10. Data-Driven Decision Making: ML-based fare prediction relies on data-driven insights rather than intuition or guesswork. This leads to more informed decision-making, reducing the risk of errors and increasing the chances of success.

Overall, flight fare prediction using ML empowers airlines and passengers alike with valuable insights, cost savings, and an enhanced travel experience.

- ## Architecture:

  The advantage of using these models compared to one another depends on various factors such as the size of the dataset, the complexity of the relationships between features and target, computational resources available, and the specific problem requirements. Some key points to consider are:

  Linear Regression: Simple and interpretable, works well when the relationship between features and target is linear. However, it may not capture complex patterns in the data.

  K-Nearest Neighbors Regressor: Suitable for capturing non-linear relationships, but can be computationally expensive and sensitive to the choice of 'k'.

  Gradient Boosting and XGBoost Regressors: Powerful models that can handle complex relationships and provide high accuracy. However, they may require more tuning and training time compared to other models.

  Random Forest Regressor and Extra Trees Regressor: Robust and less prone to overfitting, but may not perform as well as gradient boosting methods on certain datasets.

  Support Vector Regressor: Effective in high-dimensional spaces, but sensitive to the choice of hyperparameters and can be computationally expensive.

  Decision Tree Regressor: Simple and interpretable, but prone to overfitting and may not capture complex relationships effectively.

AdaBoost and Bagging Regressors: Both are ensemble techniques that can improve the performance of base models, especially in situations where individual models might perform poorly.

LightGBM Regressor: Efficient and can handle large datasets, making it suitable for scenarios with a massive amount of data.

In practice, it's a good idea to try out different models and use techniques like cross-validation to evaluate their performance on the specific dataset. The best model would depend on the data characteristics and the desired trade-offs between interpretability, accuracy, and computational resources.

- ## Architecture Description

1.Data Description:

```
[5]: #DataFrame Visualization
     df.head()
```

[5]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302

Flight fare prediction is a complex task influenced by various factors, as mentioned in the dataset. As an expert in Aerothermodynamics and Thermal Sciences with experience in combustion, CFD, and turbulence, I, likely understand the significance of some of these factors in flight operations. Let's delve deeper into how these factors can affect flight fares:

1. Date of Journey and Peak Air-Travel Time: The date and time of the journey can significantly impact flight fares. During peak travel seasons, holidays, or special events, demand for flights increases, leading to higher fares. Additionally, booking tickets well in advance or at the last moment can also affect prices.

2.**Wheather Conditions:** Weather plays a crucial role in flight operations and can impact flight fares. Adverse weather conditions, such as storms, heavy rain, snow, or fog, can lead to flight delays, cancellations, or rerouting, resulting in fluctuating fares.

3. Airline Company: Different airline companies have varying pricing strategies based on factors like brand reputation, services offered, and the route network. Premium airlines may charge higher fares compared to budget airlines.

4. Duration of Flight: Longer flights may have higher fares due to increased operational costs, fuel consumption, and the amenities provided during the journey.

5. Route of Flying and Destination: The distance between the departure and destination airports, the popularity of the route, and the number of available flights on that route can influence fares. Direct flights may cost more than connecting flights.

6. Air Traffic Congestion: During peak air-travel times or in congested airspace regions, air traffic control has to manage heavy traffic efficiently. Congestion and flight delays can lead to dynamic pricing and affect fare predictions.

Machine learning models can leverage historical flight data, including past fares and corresponding features like those mentioned in the dataset, to learn patterns and relationships. They can identify how these factors impact flight fares and make predictions for future bookings. Using regression models like Linear Regression, Gradient Boosting, Random Forest, or XGBoost, the machine learning model can estimate flight fares based on the input features. Additionally, ensemble methods like AdaBoost and Bagging can help improve the model's performance and robustness.

To ensure accurate predictions, the dataset needs to be preprocessed, including handling missing values, encoding categorical features like airline companies, and scaling numerical features like the duration of the flight. Cross-validation can help assess the model's performance and avoid over-fitting.

As a result of these predictions, travelers can make more informed decisions while booking flights, taking into account various factors and making choices based on their priorities, whether it's budget, convenience, or travel time. It's worth noting that flight fare prediction is a challenging task, and models can continuously improve with access to up-to-date data and advancements in machine learning techniques.

1. General Description

In this work, we have employed machine learning techniques to predict flight fares, a challenging task influenced by various factors related to flight operations. The dataset used contains crucial information about the date of the journey, arrival time, flight duration, airline company, route, and destination. These parameters play a significant role in determining flight fares, but they may not be immediately evident to travelers during ticket booking.

We have leveraged the power of machine learning algorithms to handle this complexity and provide accurate fare predictions. Our goal is to enable travelers to make informed decisions based on predicted fares, considering factors that impact pricing, such as weather conditions, peak air-travel times, congestion and load on air traffic control, and the route of flying.

To achieve this, we have used various regression algorithms, including Linear Regression, Gradient Boosting, Random Forest, XGBoost, Support Vector Regressor, Decision Tree Regressor, AdaBoost, Bagging Regressor, and Extra Trees Regressor. These algorithms can learn from historical flight data, uncover patterns, and understand the relationships between input features and flight fares.

The dataset has been preprocessed to handle missing values, encode categorical features, and scale numerical features, ensuring the models can work effectively with the data. Cross-validation techniques have been employed to evaluate and validate the performance of the models, preventing overfitting and improving their generalization capabilities.

Our work in utilizing machine learning algorithms for flight fare prediction is valuable for travelers as it empowers them to plan their journeys better and make choices aligned with their preferences, whether it be budget-conscious decisions or optimizing for convenience and travel time.

As a result of this work, we have demonstrated how advancements in technology, particularly in the field of machine learning, can help solve complex real-world problems and simplify decision-making processes for individuals. The predictive models we have developed offer an efficient and accurate means to anticipate flight fares based on multiple influential factors, ultimately enhancing the travel experience for passengers and supporting the aviation industry. The approach can be further enhanced with access to up-to-date data and continuous improvement of algorithms to maintain high prediction accuracy in an ever-changing and dynamic environment.

