דו"ח מסכם

JobMe

מגישים:

שי ארץ קדושה 203276258

חן ארזי 307875633

תוכן

הגדרת המשימה ומוטיבציה

כסטודנטים בתואר ראשון להנדסת מערכות תוכנה ומידע נחשפנו לתהליך של התמיינות למשרות שונות בתחום ובנוסף שמענו חוות דעת על התהליך מבוגרים משנים קודמות. כל סטודנט או בוגר שנמצא בתהליך גיוס למשרה מתבקש לרוב לעבור לפחות ראיון מקצועי אחד בו הוא נבחן בנושאים שונים שהצוות המגייס חושב שעל אותו מועמד להכיר את אותם בצורה טובה לצורך איוש המשרה. בנוסף, המועמד לרוב מחפש אנשים שעברו ראיון בעבר באותה חברה ולשמוע מהם חוות דעת על איך היה הראיון המקצועי שלהם וכך לקבל תמונה כלשהי לקראת הראיון הצפוי להם.

הבעיה המרכזית

לקראת כל ראיון מקצועי שנקבע למועמד עבור המשרה אליה הוא מתמיין הוא מתכונן לראיון בצורה אינטסיבית על מנת להיות מוכן לקראתו בצורה הטובה ביותר. כחלק מתהליך זה המועמדים לרוב מחפשים את הנושאים הרלוונטים ללמוד לקראת הראיון, קוראים חוות דעת שונות על ראיונות שמועמדים אחרים עברו עבור משרה דומה באותה חברה מגייסת ומתייעצים עם חברים ומכרים שעברו בעבר ראיון בחברה. תהליך זה לרוב הוא לא ממוקד, כלומר למועמדים בדרך כלל לא ניתן כלל מיקוד לימודי לקראת הראיון וזמן החיפוש של שאלות רלוונטיות וחוות דעת במקורות שונים עשוייה לקחת לא מעט זמן ובנוסף לעיתים לא מביאה בהכרח את המידע הרלוונטי שהמועמד צריך. בנוסף כמות החוות דעת שהמועמד משיג לרוב היא מצומצת ולא נותנת מספיק אינדקציה לקראת הראיון אותו הוא עתיד לעבור.

המטרה

ליצור מערכת נוחה ומהירה שתתן עבור מועמדים שנמצאים לקראת ראיון מקצועי למשרה מסוימת את הנושאים העיקריים שבהם עליהם להתמקד לקראת הראיון לפי אופי המשרה, זהות החברה המגייסת ושאלות עבר. בנוסף נרצה שהמועמד יקבל מהמערכת תמונה רחבה יותר על תחושות של מועמדים אחרים שכבר עברו ראיון למשרה דומה באותה החברה. נתונים אלו שהמערכת תיתן יקצרו וימקדו את תהליך הלמידה של המועמד ותכין אותו בצורה הטובה ביותר לקראת הראיון המקצועי אליו הוא לומד. כלומר, המועמד יכניס את שם החברה, מיקומה ואת התפקיד אליו הוא עתיד להתמיין והמערכת תייצר למועמד דף הכנה לראיון הכולל את התובנות הנ"ל, שיסעו לו להתכונן לראיון בצורה המיטבית.

סקירת התחום

כחלק מתהליך פיתוח המערכת ביצענו סקירה לטובת הכרתן של יכולות של מערכות שונות שקיימות שעוזרות בתהליך היערכות לקראת ראיון עבודה מקצועי ובנוסף סקרנו את הטכנולוגיות שבהן השתמשו במערכות אלה או במערכות אחרות בעליי אופי דומה למערכת שלנו.

סקירת מערכות/ אתרים מרכזיים

Glassdor- אתר ואפליקציה בעזרתם משתמש יכול למצוא שאלות עבר וחוות דעת לפי משרה ושם חברה. האתר מציג כמות גדולה של שאלות מראיונות עבר וחוות דעת ועל המשתמש לעבור על כל תוצאות החיפוש על מנת לקבל תמונה איזה נושאים עליו ללמוד לפי השאלות שנשאלו ואיך היו תחושות של מועמדים אחרים במהלך ראיון מקצועי באותה חברה.

