SIMATS ENGINEERING

Engineer to Excel

SIMATS

Saveetha Institute of Medical And Technical Sciences

(Declared as Deemed to be University under Section 3 of UGC Act 1956)

DEVELOPING A FLIGHT PRICE PREDICTION MODEL USING MACHINE LEARNING AND DATA WAREHOUSING TECHNIQUES

A CAPSTONE PROJECT REPORT

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

GIRIJA B (192311044)

Data Warehousing & Data Mining

with Detection and Extraction - CSA1618

Under the Supervision of

Dr. P. SUBRAMANIAN

March - 2025

BONAFIDE CERTIFICATE

I, **GIRIJA B** student of Department Computer Science and Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the work presented in this Capstone Project entitled **Developing a Flight Price Prediction Model Using Machine Learning and Data Warehousing Techniques.** is the outcome of our own Bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics.

Date:	Student Name:
Place:	Reg.No:

Internal Examiner **External Examiner**

ABSTRACT

The growing demand for affordable air travel and the dynamic nature of airline pricing have created a significant need for predictive systems that can estimate flight ticket prices with high accuracy. This project focuses on developing a Flight Price Prediction system using Machine Learning technique, specifically Random Forest Regression, integrated with data warehousing and preprocessing pipelines. The primary problem addressed is the unpredictability in airfare pricing due to multiple influencing factors such as booking time, airline, source, destination, number of stops, and flight duration. The goal is to build an intelligent model that can assist users in identifying optimal booking windows and travel plans based on predicted pricing trends.

The project leverages a dataset containing historical flight information, including categorical and numerical attributes. Data preprocessing techniques such as null value handling, label encoding, feature scaling, and outlier removal were applied to prepare the data for modeling. The system architecture includes a model training pipeline, an API layer developed using Flask, and a Streamlit-based web interface for real-time predictions.

Key deliverables of the project include a trained ML model capable of predicting prices based on user input, a RESTful API for integration, and an interactive UI for users to get instant fare predictions. It also underlines the importance of data engineering tasks like ETL (Extract, Transform, Load) and dimensional modeling in building scalable, production-ready ML solutions.The project highlights the real-world application of machine learning in the travel industry and offers a foundation for future enhancements such as real-time fare tracking, dynamic feature updates, or integration with airline APIs for live data. By bridging the gap between raw data and user decision-making, this work contributes to the domain of predictive analytics and intelligent travel planning.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and thanks to my supervisor, **Dr. P. SUBRAMANIAN,** for their continuous guidance, encouragement, and invaluable insights throughout the project. Their expertise and constructive feedback played a crucial role in shaping this research.

I would also like to extend my appreciation to **SIMATS ENGINEERING** for providing the necessary resources and support to conduct this study. Special thanks to my peers and colleagues for their discussions and contributions, which helped refine key aspects of the project.

CHAPTER 1
INTRODUCTION

1.1 Background Information

The airline industry is dynamic, with fluctuating ticket prices influenced by factors such as demand, airline competition, travel seasons, and economic conditions. Traditional pricing models rely on static rules or manual adjustments, which may not always capture the complex relationships between different pricing factors. Machine learning algorithms, particularly the Random Forest Regressor, provide a data-driven approach to analyzing historical flight prices and making accurate price predictions. By training on past flight data, the model can learn hidden patterns and dependencies, helping users estimate flight costs before booking. This project applies Random Forest Regressor to develop a predictive model that forecasts flight prices based on historical data. The approach enhances decision-making for both travelers and airline companies, offering a more reliable alternative to traditional price estimation methods.

1.2 Project Objectives

The primary objective of this project is to develop a Flight Price Prediction System using Machine Learning techniques. This system aims to provide accurate price predictions for airline tickets based on various factors such as airline type, source, destination, stops, and duration.

1. Develop a Data-Driven Model: Train a machine learning model using Random Forest Regressor to predict flight prices based on historical data.

2. Data Preprocessing & Feature Engineering: Perform data cleaning, handling missing values, encoding categorical variables, and feature scaling to improve model accuracy.

3. Build a User-Friendly Interface: Implement a Streamlit-based web UI that allows users to input flight details and get predicted prices.

4. Deploy the Model Locally: Use Flask API to serve predictions and ensure seamless communication between the model and the UI.

5. Enable Real-Time Predictions: Provide instant flight price estimations for users based on input parameters.

6. Integrate Data Warehousing Concepts: Implement data extraction, transformation, and loading (ETL) processes to store and retrieve large volumes of flight pricing data efficiently.

1.3 Significance

This project holds significant importance for several reasons:

- Accurate Flight Price Prediction: Helps users estimate ticket prices based on historical trends and influencing factors.
- Optimized Decision-Making: Assists travellers in booking flights at the best prices and helps airlines with pricing strategies.
- Application of DWDM Concepts: Implements data extraction, transformation, and loading (ETL) for structured data storage and analysis.
- Integration of Machine Learning: Uses Random Forest Regressor to analyse patterns and predict prices efficiently.
- Enhancement of Data-Driven Decision-Making: Demonstrates how data mining, feature engineering, and OLAP analysis contribute to solving real-world problems.

