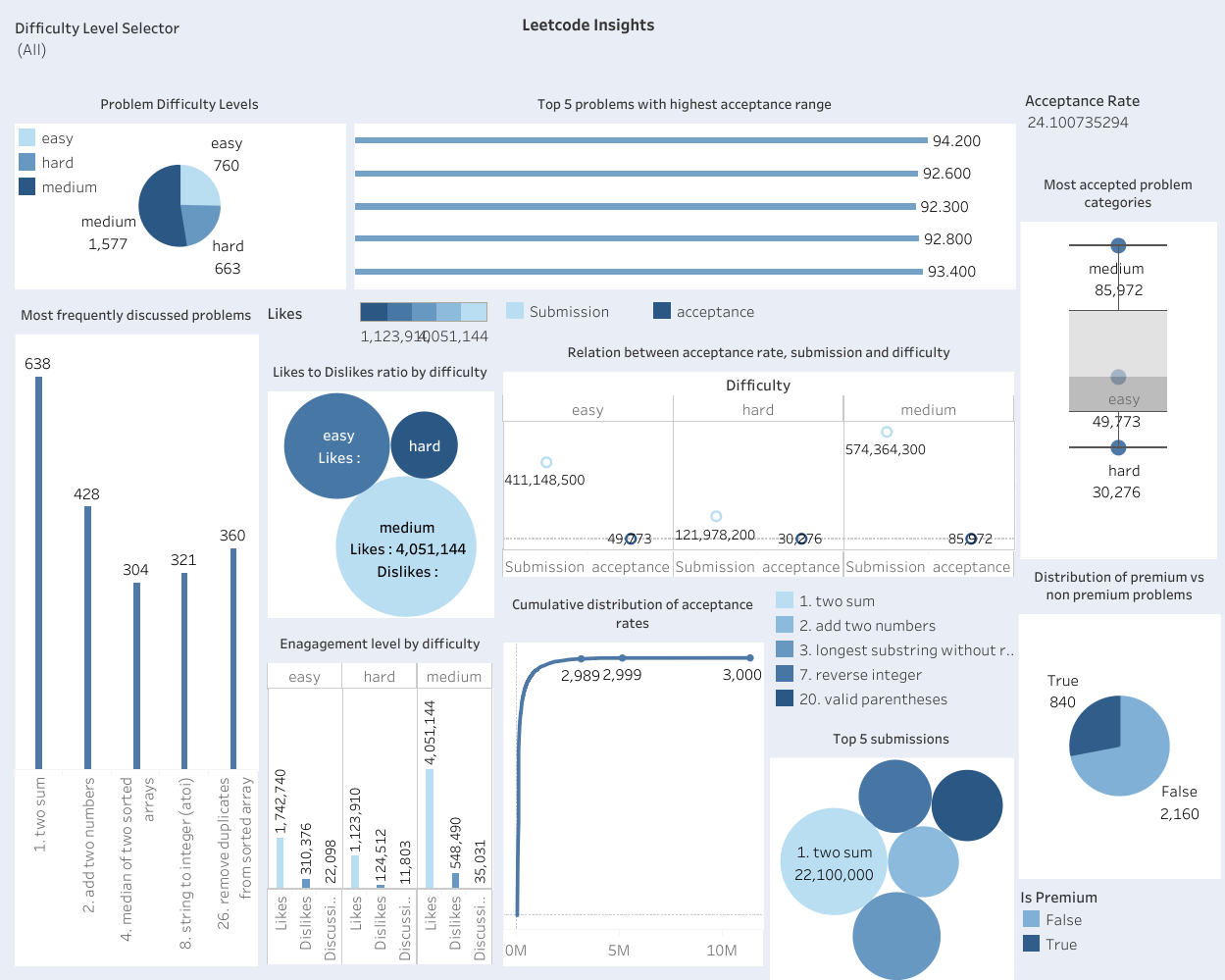
**Phase III : Dashboard Implementation**

**Section 1: The Dashboard**



The LeetCode Insights Dashboard is a multifaceted tool designed to offer invaluable insights into coding performance and trends on the LeetCode platform. With a diverse range of visualizations and filters, It's packed with different tools to help recruiters, teachers, coders, data analysts, software developers, platform managers, researchers, schools, and interview coaches. It gives them all kinds of ways to look at data like graphs and options to filter information. With this dashboard, they can check how good candidates are at coding, keep an eye on students' progress, see which coding topics are trending, find interesting facts, and see how users are interacting with the platform. It's basically a must-have for anyone wanting to get better at coding or teaching it.

**Section 2: The Dataset**

The dataset appears to be a collection of challenges from LeetCode, a popular tool for coding practice, particularly among those preparing for software engineering interviews. Below is a thorough description of each attribute identified in the dataset:

* **id**: Each problem in the dataset is uniquely identified by an ID, which is a numerical value. This ID serves as a reference point, ensuring that each problem can be distinguished from others, facilitating organization and management.

