Design of Postgraduate Entrance Examination Recommendation Platform Based on Big Data and Ensemble Learning

Liuyang Zhao[1], Zanpu Wang[1], Qingfeng Zhou[2,*]
[1]*College of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, Henan, China*
[2]*iFLYTEK Co., Ltd., Hefei, Anhui, China*
**Corresponding Author*

Abstract: To address the issues of information asymmetry and intense competition in the postgraduate entrance examination, this paper designs and implements an intelligent analysis and recommendation platform based on big data and machine learning. The system employs the Scrapy framework to collect multi-source heterogeneous data from graduate admission websites and university portals, and constructs a four-layer data warehouse based on Hadoop and Hive for data cleaning and offline computing. Furthermore, the XGBoost algorithm is introduced to build a re-test score prediction model, and a Large Language Model is integrated to develop an intelligent preparation agent that provides personalized study plans. Experimental results show that the platform achieves multi-dimensional visualization via ECharts, and the prediction model achieves an R-squared of 0.7808, effectively assisting candidates in scientific school selection and efficient preparation.

Keywords: XGBoost; Python; Data Visuali-Zation; Recommendation System; Big Data Analytics

1. Introduction

In recent years, the competition for the National Postgraduate Entrance Examination has intensified, and the landscape of enrollment publicity has undergone significant changes [1]. However, traditional methods of information acquisition suffer from severe issues of "information silos" and decision-making lags, which fail to meet candidates' demands for precise and timely data.

Specifically, the phenomenon of "involution" in higher education has led to a surge in the number of applicants, while the growth rate of admission quotas remains relatively slow. This imbalance has made the selection of the target institution and major a critical strategic decision, often referred to as "the second college entrance examination." However, the current information ecosystem presents significant challenges. Most candidates rely on scattered information from forums (e.g., Zhihu, Reddit-like platforms) or static tables released by universities. These sources suffer from unstructured data formats, lack of historical continuity, and difficulty in cross-comparison. For instance, comparing the "real admission ratio" (excluding recommended students) across multiple universities requires manual data aggregation, which is time-consuming and error-prone. Furthermore, traditional search engines cannot provide personalized analysis based on a candidate's specific learning capability and risk preference. Therefore, there is an urgent need for an integrated platform that transforms raw data into actionable intelligence.

Based on this context, this paper designs and implements an intelligent analysis and recommendation system for postgraduate institutions. By integrating Scrapy for distributed data collection, Hive for layered data warehousing, and the XGBoost prediction algorithm, the system aims to construct a one-stop service platform that combines data monitoring, trend prediction, and intelligent recommendation. Through these technological means, the system assists candidates in making scientific, data-driven decisions for school selection.

2. Key Technologies

2.1 Distributed Data Acquisition Based on Scrapy

To address the characteristics of complex data structures and strict anti-crawling mechanisms

present in graduate admission websites and university portals, this system employs the Scrapy asynchronous crawler framework within the Python ecosystem to construct the data acquisition layer. In contrast to traditional single-threaded crawlers, Scrapy is built upon the Twisted asynchronous network engine. It is capable of handling high-concurrency requests via non-blocking I/O, thereby significantly enhancing crawling efficiency [2].

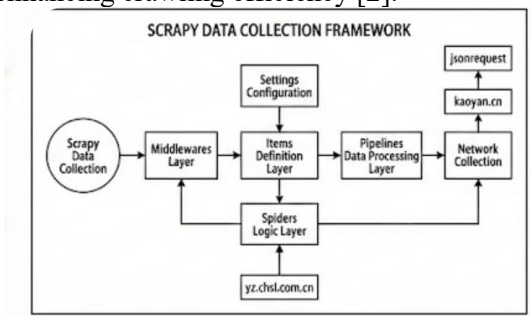

Figure 1. Design of Distributed Data Acquisition Process

As is shown in Figure 1, the data acquisition module consists of the following core components:

Scheduler: It maintains the URL request queue and utilizes Redis to implement distributed deduplication, ensuring that the same page is not fetched repeatedly. Instead of external caching, the system utilizes Scrapy's native request fingerprinting mechanism. It generates a unique 40-character SHA-1 hash for each URL and stores it in a local memory set. Before a request is enqueued, the scheduler verifies its fingerprint against this set to intercept duplicate links. Additionally, a randomized download delay (0.5s to 1.5s) is implemented within the scheduling logic. This mimics human browsing behavior, further reducing the risk of triggering the target server's anti-crawling threshold while maintaining data consistency.

