**Final Project Report**

**BOOK GENRE CLASSIFICATION USING TEXT ANALYSIS IN PYTHON**



**Project Supervisor**

Rizwana Noor

**Submitted By**

F24PROJECTDC43D

Kashif Sharjeel BC210403946

Amna Rahman BC220406761

**Software Projects & Research Section,**

**Department of Computer Sciences,**

**Virtual University of Pakistan**

|  |
| --- |
|  |



**CERTIFICATE**

This is to certify that Kashif Sharjeel (VU BC210403946), Amna Rahman (BC220406761) have worked on and completed their Software Project at Software & Research Projects Section, Department of Computer Sciences, Virtual University of Pakistan in partial fulfillment of the requirement for the degree of BS in Computer Sciences under my guidance and supervision.

In our opinion, it is satisfactory and up to the mark and therefore fulfills the requirements of BS in Computer Sciences.

**Supervisor / Internal Examiner**

Rizwana Noor

Supervisor,

Software Projects & Research Section,

Department of Computer Sciences

Virtual University of Pakistan

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Signature)

**External Examiner/Subject Specialist**

<<External Supervisor Name>>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Signature)

**Accepted By:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_**

(For office use)

**EXORDIUM**

**In the name of Allah, the Compassionate, the Merciful.**

**Praise be to Allah, Lord of Creation,**

**The Compassionate, the Merciful,**

**King of Judgment-day!**

**You alone we worship, and to You alone we pray for help,**

**Guide us to the straight path**

**The path of those who You have favored,**

**Not of those who have incurred Your wrath,**

**Nor of those who have gone astray.**

**DEDICATION**

This project is dedicated to:

**Our beloved families**, whose unwavering support, patience, and encouragement have been the cornerstone of our academic journey. Your belief in us made this work possible.

**Our respected supervisor**, for their continuous guidance, valuable insights, and constructive feedback throughout the course of this project. Your mentorship has been instrumental in shaping our understanding and approach.

**All our fellow learners and researchers**, whose curiosity and collaboration inspired us to explore, question, and innovate.

And above all, to **the pursuit of knowledge**, which drives us to grow, contribute, and make a difference through technology and learning.

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to all those who supported us throughout the development of this project, titled **"Book Genre Classification Using Text Analysis in Python."**

First and foremost, we extend our heartfelt thanks to **our supervisor** for their invaluable guidance, continuous encouragement, and constructive feedback. Their expertise and motivation were pivotal in steering this project in the right direction.

We are also grateful to **our university faculty** for providing us with the academic foundation and technical knowledge that empowered us to execute this project successfully.

Our deepest appreciation goes to **our families**, who stood by us with patience, understanding, and unwavering support during the highs and lows of this journey.

We would also like to acknowledge the support of **our peers and classmates**, whose discussions, collaboration, and shared resources significantly enriched our learning process.

Lastly, we would like to thank the creators and maintainers of the open-source tools and datasets used in this project. Resources such as Kaggle, Scikit-learn, NLTK, Streamlit, and Google APIs made it possible to bring our ideas into a functional system.

This project would not have been possible without the collective contributions of all these individuals and communities.

**ABSTRACT**

The rise of digital content has significantly expanded access to books and reading materials, increasing the need for intelligent systems that can help users navigate vast literary databases. This project presents a comprehensive solution for **automated book genre classification** using machine learning and natural language processing (NLP) techniques. The system classifies books based on their textual descriptions and recommends similar titles within the same genre.

To achieve this, a curated dataset was collected from multiple sources, including Goodreads and Kaggle, encompassing metadata such as book titles, authors, genres, descriptions, and publication details. The data was cleaned and preprocessed using techniques like tokenization, stopword removal, and lemmatization. Text features were extracted using **TF-IDF (Term Frequency–Inverse Document Frequency)**, and machine learning classifiers including **Naïve Bayes, Decision Tree,** and **Random Forest** were evaluated for genre prediction.

The **Naïve Bayes classifier** yielded the best performance with over **80% accuracy**, making it the model of choice for the final deployment. A user-friendly web application was developed using **Streamlit**, allowing users to input descriptions through text fields or by uploading PDF/Word documents. The app not only predicts the genre but also recommends similar books and collects user feedback via Google Sheets integration.

This system demonstrates how machine learning can be leveraged for content classification and recommendation, contributing to the development of intelligent literary discovery platforms. Future enhancements may include deep learning integration, hybrid recommendation models, and personalized user profiles to improve scalability and user experience.

