Analysis of Inefficiencies in Shipment Data Handling

by

Rohini Prasad
Master of Business Administration, Indian School of Business, 2014
Bachelor of Engineering, Information Technology, University of Pune, 2010

and

Gerta Malaj
Bachelor of Arts, Mathematics, Wellesley College, 2013

SUBMITTED TO THE PROGRAM IN SUPPLY CHAIN MANAGEMENT
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING IN SUPPLY CHAIN MANAGEMENT AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2017

Signature of Author... Signature redacted ......
Master of Engineering in Supply Chain Management
May 12, 2017

Signature of Author.................................. Signature redacted ....
Master of Engineering in Supply Chain Management
May 12, 2017

Certified by............................. Signature redacted ....
Dr. Matthias Winkenbach
Director, MIT Megacity Logistics Lab
Thesis Supervisor

Accepted by....................... Signature redacted ....
Dr. Yossi Sheffi
Director, Center for Transportation and Logistics
Elisha Gray II Professor of Engineering Systems
Professor, Civil and Environmental Engineering

Analysis of Inefficiencies in Shipment Data Handling

by

Rohini Prasad

and

Gerta Malaj

Submitted to the Program in Supply Chain Management
on May 12, 2017 in Partial Fulfillment of the
Requirements for the Degree of Master of Engineering in Supply Chain Management

ABSTRACT

Supply chain visibility is critical for businesses to manage their operational risks. Availability of high quality and timely data regarding shipments is a precursor for supply chain visibility. This thesis analyses the errors that occur in shipment data for a freight forwarder. In this study, two types of errors are analyzed: system errors, arising from violations of business rules defined in the software system, and operational errors, which violate business rules or requirements defined outside the software. We consolidated multifarious shipment data from multiple sources and identified the relationship between errors and the shipment attributes such as source or destination country. Data errors can be costly, both from a human rework perspective as well as from the perspective of increased risk due to supply chain visibility loss. Therefore, the results of this thesis will enable companies to focus their efforts and resources on the most promising error avoidance initiatives for shipment data entry and tracking. We use several descriptive analytical techniques, ranging from basic data exploration guided by plots and charts to multidimensional visualizations, to identify the relationship between error occurrences and shipment attributes. Further, we look at classification models to categorize data entries that have a high error probability, given certain attributes of a shipment. We employ clustering techniques (K-means clustering) to group shipments that have similar properties, thereby allowing us to extrapolate behaviors of erroneous data records to future records. Finally, we develop predictive models using Naïve-Bayes classifiers and Neural Networks to predict the likelihood of errors in a record. The results of the error analysis in the shipment data are discussed for a freight forwarder. A similar approach can be employed for supply chains of any organization that engages in physical movement of goods, in order to manage the quality of the shipment data inputs, thereby managing their supply chain risks more effectively.

Thesis Supervisor: Dr. Matthias Winkenbach
Title: Director, MIT Megacity Logistics Lab

ACKNOWLEDGEMENTS

We would like to thank everyone who has supported us in completing this thesis. Our special acknowledgements go to our thesis advisor, Dr. Matthias Winkenbach, for providing us with his inestimable guidance, insightful discussions, and deep expertise. The magnitude and significance of Matthias' leadership and constant encouragement to find new ways of exploring and presenting our thesis content cannot be overstated.

We would also like to express our gratitude to our sponsor company, Damco, for giving us their time, data and invaluable insights on this thesis. Finally, we would like to thank the entire MIT SCM cohort for their generous contributions to our education; and our families and friends for their tremendous support.

Table of Contents

List of Figures

List of Tables

Glossary of Terms and Acronyms

- ABS - Agent Based Simulation
- AHP - Analytic Hierarchic Process
- ALMIL - Adaptive Language Modeling Intermediate Layer
- ARIMA - Autoregressive Integrated Moving Average
- AUC – Area under the curve, of the ROC plot
- CART - Classification and Regression Trees
- DES - Discrete-Event Simulation
- EDI – Electronic Data Interchange
- FETA - Forecasted Expected Time of Arrival
- GET – Global Exception Tool, where all the system exceptions funnel through
- MAD - Mean Absolute Deviation
- MAE - Mean Average Error
- MAPE - Mean Average Percentage Error
- MNA – Messaging and Alerting system
- NLP - Natural Language Processing
- OLAP - Online Analytical Processing
- PCA - Principal Component Analysis
- RMSE - Root Mean Square Error
- ROC – Receiver- Output Characteristics, a graph of the model's Sensitivity against (1 – Specificity)
- RSS - Really Simple Syndication
- SCOR - Supply Chain Operations Reference

- SD - System Dynamics
- SMT - Statistical Machine Translation
- UI – User Interface, the system used by Damco to input all shipment-milestone data
- Waybill – a document issued by a carrier outlining details and instructions about a shipment of goods. Sometimes you can have many events for each waybill, and many waybills per container. However, for our dataset, it is one-to-one. Therefore, waybill and shipment are used analogously.
- XML - eXtensible Markup Language

1. Introduction

1.1 Background and Research Question

Our thesis sponsor, Damco, is a freight forwarding and supply chain management service provider. One key task that they perform is the tracking of each shipment for the customer. They classify the shipment transit lifecycle across multiple milestones, such as 'Dispatch', 'Arrival at port', and 'Awaiting customs clearance'. This thesis focuses on the application of analytics to determine the probability and likely cause of data entry errors while recording the key milestones associated with a shipment. Real-time tracking is essential for supply chain visibility. The majority of Damco's customers track an industry standard list of 8 shipment milestones per shipment. The shipment-milestone data for these customers is updated systemically using Electronic Data Interchange (EDI). However, there are a number of non-standard milestones required by a few customers, which require manual tracking and update, especially for air shipments. These entries have an increased probability of error due to the manual intervention involved, which can result in missing updates or data entry errors. Moreover, as supply chains become more global and complex, visibility becomes both more important and more challenging. The repercussions of data errors can be detrimental for Damco's clients. Any organization involved in the movement of goods faces similar challenges. Our approaches and findings can, therefore, also be effective for these organizations. The customer data analyzed for this review is from one of Damco's non-standard customers who requires tracking of multiple shipment-events in addition to the industry standard set. Although this issue currently reflects a problem with one of Damco's customers, descriptive and predictive analytics can be applied toward all customers in the near future as Damco's machine-intelligence matures, becoming more accurate and potentially prescriptive. This thesis explores some the descriptive and predictive techniques that Damco could utilize in this process.

Although we have analyzed several research papers on the analytical techniques used in supply chain and data error detection, this is a deep and diverse field, and research topics tend to specialize on specific real world problems, such as in weather forecasting, traffic incident forecasting, and natural language error detection and correction. Analysis of the current techniques used in grammatical and lexical corrections gives us useful insights on how to anticipate and remedy human data entry error. By using a hybrid model that utilizes both descriptive and prescriptive analysis, our thesis develops a reusable framework for data entry error detection and correction. This framework will help improve supply chain visibility, particularly for the logistics function, where the cost of missing data and data error is high.

1.2 Thesis Scope and Structure

The rest of this thesis is organized as follows. Section 2 presents a summary of the literature review on the use of analytics in supply chains and the main methods and challenges related to them. In section 3, we present the methodological framework that we developed to identify the root cause of the shipment error and to deduce the probability of errors in future shipment data entries based on the historical trends. We use descriptive analytics techniques for the former and predictive analytics for the latter. Additionally, we apply this framework to a practical example as a case study, which is laid out in section 4. Finally, we discuss the limitations associated with our case study, arising from data unavailability, as well as limitations of the proposed methodological approach.

2. Literature review

This literature review focuses on two key areas: First, the use of analytics in supply chains and the key trends and challenges related to them; Second, the existing techniques used and case studies demonstrating data error detection and correction.

2.1 Analytics in Supply Chains

Data analytics can be broken down into three main categories: descriptive, predictive, and prescriptive analytics. Each of these categories of analysis provides unique insights into the nature and performance of current supply chain processes, as well as the potential properties of future supply chain processes. In this section, we explore some of the work done in each of these three categories of analytics with a focus on supply chain data.

2.1.1 Descriptive Analytics

Statistical analysis is a key part of descriptive analysis. It includes both quantitative and qualitative analysis. Qualitative analysis has limited use and is usually adopted only when there is limited quantitative data available or while analyzing subjective data that requires the judgment of an expert (Wang, Gunasekaran, Ngai, & Papadopoulos, 2016). Statistical analysis may also intersect with the domain of descriptive analytics (such as time series analysis) or predictive analytics (such as regression analysis).

Oliveira, McCormack, & Trkman (2012) make a compelling case for the need for analytics in supply chains, stating that it helps in visualizing supply chain performance not just for the individual players in the supply chain - suppliers, distributors, manufacturers, retailers - but also for the supply chain as a whole. They argue that the primary role of business analytics, particularly descriptive and predictive, is to increase the propensity of information processing and exchange in an organization (Oliveira et al., 2012). To assess the impact of business analytics on different stages of the supply chain, this study uses the supply chain operations reference (SCOR) framework. SCOR is a management tool for addressing and communicating supply chain management decisions within a company and with its suppliers and customers. The study looks at supply chains as a set of four sequential phases: 'Plan', 'Source', 'Make', and 'Deliver'. The 'Deliver' phase entails the key logistics processes that are a subject of our thesis. The study shows that, except for the case of companies with the most mature supply chains (classified as Level 1), supply chain performance in the 'Deliver' phase is enhanced significantly by the deployment of business analytics (Oliveira et al., 2012).

An example of the decision making process during the SCOR's 'Plan' phase is demand forecasting using descriptive and predictive analytics. Demand forecasting is essential for supply chain planning. Different scenarios may involve different analytics techniques. Causal forecasting methods, used to analyze factors that affect demand for a product, include linear, non-linear, and logistic regression. As an example, consider the relationship between demand forecasting and the production planning process. Parts have dependent demand, as their demand depends on the SKUs that use those parts. Conversely, SKUs themselves have independent demand. While the demand for items that have dependent demand can be derived from the corresponding SKU's demand, forecasting the demand for items that have independent demand involves predictive analytics. This forecasting is generally achieved using time-series methods, for which the only predictor of demand is time.

One of the predictive analytics techniques, autoregressive modeling, involves deriving demand forecasts for any given period as a weighted sum of demands in the previous periods. Autoregressive modeling looks at a value from a time series and regresses it on previous values from that same time series, which are obtained from descriptive analytics (Souza, 2014). Thus, this technique demonstrates how predictive analytics relies on descriptive analytics.

Big data has taken center-stage in conversations on analytics. Per a definition proposed by De Mauro, Greco, & Grimaldi (2015), "Big Data represents the Information assets characterized by such a high volume, velocity and variety to require specific technology and analytical methods for its transformation into value". The availability of big data poses a new set of challenges for supply chains especially when it comes to identifying what is important and to finding the skills needed to derive insights from this data within one's supply chain (Wang et al., 2016). Descriptive analytics gives organizations a better understanding of what happened in the past, when it happened, and what is happening at present. The descriptive techniques referred to by Wang et al. (2016) are either performed at standard periods or as needed using online analytical processing (OLAP) techniques. Other techniques used in studies and practical applications include descriptive statistics such as sums and averages, as well as variants of time series and regression analysis.

2.1.2 Predictive Analytics

Predictive analytics techniques include regression, time series analysis, classification trees, and machine learning techniques such as neural networks. The most common technique used for predictive analysis and forecasting is the use of regression models. The simple linear regression model, the most basic variant of regression models, assumes that the relationship between a dependent variable (y) and an independent variable (x) is approximately linear. The method uses a least squares point estimate to find the slope and intercept of this line (Bowerman, O'Connell, & Koehler, 2005). Simple linear regressions can be extended to include multiple independent variables, giving us the multiple linear regression model. This can be expressed in the form:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

Where $X_1, X_2, ..X_p$ are p independent variables, y is a dependent variable and ε is the error term.

Alternatively, a quadratic regression model which uses the equation of a parabola of the form

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$

to establish the relationship between the dependent and independent variables is also just a variant of the simple regression model since it is a linear combination of parameters (Bowerman et al., 2005).

On the other end of the spectrum, predictive analytics includes advanced mathematical models often supplemented with programming techniques to use historical data to predict a likely future. A key step in predictive analytics is therefore to identify the appropriate explanatory variables from among all the input variables. For instance, in logistics operations, given the proliferation of big data, logistics planning problems – formulated as network flow problems – can be optimized effectively from supply to demand. Due to supply disruptions and demand uncertainty prevalent in supply chains, which directly impacts the logistics operations, predictive models play a critical role in modeling supply chain flexibility into logistics operations. The predictive analytics techniques covered in Wang et al. (2016) refers to the use of mathematical algorithms and programming models to help project what will happen in the future and why.

A key challenge faced is that the analysis of multi-dimensional data is often computationally expensive (Wang et al., 2016). Dimension reduction is a common precursor before execution of supervised learning methods (Shmueli, Bruce, Stephens, & Patel, 2016). Principal component analysis (PCA) is the most common technique used to address the 'Curse of dimensionality' problem. PCA is a statistical technique for dimensionality reduction. It seeks to capture the maximum amount of information about the data in a low dimensional representation, by performing a linear projection of the original high-dimensional feature vectors (Chang, Nie, Yang, Zhang, & Huang, 2016). These vectors, called the principal components (PCs), are a linear combination of the input variables weighted by so-called factor loadings. Within a multi-dimensional dataset, the first principal component represents the direction in which the variability is the largest. Each subsequent axis has the highest variance, after constraining the preceding principal axis.

However, the results of a PCA, the so-called principal components, are often hard to interpret. To overcome this challenge, a sparse principle component analysis (SPCA) approach can be utilized which is computationally less expensive and more generalizable. One possible approach which is commonly used to reduce dimensionality under SPCA is to manually set factor loadings (which are below a certain threshold) to zero. Another approach is to reformulate the PCA as a regression-type optimization problem and to run an optimization algorithm to calculate the factor loadings, in an attempt to end up with as few predictors as possible. To find the optimal solution efficiently for this optimization problem, the problem must be formulated as a convex function. Niculescu & Persson (2006) define convex functions as having two main properties which make them well suited for optimization problems: (i) the maximum is attained at the boundary; and (ii) any local minimum is a global minimum. The SPCA algorithm, which uses optimization to reduce the number of explicitly used variables, performs well in finding a local optimum but since it is not convex, it is difficult to ensure that it finds the global optimum. To address this problem, Chang et al. (2016) proposed a convex PCA formulated as a low rank optimization problem based on the hypothesis (and its associated proof) that SPCA is equivalent to regression. For a given dataset (represented as a vector), a sparse representation attempts to minimize the number of non-zero entries in the vector. In contrast, a low-rank representation seeks to minimize the rank of the matrix containing the data entries (Oreifej & Shah, 2014). Recall that the rank of a matrix is defined as the maximum number of linearly independent column (or row) vectors in the matrix. For a 'r x c' matrix with 'r' rows and 'c' columns, the maximum rank will be the minimum of r and c. Stat Trek (2017) offers more details on how to compute rank of a matrix.

Unsupervised and supervised learning techniques are other important prediction analysis techniques. Supervised learning techniques are used in classification and prediction. They require the availability of the outcome of interest, which is referred to as the labelled field. For example, in the case of shipment data for a freight forwarder (as shown in section 4. Case Study Analysis), a labeled field can be used to denote whether the record contains errors (Yes/ No). The data set is segmented into training and validation data. The model learns the relationship between the predictor and outcome variable from the training data. The relationship is then applied to the validation dataset to assess its performance. Classification and prediction most commonly use supervised learning (Shmueli et al., 2016).

Unsupervised learning techniques do not have any pre-specified outcome variable, so, association rules are used as part of the learning algorithm, instead. Within unsupervised learning, clustering is the most commonly used technique (Albalate & Minker, 2013). Some of the popular clustering approaches are hierarchical, partitional, model-based, density-based and graph-based algorithms. An example of partitional and non-hierarchical clustering is k-means. K-means is a method of partitioning 'n' observations into k clusters while each observation belongs to the cluster with the nearest mean. K-means is the most frequently used partitional clustering method, as it is versatile, easy to implement, and, most notably, it does not change with varying data orderings (Celebi et al., 2013).

2.1.3 Prescriptive Analytics

The logical evolution of descriptive and predictive analytics is prescriptive analytics that works by identifying and defining business rules that subsequently trigger a required action when the input condition is met. Schaffhauser (2014) lists 12 key components necessary for successful implementation and use of prescriptive analytics based on an interview with University of Wisconsin-Green Bay's CIO, Rajeev Bukralia. The two most critical questions to be asked are whether the problem lends itself well to prescriptive analysis and how it can be cross-validated for determining predictive accuracy of the model. This can be accomplished by partitioning the data into two datasets - training and validation dataset - before developing the model using the training dataset. The validation dataset can be used to test the efficacy of the model. Defining business rules is another key step for ensuring that the prescriptive output is insightful and the role of the core operations teams in this task cannot be emphasized enough. The knowledge of how the prescriptive system should behave (as captured in the business rules) lies with the people who are performing the task (which is to be enhanced with prescriptive analysis) on a daily basis. Establishing project management guidelines and data governance mechanisms – including a consistent methodology to determine what data is relevant for prescriptive analysis – is imperative. Finally, Schaffhauser (2014) reminds the reader that prescriptive analytics can fail for new unseen scenarios for which the business rules have not been defined and the model hasn't been trained.

Prescriptive analytics involves looking at historical data, applying mathematical models, and superimposing business rules in order to identify, recommend, or implement alternative decisions (Wang et al., 2016). This is valuable for the comparison of alternative decisions that involve complex objectives and large sets of constraints. Prescriptive analytics includes multi-criteria decision-making, optimization, and simulation techniques. The most common form of multi-criteria decision making technique is Analytic Hierarchic Process (AHP). AHP breaks down complex problems into multiple smaller sub-problems each with a single evaluation objective including cost-optimization and timely delivery. Through this decomposition, AHP enables pairwise comparison of alternatives or attributes with respect to a given criterion (Kou, Ergu, Peng, & Shi, 2013). Typically, one or more of these methods are needed to develop prescriptive models. Simulation involves designing the model of a system in order to predict the performance of different scenarios and thereby optimize the use of resources. Some of the most prevalent simulation models are agent-based simulation (ABS), system dynamics (SD), discrete-event simulation (DES), activity-based simulation, and Monte Carlo. Agent-based simulation allows for independent entities to interact with each other over time in order to capture complex systems. A system dynamics simulation model is a system of first-order differential or integral equations. A discrete-event simulation represents a system as a discrete sequence of occurrences in time. It is assumed that there is no variation in the system between these occurrences. Activity-Based Simulation, on the other hand, focuses on time, and occurrences are observed in the context of the timeframe in which they happen. Finally, Monte Carlo simulations empower users by providing probabilities for all possible outcomes, thereby creating thousands of 'what-if' scenarios

(Underwood, 2014). When operating with big data, running simulation techniques can be challenging as they require complex models that incorporate all relevant relationships and correlations.

Another common and differing approach to prescriptive analytics is the use of optimization to guide decision-making based on the underlying predictive model. Mathematical optimization involves identifying the best element along a defined measure from a given domain. An optimization problem often involves finding the maximum or minimum of a function (De Finetti, 2010). Optimization has long been used to improve the planning accuracy of supply chains. However, when working with big data, modeling supply chain operations and building optimization models relies on the use of large, non-smooth optimization procedures, randomized approximation algorithms, as well as parallel computing techniques. Non- smooth optimization procedures – which refers to procedures that minimize non-convex functions – are critical when operating with big data which has slow convergence rates.

It is essential to note that prescriptive analytics is not a separate category of analytical techniques in itself. Instead, it involves the application of mathematical and programmatic models to the results of predictive models.

2.2 Analytics techniques for predicting data entry errors

Techniques used to detect errors range from manual analytic approaches, such as performing searches to writing code, to sophisticated machine learning techniques that automate error detection.

2.2.1 Phrasal Statistical Machine Translation

A model for language independent error detection and correction for grammatical errors and misspelled words using phrasal statistical machine translation (SMT) is proposed by Ehsan & Faili (2013). The proposed approach is highly context-specific and complements the existing rule-based approach for grammatical and word error detection and correction algorithms used by most conventional spell checkers. The SMT uses the concept of statistical translation to model a grammar checker as a machine translator, which would therefore be language independent. The grammar checker is modeled as a noisy channel that receives erroneous sentences and suggests a correct sentence.

Ehsan & Faili (2013) split the available dataset into training and validation data and inject various grammatical errors, such as preposition omission and misspelt words, into the training data. A rule based approach for identifying and correcting errors in this training data set would require several language specific rules. SMT, on the other hand, learns the phrases in the training dataset and uses phrase probability, reordering probability, and language model to propose corrections. The experimental results of this study compare the results of translation from erroneous to correct sentences using statistical machine translation against machine translation. The results are promising and show that certain errors are only detected by using the SMT technique. These are overlooked by the traditional grammar checkers that use a rule based approach.

2.2.2 XML (eXtensible Markup Language) data models

Data models such as XML (eXtensible Markup Language) have increased rapidly in the last decade as a new standard for data representation and exchange. Existing XML data cleaning techniques involve duplicate detection in XML documents or outlier detection (whether class or attribute outliers). However, many of these techniques have shortcomings or have little efficiency. Starka et al. (2012) offer a different method which approaches data correction via an extensible system, called Analyzer. Analyzer not only addresses data correction, but also allows one to perform- data crawling (using an application, known as a 'Crawler', to systematically browse the web for the purpose of web indexing), application of analyses, and aggregation and visualization of results. These additional features would provide further insights into the data, and the subsequent correction possibilities. Starka et al. (2012), however, assume that they would have a complete data tree loaded into the system memory. Therefore, they would have direct access to all its parts. Only then would the algorithm be able to find corrections with the minimum distance to the grammar and the original data tree. Due to our limited access to relevant data, this method is not viable for our specific area of research.

2.2.3 Artificial intelligence

Among other capabilities, current intelligence exhibited by machines, commonly known as 'Artificial Intelligence', includes successfully understanding human language. One of the means to do so is via Natural Language Processing (NLP), which relies heavily on machine learning. While earlier machine learning algorithms included decision trees and constructed rigid system rules, many language recognition systems now rely upon statistical language modeling. Language modeling is a framework that computes the probability of a sequence of words, hence making soft, probabilistic decisions. Language modeling looks for possible strings in language and associates probabilities with each string. It can thus support predicting the completion of a sentence by stipulating the probability of an upcoming word. A paper by Ouazzane et al. (2012) presents a framework for intermediate layer language modeling called Adaptive Language Modeling Intermediate Layer (ALMIL). ALMIL is an artificial-intelligence-based language modeling framework that will serve as a communication layer between human and computer, analyzing data errors and providing data corrections. The layer will be applied to a QWERTY keyboard, allowing it to produce an intelligent keyboard hybrid framework. The latter will analyze users' typing patterns, and will correct mistakes, as well as predict typing objectives. In this model, the size of the training dataset plays a key role in the prediction's precision and, thus, its viability depends on the quantity and quality of the data provided.

Nonetheless, while the Artificial Intelligence (AI) approach may ultimately prove to be the ideal solution, the scope of our thesis and the data available does not allow us to implement and test such an approach. In order for us to be able to use this method, we will need large amounts of high-quality, historic data, that would provide statistically significant results and sufficient 'history' for a potential AI algorithm to learn and improve. It comes as no surprise then that the most powerful message from Jones' (2011) article is:

"Data availability is the most fundamental requirement for strong analytic capabilities." Likewise, data fragmentation is another factor that impedes the use of analytics and the search for insights.

3. Methodology

3.1 General Approach

The focus of this thesis study is to evaluate the sources of error impacting the accuracy of the shipment status for a freight forwarding company. These errors have a detrimental impact on shipment visibility and therefore are a source of supply chain risk. This section focuses on the use of descriptive analytics to identify the root cause of shipment errors. Subsequently, in section 3.3.2 Predictive Analysis, we cover the use of predictive analytics to forecast the probability of error occurrence in a shipment data entry based on the historical trends. In section 3.3.3 Prescriptive Analysis, we discuss the final stage – the development of a prescriptive algorithm to recommend appropriate corrections for data entry errors.

We used a four-phase approach in our analysis, as shown in Figure 1.

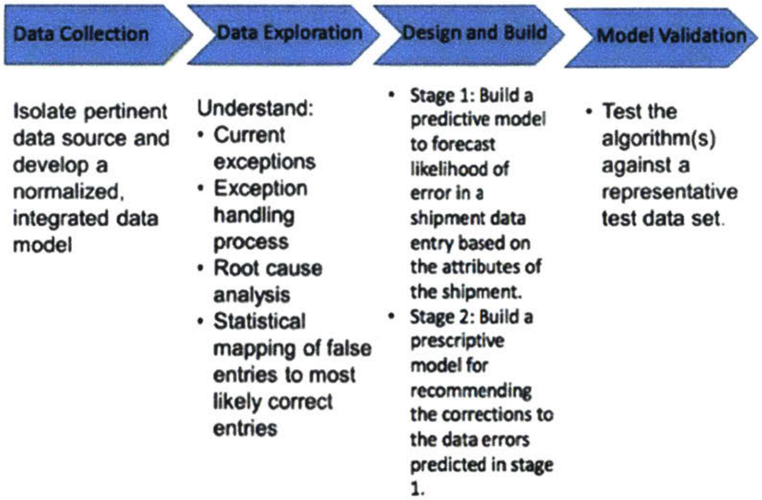

Data Collection
Data Exploration
Design and Build
Model Validation
Isolate pertinent data source and develop a normalized, integrated data model
Understand:
• Current exceptions
• Exception handling process
• Root cause analysis
• Statistical mapping of false entries to most likely correct entries
• Stage 1: Build a predictive model to forecast likelihood of error in a shipment data entry based on the attributes of the shipment.
• Stage 2: Build a prescriptive model for recommending the corrections to the data errors predicted in stage 1.
• Test the algorithm(s) against a representative test data set.

Figure 1: Four-phase analysis methodology

Furthermore, our approach uses a wide range of analytical techniques, which are outlined in Section 2.1 Analytics in Supply Chains, and are shown in Figure 2.

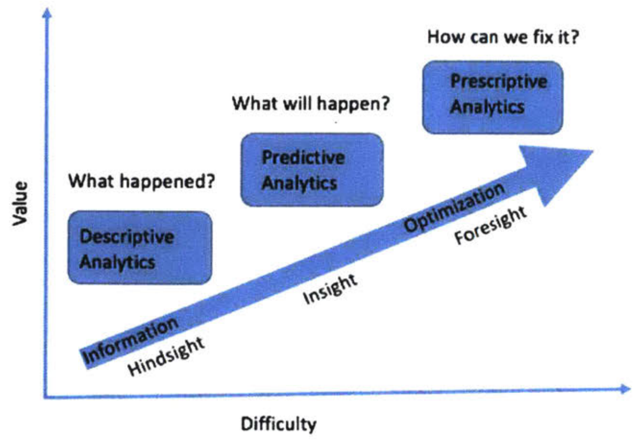


Figure 2: Analytics Maturity Model

(Adapted from Rose Technologies, 2013, Retrieved April 19, 2017, from http://www.rosebt.com/blog/descriptive-diagnostic-predictive-prescriptive-analytics.)

- **Descriptive analytics**: These techniques are used in the data exploration phase to allow us to form hypotheses about the root cause underlying the errors and to determine the variations in errors with time and shipment attributes. Descriptive analysis provides insights that help confirm or reject such hypotheses. The techniques that we will cover as part of our descriptive analysis range from basic statistical analysis of errors versus shipment attributes to clustering and regression trend analysis to better understand the historical patterns.
- **Predictive analytics**: Predictive analysis allows us to estimate which of the future shipments are likely to have erroneous data entry. In the context of our thesis, these techniques are used in the first stage of the model design and build phase, wherein we build models to predict these erroneous data entries based on shipment attributes. Predictive analytics techniques explored in this thesis include forecasting techniques, both time-series and regression analysis, and classification techniques.
- **Prescriptive analytics**: The final stage is the determination of what actions need to be taken in response to the predicted data entry errors. These techniques are key in the final stage of our analysis process. Machine learning algorithms can be formulated to suggest possible corrected data entries for the data

which we predicted to be erroneous. However, the use of this technique is contingent on the strength of the relations that we find using the descriptive analytics techniques.

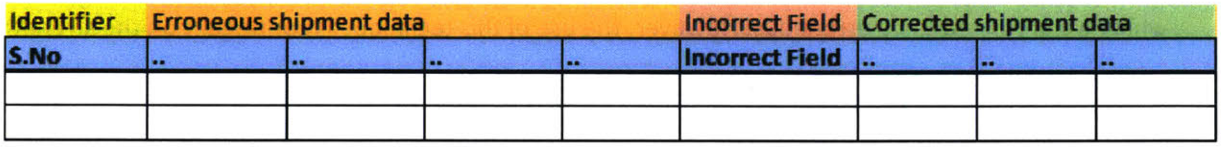
3.2 Data collection and cleaning

Before we can perform any analysis, we must first identify the data requirement, elicit the data from the sponsor, and structure it for use. In this stage, we focus on data elicitation, following an analysis of the data sources available and of their relationship to the problem that we analyze in this thesis.

3.2.1 Data sources

Our analysis relies on the availability of shipment data that includes shipment attributes, such as source and destination location, timestamps of events per shipment, information about the errors that occurred, the field that contained erroneous information, and the corrections made to impacted fields. We expect a large portion of this data to come from internal information systems that record transaction level data. However, details of how errors are rectified may need to be extrapolated from other data logs or be inferred from existing data.

Our approach to the analysis uses structured data stored in a data warehouse in a de-normalized state to support complex analytical queries. De-normalized data is read-optimized by grouping of distinct tables into one and by incorporating redundant copies of the data. This removes the need for performing joins on multiple tables when the complete information has to be read. For example: irrespective of shipments- country code, country region and city name pairs never change. Yet in a de-normalized table these are repeated multiple times within the data table. In contrast, normalized data tables will store different, but related, data in separate logical tables and relate them using certain attributes (keys). In order to support predictive and prescriptive analysis, the shipment data needs to be organized in the following structure, shown in Figure 3:



Identifier	Erroneous shipment data				Incorrect Field	Corrected shipment data		
S.No	..	..	..	..	Incorrect Field	..	..	..

Figure 3: Data Template for data consolidation

The data type for each field in the table will vary depending on the attribute that it describes. The key data types that we expect to encounter in the data structure are as follows:

- **Date-Time stamp**: Fields that record the time of data entry, time of event occurrence and FETA (Forecasted ETA).
- **Enumerated types (or categorical variables)**: Fields that denote whether an error occurred, the impacted fields, the source and destination locations, the event codes associated with the milestones, the reason code associated with the delay in shipments, and the error code related to the nature of the error in the initial entry.
- **Qualitative and continuous data-fields**: These include other shipment attributes, such as consignee, the nature of the product SKU and the price of the consignment.

In the future, in order to enhance the predictive capabilities of our model and to derive more realistic prescriptive recommendations, the set of data sources should be extended to include external data, such as weather information from RSS (Really Simple Syndication) feeds. This will be especially useful in determining the corrections for errors in fields that have a date-time data type. For example, in the case of a snow storm, the storm intensity (high, medium, low) will impact the expected delay in shipment. This will inform the correct ETA (estimated time of arrival) for an operational data error (although unavoidable). To enable a merger of diverse, multifarious data sources and the migration from a data warehouse to a data lake, a data discovery process is needed. This process will maintain the coherence of the multiple data sources and continually extend the set of data sources to add more meaning to the insights that the data provides (Shmueli et al., 2016).

A purely data-driven analysis of the root causes behind a system error can be challenging and requires additional data about the server usage. The following are some of the key aspects of the data that we need to investigate to understand the cause of the system errors better:

- **System response rate**: Analyze the trend of system response time. Missing data, particularly shipment level configuration data such as source or destination of the shipment which is retrieved from master data, but can be lost due to high system response time, can result in a system time-out.
- **Performance**: Trend of number of concurrent users and requests on the server at a given time allows us to identify period of peak load on the server. We can then examine the correlation between system error occurrence and the peak load periods.
- **Geographical reasons**: If multiple servers are used to host the system, we can analyze the correlation between the server location and the instances of system errors and/or system time-outs. This would be significant if the number of firewalls encountered for a particular server location are significantly higher than those at other sites.
- **Outages**: Data regarding instances of power or system outage or latency - for the servers that host master data - can be analyzed in relation to the frequency of system errors.

