DATA DUPLICATION REMOVAL

USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

GANESH KUMAR .G **(730920104030)**

HEMANTH KUMAR .V **(730920104035)**

NIKHIL KUMAR .K **(730920104048)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

EXCEL ENGINEERING COLLEGE(AUTONOMOUS)

KOMARAPALAYAM

APRIL 2024

EXCEL ENGINEERING COLLEGE::KOMARAPALAYAM

BONAFIDE CERTIFICATE

Certified that this project report "**DATA DUPLICATION REMOVAL USING MACHINE LEARNING**" is the bonafide work of "**GANESH KUMAR .G (730920104030)**, **HEMANTH KUMAR .V (730920104035), NIKHIL KUMAR .K (730920104048)"** who carried out the project under my supervision.

SIGNATURE

Dr. P.C. SENTHIL MAHESH M.E., Ph.D.,
HEAD OF THE DEPARTMENT,

Professor,
Department of CSE,
Excel Engineering College,
Komarapalayam-637 303.

SIGNATURE

Mr.M.SATHISH KUMAR M.E.,
SUPERVISOR,

Assistant Professor,
Department of CSE,
Excel Engineering College,
Komarapalayam-637 303.

***Submitted for the University Examination held on*_____**

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Behind every achievement lies an unfathomable sea of gratitude to those who actuated it, without them it would never have into existence. To them, we lay word of gratitude imprinted within ourselves.

We wish our heartfelt thanks to our respected Founder and Chairman of Excel Group Institutions Prof. **Dr. A. K. NATESAN, M.Com., M.B.A., M.Phil., PhD.,** FTA and Vice Chairman **Dr. N. MATHAN KARTHICK M.B.B.S., M.H.Sc (Diabetology)** for allowing us to have the extensive use of the college facilities to do our project effectively.

We express our sincere gratitude and heartfelt thanks to the respected Principal **Dr. K. BOMMANNA RAJA Ph.D.,** for his encouragement and support to complete the project.

We would like to express our profound interest and the sincere gratitude to **Dr. P. C. SENTHIL MAHESH M.E., Ph.D.,** Head of the Department, Department of Computer Science and Engineering for his encouragement and support to complete the project.

We are privileged to express our deep sense of gratitude to Project Supervisor **Mr. M. SATHISH KUMAR M.E.,** Assistant Professor, Department Computer Science and Engineering who gave guidance and support throughout our work and made this as a successful project.

We would like to give our sincere gratitude and heartfelt thanks to our Project Coordinator **Mrs. J. OBURADHA M.E.,** Assistant Professor, Department of Computer Science and Engineering, who gave guidance and support throughout my work and made this as a successful project.

Finally, we thank the Almighty, Parents, Friends and well Wishers for the moral support throughout the project.

ABSTRACT

This project focuses on employing machine learning to address the issue of data duplication. Through the application of advanced algorithms and models, the system is designed to detect and remove redundant data instances, thereby optimizing data quality and operational efficiency. The approach involves training the machine learning model on a diverse dataset, enhancing its ability to accurately identify patterns indicative of duplication. By doing so, the project contributes to the improvement of overall data management processes, ensuring a more reliable and streamlined handling of information. This initiative is particularly relevant in contemporary data-intensive environments, where the proliferation of redundant data poses challenges to data accuracy, storage, and retrieval. The integration of machine learning techniques offers a proactive solution to mitigate these challenges and foster a more effective and resource-efficient data ecosystem. The significance of this project lies in its potential to revolutionize data management practices. By automating the identification and removal of duplicated data, organizations can significantly enhance the quality and reliability of their datasets. This, in turn, leads to more informed decision-making processes, as stakeholders can rely on cleaner, more accurate data for analysis and strategic planning. Moreover, the implementation of machine learning for data duplication removal aligns with the evolving nature of data ecosystems. As the volume and complexity of data continue to grow, traditional manual approaches become increasingly impractical. Machine learning offers a scalable and efficient solution, capable of adapting to evolving data patterns and mitigating the challenges posed by data duplication at scale.

TABLE OF CONTENT

CHAPTER NO.	CONTENT	PAGE NO.
	ABSTRACT	**iv**
	LIST OF FIGURES	**v**
	LIST OF ABBREVIATIONS	**vi**
1.	**INTRODUCTION**	**1**
	1.1. Introduction	1
	1.2. Project Overview	2
	1.3 Problem Statement	2
	1.4 Motivation of Data Removal	2
	1.5 Project Scope	4
	1.6 Project Objectives	6
2.	**LITERATURE SURVEY**	**9**
	2.1 A survey on data duplication	9
	2.2 Aware massive information deduplication in cloud Environment	9
	2.3 A survey on data privacy in cloud	10
	2.4 Deduplication techniques in the cloud	10
	2.5 Review of various data removal models	11
	2.6 Relation based deduplication performance	11
3.	**SYSTEM DESIGN**	**12**
	3.1 System Design	12
	3.1.1 Datasets Collection	13
	3.1.2 Engine Creation	14
	3.1.3 User Interface Design	15
	3.1.4 Content Analysis	16

3.1.5 Quality Assurance 16
3.1.6 Simple Predict Generation 18
3.1.7 Performance Optimization 18
3.1.8 Data Verification 18
3.1.9 Deployment and User Support 18
3.2 Implementation issues and Challenges 19
4. METHOD AND METHODOLOGIES INVOLVED 21
4.1 Methodology 21
4.1.1 Planning 22
4.1.2 Requirement Analysis 22
4.1.3 Designing 23
4.1.4 Building 23
4.1.5 Testing 24
4.2 Hardware and Software Requirement 24
4.2.1 Hardware Requirements 24
4.2.2 Software Requirements 24
4.3 Language specification 25
4.3.1 Python 25
4.3.2 Jupyter Note Book 25
4.3.3 CMD.exe Prompt 26
4.3.4 Anaconda Navigator Environment 26
4.3.5 Functionalities of various toolbars 28
5. SOFTWARE TESTING 29
5.1 Testing 29
5.1.1 Unit Testing 30

5.1.2 Load Testing 30
5.1.3 Beta Testing 30
5.1.4 Acceptance Testing 31
5.1.5 System Testing 31
5.2 Testing Methodologies 34
5.2.1 White Box Testing 34
5.2.2 Black Box Testing 34
5.3 Verification and Validation 34
6. SOURCE CODE AND SCREENSHOTS 35
6.1 Source Code 35
6.2 Implementation Process and Screenshots 39
6.3 Result 39
7. CONCLUSION AND FUTURE ENHANCEMENT 40
8. REFERENCES 41

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
3.1.	System Design	11
3.2.	Support Vector Machine	19
4.1.	General Flow chart for data deduplication	21
4.3.2.	Anaconda Navigator Home page	25
4.3.4.	Jupyter Notebook Launch	26
6.3.	Result	38

LIST OF ABBREVATIONS

ADR	:	Automated Duplication Removal
CDI	:	Content Duplication Identification
D2M	:	Duplication to Minimize
DARA	:	Duplicate Analysis and Removal with AI
DDC	:	Duplicate Data Cleanup
DDM	:	Data Duplication Removal
DDP	:	Data Duplication Purge
DML	:	Duplication Management with Learning
DMD	:	Duplicate Mitigation with Data
DR	:	Data Removal
SDP	:	Similar Data Processing

CHAPTER 1

INTRODUCTION

1.1. Introduction

The introduction Embark on a transformative journey with our data duplication removal project using machine learning. Harnessing cutting-edge algorithms, we aim to enhance data precision by identifying and eliminating duplicates within datasets. This venture promises streamlined analysis, improved accuracy, and resource optimization. Join us in revolutionizing data quality, as we leverage the power of machine learning to create cleaner, more reliable datasets.

