IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

SUPERVISED TEACHABLE MACHINE

Ayush Sachin Walekar
MIT-ADT University

Bhavik Prakash Bafna
MIT ADT University

Ninad Pravin Deshmukh
MIT ADT University

Lalit Padma Kartik Saridevi
MIT ADT UNIVERSITY

Under the guidance of
Prof. Priya Khune
MIT-ADT University

ABSTRACT

Our system is designed to make developers easy in creating their own models through a systematic process. Firstly, it focuses on data preprocessing, a crucial step that lays the foundation for model performance. By meticulously cleaning, transforming, and organizing data, developers can ensure the highest quality input for their models. The main innovation comes with the ability to experiment with a diverse range of models. This platform gives a selection of cutting-edge algorithms, each with its own unique strengths. Developers can very easily switch between these models, adjusting parameters on the fly to fine-tune their performance. This dynamic model experimentation allows developers to maintain a detailed record of which models excel under specific conditions. Through extensive tracking, they can identify patterns and insights that guide them towards the most effective models for their specific tasks. This process of iterative optimization enables developers to craft models that not only meet but exceed their expectations, ultimately enhancing the efficiency and efficacy of their data-driven applications.

Keywords

Machine Learning, HDFS, Graphical User Interface ,User Interface, HyperText Markup Language, Cascading, Deep Learning, Search Engine Optimization

1. INTRODUCTION

Our framework is committed to enabling designers by giving them an organized interaction for making their own AI models. It puts an underlying accentuation on information preprocessing, an essential step that fills in as the bedrock for accomplishing ideal model execution. Through careful information cleaning, change, and association, designers can guarantee that their models get the best information. What genuinely separates our foundation is its special capacity to work with trial and error with a different scope of state of the art models. We offer a determination of best in class calculations, each having its own particular assets and qualities. Designers can undoubtedly switch between these models, making ongoing acclimations to boundaries to adjust their presentation. This unique model trial and error permits designers to keep an exhaustive record of how each model performs under unambiguous circumstances. Broad following and investigation of these model trials empower designers to distinguish repeating examples and gain important bits of knowledge. These experiences, thus, act as directing signals, driving engineers toward choosing the best models for their particular undertakings. This iterative advancement process engages engineers to make models that meet as well as outperform their underlying assumptions.A definitive objective is to upgrade the proficiency and viability of their information driven applications by utilizing the maximum capacity of our foundation. Our framework isn't simply an instrument; a complete arrangement upholds engineers in their journey to make remarkable AI models.

In this complex cycle, information preprocessing is the basic step, guaranteeing that the information taken care of into the models is of the greatest quality and consistency. This involves cleaning the information to eliminate errors, changing it into a reasonable configuration, and sorting out it for simple access and use. Following this primary stage, designers are given a variety of state of the art AI calculations, each custom-made to resolve explicit issue spaces and difficulties. These calculations cover a wide range of methods, from old style ways to deal with the most recent progressions in the field.One of the most amazing highlights of our foundation is the capacity to flawlessly switch between these models, changing their boundaries on a case by case basis to accomplish the best outcomes. Engineers can analyze uninhibitedly, permitting them to adjust the models' way of behaving to match the subtleties of their specific information and application.What really separates our foundation is the capacity to track and record these model analyses persistently. This implies that designers can keep an itemized record of how various models perform under different circumstances and settings. This abundance of information is an important asset for improving their AI abilities and upgrading model execution. Through this iterative course of enhancement and trial and error, designers can go past simply measuring up to their assumptions; they can reliably surpass them. This iterative methodology takes into account the formation of models that are compelling as well as profoundly productive, at last adding to the progress of information driven applications.

2. Review of Literature Survey

1. This paper surveys the approaches that generate source code automatically from a natural language description. We also categorize the approaches by their forms of input and

