אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

VELODROME

מגישים:

אביחי צרפתי 204520803

חיים רייס 319510475

מבוא

שיתוף האופניים של ניו יורק מאפשרת טיולי אופניים מהירים, קלים ובמחירים סבירים ברובע העיר ניו יורק. הם מפרסמים נתונים פתוחים באופן קבוע . מערך הנתונים מכיל 735,503 מידע על נסיעות אנונימיות שנעשו מינואר 2015 עד יוני 2017. באמצעות נתונים אלו נוכל לבצע Feature Engineering לחזות את מין הרוכב או במקרה שלנו יחד עם ***Velodrome*** נוכל לבצע המלצות לרכיבה על פי דרישת המשתמש.

האלגוריתם

תכננו אלגוריתם חכם לבחירת מסלול נסיעה עבור משתמשים. ראשית שאלנו את עצמנו מה עשוי לעניין משתמש במערכת שלנו. בין התשובות: מרחק המסלול, הזמן הקצוב לנסיעה, טמפרטורה, נופים, מסלול מעניין, תנאי תאורה, בחירה מושכלת מבין התוצאות המתקבלות. למזלינו, מהקובץ הנתונים שהתקבל יכולנו לאמוד הרבה מהפרמטרים בצורה ישירה או עקיפה. לדוגמא, אם קיים לנו במאגר נסיעה בשעות הלילה ממקום א' למקום ב' נוכל להעריך שתנאי התאורה במסלול הם טובים ושכנראה תנאי מזג האוויר בשעה זו אינם בעיתיים.

KNN הוא אלגוריתם מבוסס מרחק בו מסווגים נתונים על בסיס קרבה ל- K- שכנים. ואז, לעתים קרובות אנו מגלים שהתכונות של הנתונים בהם השתמשנו אינן באותה מידה / יחידות. דוגמה היא כשיש לנו תכונות גיל וגובה. ברור ששתי התכונות הללו כוללות יחידות שונות, גיל התכונה הוא בשנה והגובה הוא בסנטימטר.

השתמשנו ב Standard Scaling על מנת לפתור בעיה זו הוא פועל על ידי שינוי קנה המידה של תכונות כדי להיות מתפלג בהתפלגות נורמלית. כדי להשיג זאת, אנו משתמשים standardization כדי להפוך את הנתונים כך שיש להם ממוצע ($\bar{x}$) של 0 וסטיית תקן (σ) של 1.

$$x'_i = \frac{x_i - \bar{x}}{\sigma}$$

Formula Standard Scaling

בשימוש בספרייה `Scikit-Learn` נוכל לבצע את ה standardization על ידי שימוש בפונקציה *`StandardScaler`*. ביצענו ניסויים רבים על מנת להגיע ל K המניב את התוצאות הטובות ביותר, מצאנו כי **K=29** עונה לבקשתינו. על מנת לשפר את זמני הריצה קיפלנו את המודל לקובץ בינארי בשם `knnpickle_file` ובכך לקבל מודל מאומן ומוכל לבצע פרדיקציות על רשומות עתידיות, במקרה שלנו נזין לו את הקלט שהמשתמש הזין. במצב זה אנו מסוגלים לבצע פרדיקציה בזמן ממוצע של 3 שניות, מה שאינו פוגע בחוויית המשתמש.

עקב כך שהאלגוריתם הנ"ל מחזיר לנו מסלול אחד בלבד החלטנו להוסיף שיטת דירוג משוקללת משלנו עבור שאר ההמלצות שאנו צריכים להחזיר. בשיטה שלנו יש התחשבות בשלושה פיצ'רים מותאמים אישית שאנחנו חישבנו: הראשון הוא המרחק האוירי בין נקודת ההתחלה ונקודת הסיום (משקל 0.3), השני הבדל הזמנים בין השעה הנוכחית ושעה בה המסלול התחיל(משקל 0.3) והשלישי האם שעת ההתחלה של המסלול היא לפני רדת החשיכה או לא (משקל 0.4).

