Abhishek Milind Patwardhan
ID : 81127139

**DATA SCIENCE – II Homework – 3**

Q1)

Name : Abhishek. Milind. Patwardhan
ID : 81127139 (1)

Data Science - II - HomeWork - 3

1] Text Retrieval

- Today the Dawgs Won!
- Dawgs have won today
- Dawgs the champion!
- The Dawgs news today.

1) Preprocessing

→ 1) Today Dawgs Won
2) Dawgs ~~have~~ Won today
3) Dawgs champion
4) Dawgs news Today

2) Term- Document incidence Matrix

→

| Document → / Term ↓ | $D_1$ | $D_2$ | $D_3$ | $D_4$ |
|---|---|---|---|---|
| Today | 1 | 1 | 0 | 1 |
| Dawgs | 1 | 1 | 1 | 1 |
| Won | 1 | 1 | 0 | 0 |
| champion | 0 | 0 | 1 | 0 |
| news | 0 | 0 | 0 | 1 |

3) Implement - TF-IDF

→ Term Frequency (TF) = $\frac{\text{No. of times term 't' appears in doc 'd'}}{\text{Total No. of items in doc 'd'}}$

Document 1 : TF(Today) = 1/3
TF(Dawgs) = 1/3
TF(Won) = 1/3
TF(champion) = 0
TF(news) = 0

Abhishek Milind Patwardhan
ID : 81127139



Document 2 : TF(Today) = 1/3
TF (Dawgs) : 1/3
TF (won) = 1/3
TF (News) = 0
TF (champion) : 0

Document 3 : TF (Today) : 0
TF (Dawgs) : 1/2
TF (won) : 0
TF (News) = 0
TF (champion) : 1/2

Document 4 : TF (Today) = 1/3
TF (Dawgs) = 1/3
TF (won) : 0
TF (News) = 1/3
TF (champion = 0

Inverse Document Frequency (IDF)

$$IDF(t) = \log_{10}\left(\frac{\text{Total No. of documents}}{\text{No of documents with term 't'}}\right)$$

IDF (Today) = log (4/3) = log (4/3)
IDF ( Dawgs) = log (4/4) = 0
IDF (won) = log (4/2) = log (2)
IDF (News) = log (4/1) = log (4)
IDF (champion) = log (4/1) = log (4)

Abhishek Milind Patwardhan
ID : 81127139



∴ TF - IDF Matrix is as follows :

| Document → / Term ↓ | $D_1$ | $D_2$ | $D_3$ | $D_4$ |
|---|---|---|---|---|
| Today | $1/3 . \log(4/3)$ | $1/3 . \log(4/3)$ | 0 | $1/3 . \log(4/3)$ |
| Dawgs | 1/. 0 | 0 | 0 | 0 |
| won | $1/3 . \log(2)$ | $1/3 . \log(2)$ | 0 | 0 |
| champion | 0 | 0 | $1/2 . \log(4)$ | 0 |
| News | 0 | 0 | 0 | $1/3 . \log(4)$ |

4) Inverted Index for given query "Dawgs won".

→

1) Token Sequence

| Term | Document ID |
|---|---|
| Today | 1 |
| Dawgs | 1 |
| Won | 1 |
| Dawgs | 2 |
| Won | 2 |
| Today | 2 |
| Dawgs | 3 |
| champion | 3 |
| Dawgs | 4 |
| News | 4 |
| Today | 4 |

Abhishek Milind Patwardhan
ID : 81127139



2) Sorting by terms :

| Term | Document ID |
|---|---|
| champion | 3 |
| Dawgs | 1 |
| Dawgs | 2 |
| Dawgs | 3 |
| Dawgs | 4 |
| news | 4 |
| Today | 1 |
| Today | 2 |
| Today | 4 |
| won | 1 |
| won | 2 |

3) Dictionary & Posting :

| Term | Frequency | → | Posting Lists |
|---|---|---|---|
| Dawgs | 4 | → | 1 → 2 → 3 → 4 |
| champion | 1 | → | 3 |
| news | 1 | → | 4 |
| today | 3 | → | 1 → 2 → 4 |
| won | 2 | → | 1 → 2 |

4) Query : "Dawgs won"

→ We will locate "Dawgs" & "won" seperately & retrieve their respective postings.

| Dawgs \| 4 | → | 1 → 2 → 3 → 4 |
|---|---|---|
| won \| 2 | → | 1 → 2 |

Abhishek Milind Patwardhan
ID : 81127139



Finding the intersection of Document ID's for all terms in query:

$[1,2,3,4] \cap [1,2] = [1,2]$.

Hence Document 1 & 2 are retrieved documents for given query "Dawgs won".

5) Explain advantages of inverted index over term-document incidence matrices for organizing texts?

