Module 6 - Opinion Mining and Sentiment Analysis:

6.1 Opinion Mining and Sentiment Analysis: The Problem of Opinion Mining

1. Introduction to Opinion Mining

- Opinion mining, also known as **sentiment analysis**, is the computational study of opinions, emotions, and attitudes in textual data.
- It enables us to extract structured insights from unstructured text by identifying and analyzing opinions expressed about entities, aspects, and events.
- Opinion mining is useful in various domains, including business (product reviews), politics (public sentiment on policies), healthcare (patient feedback), and social media analysis.

2. The Structure of Opinion Mining

Opinion mining consists of two main components:

1. **Opinion Definition**
2. **Opinion Summarization**

Both help transform unstructured text into structured, actionable insights.

---

3. Opinion Definition

6.1.1 Example: Understanding Opinion Mining

Consider the following review on an **iPhone**:

1. "I bought an iPhone a few days ago."
2. "It was such a nice phone." *(Positive opinion)*
3. "The touch screen was really cool." *(Positive opinion on touch screen aspect)*
4. "The voice quality was clear too." *(Positive opinion on voice quality aspect)*
5. "However, my mother was mad with me as I did not tell her before I bought it." *(Emotion, but not an opinion on the phone itself)*
6. "She also thought the phone was too expensive and wanted me to return it to the shop." *(Negative opinion on price aspect)*

6.1.2 Components of Opinion Mining

From the example, we can break down opinion mining into three key elements:

- **Opinion Target**: The entity or aspect that is being reviewed.
  - Example: "iPhone" (entity), "touch screen" (aspect), "price" (aspect).
- **Opinion Holder**: The person expressing the opinion.
  - Example: "I" (author of the review), "my mother" (another opinion holder).
- **Opinion Orientation (Sentiment)**: Whether the opinion is **positive**, **negative**, or **neutral**.
  - Example:
    - "The touch screen was really cool" → **Positive**
    - "The phone was too expensive" → **Negative**

6.1.3 Formal Definition of Opinion Mining

A formal representation of opinions is given by a **quintuple (ei, aij, ooijkl, hk, tl)**, where:

- **ei** → The entity being reviewed (e.g., "iPhone").
- **aij** → The aspect of the entity (e.g., "voice quality", "price").
- **ooijkl** → The opinion orientation (Positive, Negative, or Neutral).
- **hk** → The opinion holder (e.g., "I", "my mother").
- **tl** → The timestamp of the opinion (e.g., "Nov-4-2010").

This structured representation enables large-scale opinion aggregation and analysis.

---

4. Entity and Aspect Representation

3.1 Entity Definition

- An **entity (e)** is a **product, service, person, event, organization, or topic** that is evaluated.
- It is represented as **e: (T, W)** where:
  - **T** = A hierarchy of components and sub-components.
  - **W** = A set of attributes (features).

Example of an Entity (iPhone)

- **Entity**: iPhone

  - **Components**: Battery, Screen
  - **Attributes**: Voice quality, Size, Weight
  - **Battery Component Attributes**: Battery life, Battery size

- The entity is usually represented as a **tree structure**, but for simplicity, it is often flattened to **two levels**:
    - **Level 1** → The entity itself (e.g., "iPhone").
    - **Level 2** → Aspects of the entity (e.g., "Battery", "Screen", "Voice quality").

---

5. Types of Opinions

There are **two main types** of opinions in sentiment analysis:

1. **Regular Opinions**
    - Express direct **positive, negative, or neutral** opinions about an entity or aspect.
    - Example: "The phone's camera is excellent." (Positive opinion on camera)
2. **Comparative Opinions**
    - Compare two or more entities based on shared aspects.
    - Often use comparative or superlative adjectives.
    - Example: "iPhone is better than Samsung in battery life."

Most opinion mining research focuses on **regular opinions**, while comparative opinions require different techniques.

