Adaptive Online Platform for Enhanced Teaching and Learning

|  |  |  |
| --- | --- | --- |
| Dr. Gaurav Dubey  *Computer Science*  *KIET Group of Institution*  Ghaziabad, Uttar Pradesh, India gaurav.dubey@kiet.edu | Amod Katiyar  *Computer Science*  *KIET Group of Institution*  Ghaziabad, Uttar Pradesh, India amod.2024cs1058@kiet.edu | Himanshu Kumar  *Computer Science*  *KIET Group of Institution*  Ghaziabad, Uttar Pradesh, India himanshukumarrai786@gmail.com |

Aniket Bhardwaj

*Computer Science*

*KIET Group of Institution*

Ghaziabad, Uttar Pradesh, India aniket.2024cs1199@kiet.edu

*Abstract*—The ever-expanding landscape of educational opportunities presents a unique challenge for students today. While countless resources and pathways exist, navigating this complex ecosystem can be overwhelming. Students grapple with the critical questions of ”what to learn” and ”how to learn it effectively.” This paper proposes a novel solution to bridge this knowledge gap. It introduces a novel platform tailored to empower students navigating their educational journeys, particularly in the realm of programming language acquisition. Faced with the perennial question of ”what to learn,” students often struggle to prioritize amidst a plethora of programming languages. Our platform aims to alleviate this dilemma by leveraging data-driven insights to illuminate the most sought-after programming languages within the industry. This paper elucidates the methodology, architecture, and functionality of the platform, underscoring its potential to enhance educational experiences and equip students with valuable skills for professional success.

*Index Terms*—Educational opportunities, Programming language , Platform, Data-driven insights, Industry demands, Methodology, Architecture, Functionality, educational experiences, Professional success.

# I. INTRODUCTION

The digital revolution has irrevocably reshaped the educational landscape. Technological advancements, spearheaded by the internet and mobile technology, have democratized access to knowledge like never before. A vast treasure trove of learning materials now resides online, accessible on laptops, tablets, and smartphones. This newfound accessibility has broken down geographical barriers, creating a truly globalized learning environment where anyone with an internet connection can become a student. However, this abundance of educational resources presents a hidden challenge: the paradox of choice. Faced with a seemingly endless array of learning opportunities, students can become overwhelmed. The crucial questions of ”what to learn” and ”how to learn it effectively” loom large, often casting a shadow over a student’s academic journey. This paper addresses this critical challenge by proposing a novel solution. We envision a platform specifically designed to empower students in the digital learning age. Our platform tackles the ”what to learn” dilemma by leveraging data driven insights to highlight the most in-demand programming languages within the industry. This empowers students with knowledge of the skills that translate into real-world career opportunities. Moving beyond mere identification, the platform serves as a comprehensive roadmap for acquiring these skills.

# II. PROBLEM STATEMENT

The educational landscape is undergoing a seismic shift. Technological advancements, particularly the ubiquitous presence of the internet and mobile technology, have unlocked a treasure trove of learning materials and revolutionized how we access knowledge. Online courses, accessible on laptops, tablets, and smartphones, have transcended geographical boundaries, democratized education and making it a truly global phenomenon. For learners with an internet connection, the world has become their classroom.

However, amidst this abundance of resources lies a hidden challenge: the paradox of choice. The very existence of countless learning opportunities can be paralyzing. Students today face an overwhelming array of possibilities, making it difficult to identify a clear path through the educational maze. Choosing ”what to learn” and ”how to learn it effectively” are crucial questions that can cast a long shadow over a student’s academic journey.

# III. OBJECTIVE

This paper proposes a solution to navigate this complex educational landscape. We envision a platform designed to empower students by addressing the what-to-learn dilemma.

Our platform leverages data-driven insights to highlight the most in-demand programming languages within the industry, equipping students with knowledge about the skills that translate into real-world career opportunities. Going beyond mere identification, the platform will serve as a comprehensive roadmap for acquiring these skills.