המרחק האווירי חושב ע"י נוסחת מרחק אוקלידי סטנדרטית, נסמנה d, ונסמן p1,p2 בתור זוגות סדורים של קווי אורך ורוחב:

**הערה בקוד עצמו יש המרה מקורבת לקילומטרים ומספר הקילומטרים מתוקן להיות קילומטר אחד בתור מינימום.*

$$d(p_1, p_2) = \max\left(1, \sqrt{\left(x_{p_2} - x_{p_1}\right)^2 - \left(y_{p_2} - y_{p_1}\right)^2}\right)$$

הבדל הזמנים בין השעה הנוכחית ושעה בה המסלול התחיל חושב ע"י מציאת מספר השעות בין הזמנים, נסמן h:

**הערה גם פה ניתן ערך מינימלי של שעה אחת*

$$h(t_1, t_2) = \max\{1, hours(abs(t\_2 - t\_1))\}$$

רדת החשיכה חושבה בהפשטה כלומר, האם שעת ההתחלה היא לפני או אחרי השעה 18:00, נסמן dl:

$$daylight(t) = \begin{cases} 1, if\ t \leq 18:00 \\ 0.5, if\ t \geq 18:00 \end{cases}$$

לאחר שסימנו את הפיצ'רים נשים לב כי d ו-h צריכים לבוא ביחס הפוך לתוצאת הדירוג ולכן נשתמש בנוסחה הסופית בהופכי שלהם (כלומר 1 חלקי הפונקציה):

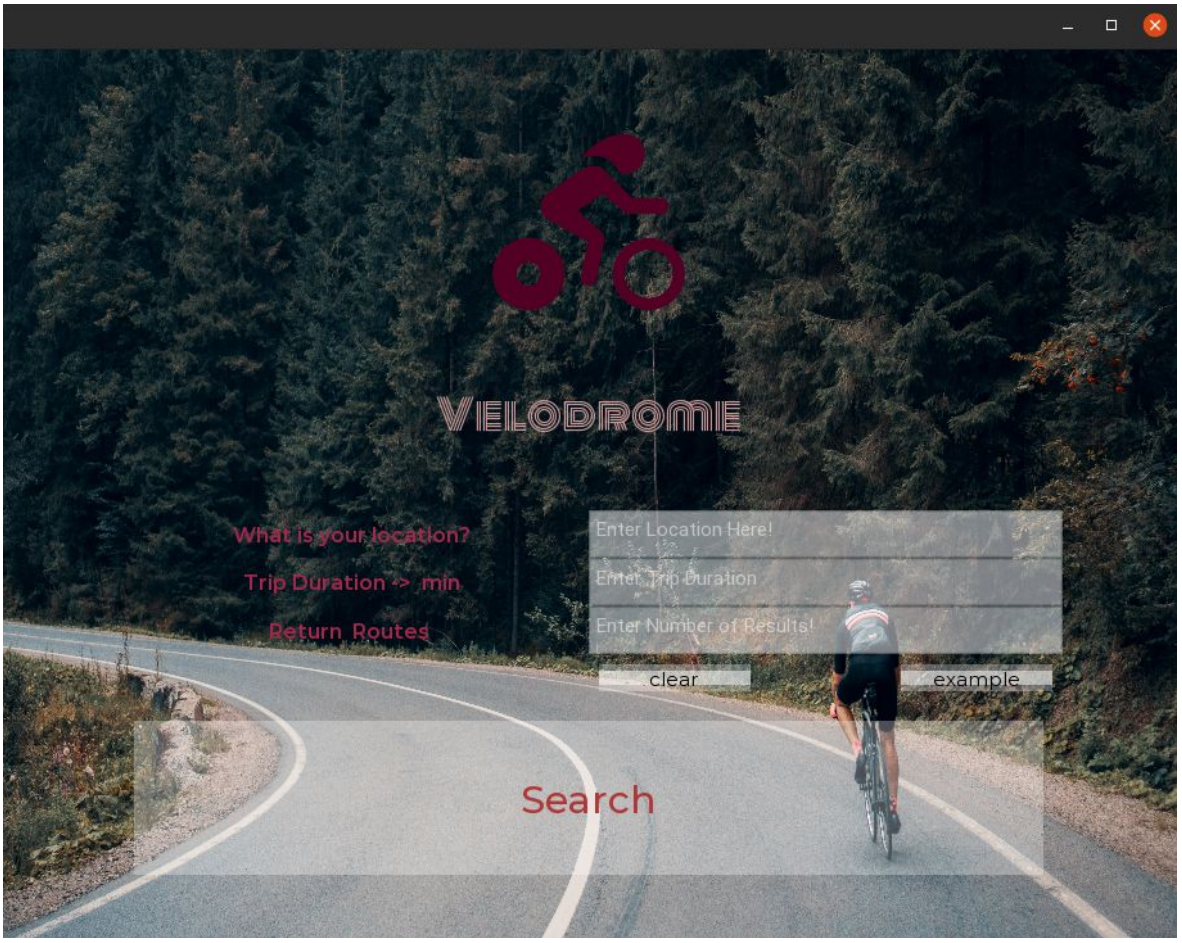
$$score = 0.1 + 0.3 * d^{-1}(start, end) + 0.3 * h^{-1}(currentTime, startTime) + 0.4 * dl(startTime)$$