The-worker- אתר מותאם בשפה העברית שמאפשר לבצע חיפוש לפי שם חברה וסוג המשרה ויציג שאלות עבר מקצועיות בהתאם שאנשים מצאו לנכון לשתף באתר.

JobHunt- אתר בעברית בו המשתמש רשאי לבחור חברה מתוך שמות החברות המופיעות באתר ולאחר מכן יוצגו לו שאלות עבר שאנשים שיתפו בפורום המיועד עבור אותה חברה.

ניתן לראות כי המערכות והפלטפורמות הקיימות לא עונות בהכרח על כל הצרכים אותם אנו מחפשים ולכן קיים צורך בפיתוח המערכת שלנו שתמקד את המועמד בנושאים מרכזיים לשאלות מקצועיות ותיתן לו תמונה רחבה על חוות דעת של מועמדים אחרים לאותה משרה בזמן קצר.

	מאפשר חיפוש לפי משרה	מאפשר חיפוש לפי חברה	מאפשר לחפש לפי מיקום	מציג נושאים מרכזיים לפי השאלות	מציג נושאים ניתוח מסוכם של חוות דעת	שפת ממשק משתמש
Glassdoor	✓	✓	✓	✗	✗	אנגלית
The-worker	✓	✓	✗	✗	✗	עברית
JobHunt	✗	✓	✗	✗	✗	עברית
JobMe	✓	✓	✓	✓	✓	אנגלית

המערכות שסקרנו לא עונות על כל הצרכים שאנו דורשים. לא מצאנו מערכת שמשלבת את אחסון והצגת המידע ושבנוסף מאפשרת הפקת תובנות מהמידע- כמו הוצאת נושאים מרכזיים משאלות עבר וניתוח חוות דעת.

על כן, חילקנו את חלק ה-backend של התוכנה שלנו ל-2 חלקים עיקריים:

1. איסוף המידע ועיבוד מקדים שלו
2. הפקת תובנות מהמידע

סקירת טכנולוגיות רלוונטיות

נבצע סקירה נפרדת לכל אחד מחלקי הפרויקט.
backend: איסוף המידע ועיבוד מקדים שלו וניתוח המידע והפקת תובנות.
frontend: בחירת טכנולוגיה שנממש איתה ממשק משתמש.

-backend

איסוף המידע ועיבודו-

על מנת ליצור את המערכת שלנו, רצינו להשתמש במידע של האתרים הקיימים וע"י עיבוד שלו ליצור את המערכת שלנו. לצורך הפקת המידע היינו זקוקים ל-API מפורט שבעזרתו נוכל לקבל את המידע מהאתרים. מפני שלא היה API שכזה, חיפשנו טכנולוגיה אחרת שתסייע לנו בהפקת המידע.

מצאנו open source שעושה שימוש בחבילה של selenium ובאמצעות ChromeDriver מייצר scraper ששואב מידע מהאתר של Glassdoor. Glassdoor הינו האתר שמכיל את המידע המתאים ביותר לפרויקט שלנו ולכן התבססנו בחלקים מהקוד הפתוח הזה על מנת לכתוב את הקוד שלנו.

Selenium WebDriver – חבילה שמאפשרת ליצור סקריפט (רובוט) שמפעיל דפדפן, מאפשר מעבר בין דפים וקבלת מידע מתוכם. חבילה זו זקוקה לקובץ הרצה (ChromeDriver) על מנת לרוץ.

ChromeDriver– http://chromedriver.chromium.org/home
קובץ שלאחר התקנתו מאפשר לשלוט בדפדפן של chrome. אין אפשרות להריץ סקריפט של selenium בלי להשתמש ב-chromeDriver.

Open source – https://github.com/MatthewChatham/glassdoor-review-scraper
הקוד הפתוח שמצאנו בונה סקיפט (רובוט) שנכנס לאתר של glassdoor , מנווט בתוכו לדף ספציפי וע"י תגיות html שואב את המידע. הקוד הפתוח עשה scrape ל-reviews בלבד.
אנו היינו צריכים לשנות לגמרי את הקוד על מנת שנוכל להפיק מידע אודות interviews ולא אודות reviews.