This roadmap will not be a solitary path; it will be a collaborative journey. Students will have the opportunity to learn from each other’s experiences by sharing their hands-on knowledge through discussion forums and other online resources. This collaborative learning environment fosters a

sense of community and allows students to guide and assist one another on their learning journeys.

Furthermore, the platform will aggregate valuable learning materials from renowned platforms, creating a centralized hub of information. This centralized hub will offer a curated selection of resources, tailored to mastering a chosen language. Structured learning paths, encompassing foundational concepts to advanced techniques, will ensure a solid grasp of the chosen programming language.

By providing both direction and resources, our platform aims to empower students to navigate the ever-expanding world of education with confidence. This comprehensive approach seeks to unlock the potential within each student, propelling them towards a future driven by passion and knowledge.

IV. SCOPE

In the future, we will provide a much simpler user interface as well as various other resources to improve the quality of content offered.

We will also be providing certification for the specific languages learned as well as practice sets for students to test their skills in real time.

Offering a premium membership that includes dedicated videos on important topics as well as live sessions with pre-experienced people who are already working in their respective fields.

Students who have completed all the test series on the given topics will receive paid certified certification, and dedicated questions focusing on placements will also be provided.

# V. METHODOLOGY

HTML5 is a new standard for HTML which allows us to build rich and interactive web pages which bring HTML into the world of application development started in the year 2004. HTML moves from simply describing the basics of a text based web for presenting audio, video and animations to enabling offline functionality, geo location and local storage in client side databases. [3].

We are starting with the most popular courses or languages. Then, we will use NLP and API to extract the best links from the web for students to learn from. All of this will be done to assist the student in choosing his path and provide him with the resources he needs to succeed. A recommender system can be viewed as a search ranking system, where the input query is a set of user and contextual information, and the output is a ranked list of items. Given a query, the recommendation task is to find the relevant items in a database and then rank the items based on certain objectives, such as clicks or purchases [6].

In essence, a recommender system functions like a search engine for personalized suggestions. It considers a combination of user data and context as the ”query,” then generates a prioritized list of relevant items. This prioritization is determined by objectives like user clicks or purchases. It is basically of 3 types:

## A. Collaborative filtering

They power recommendation systems by leveraging user similarities. These systems analyze user ratings or preferences to identify users with similar tastes. This underlying assumption is that users who agreed in the past are likely to have similar future preferences. A key strength is the ability to recommend items without relying on detailed product descriptions, making it suitable for various domains. Essentially, collaborative filtering uses past user behavior to predict what a user might like based on their similarity to others [9].

## B. Content-based filtering

Content-based filtering methods analyse the textual content of research papers, such as titles, abstracts, or full texts, to identify similarities and generate recommendations. These methods rely on extracting features from the documents and calculating the similarity between them [7]. The system then compares item profiles to user preferences, recommending items with the most similar profiles. However, this method is limited to items with well-defined features and struggles with conceptually similar items lacking explicit feature overlap.

## C. Hybrid approaches

They combine content-based and collaborative filtering techniques. This allows them to leverage the strengths of both: content-based filtering for new items or users, and collaborative filtering for personalized recommendations based on user similarities. The contributions of each method can be weighted based on the situation, with content-based filtering potentially holding more weight for new items with limited user ratings. By combining different recommendation strategies, hybrid filtering can potentially achieve more accurate and personalized recommendations, but designing and implementing them can be more complex due to the need to effectively combine the different techniques.

# VI. SYSTEMS

Web applications tend to be multi-tiered by nature, with the most common structure being the three-tiered architecture. In its most common form, the three tiers are (i) Presentation layer, (ii) Application layer and (iii) Storage layers. [1]

Web applications can be implemented either through coding a website or through Content Management System (CMS) tool WordPress. WordPress is a free CMS tool which helps us to build a website very easily also it supports lack of plug-ins whereas Coding Website is hard to code but reliable.[4]

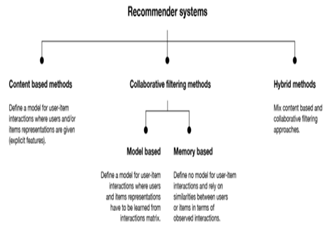
We will deploy a model of recommended systems that will analyse historical data and trends to determine the popularity and usage of languages such as Java, C, and Python and then recommend it to students. We will also show this in real-time graphs.

