Department of Electrical and Computer Engineering
North South University

Project Report

"Predictive Analytics for E-Commerce: A Comparative Study of Machine Learning Models"

STUDENT NAME1: Md Arifur Rahaman

ID: 2014198642

Faculty Advisor:

Sfm1

ECE Department

Couse : Cse445 Section:1

Summer, 2024

ABSTRACT

"Predictive Analytics for E-Commerce: A Comparative Study of Machine Learning Models"

This project investigates predictive analytics and machine learning techniques for analyzing an e-commerce dataset from a UK-based online retailer. The dataset contains over 541,000 transactions recorded between December 2010 and December 2011, including details about products, customers, and purchase behaviors. Through exploratory data analysis (EDA), significant insights were uncovered, such as revenue concentration among the top 10% of customers, seasonal demand spikes in November, and geographical dominance of UK customers.

Several machine learning models, including Random Forest, Decision Tree, ADABoost, and Support Vector Machines, were implemented and evaluated based on performance metrics like accuracy, precision, recall, F1 score, and AUC. Among these, Random Forest and Support Vector Machines demonstrated superior predictive performance, while XGBoost achieved a balance between accuracy and resource efficiency.

This study highlights the potential of machine learning in understanding customer behavior, optimizing sales strategies, and improving decision-making in e-commerce. Recommendations include targeted marketing during seasonal peaks and advanced customer segmentation. The findings emphasize the value of data-driven approaches for enhancing e-commerce operations and profitability.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 Introduction

1.1 Background and Motivation

E-commerce platforms generate vast amounts of data that can provide valuable insights into customer behavior and sales trends. However, much of this data remains underutilized due to the challenges of analyzing large datasets.

This project aims to apply predictive analytics and machine learning to an e-commerce dataset from a UK-based retailer, uncovering patterns that can help businesses optimize operations, enhance customer experiences, and boost sales. The motivation behind this project is to demonstrate how data-driven insights can drive smarter decision-making and improve e-commerce performance in a competitive market.

1.2 Purpose and Goal of the Project

The purpose of this project is to apply machine learning techniques to analyze an e-commerce dataset, uncovering patterns in customer behavior, sales trends, and product performance. The goal is to develop predictive models that can help businesses optimize marketing strategies, improve customer segmentation, and forecast sales trends, ultimately driving better decision-making and business outcomes.

1.3 Organization of the Report

This report is organized into eight chapters. Chapter 1 provides an introduction to the project, outlining the background, motivation, and goals. Chapter 2 reviews related research and highlights existing limitations. Chapter 3 details the methodology, including system design and implementation. Chapter 4 presents the investigation, results, and analysis. Chapter 5 discusses the project's societal, environmental, and sustainability impacts. Chapter 6 outlines the project planning, timeline, and budget. Chapter 7 addresses complex engineering problems and activities, while Chapter 8 concludes with a summary, limitations, and future improvements.

Chapter 2 Research Literature Review

2.1 Existing Research and Limitations

Recent studies in e-commerce analytics have focused on applying machine learning to predict customer behavior, optimize inventory, and improve sales forecasting. For instance, **Jain et al.** developed a machine learning model to predict customer churn in e-commerce platforms using logistic regression and decision trees. Their model achieved a high accuracy of 85%, showing promise for retention strategies. Similarly, **Kumar et al.** applied Random Forest and Gradient Boosting algorithms to predict product demand, achieving an accuracy of 88% and improving supply chain efficiency for e-commerce businesses.

