nlp-text-classification-siddhant

April 21, 2025

1 Google Colab Lab Assignment -NLP

Course Name: Deep Learning

Lab Title: NLP Techniques for Text Classification

Student ID: 202201040036

Objective The objective of this assignment is to implement NLP preprocessing techniques and build a text classification model using machine learning techniques.

Learning Outcomes:

1. Understand and apply NLP preprocessing techniques such as tokenization, stopword removal, stemming, and lemmatization.
2. Implement text vectorization techniques such as TF-IDF and CountVectorizer.
3. Develop a text classification model using a machine learning algorithm.
4. Evaluate the performance of the model using suitable metrics.

2 Assignment Instructions:

Part 1: NLP Preprocessing

Dataset Selection: https://www.kaggle.com/datasets/thedevastator/uncovering-financial-insights-with-the-reuters-2

Drive Link for Dataset: https://drive.google.com/drive/folders/1uLf9u9_M4nuFGGa-YTC6EmaXDSKfrFF4?usp=sharing

Choose any text dataset from **Best Datasets for Text** https://en.innovatiana.com/post/best-datasets-for-text-classification Classification, such as SMS Spam Collection, IMDb Reviews, or any other relevant dataset.

Download the dataset and upload it to Google Colab.

Load the dataset into a Pandas DataFrame and explore its structure (e.g., check missing values, data types, and label distribution).

Text Preprocessing:

Convert text to lowercase.

Perform tokenization using NLTK or spaCy.

Remove stopwords using NLTK or spaCy.

Apply stemming using PorterStemmer or SnowballStemmer.

Apply lemmatization using WordNetLemmatizer.

Vectorization Techniques:

Convert text data into numerical format using TF-IDF and CountVectorizer.

[ ]:
```python
import pandas as pd

# Load the dataset
train_df = pd.read_csv("/content/ModApte_train.csv")
test_df = pd.read_csv("/content/ModApte_test.csv")

# Display information
print("Train Data Info:")
print(train_df.info())

print("\nTest Data Info:")
print(test_df.info())

# Check for missing values
print("\nMissing Values in Train Set:")
print(train_df.isnull().sum())

print("\nMissing Values in Test Set:")
print(test_df.isnull().sum())

# Check label distribution
print("\nLabel Distribution in Train Set:")
print(train_df['topics'].value_counts())

print("\nLabel Distribution in Test Set:")
print(test_df['topics'].value_counts())
```

```
Train Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9603 entries, 0 to 9602
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   text         8816 non-null   object
 1   text_type    9603 non-null   object
 2   topics       9603 non-null   object
 3   lewis_split  9603 non-null   object
 4   cgis_split   9603 non-null   object
 5   old_id       9603 non-null   object
 6   new_id       9603 non-null   object
```

```
 7   places       9603 non-null   object
 8   people       9603 non-null   object
 9   orgs         9603 non-null   object
 10  exchanges    9603 non-null   object
 11  date         9603 non-null   object
 12  title        9549 non-null   object
dtypes: object(13)
memory usage: 975.4+ KB
None

Test Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3299 entries, 0 to 3298
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   text         3023 non-null   object
 1   text_type    3299 non-null   object
 2   topics       3299 non-null   object
 3   lewis_split  3299 non-null   object
 4   cgis_split   3299 non-null   object
 5   old_id       3299 non-null   object
 6   new_id       3299 non-null   object
 7   places       3299 non-null   object
 8   people       3299 non-null   object
 9   orgs         3299 non-null   object
 10  exchanges    3299 non-null   object
 11  date         3299 non-null   object
 12  title        3285 non-null   object
dtypes: object(13)
memory usage: 335.2+ KB
None

Missing Values in Train Set:
text           787
text_type        0
topics           0
lewis_split      0
cgis_split       0
old_id           0
new_id           0
places           0
people           0
orgs             0
exchanges        0
date             0
title           54
dtype: int64
```

```
Missing Values in Test Set:
text           276
text_type        0
topics           0
lewis_split      0
cgis_split       0
old_id           0
new_id           0
places           0
people           0
orgs             0
exchanges        0
date             0
title           14
dtype: int64

Label Distribution in Train Set:
topics
['earn']                                                       2840
[]                                                             1828
['acq']                                                        1596
['crude']                                                       253
['trade']                                                       251
                                                               …
['grain' 'corn' 'rice' 'oilseed' 'soybean' 'orange']              1
['trade' 'bop' 'money-fx' 'dlr']                                  1
['gnp' 'cpi' 'money-fx']                                          1
['zinc' 'lead' 'copper']                                          1
['grain' 'corn' 'soybean' 'oilseed']                              1
Name: count, Length: 473, dtype: int64

Label Distribution in Test Set:
topics
['earn']                                    1083
['acq']                                      696
[]                                           280
['crude']                                    121
['money-fx']                                  87
                                            …
['trade' 'carcass']                            1
['oilseed' 'soybean' 'veg-oil' 'trade']        1
['trade' 'livestock' 'carcass' 'sugar']        1
['money-fx' 'rand']                            1
['money-fx' 'dlr' 'yen' 'dmk']                 1
Name: count, Length: 258, dtype: int64
```

```python
import nltk
import spacy
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer

# Download required NLTK resources
nltk.download('stopwords')
nltk.download('wordnet')

# Load spaCy's English model
nlp = spacy.load("en_core_web_sm")

# Initialize tools
stop_words = set(stopwords.words("english"))
stemmer = PorterStemmer()
lemmatizer = WordNetLemmatizer()

def preprocess_text(text):
    # Check if the text is a string and not NaN
    if isinstance(text, str):
        # Convert to lowercase
        text = text.lower()

        # Tokenization using spaCy
        doc = nlp(text)
        tokens = [token.text for token in doc if token.is_alpha]  # Remove␣
↪non-alphabetic tokens

        # Remove stopwords
        tokens = [word for word in tokens if word not in stop_words]

        # Stemming using PorterStemmer
        stemmed_tokens = [stemmer.stem(word) for word in tokens]

        # Lemmatization using WordNetLemmatizer
        lemmatized_tokens = [lemmatizer.lemmatize(word) for word in␣
↪stemmed_tokens]

        # Return the processed text
        return " ".join(lemmatized_tokens)
    else:
        # If the value is not a string (e.g., NaN), return an empty string
        return ""

# Apply preprocessing to both train and test sets
train_df["cleaned_text"] = train_df["title"].apply(preprocess_text)
test_df["cleaned_text"] = test_df["title"].apply(preprocess_text)
```

```
# Check processed text
print("\nExample Processed Text (Train Set):")
print(train_df["cleaned_text"].head())
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data…


Example Processed Text (Train Set):
0                        bahia cocoa review
1     nation averag price farmer own reserv
2              argentin grain oilse registr
3                    usx debt dowgrad moodi
4       champion product approv stock split
Name: cleaned_text, dtype: object
```

