*Ad Exchange Workshop*

*2016-17*

*PineApple – Project Report*

*21/4/2017*

**מגישים:**

אורי בר אל 305135956

אייל וסרמן 305210189

יוסי דדוש 203317565

שני לויים 201507845



תוכן עניינים

[הקדמה 3](#_Toc471678638)

[קביעת ה-Campaign Opportunity Bid 4](#_Toc471678639)

[שערוך הרווחיות של קמפיין 4](#_Toc471678640)

[קביעת ה-Campaign Opportunity Budget בפועל: 4](#_Toc471678641)

[קביעת ה-UCS Bid 5](#_Toc471678642)

[קביעת רמת ה-UCS הרצויה: 5](#_Toc471678643)

[שיערוך המחיר הדרוש על מנת לזכות ברמה : 5](#_Toc471678644)

[קביעת Bid Bundle 6](#_Toc471678645)

[קביעת Price per Impression ממוצע לקמפיין 6](#_Toc471678646)

[קביעת ה-Bid Bundle עצמו 6](#_Toc471678647)

[ארכיטקטורה 8](#_Toc471678648)

[Classes 8](#_Toc471678649)

[Datasets 8](#_Toc471678650)

הקדמה

זהו דו"ח מסכם עבור קורס "סדנה בסוכנים ממוחשבים". במהלך הקורס מימשנו סוכן ממוחשב אשר מתחרה במשחק AdExchange.

במשחק זה, סוכנים שונים מתחרים במכרזים בסביבת AdExchange, אשר בה עליהם לקנות להתחרות בשלושה סוגים של מכרזים כאשר המטרה הסופית היא יצירת רווח כספי גדול משל הסוכנים המתחרים. שלושת סוגי המכרזים הינם: קניית קמפיינים של מסעות פרסום באתרי אינטרנט, התחרות על חשיפת אוכלוסיית הגולשים לפרסומות הקמפיינים, והתחרות על קבלת שירות סיווג הגולשים.

הסוכן שלנו מתנהל אל מול השרת, אשר מקצה קמפיינים, מקבל bids עבור הקמפיינים השונים, ואשר מעדכן את הסוכן לגבי מצבו במשחק. בעת ההתנהלות אל מולך השרת הוא מממש את האסטרטגיה שפיתחנו לקניית והשלמת מכרזים. בכדי לעשות כן, על הסוכן ללמוד את המשחק ולקבל החלטות בזמן אמת בכדי ליצור רווח כספי מקסימלי, תוך שמירה על מדד איכות גבוה.

שמו של הסוכן אשר מימשנו ואשר מתואר במסמך הינו PineApple.

למימוש ה-PineAppleAgent, קבוצתנו פעלה בשלושה שלבים:  
תחילה, נפגשנו ודנו ברעיונות בסיסיים וקוים מנחים למימוש הסוכן, תוך כדי מימוש בסיסי ויישום התכונות שרשמנו במסמך ה-specification.  
בשלב השני, לאחר המפגש עם הקבוצות האחרות, עסקנו בחידוד האסטרטגיה ומימוש המנגנונים של הלמידה החישובית.  
בשלב האחרון, הרצנו את הסוכן, וביצענו בו אופטימיזציות למקסום הרווחים.

עיקרי האסטרטגיה של הסוכן הם במישורים הבאים:

* חישוב "העומס" של כל הסגמנטים באוכלוסייה מדי יום, על מנת להעריך את מידת התחרות בהם על השגת impressions, כאשר הנחת הבסיס היא שככל שקמפיין יותר "עמוס" כך יידרש מחיר גבוה יותר כדי לחשוף את אוכלוסיית הסגמנט לפרסומות של הסוכן.
* בעת הצעת קמפיינים על ידי השרת – חישוב הצעת מחיר, ושערוך רווחיות הקמפיין בהינתן הצעת המחיר הזו (באמצעות למידה חישובית).
* חישוב של כמות ה-impressions שהסוכן מעוניין להשיג מחר, כתוצאה מכך לחשב את רמת הסיווג (UCS) שעל הסוכן להשיג, ולשערך את המחיר להשגת רמה זו (באמצעות למידה חישובית).
* מסירת bids לקבלת impressions תוך כדי התחשבות בפקטורים שונים.

