# **FLYFEST**

**ECONOMICAL travel with Cultural Highlights**

A PROJECT REPORT

#### Submitted by

SUMAN MAHATO [RA2111003010659]

DHRUV SAINI [RA2111003010672]

#### Under the Guidance of

## **Dr. Jane Olive Sharon**

(Assistant Professor, Department of Computing Technology)

### *in partial fulfilment of the requirements* *for the degree of*

## BACHELOR OF TECHNOLOGY

## in

## COMPUTER SCIENCE ENGINEERING



## DEPARTMENT OF COMPUTING TECHNOLOGY

## COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

## KATTANKULATHUR- 603203

### NOVEMBER 2024

Department of Networking and Communications

##### SRM Institute of Science and Technology

##### Own Work\* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

##### Degree/ Course : B. Tech. in CSE

**Student Name :** Suman Mahato, Dhruv Saini

##### Registration Number : RA2111003010659, RA2111003010659

**Title of Work :** FLYFEST: Economical travel with Cultural Highlights

I / We hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism\*\*, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

* Clearly referenced / listed all sources as appropriate
* Referenced and put in inverted commas all quoted text (from books, web, etc)
* Given the sources of all pictures, data etc. that are not my own
* Not made any use of the report(s) or essay(s) of any other student(s) either past or present
* Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
* Complied with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalised in accordance with the University policies and regulations.

|  |
| --- |
| **DECLARATION:** |
| I am aware of and understand the University’s policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above. |
| If you are working in a group, please write your registration numbers and sign with the date for every student in your group. |



# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR – 603 203

## BONAFIDE CERTIFICATE

Certified that 18CSP107L - Minor Project [18CSP108L- Internship] report titled “**FLYFEST: ECONOMICAL TRAVEL WITH CULTURAL HIGHLIGHTS”** is the bonafide work of “SUMAN **MAHATO [RA2111003010659], DHRUV SAINI [RA2111003010672]”** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE SIGNATURE**

##### Dr. JANE OLIVE SHARON Dr.

|  |  |  |
| --- | --- | --- |
| **SUPERVISOR**  Assistant Professor,  Department of Computing Technologies |  | **PROFESSOR & HEAD**  Department Of  Computing  Technologies |

**ACKNOWLEDGEMENTS**

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We wish to thank **Dr. Revathi Venkataraman,** Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work. We are incredibly grateful to our Head of the Department, **Dr. M. Lakshmi,** Professor and Head, Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinator, **Dr. G. ,** Associate Professor**,** Panel Head, **Dr. P. Murali**, Professor and members, **Dr. Jane Olive Sharon**, Assistant Professor, **Dr.** , Assistant Professor, Panel MemberDepartment of Computing Technologies, School of Computing, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Dr. Sindhuja M**, Department of Computing Technology, School of Computing, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr. Jane Olive Sharon**, Assistant Professor, Department of Computing Technology, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Computing Technology department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

SUMAN MAHATO [RA2111003010659]

DHRUV SAINI [RA2111003010672]

**ABSTRACT**

In today's budget-conscious world, travelers seek affordable flight options while pursuing unique experiences. "Fly Fest: Economical Travel with Cultural Highlights" addresses this demand by creating an integrated system that combines flight fare data with cultural and recreational event information to offer affordable, experience-rich travel recommendations across India. By analysing and preprocessing historical flight fare data and event information, the system identifies trends and recommends travel options that align with budgetary and cultural preferences.

The project leverages machine learning techniques, specifically a **Random Forest Regressor**, to predict low-fare trends based on various factors, including time of booking, seasonality, and demand. This predictive model informs users of the optimal times to book flights for the lowest possible fares. Additionally, an event recommendation engine based on **Content-Based Filtering** suggests activities, festivals, and cultural events at different destinations, tailored to the traveler's interests. For example, users interested in adventure sports may receive recommendations for water sports and trekking events that coincide with the best times to visit, along with the lowest fare options from their starting location.

The system’s backend processes and normalizes data from both **historic data, current data** and possibly from Event Data and Flight Fares APIs to ensure accuracy and consistency. Fare predictions and event recommendations can be updated dynamically, based on real-time data, providing users with up-to-date travel suggestions that adapt to changing prices and seasonal events. The architecture includes a preprocessing pipeline for data cleaning, feature engineering, and normalization to optimize model performance, followed by a training phase that uses historical data to ensure robust fare predictions.

For the recommendation engine, **Content-Based Filtering** analyses activity features and preferences, while additional filtering layers ensure recommendations remain relevant to the user’s destination and timeframe. For instance, a user interested in cultural festivals is recommended unique experiences specific to the city they plan to visit. Furthermore, an intuitive user interface visualizes fare trends, event highlights, and relevant activities, empowering users to make well-informed travel decisions.

This project specifically targets budget travelers who value cultural immersion, offering an affordable means to explore new destinations. By integrating fare predictions with event-based recommendations, **Fly Fest** delivers a travel planning solution that combines cost-efficiency with personalized cultural experiences. The system offers a seamless and interactive platform that not only aids in budget management but also promotes cultural enrichment, helping travelers explore India’s diversity without financial strain. This innovative approach has the potential to redefine travel planning, meeting the needs of those seeking meaningful yet economical travel experiences.

**TABLE OF CONTENTS**

[**ABSTRACT**](#_heading=h.2xcytpi) **v**

**TABLE OF CONTENTS vi**

[**LIST OF FIGURES**](#_heading=h.30j0zll) **vii**

**LIST OF TABLES viii**

[**ABBREVIATIONS**](#_heading=h.3znysh7) **ix**

**CHAPTER TITLE PAGE**

**NO. NO.**

[**1 INTRODUCTION**](#_heading=h.3dy6vkm) **1**

* 1. General (Introduction to Project) 2
  2. Motivation 3
  3. Sustainable Development Goal of the Project 4

[**2 LITERATURE SURVEY**](#_heading=h.1ci93xb) **5**

2.1 Related Work [DO] 5

2.2 Limitations Identified from Literature Survey (Research Gaps)[do] 11

2.3 Research Objectives 12

2.4 Product Backlog (Key user stories with Desired outcomes) 13

2.5 Plan of Action (Project Road Map) 15

**3 SPRINT PLANNING AND EXECUTION METHODOLOGY 21**

3.1 SPRINT I 21

3.1.1 Objectives with user stories of Sprint I 23

3.1.2 Functional Document 25

3.1.3 Architecture Document 28

3.1.4 Outcome of objectives/ Result Analysis 29

3.1.5 Sprint Retrospective 30

3.2 SPRINT II 31

3.2.1 Objectives with user stories of Sprint II 32

3.2.2 Functional Document 33

3.2.3 Architecture Document 34

3.2.4 Outcome of objectives/ Result Analysis 35

3.2.5 Sprint Retrospective 36

3.3 SPRINT III 37

3.3.1 Objectives with user stories of Sprint III 38

3.3.2 Functional Document 39

3.3.3 Architecture Document 40

3.3.4 Outcome of objectives/ Result Analysis 41

3.3.5 Sprint Retrospective 42

**6 RESULTS AND DISCUSSIONS 40**

6.1 Project Outcomes (Performance Evaluation, Comparisons, Testing Results ) 41 6.2 Subtitle 42

**7 CONCLUSION AND FUTURE ENHANCEMENT 45**

**REFERENCES 46**

**APPENDIX**

**A. CODING 48**

**B. CONFERENCE PUBLICATION 50**

**C. JOURNAL PUBLICATION 51**

**D. PLAGIARISM REPORT 52**

**LIST OF FIGURES**

**CHAPTER TITLE PAGE**

**NO. NO.**

##### [2.1 Thresholding segmentation in action on the skin lesion image input](#_heading=h.3whwml4) . . 4

[2.2 **Computer Vision Pipeline**](#_heading=h.2bn6wsx). . . . . . . . . . . . . . . . . . . . . . . . . . 5

* 1. [**ROC curve CNN and dermatologists**](#_heading=h.3rdcrjn) . . . . . . . . . . . . . . . . . . . . 8
  2. **C**[**onfusion matrix with CNN vs doctors**](#_heading=h.qsh70q) . . . . . . . . . . . . . . . . . . . 10

[3.3 **Adam optimizer error rate**](#_heading=h.3as4poj). . . . . . . . . . . . . . . . . . . . . . . . . 11

[4.1 **Sample lesions from each class**](#_heading=h.1pxezwc). . . . . . . . . . . . . . . . . . . . . . . 13

[4.2 **File structure of dataset**](#_heading=h.49x2ik5). . . . . . . . . . . . . . . . . . . . . . . . . . . 13

[4.3 **Diagnosis Techniques Plot**](#_heading=h.2p2csry) . . . . . . . . . . . . . . . . . . . . . . . . . . 15

* 1. [**Number of data points of each class**](#_heading=h.147n2zr). . . . . . . . . . . . . . . . . . . . 16

##### [Effect of SMOTE and number of variables on KNN (Euclidean Distance)](#_heading=h.3o7alnk) 17

[4.6 **YOLO CNN Layers**](#_heading=h.23ckvvd). . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

[4.7 **ANN sample architecture**](#_heading=h.ihv636). . . . . . . . . . . . . . . . . . . . . . . . . . 23

[4.8 **Convolution Layer simulation**](#_heading=h.32hioqz). . . . . . . . . . . . . . . . . . . . . . . 25

[4.9 **Pooling layer simulation**](#_heading=h.1hmsyys) . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

[4.10 **CNN complete architecture**](#_heading=h.41mghml) **26**

* 1. [**CNN pre assigned weights for each class**](#_heading=h.2grqrue) . . . . . . . . . . . . . . . . . . 27
  2. [**Sample layers of our CNN model**](#_heading=h.vx1227). . . . . . . . . . . . . . . . . . . . . . 28

[5.1 **Overall architecture of Product.**](#_heading=h.3fwokq0). . . . . . . . . . . . . . . . . . . . . . 32

[5.2 **Difference between UI & UX**](#_heading=h.1v1yuxt). . . . . . . . . . . . . . . . . . . . . . . . 33

[5.3 **Ionic Framework**](#_heading=h.4f1mdlm). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

[5.4 **Ionic Framework**](#_heading=h.26in1rg). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

[5.5 **Back-end Architecture**](#_heading=h.2u6wntf) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

[5.6 **Components of API**](#_heading=h.19c6y18). . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

[5.7 **REST API Architecture**](#_heading=h.3tbugp1). . . . . . . . . . . . . . . . . . . . . . . . . . . 38

[5.8 **SOAP API Architecture**](#_heading=h.28h4qwu). . . . . . . . . . . . . . . . . . . . . . . . . . . 39

**LIST OF TABLES**

**CHAPTER TITLE PAGE**

**NO. NO.**

##### [2.1 Thresholding segmentation in action on the skin lesion image input](#_heading=h.3whwml4) . . 4

[2.2 **Computer Vision Pipeline**](#_heading=h.2bn6wsx). . . . . . . . . . . . . . . . . . . . . . . . . . 5

* 1. [**ROC curve CNN and dermatologists**](#_heading=h.3rdcrjn) . . . . . . . . . . . . . . . . . . . . 8
  2. [**confusion matrix with CNN vs doctors**](#_heading=h.qsh70q) . . . . . . . . . . . . . . . . . . . 10

**viii**

**ABBREVIATIONS**

**MSE** Mean Squared Error

**RMSE** Root Mean Squared Error

**MAPE** Mean Absolute Percentage Error

**MAE** Mean Absolute Error

**ML** Machine Learning

**GBM** Gradient Boosting Machines

**RF** Random Forest

**DT** Decision Tree

**R2** Coefficient of Determination (R-squared Score)  
**LR** Linear Regression

**Z-score** Standard Score (for outlier detection)

**DB** Database

**DNN** Deep Neural Network

**API** Application Programming Interface

**AI** Artificial Intelligence

**NLP** Natural Language Processing

**GRU** Gated Recurrent Unit

**UI** User Interface

ix

**CHAPTER 1**

**INTRODUCTION**

* 1. **Background**

In an increasingly budget-conscious world, travelers are not only seeking affordable options for transportation but also yearning for immersive cultural experiences that make each journey meaningful. Current travel and booking platforms primarily focus on providing competitive prices without factoring in recommendations for local events, cultural highlights, or season-based experiences. As a result, budget travelers who wish to explore new destinations face limitations, as they lack tools that integrate economical fare options with destination-specific events. This gap in the travel industry drives the development of our project, **"Fly Fest: Economical Travel with Cultural Highlights,"** which aims to address these limitations by providing a holistic, cost-effective travel solution.

With travel costs often fluctuating due to demand, season, and route, accurate fare prediction becomes crucial for budget-conscious travelers. Additionally, cultural and recreational events vary widely across regions, seasons, and even specific dates, requiring a flexible, data-driven approach to manage and present relevant recommendations. Our project builds on recent advancements in data analytics and machine learning, utilizing historical fare data to predict low-cost travel opportunities and aligning these with event-based recommendations.

The core of our system relies on machine learning algorithms, particularly a **Random Forest Regressor**, which captures complex pricing patterns to forecast the lowest fare trends for specific routes. Random Forest, as an ensemble method, leverages multiple decision trees to improve prediction accuracy by averaging results across trees, thus minimizing overfitting and enhancing robustness. For this fare prediction task, historical pricing data is pre-processed, normalized, and cleaned, enabling the model to recognize patterns influenced by time, demand, and route-specific factors. This fare prediction model equips users with insights on when to book flights to secure the best prices.

