Advanced Machine Learning: Data Science and ML Refresher

Shubhankar Agrawal

Abstract

This document serves as a quick refresher for Data Science and Machine Learning interviews. It covers mathematical and technical concepts across a range of algorithms. This requires the reader to have a foundational level knowledge with tertiary education in the field. This PDF contains material for revision over key concepts that are tested in interviews.

Contents

1. Causal Inference

Building models for causality. Widespread usage of OLS models.

1.1. Key Concepts

1.1.1. Variables

Treatment [T]: Change applied to check for causality

Outcome [Y]: Result of experiment

Confounder: Variable that affects treatment and outcome

Covariates [X]: Other features that lead to outcome

Instrument Variable (IV) [Z]: Influences only Treatment

Mediator: Variable through which, treatment effects on outcome

Propensity [e]: Likelihood of receiving treatment given confounders

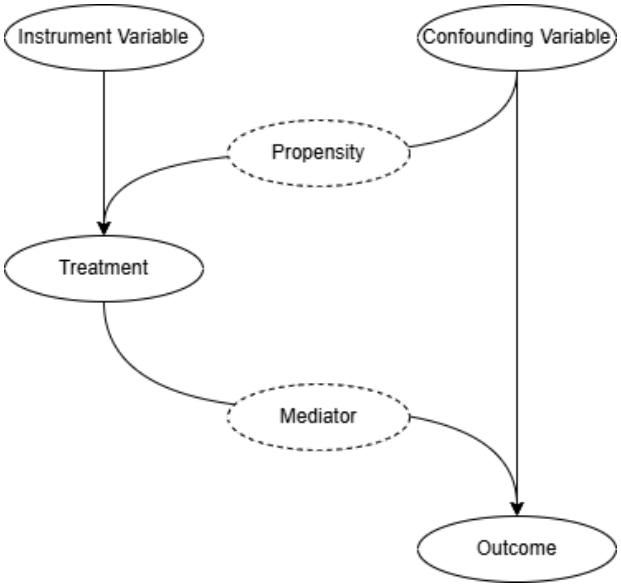

Figure 1. Causal Directed Acyclic Graph (DAG)

SUTVA: Stable Unit Treatment Value Assumption
ATET: Average Treatment Effect on Treated
Population is divided into 4 sections

Table 1. Population Behaviour

Type	Description
Compliers	Take Treatment if assigned
Always Takers	Always take Treatment
Never Takers	Never take Treatment
Defiers	Take Treatment if not assigned

1.1.2. Counterfacts

Causal inference relies on potential outcomes, representing the outcomes that would occur under different treatment conditions. Due to the "fundamental problem of causal inference," only one potential outcome is observed for each individual, while the other remains counterfactual.

$$\begin{aligned} Y(0) &= \text{Outcome if not treated (control)} \\ Y(1) &= \text{Outcome if treated (treatment)} \\ \text{Effect} &= Y(1) - Y(0) \end{aligned} \tag{1}$$

In more complex scenarios, such as stratified populations or mediation analysis, additional potential outcomes may be considered:

$$\begin{aligned} Y_0(0) &= \text{Outcome for group 0 if not treated} \\ Y_0(1) &= \text{Outcome for group 0 if treated} \\ Y_1(0) &= \text{Outcome for group 1 if not treated} \\ Y_1(1) &= \text{Outcome for group 1 if treated} \end{aligned} \tag{2}$$

We model Effect $= Y_1(1) - Y_0(0)$. The other two are unobserved.

1.1.3. Biases

Some common biases and means to fix them

Confounding Bias: Stratify confounding group
Selection Bias: Consider entire population
Omitted Variable Bias: Do not miss variables

1.1.4. Considerations

- Any control added to modelling outcome should be added when modelling treatment
- Exclude mediators, common effects of treatment and outcome

