**פרוייקט גמר**

StarTrace

**מחברים**: נטע לי להט, גל בן יאיר, יונתן אברהמי ועמית וייל

**מוגש לבית הספר למדעי המחשב**

**המסלול האדמאי, המכללה למנהל**

**כחלק מהדרישות האקדמאיות**

**לתואר B.Sc. במדעי המחשב**

אב התשע"ה ראשל"צ יולי 2015

**פרוייקט גמר**

StarTrace

**מחברים**: נטע לי להט, גל בן יאיר, יונתן אברהמי ועמית וייל

**המנחה המאשר**: ד"ר משה בוטמן

**בית הספר למדעי המחשב**

**המסלול האדמאי, המכללה למנהל**

אב התשע"ה ראשל"צ יולי 2015

**תוכן העניינים – לערוך בסוף**

[תקציר - 9 -](#_Toc425803290)

[הקדמה - 10 -](#_Toc425803291)

[הגדרת עולם הבעיה - 10 -](#_Toc425803292)

[הגדרת הבעיה ומה נעשה בתחום - 11 -](#_Toc425803303)

[פתרון - הסבר על הפתרון המוצע וכיצד שונה מאחרים - 12 -](#_Toc425803304)

[הצדקה אקדמאית - 12 -](#_Toc425803305)

[מטרת הפרויקט - 13 -](#_Toc425803306)

[תרשים בלוקים\מודולים כללי של המערכת - 13 -](#_Toc425803308)

[DR - 13 -](#_Toc425803309)

[SRS - 14 -](#_Toc425803315)

[מתודולוגיות וכלי פיתוח, חומרה ותוכנה - 14 -](#_Toc425803316)

[המערכת - כולל תמונות מסך של קלט ופלט המערכת - 16 -](#_Toc425803330)

[משימות על פי זמנים והסבר על עמידה בזמנים - 21 -](#_Toc425803351)

[קשיים במהלך הפרויקט וכיצד נפתרו - 22 -](#_Toc425803360)

[ביבליוגרפיה - 22 -](#_Toc425803362)

תקציר

בעידן הרשתות החברתיות וכמויות המידע העצומות הקיימות ברשת ניתן לצבור מידע רב היכול לשמש להצלחת שיווק ופרסום בעולם כולו.

לפני תחילת מסע פרסום ניתן להשתמש במידע זה ככלי על מנת לדעת מה דעת הרוב על מפורסם בעיר מסוימת ובכך למנוע כישלון ובזבוז כספים על פרסום לא מוצלח.

StarTrace תעזור לבחור במפורסם המתאים ביותר למסע פרסום במיקום מסוים (עיר או מדינה) בעולם בהתבסס על ניתוח סנטימנטלי של תגובות משתמשים על אותו המפורסם ברשתות חברתיות.

בממשק למשתמש יש להזין שם של מפורסם ועיר ולאחר חישוב יוצגו למשתמש אחוזי הדעות החיוביות, השליליות והניטרליות על אותו המפורסם בעיר שנבחרה.

ממשק ה- WEB ימומש ע"י טכונלוגית AngularJS ביחד עם חבילת עיצוב Bootstrap, ע"מ ליצור מראה פשוט ונקי. שרת האפליקציה ישלח בקשות אל שרת מכונת הלמידה, אשר תנתח אותן לפי אלגוריתמי הניתוח הסנטימנטלי, ויקבל חזרה את תוצאות הניתוח. מכונת הלמידה תאסוף את המידע באמצעות הממשקים השונים שמציעות הרשתות החברתיות השונות (טוויטר, פייסבוק, אינסטגרם וכו'). הקוד של מכונת הלמידה ימומש בPython 2.7

הקדמה

כיום ברשת האינטרנט קיים מידע רב אודות מפורסמים ומעריציהם. המידע אודות המפורסמים מגיע בכמויות גדולות וממקורות שונים ומגוונים. לאיסוף ועיבוד המידע פוטנציאל עסקי רב.

הגדרת עולם הבעיה

כיום כל החברות בעולם אשר מפרסמות מוצרים לציבור הרחב צריכה מפורסם שיעמוד בראשה וייצג אותה כלפי הציבור עבור שלטי חוצות\קטלוגים\פרסומות וכדומה.