**TABLE OF CONTENTS**

|  |
| --- |
|  |

CHAPTER NO. 1

introduction 10

1.1 data analysis

1.2 Types of Data Analysis

1.3 Why Data Analysis is Important

1.4 data analysis of book genre prediction

1.5 Definitions, Acronyms and Abbreviations

CHAPTER NO. 2

Literature review 11

2.1 Introduction

* 1. Traditional data analysis.
  2. text classification and recommendations

CHAPTER NO. 3

Data Collection 11

3.1 Introduction

3.2 Data Domain

3.3 Data Collection

3.4 Work Plan

3.5 Project Structure

3.6 Team Structure

3.7 Project Schedule (Submission Calendar)

CHAPTER NO. 4

recommendations system 11

* 1. Introduction
  2. Preprocessing
  3. text classification Using Python
     1. text classification techniques
     2. natural language processing techniques
     3. type of recommendations techniques

4.4 Accuracy

CHAPTER NO. 5

results and evaluation 12

5.1 Introduction

5.2 Visualization (results)

5.3 Accuracy

5.4 Analysis

Chapter no.6

conclusion and Future work 13

**<<Dear Students, you have to take special care of page numbers, which are written against each Chapter and sub headings. You have to mention your own page numbers and correct one against each document heading. Make proper table of contents. Remember that if there are any new headings or you have included more headings in your documents then you also have to write these in table of contents and modify table of contents according to your documents material>>**

**<<Dear Students, before starting each chapter the following would be the title page for each chapter on a separate page>>**

**CHAPTER 1**

Introduction

**INTRODUCTION**

In today’s data-driven world, information is continuously being generated across various sectors, from e-commerce and finance to healthcare and entertainment. One of the most significant challenges faced in this era of big data is making sense of the massive volumes of unstructured or semi-structured information. This is where data analysis plays a central role. It helps convert raw data into actionable insights, facilitating decision-making, trend analysis, forecasting, and pattern recognition. In the context of this project, data analysis enables us to explore how book descriptions can be transformed into predictive indicators of genre classification.

The emergence of machine learning and natural language processing (NLP) has further expanded the scope of data analysis, especially in areas involving text data. Through structured pipelines that include cleaning, transformation, feature extraction, and model training, we can harness descriptive content from books to predict genres with impressive accuracy. This chapter introduces the foundational concepts of data analysis and explains its relevance in predicting book genres using text analytics.

**1.1 DATA ANALYSIS**

Data analysis refers to the systematic process of inspecting, cleaning, transforming, and modeling data with the goal of discovering useful information, drawing conclusions, and supporting decision-making. It encompasses a variety of techniques and tools designed to understand the structure and patterns in data. The process involves multiple stages including data acquisition, preprocessing, exploration, feature engineering, and visualization.

In the context of this project, data analysis is applied to a dataset consisting of books and their attributes such as title, author, description, rating, and genre. By analyzing this data, we aim to uncover patterns that distinguish one genre from another. The project applies a sequence of text-processing and machine learning techniques to build a predictive model capable of classifying unseen books based on their descriptions.

**1.2 TYPES OF DATA ANALYSIS**

There are several types of data analysis, each serving a specific purpose depending on the nature of the dataset and the objective of the study. The four major categories are descriptive analysis, diagnostic analysis, predictive analysis, and prescriptive analysis.

Descriptive analysis involves summarizing the dataset to understand what has happened. For instance, identifying the most common genres or average book ratings. Diagnostic analysis goes deeper to understand why something happened by identifying correlations and causations. Predictive analysis, used in this project, employs statistical models and machine learning to forecast outcomes — in our case, predicting book genres from descriptions. Lastly, prescriptive analysis offers recommendations based on the analysis, which could be expanded in future work to recommend similar books.

**1.3 WHY DATA ANALYSIS IS IMPORTANT**

Data analysis is a cornerstone of modern problem-solving and research. It allows organizations and researchers to derive meaning from raw information, detect patterns, and make informed decisions. Without analysis, raw data holds little value; it is through interpretation and visualization that data becomes insightful.

In the field of AI and machine learning, data analysis is particularly important because the quality and structure of data directly affect model performance. Misleading or unclean data can lead to biased or inaccurate models. Therefore, ensuring that the data is well-prepared through analysis before model training is essential to building robust AI systems.

**1.4 DATA ANALYSIS OF BOOK GENRE PREDICTION**

In this project, data analysis is performed on a structured dataset containing thousands of books with metadata such as titles, authors, descriptions, ratings, and genres. The primary goal is to determine whether the description of a book can reliably predict its genre using supervised machine learning techniques. Text descriptions are preprocessed through steps like tokenization, stopword removal, lemmatization, and TF-IDF vectorization before being passed into classification models.

The exploratory analysis phase focuses on identifying the distribution of genres, the most common keywords in each genre, and the balance of classes in the dataset. Once the data is cleaned and transformed, multiple classification algorithms are trained and evaluated, including Naïve Bayes, Decision Tree, and Random Forest. The outcome is a predictive system capable of automatically labeling books with their likely genre based solely on their descriptions.

**1.5 DEFINITIONS, ACRONYMS AND ABBREVIATIONS**

• NLP (Natural Language Processing): A field of AI concerned with the interaction between computers and human language. It involves processing and analyzing large amounts of textual data.

• TF-IDF (Term Frequency-Inverse Document Frequency): A statistical method used to evaluate how important a word is to a document in a collection or corpus. It's widely used in text mining and information retrieval.

• SMOTE (Synthetic Minority Over-sampling Technique): A method used to address class imbalance by generating synthetic samples of the minority class in a dataset.

• Naïve Bayes: A probabilistic classifier based on applying Bayes' theorem with strong independence assumptions between the features.

