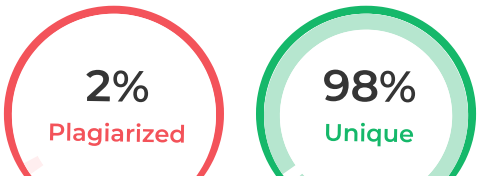
Plagiarism Detector .net

May 05, 2024

Plagiarism Scan Report



Characters:7457	Words:978
Sentences:48	Speak Time: 8 Min

Excluded URL: None

Content Checked for Plagiarism

CHAPTER 1. VISION 1.1 INTRODUCTION In the ever-evolving landscape of data management, organizations are continually faced with the challenge of navigating through vast volumes of information while ensuring its integrity, security, and efficiency. The "Data Redundancy Removal System" project emerges as a strategic initiative aimed at addressing one of the fundamental issues pervasive in contemporary data architectures – redundancy. In the digital realm where data proliferation is exponential, redundant data poses not only operational inefficiencies but also risks to security and scalability. This project endeavors to provide a comprehensive solution, leveraging advanced technologies in Big Data Analytics (BDA), fortified by robust Cyber Security measures, and implemented through a Cloud-Based Application (CBA) framework. 1.2 SCOPE The scope of the Data Redundancy Removal System, integrated with Python and leveraging relevant Python libraries, is to systematically identify and eliminate redundant data within a dataset or database. This system aims to provide a flexible and customizable solution using Python's programming capabilities and data processing libraries for efficient redundancy detection and removal. 1.3 PURPOSE / GOAL In addition to the previously mentioned goals and objectives, the integration of Python and associated libraries introduces specific focuses: Customization through Python: Leverage Python's flexibility to allow for customization of redundancy detection algorithms, catering to the unique characteristics of different datasets. Scalability: Utilize Python's scalability for processing large datasets efficiently, ensuring that the redundancy removal system can handle varying data sizes. Community Support: Benefit from the extensive Python community and ecosystem for continuous improvement, support, and the availability of diverse libraries. 1.4 OVERVIEW In response to the escalating complexities of modern data management, the Data Redundancy Removal System emerges as a transformative solution, strategically designed to optimize data efficiency, fortify security, and unlock the full potential of organizational data assets. This overview provides a comprehensive insight into the key components, features, and anticipated impacts of this innovative project. CHAPTER 2. POSITIONING 2.1 BUSINESS OPPORTUNITY In the ever-expanding digital landscape, organizations are faced with a dual challenge – the proliferation of data and the persistence of redundancy within it. The Data Redundancy Removal System not only addresses these challenges but strategically positions itself as a pivotal business opportunity, unlocking

unprecedented value in data management. Cost Savings and Resource Optimization: Redundant data incurs unnecessary costs related to storage, processing, and management. The Data Redundancy Removal System presents a tangible business benefit by reducing these costs through the elimination of superfluous data. The efficient utilization of resources allows organizations to redirect budget allocations toward strategic initiatives. Enhanced Data Security and Compliance: In an era where data breaches and regulatory compliance are critical concerns, the system provides a business opportunity to fortify data security. By integrating robust cybersecurity measures, organizations not only mitigate security risks but also enhance their compliance posture. This becomes a strategic advantage in building trust and safeguarding sensitive information. Scalability for Future Growth: The Cloud-Based Application (CBA) framework incorporated into the system offers a distinct business opportunity for scalability. Organizations can seamlessly adapt to evolving data needs, whether it's an expansion of data volumes or the integration of new data sources. This scalability positions the system as a catalyst for future growth and innovation. Strategic Decision-Making: As data becomes a critical asset for informed decision-making, the Data Redundancy Removal System creates a business opportunity by providing cleaner, more reliable data. Decision-makers can rely on accurate insights, enabling strategic and data-driven decision-making. This positions the system as an enabler of organizational success in a data-centric business environment. 2.2 PROBLEM STATEMENT Customization Challenge: Existing redundancy removal systems lack the necessary flexibility for customization. Many algorithms struggle to adapt to the unique characteristics of diverse datasets. Scalability Bottleneck: Current systems face challenges in efficiently processing large datasets. The scalability of redundancy removal algorithms needs improvement to handle varying data sizes effectively. Integration Gap with Python: Lack of seamless integration with Python hampers the utilization of its extensive community support and ecosystem. Limited availability of diverse libraries for continuous improvement in redundancy removal systems. Machine Learning Integration Deficiency: Current redundancy removal solutions often lack integration with machine learning algorithms. The absence of intelligent redundancy identification and prediction through machine learning is a notable gap in existing systems. In summary, the problem statement revolves around the development of a redundancy removal system that addresses customization challenges, scalability bottlenecks, integration gaps with Python, and deficiencies in integrating machine learning for intelligent redundancy identification and prediction. The goal is to create a comprehensive solution that overcomes these limitations, offering a more effective and adaptable approach to redundancy removal. CHAPTER 3. USER DESCRIPTIONS 3.1 MARKET DEMOGRAPHICS The Data Redundancy Removal System targets a diverse market encompassing industries such as IT, finance, healthcare, manufacturing, and telecommunications. It caters to large enterprises with extensive datasets, mid-sized companies facing scaling challenges, and startups seeking cost-effective data solutions. The global market includes

