**Automated Quality Assurance Alerts for Clickstream Anomalies on a Retail Website**

Alan Scieszinski

Department of Data Science, University of Wisconsin – Stevens Point

DS785: Capstone  
December 8, 2024

**Abstract**

Like many e-commerce websites, Acme Clothing’s website is continuously being refined and updated. When the site is updated, there can be unforeseen consequences. For example, the digital tags can start failing to fire the data they should. Alternatively, even more concerning, online orders placed by customers can stop being collected by the order management systems. Today, many hours of manually clicking and debugging the website are required to ensure the data quality. This human quality assurance is limited to the time and speed of the worker. In addition, it is more open to human error than a programmatic approach. This client-based project describes the programmatic automation of the quality assurance of clickstream data anomalies using unsupervised and semi-supervised machine learning algorithms. It also shows how joining offline with online data produces new business insights. It displays the data in online dashboards to facilitate the company's ongoing business management and decision-making needs regarding clickstream data.

Keywords: anomaly detection, clickstream, unsupervised machine learning, semi-supervised machine learning, dashboard

# Table of Contents

[Table of Contents 3](#_heading=h.30j0zll)

[List of Figures 5](#_heading=h.1fob9te)

[Chapter 1: Introduction 6](#_heading=h.3znysh7)

[Introduction 6](#_heading=h.2et92p0)

[Background of the Business Problem 6](#_heading=h.tyjcwt)

[Purpose and Significance of the Study 7](#_heading=h.3dy6vkm)

[Statement of the Overall Project Goals 8](#_heading=h.1t3h5sf)

[Project Objectives 9](#_heading=h.4d34og8)

[Definition of Terms 9](#_heading=h.2s8eyo1)

[Assumptions, Limitations, and Delimitations 9](#_heading=h.17dp8vu)

[Conclusion 10](#_heading=h.3rdcrjn)

[Chapter 2: Literature Review 11](#_heading=h.26in1rg)

[Introduction 11](#_heading=h.lnxbz9)

[Choosing a Digital Analytics Vendor While Managing Seen and Unseen Costs 11](#_heading=h.35nkun2)

[Restatement of Goals and Research Needs 14](#_heading=h.1ksv4uv)

[One Class Support Vector Machine (SVM) 16](#_heading=h.44sinio)

[Isolation Forest (iF) 16](#_heading=h.2jxsxqh)

[Local Outlier Factor (LOF) 16](#_heading=h.z337ya)

[Conclusion 17](#_heading=h.3j2qqm3)

[Chapter 3: Methodology 19](#_heading=h.1y810tw)

[Introduction 19](#_heading=h.4i7ojhp)

[Overview of Methodology with Multi-Dimensional Anomalies 19](#_heading=h.2xcytpi)

[Data collection 20](#_heading=h.1ci93xb)

[Data preparation 24](#_heading=h.3whwml4)

[Generate and evaluate models 26](#_heading=h.2bn6wsx)

[Conclusion 27](#_heading=h.qsh70q)

[Chapter 4: Results 29](#_heading=h.3as4poj)

[Introduction 29](#_heading=h.1pxezwc)

[Hyper Parameter Tuning and Results Evaluation 29](#_heading=h.49x2ik5)

[Conclusion 34](#_heading=h.2p2csry)

[Chapter 5: DISCUSSION 36](#_heading=h.147n2zr)

[Introduction 36](#_heading=h.3o7alnk)

[Summary of Findings 37](#_heading=h.23ckvvd)

[Summary of results organized by project objectives 38](#_heading=h.ihv636)

[Discussion of implications for business and suggestions for Future Research 39](#_heading=h.32hioqz)

[Conclusion 42](#_heading=h.1hmsyys)

[References: 43](#_heading=h.41mghml)

[APPENDIX 47](#_heading=h.2grqrue)

# List of Figures

[Figure 1 *Example of a single post\_event\_list variable* 1](#_heading=h.30j0zll)5

[Figure 2 *Distribution of the missing\_prod\_view variable.*](#_heading=h.30j0zll) 21

[Figure 3 *Distribution of the page\_error variable*](#_heading=h.30j0zll) 22

[Figure 4 *Distribution of the excluded\_hit variable* 23](#_heading=h.30j0zll)

[Figure 5 *Distribution of the page\_load\_time variable*](#_heading=h.30j0zll) 24

[Figure 6 *Distribution of the visit\_num variable* 2](#_heading=h.30j0zll)5

[Figure 7 *Distribution of the hit\_time\_gmt variable* 2](#_heading=h.30j0zll)6

[Figure 8 *Error factor by Browser Type* 2](#_heading=h.30j0zll)8

[Figure 9 *F1-Score Definition* 2](#_heading=h.30j0zll)9

[Figure 10 *F1 Scores by Hyper Parameter Tuning and Increasing Contamination Percent* 3](#_heading=h.30j0zll)3

[Figure 11 *Model Predicted Inliers vs Outliers of error factor by Unix Hit Time* 3](#_heading=h.30j0zll)4

[Figure 12 *Contamination Parameter Set to Auto* 3](#_heading=h.30j0zll)5

[Figure 13 *Contamination Parameter Set by Population Outlier %* 3](#_heading=h.30j0zll)6

[Figure 14 *Alert Threshold of Anomaly Contamination by Browser Type*](#_heading=h.30j0zll) 40

[Figure 15 *Marketing Channel Performance Dashboard*](#_heading=h.30j0zll) 44

# Chapter 1: Introduction

## Introduction

Many online retailers monitor their websites to ensure the business continues operating and serving their customers successfully. The collected data is further processed and analyzed, providing insight into customers’ shopping behaviors and pointing out opportunities for user experience improvements on the site. This data is vast and complex. It is an excellent example of the five Vs of Big Data (Volume, Velocity, Variety, Veracity, and Value). Data science can help extract a great deal more value from clickstream data.

This paper is a client-based project for the online clothing retailer Acme Clothing Inc. Acme Clothing's data science aims to utilize the data to uncover insights, improve the effectiveness of day-to-day operations, and improve overall decision-making. Data science will improve the quality and accuracy of clickstream data.

## Background of the Business Problem

Acme Clothing uses a third-party partner, Adobe Analytics, to embed digital tags on its website. These tags generate Server calls as users navigate and click actions on the website. Adobe Analytics collects and stores these server calls of clickstream data. The aggregation of the users’ interactions on a website is called “clickstreams.” Over time, these tags can break due to changes and improvements to the website. Acme Clothing employs people to inspect and maintain the accuracy of the digital tags actively, and the resulting server calls to Adobe Analytics.

