on Web browsers to be executed.User can see the description and

code for the same. And the step is not of use they can skip this step.

Figure 5. Handling Missing Values

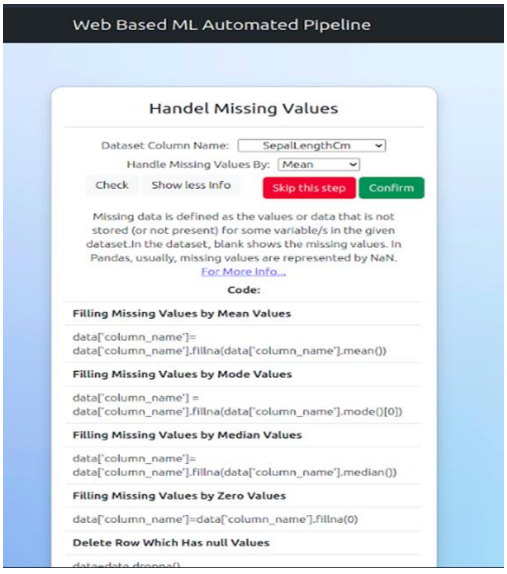


Figure 6. Handling Missing Values description

Remove unwanted Columns: Identify the columns you want to remove: Look at the dataset and identify the columns that are not useful for the machine learning task you are trying to perform.[2] For example, if you are building a model to predict house prices, the columns containing the street address or the owner's name may not be relevant.Select the relevant columns: Once you have identified the unwanted columns, select the relevant columns you want to keep. You can either manually remove the unwanted columns or use a Python library like pandas to do this. User can see the description and code for the same. And the step is not of use they can skip this step.

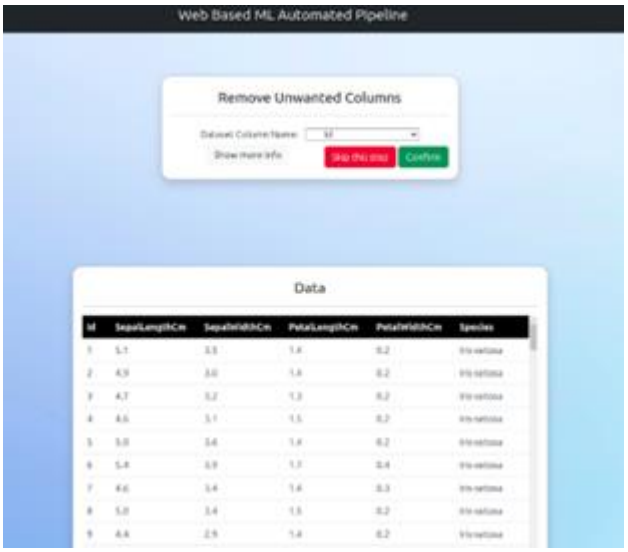


Figure 7. Remove Unwanted Columns

Categorical data refers to data that is divided into several categories or groups, such as country or purchased items in a dataset. In a dataset, there may be multiple categorical variables, each of which can take on a finite set of values or categories. For instance, the variables "Country" and "Purchased" can both be categorical variables with their respective categories.Feature scaling is a data preprocessing technique that aims to standardize the range and size of variables so that one variable does not dominate over others in a dataset. In other words, it aims to align the variables to the same scale to facilitate accurate comparisons between them. For instance, in the given dataset, the variables "Age" and "Salary" are not scaled in the same way, which can make comparisons difficult.

