TEAM MEMBERS:

NAME: Balraj Hanmanthugari	**UID: U01079536**	**EMAIL: hanmanthugari.2@wright.edu**
NAME: Deepika Kasula	**UID: U01067608**	**EMAIL: kasula.16@wright.edu**
NAME: Nitish Kota	**UID: U01074656**	**EMAIL: kota.58@wright.edu**

Code Execution Flow :

Load and preprocess the Cranfield Dataset documents and queries.

Build the inverted index using the preprocessed documents.Process each query through the search engine, retrieve documents, and evaluate the search engine's performance using predefined metrics.

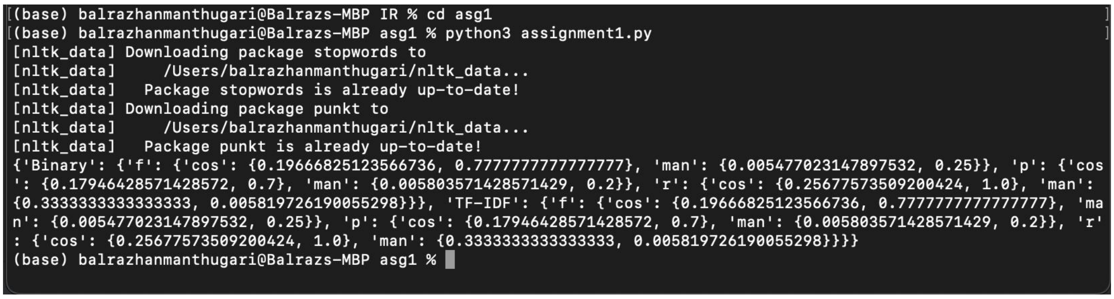
Evaluation :Uses the Cranfield evaluation methodology to assess the search engine's effectiveness based on precision, recall, and F-score.

Preprocess :

- Performs preprocessing on the given text, including tokenization, stopword removal, and stemming.
- Constructs an inverted index from the preprocessed documents docs (list of lists) - Preprocessed documents split into tokens and a dictionary representing the inverted index is generated.
- Calculates precision, recall, and F-score for the given sets of retrieved and relevant documents the function used in the application is calculate_metrics(retrieved_documents, relevant_documents) and retrieved_documents (list) , relevant_documents (list).
- Processes the given query against the inverted index to retrieve relevant documents(List of document IDs relevant to the query).
- functions used are : query (str) - The user's search query, index (dict) - The inverted index.

Top 10 relevant document query results from the Cranfield collection and the evaluation metrics for the Test Set.

Terminal Output:

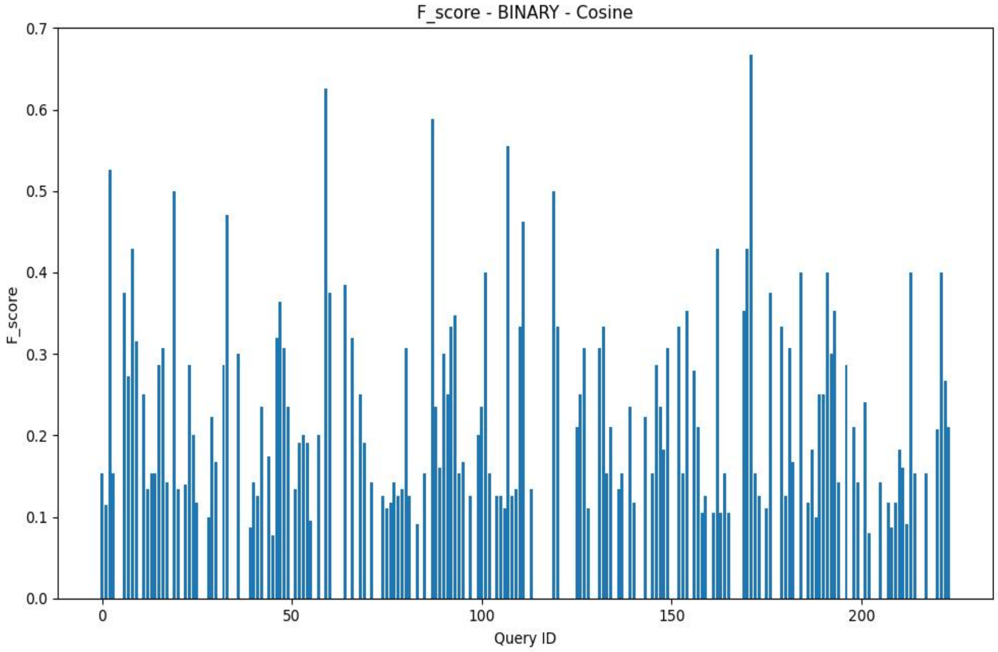


```
[(base) balrazhanmanthugari@Balrazs-MBP IR % cd asg1
[(base) balrazhanmanthugari@Balrazs-MBP asg1 % python3 assignment1.py
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/balrazhanmanthugari/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/balrazhanmanthugari/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
{'Binary': {'f': {'cos': {0.19666825123566736, 0.7777777777777777}, 'man': {0.005477023147897532, 0.25}}, 'p': {'cos
': {0.17946428571428572, 0.7}, 'man': {0.005803571428571429, 0.2}}, 'r': {'cos': {0.25677573509200424, 1.0}, 'man':
{0.3333333333333333, 0.005819726190055298}}}, 'TF-IDF': {'f': {'cos': {0.19666825123566736, 0.7777777777777777}, 'ma
n': {0.005477023147897532, 0.25}}, 'p': {'cos': {0.17946428571428572, 0.7}, 'man': {0.005803571428571429, 0.2}}, 'r'
: {'cos': {0.25677573509200424, 1.0}, 'man': {0.3333333333333333, 0.005819726190055298}}}}
(base) balrazhanmanthugari@Balrazs-MBP asg1 % 
```

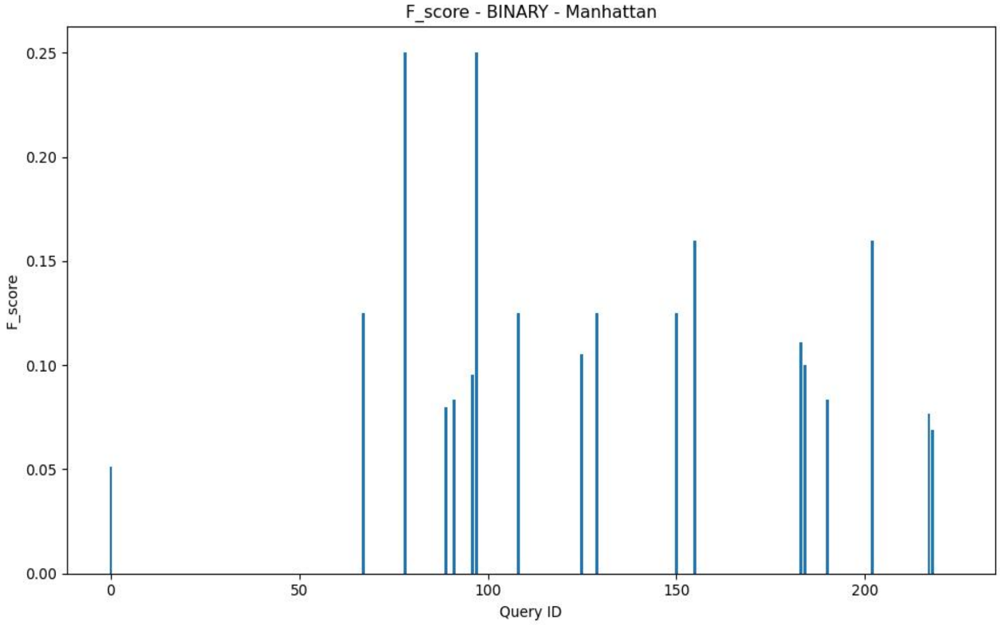
Screenshots : Graphs

Binary:

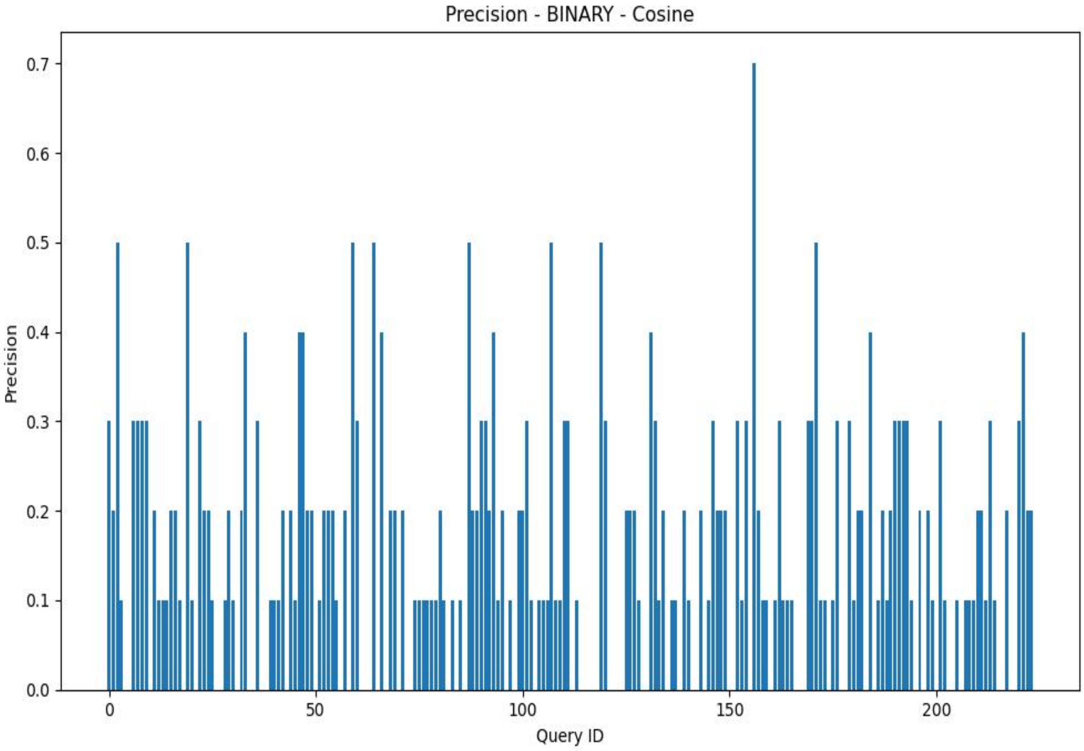
F-Score : Cosine



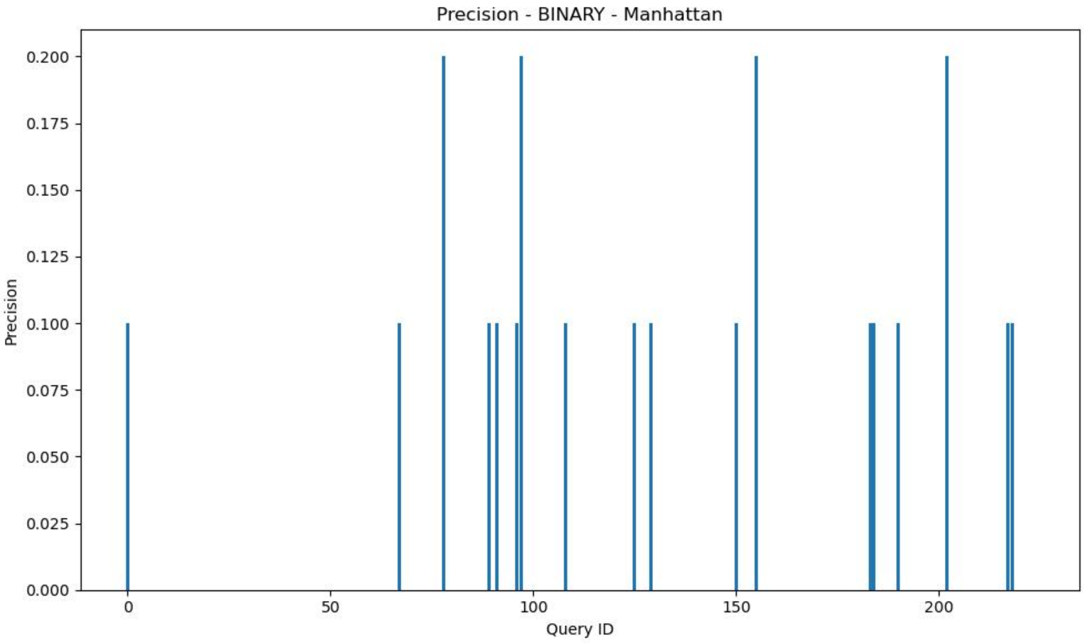
F-Score : Manhattan