multinational corporations with complex data management needs, while regional markets focus on localized data redundancy issues. The system addresses regulatory requirements in highly regulated industries, adapting to evolving standards in others. It resonates with technology-driven organizations and those undergoing digital transformations, offering modern data solutions. Its applicability extends to organizations dealing with highly sensitive data, providing robust security measures, and to those with moderately sensitive data prioritizing privacy. In a landscape shaped by varying organization sizes, geographical presence, regulatory environments, technological sophistication, and data sensitivity, the Data Redundancy Removal System provides a tailored approach to streamline data, enhance efficiency, and meet the unique challenges of each market segment. 3.2 USER ENVIRONMENT 3.2.1 Clear Layout Incorporate a clear layout with labeling on your website to make it easier for manager or users to find information. Your website should have a natural flow of information. 3.2.2 Be Efficient Keeping element to be attractive effective and satisfactory to meet the standard of UI. 3.2.3 Simple and Understandable Application should be simple user interface and easy to use for user.

Sources

2% Plagiarized

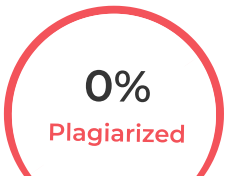
May 19, 2021 · Incorporate a clear layout with labeling on your healthcare website to make it easier for patients to find information. Your website should have a natural flow of information. Simple and intuitive. Don't reinvent the wheel. Keep it easy and intuitive to help users to find what they're looking for.