3.2.2 Data Preparation

The first key step before performing any descriptive analysis on the data is data preparation. The data available from the sources in scope includes transactional data, which captures shipment and event attributes, and error information. Data preparation will, therefore, follow five steps for the integration of these two critical pieces of information:

a) **Data Cleaning**: Before any analysis can be done on the data, it first needs to be cleaned and pre-processed. We begin this process by examining the data types of the variables in our table. To ensure that we can perform the required descriptive and predictive analysis on the data, some data fields may have to be converted into a different form. This can occur in two ways:

i. **Type-casting**: The value of the variable as viewed by the user remains the same, but the manner in which the system handles the field now changes. For example, a 'Price' field might be saved in the system as a string (sequence of characters). No mathematical operations can be performed on this field unless we type-cast it as a numeric type (integer or decimal).

ii. **Creating new fields**: We would use this approach when we need to convert the data about certain shipment attribute into a categorical variable. For example, we may add a column 'Shipment Mode' to denote whether the shipment is ocean or air freight. We derive this information based on whether the air freight ID column is populated or the ocean freight ID column is populated.

Once the necessary data structuring and formatting has been performed, we check the individual records for errors. The simplest check that we enforce is the not-null check for key values that serve as primary and foreign keys in our resulting table. Entries that have errors are either modified based on the data available in other fields, or are excluded.

b) **Current exceptions identification**: During this step we analyze all the exceptions that are either flagged in the data or are highlighted by the business user (but are not directly flagged in the data). We then drill down to the root cause of each broad exception type, and as an output of this step, we have an error categorization taxonomy. The error categories that we build upon in our thesis are "System Errors" and "Operational Errors". Errors that result in the violation of a system-enforced business rule and therefore do not allow a transaction to be successfully recorded are classified as system errors. These are easily detected by reviewing an error log as these are caught by an error management tool. Operational errors are incorrect data entries that do not violate any explicit business rule that the software system enforces. These manifest in the data in the form of multiple entries of the same information (with the requisite corrections made). We will elicit a log of these errors, which can then be integrated with the shipment transaction data. In case such a log is unavailable, we will analyze the instances of multiple entries against the same shipment waybill and shipment event pair to infer the cases of operational errors.

c) **Current exception handling process**: To inform the process of prescriptive analytics, we understand the process of exception identification and correction for all error categories. For system errors, identification rules mirror the business rules that are configured into the software system (where the shipment milestones are recorded). For operational errors, the identification process can be best understood through an interview process with the business users who record event milestone data and rectify any operational errors. Understanding the process for correction of either error category requires analysis of the data complimented by interviews with users.

d) **Data error-correction mapping**: Finally, we restructure the shipment data into a format where we connect the initial data records with their corrected versions, along with meta-data regarding the nature of error. We employ a binary variable to flag records which contain exceptions. We retain all attributes of the initial and corrected data to inform our analysis.

e) **Dimensionality reduction**: During this step, we will reduce the number of variables to only those that are needed, transform the variables, add variables such as data summaries as needed and use dimension reduction techniques such as principal component analysis (PCA) to create a new set of variables which are weighted averages of the original variables. These new variables are uncorrelated and provide most of the critical information about the initial variables. Therefore, by using this new subset of variables, we are able to reduce total number of variables and hence dimensions in our analysis. This step is necessary for mitigating the 'Curse of Dimensionality', wherein the addition of variables in a multivariate analysis makes the data space increasingly sparse. Classification and prediction models then fail, because the available data is insufficient to provide a useful model across so many variables. Other methods of dimensionality reduction include manual elimination/consolidation of variables using statistical aggregates by leveraging domain knowledge of experts.

3.3 Model Development

During this phase of the thesis, we analyze the data that we gathered during the first phase and prepared for analysis. The model development involves the following main steps: descriptive, predictive, and prescriptive analysis.

3.3.1 Descriptive analysis

For the de-normalized data, we perform detailed descriptive analysis to identify potential correlations between a subset of shipment attributes and the likelihood of error occurrence. We segregate our analysis for system and operational errors to find drivers of each separately. Finally, we combine the SKU and pricing information (if available) to quantify the magnitude of impact associated with each error - in terms of both time and money.

We conduct our descriptive analysis by identifying the key hypotheses to be tested and then performing descriptive analysis to accept or reject the hypotheses. We categorize the hypothesis that we test into the following categories:

a. **Temporal hypotheses**: In this set of hypotheses, we examine the relationship between error occurrence frequency, error impact intensity and time. Specifically, we want to determine whether errors - both system and operational - are more frequent and/or severe at certain times of the year. We make a careful distinction between the absolute number of error occurrences over time and the number of error occurrences relative to total number of data entries recorded over time.

b. **User and consignee driven hypotheses**: In these set of hypotheses, we examine the relationship between error occurrences and the user performing the entry or the consignee for the shipment.

c. **Geo-spatial hypotheses**: These hypotheses examine the relationship of error occurrence with source, destination country and event location.

d. **Other hypotheses about corrections and reason codes**: These hypotheses look at the patterns of error occurrence for specific event codes, shipments and reason codes. We also examine the fields that are most affected by a change (update or correction).

3.3.2 Predictive Analysis

Next, we focus on using the findings from our descriptive analysis to build a predictive model to forecast the likelihood of error in a shipment data entry, based on the attributes of the shipment. The model builds upon the hypotheses made during our descriptive analysis. We begin by dividing the data into two datasets – training data and validation data. We use a combination of the following predictive analytics techniques:

- **Classification and regression trees (CART)**: We use classification techniques to classify shipment event records as either having errors or not. This is a data driven approach, the results of which are easy to interpret. In this technique, we recursively partition the independent variable space by using the training dataset (Shmueli et al., 2016). The result is 'n' distinct rectangular regions, after we perform this process for the entire training data set. Having thus grown a classification tree, we can test its performance using a testing data set. We have to ensure that the model is not over-fit, which can result in a scenario where the tree begins to model the noise. One way to avoid over-fitting is to prune the tree. Therefore, we use the validation data set to determine when to stop the partitioning process. At this point, we have the classification rules for the pruned tree. We can use these to classify the future records. For instance, we can categorize records as having a high error probability by classifying based on shipment attributes. Regression trees follow a similar approach but are used for numerical variables and allow us to predict a numerical value. The advantage of the CART technique is that it is easily supported by several off-the-shelf software such as JMP. Further, it is useful for modeling nonlinear, non-parametric relationships (unlike linear regression analysis). Some of the disadvantages include requirement of a large training data set. The technique also has a tendency to favor predictors that have a greater number of split points.

- **Time-series based forecasting**: For time series based forecasting, we can use one of two methods, or a combination of both: smoothing and multiple linear regression models. Of these two, smoothing is a more data-driven approach. However, in both of the two approaches, we begin by dissecting the time series into its four components: level, trend, seasonality, and noise. We do this by graphing a time-plot, which in its simplest form is a line chart with temporal labels along the horizontal axis. As recommended by Shmueli et al. (2016), we weight the two approaches - data driven and model driven forecasting methods - by assessing the global and local patterns. When the data demonstrates clear pattern throughout the series (such as a linear trend), model-based forecasting methods can be beneficial. For local patterns which are likely to vary, a data-based model which learns quickly from limited data is preferable. As in the case of CART, we segment the data into two datasets - training and validation data. This is done to avoid the risk of overfitting and to assess the performance of the model before using it. The difference between CART and time series modeling however lies in the fact that the entire data set - training and validation dataset - is used to train the model. This is done because the data from more recent time contains valuable information about the future. Without using the validation dataset in developing the model, we will be using the training model to forecast further out into the future, which will impact the model performance. Regression-based forecasting models range from models with trend (linear, exponential or polynomial), models with seasonality, models with trend and seasonality, to autocorrelation and ARIMA (autoregressive integrated moving average) models. Autocorrelation models and ARIMA models incorporate the dependency between the individual observations. On the other hand, smoothing is a data-driven approach, which is based on averaging over multiple periods in order to reduce the noise. Variations of smoothing include using averages, moving average, centered moving average, naïve forecasts, simple exponential smoothing, and advanced exponential smoothing. Exponential smoothing uses weighted averages but assigns greater weight to more recent values. Advanced exponential smoothing uses series with a trend (and seasonality).

- **Regression analysis**: Linear regression models can be used for fitting data for the purpose of inference as well as for the purpose of prediction. A multiple linear regression model can be used for fitting a relationship between a dependent variable Y and a set of predictors X_i. The relationship can be expressed in the form:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$

where β_0, ..., β_p are coefficients and ϵ is the error term.

A critical decision to be made while performing multiple linear regression is determining how many variables to include in the model, as it requires a tradeoff between variance and bias. Having too many variables in the model leads to higher variance, and having too few results in bias. Along the same lines, the principle of parsimony states that more insights can be obtained about the impact of predictors on dependent variables when we use fewer predictors. Additionally, we must be careful not to include predictors that are correlated, as their presence can make the regression coefficients unstable. Domain knowledge must be used while reducing number of predictors. Other reasons for eliminating a predictor include high correlation with an existing predictor, large number of missing values and high cost of data collection (for a predictor value). Some software (such as JMP by SAS) allow us to assess the statistical impact (using p-Value) of the elimination of a predictor on the regression model. The predictors which have a p-Value greater than a certain threshold value (as determined by the modeler) can be excluded. R^2 (Coefficient of determination) and adjusted R^2 are commonly used to assess the performance of the selected set of predictors. R^2 denotes the proportion of the variance in the dependent variable that is explained by the predictors. Since R^2 always increases with the addition of more predictors, adjusted R^2 penalizes the increase in number of predictors used, by the relation:

$$R_{adj}^2 = 1 - (1 - R^2) * \frac{n-1}{n-p-1}$$

Where n is the sample size and p is the number of predictors. R^2 is the coefficient of determination. Higher R^2 and R^2_{adj} are desirable.

- **K-Nearest neighbors (K-NN)**: This technique can be used for classification of a categorical outcome or for prediction of a numerical outcome. This technique is based on identifying 'K' records in the data set that are similar to the new data point that we wish to classify. The 'K-Nearest neighbors' algorithm does not make any assumption (of linearity, for example) about the relationship between predictors and class membership. We look at the Euclidian distance between two records for determining the proximity to a neighbor. The K-NN algorithm can be extended to continuous variables. We take the average response value of the K-nearest neighbors to determine the prediction. The average used can be a weighted average, with weights being inversely proportional to the distance from the point at which the prediction is required.
- **Naive Bayes classifier**: Unlike the methods listed above, Naïve Bayes classification is a predictive method that performs well for categorical variables. It helps answer the question - 'What is the propensity of belonging to a particular class?' (Shmueli et al., 2016). The Naïve Bayes method is a variation of the complete (or exact) Bayes procedure. The complete Bayes method uses the concept of conditional probability to determine the probability of a record belonging to a particular class given its predictor values. To classify the record, we compute its probability of belonging to each of the '**m**' possible classes and then assign it to the class 'i', where it has the highest probability of belonging. Alternatively, a cutoff probability can also be used as an assignment rule. The limitation of this method is that it does not scale well and as the number of predictors increases, the probability of finding a

match (of all predictors for a new record to existing records) reduces. The Naïve Bayes method therefore computes the conditional probability for each predictor, rather than predictor-profiles per record. The Naïve Bayes method is simple, computationally efficient and shows reasonably good classification results. However, it relies on the availability of a large dataset to perform well. Additionally, for a new predictor value, it will classify the probability of the predictor occurring as zero.

- **Neural networks**: Neural networks are based on a model of the biological activity in the brain and have a high prediction accuracy (Shmueli et al., 2016). Neural networks combine input information in a flexible manner and try to learn as much as possible about the relationship between the input data and the response variables. Most Neural networks have a multilayer feedforward network. These are networks which consist of an input layer comprised of nodes that simply accept the input values. The successive layers of nodes receive the input from the previous layers. The last layer is called the output layer. All layers between the input and output layer are called hidden layers. A feedforward network has a one-way flow and there are no cycles. JMP (the software used in our analysis) splits the data into training and validation datasets automatically. The 'Random Holdback' value is used to specify the proportion into which the dataset will be randomly classified (SAS Institute Inc., 2017a). Further, a 'KFold' value can be specified to split the original dataset into K subsets which are then used to validate the model, and thereby to select the best model. The KFold method however, is better suited for smaller datasets and will not be used in our analysis. Activation methods are used to define the structure of the hidden layers of the neural network. JMP offers the following activation methods to transform the linear combination of the input variables (X) as shown in Table 1:

Table 1: Activation Functions for Neural Nets

Activation Function	Description
TanH	The hyperbolic tangent function is a sigmoid function which transforms values to lie between (-1 ,1). The function is of the form: $\frac{e^{2x}-1}{e^{2x}+1}$ Where x= combination of the input variables (X)
Linear	Linear combination of X is not transformed. For nominal or continuous variables, the model reduces to a logistic regression.
Gaussian	Used for Bayesian function behavior or when the response surface is normal in shape.

One of the disadvantages of Neural networks is that they have a tendency to over-fit the data. This can be overcome by using a penalty parameter and requiring model cross-validation, as is done automatically in software like JMP. The neural fitting algorithm computes the optimal value of the penalty factor and the weights (used to combine the various layers using the activation method) to minimize errors. Neural-networks have the advantage of high prediction accuracy with high tolerance to noise and the ability to model complicated relationships between predictors and a response. This approach is, however, computationally expensive and requires large training datasets.

- **Other techniques**: The other techniques that we consider in our analysis include Logistic regression, and Discriminant analysis. We use Logistic regression for classification of records as having errors or not. Discriminant analysis is a model based approach to classification, which is similar to logistic regression in terms of output and uses Least square estimates/ Euclidian distances to determine the binary predicted value.

3.3.3 Prescriptive Analysis

During this stage, we build a prescriptive model for recommending the corrections to the data errors predicted in stage 1. The approach that we will use is a combination of predictive analysis and business rules to determine appropriate actions. The business rules are based on inputs provided by the business teams, as well as on the correction patterns that we observe in the data. For example, if entries with the source incorrectly entered as "HKKG" are corrected with a revised source entry of "HKG", this would form a business rule. Machine learning can be used to arrive at a set of business rules. This is not in the scope of our thesis.

We evaluate the feasibility of the phrasal statistical machine translation technique proposed by Ehsan and Faili (2013) to our dataset. We will select the key fields that have a high incidence of error and apply the SMT technique to them. These fields are identified using our descriptive and predictive analysis. The approach predicts the instances of error and recommends the best alternatives for correction.

In essence, our methodology explores descriptive analytics in order to isolate the source of the shipment errors, predictive analytics to infer the probability of error occurrence, and prescriptive analytics to recommend appropriate corrections for these errors. The next section applies this framework to a case study for one of Damco's customers.

3.4 Model Validation

Model validation is used to assess the performance of the predictive model using the validation data partition. This helps us select a model that performs the best at predicting the erroneous entries, as well as prescribing the corrections. We begin by selecting the performance metrics that we use for analysis - which could include average absolute error (AAE) and root mean squared error (RMSE). We check the method performance against a cutoff value by plotting the ROC (Receiver output characteristics) curve (Shmueli et al., 2016). The ROC curve is a graphical plot of the true positive rate (Sensitivity) against the false positive rate (calculated as (1 - Specificity)). Sensitivity is the ability to correctly identify the occurrence of a classifier and Specificity is the ability to correctly identify the absence of the identifier. Curves closer to the top left of the graph indicate a better model performance, whereas curves closer to the diagonal indicate poor performance. For instance, Figure 4 shows the ROC curves for 4 different attributes in green, orange, red and blue. The curve represented by the green line is closest to the top-left edge of the graph, followed by the orange curve. The predictive performance of the model for these two attributes is very good. The model performance for the attribute in red is better than that for the attribute represented by the blue. The area under the graph (AUC) is a commonly used metric to compare the model performance for various attributes. The AUC for the blue curve is shown in Figure 4.

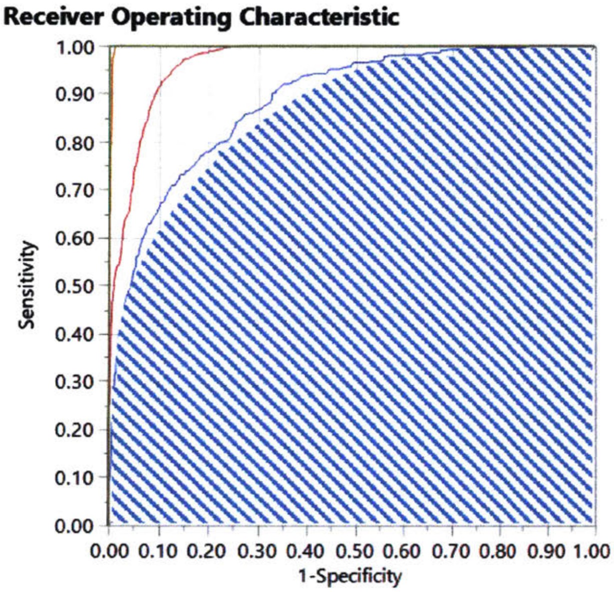


Figure 4: Illustrative ROC graph

The distinction between model performance and the goodness-of-fit of the model is an important one to bear in mind. The goodness-of-fit, which is typically determined by measure like R^2 (Coefficient of determination) or standard error (if estimated), gauges how well the model fits the training dataset. Predictive performance assesses how well the model performs when it is applied to new records, the validation dataset (Shmueli et al., 2016).

Mean-based measures such as MAE (Mean Average Error), MAPE (Mean Average Percentage Error), MAD (Mean Absolute Deviation) and RMSE (Root Mean Square Error) can be used to measure the residual for a record. The residual is defined as the difference between the predicted value and the actual observed value. However, the mean-based methods are influenced by the presence of outliers, and it is recommended that we supplement residual analysis with median-based measures or by plotting the data into a histogram.

Table 2: Goodness-of-Fit metrics

Measure	Mathematical Formulation	Description
R^2 - Coefficient of Determination	$R^2 = 1 - \frac{(Sum\ of\ squares_{residuals})}{Sum\ of\ Squares_{total}}$	Proportion of the explained variability in the model
R^2_{adj} - Adjusted Coefficient of Determination	$R^2_{adj} = 1 - (1 - R^2) * \frac{n-1}{n-p-1}$	Proportion of explained variability in the model, with the inclusion of penalty on the number of predictors.
Entropy R^2	$Entropy\ RSquare_{Training} = 1 - \frac{\log(Likelihood^{Full}_{Training})}{\log(Likelihood^{Reduced}_{Training})}$	A measure of fit which compares the log-likelihoods from the fitted model and the constant probability model. Larger values indicate better fit.
MAE - Mean Average Error	$MAE = \left(\frac{1}{n}\right) * \sum_{i=1}^{n} \lvert e_i \rvert$	Provides the magnitude of average error
MAPE - Mean Average Percentage Error	$MAPE = \left(\frac{100}{n}\right) * \sum_{i=1}^{n} \frac{\lvert e_i \rvert}{\lvert y_i \rvert}$	Provides the percentage by which the predicted values vary from the actual values
RMSE - Root Mean Square Error	$RMSE = \sqrt{\left(\frac{1}{n}\right) * \sum_{i=1}^{n} e_i^2}$	RMSE has same units as predicted variable. For partitioned data, RMSE is typically computed using the validation dataset.
Sum of $Squares_{residuals}$	$\sum_{i=1}^{n} e_i^2$	Residual sum of squares
Sum of $Squares_{total}$	$\sum_{i=1}^{n} (y_i - \bar{y})^2$	Total sum of squares proportional to the variance of the data

Where:

'n' is the sample size and 'e_i' is the prediction error calculated as the difference between actual (y_i) and predicted value($\hat{y}_i$) such that $\mathbf{e_i = y_i - \hat{y}_i}$

Further, the likelihood function is the product of probability density functions at the observed data values (SAS Institute Inc., 2017). For convenience, the maximization of the likelihood is reformulated as the minimization of the negative log of the likelihood function (-Log(Likelihood)).

Full (in $\log(Likelihood_{Training}^{Full})$) describes the negative log-likelihood for the complete model.

Reduced (in $\log(Likelihood_{Training}^{Reduced})$) describes the negative log-likelihood that results from a model with only intercept parameters.

To assess the performance of a classifier we need to assess the probability of a misclassification error. A common framework used to assess the misclassification error is the classification matrix which plots the actual values against the predicted value in a matrix. All the correct values lie along the diagonal (Shmueli et al., 2016). The classification matrix for the validation dataset needs to be analyzed to assess the true performance of the predictive model. Further, the classification matrix for the training and validation dataset can be compared to identify any signs of overfitting in the model. Many accuracy measures can be derived from the classification matrix, such as estimated misclassification rate given by:

$$err = \frac{Count\ of\ misclassfied\ records}{Total\ number\ of\ records}$$

(Shmueli et al., 2016).

The naïve rule ignores all predictor information and classifies all incoming records as belonging to the most prevalent class. However, for cases where prediction of a dependent variables is more important in one class than in another, we need to incorporate sensitivity and specificity into our accuracy analysis. For instance, in the case of data errors, it may be more important for us to correctly predict a record which will have an operational or system error than for us to predict a record which will have no errors. Sensitivity is the ability of a classifier to detect the important class correctly. Specificity is the ability of the classifier to rule out the less important members correctly. We can then plot an ROC curve against a desired cutoff value (between 0 and 1). The ROC curve plots the pair of Sensitivity and (1 - Specificity) values. Curves closer to the top-left represent models with better performance. A commonly used metric to assess the performance is AUC, (Area Under the Curve) which ranges from 1 (perfect discrimination between the classes) to 0.5 (classification performance same as a naïve rule).

4. Case Study Analysis

4.1 Damco Operational Context

Damco is a leading provider of freight forwarding and supply chain management services. Damco has a large customer base and aims to provide high value-added solutions to its customers by investing in proactive forecasting analytics. As part of its freight forwarding operations, Damco tracks, on average, a set of 8 industry standard milestones per shipment for its customers. These milestones vary depending upon the mode of the transportation: air, track, ocean or parcel. For some customers, such as the customer whose data is used in this study, there is an additional set of non-standard milestones that they want tracked (manually). The customer in question subscribes to Damco's freight forwarding services, and not their (more mature) Supply chain management (SCM) services. Yet, the customer requires granular milestone tracking. This has forced Damco to track the milestones manually.

Further, the expected turnaround time also varies significantly by the mode of transportation. For example, for air shipments, the data entry for the milestone corresponding to 'wheels-up' or departure from source location must be completed within 6 hours of the event occurring. This data will have little value if it were to be updated 24 hours later, when the shipment may have already arrived at the customer's facility. Therefore, customers require increased supply chain visibility, faster response to supply chain disruptions, and insights into their trade lane activities. Account teams at the freight forwarder, on the other hand, want to reduce manual processing and want the ability to proactively keep the customers informed of the shipment milestones. IT teams at the freight forwarder want to prioritize technical feasibility for creating a scalable and maintainable solution. Most of the data entry of the shipment milestones is currently done manually by Damco. This renders the process susceptible to data entry errors, which can have severe implications for Damco's customers, unless the errors are corrected promptly.

Ensuring data accuracy and veracity is a critical challenge faced by all global organizations whose business requires physical movement of goods. The complexity of this challenge is further amplified in the case of logistics service providers and freight forwarders like Damco.

4.2 Data Collection

The data collection process for Damco was performed in close collaboration with the IT and the business teams. This was necessary for understanding the pain-points felt by the business on the one hand and the data limitations and constraints on the other.

4.2.1 Data sources

The first stage of our project was the data gathering stage, during which we isolated pertinent data sources. Based on the attributes of the data and the problem at hand, we developed a data model. The model was then adjusted to account for any differences in the time horizon between multiple data sources. Additionally, the data model had to be an integrated image of the multifarious data sources. The data sources that we have included in our analysis are shown in Figure 5.

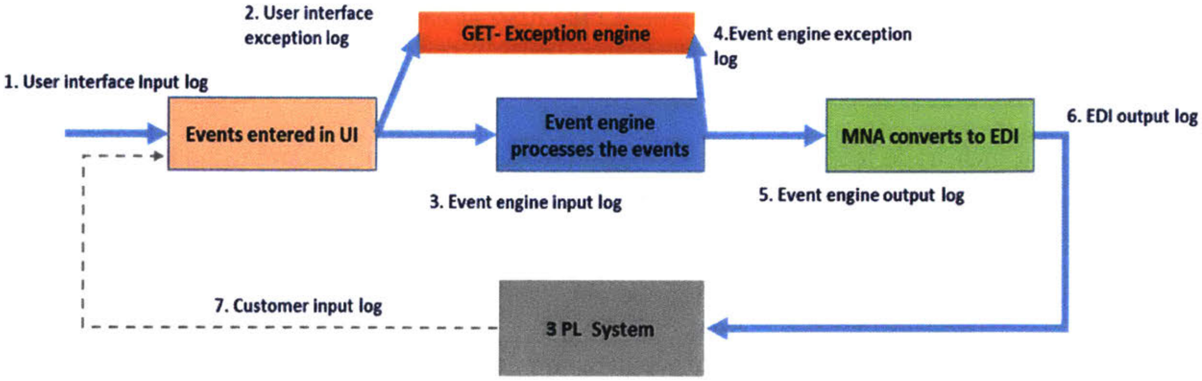

2. User interface exception log
GET- Exception engine
4.Event engine exception log
1. User interface Input log
Events entered in UI
Event engine processes the events
MNA converts to EDI
6. EDI output log
3. Event engine input log
5. Event engine output log
7. Customer input log
3 PL System

Figure 5: Inter-system Data Flow

1. **User Interface Input log**: The data entry log of every shipment milestone entered into the milestone tracking system (known as the User Interface system or UI) between July 2015 and December 2016. This log amounts to roughly 130,000 shipment-events with 45 attributes per event, including source country and timestamp data.
2. **User Interface Exception log**: Some of the data events entered in the UI will violate the system rules. These will be separated into the user interface exception log. For the time horizon spanning from July 2015 to December 2016, we have 1440 such events.
3. **Event Engine Input log**: This is the log of all the remaining events that are successfully received at the event engine stage. This log is a subset of the user interface input log which excludes all the events where an exception happens in the IT system.
4. **Event Engine Exception log**: The next system where the data is sent is the event engine. However, certain events may violate the event engine's rules and these will be captured in the Event engine exception log. For the 17-month timeframe, i.e. July 2015 to December 2016, we have only 7 such records.
5. **Event Engine Output log**: The remaining data points are successfully sent as an output from the event engine to the EDI system.
6. **EDI Output log**: Based on certain business logic, the incoming data from the event engine is trimmed and condensed into EDI (Electronic Data Interchange) messages which are sent to the customer. For the 17-month duration, we have ~70,000 such EDI messages. Some of the rules that are applied include the exclusion of redundant entries and selection of the most recent version of a milestone update per shipment within a 10 or 15-minute window.
7. **Customer Input log**: This is the log of the EDI messages which are successfully received by the customer.

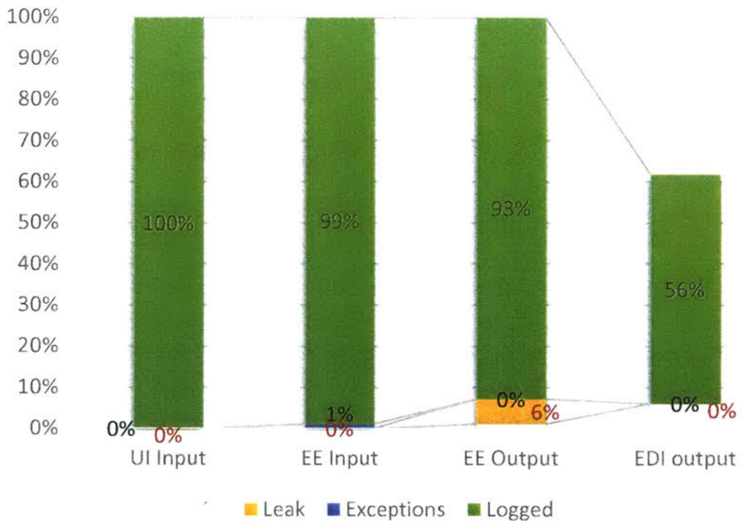


Figure 6: Data propagation through Damco systems

The sequential propagation of the data through the Damco systems referred to as UI (user Interface), EE (Event Engine) and EDI system is shown in Figure 6. Note that although data logs in points 3, 5 and 7 above are expected to have all the data events which are not captured in the exception logs (2, 4), this is not the case, since we observe certain leakage in the data for unknown reasons.

4.2.2 Data Cleansing and Organization

Before an analysis could be performed on the data, we integrated the data received from sources 1 to 5 and from sources 6 to 7 into two de-normalized tables in a MySQL database. The two tables were named **System_Input** and **EDI-Output** tables and were left in a de-normalized state for simplicity and to ensure easy integration with statistical tools. Next, we joined these two tables using a Left-inner join, to merge additional information from the EDI-Output table with the System_Input table into a final table titled **System_Milestones_Visualization**. Further, an additional attribute was added to the System_Milestones_Visualization table to act as a link between the updates made to the event- shipment data pair. This field- labeled "Last_Edit_UI"- was used to connect the multiple records that correspond to the same shipment for the same event together.

Table 3 shows a description of the fields in the System_Milestones_Visualization table.

Table 3: Data Dictionary

Attribute	Data type	Data Description
LID_UI_Input	VarChar (P KEY)	Primary key - an identifier generated by the logging system to uniquely identify each row in the database.
SSOURCE	VarChar	The source - which corresponds to the client account
SEVENTEXTERNALCODE_UI	VarChar	The unique code associate with each shipment milestone
HEVENT_UI_Input	DateTime	The timestamp which is manually recorded by the user for each event.
HCAPTURED_UI_Input	DateTime	The current timestamp automatically recorded at the time of data entry
SLOCATIONCOUNTRYCODE_UI_Input	VarChar	Country code for the country where the event takes place
SLOCATIONCITYCODE_UI_Input	VarChar	City code for the country where the event takes place
SSODOCUMENTNUMBER_UI	VarChar	This field records the shipment document (waybill) number in case the ShipmentDocNo_UI_Input field is blank due to a system error.
SCONSIGNEEBECOUNTRYCODE_UI_Input	VarChar	Consignee country code
SCONSIGNEEBECODE_UI_Input	VarChar	Consignee code
SCAPTUREUSER_UI_Input	VarChar	User Identifier of the user making the data entry
SEVENTEXTERNALREASON_UI_Input	VarChar	Reason code in case of a delay. This is associated with certain events only, such as SD (shipment delay).
Last_Edit_UI	VarChar (F-Key)	Foreign Key field that links current record to the last update for the same Shipment-Event pair.
Status_UI	VarChar	Assigns a status to each row- "Initial Entry", "Update", "Correction", "Redundant"
ShipmentMode_UI_Input	VarChar	Either "Ocean" or "Air" assigned. Air also includes "Parcel" and Ocean also includes "Truck".
ShipmentDocNo_UI_Input	VarChar	Waybill number
HasUIInputError	VarChar	Binary flag that indicates if the record has a system error (indicated by a value of "1").
UIErrorCode	VarChar	Error code of the system error if HasUIInputError=1
UIErrorDesc	VarChar	System Error description if HasUIInputError=1
TS_UIErrorCapture	DateTime	Timestamp of system error detection if HasUIInputError=1
EventEngineInputID	VarChar	Unique Identifier created when the data is transmitted from UI system to event engine.