ניתוח המידע והפקת תובנות-

על מנת ליצור דף הכנה מיטבי לראיון עבודה על סמך המידע שהפקנו. חיפשנו מודלים במאמרים ובפרויקטים שונים בהם יש שימוש בטכנולוגיות בהן אנו יכולים להשתמש כדי לנו להשיג מטרות אלו:

- Topic Modeling
- Sentiment Analysis

Topic Modeling- זהו תחום שעוסק במודלי שפה סטטיסטיים המאפשרים לזהות נושאים מופשטים באוסף של מסמכים. אחד המודלים העיקריים בתחום הוא LDA בו גם בחרנו להשתמש במערכת שלנו.

(Latent Dirichlet Allocation) LDA- מודל זה מאפשר לזהות נושאים מופשטים ביחס לכלל המילים מתוך מספר מסמכים ובנוסף בהינתן נושאים מוגדרים המודל יכול לשייך מסמך או משפט

לאחד מהם. המודל מקבל טבלה/ מטריצה שמייצגת אילו מילים מופיעות בכל מסמך (לאחר ניקוי של מילות קישור- stopwords), מספר נושאים ומספר מילים לכל נושא ובעזרת קלט זה לומד לייצר שתי טבלאות. טבלה אחת שמקשרת כל מילה לנושא ונותנת לאותה מילה משקל בנושא וטבלה נוספת שמשייכת כל מסמך לנושא. יש לציין שהנושאים הם מופשטים ולא מופרשים כלומר מתוך המילים שהמודל בוחר לשים תחת כל נושא נבין את ההקשר של נושא.
אנו נשתמש במודל זה כדי לזהות ממאגר הגדול של השאלות שהוצאנו עבור משרה בחברה מסויימת את המילים/ נושאים בעלות משקל הגבוה ביותר בנושא ומילים אלו בעצם יהיו כביכול מילות מפתח שימקדו את המועמד בלמידה לקראת השאלות שעשויות להישאל בראיון.

דוגמה להמחשה:
קלט מצומצם של שאלות מחברה, מס' נושאים- 1, מס' מילים בנושא-2:

-I asked about array and Lists
-Convert array to list
-Sort list

פלט:

Topic 1- {0.65 List, 0.35 Array}

Sentiment Analysis- טכנולוגיה זו מאפשרת לנתח מילים, משפטים או מסמכים ולתת להם תיוג בהתאם לבעיה. כשמנתחים נתוני שפה באמצעות טכנולוגיה זו ניתן לבצע אותה על מידע מתויג ומידע לא מתויג כאשר עבור כל סוג מידע ניתן להשתמש בטכניקות שונות כדי לסווג את המידע. במערכת שלנו אנו מחלצים את המידע מאתר Glassdoor כאשר הוא לא עבר ניתוח מקדים כלשהו ולכן הוא לא מתויג, כלומר אנו נשתמש בטכניקות מתאימות עבור ניתוח מידע לא מתויג.

אחת הדרכים הנפוצות לסווג נתוני שפה לא מתויגים זה באמצעות שימוש במילונים שבנויים בהתאם לבעיית הסיווג. דרך אחת היא לבנות מילון מותאם לבעיה ולהשתמש בו לצורך הסיווג דבר שלרוב עשוי לקחת המון זמן ומצריך ידע רב בניתוח נתוני שפה. דרך נוספת ומאוד מקובלת היא להשתמש במילונים מוכנים שמותאמים לבעיות סיווג נפוצות. במערכת שלנו בחרנו לסווג חוות דעת מראיונות עבר לחוות דעת חיובית ושלילית ולכן העדפנו לחפש מילון קיים שכבר נועד לשימוש עבור בעיית תיוג זו מאשר לבנות מילון מותאם לנתונים. בחרנו לעבוד בדרך זו מכיוון שכמו שצוין קודם תהליך של בנית מילון עשוי לקחת זמן רב ויכול להיות לא מדויק בהתאם ליידע המקצועי בתחום שיש לנו כרגע.

SentiWordNet- מילון די נפוץ לשימוש בבעיות סיווג נתוני שפה לקונוטציה שלילית וחיובית. בעזרתו ניתן לתת לכל מילה ציון חיובי, שלילי או נטרלי בהתאם למה שמוגדר במילון ולהשתמש במידע זה כדי לתייג משפטים ומסמכים לאופי שלילי או חיובי. בחרנו להשתמש דווקא במילון זה משום שכאשר סקרנו טכנולוגיות לביצוע Sentiment Analysis שמנו לב שהשימוש במילון זה עבור בעיות סיווג נתוני שפה לא מתויגים הניב תוצאות יחסית טובות ובנוסף יחסית קל לייבא אותו לצורך שימוש בפיתוח מערכות אחרות שמבצעות Sentiment Analysis על המידע שלהם.