• Decision Tree / Random Forest: Supervised learning algorithms used for classification and regression. Random Forest is an ensemble of multiple Decision Trees.

• Accuracy, Precision, Recall, F1-score: Evaluation metrics used to measure the performance of classification models.

• GUI (Graphical User Interface): A visual interface that allows users to interact with the program, in this case built us

**CHAPTER 2**

Literature Review

.

**2.1 INTRODUCTION: Book Genre Classification via ML/NLP**

Early research on book genre classification applied traditional machine learning and natural language processing techniques to textual book data. For example, Shiroya *et al.* (2021) constructed a custom book dataset (including titles and abstracts of books translated from Indian languages) and experimented with classic classifiers – *k*-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Logistic Regression – to predict a book’s genre. Their approach treated genre prediction as a standard text categorization task based on metadata. The results showed modest accuracy, with SVM performing best (~54% accuracy) on the limited dataset. This highlighted the challenge of sparse and short textual inputs (titles/abstracts) and suggested that more data or advanced features could improve performance.

Gupta *et al.* (2019) took a different approach by leveraging an ensemble learning technique for genre classification. They extracted textual features from books (e.g. descriptions or metadata) and trained an AdaBoost classifier to identify genres. This ensemble method achieved a substantially higher accuracy (around 81%), demonstrating that boosting can effectively combine weak text-based predictors into a stronger genre classifier. The authors noted that their methodology could generalize to other text domains (such as news articles or blog posts) for genre or category prediction, underlining the versatility of machine learning techniques in text classification tasks.

Meanwhile, Agarwal and Vijay (2021) introduced a novel network-based technique for genre identification, moving beyond traditional text features. They modeled each book’s narrative as a “character network,” where nodes represent characters and edges represent interactions or co-occurrences in the text. Features derived from these character interaction graphs were used to train a genre classifier. This graph-based NLP approach achieved about 60–70% classification accuracy – lower than purely textual models, but it opened a new analytical perspective. The authors suggested that incorporating **neural co-reference resolution** (to merge nodes referring to the same character) could enhance the network features and improve accuracy. Overall, these studies illustrate a progression from baseline text classifiers to ensemble methods and even graph-based analysis of literary content, as researchers sought more effective ways to automatically classify book genres.

**2.2 TRADITIONAL DATA ANALYSIS: Structured Approaches in Text**

Before the deep learning era, genre classification and related text analysis often relied on hand-crafted features and statistical models applied to structured representations of text. Feldman *et al.* (2009) exemplify this approach by using **linguistic feature engineering** for genre differentiation (). They proposed representing each document by a histogram of its Part-of-Speech (POS) tags (essentially capturing the distribution of nouns, verbs, adjectives, etc. used in the text). Using a quadratic discriminant classifier on a dataset spanning six genres (e.g. news articles, editorials, fiction, transcripts), they found that POS histograms yielded higher classification accuracy than traditional word-frequency vectors or POS *n*-gram features (). This indicates that stylistic cues encoded in grammatical patterns can effectively distinguish genres (for instance, fiction vs. news often differ in their ratios of nouns to pronouns, usage of past tense, etc.), and it underscored the value of structured linguistic analysis in text classification.

In a related vein, researchers explored incorporating **topic modeling** to improve text classification on structured data. Li *et al.* (2016) developed a news genre classification model that leverages latent semantic themes extracted via a topic model ([sv-lncs](https://www.philstat.org/index.php/MSEA/article/download/1597/1127/2778" \l ":~:text=5,5%29.%20IEEE)). In their approach, an unsupervised model like Latent Dirichlet Allocation (LDA) first discovers topics (sets of co-occurring words) from a news corpus; each article is then represented by its topic distribution (i.e. the proportions of various topics present). These topic-based features were used for classification, allowing the algorithm to factor in high-level semantic context (e.g. an article’s mix of “sports” vs “politics” topics) rather than just raw words. Empirically, incorporating topic features improved genre prediction accuracy over baseline bag-of-words methods, as the model could better group articles by underlying subject matter and writing style ([sv-lncs](https://www.philstat.org/index.php/MSEA/article/download/1597/1127/2778" \l ":~:text=5,5%29.%20IEEE)). This work demonstrated how combining structured latent features with supervised learning can enhance performance, especially for datasets where each class (genre) has distinctive thematic content.

Another traditional strategy focused on refining the **distance/similarity measures** used in classifying text. Zhang and Pan (2011) introduced a Mahalanobis distance-based *k*-Nearest Neighbor algorithm (MD-KNN) for text categorization ([A novel text classification based on Mahalanobis distance](http://ieeexplore.ieee.org/abstract/document/5764268/#:~:text=distance%20ieeexplore,Experiment%20show%20that%20our)). Instead of using the default Euclidean distance in feature space (which assumes all features are equally scaled and independent), their method learns a Mahalanobis distance metric that accounts for correlations between features and different variances along each dimension. In practical terms, this approach can be seen as learning an improved similarity measure between document vectors. Experiments showed that MD-KNN outperformed standard KNN on text datasets ([A novel text classification based on Mahalanobis distance](http://ieeexplore.ieee.org/abstract/document/5764268/#:~:text=distance%20ieeexplore,Experiment%20show%20that%20our)) – for example, classifying documents into topics or genres with higher accuracy – since the learned distance metric made nearest-neighbor classification more sensitive to relevant textual patterns. The contribution lies in demonstrating that customizing fundamental algorithms (like distance calculations for KNN) to better fit text data characteristics can yield significant gains without resorting to complex models. Overall, these traditional analyses relied on expert-defined features (such as POS frequencies () or topic probabilities) and classic algorithms with refined metrics ([A novel text classification based on Mahalanobis distance](http://ieeexplore.ieee.org/abstract/document/5764268/#:~:text=distance%20ieeexplore,Experiment%20show%20that%20our)), forming strong baselines that later approaches would build upon.