EventEngineEventCode	VarChar	The internal event code in the event engine that maps to the unique event code recorded in the UI system for each system milestone.
TS_ReceivedAtEE	DateTime	Timestamp of data receipt at event engine
LMETAHBL_EE_Input	VarChar	Internal meta code - ignored in our analysis
LEVENTOBJECT_EE_Input	VarChar	Internal object code - ignored in our analysis
SCONSIGNEEBECOUNTRYCODE_EE_Input	VarChar	Consignee country code recorded in Event engine at receipt. Should be the same as the value recorded in UI.
SCONSIGNEEBECODE_EE_Input	VarChar	Consignee code recorded in Event engine at receipt. Should be the same as the value recorded in UI.
SLOCATIONCOUNTRYCODE_EE_Input	VarChar	Event location country code recorded in Event engine at receipt. Should be the same as the value recorded in UI.
SLOCATIONCITYCODE_EE_Input	VarChar	Event location city code recorded in Event engine at receipt. Should be the same as the value recorded in UI.
HasEEError	VarChar	Binary flag that indicates if the record has a system error (indicated by a value of "1") at the event engine.
EEErrorCode	VarChar	Error code of the system error if HasEEError =1
EEErrorDesc	VarChar	System Error description if HasEEError =1
TS_EEErrorCapture	DateTime	Timestamp of system error detection if HasEEError =1
EventEngineOutputID	VarChar	Unique ID created when the data is transmitted from the event engine.
TS_OutputFromEE	DateTime	Timestamp of data transmission from the event engine
LMETAHBL_EE_Output	VarChar	Internal meta code - ignored in our analysis
SLOCATIONCOUNTRYCODE_EE_Output	VarChar	Event location country code recorded in Event engine at transmission. Should be the same as the value recorded on receipt.
SLOCATIONCITYCODE_EE_Output	VarChar	Event location city code recorded in Event engine at transmission. Should be the same as the value recorded on receipt.
SCONSIGNEEBECOUNTRYCODE_EE_Output	VarChar	Consignee country code recorded in Event engine at transmission. Should be the same as the value recorded on receipt.
SCONSIGNEEBECODE_EE_Output	VarChar	Consignee code recorded in Event engine at transmission. Should be the same as the value recorded on receipt.
ShipdocNo_EE_Output	VarChar	Waybill number transmitted from the event engine. Should be the same as the waybill number recorded in UI.
OriginCity	VarChar	Origin city available from the data retrieved from the customer
OriginCountry	VarChar	Origin country available from the data retrieved from the customer

OriginName	VarChar	Origin facility/ site name from the data retrieved from the customer
DestinationCity	VarChar	Destination city available from the data retrieved from the customer
DestinationCountry	VarChar	Destination country available from the data retrieved from the customer
DestinationName	VarChar	Destination facility/ site name from the data retrieved from the customer
DRegion	VarChar	Destination region
ORegion	VarChar	Origin region
milestoneDesc	VarChar	Description of the milestone - unique for each event milestone as specified in the field "SEVENTEXTERNALCODE_UI"
shipmentmode_4PL	VarChar	Shipment mode as recorded by the customer system
EditedField	VarChar	For the fields which are corrections or updated version of previous entries for the same shipment- milestone pair, this field denotes which field was modified.
ShipmentDocNo_Implied	VarChar	This field applies an OR condition on the two possible sources of the waybill number: "ShipmentDocNo_UI_Input" OR "SSODOCUMENTNUMBER_UI"
IsFinalEntry	Binary	Flag to indicate whether a record is a final version of a shipment- milestone pair (denoted by value of 1) or is over-written by a subsequent shipment-event pair.
TimeSinceLastEdit	Decimal (hours)	Hours elapsed since the time the same shipment-event data was updated.
TimeSinceFirst	Decimal (hours)	Hours elapsed since the time the same shipment-event data was first entered.

4.2.3 Data Exceptions

The current state analysis is critical for ensuring that our understanding of the current operating process is accurate and complete. This understanding was achieved through detailed data and process walkthroughs from the IT and the Operations team. This review was extended to include the study of the manual process involved in identification and correction of exceptions at present.

The next stage of the current state analysis zoomed in on the data exceptions and we analyzed of the root causes underlying each exception type. All data exceptions can be categorized into one of two types – missing data or false data - as shown in Figure 7. The effects or impacts of the errors vary depending on whether the error is caught by the system (immediate system errors) or not (delayed error identification by customer).

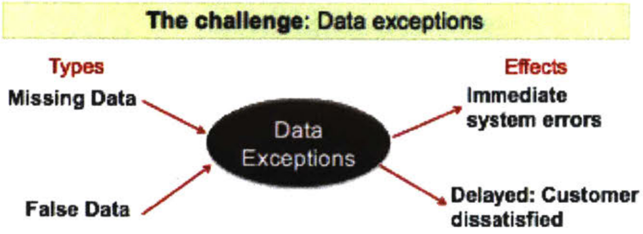


Figure 7: Data exceptions

To prioritize one exception type over another, we perform a descriptive analysis to identify the key correlations between error prone data and the attributes of the shipments that may be driving these errors. For example, we look at the correlation between the user making the event entry and the instances of errors. A subset of the shipment attributes which are found to have the highest correlation with the likelihood of error in the shipment entry are then isolated for the subsequent stages of our thesis.

4.3 Model Development

The model development in our case study is classified, along the same lines as our overarching approach, into the following 3 sections:

1. Descriptive analytics
2. Predictive analytics
3. Prescriptive analytics

4.3.1 Descriptive Analytics

The descriptive analysis phase involved three key steps:

- Data exploration
- Descriptive evidence of hypothesis
- Classification using K- Means

4.3.1.1 Data exploration

a. **Transaction status**: During this step we analyzed the transaction data (corresponding to each row) at hand and categorized all the entries in the data set into 4 distinct types. This 'type' information was recorded in the field titled 'Status_UI'. All the event-milestone entries for a given shipment can be categorized into one of the following four Status categories. Initial Entry: the first recorded instance of a given milestone for the shipment. Redundant: the same milestone is updated for the same shipment, without any change in pertinent information. Update: for certain milestones ('AG' and 'SD') where regular updates against the same milestone-shipment pair are expected; the second instance onwards of this event is recorded as an 'Update'. Correction: for all remaining events (other than 'AG' and 'SD'), if any significant data field is changed, the status is recorded as Correction.

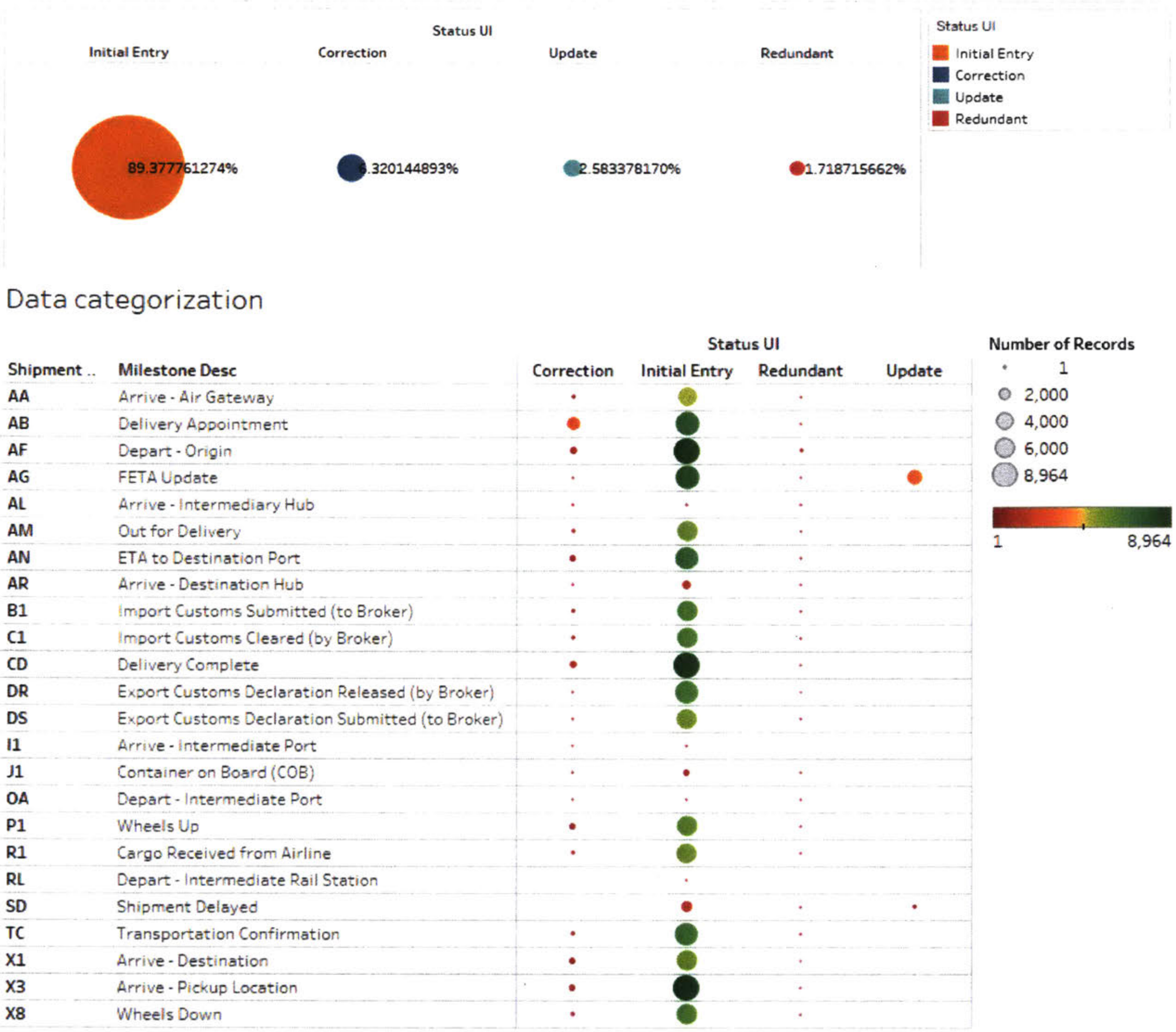


Figure 8: Data Categorization by Status of each record

From Figure 8 above, we can see that most of the milestone entries are first instances of the milestone-shipment pair (Initial entry). The maximum number of records in this category corresponds to the events X3 (arrived at pickup) and AF (pickup). This can be interpreted from the color and size of the circles in Figure 8. Although the number of redundant entries in the entire data set is fewer (1.7%), a large number of the redundant entries is concentrated in the event AF (pickup). Updates are concentrated in the AG (FETA-ETA update) event. Most of the corrections are taking place for the event AB (Delivery appointment or appointment confirmed).

b. Trade between countries (Source-destination pairwise mapping)

We have mapped part of the data (59.6%) to the source country and the destination country for the given shipment. The reason for only partial mapping is the limited data availability.

Based on the comparison of the source destination pairs (Figure 9), we find that most of the shipments during the transaction period Feb 2016 to Dec 2016 are from China to the US (45.34%). This is followed by transactions from China to Netherlands (30.26%). It is worth noting that over 99% of the transactions (events-shipments) are recorded with the source country as China.

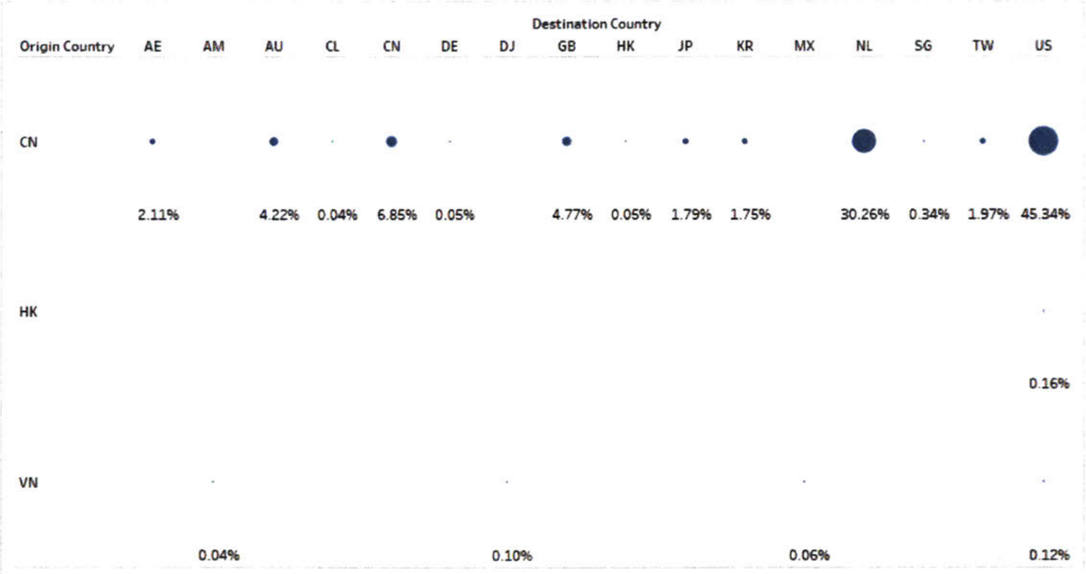


Figure 9: Transactions between Origin-Destination pairs

Similarly, for the destination countries, we see that the majority of the transactions are for the destination country US (45.6%). This is shown in Figure 10.

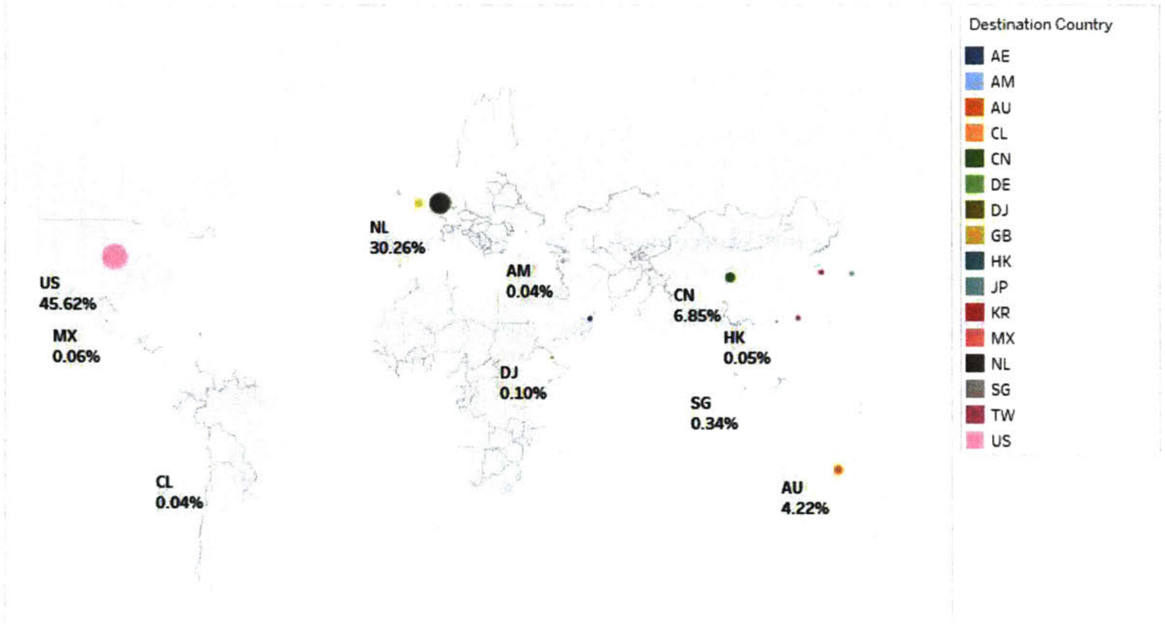


Figure 10: Transaction distribution to Destination countries

c. Shipment milestones represented in the transaction set

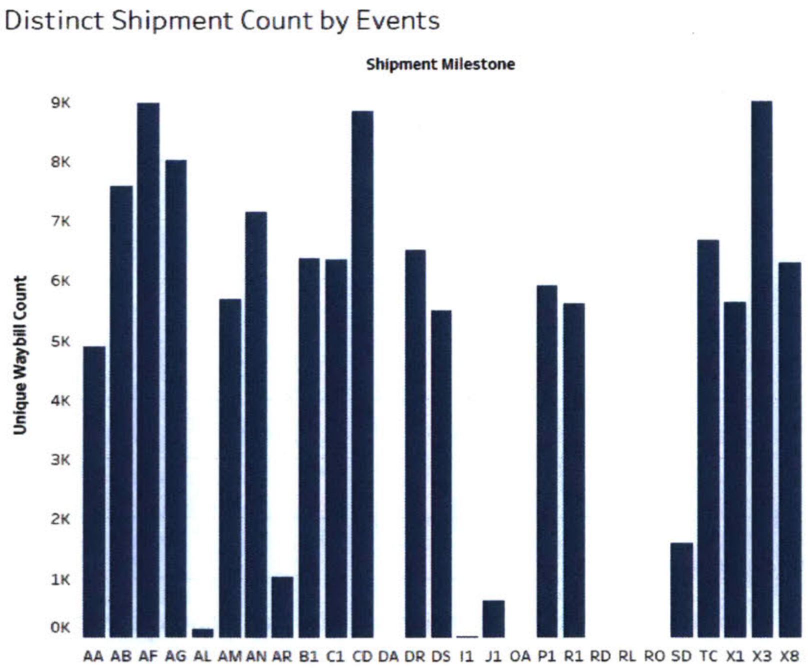


Figure 11: Plot of number of distinct shipments mapped against each event

The maximum number of distinct shipment waybills correspond to **X3** (Arrived at pickup), followed by the event **AF** (Pick up) and **CD** (Delivered/IOD per terms). This is shown in Figure 11.

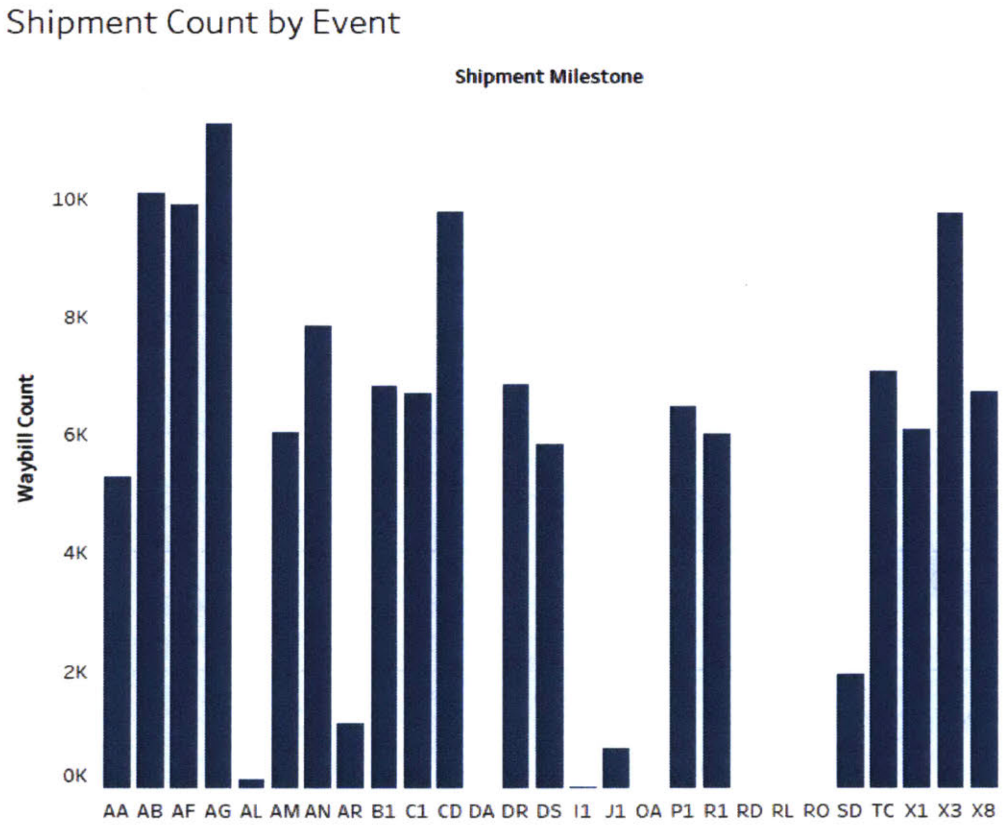


Figure 12: Plot of number of records mapped against each event

However, when we look at the same data and analyze multiple entries against the same milestone for the same waybill (as shown in Figure 12): **AG** (FETA-ETA to door) has the maximum number of entries, followed by **AB** (Appointment confirmed) and **AF** (Pick up).

d. Analysis of number of events tracked per shipment

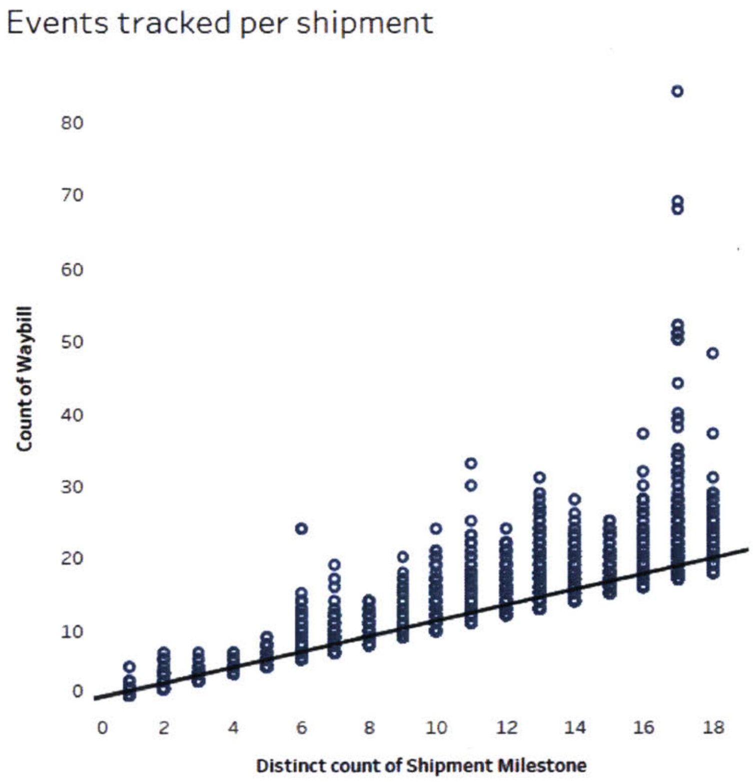


Figure 13: Plot of number of events tracked for shipment

Each point on the graph (Figure 13) corresponds to a unique shipment document number. The vertical axis represents the total number of times shipment events were updated for a particular shipment. The x-axis records the number of unique events that have been updated for shipments. For example, the point at the top right corner of the graph corresponds to a shipment for which 17 events have been recorded. These have been entered or updated a total of 84 time. Therefore, from the given dataset we can conclude that on average, 12.8 events are recorded per shipment, and these are entered or updated 14.3 times.

4.3.1.2 Descriptive evidence of hypotheses

We identified the following set of hypotheses that we evaluate to arrive at informed insights about the root cause of delays in shipment tracking. The hypotheses considered can be broadly categorized into the following sections:

1. Temporal hypotheses
    a. The frequency of system errors increases at specific periods of time.
    b. There is a correlation between number of system errors per event and the unit of time.
    c. There is a significant time delay between initial data entry and the corresponding data corrections.
    d. The time delay between initial data entry and the corresponding data corrections is correlated with the event code.
    e. There is a correlation between time of the year and number of Operational errors.
    f. There is a correlation between number of Operational errors per event and the unit of time.
    g. Edited fields have a correlation with time.
2. User- and consignee-driven hypotheses
    a. One user (or subset of users) is responsible for most of the system errors and the corrections.
    b. There is a correlation between number of errors entered and the Consignee.
3. Geo-spatial hypotheses
    a. A specific source and destination are associated with most of the System errors and the Corrections.
    b. There is a correlation between number of errors and the Origin and Destination country.
    c. Edited Fields have a correlation with Source, Destination and Event location.
4. Other hypotheses about corrections and reason codes
    a. Updates and Corrections are concentrated on a few shipments.
    b. Updates and Corrections are concentrated on a few events.
    c. A subset of reasons underlies all delays.
    d. Edited Fields have a correlation with Shipment mode, Reason code, Consignee, Event, and Status.

4.3.1.2.1 Hypotheses Results

This section lays out our findings from the hypotheses that we evaluated.

Temporal hypotheses

a. The frequency of system errors increases at specific periods of time

We analyzed the variations of system errors with time. In the graphs below, we are looking at the y-axis variable "Sum(error)/c(event)" which is the ratio between number of records that have a UI Input Error and the total number of events on a given time unit.

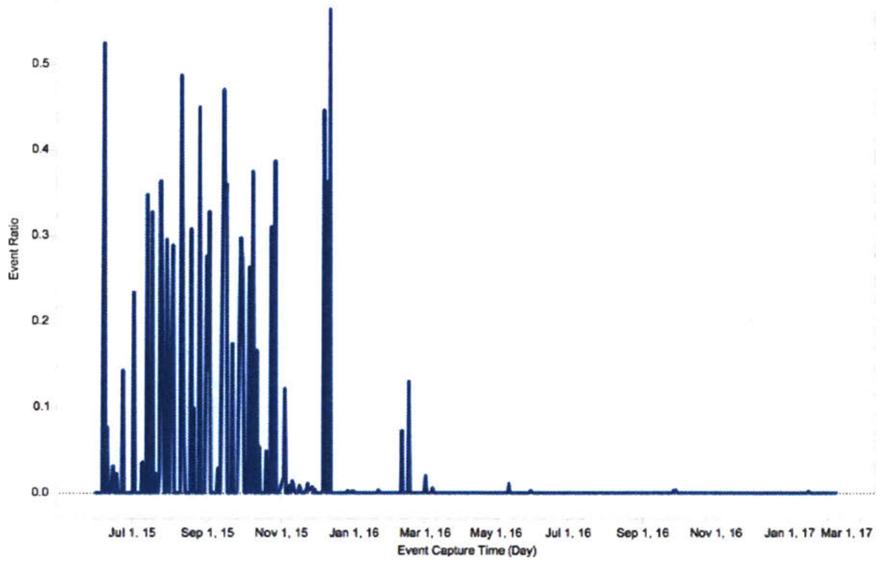


Figure 14: Relative occurrences of System errors, by day

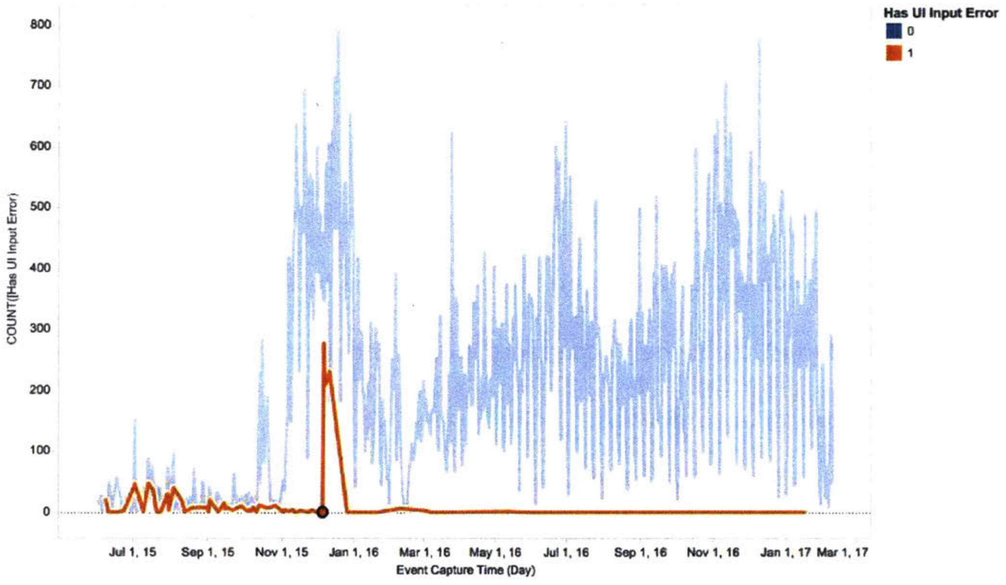


Figure 15: System errors against all entries, by day

When we look at the entire timeline, as in Figure 14 above, we notice that the ratio is highest between May and December 2015. It then goes down significantly, with the exception of February 2016, where it is much lower than most of 2015. However, the absolute occurrences, as shown in Figure 15, do not follow the same pattern.

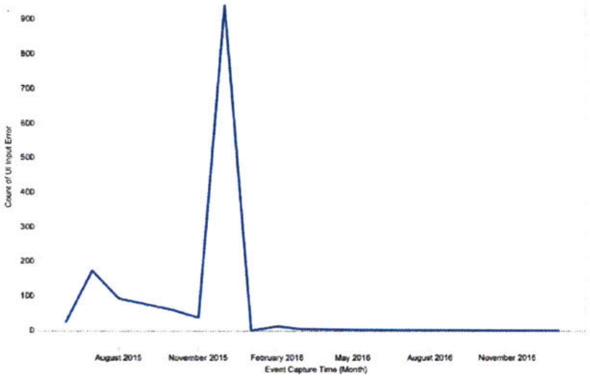


Figure 16: Absolute occurrences of System errors by month

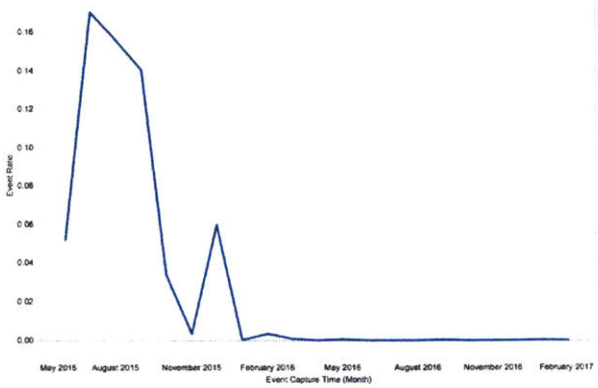


Figure 17: Relative occurrences of System errors by month

Zooming out to observe the ratio against the months of each year, as shown in Figure 16 and Figure 17, we notice that July, August, and December 2015 have the highest ratio of errors to total events, while the absolute count shows a sharp peak in December 2015.

Conclusion: When we analyze the UI system errors against time, we notice that the highest concentration of errors occurs around the first half of December 2015, followed by July and August. However, when we look at the number of system errors relative to the total number of transactions made during the same time period, the highest occurrence is observed during the month of July. Our hypotheses to explain this behavior are as follows:

(i) System errors seem to reach near-zero values post February 2017. This may be indicative of either a data collection gap or of a potential bug-fix in the system.

(ii) In absolute terms, the months with the highest number of transactions have the highest number of reported system errors, but the same does not hold when we analyze the ratio of system errors to the total number of transactions.

These two factors indicate that the system errors are potentially driven by exogenous variables which are not captured in the given dataset.

b. There is a correlation between number of system errors per event and the unit of time

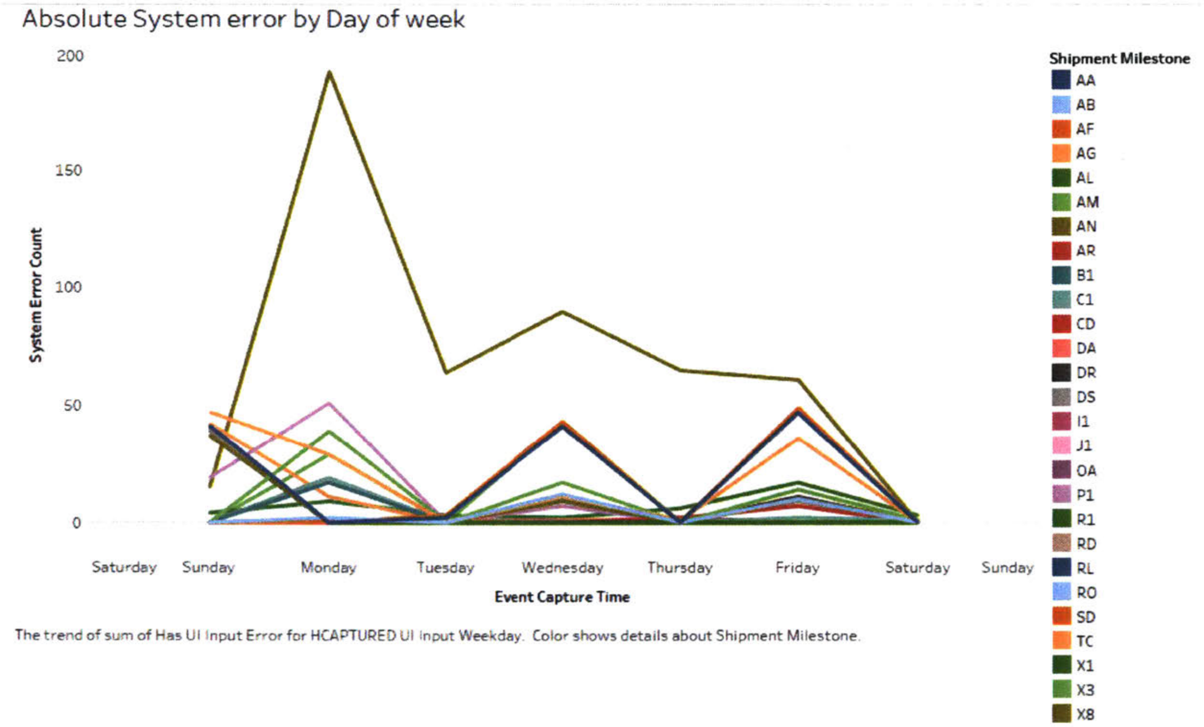


Figure 18: Absolute System Error distribution by day of week

As shown in Figure 18, a large number of the system errors are concentrated in the event X8 (Arrived at destination airport). Further, it is interesting to note that the events R1 (Cargo received from airline), AN (ETA to destination airport), TC (Transport Confirmation) and AG (FETA) follow a different trend for Monday, which is the day when the occurrence of errors is the highest.