במהלך בדיקות המשחק, הגענו למסקנה כי האלמנטים הרנדומיים במשחק מורידים מאיכות ה-predictions שהסוכן מבצע. עם זאת, בתוספת הבקרה שאינה משתמשת בלמידה חישובית, הביצועים היו טובים יחסית, כך איכות הסוכן נשמרה במשך שלבי המשחק.

חלק גדול ממימוש הסוכן היה איסוף נתונים ללמידה החישובית ועל כך השפיעו גורמים כגון בחירת המתחרים, וצורות של משחקים ספציפיים – עם זאת, הסוכן נועד להתחרות מול מתחרים כלליים.

הסבר על פעולת הסוכן

הסוכן שמימשנו מורכב משני מרכיבים עיקריים:

1. מסגרת ניהול המשחק, אשר מומשה ב-Java
2. מנוע חישוב וקבלת החלטות, אשר מומש ב-Python.

מרכיב ראשון – מסגרת ניהול המשחק

מחלקת PineAppleAgent אשר יורשת ממחלקת Agent ומממשת את הממשק שלו כלפי הסרבר. מחלקה זו אחראית לניהול הראשי של הסוכן והיא נבנתה על יסודותיו של ה- SimpleAdNetworkאשר ניתן לסטודנטים.

מחלקת ה- PineAppleAgentהוא נועדה להתממשק מול שרת המשחק בהתאם להודעות שהיא מקבלת ממנו, ובתוך כך לנהל את קבלת ושליחת ההודעות. בנוסף, היא שולחת שאילתות למנוע החישוב וקבלת ההחלטות ומקבלת ממנו תשובות, וכמו כן מתעדת נתונים וסטטיסטיקות חשובות ללמידת המשחק.

את התקשורת בין מסגרת הניהול לבין מנוע החישוב מבצעים על גבי pipe אשר מקשר בין שני התהליכים.

חבילת ה-tools אחראית על שמירת נתוני המשחקים לצורך שימוש עתידי וניתוח של המרכיב החושב בסוכן. החבילה מכילה שתי מחלקות: dataToCsv ו-CampaignStatus.

המחלקה dataToCsv

המחלקה אחראית על יצרת קבצי csv לשימוש בעת הלמידה החישובית. קבצים אלו ימולאו על פי ההודעות שמתקבלות מהשרת אצל הסוכן. הודעות אלו מפורסרות ומוצבות בקבצים באופן וקטורי לניתוח עתידי.

המחלקה CampaignStatus

המחלקה אחראית על יצירה ושמירה של סטטוס ארוך טווח של המשחק. סטטוס זה מכיל סטטיסטיקות על דרך פעולת המשחק שנשמר מהרצות רבות לצורך בחינת ההיסטוריה.

מרכיב שני – מנוע חישוב וקבלת החלטות (PinePy):

ה-PinePy חושף ממשק בצורת סקריפט בשם pyjava\_comm.py אשר מקבל עדכונים לגבי מצב המשחק מה-PineAppleAgent ועונה על השאילתות שה-PineAppleAgent שולח, כל זאת בהתאם למצב המשחק. ה-מנוע ה-PinePy מומש בשפת פייתון תחת ההכוונה להשתמש בספרייה sklearn לצורך הלמידה החישובית.

ב-PinePy נמצאת עיקר האסטרטגיה של הסוכן, ותפקידיו העיקריים הם:

1. חישוב הערכת מידת התחרות בסגמנטים השונים.
2. חישוב הצעת מחיר ל-Campaign Opportunities.
3. שערוך רווחיות הקמפיין בהינתן הצעת מחיר.
4. חישוב את רמת הסיווג (UCS) שעל הסוכן להשיג למחרת היום.
5. לשערך את המחיר להשגת רמה זו.
6. הרכבת חבילת ה-bid bundle לכל יום.

נפרט על פעולותיו של ה-PinePy בכל אחד מהמישורים הללו:

**הערכת מידת התחרות בסגמנטים השונים:**

נגדיר פונקציית אינדיקטור אשר קובעת אם הקמפיין c פעיל ופונה אל הסגמנט s ביום ה-d.

נגדיר מדד *למידת הביקוש של סגמנט ביום :*

כאשר היא קבוצת כל הקמפיינים שהתפרסמו עד כה במשחק.

