Table of contents

Data Mining and Exploration

Student Edition

By Kimberly Seefeld, EdD

Published under **DataJoyAI** imprint

Companion resources for this book are available at:

https://datajoyai.com

Edition: First Edition
ISBN: 978-1-969233-31-9

Chapter 1

Data Mining in the Age of AI

1.1 Learning Outcomes

1-1. Explain the concept of data mining and its evolution from manual analysis to AI-enhanced techniques.

1-2. Identify and differentiate types of data (numeric, binary, text, image, audio, video, categorical, time series, and spatial) and describe how AI improves analysis for each.

1-3. Describe the AI-enhanced data mining process, including data exploration, preprocessing, mining, postprocessing, and application of knowledge.

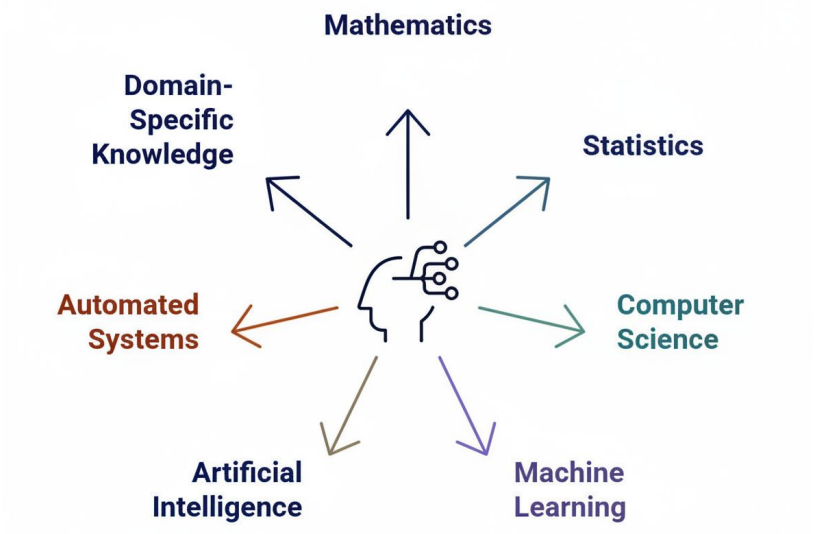
1-4. Recognize modern tools and platforms for AI-powered data mining, including traditional software, programming languages, AutoML, and no-code/low-code solutions.

1-5. Discuss ethical considerations and limitations in AI-enhanced data mining, including bias, privacy, transparency, and responsible AI practices.

1.2 Introduction

In this chapter we'll delve into the essence of data mining and cover some fundamental aspects that have been transformed by artificial intelligence. Our primary goal is to articulate what data mining entails in today's AI-enhanced landscape and set the stage for further study.

Data mining can be defined in a few ways depending on the perspective, but in 2025, it has become deeply intertwined with artificial intelligence and machine learning. Firstly, data mining can be viewed as a process. From this angle, data mining is akin to a search for patterns within a dataset, but now we have AI algorithms helping us find patterns we never could have discovered manually. Whether consumer data revealing purchasing behaviors through recommendation systems or medical data unveiling disease-symptom correlations via machine learning, the essence remains the same: identifying patterns within data. This perspective emphasizes the core aspect of data mining—uncovering meaningful patterns. In essence, data mining still mirrors the process of mining for gold, where the objective is to extract valuable insights from a vast expanse of data, but now we have much more sophisticated tools to help us dig.



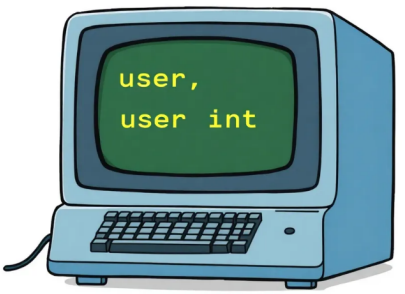
Academically, data mining is an interdisciplinary field that employs techniques from mathematics, statistics, computer science (including database management and programming), machine learning, artificial intelligence, and automated systems, along with domain-specific knowledge from fields like business and healthcare. Contrary to popular belief, data mining isn't solely a realm of mathematics or computer science, and it certainly isn't just about AI either. To set the stage for this book, it's crucial to understand that much like data analytics itself, data mining uses techniques from many academic disciplines, with AI serving as a powerful enhancement rather than a replacement

for fundamental analytical thinking.

