מציאה ואפיון של מסלולי טיול

מתן פנקס, [redacted]
נוי שטרנליכט [redacted]
בר אלוני, [redacted]

תאור הבעיה

עבור אלו שאוהבים לטייל, מידע על מסלולי טיול לא מוכרים עובר מפה לאוזן, בפורומים ובקבוצות פייסבוק. להשיג מידע מדויק ואמין על מסלול טיול לא מוכר יכול להיות קשה, ולקחת זמן רב.
לכן, בפרויקט זה נרצה ליצור מנגנון למציאה ואפיון של מסלולי טיול מתוך מידע ציבורי. בהינתן תיאור של מסלול הליכה המיוצג על ידי סדרה של נקודות GPS (קו גובה, קו אורך וזמן) נרצה ללמוד עליו פרטים רבים ככל האפשר השימושיים למטיילים: אורך המסלול, מה הן מקודות העניין במסלול, מה רמת הקושי של המסלול והאם הוא מעגלי או לא.
בנוסף, נרצה לאפשר למשתמש להזין נתונים המתארים את מאפייני המסלול אותו היה רוצה למצוא, ולהחזיר לו מסלולים מתאימים מתוך מאגר המסלולים שאפיינו.

פתרון הבעיה

המידע בו השתמשנו

נתאר את המידע בו השתמשנו לצורך פתרון הבעיה:

OpenStreetsMap Public GPS Tracks

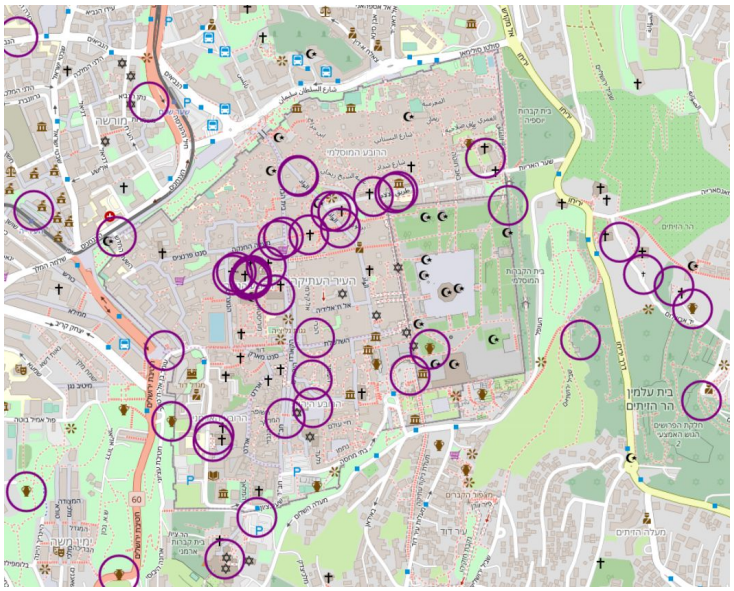
מסלולים שאנשים הקליטו ב-gps, ותרמו לאתר OpenStreetMaps. כל מסלול מורכב מנקודות המצויינות על ידי קווי אורך ורוחב על הגלובוס (ובהתאם להגדרות הפרטיות של המשתמשים, גם זמן וגובה). בנוגע לגודל, ה-tar שמכיל את כל המסלולים שמשתמשים על כדור הארץ העלו שוקל 21 ג'יגה כפי שניתן לראות כאן.
השגנו את המידע בעזרת ה-API של ל-OpenStreetMaps. בהינתן אזור גיאוגרפי, ה-API מאפשר לקבל מהאתר קובץ GPX המכיל 500 נקודות GPS שמשתמשים תרמו באותו אזור. לכן, כדי להשיג את המידע הדרוש לנו עבור אזור כלשהו, בנינו תוכנה שבהינתן ריבוע התוחם את האזור, מורידה מספר קבצי GPX כנ"ל כרצוננו.

The Hiking Project

אתר טיולים המכיל מידע על אלפי מסלולים ברחבי העולם. עבור רוב המסלולים, אפשר לחלץ מהאתר את פירוט נקודות ה-GPS שלהם (כולל מידע על גובה הנקודות מעל פני הים), אורך, דרגת קושי, ופרטי מידע נוספים (כמו לדוגמה האם המסלול ידידותי לכלבים). מבחינת גודל אין לנו הערכה מדויקת אך בהינתן שמסלול gps אחד שוקל כ- 30kb ובאתר יש כ-255 אלף מסלולים, מדובר בסדר גודל של 6.8gb.
השגנו את המידע על ידי כך שביצענו scraping לאתר.

OSM Maps

מפות מהאתר OpenStreetMaps. המפות מכילות "נקודות עניין" נקודות מתויגות המציינות אובייקט כלשהו על המפה (לדוגמה: בתי חולים, בנקים, אוניברסיטאות, מעיינות, פסגות הרים, ספסלי פיקניק ועוד ועוד). סדר הגודל של המידע הוא 800GB (עבור כל כדור הארץ). על מנת להשיג את המידע, השתמשנו בממשק בשם OverpassAPI שאפשר לקבל בעזרתו נקודות מתוייגות באזור גיאוגרפי נתון.



בויזואליזציה משמאל: נקודות עניין המתוייגות בתור אתרים היסטוריים שאיתרנו בעזרת המערכת שלנו בעיר העתיקה בירושלים.

