MARMARA UNIVERSITY
FACULTY OF ENGINEERING

Development of Intelligent Flight Ticket Cancellation Recommendation Engine Using Machine Learning

Mehmet Safa Yılmaz-150318029

Mehmet Ebrar Karademir-150318045

Oğuzhan Zengin-150318057

Koray Pehlivan-150318054

GRADUATION PROJECT REPORT
Department of Industrial Engineering

Supervisor
Prof. Dr. Özlem Şenvar

ISTANBUL, 2023

MARMARA UNIVERSITY
FACULTY OF ENGINEERING

Development of Intelligent Flight Ticket Cancellation Recommendation Engine Using Machine Learning

by

Koray Pehlivan, Oğuzhan Zengin,Mehmet Safa Yılmaz,

Mehmet Ebrar Karademir

05 June 2023, Istanbul

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE AT MARMARA UNIVERSITY

Accepted by Prof. Dr. Özlem Şenvar

Project Supervisor, Department of Industrial Engineering

Certified by Asst. Prof. Dr. Merve Er

Jury Member, Department of Industrial Engineering

Certified by Res. Assist. Dr. Murat Bilsel

Jury Member, Department of Industrial Engineering

ACKNOWLEDGEMENTS

To our esteemed advisor, Prof.Dr.Özlem Şenvar, who transferred her knowledge and experience to me during my undergraduate education;

To the esteemed Aleyna KARACA, Mehmet Ayberk BAYRAMİN, Bedriye BAL, Mustafa BAL and Eralp KANER who have always helped me during my studies;

We would like to express our gratitude to our families, who have always stood by us under all circumstances, never spared their support, and enabled us to achieve these successes.

June, 2023

Koray PEHLİVAN

Mehmet Ebrar KARADEMİR

Mehmet Safa YILMAZ

Oğuzhan ZENGİN

TABLE OF CONTENTS

ABSTRACT

This study is conducted on the ticket cancellation service, which is used in the airline and other transportation sectors and has been in greater demand recently due to the COVID-19 pandemic. The study utilizes data sets related to flights and passengers from Turna.com, a travel agency and the aim of the study is to present a solution to optimize the price of the ticket cancellation service using machine learning and artificial intelligence models by following a systematic approach consisting of, data collection, preprocessing, feature engineering, model selection, evaluation. Various classification techniques such as logistic regression, decision trees, random forest are explored to provide accurate predictions and improve decision making. The application of machine learning in this area provides valuable insights for providers, allowing them to adapt to dynamic market conditions and improve overall service quality while maintaining financial stability. The main objective of the study is to predict the risk level passengers can pose to the service provider based on their behaviors. In other words, the study aims to construct a machine learning model that can predict users' ticket cancellation behavior before they exhibit such behavior, and to use this model in the company's future pricing strategy for this service. Throughout the study, the sustainability of the solution and the benefits of the proposed method in terms of sustainability have been discussed and examined. The study concludes by providing recommendations and suggestions for further advancement and improvement of the solution.

Key words: Cancellation insurance service, airline transportation, machine learning, decision trees, random forest, decision-making.

ÖZET

Bu çalışma, havayolu ve diğer taşımacılık sektörlerinde kullanılan ve COVID-19 pandemisi nedeniyle son zamanlarda daha fazla talep gören bilet iptal hizmeti üzerinde yapılmıştır. Çalışma, seyahat acentesi Turna.com'un uçuşlar ve yolcularla ilgili veri setlerinden faydalanmaktadır. Çalışmanın amacı, makine öğrenmesi ve yapay zeka modelleri kullanarak bilet iptal hizmetinin fiyatını optimize etmek için bir çözüm sunmaktır. Veri toplama, ön işleme, özellik mühendisliği, model seçimi, değerlendirme gibi adımlardan oluşan sistemli bir yaklaşımı takip ederek, lojistik regresyon, karar ağaçları, rastgele orman gibi çeşitli sınıflandırma teknikleri kullanılarak doğru tahminler sunulmuş ve karar verme süreci geliştirilmiştir. Bu alanda makine öğreniminin uygulanması, sağlayıcılara değerli içgörüler sağlayarak, dinamik piyasa koşullarına adapte olmalarını ve genel hizmet kalitesini artırmalarını sağlamaktadır. Çalışmanın ana amacı, yolcuların davranışlarından yola çıkarak onların hizmet sağlayıcıya verebileceği risk düzeyini tahmin etmektir. Diğer bir deyişle, çalışma, kullanıcıların bilet iptal davranışını sergilemeden önce bu davranışı tahmin edebilen bir makine öğrenmesi modeli inşa etmeyi ve bu modeli şirketin bu hizmet için gelecekteki fiyatlandırma stratejisinde kullanmayı hedeflemektedir. Çalışma boyunca çözümün sürdürülebilirliği ve önerilen yöntemin sürdürülebilirlik açısından sağladığı faydalar tartışılmış ve incelenmiştir. Çalışma, çözümün ilerlemesi ve geliştirilmesi için öneriler ve tavsiyeler sunarak sonuçlanmaktadır.

Anahtar kelimeler: İptal sigortası hizmeti, havayolu taşımacılığı, makine öğrenmesi, karar ağaçları, rastgele orman, karar verme.

LIST OF SYMBOLS

h_i **:** Single decision tree model

N_r **:** Number of objects in each leaf

$Z_{spec,i}$: Low mean squared error for the spectroscopic redshift

$f_k(x_i)$: The set of functions

fn_i : False negative for C_i;

fp_i : False positive for C_i;

i^{th} : Sample is represented by the function

k^{th} **:** The value of the forecast by tree

l_L **:** The instance sets of the split left nodes

l_R **:** The instance sets of the split right nodes

r_{im} : The negative gradients

tn_i : True negative for C_i;

tp_i : True positive for C_i;

w_j^* : Weight of leaf j

x_i : The i^{th} training instance

y_i : The goal value of the loss function

Θ_k : Random variable corresponding to the kth

M : Macro averaging.

$H(x)$: Model combination for classification

T : The quantity of leaves

v : The learning rate

Ω : The regularization term

γ : The regularization gradations

$\boldsymbol{\lambda}$ **:** The regularization gradations

m : The vector of values assigned to each leaf

ABBREVIATIONS

AHP **:** Analytic hierarchy process

AI **:** Artificial intelligence

ANP **:** Analytic network process

ARC **:** Airline reporting corporation

CART **:** Classification and regression trees

CPS **:** Cancellation protection service

CPU **:** Central process unit

DBSCAN **:** Density-based clustering algorithm

DDCM **:** Dynamic discrete choice model

DTPO **:** Discrete time proportional odds

GBM **:** Gradient boosting machine

GDPs **:** Ground delay programs

GM **:** Geometric-mean

GOSS **:** Gradient-based one-side sampling

GRU **:** Gated recurrent unit

IQR **:** Interquartile range

LightGBM **:** Light gradient boosting machine

LSTM **:** Long-short term memory

MAPE **:** Models' average absolute percentage errors

MAR **:** Missing at random

MCAR : Missing completely at random

MCMC **:** Markov chain monte carlo

MNL **:** Multinomial logit

MSE : Mean squared error

Obj **:** Objective

p **:** Precision

PNR **:** Passenger name record

PODS **:** Proper orthogonal decomposition system

Q1 **:** First quartile

Q3 **:** Third quartile

QoE **:** Quality of experience

R&D **:** Research and development

r **:** Recall

RF **:** Random forest

RFM **:** Random forest measure

sn **:** Sensitivity

SOM **:** Serviceable obtainable market

SP **:** Service provider

sp **:** Specificity

SVM **:** Support vector machines

TMIs **:** Traffic management initiatives

UK **:** United Kingdom

US : United States

XGBoost **:** Extreme gradient boosting

LIST OF FIGURES

LIST OF TABLES

1.INTRODUCTION

The airline passenger transportation industry is a transportation industry that offers airline passengers fast, safe and easy travel options. It has always played an important role for countries in terms of facilitating tourism and economy. In addition to its positive contributions to the states, it also contributes to people both economically and culturally. Due to this contribution, airline passengers are high for the aviation industry. Airlines and travel agencies that provide this transportation service meet the service demand of people.

In line with these techniques, airlines and travel agencies apply different pricing policies in order to meet the demand. While creating these pricing policies, they take into account variables such as external factors affecting ticket prices, competition between airlines, the dynamic behavior of ticket pricing, and differences in pricing as a result of the economic policies of the states. For example, factors such as flight distance, class of service, airline, departure/arrival times, seasonality affect the pricing policy of the ticket and the purchasing behavior of the customer. (Lantseva, Mukhina,Nikishova,Ivanov & Knyazkov, 2015)

In particular, the COVID-19 pandemic has affected many sectors as well as the aviation industry. Airline passenger transport and airport traffic have been deeply impacted by the COVID-19 pandemic in 2020 regarding the health and safety of passengers and employees, operational and financial results due to border closures, travel restrictions, quarantine requirements and the declining air travel demand it has caused. (Senvar & Cagin,2022). Airlines and travel agencies were negatively affected by this situation. People who provide transportation by aviation, such as airlines and travel agencies, have been negatively affected economically by this process. A new demand has arisen due to the fact that the tickets purchased by the people were burned due to external or internal factors and they could not receive all or part of the ticket fees as refunds.

Airlines and travel agencies have offered cancellation insurance service, also known as cancellation protection service, as a solution to these problems that airline passengers may experience (Sadreddini, 2020). In the travel insurance industry, cancellation insurance for any reason, also known as cancellation protection service, has recently been trying to strike a balance between customer satisfaction and the profit of the service provider (Sadreddini, Dönmez & Yanikomeroğlu, 2021). While this insurance service is beneficial for customers, it poses various problems for providers.

1.1. Project Content

This study handles unconditional flight ticket cancellation and pricing policy making involving cancellation insurance for airline passengers The purpose of this study is to construct a machine learning model that can predict airline passengers' ticket cancellation behavior before it occurs. A real case study is conducted using data sets of Turnacom, which is serving as travel agency for selling online domestic and international flight tickets to increase the maximum profit via unconditional ticket cancellation service in which airline passengers can get 90% refund of the ticket price 2 hours before the flight time. Similar to the unconditional flight ticket price estimation problems in terms of airline passengers behavior in the studies carried out within the scope of unconditional flight ticket cancellation or ticket cancellation insurance, Turnacom offers this service to more airline passengers and a dynamic price parallel to airline passengers behaviors. cannot determine policy. In this study, an approach will be developed in order to predict the ticket cancellation behavior of airline passengers before they occur by focusing on airline passenger behaviors in the light of similar studies in the literature by using science-based approaches in line with these problems.

2. RESEARCH OBJECTIVE

The airline passenger transportation sector plays an important role in facilitating tourism and contributing to the national economy. In order to meet the demand for air travel, airlines, and travel agencies apply various pricing policies, taking into account factors such as external effects on ticket prices, competition, and economic policies. However, the COVID-19 pandemic has severely impacted the industry, causing financial losses for airlines and travel agencies, while creating demand for cancellation insurance services. This study aims to help cancellation insurance service providers, who aim to strike a balance between customer satisfaction and profitability, in predicting the cancellation behavior of their customers.

In order to solve this problem, a literature review of studies on similar subjects in the past has been made. In line with these studies, it was predicted that a methodology including machine learning techniques would be more suitable for predicting the cancellation behavior of customers. Machine learning, a subset of artificial intelligence, enables computers to learn from data and improve performance without explicit programming. The methodology consists of several steps, including problem identification, data collection, data preprocessing, feature engineering, model selection, model training, model evaluation, model deployment, and model monitoring and maintenance.

Classification problems are addressed in this study, where the aim is to predict the class or category of new observations based on their attributes. Various classification techniques are explored, including logistic regression, decision trees, random forest, support vector machines, and neural networks. The choice of technique depends on the nature of the problem, data size, complexity, and performance requirements.

By applying this methodology, cancellation insurance service providers can learn about customer behavior, improve pricing policies, and increase customer satisfaction while maintaining profitability. The application of machine learning techniques enables efficient analysis of vast amounts of data and provides accurate predictions to make better decisions in the airline passenger transport industry.

2.1. Sustainability

The arguments about how to balance preserving the environment with promoting economic growth gave rise to the idea of "sustainability" (Ryan and Throgmorton, 2003). It was discussed in the 1987 "Our Common Future" report of the World Commission on Environment and Development (WCED). According to the report, sustainable development is "development that satisfies present demands while preserving the capacity of future generations to satisfy their own needs" (WCED, 1987).

Additionally, according to the WCED report's definition, the core principle of sustainability is to balance social, environmental, and economic development. These three ideas do not contradict one another. Instead of emphasizing the tension between environmental protection and economic growth, WCED promotes the idea that both of these objectives are genuinely possible (Daley, 2010). Although it is still applied to various contexts (such as financial sustainability) and still interpreted from various angles today (Budd et al., 2013), the idea was originally connected to the environment (Forsyth, 2011). From this vantage point, sustainable development places a strong emphasis on the necessity of guaranteeing just and equal social and economic growth among communities while simultaneously controlling environmental effects. The primary principle of it is that sustainable development shouldn't diminish the natural resource base on which human populations rely to exist (Daley, 2010; Konuralp, 2020).

The transportation system, which generates noise, atmospheric pollution, and consumes substantial amounts of land, is one of the systems that has the greatest negative effects on the environment. Additionally, it is dependent on fossil fuels (Graham and Guyer, 1999; Walters et al., 2018). This makes an emphasis on the transportation sector within the context of sustainability inevitable, both in the realm of academia and in real-world applications. Air transport is one of the components of the transportation system that has the potential to have an impact on the environment. Global air travel contributes around 2% of all human-caused carbon dioxide (CO2) emissions, although having a lower overall environmental impact than vehicle travel.

By developing systems that protect the environment, advance economic value, and raise the standard of living for people, the aviation industry may become more sustainable.

The three main elements of aviation sustainability are: a) environmental sustainability, which is the reliance on natural resource systems; b) economic sustainability, which is the capacity, improvement, and manageability of the economy; and c) social suitability, which is the social righteousness, security, and superiority of life (Alameeri et al., 2017).

More than 9 million passengers traveled every day on more than 100,000 aircraft over a network of about 51,000 routes prior to the recent, unprecedented COVID-19-related disruption in international air travel (O'Connell, 2018). These passengers carried goods worth $17.5 billion to industry and households. In 2019, 4.5 billion people were transported by airlines around the world, generating $838 billion in revenue (ATAG, 2020). (IATA, 2020a). Around 88 million people were employed directly by the industry in aviation and related tourism in the same year (IATA, 2020b) (ATAG, 2020).

Even though it has historically expanded quickly and now plays a significant role in facilitating international travel, the airline industry has also shown a slowdown in fuel efficiency improvements, making it one of the sectors of the global economy that is emitting greenhouse gasses (GHG) at the fastest rate (Kim et al., 2019). According to ATAG, 2020, aviation contributes 3.5% to global warming when non-CO2 effects are taken into account, accounting for around 2% of total human-caused carbon dioxide emissions (Lee et al., 2020; cf. Larsson et al., 2018). Airlines must be responsive to all of their stakeholders through reporting because the industry has one of the largest stakeholder groups of any organization (PwC, 2011).

In the light of the above information, it is necessary to talk about the effects of this study on sustainability and the improvements that can be made. One potential sustainability benefit of providing insurance for canceled tickets is that it may encourage travelers to purchase tickets with more flexibility and cancellation options. This can reduce the number of last-minute cancellations, which can have a negative environmental impact due to the emissions from rebooking flights or finding new transportation at the last minute. By offering insurance, travelers may feel more confident in booking tickets with flexible cancellation policies, which can ultimately reduce the number of last-minute cancellations and the associated emissions.

However, it is important to note that the environmental impact of insurance services also needs to be considered. Insurance companies often use complex computer models and

data analysis to assess risk and set premiums, which requires energy and resources. Additionally, the process of underwriting and managing insurance policies also requires resources, such as paper and office supplies. To minimize the environmental impact of insurance services in the ticket cancellation process, insurance companies can adopt sustainable business practices, such as using renewable energy sources and implementing paperless systems. Travel companies can also encourage the use of digital insurance documents and the use of eco-friendly materials in the production of physical documents.

Considering all these sections, if we need to categorize the relationship of our project with its sustainability;

1. Reduced Carbon Emissions: Flight cancellations often result in empty seats on planes, which means wasted fuel and increased carbon emissions. Unconditional ticket cancellation services could allow passengers to cancel their flights in advance without penalty, thus reducing the number of empty seats and optimizing flight capacity. This, in turn, can lead to a reduction in overall carbon emissions associated with air travel.

2. Increased Efficiency in Flight Planning: With unconditional ticket cancellation services, passengers may feel more comfortable booking flights well in advance, knowing they have the flexibility to cancel if needed. This increased confidence can lead to more accurate passenger demand forecasts, allowing airlines to optimized flight schedules, reduce overbooking, and operate flights with higher occupancy rates. Consequently, this can improve fuel efficiency and decrease waste within the aviation industry.

3. Shift Towards Sustainable Alternatives: Unconditional ticket cancellation services might encourage travelers to consider alternative modes of transportation for shorter distances. If passengers have the option to cancel their flights without penalties, they may opt for trains, buses, or other forms of transportation that are more energy-efficient and have lower carbon footprints. This shift could contribute to a modal shift towards more sustainable travel options.

4. Customer Behavior and Awareness: The optimization of unconditional ticket cancellation services can raise awareness among travelers about the environmental

impact of air travel. By offering the option to cancel tickets, airlines and online platforms can educate customers about carbon offsets, sustainable travel practices, and the importance of reducing their overall travel footprint. This increased awareness can influence customer behavior, leading to more sustainable choices in the future.

5. Innovation and Industry Responsiveness: The introduction of unconditional ticket cancellation services can stimulate innovation within the airline industry. Airlines may invest in more fuel-efficient aircraft, explore alternative energy sources, or adopt new technologies to improve operational efficiency. Online platforms and travel agencies may also promote sustainable travel options, such as eco-friendly accommodations or carbon offset programs. The overall impact could drive the aviation industry to become more sustainable and responsive to environmental concerns.

Overall, while providing insurance services in the ticket cancellation process can have sustainability benefits by reducing last-minute cancellations, it is important to consider the environmental impact of the insurance industry as well. By adopting sustainable business practices and encouraging the use of digital documents, the sustainability impact of insurance services in the ticket cancellation process can be minimized.

3.LITERATURE REVIEW

The last two decades have seen an increasing number of studies targeting both customers and airlines. Customer-side surveys focus on saving money for the customer, while airline-side surveys aim to increase airlines' revenue. Conducted research uses a wide variety of techniques, from statistical techniques such as regression to different kinds of advanced data mining techniques. (Abdella, Zaki, Shuaib & Khan, 2021)

Sadreddini suggests a quality of experience (QoE)-based strategy for promoting customer satisfaction in the context of online flight reservations in the article "A Novel Cancellation Protection Service in Online Reservation System." To ensure early reservations while balancing customer pleasure and service provider (SP) earnings, the author creates a cancellation protection service (CPS). Fixed CPS, QoE-based CPS, and Flexible CPS are the three different CPS functions that are suggested (Sadreddini,2020). The Analytic Hierarchy Process (AHP) is used by the QoE-based CPS technique to give the various criteria the proper weights. Customers may choose to cancel their tickets and obtain a refund in accordance with the suggested CPS procedure. The suggested CPS approach is evaluated using actual data from a reservation system for an airline, and the outcomes demonstrate the impact of the various CPS functions on SP earnings. According to the findings, the SP profit and customer satisfaction are balanced by the QoE-based CPS technique.

Similar to the previous article, Sadreddini et al article's "Cancel-for-Any-Reason Insurance Recommendation Using Customer Transaction-Based Clustering" similarly discusses the difficulty of striking a balance between customer pleasure and SP profitability in the travel insurance sector (Sadreddini, Donmez,& Yanikomeroglu,2021). The authors suggest employing clustering techniques like K-means, hierarchical agglomerative, density-based clustering algorithm (DBSCAN), and self-organizing maps to recommend CPS fees based on customer risk group segmentation (SOM). Based on the cluster segmentation weights and the overall CPS revenue threshold established by the SP, an adaptive CPS approach is utilized to determine CPS fees for each group. Using actual data, it is demonstrated that the suggested methodology can help keep the balance between SP profits and CPS fees.

In the context of non-refundable ticket cancellations, both of these researchs emphasize the significance of striking a balance between customer pleasure and SP profitability in the travel sector. In order to achieve this balance, they both offer strategies

for CPS fee optimization, with the first article using customer risk group segmentation and the second article adopting a QoE-based strategy. These findings demonstrate that giving customer demands and SP profitability considerable consideration can result in CPS policies that are more successful and, eventually, enhance the overall customer experience.

In another article, "The Influence of Ticket Refund Service Towards Air Asia Customers Trust," SA Maulana, MR Gigantama, Lies Lesmini, Imam Ozali, and Cecep Budiman concluded that the ticket refund service greatly affects customer trust in AirAsia. The study employed a descriptive, quantitative research methodology with simple linear regression analysis and utilized both primary and secondary data, including questionnaires and literature reviews. The findings indicated that 80.5% of the variation in consumer trust may be attributed to the ticket refund service (Maulana, Gigantama, Lesmini, Ozali & Budiman,2019). Other elements that affect consumer trust in the aviation business may be the subject of future study.

The decision-making process that consumers go through while buying airline tickets online is examined in the article "Value Driven Decision Forecast Model: An Application on Online Flight Ticket Purchase" (Ozdemir,2015). The study makes use of the cognitive theory-based means-end chain approach, which tries to identify the qualities, advantages, and values that affect a consumer's decision. The study was carried out utilizing in-depth interviews and the laddering technique on a sample of college-educated people between the ages of 25 and 30. To create a model that incorporates individual values into the decision-making process, the data gathered via this procedure was examined using the Analytic Network Process (ANP). The study's findings imply that the ANP decision model, which emulates the means-end chain approach's hierarchical structure and permits the consideration of various relations and connections at the same level, can produce more effective results for high involvement products that demand greater mental effort. The study also showed that the ANP model may accurately predict shifting customer preferences.

The study "Dynamic Discrete Choice Model for Railway Ticket Cancellation and Exchange Decisions examines how railroad passengers behave while making decisions regarding ticket cancellation and exchange in the presence of fare and schedule uncertainty (Cirillo, Bastin & Hetrakul, 2018). The timing of ticket exchanges or cancellations is predicted by the authors using an intertemporal choice model and a dynamic discrete choice model (DDCM). The model is formulated as an optimal halting issue, and the dynamic

programming problem is approximated by a two-step look-ahead strategy. The method is used to analyze actual intercity train ticket reservation data, and the outcomes are contrasted with those from a multinomial logit (MNL) model. The DDCM, according to the authors, offers more understandable results and has superior predictive power than the MNL model (Cirillo, Bastin, & Hetrakul,2018). The strategy created in this study may be used in other sectors, including the railroad business, that employ variable refund policies.

The application of time-to-event approaches for examining airline passenger cancellation behavior is examined in the article "Customer based time -to -event models for cancellation behavior: A revenue management integrated approach". The study examines the effects of various covariates, including time from ticket purchase, time before flight departure, departure day of the week, market, and group size on cancellations. It does this using a discrete time proportional odds model with a prospective time scale and ticketing data from the Airline Reporting Corporation (ARC). The analysis discovers that the covariates indicated above have an impact on cancellations, which are often greater for recently booked tickets and for tickets with close flight departure dates. The study simulates and compares the revenue streams of a single resource capacity control under time-to-event and state-of-practice cancellation forecasts in order to test the hypothesis that cancellation forecasts based on ticketing data generate more revenue. The use of ticketing data, as opposed to booking data, is motivated by the need to analyze cancellation behavior from a financial perspective (Iliescu,2008).

Chiew, Daziano, and Garrow (2017) investigate the application of Bayesian methods to estimate hazard models of airline passenger cancellation behavior in their study titled "Bayesian estimation of hazard models of airline passengers' cancellation behavior." They provide a discrete time proportional odds (DTPO) cancellation model that can be easily estimated using Bayesian methods and reinterpreted as a fixed parameter discrete choice model. This model can be be expanded to account for unobserved heterogeneity by adding random parameters. The Bayesian estimating method enables forecasting of the means and a measure of variance (credible intervals) associated with an individual's cancellation probability in addition to the calculation of individual-specific cancellation probabilities. The technique is used on a dataset of tickets bought by university employees in Atlanta, Georgia, over a two-year period, when the major local airline canceled a deal allowing staff to buy cheap fares that could be returned or swapped without being charged. The findings indicate that when customers must pay to swap their tickets, cancellations are reduced on

average by 3.3%, and the coefficient of variation of cancellation is 43% when the state rate was available against 83% when it wasn't (Chie, Daziano, & Garrow,2017).

The findings of an AI-supported project on dynamic flight price prediction carried out by the Enuygun.com R&D Center are presented in the publication "Developing Flight Price Prediction Models With Artifical Intelligence Technology" .The accuracy of the "GB-FPPredictor" model is 90%, and the accuracy of the "RFFPPredictor" model is 92%, according to the authors, who developed two prediction models called "GB-FPPredictor" and "RFFPPredictor" using machine learning algorithms, specifically gradient boosting (GB) and random forest (RF). When compared to previous models created in the past, the models' average absolute percentage errors (MAPE) are 2.49% and 2.26%, respectively, which the authors consider to be extremely good. The study illustrates the possibility for predicting dynamic airline fares utilizing AI technology, notably machine learning algorithms (Keleş,Keleş & Keleş,2020).

Samir Kumar Bandyopadhyay, Vishal Goyel, and Shawni Dutta offer a prediction model for applying deep learning to predict airplane cancellations in their study, "Prediction of Air Flight Cancellation during COVID-19 using Deep Learning Methods." The COVID-19 pandemic has had a severe and immediate impact on air traffic figures, leading to a fast rise in flight cancellations, aircraft groundings, and travel restrictions, according to the authors (Bandyopadhyay,Goyel & DUTTA ,2020). They also emphasize that air traffic is susceptible to external variables like disease outbreaks. Airlines' revenues have been considerably decreased as a result of the decline in passenger traffic, forcing many of them to lay off staff or file for bankruptcy. The authors suggest a deep learning framework-based predictive model for forecasting airplane cancellations, which combines two distinct recurrent neural networks into a single entity to infer prediction outcomes. The model is designed by the authors using a long-short term memory (LSTM) and a gated recurrent unit (GRU), and they compare it to a conventional multi-layer perceptron model based on a neural network. The findings demonstrate that the proposed model has a 98.7% accuracy rate. This study contributes to the body of knowledge on the difficulties the aviation sector encountered during the COVID-19 epidemic and suggests a viable approach for using deep learning techniques to anticipate airplane cancellations. The authors suggest a deep learning framework-based predictive model for forecasting airplane cancellations, which combines two distinct recurrent neural networks into a single entity to infer prediction outcomes. The model is designed by the authors using a long-short term memory (LSTM) and a gated

recurrent unit (GRU), and they compare it to a conventional multi-layer perceptron model based on a neural network. The findings demonstrate that the proposed model has a 98.7% accuracy rate. This study contributes to the body of knowledge on the difficulties the aviation sector encountered during the COVID-19 epidemic and suggests a viable approach for using deep learning techniques to anticipate airplane cancellations.

Another study, "Flight Cancellations and Airline Alliances: Empirical Evidence from Europe," looks at the connection between airline alliances and flight cancellations in the European airline sector. Although airline on-time performance is frequently used as a gauge of service quality, the authors note that flight cancellations, which can cause passengers more hardship, have gotten less attention.They point out that since cancellations can be caused by a variety of circumstances, such as low passenger demand, operational difficulties, and economic concerns, airlines may behave strategically when deciding whether or not to cancel a flight. According to the authors, airline alliance membership may have an impact on an airline's attitude toward cancellations because it can lessen market discipline and competition, increase hubbing benefits by increasing the number of destinations, and possibly result in economies of scale in the management of irregular operations. By statistically analyzing the connection between airline alliance membership and flight cancellations in Europe, the study seeks to close a gap in the literature. The study's findings demonstrate that airline alliance participation significantly affects flight cancellations in the European airline sector. The authors discovered that being a part of a worldwide alliance considerably raises the likelihood of a flight being canceled using a sample of flights from 2010 to 2013. The findings also indicated that a number of factors can influence the likelihood of a flight being canceled, including hub status, low-cost carrier status, airport volume, and the day of departure. For instance, the authors discovered that low-cost carriers had a lower probability of cancellation than larger airports and flights that were scheduled for the weekdays. The findings also indicated that the time of scheduled departure had an impact on flight cancellation, with midday flights having a higher likelihood of cancellation than morning or evening flights. According to the authors, these findings can be explained by airlines' strategic conduct and the trade-offs they make between operating a scheduled flight and canceling it depending on things like the benefits of hubbing and the costs of disruption (Alderighi & Gaggero,2018).

Airline cancellations can have a big effect on revenue management since they may result in planes taking off with empty seats and lost money. Airlines need precise cancellation

forecasting systems to make up for cancellations and overbook flights beyond their physical capacity in order to solve this problem. However, excessive overbooking can lead to refused boardings and compensation expenses, which might balance any potential income increases. Oren Petraru investigated the use of time series modeling to estimate passenger cancellations and the effects on revenue management in a paper that was published in 2016. Petraru investigated a number of cancellation forecasting techniques based on historical data using the PODS booking simulation program, including the potential value of Passenger Name Record (PNR) data.According to the simulation's findings, ticket revenue increases due to overbooking and cancellation predictions ranged between 1.15% and 4.16%, while net revenue increases fell between 0.06% and 2.79% (Petraru,2016). The increases in net revenue for airlines with greater cancellation rates reached 3.59%. According to Petraru's analysis, airlines, particularly those with higher cancellation rates, can boost their revenues by using cancellation predictions. As long as overbooking is not used excessively, revenues can be increased even though different strategies for projecting cancellations may produce comparable outcomes. To strike the best balance between revenue gains and costs, compensation charges as well as the risks of refused boardings must be taken into account. Furthermore, expanding the seating options,especially in low fare classes,may reduce yields and should be carefully taken into account in the revenue management approach (Petraru,2016).

Considering the relevance of Airline cancellations to the research project, an extensive literature review was conducted to examine Airline cancellations in the literature. Through this review, a number of studies were identified that sought to address the issue of airline cancellations and explore possible solutions to this problem. The study "An investigation into the determinants of flight cancellations" focuses on the importance of passenger demand forecasting and passenger cancellation forecasting in any airline revenue management system. It highlights the fact that cancellations can lead to flights leaving with empty seats and result in loss of revenue. To prevent this, the study suggests that accurate cancellation forecasting tools are necessary for airlines to properly compensate for cancellations and overbook flights above their physical capacity. However, it also stresses the importance of being cautious not to overbook too aggressively as it can result in denied boardings and additional costs. The study uses the proper orthogonal decomposition system(PODS) booking simulation tool and various methods for cancellation forecasting and overbooking, and also discusses the potential contribution of Passenger Name Record data

to more accurate cancellation forecasting. The results indicate that the revenue gains from cancellation forecasting and overbooking range between 1.15% and 4.16%, however, aggressive overbooking can increase negative effects on revenues. Therefore, for airlines with high cancellation rates, the magnitude of the gains from cancellation forecasting and overbooking is even greater, reaching 3.59% in net revenue improvements. (Rupp & Holmes,2006)

Another paper "A study of flight cancellation and delays in the UK" the author uses this study to examine the problem of flight delays and cancellations in the aviation industry, specifically in the UK, and how research has primarily focused on studying and predicting delays. The author argues that understanding and studying the underlying patterns of cancellations is necessary as it is the most important determinant for consumer dissatisfaction and complaints, which can be detrimental for airlines' reputation and result in passengers switching carriers. The author also notes that there is a lack of clear understanding of cancellations, specifically the need for a thorough time series analysis and the relationship between delays and cancellations. Additionally, the author suggests that the outbreak of COVID-19 has made forecasting more difficult and there is a need to take into account behavioral changes of the population to understand the impact of the pandemic on air travel. (Vázquez Ibáñez,2022)

Also in another paper "Modelling airline flight cancellation decisions studies the responses of US domestic airlines to Traffic Management Initiatives (TMIs), specifically Ground Delay Programs (GDPs), in order to predict their behaviour and reveal the underlying preference structures that shape these responses. The authors use binary choice models to infer the airlines' cancellation utility functions by observing their actual flight-cancellation choices. The study finds that larger, fuller, less frequent, shorter-distance, and spoke-bound flights are less likely to be cancelled and that there is inter-airline variation in flight cancellation behaviour. (Xiong & Hansen 2013)

To summarize, a number of studies have been reviewed in the travel industry, particularly in the context of non-refundable ticket cancellations, regarding the optimization of non-refundable fares and the difficulty of striking a balance between customer satisfaction and service provider profitability. A customer risk group segmentation method that recommends CPS fees based on customer risk uses clustering algorithms like K-means, hierarchical agglomerative, DBSCAN, and self-organizing maps (SOM), as well as a value-driven decision forecast model that takes into account personal values, are among the

methods for optimizing cancellation protection service (CPS) fees that have been proposed (ANP). These findings demonstrate that careful attention to both service provider profitability and consumer needs might result in more efficient CPS policies, which will eventually enhance the entire customer experience. The literature analysis also revealed that the ticket refund service has a sizable impact on consumer confidence in the airline business. Other elements that affect consumer trust in the sector may be the subject of future study. Overall, these studies show how crucial it is to strike a balance in the travel sector between client satisfaction and service provider revenue, particularly when it comes to nonrefundable ticket prices. It is feasible to develop more efficient rules that enhance the overall customer experience by carefully taking into account both consumer wants and service provider profits.