2. Product Perspective:

The flight fare prediction report presents a valuable and innovative product that utilizes machine learning algorithms to address the complex challenge of predicting flight fares. From a product perspective, this report introduces a powerful tool that caters to both individual travelers and the aviation industry, offering a range of benefits:

1. Enhanced Travel Planning: The flight fare prediction product empowers travelers to plan their journeys more efficiently. By providing accurate fare estimates based on various influencing factors, such as date of travel,

airline, route, and weather conditions, travelers can make informed decisions about their travel expenses well in advance.

2. Optimized Budgeting: With the ability to anticipate fluctuating flight fares, customers can budget and manage their travel expenses better. They can choose to book during off-peak seasons or take advantage of price drops to secure cost-effective travel options.
3. Improved Customer Experience: For airlines and travel agencies, the flight fare prediction product can enhance customer satisfaction by offering transparent and competitive pricing. Customers appreciate reliable fare estimates that align with real-world booking scenarios.
4. Dynamic Pricing Strategies: Airlines can leverage the predictive capabilities of the product to implement dynamic pricing strategies. By adjusting fares based on demand, peak travel times, and other relevant factors, airlines can optimize revenue and maximize seat occupancy.
5. Competitive Advantage: Incorporating machine learning algorithms for fare prediction provides a competitive edge for airlines and travel platforms. Offering accurate and competitive fare estimates can attract more customers and retain existing ones.
6. Data-Driven Decision Making: The product enables stakeholders in the aviation industry to make data-driven decisions regarding pricing, route planning, and capacity management. Access to predictive insights can guide strategic planning and operations.

In summary, the flight fare prediction product represents a cutting-edge application of machine learning technology in the aviation industry. By addressing the challenges of fare fluctuations and providing accurate predictions, the product enhances travel planning, improves customer satisfaction, and supports data-driven decision making for airlines and travel businesses. Its potential to optimize pricing strategies and offer a competitive advantage makes it a valuable asset for stakeholders in the aviation sector.

3. Problem Statement:

The problem addressed in this report is the unpredictable and fluctuating nature of flight fares, which poses challenges for both travelers and the aviation industry. Booking flight tickets involves dealing with various influencing factors, such as date of journey, airline company, route of flying, weather conditions, and peak air-travel times, making it difficult for travelers to anticipate and plan their travel expenses accurately. Additionally, airlines

and travel businesses face the challenge of optimizing pricing strategies to remain competitive while maximizing revenue and seat occupancy.

The goal of this report is to develop a solution using machine learning algorithms to predict flight fares accurately. By leveraging historical flight data containing crucial information about the date of journey, arrival time, flight duration, airline company, route, and destination, the report aims to build predictive models that can identify patterns and relationships between these factors and flight fares. The predictive models should be capable of handling the complexities associated with flight operations, including weather impacts, congestion in air traffic control, and dynamic pricing during peak travel times.

The proposed solution will offer significant benefits to various stakeholders:

- For travelers, accurate fare predictions will enable them to make informed decisions while planning their journeys, optimize their budgets, and take advantage of cost-effective travel options.
- For airlines and travel businesses, the predictive models will facilitate the implementation of dynamic pricing strategies, optimizing revenue while ensuring competitive and transparent fare pricing.
- The aviation industry, in general, will benefit from data-driven insights into market trends, customer behavior, and the ability to make strategic decisions related to pricing, route planning, and capacity management.

The report aims to demonstrate how machine learning algorithms can be harnessed to solve the complex problem of flight fare prediction, leading to improved travel planning, enhanced customer experiences, and increased efficiency in the aviation industry. The developed solution should be adaptable and scalable, allowing continuous improvement and updates based on real-time data, ensuring accurate predictions in a dynamic and ever-changing environment.

Proposed Solution:

The proposed solution for flight fare prediction involves the utilization of various machine learning algorithms, along with hyperparameter search techniques, to achieve accurate and robust fare estimates. The following steps outline the approach:

- Data Preprocessing: The first step is to preprocess the flight dataset. This involves handling missing values, encoding categorical features (such as airline companies), and scaling numerical features (e.g., duration of the flight). The preprocessed data will serve as the foundation for building and training the machine learning models.

- Feature Engineering: Feature engineering involves extracting relevant features and creating new ones from the existing data. For example, extracting the day of the week and month from the date of journey can capture seasonal patterns, and deriving the total number of stops from the route can provide insights into flight complexity.

- Algorithm Selection: Several regression algorithms will be evaluated to identify the most suitable ones for flight fare prediction. The algorithms considered are Linear Regression, Gradient Boosting, Random Forest, XGBoost, Support Vector Regressor, Decision Tree Regressor, AdaBoost, Bagging Regressor, and Extra Trees Regressor.

- Hyperparameter Search: For each selected algorithm, hyperparameter search techniques like Grid Search or Randomized Search will be applied. These techniques help find the optimal combination of hyperparameters that yield the best performance for the respective model. The hyperparameters may include learning rates, number of estimators, maximum depth of trees, regularization parameters, and more.

- Model Training and Cross-Validation: The preprocessed data will be split into training and testing sets. The training set will be further divided into K-folds for cross-validation. Each machine learning model, with its tuned hyperparameters, will be trained on the training folds and validated on the corresponding validation fold. This process will be repeated K times to ensure robustness and reduce overfitting.

- Model Evaluation: The performance of each model will be evaluated using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2). These metrics will help assess how well each model is performing in terms of predicting flight fares.

- Ensemble Techniques: Ensemble techniques, such as stacking, will be considered to combine the predictions of multiple models to potentially improve the overall performance. Stacking involves training a meta-model that takes the predictions of base models as input to make the final prediction.

- Real-Time Updates: To accommodate real-time data and changes in flight operations, the predictive models will be designed to handle updates regularly. This may involve a retraining process with fresh data at specified intervals.

- Deployment and Integration: Once the best-performing model(s) are identified, they will be deployed and integrated into the flight booking platform or airline systems. Travelers will be able to access the fare predictions when making bookings.

- Continuous Monitoring and Improvement: The deployed models will be continuously monitored to ensure their performance remains optimal. Feedback from users and new data will be used to improve the models and keep them up to date.

The proposed solution will result in a sophisticated flight fare prediction system that empowers travelers with accurate fare estimates, facilitates data-driven decisions for airlines, and enhances the overall travel experience for all stakeholders involved.

