IMPROVING UNLOADING TIME PREDICTION THROUGH DRIVER AND CUSTOMER SEGMENTATION

Liyadipita Appuhami Mudiyanselage Ranula Prasanna Bandara Liyadipita

(209352E)

Master of Science in Computer Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

July 2022

IMPROVING UNLOADING TIME PREDICTION THROUGH DRIVER AND CUSTOMER SEGMENTATION

Liyadipita Appuhami Mudiyanselage Ranula Prasanna Bandara Liyadipita

(209352E)

Thesis/Dissertation submitted in partial fulfilment of the requirements for the degree Master of Science in Computer Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

July 2022

DECLARATION

I declare that this is my work and this dissertation does not incorporate without acknowledgment any material previously submitted for the Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to the University of Moratuwa, the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic, or another medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the Master's dissertation under my supervision.

Signature of the supervisor: Date:

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to all those who provided support to make my research on "IMPROVING UNLOADING TIME PREDICTION THROUGH DRIVER AND CUSTOMER SEGMENTATION" successful.

First of all, I would like to express my gratitude to my project supervisor Dr. Uthayashanker Thayasivam, Senior Lecturer, Department of Computer Science and Engineering. I am highly indebted to him for his guidance and constant supervision as well as for providing necessary information regarding the project and for his support in completing the project successfully.

I am sincerely thankful to the final year project coordinator Dr. Charith Chittaranjan, Senior Lecturer, Department of Computer Science and Engineering for the support given throughout the project time period. Further, I would like to extend my gratitude to Dr. Nisansa De Silva for participating in evaluations and providing me with very useful guidance to make my research successful.

Specially I would like to thank Prof. Indika Perera, Head of Department, Department of Computer Science and Engineering for his assistance and coordination to conduct the research without any issues during the final year.

Finally, I wish to thank the academic and non-academic staff of the Department of Computer Science and Engineering and colleagues for the support and encouragement given.

ABSTRACT

Modern-day society is driven by transportation networks. Now it is easier than ever to order your daily necessities through online platforms. The research interest in this thesis focus on the delivery aspect lies with ordering. A successfully completed delivery means a properly addressed vehicle routing problem. The data set that is involved in the study refers to a large amount of perishable good cases that are delivered through large trucks. Each truck caters to 8-10 customers in a day. Since these are perishable goods delivered to people in the foodservice industry, they expect a sound ETA of their delivery to plan ahead for meal preparations.

To provide an ETA in a multi-stop route there are two variables to be solved. One is the travel time between stops, which modern-day map services would output without a hassle. However, the next important thing is the unloading time needs to calculate with the historical data. The study suggests a way to involve customer profiling and driver profiling so that unloading time prediction can be done with those two variables along with the delivery volume of the stop.

Modeling these two variables into a regression model was a challenge on its own due to their large dimension of them. Segmentation of the said variables and using segment mean yielded better results in regression compared to using a label encoding technique blindly which introduced an orderly nature to features from the id itself. Furthermore, once segment means were clustered based on their distribution and provided a cluster identifier that justifies the orderly nature, models were able to yield their least MSE.

Finally, this study highlights the importance involving of the customer site and the driver's experience in the unloading time. Also, this study has presented a way of representing such variables with a high cardinality in a meaningful manner so that model can be built with less error. This will provide a good starting point for further analysis on similar research interests in the future

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Transport optimization is the most critical element in good distribution companies. In the process of finding optimal routes and providing estimated arrival times for customers, it is necessary to provide accurate information such as customer location, nature of the goods, vehicle fleet, warehouse, delivery limitations, etc. Even though much of the above information can be found through order details it is with utmost importance to predict the time of unloading for each of the customers in order to provide an estimated time of arrival. This estimate depends mainly on the volume of the order, the nature of the unloading location (customer), and the experience of the unloader (driver).

In the practical scenarios, from a data perspective, the customer site and the driver is only limited to a unique identifier (UUID). It becomes tricky to use this valuable information in a unload time prediction solution due to the large number of elements that belong in the respective categories. Introducing a classical encoding technique such as one-hot encoding to these features is not viable due to the large number of different customers and drivers involved. Hence it is necessary to categorize those UUIDs in a meaningful manner to gain better results during the prediction.

Regression models are often utilized for the prediction of a continuous value in machine learning. The output of a regression model depends mainly on the efficiency and effectiveness of the engineering of its features. This study aims at identifying a significant procedure to categorize the customers and drivers utilizing a novel mechanism and to compare such mechanisms with the known mechanisms in the literature.

1.1 Research Problem

Unload time prediction is a subproblem of the vehicle routing problem (VRP)[1]. In most academic research in the area of tackling VRP problems, the 'unloading time' parameter is assumed to be known in advance. But it is not a realistic scenario in a real-life situation. However, A method for estimating the time of unloading has been proposed in [2], [3], [4], and [5]. This is determined by the number of things purchased, the quantity, and the volume of the customer's order. [4] and [5] highlight how these parameters affect real-world conditions and how GPS data can be used to improve the predictions while [5] uses machine learning models and [4] uses mathematical models. All those studies stress the importance of the unloading site (customer) and the experience of the person (driver) who unloads goods, for the prediction, yet has not used those parameters. The cardinality of these features is extremely high, which poses a challenge. The challenge at hand is to make effective use of those two parameters by grouping a large number of drivers/customers into meaningful categories and feeding

the segment identifier into a machine learning model to boost predictions alongside the most widely used parameters.

1.2 Challenges

Even if this is a well-known problem, these problems are often faced by fleet management companies that are directly involved in customer satisfaction. The majority of these fleet management companies don't keep track of their past delivery reports. With that nature, it is challenging to find similar data collections and also research components that are directly aligned with the specific research problem addressed in the study.

To receive accurate data from a routing problem it is important to have GPS data collection devices, scanning devices configured correctly. Stakeholders directly involved in these distribution centers tend to be not as technologically advanced at times to have the correct configurations. This causes data tracking to be inaccurate and it heavily affects the data collection. Further to this, due to not having proper mobile network coverage during the distribution process, some of the data points can go missing which affects the quality of the data collection.

However, the most critical challenge of them all is not having a straightforward way of knowing the difference between the actual event happening time and the data recorded time. This data collection is completely dependent on the instruments and there is no way to get a verification from the end-user about the accuracy of the recorded values.

Another challenge related to this specific problem is known widely in the problem domain as the curse of dimensionality. There are a higher number of customers and trucks involved in operations for this data set and with that data, volume is high along with the dimensions. So it is important to understand that solving a similar problem in a less complex fleet management system would have been addressed in a less complicated way.

1.3 Research Objectives

- Find the right sample of data for the analysis from the vast data collection available.
- By measuring the average time of unloading, determine the categorizing mechanism for grouping drivers based on experience and unloading sites based on toughness.
- Evaluate the segmentation by predicting unloading time and comparing the results with uncategorized data fed to the same ML models.
- Publish the findings.