הרעיון מאחורי נוסחה זו הוא שכל קמפיין הפונה לסגמנט s מוסיף עומס גדול יותר ככל שה-reach שלו גדל, ככל שקבוצת כל הסגמנטים שלו קטנה יותר וככל שתקופת הקמפיין קצרה יותר. בנוסף ההערכה שלנו הייתה כי את העומסים שנוצרים על סגמנט מסוים מקמפיינים שונים ניתן לסכום בצורה חיבורית.

אולם נשים לב לאבחנה כי אם הינו היום הנוכחי, בעת החישוב, הנוסחה מדויקת עבור כל .

אולם, בימים המקיימים , עשויים להתחיל קמפיינים אשר איננו יודעים עוד על קיומם, אשר יעלו את מידת הביקוש של סגמנטים.

לכן, נרצה להוסיף גורם תיקון למדד הנ"ל. גורם תיקון זה ישערך את הביקוש העתידי שיתווסף לכל סגמנט ע"י קמפיינים עתידיים. גורם תיקון זה יחושב כקמפיין דמה שיסמלץ את אי הידיעה של קמפיינים עתידיים, אך יוסיף לעתיד את העומס הצפוי על כל סגמנט.

את תוצאה זו אנו משיגים ע"י תיעוד של סטטיסטיקת קמפיינים על פני מספר רב של משחקים וע"י הוספת קמפיין ממוצע לכל סגמנט ולכל יום עתידי בעת החישוב (עבור ימים ).

כעת נגדיר מחדש את להיות:

כאשר הם קמפיינים שפורסמו ע"י השרת, ואילו הם קמפיינים "ממוצעים" לכל יום ולכל סגמנט.

*בדרך להגדרת מדד העומס של קמפיין עלינו להרחיב את הגדרת העומס להגדרות הבאות:*

*עומס של קבוצת סגמנטים ביום ה-d:*

הרעיון מאחורי נוסחה זו הוא שכל שעומס של קמפיין יכול להיראות כסוג של "התנגדות" של מצב השוק, משום שקמפיין אשר פונה לכל הסגמנטים של יכול לתעל את ה-impressions לסגמנטים העמוסים פחות, ממש כשם שבמערך של נגדים המחוברים במקביל עיקר הזרם יזרום אל הנגדים שערכם נמוך.

*באופן דומה עבור קבוצת סגמנטים וקבוצת ימים :*

כאשר כאן חוזרת ההנחה שניתן להעביר impressions מימים עמוסים יותר לימים עמוסים פחות באותו האופן.

לבסוף נגדיר עומס של קמפיין ביום כך:

חשוב לציין:

* ביום הראשון אנו מעריכים כי לכל הסוכנים יש קמפיין זהה לזה שלנו, וכך מצפים שמידת העומס שלו תעלה בצורה מלאכותית, וזאת כדי להיות אגרסיביים בהתחלה.

**חישוב הצעת מחיר ל-Campaign Opportunities:**

החישוב מתחיל מניחוש מחיר הצעה ראשוני:

כאשר קבענו את .

המטרה היא לתת מחירים יותר נמוכים באופן יחסי לקמפיינים עם reach גדול, מכיוון שזכייה בהם משמעותית יותר.

כעת נשים לב כי בספסיפיקציה של המשחק מתואר כי כל budget bid צריך להתאים לדרישות הבאות:

ולכן אם הניחוש גבוה מדי, ניתן הצעה מעט קטנה מן המקסימום ואם קטן מדי ניתן הצעה מעט גדולה מהמינימום.

**שערוך רווחיות הקמפיין בהינתן הצעת מחיר:**

*נשתמש באלגוריתם AdaBoost עם decision stumps (עצי החלטה בגודל 1) שיתבסס על היסטוריית משחקים קודמים תוך שימוש ב-features הבאים:*

|  |  |  |  |
| --- | --- | --- | --- |
| *Day (of computation)* | *Budget [$]* | *Campaign start day* | *Campaign end day* |
| *Video coefficient* | *Mobile Coefficient* | *Reach* | *Demand(c)* |
| *OML* | *OFL* | *YML* | *YFL* |
| *OMH* | *OFH* | *YMH* | *YFH* |

Table 1 : Profitability Features

*בשורות האחרונות ה-features מבטאים שייכות לסגמנט, והערכים מקבלים אפס או אחד.*

*את ה-prediction אנו מבצעים על פני מידע שאספנו ממגוון משחקים. אלמנט ההחלטה, כלומר 1 עבור קמפיין רווחי או -1 עבור קמפיין לא רווחי, נקבע בצורה הבאה:*