בלי פרסום זה וקמפיינים לשיווק המוצר, החברה והמוצר כנראה לא יממשו את פוטנציאל השיווק שיש להם.

מעבר לכך שיש צורך בשיווק נכון, הבחירה בפרזנטור שיהיה פני החברה עבור שיווק ופרסומים אלו היא לא פחות חשובה.

ישנם כמה סוגים של פרזנטורים:

1. פרזנטור מומחה – מקצועי בתחום הנמכר ונותן תחושה מקצועית וטובה לקונים מכיוון שמבין בתחום.
2. פרזנטור צרכן – ממליץ על המוצר בעקבות התנסות
3. ידוען\סלבריטי – שיכול לייצר אצל חלק ממעריציו גם תחושה טובה לגבי המוצר.

כאשר נרצה לבחור מפורסם להיות הפרזנטור של חברה – איך נדע במי לבחור?

כל בן אדם, כל תרבות וכל סביבה תעריץ סוגים שונים של מפורסמים כך שיכול להיווצר מצב שבסביבה מסוימת מפורסם אהוב נורא ובסביבה אחרת שנוא.

סוגיה זו יוצרת בעיה עבור חברות הפרסום שצריכות לבחור פרזנטור שיגרום לכמה שיותר אנשים ללכת אחריו ולקנות את המוצר.

הגדרת הבעיה ומה נעשה בתחום - 2 עמודים לפחות

עד כה בחינה כמותית ומעמיקה של אפקטיביות פרסומות והשוואת ההשפעה של פרזנטור כזה או אחר בקטגוריה כזו או אחרת הייתה משימה כמעט בלתי אפשרית. ניתן היה לערוך בחינה שכזו בשתי שיטות שלכל אחת מהם חסרונות משלו. הראשונה, הצגת פרסומות בפני קבוצות מיקוד, מתודולוגיה אשר לא נתנה תשובות כמותיות מובהקות לסוגיות והשנייה באמצעות הצגת פרסומות בפני עוברים ושווים במרכזי קניות, מתודולוגיה שלא אפשרה להגיע למדגם כלל ארצי מייצג.

כיום, באמצעות הפאנל האינטרנטי של חברת פאנלס, ניתן לראשונה לבחון סוגיות הנוגעות לפרסומות על פני מדגם ארצי מייצג.

בסקר שערך מכון המחקר פאנלס, עבור אגוד חברות הפרסום ובית הספר של חברות הפרסום, הוצגו בפני כ- 451 נשאלים מגוון פרסומות מ- 5 קטגוריות כאשר כל פרסומת כוללת סוג שונה של פרזנטור: ."סלב", מומחה או צרכן (סה"כ 15 פרסומות).

על כל פרסומת נשאלו מספר שאלות בנוגע ל: האהדה לפרסומת, תפיסת האמינות שלה, העניין והסקרנות שהיא יצרה לגבי המוצר/השירות המפורסם, התרומה של הפרסומת להעדפת המוצר/השירות המפורסם.

מן הסקר ניתן לראות כאשר מחפשים ליצור **אהדה** למוצר מסוים, פרזנטור שהוא "סלב" מוביל בכל הקטגוריות. כלומר, על מנת להפוך את המוצר לפופולארי, יש להשיג פרזנטור שהוא "סלב".

אך כיצד ניתן לדעת מי הוא הסלב המתאים ביותר לשיווק המוצר אותו אנו רוצים להפוך לפופולארי?

נראה שאין דרך פשוטה למצוא מי מתאים מכיוון שאין את הכלים המתאימים למצוא איזה "סלב" הוא האהוד ביותר בסביבה מסוימת. ניתן להסתמך על דברים כמו רמת חשיפה או הופעות במדורי הרכילות, אך לעיתים דברים אלה עלולים להתפספס ובנוסף חסר מיקוד עבור ה"סלב" ברמת המיקרו, איפה הוא יותר פופולארי? בקרב אילו גילאים? האם יותר פופולארי בקרב נשים או גברים?

פתרון נוסף שקיים בעולם הדיגיטלי הוא Google Trends ו-Facebook Insights.