NLP (Natural Language Processing) techniques power recommendation systems for research papers. By analysing the text of titles, abstracts, or even full papers, these systems can identify relevant content and suggest similar papers to users. This analysis involves extracting key features, like keywords and concepts, from the documents. These features are then used to calculate the similarity between different papers, allowing the system to recommend related research based on the user’s interests. NLP (Natural Language Processing) techniques power recommendation systems for research papers. By analysing the text of titles, abstracts, or even full papers, these systems can identify relevant content and suggest similar papers to users. This analysis involves extracting key features, like keywords and concepts, from the documents. These features are then used to calculate the similarity between different papers, allowing the system to recommend related research based on the user’s interests.

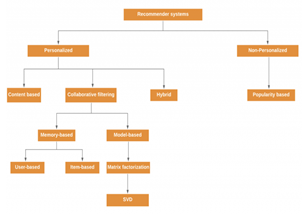
NLP (Natural Language Processing) techniques power recommendation systems for research papers. Content-based recommenders use a variety of machine-learning algorithms, including Naive Bayes, support vector machines, decision trees, and kNN. As bag-of-words and vector represen- tations can have hundreds or thousands of dimen- sions, techniques as Latent Dirichlet Allocation (LDA) are often adopted. The content may also require natural language processing (NLP) tech- niques to make use of semantic and syntactic characteristics.

Content-based recommenders use a variety of machinelearning algorithms, including Naive Bayes, support vector machines, decision trees, and kNN. As bag-of-words and vector represen- tations can have hundreds or thousands of dimensions, techniques as Latent Dirichlet Allocation (LDA) are often adopted. The content may also require natural language processing (NLP) tech- niques to make use of semantic and syntactic char- acteristics.

Hence, we will use NLP to extract words like Python data science best top to analyze and recommend courses. We will



## Fig. 1. Types of Recommender system Algorithms



## Fig. 2. Proposed Flow Chart for recommender systems

also use API for this.

In the last two decades, machine learning models have drawn attention and have established themselves as serious contenders to classical statistical models in the forecasting community. These models, also called black-box or data driven models, are examples of nonparametric nonlinear models which use only historical data to learn the stochastic dependency between the past and the future. For instance, Werbos found that Artificial Neural Networks (ANNs) outperform the classical statistical methods such as linear regression and Box-Jenkins approaches [10] .

Additionally, some machine learning models boast the ability to automatically extract relevant features from the data, reducing the need for manual feature engineering – a timeconsuming and expertise-dependent process [10] .

# VII. IMPLEMENTATION (WEBSITE)

At the presentation layer, the web pages are rendered to the browser. Traditionally, the webpages contain only HTML code. Now a days, the presentation layer of the web applications provide nearly the same user experience as desk top

applications.[1]

The basic advantage for the developers and browsers is that they would be able to do more without the need of mastering or licensing multiple proprietary technologies that can develop rich web pages, enhanced forms and web based

applications.[3]

Wordpress may be used in certain scenarios for development but it does have some drawbacks,Some of the common problems are listed below: (i)-White Screen Of Death(WSOD) problem, (ii)- WP Memory Limit

Normally no issues are occurred when we run coding website, Whereas WordPress website sometime occurs issues, to avoid it all plug-ins should be updated timely also the theme should be up to date. [4]

The Development of a website is dependent upon major technologies : (i) Frontend Technologies, (ii) Backend Technologies, (iii) Databases.

