A Work Project, presented as part of the requirements for the Award of a Master's degree in Business Analytics from the Nova School of Business and Economics.

Identifying User Groups: A Machine Learning Framework for Classifying Job Roles Based on Clickstream Data

William Esary

Work project carried out under the supervision of:

Qiwei Han

Bruno Silva

Miguel Almas

15/12/2023

Abstract

Clickstream data, the digital footprint of a user's online browsing activity, offers a unique window into an individual's interests and intentions. This work showcases a machine learning framework designed to classify website visitors by job function using features crafted from 6 months' worth of server-side clickstream data. These predicted job functions can be used to send targeted communications to end users of a low-code B2B service in order to boost engagement. The study finds success at differentiating developers, the key users, from other job roles based on their use of the company website.

Keywords: User Segmentation; End Users; Machine Learning; Targeted Advertising; OutSystems; Clickstream Data; Web Behavior

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

Introduction

The global B2B (Business to Business) market was valued at over 7 trillion USD in 2022 with an expected 18% compounded annual growth rate through the year 2030 (Vantage Market Research 2022). The booming industry has attracted much attention from researchers examining the B2B buying process, specifically the interaction between the supplier and the key decision-makers of the buying firm. One group often left out of this research is the end user. End users are the people actually using the product in question on a day-to-day basis without necessarily playing a role in its purchasing (Vivek 2012). Many B2B companies operate under a recurring revenue model, relying on customer firms to continue their subscriptions to stay afloat. As end users are the ones using the product, keeping them engaged is crucial to ensuring this happens (Amy Greiner Fehl 2023). To take steps to boost engagement among end users, it is first crucial to get an idea of who they are.

In the era where data is the new currency, understanding user behavior online has become a cornerstone for businesses to get to know their customer base. Clickstream data, the electronic record of a user's journey across a website, offers a wealth of information that can provide a profound window into a user's intent and preferences (Randolph E. Bucklin 2002). When analyzed effectively, it can yield insights that allow for effective user targeting, capitalizing on discovered interests. Clickstreams have been used widely as a tool for predicting customer actions (will a browsing session lead to a purchase conversion), but their use in grouping users into well-defined demographic segments like job roles has been minimal. B2B products are designed to be used in the workplace, so knowledge of an end user's job role could significantly boost the effectiveness of promotional content for features pertaining to a specific function.

This thesis, conducted on behalf of OutSystems, utilizes clickstream data to classify

visitors of the OutSystems website according to their job function, with the specific intent of recognizing developers. The website holds content for both stakeholders with purchasing power and end users, acting as a one-stop shop for all parties involved, aligning a user's website usage with their usage of the product. Knowledge of a user's job function will not only give OutSystems a better idea of who is using their product, who is the end user, but will also provide the opportunity for personalized promotions to help retain and boost engagement. Users identified as developers can receive information tailored to development.

All steps for the classification are presented from the selection criteria of studied users, data preprocessing, feature engineering, model selection, to the results. Sections are broken down as follows: Literature Review, Data, Data Preprocessing and Feature Engineering, Methodology, Experimental Setting, Results and Conclusion.

OutSystems

OutSystems provides a low-code development platform offering businesses a means to create, deploy, and oversee omnichannel applications for enterprise use. Recognized as an industry leader by both Gartner and Forester, the company has clients across 22 industries, including retail, pharmaceuticals, and banking (OutSystems 2023). This study will focus on the OutSystems website, which, in addition to information on pricing and purchasing, houses training content, documentation, certifications, a trial portal for cloud development, and a community hub for asking and answering questions about OutSystems development. While OutSystems is predominately sold as a service to businesses, the vast adoption of the platform has created a market for OutSystems professionals. Website visitors include potential customers, existing clients, and people trying to learn how to use the platform to upskill.

Literature Review

End User Engagement

Despite not being actively involved in the purchasing process, the end user's opinion

still weighs heavily in the procurement and continued procurement of B2B products (Amy Greiner Fehl 2023). End users can act as internal advocates for the product provided it serves their needs effectively. Recent research has highlighted that a deep understanding of end-user preferences is critical for creating valuable offerings (Homburg 2020). Homburg et al. conducted several interviews with corporate executives of global firms, with many highlighting the importance of looking to the end user for direction on product improvements based on their needs. They propose the adoption of an 'end user priority' suggesting that B2B firms turn away from the traditional approach of nurturing customer relationships with senior employees and turn their marketing focus towards creating value for end users, citing successes of this approach from 2014 onwards.

Demographics-Based Personalization

To get in touch with the end user it is first required to get their attention. Demographic-based marketing in a broad context allows businesses to concentrate on specific consumer behavior groups, leading to higher return on investment in marketing and sales campaigns (Martin 2011). Prior to the online era, a marketing campaign's performance could only be measured indirectly through increasing revenues (De Bock 2010). With the advent of the web, it has become easier to observe a consumer's interaction with promotions using measures such as click-through rates (H. Robinson 2005). It has been found that advertisements tailored to specific demographic characteristics of their viewers significantly increase engagement, suggesting demographic targeting is an effective way of getting a consumer's attention. While demographic campaigns commonly focus on gender and age characteristics, job roles have been found to foster a strong sense of a group identity that can be used for targeting (Amy Greiner Fehl 2023).

Though the value of obtaining demographics is clear, collecting the information can be problematic in the anonymous world of the web. Website operators are left with two choices if

they want to capitalize on the value they hold: user registration with self-filled fields or purchasing the data from a third-party provider (De Bock 2010). User registrations come with their own drawbacks. They can be a barrier to entry to consumers, turning away users who would have otherwise continued browsing. They also have an inherent self-selection bias because users define their own characteristics. Purchasing extensive data from a third party, however, remains expensive and will most likely not cover the entire user base.

Demographics can be used for a wide variety of personalization. In the B2B case knowing an end user's job function can allow a firm to interact with them on matters surrounding their work.

Clickstream Data

Clickstream data represents the electronic record of an individual's activity on the internet (Sismeiro 2009). The term commonly describes the sequence of HTTP requests made to a website (Gang Wang 2017). These requests imply the explicit interaction of a user with page, such as clicking a link or button or other related actions. Each request or event is attributed to a specific cookie ID. A user's clickstream effectively acts as a view of their choices as they navigate a website. Assuming users value their time and click with intention, these choices provide valuable insights into their interests. Such insights not only provide a more in-depth view into the behaviors of customers whom the company already knows (previously identified groups of users) but, with the advent of machine learning, clickstreams can serve as a means of classifying unknown users into meaningful segments.

Clickstream Classification Research

The classification and prediction of user actions using server-side clickstream data has garnered significant attention from researchers. Studies surrounding clickstreams typically fall under two categories: content mining and usage mining. Content mining strives to uncover actionable insights by considering text content within a user's visited pages as well as within

URLs themselves. Usage mining examines what pages a user visits, often in what order, and what actions, if available they took there.

Usage mining has been used to address a wide range of classification problems, with researchers utilizing both supervised and unsupervised approaches. Supervised learning models are trained using labeled datasets to predict specific outcomes. This approach is particularly suitable for situations when the desired class distinctions are known, and information on a subset of data exists that can be used to train a model accordingly. Perhaps the most heavily researched application of clickstream data has been its potential for modeling purchasing outcomes within the e-commerce industry. One such study conducted by Borja Requena et al. examines user intent using minimal clickstream data from an e-commerce platform, demonstrating the viability of both hand-crafted features fed through an XGBoost classifier and an LTSM deep learning model applied to event sequences at predicting purchase outcomes (Borja Requena 2020). This study highlighted the potential for early state prediction, needing as little as five click events to make accurate predictions.

