International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN - 2395-1990
Online ISSN : 2394-4099
OPEN ACCESS
Available Online at :www.ijsrset.com
doi : https://doi.org/10.32628/IJSRSET

A Semi-Supervised Learning Approach for Quality-Based Web Service Classification

Gundluru Hareesh[1], S. Muni Kumar[2]

[1]MCA Student, Department of Computer Science, KMM Institute of Post-Graduation Studies, Tirupati, Tirupati (D.T), Andhra Pradesh, India

[2]Associate Professor, Department of Computer Science, KMM Institute of Post-Graduation Studies, Tirupati, Tirupati (D.T), Andhra Pradesh, India

ARTICLEINFO

Article History:

Accepted : 19 May 2025
Published: 24 May 2025

Publication Issue :
Volume 12, Issue 3
May-June-2025

Page Number :
351-358

ABSTRACT

The "Web Service Classification and Recommendation" project aims to beautify the system of choosing and recommending internet offerings the use of device learning techniques. The venture is split into two primary additives: Classification Model: This model classifies internet offerings into satisfactory categories—Bronze, Silver, Gold, and Platinum—based on key Quality of Service attributes like response time, availability, reliability, and throughput. The version is built the usage of numerous gadgets studying algorithms, inclusive of Decision Trees, Support Vector Machines, Logistic Regression, K-Nearest Neighbors, Naive Bayes, Random Forest, Multi-Layer Perceptrons, and XGBoost. Additionally, Explainable AI is hired to offer transparency and interpretability within the category selections, allowing users to apprehend why a web service falls into a particular satisfactory class. Recommendation Model: Using the K-Nearest Neighbors set of rules, this version recommends the top 10 most applicable web services primarily based on consumer-inputted QoS attributes. It computes the similarity among the enter statistics and present internet services to provide tailor-made tips. The system's person-pleasant interface lets in customers to add internet provider facts, view type consequences, and receive customized hints. The challenge provides an efficient and scalable answer for choosing awesome net services, permitting users to make information-driven choices based totally on performance metrics. With its twin approach—type and advice—this machine improves selection-making, complements user revel in, and supports organizations in optimizing service choice.

Keywords: Web services, Recommendation, classification, KNN.

INTRODUCTION

In nowadays virtual panorama, choosing the most suitable internet service for a specific requirement is a complex venture because of the sizeable variety of available options. Traditional techniques like keyword matching or primary filtering often fall brief, especially while comparing services across diverse first-class metrics together with response time, availability, throughput, and reliability. This task, "Web Service Classification and Recommendation," tackles this task via combining system studying-based totally class with a recommendation system.

Objective Of Project:

The goal of this undertaking is to increase a clever machine that classifies web offerings into great classes (Bronze, Silver, Gold, Platinum) the usage of gadget learning algorithms and Explainable AI (LIME). The system additionally recommends applicable services primarily based on key overall performance metrics, aiming to enhance carrier choice and decision-making accuracy.

Problem Statement:

With rising web services, selecting the best based on quality metrics like response time and reliability is complex. This project offers classification, personalized recommendations, and transparency using Explainable AI.

Motivation:

With growing internet services, customers warfare to locate the proper one. This venture uses gadget mastering and Explainable AI to classify offerings and provide personalized, obvious suggestions for better choice-making.

Scope:

This challenge ambitions to classify and advocate net offerings the usage of performance metrics, gadget learning, and Explainable AI, ensuring accurate provider choice thru scalable, obvious models across diverse enterprise applications.

LITERATURE REVIEW

[1] Web provider category is a vital task for organizing and rating offerings primarily based on their first-rate and performance attributes. Several tactics have been proposed, which include rule-based totally class systems and system learning strategies. Some conventional systems rely upon simple threshold-based totally regulations for categorizing offerings, along with classifying services based on predefined performance benchmarks like reaction time or uptime. However, those structures often fail to provide personalised and dynamic classifications, main to suboptimal decision-making.

[2] Recommendation structures were broadly used throughout various domain names to indicate applicable objects based on person choices. In the context of web offerings, advice structures leverage performance metrics consisting of response time, availability, throughput, and reliability to signify the quality services to customers.

