Next Step

Phase 1 remaining steps

1.3 Base Sentiment Analysis Models

Task 1.3.1: Model Implementation

Create `src/ml/sentiment_models.py`:

```
"""
Implement distributed ML models:
1. Naive Bayes (baseline)
2. Logistic Regression with ElasticNet
3. Random Forest (distributed)
4. Gradient Boosting (XGBoost on Spark)
5. LSTM with distributed training
6. Transformer-based models (DistilBERT)

Include:
- Model training pipelines
- Hyperparameter tuning with Spark MLlib
- Cross-validation implementation
- Model serialization and versioning
"""
```

Expansion on Step 1.3

1. Sentiment Classification Models (Critical Missing Piece)

You need to implement and compare multiple ML models:

```
# You proposed but haven't implemented yet:
- Naive Bayes classifier
- Logistic Regression
```

```
- Random Forest
- Gradient Boosting
- Deep Learning approaches
- Ensemble methods
```

2. Model Evaluation & Benchmarking

```
# Required metrics:
- Accuracy, Precision, Recall, F1-score
- Cross-validation results
- Performance comparisons between models
- Scalability benchmarks
```

3. Analytic Approaches (As per your proposal)

```
# Missing analytics:
- Temporal sentiment analysis (trends over time)
- Topic-based sentiment analysis (clustering)
- Anomaly detection in sentiment patterns
- Time series analysis
```

4. Visualizations & Reporting

```
# Required visualizations:
- Sentiment trends over time
- Topic-sentiment heatmaps
- Model performance comparisons
- Word clouds with sentiment coloring
```

5. Big Data Performance Analysis

```
# Scalability testing:
- Processing time benchmarks
- Memory usage analysis
```

```
- Comparison of different data volumes
- Spark optimization metrics
```

Quick Implementation Plan to Complete Class Project

Here's what you need to do next:

Phase 1.5: Sentiment Models (2-3 days)

Create `src/ml/sentiment_models.py`:

```
"""
This module should include:
1. BaselineNaiveBayes class
2. DistributedLogisticRegression class
3. SparkRandomForest class
4. Model evaluation pipeline
5. Cross-validation implementation
"""
```

Phase 1.6: Analytics Implementation (1-2 days)

Create `src/spark/sentiment_analytics.py`:

```
"""
Implement:
1. Temporal trend analysis
2. Topic clustering (LDA)
3. Anomaly detection
4. Sentiment aggregations
"""
```

Phase 1.7: Visualization & Reporting (1 day)

Create `src/visualization/report_generator.py`:

"""
Generate:
1. Performance comparison charts
2. Temporal sentiment visualizations
3. Topic-sentiment heatmaps
4. Final report with all findings
"""