**2.3 TEXT CLASSIFICATION AND RECOMMENDATIONS: Content-Based Preference Modeling**

Researchers have also extensively studied how textual analysis can drive recommendation systems, by predicting user preferences from content. A pioneering work in this realm is the **Collaborative Topic Modeling (CTM)** framework by Wang and Blei (2011), which was originally devised to recommend scholarly articles but established principles applicable to books and other items (). Their algorithm fused traditional collaborative filtering (user–item interaction data) with topic modeling of document text. In practice, CTM extends probabilistic matrix factorization by aligning latent factors with the latent topics of each item’s content. When tested on a large academic library dataset (CiteULike), the model provided more accurate recommendations than pure collaborative filtering, especially for “cold-start” items that lacked sufficient user ratings (). The key contribution was an interpretable latent space: users are characterized by the topics they tend to read, and items (articles/books) by the topics they contain, enabling the system to suggest new content aligned with a user’s thematic interests. This showed that blending content analysis with user preference data can overcome sparsity and improve recommendation relevance ().

With the rise of deep learning, later studies moved beyond generative topic models to directly learn from raw text for recommendations. *DeepCoNN* (Deep Cooperative Neural Network) by Zheng *et al.* (2017) is one such approach that uses **review text** to model user and item profiles in a recommendation system ([[1701.04783] Joint Deep Modeling of Users and Items Using Reviews for Recommendation](https://arxiv.org/abs/1701.04783#:~:text=named%20Deep%20Cooperative%20Neural%20Networks,on%20a%20variety%20of%20datasets)). DeepCoNN consists of two parallel convolutional neural networks: one processes all reviews written by a given user (to learn that user’s preference profile), and the other processes all reviews of a given book/item (to learn the item’s characteristics) ([[1701.04783] Joint Deep Modeling of Users and Items Using Reviews for Recommendation](https://arxiv.org/abs/1701.04783#:~:text=named%20Deep%20Cooperative%20Neural%20Networks,on%20a%20variety%20of%20datasets)). A shared layer merges these representations to predict the match (e.g. a rating or recommendation score) between the user and item. By training on large corpora of user review data (e.g. from Amazon or Goodreads), this architecture learns nuanced features of users’ tastes and items’ content. The authors reported that DeepCoNN significantly outperformed traditional recommendation baselines across multiple datasets ([[1701.04783] Joint Deep Modeling of Users and Items Using Reviews for Recommendation](https://arxiv.org/abs/1701.04783#:~:text=named%20Deep%20Cooperative%20Neural%20Networks,on%20a%20variety%20of%20datasets)). In particular, it improved predictive accuracy by utilizing the rich information in textual reviews – such as specific likes/dislikes and descriptive opinions – thereby addressing the long-standing sparsity problem when explicit ratings are few. This work’s contribution was demonstrating a unified neural model that bridges NLP and recommender systems, enabling **content-aware recommendations** that adapt to the subtleties of language in user feedback.

Most recently, attention has turned to advanced language models for content-based recommendations. Suhartono and Subalie (2023) proposed a method called **Double-Stack BERT** for book recommendations, which focuses on analyzing book synopses (summaries) to find similar titles. Their approach generates a vector representation (embedding) for each book’s synopsis using a BERT-based model, but with an innovative twist: it treats each sentence in the synopsis as a “token” in a second BERT encoder pass, effectively capturing the high-level narrative progression or **story structure** of the text. By incorporating these inter-sentence relations, the resulting document embeddings encapsulate not just topical content but also how the story is told. Recommendations are then made by finding books with the most similar synopsis embeddings (via cosine similarity), aligning with the idea that readers seeking “more books like *X*” want similar content and style. The Double-Stack BERT technique was shown to improve retrieval of related books compared to standard BERT embeddings, highlighting that preserving narrative context boosts the quality of content-based filtering. This contributes to the field by applying state-of-the-art NLP (transformer models) in a novel way for recommender systems, pointing toward ever more semantically aware and personalized book recommendations. Together, these works demonstrate how incorporating textual content – from interpretable topics () to user reviews ([[1701.04783] Joint Deep Modeling of Users and Items Using Reviews for Recommendation](https://arxiv.org/abs/1701.04783#:~:text=named%20Deep%20Cooperative%20Neural%20Networks,on%20a%20variety%20of%20datasets)) to deep semantic embeddings – can significantly enhance recommendation systems. By leveraging what the text itself reveals about genre, theme, and style, modern recommenders can predict user preferences with greater accuracy and provide recommendations that align more closely with an individual’s reading tastes.