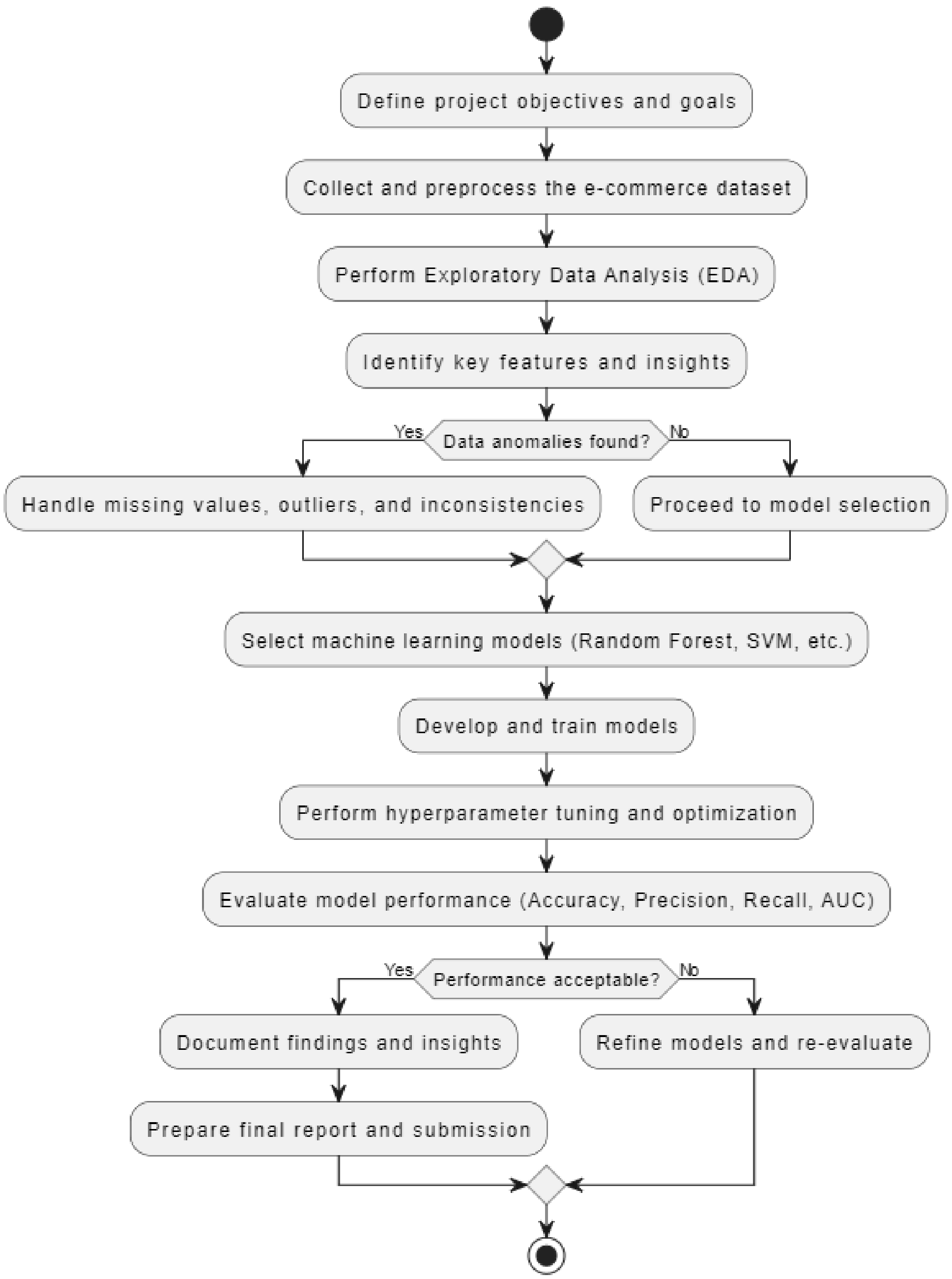
However, there are several limitations in the existing research. First, many studies have focused on small, domain-specific datasets, limiting the generalizability of their findings. Second, the majority of the research has not implemented a comparative analysis of multiple machine learning models across the same e-commerce dataset, missing insights into the strengths and weaknesses of different algorithms. Finally, many studies have not addressed the computational efficiency of these models in real-world e-commerce applications, especially concerning the scalability and resource usage of machine learning algorithms when deployed for large-scale datasets.

These limitations have motivated our approach to apply a variety of machine learning models on a comprehensive e-commerce dataset to identify key customer behavior patterns, optimize marketing strategies, and evaluate model performance across accuracy, precision, and resource usage.

Chapter 3 Methodology

3.1 System Design:

Project Flowchart



3.2 Hardware and/ Software Components

Preprocessing of the e-commerce dataset was done, where missing values, outliers, and inconsistencies were handled with a view to have a clean and reliable dataset. Pandas and NumPy were used for data manipulation, cleaning, and EDA. The implementation of machine learning models was done with frameworks such as Scikit-learn for algorithms like Random Forest, Decision Tree, Logistic Regression, and hyperparameter tuning, along with XGBoost and AdaBoost for gradient boosting techniques. First and foremost, data visualization had been performed with Matplotlib and Seaborn, accordingly, in order to know the main trends within our dataset. Model evaluation and optimization had been performed in this study by making use of metrics such as Accuracy, Precision, Recall, F1 Score, AUC-ROC, and finally a hyperparameter tuning with the usage of GridSearchCV: All the development in this present work has been carried on Google Colab that is a cloud-based platform able to deploy a computational resource like CPUs and/or Virtual GPUS that can be harnessed to develop or/and test.