---

6. Tasks in Opinion Mining

To extract structured information from opinionated text, the following tasks are performed:

1. **Entity Extraction and Grouping**
    - Identify and cluster different mentions of the same entity.
    - Example: "iPhone", "Apple iPhone", and "iPhone 14" → grouped as the same entity.
2. **Aspect Extraction and Grouping**
    - Identify key aspects of the entity.
    - Example: "Camera quality", "Battery life", "Voice quality".
3. **Opinion Holder and Time Extraction**
    - Identify the person expressing the opinion and the time of the review.
    - Example: "Posted by John on March 10, 2023".

4. **Aspect Sentiment Classification**
    - Determine if each aspect is reviewed positively, negatively, or neutrally.
5. **Opinion Quintuple Generation**
    - Combine extracted information into structured quintuples for further analysis.

Example: Opinion Quintuples

For a blog post containing:

- "The voice of my Motorola phone was unclear, but the camera was good."
- "My girlfriend was happy with her Nokia phone."

Generated quintuples:

1. (Motorola, voice quality, negative, Author, Date)
2. (Motorola, camera, positive, Author, Date)
3. (Nokia, GENERAL, positive, Girlfriend, Date)

---

7. Subjectivity and Emotion Analysis

7.1 Subjectivity Classification

- **Objective Sentences**: Present factual information (e.g., "The phone has a 6-inch screen.").
- **Subjective Sentences**: Express personal opinions or emotions (e.g., "The screen is too small.").

Not all **subjective sentences contain opinions**, and **not all objective sentences lack opinions**.

Examples

- "The earphone broke in two days." → Objective but implies a **negative** opinion.
- "I want a phone with good voice quality." → Subjective but does not express a **clear** opinion.

7.2 Emotion Analysis

- **Emotions** are distinct from opinions but related.
- **Basic emotions**: Joy, Sadness, Anger, Fear, Surprise, Love.
- Some opinions carry emotional intensity, e.g.:

- "I am very angry with this shop." (Anger)
- "I am so happy with my iPhone." (Joy)

Opinion mining is sometimes extended to **emotion detection** for deeper analysis.

---

8. Aspect-Based Opinion Summarization

- In real-world applications, opinions from multiple users need to be summarized.
- **Aspect-Based Summarization** groups opinions by entity aspects.

Example: Aspect-Based Summary for a Phone

Aspect	Positive Reviews	Negative Reviews
GENERAL	125	7
Voice Quality	120	8

- Users can drill down into **individual review sentences** to understand why opinions are positive/negative.
- Visualization tools like **bar charts and pie charts** help represent this data quantitatively.

Conclusion

- Opinion mining helps **extract structured opinions** from large-scale text data.
- The **opinion quintuple** framework enables systematic analysis.
- Applications range from **product reviews and customer feedback** to **political and social media analysis**.
- The field continues to evolve, incorporating **emotion detection, comparative opinions, and deep learning techniques**

6.2 Document Sentiment Classification: Supervised vs. Unsupervised

Document sentiment classification refers to the task of determining the sentiment or opinion expressed in an entire document. This could involve classifying a document (e.g., product review, social media post) as positive, negative, or neutral based on the sentiment conveyed.

There are two main approaches for document sentiment classification:

- **Supervised Learning**
- **Unsupervised Learning**

---

1. Supervised Learning for Document Sentiment Classification:

Overview:

In supervised learning, we train a model on a labeled dataset where each document has a known sentiment label (positive, negative, or neutral). Once trained, the model can predict the sentiment of new, unseen documents based on learned patterns.

Steps Involved:

1. **Data Collection**: Gather a large dataset of opinionated documents (e.g., product reviews).
2. **Preprocessing**: Clean the text (e.g., remove stop words, lowercase the text, tokenize).
3. **Feature Extraction**: Convert the documents into numerical vectors (e.g., using TF-IDF, word embeddings).
4. **Model Training**: Use the labeled data to train a classifier (e.g., SVM, Naive Bayes).
5. **Prediction**: Apply the trained model to classify new, unseen documents.