Table 3.1 Findings of Review Sources

Author View	Article Title	Utilize Methodology
Sadreddini, Z., Donmez, I., & Yanikomeroglu, H. (2021)	Cancel-for-Any-Reason Insurance Recommendation Using Customer Transaction- Based Clustering	K-means Clustering Agglomerative Clustering DBSCAN SOM Clustering
Sadreddini, Z. (2020)	A Novel Cancellation Protection Service in Online Reservation System	Fixed CPS QoE-Based CPS Flexible CPS
Iliescu, Dan C (2008)	Customer Based Time-to- Event Models for Cancellation Behavior: A Revenue Management Integrated Approach	Mixed Dynamic Programming (MDP)
Chiew, E., Daziano, R. A., & Garrow, L. A. (2017)	Bayesian Estimation of Hazard Models of Airline Passengers' Cancellation Behavior	Discrete-Time Proportionality Ratios (DTPO)
Cirillo, C., Bastin, F., & Hetrakul, P. (2018).	Dynamic Discrete Choice Model for Railway Ticket Cancellation and Exchange Decisions	Dynamic Discrete Choice Models (DDCM)
Maulana, S. A., Gigantama, M. R., Lesmini, L., Ozali, I., & Budiman, C. (2019)	The Influence of Ticket Refund Service Towards Air Asia Customers Trust	Regression Analysis
Ozdemir S. (2015)	Value Driven Decision Forecast Model:An Application on Online Flight Ticket Purchase	Means-End Chain AAS Paired Hotelling's T2 Test
Keleş, M. B., Keleş, A., & Keleş, A. (2020)	Developing Flight Price Prediction Models with Artificial Intelligence Technology	GB-Gradient Boosting RF-Random Forest RF-FPPredictor GB-FPPredictor
Bandyopadhyay, S. K., Goyel, V. & Dutta, S. (2020)	Prediction of Air Flight Cancellation during COVID-19 using Deep Learning Methods	Deep Learning Framework, Long-short term memory (LSTM) and Gated Recurrent Unit (GRU)

Alderighia,M., Gaggeroc, A.A. (2018)	Flight cancellations and airline alliances: Empirical evidence from Europe	Empirical Model
Petraru, O. (2016)	Airline Passenger Cancellations: Modeling, Forecasting and Impacts on Revenue Management	Time Series Modeling, PODS Booking Simulation Tool,
Vázquez Ibáñez, A. R. (2022)	A study of flight cancellation and delays in the UK	ARIMA ARIMAX
Rupp, N. G., & Holmes, G. M. (2006)	An investigation into the determinants of flight cancellations	Empirical Model Monopoly Model
Xiong, J., & Hansen, M. (2013)	Modelling airline flight cancellation decisions	Linear Piecewise Model Random Coefficient Model

4.METHODOLOGY

Methodology as a subject of philosophy and later of science is derived from the words meta, hodos and logos. The main thing in the methodology is that the information collected about a subject can contribute to the development, classification and analysis of an idea. With the development of technology and the rapid development of the digital world, methodological methods have developed. With the development of technology and the integration of the concept of "Industry 5.0" into our lives, the concepts of "Internet of Thinks (IoT)" and decentralized but interconnected machines have begun to come to the fore. The main difference between Industry 4.0 and Industry 5.0 is the increased human-machine interaction that allows people to express themselves in the form of personalized products and services (Aslam, Aimin, Li, & Rehman, 2020). When the Internet of Thinks (IoT) concept is considered as a whole, seen concepts such as machine learning, deep learning, automation and artificial intelligence in its sub-titles. Machine learning (ML), which is regarded as a subset of artificial intelligence (AI), demonstrates the experiential "learning" associated with human intelligence while also having the ability to learn from and refine its findings using computational methods (Helm, Swiergosz, & Haeberle, 2020). Rather than being explicitly programmed to perform a specific task, the machine learning system is trained on large datasets to recognize patterns and make predictions or decisions based on the data it learns.

The goal of machine learning, a subfield of computer science, is to make it possible for computers to "learn" without being explicitly taught (Bi, Goodman, Kaminsky, & Lessler, 2019) and machine learning is to develop algorithms that can learn from data to automatically improve their performance on a given task, without being explicitly programmed. Machine learning algorithms can be used for a wide variety of applications, including image and speech recognition, natural language processing, predictive modeling, and anomaly detection.

The process of a machine learning project usually includes the following steps:

Step 1. Problem Definition: Identify the problem that needs to be solved or the task that needs to be accomplished using machine learning. This step includes understanding the business problem, defining the project goals, and setting success criteria.

Step 2. Data Collection: Data collecting is a significant barrier to machine learning and is a hotly debated topic in many communities. Data collecting has recently become a crucial concern for primarily two reasons. First, as machine learning is used more frequently,

new applications are emerging that may not have enough labeled data. Second, unlike conventional machine learning, deep learning algorithms create features automatically, saving money on feature engineering but maybe require more labeled data (Roh, Heo, & Whang, 2021). So collect and collect relevant data required for the project.

Step 3. Data Preprocessing: This step includes cleaning data, processing missing values, and removing irrelevant features. This is an important step, as the quality of the data is crucial to the performance of the machine learning model. However, as the data continues to grow, the machine learning techniques that are currently in use struggle greatly to handle the unprecedented amount of data. To meet the demands of future data processing, there is a huge need now to create effective and intelligent learning methods. (Qiu, Wu, Ding, Xu, & Feng, 2016)

Step 4. Feature Engineering: The practice of creating numerical fingerprints of interested systems based on domain expertise is known as feature engineering in machine learning (Li, Ma, & Xin, 2017). This step involves creating new features or selecting the most relevant features from the data that are likely to improve the performance of the model.

Step 5. Model Selection: Select the appropriate machine learning algorithm or model for the problem at hand. Algorithm selection will depend on the nature of the problem, data type, and performance requirements.

Step 6. Model Training: Train the machine learning model on preprocessed data. This includes feeding data into the model and adjusting the model's parameters to optimize its performance.

Step 7. Model Evaluation: Evaluate the performance of the model on the test data to determine its accuracy, precision, recall, and F1 score. This step helps identify the strengths and weaknesses of the model and provides insight on how to improve its performance.

Step 8. Model Deployment: Deploy the machine learning model in a production environment to perform the task for which it was designed. This step involves integrating the model into an application or system and ensuring that it works as expected.

Step 9. Model Monitoring and Maintenance: Monitor the model's performance over time and update as needed to ensure it continues to perform well. This step includes retraining the model on new data, making adjustments to the model's parameters, and addressing any issues that arise.

4.1. Data Pre-processing

Data preprocessing plays a critical role in the field of machine learning. It involves a series of steps aimed at transforming raw data into a format suitable for analysis and model training. By carefully preprocessing the data, machine learning models can effectively learn patterns and make accurate predictions. It helps in improving data quality, reducing noise, and addressing issues that could hinder model performance. Data preprocessing serves as the foundation for successful machine learning, enabling the extraction of meaningful insights and driving informed decision-making.

4.1.1. Feature engineering

Feature engineering is a crucial process in machine learning that involves transforming raw data into a format that is suitable for training machine learning models. It is a creative and iterative process that requires deep understanding of the data and the problem at hand. In feature engineering, new features are derived or selected from the existing data to enhance the performance and predictive power of the models. These engineered features capture relevant information, patterns, and relationships in the data, allowing the models to make more accurate predictions. Feature engineering techniques include mathematical transformations, aggregation, scaling, encoding categorical variables, and creating interaction or polynomial terms. Additionally, domain knowledge and expertise play a significant role in identifying and creating meaningful features. The success of machine learning models heavily relies on the quality and relevance of the engineered features, as they directly influence the models' ability to learn and generalize from the data. Therefore, feature engineering is a critical step in the machine learning pipeline, enabling researchers to extract valuable insights and achieve improved performance in various academic domains.

4.1.2. Missing values handling

The tactics and methods used to handle and deal with missing or incomplete data within a dataset are referred to as "missing value handling." Missing values can happen for a number of reasons, including incorrect data entry, equipment failures, or survey participant non-response. Because they can induce biases, impair the accuracy of analytical results, and perhaps result in inaccurate conclusions, missing values must be handled carefully.

The steps of methodology of missing values:

Step 1. Data Collection: The dataset was collected through [describe data collection process]. Due to various factors such as data entry errors, equipment malfunctions, and participant non-response, the dataset contained missing values.

Step 2. Missing Value Identification: Initially, missing values were identified by thoroughly examining the dataset. The missing values were marked as " Not Applicable(N/A)" in the dataset.

Step 3. Missing Value Exploration: Descriptive statistics, such as the proportion of missing values in each variable, were calculated to acquire understanding of the type and pattern of missing values. In order to assess if the missingness was entirely random or had a specific structure, missing value patterns were also investigated.

Step 4. Missing Value Handling: To handle missing values appropriately, multiple strategies were employed:

- **Deletion:** Key variables' missing values cases were eliminated from the dataset. This method was used when there were a lot of missing data points and deleting the cases had little to no affect on how representative the remaining data were.
- **Mean Imputation:** Missing values in variables [list variables] were imputed using the mean of the available values in each respective variable. This method was applied when the missing values were assumed to be missing completely at random (MCAR) or missing at random (MAR). (Baijayanta R., (2019))
- **Regression Imputation:** Using a regression model, missing values in the variables (list variables) were estimated. To predict the missing values, the model was constructed utilizing the other pertinent variables as predictors. This strategy was used when it was considered that the missingness had a systematic relationship with the other variables.
- **Multiple Imputation:** To address the uncertainty associated with missing values, multiple imputations were generated using a Markov Chain Monte Carlo (MCMC) algorithm. Five imputed datasets were created, and subsequent analyses were conducted on each imputed dataset separately.
- **Indicator Variable:** A binary indicator variable was created for each variable with missing values to indicate whether a value was missing or not. This approach allowed for the inclusion of missingness as a separate category in the subsequent analysis.

Step 5. Sensitivity Analysis: A sensitivity analysis was performed to assess the impact of different missing value handling techniques on the results. The analysis involved

comparing the outcomes obtained from the original dataset, as well as the datasets with missing values handled using various strategies.

4.1.3. Outliers handling

Data points known as outliers differ greatly from the rest of the data. They can occur for a variety of causes, including measurement errors, data input problems, or uncommon occurrences. Outliers can significantly affect statistical studies and machine learning models since they can skew the findings and have an impact on how well the model performs.

Identifying and treating outliers in a dataset is referred to as outlier handling. The Interquartile Range (IQR) technique is one of many approaches that may be used to deal with outliers.

The IQR method is a robust statistical technique that uses the concept of quartiles to identify outliers. It is based on the distribution of the data and involves the following steps:

Step 1. Calculate the first quartile (Q_1) and the third quartile (Q_3) of the data.

Step 2. Compute the interquartile range (IQR) by subtracting Q_1 from Q_3: IQR = Q_3 - Q_1.

Step 3. Define the lower bound as $Q_1 - 1.5 * IQR$ and the upper bound as $Q_3 + 1.5 * IQR$.

Step 4. Any data points that fall below the lower bound or above the upper bound are considered outliers.

Once the outliers are identified using the IQR method, there are different approaches to handling them:

- **Removal:** Outliers can be simply removed from the dataset. However, this should be done with caution, as removing too many outliers can lead to information loss and bias in the data. It is generally recommended to remove outliers only if they are due to data entry errors or measurement issues.
- **Imputation:** Instead of removing outliers, they can be replaced or imputed with more reasonable values. This could involve replacing outliers with the mean, median, or a specific value based on the context of the data.
- **Binning:** Outliers can be placed into a separate bin or category to differentiate them from the rest of the data.

- **Transformation:** Applying a transformation to the data, such as a logarithmic transformation, can help reduce the impact of outliers and make the distribution more suitable for analysis.

The selection of an outlier handling method must take into account the particular dataset, the kind of outliers, and the objectives of the analytic or machine learning activity. It is frequently advised to carefully consider and comprehend the causes of outliers before choosing a successful treatment method.

4.1.4. Encoding

Encoding is the process of converting data from one format to another for the purpose of standardization, compactness, security, or enhancement of usability. In the context of data analysis, encoding refers specifically to the process of converting categorical data into numerical format that can be easily analyzed by a computer (Panda & Majhi, 2018).

Categorical data refers to data that can be divided into discrete groups or categories, such as gender, color, or type of product. Machine learning algorithms and statistical analyzes typically require numerical inputs, so encoding is necessary to transform categorical data into a numerical format that can be used for analysis.

There are several different techniques for encoding categorical data, including one-hot encoding, ordinal encoding, and binary encoding. One-hot encoding involves creating a binary variable for each category, and assigning a value of 1 to the corresponding variable for each observation that belongs to that category. Ordinal encoding assigns a numerical value to each category based on its rank or order, while binary encoding creates binary variables for each category, but uses a binary code that is more efficient than one-hot encoding.

In summary, encoding is a critical step in data analysis that involves converting categorical data into a numerical format that can be easily analyzed by a computer. The choice of encoding technique depends on the type of data being analyzed and the specific analysis being performed.

4.2. Modelling

Modeling is a crucial step in the machine learning pipeline that focuses on creating predictive models using preprocessed data. It involves selecting an appropriate algorithm or model architecture that can capture the underlying patterns and relationships within the data. The choice of the model depends on the specific problem at hand, such as classification, regression, clustering, or recommendation. During data modeling, the model is trained on the preprocessed data, optimizing its parameters to minimize the error or maximize the accuracy of predictions. The ultimate goal of data modeling is to create a reliable and effective model that can accurately predict outcomes or classify new, unseen data instances.

4.2.1. Classification methods

Classification is a type of machine learning problem where the goal is to predict the class or category of a new observation based on its properties or attributes. In a classification problem, data is labeled with a categorical or discrete outcome variable, and the goal is to learn a model that can accurately predict the class of new, invisible data points.

There are several techniques used to solve classification problems, including:

- Logistic regression: In epidemiology and medicine, logistic regression is the modeling technique most frequently employed for binary outcomes. The model explicitly depicts the relationship between the explanatory variable X and the response variable Y. It belongs to the family of generalized linear models (Levy & O'Malley, 2020).

$$logit\ (p) = log\ log\ (\frac{p}{p-1}) = \beta^{\rightarrow}.\ x^{\rightarrow} = \beta_0 + \sum_{i=1}^{N} \beta_i X_i \qquad (4.1)$$

A binary outcome Y is supposed to follow a binomial distribution conditional on the predictors, where p = P(Y = 1 | X) denotes the likelihood of the binary response given the predictors

The method above presupposes a linear relationship between the predictors and the logarithm of the probability of the result, as equivalently shown below:

$$E\ [Y|X] = P(\vec{x}) = \frac{1}{1 + e^{-\vec{\beta}.\vec{x}}} \qquad (4.2)$$

- Decision Trees: A tree-like model used to make decisions based on a set of conditions.

- Random Forest: By taking into account the creation of numerous Classification and Regression Trees (CART) models to make a forecast, the random forest method extends the basic Classification and Regression Trees (CART) algorithm. Each Classification and Regression Trees (CART) model, when bootstrapping the training samples, considers the best splitting feature out of n < N randomly selected features at a time, then fits following branches by selecting out of another random set of n features. Decision trees, on the other hand, are constructed by utilizing all of the features. Predictions are created by combining the output of a number of these estimators (Levy & O'Malley, 2020).
- Support vector machines (SVM): A technique used to find the best boundary between different classes of data.
- Neural networks: A set of algorithms used to model complex relationships between inputs and outputs.

The choice of technique will depend on the nature of the problem, the size and complexity of the data, and performance requirements.

4.2.1.1. Decision trees

The non-parametric supervised learning approach used for classification and regression applications is the decision tree. It is organized hierarchically and has a root node, branches, internal nodes, and leaf nodes.

By using a greedy search to find the ideal split points inside a tree, decision tree learning uses a divide and conquer technique. When most or all of the records have been classified under distinct class labels, this splitting procedure is then repeated in a top-down, recursive fashion. The intricacy of the decision tree plays a significant role in determining whether or not all data points are categorized as homogenous sets. Pure leaf nodes, or data points belonging to a single class, are easier to obtain in smaller trees. It gets harder to preserve this purity as a tree gets bigger, which typically leads to too little data falling under a particular subtree. This is known as data fragmentation, and it frequently results in overfitting.

The use of the decision tree regressor helps to subdivide N data according to the feature space f_i into τ branches (B_c) ending in l leaf nodes per tree (Hoyle, Rau, Zitlau, Seitz, & Weller, 2015):

$$MSE = \frac{1}{N} \sum_{c=1}^{N} \sum_{f_i \in B_r} Z_{spec,i} - < Z_{spec,r} >)^2 \quad (4.3)$$

The branches are designed with a low mean squared error for the spectroscopic redshift $Z_{spec,i}$ in each leaf. Regarding the average spectroscopic redshift:

$$< Z_{spec,r} > = \frac{1}{N_c} \sum_{f_i \in B_r} Z_{spec,i} \quad (4.4)$$

Where N_c is the number of objects in each leaf. The branches of the tree correspond to regions of input feature space.

4.2.1.2. Random forest

Leo Breiman and Adele Cutler are the creators of the widely used machine learning technique known as random forest, which mixes the output of various decision trees to produce a single outcome. Its widespread use is motivated by its adaptability and usability because it can solve classification and regression issues.

The bagging method is extended by the random forest algorithm, which uses feature randomness in addition to bagging to produce an uncorrelated forest of decision trees. The random subspace method, also known as feature bagging, creates a random subset of features that guarantees low correlation between decision trees. The main distinction between decision trees and random forests is this. Random forests merely choose a portion of those feature splits, whereas decision trees take into account all possible feature splits.

There are three key hyperparameters for random forest algorithms that must be set prior to training. Node size, tree count, and sampled feature count are a few of them. From there, classification or regression issues can be resolved using the random forest classifier.

Each decision tree in the ensemble that makes up the random forest method is built of a data sample taken from a training set with replacement known as the bootstrap sample. One-third of the training sample—also referred to as the out-of-bag (oob) sample—is set aside as test data; will return to this sample later. The dataset is subsequently given a second randomization injection by feature bagging, increasing dataset diversity and decreasing decision tree correlation. The prediction will be determined differently depending on the type of issue. The individual decision trees will be averaged for a classification task and for

a regression task, respectively. The anticipated class will be produced by the most prevalent category variable. The prediction is then finalized by cross-validation using the oob sample.

Tree-structured classifiers make up a random forest, which is a type of classifiers $\{h(x, \Theta_k), k = 1, ...\}$. If each tree casts one unit vote for the most popular class at input $\{\Theta_k\}$, and the x are independent, identically distributed random vectors. (Zhang, Wang, & Liu, 2012)

This definition demonstrates that RF is a fusion of several classifiers with tree-structures. Every tree is planted using a training sample set and a random variable in Breiman's RF model; the random variable corresponding to the kth tree is designated as Θ_k.

When any two of these random variables are independently distributed and have the same distribution, a classifier $h(x, \Theta_k)$ is produced, where x is the input vector. After k runs, the classifier array $\{h_1(x), h_2(x), ..., h_k(x)\}$ is retrieved and used to construct multiple classification model systems. The decision function for the final outcome of this system is the ordinary majority vote, as follows:

$$H(x) = arg\ arg\ max \sum_{i=1}^{k} I(h_i(x) = Y) \tag{4.5}$$

The Y is the output variable, I(.) is the indicator function, h_i is a single decision tree model, and $H(x)$ is a model combination for classification.

4.2.1.3. XGBoost algorithm

Extreme Gradient Boosting is also known as XGBoost (Chen & Guestrin, 2016). It is an example of the Gradient Boosting Machine (GBM) technique, which is mostly used to build regression and classification predictive modeling issues. The majority of current GBM models have consistently outperformed other machine learning methods, i.e. they have done better on a variety of machine learning tasks. Reference data sets for learning (Friedman, 2001).

For a given dataset, the following is assumed to better illustrate the Boost algorithm:

$$\hat{y} = \sum_{k=1}^{K} f_k(x_i), f_k \in K \tag{4.6}$$

$D = (x_i, y_i): i = 1, ..., n, x_i \in R^m, y_i \in R$ n observations with m features let be defined as the predict value by the model;

The eXtreme Gradient Boosting (XGBoost) was first introduced by Chen and Guestrin (Chen & Guestrin, 2016). The residual will be adjusted at each gradient boosting iteration in order to correct the prior predictor and enable the optimization of the given loss function:

$$Obj = \sum_{i=1}^{n} l(\hat{y}, y_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{4.7}$$

The value of the forecast made by the k^{th}tree for the i^{th} sample is represented by the function $f_k(x_i)$. By minimizing the goal function, one can learn the set of functions f_k. With:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\omega^2 \tag{4.8}$$

The classification tree's first term, I, is specified by:

$$l(\hat{y}, y_i) = y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i) \ln(1 + e^{\hat{y}_i}) \tag{4.9}$$

The difference between the predicted value $\hat{y}$ and the goal value y_i, which it represents as the loss function, is measured. The regularization term, which is a metric used to assess the complexity of the tree f_k, is represented by the second term Ω. Where γ and λ represent the regularization gradations. T and ω represent, respectively, the quantity of leaves and the vector of values assigned to each leaf. Then, using optimization, can be determined which minimizes the objective function. By extending the aforementioned function to the second order via the Taylor expansion, various loss functions can be easily accommodated:

$$Obj^{(t)} \cong \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_i(x_i) + \frac{1}{2} h_i f_t(x_i)^2] + \Omega(f_t) \tag{4.10}$$

Where, on the loss function, and;

$$h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \tag{4.11}$$

Ordering gradient statistics. Achieving the following approximation by expanding Ω as follows after deleting the constant term:

$$Obj^{(t)} = \sum_{i=1}^{n} [g_i f_i(x_i) + \frac{1}{2} h_i f_t(x_i)^2] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega^2 = \sum_{j=1}^{T} [(\sum_{i \in l_j} g_i)\omega_j + \frac{1}{2}(\sum_{i \in l_j} h_i + \lambda)\omega_j^2] + \gamma T \tag{4.12}$$

Where $l_j = \{i|q(x_i) = j|\}$ stands for the leaf j instance set. The ideal w^*_jweight of leaf j and the related ideal value can be computed for a fixed tree structure q by:

$$w_j^* = -\frac{G_j^2}{H_j + \lambda} \tag{4.13}$$

And there exist the following when w_j^* is substituted into equation (4.13):

$$Obj^* = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \lambda T \tag{4.14}$$

It is practically impossible to list every tree structure that might exist. Instead, a greedy approach that builds the tree iteratively from a single leaf is applied. The following function can determine whether a split should be added to the current tree structure:

$$G = \frac{1}{2}\left[\frac{(\sum_{i\in l_L} g_i)^2}{\sum_{i\in l_L} h_i + \lambda} + \frac{(\sum_{i\in l_R} g_i)^2}{\sum_{i\in l_R} h_i + \lambda} + \frac{(\sum_{i\in l_l} g_i)^2}{\sum_{i\in l_l} h_i + \lambda}\right] - \gamma \tag{4.15}$$

Assume that l_R and l_L represent the instance sets of the split left and right nodes, respectively.

4.2.1.4. Gradient boosting machine (GBM)

A machine learning approach called Gradient Boosting Machine (GBM) creates predictive models by sequentially integrating a group of weak prediction models. It is a member of the boosting algorithm family, which aims to enhance model performance by the iterative training of new models that concentrate on the flaws in the prior models.

Given a training dataset (x_1, y_1), (x_2, y_2),….(x_n, y_n) where x_i represent the i^{th} training instance and y_i is the corresponding label, and a differentiable loss function L(Y, F(x)) that measures the discrepancy between the predicted values F(x) and the true labels y, the GBM algorithm can be summarized in the following steps:

Step 1. Initialize the Ensemble Model: Set the initial prediction as the average of the training labels:

$$\hat{F}_0(x) = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{4.16}$$

Step 2. For m = 1 to M, Where M is the Number of Boosting Iterations:

a) Compute the negative gradient of the loss function with respect to the current ensemble predictions for each training instance:

$$r_{im} = -\left[\frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}\right], for\ i = 1,2, \dots, n \quad (4.17)$$

b) Fit a weak base learner (e.g., decision tree) to the negative gradients r_{im} as the targets and the training instances x_i as the input features. This base learner produces a new model $h_m(x)$.

c) Determine the optimal step size or learning rate v to update the ensemble predictions:

$$v_m = \arg min_v \sum_{i=1}^{n} L(\, x_i, F_{m-1}(x_i) + v.\, h_m(x_i)) \quad (4.18)$$

d) Update the ensemble predictions by adding the weighted contribution of the new model:

$$F_m(x) = F_{m-1}(x) + v_m.\, h_m(x) \quad (4.19)$$

e) Output the final ensemble model $F_M(x)$.

The GBM algorithm iteratively trains weak models to predict the negative gradients of the ensemble's current predictions. It optimizes the ensemble's predictions by minimizing the loss function using gradient descent-like updates. Each new model focuses on the mistakes made by the previous models, gradually improving the overall predictive performance of the ensemble.

4.2.1.5. Light gradient boosting machine algorithms (LightGBM Algorithms)

A well-known machine learning technique that is a member of the gradient boosting method family is called Light Gradient Boosting Machine (LightGBM). It has received a lot of interest in research and business because to its performance and ability to quickly handle enormous datasets.

Some key features and characteristics of LightGBM Algorithm:

- **Gradient Boosting:** The foundation of LightGBM is the gradient boosting framework, an ensemble learning method that combines a number of weak models (usually decision trees) to produce a powerful prediction model. It improves performance by repeatedly adding new trees to the model that concentrate on the data that were incorrectly categorized.
- **LightGBM Architecture:** LightGBM employs a leaf-wise tree growth strategy, as opposed to the level-wise strategy used by many other gradient boosting algorithms. In

the leaf-wise strategy, trees are grown by splitting the leaf that provides the maximum reduction in the loss function. This approach can lead to faster training times and better accuracy, especially for datasets with a large number of features.

- **Gradient-Based Optimization:** LightGBM uses a gradient-based optimization method known as Gradient-based One-Side Sampling (GOSS) to reduce the training time. GOSS selects a subset of instances based on the gradient information, prioritizing the instances with large gradients while randomly sampling the instances with small gradients. This technique helps in achieving a good trade-off between computational efficiency and accuracy.
- **Handling Categorical Features:** Because LightGBM has native support for categorical features, pre-processing processes like one-hot encoding are not necessary. By determining the ideal split for categorical feature groups during tree growth, it can directly handle categorical variables.
- **Regularization:** L1 and L2 regularization (to control model complexity) and feature sub-sampling (to reduce overfitting) are two regularization approaches offered by LightGBM. These regularization techniques aid in preventing overfitting of the model to the training set and improve generalization to new sets of data.
- **Parallel and Distributed Computing:** LightGBM can benefit from distributed and parallel computing to hasten the learning process. In order to handle huge datasets effectively, it allows parallel training on a single system using several CPU cores and may also be dispersed over numerous machines.

Numerous machine learning tasks, such as classification, regression, ranking, and anomaly detection, have been successfully completed using LightGBM. Due to its effective implementation, scalability, and competitive performance, it has grown in popularity and is now a useful tool in many real-world applications.

4.2.2. Performance metrics

Performance metrics are quantitative measures that evaluate how well a machine learning model or algorithm achieves its intended goals. They are used in machine learning for various purposes, such as:

- Comparing different models or algorithms on the same task and data set
- Tuning the hyperparameters of a model or algorithm to optimize its performance
- Assessing the generalization ability of a model or algorithm on unseen data

- Monitoring the performance of a model or algorithm over time and detecting anomalies or errors

Performance metrics are essential for machine learning because they provide objective and reliable feedback on the strengths and weaknesses of a model or algorithm, and guide the improvement and refinement of the machine learning process. Performance metrics are a part of every machine learning pipeline. Here are the performance metrics used in this project:

- **Accuracy**: Accuracy score is a metric that measures how well a classifier can correctly predict the labels of new data. It is calculated by dividing the number of correct predictions by the total number of predictions. Accuracy score can range from 0 to 1, where 0 means no correct predictions and 1 means all correct predictions.
- **Precision score**: Precision is the ratio of correctly predicted positive instances to the total number of predicted positive instances. Precision = (True Positives) / (True Positives + False Positives).
- **Recall score**: Recall is the ratio of correctly predicted positive instances to the total number of actual positive instances. Recall = (True Positives) / (True Positives + False Negatives).
- **F1 score**: This is a harmonic mean of precision and recall, two other important metrics for classification problems. F1 score combines both precision and recall and gives a single number that represents the quality of a classifier. F1 score = 2 * (Precision * Recall) / (Precision + Recall).
- **ROC AUC score**: This is a metric that measures how well a classifier can distinguish between positive and negative classes. ROC stands for Receiver Operating Characteristic, which is a curve that plots the true positive rate (recall) against the false positive rate (1 - specificity) at different thresholds. AUC stands for Area Under the Curve, which is the area under the ROC curve. A higher AUC score means that the classifier can better separate the positive and negative classes.

There are several metrics for evaluation of classification problems (Hossin & Sulaiman, 2015). Such as;

Table 4.1 Threshold Metrics for Classification Evaluations

Metrics	**Formula**	**Evaluation Forces**
Accuracy (acc)	$\frac{tp+tn}{tp+fp+tn+fn}$ (4.20)	In general, the accuracy metric measures the ratio of correct predictions over the total number of instances evaluated.
Error Rate (err)	$\frac{fp+fn}{tp+fp+tn+fn}$ (4.21)	Misclassification error measures the ratio of incorrect predictions over the total number of instances evaluated.
Sensitivity (sn)	$\frac{tp}{tp+fn}$ (4.22)	This metric is used to measure the fraction of positive patterns that are correctly classified.
Specificity (sp)	$\frac{tn}{tn+fp}$ (4.23)	This metric is used to measure the fraction of negative patterns that are correctly classified.
Precision(p)	$\frac{tp}{tp+fp}$ (4.24)	Precision is used to measure the positive patterns that are correctly predicted from the total predicted patterns in a positive class
Recall(r)	$\frac{tp}{tp+tn}$ (4.25)	Recall is used to measure the fraction of positive patterns that are correctly classified
RF-Measure (FM)	$\frac{2*p*r}{p+r}$ (4.26)	This metric is used to maximize the tp rate and tn simultaneously keeping both rates relatively balanced
Geometric-Mean (GM)	$\sqrt{tp*tn}$ (4.27)	This metric is used to maximize the tp rate and tn simultaneously keeping both rates relatively balanced
Averaged Accuracy	$\frac{\sum_{i=1}^{l}\frac{tp_i+tn_i}{tp_i+fn_i+fp_i}}{l}$ (4.28)	The average effectiveness of all classes
Averaged Error Rate	$\frac{\sum_{i=1}^{l}\frac{fp_i+fn_i}{tp_i+fn_i+fp_i}}{l}$ (4.29)	The average arrow rate of all classes
Averaged Precision	$\frac{\sum_{i=1}^{l}\frac{tp_i}{tp_i+fp_i}}{l}$ (4.30)	The average of per-class precision

Note: i -each class of data; tp_i – true positive for C_i; fp_i - false positive for C_i; fn_i - false negative for C_i; tn_i - true negative for C_i; and M macro averaging.