While classifying users as buyers or window shoppers showcases clickstream data's prediction power, we are more interested in its ability to distinguish demographic groups. In another application of usage mining, De Bock and Vand den Poel make use of cross-site clickstream data to predict demographic profiles of website visitors for targeted online advertising (De Bock 2010). Their method revolves around converting clickstream patterns into an aggregate feature set containing variables like page visit frequency and day of the week visit numbers. They utilize a Random Forest classifier to predict gender, age, education level, and occupational categories. The study sees reasonable performance in both the binary gender classification and multi-class demographic predictions, falling off slightly for out-of-period data. The challenge of predicting gender and age groups was also taken on by Hu et al, who employed a discriminative model to associate webpages with demographic characteristics

which were then used to predict visitors' demographic information through a Bayesian framework (Jian Hu 2007). Hu et al highlight the sparsity of clickstream data, implementing a smoothing technique that enhances the model's ability to generalize observed behavior.

Clickstream data has also been used in circumstances where labels were not readily available in the quantities needed to train a supervised model. Wang et al. preform a study collecting clickstream data containing the sequence of clicks and time gaps between events from multiple Chinese social media sites with the goal of isolating the behavior of malicious users (Gang Wang 2017). Weighted graphs were built to represent similarity distances between sequences of different users. Clustering was then applied to these graphs to group users with similar behaviors under the hypothesis that malicious accounts would exhibit distinct behaviors. The approach was later successfully validated using known malicious accounts, once again affirming the power of browsing patterns as an identification tool.

The idea that content can be used for prediction was thoroughly explored by Brian Davidson, who made use of the HTML content of pages and the headers of requests and responses to predict a user's future actions based on text similarity (Davison 2002). The study finds that a similarity ranking was moderately accurate at predicting a set of the following possible pages in a user's browsing session. Similarly, Liu et al. propose a method to extract topics from URLs, making use of natural language processing to create a document-word frequency matrix that is transformed into a probability matrix using a Latent Dirichlet Allocation algorithm (Hongri Liu 2021). Multi-class classifiers were then trained using this matrix to classify user behavior into categories. While content mining provides an easy-to-grasp classification method, it suffers from scalability issues (Yoon Ho Cho 2002) . Larger websites house massive amounts of text content that would require serious computational resources to process and analyze. The text on these pages is also frequently updated, making it difficult for a model trained in one instance to remain relevant over time.

Data

Server-Side Records

This work analyzes a user's interaction with the OutSystems website to classify them according to their perceived job function. To accomplish this, server-side clickstream data was obtained. OutSystems records server-side events of various types. For the sake of this work, only *page* and *track* type events are considered. Page events record page views and can be seen as a user's browsing history. Track type events record a user's actions on a page such as button clicks, form submissions, downloads, and page closes among others. A complete list of possible activities can be found in the Appendix (A1). Each event is associated with a cookie identifier (from here on referred to as *Anonymous ID*), timestamp, URL, and user identifier (*User ID*) if the user has registered. Cookies are small text files stored on a device via a web browser. They are commonly used to store unique identifiers to recognize individuals (SalingerPrivacy 2020). User identifiers refer to the email associated with the Anonymous ID behind an event provided a user has made an account. Examples of both page and track type events can be found below.

ANONYMOUS_ID	USER_ID	TIMESTAMP	URL	TYPE	ACTIVITY
ABCD	userA@email.com	2023-03-02 15:54:01.405125	https://www.outsystems.com/forge/	track	click
DEFG	userB@email.com	2023-03-02 15:54:01.405126	https://www.outsystems.com/low-code-platform/application-development/	page	NaN

Example of Page and Track Type Events

Six months' worth of server-side clickstream data was collected from March 1st to August 30th of the same year. This amounted to 22,826,754 events across 69,514 users. Of these, 11,030,574 were track type events and 11,796,180 were page types. As this classification was conducted with the aim of engaging with users through targeted emails based on their predicted job role, only users who signed up for an account are considered. A user has signed up if they have a non-null value for the User ID field at some point over the period. To avoid starting the analysis from the middle of a user's journey, only users who recorded their first action within the period, both in terms of Anonymous ID and User ID, were included.

Job Titles

Job titles were collected from the company's CRM system (Customer Relationship Management). Our selection of website visitors includes both existing customers who pay (or work for a company that pays) for the product and potential customers utilizing the trial platform. As job titles will serve as the labels used for modeling, only users who have signed up with an email associated with a job title in the CRM are examined. A distribution of job titles can be found in the appendix (A2).

Data Preprocessing and Feature Engineering

User ID Mapping

Server-side events are recorded before and after a user has made an account. A link must be made between each Anonymous ID and their associated User ID to ensure the entirety of a user's journey is analyzed over the chosen period. This process is met with several challenges. The first is that cookie values are not set in stone but depend on the specific browser being used. Users can access the site from a different browser or clear their cookies, resulting in different Anonymous IDs, making it hard to recognize repeat visitors based on Anonymous IDs alone (Stephanie Flosi 2013). A second potential issue is that if a user has more than one account and uses the same browser for each, all accounts will be associated with the same Anonymous ID. A third rare case found within the data involves a shared computer in which a single Anonymous ID is associated with multiple accounts belonging to different users, making it impossible to distinguish between visitors.

To deal with these complications, a single universal identifier, *UNIVERSAL_ID*, was established by linking Anonymous IDs to their most frequently associated User ID. This identifier was then applied to Anonymous IDs before and after signing up to track users across the entirety of their site usage. It was determined that an Anonymous ID with two associated accounts could very well represent one individual's personal and business accounts, therefore

implying the same job role. Anonymous IDs associated with three or more accounts were not considered.

ANONYMOUS_ID	USER_ID	UNIVERSAL_ID	TIMESTAMP	URL	TYPE
facebcda-a0d5-4637-9c88-a350f10b792a	NaN	userA@email.com	2023-05-13 18:05:17.941000+00:00	https://www.outsystems.com/Platform/Signup	track
facebcda-a0d5-4637-9c88-a350f10b792a	NaN	userA@email.com	2023-05-13 18:05:20.626000+00:00	https://www.outsystems.com/Platform/Signup	track
facebcda-a0d5-4637-9c88-a350f10b792a	NaN	userA@email.com	2023-05-13 18:05:22.431000+00:00	https://www.outsystems.com/Platform/Signup	track
facebcda-a0d5-4637-9c88-a350f10b792a	userA@email.com	userA@email.com	2023-05-13 18:05:23.082000+00:00	https://www.outsystems.com/Platform/Signup	track
facebcda-a0d5-4637-9c88-a350f10b792a	userA@email.com	userA@email.com	2023-05-13 18:05:45.276000+00:00	https://www.outsystems.com/Portal/Trial_Portal	page
facebcda-a0d5-4637-9c88-a350f10b792a	userA@email.com	userA@email.com	2023-05-13 18:05:55.152000+00:00	https://www.outsystems.com/Portal/Trial_Portal	track
facebcda-a0d5-4637-9c88-a350f10b792a	userA@email.com	userA@email.com	2023-05-13 18:05:55.156000+00:00	https://www.outsystems.com/Portal/Trial_Portal	track

Example User Journey

The above shows an example of how a user's journey was tracked before and after signup through the added UNIVERSAL_ID identifier. This Universal ID was used as the link with the CRM system mentioned before.

