JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

NEURONEX INTELLIGENCE ANALYTICS: A WEB-BASED DATA SCIENCE TOOL FOR CLASSIFICATION, CLUSTERING, AND VISUALIZATION

[1]Pooja Kadam, [2]Aditya Pathak, [3]Aarya Sambare, [4]Aditee Dhondge, [5]Sushil Sonawane
[1]Assistant Professor, [2]Student, [3]Student, [4]Student, [5]Student
[1]Department of Computer Engineering,
[1]MIT School of Computing,
[1]MIT Art, Design and Technology University, Pune, India

Abstract : Neuronex Intelligence Analytics is an advanced web application designed to help students, researchers, and data scientists explore, analyze, and visualize datasets with ease. Built using React.js for the frontend, Node.js for the backend, and Python for implementing machine learning algorithms, Neuronex allows users to perform various operations such as dataset management, algorithm analysis, and data visualization. This paper presents an overview of the system, including its architecture, key features, services, and the machine learning models employed. Additionally, we discuss the results of classification and clustering tasks, highlighting the usefulness of the tool in educational and research contexts.

IndexTerms - Data Science, Web Application, Machine Learning, Data Visualization, Classification, Clustering, Python, React.js, Node.js

---

I. INTRODUCTION

With the growing complexity of data in both academic research and industrial applications, there is an increasing demand for tools that make data analysis accessible to nonexperts. Data scientists, researchers, and students often require platforms that facilitate the exploration, analysis, and visualization of datasets without requiring deep technical expertise in machine learning or programming. Neuronex Intelligence Analytics is designed to address this need by offering a user-friendly web application that allows users to upload datasets, perform machine learning analyses (classification and clustering), and visualize the results with ease. Built using React.js for the frontend, Node.js for the backend, and Python for the machine learning and data processing tasks, the platform simplifies the workflow for users by abstracting the complexities of data science tasks. The objective of Neuronex is to provide an intuitive tool for students, data scientists, and researchers to manage datasets, analyze them using machine learning algorithms, and visualize the results in a format that is easy to interpret. This paper provides a detailed explanation of the architecture of Neuronex, its technology stack, the algorithms implemented, and the potential applications of the tool.

II. LITERATURE SURVEY

Recent years have witnessed a rapid rise in data mining platforms that integrate machine learning and visualization to make analytics accessible. Several systems have been proposed and evaluated in literature:

Han et al. (2012) introduced fundamental concepts in data mining and discussed classification, clustering, and association rules in their seminal textbook [1]. These foundational methods are utilized in Neuronex to implement core ML functionality.

Kelleher et al. (2015) provided a comprehensive introduction to predictive analytics and model evaluation metrics such as accuracy, precision, recall, and F1-score [2], all of which are implemented in the algorithm analysis module of Neuronex.

Weka, developed by Witten and Frank, is a Java-based toolkit for machine learning that offers GUI support for non-programmers [3]. Neuronex expands on this vision by offering web-based accessibility with real-time processing.

Orange is another GUI-based system that supports drag-and-drop workflows for ML pipelines. Demsar et al. (2013) evaluated its use for education and rapid prototyping [4]. However, Orange lacks integration with the web and data pipeline customizations.

Tableau, while known for powerful data visualization, focuses solely on visual output and doesn't provide end-to-end ML analysis. Recent studies show its limitations in algorithm interpretability and pipeline customization [5].

Google AutoML introduced scalable ML via APIs and auto-selection of algorithms. While effective, it is often inaccessible to student researchers due to cost and complexity [6]. Neuronex offers a cost-free, educational-friendly alternative.

Jupyter Notebooks, widely used in academia, are praised for flexibility but not user-friendly for non-technical users. Pimentel et al. (2019) identified reproducibility and interactivity issues in notebooks for large collaborative projects [7].

