**חקירת הקשר בין הולך הרגל לתאוצות ההליכה שלו**

**מודל חישובי**

**מאת**

**קאי גולן, דן קוצר וגון אייל**

**עבודה זו מוגשת כעבודה בהיקף של 2 יח' כמילוי חלקי של הדרישות לקראת קבלת ציון במדע חישובי**

**עבודה זו בוצעה בהדרכת שלמה רוזנפלד**

**יוני 2016**

**תמצית**

עבודתנו עוסקת ב"למידת מכונה"- מודל חישובי המכיל קבוצת אלגוריתמים, ונשען על תשתית של רשת נוירונים המחקה את תהליך הלמידה במוח האנושי. בפועל למידת המכונה נועדה לחקות את תהליך הלמידה במוח האנושי באמצעות עיבוד של מידע רב במודל דמוי רשת הנוירונים, תוך אימון של מערכת, וכתוצאה מכך, שיפור יכולות החיזוי שלה. בעבודתנו כתבנו קטע קוד שמסוגל להבדיל בין אנשים שונים על פי אופי תנועת מכשיר טלפון שנמצא על הגב שלהם בזמן שהם הולכים. העבודה באה לחקור האם אפשר להבדיל בין אנשים שונים על פי הליכתם, ומה היא הדרך היעילה ביותר לעשות זאת. כך מצאנו בעבודתנו את ההבדלים שקיימים בין אנשים שונים בעזרת מאפייני הליכה שונים שאותם הכנו מנתוני התאוצות של הסלולרי שקובע על גבם. בעזרת הקוד הצלחנו למצוא קשר בין חלק מהמאפיינים לבין ההולך רגל עם המכשיר על גבו.

**תוכן ענינים**

**מבוא**

**תיאוריה**

* חיישן התאוצה
* למידת המחשב
* למידת המוח
* מאפייני ההליכה

**איסוף נתונים**

**האלגוריתם**

* הכנת מאגרי הנתונים
* שלבי בלמידה
* החלק המעשי
* השפעת הפיצ'רים השונים
* הרחבת התפלגות האנשים

**שימושים אופציונליים**

* זיהוי גניבת מכשיר
* זיהוי הליכה מזיקה לגוף
* זיהוי שכרות והשפעת סמים

**נספחים וביבליוגרפיה**

* נספחים
* ביבליוגרפיה

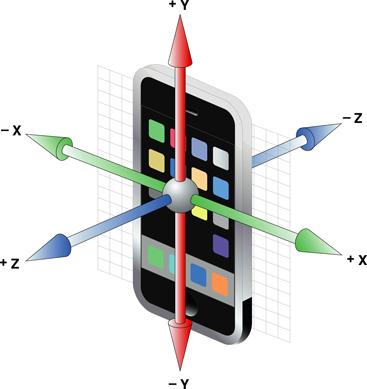
**מבוא**

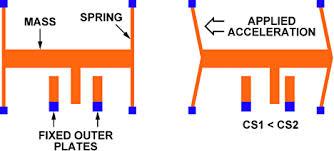
שאלת המחקר הראשונה בעבודתינו הייתה האם המחשב יכול, בעזרת למידת מכונה, להבדיל בין אנשים שונים, על פי נתוני התאוצה ממכשיר טלפון על גבם. השאלה השנייה שעבודתינו מתמקדת בה היא מה היא השיטה היעילה ביותר לתהליך למידה זה.

העבודה שלנו פותחת בחלק התיאורתי בו חקרנו את דרך המדידה של חיישן התאוצה בטלפון, את המאפיינים שניתן להפיק מהליכה, את מבנה רשת הנוירונים במוח, את הנושא למידת מכונה בכלל ורגרסייה לינארית ולמידת רשת נוירונים ממוחשבת (כולל הסבר על פונקציית היסגמואיד) בפרט. עבודתינו כוללת גם הסבר על שיטת איסוף ועיבוד הנתונים והכנתם לתהליך הלמידה. לאחר מכן הסברנו איך הרצנו על המחשב את תהליך הלמידה והשוונו בין שיטות הלמידה השונות במקרים שונים. המקרים כללו התעלמות מחלק ממאפייני ההליכה השונים על מנת למצוא את השפעתם על תהליך הלמידה ולבסוף בדקנו את יעילות האלגוריתמים שפיתחנו עבור מספר רב יותר של נתונים ואפשרויות סיווג רבות יותר- הוספנו נתונים שלאנשים חדשים.

**תיאוריה**

**חיישן התאוצה**

מכשירי טלפון חכמים ומכשירי טכנולוגיה אחרים יודעים לאן הם פונים במרחב באמצעות חיישן אשר מותקן בתוכם, שמודד את כוח התאוצה (accelerometers). הגדרת תאוצה רגעית כידוע היא: . מערכת מדי התאוצה היא מערכת בגודל קטן ביותר של כ- סנטימטר ובנויה כך שתוכל למדוד את התאוצה הלינארית בשלושה צירים מאונכים זה לזה (ניתן לראות בתרשים). חלק אלקטרו-מכני זה בטלפון מודד את גודל כוח התאוצה שפועל על המכשיר ואת כיוונו. כוח התאוצה יכול להיות כוח סטטי כמו כוח הכובד הרציף, או כמו בניסוי שלנו, כוח תאוצה דינמי שנובע מתנועה ורטט.

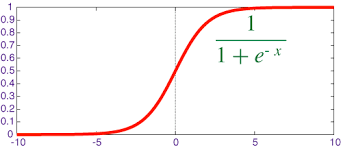
מדי התאוצה מתבססים על האפקט פיזואלקטרי בעבודתם: מבנים של קריסטלים מיקרוסקופיים שנמתחים מכוחות התאוצה הפועלים עליהם. כאשר המערכת (מכשיר הסלולר) מאיצה לכיוון אחד, פועל על הקריסטליים כוח מדומה לכיוון השני מפני שהם שואפים להתמיד (להישאר במנוחה או במהירות קבוע), וכתוצאה מכך הם נמתחים לכיוון המנוגד לכיוון תאוצת המכשיר. הקריסטליים האלו יוצרים מתח חשמלי כתוצאה מהלחץ המכאני שפועל עליהם כאשר הם נמתחים, ומד התאוצה מפרש אותו כך שהוא יכול לזהות את גודל התאוצה בכל אחד מהצירים על פי הפרש הפוטנציאלים החשמליים- המתח. המתח נמדד בכל ציר בנפרד בין שני לוחיות מתכת מקובעות לסלולרי (ניתן לראות בתרשים).

**למידת המחשב**

למידת מכונה (Machine Learning)- פיתוח אלגוריתמים המיועדים לאפשר למחשב ללמוד מתוך מספר רב של דוגמאות נתונים, בתחומים בהם אי אפשר לפתור את הבעיות בעזרת התכנות הקלאסי. למידת מכונה מאפשרת לזהות דפוסים חוזרים, חלקם מופשטים לחלוטין, מתוך הכמויות הגדולות של הנתונים שהוזן לסביבת למידת המכונה. המטרה שלשמה השתמשנו בלמידת המכונה, בשפת התכנות פייטון, היא טיפול בנתוני מאפייני ההליכה, ומציאת קשר בין הבן אדם ההולך לבין מאפייני ההליכה שלו. בעזרת שיטה זו אנו יכולנו לקבל חיזוי לגבי קשר זה ולסווג מאפייני הליכה לאדם מסוים. במהלך עבודתנו השתמשנו בשני סוגים שונים של למידת מכונה: רגרסיה לינארית ורשת עצבים מלכותית באמצעות פייביריין. השתמשנו בשתי דרכים שונות בשביל למצוא איזו אחת מהן מתאימה יותר לעבודתנו ותאפשר לנו לחזות תיוגים של תאוצות לאנשים בצורה מדויקת יותר.