→ 1) Efficient Retrieval:
- Inverted index allows fast retrieval of documents containing specific terms because index directly maps terms to the documents they appear in.

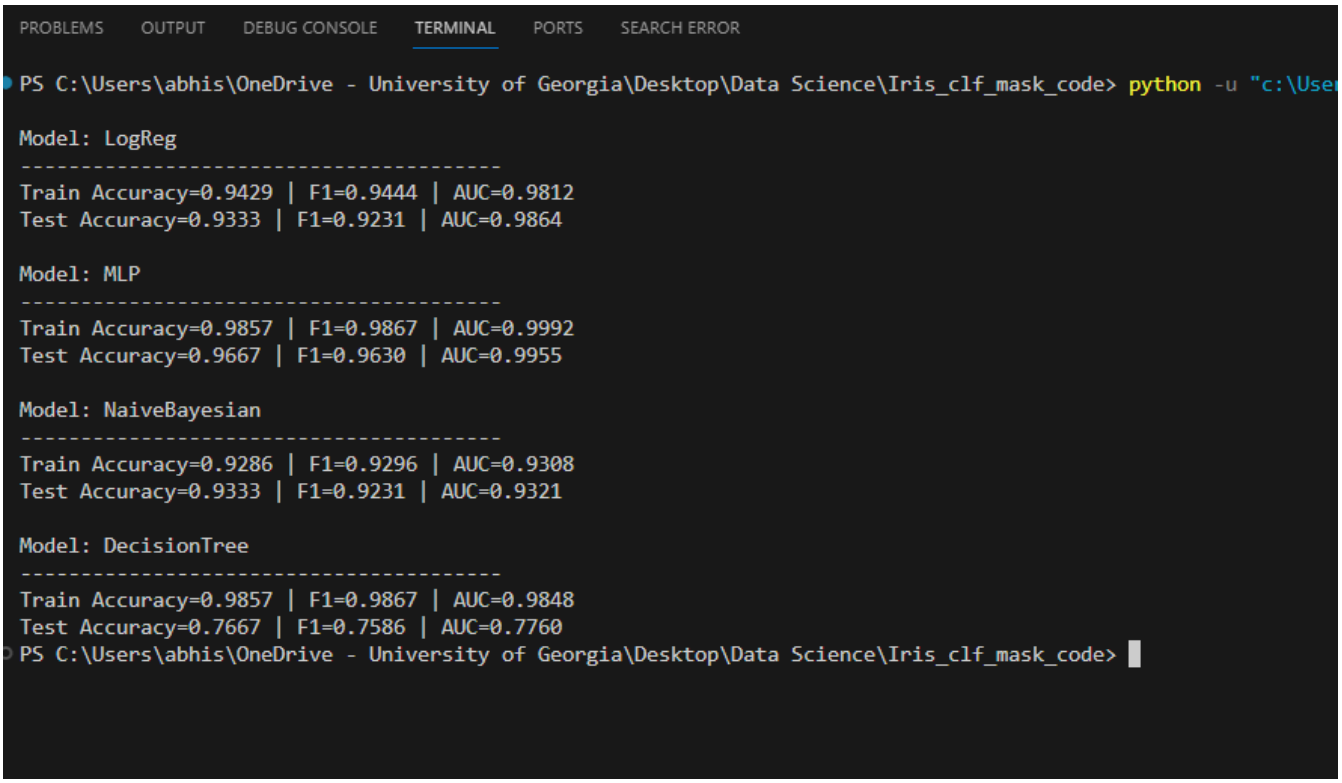
2) Scalability:
- Inverted indexes are generally more scalable than term-document incidence matrices due to their distributed and parallel nature.

3) Dynamic Updation:
- Inverted index can be easily updated when new documents are added. Whereas term-document incidence matrices may require recalculations or resizing operations.

4) Space Efficiency:
- Inverted index stores only the unique terms and their associated document references. Hence it is more space efficient than term-document incidence matrices.

Abhishek Milind Patwardhan
ID : 81127139

Q 2)




## Conclusions :

The results indicate that all models perform well on the training set, with Logistic Regression, MLP, and Decision Tree achieving high accuracy, F1 score, and AUC. However, on the test set, MLP stands out as the best-performing model with the highest accuracy, F1 score, and AUC, indicating its superior generalization ability. Logistic Regression also performs reasonably well on the test set, showing robustness in its predictions. Naive Bayesian and Decision Tree models exhibit moderate performance, with Decision Tree showing signs of overfitting as evidenced by the significant gap between training and test accuracies. In conclusion, while MLP emerges as the top-performing model, Logistic Regression demonstrates competitive performance with good generalization capabilities, making it a reliable choice when interpretability and simplicity are valued.