Example (Movie Reviews):

Consider a dataset of movie reviews with the labels "Positive", "Negative", and "Neutral."

- **Review 1**: "The movie was fantastic, full of action and suspense!" (Positive)

- **Review 2**: "I was bored the entire time; it was a waste of money." (Negative)
- **Review 3**: "The movie was okay, neither good nor bad." (Neutral)

The goal of the supervised model is to train on this data and predict the sentiment of future reviews.

Key Techniques:

- **Naive Bayes Classifier**: This simple probabilistic model uses Bayes' Theorem and assumes that the presence of each word is independent of others. It works well for text classification tasks.
- **Support Vector Machines (SVM)**: SVMs find the optimal hyperplane that separates different classes (e.g., positive vs. negative sentiment) in the feature space.
- **Logistic Regression**: A regression model used to predict the probability that a document belongs to a certain class.

Features for Supervised Classification:

- **Bag-of-Words**: The text is represented as a collection of words with their frequency counts.
- **TF-IDF (Term Frequency-Inverse Document Frequency)**: A statistical measure used to evaluate the importance of a word within a document relative to the entire corpus.
- **Sentiment Lexicons**: Use predefined lists of positive and negative words to help classify sentiments.
- **POS (Part-of-Speech) Tagging**: Certain parts of speech like adjectives and adverbs are strong sentiment indicators. For instance, words like "amazing" or "horrible" are key to determining sentiment.

Example Supervised Model:

- **Training Data**:
  - "The movie was amazing" → Positive
  - "It was the worst movie I've ever seen" → Negative

    - "The movie was fine, neither bad nor good" → Neutral
- **Feature Extraction**:
    - Words: ["amazing", "worst", "fine"]
    - TF-IDF values are computed for each word.
- **Model Training**:
    - SVM is trained on the features, learning the relationship between words and their sentiment labels.
- **Prediction**:
    - For a new review, the trained model classifies it as Positive, Negative, or Neutral based on the learned features.

Advantages of Supervised Learning:

- **High Accuracy**: Since the model is trained on labeled data, it can often produce high accuracy.
- **Structured Training Process**: Clear steps to prepare, train, and test the model.

Disadvantages:

- **Requires Labeled Data**: A large amount of labeled data is needed for training.
- **Domain Dependency**: Models trained in one domain (e.g., movie reviews) might not perform well on others (e.g., restaurant reviews).

---

2. Unsupervised Learning for Document Sentiment Classification:

Overview:

In unsupervised learning, there are no predefined labels for the training data. Instead, the model tries to learn patterns and structures from the data itself. In the context of sentiment analysis, unsupervised methods often use sentiment lexicons and heuristic rules to determine the sentiment.

Steps Involved:

1. **Data Collection**: Gather a corpus of opinionated documents without labels.

2. **Feature Extraction**: Similar to supervised learning, but no labeled data is available for training.

3. **Model Training**: Use unsupervised techniques (e.g., clustering, sentiment lexicons) to classify sentiments.

4. **Prediction**: Classify the sentiment of new documents using the patterns learned.

Example (Review Sentiment Classification with Unsupervised Learning):

Consider the same dataset of movie reviews, but without sentiment labels. The goal is to classify reviews into positive and negative categories.

- **Review 1**: "I love the cast and the direction"

- **Review 2**: "This movie was terrible; I wouldn't recommend it."

Using unsupervised techniques, the system will rely on sentiment lexicons (lists of positive and negative words) and heuristic rules to classify these reviews.

Techniques for Unsupervised Classification:

- **Sentiment Lexicons**: Predefined lists of words with associated sentiment scores. Common lexicons include AFINN, SentiWordNet, and VADER. Each word is given a score, and the sentiment of the document is determined by summing the scores of words in the text.

  Example:

  - "Fantastic" → +3 (positive)

  - "Terrible" → -3 (negative)

- A review can be classified based on the sum of these word scores.