The problem is that it is nearly impossible for a human to manually maintain the large volume and variety of digital tags, page types, web browsers, and operating system types. Different combinations of page types, web browsers, and operating system types can collect the clickstream data differently; they need separate consideration. It would take a human an enormous amount of time and effort to manually inspect every combination on every page. Small samples of pages quality assured. These samples are assured only on the browser and operating system installed on the analyst’s device. The company's website is continuously and iteratively being released/updated with improvements. Human QA focuses on the areas of the website impacted by a new release. QA accuracy on the pages, browsers, and uninspected operating systems is assumed. Bugs are discovered even after a thorough QA effort. Identifying the bugs can take several days, potentially losing the company's data and money.

Data science can passively supplement human manual maintenance with an automatic and programmatic review of the digital website tags and search for anomalies/errors. The programmatic approach can also look for browsers and operating systems outside the view of the analyst's systems. The approach will alert the analyst when anomalies are high enough to justify further review by a human. Data science techniques and methods can analyze a single day’s clickstream data with historically expected values and aggregated threshold tolerances that trigger the alert only when review is needed.

## Purpose and Significance of the Study

This study aimed to apply data science to solve general business problems directly. A human Web Analyst cannot manually check every web browser type the customers use when visiting Acme Clothing’s website. Not only would that take a tremendous amount of time, but it would likely have much human error. Human-only QA limits the quality, completeness, and speed of insights. QA insights can be made better and faster by taking a programmatic and data science approach. In addition, time otherwise spent manually clicking the website can be spent on higher-level analysis.

The secondary purpose of this project was to provide Acme Clothing with an in-depth review of current clickstreams' ingestion in their databases. The review highlighted opportunities to improve what is available in Acme Clothing's databases. Some variables exist in one clickstream and not the other. Adobe Analytics collects and stores all clickstream variables. Acme Clothing collects some of the clickstream variables into their databases. There is an opportunity to improve what is ingested into Acme Clothing databases and improve analytic breadth and capabilities.

Although it is outside the scope of this paper, a tertiary purpose of this project was to create a framework to report other clickstream insights. Website visits that convert to purchase can be isolated. The purchasing order IDs are then used as a primary key to join the backend databases containing customer relationship management (CRM), marketing/behavioral cohort, and demographic data. The resulting views into Acme Clothing’s customers will allow the company to measure better the health of their “customer file,” the effectiveness of marketing campaigns, and the effectiveness of customer diversification efforts. The ability to dig into these different slices of customers’ clickstreams adds significant value.

## Statement of the Overall Project Goals

Acme Clothing's realization of increased QA accuracy and speed by utilizing a programmatic approach that minimizes human error has clear business value. Bad data quality creates lousy business decisions, so improving QA accuracy is goal #1.

Establishing thresholds of acceptable and unacceptable deviations within the variables will help reduce errors due to human judgment. Goal #2 was to use the results of ML regression, clustering, and decision tree modeling on the binary, numerical, and categorial variables to establish the thresholds of acceptable anomalies and website errors.

Goal #3 was to have the prior day's anomaly report at the start of the day without spending human capital on manual QA. After this project demonstrates the proof of concept, the framework will utilize AWS for the daily aggregation of the data housed in S3 Athena SQL servers. The cloud-based approach will allow faster reactions and course corrections when things go wrong.

The future goal of full automation is outside the scope of this project. After this project proves the concepts, the plan will use AWS Lambda functions to trigger the AWS Glue job that would create full automation.

## Project Objectives

1. Improve QA's completeness, accuracy, and speed by utilizing a programmatic approach.
2. Develop a reusable workflow to source, clean, and analyze the product page’s clickstream data variables in preparation for modeling.
3. Establish alert thresholds for the subset of variables analyzed in this project.
4. Create a dashboard mockup to display the data in a way that the end stakeholders will understand and use.
5. Decide what clickstream variables to add to Acme Clothing’s database to improve analytics capabilities.

## Definition of Terms

* Web Analytics – “Adobe Analytics provides reporting, visualizations, and analysis of Customer Data that allows Customers to discover actionable insights.” (Adobe, 2024)
* Clickstream – digital tag records of user interactions on Acme Clothing’s websites.
* Data feeds – The raw clickstream data file transmission.
* Product Pages – Pages that display an individual product and allow the customer to add it to their shopping bag.

## Assumptions, Limitations, and Delimitations

The scope of that goal is immense. Only “product pages” will be assessed to comply with the timeline of this class. This subset of the company’s web pages represents approximately 45% of all page hits. The pages are homogenous in design and allow for a uniform analysis approach. There are approximately 56 other page types. IT updates/releases a new website, adding new page types and removing old ones.  These other page types are visited less frequently, and each is different in design. Each additional page, browser, and operating system type added to the analysis scope requires separate considerations.

## Conclusion

Chapter 1 provides an overview of this client-based project. It included background information about the purpose of the project, the goals of the project, and the deliverable objectives. Also included in Chapter 1 are the definitions of terms and assumptions. Chapter 2 will review the available past literature related to the project. It will explain the data science and machine learning methods already documented. Chapter 3 will describe the data, its preparation, and the methodologies used in this project. Chapter 4 will present the results of the chosen model. Finally, chapter 5 will summarize the project's findings and discuss the next steps.

# Chapter 2: Literature Review

## Introduction

This project aimed to take a unique perspective on using unsupervised machine learning algorithms for clickstream analysis. It proved challenging to find directly related literature. The literature focused on finding prominent themes to understand user behavior or cluster users into cohorts instead of pinpointing anomalies and outliers. For example, Huynh et al. (2020) proposed using pseudo-ID lists of each user’s clickstream events to efficiently aggregate the data for pattern mining within website browsing behavior. Another example is Requena et al. (2020) use of clickstream to predict website shopper purchase intent using Linear Regression, Random Forest, XG-Boost, and KNN classifiers. They found that KNN was the most accurate and efficient for the modeling. They also proposed expanding the predictions to include cart abandonment and retargeting marketing strategies.

This project aims to employ machine learning to automatically detect anomalies and errors within a single day’s clickstream data to save effort, time, and cost. The rest of this chapter will explain what clickstream is in a business context. It focuses on the cost-driving decisions an e-commerce retailer must make. The chapter will conclude with the machine learning algorithms explored to identify anomalies within e-commerce clickstream data.

## Choosing a Digital Analytics Vendor While Managing Seen and Unseen Costs

Since the Internet adoption by most of society in the late 90's and early 2000's, e-commerce has been growing. Companies have had to shift their operational strategies to stay competitive. Part of that shift included ensuring their websites perform as expected. Agile software development produces small and incremental website changes as needed in response to evolving business needs. Detailed web server log files are reviewed as a post hoc analysis to ensure the website performance in response to Agile development ("Agile software development," 2024). The Urchin Software Corporation was founded in 1995. It offered web hosting and services that automatically parsed web server log files to inform customers about their website performance. The company started adding JavaScript tags to aid in the available information in the web server logs. Google purchased Urchin Software Corporation in March of 2005. The purchase by Google was the birth of modern digital tag management and clickstream analysis for e-commerce websites (Dhruv, 2024).

