**s**

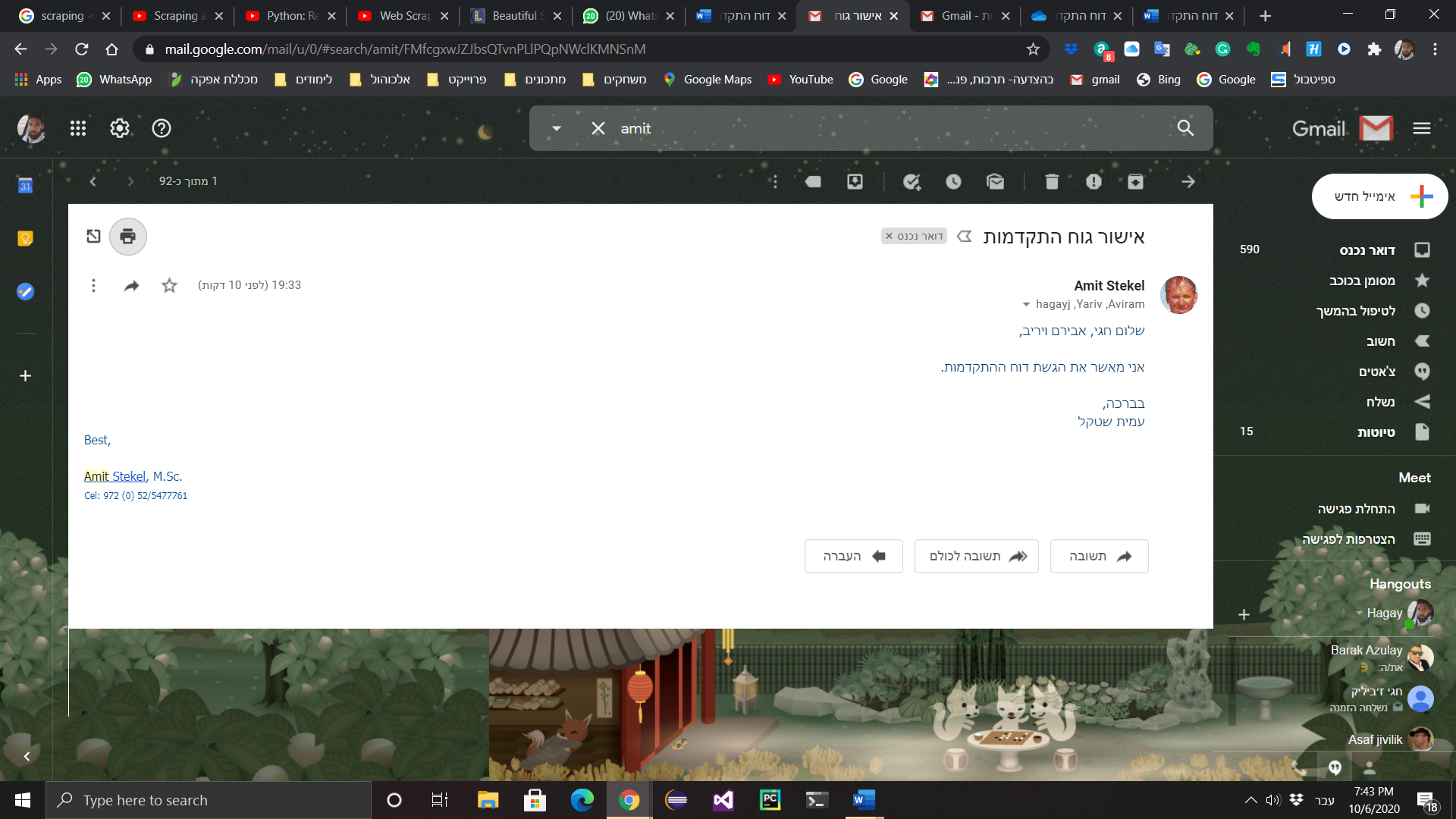
**המחלקה להנדסת תוכנה וחשמל**

שם הפרויקט: מערכת משולבת למסחר בשוק ההון

Project Name: Integrated Trading System

דוח התקדמות

|  |  |
| --- | --- |
| שם הסטודנט: | אבירם זוזות, חגי ז'ביליק, יריב מזרחי |
| מספר תעודת זהות: | 304965890, 036828069, 313576654 |
| שם המנחה: | עמית שטקל |
| חתימת המנחה: | אישור במייל בעמוד הבא |
| תאריך ההגשה: | 6.10.2020 |



