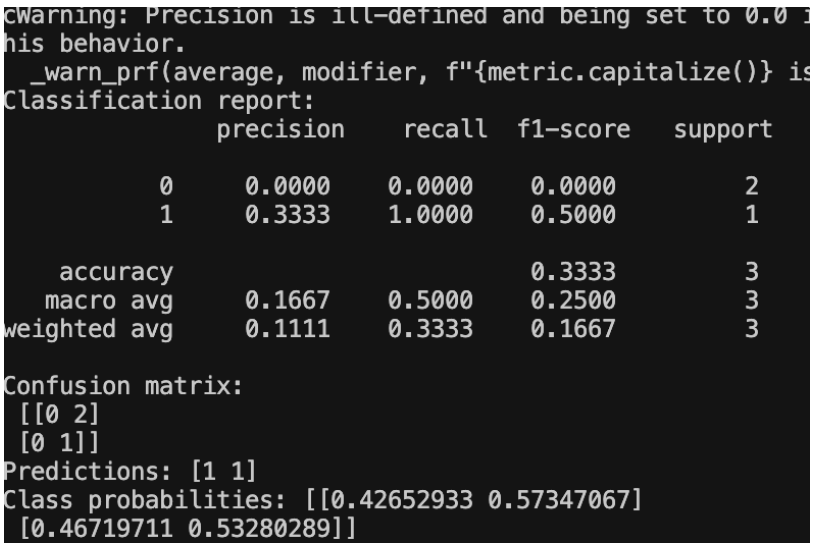
CODE 1: (1.py)

- **Concept used:** Preprocessing with spacy + regex, Text cleaning, tokenization, stopword removal, lemmatization.
- **OBSERVATION:**
- Input is normalized (lowercased, URLs/emails removed), only meaningful words kept; produces clean tokens ready for ML/NLP.
- 

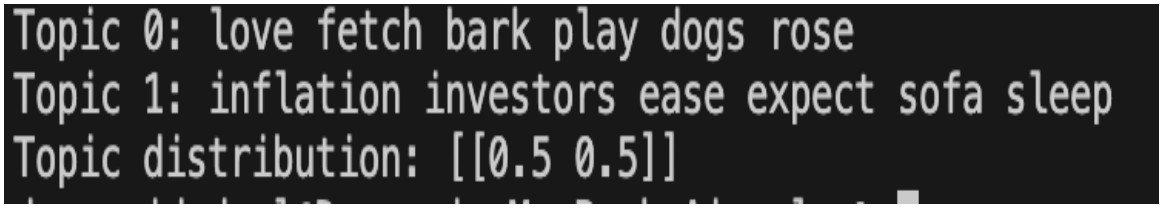
CODE 2 : (2.py)

- **Concept used:** Sentiment classification using TfidfVectorizer + supervised ML (Logistic Regression).
- **OBSERVATION:**
- Splits dataset, trains classifier, outputs the performance metrics; can predict sentiment for new samples with probabilities.
- 

CODE 3 : (3.py)

- **Concept used:** Model selection with GridSearchCV over TF-IDF + Logistic Regression pipeline.
- **OBSERVATION:**
- Tests multiple configs (ngrams, analyzer type, C values); outputs best params & score, improving generalization.
- 

CODE 4: (4.py)

- **Concept used:** Latent Dirichlet Allocation for unsupervised topic discovery.
- **OBSERVATION:**
- Groups docs into 2 topics (animals vs finance); infers topic distribution for unseen text.
- 

CODE 5 (5.py)

- **Concept used:** Named Entity Recognition, POS tagging, lemmatization, noun chunking.
- **OBSERVATION:**
- Extracts entities (Apple, Bengaluru, Tim Cook, etc.), shows grammatical roles of tokens, highlights structure of sentence.

