Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS 2023)
IEEE Xplore Part Number : CFP23CB2-ART ; ISBN : 979-8-3503-2579-9

ONTIME DELIVERY TIME PREDICTION USING MACHINE LEARNING

Mohammed Ali Shaik
School of Computer Science & Artificial Intelligence
SR University
Warangal, Telangana State, India niharali@gmail.com

Yasam Ashritha
School of CS&AI
SR University
Warangal ,Telangana, India
Ashrithayasam@gmail.com

Title: On-Time Delivery Time Prediction Using Machine Learning

Abstract:

The rapid expansion of e-commerce platforms has underscored the critical need for accurate delivery time estimation to meet customer expectations and ensure satisfaction. In response, researchers and practitioners have increasingly turned to machine learning methodologies to tackle the complexities inherent in predicting online delivery times. This paper presents a comprehensive review of the methodologies, advancements, and challenges associated with delivery time prediction within the realm of online platforms.

The review begins by delineating the multifaceted factors influencing delivery times, encompassing geographical considerations, traffic patterns, order characteristics (such as size, weight, and fragility), and historical delivery data. Understanding these factors is essential for developing robust prediction models capable of capturing the nuances of online delivery dynamics.

Subsequently, the paper delves into an extensive examination of machine learning algorithms employed in delivery time prediction. From traditional regression models to cutting-edge neural networks and ensemble methods, a diverse array of techniques have been applied to address the complexities of delivery time estimation. Each algorithm is scrutinized in terms of its strengths, limitations, and applicability within the context of online delivery operations.

Moreover, the review elucidates the evolving landscape of data sources utilized for delivery time prediction, including real-time GPS data, historical order records, weather information, and traffic reports. Harnessing these diverse data streams effectively is crucial for enhancing the accuracy and reliability of delivery time estimates.

Furthermore, the paper discusses the challenges confronting the field, ranging from data sparsity and noise to scalability and computational complexity. Addressing these challenges necessitates innovative approaches, including the integration of advanced data preprocessing techniques, the development of scalable machine learning architectures, and the exploration of hybrid models combining machine learning with optimization and simulation methods.

Finally, the review outlines potential avenues for future research and development, such as the incorporation of additional contextual information (e.g., customer preferences, delivery personnel availability) and the exploration of emerging technologies like reinforcement learning and federated learning for more adaptive and personalized delivery time prediction systems.

In conclusion, this comprehensive review serves as a valuable resource for researchers, practitioners, and stakeholders seeking to optimize online delivery operations, enhance customer experience, and drive innovation in the e-commerce industry. By synthesizing the latest advancements and highlighting future directions, this paper aims to foster continued progress in the field of online delivery time prediction.

Introduction:

Introduction to Online Delivery Time Prediction using Machine Learning

Online delivery services have become an integral part of modern life, providing convenience and efficiency to consumers. One crucial aspect of online delivery is accurately predicting the time it takes for an order to be delivered to the customer. This is influenced by various factors such as distance, traffic conditions, and order volume.

Machine learning offers a promising approach to tackle this problem by leveraging historical delivery data and other relevant features to make predictions. By analyzing past delivery patterns and real-time environmental factors, machine learning models can learn to estimate delivery times with increased accuracy.

Objectives:

The primary objective of implementing online delivery time prediction using machine learning is to provide customers and delivery service providers with reliable estimates of delivery times. This helps improve customer satisfaction, optimize delivery operations, and enhance overall efficiency.

Key Components:

1. *Data Collection:* Gathering relevant data such as historical delivery times, order details, geographic information, traffic conditions, weather data, and other relevant features.

2. *Data Preprocessing:* Cleaning and preprocessing the collected data to remove outliers, handle missing values, and format it into a suitable structure for machine learning algorithms.

3. *Feature Engineering:* Extracting meaningful features from the data that can contribute to predicting delivery times, such as distance between delivery points, time of day, day of the week, and weather conditions.

4. *Model Selection:* Choosing appropriate machine learning algorithms for the prediction task, such as regression models, time series forecasting models, or ensemble methods.