Middleware: To circumvent IP bans and anti-crawling strategies, the system integrates a dynamic User-Agent pool and proxy IP tunneling technology. Furthermore, the middleware implements an intelligent retry mechanism with exponential backoff. If a request fails due to a timeout or a 503 error, it is automatically re-queued with an increasing delay, ensuring the completeness of the dataset even under unstable network conditions.

Item Pipeline: This component performs regular expression matching and cleaning on the captured unstructured HTML text, removing whitespace and special characters. It persistently stores the cleaned structured data (such as university codes, major names, and historical admission scores) into the local file system, providing raw corpus for subsequent big data processing.

2.2 Ensemble Learning Prediction Model Based on XGBoost

To address the issue of insufficient fitting capability of traditional linear regression models when dealing with non-linear postgraduate entrance examination data, this paper introduces the XGBoost (eXtreme Gradient Boosting) algorithm [3]. This is an efficient ensemble learning algorithm based on Gradient Boosting Decision Trees (GBDT) [4], possessing excellent generalization ability and robustness, as shown in Figure 2.

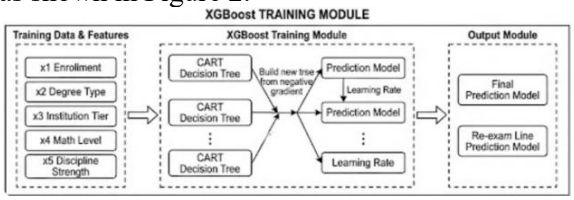

Figure 2. Training Process of Entrance Exam Score Prediction Model

XGBoost approximates the true value by iteratively adding weak learners, typically CART regression trees. Its core idea is to use the second-order Taylor expansion to optimize the objective function. Let $\hat{y}_i^{(t)}$ be the prediction value for sample i at the t-th iteration; the objective function can be expressed as:

$$fnew(x) = f(x) + \eta h(x) \quad (1)$$

The prediction update formula for the k-th weak learner is as follows:

$$\hat{y}i = \sum_{K=1}^{K} f_k(x_i) \quad (2)$$

Before model training, feature engineering is crucial for enhancing prediction accuracy. The system employs One-Hot Encoding to process categorical variables such as "University Location" and "Degree Type" (Academic vs. Professional), converting them into numerical vectors suitable for the regressor. For continuous variables like "Enrollment Quota" and "Historical Cut-off Lines," Min-Max Normalization is applied to map values to the [0, 1] interval, preventing features with larger value ranges from dominating the gradient updates.

The selection of XGBoost over other algorithms is based on a comparative analysis of characteristics suitable for this specific dataset. Unlike standard Random Forest, XGBoost utilizes a more regularized model formalization

to control over-fitting, which gives it better performance on the relatively small-scale but high-dimensional postgraduate admission datasets. Furthermore, its built-in capability to handle sparse data is particularly advantageous, as enrollment data for certain niche majors often contains missing values. The system uses the Squared Error as the loss function to minimize the discrepancy between the predicted score lines and the actual values.

In the model construction phase, the system first pre-processes the historical heterogeneous data from 2023 to 2026, eliminating missing values and outliers. Based on correlation analysis, 10 key feature dimensions were selected, including the examination year, enrollment quota, university location code, and historical national cut-off lines.

```
Model training completed.
Mean Absolute Error (MAE):
5.62 Coefficient of Determination
(R2 Score): 0.7808
Currently predicting the 2026 score line...
Prediction completed!
The result has been saved as:
2026_XGBoost_Prediction_Result.csv
```

Figure 3. Validation of Training Results

The dataset is divided into a training set and a testing set at a ratio of 8:2. By constructing a Pipeline containing data standardization and an XGBoost regressor, Grid Search is utilized to optimize hyperparameters such as the number of trees (n_estimators), maximum depth (max_depth), and learning rate (learning_rate). As shown by the final experimental results in Figure 3, the coefficient of determination (R^2) on the test set reached 0.7808, demonstrating that the model can effectively capture the non-linear impact of enrollment plan fluctuations on re-test scores.

3. System Design

3.1 Overall System Architecture Design

Following the software engineering principle of "high cohesion and low coupling," this system adopts a classic layered architecture design, achieving an end-to-end connection from data acquisition and storage computation to application display. As shown in Figure 4, the overall system architecture is logically divided into four layers from bottom to top.

(1) Data Source Layer: Python crawler tech-nology is utilized to perform full-volume data scraping from the Postgraduate Admission Information Network and official university websites to form the original dataset.