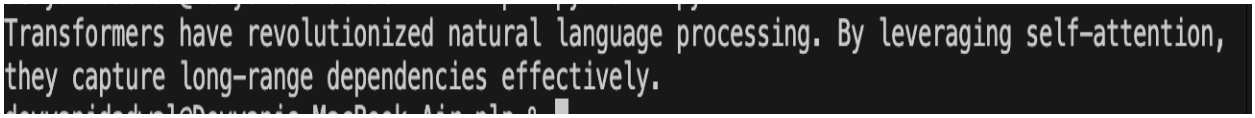
```
Named Entities (text, label):
Apple                          -> ORG
Bengaluru                      -> GPE
next quarter                   -> DATE
Tim Cook                       -> PERSON
Karnataka                      -> GPE
September 3, 2025              -> DATE

Part-of-Speech & Lemmas:
Apple            POS=PROPN   Lemma=Apple
is               POS=AUX     Lemma=be
opening          POS=VERB    Lemma=open
a                POS=DET     Lemma=a
new              POS=ADJ     Lemma=new
office           POS=NOUN    Lemma=office
in               POS=ADP     Lemma=in
Bengaluru        POS=PROPN   Lemma=Bengaluru
next             POS=ADJ     Lemma=next
quarter          POS=NOUN    Lemma=quarter
.                POS=PUNCT   Lemma=.
Tim              POS=PROPN   Lemma=Tim
Cook             POS=PROPN   Lemma=Cook
met              POS=VERB    Lemma=meet
Karnataka        POS=PROPN   Lemma=Karnataka
officials        POS=NOUN    Lemma=official
on               POS=ADP     Lemma=on
September        POS=PROPN   Lemma=September
3                POS=NUM     Lemma=3
,                POS=PUNCT   Lemma=,
2025             POS=NUM     Lemma=2025
to               POS=PART    Lemma=to
discuss          POS=VERB    Lemma=discuss
expansion        POS=NOUN    Lemma=expansion
.                POS=PUNCT   Lemma=.

Noun chunks (base NPs):
['Apple',
 'a new office',
 'Bengaluru',
 'Tim Cook',
 'Karnataka officials',
 'September',
 'expansion']
```

CODE 6 (6.py)

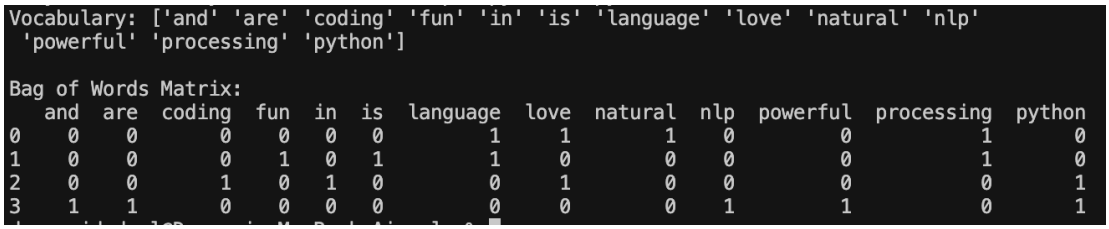
- **Concept used: Information retrieval using vector space similarity.**
- **OBSERVATION:**
- Query is matched with corpus; outputs top similar docs with similarity scores.

```
0.344  deep learning methods for image classification
0.000  transfer learning for NLP tasks
0.000  classical machine learning with SVM and logistic regression
```

CODE 7 : (7.py)

- **Concept used:** Scoring sentences by word frequency to select key sentences.
- **OBSERVATION**
- Produces concise summaries by keeping the most informative sentences; simple but effective baseline method.
- 

CODE 8: (8.py)

- **Concept used:** CountVectorizer (BoW representation).
- **OBSERVATION:**
- Creates word frequency matrix; shows vocabulary and document-term representation (raw counts).
- 

CODE 9: (9.py)

- **Concept used:** TF-IDF weighting of terms.
- **OBSERVATION:**
- Converts docs into importance-weighted vectors; downweights common words, highlights distinctive terms.