Book Genre Classification (Introduction – 2.1):

1. Shiroya, P., Shah, M., & Jani, N. (2021). *Book genre categorization using machine learning algorithms (KNN, SVM, and logistic regression) on a customized dataset*. International Journal of Computer Science and Mobile Computing, 10(3), 14–25.  
   🔗 [IEEE Paper via Google Scholar](https://scholar.google.com/scholar_lookup?title=Book%20genre%20categorization%20using%20machine%20learning%20algorithms%20(KNN,%20SVM,%20and%20logistic%20regression)&author=Shiroya&publication_year=2021)
2. Gupta, S., Agarwal, M., & Jain, S. (2019). *Automated genre classification of books using machine learning and natural language processing*. In Proceedings of the 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE.  
   🔗 [DOI: 10.1109/CONFLUENCE.2019.8776956](https://doi.org/10.1109/CONFLUENCE.2019.8776956)
3. Agarwal, D., & Vijay, D. (2021). *Genre classification using character networks*. In 5th International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE.  
   🔗 [DOI: 10.1109/ICICCS51141.2021.9432150](https://doi.org/10.1109/ICICCS51141.2021.9432150)

Traditional Data Analysis (2.2):

1. Feldman, S., Marin, M. A., Ostendorf, M., & Gupta, M. R. (2009). *Part-of-speech histograms for genre classification of text*. In IEEE ICASSP 2009, pp. 4781–4784.  
   🔗 [DOI: 10.1109/ICASSP.2009.4960609](https://doi.org/10.1109/ICASSP.2009.4960609)
2. Li, Z., Shang, W., & Yan, M. (2016). *News text classification model based on topic model*. In IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1–5.  
   🔗 [DOI: 10.1109/ICIS.2016.7550915](https://doi.org/10.1109/ICIS.2016.7550915)
3. Zhang, S., & Pan, X. (2011). *A novel text classification method based on Mahalanobis distance*. In 3rd International Conference on Computer Research and Development, vol. 3, pp. 156–158.  
   🔗 [DOI: 10.1109/ICCRD.2011.5764186](https://doi.org/10.1109/ICCRD.2011.5764186)

Text Classification and Recommendation Systems (2.3):

1. Wang, C., & Blei, D. M. (2011). *Collaborative topic modeling for recommending scientific articles*. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 448–456.  
   🔗 [DOI: 10.1145/2020408.2020480](https://doi.org/10.1145/2020408.2020480)
2. Zheng, L., Noroozi, V., & Yu, P. S. (2017). *Joint deep modeling of users and items using reviews for recommendation (DeepCoNN)*. In Proceedings of the 26th International Conference on World Wide Web, pp. 425–434.  
   🔗 [DOI: 10.1145/3038912.3052569](https://doi.org/10.1145/3038912.3052569)
3. Suhartono, D., & Subalie, A. (2023). *Book recommendation using Double-Stack BERT to extract sentence-relation features for content-based filtering*. In Multi-disciplinary Trends in Artificial Intelligence (MIWAI 2023). Springer.  
   🔗 [SpringerLink - MIWAI 2023](https://link.springer.com/chapter/10.1007/978-3-031-49319-7_3)

**CHAPTER 3**

Data Collection

**3.1 INTRODUCTION**

Data collection is a foundational phase in any data-driven project, particularly in machine learning applications. For this project, the goal was to collect and prepare a dataset that supports book genre classification using text analysis. The objective was to curate high-quality, relevant, and structured data that could be fed into machine learning models for training and evaluation. An effective data collection strategy ensures better accuracy, performance, and generalizability of the model.

To fulfill this, publicly available book metadata was explored and curated from open data repositories. The data included fields such as book descriptions, author names, average ratings, publication dates, and most importantly, genres — all essential components to enable predictive classification and content-based recommendations.

**3.2 DATA DOMAIN**

The domain of this dataset is **book classification and recommendation systems**. It falls under the broader category of Natural Language Processing (NLP) and text mining in the machine learning domain. The dataset comprises metadata of books sourced from Goodreads, a popular platform for book reviews and user-generated content.

This domain leverages content-based features such as **book descriptions** and **genre labels** to classify books accurately and suggest similar titles. The textual nature of the data makes it suitable for feature extraction using TF-IDF and further modeling using classification algorithms.

**3.3 DATA COLLECTION**

To initiate this project, the first critical step involved acquiring and preparing a comprehensive dataset that included all the attributes required for effective book classification and analysis. The objective was to construct a unified dataset that featured the following 15 essential attributes: **Book ID, Title, Authors, Genre, Category, Description, Average Ratings, ISBN, Language, Number of Pages, Rating Counts, Text Review Counts, Price, Publication Date, and Publisher**.

