# CSCI316: Big Data Mining Techniques and Implementation

## Comprehensive Exam Notes

---

## Subject Overview & Structure

**Course Focus**: Big Data project lifecycle, processing models, data mining algorithms, and real-time stream processing using popular programming libraries and platforms.

**Assessment Structure**:

- Individual Assignments: 30% (2 × 15%)
- Group Assignment: 20% (2 × 10%)
- Final Exam: 50%

---

## Learning Outcomes & Key Concepts

### Primary Learning Objectives

1. **Big Data Project Lifecycle** - Understanding end-to-end project development
2. **Processing Models & Methodologies** - Various approaches to handle big data
3. **Data Pre/Post-processing** - Cleaning, transformation, and preparation techniques
4. **Data Mining Algorithms** - Implementation of core algorithms for big data
5. **Real-time Processing** - Stream mining and live data processing methods
6. **Practical Implementation** - Using popular libraries and platforms

### Dual Learning Approach

- **Practical Perspective**: Tools, libraries, and complete project lifecycle
- **Theoretical Perspective**: Deep understanding of algorithms and low-level coding

---

## Lecture Topics & Key Areas

### Lecture 1: Introduction to Big Data & Programming Basics

**Reference**: Han et al. Chapter 1

**Key Concepts**:

- Definition and characteristics of Big Data (Volume, Velocity, Variety, Veracity, Value)
- Big Data ecosystem overview
- Data collection methods and sources

- Programming fundamentals for big data processing
- Introduction to distributed computing concepts

**Exam Focus**:

- Big Data 5 V's characteristics
- Differences between traditional data processing and big data processing
- Data collection strategies and challenges

### Lecture 2: Data Pre-processing

**Reference**: Han et al. Chapters 2 & 3

**Key Concepts**:

- Data cleaning techniques
- Data integration and transformation
- Data reduction methods
- Handling missing values, outliers, and noise
- Data normalization and standardization
- Feature selection and engineering

**Exam Focus**:

- Data quality issues and solutions
- Pre-processing techniques for different data types
- Impact of pre-processing on algorithm performance
- Scalability considerations for big data pre-processing

### Lecture 3: Big Data Project Life-cycle

**Reference**: Geron Chapter 2

**Key Concepts**:

- End-to-end project workflow
- Problem definition and scoping
- Data acquisition and exploration
- Model selection and training
- Evaluation and deployment
- Monitoring and maintenance
- Iterative improvement processes

**Exam Focus**:

- Project lifecycle phases and their importance
- Decision points in big data projects
- Best practices for project management
- Common pitfalls and how to avoid them

## Lecture 4: Classification by Splitting Data Sets

**Reference**: Han et al. Chapter 8

**Key Concepts**:

- Decision tree algorithms (ID3, C4.5, CART)
- Tree pruning techniques
- Handling categorical and continuous attributes
- Information gain and entropy
- Gini impurity and splitting criteria
- Ensemble methods overview

**Exam Focus**:

- Decision tree construction algorithms
- Splitting criteria comparison
- Overfitting prevention techniques
- Computational complexity considerations

## Lecture 5: Probabilistic Classification & Model Evaluation

**Reference**: Han et al. Chapter 8

**Key Concepts**:

- Naive Bayes classifier
- Bayesian networks
- Model evaluation metrics (accuracy, precision, recall, F1-score)
- Cross-validation techniques
- ROC curves and AUC
- Confusion matrices
- Statistical significance testing

**Exam Focus**:

- Naive Bayes assumptions and applications
- Evaluation metric selection for different problems
- Cross-validation strategies for big data
- Interpreting evaluation results

### Lecture 6: Handling Massive Data Sets

**Reference**: Chambers & Zaharia Chapters 1 & 24

**Key Concepts**:

- Distributed computing principles
- Apache Spark architecture and components
- RDDs (Resilient Distributed Datasets)
- DataFrames and Datasets
- Spark SQL and data processing
- Memory management and optimization
- Fault tolerance mechanisms