רגרסיה לינארית היא שיטת למידה מונחית שהשתמשנו בה בעבודתנו בשביל למצוא את הקשר בין מאפייני ההליכה לבין האיש ההולך. שיטה זו היא טכניקת למידת מכונה חישובית שמייצרת פונקציה (היפותזה) מתוך קבוצות שונות של נתונים. ישנם משקלים שמתארים את הקשר בין הקלט לבין הפלט. תחילה מפעילים את הרגרסיה הלינארית על נתוני למידה. עליהם להכיל התפלגות של כל קבוצות הנתונים הקיימות במערכת. פעולה זאת נותנת לנו פרמטריים ראשוניים לקשר בין הנתונים השונים לקבוצת ההשתייכות שלהם. לאחר מכן הפעולה מתבצעת על נתוני אימון, שגם הם צריכים להכיל את ההתפלגויות השונות. בעבודה שלנו ההתפלגות היא בין האנשים השונים עבורם קיבלנו מאפייני הליכה שונים. באמצעות שיטה חזינו תיוג של תאוצות שונות להולכי רגל שונים. ככל שהרצנו את פונקציית הרגרסיה הלינארית על מספר נתונים רב יותר ככה אחוז השגיאה שלנו קטן ורמת הדיוק השתפרה. במקרה זה היה יותר נוח, וזה מה שעשינו, לאמן ריגרסייה ליניארית עבור כל אחד מהנבדקים. כאשר מריצים את כולן על נתוני המבחן מקבלים שגיאה עבור כל אחת מהן ואת השגירות האלו מעבדים לשגיאה סופית אחת.

שיטה נוספת שבה השתמשנו בעבודתנו היא רשת נוירונים ממוחשבת מוכנה של החבילה "פייבריין" של פייטון אשר מהווה רשת עצבים מלכותית. השתמשנו בשיטה זו כי היא מציאה לנו גמישות עבודה והיא קלה לשימוש, אבל עדיין מהווה אלגוריתם יעיל עבור מטלות למידת מכונה במגוון רחב של סביבות בחינה מוגדרות מראש, להשוואת אלגוריתמים. היא מאפשרת לנו לעבוד עם מספר גדול של נתונים ונבדקים וכך לבדוק אפשרויות למידה שונות ביעילות. רשת נוירונים מאופיינת ע"י אופן חיבור הנוירונים, משקלי החיבור ביניהם ופונקציית האקטיבציה (במקרה שלנו היא סיגמואיד). רשת עצבים מלכותית, בה השתמשנו באמצעות הפייבריין, היא מודל שמדמה תהליכיים שונים במוח ובעזרתו הגענו לפרמטרים השונים שמבדילים בין קבוצות המאפיינים השונות, אצלנו- האנשים השונים.

רשת הנוירונים הממוחשבת כוללת מספר רב של מקומות קלט ופלט. לעומת שיטת הרגרסיה הלינארית, בין שכבת הקלט לפלט ישנן שכבות ביניים נסתרות אשר מוצאת את הקשרים בין הקלט לפלט. צורת הקישור ביניהם כוללת מידע על חוזק הקשר באופן דומה לרשת הנוירונים במוח. לרשת אצלנו מספר הנוירונים בשכבת הכניסה הוא כמספר הפיצ'רים שאנו מעבירים לה בכל הרצה. את מספר הנוירונים בשכבת הביניים הנסתרת שינינו בכל פעם על מנת לבדוק מה הוא המבנה בו הרשת תגיע לאחוז השגיאה המינימלי- החיזוי המדויק ביותר. 

בהגדרת הפלט בתוכנית שלנו השתמשנו בפונקציה מצורת סיגמואיד: (ניתן לראות את הגרף שלה).

כידוע: לכן:. יתרונה של פונקציה זו היא שהיא מספקת לנו מידע בנוגע לסיכויי הוודאות לכך שהיא צדקה. לדוגמה: אם היא תחזיר 1 היא יודעת בוודאות מיהו הולך הרגל בניסוי שלנו, ואם היא מחזירה 0.5 ישנה הסתברות של 50% ,להערכתה, שמדובר בהולך רגל מסוים. תהליך הלמידה שלה מתבצע באמצעות שיוניי הפרמטרים ברשת וערכי הסף של הנוירונים הממוחשבים במערכת וכתוצאה מכך חוזקי הקשרים השונים משתנים.

כאשר מפעילים את הרשת על ערכי כניסה מסוימים כל אחד מנוירוני המוצא משגר ערך בין 0 ל1 לפי פונקציית הסיגמואיד ואז הנוירון שהיה הכי קרוב לאחד מקבל את הערך כחיובי. כמו כן ניתן לקבוע ערך סף מסוים לרשת שאם אף אחד מהנוירונים לא יתן מספר גבוה ממנו היא לא תבצע זיהוי כלל, כלומר הרשת לא חייבת להצביע על מישהו אם היא חושבת שהסיכויים נמוכים מדי.

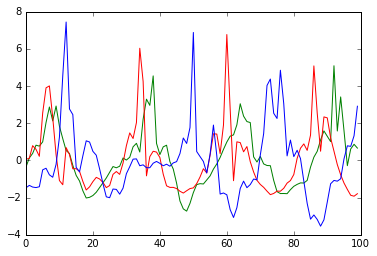
**למידת המוח**

המוח בנוי מכ-100 מיליארד תאי עצב, שפעולתם המשותפת מאפשרת למידה מורכבת. כל אחד מהם מחובר ל-1,000 עד 10,000 תאי עצב אחרים. במוח יש מערכת דומה לרשת הנוירונים הממוחשבת באופן הקישור בין הקלט לפלט. תהליכי הלמידה במוח מערבים שינויים פיזיולוגיים בקישורים הסינפטיים שבתוכו. הלמידה גורמת לסינפסות באזורים שונים להתחזק או להיחלש, ומביאה גם ליצירה של סינפסות חדשות או להתנוונות ומוות של סינפסות אחרות. סימפונה היא איזור המפגש של שני תאי עצב שונים (או תא עצב עם שריר או בלוטה) ובה מתרחשות תגובות כימיות הגורמות להעברת אותות בינהם. הדנדריטים בנוירונים הם החלקים שמקבלים מידע מנוירונים אחרים שנשלחו לסינפסה, ממירים אותם לאותות חשמליים ומעבירים אותם לגוף התא. כאשר התא מקבל גירוי חזק מספיק (מתח הסף הוא כ-40 מיליוולט) הוא מייצר פוטנציאל פעולה, דחף עצבי שנע לאורך האקסלון לדנדריטים של תאים אחרים ספציפיים וככה ישנה תגובת שרשרת של אותות שמועברים במוח בין תאי העצב. מהירות העברת האינפורמציה במוח נעה בין 0.5 ל- 120 מטר בשנייה.

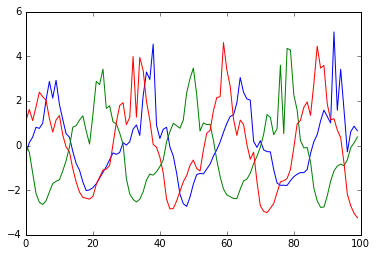
ככל שנוירון במוח מקבל יותר אותות חשמליים ככה הוא יוציא יותר אותות חשמליים. האות החשמלי מנוירוני החישה מועבר לנוירונים נוספים הנקראים נוירוני ביניים, האחראיים לתהליך הלמידה. לעומת למידת המכונה בעבודתנו במוח ישנם שכבות ביניים רבות של נוירונים, ולא אחת בלבד. הפלט של כל המערכת הזאת מועבר לנוירונים מסוג אחר הנקראים נוירוני תנועה ובהם האקסונים מחוברים ישירות לשרירים. שלושת סוגי הנוירונים האלו מהווים את שכבת הקלט שכבת הביניים ושכבת הפלט של רשת הנוירונים. באופן דומה למודל רשת העצבים הממוחשבת ככל שהתקשורת בין שני תאים היא בתדירות גדולה יותר אז הקשר ביניהם מתחזק ולהפך, עד לקבלת הרשת עם היעילות הטובה ביותר.

