An

Industry Oriented Mini Project Report

on

A Holistic Approach on Airfare Price Prediction Using Machine Learning Techniques

A Report submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

By

A. Rajavamshi Goud

(20EG105454)

K. Vinay

(20EG105458)

T. Ajay

(20EG105459)

Under the Guidance of

Dr. B. V. V. Siva Prasad

Associate Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANURAG UNIVERSITY

VENKATAPUR– 500088

TELANGANA

Year 2023-24

ANURAG UNIVERSITY | SCHOOL OF ENGINEERING

CERTIFICATE

This is to certify that the Report / dissertation entitled "**A Holistic Approach on Airfare Price Prediction Using Machine Learning Techniques"** that is being submitted by A. Rajavamshi Goud (20EG105454), K. Vinay (20EG105458), T. Ajay (20EG105459) in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering to the Anurag University is a record of bonafide work carried out by them under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University for the award of any other degree or diploma.

Signature of the Supervisor

Dr. B. V. V. Siva Prasad

Associate Professor

Dean, CSE

External Examiner 1

External Examiner 2

DECLARATION

We hereby declare that the Report entitled **A Holistic Approach on Airfare Price Prediction Using Machine Learning Techniques** submitted for the award of Bachelor of Technology Degree is our original work and the Report has not formed the basis for the award of any degree, diploma, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma.

Place: Anurag University, Hyderabad

A. Rajavamshi Goud
20EG105454

K. Vinay
20EG105458

T. Ajay
20EG105459

ACKNOWLEDGEMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Dr. B. V. V. Siva Prasad, Associate Professor, Department of CSE** for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like to acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy**, Dean, Department of CSE, Anurag University. We also express our deep sense of gratitude to **Dr**. **V V S S S Balaram,** Academic Co-Ordinator, **Dr. Pallam Ravi**, Project Co-Ordinator and Project review committee members, whose research expertise and commitment to the highest standards continuously motivated me during the crucial stage our project work.

We would like express our special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering, Anurag University, for his encouragement and timely support in our B. Tech program.

A. Rajavamshi Goud
(20EG105454)

K. Vinay
(20EG105458)

T. Ajay
(20EG105459)

ABSTRACT

Globalization of markets involves new strategies and price policies from professionals that contribute to global competitiveness. Airline companies are changing tickets' prices very often considering a variety of factors based on their proprietary rules and algorithms that are searching for the most suitable price policy.

Recently, Artificial Intelligence (AI) models are exploited for the latter task, due to their compactness, fast adaptability, and many potentials in data generalization. This paper represents an analysis of airfare price prediction towards finding similarities in the pricing policies of different Airline companies by using Machine Learning (AdaBoost, Bagging, Gradient- Boost, Decision Trees, Extra Tree, Random Forest, Support Vector Machine, Multi-Layer Perceptron), Deep Learning (VGG11, VGG13, ResNet18, ResNet34, Mobile Net V2, Mobile Net V3) and Quantum Machine Learning Techniques (QSVM, QMLP). More specifically, a set of effective features is extracted from data flights of Aegean, Turkish, Austrian and Lufthansa Airlines for six popular international destinations.

The extracted set of features is then used to conduct a holistic analysis from the perspective of the end user who seeks the most affordable ticket cost, considering a destination-based evaluation including all airlines, and an airline-based evaluation including all destinations.

Overall, the project is to implement Machine Learning, Deep Learning, Quantum Machine Learning algorithms on Airfare data and make a record of accuracies for each and every dataset. Moreover, the accuracies are analyzed and made a comparative analysis, The best algorithm for prediction is taken from the study. This is the first implementation of Quantum Machine Learning algorithms for predicting the Airfare prices and First study of both Airline Based and Destination Based. The best predicted data and algorithm is found by comparative analysis and made prediction for the users. This makes A Holistic Approach for prediction of price.

List Of Figures

List Of Tables

INDEX

1.INTRODUCTION

The proposed holistic approach is described, focusing on the used data and the selected methods. Datasets, features description and visualization material are presented to highlight the level of competition and globalization affection in airfare tickets between destinations from different airline companies. Moreover, in this section, the ML, DL, and QML models that are employed are presented and a short description for each one is given to underline the differences in performance and capabilities between them.

Fig. 1 graphically illustrates the steps of the proposed methodology. In the first step of the methodology, four airlines and six destinations are considered. The extracted features are applied to eight ML and six DL models towards evaluating the best performing model. Evaluation is performed in two different perspectives. In the first experiment, destination-based evaluation takes place, where the same set of destinations are applied to the models regardless the airline. In the second experiment, airline-based evaluation is examined, where the data from each airline company is applied to the models, for all destinations. The best two performing ML models of the first step of the methodology are used in the second step and are extended to the quantum domain. More specifically, in the second step of the methodology, the two best performing airlines and their best three performing destinations from step 1 are examined, towards comparing ML models and the corresponding QML models. Comparative evaluation is considered from the same two perspectives as in step 1.

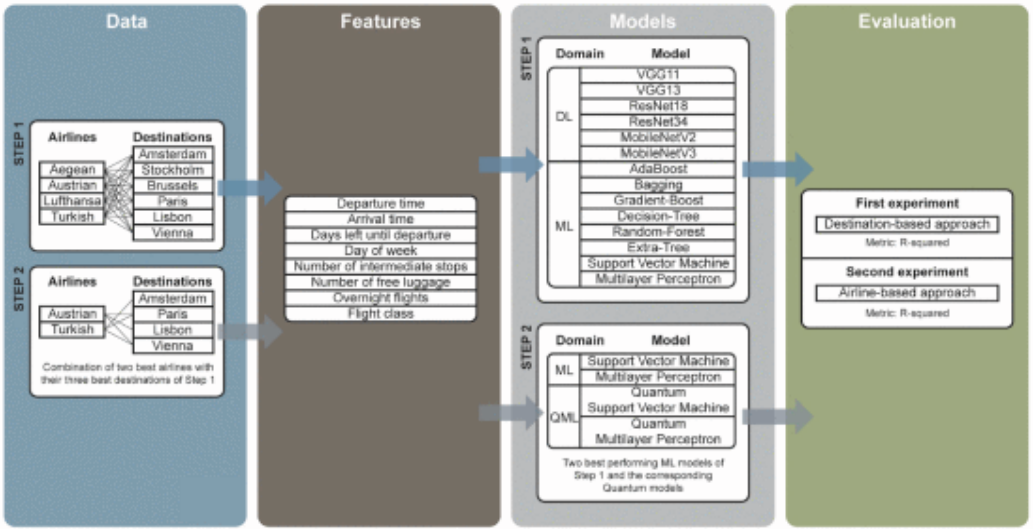

Figure 1.1:The proposed holistic approach to airfare price prediction.

1.1 Data Presentation and Description

The focus of this work is on the prediction of airfare prices for six different destinations for four airline companies. The airline companies are: Aegean Airlines, Austrian Airlines, Lufthansa Airlines and Turkish Airlines. The destinations of interest are the following:

Thessaloniki (SKG) – Amsterdam (AMS), (1907 Km)

Thessaloniki (SKG) – Stockholm (ARN), (2157 Km)

Thessaloniki (SKG) – Brussels (BRU), (1812 Km)

Thessaloniki (SKG) – Paris (CDG), (1863 Km)

Thessaloniki (SKG) – Lisbon (LIS), (2747 Km)

Thessaloniki (SKG) – Vienna (VIE), (985 Km)

The flight data are collected for the period of one year.1 It should be clarified here that flight data are not for exactly one year, due to the fact that some airline companies did not provide the same flights for all destinations all over the year, mainly due to demand variations. Table 1 summarizes the amounts of data flights for each destination and for each airline company.

