# Flight Fare Prediction Using Machine Learning

This presentation details the development of a machine learning-based flight ticket price prediction system, outlining model selection, training, evaluation, and system deployment.


**Team Members :**

UCE2023511 - Harshada Chaudhari
UCE2023541 - Pranali Nikose
UCE2023544 - Astha Nitnaware

# ✈ Introduction: Navigating Flight Fare Volatility

Flight ticket prices are dynamic, influenced by a complex interplay of factors including airline, route, seasonal demand, travel class, number of stops, and booking timing. For travellers, manually forecasting these fluctuations is a significant challenge.

Machine learning presents a robust solution by learning intricate patterns from historical pricing data to accurately estimate future fares. This project is dedicated to constructing a reliable and robust price prediction system, aimed at empowering travellers to make informed, cost-effective decisions and streamline their travel planning process.

# ✈ Problem Statement: Predicting Flight Fares

The core challenge addressed by this project is to develop a highly accurate machine-learning model capable of predicting domestic flight ticket prices. This prediction must be based on a comprehensive set of flight-related features.

Furthermore, the project aims to seamlessly integrate this predictive model into a user-friendly web application, providing an accessible and intuitive platform for real-time fare estimations.

### Dynamic Pricing Complexity

Flight prices are influenced by numerous variables, making manual prediction challenging.

### Need for Accuracy

A high-performing model is crucial for reliable price estimations.

### User Accessibility

Deployment as a web application ensures ease of access for end-users.

# ✈ Abstract: Predicting Flight Fares with ML

This project details a machine-learning system for predicting domestic flight ticket prices within India. We processed a dataset of 300,153 flight records using various preprocessing techniques.

### Data Preprocessing

Utilising encoding, feature extraction, and cleaning operations for optimal data quality.

### Model Training & Evaluation

Four  regression models, **Linear Regression, Random Forest Regressor,XGBoost,and CatBoost** were trained and rigorously evaluated.

### Superior Performance

The Random Forest model achieved the highest accuracy of **96.95%**.

### Deployment

Deployed via a Flask-based web application with an HTML-CSS-JS frontend for real-time predictions.

# Dataset Overview:

Our predictive model was developed using the **Indian Airlines.csv** dataset, comprising an extensive **300,153 rows** of flight records.

### Key Features

- Airline, Source/Destination Cities
- Departure/Arrival Times, Stops, Class
- Duration, Days Left, Price (Target)

### DataType

- A robust mix of categorical and numerical variables.

# ✈ Methodology: The System Workflow

01

### Data Acquisition & Cleaning

Load the flight dataset and perform essential cleaning of attributes.

02

### Categorical Encoding

Encode categorical values using LabelEncoder for numerical representation.

03

### Data Partitioning

Split the processed dataset into distinct training and testing sets.

04

### Model Training

Train Machine Learning models : Linear Regression , Random Forest Regressor,XGBoost and CatBoost Regressor

05

### Performance Evaluation

Evaluate model performance using MAE, RMSE, and $R^2$ metrics.

06

### Model Persistence

Save thefinal, best-performing model using joblib for future use.

07

### Frontend Development

Build a user-friendly frontend form for capturing user inputs.

08

### Backend Integration

Create a Flask backend to load the trained model and generate predictions.

09

### Prediction Display

Display the predicted flight price directly within the web browser.

# ✈️ Data Preprocessing: Preparing the Data for Models

Thorough datapreprocessing was crucial to ensure the quality and suitability of the dataset for machinelearning models.

1

### Column Removal

Eliminated unnecessary columns to streamline the dataset.

2

### Missing Values

Handled missing values to maintain data integrity.

3

### Label Encoding

Categorical features such as airline,source_city, destination_city, departure_time, arrival_time, stops, and class were label encoded.

4

### Train-TestSplit

The dataset was split into an 80% training set (**240,122 samples**) and a 20% test set (**60,031 samples**).

5

### Serialization

Preprocessing objects (label_encoders.pkl, model_columns.pkl) were saved for consistent future use.

# ✈️ Algorithms Used for Prediction

## Linear Regression

- Simple baseline linear model
- Fits a straight-line relationship between features and price
- Good for understanding basic trend Patterns.

## Random Forest Regressor

- Ensemble of multiple decision trees
- Captures non-linear relationships
- Highly robust and less prone to overfitting
- Final prediction = average of all trees

## XGBoost

- Builds many small trees where each tree fixes mistakes of the previous one.
- Learns complex patterns in flight prices better than basic models.
- Optimized for speed but requires tuning and encoded data.

## CatBoost

- Handles Airline, Source, Destination automatically (no encoding needed).
- Gives stable and accurate predictions even with many categories.
- Gives good results with minimal tuning, but training can be slower.