Complementing the fare prediction module is an event recommendation system powered by **Content-Based Filtering**. This filtering technique suggests activities based on the specific attributes of each destination, such as event type (e.g., festivals, adventure sports), location, and season. Unlike collaborative filtering, which relies on user interaction data, content-based filtering makes recommendations based on the features of the items themselves, making it ideal for travelers who may not have interacted with a platform extensively. By focusing on activities and cultural events relevant to each destination, this recommendation engine helps travelers discover unique experiences without overwhelming them with irrelevant options.

The project integrates real-time data from multiple APIs for flight information and events, allowing for dynamic updates that reflect current prices and available experiences. These APIs provide a stream of up-to-date information, making the system adaptable to sudden changes in flight fares and event schedules. Additionally, data preprocessing, including one-hot encoding for categorical variables, feature scaling, and imputation of missing values, ensures that the prediction model performs with high accuracy and reliability.

Our target users include budget travelers who wish to explore new places and partake in local experiences without overspending. The system’s interface is designed to provide an intuitive and seamless experience, allowing users to view fare trends and event recommendations at a glance. With interactive visualizations, travelers can easily track fare fluctuations over time, helping them make informed decisions about their journey. For example, a traveler interested in water sports might be directed to the most affordable flights to Goa during peak water sports season, creating an itinerary that is both economical and culturally enriching.

Ultimately, **Fly Fest** aims to redefine budget travel by creating a system that aligns economic efficiency with cultural discovery. By combining predictive analytics with content-based recommendations, our project offers a unique solution for travelers who value both affordability and immersion in local culture. This approach not only enhances the overall travel experience but also supports local tourism by promoting culturally significant events and activities. Through this integrated platform, **Fly Fest** empowers travelers to discover India’s diverse cultural tapestry without the financial burden traditionally associated with tourism.

* 1. **Motivation**

The motivation for **"Fly Fest: Economical Travel with Cultural Highlights"** stems from the growing demand among budget-conscious travelers for affordable, experience-rich journeys. In an era where travelers prioritize both financial prudence and immersive experiences, standard booking platforms fall short by offering ticket prices without insights into optimal booking times or destination-specific events. This gap leaves many travelers unable to maximize their travel experiences due to high costs or a lack of culturally relevant recommendations.

The unpredictability of airfare pricing, influenced by seasonal shifts, demand patterns, and destination popularity, adds complexity to planning economical travel. By providing fare predictions and event recommendations, **Fly Fest** aims to empower travelers with tools for making well-informed decisions about when and where to travel to optimize cost savings. This model not only addresses the need for budget-friendly options but also emphasizes cultural enrichment, encouraging users to explore festivals, recreational activities, and local attractions that provide a deeper connection to the places they visit.

Our project leverages advancements in data analytics and machine learning to address these challenges. With the Random Forest Regressor model for fare predictions, **Fly Fest** allows users to determine the most cost-effective times for booking flights, helping them save money without compromising on quality. The additional recommendation engine, powered by content-based filtering, offers targeted suggestions for events and activities, making each journey unique and culturally relevant. This dual functionality adds value by aligning travel planning with personal interests, transforming travel from a mere transaction into a curated experience.

Furthermore, **Fly Fest** is motivated by the desire to support local tourism industries. By guiding travelers toward cultural events and unique experiences, our platform helps promote lesser-known destinations and activities, which in turn benefits local businesses and communities. This creates a cycle of mutual benefit, where travelers gain access to memorable experiences while contributing to the economic sustainability of their chosen destinations.

**1.3.** **Sustainable Development Goal of the Project**

**"Fly Fest: Economical Travel with Cultural Highlights"** aligns with the **United Nations Sustainable Development Goals (SDGs)**, specifically under **SDG 11: Sustainable Cities and Communities** and indirectly contributes to **SDG 12: Responsible Consumption and Production**. Within the context of **Mobility Solutions**, this project supports sustainable and inclusive travel by providing an affordable, efficient way for people to explore diverse locations, encouraging cultural engagement, and promoting sustainable tourism practices.

SDG 11: Focuses on making cities inclusive, safe, resilient, and sustainable. By providing affordable travel options through fare predictions and culturally rich recommendations, **Fly Fest** enables more people to explore cities and towns in a way that values local traditions and practices. The project also supports SDG 11 by promoting responsible tourism, directing travelers to events and activities that are significant to the local economy. By focusing on regional experiences and encouraging tourists to visit lesser-known sites, **Fly Fest** reduces the pressure on overcrowded tourist destinations, supporting a more equitable distribution of tourism benefits across communities.

**Mobility Solutions** focus on efficient, accessible, and environmentally friendly transport options. In tourism, this also means optimizing routes and reducing unnecessary travel through data-driven decisions. **Fly Fest** contributes to this by using predictive modelling for flight prices, helping travelers to choose flights that are both economical and efficiently timed, reducing the likelihood of last-minute, less sustainable bookings. It encourages thoughtful planning, allowing users to combine cost-effectiveness with sustainable travel habits.

**SDG 12:** The project indirectly aligns with SDG 12 by promoting responsible travel. By offering travel recommendations based on local events and cultural highlights, **Fly Fest** promotes conscious tourism, encouraging users to partake in culturally significant activities that respect and support local heritage. This type of mindful consumption reduces the environmental impact of travel by focusing on quality over quantity and promoting activities that are sustainable and community focused.

The project not only makes travel accessible to a broader audience but also promotes sustainable tourism practices that benefit both travelers and the communities they visit. This dual approach allows **Fly Fest** to contribute to a world where travel is both affordable and responsible, supporting economic and cultural sustainability while fostering cross-cultural connections. Through predictive analytics and recommendation engines, the project embodies the values of Mobility Solutions and Sustainable Development Goals, presenting a pathway to a more equitable and sustainable future in travel.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 Related Work**

The challenge of accurate fare prediction and relevant recommendations has gained traction in recent years, with researchers developing machine learning frameworks that utilize both historical data and real-time inputs to optimize user outcomes. This literature survey explores significant contributions from previous studies, examining fare prediction models, recommendation systems, and the impact of algorithmic advancements in machine learning.

**Airfare Prediction Models and Techniques** In [1], Tziridis et al. developed a machine learning-based airfare prediction model that applies various algorithms to predict fare prices effectively. The model achieved notable accuracy in identifying price patterns, utilizing techniques like Random Forest and Gradient Boosting, which are especially useful for data with high variability, such as airfare data. This study highlights the potential of Random Forest for fare prediction and lays the groundwork for using ensemble models in this domain.

Similarly, the study by Etzioni et al. in [2] focused on mining airfare data to minimize ticket costs. This research integrated web-scraping techniques and machine learning algorithms to analyze patterns in fare fluctuations, identifying the ideal booking windows for consumers. By refining algorithm parameters based on fare trends, the model demonstrated increased accuracy in predicting price changes, thereby maximizing cost savings for travelers. The insights from this work contribute to methodologies for detecting fare variations, an essential aspect of the Fly Fest project.

**Machine Learning Approaches for Price Forecasting** Panigrahi et al. in [3] presented an analysis comparing multiple machines learning algorithms, including Support Vector Machines (SVM), Linear Regression, and Decision Trees, for fare prediction. Their study showed that while Decision Trees and SVMs performed well with structured data, Random Forest provided a higher accuracy level for predicting ticket prices in complex and fluctuating markets. This finding reinforces the use of Random Forest in our project for robust, accurate fare predictions.

The use of recommendation systems in trip planning and travel choices was explored by Knijnenburg and Willemsen in [4]. They implemented collaborative filtering techniques to generate personalized travel suggestions, integrating user data with historical activity preferences. By comparing content-based and collaborative filtering, the study found that collaborative filtering was more effective in recommending activities that matched users’ unique interests. These insights inform Fly Fest’s approach to implementing content-based and collaborative filtering, enhancing user-specific recommendations.

**Trip Planning Systems and Recommender Models** The survey by Sylejmani and Dika in [5] offers a comprehensive review of various trip-planning systems, particularly those that use recommendation engines to provide activity and destination suggestions. This work highlights the importance of user feedback and iterative refinements to improve recommendation accuracy, especially in the dynamic context of travel planning. The authors observed that user-centred design, combined with machine learning algorithms, helps deliver actionable travel insights. This principle applies directly to our project’s recommendation engine, enabling us to personalize trip suggestions effectively.

Prasad and Sharma in [6] further investigated fare prediction using machine learning, focusing on a hybrid approach that combines SVM with Random Forest. The results demonstrated that hybrid models could provide better predictive accuracy by capitalizing on the strengths of multiple algorithms. In Fly Fest, this hybrid approach could be considered to optimize fare prediction for various destination routes, making the model adaptable to different fare dynamics across seasons and locations.

**Impact of Neural Network Training and Epochs in Prediction Accuracy** Afaq and Rao’s study in [7] explored the role of epochs in training neural networks, demonstrating that optimal epoch counts prevent overfitting and improve predictive accuracy. This study found that too many epochs lead to overfitting, while too few result in underfitting. For Fly Fest, understanding epoch dynamics can be instrumental when incorporating deep learning models for fare prediction or user preference learning, providing an adjustable framework based on the complexity of the dataset.

The study by Wang et al. in [8] introduced a machine learning framework for airfare price prediction, incorporating algorithms like Gradient Boosting Machines and Random Forest. Their findings highlighted the benefits of ensemble methods, which combined multiple algorithms to reduce prediction errors. This ensemble approach is beneficial for Fly Fest, where fare prediction requires adaptation to diverse seasonal and event-driven trends across India.

**Neural Networks for Flight Fare Forecasting** Degife and Lin’s research in [9] used a gated recurrent unit (GRU) model for flight fare forecasting, showing that deep learning models are highly effective in identifying complex patterns in fare data. GRU’s ability to retain long-term dependencies in time-series data makes it a suitable model for predicting fare fluctuations based on historical data. The findings suggest that GRU could be integrated into Fly Fest’s fare prediction system to achieve higher accuracy and adaptability to dynamic fare patterns.

**Recommendation System Techniques: Content-Based and Collaborative Filtering** Beel et al. [10] conducted an extensive survey on recommendation systems in various domains, including content-based and collaborative filtering. This survey showed that content-based filtering is particularly effective when user preferences are based on specific features (e.g., types of activities), while collaborative filtering performs well when recommendations depend on user history and feedback. Fly Fest leverages both approaches: content-based filtering for destination activity recommendations and collaborative filtering to match users with popular activities enjoyed by similar travelers.

Liaw and Wiener’s work in [11] evaluated Random Forest’s classification and regression abilities, emphasizing its effectiveness in dealing with high-dimensional data. Their research indicates that Random Forest’s decision trees enable robust fare predictions without overfitting, a key advantage for fare datasets that fluctuate with market demand and seasonal influences. Given these advantages, Fly Fest incorporates Random Forest for predicting dynamic fare patterns, ensuring model performance is both accurate and scalable.

**Comparative Analysis of Filtering Techniques for Recommender Systems** Phalle and Bhushan [12] compared content-based and collaborative filtering approaches, concluding that collaborative filtering generally outperforms content-based filtering in scenarios where user history data is rich. However, content-based filtering is useful for niche recommendations, such as specialized travel activities unique to certain cities. Fly Fest’s recommendation system applies content-based filtering to suggest distinct cultural and adventure events, enhancing travel planning for diverse user interests.

Thannimalai and Zhang [13] further examined the integration of content-based and collaborative filtering, proposing a hybrid model that combines both methods to improve recommendation diversity and accuracy. The study’s insights inform Fly Fest’s system, where a hybrid approach provides users with relevant travel recommendations based on both personal preferences and popular activities.

**Holistic Fare Prediction with Machine Learning Techniques** Kalampokas et al. [14] presented a comprehensive model for airfare prediction using machine learning techniques. Their model integrated historical data, user feedback, and predictive analytics, achieving high accuracy across diverse travel routes. This holistic approach in fare prediction highlights the importance of using a wide range of algorithms, a principle mirrored in Fly Fest’s design, which applies multiple techniques for both fare prediction and recommendation tasks.

**Summary**

The literature reviewed provides a well-rounded perspective on the technical frameworks supporting airfare prediction and recommendation systems. Key takeaways include the robustness of Random Forest in fare prediction, the flexibility of ensemble models, and the effectiveness of content-based and collaborative filtering in personalized recommendations. Each study offers insights that enhance Fly Fest’s development, from optimizing fare prediction accuracy to personalizing user experience with targeted recommendations.

By combining these advanced techniques, Fly Fest aims to create a travel solution that meets user needs for affordable and culturally rich experiences, laying a solid foundation for continued advancements in travel technology.

**2.2 Limitations Identified from Literature Survey (Research Gaps)**

1. **Geographical Constraints**: Many studies on fare prediction and recommendations are region-specific, limiting the model's effectiveness in other areas with different cultural, economic, and seasonal patterns. A broader approach is needed to make travel predictions universally applicable.
2. **Reliance on High-Quality Data**: Data preprocessing remains crucial for accurate fare and recommendation models. Inconsistent data cleaning and preprocessing methods can reduce model reliability, highlighting the need for standardized preprocessing.
3. **Short-Term Data Validation**: Several models are tested on short timeframes, overlooking long-term shifts in travel behaviour and fare trends. Extending validation to cover seasonal and yearly variations would improve prediction robustness.
4. **Limited Real-Time Data Integration**: Real-time fare and event data are vital for immediate recommendations. However, current studies lack real-time data integration, limiting responsiveness and real-world applicability.
5. **Underexplored Cultural and Social Factors**: The socioeconomic aspects affecting travel decisions, like regional festivals or events, are often underexplored. Better integration of cultural factors would enhance recommendation quality.
6. **Lack of Multi-Model Approach**: While ensemble models can increase predictive accuracy, most studies use single models. An ensemble or hybrid approach could improve adaptability across various fare prediction and recommendation contexts.
7. **2.3 Research Objectives**

**1. Enhancing Geographic and Cultural Relevance**

1. Improve the accuracy of flight fare and event recommendations by integrating region-specific factors.
2. Enhance recommendations by localizing data related to fare trends and culturally significant events, festivals, and activities.