**מאפייני הליכה**

לאחר חיפוש עבודות קשורות לנושא שלנו מצאנו את המאמר " Activity Recognition using Cell Phone Accelerometers" של ג'ניפר ר' קוואפיץ, גריי מ' וואיס וסמואל א' מור שנכתב על ניסוי שבדק את הקשר בין תאוצות מכשיר טלפון מקובע לרגל לבין סוג הפעילות שהבן אדם מבצע (ריצה, הליכה וכולי...). תרם בדיקת מאמרים בנושא זה היה ברור לנו כי לא ניתן להבדיל בין האנשים באופן ישיר מהתאוצות של ההליכה שלהם, באמצעות למידת מכונה, מכיוון שהגדלים של התאוצות מחזוריים (הגודל שלהם תלוי במיקומם לעומת תאוצות אחרות) ולכן אין קשר בין הגודל שלהם לבין הבין אדם ההולך. בעקבות זאת חשבנו על מאפיינים אפשריים שניתן לאפיין איתם את הליכת האנשים. ממאמר זה קיבלנו את ההשערה לאפיין את ההליכה בתאוצה ממוצעת, תאוצה מקסימלית, תאוצה מינימלית, היסטוגרמה (התפלגות) של גדלי התאוצות וכל זה עבור כל ציר בנפרד. אומנם הם חיפשו את הקשר בין מאפייני ההליכה לסוג הפעילות ואנחנו את הקשר עם ההולך רגל, אבל עם מאפיינים אלו אפשר גם להבחין בהבדלים הקטנים בין תאוצת ההליכה של אנשים שונים במידה מסויימת.

בניסוי שלנו קיבלנו טבלת תאוצות בכל ציר של הטלפון בזמן ההליכה שלנו כשהוא היה על גבנו. לאחר התבוננות בתוצאות בצורה גרפית החלטנו לייצר מתוצאות תאוצה אלו, את מאפייני ההליכה הרצויים שיבדילו בין הליכתו של אדם א' להליכתו של אדם ב'. המאפיין שייצרנו בנוסף לאילו שעשינו כמו במאמר אותו סקרנו הוא מספר שינויי סימן התאוצה (מחיובי לשלילי ולהפך). את כל המאפיינים האלו עשינו עבור כל ציר בנפרד.

עוד לפני שהתחלנו לעבד את הנתונים בדקנו קודם כל האם באמת ניתן לראות הבדל בהליכה של הנבדקים. במקרה זה לקחנו שלושה נבדקים (שלושתנו) ושרטטנו גרף של התאוצה בציר ה-X כתלות בזמן של שלושתנו. מהגרף הזה למדנו שני דברים חשובים מאוד: הראשון הוא שאפשר לראות בבירור הבדלים משמעותים בין הגרפים של כל אחד מהנבדקים, משמע יהיה אפשר לסווג אותם ולאחר מכן תתאפשר לרשת היכולת ללמוד לזהות אותם. הדבר השני שהסקנו מהגרף הזה הוא שלא רק שלכל אחד מאיתנו היה דפוס הליכה שונה , אלא שהוא גם מחזורי- חוזר על עצמו בדפוס קבוע.

כמו כן שרטטנו גרף של אחד מהנבדקים של שלושת הצירים ביחד וגם פה רואים שדפוס ההליכה שלו היה מחזורי כמעט לחלוטין בשלושת הצירים.

על מנת למצוא את התאוצה הממוצעת בכל ציר עבור כל אחד מהאנשים שהלכו עם הטלפון השתמשנו בנוסחה הפשוטה הזו: כאשר פרק הזמן בו השתמשנו הוא 2 שניות ותדירות המדידות לשנייה היא 50 הרץ.

כמו כן בדקנו לכל ציר, לכל קבוצת תוצאות, את התאוצה המינימאלית והמקסימאלית. כאן ניתן לראות 3 קבוצות דגימה (הסבר על קבוצת דגימה בפרק האלגוריתם) של אותו נבדק ורואים שהמינימום והמקסימום די קרובים, משמע ההליכה קבועה.

([-1.155, -3.233, -2.525]) מינימום 1  
([ 1.955, 4.614, 3.894]) מקסימום 1

([-2.153, -2.77 , -2.027]) מינימום 2

([ 1.863, 4.353, 8.151]) מקסימום 2

([-1.524, -2.727, -3.046])מינומום 3   
([ 2.098, 5.089, 3.157])מקסימום 3

מאפיין נוסף שהשתמשנו בו על מנת למיין את התוצאות לפי האדם שהלך עם הטלפון הוא היסטוגרמה של תוצאות התאוצה עבור כל 100 תוצאות תאוצה (2 שניות של הליכה). ההיסטוגרמה בודקת כמה ערכים מתאימים לתחומים מסוימים. חשוב לציין שעשינו היסטוגרמה על כל ציר בנפרד ומכיוון שרק אופן הפילוג הוא שמעניין אותנו במציאת הקשר בין ההולך עם הטלפון לבין תאוצותיו של המכשיר ולא גבולות הפילוג השתמשנו רק בו. המאפיינים הייחודיים של גבולות הפילוג מוכלים כבר באופייני התאוצה המקסימלית והמינימלית שמצאנו בעזרת קטע הקוד. לאחר שראינו את ההפרש בין נתוני התאוצות לבין הממוצע שלהם יכולנו להעריך ש-10 הוא מספר המחלקות היעיל ביותר בהיסטוגרמה בעבור מציאת הקשר שלנו, משום שמספר מחלקות גדול יותר היה מאט את פעולת למידת המכונה כי היא הייתה צריכה לעבד יותר נתונים, ובמספר קטן מ-10 לא היינו מוצאים הבדלים במאפיין ההיסטוגרמה בעבור אנשים שונים.

מאפיין נוסף שהשתמשנו בו באלגוריתם הוא מספר הפעמים שהתאוצה משנה סימן עבור כל ציר, כשהיא עוברת מתאוצה חיובית לשלילית או להפך. כל פעם שהתאוצה משנה סימן היא משנה את כיוונה באותו הציר. בשביל למצוא מתי התאוצה משנה סימן הכפלנו את שתי תוצאות התאוצה האחרונות באותו הציר (כאשר כל פעם זזים מיקום אחד בתוצאות התאוצה) ובדקנו עם ערך זה גדול מאפס או קטן מאפס. אם ערך זה היה גדול מאפס שני תוצאות התאוצה היו בעלות אותו סימן והתאוצה לא שינתה את כיוונה באותו הציר, ואם הערך היה קטן מאפס תוצאות התאוצה היו בסימנים שונים והתאוצה שינתה את כיוונה באותו הציר.

**איסוף הנתונים**

את הניסוי שלנו בחרנו לעשות בנושא שלנתונים שלו אין גישה חופשית במרשתת לכן נזקקנו לאסוף את הנתונים בעצמנו. הלכנו כאשר מכשיר סלולר (Nexus 5) עם מד תאוצה היה מקובע לנו לנקודה קבוע בגב באמצעות חגורה. השתמשנו במכשיר זה מכיוון שבאמצעותו ניתן לגשת באופן חינמי לנתונים של חיישן כוח התאוצה. אנחנו הלכנו כ-50 שניות עם הטלפון על גבינו באופן הזה. במכשיר התקנו את האפליקציה החינמית "Accelerometer Analyzer", תוכנה המאפשרת לעקוב בזמן אמת אחר נתוני התאוצה של המכשיר סלולר באמצעות חיישן התאוצה שנמצא בו. ניסינו מספר תדירויות שונות למדידת התאוצה ומצאנו ש-50 מדידות בשנייה זו תדירות מספיקה על מנת לנתח את תנועת ההליכה ובשביל להבדיל בין הולך אחד להולך אחר. האפליקציה אספה את תוצאות ההליכה בשלושת הצירים לאחר התעלמות מתאוצת הכובד. הסרנו את הוריית תאוצת הכובד באפליקציה מכיוון שהיא לא תורמת למציאת ההבדלים בין הליכתם של אנשים שונים וככה היה לנו קל יותר לנתח את התוצאות בגלל סדרי גודל- גודל תאוצת הכובד גדול יותר מתאוצות אחרות בעת ההליכה. את הנתונים שמרנו בתור קובצי טקסט (.txt) והעברנו אותם למחשב לתיקייה של ה-Python באמצעות האימייל, כך שהם היו מוכנים לשימוש ב-Jupiter (סביבת העבודה שבחרנו).