* *אם לפחות 100% מהקמפיין הושלם והרווח מהקמפיין גדול מאפס אז ההחלטה היא 1.*
* *אם לפחות 90% מהקמפיין הושלם והרווח מהקמפיין גדול מ-10 אז ההחלטה היא 1.*
* *אם לפחות 80% מהקמפיין הושלם והרווח מהקמפיין גדול מ-20 אז ההחלטה היא 1.*
* *בכל מקרה אחר, ההחלטה היא .*

*מחיר הקמפיין אשר יציע הסוכן PineApple יהיה:*

כאשר קמפיין נקבע כגדול אם תקופת הפעילות שלו הינה לפחות 4 ימים וה-reach שלו גדול מ10000.

**חישוב מספר ה-Impressions אשר נרצה להשיג עבור קמפיין:**

*עבור קמפיין בו נזכה, נרצה לחשב את המספר האופטימלי של impressions אשר נרצה להשיג בו, כאשר בראות עינינו שני פרמטרים חשובים אשר עליהם הוא משפיע: כמות הכסף אשר נקבל בגמר הקמפיין, דירוג האיכות של הסוכן בגמר הקמפיין.*

*לצורך כך אנו משתמשים בפונקציה הבאה אשר מבטאת שקלול של שני פרמטרים אלו כפונקציה של מספר ה-impressions הסופי עבור קמפיין c:*

*הפונקציה הינה:*

האלמנט השמאלי בפונקציה מהווה הערכה לרווח הכספי ביחס לכמות ה-impressions ואילו האלמנט הימני הינו השיטה בה במשחק מחשב את ערך האיכות החדש של הסוכן לאחר השלמת קמפיין. הפרמטר נקבע בסופו של דבר להיות 1 בקירוב.

את ערך הפונקציה מיקסמנו על ידי optimize.basinhoppin עם איטרציה יחידה לאור העובדה כי הפונקציה מאוד חלקה, וכך קיבלנו את מספר ה-impressions אותו אנו רוצים להשיג, אותו כינינו target number of impressions.

בקמפיינים גדולים לעיתים הפונקציה לא נותנת אומדן טוב למספר ה-impressions האופטימלי ולכן בפועל הגבלנו את כמות זאת ללא יותר מ-.

הערה על קמפיין התחלתי וקמפיינים במתנה

**קביעת Average Price Per Impression עבור קמפיין:**

נקבע את המחיר הממוצע ל-impression עבור קמפיין מסויים באופן הבא:

אך נשנה אותו במקרים הבאים:

**חישוב את רמת הסיווג (UCS) שעל הסוכן להשיג למחרת היום:**

רמת ה-UCS הרצויה מחושבת באופן הבא:

אם אין קמפיינים פעילים נבקש את הרמה הנמוכה ביותר.

אחרת, נמיין את הקמפיינים הפעילים שלנו על פי הפונקציה הבאה:

ולאחר המיון נבחר את הקמפיין הפעיל בעל הערך הגבוה ביותר – הוא הקמפיין אשר מגדיר את רמת ה-UCS הדרושה לנו. ככל שלקמפיין יש יותר impressions שטרם הושגו, ככל שיש לו מספר מועט יותר של ימים שנותרו וככל שגודל הסגמנטים שלו קטן יותר כך הוא דורש בצורה קריטית יותר רמת ה-UCS גבוהה.

הערך שמחזירה פונקציית ה מבטא בעצם את אחוז הגולשים שעלינו להיחשף אליהם למחרת יום החישוב על מנת להספיק להשלים את הקמפיין עד היום האחרון. על כן, "נעגל" את הערך לכפולה הקרובה של 0.9, אשר היא רזולוציית רמות ה-UCS, כפי שמתואר במסמך הספסיפיקציה של המשחק.

