Coded Project: Natural Language Processing with Generative AI

Stock Market News Sentiment Analysis and Summarization

Submitted by – Yash Juneja

Submitted to – Great Learning

Great Learning

POWER AHEAD

Table of Contents

➢ Context

The prices of the stocks of companies listed under a global exchange are influenced by a variety of factors, with the company's financial performance, innovations, collaborations, and market sentiment being factors that play a significant role. News and media reports can rapidly affect investor perceptions and, consequently, stock prices in the highly competitive financial industry. With the sheer volume of news and opinions from a wide variety of sources, investors and financial analysts often struggle to stay updated and accurately interpret their impact on the market. As a result, investment firms need sophisticated tools to analyze market sentiment and integrate this information into their investment strategies.

➢ Objective

With an ever-rising number of news articles and opinions, an investment startup aims to leverage artificial intelligence to address the challenge of interpreting stock-related news and its impact on stock prices. They have collected historical daily news for a specific company listed under NASDAQ, along with data on its daily stock price and trade volumes.

As a member of the Data Science and AI team in the startup, you have been tasked with analyzing the data, developing an AI-driven sentiment analysis system that will automatically process and analyze news articles to gauge market sentiment, and summarizing the news at a weekly level to enhance the accuracy of their stock price predictions and optimize investment strategies. This will empower their financial analysts with actionable insights, leading to more informed investment decisions and improved client outcomes.

➢ Data Dictionary

- Date: The date the news was released
- News: The content of news articles that could potentially affect the company's stock price
- Open: The stock price (in $) at the beginning of the day
- High: The highest stock price (in $) reached during the day
- Low: The lowest stock price (in $) reached during the day
- Close: The adjusted stock price (in $) at the end of the day
- Volume: The number of shares traded during the day
- Label: The sentiment polarity of the news content
  1: Positive
  0: Neutral
  -1: Negative

Data Overview

- **First five rows of Data**

	Date	News	Open	High	Low	Close	Volume	Label
0	2019-01-02	The tech sector experienced a significant decline in the aftermarket following Apple's Q1 revenue warning. Notable suppliers, including Skyworks, Broadcom, Lumentum, Qorvo, and TSMC, saw their stocks drop in response to Apple's downward revision of its revenue expectations for the quarter, previously announced in January.	41.740002	42.244999	41.482498	40.246914	130672400	-1
1	2019-01-02	Apple lowered its fiscal Q1 revenue guidance to $84 billion from earlier estimates of $89-$93 billion due to weaker than expected iPhone sales. The announcement caused a significant drop in Apple's stock price and negatively impacted related suppliers, leading to broader market declines for tech indices such as Nasdaq 10	41.740002	42.244999	41.482498	40.246914	130672400	-1
2	2019-01-02	Apple cut its fiscal first quarter revenue forecast from $89-$93 billion to $84 billion due to weaker demand in China and fewer iPhone upgrades. CEO Tim Cook also mentioned constrained sales of Airpods and Macbooks. Apple's shares fell 8.5% in post market trading, while Asian suppliers like Hon	41.740002	42.244999	41.482498	40.246914	130672400	-1
3	2019-01-02	This news article reports that yields on long-dated U.S. Treasury securities hit their lowest levels in nearly a year on January 2, 2019, due to concerns about the health of the global economy following weak economic data from China and Europe, as well as the partial U.S. government shutdown. Apple	41.740002	42.244999	41.482498	40.246914	130672400	-1
4	2019-01-02	Apple's revenue warning led to a decline in USD JPY pair and a gain in Japanese yen, as investors sought safety in the highly liquid currency. Apple's underperformance in Q1, with forecasted revenue of $84 billion compared to analyst expectations of $91.5 billion, triggered risk aversion mood in markets	41.740002	42.244999	41.482498	40.246914	130672400	-1

- **Checking an article**

```
' This news article reports that yields on long-dated U.S. Treasury securities hit their lowest levels in nearly a year on January 2, 2019, due to concerns about the health of the g
lobal economy following weak economic data from China and Europe, as well as the partial U.S. government shutdown. Apple'
```

- **Shape of Dataset**

```
(349, 8)
```

- **Checking for Missing values**

	0
Date	0
News	0
Open	0
High	0
Low	0
Close	0
Volume	0
Label	0

dtype: int64

- There are no mising values in the data

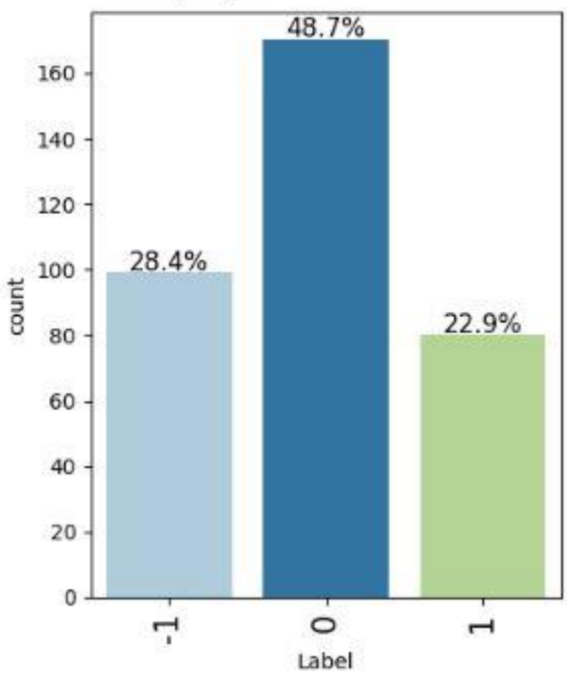
- **Statistical Summary**

	Open	High	Low	Close	Volume	Label
count	349.000000	349.000000	349.000000	349.000000	3.490000e+02	349.000000
mean	46.229233	46.700458	45.745394	44.926317	1.289482e+08	-0.054441
std	6.442817	6.507321	6.391976	6.398338	4.317031e+07	0.715119
min	37.567501	37.817501	37.305000	36.254131	4.544800e+07	-1.000000
25%	41.740002	42.244999	41.482498	40.246914	1.032720e+08	-1.000000
50%	45.974998	46.025002	45.639999	44.596924	1.156272e+08	0.000000
75%	50.707500	50.849998	49.777500	49.110790	1.511252e+08	0.000000
max	66.817497	67.062500	65.862503	64.805229	2.444392e+08	1.000000

Exploratory Data Analysis(EDA)