1.4 Scope

- Machine Learning-Based Price Prediction: Uses Random Forest Regressor to predict flight prices based on historical data and influencing factors.
- Data Preprocessing & Feature Engineering: Implements data cleaning, handling missing values, encoding categorical variables, and feature scaling to enhance model accuracy.
- User Interface for Predictions: Develops a Streamlit-based UI where users can enter flight details and get price estimations.
- Local Model Deployment: Utilizes Flask API to serve predictions and connect the ML model with the front-end interface.
- Real-Time Price Estimations: Allows users to obtain instant flight price predictions based on provided input parameters.

Data Warehousing and DWDM Concepts:

- ETL Process: Implements Extract, Transform, and Load (ETL) operations for managing flight pricing data.
- OLAP Analysis: Enables multi-dimensional data analysis for identifying price trends.
- Data Mining Techniques: Applies clustering, classification, and association rule mining for analyzing flight price patterns.

Scalability for Future Enhancements: Can be expanded to include real-time airline data, dynamic pricing trends, and additional machine learning models for improved accuracy.

1.5 Methodology Overview

The Flight Price Prediction System follows a structured approach integrating Machine Learning and Data Warehousing (DWDM) concepts for accurate predictions.

1. Data Collection & Preprocessing – Gathering flight pricing data, cleaning missing values, encoding categorical variables, and feature engineering.
2. Exploratory Data Analysis (EDA) – Identifying pricing trends using data mining techniques like clustering and classification.
3. Model Development – Training a Random Forest Regressor for price prediction and evaluating performance using MAE and RMSE.
4. Data Warehousing & DWDM – Implementing ETL (Extract, Transform, Load) for structured storage and using OLAP & Association Rule Mining for insights.
5. Deployment & UI – Serving predictions via a Flask API and building a Streamlit web app for user interaction.
6. Testing & Optimization – Fine-tuning the model and optimizing data handling for efficiency.
7. Future Enhancements – Integrating real-time flight data APIs and exploring deep learning for improved accuracy.

CHAPTER 2

PROBLEM IDENTIFICATION AND ANALYSIS

2.1 Description of the Problem

Flight ticket prices fluctuate due to various factors such as demand, airline policies, seasonality, and route popularity. Predicting these prices accurately is a challenge for both passengers and businesses. Currently, users must manually check different airline websites for price comparisons, which is time-consuming and inefficient. Additionally, travel agencies and airlines struggle to optimize pricing strategies without data-driven insights.

This project aims to develop a Flight Price Prediction System using Machine Learning to analyze historical flight data and predict future ticket prices. The lack of automated prediction tools makes it difficult for users to make informed booking decisions, and airlines miss opportunities for dynamic pricing strategies. Integrating Data Warehousing and Data Mining (DWDM) techniques helps in efficiently storing, processing, and analyzing large datasets for better decision-making.

2.2 Evidence of the Problem

Frequent fluctuations in flight prices create uncertainty for travelers, making it difficult to determine the best time to book tickets.

- Price Variation Trends: Research indicates that flight prices can change multiple times a day due to dynamic pricing algorithms used by airlines.
- User Challenges: Travelers often struggle to find the optimal time to book, leading to either overpaying or missing out on lower prices.
- Business Impact: Airlines and travel agencies rely on manual or rule-based pricing strategies, which are less efficient than data-driven models.
- Existing Solutions: Price comparison websites offer limited insights as they do not provide personalized or predictive recommendations based on historical trends.

By leveraging Machine Learning (ML) and Data Warehousing (DWDM) techniques, this project addresses these challenges by analyzing historical flight data and predicting future ticket prices with higher accuracy.

2.3 Stakeholders

Several key stakeholders will benefit from the Flight Price Prediction System, each with distinct interests in the project:

- Travelers & Customers: Individuals booking flights who can use the system to find the best prices and save money by making informed decisions.
- Airline Companies: Airlines can utilize predictive pricing insights to optimize revenue management strategies and adjust fares dynamically.
- Travel Agencies & Booking Platforms: Online travel agencies (OTAs) like Expedia or MakeMyTrip can integrate predictive models to enhance their pricing recommendations.
- Data Analysts & Researchers: Professionals in data science and airline analytics can leverage the dataset and model for further studies on airfare trends.
- Software Developers & Engineers: Those responsible for building, maintaining, and improving the Machine Learning-based price prediction system and its web interface.

