

## משימה ראשונה – קביעת מחירים ללקוחות בצורה סדרתית

השמועות על הצלחתכם הדהדו בתעשייה והגיעו לאוזני חברת הענק SomeSong, שידועה במכשיר הסמארטפון החדש שהיא התחילה לייצר, אליו יש הרבה ביקוש וקונים הממתינים לקנות אותו. SomeSong ביקשה מכם לעזור לה בקביעת המחירים למכשיר אותו היא מוכרת. מצד אחד, היא רוצה להציע מחירים ריאליסטיים ללקוחות כדי שיהיו מוכנים לקנות ממנה. מצד שני, היא רוצה גם להרוויח בתהליך ולהציע מכירים גבוהים. מכיוון שזה מוצר חדש, SomeSong אינה יודעת את המחיר שהכי כדאי להציע, ובידיה מידע מוגבל על כמה הלקוחות רוצים את הסמארטפון.

### תיאור המשימה:

אל חנות החברה נכנסים סדרת לקוחות אחד אחרי השני, כאשר כל אחד מהם מעריך את הסמארטפון של SomeSong בצורה שונה, המבוטאת כמספר ממשי בין 0 ל-1. בהמשך נפרט את ההנחות שאנחנו מניחים על ערכים אלו. עבור כל אחד מהלקוחות עליכם לקבוע מחיר למכשיר. לקוח בעל הערכה למכשיר של  $v \in [0,1]$  שהצעתם לו מחיר  $p$  יקנה את המכשיר אם  $v \geq p$ , כלומר, אם המחיר שהצעתם לו נמוך מההערכה שלו למכשיר. במידה והלקוח קנה את הסמארטפון, אתם והחברה מרוויחים רווח של  $p$ , המחיר בו מכרתם את המוצר. אם הלקוח לא קנה את הסמארטפון, אתם והחברה מרוויחים 0. לאחר כל הצעה של מחיר וקנייה או ויתור על הקנייה מצד הלקוח, אתם מקבלים פידבק – האם הלקוח קנה מכם את המוצר או לא.

**דגש חשוב: אתם מקבלים פידבק בינארי, אתם לא יודעים בדיעבד מה היה הערך של הלקוח!**

בכל אחד מהסעיפים הבאים אנחנו נניח התפלגות אחרת על ערכי המשתמשים  $D$ , וכן נניח שבידיכם מידע התחלתי שונה על אותה התפלגות (אם בכלל). בכל מקרה, אתם תמיד תדעו את משפחת ההתפלגויות על ערכי הלקוחות.

לסיכום, הפסאודו-קוד הבא מתאר את האינטרקציה שלכם (קובעי המחירים) מול הלקוחות, וכן מתאר את הרווח הכולל שלכם, שזהו המדד שעליכם למקסם. בהנתן שיש  $T$  לקוחות שמגיעים אליכם, האינטרקציה מתוארת באופן הבא.

1. הרווח הכולל של החברה מאותחל ל  $S = 0$ .

2. לכל  $t = 1, \dots, T$ :

a. הלקוח  $t$  מגיע למערכת והוא בעל ערך  $v_t \sim D$ . הערך  $v_t$  אינו ידוע.

b. המערכת שלכם מציעה לו ערך  $p_t$ .

c. אם  $v_t \geq p_t$ , המערכת שלכם מיודעת שהמשתמש קנה את הפריט שמחיר  $p_t$ .

הרווח הכולל של המערכת שלכם מעודכן להיות  $S = S + p_t$ .

d. אחרת, המערכת שלכם מיודעת שהמשתמש אינו קנה את הפריט במחיר  $p_t$ .

לחלק הבסיסי במשימה זאת יש שלושה שלבים, כל אחד מהם בעל משקל של 13% מהציון. ההבדל בין החלקים הוא בהתפלגויות על ערכי המשתמשים.

1. בחלק הראשון, הערך של כל המשתמשים הינו קבוע. כלומר, קיים איזשהו סקאלר  $x \in \mathbb{R}$

כך ש  $v_t = x$   $\forall t = 1, \dots, T$ . סקאלר זה אינו ידוע לכם בתחילת הריצה.