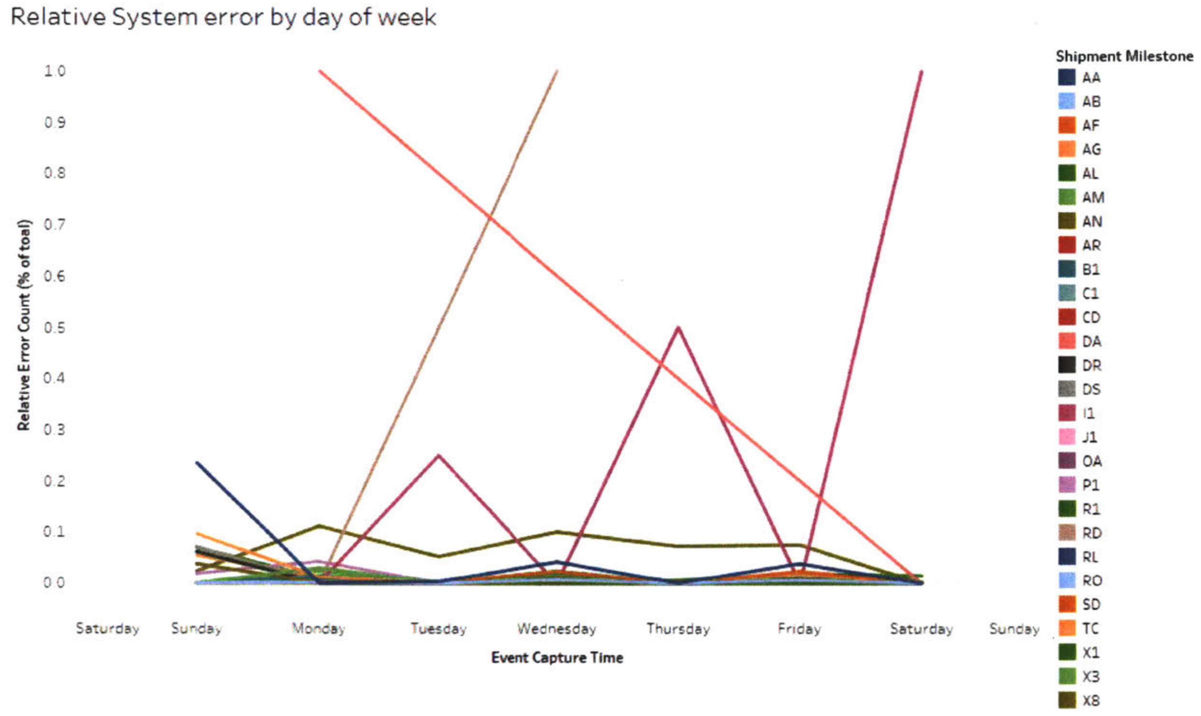


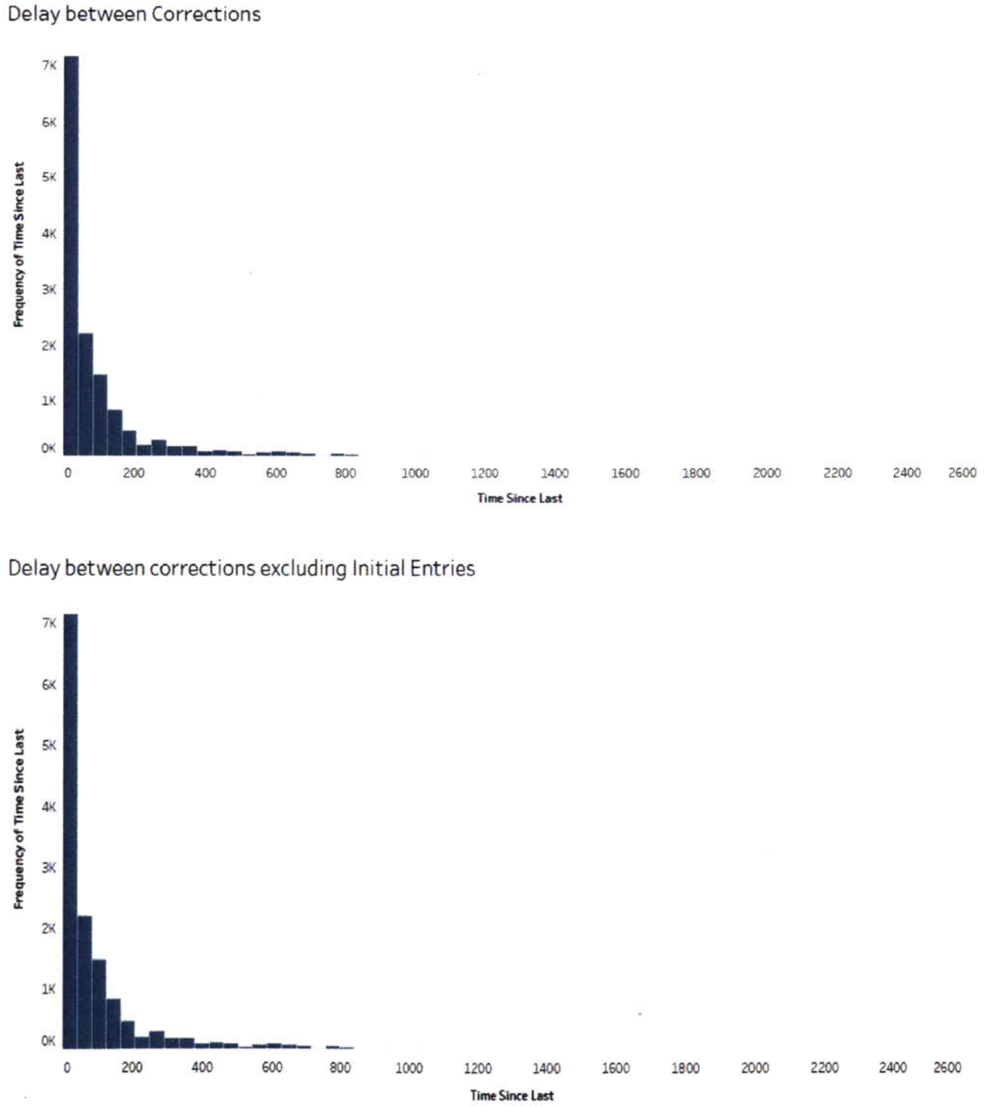
Figure 19: Relative System Error distribution by day of week

However, when we look at the ratio of the number of system errors to total number of records for the event in Figure 19, we find that the event I1 (Arrive at Intermediate Port) appears to have the highest relative count of errors. It is also interesting to note the trend for this event I1, which is the inverse of the trend for the other events. The absolute value, however, is extremely low for this event, ranging between 0 and 2 total records. This makes the event data statistically insignificant. The next most important event is once again X8 (Arrived at destination airport), although AA (Arrive Air Gateway) is significant on Sundays.

Conclusion: System errors related to event AA are more frequent on Sundays whereas X8 dominates on all remaining days.

c. **There is a significant time delay between initial data entry and the corresponding data corrections**

To assess the time taken to correct an erroneous data entry, we begin by looking at the distribution of the time taken to correct an entry as captured by the variable 'TimeSinceLast'. This is shown in Figure 20(a).


Delay between Corrections

Delay between corrections excluding Initial Entries

Figure 20: Time difference between data corrections/updates(a and b)

This data is left-skewed because all entries corresponding to the status of 'Initial Entry' will have a 'TimeSinceLast' value of 0. Figure 20 (b) shows the distribution of 'TimeSinceLast' after we filter out the 'Initial Entry' records. The median value for the TimeSinceLast is 49 hours.

We then compare this distribution with the total time taken to correct an erroneous data entry from the time the first entry for the shipment-event pair was made. This is captured by the distribution of 'TimeSinceFirst' as shown in Figure 21. The median value of TimeSinceFirst edit is 74 hours.

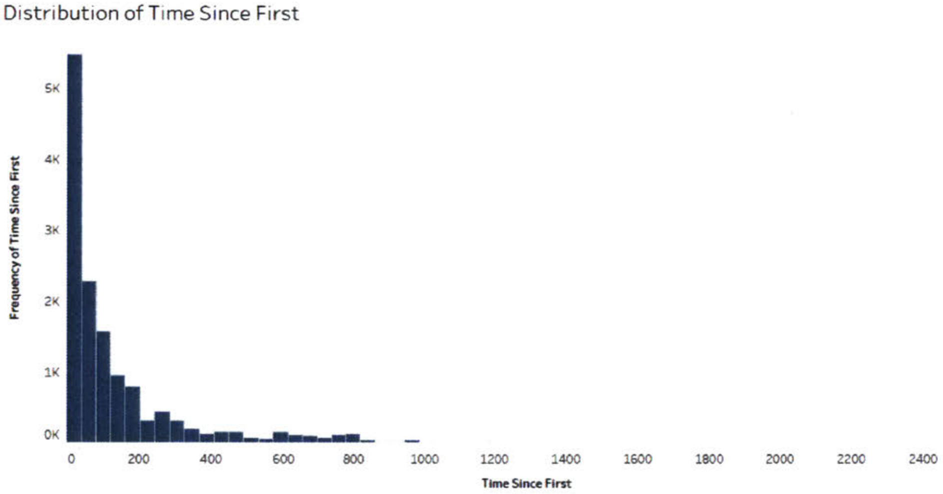


Figure 21: Histogram of 'TimeSinceFirst' distribution

Conclusion: The majority of the incorrect entries are edited within the first 40 hours and almost all of the entries are edited within 200 hours. The final correction of entry takes longer, as several of the incorrect entries undergo multiple edits. Despite this, roughly half of the errors are corrected within 40 hours of the first entry being made. However, the long-tail of the distribution is significant and poses an operational risk to Damco.

d. The time delay between initial data entry and the corresponding data corrections is correlated with the event code

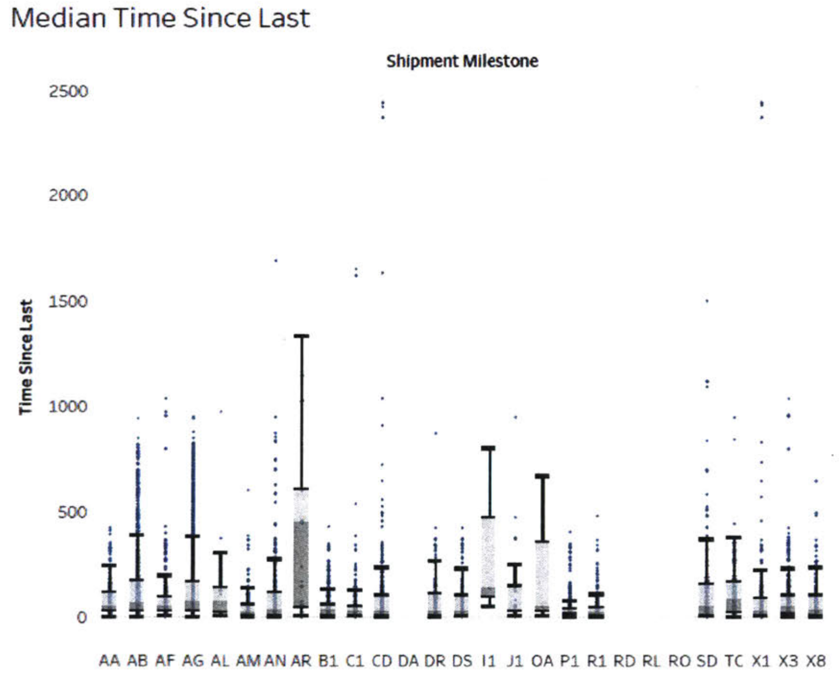


Figure 22: Median 'TimeSinceLast' by event-code

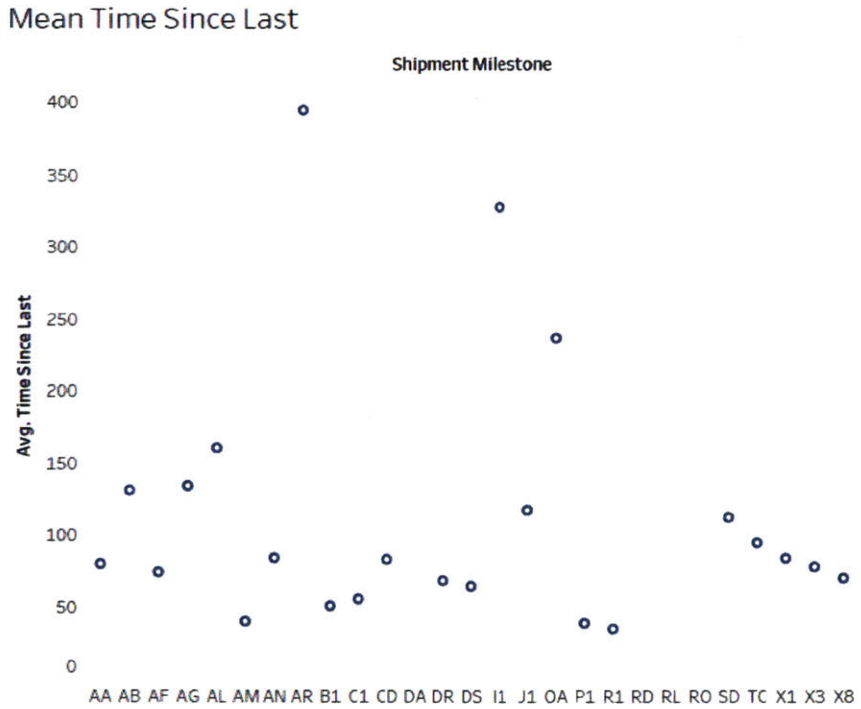


Figure 23: Mean 'TimeSinceLast' by event-code

We segment the data based on event code and observe the variation in 'TimeSinceLast' using the box and whiskers chart shown in Figure 22. The highest median value is observed for the event 'AR' (Arrived at Destination Hub) with a median value of 448 hours, and the lowest median value is observed for the event 'R1' (Cargo received from airline) with a median of 15 hours. The mean values are shown in Figure 23. As was the case for the median, the highest value of mean 'TimeSinceLast' is observed for 'AR' at 394.4 hours and the lowest value is for 'R1' at 34.4 hours.

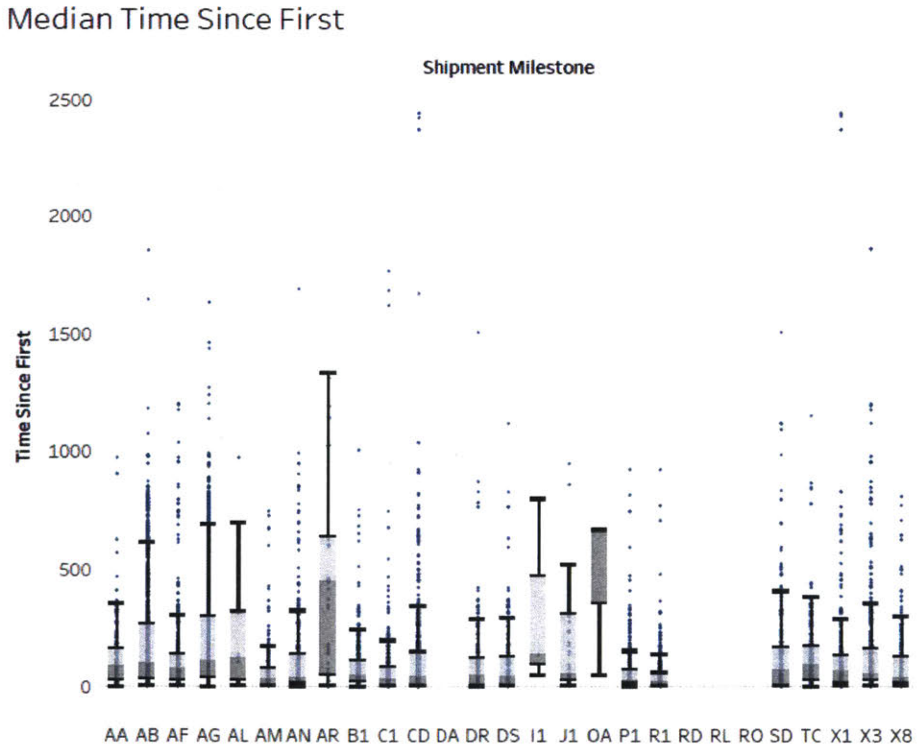


Figure 24: 'TimeSinceFirst' distribution by Event-Code

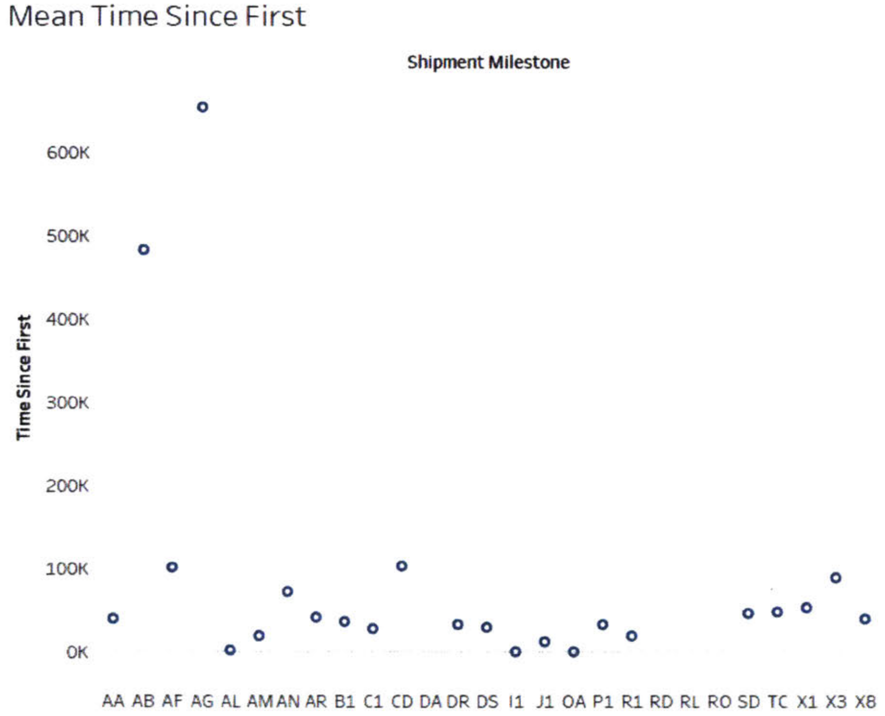


Figure 25: 'TimeSinceFirst' mean by Event-Code

The same analysis is then performed for the 'TimeSinceFirst' variable as shown in Figure 24. The highest median value is observed for the event 'AR' (Arrived at Destination Hub) with a median value of 448 hours, and the lowest median value is observed for the event 'R1' (Cargo received from airline) with a median of 8 hours. The mean values are shown in Figure 25. As in the case of the median, the highest value of mean 'TimeSinceFirst' is observed for the event 'AR' at 453.6 hours and the lowest value is for the event 'R1' at 45.7 hours.

Deep diving into the distribution of the 'TimeSinceLast' for these two events – AR and R1- by destination in Figure 26, we see that most of the R1 records are for shipments to NL, whereas the records are dispersed for AR - with a large proportion of records having destination as NL, AU, TW, KR and JP.

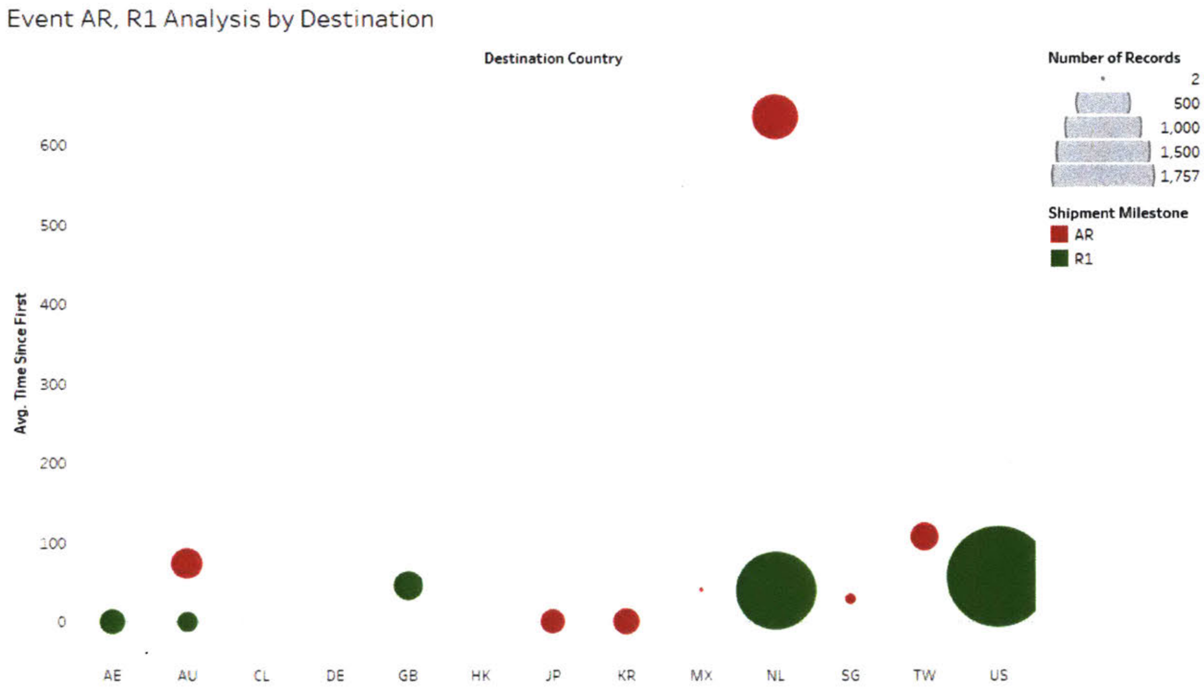


Figure 26: Distribution of Avg. Time Since last by Destination for AR and R1

e. There is a correlation between time of the year and number of Operational errors (as captured by the status Correction)

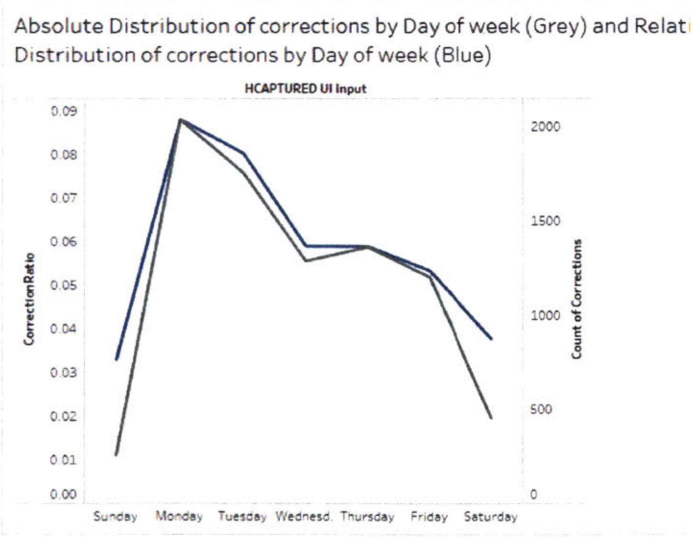


Figure 27: Absolute and relative distribution of corrections by day of week

The number of "correction" data entries appears to be significantly high on Mondays as shown in Figure 27. The correction ratio (number of corrections to total number of records) is also the highest on Monday, as shown in Figure 28.

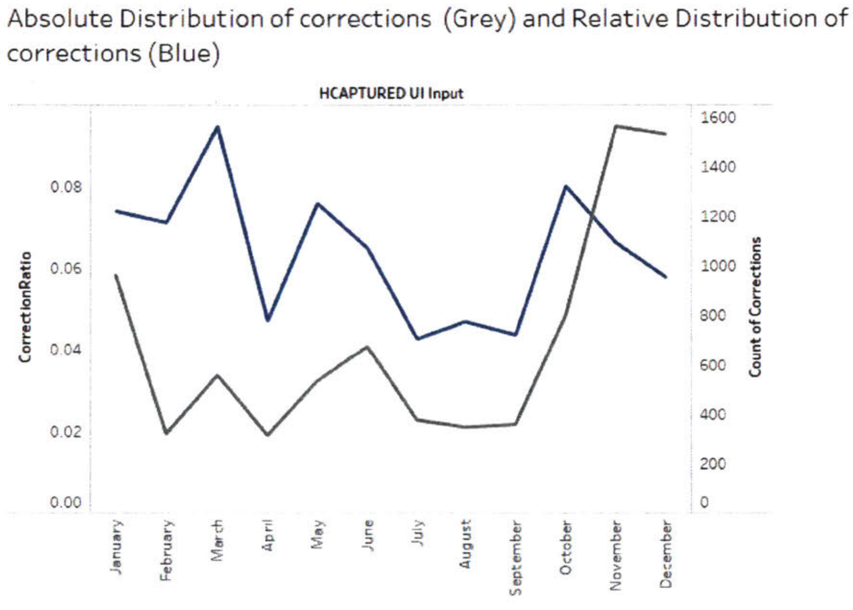


Figure 28: Absolute and relative distribution of corrections by day of month

On a monthly basis, the absolute and relative number of corrections follow different trends. We see that the highest number of corrections (relative to total number of transactions) takes place in March, followed by October. On the other hand, the absolute number of data correction is at its maximum in November, followed by December and January.

Conclusion: The data indicates the presence of correlation between the number of operational errors and the day of the week. On a monthly level, the current data suggests that the variations in number of errors is not following the same trend as the total number of transactions. This is particularly concerning for the months of March and October, and seems to indicate possible operational issues.

f. There is a correlation between number of operational errors entered per event and unit of time

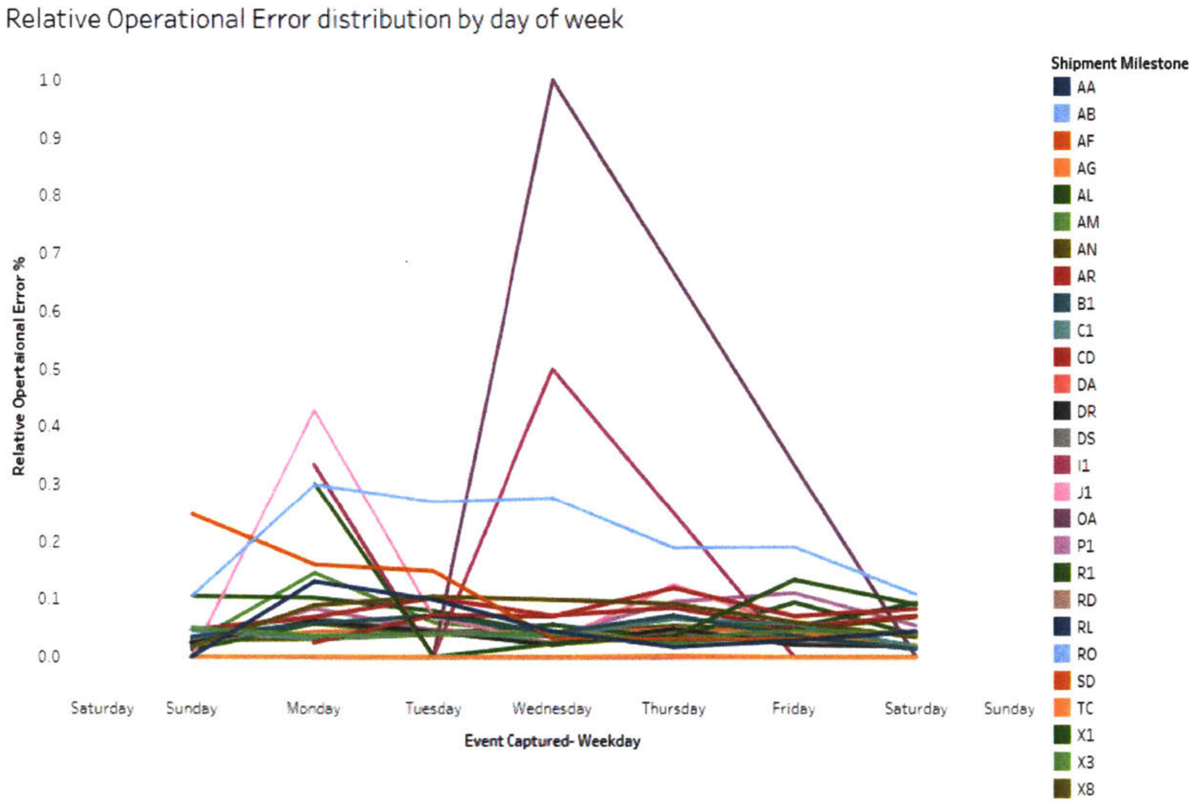


Figure 29: Relative Operational Error distribution by day of week

Operational errors on Sunday are more frequent for the event AF (Pick-up) and on Monday for J1 (Container on Board), as shown in Figure 29. On all remaining days, event AB (Delivery Appt. or Appt. Confirmed) is the dominant statistically significant event.

Conclusion: Operational errors are dominated by event AF on Sunday, J1 on Monday and AB on all other days.

g. Edited Fields correlation with time

Below, we have included three tables that show Edited Fields against both absolute number of errors and errors per number of events (relative). This view is set in different time units: quarters, months, and weekdays, respectively.