1.4 Contributions of Research

- Refined data set of ~9million rows in VRP domain which can be a starting point for many research interests in the future.
- A method of involving unloader experience and site difficulty in the prediction can further be used in similar research problems.
- Comparison of encoding mechanisms that are commonly used in data science.
- Comparison of regression models with most widely used evaluating methods in the literature.

2 LITERATURE REVIEW

Since logistics progressed within the 1950s [1], various researches were carried out inside different application spaces. The importance of logistic administration has been growing in many regions as a result of the inclination of state ownership and internationalization in later decades. Logistics provided help to firms in improving production and distribution based on the same assets through administration strategies designed to increase productivity and competitiveness. The transportation structure, which connects isolated exercises, is the most basic component of a logistic chain.

In the first part of the literature review, a detailed analysis of the VRP will be conducted reviewing the problem and how solutions for VRP evolved in the last 5 decades while focusing on the several parameters that determine the solutions. The second part of the review will focus on how the parameter unloading time is determined in the literature. Finally, the encoding techniques will be analyzed from the literature for categorical variables in general in order to evaluate the researchers' outcomes later.

2.1 Vehicle routing problem

Transportation expenditures account for 33% of logistics costs, and transportation systems have a big influence on logistics efficiency. Transportation is required throughout the production process, from product development to distribution to end customers, and even during product returns. Only if each variable was well-coordinated could the advantages be maximized. Without well-developed transport systems, logistics would be unable to fully realize its benefits. A system like this in logistics could improve performance, reduce operating costs, and improve service quality. A company's transportation operations will be considerably boosted if vehicle routing concerns are effectively resolved.

2.1.1 The problem and the solving approaches

The traveling salesman problem [6] is one of the world's most researched optimization challenges. The vehicle routing problem can be shown as a generalization of the said problem. The problem concerns a traveling salesman who is tasked with visiting a few cities in the shortest possible way, since each city is only visited once, and the departure city must also be the arrival city. To define the starting state of the vehicle routing problem, we have to consider TSP with more than one traveling salesman and have all of them in one city. The purpose is to discover the shortest possible paths for these salesmen so that just one of them visits each city. As [7] suggests that there are two major characteristics that differ VRP from TSP,

1. In comparison, no other algorithms have been shown to be as successful in tackling the VRP problem.

2. VRP is a more practical problem that occurs in practice

According to [1] VRP is viewed as the problem of the route designing in a way such that geographically scattered customers are provided with deliveries at the least cost possible. Paper suggests that various different practical issues along with the legal regulation in the distribution management spawn several variants of the existing problem.

As [8] shows, much advancement has been achieved in the field of vehicle routing over the last 50 years. It has caught the interest of the scientific world, and several of the disclosed algorithms have made their way into commercial solvers. Following four main approaches to this development are broadly discussed in this paper while analyzing the pros and cons of each in this research,

- Branch-and-Bound Algorithms
- Dynamic Programming
- Commodity Flow Formulations and Algorithms
- Set Partitioning Formulations and Algorithms

2.1.2 Parameters involved in VRP

As the customer base grows, the VRP problem becomes more complex. Paper [2] specifies the following set of features to be the most essential meaningful and standard parameters. These measures substantially increase the number of methodologies, models, and algorithms available for analyzing massive datasets.

Table 1:VRP constrains

Real-world constraints	Standard constraints
Customer's time window Time of good unloading Good packaging into vehicles The predefined capacity of the vehicles Fixed and variable vehicle costs	The number of Depots of the logistic company Maximum allowable timing or the length of the vehicle route Different vehicle capacities Time windows for beginning and finishing customer service as well as vehicle time windows. Customers' demands for delivery or collection of a certain amount of cargo during the service

The paper also claims that the VRP problem is made up of three data sets in total. They are the depot, the trucks, and the customers (users). The authors provide a comprehensive data overview for each of these datasets. According to the depot, at least one depot must be identified in the issue, and important data below must be collected.

- The location consists of address, postal code, latitude, and longitude.
- The open and closure are included in the business hours.

If there are many depots from which deliveries are made and clients can be supplied from all of them, the problem can be characterized as a Multiple Depot Vehicle Routing Problem (MDVRP). If clients are tied to a single depot, however, additional distinct VRP concerns for each depot and its purchasers must be modeled. Vehicles are generally included in several depot difficulties, according to the authors.

Vehicle parameters are as follows according to the paper.

- Loadspace capacity
- Number of pallet positions
- Number of cargo units
- Driver's working time
- Departure location
- Arrival location

Furthermore, the authors demonstrate that cargo space is limited due to a variety of circumstances. When distributing goods, for example, it's important to keep in mind the cargo space's limited capacity and volume so that it can accommodate all of the goods being transported. Certain products are light, but they take up a lot of space, so both constraints must be met. Pallets are also used to load products such that the vehicle's capacity includes the number of pallets that can be loaded into the hold. The simplest case is to deliver products of the same dimensions, allowing the ability to be expressed in terms of the maximum number of parts that can be stored in the hold.

Customers' data usually include the following according to the paper,

- Location
- Geographic position
- Order
- Time limits

Considering all 3 major factors, then they have identified the target attributes affecting the given routes and total cost. The table below shows the features of VRP algorithms

Table 2: VRP Parameters

Control Parameter(s)	Description
ToleranceWeight ToleranceVolume	For the settings of ToleranceWeight and ToleranceVolume, the vehicle can be allowed to be loaded in weight and volume during the application of VRP algorithms.
PenaltyDelay	Algorithms for solving VRP problems allow the vehicle to be delayed for the customer (to arrive outside of its time window). The PenaltyDelay displays the infringement of this parameter.
PenaltyCustomersVehicles	If the VRP algorithm is adjusted for solving Site-Dependent Vehicle Routing Problem (SDVRP), the attribute PenaltyCustomersVehicles is used for the penalization of rules violations, where the customer can't be served by a particular vehicle.
CostIncreasing	The unchanging factor used in increasing the cost, depending on the weight of vehicle transports, is presented with the CostIncreasing.
PenaltyVolumePercentage PenaltyWeightPercentage	The constants that penalize reloading of vehicles by weight or volume are PenaltyWeightPercentage and PenaltyVolumePercentage, and they present the cost increasing when the weight/volume of the vehicle is reloaded by 100%.

The novel method of fitting these parameters using prediction models based on available historical data is presented in this article. Four prediction models were used, and the SVM algorithm was found to have significantly better and superior performance in all tests as compared to the other models. In the case of the SVM model, the predictive precision is 90% for each of the control parameters examined. The benefit of SVM over other methods used, according to the authors, is that it provides better predictions on hidden test data, unique optimal solutions to the training problem, and fewer optimization parameters than other methods. Regardless of the accuracy, they admit that it is not the fastest algorithm they have created.

