**Development of Custom Open Source Intelligent Application**

Gudipalli Sai Sarat Sreeram, Jacob Noel, Nadia Ahmed, Rabia Ahmed & Sai Maneesh Veerulla

Department of Information Technology and Management, Illinois Institute of Technology

ITMS-548: Cyber Security Technologies

Dr. Maurice Dawson Jr.

April 30, 2024

**Development of Custom Open Source Intelligent Application**

1. **Introduction**

The Intelligent Application for Data Analysis is a comprehensive software tool designed to empower users with advanced capabilities for extracting insights from data. Using the power of modern technologies and artificial intelligence, this application offers a user-friendly graphical interface that enables flowless exploration, visualization, machine learning models, and interpretation of complex datasets.

Key Features:

1. Data Retrieval: The application integrates with multiple data sources and APIs, allowing users to fetch data from diverse sources effortlessly.
2. Exploratory Data Analysis (EDA): Through intuitive visualization tools and statistical analysis, users can gain deeper insights into the underlying patterns and trends within their datasets.
3. Data Merging: The application facilitates the integration of multiple datasets, enabling users to combine and analyze data from different sources in a unified manner.
4. Data Visualization: With built-in plotting capabilities, users can create interactive visualizations to communicate their findings effectively.
5. Machine Learning: The application offers machine learning algorithms for predictive modeling and classification tasks, empowering users to build and evaluate models directly within the interface.
6. Customizable Dashboard: Users can create personalized dashboards to monitor key metrics and track performance indicators in real-time.
7. Export and Sharing: The application allows users to export analysis results and visualizations in various formats and share them with colleagues or stakeholders.
8. **Purpose**

The Intelligent Application for Data Analysis aims to democratize data analytics by providing a user-friendly platform that empowers users of all skill levels to extract actionable insights from their data. Whether you're a data scientist, business analyst, or decision-maker, this application equips you with the tools you need to uncover hidden patterns in the dataset, make informed decisions based on the insights , and drive business success.

Benefits

* Efficiency: Streamline your data analysis workflow with intuitive tools and automated processes.
* Accuracy: Leverage advanced algorithms and statistical techniques to ensure the accuracy and reliability of your analyses.
* Empowerment: Democratize data analysis within your organization by enabling users to explore and analyze data independently.
* Innovation: Stay ahead of the curve with cutting-edge features and technologies that drive innovation and growth.

1. **Project Plan: Intelligent Application for Data Analysis**

1. Overview of project objectives, scope, and key features of the Intelligent Application for Data Analysis.

2. Project Setup:

* Define project infrastructure, including setting up development environments, version control systems (GitHub), and communication tools (Slack, Trello).

3. Development Phases:

3.1 Planning Phase:

* Week 1: Define project requirements and user stories.
* Week 2: Set up project infrastructure and development environments.
* Week 3: Create wireframes and mockups for the GUI.

3.2 Implementation Phase:

* Week 4: Develop data retrieval functionality.
* Week 5: Implement exploratory data analysis (EDA) and visualization features.
* Week 6: Design and develop the graphical user interface (GUI) using Tkinter.

3.3 Testing Phase:

* Week 7: Develop test cases and testing scenarios.
* Week 8: Perform unit testing, integration testing, and system testing.
* Week 9: Conduct user acceptance testing (UAT) and gather feedback from stakeholders.

4. Documentation:

* Week 10: Create technical documentation, including user manuals, installation guides, and API documentation.
* Week 11: Document project architecture, design decisions, and coding standards.
* Week 12: Provide tutorials and guides for using the application and interpreting analysis results.

5. Milestones and Timelines:

5.1 Milestone 1: Planning and Setup (Week 1-3)

* Define project requirements and user stories.
* Set up project infrastructure and development environments.
* Create wireframes and mockups for the GUI.

5.2 Milestone 2: Core Functionality Development (Week 4-6)

* Develop data retrieval, EDA, and visualization functionalities.
* Design and develop the GUI interface.

5.3 Milestone 3: Testing and Bug Fixing (Week 7-9)

* Develop and execute test cases for all functionalities.
* Identify and fix bugs or errors through testing.
* Conduct user acceptance testing (UAT) and gather feedback from stakeholders.

5.4 Milestone 4: Documentation and Finalization (Week 10-12)

* Create technical documentation and user manuals.
* Finalize the application and prepare for deployment.

6. Roles and Responsibilities:

* Project Manager: Overall coordination and management of the project.
* Software Developers: Responsible for developing and implementing the core functionalities of the application.
* Quality Assurance/Testers: Responsible for testing the application and ensuring its quality and reliability.
* Technical Writers: Responsible for creating technical documentation and user manuals.