2. בחלק השני, ידוע לכם שהתפלגות המשתמשים הינה בטא עם פרמטרים ידועים שינתנו לכם

בתחילת הריצה (פירוט על כך ניתן בפסקה המסבירה על הקבצים המצורפים). כלומר, עבור

פרמטרים  $\alpha, \beta > 0$  שינתנו לכם בתחילת הריצה, מתקיים:

$$\forall t = 1, \dots, T : v_t \sim \text{Beta}(\alpha, \beta)$$

3. בחלק השלישי, ידוע לכם שהתפלגות המשתמשים הינה ממשפחת התפלגויות בטא, אך

בניגוד לחלק 2, לא ינתנו לכם הפרמטרים בתחילת הריצה. כלומר, עבור פרמטרים  $\alpha, \beta > 0$

0 שלא יהיו ידועים לכם בתחילת הריצה, מתקיים:

$$\forall t = 1, \dots, T : v_t \sim \text{Beta}(\alpha, \beta)$$

## פירוט על קבצי ההגשה בחלק 1:

בכל אחד מהקבצים הבאים עליכם להחליף את id1, id2 במספרי תעודות הזהות שלכם.

עליכם להגיש תיקיית זיפ בעלת השם part1\_id1\_id2 המכילה את הקבצים הבאים בהגשה

המתאימה לחלק הראשון במודל.

**PriceSetter1\_id1\_id2.py:**

• קובץ המכיל אובייקט בשם PriceSetter1 שמממש את הפתרון שלכם לחלק הראשון של

החלק הבסיסי.

- על האובייקט להכיל את הפונקציות הבאות:
  - `__init__(self, rounds)`: פונקציה זו מוגבלת לזמן ריצה של שנייה.
  - `set_price(self, t)`: קביעת המחיר ללקוח ה-`t`. פונקציה זו מוגבלת לזמן ריצה של 0.1 שניות.
  - `update(self, t, outcome)`: עדכון המודל לפי הפידבק שהתקבל, כאשר `outcome=True` אם המוצר נמכר, `False` אחרת. פונקציה זו מוגבלת לזמן ריצה של 0.1 שניות.

#### **PriceSetter2\_id1\_id2.py:**

- קובץ המכיל אובייקט בשם `PriceSetter2` שמממש את הפתרון שלכם לחלק השני של החלק הבסיסי.
- על האובייקט להכיל את הפונקציות הבאות:
  - `__init__(self, rounds, alpha, beta)`: אתחול המודל עם הפרמטרים  $\alpha, \beta$  של התפלגות בטא פונקציה זו מוגבלת לזמן ריצה של 3 שניות.
  - `set_price(self, t)`: קביעת המחיר ללקוח ה-`t`. פונקציה זו מוגבלת לזמן ריצה של 0.1 שניות.
  - `update(self, t, outcome)`: עדכון המודל לפי הפידבק שהתקבל, כאשר `outcome=True` אם המוצר נמכר, `False` אחרת. פונקציה זו מוגבלת לזמן ריצה של 0.1 שניות.

#### **PriceSetter3\_id1\_id2.py:**

- קובץ המכיל אובייקט בשם `PriceSetter3` שמממש את הפתרון שלכם לחלק השלישי של החלק הבסיסי.
- על האובייקט להכיל את הפונקציות הבאות:
  - `__init__(self, rounds)`: אתחול המודל. פונקציה זו מוגבלת לזמן ריצה של שנייה.
  - `set_price(self, t)`: קביעת המחיר ללקוח ה-`t`. פונקציה זו מוגבלת לזמן ריצה של 0.3 שניות.
  - `update(self, t, outcome)`: עדכון המודל לפי הפידבק שהתקבל, כאשר `outcome=True` אם המוצר נמכר, `False` אחרת. פונקציה זו מוגבלת לזמן ריצה של 0.3 שניות.

#### **Part1\_id1\_id2.pdf:**

- קובץ פידיאף באורך של עד לכל היותר עמוד המפרט את דרך הפתרון שלכם למשימות בחלק הראשון של התרגיל.