תקציר הדו"ח:

מסחר בשוק ההון הינו מסחר בעל נפח מסחר גבוהה היכול להגיע ל50 מיליארד דולר בשנה, ע"י סוחרים החל מהאדם הממוצע עד לסוחרים מתקדמים, חברות ביטוח, בנקים ותאגידים גדולים. סוחרים ללא עזרה של מערכת אוטומטית מתקשים לעקוב אחרי מספר רב של מניות, תזמון מכירה וקניה, ניתוק רגשות ופחדים מהתנהגות השוק. עם זאת המסחר האוטומטי מתבסס על אלגוריתמים שונים, מגוונים הנותנים מענה לבעיות הנ"ל ונותנים חיזוי המשפרים את ההצלחה בהשקעה במספר מניות רב.

מטרת המערכת היא ליצר מערכת אוטומטית המבוססת על למידת מכונה, אשר תבצע שימוש וסחר בניירות ערך בשוק ההון. זאת תוך ניתוח מודלים פיננסיים, ניהול סיכונים וניתוח הודעות מתפרצות רלוונטיות ממקורות מידע ברשת ומדיה חברתית, כאשר שאיפת המערכת לייצר תשואה חיובית בשוק ההון, תוך מתן דגש על אוטומציה וחיסכון בזמן.

היעדים הם:

* 1. איסוף נתונים - בקשת מידע וריכוזו, מאתרי מדיה ומשוק המניות האמריקאי לאורך 20 השנים האחרונות.
  2. עיבוד מידע ראשוני - השלמת מידע ותיקונו.
  3. לימוד מודל באמצעות למידת מכונה - ייצור מודל על בסיס המידע שנאסף היודע לתאר את עבר התנהגות המניה וידע לחזות את העתיד.
  4. דירוג מניות - הערכת ניירות ערך בזמן אמת לפי פרמטרים סטטיסטיים (נתוני מסחר בורסה) ודינמיים (ניתוח טקסטים במדיה חברתית).
  5. ניהול מסחר - פיתוח תהליך פקודות מסחר יומיות בבורסה.
  6. ניהול סיכוני תיק - בניית מודל מתאים ליישום תהליך ניהול סיכונים.
  7. הצגה - בניית ממשק משתמש להגדרת והפעלת המערכת.

תכנון סביבת העבודה שנבחרה היא .NET כאשר הIDE שנבחר הוא Visual Studio. בחרנו בסביבה זו בעיקר בגלל נוחות, ניסיון עבר, תמיכה במספר שפות וקהילה רחבה. שפות הפיתוח שנבחרה היא Python וC#, כאשר Python היא השפה הנפוצה ביותר להבאת נתונים, בניית מודלים. בנוסף קיים המון מידע וניסיון צבור ברחבי הרשת כאשר העלות היא חינם.

במערכת המתוכננת בצד הClient נבחרה פלטפורמת Angular של Google הפופולרית בעולם הWeb עם סביבת העבודה Visual Studio שהוזכרה למעלה. בצד הServer השימוש יהיה בסביבת Net Framework של Microsoft, ובעזרת API תהיה משיכת המידע מהבורסה ורשתות חברתיות.

תכנון כריית המידע תבוצע ע"י שפת Python עם PyCharm, ובעזרת ספריות כמו nltk, keras, scikit-learn

**ריכוז שינויים:**

1. במסגרת כריית המידע, נבחנו כלי כרייה נוספים, חלקם אוטומטיים מבוססי web, וחלקם לפי דרישה של הלקוח בתצורה מותאמת לכל אתר. כמו-כן נבחנו גם שפות תכנות אחרות. הוחלט על השארת הבחירה הראשונית בשפת Python ובעזרתה להשתמש בספריות רלוונטיות כגון BeautifulSoup, Webdriver, selenium. כאשר בניית הזחלן תתקדם יוחלט באיזו ספרייה להשתמש. הסיכונים העיקריים הם שינוי מבנה האתרים וצורת משיכת המידע(API) ,אלגוריתם בנוי לכרייה ספציפית ואם מבנה האתר ישתנה האלגוריתם כבר לא יהיה יעיל). סיכון נוסף הוא מציאת אנשי מפתח במדיה היכולים להשפיע בצורה קיצונית על מחיר המניה. הדרכים להתמודד עם הבעיה היא לעקוב אחרי אנשי מפתח משפיעים לפי החדשות (דונלנד טרמפ, אלון מאסק, וורן באפ, מנכ"לים של חברות ותאגידים גדולים) ובעתיד לתת לאלגוריתם Deep Learning לבצע את הניתוח בעצמו לפי אתרי מפתח ולהחליט למי יש השפעה לאור השפעות עבר.