Google Trends יכול לתת מידע על כמה מחפשים ברחבי הרשת נושא מסוים או אדם מסוים ובאילו מקומות. Facebook Insights מציע לבעלי עמוד מסוים לקבל דוחות על כמות התנועה בעמוד שלהם ובכך יכול לספר לבעלי עמוד של "סלב" מסוים מהי רמת הפופולאריות שלו.

פתרון - הסבר על הפתרון המוצע וכיצד שונה מאחרים

מתוך שלושת הסוגים של פרזנטורים שדיברנו עליהם (צרכן\מומחה\ידוען), StarTrace באה לתת פתרון לסוג השלישי- בחינת אהדת הידוענים.

ניתן גם לתת פתרון עבור שני הסוגים הראשונים במידה והם גם דמויות הידועות לציבור.

מערכת StarTrace בונה על חוכמת ההמונים, ומחפשת שם את התשובה לשאלה שנשאלה בסוף הסעיף הקודם, מהי רמת הפופולאריות של "סלב" מסוים? היא עושה זאת באמצעות פניה אל הרשתות החברתיות בכדי להבין את רמת האהדה שלהם בקרב הציבור. באמצעות ניתוח סטאטוסים, Tweetים, תגובות ודיווחים על אותו ה"סלב", נוכל ללמוד מהי רמת האהדה שלו ברחבי הרשת. את רמת האהדה נוכל לאחר מכן להציע לחברות פרסום שונות אשר מחפשות את הפרזנטור הבא עבור המוצר שלהם. את החיפוש שלנו ניתן לבצע לפי מיקום ורשת חברתית ובהמשך תינתן האופציה לחפש גם לפי גילאים ומין.

הפתרון שלנו שונה מזה של Google Trends בכך שהוא יכול לסווג את ה"רעש" שהאדם אותו מחפשים עושה ברשת, בעוד ש-Google Trends בסך הכל יגיד **כמה** מחפשים אותו. מדדי הסיווג שלנו הם "חיובי", "שלילי" ו-"נייטרלי". בנוסף, אנו שונים מ-Facebook Insights בכך שאנחנו מאפשרים גם לאנשים שהם אינם מנהלי העמוד של ה"סלב"(ובכך למעשה מקורבים ברמה כלשהי אליו) לדעת מהי חוות הדעת עליו ברחבי העולם.