1.2. Project Overview

The project focuses on leveraging machine learning to address data duplication issues within datasets. Our objective is to enhance data quality by implementing advanced algorithms that can identify and remove duplicate entries efficiently. This initiative aims to streamline data analysis, improve accuracy, and optimize resource utilization.

The project's significance lies in its potential to revolutionize data management, providing cleaner and more reliable datasets for various applications. Through innovative techniques, we aim to deliver a solution that contributes to improved decision-making processes and reinforces the importance of data integrity in today's technological landscape.

1.3. Problem Statement

The project addresses the pervasive issue of data duplication within datasets, impeding accurate analysis and decision-making. Existing datasets often contain redundant entries, leading to inefficiencies and inaccuracies. The challenge is to develop a robust solution using machine learning algorithms to automatically identify and eliminate duplicate records. This problem statement highlights the critical need for an efficient data deduplication process, emphasizing the project's goal to enhance data quality, streamline workflows, and optimize resource utilization. By tackling this challenge, the project aims to contribute significantly to improving the reliability and usability of datasets across various domains.

1.4. Motivation of Data Removal

The motivation for data removal lies in the imperative to ensure data accuracy and reliability. Duplicates within datasets can skew analyses, compromise decision-making, and hinder overall data quality. By implementing a robust data removal process, we aim to enhance the precision and integrity of information, empowering organizations with trustworthy datasets. This initiative is driven by the need to streamline operations, improve analytical outcomes, and foster a data-driven environment. Ultimately, the motivation behind data removal is to provide users and systems with cleaner, more dependable data, facilitating more informed and accurate insights across diverse applications.

1.1.1. Diversity Challenge

The diversity challenge in data removal involves addressing the complexity arising from varied data sources, formats, and structures. Datasets often exhibit diverse characteristics, making it challenging to design a one-size-fits-all solution for duplicate identification and removal.

1.1.2. Inclusivity and Audience Diversity

Ensuring inclusivity and addressing audience diversity is paramount in our project. The solution aims to accommodate users with varying levels of technical expertise, making it accessible to a diverse audience. We prioritize inclusivity by considering different industries, ensuring the applicability of our data removal approach across various domains. Additionally, we are mindful of addressing potential biases in the machine learning algorithms to guarantee fair outcomes for diverse datasets. By fostering inclusivity and embracing audience diversity, our project endeavors to create a tool that caters to the needs of a broad spectrum of users, promoting equitable and effective data deduplication practices.

1.1.3. Democratizing Data Removal Knowledge

Our project is committed to democratizing data removal knowledge, making it accessible to a wide audience. By providing user-friendly interfaces and documentation, we aim to empower individuals with diverse backgrounds, irrespective of their technical expertise, to effectively utilize data removal tools. Through educational resources and transparent methodologies, we seek to break down complex concepts, ensuring that knowledge about data deduplication is widely understood and applied. This democratization initiative aligns with our vision of fostering inclusivity, enabling a broader community to harness the benefits of improved data quality through accessible and comprehensible data removal practices.

1.1.4. Technological Innovation

Our project is at the forefront of technological innovation, employing advanced machine learning techniques to revolutionize data removal. By integrating state-of-the-art algorithms, we push the boundaries of efficiency in identifying and eliminating duplicate entries within datasets.

1.1.5. Clarity and Formal Accuracy

Clarity and formal accuracy are paramount in our project, emphasizing precise communication and adherence to established standards. Our approach prioritizes clear documentation and transparent methodologies to ensure users understand the data removal process thoroughly. Formal accuracy is maintained through rigorous testing and validation, ensuring the reliability of the machine learning algorithms in identifying and removing duplicate entries. By upholding these principles, we aim to provide users with a robust, trustworthy, and precise data deduplication solution that aligns with industry standards and best practices, fostering confidence in the accuracy and clarity of our project outcomes.

1.1.6. Expanding Reach and Impact

Expanding the reach and impact of our project is a key objective. We are dedicated to developing user-friendly interfaces and documentation to facilitate widespread adoption across diverse industries. Collaboration with open-source communities and integration with popular data platforms are part of our strategy to broaden accessibility. Through workshops, webinars, and educational outreach, we aim to empower a global audience with the knowledge and tools for effective data removal. This concerted effort seeks to extend the positive impact of our project, promoting data quality enhancement and efficient deduplication practices on a broader scale.

1.5. Project Scope

The project scope encompasses the development and implementation of machine learning algorithms for the identification and removal of duplicate entries within datasets. It involves creating user-friendly interfaces and potentially integrating with various data platforms to enhance accessibility.

1.5.1. Data Removal

Data removal refers to the process of identifying and eliminating redundant or duplicate entries within datasets. This crucial step enhances data quality by ensuring that information is accurate, reliable, and free from unnecessary repetition. Machine learning algorithms are often employed in data removal processes to automate the identification of duplicates, contributing to more efficient and streamlined data management. The goal of data removal is to optimize the use of resources, improve analytical accuracy, and promote the creation of cleaner datasets, ultimately facilitating better decision-making and insights in various domains.

1.5.2. User-Friendly Interface

A user-friendly interface is designed to be intuitive and accessible, allowing users to interact with software or systems easily. In our data removal project, we prioritize creating a user-friendly interface that simplifies the process of deduplicating datasets. This involves a clear layout, straightforward navigation, and interactive elements to guide users through the removal process. By focusing on user experience, we aim to ensure that individuals with varying levels of technical expertise can efficiently utilize our data removal tool, contributing to a more inclusive and effective data management experience.

1.5.3. Ensuring Quality Standards

Ensuring quality standards is integral to our project. We employ rigorous testing, validation, and adherence to established best practices throughout the development process. The machine learning algorithms are fine-tuned to meet high-performance benchmarks, and user feedback is actively incorporated to enhance functionality. Additionally, documentation is crafted with precision to uphold clarity and formal accuracy.

1.6. Project Objectives

The primary objective of our project is to develop a sophisticated data removal solution using machine learning algorithms. This entails creating a user-friendly interface for efficient deduplication, accommodating diverse data types and structures. The goal is to enhance data quality by automating the identification and removal of duplicate entries, ultimately contributing to more accurate analyses and decision-making processes. Through a commitment to inclusivity, technological innovation, and adherence to quality standards, our project aims to revolutionize data management, providing users with a powerful tool for creating cleaner and more reliable datasets across various industries.

1.6.1. Enhance Accessibility

To enhance accessibility, our project prioritizes the development of user-friendly interfaces and clear documentation. We focus on creating intuitive tools that cater to users with varying levels of technical expertise, ensuring a seamless experience. Additionally, efforts include compatibility with popular data platforms and collaboration with open-source communities to broaden the reach of our solution. Through educational initiatives like workshops and webinars, we aim to empower a diverse audience with the knowledge and skills to utilize our data removal tools effectively. By fostering inclusivity and broadening access, our project seeks to make advanced data deduplication capabilities available to a wider user base.

1.6.2. Help People Understand

Our project aims to demystify data removal by providing clear documentation, user-friendly interfaces, and educational resources. We strive to help people understand the importance of deduplication in enhancing data quality and streamlining processes. Through transparent methodologies and outreach initiatives, we aim to empower users with varying levels of expertise to grasp the intricacies of our machine learning-driven data removal solution.

