# Introduction to the Problem

## Background

[To be completed]

# Overall Directions to a Solution

## Dataset

[[Yelp](https://business.yelp.com/data/resources/open-dataset/) dataset](https://business.yelp.com/data/resources/open-dataset/) is used in this research. Yelp is a platform for users to find, review, and recommend businesses in different categories (e.g. food, shopping, entertainment, etc.) across 8 metropolitan areas in the USA and Canada. This dataset was originally used for the Yelp Challenge, and it is open for academic research.

## Problems

Below is the list of problems we identified and the step-by-step strategies to address them. Some problems are prioritized higher due to their significant impact on recommendation performance. For lower-priority issues, detailed strategies may not yet be developed.

### Data Sparsity (High Priority)

**Definition**: Data sparsity refers to the lack of sufficient user-item interaction data, which hampers the system's ability to generate accurate recommendations.

**Impact**: Collaborative Filtering models, which heavily rely on user interactions, often struggle with sparse data, leading to less reliable recommendations and limited personalization.

**Methodologies**: *Comparative Evaluation*, *User Acceptance Test (UAT)*

**Solution/Strategy Flow**:

|  |  |
| --- | --- |
| Problem | Demonstrate that the dataset exhibits sparsity, a common issue in content with limited interactions (e.g., reviews without click-rate or like conversion data). |
| Provide statistics on items/users with minimal interactions (e.g., number of users with fewer than 5 reviews). |
| (Optional) Show the limitations of Collaborative Filtering through Leave-One-Out Cross-Validation or metrics like *Hit Rate* and *Top-K Evaluation*. |
| Solution | Illustrate how models like DSSM mitigate the sparsity issue by leveraging additional data or embeddings. |
| Highlight improvements in performance metrics (e.g., accuracy, F1-score) after parameter tuning or using alternative loss functions. |
| Demonstrate how the overall recommendation system improves despite sparse data. |

### Cold-Start Problem (High Priority)

**Definition**: The cold-start problem occurs when new users or items lack interaction history, making it challenging to provide personalized recommendations.

**Impact**: New users receive generic suggestions, and new items struggle to gain visibility, reducing user engagement and satisfaction.

**Methodologies**: *Comparative Evaluation*, *User Acceptance Test (UAT), Case Studies and Scenario Simulation (Optional)*

**Solution/Strategy Flow**:

|  |  |
| --- | --- |
| Problem | (Optional) Relate to data sparsity statistics (e.g., users/items with zero interactions). |
| Describe scenarios illustrating the user journey for new users or the introduction of new items. |
| Solution | Detail how solutions like assigning initial labels/interests or leveraging demographic data can improve recommendations for new users. |
| Showcase the user journey improvements with tailored recommendations. |
| (Optional) Provide updated statistics demonstrating increased engagement for new users/items (same set). |

### Long Tail Items (High Priority)

**Definition**: Long-tail items are less popular items with fewer interactions, often overlooked by recommendation systems.

**Impact**: Ignoring long-tail items reduces content diversity and user satisfaction, while perpetuating the Pareto Principle (popular items dominate recommendations).

**Methodologies**: *Comparative Evaluation*, *Case Studies and Scenario Simulation (Optional)*

**Solution/Strategy Flow**: describes what is used to boost unpopular items and how the performance is.

|  |  |
| --- | --- |
| Problem | Explain the Pareto Principle and how it affects content recommendation. |
| Highlight challenges in training models with long-tail items, such as biased negative sampling in DSSM or Self-Supervised Learning. |
| Provide statistics showing the low exposure of long-tail items. |
| Solution | Outline strategies to promote long-tail items, such as tuning model parameters, using balanced sampling methods, or adding filtering layers. |
| Present updated exposure statistics to demonstrate improved visibility for long-tail items. |

### Scalability (Medium Priority)

**Definition**: Scalability refers to the system's ability to handle a growing number of users and items efficiently without significant performance degradation.

**Impact**: As the platform expands, computational demands increase, potentially leading to slower response times and reduced performance.

**Methodologies**: *Comparative Evaluation*

**Strategy Flow**:

* Evaluate system response times and performance metrics as data volume increases.
* Incorporate scalability-focused techniques like caching, approximate nearest neighbor search, or distributed training. Compare system performance before and after scaling (Can use synthetic datasets to simulate the situation).

### Diversity vs. Relevance Trade-off (Low Priority)

**Definition**: Balancing diversity in recommendations with relevance to user preferences.

**Impact**: Overemphasizing relevance leads to narrow recommendations, while focusing on diversity may reduce personalization.

**Solution/Strategy Flow**:

* Explore strategies to balance diversity and relevance, such as multi-objective optimization.
* Measure trade-offs using metrics like intra-list diversity and relevance scores.

