# Graph-Based Classification of Web Documents


**Project Supervisor**
Mr. Waqas Ali

Muhammad Shahzaib Ijaz 2021-CS-75

Department of Computer Science
**University of Engineering and Technology**
**Lahore Pakistan**

# Acknowledgement

We cannot express enough how thankful I am to Allah Almighty for granting us the determination and capacity to complete this project within the given deadline. The project was undoubtedly a challenging task, but with the support, guidance, and attentive mentoring of Mr. Waqas Ali, I was able to surpass my limits and produce the best project I could. His valuable insights on work management and scheduling have not only helped me in this project but will be beneficial for me in my future endeavors as well.

Moreover, my friends have been my pillars of strength throughout this project. They were always there to lend a hand whenever I needed them, and their unwavering support kept me motivated to keep pushing forward. I am grateful to have such amazing friends who have been our constant source of encouragement and positivity.

# Contents


Acknowledgement . . . . . i

**1 Introduction** **1**
1.1 Background . . . . . 1
1.2 Objective . . . . . 1

**2 Methadology** **2**
2.1 Data Collection . . . . . 2
2.1.1 Selection of Data Sources: . . . . . 2
2.1.2 Web Scraping Implementation: . . . . . 2
2.1.3 Scraping Considerations: . . . . . 2
2.1.4 Data Extraction and Storage: . . . . . 2
2.2 Data Processing . . . . . 2
2.2.1 Text Lowercasing: . . . . . 2
2.2.2 Tokenization: . . . . . 2
2.2.3 Stopword Removal: . . . . . 3
2.2.4 Stemming: . . . . . 3
2.2.5 Preprocessing Function: . . . . . 3
2.2.6 Batch Processing: . . . . . 3
2.3 Graph Construction . . . . . 3
2.3.1 Frequent Subgraph Mining: . . . . . 3
2.3.2 K-Nearest Neighbors (KNN): . . . . . 4
2.3.3 Confusion Matrix: . . . . . 4

**3 Project Management** **7**

**4 Chalenges faced** **8**
4.0.1 Computational Resource Limitations: . . . . . 8
4.0.2 Complexity of Feature Extraction: . . . . . 8
4.0.3 gSpan Input File Structure Enhancement: . . . . . 8

**5 Future Work** **9**
5.0.1 Graph Mining Technique Enhancement: . . . . . 9
5.0.2 Maximal Common Subgraph (MCS) Improvement: . . . . . 9

# List of Figures

2.1 Example Graph . . . . . 4
2.2 Knn accuracy . . . . . 5
2.3 Confusion Matrix . . . . . 6

3.1 Project Management . . . . . 7

# Chapter 1

# Introduction

The project "Classification of Documents Using Graph-Based Features and KNN" navigates the convergence of graph theory and machine learning to tackle the intricate task of document classification. By encapsulating documents as directed graphs and harnessing the power of the K-Nearest Neighbors (KNN) algorithm, this initiative endeavors to elevate classification accuracy while enriching our understanding of document categorization. Central to this exploration is the utilization of maximal common subgraphs (MCS) and the gSpan algorithm, which enable the extraction of recurring patterns within document graphs. Through immersive engagement with graph-based features and the refinement of feature extraction techniques, participants gain invaluable insights into data representation, algorithmic implementation, and analytical reasoning. Rooted in foundational research and propelled by cutting-edge methodologies, this project not only serves as a platform for innovation in document classification but also paves the path towards future breakthroughs in the realm of machine learning and natural language processing.

## 1.1 Background

In today's vast internet-connected world, an increasing amount of documents are being created. The current moment is more elevated than ever. Every day, numerous papers are included in this extensive assortment. Automated categorization is a crucial area of study that focuses on cutting down on time and expenses. The vector model employed was inefficient and less precise as it overlooked important text structure details, and this database lacks the capability to manage it. Hence, it was important to utilize a different data structure that enables efficient storage and retrieval of extra information (such as text organization for better categorization) while also offering methods and ideas to minimize time and expenses. Graphs are crucial in that situation.

## 1.2 Objective

The primary goal of this project is to create a strong system for categorizing documents that makes use of graph-based characteristics and the KNN method. Some of the specific goals include:

- Collecting a large dataset of documents from the predefined categories
- Representing each document as a directed graph.
- Identifying common subgraphs within the training set of documents.
- Implementing the KNN algorithm based on graph similarity measures.
- Classifying test documents based on their similarity to training documents in the feature space defined by common subgraphs.

