




MAEVE: An Agnostic Dataset Generator Framework for Predicting Customer Behavior in Digital Marketing

William Ferreira da Silva Filho^{1,2}^a, Seyed Jamal Haddadi^{1,2,3}^b and Julio Cesar dos Reis^{1,2}^c

¹Hub de Inteligência Artificial e Arquiteturas Cognitivas (H.IAAC), Campinas, Brazil

²Artificial Intelligence Laboratory (Recod.ai), Campinas, Brazil

³Instituto de Computação, Universidade Estadual de Campinas (UNICAMP), Campinas, Brazil

Keywords: Customer Behavior Prediction, Digital Marketing, Event Logging, Dataset Generation.

Abstract: Data analysis plays a crucial role in assessing the effectiveness of business strategies. In Digital Marketing, analytical tools predominantly rely on traffic data and trend analysis, focusing on user behaviors and interactions. This study introduces a dataset generation framework to assist marketing professionals in conducting micro-level analyses of individual user responses to digital marketing strategies. The implemented proof of concept demonstrates that the framework can be integrated with enterprise software monitoring applications to ingest logs and, through appropriate configuration, generate comprehensive and valuable datasets. This research centers on the application of the framework for predicting customer behavior. The evaluation examines the extent to which the generated datasets are suitable for training various machine learning (ML) algorithms. The framework has shown promise in producing machine learning-ready datasets that accurately represent complex real-world scenarios.

1 INTRODUCTION

In the contemporary digital landscape, data is essential for business success and the validation of scientific theories. More specifically, large amounts of data can be refined into datasets, which, in turn, enable data-driven decision-making. What those datasets look like and how to create them depends on their source and platform (Renear et al., 2010).


Digital data sources include websites, mobile applications, and plugins. Data may also come from other sources, such as the Internet of Things, the financial market, or health care. Data can be applied across various domains, including agricultural monitoring, home surveillance, and the management of automotive or office environments. It may improve your user-targeted marketing or user experience. Data can enhance investment strategies and support the early detection of diseases. Data takes all kinds of shapes, coming from different systems in various industries. It is crucial to have access to a large amount of data to enable data-driven decision-making, analyze it to


understand how data elements correlate to a question that needs answering, and plot it accordingly - creating a dataset.


Gathering data from different platforms infers different methodologies due to the differences in communication protocols, programming languages, data format, etc., making it challenging to create unique software to gather data from all of them. Data collection from a system requires the integration of specialized capabilities within the software or the development of auxiliary software dedicated solely to extracting system output.

This scenario drove the analytics market to build enterprise applications to monitor systems by capturing their data (for instance, Datadog (Datadog Documentation Authors, 2024) or Google Analytics). Their primary purpose is not to acquire data but to monitor and report specific system characteristics.

Many current analytics applications use macro-scale analyses because they provide metrics based on large-scale traffic data instead of user interaction. Monitoring applications such as Datadog enable micro-scale analysis by ingesting logs resulting from user interaction with graphical interfaces. Literature has shown platform-specific software to accumulate data (de Santana and Baranauskas, 2015) (Ma et al.,

^a <https://orcid.org/0009-0003-1763-1562>

^b <https://orcid.org/0000-0001-6022-4143>

^c <https://orcid.org/0000-0002-9545-2098>

2013) (Froehlich et al., 2007)(Rawassizadeh et al., 2013) (Pielot et al., 2014) (Ferreira et al., 2014). On a broader scale, enterprise applications accomplished much on data collection with platform agnosticism.

This article proposes MAEVE as a dataset generator framework. This study aims to answer the following research question: Can enterprise monitoring applications be leveraged to solve platform-agnosticism in dataset generation? Our study further investigates a novel ecosystem responsible for generating datasets based on API communication with those enterprise monitoring applications to generate datasets - assuming the enterprise application supports its platform. Such an ecosystem can help marketing strategies thrive with data-driven decision-making by providing a “plug-and-play” dataset generation framework for any software.

This research explores machine learning algorithms to experimentally analyze the framework’s results and the quality of the datasets as our metric of success. We demonstrate how the datasets generated by the framework are performed using different machine learning algorithms.