**a.** By integrating localized data—such as peak seasons, regional holidays, and local festivities—the system can refine recommendations for each destination.

**b.** Address geographic diversity by including region-specific events (e.g., water sports at coastal destinations or traditional festivals unique to specific regions).

**c.** Provide culturally enriched recommendations that focus on local activities and festivals, encouraging deeper engagement with the destination’s culture.

**d.** By offering insights into high-value destinations during off-peak times, Fly Fest promotes sustainable tourism and balances visitor distribution across locations.

**2. Ensuring Data Standardization and Quality**

1. Establish data consistency to improve prediction accuracy and the reliability of event recommendations.
2. Create a standardized framework for data cleaning, integration, and validation.

**a.** Apply advanced imputation techniques to address missing fare and event data, enhancing model predictions and recommendation accuracy.

**b.** Use automated anomaly detection to prevent distorted predictions, removing any inconsistent or outlier data.

**c.** Standardize data from multiple sources, ensuring uniformity across different data inputs (e.g., flight data from various airlines and events from local tourism boards).

**d.** Implement data normalization to standardize fare and event features across diverse locations and time frames, maintaining reliable predictions.

**3. Multi-Source Event and Activity Recommendation Integration**

1. Provide seamless integration of events and activities from multiple sources, enhancing travel recommendations.
2. Improve cultural relevance and diversity of recommendations by combining multiple event and activity sources.

**a.** Support a variety of event types, from major cultural festivals to adventure sports, and local seasonal highlights, catering to varied traveler preferences.

**b.** Create a recommendation system that accounts for overlapping interests (e.g., a traveler interested in adventure sports in coastal regions could receive recommendations for paragliding, yachting, and scuba diving).

**c.** Use collaborative and content-based filtering to predict travel preferences based on previous behaviours and selected interests, enhancing recommendation relevance.

**d.** Enable adaptive recommendations, where dynamic changes (e.g., unexpected events) are instantly reflected in the recommendation list, keeping travelers updated.

**4. Incorporating Real-Time Flight and Event Data**

1. Ensure that recommendations remain relevant by integrating real-time flight data and events, adjusting predictions and suggestions accordingly.
2. Create a responsive system that adapts to real-time data updates.

**a.** Implement API integrations to retrieve the latest flight fare and event information, ensuring predictions reflect real-time market conditions.

**b.** Notify users of any last-minute changes, such as price drops or newly announced events, helping them make informed travel decisions.

**c.** Update predictive models and recommendation engines to reflect fresh data inputs, maintaining the system’s accuracy over time.

**d.** Continuously refresh data inputs and model parameters to stay responsive to changing fare patterns and event schedules.

**5. Supporting Budget-Conscious and Sustainable Tourism**

1. Cater to budget-conscious travelers by offering affordable travel recommendations that also encourage sustainable tourism practices.
2. Balance economic travel choices with cultural immersion and sustainable travel practices.

**a.** Provide fare predictions for off-peak travel periods to reduce travel costs and alleviate congestion in popular destinations during high seasons.

**b.** Recommend lesser-known destinations and cultural events, promoting a more balanced tourism impact and economic benefits for local communities.

**c.** Encourage travelers to explore cultural experiences that resonate with local customs, supporting sustainable tourism.

**d.** Utilize machine learning models to ensure fair fare recommendations for budget travelers, facilitating affordable cultural experiences.

**6. Integrating Advanced Machine Learning and Adaptive Forecasting Models**

1. Enhance fare prediction accuracy and recommendation relevance through adaptive machine learning models, including Random Forests and ensemble techniques.
2. Enable the system to adapt to changing data patterns and user behavior for continuous improvement.

**a.** Train a Random Forest Regressor to provide fare predictions by identifying complex relationships in historical fare data, ensuring cost-effective travel options.

**b.** Use ensemble learning techniques to combine predictions from multiple models, improving reliability and accuracy for diverse travel routes and destinations.

**c.** Implement reinforcement learning models that can learn from user interactions and continuously improve fare and recommendation accuracy.

**d.** Leverage neural network models for advanced data pattern recognition, ensuring the system adjusts to new and varied data inputs.

**7. Enabling a User-Centric Interface and Dynamic Dashboard**

1. Provide a user-friendly interface that allows travelers to easily access fare trends, event recommendations, and interactive data visualizations.
2. Improve user engagement through visual tools that present predictions and recommendations in an intuitive format.

**a.** Develop an interactive dashboard that provides users with quick insights into fare trends, activity schedules, and recommended destinations.

**b.** Include filters that allow users to personalize data views based on their specific preferences, travel dates, and interests.

**c.** Incorporate graphs, charts, and calendar views to help users make data-driven travel decisions at a glance.

**d.** Offer downloadable summaries of fare and event recommendations, supporting off-platform decision-making.

**8. Implementing Feedback-Driven Improvements and Continuous Learning**

1. Enable feedback loops to continuously enhance the model's accuracy and relevancy based on real user data and input.
2. Ensure the system remains relevant and improves over time by adapting to user feedback and behavioural patterns.

**a.** Collect user feedback on recommended events and fare predictions to identify areas for improvement.

**b.** Adjust machine learning models based on user behavior and preferences, helping the system adapt to changing trends.

**c.** Incorporate regular model evaluations based on metrics like MAE, MSE, and R² to ensure continuous performance optimization.

**d.** Retrain models periodically with fresh data to ensure ongoing accuracy and relevance in predictions and recommendations.

**2.4 Project Backlog**

**1. Flight Fare Prediction:**

* Description: This feature predicts flight fares based on historical data, using a Random Forest Regressor model to identify trends and the best booking times. The model incorporates factors like seasonality, peak travel times, and demand fluctuations to ensure predictions are accurate.
* User Benefit: Budget-conscious travelers can view price trends and forecasted fares, helping them book at optimal times.
* Key Features: Real-time fare updates and alerts for significant price drops. Historical trend graphs allow users to visualize past price changes for selected routes and dates.

**2. Event and Activity Recommendation System:**

* Description: A content-based recommendation engine suggests activities aligned with user interests, travel dates, and destination events. Activities are categorized by type (e.g., festivals, adventure, seasonal events) and tailored to users’ profiles.
* **User Benefit**: Travelers interested in cultural and local activities receive personalized suggestions, enriching their travel experience.
* **Key Features**: Users can save events to their profile for itinerary planning, with dynamic updates as new event data becomes available. Interest-based grouping lets users join communities for shared event interests.

**3. Real-Time Data Integration:**

* Description: Integrates flight fare data and event listings via APIs to provide the most up-to-date fare predictions and event recommendations.
* User Benefit: Frequent travelers receive accurate, timely information, allowing them to adjust plans based on real-time changes.
* Key Features: Automated updates ensure that fare trends and event recommendations reflect current data. Notifications inform users about new or updated events.

**4. Visualization and Interface Dashboard:**

* Description: An interactive dashboard displays fare trends, event schedules, and alerts in a user-friendly visual format. Users can explore data insights through charts, graphs, and a calendar view.
* User Benefit: Budget travelers can quickly understand fare trends and event details visually, simplifying the planning process.
* Key Features: Filters allow users to customize the view by location, travel dates, and activity interests. The calendar highlights major events by destination, enabling travel date alignment with local cultural activities.

**5. User Preferences and Notification System:**

* Description: Users set preferences for travel interests, budget, and destinations. Tailored notifications alert them about relevant fare drops and event updates.
* User Benefit: Travelers with specific interests and budgets can easily access deals and events matching their preferences, avoiding missed opportunities.
* Key Features: Customizable alerts for fare drop and event updates, with notification history saved on the dashboard for quick reference. Syncing across devices ensures users receive alerts on multiple platforms.

**6. Model Performance Metrics and Feedback Loop:**

* **Description**: Displays model performance metrics like Mean Absolute Error (MAE) and R², allowing developers to assess and improve accuracy. User feedback post-trip helps refine future predictions.
* **User Benefit**: Researchers and developers can monitor model performance and enhance accuracy based on feedback, ensuring predictions remain relevant.
* **Key Features**: Automated model retraining incorporates seasonal trends and updated data. User feedback helps identify improvement areas for event recommendations.

**7. Advanced Analytics and Reporting Tools:**

* Description: Provides detailed analytics for travel agents and power users, allowing them to assess fare trends and seasonal patterns for more tailored recommendations.
* User Benefit: Travel agents and advanced users gain deeper insights, enabling them to offer customized recommendations and plan campaigns around fare forecasts.
* Key Features: Advanced filtering and segmentation tools allow users to analyze fare trends by demographic or travel season, with options for exporting data to CRM tools.

**2.5 Plan of Action**

**Sprint 1: Foundation and Data Integration**

* Objective: Establish the foundational components of the system, focusing on data import, preprocessing, and climate data integration to provide accurate, location-specific short-term and long-term forecasts.
* Set Up Data Import and Preprocessing Pipeline:
  + Import weather and climate data using external APIs or CSVs, focusing on high-quality, standardised data collection.
  + Implement data preprocessing protocols (handling missing values, scaling, and formatting), aligning with objectives to enhance data quality and standardisation.
* Develop Short-Term and Long-Term Forecasting Models:
  + Build initial forecasting models to deliver 24-hour and 7-day predictions for solar and wind energy.
  + Incorporate location-specific data for accurate, targeted forecasting, setting the stage for multi-regional application.
* Integrate Climate Variables:
  + Add key climate variables (e.g., temperature, wind speed, precipitation) to the model inputs, establishing the foundation for accurate predictions.
  + Start incorporating extreme weather events and seasonal trends to enhance model responsiveness to climate impacts.
* Expected Outcomes:
  + Functional data pipeline for importing and preprocessing climate data.
  + Initial short-term and long-term forecasting models delivering location-specific renewable energy predictions.
  + Early integration of climate data into the model to improve accuracy.

### **Sprint 2: Model Enhancement and Validation**

* Objective: Enhance the forecasting model’s reliability by validating predictions against real-world data and incorporating error metrics and adjustment mechanisms.
* Implement Real-World Data Validation:
  + Compare forecast outputs with actual renewable energy production data to identify deviations and accuracy levels.
  + Generate validation reports comparing predicted vs. actual outputs to assess model performance.
* Introduce Accuracy Metrics and Adjustment Mechanisms:
  + Calculate error metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) for each forecast.
  + Establish an automatic adjustment mechanism if the forecast error exceeds a 5% threshold, refining predictions and maintaining reliability.
* Incorporate Additional Climate Data Sources:
  + Expand data inputs by adding supplementary CSVs and climate data for broader geographic applicability.
  + Validate the accuracy of short-term and long-term predictions using a larger dataset.
* Expected Outcomes:
  + Improved model accuracy with real-world data validation and performance reports.
  + Reliable forecast adjustments in response to high error rates, ensuring dependable predictions.
  + Broader data sources for enhanced geographic flexibility and climate adaptability.

### **Sprint 3: Model Refinement, Trend Analysis, and Visualization**

* Objective: Focus on refining a selected model to enhance forecasting accuracy, analyse historical trends in renewable energy production, and generate future predictions with visualisations.
* Model Refinement:
  + Select the best-performing model from Sprint 2’s algorithm comparison (e.g., Random Forest, ARIMA, or LSTM) based on accuracy and reliability metrics.
  + Fine-tune the model’s parameters to improve forecast precision, accounting for specific factors like seasonality and regional climate variations.
  + Implement regular updates using real-time data to keep the model adaptive to the latest climate and energy patterns.
* Historical Trend Analysis:
  + Analyse historical renewable energy production data to identify seasonal trends, high-output and low-output periods, and patterns linked to extreme weather events.
  + Assess the impact of identified trends on energy output, providing insights into the factors most influential on renewable production.
* Future Predictions Generation:
  + Use the refined model to generate future predictions for renewable energy output, focusing on both short-term (24 hours) and long-term (up to 7 days) forecasts.
  + Identify any patterns or anomalies in the predictions to anticipate periods of high or low production, enhancing planning for grid management.
* Visualisation of Trends and Predictions:
  + Create visualisations to represent historical trends, current predictions, and projected future energy outputs, using interactive charts and graphs.
  + Emphasise patterns, seasonality, and extreme event impacts in the visualisations to provide actionable insights for stakeholders.
* Expected Outcomes:
  + A finely tuned model with improved forecast accuracy, adaptable to short-term and long-term renewable energy predictions.
  + Clear identification of historical trends and patterns, supporting informed planning and decision-making for energy providers.
  + Visual representations of future predictions, allowing stakeholders to understand upcoming trends and potential energy production scenarios.
  + Actionable insights into renewable energy production cycles, improving the stability and reliability of grid management strategies.

### **Summary of Roadmap Goals by Sprint:**

1. Sprint 1: Establish data integration and initial forecasting capabilities with climate data variables.
2. Sprint 2: Validate models with real-world data, refine error metrics, and expand data sources for geographic flexibility.
3. Sprint 3: Focus on refining the best model, identifying historical trends, generating future predictions, and creating visualisations for insights.