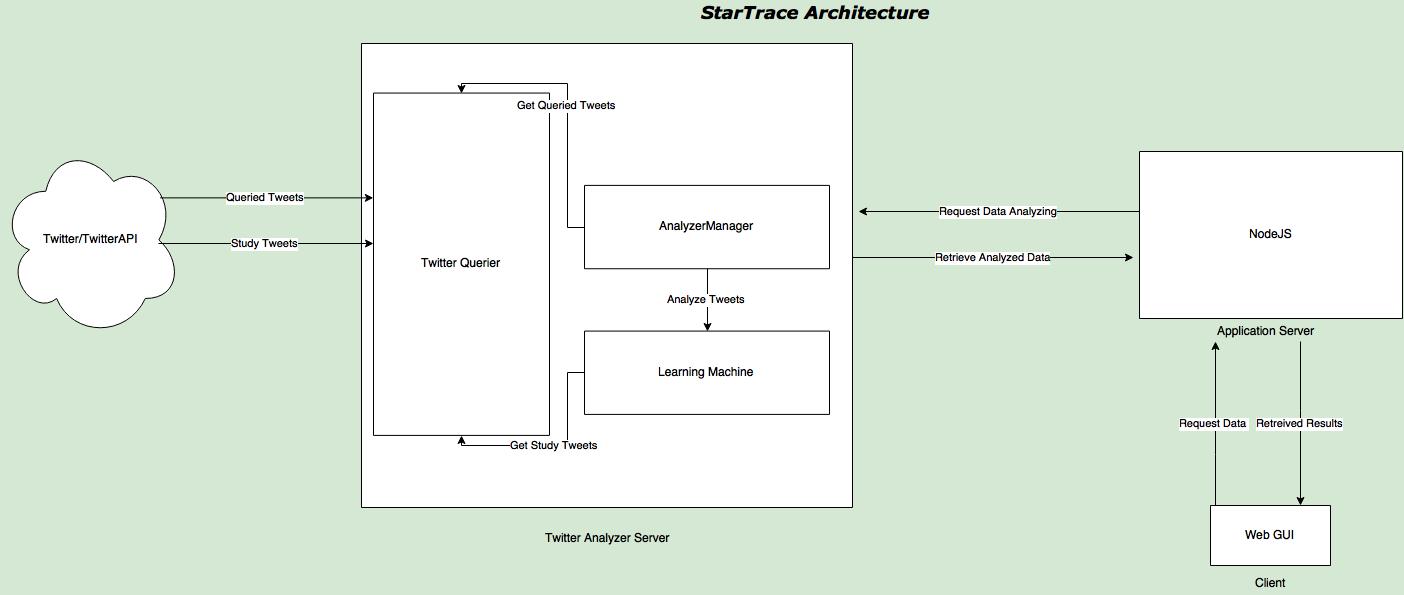
הצדקה אקדמאית

כדי להסיק תובנות בעלות פוטנציאל עסקי מתוך מאגרי הנתונים הנאספים מרשתות חברתיות על מפורסמים השתמשנו בדברים שלמדנו בקורס למידה חישובית וכריית נתונים.

את עולם כריית המידע ניתן לחלק לסוגי בעיות למידה שונות, במקרה שלנו זוהי בעיית סיווג. השתמשנו באחת הדרכים המומלצות לפתרון בעיות למידה מסוג זה, דרך זו בדרך כלל מגיעה לתוצאות סיווג גבוהות ומוצלחות מאוד. המסווג בו השתמשנו הוא Naive Bayes classifier, מסווג בייסיאני נאיבי הוא ממשפחת מסווגים הסתברותיים פשוטים המבוססים על יישום חוק בייס עם השערות בלתי תלויות חזקות (נאיביות) בין התכונות. המסווג לומד מסט דוגמאות שכבר תויגו (קיבלו ערך סיווג) ועל פי תכונותיהן שלמד לכל תיוג הוא יסווג את סט הדוגמאות הלא מתויגות.

מטרת הפרויקט

מטרת הפרויקט היא מציאת תובנות עסקיות בעזרת אלגוריתמי למידה על בסיס מידע רב שנאסף מרחבי האינטרנט. כדי לקבל החלטות עסקיות בתחום הפרסום נדרש להשתמש במידע של דעת הקהל הרחב על מפורסמים כדי שתהליך הפרסום יהיה מוצלח. נבנה מערכת שמספקת ידע לגבי דעת הקהל על מפורסמים על פי ערים בעולם על בסיס מידע.

תרשים בלוקים\מודולים כללי של המערכת

DR

כאשר באנו לפתח את המערכת חשבנו כיצד ניתן לכתוב מימוש פשוט וגמיש עבור האפליקציה שלנו, בנוסף רצינו לעבוד בטכנולוגיה אשר קיימות בה כבר ספריות הממשות שיטות סיווג וממשקים נוחים לרשתות החברתיות. בסופו של דבר ראינו שהטכנולוגיה הכי מתאימה לדרישות אלו היא Python בשל הקיום של ספריות שנוכל להיעזר בהן על מנת לסווג את המידע וכמובן מימושים איכותיים לממשקים מול הרשתות החברתיות.

בשלב הבא גיבשנו את הארכיטקטורה של המערכת. החלטנו שיהיה שרת ייעודי המוקדש ללמידה. שרת זה ימשוך נתונים מהרשתות החברתיות (מקור המידע החיצוני שלנו), יבצע עיבוד של המידע וישמור אותו אצלו. על שרת זה להיות מאופיין בדברים הבאים:

* מפרט טכני חזק. ביצוע פעולות למידה עשוי להיות פעולה מאוד כבדה, במיוחד כאשר מגיעות בקשות ממספר לקוחות שונים, לכן שרת זה זקוק להרבה זיכרון פנימי (RAM) ומעבד רב ליבות על מנת לעמוד במשימה.
* ייחצון ממשק RESTful – החלטנו שבכדי להפריד בצורה טובה בין השרת ללקוח, על השרת לייחצן ממשק RESTful אשר מאפשר ללקוח לשלוח בקשת POST מעל HTTP על מנת לקבל חזרה נתונים. כך למעשה שרת הלמידה לא צריך לדעת מי הלקוחות שלו, ויהיה ניתן לממש את צד הלקוח בטכנולוגיות שונות (Web, Mobile, Rich Client etc.)