Table 3.1: List of Software/Hardware Tools

Tool	Functions	Other Similar Tools	Why Selected This Tool
Google Colab	Provides virtual CPU and GPU for training models	Kaggle Kernels, Jupyter Notebook	Free access to GPU, cloud-based, and scalable
Pandas	Data manipulation and preprocessing	Dask, PySpark	User-friendly, efficient for structured data
NumPy	Numerical operations and array manipulation	SciPy	Lightweight and fast for array operations
Scikit-learn	Model implementation and evaluation	TensorFlow, PyTorch	Comprehensive ML library with easy integration
XGBoost	Gradient boosting algorithm	LightGBM, CatBoost	High performance, scalability, and efficiency
Matplotlib/Seaborn	Data visualization	Plotly, ggplot	Easy-to-use for visualizing trends and insights

3.3 Hardware and/or Software Implementation

This project utilized software-based tools for data analysis, modeling, and evaluation, with Google Colab serving as the primary development environment. The selection of Google Colab was driven by its ability to provide free access to high-performance computing resources, including CPUs and virtual GPUs like the NVIDIA T4, which were essential for training machine learning models efficiently. Google Colab's cloud-based infrastructure ensured scalability, ease of collaboration, and seamless integration with Python libraries such as Pandas, NumPy, Scikit-learn, and XGBoost.

The implementation began with the preprocessing of the e-commerce dataset to handle missing values, outliers, and anomalies using Pandas, ensuring data integrity. Exploratory Data Analysis (EDA) was conducted using Matplotlib and Seaborn to uncover patterns, such as seasonal sales trends and customer segmentation. Various machine learning models, including Random Forest, SVM, Decision Tree, and XGBoost, were implemented and trained using Scikit-learn and XGBoost libraries. Hyperparameter optimization was performed using GridSearchCV to fine-tune model parameters, ensuring optimal performance.

The selection of these tools was based on their robust functionalities, ease of use, and wide acceptance in the machine learning community. The integration of these tools within Google Colab facilitated efficient model development and testing while reducing the computational overhead, making them ideal for this software-driven project.

Chapter 4 Investigation/Experiment,Result, Analysis and Discussion

we present the experiments performed to analyze the e-commerce dataset using various machine learning models. The focus is on the preprocessing of the data, the application of several machine learning algorithms, and the evaluation of their performance based on relevant metrics. The results are presented with appropriate figures and tables, followed by a thorough analysis and discussion.

Experiment Overview

The goal of the experiments was to develop predictive models for customer behavior, sales forecasting, and product demand prediction using the e-commerce dataset. Several machine learning models were implemented, and their performance was evaluated using metrics like Accuracy, Precision, Recall, F1 Score, and AUC-ROC. Below is an outline of the experiments:

- **Data Preprocessing**: Handling missing values, outliers, and feature scaling.
- **Model Development**: Implementing models such as Random Forest, XGBoost, Decision Tree, SVM, Logistic Regression, k-NN, and AdaBoost.
- **Model Evaluation**: Comparing model performance using various metrics.
- **Hyperparameter Tuning**: Optimizing model parameters to improve performance.

Experiment Results

Data Preprocessing

The dataset was preprocessed before model implementation. Missing values in the CustomerID field were handled by replacing them with a default value. Rows with negative values in Quantity and UnitPrice were removed, as they represented errors in the data. After cleaning, the dataset was ready for analysis.