[ ]:
```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

# CountVectorizer: Converts a collection of text documents into a matrix of␣
↪token counts
count_vectorizer = CountVectorizer(max_features=100)  # Limiting to 100␣
↪features for demonstration
X_train_count = count_vectorizer.fit_transform(train_df["cleaned_text"])
X_test_count = count_vectorizer.transform(test_df["cleaned_text"])

# Convert to DataFrame to view top words
count_df = pd.DataFrame(X_train_count.toarray(), columns=count_vectorizer.
↪get_feature_names_out())
print("\nTop words in CountVectorizer (Train Set):")
print(count_df.head())

# TF-IDF Vectorizer: Converts text data into numerical values based on term␣
↪frequency and inverse document frequency
tfidf_vectorizer = TfidfVectorizer(max_features=100)  # Limiting to 100␣
↪features for demonstration
X_train_tfidf = tfidf_vectorizer.fit_transform(train_df["cleaned_text"])
X_test_tfidf = tfidf_vectorizer.transform(test_df["cleaned_text"])

# Convert to DataFrame to view top words
tfidf_df = pd.DataFrame(X_train_tfidf.toarray(), columns=tfidf_vectorizer.
↪get_feature_names_out())
print("\nTop words in TF-IDF (Train Set):")
print(tfidf_df.head())
```

```
Top words in CountVectorizer (Train Set):
```

```
   acquir  acquisit  american  bank  bid  bill  billion  bond  brazil  buy  \
0       0         0         0     0    0     0        0     0       0    0
1       0         0         0     0    0     0        0     0       0    0
2       0         0         0     0    0     0        0     0       0    0
3       0         0         0     0    0     0        0     0       0    0
4       0         0         0     0    0     0        0     0       0    0

   …  tender  trade  treasuri  two  unit  usda  week  wheat  year  yen
0  …       0      0         0    0     0     0     0      0     0    0
1  …       0      0         0    0     0     0     0      0     0    0
2  …       0      0         0    0     0     0     0      0     0    0
3  …       0      0         0    0     0     0     0      0     0    0
4  …       0      0         0    0     0     0     0      0     0    0

[5 rows x 100 columns]

Top words in TF-IDF (Train Set):
   acquir  acquisit  american  bank  bid  bill  billion  bond  brazil  buy  \
0     0.0       0.0       0.0   0.0  0.0   0.0      0.0   0.0     0.0  0.0
1     0.0       0.0       0.0   0.0  0.0   0.0      0.0   0.0     0.0  0.0
2     0.0       0.0       0.0   0.0  0.0   0.0      0.0   0.0     0.0  0.0
3     0.0       0.0       0.0   0.0  0.0   0.0      0.0   0.0     0.0  0.0
4     0.0       0.0       0.0   0.0  0.0   0.0      0.0   0.0     0.0  0.0

   …  tender  trade  treasuri  two  unit  usda  week  wheat  year  yen
0  …     0.0    0.0       0.0  0.0   0.0   0.0   0.0    0.0   0.0  0.0
1  …     0.0    0.0       0.0  0.0   0.0   0.0   0.0    0.0   0.0  0.0
2  …     0.0    0.0       0.0  0.0   0.0   0.0   0.0    0.0   0.0  0.0
3  …     0.0    0.0       0.0  0.0   0.0   0.0   0.0    0.0   0.0  0.0
4  …     0.0    0.0       0.0  0.0   0.0   0.0   0.0    0.0   0.0  0.0

[5 rows x 100 columns]
```

Splitting the Data:

Divide the dataset into training and testing sets (e.g., 80% training, 20% testing).

Building the Classification Model:

Train a text classification model using Logistic Regression, Naïve Bayes, or any other suitable algorithm.

Implement the model using scikit-learn.

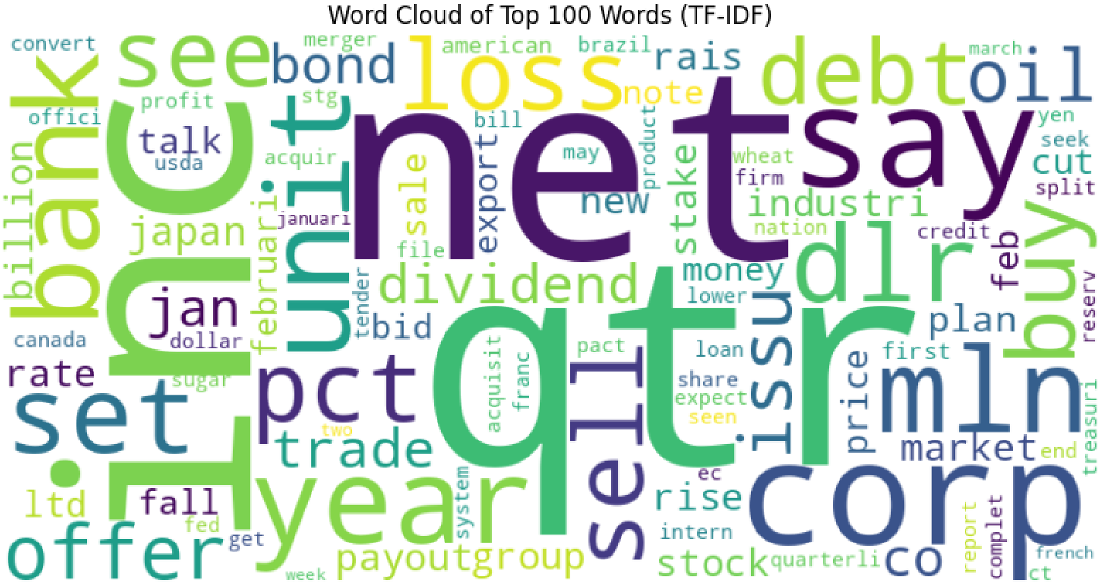
Model Evaluation:

Evaluate the model using accuracy, precision, recall, and F1-score.

Use a confusion matrix to visualize the results.

[ ]:
```python
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Get feature names (top 100 words from TF-IDF)
feature_names = tfidf_vectorizer.get_feature_names_out()

# Compute average TF-IDF score for each word
word_tfidf_scores = X_train_tfidf.mean(axis=0).A1  # Convert sparse matrix to␣
↪array

# Create a dictionary of words and their scores
word_freq = dict(zip(feature_names, word_tfidf_scores))

# Generate word cloud
plt.figure(figsize=(10, 6))
wordcloud = WordCloud(width=800, height=400, background_color='white').
↪generate_from_frequencies(word_freq)
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title("Word Cloud of Top 100 Words (TF-IDF)")
plt.show()
```

Word Cloud of Top 100 Words (TF-IDF)