**האלגוריתם**

**הכנת מאגרי הנתונים**

הקוד שלנו בנוי על מנת לעבד את הנתונים שמתקבלים מהטלפון ולאחר מכן לנסות ללמוד אותם ולהגיע לזיהוי הטוב ביותר האפשרי. קטע הקוד שלנו הוא מורכב ומכיל פרטים ושינויים רבים. אם היינו כותבים את הכל כמקשה אחד הוא היה מסורבל וגדול והיה לא נוח להבנה ושימוש. במקום זה החלטנו לעבוד עם שיטת פעולה שנקראת "פונקציות עזר"- כתיבת קוד בעזרת פונקציות רבות שבנינו, ישנן פונקציות מסוגים שונים ובתפקידים שונים, למשל, ייבוא וניקוי של הנתונים או חישוב התאוצה הממוצעת. בסוף השתמשנו בפונקציה אחת שמבוססת על פעולת שאר הפונקציות. היתרון בשיטה זו היא שכך הקוד יותר מובן ואם קיימת בו תקלה צריך לשנות רק את החלק המסוים בו היא קיימת. כמו כן את האלגוריתם שלנו פיצלנו לכמה קבצים שונים על מנת שבעתיד יהיה ניתן לראות בדיוק כיצד יצרנו או בדקנו כל קובץ נתונים או תוצאות מסוימות.

תחילה יצרנו קובץ קוד שמטרתו להכין את הנתונים לעבודה. הוא מקבל אליו את הנתונים שנאספו מהטלפון של כל נבדק ומנקה מהם את התוצאות הראשונות והאחרונות בשביל שלא ייכנסו לנתונים מדידות מתחילת וסוף התנועה בהם האדם התחיל ללכת או עצר וזה ישפיע על התוצאות. לאחר מכן לכל אחד מן המאפיינים (הפיצ'רים ) שבחרנו להשתמש בהם ישנה פונקצייה שמטרתה לקבל רשימה של מדידות ולהחזיר את אותו פיצ'ר.חשוב לציין שכל אחת מן הפונקציות האלו מתוכנתת כך שהיא מחזירה רשימה של הפיצ'ר שלה **עבור כל ציר**. בשלב זה פיצלנו את המדידות של כל נבדק לקבוצות דגימה של 100 (שתי שניות של הליכה) כדי שיהיו לנו יותר דוגמאות ללמידה. בדרך זו הפכנו רשימה אחת גדולה של נתוני תאוצות להרבה רשימות קצרות, במקום אחת גדולה, מהן יכולנו להפיק מספר מאפיינים עבור כל איש. בנינו פונקציה נוספת שמקבלת רשימה והיא מפעילה את פונקציות העזר שמחלצות את הפיצ'רים ומחברת אותם לרשימה אחת שתשמש בעתיד את ערכי הכניסה לרשת הנוירונית או תשמש עבור הלוגיסטיק ריגרשן. הפונקציה המסכמת שלנו משתמשת בפונקציה הקודמת לגבי כל אחת מן הרשימות שפיצלנו ממדידות הנבדקים והיא מחזירה רשימה של רשימות שכל אחת מהן היא ערכי כניסה (הפיצרים) וערכי מטרה אותם מעבירים לפונקציה כאחד מן הפרמטרים. לבסוף השתמשנו בפונקציה המסכמת עבור כל אחד מהנבדקים ואיחדנו את כל הרשימות הללו לבסיס נתונים מוכן לשימוש ללמידה.

**שלב הלמידה**

לאחר שבנינו מאגר נתונים ניגשנו לשלב העיקרי של החקר שהוא לנסות להגיע לזיהוי היעיל ביותר של האנשים השונים. ייבאנו פונקציות בנויות וספריות של פייטון שאנו צריכים בשביל העבודה וכמו כן את קובץ הנתונים עליו אנחנו נעבוד. את מאגר התנונים חילקנו לערכי כניסה ולערכי מטרה וכל אחד מן המערכים הללו פיצלנו באופן הבא: שני שליש מהדוגמאות יופנו לאימון ואילו השליש הנותר לבדיקה של הרשתות או הלוגיסטיק ריגרשן. חשוב לציין שהחלוקה מתבצעת באופן רנדומלי כך שבכל הרצה של הקוד החלוקה שונה, מה שמשפיע על התוצאות שבכל הרצה משתנות מעט.

קודם כל בדקנו מה קורה בלוגיסטיק ריגרשן- יצרנו שתי פונקציות לשם כך. הראשונה בונה לוגטיסטיק ריגרשן לה מוכנסים ערכי הכניסה שהיו החלק של הלמידה מהנתונים וערכי המטרה שהיא אמורה לזהות. את פעולה זו ביצענו באמצעות הפונקציה המובנית של פייטון linear\_model.LogisticRegression שאחראית על בניית הרגרשן ובנוסף fit שבעזרתה מעבירים לרגרשן את הערכים-כניסה ומטרה. הפונקציה השניה משתמשת בנתונים שנשארו לבדיקה ומנסה לזהות את הנבדק אותה היא אמורה ע"י הפעולה predict, לאחר מכן היא מחשבת את הטעות היחסית שלה ומדפיסה אותה. השתמשנו בשתי הפונקציות האלו לגבי כל אחד מהנבדקים וכך קיבלנו את השגיאות של הלוגינטיק ריגרשן בכל אחד מן המקרים. השגיאות מורכבות משלוש שגיאות וניתנות לנו בצורת מערך באופן הבא [0,17.6,3.8] המספרים שהודפסו לא נראים ככה אלא עם הרבה ספרות לאחר הנקודה ופה זה רק אומדן. כל אחד מן המספרים במערך מציין את אחוז השגיאה של כל נבדק, בהתאם. אפשר לקבל טעות סופית ע"י ממוצע בין שלושת ערכי הטעות של כל הלוגיסטיק רגרשנים אבל זה לא בדיוק נכון מכיוון שמספר הדוגמאות שניתן עבור כל אחד מהנבדקים היה שונה במקצת, הן בנתוני הלמידה והן בנתוני המבחן.

החלק הבא בקוד שלנו עוסק ברשת נוירונים מדומה אותה ננסה ללמד באופנים שונים לזהות בין הנבדקים לפי הפיצרים שלהם שחילצנו מנתוני ההליכה. בחלק זה 3 קבוצות נתונים; הקבוצה הראשונה היא קבוצת נתוני הלמידה שהכנו מקודם שמהווה שני שליש מהדוגמאות. קבוצה זו תשמש את הרשת לשלב של הלמידה. את הקבוצה השנייה ממקודם, שליש מהדוגמרות, אנחנו מפצלים עתה לשני חצאים-אחד מהם ישמש לבדיקה של זיהוי הרשת על מנת למצוא את השגיאה שלה ולהקטין אותה, ואילו החצי השני ישמש לבדיקה סופית של הרשת שתמצא הטובה ביותר.

בקטע הקוד הבא הגדרנו את הקבוצות בתור בסיסי נתונים של פייטון ובנינו רשת נוירונים בעל שכבות כניסה, נסתרת ומוצא בגודל המתאים לכל ניסיון. עשינו זאת בעזרת הפונקצייה המובנית של פייטון buildNetwork. לאחר מכן הגדרנו BackpropTrainer שהיא פונקציה שמטרתה לאמן רשת מסוימת שמגדירים להעל בסיס נתונים מסוים שגם כן מגדירים לה.

את שלב הלמידה עשינו בעזרת הפונקציה trainUntilConvergence שהיא אחת מהפונקציות שמאמן הרשת יכול לבצע. הגדרנו לפונקציה פרופורציית חלוקה שעל פיו היא מחלקת את הנתונים שמכניסים לה לקבוצת אימון וקבוצת בדיקה. הפלט של פונקציה זו הינו שתי רשימות של שגיאות: האחת על נתוני האימון והשניה על נתוני הבדיקה. לפונקציה זו שתי בעיות עיקריות: הראשונה היא מינימום מקומי- הפונקציה יכולה לחשוב שהגיעה לשגיאה מינימלית אבל בעצם אם תעשה עוד צעדים היא תגיע לנקודת מינימום יותר נמוכה. בשביל להתמודד אם בעיה זו הגדרנו משתנה שנקרא continueEpoch שמשמעותו היא שברגע שהפונקציה מגיעה כביכול למינימום היא תמשיך עוד מספר צעדים כדי לודא שזהו המינימום המוחלט. הבעיה השניה היא שאם ממשיכים יותר מדי צעדים השגיאה של הרשת מתחילה לגדול. גם בעיה זו פתרנו בעזרת אותו משתנה. כמו כן הגדרנו מספר מחזורים מקסימלי שעל הפונקציה לבצע.