# ✈ Evaluation Metrics: Quantifying Performance

To thoroughly assess our models, we employed industry-standard evaluation metrics, providing a clear picture of their predictive capabilities.

1

### Mean Absolute Error (MAE)

Measures the average magnitude of the errors in a set of predictions, without considering their direction.

2

### Root Mean Squared Error (RMSE)

Calculates the square root of the average of the squared errors, giving higher weight to larger errors.

3

### $R^2$ Score

Represents the proportion of the variance in the dependent variable that is predictable from the independent variables, indicating prediction accuracy.

4

### Accuracy (%)

Derived directly from the $R^2$ Score, expressed as a percentage: × 100.

# ✈️ Experimental Results & Model Comparison

| Model | $R^2$ Score | Accuracy (%) | MAE | RMSE |
| --- | --- | --- | --- | --- |
| Linear Regression | 0.8546 | 85.46% | 4624.99 | 7014.31 |
| Random Forest | 0.9695 | 96.95% | 2137.69 | 3967.47 |
| XGBoost | 0.9825 | 98.25% | 1651.59 | 3000 |
| CatBoost | 0.9755 | 97.55% | 1887.05 | 3550.99 |

**Random Forest Achieved highest accuracy ($R^2$ ~ 0.97) in the project.**
**Selected as the final model for deployment in the Flask frontend**

# Frontend Implementation: User Experience

Our interactive frontend provides as eamless experience for users to obtain real-time flight price predictions.

### HTML

Forms the structural backbone of the input interface, capturing essential flight details.

### CSS

Ensures a modern, responsive, and aesthetically pleasing user interface across devices.

### JavaScript

Facilitates client-side validation, constructs JSON requests, and dynamically displays predictions.

# Backend Implementation using Flask

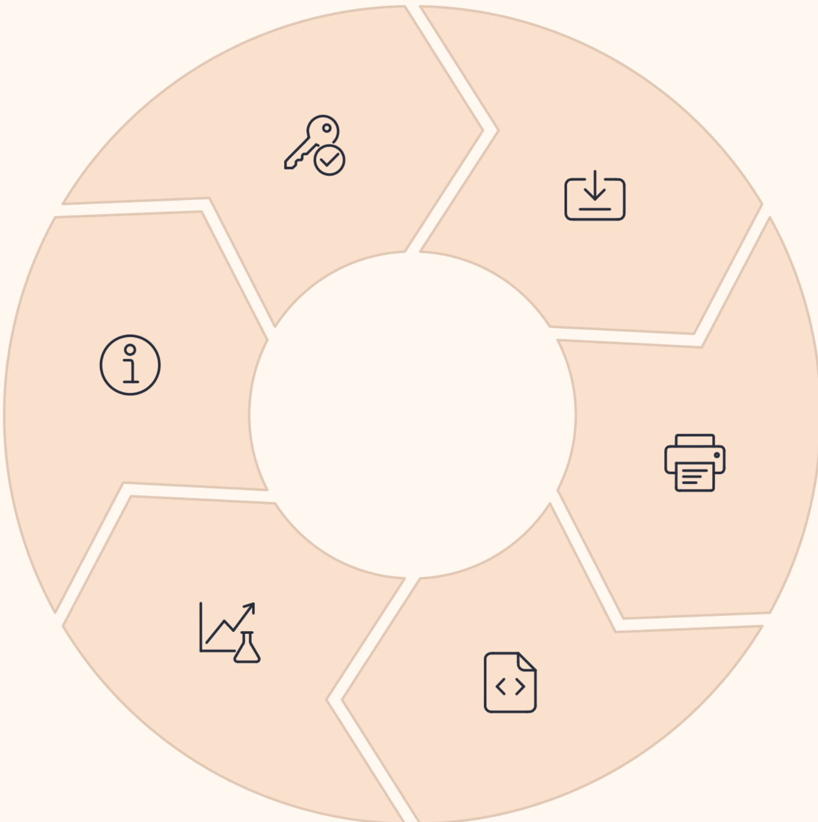
Flask serves as the robust intermediary, connecting the user interface to our sophisticated machine learning models.

### Routing

Manages traffic between HTML pages and prediction endpoints.

### Data Handling

Receives and processes user-provided input data.



### JSON Response

Returns the predicted price to the frontend in a structured format.

### Model Loading

Loads the pre-trained Random Forest model and associated preprocessors.

### Prediction

Utilises the loaded model to forecast flight prices.