This study provides the following contributions:

- The conception and development of a novel framework “MAEVE” for micro-scale insights on user responses to digital marketing strategies.
- Integration with enterprise software monitoring applications as platform agnostic data sources to ingest logs and generate extensive and analyzable datasets.
- Demonstration of the dataset’s readiness for machine learning algorithms, reflecting real-world complexities and showcasing how the novel framework can enable data-driven decision-making.

The remainder of this article is organized as follows: Section 2 reviews the related works that laid the foundation for this research. Section 3 outlines the MAEVE framework proposal in detail, explaining its components and functionalities. Section 4 describes the experimental methodology used to evaluate MAEVE’s effectiveness, along with the results obtained. Section 5 discusses the key findings and highlights the open challenges in the field. Finally, Section 6 presents the concluding remarks and summarizes the contributions of this study.

2 RELATED WORK

Applied data science can be achieved mainly in two ways. The first is to use enterprise software directed

to your needs, such as Google Analytics. Such tools provide you with data gathering solutions to improve business, such as access traffic to your website, peak access timelines, most used pages, etc. The second one is more specific and less friendly for people with no technology background, which is to build your data pipelines.

The objective of the MAEVE framework is to generate datasets useful to the user’s (the “user” being the person who benefits from the framework’s dataset) specific needs in digital marketing. This allows users to configure the framework to generate datasets that facilitate predictions regarding client return rates, purchasing likelihood, and other key behaviors.

In practical terms, the user must identify and specify the location of the desired outcome within the system’s data (For instance, a log entry that records the precise moment a user interacts with the purchase button). Additionally, the user should define the interface interactions or characteristics that exhibit a correlation with that outcome. This approach enables the framework to identify relevant patterns and correlations in the data, thereby supporting predictive analytics.

Thus, MAEVE delivers an analytics solutions that for Digital Marketing professionals, empowering their decision-making with few configurations.

WELFIT (de Santana and Baranauskas, 2015) and Xiaoxiao Ma *et al.* (Ma et al., 2013) represent models of user-triggered event recorders. Aligning with the architectural principles outlined in both studies, logs are imported to establish independent modules. Moreover, the prevailing approach in mobile event logging, as seen in MyExperience (Froehlich et al., 2007), predominantly relies on operating system APIs to collect sensor and OS-specific events. However, such data does not align with the focus of this research, which prioritizes user interaction with application interfaces as the primary dataset for analysis.

As per software monitoring applications, a standard functionality exists in log management that allows the storage of logs from diverse platforms. Platforms like Datadog (Datadog Documentation Authors, 2024) showcase a promising solution, with distinct SDKs for various platforms converging into a unified log management system. This facilitates multi-platform event logging by being the platform itself, an event logger, and establishing a robust foundation for comprehensive monitoring purposes, particularly with Datadog’s (Datadog Documentation Authors, 2024) unique incorporation of Real User Monitoring (RUM) capabilities.

Moreover, our research underscores the need for an open-source ETL framework to generate digital

marketing datasets across multiple platforms. While existing market ETL applications suffice for dataset generation, their platform-specific nature limits the research's objective (Informatica Power Center, 2024) (Talend Documentation Authors, 2024) (Microsoft, 2024).

Our proposed framework aims to fill this gap by being versatile, agnostic to data types, and exclusively utilizing NoSQL databases to align with its objectives. In summary, while WELFIT (de Santana and Baranauskas, 2015) and Xiaoxiao Ma *et al.* (Ma *et al.*, 2013) highlight cutting-edge logging capabilities, the versatility, and comprehensiveness of enterprise monitoring applications like Datadog (Datadog Documentation Authors, 2024) position them as optimal choices for event logging in the proposed framework, which emphasizes multi-platform readiness and NoSQL compatibility.

The novelty in our proposed framework comes from creating an ecosystem that, joining the technologies presented above, can be used to create marketing-directed datasets on a platform-agnostic basis. The monitoring application brings state of the art in providing an event logger and a centralized, platform-agnostic log management system.

By creating MAEVE's modules based on API communications with monitoring applications, we abstract the implementation of that system, meaning that the event logger can be Datadog, Grafana - any application with an API for log ingestion. MAEVE's modules serve as the ETL that ingests logs, the products of which are the datasets. With this framework, with minimum knowledge of the data and minimum configuration, MAEVE allows a fast way of generating almost real-time datasets based on user interactivity with graphical interfaces.