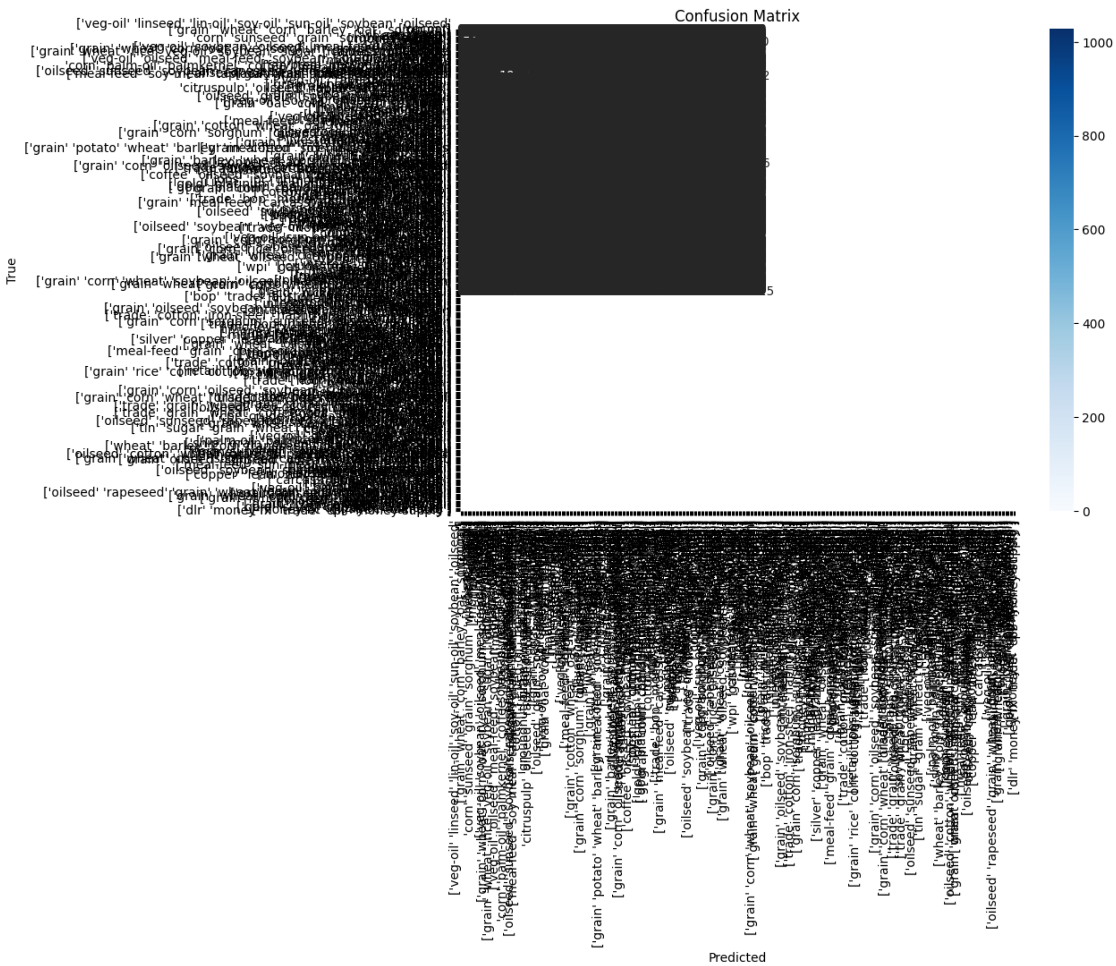


[ ]:
```python
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
↪f1_score, confusion_matrix
import seaborn as sns
```

```python
import matplotlib.pyplot as plt

# Prepare the labels
y_train = train_df['topics']
y_test = test_df['topics']

# Train a Naïve Bayes model using the CountVectorizer features (X_train_count)
nb_model = MultinomialNB()
nb_model.fit(X_train_count, y_train)

# Predict the labels for the test set
y_pred = nb_model.predict(X_test_count)

# Model Evaluation
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1-Score: {f1:.4f}")

# Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# Visualize confusion matrix using seaborn heatmap
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',␣
 ↪xticklabels=train_df['topics'].unique(), yticklabels=train_df['topics'].
 ↪unique())
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

```
Accuracy: 0.6369
Precision: 0.5276
Recall: 0.6369
F1-Score: 0.5701
```




```
[ ]: from sklearn.metrics import classification_report

     # Ensure that the labels are consistent with the unique values in y_test and␣
      ↪y_pred
     labels = train_df['topics'].unique()  # Use the unique labels from your train␣
      ↪dataset

     # Generate the classification report
     report = classification_report(y_test, y_pred, labels=labels)
     print(report)
```

```
precision    recall  f1-score   support

                                                                   ['cocoa']
0.00      0.00      0.00        15
                       ['grain' 'wheat' 'corn' 'barley' 'oat' 'sorghum']
0.00      0.00      0.00         1
```

```
                                       ['veg-oil' 'linseed' 'lin-oil' 'soy-oil'
'sun-oil' 'soybean' 'oilseed'
 'corn' 'sunseed' 'grain' 'sorghum' 'wheat']       0.00      0.00      0.00
0
                                                                          []
0.50      0.80      0.61       280
                                                                    ['earn']
0.74      0.95      0.83      1083
                                                                     ['acq']
0.76      0.78      0.77       696
                                                              ['earn' 'acq']
0.00      0.00      0.00         0
                                                           ['wheat' 'grain']
0.00      0.00      0.00         3
                                                                  ['copper']
0.40      0.31      0.35        13
                                                                 ['housing']
0.00      0.00      0.00         2
                                                            ['money-supply']
0.36      0.46      0.41        28
                                                                  ['coffee']
0.05      0.05      0.05        22
                                                              ['acq' 'ship']
0.00      0.00      0.00         0
                                                                   ['sugar']
0.52      0.88      0.66        25
                                                                   ['trade']
0.46      0.61      0.53        75
                                                                ['reserves']
0.75      0.75      0.75        12
                                                                    ['ship']
0.00      0.00      0.00        36
                                                            ['grain' 'corn']
0.19      0.21      0.20        19
                  ['veg-oil' 'soybean' 'oilseed' 'meal-feed' 'soy-meal']
0.00      0.00      0.00         0
      ['grain' 'wheat' 'corn' 'oat' 'rye' 'sorghum' 'soybean' 'oilseed']
0.00      0.00      0.00         0
                                                                  ['cotton']
0.00      0.00      0.00         9
                                                            ['grain' 'ship']
0.00      0.00      0.00         5
                                                     ['carcass' 'livestock']
0.00      0.00      0.00         2
                                                                   ['grain']
0.67      0.20      0.31        10
                                                                   ['crude']
0.48      0.60      0.53       121
```

```
                                                      ['nat-gas']
0.00      0.00      0.00        12
                                                    ['cpi' 'gnp']
0.00      0.00      0.00         1
                                                ['grain' 'wheat']
0.41      0.91      0.56        32
                                           ['grain' 'corn' 'oat']
0.00      0.00      0.00         0
        ['veg-oil' 'oilseed' 'meal-feed' 'soybean' 'soy-oil' 'soy-meal']
0.00      0.00      0.00         0
                                                          ['cpi']
0.18      0.24      0.21        17
                                          ['money-fx' 'interest']
0.54      0.46      0.50        28
                                                     ['interest']
0.59      0.63      0.61        81
                                                    ['gnp' 'bop']
0.00      0.00      0.00         2
                                                 ['grain' 'rice']
0.00      0.00      0.00         7
                                 ['soybean' 'red-bean' 'oilseed']
0.00      0.00      0.00         0
['grain' 'wheat' 'rice' 'veg-oil' 'soybean' 'sugar' 'rubber' 'copra-cake'
 'corn' 'palm-oil' 'palmkernel' 'coffee' 'tea' 'plywood' 'soy-meal'
 'cotton']       0.00      0.00      0.00         0
                                                     ['money-fx']
0.17      0.13      0.14        87
                                       ['meal-feed' 'copra-cake']
0.00      0.00      0.00         0
                                                         ['alum']
0.00      0.00      0.00        19
                                           ['veg-oil' 'palm-oil']
0.00      0.00      0.00         4
                                         ['tea' 'cocoa' 'coffee']
0.00      0.00      0.00         0
                                            ['oilseed' 'soybean']
0.00      0.00      0.00         6
                    ['oilseed' 'soybean' 'meal-feed' 'soy-meal']
0.00      0.00      0.00         0
                           ['gold' 'platinum' 'strategic-metal']
0.00      0.00      0.00         0
                                          ['meal-feed' 'tapioca']
0.00      0.00      0.00         0
                                                          ['tin']
0.00      0.00      0.00        10
                                                  ['trade' 'bop']
0.00      0.00      0.00         5
                                                     ['oilseed'
```

```
'sunseed' 'soybean' 'rapeseed' 'veg-oil' 'soy-oil' 'palm-oil'
 'groundnut-oil']       0.00      0.00      0.00         0
                                                              ['gold']
0.00      0.00      0.00        20
                                       ['veg-oil' 'rape-oil' 'palm-oil']
0.00      0.00      0.00         0
                                       ['meal-feed' 'soy-meal' 'tapioca' 'grain'
'corn' 'cornglutenfeed'
 'citruspulp' 'oilseed' 'rapeseed' 'rape-meal']       0.00      0.00      0.00
0
                                                   ['strategic-metal']
0.00      0.00      0.00         6
                                                      ['crude' 'ship']
0.00      0.00      0.00        22
                                       ['grain' 'wheat' 'corn' 'barley']
0.00      0.00      0.00         0
                                                       ['grain' 'oat']
0.00      0.00      0.00         0
                                          ['grain' 'wheat' 'wool' 'dlr']
0.00      0.00      0.00         0
                                                ['livestock' 'l-cattle']
0.00      0.00      0.00         2
                                                            ['retail']
0.00      0.00      0.00         1
                                               ['gold' 'acq' 'platinum']
0.00      0.00      0.00         0
                                                               ['ipi']
1.00      0.45      0.62        11
                                                           ['oilseed']
0.00      0.00      0.00         0
                                                     ['gold' 'silver']
0.00      0.00      0.00         1
                                       ['grain' 'corn' 'wheat' 'barley']
0.00      0.00      0.00         1
                                                        ['iron-steel']
0.00      0.00      0.00        12
                                                            ['rubber']
0.00      0.00      0.00         9
                              ['oilseed' 'grain' 'soybean' 'wheat' 'corn']
0.00      0.00      0.00         0
                                                   ['crude' 'nat-gas']
1.00      0.11      0.20         9
                                                   ['livestock' 'hog']
0.00      0.00      0.00         0
                                                ['propane' 'heat' 'gas']
0.00      0.00      0.00         0
                                ['veg-oil' 'soy-oil' 'oilseed' 'soybean']
0.00      0.00      0.00         0
```