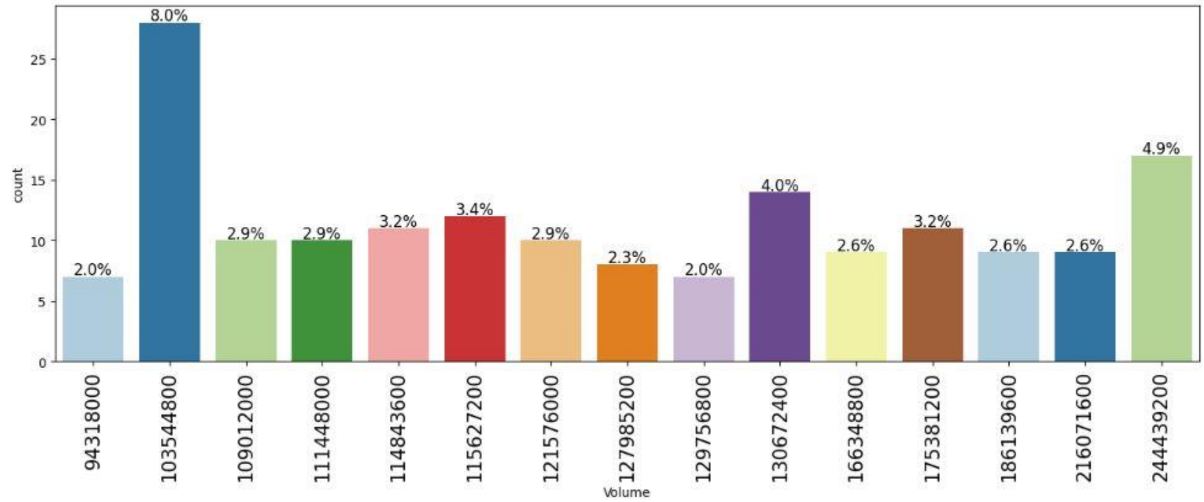
Univariate Analysis

Observation By Label



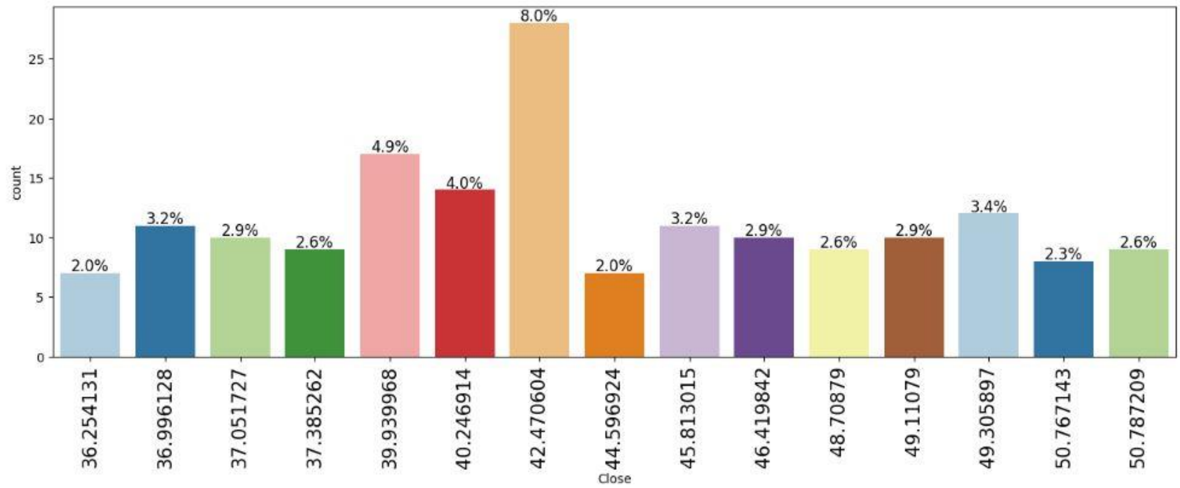
- Label **0** is the most frequent class, accounting for **48.7%** of the data, This indicates that nearly **half** of the dataset belongs to this category.
- Label **1** is the least represented, with only **22.9%** of the total, This suggests a **class imbalance**, which might impact model performance if not addressed.

Observation By Volume



- **Volume = 103,544,800** appears most frequently, with a **count percentage of 8.0%**, This volume clearly stands out and may represent a common trading volume or threshold in the dataset.
- **Volume = 24,443,920** shows a significant count of **4.9%**, second highest among the rest.
- **Volumes = 94,318,000 and 129,756,800** are the least frequent, each with only **2.0%** count, These could be considered outliers or rare cases in the volume distribution.

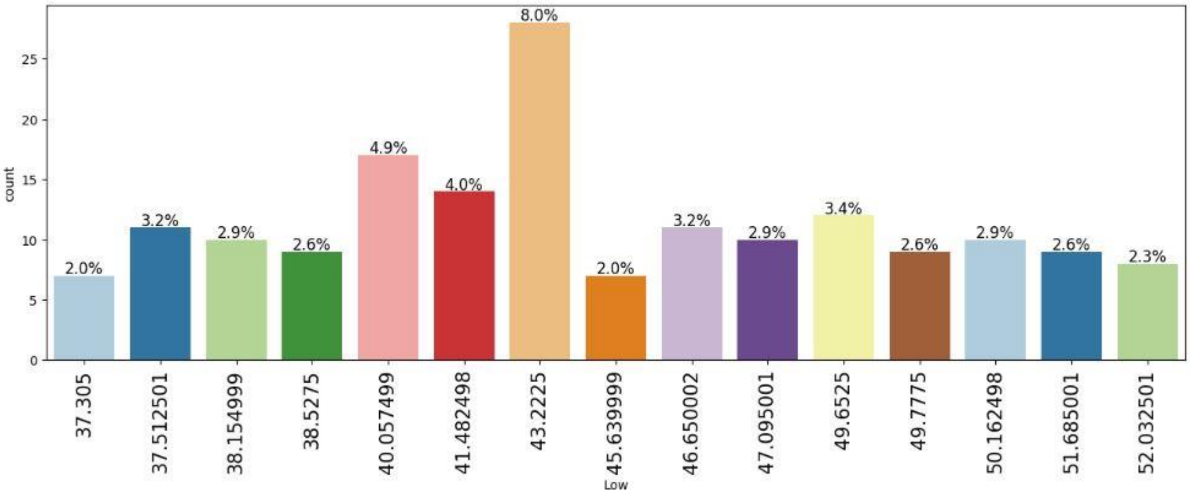
Observation By Close



- **42.470604** is the most frequently occurring value, with a **count percentage of 8.0%**, This indicates that this specific close price is a **dominant mode** in the dataset.
- **39.399968** accounts for **4.9%** of the records, **40.246914** contributes **4.0%**, **49.05897** stands out among the higher close prices with **3.4%**, These could represent common price levels or psychological support/resistance zones in trading.