Analyzing customer interactions on a website provides information about customer behaviors and ensures the website's functionality. Taking web analytics a step further was the opportunity to control and route user visits into separate website experiences. Controlling web user routes allowed companies to A/B test an experimental website build against the current standard website experience. A/B testing provides statistical validation of iterative website builds and the ability to test and learn with scientific rigor. This test-and-learn mindset was now the standard across e-commerce worldwide (Burke, 2009).

Companies can choose a web analytics vendor, such as Google Analytics or Adobe Analytics. Both have pros and cons. Cost is a primary consideration, but there are other considerations as well. The ability to customize to specific business needs was essential. Control of a company's customer data is important. There are legal and reputational repercussions when customers' personal and private information is not secured (Carre, 2018).

Google Analytics offers its entry-level product at no cost. However, the reporting and customization capabilities were minimal. Google offered paid options that had more advanced reporting and customizable capabilities. The cost is estimated to be between $50,000 and $150,000, depending on the company's needs. Google cannot A/B test a website for any of their options. A third-party contract with additional cost is required for A/B testing using Google Analytics. Not being able to perform A/B tests is a significant limitation of Google Analytics. Another important consideration before choosing Google is that its main revenue stream comes from digital marketing. A core use of web analytics data collection is to report and understand the company's digital marketing efforts and efficiencies. Google will appear as one of the most favorable marketing channel investments (Iyengar, 2022). While Google's entry product does not require a direct fee, the unseen costs to a corporation are significant. We only suggest Google Analytics for smaller companies that do not have the budget for advanced web analytics. Trading customers' data to Google for the ability to have some insight into a company's website does make good business sense for some companies.

Adobe is not a marketing company. They are not concerned with impacting what digital marketing efforts perform better for their clients. They only report what the digital tags record. We suspect that they are fairer in reporting digital marketing channel efforts. Adobe Analytics is also highly customizable and can suit just about any need a business has.  It does not come cheap, however. According to Ghosh (2023), "No, Adobe Analytics is not free. It's one of the most expensive analytics tools on the market and is mainly focused on enterprise customers. The prices range from $48,000 to $350,000 per year". Adobe Analytics' ability for customization and better control over the generated data are unseen costs. These unseen costs were essential when Acme Clothing used Adobe Analytics as their web analytics vendor.

In addition to the costs from the chosen vendor, both seen and unseen, there were internal costs of implementing and maintaining the digital tags regardless of the vendor chosen, both seen and unseen. Most large companies that have chosen a paid web analytics vendor have an internal IT department. Some companies had their IT department implement and maintain web analytics tags. Acme Clothing hired an implementation consulting firm to write the code for the digital tags. There was an annual consulting retainer agreement to maintain the validity of the digital tags as Acme Clothing’s IT department makes website updates. The cost of this implementation and the retainer can vary depending on the chosen firm. If Acme Clothing had implemented and maintained the tags with their internal IT, then that person's salary would have been considered as the cost seen. The unseen cost of using an external contractor was that they were not always available and had other clients to focus on while balancing their workload. While Acme Clothing used a contractor for the implementation and tag validity, they chose to have internal employees be subject matter experts to get business insights. The internal employee's job title was "Web Analyst". Web Analyst performed the QA of the digital tags. The Web Analyst reviews the digital tags manually. The contractor fixed the digital tags after Analyst found the issues.

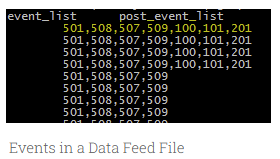
The main issue with this approach was the lag time to correction and the cost of human capital. Human capital was both a seen and unseen cost. The time and effort of the employee were clear to see. A human could not emulate every possible web browser and operating system. They could only click some web pages and events on some page types. In addition, there is a higher chance of human error when engaged in a technical task for an extended period. The unseen cost was the opportunity cost of the Web Analyst's time. The time could be spent informing and driving important business decisions instead of manually clicking through the website in search of bugs in the digital tags. Alternatively, alerts could be triggered to tell the analyst when it is essential to review—saving time and effort while avoiding taxing the analyst's attention span unnecessarily.

## Restatement of Goals and Research Needs

Chapter 1 stated the scope and complexity of the e-commerce website and the digital analytics tags used. There can be millions of rows or records. Each page can produce 500 or more separate variables and 400 possible events the digital tags assign (Exner, 2014). An example of the complexity of a single event list is seen below in image 1.

**Figure 1**

Example of a single post\_event\_list variable.



Only product-displaying pages will be considered in this project to comply with the project timeline. Acme Clothing collects approximately 100 variables and approximately 50 events for each product page to their AWS S3 storage for further analysis. For a customer purchase on an e-commerce website, the user must view a product displaying the page, add the product to their shopping bag, and go through the checkout process to place the order. For product displaying pages, the most essential variable for the business is understanding how the user navigated to the specific product page. Known as a “Product Finding Method,” the resulting “product view” event should fire when the customer views a product page. The shopping bag is its page type with the variable collecting cart additions. The checkout process has pages that record the purchase event if the customer converts to a purchase. This project focuses on the categorical variable of the “Product-finding Method” along with the page load times of each product page view. An alert can be triggered, allowing issues to be brought to the attention of the web analyst for further evaluation only when a particular product-finding method results in slower page load times.

Stated in Chapter 1 were the three main goals of the project. Goal #1 was to increase QA accuracy and speed; goal #2 was to use the results of ML modeling on the binary, numerical, and categorial variables to establish the thresholds of acceptable anomaly variance; and goal #3 was to have reported results at the start of the day without spending human capital on manual QA. The data science application of cloud-based computing was needed to solve goals 1 and 3 due to the sheer volume and complexity of the clickstream data. Goal #2 is where data science statistical and machine learning techniques were needed. Below are examples found from other industries and sciences research due to this project's unique nature.

## One Class Support Vector Machine (SVM)

In their effort to predict credit card fraud, Hejazi and Singh (2013) found that using a single class in support vector machine produced higher accuracy than two-classed examples. They describe, “The main idea of one-class SVM is that training data in input space are mapped into feature space via a kernel function and find a hyperplane with a maximum margin in feature space to separate the mapped data from the origin.” The above concept is extended in this project to use single-class SVM to identify anomalies within clickstream data.