U.S. Releases Enhanced Shuttle Land Elevation Data

דאטה-סט המספק את ערכי הגובה מעל פני הים של תאי שטח גיאוגרפיים. האתר https://dwtkns.com/srtm30/ מהווה ממשק להורדת נתונים אלו כפי שנאספו בפרוייקט Shuttle Radar Topography של נאס"א. השתמשנו במידע שאצור בקבצים הללו כדי להוסיף למסלולי ה-GPS המתוארים כרצף נקודות GPS, מימד שלישי שמסייע בניתוחנו אותם בצורה מעמיקה יותר. נדגיש כי הגובה ניתן ברזולוציה של תאי שטח בגודל 30 מ"ר. מסד הנתונים מספק בערך 2,000 אריחים, כל אריח בגודל ממוצע של כ-25mb. לכן גודל המאגר מוערך בכ-50gb.

איסוף הנתונים

כרינו מידע לגבי מסלולי טיול מהאתרים The Hiking Project ו- Open Street Maps.
לכל אתר מימשנו זחלן אינטרנט אחר, מכיוון שכל אתר בנוי בצורה שונה:
את osm אנחנו מתשאלים ע"י מתן שתי נקודות גובה ורוחב המתווים תיבה, שאת המידע שמופיע בה נרצה לכרות. תשאול זה נעשה ע"י שאילתא אחת ולכן הוא פשוט ואין בו נקודות כשל רבות. לעומת זאת כריית נתונים מ-hp מסובכת יותר. במימוש הזחלן האחרון רצינו ליצור מודל עצמאי שנוכל להאכיל אותו ברשימה של שמות של מדינות, והוא ייכרה את המידע על המסלולים השייכים למדינות אלו, מידע שכולל קבצי gpx ומידע לקסיקלי נוסף כגון תגיות עניין, רמת קושי וכדומה. מודל זה מבלה את זמנו בקשר רציף עם הרשת ולכן מאוד רגיש לכשלונות. כדי להתמודד עם מאפיין זה של הזחילה, ובכל זאת ליצור מודל עמיד, אנחנו שומרים את ההתקדמות שלנו בכל שלב (כמו לדוגמא כאשר אנחנו אוספים את כל ה- urls של המסלולים המשוייכים למדינה מסויימת, או מורידים את קובץ ה-gpx של מסלול מסויים ושואבים את שאר המאפיינים שלו מאותו העמוד ועוד) כאשר בכל מעבר כזה בין שלבים אנחנו יכולים להיתקל בכישלון. לכן מימשנו את המודל באופן שיאפשר לנו באופן אוטומטי להמשיך מהנקודה בה הפסקנו, ללא שמירת כפילויות, או חזרה על מסלול או מדינה שכבר נכרתה בהצלחה.

מתודולוגיה

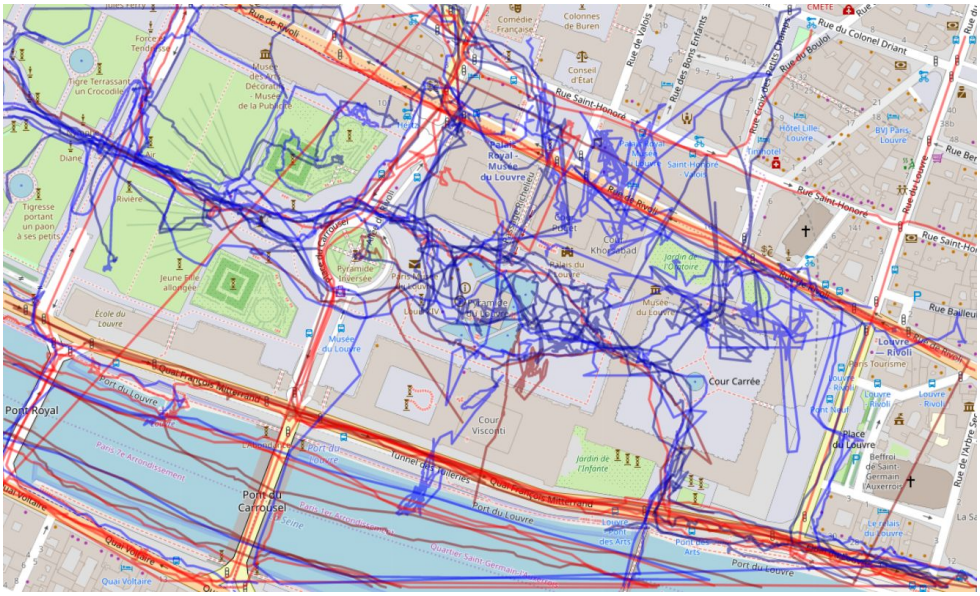
אפיון מסלולים ממאגר ציבורי

כמו שתיארנו בפרק על איסוף המידע, אנחנו מורידים מ-OpenStreetMap (או בקיצור OSM) מסלולים ציבוריים. המסלולים הנ"ל אינם מכילים אף פרט מלבד הקואורדינטות של הנקודות המרכיבות את המסלול ותג הזמן שלהן. על מנת לאפיין את מסלולי ה-OSM, היה עלינו לעבד את המידע הבסיסי הזה ולהצליב אותו עם מקורות שונים על מנת להעשיר את מה שאנחנו יודעים על המסלול. נתאר כעת כיצד עשינו את זה.

זיהוי מסלולי הליכה

כפי שנאמר, מסלולי ה-OSM לא מתוייגים בכל דרך שהיא ולא ניתן לדעת אם מסלול כלשהו התבצע בהליכה, ברכיבה על אופניים, בנסיעה במכונית או באמצעי תחבורה ציבורית. היות ובפרוייקט הזה בחרנו להתמקד במסלולי טיול, רצינו לסנן מסלולים שלא בוצעו בהליכה או ריצה.