[3] Machine gaining knowledge of strategies, together with K-Nearest Neighbors, Random Forest, Naive Bayes, and XGBoost, have tested to be powerful in both category and advice tasks. These fashions are mainly useful in managing large datasets with more than one first-rate attribute, as they can adapt to various sorts of input and output codecs.

[4] Explainable AI has won significant significance in current years, specifically for programs that require transparency and interpretability. LIME (Local Interpretable Model-agnostic Explanations) is one of the most distinguished XAI strategies used to explain machine getting to know version predictions. In the context of web provider category, LIME allows to offer insights into why a particular provider become categorized into a specific exceptional class, along with Bronze, Silver, Gold, or Platinum. The use of LIME in provider classification fashions enhances user agree with inside the device with the aid of imparting obvious motives for the predictions. Several research have carried out LIME to category

responsibilities in various domain names, demonstrating its capability to enhance the interpretability of complicated fashions like Random Forest and XGBoost.

[5] The effectiveness of an internet service recommendation or category system relies upon in large part on the quality metrics used to assess offerings. Web services are regularly evaluated based totally on key Quality of Service (QoS) attributes, consisting of response time, availability, throughput, reliability, and fulfillment fee. Several studies have focused on defining and amassing relevant QoS metrics that may offer a correct representation of carrier overall performance. Response time, for example, is an essential metric for comparing the responsiveness of net services, while availability and reliability indicate the carrier's uptime and consistency. Some researchers have explored how to dynamically replace these metrics in real-time to mirror converting carrier conditions. Recent advancements have caused the improvement of frameworks that permit for the efficient series and analysis of QoS information, enhancing the reliability of net service classification and recommendation systems.

EXISTING METHOD

Traditional web carrier selection systems rely on fundamental filtering strategies, keyword matching, or rule-based guidelines. These strategies use a constrained range of predefined criteria, consisting of reaction time or availability, to rank net offerings. And additionally, this for classification and recommendation there is no combined device.

METHODOLOGY

The proposed system integrates two key components: a classification model and a recommendation model, both powered by advanced machine learning algorithms. The classification model uses algorithms such as Decision Trees, Support Vector Machines, K-Nearest Neighbors, Random Forest, XGBoost, and others to classify web services into quality categories like Bronze, Silver, Gold, and Platinum. These categories are based on a comprehensive set of quality attributes, including response time, availability, reliability, throughput, and more. The recommendation model uses KNN to recommend the top 10 most relevant web services based on user-provided input data. To ensure transparency, the classification model is augmented with Explainable AI, enabling users to understand why a service was classified into a certain category.

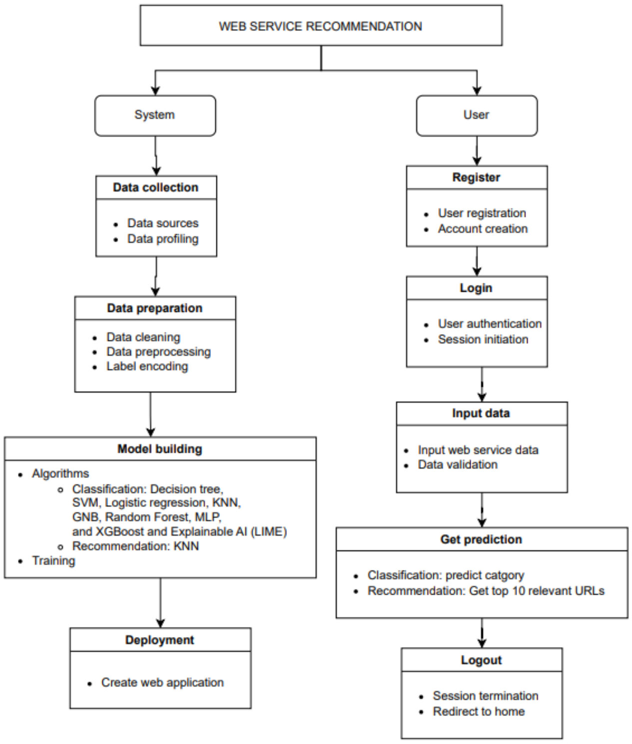


Fig: Flow

In today's digital panorama, choosing the most suitable web carrier for a selected requirement is a complex task due to the sizable number of available alternatives. Traditional strategies like keyword matching or fundamental filtering often fall brief, in particular when comparing offerings across diverse first-rate metrics along with response time, availability, throughput, and reliability. This assignment, "Web Service Classification and Recommendation," tackles this mission by combining machine studying-primarily based category with an advice system.