[3] claims that various statistics suggest that, in order to put their transportation and logistics plan into action, it's vital to know the number of clients served and whether the sold/delivered items are parcels or parts. They argue that these have a direct impact on how goods are stored and loaded into vehicles due to various natural constraints such as,

- Customers are frequently forced to use the same automobile due to weather or other circumstances.
- Unloading time duration in delivery stops
- Route cost calculation
- Type of the vehicle and loading constraints
- Limits on a vehicle's and a driver's maximum service time are set by law.

Their method, which is focused on areas, cannot, however, take into account all of the aforementioned considerations when designing transportation routes. Most available software solutions, according to the authors, work on the same principle, with the ability to further connect neighbour areas that are on a similar path. This grouping method has the drawback of not being able to solve complex problems. As a result, the solution [3] is divided into two parts: (i) Solving a complex real-world VRP in logistics using the conceptual modeling multi-step algorithm while meeting all of the practical constraints, and (ii) Adaptively adjusting the parameters and constants of the given model and algorithm using historical data.

[9] provides a unique perspective on VPR constraints by applying sustainability criteria to the policy framework for urban freight, which necessitates the calculation of all expenses and downsides. The emphasis of this paper is on the issue of entering time frame constraints, which prevent freight vehicles from entering central busy areas in many European cities. The authors claim despite the fact that this measure aims to minimize congestion and pollution at the busiest times of the day, it comes at a cost and causes increased pollution and power consumption. A mathematical model for the VRP with Access Timeframes, a form of the VRP ideal for designing delivery routes in a municipality with this type of mobility constraint, is presented in this paper. Based on a case study, they used the model to find an optimal solution to small problem scenarios, then compared the output of a customized savings technique, a genetic algorithm, and a metaheuristic local search method used for mathematical optimization over larger instances, finding no consistent occurrence of any of them but confirming the importance of those added expenses and economic effects.

2.2 Unloading time prediction

The anticipated time for unloading the products is depending on the number of items requested and the complexity of the site (parking time, customer admin work, etc.).

These are the most important constraints to consider when modeling a real-world VRP problem. Depending on the sophistication of the supply chain, a number of constraints may arise in practice. The majority of data is now processed in the firm's information management, allowing data preparation and problem modeling to be automated.

One of the most challenging elements to precise forecast in realistic conditions is the unloading time for each customer. The defined parameter is calculated in two ways in the definition mentioned in the paper [4].
1. On the basis of a study of experiments,
2. On the basis of gathered previous orders and assessment of Geolocation collected data, with a correction factor. The equation can be used to indicate the initial unloading time.

$$Unloadinime_k = UnloadingConst + \frac{NA_k}{4} + \frac{TV_k}{0.2} + \frac{TW_k}{100}, \qquad (1)$$

whereby it is:
NA_k – Number of items of customer k,
TV_k – Total ordered volume (all items) of customer k,
TW_k – Total ordered weight (all items) of customer k,
$UnloadingConst$ – A predefined constant.

Figure 1: Unloading time calculation equation

After calculating the unloading time, the correction factor is applied. Using the previously available n past data for customer k in this study, the correction coefficient for client k is derived as follows.

$$CorCoeff_k(NA_k, TV_k, TW_k) = SIGN \cdot \frac{UnloadingTime_k \cdot \frac{NA_k}{4} \cdot \frac{TV_k}{0.2} \cdot \frac{TW_k}{100}}{\frac{1}{n} \cdot \sum_{i=1}^{n} \left((UnloadingTime_k)_i \cdot \frac{NA_i}{4} \cdot \frac{TV_i}{0.2} \cdot \frac{TW_i}{100} \right)} \qquad (2)$$

The sign of the given coefficient $SIGN$ is calculated in the following way:

$$sign\left(\frac{\sum_{i=1}^{n} \left((UnloadingTime_k)_i \cdot \frac{NA_i}{4} \cdot \frac{TV_i}{0.2} \cdot \frac{TW_i}{100} \right)}{n} - UnloadingTime_k \cdot \frac{NA_k}{4} \cdot \frac{TV_k}{0.2} \cdot \frac{TW_k}{100} \right) \qquad (3)$$

The unit of measure for these formulas is [min].

Figure 2: Unloading time calculation coefficient equation

The paper [5] offers an end-to-end pipeline for predicting unloading time. The paper claims that expected behavior is realized in four phases, which are then assigned to a separate module consisting of data preparation, model building, prediction, and

operation. The data preparation phase is in charge of retrieving necessary data from the database. The following attributes are recorded per each attribute in their approach.

- customer ID,
- total weight and volume,
- total article
- time of unloading.

A distribution has a one-to-one mapping to a customer's order in the data set of interest to the authors. Each order includes one or more posts, each with its own weight and length. The total weight of the delivery is the summation weights of all goods in the order. In the same way, the total volume calculation is performed. Unloading time is calculated in this study using GPS history info. Furthermore, since some models require that all input variables be similarly scaled, normalization must be performed before the model evaluation. During the development process, several well-known normalizing approaches were examined, supporting the notion that this selection has no significant impact on model outcomes. Because of its resistance to outliers, a system based on quartiles was chosen. The inability to entirely eliminate outliers is related to the difficulty of detecting them leading to a shortage of sufficient data points per customer to assess the distribution of input variables.

Each customer has their own model for predicting unloading time in this study. However, a big number of clients in the database slows down the process. For a given customer ID, number of articles, total mass, and quantity of delivery, the research's prediction module implements a prediction feature that returns the estimated value of offloading time, the size of the data points the estimation is relied on, and a 1-10 score gauging the confidence within the estimation. Because this is a time-consuming procedure, it is scheduled to run every week, and the final models are saved in binary format along with the normalizing process, RMSE, and r2 values.

In the means of evaluating a model, a confidence score is calculated based on the r2 and RMSE values. These two scores are then averaged which produces the final confidence score. If a forecast differs from the correct figure by no more than 3 minutes or 20%, it is considered reliable. After calculating the unloading period, the correction factor is applied, and it is calculated using previously available historical data for the relevant customer. Their findings, according to the paper, outperform the competition [4].

2.3 Representing Categorical Variables

A categorical variable is one whose values are determined by the labels attached to it. The variable "color," for example, might have the values "pink," "white," and "purple." In categorical data, there could be an ordered association in-between category, such as "top," "middle," and "last." Ordinal data is a sort of categorical data that includes useful

ordering information. Machine learning algorithms and deep learning neural networks require numerals as variables. This means that before fitting and assessing a model, categorical data must be transformed into numbers.