5. *Model Training:* Training the selected machine learning model using the preprocessed data to learn patterns and relationships between input features and delivery times.

6. *Model Evaluation:* Evaluating the performance of the trained model using metrics such as mean absolute error, root mean squared error, or mean absolute percentage error.

7. *Deployment:* Integrating the trained model into the online delivery platform to make real-time predictions based on incoming orders and environmental factors.

Benefits:

- *Improved Customer Experience:* Accurate delivery time predictions help manage customer expectations and reduce frustration caused by delays.

- *Optimized Operations:* Delivery service providers can optimize their logistics and resource allocation based on predicted delivery times, leading to cost savings and increased efficiency.

- *Real-Time Adaptation:* Machine learning models can adapt to changing conditions in real-time, such as traffic congestion or adverse weather, to provide up-to-date delivery time estimates.

Online delivery time prediction using machine learning offers a data-driven solution to a critical aspect of the delivery process. By harnessing the power of historical data and real-time information, machine learning models can provide accurate and timely predictions, ultimately enhancing the customer experience and operational efficiency of online delivery services.

.Literature Review:

Paper[1]

The research paper focuses on the integration and management of the performance management (PM) process of on-time delivery with suppliers in manufacturing companies. The study aims to analyze the perceived importance of integrating this process, describe how it is managed, and compare integration and management issues between companies with high and low perceived on-time delivery performance.

Key points from the paper include:

Managing relationships with suppliers involves integrating core operational processes like procurement and order fulfillment in line with strategic objectives. Previous studies highlight the importance of identifying which processes to

integrate with supply chain partners and the level of integration required.the study found that a lack of understanding of the benefits of integrating PM with supply chain partners could be identified.The research methodology involved a survey questionnaire with 20 questions, data collection from Swedish manufacturing companies, and statistical analysis using SPSS software.Findings revealed differences in the degree of integration of PM activities related to on-time delivery, with the measurement activity being significantly less integrated. Companies with high perceived on-time delivery performance showed differences in issues such as manual data collection and report generation compared to those with low perceived performance.

Limitations of the study include potential biases in data collection and the focus on Swedish manufacturing companies.

Overall, the paper emphasizes the importance of integrating the PM process of on-time delivery with suppliers and highlights key differences in management and integration practices between companies with varying levels of on-time delivery performance.

Paper[2]

The study identified various organizational, people, process, project, and technical factors that impact on-time delivery, such as requirements refinement, task dependencies, organizational alignment, and historical delivery performance. Hierarchical interactions among these factors were discovered, highlighting the complex relationships within software development projects.

Practitioners are advised to consider a combination of expert- and data-based selection methods to gain detailed insights and improve understanding of their software development processes. The document emphasizes the importance of accurate effort estimation for reducing delays, enhancing customer satisfaction, and optimizing resource allocation in software projects.

In the realm of software development effort estimation and on-time delivery, a plethora of studies have delved into understanding the influential factors that impact project success. Researchers have employed various methodologies to extract insights from experts and practitioners in the field.

Studies have emphasized the importance of accurate estimation in software projects, highlighting the need to consider factors such as organizational environment, user requirements, project complexity, and team dynamics. Efforts have been made to quantify and model these factors using statistical analysis and software repository data to enhance prediction accuracy.

Furthermore, research has explored the interplay between different factors, uncovering hierarchical interactions among organizational, people, process, project, and technical elements. By integrating qualitative and quantitative approaches, scholars have strived to provide a comprehensive understanding of the multifaceted nature of software project management.

Overall, the study provides a conceptual framework for practitioners to identify and manage risks associated with on-time software delivery, offering actionable insights to address the root causes of delays and improve project predictability in agile settings. the literature underscores the significance of factors affecting on-time delivery in agile software development, offering valuable insights for practitioners to improve project planning, estimation, and execution.