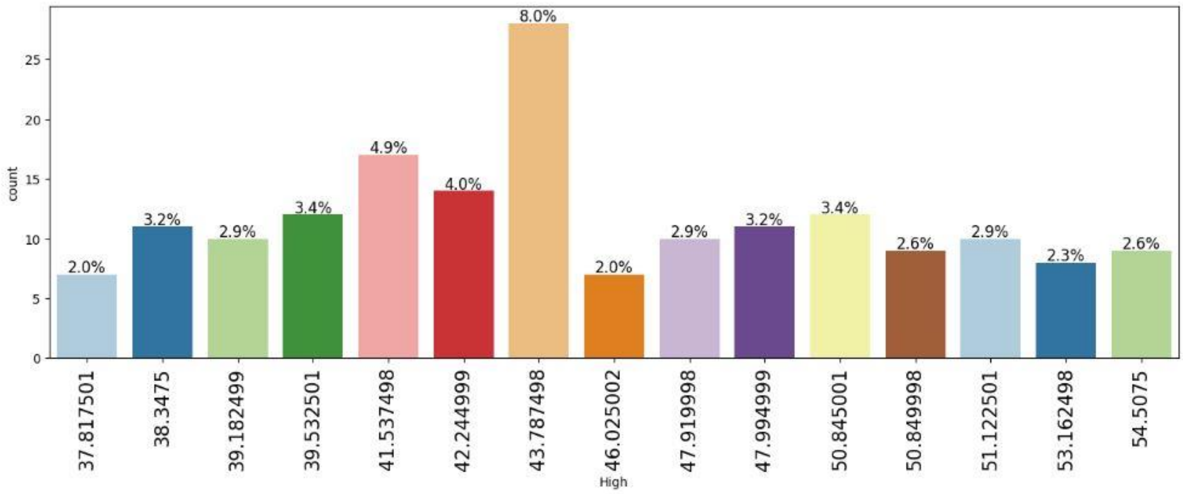
- **36.254131 and 44.596924** both have the lowest count, at **2.0%**, Indicates they are relatively rare occurrences in the dataset.

Observation By Low



- The distribution appears to be **fairly uniform with a mild central peak**, but not strongly skewed.
- The bin centered at **43.2225** has the **highest count (8.0%)**, making it the **mode** of this distribution.
- The lowest frequency bins are around **37.305** and **45.639999**, both with **2.0%** of the data.
- Several bins have very close percentages (~2.6% to 3.4%), suggesting a relatively **even spread** in the data apart from the peak.

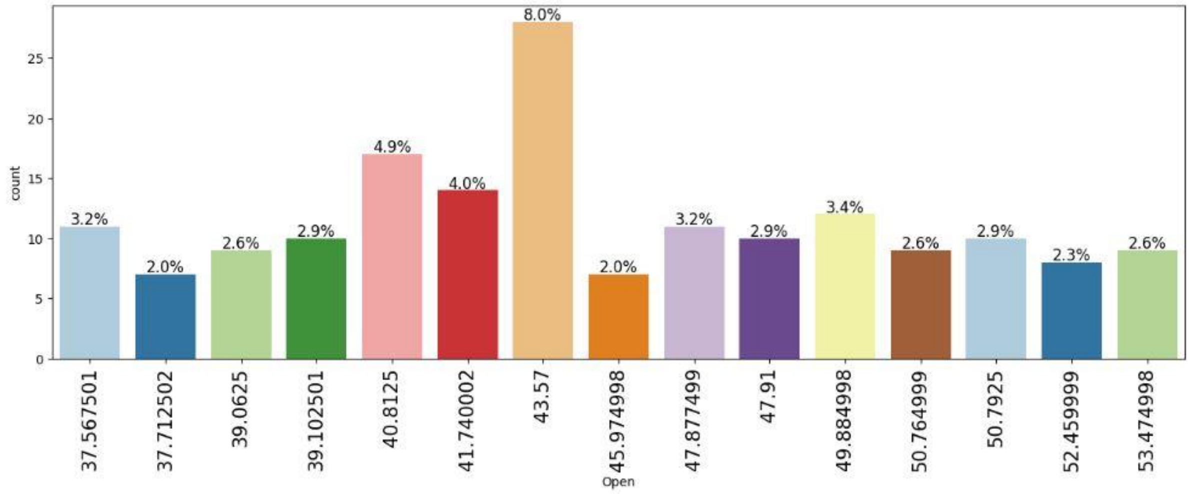
Observation By High



- The bin centered at **43.78** has the **highest count: 8.0%**, indicating a significant cluster in this range, This mirrors the pattern seen in the "Low" variable, suggesting a central tendency around the **mid-40s.**

- Bins around **37.81** and **46.02** have the **lowest count: 2.0%**.
- Other bins mostly hover between **2.3% and 4.9%**, showing a **relatively even spread** across the rest of the data.

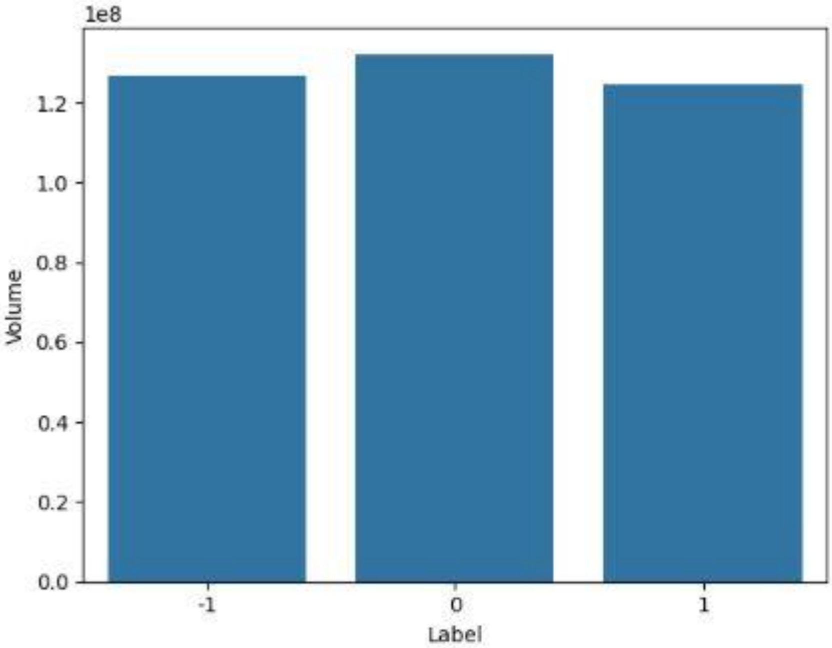
Observation By Open



- The bin centered at **43.57** is the **most frequent**, with **8.0%**, forming a clear **central peak**.
- The bins around **37.12** and **45.97** have the **lowest counts (2.0%)**, showing less concentration at the extremes.
- Most bins have values between **2.6% and 4.9%**, indicating a **relatively balanced distribution** outside the central mode, Slight secondary bumps are seen around **40.81 (4.9%)** and **41.74 (4.0%)**, suggesting **additional clustering near early 40s**.
- The **central peak** around **43–44** is consistent across the "Low," "High," and now "Open" values.