Data Type: Ratio

Domain: Positive integers (e.g., 1, 2, 3, ...)

* **page\_number**: This attribute indicates the page number where a problem is located within the dataset or on the LeetCode website. It aids in pagination, allowing users to navigate through the list of problems systematically or locate them on the website with ease.

Data Type: Ratio

Domain: Positive integers indicating the page number (e.g., 1, 2, 3, ...)

* **is\_premium**: This field is a boolean value (True/False) indicating whether a problem is exclusively available to premium members of LeetCode. Premium problems require a subscription for access, thus helping users identify which problems they can attempt without a subscription.

Data Type: Categorical (Binary)

Domain: {True, False}

* **title**: Each problem is accompanied by a title, which typically includes a numerical identifier and a concise description of the challenge. The title provides a quick overview of the problem's theme or focus, assisting users in deciding if it aligns with their interests or capabilities.

Data Type: Categorical (Nominal)

Domain: Set of strings, each representing the title of a LeetCode problem (e.g., "1. Two Sum", "2. Add Two Numbers", ...)

* **problem\_description**: This attribute furnishes a comprehensive description of the coding problem, encompassing the scenario, requirements, and constraints. It is essential for understanding the problem's objectives, input/output specifications, and any specific conditions that must be met for a solution to be deemed correct.

Data Type: Categorical (Nominal)

Domain: Set of strings, with each string providing a detailed problem statement

* **topic\_tags**: These are like labels that group the problem into different areas of computer science, such as 'Array', 'String', or 'Dynamic Programming'. They help users find problems related to specific concepts or algorithms, making it easier to focus on areas they want to practice or improve.

Data Type: Categorical (Nominal)

Domain: Set of strings, each listing the topic tags associated with a problem (e.g., "'Array', 'Hash Table'", "'Linked List', 'Math', 'Recursion'", ...)

* **difficulty**: It shows how hard or easy the problem is, usually labeled as Easy, Medium, or Hard. This helps users choose problems that match their skill level or push themselves to tackle more challenging ones.

Data Type: Ordinal

Domain: {Easy, Medium, Hard}

* **similar\_questions**: This lists other problems that are similar to the current one, either in concept or difficulty. It's handy for users who want to practice more problems like the one they just solved, to strengthen their skills.

Data Type: Categorical (Nominal)

Domain: Set of strings, with each string listing similar questions to the current problem

* **no\_similar\_questions**: This number tells you how many other similar questions are available. It gives users an idea of how many more practice problems they can find related to a particular topic.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...)

* **acceptance**: The acceptance rate shows the percentage of successful submissions out of all submissions for this problem. It gives an idea of how difficult the problem is for the community.

Data Type: Ratio

Domain: Real numbers between 0 and 100 (inclusive), representing a percentage

* **accepted**: This tells you how many times the problem has been successfully solved and accepted by users.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...), representing the count of accepted submissions

* **submission**: The total number of attempts made to solve the problem, whether successful or not. It indicates how popular or engaging the problem is among users.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...), representing the total number of submissions

* **solution**: It shows if there are official solutions or discussions available to help understand the problem's solution. This could be a number indicating the available solutions.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...), indicating the number of solutions or solution articles available

* **discussion\_count**: This tells you how many discussions are available for the problem, including hints, solutions, and general advice from users.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...), indicating the number of discussion threads available

* **likes**: The total number of likes the problem has received from users, indicating its popularity or the quality of the problem statement.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...), representing the total number of likes a problem has received

* **dislikes**: The total number of dislikes, which can highlight issues with the problem statement, its difficulty level, or user dissatisfaction.

Data Type: Ratio

Domain: Non-negative integers (0, 1, 2, ...), representing the total number of dislikes a problem has received

* **problem\_URL**: A direct link to the problem's page on the LeetCode website, where users can read the problem statement, submit solutions, and join discussions.

Data Type: Categorical (Nominal)

Domain: Set of URLs, each directing to the LeetCode page of a specific problem

* **solution\_URL**: Similar to the problem URL, this links directly to the official solution or a page with solutions and discussions about the problem, helping users learn and understand.

Data Type: Categorical (Nominal)

Domain: Set of URLs, each directing to the solution page on LeetCode for a specific problem

**Preprocessing:**

* In the pre-processing phase, the dataset underwent several essential steps to ensure its quality and suitability for analysis:
* Handling Missing Values:

Missing values were identified in key columns such as 'difficulty', 'acceptance', 'likes', 'dislikes', and 'discussion\_count'.To maintain data integrity, missing values were handled through imputation, where feasible, based on the significance of the missing values and their impact on analysis. For instance, missing values in 'difficulty' were imputed with the mode value of the difficulty level, while missing acceptance rates were imputed with the median value.