מהירות הליכה ממוצעת של אדם היא בערך 4 ק"מ לשעה ומהירות ריצה קלה (jogging) היא בערך 10 ק"מ לשעה, לכן, אנחנו חושבים שסביר להניח שרוב המסלולים שהמהירות הממוצעת בהם קטנה מסף מסוים הם מסלולים שהאנשים שביצעו אותם הלכו או רצו. בהנתן הנקודות של מסלול ציבורי ותיוגי הזמן שלהם, אפשר לחשב בקלות את המהירות בין כל זוג נקודות (המרחק בין הנקודות חלקי הפרש הזמן ביניהן), לחשב את המהירות הממוצעת, ולסנן מסלולים שהמהירות הממוצעת בהם גבוהה מדי. במודל שלנו הגדרנו "מהירות גבוהה מדי" כ-12 ק"מ לשעה וסיננו כל מסלול שהמהירות הממוצעות שלו גבוהה מסף זה.



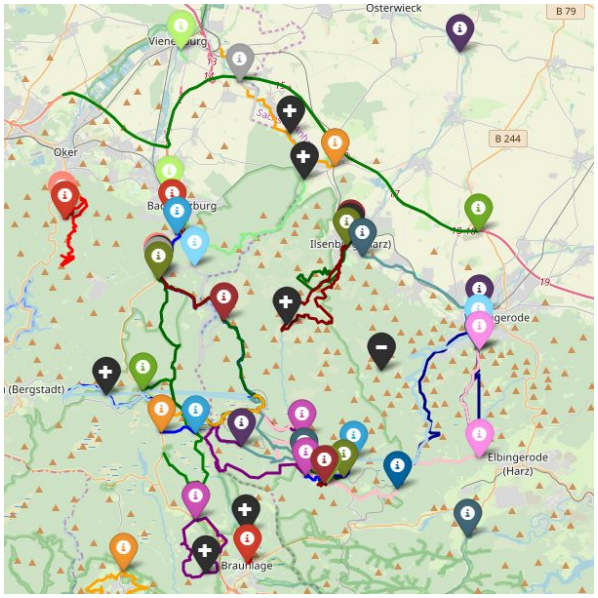
להלן ויזואליזציה להמחשת הסינון: הקווים הכחולים והאדומים בתמונה הם מסלולי OSM שכרינו מאזור מוזיאון הלובר בפריז. **בכחול** מסלולים שהמהירות הממוצעת שלהם היא לא יותר מ-5 קמ"ש **ובאדום** מסלולים שמהירותם גבוהה מכך. ניתן לראות שמרבית המסלולים באזור המוזיאון והגנים המקיפים אותו (איזור שבו מרבים לטייל ברגל) הם כחולים ואילו המסלולים בכבישים המקיפים את המוזיאון נוטים להיות אדומים.

מציאת האורך, נקודות העניין וצורת המסלול

פרטים נוספים שהסקנו לגבי כל מסלול הם: אורך המסלול, נקודות העניין במסלול וצורת המסלול.

בעבודה זו ביצענו חלוקה גסה של המסלולים שאספנו לשתי קטגוריות צורניות שונות: לולאות ומסלולים שאינם לולאות. הכוונה במסלול 'לולאתי' היא למסלול שנקודת הסיום שלו היא נקודת ההתחלה.
על מנת לקבוע את צורת המסלול (האם הוא לולאה או לא) בדקנו אם המרחק מנקודת ההתחלה לנקודת הסיום של המסלול קטן מסף מסוים. אם המרחק קטן מהסף, סיווגנו את המסלול כלולאה, ואחרת לא.

נמחיש את הסיווג של צורת המסלול על ידי שימוש בויזואליזציה הבאה: בתמונה רואים את האתר ההיסטורי Stonehenge, שהוא בעיקרון מעגל אבנים מפורסם באמצע אחו, מה שיוצר את תבנית התנועה שרואים בתמונה: אנשים מגיעים ל-Stonehenge מהכביש, מקיפים אותו, וחוזרים על עקבותיהם, מה שמוביל ליצירת מסלולים מעגליים (המסלולים שהמודל שלנו סיווג כלולאות מסומנים בתמונה **באדום** **ובכתום**).
נשים לב למסלולים שהמודל סיווג כלא לולאות (**בכחול** **ובסגול**). ניתן להבחין שאלו כל המסלולים של האנשים שחלפו ליד Stonehenge בכביש מבלי להיכנס אליו. יוצא הדופן היחיד הוא מסלול אחד (בסגול) שבו דווקא כן נכנסים ל-Stonehenge, אבל נקודות ההתחלה והסיום שלו רחוקות יחסית (הסוף שלו מופיע בקצה העליון של התמונה, באחו מהצד השני של הכביש).



נקודות עניין הן נקודות שעשויות לעניין את המטיילים במסלול. בעבודה זו בחרנו להתמקד ב-8 סוגים אפשריים של נקודות עניין: נהרות, מפלים, מעיינות, מערות, מקומות בעלי חשיבות היסטורית, מקומות בעלי חשיבות גאולוגית, ונקודות תצפית על ציפורים.
עבור כל מסלול, ולכל סוג נקודות עניין, השגנו את הקורדינטות של נקודות העניין על ידי תשאול מפות OSM, וקבענו שנקודת עניין שייכת למסלול מסוים אם המרחק המינימלי שלה ממנו קטן מסף כלשהו. כדי לייעל את תהליך שיוך נקודות העניין למסלול תחמנו כל מסלול ב-bounding box, ארבע הגבולות של המסלול מצפון, מדרום ממזרח וממערב, ובבואנו לשייך נקודות עניין למסלול התמקדנו רק בנקודות העניין על המפה שנופלות בתוך התיחום של המסלול. יעול נוסף שהכנסנו לתהליך, הוא 'לפשט' את המסלול הנבדק, כלומר להקטין את מספר

הנקודות במסלול על ידי דגימתו במרווחים שווים. באופן זה כמות הנקודות במסלול שצריך לחשב את המרחק מהן לנקודת העניין שרוצים לבדוק האם היא חלק מהמסלול קטנה משמעותית.