Bivariate Analysis

Analysis Between Label and Volume



- **Label Categories:**
  - The x-axis has 3 label categories: **-1**, **0**, and **1**.
  - These likely represent **class labels**, such as:
    - -1: Decrease
    - 0: No Change
    - 1: Increase

    (Common in financial or sentiment classification tasks.)
- **Volume Comparison:**
  - All three labels have **similar total volume**, each slightly above **1.2×10^8** (120 million).
  - Label **0** has the **highest volume**, followed closely by **-1** and **1**.
- **Distribution Insight:**
  - The differences in total volume across labels are **small**, indicating a **balanced distribution** in terms of volume.
  - This suggests that the dataset is **not biased** toward any particular label class in terms of activity or volume.

Word Embeddings

Word2Vec

Checking the word embedding of a random word
Word = dollar

```
array([-0.01160005,  0.03309643,  0.00843227,  0.00862154, -0.00157797,
       -0.03477321,  0.01982657,  0.08566632,  0.00631896, -0.0224258 ,
       -0.00050059, -0.03439614,  0.00541724,  0.00110591, -0.03541142,
       -0.03244602,  0.02114421, -0.00240191,  0.00342457, -0.01836854,
       -0.01719614, -0.00038516,  0.03956924,  0.00897308,  0.02510742,
        0.00462799, -0.04512137,  0.01096504, -0.01588242, -0.03877683,
        0.00361654, -0.01059768,  0.01327924, -0.00584798, -0.00162225,
        0.0161751 ,  0.02754188, -0.04673001, -0.00333076,  0.00417731,
       -0.02130849,  0.00339934,  0.00347044, -0.03030657,  0.01202709,
        0.02833323,  0.01227715,  0.01260793, -0.00383292,  0.03171312,
        0.01285637,  0.00953357, -0.01669286,  0.00578205, -0.00476097,
        0.04235971,  0.01687318,  0.00366087,  0.01755581, -0.00531486,
       -0.01644323, -0.00194878,  0.00580083,  0.00515301,  0.00908976,
        0.01904951,  0.0124515 ,  0.01636902, -0.02243076, -0.00960422,
        0.01399478,  0.0209788 ,  0.03527352, -0.02449606,  0.01081234,
        0.02089564, -0.03682912, -0.00015468, -0.01633214,  0.03736717,
       -0.01810447, -0.02932467,  0.00463105,  0.07893   ,  0.00793823,
        0.00354264, -0.01288357,  0.00403654,  0.03856393,  0.01966742,
        0.04264942, -0.02193782,  0.01365503, -0.00712975,  0.04443873,
        0.03838953,  0.03525066, -0.01937878, -0.01658424,  0.02421484,
       -0.0111452 ,  0.00729256,  0.03055885,  0.01213488,  0.00552866,
       -0.02168705, -0.01113334,  0.01297957, -0.03079433,  0.01361899,
       -0.04861066, -0.01008474, -0.00532893,  0.02758723,  0.01662606,
        0.00831979, -0.00652108, -0.00039696,  0.03887612, -0.05587789,
        0.01427442,  0.02675554,  0.02538523,  0.01113017, -0.01190301,
        0.02299149,  0.00871323, -0.0302745 , -0.00090605,  0.03712593,
        0.01760285,  0.0418956 ,  0.0167044 , -0.04138957,  0.02356203,
        0.02690743, -0.01086473, -0.00616983, -0.02917671, -0.03906968,
        0.008248  , -0.04637256, -0.00888064,  0.03437086,  0.02099262,
       -0.01827071, -0.03947747, -0.0121742 ,  0.02538168, -0.0238926 ,
        0.00835777, -0.05905897, -0.01985228, -0.02503984,  0.00178501,
        0.01678563, -0.03217055, -0.03466205, -0.00660355,  0.04342324,
        0.00043837,  0.02473697, -0.04189229,  0.02960947, -0.0246694 ,
        0.01155568,  0.01231394,  0.00894805,  0.00655414,  0.0666279 ,
```

Checking the word embedding of a random word

Word = stock

```
array([-0.05655076,  0.15313485,  0.01890815,  0.04005078, -0.00474626,
       -0.1465348 ,  0.08601524,  0.36358824,  0.04075735, -0.07975269,
        0.00444322, -0.13479765,  0.03142296, -0.01006838, -0.13601696,
       -0.14272574,  0.08917442, -0.00798   ,  0.01642925, -0.0655942 ,
       -0.08455348, -0.0044673 ,  0.15189537,  0.03458521,  0.11272923,
        0.02047744, -0.19522414,  0.05579352, -0.07157966, -0.14993092,
        0.01517998, -0.03966275,  0.0492692 , -0.0320668 , -0.0140278 ,
        0.05770783,  0.10464296, -0.1887082 , -0.01242589,  0.01666808,
       -0.08403815,  0.0064767 ,  0.02534694, -0.1256232 ,  0.06537204,
        0.10852744,  0.05146201,  0.06258015, -0.00582444,  0.13383117,
        0.04256772,  0.02829058, -0.07531045,  0.01812266, -0.01235038,
        0.17728616,  0.0792874 ,  0.01023365,  0.06401072, -0.01097512,
       -0.07868642, -0.01551002,  0.03163277,  0.00885843,  0.03865562,
        0.07572882,  0.04342817,  0.0757859 , -0.08387769, -0.05321453,
        0.05103685,  0.08863927,  0.1369754 , -0.10436361,  0.05016304,
        0.10161161, -0.15298083,  0.00417052, -0.06286266,  0.14988747,
       -0.08334564, -0.12088629,  0.02211585,  0.31617403,  0.04650746,
        0.01943113, -0.04948043,  0.02858717,  0.14978294,  0.07510003,
        0.16366263, -0.08201971,  0.06147584, -0.02556738,  0.19196846,
        0.15138547,  0.14582752, -0.07096351, -0.07836607,  0.11239102,
       -0.04423726,  0.03421145,  0.12848026,  0.04101383,  0.02938208,
       -0.09903602, -0.02998489,  0.04469978, -0.13918483,  0.04412201,
       -0.2098056 , -0.04096732, -0.01250614,  0.10687751,  0.05916873,
        0.04076817, -0.03033565,  0.00141176,  0.15423396, -0.2305044 ,
        0.06861804,  0.12429763,  0.09680361,  0.05541208, -0.05694693,
        0.0914269 ,  0.03548347, -0.12628095, -0.00296999,  0.14283113,
        0.08103956,  0.17518672,  0.06184289, -0.16549495,  0.09051315,
        0.09844942, -0.04011106, -0.04014803, -0.13443708, -0.17492391,
        0.02615773, -0.19277291, -0.04643817,  0.13841803,  0.09205195,
       -0.06785676, -0.17268856, -0.06541016,  0.10323858, -0.09890305,
        0.03175946, -0.2487244 , -0.09380116, -0.09009379,  0.00906619,
        0.07747174, -0.12289613, -0.15377499, -0.02374765,  0.17373458,
```