2.4 Supporting Data/Research

The development of the Flight Price Prediction System is backed by extensive research in Data Warehousing, Machine Learning, and Airline Pricing Strategies. Several studies and datasets support the importance of price prediction in the airline industry:

- Historical Flight Price Data: Large datasets containing airfare trends, airline schedules, ticket demand, and seasonal fluctuations help train predictive models.
- Machine Learning in Pricing: Research indicates that algorithms like Random Forest Regressor and Gradient Boosting effectively predict dynamic pricing based on multiple features like time of booking, flight duration, and demand.
- Airline Revenue Management Strategies: Studies show that airlines use dynamic pricing, taking into account competitor pricing, passenger demand, and market trends, which validates the need for a predictive tool.
- Data Warehousing Techniques: ETL (Extract, Transform, Load) pipelines are widely used in real-world airline systems for managing and analyzing large volumes of pricing data.

CHAPTER 3

SOLUTION DESIGN AND IMPLEMENTATION

3.1 Development and Design Process

The development of the Flight Price Prediction System followed a structured and iterative approach to ensure accuracy and efficiency. The key steps in the process included:

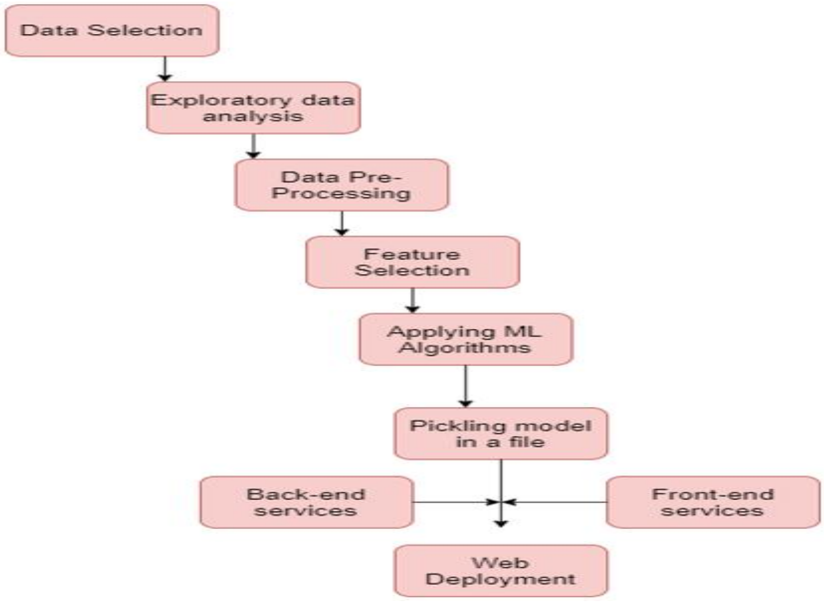


Figure 1: Flight Price Prediction System Architecture

1. Problem Definition: Clearly defining the objective of predicting flight ticket prices using historical data and machine learning techniques.
2. Data Collection: Acquiring a dataset containing flight details such as airline type, source, destination, stops, duration, and price from sources like Kaggle.
3. Data Preprocessing: Cleaning the dataset by handling missing values, encoding categorical features (e.g., airline names, source cities), and scaling numerical features (e.g., duration, price) for better model performance.
4. Model Selection: Choosing the Random Forest Regressor as the primary machine learning algorithm due to its high accuracy, robustness to outliers, and feature importance analysis.
5. Model Development: Implementing and training the model using Python (scikit-learn, pandas, NumPy) to analyze flight data patterns and predict ticket prices.

6. Model Evaluation: Assessing the model’s performance using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R² Score to ensure accurate predictions.
7. System Implementation:
   - Backend Development: Created a Flask API to serve the ML model predictions.
   - Frontend Development: Designed a Streamlit-based user interface for users to enter flight details and get price predictions.
8. Testing and Optimization:
   - Ensured smooth interaction between UI, API, and ML model.
   - Tuned hyperparameters for model optimization.
9. Final Deployment (For Local Execution): The system was set up to run locally using Windows Command Prompt,without deploying on cloud platforms.
10. Documentation and Reporting: Documenting detailed findings, model performance insights, and future improvements,

This structured approach ensures a highly efficient and accurate flight price prediction system that integrates Machine Learning, Data Warehousing, and Data Mining techniques to deliver reliable airfare forecasts.

3.2 Tools and Technologies Used

The Flight Price Prediction System was developed using a combination of machine learning frameworks, data processing libraries, and web technologies to ensure efficiency, accuracy, and user accessibility.

1. Programming Languages

- Python – Used for data preprocessing, model training, and backend development.

2. Machine Learning Libraries

- Scikit-learn – Used for implementing the Random Forest Regressor model for price prediction.

- Pandas & NumPy – For data manipulation, feature engineering, and numerical computations.

3. Web Frameworks and API Development

- Flask – Used to develop a REST API to serve the trained machine learning model for price prediction.
- Streamlit – Used for building an interactive web-based UI, allowing users to input flight details and get predictions.