A comparison of seven categorical variable encoding approaches to be utilized for classification using Artificial Neural Networks on a categorical dataset is presented in the publication [10]. Paper provides a detailed description of 7 different encoding techniques.

Table 3: Encoding Techniques

Encoding Technique	Description
One Hot Coding	Used when converting a single variable with p values and k unique values into k binary variables with p values each. It compares each level of the categorical variable to a fixed reference level.
Ordinal Coding	Each category is given an integer in this case, assuming the category dimension is available. It doesn't introduce new columns to the results, but it does imply a variable order that may or may not exist.
Sum Coding	Compares the dependent variable's mean for a given category to the dependent variable's overall mean across all categories.
Helmert Coding	Compares the mean of the succeeding levels of a categorical variable to each level of the categorical variable.
Polynomial Coding	In the categorical variable examines polynomial trends. This sort of coding scheme should always be used with only an ordinal scale with equally spaced levels.
Backward Difference Coding	The dependent variable's mean for one level of such categorical variable is assessed to the dependent variable's mean for the level preceding.
Binary Coding	The categories are first encoded as ordinal numbers, then converted to binary format, and finally, the digits from such binary format are separated into various columns.

The paper [11] examines categorical data encoding strategies in machine learning, including binary vs one-hot encoding and feature hashing. Systematic comparison and contrast of encoding methodologies are performed in this paper. One-hot, according to the paper, necessitates the storage of a dictionary that maps categorical features to vector indices. These dictionaries will place a considerable strain on a computer's memory resources when the cardinality of the categorical features is high. Furthermore, even for simple models, storing the parameter vectors for one-hot encoded data becomes difficult in sparse and high-dimensional function domains.

Feature hashing has become a common method for addressing the scalability issues that come with one-hot encoding. This is accomplished by eliminating the need for a dictionary and allowing dimensionality reduction. The method has been used to solve large-scale machine learning problems with great success. The key issue with function hashing, according to the paper, is the possibility of hashing collisions, which occur when two different values hash to the same index. It has been shown empirically that the existence of hashing collisions does not prevent good model results.

The paper proposes using a binary encoding scheme as an alternative to one-hot encoding to solve the problems associated with it. In other words, a function with eight distinct values will be interpreted as a three-dimensional vector (log2(8)). This, like one-hot, necessitates a mapping from categorical values to integers, but the integer is represented in binary. A categorical value mapped to an integer value of five will be represented in a three-dimensional vector like [1, 1, 0] (five in binary format). Using one-hot encoding one would have to use a five-dimensional vector: [0, 0, 0, 0, 1].

Another method of encoding categorical data is to use studied embedding. [12] provides a basic introduction to this. A trained embedding, also known as an "embedding," is categorical data distribution. As the neural network is trained, each segment is assigned to a unique vector, and the vector's attributes are altered or learned. Closely related categories naturally cluster together in the vector space, which works as a projection of the categories. This offers the advantages of a one-hot encoding in that each segment has a vector representation, as well as an ordinal relationship in that any such relationship may be learned from data. The input vectors are not sparse, unlike one hot encoding (not filled with 0s). The disadvantage is that it involves learning as part of the model and the introduction of a large number of additional input variables (columns).

2.4 Regression models and evaluation techniques

The statistical method of regression analysis is used to investigate relationships between variables [13]. There are machine learning approaches for predicting a continuous result (y) based on the values of one or more response variable (x). A regression model's aim is to create an equation that describes y via x. Then, using new values for x, this equation can be used to calculate the result (y). A linear model is a scenario where input (x) and (y) are assumed to have a linear relationship (y). Opposing to linear relationships between input features and the target variable can be found using decision trees. Random Forest Regression [14] is an ensemble learning-based supervised learning approach for regression. Ensemble learning takes predictions from multiple machine learning algorithms and combines them to give a more accurate forecast than a single model.

The model with the lowest prediction error can be identified as the best model. There are many metrics for evaluating and examining regression models [15], including the following.

- Root Mean Squared Error (RMSE): The model prediction error is measured. It represents the average error between the actual known outcome values and the model's anticipated value.
- Adjusted R-square (R^2): Represents the proportion of variation in data explained by the model. This corresponds to the overall quality of the model. The higher the adjusted R2, the better the model. 0 means model is as good as giving out the mean for all inputs. Negative means it is worse than giving out the mean,

[15] argues that a predictive power index can be constructed based on the degree of categorical granularity a regression model can accomplish. This index of resolution power increases non-uniformly with the well-known r2 value statistic, even under differing distributional assumptions. This connection also shows that the predictive capability of models with r2 0.65 is modest and virtually constant, but increases significantly with higher r 2 values, indicating that in models that already explain a significant percentage of the variation, explicative variables are needed.

2.5 Avoid overfitting in regression models

[16] has provided a deep analysis of the effects of overfitting in regression models. The paper's core idea is that overfitting leads to unnecessarily positive model results, and results that show in an overfitted model might not really present in the distribution and thus won't be seen again. The author suggests three strategies to be considered so this pitfall can be avoided. The first one is to collect more data so that each item will have a minimum of 10-20 elements so it will reflect better in modeling. The second is to reduce the number of predictors in the model so DoF (degrees of freedom) is preserved. The author claims even if we follow the first and second it is possible to

have overly optimistic results. He recommends using shrinkage techniques to determine the amount of the overfitting and generate an estimate of how well the model would fit in unseen data. A sort of shrinkage estimator is the adjusted R2 that appears on the output of several statistical software.

3 METHODOLOGY

The methodology followed in the research is depicted in the diagram below. The first step is forming the dataset in a way that calculations to follow can be performed in an effective manner. After features are selected from the formed, pre-processed dataset customerIDs and driverIDs were studied along with the unloading time to form the categories. The methodology contains training two sets of ML models utilizing the following set of features,

- Label encoded customer and Driver Ids along with other features
- Segment means of customer and Driver Ids along with other features
- Segment identifier for customer and Driver Ids along with other features

Finally, an evaluation of the two sets of models was performed to validate the research problem at hand. The remainder of the document provides a detailed summary of each step proposed.