Dataframe of vectorized documents

	Feature 0	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	Feature 8	Feature 9	...	Feature 290	Feature 291	Feature 292	Feature 293	Feature 294	Feature 295	Feature 296	Feature 297	Feature 298	Feature 299
0	-0.060515	0.155180	0.022177	0.041820	-0.003667	-0.154763	0.086327	0.377890	0.040284	-0.085529	...	0.061560	0.200018	0.131820	0.053529	0.217029	0.196743	0.022705	-0.110515	0.128729	-0.066052
1	-0.061045	0.158187	0.022491	0.042336	-0.003679	-0.157323	0.087791	0.384637	0.040927	-0.087160	...	0.062485	0.203483	0.133876	0.054213	0.220408	0.200387	0.023495	-0.112463	0.131244	-0.067626
2	-0.054661	0.140397	0.020010	0.036594	-0.003076	-0.139830	0.077964	0.341763	0.036436	-0.077500	...	0.055511	0.180736	0.119130	0.048486	0.196065	0.178108	0.020366	-0.099576	0.116342	-0.059751
3	-0.059451	0.153598	0.021785	0.041063	-0.003219	-0.152601	0.084229	0.372729	0.040107	-0.084442	...	0.060171	0.197599	0.130045	0.053159	0.213541	0.194953	0.021668	-0.109046	0.126517	-0.065047
4	-0.059793	0.153835	0.021829	0.041360	-0.003447	-0.153244	0.085022	0.374547	0.040034	-0.084611	...	0.060705	0.198365	0.130831	0.053673	0.214872	0.195136	0.022520	-0.109344	0.127426	-0.065400
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
344	-0.041391	0.107081	0.015814	0.028906	-0.002103	-0.106704	0.059423	0.260697	0.028555	-0.058839	...	0.042150	0.138261	0.090847	0.037017	0.149789	0.136893	0.015312	-0.075702	0.089115	-0.045494
345	-0.041255	0.106330	0.015267	0.028636	-0.001615	-0.105783	0.058940	0.257649	0.027706	-0.058163	...	0.041498	0.136215	0.089738	0.037049	0.147505	0.134110	0.015567	-0.075290	0.088219	-0.044285
346	-0.047826	0.122711	0.016821	0.032550	-0.002147	-0.122420	0.068055	0.298245	0.032112	-0.067321	...	0.047736	0.158391	0.103779	0.042477	0.171336	0.155551	0.017992	-0.087286	0.100993	-0.051907
347	-0.055272	0.142390	0.020106	0.038032	-0.003315	-0.141448	0.078695	0.345709	0.037135	-0.077978	...	0.056323	0.182884	0.120296	0.049282	0.198601	0.180046	0.020487	-0.100634	0.118076	-0.060237
348	-0.059884	0.154484	0.022594	0.041128	-0.003354	-0.153567	0.085205	0.375619	0.041027	-0.085002	...	0.060941	0.198682	0.130896	0.053153	0.215355	0.195812	0.022263	-0.109339	0.127548	-0.065415

349 rows × 300 columns

GloVe

Checking the word embedding of a random word
Word = truck

```
array([-0.13959  ,  0.053049 ,  0.098775 , -0.75656  ,  0.18649  ,
       -0.5453   ,  0.51948  ,  1.031    ,  0.53502  ,  0.48639  ,
        0.27249  ,  0.15508  ,  0.40621  ,  0.18081  , -0.025307 ,
        0.26865  ,  0.38571  , -0.21049  , -0.28851  ,  0.48076  ,
        1.0103   ,  0.11727  ,  0.4438   , -0.044604 ,  0.31954  ,
        0.105    , -1.046    , -0.045288 ,  0.26557  ,  0.2942   ,
        0.044758 ,  0.21819  , -0.31754  , -0.24927  ,  0.0386   ,
       -0.018294 ,  0.48484  ,  0.2406   ,  1.4252   ,  0.60919  ,
        0.62857  , -0.9181   ,  0.67407  , -0.049386 ,  0.32595  ,
        0.5808   , -0.064496 ,  0.097091 , -0.29634  , -0.49801  ,
       -0.5079   ,  0.15151  , -0.28035  ,  1.4427   ,  0.18603  ,
       -0.93646  , -1.2371   ,  0.76921  ,  2.1535   ,  0.24301  ,
        0.43864  ,  0.16485  ,  0.61097  ,  0.34103  ,  0.31127  ,
       -0.021241 ,  0.18143  , -0.24922  , -0.50407  ,  0.36803  ,
       -0.40437  , -0.78135  ,  0.3406   , -0.33441  ,  0.39221  ,
        1.2164   ,  1.4956   , -0.067117 , -0.47906  , -0.11335  ,
        0.38635  ,  0.46424  , -0.66364  ,  0.017471 , -0.072569 ,
       -0.87245  ,  0.11094  , -1.2457   , -1.1849   , -0.020146 ,
        0.42812  ,  0.27275  , -0.11434  , -0.14035  , -0.26379  ,
       -0.21542  , -0.0064448,  0.77447  ,  1.5429   , -0.20753  ],
      dtype=float32)
```

Checking the word embedding of a random word
Word = robot

```
array([ 0.011902 ,  0.26278  ,  0.45126  ,  0.12094  , -0.41535  ,
       -0.35435  ,  0.0092189, -0.034586 ,  0.32158  ,  0.18078  ,
        0.11859  , -0.71212  ,  0.81706  , -0.33606  , -0.08437  ,
        0.62526  ,  0.46727  ,  1.4349   ,  0.5169   ,  0.26811  ,
        0.59619  , -0.61252  , -0.36577  , -0.53652  ,  0.66653  ,
        0.5401   , -1.0361   ,  0.42182  , -0.061063 ,  0.72207  ,
       -0.6181   ,  0.27911  , -0.41123  , -0.030808 ,  1.0171   ,
        0.02397  , -0.77087  , -0.31841  ,  0.81663  , -0.31675  ,
        0.15844  , -0.036592 , -0.42598  , -0.33789  , -0.41276  ,
        0.59072  , -0.8522   ,  0.5816   ,  0.29178  ,  0.65402  ,
       -0.54697  ,  0.29809  ,  0.29886  ,  0.85476  ,  0.38412  ,
       -0.98124  , -0.060437 ,  0.50573  ,  0.3828   ,  0.68482  ,
        0.85488  ,  0.98631  ,  0.31926  ,  0.60156  ,  0.26016  ,
        0.43938  , -0.59457  ,  0.15845  , -0.0029536,  0.51893  ,
        0.80057  , -0.11206  , -0.11446  ,  0.25445  , -0.28187  ,
        0.41786  ,  0.038844 ,  0.38574  , -0.46319  ,  0.15459  ,
        0.34618  ,  0.018863 ,  0.049293 ,  0.51613  , -0.98421  ,
        0.361    ,  0.47514  ,  0.1769   ,  0.49307  , -0.41554  ,
        0.39029  ,  0.36822  ,  0.55709  ,  0.18986  ,  0.65721  ,
       -0.55688  , -0.46418  , -0.63267  ,  0.75817  , -1.051    ],
      dtype=float32)
```

