Q1) Word similarity

```
import numpy as np
import pandas as pd
from itertools import combinations
```

The problem is to match the user's free-form input against a pre-determined list of banks. For example, user input 'bawag bank' should be matched to 'BAWAG Group AG'.

```
# List of banks to compare
banks =    ['Sberbank Europe AG',
           'BAWAG Group AG',
           'Raiffeisenbankengruppe OÖ Verbund eGen',
           'Raiffeisen Bank International AG',
           'Volksbanken Verbund',
           'Erste Group Bank AG',
           'KBC Groep',
           'Investeringsmaatschappij Argenta',
           'Belfius Bank',
           'AXA Bank Belgium',
           'The Bank of New York Mellon SA/NV',
           'First Investment Bank AD',
           'RCB Bank Ltd',
           'Bank of Cyprus Holdings Public Limited Company',
           'Hellenic Bank Public Company Limited',
           'DekaBank Deutsche Girozentrale',
           'Erwerbsgesellschaft der S-Finanzgruppe mbH & Co. KG',
           'UBS Europe SE',
           'DEUTSCHE APOTHEKER- UND ÄRZTEBANK EG',
           'Volkswagen Bank Gesellschaft mit beschränkter Haftung',
           'Münchener Hypothekenbank eG',
           'DZ BANK AG Deutsche Zentral-Genossenschaftsbank, Frankfurt
am Main',
           'HASPA Finanzholding',
           'State Street Europe Holdings Germany S.a.r.l. & Co. KG',
           'J.P. Morgan AG',
           'DEUTSCHE BANK AKTIENGESELLSCHAFT',
           'COMMERZBANK Aktiengesellschaft',
           'Landesbank Baden-Württemberg',
           'Landesbank Hessen-Thüringen Girozentrale',
           'Norddeutsche Landesbank - Girozentrale -',
           'Deutsche Pfandbriefbank AG',
           'Aareal Bank AG',
           'Hamburg Commercial Bank AG',
           'Bayerische Landesbank',
           'Jyske Bank A/S',
           'Sydbank A/S',
           'Nykredit Realkredit A/S',
           'Danske Bank A/S',
           'Luminor Holding AS',
           'Abanca Corporacion Bancaria S.A.',
           'Banco Santander S.A.',
           'Ibercaja Banco S.A.',
           'Kutxabank S.A',
           'Unicaja Banco S.A.',
           'CaixaBank S.A.',
           'Banco de Crédito Social Cooperativo',
           'Banco Bilbao Vizcaya Argentaria S.A.',
           'Banco de Sabadell S.A.',
           'Bankinter S.A.',
           'Kuntarahoitus Oyj',
           'Nordea Bank Abp',
           'OP Osuuskunta',
           'SFIL',
           'RCI Banque',
           'Confédération Nationale du Crédit Mutuel',
           'La Banque Postale',
           'Bpifrance',
           "C.R.H. - Caisse de refinancement de l'habitat",
           'HSBC Continental Europe',
           'Groupe BPCE',
           'Groupe Crédit Agricole',
           'Société générale',
           'BNP Paribas',
           'ALPHA SERVICES AND HOLDINGS S.A.',
           'National Bank of Greece S.A.',
           'Eurobank Ergasias Services and Holdings S.A.',
           'Piraeus Financial Holdings',
           'OTP-csoport',
           'Magyar Bankholding',
           'Barclays Bank Ireland plc',
           'Citibank Holdings Ireland Limited',
           'AIB Group plc',
           'Bank of Ireland Group plc',
           'Ulster Bank Ireland Designated Activity Company',
           'Bank of America Europe Designated Activity Company',
           'Íslandsbanki hf.',
           'Landsbankinn hf.',
           'Arion banki hf',
           'Intesa Sanpaolo S.p.A.',
           'Gruppo Bancario Finecobank  ',
           'UniCredit S.p.A.',
           'Gruppo Bancario Mediolanum  ',
           'Credito Emiliano Holding S.p.A.',
           'Banco BPM SpA',
           'Banca Popolare di Sondrio, Società Cooperativa per Azioni',
           'Banca Monte dei Paschi di Siena S.p.A.',
           'CASSA CENTRALE BANCA',
           'ICCREA BANCA S.P.A.',
           'Mediobanca - Banca di Credito Finanziario S.p.A.',
           'Akcine bendrove Šiauliu bankas',
           'Precision Capital S.A.',
           'RBC Investor Services Bank S.A.',
           'J.P. Morgan Bank Luxembourg S.A.',
           'Banque Internationale à Luxembourg',
           'Banque et Caisse d´Epargne de l´Etat, Luxembourg',
           'Akciju sabiedriba "Citadele banka"',
           'MDB Group Limited',
           'Bank of Valletta Plc',
           'HSBC Bank Malta p.l.c.',
           'BNG Bank N.V.',
           'ING Groep N.V.',
           'LP Group B.V.',
           'de Volksbank N.V.',
           'ABN AMRO Bank N.V.',
           'Coöperatieve Rabobank U.A.',
           'Nederlandse Waterschapsbank N.V.',
           'Bank Polska Kasa Opieki S.A.',
           'Powszechna Kasa Oszczednosci Bank Polski S.A.',
           'LSF Nani Investments S.à r.l.',
           'Banco Comercial Português SA',
           'Caixa Geral de Depósitos SA',
           'Banca Transilvania',
           'Länförsäkringar Bank AB (publ)',
           'Kommuninvest - group',
           'Skandinaviska Enskilda Banken - group',
           'SBAB Bank AB - group',
           'Swedbank - group',
           'Svenska Handelsbanken - group',
           'Biser Topco S.à r.l.',
           'Nova Ljubljanska Banka d.d. Ljubljana']

 # Examples of search strings
 s1 = 'Bawag bank' # other options: 'Bawag bank', 'Erste', 'Raiffaisen
 bank'
```

