**Architecture**

**Flight Price Prediction**

|  |  |
| --- | --- |
| **Written by** | **Denistan B** |
| **Documented Version** | **1.0** |
| **Last Revised Date** | **13 December 2024** |

**Table of Contents**

1. **Architecture**
2. **Architecture Description** 2.1. Data Description  
   2.2. Data Collection  
   2.3. Data Transformation  
   2.4. Data Pre-processing  
   2.5. Model Building  
   2.6. Data from User  
   2.7. Data Validation  
   2.8. Model Call for Prediction  
   2.9. Deployment

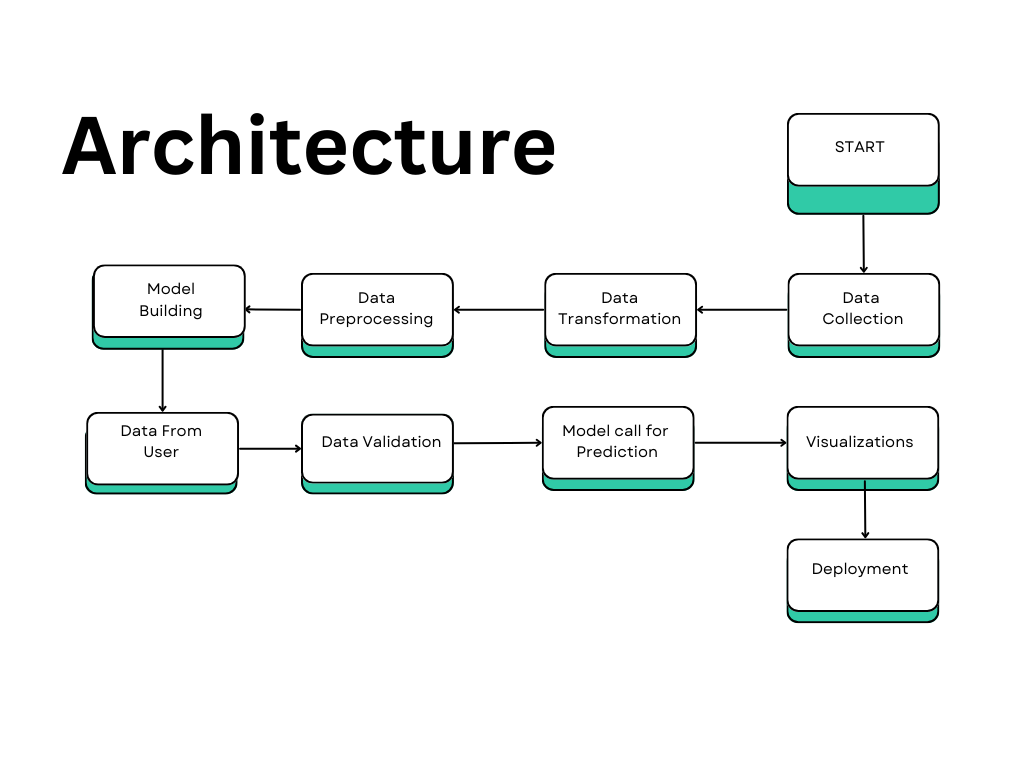
**1. Architecture**

The architecture of the flight fare prediction system is designed to efficiently predict the flight fares based on various factors such as flight duration, destination, and time of booking. The system integrates machine learning models, specifically Linear Regression and Random Forest Regressor, to forecast flight fares. Additionally, it leverages techniques like data transformation, validation, and deployment via Streamlit to create a seamless user experience. The ultimate goal is to predict flight fares that can guide users in planning their trips effectively and saving costs.

**2. Architecture Description**

The architecture consists of several interconnected components:

* **Data Input Layer**: The user provides input such as flight duration, source, destination, airline, and time of booking.
* **Data Preprocessing Layer**: This layer handles data cleaning, transformation, and ensures data quality for effective predictions.
* **Modeling Layer**: This includes the use of trained machine learning models (Linear Regression and Random Forest Regressor) to predict flight fares.
* **Output Layer**: The predictions and related visualizations are displayed for the user to analyze.
* **Deployment Layer**: The system is deployed on a cloud platform using Streamlit, allowing real-time interaction with the system.