```
                                                  ['heat']
0.00      0.00      0.00       4
                                           ['gnp' 'trade']
0.00      0.00      0.00       0
                 ['grain' 'oat' 'corn' 'oilseed' 'soybean']
0.00      0.00      0.00       0
                                                  ['jobs']
0.25      0.08      0.12      12
                                                   ['lei']
0.00      0.00      0.00       3
                               ['money-fx' 'yen' 'dlr']
0.00      0.00      0.00       0
                                                   ['bop']
0.00      0.00      0.00       9
                               ['money-fx' 'saudriyal']
0.00      0.00      0.00       0
                                           ['earn' 'alum']
0.00      0.00      0.00       0
                                ['interest' 'money-fx']
0.00      0.00      0.00      11
                                          ['earn' 'crude']
0.00      0.00      0.00       0
                                        ['coffee' 'crude']
0.00      0.00      0.00       0
                                                   ['gnp']
0.07      0.07      0.07      15
                              ['grain' 'wheat' 'barley']
0.00      0.00      0.00       0
                                                  ['zinc']
0.00      0.00      0.00       5
                         ['veg-oil' 'livestock' 'carcass']
0.00      0.00      0.00       0
                              ['grain' 'corn' 'sorghum']
0.00      0.00      0.00       0
                                  ['oilseed' 'rapeseed']
0.00      0.00      0.00       4
                                               ['veg-oil']
1.00      0.27      0.43      11
                   ['meal-feed' 'soy-meal' 'grain' 'corn']
0.00      0.00      0.00       1
                                ['grain' 'wheat' 'ship']
0.00      0.00      0.00       1
                                                ['orange']
0.00      0.00      0.00       9
                                ['livestock' 'carcass']
0.00      0.00      0.00       3
                                          ['wheat' 'corn']
0.00      0.00      0.00       0
```

```
                    ['grain' 'cotton' 'wheat' 'oat' 'oilseed' 'soybean']
0.00      0.00      0.00        0
                                                           ['carcass']
0.00      0.00      0.00        5
                                                          ['pet-chem']
0.00      0.00      0.00        6
                                                    ['dlr' 'money-fx']
0.00      0.00      0.00       12
                                                               ['gas']
0.00      0.00      0.00        8
                                                    ['money-fx' 'dlr']
0.32      0.60      0.42       15
                                     ['livestock' 'carcass' 'grain']
0.00      0.00      0.00        0
                ['grain' 'corn' 'sorghum' 'oilseed' 'sunseed' 'soybean']
0.00      0.00      0.00        0
                                   ['grain' 'wheat' 'cotton' 'rice']
0.00      0.00      0.00        0
                                                   ['gold' 'copper']
0.00      0.00      0.00        1
                                               ['bop' 'trade' 'gnp']
0.00      0.00      0.00        0
                                           ['grain' 'barley' 'corn']
0.00      0.00      0.00        0
                                                      ['gas' 'fuel']
0.00      0.00      0.00        1
                                                ['nat-gas' 'crude']
0.00      0.00      0.00        2
                                     ['livestock' 'carcass' 'trade']
0.00      0.00      0.00        0
                                            ['grain' 'corn' 'wheat']
0.00      0.00      0.00        1
                               ['grain' 'wheat' 'oilseed' 'soybean']
0.00      0.00      0.00        0
                                               ['money-fx' 'trade']
0.00      0.00      0.00        1
                                                     ['gas' 'crude']
0.00      0.00      0.00        4
                                              ['crude' 'gas' 'fuel']
0.00      0.00      0.00        0
                                                      ['acq' 'earn']
0.00      0.00      0.00        2
                                                     ['crude' 'gas']
0.00      0.00      0.00        2
                      ['grain' 'cotton' 'rice' 'oilseed' 'soybean']
0.00      0.00      0.00        0
                                                     ['bop' 'trade']
0.00      0.00      0.00        6
```

```
                                                      ['ship' 'crude']
0.00      0.00      0.00        16
                                              ['money-fx' 'dlr' 'yen']
0.00      0.00      0.00         5
                                                  ['reserves' 'trade']
0.00      0.00      0.00         0
                                                     ['silver' 'gold']
0.00      0.00      0.00         1
                                                               ['wpi']
0.00      0.00      0.00         9
                                                              ['grain'
'potato' 'wheat' 'barley' 'meal-feed' 'soy-meal' 'hog' 'carcass'
 'livestock']       0.00      0.00      0.00         0
                                       ['grain' 'wheat' 'barley' 'corn']
0.00      0.00      0.00         0
                                            ['grain' 'wheat' 'copper']
0.00      0.00      0.00         0
                                                         ['livestock']
0.00      0.00      0.00         5
                                                    ['grain' 'barley']
0.00      0.00      0.00         5
                  ['grain' 'barley' 'wheat' 'corn' 'oilseed' 'rapeseed']
0.00      0.00      0.00         0
                                                    ['money-fx' 'yen']
0.00      0.00      0.00         0
                              ['copper' 'lead' 'zinc' 'strategic-metal']
0.00      0.00      0.00         0
                                                      ['sugar' 'acq']
0.00      0.00      0.00         0
                                                       ['gnp' 'jobs']
0.00      0.00      0.00         1
                                         ['livestock' 'hog' 'carcass']
0.00      0.00      0.00         0
                                                 ['trade' 'money-fx']
0.00      0.00      0.00         2
                        ['trade' 'sugar' 'cotton' 'groundnut' 'oilseed']
0.00      0.00      0.00         0
                           ['grain' 'wheat' 'corn' 'soybean' 'oilseed']
0.00      0.00      0.00         0
                              ['livestock' 'l-cattle' 'carcass' 'sugar']
0.00      0.00      0.00         0
                                                    ['money-fx' 'can']
0.00      0.00      0.00         0
                                               ['veg-oil' 'groundnut']
0.00      0.00      0.00         0
                                                        ['grain' 'corn'
'oilseed' 'soybean' 'veg-oil' 'soy-oil' 'meal-feed'
 'soy-meal' 'cotton']       0.00      0.00      0.00         0
```

['trade' 'crude' 'nat-gas']

0.00 0.00 0.00 0

['coffee' 'oilseed' 'soybean' 'trade' 'sugar' 'cocoa']

0.00 0.00 0.00 0

['trade' 'grain' 'wheat']

0.00 0.00 0.00 0

['fishmeal' 'meal-feed']

0.00 0.00 0.00 0

['hog' 'livestock']

0.00 0.00 0.00 2

['jobs' 'ipi' 'gnp' 'income' 'trade' 'retail']

0.00 0.00 0.00 0

['money-supply' 'reserves']

0.00 0.00 0.00 0

['interest' 'gnp']

0.00 0.00 0.00 1

['grain' 'wheat' 'corn']

0.00 0.00 0.00 3

['sugar' 'corn' 'grain']

0.00 0.00 0.00 0

['gold' 'platinum' 'palladium' 'nickel' 'copper']

0.00 0.00 0.00 0

['lumber']

0.00 0.00 0.00 4

['ship' 'gas']

0.00 0.00 0.00 0

['gold' 'platinum' 'palladium' 'copper' 'nickel']

0.00 0.00 0.00 0

['grain' 'corn' 'cornglutenfeed' 'meal-feed']

0.00 0.00 0.00 0

['interest' 'crude']

0.00 0.00 0.00 0

['crude' 'fuel' 'jet']

0.00 0.00 0.00 0

['tapioca' 'meal-feed']

0.00 0.00 0.00 0

['cotton' 'sugar' 'veg-oil' 'grain']

0.00 0.00 0.00 0

['instal-debt']

0.00 0.00 0.00 1

['trade' 'gnp' 'bop' 'dlr']

0.00 0.00 0.00 0

['interest' 'money-fx' 'dlr']

0.00 0.00 0.00 1

['gnp' 'jobs' 'cpi' 'bop' 'dfl']

0.00 0.00 0.00 0

['money-fx' 'gnp']

0.00 0.00 0.00 0