Approach 1 to increase similarity score

```
from sklearn.feature_extraction.text import TfidfVectorizer

def preprocess(text):
    stop_words = set(stopwords.words('english'))
    tokens = [word.lower() for word in word_tokenize(text) if
word.lower() not in stop_words]
    return ' '.join(tokens)
```

```
s1_processed = preprocess(s1)

banks_processed = [preprocess(bank) for bank in banks]

tfidf_vectorizer = TfidfVectorizer()

bank_tfidf = tfidf_vectorizer.fit_transform(banks_processed)

s1_tfidf = tfidf_vectorizer.transform([s1_processed])

similarities = cosine_similarity(s1_tfidf, bank_tfidf)

df2 = pd.DataFrame({'Bank 1': s1, 'Bank 2': banks, 'Similarity': 
similarities.flatten()})

df2 = df2.sort_values(by=['Similarity'], ascending=False)

print(df2.head())
```

```
        Bank 1           Bank 2  Similarity
1   Bawag bank   BAWAG Group AG    0.663232
34  Bawag bank   Jyske Bank A/S    0.168777
8   Bawag bank     Belfius Bank    0.168777
99  Bawag bank    BNG Bank N.V.    0.168777
37  Bawag bank  Danske Bank A/S    0.168777
```

```
idx = df2['Bank 2'].isin(['BAWAG Group AG'])

sorted_df = df2[idx].sort_values(by=['Similarity'], ascending=False)

print(sorted_df.head())
```

```
       Bank 1          Bank 2  Similarity
1  Bawag bank  BAWAG Group AG    0.663232
```