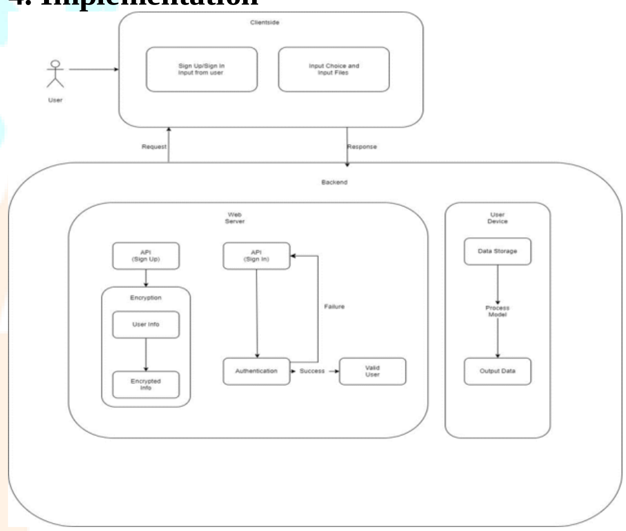
output. Finally, we analyze the current trend of approaches and suggest the future direction of this research domain to improve automatic code generation with natural language

2. Most present-day user-facing software programming applications are Graphical User Interface (GUI) driven, and depend on an alluring User Interface (UI). But implementing GUI code is, however, time-consuming and prevents developers from dedicating the majority of their time to implementing the actual functionality and logic of the software they are building. The project is from a single GUI image as input to generate computer UI code, using Deep Learning Techniques. To train the model on different data sets for different effective output codes. We want to build a neural network that will generate HTML/CSS markup that corresponds to a screenshot.

3. Features and Functionalities of Teachable Machines

1. Data Input and Collection: Teachable Machines offers flexible data input and collection options. Users can upload their datasets via an API or an upcoming web interface. It also supports multiple upload formats, including CSV and Excel. This variety ensures that users can easily bring in their data from different sources.
2. Data Cleaning: Teachable Machines prioritizes data quality. It employs advanced data cleaning techniques to handle issues like missing values, duplicates, and outliers. The system aims to automate the data cleaning process to save users time and effort, but it also allows for user customization, recognizing that different datasets may have specific requirements.
3. Preprocessing and Transformation: This platform goes beyond cleaning to provide comprehensive preprocessing and transformation features. Users can perform tasks such as normalization, standardization, and categorical variable encoding to prepare their data for analysis. Feature engineering capabilities are also available, allowing users to create new variables tailored to their dataset. The user-friendly interface makes customization straightforward.
4. Visualization and Summary: To aid in data understanding, Teachable Machines generates visualizations such as box plots, scatter plots, and correlation matrices. These visual representations help users uncover patterns and relationships in their data. Additionally, the system provides essential dataset statistics, including mean, median, and standard deviation for numerical features, as well as frequency counts for categorical features, enhancing comprehension.
5. Model Training and Predictions: Integration of machine learning models is a key feature. Users can perform tasks like automatic feature selection and data imputation, and they can train models on their cleaned data. This facilitates predictions and classifications. The platform also provides model evaluation metrics, allowing users to assess the performance of their models.
6. User-Friendly Interface: Teachable Machines places a strong emphasis on user experience. The platform is designed with an intuitive web interface that includes clear instructions, tooltips, and guidance. This user-friendly design assists users throughout the data cleaning and preprocessing process. Moreover, interactive data visualization tools empower users to explore their data and derive insights.
7. Customization and Configuration: Recognizing the diversity of data requirements, Teachable Machines allows users to configure the system to meet their specific needs. It caters to both experienced users and beginners by offering advanced settings for those who require them while providing basic options for users new to data analysis.
8. Data Export and Integration: Users can easily export their cleaned datasets in various formats, including CSV and Excel, ensuring compatibility with their preferred analysis tools. The platform is also considering the implementation of APIs and integration with popular data analysis tools like Python, R, or Jupyter notebooks to offer versatility in data handling and analysis.

