A Project Report on

Forecasting Train Delays and Providing Information to the Passengers

submitted in partial fulfillment for the award of

Bachelor of Technology

in

Computer Science & Engineering

by

D.Kusumitha (Y20ACS433) **CH.Sri Vyshnavi (Y20ACS429)**

CH .Greeshma (Y20ACS421) **I. Kavya Sudha(Y20ACS460)**

Under the guidance of

Dr .D. Kishore Babu

Associate Professor

Department of Computer Science and Engineering

Bapatla Engineering College

(Autonomous)

(Affiliated to Acharya Nagarjuna University)

BAPATLA – 522 102, Andhra Pradesh, INDIA

2023-2024

Department of
Computer Science & Engineering

<u>CERTIFICATE</u>

This is to certify that the project report entitled **<u>Forecasting Train Delays and Providing Information to the Passengers.</u>**that is being submitted by D.Kusumitha (Y20ACS433), CH.Sri Vyshnavi (Y20ACS429), CH.Greeshma (Y20ACS421), and I.Kavya Sudha(Y20ACS460) in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

Signature of the Guide	**Signature of the HOD**
Dr.D.Kishore Babu	**Dr. M Rajesh Babu**
Associate Professor	**Assoc Prof. & Head**

DECLARATION

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

D.Kusumitha (Y20ACS433)
CH.Sri Vyshnavi (Y20ACS429)
CH.Greeshma (Y20ACS421)
I.Kavya Sudha(Y20ACS460)

Acknowledgement

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **Dr.D.Kishore Babu, Associate Professor,** Department of CSE, for his valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Dr.M.Rajesh Babu**, Prof. & Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal **Dr.Nazeer Shaik,** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator **Dr. N. Sudhakar,** Prof. Dept. of CSE for his helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

D.Kusumitha (Y20ACS433)
CH.Sri Vyshnavi (Y20ACS429)
CH.Greeshma(Y20ACS421)
I.Kavya Sudha(Y20ACS460)

Abstract

Efficient transportation infrastructure is the backbone of modern societies, facilitating the movement of people and goods across vast distances. However, the reliability of this infrastructure is often challenged by unpredictable train delays, which can have far-reaching implications for commuters and the broader economy. These delays can disrupt daily routines, affecting individuals' ability to get to work on time, attend school, or keep appointments, leading to frustration and inconvenience. With the existing system it is hard to forecast the train delays because a large amount of historical data is not available to train and test the model. So to address this issue, this project proposes a solution by working on the Real time data provided by Rapid API and applying the cosine similarity and TF-IDF techniques of Machine Learning.

As individuals traverse borders and embark on journeys that span diverse regions, linguistic differences often pose obstacles to efficient communication and information exchange within the railway ecosystem. The language barrier manifests in various aspects of railway travel, from ticket booking and itinerary planning to onboard announcements and customer service interactions. For travellers who are not proficient in the predominant language(s) of the railway system they are navigating, accessing essential information and services can be daunting and, at times, frustrating.

In conclusion, the Multilingual Railway Enquiry Webpage represents a significant step towards creating a more connected and accessible railway network. Enabling Railway operators to efficiently disseminate information to global audience while streamlining customer support processes. By embracing multilingualism in the railway industry can enhance accessibility, promote tourism and remove the Barrier.

Table of Contents

Abstract iv

Table of Contents v

List of Figures: viii

1 Introduction 1

1.1 Train Delays 1

1.2 Importance of Train Delay Prediction 2

1.3 Statement and Solution to the Problem 2

1.3.1 Problem Statement 2

1.3.2 Solution to the Problem 3

1.4 Scope of the Project 3

1.5 Machine Learning 4

1.5.1 Why Choosed Machine Learning ? 5

2 Literature Survey 6

3 System Design 8

3.1 Existing System 8

3.1.1 Random Forest Algorithm 9

3.1.2 Gradient Boosting Algorithm 9

3.2 Proposed System 10

3.2.1 Cosine Similarity 10

3.2.2 TF-IDF 11

3.2.3 Django 11

3.2.4 LLAMA -2-7B 12

4 Requirement Analysis 13

4.1 Software Requirements 13

4.2 Hardware Requirements 15

4.3 Functional Requirements 16

4.4 Non Functional Requirements 17

4.5 Libraries, API's &Packages 18

5 Implementation 23

5.1 Workflow for the Train delay prediction 24

5.1.1 Data Collection 24

5.1.2 Preprocessing Data 25

5.1.3 Feature Extraction 25

5.1.4 Delay Detection 27

5.1.5 Performance Evaluation 28

5.1.6 Data Visualization 29

5.2 Flowchart of Multilingual Web Page 31

5.3 Working of the Multilingual Webpage 35

5.4 Results 39

6 Conclusion 44

7 Future Work 45

8 Bibliography 46

List of Figures:

Figure 3.1 Existing System Diagram 9

Figure 5.1 Workflow for the Train Delay Prediction 23

Figure 5.2 Data Visualization picture 30

Figure 5.3 Flowchart of the Webpage 31

Figure 5.4 Working of the Multilingual Webpage 36

Figure 5.5 User Interface for the Train Delay Prediction 40

Figure 5.6 Output for the Train Delay Prediction 40

Figure 5.7 Histogram of Delay Status 41

Figure 5.8 User Interface Output in English 42

Figure 5.9 Output in French 42

1 Introduction

Railway transport is integral to economic development, connectivity, and sustainability worldwide. It serves as a critical artery for the movement of goods and people, facilitating commerce and trade between production centres, distribution hubs, and markets. With its cost-effectiveness and large carrying capacity, railways excel in transporting bulk commodities over long distances, contributing to economic growth, job creation, and regional development. Additionally, railways enhance connectivity and accessibility, particularly in areas with limited road infrastructure, connecting remote communities to urban centres and essential services.

Moreover, railway transport offers environmental benefits by reducing greenhouse gas emissions and mitigating the environmental impact of transportation.. Hence Indian Railways is the lifeline of the nation, connecting people, cultures, and economies across vast distances, epitomizing the indispensable role of transportation infrastructure in fostering national unity and progress.

1.1 Train Delays

A delay refers to a situation where an event, task, or process does not occur or complete within the expected or planned timeframe. In the context of transportation, such as with trains, a delay occurs when a train does not arrive or depart at its scheduled time. Train delays are disruptions to scheduled train services, causing departures or arrivals to occur later than planned. These delays can stem from a variety of factors, each contributing to the complexity of managing railway operations. Common causes include infrastructure issues such as track maintenance or signal failures, technical malfunctions with trains or equipment, adverse weather conditions like snow, rain, or extreme temperatures, operational disruptions such as strikes or protests, and incidents like accidents or trespassing on railway tracks.

1.2 Importance of Train Delay Prediction

Train delay prediction is crucial for enhancing the efficiency and reliability of railway operations, benefiting both passengers and operators. Accurate prediction models empower passengers by providing valuable information to plan their journeys effectively and minimize the impact of potential delays. With advanced notice of potential delays, passengers can adjust their travel plans, choose alternative routes, or make arrangements to avoid inconvenience, ultimately enhancing their overall travel experience and satisfaction.

For railway operators, the ability to predict train delays enables proactive management of resources and operations. By anticipating potential disruptions, operators can implement preventive maintenance measures, optimize scheduling, and allocate resources more efficiently to mitigate the impact of delays. enhancing the reputation and competitiveness of the railway company.

1.3 Statement and Solution to the Problem

1.3.1 Problem Statement

Traditional methods used for Train Delay Prediction can be prone to errors and are no longer sufficient to deal with the growing complexity of Train Delay data. Although machine learning techniques can be leveraged for Train Delay prediction, they may not always produce the desired level of accuracy when utilized in Real Life, Because of the lack of insufficient data about the train delays.

So the problem statement typically involves formulating a predictive model that can accurately anticipate delays, enabling operators to take proactive measures to mitigate disruptions and improve overall efficiency and passenger satisfaction. The challenge lies in identifying relevant features, selecting appropriate Techniques, and optimizing the model to achieve reliable predictions in real-time scenarios.

1.3.2 Solution to the Problem

To address this, a robust Train Delay prediction system that leverages the strengths of multiple machine learning Techniques is needed. This system should provide a user-friendly interface for Passengers and Railway Operator to input test results and receive accurate Train Delay predictions. By combining the power of Machine learning with interactive web interface design, this project aims to improve Passenger Experience.