# Chapter 2

# Methadology

## 2.1 Data Collection

### 2.1.1 Selection of Data Sources:

1. Identified relevant websites containing documents related to each of the three assigned topics.
2. Ensured that the selected websites provided diverse and comprehensive coverage of the topics of interest.

### 2.1.2 Web Scraping Implementation:

1. Utilized web scraping techniques to automate the process of extracting text data from web pages.
2. Employed tools such as Selenium or Beautiful Soup to programmatically navigate through web pages and extract desired content.

### 2.1.3 Scraping Considerations:

1. Implemented robust error handling mechanisms to handle various issues encountered during scraping, such as timeouts, missing elements, or changes in website layout.
2. Incorporated delays between requests to mimic human browsing behavior and avoid being blocked by website security measures.

### 2.1.4 Data Extraction and Storage:

1. Extracted text content from the web pages containing relevant documents using XPath or CSS selectors.
2. Stored the extracted text data in a structured format, such as plain text files, for further processing and analysis.

## 2.2 Data Processing

### 2.2.1 Text Lowercasing:

1. Converted the text to lowercase to ensure uniformity and consistency in text processing.

### 2.2.2 Tokenization:

1. Utilized the NLTK library to tokenize the text into individual words or tokens.

### 2.2.3 Stopword Removal:

1. Removed common stopwords such as "the," "is," and "and" from the tokenized text to reduce noise and improve model performance.

### 2.2.4 Stemming:

1. Employed the Porter Stemmer algorithm to reduce words to their base or root form, removing inflectional endings and reducing the dimensionality of the feature space.

### 2.2.5 Preprocessing Function:

1. Defined a custom preprocessing function to encapsulate the above steps and streamline the text preprocessing pipeline.

### 2.2.6 Batch Processing:

1. Applied the preprocessing function to each document in the dataset, iterating through all documents for each topic.

## 2.3 Graph Construction

**Following are the points for graph construction:**

- **Representation of Documents:** Modeled each document as a directed graph, with nodes representing unique terms (words) and edges denoting term relationships based on their sequence in the text.
- **Node Creation:** Created nodes in the graph corresponding to each unique term extracted from the preprocessed text data.
- **Edge Formation:** Established edges between nodes based on the sequential occurrence of terms in the text, representing the relationships between adjacent terms.
- **Graph Representation:** Utilized a graph data structure, such as NetworkX in Python, to construct and manipulate the document graphs efficiently.
- **Document-Graph Mapping:** Maintained a mapping between documents and their corresponding graphs to facilitate further analysis and model training.

### 2.3.1 Frequent Subgraph Mining:

**Following are the points for graph mining:**

- Utilized the gSpan algorithm, a popular method for frequent subgraph mining, to extract recurring patterns within document graphs.
- Transformed document graphs into a format compatible with gSpan by writing them into input files, typically in an edge list or adjacency list format.
- Defined parameters such as minimum support threshold, maximum pattern size, and output options to configure the behavior of the gSpan algorithm.
- Ran the gSpan algorithm on the prepared input files to identify frequent subgraphs that occur with a minimum support threshold across the dataset.
- Extracted the identified frequent subgraphs as features, representing common patterns within document graphs that contribute to classification.
- Analyzed the extracted frequent subgraphs to gain insights into the underlying structure of document categories and the discriminative features contributing to classification.

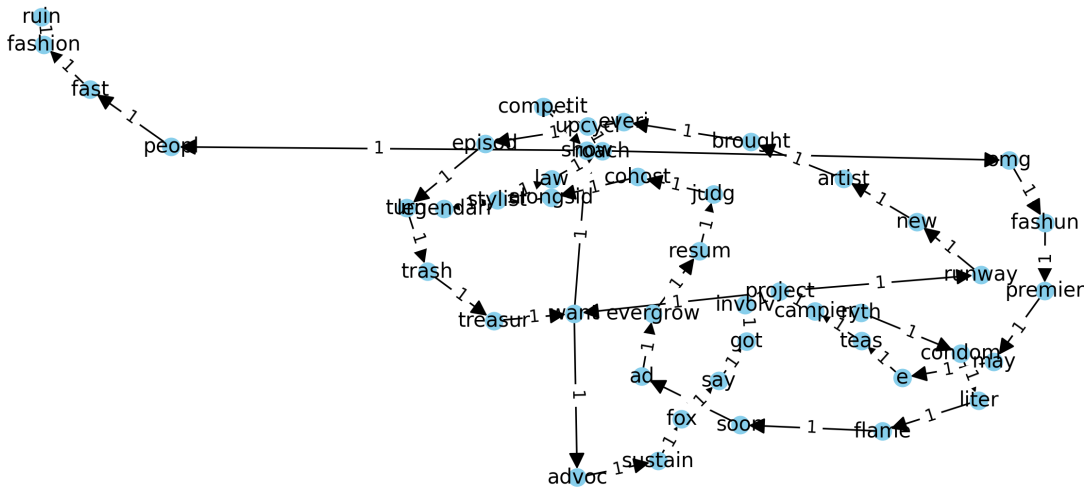


Figure 2.1: Example Graph

- Conducted parameter tuning and optimization to enhance the performance and efficiency of the frequent subgraph mining process.