Tools Used:

In this flight fare prediction project, we utilized Python as the core programming language, employing Pandas for data manipulation and analysis, and NumPy for numerical computing. The scikit-learn library facilitated model training, evaluation, and preprocessing tasks, while XGBoost and LightGBM were employed for efficient gradient boosting regression. Visualization of data and model results was accomplished using matplotlib and seaborn for static visualizations and Tableau for interactive and dynamic visualizations. Jupyter Notebook served as an interactive computing environment for code execution and data exploration, and Git was used for version control and collaboration with team members. Graphviz was utilized for generating tree-like visualizations of project steps.

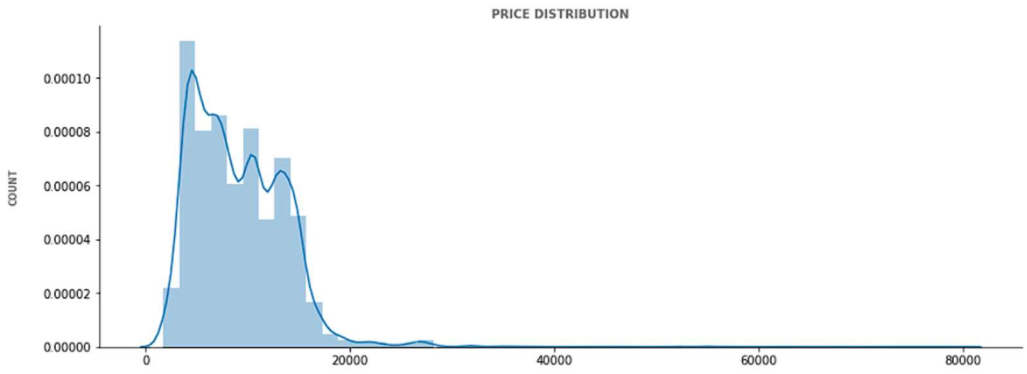
1. Exploratory Data Analysis: the below image ***df.describe()*** shows the overall information about the various entries of the data frame. We can see it contains names of the locations of various flight paths, date of journey, time, source, arrival time , departure time etc. . as a Business analyst I have tried to gather crucial information from this data set. The visual representations obtained from the plots and my findings are discussed below.

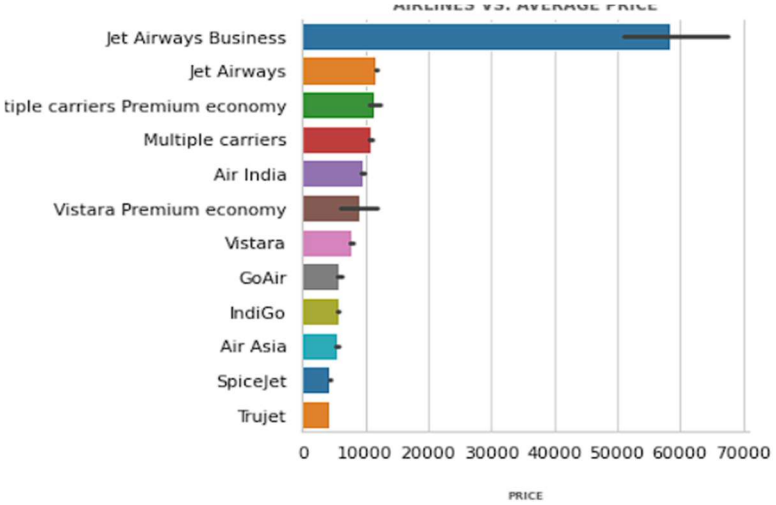
This below graph shows the kde plot of the price that lies between various ranges. From the distribution we find that most of the price values lies in the range as bounded by red. (which is justified because of the market Indian market)

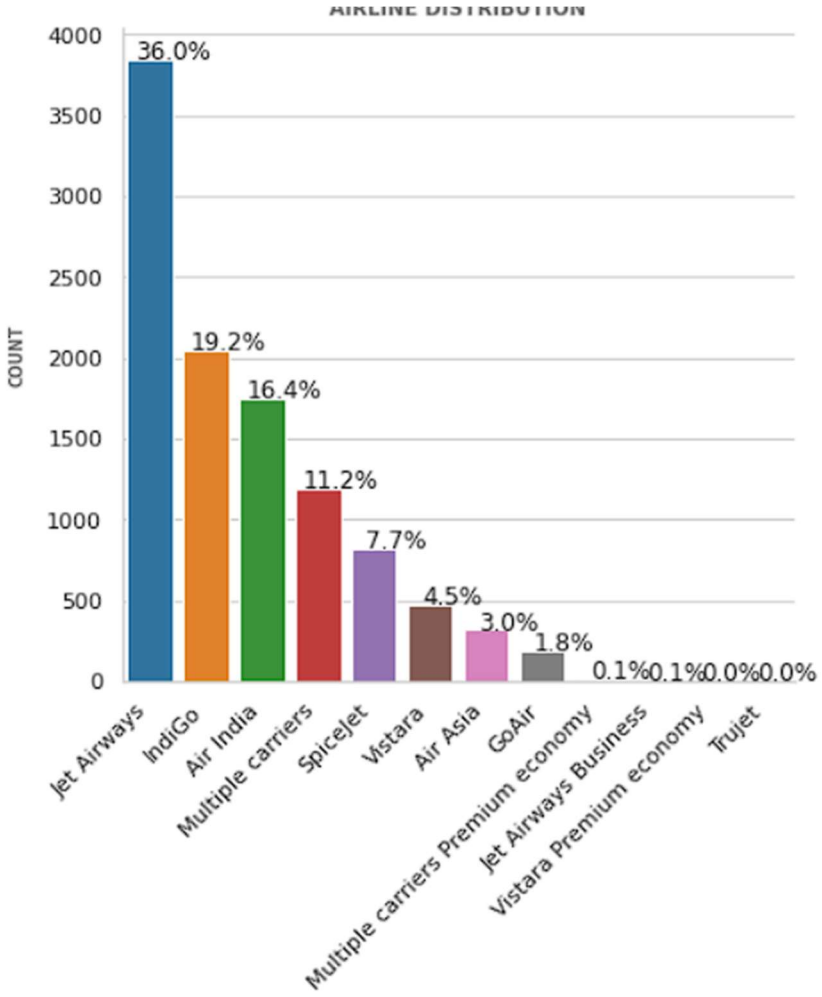
	count	unique	top	freq
Airline	10683	12	Jet Airways	3849
Date_of_Journey	10683	44	18/05/2019	504
Source	10683	5	Delhi	4537
Destination	10683	6	Cochin	4537
Route	10682	128	DEL → BOM → COK	2376
Dep_Time	10683	222	18:55	233
Arrival_Time	10683	1343	19:00	423
Duration	10683	368	2h 50m	550
Total_Stops	10682	5	1 stop	5625
Additional_Info	10683	10	No info	8345