<u>frontend -</u>
בבואנו לבחור חבילת ממשק משתמש התלבטנו בין kivy שנלמד בקורס לבין חבילת tkinter הפופולרית.

לאחר סקירה הבנו כי השימוש בחבילת tkinter הוא יותר קל למשתמש ולא דורש התקנות רבות נוספות מעבר לחבילה סטנדרטית של python.

לעומת זאת, kivy מאפשר גמישות רבה יותר לבניית ממשקים מורכבים שנתמכים ע"י מכשירים שונים.

לבסוף החלטנו להשתמש ב- tkinter משתי סיבות עיקריות:

1. ממשק המשתמש שלנו לא מורכב.
2. משתמשי התוכנה שלנו יכנסו לתוכנה ממחשב בבואם ללמוד לראיונות עבודה ולא ממכשירים אחרים, על כן לא ראינו צורך להשתמש בממשק שמתאים למכשירים רבים.

הוראות התקנה

חבילות, התקנות וקבצים נוספים-

כמצוין הנ"ל, המערכת ממספר חלקים, כל חלק צריך התקנות אחרות:
ראשית יש להתקין Python 3

איסוף המידע-

- התקנת pandas, selenium, numpy
- התקנת Chromedriver (קובץ ההרצה- chromedriver.exe נמצא כבר בקוד).
- למטרת איסוף המידע, אתר glassdoor דורש משתמש קיים באתר.
  יצרנו משתמש מנהל לצורך התוכנה שלנו שדרכו נתחבר לאתר.
  פרטי המשתמש נמצאים בקובץ - secret.json:
  {"username": "tofindmyjob@gmail.com", "password":"ToFindMyJob123"}
  במידה ותרצו לשנות את הפרטים מפרטי המנהל למשתמש glassdoor אחר, ניתן לקנפג פרטים אחרים בקובץ זה.

ניתוח המידע -

- להרצת מודל topic modeling יש להתקין חבילות: nltk, genism ולייבא מהן מילון stopwords, WordNetLemmatizer ו- corpora.
- חבילת genism משתמשת בחבילת - smart_open, יש צורך להתקין אותה באופן עצמאי כך:

```
pip3 uninstall smart_open
pip3 install smart_open
```

- להרצת מודל Sentiment Analysis יש להתקין את החבילה nltk ולייבא ממנה מילון stopwords ומילון SentiWordNet. כמו כן גם את matplotlib.

ממשק משתמש -

- התקנת pillow

כל החבילות הדרושות הנ"ל נמצאות בקובץ – requirements.txt

מערכת הפעלה-

המערכת פותחה ורצה על מערכת הפעלה של win10.