The middle goal is to classify net offerings into 4 best tiers—Bronze, Silver, Gold, and Platinum—the usage of overall performance metrics. Classification algorithms inclusive of Decision Trees, Support Vector Machines (SVM), and Logistic Regression are applied to institution offerings based totally on best signs. This categorization offers users with a clean expertise of provider reliability and performance.

To enhance usability, a advice engine primarily based at the K-Nearest Neighbors set of rules shows the pinnacle 10 services that align carefully with person-described wishes. For more transparency, Explainable AI techniques like LIME are incorporated, permitting users to understand why a provider become located in a positive category.

Designed to address massive-scale, dynamic datasets, the device is scalable, adaptable, and relevant across domain names like e-commerce, banking, and cloud computing, enhancing both user decision-making and carrier first-class choice.

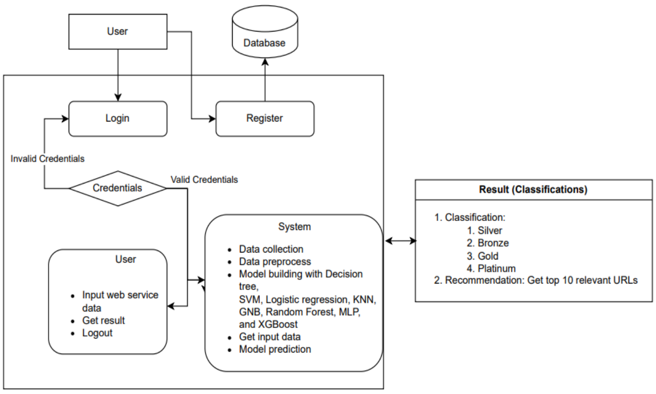


Fig: Architecture

Step 1: Data Collection: The first step includes amassing a dataset that carries great metrics of diverse net offerings. These metrics consist of important features which include Response Time, Availability, Throughput, Success ability, Reliability, Compliance, Best Practices, Latency, and Documentation. This statistic gives the muse for education the advice version.

Step 2: Data Pre-processing: To make sure that the K-Nearest Neighbors model can examine these metrics uniformly, the code applies a Min Max Scaler to normalize the feature values. This scales all values to a 0-1 variety, stopping functions with larger degrees from dominating the distance calculations. Preprocessing steps consist of:

- Selecting relevant functions from the dataset.
- Applying normalization to standardize the statistics.

Step 3: Model Training: The recommendation model is educated the usage of the K-Nearest Neighbors set of rules, applied with Nearest Neighbors from the sklearn Buddies module. The key steps in model training include:

- Initializing the KNN version with neighbors=10 to retrieve the ten maximum comparable services for any given enter.
- Using the in-shape method to educate the version on the pre-processed dataset, where it learns the patterns in the provider metrics.

Step 4: Model Saving: The educated version and the Min Max Scaler are stored the use of the joblib library. This step guarantees that the model may be reloaded for destiny predictions without retraining, enhancing efficiency for real-time or batch predictions. Specifically:

- The model is saved as knn_recommender_model. Joblib.
- The scaler is saved as scaler. Joblib.

Step 5: Model Prediction: The code includes a suggestions characteristic to generate predictions for user inputs. When a person offers enter statistics (e.G., preferred great metrics), the method entails:

- Scaling the enter records the use of the stored Min Max Scaler to align it with the education records' scale.
- Passing the scaled enter to the KNN model's neighbors' approach to become aware of the 10 nearest net offerings based totally at the similarity in pleasant metrics.

- Retrieving the endorsed net services' URLs using the indices back through the neighbors' method.