לאחר מכן ניגשנו למימוש צד הלקוח. החלטנו לממש את צד הלקוח בשלב הראשון כשרת WEB אשר מכיל אתר המאפשר לבצע חיפוש על הסלבי המעניינים את המשתמש. נטע לי אולי תוסיפי עוד קצת כאן?

SRS

מתודולוגיות וכלי פיתוח, חומרה ותוכנה

המערכת מחולקת לשני חלקים – צד שרת וצד לקוח.

1. חלק א' – צד שרת

צד השרת פותח לחלוטין ב-Python על מנת להשיג זריזות וגמישות בעבודה עם הממשקים. את הקוד אשר עוסק בלמידה פיתחנו באמצעות ספריות בשם SCM אשר מממשות מודלים של למידה חישובית. אל Twitter ניגשנו באמצעות ספריית python בשם TwitterAPI אשר יחצנה עבורנו את כל ממשקי החיפוש להם היינו זקוקים. על מנת להפוך מיקום משם של מקום לקואורדינציות השתמשנו בשירות רשת של GeoNames שמוחצן על ידי ספריית geopy, ספרייה שעוטפת ממשקי REST שונים ברחבים הרשת. את ממשק ה-REST פיתחנו באמצעות טכנולוגיה מבוססת Python בשם Flask, המאפשרת פיתוח אפליקציות אינטרנטיות פשוטות ומהירות. הבקשות הן בקשות POST אשר עוברות מעל HTTP.

1. חלק ב' – צד לקוח
   1. שרת ה-Web – Node.js

מה שמייחד שרתי אינטרנט הוא היכולת שלהם להתמודד עם מספר בקשות בו זמנית לאותו קוד. מספר גולשים למשל, מנסים להגיע בו זמנית לאותו הדף. במטרה לטפל בדרישה זאת שרתי האינטרנט משתמשים בתקשורת מבוססת threads. השרתים תומכים בריבוי threads ובדרך כלל לכל connection לשרת מוקצה thread. הבעיה בתקשורת זאת שהיא לא תמיד יעילה ודי קשה לשימוש, במיוחד שמבקשים לבצע גישה מהירה ומקבילית למשאבים ולקבצים: יכול להיות בזבוז זיכרון נרחב ויכולות להיות גם נעילות. בעיה זאת מחמירה ככל שהעומס על השרת גבוה יותר ונדרשות יכולות גבוהות מהמפתחים לבצע אופטימיזציות ושיפור ביצועים.  
לכן בחרנו לממש את שרת ה-Web עבור הפרוייקט ע"ג טכנולוגית Node.js, במטרה ליצור יכולת א-סינכרונית המאפשרת ביצועים גבוהים.  
node.js תוקפת ישירות את בעיה הנ"ל ומנסה לשחרר את צוואר הבקבוק מפעולות ה- I/O שחונקות את האפליקציה. אפליקציית node.js יכולה לטפל בסוגים שונים של בקשות I/O באותו הזמן משום שברירת המחדל של כל הפונקציות והספריות היא אסינכרונית. (יש גם מספר פונקציות סינכרוניות אבל הרוב המוחלט אסינכרוני).  
כל בקשות ה- I/O מהרשת מוגדרות כ- nonblocking וכל בקשות ה- I/O לקבצים מוגדרות אסינכרוניות. זאת אומרת, שהאפליקציה יכולה להמשיך לרוץ והגישה ל- I/O לא נחסמת.  
באמצעות שימוש ב- node ניתן להגיע לייעול מיטבי של הזיכרון תחת עומסים כבדים. מי שמשתמש ב- node לא צריך לחשוש מ- dead-locking של processes כיוון שאף פונקציה של node לא מבצעת בצורה ישירה פעולות I/O ולכן אין נעילות.  
  