**Exam Focus**:

- Spark architecture and execution model
- RDD operations and transformations
- Performance optimization strategies
- When to use Spark vs traditional processing

### Lecture 7: Training Artificial Neural Networks

**Reference**: Geron Chapter 10

**Key Concepts**:

- Neural network fundamentals
- Backpropagation algorithm
- Gradient descent optimization
- Activation functions
- Deep learning architectures
- TensorFlow implementation
- Training strategies and hyperparameter tuning

**Exam Focus**:

- Neural network training process
- Common activation functions and their properties
- Optimization algorithms comparison
- Deep learning best practices

---

## Programming & Technical Components

### Core Technologies

- **Python 3** with scientific libraries (NumPy, Pandas, Matplotlib)
- **Scikit-Learn** for machine learning
- **Apache Spark & PySpark** for big data processing
- **TensorFlow** for deep learning
- **Google Colab** as development environment

### Implementation Skills

- Data manipulation with Pandas
- Distributed processing with Spark
- Machine learning pipeline development
- Neural network implementation
- Performance optimization techniques

---

## Key Algorithms & Techniques

### Data Mining Algorithms

1. **Decision Trees**: ID3, C4.5, CART
2. **Naive Bayes**: Gaussian, Multinomial, Bernoulli variants
3. **Neural Networks**: Feedforward, backpropagation
4. **Ensemble Methods**: Random Forest, Gradient Boosting

### Big Data Processing Techniques

1. **MapReduce paradigm**
2. **Spark transformations and actions**
3. **Stream processing concepts**
4. **Distributed storage systems**

### Evaluation Methods

1. **Cross-validation strategies**
2. **Performance metrics selection**
3. **Statistical significance testing**
4. **Scalability assessment**

---

## Exam Preparation Strategy

### Theoretical Understanding

- Master fundamental concepts from each lecture topic
- Understand algorithm mechanics and mathematical foundations
- Know when to apply different techniques
- Comprehend scalability and performance implications

### Practical Skills

- Practice implementing algorithms from scratch
- Work with provided libraries and frameworks
- Understand parameter tuning and optimization
- Experience with real datasets and processing pipelines

### Integration Knowledge

- Connect theoretical concepts with practical implementation
- Understand trade-offs between different approaches
- Know how to design complete big data solutions
- Appreciate the importance of both levels of understanding

---

## Important Formulas & Concepts

### Information Theory

- **Entropy**: $H(S) = -\Sigma\ p(x) \log_2 p(x)$
- **Information Gain**: $IG(S,A) = H(S) - \Sigma\ |Sv|/|S| \times H(Sv)$
- **Gini Impurity**: $Gini(S) = 1 - \Sigma\ p(x)^2$

### Evaluation Metrics

- **Accuracy**: (TP + TN) / (TP + TN + FP + FN)
- **Precision**: TP / (TP + FP)
- **Recall**: TP / (TP + FN)

- **F1-Score**: 2 × (Precision × Recall) / (Precision + Recall)

### Neural Networks

- **Sigmoid**: $\sigma(x) = 1 / (1 + e^{-x})$
- **ReLU**: $f(x) = \max(0, x)$
- **Gradient Descent**: $\theta = \theta - \alpha \times \nabla J(\theta)$

---

## Study Tips

1. **Focus on both theory and implementation** - The course emphasizes understanding at both levels
2. **Practice with real datasets** - Use tools like Google Colab for hands-on experience
3. **Understand scalability implications** - Big data solutions must handle massive datasets
4. **Connect concepts across lectures** - See how pre-processing affects algorithm performance
5. **Review reference materials** - Use the three main textbooks for deeper understanding
6. **Practice coding** - Implement algorithms from scratch to test understanding

---

*Note: This overview is based on the subject outline. Refer to actual lecture materials, assignments, and additional resources provided on Moodle for complete preparation.*