KNIME is an open-source analytics platform that supports visual workflows and Python integration. It has proven useful in industry settings but lacks the web-first experience provided by Neuronex [8].
Dash by Plotly is a Python framework for building data apps but requires extensive programming knowledge. Becker et al. (2020) note the steep learning curve despite its powerful backend features [9].
MIT's WebDataRocks presented a lightweight solution for embedded data visualizations. However, its limitation lies in the absence of backend data mining support or algorithm integration, which Neuronex covers thoroughly [10].

III. System Architecture and Technology Stack

Neuronex was developed with a modular architecture that ensures scalability, flexibility, and performance. The system includes three main components:
• Frontend: React.js-based UI for an interactive and responsive user experience.
• Backend: Node.js for handling business logic and API communication.
• Machine Learning Engine: Python for implementing and executing machine learning models.

A. **Frontend (React.js)** The frontend is developed using React.js, a powerful JavaScript library for building user interfaces. React enables the creation of interactive and dynamic components, such as forms for dataset uploading and real-time visualizations for data analysis. React's component-based architecture allows for efficient updating of parts of the UI without requiring the entire page to reload, ensuring a fast and responsive experience for users. Key features of the frontend include:
  • Dataset Uploading: Users can easily upload datasets in CSV or Excel format via a drag-and-drop interface.
  • Visualization Interface: Interactive charts and graphs, such as bar charts, pie charts, line charts, and radar charts, allow users to visualize their data in various formats.
  • Data Management: Users can view, delete, and update their datasets directly from the dashboard. • Responsive Design: The web application is optimized for various devices and screen sizes, allowing users to access it from desktops, tablets, and smartphones.

B. **Backend (Node.js)** Node.js powers the backend of Neuronex, offering a non blocking, event-driven architecture that is perfect for handling a large number of concurrent requests efficiently. Node.js is lightweight and scalable, making it an ideal choice for developing real-time applications. Key functions of the Node.js backend include:
  • API Endpoints: The backend exposes RESTful APIs for dataset management (upload, delete, view) and triggering machine learning tasks (classification, clustering).
  • Data Processing: The backend interacts with the Python based machine learning engine to send data for analysis and receive results for visualization.
  • User Authentication: Provides secure login and user management to ensure that each user's data and analysis are isolated.

C. **Machine Learning Engine (Python)** The Python engine handles all the machine learning tasks, including data preprocessing, training models, and evaluating results. Python was chosen for its rich ecosystem of data science libraries, such as scikit-learn, pandas, and numpy, which facilitate efficient data analysis and model implementation. Key features of the Python engine include:
  • Classification Algorithms: Decision Tree, Logistic Regression, Naive Bayes, KNN, and Random Forest are implemented to handle supervised learning tasks.
  • Clustering Algorithms: Agglomerative Clustering, DB SCAN, and BIRCH are available for unsupervised clustering tasks. • Model Evaluation: After training models, evaluation metrics such as accuracy, confusion matrices, and classification reports (precision, recall, F1-score) are generated for classification tasks.
  • Visualization Support: Uses libraries like Matplotlib and Seaborn to create visualizations for data exploration and results interpretation.