## Isolation Forest (iF)

An isolation forest was an unsupervised machine learning algorithm focusing on difference instead of sameness. “Anomalies produce mean paths from the root to leaves that are longer than those for normal attributes. Trees are called isolation trees (iTs). Given a dataset, each iT is obtained by selecting a random subset of attributes and dividing it by randomly selecting a feature and splitting the branch until the node has only one instance. The iF defines an anomaly score, which is a quantitative index that defines an outlier’s degree of isolation. The anomaly score is defined for an observation q as given in Equation (7)” (Change et al, 2021). Essentially a decision tree on difference, the Isolation Forest was accessible to business users and very computationally performant.

## Local Outlier Factor (LOF)

LOF found anomalies by scoring each sample point's local density concerning the points in its surrounding k-nearest neighborhood. It compared each point's local density to the local densities of its neighbors. Grouping regions of similar density makes the points with the lowest density considered anomalous. "The anomaly score in LOF is known as the local outlier factor score; its denominator is the local density of a sample point, and its numerator is the average local density of the nearest neighbors of that sample point. LOF assumes that anomalies are more isolated than normal data points such that anomalies have a lower local density, or equivalently, a higher local outlier factor score" (Xu et al., 2019). Xu found that their method automatically approximated hyperparameters. "LOF uses two hyperparameters: neighborhood size and contamination. The contamination determines the proportion of the most isolated points (points that have the highest local outlier factor scores) to be predicted as anomalies" (Xu et al., 2019).

## Conclusion

This chapter explains the challenge of finding specific literature about clickstream data anomaly detection. Most of the literature found for clickstream analysis focused on user behavior and cohort clustering, while this project focused on individual variables and implementation correctness. For example, Sen (2020) used Random Forest and Support Vector Machine algorithms to predict user drop-off from a website. That project found that Random Forest could predict user drop-offs with 93% accuracy. The use of an Isolation Forest and One-Class Support Vector Machine in this project is due to Sen's findings. Another reason finding literature was difficult is that e-commerce websites significantly use web tags. One company may call a specific variable one naming convention and place it within a variable slot “A.” In contrast, another may call the same variable a different name and place it in variable slot “B.” There is very little standardization across different implementations.

Chapter 2 explains the business context of the overall requirements of a digital analytics implementation for Acme Clothing and, more generally, a modern e-commerce website. It covers businesses' choices and offers suggestions depending on their size and goal. Both seen and unseen costs are associated with a digital analytics implementation. The overarching goal of this project was to minimize costs and maximize business value by only employing human capital when needed. Chapter 3 will provide the projects definition of an anomaly, an overview of the data preparation, and an evaluation of the models. Chapter 4 will deploy the model on the population data and evaluate the results. Finally, chapter 5 will summarize the project's findings and discuss the next steps.

# Chapter 3: Methodology

## Introduction

Clickstream data is extensive in volume, velocity, variety, veracity, and value. Modern e-commerce websites generate many clickstream user data, allowing businesses to glean insights and achieve competitive advantages. Ensuring the accuracy of the data collected is important because bad data causes bad decision-making. This chapter describes the initial framework for monitoring and reporting anomalies within the clickstream data collected. Human analysts will then explore these anomalies to decide if further intervention is needed to remedy the error. Described below are the data collection, preparation, and model selection. Also explained is feature selection and rational. Finally, sample data compared the accuracy of algorithm predictions. The evaluation of the results in Chapter 4 will then use the best model for the total population.

## Overview of Methodology with Multi-Dimensional Anomalies

Clickstream data is messy and confusing. There can be thousands of categories within each of the variables. The digital tag values change with each website release iteration. New web browser versions are released, and each version can perform differently on each new website release. Using standard statistical techniques on this shifting baseline population and variable set is limited. This project aims to find a more robust anomaly and outlier detection approach. This project sought to create a flexible and generalized framework to cluster and detect anomalies for the large variety of measurements possible in a digital analytics implementation. Identifying what is anomalous in a large and diverse data set is challenging. Anomalies within clickstream need to consider more than a single dimension. Due to multiple dimensional spaces, traditional statistics using general guidelines, such as standard deviation and interquartile range bounds, won't be enough to detect actual anomalies. Clickstream's diverse and changing nature requires non-traditional methods to automatically produce reliable anomaly detection within a feature space. This project employed various unsupervised and semi supervised machine-learning approaches to wrangle the data and find meaningful anomalies.

## Data collection

The data used in the project came from the Athena SQL engine reading data from an S3 bucket. The original data source was a nightly batch process where Adobe Analytics sends a compressed .csv file of the prior day’s clickstream hits. The ETL process ingests the .csv file to Athena, ensuring the data within a column is uniform. The goal was to report on a full day’s worth of clickstream. The arbitrary date chosen was October 1st, 2024. The total population of the original data set was 825,008.

Variable selection was essential to the project’s success. Most available variables would not indicate an anomaly unless missing when it should not be. Some variables will help humans know where to look for anomalies, even if they do not help predict them by themselves. Ultimately, 11 variables were selected: four categorical and seven numerical.

Categorical variables:

1. “post\_evar3” is a string representing the way the user used to navigate to the product page.
2. “browser\_type” is a numeric representation of a browser’s type, i.e., Google.
3. “brows\_os” is a string operating system name and version.
4. “page\_event\_var2” is a string indicating if there were exit links.

Numerical variables:

1. “missing\_prod\_view” is a binary flag indicating if the hit missed the product view event.
   1. This is a critical error that a human would review.
   2. The figure 2 below shows very small counts of true missing product views.

**Figure 2**

Distribution of the missing\_prod\_view variable.

A blue bar graph with numbers

Description automatically generated

1. “page\_error” is a binary flag indicating if the hit had an error recorded on the hit.
   1. This is a critical error that a human would review.
   2. The figure 3 below shows very small counts of true page errors.

**Figure 3**

Distribution of the page\_error variable.

A blue bar graph with numbers

Description automatically generated

1. “excluded\_hit” is a binary flag indicating if that hit should be excluded from reporting.
   1. This is a critical error that a human would review.
   2. The figure 4 below shows very small counts of true excluded hits.

**Figure 4**

Distribution of the excluded\_hit variable.

A blue bar graph with black text

Description automatically generated

1. “page\_load\_time” is the time in milliseconds that it took the page to load for the hit.
   1. When exceeding a threshold, this becomes a critical error that a human would review.
   2. The figure 5 below shows the highly skewed distribution of page load times.

**Figure 5**

Distribution of the page\_load\_time variable.

A blue bar graph with white text

Description automatically generated

1. “visit\_num” is an integer that indicates what visit number the user is on.
   1. It is not a critical signal, but often, first-time visits see more errors than others.
   2. The figure 6 below shows the highly skewed distribution of visit numbers.

**Figure 6**

Distribution of the visit\_num variable.