בויזואליזציה בעמוד הקודם ניתן לראות מסלולי osm **צבעוניים**, כאשר בתחילה של כל מסלול מצוייה סיכת מידע (עם האות i) הצבועה באותו הצבע כשל המסלול והמחזיקה את המספר המזהה שלו. הסיכות ה**שחורות** מציינות את נקודות העניין (בויזואליזציה הנתונה מדובר על מפלים). נקודות עניין שחורות עם סמל '**+**' הן נקודות עניין שתוייגו כקרובות למסלול אחד או יותר מאלו המופיעים בתמונה, והן מחזיקות את המספרים המזהים של המסלולים אליהם הן תוייגו כקרובות. הסיכות השחורות המסומנות ב:'**-**' הן נקודת עניין (מפלים) שמופיעות בתא השטח אך לא תוייגו כקרובות לאף אחד מהמסלולים המוצגים.

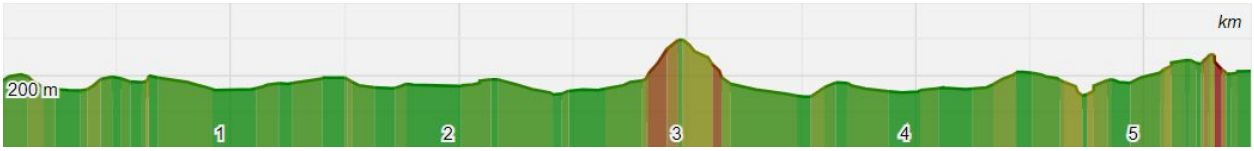
לבסוף, על מנת להעריך את אורך המסלול, סכמנו את המרחק בין כל זוג נקודות בו.

הערכת רמת הקושי של המסלול

פריט מידע נוסף שאנו משלימים עבור המסלולים הציבוריים הוא רמת הקושי שלהם. לכל מסלול OSM ציבורי, נקבע את רמת הקושי שלו להיות אחת מהאפשרויות הבאות:

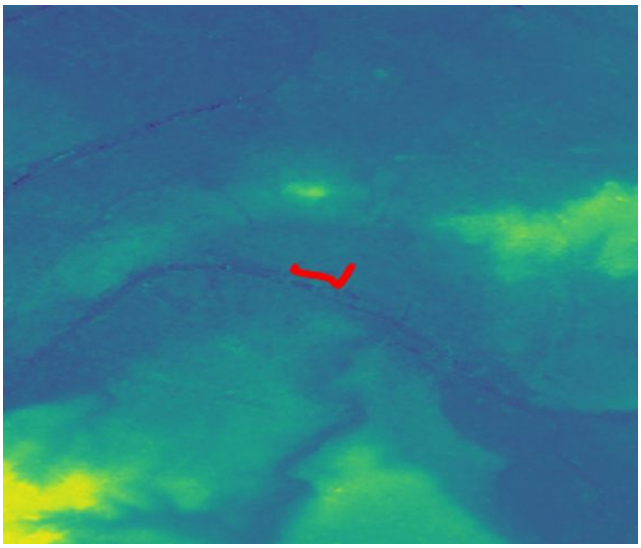
- Easy
- Intermediate
- Difficult
- Very Difficult

האסטרטגיה הכללית שלנו למציאת רמת הקושי של מסלול OSM היתה להשתמש באלגוריתם K Nearest Neighbors: החלטנו על הסיווג של מסלול OSM לאחר שהשוונו אותו למסלולים מתוייגים שכרינו מ-The Hiking Project וקבענו את התיוג שלו לפי K המסלולים הדומים ביותר שנמצאו לו (ממוצע משוקלל).
החלק המאתגר באלגוריתם, היה להבין באיזו פונקציית דמיון יש להשתמש על מנת להכריע מה הם זוגות המסלולים שסביר להניח שיחלקו את אותה רמת קושי. הנחה שלנו בנושא זה היתה שהדברים שקובעים את רמת הקושי של המסלול הם אורך המסלול, והשיפועים בו. לכן, על מנת לקבוע מי יהיו מסלולים דומים, חשבנו שכדאי להשוות בין גרפי הגבהים של מסלולים באורכים דומים.
גרף הגבהים של מסלול הוא גרף המתאר את הגובה מעל פני הים של הנקודות במסלול בכל מרחק מנקודת ההתחלה. לדוגמה, באיור הבא מופיע גרף הגבהים של מסלול כלשהו מ-The Hiking Project:



הגרף מתאר את הגובה של כל נקודה במסלול. לדוגמה, הגובה של הנקודה הראשונה במסלול הנ"ל (נמצאת 0 ק"מ מההתחלה) הוא 200 מטר מעל פני הים.
אם נייצר גרף גבהים כזה גם עבור מסלול ה-OSM הציבורי שאת הקושי שלו אנחנו רוצים לגלות, נוכל לנסות למצוא K מסלולים מתוייגים (מ-The Hiking Project) שהאורך שלהם והשיפועים שלהם דומים לאורך ולשיפועים של מסלול ה-OSM שמנסים לסווג.
חשבנו שהבעיה הזו מאוד דומה לבעיית מציאת מסמכים דומים באמצעות shingles שראינו בכיתה: גם פה אנחנו רוצים להשוות בין שני "רצפים של מידע" וגם פה נרצה שרצפי מידע שמכילים תוכן דומה בסדר אחר יחשבו קרובים. במקום לקחת רצפים של אותיות בתור shingles, על מנת להשוות בין גרפי הגבהים של שני מסלולים, ה-shingles שהשתמשנו בהם היו השיפועים בגרפים. נסביר כעת על תהליך יצירת גרפי הגבהים, ה-shingles ומציאת הדמיון:

ראשית, על מנת לייצר את ה-shingles של מסלול OSM ציבורי, יש לבנות את גרף הגבהים שלו. כפי שצויין קודם, הגבהים של הנקודות המרכיבות את המסלולים הציבוריים אינם ידועים. לכן, נעזרנו בפרויקט Shuttle Radar Topography של נס"א על מנת לאחזר את הגבהים של כל נקודה, ליצר גרף גבהים עבור מסלול ה-OSM הציבורי ולהבין מה הם השיפועים שמרכיבים אותו.