Traditional algorithms of Machine Learning are considered to have low classifier accuracy. In this context, many researchers are working on algorithms that combine classifiers to increase the percentage of accuracy. At this point, it was aimed to use algorithms with high accuracy that can be used in the study. Along with the reviewed articles, it was predicted that it would be more appropriate to use the following algorithms in machine learning applications.

4.2.3. Cross validation

Cross-validation is a resampling technique used in machine learning and statistical modeling to assess the performance and generalization ability of a model. It involves partitioning the available data into multiple subsets, or folds, to train and evaluate the model on different combinations of these subsets. The algorithm for cross-validation can be described as follows:

Step 1. Splitting the Data:

- The original dataset is divided into K approximately equal-sized subsets or folds. Common choices for K are 5 or 10, but it can vary depending on the dataset size and computational resources.
- Each fold is typically stratified to maintain the class distribution or representativeness of the data.
- The model will be trained and evaluated K times, each time using a different combination of folds as the training and testing sets.

Step 2. Training and Testing:

- For each iteration k from 1 to K:
- The k-th fold is held out as the testing set.
- The remaining K−1 folds are used as the training set.
- The model is trained on the training set, using the chosen algorithm and any hyperparameters.
- The trained model is then evaluated on the testing set, and a performance metric (such as accuracy, precision, recall, or mean squared error) is computed.

Step 3. Performance Evaluation:

- After completing the K iterations, the performance metric values obtained from each fold are averaged to provide an overall estimate of the model's performance.
- Additional metrics, such as standard deviation or confidence intervals, may also be calculated to assess the variability or uncertainty of the performance estimate.

Cross-validation helps to overcome issues such as overfitting or bias that can occur when training and evaluating a model on the same dataset. It provides a more robust and reliable estimate of a model's performance by evaluating it on multiple different subsets of the data. Cross-validation is widely used in model selection, hyperparameter tuning, and comparing different algorithms to ensure fair and accurate assessments of model performance.

4.2.4. Stratified cross-validation

Stratified cross-validation is a variant of cross-validation that ensures the class distribution or target variable distribution is maintained across the folds. It is particularly useful when dealing with imbalanced datasets where the classes or target variable categories are unevenly represented. Stratified cross-validation helps to ensure that each fold contains a representative sample of each class or category, thus providing a more reliable and unbiased evaluation of the model's performance.

4.2.5. Hyperparamater optimization

Hyperparameter optimization, often referred to as hyperparameter tuning, is the process of "searching for the best combination of hyperparameter values that maximize the performance of a machine learning model" (Bergstra and Bengio, 2012). Hyperparameters are configuration settings that are set by the user prior to training the model, such as learning rate, batch size, or regularization strength. The objective of hyperparameter optimization is to "find the optimal hyperparameter values that result in the best model performance, typically measured by an evaluation metric such as accuracy or mean squared error" (Snoek, Larochelle, and Adams, 2012). By systematically exploring different combinations of hyperparameters and evaluating their impact on the model's performance, hyperparameter optimization helps to "fine-tune the model, improve its generalization capabilities, and achieve better results on unseen data" (Li and Malik, 2017). It is an important step in the machine learning workflow to ensure the model's optimal performance. Mathematical Model for that optimization is:

Let's consider a machine learning model with n hyperparameters, denoted as $\theta_1, \theta_2, \ldots, \theta_n$. The goal is to find the optimal values for these hyperparameters, denoted as $\theta^*_1, \theta^*_2, \ldots, \theta^*_n$. That maximize the performance metric M of the model.

The mathematical model for hyperparameter optimization can be represented as:

$$\theta^*_1, \theta^*_2, \ldots, \theta^*_n = argmax_{\theta_1,\theta_2,\ldots,\theta_n} M(\theta_1, \theta_2, \ldots, \theta_n) \quad (4.31)$$

Here, $M(\theta_1, \theta_2, \ldots, \theta_n)$ represents the performance metric that quantifies the qualify of the model wih the given hyperparameter values. The goal is to find the values of $\theta_1, \theta_2, \ldots, \theta_n$ that maximize this performance metric.

The optimization problem can be solved using various techniques, such as grid search, random search, Bayesian optimization, or evolutionary algorithms. These techniques explore the space of hyperparameter values and evaluate the model's performance for different combinations, aiming to find the combination that maximizes the performance metric.

It's important to note that the mathematical model presented here is a general representation of hyperparameter optimization. The specific implementation and algorithm used for optimization may vary depending on the chosen technique.

"GridSearchCV is a 'hyperparameter optimization technique' that exhaustively searches through a predefined set of hyperparameter values to find the best combination for a machine learning model. It is implemented as part of the scikit-learn library in Python and automates the process of 'systematically exploring the hyperparameter space' (Pedregosa et al., 2011). GridSearchCV performs a 'brute-force search' by evaluating the model's performance for every possible combination of hyperparameter values specified in a grid or user-defined parameter space. It then selects the hyperparameter values that yield the highest performance according to a specified scoring metric. GridSearchCV provides an efficient way to tune hyperparameters and 'optimize model performance' (Müller and Guido, 2017). It is widely used for fine-tuning machine learning models and finding the best hyperparameter values to improve model accuracy and generalization."

4.2.6. Feature importance

Feature importance can be used to guide model selection, feature selection, and optimization, as well as to provide insights into the underlying patterns and relationships in

the data. As such, it is a key concept in machine learning and data analysis, and has many practical applications in fields like healthcare, finance, marketing, and more.

According to Sebastian Raschka and Vahid Mirjalili in their book "Python Machine Learning", The term "feature importance" refers to methods that rate input features according to how well they can predict a target variable (Raschka & Mirjalili, 2019). This highlights the importance of understanding the relevance of input features in a given context, and how they contribute to the overall prediction or analysis.

When talking about feature significance with identification, these two concepts are essentially combined to explain the process of attributing importance to features based on some external information or criteria, in addition to the analysis of the data itself.This can help us better understand the factors that are most relevant in a given context, and can be particularly useful in fields like healthcare, finance, or social sciences where external factors may play an important role in the analysis.

5. APPLICATION AND RESULTS

In this section, the operations will be explained together with the processes and the results obtained through the outputs will be explained.

5.1. Application

In this section, the technical implementation and application steps of the study are elaborated in detail. The application steps of previous studies similar to this work have been taken as examples, utilizing the information obtained from the literature review. Various tables and visuals have been employed to facilitate a clearer understanding of the implemented applications.

5.1.1. Exploratory data analysis

Exploratory data analysis (EDA) holds paramount importance in the realm of data science and research due to its ability to uncover valuable insights and inform subsequent analyses. As an integral part of the data exploration process, EDA involves the comprehensive examination and visualization of data sets to gain a deeper understanding of their underlying characteristics. Through EDA, researchers can identify patterns, trends, and anomalies within the data, enabling them to formulate hypotheses, generate research questions, and make informed decisions regarding subsequent analyses. Furthermore, EDA facilitates the detection of data quality issues, missing values, and outliers, ensuring the integrity and reliability of the ensuing analysis. The information gleaned from EDA aids researchers in selecting appropriate statistical techniques, determining variable transformations, and identifying potential limitations or biases within the data.

The dataset used in the study includes information about domestic and international flight ticket transactions of Turna.com, an online travel agency, between 2022-06-01/2023-07-01. When the dataset is examined, there are 11256 observations and 152 variables.

The main objective aimed in this study is to establish a machine learning model that predicts passengers' ticket cancellation behavior after purchasing tickets. In this regard, the "target" value has been determined as whether the passenger cancels the ticket or not, and this variable has been used in the subsequent development of the machine learning model. The "target" variable is represented as binary, where tickets with no cancellation request are denoted as "0" and tickets with a cancellation request are denoted as "1". As a result of this observation and analysis, it can be observed that 10406 airway passengers did not request

cancellation after ticketing, while 850 airway passengers requested cancellation after ticketing. The pie chart represents the percentage and distribiton of the target variable can be seen in the below Figure 5.1.

Figure 5.1 shows the distribution of the target variable in the form of a pie chart.

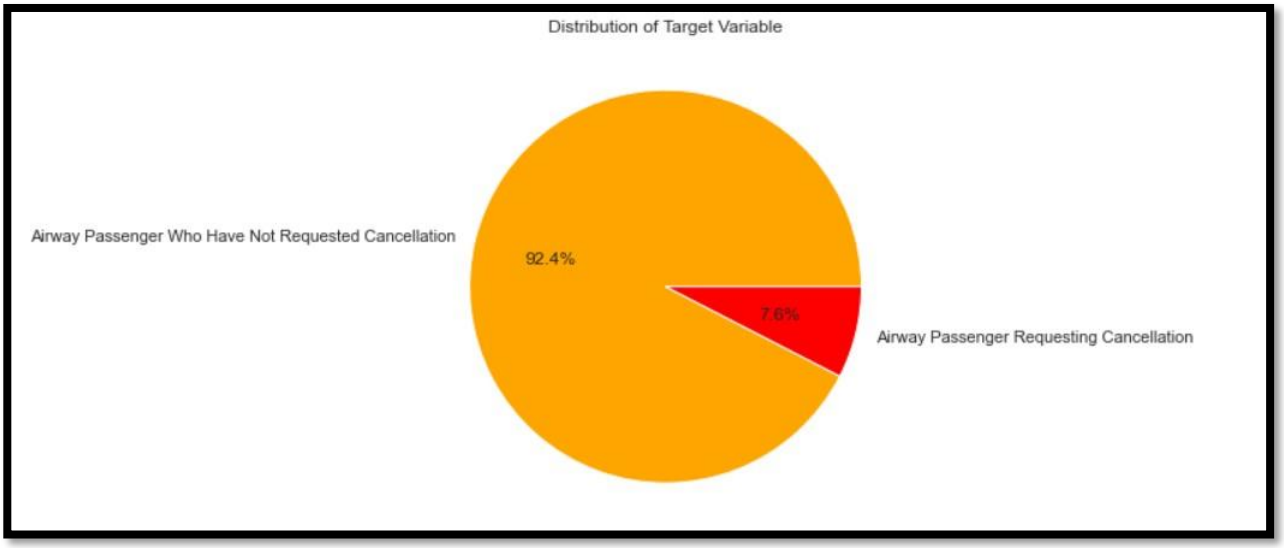

Figure 5.1 Pie Chart of Target Variables Distribution

When the variable structure of the data is examined, there are 84 categorical variables, 43 numerical variables and 25 cardinal variables. Figure 5.2 shows the countplot of the distribution of the variables.

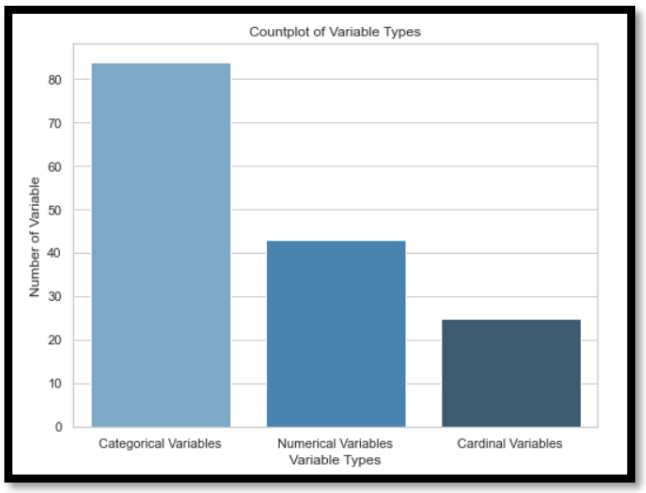

Figure 5.2 Distribution of Variable Types of Data

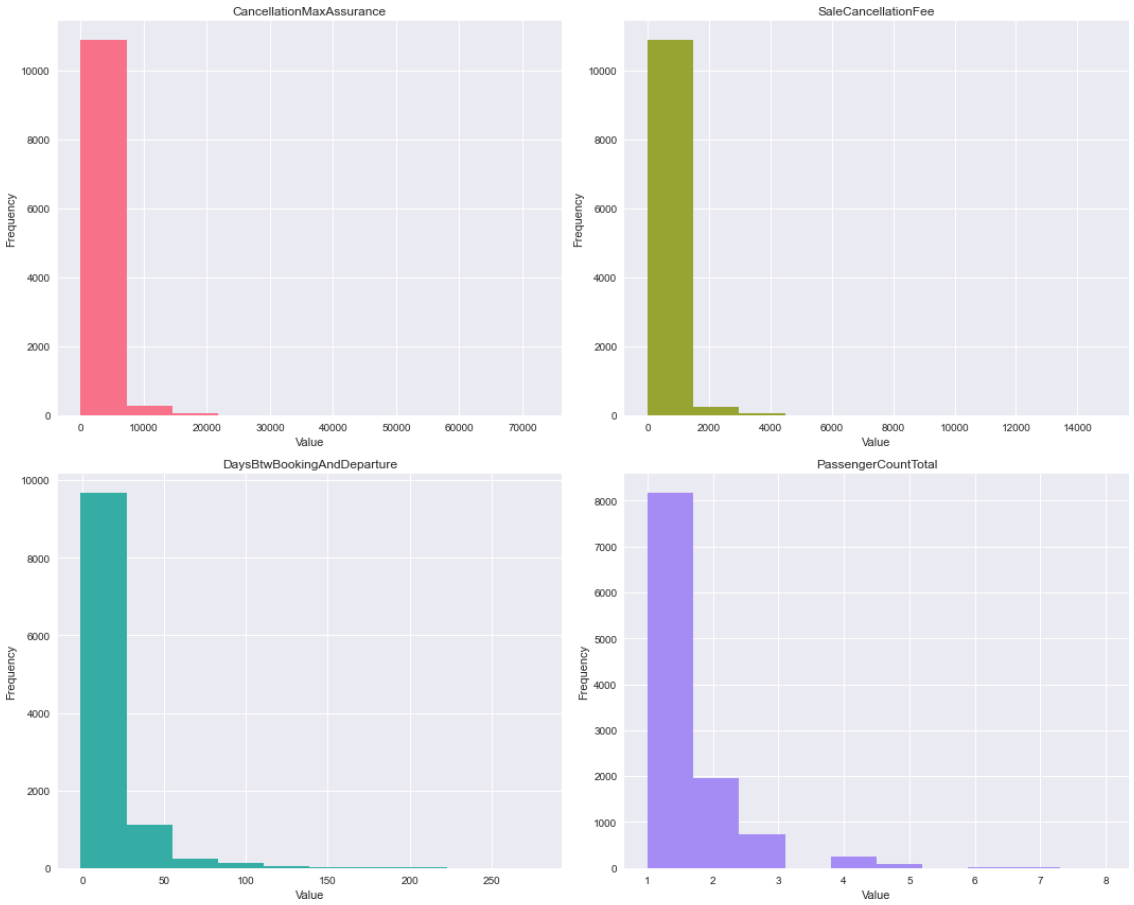

Figure 5.3 Histogram Plots of CancellationMaxAssurance, SaleCancellationFee, DaysBtwBookingAndDeparture and PassengerCountTotal

When the distribution charts of the columns are examined, it can be observed that most of the numeric data have a left-skewed distribution. From the Figure 5.3 and Figure 5.4 it can be observed that CancellationMaxAssurance, SaleCancellationFee, DaysBtwBookingAndDeparture, PassengerCountTotal and SaleTotalFare variables are not normally distributed.

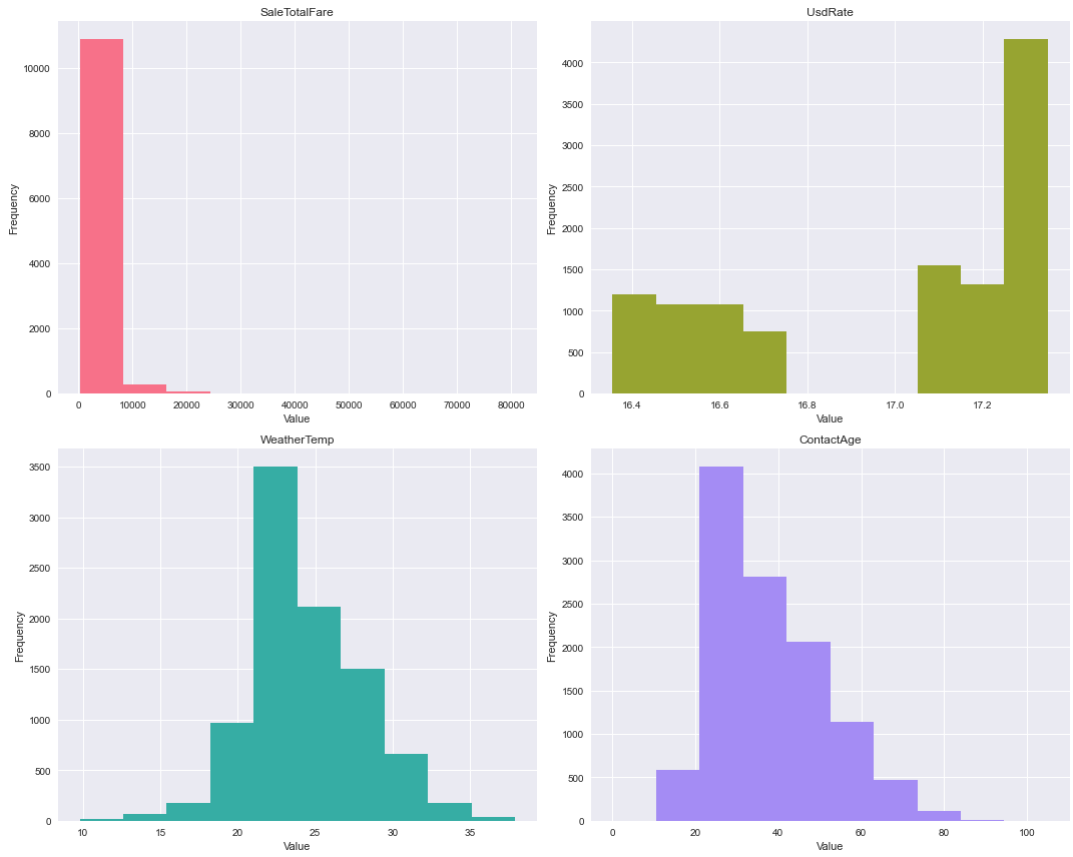

Figure 5.4 Histogram Plots of SaleTotalFare, UsdRate, WheatherTemp, ContactAge

Moreover, WeatherTemp are normally distributed, ContactAge closely normally distributed but for UsdRate same conclusion cannot be made.

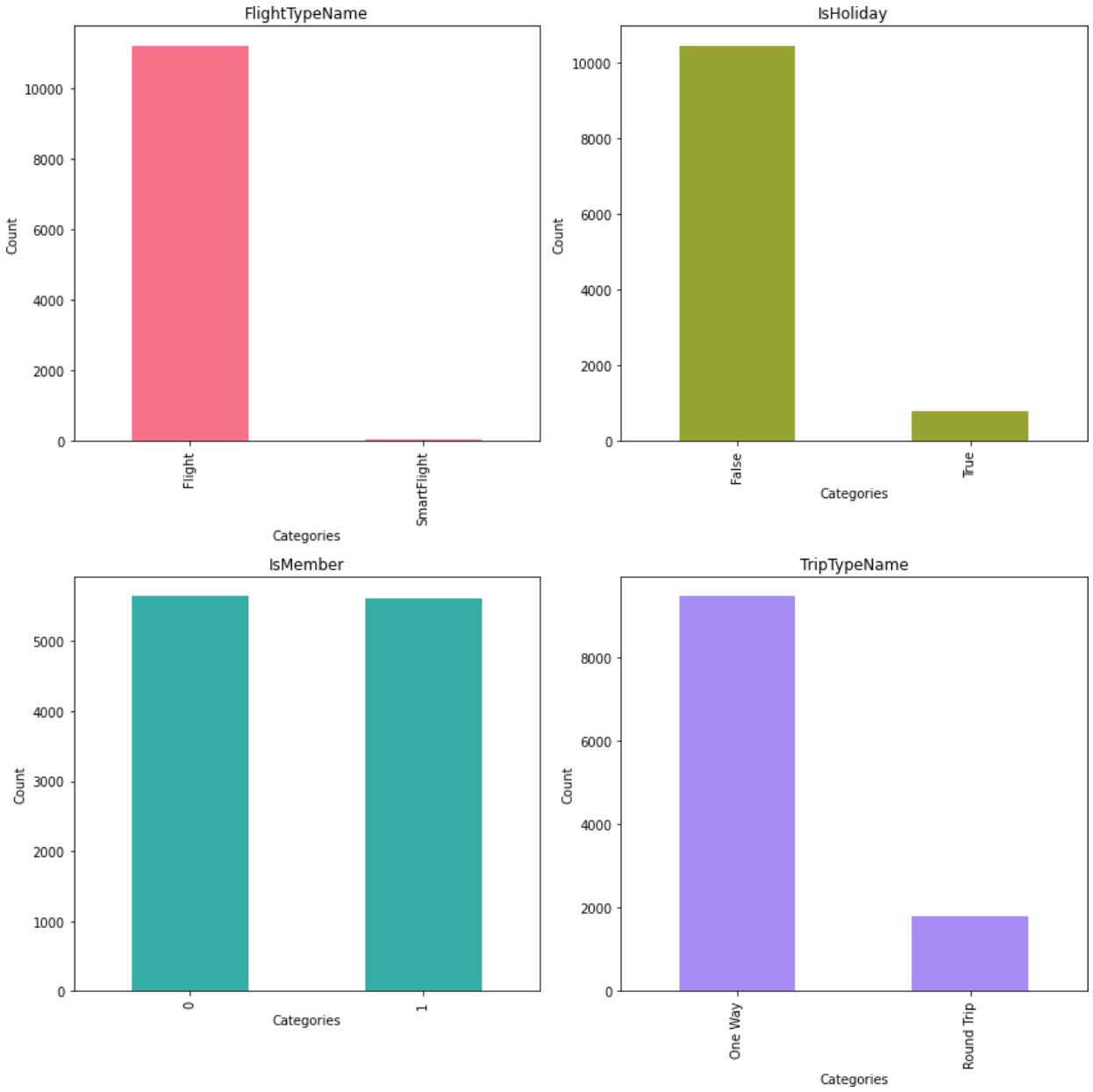

Figure 5.5 Histogram Plots of FlightTypeName, IsHoliday, IsMember, TripTypeName

For the further analysis categorical variables are examined. From the bar plots of categorical variables on the Figure 5.5 above valuable insights can be gained. For FlightTypeName variable the Flight category is highly dominant on SmartFlight category. When looking at the isHoliday graph, it appears that tickets are predominantly purchased on weekdays. Furthermore, it can be observed that half of the total ticket sales are made by members, while the other half is made by non-members. When examining the proportion of one-way and round-trip tickets sold, it is evident that the majority of them are one-way tickets.

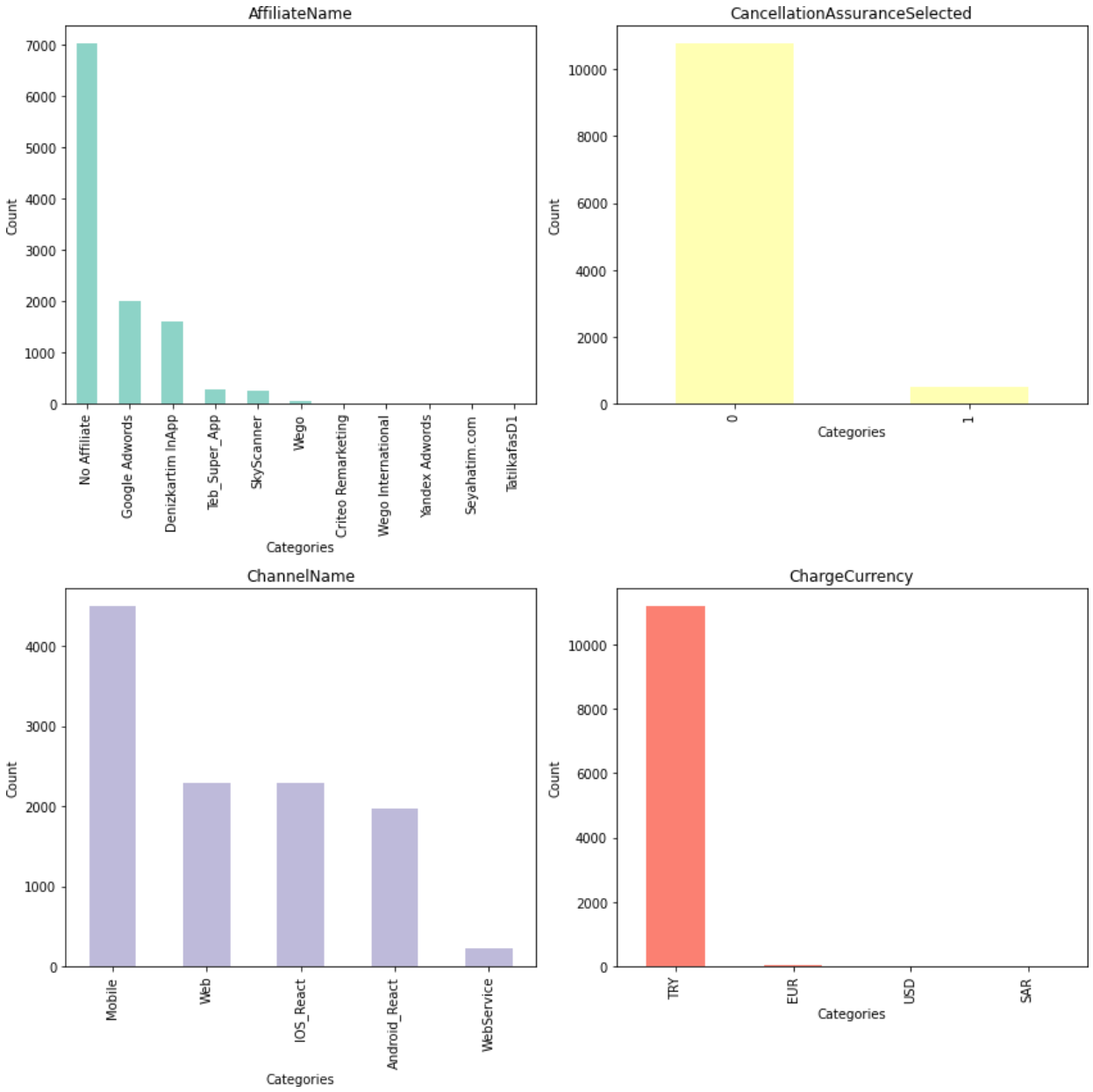

Figure 5.6 Bar Plots for AffiliateName, CancellationAssuranceSelected, ChannelName and ChargeCurrency

When examining Figure 5.6, it has been determined that the most frequently used affiliate is Google AdWords. When comparing those who opted for ticket cancellation services, it can be seen that the majority of passengers did not choose the ticket cancellation service. When examining the ChannelName column, it is observed that the majority of ticket bookings are made through mobile platforms. Additionally, it is understood that the currency used predominantly is TRY (Turkish Lira).

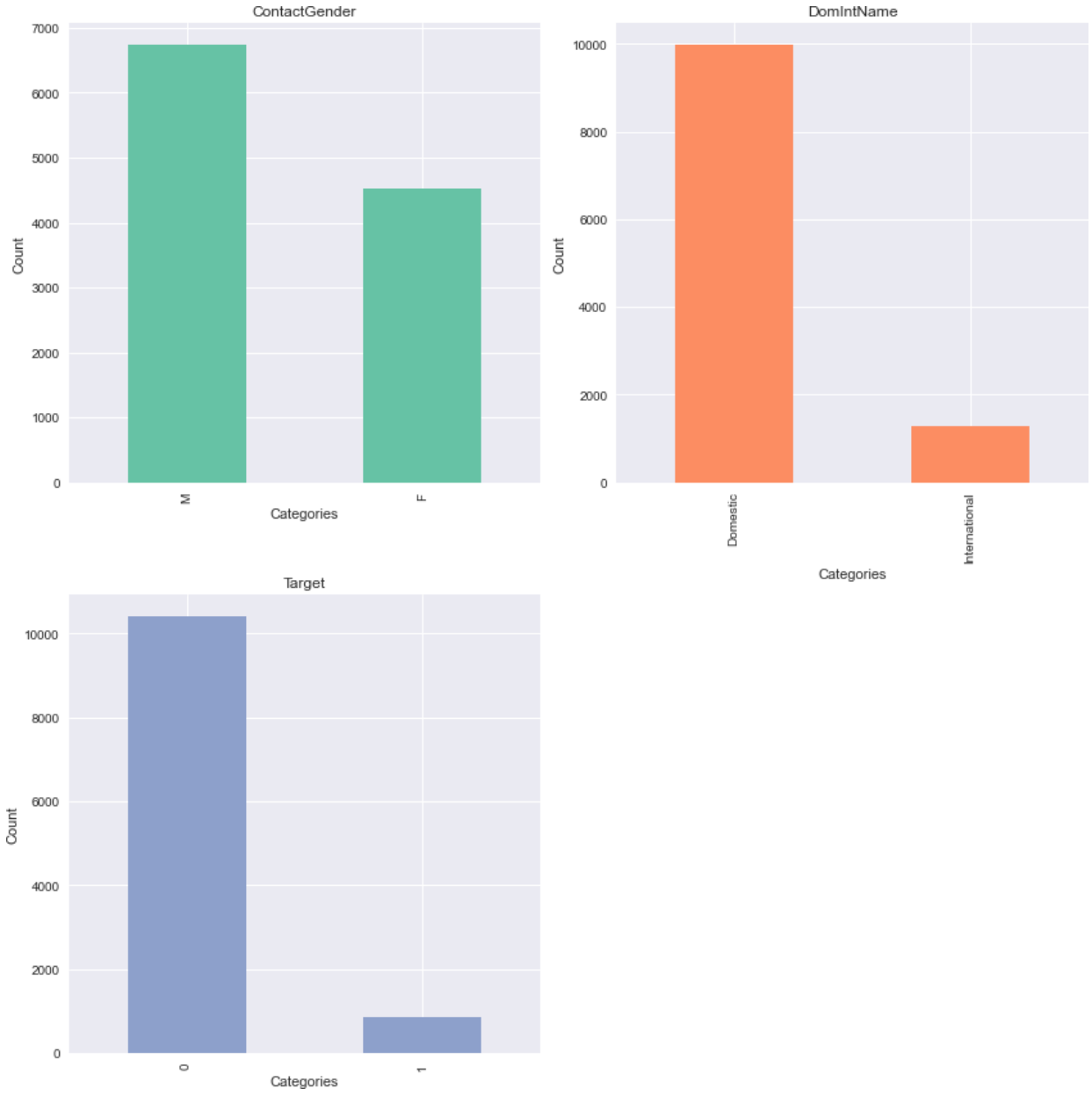

Figure 5.7 Bar Plots for ContactGender, DomIntName and Target

In further analysis, in the Figure 5.7, it is observed that the individuals predominantly provided as emergency contact information by passengers who made bookings are male. When examining the density of domestic and international flights, it is seen that there are significantly more domestic ticket sales. Lastly, when investigating whether passengers who have made flight bookings, which is the target variable, have canceled their plane tickets or not, it is determined that the number of passengers who did not cancel exceeds 10,000, while the number of passengers who canceled is approximately close to 1,000.