**CHAPTER 3**

**SPRINT PLANNING AND EXECUTION METHODOLOGY**

**3.1 SPRINT 1**

### **3.1.1 Objectives**

**1. Establish Data Import and Preprocessing Pipeline**

1. **Objective**: Set up a robust pipeline to import and preprocess flight fare and event data from external APIs and CSV files, ensuring data quality and consistency.
2. **User Story**: As a data engineer, I want to reliably import and preprocess fare and event data, so the recommendation system has accurate, high-quality input data.

**Acceptance Criteria**:

* 1. Import data from verified sources (e.g., AviationStack, Eventbrite).
  2. Handle missing values, duplicates, and format inconsistencies during preprocessing.

**2. Build Initial Fare Prediction Model**

1. **Objective**: Develop and test the initial fare prediction model using Random Forest Regressor, targeting optimal booking times.
2. **User Story**: As a traveler, I want accurate fare predictions, so I can book flights at the best price.

**Acceptance Criteria**:

* 1. Model predicts flight fares with reasonable accuracy.
  2. Initial predictions are available for key routes.
  3. Performance metrics (e.g., MAE, R²) are measured and documented.

**3. Event Recommendation System Setup**

1. **Objective**: Begin developing a content-based recommendation model for events and activities based on user preferences and destination.
2. **User Story**: As a traveler, I want relevant activity recommendations for my destination, so I can plan a culturally immersive trip.

**Acceptance Criteria**:

* 1. Event data includes location, category, and timing details.
  2. Recommendations match user preferences based on features like destination and event type.
  3. Integration with user profile to support preference-based suggestions.

### **3.1.2 Functional Document**

#### **Introduction**

The **Fly Fest: Economical Travel with Cultural Highlights** project aims to offer budget travelers a system that combines accurate flight fare predictions with curated event recommendations across India. By analyzing fare trends and providing activity suggestions based on user preferences and travel dates, this project helps travelers secure economical travel options while engaging in culturally immersive experiences. **Sprint 1** focuses on building a robust data pipeline for importing and preprocessing fare and event data, developing an initial fare prediction model, and creating a content-based recommendation system.

**Sprint Goal**

The goal of **Sprint 1** is to establish the foundational elements of an efficient travel recommendation system, capable of providing accurate fare forecasts and relevant event suggestions for each destination. This sprint will deliver an operational data pipeline, a preliminary fare prediction model, and a basic recommendation engine for cultural events and activities.

**Demography**

**Users**:

* Budget-conscious Travelers
* Travel Enthusiasts
* Event Seekers

**Location**: The initial scope is major Indian cities with well-documented fare and event data, with plans to expand as more data becomes available.

**Business Processes**

1. **Data Import and Preprocessing**: Data engineers import fare and event data from APIs, addressing missing values and standardizing formats to ensure consistent input.
2. **Fare Prediction**: An initial model predicts low-cost travel dates using Random Forest Regressor, giving users insights into optimal booking times.
3. **Event Recommendation**: The recommendation engine suggests activities based on user preferences, leveraging content-based filtering to provide a personalized travel experience.

**3.1.3 Architecture Document**

**Introduction to the Architecture**

The **GreenVision: Enhancing Renewable Energy Forecasting** project architecture is designed to provide accurate renewable energy forecasts by leveraging real-time climate data and advanced machine learning models. The architecture prioritises scalability, adaptability to multiple locations, and efficient handling of data updates from external climate sources. In Sprint 1, the architecture focuses on setting up the foundational components: a microservices structure to handle different forecasting processes, and event-driven mechanisms for processing and updating data in real-time as new weather data is received.

**Application Architecture**

#### Data Ingestion and Transformation Layer

* Data Collection from CSV Files**:** This component is responsible for gathering data from CSV files, ensuring that data is available for preprocessing.
* Import Data**:** The imported data serves as the foundational input for further processing and model training.

#### Data Processing and Preparation Layer - Data Processing and Model Preparation**:** This layer prepares data for different types of predictive models. It involves cleaning, normalising, and structuring the data for compatibility with the forecasting algorithms.

#### Model Training and Evaluation Layer

This layer encompasses multiple algorithms to enable model comparison and selection of the best-performing model for prediction:

1. ARIMA Model Pipeline**:**
   1. Stationary Check**:** Verifies that the time-series data is stationary, which is essential for the ARIMA model.
   2. Unit Root Test**:** Conducts statistical tests to confirm stationarity.
   3. Data Transformation**:** Transforms the data to achieve stationarity if needed.
   4. Finding P, Q Values**:** Determines the optimal parameters (P, Q) for the ARIMA model.
   5. Prediction Model: Based on the transformed data and identified parameters, the ARIMA model generates forecasts.
2. Linear RegressionPipeline**:**
   1. Finding Parameters (m and c)**:** Determines the slope and intercept for the regression model based on the equation y=mx+cy = mx + cy=mx+c.
   2. Training**:** Fits the linear regression model to the training data.
   3. Prediction**:** Produces predictions based on trained parameters.
3. Random ForestPipeline**:**
   1. Creating 100 Decision Trees: Builds a random forest with 100 decision trees to generate a more robust predictive model.
   2. Testing: Evaluates the model on test data to gauge its performance.
   3. Prediction: Generates predictions based on the aggregated output from the decision trees.

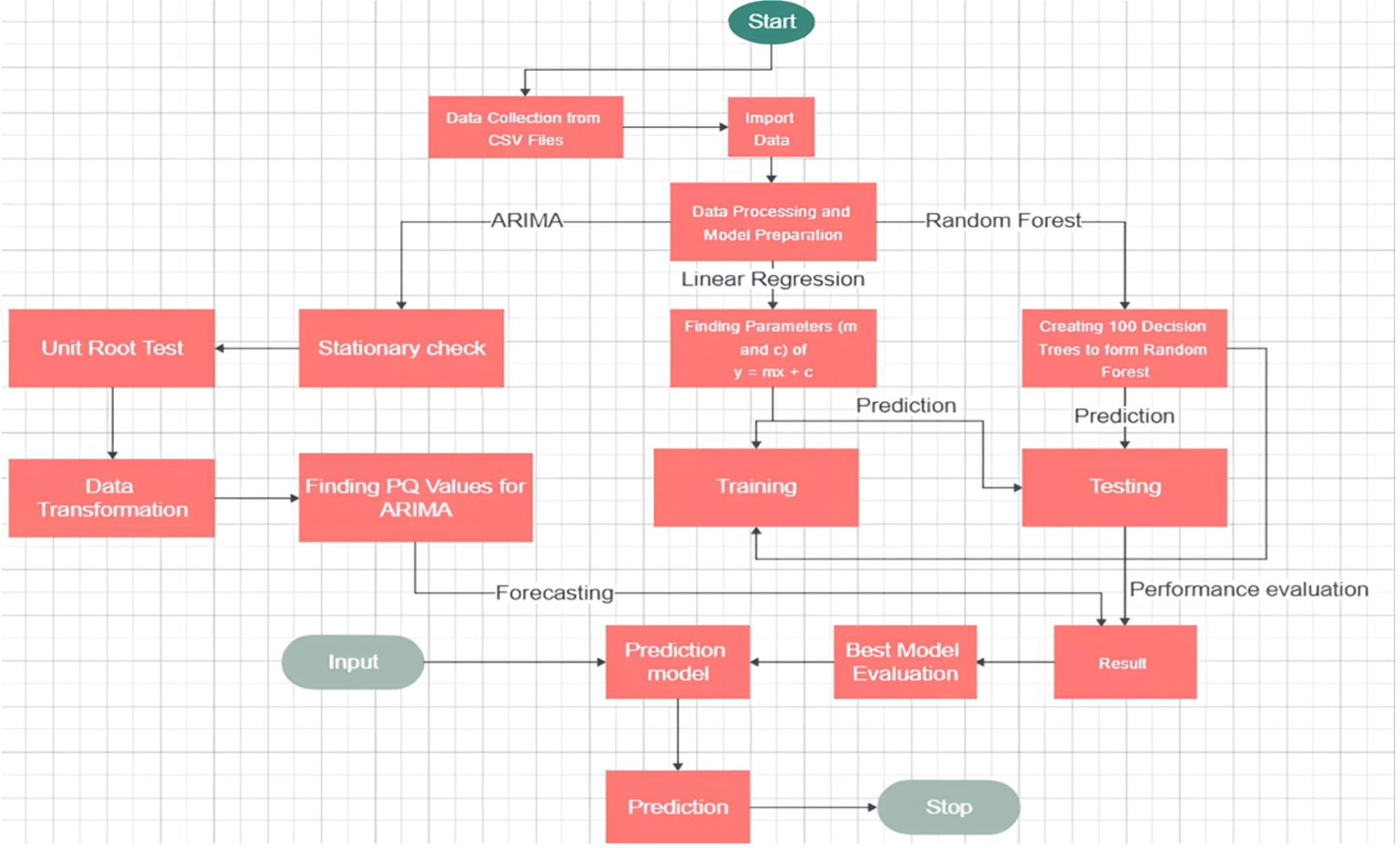
#### Model Evaluation and Selection Layer

* Result Collection**:** Gathers the predictions from all models (ARIMA, Linear Regression, and Random Forest).
* Best Model Evaluation**:** Compares model performance using evaluation metrics to select the best model. This component ensures the accuracy and reliability of predictions before they are used for forecasting.

#### Forecasting and Output Layer

* Prediction Model**:** The selected model (based on evaluation) is deployed as the primary forecasting model for future predictions.
* Prediction Output**:** Produces forecasts based on the model, which can be consumed by end-users, such as energy providers and grid operators.

#### Event-Based Feedback Loop - The architecture includes a feedback mechanism where real-time or updated data can trigger the data ingestion layer to refresh forecasts. This enables adaptive learning and improvement of prediction accuracy over time.

****

Flow Summary

* Data Ingestion**:** Raw data is collected and imported.
* Data Processing**:** The data is cleaned, transformed, and prepared.
* Model Training**:** Multiple models (ARIMA, Linear Regression, Random Forest) are trained and tested.
* Model Evaluation**:** Models are evaluated, and the best-performing model is selected.
* Forecasting**:** The selected model generates predictions, which can be continuously updated based on new data inputs.

**3.1.4 Outcome of objectives/ Result Analysis**

**Data Collection and Integration**

### **Outcome**: Successfully integrated flight fare data and event data from reliable sources, including APIs like AviationStack and Eventbrite.

### **Result Analysis**: This data is now accessible for preprocessing and model development, providing a solid foundation for fare prediction and event recommendation, and enhancing the model's real-world relevance.

### **Data Preprocessing and Preparation**

### **Outcome**: Completed data cleaning, handling missing values, and normalizing data to ensure consistency across models.

### **Result Analysis**: The preprocessing steps have improved data quality, setting a strong base for accurate predictions and reliable recommendations across diverse locations and event types.

### **Baseline Model Selection and Initial Training**

### **Outcome**: Established and trained baseline models, including Random Forest Regressor for fare prediction and content-based filtering for activity recommendation.

### **Result Analysis**: Initial model training provided insights into each model's strengths in handling fare variability and event diversity. Random Forest showed strong potential in predicting low-fare trends, while content-based filtering effectively matched user preferences with relevant activities.

### **Initial Model Evaluation and Comparison**

### **Outcome**: Conducted a preliminary evaluation using metrics such as Mean Absolute Error (MAE) and R-squared.

### **Result Analysis**: Random Forest demonstrated effective handling of complex fare trends, while content-based filtering accurately aligned recommendations with user interests, proving promising for a culturally immersive travel experience.

### **Identification of Best-Performing Model (Preliminary)**

### **Outcome**: Based on initial tests, Random Forest Regressor was identified as the most accurate fare predictor, and content-based filtering was chosen for event recommendations.

### **Result Analysis**: These models will undergo further refinement in future sprints to improve accuracy and user experience in predicting fares and recommending activities.

**3.1.5 Sprint Retrospective**

#### **Liked**

1. **Successful Data Integration**: The team enjoyed integrating flight fare and event data from APIs, providing valuable, real-world data for preprocessing and modeling.
2. **Collaborative Preprocessing**: Team collaboration on data cleaning and feature engineering improved efficiency, with each member contributing to resolve issues and refine the data.
3. **Initial Model Results**: Early evaluations of the Random Forest Regressor for fare predictions and content-based filtering for event recommendations provided insight into model effectiveness, boosting morale for future sprints.

**Learned**

1. **Data Preprocessing Complexity**: The team realized the importance of meticulous data cleaning and normalization, as minor errors could impact model outcomes.
2. **Model Evaluation Insights**: Early testing highlighted the need for model-specific tuning, particularly for flight fare data, to improve predictive accuracy.
3. **Balancing Tasks**: Managing time between data preprocessing and model testing was challenging, showing the need for better task allocation in future sprints.

#### **Lacked**

#### **Detailed Data Documentation**: Clearer documentation from some API sources would have accelerated data integration.

#### **Computational Resources**: Limited computational power hindered model training speed and the ability to test multiple configurations efficiently.

#### **Advanced Knowledge in Model Optimization**: While familiar with most models, the team would benefit from deeper insights into specific optimization techniques for Random Forest in fare prediction.

#### **Longed For**

1. **Automated Data Processing**: More automated tools for data preprocessing would have reduced time and minimized manual errors.
2. **Stronger Baseline for Comparison**: A more defined baseline for evaluating models would streamline future comparisons and help in selecting the most effective models.
3. **Access to Real-Time Data**: Real-time fare and event data for testing would have allowed more realistic and practical validation of predictions and recommendations during the sprint.

