In [1]:

```
import pandas as pd
import numpy as np

from sklearn.preprocessing import MinMaxScaler

import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
# Load the dataset
df = pd.read_csv('B:/DCU/Practicum/Proj/Dataset/main/processed/multimodal_dataset_final3.csv')
print(f"Dataset loaded: {df.shape[0]:,} rows × {df.shape[1]} columns")
```

```
Dataset loaded: 1,989 rows × 59 columns
```

In [3]:

```
intent_columns = [col for col in df.columns if '_intent' in col]
for col in intent_columns:
    print(f"Column: {col}")
    print(f"  Max: {df[col].max()}")
    print(f"  Min: {df[col].min()}")
    print()
```

```
Column: news_buying_intent
  Max: 7
  Min: 0

Column: news_selling_intent
  Max: 6
  Min: 0

Column: news_uncertainty_intent
  Max: 3
  Min: 0

Column: news_urgency_intent
  Max: 4
  Min: 0

Column: news_prediction_intent
  Max: 6
  Min: 0

Column: news_fear_intent
  Max: 6
  Min: 0

Column: news_greed_intent
  Max: 4
  Min: 0

Column: news_question_intent
  Max: 11
  Min: 0

Column: news_action_intent
  Max: 8
  Min: 0

Column: reddit_buying_intent
  Max: 7
  Min: 0

Column: reddit_selling_intent
  Max: 6
  Min: 0

Column: reddit_uncertainty_intent
  Max: 3
  Min: 0

Column: reddit_urgency_intent
  Max: 4
  Min: 0

Column: reddit_prediction_intent
  Max: 6
  Min: 0

Column: reddit_fear_intent
  Max: 6
  Min: 0

Column: reddit_greed_intent
```

```
  Max: 4
  Min: 0

Column: reddit_question_intent
  Max: 11
  Min: 0

Column: reddit_action_intent
  Max: 8
  Min: 0

Column: total_buying_intent
  Max: 14
  Min: 0

Column: total_selling_intent
  Max: 12
  Min: 0

Column: total_uncertainty_intent
  Max: 6
  Min: 0

Column: total_urgency_intent
  Max: 8
  Min: 0

Column: total_prediction_intent
  Max: 12
  Min: 0

Column: total_fear_intent
  Max: 12
  Min: 0

Column: total_greed_intent
  Max: 8
  Min: 0

Column: total_question_intent
  Max: 22
  Min: 0

Column: total_action_intent
  Max: 16
  Min: 0
```

In [4]:

```
sentiment_columns = [col for col in df.columns if 'sentiment_minus' in col]
for col in sentiment_columns:
    print(f"Column: {col}")
    print(f"  Max: {df[col].max()}")
    print(f"  Min: {df[col].min()}")
    print()
```

```
Column: sentiment_minus_uncertainty
  Max: 0.8709523916244506
  Min: -6.0

Column: sentiment_minus_fear
  Max: 0.8709523916244506
  Min: -12.0

Column: sentiment_minus_action
  Max: 0.1421094663441181
  Min: -16.0

Column: sentiment_minus_urgency
  Max: 0.8446768168359995
  Min: -8.0

Column: sentiment_minus_prediction
  Max: 0.811383741348982
  Min: -12.0
```

In [5]:

```
# Since the min-max values of sentiment and intent were a mismatch, they had to be normalized
# before feeding them to the models
```

In [6]:

```python
# Normalize all intent columns
intent_columns = [col for col in df.columns if col.endswith('_intent')]
print(f"Normalizing {len(intent_columns)} intent columns")

scaler = MinMaxScaler(feature_range=(0, 1))
df[intent_columns] = scaler.fit_transform(df[intent_columns])
print("Intent columns normalized to [0,1] range")

# Drop unnecessary columns
text_columns = ['combined_news', 'Combined_Reddit_News']
label_columns = [col for col in df.columns if col.endswith('_label')]
columns_to_drop = text_columns + label_columns
existing_columns_to_drop = [col for col in columns_to_drop if col in df.columns]

df = df.drop(columns=existing_columns_to_drop)
print(f"Dropped {len(existing_columns_to_drop)} columns")

# Create new prediction targets
df['Next_3_Close'] = df['Close'].shift(-3)
df['Next_7_Close'] = df['Close'].shift(-7)
print("Created Next_3_Close and Next_7_Close targets")
# ^ These prediction targets were dropped in the next versions of the code as
# they were just shifted versions of existing columns whinch introduced data leakage.
# Recompute derived sentiment-intent features
df['finbert_final_sentiment'] = 0.6 * df['FinBERT_news_sentiment'] + 0.4 * df['FinBERT_reddit_sentiment']

df['sentiment_minus_fear'] = df['finbert_final_sentiment'] - df['news_fear_intent'] - df['reddit_fear_intent']
df['sentiment_minus_uncertainty'] = df['finbert_final_sentiment'] - df['news_uncertainty_intent'] - df['reddit_uncertainty_intent']
df['sentiment_minus_urgency'] = df['finbert_final_sentiment'] - df['news_urgency_intent'] - df['reddit_urgency_intent']
df['sentiment_minus_prediction'] = df['finbert_final_sentiment'] - df['news_prediction_intent'] - df['reddit_prediction_intent']
df['sentiment_minus_action'] = df['finbert_final_sentiment'] - df['news_action_intent'] - df['reddit_action_intent']
print("Recomputed all derived sentiment-intent features")
```

```
Normalizing 27 intent columns
Intent columns normalized to [0,1] range
Dropped 6 columns
Created Next_3_Close and Next_7_Close targets
Recomputed all derived sentiment-intent features
```

In [ ]:

```python
# Save the dataset
df.to_csv('multimodal_dataset_final4.1.csv', index=False)
print(f"Dataset v4.1 saved: {df.shape[0]:,} rows × {df.shape[1]} columns")

# Quick validation
intent_cols = [col for col in df.columns if col.endswith('_intent')]
print(f"Intent columns range: [{df[intent_cols].min().min():.3f}, {df[intent_cols].max().max():.3f}]")
```

```
Dataset v4.1 saved: 1,989 rows × 55 columns
Intent columns range: [0.000, 1.000]
```