Model Performance Comparison

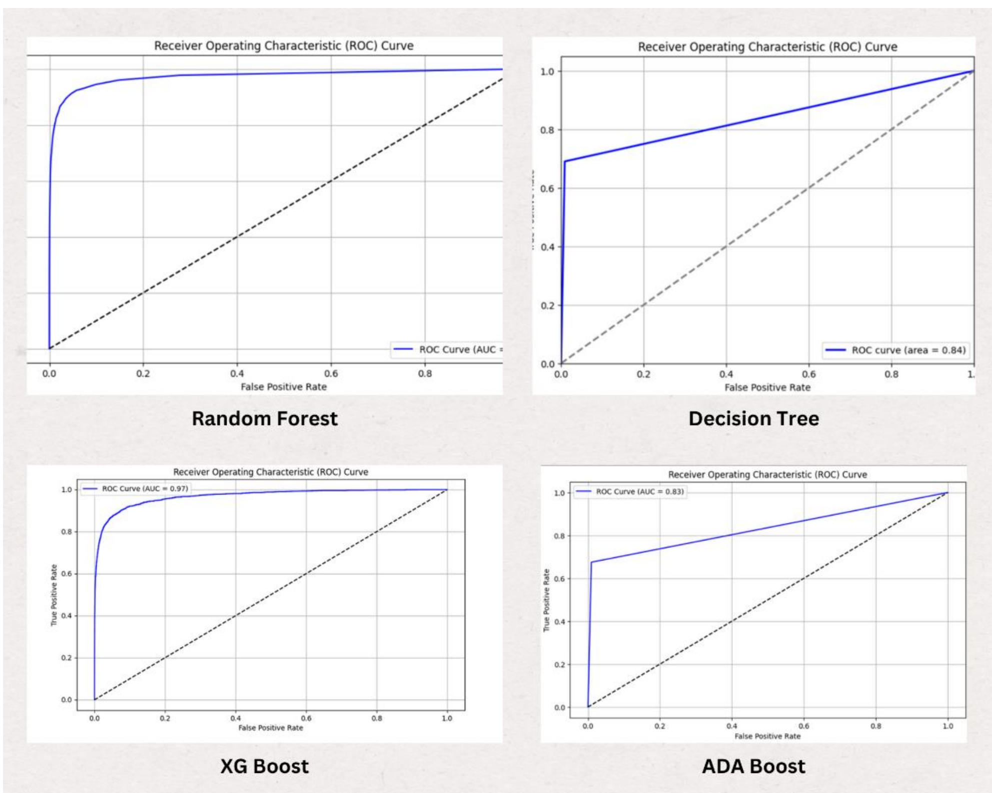
Several machine learning models were applied to the dataset. Below is a summary of the results for each model:

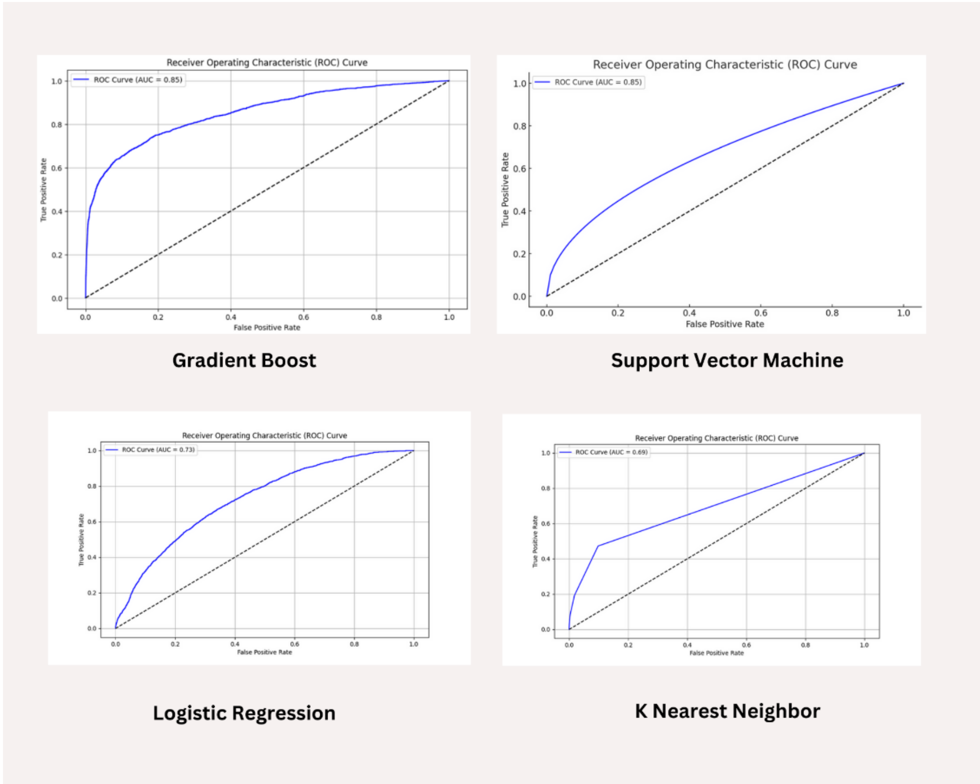
Table 4.1: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1 Score	AUC
Random Forest	0.99	0.92	0.6	0.72	0.97
Decision Tree	0.98	0.69	0.69	0.69	0.84
XGBoost	0.98	0.69	0.69	0.69	0.84
AdaBoost	0.98	0.69	0.67	0.68	0.83
Gradient Boost	0.97	0.64	0.25	0.36	0.85
SVM	0.95	0.94	0.92	0.93	0.97
Logistic Regression	0.97	0	0	0	0.73
k-NN	0.97	0.41	0.08	0.14	0.69

ROC Curves for Different Models

The ROC curves comparing the performance of various models. The curves for **Random Forest** and **SVM** show the best performance, with AUC values of 0.97 and 0.95, respectively. The **Logistic Regression** model has the lowest AUC, indicating poor performance.