## פירוט על הרכב הציון בחלק 1:

- כדי לקבל את 13% הבסיסיים על המודל הראשון, עליכם לעבור את הציון הממוצע 467 בהרצת ה main בקובץ המתאים.
- כדי לקבל את 13% הבסיסיים על המודל השלישי, עליכם לעבור את כל המבחנים בהרצת ה main בקובץ המתאים.
- כדי לקבל את 13% הבסיסיים על המודל השלישי, עליכם לעבור את הציון הממוצע 230 בהרצת ה main בקובץ המתאים.
- 11 האחוזים התחרותיים בחלק זה יקבעו לפי תוצאות כל הקורס על 3 המשימות הבסיסיות. יתכנו בונוסים גבוהים על פתרונות מצוינים.

## משימה שנייה – השתתפות במכרזים סדרתיים

### תיאור המשימה:

חברת SomeSong החליטה לצאת במבצע לזמן מוגבל ולהציע ביטוח ללקוחותיה המוסדיים (ספקי מכשירים, רשתות סלולר, וכו') כדי להבטיח את שביעות רצונם ולהעניק להם ביטחון נוסף. החברה החליטה להשתמש במכרז סדרתי, כאשר בכל סיבוב כלל הלקוחות שעדיין לא רכשו ביטוח מתמודדים על פוליסת ביטוח יחידה (ראו פרוטוקול מטה). הלקוחות, שהיו רוצים ביטוח כמה שיותר ארוך על המוצרים שלהם, מתחרים במכרז בכל נקודת זמן על האפשרות לרכוש את הביטוח המוצע. במשימה זו, תתבקשו לייצר סוכן אוטומטי עבור לקוח בודד, אשר ישתתף במכרז הסדרתי בשם הלקוח אותו הוא מייצג. כל לקוח מנסה למקסם את הרווח שלו על ידי השגת עסקה המבטיחה ביטוח ארוך ככל שניתן במחיר הנמוך ביותר.

הסוכן שלכם פועל בשם לקוח אחד מתוך  $N$  הלקוחות שרוצים לקבל ביטוח. לכל לקוח  $i$  כזה יש ערך פרטי לקבל הביטוח לשנה שמתפלג  $v_i \sim \text{Beta}(5,2)$ . ערך זה ידוע ללקוח לפני תחילת האינטראקציה.

המבצע תקף ל  $M$  תקופות זמן, כאשר בכל תקופת זמן מוצע ללקוחות המתחרים ביטוח יחיד שרק אחד מהם יקבל. אורך הביטוח (בשנים) בסיבוב ה  $j$ , אשר נסמן ע"י  $r_j$ , מתפלג כמו  $r_j = 3X_j$ , כאשר  $X_j \sim \text{Beta}(2,5)$ . כלומר,  $r_j \in [0,3]$  לכל סיבוב  $j$ .

הביטוחים השונים שמוצעים על ידי החברה נחשפים ללקוחות המתחרים בצורה סדרתית ועבור כל ביטוח כזה נערך מכרז מחיר שני בין הלקוחות המתחרים עבור מי יקבל אותו. המחיר שמתחרים עליו במכרז הוא עבור יחידת שנה אחת מהביטוח, הלקוח שזוכה במכרז צריך לשלם מחיר זה עבור כל שנת ביטוח שהוצעה במסגרת הביטוח שזכה בו. הלקוח שזוכה משלם את הסכום שנקבע עבור ויצא מקבוצת הלקוחות המתחרים, שכן כבר קיבל ביטוח ואינו יכול לרכוש ביטוח נוסף. לאחר מכן הביטוח הבא נחשף ללקוחות שנשארו והתהליך ממשיך עד שאין יותר לקוחות פעילים או נגמרים הביטוחים (המבצע נגמר). ההצעות חייבות להיות אי שליליות.