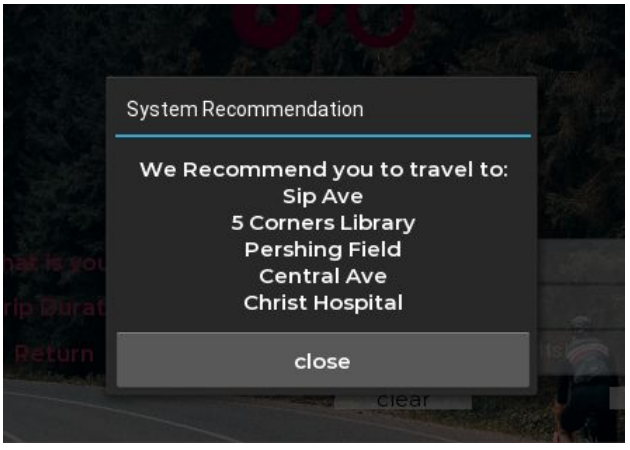
כעת במידה והרשומה אותה אנחנו בוחנים זאת אותה הרשימה שאלגוריתם ה KNN המליץ לנו עליה, אנו נוסיף 15% לניקוד עבורה על מנת לתת משקל נוסף הודות להצלחתה בבחינת האלגוריתם.

ממשק משתמש - GUI

בעזרת שימוש בחבילת kivy יצרנו ממש משתמש בסיס (איור 1) המספק את החוויה הדרושה לשימוש בתוכנית אותה כתבנו. הוספנו פונטים מתאימים במידה להגברת חווית המשתמש. הוספנו שני כפתורים פונקציונליים לשליטה על הקלט, clear עבור ניקוי השדות ו example מציב קלט שהוגדר מראש עבור דוגמא. הקפדנו על עיצוב נקי ונוח ככל האפשר.

איור 1: מסך ראשי





איור 2 : מסך המלצות

הנחות

הקוד מניח שקיים קובץ בשם `BikeShare.csv` וקובץ `knnpickle_file` המייצג את המודל כקובץ בינארי, שניהם בשורש התקייה.

חבילות שהשתמשנו בהן: `Pandas, numpy, sqlite3, Kivy, Flask, Scikit-Learn`.

מצורף קובץ requirements.txt עבור התקנה קלה של התלויות על ידי הפקודה

`pip install -r requirements.txt` דרך שורת הפקודה.