The American University of Armenia

Capstone Project

CRM RETENTION WORKLOAD AUTOMATION

FOR IGAMING COMPANY

Proposal Prepared by **Esmila Sahakyan** in part fulfillment of the degree requirements

for the Degree of BS in Data Science in the

Zaven & Sonia Akian College of Science and Engineering

Supervisor: Hripsime Atanyan

Coordinator: Habet Madoyan

09/05/24

Abstract

In the rapidly developing sphere of iGaming, sustaining customer loyalty is crucial for the success of companies in this dynamic industry. The purpose of this Capstone project is to enhance the Customer Relationship Management (CRM) of one of the Armenian gaming companies with an innovative approach by automating the Customer Retention team's workload processes. The current manual CRM practices often result in inefficiencies, errors and delays in addressing customer needs, leading to potential customer dissatisfaction and churn. The aim of this project is to create an automated system that will optimize CRM retention workload and make it more efficient and effective. The system will help the company to increase customer satisfaction as it will reduce manual intervention and enable employees to provide timely, personalized interactions to customer preferences, behaviors and engagement.

Problems

There may be many problems encountered during this project. First of all, automation involves handling sensitive customer data, raising concerns about data security and privacy. So the data for testing the system may be artificially generated, trying to keep similar patterns with the original data. Another problem is that designing a complex system architecture may lead to challenges in scalability, maintenance, or unexpected dependencies. There can be problems related to employees who may resist adopting new automated processes, fearing job insecurity or unfamiliarity with the new system. After deployment, ongoing monitoring and maintenance are crucial. Neglecting this aspect may lead to system degradation or inefficiencies.

Results

Anticipated outcomes include a significant reduction in manual workload, improved accuracy in customer interactions, and increased customer loyalty. The study is essential for iGaming companies seeking to stay competitive in a market driven by customer experiences. By addressing the unique challenges of CRM retention workload through automation, this project aims to contribute to the overall growth and sustainability of iGaming businesses.

This capstone project aligns with current trends in iGaming, where technological advancements play a pivotal role in shaping customer engagement strategies. The proposed automated CRM retention system will be a strategic advantage, which will allow iGaming companies to rapidly and efficiently adapt to customer needs and promote long-term relationships and business success.

Literature review

During the literature review, I found several CRM tools that are specifically designed for gaming companies to manage player relationships and enhance engagement. Some of these tools offer features linked to the unique needs of the gaming industry, including player segmentation, in-game analytics, communication channels, and loyalty program management. Some of the tools used by gaming companies are:

- PlayerXP: PlayerXP is a CRM platform designed specifically for the gaming industry. It offers features like player segmentation, behavior tracking, personalized messaging, and performance analytics. PlayerXP helps gaming companies understand player preferences, optimize engagement strategies, and maximize player lifetime value.
- deltaDNA: deltaDNA is a comprehensive analytics and CRM platform for game developers and publishers. It provides tools for player segmentation, event tracking, A/B testing, and real-time messaging. deltaDNA helps gaming companies analyze player behavior, optimize game design, and deliver personalized experiences to players.
- Xsolla: Xsolla is a platform that offers a range of services for game developers, including payment processing, player management, and CRM tools. Its CRM features enable gaming companies to classify players, send targeted promotions, and manage loyalty programs. Xsolla helps gaming companies monetize their games effectively and build long-term relationships with players.
- GameAnalytics: GameAnalytics is an analytics platform that provides insights into player behavior, engagement, and monetization. It offers player segmentation, cohort analysis, and predictive modeling features. GameAnalytics helps gaming companies understand player preferences, optimize game performance, and increase revenue.
- PlayFab: PlayFab is a comprehensive platform for game developers, offering features such as player management, analytics, and monetization tools. Its CRM capabilities enable gaming companies to segment players, send targeted messages, and manage in-

game economies. PlayFab helps gaming companies build scalable, data-driven experiences for players.

These are just a few examples of CRM tools created for the gaming industry that I found during my literature review. There are many other CRM solutions available that offer similar features and functionalities depending on the specific needs and preferences of a gaming company. It's essential for gaming companies to evaluate different CRM tools based on their requirements, budget, and integration capabilities to choose the right solution for their business.