1.2. Methods

1.2.1. IV

Instrument Variable approach: Mitigates Omitted Variable Bias

Reduced Form Approach

- Reduced Form (RF): The direct relationship between the instrument and the outcome
- First Stage (FS): The relationship between the instrument and the treatment

$$\begin{aligned} \text{First Stage } T_i &= \pi_0 + \pi_1 Z_i + \epsilon_i \\ \text{Reduced Form } Y_i &= \gamma_0 + \gamma_1 Z_i + \nu_i \\ \text{IV Estimate } \beta_{IV} &= \frac{\gamma_1}{\pi_1} \end{aligned} \tag{3}$$

2SLS: Two-Stage Least Squares: Alternate Approach

$$\begin{aligned} \hat{T}_i &= \pi_0 + \pi_1 Z_i \\ Y_i &= \beta_0 + \beta_1 \hat{T}_i + u_i \end{aligned} \tag{4}$$

1.2.2. Matching

Pairing Treatment and Control group with similar co-variates (X)

- Use Distance function on covariates
- Use similar propensity score (preferred)

$$\begin{aligned} e(X_i) &= P(T_i = 1 \mid X_i) \\ ATET &= \frac{1}{N_T} \sum_{i \in T} \left(Y_i(1) - \sum_{j \in C} w_{ij} Y_j(0) \right) \\ w_{ij} &= \frac{K\left(\frac{d_{ij}}{h}\right)}{\sum_{k \in C} K\left(\frac{d_{ik}}{h}\right)} \end{aligned} \tag{5}$$

where

- w: Weight of sample
- K: Kernel function
- d: Distance function

1.2.3. Propensity

Use propensity scores

Propensity Weighting

Create weighted samples with propensity to balance control and treatment

$$ATET = \frac{\sum_{i \in T} \frac{Y_i(1)}{e(X_i)}}{\sum_{i \in T} \frac{1}{e(X_i)}} - \frac{\sum_{i \in C} \frac{Y_i(0)}{1-e(X_i)}}{\sum_{i \in C} \frac{1}{1-e(X_i)}} \tag{6}$$

Propensity Regression

Use propensity as a covariate in regression

$$Y_i = \alpha + \beta T_i + \gamma e(X_i) + \epsilon_i \tag{7}$$

1.2.4. DiD

Difference in Differences: Compare difference before and after applying treatment while controlling for time trends

$$\begin{aligned} \hat{ATET} = &(E[Y(1)|D=1] - E[Y(1)|D=0]) \\ &- (E[Y(0)|D=1] - E[Y(0)|D=0]) \end{aligned} \tag{8}$$

Where:

- D indicates the group (1 for treatment, 0 for control),
- $E[Y(1)|D=1]$ outcome for treated units after treatment,
- $E[Y(1)|D=0]$ outcome for untreated units after treatment,
- $E[Y(0)|D=1]$ outcome for treated units before treatment,
- $E[Y(0)|D=0]$ outcome for untreated units before treatment.

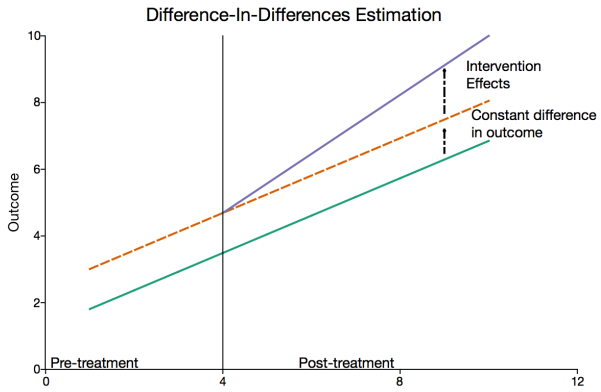

Figure 2. Difference in Differences [1]

NOTE: DiD assumes parallel trends

1.2.5. Panel Data

Longitudinal / Time Series Cross Sectional Data

Observations for the same units across time, with time-varying and time-invariant features

Fixed Effects: Individual specific characteristics are constant over time

$$Y_{it} = \alpha_i + \beta X_{it} + \epsilon_{it} \tag{9}$$

Where:

- Y_{it} outcome for individual i at time t,
- α_i individual-specific intercept (captures unobserved heterogeneity),
- β coefficient for the explanatory variable X_{it},
- ϵ_{it} error term.