1.6.3. Various Document Types

If you're referring to various document types in the context of our data removal project, we ensure compatibility with diverse file formats and structures. Our solution is designed to handle different types of documents, such as spreadsheets, databases, and text files, accommodating the variability in data sources. This flexibility allows users to apply our data removal algorithms to a wide range of document types, contributing to the versatility and practicality of the solution across various industries and use cases.

1.6.4. Capture Problem

The capture problem in our data removal project revolves around accurately identifying and capturing duplicate entries within datasets. It involves developing machine learning algorithms that can effectively recognize redundancies across various data types and structures. Addressing the capture problem is crucial to ensuring the precision of the removal process, as inaccuracies at this stage can lead to the retention of duplicates in the dataset. Our project focuses on fine-tuning algorithms and methodologies to overcome the capture problem, aiming for a solution that robustly captures and eliminates duplicate entries, ultimately enhancing the overall quality of the data.

CHAPTER 2

LITERATURE SURVEY

2.1. A survey on data dupilication

Authors: Cho,Ei Mon et.al., 2022 Nov;
Big knowledge Cloud Deduplication supported Verifiable Hash oblique cluster Signcryption‖ during this paper, we've designed a theme that supports secure deduplication wherever many teams are sharing knowledge by exploitation VHCGS. this can be an endeavor to undertake out cross-group user deduplication in an exceedingly real huge knowledge management. In doing therefore, we are taking the utility of existing schemes instead of proposing a wholly new one. we introduce a framework for a gaggle signcryption theme which might shield against duplication for the cloud suppliers and defend against unpredictable knowledge attacks.

2.2. Application-Aware massive information Deduplication in Cloud Environment

Authors :Fu,Yinjin,et.al.,
Application-Aware massive information Deduplication in Cloud Environment‖during this paper, we describe AppDedupe, an application-aware ascendable inline distributed deduplication frame-work for large information management, that achieves a trade-off between ascendable performance and distributed deduplication effectiveness by exploiting application awareness, information similarity and neighborhood. It adopts a two-tiered information routing theme to route information at the super-chunk granularity to scale back cross-node information redundancy .

2.3. A survey on data privacy in cloud

Authors: Yang,Xue,et.al.,

Achieving economical and Privacy-Preserving Cross Domain massive knowledge Deduplication in Cloud. Cloud storage adoption, notably by organizations, is probably going to stay a trend within the predictable future. This is, unsurprising , because of the conversion of our society. One associated analysis challenge is the way to effectively scale back cloud storage prices because of knowledge duplication. during this paper, we projected an economical and privacy-preserving massive knowledge deduplication in cloud storage for a three-tier cross domain design. we then analyzed the safety of our projected theme and incontestable that it achieves improved privacy protective, answerableness and knowledge handine.

2.4. Machine learning for deduplication techniques in cloud

Authors: Karthika et.al.,

Perlustration on techno level classification of deduplication techniques in cloud for giant knowledge storage‖ a brand new trend has set in storage wherever knowledge de-duplication plays, a completely unique role in compression and alternative knowledge reduction in sophisticated as a bottom component of the answer. knowledge deduplication technique improves knowledge protection, that will increase the speed of service, and reduces the prices and therefore the use of information measure. De-duplication is integrated with cloud space for storing, this helps within the simple maintenance of knowledge and eradicating the replica‘s within the cloud server. Cloud computing, storage resources is with efficiency used this permits each organization to create their own non-public cloud and hybrid cloud in step with their functions and desires.

2.5. A Comprehensive Review of Various Data Removal Models

Author: Naresh et.al.,

Bucket based mostly information deduplication technique for information storage system‖ In big information storage information is just too large and expeditiously store information is tough task. to unravel this downside Hadoop tool provides HDFS that manages information by maintain duplication of information however this inflated duplication. To expeditiously stores information and de-duplication the info this paper presents a bucket based mostly technique. In planned technique totally different buckets are wont to store information and once same information is accessed by map cut back i.e. already keep in bucket then that information are going to be discarded therefore this method positively will increase potency of big data storage.

2.6. A survey on relation based deduplication performance booster

Author:Yujuan Tan et.al.,

In this analysis article propose a relation based mostly deduplication performance booster for each cloud backup and restore operations known as CAB dedupe. In which, they establish that file and knowledge chunk are modified or stay unchanged, They illustrate the role of CABdedupe in a backup system by mistreatment backup consumer and backup server to represent the functionalities of original consumer and server in module in existing backup system.

CHAPTER 3

SYSTEM DESIGN

3.1. System Design

The system for data duplication removal using machine learning involves preprocessing data, extracting relevant features, and selecting a suitable model like Siamese Networks or Random Forest. Training data, incorporating labeled pairs of duplicates and non-duplicates, is crucial. Defining similarity metrics, fine-tuning thresholds, and integrating the model into the database system ensure effective duplicate detection. Scalability considerations, monitoring, and periodic retraining for model maintenance are essential. A user-friendly interface aids manual validation, while security measures protect sensitive data. Continuous testing with synthetic and real-world datasets ensures robust performance and compliance with privacy regulations. Adapting the design to specific data characteristics and application requirements is key for successful implementation.

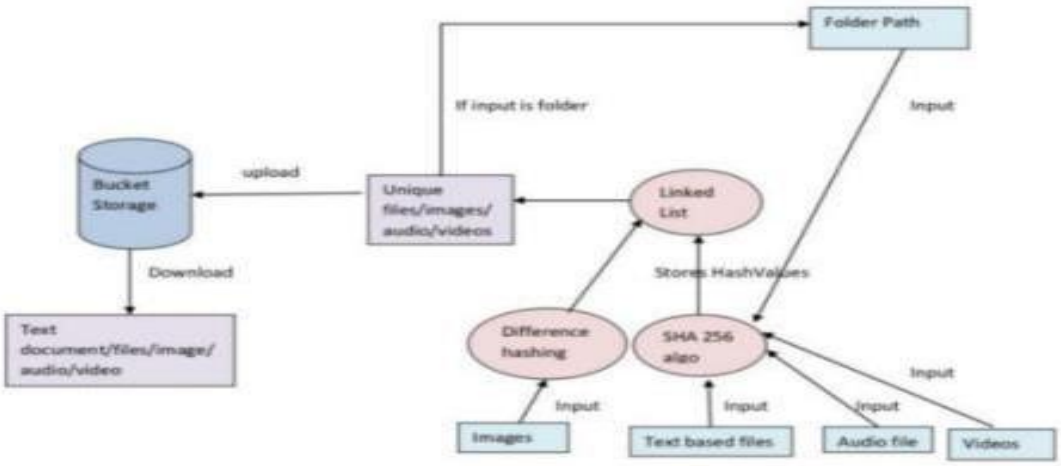


Fig: 3.1 System Design

3.1.1. Datasets Collection

To collect a dataset for your data duplication removal machine learning project, first, clearly define your data requirements. Identify the sources, which can include databases, APIs, or files, and consider legal and ethical considerations. Decide on a sampling strategy to obtain a representative dataset and then extract the data using appropriate tools or scripts. If using supervised learning, label the data to indicate duplicates. The steps in dataset collection are:

a) Define Data Requirements

Clearly outline the objectives of data collection, specifying the information needed to achieve the project's goals, such as patient demographics, medical history, and lifestyle factors.