Step 6: Displaying Results: The system shows the pinnacle 10 maximum applicable net provider URLs to the person. This enables customers to view and select from a listing of suggestions that first-class healthy their enter criteria based at the KNN version's similarity calculations. This methodology permits the recommendation machine to provide real-time, personalised tips based totally on great metrics, helping customers select ultimate internet offerings tailor-made to their requirements.

K-Nearest Neighbors:

The K-Nearest Neighbors set of rules is a simple but effective approach for recommendation structures. It works through finding the "K" maximum comparable items to a given question item primarily based on a hard and fast of functions. First, the facts are pre-processed, typically through normalization, to make sure consistent scaling of features. Then, the KNN version calculates the gap (often Euclidean) among the query and each object within the dataset. The K nearest gadgets is recognized because the maximum similar. In a recommendation device, those nearest acquaintances represent the recommended items, offering users with options that closely align with their possibilities or requirements.

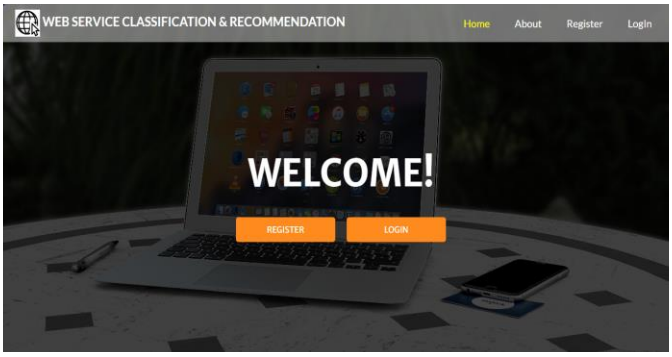
Decision Tree:

Decision Trees are extensively used machine mastering algorithms recognised for his or her simplicity and interpretability. They classify data by using recursively splitting it into subsets primarily based on the most widespread functions, making choices at every node. In this task, Decision Trees are applied to categorize internet offerings into four first-rate levels: Bronze, Silver, Gold, and Platinum. The class system starts by selecting the function that offers the highest facts advantage or the biggest reduction in Gini impurity. This selected feature is used to cut up the dataset into branches, every representing a probable outcome. The procedure maintains recursively, forming a tree shape wherein every leaf node represents a final category label.

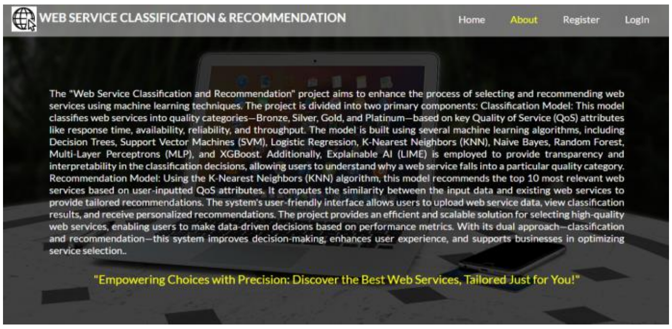
In Python, accuracy is a common metric used to evaluate the performance of classification models in machine learning. It represents the proportion of correctly predicted labels compared to the total number of predictions made. Accuracy gives a quick and straightforward understanding of how well a model is performing. It is especially useful when the dataset is balanced, meaning the classes have an equal or similar number of samples. However, in cases of imbalanced datasets, accuracy alone might be misleading, as it can be high even if the model fails to predict the minority class correctly.

RESULTS

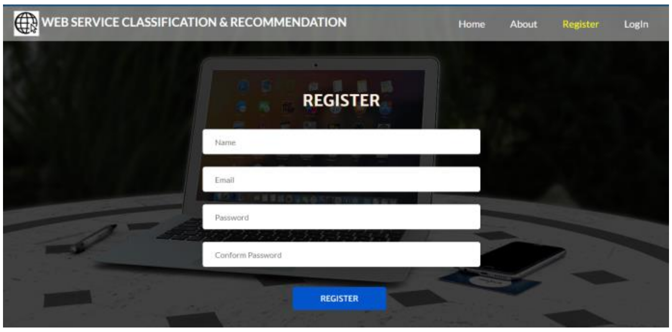
HOME PAGE: The Home page serves as the landing page of your application.