בשלב הבא השתמשנו בactivateOnDataset, פונקציה של רשת נוירונים, שמנסה לזהות בסיס נתונים שנותנים לה. הפלט שלה הינו רשימת מערכים בצורת המטרה, שהוגדרה לרשת, לפי מה שהיא זיהתה. על מנת לחשב את שגיאתה באחוזים בדקנו כמה פעמים היא טעתה ואת מספר זה חילקנו במספר הדגימות.

עכשיו הגיע שלב המחקר האמיתי- בנינו לולאה שמטרתה לבדוק מה המבנה בו ישנה שגיאה מינימלית. בכל כניסה ללולאה מוגדרת רשת נוירונית בעלת שכבה נסתרת בגודל שונה, וחוזרים על הליך האימון כמו מקודם. בכל הרצה בה השגיאה קטנה מהשגיאה עד כה הרשת הטובה ביותר(הקודמת) מוחלפת בחדשה וכמו כן משתנה שמספרו כמספר הנוירונים בשכבה הנסתרת של הרשת הטובה ביותר מתעדכן.

לאחר שקטע הקוד אחרון סיים והגיע לרשת הטובה ביותר בהרצה הזאת הדפסנו את המשתנה שמציין את מספר הנוירונים של הרשת הטובה ביותר שנמצאה וכך גילינו מה המבנה בטוב ביותר של הרשת. עכשיו הגיע השלב הקריטי ביותר והוא בדיקת הרשת הטובה ביותר על נתוני הבחינה הסופיים (בהם לא נגענו עד כה). עשינו זאת באותו אופן שבוא חישבנו את טעות הרשת בקטעים הקודמים רק על בסיס הנתונים שהיא לא פגשה עדיין וכך קילנו את אחוז השגיאה המינימלי של הרשת. השתמשנו בחבילה pickle כדי לשמור את הרשת בתור אובייקט על מנת שנוכל בעתיד לחזור ולהשתמש בה.

**החלק המעשי**

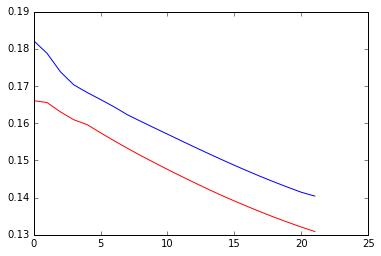
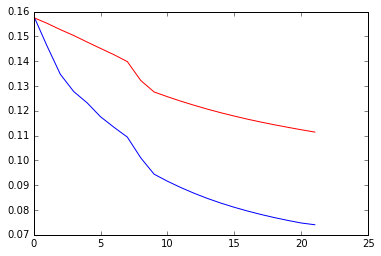
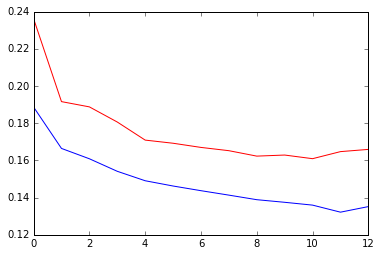
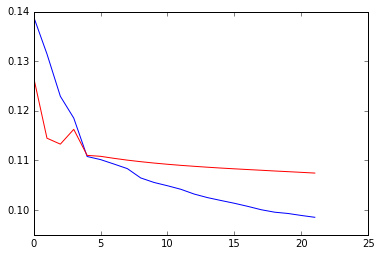
בחלק המעשי של הפרויקט בעצם בדקנו כיצד תפעל למידת מכונה על גבי הפיצ'רים שחילצנו מתוצאות ההליכה של הנבדקים. בדקנו מה אחוז ההצלחה הגבוהה ביותר שנצליח להגיע אליו בדרכים שונות וכמו כן חקרנו מקרים שונים על שתי השיטות שבחרנו- רשת נוירונים מדומה ולוגיסטיק רגרשן. את חלק זה ביצענו בלי לדעת לאן הוא יוביל וכיצד התקדם אלא התלחנו ותוך כדי עבודה למדנו אותו והתעמקנו בו. לא מצאנו מחקרים שתואמים בדיוק את מה שעשינו ולכן נוצר לנו קושי שלא ידענו בעצם למה לצפות ולאן אנחנו אמורים להגיע או אפילו מה לעשות בתחילת הדרך.

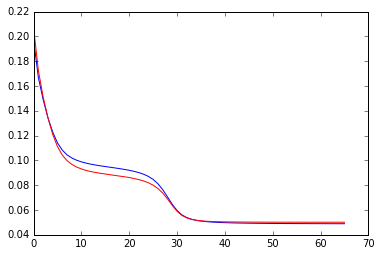
חשוב לציין שבמהלך הרצות הקוד, בכל אחד מן המקרים שנפרט בהמשך, חלוקת הנתונים לאימון, וידוא ובחינה מתבצעת באופן רנדומלי לחלוטין ולכן ישנן פעמים בהן הרשתות יגיעו לתוצאות טובות יותר או פחות בהבדלים גדולים. כדי לפתור את בעיה זו, עבור כל מקרה, הרצנו את הקוד מספר רב של פעמים ובכל פעם שהצלחנו להגיע ללוגיסטיק רגרשן או רשת סופית עם אחוזי הצלחה גבוהים מההרצות עד כה שמרנו אותה על מנת שבעתיד נוכל לעבוד איתה.

עדו דבר כללי לגבי העבודה שלנו הוא תוצאת 0. תוצאת 0 משמעה שקיבלנו שאחוזי השגיאה, בין אם של אחד מהלוגיסטיק רגרשן של נבדק, של הנרמול של סה"כ הלוגיסטיק רגרשנים, או של בדיקת זיהוי של הרשת הנוירונית, הם 0. במציאות דבר זה אינו יכול להתחרש הרי אין תכנה שמזהה ללא אף טעות, אז איך זה קרה בעצם? ההסבר הוא שאנו עובדים עם קבוצות נתונים מאוד קטנות יחסית, מה גם שאנו מחלקים את מאגר התנונים לקבוצות עוד יותר קטנות. בעקבות זאת היו הרצות בהן בוצת הבחינה כללה כ15 דוגמאות שזהו מספר דוגמאות שאם נריץ מספיק פעמים יש סיכוי שהרשת או הלוגיסטיק יצליחו לזהות את כולם. נכיוון שהשגיאה שלנו היא יחסית אז המחשב ידפיס שאחוזי השגיאה שלו היו 0. לכאורה הדבר נכון הרי הוא זיהה את כל הדוגמאות אבל אם ניתן לו מספר גבוה יותר של דוגמאות סביר להניח שהוא כן יטעה מספר פעמים.

התחלנו בכך שאספנו דגימות של שלושה נבדקים, בדרך שציינו מקודם. את הבדיקות עיבדנו ומהן חילצנו את הפיצ'רים שלנו וכך יצרנו את בסיס הנתונים הראשון שלנו לעבודה. הרצנו את הקוד של הלוגיסטיק רגרשן וקיבלנו תוצאות בפעם הראשונה .הרצנו את הקוד מספר פעמים ו בכל פעם קיבלנו שלשת טעויות שונה. בסופו של דבר הגענו לשלשה הטובה ביותר [0,7.6,0]. אפשר להתייחס לסה"כ אחוזי הטעות כ3.5% שזה מספר מאוד טוב עבורנו.