- **Clustering**: Group similar documents together, then analyze the dominant sentiment in each cluster. For example, documents discussing products could be grouped by sentiment (positive or negative) based on similarity in word usage.

- **Pointwise Mutual Information (PMI)**: Used to estimate the semantic orientation of phrases based on the co-occurrence of words in a corpus. This helps in understanding whether a phrase (e.g., "bad service") is typically associated with positive or negative sentiment.

  Example of a sentiment classification:

  - Extract "bad service" and "excellent performance" from reviews.
  - Calculate PMI for each phrase.
  - Classify as negative if PMI for "bad service" > PMI for "excellent performance."

Advantages of Unsupervised Learning:

- **No Labeled Data Required**: Unsupervised learning does not need manually labeled training data, which is often expensive and time-consuming to create.
- **Domain Flexibility**: The methods can be applied to any domain as they do not rely on labeled data specific to a particular domain.

Disadvantages:

- **Lower Accuracy**: Since the system is not trained on labeled data, its performance might be less accurate compared to supervised models.
- **Complexity**: The system might require more advanced techniques to extract meaningful features from unstructured text.

Summary of Key Differences Between Supervised and Unsupervised Learning:

Aspect	Supervised Learning	Unsupervised Learning
Data Requirement	Requires labeled data.	Does not require labeled data.
Accuracy	Typically more accurate due to labeled data.	May have lower accuracy as it relies on heuristic rules or lexicons.

Training Process	Involves training on labeled data using a classification algorithm.	Focuses on extracting patterns or using sentiment lexicons.
Domain Dependency	Can be domain-specific, requiring retraining for different domains.	More flexible and applicable to different domains.
Methods	Naive Bayes, SVM, Logistic Regression, etc.	Sentiment lexicons, Clustering, PMI, Heuristics.

Example of Supervised vs Unsupervised Classification:

- **Supervised Example**: You train a sentiment classifier using labeled data where reviews are marked as "positive," "negative," or "neutral." The classifier learns from these labels and predicts sentiments for new reviews.

- **Unsupervised Example**: You use a sentiment lexicon (e.g., SentiWordNet) to calculate the sentiment score for a given review, classifying it as positive or negative based on the presence of specific words like "good" (positive) or "terrible" (negative).

6.3 Opinion Lexicon Expansion

1. Introduction to Opinion Lexicon

An **opinion lexicon** is a collection of words, phrases, and idioms that express sentiment. It is widely used in **sentiment classification** tasks.

1.1 Types of Opinion Words

Opinion words can be classified into two main types:

1. **Base Type:** Expresses a direct opinion about an entity.
    - **Examples:**
        - Positive: *beautiful, wonderful, good, amazing*
        - Negative: *bad, poor, terrible*
2. **Comparative Type:** Expresses a **relative** sentiment between two or more entities.
    - **Examples:**
        - **Comparative Forms:** *better, worse*
        - **Superlative Forms:** *best, worst*
    - **Example Sentences:**
        - *"Car X is better than Car Y."* (Indicates that Car X is superior but does not specify if either is good or bad.)
        - *"This laptop is worse than my previous one."* (Indicates a negative comparison.)

2. Approaches to Expanding the Opinion Lexicon

There are three main methods to build an opinion lexicon:

2.1 Manual Approach

- **Process:** A human manually collects opinion words.
- **Advantage:** High accuracy.
- **Disadvantage:** Time-consuming and labor-intensive.
- **Usage:** Often used as a **final verification** step after automated methods.