In this work, the most descriptive features that affect the airfare price and were publicly available, were selected. For each flight data, a set of eight features (0:7) was used. Due to the difficulty of collecting flight data manually, Data Mining techniques were applied to acquire as many data as possible. Finally, for each flight the following eight features were considered:

Feature 0: Departure time
Feature 1: Arrival time
Feature 2: Days left until departure (0 - 350+)
Feature 3: Day of week (1-7)
Feature 4: Number of intermediate stops (0 - 2)
Feature 5: Number of free luggage (0 - 2)
Feature 6: Overnight flight (yes - 1 or no - 0)
Feature 7: Flight class (three-digit number, each digit 0 - 5)

Regarding feature 7, note that flight class is a three-digit integer number. Each digit independently represents a flight class, considering up to three correspondences per voyage. For instance, if the third digit of fight class is not zero, it means that the flight had two intermediate stops, thus, the voyage involved three corresponding flights in total, and each of the three digits informs about the involved ticket class. If the third digit is zero, it means that there was no third flight (only two flights) and so on. Every digit's value is ranged from 0 to 224, depending on the flight class of each of the corresponding flights, as follows:

Economy class– 110
Economy Standard class – 220
Business class – 222
First class – 224
No flight – 0

In what follows, a feature correlation analysis is presented based on the Pearson correlation [25] coefficient, in order to justify the eight features' selection and to highlight similarities among airline companies pricing policies through data

analysis. Pearson correlation coefficient takes values in the range -1 and 1 for each combination of features. Its mathematical formulation is:

$$r=\sum(xi-\bar{x})\,(Yi-\bar{y})/\sqrt{\sum(xi-\bar{x})^2\sum(Yi-\bar{y})^2} \quad \text{------- (1)}$$

In Equation (1) xi and Yi are the values of the data sample, $\bar{x}$ and $\bar{y}$ represent the mean of feature values and r is the correlation coefficient. The results of Pearson correlation for the airfare price dataset for all destinations and airline companies, are presented in Fig. 2 in the form of heatmaps.

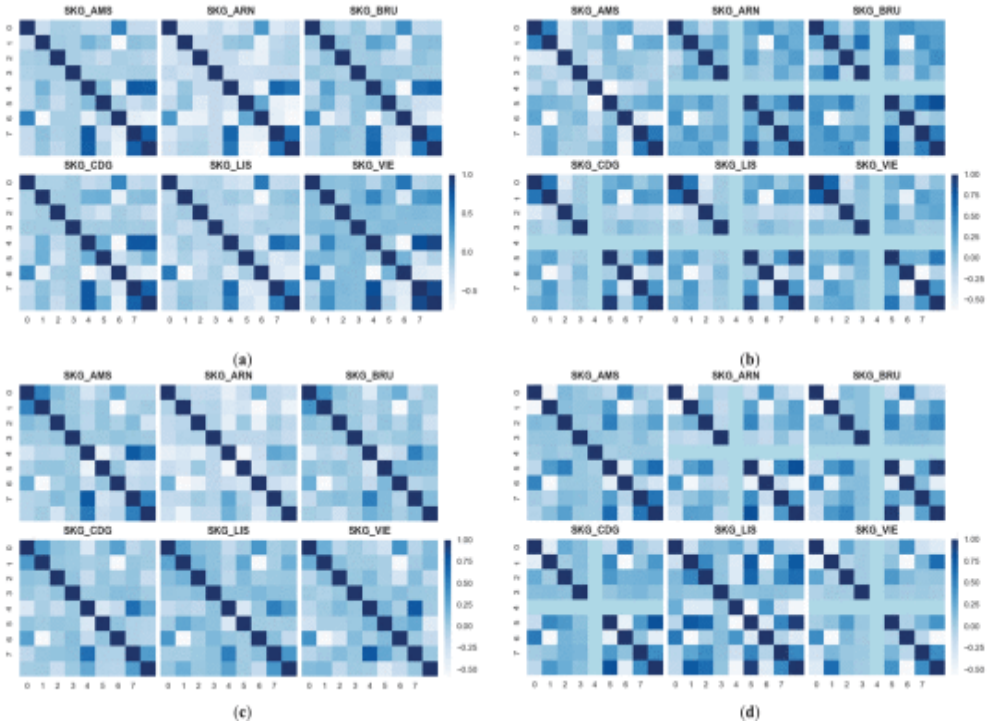

Figure 1.1.1. Pearson correlation coefficients heatmaps for each destination:

Austrian and Turkish airlines, as it can be observed from Fig. 1.1.1(b) and Fig. 1.1.1(d) have very few flights in the selected destinations and, thus, the number of stops (feature 4) has a low diversity and the correlation coefficients of this value equal to zero. A first notice is that Aegean displays more light colors in its heatmap, translated to less correlations between features in destinations of

greater distance (SKG_ARN, SKG_LIS) compared to other destinations which seem to have darker color values, translated to stronger correlations. The same observation can be made for Austrian, Turkish and Lufthansa, but only in the destination SKG_ARN. It is also important to mention that for every airline company and destination, it seems that flight class (feature 7) and price have a strong correlation despite the differences in the number of flights of each company. Based on this fact, it is easy to conclude that flight class has a strong impact on the competition between airline companies. In Fig. 3, the heatmaps of Pearson correlation coefficient for each airline company are presented with the destination as an extra feature (feature 8).

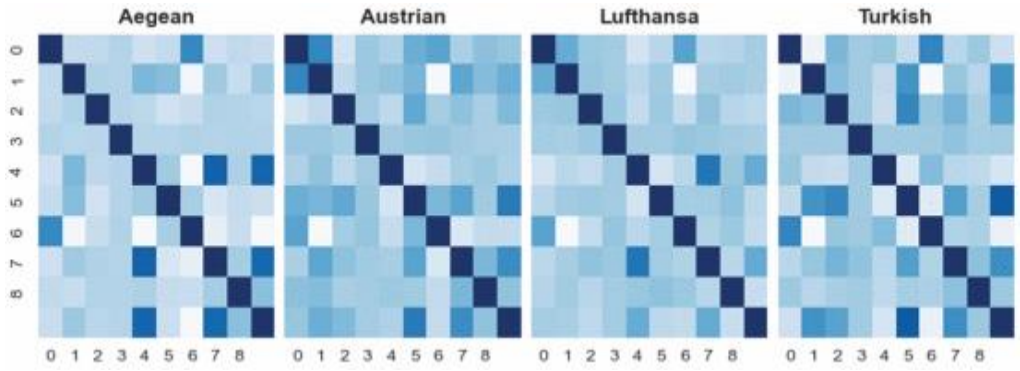

Figure 1.1.2. Pearson correlation coefficient for each airline company.

As observed in Fig. 1.1.2, flight class and destination display a strong correlation with price for every airline company. Moreover, Austrian and Lufthansa have similar correlations for departure and arrival time, while Aegean and Turkish have opposite correlations for the same two features. A more general notice is that Aegean and Turkish airlines have more diversity in correlations between features, while Austrian and Lufthansa tend to be smoother. Finally, considering Fig. 2 and Fig. 3, similarities between airline companies and their price policies can be highlighted despite the huge differences between the amounts of flights from each company. It can be concluded that by conducting a small data analysis, market globalization and competition level can be spotted between airline companies and their flight

destinations. In what follows, the AI models that have been used in this work are presented along with their characteristics

1.2 Problem Statement with Illustration:

Problem Statement:

Airfare price prediction is a critical challenge in the airline industry, where airlines need to adjust ticket prices dynamically to maximize revenue and stay competitive. However, existing pricing methods often fall short, leading to inaccurate predictions. We aim to improve prediction accuracy using advanced algorithms from three domains: Machine Learning, Deep Learning and Quantum Machine Learning (QML). The objective is to identify an algorithm that can consistently achieve predictive accuracies exceeding 89% to 99%.

Illustration:

Departure time: 9:15

Arrival time: 6:30

Days left until departure: 4

Day of the week: 3

Number of intermediate stops: 2

Number of free luggage: 1

Overnight flight: 1 (yes)

Flight class: First class

The accuracy predicted by the previously used machine learning algorithms like ML Models for Customer Segmentation(k-means clustering, DBSCAN, Agglomerative Clustering), Custom Recurrent Neural Networks (RNNs), fall under 70 percent.

The value predicted by clustering algorithms is 500

The value predicted by RNN is 530

The actual price is 650

Finally, The problem of false or inaccurate airfare price predictions from previously used models has significant implications for both consumers and airlines. This issue not only results in potential financial losses for travelers but also affects airlines' revenue and customer satisfaction. It's imperative to expand upon this problem to provide a more comprehensive understanding.

2.Literature Survey

The literature survey presented here covers a broad spectrum of topics, including yield management, nervous activity, perceptron, training algorithms, and the evolution of artificial intelligence. This review will elaborate on these topics and interconnections to provide a comprehensive overview.

The first abstract introduces the concept of yield management, which is a strategy widely applied in the airline and hospitality industries. It focuses on optimizing revenue by strategically managing pricing and capacity. The note highlights the allocation of perishable inventory among diverse customer segments. This concept underscores the significance of dynamic pricing and resource allocation in maximizing profits.