Report

All of the CRM tools mentioned in the literature review are slightly different from the tool that I am creating for this project. During the research done with the gaming company's retention team's employees, I have found out that they are completing lots of manual work, including bonus calculations, choosing targets for their ongoing tournaments or campaigns, cleaning and processing data, categorizing players, etc. There are many CRM tools available that mainly use artificial intelligence and machine learning algorithms to automate CRM processes. Although they are useful for gaining insights into customer behavior and classification, most of them do not have functionalities used for data collection, cleaning, preprocessing and preparation. As far as the gaming company with whom I currently work already had a CRM tool similar to the ones that use already cleaned and preprocessed data to complete a specific work, I have decided to automate the manual processes of preprocessing and cleaning data, bonus calculations, choosing targets and player categorization.

I am going to create a web tool that will include most of the necessary functionalities for automating the retention team's manual process. The tool will be as simple as possible to make sure that the employees will not have any difficulties with using and adopting the new automated system. The automated system will include birthday and lost bet bonus calculations, campaigns and tournaments management, cleaning customers' data, categorizing players, and retrieving data about customers' statistics, including their playing behavior and activity status at the specified period of time. The codes included in the system are described below.

- **Automating Birthday Bonus Calculation** ***(code: Birthdays)***

  The purpose of this code is to automate the process of calculating birthday bonuses for customers within a CRM system. By developing a Python script, we aim to efficiently identify eligible customers, determine bonus amounts based on predefined criteria, and prepare necessary communication and bonus allocation files. The script developed for this code focuses specifically on automating the calculation of birthday bonuses. It involves data retrieval, manipulation, bonus calculation, and preparation of bonus allocation files.

- **Automating CRM Campaign Management** **(*code: CRM Campaigns*)**

  The objective of this code is to automate the management of CRM campaigns, including various types such as tournaments, support campaigns, and VIP campaigns. By developing a Python script, we aim to efficiently execute campaigns, retrieve relevant data, and prepare necessary files for campaign execution The script developed for this code focuses specifically on automating CRM campaign management tasks. It involves setting campaign parameters, reading data, retrieving user information, and preparing communication and bonus allocation files.

- **Automated User Last Category Function** ***(code: User Last Category)***

  The aim of this code is to develop a Python function, named Last Category, which facilitates the extraction and analysis of user category data from multiple Excel files stored in specific directories. By leveraging various libraries and functionalities, the function automates the process of data retrieval, processing, and analysis, ultimately generating summarized results in an Excel file. The function is used to retrieve users last category and/or most frequent, minimum , maximum categories based on the period provided by the employee.

- **Base Game ID Exclusions** ***(Code Base GameID with Exceptions)***

  This Python script retrieves information about various games from a database and categorizes them based on certain criteria. It focuses on TV games and virtual sports, categorizing them separately from other types of games. The script then exports the processed data into CSV files, allowing users to filter the data based on parameters such as country, provider name, and game type. Overall, it provides a streamlined way to extract, categorize, and export game data for further analysis or reporting purposes.

- **Tournaments, Campaigns, Excluded Users** ***(Code: Campaigns, Tournaments, Excluded Players)***

  This Python script interacts with a database to extract information related to tournaments, campaigns, and excluded users, then exports the retrieved data into Excel files. The user is prompted to choose which data they want to retrieve by specifying whether they are interested in tournaments, campaigns, or excluded users.

  The script contains three main functions:

  1. TournamentResults: Retrieves tournament results based on the provided tournament name and exports the data into an Excel file.
  2. Campaigns: Fetches campaign results using the campaign ID and saves the data into an Excel file.
  3. Excluded_users: Retrieves details of excluded users for a specific campaign and exports the information into an Excel file.

  Each function establishes a connection to the database, executes the corresponding SQL query, retrieves the data into a Pandas DataFrame, closes the database connection, and finally exports the DataFrame into an Excel file.

- **Clean User Data** ***(Code: Cleaning)***

This Python script is designed to clean and organize data related to marketing campaigns. It prompts the user to input various parameters such as user category, product, campaign type, communication type, and other details relevant to the campaign.