## A. Frontend Technologies

JavaScript could be a Scripting language. it’s principally abbreviated as JS. It is aforementioned that JavaScript is that the updated version of the ECMA script.JavaScript could be a light-weight, cross-platform, and taken scripting language[5]

## B. Backend Technologies

Back-end development focuses on the server-side aspects of a netsite—an internet site—a web site or web application. this kind of development cares with web site design, scripting, and communication with databases. Back-end code permits the communication between browsers and data from databases. Back-end development works in conjunction with front-end development to supply users with a useful and interactive expertise.[5]

## C. Databases

Database is that the assortment of inter-related knowledge that helps in economical retrieval, insertion and deletion of information from information and organizes the info within the style of tables, views, schemas, reports etc. SQL databases square measure structured, and NoSQL databases don’t seem to be structured.[5]

VIII. IMPLEMENTATION (RECOMMENDATION SYSTEM)

Recommendation systems recommend an item to which a user prefers by using automatic information filtering method. It deals with the detection and delivery of information that the user is likely to find interesting or useful. It assists users by filtering the data source and deliver relevant information to the users. There are two main approaches to build a recommendation system - collaborative filtering and content based with the development of the internet, especially the mobile Internet, information has undergone an increase. More than

80recent years. With the increase of information, the access of people to useful information is more difficult. Hence, the role of recommendation systems have become inevitable.[9]

## A. Collaborative filtering

*a) Data Acquisition through extracting GitHub Event Data::* The initial stage of the process involves data acquisition. Our code delves into a treasure trove of JSON files, each one meticulously recording GitHub events. These events encompass a wide range of interactions, including issues raised, pull requests submitted, and stars bestowed upon repositories. By meticulously parsing through these JSON files, we extract the valuable data they contain.

Collaborative recommendation systems aggregate ratings or recommendations of objects, recognize commonalities between the users on the basis of their ratings, and generate new recommendations based on inter-user comparisons. The greatest strength of collaborative techniques is they are not dependent of any machine-readable representation of the objects being recommended. Collaborative filtering is based on the assumption that people who agreed in the past will agree in the future and that they will like similar kind of objects as they liked in the past. Basically collaborative filtering is based on collecting and analyzing a large amount of information on users’ behaviors, activities or preferences and predicting what users will like based on their similarity to other users.[9]

*B. Data Consolidation by building a Unified Dataset:*

Following the successful extraction of data from individual JSON files, we embark on the task of data consolidation. This crucial step involves merging the information gleaned from each separate file into a singular, unified dataset. This consolidated dataset serves as the foundation for subsequent data processing and analysis. By combining the data, we create a comprehensive picture of user activity and code contributions across various repositories. This unified structure facilitates a more holistic exploration of trends and patterns within the data.

## C. Data Structuring by tailoring for Time Series Analysis

Once the data has been successfully consolidated, we proceed with data structuring. This step involves transforming the raw data into a format specifically tailored for time series analysis. To achieve this, we meticulously group the data based on three key dimensions:

1. *Year: :* By grouping data by year, we can identify trends and patterns that unfold over time. This allows us to examine how user activity and code contributions evolve across different temporal segments.
2. *Language Name: :* Grouping data by language name enables us to investigate trends specific to individual programming languages. This facilitates a deeper understanding of how developers engage with different languages and how language popularity might fluctuate over time.
3. *Counts: :* The ”counts” dimension refers to the number of events (issues, pull requests, stars) associated with each unique combination of year and language name. These counts provide the numerical foundation for our time series analysis, allowing us to quantify and visualize trends in user behavior.

Through this meticulous data structuring process, we transform the raw data into a well-organized format specifically designed to unlock the secrets hidden within the time series. This structured data paves the way for the application of powerful time series analysis techniques.

1. *Time Series Analysis:*

Auto-Regressive Integrated Moving Average (ARIMA) models are utilized for time series analysis. The basic form of the ARIMA model is ARIMA (p, d, q), “p” is autoregressive coefficient, “d” is the order of difference made when the time series becomes stationary, “q” is number of moving average terms. First perform stationarity test on the time series, if the sequence is not stationary, use methods such as difference and logarithm to make the sequence stationary. Then use the autocorrelation function (ACF) graph and partial autocorrelation function (PACF) graph recognize and rank the model [8]. The ARIMA models are applied to predict the future trends in language popularity.