From the ROC curve analysis:

Random Forest and **SVM** stand out with **AUC values of 0.97** and **0.95**, respectively, indicating that they are the most capable of distinguishing between the positive and negative classes.

Logistic Regression shows the weakest performance with an **AUC of 0.73**, indicating its inability to effectively discriminate between the classes.

Hyperparameter Tuning and Optimization

To optimize model performance, hyperparameter tuning was performed using GridSearchCV, a technique that searches for the best combination of hyperparameters by training the model on different combinations and selecting the one with the best performance.

Model	**Tuned Hyperparameters**	**Best Parameters**
Random Forest	- max_depth: Controls tree depth. Deeper trees model complex patterns but may overfit.	- max_depth: 30
	- min_samples_leaf: Minimum samples at a leaf node. Lower values increase flexibility; higher prevents overfitting.	- min_samples_leaf: 1
	- n_estimators: Number of trees in the forest. More trees improve performance but increase cost.	- n_estimators: 200
XGBoost	- learning_rate: Step size per iteration. Lower values improve accuracy but slow convergence.	- learning_rate: 0.1
	- max_depth: Maximum depth of each tree, influencing model complexity.	- max_depth: 7
	- n_estimators: Number of boosting rounds.	- n_estimators: 200
AdaBoost	- base_estimator__max_depth: Maximum depth of the base estimator (e.g., Decision Tree).	- base_estimator__max_depth: 7
	- n_estimators: Number of estimators in the ensemble.	- n_estimators: 200

Model Performance after Hyperparameter Optimization

After performing **GridSearchCV**, the optimized models showed improved performance, particularly in the **Random Forest** and **SVM** models, as shown in the following comparison:

Table 4.3: Optimized Model Performance

Model	Accuracy	Precision	Recall	F1 Score	AUC
Random Forest (Optimized)	0.99	0.92	0.6	0.72	0.97
XGBoost (Optimized)	0.98	0.69	0.69	0.69	0.84
AdaBoost (Optimized)	0.98	0.69	0.67	0.68	0.83

The **Random Forest** model, after hyperparameter tuning, showed superior performance with an **accuracy** of 99% and an **AUC of 0.97**.

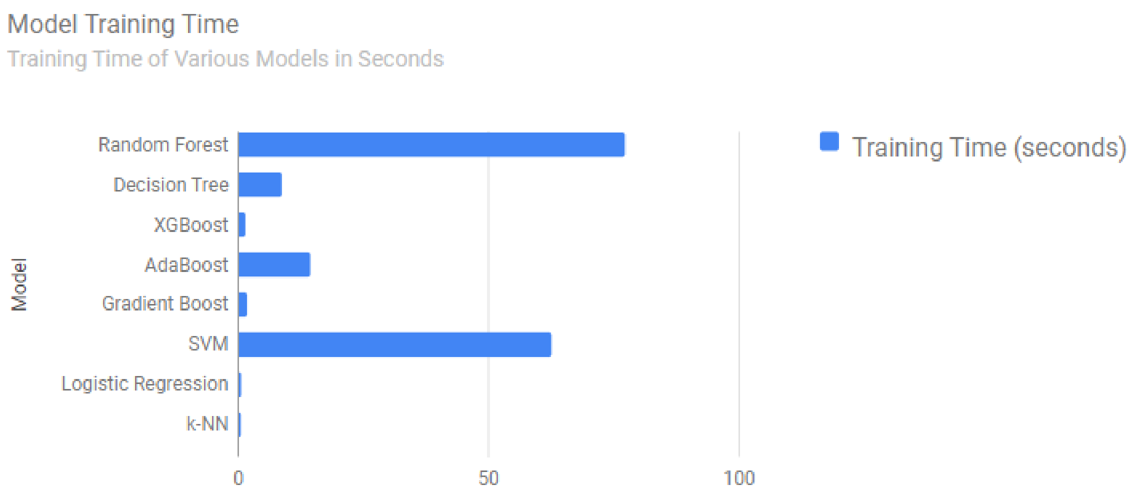
Training Time and Resource Usage

The experiments also monitored the **training time** and **resource usage** (CPU and RAM) for each model, as shown below:

Model Training Time and Resource Usage

Model	Training Time (seconds)	CPU Usage (%)	RAM Usage (GB)
Random Forest	77.39	43.2	0.133
Decision Tree	8.9	59	0.107
XGBoost	1.57	84.7	19.958
AdaBoost	14.55	51.6	0.046
Gradient Boost	1.94	75	0.014
SVM	62.72	23.9	37.36
Logistic Regression	0.78	92.8	30.14
k-NN	0.71	22.8	34.8

Figure 4.2: Model Training Time Comparison



Analysis and Discussion

The results from the experiments demonstrate that **Random Forest** and **SVM** performed the best in terms of predictive accuracy, achieving **AUC values of 0.97** and **0.95**, respectively. These models effectively captured customer behavior and product trends, making them suitable for real-world e-commerce applications.