Features

Multiple previous works have highlighted the importance of time spent as an indicator of user interest. As each of our events is recorded alongside a timestamp, time spent was calculated by taking the difference in timestamps between a user's successive page events. Google Analytics characterizes the end of a user session by 30 minutes of inactivity, so the calculated time spent variable was capped at 30 minutes, with the subsequent page event marking the beginning of a new session (Google 2023). The addition of a session variable allowed for the creation of aggregate session features such as time spent per session, average clicks per session, and unique pages per session, among others. A complete list can be found in the Appendix (A3).

The six months of clickstream data holds over 690,000 unique URLs, many of these stemming from URLs with personalized queries embedded (marked by the presence of a "?" within the URL) or specific forum entries. To transform these into something more manageably modeled on a personal computer, URLs were sliced only to contain information between the third and fourth slashes. These URL segments effectively act as 'categories'. Country codes

were removed from URLs to ensure they did not appear in the list of segmented page paths and any remaining segments with queries were not considered, leaving us with 164 categories. These categories house events associated with URLs up to and after the fourth slash. Events with less than four slashes (home page events) were removed as they were assumed to hold no valuable information. As well as limiting complexity, slicing the URLs into segments is hoped to make the model more robust over time; while individual pages may change, sections are more likely to stay the same. Below are examples of categories created from raw URLs.

URL	Category
https://www.outsystems.com/profile/qdj5rpvyud/overview	profile
https://www.outsystems.com/training/lesson-quiz/1917/quiz?LearningPathId=18	training
https://www.outsystems.com/forge/component-overview/1417/common-plugin	forge
https://www.outsystems.com/Platform/Signup	Platform
https://www.outsystems.com/Portal/Trial_Portal	Portal

Category Examples

To capture the activity component of track type events, activities were added as suffixes to categories. The category *Training* now contains subcategories *Training_Close, Training_Download*, *Training_Submit,* and so on. All events within Training are counted under both *Training* and their activity subcategory if applicable.

Methodology

Many of the works that have made use of clickstreams classify or predict user behavior within a single session based solely on events collected during that session, for example, the prediction of user exits (Tobias Hatt 2020). In this work, users will instead be classified using the entirety of their recorded events over the chosen period. This choice was made for two reasons. The first is that OutSystems does not record session statistics. Session identifiers were a created variable using the best estimates of an industry leader (Google 2023). Because of this, we cannot know for sure where one session ends and another begins. The second is that this classification is based on the premise that users are repeat visitors; only users who sign up have been considered. Unlike in the e-commerce applications of clickstream research, we are not

concerned with what a visitor will do within a single session but instead with their displayed behavior over time.

A significant portion of the literature also makes use of the sequence of a visitor's clicks as a factor for behavior-based classification. Sequence importance, however, falls off beyond the session level as visitors' intentions might shift across sessions (Mitra 2020). Sequencing also typically involves the use of recurrent neural networks that are particularly good at dealing with data where order matters (Zachary C. Lipton 2015). These neural networks are complex models that can take significant time to train and take up considerable computational resources. Striving for a more simplistic approach, this work focuses on events at the footprint level. All recorded user events are considered without placing importance on where they fall in the scope of the user's journey. As this work was conducted with production in mind, performance is a pivotal factor calling for less complexity. Therefore, focus is placed on where users clicked, not how they got there.

To effectively model a user's browsing behavior across the OutSystems website on a footprint basis, primary importance is placed on the frequency of events, activities, and time spent across the created categories. To accomplish this, three pivot tables were created. The first shows the frequency of events a user has across categories, both page and track. This measures how many actions a user commits under a category, whether it be a page view or another interaction. The second shows which activities users performed across those categories, measuring what kind of interaction. The final pivot table shows the time a user spent under each category.

UNIVERSAL_ID	certifications	webinars	forums	training	community
userA@email.com	6	0	35	0	0
userB@email.com	18	19	0	0	0
userC@email.com	0	0	0	9	13

Events Frequency Table

UNIVERSAL_ID	certifications_Close	certifications_Download	certifications_Submit	webinars_Pause	webinars_Play
userA@email.com	5	0	1	0	0
userB@email.com	15	3	0	0	0
userC@email.com	0	0	0	7	7

Activity Frequency Table

UNIVERSAL_ID	certifications_time	webinars_time	forums_time	training_time	community_time
userA@email.com	5.03	0	60	0.00	0.00
userB@email.com	15.32	30	0	0.00	0.00
userC@email.com	0.00	0	0	7.32	8.11

Time Spent Table

The tables were merged by Universal ID along with the earlier mentioned aggregate session features to form the final features that would be used to classify our users in a tabular format that accommodates the use of multiple machine learning frameworks (Ravid Shwartz-Ziv 2022).

Model Selection

Despite taking measures to limit the number of raw page paths considered by transforming URLs into categories, the feature table encompassing time spent as well as frequency of events and activities is still highly dimensional with 1438 columns. On top of that, only a subset of total users have events under certain categories, leaving many columns with an abundance of zero values, a sparsity challenge common when analyzing clickstream data (Jian Hu 2007). The chosen model, therefore, must be effective at handling both high dimensionality and sparse data.

Tree Methods

Tree-based methods are among the most effective models when it comes to dealing with high-dimensional data (Cutler 2008). A tree is grown from a single node, the ‘root’, containing all the observations in a dataset. Observations from the root are then split into two branches based on their values for a selected feature. This process is continued with the tree making decisions at each new node, creating smaller, more specific groups based on different features. Eventually, the branch splitting process ends, leaving terminal nodes. To make classifications, new data is passed along the tree starting from the root until it reaches a terminal node and is

classified as the majority label of that node found in training.

Ensemble tree methods amalgamate the predictions of many trees to give an aggregated prediction. Random Forest is one such ensemble method that utilizes bootstrap samples and randomness throughout the tree growing process (Cutler 2008). They are fit to bootstrap samples using a random sample of predictors on which to split each node (Breiman 2001). Random Forest benefits from growing large trees which allows for low correlation between individual trees. The random sampling of predictors at each node forces trees to be unique, improving overall prediction power. Importantly for this work, Random Forests are capable of handling high-dimensional data without the need for formal feature selection.

XGBoost is another form of ensemble method that employs a refined boosting technique. It sequentially builds trees, each one focusing on correcting the errors of previous trees, thereby improving accuracy (Tianqi Chen 2016). XGBoost incorporates elements of regularization to avoid overfitting, which goes a step beyond traditional boosting algorithms. This regularization, coupled with its ability to handle sparse data and work with various types of predictive modeling problems, makes it exceptionally useful.