A graph with a bar

Description automatically generated

1. “hit\_time\_gmt” is a Unix timestamp of the time of the hit.
   1. The figure 7 below shows the distribution of hit times on the Unix timestamp scale.

**Figure 7**

Distribution of the hit\_time\_gmt variable.

A graph with numbers and a number on it

Description automatically generated

1. “error\_factor” is a combination of weighted values of missing\_prod\_view + page\_error + excluded\_hit + page\_load\_time.
   1. This is not used to predict anomalies in the final modeling. Instead, it identifies the actual anomalies to score the models accuracy against each other.

## Data preparation

All records had the data due to the Athena SQL source. As seen in the variable histograms above, most of the numeric variables were highly skewed. There was also much variation in the magnitude of values. For example, the hit time Unix time stamp’s maximum value was approximately 1.7 billion while the page load time’s minimum value was 9. Data-science-based algorithms that use distance calculations in decision-making require a standardization of the data in relation to each other to avoid the biased impact of naturally larger-scaled variables over smaller-scaled variables.

On the other hand, tree-based algorithms, such as Random Forest, are not impacted by scaling bias between variables. Standardization of the data makes no difference in the accuracy of these types of models. Since the final algorithm was not yet determined, all data were scaled and standardized to each other using sklearn.preprocessing.StandardScaler() before loading into the unsupervised and semi-supervised ML models.

Unfortunately for the project, very few anomalies existed in the total population's data set. 10,000 records of heavily anomalous data were added to the population data set to simulate the expected performance in the real world. 835,008 is the new total for the population. Since the project only explored unsupervised ML algorithms, no test/train split was required. Instead, highly anomalous records were identified prior to modeling and the models were scored on their ability to accurately identify the anomalies.

Deciding the subset of the categorical variables available relied heavily on expert knowledge of the subject matter professional. The variable “post\_evar3” was explored and found to be unhelpful in pinpointing anomalies. Dropping this variable improved the final model accuracy assessment. The brows\_os variable had much variation and was very sparse in data. It had 151 types, with the majority being very low counts. In addition, 44% of the records had “None” as the brows\_os, meaning it is not very descriptive of underlying patterns. Removing it before modeling improved accuracy. The variable page\_event\_var2 is also unhelpful, with only 480 records with a value other than “None.” Before the model selection process, removing the variable from the data set was crucial. Some web browser types are known to produce more errors than others. Browser type was the only categorical variable used in the final model selection. Figure 8 below shows sampled counts of error\_factor by browser\_type. The naturally occurring anomalies have an error\_factor value of less than 400. The simulated and heavily anomalous records are seen in the “None” browser type with values of 999 on the far right of the plot.

**Figure 8**

Error factor by Browser Type

A graph with different colored dots

Description automatically generated

## Generate and evaluate models

This project focused on evaluating unsupervised machine learning because the goal was to provide a generalized framework for anomaly detection. The research in Chapter 2 showed that three primary methods are used for this type of detection: One-Class Support Vector Machines, Local Outlier Factors, and Isolation Forests. This project utilized Scikit-learn library 2.7. This project compares eight simulations using different hyperparameters to assess the impact of the detection accuracy. The F1-Score produced by each simulation was the metric used to compare the models. The F1-score is defined as:

**Figure 9**

F1-Score Definition.

A close-up of words

Description automatically generated

(Xu, 2019)

One Class Support Vector Machines require standardized data. It was, however, found to have limitations that almost immediately excluded it from consideration. It only allows two numeric variables, but the goal was to utilize the six error variables. These six variables produce anomalies within clickstream data. It was also computationally expensive. It was not possible to model the entire population. The method had a maximum of 50,000 records. The F1-scores produced from the sample set were consistently around 0.5. Accuracy was considerably worse than any of the other two models. The model is inappropriate in this context and excluded from the final selection.

Local Outlier Factors require standardized data. It was also able to handle all six error variables. LOF was very accurate on tests using a sample of 50,000. It produced the best accuracy on the samples, with tuned F1 scores reaching 0.89. The optimized hyperparameter was n\_neighbors=2500. It was, however, exceptionally computationally expensive. Many runs using the entire population caused the kernel to crash and required manual intervention to complete. LOF is excluded from the final consideration due to the above.

Isolation Forest is a decision tree type that does not require standardized data. Having normalized data does not negatively impact prediction accuracy. We normalized the data in comparison testing but not in the final modeling. Isolation Forest was also able to handle all 6 of the error variables. It was also the most computationally performant. It could always handle the total population while completing within approximately 5 minutes of run time. The optimized hyperparameter was n\_estimators=300 while testing sample data. The best F1-Score in the population was 0.82. While this is lower than the best F1-Score found with LOF, the computational performance gains were worth the trade-off in prediction precision. Isolation Forest was the chosen winner for the project’s deployment.

## Conclusion

Chapter 3 explains the steps taken in the project to prepare unsupervised machine learning models for detecting anomalies within a clickstream data set. The data is from the company's AWS Athena databases. Further data cleaning was needed. Subject matter experts identified features of importance. Split into a training and test set was optional due to the unsupervised nature of the models employed. The average F1-scores for each method assed the winning model. Computational performance became an issue as the data set grew from testing samples to the entire population data set. Chapter 4 will compare the results of the selected model across time and with different levels of anomaly contamination to assess the method's robustness in real-world scenarios. Finally, chapter 5 will summarize the project's findings and discuss the next steps.

# Chapter 4: Results

## Introduction

Clickstream data can include every aspect of an e-commerce website. Anomalies occur in many different forms and many different magnitudes of signal. An example of a large-scale anomaly with a substantial signal magnitude is when a company’s backend order collection system goes offline. Customers cannot place orders online, and online sales drop to zero. These customers will either drop their order and go to a competitor or call in to place their order. The former is much more common than the latter. Sales would be lost, and the signal strength of this anomaly would be seen and felt across the organization. An example of a minor scale anomaly with less signal strength would be if a new website release caused a lousy shopping experience for only a small subset of users and their shopping devices. These signals can go unfelt when the release only impacts a small user base on an outdated device type. Losing sales is avoided by bringing minor anomalies to the company’s attention.

Each facet of the business has different focuses and KPIs to monitor. The digital marketing team focuses on customer acquisition and conversion rates for each channel they manage. Email is a digital marketing channel focusing on email opens, specific link clicks, customer acquisition, and conversion rates. The business operations team has completely different KPIs and focuses. Monitoring anomalies can help that team plan and optimize the limited company resources they manage. For example, if there is a spike in demand for customized products, they could bolster the people and other resources needed to avoid disrupting customer expectations. This project aimed to narrow focus to a single type of anomaly and use it as a proof of concept for automatic detection and alerts.