1.4 Scope of the Project

In addressing the problem of train delay prediction, leveraging cosine similarity and TF-IDF techniques presents a promising solution. By utilizing TF-IDF (Term Frequency-Inverse Document Frequency), we can effectively weigh the importance of terms within textual data. This allows us to extract relevant features and quantify their significance in predicting delays. Additionally, cosine similarity enables us to measure the similarity between the textual features, providing insights into potential delay patterns. By employing these techniques, we can build a predictive model that identifies similarities between current conditions and past instances of delays, enabling timely interventions and proactive management of train schedules to minimize disruptions and improve overall service reliability.

1.5 Machine Learning

The machine learning domain encompasses a diverse set of techniques, algorithms, and methodologies that enable computers to learn from data and make predictions or decisions without explicit programming. It draws from various fields such as statistics, mathematics, computer science, and artificial intelligence to develop models that can analyze complex datasets, identify patterns, and make informed decisions or predictions.

Machine learning algorithms can be categorized into supervised learning, unsupervised learning, and reinforcement learning, each serving different purposes depending on the task at hand. Supervised learning involves training models on labeled data to make predictions, while unsupervised learning discovers patterns and structures within unlabeled data. Reinforcement learning focuses on learning optimal behaviors through trial and error in a dynamic environment.

The machine learning domain has seen rapid growth and innovation in recent years, with applications spanning diverse industries such as healthcare, finance, transportation, and beyond. Its ability to extract insights from data and automate decision-making processes makes it a powerful tool for addressing complex problems and driving innovation in various domains.

1.5.1 Why Choosed Machine Learning ?

Machine learning is used in train delay prediction because of its ability to analyze large volumes of complex data, identify patterns, and make predictions based on historical and real-time information. Train delay prediction involves processing a wide range of factors such as weather conditions, track maintenance schedules, historical delay patterns, and operational data. Traditional statistical methods may struggle to handle the complexity and variability of these factors effectively.

Machine learning algorithms, on the other hand, excel at handling large and diverse datasets, allowing them to capture intricate relationships between various factors and predict outcomes accurately. By training machine learning models on historical data, they can learn patterns and trends associated with train delays, enabling them to make predictions about future delays based on current conditions.

Moreover, machine learning models can adapt and improve over time as they receive new data, making them well-suited for dynamic environments like railway systems where conditions and factors influencing delays may change frequently.

Overall, machine learning provides a powerful framework for train delay prediction by leveraging data-driven insights to optimize scheduling, resource allocation, and decision-making, ultimately improving the efficiency and reliability of railway services.

2 Literature Survey

Passenger train delays have become a pressing issue globally, affecting commuter satisfaction and transportation efficiency. The INDIA and various other countries have grappled with persistent delays due to increased demand and near-maximum capacity on rail networks. The cascading effect of delays on subsequent trains underscores the urgency for accurate and reliable delay prediction methods to facilitate proactive decision-making by train controllers.

[1] A novel approach proposed in one study involves building heterogeneous ensembles, leveraging two innovative model selection methods based on accuracy and diversity. Real-world testing revealed these ensembles to be more accurate and robust than single models and existing homogeneous ensembles like Random Forest and XG Boost . Moreover, their consistent performance across different operating companies highlights their potential for widespread application .

[2] Another study focuses on the impact of train delays on traveler's mode choice, emphasizing the importance of real-time passenger train delay prediction (PTDP) models. By employing machine learning techniques and exploring the influence of external variables such as historical delay profiles, ridership, geography, and weather data, researchers developed PTDP models with superior performance. Notably, models combining multi-layer perceptron (MLP) with Real-time with Historical-based Data-frame Structure (RWH-DFS) demonstrated the highest accuracy, outperforming other approaches. This underscores the significance of incorporating both real-time and

historical data for more precise delay predictions, crucial for enhancing transportation effectiveness and passenger satisfaction.

[3] In the context of the Indian railway network, overcrowding and speed restrictions contribute significantly to train delays. To address this issue, a study utilized real-world data to implement machine models for classification. Among these models, Random Forest learning emerged with the highest prediction accuracy. This underscores the efficacy of machine learning in analyzing and mitigating delays in complex railway systems, such as those in densely populated regions like India.

[4] Train delay analysis and forecasting are imperative for maintaining on-time performance and passenger trust. Leveraging Support Vector Machines (SVM) and Principal Component Analysis (PCA), one study aimed to forecast delay performance, enabling traveller's to adjust their schedules accordingly. By reducing the dimensionality of datasets and employing advanced statistical techniques, this approach enhances the accuracy of delay predictions, empowering passengers to make informed decisions about their travel plans.

[5] In conclusion, the reviewed literature underscores the critical role of accurate and reliable passenger train delay prediction models in addressing the challenges faced by railway systems worldwide. By integrating machine learning techniques with real-time and historical data, researchers have made significant strides in improving prediction accuracy and mitigating the impact of delays on commuter satisfaction and transportation efficiency.

3 System Design

The system design for the train delay prediction project encompasses a comprehensive architecture integrating historical data analysis, real-time information feeds, and predictive modeling algorithms. It involves data collection from various sources. Machine learning models, including regression and classification algorithms, are employed to forecast potential delays based on historical patterns and current contextual factors. Concurrently, the multilingual query answering system is designed with natural language processing (NLP) techniques to facilitate user interaction in multiple languages. It employs a combination of language translation algorithms, intent recognition, and knowledge graph-based query processing to understand and respond to user queries accurately across diverse linguistic contexts. Both systems aim to enhance the efficiency and accessibility of railway services by providing information and support to users and operators.

3.1 Existing System

A substantial amount of research has been conducted on the forecasting the train delays by collecting large amount of data of the trains over years and processing that data and applying various Machine Learning Algorithms like Random Forest ,Gradient Boosting Algorithms to predict whether the train is delayed or not.

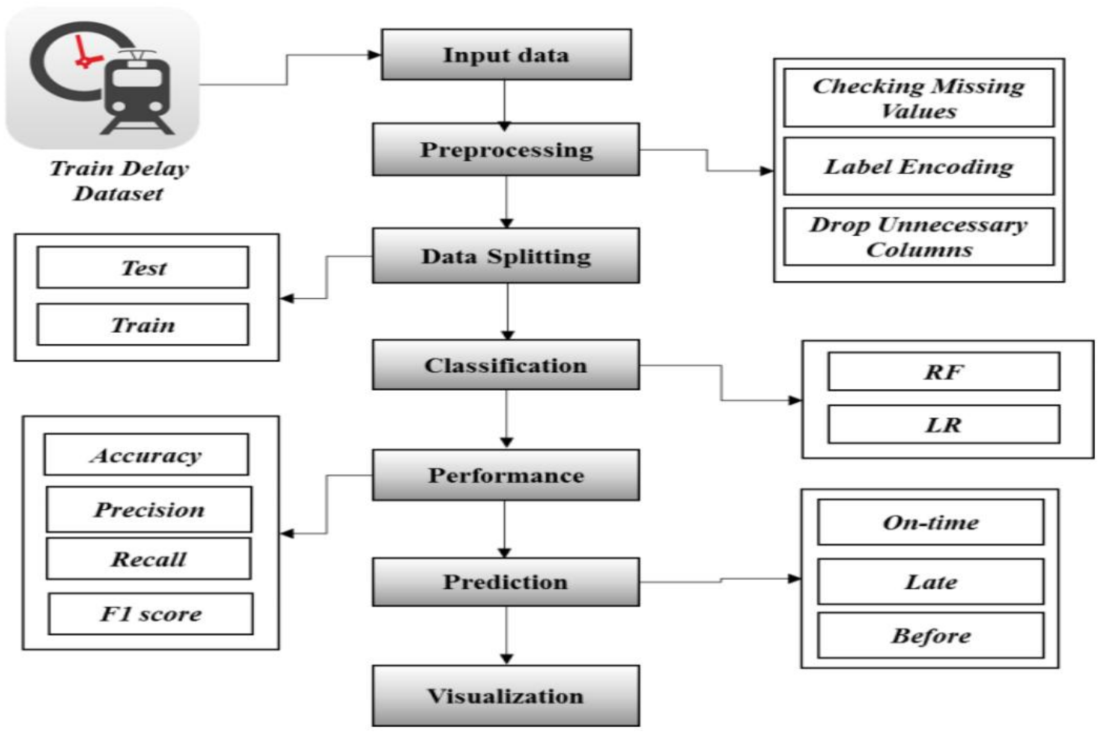


Figure 3.1 Existing System Diagram

3.1.1 Random Forest Algorithm

Random Forest is an ensemble learning method used for classification and regression tasks. It operates by constructing multiple decision trees during training and outputs the mode (for classification) or average (for regression) of the individual predictions made by each tree. Each decision tree is trained on a random subset of the training data and a random subset of features, ensuring diversity among the trees. During prediction, the ensemble of trees aggregates their individual predictions to make a final prediction,.