The second abstract addresses the application of propositional logic in understanding nervous activity. It proposes that neural events and their relationships can be analyzed using logical constructs. This perspective offers a unique intersection between neuroscience and formal logic. By approaching nervous activity with propositional logic, this abstract implies the potential for formalizing and understanding complex neural processes.

The third abstract introduces perceptron, one of the pioneering models in neural networks. Perceptron's have played a crucial role in the development of machine learning and neural networks. This abstract emphasizes the importance of information storage and organization within the context of the brain. It points to the connections between artificial intelligence and neuroscience, highlighting how early models like perceptron paved the way for modern machine learning.
The fourth abstract presents a training algorithm designed to maximize the margin between decision boundaries in classification tasks. This technique has wide-ranging applications across different classification functions. The automatic adjustment of the number of parameters is a key feature, ensuring the algorithm's adaptability to problem complexity. Moreover, it offers insights into the generalization performance and accuracy of classification models.

The fifth segment provides a historical perspective on the evolution of artificial intelligence. It traces the journey from the mathematical models of biological neurons introduced by Walter Pitts and Warren McCulloch in 1943 to the emergence of the perceptron by Frank Rosenblatt in 1950. The perceptron marked a significant step in the development of machine learning and neural networks, inspiring a host of subsequent models such as Support Vector Machines (SVM), k-Nearest Neighbors (KNN), and Boosting methods.

The final segment extends the literature survey to a practical application of machine learning and quantum machine learning in airfare price prediction. This approach encompasses the extraction of relevant flight features to predict airfare prices for various airline companies and destinations. What sets this study apart is its holistic approach, examining the problem from both destination-based and airline-based perspectives. Furthermore, it introduces Quantum Machine Learning as an innovative technique in solving this complex problem.

It explores a diverse range of sources spanning topics such as yield management, neural activity modeling, machine learning, and airfare price prediction. Within these sources, several existing methods are established. "Introduction to the Theory and Practice of Yield Management" and "A Logical Calculus of the Ideas Immanent in Nervous Activity" elucidate existing principles, focusing on yield management and neural activity representation through propositional logic, respectively. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain" delves into the foundational machine learning model, the perceptron. "A Training Algorithm for Optimal Margin Classifiers" provides an existing method that optimizes margins in classifiers. "Artificial Intelligence and Machine Learning Evolution" serves as a historical reference for the evolution of AI and ML. In contrast, "Airfare Price Prediction with a Holistic Approach" introduces a novel proposed method for airfare price prediction. It applies Machine Learning, Deep Learning, and Quantum Machine Learning models holistically, addressing the problem through destination-based and airline-based approaches, offering a fresh perspective on a long-standing issue.

In conclusion, this literature survey provides a comprehensive exploration of diverse topics within the realms of yield management, nervous activity, perceptron, training algorithms, and the evolution of artificial intelligence. These abstracts shed light on the interconnected nature of these fields and their ongoing contributions to the advancement of knowledge in both the scientific and practical domains. Moreover, the literature survey highlights the continuous evolution of artificial intelligence, from its nascent stages to its application in solving real-world problems like airfare price prediction.

2.1. Comparison of Existing Methods.

SI. No	Author(s)	Method	Advantages	Disadvantages
1	Vu et al	Implemented an airfare price prediction application using two ML models and focusing on features related to time for Vietnamese national airline company flights.	Specific focus on time related features.	Limited number of models considered and focus on a single airline company.
2	Theofanis Kalampokas	Constructed a custom RNN and compared it with classical ML	Integration of features from different domains.	Lack of information about the specific models used and limited

		models in airfare price prediction, incorporating features related to both basketball matches and airline flights		explanation of the results.
3	Joshi et al	Applied feature selection algorithms and hyperparameter optimization to find optimal model parameters and features for flight description in airfare price prediction.	Emphasis on feature selection and optimization techniques.	Limited information about the specific ML models used and their performance.

3.Proposed Method

The proposed method represents a robust framework designed to predict airfare prices with the aim of comparing and contrasting various machine learning models from three domains: Machine Learning (ML), Deep Learning (DL), and Quantum Machine Learning (QML). This approach provides an exhaustive analysis of their effectiveness in handling real-world airline data for precise airfare predictions.

3.1 Methodology Overview:

The proposed method unfolds in two distinct steps, each contributing valuable insights into airfare price prediction:

Step 1: Comparative Analysis of ML, DL, and QML Models

In the first step, the primary objective is to conduct a comprehensive comparative analysis of models from ML, DL, and QML domains. This comparison aims to assess the strengths and limitations of these models by examining their performance on real airline data. This step is further divided into two main approaches:

Destination-Based Approach:

This approach centers around evaluating model performance in the context of specific airline companies and their individual destinations.

It provides valuable insights into how different models perform in diverse scenarios and helps uncover the unique pricing strategies employed by airline companies. Analyzing specific destinations allows for a deeper understanding of the pricing dynamics and competition levels within the airline industry.

Airline-Based Approach:

In this approach, the scope is broadened to encompass all available destinations for selected airline companies. By considering multiple destinations, this approach offers a more comprehensive perspective on how different airline companies formulate their pricing strategies and adapt to diverse market conditions.

Step 2: Deployment and Real-Time Prediction

In this final step of the proposed method, the focus shifts from model development and optimization to the practical deployment of the airfare price prediction system. Step 3 includes the following essential components:

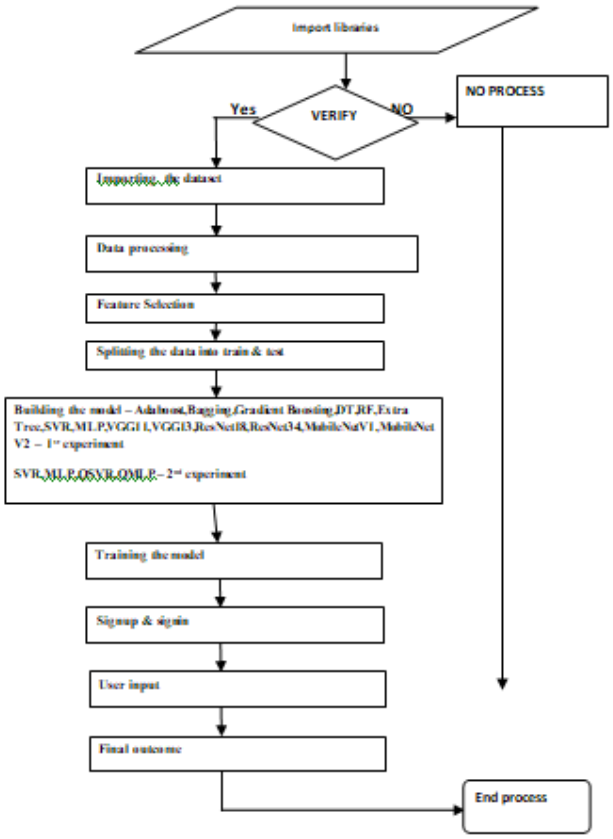

Figure 3.1.1: Dataflow of the proposed method

Model Deployment:

Scalable Architecture: The ensemble models, now fully optimized, are deployed on a scalable architecture, ensuring that they can handle real-time prediction requests efficiently.

Real-Time Prediction:

User Interface: A user-friendly interface is developed, providing an intuitive experience for travelers. Users can input their departure, destination, travel dates, and other relevant information to receive instant airfare price predictions.

Request Handling: The system is designed to handle a high volume of real-time prediction requests. Advanced load balancing techniques are employed to ensure smooth and responsive user experiences.