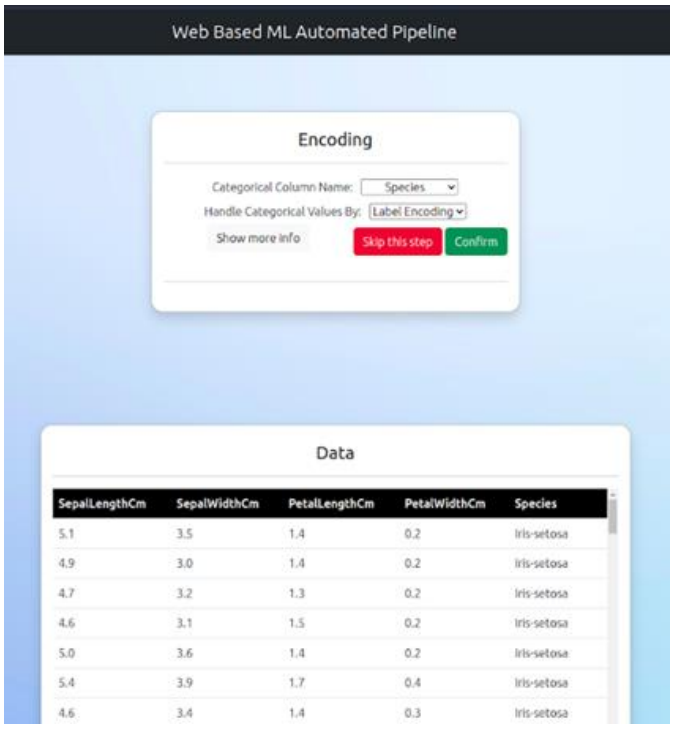


Figure 8. Categorical Data Encoding

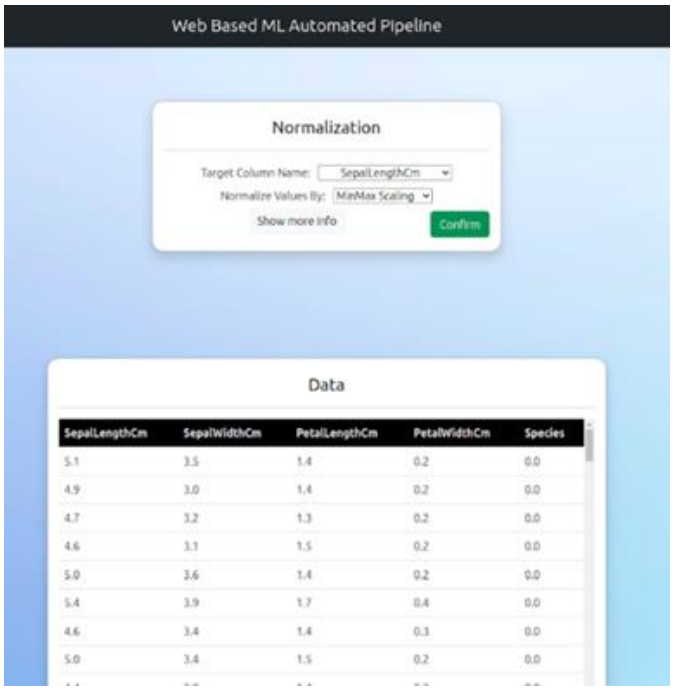


Figure 9. Normalization

Outliers can have a significant impact on the performance of machine learning models, as they can skew the training process and result in models that are less accurate [4]and less generalizable.

There are several techniques for detecting outliers in machine learning, including:

Z-Score Method: This method involves calculating the z-score of each data point, which measures how many standard deviations the data point is from the mean. Data points with a z-score greater than a certain threshold (usually 3 or 4) are considered outliers.

Interquartile Range Method: This method involves calculating the interquartile range (IQR) of the dataset, which is the range between the 25th and 75th percentiles. Data points that fall outside a certain range (usually 1.5 times the IQR) are considered outliers.