Demeaning

$$\begin{aligned} \tilde{Y}_{it} &= Y_{it} - \bar{Y}_i \\ \tilde{X}_{it} &= X_{it} - \bar{X}_i \\ \tilde{Y}_{it} &= \beta \tilde{X}_{it} + \ddot{e}_{it} \end{aligned} \tag{10}$$

Where:

- $\bar{Y}_i$ and $\bar{X}_i$ are the means of Y and X for individual i across time.

Dummies

Include categorical dummies for each individual
Provides for individual intercepts
Can become computationally challenging

1.2.6. Synthetic Control

Creates a weighted combination of control units to approximate the counterfactual.

Pre-Treatment Matching: Identify untreated units with similar characteristics to the treated unit before intervention

Weighting Assign weights to untreated units so that their weighted average closely matches the treated unit pre-treatment

Post-Treatment Comparison Compare the post-treatment outcomes of the treated unit and the synthetic control, where the difference is considered the causal effect of the intervention.

$$\begin{aligned} \text{Weights} \min_{w_1,w_2,\ldots,w_n} &\|Y_{\text{treated}} - \sum_{i\in\mathcal{C}} w_i Y_i\|^2 \\ Y_{\text{synthetic}} &= \sum_{i\in\mathcal{C}} w_i Y_i \\ \hat{ATET} &= Y_{\text{treated}} - Y_{\text{synthetic}} \end{aligned} \tag{11}$$

Where:

- Y_t outcome for the treated unit at time t,
- X_t vector of pre-treatment covariates,
- W_0 vector of weights for the untreated units,
- Y_0^* synthetic control, the weighted average of untreated units.

1.2.7. RDD

Regression Discontinuity Design

Running Variable (r): Function defining treatment

- $r > c$: $T = 1$
- $r < c$: $T = 0$

$$y_i = \beta_0 + \beta_1 r_i + \beta_2 \mathbb{1}\{r_i > c\} + \beta_3 \mathbb{1}\{r_i > c\} r_i + \epsilon_i \tag{12}$$

Where:

- y_i is the outcome variable for unit i
- r_i is the running variable (assignment variable)
- $\mathbb{1}\{r_i > c\}$ indicator function
- ϵ_i is the error term

At the cutoff, $r_c \to 0$

- β_0 outcome just below the cutoff (pre-treatment)
- β_2 discontinuity (treatment effect) at the cutoff

1.3. More Methods

1.3.1. Sensitivity Analysis

How sensitive is treatment effect to changes in assumptions, data or methodology

1.3.2. Uplift Modelling

Estimate incremental impact of treatment with Machine Learning

Single Model with Treatment Indicator

Add a column of treatment to the data

$$\begin{aligned} \text{Train } \hat{Y}_i &= f(X_i, T_i) \\ \text{Infer } \hat{Y}_i &= f(X_i, T_i(1)) - f(X_i, T_i(0)) \end{aligned} \tag{13}$$

where:

- f is the predictive model (e.g., regression, decision tree)
- X_i are the features of individual i
- T_i is the binary treatment indicator (0 control, 1 treatment)
- Y_i is the outcome for individual i

Two Separate Models

Two separate algorithms to model the different treatment and control behaviours

$$\begin{aligned} \text{Train } \hat{Y}_i^{\text{treated}} &= f_{\text{treated}}(X_i) \\ \text{Train } \hat{Y}_i^{\text{control}} &= f_{\text{control}}(X_i) \\ \text{Infer Uplift}_i &= \hat{Y}_i^{\text{treated}} - \hat{Y}_i^{\text{control}} \end{aligned} \tag{14}$$

where:

- $\hat{Y}_i^{\text{treated}}$ is the predicted outcome for individual i if treated,
- $\hat{Y}_i^{\text{control}}$ is the predicted outcome for individual i if not treated,
- f_{treated} is the model trained on the treated group,
- f_{control} is the model trained on the control group.

2. A / B Testing and Bandits

2.1. A / B Testing

Comparing two (or more) versions of a product or service (e.g., Version A vs. Version B) to determine which performs better with respect to a predefined metric (e.g., conversion rate, click-through rate).