בעיה מיוחדת שעלתה בנושא היא שהסטודנט שאחראי על תחום כריית המידע הוקפץ בצו-9 למילואים לתקופה ממושכת שיצרה פער במוכנות הSOW וגם בבניית דו"ח ההתקדמות. הבעיה היא שעיכוב בחלק זה של הפרויקט משפיע באופן ישיר ויכול לעכב את התקדמות שאר החלקים בפרויקט.

1. בהיבטי משימות הנוגעות לטקסט (ניתוח טקסט, הבנה סמנטית, חילוץ ישויות ואימון מודלים) נבחנו מספר ספריות PYTHON, המסוגלות למלא את המשימות המבוקשות, ביניהן NLTK, SPACY, ALLEN NLP, SPARKNLP, STANFORD. חלקן לא עמדו בכל המשימות הנדרשות ונפסלו וחלקן היו בשימוש מצומצם לעומת האחרות. לבסוף הוחלט להתקדם עם ספריית NLTK, היודעת לעמוד בכלל המשימות (חילוץ ישויות, קשרים, פירקו משפטים, ניתוח תחבירי ואימון מודלים) והיא ספריית בעלת השימוש הרחב ביותר כיום בתחום הNLP. בשלב ראשוני, על מנת לבצע ניתוח טקסט מבלי לחכות לכריית המידע, נפנה אל תוצרים מוכנים הנמצאים כיום ברשת וכך נוכל להפיק תובנות וניסיון כאשר המידע שאנו מעוניינים בו יהיה מוכן. סיכון בהיבט הזה הוא הבנה סמנטית, שכל עולם הNLP מנסה לפענח סמנטיקה בצורה גבוהה(בעיית הציניות למשל) מכיוון שזאת בעיה גלובלית שעדיין אל נפתרה נוכל להסתכל על כמות רבה של דיווחים לכל מניה ועל כמה אנשים שנתייג כבעלי אמינות גבוהה ולא נסתמך רק על דעה אחת.
2. בתחום המודל: נשתמש בשפת Python בהתאם לחלקים הקודמים בפרויקט.

מבחינת בחירת הספריות והמודל למימוש נבחנו מספר אפשרויות המסוכמות בטבלה הבאה:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **שם הספרייה** | **תחום התמחות** | **פופולריות** | **שימוש עיקרי** | **עדכני** | **דירוג התאמה** |
| TensorFlow | רשתות נוירונים | 5 | גרפים | 5 | 4 |
| **Scikit-learn** | **למידת מכונה, למידה עמוקה** | **4** | **נתונים** | **4** | **5** |
| PyTorch | למידת מכונה ורשתות נוירונים | 3 | גרפים | 5 | 2 |
| Keras | למידת מכונה | 4 | הדגמה | 4 | 1 |
| Shogun | למידת מכונה | 1 | אקדמי | 1 | 3 |
| Torch | למידת מכונה | 3 | חברות ענק | 2 | 3 |

הוחלט להתקדם עם ספריית Scikit-learn מכיוון והיא ממוקדת לתחום למידת המכונה, מכילה פונקציות רבות שעוסקות בניתוח נתונים, היא יחסית פופולרית ועדכנית. סיכונים אפשריים בתחום המידול:

-תוצאות לא מספקות יכולות להגיע משיטת המודל או האלגוריתם, הפתרון שמוצע הוא ניסוי וטעיה בשיטות וספריות שונות לקבלת פתרון מיטבי.

-אי מציאת קשר ישיר בין עלייה/ירידה של מניה לניתוח ידיעות טקסט, הפתרון שמוצע הוא שימוש במידע סטטיסטי וטכני במקביל להשבחת התוצאות.