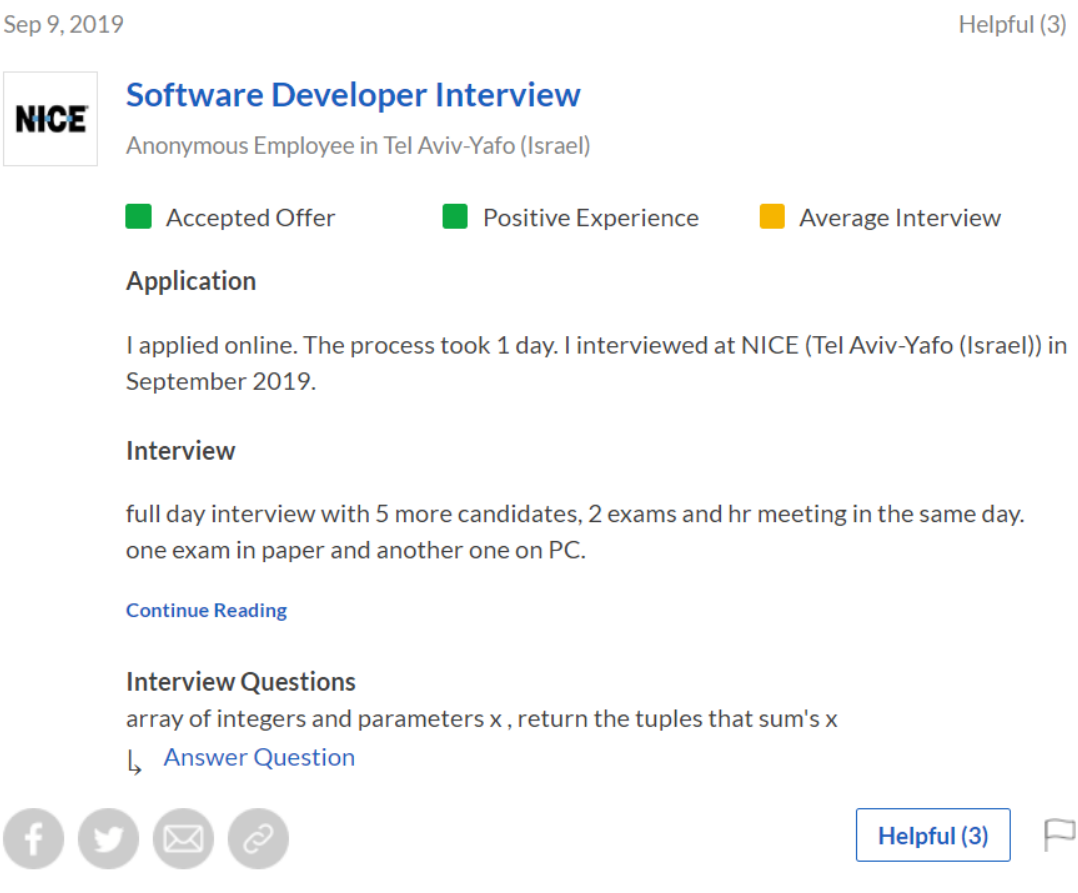
מאגרי מידע

כפי שתיארנו בשלב הסקירה הטכנולוגית, לא השתמשנו במאגר מידע קיים אלא בנינו את ה-data מאתר Glassdoor. בחרנו דווקא באתר זה מפני שהוא הכיל את כל המידע שרצינו.

כזכור, משתמש המערכת שלנו מכניס שם חברה, מיקום ושם משרה אליה הוא עתיד להתמיין. לכן, ה-data הרלוונטי הוא מידע אודות ראיונות עבודה למשרה זו (או משרות דומות) באותה החברה.

באתר Glassdoor, מרואיינים רבים מעלים מידע אודות ראיון שהם עברו.
ניתן לחפש מידע אודות ראיונות בחברה מסוימת ע"י שם חברה והמיקום שלה. כלומר, לא ניתן לחפש את המידע עבור משרה מסוימת כפי שאנחנו צריכים למטרת הפרויקט שלנו. על כן, בתהליך הכנת המידע, אנו סיננו את המשרות והכנסנו רק את המידע של הראיונות הרלוונטיים למשרה.

המידע באתר מוצג באופן הבא:



מכל ראיון שמרנו את המידע הבא:

Job title- שם המשרה.
Interview- חוות דעת כללית מהראיון או מהלך הראיון.
Interview Questions- שאלות זכורות מהראיון.

שמרנו את כל המידע ב-dataframe ולאחר מכן ייצאנו לקובץ csv, על מנת שנוכל להשתמש במידע זה בקלות ובמהירות לצורך הפקת התובנות.

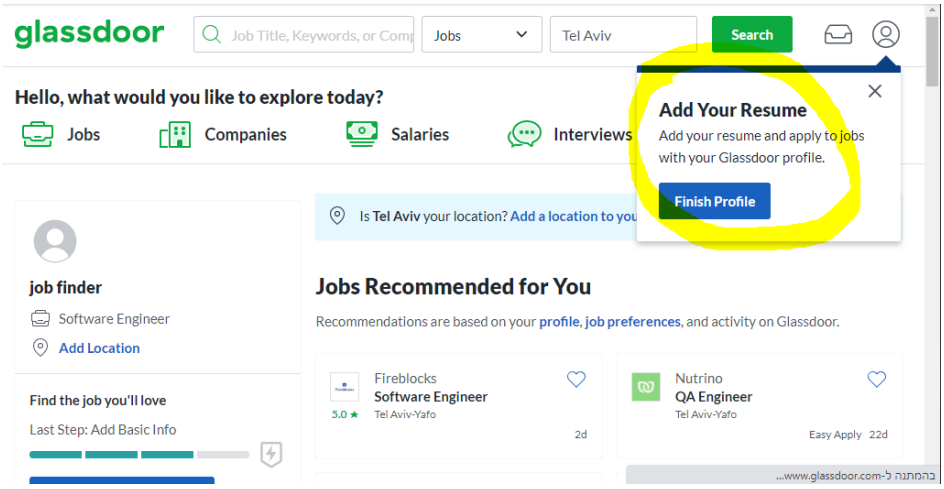
מבנה קובץ ה-csv:

	Job title	Interview	Interview Questions
0	Software Engineering Interview	Behavioral screen (phone) followed by 2 technical inte	Given to lists, return a list of all common elements Given the most recent (
1	Software Engineer Interview	got an interview when attending a conference but I wa	tell me about yourself, please?
2	Software Engineer(Internship) Interview	Pretty painless for a explore internship interview. I had	A question about string interpolation
3	Software Engineer Interview	There was a coding question at the start of the virtual	Talk about time you worked on a team project? 6 Answers
4	New Grad Software Engineer Interview	The timeline was not as quick as I had anticipated. He	Create a minesweeper board given board dimensions and number of mines
5	Software Engineer(Internship) Interview	Over the phone - coding module; If passed, you will g	Q: Coding a Chess move
6	Software Engineer Summer Intern Interview	Got to their virtual onsite. One of the interviewers was	not engaged in seeing my solution to the technical problem. We agreed on
7	Software Engineer Interview	Applied to the role and was asked to fill out a survey (	Describe you experience with x, y, z...
8	Software Eginner Interview	Online assessment, phone screen, questions already	NDA, cannot disclose the questions
9	New Grad Software Engineer Interview	I had one technical interview that lasted for an hour ar	Copy a linked list Make an iterator
10	Software Engineer Interview	Great ! The requirements are fair. You need to practic	Graphs, 2D Matrix, String manipulation 2 Answers
11	Software Engineer - Intern Interview	Initial phone call screen that was pretty much all behav	Q: Teach us about a technical concept and pretend you're teaching it to a c
12	Software Engineer II Interview	It is ok. All practical questions, no tricks. Write small fi	Reverse string. Reverse linked list. Design vending machine system.
13	Software Engineer New Grad Interview	I have a screening 30 minute phone call. The worst int	Example of questions: what is the difference between good code and great
14	Software Engineer Interview	They gave me a 30 minute coding challenge where I h	Why did you write your code this way? 2 Answers
15	Software Engineer - New Grad Interview Interview	1: Recruiter Screening 2: Virtual interviews 3-4 intervi	Why do you want to work for Microsoft? Easy/Medium Leetcode 2 Answe
16	Software Engineer Interview	The interviewers were all super nice (except one guy v	JSON file parsing problem with follow up Dynamic Programming LC medi
17	Software Engineer New Grad Interview	(August 2020) Submitted application online without re	1. A Software Manager asking about projects and technical decisions withir
18	Software Engineer(Internship) Interview	Applied online in August. Got a phone interview in late	Linked List LC Easy, BST LC medium
19	Full-time Software Engineer Interview	Scheduled a 30 minute behavioral interview. HR repre	What was your favorite Microsoft product and what would you improve abc
20	Software Engineer Interview	Didn't get an offer but it was a great experience overa	Questions about Tree, Linked List...

<u>אתגרים במהלך יצירת המידע:</u>

כפי שנלמד בקורס, כאשר אנו בונים מערכת שמסתמכת על אתרים חיצוניים ישנם סיכונים שאנחנו צריכים לקחת בחשבון. נפרט על מספר סיכונים שנתקלנו בהם במהלך תהליך יצירת ה-data:

1. כל אתר בנוי באופן שונה, על כן יצירת קוד גנרי שמתאים לכמות רחבה של אתרים הוא אתגר גדול.
2. תחזוקה של הקוד- נכונות הקוד ופעולתו התקינה תלוי במבנה האתר, במידה ומבנה האתר משתנה- יש לשנות את הקוד בהתאם.
3. בשימוש בסקריפט שמשתלט על הדפדפן (שימוש ב-selenium כפי שהוסבר בשלב סקירת הטכנולוגיה), ישנו חשש כי האתרים יתנהגו כל פעם באופן אחר.
לדוגמא: לאחר מספר הרצות של הקוד, האתר הציג בקשה למשתמש שיש להגיב עליה. ניתן לראות בתמונה מתחת כי האתר מבקש מהמשתמש לעדכן את הפרופיל שלו. זו בקשה לא צפויה ולכן קוד כתוב מראש לא יידע להתמודד עם שינוי זה.