7. Communication Plan:

* Weekly status meetings to discuss progress, challenges, and next steps.
* Use of project management tools (e.g., Trello, Jira) for task tracking and collaboration.
* Open communication channels for team members to raise concerns or ask for assistance.

8. Risk Management Plan:

* Identify the potential risks and uncertainties that may impact the project.
* Develop strategies to mitigate, monitor, and respond to these risks throughout the project lifecycle.
* Regularly review and update the risk register to address new risks as they arise.

9. Earned Value Management:

* Set up an earned value management system to track project progress, budget, and performance indicators.
* Calculate earned value metrics such as Planned Value (PV), Earned Value (EV), Actual Cost (AC), Cost Variance (CV), and Schedule Variance (SV).
* Use these metrics to assess project performance and make data-driven decisions.

1. **Risk Management Plan: Intelligent Application for Data Analysis**

1. The risk management plan outlines strategies to identify, assess, and mitigate potential risks that may later affect the successful completion of the Intelligent Application for Data Analysis project.

2. Risk Identification:

|  |  |
| --- | --- |
| Risk Category | Potential Risks |
| Technical Risks | Complexity in integrating multiple data sources and APIs |
|  | Compatibility issues across different operating systems |
| Resource Risks | Limited availability of skilled developers and domain experts |
|  | Resource constraints or competing priorities |
| Schedule Risks | Unrealistic timelines leading to project delays |
|  | Dependencies on external factors such as data availability |
| Quality Risks | Inadequate testing leading to the release of a buggy app |
|  | Lack of thorough documentation affecting usability |
| Security Risks | Vulnerabilities in data handling and storage |
|  | Insufficient security measures leading to unauthorized access |

3. Risk Mitigation Strategies:

* Technical Risks:
  + Conduct extensive research and prototyping during the planning phase to identify and address technical challenges.
  + Implement a modular and scalable architecture to facilitate integration and maintenance.
* Resource Risks:
  + Ensure adequate staffing and allocate resources effectively to prevent overallocation or burnout.
  + Identify backup resources or external expertise to mitigate skill gaps or resource shortages.
* Schedule Risks:
  + Break down the project into smaller pieces of manageable tasks with realistic timelines and milestones.
  + Regularly monitor project progress and adjust schedules as needed to accommodate changes.
* Quality Risks:
  + Implement a comprehensive testing strategy, including unit testing strategies, integration testing methods, and user acceptance testing.
  + Establish quality assurance processes and standards to ensure adherence to best practices.
* Security Risks:
  + Implement a diverse security measures, including data encryption, access controls, and secure authentication mechanisms.
  + Carry out regular audit plans and update the security protocols to address emerging threats and vulnerabilities.

4. Risk Monitoring and Control:

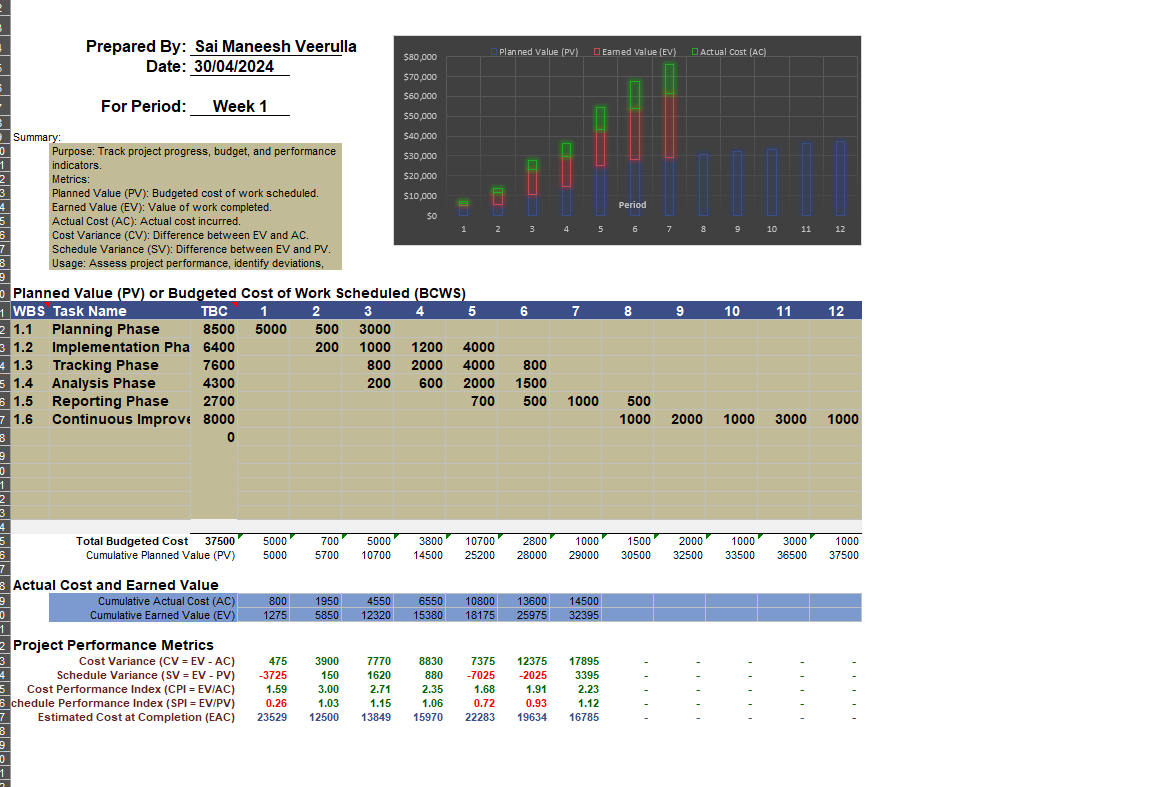
* Carry out regular risk review meetings to assess the status of identified risks and monitor new risks.
* Ensure to maintain a risk register to document that will be used in identification of risks, their known potential impacts, and mitigation that can be strategized.
* Assign ownership for each risk and establish clear escalation paths for addressing high-priority risks.
* Implement proactive measures to mitigate risks before they escalate, and monitor key risk indicators to identify early warning signs.