3.1.2 Gradient Boosting Algorithm

Gradient Boosting is another ensemble learning technique used for regression and classification tasks. Unlike Random Forest, which builds multiple trees independently, Gradient Boosting builds trees sequentially in a greedy manner. Each new tree in the sequence corrects the errors made by the previous ones, with subsequent trees focusing more on the instances that were misclassified by earlier trees. This iterative process minimizes the overall prediction error, resulting in a strong predictive model. Gradient Boosting is highly effective in producing accurate predictions, especially when combined with shallow decision trees (boosting with decision trees is often referred to as Gradient Boosted Trees or GBMs).

3.2 Proposed System

The objective of our proposed system is to build a model that will predict whether the train is delayed or not more accurately using the Machine Learning Techniques like Cosine Similarity and TFIDF on real time data and also in addition to that build a passenger enquiry website using Django and large language model llama-2-7B.

3.2.1 Cosine Similarity

Cosine similarity is a measure used to determine the similarity between two vectors in a multi-dimensional space. It calculates the cosine of the angle between the vectors, reflecting their orientation rather than their magnitude. The range of cosine similarity is between -1 and 1, with 1 indicating perfect similarity, 0 indicating no similarity, and -1 indicating perfect dissimilarity. Cosine similarity is widely used in various applications such as information retrieval, document similarity analysis, and recommendation systems. Additionally, cosine similarity is computationally efficient and suitable for high-dimensional data, making it a popular choice in many machine learning tasks.

3.2.2 TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) is a numerical statistic used in natural language processing and information retrieval to reflect the importance of a term in a document relative to a collection of documents. It consists of two components: TF, which measures the frequency of a term in a document, and IDF, which measures the rarity of the term across the entire document collection. TF-IDF assigns higher weights to terms that are frequent within a document but rare across the entire collection, effectively filtering out common terms and highlighting unique ones. The formula for TF-IDF involves multiplying the TF and IDF scores for each term. TF-IDF is widely used in tasks such as text classification, clustering, and search engine ranking, providing a robust method for extracting meaningful features from text data.

3.2.3 Django

Django is a high-level Python web framework that promotes rapid development and clean, pragmatic design. It follows the MVC (Model-View-Controller) pattern, emphasizing the reusability of components and the "don't repeat yourself" (DRY) principle. Django comes bundled with a built-in admin interface, authentication system, and ORM (Object-Relational Mapping) for database interaction. It prioritizes security, offering protection against common web vulnerabilities. Its templating engine allows for the creation of dynamic web pages with minimal code. It supports various databases including PostgreSQL, MySQL, and SQLite, providing flexibility in deployment. Django REST Framework extends Django's capabilities to build APIs quickly and efficiently. Overall, Django empowers developers to build robust, scalable web applications with ease.

3.2.4 LLAMA -2-7B

Llama 2 is an open-source large language model (LLM) developed and released by Meta AI in July 2023. The model has been trained on over 2 trillion public data tokens (chunks of information). It uses this data to understand word meanings and intent, and then generates output accordingly .While Llama 2 shares some similarities with OpenAI's ChatGPT model, it is designed for other use cases. Another news is that the Llama 2 language model is not trained with data from Meta products and services. The developer team avoided using data containing personal information while building the Llama 2 language model. Our knowledge on this subject is limited, as Meta AI does not specify the source of the data it uses for the train.

4 Requirement Analysis

Requirement analysis is a crucial phase in any project, encompassing the systematic examination. It involves gathering and prioritizing requirements through techniques

4.1 Software Requirements

Software requirements for a project can vary depending on its specific needs and technologies used. However, here are some common software components required for this project.

1) Jupyter notebook:Jupyter notebook is an interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality.The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.
2) Windows 11 OS:Windows 11, released in October 2021, offers a refreshed user interface with centered taskbar icons and rounded corners, providing a modern look and feel. It introduces Snap Layouts and Snap Groups for improved multitasking, along with virtual desktops. Gaming enhancements include DirectStorage for faster loading times and Auto HDR for improved visuals. Security features like Windows Hello and Secure Boot ensure enhanced

protection against threats. Compatibility with Android apps through the Microsoft Store expands the ecosystem for users.

3) Python 3.10: Python 3.10, released in October 2021, introduces several new features. It includes pattern matching, allowing concise conditional expression syntax. It introduces a new syntax for specifying types, simplifying annotations. Structural pattern matching allows for more expressive and readable code. Improved error messages enhance debugging experiences. Python 3.10 also brings performance enhancements and security improvements.
4) Visual Studio Code: Visual Studio is a comprehensive integrated development environment (IDE) developed by Microsoft, catering to the diverse needs of software developers across various platforms and programming languages. Boasting a rich set of features, including code editing, debugging, and project management tools, Visual Studio offers a seamless development experience for individuals and teams alike. Its intuitive user interface, coupled with robust debugging capabilities and extensive language support, empowers developers to create a wide range of applications, from desktop and web to mobile and cloud-based solutions. With integrated collaboration features and support for popular version control systems, Visual Studio facilitates efficient teamwork and code sharing, enabling developers to bring their ideas to life quickly and effectively.
5) Google Colaboratory : Google Colaboratory is a cloud-based platform provided by Google that enables users to write, execute, and share Python code in a collaborative environment. It offers free access to computing resources, including GPU and TPU accelerators, making it particularly attractive for tasks such as

machine learning and data analysis. Users can create and run Jupyter notebooks directly in their web browser, leveraging Google's infrastructure without the need for any local setup. Colab also integrates seamlessly with Google Drive, allowing users to save and share their notebooks effortlessly. With its combination of ease of use, powerful computing resources, and collaboration features, Google Colab has become a popular choice for developers, researchers, and educators seeking a flexible and accessible platform for Python programming and experimentation.

4.2 Hardware Requirements

The hardware requirements for a project depend on its specific needs, such as the type of tasks it performs, the scale of data it handles, and the software tools it utilizes. Here are some general Hardware

1) Memory (RAM) : At least 8 GB of RAM is recommended, but for larger datasets and more complex models, 16 GB or more may be beneficial to handle the computational load.
2) Storage: Sufficient storage space is necessary to store datasets, model files, and any other project-related files. An SSD (Solid State Drive) is preferable for faster data access.
3) Intel i5 processor: The Intel Core i5 processor is a popular mid-range processor known for its balance of performance and affordability. It is part of Intel's Core series, which is designed for a wide range of computing tasks, including general productivity, gaming, multimedia, and light content creation.

4.3 Functional Requirements

Functional Requirements are a requirement that refers to the data of train delay prediction based on the parameters such as arrival time, departure time , actual arrival time and actual departure time etc. In this, we input real time data of trains which contains parameters such as arrival time, departure time ,actual arrival time and actual departure time, Latitude and Longitude etc.

Data Collection: The system should be able to collect data from various sources such as historical records, real-time sensor data, weather reports, maintenance logs, and crowd-sourced information.

Preprocessing: It should preprocess the collected data, which involves cleaning, transforming, and aggregating the data to make it suitable for analysis and modeling.

Feature Extraction: The system should extract relevant features from the preprocessed data, including but not limited to arrival and departure times, weather conditions, maintenance schedules, and historical delay patterns.

Model Training: It should train machine learning models using historical data to learn patterns and relationships between features and train delays.

Prediction: The system should be able to use the trained models to predict the likelihood of train delays based on current conditions and input parameters.

Evaluation: It should evaluate the performance of the prediction models using appropriate metrics such as accuracy, precision, recall, and F1-score.

4.4 Non Functional Requirements

The non-functional requirement of the proposed system deals with how well the system provides service to the user.

1) Easy to use: Anyone can use it by providing proper information because the model is easy and simple to use.
2) Fast and reliable: The time needed to access the data by administrators is much less than the existing system.
3) Maintainability: After deploying the proposed system we will ensure that the model continues to work properly by checking it regularly and making repairs and adjustments if required.
4) Usability: Also the model should be easy to use by all.
5) Robustness: when users enter incorrect format it should be tolerant of errors and produce error reports.
6) Natural Language Understanding: Utilizes advanced language processing to understand user queries conversationally.
7) Multilingual Support: Supports various languages to cater to diverse passenger needs.
8) Real-time Information: Provides up-to-date information on train schedules, delays, and platform changes.
9) Interactive Interface: Engages users in a chat-like interface for a seamless and user-friendly experience.