```
                                                  ['meal-feed']
0.00      0.00      0.00        1
                ['trade' 'bop' 'money-fx' 'crude' 'gnp' 'dlr']
0.00      0.00      0.00        0
                                   ['money-fx' 'dlr' 'dmk']
0.00      0.00      0.00        1
                                        ['grain' 'oilseed']
0.00      0.00      0.00        1
        ['grain' 'meal-feed' 'carcass' 'soy-meal' 'livestock']
0.00      0.00      0.00        0
                                         ['pet-chem' 'acq']
0.00      0.00      0.00        1
                                   ['meal-feed' 'fishmeal']
0.00      0.00      0.00        0
                                  ['acq' 'crude' 'nat-gas']
0.00      0.00      0.00        5
                                   ['grain' 'corn' 'sugar']
0.00      0.00      0.00        0
                                                   ['lead']
0.00      0.00      0.00        4
                                     ['gnp' 'money-supply']
0.00      0.00      0.00        0
                           ['money-fx' 'dlr' 'trade' 'acq']
0.00      0.00      0.00        0
                             ['hog' 'l-cattle' 'livestock']
0.00      0.00      0.00        0
                 ['oilseed' 'soybean' 'grain' 'corn' 'wheat']
0.00      0.00      0.00        1
                                         ['interest' 'stg']
0.00      0.00      0.00        0
                          ['grain' 'corn' 'wheat' 'oilseed']
0.00      0.00      0.00        0
                           ['cocoa' 'coffee' 'sugar' 'heat']
0.00      0.00      0.00        0
                                                 ['potato']
0.00      0.00      0.00        3
                              ['carcass' 'livestock' 'hog']
0.00      0.00      0.00        0
                            ['trade' 'bop' 'money-fx' 'dlr']
0.00      0.00      0.00        0
                              ['grain' 'corn' 'wheat' 'rice']
0.00      0.00      0.00        0
                                   ['gnp' 'cpi' 'money-fx']
0.00      0.00      0.00        0
                                   ['grain' 'wheat' 'rice']
0.00      0.00      0.00        1
                                   ['zinc' 'lead' 'copper']
0.00      0.00      0.00        0
```

['earn' 'strategic-metal']	0.00	0.00	0.00	0
['money-supply' 'interest']	0.00	0.00	0.00	1
['money-fx' 'yen' 'trade']	0.00	0.00	0.00	0
['money-fx' 'stg' 'can']	0.00	0.00	0.00	0
['oilseed' 'soybean' 'veg-oil' 'palm-oil' 'coconut-oil']	0.00	0.00	0.00	0
['trade' 'money-fx' 'cpi' 'reserves']	0.00	0.00	0.00	0
['copper' 'earn']	0.00	0.00	0.00	0
['money-fx' 'stg']	0.00	0.00	0.00	0
['trade' 'sugar']	0.00	0.00	0.00	0
['gas' 'grain' 'corn']	0.00	0.00	0.00	0
['copper' 'zinc' 'silver']	0.00	0.00	0.00	0
['acq' 'silver']	0.00	0.00	0.00	0
['acq' 'crude']	0.00	0.00	0.00	0
['lumber' 'plywood']	0.00	0.00	0.00	0
['veg-oil' 'sun-oil' 'corn-oil' 'rape-oil']	0.00	0.00	0.00	0
['coffee' 'ship']	0.00	0.00	0.00	0
['grain' 'corn' 'soybean' 'oilseed']	0.00	0.00	0.00	0
['grain' 'corn' 'sorghum' 'sunseed' 'oilseed']	0.00	0.00	0.00	0
['gas' 'fuel' 'crude']	0.00	0.00	0.00	0
['interest' 'retail' 'ipi']	0.00	0.00	0.00	0
['crude' 'nat-gas' 'iron-steel']	0.00	0.00	0.00	0
['ship' 'iron-steel']	0.00	0.00	0.00	0
['acq' 'gold']	0.00	0.00	0.00	1
['grain' 'wheat' 'sugar']	0.00	0.00	0.00	0

```
                    ['oilseed' 'rapeseed' 'soybean' 'sunseed']
0.00      0.00      0.00      0
                                      ['money-fx' 'reserves']
0.00      0.00      0.00      1
          ['grain' 'corn' 'rice' 'oilseed' 'soybean' 'orange']
0.00      0.00      0.00      0
                                          ['cocoa' 'coffee']
0.00      0.00      0.00      0
                                             ['acq' 'trade']
0.00      0.00      0.00      1
                                 ['earn' 'crude' 'nat-gas']
0.00      0.00      0.00      0
                                          ['earn' 'copper']
0.00      0.00      0.00      0
                                             ['crude' 'acq']
0.00      0.00      0.00      1
                ['grain' 'wheat' 'corn' 'oilseed' 'soybean']
0.00      0.00      0.00      0
                                    ['gnp' 'cpi' 'reserves']
0.00      0.00      0.00      0
          ['grain' 'wheat' 'oilseed' 'soybean' 'cotton' 'rice']
0.00      0.00      0.00      0
                                 ['l-cattle' 'livestock']
0.00      0.00      0.00      0
                                              ['gnp' 'cpi']
0.00      0.00      0.00      1
                                           ['rice' 'grain']
0.00      0.00      0.00      0
                                                ['nickel']
0.00      0.00      0.00      1
                                           ['ship' 'grain']
0.00      0.00      0.00      1
                                           ['inventories']
0.00      0.00      0.00      0
                              ['interest' 'gnp' 'ipi' 'wpi']
0.00      0.00      0.00      0
                             ['crude' 'gas' 'nat-gas' 'wpi']
0.00      0.00      0.00      0
                      ['wpi' 'gas' 'nat-gas' 'crude' 'heat']
0.00      0.00      0.00      0
                                      ['veg-oil' 'soy-oil']
0.00      0.00      0.00      2
                                              ['cpi' 'gas']
0.00      0.00      0.00      0
                                      ['pet-chem' 'crude']
0.00      0.00      0.00      0
                              ['iron-steel' 'zinc' 'lead']
0.00      0.00      0.00      0
```

