ARCHITECTURE DESIGN (AD)

Flight Fare Prediction System

Revision Number: 1.0

Last date of revision: 23/05/2024

Authors:

[GOBIKRISHNAN

SRIRAM

SHINY]

Document Version Control

Date	Version	Description	Authors
23rd May 2024	1.0	Initial Draft	GOBIKRISHNAN SRIRAM SHINY

Contents

Abstract:

The Flight Fare Prediction System aims to provide users with an estimated flight fare based on various parameters such as departure time, arrival time, total stops, airline, source, and destination. By leveraging a pre-trained machine learning model, this system can help users make informed decisions when planning their travel. This document outlines the detailed design of the system, focusing on the technical aspects, including the dataset, prediction mechanism, logging, database integration, and deployment.

1. Introduction

1.1 Why this Low-Level Design Document?

The purpose of this document is to present a detailed description of the Flight Fare Prediction System. It will explain the purpose and features of the system, the interfaces, what the system will do, the constraints under which it must operate, and how the system will react to external stimuli. This document is intended for both stakeholders and developers and will be proposed to higher management for approval.

1.2 Scope

The system is designed to predict flight fares based on user inputs. The primary objective is to provide accurate fare predictions to users based on historical data and machine learning algorithms.

1.3 Constraints

Limited to predicting fares for a predefined set of airlines, sources, and destinations.

The accuracy of predictions depends on the quality and quantity of the historical data.

1.4 Risks

Data quality and completeness can affect prediction accuracy.

Changes in airline pricing algorithms over time can impact the model’s effectiveness.

1.5 Out of Scope

Real-time fare updates.

Integration with live booking systems.

Predicting fares for airlines, sources, or destinations not included in the training dataset.

2. Technical Specifications

2.1 Dataset

The dataset consists of historical flight fare data, including:

- Departure Time
- Arrival Time
- Total Stops
- Airline
- Source
- Destination
- Fare

2.2 Predicting Flight Fare

The system displays a form for users to input flight details.

Upon form submission, the system processes the input data, encodes categorical variables, and applies the pre-trained model to predict the fare.

The predicted fare is displayed to the user.

2.3 Logging

The system logs each user request and the corresponding prediction result.

Logs include timestamps, input parameters, and predicted fares.

Logging can be done to a file or a database for audit and debugging purposes.

2.4 Database

Cassandra is used for storing user inputs and prediction results.

Schema includes tables for storing user details, flight details, and prediction history.