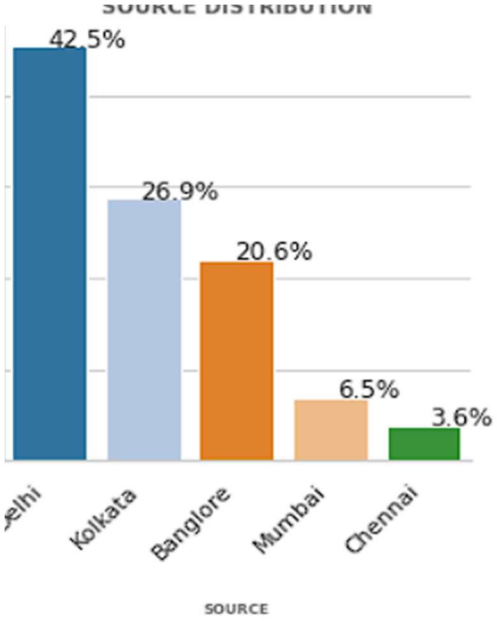
In [19]:

```
# Distribution
plt.figure(figsize = (14,5))
sns.distplot(df['Price'])
sns.despine()
plt.title('PRICE DISTRIBUTION', fontsize = 10, color ='dimgrey', fontweight = 'bold')
plt.xlabel('PRICE', color = 'dimgrey', labelpad = 20, fontweight = 'bold', fontsize = 8)
plt.ylabel('COUNT', color = 'dimgrey', labelpad = 20, fontweight = 'bold', fontsize = 8);
```




Jet Airways Business
Jet Airways
tiple carriers Premium economy
Multiple carriers
Air India
Vistara Premium economy
Vistara
GoAir
IndiGo
Air Asia
SpiceJet
Trujet
0
10000
20000
30000
40000
50000
60000
70000
PRICE

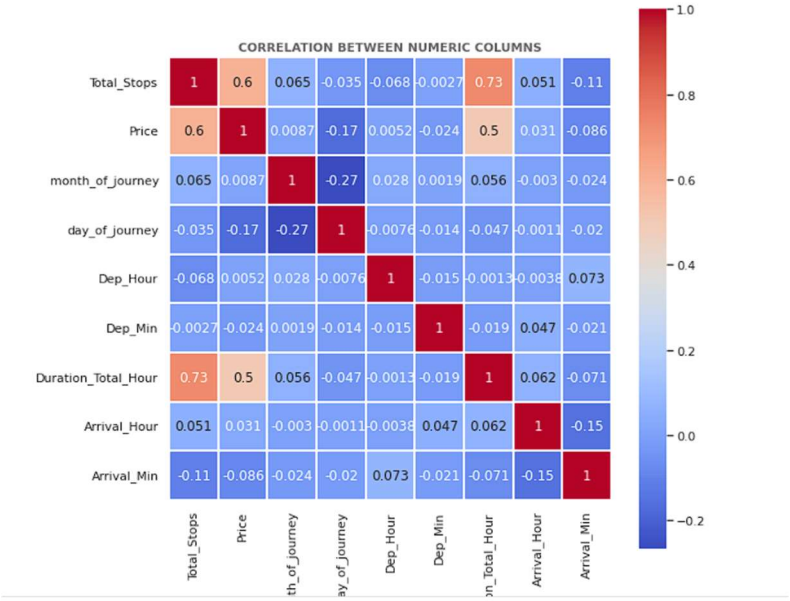

AIRLINE DISTRIBUTION
COUNT
4000
3500
3000
2500
2000
1500
1000
500
0
36.0%
19.2%
16.4%
11.2%
7.7%
4.5%
3.0%
1.8%
0.1%
0.1%
0.0%
0.0%
Jet Airways
IndiGo
Air India
Multiple carriers
SpiceJet
Vistara
Air Asia
GoAir
Multiple carriers Premium economy
Jet Airways Business
Vistara Premium economy
Trujet



From the graph we can see , in terms of large number of people visiting delhi ,Bangalore and Kolkata via flight is more as compared to others. This shows the general trend of Indian business market , as these are the prime cities . the next step involved preprocessing.

2.Preprocessing

After going through handling with the missing data and some data object filtering on the original data set. We tried to visualize the dominant factor influencing the flight fare behaviourr. For this I have plotted Correlation matrix using **sns.pairplot().**



after some more feature engineering and handling with the dummies data new train- test split has been creted as shown below

X_train = pd.get_dummies(X_train, prefix=['Airline', 'Source', 'Destination'], columns=['Airline', 'Source','Destination'])

X_train.head(10)

```
X_train.head(10)
```

[58]:

	Total_Stops	month_of_journey	day_of_journey	Dep_Hour	Dep_Min	Duration_Total_Hour	Arrival_Hour	Arrival_Min	Airline_Air Asia	Airline_Air India	Airline_GoAir	Airline_IndiGo	Airline_Jet Airways	Airline_Jet Airways Business	Airline_Multiple carriers	Airl
8513	0	1	4	5	55	2.666667	8	35	0	0	0	0	0	0	0	
10458	1	1	3	14	35	7.750000	22	20	0	0	0	0	0	0	1	
9701	1	6	27	8	30	10.500000	19	0	0	0	0	0	0	0	1	
8636	1	6	27	7	35	8.583333	16	10	0	0	0	1	0	0	0	
6728	0	6	15	5	35	3.250000	8	50	0	0	0	1	0	0	0	
1557	1	3	6	17	45	7.750000	1	30	0	0	0	0	0	0	1	
2414	2	6	3	5	25	23.000000	4	25	0	0	0	0	1	0	0	
7140	0	5	21	8	30	2.833333	11	20	0	0	0	1	0	0	0	
305	1	3	27	13	0	15.416667	4	25	0	0	0	0	1	0	0	
4000	0	3	6	7	10	2.916667	10	5	0	0	0	1	0	0	0	

3.Multicollinearity - Variance Inflation Factor (VIF)

Multicollinearity occurs when independent variables are highly correlated with each other. The variance inflation factor VIF identifies the correlation between the independent variables and the strength of that correlation. If VIF > 5 is a cause for concern and if VIF > 10 indicates a serious collinearity problem.