* Handling Outliers:

Outliers in numerical columns such as 'acceptance', 'accepted', 'submission', 'likes', 'dislikes', 'discussion\_count', and 'no\_similar\_questions' were identified.

To mitigate their impact on analysis, appropriate methods were employed to address outliers. For instance, outliers in 'likes' and 'dislikes' were winsorized, while outliers in 'discussion\_count' were capped at a certain percentile to prevent skewing of results.

* Normalization and Binning:

Numerical metrics like 'likes', 'dislikes', and 'acceptance' were normalized to ensure comparability and prevent bias in analysis.

Continuous data such as 'acceptance rate' were converted into categorical bins ('Low', 'Medium', 'High') using a collaborative binning strategy. This facilitated specific visualizations and analysis by establishing meaningful categories for acceptance rates.

* Data Consistency and Accuracy:

Consistency was maintained by uniformly applying data transformations and cleaning steps throughout the dataset. Data validation procedures were conducted to ensure adherence to expected formats, domain ranges, and data types. Any inconsistencies or errors were rectified to enhance accuracy and reliability.

**Section 3: Dashboard Users**

**Recruiters:** Recruiters and HR professionals have the opportunity to utilize the dashboard in order to assess the coding skills of candidates. This can be achieved by analyzing their performance on LeetCode problems. By doing so, recruiters can easily identify the top performers and evaluate their problem-solving abilities. Additionally, the dashboard allows them to track the trends in coding proficiency over a period of time.

**Educators:** Educators and trainers have the ability to utilize the dashboard to track the progress of their students and pinpoint any areas that may require additional support or instruction. Furthermore, they can evaluate the efficiency of their teaching methods and curricula by analyzing the performance data of their students.

**Programmers:** Competitive programmers or enthusiasts can utilize the dashboard to effectively monitor their rankings, identify patterns in problem-solving, and pinpoint areas for enhancing their skills. By analyzing trends in competition participation, success rates in various contest categories, and performance relative to their peers, they can gain valuable insights.

**Data analysts:** Data analysts and researchers have the opportunity to delve into the dataset and discover valuable insights regarding coding trends, popular problem topics, and the relationships between various factors like submission frequency and problem difficulty. By utilizing the dashboard, they can effectively visualize patterns in user behavior, submission outcomes, and the progression of problem difficulty over time.

**Software Engineers:** Software engineers and developers have the ability to utilize the dashboard to monitor their individual advancement on LeetCode. This includes visualizing various metrics like the number of problems solved over time, the distribution of difficulty levels, the accuracy rate, and their performance in specific problem categories such as algorithms and data structures. Additionally, they can compare their progress with their peers or industry benchmarks, enabling them to gain valuable insights.

**LeetCode Platform Administrators:**The dashboard provided by the LeetCode platform enables administrators to effectively monitor user engagement, track platform usage metrics, and pinpoint areas that require improvement. By analyzing user activity patterns, administrators can identify popular features and evaluate the impact of updates or changes made to the platform.

**Academic Scholars:** Academic scholars specializing in computer science education and learning analytics have the opportunity to utilize the dashboard for investigating problem-solving techniques, learning patterns, and the influence of online platforms such as LeetCode on the enhancement of programming skills.

**Educational Institutions:** Educational establishments providing computer science courses have the ability to utilize the dashboard for assessing the efficiency of their curriculum, pinpointing areas that may need additional focus, and monitoring the progress of students in coding-related classes.

**Technical Interview Coaches:** Technical interview coaches have the ability to utilize the dashboard to assess their clients' performance on LeetCode problems, pinpoint areas of strength and areas for improvement, and customize coaching sessions to meet individual needs.

**Section 4: Questions**

1. What is the distribution of problem difficulty levels among the LeetCode problems available in the dataset?
2. What are the most frequently discussed problems, and what insights can we gain from these discussions?
3. Which LeetCode problems have the highest acceptance rates?
4. How does the acceptance rate vary across different problem difficulty levels?
5. What is the likes to dislikes ratio by problem difficulty?
6. How does the engagement level (likes, dislikes, discussion counts) vary across different problem difficulty levels?
7. What is the cumulative distribution of problem acceptance?
8. What is the problem difficulty to engagement metrics correlation?
9. What are the most challenging problem categories based on submission frequency and acceptance rates?
10. What is the relationship between problem difficulty and number of submissions and acceptance rate?
11. What is the distribution of premium versus non-premium problems within the dataset?
12. Which problems have the highest number of submissions in the dataset?