3 MAEVE DATA GENERATOR FRAMEWORK

This research objective is to contribute to the context of digital marketing, by enabling the creation of datasets on a platform-agnostic basis, enabling data-driven decision making. The proposal is to create a dataset generation framework composed of three main modules: the log importer; the data normalizer; and the dataset generator. We named this framework as "MAEVE", which stands for Marketing Event Logging and Dataset Generation Framework. Figure 1 presents the relationship between those three modules. The following sections describes the chosen event logger and each component of the framework and its responsibilities.

3.1 Datadog: The Chosen Event Logger

For the implementation of the event logger abstraction, Datadog (Datadog Documentation Authors, 2024) was selected. Datadog is a robust software monitoring application designed to serve as a comprehensive platform for engineers to manage logs, and create alerts based on system performance, abnormal behavior, and more.

Several factors contributed to the decision to choose Datadog. Firstly, it offers a user-friendly experience with minimal setup requirements. Installation merely involves integrating its SDK into the system to be monitored and configuring a simple API and Application key. Once configured, Datadog seamlessly aggregates application logs.

Secondly, Datadog's log entries extend beyond simple text content. They encompass a wealth of contextual information such as the user's browsing activity, interacted elements, time zone, browser details, user session information, and more, providing invaluable insights.

Lastly, Datadog stands out from its competitors like Grafana and Google Analytics due to its extensive API. While many tools confine logs within their own ecosystems to promote platform lock-in, Datadog offers a well-documented API empowering users to access virtually any data sent to the platform.

It is important to notice that by choosing Datadog we leverage the state of the art from monitoring applications. Datadog is not conceptually an event logger. Its purpose is to monitor applications. However, to do so, it needs to import its logs, becoming, by extension, an event logger. We harvest that functionality from Datadog to create a platform-agnostic dataset generation framework, since Datadog provides SDK for several platforms, such as Android, iOS, Windows, Linux, etc. Thus, by connecting MAEVE with Datadog, we can generate datasets from a wide range of applications.

3.2 Importer

A pivotal aspect of this module is its abstraction. Leveraging the capabilities of an object-oriented language, we define interfaces and abstract classes that can be implemented to establish connections between the importer and any event logger. This design approach allows for flexibility in integration, as the event logger is not constrained to providing a specific API. Instead, it could be an event stream, files, a database, or any other data source.

The importer operates as a job-based application specifically tailored for API-based event loggers. This

