**Phase-1**

1. ***Define Problem statements***

*Questions:-*

1.Print 5 sample rows of the dataset for the understanding of it.

2.Understand the information of the dataset and print the same.

3.What are the statistics of the dataset, how do calculate it and print the same.

4.Describe statistics of fatality\_count column, and print the mean, median, mode, standard deviation of the same.

5.Plot a Histogram for the injury\_count column.

6.Explore the correlation between fatality\_count and injury\_count.

7.Create a pie chart for the injury\_count column with the percentage.

8.Plot a scatter plot to detect outliers in the dataset.

9.Plot a qq plot for the column gazeteer\_distancefrom.

10.Create a heat map for any one of the columns from the dataset.

***2.Create project plan and product backlog***

Creating a project plan and product backlog for a Python-based analysis project on the landslide dataset involves defining the steps and tasks required to accomplish your goals. Below is a simplified project plan and product backlog:

**Project Plan:**

**1. Project Initiation**

* Define Project Objectives: Clearly articulate the goals and objectives of the analysis.
* Identify Stakeholders: Determine who will be involved and impacted by the project.
* Set Project Timeline: Establish a project timeline with milestones and deadlines.
* Define Communication Channels: Determine how team members will communicate and collaborate.

**2. Data Acquisition and Preprocessing**

* Collect Landslide Dataset: Acquire the landslide dataset from reliable sources.
* Data Cleaning: Clean the dataset by addressing missing values, duplicates, and outliers.
* Data Exploration: Explore the dataset to understand its structure and characteristics.
* Data Transformation: Prepare the dataset for analysis, including feature engineering if necessary.

**3. Data Analysis in Python**

* Define Data Analysis Goals: Determine the specific analysis objectives (e.g., risk assessment, identifying factors).
* Descriptive Analysis: Calculate basic statistics and visualizations to summarize the dataset.
* Identify Key Features: Determine which features or variables are relevant to the analysis.
* Statistical Analysis: Use Python libraries to conduct statistical tests and analyses.

**4. Data Visualization and Reporting**

* Create Data Visualizations: Use Python libraries like Matplotlib, Seaborn, or plotly to visualize the data.
* Generate Reports: Create reports or presentations summarizing key findings.
* Identify Patterns and Trends: Interpret the visualizations and analysis results to identify patterns and trends in the data.

**5. Model Building (if applicable)**

* Define Machine Learning Objectives: If applicable, specify the objectives for building machine learning models (e.g., predictive modeling).
* Feature Selection/Engineering: Choose relevant features or engineer new ones for model building.
* Model Selection and Training: Select and train machine learning models using libraries like Scikit-Learn.
* Model Evaluation and Optimization: Evaluate model performance and optimize parameters.

**6. Product Deployment (if applicable)**

* Create Interactive Dashboards: Develop interactive dashboards using Python web frameworks like Dash or Flask.
* Deploy Models: If machine learning models are used, deploy them for real-time predictions using Python-based tools.
* Implement Data Monitoring: Set up data monitoring to track changes and updates to the dataset.

**7. Documentation**

* Document the Process: Create detailed documentation of the data analysis process and code.
* User Guides: Develop user guides for stakeholders who wish to understand and utilize the results.
* Share Documentation: Share documentation with the project team and stakeholders.

**8. Testing and Validation**

* Verify Analysis Results: Test and verify the accuracy and validity of analysis results.
* Validate Models: If machine learning models are used, validate them through appropriate validation techniques.
* Address Issues and Concerns: Resolve any issues or concerns raised during the testing phase.

**9. Project Closure**

* Present Results: Present the analysis results to stakeholders.
* Gather Feedback: Collect feedback and address any final concerns or questions.
* Prepare Final Report: Create a final project report summarizing the analysis and its implications.
* Conclude the Project: Wrap up the project and finalize all project-related tasks.

**Product Backlog:**

The product backlog is a dynamic list of tasks, and it will evolve as the project progresses. Below are some sample backlog items:

1. Collect landslide dataset from reliable sources.
2. Clean the dataset to handle missing values and inconsistencies.
3. Explore the dataset to understand its structure and contents.
4. Define specific analysis goals, such as risk assessment or factor identification.
5. Perform descriptive analysis using Python, including summary statistics and visualizations.
6. Identify key features and variables for analysis.
7. Use Python for statistical analysis to uncover insights.
8. Create data visualizations using Python libraries.
9. Generate reports and summaries of key findings.
10. Prepare Python code for modeling (if applicable).
11. Train and evaluate machine learning models (if required).
12. Develop interactive dashboards using Python web frameworks.
13. Deploy models for real-time predictions (if applicable).
14. Document the analysis process and code.
15. Create user guides for stakeholders.
16. Test and validate the analysis results.
17. Address any issues or concerns raised during testing.
18. Prepare the final project report and presentation.
19. Conduct project closure activities.

These tasks should be prioritized based on project requirements and dependencies, and new items can be added as the project evolves. It's essential to manage the backlog and regularly update it as tasks are completed or new requirements emerge.

* 1. Creation of Git Repository
* **Open your terminal or command prompt.**
* **Navigate to the directory where you want to create your Git repository.**

1.Use the **cd** command to change to the desired directory.

* **Initialize a Git repository:**

2.Run the command **git init** in your chosen directory.

* **Add your project files to the repository:**
  1. Place your project files in the directory.
* **Stage your files:**

4.Use **git add .** to stage all files or specify individual files to stage.

* **Commit your changes:**

5.Run **git commit -m "Initial commit"** to save the changes with a commit message.

You've created a Git repository, added your files, and made your first commit. You can now continue to work on your project, track changes, and collaborate with others using Git.

***This is our real git hub repository link:-***

***https://github.com/meherbanu/mini-project1.git***