# IT 335 - Introduction to Natural Language Processing

Github Profile Analyzer

IBU | International Burch University

Prepared by:

**Faris Muhović**

**Ali Marat**

# Table of Contents

# 1. Introduction

## 1.1 About the Project

The **GitHub Repository Analyzer** is a Python-based script designed to analyze public repositories of a specified GitHub user. It fetches files containing code from the repositories, processes the content using natural language processing (NLP) techniques, and generates insights about programming language usage and topic distribution. This analysis employs text preprocessing, TF-IDF vectorization, and topic modeling via Latent Dirichlet Allocation (LDA).

## 1.2 Project Functionalities and Illustrations

The main features of the GitHub Repository Analyzer include:

- **File Retrieval:** Extracting files from the public repositories of a specified GitHub user using the GitHub API.
- **Text Preprocessing:** Cleaning up text by removing stopwords, punctuation, and numbers.
- **Keyword Extraction:** Generating vocabulary arrays using Count Vectorizer and TF-IDF Vectorizer.
- **Topic Modeling:** Identifying topics from the repository files using LDA.
- **Programming Language Detection:** Inferring the dominant programming languages based on keyword vocabularies.

# 2. Test Plan

## 2.1 Scope

This testing scope covers the verification of the following functionalities of the GitHub Repository Analyzer:

- Correct retrieval of files from public repositories.
- Accurate text preprocessing and removal of irrelevant elements (e.g., stopwords, numbers).
- Proper generation of TF-IDF matrices and identification of dominant keywords.
- Accurate identification of topics using LDA.

## 2.2 Testing Environment and Tools

The script testing was conducted in a Python environment using the following tools:

- **Python Libraries:**
  - **nltk:** Used for tokenization and stopword removal.
  - **scikit-learn:** For Count Vectorizer and TF-IDF Vectorizer implementation.For LDA-based topic modeling.
  - **TextBlob:** For determining the sentiment polarity.
  - **Matplotlib**: For plotting graphs.
- **GitHub API:** For fetching files from the user's repositories.
- **Ollama LLM version 3.2:** For summarizing and interpreting the data as a general text.

## 3. Literature Review

### Introduction

This section reviews techniques and approaches relevant to repository analysis, NLP preprocessing, and topic modeling in the context of GitHub repositories and software engineering research.

### Summary of Prior Research

- **Study the Correlation Between the README File of GitHub Projects and Their Popularity**:
  Previous studies have explored the relationship between GitHub repository popularity and the content within the README files. Research indicates that the structure, clarity, and content quality of README files can have a significant impact on the visibility and success of a project. TF-IDF (Term Frequency-Inverse Document Frequency) and LDA (Latent Dirichlet Allocation) have been applied to analyze these files and extract relevant keywords and topics that correlate with repository engagement and popularity.
- **How do Software Engineering Researchers Use GitHub? An Empirical Study of Artifacts & Impact**:
  This study investigates the ways in which software engineering researchers use GitHub as a platform for collaboration and dissemination of research artifacts. The research highlights that while GitHub is a valuable tool for version control and code sharing, it is underutilized in terms of extracting meaningful insights from repository content itself. Methods such as LDA for topic modeling and TF-IDF for keyword extraction have shown potential in revealing important patterns within research-driven repositories.

**Topic Modeling with LDA**:
Topic modeling using LDA has been widely studied as an effective technique for uncovering latent topics within large text datasets, including code

documentation and repository contents. The application of LDA on GitHub repositories has revealed that certain topics, such as programming languages, frameworks, and methodologies, frequently appear in popular repositories, making it a useful tool for understanding trends and interests within the developer community.

**GitHub Data Utilization**:
Most tools and studies focusing on GitHub data primarily analyze repository metadata, such as the number of stars, forks, and commits. While these approaches provide valuable statistical insights, they often overlook the textual content within repositories. This gap in content-level analysis is addressed by this project, which emphasizes the extraction of meaningful topics and keywords directly from code files and documentation.

**Gaps Addressed**
While previous research and tools have concentrated on repository metadata and statistical analysis, this project introduces a focus on content-level topic analysis and keyword extraction. By utilizing NLP techniques like LDA and TF-IDF, this project aims to provide deeper insights into the content of GitHub repositories, specifically in the context of README files and code files, which are often rich with information about project goals, technologies, and key features.

# 4. Dataset

### Introduction

This section describes the dataset fetched from public GitHub repositories and the preprocessing steps performed to prepare it for analysis.