https://pyxl.com/blog/best-healthcare-website-design/

Plagiarism Detector .net

May 05, 2024

Plagiarism Scan Report



Characters:5711	Words:815
Sentences:38	Speak Time: 7 Min

Excluded URL | None

Content Checked for Plagiarism

3.3 USERS NEED ● User-Friendly Interface: Need an intuitive and easy-to-use interface for uploading data files, selecting fields, and viewing redundancy removal results. ● Customization Options: Require the ability to customize the display of redundancy removal results based on specific fields or criteria. ● Efficient Redundancy Analysis: Need a system that efficiently analyzes data to identify and remove redundancy, providing accurate and reliable results. ● Security Assurance: Expect robust security measures to protect sensitive data during the redundancy removal process. ● Collaboration Features: Benefit from features that support collaboration, such as sharing workspaces and commenting on data analysis. CHAPTER 4. PRODUCT FEATURES 4.1 FEATURE 1. User Interface (UI/UX): ○ File Input Module: ■ Allows users to upload files containing data with potential redundancy. ○ Supports various file formats (e.g., CSV, Excel). ○ Field Selection Module: ■ Enables users to choose fields for redundancy analysis. ■ Provides a user-friendly interface for field selection. ○ Result Display Module: ■ Displays the final result after redundancy removal. ■ Allows customization of the display based on user-selected fields. 2. Security Integration: ○ Data Security Module: ■ Implements Python libraries for enhanced data security. ■ Ensures encryption and secure handling of sensitive information. 3. Redundancy Removal: ○ Analysis Module: ■ Conducts analysis to identify redundant data. ■ Utilizes algorithms to detect and mark duplicate entries. ○ Redundancy Removal Module: ■ Identifying Functional Dependencies: We iterated over each pair of columns in the dataset to check for functional dependencies. Functional dependencies were identified by assessing whether one column (X) uniquely determines another column (Y). If the combination of values in X uniquely determines the values in Y, a functional dependency is established. Each identified functional dependency was stored in a dictionary for further analysis. ■ Data Normalization: To achieve normalization, we aimed to eliminate partial dependencies for 2NF or transitive dependencies for 3NF. Tables were created based on the identified functional dependencies. Each table comprised the primary key and attributes that were functionally dependent on it. We ensured that each table had a unique primary key by checking if the primary key uniquely identified records in the table. Finally, normalized tables were written into separate CSV files for further analysis. Other Functions: 1. Creates a New Table or CSV File with Redundant Entries Removed: Upon processing a dataset, the Duolicay Removal System is capable of generating a new table or

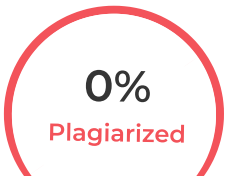
CSV file where redundant entries have been effectively eliminated. This feature aids in streamlining data sets, making them more manageable and conducive to analysis. 2. Ensures Data Integrity and Consistency: One of the primary objectives of the Duolicay Removal System is to uphold data integrity and consistency. By removing redundant entries, it helps maintain the accuracy and reliability of the dataset, ensuring that it remains trustworthy for analytical and decision-making purposes. 3. Exact Match Data Redundancy Removal: The system employs an exact match data redundancy removal technique to identify and eliminate duplicate records that are identical in all fields. This precise method ensures that only true duplicates are removed, leaving behind the most relevant and distinct data entries. 4. Fuzzy Method to Remove Data: In addition to exact match removal, the Duolicay Removal System utilizes a fuzzy method to identify and eliminate redundant entries that may not be exact matches but exhibit similarities. This fuzzy matching algorithm employs techniques such as string similarity or phonetic matching to identify potential duplicates based on similarity thresholds. 5. KNN Clustering Method: The system implements the K-Nearest Neighbors (KNN) clustering method to group similar data points together. By clustering data based on their proximity in a multidimensional space, the system can identify redundant entries and remove them effectively. This method is particularly useful in scenarios where exact matching or fuzzy methods may not be suitable. 6. CWOA Hospital Data Redundancy Removal Method: The Duolicay Removal System incorporates a specialized redundancy removal method tailored for healthcare datasets, such as those from hospitals. The CWOA (Chaotic Whale Optimization) method is designed to efficiently identify and eliminate redundant patient records or medical data entries while considering the unique characteristics and complexities of healthcare data. Column-Based Data Redundancy Removal: Apart from removing redundant entries across the entire dataset, the system offers column-based redundancy removal capabilities. This feature allows users to specify particular columns or fields where redundancy should be addressed, enabling more targeted and customizable data cleansing processes. By integrating these diverse functionalities, the Duolicay Removal System provides a comprehensive solution for effectively identifying, managing, and eliminating redundant data entries across various types of datasets, ensuring enhanced data quality and facilitating more accurate and insightful analysis. 4.2 OTHER FEATURE ■ User Authentication: ● Implements secure user authentication for system access. ■ Audit Trail: ● Logs user activities and changes made to the data. ● Enhances accountability and traceability. ■ Notification Module: ● Sends notifications/alerts for successful redundancy removal. ● Alerts users in case of any potential security issues