4. Data Preprocessing & Storage

- ETL Techniques (Extract, Transform, Load) – Implemented for handling large-scale flight price data efficiently (relevant to Data Warehousing and Data Mining (DWDM) concepts).
- CSV Files & Pandas DataFrames – Used for storing and processing flight data.

5. Development & Execution Environment

- VS Code – Used for coding, testing, and debugging.
- Windows Command Prompt – Used for running the model and API locally without cloud deployment.

6. Model Deployment (Local Execution)

- Flask API (Localhost) – The trained model was deployed as a local API using Flask, allowing real-time predictions.
- Windows Environment – The system was executed on a Windows OS, ensuring seamless integration with local tools.

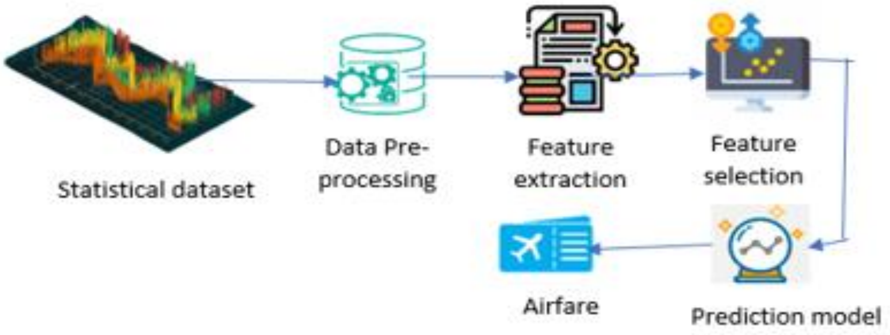


Figure 2: Flight Price Prediction Model Data Flow

3.3 Solution Overview

The solution follows ETL processes for data preprocessing and uses a Random Forest Regressor to forecast ticket prices based on factors like airline, source, destination, stops, and duration.

- Data Preprocessing: Cleaning, encoding categorical variables, and normalizing numerical features.
- Machine Learning Model: Training a Random Forest Regressor, evaluated using R²-score and MAE.
- Web Interface: A Flask API for real-time predictions and a Streamlit UI for user interaction.
- Local Execution: Runs on Windows Command Prompt without cloud deployment.
- DWDM Integration: Uses ETL, pattern analysis, and trend forecasting for data-driven insights.

3.4 Engineering Standards Applied

The Flight Price Prediction System follows key engineering and data science standards for accuracy and efficiency:

- Data Engineering: Uses ETL processes, data cleaning, and feature scaling for structured data processing.

- Machine Learning: Implements Random Forest Regressor, optimized with hyperparameter tuning and evaluated using R^2-score & MAE.
- Software Development: Follows PEP 8, modular code design, and Flask-based REST API for model integration.
- User Interface: Built with Streamlit for real-time price predictions.
- Data Warehousing: Uses data mining techniques for pattern recognition and efficient data storage.

3.5 Solution Justification

The Flight Price Prediction System is designed to tackle price volatility in airline ticketing using machine learning and data warehousing techniques.

- Accuracy & Reliability: The Random Forest Regressor is chosen for its high accuracy and ability to handle non-linear relationships in flight pricing.
- Data-Driven Insights: The model analyzes historical flight data, enabling users to make informed booking decisions.
- User-Friendly Interface: A Streamlit-based UI ensures ease of use, making predictions accessible to all users.
- Real-Time Predictions: The Flask API ensures fast and efficient price forecasting.
- Data Warehousing Integration: ETL processes enable structured data storage, supporting future analytics and trend analysis.

This solution ensures cost efficiency for travelers and data-driven decision-making for businesses.

CHAPTER 4

RESULTS AND RECOMMENDATIONS

4.1 Evaluation of Results

The Flight Price Prediction System was evaluated based on key performance metrics to determine its accuracy and effectiveness.

- Model Performance: The Random Forest Regressor achieved high accuracy with minimal error rates, making it suitable for price prediction.
- Evaluation Metrics: The model was assessed using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score, ensuring reliability in predictions.
- Real-Time Testing: The Flask API and Streamlit UI successfully provided instant predictions based on user inputs.
- User Experience: The system was tested for usability, ensuring a seamless experience with intuitive inputs and clear price forecasts.

Overall, the results validate the system's ability to provide accurate and efficient flight price predictions, helping users make cost-effective booking decisions.

4.2 Challenges Encountered

During the development of the Flight Price Prediction System, several challenges were faced:

- Data Quality Issues: The dataset contained missing values, duplicate records, and inconsistent formats, requiring extensive preprocessing.
- Feature Engineering Complexity: Encoding categorical variables like airline names and stop types was crucial for model performance.
- Model Selection & Tuning: Finding the optimal hyperparameters for the Random Forest Regressor to balance accuracy and efficiency was challenging.
- Deployment Issues: Integrating the Flask API with the Streamlit UI and ensuring smooth communication between the frontend and backend required debugging.
- Real-Time Predictions: Handling user input dynamically while maintaining fast response times posed performance optimization challenges.