10) Performance: A performance attribute type of non-functional requirement measures system performance. In our model we are using the performance evaluation metrics like accuracy, precision, recall and f1score.

4.5 Libraries, API's &Packages

Python offers a rich ecosystem of libraries and frameworks and packages covering various domains such as data science, machine learning, web development, and more. Some Python libraries, packages and API's that we used in our project are:-

1) NumPy : NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors

2) Pandas: pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase

"Python data analysis" itself. Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010.

3) Matplotlib: Matplotlib is easy to use and an amazing visualizing library in Python. It is built on NumPy arrays and designed to work with the broader SciPy stack and consists of several plots like line, bar, scatter, histogram, etc.
4) Scikit-learn :scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. This software library features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
5) Seaborn : Seaborn is a powerful Python library for statistical data visualization built on top of Matplotlib. It provides a high-level interface for creating visually appealing and informative plots, making it an essential tool for data analysts, scientists, and researchers. Seaborn is particularly renowned for its ability to generate complex statistical plots with just a few lines of code, enabling users to explore and visualize their data effectively.

   One of Seaborn's key strengths is its extensive collection of built-in themes and color palettes, which allow users to customize the appearance of their plots easily. Whether it's adjusting the color scheme, font size, or overall style, Seaborn offers a wide range of options to tailor visualizations to specific preferences or requirements.

6) RapidAPI : RapidAPI is a platform that simplifies the process of accessing and integrating with a vast array of APIs (Application Programming Interfaces). APIs are tools that allow different software applications to communicate and interact with each other, enabling developers to access specific functionalities or data from third-party services. Rapid API acts as a centralized marketplace for APIs, offering developers a convenient way to discover, subscribe to, and manage a wide range of APIs from various providers. The platform hosts thousands of APIs across multiple categories, including data, finance, social media, machine learning, and more. Overall, Rapid API simplifies the process of integrating third-party APIs into software applications, enabling developers to access a wealth of functionalities and data with ease.

7) Langchain: Langchain seems to be a custom package designed for language-related tasks. It likely offers features for processing, analyzing, and manipulating text data within Python applications .Langchain could potentially provide utilities for tasks such as tokenization, stemming, part-of-speech tagging, and more, aiming to streamline the handling of text data in various contexts.

8) Ctransformers : Ctrnasformers is a Python library that appears to focus on transformers and neural network-based language models. It likely provides tools for working with pre-trained transformer models, fine-tuning them, or integrating them into applications. With its emphasis on transformers, ctransformers may enable users to leverage state-of-the-art natural language processing capabilities,

such as text generation, sentiment analysis, and language understanding, within their Python projects.

9) sentence-transformers: Sentence-transformers is a library that specializes in sentence embedding techniques, allowing users to encode sentences into dense vectors that capture their semantic meaning. By utilizing advanced embedding models, sentence-transformers facilitates tasks such as semantic similarity calculation, clustering, and information retrieval. It likely offers a range of pre-trained models and utilities for embedding sentences efficiently and effectively.

10) faiss-cpu**:** Faiss-cpu is a Python library that provides efficient similarity search and clustering algorithms for large-scale datasets. It is likely an optimized version of Faiss (Facebook AI Similarity Search) designed to run on CPUs, offering fast and memory-efficient algorithms for similarity search tasks. Faiss-cpu is particularly useful for applications requiring nearest neighbor search, such as recommendation systems, information retrieval, and data mining.

11) Llama- cpp-python**:** Llama- cpp-python appears to be a package that bridges the gap between C++ and Python, potentially providing utilities or wrappers for integrating C++ code into Python applications or vice versa. This integration may enable developers to leverage existing C++ libraries or functionalities within their Python projects, enhancing performance or accessing specific features not available in pure Python environments.

12) Googletrans : Googletrans is a Python library that serves as a client for Google Translate API, enabling translation of text between different languages. With

googletrans, users can easily integrate translation capabilities into their Python applications, facilitating multilingual text processing and communication.

13) Django**:** Django is a high-level Python web framework renowned for its efficiency in enabling developers to construct web applications swiftly and effectively. It follows the "batteries-included" philosophy, providing a comprehensive set of tools and libraries that simplify common web development tasks, such as URL routing, database management, and user authentication. Django empowers developers to build secure, scalable, and feature-rich web applications with minimal effort.

14) openai-whisper: OpenAI Whisper is a package that seems related to OpenAI's Whisper project.It involve tools or utilities for working with Whisper, which is a protocol for decentralized and privacy-preserving messaging. OpenAI Whisper could potentially offer features for secure communication, privacy protection, or decentralized network management within Python applications.

15) gTTS: gTTS (Google Text-to-Speech) is a Python library that enables users to convert text into speech using Google's text-to-speech API. It provides a simple interface for synthesizing speech from text strings, with options for language selection and audio format customization. With gTTS, developers can easily integrate text-to-speech functionality into their Python applications, enabling speech synthesis for various use cases such as accessibility, audio content generation, and interactive applications.

5 Implementation

This section may cover various aspects such as the execution of tasks, allocation of resources, timeline management, collaboration among team members, and any challenges encountered during the implementation process. Additionally, it may include discussions on how specific methodologies, techniques, or technologies are utilized to achieve project objectives. Overall, the Implementation section serves as a roadmap for turning plans and strategies into tangible outcomes, providing insights into the practical execution of the project.

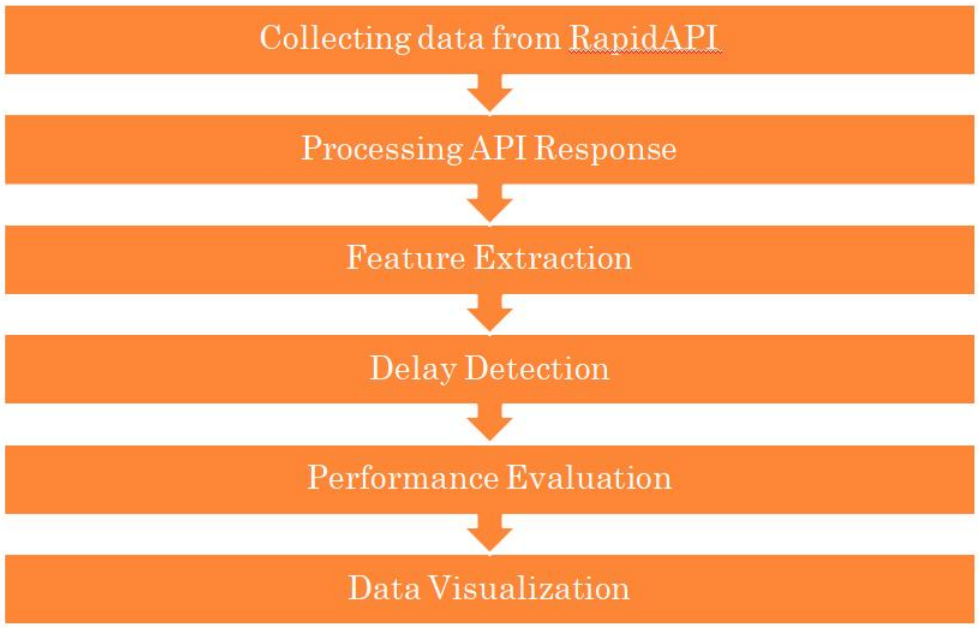


Figure 5.1 Workflow for the Train Delay Prediction

5.1 Workflow for the Train delay prediction

5.1.1 Data Collection

Collecting data from RapidAPI involves several steps to ensure efficient retrieval and integration of desired information. Initially, you would need to register an account on the RapidAPI platform and subscribe to relevant APIs that provide access to the data you require.In our case we have to Subscribe to the IRCTC API of the RapidAPI which provides the data about the Indian trains data. There are different plans for subscribing we can pay also and as well as we can use it for free also.Once subscribed, you can obtain an API key, which serves as authentication for accessing the APIs. Next, you would need to identify the specific endpoints or methods within the APIs that offer the data you need. This may involve reading documentation provided by RapidAPI and understanding the parameters and endpoints available for retrieving the desired data. After determining the appropriate endpoints, you can begin making HTTP requests to the API using your API key, specifying any required parameters such as location, date range, or data format. Upon receiving responses from the API, you'll need to parse and process the data according to your project's requirements.