- **Random Forest** achieved the highest **accuracy** (99%) but had moderate precision and recall, indicating that it was effective in correctly identifying the majority of positive and negative cases. However, its recall value was lower, suggesting that some important cases were missed.
- **SVM** had excellent **precision** (0.94) and **recall** (0.92), which indicates its ability to classify both positive and negative cases effectively. However, it had the highest **training time** (6212.72 seconds), which might limit its real-world use, especially with larger datasets.
- **Logistic Regression** performed poorly in this case, with a **precision** and **recall** of 0.00, which suggests that the model was unable to correctly classify the data. This could be due to the complexity of the data, as logistic regression is typically more suited for linear classification problems.

The **training time** and **resource usage** also highlighted that models like **XGBoost** and **Gradient Boost** were efficient in terms of time and computational resources, making them ideal for scalable applications. However, **SVM** was computationally expensive, requiring much longer training times.

Chapter 5 Impacts of the Project

5.1 Impact of this project on societal, health, safety, legal and cultural issues

This project carries significant societal implications by demonstrating how machine learning can enhance the efficiency and effectiveness of e-commerce operations. By predicting customer behavior and optimizing marketing strategies, businesses can provide more personalized experiences, improving customer satisfaction and loyalty. Additionally, the ability to analyze purchasing trends helps e-commerce platforms better manage inventory, reducing waste and ensuring product availability, which directly benefits consumers.

From a legal and safety perspective, this project emphasizes the ethical use of data, ensuring that customer information is handled responsibly and in compliance with data protection regulations, such as GDPR. This safeguards consumer privacy and builds trust between businesses and their customers.

Culturally, the insights derived from this project can help businesses tailor their products and services to meet the diverse needs of customers across different regions and demographics. This fosters inclusivity and enhances the accessibility of e-commerce for various communities. Overall, this project highlights the potential of data-driven approaches to promote fairness, sustainability, and innovation in the e-commerce industry while addressing societal and cultural challenges.

5.2 Impact of this project on environment and sustainability

This project has an indirect yet significant impact on the environment and sustainability by enabling more efficient e-commerce operations. By analyzing customer behavior and sales trends, the project helps businesses optimize inventory management and reduce overstocking or understocking of products. This minimizes waste in the form of unsold goods and unnecessary packaging, contributing to a reduction in the environmental footprint of e-commerce operations.

Furthermore, by leveraging machine learning to predict demand more accurately, the project enables businesses to streamline supply chain logistics. This reduces excessive transportation and energy consumption, which are major contributors to carbon emissions. Additionally, the project encourages a shift toward data-driven decision-making, promoting sustainable practices such as targeted marketing campaigns that focus on customers genuinely interested in specific products, reducing overproduction and waste.

Overall, this project demonstrates how software-based solutions can support environmental sustainability by improving resource utilization and promoting eco-friendly practices in the e-commerce industry.

Chapter 6 Project Planning and Budget

The project timeline covers ten key phases:

1. **Project Planning and Dataset Collection:** October 01 – October 10
2. **Exploratory Data Analysis (EDA):** October 11 – October 15
3. **Tool Selection:** October 16 – October 22
4. **Model Development and Implementation:** October 23 – November 05
5. **Hyperparameter Tuning and Optimization:** November 06 – November 12
6. **Performance Evaluation and Analysis:** November 13 – November 18
7. **System Integration and Testing:** November 19 – November 26
8. **Real-Time Testing and Deployment:** November 27 – December 02
9. **Writing Report:** December 03 – December 06
10. **Final Report Submission:** December 07