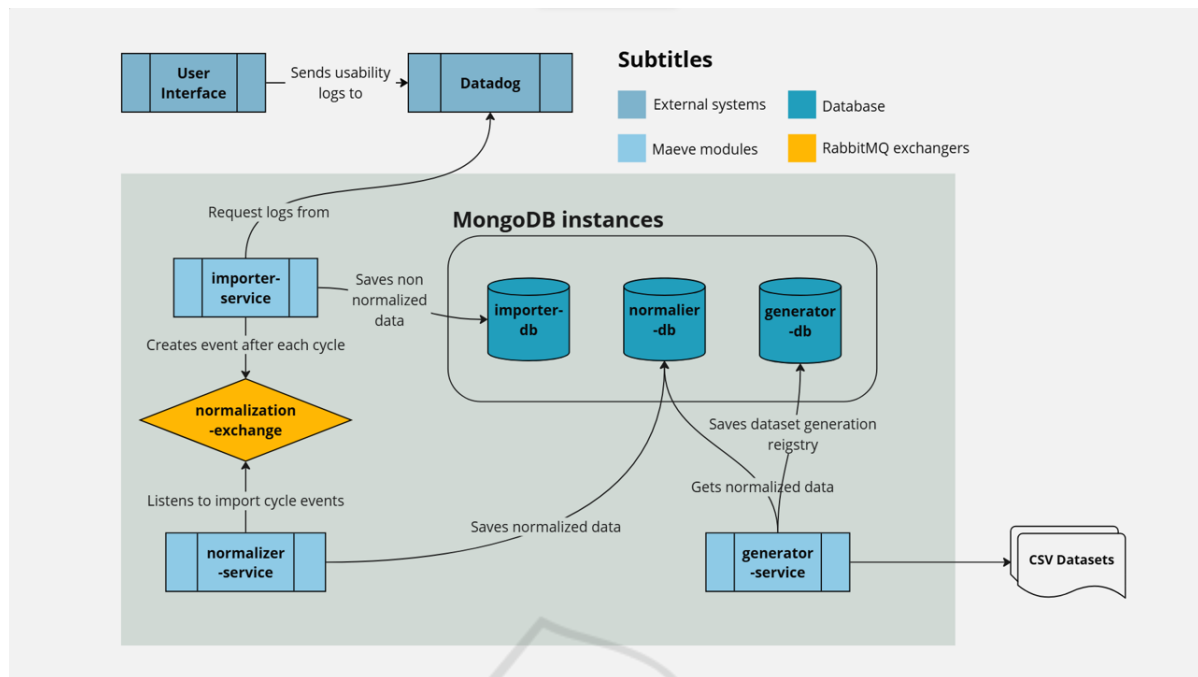


Figure 1: Component diagram of MAEVE.

design is essential due to the lack of real-time access to incoming logs. As a result, the importer employs its own CRON scheduled jobs. During each job execution, the importer queries Datadog (the event logger we have chosen for this experimentation) for all logs generated since the last job run up to the current timestamp. Each retrieved log from Datadog is then stored in its raw form within a document-oriented NoSQL database. If Datadog were to function as an event stream rather than an API, the need for scheduled jobs would be obviated, as we could subscribe to the stream and listen for incoming logs. However, given the versatility and applicability of job-based imports across various communication channels, this approach was chosen.

Another crucial aspect of the importer is its role as the trigger for log normalization. With the logs securely within MAEVE’s ecosystem, they become available for manipulation as needed. Additionally, the importer facilitates communication with other modules by employing an event-driven architecture implemented with RabbitMQ.

Upon successfully saving a log in the database, the importer dispatches a message to a RabbitMQ exchange containing the ID of the log. This notification signals to other modules that a new log is ready for transformation.

3.3 Normalizer

The normalizer module subscribes to the message exchange, to which the importer sends messages to. This is done to achieve an event-driven architecture (Cassandras, 2014), which is essential for the framework to be scalable. Upon receiving a message, the normalizer consumes it, initiating the normalization process for the log identified by the message’s ID.

Normalization entails two configurable steps: a) determining relevant fields that will become features in the datasets and b) formatting field values. The first step can be configured within the module’s settings, allowing users to define the desired structure for the log. The normalizer then removes unnecessary fields and retains only those designated for persistence.

The second configuration involves coding, employing a factory design pattern (Shvets, 2018) to implement a set of normalization rules. These rules are implemented as classes to clean and transform field values within the logs. For instance, rules could anonymize personal user data or standardize timestamps to a specific time zone. The factory design pattern ensures that all configured rules are applied to the log during normalization.

It is important to notice that this factory design pattern is crucial in MAEVE’s architecture. Leveraging the abstraction capabilities provided by Java, this design pattern allows the normalizer to be the heart of MAEVE. This module allows developers to create

essentially any rules to deal with the logs being ingested, making it a data engineering silver bullet.

After passing the two layers of data normalization, the normalized log is then saved in the normalizer's document NoSQL database instance. These logs represent the final form of the data and serve as the basis for dataset creation by the next module.

3.4 Generator

The generator module serves as the engine that leverages the storytelling capabilities inherent in normalized logs to address specific questions. In the context of this research, the question the generator is trying to answer through datasets is: will the user buy a product?

This module can read a normalized database, treat and organize the data based on its configuration. The configuration is a simple map of where the necessary fields can be found, and what field is the dataset meant to predict.

The generator's output is a CSV file in which each line refers to a normalized log, each column is a feature, and the final column is the binary response of what is being analyzed.

Utilizing abstraction and the factory design pattern, the generator empowers developers to create implementations of dataset generation in various formats. For this paper, CSV file implementation was chosen to facilitate experimentation. Bayesian prediction (Kruschke, 2014) with Python and the pandas framework (pandas Development Team, 2024), known to work well with CSV format, will be employed for testing the datasets.

3.5 External System and Usability Logs Generation

To evaluate the framework, it is necessary to find a suitable graphical interface system into which Datadog can be installed. To do so, keeping in mind the intention of generating marketing directed datasets, we used an open source marketplace UI.