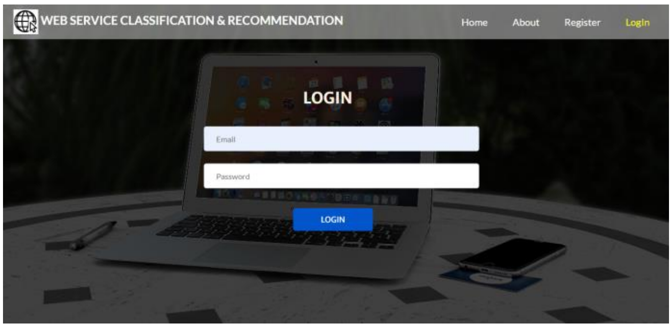
ABOUT PAGE: The About Page offers detailed information about the project.



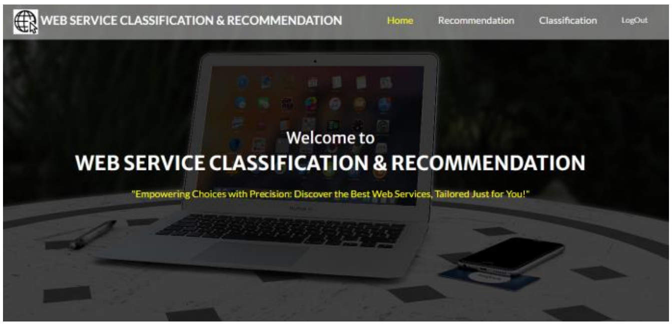
REGISTRATION: The Registration Page allows new users to create an account with the application. It typically includes fields for entering personal information such as name, email, password.



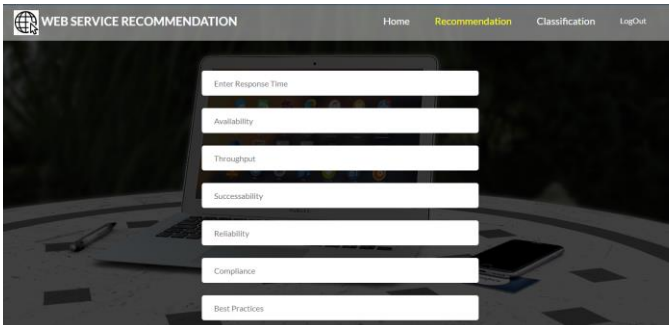
LOGIN PAGE: The Login Page enables users to access their existing accounts by entering their credentials. It usually includes fields for entering a username/email and password.



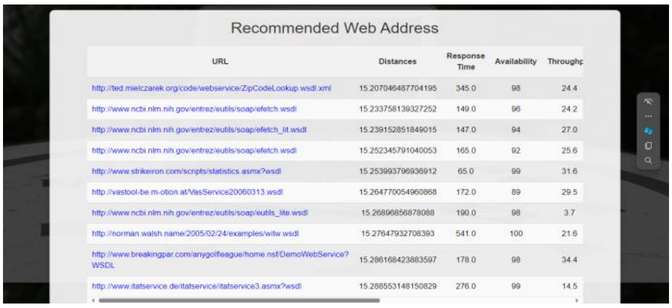
HOME PAGE: After user successfully login this page will be appear.



Prediction PAGE: The Prediction Page allows users to input data and receive predictions based on the trained machine learning models. This page typically includes a form or interface for uploading or entering data



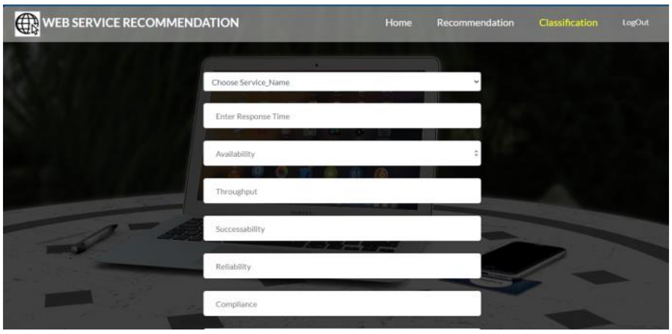
Result PAGE: In here result will be display which is predicted by our trained model.