**שיערוך המחיר הדרוש על מנת לזכות ברמה :**

*הסוכן משערך המחיר הצפוי על-ידי אלגוריתם רגרסיה מסוג Support Vector Machine עם kernel מסוג RBF, אשר הביא לנו תוצאות טובות למדי.*

*ה-prediction משתמש ב-features הבאים:*

|  |  |  |  |
| --- | --- | --- | --- |
| *Next Day* | *Active networks* | *Last day networks* | *OML* |
| *OMH* | *OFL* | *OFH* | *YML* |
| *YMH* | *YFL* | *TFH* | *---* |

Table 2: UCS Prediction Features

* *הערך Active networks מבטא את מספר הסוכנים אשר מחזיקים בלפחות קמפיין פעיל יחיד ביום הבא.*
* *הערך Last day networks מבטא את מספר הסוכנים אשר מחזיקים בקמפיין פעיל אשר מחר זהו יומו האחרון.*
* *עבור הסגמנטים השונים: ה-features כאן מבטאים את מספר הקמפיינים אשר יפנו לסגמנט הרלוונטי למחרת יום החישוב.*

*אלגוריתם זה ישערך את הערך הבא של בסדרת של רמות ה-UCS, בערך זה נשתמש בתור ערך ה-ucs bid של הסוכן.*

**קביעת Bid Bundle:**

לכל קמפיין ה- bid bundle entries ייראו בצורה הבאה:

נחשב את מטרת מספר ה-impressions שאנו מקווים להשיג ביום הקרוב:

כאשר lvl היא הכפולה של 0.9 אשר תואמת את רמת ה-UCS הרצויה למחר.

כעת נמיין את הסגמנטים שאליהם פונה הקמפיין לפי ה-demand שלהם מחר, מקטן לגדול.

בנוסף, נחשב את ה"עומס הממוצע" של כל הסגמנטים הללו.

לכל קומבינציה של סגמנט, סוג פרסומת, סוג פלטפורמה, נחשב את המקדמים הבאים:

* מקדמים כפליים של הקמפיין:
* מקדם נרמול – מספר קבוע אשר בתור אופטימיזציה.
* מקדם "פאניקה" – שווה ל-1, אלא אם זהו אחד מהיומיים שאחרונים של הקמפיין, שם עולה ערכו.
* מקדם תפוקה – מחושב כך שאם מספר ה-impressions שאנו רוצי ם להשיג היום גבוה ממספר ה-impressions הממוצע שיכולנו להשיג על פני כל תקופת הפעילות של הקמפיין, אז ערכו עולה באופן יחסי:

לכל סגמנט של הקמפיין נקבע את המחיר ל-impression כך:

*נבחר את ה- המינימלי של הסגמנטים הראשונים בסדרה הממויינת שגודלם הכולל מבחינת אוכלוסייה הוא כמות ה-impressions היומי שאנו מעוניינים להשיג, כפול .*

*לכל אחד מהסגמנטים הללו נייצר bid מהצורה:*

הפרמטר הוא גודל הסגמנט, במידה וזהו לא הסגמנט ה-k ברשימה, אחרת הוא מבטא את מספר ה-impressions שעוד נותרו להשיג ממנו כדי להשלים את .

בסגמנט ה-k, כפי שנראה, ייתכן ונמלא רק חלק ממנו, על מנת לא לקבל יותר targeted impressions מאשר שהתכוונו.

## תרחישי קצה

כחלק מהרצת המשחק, ומחקר אופן הפעולה שלנו, החלטנו שבשל הרנדומיות במלווה במשך כל חלקיו, ישנם מקרים שעלינו להסתמך על כללי החלטה אסולוטים ולא רק למידה ממשחקי עבר מנגנונים לדוגמה בהם החלטנו להשתמש:

* קניית ucs באופן יחסי ליום בו המשחק נמצא – ככל שהזמן במשחק מתקדם, כך פחות קמפיינים מוקצים ומדד האיכות מתייצב. לפיכך בתחילת המשחק עלינו לשמור על מדד איכות גבוה, ולהשקיע בהשלמת הקמפיינים באופן הטוב ביותר, ובכך לזכות במוניטין גבוה להמשך המשחק.
* התנהלות קמפיין על סף סיום – ברגע שישנו קמפיין לקראת סוף תם החוזה, אנו שמים השקעה מוגברת להשלמתו ביומיים האחרונים לקמפיין. החלטה זו נקבעה כי כדי לקבל את התקציב המלא יש להשלים את הקמפיין, ולאחר שבזבזנו עליו תקציב, עלינו להשלים אותו בצורה הטובה ביותר כדי להצדיק הוצאות אלו. כמובן שזה בנוסף לשמירה על המוניטין של הסוכן לצורף קבלת קמפיינים עתידיים.
* התמקדות בקמפיינים בעלי impression גבוהה ופרק זמן ארוך – מהרצות, הגענו למסקנה שקמפיינים כאלו הינם רווחיים, ואף שהשלמת קמפיינים אלו הם מפתח בניצחון במשחק. בנוסף, בשל פרק הזמן הארוך, יש סיכוי גבוה להשלים את הקמפיין. בשל כדאיות זו אנו מבקשים רווח נמוך (כמעט מינימאלי) עבור impression כדי לקבל את הקמפיין בהסתברות גבוה. למרות שמדובר על תקציב נמוך, בשל מספר גבוה של impressions, בסופו של דבר הרווח הכולל יוצא משתלם.

