

ComputEach

מגישים: מנחם ישראל בוגנים, ברוך מרדכי רוטקוף

מנחה: ד"ר שירה צוקר

הבעיה:

- אנשים רבים המעוניינים לקנות או לשדרג את המחשב שלהם, אינם יודעים איזה פריטי חומרה משפיעים על איזה פרמטר. (מה משפיע על המהירות, למשל?)
- כתוצאה מכך, אנשים הבוחנים מחשבים אופציונאליים, לא יודעים מה המחשב המתאים לצורך הספציפי שלהם. למשל לשימוש ביתי כמו גלישה באינטרנט ועבודה עם Microsoft Office יספיק מפרט פשוט יחסית, בעוד שגיימר יצטרך מפרט חזק למדי.

הלקוח:

- אנשים המשתמשים במחשב אך אינם מבינים בתכונותיו.
- משתמשים להם יש מחשבים שלא מספקים את צורכיהם ומעדיפים שדרוג על פני קנייה.

מענה לבעיה:

- עזרה בבחירת מחשב תוך הסבר על פריטי החומרה והדברים שצריך לבחור.
- בדיקת חומרת מחשב קיימת והצעת אופציות לשיפור.

יישום המענה על פי אפיון הלקוח:

- תוכנה לבדיקת חומרה של מחשב והצעת שיפורים על ידי תנאים ספציפיים (כמות סוקטים פנויים במחשב וכו') ואיסוף מידע למערכת ההמלצות (יפורט בהמשך).
- שאלון אינטראקטיבי עם הסברים מפורטים וידידותיים למשתמש על אפשרויות הבחירה העומדות בפניו, ונתינת תשובה על ידי ניתוח קבוע של תוצאות וכן תוספת המלצות ממערכת המלצות.

מימוש הפתרון

- **ממשק משתמש:** כדי לתקשר עם השרת הוספנו לו שכבת Bot, שמאפשרת לנו להגדיר סדרת פעולות שהמשתמש צריך לבצע, ללא התעסקות בגרפיקה.
- **כלי עזר:** כתבנו סקריפט בPython שמושך את המפרט של המחשב (כרגע רק בWindows) ומעלה אותו לשרת. השרת מעבד את הנתונים ונותן למשתמש המלצות לשדרוג.
- **המלצות:** המשתמש יכול לבקש מה-Bot המלצה על מחשב חדש, ולענות על מספר שאלות שיישאל על ידי ה-Bot שחלקן מכוונות למפרט הטכני וחלקן לשימוש הרצוי. בשלב שלאחר מכן השרת ינתח את דרישות המפרט הטכני, כמו כן את אופן השימוש ויחזיר מחשבים מומלצים.

- האלגוריתם:** על מנת להתאים בין הלקוח למחשב, מימשנו אלגוריתם מבוסס Machine Learning (ML) אשר משלב רגרסיה לינארית רב מימדית (חיזוי נקודות המייצגות ערכים רציפים במרחב התוצאות), עם אלגוריתם מסוג Unsupervised learning (למידה לא מושגחת) ונקרא KNN-K Nearest Neighbors. בהינתן דוגמה חדשה, האלגוריתם מחזיר ערך מאפיין לדוגמה. ערך זה הוא ממוצע ערכים של ערכי k השכנים הקרובים ביותר.

האלגוריתם מניח כי קיים ייצוג וקטורים ב \mathbb{R}^n של **המשתמשים** וייצוג ב \mathbb{R}^m של **המחשבים**.

אנו מוצאים בעזרת שיטות אופטימיזציה מבוססות Gradient descent את ההעתקה הלינארית האופטימלית ממרחב **המשתמשים** למרחב **המחשבים**. כאשר מתקבלים מהלקוח נתונים מאפיינים אנו ממירים אותם לייצוג וקטורי במרחב **המשתמשים**, מוצאים את **המשתמשים** הקרובים ביותר במרחב וממצעים בין הוקטורים המועתקים על ידי הטרנספורמציה למרחב הנסתר (latent space) של **המחשבים**. לאחר קבלת וקטור התוצאה הוא מומר חזרה מייצוגו הוקטורי לנתוני המחשב ומפיק המלצה ללקוח.
- אמצעים טכנולוגיים:** לצד השרת השתמשנו ב ASP.NET Core. ה-Bot הוא ספריית BotFramework, והכל מאוחסן ב-Azure.

לצד הלקוח השתמשנו ב-Python.

לתהליכי CI/CD השתמשנו ב Github Actions עם שימוש ב Docker וביצוע Static Code Analysis לבעיות אבטחה בעזרת Checkmarx.