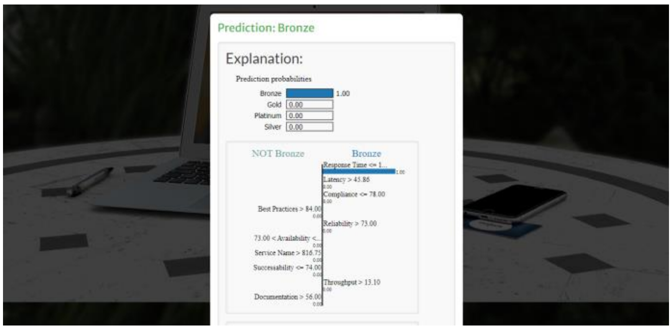

Recommended Web Address

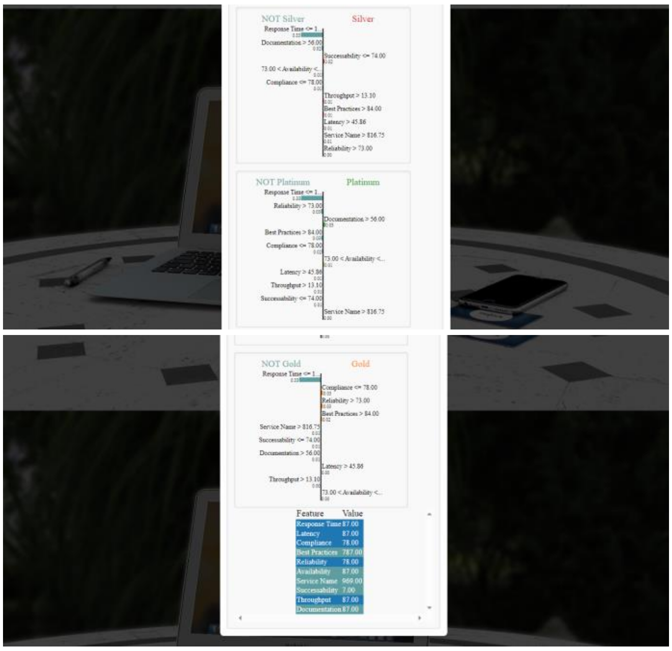
URL	Distances	Response Time	Availability	Throughp
http://ted.mielczarek.org/code/webservice/ZipCodeLookup.wsdl.xml	15.207046487704195	345.0	98	24.4
http://www.ncbi.nlm.nih.gov/entrez/eutils/soap/efetch.wsdl	15.233758139327252	149.0	96	24.2
http://www.ncbi.nlm.nih.gov/entrez/eutils/soap/efetch_lit.wsdl	15.239152851849015	147.0	94	27.0
http://www.ncbi.nlm.nih.gov/entrez/eutils/soap/efetch.wsdl	15.252345791040053	165.0	92	25.6
http://www.strikeiron.com/scripts/statistics.asmx?wsdl	15.253993796936912	65.0	99	31.6
http://vastool-be.m-otion.at/VasService20060313.wsdl	15.264770054960868	172.0	89	29.5
http://www.ncbi.nlm.nih.gov/entrez/eutils/soap/eutils_lite.wsdl	15.26896856878088	190.0	98	3.7
http://norman.walsh.name/2005/02/24/examples/witw.wsdl	15.27647932708393	541.0	100	21.6
http://www.breakingpar.com/anygolfleague/home.nsf/DemoWebService?WSDL	15.286168423883597	178.0	98	34.4
http://www.itatservice.de/itatservice/itatservice3.asmx?wsdl	15.288553148150829	276.0	99	14.5


Recommended Web Address

	Response Time	Availability	Throughput	Successability	Reliability	Compliance	Best Practices	Latency	Documentation
31	41.0	97	43.1	99	73	100	84	1.0	5
29	82.0	83	41.2	84	67	89	77	2.0	10
366	67.0	86	41.0	95	73	100	84	5.0	3
233	89.0	83	40.4	84	78	89	89	3.0	3
586	57.0	86	40.1	95	73	100	84	1.0	10
508	82.0	83	38.7	84	80	89	57	2.0	3
93	112.0	83	37.7	84	73	78	75	1.0	3
237	107.0	91	36.9	97	83	89	91	3.0	12
206	78.0	92	36.9	97	73	100	84	1.0	89
716	103.0	94	36.6	98	80	100	87	1.0	8