המידע שקיים בפרויקט ה-SRT הוא מפת גבהים של כל העולם בחלוקה לתאים בגודל של 30 מטר על 30 מטר (במילים אחרות כל נקודה ב-dataset מחשיבה שטח בגודל זה כבעל גובה אחיד במטרים מעל פני הים). רמת הדיוק של האינפורמציה מגבילה את היכולת לאפיין מסלולים באופן איכותי שכן אנו נאלצים לדגום את המסלולים במרחקים יותר גדולים (וכן ייתכן שבתא שטח בגודל 30 על 30 יהיו שינויי גובה קיצוניים לדוגמה קצה של צוק).

<u>בויזואליזציה משמאל</u>: מסלול osm מהלובר בפריז על גבי מפת גבוהים שהתקבלה מהמידע בפרויקט SRT.

נזכיר שוב שהגבהים של הנקודות במסלולים המתוייגים ידועים, ולכן בניית גרף הגבהים שלהם לא מהווה בעיה.

לאחר שקיבלנו את גרף הגבהים של כל המסלולים, ניצור את ה-shingles שלהם בשני שלבים. תחילה נחשב את השיפועים של גרף הגבהים בכל מקטע באורך 0.25 ק"מ, ונחזיר רשימת שיפועים גולמיים במעלות. את השיפועים הללו נעגל כלפי מטה ונמפה לערכים בין 0 ל-19 (כל 10 מעלות מופו למספר בין 0 ל-19).

המוטיבציה למיפוי השיפועים בעזרת 19 ערכים בלבד היא להקל על החשבת שני סטי shingles של מסלולים כדומים. מכיוון שהשתמשנו ב-Jaccard Index למציאת דמיון, נוצר מצב בו שני סטי שינגלים שהשיפועים שלהם דומים במציאות אינם דומים בכלל לפי דמיון Jaccard. לדוגמה: נניח שלא היינו ממפים את מעלות השיפועים ל-19 ערכים והיינו משתמשים במעלות השיפועים כפי שהן. נניח שסט ה-shingles של מסלול א' הוא:

{27, 31, 36}

וסט ה-shingles של מסלול ב' הוא:

{35, 32, 28}

אף על פי שהשיפועים של שני המסלולים דומים, וסביר להניח שהם משקפים את אותה רמת קושי, דמיון Jaccard שלהם יהיה 0.

לאחר עיבוד השיפועים הגולמיים, כתלות בפרמטר אורך נבנה את ה-shingles השונים על סמך רצפי שיפועים (כך למשל shingle באורך 2 שיפועים מיוצג על ידי המספר 1315 שאומר שהיה בגרף הגבהים של המסלול שיפוע שמופה ל-13 ואחריו שיפוע שמופה ל-15).

תהליך יצירת ה-shingles נעשה בזמן הריצה עבור מסלולים רלוונטים אך על מנת לשפר את זמני הריצה התוצאות נשמרות מקומית (גם לריצה הנוכחית וגם לריצות הבאות) כדי לא לחשב את השינגלים מחדש בכל פעם.

כעת, לאחר שהסברנו איך מתרגמים מסלולים לסט shingles, ואיך יוצרים גרף גבהים עבור מסלול ציבורי, נסכם את תהליך מציאת המסלולים הדומים:
בהינתן מסלול OSM שיש לתייג, נעריך את הגבהים שלו בכל נקודה, ונמיר אותו לסט shingles. את הסט שהתקבל, נשווה לסטי shingles של מסלולים מתוייגים שכרינו מ-The Hiking Project. נבחר את K המסלולים הדומים ביותר מבין המסלולים המתוייגים, ונקבע את התיוג של המסלול הציבורי באופן הבא:
ניצג כל רמת קושי אפשרית בתור מספר מ-1 עד 4, ועבור כל אחד מ-K המסלולים הדומים נכפול את המספר שמצייג את רמת הקושי שלו בערך הדמיון של המסלול המתוייג למסלול ה-OSM שיש לתייג. נסכום את המכפלות ונחלק אותן ב-K. התוצאה תהיה מספר בין 1 ל-4, אותו נעגל ונקבע בתור רמת הקושי של המסלול אותו היה צריך לתייג.