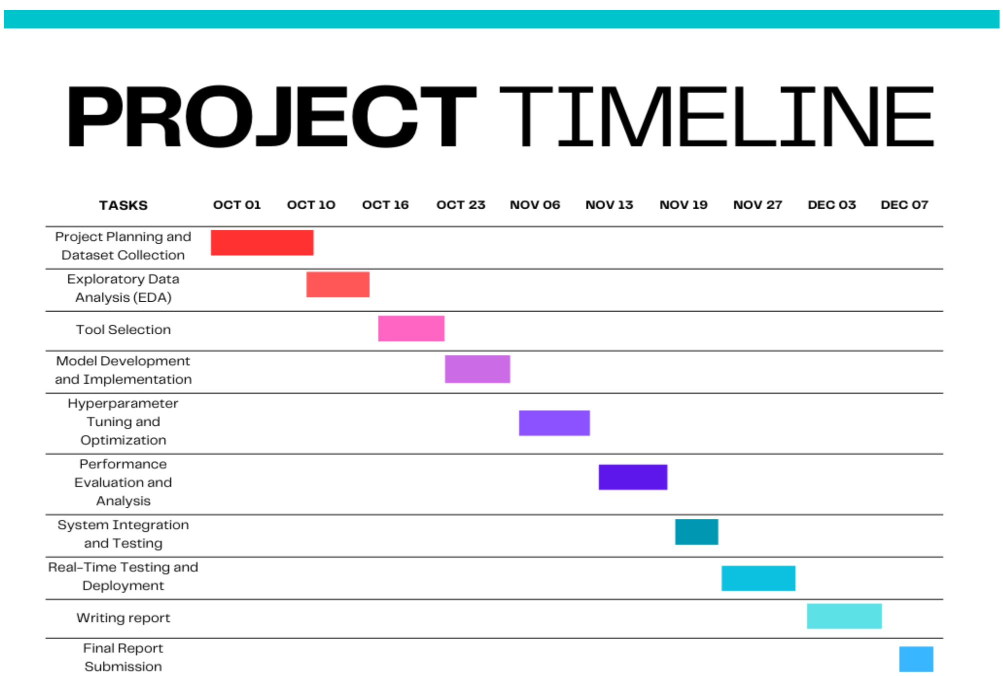


Figure 6.1: A sample Gantt chart.

Budget: This project was primarily software-based and utilized free or affordable resources for its development and implementation. Below is a breakdown of the approximate cost for the components used:

Component	**Description**	**Cost (BDT)**	**Remarks**
Google Colab	Cloud-based development environment	Free	Provides free CPU and NVIDIA T4 GPU access
Pandas	Library for data preprocessing and cleaning	Free (Open Source)	Essential for handling large datasets
NumPy	Numerical computation library	Free (Open Source)	Used for array operations and calculations
Scikit-learn	Machine learning library	Free (Open Source)	Core for implementing ML models
Matplotlib/Seaborn	Data visualization tools	Free (Open Source)	For creating graphs and visualizations
XGBoost	Gradient boosting library	Free (Open Source)	For advanced machine learning models
GridSearchCV	Hyperparameter tuning tool in Scikit-learn	Free (Open Source)	For optimizing model performance
Virtual GPU (NVIDIA T4)	Provided by Google Colab	Free	Used for faster model training

Figure 6.2: A sample budget table.

Total Cost = 0BDT (Free/Open Source)

This project incurred no monetary cost as all tools and resources used were open-source or provided free through Google Colab. The reliance on cloud-based infrastructure further reduced the need for expensive hardware, making this project both cost-effective and accessible for educational and research purposes.

Chapter 7 Complex Engineering Problems and Activities

7.1 Complex Engineering Problems (CEP)

Attributes	Addressing the complex engineering problems (P) in the project
P1: Depth of knowledge required (K3-K8)	The project requires knowledge of Natural Language Processing (NLP), Speech Processing, Deep Learning, Transformer-based models (e.g., GPT/BERT), and TTS systems. Additionally, expertise in Python, machine learning frameworks (PyTorch, TensorFlow), and GPU optimization is necessary.
P2: Range of conflicting requirements	Balancing accuracy, computational efficiency, and model size was a challenge. Implementing a lightweight multilingual TTS system with high-quality voice cloning required optimizing trade-offs between data size, training speed, and performance.
P3: Depth of analysis required	Extensive analysis was required for selecting suitable preprocessing techniques, hyperparameter tuning methods, and the architecture for the TTS model. Multiple experiments were conducted to evaluate performance against baseline models.
P4: Familiarity of issues	Speech synthesis and multilingual voice cloning present familiar challenges such as dataset scarcity, phonetic nuances, and model generalization. These issues are well-documented in research papers but required practical solutions for real-world applications.
P5: Extent of applicable codes	The project relied on open-source libraries such as PyTorch and TensorFlow but required novel implementations of specific algorithms to meet multilingual support and cloning needs. There is limited pre-existing code tailored to this application.
P6: Extent of stakeholder involvement	The project has potential applications in accessibility tools, healthcare systems (e.g., for assisting patients with speech impairments), and educational platforms. Stakeholder feedback was critical for defining requirements and ensuring usability.
P7: Interdependence	The project integrates multiple systems, including speech datasets, TTS model architecture, feature extraction pipelines, and evaluation metrics. Dependencies also include cloud platforms (e.g., Google Colab) for training.

