

A

Project Report

On

**“****Index Movement Prediction   
via  
News Sentiment Analysis”**

DATA SCIENCE AT PRATICE

(August 2024 Term)  
  
  
  
**Submitted By:**

**Trend Tracker**

**Authors and Email Addresses:**

Brijgopal Bharadwaj: brijgopalb@iisc.ac.in  
Chandan Kumar Rao chandankuma4@iisc.ac.in

Danish Akhlaq: danishakhlaq@iisc.ac.in

Department of Computer Science   
Indian Institute of Science Bangalore

Project Supervisor  
Pandarasamy Arjunan

**Introduction**

The news media plays a pivotal role in shaping public opinion, including investor sentiment. This influence can significantly impact stock prices and market trends. By leveraging the power of data science, we aim to quantify and analyse the impact of news media on stock market movements.

**Problem Statement**

While news media is a valuable source of information for investors, it can also be a tool for manipulating public perception and influencing investment decisions. This dynamic presents a significant challenge for investors seeking to make informed decisions.

**Why is it important?**

* + This dynamic provides us a metric that can be utilized in forecasting and market-movement prediction models.
  + Knowledge about these influences might help the common investor be more perceptive to the way these influences can creep into their investment patterns.
  + An extremely biased media-source might be a candidate for further investigation by the regulatory authorities like SEBI to uncover undue practices/agendas.

**Objectives of the project**

* + We want to develop a machine-learning pipeline that enables real-time sentiment analysis of incoming news articles, and computation of correlation scores that feed into a forecasting model for near-term market movement prediction.

**How can Data Science solve the problem?**

* + Build a market-news bias-score assigning model, using BERT embeddings and DNN regressor.
  + Use the model to assign bias scores to news in real-time, and compute correlation scores with the index movements on the fly.
  + Use the insights gathered in various time-windows to forecast market trends in a limited capacity.

**Project Flow**

**A diagram of a company

Description automatically generated**

**Dataset Overview**

The dataset consists of two main components:

**News Articles Dataset:**

* + Contains news articles from various sources related to companies and the stock market.
  + The datasets have been generated by fetching news articles from the NewsAPI and web scrapping (Google Custom Search API).
  + Features include Author, Title, Description, URL, UrlToImage, Source, Content and Published date.

**Market Data Dataset:**

* + This datasets have been extracted using yahoo finance.
  + Contains historical stock market data, including Open, High, Low, Close, Volume, and Adjusted Close prices.
  + Features include Date, Ticker, and Company Name.

**Data Preparation and Preprocessing**

Processing a raw dataset of news articles to clean, structure, and enrich the data with sentiment analysis metrics, aimed at facilitating advanced insights.

Below are the key steps and details:

**Data Loading and Preprocessing**:

* Loaded a dataset of news articles using pandas.
* Extracted relevant information from structured fields like source and publishedAt.
* Converted publishedAt timestamps to a standardized datetime format and removed microseconds.

**Data Cleaning**:

* Removed duplicate rows and retained essential columns: source, author, title, description, publishedAt, URL, and content.
* Renamed columns for clarity and adjusted the companyName field by truncating redundant words.
* Extracted base URLs from article links for easier source identification.
* Removed rows with null values to ensure data consistency.

**Sentiment Analysis**:

* Integrated the VADER (Valence Aware Dictionary and sEntiment Reasoner) sentiment analysis tool.
* Calculated sentiment scores for article content (summary\_vader) and descriptions (description\_vader), providing compound sentiment scores to identify overall tone.

**Outputs and Insights**:

* The processed dataset includes cleaned and structured information, along with sentiment metrics, enabling further analysis of media sentiment across different sources and topics.

A screenshot of a black screen

Description automatically generated

**Exploratory Data Analysis (EDA)**

This structured analysis facilitates a deeper understanding of news content and its evolution, helping to identify patterns and trends that could influence further decision-making or predictive modelling.

Analyses datasets of preprocessed company news articles, focusing on content length, sentiment trends, and temporal variations.

Below is an organized summary of the key steps:

**Dataset Overview**

* Loading Data: The dataset is loaded into a Pandas DataFrame from a CSV file. The publishedAt column is parsed as datetime to facilitate temporal analysis.
* Initial Exploration:
  + head() is used to inspect the first few rows for data structure and validation.
  + shape reveals the number of rows (articles) and columns (attributes).
  + describe() provides a statistical summary of numerical columns, identifying potential outliers or trends.
* Null Value Check: The count of missing values in each column is assessed to identify any data quality issues.