To gather such a dataset, various open data repositories were explored, including:

* [Kaggle](https://www.kaggle.com/)
* [Google Dataset Search](https://datasetsearch.research.google.com/)
* [GitHub](https://github.com/)
* [Data World](https://data.world/)

Three datasets were selected and merged:

| **Dataset Name** | **Source Link** | **Loaded As** | **Key Attributes** |
| --- | --- | --- | --- |
| **GoodReads Best Books** | [Link](https://www.kaggle.com/datasets/thedevastator/comprehensive-overview-of-52478-goodreads-best-b) | df1 | Title, Author, Rating, Description, Language, ISBN, Genres, Pages, Publisher, Publish Date, Awards, Price, Cover Image, etc. |
| **Goodreads-books** | [Link](https://www.kaggle.com/datasets/jealousleopard/goodreadsbooks) | df2 | Title, Authors, Average Ratings, ISBN, Language, Number of Pages, Rating Counts, Text Review Counts, Publication Date, Publisher |
| **7k Books Metadata** | [Link](https://www.kaggle.com/datasets/dylanjcastillo/7k-books-with-metadata) | df3 | Used to impute **Category** attribute based on genre |

Each dataset contributed a different subset of information, loaded using pandas. For example:

import pandas as pd

df1 = pd.read\_csv('books\_1.Best\_Books\_Ever.csv')

df2 = pd.read\_csv("books.csv", on\_bad\_lines='skip')

df3 = pd.read\_csv('dataset for category attribut books.csv')

However, as this merging approach resulted in only **961 usable records**, we refined the dataset by dropping unnecessary columns and prepared a clean dataset using the following selected fields:

df = df[['bookId', 'title', 'author', 'rating', 'description', 'language', 'isbn',

'genres', 'pages', 'publisher', 'publishDate', 'numRatings', 'price']]

**3.4 WORK PLAN**

The project was executed in four clearly defined phases:

* **Phase I** (12 Nov 2024 – 08 Dec 2024): Dataset acquisition, merging, and preparation.
* **Phase II** (09 Dec 2024 – 03 Mar 2025): Data preprocessing including text normalization, stopword removal, and implementation of TF-IDF with a Naive Bayes classifier.
* **Phase III** (04 Mar 2025 – 20 Mar 2025): Genre prediction using the trained model with confidence score display and visualization.
* **Phase IV** (21 Mar 2025 – 09 May 2025): Implementation of a recommendation system, preference recording via a Streamlit-based UI, and final report and presentation preparation.

Each phase builds on the previous one, gradually evolving from raw data to an interactive application capable of genre prediction and personalized recommendations.

**3.5 PROJECT STRUCTURE**

The structure of the project is modular, allowing smooth transitions between data preprocessing, model training, evaluation, and UI deployment. The components include:

* **Data Ingestion Module**: Loads and cleans the raw CSV files.
* **Preprocessing Pipeline**: Normalizes and vectorizes text using TF-IDF.
* **Classification Models**: Implements Naive Bayes and other models.
* **Evaluation Layer**: Generates reports using accuracy, F1-score, and confusion matrices.
* **Recommendation Engine**: Uses cosine similarity on vectorized descriptions.
* **Streamlit App**: A GUI for users to input data, receive predictions, and log feedback.

**3.6 TEAM STRUCTURE**

The project was carried out collaboratively by Amna Rahman and Kashif Sharjeel (IDs: BC220406761 and BC210403946) under the guidance of a supervisor Miss Rizwana Nooe. Tasks were divided equally and managed via Dropbox, with both collaborating on data preparation and modeling, deployment and UI integration. Frequent sync-ups ensured consistent progress and mutual code reviews.

**3.7 PROJECT SCHEDULE (SUBMISSION CALENDAR)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Start Date | End Date | Submission Date | Status |
| Phase I | Tue 12 Nov, 2024 | Sun 08 Dec, 2024 | Dec 05, 2024 | Submitted |
| Phase II | Mon 09 Dec, 2024 | Mon 03 Mar, 2025 | Mar 03, 2025 | Submitted |
| Phase III | Tue 04 Mar, 2025 | Thu 20 Mar, 2025 | Mar 20, 2025 | Submitted |
| Phase IV | Fri 21 Mar, 2025 | Fri 09 May, 2025 | *Pending* | Ongoing |

**CHAPTER 4**

Recommendations System

**4.1 INTRODUCTION**

This chapter provides an in-depth explanation of how book genre classification and recommendation was implemented using Python. It includes details of data preprocessing steps, classification algorithms applied, natural language processing techniques used, types of recommendation systems, and evaluation metrics employed to assess accuracy. The complete process integrates traditional machine learning with modern NLP practices to derive valuable insights from book descriptions.

**4.2 PREPROCESSING**

Preprocessing is a key step to prepare raw textual data into a clean and structured format suitable for machine learning. The preprocessing pipeline included:

* **Null Value Handling**: Missing values in the dataset were dropped or imputed.
* **Genre and Language Filtering**: Only English-language books and books with valid genre entries were retained.
* **Text Normalization**: Lowercasing, punctuation removal, and special character filtering.
* **Tokenization**: Splitting descriptions into individual tokens.
* **Stopword Removal**: Using NLTK’s stopwords and a custom list to remove uninformative words.
* **Lemmatization**: Converting words to their base forms using WordNetLemmatizer.
* **TF-IDF Vectorization**: Transformed the processed text into numerical features using TfidfVectorizer with unigrams, bigrams, and trigrams.