LightGBM, a similar gradient boosting decision tree method, further optimizes performance for high-dimensional data (Guolin Ke 2017). It introduces Gradient-Based One-Sided Sampling (GOSS) to focus learning on more informative instances by keeping instances with larger gradients and sampling those with smaller gradients. This maintains accuracy while reducing data size. Additionally, LightGBM's Exclusive Feature Bundling effectively reduces the feature space by bundling exclusive features, exploiting data sparsity for improved computation and memory usage. These innovative tools make LightGBM particularly adept at handling the challenges of large-scale, high-dimensional datasets, offering a robust solution for complex predictive modeling tasks.

Voting Classifier

The model chosen for this work is a soft-voting classifier composed of the three previously discussed models: Random Forest, XGBoost, and LightGBM. The use of a voting mechanism lowers the risk of mistakes made by a single model. A soft-voting classifier differs from a hard-voting classifier by using the predicted probabilities generated by base estimators to determine the final class prediction rather than a simple majority vote on the outcomes. In this ensemble, each base estimator contributes probabilistic forecasts for target classes, and the final class prediction is made by averaging these probabilities across all models. For a binary problem, if the average of probabilities assigned to a prediction of being the positive class by the base estimators is above 0.5, then the soft-voting classifier will predict the positive class.

Job Title Targets

The range of job titles collected through the CRM is quite large with 11 distinct occupations. The goal in predicting a user's occupation is to be able to engage with them based on the work they do. As many of the collected roles have similar functions, groupings were made to make sure that classification was possible. OutSystems is a low-code development platform with features mainly centered around development, so the company's prime interest is in identifying developers. Therefore, the model was trained once with the objective of identifying developers in a binary one vs. all format. To test the model's ability to recognize multiple classes, job titles were then grouped into IT, business, and developer roles for a second classification. The groupings and their sizes can be found in the appendix (A4).

The binary classification is imbalanced at an almost 1:2 ratio despite developers having more events than the other roles combined. An imbalance of observations runs the risk of the model not sufficiently learning from the minority developer class. There are several methods of artificially balancing a dataset to make up for imbalances, one of the more popular being SMOTE. SMOTE, which stands for Synthetic Minority Over-Sampling Technique, works by creating synthetic examples of the minority class using k-nearest neighbors (N. V. Chawla

2011). K-nearest neighbors, however, is known to struggle with high-dimensional sparse data so it's risky to apply it here as the generated synthetic samples might not be accurate representations of the minority class. Instead, it was decided to use class weighting. Class weighting adjusts the importance given to each class during the training of the model. Instead of generating new samples as SMOTE does, class weighting modifies the learning process of a model so that it pays more attention to the minority class, affecting the calculation of the loss during training. By doing so misclassifications of the minority class incur a larger penalty than misclassifications of the majority class.

Class weights for all three base estimators were set to be inversely proportional to the class frequencies in the data for both the binary and multi-class problems. Though the ratio in the future might change, the nature of OutSystems' product assumes some level of interest in development, making the increased weight on developers in both binary and multi-class contexts relatively harmless from an engagement perspective.

Hyperparameter Tuning

The performance of many machine learning algorithms depends heavily on hyperparameter settings. Although tuning can be a computationally expensive process, particularly as the search space increases, it is often necessary to achieve the best performance on a given task (Hilde J.P. Weerts 2020). Each of our selected base estimators has a set of tunable parameters to be optimized. Bayesian search, also known as Bayesian optimization, is an approach to hyperparameter tuning that seeks to minimize the number of function evaluations needed to find the best hyperparameters. Unlike a standard grid search which exhaustively tests parameters across the entire grid, Bayesian search uses a probabilistic model to predict which hyperparameter values are the most promising and chooses the next values based on this model (Ryan Turner 2021). Bayesian optimization is considered superior to random grid search (which selects the next parameter arbitrarily) as it intelligently uses

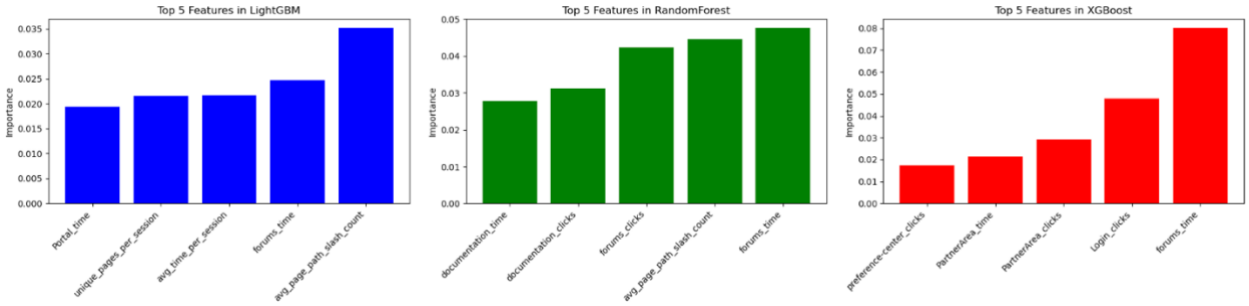
information from previous evaluations to inform future searches, making it more sample efficient and often leading to higher convergence to optimal hyperparameter settings. The optimal hyperparameter values for the binary classification using 10-fold cross-validation can be found below. The multi-class classifier was also tuned for the same parameters. Optimal values for the multi-class problem and descriptions for the parameters can be found in the Appendix (A5).

LightGBM		Random Forest		XGBoost	
learning_rate	0.01161	max_depth	16	gamma	0.26166
max_depth	20	max_features	auto	learning_rate	0.02066
num_leaves	80	min_samples_leaf	2	max_depth	7
n_estimators	380	min_samples_split	6	min_child_weight	3
		n_estimators	480	n_estimators	365

Optimal Hyperparameter Settings Binary Problem

Feature Importance

Feature importance measures allow us to get an understanding of what features the model is placing the most importance on when it comes to classification. It takes some of the ambiguity out of the machine learning process allowing results to be more palpable. Each of our base estimators allows for the extraction of feature importances, calculating them in different ways. Random Forest computes feature importance by averaging the decrease in impurity across all trees in the forest upon the addition of each feature. Features leading to a greater decrease in impurity are considered more important. XGBoost uses a similar approach but focuses on the gains brought by each feature when it's used in trees. The gain is the improvement in accuracy brought by a feature to the branches it's on. LightGBM uses a split based approach, calculating feature importance based on the number of times a feature is used in a split across all boosting trees. More frequent use in splits indicates higher importance.



Feature Importances

Above are the top five most important features for each of our base estimators for the binary problem of classifying developers. Each estimator is choosing different features to place importance on. All three models place importance on forum access both in terms of frequency and time spent. The OutSystems forum serves as a place to ask community leaders and experts questions and receive quick responses. A look through the most popular questions points to this being a strong indicator of being a developer. The portal and documentation categories also appear among the most important. The portal section of the website houses the cloud version of the OutSystems development platform and documentation provides details about how to use different features of the OutSystems product, both also seeming like strong development signals. It is important to note that there is not a clear domination of features by feature importance among any of our base estimators. Feature importance scores have been normalized to sum to one; apart from the .08 importance XGBoost places on the time spent on forums pages, no other features among the most important have scores above .05. This can be explained by the high number of features in use.