Here's what the script does in a nutshell:

1. **Initialization**: It initializes various parameters based on user input, such as user category, product, campaign type, communication type, etc.
2. **Data Cleaning**: The script performs several data cleaning operations:
    - It reads input data files and prepares them for processing.
    - Cleans data by removing duplicates and irrelevant entries.
    - Cleans data by excluding entries present in the blacklist.
    - Cleans data by excluding entries from daily lists.
3. **Data Organization**: The cleaned data is organized into a structured format suitable for analysis and reporting.
    - Adds necessary columns for campaign details.
    - Prepares data for communication type-specific processing (SMS, email, call, etc.).
4. **Output Generation**: The organized data is then exported into Excel files for further analysis and reporting.
    - Generates Excel files with cleaned and structured data.
    - Optionally adds dropdown lists for communication status, feedback, and agents for call campaigns.

Overall, the script automates the process of cleaning and organizing marketing campaign data, making it easier for marketers to analyze campaign performance and effectiveness.

- **Deposits in a given period of time** ***(Code: Despoitors)***

This Python script interacts with a database to retrieve and process data related to depositors. Here's a summary of what it does:

1. User Input: The script prompts the user to input various date ranges for deposit transactions and registration dates.
2. Data Processing: It processes the input dates and converts them into the required format for querying the database.
3. Database Query: The script connects to the database using pyodbc and executes SQL queries to fetch data based on the provided date ranges.
4. Data Extraction: It retrieves data about depositors from the database, including player IDs and their corresponding deposit amounts.
5. Data Filtering: The script filters out depositors who have made deposits within the specified date range but have not made any deposits outside that range.
6. Output Generation: The filtered data is then exported to a CSV file, including player IDs and their corresponding deposit amounts.

Overall, the script automates the process of identifying depositors who have made deposits within a specified date range and have not made any deposits outside that range, providing valuable insights for analysis and reporting purposes.

- **Users Played in a given period of time** ***(Code: Played not Played)***

This Python script aims to identify users who have played games and those who haven't within specified date ranges. Here's a simplified overview:

1. Initialization: The script initializes various parameters such as username, country, date ranges for bets played and not played, and game categories.

2. Date Conversion: It converts the input date ranges into the required format for SQL queries.
3. Data Retrieval: The script fetches data from a database based on the specified date ranges for both played and not played bets.
4. Data Filtering: It filters the retrieved data based on game categories, providers, and specific games if provided.
5. Identification: The script identifies users who have played games but haven't played within the specified date range.
6. Data Saving: It saves the list of identified users into a CSV file.
7. Logging: The script logs the execution details, including the username, category, provider, and game.
8. Execution: Finally, it runs the main process to execute all the defined steps.

In essence, this script automates the process of identifying users who have not played games within a specified time frame, providing insights for further analysis or targeted actions.

- **LostBet** ***(Code:Lostbet)***

This script is a part of a larger system for managing campaigns related to casino bonuses from lost bets. Here's a simplified what it does:

1. **Function Definitions**:
   - check1: This function checks if the Data Warehouse (DWH) is working and if free spins should be given. Depending on the conditions, it retrieves, filters,

    categorizes, calculates bonuses, and cleans player data. It then processes the data for sending SMS messages, excludes certain users, applies manual adjustments, and potentially creates notification files and splits the data for further processing.
    - check2: This function deals with a similar set of operations as check1, primarily focusing on handling manual responses to the campaign, making communication lists, creating notification files, and preparing data for Adminka.
2. **Conditional Execution**: Both functions are conditionally executed based on certain parameters such as whether free spins are to be given or if the DWH (Database) is working.

In summary, this script automates various steps involved in managing a casino bonus campaign, including data processing, user categorization, bonus calculation, communication list creation, and file preparation for further processing or notifications.

- **User GGR, Betamount, Bets** ***(Code: Users GGR, Bet Amount, Deposits, Bets)***

This script calculates various metrics related to user activity in a gaming platform (likely a casino or sports betting platform) within a specified date range. Here's what it does:

1. Function Definition (metrics):
    - This function takes parameters like specific user lists, start and end dates, and the source of the data (sport, casino, or both).
    - It reads user data from an Excel file if specific users are chosen.