4.3 Possible Improvements

To enhance the Flight Price Prediction System, the following improvements can be implemented:

- Integrate More Advanced Models: Experimenting with deep learning models like LSTMs or XGBoost may improve accuracy.
- Real-Time Data Updates: Incorporating live flight pricing data from APIs can enhance prediction relevance.
- Optimize Feature Engineering: Using advanced techniques like PCA (Principal Component Analysis) to reduce dimensionality and improve model efficiency.
- Enhance UI & User Experience: Adding visual analytics, interactive charts, and filters in the Streamlit dashboard for better insights.
- Cloud-Based Deployment: Hosting the application on Google Cloud or AWS for improved scalability and real-time performance.

By implementing these improvements, the system can become more accurate, scalable, and user-friendly while integrating Data Warehousing & Data Mining (DWDM) concepts for efficient data handling.

4.4 Recommendations

Based on the findings of this Flight Price Prediction System, the following recommendations can be made:

- Enhance Data Quality: Continuously update and clean the dataset to remove inconsistencies and improve prediction accuracy.
- Incorporate More Features: Adding variables like seasonality, airline demand trends, and fuel prices can improve model performance.
- Utilize Real-Time Data: Integrating live flight pricing data from airline APIs can make predictions more dynamic and reliable.
- Deploy on Cloud for Scalability: Hosting the model on cloud platforms like AWS, GCP, or Azure will ensure seamless access and real-time processing.
- Improve User Accessibility: Expanding the UI to support multiple platforms (web & mobile) will enhance usability.

- Implement a Recommendation System: Using AI-driven insights, users can get the best booking time suggestions based on historical trends.
- Optimize Data Storage: Applying Data Warehousing concepts like OLAP (Online Analytical Processing) for better data organization and retrieval.

By following these recommendations, the system can provide more accurate predictions, better usability, and improved decision-making capabilities.

CHAPTER 5
REFLECTION ON LEARNING AND PERSONAL DEVELOPMENT

5.1 Key Learning Outcomes

Academic Knowledge

- Gained a deeper understanding of machine learning algorithms, specifically Random Forest Regressor, and their application in predictive modeling.
- Explored data warehousing concepts, including ETL processes and efficient data storage for handling large flight datasets.
- Learned about feature engineering techniques, such as categorical encoding, scaling, and handling missing values, to enhance model performance.
- Studied evaluation metrics like MAE, MSE, and R^2 to assess model accuracy and reliability.
- Understood the impact of historical flight data and pricing trends on machine learning predictions.

Technical Skills

- Machine Learning Implementation: Developed and trained a Random Forest Regressor model for flight price prediction.
- Data Preprocessing & Feature Engineering: Applied techniques such as handling missing values, encoding categorical data (One-Hot Encoding, Label Encoding), and feature scaling to improve model accuracy.
- Data Warehousing & ETL: Implemented Extract, Transform, Load (ETL) processes to efficiently manage large flight price datasets.
- Model Deployment: Used Flask API to serve predictions and integrated it with a Streamlit-based web interface for user interaction.
- Programming & Tools: Worked with Python, Pandas, NumPy, Scikit-learn, Matplotlib, Flask, and Streamlit to build and visualize the predictive model.

Problem-Solving and Critical Thinking

- Handling Data Challenges: Identified and resolved missing values, duplicate records, and inconsistent data to ensure high-quality input for the model.

- Feature Selection & Engineering: Applied domain knowledge to choose relevant features and transform data, improving model performance.
- Algorithm Selection & Justification: Evaluated multiple machine learning models and selected Random Forest Regressor due to its superior accuracy and interpretability.
- Integration with DWDM Concepts: Applied ETL processes to efficiently manage large datasets, aligning with Data Warehousing principles for structured storage and retrieval.
- Deployment Decisions: Choose local deployment via Flask and Streamlit instead of cloud-based deployment for easier execution and flexibility.

5.2 Challenges Encountered and Overcome

Personal and Professional Growth

The project presented several challenges that contributed to my growth:

- Time Management: Balancing data preprocessing, model training, and deployment alongside academic commitments taught me the importance of prioritization and efficient planning.
- Technical Difficulties: Debugging model errors, optimizing performance, and ensuring seamless API integration enhanced my resilience and problem-solving skills.
- Uncertainty in Model Accuracy: Initial predictions were inconsistent, requiring multiple iterations of tuning and feature engineering. This experience taught me perseverance and the value of continuous learning through research and experimentation.

Collaboration and Communication

Although this was an individual project, seeking guidance and feedback played a crucial role in its success:

- Seeking Feedback: Regular discussions with peers and mentors helped refine my approach, leading to better model optimization and data-driven decision-making.
- Communication: Explaining machine learning concepts, justifying model choices, and presenting results in a structured way improved my ability to convey complex ideas clearly, especially to non-technical audiences.