לאחר מכן ניגשנו לעבוד עם רשת הנוירונים שלנו. הגדרנו רשת מתאימה(עם שלושה נוירוני מוצא), שכבת הביניים לא משנה בשלב זה, ואותה הרצנו על נתוני האימון שלנו. שלב זה נעשה לא עלמנת להגיע לתוצאה מחקרית כלשהי אלא בשביל שנראה איך נראים גרפים של למידת הרשת. עשינו זאת בעזרת שרטוט של גרף הטעות על נתוני האימון וגרף הטעות על נתוני הוידוא(שהפונקציה שאיתה עבדנו גזרה לבד מתוך נתוני האימון) על אותה מערכת צירים. מכיוון שחלוקת הנתונים מבצעת באופן רנדומלי כמו שאמרנו כל פעם הגרפים יצאו בצורות שונות משמע בקצב והצלחת למידה שונים. להלן כמה דוגמאות:





ניתן לראות בכל אחת מן ההרצות הלמידה מתבצעת באופן שונה. הגרף הכחול הינו הגרף של הטעות במהלך האימון והאדום של הטעות על נתוני הוידוא.

המשכנו בכך שבדקנו מה המבנה הטוב ביותר כמו שהסברנו מקודם, במקרה זה בדקנו שכבות מוצא בין 20 לבין 80 נוירונים. בכל הרצה נמצאה הרשת הטובה ביותר, אותה הרצנו על נתוני המבחן הסופיים ותוצאת הטעות במקרה זה הינו אחוז השגיאה הסופי של הרשת. בכל פעם שמצאנו רשת יותר טובה מהקודמות שמרנו אותה במקום הקודמת(בדומה למה שעשינו בעת מציאת המבנה הטוב ביותר). לבסוף מצאנו שהרשת הטובה ביותר היא בעלת מבנה של 44 נוירונים בשכבת המוצא והיא הגיע ל15% טעות, כלומר 85% הצלחה. אמנם זה נראה טעות די גדולה אבל חשוב לזכור שאם בן אדם היה צריך לזהות הוא היה עושה זאת באמצעות ניחוש. ניחוש מתוך שלושה משמעו הצלחה של כ33%. אנחנו הגענו לתוצאה שטובה פי 2 וחצי מניחוש זאת אומרת שהתבצעה למידה משמעותית ברשת.

**השפעת הפיצ'רים השונים**

בשלב זה החלטנו לבדוק כיצד הפיצ'רים השונים משפיעים על למידת והצלחת הרשתות. קיימת בעייתיות בפיצ'רים שלנו מכיוון שרובם קטנים מאוד, מכילים כמה ערכים בודדים, מלבד ההיסטוגרמה שהיא לבד מכילה 30 ערכים. בעקבות זאת לא יכולנו לבודד כל אחד מהפיצ'רים אז החלטנו לבדוק פעם נוספת את הלמידה בשתי דרכים. הראשונה היא לנסות ללמד את הרשתות והלוגיסטיק רגרשן בעזרת ההיסטוגרמה בלבד והשיה היא כל שאר הפיצ'רים ללא ההיסטוגרמה. לשם כך יצרנו שני קבצים שכל אחד מהם אחראי על יצירת מאגר נתונים מתאים עבוד הבדיקות שלנו.

לאחר יצירת מאגרי הנתונים ניגשנו חזרה לקובץ של למידת המכונות, בנינו העתקים שלו לפי מאגרי הנתונים החדשים, זאת אומרת ששינינו את הגודל של בסיסי הנתונים המוגדרים בקוד וכמו כן את שכבת הכניסה והביניים של הרשתות. כעת ניגשנו לבדיקות עצמן:

תחילה בדקנו רק את מאגר הנתונים שמכיל רק את ההיסטוגרמה של כל נבדק, הלוגיסטיק רגרשן במקרה זה הגיעה, לאחר בדיקות רבות הגענו לשגיאה של [3.8,7.6,0] שזה בעך 4.7% אחוזים פחות טובים ממה שהגענו אליהם שהיו שאר הפיצ'רים, אפשר להבין מפה ששאר הפיצ'רים יור מועילים ללמידת לוגיסטיק רגרשן מאשר ההיסטוגרמה. רשתות הנוירונים הגיעו לאחר הליך הלמידה ומספר רב של הרצות ל38% שגיאה- פחות טוב פי יותר מ2 מכאשר היו כל הפיצ'רים. שגיאה מינימלית זו התקבלה כאשר מבנה הרשת היה 43 נוירונים בשכבת הביניים. אמנם שגיאות אלו גבוהות מבניסוי הקודם אך גם פה קיימת הצלחה של כמעט פי 2 מניחוש, מה שמראה שהתבצעה למידה. גם מכאן הסקנו ששאר הפיצ'רים היו חסרים ללמידה יותר משמעותית.

לאחר מכן חזרנו שוב על כל התהליך הנ"ל כאשר בסיס הנתונים הוא של כל הפיצ'רים **מלבד** ההיסטוגרמה. במקרה זה הגיעה הלוגיסטיק רגרשן ל0% טעות. רשתות הנוירונים גם כן הגיעו ל0% טעות במבנה של 26 נוירונים בשכבת הביניים. זה כמובן התוצאה הטובה ביותר שהצלחנו וניתן להגיע אליה כלומר שהן זיהו נכון 100% מן הדוגמאות. עם זאת חשוב לציין ש0 אחוז טעות לא באמת אפשרי וזה קרה כי מספר הדוגמאות שלנו היה קטן יחסית כמו שהסברנו למעלה.

**הרחבת התפלגות האנשים**

בשלב זה של הפרויקט החלטנו לנסות להרחיב את הלמידה כך שהרשתות והלוגיסטיק יצטרכו לזהות יותר נבדקים. לשם כך אספנו תוצאות מעוד 6 נבדקים כך שסה"כ היו בידינו תוצאותיהם של 9 אנשים. לאור הגילוי בפרק האחרון-שהלמידה טובה בהרבה כאשר לא משתמשים בהיסטוגרמה, יצרנו את בסיס הנתונים ללא ההיסטוגרמה גם כאן.

מכיוון שהעבודה עם הלוגיסטיק רגרשן הייתה מסורבלת עבורנו החלטנו להמשיך חלק זה בעזרת רשת נוירונים בלבד,מה גם שאותנו אישית היא הרבה יותר מעניינת. לעומת זאת מקודם ראינו שדווקא הלוגיסטיק רגרשן יותר מוצלחת עבור המקרה שלנו אז מאוד יכול להיות שהיה שווה להמשיך איתה ולייעל את הקוד עבורה. בנינו קוד שמותאם למקרה זה עם שכבת מוצא גדולה יותר ותווך מתאים של מבנה שכבת הביניים.

הרצנו את הקוד מספר רב של פעמים עד שהגענו לתוצאה של 58% טעות במבנה של 35 נוירונים בשכבת הביניים. שגיאה זו הינה גבוהה יחסית אך במקרה זה היא טובה כמעט פי 4 מאשר ניחוש זאת אומרת שגם כא קיימת למידה משמעותית.

**שימושים אופציונליים**

בעל המכשיר צריך להגדיר לאפליקציות שנציג בהמשך את עצמו תחילה, זאת אומרת להריץ אותה שתאסוף נתונים (היא תהיה מורשת לגשת לחיישנים בטלפון) בבעת הליכה שלו בזמן של כ2-3 דקות. רצוי לעשות זאת בכמה דרכים: פעם אחת בכל כיס, פעם אחת ביד, כמה פעמים בתיק גב, צד וכל דרך שעולה בדעתו. הסיבה לכך היא כדי שכאשר הוא ילך עם הטלפון בצורה אחרת ממה שהגדירה האפליקציה תחשוב שלא הוא הולך עם הטלפון ותתריע. אופציה אחרת היא שהאפליקציה, בעת הגישה לחיישני המכשיר, תנרמל את התאוצות של המכשיר כאילו היה במצב אנכי כמו בקוד שלנו וכך הלמידה ובעיקר השימוש יהיו פחות מסורבלים.