## Hyper Parameter Tuning and Results Evaluation

The research reviewed in Chapter 2 focused on unsupervised machine learning methods—these methods we explored because of their flexibility and broad application possibilities. According to Sebastian et al. (2024), unsupervised machine learning methods do not require extensive training data or even data labels yet can still accurately predict. While researching for Chapter 2, it became clear that most research applications of unsupervised machine learning revolved around finding large clusters and similarities within the data sets. An example of this is Gallaugher and McNicholas's (2024) research of using first-order continuous-time Markov models to cluster groups of website users by clustering user behaviors with clickstream data to find groups that browse the sites in similar fashions. Another example is Moe and Fader’s (2004) research measuring user buying propensity based on browsing behavior on an e-commerce website.

This project aimed to use reverse methods to find anomalous needles in the haystack of non-anomalous data instead of identifying considerable similarities and grouping cohorts into clusters. This project aimed to make identification automatic. To test how well the chosen Isolation Forest algorithm performed without any supervision vs how well it performed with minimal supervision. The main hyperparameter for the Isolation Forest algorithm is the contamination parameter. It is the expected factor of outliers for the entire data set. The contamination parameter has two options. One option is automatic, where no additional input is needed, and the data tells the algorithm how to proceed. The second option was manually setting it with a value between 0 and 0.5 (Scikit-learn developers). For each of the days tested, there was approximately 0.06 contamination. After adding the strongly signaled and artificially simulated outliers into the populations, this value grew from 1.6% to 6.9% of the total population for each day’s simulation. As seen below in Figure 10, the Auto contamination parameter’s F1-score (blue below) consistently underperformed against the manually set and observed contamination (orange below). Interestingly, the Auto contamination parameter setting’s F1 score improved significantly as the total contamination of outliers increased.

**Figure 10**

F1 Scores by Hyper Parameter Tuning and Increasing Contamination Percent.

Chapter 3 explained the six numeric variables that produced a website error or anomaly during a user’s session. True anomalies in the data set were identified by compositing these six numeric variables into a single error factor. The Algorithm's training did not include the composite error factor. If the error factor is > 0 for a given record, it was indeed an anomaly. Specificity is the actual negative rate. Sensitivity is the actual positive rate. Specificity and sensitivity scored the results.

Interestingly, any of the model's iterations identify all strongly signaled and artificially simulated outliers. The error factor composite variable identified actual and false positive outliers. Figure 11 below shows what the model identified as an outlier when the contamination parameter was set to the actual outlier percent. The points identified as anomalous outliers are red dots, while the inliers are blue. The red band of dots seen in the upper left-hand corner are the strongly signaled and artificially simulated outliers added to the daily populations before modeling.

**Figure 11**

Model Predicted Inliers vs Outliers of error factor by Unix Hit Time

A red and blue dotted chart

Description automatically generated

Figures 12 and 13 below compare the results with the model’s contamination parameter set to Auto vs. set manually based on the expected outlier percentage seen in the populations. Figure 12 below overstates identified outliers and is less sensitive to the nuances within the data set itself. The manually set model performed better as seen in figure 13. Figure 13 below does not overstate the identified outliers.

**Figure 12**

Contamination Parameter Set to Auto

**Figure 13**

Contamination Parameter Set by Population Outlier %

While the utterly unsupervised model using the Auto contamination parameter did predict outliers, it needed to be more precise. On average, this model produced 5 times more outliers than what was in the populations. Setting the contamination parameter to the actual outlier percent within the population avoids these false positives. The strongly signaled and artificially simulated outliers were held at 10,000 records from 10/1/24 through 10/6/24. It was increased to 20,000 records on 10/7/24, then reduced back to 10,000 records on 10/8/24. 100,000 of these artificial outliers were added on the last day, 10/9/24, causing both graphs above to reflect the spike in anomalies.

## Conclusion

Chapter 4 compared the Isolation Forest’s ability to accurately predict outliers/anomalies within clickstream data sets for the first 9 days of October 2024. While the utterly unsupervised version accurately predicted true outliers, it produced many more false positives. Adding some supervision and knowledge about the data sets significantly reduced the instances of false positives, proving the use of machine learning algorithms to aid in discovering QA problems within clickstream. False positives may not be harmful in the context of QA. It would simply cause the analyst to review more instances manually. It still allows automation and takes some of the analyst's manual workload. Chapter 5 will be the concluding chapter, which summarizes the project in totality and discusses limitations and other applications of the methods for other anomaly types.

# Chapter 5: Discussion

## Introduction

Clickstream data is extremely valuable to e-commerce businesses, and its value is growing. According to the report by Market Research Future, “the market [clickstream] is predicted to thrive substantially during the assessment era from 2021 to 2030 at a healthy CAGR of approximately 11% to attain a valuation of around USD 1.3 Billion by the end of 2030” (2023). The value of clickstream is directly related to the company’s ability to extract insights. According to Chen et al. (2024), clickstream is the source of insights for user/shopper analytics, first-party customer data platforms (CDP), and digital marketing analytics. Shopper analytics is helpful for businesses in understanding customer preferences and engagement levels. A first-party CDP allows businesses to combine historical offline customer data with online data. Enabling personalization, optimized journeys, and targeted messaging. Digital marketing analytics details users’ interactions with digital marketing campaigns. Facilitating the ability to measure campaign effectiveness and optimization and ultimately improve conversion rates and the company’s bottom line.

Ensuring the quality of clickstream data is essential to the success of the use cases described above. Sizing the cost expenditures to acquire quality clickstream data helped frame the importance of quality assurance for Acme Clothing. Minimizing seen and unseen cost expenditures was a priority for Acme Clothing and the justification for this project.  This project aimed to take a small bite from the clickstream QA process and automate it using data science techniques. This project focused on finding anomalies and errors, each with different signal strengths, and within full days’ clickstream data for product-displaying pages only. Of the three machine learning methods attempted, only one was appropriate for deployment into a production environment.

## Summary of Findings

Chapter 2 explained that prior literature for the specific task of anomaly detection in clickstream was minimal. This project explores three modeling methods because they are used in similar anomaly detection efforts in other industries, such as credit card fraud detection. Due to the unpredictable nature of anomalies, unsupervised and semi-supervised machine learning algorithms were employed for the project.

This project focused on finding anomalies only on product-displaying pages. An anomaly for these page types was defined as a clickstream user's hit missing a product view event, producing a page error event, producing an exclude hit error event, having an unusually long page load time, or having an unusual visit number. An outlier in any of these variables qualifies the hit as an anomaly. Some clickstream records had more than one of the variables producing a stronger anomalous signal than others. Nine days of clickstream data of these variables were assessed by two machine learning algorithms, one utterly unsupervised and one semi-supervised. The algorithms separated the inlier non-anomalies from the outlier anomalies analyzed daily. Fortunately for Acme Clothing, but unfortunately for this project, few natural anomalies were found. Additional strongly signaled records were simulated and injected into the natural data set to prove the concepts in this project.