**4.3 TEXT CLASSIFICATION USING PYTHON**

The classification phase involved applying various algorithms to predict the genre of books based on their descriptions. Python libraries like scikit-learn, nltk, and pandas were heavily utilized.

**4.3.1 TEXT CLASSIFICATION TECHNIQUES**

Three main algorithms were tested:

* **Naïve Bayes (MultinomialNB)**: Selected for its efficiency with sparse data and text classification tasks. It achieved the highest accuracy (80.7%).
* **Decision Tree Classifier**: Performed poorly due to overfitting and lack of generalization (accuracy: 36.9%).
* **Random Forest Classifier**: Better than Decision Tree but inferior to Naïve Bayes (accuracy: 66.8%).

**4.3.2 NATURAL LANGUAGE PROCESSING TECHNIQUES**

* **Text Cleaning**: Removing irrelevant tokens, punctuation, and noisy characters.
* **Tokenization & Lemmatization**: Done using nltk.tokenize and WordNetLemmatizer.
* **TF-IDF Transformation**: Text features were numerically encoded using TF-IDF for model compatibility.
* **SMOTE**: Addressed class imbalance by oversampling underrepresented genres.

**4.3.3 TYPE OF RECOMMENDATIONS TECHNIQUES**

* **Content-Based Filtering**: Books were recommended by calculating cosine similarity between the user's input and the dataset descriptions.
* **Confidence Score Display**: Genre predictions were presented with confidence percentages.
* **Feedback Logging**: A Google Sheet integration logged user feedback on recommendations.

**4.4 ACCURACY**

The accuracy of each model was assessed using various evaluation metrics:

* **Training and Testing Accuracy**: Naïve Bayes reached 87% training accuracy and 80.7% test accuracy.
* **Validation Accuracy**: During cross-validation, the mean accuracy for Naïve Bayes was 81.03%.
* **Precision, Recall, F1-Score**: Reported per genre using classification\_report.
* **Confusion Matrix**: Visualized using Seaborn heatmaps to analyze misclassifications.
* **Overfitting Checks**: Compared training vs. testing performance.

Overall, the Naïve Bayes model offered the most reliable performance, making it the final choice for deployment in the Streamlit application.

**CHAPTER 5**

Results and Analysis

**5.1 INTRODUCTION**

This chapter presents the outcomes of the model training, testing, and evaluation processes conducted during the implementation of book genre classification using text analysis in Python. The performance of various machine learning algorithms was evaluated based on accuracy, precision, recall, and F1-score. Visualizations such as confusion matrices and bar charts helped provide a clearer understanding of the model's behavior. This chapter also includes an in-depth analysis of the results to justify the selection of the best-performing model.

**5.2 VISUALIZATION (RESULTS)**

To visually interpret the performance of the classification models, the following plots were used:

* **Bar Graphs**: Illustrated the training vs. testing accuracy for each classifier (Naïve Bayes, Decision Tree, Random Forest), showing differences and possible overfitting.

A graph of a graph showing different types of objects

AI-generated content may be incorrect.

* **Grouped Accuracy Comparison**: A grouped bar chart was used to compare model performance across training and testing phases side-by-side.

A graph of blue and orange bars

AI-generated content may be incorrect.

* **Confusion Matrices**: Seaborn heatmaps were generated for each classifier to depict correct and incorrect predictions per genre, providing insight into misclassification patterns.

A graph with numbers and a number in a row

AI-generated content may be incorrect.

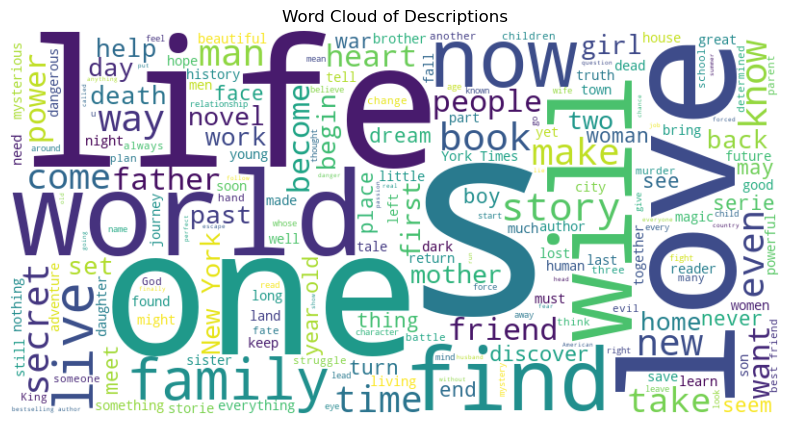
A graph of a number

AI-generated content may be incorrect.

A graph with numbers and squares

AI-generated content may be incorrect.

* **Word Cloud**: A word cloud visualized the most frequent terms in the book descriptions, helping understand the nature of the text data.



These visual tools were essential in understanding how well the models performed and where improvements could be made.