MAEVE's inputs are logs, and to generate logs we need users interacting with the system's interface. For the generation of the logs, we mimic user interactivity using Cypress, a front-end integration testing framework, to create bots. Those bots are essentially automated tests that interact with the UI in a configured manner. The bots were configured to operate in three different behaviors: an assertive user to buy, a user that is not interested in the product and does not buy and a user that is frustrated by the UI and leaves the website.

4 EXPERIMENTAL EVALUATION

This experimental evaluation aims to apply five machine learning techniques to the created dataset to evaluate how the dataset generator framework is appropriate for predicting customer behavior in digital marketing.

4.1 Dataset

The dataset used in this experimentation comprises 15,000 rows and 25 features. The features can be seen on table 1. The features represent four aspects of a user session:

- **User Personal Data:** personal information on the user, such as gender, age, and for how long the user account has been active
- **Historical User Activity (Sessions):** features that show if the user was logged in during the session, the average amount of active sessions last month, etc.
- **Product Data:** Product rating, price, and important information that might lead to the purchase, such as is the product currently in the user's favorites list?
- **UI Interactivity Logs:** Most of the features are categorized as interactivity data, and they represent actions the user takes on the UI during the session, such as: did the user access the wallet, scrolls through the product, was there any frustration recognized from the usability pattern, how much time did the user spend on the product page?

The label distribution is divided into two distinct categories. The larger category, comprising 62.8% of the total, represents instances labeled as "Purchased." In contrast, the smaller category, accounting for 37.2% of the data, corresponds to entries labeled as "Not Purchased." This distribution is visually represented in a pie chart, highlighting the proportional difference between the two segments.

The next section details the machine learning techniques applied to this dataset.

4.2 Machine Learning Techniques

Since one of the research questions in this study is to predict whether a customer purchases a product, we used machine learning models to conduct this binary classification prediction. To this end, given the problem is a supervised learning problem, four classical methods, and a deep neural network model are chosen to solve this binary classification problem.

Table 1: The 25 features in the dataset generated by MAEVE.

Feature Name
session_id
logged
gender
age
home_page_access
product_accessed
is_product_favorite
product_view_time
home_page_rum_frustration_count
product_page_rum_count
item_in_cart
wallet_page_access
has_payment_method_registered
wallet_page_rum_frustration_count
sign_up_page_access
payment_page_rum_frustration_count
product_rating
product_price
tenure
num_sessions_last_month
total_spent_last_month
avg_time_on_product_pages_seconds
session_duration_minutes
product_page_scroll_depth
purchased

Support Vector Classifier (SVC). The Support Vector Machine (SVM) Classifier, or SVC, was incorporated due to its ability to identify the optimal hyper-plane within a transformed feature space, thereby segregating classes by the widest margin possible (Cortes and Vapnik, 1995). SVM's utility in managing imbalanced datasets is underscored through its focus on maximizing margins while being minimally affected by the presence of majority classes. The implementation of SVC here involves both linear and Radial Basis Function (RBF) kernels.

KNeighbors (KNN). This non-parametric and lazy learning algorithm, classifies objects by aggregating the majority votes from their k closest neighbors within the feature space (Cover and Hart, 1967). To identify the optimal neighborhood size, the performance of the KNN algorithm was assessed using various k values (1, 3, 5, 7, and 9).

Gaussian Naive Bayes. Naive Bayes classifiers encompass a suite of algorithms for classification grounded in Bayes' Theorem. For continuous data, the Gaussian Naive Bayes approach posits that the feature values associated with each class follow a Gaussian distribution (Kamel et al., 2019).

Logistic Regression. This provides an approach

for analyzing qualitative dependent variables that are categorical instead of continuous. It addresses the constraints of least squares regression in scenarios with binary or categorical outcomes by calculating the likelihood of particular events (De Menezes et al., 2017).

Deep Neural Network. Deep learning falls under machine learning techniques that utilize artificial neural networks to learn representations. A Fully Connected Feedforward Neural Network (FCNN) is utilized for binary classification tasks and is specially designed for such purposes. This network type is often known as a Dense Neural Network or, more generally, a Deep Neural Network (DNN) when it features multiple hidden layers.