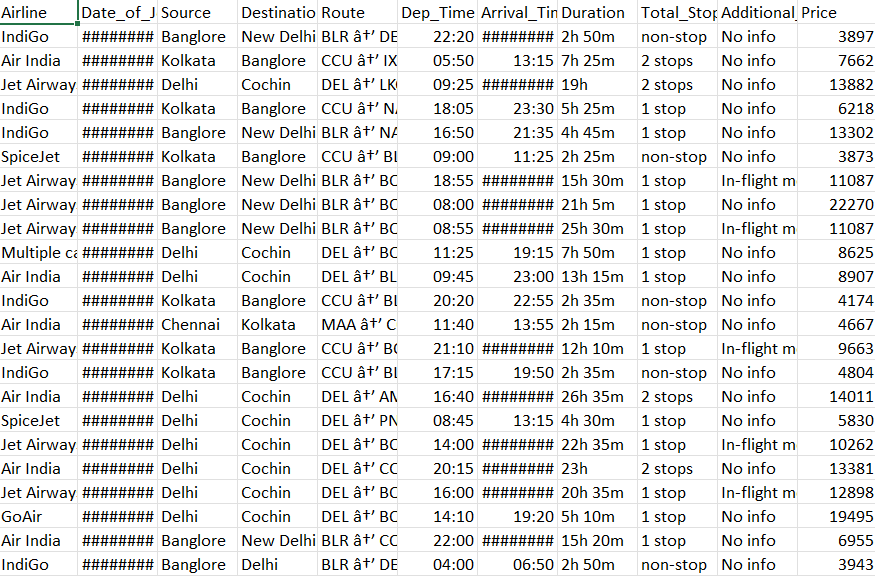
**2.1 Data Description**

The dataset used for training the model includes historical data on flight prices along with various factors that influence flight fares. Key features in the dataset include:

* **Flight Duration**: The time it takes for the flight to reach its destination.
* **Source and Destination**: The departure and arrival airports for the flight.
* **Airline**: The carrier operating the flight.
* **Flight Timing**: The timing of the flight, including whether it is on a peak day or during a holiday season.
* **Fare**: The target variable, representing the fare for the flight.

**2.2 Data Collection**

The data is collected from multiple sources, including:

* **Historical Flight Fare Data**: This includes past fare data for various flights across different airlines, which is collected in CSV or other suitable formats.
* **External Factors**: Additional data, such as holidays, weather, and events, is collected to evaluate its impact on fare pricing.
* **Flight Attributes**: Information about flight duration, source, destination, and airline is gathered.
* 

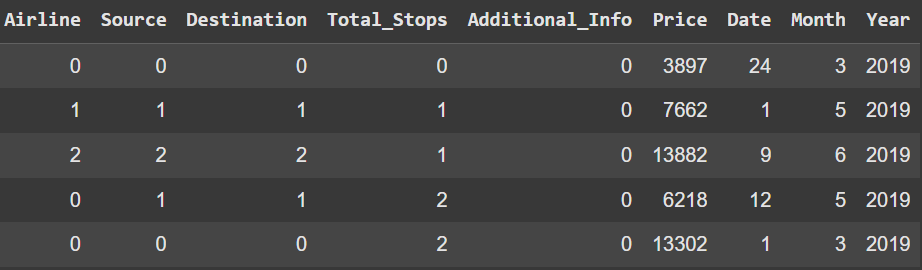
**2.3 Data Transformation**

Data transformation is performed to prepare the dataset for modeling:

* **Feature Encoding**: Categorical variables like airlines are one-hot encoded to make them suitable for machine learning models.
* **Log Transformation**: Continuous features like fare may be log-transformed to handle skewness in the data.
* **Normalization**: Numerical features such as flight duration may be normalized to ensure uniformity and improve model performance.

**2.4 Data Pre-processing**

Before training the models, several pre-processing steps are conducted:

* **Missing Data Handling**: Missing values are filled using mean or mode imputation based on the type of variable.
* **Outlier Detection**: Outliers are identified and handled to prevent them from skewing the predictions.
* **Data Splitting**: The dataset is split into training and testing sets to evaluate model performance.
* 

**2.5 Model Building**

The prediction models are built using:

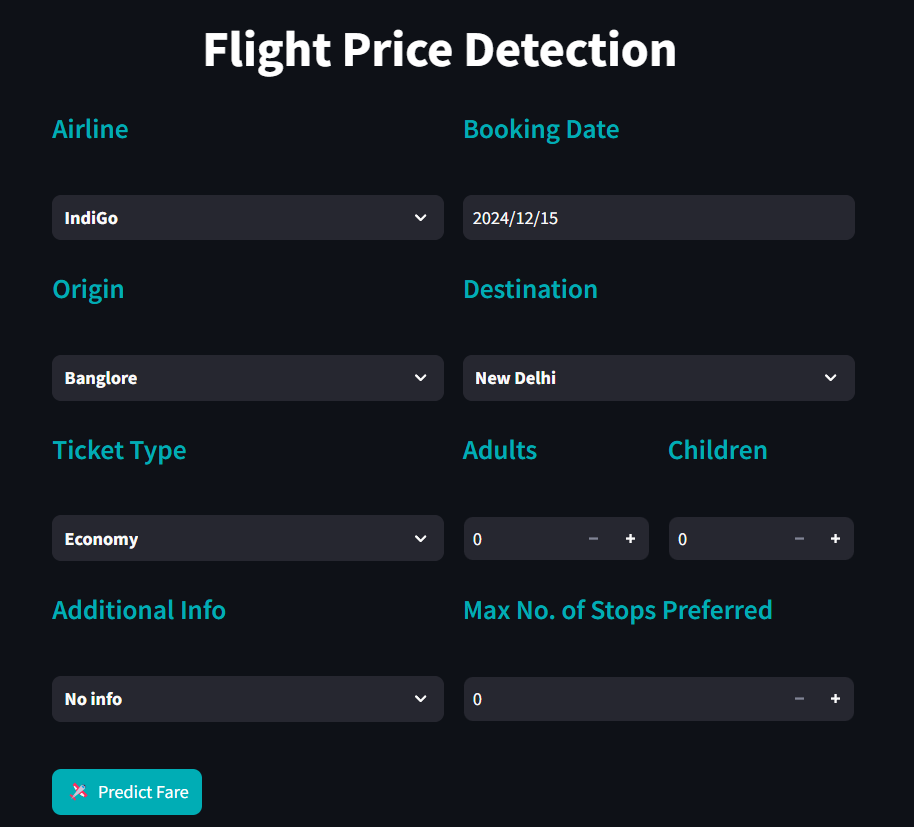
* **Linear Regression**: A statistical model that assumes a linear relationship between input variables and the target variable (flight fare). It is suitable for understanding the overall trend in the data.
* **Random Forest Regressor**: An ensemble learning method that combines multiple decision trees to improve prediction accuracy. It handles large datasets and captures complex patterns well.

**2.6 Data from User**

The user interacts with the system by providing the following data:

* **Flight Duration**: Time taken by the flight.
* **Source and Destination**: Departure and arrival locations.
* **Airline**: The airline operating the flight.
* **Flight Timing**: Time of day or seasonality of the flight.
* **Additional Features**: Any other factors, such as promotions, that might impact the fare. This data is entered through the Streamlit web interface.

Here's a detailed explanation of each topic in **Flight Price Prediction** architecture document:



**2.7 Data Validation**

Data validation ensures the accuracy and consistency of user inputs:

* **Format Checking**: Ensures all inputs are in the correct format (e.g., numeric inputs for flight duration, text inputs for airline).
* **Range Checking**: Ensures the input values fall within reasonable ranges (e.g., the duration must be a positive number).
* **Missing Data**: Flags any missing or incomplete inputs and prompts the user to complete the required fields.

**2.8 Model Call for Prediction**

Once the user input is validated, the pre-trained models (Linear Regression or Random Forest Regressor) are called to make predictions:

* The system loads the appropriate model based on user inputs such as airline or flight duration.
* The model processes the input data, applies necessary transformations, and generates a predicted flight fare.

**2.9 Deployment**

The system is deployed on a cloud platform (AWS, Heroku, etc.) using Streamlit. This allows users to access the system via a web interface and interact with it in real-time. The deployment ensures that the application is scalable and can handle a large number of users.

**2.10 Workflow Description**

1. **User Input**: The user interacts with the Streamlit app, providing necessary details such as flight duration, source, destination, airline, and other travel-related inputs.
2. **Data Transfer to Pickle File**: Once the user submits the input, the data is sent to a Pickle file containing the pre-trained machine learning model, which was built using historical flight fare data.
3. **Model Loading and Prediction Process**:
   * The Pickle file loads the trained model (either Linear Regression or Random Forest Regressor).
   * The model processes the input data and applies necessary transformations, such as log transformations for skewed variables.
   * Based on the inputs, the model predicts the flight fare.
4. **Output Display**:
   * After the model generates the predicted fare, the result is returned to the Streamlit app.
   * The predicted fare is displayed to the user in a user-friendly format.
5. **Visualization**:
   * Along with the predicted fare, the Streamlit app generates visualizations such as:
     + **Flight Duration vs Predicted Fare** graph.
     + **Airline vs Average Fare** graph.
   * These visualizations help users understand how different factors (e.g., flight duration, airline) correlate with the predicted flight fare.
6. **Final Display**: The user can view the predicted fare and accompanying visualizations, which provide valuable insights for trip planning.

**Final View:**