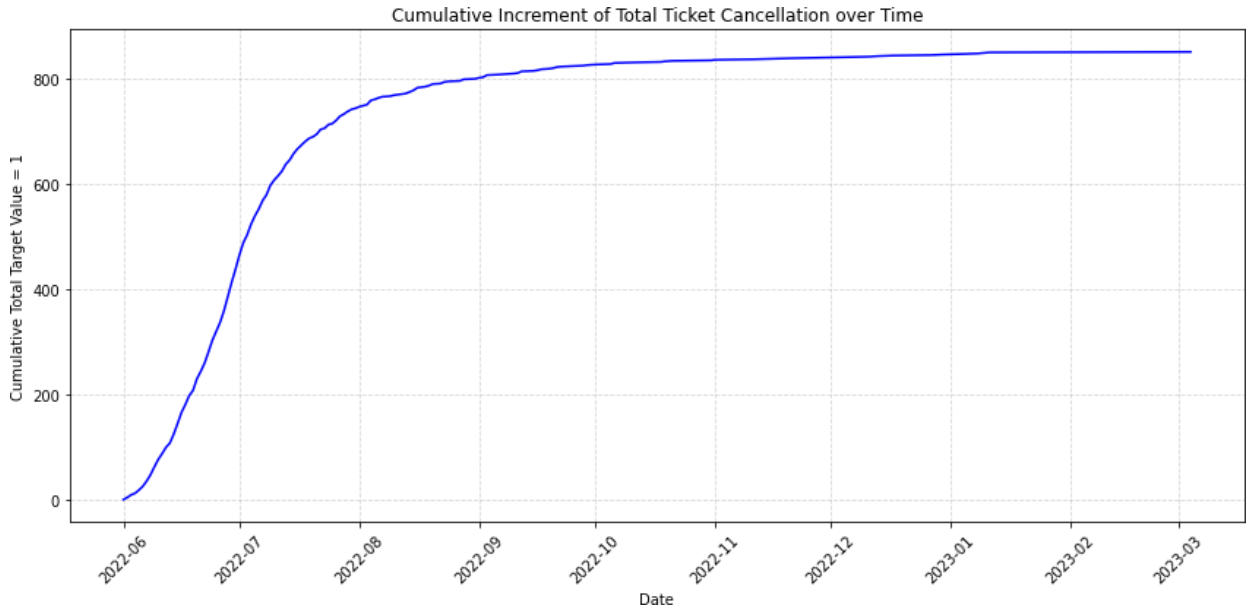

Figure 5.8 Cumulative Increment of Total Ticket Cancellation over Time

In order to gain a more detailed understanding of passengers' ticket cancellation behavior, the cumulative increase in canceled tickets has been examined. As seen in Figure 5.8, the number of ticket cancellations between June 2022 and August 2022 is significantly higher compared to other dates. The reason for this increase could be attributed to a higher volume of ticket sales during these dates, resulting in a proportional increase in cancellation rates. Additionally, the passenger behavior observed here may also have a seasonal factor.

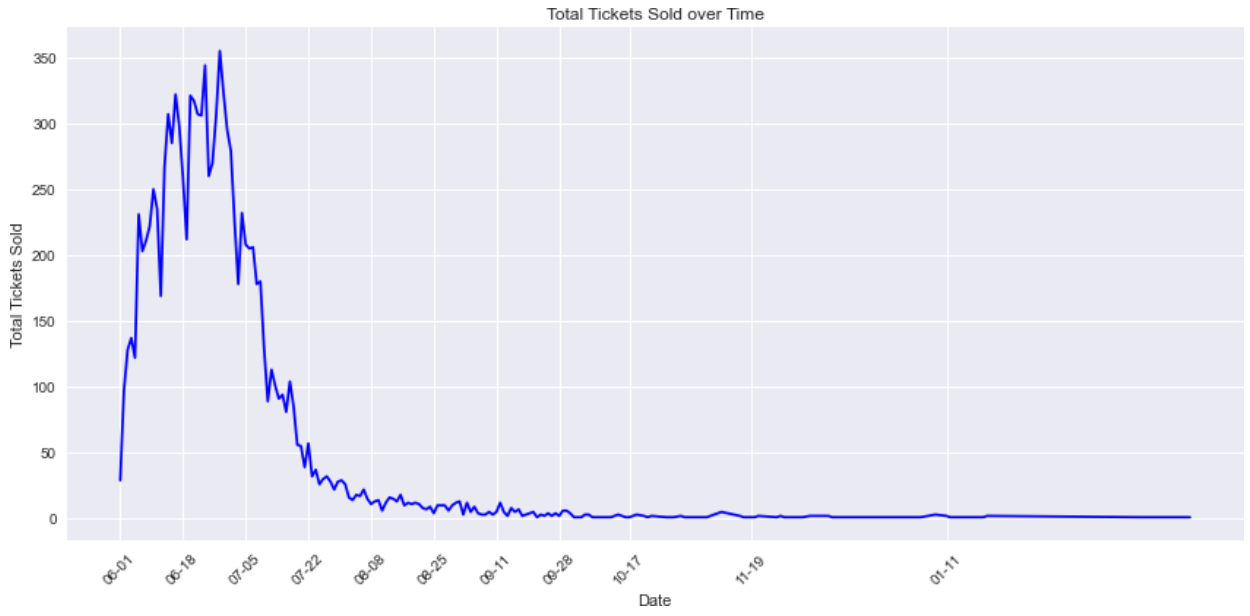

Figure 5.9 Total Tickets Sold over Time

Indeed, when examining Figure 5.9, it is evident that there is a higher volume of ticket sales during the dates when the most ticket cancellations occur. Therefore, it is understandable that there is a cumulative increase in the number of canceled tickets during these specific dates, as reflected in the figure.

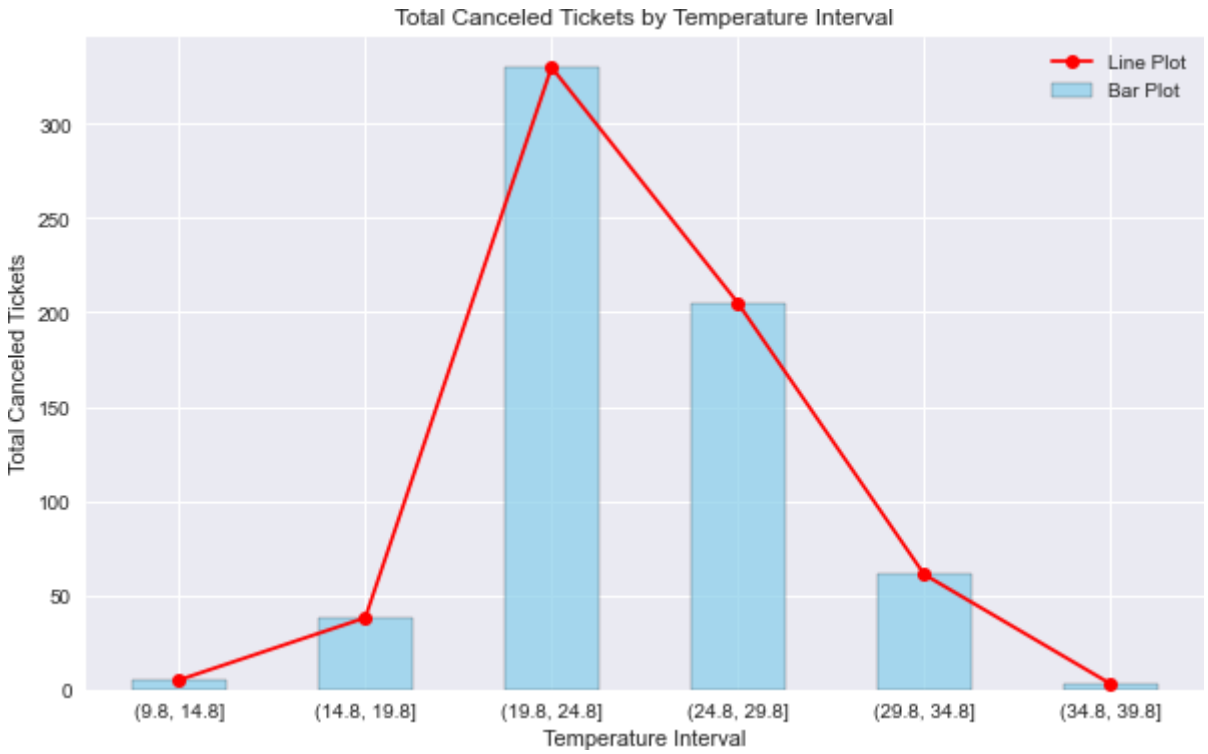

Figure 5.10 Total Canceled Tickets by Temperature Interval

In order to analyze the influence of temperature on ticket cancellation behavior, the total number of ticket cancellations within specific temperature ranges has been examined. As seen in Figure 5.10, it has been determined that there are more ticket cancellations within the range of 19.8 to 24.8.

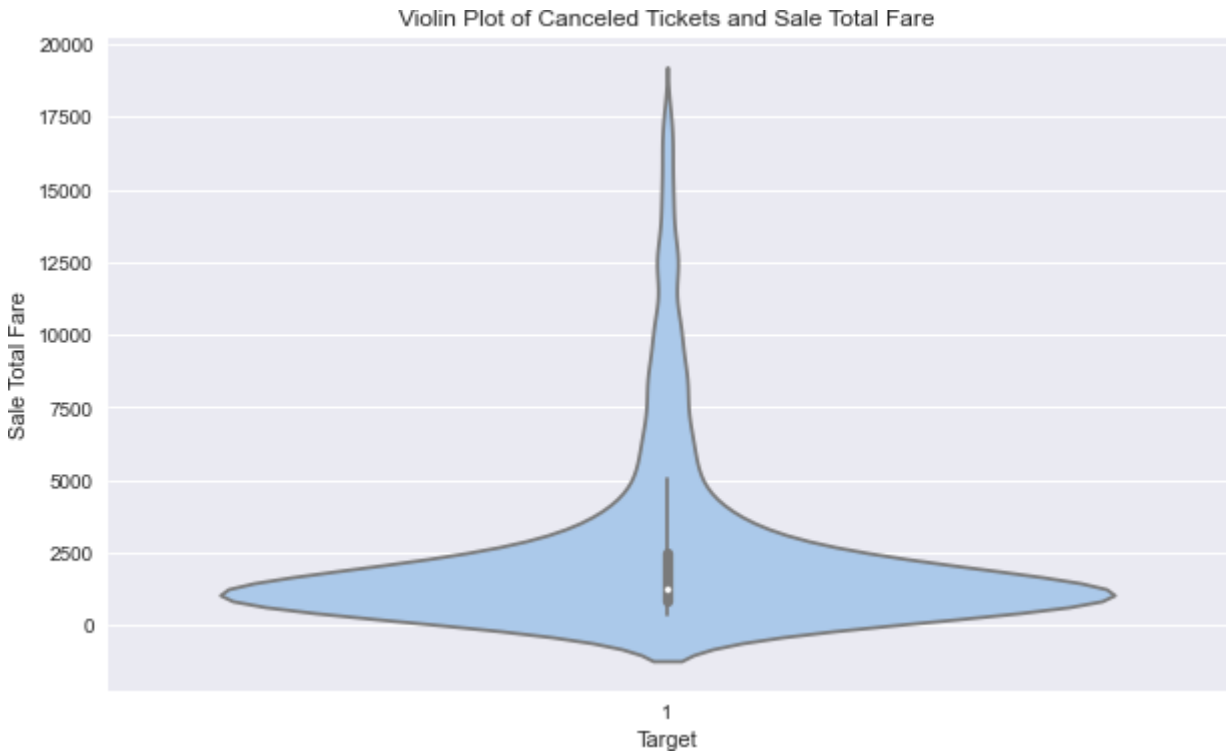

Figure 5.11 Violin Plot of Canceled Tickets and Sale Total Fare

The SaleTotalFare value is also an important factor influencing passengers' decision in ticket cancellation behavior. The higher the ticket price, the greater the amount of loss for the passenger when canceling the ticket. When examining these two variables, as shown in Figure 5.11, it can be observed that the price range of canceled tickets is mostly spread between 0 and 2500 ticket fare interval. As the airplane ticket price increases, the number of canceled tickets decreases rapidly.

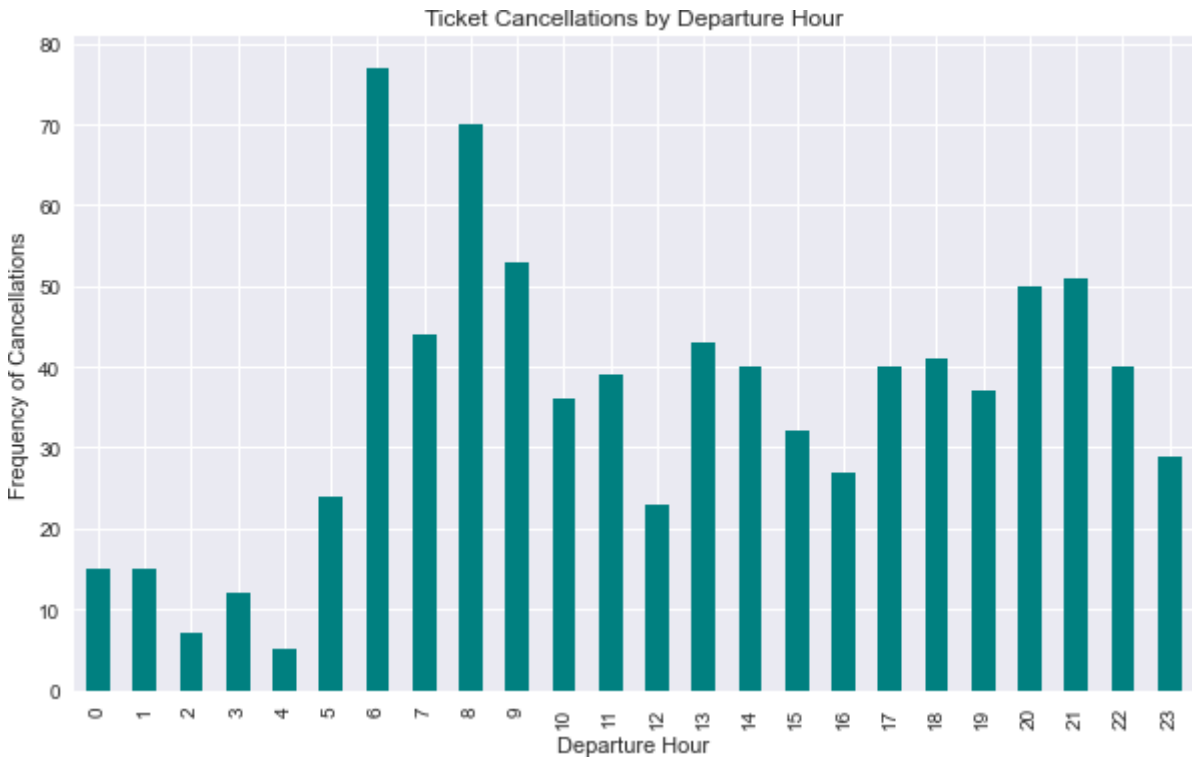

Figure 5.12 Ticket Cancellations by Departure Hour

The departure time of the flight plays a significant role in ticket cancellation or missing the flight. When examining this aspect, as shown in Figure 5.12, it can be observed that the highest number of ticket cancellations occurred between 6 AM and 8 AM. The reason behind this behavior of passengers could be attributed to a higher volume of ticket sales during those hours, which may lead to more cancellations during that time period.

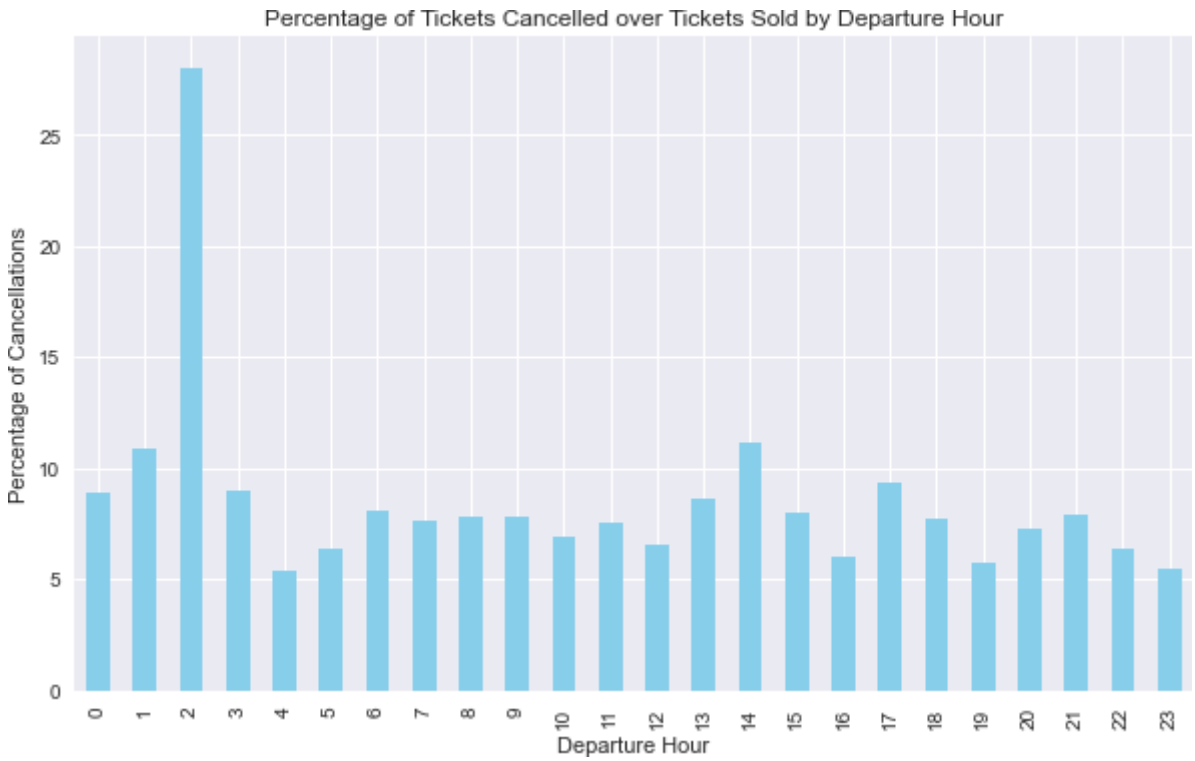

Figure 5.13 Percentage of Tickets Cancelled over Tickets Sold by Departure Hour

To better analyze the passengers' behavior, when examining the ratio of canceled tickets to the number of tickets at departure times, from the Figure 5.13 it can be observed that although the highest number of cancellations occurs between 6 AM and 8 AM, flights scheduled at 2 AM are proportionally canceled more frequently.

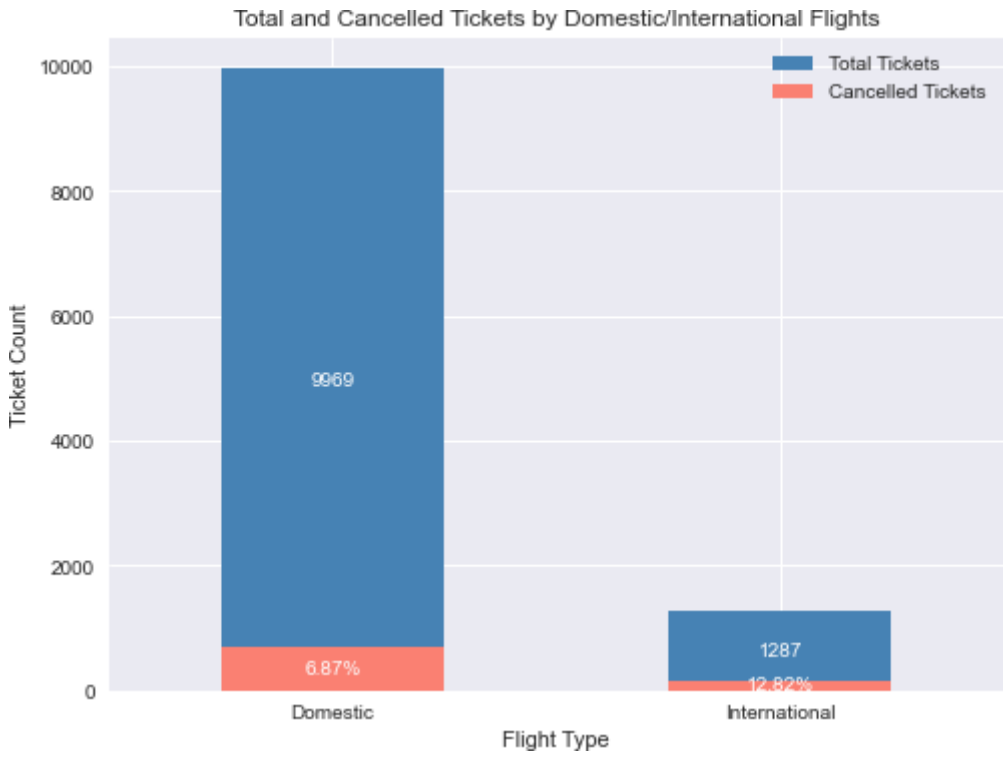

Figure 5.14 Total and Cancelled Tickets by Domestic/International Flights

When examining the passengers' ticket cancellation behavior based on domestic and international flights, as shown in Figure 5.14, the cancellation rate for a total of 1287 international flights is 12.82%. However, when looking at domestic flights with a total of 9969 bookings, the ticket cancellation rate is 6.87%.

The exploratory data analysis (EDA) conducted in this study has yielded valuable results and provided insightful findings. Through thorough examination and visualization of the data, various patterns, trends, and relationships have been uncovered, shedding light on the underlying characteristics of the dataset. Notably, correlation of passengers ticket cancellation behaviour with departure time, departure date, weather temperature, data types, distributions of datas, dominance and predominance of categories provied valuable information. These findings contribute significantly to our understanding of the research problem and provide a solid foundation for further analysis.

5.1.2. Handling insignificant variables

In the context of machine learning, nonsignificant variables refer to features or variables that do not carry useful information for predicting the target variable. Insignificant variables may appear for various reasons such as redundancy, irrelevance, missing values, etc. Nonsignificant variables also mean that there is no causal or logical relationship to the

target variable. Including these types of variables in the model can lead to overfitting or erroneous estimates. On the other hand, meaningless variables increase the complexity of the dataset. As a result, identifying and processing nonsense variables plays an important role in feature selection and feature engineering.

In this direction, as a result of examining the structure and data of the data set used in this study, variables that would not make sense in terms of the model were excluded from the dataset in parallel with the literature. In Table 5.1, the nonsignificant variables removed from the dataset are listed in categories.

Table 5.1 Meaningless Variables in the Dataset

Variables Represented by Information Contained	**Variables Representing Refund Information**	**Categorically Represented Variables**	**Unique Identifier Represented Variables**	**Variables Representing Meaningless Information**	**Variables Representing Incorrect Information**
CancellationExpireDate	DaysBtwBookingAndRefund	AffiliateId	BasketId	BookingDateWeek	InDepartureDateHour
ContactBirthYear	DaysBtwRefundAndDeparture	BookingDateWeekdayNo	MemberId	Classes	OutDepartureDateHour
CustomerCurrency	IsRefunded	Channel	SessionId	EntryDateWeek	
DurationOfStay	IsRefundedWCancelAssurance	DomInt		InDepartureDateWeek	
IsHoliday	RefundAddOn	EntryDateWeekdayNo		IsBooked	
IsMemberAndContactSame	RefundCancellationAssurance	FlightType		MarketingAirlines	
MemberAge	RefundCancellationFee	InDepartureDateWeekdayNo		NumberOfOperatingAirlines	
MemberBirthYear	RefundDate	InDestCity		OperatingAirlines	
MemberGender	RefundDiscount	InOrigCity			
PassengerCountAgeBtw18And24	RefundLP	OutDepartureDateWeekdayNo			
PassengerCountAgeBtw25And34	RefundNetFare	OutDestCity			

PassengerCountAgeBtw35And49	RefundSC	OutOrigCity			
PassengerCountAgeOver50	RefundTotalFare	TripType			
PassengerCountAgeUnder18		UserId			
PassengerCountFemaleADT					
PassengerCountFemaleCHD					
PassengerCountFemaleINF					
PassengerCountFemaleSRC					
PassengerCountFemaleSTD					
PassengerCountMaleADT					
PassengerCountMaleCHD					
PassengerCountMaleINF					
PassengerCountMaleSRC					
PassengerCountMaleSTD					
SaleAddOn					
SaleCancellationFee					
SaleDiscount					
SaleLP					
SaleNetFare					
SaleSC					

The explanation of the columns in Table 5.1 is as follows.

1. Variables Represented by Information Contained: The information contained in the variables in this column is represented by other variables in the data set. According to the related studies, using the same data repeatedly, which contains the same information, is detrimental to the efficiency of the model and adversely affects its performance. Therefore, these data have been excluded from the study.

2. Variables Representing Refund Information: The information contained in the variables in this column represents information about the refund process. Since this study examines the ticket cancellation behaviour of the passengers, data related the refund process is not meaningfull for the machine learning model. It is out of scope for the objective of the model. Hence, these variables were excluded from the study.
3. Categorically Represented Variables: The information contained in the variables in this column is represented categorically in the data set. These variables were excluded from the study because they would not be beneficial in terms of model generalizability and significance.
4. Unique Identifier Represented Variables: The information contained in the variables in this column is represented in the data set in unique identifier format. Unique identifiers do not provide any beneficial information to the model and analysis. Hence, these variables were excluded from the study.
5. Variables Representing Meaningless Information: Variables in this column were excluded from the study because they contain meaningless information for the objective model. The data used in the model should have a correlation with the objective of the model.
6. Variables Representing Incorrect Information: The information contained in the variables in this column is incorrect, erroneous. These variables were excluded from the study because they would not be beneficial in terms of model generalizability and significance.

As a result, the examination and analysis of the data in this direction, total of 68 variables were removed from the data set used in the study.

5.1.3. Handling variables with date and time knowledge

Variables containing date and time information cannot be used as a structure in machine learning models. Variables in machine learning models should be expressed numerically in terms of structure. Accordingly, variables containing time and time information should be made meaningful for the model.

In this context, the variables 'BookingDate', 'EntryDate', 'InArrivalDate', 'InDepartureDate', 'OutArrivalDate' and 'OutDepartureDate', which are time information in the "year-month-day" format in the dataset used in the study, are numerically day, month. ,year,day of the week, week of the year information was split to represent the information and added to the data as a variable.

The variables 'InArrivalTime', 'InDepartureTime', 'OutArrivalTime' and 'OutDepartureTime', which are time information in the "hour-minute" format, were split into numerical representations and added to the data as variables.

Table 5.2 Fragmentation of Variables Containing Date and Time Information

Variables Containing Date and Time Information	**Variables Generated from Variables Containing Date and Time Information**
BookingDate	BookingDateDay,BookingDateMonth,BookingDate,BookingDateWeek,BookingDateWeekday
EntryDate	EntryDateDay,EntryDateMonth,EntryDateYear,EntryDateWeek,EntryDateWeekday
InArrivalDate	InArrivalDateDay,InArrivalDateMonth,InArrivalDateYear,InArrivalDateWeek,InArrivalDateWeekday
InDepartureDate	InDepartureDateDay,InDepartureDateMonth,InDepartureDateYear,InDepartureDateWeek,InDepartureDateWeekday
OutArrivalDate	OutArrivalDateDay,OutArrivalDateMonth,OutArrivalDateYear,OutArrivalDateWeek,OutArrivalDateWeekday
OutDepartureDate	OutDepartureDateDay,OutDepartureDateMonth,OutDepartureDateYear,OutDepartureDateWeek,OutDepartureDateWeekday
InArrivalTime	InArrivalDateHour
InDepartureTime	InDepartureDateHour
OutArrivalTime	OutArrivalDateHour
OutDepartureTime	OutDepartureDateHour

As a result, 33 new variables were produced from the variables containing date and time information as a result of these processes.

5.1.4. One-way flights data & round trip flights data

After a detailed examination of the data within the scope of the study and the examination of similar studies in the literature, it was determined that airway passengers would describe the ticket cancellation process differently due to the fact that the information of flights with One-Way flight type and Round-Trip flight type make sense in their own way. It was filtered according to One-Way flight type and Round-Trip flight type and analyzed in terms of two separate datasets.

5.1.5. Explanatory data analysis

The data used in the study was filtered according to flights with one-way flight type, and as a result of this process, the filtered data began to be examined by positioning it as a separate dataset.

The one-way dataset includes 9473 observations and 86 variables. After examining the structure of the data, since the data contains one way flight information, the variables that contain return flight information are "InDepartureDateHour", "InArrivalDateHour", "InArrivalDateWeekday", "InArrivalDateMonth", "InArrivalDateDay", "InArrivalDateYear", "InArrivalDateYear", "IDepartureInArrivalDateWeekday" InOrigCountry", "InOrigContinent", "InOrigCityName", "InOrigAirport","InDestCountry","InDestContinent", "InDestCityName", "InDestAirport", "TripTypeName", and "DurationOfTrip" were omitted from the data because they did not provide meaningful model information. As a result, 17 variables were removed from the data as a result of these operations.

When the variable structure of the data is examined, there are 43 categorical variables, 17 numerical variables and 8 cardinal variables. Figure 5.15 shows the countplot of the distribution of the variables.

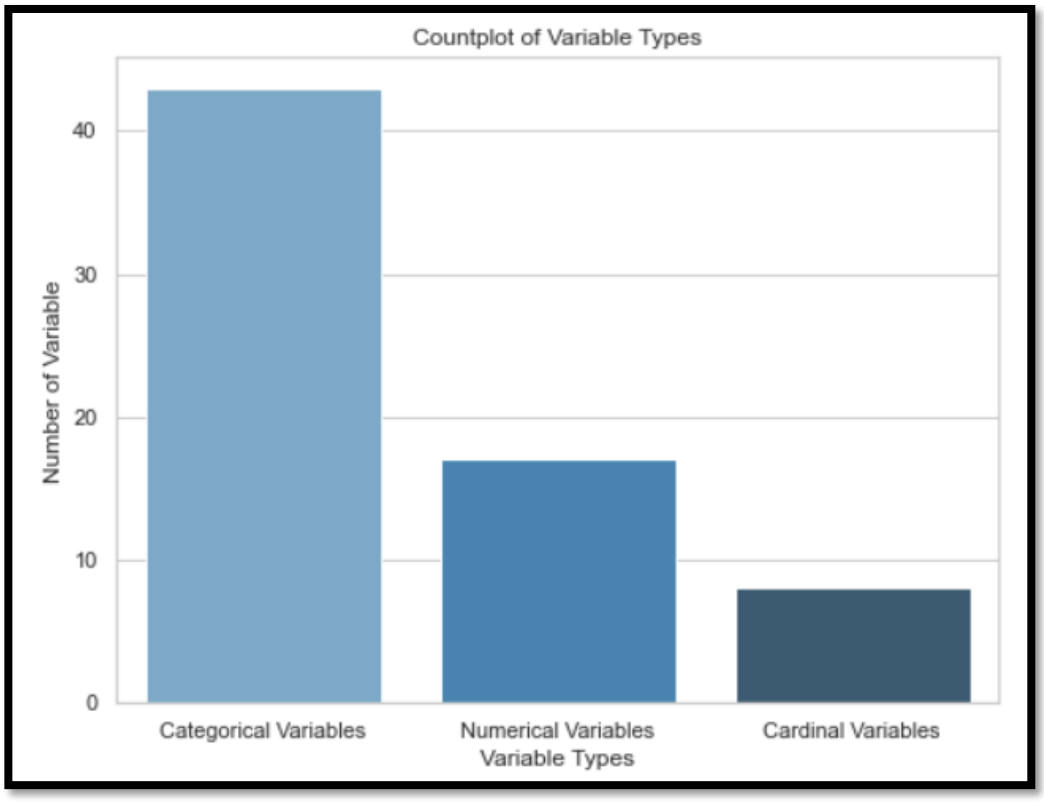

Figure 5.15 Distribution of Variable Types of One-Way Flight Data

When the distribution of the target variable in One-Way flight data is examined, it is seen that 8819 airway passenger did not request cancellation after ticketing, and 654 airway passenger requested cancellation after ticketing. Figure 5.16 shows the pie chart of the target variable distribution.