Test Cases:

Input	Output	Result
Input	Tested for different model given by user on the different model.	Success
Model	Tested for different input given by the user on different models are created using the different algorithms and data.	Success
Prediction	Prediction will be performed using the different models build from the algorithms.	Success

Test cases Model building:

S.NO	Test cases	I/O	Expected O/T	Actual O/T	P/F
1	Read the datasets.	Dataset's path.	Datasets need to read successfully.	Datasets fetched successfully.	It produced P. If this not F will come
2	Registration	Valid username, email, password.	Verify that the registration form accepts valid user inputs and successfully creates a new account.	User is successfully registered, and an account is created	It produced P. If this is not, it will undergo F.
3	Login	Valid username and password	Verify that users can log in with valid credentials	User is successfully logged in and redirected to the dashboard	It produced P. If this is not, it will undergo F.
4	Classification	Input Web service data	Output as category from silver, gold, platinum, bronze	Output as given any one of these categories	It produced P. If this is not, it will undergo F
5	Recommendation	Input Web service data	Output as top 10 relevant web address	Output as top 10 relevant web address	It produced P. If this is not, it will undergo F

CONCLUSION

In conclusion, this challenge addresses the growing complexity of selecting net offerings by using offering a statistics-pushed, scalable, and user-pleasant solution. It has big packages across diverse industries, together with e-trade, cloud computing, and on-line platforms, where green and reliable service selection is critical for making sure seamless operations and more desirable person experiences. The mission lays the inspiration for future improvements, together with integrating collaborative filtering, real-time service excellent updates, and multi-language support, to in addition extend its skills and person attain.

REFERENCES

[1]. Z. Jia, Y. Fan, J. Zhang, X. Wu, C. Wei and R. Yan, "A Multi-Source Information Graph-Based Web Service Recommendation Framework for a Web Service Ecosystem," in Journal of Web Engineering, vol. 21, no. 8, pp. 2287-2312, November 2022.

[2]. B. Jiang, J. Yang, Y. Qin, T. Wang, M. Wang and W. Pan, "A Service Recommendation Algorithm Based on Knowledge Graph and Collaborative Filtering," in IEEE Access, vol. 9, pp. 50880-50892, 2021,

[3]. Sivanandam, C., Seethapathy, B.K. & Doss, D. HBO-BiLSTM: hybrid bat optimizer-based bidirectional long short-term memory for secure web service recommendation. Wireless Netw (2024).

[4]. Pandey, A., Mannepalli, P.K., Gupta, M. et al. A Deep Learning-Based Hybrid CNN-LSTM Model for Location-Aware Web Service Recommendation. Neural Process Lett 56, 234 (2024).

[5]. Masoumeh Alinia, Seyed Mohammad Hossein Hasheminejad, DiSA-CF: A distance-integrated self-attention model for collaborative filtering in web service recommendation, 2024.

[6]. Zhang, Y., & Zhou, M. (2023). "A Novel Hybrid Recommender System for Web Services Using Deep Learning and Collaborative Filtering." Journal of Internet Services and Applications, vol. 14, no. 3, pp. 189-206.

[7]. Chen, X., Liu, Y., & Xu, T. (2022). "Dynamic Service Recommendation Using Graph Neural Networks for Complex Web Service Ecosystems." IEEE Transactions on Services Computing, vol. 15, no. 6, pp. 1175-1187.

[8]. Wang, L., & Huang, R. (2023). "A Self-Attention Based Deep Learning Model for Real-Time Web Service Recommendation." IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 5, pp. 1238-1250.

[9]. Singh, A., & Verma, P. (2022). "Enhancing Web Service Recommendations with Multi-Objective Optimization and Ensemble Learning." Expert Systems with Applications, vol. 196, article 116481.

[10]. Lee, J., & Kim, H. (2023). "Context-Aware and Personalization-Driven Web Service Recommendation Using Reinforcement Learning." Knowledge-Based Systems, vol. 259, article 110011.