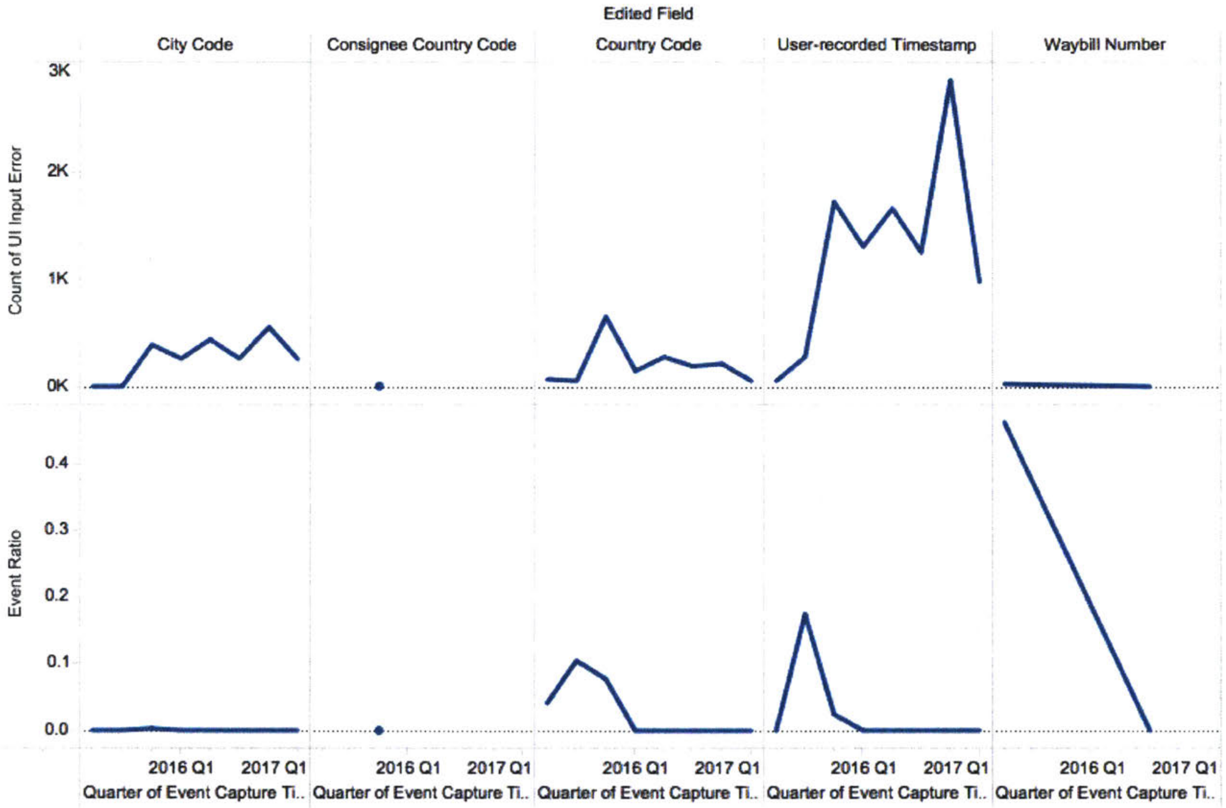


Figure 30: Occurrence of Edited Fields and both relative and absolute corrections by quarter

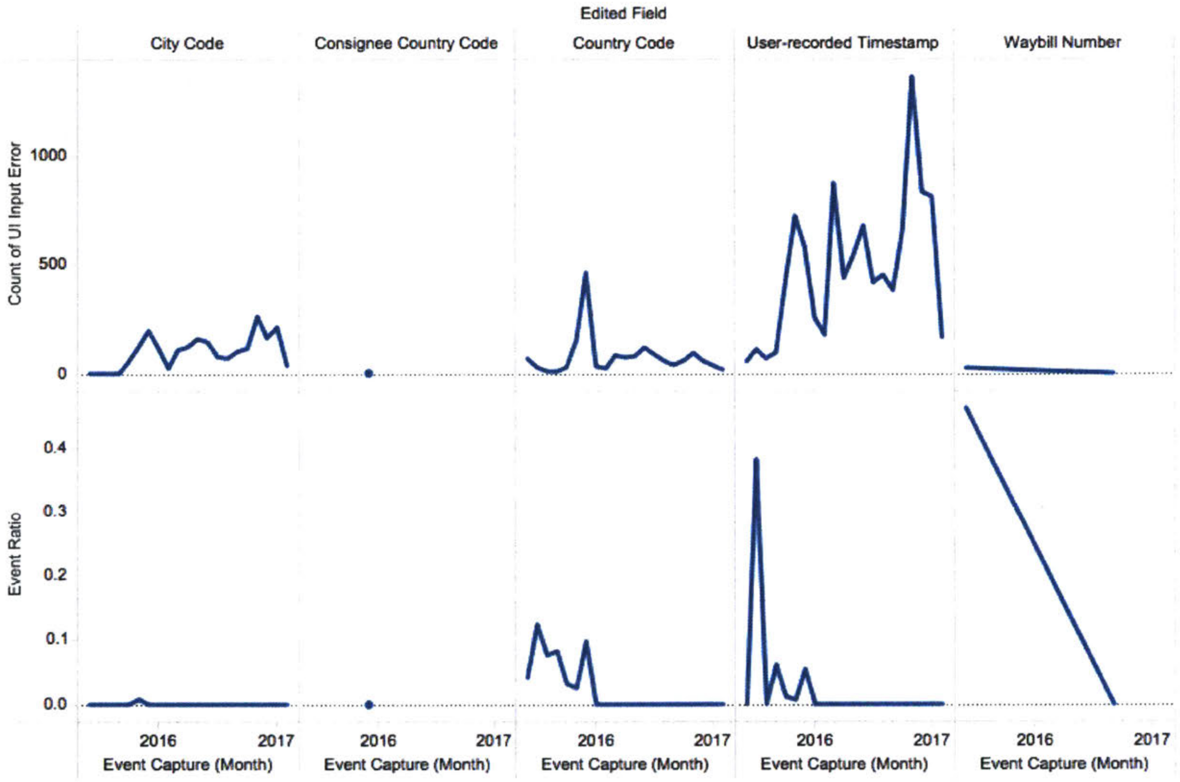


Figure 31: Occurrence of Edited Fields and both relative and absolute corrections by month

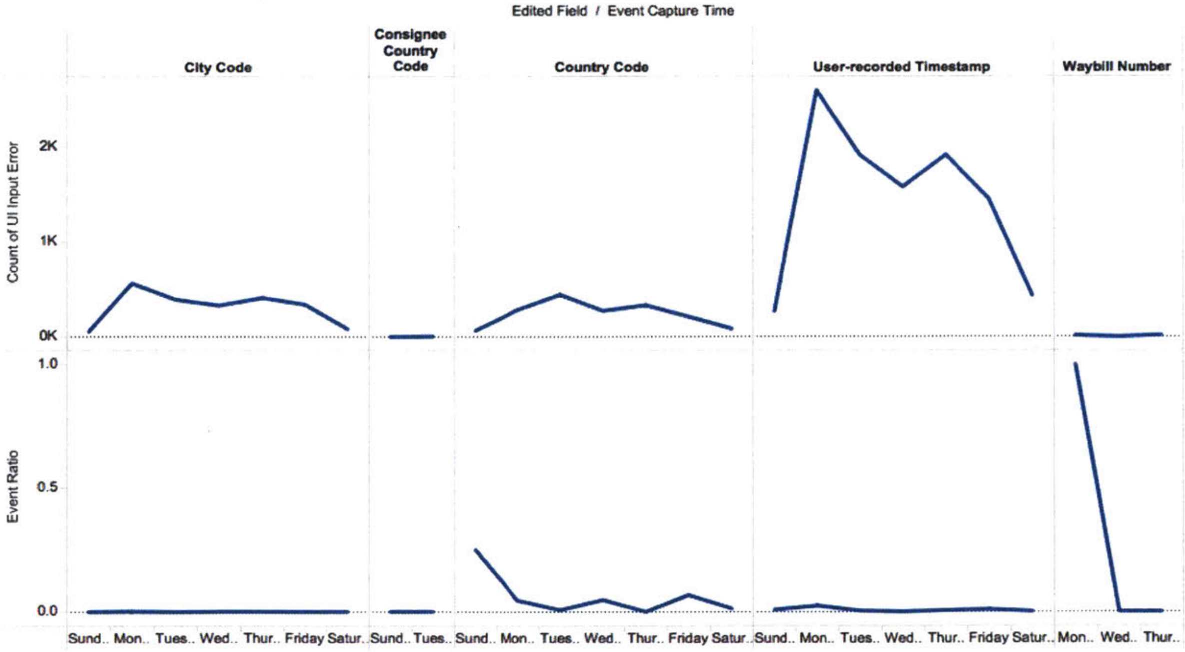


Figure 32: Occurrence of Edited Fields and both relative and absolute corrections by weekday

Conclusion: The table below summarizes the observations from the three figures (Figure 30, Figure 31, Figure 32) above. Most notably, it is a rare occurrence for an absolute value to align with a relative value. Thus, we can infer that there are explanations that account for an increase in errors other than simply an increase in volume of events.

Table 4: Temporal variation of Edited Fields

	Event Timestamp	**Consignee country**	**Waybill no**	**Event city**	**Event country**
Day of Week Figure 32	**Absolute** Highest: Monday Lowest: Sunday **Relative** Highest: Monday Lowest: Saturday	**Absolute** Highest: Tuesday Lowest: Sunday **Relative** Highest: Tuesday Lowest: Sunday	**Absolute** Highest: Thursday Lowest: Wednesday **Relative** Highest: Monday Lowest: Wednesday	**Absolute** Highest: Friday Lowest: Sunday **Relative** Highest: Monday Lowest: Sunday	**Absolute** Highest: Friday Lowest: Sunday **Relative** Highest: Sunday Lowest: Saturday
Month Figure 31	**Absolute** Highest: Nov 2016 Lowest: June 2015 **Relative** Highest: July 2015 Lowest: N/A	N/A	**Absolute** Highest: June 2015 Lowest: Sept 2016 **Relative** Highest: June 2015 Lowest: Sept 2016	**Absolute** Highest: Nov 2016 Lowest: June & Sept 2015 **Relative** Highest: Dec 2015 Lowest: N/A	**Absolute** Highest: Dec 2015 Lowest: Aug 2015 **Relative** Highest: July 2015 Lowest: N/A
Quarter Figure 30	**Absolute** Highest: Q4 2016 Lowest: Q2 2015 **Relative** Highest: Q3 2015 Lowest: N/A	N/A	**Absolute** Highest: Q2 2015 Lowest: Q3 2016 **Relative** Highest: Q2 2015 Lowest: Q3 2016	**Absolute** Highest: Q4 2016 Lowest: Q2 & Q3 2015 **Relative** Highest: Q4 2015 Lowest: N/A	**Absolute** Highest: Q4 2015 Lowest: Q1 2017 **Relative** Highest: Q3 2015 Lowest: N/A

User and consignee driven hypotheses

a. One user (or subset of users) is responsible for most of the system errors and the corrections.

System errors: As described earlier, these are the errors that are caught by the system.

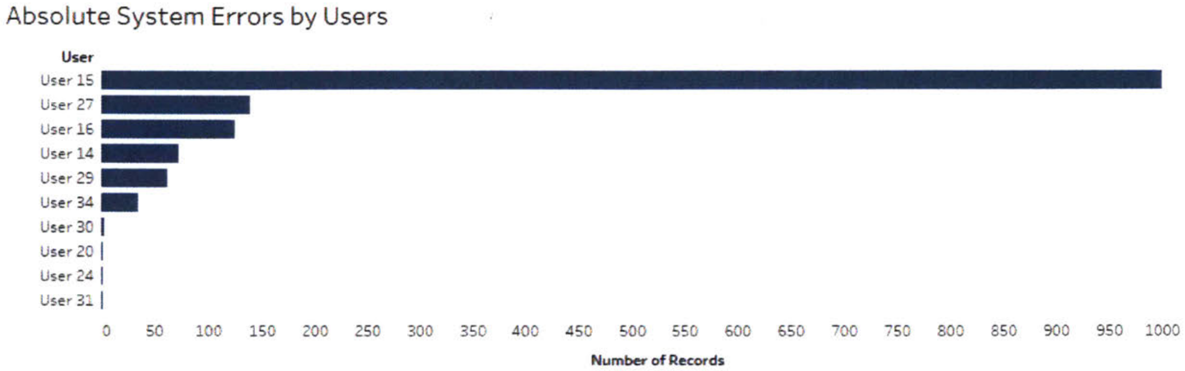


Figure 33: Absolute System Errors by Users

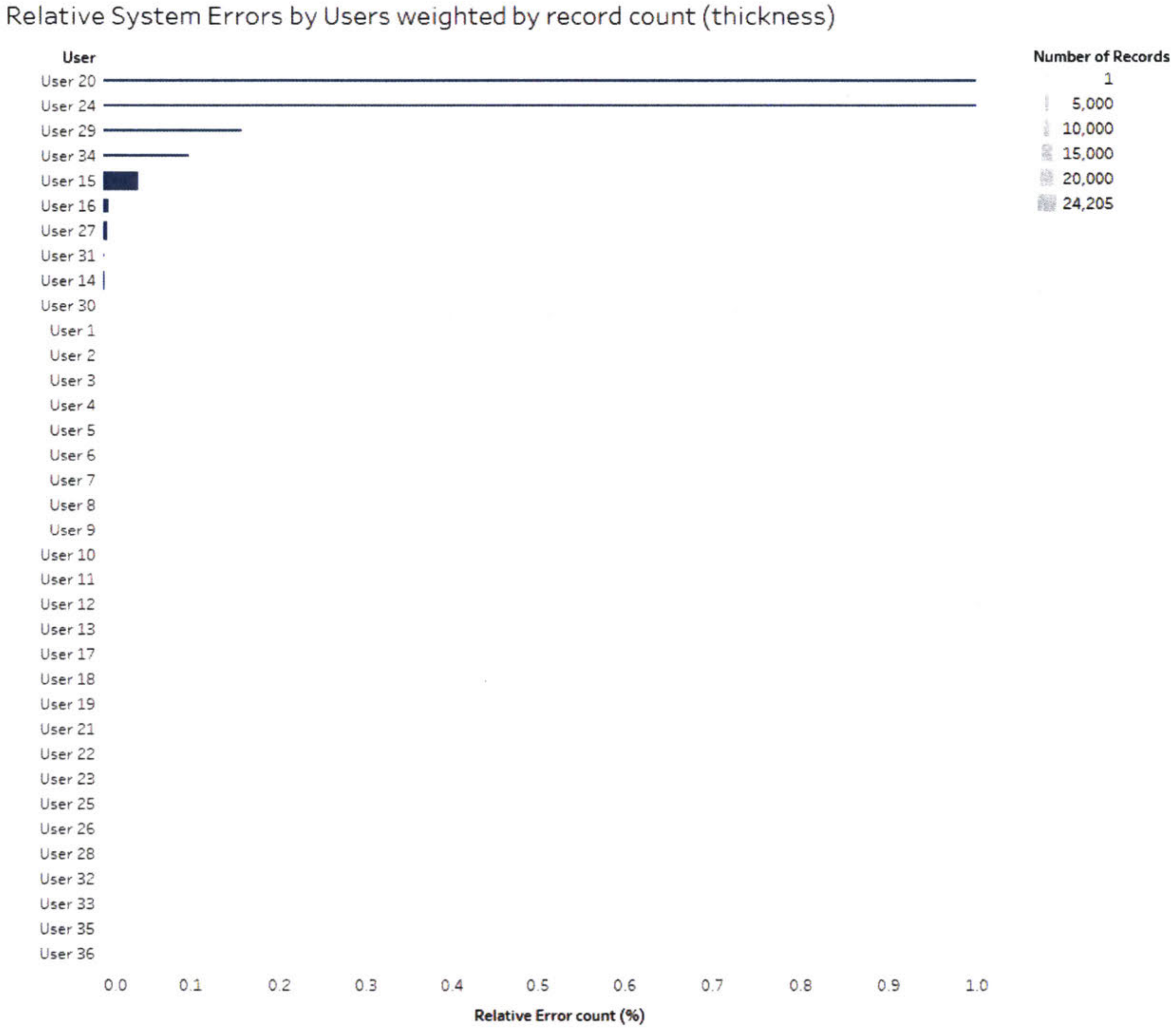


Figure 34: Relative System Errors by Users

In absolute terms (Figure 33), most of the system errors are attributable to the user with User-ID "User 15". When we look at the data relative to the total number of records entered by the user

Figure 34, we get a different insight. The thickness of the bars represents the number of records entered by the user, and the length represents the ratio of erroneous entries to total number of entries. The entries made by the users "User 29" and "User 34" consist of a significant number of system entries. The reason we are not as concerned about "User 20" and "User 24" despite the high ratio of system errors/total entries is because the number of entries made by this user is extremely small which makes the ratio look significantly worse than the problem.

Conclusion: The users whose entries warrant deeper analysis are "User 29" and "User 34" because both the number of data entries made by them and the percentage of entries made by them which require corrections are significant.

Operational errors: These are the errors which result in records with the status correction.

Here we analyze the volume of transactions made per user which are not Final entries. This is suggested by a value of 0 for 'IsFinalEntry' and the results are shown Figure 35:

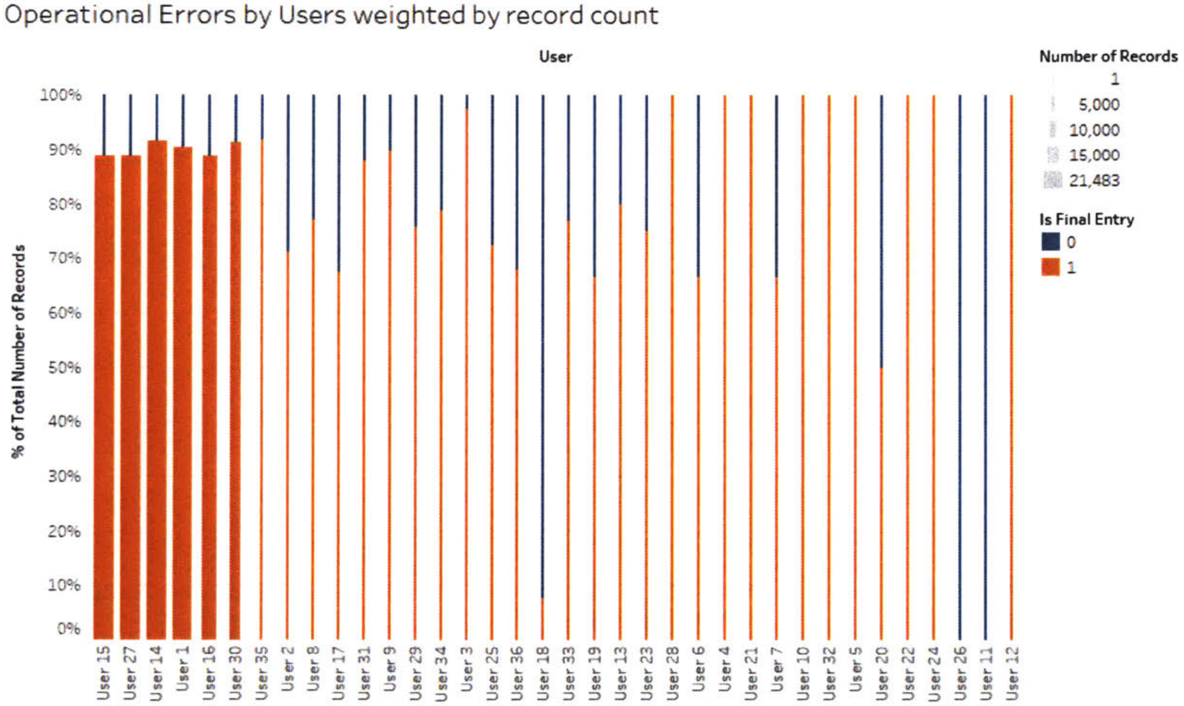


Figure 35: Operational Errors by Users

The color represents the proportion of entries made by the user (orange: final, blue: non final). The thickness of the bars represents the number of records in the category. The ideal situation is thick-tall orange bars and the least desirable is thick-tall blue bars.

Conclusion: User "User 18" makes a significantly high number of these non-final entries as compared to the number of final entries. Users "User 14" and "User 15" make a significant number of final entries.

b. There is a correlation between number of errors entered and consignee

We observe a long tail in the number of consignees as shown in Figure 36. We look at the event ratio for each of the consignees and analyze the corresponding variation in Status. The event ratio is defined as the number of records for the consignee divided by the number of unique events spanning those records. The bars represent the number of distinct shipments covered and the dots represent the event ratio.

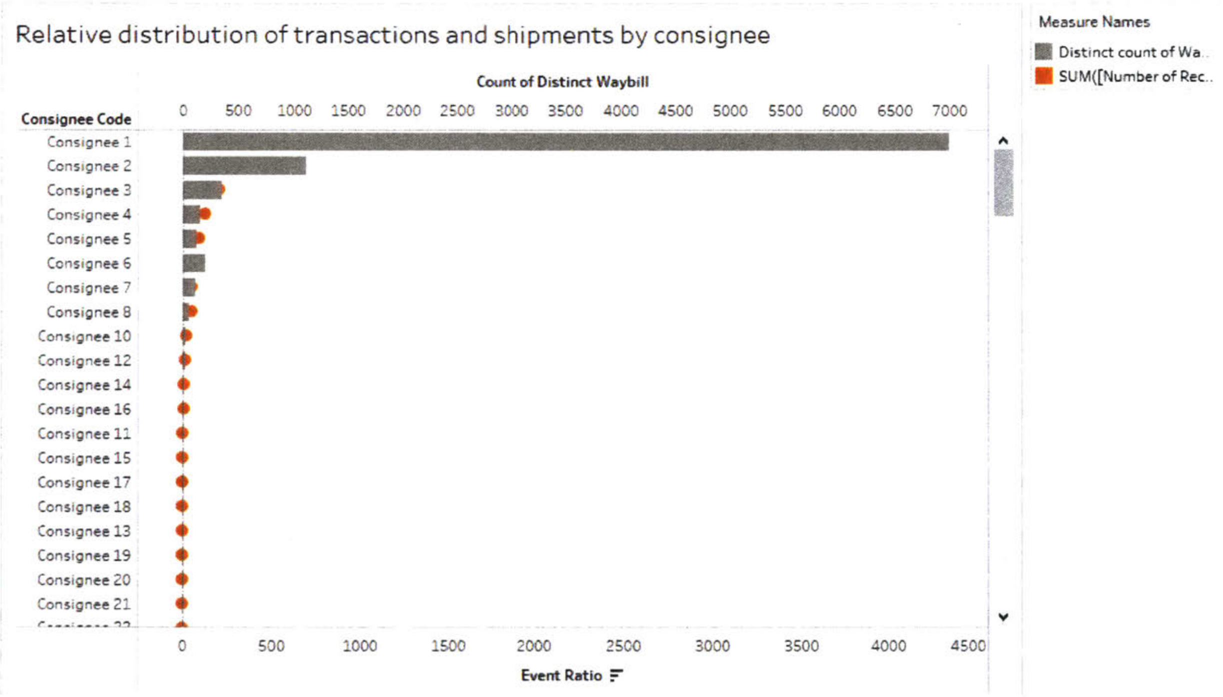


Figure 36: Relative distribution of transactions and shipments by consignee

A high event ratio implies that a large number of entries are made against the same events. However, a large portion of these are likely to be "Initial entries", which does not flag a problem. Therefore, we need to segment the event ratio by the status to make a meaningful inference.

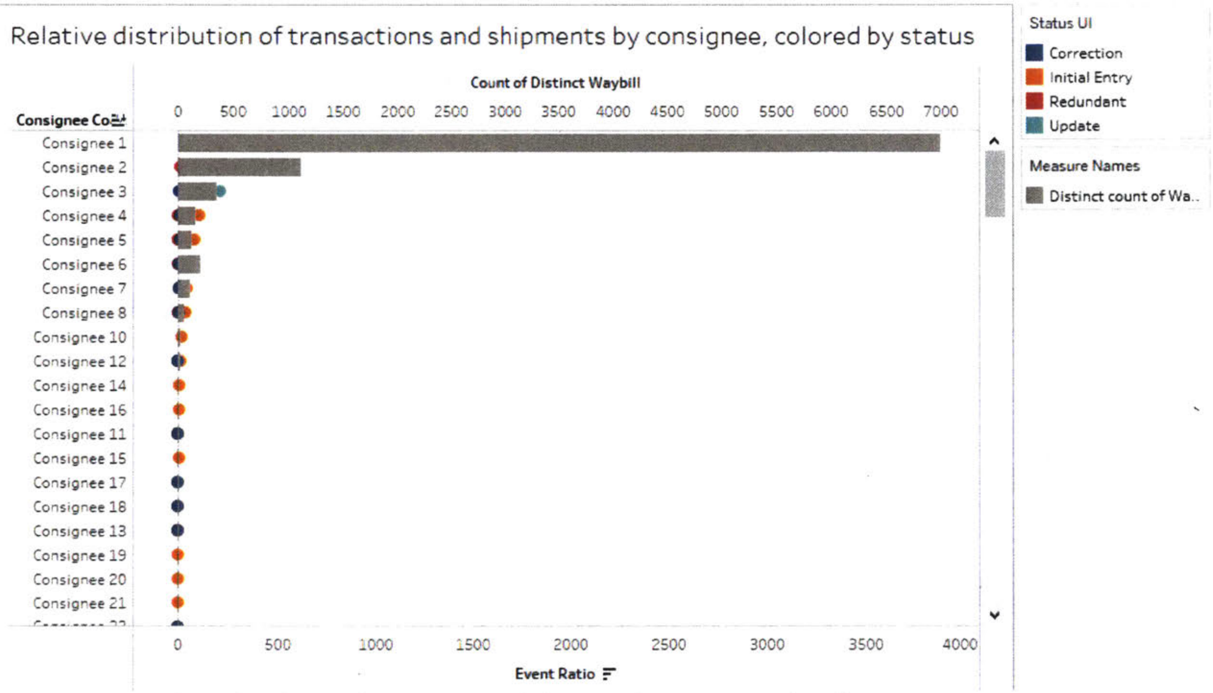


Figure 37: Relative distribution of transactions and shipments by consignee, colored by status

For 'Consignee 1' and 'Consignee 2' we see a high event ratio for the correction values (Figure 37).

Conclusion: Initial analysis suggests that the shipments which have Consignee values of 'Consignee 1' and 'Consignee 2' have higher number of corrections. This conclusion however does require further validation because we know that the total number of events is a short and finite list. Therefore, beyond a point if the number of transactions attributable to a particular consignee increases, the event ratio will inevitably increase.

Geo-spatial hypotheses

a. **A specific source and destination are associated with most of the system errors and the corrections**

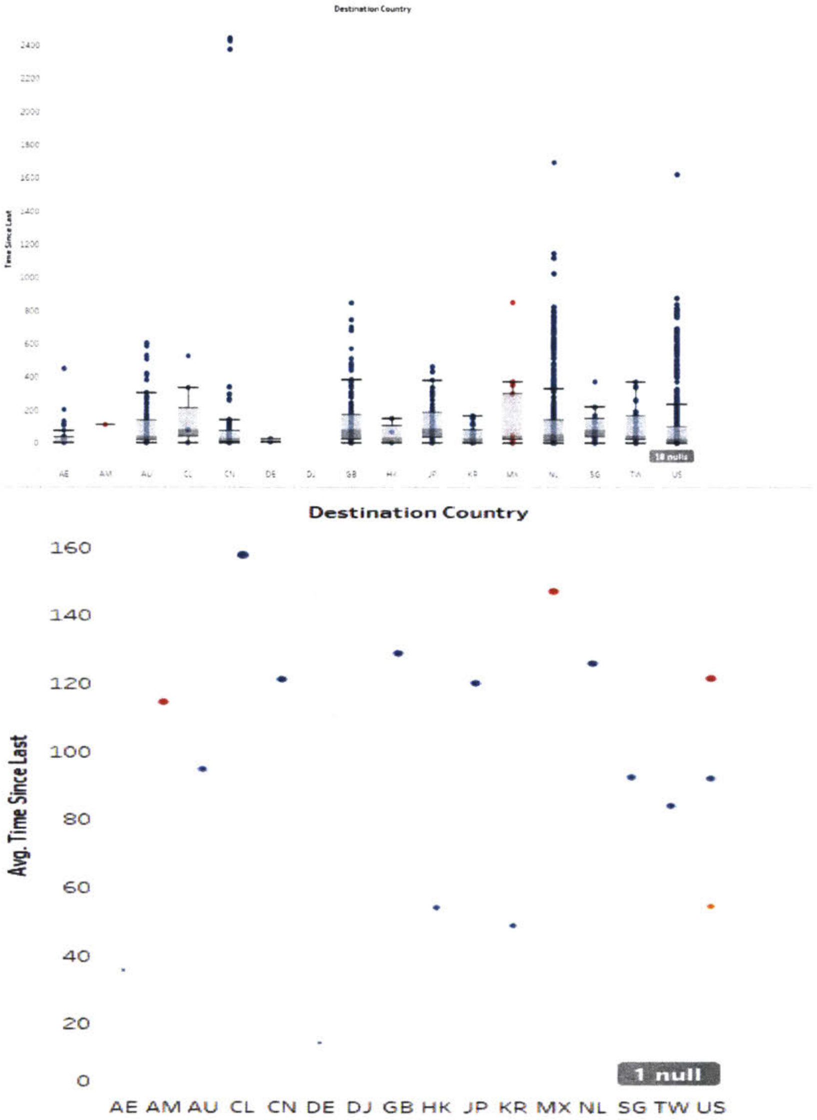


Figure 38: 'TimeSinceLast' by destination country (a and b)

When we analyze the 'TimeSinceLast' by country we find that the highest median number of hours since last edit is observed for SG at 83 hours and the lowest is for AE at 5 hours. The mean follows a similar trend as shown in Figure 38 (b).

The behavior for the 'TimeSinceFirst' is a little different. The country with the highest median number of hours since first entry is TW and the lowest is AE as before. This is shown in Figure 39:

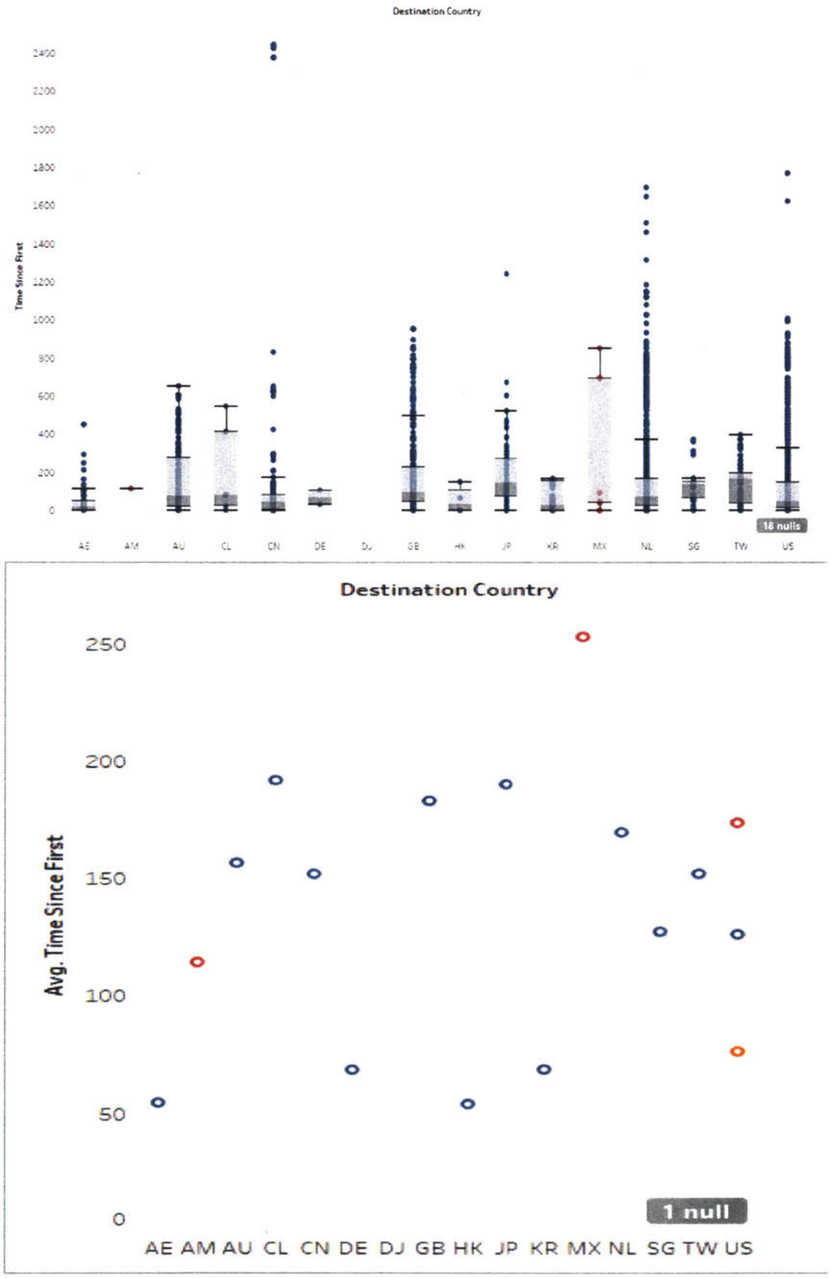


Figure 39: 'TimeSinceFirst' by destination country (a and b)

Conclusion: The total time taken for fixing errors is the highest for TW but the time difference between subsequent edits is maximum for SG. This suggests that it takes more number of entries to correct an error for TW than it does for SG (and the other countries).

b. **There is a correlation between number of errors and the origin and destination country**

No significant inferences can be made based on the origin country, since most of the data is for shipment with China as the source country. For the destination country, we notice that the absolute number (height of the column) and relative number (as denoted by the thickness of the column) of updates and corrections for US and Netherlands is high (Figure 40).