4.3 Procedures

4.3.1 Data Labeling

To implement binary classification, the dataset needs to be labeled. For this reason, a criterion is defined to label the data which the mathematical formulation of the revised labeling criterion can be expressed as:

$$\text{Label} = \begin{cases} 1, & \text{if } (E \vee Q) \wedge (H \vee P) \\ 0, & \text{otherwise} \end{cases}$$

where:

$$E = (V > 50) \vee (S > 90)$$

$$Q = (D > 45) \vee (F < 5)$$

$$H = (T > 100) \vee (N > 20)$$

$$P = (R > 5) \vee (C < 600)$$

Here, E , Q , H , and P represent conditions related to engagement, session quality, historical behavior, and product factors, respectively. The logical operators \vee and \wedge denote the logical OR and logical AND operations, respectively.

4.3.2 Data Preprocessing

Data Preprocessing encompasses loading the dataset, inspecting its structure, and splitting it into training (%70), validation (%20), and test sets (%10). Subsequently, categorical variables are encoded, while numerical features are standardized. This ensures uniform scaling across features. Finally, the dataset is prepared for model training, ensuring integrity and optimal utilization for subsequent optimization steps.

4.3.3 Model Creation and Fine-Tuning

In this phase, a neural network model with varying hyperparameters and other traditional machine

Table 2: Model Evaluation Metrics and Correct/Incorrect Classified Instances for Machine Learning Methods used.

Model	Recall	F1-score	Precision	Accuracy	Correctly/Incorrectly Classified Instances (%)
KNN 1	0.70	0.70	0.70	0.70	69.50 30.50
KNN 3	0.73	0.73	0.73	0.73	73.43 26.57
KNN 5	0.76	0.75	0.75	0.75	75.50 25.50
KNN 7	0.76	0.75	0.76	0.76	76.13 23.87
KNN 9	0.77	0.75	0.76	0.76	76.53 23.47
GaussianNB	0.82	0.82	0.85	0.82	82.00 18.00
SVC_RBF	0.85	0.85	0.85	0.85	85.33 14.67
SVC_Linear	0.82	0.82	0.82	0.82	81.57 14.43
LR	0.80	0.80	0.80	0.80	80.20 19.80
DNN	0.88	0.88	0.88	0.88	87.78 12.22

learning models is dynamically generated. To optimize these models, Optuna (Akiba et al., 2019), an open-source automated hyperparameter optimization framework, provides a flexible and efficient platform. It automates the hyperparameter search process, leveraging advanced techniques such as Bayesian optimization to identify the optimal configurations.

4.4 Results

Table 2 presents the obtained results. Among the models analyzed, including KNN with different neighbors, GaussianNB, SVC with RBF and linear kernels, Logistic Regression (LR), and Deep Neural Networks (DNN), the DNN model achieved superior performance with the highest Recall, F1-score, Precision, and Accuracy at 0.88. It also exhibited the best classification performance, with 87.78% of instances correctly classified and an error rate of 12.22%. This indicates a significant advantage of deep learning techniques in handling complex classification tasks, highlighting their potential in predictive analytics and pattern recognition within diverse datasets.

5 DISCUSSION

This study addresses the challenge of designing and implementing an automated solution for generating high-quality datasets from user interaction logs across diverse platforms. Such datasets are essential for enabling precise, data-driven decision-making in digital marketing. The MAEVE dataset generator framework was developed as a flexible and scalable software tool, designed to transform raw, unstructured log data into structured datasets that accurately reflect real-world complexities. By capturing granular details of user interactions and behavior, MAEVE facilitates the extraction of actionable insights, enhancing

the effectiveness of digital marketing strategies.

The framework presents several notable strengths. Firstly, fidelity, ensuring that the datasets generated preserve the integrity of the original data, maintaining the nuances and complexities necessary for robust analysis. Second, feature richness is achieved through the inclusion of a wide range of interaction features, thereby enhancing the efficacy of machine learning models in predicting outcomes. Third, customizability, allows users to adapt dataset generation processes to specific investigative needs or marketing contexts. Finally, scalability, ensures the framework's seamless integration with various machine learning methodologies, enabling efficient processing of large-scale datasets.

The experimental evaluation confirmed that the datasets produced by MAEVE are well-suited for training machine learning models, yielding high accuracy in predicting customer behavior. This underscores MAEVE's utility as an effective tool for data-driven research and digital marketing analytics. Its core attributes—fidelity, feature richness, customizability, and scalability—position the framework as an effective solution for generating realistic and machine learning-ready datasets applicable across diverse digital marketing environments.