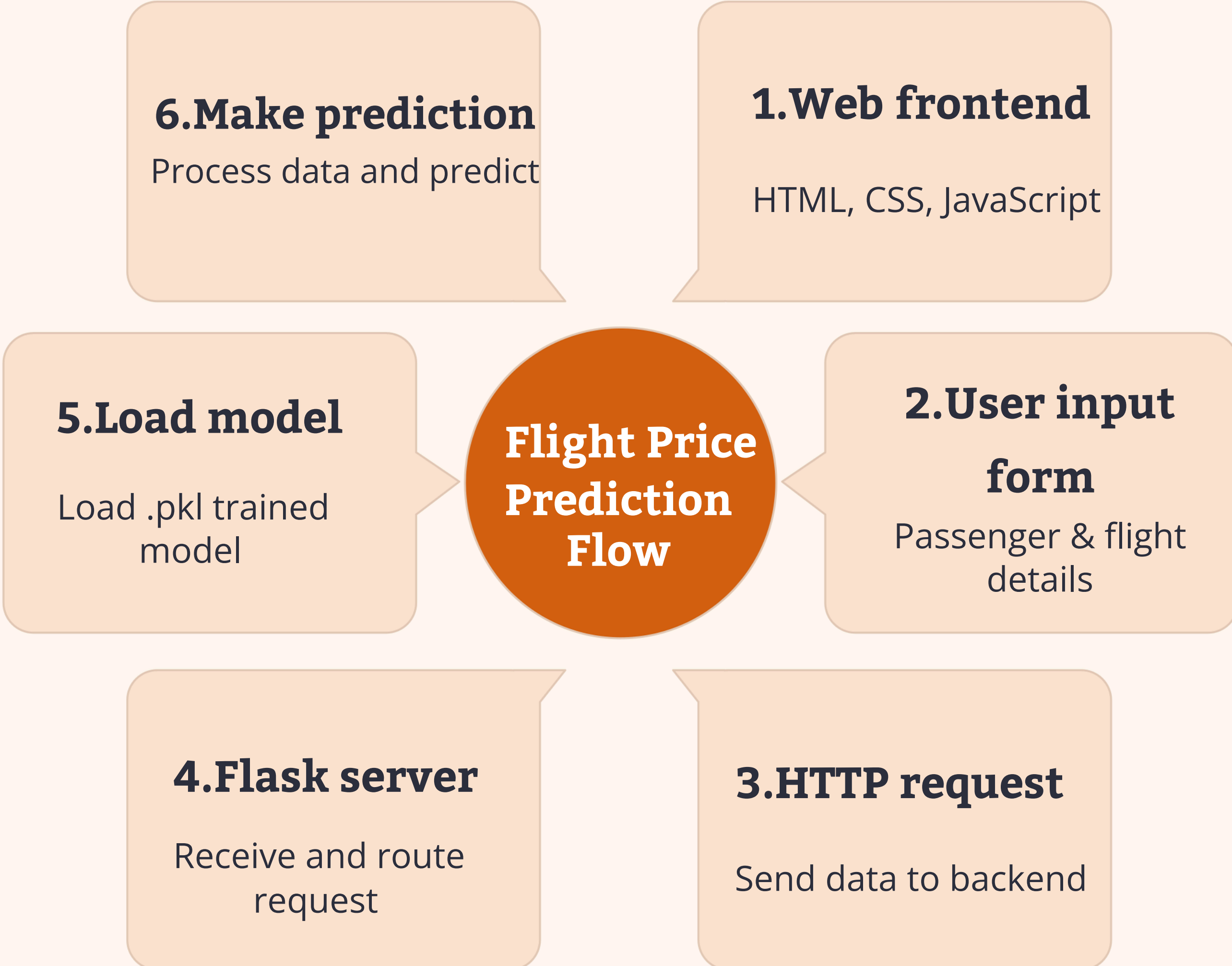
### Preprocessing

Applies necessary transformations (e.g., LabelEncoding) to input data.

**Critical Model Files: Random_Forest_model.pkl, model_columns.pkl, label_encoders.pkl.**

# ✈️ System Implementation: Frontend & Backend

An end-to-end flight price prediction system was developed, integrating the trained model into a user-friendly web application.




This architecture ensures a lightweight and efficient system, suitable for deployment on low-cost cloud infrastructure.

# ✈️Conclusion & Future Scope

This project successfully demonstrates the power of machine learning in solving real-world challenges, with ample room for future enhancements.

## Key Outcomes

Effective data preprocessing and cleaning.

XGBOOST achieved **98% accuracy**

Robust web application built using HTML, CSS, JS, and Flask.

Operational real-time prediction system.

Provides a valuable decision-making tool fortravellers.

## Future Enhancements

Integration of advanced deep learning models ( LSTM).

Real-time API data for live

flight price updates.

Deployment on scalable cloud platforms (AWS/Render).

Incorporation of route optimisation suggestions.

Development of a dedicated mobile application version.

# THANK YOU