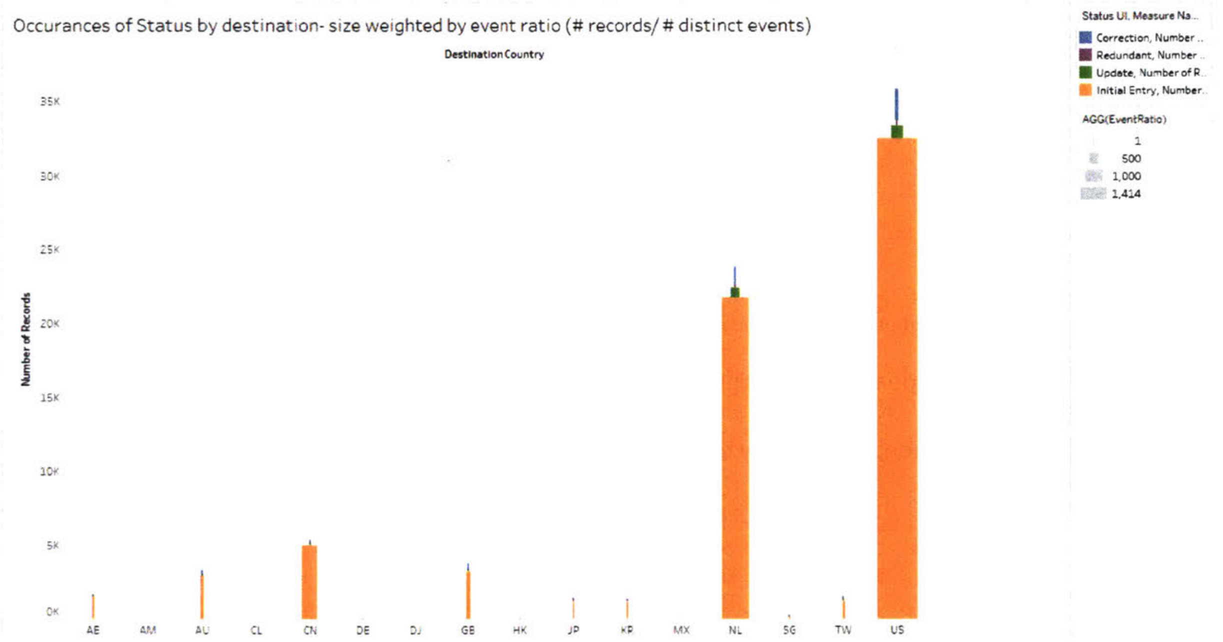


Figure 40: Relative distribution of records by event ratio and destination country

If we overlay the number of unique shipments (captured by the waybills), we see that it closely mirrors the same trend as the number of transactions (Figure 41).

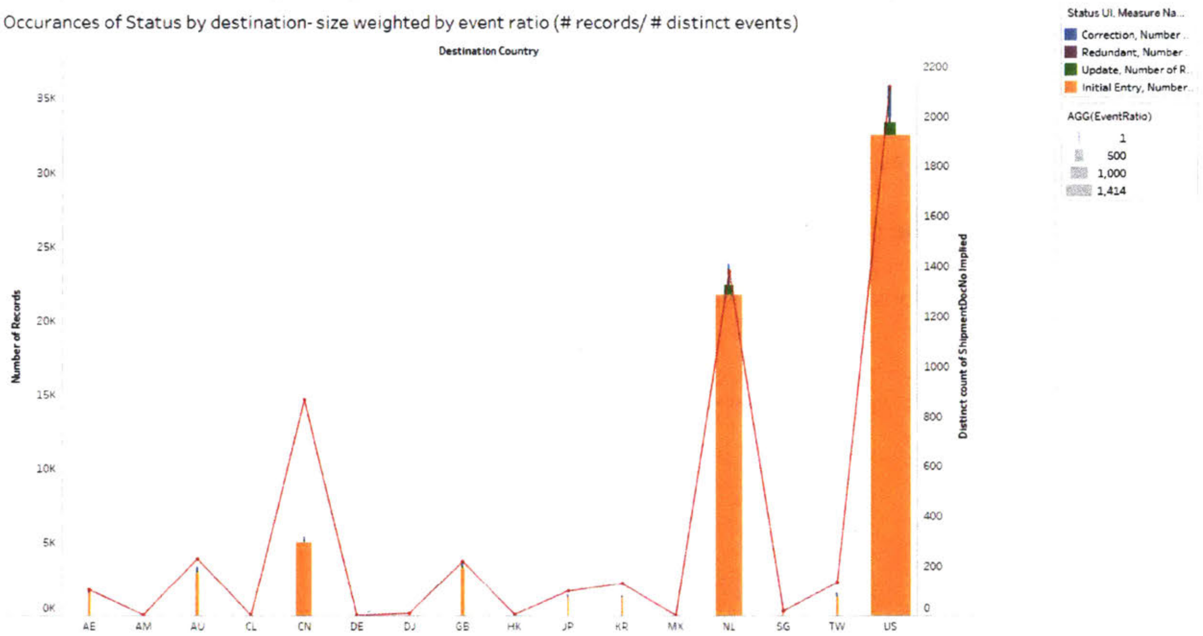


Figure 41: Relative distribution of records and shipments by event ratio and destination country

Conclusion: There appears to be a significant degree of correlation between destination country and status of the transaction, particularly the instances of updates and corrections.

c. Edited Fields have a correlation with source, destination and event location

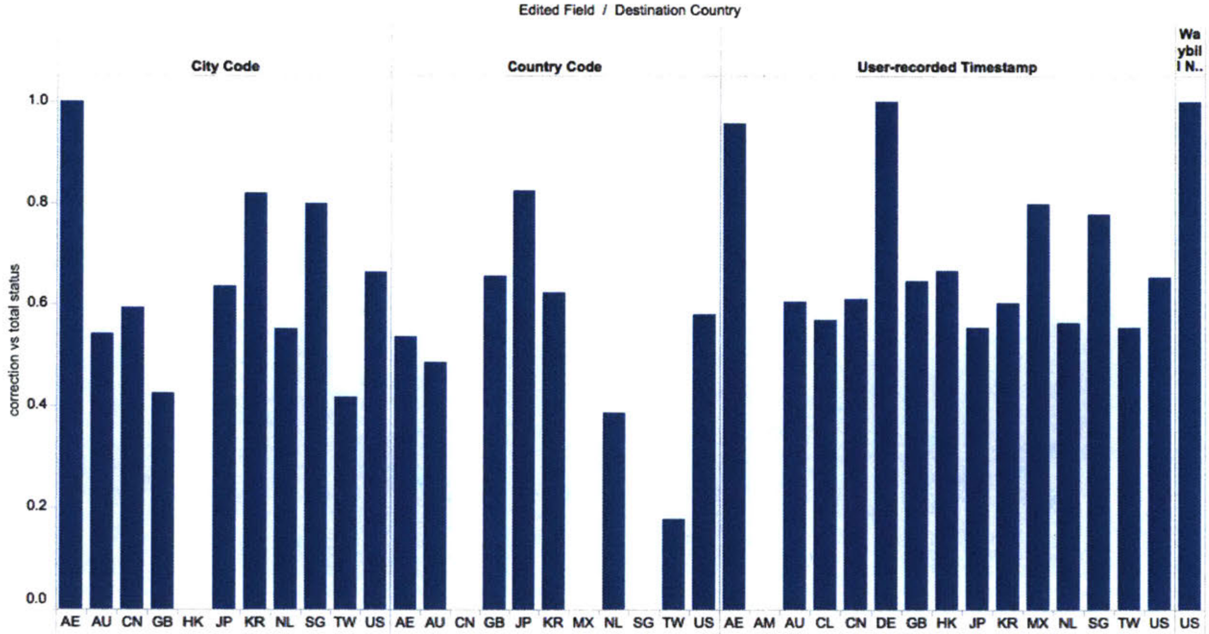


Figure 42: Correction versus total status for each Edited Field, broken down by Destination Country

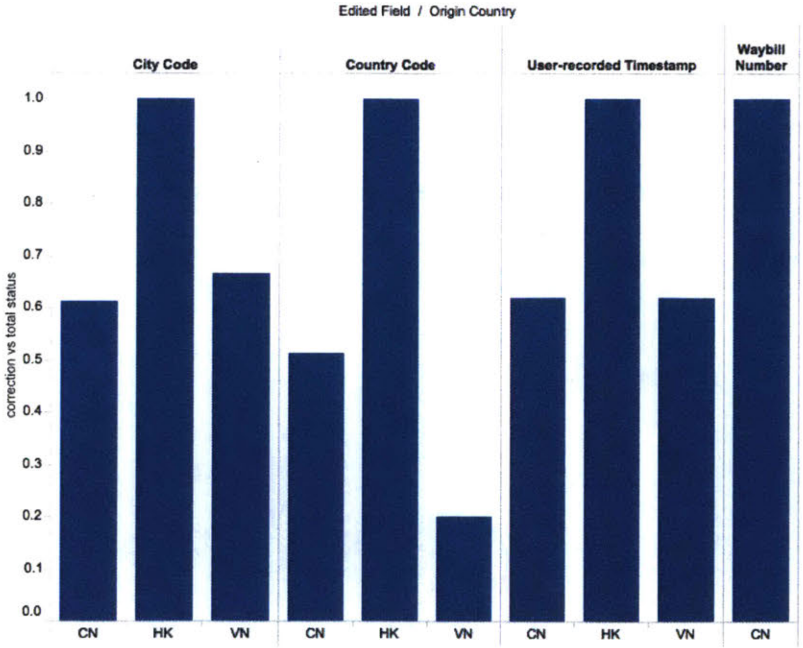


Figure 43: Correction versus total status for each Edited Field, broken down by Origin Country

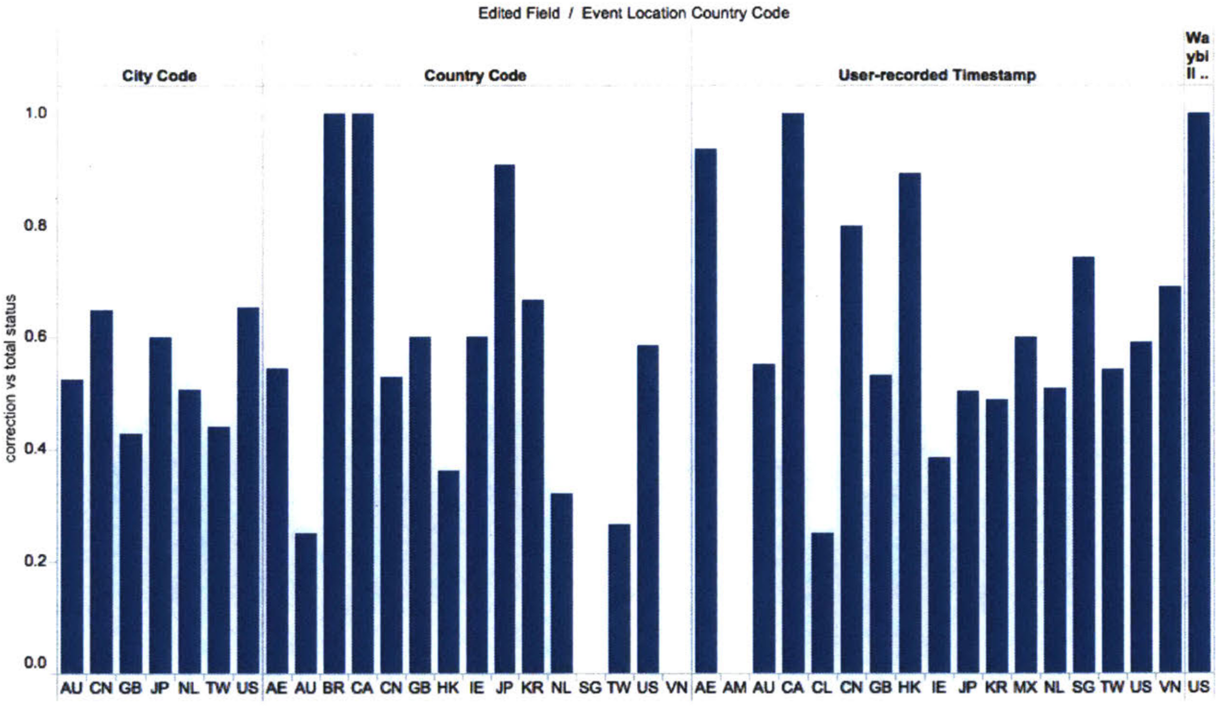


Figure 44: Correction versus total status for each Edited Field, broken down by Event Country

Conclusion: From Figure 42, Figure 43 and Figure 44, we see that for the destination country, corrections account for more than 50% of the total status edits. The US has the highest number of corrections relative to the total edits, followed by United Arab Emirates. In terms of origin countries, Hong Kong accounts for most of the corrections. Finally, regarding event countries, several countries account for a significant portion of the errors, with the US leading.

Other hypotheses about corrections and reason codes

a. Updates and corrections are concentrated on a few shipments

We analyze the relative occurrences of transactions per shipment (waybill). To compute this, we divide the number of transactions for a given waybill by the number of unique events that the transactions span. The color in the following graphs (Figure 45) represents this weighted number of transactions, and the size of the circles represents the total number of transactions covered. Therefore, the points of interest are the dark blue dots. The small dark blue dots are especially important since these represents a relatively large number of edits for a few events.

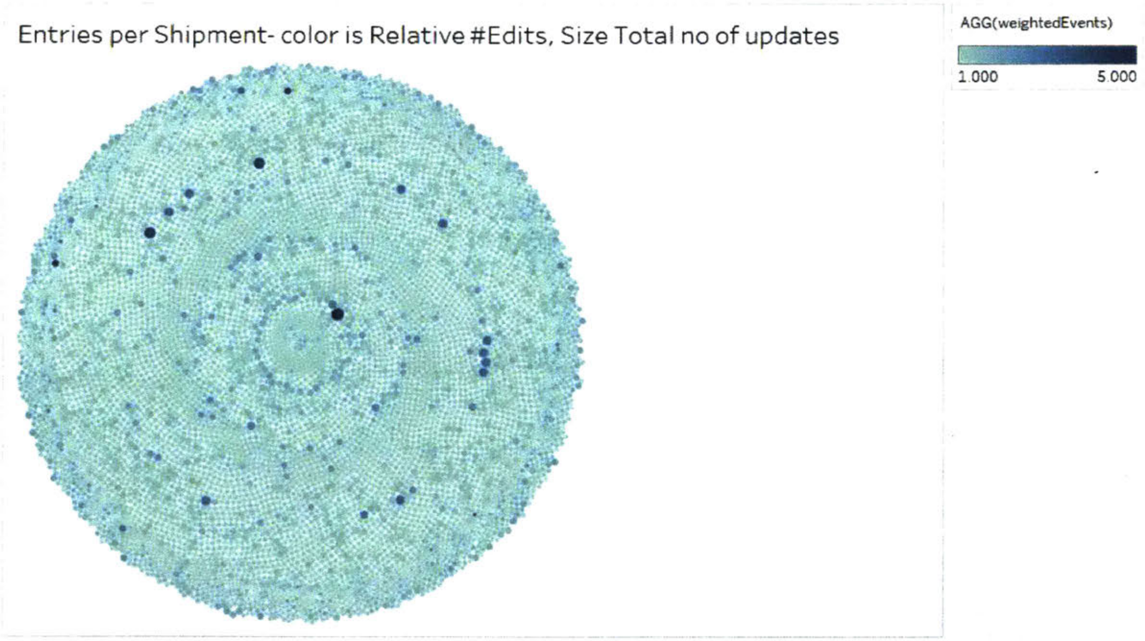


Figure 45: Relative distribution of transactions by event count

Since the data has a large number of initial entries (which is not relevant for our analysis in this case), we filter out the data by the individual status.

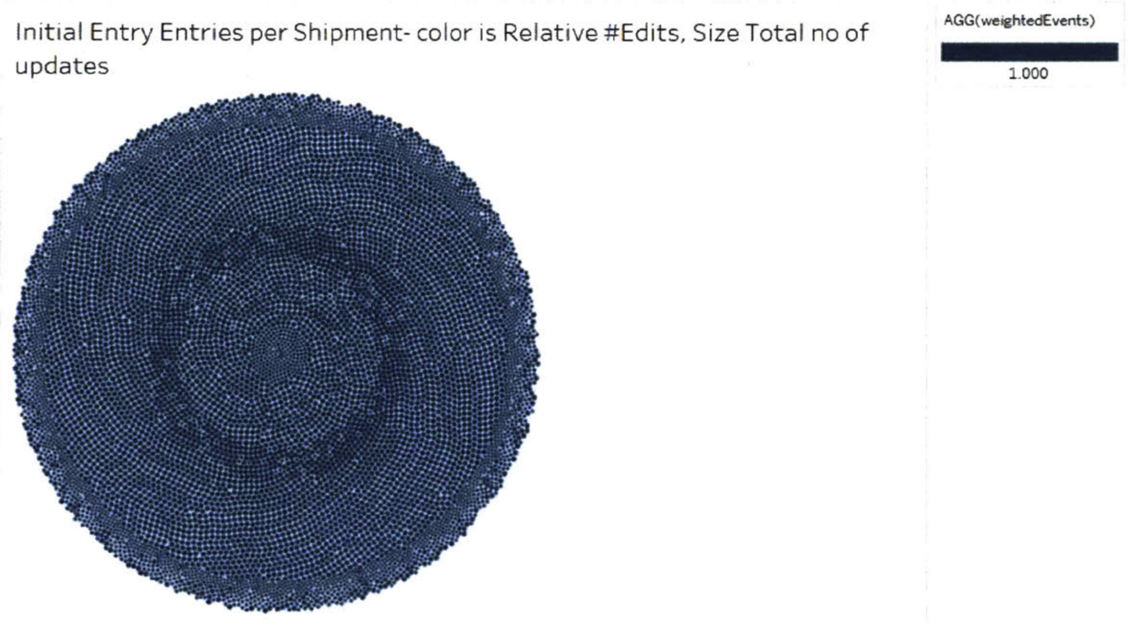


Figure 46: Relative distribution of transactions by event count Initial Entries

For the "Initial Entry" data (Figure 46), we expect the weighted occurrences to be 1 as shown above. This is because for each unique shipment- milestone that exists in the data, exactly one entry corresponding to the status "Initial entry" exists.

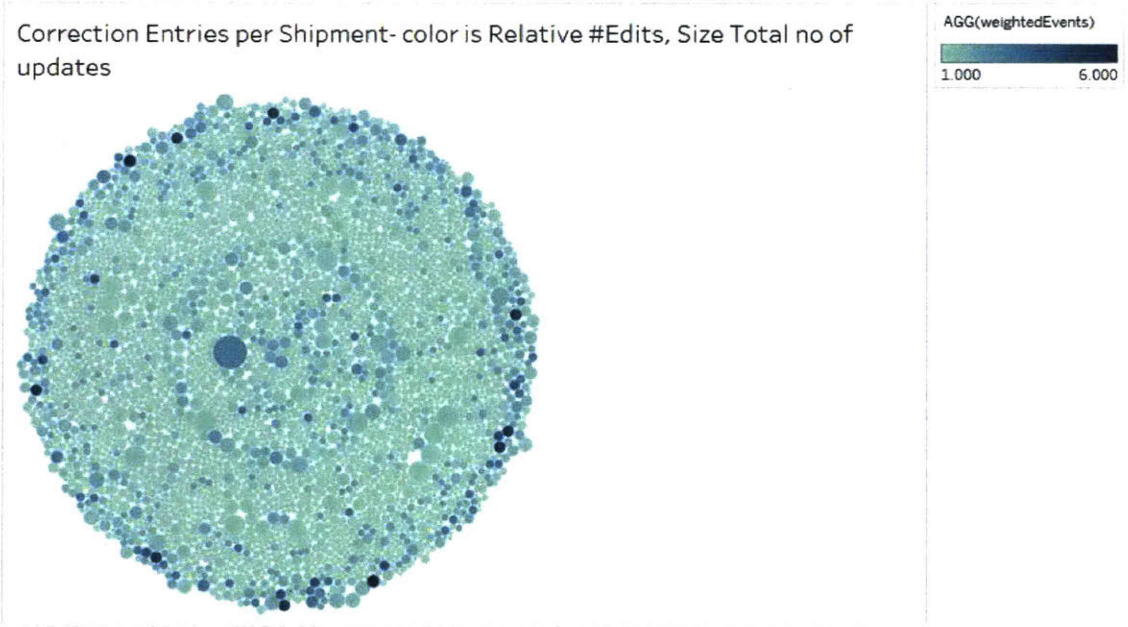


Figure 47: Relative distribution of transactions by event count Correction

For the "Correction" entries (Figure 47), we observe several dark spots. These capture shipments where the same event is corrected five to six times.

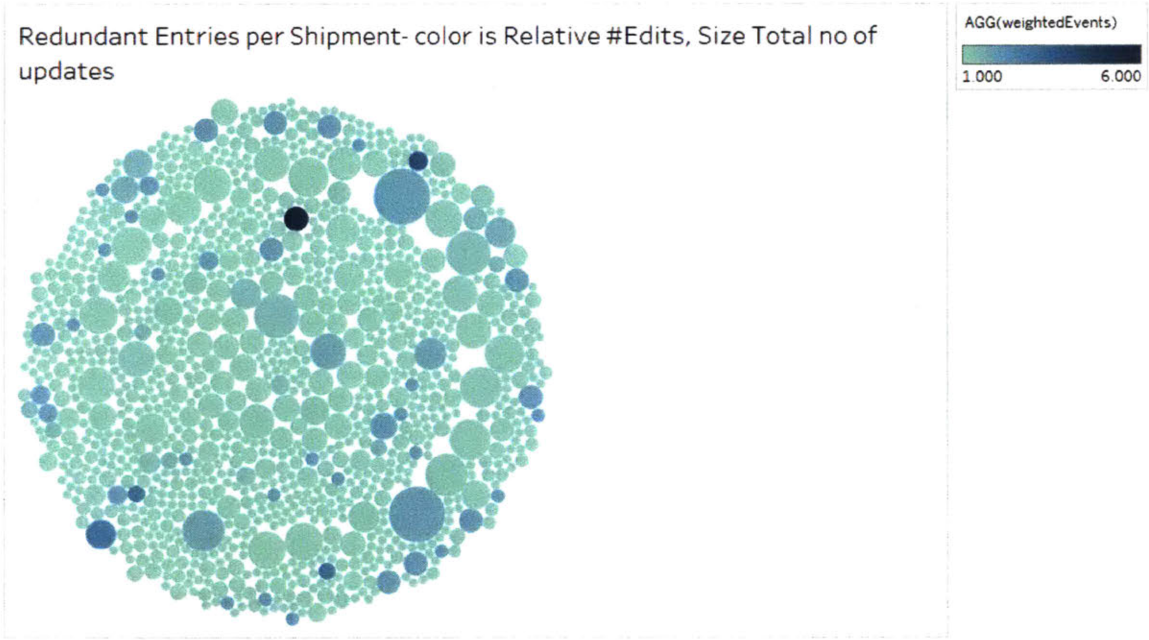


Figure 48: Relative distribution of transactions by event count - Redundant

When we look at the redundant data (Figure 48), we observe just a few dark blue points with the same event being updated for the same shipment up to five or six times.

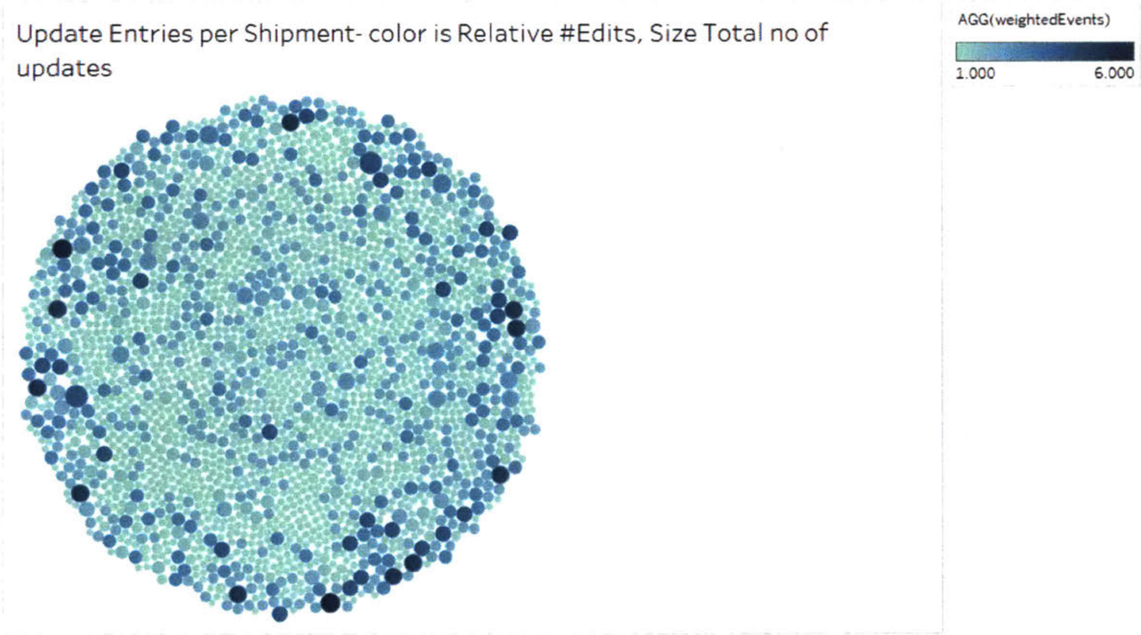


Figure 49: Relative distribution of transactions by event count Update

When we look at the update data (Figure 49), we once again see several dark blue points with the same event being updated for the same shipment up to five or six times. Updates, however, are not as concerning, as they are made for fields which we would expect to demonstrate multiple updates (such as Forecasted ETA).

Conclusion: For the 'Correction' and 'Redundant' entries, we see a strong tendency of clustering of number of corrections around a few shipments.

b. Updates and corrections are concentrated on a few events

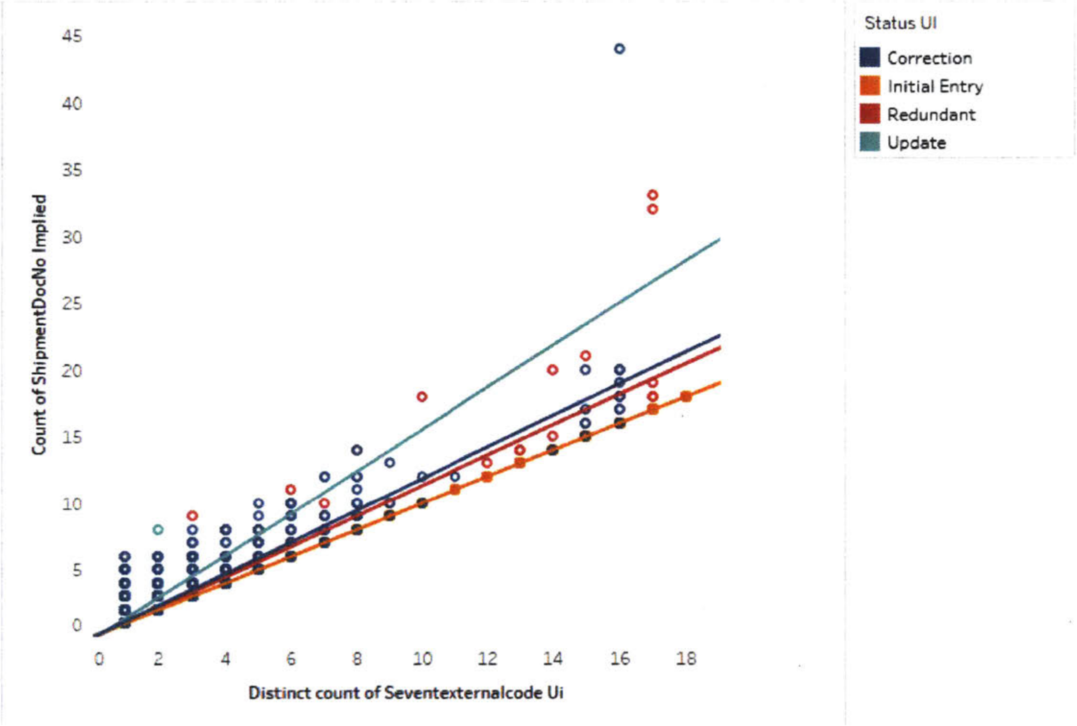


Figure 50: Absolute distribution of events per shipment colored by status

When we analyze the number of events tracked per shipment and their variations by Status, we see that a limited number of events are corrected for a larger number of shipments, as given by the higher slope in Figure 50.

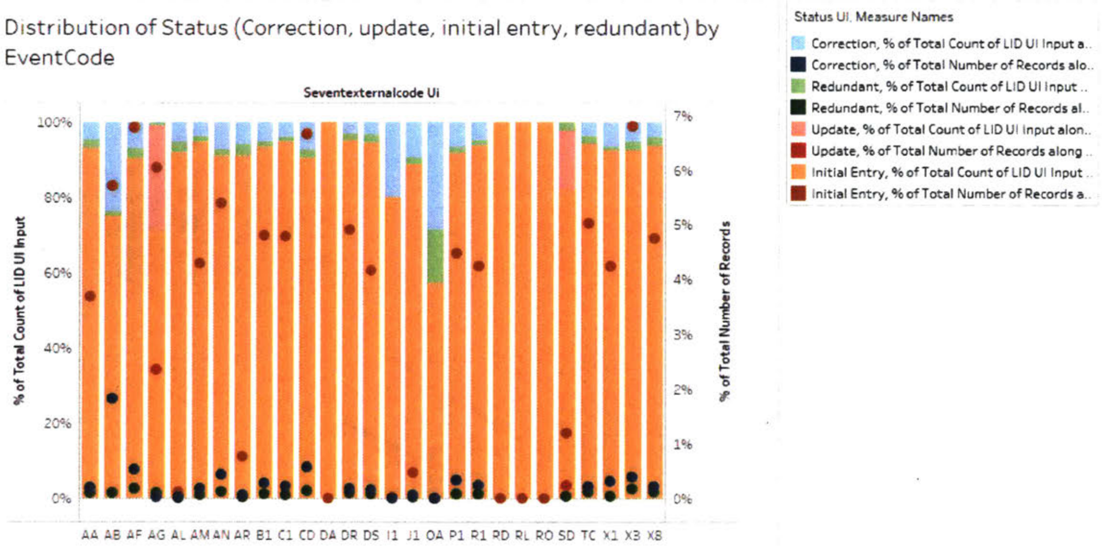


Figure 51: Relative distribution of Shipment by status and event-code

Looking at the distribution of transactions across events and the relative occurrences of corrections, updates and redundant entries across them (Figure 51), we see that 1.82% of all corrections are for the event AB (Delivery Appt. or Appt. Confirmed), and these corrections constitute 23.8% of all AB entries made. Also notable are corrections for events AF (Depart Origin) (0.53% all entire data set and 7.13% of all AF entries) and CD (Delivered/IOD per Terms) (0.57% all entire data set and 7.69% of all CD entries).