Experimental Setting

Validation Set

A difficulty in successfully building a model that is continuously able to classify users based on website usage is the propensity for change in web content and format. To accurately assess the robustness of the model, it is necessary to select a date-separated user group for the validation set. The same selection criteria and user mapping as before were conducted again

but only users that recorded their first event from September 2023 to the end of November 2023 were chosen, resulting in 7,177,164 events from 34,420 users. Selecting users by first event ensures there is no overlap in users between the training and validation set. Categories remain those created from the training set so any new categories that might have been created in the validation period will not be considered. The original 6-month dataset was also split 80/20 based on Universal IDs to generate an in-period test set.

Metrics

The evaluation metric chosen for the model was the Area Under the Receiver Operating Characteristic Curve (ROC AUC). The ROC curve is the plot of the True Positive Rate over the False Positive Rate across threshold levels providing a comprehensive view of model performance. The AUC, area under the curve, provides a singular scalar value summarizing the ROC curve. This value indicates the model's discrimination ability between the positive and negative classes with a score of 1.0 representing perfect discrimination and 0.5 denoting a performance no better than random guessing. ROC AUC is superior to traditional accuracy for several reasons. It provides a more reliable measure of performance in cases of unbalanced datasets, it measures performance across all thresholds giving a fuller picture of a model's predictive power, and the AUC standard error has been shown to decrease as the AUC and number of test samples increase (Bradley 1997).

To evaluate the multi-class problem in a similar way multi-class AUC was calculated by averaging pairwise class comparisons.

Results

ROC-AUC

Results obtained from the binary classification for the benchmark models (our untuned base estimators) and the soft-voting classifier can be found below.

Model	Training Average ROC AUC	Test Average ROC AUC	Validation Average ROC AUC
LightGBM	0.824	0.779	0.795
XGBoost	0.855	0.777	0.789
Random Forest	0.999	0.764	0.780
Soft Voting	0.835	0.779	0.798

Binary ROC AUC

The soft-voting classifier boasts the highest ROC-AUC score for the out-of-period validation set and performs equally to LightGBM for the in-period test set. Model performance is, however, relatively similar across all benchmarks and the selected model, indicating that model selection is not a highly important factor for our binary classification problem. Notably, each model maintained a similar performance for both in and out-of-period test sets, suggesting that the frequency and time spent methodology across categories holds up at least over the short period of time in between periods.

Model	Training Average ROC AUC	Test Average ROC AUC	Validation Average ROC AUC
LightGBM	0.790	0.728	0.734
XGBoost	0.825	0.727	0.737
Random Forest	0.999	0.706	0.712
Soft Voting	0.804	0.729	0.743

Multiclass ROC-AUC

The above showcases the results of the multi-class business, IT, and developer role classification. The way ROC-AUC was applied to the multi-class problem was by performing a one vs. all classification for each of the three classes and averaging the resulting ROC-AUC scores. The multi-class problem sees a decrease in AUC for all models across both the in-period and out-of-period test sets. The soft-voting classifier remains the superior model, but it is again clear that performance is similar across models. The decrease we see in average ROC-AUC indicates that all the models were not as effective at isolating IT or business roles as they were developers.

Probabilities

Though emails have essentially zero marginal cost in terms of sending, studies have shown that customers are overwhelmed with the amount of emails they receive (ROESLER 2017). An influx of non-relevant emails may cause users to avoid opening them altogether. It

would therefore be useful to have a level of certainty regarding predictions before sending anything to users. The way the soft-voting classifier works is by averaging out the probabilities each base estimator generates for the likelihood of an individual belonging to a certain class. If the average probability of being in the positive class is above 0.5 for a prediction in the binary-classification problem, it is labeled positive. These probabilities represent the model's certainty in its predictions. To have a better chance of sending emails to the desired audience it might be better to only consider predictions above a certain probability threshold.

Predicted Probability	Number Predicted (In Sample)	Correct Predictions (In Sample)	Number Predicted (Out of Sample)	Correct Predictions (Out of Sample)
70%	2594	1982	8262	6540
80%	1533	1278	4051	3406
90%	491	454	432	394

High Probability Threshold Predictions

The above table shows the number of developers predicted across 70, 80, and 90% probability thresholds for the binary classification problem as well as the number of correct predictions among them. The in-sample accuracy at these thresholds is 76.4, 83.3, and 92.5% respectively, showing the model gets progressively more accurate the more confident it is in its predictions. For out-of-sample predictions, accuracies of 79.2, 84.1, and 91.2% are seen, with the model continuing to perform well with high confidence predictions. It is important to note, however, that the number of individuals predicted at very high thresholds is minimal. The out-of-sample data has 15338 developers; if a 90% probability threshold was used, only 394 of them would be correctly identified, less than 3 percent. The mean probability assigned to developer predictions in the binary problem for the out-of-period test set is 0.71 (Appendix A8).

Setting probability thresholds for the multi-class predictions further showcases the model’s struggle with identifying IT and business roles as it makes no 90% probable predictions for either class. Business and IT role predictions have a mean probability assigned to them of .47 and .45, respectively, with the developer roles having a mean probability of .61 (Appendix A9). The quantity of IT and business predictions falls of substantially at an above 60 %

threshold showing that the model is not very confident in predicting said roles. The quantity of developer predictions at high thresholds (70% and above) also falls off in the multi-class context. When the model makes predictions with high confidence, accuracy remains impressive for the out-of-period test set (Appendix A7) with a 70% probability threshold leading to accuracies of .77, .83, and .87 for business, developer, and IT roles, respectively.

Discussion

The binary classification model displays relatively strong performance when it comes to discriminating developers. The similar performance across date separated test sets suggests that the aggregating of distinct URLs into categories successfully managed to nullify the effect of website changes and temporal behavior trends on the model's performance. In comparison with De Bock and Van den Poel's results when attempting to classify gender based on cross-site clickstream data, a similar binary demographic classification problem, the results of the proposed model are positive with an increase of more than 5 points in ROC AUC on the in-test period and 10 points on the out-of-test period when compared to their best model. Outside of binary developer identification, the model also outperforms their multiclass occupational classification with a 3-point increase in performance on the in-test period and a 5-point increase on the out-of-test period (De Bock 2010).

A main point of concern is the model's relative lack of confidence in making predictions in both the binary and multi-class contexts. In the binary context, only selecting visitors with a 90% chance or above of being developers, as predicted by the model, results in less than 3% of total developer observations in the out-of-period test set. In the multiclass context, confidence decreases further with two of the job role groupings not receiving a single prediction with a probability above 90%. This lack of confidence suggests similarity in behaviors across job roles.

Developers	Business	IT
Training	Training	Training
Forums	Portal	Documentation
Documentation	Documentation	Forums
Portal	Forums	Portal
Forge	Login	Login
Login	Forge	Forge
Community	Event	Support
Profile	Blog	Community
Certifications	Pricing-and-Editions	Cs Portal
Support	Partners Area	Event

Top 10 Categories by Time Spent

The above shows the top 10 categories by time spent across the selected job groupings making up 90 % of time spent by developers, 74% by business roles, and 82% by IT. It is clear to see that there is significant overlap across roles with Training, Forums, and Documentation categories (all thought to be strong developer indicators) found among the most popular categories for the three groupings. OutSystems is a low-code solution platform. The company website itself offers users a place to try the product, learn, and interact with other members of the low-code development community. It is, therefore, not surprising that most website visitors, no matter their official job role, display some level of developer behavior. The model's superior ability to distinguish developers as opposed to IT and business roles is perhaps an indicator of the ability to distinguish the professional from the amateur. Looking back at the feature importance, both Random Forest and LightGBM placed importance on aggregate features like time spent per session, clicks per session, and unique pages per session for the binary classification, suggesting the model is placing strong importance on how much the website is used when making its classifications. Developers account for more actions on the site than all other job titles combined as previously mentioned. They also have the highest values in the majority of the crafted aggregate session features (Appendix A10).