### 2.3.2 K-Nearest Neighbors (KNN):

**Following are the points for knn:**

- **Algorithm Overview:** KNN is a simple yet effective machine learning algorithm used for classification tasks.It classifies data points based on the majority class among their k nearest neighbors in the feature space.
- **Feature Space Representation:** Represented documents using graph-based features extracted from frequent subgraphs identified through subgraph mining techniques.
- **Model Implementation:** Implemented the KNN algorithm using a distance measure based on maximal common subgraph (MCS) similarity between document graphs.
- **Hyperparameter Tuning:** Selected an appropriate value for the number of neighbors (k) through cross-validation to optimize classification performance.
- **Training Phase:** Trained the KNN classifier on the training set, consisting of labeled documents and their corresponding graph-based features.
- **Classification Process:** Classified test documents based on the majority class of their k-nearest neighbors in the feature space created by common subgraphs.
- **Evaluation Metrics:** Evaluated the performance of the KNN classifier using standard metrics such as accuracy, precision, recall, and F1-score.
- **Optimization and Iteration:**Iteratively refined the model by fine-tuning hyperparameters and optimizing the feature extraction process based on performance feedback and analysis results.

### 2.3.3 Confusion Matrix:

**A confusion matrix is a tabular representation that summarizes the performance of a classification model by presenting the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) classifications.**

- **True Positive (TP):** The number of instances correctly classified as positive by the model.