Depending on the complexity of the data and your project's needs, this may involve data cleaning, transformation, or aggregation. Finally, you can store the collected data in a suitable format for further analysis or integration into your application or system. Throughout this process, it's essential to adhere to the terms of use and rate limits imposed by RapidAPI to ensure compliance and avoid disruptions in data access.The data we will get will be real time data. The schedule will be updated by the API ,This will reflect in our Project only when we would refresh the whole page .So to know the train delay status we have to refresh the page for efficient results.

5.1.2 Preprocessing Data

The processing phase in a train delay prediction project is pivotal for ensuring the quality and suitability of the data for subsequent analysis and modeling. This phase involves several key steps, beginning with data collection from various sources such as historical records, weather databases, and maintenance logs. Once collected, the data undergoes cleaning to address inconsistencies, errors, and missing values, ensuring data integrity. Feature engineering techniques are then applied to extract relevant information and create new features that may better capture patterns related to train delays. Additionally, data normalization or scaling may be performed to standardize feature scales, preventing certain features from dominating the model. The dataset is often split into training, validation, and testing sets to accurately evaluate model performance. Finally, data encoding is applied to convert categorical variables into a format suitable for machine learning algorithms. Overall, the preprocessing phase sets the foundation for robust and accurate train delay predictions by optimizing the quality and structure of the input data.

5.1.3 Feature Extraction

Feature extraction is a crucial step in the preprocessing phase of a train delay prediction project, involving the identification and extraction of relevant information from raw data to create informative features for modeling. In the context of train delay prediction, feature extraction typically involves deriving meaningful variables from diverse data sources such as historical records, weather reports, maintenance logs, and IoT sensor data.

These features may include temporal variables such as time of day, day of week, and seasonality, as well as contextual factors such as weather conditions, track maintenance schedules, and historical delay patterns. Feature extraction techniques may also involve transforming variables, creating lag features to incorporate historical information, or aggregating data over specific time intervals to capture trends and patterns. The goal of feature extraction is to represent the underlying structure of the data in a format that is conducive to training predictive models, enabling the identification of key factors contributing to train delays and facilitating accurate predictions. By carefully selecting and engineering informative features, feature extraction enhances the predictive power of the models and improves their ability to capture complex relationships within the data.

In train delay prediction, capturing data about arrival time, departure time, actual arrival time, and actual departure time is essential for accurately assessing delays and predicting potential disruptions. Arrival and departure times provide baseline schedules against which actual arrival and departure times are compared, enabling the calculation of delays. Actual arrival and departure times represent the observed times at which trains arrive at and depart from stations, respectively, allowing for real-time tracking of train movements.

By comparing scheduled and actual times, delays can be calculated and categorized based on their magnitude and impact on train operations. These data points also enable the identification of patterns and trends in delays, such as peak hours or frequent causes of disruptions.

In conclusion, the feature extraction phase in a train delay prediction project is instrumental in transforming raw data into informative features that drive the predictive modeling process. By extracting relevant information from diverse data sources such as historical records, weather reports, maintenance logs, and IoT sensor data, this phase lays the foundation for accurate and effective predictions.

5.1.4 Delay Detection

In a train delay prediction project, the application of cosine similarity and TF-IDF techniques plays a crucial role in harnessing the predictive power of textual data sources such as weather reports, maintenance logs, and historical records.

Cosine similarity serves as a valuable metric for quantifying the similarity between textual features extracted from various sources. By representing textual data as vectors in a high-dimensional space, cosine similarity enables the comparison of historical delay reports or maintenance logs with real-time data, such as current weather conditions or maintenance schedules. This comparison allows the project to identify similarities or patterns between historical occurrences of delays and present circumstances. For instance, if the textual features of historical delay reports closely match the current weather conditions or maintenance activities, it suggests a higher likelihood of similar delays occurring. Cosine similarity thus facilitates the identification of relevant textual features that may contribute to predicting train delays.

TF-IDF, on the other hand, is instrumental in preprocessing textual data to extract informative features for modeling. TF-IDF evaluates the importance of terms within a document relative to a larger corpus by weighing terms based on their frequency and rarity. In the context of train delay prediction, TF-IDF can be applied to prioritize relevant keywords or terms associated with delays. By weighting terms based on their frequency in a document and their rarity across the entire corpus, TF-IDF helps to identify and extract key textual indicators of delays, such as weather-related terms or maintenance-related issues. These TF-IDF-weighted features serve as valuable inputs for predictive models, enabling them to focus on the most influential textual factors contributing to delays.

Overall, the application of cosine similarity and TF-IDF techniques empowers a train delay prediction project to leverage textual data effectively, enhance the accuracy of predictions, and identify meaningful patterns indicative of potential delays. By integrating these techniques into the predictive modeling pipeline, the project can extract valuable insights from textual sources and improve the overall performance of the delay prediction system.

5.1.5 Performance Evaluation

Certainly Accuracy, Precision, recall, and F1 score are commonly used performance evaluation metrics . Let's delve into each metric in detail.

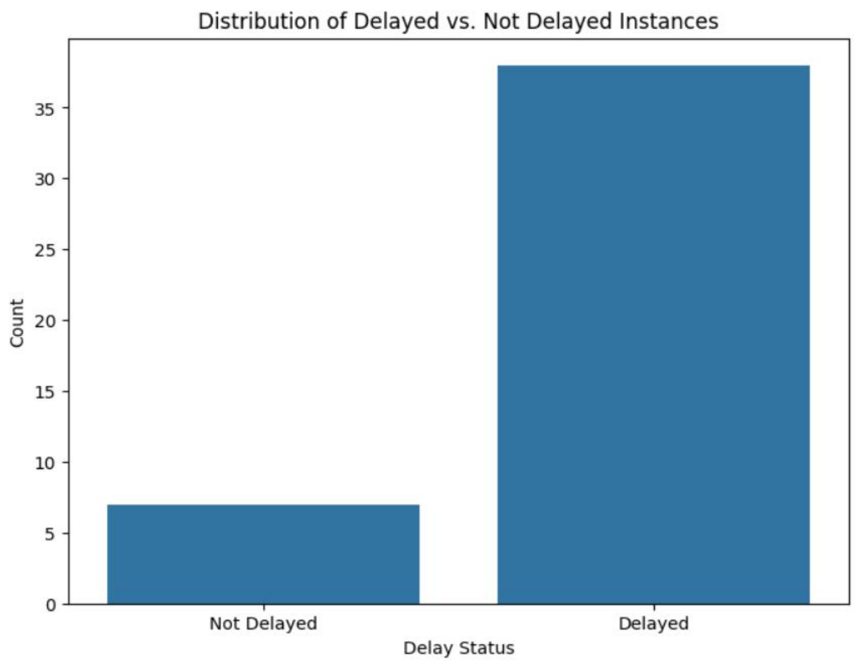
1) Accuracy: Accuracy is a common performance metric used to evaluate the overall effectiveness of a classification model. It measures the proportion of correctly classified instances out of the total instances in the dataset. Accuracy provides an overall measure of how well the model performs across all classes. However, it may not be the most suitable metric for imbalanced datasets where one class

dominates the others. In such cases, a high accuracy score can be misleading if the model predominantly predicts the majority class

2) Precision: Precision measures the accuracy of positive predictions made by a model. It calculates the ratio of true positive predictions to all positive predictions made by the model. Precision focuses on the correctness of positive predictions, indicating how many of the predicted positive cases are actually positive.
3) Recall: Recall, also known as sensitivity or true positive rate, measures the ability of the model to correctly identify positive instances out of all actual positive instances. Recall emphasizes the model's ability to capture all positive instances, indicating how many of the actual positive cases are correctly identified by the mode
4) The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall, offering a single metric to evaluate the model's overall performance. The F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0. It considers both false positives and false negatives, making it suitable for imbalanced datasets where the number of negative instances outweighs the positive instances and also helps in the evaluating the predictions made.

5.1.6 Data Visualization

The code includes visualization using `matplotlib` and `seaborn` to illustrate the distribution of delay statuses (`ground_truth_delays`) and the distribution of cosine similarity scores (`cos_sim`). Visualization aids in understanding the characteristics of the data and



the.