Paper[3]

This document presents research conducted by a team of experts from various institutions, including Nanyang Technological University in Singapore and JD Intelligent Cities Research in Beijing. The study was published in the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, showcasing innovative approaches to improving

delivery services.The key findings of the research on delivery time inference based on couriers' trajectories include:

Challenges in Delivery Time Inference: The study identified two main challenges in inferring delivery time based on couriers' trajectories: inaccurate delivery locations and various stay scenarios [T1], [T3].

Proposed Solution - DTInf: The researchers proposed a solution called Delivery Time Inference (DTInf) to automatically infer the delivery time of waybills based on couriers' trajectories. This solution consists of three main steps: Data Pre-processing, Delivery Location Correction, and Delivery Event-based Matching [T3].

System Deployment: An automated system based on DTInf was developed and deployed internally in JD Logistics, demonstrating the practical application of the proposed solution in real-world logistics operations [T3].

Effectiveness Confirmation: Extensive experiments and case studies based on large-scale real-world waybill and trajectory data from JD Logistics confirmed the effectiveness of the DTInf approach in inferring delivery times accurately [T3].

Value for Couriers and Logistics Companies**: The research highlighted the value of inferring delivery times automatically for both couriers and logistics companies, as it can streamline operations, improve delivery performance evaluations, and enhance customer satisfaction levels [T2], [T3].

These findings underscore the importance of leveraging couriers' trajectories to enhance delivery efficiency and accuracy in the logistics industry.

Paper[4]

South African Journal of Industrial Engineering December 2017 issue! In this article, G.M Ramachandran and S. Neelakrishnan present an approach to improving customer on-time delivery against the original promised date in a low-volume high-variety product manufacturing industry. By incorporating lean thinking methodology, a lean live tracking tool, and cross-functional team approaches, they were able to significantly improve on-time delivery.

The real-time data from an industrial valve manufacturing LVHV firm demonstrated the effectiveness of the approach in improving on-time delivery by showing a significant improvement in on-time delivery performance over a period of eight months. The average on-time delivery (OTD) percentage increased from 30 percent to 90 percent during this timeframe [T5]. This improvement was achieved through the implementation of the lean live tracking (LLT) tool, which enabled better tracking of products from raw material input to final product shipment to customers. The LLT system enhanced visibility in the LVHV industry, leading to improved supplier on-time receipt (OTR) and subsequently influencing an enhancement in on-time delivery (OTD) performance [T3].

Furthermore, the LLT system captured and tracked critical processes such as TPI completion, production test completion, product assembly completion, material feeding to assembly completion, shortage reporting for matched set components, and purchase order status updates. By streamlining these processes and reducing manual effort, the LLT system helped the firm achieve on-time delivery against the original promised date (OPD) more efficiently [T3]. The success of the approach was evident in the data reflecting the improved OTD percentage and the reduction in fines to be paid, indicating a tangible impact on operational performance and customer satisfaction [T3].

Paper[5]

PROPOSED MODEL

About dataset used

Delivery time prediction has long been a part of city logistics, but refining accuracy has recently become very important for services such as Deliveroo, Foodpanda and Uber Eats which deliver food on demand.

These services must receive an order and have it delivered within 30 minutes to appease their users. In these situations +/- five minutes can make a big difference so, for customer satisfaction, it’s important that the initial prediction is accurate and that any delays are communicated effectively.

In this article, I’ll discuss my experience building a real-world delivery time prediction model for a food delivery startup and how it came to give better predictions than our trained operations team. I’ll touch on the technical topics of machine learning while

focusing on the business knowledge required to create a well-functioning predictive model.

Problem statement:

The deliveries are large. We can assume they require cars.

The supplier (or suppliers) range from restaurants to caterers and can have very different characteristics.

Let's assume drivers arrive at the supplier at exactly the dictated pick-up time. In reality this isn't always true, but with a sufficient number of drivers, aberrations can be kept to a minimal amount.

Food should not be delivered too late as the customer will be left waiting and angry; nor can it arrive too early as it will have to sit out before the customer is ready to eat it.