```
                                          ['trade' 'veg-oil']
0.00      0.00      0.00       0
                                      ['reserves' 'money-fx']
0.00      0.00      0.00       0
                                        ['ipi' 'inventories']
0.00      0.00      0.00       0
                                     ['money-fx' 'yen' 'gnp']
0.00      0.00      0.00       0
                                ['money-fx' 'dlr' 'interest']
0.00      0.00      0.00       1
                                ['veg-oil' 'palm-oil' 'ship']
0.00      0.00      0.00       0
                                  ['money-fx' 'grain' 'corn']
0.00      0.00      0.00       0
                                                      ['cpu']
0.00      0.00      0.00       1
                              ['oilseed' 'veg-oil' 'soybean']
0.00      0.00      0.00       0
                                        ['money-fx' 'peseta']
0.00      0.00      0.00       0
                                             ['acq' 'copper']
0.00      0.00      0.00       2
                                              ['trade' 'gnp']
0.00      0.00      0.00       0
             ['grain' 'wheat' 'corn' 'cotton' 'sorghum' 'barley' 'corn']
0.00      0.00      0.00       0
                       ['grain' 'corn' 'wheat' 'oilseed' 'soybean']
0.00      0.00      0.00       3
                                     ['trade' 'ship' 'crude']
0.00      0.00      0.00       0
                                          ['trade' 'coffee']
0.00      0.00      0.00       2
                                                              ['grain'
'corn' 'wheat' 'soybean' 'oilseed' 'barley' 'sorghum' 'cotton'
 'rice']      0.00      0.00      0.00       0
                                      ['ship' 'crude' 'fuel']
0.00      0.00      0.00       0
                                      ['meal-feed' 'veg-oil']
0.00      0.00      0.00       0
                         ['grain' 'wheat' 'corn' 'sorghum']
0.00      0.00      0.00       0
                                             ['crude' 'fuel']
0.00      0.00      0.00       0
                         ['acq' 'nickel' 'strategic-metal']
0.00      0.00      0.00       0
                                        ['soybean' 'oilseed']
0.00      0.00      0.00       0
                                        ['oilseed' 'veg-oil']
```

0.00	0.00	0.00	1	['bop' 'trade' 'austdlr' 'money-fx' 'interest']
0.00	0.00	0.00	0	['money-fx' 'austdlr']
0.00	0.00	0.00	0	['interest' 'money-supply']
0.00	0.00	0.00	1	['corn' 'grain']
0.00	0.00	0.00	1	['ship' 'trade' 'crude']
0.00	0.00	0.00	0	['livestock' 'l-cattle' 'carcass']
0.00	0.00	0.00	0	['veg-oil' 'soybean']
0.00	0.00	0.00	1	['interest' 'bop' 'money-supply']
0.00	0.00	0.00	0	['ipi' 'jobs']
0.00	0.00	0.00	0	['housing' 'interest' 'gnp']
0.00	0.00	0.00	0	['trade' 'grain']
0.00	0.00	0.00	1	['grain' 'oilseed' 'soybean']
0.00	0.00	0.00	0	['grain' 'oilseed' 'soybean' 'carcass' 'corn' 'cotton' 'rice']
0.00	0.00	0.00	0	['interest' 'trade' 'gnp' 'bop' 'cpi']
0.00	0.00	0.00	0	['veg-oil' 'sun-oil' 'cotton-oil']
0.00	0.00	0.00	0	['veg-oil' 'coconut-oil']
0.00	0.00	0.00	0	['zinc' 'lead']
0.00	0.00	0.00	1	['l-cattle']
0.00	0.00	0.00	0	['silver']
0.00	0.00	0.00	0	['trade' 'cotton' 'iron-steel' 'naphtha' 'veg-oil' 'palm-oil']
0.00	0.00	0.00	0	['grain' 'corn' 'veg-oil']
0.00	0.00	0.00	0	['grain' 'corn' 'ship']
0.00	0.00	0.00	0	['crude' 'nat-gas' 'earn']
0.00	0.00	0.00	0	['fuel']

0.00	0.00	0.00	7	
				['jet']
0.00	0.00	0.00	1	
				['grain' 'corn' 'sorghum' 'sunseed' 'oilseed' 'soybean']
0.00	0.00	0.00	0	
				['money-fx' 'nzdlr']
0.00	0.00	0.00	2	
				['trade' 'bop' 'rubber' 'veg-oil' 'palm-oil']
0.00	0.00	0.00	0	
				['trade' 'bop' 'gnp']
0.00	0.00	0.00	1	
				['income']
0.00	0.00	0.00	4	
				['money-fx' 'income' 'money-supply']
0.00	0.00	0.00	0	
				['veg-oil' 'oilseed' 'soybean']
0.00	0.00	0.00	0	
				['money-fx' 'rand']
0.00	0.00	0.00	1	
				['crude' 'earn' 'nat-gas']
0.00	0.00	0.00	0	
				['money-supply' 'money-fx' 'interest']
0.00	0.00	0.00	0	
				['heat' 'naphtha' 'jet' 'fuel']
0.00	0.00	0.00	0	
				['grain' 'rice' 'wheat' 'tea' 'sugar']
0.00	0.00	0.00	0	
				['grain' 'meal-feed']
0.00	0.00	0.00	0	
				['money-fx' 'interest' 'money-supply']
0.00	0.00	0.00	0	
				['oilseed' 'soybean' 'veg-oil']
0.00	0.00	0.00	0	
				['trade' 'iron-steel' 'cotton']
0.00	0.00	0.00	0	
				['grain' 'corn' 'oilseed' 'soybean']
0.00	0.00	0.00	3	
				['silver' 'copper' 'lead' 'zinc' 'gold' 'strategic-metal']
0.00	0.00	0.00	0	
				['lead' 'zinc']
0.00	0.00	0.00	2	
				['grain' 'rice' 'corn' 'cotton']
0.00	0.00	0.00	0	
				['livestock' 'pork-belly']
0.00	0.00	0.00	0	
				['oilseed' 'meal-feed']
0.00	0.00	0.00	0	
				['grain' 'wheat' 'oilseed' 'soybean' 'veg-oil']

0.00	0.00	0.00	0	['sugar' 'grain' 'corn']
0.00	0.00	0.00	0	['crude' 'nat-gas' 'fuel']
0.00	0.00	0.00	0	['gnp' 'ringgit']
0.00	0.00	0.00	0	['oilseed' 'coconut']
0.00	0.00	0.00	0	['trade' 'oilseed' 'grain']
0.00	0.00	0.00	0	['interest' 'gnp' 'trade']
0.00	0.00	0.00	0	['meal-feed' 'grain' 'corn' 'sorghum' 'oilseed' 'soybean']
0.00	0.00	0.00	0	['trade' 'coffee' 'rubber' 'palm-oil']
0.00	0.00	0.00	0	['gnp' 'interest' 'money-fx' 'trade']
0.00	0.00	0.00	0	['veg-oil' 'palm-oil' 'rape-oil' 'ship']
0.00	0.00	0.00	0	['grain' 'corn' 'rice']
0.00	0.00	0.00	2	['gnp' 'cpi' 'reserves' 'grain']
0.00	0.00	0.00	0	['acq' 'grain' 'corn']
0.00	0.00	0.00	0	['gnp' 'trade' 'jobs' 'retail']
0.00	0.00	0.00	0	['money-fx' 'dlr' 'trade']
0.00	0.00	0.00	1	['grain' 'barley' 'oilseed' 'rapeseed']
0.00	0.00	0.00	0	['grain' 'corn' 'barley']
0.00	0.00	0.00	0	['trade' 'cotton' 'grain' 'corn' 'wheat' 'oilseed']
0.00	0.00	0.00	0	['ship' 'grain' 'oilseed']
0.00	0.00	0.00	0	['grain' 'oilseed' 'corn']
0.00	0.00	0.00	0	['trade' 'veg-oil' 'coconut-oil']
0.00	0.00	0.00	0	['reserves' 'trade' 'money-fx']
0.00	0.00	0.00	0	['ipi' 'trade']
0.00	0.00	0.00	0	['cpi' 'wpi']

0.00	0.00	0.00	0	['ship' 'livestock']
0.00	0.00	0.00	0	['retail' 'jobs' 'gnp' 'inventories' 'trade' 'cpi']
0.00	0.00	0.00	0	['grain' 'rice' 'corn' 'cotton' 'wheat' 'sorghum' 'barley' 'oat']
0.00	0.00	0.00	0	['grain' 'sugar' 'carcass' 'livestock']
0.00	0.00	0.00	0	['acq' 'gold' 'silver' 'zinc' 'lead']
0.00	0.00	0.00	0	['acq' 'gold' 'silver' 'lead' 'zinc']
0.00	0.00	0.00	0	['money-fx' 'stg' 'interest']
0.00	0.00	0.00	0	['earn' 'ship']
0.00	0.00	0.00	0	['ship' 'coffee']
0.00	0.00	0.00	0	['earn' 'crude' 'gas']
0.00	0.00	0.00	0	['earn' 'crude' 'pet-chem']
0.00	0.00	0.00	0	['trade' 'hog' 'carcass' 'livestock']
0.00	0.00	0.00	0	['pet-chem' 'nat-gas']
0.00	0.00	0.00	0	['coffee' 'tea' 'rubber']
0.00	0.00	0.00	0	['veg-oil' 'sun-oil']
0.00	0.00	0.00	0	['silver' 'copper']
0.00	0.00	0.00	0	['tea' 'orange']
0.00	0.00	0.00	0	['rand']
0.00	0.00	0.00	0	['platinum']
0.00	0.00	0.00	2	['sugar' 'ship']
0.00	0.00	0.00	2	['grain' 'corn' 'oilseed' 'soybean' 'sorghum' 'sunseed']
0.00	0.00	0.00	0	['trade' 'iron-steel']
0.00	0.00	0.00	2	['money-fx' 'yen' 'interest']
0.00	0.00	0.00	0	['jobs' 'trade']