b) Select Source Language Materials

Enumerate and prioritize the variables critical to diabetes prediction, including blood glucose levels, BMI, age, family history, and lifestyle habits.

c) Review Existing Datasets:

Explore publicly available healthcare datasets, research repositories, and clinical databases to identify potential sources that align with the project's scope.

d) Consider Ethical and Legal Aspects:

Ensure compliance with ethical standards and data protection regulations. Obtain necessary approvals and permissions before accessing or using any datasets.

e) Define Inclusion and Exclusion Criteria:

Clearly define the criteria for including or excluding data points, ensuring relevance and consistency across the dataset.

f) Select Data Collection Methods:

Choose appropriate methods for data collection, which may include electronic health records, surveys, wearable devices, or a combination of sources.

g) Ensure Data Quality:

Implement measures to maintain data accuracy and completeness, addressing issues such as missing values, outliers, and inconsistencies. **h)**

3.1.2. Engine Creation:

Translation engine is the core part of the software and it is developed using machine learning and NLP (natural language processing) techniques. We train the translation engine with bilingual data (which we have collected) for improved accuracy.

Steps in translation engine creation are:

a) Algorithm Selection:

Choose suitable machine learning algorithms for diabetes prediction, considering factors such as decision trees, support vector machines, or neural networks based on the project's requirements. **b) Feature Integration:**

Integrate relevant features identified during the feature selection phase into the predictive model, ensuring that the chosen variables contribute effectively to accurate predictions.

c) Data Preparation:

Preprocess the collected datasets, addressing missing values, normalizing numerical features, and encoding categorical variables to create a clean and standardized input for the machine learning model.

d) Model Training:

Train the selected machine learning model using the preprocessed datasets. This involves exposing the model to historical data to learn patterns and relationships that can later be applied to new, unseen data..

e) Hyperparameter Tuning:

Optimize the hyperparameters of the chosen machine learning algorithm to enhance the model's performance. This iterative process involves adjusting parameters to achieve the best predictive accuracy.

f) Validation and Cross-Validation:

Validate the model using a separate dataset not used during training to assess its generalization ability. Implement cross-validation techniques to ensure robustness and minimize overfitting.

g) Evaluation Metrics:

Utilize relevant evaluation metrics such as accuracy, precision, recall, and F1 score to assess the model's performance and its ability to correctly predict instances of diabetes.

3.1.3. User Interface Design:

User interface (UI) design involves creating interfaces that are visually appealing and easy to use. To start, understand the target audience through user research. Develop wireframes as a blueprint for the interface, focusing on functionality and user flow. Design a visually cohesive interface that aligns with the brand. Ensure an intuitive navigation system, allowing users to easily move through the interface. Maintain consistency in design elements for a unified look. Make the UI responsive to different devices and screen sizes.

3.1.4. Content Analysis:

Content analysis is a research method that systematically examines the content of communication, such as text or multimedia, to extract meaningful insights. The process begins with defining a clear research objective and selecting a representative sample for analysis. Researchers determine the unit of analysis, whether it be words, phrases, or entire documents. A coding scheme is then developed to categorize content based on predefined themes or categories. This scheme is consistently applied during the coding process. To ensure accuracy, inter-coder reliability is established when multiple coders are involved. Finally, data analysis involves extracting patterns, relationships, and themes from the coded content using either qualitative or quantitative methods, depending on the research goals.

3.1.5. Quality Assurance:

Quality Assurance is a meticulous process ensuring that every aspect of the project meets predefined standards. It involves systematic checks, validations, and testing procedures to guarantee the reliability, functionality, and overall excellence of the project deliverables. Quality assurance encompasses various aspects, including code reviews, documentation accuracy, and adherence to project requirements.

a) Define Quality Metrics:

Quality metrics are quantifiable measures used to assess the excellence and compliance of a project or product against predefined standards.

b) Automated Assessment:

Automated Assessment involves the use of automated tools and algorithms to evaluate, analyze, and provide feedback on various aspects of a system or project.

c) Manual Assessment:

Manual assessment for diabetic prediction involves human-driven evaluation of data, models, and results to ensure accuracy, relevance, and clinical validity.

d) Feedback Loop:

continuous process where outputs of a system are circled back as inputs, enabling ongoing refinement and improvement. In various contexts, including software development.

e) Domain-Specific Review:

Domain-Specific Review tailors evaluations to industry standards, regulations, and unique requirements, ensuring project alignment and effectiveness within a specific domain.

f) Cultural Sensitivity Check:

A Cultural Sensitivity Check involves assessing content, communication, or actions to ensure they respect and align with diverse cultural perspectives, promoting inclusivity and avoiding unintentional offense.

g) Prediction Validation:

Prediction validation is the process of systematically assessing and confirming the accuracy, and effectiveness of predictions made by a model.

h) Final Review:

The Final Review is a comprehensive assessment conducted before project completion, ensuring all requirements are met.

3.1.6. Simple prediction Generation:

Simple Prediction Generation involves the creation of straightforward predictive models to generate basic forecasts or insights based on available data. This phase focuses on developing models with simplicity and efficiency, often using fundamental algorithms, to quickly generate predictions and insights for initial assessments or basic decision-making processes.

3.1.7. Performance Optimization:

Performance Optimization is the process of refining and enhancing the efficiency, speed, and overall effectiveness of a system or application. This involves identifying bottlenecks, improving algorithms, and streamlining processes to achieve optimal performance, ensuring that the system operates at its highest potential and meets specified performance criteria. This phase is crucial for delivering a high-performing and responsive solution, enhancing user experience and overall system functionality.

3.1.8. Data Verification:

Data Verification is the process of confirming the accuracy, integrity, and reliability of collected data. This involves cross-referencing information against authoritative sources, checking for completeness and consistency, and ensuring that the data aligns with predefined standards. The goal of data verification is to enhance the overall quality of datasets, reduce errors, and ensure that the information is trustworthy and suitable for analysis or decisionmaking processes.

3.1.9. Deployment and User Support:

Deployment involves the seamless rollout of the solution into the live environment, ensuring accessibility for users. Simultaneously, user support encompasses comprehensive training, technical assistance, and continuous monitoring to facilitate a smooth post-deployment experience.

3.2. Implementation issues and challenges

In the implementation phase several issues and challenges may arise:

a). Technical Hurdles:

Overcoming technical complexities, such as integration challenges, system compatibility issues, or unforeseen technical constraints during implementation.

b) Resource Allocation:

Efficiently allocating resources, including human resources, time, and budget constraints, to meet project timelines and objectives.

c) Adaptation to Change:

Adapting to changes in project requirements, evolving technologies, or unforeseen external factors that may impact the implementation plan.

d) Stakeholder Collaboration:

Ensuring effective collaboration and communication among stakeholders to address concerns, gather feedback, and align expectations during the implementation process.

e) Quality Assurance:

Implementing robust quality assurance measures to detect and rectify any defects or issues that may arise during the implementation, ensuring a highquality final product.

f) Data Migration Challenges:

Managing the migration of data from existing systems to the new solution, addressing potential data inconsistencies, and ensuring data integrity.

g) User Training and Adoption:

Facilitating user training and ensuring a smooth transition to the new system, addressing any resistance or challenges in user adoption.

h) Scalability Concerns:

Planning for scalability and addressing potential challenges related to system growth and increased user load over time.

i) Regulatory Compliance:

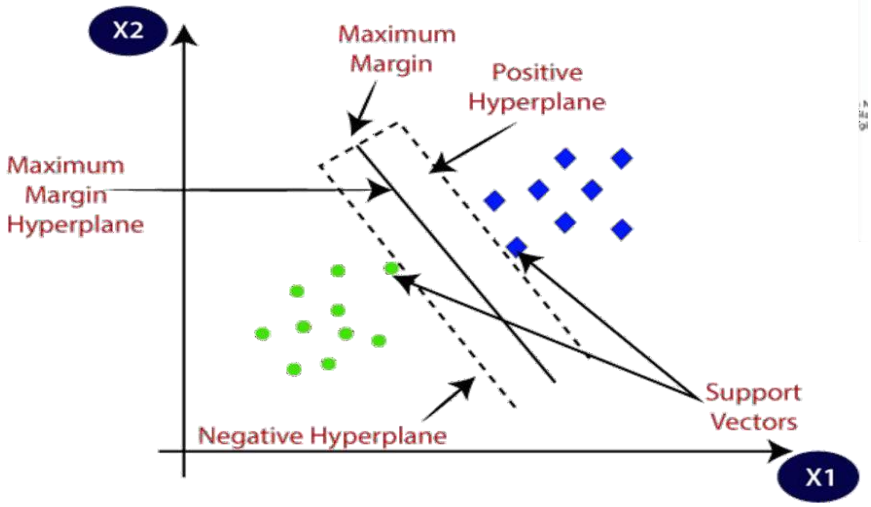


Fig 3.2. Support vector Machine

Navigating and adhering to relevant regulatory requirements and compliance standards applicable to the implementation domain.

CHAPTER 4

METHOD AND METHODOLOGIES INVOLVED

4.1. Methodology

Methodology is the systematic approach used in a research study, outlining the steps for data collection, analysis, and interpretation. It involves defining the research design (experimental, observational, qualitative, or quantitative), specifying the target population, and detailing the methods for selecting participants or data points. The methodology also covers the tools or instruments used for data collection, the definition of variables and measures, and the analytical methods applied for interpretation, including any software used. Ethical considerations, such as participant confidentiality and informed consent, are addressed, and a timeline for each research phase is provided. Researchers also acknowledge potential limitations or boundaries of the study, ensuring transparency and allowing for the replication or extension of the research.

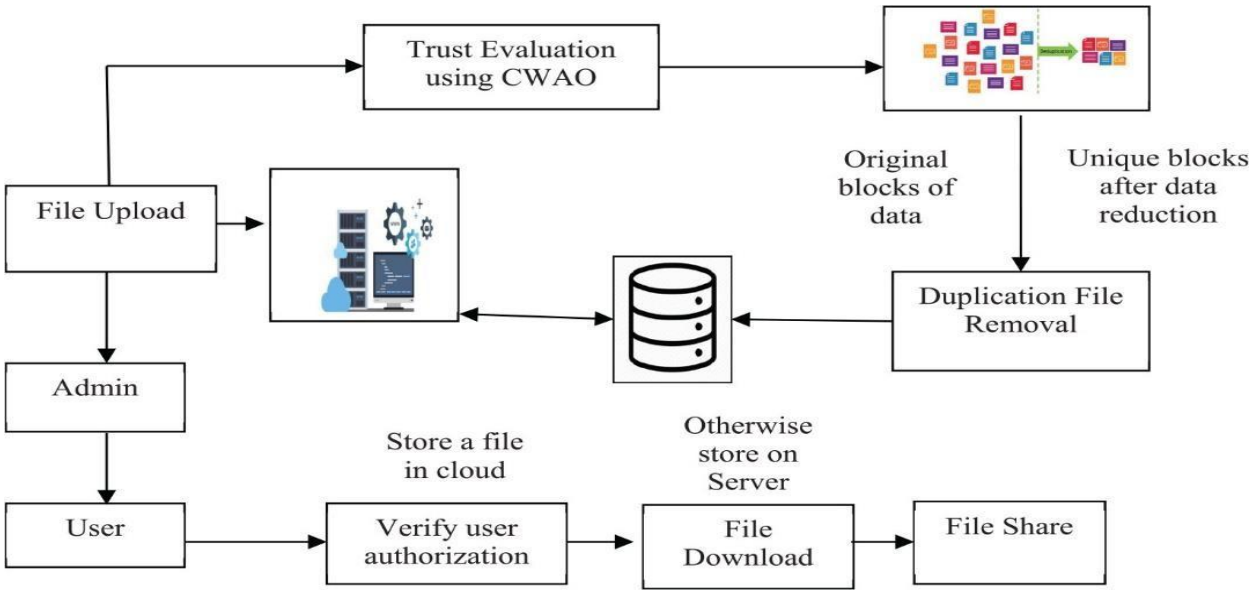


Fig 4.1. General flow chat for data deduplication

4.1.1. Planning:

Planning is a systematic process involving the definition of clear objectives, breaking them down into manageable tasks, and allocating resources effectively. It encompasses setting specific goals, identifying the necessary actions, and determining the required resources such as personnel, finances, technology, and time. Establishing a timeline with deadlines for each task helps track progress and maintain schedule adherence. Additionally, planning involves assessing potential risks and developing strategies to mitigate challenges. Open communication and collaboration among team members are crucial for successful plan execution. Overall, planning provides a structured approach to achieving goals, ensuring efficiency and alignment within the team or organization.

4.1.2. Requirement Analysis:

For a data duplication removal project using machine learning, the requirement analysis involves identifying key stakeholders such as data analysts and administrators. It includes gathering specific details on the types of duplication to be addressed, desired accuracy levels, and preferences for real-time or batch processing. Documenting these requirements involves detailing characteristics of duplicate records, preferred similarity metrics, and any relevant business rules. The analysis phase prioritizes features based on factors like critical data fields and acceptable thresholds for false positives/negatives. Validation ensures accurate representation of stakeholder needs, while verification checks for consistency and feasibility within machine learning capabilities.

4.1.3. Designing:

Designing a data duplication removal system using machine learning involves defining the system architecture, mapping data flow, selecting appropriate algorithms, and implementing features for effective model training. Integration with the database, optional user interface design, scalability considerations, and security measures are integral aspects.

4.1.4. Building:

Building a data duplication removal system involves coding components based on the design, training the machine learning model, and integrating it with the database. Optional development of a user interface enhances interaction. Scalability measures ensure adaptability to growing datasets, while security features protect sensitive information. Rigorous testing, including unit and integration tests, ensures functionality and reliability. Once built, the system is deployed into production, where its performance is monitored, and any issues are promptly addressed. This systematic building process transforms the designed solution into a functional and efficient tool for data deduplication using machine learning.

4.1.5. Testing:

Testing the data duplication removal system is critical for reliability. Unit testing verifies individual components, integration testing assesses seamless interactions, and performance testing evaluates responsiveness and scalability. Optional user interface testing ensures a smooth user experience, while security testing checks for vulnerabilities. Scalability testing assesses efficiency with increasing data. End-to-end testing evaluates the entire system's functionality, ensuring a robust and effective solution for data deduplication. Rigorous testing safeguards against issues, ensuring optimal performance and user satisfaction, and contributes to the overall success of the machine learning-based deduplication project.