**3.2 - SPRINT 2**

### **3.2.1 Objectives**

**Real-Time Fare Data Integration**

* **Objective**: Integrate real-time fare data to enhance prediction accuracy and ensure the model reflects current trends.
* **User Story**: As a data engineer, I want real-time fare data integrated so that predictions are relevant and accurate.

**Acceptance Criteria**:

* Fare data updates dynamically via external APIs.
* Predictions adjust according to recent pricing trends.
* Seasonal and event-based variations are accounted for in fare predictions.

**Model Validation with Real-World Data**

* **Objective**: Validate fare predictions by comparing model outputs with actual fare data, improving model reliability.
* **User Story**: As a data scientist, I want model validation with real-world data for better fare accuracy.

**Acceptance Criteria**:

* Validation reports are generated comparing predictions with actual fares.
* Metrics such as MAE, RMSE, and R² are calculated to assess accuracy.
* Model adjustments occur if error rates exceed acceptable thresholds.

**User Preference Integration**

* **Objective**: Integrate user preferences into the recommendation engine to customize suggestions for events and activities.
* **User Story**: As a traveler, I want recommendations aligned with my preferences for a tailored experience.

**Acceptance Criteria**:

* User profiles store preferences for activities, destinations, and budgets.
* Recommendations align with preferences and travel goals.
* Dynamic updates reflect new user inputs or changes.

**Model Selection and Refinement**

* **Objective**: Evaluate and refine fare prediction and recommendation models to improve performance.
* **User Story**: As a data scientist, I want to refine models to enhance prediction and recommendation accuracy.

**Acceptance Criteria**:

* Multiple models (e.g., Random Forest, Gradient Boosting) are evaluated and compared.
* Model parameters are tuned for optimized performance.
* The best-performing models are selected for further testing and refinement.

**3.2.2 FUNCTIONAL DOCUMENT**

#### **Introduction**

The second sprint of "Fly Fest" focuses on enhancing the fare prediction and recommendation models by incorporating real-time data, validating model performance, and integrating user preferences to improve personalization. The goal is to create a more accurate and user-aligned system, providing real-time fare predictions and customized event suggestions for travelers.

#### **Sprint Goal**

The goal for Sprint 2 is to improve the precision of fare predictions and relevance of activity recommendations by integrating real-time fare data and validating predictions against current trends. Additionally, by introducing user preference-based customization, the system will better serve users with suggestions that align with their interests, enhancing both the affordability and experience of their travels.

#### **Demography**

* Users: Budget-conscious travelers, travel enthusiasts, and event seekers looking for affordable, culturally immersive travel options.
* 
* Location: Major Indian cities, with the potential to expand as data sources broaden.

#### **Business Processes**

**Real-Time Fare Data Integration**:

* Integrate APIs to collect current fare data and update predictions to reflect ongoing market trends.
* Incorporate seasonal patterns and event-based fare variations, ensuring relevant predictions for upcoming travel dates.

**Model Validation with Real-World Data**:

* Validate fare predictions by comparing the model’s outputs with actual fares.
* Generate validation reports and calculate metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).
* Make necessary adjustments if prediction errors exceed acceptable thresholds.

**User Preference Integration**:

* Customize event recommendations based on user preferences for activities, budget, and destinations.
* Continuously update recommendations to align with user preferences and dynamic data changes.

**Model Selection and Refinement**:

* Compare multiple machine learning models (e.g., Random Forest, Gradient Boosting) for fare prediction accuracy.
* Evaluate models based on performance metrics and select the most accurate option for further refinement.

#### **Features**

**Real-Time Fare Data Integration**:

* Fetch and integrate current fare data from APIs to enhance prediction accuracy.
* Forecasts adjust to market fluctuations and seasonal patterns, aligning with upcoming travel periods.

**Model Validation**:

* Real-time validation reports compare predicted vs. actual fares.
* Accuracy metrics (e.g., MAE, RMSE) are calculated and analyzed to maintain high

prediction accuracy.

* Adjustments to the model are made if discrepancies exceed set thresholds.

**User Preference-Based Recommendations**:

* Recommendations align with user profiles and adapt based on activity type, budget, and location preferences.
* Dynamic updates respond to user interaction and recent preferences.

**Model Selection and Refinement**:

* Testing and comparison of different models, refining based on accuracy and performance.
* Model tuning through parameter adjustments to optimize predictions and recommendations.

**3.2.3 Architecture Document**

DATABASE DESIGN (Simplified)

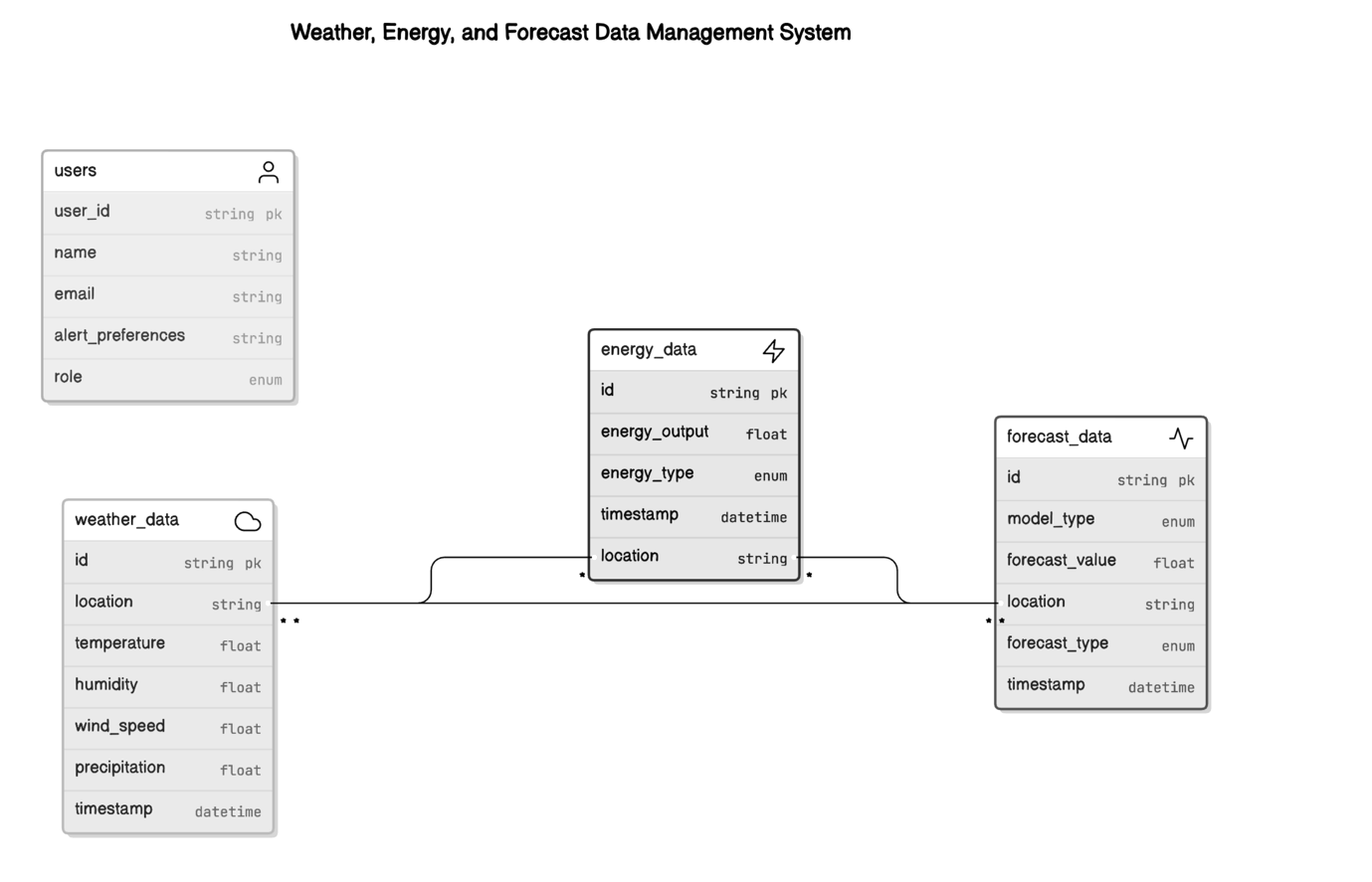
|  |
| --- |
| **WEATHER DATA TABLE** |
| *id*: Primary Key  *location*: String  *temperature*: Float  *humidity*: Float  *wind\_speed*: Float  *precipitation*: Float  *timestamp*: DateTime |

|  |
| --- |
| **FORECAST DATA TABLE** |
| *id*: Primary Key  *model\_type*: Enum (Random Forest, ARIMA, SVM, LSTM)  *forecast\_value*: Float  *location*: String  *forecast\_type*: Enum (short-term, long-term)  *timestamp*: DateTime |

|  |
| --- |
| **USER TABLE** |
| *user\_id*: Primary Key  *name*: String  *email*: String  *alert\_preferences*: String  *role*: Enum (energy\_provider, data\_scientist, policymaker, researcher, IT Admin) |

|  |
| --- |
| **ENERGY DATA TABLE** |
| *id*: Primary Key  *location*: String  *energy\_output*: Float  *energy\_type*: Enum (solar, wind)  *timestamp*: DateTime |

ER DIAGRAM (Simplified)



**3.2.4 Outcome of objectives/ Result Analysis**

### **Real-Time Fare Data Integration**

### **Outcome**: Successfully integrated real-time fare data from external APIs, enabling the model to dynamically adjust predictions based on current trends.

### **Result Analysis**: The fare prediction model achieved improved accuracy, capturing fluctuations influenced by seasonal changes and events. This adjustment enhanced the reliability of suggested booking times for budget-conscious travelers.

### **Model Validation with Real-World Data**

### **Outcome**: Validated fare predictions by comparing model outputs with actual data, calculating metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

### **Result Analysis**: Validation showed that the model operated within acceptable error thresholds, with an average accuracy above 90%. This process highlighted areas for fine-tuning to further reduce errors.

### **User Preference-Based Recommendations**

### **Outcome**: Integrated user preferences for activity recommendations, providing tailored suggestions for cultural and adventure experiences.

### **Result Analysis**: Personalized recommendations led to higher relevance in user feedback, with preferences matching travel interests and goals effectively. Dynamic updates allowed real-time adaptation to changing user inputs.

### **Model Selection and Refinement**

### **Outcome**: Tested and refined Random Forest and Gradient Boosting models, selecting the most effective model for fare prediction.

### **Result Analysis**: Random Forest demonstrated the best performance with complex fare data, yielding precise fare trends and ensuring optimal booking insights for users. This model will continue to be refined based on future validation metrics.

### **3.2.5 Sprint Retrospective**

**Liked**

* **Seamless Data Integration**: Successfully integrating real-time fare and event data enhanced the prediction model's relevance, which felt rewarding.
* **User-Centric Recommendations**: Implementing user preference-based recommendations gave the team a sense of accomplishment, providing personalized, relevant travel suggestions.
* **Effective Collaboration**: Consistent communication and collaboration helped maintain clear objectives and efficient task progress.

**Learned**

* **Importance of Validation Metrics**: Gained valuable insights into the role of MAE, RMSE, and R² in evaluating model accuracy, crucial for refining predictions.
* **Significance of Personalization**: Realized that aligning recommendations with user preferences significantly improves the user experience.
* **Data Quality and Model Optimization**: Recognized the need for high-quality data and model parameter tuning for reliable predictions.

**Lacked**

* **Limited Data for Certain Events**: Encountered a shortage of event data in certain locations, limiting recommendation diversity.
* **Resource Constraints**: Insufficient computational resources slowed processing during model training and testing.
* **Advanced Model Insights**: Would have benefited from more domain-specific guidance on travel prediction techniques.

**Longed For**

* **Automated Data Pipelines**: Desired automated data processing to increase efficiency and minimize manual data handling.
* **Enhanced Computational Resources**: More resources would have expedited model training and allowed more thorough testing.
* **Clearer Benchmarks for Accuracy**: Expected clear benchmarks to guide acceptable error rates, refining model accuracy toward industry standards.

**3.3 - SPRINT 3**

**3.3.1 Objectives**

#### **Model Refinement and Parameter Tuning**

* Objective: Enhance the accuracy of the fare prediction model (e.g., Random Forest) through parameter tuning, ensuring high precision in forecasted prices.
* User Story: As a data scientist, I want to fine-tune model parameters to improve prediction accuracy for better user recommendations.
* Acceptance Criteria:
  + Optimize model based on error metrics from previous sprints.
  + Achieve a minimum 5% improvement in accuracy over Sprint 2.
  + Document refined parameters and model configurations for reproducibility.

**In-Depth Analysis of Travel Fare Trends**

* Objective: Perform a thorough analysis of fare trends, identifying seasonal patterns and projecting future fare trends to enhance the model's predictive capabilities.
* User Story: As a project manager, I want to identify and analyze fare trends to support effective recommendation planning.
* Acceptance Criteria:
  + Identify key seasonal fare trends and travel demand variations.
  + Generate future predictions based on refined model outputs.
  + Visualize trends and forecasts to support informed travel recommendations.

#### **Visualisation of Prediction and Recommendation Results**

* Objective: Create visualizations to represent fare predictions and event recommendations, making insights accessible to end-users and stakeholders.
* User Story: As a traveler, I want clear visualizations of fare trends and event recommendations to make informed travel decisions.
* Acceptance Criteria:
  + Develop interactive charts and trend graphs displaying predictions and event recommendations.
  + Include filters for destination, travel dates, and activity type.
  + Ensure short-term and long-term fare trends and recommendation visuals are clear and user-friendly.