Sources

Plagiarism Detector .net

May 05, 2024

Plagiarism Scan Report



Characters:6194	Words:827
Sentences:32	Speak Time: 7 Min

Excluded URL | None

Content Checked for Plagiarism

CHAPTER 5. CONSTRAINTS AND PRODUCT REQUIREMENT CONSTRAINTS: The constraints for the project from the perspective of Cyber Security, Big Data Analytics, and Cloud-Based Applications: Cyber Security Constraints: • Data Privacy: Ensuring that sensitive information within the .csv files is protected from unauthorized access or breaches during upload, processing, and storage. Secure Communication: Implementing secure channels for communication between the client uploading the file, the server processing it, and the storage system. • Authentication and Authorization: Verifying the identity of users accessing the system and restricting access to authorized personnel only. • Vulnerability Management: Regularly updating and patching software components to address any potential security vulnerabilities that could be exploited by attackers. • Secure Configuration: Ensuring that the Python code, MongoDB database, and AWS services are configured securely to prevent common security misconfigurations. Big Data Analytics Constraints: • Scalability: Ensuring that the system can efficiently handle large volumes of data without compromising performance during redundancy removal and processing. • Data Quality: Maintaining the accuracy, consistency, and completeness of the data throughout the redundancy removal process to ensure reliable results. • Algorithm Efficiency: Optimizing the MDNN, exact match, fuzzy match, CWOA match, and selected column removal algorithms to handle big data efficiently and effectively. • Resource Utilization: Efficiently managing computing resources such as CPU, memory, and storage to minimize costs and maximize performance during data processing. Cloud-Based Applications Constraints: • Availability: Ensuring high availability of the application on AWS to minimize downtime and ensure continuous access for users. • Data Transfer Costs: Minimizing data transfer costs associated with uploading .csv files to the cloud and transferring processed data back to users. • Cloud Service Dependencies: Managing dependencies on AWS services and ensuring compatibility and interoperability with other cloud platforms or services. • Compliance: Adhering to regulatory requirements and compliance standards for data storage, processing, and transmission in the cloud environment. • Performance Optimization: Optimizing the application's performance on AWS by leveraging cloud-native services, auto-scaling, and load balancing to meet user demand efficiently. These constraints should be considered to ensure the security, efficiency, and reliability of the data redundancy removal system across different domains. 5.1 SYSTEM

REQUIREMENT Software Windows 7+, Brevo mailer, NGINX Database MongoDB 6.0+ FrontEnd HTML, CSS, JS, Jquery, Bootstrap BackEnd/Framework Django, Python 5.2 PERFORMANCE REQUIREMENT 4GB system RAM is enough to run the software, but more memory will increase the speed and performance of the software. - Chrome/Mozilla v90+ is required for smooth web performance. 6.3 MongoDB Collection Structure 6.3.1 USERS ○ _id: user id ObjectID() ○ firstName: String ○ lastName : String ○ email: String ○ phone_number: String ○ password: SHA256() hash value ○ occupation: String ○ ProfilePhoto: Blob() ○ request: [{ fileUploadedAt: Date(), file_id: , changes:[undo_perform, remove column, data_type_change], }] ○ userCreatedAt: Date() 6.3.2 GridFS database ● fileMetadata {file_id,finalename, datatype, storageType} ● File Chunks 6.3.3 log_storage ● log_id ● log_title ● log_content ● createdAt ● userid ● module_title Chapter 7 Deployment on AWS Cloud 7.1 Instance Details Instance Type: EC2 t2.micro Instance Name: drrs Instance Platform: Ubuntu Platform Details: Linux/UNIX – SERVER: NGINX 7.2 Load Balancing with Nginx and Gunicorn in a Django Application In our Django application, we employed a combination of Nginx and Gunicorn for efficient load balancing. This setup optimizes the distribution of incoming web traffic across multiple instances of our Django application, ensuring high availability, improved performance, and fault tolerance. Nginx: Nginx acts as a reverse proxy server, handling client requests and distributing them among multiple Guni corn workers. It efficiently manages incoming connections, optimizes resource utilization, and provides additional functionalities such as caching, SSL termination, and request routing. By leveraging Nginx's event-driven architecture and asynchronous processing capabilities, we achieved low-latency response times and high throughput, even under heavy loads. Furthermore, Nginx's robust configuration options allowed us to fine-tune our load balancing strategy according to our application's specific requirements. Gunicorn: Gunicorn, short for Green Unicorn, serves as the WSGI HTTP server for our Django application. It interfaces between Nginx and the Django web framework, handling incoming requests, processing them, and generating responses. Gunicorn utilizes multiple worker processes to concurrently serve incoming requests, effectively utilizing the available system resources and maximizing performance. Through Gunicorn's seamless integration with Django and its ability to scale horizontally by adding more worker processes or deploying multiple instances, we ensured scalability and responsiveness, even during periods of increased traffic or sudden spikes in demand. Benefits: ● Scalability: The combination of Nginx and Gunicorn allowed us to horizontally scale our Django application by adding more server instances or worker processes as needed, accommodating growing user traffic and maintaining consistent performance levels. ● Fault Tolerance: By distributing incoming requests across multiple server instances, our load balancing setup enhanced the fault tolerance of our application, mitigating the impact of server failures or downtime and ensuring uninterrupted service availability. ● Performance Optimization: Nginx's efficient request handling and Gunicorn's multi-worker architecture optimized the performance of our Django application, delivering