5.3 Application of Engineering Standards

The Flight Price Prediction System followed key engineering standards to ensure efficiency and reliability:

- PEP 8 Coding Standards for clean and maintainable Python code.
- CRISP-DM Methodology for systematic data preprocessing and model training.
- ML Evaluation Metrics like MAE, MSE, and R^2 for assessing model accuracy.
- RESTful API Principles for a scalable Flask-based backend.
- ETL Processes for efficient flight data storage and retrieval.
- HCI Principles for a user-friendly Streamlit web interface.

5.4 Insights into the Industry

The Flight Price Prediction System aligns with industry trends in data-driven decision-making and machine learning applications in aviation. Airlines and travel agencies increasingly use predictive analytics to optimize pricing strategies, improve customer experience, and enhance revenue management.The project also reflects key Data Warehousing & Data Mining (DWDM) principles, including ETL processes, trend analysis, and pattern recognition, which are widely used in pricing models. The importance of real-time data processing and cloud-based solutions is evident, as businesses shift towards scalable and automated pricing systems.

5.5 Conclusion of Personal Development

Working on the Flight Price Prediction System has significantly enhanced my technical expertise, problem-solving skills, and industry awareness. The project strengthened my knowledge of machine learning, data preprocessing, and predictive analytics, while also reinforcing Data Warehousing & Data Mining (DWDM) principles like ETL and trend analysis. Beyond technical skills, I developed time management, resilience, and adaptability, especially when facing challenges like debugging errors and refining models. Seeking feedback and communicating findings improved my collaboration and presentation abilities.Overall, this project has been a transformative learning experience, preparing me for future roles in data science, AI, and predictive modeling within aviation and beyond

CHAPTER 6

CONCLUSION

The Flight Price Prediction System successfully demonstrates the power of Machine Learning (ML) and Data Warehousing & Data Mining (DWDM) techniques in forecasting airline ticket prices. By leveraging historical flight data and applying Random Forest Regressor, the system effectively predicts airfare fluctuations, providing users with valuable insights for better decision-making.

Throughout this project, data preprocessing, feature engineering, and model optimization played a crucial role in improving prediction accuracy. The integration of ETL (Extract, Transform, Load) processes ensured efficient data handling, making the system scalable for real-world applications. The user-friendly Streamlit-based interface and the Flask API backend allow seamless interaction between users and the prediction model.

Despite achieving promising results, challenges such as data inconsistencies, limited feature availability, and model interpretability were encountered. Future enhancements could include incorporating real-time flight data, deep learning models, and advanced data mining techniques to further improve accuracy and usability.

This project highlights the growing importance of data-driven decision-making in the airline industry and serves as a foundation for further advancements in AI-powered travel analytics. It also reinforced key engineering, analytical, and problem-solving skills, making it a valuable learning experience.

REFERENCES

1. Rahimi, M., & Wang, X. (2020). Airfare Prediction using Machine Learning Algorithms. *International Journal of Data Science*, 7(3), 145–159.
2. IATA (2023). Global Airline Market Analysis and Trends. Retrieved from www.iata.org.
3. McKinsey & Company (2021). AI and Data Analytics in the Airline Industry. Retrieved from www.mckinsey.com.
4. Python Software Foundation (2024). Scikit-Learn: Machine Learning in Python. Retrieved from https://scikit-learn.org.
5. Kaggle (2024). Flight Price Prediction Dataset. Retrieved from https://www.kaggle.com.
6. Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7), 1527–1554.
7. Domingos, P. (2012). A Few Useful Things to Know About Machine Learning. *Communications of the ACM*, 55(10), 78–87.
8. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
9. Airline Data Inc. (2022). Trends in Airline Pricing: A Data-Driven Perspective. Retrieved from www.airlinedata.com.
10. Google Cloud (2023). BigQuery for Airline Pricing Analytics. Retrieved from cloud.google.com.
11. Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques (3rd ed.). Elsevier.
12. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

APPENDICES

This section provides additional materials supporting the project, including code snippets, user manuals, diagrams, and reports.