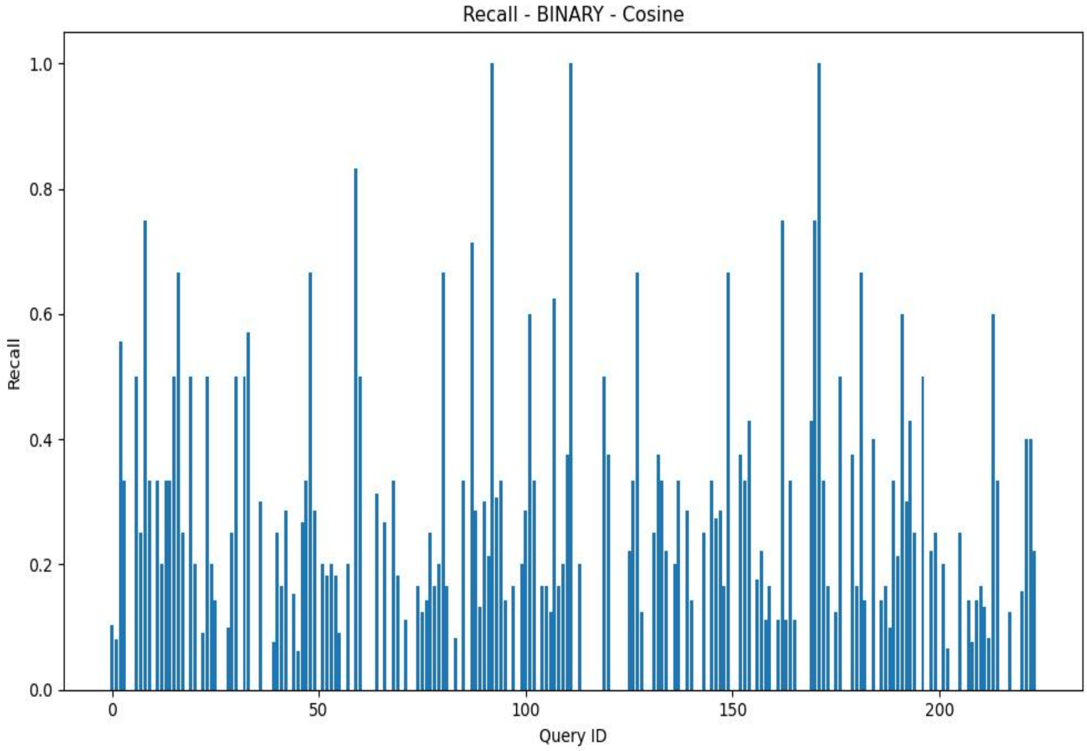
Precision : Cosine



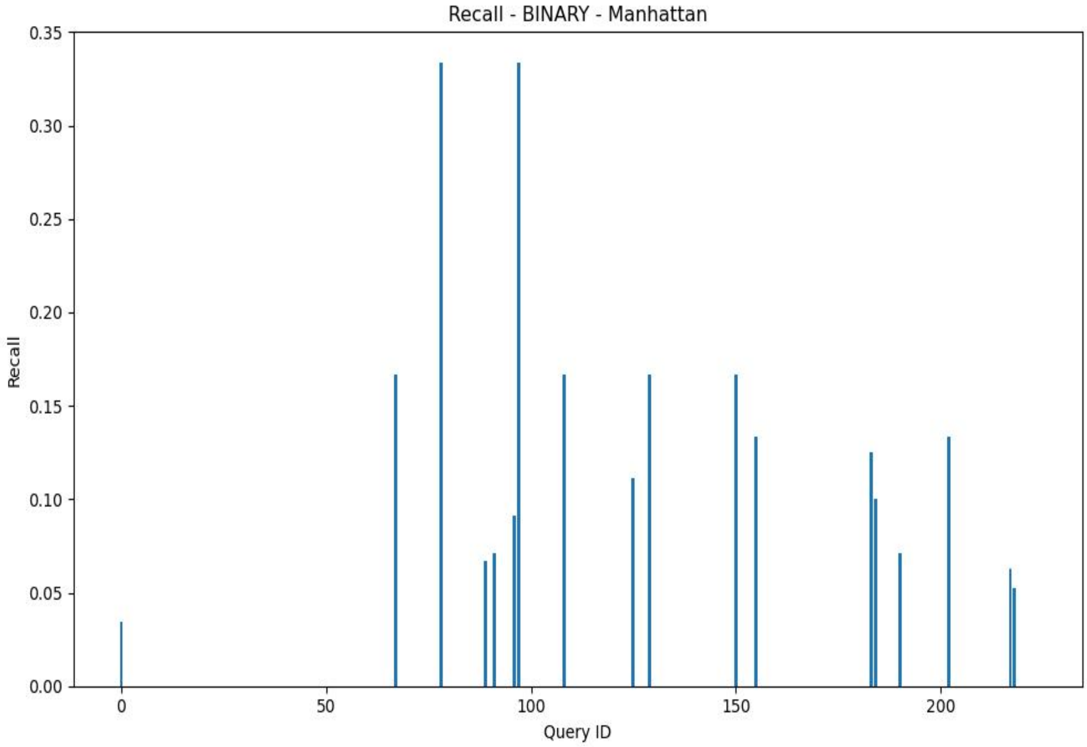