```
0.00      0.00      0.00        0
                                        ['saudriyal' 'money-fx']
0.00      0.00      0.00        0
                               ['veg-oil' 'meal-feed' 'oilseed']
0.00      0.00      0.00        0
                           ['trade' 'bop' 'interest' 'money-fx']
0.00      0.00      0.00        0
                     ['trade' 'bop' 'interest' 'stg' 'money-fx']
0.00      0.00      0.00        0
                                               ['trade' 'crude']
0.00      0.00      0.00        0
                                                ['trade' 'jobs']
0.00      0.00      0.00        1
                                                 ['trade' 'acq']
0.00      0.00      0.00        0
                                     ['coffee' 'cocoa' 'sugar']
0.00      0.00      0.00        0
                                          ['iron-steel' 'ship']
0.00      0.00      0.00        0
                            ['grain' 'sugar' 'livestock' 'carcass']
0.00      0.00      0.00        0
                                        ['grain' 'corn' 'wheat'
'oilseed' 'soybean' 'meal-feed' 'veg-oil'
 'soy-oil' 'sorghum' 'barley']      0.00      0.00      0.00        0
          ['trade' 'grain' 'wheat' 'tea' 'coffee' 'iron-steel' 'crude']
0.00      0.00      0.00        0
                                         ['iron-steel' 'trade']
0.00      0.00      0.00        0
                    ['oilseed' 'veg-oil' 'castorseed' 'castor-oil']
0.00      0.00      0.00        0
                                         ['veg-oil' 'rape-oil']
0.00      0.00      0.00        0
                                             ['gnp' 'money-fx']
0.00      0.00      0.00        0
                                             ['money-fx' 'dfl']
0.00      0.00      0.00        0
                                            ['carcass' 'sugar']
0.00      0.00      0.00        0
          ['trade' 'grain' 'wheat' 'coffee' 'tea' 'iron-steel' 'crude']
0.00      0.00      0.00        0
                                  ['gold' 'silver' 'zinc' 'lead']
0.00      0.00      0.00        0
                                             ['gnp' 'reserves']
0.00      0.00      0.00        0
                                            ['gold' 'money-fx']
0.00      0.00      0.00        0
                         ['cpi' 'crude' 'nat-gas' 'heat' 'propane']
0.00      0.00      0.00        0
```

```
                                                   ['ship' 'acq']
0.00      0.00      0.00       0
                                                ['money-fx' 'lit']
0.00      0.00      0.00       0
                              ['money-fx' 'rupiah' 'dmk' 'yen']
0.00      0.00      0.00       0
     ['oilseed' 'sunseed' 'rapeseed' 'veg-oil' 'sun-oil' 'rape-oil']
0.00      0.00      0.00       0
                                             ['rubber' 'pet-chem']
0.00      0.00      0.00       0
                                              ['nzdlr' 'austdlr']
0.00      0.00      0.00       0
                         ['grain' 'wheat' 'corn' 'sorghum' 'cotton']
0.00      0.00      0.00       0
                                   ['iron-steel' 'alum' 'earn']
0.00      0.00      0.00       0
                                                         ['dlr']
0.00      0.00      0.00       3
                                          ['acq' 'sugar' 'crude']
0.00      0.00      0.00       0
           ['tin' 'sugar' 'grain' 'wheat' 'cocoa' 'coffee' 'rubber']
0.00      0.00      0.00       0
                                     ['crude' 'nat-gas' 'sugar']
0.00      0.00      0.00       0
                                             ['trade' 'yen' 'dlr']
0.00      0.00      0.00       0
                                      ['coffee' 'sugar' 'cocoa']
0.00      0.00      0.00       0
                                                    ['gnp' 'lei']
0.00      0.00      0.00       0
                                           ['crude' 'gas' 'heat']
0.00      0.00      0.00       0
                                              ['cotton' 'sorghum']
0.00      0.00      0.00       0
                                       ['grain' 'rice' 'cotton']
0.00      0.00      0.00       1
                             ['veg-oil' 'grain' 'wheat' 'cotton']
0.00      0.00      0.00       0
                                        ['ship' 'grain' 'wheat']
0.00      0.00      0.00       0
                                       ['grain' 'corn' 'cotton']
0.00      0.00      0.00       0
                                  ['interest' 'gnp' 'money-fx']
0.00      0.00      0.00       0
                       ['palm-oil' 'palmkernel' 'oilseed' 'veg-oil']
0.00      0.00      0.00       0
                                                 ['coffee' 'acq']
0.00      0.00      0.00       0
```

```
                                           ['coconut' 'oilseed']
0.00      0.00      0.00        0
                       ['ship' 'grain' 'oilseed' 'veg-oil' 'meal-feed']
0.00      0.00      0.00        0
                                                  ['heat' 'gas']
0.00      0.00      0.00        0
                                           ['gnp' 'coffee' 'bop']
0.00      0.00      0.00        0
          ['wheat' 'barley' 'corn' 'rapeseed' 'grain' 'oilseed' 'ship']
0.00      0.00      0.00        0
                                      ['wheat' 'grain' 'veg-oil']
0.00      0.00      0.00        0
                                            ['earn' 'iron-steel']
0.00      0.00      0.00        0
                               ['grain' 'oilseed' 'corn' 'soybean']
0.00      0.00      0.00        0
                                                ['gnp' 'coffee']
0.00      0.00      0.00        0
                         ['corn' 'sunseed' 'soybean' 'grain' 'oilseed']
0.00      0.00      0.00        0
                       ['money-fx' 'dlr' 'stg' 'skr' 'nkr' 'dkr' 'dmk']
0.00      0.00      0.00        0
                                         ['acq' 'strategic-metal']
0.00      0.00      0.00        0
                           ['wheat' 'corn' 'soybean' 'grain' 'oilseed']
0.00      0.00      0.00        0
                                        ['corn' 'sorghum' 'grain']
0.00      0.00      0.00        0
                                           ['rapeseed' 'oilseed']
0.00      0.00      0.00        0
                                                         ['stg']
0.00      0.00      0.00        0
                                               ['stg' 'money-fx']
0.00      0.00      0.00        0
                ['grain' 'oilseed' 'sunseed' 'corn' 'soybean' 'sorghum']
0.00      0.00      0.00        0
                              ['grain' 'wheat' 'soybean' 'oilseed']
0.00      0.00      0.00        0
                                                      ['oilseed'
'cotton' 'wheat' 'grain' 'sunseed' 'linseed' 'rapeseed'
 'soybean' 'groundnut']       0.00      0.00      0.00        0
                                          ['groundnut' 'oilseed']
0.00      0.00      0.00        0
                                               ['interest' 'dlr']
0.00      0.00      0.00        0
                         ['meal-feed' 'sun-meal' 'lin-meal' 'soy-meal']
0.00      0.00      0.00        0
                                                ['grain' 'wheat' 'corn'
```