Figure 5.2 Data Visualization picture

performance of the delay detection algorithm In summary, the provided code demonstrates a comprehensive pipeline for retrieving, processing, analyzing, and evaluating live train status data using an API. It leverages text data processing techniques (TF-IDF vectorization) and similarity metrics (cosine similarity) to detect delays between train stations, followed by performance evaluation using machine learning metrics and visualization for result interpretation. Each step in the implementation contributes to a systematic approach towards understanding and analyzing train data for delay prediction and evaluation

5.2 Flowchart of Multilingual Web Page

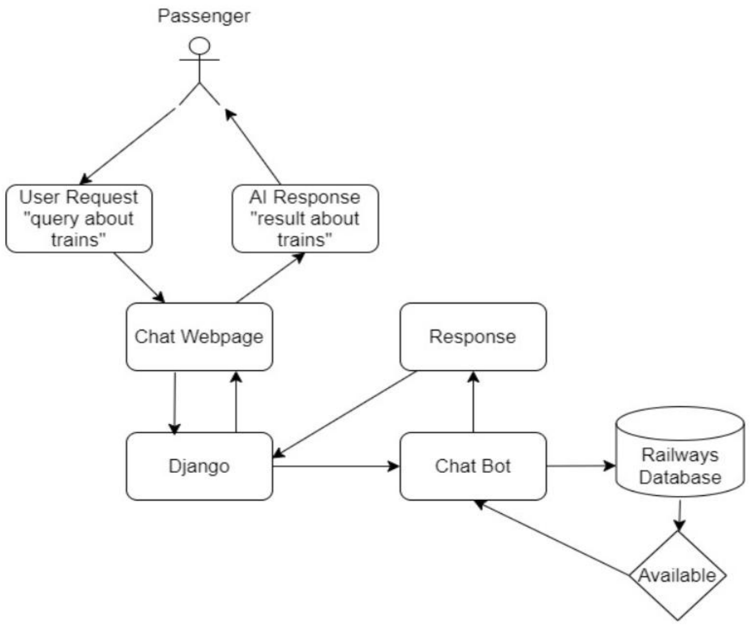


Figure 5.3 Flowchart of the Webpage

Explanation about the each block in the flowchart of the webpage

Passenger

Function: This block represents the user initiating a conversation with the chatbot. It signifies the user's side of the interaction.

Details:

The user interacts with the chat interface (Chat Webpage) to ask a question or provide input related to trains.

The type of input can vary depending on the chatbot's design. It could be a natural language question, a selection from a menu, or even voice input (if supported).

Once the user submits their input, the chat interface transmits it to the chatbot software (Chat Bot) to be processed.

User Request

Function: This block represents the specific information the user is seeking from the chatbot. It captures the essence of the user's question or request.

Details:

This block contains the actual content of the user's query about trains.

Examples might include: "What time is the next train to Hyderabad?" or "Can I book a ticket for the 8 am train?".

The User Request block essentially summarizes the user's goal for the interaction with the chatbot.

AI Response

Function: This block represents the chatbot's response to the user's query. It captures the information retrieved or the action taken based on the User Request.

Details:

This block contains the information or action generated by the chatbot in response to the user's query.

Examples might include: "The next train to Hyderabad departs at 10:30 am" or "Here is a link to book your ticket for the 8 am train."

The AI Response block reflects the chatbot's attempt to fulfill the user's request as derived from the User Request block.

Chat Webpage

Function: This block represents the user interface where the conversation with the chatbot takes place. It acts as the communication channel between user and chatbot.

Details:

This block signifies the chat window or interface where the user sees prompts, types their questions, and receives responses from the chatbot.

It acts as a visual representation of the conversation, allowing the user to interact with the chatbot in a user-friendly way.

The Chat Webpage might display additional information or buttons depending on the specific functionalities of the chatbot.

Django

Function: This block indicates that the chatbot is likely built using the Django web framework.

Details: Django is a high-level Python web framework that simplifies the development of web applications.

Its presence suggests that the creators used Django to structure the chatbot's functionalities within the web interface (Chat Webpage).

Chat Bot

Function: This block represents the software program responsible for understanding user requests, processing information, and generating responses.

Details:

This block signifies the core logic and functionalities of the chatbot system.

The Chat Bot receives user input from the Chat Webpage, interprets the User Request, retrieves relevant information from the Railways Database (if needed), and generates an AI Response to be displayed on the Chat Webpage.

It acts as the brain behind the chatbot, handling conversation flow, information retrieval, and response generation.

Railways Database

Function: This block represents the storage system containing information about trains. It acts as the source of data for the chatbot's responses.

Details:

This block signifies the database that holds information such as train schedules, routes, fares, and possibly additional details depending on the system's capabilities.

The Chat Bot might access the Railways Database to retrieve relevant information in order to respond to user queries about trains.

The database's availability (indicated by "Available") is crucial for the chatbot to provide accurate and up-to-date information.

5.3 Working of the Multilingual Webpage

1) Data Acquisition and Ingestion

The initial step of the system involves gathering data from diverse sources to build a comprehensive dataset for analysis. This process encompasses multiple approaches, including uploading CSV files containing text documents such as product descriptions, customer reviews, news articles, or any other text-based information relevant to the project's objectives. Additionally, the system integrates with databases or APIs that store textual content, facilitating access to structured data repositories. Moreover, to enrich the dataset further, the system employs web crawling and scraping techniques to extract text data from websites, adhering to ethical guidelines to ensure responsible data acquisition. By leveraging these methods, the system aggregates a wide range of textual sources, enabling comprehensive analysis and modeling to derive meaningful insights and make informed decisions.

2. Text Preprocessing :

Preprocessing steps are essential for ensuring the quality and consistency of textual data before analysis. The first step, cleaning, involves removing noise such as punctuation, special characters, and HTML tags to create cleaner representations of text. This process helps eliminate irrelevant information that may interfere with downstream analysis. Following cleaning, normalization techniques are applied to standardize the text by converting it to either lowercase or uppercase. This ensures consistency in the representation of words and reduces the complexity of the dataset.

Tokenization then breaks down the text into smaller units, such as words or n-grams (sequences of words), to facilitate analysis at a granular level.

Finally, stemming or lemmatization techniques are employed to reduce words to their root forms, enhancing semantic similarity detection. For instance, stemming would convert words like "running" and "runs" to their root form "run," while lemmatization would consider their base form as "run." These preprocessing steps collectively prepare the textual data for subsequent analysis, enabling more accurate and insightful results.

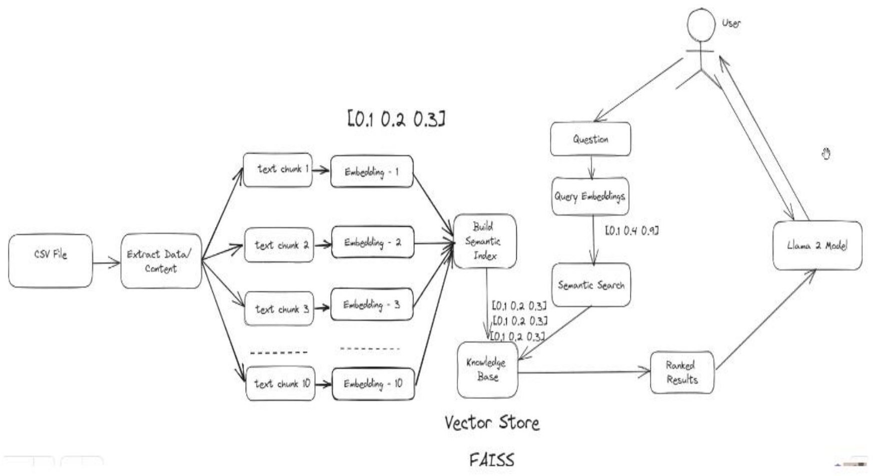


Figure 5.4 Working of the Multilingual Webpage

3. **Text Chunking:**

Text chunking is a preprocessing step commonly employed for handling very large documents efficiently. In scenarios where the dataset comprises extensive textual content, such as lengthy articles, books, or reports, processing the entire document at once can be computationally intensive and may not yield optimal results. Text chunking addresses this challenge by dividing the document into smaller, manageable sections or chunks. By breaking down the document into smaller units, text chunking improves the efficiency of subsequent tasks, such as vectorization and analysis. This process allows for more focused analysis of individual sections, enabling the system to capture semantic meaning accurately within each chunk. Additionally, text chunking facilitates parallel processing, enabling distributed systems to handle large volumes of text data more effectively. Overall, text chunking enhances the scalability and performance of text processing pipelines, particularly when dealing with extensive textual content, thereby improving the accuracy and efficiency of downstream tasks such as natural language processing and machine learning..

4. Text Embedding:

This is the heart of the system, where the magic happens! Each text chunk (or entire document) is transformed into a mathematical representation called a vector embedding. Popular embedding techniques include:

Word2Vec: Analyzes word co-occurrence patterns to learn word relationships and create vector representations.