IV. Machine learning algorithms

Neuronex implements several machine learning algorithms, categorized into classification and clustering tasks.
A. Classification Algorithms: Neuronex includes five popular classification algorithms that provide users with the ability to analyze data based on known labels.
1) Decision Tree: Decision Trees are a powerful algorithm for classification tasks. They split the dataset into smaller subsets based on feature values, creating a tree-like structure where the decision nodes correspond to the features and the leaves represent the output class.
2) Logistic Regression: Logistic Regression is used for binary classification problems. It models the probability that a given input point belongs to a particular class using the logistic function.
3) Naive Bayes: Naive Bayes classifiers are based on the Bayes' theorem, assuming that the features in the dataset are conditionally independent. It is widely used in text classification, such as spam detection.
4) K-Nearest Neighbors (KNN): KNN is a non-parametric method used for both classification and regression. It assigns the class of the majority of the k-nearest neighbors to the test point.
5) Random Forest: Random Forest is an ensemble method that constructs multiple decision trees and aggregates their predictions. It is robust to overfitting and can handle high dimensional data effectively.
B. Clustering Algorithms: Clustering algorithms are used for unsupervised learning, where the goal is to group data points into clusters based on their similarity.
1) Agglomerative Clustering: Agglomerative clustering is a bottom-up approach to clustering. It starts by treating each data point as its own cluster and iteratively merges the closest clusters based on a distance metric.
2) DBSCAN: DBSCAN is a density-based clustering algorithm that groups together data points that are closely packed, marking outliers as noise. It does not require the number of clusters to be specified in advance.
3) BIRCH: BIRCH is an efficient clustering algorithm designed for large datasets. It builds a clustering feature tree and then applies a refinement method to obtain the final clusters.

V. RESULTS AND DISCUSSIONS

In this section, we present the results obtained using Neuronex for both classification and clustering tasks.

A. *Dashboard*

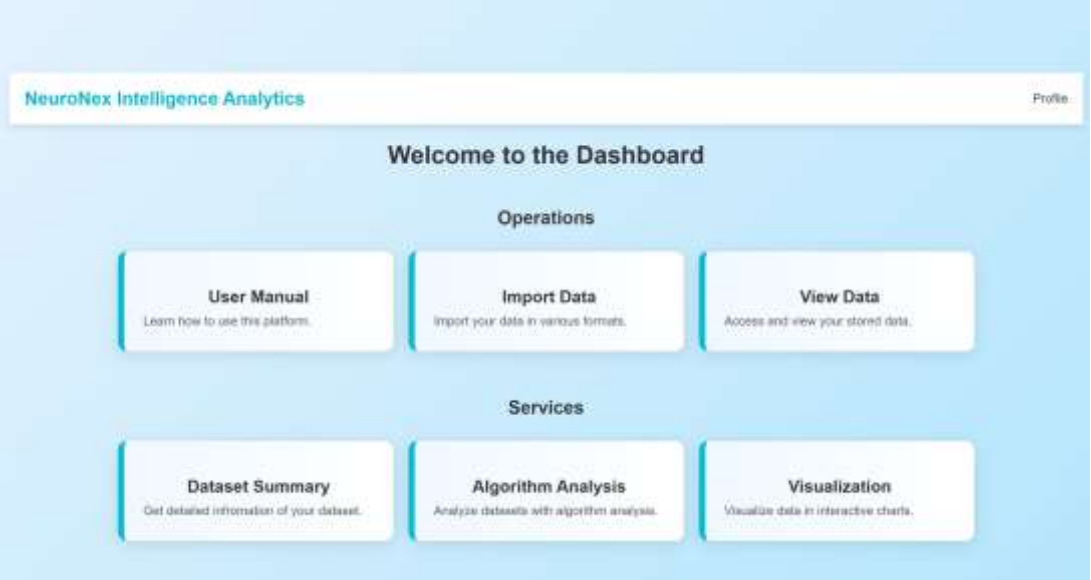

Fig. 1. Neuronex Dashboard: Overview of all available operations and services.

B. Classification Results

The classification models were evaluated on multiple datasets, and their performance was measured in terms of accuracy, precision, recall, and F1-score. The confusion matrix and classification report were used to evaluate the results.