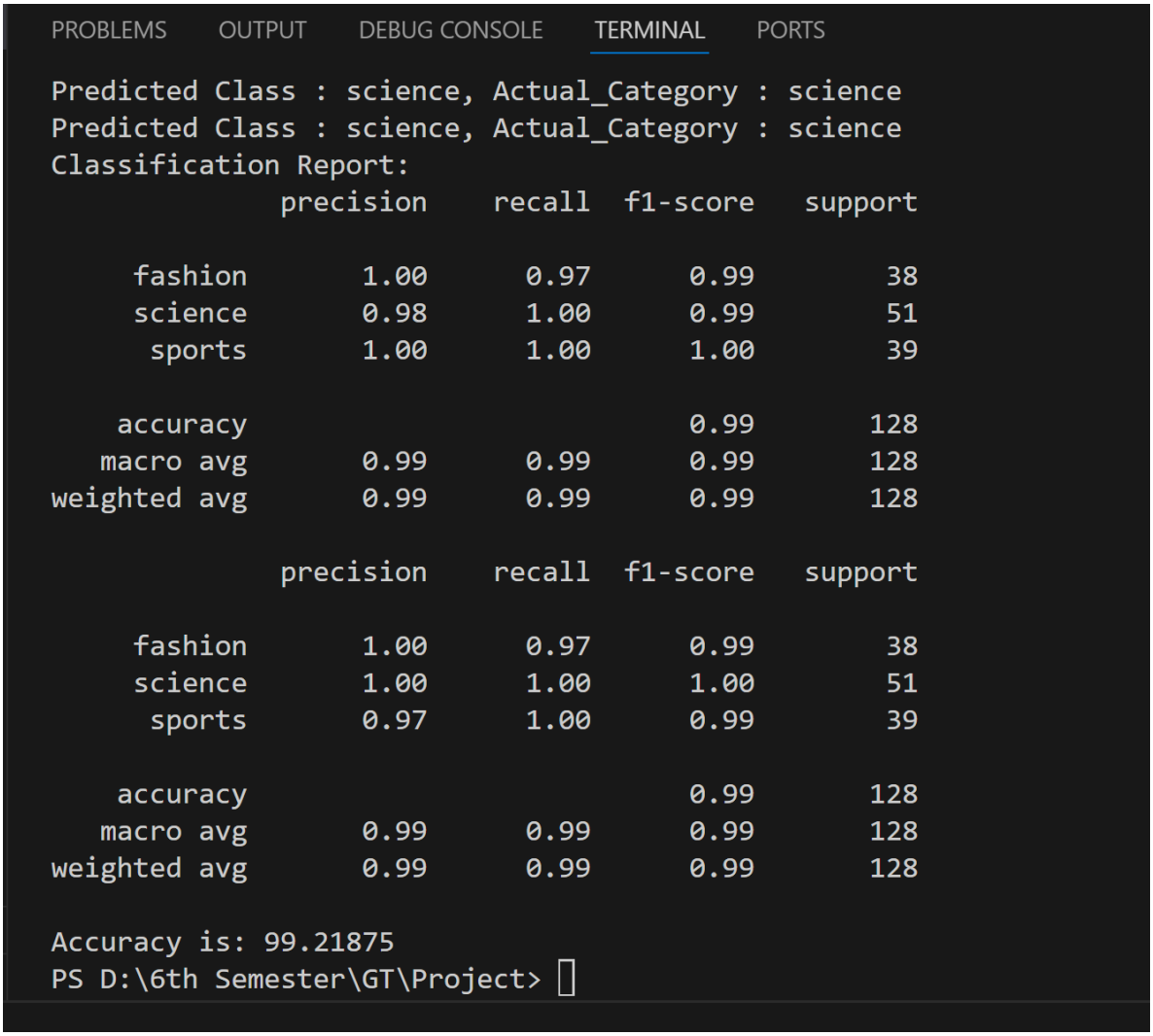


Figure 2.2: Knn accuracy

- **True Negative (TN):** The number of instances correctly classified as negative by the model.
- **False Positive (FP):** The number of instances incorrectly classified as positive by the model (i.e., instances that are actually negative but predicted as positive).
- **False Negative (FN):** The number of instances incorrectly classified as negative by the model (i.e., instances that are actually positive but predicted as negative).
- **Visual Representation:** The confusion matrix is often visualized as a square matrix, with actual class labels on one axis and predicted class labels on the other axis, making it easy to interpret and analyze the classification results.