### Source and Structure

The dataset comprises code files collected from public GitHub repositories, specifically from the profiles of the authors and their friends (Warlock1305, FarisMuhovic, Muha0644). Files were selected based on extensions commonly associated with programming languages, such as .py, .java, and .js.

### Example Dataset Entry:

| Repository Name | File Name | Content Example |
|---|---|---|
| Online Quiz Platform | index.html | “<!DOCTYPE html><br><html lang="en"><br>…” |

### Preprocessing Steps

1. **Stopword Removal**:
   Commonly occurring words (e.g., "the," "is," "in") are removed to focus on more meaningful content.
2. **Punctuation Removal**:
   All punctuation marks are stripped from the text to standardize the content for further analysis.

3. **Tokenization**:
   Code files are split into tokens (words, identifiers, etc.) for easier processing and analysis.

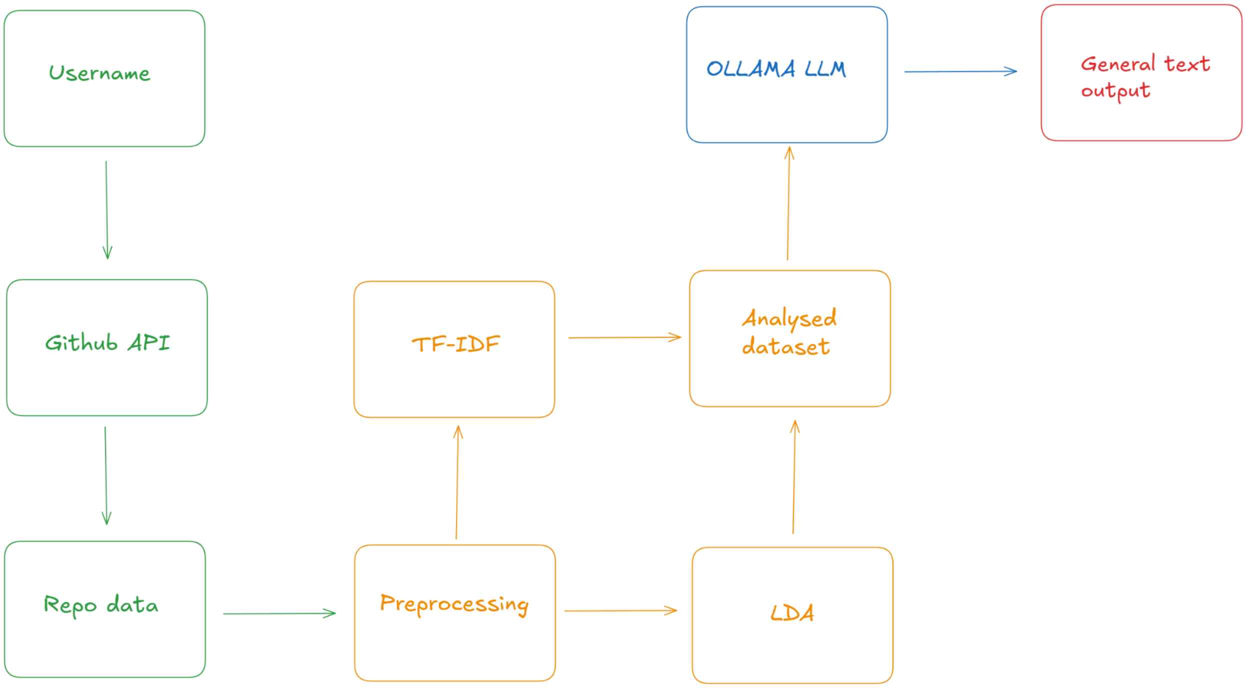
## 5. Methodology

### Introduction

This section outlines the steps followed in the script, from fetching files to analyzing topics and keywords.

### Workflow Diagram

### Figure 1: Workflow of the GitHub Repository Analyzer



### Steps in Detail

1. **File Retrieval:**

- Files are retrieved using the GitHub API, filtering for supported programming languages.

2. **Text Preprocessing:**
   - Tokenization and stopword removal are performed using nltk.
   - Numbers and punctuation are removed to clean the dataset.
3. **TF-IDF Vectorization:**
   - A vocabulary array is generated, and term frequencies are calculated using scikit-learn.
4. **Topic Modeling:**
   - LDA is used to identify latent topics in the cleaned dataset.