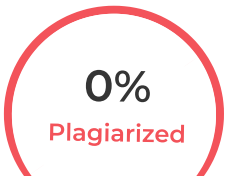
fast response times, low latency, and high throughput, even under heavy loads.

Sources

Plagiarism Detector .net

May 05, 2024

Plagiarism Scan Report



Characters:2707	Words:326
Sentences:15	Speak Time: 3 Min

Excluded URL | None

Content Checked for Plagiarism

CHAPTER 8. PROJECT CONCLUSION The project revolves around the development of a data redundancy removal system aimed at optimizing non-optimized .csv files. To achieve this, a suite of methods including MDNN removal, Exact match removal, Fuzzy match removal, CWOA Match removal, and Selected column removal are employed. These methods collectively serve to streamline the data by eliminating duplicate entries and enhancing data integrity. Technologically, Python is utilized for scripting the system's functionalities, while MongoDB serves as the local database management system. Furthermore, AWS is leveraged for deploying the system publicly on the internet, making it accessible to users beyond local environments. However, several constraints need to be addressed to ensure the system's effectiveness and viability. Performance optimization stands out as a primary concern due to the potentially large datasets the system will encounter. Scalability is also crucial to accommodate increasing loads and data volumes as the project gains traction. Additionally, robust data security measures are imperative to safeguard sensitive information processed by the system, both locally and in the cloud. Furthermore, attention must be given to designing a user-friendly interface to facilitate seamless file uploading, method selection, and result visualization, enhancing overall user experience. Lastly, prudent cost management strategies must be implemented, especially concerning AWS usage, to optimize expenditure and ensure the project's financial sustainability in the long run. Addressing these constraints effectively is paramount for the successful implementation and widespread adoption of the data redundancy removal system. CHAPTER 9. REFRENCES 1. https://bhuvan.nrsc.gov.in/hackathon/iisf2023/topics/Topic12.pdf 2. [An approach to remove duplication records in healthcare dataset based on Mimic Deep Neural Network (MDNN) and Chaotic Whale Optimization (CWO)] Anto Praveena M.D, and Bharathi B, https://journals.sagepub.com/doi/pdf/10.1177/1063293X21992014 3. GRIDFS-FILE Management MongoDB: https://www.mongodb.com/docs/manual/core/gridfs/ 4. Pandas library of Python: https://pandas.pydata.org/docs/getting_started/index.html 5. DEDUPLICATING DATA AND REMOVING REDUNDANCY IN CLOUD - Arun Singh Kaurav, T.Santhosh Kumar, V.Yadigiri Assistant Professor, Assistant Professor, Assistant Professor Computer Science and Engineering Guru Nanak Institutions Technical Campus, Hyderabad, India

https://www.ijcrt.org/papers/IJCRT1892578.pdf 6. https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-gunicorn-on-ubuntu, Erin Glass, Jamon Camisso, and Easha Abid

Sources