6 CONCLUSION

Given the differences in capturing user interaction logs from different applications, platform-agnostic dataset generation is difficult to achieve. Moreover, given the platform differences, those data might look distinct and might not be normalized. In this scenario, predicting customer behavior based on graphical interface interactivity is costly for any new digital marketing endeavor. This study proposed a framework, MAEVE, that uses an abstraction for enterprise monitoring applications and created ETL modules for logs

ingestion and normalization to, finally, generate digital marketing-directed datasets. Our framework enables (a) generate digital marketing-directed datasets based on user interactivity with graphical interfaces; and (b) to be platform agnostic, meaning that the same framework can be used to generate datasets for mobile, web, embedded applications, etc. Our solution contributes to the literature on predicting customer behavior while providing a technical approach that enables marketing experts and data scientists to have a quick start on their endeavors.

ACKNOWLEDGEMENTS

This work was supported by the Brazilian Ministry of Science, Technology and Innovations, with resources from Law n° 8,248, of October 23, 1991, within the scope of PPI-SOFTEX, coordinated by Softex and published Arquitetura Cognitiva (Phase 3), DOU 01245.003479/2024 -10. This work is also supported by the 'PIND/FAEPEX - "Programa de Incentivo a Novos Docentes da Unicamp" (#2560/23).

REFERENCES

- Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.
- Cassandras, C. G. (2014). The event-driven paradigm for control, communication and optimization. *Journal of Control and Decision*, 1(1):3–17.
- Cortes, C. and Vapnik, V. (1995). *Support-vector networks*, volume 20. Springer.
- Cover, T. M. and Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.
- Datadog Documentation Authors (2024). Datadog logs documentation. Online; accessed March 20, 2024.
- De Menezes, F. S., Liska, G. R., Cirillo, M. A., and Vivanco, M. J. (2017). *Data classification with binary response through the Boosting algorithm and logistic regression*, volume 69. Elsevier.
- de Santana, V. F. and Baranauskas, M. C. C. (2015). Welfit: A remote evaluation tool for identifying web usage patterns through client-side logging. *International Journal of Human-Computer Studies*, 76:40–49.
- Ferreira, D., Goncalves, J., Kostakos, V., Barkhuus, L., and Dey, A. K. (2014). Contextual experience sampling of mobile application micro-usage. *MobileHCI 2014 - Proceedings of the 16th ACM International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 91–100. Cited By :99.
- Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., and Landay, J. A. (2007). Myexperience: A system for in situ tracing and capturing of user feedback on mobile phones. *Association for Computing Machinery*, page 57–70.
- Informatica Power Center (2024). Informatica powercenter. Online; accessed March 20, 2024.
- Kamel, H., Abdulah, D., and Al-Tuwaijari, J. M. (2019). *Cancer classification using gaussian naive bayes algorithm*. International Engineering Conference (IEC).
- Kruschke, J. K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan*. Academic Press; 2nd Revised ed, 2nd edition.
- Ma, X., Yan, B., Chen, G., Zhang, C., Huang, K., Drury, J., and Wang, L. (2013). Design and implementation of a toolkit for usability testing of mobile apps. *Mobile Networks and Applications*, 18(1):81–97. Cited By :26.
- Microsoft (2024). Sql server integration services (ssis). Online.
- pandas Development Team (2024). pandas documentation. Online; accessed March 20, 2024.
- Pielot, M., Church, K., and De Oliveira, R. (2014). An in-situ study of mobile phone notifications. *MobileHCI 2014 - Proceedings of the 16th ACM International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 233–242. Cited By :219.
- Rawassizadeh, R., Tomitsch, M., Wac, K., and Tjoa, A. M. (2013). Ubiqlog: A generic mobile phone-based life-log framework. *Personal and Ubiquitous Computing*, 17(4):621–637. Cited By :75.
- Renear, A. H., Sacchi, S., and Wickett, K. M. (2010). Definitions of dataset in the scientific and technical literature. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–4.
- Shvets, A. (2018). *Dive Into Design Patterns*, volume 1. Refactoring.Guru.
- Talend Documentation Authors (2024). Talend data integration. Online; accessed March 20, 2024.