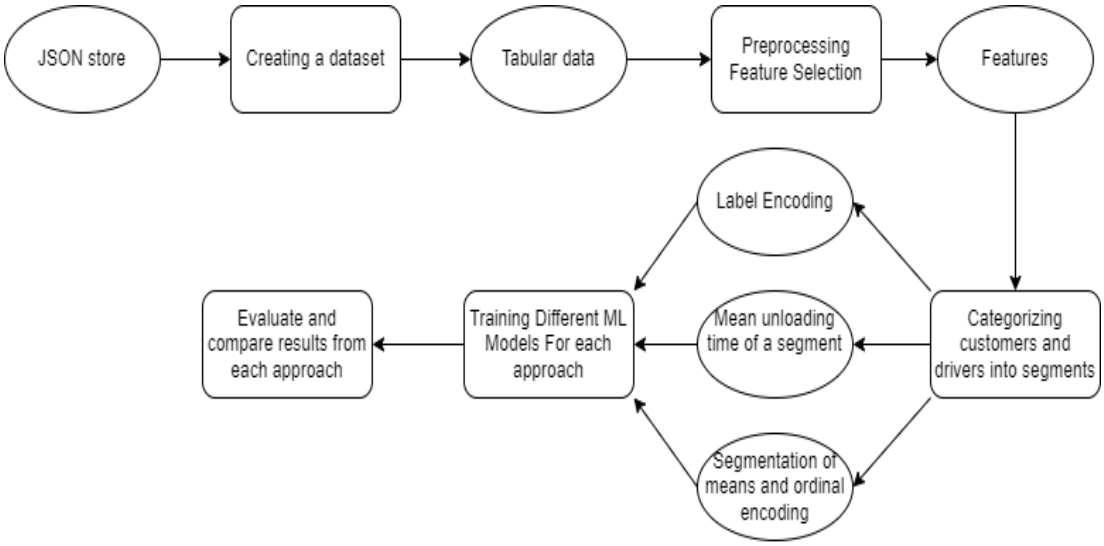

Figure 3: Methodology Steps

3.1 Creating a dataset

A JSON store contains the raw data that will be used in this study. A delivery that occurred in the year 2020 can be identified by a single document. This data was captured in real time as the delivery took place on that particular day. Data is preserved on a regular basis since data flow is high and it takes up a lot of space in the database. The data pipeline had to be built for this exercise so that the tabular data format could be created from the archive and used in future experiments.

The AWS Glue ETL service is used in this exercise. It allowed me to take the S3 archive as the input and use a spark job to build the tabular data format. The end result was a 9 million row CSV file with various deliveries in it. The conversion workflow is depicted in the diagram below.

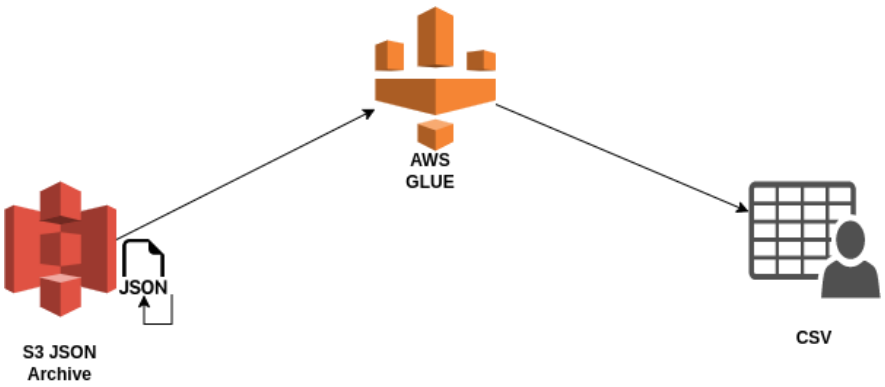

Figure 4: CSV generation steps

3.2 Dataset Description

Before delving into the contents of the data, it's crucial to understand what a single data unit represents. In this study, a data unit reflects information on a delivery that occurred. A truck made the delivery, which was making many deliveries along a route. A GPS tracking gadget is installed in each truck. In addition, the driver carries a scanner on his person (STS scanner). The time it takes for a delivery to arrive at a stop is determined by either GPS or STS scan times.

Once a truck's tracking unit enters the virtual geofence around a stop, the GPS arrival is populated. It will record the departure time from the stop once it is penetrated again. The information recorded by the STS device is based on the scan times of the goods assigned to a given delivery. Each attribute of a data unit is summarized in the table below.

Table 4: Dataset Description

Feature	Description	Example
STS_EndUnloadTime	STS device end unloading time stamp in milliseconds.	1610531553
STS_FirstScanTime	STS device first timestamp in milliseconds.	1610531553
STS_LastScanTime	STS device last timestamp in milliseconds.	1610531553
STS_StartUnloadTime	STS device unloading started timestamp in milliseconds.	1610531553
Telogis_ActualArrivalTime	Entering time of the virtual geo-fence.	1610531553
Telogis_ActualDepartureTime	Leaving time of the virtual geo-fence.	1610531553
Telogis_ExpectedArrivalTime	Expectation to reach the virtual geo-fence	1610531553
Telogis_ExpectedDepartureTime	Expectation to leave the virtual geo-fence	1610531553
__actualArrivalTime	Algorithm decided arrival from either GPS or scanning	1610531553
__actualDepartureTime	Algorithm decided departure from either GPS or scanning	1610531553
stopTime	Pre calculated stop time from a third party service	12
customerId	UUID to identify the stop.	100-12345
dayOfWeek	Weekday of the delivery	0
deliveryDate	Delivery date	2021-01-13

deliveryDateISO	ISO formatted delivery date	2021-01-13T00:00:00.000+00:00
deliveryStatus	Status of the delivery	delivered
driverId	UUID of the driver	12319460422
isPreScan	Flag to identify whether the scanning goods happened within the geo-fence	true
isSTSDetected	Flag to identify whether scanner is detected for the route.	true
zone	Zone that stop belongs	100
stopLocation	Location information of the stop.	latitude:39.718601286924 longitude: -99.8913209075523
stopNumber	Sequence number of the stop within route	7
unitId	UUID to identify the GPS unit	1051766598

3.3 Data Pre-processing and Preparation and Feature Selection

Unloading time was computed using the arrival and departure timestamps as a prerequisite for this step. Two types of unloading times were discovered for study reasons. The first is unloading time based on GPS. The timestamps from the virtual geo-fence crossing are shown below. Another is the arrival times calculated by the algorithm after taking into account both the handheld scanner and GPS times. Both types of unloading are addressed during the analysis. However, GPS unloading time is only considered after the analysis in the research experiment.

3.3.1 Analysing the dataset and preprocessing

Initially the new two variables were introduced in the following way for future research purposes,

- GPS unloading time (telogis_actual) = Telogis_ActualDepartureTime - Telogis_ActualArrivalTime
- Algorithmic unloading time (actual) =__actualDepartureTime - __actualArrivalTime

The number of delivery cases is the most relevant variable in terms of unload time forecast, according to the domain knowledge of the research problem. Following analysis was performed for both GPS-based and algorithm-based unloading times to determine the outlier distribution.