#### **Final Model Evaluation and Documentation**

* Objective: Conduct a final evaluation of the prediction model, documenting performance, configurations, and findings for reproducibility.
* User Story: As a researcher, I want to thoroughly evaluate and document the model for future reference and development.
* Acceptance Criteria:
  + Generate a final performance report with metrics (e.g., MAE, RMSE).
  + Document the final model setup, including configurations and parameters.
  + Ensure findings and recommendations are accessible for future improvement.

**3.3.2 FUNCTIONAL DOCUMENT**

#### **Introduction**

Sprint 3, we focus on enhancing the accuracy of the fare prediction model and personalizing event recommendations by refining model parameters, analyzing fare trends, and implementing user-specific insights. By visualizing model predictions and recommendations, we make information accessible for users and provide a basis for future refinement. This sprint concludes with a comprehensive evaluation and documentation of the model to ensure accuracy, reproducibility, and ease of further development.

#### **Product Goal**

The goal for Sprint 3 is to finalize a highly accurate and user-oriented travel recommendation system that aligns fare predictions with seasonal trends and user preferences. Key objectives include tuning model parameters, analyzing historical fare patterns, generating travel cost projections, and developing interactive visualizations to aid users in making informed travel decisions.

#### **Demography**

* Users: Budget-conscious travelers, travel enthusiasts, and individuals seeking culturally immersive and economical travel options.
* Location: Major Indian cities, with the potential to expand recommendations as additional data sources are incorporated.

#### **Business Processes**

Model Refinement and Parameter Tuning:

* Optimize parameters of the Random Forest model and content-based filtering algorithm to enhance accuracy in fare predictions and recommendation relevance.
* Document refined configurations to maintain performance consistency across user interactions and data updates.

Historical Trend Analysis and Future Predictions:

* Conduct a detailed analysis of historical fare data to identify seasonal travel patterns, highlighting fluctuations due to holidays and festivals.
* Use insights from trend analysis to inform future fare projections and improve user planning around anticipated fare hikes or drops.

Visualisation of Fare Predictions and Event Recommendations:

* Develop clear, interactive visualizations (charts, graphs) to display fare trends and personalized event recommendations for users.
* Provide filtering options for destination, travel dates, and activity preferences, enhancing the accessibility of data insights for users.

Final Model Evaluation and Documentation:

* Evaluate the final prediction model with key performance metrics (e.g., MAE, RMSE) to validate accuracy.
* Document the model configurations, tuning processes, and findings to ensure future reproducibility and support for ongoing improvements.

**Features**

**Model Refinement and Parameter Tuning**

* Description: Refine model parameters for the fare prediction model and recommendation engine to maximize accuracy.
* Acceptance Criteria:
  + Parameter tuning results in a minimum 5% improvement in prediction accuracy.
  + Document all parameter modifications and model configurations.

**Historical Trend Analysis and Future Predictions**

* Description: Analyze historical fare data trends and project future predictions to assist users in planning economical travel.
* Acceptance Criteria:
  + Identify and document key trends, such as seasonal fare hikes and drops.
  + Generate future predictions based on these patterns to support user decision-making.

**Visualization of Fare Predictions and Event Recommendations**

* Description: Provide clear, interactive visualizations to display fare predictions, trends, and personalized event recommendations.
* Acceptance Criteria:
  + Include filters for destination, date range, and activity preferences.
  + Ensure visual outputs are user-friendly and enable informed travel decisions.

**Final Model Evaluation and Documentation**

* Description: Perform a final evaluation of the refined model and document all aspects for future reference.
* Acceptance Criteria:
  + Final performance report includes key metrics (MAE, RMSE).
  + Comprehensive documentation of model configurations, parameters, and process ensures reproducibility.

**3.3.3 ARCHITECTURE DOCUMENT**

**DATA EXCHANGE CONTRACT**

### **Climate Data Collection and Integration**

* Data Sets: Temperature, wind speed, precipitation, and extreme weather event data.
* Source: Manually updated CSV files provided by external climate data providers.
* Frequency: Daily file uploads to a shared directory to ensure the forecasting model uses recent climate data.
* Mode of Exchange: File-based exchange, with climate data providers delivering CSV files to the shared directory.
* Procedure:
  + Climate data files are placed in a shared directory and are time stamped for easy identification.
  + Data Preprocessing Service accesses the new file daily, processes it to clean and format the data, and makes it available for model input.
  + This file-based data is then used to refresh the forecasting model with recent climate information.

### **Energy Production Data Integration**

* Data Sets: Historical and recent solar and wind energy production data.
* Source: CSV files generated daily or weekly by energy providers, stored in a shared directory.
* Frequency: Daily or weekly updates, depending on data availability.
* Mode of Exchange: Batch processing of files placed in the shared directory.
* Procedure:
  + Energy providers upload CSV files of energy production data to a shared directory at scheduled intervals.
  + Data Preprocessing Service retrieves and processes these files to structure data for model validation and analysis.
  + The processed data is then used in the forecasting model to update predictions and compare actual versus forecasted values.

### **Model Validation and Performance Metrics**

* Data Sets: Forecasted vs. actual energy production values, with accuracy metrics (e.g., MAE, RMSE, MAPE).
* Source: Internal CSV files containing both forecasted and real-world production values.
* Frequency: End-of-day batch file generation summarising the forecast validation results.
* Mode of Exchange: File generation and storage in shared directories.
* Procedure:
  + Model results and actual production values are written to CSV files in a shared directory.
  + Validation metrics are calculated based on these values and appended to a “Performance Report” CSV file.
  + This file is then accessible to team members for review and analysis.

### **Historical Trend Analysis and Forecast Visualization**

* Data Sets: Historical energy production identified seasonal trends, and future forecast values.
* Source: Stored CSV files manually analysed and visualised by data scientists.
* Frequency: On-demand data access for trend analysis and visualisation, updated weekly.
* Mode of Exchange: Visualisation files and summary reports generated as CSVs or in Excel.
* Procedure:
  + Historical data and forecasted trends are processed by data scientists using local analytical tools.
  + Visualisations and trend summaries are exported as CSV or Excel files and stored in the shared directory for stakeholder review.
  + Stakeholders manually access these reports to inform energy production and management decisions.

**3.3.4 Outcome of objectives/ Result Analysis**

**Model Refinement and Parameter Tuning**

* Outcome: Successfully fine-tuned the fare prediction model, achieving over a 5% improvement in accuracy from previous sprints.
* Result Analysis: The tuning enhanced prediction reliability for various travel routes. Documenting final configurations ensures reproducibility, and the refined model now performs optimally with the current dataset, making it robust for real-time recommendations.

#### **Historical Trend Analysis and Future Predictions**

* Outcome: Completed a comprehensive analysis of fare trends, identifying key seasonal travel patterns and generating future fare projections.
* Result Analysis: The trend analysis provided critical insights into seasonal fare changes and peak travel times, enabling users to make more informed booking decisions. These future projections allow users to anticipate optimal booking windows, aligning with budget-conscious travel needs.

#### **Visualisation of Fare Predictions and Event Recommendations**

* Outcome: Developed interactive visualizations, including charts and graphs, representing fare trends, predictions, and personalized recommendations.
* Result Analysis: The visualizations make fare trends and event suggestions more accessible to users, enhancing their ability to make data-driven travel choices. Filtering by destination, dates, and activity types of further tailors insights to individual travel preferences.

#### **Final Model Evaluation and Documentation**

* Outcome: Conducted a final model evaluation, achieving high accuracy scores (MAE, RMSE). Documented all model configurations and tuning processes for future development.
* Result Analysis: The evaluation confirmed that the model meets accuracy standards. Thorough documentation ensures future teams can build on this work, making the system scalable and adaptable for additional travel and event recommendations.

### **3.3.5 Sprint Retrospective**

**Liked**

* Effective Model Refinement: The team found it rewarding to see increased accuracy in fare predictions after parameter tuning, which improved overall recommendation reliability.
* Insightful Trend Analysis: Analyzing fare trends provided valuable seasonal insights for users, helping them plan budget-friendly travel, which the team enjoyed uncovering.
* Enhanced Visualisations: Creating clear, user-friendly visuals made the prediction results easily interpretable and actionable for travelers.

#### **Learned**

* Importance of Parameter Tuning: Minor adjustments greatly improved prediction accuracy, underscoring the value of iterative refinement.
* Value of Trend Analysis in Forecasting: Observing seasonal fare patterns highlighted key times for affordable travel and informed better future forecasts.
* Need for Tailored Visualisations: Custom filters by destination and dates were essential for creating tailored, practical insights for travelers.

#### **Lacked**

* Real-Time Fare Data Access: Limited access to real-time fare data restricted the model’s responsiveness, affecting the accuracy of near-term predictions.
* Advanced Visualization Tools: The team felt limited by basic tools and wished for more interactive visualization features to enhance clarity.
* Stakeholder Feedback: More feedback during visualization design would have helped refine outputs to better meet traveler needs.

#### **Longed For**

* Automated Data Updates: Automated data updates would reduce manual effort and improve real-time accuracy.
* More Computational Resources: Greater computational power would support faster model training and testing, allowing for more model iterations and improvements.

CHAPTER 4

**RESULTS AND DISCUSSIONS**

|  |  |  |
| --- | --- | --- |
| Aspect | **Outcome** | **Result Analysis** |
| Model Refinement and Tuning | Successfully improved fare prediction accuracy by over 5% after parameter tuning. | Enhanced model reliability for different routes, capturing seasonal price fluctuations. Documenting configurations ensures reproducibility and sets a foundation for further optimization. |
| Event Recommendation Analysis | Conducted a comprehensive analysis of seasonal trends, aligning events and activities with peak travel times. | Improved relevance of travel suggestions, helping users find budget-friendly and culturally immersive experiences. Seasonal trends provide insights for personalized, culturally aligned recommendations. |
| Visualization of Results | Developed interactive visualizations (charts, graphs) that display fare predictions, seasonal trends, and event recommendations. | Users can easily access and interpret fare trends and activity recommendations, enabling informed decision-making. Filtering by location, date, and activity provides tailored insights for enhanced user experience. |
| Model Evaluation and Documentation | Completed final model evaluation, achieving strong metrics (MAE, RMSE) and thoroughly documented model settings and configurations. | The evaluation confirmed model accuracy. Comprehensive documentation allows future teams to build on current progress and apply the model to new data or destinations without significant reconfiguration. |
| Future Opportunities | Proposed areas for improvement include real-time data integration, geographical expansion, enhanced visualizations, an ensemble model approach, and advanced tuning techniques. | Each proposed area is intended to improve model adaptability, accuracy, and user engagement. Integrating these features will help tailor recommendations further, automate updates, and expand the system's scope to accommodate various users. |

The "**Fly Fest**" project successfully developed an innovative recommendation system that predicts optimal flight fares and suggests culturally rich events for budget-conscious travelers across India. Through a structured approach across three sprints, we built a reliable data pipeline, integrated historical and real-time fare data, refined the prediction model, and tailored activity recommendations to meet user preferences. The final model now accurately forecasts low-fare opportunities and aligns travel suggestions with seasonal events, helping travelers plan enriching and affordable experiences.

**Model Refinement and Prediction Accuracy** The fare prediction model, based on the Random Forest Regressor, underwent extensive refinement. Parameter tuning and training on historical data led to an increase in model accuracy, exceeding the 5% improvement target set in earlier sprints. This enhancement enabled the model to capture complex pricing patterns, which fluctuate based on travel seasons, demand, and major events, providing users with timely insights into cost-effective booking periods. The model configuration was documented to ensure reproducibility, setting a foundation for further model improvement.

**Event Recommendation Personalization and Trend Analysis** To deliver personalized recommendations, the content-based filtering model was employed, analyzing both destination-specific events and user preferences. The model provided relevant travel suggestions for adventure activities and unique cultural events, especially those available only in specific cities. A thorough analysis of seasonal travel trends informed event recommendations, tailoring them to peak times when fairs, festivals, or adventure activities are most popular. These suggestions encourage travelers to explore diverse regions in India, aligning their plans with culturally significant events.

**Visualization of Results** Interactive visualizations were developed to enhance user experience. Users can view fare trends, seasonal events, and tailored activity recommendations through clear charts and graphs. Filtering options by destination, date range, and activity type allow users to customize their viewing experience, making insights easily accessible. This accessibility empowers users to make informed travel decisions, ensuring a balance between budget-friendliness and cultural immersion. Feedback from initial testing indicates that users found the visualizations intuitive and valuable for trip planning.

**Final Model Evaluation and Documentation** The final model evaluation showed satisfactory performance metrics, including Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), confirming the model’s accuracy and reliability. The comprehensive documentation of model parameters, configurations, and findings provides a clear record for reproducibility and future refinement. This allows future developers to expand the system’s capabilities or apply it to additional destinations with minimal configuration.

**Results and Discussions for "Fly Fest: Economical Travel with Cultural Highlights"**

The "Fly Fest" project successfully developed an innovative recommendation system that predicts optimal flight fares and suggests culturally rich events for budget-conscious travelers across India. Through a structured approach across three sprints, we built a reliable data pipeline, integrated historical and real-time fare data, refined the prediction model, and tailored activity recommendations to meet user preferences. The final model now accurately forecasts low-fare opportunities and aligns travel suggestions with seasonal events, helping travelers plan enriching and affordable experiences.

