

## Roadmap Overview

1. **Foundations:** Learn Python, NLP basics, and sentiment analysis concepts.
2. **Tools & Libraries:** Get comfortable with NLP libraries and pre-trained models.
3. **LLMs Introduction:** Understand LLMs, their architecture, and usage.
4. **Sentiment Analysis with LLMs:** Apply LLMs to sentiment tasks.
5. **Fine-Tuning & Optimization:** Customize models for better performance.
6. **Deployment & Scaling:** Build and deploy a real-world application.

## 8-Week Program

### Week 1: Python & NLP Basics

**Goal:** Build a foundation in Python and understand NLP fundamentals.

**Tasks:**

1. Learn Python essentials: variables, loops, functions, and libraries (e.g., NumPy, Pandas).
2. Explore basic NLP concepts: tokenization, stemming, lemmatization, stop words.
3. Install tools: Python, Jupyter Notebook, and libraries (e.g., NLTK, spaCy).

**Key Task:**

- Write a Python script to tokenize and clean a sample text (e.g., a movie review).

**Resources:**

- Python: "Python Crash Course" (book) or freeCodeCamp tutorials.
  - NLP: NLTK documentation or "Natural Language Processing with Python" (book).
- 

### Week 2: Sentiment Analysis Basics

**Goal:** Understand sentiment analysis and experiment with rule-based methods.

**Tasks:**

1. Learn sentiment analysis theory: positive/negative/neutral classification, lexicon-based approaches.
2. Use a lexicon-based tool (e.g., VADER from NLTK) for sentiment analysis.
3. Process a small dataset (e.g., tweets or reviews) and analyze sentiment.

**Key Task:**

- Apply VADER to classify sentiment in a dataset of 10 sentences and print results.

**Resources:**

- VADER: NLTK documentation.
  - Dataset: Kaggle (e.g., Twitter Sentiment Analysis dataset).
- 

### Week 3: Machine Learning for NLP

**Goal:** Introduce ML concepts and traditional NLP models.

**Tasks:**

1. Learn ML basics: supervised learning, feature extraction (e.g., TF-IDF, Bag of Words).
2. Use scikit-learn to build a simple sentiment classifier (e.g., Logistic Regression).
3. Evaluate model performance: accuracy, precision, recall, F1-score.

**Key Task:**

- *Train a Logistic Regression model on a labeled sentiment dataset and evaluate it.*

**Resources:**

- Scikit-learn docs.
  - Tutorial: "Sentiment Analysis with Scikit-learn" (Towards Data Science).
- 

**Week 4: Introduction to LLMs**

**Goal:** Understand LLMs and experiment with pre-trained models.

**Tasks:**

1. Learn LLM basics: transformers, attention mechanisms, pre-training vs. fine-tuning.
2. Install Hugging Face Transformers library and explore pre-trained models (e.g., BERT).
3. Run a pre-trained LLM for basic text generation or classification.

**Key Task:**

- *Use BERT to classify a sample sentence's sentiment via Hugging Face pipeline.*

**Resources:**

- Hugging Face: Transformers docs & tutorials.
  - Book: "Deep Learning with Python" (Chollet) for transformers intro.
- 

**Week 5: Sentiment Analysis with LLMs**

**Goal:** Apply LLMs specifically to sentiment analysis.

**Tasks:**

1. Explore sentiment-specific models (e.g., DistilBERT, RoBERTa fine-tuned for sentiment).
2. Load a pre-trained sentiment model and test it on a dataset.
3. Compare results with traditional methods (e.g., VADER or ML model from Week 3).

**Key Task:**

- *Run a pre-trained sentiment LLM on a dataset and compare accuracy with VADER.*

**Resources:**

- Hugging Face Model Hub (sentiment models).
  - Tutorial: "Sentiment Analysis with BERT" (Hugging Face or YouTube).
-

## Week 6: Fine-Tuning LLMs

**Goal:** Customize an LLM for improved sentiment analysis.

**Tasks:**

1. Learn fine-tuning concepts: datasets, loss functions, hyperparameter tuning.
2. Prepare a labeled sentiment dataset (e.g., IMDb reviews or custom data).
3. Fine-tune a small LLM (e.g., DistilBERT) on your dataset using Hugging Face.

**Key Task:**

- *Fine-tune DistilBERT on a sentiment dataset and save the model.*

**Resources:**

- Hugging Face: Fine-tuning guide.
  - Dataset: IMDb Sentiment Dataset (Kaggle).
- 

## Week 7: Evaluation & Optimization

**Goal:** Assess and improve your LLM's performance.

**Tasks:**

1. Evaluate fine-tuned model: confusion matrix, precision/recall, overfitting checks.
2. Optimize: experiment with batch size, learning rate, or quantization for efficiency.
3. Handle edge cases: sarcasm, multilingual text, or ambiguous sentiment.

**Key Task:**

- *Test your fine-tuned model on tricky examples (e.g., sarcastic reviews) and tweak it.*

**Resources:**

- Hugging Face: Evaluation metrics guide.
  - Blog: "Optimizing Transformers" (Medium or Towards Data Science).
- 

## Week 8: Deployment & Real-World Application

**Goal:** Deploy your sentiment analysis model and wrap up with a project.

**Tasks:**

1. Build a simple API: Use Flask or FastAPI to serve your LLM.
2. Deploy to a cloud platform (e.g., Heroku, AWS, or Google Cloud).
3. Create a mini-project: e.g., a web app that analyzes user-input text sentiment.

**Key Task:**

- *Deploy your sentiment analysis API to Heroku and test it with live inputs.*

**Resources:**

- FastAPI/Flask docs.
  - Tutorial: "Deploy Hugging Face Model" (YouTube or blogs).
-

## Weekly Summary

- **Week 1:** Python & NLP basics (3 tasks).
- **Week 2:** Sentiment analysis basics (3 tasks).
- **Week 3:** ML for NLP (3 tasks).
- **Week 4:** Intro to LLMs (3 tasks).
- **Week 5:** Sentiment with LLMs (3 tasks).
- **Week 6:** Fine-tuning LLMs (3 tasks).
- **Week 7:** Evaluation & optimization (3 tasks).
- **Week 8:** Deployment & project (3 tasks).

Total: **24 tasks over 8 weeks** (~3 tasks/week).

## Tips for Success

- **Practice Daily:** Spend 1-2 hours coding or experimenting.
- **Datasets:** Use free datasets (Kaggle, Hugging Face Datasets) to test your skills.
- **Hardware:** Start with a laptop; for fine-tuning, consider Google Colab (free GPU).
- **Community:** Join NLP groups (e.g., Reddit's r/MachineLearning, Hugging Face forums).

## Final Project Idea

By Week 8, build a sentiment analysis tool that takes user reviews (e.g., from a website or X posts) and returns sentiment scores, deployed as an API or simple web app.

Let me know if you want to adjust the timeline, dive deeper into a week, or get code snippets for any task!