2.2 Dictionary-Based Approach

- **Process:**
  - Start with a **small set of manually collected opinion words** (seed words).
  - Use **WordNet** (or similar dictionaries) to find synonyms and antonyms.
  - Expand the opinion lexicon iteratively.
  - Manually review to correct errors.
- **Example:**
  - Seed Word: *good*
  - Synonyms: *great, excellent, fantastic* → Added to the lexicon
  - Antonyms: *bad* → Added as a negative opinion word
- **Advantage:**
  - Easy to implement.
  - Works well for **general-purpose** sentiment words.
- **Disadvantage:**

- **Context-Specific Issues:** Some words change sentiment depending on the domain.
    - **Example:**
        - *"A quiet car"* (Positive)
        - *"A quiet speakerphone"* (Negative)

2.3 Corpus-Based Approach

- **Process:**
    1. Uses **large text corpora** and **linguistic rules** to discover new opinion words.
    2. Relies on **word co-occurrence patterns** and **seed words** to identify new sentiment words.
    3. Expands the opinion lexicon iteratively.

2.3.1 Sentiment Consistency in Corpus-Based Approach

- **Concept:** Opinion words appearing together in a sentence tend to share the same sentiment.
- **Example:**
    - *"The car is beautiful and spacious."*
        - Since "beautiful" is positive, "spacious" is likely positive too.
    - *"The car is beautiful but difficult to drive."*
        - "but" introduces contrast, so "difficult to drive" likely has a negative sentiment.
- **Implementation:**
    - Uses **machine learning models** and **graph-based clustering** to categorize words into **positive** and **negative** clusters.
- **Advantage:**

- Can find **context-specific** opinion words that the dictionary-based approach misses.

* **Disadvantage:**
    - Requires a **large domain-specific corpus** to be effective.

---

3. Context-Specific Opinion Words

* **Challenge:**
    - Some words change sentiment depending on the **aspect** being discussed.
* **Example (Digital Camera Domain):**
    - *"The battery life is long."* (Positive)
    - *"The time taken to focus is long."* (Negative)
* **Solution:**
    - Instead of classifying words in isolation, consider **(aspect, opinion_word) pairs**
    - Example: (*battery life*, *long*) → Positive
    - Example: (*focus time*, *long*) → Negative

---

4. Limitations of Opinion Lexicon Expansion

1. **Word Sense Ambiguity:**
    - Just because a word is in an opinion lexicon does not mean it expresses sentiment in every sentence.
    - **Example:**

        - *"I am looking for a good health insurance plan."*
        - "Good" does not express a **positive opinion**, just a requirement.

2. **Opinion Expressions Beyond Opinion Words:**
    - Some **sentiments** are expressed **without traditional opinion words**.
    - Example:
        - *"This phone drains battery in 2 hours."*
        - No explicit negative word, but implies dissatisfaction.

---

5. Summary

Approach	Description	Advantages	Disadvantages
Manual Approach	Human expert collects opinion words.	High accuracy.	Slow and time-consuming.
Dictionary-Based	Uses seed words and a dictionary (e.g., WordNet) to find synonyms/antonyms.	Easy to implement, good for general lexicons.	Fails for domain-specific sentiment words.
Corpus-Based	Uses real-world text data and co-occurrence patterns to expand the lexicon.	Identifies context-specific sentiment words.	Requires a large domain-specific dataset.

6.4 Opinion Spam Detection

1. Introduction

Opinion spam refers to **deliberate attempts to mislead** readers or automated sentiment analysis systems by writing **false or misleading reviews**. Similar to web spam (which manipulates search engine rankings), opinion spam influences consumer decisions and damages or promotes products unfairly.

2. Types of Opinion Spam

Opinion spam can be categorized into three types:

2.1 Type 1: Fake Reviews

These are intentionally deceptive reviews that:

- Give **undeserving positive opinions** to promote a product.
- Give **unjustified negative opinions** to harm competitors.

Example:

- *"This product changed my life! Best purchase ever!"* (Fake positive review)
- *"Worst product ever. Do not buy!"* (Fake negative review from a competitor)

2.2 Type 2: Brand-Centric Reviews

These reviews focus on the brand rather than the product.

Example:

- *"I hate HP. I never buy any of their products."* (Irrelevant to the specific product being reviewed)

2.3 Type 3: Non-Reviews

These do not contain a genuine opinion but appear in review sections. They include:

1. **Advertisements** disguised as reviews.
2. **Irrelevant text**, such as questions or random comments.