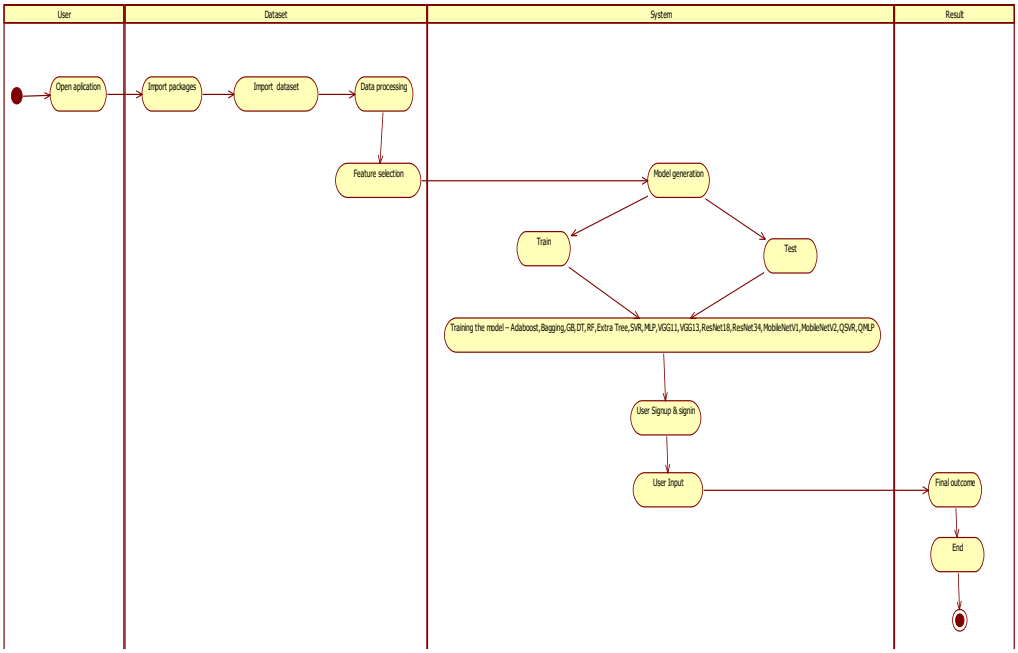

Figure 3.1.2: The Activity Diagram of proposed model

3.2 Key Components:

Machine Learning Models (ML): The ML models selected for this analysis include well-established algorithms such as AdaBoost, Bagging, Gradient Boost, Decision Trees, Random Forest, Extra Tree, Support Vector Machines (SVM), and Multi-Layer Perceptron (MLP). These models are known for their versatility and have found applications in a wide range of machine learning tasks.

Deep Learning Models (DL): The DL models chosen for examination are primarily from the Convolutional Neural Network (CNN) domain. These include prominent models like VGG, ResNet, and MobileNet. DL models are characterized by their ability to extract intricate features from data, making them particularly suitable for image-based prediction tasks.

Quantum Machine Learning (QML) Models: The QML models included in the analysis are QSVM and QMLP. These models leverage the principles of

quantum computing, notably the use of quantum bits (qubits), to enhance data separation and improve generalization capabilities.

Model Selection: Identify the best-performing models from the first step of the experiments, which includes machine learning (ML), deep learning (DL), and quantum machine learning (QML) models.

Model Stacking: Create an ensemble by stacking the selected models. Ensemble learning combines multiple models to improve predictive performance by leveraging their complementary strengths.

Weighted Averaging: Assign weights to each model in the ensemble. These weights are determined through cross-validation and aim to give more influence to models that consistently perform well on different routes and destinations.

Hyperparameter Tuning: Conduct an extensive hyperparameter tuning process for individual models and the ensemble as a whole. This involves searching through various hyperparameter combinations to find the best settings for each model..

Outlier Handling: Apply robust preprocessing techniques to identify and manage outliers that might adversely affect predictions. Common methods include Z-score scaling, robust scaling, and outlier removal.

Cross-Validation: Perform extensive cross-validation to assess the ensemble's performance. Cross-validation helps ensure that the ensemble's predictions are consistent and reliable across different subsets of the dataset.

3.3 Illustration of Proposed method:

Departure time: 9:15

Arrival time: 6:30

Days left until departure: 4

Day of the week: 3

Number of intermediate stops: 2

Number of free luggage: 1

Overnight flight: 1 (yes)

Flight class: First class

Actual Cost: $600

Predictions:

AdaBoost: Predicted Price - $421 (Accuracy: 70.1%)

Bagging: Predicted Price - $514 (Accuracy: 85.7%)

Gradient Boosting: Predicted Price - $504 (Accuracy: 84.1%)

Decision Tree: Predicted Price - $575 (Accuracy: 94.9%)

Random Forest: Predicted Price - $458 (Accuracy: 76.3%)

Extra Tree: Predicted Price - $515 (Accuracy: 85.9%)

SVR: Predicted Price - $318 (Accuracy: 53.0%)

MLP: Predicted Price - $421 (Accuracy: 70.3%)

VGG11: Predicted Price - $464 (Accuracy: 77.1%)

VGG13: Predicted Price - $450 (Accuracy: 75.6%)

ResNet18: Predicted Price - $460 (Accuracy: 76.7%)

ResNet34: Predicted Price - $455 (Accuracy: 75.1%)

MobileNetV1: Predicted Price - $466 (Accuracy: 73.6%)

MobileNetV2: Predicted Price - $459 (Accuracy: 76.9%)

QSVR: Predicted Price - $471 (Accuracy: 78.5%)

QMLP: Predicted Price - $6 (Accuracy: 0.1%)

In the proposed method we will implement all the selected algorithms of these three domains and store the accuracies. The comparative analysis is done on the stored data and the best predicting algorithms is identified from analysis and the further prediction is done by the best algorithm. This will help the users to get the accurate prices of airlines.

4.IMPLEMENTATION

4.1 Module Description:

Data Exploration: In this module, data is loaded into the system. This typically involves collecting and importing datasets from various sources. Data exploration includes tasks such as understanding the structure of the data, checking for missing values, and exploring basic statistics.

Data Processing: After data exploration, the data is processed to prepare it for modeling. This step may involve data cleaning, feature engineering, and transformation. It ensures that the data is in a suitable format for training machine learning models.

Splitting Data into Train & Test: The dataset is divided into two subsets: a training set and a testing set. The training set is used to train machine learning models, while the testing set is used to evaluate their performance. This is crucial to assess how well the models generalize to new, unseen data.

Model Generation: This is a critical phase in the project. It involves building and training machine learning models using various algorithms. It appears that you have two experiments:

Experiment 1 (Airlines for Specific States): You're using a range of regression algorithms, including AdaBoost Regression, Bagging Regression, Gradient Boosting Regression, Decision Tree Regression, Random Forest, Extra Tree, Support Vector Regression (SVR), Multilayer Perceptron (MLP), and several deep learning models (VGG11, VGG13, ResNet18, ResNet34, MobileNetV1, MobileNetV2).

Experiment 2 (Airline Destination): In this experiment, you're focusing on Support Vector Regression (SVR), Multilayer Perceptron (MLP), Quantum

Support Vector Regression (QSVR), and Quantum Multilayer Perceptron (QMLP). These algorithms are used to predict airline destinations, and you'll evaluate their accuracy.

User Signup & Login: This module is responsible for user registration and login. It provides access to the system and ensures that users can interact with the application securely.

User Input: Users can provide input to the system for making predictions. This input can be related to airline-related queries, and the system will use machine learning models to generate predictions based on the input.

Prediction: The system displays the final predictions based on user input and the trained machine learning models. Users can receive predictions related to their queries, such as flight details, pricing, or destinations.

4.2 Algorithms Used:

Machine Learning:

AdaBoost (Adaptive Boosting):

AdaBoost is an ensemble learning method that aims to improve the performance of weak classifiers. It operates by sequentially training a series of classifiers, where each classifier is assigned, weights based on its performance. Misclassified instances are given increased weight, allowing subsequent classifiers to focus on correcting the mistakes made by earlier ones.
AdaBoost assigns weights to the training instances and combines the outputs of individual classifiers with different weights to make a final prediction. The weight updates emphasize the mistakes, leading to more accurate results.
AdaBoost is commonly used in classification tasks, face detection, and text classification.

Bagging (Bootstrap Aggregation):

Bagging, or bootstrap aggregation, is an ensemble learning method aimed at reducing variance within a dataset. It works by creating multiple subsets of the training data through random sampling with replacement.

Bagging trains multiple models independently on the data subsets and aggregates their predictions, typically through averaging (for regression) or majority voting (for classification) to make a final prediction. Bagging is widely used in classification and regression tasks, particularly with decision trees.

Gradient Boosting:

Gradient Boosting is an ensemble learning technique used for classification and regression tasks. It sequentially trains models, with each new model correcting the errors of the previous one.

Gradient Boosting combines multiple weak learners into a strong learner by minimizing the residual errors in each iteration. It optimizes a differentiable loss function using gradient descent.

Gradient Boosting is a powerful method for various applications, including ranking problems and Kaggle competitions.

Decision Trees:

A Decision Tree is a non-parametric supervised learning algorithm used for classification and regression tasks. It creates a hierarchical tree structure with nodes and branches that make decisions based on feature values.

Decision Trees split data at each node based on feature attributes to create a tree structure. Leaf nodes provide classification labels or regression values.

Decision Trees are versatile and interpretable, used in a wide range of tasks, including medical diagnosis, fraud detection, and recommendation systems.