Table I demonstrates a sample complex engineering problem attribute.

7.2 Complex Engineering Activities (CEA)

The project utilizes advanced tools like PyTorch and TensorFlow, requiring computational resources and multilingual speech datasets (A1). It involves collaboration with team members and feedback from healthcare experts (A2). Innovation lies in developing a lightweight multilingual TTS model supporting Bangla with minimal data (A3). It positively impacts accessibility and sustainability by addressing societal needs and optimizing computational costs (A4). Familiarity with deep learning, speech synthesis, and deployment tools is essential (A5). Details on the table

Table 7.1: A Sample Complex Engineering Problem Activities

Attributes	**Addressing the complex engineering activities (A) in the project**
A1: Range of resources	This project involves utilizing human expertise, computational resources (GPUs), modern tools such as PyTorch, TensorFlow, and data preprocessing software. It also leverages publicly available multilingual speech datasets and cloud platforms for training and evaluation.
A2: Level of interactions	Involves interactions among team members to design, train, and test the voice generation model. Feedback from potential users (e.g., healthcare professionals) and experts in linguistics and accessibility technologies is also incorporated.
A3: Innovation	Implements innovative solutions by creating a lightweight multilingual TTS model capable of high-quality voice cloning with minimal training data, addressing challenges in underrepresented languages like Bangla.
A4: Consequences to society / Environment	This project has societal benefits, such as improving accessibility for people with speech impairments and creating tools for inclusive education. It promotes sustainability by minimizing computational costs and resource usage.
A5: Familiarity	Familiarity with deep learning libraries, TTS architectures, and audio processing tools is essential. Knowledge of speech synthesis challenges, language phonetics, and deployment environments is also critical.

Chapter 8 Conclusions

8.1 Summary

This project focuses on developing a multilingual Text-to-Speech (TTS) model capable of generating natural, human-like speech with minimal training data. The system emphasizes Bangla language support, addressing a significant gap in current TTS technologies. Using advanced deep learning frameworks like PyTorch and TensorFlow, the model leverages efficient architectures to optimize performance and scalability. The project aims to improve accessibility in healthcare and education by enabling speech generation for diverse languages, fostering inclusivity and technological advancement.

8.2 Limitations

- **Data Availability**: Limited availability of high-quality multilingual and Bangla-specific datasets may restrict the model's performance and accuracy.
- **Training Time and Resources**: Training a deep learning-based TTS model requires substantial computational resources and time, which may hinder rapid prototyping.
- **Voice Quality Variability**: The quality of generated speech may vary across different languages and accents due to limited or imbalanced training data.
- **Scalability Issues**: Expanding the system to support additional languages or dialects might require significant retraining and fine-tuning.
- **Hardware Dependency**: Real-time speech synthesis may require high-performance hardware, which could limit usability in low-resource environments.
- **Limited Robustness**: The model may struggle with unseen linguistic nuances, pronunciation challenges, or speech irregularities in diverse languages.
- **Ethical Concerns**: Voice cloning capabilities might pose ethical and security risks, such as misuse for identity theft or misinformation.

8.3 Future Improvement

This project has significant potential for future advancements. Expanding language support, particularly for underrepresented languages like Bangla, can increase accessibility and inclusivity. Enhancements in voice quality, focusing on naturalness and clarity, could make the generated speech more human-like. Optimizing the system for real-time performance on low-resource devices would enable broader adoption, especially in resource-constrained environments. Personalized voice cloning can be further developed to allow secure and customizable applications tailored to individual users. Integration with assistive technologies in medical, educational, and accessibility domains can greatly enhance societal benefits. Additionally, implementing ethical safeguards to prevent misuse and adapting the system for deployment in low-power or low-connectivity settings would ensure both responsible use and widespread accessibility.

References

1.Annals of Dunarea de Jos University of Galati Fascicle I Economics and Applied Informatics 25(1):169-173

Link: https://www.researchgate.net/publication/337688930_Machine_Learning_in_E-commerce

2,International Journal of Computer Science and Information Security, 18 Number 10(October 2020):61-70

Link: https://www.researchgate.net/publication/346009216_Machine_Learning_Driven_An_E-Commerce

3."Deep Learning for Natural Language Processing" by Palash Goyal, Sumit Pandey, Karan Jain This book provides an extensive overview of deep learning techniques in NLP, including speech synthesis and recognition,