שרת ה- Web בפרוייקט מקבל בקשות מהמשתמש, מעביר אותן לשרת ה-Phyton , המעבד את הנתונים שנשלחו אליו, ומחזיר אל שרת ה-Web את התוצאות בפורמט JSON.  
שרת ה-Web מחזיר את ה-JSON לצד הלקוח (AngularJS), והוא מציג את התוצאות.

* 1. AngularJS – צד הלקוח

Angularjs הינה פלטפורמה לפיתוח אפליקציות Web בצד הלקוח מבית היוצר של חברת גוגל, הפלטפורמה בנויה בשיטת MVC ובשונה מפלטפורמות פיתוח אחרות היא מסוגלת להשתלט על כל חלקי האפליקציה ולהכתיב את אופי העבודה כולו.  
Angularjs היא ספריית MVC הכוללת את המודלים Services, Controller ו – View.   
עקרון העבודה המנחה הוא לקיחת יותר החלטות עבור המשתמש דבר הבא לידי ביטוי במספר שורות הקוד הקצר יותר הנדרש.  
Angularjs שמה דגש על נושא בדיקתיות המערכת וניתן לכתוב באמצעותה תכניות בדיקות אינטגרציה ובדיקות מערכת בקלות יחסית.  
  
פלטפורמה זאת מאפשרת לנו, מפתחי הפרוייקט, להציג את תוצאות העיבוד בשרת בצורה נוחה ומאורגנת.

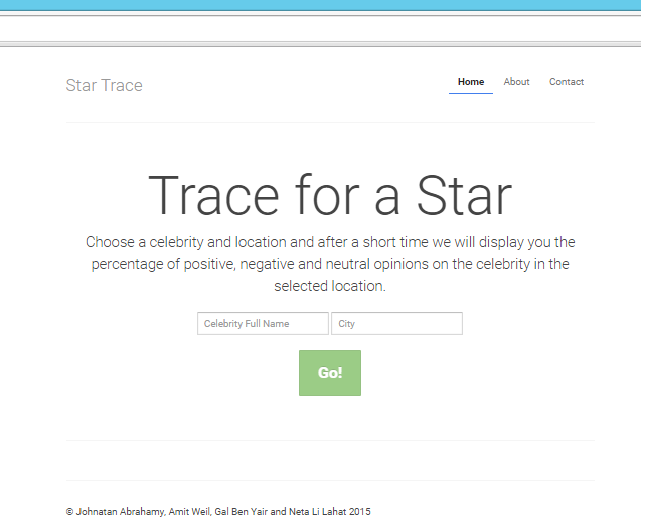
* + 1. פירוט המודולים בפרוייקט
       1. Controllers –  
          MainController – מודול זה אחראי לכל מה שמוצג בדף – כותרת, תיבות טקטס, הודעות למשתמש, תוצאות החיפוש, טבלאות ונתונים. דרכו מתבצעת שליחת הטקסט לחיפוש ל-Service.
       2. Services –   
          DataService – מודול זה אחראי לשלוח את את בקשת ה-http לשרת. מודול זה גם מקבל בחזרה את התוצאות (json) ומעביר ל-MainController.
       3. Views – התוכן הדינאמי המשתנה בהתאם למצב השימוש.  
          מצב הפתיחה – דף הפתיחה – בתוך הדף ישנו מקום לתוצאות החיפוש. כאשר תוצאות החיפוש מוחזרות, הן "נכתבות" לתוך הדף באופן דינאמי.  
          מצב ה"אודות" – דף האודות  
          מצב ה"צור קשר" – דף צור קשר
  1. BootStrap – ניראות האתר  
     FrameWork המספק סט בסיסי של כלי css לעיצוב אתרים הכולל ממשקים למשתמש, תפריטים, כפתורים, שדות וכו'. בנוסף, העיצוב הוא רספונסיבי.  
     FrameWork זה מכיל יישומי JS (בשילוב jQuery) לממשק משתמש.  
     מטרתו היא ליצור ממשק איכותי ולקדם אחידות בשימוש בכלים באתרים, לפשט את השימוש, ולהאיץ את תהליך הבנייה.