GloVe: Similar to Word2Vec but incorporates global word information for richer semantic understanding.

BERT or other contextual models: Capture the meaning of words based on their context in a sentence, leading to more nuanced embeddings.

5. Semantic Indexing:

The generated vector embeddings are then stored and indexed using a specialized data structure called a semantic index. This index allows for fast and efficient searching based on similarity. Popular libraries like FAISS (Facebook AI Similarity Search) are often used for this purpose. The semantic index essentially maps similar vectors close together in a high-dimensional space.

6. User Search:

When a user submits a search query, it undergoes a similar process as the stored data:

Preprocessing (if applicable) to clean and normalize the query text.

Conversion into a vector embedding using the same embedding technique employed for the stored data.

7. Similarity Search:

The user's query vector is compared against the indexed vector embeddings in the store using the semantic index. This efficiently identifies data points with vector embeddings closest to the user's query in terms of semantic meaning. Unlike traditional keyword matching, the system understands the underlying intent and retrieves relevant information even if the exact keywords aren't present.

8. Ranked Results:

The system retrieves the data points associated with the most similar vector embeddings. These retrieved data points are presented to the user as ranked results, with the most relevant items at the top of the list. The ranking is based on the distance between the user's query vector and the retrieved vectors in the high-dimensional space. Smaller distances indicate greater semantic similarity.

9. Knowledge Base Integration :

Some vector store systems might incorporate a knowledge base. This external source of information can be used in various ways to enhance search results like :-

Query Expansion: Identifying synonyms or related concepts in the knowledge base to broaden the user's search and potentially retrieve more comprehensive results.

Entity Linking: Linking entities mentioned in the user's query (e.g., locations or people) to relevant information within the knowledge base for richer context.

In essence, the vector store system bridges the gap between human language queries and the underlying data by converting both into a common mathematical representation (vectors). This allows for efficient retrieval of information based on semantic similarity rather than exact keyword matching. It facilitates a more natural and intuitive search experience, understanding the user's intent beyond literal keywords.

5.4 Results

The figure 5.4 is the Home page of the User Interface which consists of two options in Menu which is a drop down list containing the Train ID Info and Live Status .

In Train ID Info we can get the information about the Train which includes the Train Number .

And in the Live Status we would be able to get the Train’s Live Status.

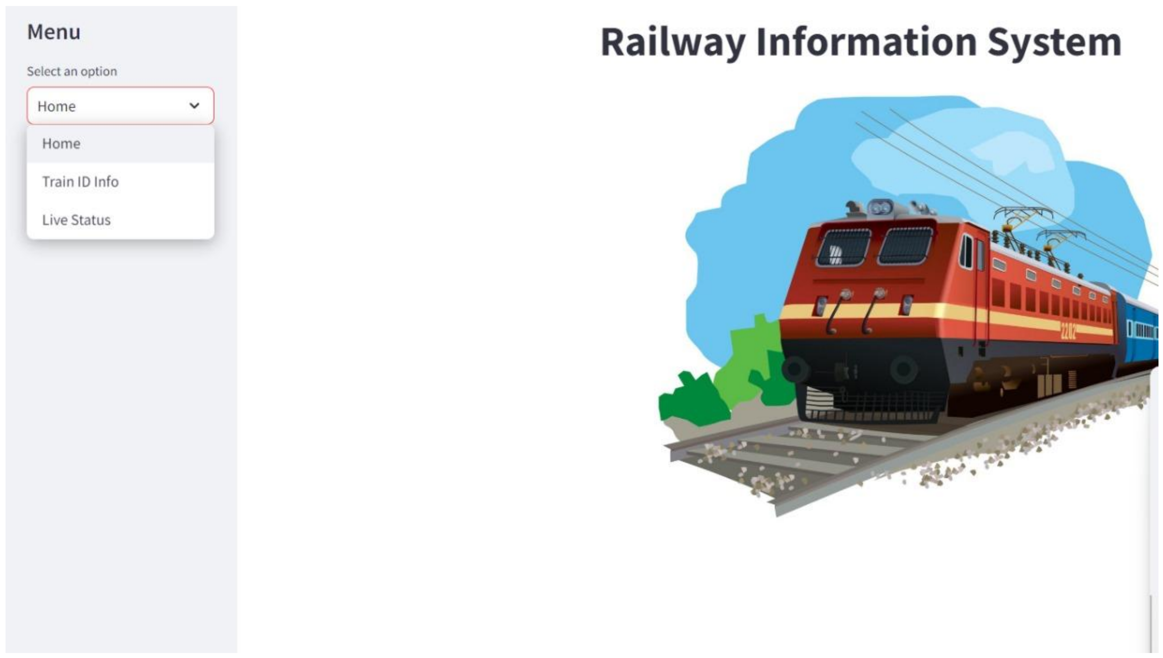
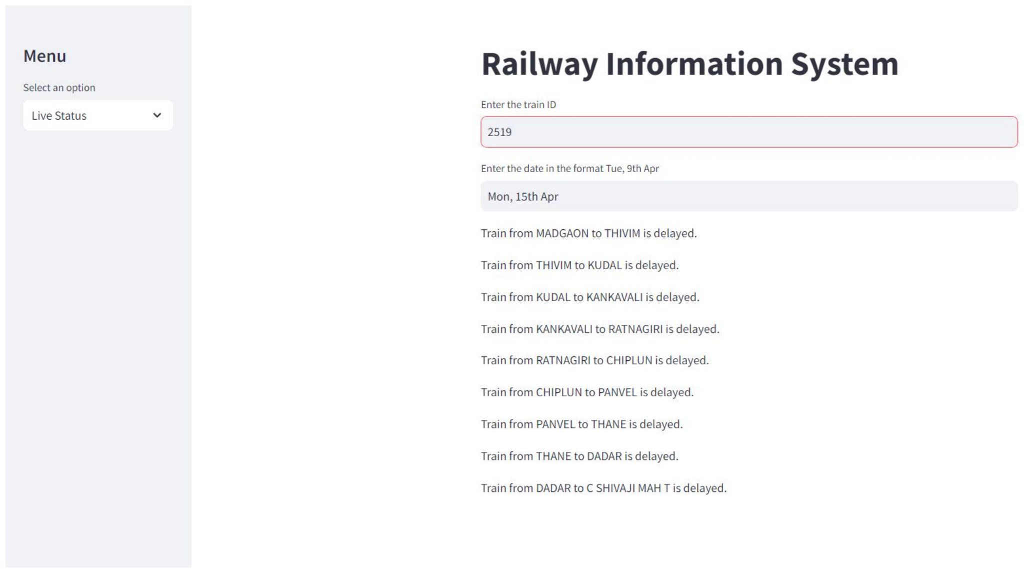


Figure 5.5 User Interface for the Train Delay Prediction



Railway Information System

Enter the train ID

2519

Enter the date in the format Tue, 9th Apr

Mon, 15th Apr

```
Train from MADGAON to THIVIM is delayed.
Train from THIVIM to KUDAL is delayed.
Train from KUDAL to KANKAVALI is delayed.
Train from KANKAVALI to RATNAGIRI is delayed.
Train from RATNAGIRI to CHIPLUN is delayed.
Train from CHIPLUN to PANVEL is delayed.
Train from PANVEL to THANE is delayed.
Train from THANE to DADAR is delayed.
Train from DADAR to C SHIVAJI MAH T is delayed.
```

Figure 5.6 Output for the Train Delay Prediction

The figure 5.5 is the output when we opt for the Live Status and enter the Train ID of a train we want to search for then we will be getting the output which tells us the Delay Status of the Train.