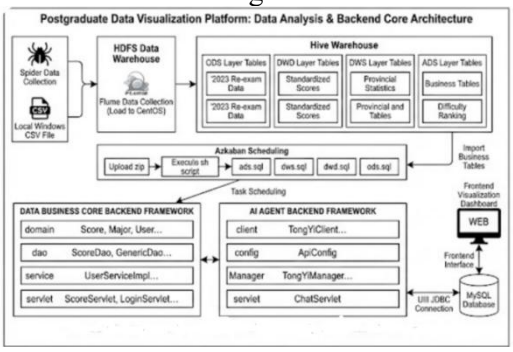

Figure 4. Overall Functional Architecture Design of the System

(2) Big Data Processing Layer: This is the core computing engine of the system. The Flume component is used to achieve secure data transmission from local logs to the HDFS distributed file system; an offline data warehouse is built based on Hive to complete data cleaning and modeling; the Azkaban workflow scheduler is used to orchestrate scheduled tasks, ensuring the automated operation of the data link [5].

(3) Business Logic Layer: The backend adopts the Java Web technology stack. It responds to frontend requests via Servlets, integrates the Service layer to handle complex business logic (such as permission management and AI agent invocation), and utilizes the DAO layer to implement persistent interaction with the MySQL database.

(4) Presentation Layer: The frontend employs the ECharts visualization library and asynchronous Ajax technology to construct a postgraduate data dashboard and an interactive Web interface, providing users with intuitive data insights.

3.2 Data Warehouse and ETL Process Design

The data warehouse follows the Dimensional Modeling methodology. We adopt a Star Schema for the DWS layer, where the "Admission Fact Table" serves as the center, linked to multiple dimension tables including "Time Dimension" (Year), "Geography Dimension" (Province/City), and "Institution Dimension" (985/211/Double First-Class status). This structure optimizes query performance for complex aggregation operations, such as calculating the average acceptance rate of all "211" universities in "Jiangsu Province" over the last three years. The ETL process not only cleans the data but also handles Slowly Changing Dimensions (SCD), ensuring that changes in

university attributes (e.g., a university being upgraded to a higher tier) are accurately reflected in the historical analysis.

To support the multi-dimensional analysis requirements of the upper layers, this system designs a standard four-layer data warehouse model based on Hive, realizing the extraction of value from data by transforming it from a disordered to an ordered state.

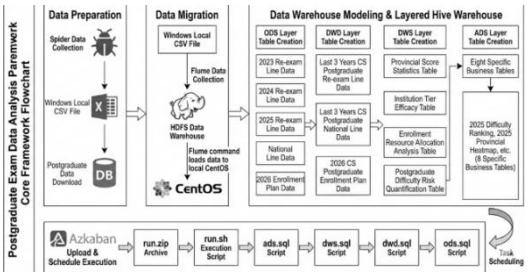

Figure 5. Data Warehouse Layering and ETL Process Design

As shown in Figure 5, the functional definitions of each layer are as follows:

ODS Layer (Operational Data Store): The source-aligned layer. Raw CSV/JSON files collected via Flume are mapped directly to this layer. The data preserves its original appearance without any modification, serving as a basic backup.

DWD Layer (Data Warehouse Detail): The detail layer. Large-scale data cleaning (ETL) is performed in this layer. HQL scripts are utilized to filter out non-computer science major data [6], standardize university hierarchy tags (such as "985" and "211"), and unify field names, thereby resolving the issue of data heterogeneity.

DWS Layer (Data Warehouse Service): The service layer. Light aggregation is performed based on analysis themes (such as "Province Popularity" and "University Difficulty") to generate wide tables grouped by province and year, effectively reducing the data scanning volume for subsequent queries.

ADS Layer (Application Data Service): The application layer. This layer is directly oriented towards frontend visualization needs, producing 8 result tables including the "Difficulty Ranking," "Competition Heatmap," and "Enrollment Expansion Opportunity Table." Azkaban manages the dependencies between layers by writing .job files, realizing the automated calculation flow from "ODS -> ADS."

Through this hierarchical design, the system achieves a significant improvement in data governance. By decoupling the raw data (ODS) from the application logic (ADS), the Data Lineage becomes clear and traceable. Furthermore, although the data is stored in HDFS, the layered aggregation strategy in the DWS layer drastically reduces the computational load for frequent queries. This ensures that even when the underlying data volume grows significantly in the future, the response time for the frontend visualization dashboard remains stable, fulfilling the requirements for low-latency interactive analysis.