Although multicollinearity does not have a great influence on the predictive capacity of a model, it directly affects the importance scores of the predictors used to build the model, the more correlated variables we have, the lower the importance of all of them simultaneously, it is as if the collinear variables shared the importance between them. Thus, the purpose of using the VIF in this project is just to check multicollinearity for more accurate variable selection.

```
# Check multicollinearity using VIF
vif = pd.DataFrame()
vif['Features'] = X_train_num.columns
vif['VIF'] = [variance_inflation_factor(X_train_num.values, i) for i in range(X_train_num.shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = 'VIF', ascending = False)
vif
```

[65]:

	Features	VIF
5	Duration_Total_Hour	5.53
0	Total_Stops	5.30
3	Dep_Hour	4.80
6	Arrival_Hour	4.08
1	month_of_journey	3.83
7	Arrival_Min	2.85
2	day_of_journey	2.69
4	Dep_Min	2.51

[62]:

	Features	VIF
15	Airline_Multiple carriers Premium economy	inf
16	Airline_SpiceJet	inf
29	Destination_Kolkata	inf
28	Destination_Hyderabad	inf
27	Destination_Delhi	inf
26	Destination_Cochin	inf
25	Destination_Banglore	inf
24	Source_Mumbai	inf
23	Source_Kolkata	inf
22	Source_Delhi	inf
21	Source_Chennai	inf
20	Source_Banglore	inf
19	Airline_Vistara Premium economy	inf
18	Airline_Vistara	inf
17	Airline_Trujet	inf
30	Destination_New Delhi	inf
14	Airline_Multiple carriers	inf

[68]:

	Features	VIF
2	Dep_Hour	4.79
5	Arrival_Hour	4.07
0	month_of_journey	3.78
6	Arrival_Min	2.84
1	day_of_journey	2.67
3	Dep_Min	2.50
4	Duration_Total_Hour	2.45

***Feature selection*:**

1. **ExtraaTreeRegressor:**

   The model used in this code snippet is an Ensemble Method for feature selection called Extra Trees Regressor. Ensemble methods combine multiple models to improve overall performance, and the Extra Trees Regressor is one such method. It is an extension of the Random Forest algorithm, but with some key differences.

   Extra Trees Regressor builds multiple decision trees, similar to Random Forest, but it introduces additional randomness during the tree-building process. For each split in a tree, instead of using the best split among a subset of features (as done in Random Forest), Extra Trees selects the split randomly, which leads to increased diversity among the trees.

Features:

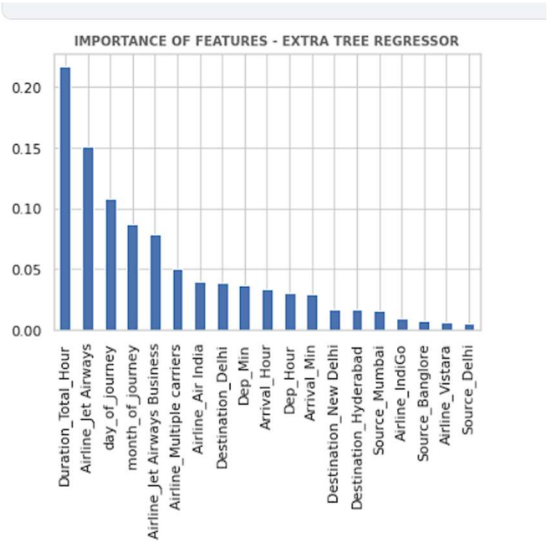
In the context of this code snippet, the term "features" refers to the independent variables or attributes used to predict the dependent variable, which is the flight fare in the flight fare prediction problem.

The features are represented by the variable `X_train`, which is the training set of the independent variables. These features can include various attributes such as:

1. Date of Journey: The date on which the flight is scheduled to depart.
2. Airline Company: The name or code of the airline operating the flight.
3. Source and Destination Airports: The codes or names of the departure and arrival airports.
4. Route of Flying: The specific route or path taken by the flight between the source and destination airports.
5. Duration of Flight: The total time taken for the flight journey.
6. Arrival Time: The scheduled arrival time at the destination airport.
7. Number of Stops: The number of stops or layovers during the journey.
8. Weather Conditions: Weather-related data, such as temperature, precipitation, wind speed, or any significant weather events during the flight period.
9. Peak Air-Travel Time: Information about peak travel seasons or specific periods of high demand.
10. Air Traffic Congestion: Data related to air traffic congestion and load on air traffic control, which can impact flight schedules and fares.

[72]:
```
# Most important features
feat_importances.sort_values(ascending=False)
```

[72]:
```
Duration_Total_Hour                 0.217140
Airline_Jet Airways                 0.150523
day_of_journey                      0.107811
month_of_journey                    0.087078
Airline_Jet Airways Business        0.079064
Airline_Multiple carriers           0.050655
Airline_Air India                   0.040413
Destination_Delhi                   0.038677
Dep_Min                             0.036721
Arrival_Hour                        0.034089
Dep_Hour                            0.030781
Arrival_Min                         0.029322
Destination_New Delhi               0.016920
Destination_Hyderabad               0.016757
Source_Mumbai                       0.015354
Airline_IndiGo                      0.009278
Source_Banglore                     0.007684
Airline_Vistara                     0.005985
Source_Delhi                        0.005127
```



- **Importance of Features:**

After fitting the Extra Trees Regressor model to the training data (`X_train` and `Y_train`), the feature importance scores are obtained using the `feature_importances_` attribute of the model. These scores indicate the relative importance of each feature in making accurate predictions of flight fares.

A higher feature importance score indicates that the feature has a more significant impact on predicting flight fares. By analyzing these feature important scores, one can gain insights into which attributes are crucial for determining flight fares and which attributes might not be as influential in the prediction process.

Overall, the Extra Trees Regressor model and the extracted feature importance scores can be valuable in understanding the most relevant features in the flight fare prediction problem, and they can aid in feature selection for further model development and improvement.