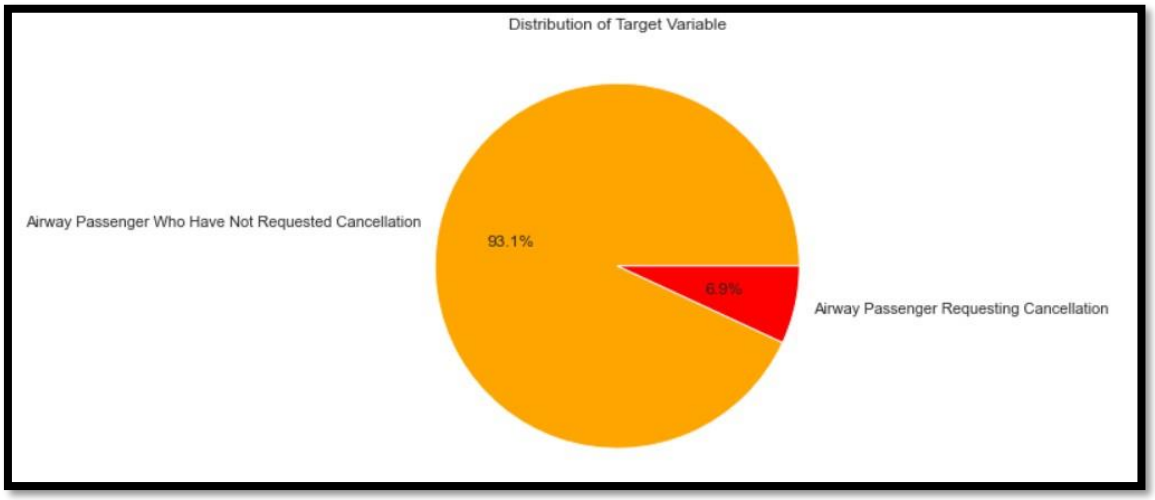

Figure 5.16 Pie Chart for Target Variable of One-Way Data

In order to understand the interdependencies between the variables, to identify possible multicollinearity problems and to guide feature selection, correlation analysis is performed for the data set and a correlation matrix is created. Accordingly, the correlation matrix of One Way flight data is given in Figure 5.17.

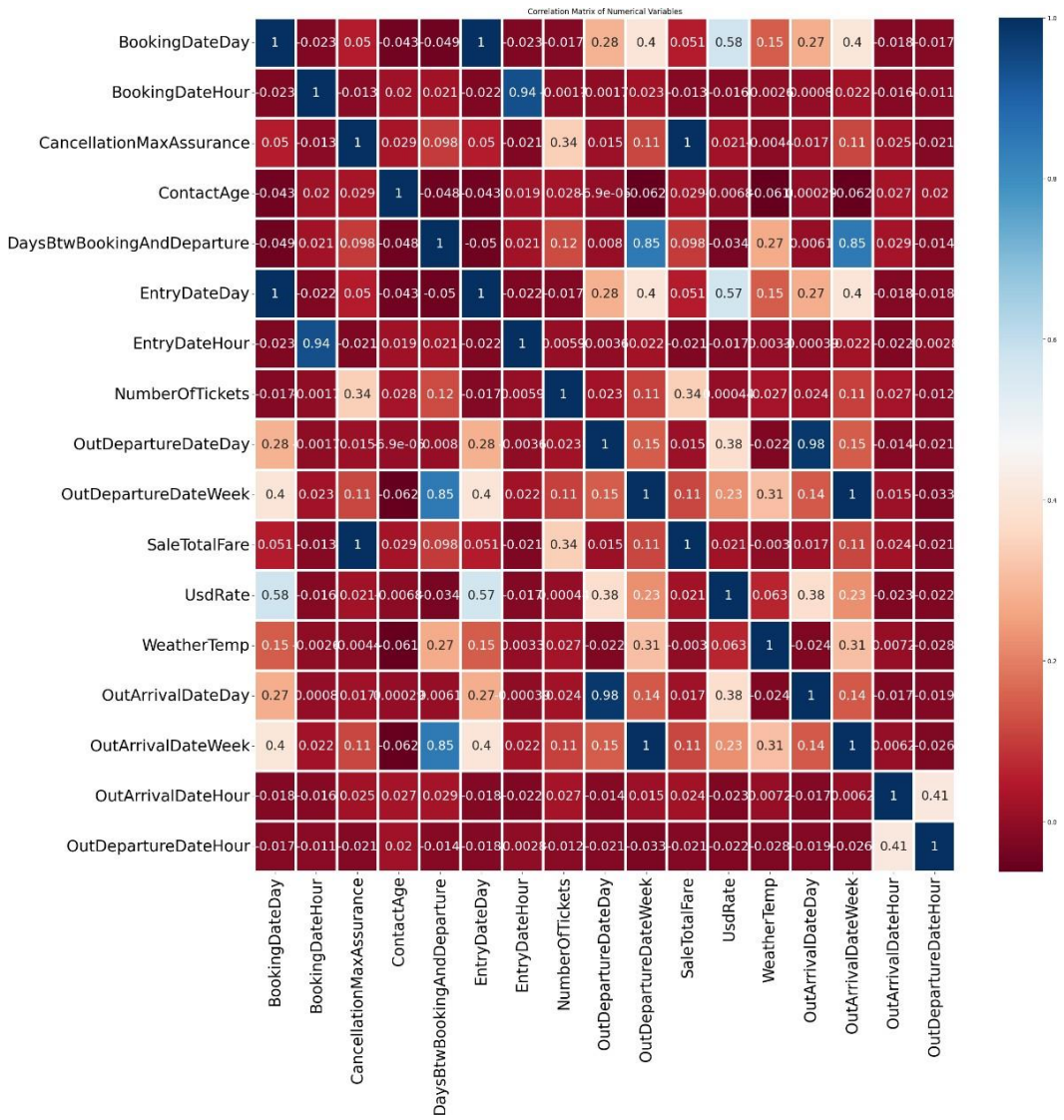

Figure 5.17 Correlation Matrix of One Way Trip Data

Variables with 90% or more correlation in the correlation matrix given in Figure 5.1 6 are 'EntryDateDay', 'EntryDateHour', 'InDepartureDateYear', 'PassengerCountTotal', 'Sale TotalFare', 'OutArrivalDateYear', 'OutArrivalDateDay', 'OutArrivalDateMonth' and 'OutArr ivalDateWeek'.

After conducting examinations on the one-way dataset, the analysis transitioned to the round-trip dataset, where the same examinations were performed. The round-trip data includes 1783 observations and 86 variables. When the variable structure of the data is examined, there are 47 categorical variables, 26 numerical variables and 13 cardinal variables. Figure 5.18 shows the countplot of the distribution of the variables.

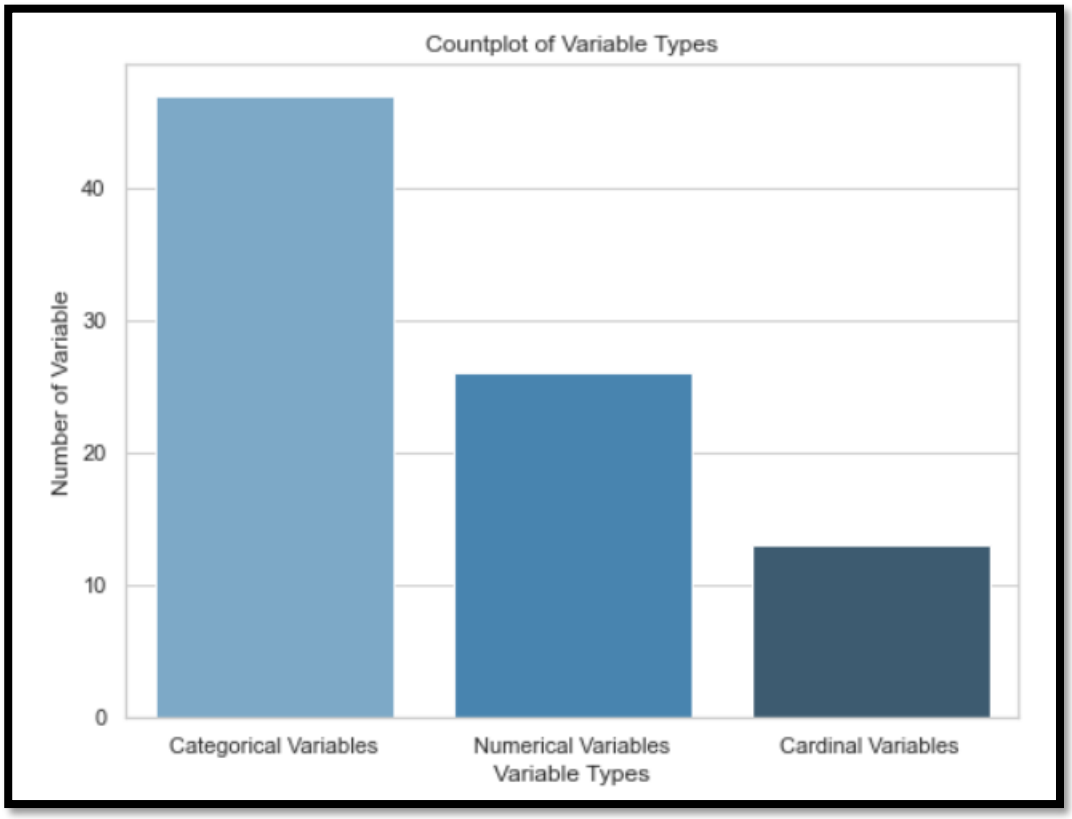

Figure 5.18 Distribution of Variable Types of Round Trip Flight Data

When the distribution of the target variable in Round Trip flight data is examined, it is seen that 1587 airway passenger did not request cancellation after ticketing, and 196 airway passenger requested cancellation after ticketing. Figure 5.19 shows the pie chart of the target variable distribution.

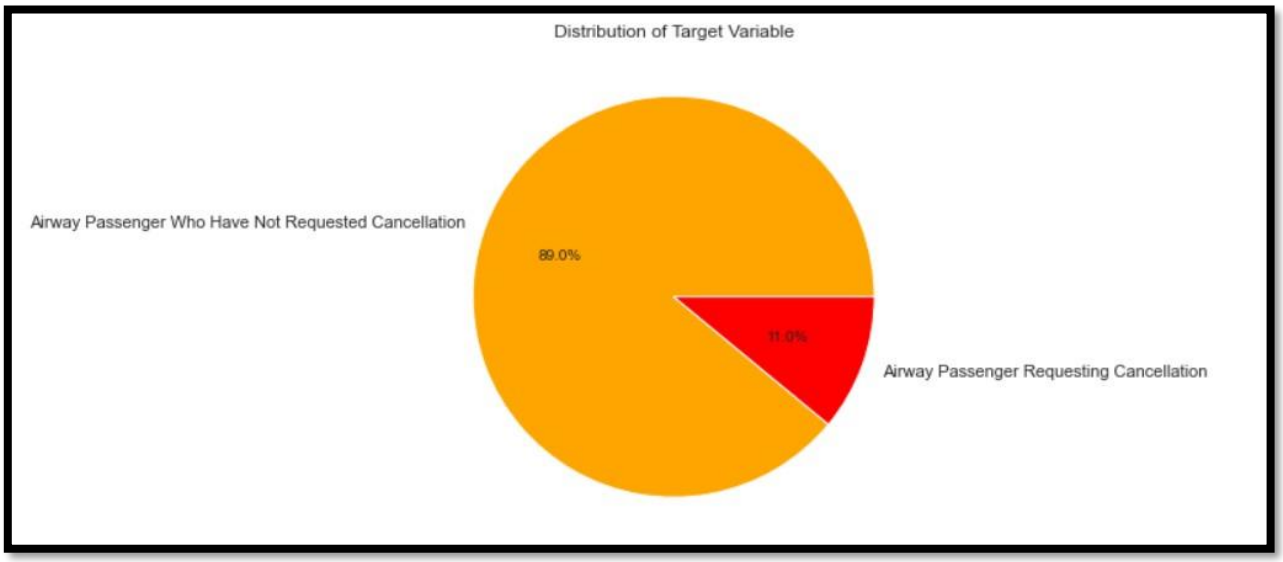

Figure 5.19 Pie Chart for Target Variable of Round Trip Data

The correlation matrix of Round Trip flight data is given in Figure 5.20.

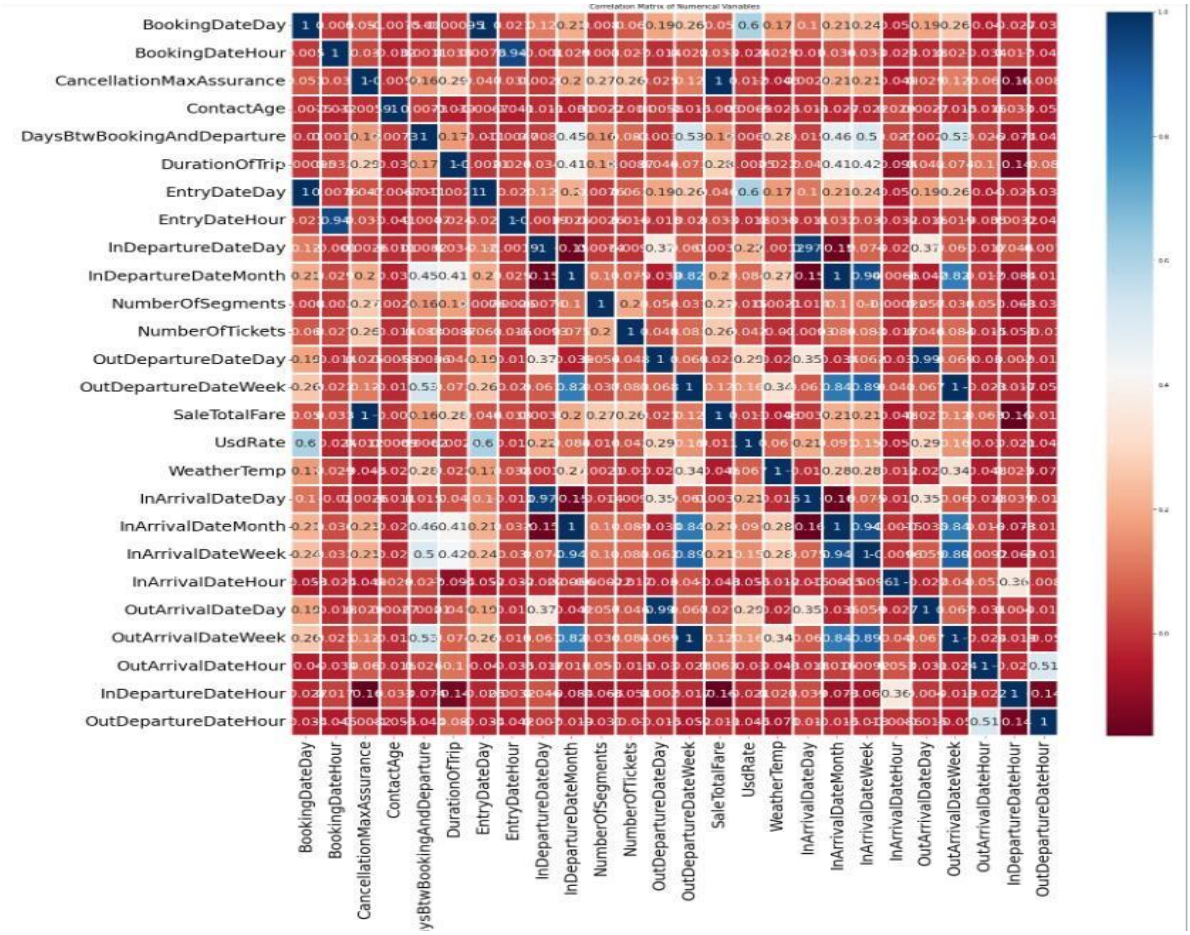

Figure 5.20 Correlation Matrix of Round Trip Flight Data

Variables with 90% or more correlation in the correlation matrix given in Figure 5 .19 are 'EntryDateDay', 'EntryDateHour', 'OutDepartureDateWeek', 'SaleTotalFare', 'InArriv

alDateYear', 'InArrivalDateDay', 'InArrivalDateMonth', 'InArrivalDateWeek', 'OutArrivalDateYear', 'OutArrivalDateDay', 'OutArrivalDateMonth' and 'OutArrivalDateWeek'.

5.1.6. Handling missing values

Missing values in the data have a negative effect on the model's learning process of the data pattern. After examining the structure of the variables with missing values, filling in the missing values will have a positive effect on the models learning the pattern of the data.

Accordingly, in Table 5.3 and Figure 5.21, the values of the missing values in the One-Way flights data, specific to the variables, are given.

Table 5.3 Missing Value Table for One-Way Data

Variables	Number of Missing Values	Ratio of Missing Values (%)	Data Type
WeatherTemp	1709	18	float64
SaleCountryCode	379	4	object
OutDestContinent	9	0.1	object
OutOrigContinent	7	0.07	object
ContactNationality	5	0.05	object

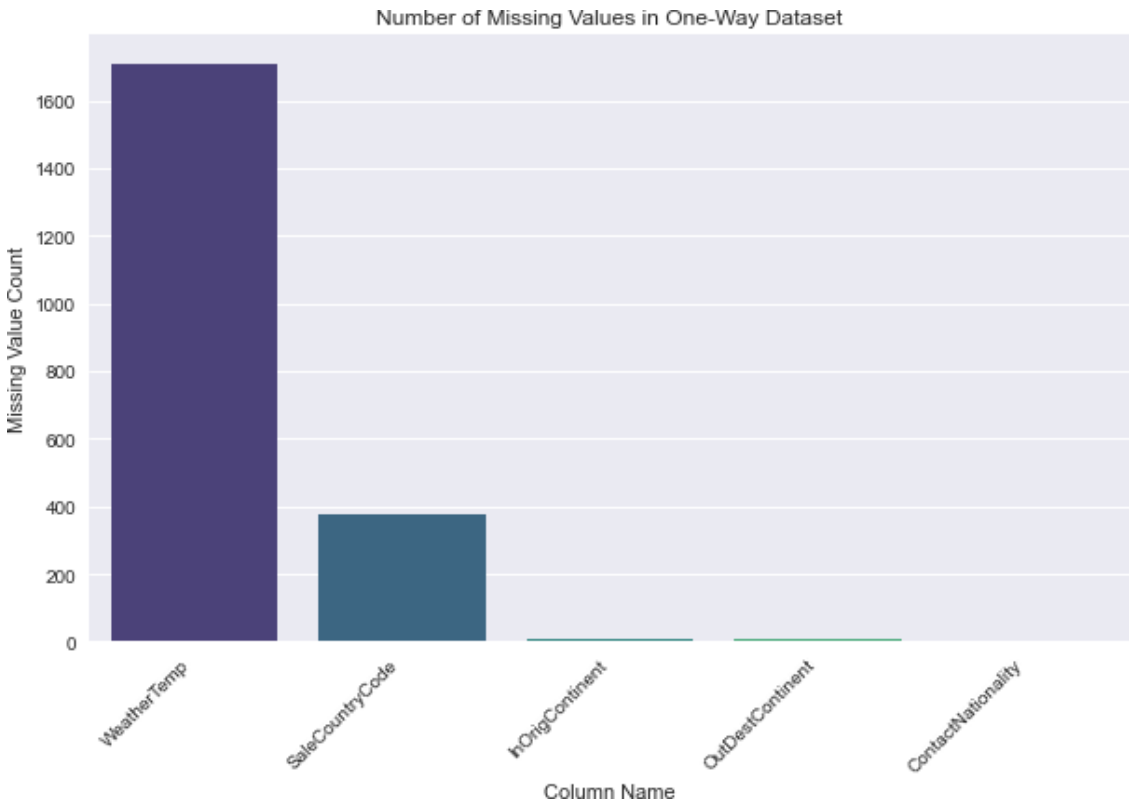

Figure 5.21 Number of Missing Values in One-Way Dataset

In line with the information in Table 5.3, variable-based missing values were examined. After the examinations, since the "WeatherTemp" variable is a numerical type variable and has a normal distribution, the loss observations were filled with the mean of the variable. Since the variable "SaleCountryCode" is a variable of categorical type, the missing observations were filled with the mode of the variable. Since the "ContactNationality" variable is a categorical type variable, the missing observations were filled with the mode of the variable. As a result of the examination of the missing observations of the "OutDestContinent" and "OutOrigContinent" variables, it was determined that there were flights to America. In this direction, the missing observations of the relevant variables were filled with the continental code of America "US". Since the variable "ContactNationality" is a variable of categorical type, the missing observations were filled with the mode of the variable.

Table 5.4, the values of the missing values in the Round Trip flights data, specific to the variables, are given.

Table 5.4 Missing Value Table for Round Trip Data

Variables	Number of Missing Values	Ratio of Missing Values(%)	Data Type
WeatherTemp	327	18.3	float64
SaleCountryCode	51	2.9	object
InOrigContinent	19	1.1	object
OutDestContinent	18	1.1	object
ContactNationality	2	0.1	object
InDestContinent	2	0.1	object
InDestCountry	1	0.1	object
InOrigAirport	1	0.1	object
InOrigCityName	1	0.1	object
InOrigCountry	1	0.1	object
InDepartureDateWeekday	1	0.1	object
OutOrigContinent	1	0.1	object
InDestCityName	1	0.1	object
InDestAirport	1	0.1	object

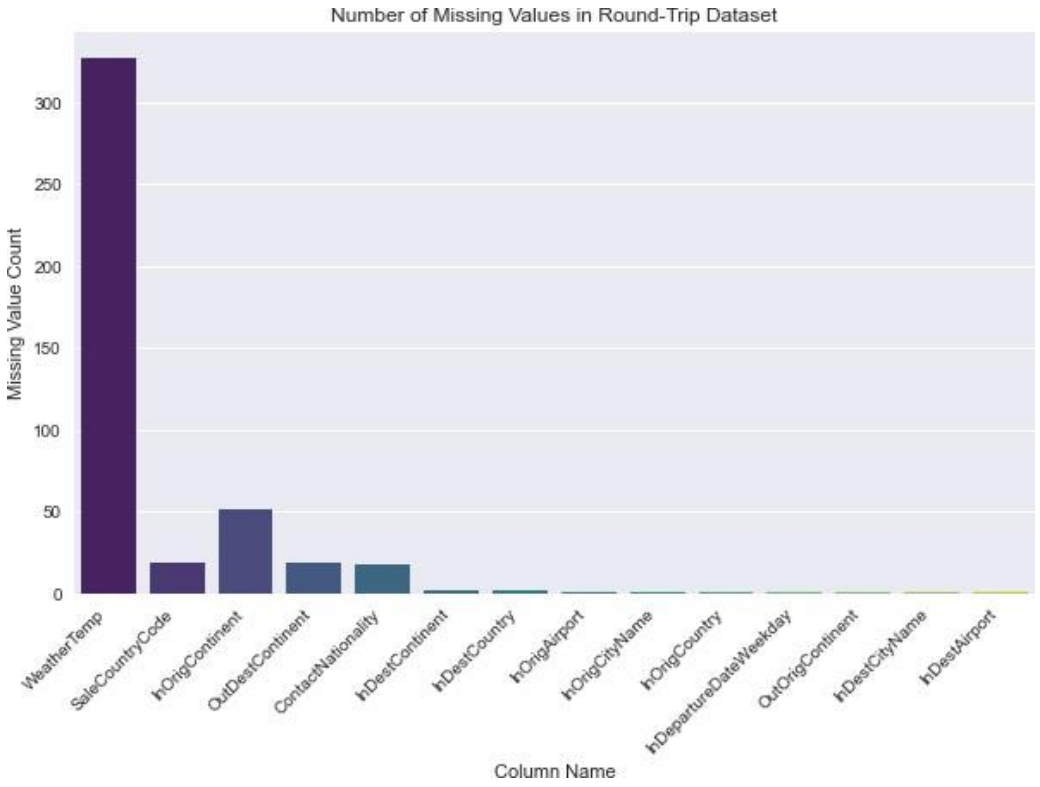

Figure 5.22 Number of Missing Values in Round-Trip Dataset

In line with the information in Table 5.4, variable-based missing values were examined. After the examinations, since the "WeatherTemp" variable is a numerical type variable and has a normal distribution, the loss observations were filled with the mean of the variable. Since the variable "SaleCountryCode" is a variable of categorical type, the missing observations were filled with the mode of the variable. Since the "ContactNationality" variable is a categorical type variable, the missing observations were filled with the mode of the variable. As a result of the examination of the missing observations of the "OutDestContinent", "InOrigContinent" and “InDestContinent” variables, it was determined that there were flights to America. In this direction, the missing observations of the relevant variables were filled with the continental code of America "US". Since the variable "ContactNationality" is a variable of categorical type, the missing observations were filled with the mode of the variable. Since the missing values in the variables "InOrigAirport", "InOrigCityName", "InOrigCountry", "InDepartureDateWeekday", "OutOrigContinent", "InDestCityName" and "InDestAirport" contain null values in the information for a single ticket, the observation with this single row is deleted from the data and lost. observations have been taken.

5.1.7. Handling outliers

Outliers in the data negatively affect the model in terms of providing generalizability while learning the data pattern of the model. It is necessary to detect outliers and process them in accordance with the structure of the data. According to the distribution of numerical variables within the scope of descriptive statistics, information can be obtained in terms of outliers and actions can be taken in line with this information.

In this context, the descriptive statistics values of the numerical variables in the One Way flight data are given in Table 5.5.

Table 5.5 Descriptive Statistics of One-Way Data

Variables	Number of Observation	Mean of Observation	Standart Deviation of Observation	Minimum Value of Observation	0.25 Quantile of Observation	0.50 Quantile of Observation	0.75 Quantile of Observation	Maximum Value of Observation
BookingDateDay	9473	15.0966	8.4244	1	8	15	22	30
BookingDateHour	9473	14.9295	5.6072	0	12	15	19	23
CancellationMaxAssurance	9473	1434.324	1899.8372	0	674.99	837.89	1487.83	33919.16
ContactAge	9473	37.067	13.5855	0	26	34	46	94
DaysBtwBookingAndDeparture	9473	11.3865	17.5103	0	1	5	15	215
EntryDateDay	9473	15.0908	8.4291	1	8	15	22	31
EntryDateHour	9473	14.9602	5.5257	0	12	15	19	23
NumberOfTickets	9473	1.4523	0.8963	1	1	1	2	16
OutDepartureDateDay	9473	15.4059	8.6195	1	8	15	23	31
OutDepartureDateWeek	9473	25.5965	2.7244	1	24	25	27	50
SaleTotalFare	9473	1595.7187	2110.8458	230	749.99	930.99	1653.98	37687.96
UsdRate	9473	16.9862	0.357	16.3562	16.6189	17.1985	17.2905	17.3478
WeatherTemp	9473	24.2472	3.3226	9.8	22.2	24.2472	25.6	37.9
OutArrivalDateDay	9473	15.4208	8.6165	1	8	15	23	31
OutArrivalDateWeek	9473	25.6105	2.7268	1	24	25	27	50
OutArrivalDateHour	9473	12.9905	6.7904	0	8	13	19	23
OutDepartureDateHour	9473	13.4684	6.214	0	8	13	19	23

When Table 5.5 is examined, it is seen that there is a discrepancy in the numerical variables in the One Way flight data. In this context, the results of the anomaly analysis according to the observations of the numerical variables found in the One Way flight data below and above the quarterly values of 0.05 and 0.95 are given in Table 5.6.

Table 5.6 Outlier Analysis of One-Way Data

Variables	**Outliers**
BookingDateDay	False
BookingDateHour	False
CancellationMaxAssurance	True
ContactAge	False
DaysBtwBookingAndDeparture	True
EntryDateDay	False
EntryDateHour	False
NumberOfTickets	True
OutDepartureDateDay	False
OutDepartureDateWeek	True
SaleTotalFare	True
UsdRate	False
WeatherTemp	False
OutArrivalDateDay	False
OutArrivalDateWeek	True
OutArrivalDateHour	False
OutDepartureDateHour	False

When Table 5.6 is examined, a discrepancy has been detected in the numerical variables "CancellationMaxAssurance","NumberOfTickets","OutDepartureDateWeek", "SaleTotalFare" and "OutArrivalDateWeek" in the data containing One Way flight data. When the structure of the variables is examined, the 0.05 and 0.95 quartile values of the

"SaleTotalFare" and "CancellationMaxAssurance" variables were determined as the lower and upper limits, and outliers were found to these thresholds with the IQR method.

The descriptive statistics values of the numerical variables in the Round Trip flight data are given in Table 5.7.

Table 5.7 Descriptive Statistics of Round Trip Data

Variables	Number of Observation	Mean of Observation	Standart Deviation of Observation	Minimum Value of Observation	0.25 Quantile of Observation	0.50 Quantile of Observation	0.75 Quantile of Observation	Maximum Value of Observation
BookingDateDay	1781	14.1915	8.4108	1	7	14	21	30
BookingDateHour	1781	14.7451	5.2378	0	12	15	19	23
CancellationMaxAssurance	1781	3698.6508	4820.5549	572.38	1443.58	1825.18	3750.28	72602.95
ContactAge	1781	39.3498	12.3991	12	30	37	48	105
DaysBtwBookingAndDeparture	1781	20.2695	30.8225	0	3	9	26	279
DurationOfTrip	1781	8.3622	12.1334	1	3	5	9	173
EntryDateDay	1781	14.1729	8.407	1	7	14	20	30
EntryDateHour	1781	14.6912	5.1961	0	11	15	19	23
InDepartureDateDay	1781	16.1679	8.1276	1	10	16	23	31
InDepartureDateMonth	1781	6.7788	1.1089	1	6	7	7	12
NumberOfTickets	1781	1.9646	1.4235	1	1	2	2	12
OutDepartureDateDay	1781	14.7631	8.3197	1	8	14	22	31
OutDepartureDateWeek	1781	26.4166	4.3485	1	24	26	27	52
SaleTotalFare	1781	4102.3931	5353.5002	635.98	1603.98	2027.98	4105.92	80669.95
UsdRate	1781	16.9607	0.3608	16.3876	16.5588	17.0925	17.2905	17.3478
WeatherTemp	1781	24.4711	2.8568	12.2	22.7	24.4711	25.7	34.2
InArrivalDateDay	1781	16.1465	8.155	1	10	16	23	31
InArrivalDateMonth	1781	6.7838	1.1091	1	6	7	7	12
InArrivalDateWeek	1781	27.4104	4.7715	2	24	26	29	52

InArrivalDateHour	1781	14.8591	7.2515	0	10	17	21	23
OutArrivalDateDay	1781	14.7754	8.3179	1	8	14	22	31
OutArrivalDateWeek	1781	26.4228	4.3436	1	24	26	27	52
OutArrivalDateHour	1781	12.0943	6.0062	0	8	11	17	23
InDepartureDateHour	1781	15.5699	5.9231	0	11	18	20	23

When Table 5.7 is examined, it is seen that there is a discrepancy in the numerical variables in the One Way flight data. In this context, the results of the anomaly analysis according to the observations of the numerical variables found in the Round Trip flight data below and above the quarterly values of 0.05 and 0.95 are given in Table 5.8.

Table 5.8 Outlier Analysis of Round Way Data

Variables	**Outliers**
BookingDateDay	False
BookingDateHour	False
CancellationMaxAssurance	True
ContactAge	False
DaysBtwBookingAndDeparture	True
EntryDateDay	False
EntryDateHour	False
NumberOfTickets	True
OutDepartureDateDay	False
OutDepartureDateWeek	True
SaleTotalFare	True
UsdRate	False

WeatherTemp	False
OutArrivalDateDay	False
OutArrivalDateWeek	True
OutArrivalDateHour	False
OutDepartureDateHour	False
DurationOfTrip	True
InDepartureDateDay	False
InDepartureDateMonth	True
InArrivalDateDay	False
InArrivalDateMonth	True
InArrivalDateWeek	False
InArrivalDateHour	False
InDepartureDateHour	False

When Table 5.8 is examined, a discrepancy has been detected in the numerical variables “DurationOfTrip”, “InDepartureDateMonth”,” InArrivalDateMonth “,"CancellationMaxAssurance","NumberOfTickets", "SaleTotalFare, ”DaysBtwBookingAndDeparture” and "OutArrivalDateWeek" in the data containing One Way flight data. When the structure of the variables is examined, the 0.05 and 0.95 quartile values of the "SaleTotalFare" and "CancellationMaxAssurance" variables were determined as the lower and upper limits, and outliers were found to these thresholds with the IQR method.

5.1.8. Encoding of categorial variables

Machine learning algorithms perform learning processes on numerical type data. As a result of examining the structure of categorical variables expressed as strings, they should be represented numerically in accordance with their structure. In addition, the ratio of the classes of categorical variables in the data is important in order for the algorithms to learn

the pattern of the data. Classes that are small in the class distribution become poor learners for the model. Weak learners can be brought together according to a threshold determined according to the structure of the data, making it meaningful for the model.