1.3 History of Data Mining

To understand the history of data mining, it's essential to recognize that it's not entirely a novel field and has evolved over decades, with the most recent evolution being the integration of artificial intelligence. Data mining has roots that stretch back quite a while, though not in the AI-enhanced context we typically associate with it today. Let's review some key highlights to provide perspective on its evolution.

1.3.1 Early Computing (1960s–1980s)



The origins of data mining can be traced back to the 1960s. Early computing machines, though primitive by today's standards, hinted at the potential for automating analytical tasks. Before this period, data management primarily relied on manual processes, with information stored in physical documents within office settings.

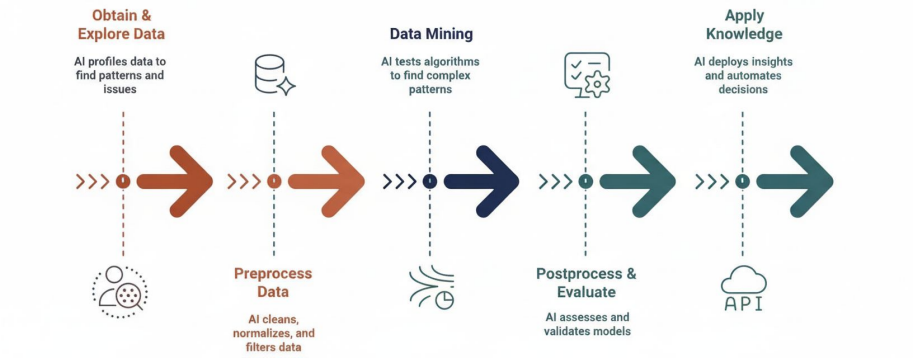
However, the landscape began to shift in the 1970s, marking the initial steps towards what we now recognize as data mining. As we moved into the 1970s, there was a gradual shift towards computerized data management. While still in its infancy, the concept of automating analytical processes became increasingly feasible with the emergence of rudimentary computing technology.

tal studies. AI has enhanced our ability to analyze location-based patterns and optimize routes and logistics.

Each type of data presents its own set of challenges and opportunities for analysis. Understanding these distinctions is essential for effectively leveraging data analytics techniques to derive actionable insights. The key difference in the AI era is that techniques of data mining can now be applied to all kinds of data more effectively, with AI serving as a powerful tool to uncover patterns that would be impossible to find manually.

1.5 AI-Enhanced Data Mining Process

The data mining process we're following is inspired by the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology, but it has evolved significantly with the integration of artificial intelligence and automation. While CRISP-DM provides a specific algorithm for data mining, our approach is adapted and modified to incorporate modern AI capabilities. The general pattern remains the same, though the specifics have been enhanced by intelligent automation.



1.5.1 Step 1: Obtain and Explore Data with AI Assistance

The initial phase involves gathering and examining the data, but now we have AI tools to help us do this more efficiently and thoroughly. Typically, data still comes from a database or data warehouse system maintained by IT professionals. An analyst will either receive the data from IT or retrieve it directly using SQL or related technology. Once the data is obtained, the analyst uploads it

Chapter 6

Classification Analysis

6.1 Learning Outcomes

6-1. Explain the concept of classification analysis as a supervised learning method and distinguish it from clustering and regression.

6-2. Describe the classification process, including data preparation, feature selection, training/testing split, model training, and validation.

6-3. Compare and contrast logistic regression, decision trees, and K-nearest neighbors (KNN) in terms of interpretability, accuracy, and applicability.

6-4. Evaluate classification models using performance metrics such as confusion matrices, accuracy, precision, recall, F1-score, ROC curves, and cross-validation.