Limitations

The OutSystems website caters heavily to those interested in learning more about their development product and how to use it. Because of this, most visitors display some level of developer-like behavior, making it difficult to distinguish between classes with high

confidence. While some members with business roles do spend time looking at Pricing and Editions, most of their actions are in line with developer and IT roles. The job roles used as labels for this study were pulled from the CRM for users who met the qualifying conditions. It is not known for certain whether a user is using the website for professional or personal use or whether their job role in the CRM matches their intention with the OutSystems product professionally. Many website visitors opt for a free trial and make use of a free cloud development environment suggesting that they are not operating professionally or on behalf of their company. Because of this their affiliated job title might not be representative of their intentions with the website. A visitor might be a business analyst professionally with interest in learning about OutSystems for their own sake, perhaps to learn a new skill. Their use of the website then will have less to do with their current listed profession and more to do with their development interest.

Future Work

While the model shows success at differentiating professional developers from the rest of users on the OutSystems website, which was the main goal of this research, the lack of confidence, especially in the multiclass context, suggests the clickstream data might not be being used to its full potential. Wang et al. showed high success in their ability to classify users as malicious or otherwise using unsupervised learning to model behavioral patterns on social media sites, and then comparing their results with actual flagged accounts (Gang Wang 2017). Taking the context of the OutSystems website into account, an unsupervised study could be conducted to extract more actionable insights from the clickstream data. An attempt at classifying users by their respective development interests, labels for which are not readily available, for example web vs mobile development, might help further understand what individual users are actually interested in allowing for more successful targeted communications. As we are trying to decipher information about the end users for the purpose

of engagement, knowledge of their development interests could allow for further effective targeting.

Conclusion

Getting to know the end user of a B2B product and engaging with them can be a critical factor in making sure the key decision makers at a firm continue purchasing your product. With the advent of AI, companies can now generate informed estimates about users based on their interaction with the product or website. Clickstream data analysis, in particular, has proven to be an effective method of classifying users across a wide range of desired groupings, whether that is determining if a user is going to purchase a product in this session or if a user is a member of some group of interest. This thesis, undertaken on behalf of OutSystems, takes a simplified approach to clickstream analysis to identify users according to their job roles. Avoiding complex and computationally costly methods that make use of neural networks, the proposed model performs reasonably well in identifying developers, the main target of interest, boasting an ROC-AUC score of .798 on a group of users completely isolated from the training set having performed their first action after the date cutoff. The model will allow OutSystems to identify professional developers with a sufficient level of certainty making it possible to interact with them on topics they might be interested in to boost engagement. Perhaps more importantly, the model provides hope for further studies that can potentially reap further insights from the readily available clickstream data such as what kinds of development a user is interested in.

References

Amy Greiner Fehl, Valerie Good & Todd Arnold. 2023. "Exploring the drivers of B2B end user engagement." *Journal of Personal Selling & Sales Management* 43:3,159-177.

Borja Requena, Giovanni Cassani, Jacopo Tagliabue, Ciro Greco & Lucas Lacasa. 2020. "Shopper intent prediction from clickstream e-commerce data with minimal browsing information." *Scientific Reports* 10.

Bradley, Andrew P. 1997. "The use of the area under the ROC curve in the evaluation of machine learning algorithms." *Pattern Recognition* Volume 30, Issue 7, Pages 1145-1159.

Breiman, Leo. 2001. "Random Forests ." *Machine Learning* Volume 45, pages 5-32.

Cutler, Adele & Cutler, David & Stevens, John. 2008. *Tree-Based Methods.*

Davison, Brian D. 2002. "Predicting web actions from HTML content." *HYPERTEXT '02: Proceedings of the thirteenth ACM conference on Hypertext and hypermedia.* 159-168.

De Bock, KoenW. and Van den Poel, Dirk. 2010. "Predicting Website Audience Demographics forWeb Advertising Targeting Using Multi-Website Clickstream Data." *Fundamenta Informaticae* vol. 98, no. 1, pp. 49-70.

Gang Wang, Xinyi Zhang,Shiliang Tang,Christo Wilson,Haitao Zheng,Ben Y. Zhao. 2017. "Clickstream User Behavior Models." *ACM Transactiobs on the Web* Volume 11, Issue 4, pp 1-37.

Google . 2023. *About Analytics sessions.* https://support.google.com/analytics/answer/9191807?hl=en#:~:text=By%20default%2C%20a%20session%20ends,long%20a%20session%20can%20last.

Guolin Ke, Qi Meng,Thomas Finley,Taifeng Wang,Wei Chen,Weidong Ma,Qiwei Ye,Tie-Yan Liu. 2017. "LightGBM: a highly efficient gradient boosting decision tree." *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems.* Pages 3149-3157.

H. Robinson, A.Wysicja and C. Hand. 2005. "Internet advertising effectiveness-The effect of design on click-through rates for banner ads ." *International Journal of Advertising* 26(4),527-541.

Hilde J.P. Weerts, Andreas C. Mueller, Joaquin Vanschoren. 2020. "Importance of Tuning Hyperparameters of Machine Learning Algorithms."

Homburg, Christian, Marcus Theel, and Sebastian Hohenberg. 2020. "Marketing Excellence: Nature, Measurement, and Investor Valuations." *Journal of Marketing* 84 (4):1–22.

Hongri Liu, Shuo Wang, Yuliang Wei, Bailing Wang. 2021. "A novel classification model of collective user web behaviour based on network traffic contents." *IET Networks* Volume 10, Issue 4: Pages 173-144.

Jian Hu, Hua-Jun Zeng,Hua Li,Cheng Niu,Zheng Chen. 2007. "Demographic prediction based on user's browsing behavior." *WWW '07: Proceedings of the 16th international conference on World Wide Web.* 151-160.

Martin, Gillian. 2011. "The Importance Of Marketing Segmentation." *American Journal of Business Education* Volume 4, Number 6.

Mitra, Soumyadeb. 2020. *Clickstream Data Mining Techniques: An Introduction.* July 23. https://www.rudderstack.com/blog/data-mining-for-clickstream-analytics/.

N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer. 2011. "SMOTE: Synthetic Minority Over-sampling Technique."

OutSystems. 2023. *Who uses OutSystems?* https://www.outsystems.com/evaluation-guide/who-uses-outsystems/.

Randolph E. Bucklin, James M. Lattin, Asim Ansari, Sunil Gupta, David Bell, Eloise Coupey, John D. C. Little, Carl Mela, Alan Montgomery & Joel Steckel. 2002. "Choice and the Internet: From Clickstream to Research Stream." *Marketing Letters* 13,245–258.

Ravid Shwartz-Ziv, Amitai Armon. 2022. "Tabular data: Deep learning is not all you need." *Information Fusion* Volume 81, Pages 84-90.