Model Building

Let's think about what is involved in food delivery. We can break it down into three principal components:

1. Pick up the food from the supplier.
2. Drive from the supplier to the customer.
3. Drop off the food to the customer

Alternatively, we could train a model to predict the entire delivery time. The reason we should break this up is to create *specialty* models which will have superior ability over one generic model. If we think about the models as people it makes sense. In your business development team you would have a person who knows the suppliers really well and how long they might take for delivery. In your client service team you would have someone who knows each client well and can predict how long the delivery might take based on their location and building. We're structuring our process with this in mind.

Data preprocessing:

Handle missing values
Select relevant features
Scale numerical features
Encode categorical variables
Address outliers if needed
Split dataset into training/testing sets
Normalize target variable if necessary
Ensure balanced data distribution

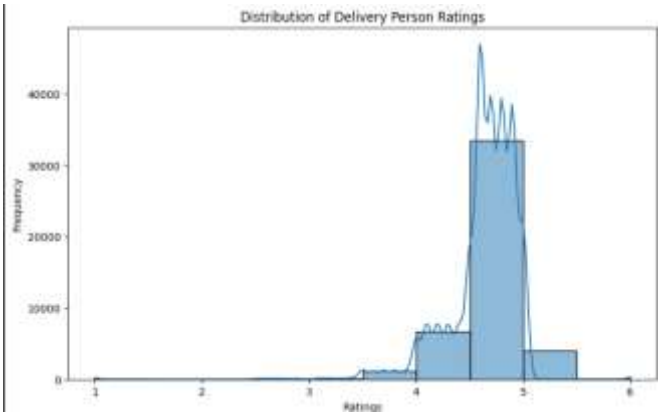


Fig-1: Rating

Fig-2: Order type

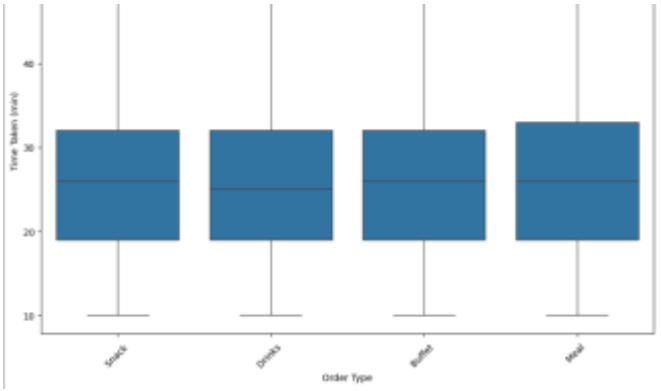


Fig-3: time taken to deliver certain order

ALGORITHMS USED:

Decision tree regression: is a machine learning algorithm commonly used for predicting continuous values. It's a type of supervised learning algorithm that works by partitioning the feature space into regions and fitting a simple model (like a constant value) in each region. In the context of predicting delivery times, decision tree regression can be a valuable tool.

Here's how decision tree regression works in the context of predicting on-time delivery:

Dataset Preparation: You start with a dataset containing features that might influence delivery times, such as distance, time of day, traffic conditions, weather, etc. Each data point in your dataset should also include the actual delivery time.

Training the Model: The decision tree algorithm is then trained on this dataset. During training, the algorithm recursively splits the data based on the features that result in the best reduction of variance in the target variable (delivery time in this case). It continues splitting until a stopping criterion is met, such as reaching a maximum depth or having too few data points in a node.

Predicting Delivery Time: Once trained, the decision tree can be used to predict delivery times for new data points. When you have a new delivery to make, you input the relevant features (distance, time of day, etc.) into the trained decision tree, and it traverses the tree based on the feature values to arrive at a predicted delivery time.

Evaluation: The performance of the decision tree model can be evaluated using metrics like mean squared error (MSE), mean absolute error (MAE), or R-squared value on a separate test dataset.

Decision tree regression has several advantages, including:

Interpretability: Decision trees are easy to interpret and visualize, which can be helpful in understanding the factors influencing delivery times.