1. *Model Parameter Tuning:*

For each language, the ARIMA model parameters, including order and seasonal order, are fine-tuned to optimize predictive accuracy. The code uses Mean Absolute Error (MAE) as the evaluation metric to determine the best model.

1. *Grid Search for Optimization:*

A set of hyperparameters and their values are feed to it first and then run an exhaustive search overall all possible combination of given values then training the model for each set of values is done. Then Grid Search algorithm will compare the score of each model it trains and keeps the best one. A common extension of Grid Search is to use cross-validation i.e., training the model on several different folds with different hyperparameter combinations to find more accurate results [9] .

Grid Search provides a reliable way to pinpoint the best hyperparameter combination. However, for scenarios with a high number of hyperparameters, evaluating every single combination can become time-consuming. To address this, another approach called Randomized Search emerges. Instead of an exhaustive evaluation, Randomized Search samples a predefined number of parameter configurations at random from the given ranges. These sampled configurations are then used to train individual models. Finally, the model demonstrating the best performance based on the chosen metric (e.g., Mean Absolute Error) is selected as the champion. This method prioritizes efficiency by focusing on a representative subset of possible configurations, making it a valuable tool for streamlining hyperparameter tuning, especially when dealing with larger datasets or complex models.

*G. Best Model Selection via MAPE:*

Following the meticulous fine-tuning of ARIMA models for each language under investigation, a critical step emerges: the selection of the optimal model for forecasting purposes. While the selection process might intuitively gravitate towards the model exhibiting the lowest absolute error, a more nuanced approach is necessary. This is where the utility of Mean Absolute Percentage Error (MAPE) becomes apparent.

In contrast to Mean Absolute Error (MAE), which quantifies the raw discrepancy between predicted and actual values, MAPE expresses the error as a relative measure – a percentage of the actual language usage. This shift in perspective proves particularly advantageous when dealing with languages exhibiting significant disparities in usage volumes. Consider the scenario of comparing predicted popularity between a niche language with a limited speaker base (e.g., a few thousand) and a widely spoken language boasting millions of users. Sole reliance on MAE could lead to a suboptimal selection, potentially favouring a model that performs well on the high-volume language but falters with the niche one.

MAPE offers an elegant solution by normalizing the error relative to the actual usage, thereby facilitating a more equitable and standardized comparison across languages. Consequently, the model with the lowest MAPE for a specific language signifies the one that generates the most accurate forecasts relative to its current usage patterns. This optimal model is subsequently entrusted with the critical task of predicting future trends in language popularity for that particular language.

IX. FORECASTING AND VISUALIZATION:

With the best models selected, we generate forecasts for language counts in 2023, 2024, and 2025. Compelling visualizations are then created to translate the numerical data into clear trends, revealing potential surges, declines, or stability in language popularity.

## A. Results

The code aims to improve the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) values by refining the model parameters. Enhanced visualizations are generated to represent the forecasted data effectively.

Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots are instrumental tools in the exploration and modelling of time series data. These plots act as visual representations of the correlation between a time series and its lagged versions, offering valuable insights into the underlying structure of the data.

One of the key aspects to analyze in ACF and PACF plots is the presence of significant peaks that fall outside the designated confidence interval. These peaks signify the existence of statistically relevant relationships between the current value of the time series and its past values at specific lags.

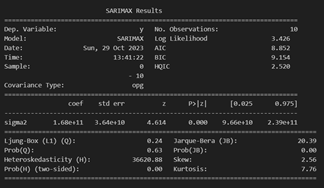
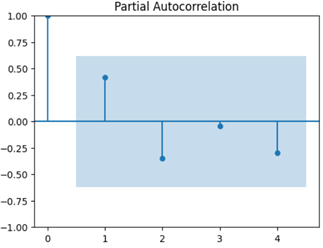


Fig. 3. SARIMAX Results



• Peaks in the ACF plot can point towards two potential phenomena:

1. *Seasonal Patterns :* If the peaks exhibit a recurring pattern at consistent intervals (e.g., every 12 lags for monthly data), it suggests the presence of seasonal components in the data. These seasonal components could represent recurring trends based on time of year, holidays, or other cyclical factors.
2. *Moving Average (MA) Components :* ACF peaks can also indicate the influence of past errors (residuals) on current values. This scenario suggests the presence of an MA component, where past forecast errors impact future values.