ROESLER, PETER. 2017. *Nearly Half of Consumers Feel They Receive Too Many Marketing EmailsSurvey from Campaigner shows that email remains popular, but*

consumers can feel overwhelmed. . Inc. https://www.inc.com/peter-roesler/nearly-half-of-consumers-feel-they-receive-too-many-marketing-emails.html.

Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, Isabelle Guyon. 2021. "Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020." *Proceedings of the NeurIPS 2020 Competition and Demonstration Track.* 3-26.

SalingerPrivacy. 2020. *Cookies and Other Online Identifiers: Research Paper for the Office of the Australian Information Commissioner.*

Sismeiro, Randolph E. Bucklin and Catarina. 2009. "Click Here for Internet Insight: Advances in Clickstream Data Analysis in Marketing." *Journal of Interactive Marketing* Volume 23, Issue 1, 35-48.

Tianqi Chen, Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* Pages 785-794.

Tobias Hatt, Stefan Feuerriegel. 2020. "Early Detection of User Exits from Clickstream Data: A Markov Modulated Marked Point Process Model." *WWW '20: Proceedings of The Web Conference 2020.* 1671-1681.

Vantage Market Research. 2022. "Business-To-Business E-Commerce Market-Global Assessment and Forecast." *Business-To-Business E-Commerce Market* , March.

Vivek, Shiri, Sharon Beatty, and Robert Morgan. 2012. "Customer Engagement: Exploring Customer Relationships beyond Purchase." *Journal of Marketing Theory and Practice* 20(2),122-46.

Yoon Ho Cho, Jae Kyeong Kim, Soung Hie Kim. 2002. "A personalized recommender system based on web usage mining and decision tree induction." *Expert Systems with Applications* Volume 23, Issue 3,Pages 329-342.

Zachary C. Lipton, John Berkowitz, Charles Elkan. 2015. "A Critical Review of Recurrent Neural Networks for Sequence Learning."

Appendix

A1. List of Activities Associated with Track Events

Activities

- Submit
- Viewed
- Click
- Close
- Registration Begin
- Resume
- Complete
- Captured
- Seek
- Registration Success
- Top
- Play
- Filter
- Begin
- Failure
- Registration Failure
- Success
- Pause
- Schedule Success
- Download
- Schedule Begin
- Schedule Failure

A2. Job Title Distribution

Job Title	Count
Business Analyst	2742
CFO	225
CIO/CTO/Sr Executive	8448
Developer	26910
Dir/Sr Dir of Applications or IT	5270
Enterprise or Technical Architect	6624
IT Operations	5444
Line of Business Leader	2088
Manager of IT	10139
Media/Analyst	733
Security	885

A3. Aggregated Session Features

Feature	Description
Total time	Sum of a user's time spent across all pages over the time period
Clicks per session	How many actions a user preforms on a per session basis, could be page views or track types
Unique pages per session	Unique urls visited in a session, only page type events
Total clicks	Total actions a user preformed over the period
Avg page path slash count	Measure of complexity averaging the number of slices in URLs users visit. More slices implying a page housing specific content
Time spent per session	Average session length

A4. Job Groupings Distributions

Role	# of Individuals	# of Events
Developers	26910	14230205
All Other Roles	42598	8596549

Role	# of Individuals	# of Events
Developers	26910	14230205
IT	28368	6078083
Business	14236	2518466

A5. Hyperparameter Tuning for Multi-Class

LightGBM

PARAMETER	Value
learning_rate	0.16435523778789904
max_depth	3
n_estimators	339
num_leaves	84

Random Forest

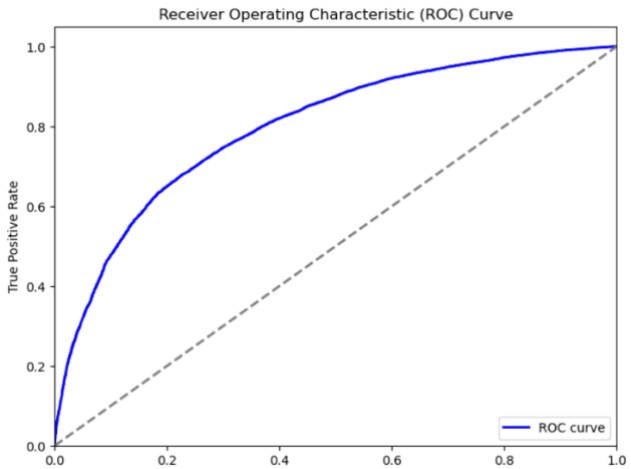
Parameter	Value
max_depth	57
max_features	auto
min_samples_leaf	3
min_samples_split	9
n_estimators	465

XGBoost

Parameter	Value
gamma	0.19339763210884187
learning_rate	0.07490319099950093
max_depth	9
min_child_weight	4
n_estimators	111

Parameter	Definition
Learning Rate	Determines the step size at each iteration while moving towards a minimum loss function.
Max Depth	Maximum depth of the trees being trained
Number of Estimators	Number of trees to be used in the ensemble
Number of Leaves	Maximum number of leaves for each tree
Max Features	Number of features to consider when looking for the best split
Min Samples Leaf	The minimum number of samples required to be at a leaf node.
Min Samples Split	The minimum number of samples required to split an internal node
Gamma	A node is split only when the resulting split gives a positive reduction in the loss function. Gamma specifies the reduction required to make a split
Min Child Weight	Minimum sum of instance weight needed in a child

A6. Binary Developer Prediction ROC Curve



A7. Predictions with Probability Thresholds Multi-class

Predicted Probability	Number Predicted (In Sample)	Correct Predictions (In Sample)	Number Predicted (Out of Sample)	Correct Predictions (Out of Sample)
70%	69	51	88	68
80%	5	4	17	17

Business

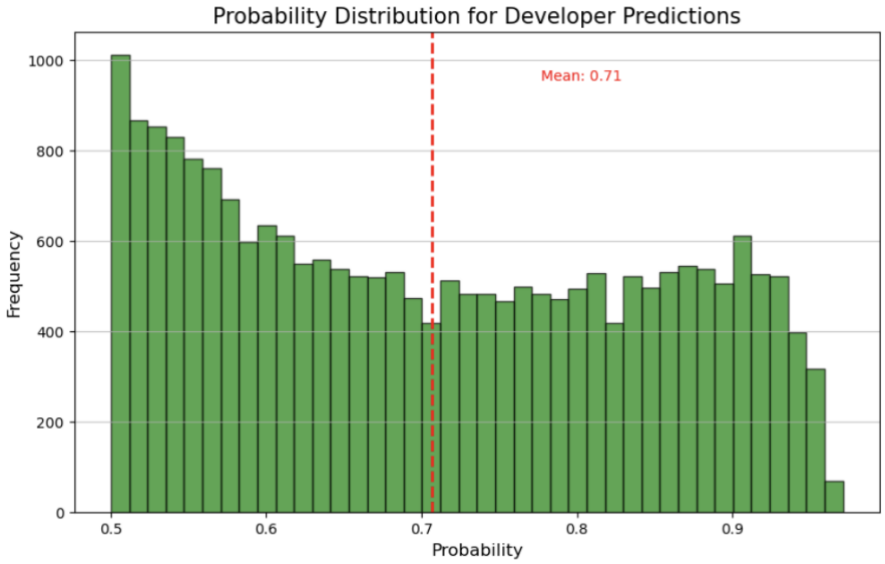
Predicted Probability	Number Predicted (In Sample)	Correct Predictions (In Sample)	Number Predicted (Out of Sample)	Correct Predictions (Out of Sample)
70%	1715	1448	4414	3674
80%	899	798	1558	1383
90%	152	150	18	18