Null Hypothesis H_0: No difference between A and B

Alternate Hypothesis H_1: Significant difference between A and B

$$\begin{aligned} \text{t-stat} &= \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \\ \text{z-stat} &= \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \end{aligned} \tag{15}$$

Where:

- $\bar{x}_1, \bar{x}_2$: Sample means for groups A and B.
- σ_1^2, σ_2^2: Variances of the samples for groups A and B.
- n_1, n_2: Sample sizes for groups A and B.

- $\hat{p}_1 = \frac{x_1}{n_1}$: Proportion of successes in group A.
- $\hat{p}_2 = \frac{x_2}{n_2}$: Proportion of successes in group B.
- $\hat{p} = \frac{x_1 + x_2}{n_1 + n_2}$: Pooled proportion of successes across both groups.
- x_1, x_2: Number of successes in groups A and B.

2.2. Multi-Armed Bandits

Sequential decision-making framework where an agent chooses among k options (arms), aiming to maximize cumulative reward while balancing exploration and exploitation

ϵ Greedy

$$a = \begin{cases} \text{random action} & \text{with probability } \epsilon \\ \arg\max_a Q(s,a) & \text{with probability } 1-\epsilon \end{cases} \tag{16}$$

Upper Confidence Bound (UCB)

$$a = \arg\max_a \left[\hat{\mu}_a + c\sqrt{\frac{\ln t}{n_a}} \right] \tag{17}$$

where

- $\hat{\mu}_a$ Estimated mean reward for arm
- n_a Number of times arm a has been pulled
- t Total number of trials
- c Exploration parameter.

2.3. Contextual Bandits

An extension of the multi-armed bandit problem where the agent observes contextual information (e.g., user features, time of day) before selecting an arm.

Fits a model trained on user features and history to predict rewards and identify exploitation arm

$$\begin{aligned} r &= \theta^\top x + \epsilon \\ \theta &\leftarrow \theta + \alpha(r - \theta^\top x)x \end{aligned} \tag{18}$$

3. Recommender Systems

Recommend Items to Users

Explicit Feedback: Ratings

Implicit Feedback: Usage Metrics

Common Issues

- Cold Start: New Users generalized with popular items
- Long Tail: Rare Items to included via randomization

3.1. Association Rules

Relationships between items and user behaviour

$$\begin{aligned} \text{Support}(A) &= \frac{\text{Number of samples containing } A}{\text{Total number of samples}} \\ \text{Confidence}(A \to B) &= \frac{\text{Support}(A \cup B)}{\text{Support}(A)} \\ \text{Uplift}(A \to B) &= \frac{\text{Support}(A \cup B)}{\text{Support}(A) \cdot \text{Support}(B)} \end{aligned} \tag{19}$$

3.2. Content Based Filtering

3.2.1. Similarity Metrics

Import measures to capture similar items / users

$$\begin{aligned} \text{Cosine Similarity } (u, v) &= \frac{u \cdot v}{\|u\|\|v\|} \\ \text{Euclidean Distance } (u, v) &= \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2} \\ \text{Jaccard Similarity } (A, B) &= \frac{|A \cap B|}{|A \cup B|} \\ \text{Pearson Correlation } (u, v) &= \frac{\sum_{i=1}^{n}(u_i - \bar{u})(v_i - \bar{v})}{\sqrt{\sum_{i=1}^{n}(u_i - \bar{u})^2 \sum_{i=1}^{n}(v_i - \bar{v})^2}} \end{aligned} \tag{20}$$

3.2.2. Filtering

Recommend items based on item features

- Calculate item features from CNN / TF-IDF / Embeddings
- Build user preferences from weighted item interactions
- Recommend closest items