4. System Implementation and Analysis

4.1 Implementation of Data Visualization Dashboard

To address the "information silo" problem caused by scattered data sources and large spatiotemporal spans in postgraduate examination data, this system constructs a dynamic visualization monitoring dashboard based on Vue + ECharts [7], as shown in Figure 6. This module is not merely a simple display of data, but the result of data value mining based on the ETL process

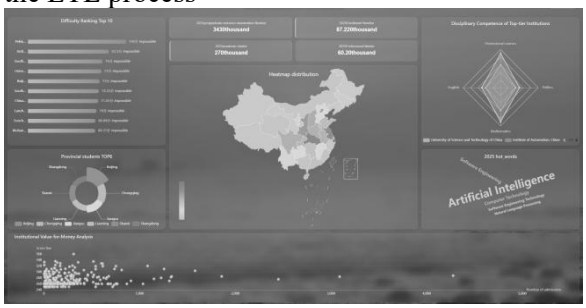

Figure 6. Monitoring and Analysis Dashboard of Data

In terms of visualization logic design, this section adopts a "Macro-Micro" progressive visual encoding strategy:

First, at the macro level, a Geographic Information System (GIS) heatmap is used to map the competition intensity of various provinces across the country. The color saturation is positively correlated with the difficulty of application, intuitively revealing the geographical distribution differences between "arid areas" (low competition) and "water areas" (high competition).

Second, at the micro level, Nightingale rose charts and multi-axis radar charts are introduced to address the quantification of the imbalance in student origin distribution and the multi-dimensional characterization of the discipline strength of a single institution,

respectively. For example, through the coverage area differences in the radar chart, candidates can quickly identify the special requirements of target institutions in single subjects such as "Mathematics" and "English," thereby formulating differentiated review strategies. This module effectively reduces the cognitive load for candidates to obtain key information.

4.2 Implementation of Recommendation Based on Feature Matching

Traditional retrieval systems often fall into the inefficient cycle of "looking for a needle in a haystack." This system achieves a paradigm shift from "people finding information" to "information finding people" by introducing machine learning algorithms. As shown in Figure 7, the core recommendation module no longer relies on simple SQL fuzzy queries but constructs a matching mechanism based on the Vector Space Model (VSM).

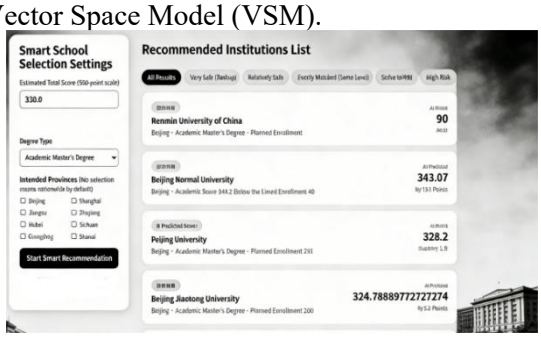

Figure 7. Hierarchical Intelligent Recommendation Results Interface

As indicated in Figure 7, the system first performs vector encoding on the user's multi-dimensional features (including target score range, intended city, and academic/professional degree preference) to construct a User Demand Vector. Simultaneously, features such as historical admission average scores and acceptance ratios in the institution database are mapped to Item Vectors. The matching score is quantified by calculating the Cosine Similarity between them. Mathematically, the cosine similarity measures the cosine of the angle between the user vector and the institution vector in a multi-dimensional space. A value closer to 1 indicates that the institution's characteristics (e.g., admission difficulty, geographical location) are highly aligned with the user's capabilities and preferences. This geometric approach allows the system to capture implicit relationships between candidate needs and university offerings that simple keyword matching often misses. To solve the problem of singularity in recommendation results, this system innovatively designs a "Tiered Recommendation Strategy," dividing high-similarity results into three echelons: "Conservative" (Similarity > 0.9), "Balanced" (0.8 < Similarity < 0.9), and "Sprint" (0.7 < Similarity < 0.8). This design ensures recommendation accuracy while taking into account the candidate's risk preference, effectively solving the "filter bubble" effect common in traditional recommendation algorithms.

4.3 Implementation of AI Agent for Exam Preparation

To break through the limitation that traditional systems only provide static data, this system integrates Generative AI technology and develops an intelligent preparation Agent capable of handling long-text contexts. This paper designs a structured Chain-of-Thought (CoT) prompt template. The system-level prompt consists of three distinct modules: Role Definition, Context Injection, and Output Constraints.