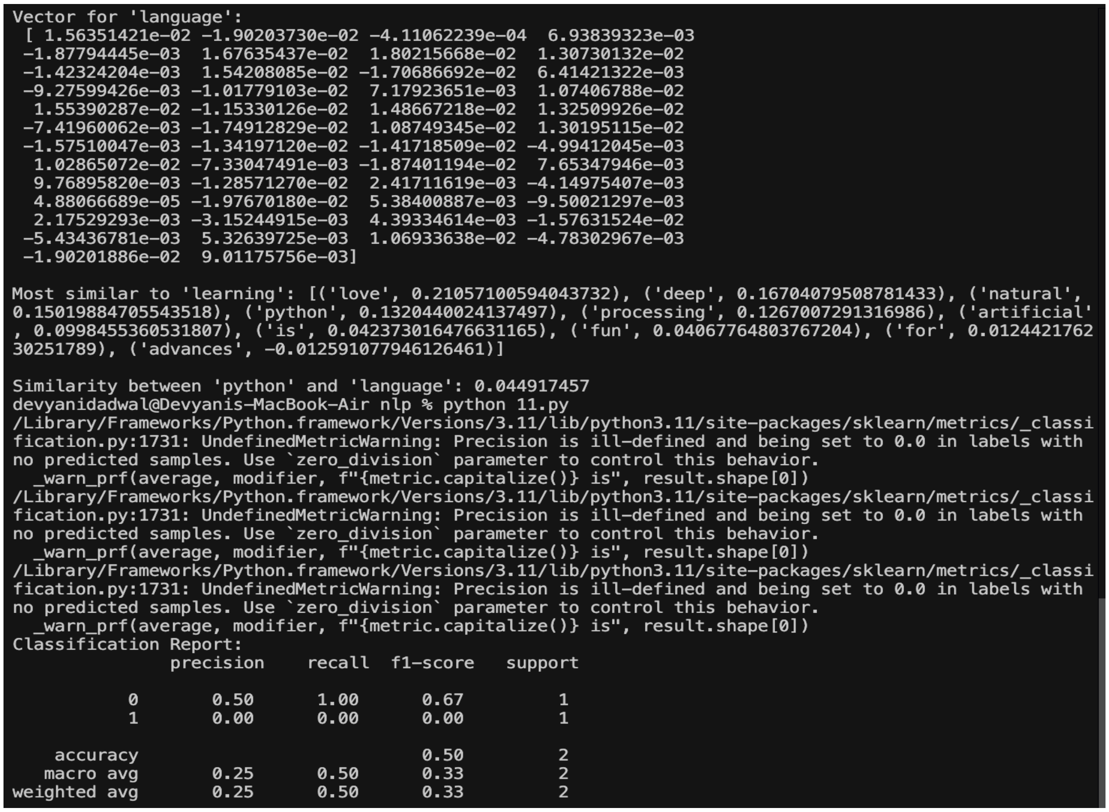
```
Vocabulary: ['advances' 'and' 'artificial' 'deep' 'fun' 'intelligence' 'is' 'learning'
 'machine']

TF-IDF Matrix:
   advances    and  artificial   deep    fun  intelligence     is  learning  machine
0     0.000  0.000       0.000  0.000  0.609          0.00  0.609     0.360    0.360
1     0.552  0.000       0.000  0.552  0.000          0.42  0.000     0.326    0.326
2     0.000  0.552       0.552  0.000  0.000          0.42  0.000     0.326    0.326
devyanidadwal@Devyanis-MacBook-Air nlp %
```

CODE 10: (10.py)

- **Concept used:** Distributional semantics with Word2Vec (Skip-gram).
- **OBSERVATION:**
- Generates dense embeddings; allows similarity checks, nearest neighbors, and word relationships.

```
Vector for 'language':
 [ 1.56351421e-02 -1.90203730e-02 -4.11062239e-04  6.93839323e-03
 -1.87794445e-03  1.67635437e-02  1.80215668e-02  1.30730132e-02
 -1.42324204e-03  1.54208085e-02 -1.70686692e-02  6.41421322e-03
 -9.27599426e-03 -1.01779103e-02  7.17923651e-03  1.07406788e-02
  1.55390287e-02 -1.15330126e-02  1.48667218e-02  1.32509926e-02
 -7.41960062e-03 -1.74912829e-02  1.08749345e-02  1.30195115e-02
 -1.57510047e-03 -1.34197120e-02 -1.41718509e-02 -4.99412045e-03
  1.02865072e-02 -7.33047491e-03 -1.87401194e-02  7.65347946e-03
  9.76895820e-03 -1.28571270e-02  2.41711619e-03 -4.14975407e-03
  4.88066689e-05 -1.97670180e-02  5.38400887e-03 -9.50021297e-03
  2.17529293e-03 -3.15244915e-03  4.39334614e-03 -1.57631524e-02
 -5.43436781e-03  5.32639725e-03  1.06933638e-02 -4.78302967e-03
 -1.90201886e-02  9.01175756e-03]

Most similar to 'learning': [('love', 0.21057100594043732), ('deep', 0.16704079508781433), ('natural',
0.15019884705543518), ('python', 0.1320440024137497), ('processing', 0.1267007291316986), ('artificial'
, 0.0998455360531807), ('is', 0.042373016476631165), ('fun', 0.04067764803767204), ('for', 0.0124421762
30251789), ('advances', -0.012591077946126461)]

Similarity between 'python' and 'language': 0.044917457
```