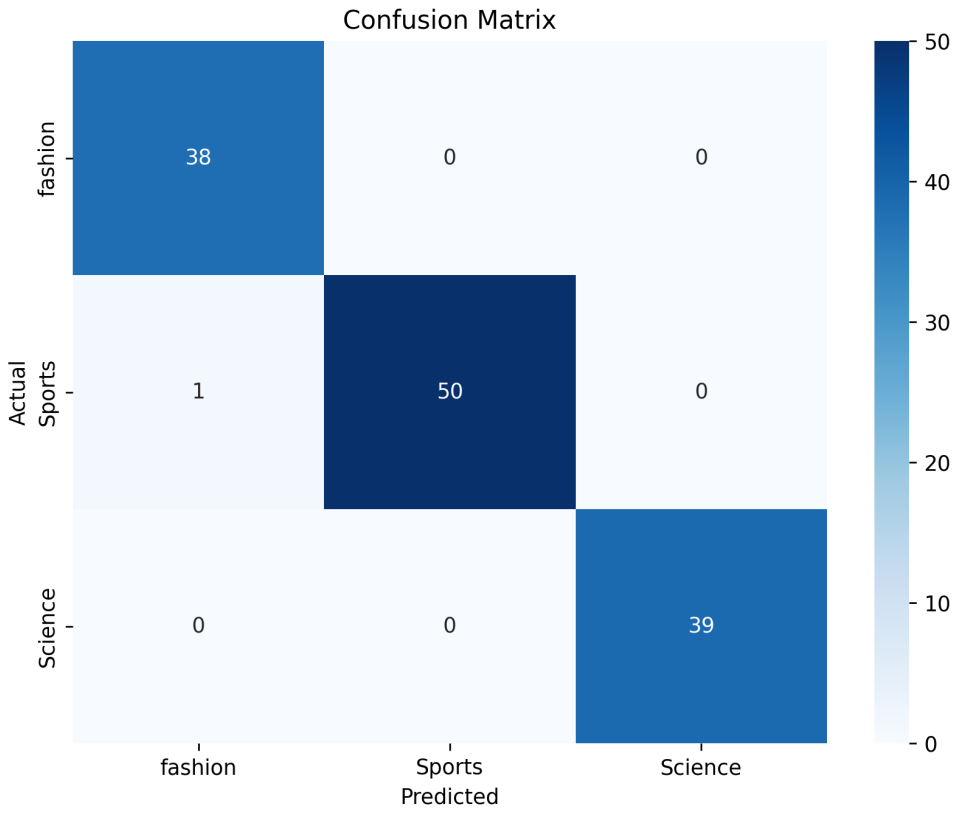


Figure 2.3: Confusion Matrix

# Chapter 3

# Project Management

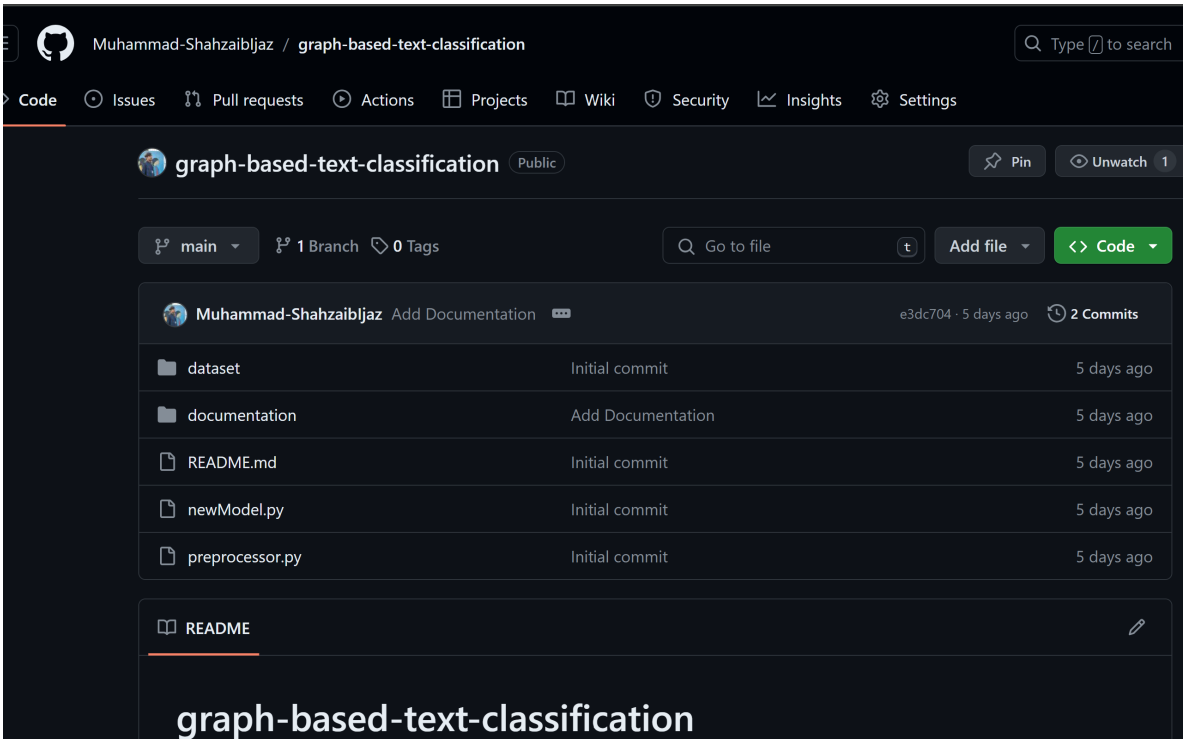


Figure 3.1: Project Management

# Chapter 4

# Chalenges faced

### 4.0.1 Computational Resource Limitations:

Struggled with limited computational resources, especially GPU and CPU constraints, hindering the efficient execution of graph-based algorithms and feature extraction processes.

### 4.0.2 Complexity of Feature Extraction:

Encountered challenges in developing a robust feature extraction function to accurately capture relevant graph-based features from document graphs while managing computational complexity and ensuring model interpretability..

### 4.0.3 gSpan Input File Structure Enhancement:

Address challenges associated with the input file structure required by the gSpan algorithm, exploring methods to streamline and simplify the preparation of input files to facilitate efficient subgraph mining and feature extraction processes.

# Chapter 5

# Future Work

### 5.0.1 Graph Mining Technique Enhancement:

Investigate methods to enhance the efficiency and effectiveness of graph mining techniques, particularly in the context of frequent subgraph mining, by exploring alternative algorithms, optimizing parameters, and addressing computational challenges.

### 5.0.2 Maximal Common Subgraph (MCS) Improvement:

Improve the robustness and performance of maximal common subgraph (MCS) identification methods by refining algorithms, optimizing parameters, and incorporating domain-specific knowledge to better capture common patterns and relationships within document graphs. Struggled with limited computational resources, especially GPU and CPU constraints, hindering the efficient execution of graph-based algorithms and feature extraction processes.