המערכת - כולל תמונות מסך של קלט ופלט המערכת

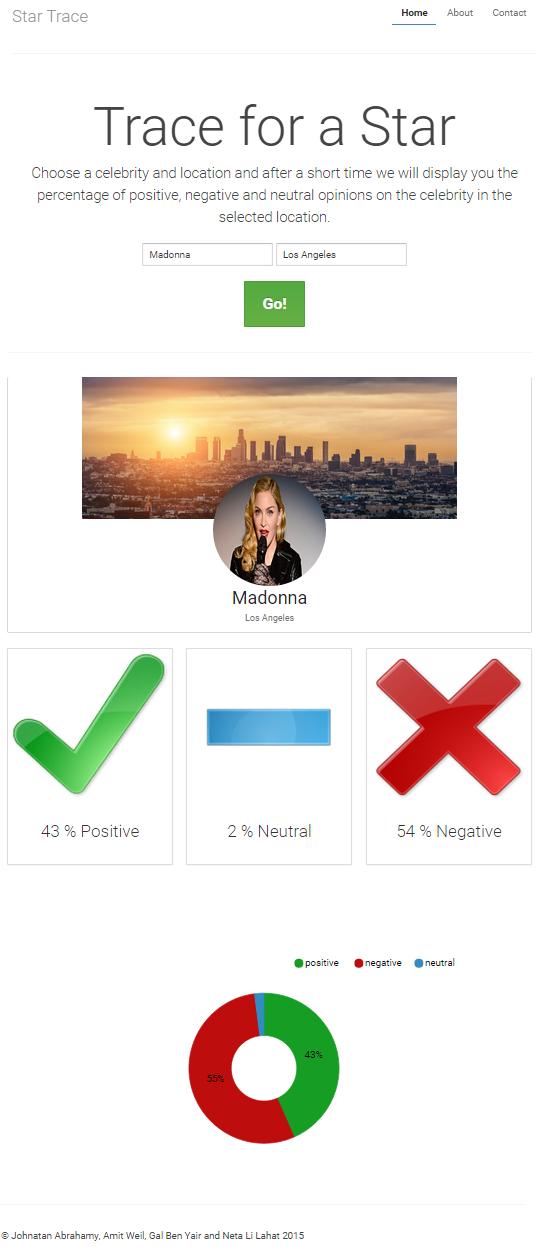
מסך הבית של המערכת:

התחברות המשתמש לאתר תביא אותו ישר לצפיהה בדף הHome.

* מאפשר הזנה של שם מפורסם ועיר בעולם עליהם ירצה המשתמש להריץ את סקירת התגובות.



לאחר הזנת הנתונים ולחיצה על "GO" מופעל האלגוריתם והמשתמש מקבל את תוצאות המערכת.

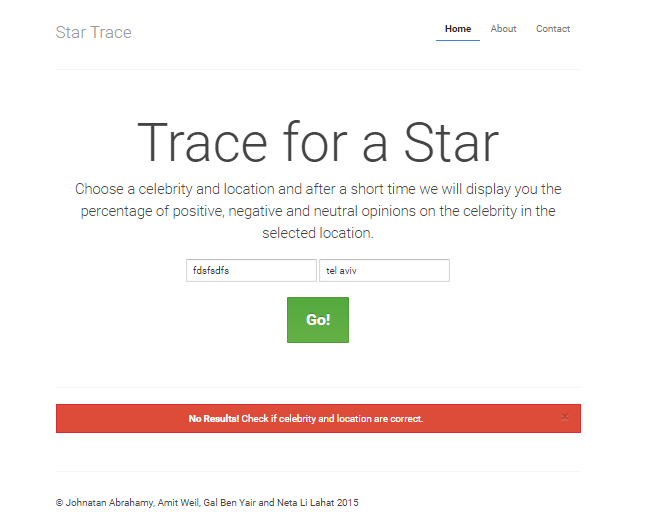


* מוצגת תמונה המפורסם ותמונה המקושרת לעיר שנבחרה
* התוצאות מתחלקות ל3 קטגוריות:

1. חיובי
2. שלילי
3. ניטרלי

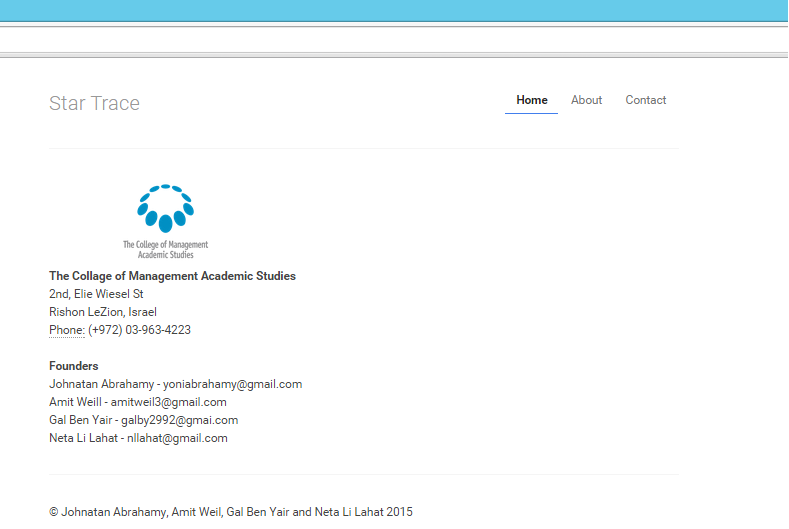
* ניתן לראות את התוצאות גם בצורת גרף.

במידה והמשתמש הזין נתונים (עיר\מפורסם) לא קיימים, המערכת לא תחזיר נתונים ותתקבל ההודעה הבאה:



מסך יצירת הקשר

בעת לחיצה על כפתור הcontact בתפריט המשתמש יגיע לדף יצירת הקשר הבא :



מסך הabout

מכיל פרטים כלליים על המוצר StarTrace .

מסך זה מציג גם את היכולת העסקית של המוצר עבור חברות שמחפשות מפורסמים.

משימות על פי זמנים והסבר על עמידה בזמנים

* בדה"ת לטכנולוגיות ודרכי איסוף נתונים מרשתות חברתיות שונות- ינואר 15
* בחירות טכנולוגיות ותחילת מימוש – פברואר 15
* פיתוח אופן הלמידה ובחינת פלט האלגוריתם – מרץ 15
* פיתוח ממשק למשתמש – אפריל 15
* הכנת תקצירים, ברושור ופוסטר לפרויקט - מאי 15
* הצגת הפרויקט – יוני 15
* ספר פרויקט – יולי 15

צורת העבודה שבה עבד הצוות למימוש הפרויקט הייתה חלוקת משימות בין חברי הצוות כדי שנעמוד בלוח הזמנים הנדרש לפרויקט, עמדנו בזמני העבודה שנדרשו.

קשיים במהלך הפרויקט וכיצד נפתרו

הקושי המרכזי במהלך הפרויקט היה למצוא API ידידותי אשר יאפשר לנו להוציא מידע על אנשים מרחבי הרשתות החברתיות. ראשית ניסינו לעבוד מול Facebook, אבל לאחר שגילינו שהם ביצעו שינויים ב-API שלהם שמונעים הוצאת מידע בסיסי על משתמשים החלטנו לעשות שינוי כיוון (Pivotting) ולהוציא מידע מ-Twitter במקום. כעת נתקלנו ב-API ידידותי יותר המאפשר לאחזר Tweetים ממשתמשים שונים ובנוסף לקבל Tweetים לפי מיקום. לאחר שאחזרנו את ה-Tweetים כל מה שנותר הוא לנתח אותם ולייחצן את התוצאת באמצעות ממשק REST מעל HTTP. קושי נוסף שנתקלנו בו הוא שה-API של Twiiter מאפשר להוציא רק מספר מסוים של Tweetים מתקופה שלא תעלה על חודש. פתרון שחשבנו על מנת לפתור את הבעיה הזו היא לשמור את הTweetים במסד נתונים משלנו, ובהמשך ליצור ממשק המבצע שילוב בין אחזור מידע מ-Twitter לבין עבודה מול מסד הנתונים שלנו על מנת להפיק ביצועי למידה טובים יותר ולהגיע לתוצאות איכותיות יותר.

ביבליוגרפיה

מאמר בנושא בעיית בחירת הפרזנטור - <http://www.allmarketing.co.il/?CategoryID=58&ArticleID=7500>