Non-linearity Handling: Decision trees can capture non-linear relationships between features and the target variable.

Robustness to Outliers: Decision trees are robust to outliers and can handle them well without significantly affecting the model's performance.

However, decision tree regression also has limitations:

Overfitting: Decision trees can easily overfit the training data, especially if the tree is allowed to grow too deep. Regularization techniques like pruning or using ensemble methods like Random Forests or Gradient Boosted Trees can mitigate this issue.

Instability: Small variations in the data can lead to different decision trees, which can make the model unstable.

In practice, decision tree regression is often used as part of more complex models or in combination with other algorithms to improve performance.

Support Vector Regression (SVR): is a machine learning algorithm used for predicting continuous values, making it applicable to on-time delivery time prediction datasets. Here's a brief overview of how SVR works:

Kernel Trick: SVR uses a kernel function to transform the input features into a higher-dimensional space. This transformation allows SVR to find a non-linear decision boundary in the original feature space.

Margin Maximization: Like Support Vector Machines (SVM) for classification, SVR aims to maximize the margin between the predicted values and the actual target values while ensuring that deviations (errors) are within a specified threshold.

Loss Function: SVR minimizes the error between the predicted and actual values, typically using a loss function such as epsilon-insensitive loss or squared loss.

Regularization: SVR includes a regularization parameter to control the trade-off between maximizing the margin and minimizing the error. This helps prevent overfitting by penalizing overly complex models.

Prediction: Once trained, SVR can be used to predict delivery times for new data points by finding the hyperplane that best fits the training data within the specified error margin.

K-Nearest Neighbors (KNN) regression: is a simple yet effective algorithm for predicting continuous values, such as delivery times in on-time delivery prediction datasets. Here's how it works:

Data Representation: The dataset consists of feature vectors representing various factors that may influence delivery times (e.g., distance, time of day, weather conditions).

Nearest Neighbor Search: When predicting the delivery time for a new instance, KNN finds the K nearest data points in the training set based on a distance metric (e.g., Euclidean distance) between the feature vectors. These data points are the "neighbors."

Aggregation: The predicted delivery time is then calculated as the average (or weighted average) of the delivery times of the K nearest neighbors.

Parameter Selection: The choice of K (the number of neighbors) is a crucial parameter in KNN regression. It determines the trade-off between bias and variance in the model. Smaller values of K lead to more flexible models but may be prone to overfitting, while larger values of K may lead to underfitting.

Random Forest Regression: is a robust and versatile machine learning algorithm used for predicting continuous values, such as delivery times in on-time delivery prediction datasets.

Ensemble of Decision Trees: Random Forest Regression builds multiple decision trees during training. Each tree is constructed using a random subset of the training data and a random subset of the features. This randomness helps to decorrelate the individual trees and reduce overfitting.

Prediction: When making predictions, Random Forest Regression aggregates the predictions of all the individual trees to obtain the final prediction. For regression tasks, this typically involves averaging the predictions of all the trees.

Feature Importance: Random Forest Regression provides a measure of feature importance, indicating which features have the most significant impact on the prediction. This can be valuable for understanding the factors driving delivery times.

Handling Non-linearity and Interactions: Random Forest Regression can capture complex non-linear relationships and interactions between features and the target variable, making it suitable for datasets with intricate dependencies.

Robustness to Overfitting: Random Forest Regression is less prone to overfitting compared to individual decision trees, thanks to the ensemble approach and the randomness injected during training.

Hyperparameter Tuning: Random Forest Regression has several hyperparameters that can be tuned to optimize performance, such as the number of trees in the forest, the maximum depth of each tree, and the number of features considered at each split.

Random Forest Regression: is a robust and versatile machine learning algorithm used for predicting continuous values, such as delivery times in on-time delivery prediction datasets. Here's how it works:

Ensemble of Decision Trees: Random Forest Regression builds multiple decision trees during training. Each tree is constructed using a random subset of the training data and a random subset of the features. This randomness helps to decorrelate the individual trees and reduce overfitting.