    - It constructs SQL queries based on the chosen data source (sport, casino, or both) and retrieves relevant data from the database.
    - It aggregates the data if the source is chosen as "All" and combines it with the user list to ensure all users are accounted for.
    - Finally, it saves the calculated metrics to an Excel file and displays a message with the file location.
2. User Interface Interaction: It seems like there are some commented-out lines that may have been used for user interaction via command-line input, but it's currently relying on parameters passed directly to the function.

In summary, this script facilitates the extraction and analysis of user activity metrics from a gaming platform database within specified date ranges and user categories, providing the output in an Excel file for further analysis.

- **Sending notifications** ***(Code: Send_notification)***

This script handles the sending of messages or notifications (inbox or push) to users for various campaigns (like birthday, lost bet, high, VIP bonuses) in a gaming platform.

1. Function Definition:
    - This function takes parameters like the campaign type, communication type (inbox or push), and whether it's a freespin or bonus campaign.
    - It constructs a message based on the campaign type and communication type.
    - It specifies the file path where the campaign data is stored.

  - It interacts with an object called Adminka, possibly representing an administrative interface or tool, to send the message or notification. If it's a push notification, it includes a link.
2. User Interface Interaction: There are some commented-out lines that might have been used for command-line input to interactively choose the campaign, communication type, and bonus type. Currently, it's not interactive and relies on parameters passed directly to the function.

Overall, this script automates the process of sending messages or notifications to users for different campaigns in a gaming platform, facilitating engagement and communication with the user base.

—--------------*Other Functions, Classes used in the codes above*—-----------------

1.python_libraries

- ***code : good_functions***

This code performs several tasks related to data processing and communication:

1. Data Retrieval and Processing:
   - It retrieves data from SQL databases using specific queries provided as parameters.
   - The get_data_from_sql function connects to a database, executes a query, and returns the results as a pandas DataFrame.

- The get_bets and get_sport_bets functions generate SQL queries to retrieve betting data, such as bets placed on casino games and sports events, respectively.
- These functions allow filtering data based on various parameters like date, user ID, bet category, and more.

2. Data Manipulation:
   - Functions like convert_str_into_SQL_date and convert_into_adminka_date convert date strings into SQL-compatible or Adminka-compatible formats, respectively.
   - assign_game_types assigns types to gaming data based on certain criteria, possibly for categorization purposes.
3. Communication:
   - The code includes functions to send notifications via different channels:
     - send_notification_via_pushbullet sends notifications using the Pushbullet service.
     - send_mail sends emails using Microsoft Outlook.
4. Dependency Management:
   - The import_or_install function checks if a Python package is installed. If not, it installs the package using pip.

Overall, the code facilitates the extraction, processing, and communication of data from SQL databases, allowing for efficient analysis and user engagement in a gaming platform.

❖ ***code: OOP1***

1. Class Definitions: There are several classes defined in the code:

- ➢ P2P: This class handles P2P (Player to Player) bonus calculations. It reads data from an Excel file, processes it to determine bonus eligibility, and generates bonus lists for users.
- ➢ Analytics: This class is responsible for various analytics tasks, including merging SMS results, merging campaign lists, and calculating bonuses.
- ➢ BOG: This class seems to handle a specific type of bonus calculation, possibly related to a campaign named BOG.

2. Function Definitions: The code defines several functions within the classes to perform specific tasks related to each class's purpose. These functions include tasks like reading data from files, preprocessing data, generating bonus lists, etc.

Overall, the code is designed to handle data processing, analytics, and bonus calculations related to a gaming platform or similar online service. It involves reading data from various sources, processing it, and generating outputs such as bonus lists and analytics reports.

❖ ***Code: Adminka***

This code is a Python script designed to interact with a web application, specifically the administrative interface of the TotoGaming platform. Let's break down its functionality:

1. Imports and Setup: The code imports necessary libraries and sets up the environment, including Selenium for web automation.
2. Class Definition (Adminka): This class has the functionalities related to interacting with the TotoGaming admin interface. It includes methods for logging in, sending messages, and performing various administrative tasks.