Interestingly, both the utterly unsupervised and the more refined semi-supervised machine learning algorithms could identify all the strongly signaled simulated records in all runs. Also interesting is that both algorithms correctly identified the non-anomalous inliers with greater than 99% accuracy in all runs. The completely unsupervised machine learning algorithm did predict false positive anomalies more often than the semi-supervised algorithm, producing 12 – 17 times more false positive anomalies. Adding some level of supervision to the algorithm improved the predictive accuracy. The average F1 score of the utterly unsupervised algorithm was 0.3. The average F1 score of the semi-supervised algorithm was considerably better at 0.83.

After identifying the anomalies, a threshold of 0.01 was established based on historical and expected future values. When the threshold is crossed, a programmatic alert is triggered, and an email with the alert is sent to the human analysts. The human analysts will review the anomalies and use their judgment to decide if additional follow-up is needed to correct the errors. Figure 14 below shows the dashboard with the alert threshold set to 0.01 anomaly contamination.

**Figure 14**

Alert Threshold of Anomaly Contamination by Browser Type

## Summary of results organized by project objectives

The analysis in this project is a proof of concept for automating QA efforts on a retail website. It focused on a tiny part of the overall QA needs. Recall from Chapter 1 that there were five objectives for this project. The first objective of this project, using a programmatic approach, was to improve QA's completeness, accuracy, and speed. The analysis was produced using Python Jupyter notebooks. The singular proof of concept is shown through the automatic anomaly detection of product page errors and long-running page load times. The second objective was to develop a reusable workflow to source, clean, and analyze clickstream data variables in preparation for modeling. The framework provided by the Jupyter notebooks will be reused to assess other QA efforts in the future. Alert emails for each automated QA effort will be sent, and when needed, the human analyst will assess and follow up.

The third objective was to establish the alert threshold. The threshold for this specific anomaly detection task on product pages is 0.01. When anomalies are detected above the threshold, an alert is sent for further review. The fourth objective was to create a dashboard mockup to display the data in a way that the end stakeholders would understand and use. Figure 14 above shows an example of this dashboard. It is easy for stakeholders to understand that when anomaly counts exceed the red line threshold, it is cause for review.

The fifth objective was to decide what additional clickstream variables to add to Acme Clothing’s database to improve analytics capabilities. All implementations of Adobe Analytics on a website automatically come with 224 out-of-the-box variables (Adobe, 2024). Five variables were identified and added to the current collection for better analysis. Two of the variables identified are beneficial for marketing channel tracking. Three of these variables identify a hit as containing a duplicate page, purchase ID, and form-filled event. Identifying these duplicate events was critical for Acme Clothing’s real-time data reporting; otherwise, overcounting the events would make the real-time reporting misleading.

## Discussion of implications for business and suggestions for Future Research

Managing the effort and costs to wrangle clickstreams quality assurance is not unique to Acme Clothing. An entire industry has been emerging to aid companies with their QA of clickstream data. ObservePoint is an example. They are a company that will map an entire organization’s website and scan it using various proprietary tools to ensure clickstream data quality. “Tools from ObservePoint save businesses time by not requiring Web analysts to manually audit digital tags on each page of a website to ensure the information is compiled correctly.” (ObservePoint, 2013). Amazon also has a service to create an anomaly detection pipeline using Amazon Kinesis (Marshall, 2016). A web search will show that many consulting firms have made clickstream auditing their primary business.

This project has shown that the quality assurance of web analytics digital tagging is costly in both time and money. We show that some of these costs can be avoided by utilizing automation and a programmatic approach. Instead of the human analysis needed to review data daily, they only need to review when the alert is delivered. This project focuses only on a small subset of QA needs. The concept can be extended to all other page types and QA situations.

For example, a human analyst currently compares the total revenue seen by Adobe Analytics against the backend order system for each day. It was understood that Adobe Analytics is not a perfect capture of every order. Some users have ad blockers on their devices. Because of this, approximately 5-10% of total website revenue was expected to be added to Adobe Analytics. If the missing revenue grows above the 10% threshold, an email alert will be sent directing the human analyst to look deeper into the order collection on Adobe Analytics. This example saves the human 10 to 15 minutes out of their day. While this may seem trivial, the compounded impact of automating all possible QA is significant.

While anomalies are not predictable from historical reviews because of their unpredictable nature, other QA efforts are predictable based on the past. Standard statistical measures on these variables with past observations can be used to trigger the email alert. For example, A chi-square goodness of fit test on the proportions of revenue attributed to each product-finding method is predictable for past customer behavior. When one of these product-finding methods shrinks or grows beyond expectations, the email alert will trigger the human analyst to review and act if needed manually. Cost-saving is realized by freeing up human capital and employing the programmatic solution. Future work to chip away at what can be automated regarding the QA of clickstream, as well as clickstream utilization in general, is outlined below:

* **Extend Automated Anomaly detection to all Page Types** – extend the proof of concept shown in the project to include all page types on Acme Clothing’s website.
* **Automate Revenue Validation with backend order system** – an easy automation addition that will free up 10 to 15 minutes of daily manual effort. Alerts will be triggered only when the discrepancy grows above the tolerated 5-10% missing revenue in the clickstream compared to the backend order entry system.
* **Measure Product Finding Method Proportions and Alert When Changed** – using historical proportions of revenue by product finding method to assess when significant changes occur and alert analysts for review.
* **Join offline Marketing Data with Online Data for Holistic Reporting** – offline data such as digital marketing spend by campaign and targeted customer cohorts isn’t readily available within Adobe Analytics. By joining the online signal data from Adobe Analytics with offline business measures, more accurate and holistic reporting dashboards can provide real-time insights. An example of this digital marketing dashboarding is found below in Figure 15.

**Figure 15**

Marketing Channel Performance Dashboard

A screenshot of a computer dashboard

Description automatically generated

## Conclusion

This project is a proof-of-concept showing that anomalies within clickstream can be automatically detected utilizing a programmatic approach backed by data science techniques, and alerts can be sent to a human analyst for further assessment when warranted. The semi-supervised algorithm outperformed the utterly unsupervised algorithm for automatically detecting clickstream anomalies.  All client objectives were delivered. This project proved that a programmatic approach to the quality assurance of the digital tags on an e-commerce website can supplement human efforts and free up human capital for more important work. The QA tasks that are quantifiably definable can be automated to free up time otherwise spent doing repetitive tasks that do not produce value or insight. Time can now be spent on higher-level analysis that provides business insights instead of the repetitive auditing of digital tags. The concept can be extended to other business areas, such as page type and product finding method performance, ensuring order collection audits with the backend order entry systems, and optimizing digital marketing campaigns.