כל דגימה של התאוצות האפליקציה תעבד בדיוק כמו שאנחנו עשינו בקוד שלנו (ללא ההיסטוגרמה) ותחלץ את הפיצ'רים של אותו אדם. אפשר להוסיף לה מנגנון שאם היא לא מצליחה למצוא מכנה משותף אצל אדם מסוים, אם הדגימות שנתן לא היו אחידות למשל, היא תתריע שהיא עדיין לא מוכנה לשימוש. לאחר שהמשתמש יגדיר את עצמו לאפליקציה מספר מספק של דגימות והיא תגיע לרף טעות מסוים שיוגדר לה כמקסימלי היא תהיה מוכנה לשימוש. אנו מצפים להצלחה רבה אף יותר מאצלנו בקוד משתי סיבות; הראשונה היא שכמות הדוגמאות עבור אותו בן אדם תהיה גדולה בהרבה. השניה היא שהאפליקציה צריכה לזהות אדם אחד בלבד ולא בין כמה אנשים מה שמייעל אותה ומקל על הזיהוי שלה. כמו כן ניתן יהיה להוריד את הסף של נוירון כך שכדי שהוא יזהה את ההולך כאותו אדם ההתאמה תצטרך להיות טובה יותר מה שיוריד מהשגיאה זיהויים לא נכונים.

חשוב לציין שאם האפליקציה תעבוד כל הזמן יהיו הרבה התראות שוא, בשביל זה יש 2 מנגנונים. הראשון כמו שציינו כבר הוא שאם האפליקציה מזהה מישהו שהוא לא בעל המכשיר- היא מחכה זמן מה על מנת לודא את הזיהוי ושלא בטעות תחשוב כך כי הבעלים הלך באופן שונה במקצת. המנגנון השני הוא מנגנון שמזהה מצבים בהם מאפייני ההליכה שונים באופן קיצוני דוגמת ריצה, טיפוס או קפיצות. מקרים אלו בכלל לא יחשבו כהליכה והאפליקציה **לא** תזהה אותם לא נכון **ולא** תתריע שלא כנדרש. כמו כן חשוב שאם למשתמש בעיה בריאותית כלשהי או גורם אחר שעלול להשפיע הליכתו הוא יכבה את האפליקציה מכיוון שהליכתו במצב חריג לא תואמת את הליכתו בד"כ והיא עלולה להתריע התרעות שוא ואף ללמוד על דוגמאות שאינן תואמות להליכה הרגילה של אותו אדם.

**זיהוי גניבת מכשיר**

מכיוון שעבודתינו נועדה לזהות אנשים שונים, והיא מצליחה בכך, היא מתאימה ביותר לאבטחת מכשירי סלולר מפני גניבות. אחוזי ההצלחה שהגענו אליהם מרשימים ביותר ומראים שבהחלט ניתן לזהות בן אדם ספציפי לפי נתוני ההליכה שלו. האפליקציה יכולה להיות מובנית בטלפון או ניתנת להורדה אבל בשני המקרים דרך העבודה איתה צריך להיות דומה- היא תלמד את הליכתו של בעל המכשיר וכאשר תזהה שלא הוא מחזיק במכשיר, אלה איש אחר, היא תתריע לנמען נבחר- שמחזיק המכשיר סומך עליו.

במקרה של גניבה המכשיר יוחזק ע"י הגנב סביר להניח, מתי שהוא בכל אופן מישהו שהוא לא המשתמש ילך עם הטלפון. מאותו רגע של זיהוי של אדם אחר האפליקציה תחכה פרק זמן מסוים, על מנת למנוע טעויות, עד שההסתברות שהטלפון מוחזק ע"י גנב תעבור את ה-50%. אם גם אחרי פרק זמן זה האפליקצייה עדיין תחשוב שלא המשתמש המורשה מחזיק במכשיר היא תשלח הודעה או תתריע בדרך כלשהי באיזה אופן שיבחר המשתמש- או התראה למוקד או שיחת טלפון לאיש קשר מסוים וכו'. כמו כן האפליקציה יכולה להשתמש ולשלוח את שירותי במיקום באופן אוטומטי כך שידעו בדיוק היכן ממוקקם הטלפון שחשוד כגנוב. או אפילו לנעול את הטלפון עד להודעה חדשה מבעליו, כך שהגנב לא יוכל להשתמש בו.

בכל מקרה של התראת שוא המשתמש יכנס לאפליקציה, שתהיה מוגנת בסיסמא כמובן, ויגדיר לה שההתראה הייתה שגויה. מכאן היא תוכל בעצמה להפיק לקחים ולהוסיף את הנתונים האלו למאגר הלמידה שלה וכך להשתפר באופן עצמאי.

משימוש זה אפשר להוסיף עוד שימוש הקשור אליו והוא זיהוי הגונב. במקרה של זיהוי אופציונלי לגניבה האפליקציה באופן אוטומטי תתחיל ללמוד את דפוס ההליכה של המחזיק העדכני (בלי לפגוע בלמידה של הבעלים הרגיל) וכך תוכל לייצר לו מאין דיוקן או פרופיל שמבוסס על הליכתו בלבד.

**זיהוי הליכה מזיקה לגוף**

שימוש נוסף שאנו יכולים לעשות בקוד שלנו הוא למטרות בריאותיות- רפואיות. אם התאמה מתאימה, שתכלול כמות גדולה של נתונים מאנשים עם צורת הליכה בריאה וצורת הליכה אשר מזיקה למפרקים ועמוד השדרה, הקוד שלנו יכול במקום לסווג אנשים שונים, לסווג אופיין הליכה על פי רמת הנזק שהוא גורם לגוף של ההולך. תחילה נצטרך לייצר מאגר נתונים של אנשים עם הליכה אשר פוגעת בגוף (עדיף עם כמה בעיות שונות) והליכה בריאה אשר לא פוגעת בגוף. נצטרך להיעזר במומחים בתחום אשר יודעים להבדיל בין בעיות הליכה שונות וגם לזהות הליכה בריאה. המשתתפים ילכו עם טלפון סללורי על גבם ואנו נאסוף את הנתונים.

בשלב הבא נוכל להפעיל אלגוריתם דומה מאוד לזה שעשינו בעבודתינו רק שבמקום לזהות אנשים שונים יהיה עליו לקבוע האם ההליכה מזיקה לגוף או לא. האלגורתים ימצא את מאפייני ההליכה השונים ואחר כך גם את הקשר ביניהם לבין הבעיה בהליכה (אם קיימת כזו בכלל). אחרי שיהיו לנו את הפרמטרים השונים אנו נוכל לקחת דגימות מאנשים שונים, אשר הלכו עם המכשיר סלולר על גבם, ולהגיד להם מה היא הבעיה בצורת הליכתם. דרך נוספת שאנו יכולים לאסוף את הנתונים היא באמצעות יישומון לטלפון שיוזנו בו כבר הפרמטרים השונים מתהליך הלמידה שהתבצע על מחשבים חזקים וכך כל אדם עם מכשיר סלולרי יוכל ללכת איתו מוצמד לגבו ולבדוק בעצמו אם הליכתו בריאה ונכונה לגוף בתהליך קצר ופשוט.

השלב הבא הוא לקבל אישור ממשרד הבריאות לשיטת הבדיקה שלנו וכך נוכל להפנות את אותם אנשים שיש להם בעיה בהליכה לרופא המתאים שיוכל להגיד להם מה הבעיה בהליעתם וכיצדניתן לשפר אותה על מנת לשמור על הגוף. גילוי מקדים של הליכה שגויה המזיקה לגוף יכול לעזור לאנשים למנוע כאבי מפרקים וגב בגילאים מבוגרים יותר וכך לעזור להם לשמור על רמת תחיים גבוה.

**זיהוי שכרות והשפעת סמים**

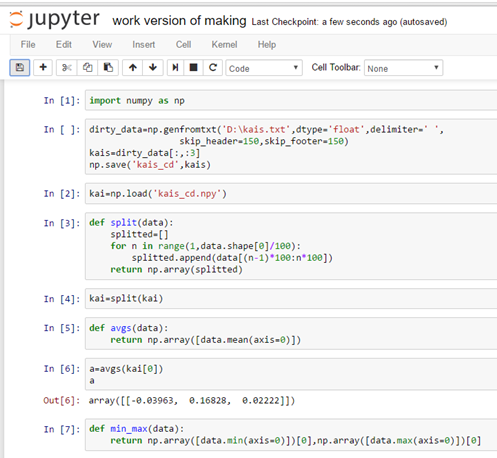
שימוש נוסף שאנו יכולים לעשות עם עבודתינו הוא לזהות האם הבן אדם הולך תחת השפעת אלכוהול או סמים. אנחנו נצטרך לאסוף נתונים מקבוצת אנשים שחלקם תחת השפעת אלכוהול, חלקם תחת השפעת סמים וחלקם פיקחים לחלוטין.