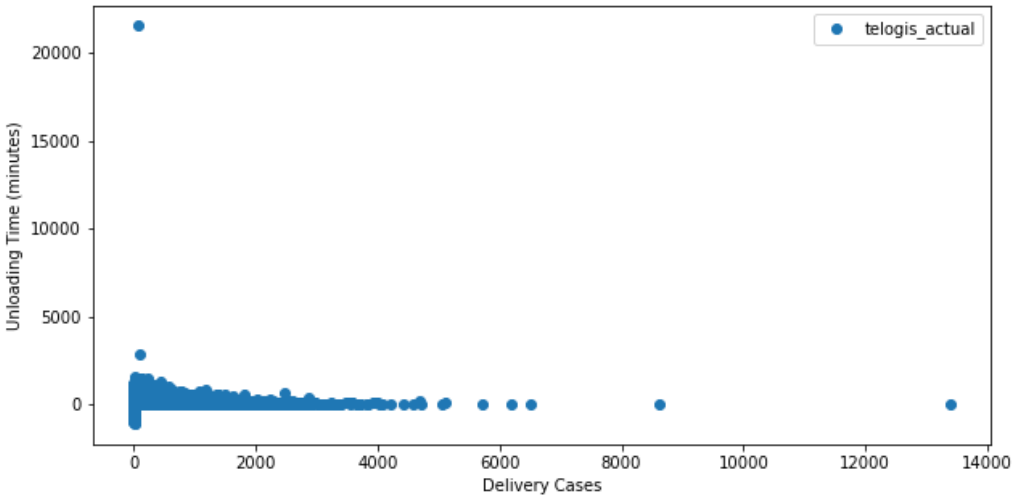

Figure 5: GPS unloading time VS delivery cases

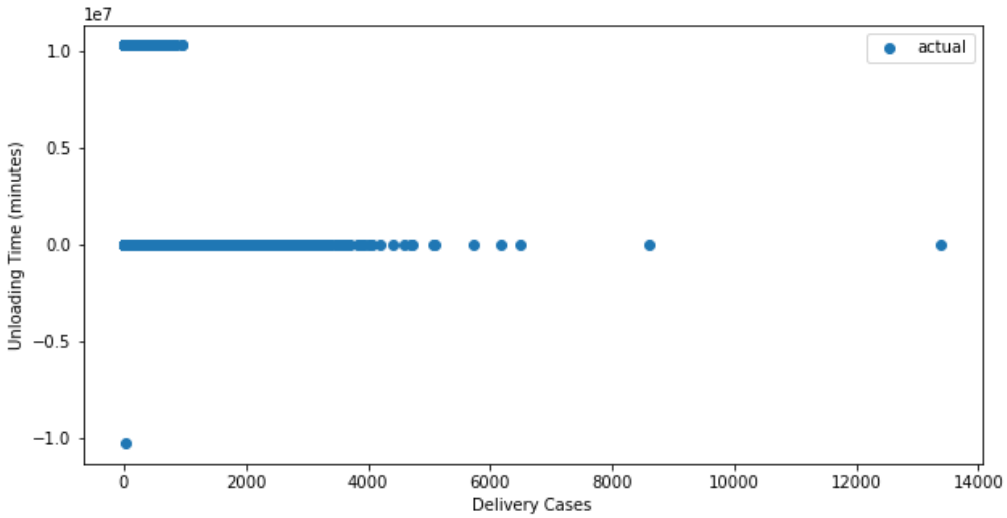

Figure 6: Algorithmic unloading time vs delivery cases

From the start, the two diagrams have acted in opposite ways. In rare cases, where GPS-based unloading times behave in a less extreme manner, algorithmic time calculation has reported extreme results. After determining the problem's context, the same analysis was carried out on the dataset after removing outliers. Those context-based restrictions were,

- Because the truck that is delivering the cases does not have enough capacity, the number of delivery cases for delivery cannot exceed 5000.
- Unless the truck breaks down, the unloading time for a stop cannot exceed 4 hours.
- Unloading time negative values are purely caused by data mishaps.

The following diagrams show the data point distribution after these filters are added.

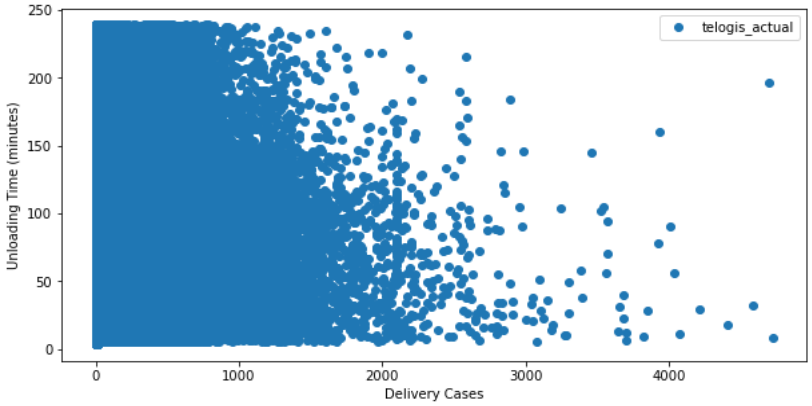

Figure 7: GPS unloading time VS delivery cases - Without Outliers

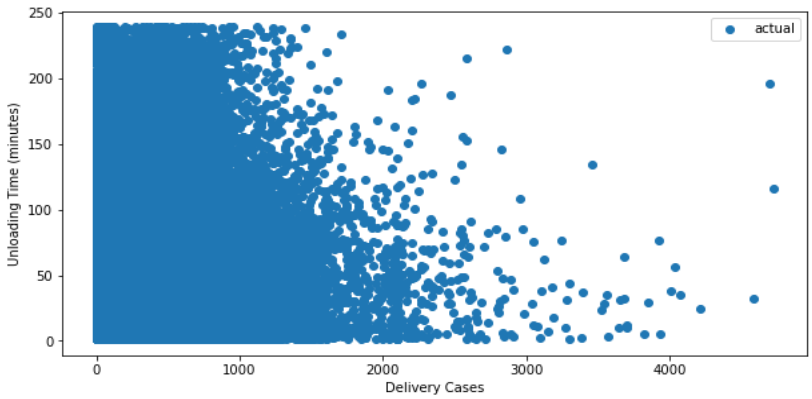

Figure 8: Algorithmic unloading time vs delivery cases – Without Outliers

Now both the distributions have a similar data point distribution. However, with the previous distributions, it was shown that algorithmic arrival time has more outliers and tends to be error-prone. So, for the experiments described in the latter part of the thesis GPS based unloading time is considered as the target variable.

3.3.2 Feature Selection

The illustrations in the preceding sections, on the other hand, show how unloading times can differ extensively even for the same number of delivery cases. As a result, it was critical to assess the remaining features from a correlation standpoint.

Figure 9: Pearson correlation Plot

The initial assumption on the delivery cases variable is supported by the Pearson correlation[18] heat map. Apart from that, the stop time variable is the only other variable that has a substantial impact on the GPS unloading time (telogis actual). However, we can't use this property in our experiment because its value isn't available when we make the prediction in the future. This is only filled up later using a third-party API. Because there is no association between the features and the target variable, feature selection becomes more difficult in this scenario.