Random Forest:

Random Forest is an ensemble learning algorithm that combines the outputs of multiple Decision Trees to reach a final prediction. It is known for its robustness and ability to handle both classification and regression problems.

Random Forest builds an ensemble of Decision Trees, and the final prediction is made by averaging (for regression) or majority voting (for classification) of the individual tree predictions.

Random Forest is widely used in various domains, including finance, ecology, and bioinformatics.

Extra Trees (Extremely Randomized Trees):

Extra Trees is an ensemble method similar to Random Forest but introduces an extra layer of randomness.

Extra Trees selects random subsets of features for each node split, reducing the risk of overfitting. The final prediction is made by averaging the predictions of individual trees.

Extra Trees is effective when feature selection and model efficiency are important, commonly used in image classification and anomaly detection.

Support Vector Regression (SVR):

Support Vector Regression (SVR) is a machine learning algorithm used for regression analysis. It differs from traditional linear regression by finding a hyperplane that best fits the data points in a continuous space.

SVR seeks to find a hyperplane that maximizes the margin between data points while allowing some points to fall within the margin. Kernel functions are used to transform the data into a higher-dimensional space for non-linear regression.

SVR is employed in tasks such as financial forecasting, stock price prediction, and weather forecasting.

Multilayer Perceptron (MLP):

A Multilayer Perceptron (MLP) is a type of feedforward artificial neural network consisting of fully connected neurons with non-linear activation functions.

MLPs have multiple layers, including input, hidden, and output layers. They learn to map input data to output through a series of weighted connections and activation functions. Backpropagation is used for training.

MLPs are widely used in various applications, including image classification, natural language processing, and speech recognition.

Model	Algorithm type
AdaBoost Regressor	Boosting family
Bagging Regressor	Boosting family
Gradient Boost Regressor	Boosting family
Decision Tree Regressor	Tree based
Random Forest Regressor	Tree based
Extra Tree Regressor	Tree based
Support Vector Regressor (SVM)	Kernel function
Multi-Layer Perceptron (MLP)	Neural Network

Table 4.2.1: Selected Machine Learning (ML) Models

DEEP Learning:

VGG11 and VGG13:

VGG-11 and VGG-13 are deep convolutional neural network models known for their accuracy in image classification tasks.

These models consist of multiple convolutional and fully connected layers, designed to learn hierarchical features from images. They have a fixed architecture with a series of convolutional layers and fully connected layers.

VGG models are used in image classification, object recognition, and feature extraction in computer vision.

ResNet-18 and ResNet-34:

ResNet-18 and ResNet-34 are convolutional neural networks that belong to the ResNet family, designed for image classification.

These models use deep residual learning, where shortcut connections skip one or more layers. This architecture helps mitigate the vanishing gradient problem, making it possible to train very deep networks.

ResNet models are widely used in image recognition tasks, including image classification and object detection.

MobileNetV1 and MobileNetV2:

MobileNetV1 and MobileNetV2 are convolutional neural networks designed for mobile and embedded vision applications.

These models are optimized for efficiency and reduced model size while maintaining strong performance in image classification tasks.

MobileNet models are used in applications where computational resources are limited, such as mobile devices and edge computing.

Model	Training parameters (Millions)
VGG11	133
VGG13	133
Resnet18	11.4
Resnet34	21.5
MobileNetV2	3.4
MobileNetV3	2.4

Table 4.2.2: Selected Deep Learning (DL) Models

Quantum Machine Learning:

Quantum Support Vector Regression (QSVR):

QSVR is a quantum machine learning approach that leverages quantum computing to solve linear regression problems.

It employs quantum algorithms, such as the Harrow-Hassidim-Lloyd (HHL)

algorithm, to efficiently solve linear equations on a quantum computer.
QSVR can be used for regression tasks where quantum computing can provide advantages in terms of speed and efficiency.

Quantum Multilayer Perceptron (QMLP):

QMLP is a quantum variant of the traditional multilayer perceptron, designed to harness the power of quantum computing.
QMLP uses quantum principles to enhance the scalability and efficiency of neural network models, offering potential improvements in training deep networks.
QMLP has the potential to revolutionize neural network training for a wide range of applications, especially in domains where quantum computing can provide an edge.

Model	Algorithm type
Quantum Support Vector Regressor (QSVM)	Quantum kernel
Quantum Multiplayer Perceptron (QMLP)	Quantum neural network with 48 parameters

Table 4.2.3: Selected Quantum Machine Learning (QML) Models

4.3 Technologies Used:

TensorFlow: TensorFlow is an open-source library for machine learning and deep learning developed by Google. It is widely used for research and production applications, providing tools for creating, training, and deploying machine learning models, especially neural networks.

NumPy: NumPy is a fundamental library for scientific computing in Python. It provides a powerful N-dimensional array object and various tools for working with these arrays, including mathematical functions, linear algebra, and more.

Pandas: Pandas is an open-source Python library for data manipulation and analysis. It offers two primary data structures, Series and Data Frames, making it a versatile tool for handling and analyzing data.

Matplotlib: Matplotlib is a Python 2D plotting library that generates high-quality figures and charts for various data visualization needs. It's highly customizable and suitable for creating various types of plots.

Scikit-learn: Scikit-learn is a free and open-source machine learning library for Python. It provides a wide range of supervised and unsupervised learning algorithms and tools for data preprocessing, model selection, and performance evaluation.

Qiskit: Qiskit is an open-source framework for quantum computing developed by IBM. It allows researchers and developers to explore and experiment with quantum algorithms, circuits, and quantum hardware. It's a significant tool in the quantum computing revolution.

Flask: Flask is a lightweight and flexible web framework for Python. It's designed to create web applications quickly, with a minimalistic approach. Flask offers features for routing, request handling, templating, and more, making it suitable for various web development projects.

4.4 Sample Code:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.ensemble import AdaBoostRegressor, BaggingRegressor,
GradientBoostingRegressor, RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.neural_network import MLPRegressor
from qiskit import BasicAer
from qiskit.circuit.library import ZZFeatureMap
from qiskit_machine_learning.algorithms import QSVR
from qiskit_machine_learning.kernels import QuantumKernel
```

Load the dataset

```
data = pd.read_excel("../../dataset/First
Experiment/Aegean/Aegean(SKG_AMS).xlsx")
df = pd.DataFrame(data)
```

Data Preprocessing (Data cleaning and formatting)

```
for i in df.columns:
    print(i, "\t\t-\t", df[i].isna().mean() * 100)
df.info()
```

Split the data into features (X) and target (y)

```
X = df.drop(['price'], axis=1)
y = df['price']
```

Split the data into training and testing sets

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
shuffle=True)
```

Initialize a list to store model names and their respective accuracies

```
ML_Model = []
accuracy = []
```

Function to store model results

```
def storeResults(model, a):
    ML_Model.append(model)
    accuracy.append(round(a, 3))
```

AdaBoost

```
adr = AdaBoostRegressor(n_estimators=10000, learning_rate=0.01,
random_state=1)
adr.fit(X_train, y_train)
y_pred = adr.predict(X_test)
ab_sc = r2_score(y_test, y_pred)
storeResults('AdaBoost', ab_sc)
```

Bagging Regressor

```
clf = BaggingRegressor(DecisionTreeRegressor(), n_estimators=10000,
random_state=0)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
bag_sc = r2_score(y_test, y_pred)
storeResults('Bagging', bag_sc)
```

Gradient Boosting

```
gbr = GradientBoostingRegressor(n_estimators=10000, random_state=0)
gbr.fit(X_train, y_train)
y_pred = gbr.predict(X_test)
gb_sc = r2_score(y_test, y_pred)
storeResults('Gradient Boosting', gb_sc)
```

Decision Trees

```
dtr = DecisionTreeRegressor(random_state=0)
dtr.fit(X_train, y_train)
y_pred = dtr.predict(X_test)
dt_sc = r2_score(y_test, y_pred)
storeResults('Decision Tree', dt_sc)
```

Random Forest

```
from sklearn.ensemble import RandomForestRegressor
param_grid = {
    'n_estimators': [25, 50, 100, 150],
    'max_features': ['sqrt', 'log2', None],
    'max_depth': [3, 6, 9],
    'max_leaf_nodes': [3, 6, 9],
}
rfr = RandomForestRegressor(max_depth=2, random_state=0)
grid_forest = GridSearchCV(rfr, param_grid=param_grid)
grid_forest.fit(X_train, y_train)
y_pred = grid_forest.predict(X_test)
```

```
rf_sc = r2_score(y_test, y_pred)
storeResults('Random Forest', rf_sc)
```