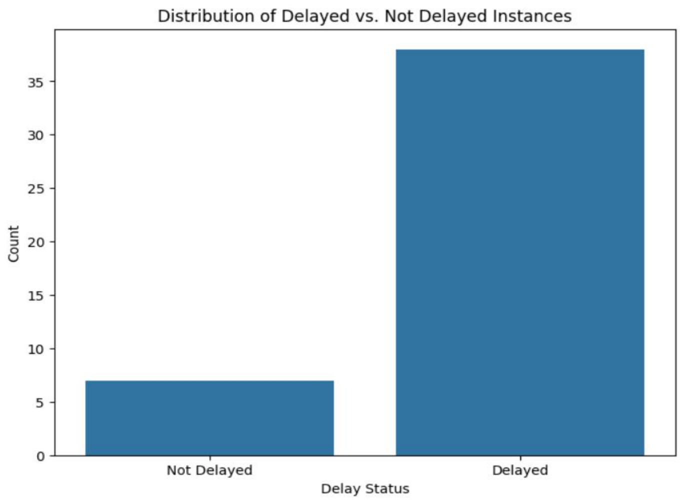
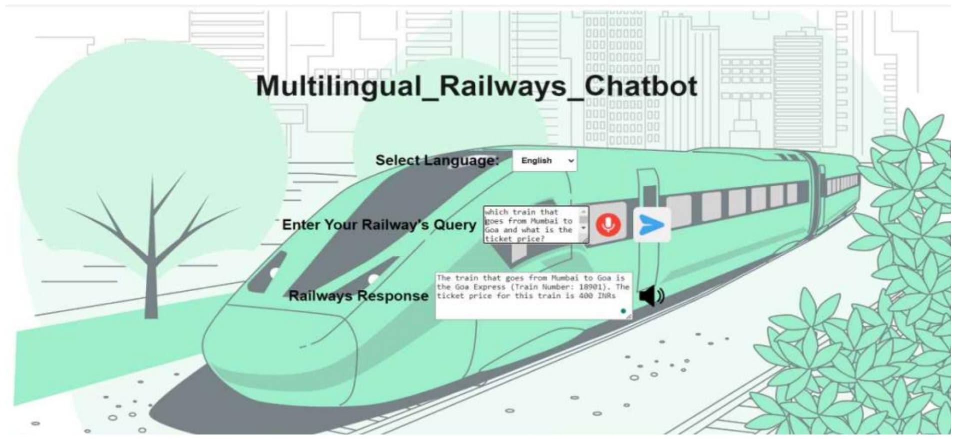


Figure 5.7 Histogram of Delay Status

The figure 5.6 shows the distribution of delayed vs. not delayed instances. The x-axis represents the delay status, with "Not Delayed" on the left and "Delayed" on the right. The y-axis represents the count, or the number of instances.

In this case, there are more instances that are classified as "Not Delayed" than "Delayed". This suggests that the majority of the data points likely represent situations where there were no delays. Without knowing more about the context of this data, it's difficult to say for certain what the reasons might be behind the delays or why there are more instances of on-time situations.

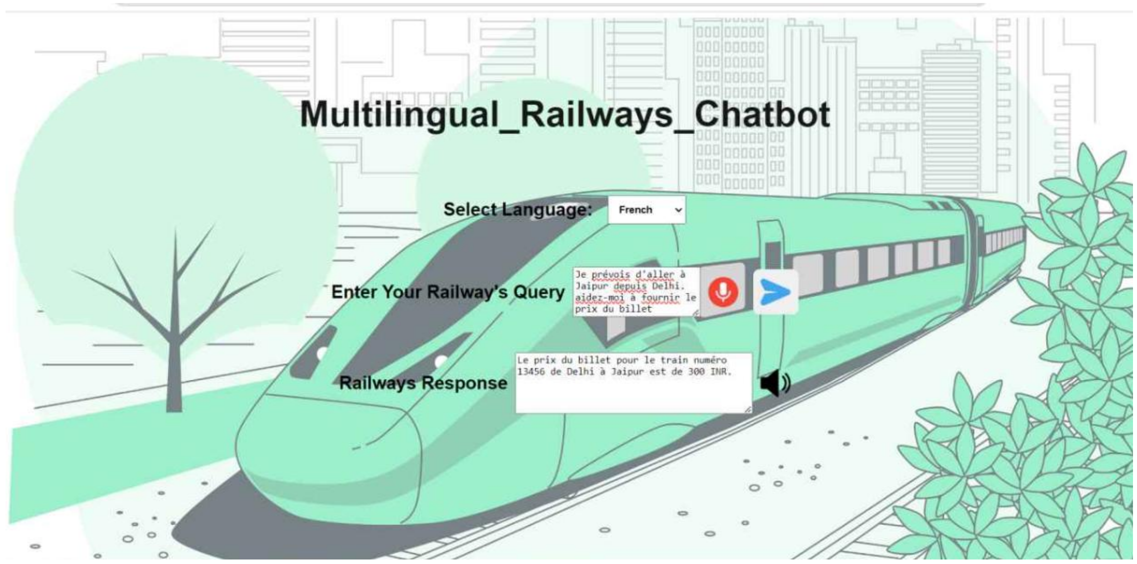


which train that goes from Mumbai to Goa and what is the ticket price?

The train that goes from Mumbai to Goa is the Goa Express (Train Number: 18901). The ticket price for this train is 400 INRs

Figure 5.8 User Interface Output in English

The figure 5.7 is the output for the Multilingual Webpage , In this page first most there is a drop down list to select the Language of our choice and then we can enter our Query either by typing with the Keyboard and even we can pass the input by voice and click on the send icon then it will start processing.



Je prévois d'aller à Jaipur depuis Delhi. aidez-moi à fournir le prix du billet

Le prix du billet pour le train numéro 13456 de Delhi à Jaipur est de 300 INR.

Figure 5.9 Output in French

After that we would be getting the response for our query in the Railway Response Text Area .Even the Output can also be in the text and as well as the Voice format which enhances the User Interactivity and Experience.

6 Conclusion

In conclusion, the integration of real-time data sources, cosine similarity, TF-IDF techniques, and the development of a multilingual railway query answering web page represent significant advancements in the domain of train delay prediction and passenger inquiry systems. By harnessing the power of real-time data, predictive models can offer more accurate and timely predictions of train delays, enabling operators to take proactive measures to minimize disruptions and improve service reliability. Moreover, techniques such as cosine similarity and TF-IDF facilitate the extraction of valuable insights from textual data, enhancing the predictive capabilities of the models.

Simultaneously, the development of a multilingual web page for railway inquiries addresses the diverse linguistic needs of users, fostering inclusivity and accessibility. Integration of natural language processing technologies enables seamless translation of queries and responses, facilitating efficient communication between users and the railway system across different languages.

7 Future Work

In future train delay prediction endeavors, integrating real-time data from IoT sensors and crowd-sourced information can refine accuracy and timeliness. Advanced machine learning techniques like deep learning architectures offer potential for capturing intricate data patterns, thus enhancing predictive capabilities. Expanding language support on railway inquiry web pages can accommodate diverse user demographics, amplifying accessibility and user experience. Integration of natural language processing (NLP) facilitates seamless translation, fostering efficient communication across languages. Geolocation-based services can further enhance user experience by providing personalized recommendations and location-specific information, ensuring a more intuitive and user-friendly interface for a global audience.

Moreover, enhancing the accessibility of railway inquiry platforms for users with disabilities should be a priority in future developments. Implementing features such as screen readers, voice commands, and compatibility with assistive technologies can ensure that all passengers, regardless of their physical abilities, can easily access and navigate the railway inquiry system.

Overall, the future of train delay prediction and railway inquiry systems lies in the convergence of advanced technologies, user-centric design principles, and a commitment to continuous improvement. By embracing innovation and leveraging the power of data-driven insights, railway operators can create a seamless and enjoyable experience for passengers while optimizing operational efficiency and minimizing disruptions.

8 Bibliography

[1] *Heterogeneous Machine Learning Ensembles for Predicting Train Delays.* **Mostafa Al, Ghamdi, Gerard, Parr and Wenjia, Wang.** 12, UK : IEEE, 2024, IEEE Xplore, Vol. I, pp. 50-62. 1524-9050.

[2] *REAL-TIME PASSENGER TRAIN DELAY PREDICTION USING MACHINE.* **Shashi Kumar, BN and Swetha, Shri K.** 6, U.S : International Research Journal of Modernization in Engineering Technology and Science, 2020, IEEE, Vol. II, p. 261. 2582-5208.

[3] *Train Delay Prediction System in India Using.* **Heena, Gupta and Vidya Shree, MG.** 3, Bengaluru : International Advanced Research Journal in Science, Engineering and Technology, 2021, Vol. 8. 2393-8021.

[4] *TRAIN DELAY PREDICTION USING A SUPPORT VECTOR MACHINE.* **Shetty, Shraddha G.** 3, Karnataka : International Research Journal of Modernization in Engineering Technology and Science, 2023, Vol. 5. 2582-5208 .

[5] *Real-Time Passenger Train Delay Prediction Using Machine Learning: A Case Study With Amtrak Passenger Train Routes.* **Pipatphon, Lapamonpinyo, Sybil ,Derrible and Francesco, Corman.** 5, Amtrak : IEEE , 2021, Vol. 3. 2687-7813.