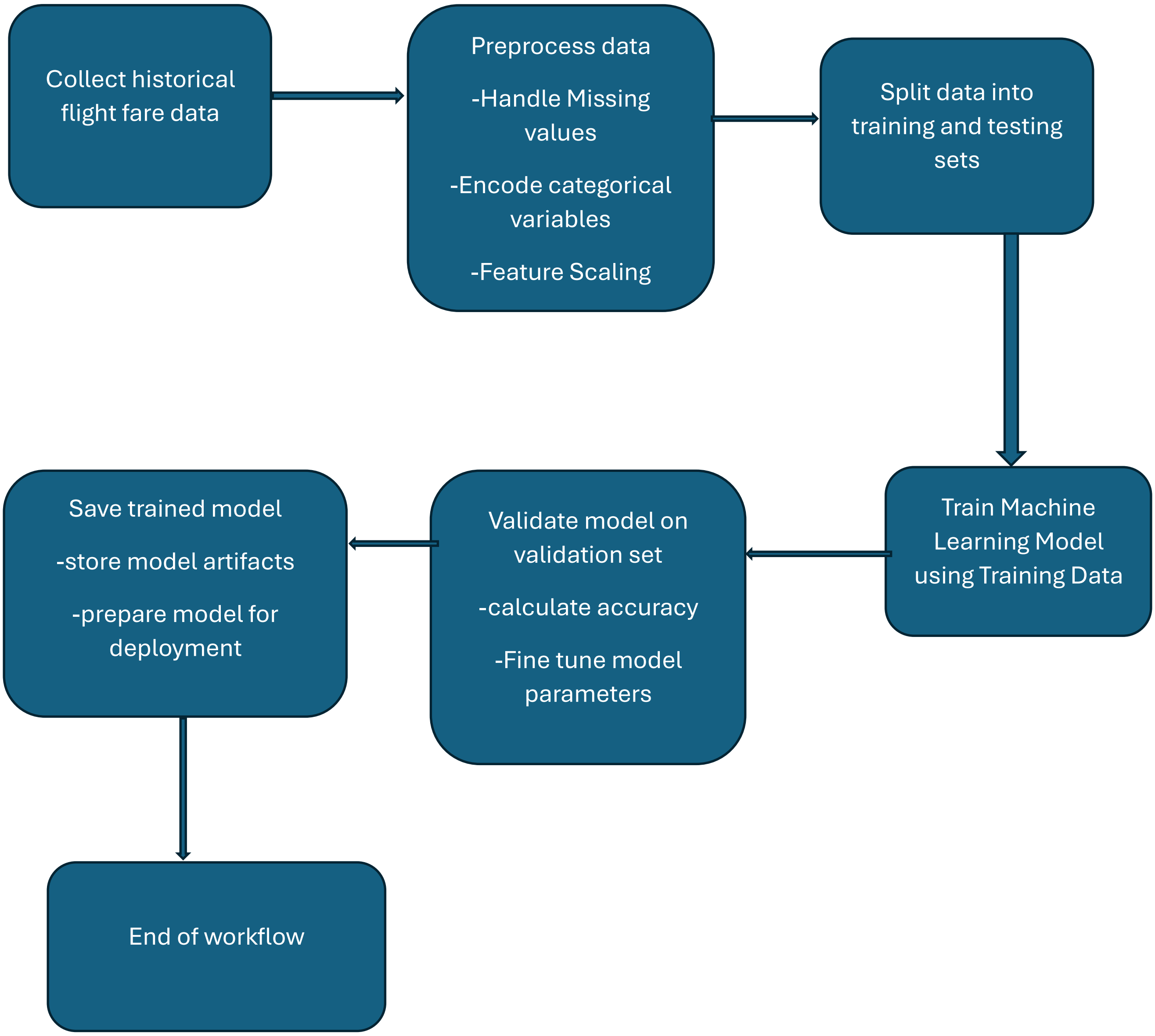
3. Technology Stack

Component	Technology
Frontend	HTML, CSS, JavaScript
Backend	Python, Flask
Database	Cassandra
Machine Learning	Scikit-learn
Deployment	Google cloud

4. Proposed Solution

The solution involves creating a web-based application where users can input flight details and receive fare predictions. The backend processes the inputs, encodes them appropriately, and uses a machine learning model to predict the fare. The results are displayed on the frontend, and all interactions are logged and stored in the database.

5. Model Training/Validation Workflow


Collect historical flight fare data
Preprocess data
-Handle Missing values
-Encode categorical variables
-Feature Scaling
Split data into training and testing sets
Train Machine Learning Model using Training Data
Validate model on validation set
-calculate accuracy
-Fine tune model parameters
Save trained model
-store model artifacts
-prepare model for deployment
End of workflow

Collect historical flight fare data.

Preprocess data: handle missing values, encode categorical variables, etc.