(1) Role Definition: "You are an expert consultant for the Chinese Postgraduate Entrance Examination with 10 years of experience."

(2) Context Injection: The system dynamically retrieves the specific user's target major data (e.g., "Software Engineering at Nanjing University") from the database and injects it into the prompt context window, ensuring the LLM's answers are grounded in real data rather than general knowledge.

(3) Output Constraints: "Please provide the advice in a structured format: Step 1 (Resource Analysis), Step 2 (Monthly Plan), and Step 3 (Risk Warning)." This Retrieval-Augmented Generation (RAG) approach significantly improves the reliability of the generated advice.

As shown in Figure 8.

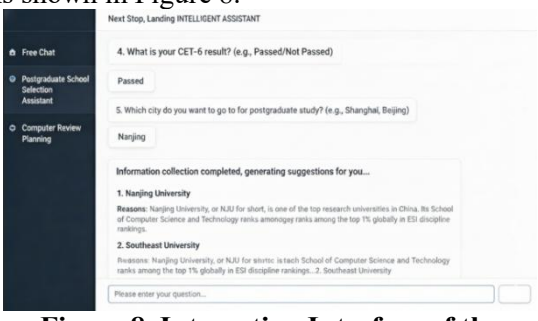

Figure 8. Interaction Interface of the Postgraduate Preparation AI Agent

The technical difficulty of this module lies in

how to adapt the general large model to the knowledge boundaries of the specific vertical domain of the postgraduate entrance examination. The system adopts Prompt Engineering technology, setting specific "expert role" constraints for the model by pre-setting system-level Prompts (containing the examination timeline, review stage theory, and methodology for each subject) [8].

As shown in Figure 8, when a user inputs their current review progress and pain points, the Agent can dynamically generate a structured review schedule based on the remaining preparation time. Compared with traditional static experience posts, this Agent possesses "memory" and "reasoning" capabilities, allowing it to continuously iterate suggestion strategies based on user feedback, thereby realizing the personalization and dynamization of preparation guidance. Preliminary user feedback indicates that the intelligent agent has significant advantages in resolving unstructured preparation queries.

5. Conclusion

Addressing the issue of information asymmetry in postgraduate entrance examinations, this paper comprehensively utilizes Python crawlers, the Hadoop ecosystem, and machine learning technologies to construct an intelligent analysis and recommendation platform for postgraduate institutions. The system realizes a full-link closed loop from data acquisition to value mining: it intuitively displays macro trends in enrollment through a visualization dashboard, achieves effective prediction of re-test scores using the XGBoost model (R^2 =0.7808), and provides personalized preparation planning by integrating an AI agent. Although the current system data mainly covers Computer Science majors and the consideration of sudden policy changes is still insufficient, the feasibility of the proposed scheme has been preliminarily verified. Future work will focus on introducing Natural Language Processing (NLP) technology to analyze public sentiment regarding institutions and attempting to adopt Transformer-based models to further improve the robustness and accuracy of time-series prediction.

References

[1] Dogan M E, Goru Dogan T, Bozkurt A. The use of artificial intelligence (AI) in online learning and distance education processes: A systematic review of empirical studies. Applied sciences, 2023, 13(5): 3056.

[2] Abodayeh A, Hejazi R, Najjar W, et al. Web scraping for data analytics: A beautifulsoup implementation//2023 sixth international conference of women in data science at prince Sultan University (WiDS PSU). IEEE, 2023: 65-69.

[3] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016: 785-794.

[4] Yağcı M. Educational data mining: prediction of students' academic performance using machine learning algorithms. Smart Learning Environments, 2022, 9(1): 11.

[5] Thusoo A, Sarma J S, Jain N, et al. Hive: a warehousing solution over a map-reduce framework. Proceedings of the VLDB Endowment, 2009, 2(2): 1626-1629.

[6] Thusoo A, Sarma J S, Jain N, et al. Hive-a petabyte scale data warehouse using hadoop//2010 IEEE 26th international conference on data engineering (ICDE 2010). IEEE, 2010: 996-1005.

[7] Kaur P. Sentiment analysis using web scraping for live news data with machine learning algorithms. Materials today: proceedings, 2022, 65: 3333-3341.

[8] Nalla L N, Reddy V M. AI-driven big data analytics for enhanced customer journeys: A new paradigm in e-commerce. International Journal of Advanced Engineering Technologies and Innovations, 2024, 2(1): 719-740.