### Privacy Concerns (Low Priority)

**Definition**: Ensuring user data is handled responsibly to protect privacy while delivering personalized recommendations.

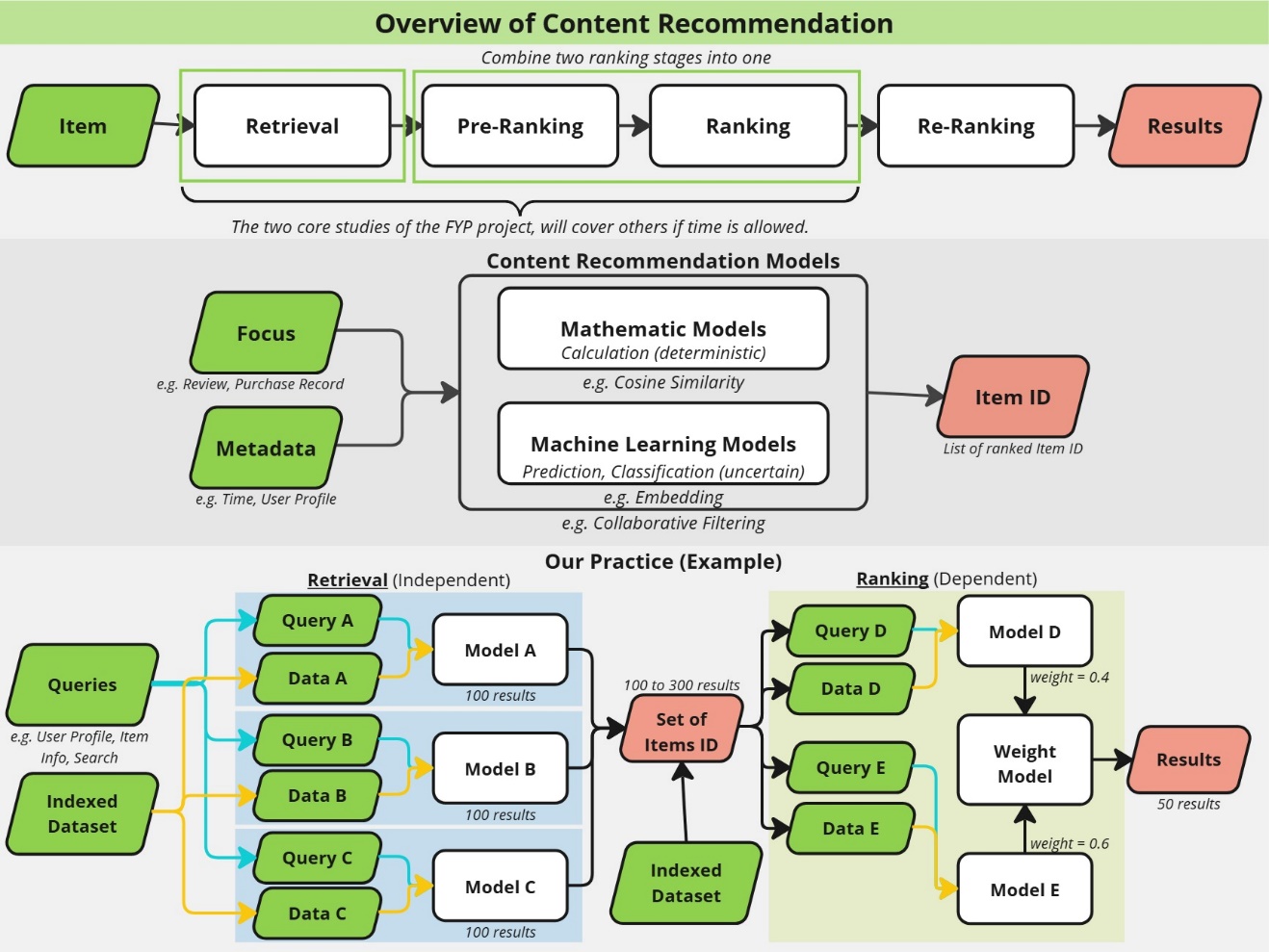
**Impact**: Mismanagement of personal data can lead to privacy breaches, loss of trust, and regulatory non-compliance.

### Evaluation Challenges (Low Priority)

**Definition**: Difficulties in assessing recommendation algorithms accurately, particularly with limited ground truth or real-time feedback data.

**Impact**: Inadequate evaluation metrics may lead to deploying suboptimal models, reducing user satisfaction.

# Design of the Solution



### Dataset

To shorten the training time, we down-sample the dataset in training as below:

A diagram of data sampling

AI-generated content may be incorrect.

You can refer to Appendix#1 for the detail of dataset property.

# Testing of the Solution