Developers

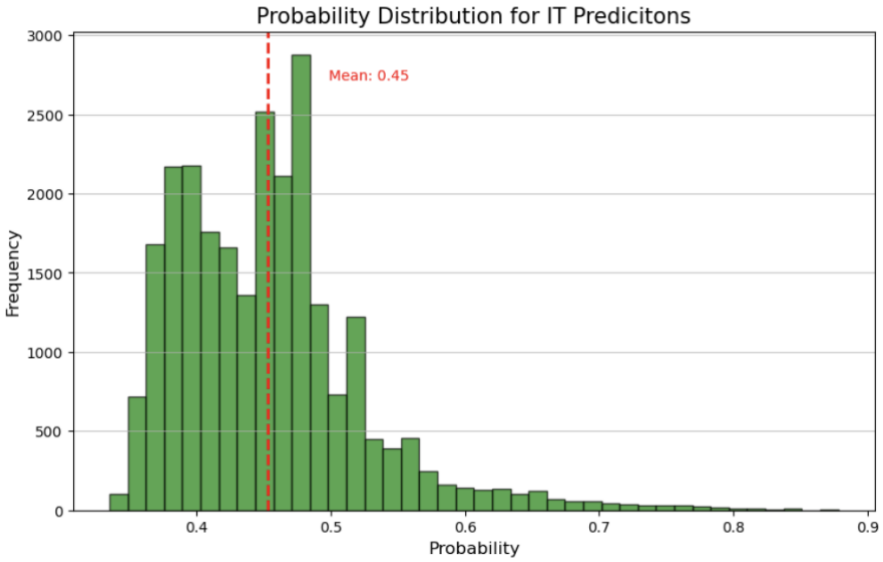
Predicted Probability	Number Predicted (In Sample)	Correct Predictions (In Sample)	Number Predicted (Out of Sample)	Correct Predictions (Out of Sample)
70%	12	8	15	13

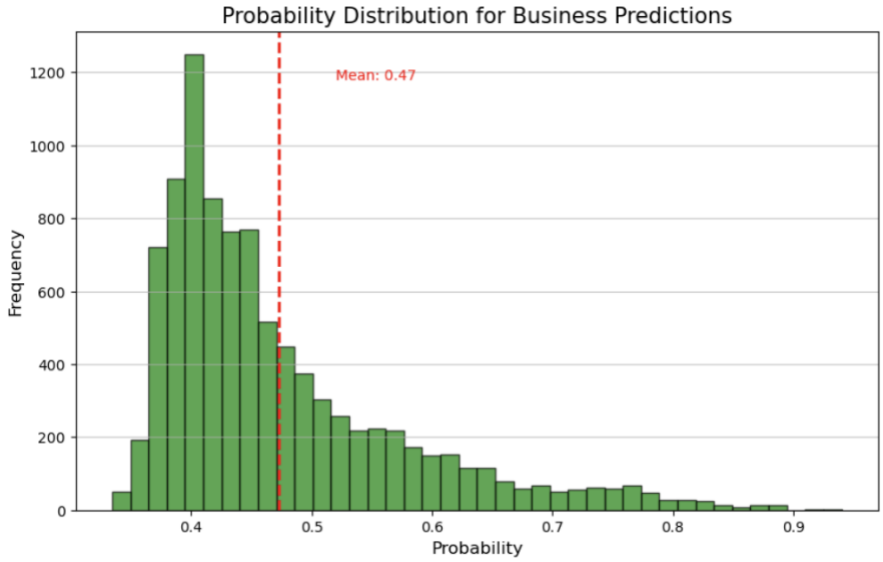
IT

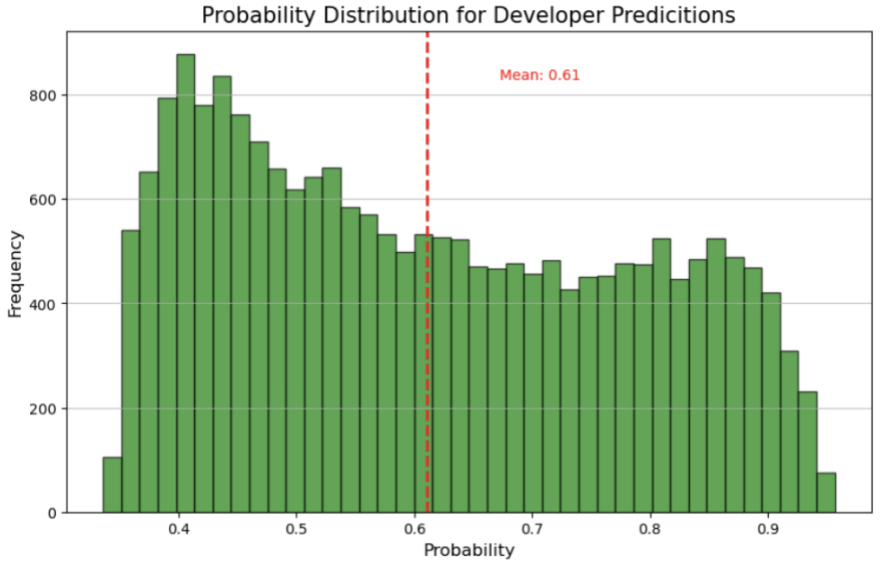
A8. Probability Distribution for Developer Predictions Binary



A9. Probability Distributions for Multi-class Predictions


Probability Distribution for IT Predicitons
Mean: 0.45
Frequency
Probability
0
500
1000
1500
2000
2500
3000
0.4
0.5
0.6
0.7
0.8
0.9


Probability Distribution for Business Predictions
Mean: 0.47
Frequency
Probability
0
200
400
600
800
1000
1200
0.4
0.5
0.6
0.7
0.8
0.9


Probability Distribution for Developer Predicitions
Mean: 0.61
Frequency
Probability
0
200
400
600
800
0.4
0.5
0.6
0.7
0.8
0.9

A10. Session Feature Values

JOB_TITLE	total_time_mean	total_time_median
Business	6692.81	3092.32
Developer	9195.61	5921.25
IT	8998.51	3993.39

Total Time

JOB_TITLE	clicks_per_session_mean	clicks_per_session_median
Business	4126.92	47.0
Developer	1367.35	37.0
IT	3201.45	38.0

Clicks per Session

JOB_TITLE	avg_time_per_session_mean	avg_time_per_session_median
Business	73.42	58.19
Developer	82.18	70.54
IT	85.34	62.67

Average Time per Session

JOB_TITLE	max_session_number_mean	max_session_number_median
Business	160.83	72.0
Developer	206.99	140.0
IT	187.81	95.0

Maximum Number of Sessions

JOB_TITLE	unique_pages_per_session_mean	unique_pages_per_session_median
Business	11.08	8.98
Developer	15.79	12.14
IT	12.75	9.83

Unique Pages Per Session

JOB_TITLE	total_clicks_mean	total_clicks_median
Business	7481.77	1710.0
Developer	5345.64	2069.0
IT	7129.80	1563.0

Total Clicks