Precision : Manhattan



Recall : Cosine

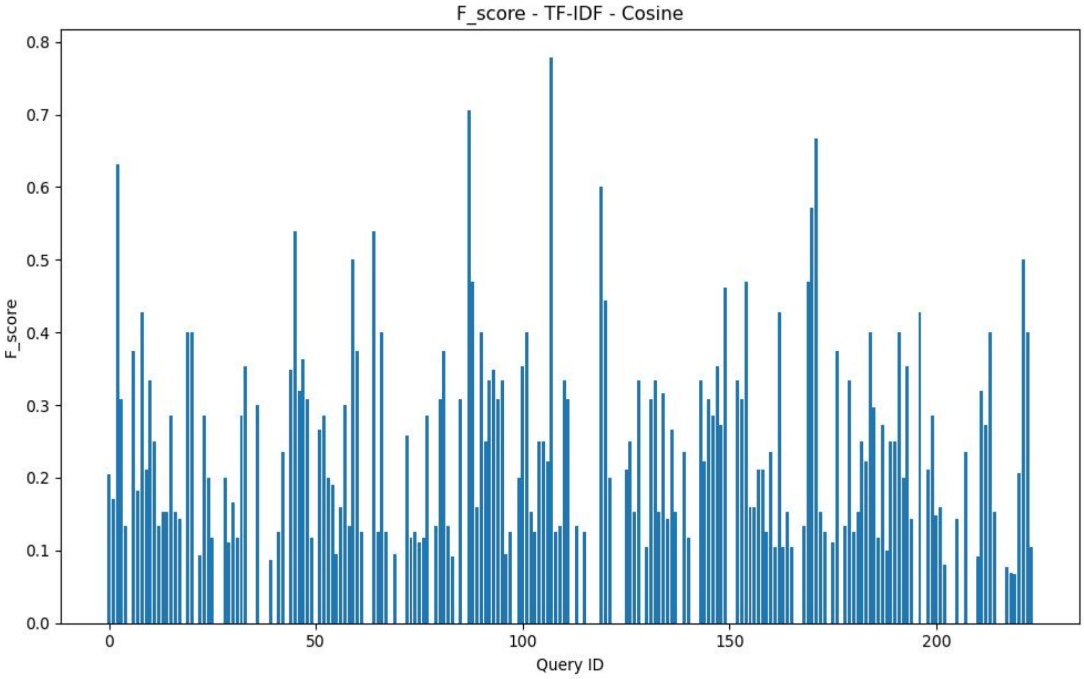


Recall : Manhattan

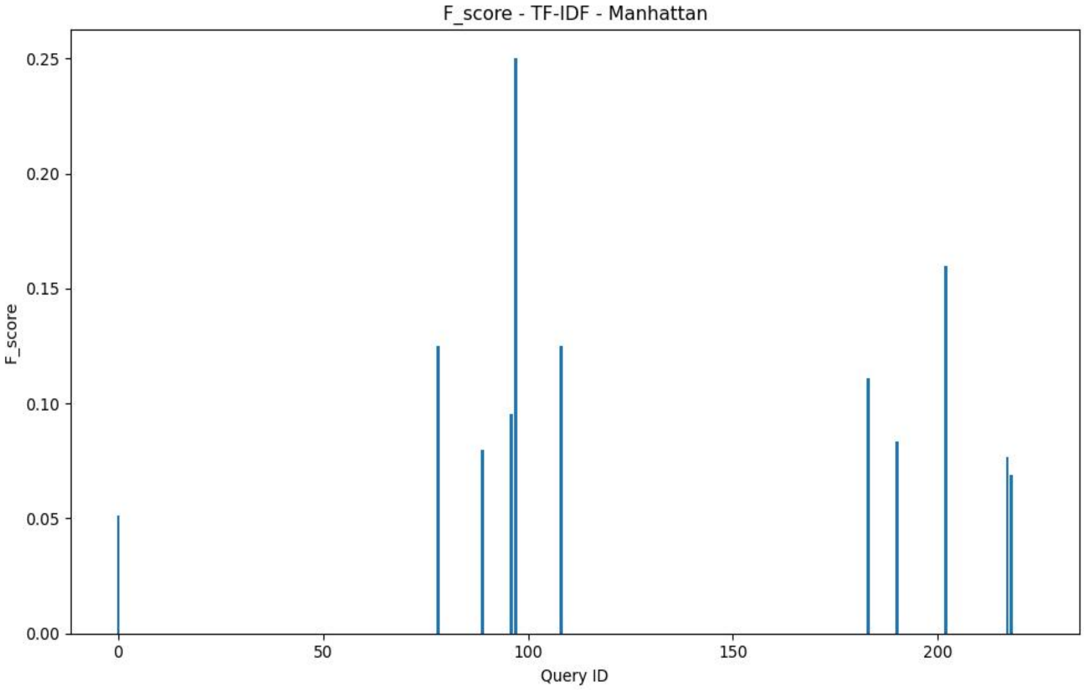