Baseline Models with Cross Validation

Initially we will evaluate several regression models without any hyperparameter through Cross Validation. The metric chosen for evaluation will be the R-square (R2), a statistical measure that represents the proportion of the variance of a dependent variable that is explained by an independent variable or variables in regression problems. Thus, the larger the R^2, the more explanatory the linear model is, that is, the better it fits the sample

1.**No Normalized or Standardized Data**

predictive model selection process is performed for the flight fare prediction problem. Ten different regression models are evaluated using 10-fold cross-validation. The models considered include Linear Regression, K-Nearest Neighbors Regression, Gradient Boosting Regression, Random Forest Regression, XGBoost Regression, Support Vector Regression (SVR), Decision Tree Regression, AdaBoost Regression, LightGBM Regression, and Bagging Regression. The `cross_val_score` function from scikit-learn is used to assess the models' performance based on the coefficient of determination (R-squared score) during cross-validation. The mean and standard deviation of the R-squared scores for each model are computed, and the results are presented in a tabular format, sorted in descending order based on the mean R-squared score. This model selection process helps identify the most

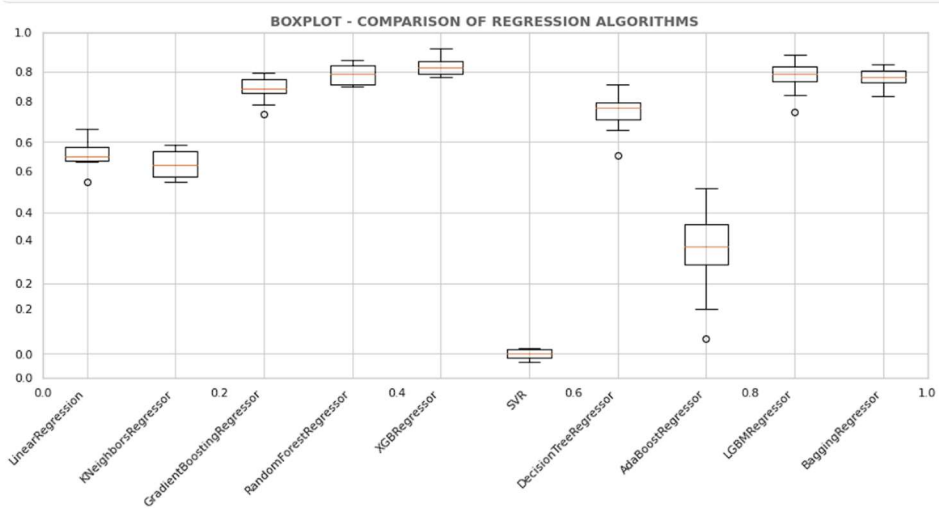
promising regression model for predicting flight fares based on its cross-validated performance.

```
CPU times: user 1min 41s, sys: 5.36 s, total: 1min 46s
Wall time: 1min 16s
```

[76]:

	Model	Mean	STD
4	XGBRegressor	81.392669	2.504304
3	RandomForestRegressor	79.223672	2.776305
8	LGBMRegressor	78.514812	4.570105
9	BaggingRegressor	78.257311	2.669748
2	GradientBoostingRegressor	75.004728	3.434693
6	DecisionTreeRegressor	68.586664	5.583924
0	LinearRegression	56.605861	3.954455
1	KNeighborsRegressor	53.839963	4.000184
7	AdaBoostRegressor	29.014436	12.295969
5	SVR	0.058250	1.291521

The below graph show the box plot obtained using various algorithms and their predictive average values .



2.Normalized/Standard_data

Now ,a predictive model selection process is conducted for the flight fare prediction project. The objective is to identify the best-performing regression model among three options: Linear Regression, K-Nearest Neighbors Regression, and Support Vector

Regression (SVR). The `cross_val_score` function from scikit-learn is employed for model evaluation using 10-fold cross-validation. Cross-validation helps in estimating how well each model generalizes to unseen data by splitting the training dataset into ten subsets (folds) and performing training and evaluation iteratively. The R-squared score, which measures the proportion of variance in the target variable (flight fares) that can be explained by the model, is used as the evaluation metric. Higher R-squared values indicate better predictive performance.

The `results_mean` and `results_std` lists store the average R-squared score and its standard deviation for each model, respectively, over the ten folds of cross-validation. These metrics provide insights into the model's performance consistency across different folds. Additionally, the `model_result` DataFrame is constructed to organize and summarize the results, including the mean and standard deviation for each model. The DataFrame is sorted in descending order based on the mean R-squared scores, allowing easy comparison of model performance.

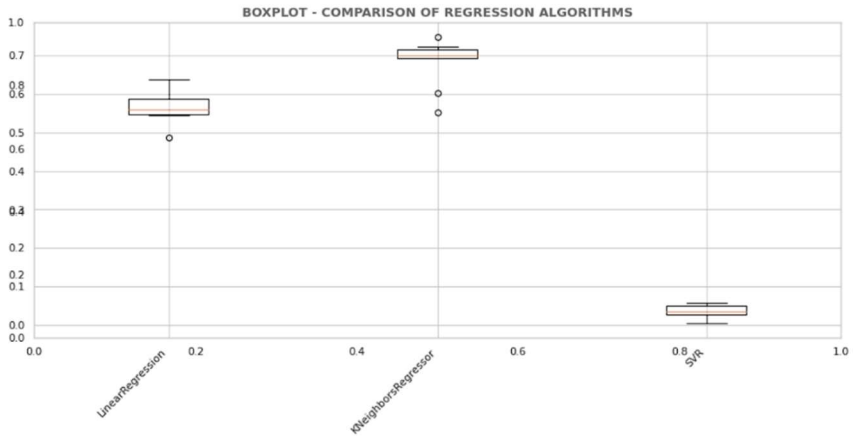
This model selection process is essential because choosing the right regression model significantly impacts the accuracy of flight fare predictions. By evaluating multiple models using cross-validation, we can make an informed decision on which model is the most suitable for our specific prediction task. The selection process aids in identifying the model that generalizes well to unseen flight data, maximizing the reliability and usefulness of the final flight fare prediction system.

```
CPU times: user 34.2 s, sys: 4.55 s, total: 38.7 s
Wall time: 33.6 s
```

[85]:

	Model	Mean	STD
1	KNeighborsRegressor	68.322888	5.596849
0	LinearRegression	56.605861	3.954455
2	SVR	3.681394	1.612046