3. Initialization: The __init__ method initializes the class instance, setting up necessary attributes like username, password, and country. It also opens a browser window and logs in if required.
4. Web Automation: Methods like set_driver and open_adminka handle the setup of the Selenium web driver and opening the admin interface URL, respectively.
5. 2-Factor Authentication Handling: The pass_2f_authentication method deals with the two-factor authentication process if required.
6. Login: The login method handles the login process by entering credentials into the login form.
7. Message Sending: Methods like send_inbox and send_push facilitate sending messages via the admin interface. They fill out message details like subject, content, recipient lists, and file attachments before sending.
8. Element Locators: Various XPath expressions are defined as string constants to locate specific elements on the web page, like input fields, buttons, and dropdowns.

Overall, this script automates the process of logging into the TotoGaming admin interface, sending messages, and possibly performing other administrative tasks, enhancing efficiency and reducing manual effort.

❖ ***Code : Paths***

This script is for setting up file paths and defining some variables related to file locations and balances. Here's a human-friendly summary:

1. Current Date Setup: The script retrieves today's date and extracts the year and month.

2. File Path Setup: Various file paths are defined, including paths for inbox folders, blacklist files, bonus-related files, daily call lists, and more. These paths are constructed based on the current year and month.

Overall, the script seems to be preparing file paths and defining variables related to file locations and balances for further use in data processing or file manipulation tasks.

2.User_category

- ***Code: user_cat***

This script contains several functions related to categorizing types of games, assigning rates, and calculating new bet amounts and user categories.

1. Assigning Game Types: The assign_type function categorizes games into different types based on their category, provider, and name. Types include virtual games, live casino games, poker, slots, and others.
2. Assigning Rates: The assign_rate function assigns rates to different game types. Rates are assigned based on the type of game, such as virtual sports, TV games, live casino games, etc.
3. Calculating New Bet Amounts: The NewBetAmountCalculator function calculates a new bet amount based on the month, initial bet amount, and rate. It adjusts the bet amount depending on whether it's the first day of the month or not.
4. Calculating Old Categories: The OldCategoryCalculator function categorizes users into different tiers (e.g., VIP, High, Medium, Low, FreeRoll) based on their new bet amount.

5. Calculating New Categories: The NewCategoryCalculator function calculates a new category for users based on their old category and deposit count.
6. Assigning Categories for Romanian Users: The ro_users_category function assigns categories specifically for users in Romania based on their new bet amount.

Overall, these functions are designed to automate the process of categorizing games and users, assigning rates, and calculating new bet amounts and user categories in a gaming environment.

❖ *Code: sql_queries*

This script is essentially a collection of SQL queries and functions designed to retrieve and process data related to casino bets, sport bets, poker bets, golden bets, and deposits.

1. C_Game Tables Queries: These SQL queries retrieve data from the C_Game, C_GameCategory, and C_GameProvider tables, which likely contain information about different casino games, their categories, and providers.
2. Casino Bets Function: This function retrieves data about casino bets within a specified time range, including details like player ID, bet amount, GGR (Gross Gaming Revenue), bet count, game category, and provider.
3. Sport Bets Function: Similar to the casino bets function, this one retrieves data about sport bets. It includes information such as player ID, bet amount, GGR, bet count, and game ID.
4. Poker Bets Function: This function retrieves data about poker bets, including player ID, bet amount, GGR, bet count, and session end date.

5. Golden Bets Function: Similar to sport bets, this function retrieves data about golden bets, including player ID, bet amount, GGR, bet count, and game ID.
6. Deposits Function: This function retrieves data about deposits made by users within a specified time range. It includes details such as player ID, deposit amount, deposit count, latest deposit date, and the platform (sport or casino) where the deposit was made.

In summary, these SQL queries and functions are designed to extract and process various types of betting-related data, including bets made on different types of games (casino, sport, poker, golden), as well as user deposits.