6-5. Identify practical considerations for algorithm selection and understand when to explore advanced classification methods like neural networks or ensemble approaches.

6.2 Introduction

Classification analysis represents one of the most fundamental and widely used techniques in data mining. Unlike clustering, which works with unlabeled data, classification is a supervised learning method that uses labeled training data to predict categories or classes for new, unseen data. This chapter focuses on three core classification methods that form the foundation of predictive analytics, with

hands-on examples using modern AI-enhanced tools.

6.3 Supervised Learning

Classification falls under the umbrella of supervised learning, a branch of machine learning where algorithms learn from labeled training data to make predictions about new data. The fundamental difference between supervised and unsupervised learning lies in the presence of target variables or labels that guide the learning process.

6.3.1 Labeled Data and Target Variables

In supervised learning, we work with datasets that contain both input features (independent variables) and known outcomes (dependent variables or labels). For example, if we're building a model to predict whether an email is spam or not spam, our training data would include email features like sender information, subject line characteristics, and word frequencies, along with the known classification of each email as "spam" or "not spam."

The target variable, also called the response variable or dependent variable, represents what we want to predict. In classification problems, this target variable is categorical, meaning it represents discrete classes or categories rather than continuous numerical values. Common examples include predicting customer churn (yes/no), medical diagnosis (disease/no disease), or product categories

(electronics/clothing/books).

Modern AI tools can now automatically detect the type of target variable in your dataset and suggest appropriate classification algorithms. Platforms like Orange, WEKA, or even business intelligence tools like Tableau can analyze your data structure and recommend whether classification or regression techniques are most suitable.

6.3.2 Classification vs. Regression

While both classification and regression are supervised learning techniques, they differ in their target variables. Classification predicts discrete categories or classes, while regression predicts continuous numerical values. For instance, predicting house prices would be a regression problem (continuous values), while predicting house price categories (low/medium/high) would be a classification problem (discrete categories).

Understanding this distinction is crucial because it determines which algorithms and evaluation metrics are appropriate for your analysis. Classification algorithms are specifically designed to handle categorical outcomes and use different mathematical approaches than regression algorithms.

ing or decreasing the likelihood of a particular outcome. This makes the model especially useful when understanding the relationship between variables is just as important as making predictions. Logistic regression also works best when the classes can be reasonably separated using a linear boundary, meaning a straight-line (or flat surface in higher dimensions) can divide the groups.

6.13.1.6.1 Implementing Logistic Regression

In this section, the logistic regression model is created and trained using the training dataset. The model learns patterns by adjusting its internal coefficients so that the predicted probabilities match the actual class labels as closely as possible. The fit() function performs this learning process.

After training, the model is used to make predictions on new, unseen data stored in the test set. The predict() function returns the final predicted class for each observation, while predict_proba() provides the probability that each observation belongs to each possible class. This distinction is important because it allows us to see not just what the model predicts, but how confident it is in those predictions.

Finally, the model's performance is evaluated using accuracy scores on both the training and testing datasets. The training accuracy shows how well the model learned from the data it was given, while the testing accuracy indicates how well it generalizes to new data.

```
from sklearn.linear_model import LogisticRegression

log_reg = LogisticRegression(random_state=42, max_iter=1000)

# Create and train logistic regression model
print("LOGISTIC REGRESSION ANALYSIS")
print("=" * 50)

# Initialize the model
log_reg = LogisticRegression(random_state=42, max_iter=1000)

# Train the model
```

```
log_reg.fit(X_train, y_train)

# Make predictions
y_pred_log = log_reg.predict(X_test)
y_pred_proba_log = log_reg.predict_proba(X_test)

print("Model Training Complete!")
print(f"Training Accuracy: {log_reg.score(X_train, y_train):.3f}")
print(f"Testing Accuracy: {log_reg.score(X_test, y_test):.3f}")
```

Output:

```
Model Training Complete!
Training Accuracy: 0.931
Testing Accuracy: 0.933
```