Prediction: When making predictions, Random Forest Regression aggregates the predictions of all the individual trees to obtain the final prediction. For regression tasks, this typically involves averaging the predictions of all the trees.

Feature Importance: Random Forest Regression provides a measure of feature importance, indicating which features have the most significant impact on the prediction. This can be valuable for understanding the factors driving delivery times.

Handling Non-linearity and Interactions: Random Forest Regression can capture complex non-linear relationships and interactions between features and the target variable, making it suitable for datasets with intricate dependencies.

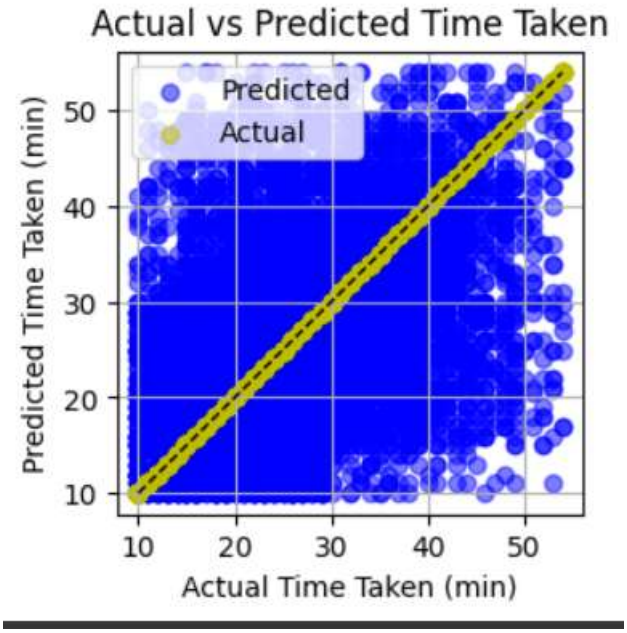
Robustness to Overfitting: Random Forest Regression is less prone to overfitting compared to individual decision trees, thanks to the ensemble approach and the randomness injected during training.

Hyperparameter Tuning: Random Forest Regression has several hyperparameters that can be tuned to optimize performance, such as the number of trees in the forest, the maximum depth of each tree, and the number of features considered at each split.

RESULTS:

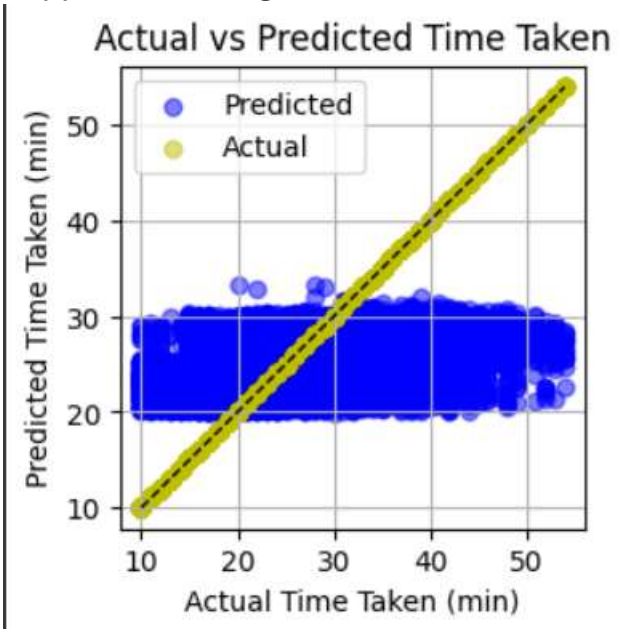
PREDICTIONS

Decision tree regression

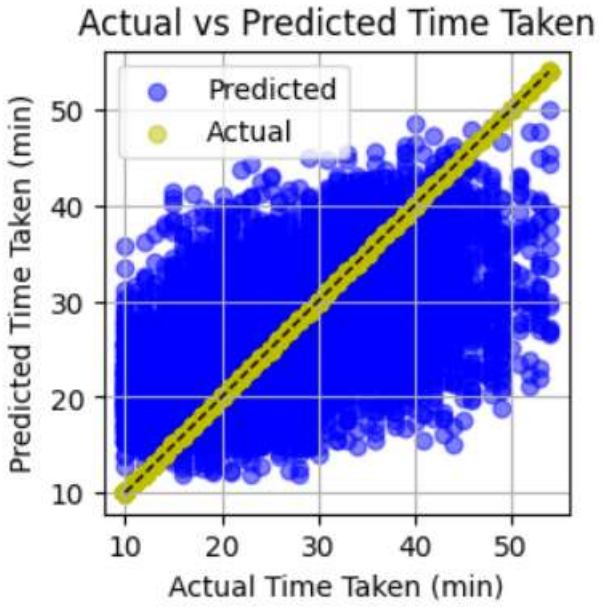


Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS 2023) IEEE
Xplore Part Number : CFP23CB2-ART ; ISBN : 979-8-3503-2579-9

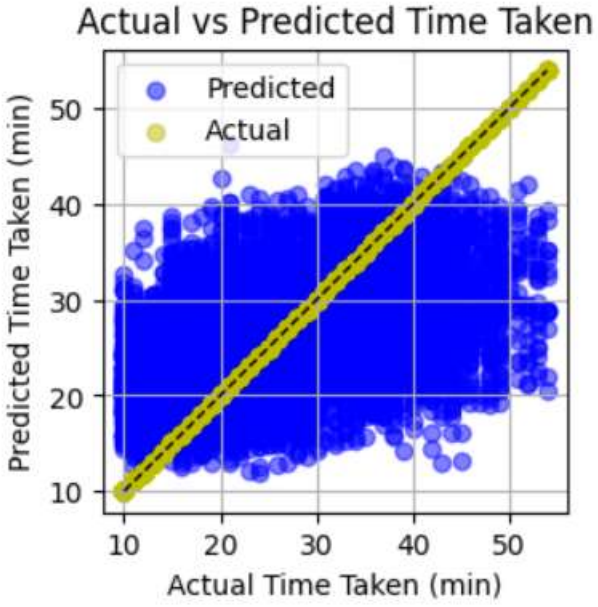
Support vector regression



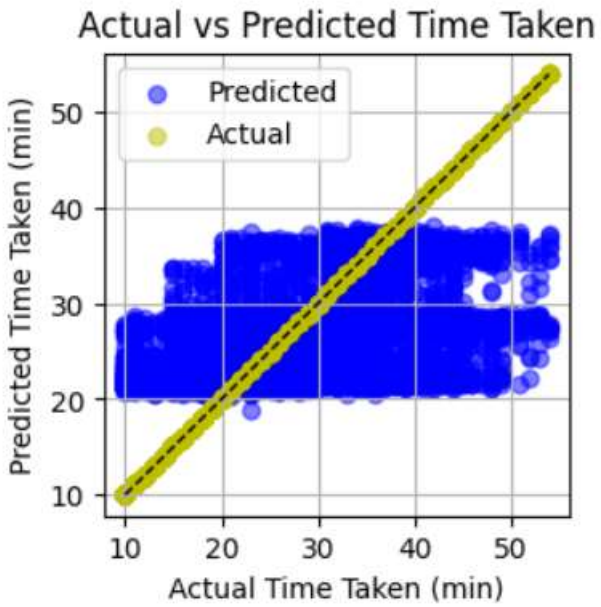
Random Forest Regression



K-Nearest neighbour's(KNN) Regression



Neural Network Regression



CONCLUSION:

The provided code implements a variety of machine learning algorithms to tackle the task of on-time delivery time prediction classification. Each algorithm is briefly explained, along with its application in the code. Here's a concise conclusion:

The code applies logistic regression, decision tree classifier, k-nearest neighbors (KNN), random forest classifier, support vector machine (SVM), naive Bayes classifier, multi-layer perceptron (MLP), XGBoost classifier, and LightGBM to solve the on-time delivery time prediction classification problem. These algorithms are chosen based on their suitability for binary classification tasks and their performance across different types of datasets.