Figure 10. Detect and Remove Outliers

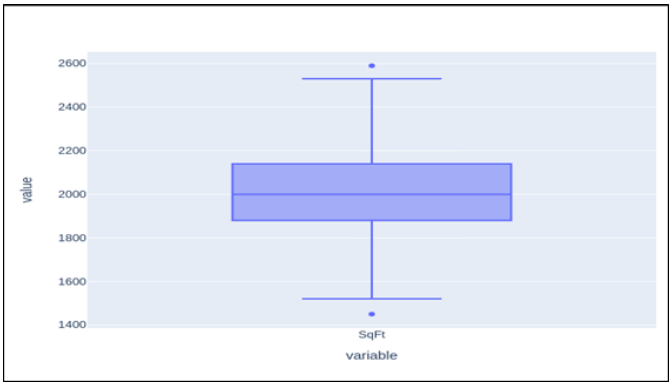


Figure 11. Outliers

Train-Test Split :- The training and testing of the data is to performed so that we divide the dataset into parts for testing and training the dataset differently. [1]Training dataset is used to train the model and testing is used to predict the output of the data.By

splitting the data into a training set and a test set, you can estimate the performance of the model on new, unseen data. This can help you avoid overfitting, where the model performs well on the training data but poorly on the test data. Train-test split is an important step in the machine learning workflow that helps ensure the model's generalizability. User can see the description and code for the same. And the step is not of use they can skip this step.As well as user can see the dataset after splitting into test and train.

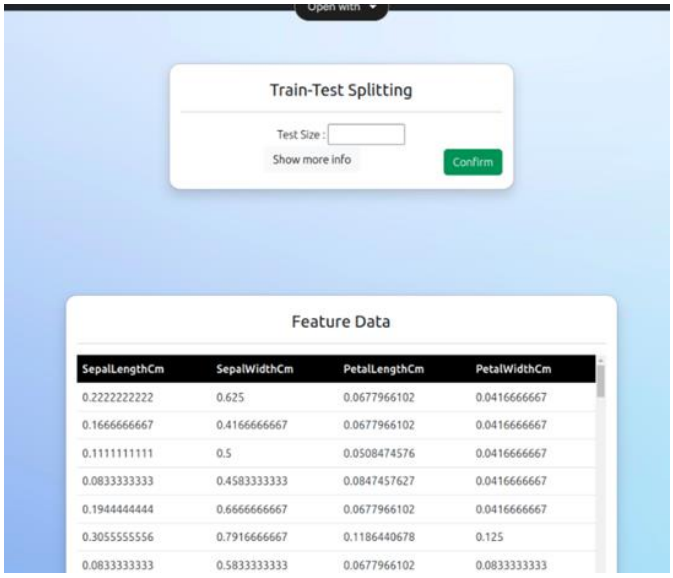


Figure 12. Train Test Splitting

Model Building:- During Model Building[1] we use several model like linear regression, logistic regression, decision tree, random forest, and support vector machine are the models which we used for model training and meanwhile we test it in further process.

The process of building a model in machine learning involves several steps, from defining the problem to deploying the model in production. Each step is important, and a thorough understanding of the problem, the data, and the algorithm is necessary for building an accurate and reliable model.

Figure 13. Model Building

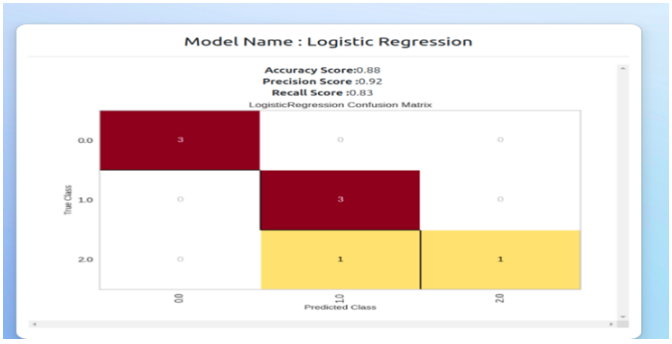


Figure 14. model output

Model Prediction :- [5]After the model Building the model which is trained is tested. We test it with the different model and compute the results. After that we are ready to download the tracable pdf and the available csv.Collect the input data: Collect the input data that you want to make predictions on.Preprocess the input data: Preprocess the input data so that it is in the same format as the data used to train the model. This may involve scaling, normalization, or other preprocessing techniques. Load the trained model: Load the trained model into memory.