**Text Length Analysis**

* Description Length:
  + A histogram displays the distribution of text lengths in the description column.
  + It helps assess whether article descriptions are consistently detailed or vary significantly.
* Summary Length:
  + A similar histogram is plotted for the summary column (preprocessed summaries using BART).
  + It highlights the variability in the summaries' text lengths.

**Trend Analysis**

* Temporal Variations in Description Length:
  + The code visualizes how the length of article descriptions has changed over time.
  + This trend can reveal periods of more detailed reporting or content changes.
* Description Sentiment (VADER Score):
  + Sentiment scores (description\_vader) are plotted over time.
  + This provides insights into how the tone of reporting (positive, neutral, or negative) evolved.
* Summary Length Over Time:
  + Similar to descriptions, the summary lengths are analysed to track variations over time.
* Summary Sentiment (VADER Score):
  + Sentiment scores for summaries (summary\_vader) are plotted over time, providing an alternative perspective on sentiment trends.

**Visualizations**

* Each trend is plotted using line graphs, with the x-axis representing dates (publishedAt) and the y-axis representing respective metrics (lengths or sentiment scores).
* The graphs feature clear labels and rotated x-axis ticks for readability.

Histogram distribution of the lengths of the text Histogram of "BART summary length"

A graph of a number of blue bars

Description automatically generated A blue graph with numbers

Description automatically generated

Class-frequency Distribution

A graph of blue bars

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated

Trend in the lengths of article descriptions over time

A graph showing a number of data

Description automatically generated

Trend of sentiment analysis scores (summary\_vader) over time

A graph showing a number of blue lines

Description automatically generated with medium confidence

**Model and Prediction**

**Market Sentiment Classification:**

* + Used BERT embeddings and DNN classifier to assign bearish / neutral / bullish labels to news articles.
  + Achieved an accuracy of 90% on the test set.

**Market Price Forecasting (Regression):**

* + Used xgboost to predict the next-day market movement based on news sentiment scores and technical parameters of previous days.
  + Achieved an RMSE score of 138.79 on the test set.

**Hyperparameter Tuning**

Using hyperparameter tuning to optimize the performance of an XGBoost regression model for predicting stock prices, specifically the "Close" price. Below is a summary of the main steps:

**Feature Selection**

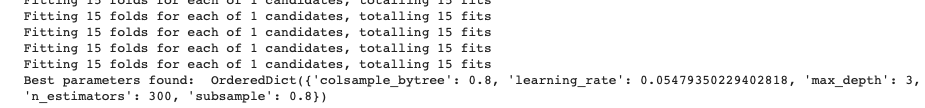
* The model excludes non-predictive columns (High, Low, Close, Daily\_Change, Daily\_change\_percent) from the feature set.
* The target variable is set as the "Close" price.

**Data Preparation**

* Training (X\_train, y\_train) and testing (X\_test, y\_test) datasets are split from the main dataset.

**Hyperparameter Tuning with Bayesian Optimization**

* A base XGBRegressor model is initialized.
* A parameter grid is defined for tuning key hyperparameters like:
  + n\_estimators: Number of trees.
  + learning\_rate: Learning rate for gradient descent.
  + max\_depth: Maximum depth of trees.
  + subsample: Fraction of samples used for training each tree.
  + colsample\_bytree: Fraction of features considered at each split.
* BayesSearchCV is used for tuning the hyperparameters, optimizing for negative mean squared error (neg\_mean\_squared\_error) using 15-fold cross-validation.
* The best hyperparameters are identified and printed as below.



**Model Training**

* The model is re-trained with the best hyperparameters and early stopping rounds to prevent overfitting.
* Both the training and testing datasets are evaluated during training.

**Prediction and Visualization**

* The predictions are merged with the dataset for comparison.
* A line plot shows the true values and model predictions, with a vertical line marking the train-test split.

**A graph showing a line going up

Description automatically generated**

**Evaluation Metrics**

**Model 1 – Using BERT Market News Sentiment Labels**

|  |  |
| --- | --- |
| **Performance Metrics** | **Values** |
| **Accuracy** | 96.91% |
| **Precision** | 96.94% |
| **Recall** | 96.91% |
| **F1 Score** | 96.91% |

**A graph with a line

Description automatically generated** **A yellow and purple squares with numbers

Description automatically generated**

**Model 2 – Market-Price Movement Prediction**

|  |  |
| --- | --- |
| **Model Performance Metrics** | **Model Performance Summary** |
| **Accuracy** | **90%** |
| **R2 score** | **0.7835** |
| **RMSE Value** | **138.79** |

**A screen shot of a graph

Description automatically generated**

**Future Work**

* Compare BERT performance with other text-embedding models like Word2Vec, etc.
* Assimilate dataset for a longer time-window to obtain higher performance.