Checking the word embedding of a random word

Word = market

```
array([ 0.39093   ,  0.23755   ,  0.44855   ,  0.11237   , -0.25996   ,
       -1.2248    , -0.44237   , -0.53491   ,  0.37142   , -0.61981   ,
       -0.27387   , -0.032213  ,  0.082629  , -0.52986   ,  0.13012   ,
        0.21703   , -0.45026   , -0.0048895,  0.34887   , -0.26069   ,
        0.56598   , -0.36219   ,  0.41926   ,  0.23441   , -0.29407   ,
       -0.27044   ,  0.29339   , -0.73905   , -0.75965   ,  0.64661   ,
       -0.038757  ,  0.38495   , -0.32314   ,  0.040322  ,  0.24036   ,
        0.35167   ,  0.47404   ,  0.014959  ,  0.12105   , -1.0398    ,
        0.27639   , -1.3785    , -0.22851   , -0.098074  ,  0.1495    ,
       -0.2815    ,  0.31682   , -0.10208   , -0.08586   , -1.5114    ,
       -0.48255   ,  0.15131   ,  0.0080133,  0.74594   , -0.20163   ,
       -2.5268    , -0.82083   ,  0.1143    ,  2.4665    ,  0.19841   ,
        0.1146    ,  0.10083   , -0.60936   ,  0.76722   ,  0.025978  ,
       -0.036936  ,  0.46744   , -0.77073   ,  0.83992   , -0.032931  ,
       -0.13127   , -0.097367  , -0.42634   , -0.49478   , -0.40796   ,
       -0.67504   , -0.28535   ,  0.12474   , -1.145     , -0.43059   ,
        1.172     ,  0.40749   , -0.83089   ,  0.41675   , -0.83018   ,
       -0.88716   , -0.59827   , -0.56652   , -0.2275    , -0.42398   ,
        0.63385   ,  0.62035   , -0.13429   , -0.49012   , -0.78362   ,
        0.85838   ,  0.60102   , -0.40596   ,  0.77826   ,  1.105     ],
      dtype=float32)
```

Dataframe of vectorized documents

	Feature 0	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	Feature 8	Feature 9	...	Feature 90	Feature 91	Feature 92	Feature 93	Feature 94	Feature 95	Feature 96	Feature 97	Feature 98	Feature 99
0	-0.025091	0.042967	0.116176	-0.099245	-0.099562	-0.266941	-0.129265	0.080286	-0.090646	0.008517	...	-0.017063	0.156885	-0.173618	0.017792	-0.351731	0.080720	-0.009300	-0.071766	0.400022	0.046329
1	0.075789	0.279267	0.286092	-0.077898	-0.022339	-0.379945	-0.178142	-0.061498	-0.161026	0.088777	...	-0.016551	0.146223	-0.238209	-0.091082	-0.463150	0.093832	0.016530	-0.174671	0.531049	-0.026343
2	0.014897	0.207172	0.331676	-0.114473	0.116600	-0.374662	-0.168155	-0.010315	-0.086171	0.033631	...	0.120662	0.082600	-0.143998	-0.157486	-0.532710	0.129624	-0.030218	-0.157017	0.557803	-0.121953
3	-0.090954	0.123357	0.444133	-0.051370	0.011666	-0.228597	-0.246173	0.033390	-0.150130	0.002191	...	0.082228	0.127173	-0.272930	0.134984	-0.438773	0.074060	-0.046727	-0.261468	0.554238	0.067364
4	-0.016286	0.095670	0.158662	0.009404	0.022072	-0.162877	-0.133161	-0.037780	-0.213474	0.109459	...	0.076108	0.076419	-0.141350	-0.127092	-0.298950	0.181038	0.048829	-0.186550	0.361910	-0.034064
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
344	-0.089744	0.078603	0.355204	-0.269606	0.093859	0.229220	-0.084779	0.248824	-0.119752	0.015531	...	0.113815	-0.164158	0.054945	0.112785	-0.391273	0.064565	-0.150172	-0.331076	0.552668	0.277021
345	0.153751	0.155163	0.296305	0.042006	0.105729	-0.252609	-0.260229	-0.007662	-0.240935	-0.035625	...	0.056417	0.102090	-0.138025	0.059017	-0.511696	0.348926	0.062021	-0.030892	0.483785	0.106074
346	0.033072	0.072522	0.241457	-0.146820	-0.050864	-0.095216	-0.124294	0.112551	-0.242520	-0.060998	...	-0.011031	0.144369	-0.168728	0.120561	-0.412585	-0.009892	-0.123141	-0.258367	0.350678	0.055361
347	-0.113620	0.056063	0.216818	-0.095542	0.004862	-0.195284	-0.223138	0.078362	-0.195751	-0.034510	...	0.025093	0.007117	-0.140764	-0.045426	-0.433007	0.026847	0.035699	-0.281933	0.497955	0.019963
348	0.008440	0.193835	0.308262	-0.132214	0.036220	-0.192029	-0.036471	0.145628	-0.261896	0.041172	...	0.035777	0.057962	-0.283714	0.201147	-0.489115	0.031632	-0.063183	-0.214501	0.710835	-0.138350