# References:

Adobe. (2024, September). Data column reference. Retrieved Nov, 2024, from

<https://experienceleague.adobe.com/en/docs/analytics/export/analytics-data-feed/data-feed-contents/datafeeds-reference>

Adobe. (2024, May). *What is Adobe Analytics?*. Retrieved Oct, 2024, from

<https://helpx.adobe.com/legal/product-descriptions/adobe-analytics.html>

Agile software development. (2024, October 6). In *Wikipedia*.

<https://en.wikipedia.org/wiki/Agile_software_development>

Burke, K. (2009). Testing, Testing, 1, 2, 3. *Target Marketing*, *32*(11), 19-.

Carre et al. (2018). Ascribing responsibility for online security and data breaches. *Managerial Auditing*

*Journal*, *33*(4), 436–446. <https://doi.org/10.1108/MAJ-11-2017-1693>

Chan et al. (2024, November). *Clickstream Analytics on AWS*. aws.amazon. Retrieved Nov, 2024, from

<https://docs.aws.amazon.com/pdfs/solutions/latest/clickstream-analytics-on-aws/clickstream-analytics-on-aws.pdf>

Change et al. (2021). Anomaly Detection Using Signal Segmentation and One-Class Classification in

Diffusion Process of Semiconductor Manufacturing. *Sensors*, (Basel, Switzerland), 21(11), 3880-. <https://doi.org/10.3390/s21113880>

Click tracking. (2024, June 11). In *Wikipedia*.

<https://en.wikipedia.org/wiki/Click_tracking>

Click path. (2024, June 11). In *Wikipedia*.

<https://en.wikipedia.org/wiki/Click_path>

Clickstream Analytics Market Revenue To Surpass USD 1.3 Billion at a 11% CAGR By 2030 - Report by

Market Research Future (MRFR): Increasing adoption in social media to boost clickstream

analytics market growth. (2023). In *NASDAQ OMX’s News Release Distribution Channel*. NASDAQ

OMX Corporate Solutions, Inc.

Cowie, K. (2022, March). *Anomaly Detection in Clickstream Analytics*. Medium.

<https://blog.eqworks.io/anomaly-detection-in-clickstream-analytics-f49dc82b0579>

Dhruv. (2024, May 8). *A Journey through the History of Google Analytics*. eyeuniversal.com.

<https://www.eyeuniversal.com/blog/seo/history-of-google-analytics/>

Exner, J. (2014, September 2). *Using Data Feeds for Debugging*. webanalyticsfordevelopers.com.

<https://webanalyticsfordevelopers.com/2014/09/02/using-data-feeds-for-debugging/>

Gallaugher, M. P. B., & McNicholas, P. D. (2024). Clustering and semi‐supervised classification for

clickstream data via mixture models. *Canadian Journal of Statistics*, 52(3), 678–695.

<https://doi.org/10.1002/cjs.11795>

Ghosh, A. (2023, August 15). *Adobe Analytics Pricing and a Better Alternative*. Simpleanalytics.

[https://www.simpleanalytics.com/resources/analytics-pricing/adobe-analytics-pricing-and-a- better-alternative](https://www.simpleanalytics.com/resources/analytics-pricing/adobe-analytics-pricing-and-a-%20better-alternative)

Hejazi, M., & Singh, Y. P. (2013). ONE-CLASS SUPPORT VECTOR MACHINES APPROACH TO ANOMALY

DETECTION. *Applied Artificial Intelligence*, 27(5), 351–366.

<https://doi.org/10.1080/08839514.2013.785791>

Hochenbaum et al. (2017, April). *Automatic Anomaly Detection in the Cloud Via Statistical Learning*.

Twitter Inc. <https://arxiv.org/pdf/1704.07706>

Huynh et al. (2020). Efficient algorithms for mining clickstream patterns using pseudo-ID Lists. *Future*

*Generation Computer Systems*, 107, 18–30. <https://doi.org/10.1016/j.future.2020.01.034>

Iyengar, N. (2022, August 30). *The GA4 360 Pricing Model, Explained*. Cardinalpath.

<https://www.cardinalpath.com/blog/ua-360-vs-ga4-360-pricing-model>

Marshall, C. (2016, September 20). *Real-time Clickstream Anomaly Detection with Amazon Kinesis*

*Analytics*. aws.amazon. Retrieved Nov, 2024, from <https://aws.amazon.com/blogs/big-data/real-time-clickstream-anomaly-detection-with-amazon-kinesis-analytics/>

Moe, W., & Fader, P. (2004). Capturing evolving visit behavior in clickstream data. *Journal of*

*Interactive Marketing*, *18*(1), 5–19. <https://doi.org/10.1002/dir.10074>

Nasraoui, O., Rojas, C., & Cardona, C. (2006). A framework for mining evolving trends in Web data

streams using dynamic learning and retrospective validation. *Computer Networks (1999)*, *50*(10), 1488–1512. <https://doi.org/10.1016/j.comnet.2005.10.021>

ObservePoint. (2021, September 23). *3 Ways To Debug Adobe Analytics* [Video]. YouTube.

<https://www.youtube.com/watch?v=b4Tkuh82FNc>

ObservePoint Partnering with Optimal Ways. (2013). *Entertainment Close-Up*.

Regression testing. (2024, May 24). In *Wikipedia*.

<https://en.wikipedia.org/wiki/Regression_testing>

Requena et al. (2020). Shopper intent prediction from clickstream e-commerce data with minimal

browsing information. *Scientific Reports*, 10(1), 16983–16983.

<https://doi.org/10.1038/s41598-020-73622-y>

Sebastian et al. (2024, March 4). Unsupervised Machine Learning Approaches for

Test Suite Reduction. *Applied Artificial Intelligence*, *38*(1). <https://doi.org/10.1080/08839514.2024.2322336>

Sen, A. (2020, July 17). *Clickstream Pattern Analysis and Prediction using Machine Learning [GHC ‘19]*.

Medium. <https://medium.com/@ananya1sen/clickstream-pattern-analysis-prediction-using-machine-learning-ghc-19-63eeb1e3931f>

The scikit-learn developers. IsolationForest. *scikit-learn.org*. Retrieved Nov, 2024, from

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html>

Xu et al. (2019, February 5). Automatic Hyperparameter Tuning Method for Local Outlier Factor, with

Applications to Anomaly Detection.  *arXiv.Org*, <https://doi.org/10.48550/arxiv.1902.00567>

# APPENDIX

Link to code: <https://github.com/ascie150/DS785-Final-Project/tree/main>