The logistic regression model is a simple yet effective linear classifier, while decision trees offer interpretability and flexibility in modeling complex decision boundaries. KNN utilizes the similarity between instances to make predictions, while random forest builds an ensemble of decision trees to improve prediction accuracy.

SVM finds the hyperplane that best separates classes in the feature space, while naive Bayes relies on probabilistic reasoning and feature independence assumptions. MLP, XGBoost, and LightGBM leverage neural networks and gradient boosting techniques to capture complex patterns in the data and achieve high predictive performance.

In conclusion, the provided code demonstrates the versatility of various machine learning algorithms in solving the on-time delivery time prediction classification problem, catering to different requirements and characteristics of the dataset.

REFERENCES

[1] H. M. Z. A. Shebli and B. D. Beheshti, "A study on penetration testing process and tools," 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 2018, pp. 1-7, doi: 10.1109/LISAT.2018.8378035.

[2] K. Patel, "A Survey on Vulnerability Assessment & Penetration Testing for Secure Communication," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 320-325, doi: 10.1109/ICOEI.2019.8862767.

[3] Mohammed Ali Shaik, MD.Riyaz Ahmed, M. Sai Ram and G. Ranadheer Reddy, (2022), "Imposing Security in the Video Surveillance", International Conference on Research in Sciences, Engineering & Technology, AIP Conf. Proc. 2418, 020012-1– 020012-8; https://doi.org/10.1063/5.0081720,

[4] T. Walter, "Architectural Pen-Test Generation and Vulnerability Prediction for Cyber-Physical Systems," 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C), Honolulu, HI, USA, 2022, pp. 45-46, doi: 10.1109/ICSAC54293.2022.00016.

[5] Ö. Aslan and R. Samet, "Mitigating Cyber Security Attacks by Being Aware of Vulnerabilities and Bugs," 2017 International Conference on Cyberworlds (CW), Chester, UK, 2017, pp. 222-225, doi: 10.1109/CW.2017.22.

[6] Mohammed Ali Shaik, "Time Series Forecasting using Vector quantization", International Journal of Advanced Science and Technology (IJAST), ISSN:2005-4238,Volume-29,Issue-4 (2020), Pp.169-175.

[7] N. Koroniotis, N. Moustafa, B. Turnbull, F. Schiliro, P. Gauravaram and H. Janicke, "A Deep Learning-based Penetration Testing Framework for Vulnerability Identification in Internet of Things Environments," 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Shenyang, China, 2021, pp. 887-894, doi: 10.1109/TrustCom53373.2021.00125.

[8] M. A. Shaik, S. k. Koppula, M. Rafiuddin and B. S. Preethi, "COVID19 Detector Using Deep Learning," 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022, pp. 443-449, doi: 10.1109/ICAAIC53929.2022.9792694.

[9] S. Bera, L. Glenn, A. Raghavan, E. Meno, T. Cody and P. A. Beling, "Deterring Adversarial Learning in Penetration Testing by Exploiting Domain Adaptation Theory," 2023 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2023, pp. 314-318, doi: 10.1109/SIEDS58326.2023.10137792.

[10] Sahu, S., & Deo, R. C. (2017). Penetration testing and its methodologies: A review. Journal of Information Security, 8(2), 111- 120. doi: 10.4236/jis.2017.82009

[11] M. A. Shaik, R. Sreeja, S. Zainab, P. S. Sowmya, T. Akshay and S. Sindhu, "Improving Accuracy of Heart Disease Prediction through Machine Learning Algorithms", 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Uttarakhand, India, 2023, pp. 41-46, doi: 10.1109/ICIDCA56705.2023.10100244.

[12] Verma, N., & Bhaskar, P. (2018). Penetration testing: An approach towards securing computer systems. International Journal of EngineeringandTechnology,7(4.18),63-66.doi:10.14419/ijet.v7i4.18.22775.