In this direction, when the class distribution of the categorical variables of the One Way flight data and Round Trip data are examined, the weak learner classes with a rate below this threshold compared to the 0.05 threshold were brought together to increase the representation power in the "Rare" class. Since categorical variables need to be represented numerically in order to be represented in terms of the model, One-Hot Encoding method was applied to the categorical variables in the data, so that they were represented numerically.

After the data preprocessing of the One Way flight data, 9473 observations to be trained on the model had 155 variables.

5.1.9. Model training

Within the scope of this study, data preprocessing steps for modeling on two separate datasets in One Way flight type and Round Trip flight type breakdown are explained in the "Data Preprocessing" section in order to predict the cancellation request behavior for the tickets purchased by airway passengers with machine learning approaches. In this part of the study, after the training of two separate datasets, whose preprocessing has been completed, on the algorithms used for classification problems, the examination of performance metrics, the examination of performance metrics as a result of hyperparameter optimizations of the algorithms and their feature importance are examined.

When the studies carried out within the scope of this study were examined, One Way flight data and Round Trip flight data were fitted to the Random Forest (RF), Gradient Boosting Machine (GBM), XGBoosting(XGBoost) and Light Gradient Boosting Machine(LGBM) algorithms, which are classification algorithms under the title of advanced tree methods. After the fitting process, the Cross Validation model validation method was used and the classification performance metrics Accuracy, Precision and ROC-AUC were calculated by using the Stratify Cross Validation method in order to prevent the dominance of the majority class over the minority class and the performance metrics of the data to avoid bias. Table 5.9, Table 5.10 shows the performance values obtained as a result of these processes.

Table 5.9 Base Model Validation Results for One-Way Flight Data

Algorithms	Accuracy Score	Precision Score	ROC-AUC Score
RF Algorithm	0.9319	0.6917	0.6571
GBM Algorithm	0.9312	0.5312	0.6799
XGBoost Algorithm	0.9293	0.4285	0.6136
LGBM Algorithm	0.932	0.5755	0.6382

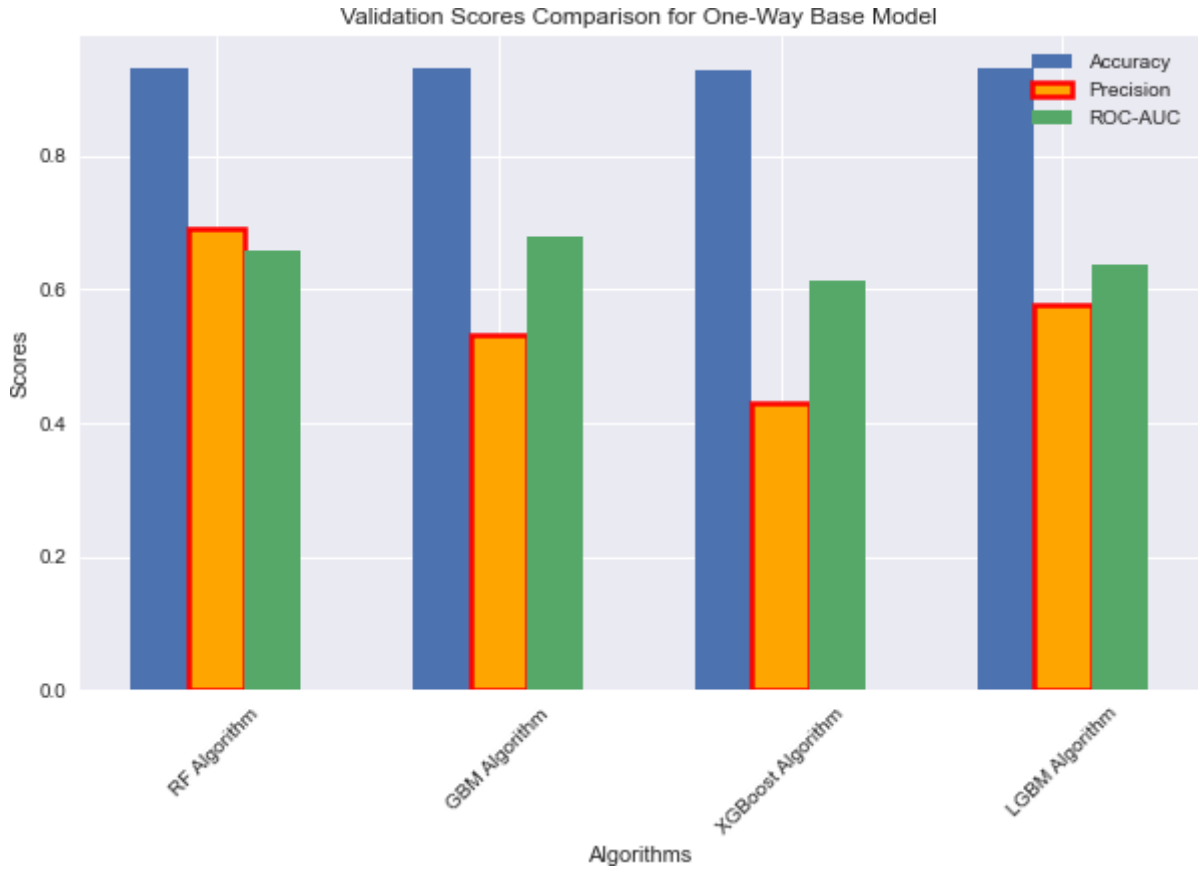

Figure 5.23 Validation Scores Comparison for One-Way Base Model

Table 5.10 Base Model Validation Results for Round Trip Data

Algorithms	Accuracy Score	Precision Score	ROC-AUC Score
RF Algorithm	**0.8899**	**0.4833**	**0.6435**
GBM Algorithm	**0.886**	**0.4199**	**0.6516**
XGBOOST Algorithm	**0.8815**	**0.3452**	**0.616**
LGBM Algorithm	**0.8883**	**0.4284**	**0.6125**

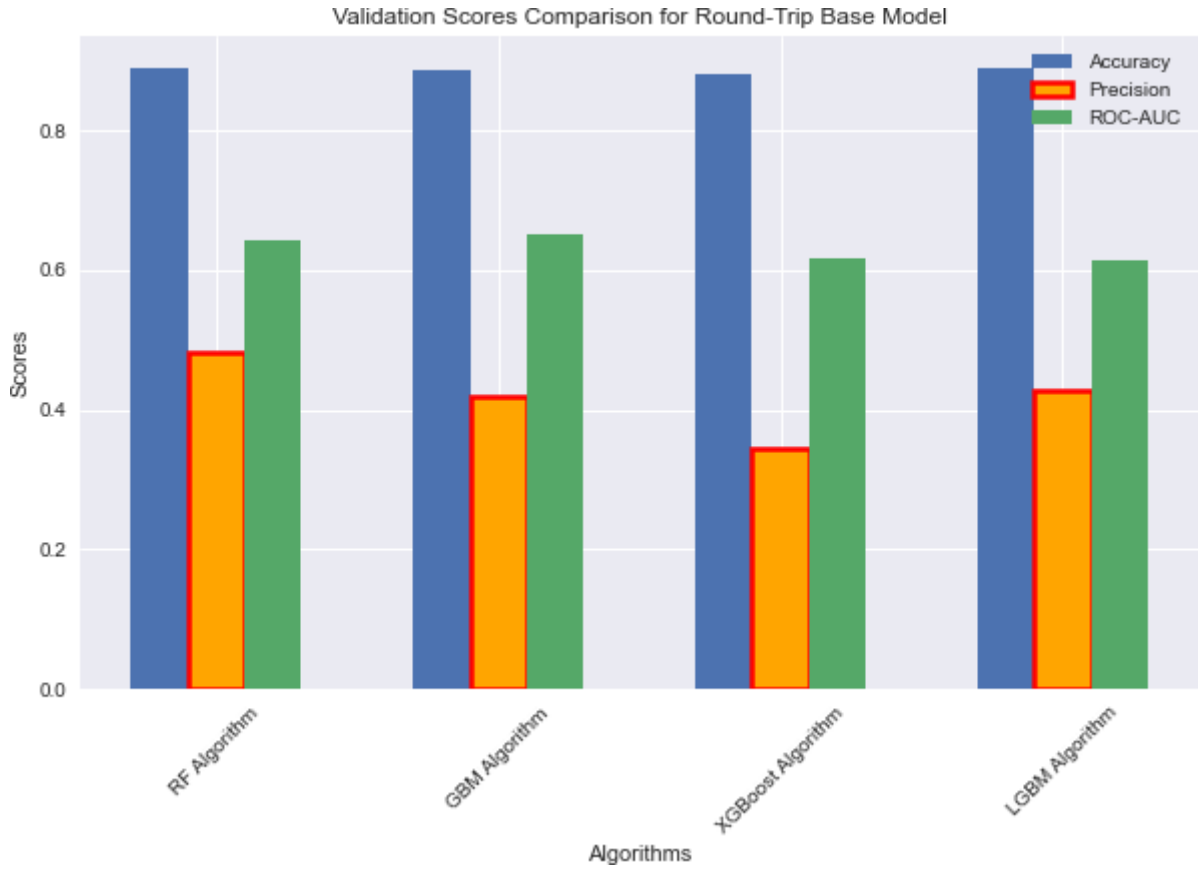

Figure 5.24 Validation Scores Comparison for Round-Trip Base Model

5.1.10. Hyperparameter optimization process and final model

After obtaining the results from the base models, further steps were taken to improve these results such as hyperparameter optimization. Hyperparameter optimization refers to the process of finding the optimal set of hyperparameters for a machine learning model. Hyperparameters are parameters that are set before the learning process begins, unlike the model's internal parameters that are learned from the data. The process of hyperparameter optimization typically involves techniques such as grid search, random search, or more advanced methods like Bayesian optimization or genetic algorithms. In this study, grid search algorithm is used. Grid search is a hyperparameter optimization technique that exhaustively searches through a predefined grid of hyperparameter values to find the optimal combination that yields the best model performance.

In this regard, it is necessary to know which parameters need to be modified. In this study, changes were made to various parameters to improve model performance. The

parameters tuned in this study are max_depth, max_features, min_samples_split, n_estimators, subsample and learning_rate.

- The max_depth parameter sets an upper limit on the depth of a decision tree, preventing overfitting and controlling the complexity of the model.
- The max_features parameter determines the maximum number of features to consider during the tree splitting process or feature selection, providing flexibility and preventing excessive reliance on certain features.
- The min_samples_split parameter specifies the minimum number of samples required to split an internal node in a decision tree, allowing for fine-tuning the tree structure and avoiding unnecessary splits.
- In ensemble methods like random forests or gradient boosting, the n_estimators hyperparameter defines the number of individual estimators, such as decision trees, to be used, influencing the model's overall complexity and predictive power.
- The subsample parameter, used in gradient boosting algorithms, randomly samples a fraction of the training data for each tree, enabling stochastic gradient boosting and reducing overfitting.
- Lastly, the learning_rate parameter determines the contribution of each tree in gradient boosting algorithms, affecting the weight of each tree's prediction and influencing the overall model's convergence speed and regularization.

By tuning these hyperparameters appropriately, machine learning models are optimized for better performance and generalization.

The success metrics obtained as a result of hyperparameter optimization are as in Table 5.11 and 5.12.

Table 5.11 Tuned Model Validation Results for One-Way Flight Data

Algorithms	Accuracy Score	Precision Score	ROC-AUC Score
RF Algorithm	**0.9315**	**0.7333**	**0.6783**
GBM Algorithm	**0.932**	**0.6137**	**0.6864**
XGBoostAlgorithm	**0.9314**	**0.7**	**0.6894**
LGBM Algorithm	**0.9318**	**0.75**	**0.6654**

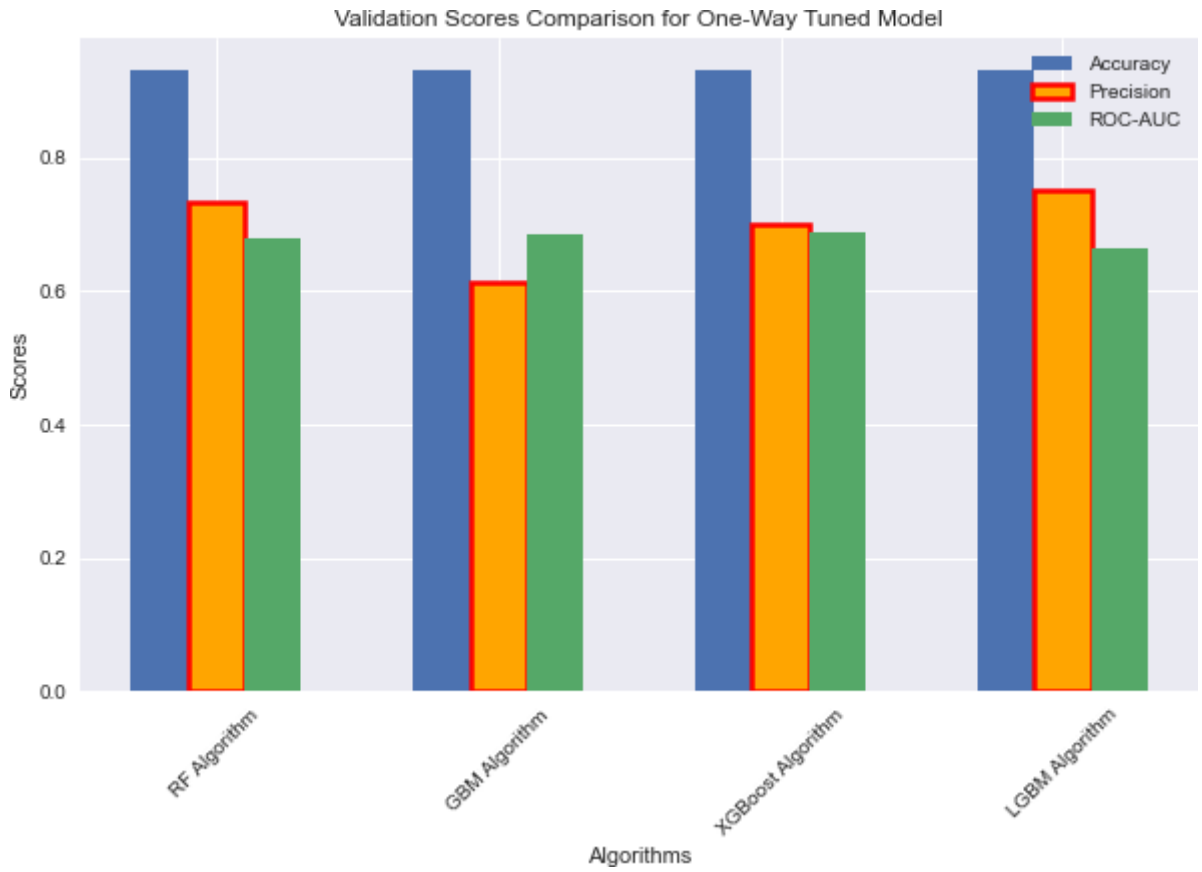

Figure 5.25 Validation Scores Comparison for One-Way Tuned Model

Table 5.12 Tuned Model Validation Results for Round Trip Flight Data

Algorithms	Accuracy Score	Precision Score	ROC-AUC Score
RF Algorithm	**0.8905**	**0.5214**	**0.6328**
GBM Algorithm	**0.8888**	**0.5874**	**0.6172**
XGBoost Algorithm	**0.8911**	**0.6115**	**0.6332**
LGBM Algorithm	**0.8911**	**0.6642**	**0.6199**

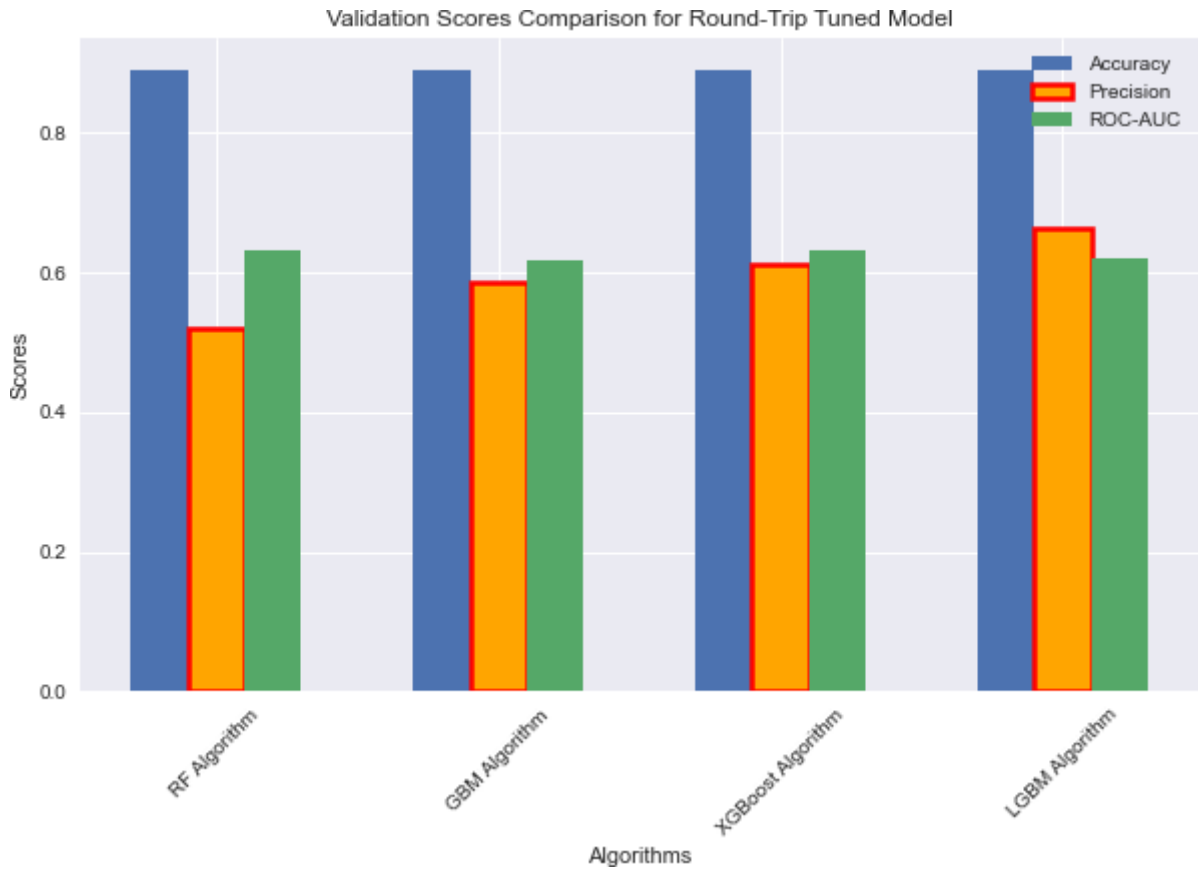

Figure 5.26 Validation Scores Comparison for Round-Trip Tuned Model

After performing hyperparameter optimization on models trained on the one-way dataset, certain improvements in the validation scores were achieved. As shown in Figure 5.27, for the model trained using the Random Forest (RF) algorithm, a hyperparameter optimization study resulted in a 0.04 increase in the precision score. In the model trained using the Gradient Boosting Machine (GBM) algorithm, this improvement was measured as 0.08. The highest score improvement, 0.27, was observed in the model trained with the XGBoosting (XGBoost) algorithm. However, the model trained with the Light Gradient Boosting Machine (LGBM) algorithm achieved the highest precision score of 0.75 with a 0.17 increase.

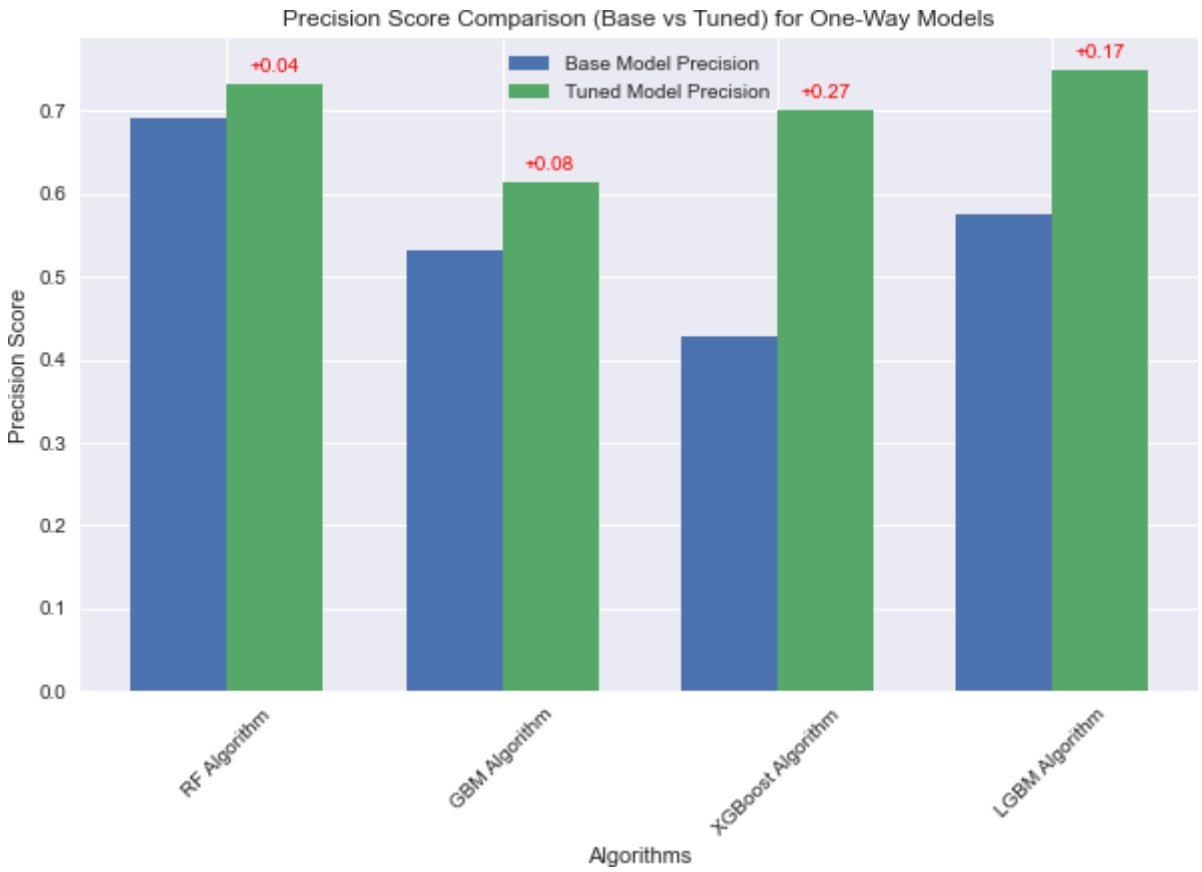

Figure 5.27 Precision Score Comparison (Base vs Tuned) for One-Way Models

Similarly, models trained on the round-trip dataset also showed various improvements after hyperparameter optimization. As shown in Figure 5.28, for the model trained using the Random Forest (RF) algorithm, a hyperparameter optimization study resulted in a 0.04 increase in the precision score. In the model trained using the Gradient Boosting Machine (GBM) algorithm, this improvement was measured as 0.17. The model trained with the XGBoosting (XGBoost) algorithm achieved a precision score improvement of 0.27, reaching a precision score of 0.6642, which was the highest. The model trained with the Light Gradient Boosting Machine (LGBM) algorithm achieved a precision score improvement of 0.24, reaching a precision score of 0.6642.

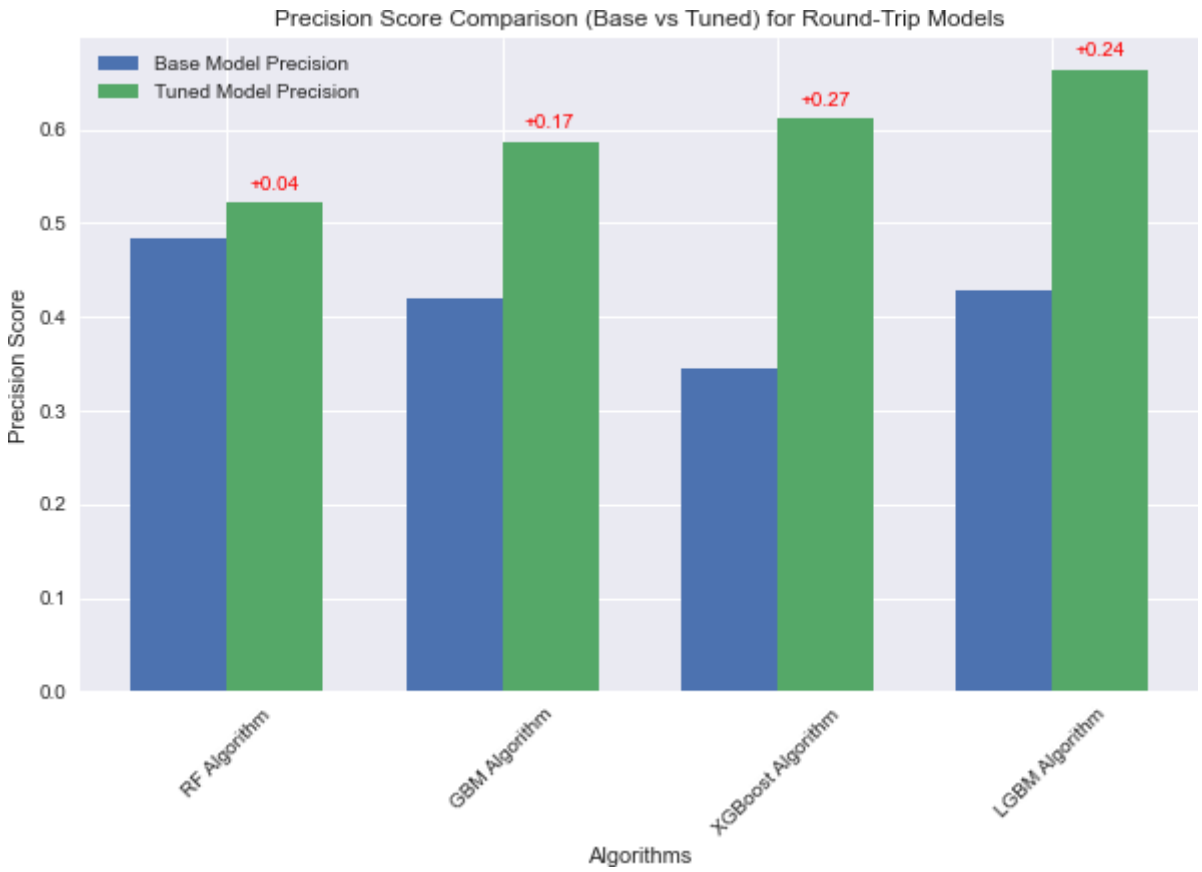

Figure 5.28 Precision Score Comparison (Base vs Tuned) for Round-Trip Models

5.1.11. Feature importances

For machine learning algorithms, the information added to the model by the variables in the learning process of the model is as important as the importance of the model performance success metrics values. Especially in tree-based machine learning algorithms, the contribution of the variables to the model in terms of learning is important since the models are trained with random variable selection and the best performance models are created. In this way, the model performance can be increased by processing high-importance variables.

In this context, the 10 variables with the highest significance level in the final models created after hyperparameter optimization were represented by bar graphs.