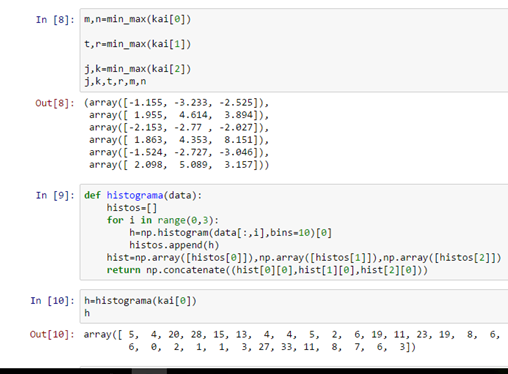
אחרי שנריץ קטע קוד המסוגל למצוא, בעזרת למידת מכונה, את הקשרים השונים בין מאפייני ההליכה לבין השפעת סמים ואלכוהול על הליכת הבן אדם, נוכל לקבוע עבור כל מכשיר, עם יישומון מתאים, האם ההולך איתו הוא תחת השפעתם. את האפליקצייה לא יהיה ניתן להסיר מהמכשיר והיא תיהיה חובה להתקנה במקומות שבהם השימוש בסמים ואלכוהול פוגע בחיי אדם או בבטחון המדינה.

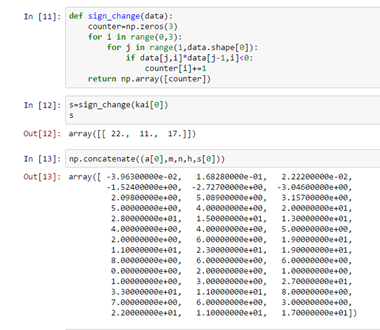
האפליקציה תוכל להיות שימושית עבור המערכת הצבאית ומערכת הרפואה בארץ. כיום הצבא מבצע עשרות אלפי בדיקות למציאת סמים בחיילים שמשרתים במקומות שבטחון המדינה תלוי בכך. אם הצבא יוכל לדעת מראש מה היא ההסתברות שאותו בן אדם לקח סמים באמצעות האפליקצייה שלנו הוא יוכל לבצע את הבדיקה רק על חלק מהחיילים ולחסוך ככה בכסף במקום ערכות בדיקה רבות ובכל זאת לתפוס את העוברים על החוק ולשמר את ההרתעה. אם ייעשה שימוש באפליקציה שלנו בבתי חולים עבור רופאים מנתחים הם יוכלו להיות מושעים מתפקידם ולהענש בהתאם על סיכון חיי אדם ברשלנות אם נמצא והם תחת השפעת סמים או אלכוהול.

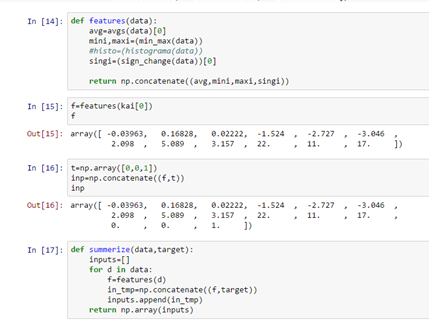
ברור כי לאפליקצייה כזו אין יכולת לקבוע בודאות עם אותו אדם נמצא תחת השפעת אלכוהול או סמים אך היא יכולה לחזות על מי עדיף לבצע בדיקות שמאתרות חומרים אילו ועל מי עדיף לדלג בשביל לשמור על פרטיותם של אותם נבדקים, משאבים כלכליים וזמן.

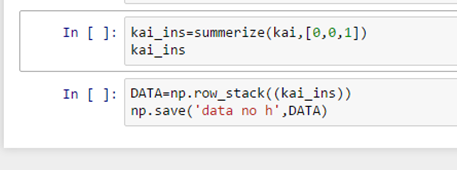
**נספחים וביבליוגרפיה**

**נספחים**



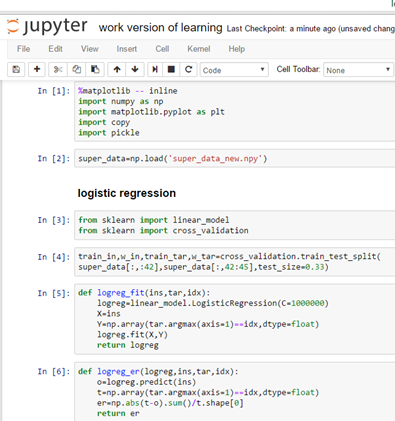






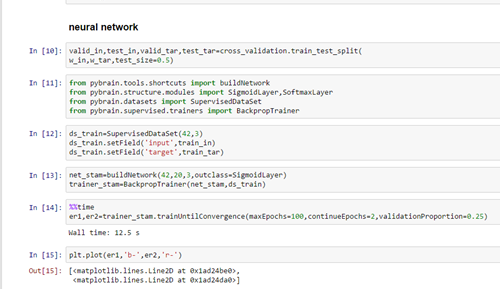
תחילה ניתן לראות את אחד מקבצי הכנת מאגר הנתונים שלנו. זהו קובץ מקור שכולל דוגמא של נבדק אחד בלבד, הפעולות חוזרות על עצמן עבוד כל הנבדקים ובהתאם לניסוי.

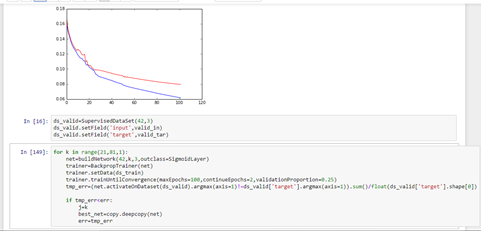
אחר כך רואים את אחד מקובצי הניסוי המעשיים שלנו. כל קובץ עסק בניסוי אחר ובהתאם הוגדרו לו מאפיינים כגון שכבת המוצא או הכניסה של רשת.

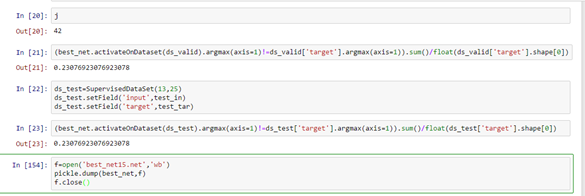


בחלק זה ניתן לראות את הקוד שבנינו עבור הלוגיסטיק רגרשן.









**ביבליוגרפיה**

GSMARENA- הסבר על חיישן תאוצה: <http://www.gsmarena.com/glossary.php3?term=sensors>

Livescience- הסבר על חיישן תאוצה: <http://www.livescience.com/40102-accelerometers.html>

[Pybrain- הסבר על החבילה:](http://kdd.org/exploration_files/v12-02-19-SensorKDD-Kwapisz.pdf) <http://pybrain.org/>

עבודת החקר שנעשת בנושא שלנו:<http://kdd.org/exploration_files/v12-02-19-SensorKDD-Kwapisz.pdf>

ויקיפדיה- הערך למידה חישובית:<https://he.wikipedia.org/wiki/%D7%9C%D7%9E%D7%99%D7%93%D7%94_%D7%97%D7%99%D7%A9%D7%95%D7%91%D7%99%D7%AA>

ויקיפדיה- הערך למידה מונחית:<https://he.wikipedia.org/wiki/%D7%9C%D7%9E%D7%99%D7%93%D7%94_%D7%9E%D7%95%D7%A0%D7%97%D7%99%D7%AA>

ויקיפדיה- הערך רשת עצבים מלאכותית:<https://he.wikipedia.org/wiki/%D7%A8%D7%A9%D7%AA_%D7%A2%D7%A6%D7%91%D7%99%D7%AA_%D7%9E%D7%9C%D7%90%D7%9B%D7%95%D7%AA%D7%99%D7%AA>

ויקיפדיה- הערך רשת עצבית:<https://he.wikipedia.org/wiki/%D7%A8%D7%A9%D7%AA_%D7%A2%D7%A6%D7%91%D7%99%D7%AA>