3.3. Collaborative Filtering

Predict missing values in the User Item interaction matrix

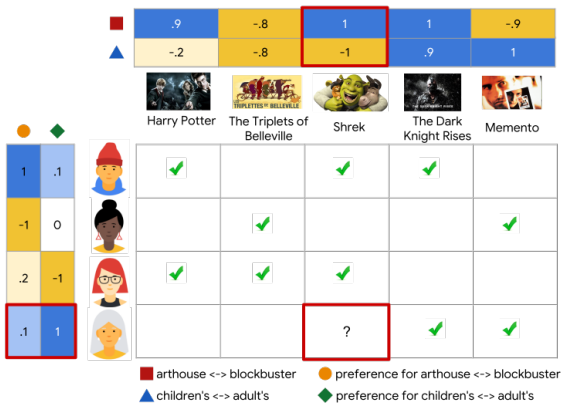

Figure 3. User Item Interaction Matrix [3]

3.3.1. User Item Filtering

User Based Filtering

Aggregate ratings of missing item from similar users having rated the same item

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in U} \text{Similarity}(u, v) \cdot (r_{v,i} - \bar{r}_v)}{\sum_{v \in U} |\text{Similarity}(u, v)|} \tag{21}$$

Item Based Filtering

Aggregate rating of missing user from similar items rated by the user

$$\hat{r}_{u,i} = \frac{\sum_{j \in I} \text{Similarity}(i, j) \cdot r_{u,j}}{\sum_{j \in I} |\text{Similarity}(i, j)|} \tag{22}$$

3.3.2. Matrix Factorization

Factorize Matrix into User and Item Matrices

$$R \approx P \cdot Q^T \tag{23}$$

where

- $R(m \times n)$ User Item Rating Matrix
- $P(m \times k)$ User matrix (m users, k features)
- $Q(n \times k)$ Item matrix (n items, k features)

Goal:

$$\min_{P,Q} \sum_{(u,i) \in \text{ Observed}} \left(R_{u,i} - P_u Q_i^T\right)^2 + \lambda \left(\|P_u\|^2 + \|Q_i\|^2\right) \tag{24}$$

Alternating Least Squares (ALS)

Iteratively solves by alternately fixing P and Q, and solving for the other using least squares (OLS).

$$\begin{aligned} \min_P \sum_{(u,i) \in \text{Observed}} \left(R_{u,i} - P_u Q_i^T\right)^2 + \lambda \|P_u\|^2 \\ \min_Q \sum_{(u,i) \in \text{Observed}} \left(R_{u,i} - P_u Q_i^T\right)^2 + \lambda \|Q_i\|^2 \end{aligned} \tag{25}$$

Singular Value Decomposition (SVD)

$$R = U \cdot \Sigma \cdot V^T \tag{26}$$

Where:

- $U(m \times k)$: User feature matrix
- $\Sigma(k \times k)$: Importance of features
- $V(n \times k)$: Item feature matrix

Stochastic Gradient Descent (SGD)
Iteratively updating the parameters to minimize the loss

$$\begin{aligned}\mathcal{L}(P,Q) &= \sum_{(u,i)\in\text{Observed}} \left(R_{u,i} - P_u Q_i^T\right)^2 + \lambda\left(\|P_u\|^2 + \|Q_i\|^2\right)\\ P_u &\leftarrow P_u + \alpha\left(2e_{u,i}Q_i - \lambda P_u\right)\\ Q_i &\leftarrow Q_i + \alpha\left(2e_{u,i}P_u - \lambda Q_i\right)\end{aligned} \tag{27}$$

3.4. Neural Network Approaches

Standard Neural Networks:

- Classify interaction
- Regress rating / feedback

3.4.1. Wide and Deep

Combine memorization (wide) and generalization (deep)

Wide

- Linear Model
- Cross Product of features
- Learns Simple Feature Interactions

Deep

- Feed forward Neural Network
- Dense embeddings of features
- Captures complex non-linear patterns

$$\hat{y} = \sigma(\mathbf{w}_{\text{wide}}^T \mathbf{x} + \mathbf{w}_{\text{deep}}^T f_{\text{deep}}(\mathbf{x})) \tag{28}$$

3.4.2. Two Tower

User Tower: Deep Neural Network for user embeddings
Item Tower: Deep Neural Network for item embeddings
Scoring: Dot product of embeddings is used to score

3.4.3. NeuCF

Neural Collaborative Filtering: Replace matrix factorization with a Neural Network approach