5. Risk Response Planning:

* Implement a working contingency plans and fallback options for the high-impact risks and the risks that cannot be fully mitigated or eliminated.
* Communicate risk response plans to relevant stakeholders and ensure alignment on risk management strategies.
* Perform a regular review and regular update risk response plans based on changes in project dynamics and external factors.

1. **Earned Value Sheet: Intelligent Application for Data Analysis**

The earned value sheet serves as a tool to track project progress, budget, and performance indicators for the Intelligent Application for Data Analysis project.



1. **Project Management Plan: Intelligent Application for Data Analysis**

1. The project management plan outlines the strategies and guidelines for effectively managing the Intelligent Application for Data Analysis project. It encompasses communication strategies, team collaboration tools, and workflow guidelines to ensure successful project execution.

2. Communication Strategies:

2.1 Stakeholder Communication:

* Regular project updates and progress reports will be shared with stakeholders through weekly status meetings, email updates, and quarterly reviews.
* Stakeholder feedback will be actively solicited and incorporated into project decisions and planning.

2.2 Team Communication:

* A dedicated communication platform, such as Slack or Microsoft Teams, will be used for day-to-day team communication, including discussions, announcements, and document sharing.
* Weekly team meetings will be conducted to review progress, address challenges, and align on upcoming tasks and priorities.

3. Team Collaboration Tools:

3.1 Project Management Tool:

* Trello or Asana will be used as the project management tool to track tasks, assign responsibilities, and monitor project progress.
* Tasks will be organized into boards or projects, with clear deadlines, priorities, and dependencies defined.

3.2 Version Control System:

* GitHub will serve as the version control system for managing code repositories, documenting changes, and facilitating collaborative development among team members.
* Branching strategies and pull request workflows will be established to ensure code quality and minimize conflicts.

4. Workflow Guidelines:

4.1 Development Workflow:

* The development process will follow an agile methodology, with iterative development cycles, frequent testing, and continuous integration and deployment (CI/CD) practices.
* User stories and feature requirements will be documented in a backlog, prioritized, and scheduled for implementation in sprints.

4.2 Testing and Quality Assurance:

* Automated testing frameworks, such as pytest or Selenium, will be used to automate test cases and ensure code reliability and quality.
* Code reviews and peer evaluations will be conducted regularly to identify bugs, review code quality, and provide constructive feedback.

4.3 Documentation Guidelines:

* Comprehensive documentation will be maintained for all aspects of the project, including system architecture, codebase, APIs, and user manuals.
* Documentation will be version-controlled and accessible to all team members to facilitate knowledge sharing and onboarding.