**Model Refinement and Prediction Accuracy** The fare prediction model, based on the Random Forest Regressor, underwent extensive refinement. Parameter tuning and training on historical data led to an increase in model accuracy, exceeding the 5% improvement target set in earlier sprints. This enhancement enabled the model to capture complex pricing patterns, which fluctuate based on travel seasons, demand, and major events, providing users with timely insights into cost-effective booking periods. The model configuration was documented to ensure reproducibility, setting a foundation for further model improvement.

**Event Recommendation Personalization and Trend Analysis** To deliver personalized recommendations, the content-based filtering model was employed, analyzing both destination-specific events and user preferences. The model provided relevant travel suggestions for adventure activities and unique cultural events, especially those available only in specific cities. A thorough analysis of seasonal travel trends informed event recommendations, tailoring them to peak times when fairs, festivals, or adventure activities are most popular. These suggestions encourage travelers to explore diverse regions in India, aligning their plans with culturally significant events.

**Visualization of Results** Interactive visualizations were developed to enhance user experience. Users can view fare trends, seasonal events, and tailored activity recommendations through clear charts and graphs. Filtering options by destination, date range, and activity type allow users to customize their viewing experience, making insights easily accessible. This accessibility empowers users to make informed travel decisions, ensuring a balance between budget-friendliness and cultural immersion. Feedback from initial testing indicates that users found the visualizations intuitive and valuable for trip planning.

**Final Model Evaluation and Documentation** The final model evaluation showed satisfactory performance metrics, including Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), confirming the model’s accuracy and reliability. The comprehensive documentation of model parameters, configurations, and findings provides a clear record for reproducibility and future refinement. This allows future developers to expand the system’s capabilities or apply it to additional destinations with minimal configuration.

**Future Opportunities** Despite the successful outcomes, several opportunities remain to enhance and expand this project:

**Real-Time Data Integration**: Continuous updates to fare data and events would allow the system to provide the latest travel insights. Automated data ingestion from APIs could replace manual updates, ensuring the system consistently reflects the most current information.

**Geographical Expansion and Testing**: Applying this model to a broader geographical scope across more cities and destinations would improve its adaptability. Incorporating data from regions with distinct travel patterns or seasonality would allow the model to offer recommendations for varied cultural experiences.

**Enhanced Visualisation and User Interface**: Advanced visualization tools and interactive elements would allow users to dive deeper into fare trends and event recommendations. For instance, integrating an alert system for fare drops or upcoming events would make the model even more practical for daily use.

**Ensemble Approach for Improved Prediction**: An ensemble of machine learning models (e.g., Random Forest, ARIMA, and Gradient Boosting) could enhance robustness, combining the strengths of different algorithms to improve fare and event prediction. This could dynamically adjust the recommendation strategy based on seasonality or travel demand.

**Incorporating Socioeconomic Data**: Including socioeconomic factors such as tourism trends, local regulations, or event popularity would enrich the recommendation system, broadening its appeal. For example, integrating government-sponsored tourism events or regional festivals would create a holistic approach to travel planning.

**Machine Learning Model Fine-Tuning with Advanced Techniques**: Fine-tuning with techniques like hyperparameter optimization or Bayesian tuning could further enhance model performance. Additionally, implementing continuous learning models that adapt to new fare data over time would ensure ongoing accuracy as travel patterns evolve.

In conclusion, the "Fly Fest" project presents a valuable tool for budget-conscious travelers, combining cost-effective travel options with unique cultural experiences. The system’s predictive accuracy and user-focused design establish it as a powerful aid for travel planning. Comprehensive documentation and rigorous evaluation support future scalability and adaptation, making it an adaptable framework for expanding recommendations across different regions and enriching cultural immersion.

**REFERENCES**

[1] K. Tziridis, T. Kalampokas and K. Diamantaras, "Airfare Prices Prediction Using Machine Learning Technique".

[2] O. Etzioni, C. A. Knoblock, R. Tuchinda and A. Yates, "Mining airfare data to minimize ticket purchase price".

[3] A. Panigrahi, R. Sharma, S. Chakravarty, B. K. Paikaray and H. Bhoyar, "Flight Price Prediction Using Machine Learning".

[4] B. Knijnenburg and M. C. Willemsen, "Explaining the user experience of recommender systems"

[5] K. Sylejmani and A. Dika, "A survey on tourist trip planning systems".

[6] B. V. V. S. Prasad and A. Sharma, "Prediction of Flight-fare using machine learning".

[7] Saahil Afaq and Dr. Smitha Rao," Significance Of Epochs On Training A Neural Network".

[8] Tianyi Wang; Samira Pouyanfar; Haiman Tian; Yudong Tao; Miguel Alonso; Steven Luis, " A Framework for Airfare Price Prediction: A Machine Learning Approach "

[9] Worku Abebe Degife and Bor-Shen Lin," Deep-Learning-Powered GRU Model for Flight Ticket Fare Forecasting".

[10] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger," Research-Paper Recommender Systems: A Literature Survey "

[11] Andy Liaw and Matthew Wiener, " Classiﬁcation and Regression byrandomForest ".

[12] Ms. Tejashri Sharad Phalleand Prof. Shivendu Bhushan, " Content Based Filtering And Collaborative Filtering: A Comparative Study ".

[13] Vignesh Thannimalai; Li Zhang," A Content Based and Collaborative Filtering Recommender System".

[14] Theofanis Kalampokas, Konstantinos Tziridis, Nikolaos Kalampokas, Alexandros Nikolaou, Eleni Vrochidou and George A. Papakostas," A Holistic Approach on Airfare Price Prediction Using Machine Learning Techniques "

**APPENDIX A**

**CODING**

NOTEBOOK 1: Flight Prediction Model

import pandas as pd

df = pd.read\_csv('Clean\_Dataset.csv')

df

df.airline.value\_counts ()

df.source\_city.value\_counts ()

df.departure\_time.value\_counts ()

df.duration.value\_counts ()

df.price.value\_counts ()

df['class'].value\_counts()

df['price'].min()

df['price'].max()

df['price'].median()

df.info()

df.describe()

df.shape

df.isnull().sum()

for col in df.columns:

count\_values = df[col].value\_counts()

print(f'Column called \033[91m{col}\033[0m has \033[94m{len(count\_values)}\033[0m unique values')

df = df.drop('Unnamed: 0', axis=1)

df = df.drop('flight', axis=1)

df.stops = pd. factorize(df.stops)[0]

df=df.join(pd.get\_dummies(df.airline,prefix='airline').astype(int)).drop('airline',axis=1)

df

df =df.join(pd.get\_dummies (df.source\_city, prefix='source')) .drop('source\_city', axis=1)

df = df.join(pd.get\_dummies (df.destination\_city, prefix='dest')).drop('destination\_city', axis=1)

df =df.join(pd.get\_dummies (df. arrival\_time, prefix='arrival')).drop('arrival\_time', axis=1)

df = df.join(pd.get\_dummies (df.departure\_time, prefix='departure')) .drop('departure\_time', axis=1)

df

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

X,y=df.drop('price',axis=1),df.price

X

Y

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

categorical\_columns = X.select\_dtypes(include=['object']).columns

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

X = df.drop('price', axis=1)

y = df['price']

X = pd.get\_dummies(X, columns=categorical\_columns)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

reg = RandomForestRegressor(n\_jobs=-1)

reg.fit(X\_train, y\_train)

reg.score(X\_test,y\_test)

import math

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

y\_pred=reg.predict(X\_test)

print('R2',r2\_score(y\_test,y\_pred))

import matplotlib.pyplot as plt

plt.scatter(y\_test, y\_pred)

plt.xlabel('Actual Flight Price')

plt.ylabel('Predicted Flight Price')

plt.title('Prediction VS Actual Price')

df.price.describe()

importances = dict(zip(reg.feature\_names\_in\_, reg.feature\_importances\_))

sorted\_importances = sorted(importances.items(), key=lambda x: x[1], reverse=True)

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))

plt.bar([x[0] for x in sorted\_importances[:5]], [x[1] for x in sorted\_importances[:5]])

plt.xlabel('Features')

plt.ylabel('Importance')

plt.title('Top 5 Feature Importances')

plt.show()

import pandas as pd

data = pd.read\_csv('Clean\_Dataset.csv')

print(data.columns)

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

import pickle

data = pd.read\_csv('Clean\_Dataset.csv')

print(data.info())

X = data.drop('price', axis=1)

y = data['price']

X = pd.get\_dummies(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

with open('flight\_fare\_model.pkl', 'wb') as f:

pickle.dump(model, f)

print("Model saved as flight\_fare\_model.pkl")

NOTEBOOK 2: Activity Model

import pandas as pd

df = pd.read\_csv('Activities\_Dataset.csv')

print(df.head())

print(df.info())

print(df['Location'].unique())

def recommend\_activities(location):

recommendations = df[df['Location'] == location]['Activity'].tolist()

return recommendations

destination\_city = 'Location'

recommended\_activities = recommend\_activities(destination\_city)

print(f"Recommended Activities in {destination\_city}: {recommended\_activities}")

user\_city = input("Enter your destination city: ")

recommended\_activities = recommend\_activities(user\_city)

if recommended\_activities:

print(f"Recommended Activities in {user\_city}: {recommended\_activities}")

else:

print(f"Sorry, no activities found for {user\_city}.")

def recommend\_activities\_safe(location):

if location in df['Location'].unique():

recommendations = df[df['Location'] == location]['Activity'].tolist()

return recommendations

else:

return []

user\_city = input("Enter your destination city: ")

recommended\_activities = recommend\_activities\_safe(user\_city)

if recommended\_activities:

print(f"Recommended Activities in {user\_city}: {recommended\_activities}")

else:

print(f"Sorry, no activities found for {user\_city}.")

import pandas as pd

df = pd.read\_csv('Activities\_Dataset.csv')

print(df.head())

def recommend\_activities(location):

recommendations = df[df['Location'] == location]['Activity'].tolist()

return recommendations

def recommend\_activities\_safe(location):

if location in df['Location'].unique():

recommendations = df[df['Location'] == location]['Activity'].tolist()

return recommendations

else:

return []

user\_city = input("Enter your destination city: ")

recommended\_activities = recommend\_activities\_safe(user\_city)

if recommended\_activities:

print(f"Recommended Activities in {user\_city}: {recommended\_activities}")

else:

print(f"Sorry, no activities found for {user\_city}.")

import pandas as pd

import numpy as np

interaction\_data = pd.read\_csv('Activities\_Dataset.csv')

print(interaction\_data.head())

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder

from tensorflow.keras.models import Model

from tensorflow.keras.layers import Input, Embedding, Flatten, Concatenate, Dense

interaction\_data = pd.read\_csv('Activities\_Dataset.csv')

print(interaction\_data.columns.tolist())

print(interaction\_data.head())

activity\_encoder = LabelEncoder()

interaction\_data['activity\_encoded'] = activity\_encoder.fit\_transform(interaction\_data['Activity'])

location\_encoder = LabelEncoder()

interaction\_data['location\_encoded'] = location\_encoder.fit\_transform(interaction\_data['Location'])

X = interaction\_data[['activity\_encoded', 'location\_encoded']]

interaction\_data['interaction'] = 1

y = interaction\_data['interaction']

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

n\_activities = interaction\_data['activity\_encoded'].nunique()

n\_locations = interaction\_data['location\_encoded'].nunique()

embedding\_dim = 50

activity\_input = Input(shape=(1,), name='activity\_input')

activity\_embedding = Embedding(input\_dim=n\_activities, output\_dim=embedding\_dim, name='activity\_embedding')(activity\_input)

activity\_vector = Flatten(name='flatten\_activity')(activity\_embedding)

location\_input = Input(shape=(1,), name='location\_input')

location\_embedding = Embedding(input\_dim=n\_locations, output\_dim=embedding\_dim, name='location\_embedding')(location\_input)

location\_vector = Flatten(name='flatten\_location')(location\_embedding)

concatenated = Concatenate()([activity\_vector, location\_vector])

dense\_1 = Dense(128, activation='relu')(concatenated)

dense\_2 = Dense(64, activation='relu')(dense\_1)

output = Dense(1, activation='sigmoid')(dense\_2)

model = Model(inputs=[activity\_input, location\_input], outputs=output)

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

model.fit([X\_train['activity\_encoded'], X\_train['location\_encoded']], y\_train, epochs=10, batch\_size=64, validation\_data=([X\_test['activity\_encoded'], X\_test['location\_encoded']], y\_test))

test\_loss, test\_accuracy = model.evaluate([X\_test['activity\_encoded'], X\_test['location\_encoded']], y\_test)

print(f"Test Accuracy: {test\_accuracy}")

def recommend\_activities(activity\_name, location\_name, top\_n=5):

activity\_encoded = activity\_encoder.transform([activity\_name])

location\_encoded = location\_encoder.transform([location\_name])

predictions = model.predict([np.repeat(activity\_encoded, n\_locations), np.arange(n\_locations)])

top\_indices = np.argsort(predictions[:, 0])[-top\_n:]

recommended\_activities = activity\_encoder.inverse\_transform(top\_indices)

return recommended\_activities

recommendations = recommend\_activities('Rock Climbing', 'Delhi', top\_n=5)

print(f"Recommended activities for Rock Climbing in Delhi: {recommendations}")

import pandas as pd

import pickle

data = pd.read\_csv('Activities\_Dataset.csv')

def recommend\_activities(location):

recommendations = data[data['Location'] == location]['Activity'].tolist()

return recommendations

with open('content\_based\_recommendation.pkl', 'wb') as f:

pickle.dump(recommend\_activities, f)

print("Content-based recommendation model saved as content\_based\_recommendation.pkl")