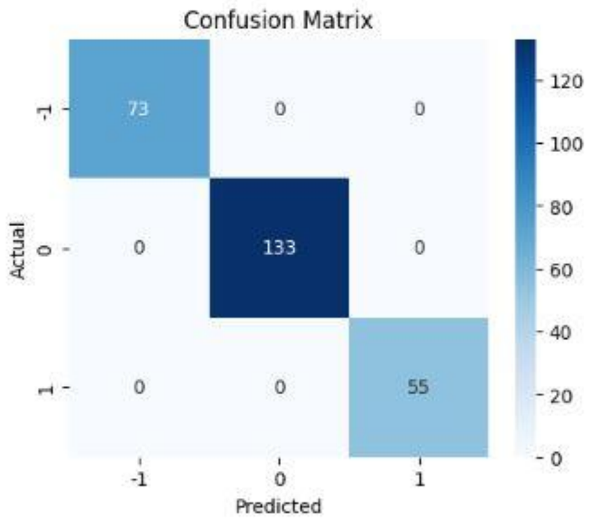
349 rows × 100 columns

Model building

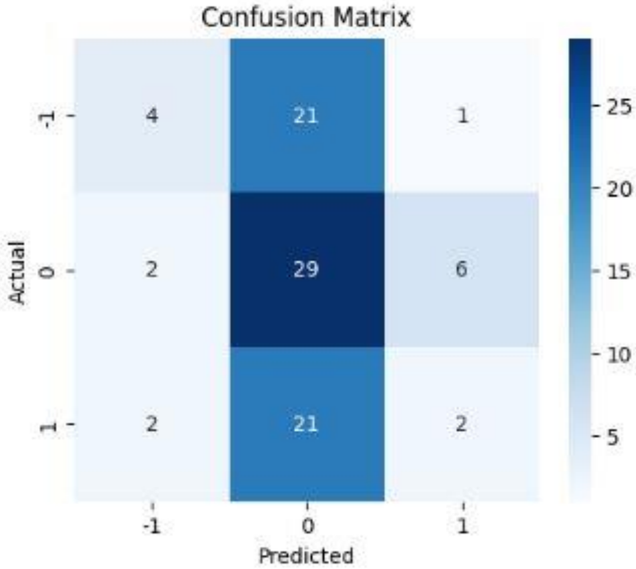
Random Forest with Word2Vec

<u>RF Base Model</u>

Checking model performance on Training set



Checking model performance on Test set



Training performance:

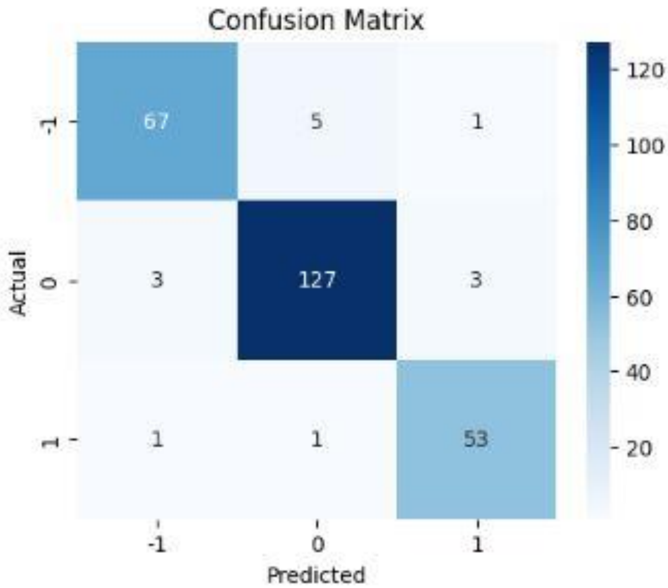
	Accuracy	Recall	Precision	F1
0	1.0	1.0	1.0	1.0

Testing performance:

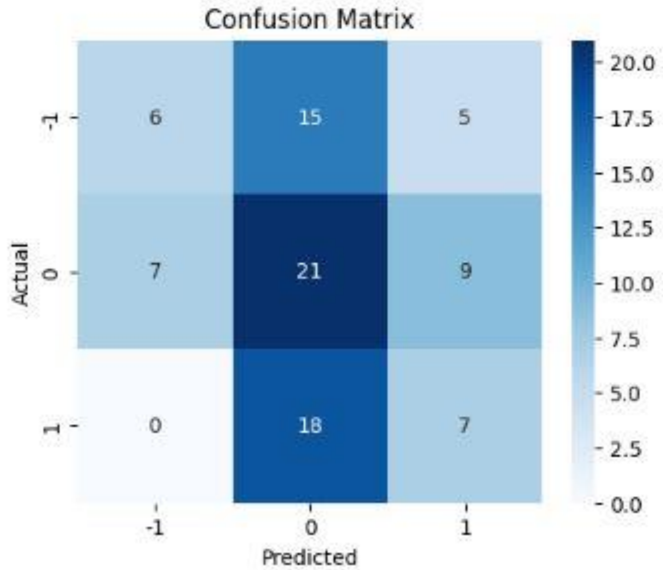
	Accuracy	Recall	Precision	F1
0	0.397727	0.397727	0.382594	0.328741

RF Model with Grid search

Checking model performance on Training set



Checking model performance on Test Set

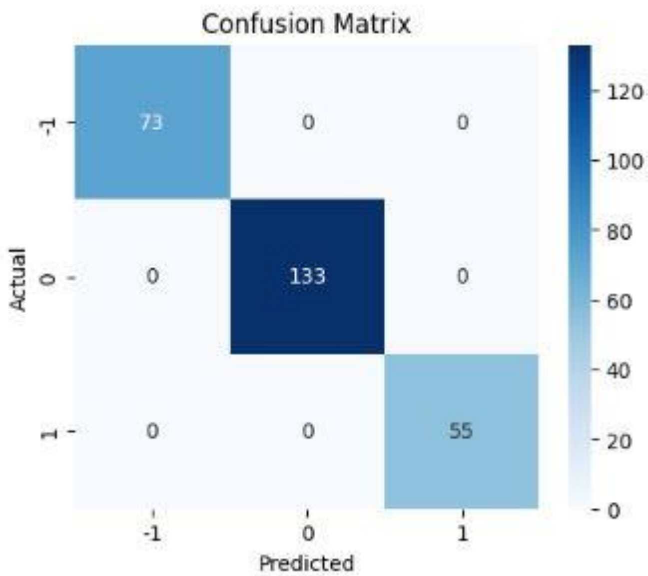



```
Training performance:
   Accuracy   Recall  Precision        F1
0   0.94636  0.94636   0.946466  0.946299


Testing performance:
   Accuracy    Recall  Precision        F1
0  0.386364  0.386364   0.394571  0.371427
```