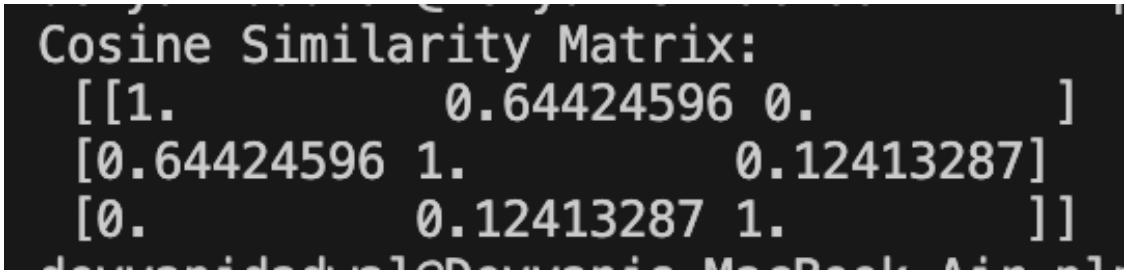
CODE 11: (11.py)

- **Concept used:** MultinomialNB + Bag-of-Words for sentiment.
- **OBSERVATION:**
- Trains probabilistic model; outputs classification report; works well for small text datasets.
- 

```
Vector for 'language':
 [ 1.56351421e-02 -1.90203730e-02 -4.11062239e-04  6.93839323e-03
 -1.87794445e-03  1.67635437e-02  1.80215668e-02  1.30730132e-02
 -1.42324204e-03  1.54208085e-02 -1.70686692e-02  6.41421322e-03
 -9.27599426e-03 -1.01779103e-02  7.17923651e-03  1.07406788e-02
  1.55390287e-02 -1.15330126e-02  1.48667218e-02  1.32509926e-02
 -7.41960062e-03 -1.74912829e-02  1.08749345e-02  1.30195115e-02
 -1.57510047e-03 -1.34197120e-02 -1.41718509e-02 -4.99412045e-03
  1.02865072e-02 -7.33047491e-03 -1.87401194e-02  7.65347946e-03
  9.76895820e-03 -1.28571270e-02  2.41711619e-03 -4.14975407e-03
  4.88066689e-05 -1.97670180e-02  5.38400887e-03 -9.50021297e-03
  2.17529293e-03 -3.15244915e-03  4.39334614e-03 -1.57631524e-02
 -5.43436781e-03  5.32639725e-03  1.06933638e-02 -4.78302967e-03
 -1.90201886e-02  9.01175756e-03]

Most similar to 'learning': [('love', 0.21057100594043732), ('deep', 0.16704079508781433), ('natural',
0.15019884705543518), ('python', 0.1320440024137497), ('processing', 0.1267007291316986), ('artificial'
, 0.0998455360531807), ('is', 0.042373016476631165), ('fun', 0.04067764803767204), ('for', 0.0124421762
30251789), ('advances', -0.012591077946126461)]

Similarity between 'python' and 'language': 0.044917457
devyanidadwal@Devyanis-MacBook-Air nlp % python 11.py
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/sklearn/metrics/_classi
fication.py:1731: 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", result.shape[0])
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/sklearn/metrics/_classi
fication.py:1731: 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", result.shape[0])
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/sklearn/metrics/_classi
fication.py:1731: 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", result.shape[0])
Classification Report:
              precision    recall  f1-score   support

           0       0.50      1.00      0.67         1
           1       0.00      0.00      0.00         1

    accuracy                           0.50         2
   macro avg       0.25      0.50      0.33         2
weighted avg       0.25      0.50      0.33         2
```

CODE 12: (12.py)

- **Concept used:** Document similarity measurement.
- **OBSERVATION:**
- Produces similarity matrix; similar docs (1 & 2) score high, unrelated docs (e.g., cooking) score low.

- 

CODE 13: (13.py)

- **Concept used:** Latent Dirichlet Allocation with gensim.
- **OBSERVATION:**
- Identifies 2 topics (tech/AI vs cooking); assigns words with probabilities; interpretable topic-word distribution.
-