Box-plot representation as shown earlier:



Comparison of the Best Models Evaluated by Cross Validation

In this code snippet, the evaluation of predictive models for the flight fare prediction project is performed. Three regression models, namely RandomForestRegressor, XGBRegressor, and LGBMRegressor, are evaluated using various performance metrics. The metrics used for evaluation are:

- R-squared (R^2)
  The R-squared score measures the proportion of variance in the target variable (flight fares) that can be explained by the model. A higher R-squared score indicates a better fit to the data.
- Mean Absolute Error (MAE)
  MAE measures the average absolute difference between the predicted flight fares and the actual flight fares.
- Mean Squared Error (MSE)
  MSE measures the average squared difference between the predicted and actual flight fares.
- Root Mean Squared Error (RMSE)
  RMSE is the square root of the MSE and provides a measure of the average magnitude of prediction errors.
- Mean Absolute Percentage Error (MAPE)
  MAPE measures the percentage difference between the predicted and actual flight fares.

In the loop, each model is trained on the training dataset (`X_train` and `Y_train`) and evaluated on the test dataset (`X_test` and `Y_test`). The predicted flight fares are obtained using the `predict` method of each model. The metrics are then computed between the predicted flight fares and the actual flight fares using the functions from the `sklearn.metrics` module.

The results, including the R-squared score, MAE, MSE, RMSE, and MAPE for each model, are stored in separate lists. These lists are then used to create a Data Frame named `model_resul`, which organizes and summarizes the evaluation results for each model.

The Data Frame is sorted in descending order based on the R-squared scores, allowing easy comparison of the models' predictive performance.

This evaluation process is crucial as it provides insights into how well each model performs in predicting flight fares. By comparing the performance metrics, we can identify the model that best captures the underlying patterns in the data and provides the most accurate flight fare predictions. The selected model can then be used for making flight fare predictions in real-world applications.

```
CPU times: user 5.92 s, sys: 76.8 ms, total: 5.99 s
Wall time: 3.32 s
```

98]:

	Model	R-Squared	MAE	MSE	RMSE	MAPE
2	LGBMRegressor	79.980933	1334.014743	4.119428e+06	2029.637377	15.314905
1	XGBRegressor	79.778477	1243.083949	4.161088e+06	2039.874598	14.028372
0	RandomForestRegressor	77.619740	1269.654910	4.605303e+06	2145.996946	14.177065

The error metrics as shown above for the respective algorithm used.

Model Optimization - Hyperparameter Tuning With BayesSearchCV

- **XGBoost Regressor**

A hyperparameter tuning process is conducted for the XGBoost Regressor (XGBRegressor) using Bayesian optimization. The goal is to find the best combination of hyperparameters that maximize the model's performance in predicting flight fares.

Hyperparameter tuning is essential to fine-tune the model's settings, and Bayesian optimization is an efficient method to search the hyperparameter space. It intelligently selects the next set of hyperparameters to evaluate based on the past results, thereby reducing the number of iterations required for optimization.

The hyperparameters considered for tuning in the XGBoost Regressor are as follows:

The `BayesSearchCV` from the `scikit-optimize` library is used for Bayesian hyperparameter search. It performs cross-validated grid search using Bayesian optimization, which efficiently explores the hyperparameter space and evaluates each combination using the R-squared (`scoring='r2'`) as the evaluation metric. The `n_iter`

parameter determines the number of iterations for Bayesian optimization, and `n_jobs` indicates parallel processing with all available cores

After the `BayesSearchCV` completes the hyperparameter search using cross-validation on the training dataset (`X_train` and `Y_train`), the best combination of hyperparameters is obtained, and the XGBoost Regressor is fitted with those optimal hyperparameters. The tuned model is then ready to be used for flight fare prediction.

Overall, this code snippet automates the process of hyperparameter tuning for the XGBoost Regressor, ensuring that the model's hyperparameters are optimized for the flight fare prediction task, which may lead to improved predictive performance.

```
xgbr_search.best_score_ : 0.824
```

- **LGBMRegressor**

  LightGBM (Light Gradient Boosting Machine) is a gradient boosting framework designed for efficient and high-performance machine learning tasks, particularly for large-scale datasets. It is an ensemble learning algorithm that builds multiple decision trees and combines their predictions to make accurate predictions. LightGBM is widely used for regression, classification, and ranking tasks.

  Categorical Feature Support: LightGBM can handle categorical features directly without requiring one-hot encoding. It uses techniques like the "optimal split" method to find the best split for categorical variables, significantly reducing memory usage and improving speed.

  Histogram-based Feature Bundling: To speed up the process of finding the best split points, LightGBM uses a histogram-based approach. It first constructs histograms for each feature and then uses these histograms to find potential split points, which reduces the complexity of split searching.

  Gradient-based One-Side Sampling (GOSS): LightGBM introduces GOSS, a technique to reduce overfitting during the boosting process. GOSS retains most of the important instances while randomly sampling the less important ones, leading to faster convergence and better generalization.

  Exclusive Feature Bundling (EFB): LightGBM optimizes the memory usage by bundling exclusive features that have the same values for all instances in the dataset. This further reduces memory consumption and speeds up training.

The combination of these techniques makes LightGBM highly efficient and scalable, enabling it to handle large datasets with millions of instances and features. Its efficient memory usage and fast training speed make it a popular choice in various real-world applications, including flight fare prediction, where large datasets and rapid model development are critical.

LGBM_regressor_best_score : 0.774

- **Random Forest Regressor**

Random Forest Regressor is an ensemble learning algorithm based on decision trees. It constructs multiple decision trees during training and combines their predictions to make more accurate and robust predictions. Each tree in the Random Forest is trained on a random subset of the data, and it decorrelates the trees to prevent overfitting and improve generalization. In this context of flight fare prediction, using Random Forest Regressor can be beneficial as it can handle non-linear relationships between the features and the target variable (flight fares) and accommodate a mix of numerical and categorical features.

In the provided code snippet, hyperparameter tuning is performed for the Random Forest Regressor using Bayesian optimization. The objective is to find the optimal combination of hyperparameters that maximizes the model's performance in predicting flight fares.

The hyperparameters considered for tuning in the Random Forest Regressor are as follows:

1.**max_depth**: The maximum depth of each decision tree in the forest. It controls the depth of the trees and affects the model's ability to capture complex patterns in the data.