Example:

- *"Buy this product at a discount from our store!"* (Advertisement)
- *"Does anyone know if this works with Windows 11?"* (Not a review)

3. Harmful Fake Reviews

Fake reviews can have different levels of harm depending on their intent:

Product Quality	Positive Spam Review	Negative Spam Review
Good Quality	Not very harmful	Can damage reputation
Bad Quality	Misleads consumers	May reveal true flaws
Average Quality	Misleads buyers	Can unfairly damage sales

Spam detection should focus on **highly misleading reviews** that falsely promote or unfairly criticize products.

4. Types of Spammers

4.1 Individual Spammers

- Act alone and write multiple fake reviews.

- Create multiple accounts to avoid detection.
- May spam across multiple review sites.

4.2 Group Spammers

- A coordinated team writes fake reviews for promotion or defamation.
- Can control overall product sentiment by posting many reviews.
- May operate across multiple platforms to hide manipulation.

5. Hiding Techniques Used by Spammers

5.1 Techniques Used by Individual Spammers

1. **Building trust first**: Writing genuine reviews before posting fake ones.
2. **Creating multiple fake accounts**: Writing spam under different usernames.
3. **Selective spamming**: Only posting positive or negative reviews to avoid detection.
4. **Rating manipulation**: Giving high ratings but writing negative content to confuse automated systems.

5.2 Techniques Used by Group Spammers

1. **Coordinated reviews**: Posting reviews as a group to control overall sentiment.
2. **Posting at product launch**: Writing early fake reviews to set the initial perception.
3. **Spaced-out spam**: Posting at intervals to avoid detection spikes.
4. **Distributing spam across sites**: Using different review platforms to avoid detection algorithms.

Opinion spam is a growing issue that **misleads consumers** and **manipulates market reputation**. Understanding its types, identifying harmful fake reviews, and recognizing spammers' techniques are crucial for effective spam detection and prevention.

A)Spam Detection Based on Supervised Learning

Concept:

Spam detection is treated as a binary classification problem:

- **Spam reviews** (fake, misleading, promotional, or malicious)
- **Non-spam reviews** (genuine customer feedback)

Different types of spam require different handling. **Type 2 and Type 3 spam** (e.g., repetitive fake reviews) can be detected using labeled datasets and classification models, but **Type 1 spam** (carefully crafted fake reviews) is harder to detect.

Key Features for Model Training:

Three feature categories help in classification:

1. **Review-centric Features** (content-related)
    - Example: Number of brand mentions in a review, percentage of opinion words
    - Example: *"This product is the best! XYZ brand is amazing. XYZ is great."* (Excessive brand mention → likely spam)
2. **Reviewer-centric Features** (reviewer behavior)
    - Example: Average rating given by a reviewer, standard deviation of ratings
    - Example: A reviewer who always rates **5 stars** or **1 star** might be suspicious.
3. **Product-centric Features** (product characteristics)

    - Example: Product price, sales rank, review rating distribution
    - Example: A low-selling product with **sudden high ratings** might indicate fake promotions.

Example:

Suppose a reviewer gives **5-star ratings** to all products from one brand but rates other brands **1-star**. This behavior suggests possible bias or manipulation.

Model Used: Logistic Regression

- Identifies patterns in labeled spam vs. non-spam reviews
- Experimental results on **Amazon reviews** showed that Type 2 and Type 3 spam reviews are relatively easy to detect.

Challenges in Detecting Type 1 Spam (Fake Opinions)

- **Manually labeling** fake reviews is difficult.
- **Spammers craft reviews carefully** to resemble genuine reviews.
- **Possible Solution**: Use **duplicate reviews** for training.