4.2. Hardware and Software Requirement

4.2.1. Hardware Requirements

The hardware requirements for a data duplication removal project using machine learning depend on the size and complexity of your dataset. Generally, you'll need a machine with a powerful CPU (e.g., Intel Core i7 or higher), sufficient RAM (16GB or more), and a GPU (NVIDIA GeForce or equivalent) to accelerate the training process. Additionally, having ample storage space is crucial for handling large datasets. Consider SSDs for faster read/write speeds.

4.2.2. Software Requirements

- **Operating system** : Windows 8 / 10
- **Programming Language** : Python
- **DL Libraries** : Numpy, Pandas

4.3. Language specification

4.3.1. Python

Python is an easy to learn, powerful programming language. It has efficient high- level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting

- **Python is Interpreted** − Python is processed at runtime by the interpreter.

You do not need to compile your program before executing it. This is similar to PERL and PHP.

- **Python is Interactive** − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

4.3.2. Jupyter Note Book :

Jupyter Notebook is the one of the applications of Anaconda Navigator .Anaconda Navigator has several other applications like CMD.exe Prompt, Datalore, IBM Watson Studio Cloud, Jupyter Lab, Jupyter Notebook, Powershell Prompt, Qt Console, Spyder ,Glueviz, Orange 3 ,PyCharm Professional and RStudio. These applications are useful for various developing and designing various systems.

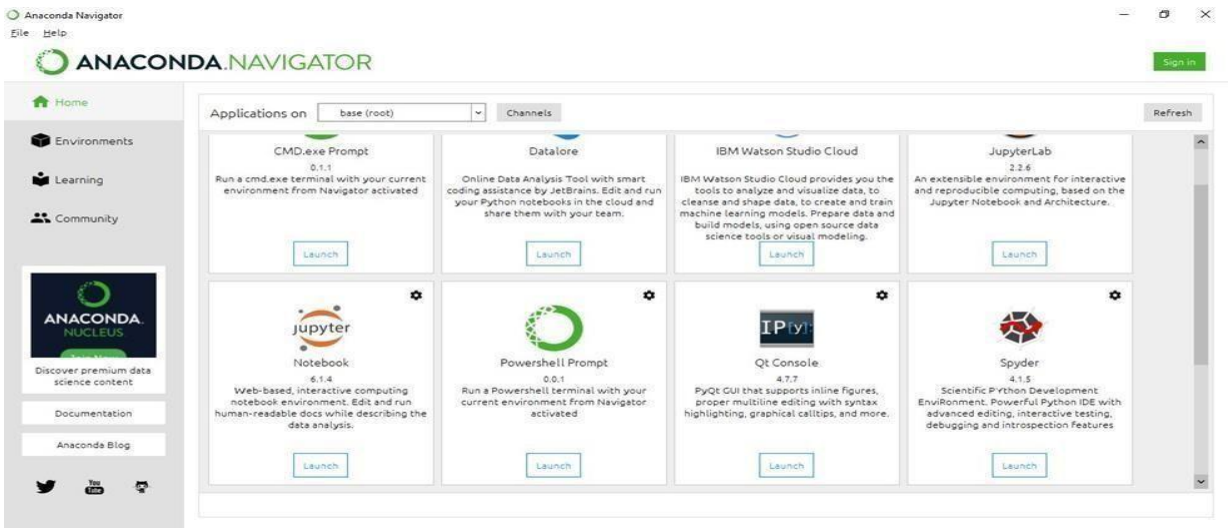


Fig 4.3.2. Anaconda Navigator Home page with various application.

The Anaconda software is Open Source, which means the code, the schematics, design, etc. are all open for anyone to take freely and do what they like with it. This means there is nothing stopping anyone from taking the schematics and anyone can contribute to the open source. The main purpose of open source anyone contribute the code. This increases the reusability of the code.

4.3.3. CMD.exe Prompt:

Run a cmd.exe terminal with your current environment from Navigator activated

4.3.4. Anaconda Navigator Environments:

The Navigator consists of Environment tab which specifies various environments available.

There were of four types of packages possible

- Installed
- Not Installed
- Updatable • Selected

Fig 4.3.4. Jupyter Notebook Launch

Jupyter NoteBook:

Jupyter notebook is a web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. Since it is an web-based application it will be launched from anaconda navigator .The main page of Jupyter notebook contains three functionalities Files, Running and Clusters

Files: In the files we can access all the system and we can upload a new file or create a new file. we will also have details like Name, Last Modified, File Size.

Running: In the running we can see the terminals and notebooks that are currently running.

Clusters: Clusters tab is now provided by IPython parallel.

When we open the required file for working the tool bar contains various operations.

Names of toolbars in above image in the order

1)Save and Check Point 2) Insert cell below

3)Cut selected cells 4) Copy Selected cells

5) Paste cells below 6) move selected cells up

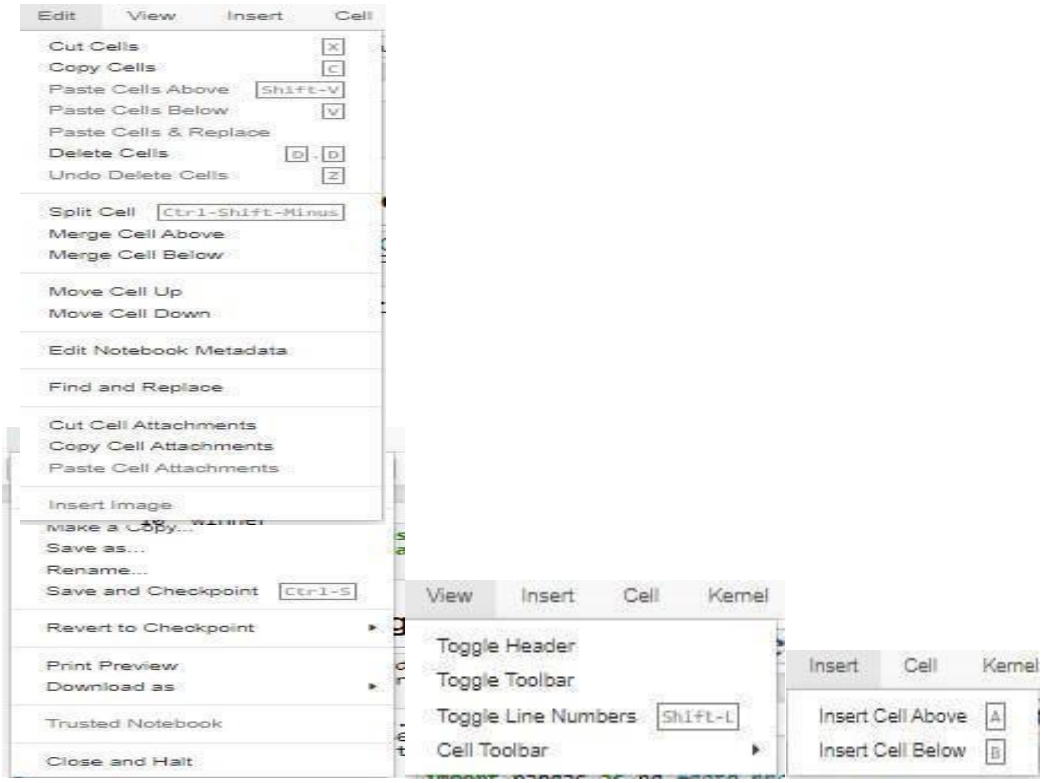
7) Move selected cells down select below 8) Run current cell and

9)Interrupt the kernel 10) Restart the kernel 11)Restart the kernel and re-run the whole note book 12) open the command pallete The table shows the functionalities of the various toolbars.