1. **Intelligent Application with GUI: Develop a fully custom open source Intelligent Application**

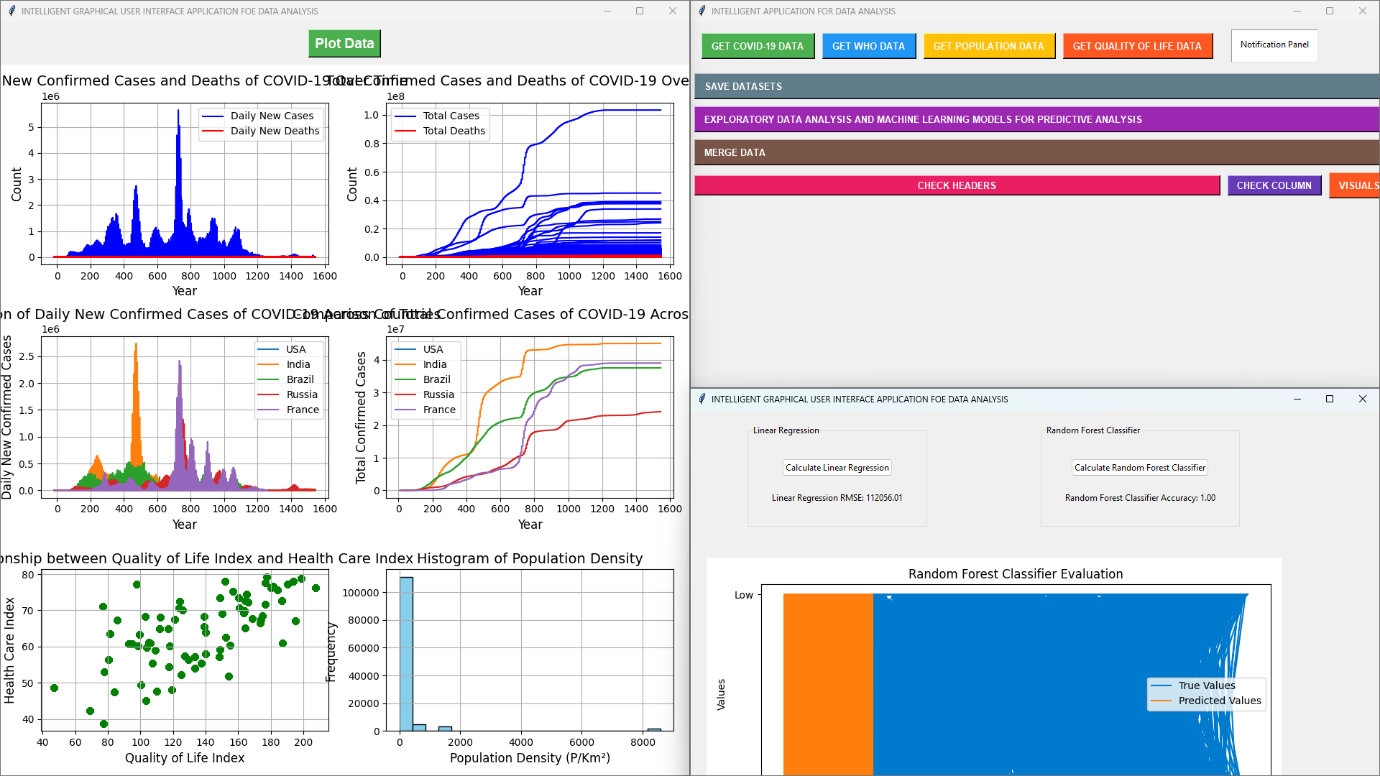


Figure 1. Intelligent Application with GUI

GitHub link - <https://github.com/ManeeshVeerulla/ITMS-548-Final-Project/tree/master>

1. **Conclusion**

The project provides a framework for effective communication, collaboration, and workflow management throughout the Intelligent Application for Data Analysis project. By implementing these strategies and guidelines, the project team aims to achieve project objectives efficiently and deliver a high-quality product.

1. **References**

Custom and Open Source Code: A New Approach to Application Security Management (2016)

<https://securityintelligence.com/custom-and-open-source-code-a-new-approach-to-application-security-management/>

Data Analytics Project Plan Template to Help You Get Started (2024)

<https://www.projectpro.io/article/data-analytics-project-plan-template/613>

Earned Value Management (EVM) Template (2021)

<https://www.vertex42.com/ExcelTemplates/earned-value-management.html>

Earned Value Management Defined: Formulas & Examples

<https://thedigitalprojectmanager.com/projects/pm-methodology/earned-value-management/>

PMP Risk Management Part IV – Benefits of Project Planning (2016)

<https://www.simplilearn.com/pmp-risk-management-part-iv-benefits-of-project-planning-rar237-article>

The Champion’s Guide to Earned Value Management with Templates and Insider Tips (2021)

<https://www.smartsheet.com/guide-to-earned-value-management-templates-and-tips>