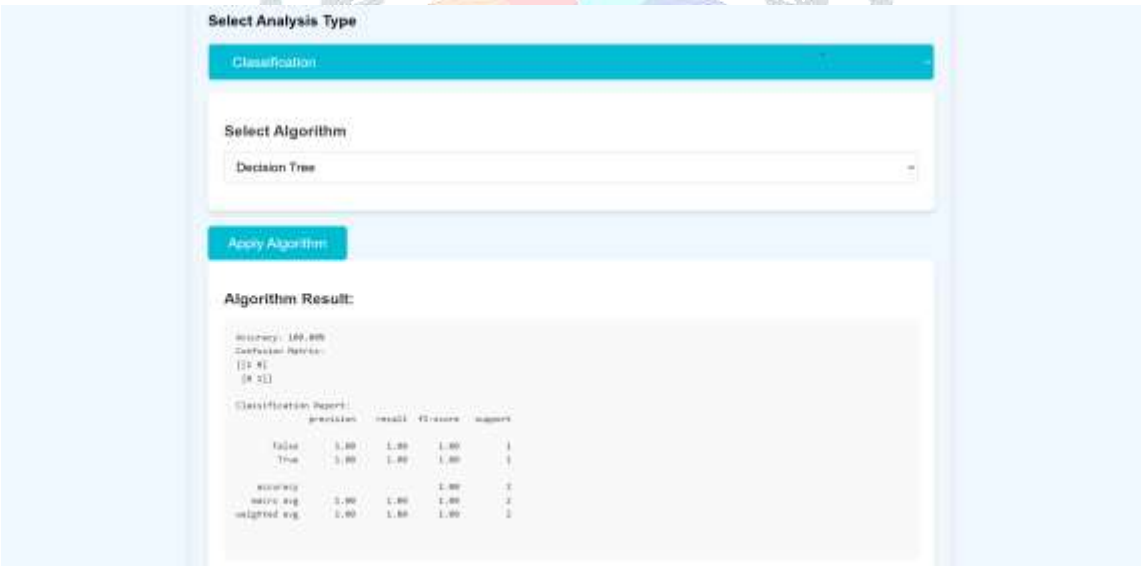

Fig. 2. Example of Classification Results Visualized as a Confusion Matrix.

C. Clustering Results

The clustering algorithms were applied to different datasets to analyze the distribution of data points into clusters. The following radar chart shows the distribution of clusters for a sample dataset.

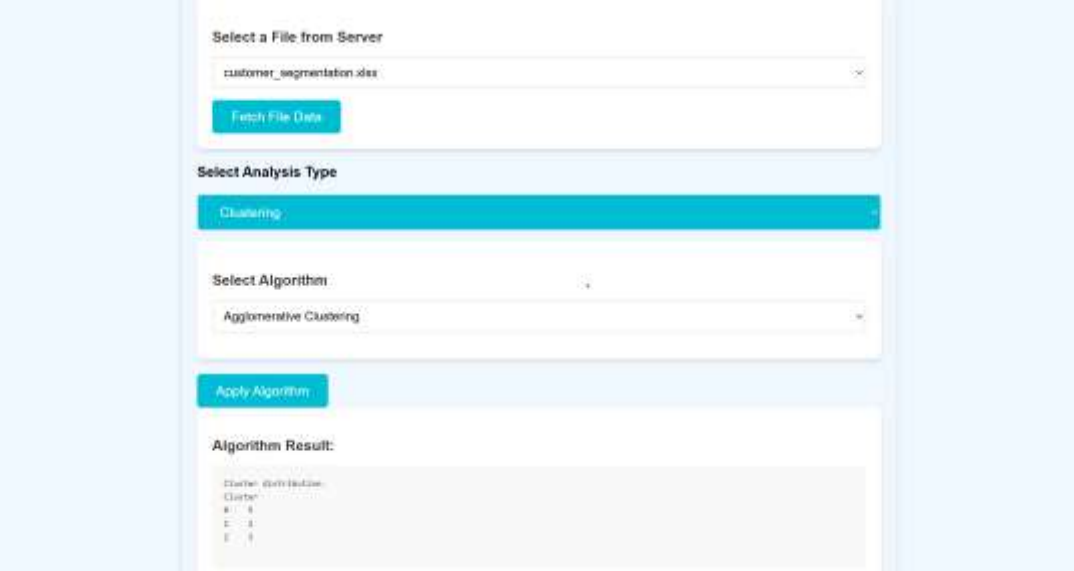

Fig. 3. Clustering Results Displayed with a Radar Chart.