❖ *Code: user_categorization*

This script performs several tasks related to data processing and analysis. Let's break down each step in simpler terms:

1. **Importing Libraries**: The script imports necessary libraries like os, datetime, pandas, and custom functions from user_categories_functions and good_functions to handle data and file operations efficiently.
2. **Declaring Dates**: It calculates the start and end dates based on yesterday's date.
3. **Retrieving C_Game Tables**: SQL queries are used to fetch raw data related to gaming categories, providers, and games.
4. **Retrieving Bets Data**: More SQL queries are executed to retrieve data about bets made in the casino and sports categories within the specified time range.
5. **Retrieving Deposits Data**: Another SQL query fetches data about user deposits during the same period.

6. **Cleaning C_Game Tables**: The raw data retrieved earlier is cleaned and organized into a more structured format.
7. **Assigning Type and Rate**: The type and rate of each bet are determined based on the game category and provider.
8. **Cleaning Sport Bets Data**: Similar to casino bets, the data related to sport bets is cleaned and processed.
9. **Joining Bets Tables**: The cleaned data from casino and sport bets are combined into a unified table.
10. **Calculating New Bet Amount**: A new bet amount is calculated based on the month, bet amount, and assigned rate.
11. **Grouping Bets and Calculating Category Without Deposit**: Bets are grouped, and the old category is calculated based on the new bet amount.
12. **Cleaning and Joining Deposits Data**: Deposits data is cleaned and joined with the grouped bets data.
13. **Calculating Category With Deposit and Exporting**: The final category, considering deposits, is calculated, and the resulting data is exported to CSV files, one for user categories and another for general data.

In essence, this script automates the process of analyzing user bets and deposits, categorizing users based on their betting behavior, and exporting the results for further analysis or reporting.

3. vip_function

- ***Code: functions1***

This script is a toolset for handling data related to VIP bonuses and campaigns in an online casino setting.

1. Setup and Configuration: It imports necessary libraries and sets up some initial configurations like paths and date variables.
2. Data Retrieval from Adminka: It fetches data from a system called Adminka, presumably related to casino gameplay and bets. It prompts for Adminka credentials if required.
3. Data Filtering Functions: Several functions are defined to filter and process the retrieved data based on different criteria such as game type, provider, or category.
4. Calculation Functions: Functions for calculating Gross Gaming Revenue (GGR) and determining VIP bonuses based on GGR are defined.
5. File Handling Functions: Functions for generating output files, especially for Adminka and push notifications, are defined.
6. Execution and User Interaction: The script orchestrates the execution of the defined functions based on user input or predefined configurations.

In essence, this script automates the process of extracting, processing, and analyzing data related to casino gameplay and VIP bonuses, allowing for efficient management and execution of VIP campaigns and bonus distributions.

4. App

All of the codes described above are integrated in a web tool created with streamlit. The app is not deployed as the codes belong to a specific organization and not everyone should have access to it. The app is designed to manage various aspects of a CRM dashboard. It allows users to perform the following tasks:

- Fetch data on users' Gross Gaming Revenue (GGR), Bet Amount, Deposits, and the number of bets placed during a specified time period, with options to filter by different product types.
- Gain insights into users' last categories, showing categories from the last Monday based on selected dates, along with additional statistics.
- Perform data cleaning operations related to player lists, blacklists, and campaign data, with customizable parameters.
- Calculate cashback for a predefined percentage of yesterday's lost money and handle scenarios like giving free spins.
- Calculate and distribute birthday bonuses for users, with options to choose the country.
- Manage the calculation and distribution of registration bonuses for users.
- Set up and manage CRM promotions based on specified conditions, including campaign type, start and end dates, bonus types, and more.
- Handle scenarios related to user betting activity over specified periods.

Overall, the app provides a user-friendly interface for managing and analyzing CRM-related data and operations.

References

1. PlayerXP: PlayerXP. (n.d.). Retrieved from PlayerXP

2. deltaDNA: deltaDNA. (n.d.). Retrieved from deltaDNA

3. Xsolla: Xsolla. (n.d.). Retrieved from Xsolla

4. GameAnalytics: GameAnalytics. (n.d.). Retrieved from GameAnalytics

5. PlayFab: PlayFab. (n.d.). Retrieved from PlayFab