3.4.4. NeuMF

Combines NeuCF and Two Tower Approach

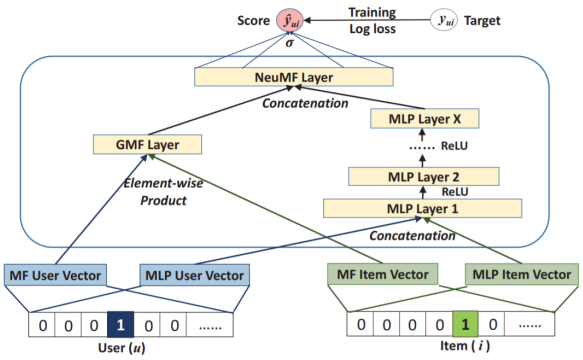

Figure 4. NeuMF - Neural Collaborative Filtering [2]

3.5. Metrics

3.5.1. NDCG

Normalized Discounted Cumulative Gain: Continuous Target

$$\begin{aligned}\text{NDCG} &= \frac{\text{DCG}_p}{\text{IDCG}_p}\\ \text{DCG}_p &= \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}\\ \text{IDCG} &= \text{Ideal Ranking}\end{aligned} \tag{29}$$

3.5.2. MAP

Mean Average Precision: Binary Target

$$\text{MAP} = \frac{1}{N}\sum_{q=1}^{N}\frac{1}{m_q}\sum_{k=1}^{m_q} P(k) \tag{30}$$

3.5.3. MRR

Mean Reciprocal Rank: First Relevant item

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\text{rank}_i} \tag{31}$$

3.6. Learning to Rank

LETOR (Learning to Rank) aims to rank a constant set of identical items for each user.

3.6.1. LambdaRank

Compute pairwise distances in scores between items

$$\begin{aligned}\Delta s_{ij} &= s_i - s_j\\ Loss &= \sum_{i,j}\sigma(-\Delta s_{ij})\cdot\log\left(1+e^{-\Delta s_{ij}}\right)\\ \frac{\partial L}{\partial s_i} &= \sum_{j\neq i}\lambda_{ij}\\ \lambda_{ij} &= \sigma\cdot\left|\Delta\text{NDCG}_{ij}\right|\cdot\frac{1}{1+e^{\Delta s_{ij}}}\end{aligned} \tag{32}$$

3.6.2. LambdaMART

Combination of LambdaRank and Gradient Boosted Decision Trees

- Updates performed on residuals
- Gradient for single item is sum with respect to all others for leaf value

$$\lambda_i = \sum_{j:i,j\in I}\lambda_{ij} - \sum_{j:j,i\in I}\lambda_{ij} \tag{33}$$

4. More Topics

4.1. Survival Analysis

Analyzing the expected duration until an event has to occur

4.1.1. Key Concepts

Important terminology

Hazard: $h(t) = \frac{f(t)}{S(t)}$ Rate of event at time, given survival till t

Censoring: Event has not occured till end, or individual has dropped out

Survival:$S(t) = P(T > t)$ Probability individual survives beyond t

4.1.2. Cox Proportional Hazards

Estimates the time to an event using other variables
Assumptions:

- Hazard ratio constant over time
- Hazard ratio independent of time

$$h(t|X) = h_0(t)\exp(\beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p) \tag{34}$$

where:

- $h(t|X)$ is the hazard function at time t given covariates $X_1, X_2, \ldots, X_p$,
- $h_0(t)$ is the baseline hazard function,
- $\beta_1, \beta_2, \ldots, \beta_p$ are the coefficients for covariates $X_1, X_2, \ldots, X_p$.

4.1.3. Kaplan Meier

Non-parametric statistic to estimate survival. Handles censoring

$$\hat{S}(t) = \prod_{i:t_i \le t} \left(1 - \frac{d_i}{n_i}\right) \tag{35}$$

where:

- d_i is the number of events at time t_i
- n_i is the number of individuals at risk just before time t_i

4.1.4. Metrics

Concordance (C) Index: Measure ability to rank pairs of observations

$$C = \frac{\sum_{i \neq j} \mathbb{I}(T_i < T_j \text{ and } \hat{T}_i < \hat{T}_j)}{\sum_{i \neq j} \mathbb{I}(T_i \neq T_j)} \tag{36}$$

where:

- T_i and T_j are the actual survival times for individuals
- $\hat{T}_i$ and $\hat{T}_j$ are the predicted survival times for individuals
- I Indicator function is 1 if predicted order matches actual order

4.2. LLMs Retrieval Augmented Generation

Enhance Large Language Models (LLMs) by retrieving relevant external information before generating responses.