Extra Trees

```
from sklearn.ensemble import ExtraTreesRegressor
etr = ExtraTreesRegressor(n_estimators=10000, random_state=0)
etr.fit(X_train, y_train)
y_pred = etr.predict(X_test)
et_sc = r2_score(y_test, y_pred)
storeResults('Extra Tree', et_sc)
```

Support Vector Regression (SVR)

```
sr = SVR(C=1.0, epsilon=0.2)
sr.fit(X_train, y_train)
y_pred = sr.predict(X_test)
svr_sc = r2_score(y_test, y_pred)
storeResults('SVR', svr_sc)
```

MLP (Multi-layer Perceptron)

```
mlp = MLPRegressor(random_state=1, max_iter=500)
mlp.fit(X_train, y_train)
y_pred = mlp.predict(X_test)
mlp_sc = r2_score(y_test, y_pred)
storeResults('MLP', mlp_sc)
```

DeepLearning:

```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Conv1D, Flatten, Dense
def create_model(input_shape, architecture):
   model = Sequential()
```

VGG11

```
   if architecture == 'VGG11':
      model.add(Conv1D(32, kernel_size=3, padding='same', activation='relu',
input_shape=input_shape))
      model.add(Conv1D(64, kernel_size=3, padding='same', activation='relu'))
      model.add(Conv1D(128, kernel_size=5, padding='same',
activation='relu'))
```

VGG13

```
 elif architecture == 'VGG13':
      model.add(Conv1D(16, kernel_size=3, padding='same', activation='relu',
```

```
input_shape=input_shape))
        model.add(Conv1D(64, kernel_size=5, padding='same', activation='relu'))

        ResNet18
elif architecture == 'ResNet18':
        model.add(Conv1D(16, kernel_size=3, padding='same', activation='relu',
input_shape=input_shape))
        model.add(Conv1D(32, kernel_size=3, padding='same', activation='relu'))
        model.add(Conv1D(64, kernel_size=5, padding='same', activation='relu'))

        ResNet34
    elif architecture == 'ResNet34':
        model.add(Conv1D(64, kernel_size=3, padding='same', activation='relu',
input_shape=input_shape))
        model.add(Conv1D(128, kernel_size=3, padding='same',
activation='relu'))
        model.add(Conv1D(516, kernel_size=5, padding='same',
activation='relu'))

        MobileNetV1
    elif architecture == 'MobileNetV1':
        model.add(Conv1D(8, kernel_size=3, padding='same', activation='relu',
input_shape=input_shape))
        model.add(Conv1D(16, kernel_size=3, padding='same', activation='relu'))
        model.add(Conv1D(32, kernel_size=5, padding='same', activation='relu'))

        MobileNetV2
    elif architecture == 'MobileNetV2':
        model.add(Conv1D(128, kernel_size=3, padding='same',
activation='relu', input_shape=input_shape))
        model.add(Conv1D(516, kernel_size=3, padding='same',
activation='relu'))
        model.add(Conv1D(1024, kernel_size=5, padding='same',
activation='relu'))
    model.add(Flatten())
    model.add(Dense(50, activation='relu'))
    model.add(Dense(20, activation='relu'))
    model.add(Dense(1)
    model.compile(loss='mean_squared_error', optimizer='adam')
    return model
```

Quantum Support Vector Regression (QSVR)

```
statevector_simulator = QuantumInstance(
```

```
    BasicAer.get_backend("statevector_simulator"),
    shots=1,
    seed_simulator=algorithm_globals.random_seed,
    seed_transpiler=algorithm_globals.random_seed,
)
feature_map = ZZFeatureMap(feature_dimension=7, reps=2)
qkernel = QuantumKernel(
    feature_map=feature_map, quantum_instance=statevector_simulator
)
qsvr = QSVR(quantum_kernel=qkernel)
qsvr.fit(X_train, y_train)
labelsTest = qsvr.predict(X_test)
qsvr_sc = r2_score(y_test, labelsTest)
storeResults('QSVR', qsvr_sc)
```

Quantum MLP (QMLP)

```
#from skqulacs.circuit.pre_defined import create_qcl_ansatz
from skqulacs.qnn.solver import Bfgs
from skqulacs.qnn import QNNRegressor
from sklearn.metrics import r2_score

# Define the number of qubits, depth, and other parameters
n_qubit = 5
depth = 3
time_step = 0.5
solver = Bfgs()
maxiter = 200

# Create the quantum circuit using a predefined ansatz
circuit = create_qcl_ansatz(n_qubit, depth, time_step, 0)

# Initialize the QNNRegressor model
model = QNNRegressor(circuit, solver, cost='mse', do_x_scale=True,
do_y_scale=True, x_norm_range=0.6, y_norm_range=0.7)

# Train the QMLP model
opt_loss, theta = model.fit(X_train, y_train, maxiter)

# Predict using the trained QMLP model
y_pred = model.predict(X_test)

# Calculate and display the accuracy of the predictions
qmlp_sc = r2_score(y_test, y_pred)
print('QMLP Accuracy:', qmlp_sc)
```

5.Experiment Results:

5.1 Experimental Setup

In this work, two experiments are conducted in order to cover the proposed holistic approach for the target application problem. In the first experiment, namely the destination-based approach, the selected models from ML, DL, and QML domains are used to find the best choice for each destination per Airline Company. With this experiment, it can be concluded the optimal set of models that describe the same destinations for separate airline companies, having similar airfare price prediction accuracies. To accomplish that, the entire dataset was split for each destination for each airline company. More specifically, 24 datasets were created for four airline companies and six destinations. In the second experiment, namely the airline-based approach, the same strategy as the first experiment was followed, with the intention this time to locate the best models for each airline company that could describe all six destinations at the same time. For this reason, the dataset was split into four parts, based on the four selected airline companies. After that, a new feature was added to the four datasets, which describe the destination, ranging from 0 to 6, to make the dataset of the second experiment more distinct among the ML, DL, and QML models. For DL models the datasets features' values are normalized and converted to images in order to be used as inputs in the CNN models.

QML models were excluded at this phase since 28 different experiments would take a very long time to be processed, considering the amounts of flights in Table 1. This exception is also justified by the units of time during the learning process of the models from each domain, where for ML and DL the training process took hours and, therefore, for QML models would require days for only one destination. To computationally verify this, consider that in order to simulate on a classical computer a six-dimensional qubit state, a 64-dimensional vector is required since $26=64$. This computation for a classical computer is hard since bits can be only in one state at a time. Based on the proposed dataset of features, 8 qubits are used for the first experiment and 9 for the second. Thus, the dimensions are 256 and 512, respectively, for each datum.

In addition, the computational branches in QMLP are doubled for the weight values of feature and, thus, 65.536 and 262.144 dimensions, respectively, are finally required, translated in millions of flops for a classical machine. Based on the above, only the comparison of QML with ML models for only two airlines, Austrian and Turkish, have been conducted, only for the three best destinations from each of the two selected airline companies.

In general, for each experiment including the QML domain, the prediction accuracy (%) was measured and recorded, using Cross Validation method after a fit-and-predict phase completed with a dataset split in 80% percent for training and 20% for validation. The prediction accuracy is measured by R-squared (R^2) score metric:

$$R2=1-\sum(yi-y\hat{}i)^2\sum(yi-y)^2 \quad \text{——————} (2)$$

where, yi is the target value, y^i is the predicted value and y is the mean of all target values. In Equation (2), the numerator is the sum squared of regression, which is the difference between predicted and real values. The denominator is the sum of the total squared and expresses the distance of the real data from the mean of the total. R^2 score measures the variation between the predicted and input data and takes values from 0 to 1. For the error rate, Mean Squared Error is selected:

$$MSE=1/n\sum(yi-y\hat{}i)^2$$

where, yi is the real value and y^i is the predicted value and the sum of squared differences between these two values is divided by the total number of data, expressing the distance of each input datum from the regression line that is formed by each model. Thus, the less the error, the closer to the regression line the input datum.

Framework	Domain	Models
PyTorch	DL	[36]
Scikit-Learn	ML	[37]
PennyLane	QML	[38]
Qiskit	QML	[39]

Table 5.1.1: Frameworks for the Implementation of the Experiments

From PyTorch [36] all the presented CNN [7] models were applied along with the learning process on a GPU unit. From Scikit-Learn [37] all the ML models were used and fitted on a CPU unit. From PennyLane [38] QMLP network was formed and executed on a simulator that benefited CPU unit. Under the same principles, QSVM was applied from the Qiskit framework [39]. The hardware specifications where all the above experiments have been conducted are presented below:

CPU: AMD Ryzen™ Threadripper™2920X, 12 cores (24 threads), 3.5GHz base clock.
RAM: 32 GB DDR 4.
GPU: NVIDIA GeForce RTX 2060 SUPER 8 GB VRAM.
STORAGE: Viper M.2 vpn100 3450 MB/s-read, 3000 MB/s-write.