שיטות לא מוצלחות שנוסו

במשך המימוש כולו פעלנו בהתאם הקו המנחה שהצגנו במסמך הספסיפיקציה הראשוני, ואכן כל המנגנונים שתוארו שם נכנסו בצורה זו או אחרת לסוכן שמימשנו. למרות ההצמדות למסמך המקורי, היו מספר שינויים והתאמות שגילינו בזמן הריצה, שעל פיהם שינינו את הלוגיקה של הסוכן.

* שינוי פונקציית ה-demand. (הסבר על ישן מול חדש ולמה היה שינוי(
* אולי עוד פונק' או רעיון לוגי שנשאר בחוץ? אורי/אייל – אתם התעסקתם בזה יותר

תהליך העבודה על הסוכן

בשלבים הראשונים של הסדנה, כל אחד מחברי הצוות למד את חוקי המשחק וחלב מה הדרך הנכונה להביא את הסוכן שלנו למצב של רווח מקסימאלי וכמובן עלות מינימאלית. בשלב ראשוני חשבנו כיצד נוכל להשתמש בנתוני המשחק כדי ללמוד לפעמים הבאות (באלו אלגוריתמים של למידה חישובים אנו מעוניינים להשתמש) ובתור מימוש ראשוני, מימשנו מתודות שאוספות ומנהלות נתונים אלו לצור שימוש עתידי.

ההחלטה שלנו לשימוש באלגוריתמים לומדים הביאה אותה להחלטה לממש את אלגוריתמים אלו בעזרת חבילות בפייתון כאשר הסוכן עצמו, שממומש בג'אווה, יתשאל את תוכנית הפייתון בקבלת ההודעות הרלוונטיות, וזה יחזיר לו תשובה. כמו כן, בחרנו שאת התכנית נשתף בין חברי הקבוצה דרך git.

ברגע שבחרנו את התשתיות והכלים איתם נעבוד, חילקנו את העבודה בין חברי הצוות לצורך פיתוח ראשוני של הסוכן חצי מהצוות יעסוק במנגנון הלמידה החישובית והחצי השני יעסוק בחלק הניהולי של הסוכן, ההתממשקות שלו עם הגורמים השונים וכן ניהול הנתונים בהם יעשה שימוש ע"י הפייתון, כל זה כמובן תוך התדיינות ושיתוף רעיונות והתלבטויות בכל שלב במימוש.