4.2.1. Embeddings

Text embeddings are calculated with pre-trained models.

These are stored in a Vector Database suitable for fast retrieval and comparison

Comparison can be done by means of:

- Keyword search
- Similarity (Cosine, Euclidean, Jaccard...)
- Approximate Nearest Neighbours

4.2.2. ANNs

Approximate Nearest Neighbours (ANN): Efficient way to search through a large space with a trade-off on accuracy for speed

Local Sensitive Hashing

Hash functions to group similar items into the same buckets, allowing fast lookups.

1. Convert embeddings into hash values with family of hash functions $h_i(E_d)$.
2. Store embeddings with similar hash values in same bucket.
3. compute the hash for query and retrieve only items from the same bucket.

$$h(E) = \lfloor aE + b \rfloor \mod W \tag{37}$$

where:

- E is the embedding vector,
- a and b are randomly chosen parameters,
- W is the hash table width.

KD-Trees

Binary space-partitioning trees used for nearest neighbuor searches in lower dimensions (< 50).

1. Build tree by recursively splitting points along the axis with the highest variance.
2. Traverse the tree to find the closest stored points to query embedding.

Each node splits the space at the median of the dataset along dimension j:

$$\text{Split at } x_j = \text{median}(\{x_j^{(i)}\}) \tag{38}$$

4.2.3. Augmenting LLMs

1. Convert the query Q into an embedding $E_q = f(Q)$
2. Use an ANN method to retrieve the top k nearest neighbours
3. Return the most similar documents for augmentation
4. Add context to prompt for targeted response

4.2.4. Agent Flows

Agentic RAG involves iterative interactions between the retriever and the LLM, improving responses dynamically.

1. Agent iteratively refines query if initial retrieval is insufficient.
2. Retrieved documents guide agent to ask clarifying questions.
3. LLM adapts its output by refining context dynamically.

Agents can also be tasked to identify the steps to perform

This dynamic loop enables a more flexible, context-aware response generation compared to a single-pass retrieval model

4.3. Working with Less Data

4.3.1. Zero Shot

Generalize to tasks with no direct training examples by leveraging semantic relationships or pre-trained knowledge.

Key Techniques:

- **Embedding Similarity**: Matching inputs with known categories in a high-dimensional space (e.g., CLIP for vision-language tasks).
- **Natural Language Inference (NLI)**: Using entailment-based models to infer labels based on textual reasoning.
- **Prompt Engineering**: Guiding LLMs with well-crafted natural language instructions.

4.3.2. One Shot

Recognize new categories based on just one labelled example by relying on metric-based or memory-augmented methods.

Key Techniques:

- **Siamese Networks**: Learn a similarity function between pairs of inputs. Train cross product of each instance with all others learning both positive and negative relationships
- **Prototypical Networks**: Learn class prototypes (mean embeddings) in a space and classify by nearest prototype.

4.3.3. Few Shot

Extends one-shot learning to cases with a limited number of labelled examples per class. Useful when data collection is expensive or infeasible.

Key Techniques:

- **Meta-Learning ("Learning to Learn")**: Training models to quickly adapt to new tasks with minimal data.
- **Fine-Tuning Pre-trained Models**: Adapting large-scale pre-trained models with a small dataset.

References

[1] *DiD*. [Online]. Available: https://www.aptech.com/blog/introduction-to-difference-in-differences-estimation/.

[2] *NeuMF*. [Online]. Available: https://recbole.io/docs/user_guide/model/general/neumf.html.

[3] *UserItem*. [Online]. Available: https://developers.google.com/machine-learning/recommendation/collaborative/basics.