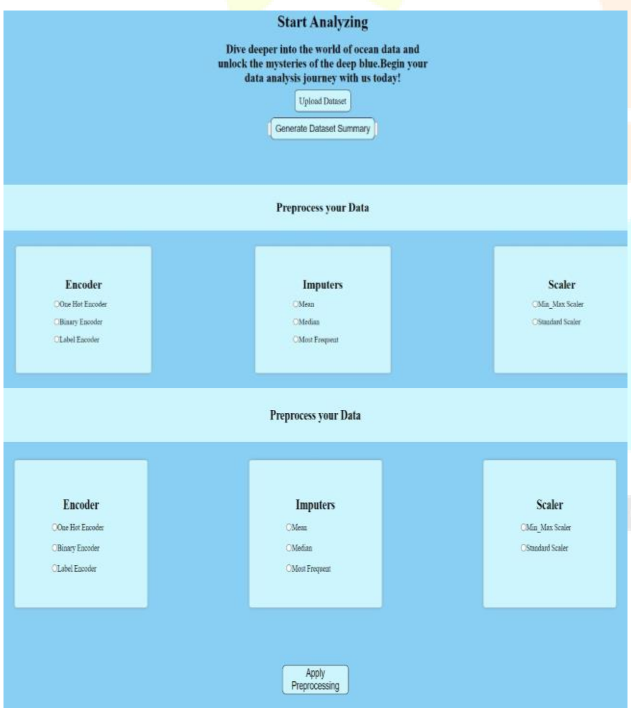
4. Implementation



1. Data Input and Collection: Users can upload their datasets through an API, with a web interface planned for the future. Multiple data upload formats, including CSV and Excel, are supported.
2. Data Cleaning: Incorporates advanced data cleaning techniques to address missing values, duplicates, and outliers. Aims to automate the data cleaning process to save users time and effort. Provides options for users to customize the cleaning process according to their specific requirements.
3. Preprocessing and Transformation: Offers data preprocessing steps such as normalization, standardization, and categorical variable encoding. Includes feature engineering capabilities to create new variables tailored to the dataset. A user-friendly interface enables easy customization of preprocessing steps.
4. Visualization and Summary: Generates visualizations, including box plot, scatter plots, and correlation matrices, to aid in data understanding. Summarizes dataset statistics, such as mean, median, and standard deviation for numerical features, and frequency counts for categorical features.
5. Model Training and Predictions: Integrates machine learning models to facilitate automatic feature selection and data imputation. Enables users to train models on cleaned data and evaluate model performance using relevant metrics.
6. User-Friendly Interface: Focuses on creating an intuitive web interface for ease of interaction with the system. Incorporates clear instructions, tooltips, and guidance to assist users throughout the data cleaning and preprocessing process. Provides interactive data visualization tools to empower users to explore data insights.

7. Customization and Configuration: Allows users to configure the system based on their specific data cleaning and preprocessing requirements. Offers advanced settings for experienced users while maintaining basic options for beginners.
8. Data Export and Integration: Permits users to export the cleaned dataset in various formats (e.g., CSV, Excel) for seamless integration with their analysis tools. Considers the implementation of APIs and integration with popular data analysis tools like Python, R, or Jupyter notebooks.

5. Outputs



6. Conclusion

So this teachable machine system will help developers to develop their own model by preprocessing data firstly and then applying different models. By automating data preparation and making it easier to explore different models, it streamlines the procedure. By experimenting with various parameters and methods, developers may find the ideal combination. This method reduces the requirement for human data manipulation by streamlining the model creation process. By facilitating effective model comparison and tracking, the system helps developers make well-informed judgments. All things considered, it provides a workable option for those who want to use machine learning but lack a lot of experience.

7. References

[1].https://www.thetechedvocate.org/8-ways-machine-learning-will-improve-education, (2018), Accessed on: 2018-09-25

[2]. Luckin, R., Holmes, W., Griffiths, M., & Forcier, L. B. (2016). Intelligence unleashed: An argument for AI in education.

[3]. Roblyer MD, Doering AH (2009) Integrating educational technology into teaching, 5th Edition. Allyn &Bacon, Boston.

[4] Alpaydin, E. (2009). Introduction to machine learning. MIT press.

[5] Kitchenham. B.A., Charters,S. (2007) Guidelines for performing Systematic Literature Reviews in Software Engineering. Version 2.3, EBSE Technical Report EBSE-2007-01, Keele University, Keele, Staffordshire, United Kingdom

[6]. Kitchenham. B.A., Charters,S. (2007) Guidelines for performing Systematic Literature Reviews in Software Engineering. Version 2.3, EBSE Technical Report EBSE-2007-01, Keele University, Keele, Staffordshire, United Kingdom.

[7]. Anozie, N., Junker, B. W. (2006, July). Predicting end-of-year accountability assessment scores from monthly student records in an online tutoring system. Educational Data Mining: Papers from the AAAI Workshop. Menlo Park, CA: AAAI Press.

[8]. Samuel, A.L. (1959) Some Studies in Machine Learning Using the Game of Checkers, IBM Journal of Research and Development, vol. 3, no. 3, pp. 210-229, doi: 10.1147/rd.33.0210