**Section 5: Plots**

**1. Distribution of Problem Difficulty Levels**

* **Question Addressed:** What is the distribution of problem difficulty levels among the LeetCode problems available in the dataset?
* **How it Addresses It:** This pie chart visually represents the distribution of problem difficulties by categorizing them into Easy, Medium, and Hard levels. It allows stakeholders to easily grasp the proportion of each difficulty category within the dataset.

**2. Most Frequently Discussed Problems**

* **Question Addressed:** What are the most frequently discussed problems, and what insights can we gain from these discussions?
* **How it Addresses It:** The bar chart orders problems based on the volume of discussions they generate, providing insights into which problems attract the most attention and potential areas of interest or difficulty for users.

**3. Problems with the Highest Acceptance Rates**

* **Question Addressed:** Which LeetCode problems have the highest acceptance rates?
* **How it Addresses It:** This horizontal bar chart directly showcases the problems with the highest acceptance rates, offering a clear indication of which problems are solved successfully most often.

**4. Relationship Between Problem Difficulty, Number of Submissions, and Acceptance Rate**

* **Question Addressed:** What is the relationship between problem difficulty and number of submissions and acceptance rate?
* **How it Addresses It:** This chart maps problem complexity against approval rates and submission counts, allowing stakeholders to visualize the relationship between problem difficulty, acceptance rates, and popularity.

**5. Likes to Dislikes Ratio by Problem Difficulty**

* **Question Addressed:** What is the likes to dislikes ratio by problem difficulty?
* **How it Addresses It:** This Bubble chart visualizes the likes to dislikes ratio for each problem difficulty level, providing a clear comparison of user sentiment across different difficulty categories.

**6. Engagement Level Variation by Difficulty**

* **Question Addressed:** How does the engagement level (likes, dislikes, discussion counts) vary across different problem difficulty levels?
* **How it Addresses It:** By presenting mean values of engagement metrics (likes, dislikes, discussion counts) for each difficulty level in a grouped bar chart, it allows stakeholders to discern how user engagement varies across different levels of problem difficulty**.**

**7. Cumulative Distribution of Problem Acceptance**

* **Question Addressed:** What is the cumulative distribution of problem acceptance?
* **How it Addresses It:** The line graph depicting the Cumulative Distribution Function (CDF) of problem acceptance rates offers a clear depiction of how acceptance rates are distributed across different problems, aiding in understanding the overall acceptance rate distribution.

**8. Top 5 Submissions**

* **Question Addressed:** Which problems have the highest number of submissions in the dataset?
* **How it Addresses It:** This bubble chart visualizes the number of submissions for each problem title, highlighting the most attempted problems and their popularity or difficulty level based on submission volume.

**9. Distribution of Premium vs Non-Premium Problems**

* **Question Addressed:** What is the distribution of premium versus non-premium problems within the dataset?
* **How it Addresses It:** The pie chart visually compares the proportion of premium and non-premium problems, offering insights into the availability of premium content within the dataset.

**10. Most Challenging Problem Categories**

* **Question Addressed:** What are the most challenging problem categories based on submission frequency and acceptance rates?
* **How it Addresses It:** This box plot illustrates the distribution of acceptance rates for different problem difficulty categories, providing insights into which categories pose the greatest challenge based on both submission frequency and acceptance rates.

**Section 6:**

1. **Difficulty Level Filter:**

**Purpose**: To sort information in different visualizations based on the level of difficulty of the problem (Easy, Medium, Hard).

**Connected plots**: This filter will impact all visualizations that include problem difficulty as a factor, including:

Distribution of Problem Difficulty Levels

Most Frequently Discussed Problems

Problems with the Highest Acceptance Rates

Relationship Between Problem Difficulty, Number of Submissions, and Acceptance Rate

Likes to Dislikes Ratio by Problem Difficulty

Engagement Level Variation by Difficulty

Cumulative Distribution of Problem Acceptance

Top 5 Submissions

Distribution of Premium vs Non-Premium Problems

Most Challenging Problem Categories

**Value Range**: The available options will be ['Easy', 'Medium', 'Hard'], directly retrieved from the 'difficulty' attribute in the data.

1. **Acceptance Rate Slider**

**Purpose**: Modifying visualizations according to various acceptance rates enables users to analyze the impact of different levels of problem acceptance on metrics.

**Connected Plots:** This slider will impact charts illustrating acceptance rates, including:

Relation between acceptance rate, submission and difficulty,

Most accepted problem categories

**Value Range**: 0% to 100%, allowing users to set minimum and maximum acceptance rate thresholds.

**Dashboard Link:**[**https://public.tableau.com/views/visualisations-backup/Dashboard1?:language=en-GB&:sid=&:display\_count=n&:origin=viz\_share\_link**](https://public.tableau.com/views/visualisations-backup/Dashboard1?:language=en-GB&:sid=&:display_count=n&:origin=viz_share_link)