TF_IDF

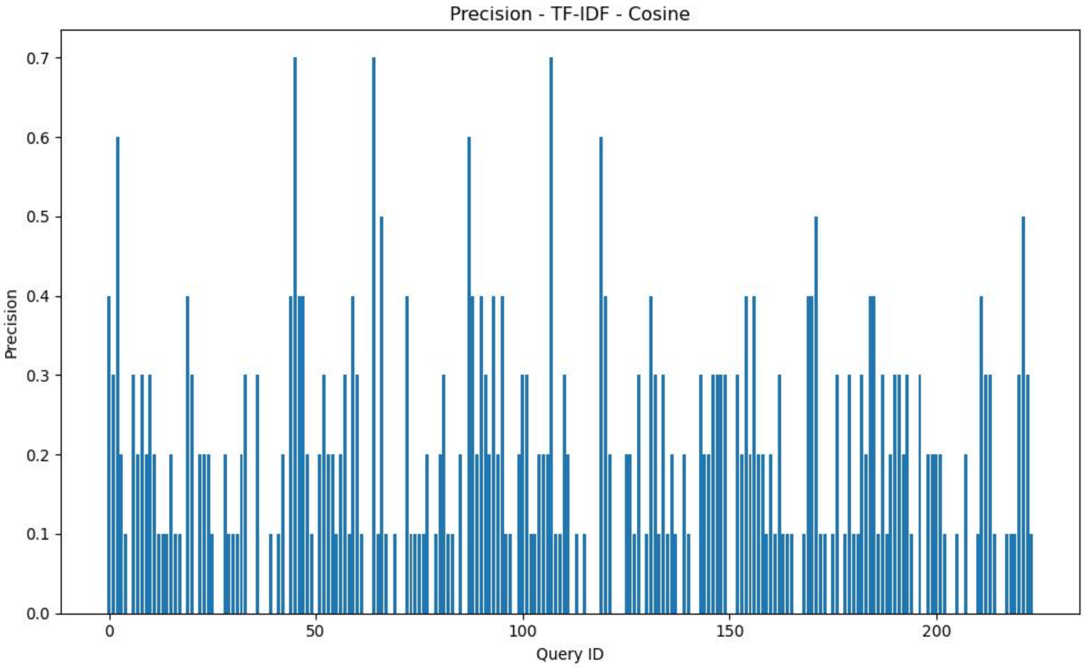
F-Score : Cosine



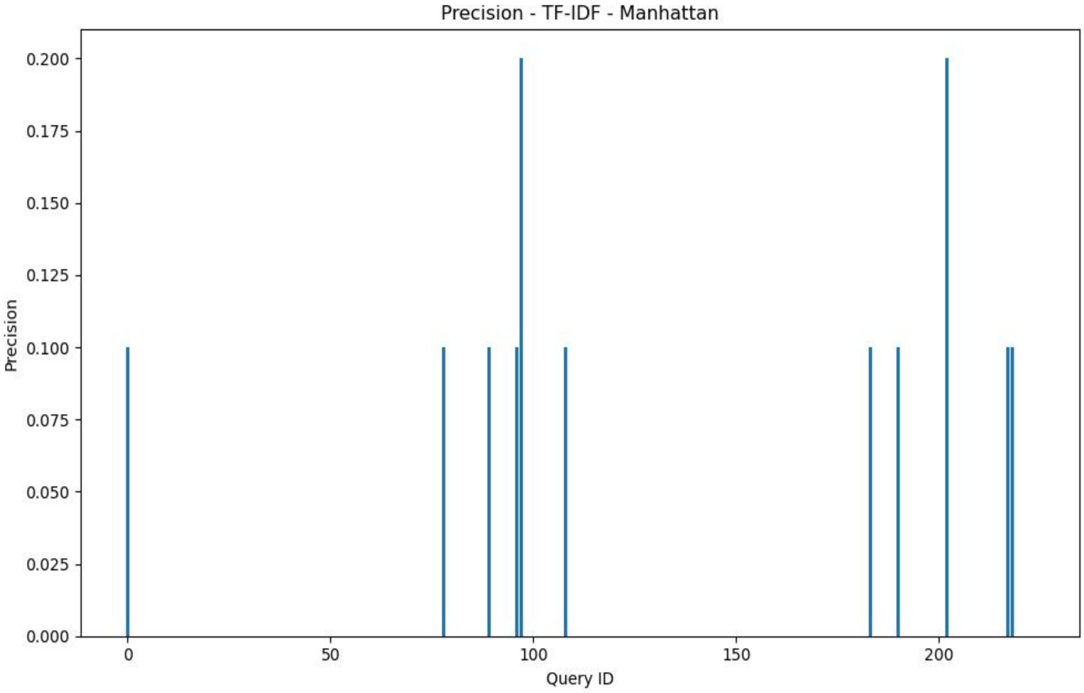