6.13.1.6.2 Interpreting Logistic Regression Coefficients

In this section, the model's coefficients are examined to understand how each feature influences the predictions. When the model was trained, it assigned a weight to every feature, and these weights are stored in log_reg.coef_. By organizing these values into a table, we can clearly see how each variable contributes to predicting each class.

```
# Analyze feature importance through coefficients
feature_names = X.columns
classes = log_reg.classes_

print("\nLogistic Regression Coefficients Analysis")
print("=" * 45)

# Create coefficient interpretation table

coef_df = pd.DataFrame(
    log_reg.coef_.T,
    index=feature_names,
    columns=[f'Coef_{c}' for c in classes]
```

```
1 Decision Tree 0.933


2 K-Nearest Neighbors 0.933
```

```python
# Detailed classification reports
for model_name, (predictions, _) in models.items():
    print(f"\n{model_name} - Detailed Performance:")
    print("-" * 45)
    print(classification_report(y_test, predictions))
```

Output:

```
Logistic Regression - Detailed Performance:
---------------------------------------------
                   precision    recall  f1-score   support

        Excellent       0.93      1.00      0.97       140
Needs Improvement       0.00      0.00      0.00         1
     Satisfactory       0.00      0.00      0.00         9

         accuracy                           0.93       150
        macro avg       0.31      0.33      0.32       150
     weighted avg       0.87      0.93      0.90       150


Decision Tree - Detailed Performance:
---------------------------------------------
                   precision    recall  f1-score   support

        Excellent       0.93      1.00      0.97       140
Needs Improvement       0.00      0.00      0.00         1
     Satisfactory       0.00      0.00      0.00         9

         accuracy                           0.93       150
        macro avg       0.31      0.33      0.32       150
     weighted avg       0.87      0.93      0.90       150
```

```
K-Nearest Neighbors - Detailed Performance:
--------------------------------------------
                   precision    recall  f1-score   support

        Excellent       0.93      1.00      0.97       140
Needs Improvement       0.00      0.00      0.00         1
     Satisfactory       0.00      0.00      0.00         9

         accuracy                           0.93       150
        macro avg       0.31      0.33      0.32       150
     weighted avg       0.87      0.93      0.90       150
```

6.13.1.9.1 Confusion Matrix Analysis

The final part focuses on confusion matrix analysis, which visualizes where each model is making correct versus incorrect predictions. A confusion matrix shows actual classes along one axis and predicted classes along the other. The diagonal values represent correct predictions, while off-diagonal values indicate misclassifications. The code plots all three confusion matrices side by side using heatmaps, making it easier to compare model performance visually.

Overall, this workflow helps not only to quantify how well each model performs but also to interpret the types of errors each model makes. By combining accuracy scores, classification reports, and confusion matrices, you get a comprehensive picture of model strengths and weaknesses, enabling informed decisions about which model to choose for deployment or further tuning.

```
# Create confusion matrices for all models
print("\nCONFUSION MATRICES (TABLE FORMAT)")
print("=" * 40)

for model_name, (predictions, _) in models.items():
    cm = confusion_matrix(y_test, predictions)

    print(f"\n{model_name}")
    print("-" * len(model_name))
```

```
    print(pd.DataFrame(
        cm,
        index=[f"Actual {cls}" for cls in log_reg.classes_],
        columns=[f"Pred {cls}" for cls in log_reg.classes_]
    ))
```