מציאת מסלול המתאים לדרישות המשתמש

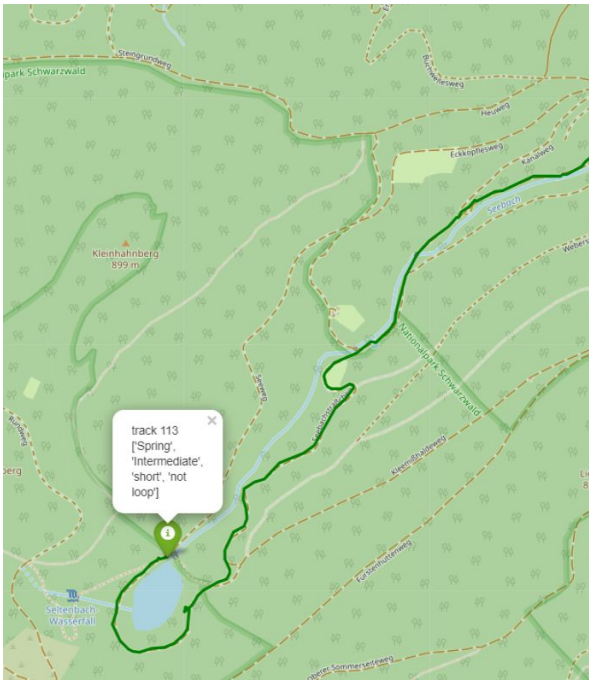
לאחר שהעשרנו את המידע על המסלולים הציבוריים שכרינו, שמרנו תמצות של המידע במספר קבצי JSON בהתאם לאזור ממנו נכרו המסלולים. כעת אפשר לקבל ממשתמש מידע על מסלול טיול אותו היה רוצה למצוא, ולחפש מסלול כזה ביעילות מבין המסלולים שמצאנו (מבלי שנצטרך לכרות שוב את כל המידע ולאפיין את המסלולים מחדש). לעת עתה המערכת תומכת רק באיזור אחד שהגדרנו מראש: baiersbronn שביער השחור בדרום גרמניה.
המערכת מקבלת מהמשתמש סדרה של הפרמטרים שמתארים את העדפותיו בתור command-line arguments:

- איזור: שם האזור מבין האזורים שאנחנו תומכים בהם.
- גבולות המסלול: ארבע קואורדינטות המתארות אזור גיאוגרפי קטן יותר מבין אזור העל עליו מתבצע החיפוש.
- אורך המסלול המבוקש (קצר, בינוני או ארוך)
- רמת הקושי המבוקשת. (קל, מתקדם או קשה)
- נקודות העניין הרצויות במסלול (נהר, מפלים, מערות וכדומה)
- צורת המסלול הרצוייה.

לאחר שהאיזור הגיאוגרפי שהמשתמש מעוניין בו ידוע, נתרגם את שאר ההעדפות שלו (קושי, אורך, נקודות עניין וצורת המסלול) לסט של tokens. לדוגמה, אם המשתמש רצה מסלול קשה, קצר, שאינו לולאה ושמכיל נקודת עניין היסטורית, נתרגם את ההעדפות שלו לסט הבא:

{hard, short, not_loop, historical significance}

נתרגם באותו אופן את המידע ששמרנו על המסלולים הציבוריים באזור הגיאוגרפי המבוקש לקבוצות tokens, ונשתמש בדמיון jaccard על מנת למצוא את המסלולים הקרובים ביותר להעדפות המשתמש. על מנת לחשב את הדמיון ביעילות ולתת מענה מהיר למשתמש, התייחסנו ל-tokens כאל shingles והשתמשנו ב-Minhashing יחד עם LSH על מנת למצוא את כל המסלולים הציבוריים שדמיון jaccard שלהם למה שהמשתמש רוצה גדול מסף מסויים. את המסלולים הדומים שנבחרו ציירנו על מפה אינטראקטיבית, יחד עם המידע שהשגנו בתהליך האפיון שלהם.



<u>בויזואליזציה משמאל</u>: תצלום של המפה האינטרקטיבית שהמודול שלנו מוסר למשתמש כפלט, כאשר בקשת המשתמש היתה מסלול שמקיים את התכונות הבאות:
{short, not loop, historical significance, Intermediate, spring}
בסמוך לעיירה baiersbronn שבגרמניה.

הערכה

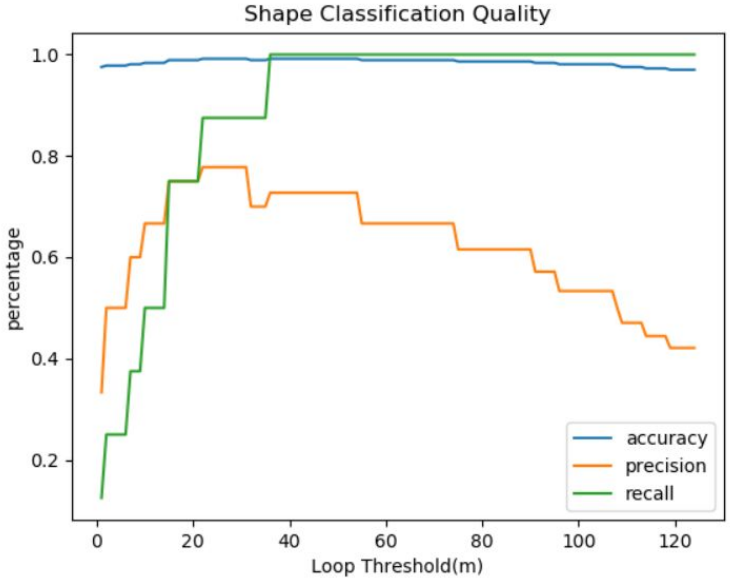
ניסוי ראשון - איכות הסיווג של צורת המסלול

בניסוי זה ננסה לבדוק עד כמה המודל שבנינו מעריך היטב את צורת המסלול (יודע להגיד האם מדובר בלולאה או לא). נשים לב שהמודל מבצע פעולת סיווג, ולכן קריטריוני ההערכה שנשתמש בהם הם ה-accuracy, ה-precision וה-recall של מסווג צורת המסלול שלנו. נגדיר שהמודל שלנו מוצלח, אם הוא מצליח להשיג accuracy, precision ו-recall של יותר מ-0.5 (טוב יותר ממסווג אקראי).