According to the literature on the topic, unloading time largely depends on the unloading site (client) and the person unloading the goods (driver). As a result, we will use those two attributes along with the delivery cases variable to predict unloading time in this research problem.

3.4 Categorizing customers and drivers into segments

Now that we've decided on customer and driver as features, we need to model them in such a way that they can be represented mathematically and used to train a prediction

model. Several options are suggested in the literature. However, because of the nature of the problem, the dimension of the features in this situation is extremely high.

The research problem of interest is a regression problem. One-hot encoding is a great tool for turning some of these categorical features into multiple binary features; the presence or absence of the individual categorical unit can then be fit into the linear regression. However, in this context, there are 10000 different drivers and 300000+ different stops. If the one-hot encoding is to be used then we will end up with an unusably large feature set due to this limitation. Literature suggests a mode to keep feature size low via keeping the top few values and using the rest as another value. If we use this in our problem domain, we will be setting the majority of stops or drivers into a single segment which is a fundamentally flawed way of using the feature effectively.

Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on ordering. Literature suggests that ordinal encoding should be used only when there is an ordinal variable in use. However, for the research experiment label encoding with ordinal coding is also considered with the initial values.

3.4.1 Label Encoding with ordinal coding

For this we used sklearn label encoder. When it is provided with a data row with categorical variables it encodes target labels with values between 0 and n_classes-1. Following are the results from the label encoding of the two variables.

Table 5: Label Summary

Variable	# of Labels
Driver	347225
Customer	9910

featureSet1

- DriverId label encoded value
- CustomerId label encoded value
- deliveryCases

3.4.2 Using mean unloading time of a segment

This section elaborates on using the mean unloading time of the drive and customer instead of using a direct identifier of the variable. The following table contains the

description of the number of categories and their breakdown for the two variables. Please note that a segment represents a unique driver or a customer in this sense.

Table 6: Driver and Customer Overall Segment Details

	Driver	Customer
Number of categories	9910	347225
Minimum points in a segment	27	10
Maximum points in a segment	3492	1737
Mean Group size	855	24

Next mean is calculated for each of the categories in the following way. The expectation of this exercise is to find out the mean time it takes to unload a delivery case by either of the entities.

Mean unloading time for a delivery case;
$\sum_{x=1}^{n} \quad Unloading\ time \div \sum_{x=1}^{n} \quad Delivery\ case$
Where there are n entries in each segment

Core idea behind this is the mean value will differ depending on the experience of the driver for the driver and unloading difficulty for the customer.

featureSet2

- DriverId segment mean
- CustomerId segment mean
- deliveryCases

3.4.3 Segmentation of means and use of ordinal label encoding

Next step is to analyze the distribution of the mean unloading time of a case. The expectation of this exercise is to map the unloading time mean to a measurable categorical variable of a lesser dimension. Below two tables show the details of the mean distributions.

Table 7: Distribution Information of Mean Unloading Time of a Delivery Case For Customer

mean	0.925019 minutes for a case
std	2.116362
min	0.003000 minutes for a case
25% (Q1)	0.477126 minutes for a case
50% (Q2)	0.637816 minutes for a case
75% (Q3)	0.887500 minutes for a case
max	214 minutes for a case

Table 8: Distribution Information of Mean Unloading Time of a Delivery Case For Driver

mean	0.540314 minutes for a case
std	0.534615
min	0.009809 minutes for a case
25% (Q1)	0.436351 minutes for a case
50% (Q2)	0.507977 minutes for a case
75% (Q3)	0.595982 minutes for a case
max	38.065217 minutes for a case

Next step is to provide a meaningful segmentation to the mean value distribution. For this purpose, delivery case unloading time distribution was segmented according to the distribution quantile values. According to the order of the quantile, each quantile was assigned a numerical value from the set [1,2,3,4]. Following is a description of the mapping technique,

$$
\begin{gathered}
if\ \mu < Q1, featureValue = 4 \\
else\ if\ Q1 <= \mu < Q2, featureValue = 3 \\
else\ if\ Q2 <= \mu < Q3, featureValue = 2 \\
else\ featureValue = 1
\end{gathered}
$$

For an example if the driver segment 00001 has a mean unloading time of 0.46 mins for a case, Considering the values if the table above, this segment will now belong to the segment 3 as its mean lies among the distributions Q1 and Q2.

featureSet3;

- DriverId segment identifier
- CustomerId segment identifier
- deliveryCases

3.5 Training ML models

As Section 3.4 showed there were three different feature sets built upon the ~8 million row containing data set. For the experiment, these three feature sets from the data set were fed into the following types of models. For each model type, state of the art techniques was used to eliminate overfitting

- LinearRegressor Model with L1 and L2 regularization (ElasticNet)
- RandomForrestRegressor Model with fixed max depth
- GradientBoostingRegressor Model with fixed max depth

4 EVALUATION

Model evaluation is one of the key steps in the process of building a prediction model. Model evaluation helps to quantify the capability of the evaluator to perform better for unobserved examples. Further, it measures how precisely the model can accomplish those. In this study, all the constructed models were tested carefully to confirm that the model is suitably fitted to the training dataset adhering to the methods from the literature to avoid fitting problems. Models can be evaluated by comparing the ground truth data and associating it with the model predictions.

The simplest approach is to split the dataset into two sets called a training and testing dataset with 80% to 20% portions of the original dataset. The first part is to be used as the training set while the other set serves the purpose of testing the model. However, the train/test split evaluating method cannot detect the problem of overfitting. Due to that, the trained model can lead to being less precise on unseen data. The cross-validation evaluation technique can be used to detect whether a trained model has overfitted or not.

In this study 10-fold, a cross-validation technique was used to evaluate the models. Here the data gets partitioned into ten subgroups every time, one of the subsets gets utilized for testing while the other nine subsets get to construct the training set.

4.1 RMSE

Root Mean Squared Error (RMSE): The model prediction error is measured. It represents the average error between the actual known outcome values and the model's anticipated value.

4.2 R square

Adjusted R-square (R^2): Represents the proportion of variation in data explained by the model. This corresponds to the overall quality of the model. The higher the adjusted R2, the better the model. 0 means the model is as good as giving out the mean. Negative means it is worse than giving out mean.

4.3 Accuracy

$$accuracy\_for\_margin\_x = \#\ accurate\ values\ in\ margin\ x\ \div\ total\ predictions$$

$$where; accurate\ value\ in\ margin\ x = |prediction - prediction * x\%| \leq actual\ value$$

$$where;\ x\epsilon[10,20,30,40,50,60,70,80,90]$$

This measurement provides an understanding of the model's performance in a simpler way so that especially business stakeholders can understand better on the comparison.