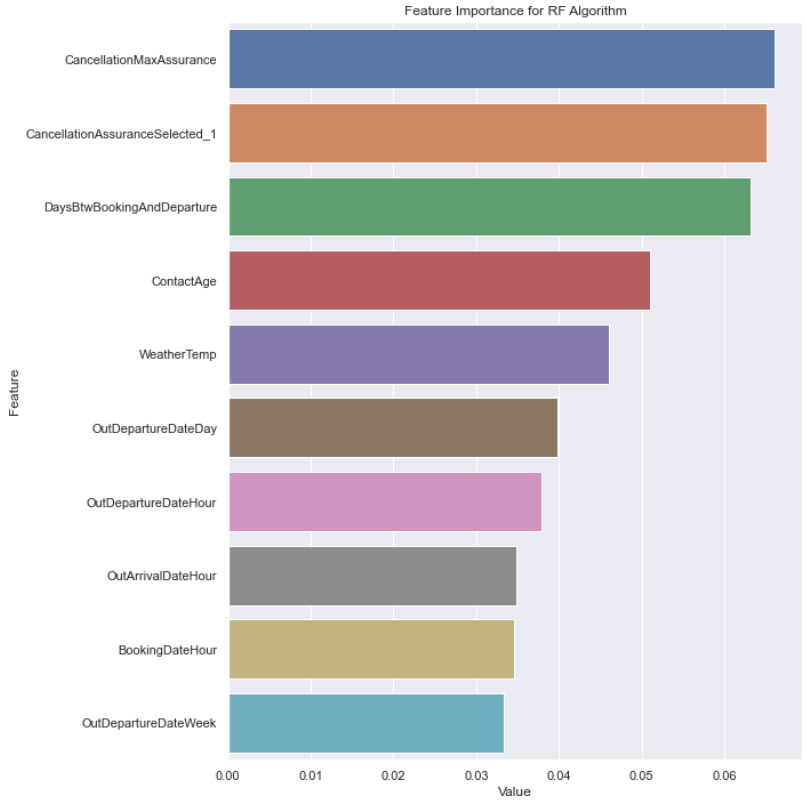

Figure 5.29 Feature Importance Plot of RF Algortihm of One Way Flight Data

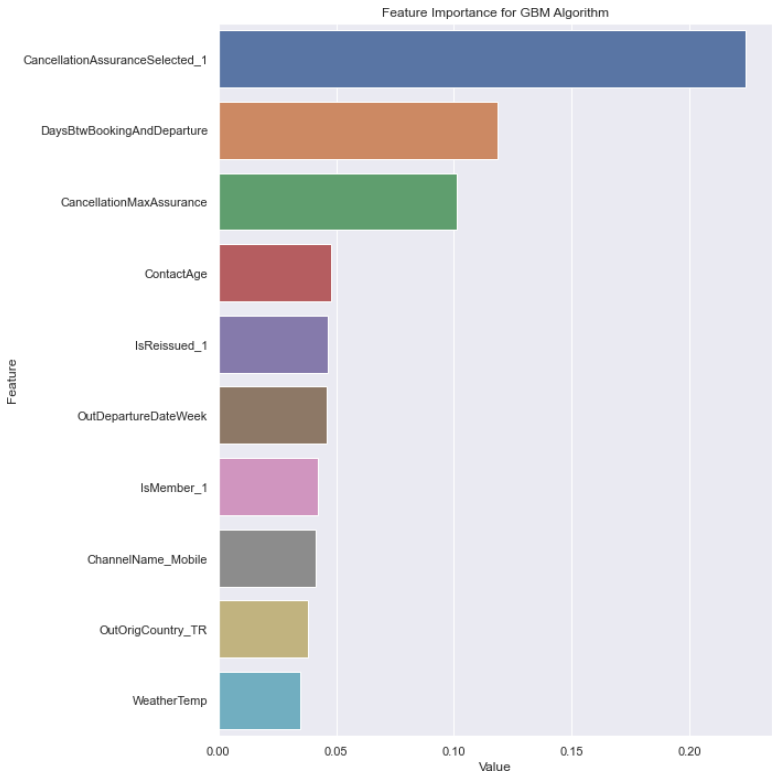

Figure 5.30 Feature Importance Plot of GBM Algortihm of One Way Flight Data

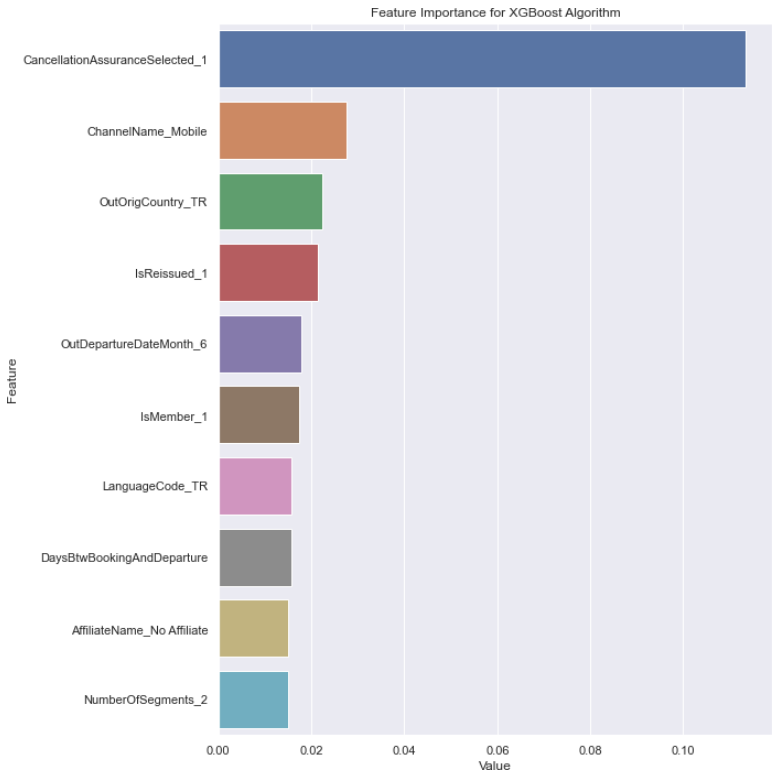

Figure 5.31 Feature Importance Plot of XGBoost Algortihm of One Way Flight Data

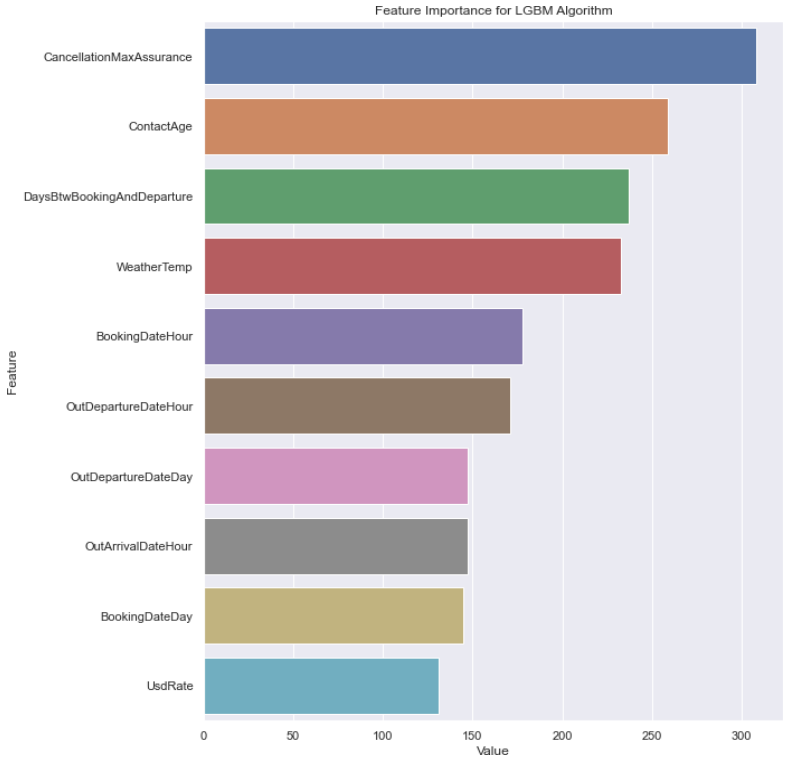

Figure 5.32 Feature Importance Plot of LGBM Algortihm of One Way Flight Data

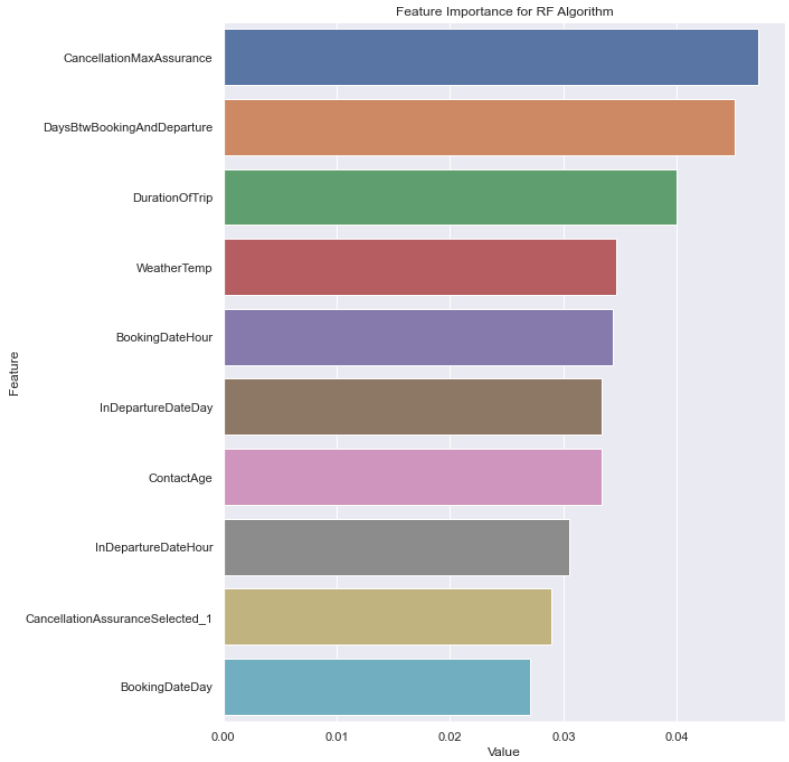

Figure 5.33 Feature Importance Plot of RF Algortihm of Round Trip Flight Data

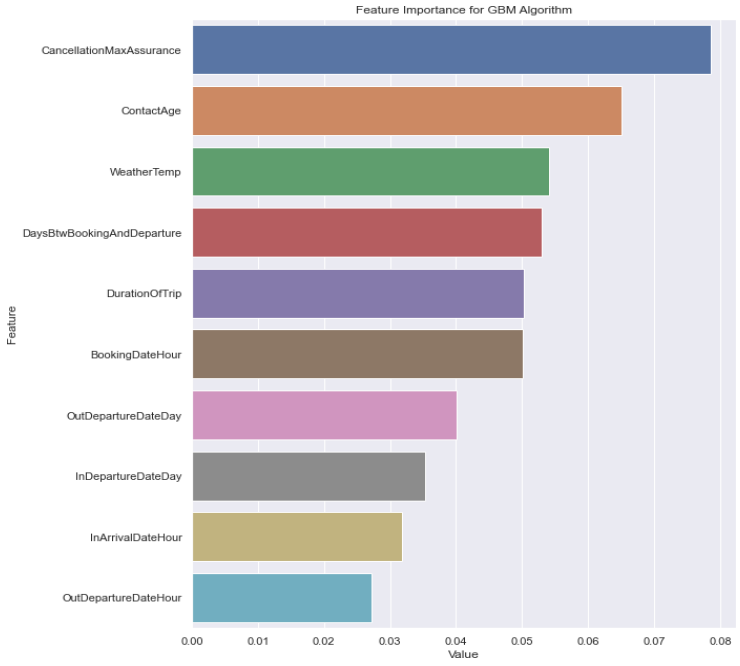

Figure 5.34 Feature Importance Plot of GBM Algortihm of Round Trip Flight Data

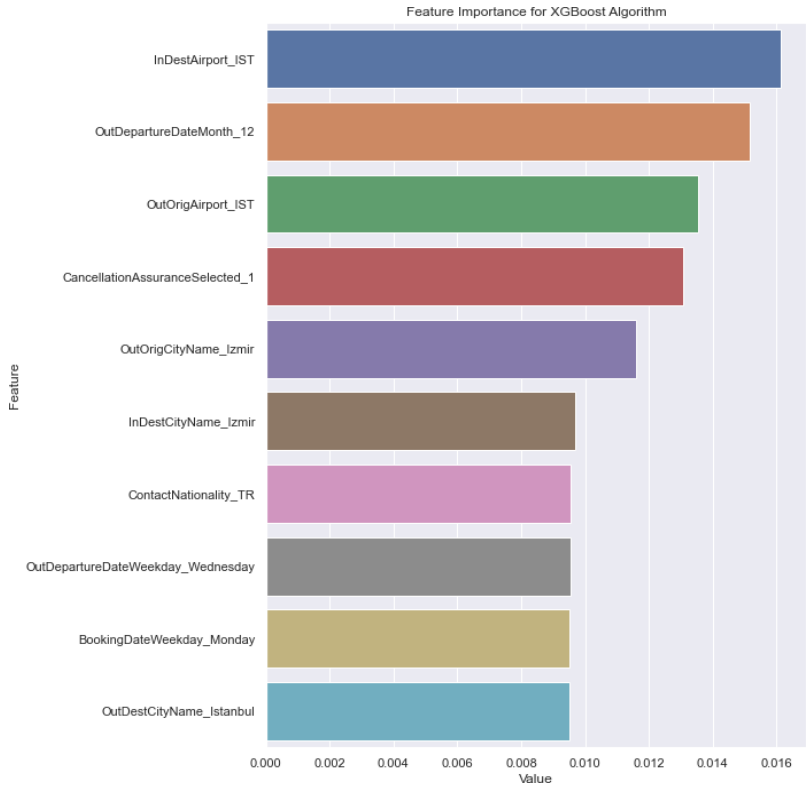

Figure 5.35 Feature Importance Plot of XGBoost Algortihm of Round Trip Flight Data

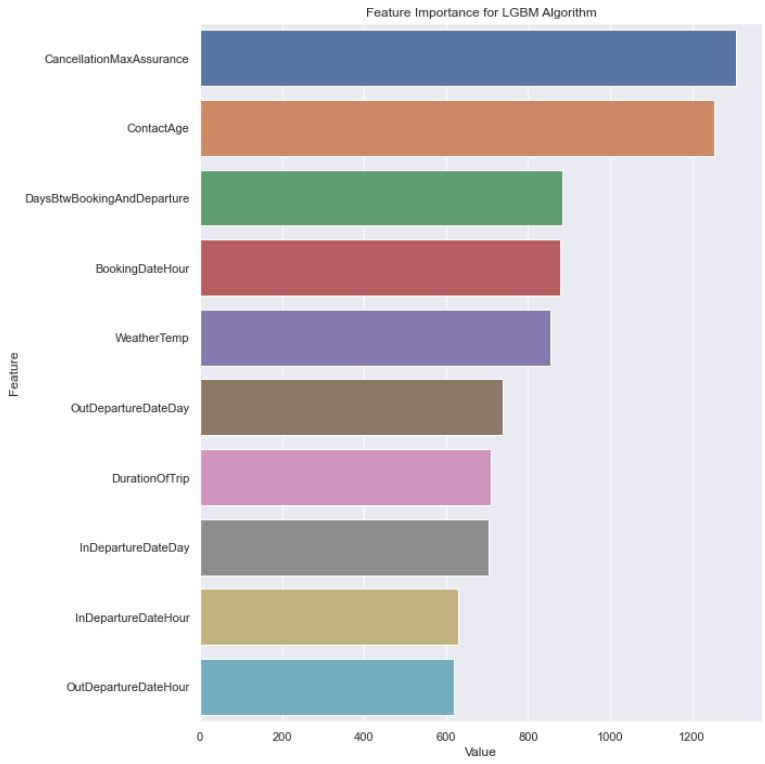

Figure 5.36 Feature Importance Plot of LGBM Algortihm of Round Trip Flight Data

5.2. Results and Discussion

As a result, in this study, two different models were trained to predict passengers' ticket cancellation behavior on a dataset obtained from Turna.com, which includes flight information, passenger details, and environmental factors. The first model was trained on the dataset of one-way flight tickets, while the second model was trained on the dataset of round-trip tickets. Various machine learning models were used, and they were compared using different validation techniques. When comparing the models, the "precision" score was considered since the trained machine learning models were designed to accurately predict passengers' ticket cancellation behavior. The precision score indicates the percentage of users predicted to exhibit ticket cancellation behavior who actually canceled their tickets. This percentage is crucial for companies to consider in order to maximize profits. Therefore, the models were primarily evaluated based on the precision score.

Multiple machine learning algorithms were used in building the models, and a comparison was made among them. When looking at the main model built on one-way tickets, the highest precision score of 0.6917 was achieved by the Random Forest (RF) algorithm, followed by the Light Gradient Boosting Machine (LGBM) algorithm with a precision score of 0.5755. After performing hyperparameter optimization on these models, the results were improved, leading to better outcomes. Upon examining these results, it can be observed that in the optimized model, the best precision score of 0.75 is obtained by the Light Gradient Boosting Machine (LGBM) algorithm, followed by the Random Forest (RF) algorithm with a precision score of 0.7333.

For the main models trained on round-trip tickets, it is evident that the Random Forest (RF) algorithm achieves the highest precision score of 0.4833. It is followed by the Light Gradient Boosting Machine (LGBM) algorithm with a precision score of 0.4284. When performing hyperparameter optimization on these models, it is observed that the best precision score of 0.6642 belongs to the Light Gradient Boosting Machine (LGBM) algorithm, and the second-best precision score of 0.6116 belongs to the XGBoosting algorithm.

The precision scores of the models trained on one-way flight tickets are promising and can accurately predict passengers' ticket cancellation behavior. However, the same cannot be said for the models trained on round-trip tickets. The main reason for this is the

difference in dataset size after splitting the main data. There is more data available for one-way tickets**.**

Furthermore, feature importance analyses is held the interpret the results of the machine learning models better. The top 10 variables that have the most influence on the trained machine learning models are crucial in understanding the predictions and performance of the model. Since the tuned LGBM model has the highest precision score (0.75) on test set top 10 important features of this model are analyzed:

- CancellationMaxAssurance emerges as a significant factor, representing the amount promised to be covered under the unconditional ticket cancellation service. This variable indicates the level of assurance provided to customers, as it covers a substantial portion of the ticket fare.
- ContactAge, the age of the contact person at the time of ticketing, plays a role in determining customer behavior and preferences. Age can influence decision-making, travel choices, and other factors that impact ticket bookings.
- DaysBtwBookingAndDeparture is another important variable, capturing the duration between ticketing and the departure date. This variable provides insight into the booking habits of customers and their propensity for advance planning or last-minute bookings.
- WeatherTemp, representing the weather temperature at the departure location on the outbound flight's departure date, indicates the potential impact of weather conditions on travel decisions. Weather can influence travel patterns and affect customers' likelihood of booking or canceling flights.
- BookingDateHour and BookingDateDay reflect the time and date when the booking was made. These variables capture temporal patterns and potential correlations with customer behavior, allowing for insights into peak booking hours or specific days that drive higher booking volumes.
- OutDepartureDateHour and OutDepartureDateDay represent the hour and day of the departure date for the outbound flight, respectively. These variables provide information on departure timing patterns, such as peak travel hours or specific days that are more popular for travel.
- Finally, the UsdRate variable, denoting the exchange rate on the transaction date, is relevant for international travel. Fluctuations in exchange rates can impact ticket

prices and customer decisions, making this variable significant for predicting outcomes accurately.

Understanding the importance of these variables sheds light on the underlying factors that influence the machine learning model's predictions. By considering these variables and their associated meanings, valuable insights can be gain about customer behavior, temporal patterns, weather-related impacts, and other factors crucial for effective decision-making and model performance.

6. CONCLUSION

This study on the development of an Intelligent Flight Ticket Cancellation Recommendation Engine using Machine Learning provides valuable insights into the aviation industry and addresses the challenges faced by providers of cancellation insurance services. As known, the aviation industry has faced significant challenges, especially during the COVID-19 pandemic, which has led to financial losses for airlines and travel agencies. To mitigate these challenges, cancellation insurance services have been offered as a solution for airline passengers. However, balancing customer satisfaction, profitability, and environmental impact has posed challenges for providers of cancellation insurance services.

Also, this study recognizes the importance of sustainability in the aviation industry and advocates for providing insurance for canceled tickets to reduce last-minute cancellations and associated emissions. It discusses the three main elements of aviation sustainability, namely environmental sustainability, economic sustainability, and social suitability. By developing systems that protect the environment, enhance economic value, and raise the standard of living, the aviation industry can become more sustainable.

In this direction, literature review held in this study examines airline cancellations and explores solutions to the issue. It emphasizes the importance of balancing customer satisfaction and service provider profitability. Potential solutions include optimizing cancellation protection service (CPS) policies through customer segmentation and quality of experience (QoE) approaches. The study highlights the significant impact of ticket refund services on customer trust. It also discusses the use of time series modeling, machine learning, and deep learning for predicting cancellations. The review emphasizes the need for a better understanding of cancellation patterns and their relationship with delays. Overall, the findings provide valuable insights into improving cancellation policies and enhancing customer experience.

Furthermore, the study explores the application of machine learning techniques to gain insights into customer behavior and enhance pricing policies for cancellation insurance services by utilizing a dataset containing information on domestic and international flight transactions of an online travel agency, Turna.com, between June 1, 2022, and July 1, 2022. In this paper various strategies is used in the application section

such as handling of variable types, exclusion of irrelevant variables, transformation of date and time information, analysis of different flight types, filling missing values and handling outliers, encoding of categorical variables, building machine learning models, hyperparameter optimization and lastly feature importance analysis. Two seperate datasets for One Way and Round Trip flight types created and machine learning models are builded for predicting the likelihood of passenger cancellations after ticketing. In this study various machine learning algorithms such as Random Forest, Gradient Boosting Machine, XGBoosting, and Light Gradient Boosting Machine are used to make this predictions.

As a result, the findings of this study provide valuable insights into customer behavior and pricing policies for cancellation insurance services, leading to improved customer satisfaction while maintaining profitability in the airline passenger transportation industry. Additionally, the study highlights the role of machine learning in facilitating better decision-making within the aviation industry. The efficiency of machine learning in analyzing large volumes of data enables accurate predictions, aiding decision-making processes.

6.1. Limitations of Study and Recommendation for Further Directions

While this study contributes valuable insights, it is important to acknowledge its limitations. First, the study focused on the development of an Intelligent Flight Ticket Cancellation Recommendation Engine using Machine Learning and with the division of the main dataset into one-way and round-trip tickets, two different machine learning models were trained using two separate datasets. When comparing the model trained on the one-way dataset with the model trained on the round-trip dataset, it can be observed that the validation scores of the first model are better. The main reason for this is the size of the dataset. In future studies, this issue can be addressed by using more data for training the machine learning model for round-trip tickets. Additionally, increasing the time range covered by the main dataset can lead to improvements in the performance of both models. The data used in the study was obtained from one company, but in future studies, more realistic model results can be obtained by blending and using data from other sources. The dataset used includes a wide range of data, most of which have provided valuable information for the study. Moreover, having a diverse range of data for training the machine learning models has resulted in more reliable outcomes

While this study has provided valuable insights into the development of an Intelligent Flight Ticket Cancellation Recommendation Engine using Machine Learning, there are several areas that can be further explored to enhance the analysis of user behavior and provide more effective and sustainable solutions in the aviation industry. Future studies should aim to expand the dataset used in this research in terms of both size and diversity. By incorporating data from multiple companies and diverse data sources, more realistic and comprehensive model results can be obtained. Gathering additional information on user behavior and preferences can lead to better analyses and personalized recommendations.To make the developed recommendation engine more practical and beneficial for the company, it is crucial to integrate real-time data. This requires data engineering efforts to establish a seamless connection with the airline systems and continuously update the recommendation engine with the latest flight and customer information. By incorporating real-time data, the model can adapt to dynamic changes in flight availability, pricing, and customer preferences, resulting in more accurate and relevant recommendations.In-depth analysis of user behavior can provide valuable insights into cancellation patterns, customer preferences, and decision-making processes. Future studies should explore additional types of data, such as social media sentiment analysis, customer reviews, and demographic information, to gain a better understanding of the factors influencing ticket cancellations. By incorporating such data, the recommendation engine can be refined to cater to individual preferences and offer personalized cancellation protection services.While the machine learning models developed in this study have shown promising results, there is room for exploring more advanced techniques. Researchers can investigate the application of deep learning algorithms, natural language processing, or reinforcement learning to improve the accuracy and performance of the recommendation engine. These advanced techniques can handle complex patterns and dependencies within the data, leading to more sophisticated and accurate predictions. To ensure the practicality and widespread adoption of the Intelligent Flight Ticket Cancellation Recommendation Engine, collaboration with airlines, insurance providers, and industry stakeholders is essential. By collaborating with industry partners, researchers can gain access to proprietary data, validate the model's performance in real-world scenarios, and refine the recommendation engine based on industry-specific requirements and feedback. Although the results obtained in this study are promising and significant, addressing these limitations and further increasing the options and diversity of the main data will enable better analysis of user behavior and provide more effective and sustainable solutions to the problems encountered in the aviation industry.

To summarize, this study offers valuable insights into the aviation industry and the challenges faced by cancellation insurance service providers through the development of an Intelligent Flight Ticket Cancellation Recommendation Engine using Machine Learning. It emphasizes the significance of sustainability in aviation, advocating for insurance coverage to reduce last-minute cancellations and associated emissions. Balancing customer satisfaction and profitability is identified as a key challenge, with potential solutions discussed, such as optimizing cancellation protection service policies through customer segmentation and quality of experience approaches. The application of machine learning techniques proves beneficial in understanding customer behavior and improving pricing policies for cancellation insurance services. The study acknowledges its limitations and suggests future research directions to address them, including expanding the dataset, integrating real-time data, and further analyzing user behavior for more effective and sustainable solutions in the aviation industry.

REFERENCES

Hossin, M., & Sulaiman, M. (2015). A Review on Evaluation Metrics for Data. International Journal of Data Mining & Knowledge Management Process.

Hoyle, B., Rau, M., Zitlau, R., Seitz, S., & Weller, J. (2015). Feature importance for machine learning redshifts applied to SDSS galaxies. Monthly Notices of the Royal Astronomical Society, s. 1275-1283.

Panda, P., & Majhi, B. (2018). A survey on encoding techniques for categorical data in machine learning. International Journal of Computer Applications, s. 179 - 213.

Raschka, S., & Mirjalili, V. (2019). Python Machine Learning .

Zhang , J., Wang , Y., & Liu, Y. (2012). New Machine Learning Algorithm: Random Forest. Information Computing and Applications, s. 246-252.

Sadreddini, Z., Donmez, I., & Yanikomeroglu, H. (2021). Cancel-for-Any-Reason Insurance Recommendation Using Customer Transaction-Based Clustering. IEEE Access, 9, 39363-39374.

Sadreddini, Z. (2020). A novel cancellation protection service in online reservation system. IEEE Access, 8, 129094-129107.

Chiew, E., Daziano, R. A., & Garrow, L. A. (2017). Bayesian estimation of hazard models of airline passengers' cancellation behavior. Transportation Research Part A: Policy and Practice, 96, 154-167.

Cirillo, C., Bastin, F., & Hetrakul, P. (2018). Dynamic discrete choice model for railway ticket cancellation and exchange decisions. Transportation Research Part E: Logistics and Transportation Review, 110, 137-146.

Keleş, M. B., Keleş, A., & Keleş, A. (2020). Yapay zekâ teknolojisi ile uçuş fiyatı tahmin modeli geliştirme. Turkish Studies, 15(4), 511-520.

Maulana, S. A., Gigantama, M. R., Lesmini, L., Ozali, I., & Budiman, C. (2019). The Influence of Ticket Refund Service Towards Air Asia Customers Trust. Advances in Transportation and Logistics Research, 2, 150-154.

Ozdemir S. (2015), Value Driven Decision Forecast Model: An Application on Online Flight Ticket Purchase

Iliescu, Dan C.(2008), Customer Based Time-to-Event Models for Cancellation Behavior

Naboush, E. (2019). Cancellation of Flights-Complicated Issues for Passengers. Journal Sharia and Law, 2019(78), 10.

Geraldi, J., Lechter, T. (2012). Gantt charts revisited:A critical analysis of its roots and implications to the management of projects today: International Journal of Managing Projects in Business, 5, 578-594.

Lantseva, A., Mukhina, K., Nikishova, A., Ivanov, S., & Knyazkov, K. (2015). Data-driven modeling of airlines pricing. Procedia Computer Science, 66, 267-276.

Abdella, J. A., Zaki, N. M., Shuaib, K., & Khan, F. (2021). Airline ticket price and demand prediction: A survey. Journal of King Saud University-Computer and Information Sciences, 33(4), 375-391.

Dube, K., Nhamo, G., & Chikodzi, D. (2021). COVID-19 pandemic and prospects for recovery of the global aviation industry. Journal of Air Transport Management, 92, 102022.

Malgorzata Z., Eljas J. (2022) Sustainability reporting in the airline industry: Current literature and future research avenues

O'Connell, J.F., (2018). The global airline industry. In: Helpern, N., Graham, A. (Eds.), The Routledge Companion to Air Transport Management. Routledge,

Y. Kim, J. Lee, J. (2019) Ahn Technological Forecasting & Social Change Innovation towards sustainable technologies: A socio-technical perspective on accelerating transition to aviation biofuel Technol. Forecast. Soc. Chang., 145

PwC, (2011). Building trust in the air: Is airline corporate sustainability reporting taking off?

IATA, 2020. Industry Statistics - Fact Sheet November 2020

ATAG, 2020 ATAG, 2020. Fact & Figures

H. Paramesh1 (April 2018) Air Pollution and Allergic Airway Diseases: Social Determinants and Sustainability in the Control and Prevention

WCED (1987), Our Common Future, World Commission on Environment and Development, World Commission on Environment and Development, Oxford University Press, New York, NY.

Ryan, S. and Throgmorton, J.A. (2003), “Sustainable transportation and land development on the periphery: a case study of Freiburg, Germany and Chula Vista, California”, Transportation Research Part D: Transport and Environment, Vol. 8 No. 1, pp. 37-52.

Gamze O. (2021) The effects of airline strategies on environmental sustainability Volume 93 · Number 8 1346–1357

Daley, B. (2010), Air Transport and the Environment, Ashgate, England.

Forsyth, P. (2011), “Environmental and financial sustainability of air transport: are they incompatible?”, Journal of Air Transport Management, Vol. 17 No. 1, pp. 27-32.

Budd, L., Griggs, S. and Howarth, D. (2013), Sustainable Aviation Futures: Crises, Contested Realities and Prospects for Change, Sustainable Aviation Futures, Emerald Group Publishing Limited, UK.

Graham, B. and Guyer, C. (1999), “Environmental sustainability, airport capacity and European air transport liberalization: irreconcilable goals?”, Journal of Transport Geography, Vol. 7 No. 3, pp. 165-180.

Walters, N.W., Rice, S., Winter, S.R., Baugh, B.S., Ragbir, N. K., Anania, E.C., Capps, J. and Milner, M.N. (2018), “Consumer willingness to pay for new airports that use renewable resources”, International Journal of Sustainable Aviation, Vol. 4 No. 2, pp. 79-98.

Alameeri, A., Ajmal, M. M., Hussain, M., and Helo, P. T., Sustainability practices in the aviation sector: a study of UAE-based airlines, International Journal of Sustainable Society, Vol. 9, no. 2, pp. 119-147, 2017.

Eman R. Elhmoud, Sustainability Assessment in Aviation Industry: A Mini- Review on the Tools, Models and Methods of Assessment Proceedings of the 2nd African International Conference on Industrial Engineering and Operations Management Harare, Zimbabwe, December 7-10, 2020

(Chen & Guestrin, 2016).

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. Ann. Stat. 28, 337–407. doi: 10.1214/aos/1016218222

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In Advances in neural information processing systems (pp. 2951-2959).

Li, L., & Malik, J. (2017). Learning to optimize. International Conference on Learning Representations (ICLR).

Pedregosa, F., et al. (2011). "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research, 12(Oct), 2825-2830.

Müller, A. C., & Guido, S. (2017). Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media.

Helm, J., Swiergosz, A., & Haeberle, H. (2020, 1). Machine Learning and Artificial Intelligence: Definitions, Applications, and Future Directions. Machine Learning and Artificial Intelligence: Definitions, Applications, and Future Directions, s. 69-76.

Hossin, M., & Sulaiman, M. (2015, 3). A REVIEW ON EVALUATION METRICS FOR DATA. International Journal of Data Mining & Knowledge Management Process.