In what follows, the experimental results for the ML and DL models are comparatively presented and analyzed, including ML and DL models, for four airline companies and six destinations, by conducting two experimental approaches. Then, the best ML models are compared with the QML models' performance results for two airline companies and three destinations, for the same two experimental approaches.

5.2 Parameters considered for Project:

Airlines and Destinations:

The study focuses on four airline companies: Aegean Airlines, Austrian Airlines, Lufthansa Airlines, and Turkish Airlines.

Six destinations are considered: Thessaloniki (SKG) – Amsterdam (AMS), Thessaloniki (SKG) – Stockholm (ARN), Thessaloniki (SKG) – Brussels (BRU), Thessaloniki (SKG) – Paris (CDG), Thessaloniki (SKG) – Lisbon (LIS), and Thessaloniki (SKG) – Vienna (VIE).

Features for Predictive Models:

The models use a set of eight features for each flight data, which include:

Feature 0: Departure time

Feature 1: Arrival time

Feature 2: Days left until departure (ranging from 0 to 350+)

Feature 3: Day of the week (ranging from 1 to 7)

Feature 4: Number of intermediate stops (ranging from 0 to 2)

Feature 5: Number of free luggage (ranging from 0 to 2)

Feature 6: Overnight flight (binary: 1 for yes, 0 for no)

Feature 7: Flight class (a three-digit number representing flight class with values ranging from 0 to 5)

Experimental Approaches:

The study is divided into two steps: Step 1 and Step 2, with each step having two experimental approaches. Step 1 focuses on ML vs. DL models, and Step 2 involves a comparison between ML and QML (Quantum Machine Learning) models.

Model Evaluation Metrics:

The evaluation of model performance is based on the coefficient of determination (R2), and the results are presented for different destinations and airlines.

Comparison of Models:

The text discusses the performance of various models, including Bagging, Multilayer Perceptron (MLP), Random Forest, Extra-Tree, Quantum Machine Learning Perceptron (QMLP), and Quantum Support Vector Machine (QSVM), among others.

Performance Observations:

The text provides insights into which models perform best for different destinations and airlines based on R2 scores. It discusses how various factors, such as the number of flights, correlations, and pricing policies, impact the model's performance.

Accuracy:

Accuracy plays a major role in the project; the project is mainly focused on capturing of accuracies of every algorithm for different datasets and perform a comparative analysis. This study will help to identify best predicting algorithm.

6.Discussion Of Results:

We have conducted two experiments, where the First experiment is the comparison between Machine Learning and Deep Learning algorithms. The Second Experiment is the comparison between Quantum Machine Learning and Two Algorithms for Machine Learning.

6.1 First Experiment (ML VS DL):

In the first experiment, a comprehensive analysis was conducted, involving the implementation of 14 distinct algorithms. These algorithms were divided into two categories: Machine Learning (ML) and Deep Learning (DL), with 8 belonging to ML and 6 to DL.

The goal was to predict airfare prices for four different airlines across six sets of destinations, resulting in a total of 24 datasets.

After performing extensive evaluations, it was observed that the Decision Tree algorithm consistently stood out as the top performer. This algorithm exhibited a remarkable capability for predicting airfare prices accurately. Notably, in the case of Austrian Airlines and its flights to Stockholm, the Decision Tree algorithm achieved a remarkable 100 percent accuracy, surpassing the initial expectations. This remarkable performance highlights the Decision Tree's effectiveness in modeling and predicting airfare prices across various scenarios. The high accuracy achieved in the specific instance of Austrian Airlines and the Stockholm destination demonstrates its potential to provide exceptionally reliable predictions.

In summary, the first experiment's results clearly indicate that the Decision Tree algorithm is the best choice for airfare price prediction. Its consistent high accuracy across different airlines and destinations makes it a standout performer in this context. This analysis underscores the Decision Tree's proficiency in tackling the complex task of airfare price prediction with exceptional precision.

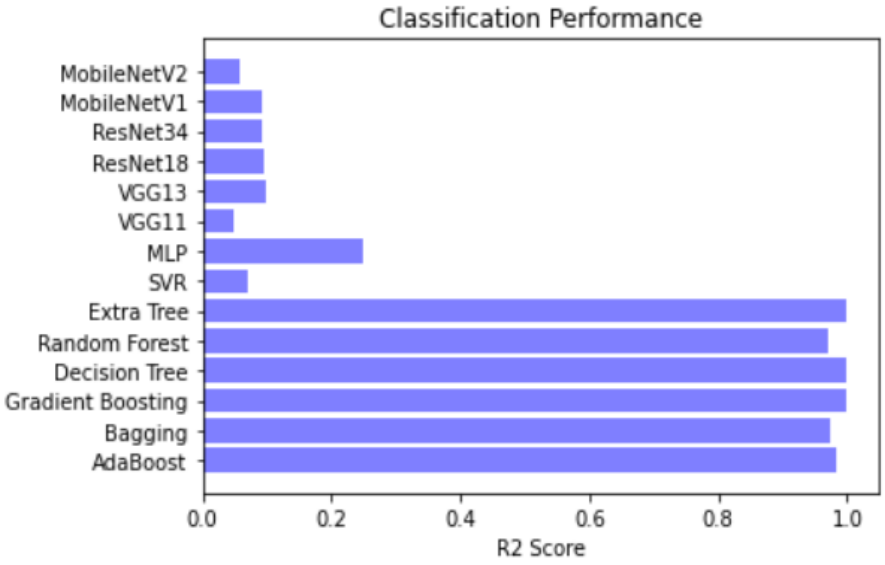

Figure 6.1.1 : The Graphical Representation of accuracies for Austrian Stockholm data.

```
result
```

	ML Model	R2-Score
0	AdaBoost	0.983
1	Bagging	0.974
2	Gradient Boosting	1.000
3	Decision Tree	1.000
4	Random Forest	0.969
5	Extra Tree	1.000
6	SVR	0.070
7	MLP	0.249
8	VGG11	0.049
9	VGG13	0.097
10	ResNet18	0.094
11	ResNet34	0.091
12	MobileNetV1	0.092
13	MobileNetV2	0.058

Figures 6.1.2: The Numerical accuracies for Austrian Stockholm data.

6.2 Second Experiment (ML VS QML):

In the second experiment, four different algorithms were employed to predict airfare prices, with two being machine learning (ML) algorithms (Support Vector Regression - SVR and Multilayer Perceptron - MLP) and two being quantum machine learning (QML) algorithms (Quantum Support Vector Regression - QSVR and Quantum Multilayer Perceptron - QMLP). These algorithms were evaluated across four different airlines, with the goal of determining their predictive accuracy in an airline-based approach.

The results of this second experiment revealed that QMLP, one of the quantum machine learning algorithms, provided lower accuracy levels compared to the other algorithms. On the other hand, QSVR showed somewhat satisfactory predictive performance, particularly when considering Turkish Airlines. For this specific airline, QSVR achieved an accuracy of 78.5%, which was higher than the accuracies obtained with the other three airlines.

This suggests that, in the context of the second experiment, quantum machine learning algorithms did not deliver the level of accuracy expected, with QSVR outperforming QMLP in most cases. Consequently, the choice of the best algorithm for airfare price prediction would depend on the airline in question.
In this specific experiment, it was found that the Decision Tree algorithm performed the best when used in combination with Austrian Airlines for airfare price prediction. The Decision Tree algorithm demonstrated the highest accuracy among the algorithms tested.

In summary, while the second experiment did not yield the expected accuracy levels for the quantum machine learning algorithms, QSVR showed relatively better results. The Decision Tree algorithm, when coupled with Austrian Airlines, proved to be the most effective choice for airfare price prediction in this context. These findings highlight the importance of selecting an appropriate algorithm that aligns with the specific airline's dataset and pricing characteristics for accurate predictions.

```
result
```

	ML Model	R2-Score
0	SVR	0.174
1	MLP	0.785
2	QSVR	0.785
3	QMLP	0.001

Figure 6.2.1: The Numerical accuracies for Turkish Airline.