אופן ביצוע הניסוי

לכל המסלולים באתר The Hiking Project קיים "תיוג צורני", כלומר, על כל מסלול באתר נאמר האם מדובר בלולאה, או במסלול מסוג אחר (באתר קיימים שני תיוגים עבור מסלולים שאינם לולאות: point to point ו-out and back). על מנת לבצע את הניסוי, הורדנו מ-The Hiking Project מסלולים ואת התיוג הצורני שלהם. עיבדנו את המידע כך ששני התיוגים out and back ו- point to point מתמפים שניהם לתיוג אחד - not loop, שמרנו את תיוגי האמת בצד ונתנו למודל שלנו לסווג את המסלולים עם ספים שונים. נזכיר שעל מנת לסווג מסלול כלולאה או לא, המודל בודק האם המרחק בין נקודת ההתחלה לסיום של המודל קטן מ-threshold מסויים (Loop Threshold), ולכן המונח 'ספים שונים' מתייחס ל-Loop thresholds שונים.

תוצאות הניסוי



בתרשים משמאל מופיעות תוצאות הניסוי שתארנו, כאשר המידע שאנו משתמשים בו לצורך ההערכה הוא כ-400 מסלולי Hiking project שכרינו מאזור ניו-זילנד.
התוצאות מתארות את ה-accuracy, ה-precision וה-recall שהמודל שלנו השיג, כתלות ב-Loop Threshold. בניסוי זה בחרנו להתייחס ל- Loop בתור תיוג Positive ול-Not Loop כאל תיוג Negative.
נסביר את ההתנהגות המתוארת בתרשים:
<u>Recall</u>
נזכיר ש-Recall=TP/P.
ככל שמגדילים את ה-Loop Threshold יותר מסלולים מתוייגים כלולאות (Positive), וכך גם יותר מסלולים שהתיוג האמיתי שלהם הוא לולאה מתוייגים על ידי המודל כלולאות, מה שמסביר את העליה ב-Recall בין ערכים 0-40 של ה-Loop Threshold. עבור ערכי Loop Threshold גדולים מ-40 מטר, ה-Recall שווה ל-1 כי כל המסלולים שהתיוג האמיתי שלהם הוא לולאה תוייגו על ידי המודל כלולאה.

Precision

נזכיר ש-Precision=TP/TP+FP.

ניתן לראות שה-Precision של המודל שלנו עולה עד למקסימום של 0.8 ב-Loop Threshold=30, ולאחר מכן יורד. ניתן להסביר את המגמה כך: בהתחלה ב-Loop Threshold קטן, ורוב המסלולים לא מסווגים כלולאות, ולכן יש מספר קטן של TP. ככל שה-Loop Threshold עולה מספר המסלולים שמסווגים כלולאות עולה וגם מספר ה-TP של המודל עולה. לבסוף, ה-Loop Threshold גדל כך שגם הרבה מסלולים שהם לא באמת לולאות מסווגים על ידי המודל כלולאות, ומספר ה-FP עולה, מה שמקטין את ה-Precision.

Accuracy

נזכיר ש-Accuracy=(TP+TN)/(P+N)

בדומה ל-Precision, ה-Accuracy עולה עד לשיא ב-Loop Threshold=30, ויורד כאשר המודל נהיה "גס מדי" עם הגדלת יתר של ה-Loop Threshold ומתחיל לסווג הרבה מסלולים שאינם לולאות כלולאות.

ניסוי שני - איכות הסיווג לרמות קושי

קריטריון הערכה

בניסוי זה ננסה לבדוק עד כמה המודל שבנינו מעריך היטב את רמת הקושי של המסלול. גם פה המודל פועל כמסווג, ומחלק את המסלולים לארבע קטגוריות שונות:

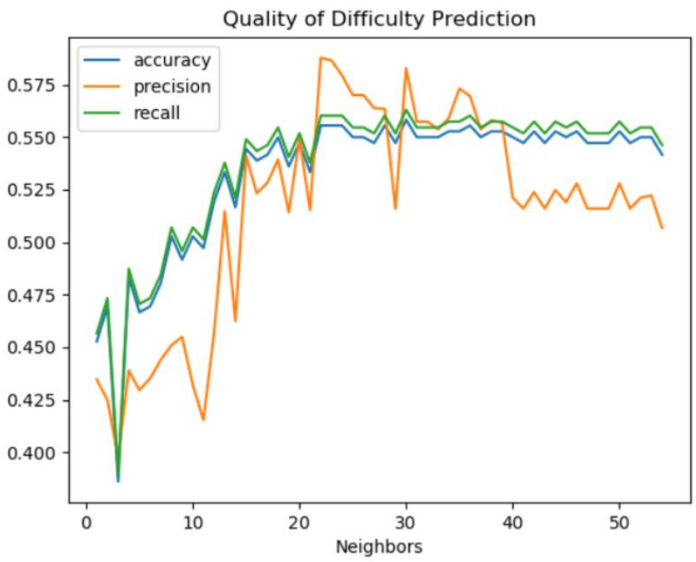
- Easy
- Intermediate
- Difficult
- Very Difficult

שוב, מכיוון שמדובר בסיווג, קריטריוני ההערכה שנשתמש בהם הם accuracy, precision ו-recall, ונגדיר שהמודל שלנו מוצלח אם הוא מצליח להשיג accuracy, precision ו-recall של יותר מ-0.25 (טוב יותר ממסווג אקראי).