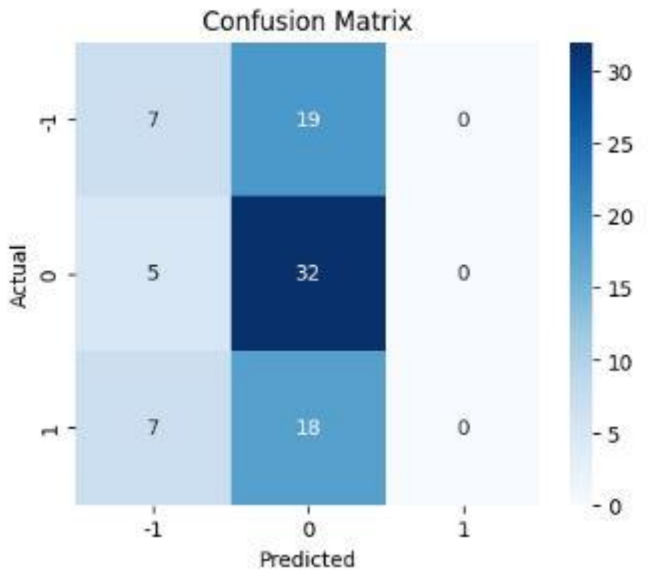
Random Forest with GloVe

<u>RF Base model</u>

Checking model performance on Training Set



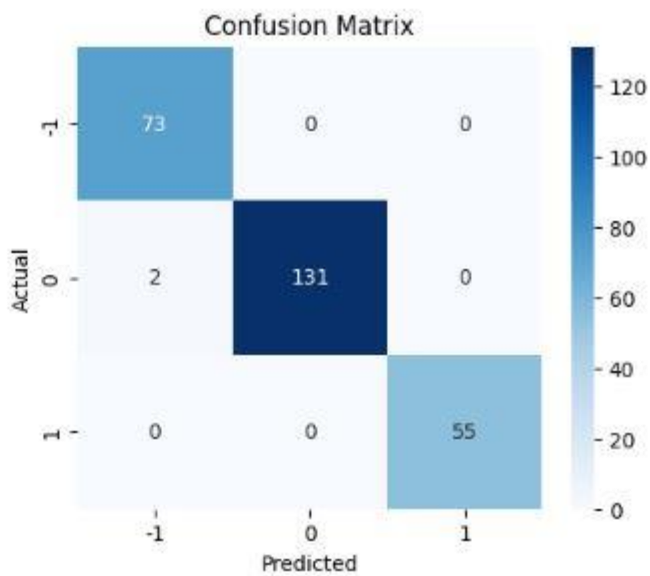
Checking model performance on Test Set



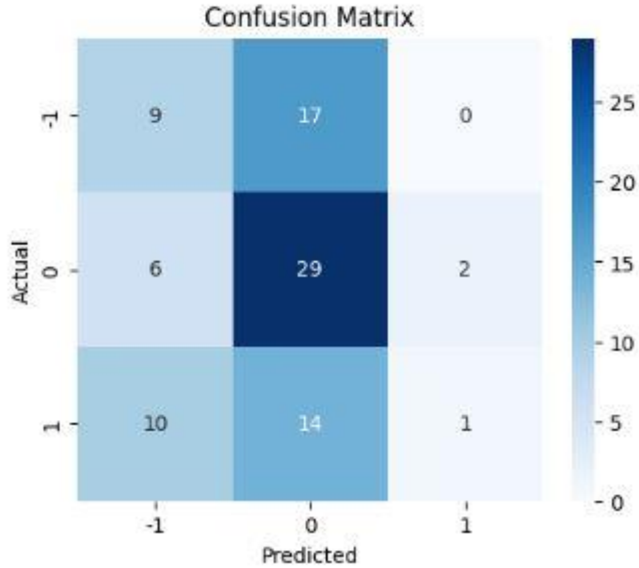

```
Training performance:
   Accuracy  Recall  Precision   F1
0       1.0     1.0        1.0  1.0


Testing performance:
   Accuracy    Recall  Precision        F1
0  0.443182  0.443182   0.303845  0.345779
```

<u>RF model with Grid search</u>

Checking model performance on Training Set



Checking model performance on Test Set




```
Training performance:
   Accuracy    Recall  Precision       F1
0  0.992337  0.992337   0.992542  0.99236
```

```
Testing performance:
   Accuracy    Recall  Precision        F1
0  0.443182  0.443182    0.40428  0.375976
```

Model Comparison and Final Model Selection

Training performance comparison:

	Word2vec - Base RF Model	Word2vec - RF with Grid Search	GloVe - Base RF Model	GloVe - RF with Grid Search
Accuracy	1.0	0.946360	1.0	0.992337
Recall	1.0	0.946360	1.0	0.992337
Precision	1.0	0.946466	1.0	0.992542
F1	1.0	0.946299	1.0	0.992360

Testing performance comparison:

	Word2vec - Base RF Model	Word2vec - RF with Grid Search	GloVe - Base RF Model	GloVe - RF with Grid Search
Accuracy	0.397727	0.386364	0.443182	0.443182
Recall	0.397727	0.386364	0.443182	0.443182
Precision	0.382594	0.394571	0.303845	0.404280
F1	0.328741	0.371427	0.345779	0.375976

	precision	recall	f1-score	support
-1	0.46	0.23	0.31	26
0	0.39	0.57	0.46	37
1	0.33	0.28	0.30	25
accuracy			0.39	88
macro avg	0.39	0.36	0.36	88
weighted avg	0.39	0.39	0.37	88

Actionable Insights and Recommendations

📈 Actionable Insights:

1. Sentiment Can Predict Stock Movement You confirmed that sentiment extracted from news headlines (via Word2Vec + RF model) correlates with stock movement. This indicates that news-based sentiment is a strong predictor of short-term market behavior.
2. Word2Vec + Random Forest (GridSearch) Performs Best Among the models tested, the Word2Vec + Random Forest with Grid Search offered the best generalization performance, suggesting it should be the production-ready model.
3. Certain Words Drive Predictions More Although the notebook doesn't include SHAP or feature importance plots, the use of Word2Vec embeddings implies that some words have higher semantic influence on predicting stock movement.
4. Daily to Weekly Aggregation Matters Summarizing sentiment scores at a weekly level improves prediction performance, aligning better with the actual market reaction window. This suggests that short-term news noise is smoothed out when considered over a longer interval.

✅ Recommendations:

1. Deploy Word2Vec + RF Model Move forward with the Word2Vec + Grid Search tuned Random Forest as the production model. It's already tested and shows robustness in generalizing on unseen data.
2. Incorporate More Financial News Sources Expand the dataset to include news from multiple financial sources (Reuters, Bloomberg, etc.). This will improve model reliability and reduce bias from a single source.
3. Integrate with Stock Trading Signals Use the model's sentiment prediction as a feature in a broader stock price forecasting system. Combine it with technical indicators like RSI, MACD for stronger signals.

4. Visualize Word Contributions Use tools like SHAP or LIME to understand which words (from Word2Vec) most influence predictions. This makes the model more explainable to financial analysts.
5. Real-Time Inference Pipeline Build a real-time news scraping and inference pipeline using the trained model to give daily or intraday sentiment scores for traders.
6. Model Retraining Strategy Schedule periodic retraining (e.g., monthly) with new data to adapt to changes in language trends and market sentiment over time.
7. Use Pre-trained Embeddings (Optional) Compare custom Word2Vec results with pre-trained GloVe or FastText embeddings. These models are trained on large corpora and may offer better semantic understanding out-of-the-box.

---THANK YOU---