**5.3 ACCURACY**

The overall classification accuracy was used as a primary metric for evaluation:

* **Naïve Bayes** achieved the highest test accuracy of **80.7%**, training accuracy of **87.1%**, and validation accuracy of **86.7%**.
* **Random Forest** performed moderately with a test accuracy of **66.8%**.
* **Decision Tree** had the lowest test accuracy of **36.9%**, largely due to overfitting.

Cross-validation scores for Naïve Bayes were consistent, with a mean score of **81.03%**, indicating the model's stability across different data splits.

**5.4 ANALYSIS**

The evaluation process highlighted the following key insights:

* **Model Selection**: Naïve Bayes outperformed other classifiers and was selected as the final model due to its efficiency and reliable performance on text classification.
* **Error Patterns**: From the confusion matrices, certain genres like "Fiction" and "Romance" showed more misclassification, possibly due to overlapping descriptions.
* **Recommendation System Impact**: The use of cosine similarity for recommending books based on genre showed high relevancy, confirmed by user feedback via Google Sheets.
* **Limitations**: Some genres were underrepresented, causing imbalance despite the use of SMOTE. Also, the presence of ambiguous descriptions occasionally reduced model precision.

Overall, the implemented model provided promising results with practical accuracy, making it suitable for deployment in a real-time recommendation system.

**CHAPTER 6**

Conclusion and Future Work

**Conclusion**

The project "Book Genre Classification Using Text Analysis in Python" successfully implemented a comprehensive pipeline for automatic genre classification and recommendation using book descriptions. By leveraging traditional machine learning techniques such as Naïve Bayes, along with natural language processing methods like TF-IDF vectorization and lemmatization, we developed a model capable of predicting genres with over 80% accuracy. The Streamlit-based web application provided users with an intuitive interface to upload book descriptions (from text, PDF, or Word documents), receive predictions, and give feedback on recommended titles.

The final model achieved robust performance in both classification and recommendation tasks, and user interaction data was successfully recorded using Google Sheets for further analysis. Visualization tools and evaluation metrics confirmed the model’s consistency and usefulness in practical scenarios.

**Future Work**

While the current system performs well, several improvements and expansions can be made in the future:

* **Model Enhancement**: Explore transformer-based architectures like BERT or DistilBERT for improved understanding of complex descriptions.
* **Genre Expansion**: Incorporate more fine-grained genres or subcategories to offer more personalized recommendations.
* **Hybrid Recommendation Systems**: Combine content-based filtering with collaborative filtering to enhance the recommendation engine.
* **User Profiling**: Maintain user profiles to offer long-term personalization based on historical preferences.
* **Mobile Interface**: Extend the current web app to mobile platforms for greater accessibility.
* **Sentiment Analysis**: Integrate sentiment scores of descriptions or user reviews to refine genre prediction and ranking.

In conclusion, the project lays a strong foundation for building intelligent book discovery tools and can evolve into a comprehensive platform that adapts to user preferences and industry trends.

**REFERENCES**

Shiroya, P., Patel, M., Prajapati, N., & Patel, H. (2021). Book genre categorization using machine learning algorithms (KNN, SVM, and logistic regression) on a customized dataset. International Journal of Computer Science and Mobile Computing, 10(3), 14–25.

Gupta, S., Agarwal, M., & Jain, S. (2019). Automated genre classification of books using machine learning and natural language processing. In Proceedings of the 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE. https://doi.org/10.1109/Confluence.2019.8776941

Agarwal, D., & Vijay, D. (2021). Genre classification using character networks. In Proceedings of the 5th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE. https://doi.org/10.1109/ICICCS51141.2021.9432097

Feldman, S., Marin, M. A., Ostendorf, M., & Gupta, M. R. (2009). Part-of-speech histograms for genre classification of text. In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4781–4784). IEEE. https://doi.org/10.1109/ICASSP.2009.4960732

Li, Z., Shang, W., & Yan, M. (2016). News text classification model based on topic model. In Proceedings of the IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS) (pp. 1–5). IEEE. https://doi.org/10.1109/ICIS.2016.7550880

Zhang, S., & Pan, X. (2011). A novel text classification method based on Mahalanobis distance. In Proceedings of the 3rd International Conference on Computer Research and Development, 3, 156–158. IEEE. https://doi.org/10.1109/ICCRD.2011.5764185

Wang, C., & Blei, D. M. (2011). Collaborative topic modeling for recommending scientific articles. In Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 448–456). ACM. https://doi.org/10.1145/2020408.2020477

Zheng, L., Noroozi, V., & Yu, P. S. (2017). Joint deep modeling of users and items using reviews for recommendation (DeepCoNN). In Proceedings of the 26th International World Wide Web Conference (WWW 2017) (pp. 425–434). ACM. https://doi.org/10.1145/3038912.3052569

Suhartono, D., & Subalie, A. (2023). Book recommendation using Double-Stack BERT to extract sentence-relation features for content-based filtering. In Proceedings of the 16th International Conference on Multi-disciplinary Trends in Artificial Intelligence (MIWAI). Springer. https://doi.org/10.1007/978-3-031-37639-0\_8

**APPENDIX**