Conclusion: The data suggests that there is a strong correlation between the Event and the incidences of 'Correction' entries.

c. A subset of reasons underlies all delays

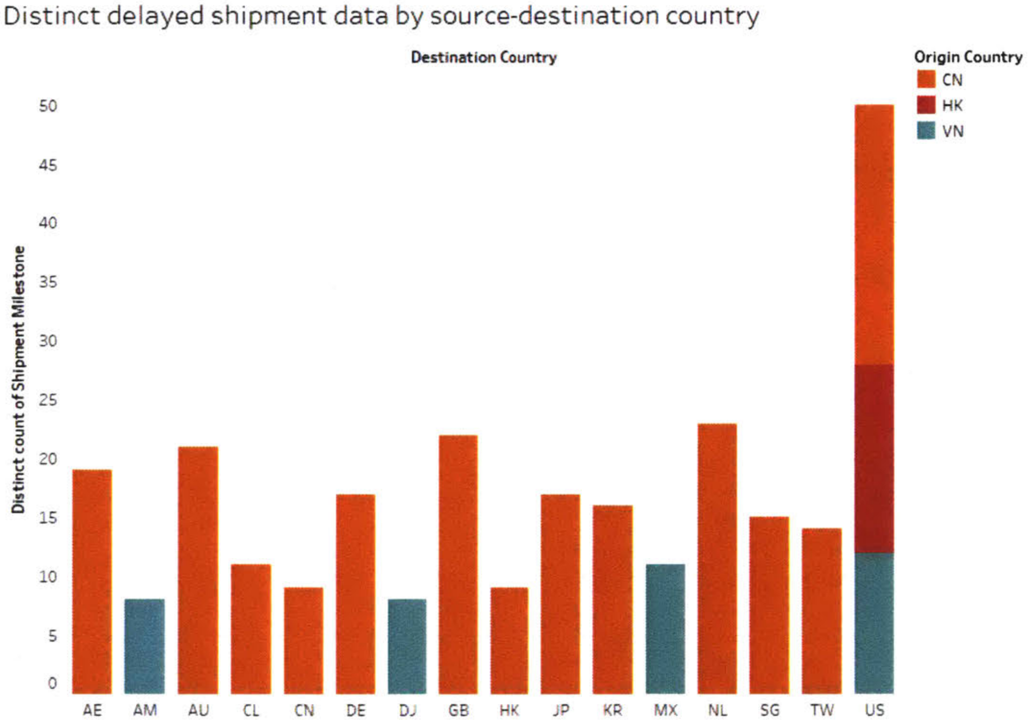


Figure 52: Distinct delayed shipment data by source-destination country

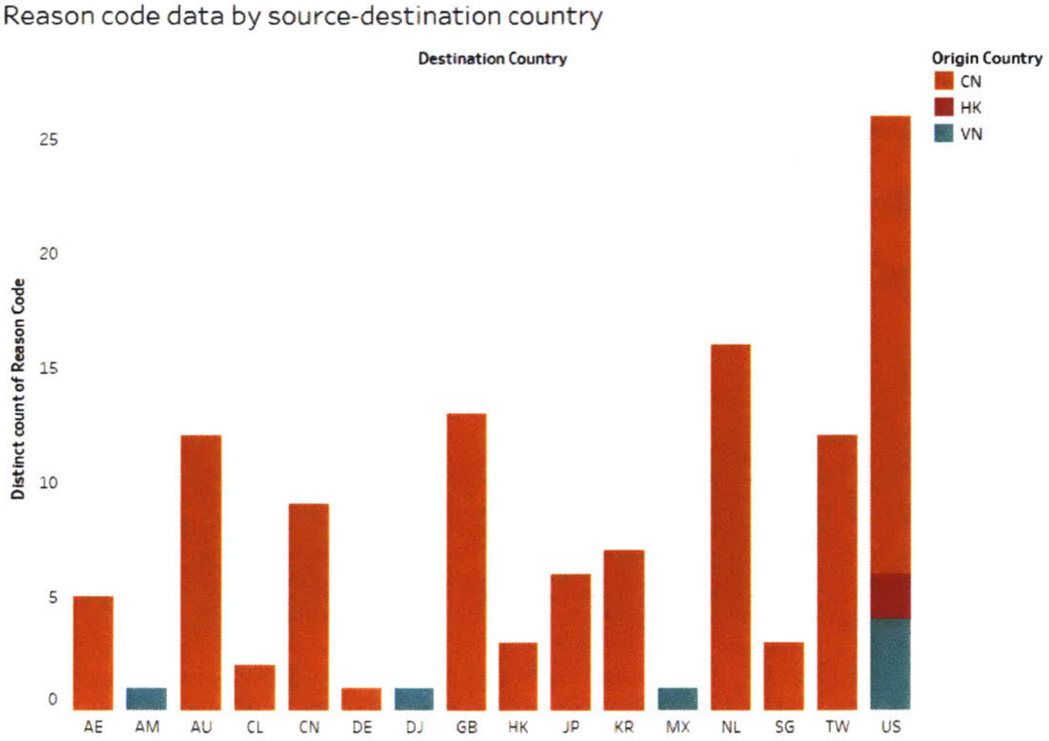


Figure 53: Reason code data by source-destination country

As can be expected, the maximum number of delay events occur for trades from China to US (Figure 52, Figure 53). This is expected because most of the transactions are for this source-destination pair. There are 19 unique reason codes that cover these transactions over 559 instances.

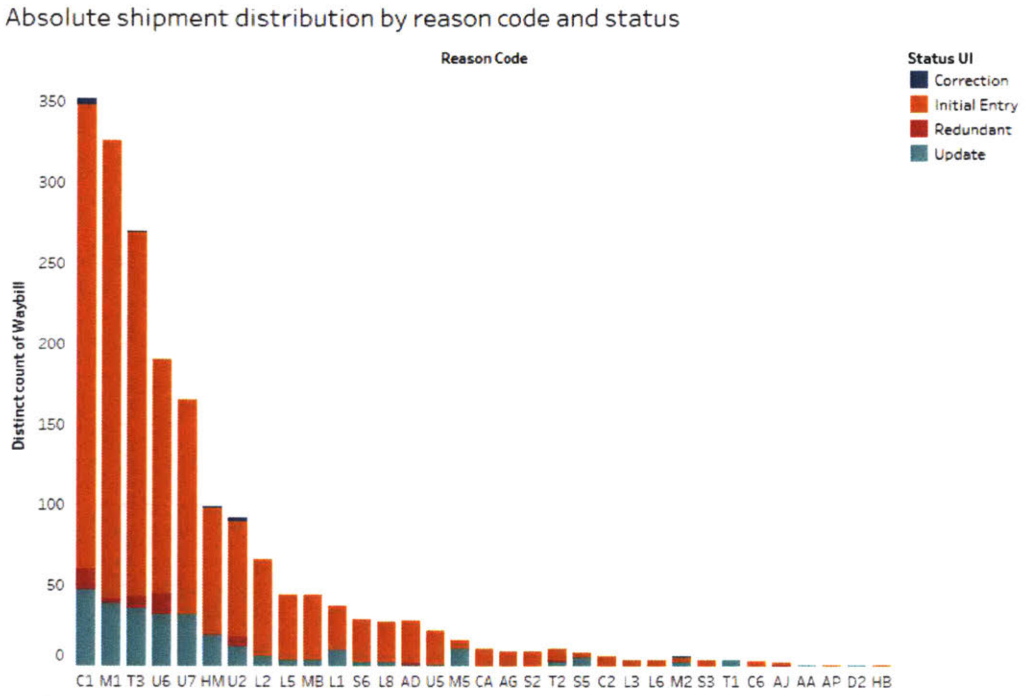


Figure 54: Absolute shipment distribution by reason code and status

When we look at the absolute distribution of delayed shipments against the reason codes (Figure 54), we see that a large portion of them are driven by C1 - "Late delivery due to Customer request", M1 - "Wrong Lead Times" and T3 - "Customs Clearance Delay".

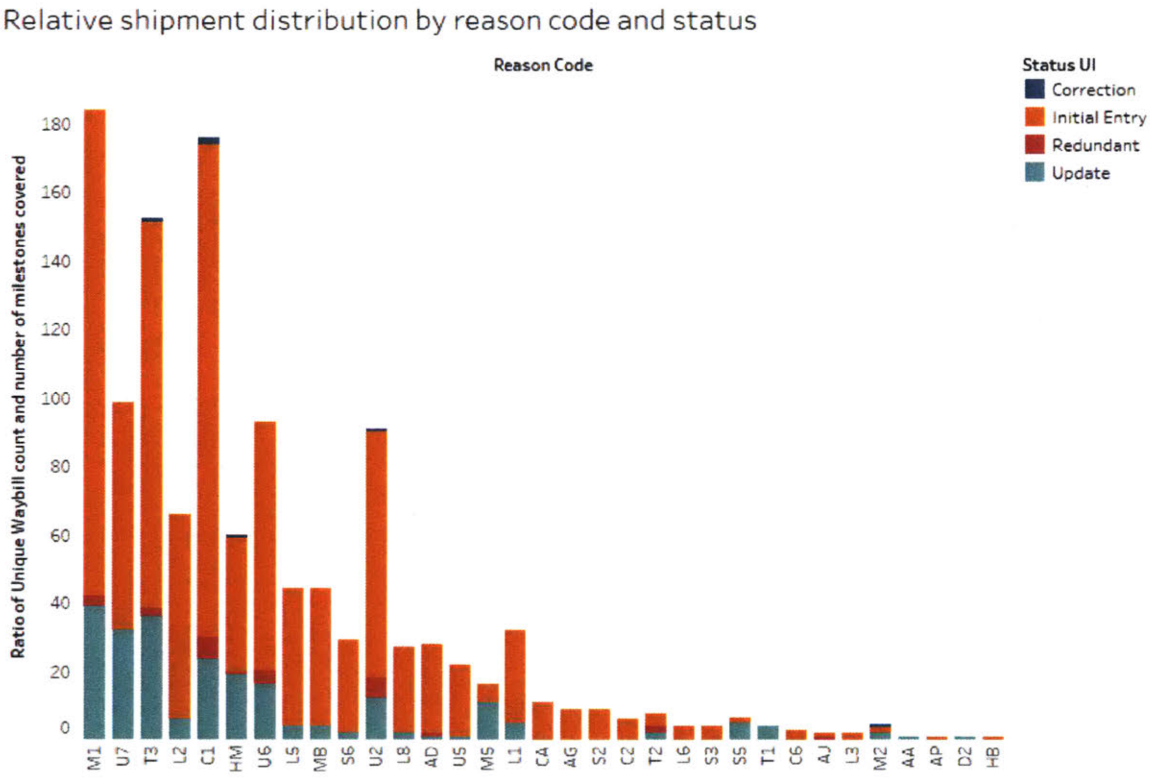


Figure 55: Relative shipment distribution by reason code and status

We then analyzed how the distribution changes when we plot the transaction data weighted by number of shipment events covered (Figure 55). For most of the shipments, the delay is attributable to the reason code M1 - "Wrong Lead Times", C1 - "Late delivery due to Customer request" and T3 - "Customs Clearance Delay".

Also, the concentration of updates is significantly high for these records (aside from initial entries). This is expected because the main events for which the reason-codes are tracked are SD and AG. All entries after the initial entry against the SD event for a shipment will be treated as updates.

Conclusion: M1, C1 and T3 are the most common reasons for delays of shipments.

d. Edited Fields have a correlation with shipment mode, reason code, consignee, event, and status

Below, we analyzed the variation of Edited fields by the shipment mode:

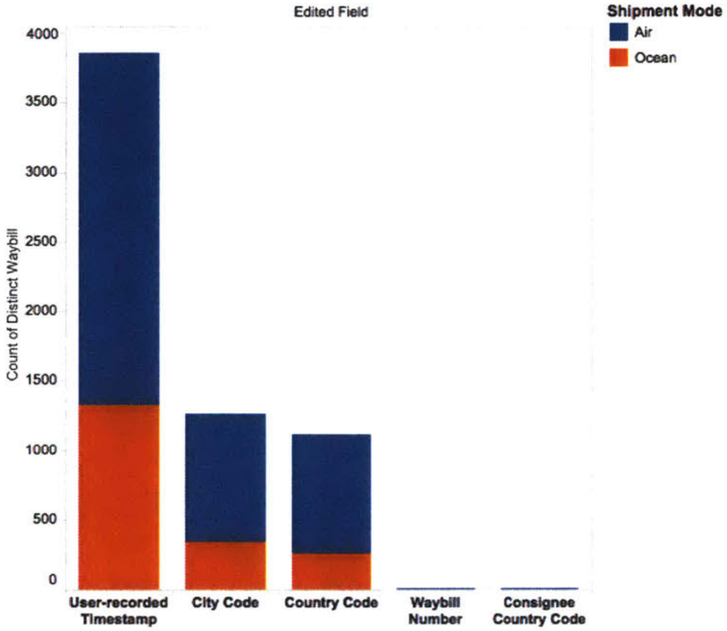


Figure 56: Edited Fields by Mode

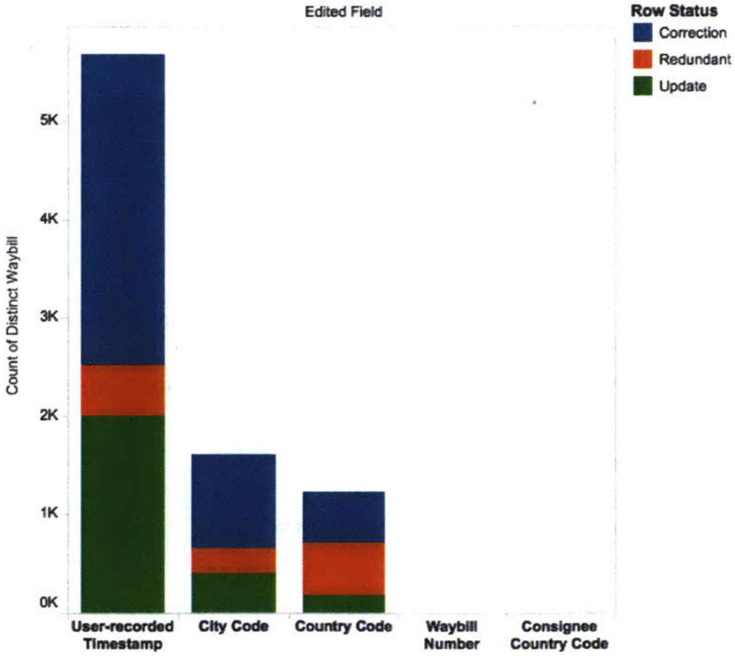


Figure 57: Edited Fields by Status

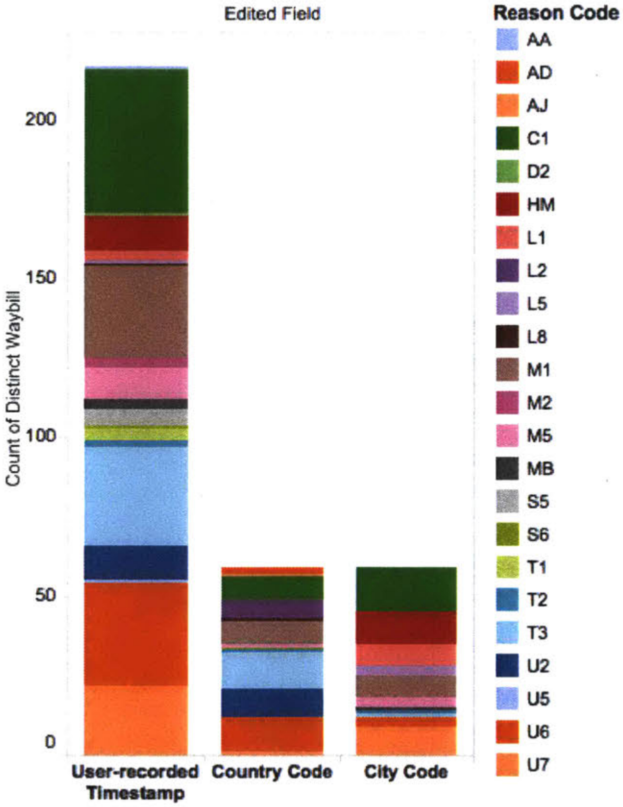


Figure 58: Edited Fields by Reason Code

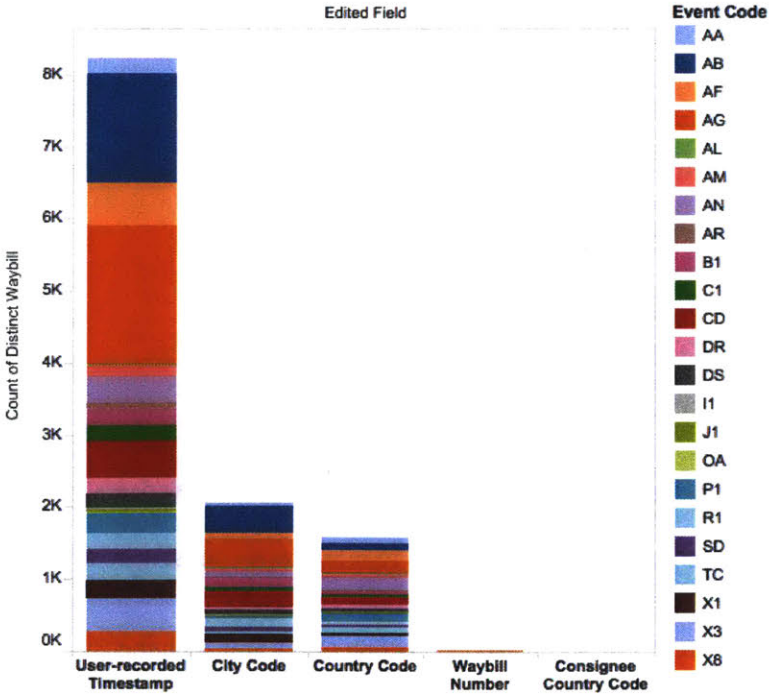
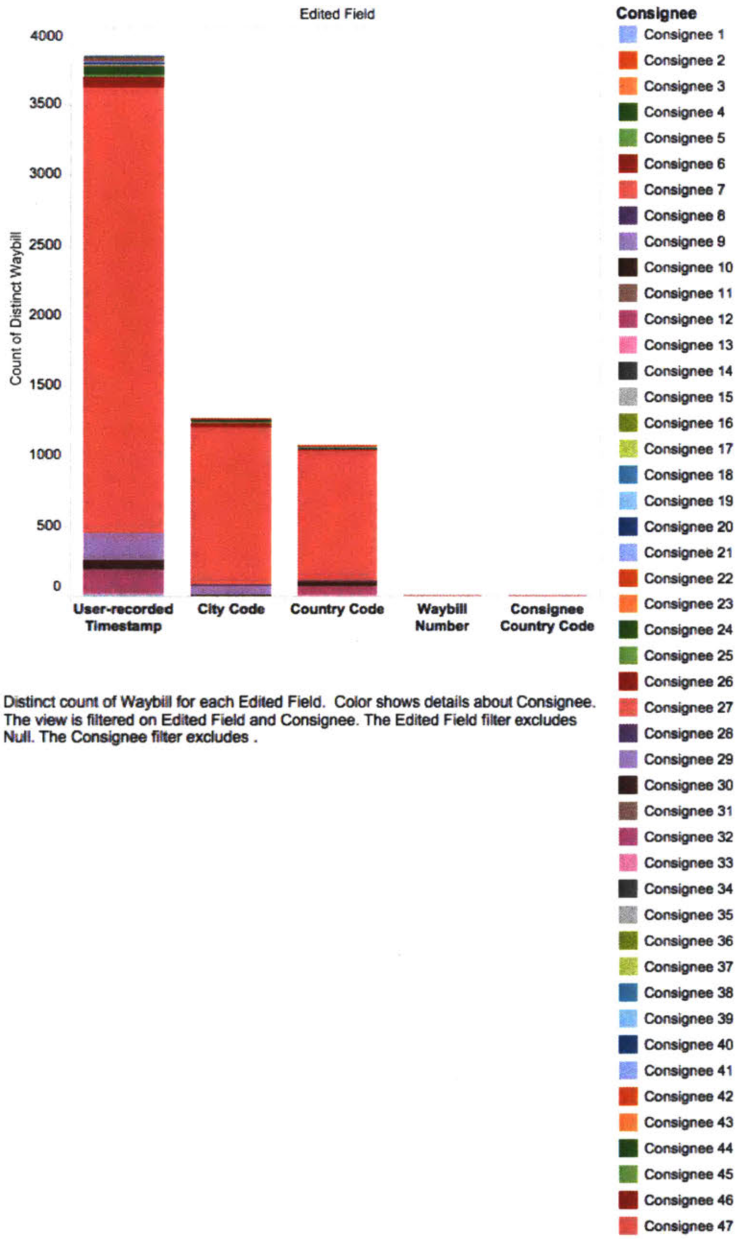


Figure 59: Edited Fields by Event Code



Distinct count of Waybill for each Edited Field. Color shows details about Consignee. The view is filtered on Edited Field and Consignee. The Edited Field filter excludes Null. The Consignee filter excludes .

Figure 60: Edited Fields by Consignee

Conclusion: We analyzed the variation of Edited fields by shipment mode, reason code, consignee, event, and status. As expected, the size of the bars is higher for Air than for Ocean, since the data covers more air shipments. Thus, one can infer that the increase in shipments leads to more corrections.

Furthermore, the figures show that Corrections make up the largest portion of the status changes. Reason codes C1 (Late delivery due to Customer request) and U6 (Weather) constitute the largest portions of reason codes. The number of event codes AG (FETA Date Requirements Throughout) and AB (Delivery Appt or Appt Confirmed) is the highest. And, Consignee 27 is the most common consignee.

4.3.1.3 Classification using K- Means

We use the K-means clustering approach to cluster the data based on its shipment attributes. As described in 3.3.2 Predictive Analysis, K-means is a non-hierarchical clustering method that aims to find k records in the training dataset that are similar to a new record. Given the large size of the dataset, K-means is preferable (over hierarchical clustering). The similarity is computed using distance between points. The most commonly used measure is the Euclidian distance. Other distance measures that can be used are statistical distance (also known as Mahalanobis distance and which takes into account the correlation between measurement), Manhattan distance and maximum coordinate distance. Similarity measures are often used, instead of distance measure, for categorical variables. From several K-NN models on the training set (with different K values), we choose the value of K which gives the best classification performance. JMP provides a criterion, the Cubic Clustering Criterion (CCC), to help determine the number of clusters to be produced. Additionally, an elbow chart of the number of clusters plotted against average-within cluster distance can be used to visually ascertain the suitable K value.

We perform the clustering of the dataset using the following three variables: 'TimeSinceFirst', 'TimeSinceLast' and a calculated field 'DaysSinceFirst'.

'DaysSinceFirst counts the number of days since the entry of the first record of the dataset. After running the clustering model with various K values, we select the K value as: 6 given its CCC performance shown in Table 5:

Table 5: K-Means Clustering CCC

Method	NCluster	CCC	Best
K-Means Clustering	3	-81.46	
K-Means Clustering	7	-16.18	
K-Means Clustering	5	-9.8189	
K-Means Clustering	4	-22.66	
K-Means Clustering	6	-5.4729	Optimal CCC

Table 6 and Figure 61 show the distribution of records into clusters:

Table 6: Count of records per cluster

Cluster	Count
1	5873
2	16
3	1161
4	490
5	5830
6	562

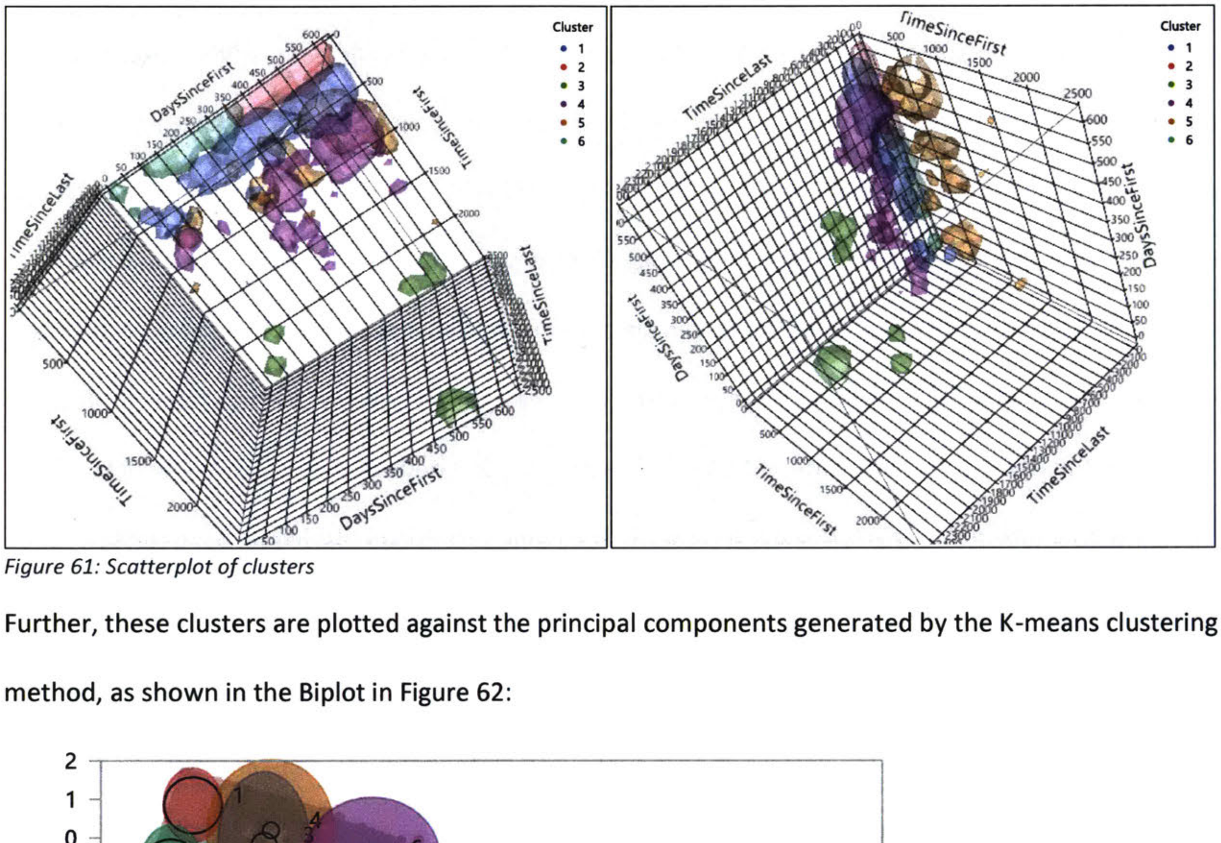


Figure 61: Scatterplot of clusters

Further, these clusters are plotted against the principal components generated by the K-means clustering method, as shown in the Biplot in Figure 62:

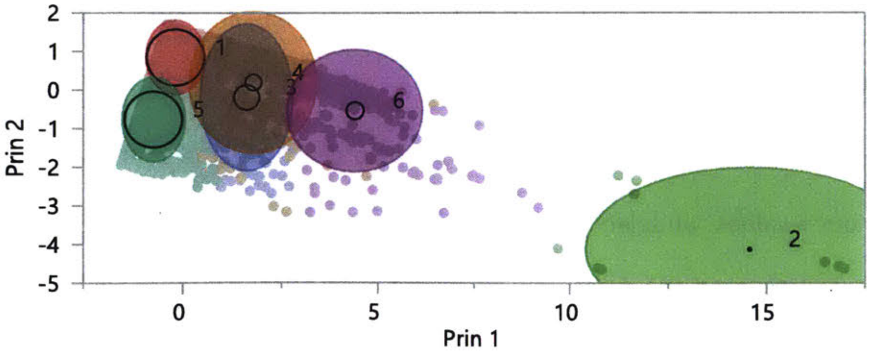


Figure 62: K-means Biplot

Conclusion: We observe six key clusters in the data with the range of values of 'TimeSinceLast' and 'TimeSinceFirst'. As can be seen in Figure 61, two clusters (green and brown) are distinct from the other clusters with little overlap. The green cluster corresponds to records with high value of 'TimeSinceLast' and ‘TimeSinceFirst’ and brown for low values of the same. The other clusters have some overlap but are still significantly separate from the rest along one or more dimension. The extent of separation between the individual clusters, however, is insufficient for us to draw any meaningful insights about the transactions.

4.3.2 Predictive Analytics

One of the constraints of the available data is that although it has a large number of attributes, most of these attributes are categorical variables, which do not lend themselves well to predictive analysis. Two techniques that perform reasonable well with categorical variables have been applied in this section using the JMP software. These are Naïve-Bayesian classification and Neural networks.

Naïve –Bayes classifier

We developed Naïve-Bayes models to predict the value of 'IsFinalEntry' and 'Status' in the following section.

a. **Naïve Bayes model to predict 'IsFinalEntry' using Event-code, Event-location, User, shipment mode, 'HasInputError'**

The result of the Naïve Bayes classification is shown in Table 7.

Table 7: Naive Bayes-1 classification result

	Most Probable(IsFinalEntry)			
	0		**1**	
IsFinalEntry	**% of Total**	**N**	**% of Total**	**N**
0	2.56%	6788	8.10%	21466
1	5.71%	15138	83.62%	221532

To assess the performance of this model, we need to compare the misclassification rate of our model with the misclassification for a naïve model. The naïve classification rule would classify all records as having 'IsFinalEntry' = 1 (the most frequent value). Therefore, it would misclassify 10.66% (2.56% + 8.10%) of the input records. In comparison, our model misclassifies 13.81% (5.71%+8.10%) of the records. Therefore, our naïve-Bayes model performs worse than a naïve rule that ignores predictor values.

b. Naïve Bayes model to predict 'IsFinalEntry' using origin, destination city and destination country

The result of the Naïve Bayes classification is shown in Table 8.

Table 8: Naive Bayes-2 classification result

	Most Probable(IsFinalEntry)			
	0		1	
IsFinalEntry	% of Total	N	% of Total	N
0	117	0.06%	20316	10.96%
1	238	0.13%	164659	88.85%

The naïve classification misclassifies 11.02% (0.06% + 10.96%) of the input records. In comparison, our model misclassifies 11.09% (0.13%+10.96%) of the records. Therefore, our naïve-Bayes model performs slightly worse than a naïve rule that ignores predictor values.

c. Naïve Bayes model to predict 'Status' using Event-code, Event-location and User

The result of the Naïve Bayes classification for predicting the status is shown in Table 9.

Table 9: Naive Bayes-3 classification result

	Most Probable(Status_UI)							
	Correction		Initial Entry		Redundant		Update	
Status_UI	N	% of Total	N	% of Total	N	% of Total	N	% of Total
Correction	392	0.29%	7933	5.97%	2	0.00%	5	0.00%
Initial Entry	408	0.31%	115177	86.62%	3	0.00%	3310	2.49%
Redundant	13	0.01%	2201	1.66%	1	0.00%	59	0.04%
Update	0	0.00%	1468	1.10%	0	0.00%	1993	1.50%

The Status value that we are interested in classifying is 'Correction' as it represents the number of errors present in the data. The naïve rule would classify all values as not requiring a correction ('Others'). The adjusted confusion matrix is shown in Table 10:

Table 10: Naive Bayes-3 adjusted confusion matrix

	Predicted Count	
Status_UI	Correction	Others
Correction	392	7940
Others	421	124212

The naïve classification misclassifies 6.27% of the input records. In comparison, our model misclassifies 6.29% of the records. Therefore, our naïve-Bayes model performs slightly worse than a naïve rule that ignores predictor values.

d. Naïve Bayes model to predict 'Status' using Event code, Mode and Destination city

The result of the Naïve Bayes classification for predicting the status is shown in Table 11 and the adjusted confusion matrix is shown in

Table 12.

Table 11: Naive Bayes-4 classification result

	Correction		Initial Entry		Update	
Status_UI	**N**	**% of Total**	**N**	**% of Total**	**N**	**% of Total**
Correction	33	0.02%	12024	6.49%	0	0.00%
Initial Entry	30	0.02%	163871	88.42%	980	0.53%
Redundant	2	0.00%	3682	1.99%	17	0.01%
Update	0	0.00%	3941	2.13%	750	0.40%

Table 12: Naive Bayes-4 adjusted confusion matrix

	Predicted Count	
Status_UI	**Correction**	**Others**
Correction	33	12024
Others	32	173241

The naïve classification misclassifies 6.506% of the input records. In comparison, our model misclassifies 6.505% of the records. Therefore, our naïve-Bayes model performs slightly better than a naïve rule that ignores predictor values.

