ICICI INTERNSHIP
Financial News Sentiment Analyzer

WEEK 2 REPORT

Report on Data Cleaning and Exploratory Data Analysis (EDA)

I. Data Cleaning Techniques

Data cleaning is essential to remove noise, inconsistencies, and irrelevant information from the dataset. For textual data like financial news, cleaning ensures better feature extraction and accurate sentiment prediction.

1. Lowercasing

Convert all text to lowercase to avoid treating words like "Market" and "market" differently.

2. Removing Punctuation

Strip punctuation marks (e.g., `.`, `!`, `?`, `,`) to retain only words and numbers that contribute to sentiment.

3. Stop Word Removal

Eliminate common words (like *is, the, in*) that do not add value to sentiment or topic modeling.

4. Tokenization

Split text into individual words or tokens, preparing the data for lemmatization or vectorization.

5. Lemmatization

Reduce words to their root form based on their dictionary meaning (e.g., *running → run*, *better → good*), which helps unify word forms.

6. Stemming (Optional)

Trim words to their base stem (e.g., *running → run*), though less accurate than lemmatization. Used for speed in larger corpora.

7. Removing Numbers

Optional in financial sentiment – depending on context, you may retain numbers if they influence sentiment (*stock up 10%*).

8. Removing Extra Whitespace

Strip unnecessary spaces or newline characters that may result from prior cleaning steps.

9. Handling Contractions

Expand contractions (*don't → do not*, *can't → cannot*) to maintain clarity and improve NLP pipeline results.

10. Special Character and HTML Tag Removal

Remove non-alphanumeric characters, symbols, and leftover HTML tags from web-scraped datasets.

11. Named Entity Recognition (NER) Filtering (Optional)

Use NER to extract or remove names of organizations, companies, or locations depending on the task requirements.

12. Spelling Correction (Optional)

Apply algorithms like SymSpell or TextBlob to fix typos that might mislead sentiment scoring.

13. Duplicate and Null Entry Removal

Remove repeated articles and entries with empty or meaningless content.

14. Language Detection and Filtering

Remove non-English articles if working only with English sentiment models like BERT.

15. Custom Stopwords

Remove financial-specific irrelevant terms like *breaking*, *news*, *update* if they don't impact sentiment.

II. Exploratory Data Analysis (EDA) Techniques

EDA is used to understand the structure, patterns, and relationships in the data. Here are one-line summaries of various EDA techniques tailored for financial news text data.

A. Data Structure and Overview

- **Shape and Info:** Understand dataset size, types, and missing values (`df.shape`, `df.info()`).
- **Null Values Check:** Identify missing entries using `df.isnull().sum()`.

B. Textual Feature Exploration

- **Word Count & Sentence Length Distribution:** Assess verbosity and identify outliers.
- **Most Frequent Words:** Detect dominant financial keywords using `Counter()` or `FreqDist`.
- **N-gram Analysis:** Explore common bigrams and trigrams (e.g., *stock market*, *interest rates*).
- **Word Cloud:** Visual representation of top words for qualitative insight.

C. Class Imbalance Check (if labeled)

- **Target Distribution Plot:** Check balance in sentiment classes using bar plots or pie charts.

D. Sentiment Score Exploration

- **Sentiment Score Histogram:** If using pre-generated sentiment scores, visualize their spread.
- **Mean Sentiment by Source/Date:** Temporal or publisher-wise sentiment trends.

E. Correlation & Grouping

- **Group By Date or Company:** Analyze trends in sentiment by time or entity.

- **Correlation of Sentiment with External Metrics:** Link sentiment with stock returns or volatility.

F. Text Embedding Visualization

- **t-SNE / PCA of Embeddings:** Visualize clusters in BERT or TF-IDF embeddings of articles.

G. Time Series Visualization

- **Sentiment Over Time:** Line chart to show sentiment trends across days, weeks, or months.
- **Event Overlay:** Mark key financial events (e.g., earnings, Fed announcements) on sentiment timeline.

H. Named Entity Distribution

- **Top Entities Mentioned:** Count companies, organizations, and locations mentioned across headlines.

I. Outlier Detection

- **Box Plot on Text Length:** Spot unusually long or short articles.
- **Sentiment Outliers:** Articles with extreme positive or negative values.

J. Word Embedding Similarity (Advanced)

- **Cosine Similarity:** Measure similarity between news headlines to detect repeated or copied content.

Libraries we can use:

NLTK (Natural Language Toolkit)

NLTK is one of the earliest and most widely-used libraries for text processing in Python. It provides a rich set of tools for tokenization, stopword removal, stemming, lemmatization, and basic part-of-speech tagging. It also includes sentiment analysis features such as VADER and TextBlob integration. Although not optimized for large-scale industrial applications, NLTK is an excellent tool for prototyping and educational purposes, offering a deep understanding of the underlying NLP techniques.

spaCy

spaCy is a modern and high-performance NLP library designed for practical, real-world use. It supports advanced linguistic features such as part-of-speech tagging, dependency parsing, named entity recognition (NER), and efficient lemmatization. Unlike NLTK, spaCy is built for speed and scalability, making it ideal for large datasets like financial news corpora. While it does not include built-in sentiment analysis, it excels in text preprocessing and can be effectively combined with sentiment tools like VADER or custom models for a complete sentiment analysis pipeline.

Different techniques we can use:

N-grams Analysis

N-grams analysis is a fundamental technique in Natural Language Processing (NLP) used to identify and analyze contiguous sequences of *n* words (or tokens) from a text corpus. This method helps in capturing local linguistic context and is instrumental in understanding how words combine to form meaningful expressions. For instance, in financial news sentiment analysis, while a single word like "rise" may indicate positivity, a bigram such as "interest hike" or a trigram like "stock market crash" provides more nuanced, domain-specific insights into sentiment.

During n-gram analysis, unigrams (n=1) reveal the most frequent individual words, bigrams (n=2) and trigrams (n=3) help uncover common word pairings or triplets that may hold sentiment-heavy meanings. Analysts often use this method to find and quantify patterns, trends, or keywords that recur across news articles. This aids in feature engineering, sentiment scoring, and building predictive models.

Libraries like NLTK, spaCy, and scikit-learn provide robust tools to generate n-grams using vectorizers such as `CountVectorizer` and `TfidfVectorizer`. These can be analyzed further through frequency distributions or visualizations like word clouds, bar plots, or network graphs. Overall, n-grams analysis is critical in financial sentiment mining, helping bridge the gap between raw text and quantitative analysis.

VADER (Valence Aware Dictionary and sEntiment Reasoner)

VADER is a rule-based sentiment analysis tool specifically designed for social media and short text data. It is part of the NLTK library and provides polarity scores across four categories: positive, negative, neutral, and compound (a normalized, weighted composite score). VADER is particularly useful for its ability to handle emojis, slangs, and punctuation-based emphasis (e.g., exclamation marks), making it well-suited for financial headlines or news snippets. Its ease of use and minimal preprocessing requirements make it a popular choice for quick, interpretable sentiment analysis.