## 6. Results

### Introduction

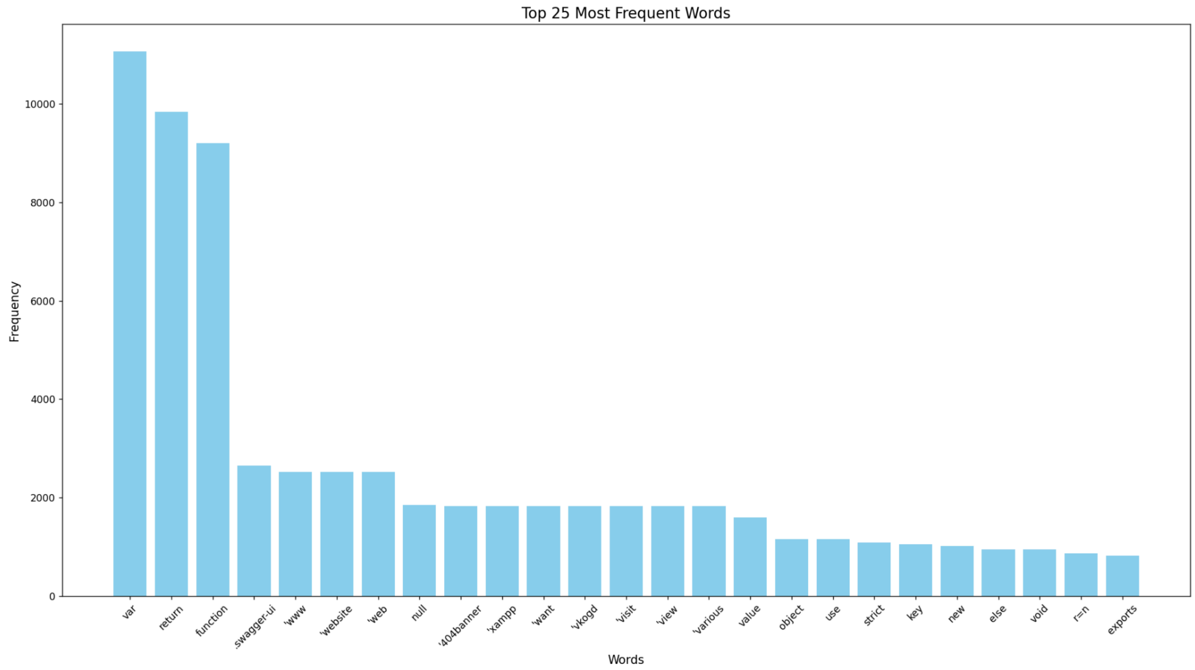
This section presents the analysis results, showcasing keyword extraction and topic modeling outputs.

### Evaluation Metrics

- **Topic Coherence:** Measures the interpretability of identified topics.
- **Keyword Significance:** Evaluated based on frequency and TF-IDF scores.

### Example Output from Analysis

```
“Project,Sentiment,Word Count,Sentence
Count,Sentiment,Topics,Keywords
Online-Quiz-Platform,-0.315625,235,38,-0.315625,"({'Topic': 'Topic 1',
'Relevant Words': ['ajax']}, {'Topic': 'Topic 2', 'Relevant Words': ['ajax']},
{'Topic': 'Topic 3', 'Relevant Words': ['html', 'css']})","{'JavaScript':
['var', 'function'], 'PHP': ['function']}"
```



# 7. Conclusion

## Key Takeaways

The GitHub Repository Analyzer successfully:

- Retrieves and preprocesses repository files.
- Extracts significant keywords using TF-IDF.
- Identifies latent topics using LDA.

## Limitations

- Analysis is limited to public repositories.
- Files without significant textual content may yield suboptimal results.
- Doesn’t identify latent topics without readme files.

## Future Work

- Extend support to private repositories via authentication.

- Improve LDA performance by integrating advanced topic coherence measures.
- Add visualization tools for better insights.

## 8. Abstract

The GitHub Repository Analyzer is a Python-based script designed to analyze public repositories of a specified GitHub user. Using natural language processing techniques, the script preprocesses repository files, extracts significant keywords with TF-IDF vectorization, and identifies latent topics using Latent Dirichlet Allocation. The analysis reveals programming language usage patterns and key topics present in repository files. While effective for content-level analysis, the tool's functionality is limited to public repositories, with future work focusing on private repository support and enhanced visualizations.

## References

1. Author(s) (Year). Study Name. Journal Name, Volume(Issue), Pages.
   DOI: 10.1016/j.jenvman.2023.117624
   Link: ScienceDirect Article
2. Author(s) (Year). Study Name. arXiv Preprint.
   URL: arxiv.org/abs/2310.01566v2