D. Dataset Summary

The Dataset Summary feature provides an overview of key statistics for the uploaded dataset, such as the number of rows, columns, missing values, and statistical measures like mean, mode, and median.

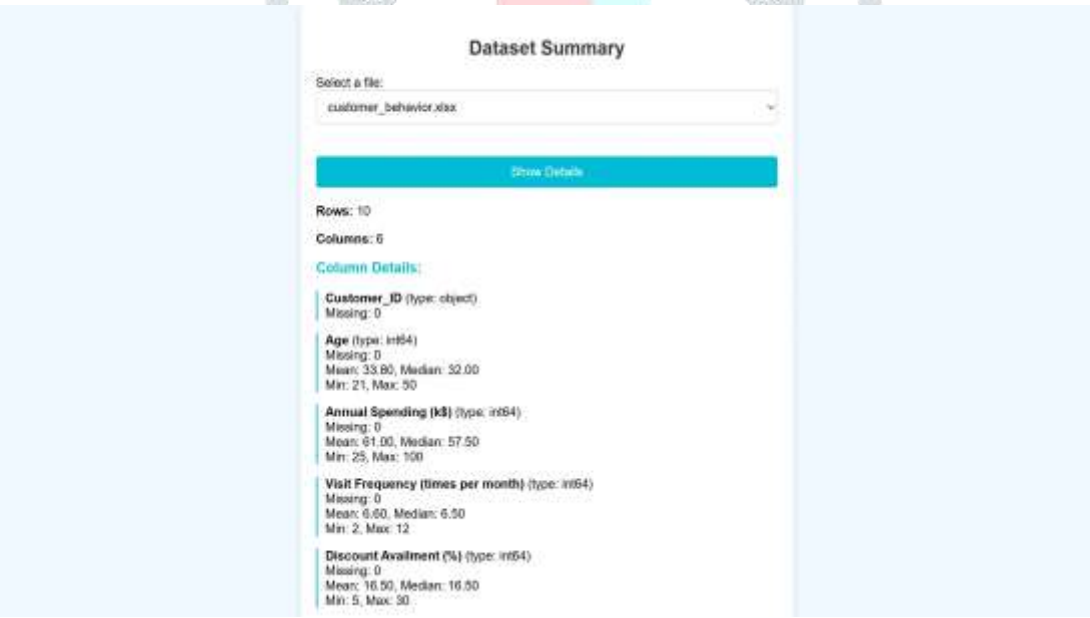

Fig. 4. Dataset Summary: Overview of the dataset's characteristics (rows, columns, missing values).

E. Import Data

The Import Data feature allows users to upload datasets in CSV or Excel format. This is a key feature that makes the system adaptable to various types of data.

Fig. 5. Import Data: User Interface for uploading datasets in CSV/Excel format.

F. Line Chart Visualization

The Line Chart feature provides a clear visualization of trends in data over time, making it easy for users to track changes in the dataset.