```
'sugar' 'carcass' 'livestock' 'groundnut'
 'oilseed' 'cotton' 'veg-oil']       0.00       0.00       0.00          0
                                             ['money-fx' 'interest' 'dlr']
0.00       0.00       0.00          1
                                                         ['trade' 'cocoa']
0.00       0.00       0.00          0
                                             ['money-fx' 'interest' 'stg']
0.00       0.00       0.00          0
                      ['oilseed' 'soybean' 'soy-meal' 'veg-oil' 'soy-oil']
0.00       0.00       0.00          0
                                                    ['money-supply' 'wpi']
0.00       0.00       0.00          0
                                                    ['pet-chem' 'oilseed']
0.00       0.00       0.00          0
                                             ['interest' 'money-fx' 'stg']
0.00       0.00       0.00          0
                                       ['money-fx' 'dlr' 'yen' 'interest']
0.00       0.00       0.00          0
                         ['copper' 'lead' 'zinc' 'silver' 'nickel' 'alum']
0.00       0.00       0.00          0
                                                                  ['wool']
0.00       0.00       0.00          0
                                                         ['austdlr' 'dmk']
0.00       0.00       0.00          0
                                                    ['iron-steel' 'crude']
0.00       0.00       0.00          0
                                                    ['money-supply' 'gnp']
0.00       0.00       0.00          0
                                          ['carcass' 'livestock' 'orange']
0.00       0.00       0.00          0
                                                    ['palm-oil' 'veg-oil']
0.00       0.00       0.00          0
                                                  ['meal-feed' 'soy-meal']
0.00       0.00       0.00          2
                                                                   ['tea']
0.00       0.00       0.00          3
                                                       ['pet-chem' 'ship']
0.00       0.00       0.00          0
                                                       ['cpi' 'gnp' 'ipi']
0.00       0.00       0.00          0
                                                  ['soy-meal' 'meal-feed']
0.00       0.00       0.00          0
                                                     ['plywood' 'lumber']
0.00       0.00       0.00          0
                                      ['veg-oil' 'palm-oil' 'coconut-oil']
0.00       0.00       0.00          0
                                                        ['barley' 'grain']
0.00       0.00       0.00          0
```

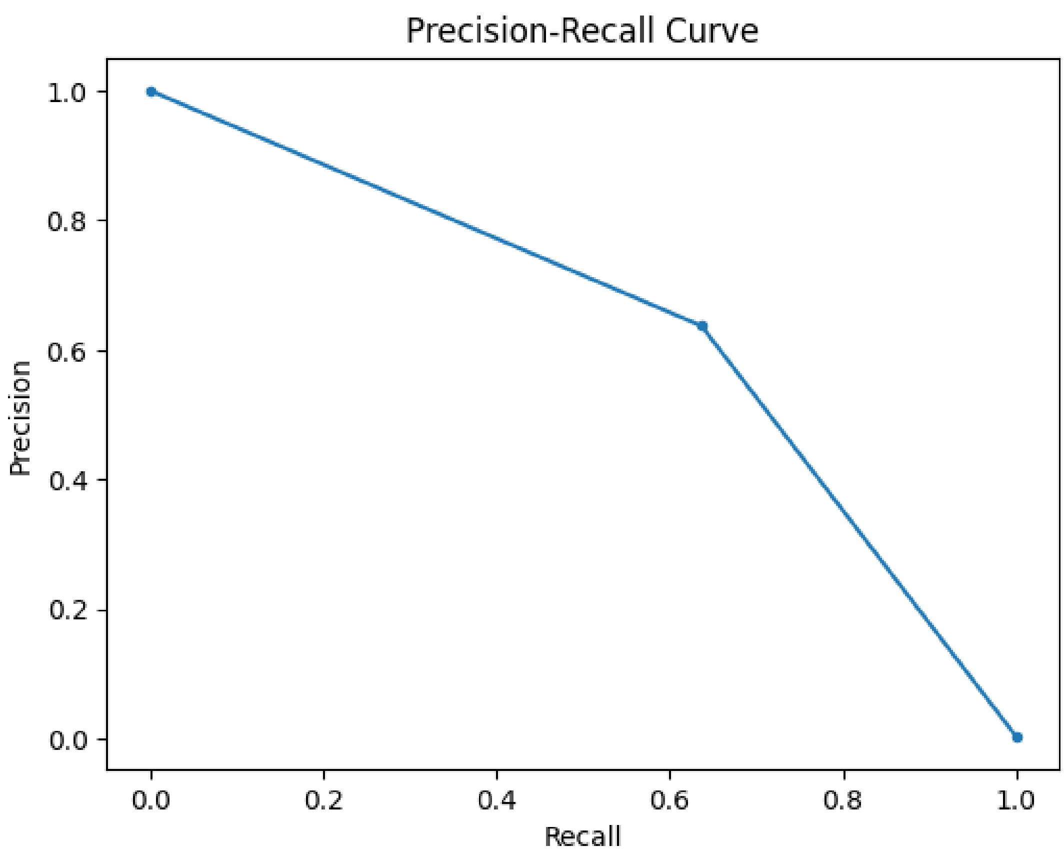
```
                                                   ['nat-gas' 'propane']
0.00       0.00       0.00         0
                                    ['grain' 'oilseed' 'soy-oil' 'corn']
0.00       0.00       0.00         0
['oilseed' 'rapeseed' 'grain' 'wheat' 'corn' 'palm-oil' 'soy-oil' 'ship']
0.00       0.00       0.00         0
                                  ['grain' 'oilseed' 'wheat' 'rapeseed']
0.00       0.00       0.00         0
                                                     ['gold' 'reserves']
0.00       0.00       0.00         0
                                                      ['wheat' 'barley']
0.00       0.00       0.00         0
                       ['grain' 'wheat' 'corn' 'sorghum' 'barley' 'oat']
0.00       0.00       0.00         0
                             ['grain' 'wheat' 'corn' 'sorghum' 'barley']
0.00       0.00       0.00         0
                                                       ['copper' 'zinc']
0.00       0.00       0.00         0
                                       ['oilseed' 'soybean' 'soy-oil']
0.00       0.00       0.00         0
                                              ['dlr' 'dmk' 'money-fx']
0.00       0.00       0.00         0
                                           ['money-supply' 'money-fx']
0.00       0.00       0.00         0
                                                    ['copper' 'nickel']
0.00       0.00       0.00         0
                                                  ['sugar' 'livestock']
0.00       0.00       0.00         0
                                  ['grain' 'corn' 'oilseed' 'livestock']
0.00       0.00       0.00         0
                                           ['grain' 'wheat' 'veg-oil']
0.00       0.00       0.00         0
                                ['gold' 'silver' 'copper' 'zinc' 'lead']
0.00       0.00       0.00         0
                                                ['cruzado' 'money-fx']
0.00       0.00       0.00         0
                                                ['gas' 'crude' 'fuel']
0.00       0.00       0.00         0
                                    ['money-fx' 'dlr' 'yen' 'can' 'stg']
0.00       0.00       0.00         0
                         ['dlr' 'money-fx' 'trade' 'cpi' 'money-supply']
0.00       0.00       0.00         0

                                                               micro avg
0.64       0.67       0.65      3156
                                                               macro avg
0.02       0.02       0.02      3156
                                                            weighted avg
```

```
0.55      0.67      0.60      3156
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels
with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in labels
with no true nor predicted samples. Use `zero_division` parameter to control
this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels
with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in labels
with no true nor predicted samples. Use `zero_division` parameter to control
this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels
with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 in labels
with no true nor predicted samples. Use `zero_division` parameter to control
this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

[ ]:
```
from sklearn.preprocessing import LabelBinarizer
from sklearn.metrics import precision_recall_curve, classification_report
```

```python
import matplotlib.pyplot as plt

# Binarize the labels
lb = LabelBinarizer()
y_test_bin = lb.fit_transform(y_test)
y_pred_bin = lb.transform(y_pred)

# Compute precision-recall curve for each class
precision, recall, _ = precision_recall_curve(y_test_bin.ravel(), y_pred_bin.
 ↪ravel())

# Plot Precision-Recall curve
plt.plot(recall, precision, marker='.')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve')
plt.show()
```



Discussion

The dataset consists of titles of news articles. The dataset was already split into train and test. The objective of this Classification was to classify finance news.

We have used NLTK for stopwords, spaCy for tokenization, PorterStemmer for stemming and WordNetTokenizer for tokenization.

We have used Naive Bayes to train the model. Since there were around 750 unique words, the confusion matrix was unpleasant. Therefore we have used simple classification report and Precision VS Recall curve.

The metrics are as follows:

Accuracy: 0.6369

Precision: 0.5276

Recall: 0.6369

F1-Score: 0.5701

The Precision VS Recall curve shows that the model is neither very good (as it is not extending to the right), nor very poor (as it is not a straight diagonal), therfore the area under the curve is neither too poor nor too good.

Performance is affected because we have used the top 100 most appeared words only instead of all the words appearing in the dataset as you can see in the word cloud.

Submission Guidelines:

Google Colab Notebook Submission:

Save your notebook as NLP_Text_Classification_YourName.ipynb.

Ensure all code cells are executed, and the output is visible.

Include proper documentation and comments explaining each step.

Report Submission (Optional):

Prepare a short report (2-3 pages) summarizing your approach, findings, and model performance.

Upload the report along with the Colab Notebook.

Grading Criteria:

Correct implementation of NLP preprocessing (30%)

Effective use of vectorization techniques (20%)

Model accuracy and performance evaluation (30%)

Code clarity, documentation, and presentation (20%)

[ ]:
```
```

Declaration

I, Siddhant Mishra, confirm that the work submitted in this assignment is my own and has been completed following academic integrity guidelines. The code is uploaded on my GitHub repository account, and the repository link is provided below:

Signature: Siddhant Mishra

Submission Checklist

Ultralitycs Platform Documentsation Like hel file for Given Task

Code file (Python Notebook or Script)

Dataset or link to the dataset

Visualizations (if applicable)

Screenshots of model performance metrics

Readme File

Evaluation Metrics Details and discussion