4.3.5. Functionalities of various toolbars

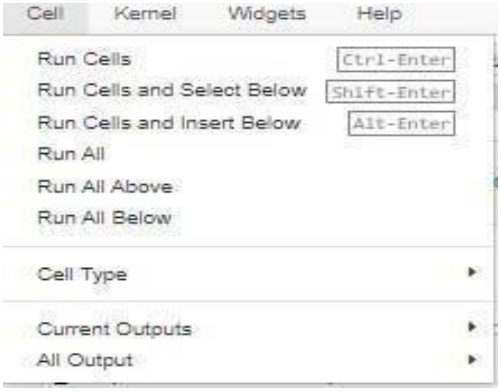
It also provides some more functionalities through menu bar.

- File menu contains functionalities like New Notebook, open, make a copy, save as ,Rename , Revert check point etc.
- Edit menu consists of Cut Cells, Copy Cells, Delete Cells etc.



- View menu consist of Toggle header , Toggle toolbar , Toggle Line Numbers, Cell Toolbar
- Insert menu consists Insert cell above, Insert Cell Below
- Run menu consists of run cells, run cells and select below. run cells and insert below , run all

  , run all above , run all below , cell type , current outputs , all output



CHAPTER 5

SOFTWARE TESTING

5.1. TESTING

Software testing plays a crucial role in offering an unbiased and independent evaluation of software, enabling businesses to comprehend and assess the risks associated with software implementation. Test techniques encompass the systematic execution of a program or application with the aim of identifying software bugs, errors, or defects, and validating the software product's suitability for use..

In the current project, six types of testing are conducted, each contributing to a comprehensive evaluation of the software's performance and robustness.

There are six kinds of testing is done in our project.

1. Unit testing
2. Load Testing
3. Beta Testing
4. Acceptance Testing
5. System Testing

5.1.1. Unit Testing

Unit testing is a critical phase in software development, focusing on evaluating individual units or components of an application. As an application comprises various units and modules, detecting errors at the unitlevel is relatively straightforwardand consumes less time, given the smaller scale of these components.

5.1.2. Load testing

Load testing is a crucial process involving the application of demand to a system or device to assess and measure its response. In the professional application testing community, the term "load testing"

5.1.3. Beta Testing

Load testing is a vital facet of application testing, where demand is applied to assess a system's response. Within the professional testing community, "load testing" typically denotes simulating anticipated usage scenarios for an application program. This process helps gauge the system's performance under various workloads and user interactions, ensuring stability and reliability. Load testing is crucial for identifying performance bottlenecks, predicting scalability, and optimizing resource allocation.

5.1.4. Acceptance Testing

In engineering and its sub-disciplines, acceptance testing is a crucial evaluation conducted to ascertain whether the stipulations of a specification or contract are fulfilled. This testing may encompass chemical, physical, or performance assessments. When customers undertake acceptance testing for applications, it is termed user acceptance testing, end-user testing, site testing, or field testing.

5.1.5. System Testing

System testing, alternatively known as system-level testing or system integration testing, is a pivotal phase in quality assurance. During this process, a QA team systematically assesses the interactions among diverse components within an application within the complete, integrated system. The primary objective is to ensure seamless collaboration among different elements, validating that the integrated system or application functions cohesively and meets specified requirements. System testing encompasses a comprehensive evaluation of the overall system, focusing on its performance, functionality, and interoperability.

a) Improved product quality:

The exhaustive system testing process significantly enhances product quality. As an integrated system undergoes scrutiny through multiple test sets in the product development cycle, it offers insights into the product's capability to function effectively across diverse platforms and environments. This comprehensive evaluation ensures that the product not only meets individual component specifications but also excels in the broader context of the entire system.

b) Error reduction:

In the development of complex systems, errors are inevitable, and system testing plays a crucial role in identifying these issues. This phase verifies the system's code and functionality against its specified requirements. Importantly, system testing serves as a final check, revealing errors that might not have been detected during earlier stages like integration and unit testing.

c) Cost savings:

Detecting system defects later in the project lifecycle can be more time-consuming to address. Timely and continuous system testing not only helps in reducing unexpected costs and project delays but also grants project managers better control over budgets.

d) Security:

Systematic testing helps ensure that the tested system is free from weaknesses that could pose risks to end users and system data. By identifying and addressing potential threats during the testing phase.

f) Customer satisfaction:

System testing provides crucial visibility into the stability of a product at every stage of development. This transparency not only builds customer confidence but also enhances the overall user experience. By systematically assessing the product's functionality, performance, and integration, system testing ensures that potential issues are identified and addressed early in the development process.

h) Easier code modification:

System testing is a critical phase in software development that plays a vital role in identifying code problems. Detecting and rectifying issues during the system testing phase is generally more efficient than

System testing includes various testing which are as:

Performance testing:

Performance testing measures the speed, average load time, stability, reliability and peak response times of the system under various conditions.

Usability testing:

These are tests to evaluate if a system is easy to use and functional for the end user. Metrics, including user error rates, task success rates, the time it takes a user to complete a task and user satisfaction, are used during testing.

Load testing:

This is testing to determine how a system or software performs under a real-life extreme load and test scenarios. Metrics, such as throughput, number of users and latency, are measured through this testing.

Migration testing:

This is conducted to ensure smooth migration of legacy systems to new systems without disruptions, data loss or downtimes.

5.2. TESTING METHODOLOGIES

5.2.1. White-Box Testing (GLASS-BOX TESTING)

White-box testing is a test methodology that concentrates on the internal control structures of a program.

5.2.2. Black-Box Testing:

Black-box testing concentrates on the functional requirements of software, serving as a complementary rather than an alternative approach to

5.3. Verification and Validation:

Validation is a process of finding out if the product being built is right?Whatever the software product is being developed