F-Score : Manhattan



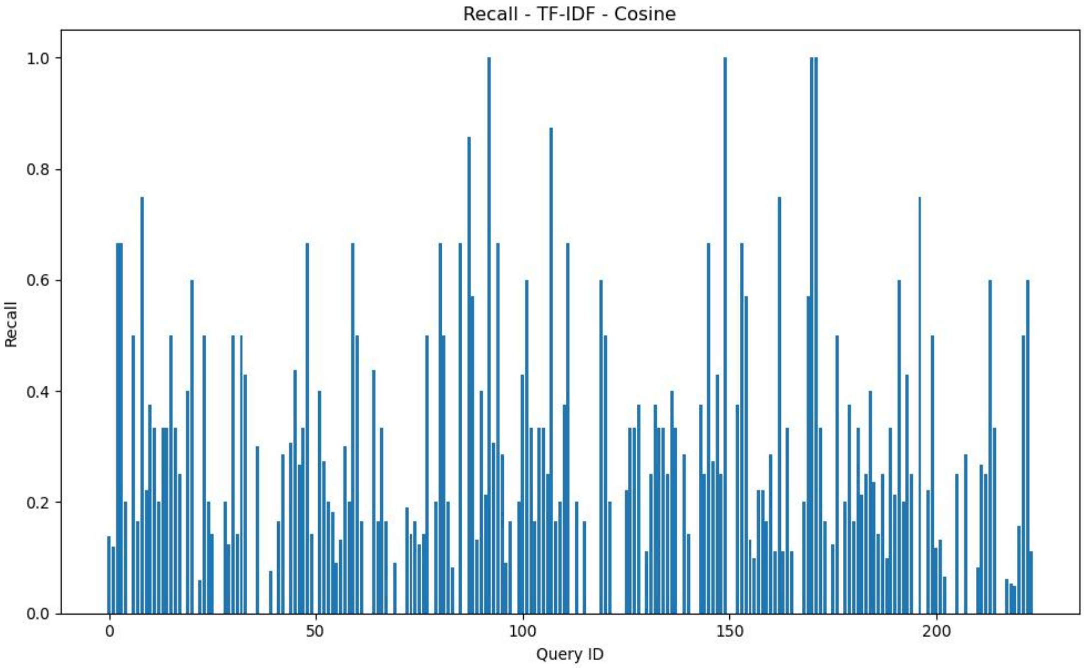
Precision: Cosine



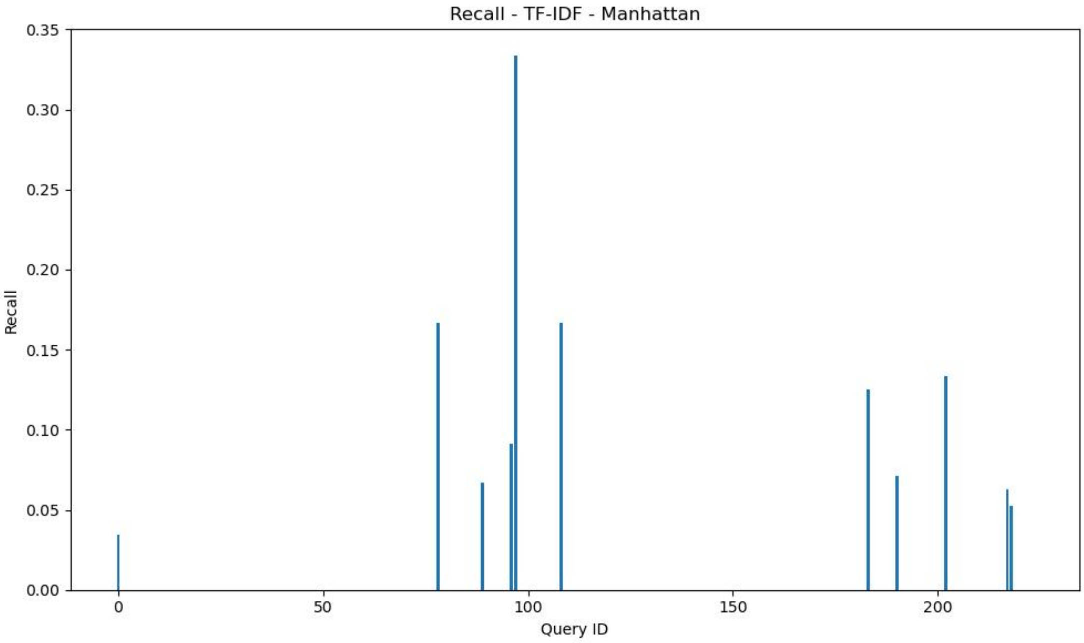
Precision: Manhattan



Recall : Cosine



Recall : Manhattan


Recall - TF-IDF - Manhattan
Recall
Query ID