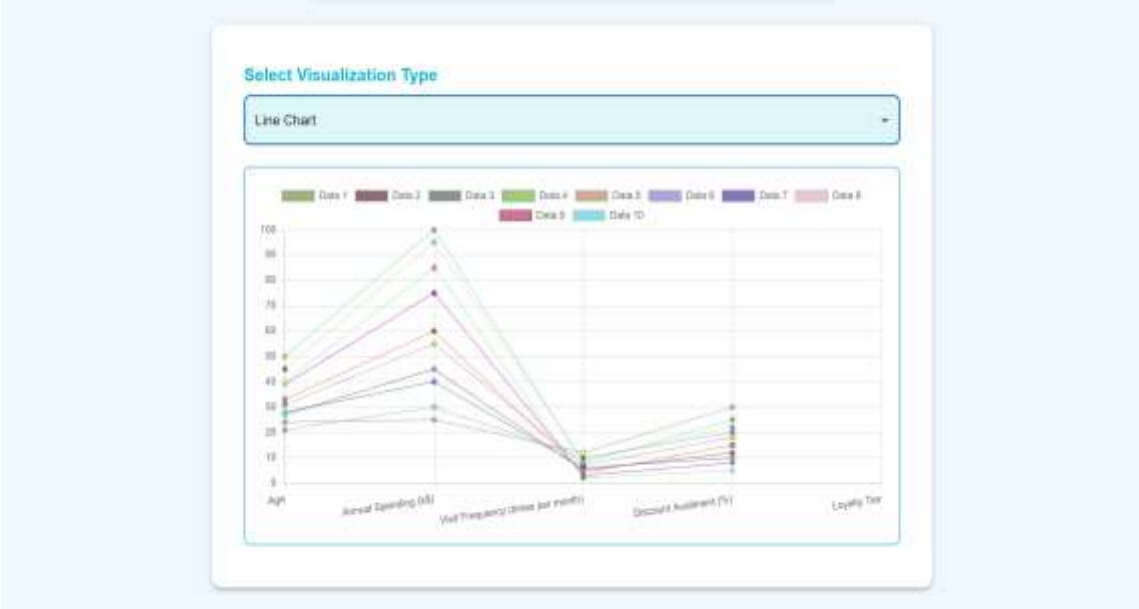

Fig. 6. Line Chart Visualization: Representation of data trends over time.

G. Bar Chart Visualization

The Bar Chart feature compares various categories of data, providing a clear and visually appealing way to analyze categorical data distributions.

Fig. 7. Bar Chart Visualization: Comparison of different categories in the dataset.

VI. CONCLUSION AND FUTURE WORK

Neuronex provides an accessible and intuitive platform for students, data scientists, and researchers to perform machine learning tasks and visualize results. The application integrates the latest machine learning algorithms with a user-friendly interface to simplify data analysis workflows. Future improvements will include adding more machine learning algorithms, enhanced visualizations, and deploying the platform for cloud- based collaborative research.

VII. ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to the following individuals and groups for their invaluable support:

- **Prof. Pooja Kadam**, Project Guide, for her guidance and mentorship throughout the project.
- **Prof. Dr. Hyder Ali Hingoliwala.** RWSC course instructor for providing guidance on academic writing and the publication process.
- **Prof. Dr. Jayshree Prasad**, Head of Department, for her continuous support and encouragement.
- **Prof. Dr. Ganesh Pathak**, for his technical assistance and advice on implementing machine learning algorithms.
- The entire **MIT ADT University**, for providing a conducive environment for research and development.

VIII. REFERENCES

[1] Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. Elsevier.
[2] Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2015). *Fundamentals of Machine Learning for Predictive Data Analytics*. MIT Press.
[3] Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.
[4] Demšar, J., Curk, T., Erjavec, A., et al. (2013). Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research*, 14, 2349–2353.
[5] Mortari, F., & Lucini, F. R. (2020). Challenges in Using Tableau for Educational Visual Analytics. *IEEE Transactions on Education*.
[6] Liu, Q., Zhou, Y., & Wang, X. (2019). AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems*, 212.
[7] Pimentel, J. F., Murta, L., Braganholo, V., & Freire, J. (2019). A Large-Scale Study About Quality and Reproducibility of Jupyter Notebooks. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW).
[8] Berthold, M. R., Cebron, N., Dill, F., et al. (2009). KNIME - The Konstanz Information Miner: Version 2.0 and Beyond. *SIGKDD Explorations*, 11(1).
[9] Becker, A., Shah, D., & Gopalan, K. (2020). Building Interactive Data Apps with Plotly Dash. *ACM Crossroads*, 26(4).
[10] Shpak, M., & Nikiforov, A. (2018). WebDataRocks: Lightweight Web-Based Pivot Table for Data Exploration. *International Journal of Web Engineering*.