Approach 2 to increase similarity score

```
!pip install sentence-transformers
from sentence_transformers import SentenceTransformer
import numpy as np

model = SentenceTransformer('bert-base-nli-mean-tokens')

s1_embedding = model.encode([s1])
bank_embeddings = model.encode(banks)

similarities = []
for bank_embedding in bank_embeddings:
    similarity = np.dot(s1_embedding, bank_embedding.T) / 
(np.linalg.norm(s1_embedding) * np.linalg.norm(bank_embedding))
    similarities.append(similarity[0])
```

```
df2 = pd.DataFrame({'Bank 1': s1, 'Bank 2': banks, 'Similarity':
similarities})

df2 = df2.sort_values(by=['Similarity'], ascending=False)

print(df2.head())

Collecting sentence-transformers
  Downloading sentence_transformers-2.5.1-py3-none-any.whl (156 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/156.5 kB ? eta -:--:--
━━━━━━━━━━━━━━━╺━━━━━━━━━━━━━━━━━━━━━━━━ 61.4/156.5 kB 1.6 MB/s eta
0:00:01 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 156.5/156.5 kB 2.4
MB/s eta 0:00:00
ent already satisfied: transformers<5.0.0,>=4.32.0 in
/usr/local/lib/python3.10/dist-packages (from sentence-transformers)
(4.38.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-
packages (from sentence-transformers) (4.66.2)
Requirement already satisfied: torch>=1.11.0 in
/usr/local/lib/python3.10/dist-packages (from sentence-transformers)
(2.1.0+cu121)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from sentence-transformers)
(1.25.2)
Requirement already satisfied: scikit-learn in
/usr/local/lib/python3.10/dist-packages (from sentence-transformers)
(1.2.2)
Requirement already satisfied: scipy in
/usr/local/lib/python3.10/dist-packages (from sentence-transformers)
(1.11.4)
Requirement already satisfied: huggingface-hub>=0.15.1 in
/usr/local/lib/python3.10/dist-packages (from sentence-transformers)
(0.20.3)
Requirement already satisfied: Pillow in
/usr/local/lib/python3.10/dist-packages (from sentence-transformers)
(9.4.0)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1-
>sentence-transformers) (3.13.1)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1-
>sentence-transformers) (2023.6.0)
Requirement already satisfied: requests in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1-
>sentence-transformers) (2.31.0)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1-
>sentence-transformers) (6.0.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
```

```
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1-
>sentence-transformers) (4.10.0)
Requirement already satisfied: packaging>=20.9 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.15.1-
>sentence-transformers) (23.2)
Requirement already satisfied: sympy in
/usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-
transformers) (1.12)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-
transformers) (3.2.1)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-
transformers) (3.1.3)
Requirement already satisfied: triton==2.1.0 in
/usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-
transformers) (2.1.0)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from
transformers<5.0.0,>=4.32.0->sentence-transformers) (2023.12.25)
Requirement already satisfied: tokenizers<0.19,>=0.14 in
/usr/local/lib/python3.10/dist-packages (from
transformers<5.0.0,>=4.32.0->sentence-transformers) (0.15.2)
Requirement already satisfied: safetensors>=0.4.1 in
/usr/local/lib/python3.10/dist-packages (from
transformers<5.0.0,>=4.32.0->sentence-transformers) (0.4.2)
Requirement already satisfied: joblib>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn->sentence-
transformers) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn->sentence-
transformers) (3.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.11.0-
>sentence-transformers) (2.1.5)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.15.1->sentence-transformers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.15.1->sentence-transformers) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.15.1->sentence-transformers) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.15.1->sentence-transformers) (2024.2.2)
Requirement already satisfied: mpmath>=0.19 in
/usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.11.0-
>sentence-transformers) (1.3.0)
Installing collected packages: sentence-transformers
Successfully installed sentence-transformers-2.5.1
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/
_token.py:88: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your
settings tab (https://huggingface.co/settings/tokens), set it as
secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to
access public models or datasets.
  warnings.warn(
```

```
{"model_id":"56794dfcabff406eafbdd9a70a22f198","version_major":2,"vers
ion_minor":0}

{"model_id":"550de3935c094c0197b8193cabff1b77","version_major":2,"vers
ion_minor":0}

{"model_id":"b604ab86d4ac431e952f6934c65c2547","version_major":2,"vers
ion_minor":0}

{"model_id":"29b6b1f8f3714909a73c991ab7375762","version_major":2,"vers
ion_minor":0}

{"model_id":"738243b039fd415c97ef1459d1984a8f","version_major":2,"vers
ion_minor":0}

{"model_id":"0941df612de24353a38df6e4f533ebe5","version_major":2,"vers
ion_minor":0}
```

```
/usr/local/lib/python3.10/dist-packages/torch/_utils.py:831:
UserWarning: TypedStorage is deprecated. It will be removed in the
future and UntypedStorage will be the only storage class. This should
only matter to you if you are using storages directly.  To access
UntypedStorage directly, use tensor.untyped_storage() instead of
tensor.storage()
  return self.fget.__get__(instance, owner)()
```

```
{"model_id":"ce449f470ee54fad921aacb0e070fc71","version_major":2,"vers
ion_minor":0}

{"model_id":"e7a0dd9a1e0042a5ae4e8b2e128d4c10","version_major":2,"vers
ion_minor":0}

{"model_id":"c4ec5339cf2d4206b9e3daaa66f5ae7c","version_major":2,"vers
ion_minor":0}

{"model_id":"2b395e9bf0dc498ba50a9aec45747f0c","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"d50b219e7b6349c699d042e78e707513","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"ce00119da33048e8976978bad468aaac","version_major":2,"vers
ion_minor":0}
```

```
        Bank 1          Bank 2  Similarity
8   Bawag bank    Belfius Bank    0.888046
48  Bawag bank  Bankinter S.A.    0.855527
77  Bawag bank  Arion banki hf    0.847288
31  Bawag bank  Aareal Bank AG    0.846864
12  Bawag bank    RCB Bank Ltd    0.836518
```

```
idx = df2['Bank 2'].isin(['BAWAG Group AG'])

sorted_df = df2[idx].sort_values(by=['Similarity'], ascending=False)

print(sorted_df.head())
```

```
       Bank 1          Bank 2  Similarity
1  Bawag bank  BAWAG Group AG    0.782286
```

Q2) Sentiment Analysis

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer,
TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression, Perceptron
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
import pandas as pd


dataset = pd.read_csv('Q2 Sentiment Analysis Dataset.csv',
encoding='latin1')