Figure 15 Model Prediction

Generating pdf and tracable csv:- we get the output with the tracable pdf and csv with all the target attributes, and we can compute the differnet models of the dataset.Static PDFs are often used to document static data, such as the theoretical aspects of different processes involved in a project, including tasks like loading datasets, handling missing values, model building, removing unwanted columns, normalization, and dealing with outliers. In addition to providing.

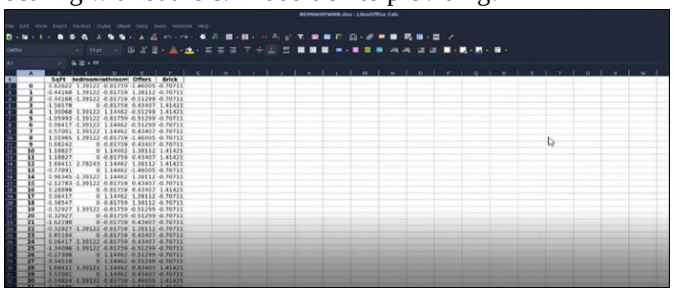
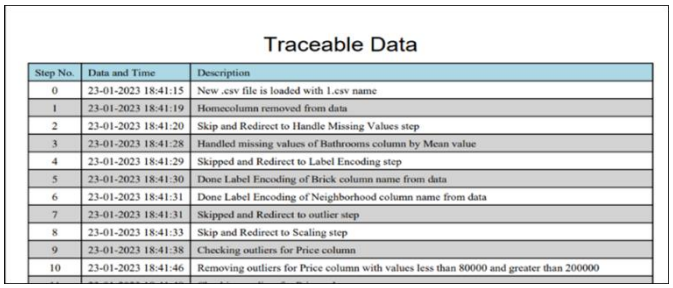


Figure 16. Csv Output data

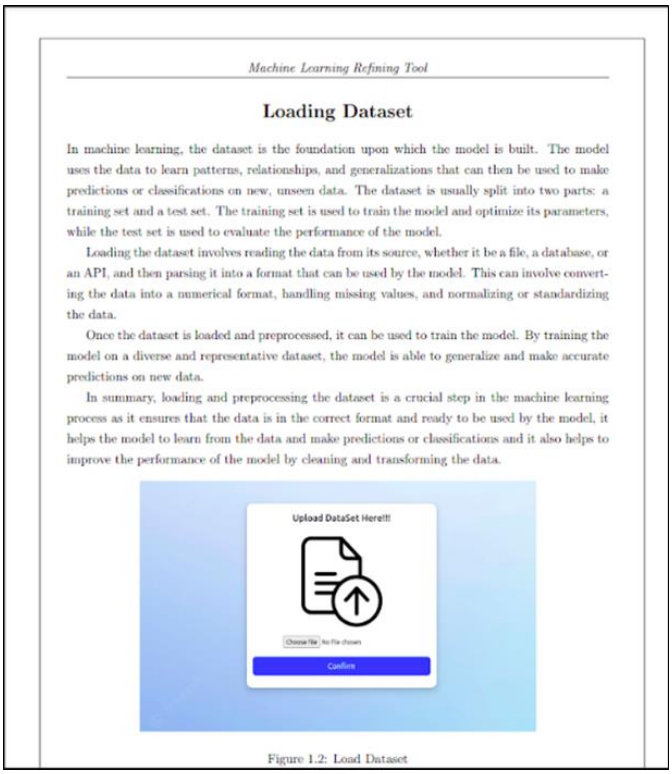


Traceable Data

Step No.	Data and Time	Description
0	23-01-2023 18:41:15	New .csv file is loaded with 1.csv name
1	23-01-2023 18:41:19	Homecolumn removed from data
2	23-01-2023 18:41:20	Skip and Redirect to Handle Missing Values step
3	23-01-2023 18:41:28	Handled missing values of Bathrooms column by Mean value
4	23-01-2023 18:41:29	Skipped and Redirect to Label Encoding step
5	23-01-2023 18:41:30	Done Label Encoding of Brick column name from data
6	23-01-2023 18:41:31	Done Label Encoding of Neighborhood column name from data
7	23-01-2023 18:41:31	Skipped and Redirect to outlier step
8	23-01-2023 18:41:33	Skip and Redirect to Scaling step
9	23-01-2023 18:41:38	Checking outliers for Price column
10	23-01-2023 18:41:46	Removing outliers for Price column with values less than 80000 and greater than 200000