Conclusion: Naïve Bayes models are useful for predicting with categorical variables as predictors. For our specific data set, and operating under computational limitations of the software used, we did not find a Naïve-Bayes model which significantly out-performs a naïve approach to classification (to the largest class). The Naïve Bayes model which we used to predict the 'Status' of a record using Event code, Mode and Destination city performs marginally better than the naïve approach.

Despite this result, the Naïve-Bayes approach is definitely promising for our data set given the categorical nature of the variables. The predictive performance of the model is expected to improve with the inclusion of more predictor variables.

Neural network

We develop four different neural nets by selecting two sets (input, output) of variables pairs. For neural net modeling, JMP automatically splits the data into training and validation data. As explained in section 3.3.2 Predictive Analysis, we use the default value of 'Random Holdback' (33%) and the default activation function selection as TanH (sigmoid function) for the hidden layers. We compare the performance of the two neural nets by analyzing the performance measures for both the training and the validation data.

a. Neural net using Event-code, Event-city, Destination- City and Origin-City to predict if the record 'IsFinalEntry' =1

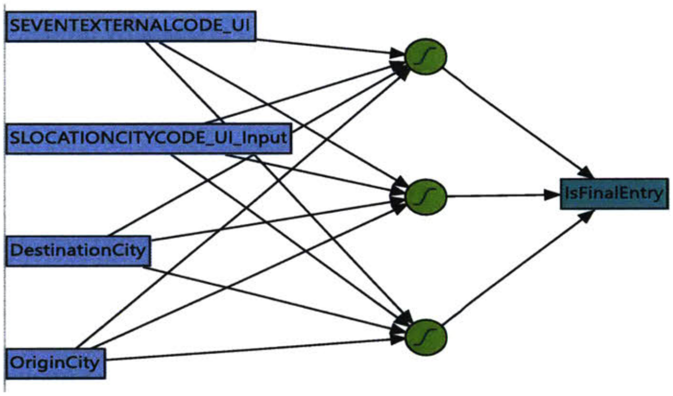


Figure 63: Neural Net 1 Model

To assess the model performance, we look at the fit-measures (Table 13), the confusion rates for the model and validation datasets (Table 14) and the Receiver - Output - characteristics (Figure 64):

Table 13: Neural Net 1 Performance

Measures	Training Data	Validation Data
Generalized RSquare	0.2681075	0.2099155
Entropy RSquare	0.212545	0.1639264
RMSE	0.2660061	0.2720001
Mean Abs Dev	0.1413846	0.1437831
Misclassification Rate	0.0904319	0.0932656

Table 14: Neural Net 1 Confusion Matrix and Rates

	Training Data				Validation Data			
Actual	Predicted Count		Predicted Rate		Predicted Count		Predicted Rate	
IsFinalEntry	0	1	0	1	0	1	0	1
0	582	4417	0.116	0.884	229	2271	0.092	0.908
1	321	47073	0.007	0.993	172	23522	0.007	0.993

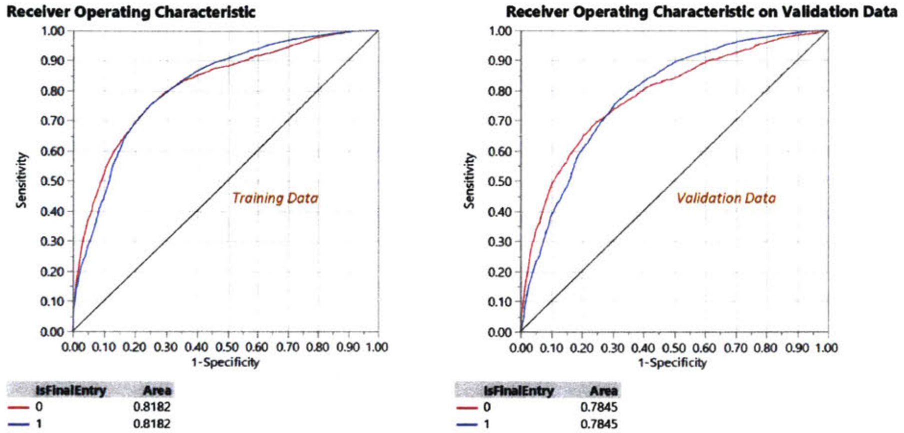


Figure 64: Neural Net 1- ROC curve

Performance summary: The Generalized RSquare is based on the likelihood function. The Entropy RSquare compares the log-likelihoods from the fitted model and the constant probability model. The model appears to have a poor performance if we consider only the Generalized RSquare, Entropy RSquare and RMSE. These are goodness of fit measures and don't indicate prediction accuracy. However, the model has a low misclassification rate of 9%. Additionally, the ROC curve is close to the top-left with high 'Area under the curve' values, suggesting high model sensitivity and specificity. This in turn implies low rate of false-positives and false-negatives (refer section 3.3.2 Predictive Analysis). For two classes C1 (important class, for example- 'IsFinalEntry') and C0 ('IsNotFinalEntry'), false positive is the proportion of C1 predictions which are wrong (incorrect classification of records as final) and false negatives are the proportion of C0 predictions which are wrong (incorrect classification of records as not-final).

Finally, the confusion rates suggest that the model performs better than a naïve model (which would set IsFinalEntry = 1 for all records). The naïve method has a misclassification rate of 9.54%, for training and validation data, whereas our model has a misclassification rate of 9.04% for the training data and 9.32% for the validation data. Note that most of the records (89.4%) have 'IsFinalEntry' value=1.

b. Neural net using Event-code, Event-city, 'TimeSinceLast', Shipment Mode and Destination- City to predict if the record 'IsFinalEntry' =1

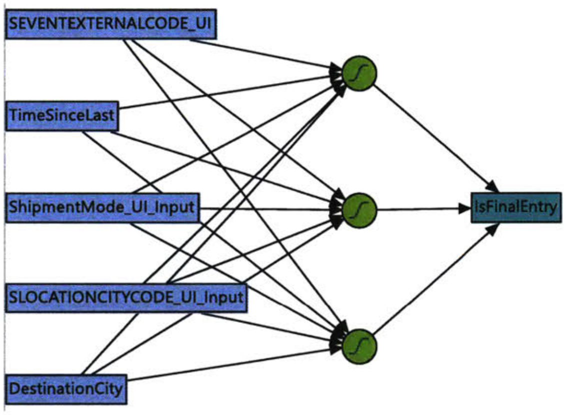
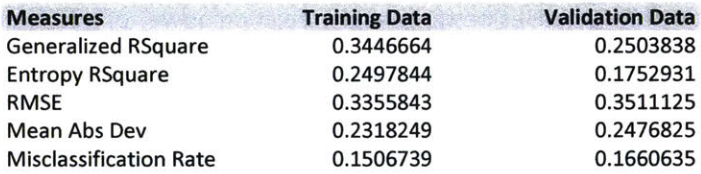


Figure 65: Neural Net 2 Model

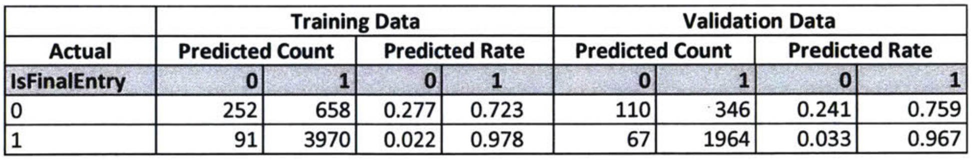
Table 15, Table 16 and Figure 66 below show the model performance metrics:

Table 15: Neural Net 2 Performance



Measures	Training Data	Validation Data
Generalized RSquare	0.3446664	0.2503838
Entropy RSquare	0.2497844	0.1752931
RMSE	0.3355843	0.3511125
Mean Abs Dev	0.2318249	0.2476825
Misclassification Rate	0.1506739	0.1660635

Table 16: Neural Net 2 Confusion Matrix and Rates



	Training Data				Validation Data			
Actual	Predicted Count		Predicted Rate		Predicted Count		Predicted Rate	
IsFinalEntry	0	1	0	1	0	1	0	1
0	252	658	0.277	0.723	110	346	0.241	0.759
1	91	3970	0.022	0.978	67	1964	0.033	0.967

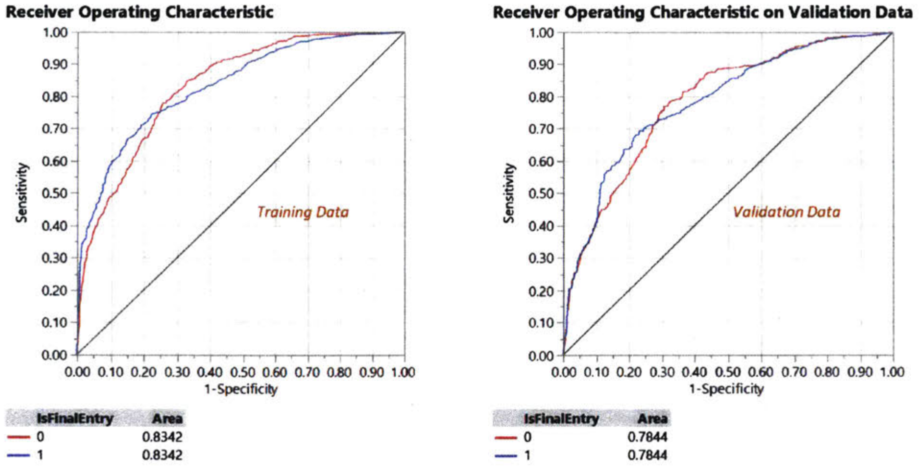


Figure 66: Neural Net 2- ROC curve

Performance summary: Although the Generalized RSquare, Entropy RSquare and RMSE are significantly better for this model than for Neural net -1 in part (a), the model has a higher mean-absolute deviation and misclassification rate of 15% compared to 9% for the previous model. The ROC curve is close to the top-left with high Area under the curve (AUC) values. The area under the curve for the training data is higher for this model than the first, but the area under the curve for the validation data is the same. High value of AUC implies better fit and prediction power of this model. Finally, the confusion rates suggest that the model performs better than a naïve model (which would set IsFinalEntry = 1 for all records). The naïve method has a misclassification rate of 18.3%, for training and validation data, whereas our model has a misclassification rate of 15.06% for the training data and 16.6% for the validation data. This is poor in comparison to model 1's performance (~9%). Further, the large number of misclassifications are false-positives (Predicted value of IsFinalEntry=1, Actual value=0). For us this means that we misclassify a large number of non-final entries as being final.

c. **Neural net using Event-code, Event-city, Destination- City and Origin-City to predict the Status of the record**

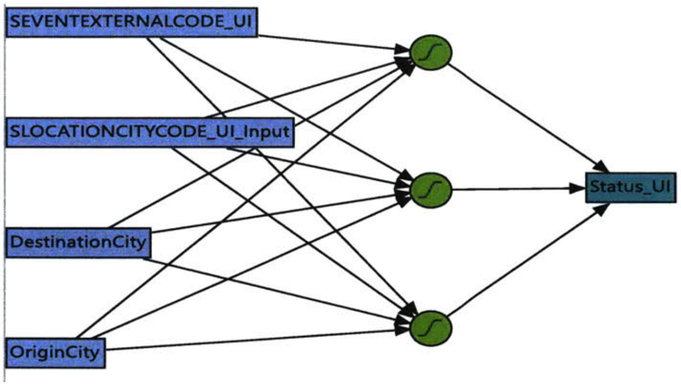


Figure 67: Neural Net 3 model

Table 17, Table 18, Table 19 and Figure 68 below show the model performance metrics:

Table 17: Neural Net 3 Performance

Measures	Training Data	Validation Data
Generalized RSquare	0.3907759	0.3466103
Entropy RSquare	0.3026084	0.2644118
RMSE	0.2744576	0.2800839
Mean Abs Dev	0.1461561	0.1494117
Misclassification Rate	0.0919051	0.0955108

Table 18: Neural Net 3 Confusion Matrix and Rates (Training)

Actual	Predicted Count				Predicted Rate			
Status_UI	Correction	Initial Entry	Redundant	Update	Correction	Initial Entry	Redundant	Update
Correction	153	2887	0	0	0.050	0.950	0.000	0.000
Initial Entry	69	47212	0	106	0.001	0.996	0.000	0.002
Redundant	1	553	0	2	0.002	0.995	0.000	0.004
Update	0	1197	0	211	0.000	0.850	0.000	0.150

Table 19: Neural Net 3 Confusion Matrix and Rates (Validation)

Actual	Predicted Count				Predicted Rate			
Status_UI	Correction	Initial Entry	Redundant	Update	Correction	Initial Entry	Redundant	Update
Correction	59	1462	0	0	0.039	0.961	0.000	0.000
Initial Entry	57	23551	0	83	0.002	0.994	0.000	0.004
Redundant	4	275	0	0	0.014	0.986	0.000	0.000
Update	0	621	0	84	0.000	0.881	0.000	0.119

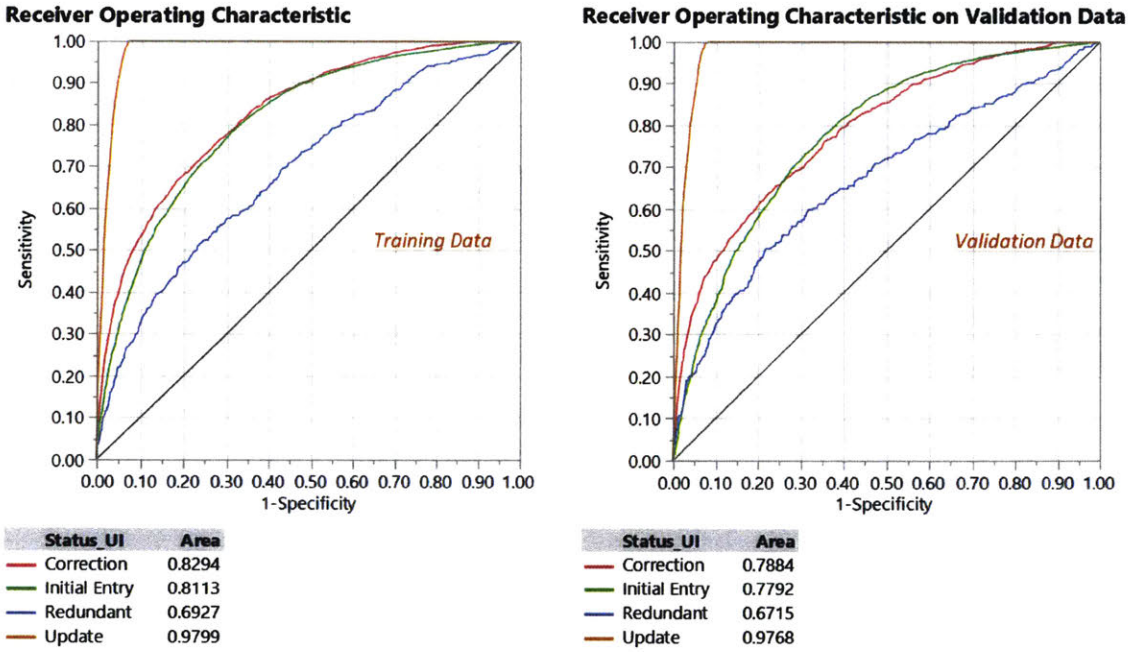


Figure 68: Neural Net 3- ROC curve

Performance summary: The Generalized RSquare, Entropy RSquare and RMSE are low, indicating a poor model fit. But in terms of predictive performance, the model has a low misclassification rate at 9%. The ROC curve is close to the top-left with high Area under the curve (AUC) values. But the area under the curve for the 'Correction' data is lower than desired. This value of AUC implies unsatisfactory fit and prediction power of the model.

d. Neural net using Event-code, Event-city, 'TimeSinceLast', Shipment-Mode and Destination- City to predict the status of the record

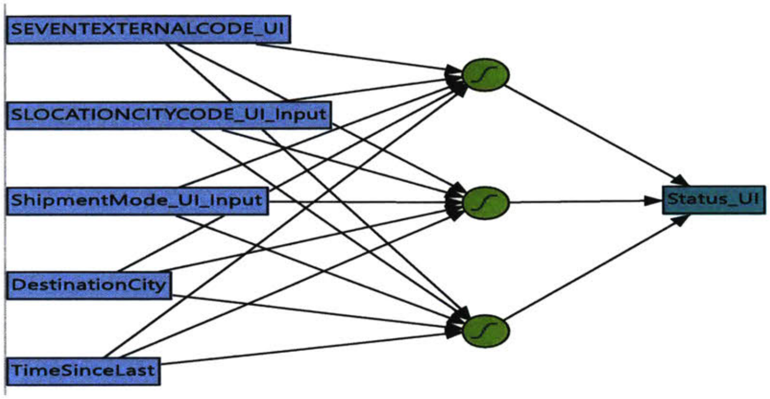


Figure 69: Neural Net 4 model

Table 20, Table 21, Table 22 and Figure 70 below show the model performance metrics:

Table 20: Neural Net 4 Performance

Measures	Training Data	Validation Data
Generalized RSquare	0.8832432	0.8271848
Entropy RSquare	0.7411954	0.6503079
RMSE	0.2621329	0.3015284
Mean Abs Dev	0.140029	0.1626341
Misclassification Rate	0.0877087	0.1174105

Table 21: Neural Net 4 Confusion Matrix and Rates (Training)

Actual	Predicted Count				Predicted Rate			
Status_UI	Correction	Initial Entry	Redundant	Update	Correction	Initial Entry	Redundant	Update
Correction	2944	0	72	2	0.975	0.000	0.024	0.001
Initial Entry	0	0	0	1	0.000	0.000	0.000	1.000
Redundant	317	0	197	42	0.570	0.000	0.354	0.076
Update	0	0	2	1394	0.000	0.000	0.001	0.999

Table 22: Neural Net 4 Confusion Matrix and Rates (Validation)

Actual	Predicted Count				Predicted Rate			
Status_UI	Correction	Initial Entry	Redundant	Update	Correction	Initial Entry	Redundant	Update
Correction	1439	0	67	4	0.953	0.000	0.044	0.003
Initial Entry	0	0	0	0	.	.	.	.
Redundant	193	0	63	23	0.692	0.000	0.226	0.082
Update	1	0	4	693	0.001	0.000	0.006	0.993

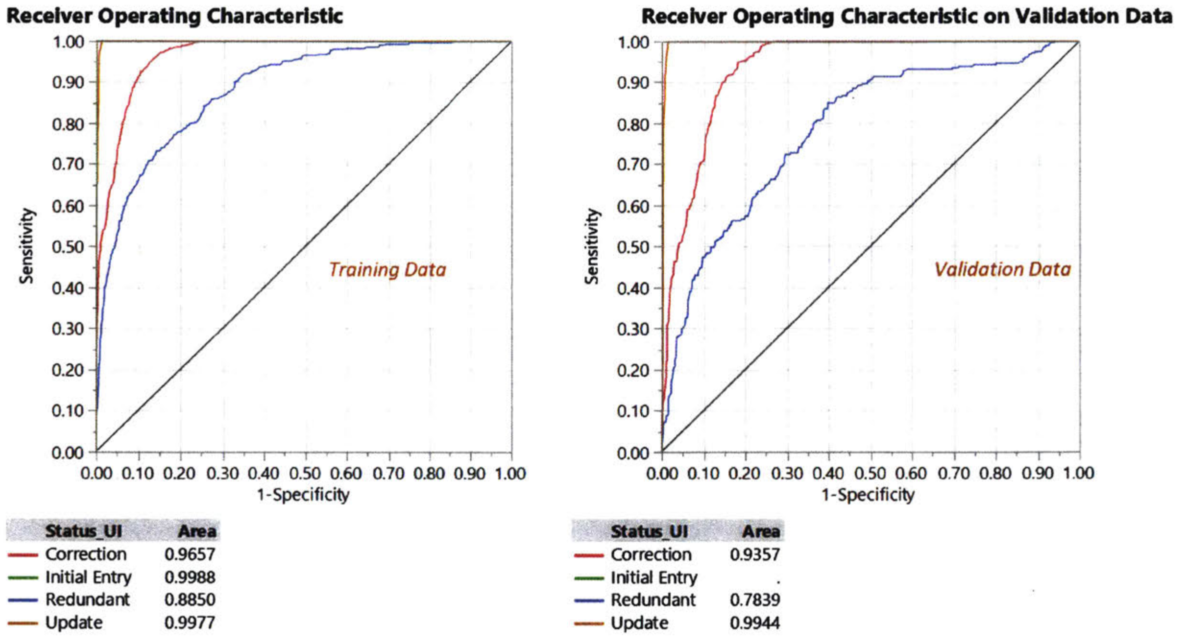


Figure 70: Neural Net 4- ROC curve

Performance summary: The Generalized RSquare and Entropy RSquare are high for this model, although the RMSE is poorer in comparison to Neural Net -3 in part (c). The model has a higher mean-absolute deviation and misclassification rate of 14% to 16% compared to 9% for the Neural-Net - 3. The ROC curve is close to the top-left with high Area under the curve values. The area under the curve for the training data and the validation data is higher for this model than all other models. We are particularly interested in the Area under the curve (AUC) for the 'Correction' data, which is significantly high in this neural net model. High value of AUC implies better fit and prediction power of the model.

Finally, the confusion rates suggest that the model performs significantly better than a naïve model. To understand the behavior of the Naïve model, we assume that the Naïve model would classify all records as the type 'Correction'. We do not distinguish between these three Status values and therefore we reconstruct the confusion matrix in the form shown:

Table 23: Neural Net-4 Adjusted Confusion Matrix

	Training Data		Validation Data	
Actual	**Predicted Count**		**Predicted Count**	
Status_UI	**Correction**	**Others**	**Correction**	**Others**
Correction	2944	74	1439	71
Others	317	1636	194	783

The naïve model misclassifies 39% of the records, whereas our model misclassifies 8% of the training data and 11% of the validation data per the confusion matrix above in Table 23.

Conclusion: Neural network models (Neural Net model 4) provide good prediction for the Status of each record when the input variables are Event-code, Event-city, 'TimeSinceLast', Shipment Mode and Destination- City. This suggests that looking at these attributes of a record, we can predict whether a record will be a 'Correction' to an error.

Neural – net model 2 can also be used to determine whether a given record is the final record (or likely to require a correction instead). This model however has a high rate of false positives and further analysis of the records classified as 'IsFinalEntry'=1 is required to reduce the misclassifications.

Improving predictive performance

Predictive performance of both the Naive-Bayes classifier as well as the neural networks model is expected to improve if we apply a stratified sampling approach prior to running the prediction model. This is because certain classes of data, for example the class of 'Redundant' and 'Update' records, form a smaller proportion of the total dataset. Stratified sampling can be used to oversample these rare classes and improve the performance of the classifiers (Shmueli et al., 2016). This method is better suited for data which has 2 classes, as is the case for our response variable 'IsFinalEvent'.

4.3.3 Prescriptive analytics

Based on our analysis of the underlying patterns of error and their relationship to attributes of the shipment, this case study is not suitable for performing prescriptive analytics for the following reason:

- The data is entered into the user interface using drop-down menus for the most critical fields (except reasons and comments). This eliminates the possibility of typographical errors.
- The most commonly impacted field during operational errors is the event-time. This field does not lend itself well to prescriptive analysis since the changes in the event-time are not driven solely by the attributes of the shipment. Therefore, defining business rules to correct event time stamps based on shipment attributes is unfeasible.
- The prediction accuracy of the models developed is low and limited to the classification of the records (as correct or erroneous). This is insufficient for us to determine a business rule to suggest or perform the corrections in errors.
- The availability of numerical fields would have allowed us to make a more robust predictive and prescriptive model. Data such as the commercial (dollar) value of the shipments would allow us to estimate the impact of the shipment delay in monetary terms. Even certain categorical variables such as the SKUs (in each shipment) would have been pivotal inputs for creating a more logical clustering model for the data.
- Further, business rules regarding how data error corrections are prioritized would have been beneficial in building a more robust predictive model.

4.4 Implications and Discussion

With respect to our thesis sponsor company, Damco, we gained significant insights on the nature and frequency of data errors which impede their ability to meet customer Service Level Agreements (SLAs). The proposed approach will accelerate and standardize the resolution of a portion of the shipment errors and provide savings in terms of time and cost.

This framework can be extended to data entry correction for other industries. In the future, it can be extended to include external data sources (such as news and weather) adding to the predictive power for operational exceptions.

4.5 Limitations

Here, we discuss the limitations of our data. Our approach in this thesis is developed bearing in mind the following constraints and requirements:

1. **Error correction validation**: the data available does not explicitly indicate whether an entry is correct or erroneous. We inferred this information from the context, such as via time variables and unique identifiers. Likewise, there was no denotation of the appropriate way of correcting the erroneous entries.
2. **Duration of data involved**: the data available for analysis is for 10 months only. Although this allows us to identify key correlations and trends between variables whose relationship is non-variable, it does not allow us to forecast future errors with a high degree of confidence.
3. **Data legacy systems**: the data has been extracted from multiple legacy systems. This makes the integration of the multiple data files challenging and in some cases impossible. Therefore, in some cases, two individual sets of analysis are performed, which are then used to cross validate our findings.
4. **Data type**: even though the data has a large number of attributes, most of the attributes are categorical variables, which are not very conducive to predictive analysis.

5. Conclusion

Our thesis focuses on analyzing the sources of errors in the tracking of shipments for a provider of global logistics solutions. We develop a general approach that will allow entities facing challenges similar to Damco's to investigate the relationship between shipment attributes and errors in data. To that end, we propose a data analytics framework that involves using descriptive, predictive, and prescriptive analytics. These methods can help identify the cause of the errors, forecast the probability of future error occurrence, and recommend applicable corrections, respectively.

We apply this framework, to the largest possible extent, to a real-world case study with data obtained from our thesis sponsor Damco. From our descriptive analysis, we derive three main insights: First, there is a correlation between the number of system and operational errors entered per event and a given unit of time (such as weekday, month, or quarter). Second, in absolute terms, most of the system errors are attributable to a specific user. Third, there is a strong relationship between the shipment – milestone and the 'TimeSinceLast' field. Predictive analysis using neural networks reveals that Event-code, Event-city, 'TimeSinceLast', Shipment-Mode and Destination-City can be used to predict the status of the record, particularly to determine whether a record will be a Correction or not. Damco could leverage these insights in ways ranging from redistributing its resources towards paying closer attention to certain events, to addressing the sources of errors correlated with specific users (Figure 35) or events (Figure 23). In this way, Damco can improve in cost, time, and quality of service to its clients.

One of the main constraints we encountered is the data's time span, which is less than a year and thus limits the potential inferences related to seasonality or for a high degree of confidence in the error forecasts. Likewise, the data does not identify entries as correct or incorrect, and does not indicate the applicable corrections for the incorrect entries. Further, the data integration is difficult as it is extracted from several differing legacy systems. Moreover, the K-means clustering method does not reveal any meaningful insights about the attributes of the records, and the predictive performance of the Naïve-Bayes and the neural net models is largely unsatisfactory in predicting whether a record is final. Therefore, we believe that entities seeking similar solutions should aim for a larger and richer dataset to allow for a more robust analysis. The insights derived in this research will allow these entities to identify opportunities for enhancements and expansions in their approach of gathering the data before initiating the analysis.

6. References

Albalate, A., & Minker, W. (2013). State of the Art in Clustering and Semi-Supervised Techniques. In Semi-Supervised and Unsupervised Machine Learning (pp. 15–89). John Wiley & Sons, Inc. Retrieved from http://onlinelibrary.wiley.com.libproxy.mit.edu/doi/10.1002/9781118557693.ch2/summary

Bowerman, B. L., O'Connell, R. T., & Koehler, A. B. (2005). Forecasting, time series, and regression : an applied approach. Belmont, Calif. : Thomson Brooks/Cole, c2005.

Celebi, M. E., Kingravi, H. A., & Vela, P. A. (2013). A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Systems with Applications, 40(1), 200–210. https://doi.org/10.1016/j.eswa.2012.07.021

Chang, X., Nie, F., Yang, Y., Zhang, C., & Huang, H. (2016). Convex Sparse PCA for Unsupervised Feature Learning. ACM Transactions on Knowledge Discovery from Data, 11(1), 1–16. https://doi.org/10.1145/2910585

De Finetti, B. (2010). Mathematical optimization in economics. [electronic resource] : lectures given at the Centro internazionale matematico estivo (C.I.M.E.) held in L'Aquila, Italy, August 30-September 7, 1965. Berlin ; London : Springer, 2010.

De Mauro, A., Greco, M., & Grimaldi, M. (2015). What is Big Data? A Consensual Definition and a Review of Key Research Topics. AIP Conference Proceedings, 1644(1), 97–104. https://doi.org/10.1063/1.4907823

Ehsan, N., & Faili, H. (2013). Grammatical and context-sensitive error correction using a statistical machine translation framework: GRAMMAR AND CONTEXT-SENSITIVE ERROR CHECKER. Software: Practice and Experience, 43(2), 187–206. https://doi.org/10.1002/spe.2110

Galit Shmueli, Peter C. Bruce, Mia L. Stephens, & Nitin R. Patel. (2016). DATA MINING FOR BUSINESS ANALYTICS.

Jones, E. (2011). Supply-Chain Analytics: Solving the Future. Biopharm International, 24(4), 50,48. Retrieved from http://search.proquest.com.libproxy.mit.edu/docview/859258034/abstract/926BB23612C24D5APQ/1.

Kou, G., Ergu, D., Peng, Y., & Shi, Y. (2013). Data Processing for the AHP/ANP (Vol. 1). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-642-29213-2

Niculescu, C., & Persson, L. E. (2006). Convex functions and their applications: a contemporary approach. New York: Springer.

Oliveira, M. P. V. de, McCormack, K., & Trkman, P. (2012). Business analytics in supply chains – The contingent effect of business process maturity. Expert Systems with Applications, 39(5), 5488–5498. https://doi.org/10.1016/j.eswa.2011.11.073

Oreifej, O., & Shah, M. (2014). Robust Subspace Estimation Using Low-Rank Optimization. Springer. Retrieved from http://link.springer.com/content/pdf/10.1007/978-3-319-04184-1.pdf

Ouazzane, K., Jun, L., Kazemian, H., Yanguo, J., & Boyd, R. (2012). An Artificial Intelligence-based language modeling framework. Expert Systems With Applications, 39(5), 5960-5970. doi:10.1016/j.eswa.2011.11.121.

SAS Institute Inc. (2017a). JMP 12 Online Documentation, Overview of Neural Networks. Retrieved April 21, 2017, from http://www.jmp.com/support/help/Overview_of_Neural_Networks.shtml

SAS Institute Inc. (2017b). JMP 12 Online Documentation, The Logistic Fit Report. Retrieved May 3, 2017, from http://www.jmp.com/support/help/The_Logistic_Fit_Report.shtml

Schaffhauser, D. (2014). 12 Essentials of Prescriptive Analytics.pdf.

Souza, G. C. (2014). Supply chain analytics. Business Horizons, 57(5), 595–605. https://doi.org/10.1016/j.bushor.2014.06.004

Stárka, J., Svoboda, M., Sochna, J., Schejbal, J., Mlýnková, I., & Bednárek, D. (2012). Analyzer: A Complex System for Data Analysis. Computer Journal, 55(5), 590-615.

Stat Trek. (2017). Matrix Rank. Retrieved May 4, 2017, from http://stattrek.com/matrix-algebra/matrix-rank.aspx

Underwood, J. (2014, January 21). Prescriptive Analytics: Making Better Decisions with Simulation [Web log post]. Retrieved May 4, 2017, from http://www.b-eye-network.com/view/17224.

Wang, G., Gunasekaran, A., Ngai, E. W. T., & Papadopoulos, T. (2016). Big data analytics in logistics and supply chain management: Certain investigations for research and applications. International Journal of Production Economics, 176, 98–110. https://doi.org/10.1016/j.ijpe.2016.03.014