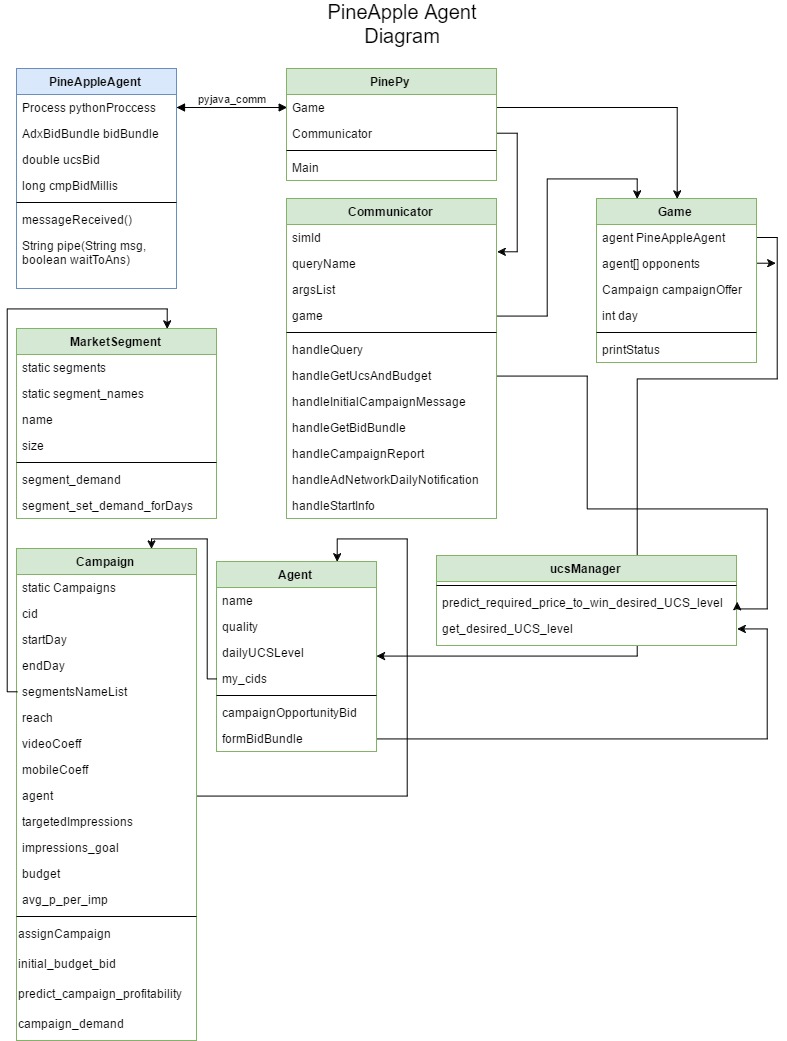
לאחר סיום המימוש הראשוני, הגיע שלב האינטגרציה בין שני החלקים. כולנו ישבנו כוות והסברנו על אופן המימוש של כל אחד מהחלקים ודרך הפעולה בה כל אחד נקט במימוש. בימים אלו בהם ישבנו ביחד התחלנו לחבר את הסוכן עם המוח עד לקבלת סוכן רץ שמבצע את הפעולות הנדרשות ממנו. אחת הבעיות העיקריות בשלב זה היה בהתממשקות בין שתי השפות – גילינו שהחבילות בהן נעשה שימוש בפייתון לוקחות זמן ריצה רב ושאילתות רבות גורמות לסוכן לא לעמוד בחלון הזמנים שניתן לו לקבל החלטה. בשל אילוץ זה החלטנו ליצור את השכבה האחרונה – לעטוף את קוד הפייתון כאפליקציה עצמאית שמורצת פעם אחת על ידי הסוכן ועונה לשאילתות ע"י pipe משותף ביניהן.

בישורת האחרונה, הגיע שלב האופטימיזציה והריצה של הסוכן – בשלב זה ביצענו החלטות ושינויים של קבועים ומנגנונים במשחק, וכן ביצענו הרצות מול סוכנים משנים קודמות כדי לשפר את ביצועי הסוכן שלנו לצורך רווח מקסימלי. בשלב זה גם עדכנו את קבצי ההיסטוריה לצורך התמודדות טובה יותר אל מול סוכנים חכמים.

ניתוח ביצועים והצגת סימולציות

יש להציג כאן תוצאות הרצה

ארכיטקטורה



מאגרי מידע (Datasets) -

* סטטיסטיקות ממשחקים קודמים על מנת שנוכל לעשות predictions ו-classifications על סמך ההיסטוריה. סטטיסטיקות אלו יכללו מידע ממשחקים שאנחנו הרצנו, מידע שנאסוף בין משחק למשחק לאורך התחרות, וכן מידע שנאסוף בזמן המתרחש ביום המשחק עצמו.
* מידע אודות הקמפיינים השונים שהוקצו לאורך ימי המשחק: בכמה הם נקנו, לאיזה סגמנטים הם פנו, מה היה הreach, מהו משך זמנם, מה היה הרווח מאותם קמפיינים, מה היה אחוז ההשלמה שלהם, ומה היה השינוי באיכות לפני ואחרי.
* מחירים של רמות UCS בכל יום.
* עומסים של סגמנטים בימים שונים (מבחינת ביקוש עבור אותו סגמנט על סמך העבר, ועל סמך הסטטיסטיקות במסמך המשחק).
* כמה סוכנים מחזיקים בקמפיין בכל יום.
* כמה סוכנים שונים זוכים ביותר קמפיינים מהשאר.