X = dataset['text']
y = dataset['sentiment']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)


def evaluate_classifier(y_true, y_pred):
    accuracy = accuracy_score(y_true, y_pred)
    precision = precision_score(y_true, y_pred, average='macro')
    recall = recall_score(y_true, y_pred, average='macro')
    f1 = f1_score(y_true, y_pred, average='macro')
    return accuracy, precision, recall, f1


def train_and_evaluate_classifier(classifier, X_train, X_test,
y_train, y_test, feature_extraction):
    if feature_extraction == 'bag_of_words':
        vectorizer = CountVectorizer()
    elif feature_extraction == 'tfidf':
        vectorizer = TfidfVectorizer()
    elif feature_extraction == 'unigram':
        vectorizer = CountVectorizer(ngram_range=(1, 1))
    elif feature_extraction == 'bigram':
        vectorizer = CountVectorizer(ngram_range=(2, 2))
    elif feature_extraction == 'trigram':
        vectorizer = CountVectorizer(ngram_range=(3, 3))

    X_train_features = vectorizer.fit_transform(X_train)
    X_test_features = vectorizer.transform(X_test)

    classifier.fit(X_train_features, y_train)
    y_pred = classifier.predict(X_test_features)

    accuracy, precision, recall, f1 = evaluate_classifier(y_test,
y_pred)
    return accuracy, precision, recall, f1


classifiers = {
    'Naive Bayes': MultinomialNB(),
    'Logistic Regression': LogisticRegression(),
    'Random Forest': RandomForestClassifier(),
    'SVM': SVC(),
    'Perceptron': Perceptron()
}


feature_extraction_methods = ['bag_of_words', 'tfidf', 'unigram',
'bigram', 'trigram']


results = []
for classifier_name, classifier in classifiers.items():
    for feature_extraction_method in feature_extraction_methods:
        accuracy, precision, recall, f1 =
train_and_evaluate_classifier(classifier, X_train, X_test, y_train,
y_test, feature_extraction_method)
        results.append([classifier_name, feature_extraction_method,
accuracy, precision, recall, f1])
```

```
results_df = pd.DataFrame(results, columns=['Classifier', 'Feature
Extraction', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic
.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic
.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic
.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic
.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
n.py:1344: 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, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
```

```
n.py:1344: 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, msg_start, len(result))
```

```
print(results_df)
```

```
             Classifier Feature Extraction  Accuracy  Precision
Recall  \
0           Naive Bayes       bag_of_words  0.726221   0.525596
0.429674
1           Naive Bayes              tfidf  0.751928   0.632412
0.423037
2           Naive Bayes            unigram  0.726221   0.525596
0.429674
3           Naive Bayes             bigram  0.728792   0.469111
0.425637
4           Naive Bayes            trigram  0.661954   0.467093
0.363156
5   Logistic Regression       bag_of_words  0.748072   0.509881
0.451867
6   Logistic Regression              tfidf  0.740360   0.544822
0.434823
7   Logistic Regression            unigram  0.748072   0.509881
0.451867
8   Logistic Regression             bigram  0.704370   0.535677
0.408070
9   Logistic Regression            trigram  0.651671   0.550386
0.352615
10        Random Forest       bag_of_words  0.740360   0.584411
0.440504
11        Random Forest              tfidf  0.736504   0.552356
0.429567
12        Random Forest            unigram  0.755784   0.576828
0.444010
13        Random Forest             bigram  0.694087   0.543023
0.399105
14        Random Forest            trigram  0.586118   0.470612
0.366430
15                  SVM       bag_of_words  0.735219   0.586433
0.418947
16                  SVM              tfidf  0.750643   0.587932
0.440222
17                  SVM            unigram  0.735219   0.586433
0.418947
18                  SVM             bigram  0.697943   0.556928
0.400256
19                  SVM            trigram  0.601542   0.581337
0.304559
20           Perceptron       bag_of_words  0.733933   0.545199
```

```
0.463601
21           Perceptron              tfidf  0.700514   0.506361
0.473202
22           Perceptron            unigram  0.733933   0.545199
0.463601
23           Perceptron             bigram  0.651671   0.440451
0.408660
24           Perceptron            trigram  0.620823   0.420938
0.366162

    F1 Score
0   0.425850
1   0.422528
2   0.425850
3   0.429719
4   0.367223
5   0.462568
6   0.444730
7   0.462568
8   0.414065
9   0.357948
10  0.454143
11  0.433620
12  0.454490
13  0.403780
14  0.349526
15  0.421970
16  0.451234
17  0.421970
18  0.397639
19  0.283929
20  0.485623
21  0.480116
22  0.485623
23  0.417312
24  0.377225
```