In contrast, significant peaks in the PACF plot specifically highlight the influence of past values (excluding the effects of intervening lags) on the current value. These peaks indicate the presence of Autoregressive (AR) components in the data. AR models posit that future values are predicted based on a linear combination of past values.Enhanced visualizations are generated to represent the forecasted data effectively.

By analyzing both ACF and PACF plots in tandem, we gain a comprehensive understanding of the autocorrelation structure within the time series. This combined knowledge proves invaluable when selecting appropriate models for time series forecasting. Models like ARIMA (Autoregressive Integrated Moving Average) and SARIMA (Seasonal ARIMA) leverage these identified structures (AR, MA, and seasonality) to build effective predictive models.

In essence, ACF and PACF plots serve as a bridge between the raw data and the selection of optimal forecasting models. By interpreting the peaks and patterns within these plots, we can decipher the inherent relationships within the data,

### Fig. 4. Results based on testing of ARIMA models

ultimately leading to more accurate and informed time series forecasting.

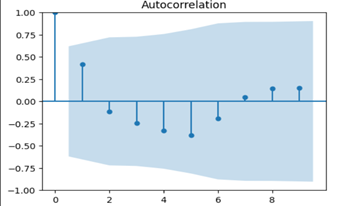


Fig. 5. Results based on testing of ARIMA models

### ACKNOWLEDGMENT

It gives us a great sense of pleasure to present the Research Paper of the B. Tech Mini Project undertaken during B.Tech. Third Year. We owe a special debt of gratitude to Dr. Gaurav

Dubey (Professor), Department of Computer Science, KIET Group of Institutions, Delhi-NCR, Ghaziabad, for his/her constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavours have seen the light of the day. We also take the opportunity to acknowledge the contribution of Dr. Ajay Kumar Shrivastava, Head of the Department of Computer Science, KIET Group of Institutions, Delhi- NCR, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution to the completion of the project.

### REFERENCES

1. A review and analysis of technologies for developing web applications, BY-SOLOMON ANTONY, 2012.
2. Web Application Development -a study on UML Web Application Extension, Andreas Oskarsson, Martin Kling, Tobias Norberg, 2002.
3. Research on HTML5 in Web Development, Ch Rajesh, K S V Krishna

Srikanth, 2014.

1. Review of WordPress website vs coding website By-Swapnil S More, 2018.
2. Website Development Technologies: A Review By - Pratiksha D Dutonde , Shivani S Mamidwar , Monali Sunil Korvate, 2022.
3. Wide and Deep Learning for Recommender Systems Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi

Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, Hemal Shah, 2016.

1. Research Paper Recommendation System Using Natural Language Processing Submitted in partial fulfilment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering by Siddharth Sanjay Gandhi 19BCE0005 Under the guidance of Dr. Akila Victor School of Computer Science and Engineering VIT Vellore Dr. Felix Dietrich, Department of Informatics Technical University of Munich, 2023.
2. RESEARCH ON COVID-19 EPIDEMIC BASED ON ARIMA MODEL

Li Zhihao et al 2021 J. Phys.: Conf. Ser. 2012 012063, 2012.

1. Research Paper On Recommendation System Dr. Alka Singhal Assit Professor,CSE department, IPEC Ghaziabad, India Shivangi Rastogi

Student, CSE department, IPEC Ghaziabad, India Nikhil Panchal

Student, CSE department, IPEC Ghaziabad, India Shivani Chauhan

Student, CSE department IPEC Ghaziabad, India Shradha Varshney Student, CSE department, IPEC Ghaziabad, India, 2021.

1. Forecasting with Machine Learning Dr. Osman Mohamed Abbas, 2018.