Duplicate Review Analysis

Researchers analyzed **5.8 million Amazon reviews** and identified **duplicate and near-duplicate reviews**:

1. **Same user, same product** → Accidental repost or genuine update
2. **Different users, same product** → Possible coordinated fake reviews
3. **Same user, different products** → Suspicious brand promotion
4. **Different users, different products** → Fake reviews spread across multiple products

Key Findings:

- **Negative Outlier Reviews**: Extremely low ratings were often **spam**.
- **First & Only Reviewers**: Single-review users were more likely **spammers**.
- **Top-ranked Reviewers**: Some highly ranked Amazon reviewers were **suspicious**, writing thousands of reviews.
- **Fake Reviews Can Get "Helpful" Votes**: Spammers manipulate feedback to make fake reviews appear **credible**.
- **Low-selling Products Are More Targeted**: Harder to spam popular products due to genuine customer feedback.

B) Spam Detection Based on Abnormal Behaviors

Instead of using supervised learning, this method **detects unusual behaviors**.

Spam Behavior Models

1. **Targeting Products**
   - Example: A spammer consistently **rates one product highly** while attacking its competitor.
   - *User A gives Product X 5 stars and Product Y 1 star → Suspicious*
2. **Targeting Groups**
   - Example: A spammer **posts reviews on multiple products from the same brand** within a short time.
   - *User B writes 10 five-star reviews for Brand Z in 2 hours → Possible fake reviews*
3. **General Rating Deviation**
   - Example: A reviewer's rating **differs significantly** from the average.
   - *If 90% of users rate 4 stars but User C rates 1 star, they might be a spammer.*
4. **Early Rating Deviation**

- Example: A spammer **posts an extreme review right after launch** to influence early buyers.
- *Product P launched today, and a 5-star review appears within 10 minutes → Likely spam.*

Unexpected Rule Discovery (CAR-based Detection)

Instead of manually defining spam patterns, **Class Association Rules (CARs)** automatically find unusual behaviors.

Types of Unexpectedness Detected

1. **Confidence Unexpectedness**
   - *Example*: A reviewer gives **all positive ratings** to a brand, while most users rate negatively.
2. **Support Unexpectedness**
   - *Example*: A reviewer writes **multiple reviews for a single product**, while others write only one.
3. **Attribute Distribution Unexpectedness**
   - *Example*: Most positive reviews for a product come from a **single reviewer**, even though many people bought it.
4. **Attribute Unexpectedness**
   - *Example*: A reviewer gives only **positive reviews to Brand A** and only **negative reviews to Brand B** → Brand manipulation suspected.

Example Outcome:
Using **Amazon.com reviews**, researchers found that **many detected spammers followed these patterns**.

C) Group Spam Detection

Rather than targeting individual spammers, this approach detects **coordinated spam groups**.

Steps to Detect Group Spam

1. **Frequent Pattern Mining**
    - Identifies groups of **reviewers who frequently review the same products together**.
    - *Example*: A group of 5 reviewers **always rates the same brand highly** within hours of launch.
2. **Group Spam Indicators (Ranked with SVM Rank)**
    - **Timing Patterns**: Reviews posted together in a short period.
    - **Early Review Behavior**: Reviews posted **immediately after a product launch**.
    - **Content Similarity**: Copy-pasted reviews across products.
    - **Rating Deviation**: Group consistently gives extreme ratings.

Example:

- **Scenario**: A company hires a team to write fake 5-star reviews.
- **Detection**: Frequent pattern mining finds that the **same 10 reviewers** consistently rate the same **low-selling products**.

Conclusion:

- **Spam groups** often operate **systematically**, making them easier to detect than individual spammers.
- This method effectively catches **paid reviewers** and **bot-driven spam**.

Key Takeaways

1. **Supervised learning** (e.g., logistic regression) can detect **certain types of spam** (Type 2 & 3).

2. **Unsupervised methods** focus on **reviewer behavior anomalies** to catch more sophisticated spam (Type 1).

3. **Class Association Rules (CARs)** identify unexpected behaviors without predefined rules.

4. **Group spam detection** finds **coordinated** spam efforts using pattern mining.