Figure 17. Tracable pdf

Static PDFs also typically include screenshots or images of the various steps taken during the project to offer a visual representation of the process.



Machine Learning Refining Tool

Loading Dataset

In machine learning, the dataset is the foundation upon which the model is built. The model uses the data to learn patterns, relationships, and generalizations that can then be used to make predictions or classifications on new, unseen data. The dataset is usually split into two parts: a training set and a test set. The training set is used to train the model and optimize its parameters, while the test set is used to evaluate the performance of the model.

Loading the dataset involves reading the data from its source, whether it be a file, a database, or an API, and then parsing it into a format that can be used by the model. This can involve converting the data into a numerical format, handling missing values, and normalizing or standardizing the data.

Once the dataset is loaded and preprocessed, it can be used to train the model. By training the model on a diverse and representative dataset, the model is able to generalize and make accurate predictions on new data.

In summary, loading and preprocessing the dataset is a crucial step in the machine learning process as it ensures that the data is in the correct format and ready to be used by the model, it helps the model to learn from the data and make predictions or classifications and it also helps to improve the performance of the model by cleaning and transforming the data.

Figure 1.2: Load Dataset

IV. FUTURE WORK

1) Scaling: Depending on the size of the CSV data, the project may need to scale to handle large datasets. This can be achieved by optimizing the code, using parallel processing, and leveraging cloud computing.

2) Automation: If the project is successful, it may be integrated into a larger system for automation. This could include automating the data collection process, scheduling the pipeline, and integrating the pipeline with other systems.

3) Integration: The project can be integrated with other tools such as data visualization tools, dashboards, and APIs to provide more value to the end-users.

4) Optimization: The project can be optimized to improve the accuracy of the machine learning models. This can include optimizing hyperparameters, feature selection, and data cleaning.

5) Extension: The project can be extended to handle different types of data inputs and machine learning models. This can help to expand its applicability to different use cases and domains.

V. CONCLUSION

As the project progresses, it effectively improves the visualization of the material. The integrity and consistency of the material is maintained throughout the request response cycle. Provides an important feature that is easy to use when analyzing data and provides meaningful information when extracting data.We present a flow-based view of services through case studies and an overview of the business. One may argue that at the first stages of design, flow-based conceptualization promises to provide Web application development with a more stable basis. This flow-based approach may be used with modern software development methodologies.

VI. REFERENCES

[1]. I. F. Qayyum and D.-H. Kim, “A Survey of datasets, preprocesssing, modelling mechanisms,” 2022.
[2]. T. Petrou, “Pandas Cookbook”.
[3]. J. Grus, “Data Science from Scratch”.
[4]. IEEE, “A dataset of attributes from papers of a machine learning conference Algorithm,” 2019.
[5]. IEEE, “Missing Data Analysis in Regression,” 2022.
[6]. IEEE, “A survey on outlier explanations,” 2022.
[7]. S. Raschka and V. Mirjalili, “Python Machine Learning”.

Cite this article as :
Prof. Sachin Sambhaji Patil, Mahesh Manohar Sirsat, Ajitkumar Vishwakarma Sharma, Aashish Shahi, Omkar Maruti Halgi , "Web Based Machine Learning Automated Pipeline", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 10 Issue 3, pp. 43-51, May-June 2023. Available at doi : https://doi.org/10.32628/IJSRSET231035
Journal URL : https://ijsrset.com/IJSRSET231035