

## Day 1: Introduction to Text Mining and NLP

- **Overview of Text Mining:** Definitions, applications, and use cases.
- **Introduction to NLP:** Key NLP tasks (e.g., sentiment analysis, topic modeling, information extraction).
- **NLP vs. Text Mining:** Understanding the similarities and differences.
- **NLP Tools and Libraries:** Overview of popular NLP tools (NLTK, spaCy, Gensim) and frameworks (TensorFlow, PyTorch).

## Day 2: Text Preprocessing

- **Data Collection and Sources:** Methods for collecting textual data (web scraping, APIs).
- **Cleaning Text Data:** Removing punctuation, numbers, and special characters.
- **Tokenization:** Splitting text into words, sentences, and characters.
- **Stop Words Removal:** Filtering out common words.
- **Stemming and Lemmatization:** Reducing words to root forms.

## Day 3: Text Analysis in R

- **Setting up R for Text Mining:** Installing necessary packages (tm, quanteda, tidytext).
- **Data Import and Management:** Reading and handling text data.
- **Basic Text Manipulation:** Word frequency analysis, term-document matrix (TDM), word clouds.
- **Visualization:** Creating visualizations for text insights using ggplot2 and other libraries.

## Day 4: Text Analysis in Python

- **Setting up Python for Text Analysis:** Overview of NLTK, spaCy, and textblob.
- **Data Import and Exploration:** Handling text data with Pandas.
- **Basic NLP Operations:** Tokenization, POS tagging, named entity recognition (NER).
- **Visualization:** Creating word clouds, frequency distributions, and other text-based visualizations.

## Day 5: Sentiment Analysis

- **Overview of Sentiment Analysis:** Use cases in marketing, finance, and social media.
- **Rule-Based vs. Machine Learning Approaches:** Approaches to sentiment analysis.
- **Sentiment Analysis in Python:** Using TextBlob and Vader for sentiment scoring.
- **Sentiment Analysis in R:** Using tidytext for sentiment analysis.

## Day 6: Topic Modeling

- **Understanding Topic Modeling:** Applications and basic concepts.
- **Latent Dirichlet Allocation (LDA):** Theory and implementation in Python (Gensim) and R.
- **Topic Coherence:** Techniques for evaluating topic models.
- **Visualizing Topics:** Using pyLDAvis in Python or LDAvis in R.

## Day 7: Advanced NLP Techniques

- **Word Embeddings:** Overview of word2vec, GloVe, and FastText.
- **Transformers and BERT:** Introduction to transformer models and BERT architecture.
- **Named Entity Recognition (NER):** Implementation using spaCy and other libraries.
- **Text Summarization:** Extractive vs. abstractive summarization techniques.

## Day 8: Building NLP Models

- **Text Classification:** Building and evaluating classifiers (e.g., Naive Bayes, SVM, neural networks).
- **Sequence Models:** RNN, LSTM, and GRU architectures for NLP.
- **Transfer Learning in NLP:** Fine-tuning pre-trained models (Hugging Face Transformers).
- **Model Evaluation:** Metrics specific to NLP (precision, recall, F1-score, accuracy).

## **Day 9: Case Studies and Applications**

- **Real-World Case Studies:** Review of successful NLP applications across different sectors.
- **Hands-On Projects:** Small group projects where participants work on data to build and test models.
- **Challenges and Limitations in NLP:** Discussion on biases, ethical concerns, and limitations in NLP models.

## **Day 10: Review and Q&A**

- **Review of Key Concepts and Techniques:** Recap of topics covered and practical applications.
- **Open Q&A Session:** Address participant questions, troubleshooting, and clarifications.
- **Final Project Presentations:** Participants present their final projects and findings.
- **Feedback and Future Directions:** Collect feedback and discuss further learning resources and trends in NLP.