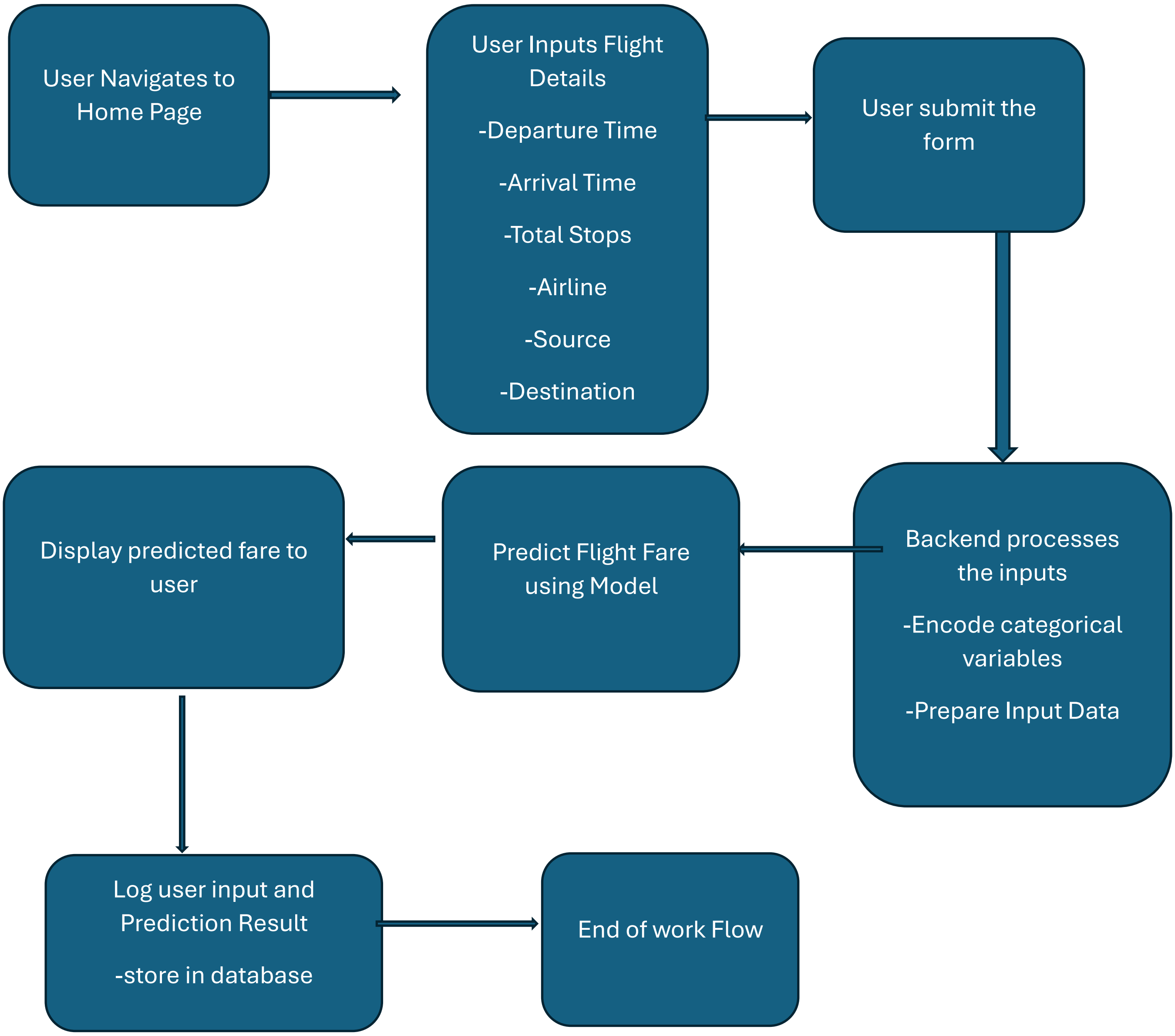
Split data into training and validation sets.

Train the machine learning model using training data.

Validate the model on the validation set to ensure accuracy.

Save the trained model to be used in the prediction system.

6. User I/O Workflow


User Navigates to Home Page
User Inputs Flight Details
-Departure Time
-Arrival Time
-Total Stops
-Airline
-Source
-Destination
User submit the form
Backend processes the inputs
-Encode categorical variables
-Prepare Input Data
Predict Flight Fare using Model
Display predicted fare to user
Log user input and Prediction Result
-store in database
End of work Flow

User navigates to the home page.

User inputs flight details (departure time, arrival time, total stops, airline, source, destination).

User submits the form.

Backend processes the inputs and predicts the fare using the machine learning model.

The predicted fare is displayed to the user.

The user's inputs and prediction result are logged and stored in the database.

7. Exceptional Scenarios

Invalid Input: Display an error message if user inputs are invalid or incomplete.

Model Prediction Failure: Log the error and display a generic error message to the user.

Database Connection Failure: Log the error and display a message indicating the issue to the user.

This Low-Level Design document provides a comprehensive overview of the Flight Fare Prediction System, outlining its components, functionalities, and workflows. The document serves as a guide for developers and stakeholders to understand and implement the system effectively.