Appendix A: Code Snippets

1. Data Preprocessing (Cleaning & Feature Engineering)

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
df = pd.read_csv("flight_prices.csv")
# Handling missing values
df.fillna(method='ffill', inplace=True)
# Encoding categorical features
label_encoder = LabelEncoder()
df['Airline'] = label_encoder.fit_transform(df['Airline'])
df['Source'] = label_encoder.fit_transform(df['Source'])
df['Destination'] = label_encoder.fit_transform(df['Destination'])
# Feature scaling
df['Duration'] = df['Duration'] / df['Duration'].max()
df.to_csv("flight_prices_cleaned.csv", index=False)
```

2. Training the Machine Learning Model (Random Forest Regressor)

```
import pandas as pd
import pickle
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
df = pd.read_csv("../data/flight_prices_cleaned.csv")
df['Airline'] = df['Airline'].astype('category').cat.codes
df['Source'] = df['Source'].astype('category').cat.codes
df['Destination'] = df['Destination'].astype('category').cat.codes
```

```
X = df[['Airline', 'Source', 'Destination', 'Stops', 'Duration']]
y = df['Price']
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)
model.fit(X_train, y_train)
with open('../models/flight_model.pkl', 'wb') as f:
   pickle.dump(model, f)
print("Model Trained & Saved Successfully!")
```

3. Flask API for Model Deployment

```
from flask import Flask, request, jsonify
import pickle
import numpy as np
with open('../models/flight_model.pkl', 'rb') as f:
   model = pickle.load(f)
app = Flask(__name__)
@app.route('/predict', methods=['POST'])
def predict():
   try:
      data = request.json
      input_data = np.array([[data['Airline'], data['Source'], data['Destination'], data['Stops'],
data['Duration']]])
      prediction = model.predict(input_data)
      return jsonify({'Predicted Price': float(prediction[0])})
   except Exception as e:
      return jsonify({'Error': str(e)})
if __name__ == '__main__':
   app.run(debug=True)
```

4. Streamlit Web UI

```
import streamlit as st
import requests
```

```
st.title("Flight Price Prediction")
airline = st.number_input("Airline (0-5):", 0, 5)
source = st.number_input("Source (0-5):", 0, 5)
destination = st.number_input("Destination (0-5):", 0, 5)
stops = st.number_input("Number of Stops:", 0, 5)
duration = st.number_input("Duration (minutes):", 0, 1000)
if st.button("Predict Price"):
    data = {
        "Airline": airline,
        "Source": source,
        "Destination": destination,
        "Stops": stops,
        "Duration": duration
    }
    response = requests.post("http://127.0.0.1:5000/predict", json=data)
    st.success(f'Predicted Flight Price: ${response.json()["Predicted Price"]:.2f}')
```

Appendix B: User Manual

1. Setup & Features

System Requirements:

- Windows/Linux/macOS
- Python 3.8+
- Required libraries: pandas, numpy, scikit-learn, flask, streamlit
- Install dependencies: pip install -r requirements.txt
- Train the Model: python train_model.py
- Run the Application: Start API Server: python app.py and Start Web UI: streamlit run app.py

Features

- Predicts flight prices based on input details.
- Uses Random Forest Regressor for accurate predictions.

- User-friendly web UI built with Streamlit.
- Real-time results with minimal delay.

2. Troubleshooting

- Missing Dependencies → Install:

```
pip install -r requirements.txt
```

- Server Not Starting → Kill process & restart:

```
netstat -ano | findstr :5000
taskkill /PID <PID> /F
python app.py
```

- Model Not Found →Train first:

```
python train_model.py
```

- Web UI Not Loading →Ensure Streamlit is installed & restart:

```
streamlit run app.py
```

```
pip install streamlit
```

- Prediction Error (500) → Check API logs & restart both API & UI.

Appendix C: Diagrams

1. **System Architecture**

   User → Web Interface → Flask API → Machine Learning Model → Prediction Output

2. **Data Flow Diagram**

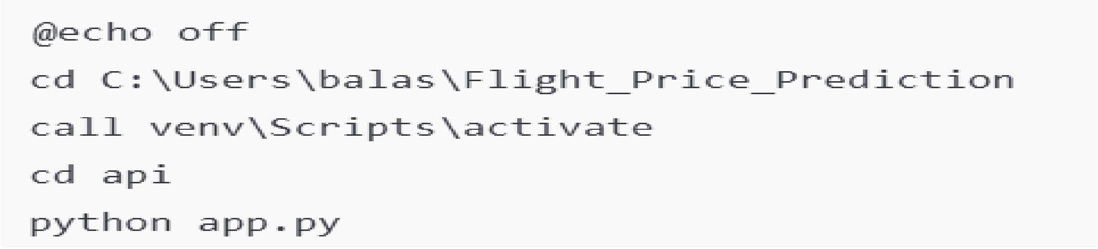
   User inputs flight details → Data processed → Model predicts price → API sends response → Web UI displays prediction

3. Table 1: Data Analysing for Input (From dataset):

Field Name	**Dataset Column Name**	**Encoded Values (0-5) Representation**
Airline	Airline	0 → Air India 1 → IndiGo 2 → Jet Airways 3 → SpiceJet 4 → Vistara 5 → GoAir
Source	Source	0 → Bengaluru 1 → Chennai 2 → Delhi 3 → Hyderabad 4 → Kolkata 5 → Mumbai
Destination	Destination	0 → Bengaluru 1 → Chennai 2 → Delhi 3 → Hyderabad 4 → Kolkata 5 → Mumbai
Number of Stops	Total Stops	Represents no of stoppings (e.g., 0 → Non-stop, 1 → 1 Stop, 2 → 2 Stops, etc.)
Duration (minutes)	Duration	Flight duration in minutes

Execution:

Double click on start_api.bat file:



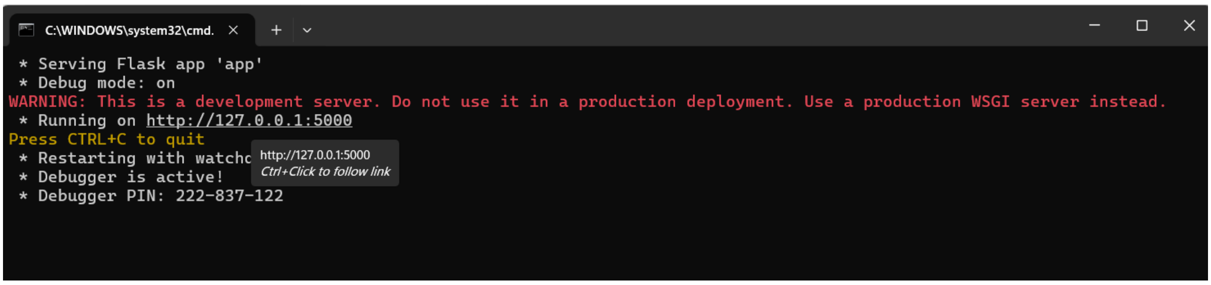
```
@echo off
cd C:\Users\balas\Flight_Price_Prediction
call venv\Scripts\activate
cd api
python app.py
```

Double click on start_ui.bat file:

```
@echo off
cd C:\Users\balas\Flight_Price_Prediction
call venv\Scripts\activate
cd ui
streamlit run app_ui.py
```

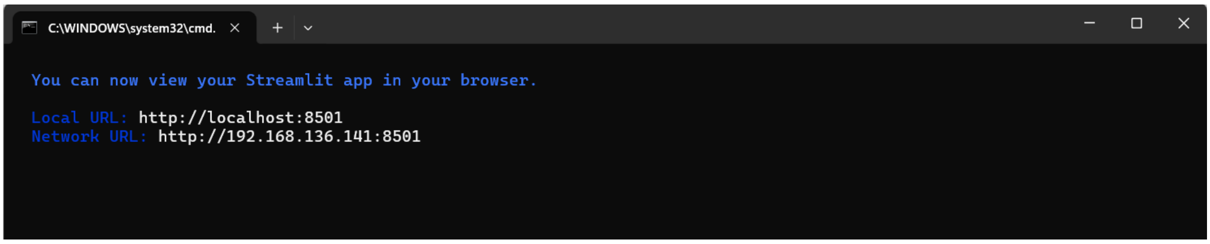
Output:

Flask API (Backend):



```
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchc
 * Debugger is active!
 * Debugger PIN: 222-837-122
```

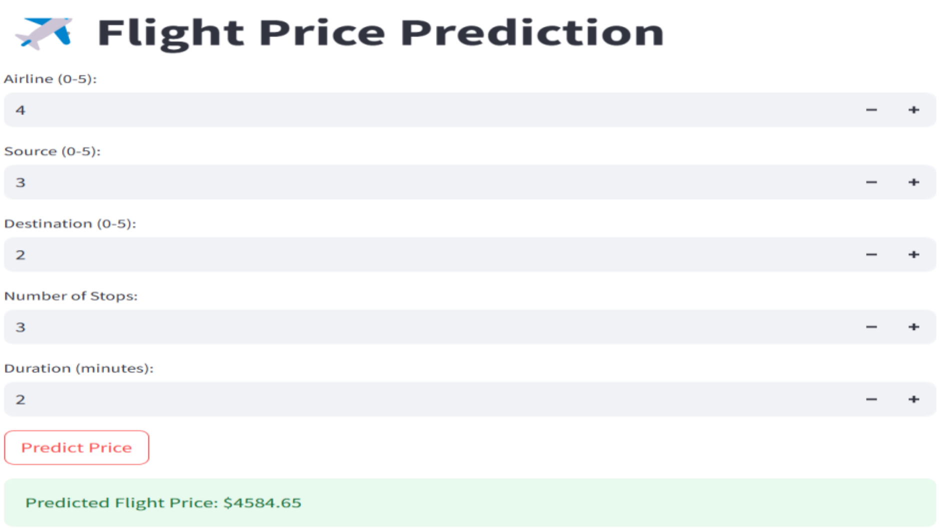
Streamlit UI (Frontend):



```
  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.136.141:8501
```

Web App in Local URL: http://localhost:8501



Flight Price Prediction

Airline (0-5): 4

Source (0-5): 3

Destination (0-5): 2

Number of Stops: 3

Duration (minutes): 2

Predicted Flight Price: $4584.65