APPENDIX

Python Codes for Machine Learning Solution

```
import numpy as np

import pandas as pd

import seaborn as sns

from matplotlib import pyplot as plt

import warnings

from sklearn.exceptions import ConvergenceWarning

from datetime import date

from datetime import datetime

from sklearn.preprocessing import RobustScaler

from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier, VotingClassifier

from xgboost import XGBClassifier

from lightgbm import LGBMClassifier

from sklearn.model_selection import GridSearchCV,cross_validate,StratifiedKFold,train_test_split

from imblearn.over_sampling import SMOTE

import graphviz

pd.set_option('display.max_columns', None)

pd.set_option('display.max_rows', None)

pd.set_option('display.float_format', lambda x: '%.4f' % x)

pd.set_option('display.width', 500)

def is_weekend(date):
```

```
    weekend_dates = pd.date_range(start=date - pd.Timedelta(days=2), end=date + pd.Timedelta(days=2), freq="D")
    return 1 if any(date.weekday() >= 5 for date in weekend_dates) else 0
def grab_col_names(dataframe, cat_th=10, car_th=20):
    cat_cols = [col for col in dataframe.columns if dataframe[col].dtypes == "O"]
    num_but_cat = [col for col in dataframe.columns if dataframe[col].nunique() < cat_th and
                   dataframe[col].dtypes != "O"]
    cat_but_car = [col for col in dataframe.columns if dataframe[col].nunique() > car_th and
                   dataframe[col].dtypes == "O"]
    cat_cols = cat_cols + num_but_cat
    cat_cols = [col for col in cat_cols if col not in cat_but_car]
    num_cols = [col for col in dataframe.columns if dataframe[col].dtypes != "O"]
    num_cols = [col for col in num_cols if col not in num_but_cat]
    return cat_cols, cat_but_car, num_cols
def check_df(dataframe, head=5):
    print("##################### Shape #####################")
    print(dataframe.shape)
    print("##################### Types #####################")
    print(dataframe.dtypes)
    print("##################### Head #####################")
    print(dataframe.head(head))
    print("##################### Tail #####################")
    print(dataframe.tail(head))
    print("##################### NA #####################")
    print(dataframe.isnull().sum())
    print("##################### Quantiles #####################")
```

```
    print(dataframe.quantile([0, 0.05, 0.50, 0.95, 0.99, 1]).T)
def cat_summary(dataframe, col_name, plot=False):
    print(pd.DataFrame({col_name: dataframe[col_name].value_counts(),
                        "Ratio": 100 * dataframe[col_name].value_counts() / len(dataframe)}))
    print("##########################################")
    if plot:
        sns.countplot(x=dataframe[col_name], data=dataframe)
        plt.show(block=True)
def num_summary(dataframe, numerical_col, plot=False):
    quantiles = [0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 0.95, 0.99]
    print(dataframe[numerical_col].describe(quantiles).T)
    if plot:
        dataframe[numerical_col].hist(bins=20)
        plt.xlabel(numerical_col)
        plt.title(numerical_col)
        plt.show(block=True)
def target_summary_with_num(dataframe, target, numerical_col):
    print(dataframe.groupby(target).agg({numerical_col: "mean"}), end="\n\n\n")
def target_summary_with_cat(dataframe, target, categorical_col):
    print(pd.DataFrame({"TARGET_MEAN":
dataframe.groupby(categorical_col)[target].mean()}), end="\n\n\n")
def correlation_matrix(df, cols,save =False):
    fig = plt.gcf()
    fig.set_size_inches(25, 25)
    plt.title("Correlation Matrix of Numerical Variables")
    plt.xticks(fontsize=25)
```

```
    plt.yticks(fontsize=25)
    fig = sns.heatmap(df[cols].corr(), annot=True, linewidths=4, annot_kws={'size': 20}, linecolor='w', cmap='RdBu')
    plt.show(block=True)
    if save:
        plt.savefig('importances.png')


def high_correlated_cols(dataframe, plot=False, corr_th=0.90):
    corr = dataframe.corr()
    cor_matrix = corr.abs()
    upper_triangle_matrix = cor_matrix.where(np.triu(np.ones(cor_matrix.shape), k=1).astype(bool))
    drop_list = [col for col in upper_triangle_matrix.columns if any(upper_triangle_matrix[col] > corr_th)]
    if plot:
        import seaborn as sns
        import matplotlib.pyplot as plt
        sns.set(rc={'figure.figsize': (15, 15)})
        sns.heatmap(corr, cmap="RdBu")
        plt.show()
    return drop_list
def missing_values_table(dataframe):
    na_columns = [col for col in dataframe.columns if dataframe[col].isnull().sum() > 0]
    n_miss = dataframe[na_columns].isnull().sum().sort_values(ascending=False)
    ratio = (dataframe[na_columns].isnull().sum() / dataframe.shape[0] * 100).sort_values(ascending=False)
    missing_df = pd.concat([n_miss, np.round(ratio, 2)], axis=1, keys=['n_miss', 'ratio'])
```

```
    print(missing_df, end="\n")
    return na_columns, missing_df
def missing_zero_values_table(df):
    zero_val = (df == 0.00).astype(int).sum(axis=0)
    mis_val = df.isnull().sum()
    mis_val_percent = 100 * df.isnull().sum() / len(df)
    mz_table = pd.concat([zero_val, mis_val, mis_val_percent], axis=1)
    mz_table = mz_table.rename(
    columns = {0 : 'Zero Values', 1 : 'Missing Values', 2 : '% of Total Values'})
    mz_table['Total Zero Missing Values'] = mz_table['Zero Values'] + mz_table['Missing Values']
    mz_table['% Total Zero Missing Values'] = 100 * mz_table['Total Zero Missing Values'] / len(df)
    mz_table['Data Type'] = df.dtypes
    mz_table = mz_table[mz_table.iloc[:,1] != 0].sort_values('% of Total Values', ascending=False).round(1)
    print("Your selected dataframe has " + str(df.shape[1]) + " columns and " + str(df.shape[0]) + " Rows.\n"
        "There are " + str(mz_table.shape[0]) +
         " columns that have missing values.")
    return mz_table
def outlier_thresholds(dataframe, col_name, q1=0.01, q3=0.99):
    quartile1 = dataframe[col_name].quantile(q1)
    quartile3 = dataframe[col_name].quantile(q3)
    interquantile_range = quartile3 - quartile1
    up_limit = quartile3 + 1.5 * interquantile_range
    low_limit = quartile1 - 1.5 * interquantile_range
```

```
    return low_limit, up_limit
def replace_with_thresholds(dataframe, variable):
    low_limit, up_limit = outlier_thresholds(dataframe, variable)
    dataframe.loc[(dataframe[variable] < low_limit), variable] = low_limit
    dataframe.loc[(dataframe[variable] > up_limit), variable] = up_limit
def check_outlier(dataframe, col_name, q1=0.01, q3=0.99):
    low_limit, up_limit = outlier_thresholds(dataframe, col_name, q1, q3)
    if dataframe[(dataframe[col_name] > up_limit) | (dataframe[col_name] < low_limit)].any(axis=None):
        return True
    else:
        return False
def rare_encoder(dataframe, rare_perc):
    temp_df = dataframe.copy()
    rare_columns = [col for col in temp_df.columns if temp_df[col].dtypes == 'O'
                    and (temp_df[col].value_counts() / len(temp_df) < rare_perc).any(axis=None)]
    for var in rare_columns:
        tmp = temp_df[var].value_counts() / len(temp_df)
        rare_labels = tmp[tmp < rare_perc].index
        temp_df[var] = np.where(temp_df[var].isin(rare_labels), 'Rare', temp_df[var])
    return temp_df
def rare_analyser(dataframe, target, cat_cols):
    for col in cat_cols:
        print(col, ":", len(dataframe[col].value_counts()))
        print(pd.DataFrame({"COUNT": dataframe[col].value_counts(),
                            "RATIO": dataframe[col].value_counts() / len(dataframe),
```

```
                "TARGET_MEAN": dataframe.groupby(col)[target].mean()}), end="\n\n\n")

def one_hot_encoder(dataframe, categorical_cols, drop_first=False):
    dataframe = pd.get_dummies(dataframe, columns=categorical_cols, drop_first=drop_first)
    return dataframe

def plot_importance(model, features, num=10, save=False,col_name = "X Algorithm"):
    feature_imp = pd.DataFrame({'Value': model.feature_importances_, 'Feature': features.columns})
    plt.figure(figsize=(10, 10))
    sns.set(font_scale=1)
    sns.barplot(x="Value", y="Feature", data=feature_imp.sort_values(by="Value",
                                                     ascending=False)[0:num])
    plt.title(str(col_name))
    plt.tight_layout()
    plt.show()
    if save:
        plt.savefig('importances.png')

df_ = pd.read_excel("IE4198_2023_EKOS_ML.xlsx",sheet_name="data")

Airway_Passenger_Who_Have_Not_Requested_Cancellation = df.Target.value_counts()[0]

Airway_Passenger_Requesting_Cancellation = df.Target.value_counts()[1]

colors = ['#FFA500', '#FF0000']

data = [Airway_Passenger_Who_Have_Not_Requested_Cancellation, Airway_Passenger_Requesting_Cancellation]

labels = ['Airway Passenger Who Have Not Requested Cancellation', 'Airway Passenger Requesting Cancellation']

title = 'Distribution of Target Variable'
```

```
plt.figure(figsize=(8, 6))
plt.pie(data, labels=labels, colors=colors, autopct='%1.1f%%')
plt.title(title)
plt.show(block = True)
cat_cols, cat_but_car, num_cols = grab_col_names(df)
{
"Total Number of Variables": len(cat_cols+num_cols+cat_but_car),
"Categorical Variables": len(cat_cols),
"Numerical Variable ": len(num_cols),
"Cardinal Variable": len(cat_but_car)
}
x_values = ['Categorical Variables', 'Numerical Variables', 'Cardinal Variables']
y_values = [len(cat_cols), len(num_cols), len(cat_but_car)]
sns.set(style="whitegrid")
plt.figure(figsize=(8, 6))
sns.barplot(x=x_values, y=y_values, palette="Blues_d");
plt.xlabel('Variable Types')
plt.ylabel('Number of Variable')
plt.title('Countplot of Variable Types')
meaningless_variables = ['AffiliateId',
'BasketId',
 'BookingDateWeek',
 'BookingDateWeekdayNo',
 'CancellationExpireDate',
 'Channel',
 'Classes',
```

'ContactBirthYear',

'CustomerCurrency',

'DaysBtwBookingAndRefund',

'DaysBtwRefundAndDeparture',

'DomInt',

'DurationOfStay',

'EntryDateWeek',

'EntryDateWeekdayNo',

'FlightType',

'InDepartureDateHour',

'InDepartureDateWeek',

'InDepartureDateWeekdayNo',

'InDestCity',

'InOrigCity',

'IsBooked',

'IsHoliday',

'IsMemberAndContactSame',

'IsRefunded',

'IsRefundedWCancelAssurance',

'MarketingAirlines',

'MemberAge',

'MemberBirthYear',

'MemberGender',

'MemberId',

'NumberOfOperatingAirlines',

'OperatingAirlines',

'OutDepartureDateHour',

'OutDepartureDateWeekdayNo',

'OutDestCity',

'OutOrigCity',

'PassengerCountAgeBtw18And24',

'PassengerCountAgeBtw25And34',

'PassengerCountAgeBtw35And49',

'PassengerCountAgeOver50',

'PassengerCountAgeUnder18',

'PassengerCountFemaleADT',

'PassengerCountFemaleCHD',

'PassengerCountFemaleINF',

'PassengerCountFemaleSRC',

'PassengerCountFemaleSTD',

'PassengerCountMaleADT',

'PassengerCountMaleCHD',

'PassengerCountMaleINF',

'PassengerCountMaleSRC',

'PassengerCountMaleSTD',

'RefundAddOn',

'RefundCancellationAssurance',

'RefundCancellationFee',

'RefundDate',

'RefundDiscount',

'RefundLP',

'RefundNetFare',

```
 'RefundSC',
 'RefundTotalFare',
 'SaleAddOn',
 'SaleCancellationFee',
 'SaleDiscount',
 'SaleLP',
 'SaleNetFare',
 'SaleSC',
 'SessionId',
 'TripType',
 'UserId']
df.drop(meaningless_variables,axis=1,inplace =True)
   df['InArrivalDate'] = pd.to_datetime(df['InArrivalDate'], format='%Y-%m-%d %H:%M:%S', errors='coerce')
   df[str("InArrivalDate") + 'Year'] = df["InArrivalDate"].dt.year
   df[str("InArrivalDate") + 'Day'] = df["InArrivalDate"].dt.day
   df[str("InArrivalDate") + 'Month'] = df["InArrivalDate"].dt.month
   df[str("InArrivalDate") + 'Weekday'] = df["InArrivalDate"].dt.day_name()
   df[str("InArrivalDate") + 'Week'] = df["InArrivalDate"].dt.isocalendar().week
   df["InArrivalTime"] = pd.to_datetime(df["InArrivalTime"], format='%H:%M:%S', errors='coerce')
   df["InArrivalDateHour"] = df["InArrivalTime"].dt.hour
   df['OutArrivalDate'] = pd.to_datetime(df['OutArrivalDate'], format='%Y-%m-%d %H:%M:%S', errors='coerce')
   df[str('OutArrivalDate') + 'Year'] = df['OutArrivalDate'].dt.year
   df[str('OutArrivalDate') + 'Day'] = df['OutArrivalDate'].dt.day
   df[str('OutArrivalDate') + 'Month'] = df['OutArrivalDate'].dt.month
```

```
    df[str('OutArrivalDate') + 'Weekday'] = df['OutArrivalDate'].dt.day_name()
    df[str('OutArrivalDate') + 'Week'] = df['OutArrivalDate'].dt.isocalendar().week
    df["OutArrivalTime"] = pd.to_datetime(df["OutArrivalTime"], format='%H:%M:%S', errors='coerce')
    df["OutArrivalDateHour"] = df["OutArrivalTime"].dt.hour


    df["InDepartureTime"] = pd.to_datetime(df["InDepartureTime"], format='%H:%M:%S', errors='coerce')
    df["InDepartureDateHour"] = df["InDepartureTime"].dt.hour


    df["OutDepartureTime"] = pd.to_datetime(df["OutDepartureTime"], format='%H:%M:%S', errors='coerce')
    df["OutDepartureDateHour"] = df["OutDepartureTime"].dt.hour
    meaningless_variables = ["BookingDate","EntryDate","InDepartureDate","OutDepartureDate","InArrivalDate","OutArrivalDate",
    "InArrivalTime","OutArrivalTime","InDepartureTime","OutDepartureTime"]
    df.drop(meaningless_variables,axis=1,inplace=True)
df["InArrivalDateWeek"] = df["InArrivalDateWeek"].dropna().astype("int64")
df["OutArrivalDateWeek"] = df["OutArrivalDateWeek"].dropna().astype("int64")
df["DaysBtwBookingAndDeparture"] = abs(df["DaysBtwBookingAndDeparture"])
for col in num_cols:
    plt.figure(figsize=(8, 6))
    plt.hist(df[col], bins=10, color='steelblue', edgecolor='black')
    plt.title("Histogram" + " of " + str(col))
    plt.xlabel('Values')
    plt.ylabel('Frequence')
```

```
    plt.show(block=True)
for col in cat_cols:
    print((pd.DataFrame({str(col): df[col].value_counts(),"Ratio": 100 *
df[col].value_counts() / len(df)})))
df.LanguageCode = df.LanguageCode.str.upper()
for col in cat_cols:
    plt.figure(figsize=(8, 6))
    sns.set(style="darkgrid")
    sns.countplot(x=col, data=df)
    plt.title("Count Plot" + " of " + str(col))
    plt.xlabel('Values')
    plt.ylabel('Frequence')
    plt.show(block=True)
ow = df.loc[df["TripTypeName"] == "One Way"]
meaningless_variables = ["InDepartureDateHour",
    "InArrivalDateHour",
    "InArrivalDateWeekday",
    "InArrivalDateMonth",
    "InArrivalDateDay",
    "InArrivalDateYear",
    "InDepartureDateWeekday",
    "InArrivalDateWeek",
    "InOrigCountry",
    "InOrigContinent",
    "InOrigCityName",
    "InOrigAirport",
```

```
    "InDestCountry",
    "InDestContinent",
    "InDestCityName",
    "InDestAirport",
    "TripTypeName",
    "DurationOfTrip"]
ow.drop(meaningless_variables,axis=1,inplace=True)
correlation_matrix(ow, num_cols,True)
missing_zero_values_table(ow)
na_columns, missing_df = missing_values_table(ow)
ow['WeatherTemp'].fillna(ow['WeatherTemp'].mean(),inplace=True)
ow['SaleCountryCode'].fillna(ow['SaleCountryCode'].mode()[0],inplace=True)
ow['OutDestContinent'].fillna("US",inplace=True)
ow['OutOrigContinent'].fillna("US",inplace=True)
ow['ContactNationality'].fillna(ow['ContactNationality'].mode()[0],inplace=True)
for col in num_cols:
    print(str(col),' ---->',check_outlier(ow, col, q1=0.05, q3=0.95))
for col in ["CancellationMaxAssurance","SaleTotalFare"]:
    replace_with_thresholds(ow,col)
for col in num_cols:
    plt.figure(figsize=(8, 6))
    sns.boxplot(y=ow[col])
    plt.title("Box Plot" + " of " + str(col))
    plt.xlabel('Values')
    plt.ylabel('Frequence')
    plt.show(block=True)
```

```
meaningless_variables = high_correlated_cols(ow, plot=False, corr_th=0.90)
meaningless_variables
rare_analyser(ow, "Target", cat_cols)
ow = rare_encoder(ow,0.05)
meaningless_variables = [col for col in ow.columns if len(ow[col].unique()) == 1]
meaningless_variables
ow.drop(meaningless_variables,axis=1,inplace=True)
cat_cols, cat_but_car, num_cols = grab_col_names(ow)
cat_cols = [col for col in cat_cols if str(col) not in "Target"]
ow = one_hot_encoder(ow, cat_cols, drop_first=True)
y = ow["Target"]
X = ow.drop(["Target"], axis=1)
rf_model = RandomForestClassifier(random_state=17).fit(X,y)
gbm_model = GradientBoostingClassifier(random_state=17).fit(X,y)
xgboost_model = XGBClassifier(random_state=17, use_label_encoder=False).fit(X,y)
lgbm_model = LGBMClassifier(random_state=17).fit(X,y)
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
scoring = ['accuracy', 'precision',"roc_auc"]
cv_results_rf = cross_validate(rf_model, X, y, scoring=scoring, cv=skf)
cv_results_gbm = cross_validate(gbm_model, X, y, scoring=scoring, cv=skf)
cv_results_xgboost = cross_validate(xgboost_model, X, y, scoring=scoring, cv=skf)
cv_results_lgbm = cross_validate(lgbm_model, X, y, scoring=scoring, cv=skf)
comparison_of_metrics_dic = {'Accuracy Score':
[cv_results_rf['test_accuracy'].mean(),cv_results_gbm['test_accuracy'].mean(),cv_results_x
gboost['test_accuracy'].mean(),cv_results_lgbm['test_accuracy'].mean()],
```

```
'Precision Score':
[cv_results_rf['test_precision'].mean(),cv_results_gbm['test_precision'].mean(),cv_results_
xgboost['test_precision'].mean(),cv_results_lgbm['test_precision'].mean()],

'ROC-AUC Score':
[cv_results_rf['test_roc_auc'].mean(),cv_results_gbm['test_roc_auc'].mean(),cv_results_xg
boost['test_roc_auc'].mean(),cv_results_lgbm['test_roc_auc'].mean()]}

comparison_of_metrics_df = pd.DataFrame(comparison_of_metrics_dic, index=['RF
Algorithm', 'GBM Algorithm', 'XGBOOST Algorithm',"LGBM Algorithm"])

comparison_of_metrics_df

rf_params = {"max_depth": [8, 15, None],

        "max_features": [5, 7, "auto"],

        "min_samples_split": [15, 20],

        "n_estimators": [200, 300]}

xgboost_params = {"learning_rate": [0.1, 0.01],

            "max_depth": [5, 8],

            "n_estimators": [100, 200]}

lightgbm_params = {"learning_rate": [0.01, 0.1],

             "n_estimators": [300, 500]}

gbm_params = {"learning_rate": [0.01, 0.1],

        "max_depth": [3, 8, 10],

        "n_estimators": [100, 500, 1000],

        "subsample": [1, 0.5, 0.7]}

rf_best_grid = GridSearchCV(rf_model, rf_params, cv=5, n_jobs=-1, verbose=True).fit(X,
y)

rf_final = rf_model.set_params(**rf_best_grid.best_params_, random_state=17).fit(X, y)

xgboost_best_grid = GridSearchCV(xgboost_model, xgboost_params, cv=5, n_jobs=-1,
verbose=True).fit(X, y)
```

```
xgboost_final = xgboost_model.set_params(**xgboost_best_grid.best_params_, random_state=17).fit(X, y)

lgbm_best_grid = GridSearchCV(lgbm_model, lightgbm_params, cv=5, n_jobs=-1, verbose=True).fit(X, y)

lgbm_final = lgbm_model.set_params(**lgbm_best_grid.best_params_, random_state=17).fit(X, y)

gbm_best_grid = GridSearchCV(gbm_model, gbm_params, cv=5, n_jobs=-1, verbose=True).fit(X, y)

gbm_final = gbm_model.set_params(**gbm_best_grid.best_params_, random_state=17).fit(X, y)

scoring = ['accuracy', 'precision',"roc_auc"]

cv_results_rf = cross_validate(rf_final, X, y, scoring=scoring, cv=skf)

cv_results_gbm = cross_validate(gbm_final, X, y, scoring=scoring, cv=skf)

cv_results_xgboost = cross_validate(xgboost_final, X, y, scoring=scoring, cv=skf)

cv_results_lgbm = cross_validate(lgbm_final, X, y, scoring=scoring, cv=skf)

comparison_of_metrics_dic = {'Accuracy Score': [cv_results_rf['test_accuracy'].mean(),cv_results_gbm['test_accuracy'].mean(),cv_results_xgboost['test_accuracy'].mean(),cv_results_lgbm['test_accuracy'].mean()],

                             'Precision Score': [cv_results_rf['test_precision'].mean(),cv_results_gbm['test_precision'].mean(),cv_results_xgboost['test_precision'].mean(),cv_results_lgbm['test_precision'].mean()],

                             'ROC-AUC Score': [cv_results_rf['test_roc_auc'].mean(),cv_results_gbm['test_roc_auc'].mean(),cv_results_xgboost['test_roc_auc'].mean(),cv_results_lgbm['test_roc_auc'].mean()]}

comparison_of_metrics_df = pd.DataFrame(comparison_of_metrics_dic, index=['RF Algorithm', 'GBM Algorithm', 'XGBOOST Algorithm',"LGBM Algorithm"])

comparison_of_metrics_df

plot_importance(model = rf_final, features = X, num = 10, col_name = "Feature Importance for RF Algorithm")
```

```
plt.show(block=True)

plot_importance(model = gbm_final, features = X, num = 10, col_name = "Feature Importance for GBM Algorithm")

plt.show(block=True)

plot_importance(model = xgboost_final, features = X, num = 10, col_name = "Feature Importance for XGBoost Algorithm")

plt.show(block=True)

plot_importance(model = lgbm_final, features = X, num = 10, col_name = "Feature Importance for LGBM Algorithm")

rt = df.loc[df["TripTypeName"] == "Round Trip"]

correlation_matrix(rt,num_cols)

rt['WeatherTemp'].fillna(rt['WeatherTemp'].mean(),inplace=True)

rt['SaleCountryCode'].fillna(rt['SaleCountryCode'].mode()[0],inplace=True)

rt['OutDestContinent'].fillna("US",inplace=True)

rt['ContactNationality'].fillna(rt['ContactNationality'].mode()[0],inplace=True)

rt['InDestContinent'].fillna("US",inplace=True)

rt['InOrigContinent'].fillna("US",inplace=True)

rt.drop(rt[rt.isna().any(axis=1)].index.to_list(),inplace=True)

cat_cols, cat_but_car, num_cols = grab_col_names(rt)

rt[num_cols].describe().T

for col in num_cols:
    print(str(col),' ---->',check_outlier(rt, col, q1=0.05, q3=0.95))

for col in ["CancellationMaxAssurance","SaleTotalFare"]:
    replace_with_thresholds(rt,col)

correlation_matrix(rt, num_cols)

meaningless_variables = high_correlated_cols(rt, plot=False, corr_th=0.90)

meaningless_variables
```

```
rt.drop(meaningless_variables,axis=1,inplace=True)

rt = rare_encoder(rt,0.05)

cat_cols, cat_but_car, num_cols = grab_col_names(rt)

cat_cols = [col for col in cat_cols if str(col) not in "Target"]

rt = one_hot_encoder(rt, cat_cols, drop_first=True)

y = rt["Target"]

X = rt.drop(["Target"], axis=1)

rf_model = RandomForestClassifier(random_state=17).fit(X,y)

gbm_model = GradientBoostingClassifier(random_state=17).fit(X,y)

xgboost_model = XGBClassifier(random_state=17, use_label_encoder=False).fit(X,y)

lgbm_model = LGBMClassifier(random_state=17).fit(X,y)

skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

scoring = ['accuracy', 'precision',"roc_auc"]

cv_results_rf = cross_validate(rf_model, X, y, scoring=scoring, cv=skf)

cv_results_gbm = cross_validate(gbm_model, X, y, scoring=scoring, cv=skf)

cv_results_xgboost = cross_validate(xgboost_model, X, y, scoring=scoring, cv=skf)

cv_results_lgbm = cross_validate(lgbm_model, X, y, scoring=scoring, cv=skf)

comparison_of_metrics_dic = {'Accuracy Score': [cv_results_rf['test_accuracy'].mean(),cv_results_gbm['test_accuracy'].mean(),cv_results_xgboost['test_accuracy'].mean(),cv_results_lgbm['test_accuracy'].mean()],

                'Precision Score': [cv_results_rf['test_precision'].mean(),cv_results_gbm['test_precision'].mean(),cv_results_xgboost['test_precision'].mean(),cv_results_lgbm['test_precision'].mean()],

                'ROC-AUC Score': [cv_results_rf['test_roc_auc'].mean(),cv_results_gbm['test_roc_auc'].mean(),cv_results_xgboost['test_roc_auc'].mean(),cv_results_lgbm['test_roc_auc'].mean()]}

comparison_of_metrics_df = pd.DataFrame(comparison_of_metrics_dic, index=['RF Algorithm', 'GBM Algorithm', 'XGBOOST Algorithm',"LGBM Algorithm"])
```

```
comparison_of_metrics_df

rf_params = {"max_depth": [8, 15, None],
             "max_features": [5, 7, "auto"],
             "min_samples_split": [15, 20],
             "n_estimators": [200, 300]}

xgboost_params = {"learning_rate": [0.1, 0.01],
                  "max_depth": [5, 8],
                  "n_estimators": [100, 200]}

lightgbm_params = {"learning_rate": [0.01, 0.1],
                   "n_estimators": [300, 500]}

gbm_params = {"learning_rate": [0.01, 0.1],
              "max_depth": [3, 8, 10],
              "n_estimators": [100, 500, 1000],
              "subsample": [1, 0.5, 0.7]}

rf_best_grid = GridSearchCV(rf_model, rf_params, cv=5, n_jobs=-1, verbose=True).fit(X, y)

rf_final = rf_model.set_params(**rf_best_grid.best_params_, random_state=17).fit(X, y)

xgboost_best_grid = GridSearchCV(xgboost_model, xgboost_params, cv=5, n_jobs=-1, verbose=True).fit(X, y)

xgboost_final = xgboost_model.set_params(**xgboost_best_grid.best_params_, random_state=17).fit(X, y)

lgbm_best_grid = GridSearchCV(lgbm_model, lightgbm_params, cv=5, n_jobs=-1, verbose=True).fit(X, y)

lgbm_final = lgbm_model.set_params(**lgbm_best_grid.best_params_, random_state=17).fit(X, y)

gbm_best_grid = GridSearchCV(gbm_model, gbm_params, cv=5, n_jobs=-1, verbose=True).fit(X, y)
```

```
gbm_final = gbm_model.set_params(**gbm_best_grid.best_params_, random_state=17).fit(X, y)

scoring = ['accuracy', 'precision',"roc_auc"]

cv_results_rf = cross_validate(rf_final, X, y, scoring=scoring, cv=skf)

cv_results_gbm = cross_validate(gbm_final, X, y, scoring=scoring, cv=skf)

cv_results_xgboost = cross_validate(xgboost_final, X, y, scoring=scoring, cv=skf)

cv_results_lgbm = cross_validate(lgbm_final, X, y, scoring=scoring, cv=skf)

comparison_of_metrics_dic = {'Accuracy Score': [cv_results_rf['test_accuracy'].mean(),cv_results_gbm['test_accuracy'].mean(),cv_results_xgboost['test_accuracy'].mean(),cv_results_lgbm['test_accuracy'].mean()],

                             'Precision Score': [cv_results_rf['test_precision'].mean(),cv_results_gbm['test_precision'].mean(),cv_results_xgboost['test_precision'].mean(),cv_results_lgbm['test_precision'].mean()],

                             'ROC-AUC Score': [cv_results_rf['test_roc_auc'].mean(),cv_results_gbm['test_roc_auc'].mean(),cv_results_xgboost['test_roc_auc'].mean(),cv_results_lgbm['test_roc_auc'].mean()]}

comparison_of_metrics_df = pd.DataFrame(comparison_of_metrics_dic, index=['RF Algorithm', 'GBM Algorithm', 'XGBOOST Algorithm',"LGBM Algorithm"])

comparison_of_metrics_df

plot_importance(model = rf_final, features = X, num = 10, col_name = "Feature Importance for RF Algorithm")

plt.show(block=True)

plot_importance(model = gbm_final, features = X, num = 10, col_name = "Feature Importance for GBM Algorithm")

plt.show(block=True)

plot_importance(model = xgboost_final, features = X, num = 10, col_name = "Feature Importance for XGBoost Algorithm")

plt.show(block=True)
```

```
plot_importance(model = lgbm_final, features = X, num = 10, col_name = "Feature Importance for LGBM Algorithm")

def plot_histograms(df, numerical_columns):
    num_cols = len(numerical_columns)
    num_rows = math.ceil(num_cols / 2)
    fig, axes = plt.subplots(nrows=num_rows, ncols=2, figsize=(15, num_rows * 6))
    # Define color palette
    colors = sns.color_palette('husl', num_cols)
    for i, column in enumerate(numerical_columns):
        row = i // 2
        col = i % 2
        ax = axes[row, col]
        df[column].hist(ax=ax, color=colors[i]) # Set color for each histogram
        ax.set_title(column)
        ax.set_xlabel('Value')
        ax.set_ylabel('Frequency')

    # Remove any empty subplots
    if num_cols % 2 != 0:
        fig.delaxes(axes[num_rows-1, 1])
    plt.subplots_adjust(top=0.85)
    plt.tight_layout()
    plt.show()
# Convert date column to datetime format
df['OutDepartureDate'] = pd.to_datetime(df['OutDepartureDate'])
# Filter the data for target value equal to 1
```

```
filtered_data = df[df['Target'] == 1]
# Group the filtered data by date and calculate the cumulative sum of the target column
grouped_data = filtered_data.groupby('OutDepartureDate')['Target'].sum().cumsum()
# Set the figure size
plt.figure(figsize=(12, 6))
# Create the line plot
plt.plot(grouped_data.index, grouped_data.values, linestyle='-', color='blue')
plt.xlabel('Date')
plt.ylabel('Cumulative Total Target Value = 1')
plt.title('Cumulative Increment of Total Ticket Cancellation over Time')
# Customize the plot style
plt.grid(True, linestyle='--', alpha=0.5)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
# Assuming your DataFrame is called 'df' with 'OutDepartureDate' as the date column
# Convert date column to datetime format
df['OutDepartureDate'] = pd.to_datetime(df['OutDepartureDate'])
# Calculate the total number of tickets sold within each date interval
tickets_sold = df.groupby('OutDepartureDate').size()
# Set the figure size
plt.figure(figsize=(12, 6))
# Create the line plot for cumulative total target value = 1
plt.plot(tickets_sold.index, tickets_sold.values, linestyle='-', color='blue')
# Customize the plot
plt.xlabel('Date')
```

```
plt.ylabel('Total Tickets Sold')

plt.title('Total Tickets Sold over Time')

# Set the x-axis tick labels to show only the month and day

tick_freq = max(len(tickets_sold.index) // 10, 1) # Adjust the tick frequency as desired

plt.xticks(tickets_sold.index[::tick_freq], tickets_sold.index.strftime('%m-%d')[::tick_freq], rotation=45)

plt.tight_layout()

plt.show()

# Assuming your DataFrame is called 'df' with 'temperature_column' and 'target_column' as the respective column names

# Define the temperature interval bins

temperature_bins = np.arange(df['WeatherTemp'].min(), df['WeatherTemp'].max() + 5, 5)

# Create a new column with the temperature interval category

df['temperature_interval'] = pd.cut(df['WeatherTemp'], bins=temperature_bins)

# Group the data by temperature interval and calculate the total canceled tickets within each interval

grouped_data = df[df['Target'] == 1].groupby('temperature_interval')['Target'].count()

# Set the figure size and style

plt.figure(figsize=(10, 6))

plt.style.use('seaborn')

# Plot the bar plot

grouped_data.plot(kind='bar', color='skyblue', edgecolor='black', alpha=0.7)

# Plot a line within the bar plot

grouped_data.plot(kind='line', marker='o', color='red')

plt.xlabel('Temperature Interval')

plt.ylabel('Total Canceled Tickets')

plt.title('Total Canceled Tickets by Temperature Interval')
```

```
plt.legend(['Line Plot', 'Bar Plot'])

plt.show()

# Assuming your DataFrame is called 'df' with 'Target' and 'SaleTotalFare' as the respective column names


# Define the range of interest for sale total fare values

fare_range = (0, 20000)

# Subset the data based on the fare range and target = 1

df_subset = df[(df['SaleTotalFare'] >= fare_range[0]) & (df['SaleTotalFare'] <= fare_range[1]) & (df['Target'] == 1)]

# Create a violin plot for the subsetted data

plt.figure(figsize=(10, 6))

pt.style.use('seaborn')

# Plot the violin plot

sns.violinplot(x='Target', y='SaleTotalFare', data=df_subset, palette='pastel')

plt.xlabel('Target')

plt.ylabel('Sale Total Fare')

plt.title('Violin Plot of Canceled Tickets and Sale Total Fare')

plt.show()

# Assuming your DataFrame is called 'df' with 'Target' and 'OutDepartureDate' as the respective column names

# Convert 'OutDepartureDate' to datetime type with the appropriate format

df['OutDepartureDate'] = pd.to_datetime(df['OutDepartureDate'], format='%d.%m.%Y %H:%M:%S')

# Define the date intervals

date_intervals = pd.date_range(start=df['OutDepartureDate'].min(), end=df['OutDepartureDate'].max(), freq='1M')
```

```
# Group the data by the departure date intervals and calculate the cancellation frequency
cancellation_freq = df[df['Target'] == 1].groupby(pd.cut(df['OutDepartureDate'], bins=date_intervals)).size()
# Create a bar plot
plt.figure(figsize=(12, 6))
plt.style.use('seaborn')
cancellation_freq.plot(kind='bar', color='teal')
plt.xlabel('Departure Date Intervals')
plt.ylabel('Frequency of Cancellations')
plt.title('Ticket Cancellations by Departure Date Intervals')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
def plot_validation_scores(algorithms, scores,title):
    # Create a list of algorithm names and corresponding scores
    algorithm_names = list(scores.keys())
    accuracy_scores = [score[0] for score in scores.values()]
    precision_scores = [score[1] for score in scores.values()]
    roc_auc_scores = [score[2] for score in scores.values()]
    # Set the positions of the bars on the x-axis
    x = range(len(algorithm_names))
    width = 0.2
    # Create the bar plot
    plt.figure(figsize=(10, 6))
    bar1 = plt.bar(x, accuracy_scores, width, label='Accuracy')
    bar2 = plt.bar([val + width for val in x], precision_scores, width, color='orange', label='Precision')
```

```
    plt.bar([val + 2 * width for val in x], roc_auc_scores, width, label='ROC-AUC')
    # Add red outline to the second bars
    for rect in bar2:
        rect.set_edgecolor('red')
        rect.set_linewidth(2.5)
    # Add labels, title, and legend
    plt.xlabel('Algorithms')
    plt.ylabel('Scores')
    plt.title(title)
    plt.xticks([val + width for val in x], algorithm_names, rotation=45)
    plt.legend()
    plt.show()
plot_validation_scores(base_round, tuned_round)
```