אופן ביצוע הניסוי

כל המסלולים ב- The Hiking Project מחולקים לקטגוריות לפי רמת הקושי שלהם (קל, קושי בינוני, קשה וקשה מאוד). על מנת לבצע את הניסוי, הורדנו מ-The Hiking Project קבצי GPX של מסלולים (קבצים המכילים את הנקודות במסלול ומידע עליהן) שהמודל שלנו **לא** משתמש בהם על מנת לבצע פרדיקציות, יחד עם תיוג הקושי שצוות האתר נתן להם לפי גורמים שונים (האם יש מכשולים במסלול, שיפועים חדים וכדומה). את התיוגים האמיתיים שמרנו בצד, ונתנו למודל שלנו את המסלולים שהורדנו מהאתר על מנת שיתייג אותם לפי רמת הקושי שלהם.

תוצאות הניסוי



התרשים משמאל מתאר את תוצאות הניסוי שערכנו, כאשר המידע בו השתמשנו הוא:
טסט: כ-400 מסלולים מתוייגים שכרינו מ-The Hiking Project מאזור ניו-זילנד.
אימון: המודל שלנו מבצע פרדיקציות באמצעות 5000 מסלולים מתוייגים באזורים שונים בעולם, שכרינו מ-The Hiking Project.
מכיוון שמדובר ב-multi-class, התוצאות המוצגות בתרשים הן מיצוע ה-precision, ה-accuracy וה-recall של כל class שנבדק (Easy, Intermediate, Difficult, Very Difficult)
ניתן לראות שכל מדדי האיכות (ה-precision, ה-accuracy וה-recall) עולים ככל שמגדילים את מספר השכנים, אבל שהעליה נעצרת החל משלב מסויים, וב-precision אפילו מתרחשת ירידה. ניתן להסביר את התופעה על ידי כך שהחל מ-K (כמות שכנים) מסויים, לא כל השכנים שמשפיעים על בחירת מאוד דומים למסלול שרוצים לתייג, מה שגורם למודל לטעות.
כמו כן, אף על פי שהביצועים של המודל שלנו טובים ביותר מפי 2 משל מסווג אקראי, אנחנו חושבים שיהיה ניתן לשפר אותו אפילו יותר באמצעות כריית מידע נוסף מ-The Hiking Project.

קשיים ודרכי התמודדות

בגרסאות מוקדמות של הניסוי ניסינו להתייחס למסלולים שהורדנו מ-The Hiking Project בדיוק כמו אל מסלולים ציבוריים מ-OSM ולטפל בהם באותו אופן. בין היתר, התעלמנו מכך שעבור המסלולים מ-The Hiking Project הגובה של כל נקודה ידוע, וניסינו לשערך את הגבהים מחדש בעזרת פרוייקט ה-Shuttle Radar Topography, בדיוק כמו שקורה עבור מסלולי OSM. המידע המשוערך מ-Shuttle Radar Topography מכיל חוסר דיוק מסויים שפגע בתוצאות, ולכן החלטנו לוותר על הערכת הגבהים, ולהשתמש בגבהים הידועים מ-The Hiking Project לצורך ביצוע הניסוי. מצד אחד, ההחלטה הזו גורמת לניסוי להיות ממוקד רק בסיווג לרמות קושי כמו שהוא אמור להיות (ולא לבדוק גם את הערכת הגבהים וגם את הסיווג לרמת הקושי), אבל מצד שני, הוא מעריך פחות טוב את ההתנהגות על מסלולי OSM ציבוריים שהגובה שלהם לא ידוע. על מנת לפתור את זה, אפשר להשתמש במידע על מסלולים ציבוריים שהוא לא חינמי, אבל כן מכיל גבהים יותר מדוייקים של כל נקודה במסלול, כמו Strava.
בנוסף, היה לנו קושי רב לשפר את איכות החיזוי של המודל. לאחר נסיונות רבים לכוון את הפרמטרים השונים שהמודל מסתמך עליהם (מספר השיפועים ב-shingle, מספר הערכים ששיפוע מתמפה אליהם...) ראינו שהדבר שעוזר הכי הרבה הוא פשוט לתת למודל כמות גדולה יותר של מסלולים מתוייגים לביצוע פרדיקציות. לצערנו, כריית המידע נעשית על ידי crawling, שהוא תהליך איטי, והשתמשנו במה שהספקנו לכרות מהרגע שהבנו את זה עד לשלב ההגשה.

סיכום

בפרויקט ניסינו לבנות אלגוריתם המזהה מסלולי טיול מתוך מסלולי GPS ציבוריים ומעשיר את המידע עליהם. על מנת לעשות זאת, כרינו מידע ציבורי, והצלבנו אותו עם מספר מקורות מידע, ומהתוצאות הסקנו פרטים על מסלולים תוך שימוש בכלים שלמדנו בכיתה ולמידה.
התוצאות שקיבלנו היו טובות. כאשר האלגוריתם שלנו פעל כמסווג, הוא עשה זו בצורה טובה באופן ניכר מניחוש אקראי. אנחנו משערים שאפשר לשפר את יכולות הסיווג של המודל אפילו יותר על ידי הגדלת כמות המידע שמשמשת אותו לביצוע פרדיקציות.

רעיונות לעבודות המשך

1) אפיון מלא יותר של המסלולים: מציאת העונה המומלצת לטיול, אפיון הצמחייה והחי באזור המסלול, פרטים נוספים (כמו אזהרות מסע, או שעות פתיחה של אתר הטיול) וחיזוי אופי המסלול (מתאים למשפחות, רומנטי, ידידותי לכלבים).
2) קישור תמונות ציבוריות למסלולים: יש רשתות חברתיות (לדוגמה טוויטר) שהמידע בהן ציבורי ולחלקו מוצמד תיוג גאוגרפי. אפשר לנסות להתאים בין תמונות שאנשים העלו לרשת החברתית למסלולי טיול באיזורים גיאוגרפים זהים.