תועלת הלקוח: כאשר לקוח  $i$  בעל ערך  $v_i$  מקבל את ביטוח  $j$  בעל אורך של  $r_j$  שנים במחיר  $p_{ij}$ , הרווח הכולל שלו מהעסקה הוא  $r_j(v_i - p_{ij})$ . כלומר, הרווח שלו מהביטוח לאורך השנים הינו  $r_j v_i$  והמחיר שהוא משלם לאורך השנים הוא  $r_j p_{ij}$ . בנוסף לכך, לכל לקוח מוצעת תוכנית ברירת מחדל  $r_0 = 0.5$  בחינם. כלומר, במקרה ובו לקוח  $i$  לא זכה בשום ביטוח במכרז, הוא יקבל

$$r_0(v_i - p_{ij}) = 0.5v_i.$$

להלן פסאודו-קוד המסכם את האינטראקציה:

1. לכל לקוח  $i = 1, \dots, N$  נדגם ערך לשנת ביטוח  $v_i \sim \text{Beta}(5,2)$ .
2. לכל יחידת ביטוח  $j = 1, \dots, M$  נדגם אורך  $r_j = 3X_j$ , כאשר  $X_j \sim \text{Beta}(2,5)$ .
3. קבוצת הלקוחות הפעילים מאותחלת להיות  $\{1, \dots, N\}$ .
4. לכל יחידת ביטוח  $j = 1, \dots, M$  או עד שקבוצת הלקוחות הפעילים ריקה:
  - a. הלקוחות הפעילים רואים את אורך יחידת הביטוח  $r_j$  ומחליטים האם להתחרות עליו.
  - b. נערך מכרז מחיר שני על קבלת יחידת הביטוח ה- $j$ , כאשר שבירת שוויון נקבעת באקראי במידת הצורך.
  - c. הלקוח המנצח  $i$  משלם לחברה את המחיר שהוסכם במכרז  $p_{ij}$  לכל שנת ביטוח שקיבל, כלומר  $r_j p_{ij}$  סה"כ, התועלת שלו מהעסקה היא  $r_j(v_i - p_{ij})$ .
5. כל לקוח שלא זכה בביטוח מקבל את ברירת המחדל.

## פירוט על קבצי ההגשה בחלק 2:

בכל אחד מהקבצים הבאים עליכם להחליף את id1, id2 במספרי תעודות הזהות שלכם. עליכם להגיש תיקיית זיפ בעלת השם part2\_id1\_id2 המכילה את הקבצים הבאים בהגשה המתאימה לחלק השני במודל.

**AuctionClient\_id1\_id2.py:**

- קובץ המכיל אובייקט בשם AuctionCompany שמממש את הפתרון שלכם להשתתפות במכרזים הסדרתיים.
- על האובייקט להכיל את הפונקציות הבאות:
  - `__init__(self, value, clients_num, insurances_num)`: אתחול המודל עם מספר הלקוחות המתחרים ומספר הביטוחים המוצעים. פונקציה זו מוגבלת לזמן ריצה של 2 שניות.
  - `decide_bid(self, t, quality)`: מחזיר הצעת מחיר עבור משך הביטוח המוצע במכרז ה-`t`. אם ההצעה היא -1, זה מייצג החלטה לא להשתתף המכרז ולא תוכלו לזכות בו. פונקציה זו מוגבלת לזמן ריצה של 0.5 שניות.
  - `update(self, t, price)`: עדכון המודל לפי התוצאה של המכרז ה-`t` לפי המחיר שהלקוח המנצח שילם בסיבוב זה. פונקציה זו מוגבלת לזמן ריצה של 0.5 שניות.

#### Part2\_id1\_id2.pdf:

- קובץ פידיאף באורך של עד לכל היותר עמוד המפרט את דרך הפתרון שלכם למשימות בחלק השני של התרגיל.

### פירוט על הרכב הציון בחלק 2:

- כדי לקבל את 37% הבסיסיים על חלק זה, עליכם לעבור את כל המבחנים בהרצת ה-`main` בקובץ המתאים. בחלק זה תתחרו נגד סוכנים נאיביים שלנו שמציעים את הערך האמיתי שלהם למוצר.
- 13 האחוזים התחרותיים בחלק זה יקבעו לפי תוצאות כל הקורס. בחלק זה נעשה תחרות בין המודלים שלכם בכל מיני סביבות שונות של המשחק (המודלים שלכם יתחרו בינם לבין עצמם, בלי מודלים סינטטיים שלנו), והמודל שיקבל את הרווח הממוצע הכי גבוה על פני כל התחרויות יקבל ציון גבוה יותר. יתכנו בונוסים גבוהים על פתרונות מצוינים.