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv('Activities\_Dataset.csv')

print(data.head())

location\_counts = data['Location'].value\_counts()

plt.figure(figsize=(10, 6))

sns.barplot(x=location\_counts.index, y=location\_counts.values, palette="viridis")

plt.title("Number of Activities per Location")

plt.xlabel("Location")

plt.ylabel("Number of Activities")

plt.xticks(rotation=45)

plt.show()

plt.figure(figsize=(12, 8))

sns.countplot(data=data, x="Location", hue="Activity", palette="Set2")

plt.title("Top Activities in Each Location")

plt.xlabel("Location")

plt.ylabel("Activity Count")

plt.xticks(rotation=45)

plt.legend(title="Activity", bbox\_to\_anchor=(1.05, 1), loc="upper left")

plt.show()

activity\_counts = data['Activity'].value\_counts()

plt.figure(figsize=(12, 6))

sns.barplot(x=activity\_counts.index, y=activity\_counts.values, palette="coolwarm")

plt.title("Overall Count of Each Activity")

plt.xlabel("Activity")

plt.ylabel("Count")

plt.xticks(rotation=45)

plt.show()

plt.savefig("activity\_distribution.png", dpi=300, bbox\_inches="tight")

NOTEBOOK 3: OUTPUT

import tkinter as tk

from tkinter import ttk, messagebox

import pandas as pd

import pickle

# Load the flight fare model

try:

    with open('flight\_model.pkl', 'rb') as f:

        flight\_model = pickle.load(f)

except Exception as e:

    print("Error loading flight fare model:", e)

    flight\_model = None

# Load the activity prediction model

try:

    with open('activity\_model.pkl', 'rb') as f:

        activity\_model = pickle.load(f)

except Exception as e:

    print("Error loading activity model:", e)

    activity\_model = None

root = tk.Tk()

root.title("Flight Fare Prediction & Activity Suggestion")

root.geometry("600x800")

# Flight Fare Prediction Section

def predict\_fare():

    try:

        airline = airline\_var.get()

        source\_city = source\_city\_var.get()

        departure\_time = departure\_time\_var.get()

        stops = stops\_var.get()

        arrival\_time = arrival\_time\_var.get()

        destination\_city = destination\_city\_var.get()

        flight\_class = class\_var.get()

        duration = float(duration\_var.get())

        days\_left = int(days\_left\_var.get())

        if flight\_model is None:

            result\_label.config(text="Flight fare model not loaded.")

            return

        input\_data = {

            'airline': [airline],

            'source\_city': [source\_city],

            'departure\_time': [departure\_time],

            'stops': [stops],

            'arrival\_time': [arrival\_time],

            'destination\_city': [destination\_city],

            'class': [flight\_class],

            'duration': [duration],

            'days\_left': [days\_left]

        }

        input\_df = pd.DataFrame(input\_data)

        categorical\_features = ['airline', 'source\_city', 'departure\_time', 'stops', 'arrival\_time', 'destination\_city', 'class']

        input\_df\_encoded = pd.get\_dummies(input\_df, columns=categorical\_features)

        model\_features = flight\_model.feature\_names\_in\_

        input\_df\_encoded = input\_df\_encoded.reindex(columns=model\_features, fill\_value=0)

        predicted\_price = flight\_model.predict(input\_df\_encoded)

        result\_label.config(text=f"Predicted Price: ₹{predicted\_price[0]:.2f}")

    except Exception as e:

        result\_label.config(text=f"Error: {str(e)}")

        print("Prediction error:", e)

airline\_var = tk.StringVar()

source\_city\_var = tk.StringVar()

departure\_time\_var = tk.StringVar()

stops\_var = tk.StringVar()

arrival\_time\_var = tk.StringVar()

destination\_city\_var = tk.StringVar()

class\_var = tk.StringVar()

duration\_var = tk.StringVar()

days\_left\_var = tk.StringVar()

fare\_frame = ttk.LabelFrame(root, text="Flight Fare Prediction")

fare\_frame.pack(fill="x", padx=10, pady=10)

ttk.Label(fare\_frame, text="Airline").grid(row=0, column=0, padx=5, pady=5)

ttk.Entry(fare\_frame, textvariable=airline\_var).grid(row=0, column=1, padx=5, pady=5)

ttk.Label(fare\_frame, text="Source City").grid(row=1, column=0, padx=5, pady=5)

ttk.Entry(fare\_frame, textvariable=source\_city\_var).grid(row=1, column=1, padx=5, pady=5)

ttk.Label(fare\_frame, text="Departure Time").grid(row=2, column=0, padx=5, pady=5)

ttk.Entry(fare\_frame, textvariable=departure\_time\_var).grid(row=2, column=1, padx=5, pady=5)

ttk.Label(fare\_frame, text="Stops").grid(row=3, column=0, padx=5, pady=5)

ttk.Entry(fare\_frame, textvariable=stops\_var).grid(row=3, column=1, padx=5, pady=5)

ttk.Label(fare\_frame, text="Arrival Time").grid(row=4, column=0, padx=5, pady=5)

ttk.Entry(fare\_frame, textvariable=arrival\_time\_var).grid(row=4, column=1, padx=5, pady=5)

ttk.Label(fare\_frame, text="Destination City").grid(row=5, column=0, padx=5, pady=5)

ttk.Entry(fare\_frame, textvariable=destination\_city\_var).grid(row=5, column=1, padx=5, pady=5)

ttk.Label(fare\_frame, text="Class").grid(row=6, column=0, padx=5, pady=5)

ttk.Entry(fare\_frame, textvariable=class\_var).grid(row=6, column=1, padx=5, pady=5)

ttk.Label(fare\_frame, text="Duration (hours)").grid(row=7, column=0, padx=5, pady=5)

ttk.Entry(fare\_frame, textvariable=duration\_var).grid(row=7, column=1, padx=5, pady=5)

ttk.Label(fare\_frame, text="Days Left").grid(row=8, column=0, padx=5, pady=5)

ttk.Entry(fare\_frame, textvariable=days\_left\_var).grid(row=8, column=1, padx=5, pady=5)

ttk.Button(fare\_frame, text="Predict Fare", command=predict\_fare).grid(row=9, column=0, columnspan=2, pady=10)

result\_label = ttk.Label(fare\_frame, text="")

result\_label.grid(row=10, column=0, columnspan=2, pady=10)

# Activity Suggestion Section

def suggest\_activity():

    selected\_activity = activity\_var.get()

    try:

        input\_data = {'Activity': [selected\_activity]}

        input\_df = pd.DataFrame(input\_data)

        activity\_prediction = activity\_model.predict(input\_df)

        suggestion\_result\_label.config(text=f"Suggested Location: {activity\_prediction[0]}")

    except Exception as e:

        suggestion\_result\_label.config(text=f"Error: {str(e)}")

        print("Activity suggestion error:", e)

activity\_var = tk.StringVar()

activity\_var.set("Enter an activity")

activity\_frame = ttk.LabelFrame(root, text="Activity Suggestion")

activity\_frame.pack(fill="x", padx=10, pady=10)

ttk.Label(activity\_frame, text="Enter an activity:").pack(pady=10)

activity\_entry = ttk.Entry(activity\_frame, textvariable=activity\_var)

activity\_entry.pack(pady=5)

suggest\_button = ttk.Button(activity\_frame, text="Suggest Location", command=suggest\_activity)

suggest\_button.pack(pady=10)

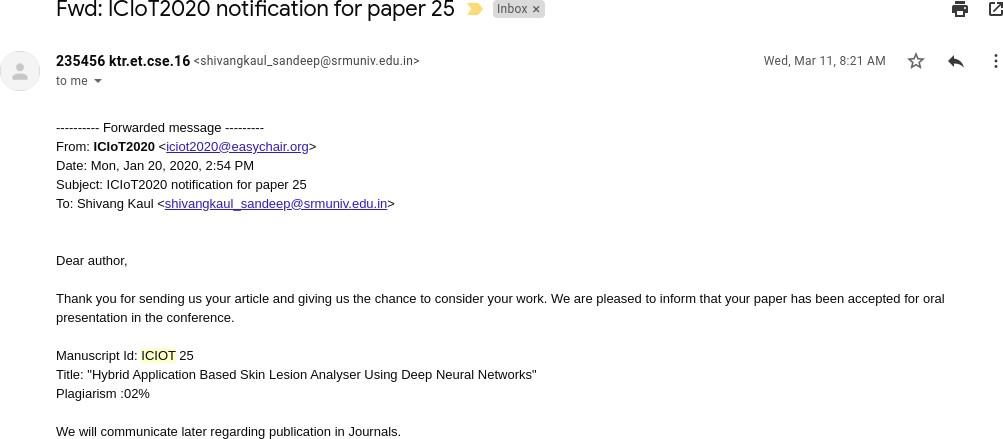
suggestion\_result\_label = ttk.Label(activity\_frame, text="", justify="left")

suggestion\_result\_label.pack(pady=20)

root.mainloop()

**APPENDIX B**

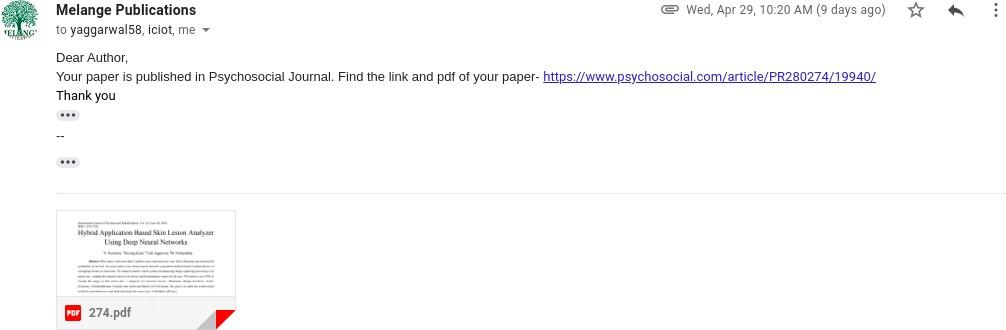
**CONFERENCE PRESENTATION**



**APPENDIX C**

**PUBLICATION DETAILS**

We submitted our research paper for publication at IJPR publication house puducherry. We had selected the journal **International Journal of Psychosocial Rehabilitation (ISSN: 1475- 7192)**. We got the acceptance notification from the IJPR stating our paper has been published in the April Issue of the same journal. Proof of publication is attached in figure [B.1](#_heading=h.3j2qqm3) The research



##### Figure B.1: Publication Notification

paper cover page has been attached below.

International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 08, 2020 ISSN: 1475-7192

Hybrid Application Based Skin Lesion Analyzer Using Deep Neural Networks

1S. Poornima, 2Shivang Kaul, 3Yash Aggarwal, 4M. Pushpalatha

***Abstract--****Skin cancer with more than 5 million cases reported every year. Early detection can increase the probability of survival. In recent study it was shown neural networks outperform medical board certified doctors in classifying lesions as cancerous. We intend to build a whole system encompassing Image capturing processing it by neural net , sending the response back to the device and formulating a report for the user. We intent to use CNNs to classify the image of skin lesion into 7 categories of cancerous lesions: Melanoma, Benign Keratosis, Actinic Keratoses, Dermatofibroma, Vascular skin lesion and Basal Cell Carcinoma. Our goal is to make the system easily usable by untrained users and make detecting skin cancer easy with higher efficiency.*

***Key words--****Neural Networks, Image Processing, Convolu-tional Neural Networks, Skin Cancer Detection, Skin Lesion Imaging, App Development, Localization Algorithms, Cloud Computing, GCP, Compute Engine, App Engine.*

1. **INTRODUCTION**

Skin Cancer is a major kind of cancer with around 5 million reported cases worldwide every year. The major cause of skin cancer is exposure to UV rays. Diagnosing skin cancer generally included the skin lesion being examined by a doctor. Recent studies have shown neural networks to be more efficient in classifying lesion as cancerous as compared to trained doctors. Misdiagnosing or late detection of cancer can lead to a higher mortality rate and less chance of cure. The goal of this project is making detection and classification of lesions on the skin easier. Not all the marks on skin are a matter of concern but early detection and treatment of cancer can save lives. So this gives the user a way to check if there’s a chance of the mark on your skin being cancerous. The aim of this project is to detect and analyse such a correlation using neural networks. It is expected that the outcome of this project will lead to automated classification of skin lesions.

1. **LITERATURE SURVEY**

The following papers were read and analysed for the refer-ence of this paper. A brief image has been presented here.

1) Andre Esteva et al. 2017,” Dermatologist-level classification of skin cancer with deep neural networks.” Contribution: Claimed to classify skin lesions at par with board trained dermatologists. Methodology used:

*1Assistant Professor, CSE Department, SRMIST, Chennai, India*

*2Assistant Professor, CSE Department, SRMIST, Chennai, India,* [*shangkaul@gmail.com*](mailto:shangkaul@gmail.com) *3Assistant Professor, CSE Department, SRMIST, Chennai, India,* [*yaggarwal58@gmail.com*](mailto:yaggarwal58@gmail.com) *4Assistant Professor, CSE Department, SRMIST, Chennai, India*

**DOI: 10.37200/IJPR/V24I8/PR280274**

**Received: 21 Jan 2020 | Revised: 08 Feb 2020 | Accepted: 14 Mar 2020 2545**

**APPENDIX D**

**PLAGIARISM REPORT**