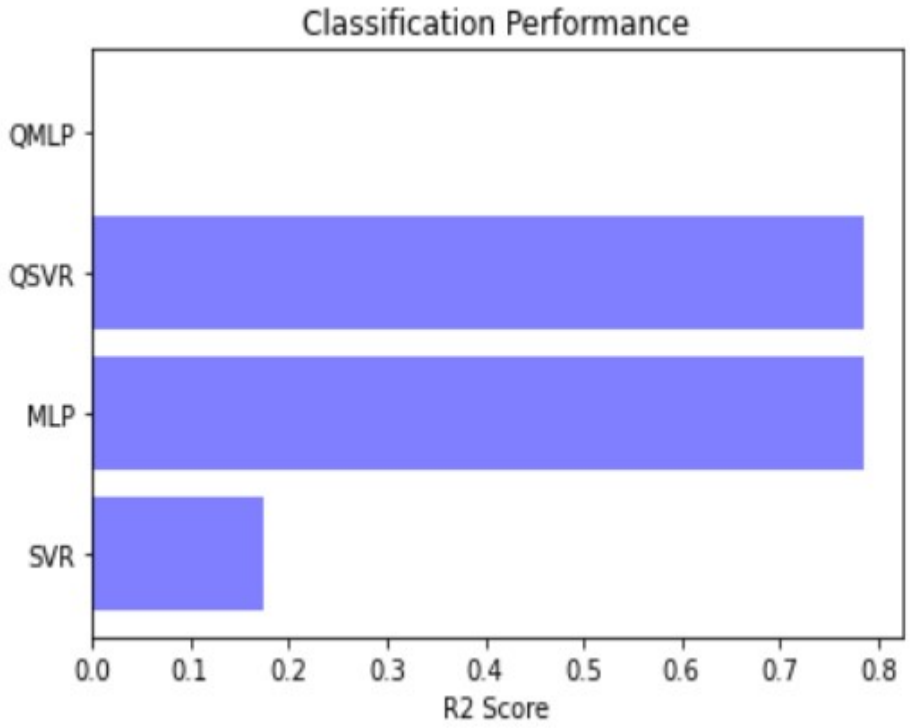

Figure 6.2.2: The Graphical Representation of accuracies for Turkish Airline

6.3 Testcases:

Arrival Time	Destination Time	Day of week	Days before Departure	No. of stops	No. of luggage's	Over night flight	Class	Result
16:10	19:45	4	11	1	1	0	220	**174.77**
16:15	9:15	5	12	1	1	1	110	**225.77**
16:15	22:50	1	15	1	2	0	222	**1007.1**
16:10	9:15	4	18	1	1	1	110	**174.77**
16:10	9:15	4	25	1	1	1	110	**290.77**
16:10	19:45	4	25	1	1	0	110	**336.77**
16:15	9:15	4	32	1	1	1	110	**245.77**
16:10	19:45	5	103	1	1	1	220	**215.77**
16:10	9:15	7	196	1	1	0	220	**250.77**
16:10	19:45	2	226	1	1	0	110	**188.77**
16:15	9:15	5	229	1	1	1	220	**220.77**
16:10	19:45	7	329	1	2	0	222	**2070.58**
16:15	9:15	7	329	1	2	1	222	**1185.4**
16:10	19:45	5	327	1	2	0	222	**1033.18**
16:15	9:15	5	327	1	2	0	224	**2567.58**
16:10	19:45	4	326	1	2	1	222	**1043.85**
16:15	9:15	4	326	1	2	1	222	**996.25**

6.4 Experiment Screenshots:

Figure 6.4.1: Image of Interface

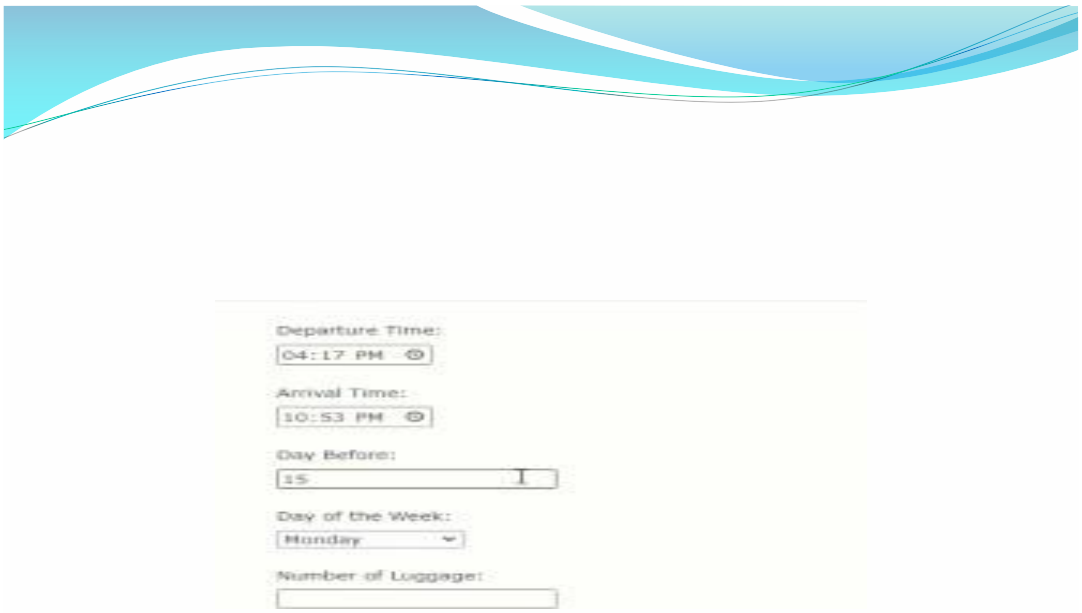

Figure 6.4.2: Image of entering the data in Interface

Figure 6.4.3: Image of displaying the output.

7. Conclusion

This research explored airfare price prediction using a diverse set of machine learning, deep learning, and quantum machine learning models across different airlines and destinations. Notably, many models achieved high accuracies, offering valuable insights for airlines to optimize pricing strategies. The study identified the potential of quantum machine learning, despite its challenges, and emphasized the need for improved quantum hardware accessibility. Additionally, it highlighted the importance of selecting the most suitable algorithm for specific airline datasets, with the Decision Tree algorithm outperforming quantum machine learning models in certain cases, such as with Austrian Airlines.

8.REFERENCES

[1] S. Netessine and R. Shumsky, ‘‘Introduction to the theory and practice of yield management,’’ INFORMS Trans. Educ., vol. 3, no. 1, pp. 34–44, Sep. 2002.

[2] W. S. McCulloch and W. Pitts, ‘‘A logical calculus of the ideas immanent in nervous activity,’’ Bull. Math. Biophys., vol. 5, no. 4, pp. 115–133, Dec. 1943.

[3] F. Rosenblatt, ‘‘The perceptron: A probabilistic model for information storage and organization in the brain,’’ Psychol. Rev., vol. 65, no. 6, pp. 386–408, 1958.

[4] B. E. Boser, I. M. Guyon, and V. N. Vapnik, ‘‘A training algorithm for optimal margin classifiers,’’ in Proc. 5th Annu. workshop Comput. Learn. theory, Jul. 1992, pp. 144–152.

[5] E. Fix and J. L. Hodges, ‘‘Discriminatory analysis. Nonparametric discrimination: Consistency properties,’’ Int. Stat. Rev./Revue Internationale de Statistique, vol. 57, no. 3, p. 238, Dec. 1989.

[6] R. E. Schapire, "The strength of weak learnability", Mach. Learn., vol. 5, no. 2, pp. 197-227, Jun. 1990.

[7] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", Biol. Cybern., vol. 36, no. 4, pp. 193-202, Apr. 1980.

[8] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition", Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[9] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, et al., "Mastering the game of go with deep neural networks and tree search", Nature, vol. 529, no. 7587, pp. 484-489, Jan. 2016.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S.

Ozair, et al., "Generative adversarial networks", Commun. ACM, vol. 63, no. 11, pp. 139-144, Oct. 2020.

[11] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring", Proc. 35th Annu. Symp. Found. Comput. Sci., pp. 124-134, 1994.

[12] L. K. Grover, "A framework for fast quantum mechanical algorithms", Proc. 30th Annu. ACM Symp. Theory Comput. (STOC), pp. 53-62, 1998.

[13] M. Andrecut and M. K. Ali, "Quantum associative memory", Int. J. Mod. Phys. B, vol. 17, no. 12, pp. 2447-2472, May 2003.