CHAPTER 6

SOURCE CODE AND SCREENSHOTS

6.1. Source Code

```
import pandas as pd
import hashlib
# Read the CSV file
data_before = pd.read_csv("C:\\Users\\91630\\Downloads\\inputt.csv"
# Calculate checksum before removing duplicates
checksum_before = hashlib.md5(data_before.to_string().encode()).hexdigest()
# Remove duplicates
data_after = data_before.drop_duplicates()
# Display the duplicate data
duplicate_data = data_before[data_before.duplicated(keep=False)]
print("Duplicate Data:")
print(duplicate_data)# Calculate checksum after removing duplicates
checksum_after = hashlib.md5(data_after.to_string().encode()).hexdigest()

# Compare checksums
if checksum_before == checksum_after:
    print("No significant changes after removing duplicates.")
else:
    print("Changes detected after removing duplicates.")

print("Checksum before:", checksum_before)
print("Checksum after:", checksum_after)
```

```
import pandas as pd
from tabulate import tabulate
import subprocess

def read_data(file_path):
    """
    Read data from a CSV file.

    Parameters:
        file_path (str): Path to the CSV file.

    Returns:
        pd.DataFrame: DataFrame containing the read data.
    """
    try:
        data = pd.read_csv(file_path)
        return data
    except FileNotFoundError:
        print(f"Error: File '{file_path}' not found.")
        return None
    except Exception as e:
        print(f"An error occurred while reading the file: {e}")
        return None

def display_duplicate_data(data):
    """
    Display duplicate data from a DataFrame.

    Parameters:
        data (pd.DataFrame): DataFrame containing the data.
    """
    duplicate_data = data[data.duplicated(keep=False)]
    if not duplicate_data.empty:
        print("Duplicate Data:")
```

```
        print(tabulate(duplicate_data, headers='keys', tablefmt='psql'))
    else:
        print("No duplicate data found.")
def remove_and_merge_duplicates(data):
    """
    Remove duplicates from a DataFrame and merge similar records.

    Parameters:
        data (pd.DataFrame): DataFrame containing the data.

    Returns:
        pd.DataFrame: DataFrame with duplicates removed and similar records
merged.
    """
    data.drop_duplicates(inplace=True)
    return data

def export_to_pdf(data, file_path):
    """
    Export DataFrame to a PDF file.

    Parameters:
        data (pd.DataFrame): DataFrame containing the data.
        file_path (str): Path to save the PDF file.
    """
    try:
        latex = data.to_latex(index=False)
        with open(file_path.replace('.pdf', '.tex'), 'w') as f:
            f.write(latex)
        subprocess.run(['pdflatex', file_path.replace('.pdf', '.tex')])
        print(f"Data exported to PDF: {file_path}")
    except Exception as e:
        print(f"An error occurred while exporting to PDF: {e}")
```

```
def export_to_excel(data, file_path):
    """
    Export DataFrame to an Excel file.
Parameters:
        data (pd.DataFrame): DataFrame containing the data.
        file_path (str): Path to save the Excel file.
    """
    try:
        data.to_excel(file_path, index=False)
        print(f"Data exported to Excel: {file_path}")
    except Exception as e:
        print(f"An error occurred while exporting to Excel: {e}")

def main():
    file_path = "C:\\Users\\91630\\Downloads\\inputt.csv"
    pdf_file_path = "output.pdf"
    excel_file_path = "output.xlsx"

    # Read data
    data = read_data(file_path)
    if data is None:
        return

    # Display duplicate data
    display_duplicate_data(data)

    # Remove duplicates and merge similar records
    new_data = remove_and_merge_duplicates(data)

    # Display new data after removal and merging
    print("\nNew Data After Removal and Merging:")
    print(tabulate(new_data, headers='keys', tablefmt='psql'))

    # Export to PDF
    export_to_pdf(new_data, pdf_file_path)

    # Export to Excel
    export_to_excel(new_data, excel_file_path)
```

```
if _name_ == "_main_":
    main()
```

6.2. IMPLEMENTATION PROCESS SCREENSHOTS

AI-based techniques for data deduplication: The company can use AI-based techniques such as machine learning and deep learning to identify and remove duplicate customer entries. These techniques use algorithms to analyze the data and find patterns that indicate duplicate entries.

Training datasets for AI-based techniques: To use AI-based techniques for data deduplication, the company needs to prepare a training dataset. The dataset should include examples of duplicate and non-duplicate customer entries to train the AI model.

6.3. RESULT

Duplicate Data:

	ln	dob	gn	fn	is_duplicate
5	BLAND JR	21-02-1962	F	WILLIAM	1
17	BLAND JR	21-02-1962	F	WILLIAM	1
18	BLAND JR	21-02-1962	F	WILLIAM	1
24	MICHAELSON JR	25-10-1953	M	ROY	1
26	MICHAELSON JR	25-10-1953	M	ROY	1
36	HOUGHTON JR	31-01-1946	M	LAWRENCE	1
37	HOUGHTON JR	31-01-1946	M	LAWRENCE	1

New Data After Removal and Merging:

	ln	dob	gn	fn	is_duplicate
0	SMITH JR	01-03-1968	F	WILLIAM	0
1	ROTHMEYER JR	01-03-1968	F	WILLIAM	0
2	BLAND III	21-02-1962	F	WILLIAM	1
3	BLAND JR	21-02-1962	F	BILL	0
4	BLAND	21-02-1962	F	WILLIAM	1
5	BLAND JR	21-02-1962	F	WILLIAM	1
6	BLAND JR	08-06-1954	F	WILLIAM	0
7	BLAND JR	08-06-1954	F	WILLIAM	1
8	BLAND JR	25-10-1953	F	WILLIAM	0
9	BLAND JR	25-10-1953	F	WILLIAM	1
10	SHAFFER JR	25-10-1953	F	WILLIAM	0
11	DUNCAN JR	25-10-1953	F	THOMAS	0
12	CARLSON JR	25-10-1953	F	ROY	0
13	ASBY JR	01-03-1968	F	WILLIAM	0
14	SALTER JR	01-03-1968	F	WILLIAM	0
15	SALTER JR	01-03-1968	F	WILLIAM	1

CHAPTER 7

CONCLUSION

In conclusion, the data duplication removal project utilizing machine learning follows a structured journey. Beginning with requirement analysis, stakeholders' needs are carefully documented, guiding subsequent phases. The design phase maps out the architecture, algorithms, and potential user interfaces, creating a blueprint for implementation. Building the system involves coding, machine learning model training, and integration, transforming the design into a functional solution. Testing is a critical step, encompassing unit, integration, performance, and security testing, ensuring the system's reliability and effectiveness. The optional inclusion of a user interface enhances the user experience, providing a manual validation and feedback mechanism.

Scalability measures accommodate growing datasets, while thorough end-to-end testing evaluates the entire system's functionality. Deploying the system into the production environment marks the culmination of the project, with ongoing monitoring for performance and issue resolution.

Futurework

One drawback of data duplication removal using machine learning is its dependence on training data quality, where biases or errors can lead to poor generalization. Overfitting poses a risk, especially with limited or noisy data. Complex models may incur high computational costs, and their lack of interpretability can be problematic. Additionally, concept drift and privacy concerns in sensitive data environments require ongoing maintenance and careful consideration.

CHAPTER 8

REFERENCES

[1] T. Y. Wu, J. S. Pan, and C. F. Lin (2014), ―Improving accessing efficiency of cloud storage using de-duplication and feedback schemes,‖ IEEE System Journals, vol. 8, no. 1, pp. 208–218, DOI:10.1109/JSYST.2013.2256715.

[2] C. Fan, S. Y. Huang, and W. C. Hsu(2012), ―Hybrid data deduplication in cloud environment,‖ in Processing International Conference Inf. Security Intelligent. Control, pp. 174–177, DOI:10.1109/ISIC.2012.6449734.

[3] J. W. Yuan and S. C. Yu (2013), ―Secure and constant cost public cloud storage auditing with deduplication,‖ in Proc. IEEE International Conference Communication Network Security, pp. 145–153, doi: 10.1109/ CNS.2013. 6682702.

[4] N. Kaaniche and M. Laurent (2014), ―A secure client side deduplication scheme in cloud storage environments,‖ in Proc. 6th Int. Conference .New Technol. Mobility Security, pp. 1–7, DOI: 10.1109/NTMS.2014.6814002.

[5] Z. Yan, W. X. Ding, and H. Q. Zhu (2015), ―A scheme to manage encrypted data storage with deduplication in cloud,‖ in Proc.ICA3PP2015,Theoretical Computer Science and General Issue, Springer, pp. 547–561

[6] https://cdn.guru99.com/images/Big_Data/061114_0759_WhatIsBigDa3.jpg

[7] Z. C. Wen, J. M. Luo, H. J. Chen, J. X. Meng, X. Li, and J. Li (2014), ―A verifiable data deduplication scheme in cloud computing,‖ in Proc. International Conference Intelligent Network Collaborative System, pp. 85–90, DOI: 10.1109/INCoS.2014.111.