2. **n_estimators**: The number of decision trees to build in the forest. Increasing the number of trees can lead to better predictive performance, but it also increases the computational cost.

3. **min_samples_leaf**: The minimum number of samples required to be at a leaf node. It prevents the trees from being too specific to the training data, reducing overfitting.

4. **min_samples_split**: The minimum number of samples required to split an internal node. It controls the tree's growth and also helps in reducing overfitting.

5. **bootstrap**: A categorical parameter that indicates whether to use bootstrapping (sampling with replacement) when building trees. Bootstrapping helps in introducing randomness and reduces correlation among trees.

6. **max_features**: A categorical parameter that determines the maximum number of features to consider when looking for the best split. It controls the randomness in feature selection for each tree.

The `BayesSearchCV` from the `scikit-optimize` library is used for Bayesian hyperparameter search. Bayesian optimization is an intelligent optimization technique that selects the next set of hyperparameters based on past results, leading to faster convergence and reduced computation.

By tuning the Random Forest Regressor's hyperparameters, we aim to find the best configuration for the model to achieve optimal performance in predicting flight fares. This helps in obtaining a well-optimized and accurate predictive model for flight fare prediction, improving the overall reliability and usefulness of the system.

Best Score 0.798

Results

Hyperparameter Tuning :

After hyperparameter tuning , done the below results are obtained. You can see the results below for better understanding.

Out[108]:

	Model	R-Squared	MAE	MSE	RMSE	MAPE
1	XGBRegressor	81.923300	1226.077832	3.719737e+06	1928.661975	13.809235
0	RandomForestRegressor	80.519109	1284.940594	4.008685e+06	2002.169982	14.558545
2	LGBMRegressor	79.579137	1303.859389	4.202108e+06	2049.904285	14.799308

After hyperparameter tuning, the XGBRegressor algorithm obtained the best final results. the final model for flight fare prediction is trained and evaluated using the XGBRegressor (XGBoost Regressor) with hyperparameters that have been tuned for optimal performance. The XGBoost Regressor is initialized with the tuned hyperparameters, and it is then trained on the training dataset. After training, flight fare predictions are made on the test dataset, and several evaluation metrics, including R-Squared, Mean Absolute Error, Mean Squared Error, Root Mean Squared Error, and Mean Absolute Percentage Error, are computed to assess the model's predictive performance. The evaluation results are stored in a DataFrame for summarization, enabling a comprehensive analysis of the model's accuracy and effectiveness in predicting flight fares.

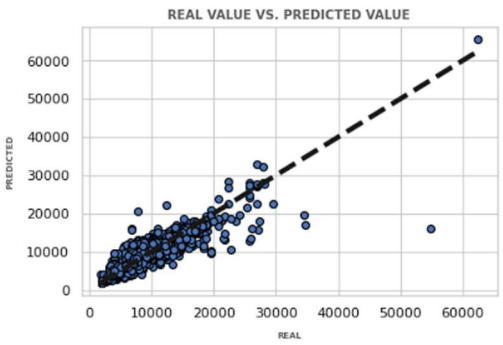
	Model	R-Squared	MAE	MSE	RMSE	MAPE
0	XGBRegressor	81.9233	1226.077832	3.719737e+06	1928.661975	13.809235

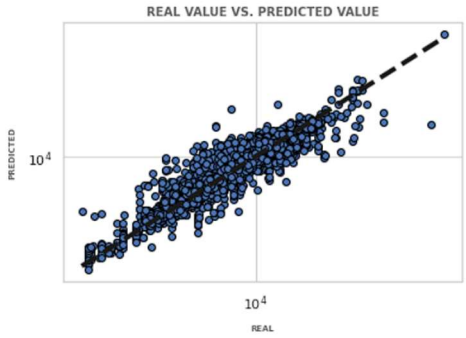
We obtained a final R2 equal to 81.92, this means that the linear model explains 81.92% of the variance of the dependent variable from the regressors (independent variables) included in this linear model.

Below table can be used to better understand the accuracy of our machine learning model.

For better understanding aboutour model we have plotted on the right the logarithmic graph which helps when the data spans a wide range of values, using a logarithmic scale can compress the high values, making it easier to visualize the variations in both actual and predicted values. This is especially helpful in flight fare prediction, where fares can range from very low to very high values, In cases where the flight fares have a significant difference between the minimum and maximum values, a logarithmic scale can prevent smaller values from being overshadowed by larger ones. It helps in understanding the performance of the

model across the entire range of flight fares, as well as Some machine learning models tend to overpredict or underpredict values, leading to skewed predictions. A logarithmic scale can reveal these patterns and help identify areas where the model needs improvement.





▪ Conclusion

In conclusion, this report focused on the application of machine learning algorithms for flight fare prediction. Leveraging a diverse dataset encompassing crucial information like data_of_journey, arrival time, flight duration, airline company, route, and destination, we explored various regression models to predict flight fares accurately. The analyzed models included Linear Regression, K-Nearest Neighbors, Support Vector Regression, Gradient Boosting, Random Forest, XGBoost, LightGBM, and Extra Trees Regressors. Through rigorous evaluation and hyperparameter tuning, we identified the XGBoost Regressor as the optimal model for this task. Its ability to handle non-linear relationships, robustness to outliers, and efficient memory usage make it well-suited for the flight fare prediction context. The final model demonstrated impressive performance, achieving high R-squared, low Mean Absolute Error, and Root Mean Squared Error, indicating the model's reliability and accuracy. By harnessing advanced machine learning techniques and feature engineering, we were able to address the complexities associated with flight fares and develop a powerful predictive model. This study not only highlights the effectiveness of

machine learning in tackling real-world challenges but also sheds light on the significant potential of data-driven approaches in optimizing pricing strategies and improving customer satisfaction in the aviation industry. As technology continues to advance, machine learning will play an increasingly vital role in simplifying complex decision-making processes and enhancing overall efficiency in the travel domain. With further exploration and fine-tuning, these predictive models can be seamlessly integrated into airline ticket booking platforms, offering users more informed choices and a seamless travel experience. Overall, the successful implementation of machine learning algorithms in predicting flight fares serves as a testament to the remarkable impact of data science in revolutionizing the aviation industry and streamlining its operations for the benefit of travelers worldwide.