5 RESULTS

This section presents the results obtained for the proposing approach including the performance for different feature sets. In order to compare the performance, the traditional machine learning models that have been used in literature have been used on the same dataset.

5.1 featureSet1

Table 9: Results for featureSet1

Model	**MSE**	**R^2**	**Accuracy with 40% confidence**
Linear Regression	12.86	0.27	50%
Random Forest	11.59	0.37	57.8%
Gradient boost	11.20	0.41	58%

5.2 featureSet2

Table 10: Results for featureSet2

Model	**MSE**	**R^2**	**Accuracy with 40% confidence**
Linear Regression	12.53	0.29	53%
Random Forest	11.29	0.42	58.3%
Gradient boost	9.38	0.57	65.4%

5.3 featureSet3

Table 11: Results for featureSet3

Model	**MSE**	**R^2**	**Accuracy with 40% confidence**
Linear Regression	12.46	0.30	53.7%
Random Forest	10.13	0.50	63.5%
Gradient boost	9.06	0.59	68.7%

6 DISCUSSION

The results of three separate regression models, each for three different feature sets that were found, are shown in the section above. The cross-validation method was used to determine whether the models were overfitted when evaluating their performance. To avoid overfitting as a safety net, state-of-the-art procedures were applied for each model type.

The Gradient boost regressor outperformed both the random forest regressor and the linear regressor when comparing the three model types. The data clearly show that featureSet3 produced the best outcomes of the three sets. The mean unloading time per case was segmented according to its distribution to create FeatureSet3. Interestingly, it outperformed featureSet2, in which models were fed the segment mean.

Because label encoding has changed driver and customer into an ordinal variable based on its id, featureSet1's label encoded features have the lowest values in the results. As a result, throughout the learning process, the model tries to find a relationship between the encoded value and the target variable, which does not correspond to reality. It turns out that providing the mean of a segment is a superior method to handle this feature. However, in this study question, it appears that creating a new ordinal variable based on the mean distribution is the most effective technique of handling features. In comparison to the other two ways, that method has offered a more realistic description of feature behaviour.

7 CONCLUSION

The goal of this research is to find ways to improve the unloading time forecast by segmenting customers and drivers. The delivery cases included were discovered to be the most related variable to the unloading time during the initial examination of the data set. The driverId and customerId parameters were the most beneficial of the other attributes in the data set. However, due to their high dimensions, incorporating these two variables into a regression model was a difficulty in and of itself. When compared to employing an ordinal encoding technique indiscriminately, which imparted an orderly nature to features from the id itself, segmentation of the stated variables and using segment mean yielded better regression results. Furthermore, models were able to offer their least MSE once segment means were grouped based on their distribution and supplied a cluster identification that justifies the orderly structure. Finally, it is clear that improving the unloading time forecast improves consumer and driver segmentation.

8 REFERENCES

[1] G. B. Dantzig and J. H. Ramser, “The Truck Dispatching Problem,” *Manag. Sci.*, vol. 6, no. 1, pp. 80–91, Oct. 1959, doi: 10.1287/mnsc.6.1.80.

[2] E. Zunic and D. Donko, *Parameter Setting Problem in the Case of Practical Vehicle Routing Problems with Realistic Constraints*. 2019, p. 759.

[3] E. Žunić, D. Đonko, and E. Buza, “An Adaptive Data-Driven Approach to Solve Real-World Vehicle Routing Problems in Logistics,” *Complexity*, vol. 2020, p. e7386701, Jan. 2020, doi: 10.1155/2020/7386701.

[4] “Adaptive multi-phase approach for solving the realistic vehicle routing problems in logistics with innovative comparison method for evaluation based on real GPS data: Transportation Letters: Vol 0, No 0.” https://www.tandfonline.com/doi/full/10.1080/19427867.2020.1824311 (accessed Mar. 10, 2021).

[5] “(17) (PDF) Improving unloading time prediction for Vehicle Routing Problem based on GPS data.” https://www.researchgate.net/publication/344173524_Improving_unloading_time_prediction_for_Vehicle_Routing_Problem_based_on_GPS_data (accessed Mar. 10, 2021).

[6] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, “Erratum: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization,” *J. Oper. Res. Soc.*, vol. 37, no. 6, pp. 655–655, Jun. 1986, doi: 10.1057/jors.1986.117.

[7] S. N. Kumar and R. Panneerselvam, “A Survey on the Vehicle Routing Problem and Its Variants,” vol. 2012, May 2012, doi: 10.4236/iim.2012.43010.

[8] “(17) Fifty Years of Vehicle Routing.” https://www.researchgate.net/publication/220413044_Fifty_Years_of_Vehicle_Routing (accessed Mar. 10, 2021).

[9] R. Grosso de la Vega, J. Muñuzuri, A. Escudero-Santana, and E. Barbadilla-Martín, “Mathematical Formulation and Comparison of Solution Approaches for the Vehicle Routing Problem with Access Time Windows,” *Complexity*, vol. 2018, pp. 1–10, Feb. 2018, doi: 10.1155/2018/4621694.

[10] K. Potdar, T. Pardawala, and C. Pai, “A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers,” *Int. J. Comput.*

Appl., vol. 175, pp. 7–9, Oct. 2017, doi: 10.5120/ijca2017915495.

[11] C. Seger, “An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing,” p. 41.

[12] J. Brownlee, “3 Ways to Encode Categorical Variables for Deep Learning,” *Machine Learning Mastery*, Nov. 21, 2019. https://machinelearningmastery.com/how-to-prepare-categorical-data-for-deep-learning-in-python/ (accessed Mar. 10, 2021).

[13] A. O. Sykes, “An Introduction to Regression Analysis,” p. 34.

[14] A. Liaw and M. Wiener, “Classification and Regression by RandomForest,” *Forest*, vol. 23, Nov. 2001.

[15] Y. T. Prairie, “Evaluating the predictive power of regression models,” *Can. J. Fish. Aquat. Sci.*, Apr. 2011, doi: 10.1139/f95-204.

[16] Babyak, Michael A. PhD What You See May Not Be What You Get: A Brief, Nontechnical Introduction to Overfitting in Regression-Type Models, Psychosomatic Medicine: May 2004 - Volume 66 - Issue 3 - p 411-421

[17] docs.aws.amazon.com. 2022. AWS Glue and AWS Glue Studio. [online] Available at: <https://docs.aws.amazon.com/glue/latest/dg/what-is-glue.html> [Accessed 30 April 2022].

[18] Samuels, P. Gilchrist, M. Pearson Correlation; Birmingham City University: Birmingham, UK, 2014; pp. 1–4.

[19] scikit-learn.org. 2022. PreProcessing Label Encoder. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html> [Accessed 30 April 2022].