Output:

```
Logistic Regression
-------------------
                         Pred Excellent  Pred Needs Improvement  \
Actual Excellent                    140                       0
Actual Needs Improvement              1                       0
Actual Satisfactory                   9                       0

                         Pred Satisfactory
Actual Excellent                         0
Actual Needs Improvement                 0
Actual Satisfactory                      0

Decision Tree
-------------
                         Pred Excellent  Pred Needs Improvement  \
Actual Excellent                    140                       0
Actual Needs Improvement              1                       0
Actual Satisfactory                   9                       0

                         Pred Satisfactory
Actual Excellent                         0
Actual Needs Improvement                 0
Actual Satisfactory                      0

K-Nearest Neighbors
-------------------
```

```
                'Support': f"{rule['support']:.3f}",
                'Confidence': f"{rule['confidence']:.3f}",
                'Lift': f"{rule['lift']:.3f}",
                'Count': rule['count']
            })

        return pd.DataFrame(summary_data)

# Run the Apriori implementation
apriori = AprioriAlgorithm(min_support=0.2, min_confidence=0.6)
apriori.fit(transactions)

# Display results
print(f"\nFINAL ASSOCIATION RULES")
print("=" * 30)
rules_summary = apriori.get_rules_summary()
print(rules_summary.to_string(index=False))
```

When the Apriori algorithm starts, it first prints out the settings and size of your dataset. Here, the minimum support is set at 0.2, meaning only itemsets that appear in at least 20% of transactions will be considered "frequent." The minimum confidence is 0.6, so only rules that are correct at least 60% of the time will be kept. The algorithm sees that there are 10 transactions in total.

7.9.1.4.1 Phase 1: Finding Frequent Itemsets

The algorithm begins by examining individual items (1-itemsets) to see which ones appear frequently enough. It finds 7 frequent 1-itemsets, such as bread, milk, and butter. Then it combines these frequent items to form 2-itemsets (pairs) and checks which of these pairs meet the minimum support. For example, bread & milk occurs in 40% of transactions, so it is included. This process continues to 3-itemsets (triples), such as bread, butter, milk, which occurs in 30% of transactions. The algorithm stops when no larger frequent combinations can be found.

By the end of Phase 1, the algorithm has 18 frequent itemsets: 7 single items, 8

8.9 Key Terms

AI-Enhanced Data Mining – Using artificial intelligence to improve or automate data mining tasks.

Automated Feature Engineering – AI-driven creation of new variables to improve predictive performance.

Feature Selection – Identifying the most relevant variables for analysis to avoid overfitting and improve interpretability.

Natural Language Processing (NLP) – AI techniques for analyzing and extracting insights from text data.

Sentiment Analysis – Determining emotional tone or opinion from text.

Named Entity Recognition (NER) – Identifying and classifying key elements in text, such as names, dates, or products.

Topic Modeling – Discovering hidden themes in large text datasets.

Computer Vision – AI methods to interpret and analyze images and videos.

Image Classification / Object Detection / Image Segmentation / OCR – Core computer vision techniques for analyzing visual data.

Ethical AI – AI designed to be fair, transparent, accountable, and aligned with human values.

Explainable AI (XAI) – Methods that make AI decision-making understandable to humans.

Federated Learning – Collaborative machine learning without centralizing data.

Generative AI – AI that can create new content or synthetic data for training and analysis.

Human-AI Collaboration / Augmented Analytics – Combining AI capabilities with human expertise for better insights and decisions.

8.10 Review Questions

1. How does AI enhance traditional data mining techniques like clustering and classification?
2. What is automated feature engineering, and why is it important for predictive modeling?
3. Explain the difference between NLP and computer vision in AI-enhanced data mining.
4. What are some key techniques used in NLP for data mining applications?
5. How does computer vision transform analysis of image and video data?
6. Why is ethical AI important, and what types of bias can occur in AI systems?
7. Describe strategies for mitigating bias and ensuring transparency in AI-enhanced data mining.
8. How can generative AI be used to create synthetic data or generate insights?
9. What is federated learning, and why is it useful for privacy-sensitive domains?
10. Explain the concept of human-AI collaboration and how it supports augmented analytics.

8.11 Lab 8: AI Enhanced Data Mining

This lab demonstrates three core areas:

Automated Feature Engineering – Learn how AI-inspired methods can generate new variables from raw data, capture complex relationships, and improve the predictive potential of datasets. Students will practice creating mathematical transformations, interaction features, aggregation metrics, and categorical bins, followed by intelligent feature selection using statistical methods.

Natural Language Processing (NLP) for Text Mining – Examine how unstructured text data, such as customer reviews, can be transformed into structured