4. בשימוש בסקריפט שמשתלט על הדפדפן יש חשיבות לזמן התגובה של השרת, על כן הקוד שלנו צריך להתחשב בכך ולהמתין לתשובה של הדפדפן. תזמון זה אינו טרויאלי ולכן מוסיף סיכון פוטנציאלי שיש לקחת בחשבון.

5. בגלל סיבה 4- תהליך יצירת המידע ע"י ה-scraper לוקח זמן רב. זהו אתגר שיש לקחת בחשבון כאשר אנו מייצרים מידע.

מפאת כל הסיבות הנ"ל, ולאחר התייעצות עם מרצה הקורס- החלטנו להתמקד ב-3 משרות בחברות שונות בלבד. תעדפנו את ניתוח העומק של ה-data של מספר מצומצם של חברות על פני השמשת שלב איסוף הנתונים על פני חברות רבות.

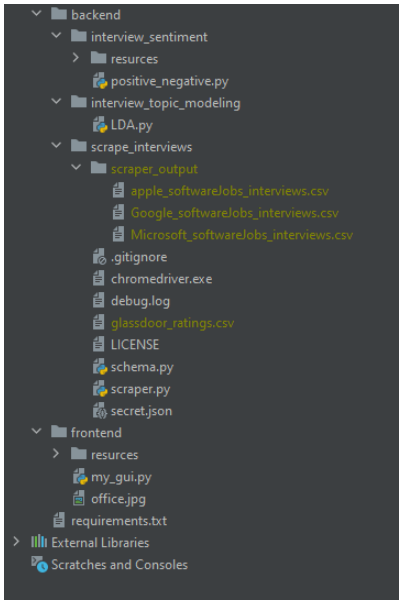
על כן החלטנו לייצר קובץ csv עם הנתונים ולהתמקד בראיונות לתפקיד software engineer ב-3 חברות ספציפיות שלהן הכנו מאגר מידע מראש.

החברות הן: Google, Apple, Microsoft.

שימוש פונקציונלי

הפרויקט שלנו בנוי מ-2 חבילות עיקריות:

1. Backend
2. Frontend



3 הכלים שבנינו- כלי לאיסוף המידע, כלי להרצת מודל topic modeling וכלי להרצת sentiment analysis משמשים את הכלי שלנו ע"י כך שהם מייצרים את תוצריו.

כפי שהוזכר בשלב מאגרי המידע, בחרנו להתמקד במערכת שלנו ב-3 חברות עיקריות: Microsoft, apple ו– google ולייצר עבור כל אחת מהן דף הכנה לראיון לתפקיד software engineer.

מפני שבחרנו להתמקד בנתונים אלו בלבד בפרויקט שלנו (ולאפשר כמובן הרחבת המודל ע"י API של הפונקציות שיפורט בהמשך), בנינו ממשק משתמש ייעודי להתמקדות זו. ניתן להפעיל אותו על ידי הרצת קובץ Gui.py שנמצא תחת חבילת frontend.

לאחר הפעלת ממשק המשתמש, יוכל המשתמש לבחור אחת מ-3 החברות הנ"ל, ובעבור על חברה יוכל לסמן את המיקום והתפקיד. המשתמש יראה את תוצרי המערכת עבור הנתונים שהכניס. פירוט התוצרים מופיע תחת כותרת ה-תוצאות.

על מנת להריץ את המערכת על פני משרות וחברות נוספות ניתן להשתמש ב- API הבא:

- איסוף הנתונים ע"י הרצת קובץ scraper.py שנמצא ב- חבילת scrape_interviews.
  ההרצה תפתח console application שמקבלת כקלט:
  1. שם חברה
  2. מיקום חברה
  3. תפקיד בחברה

  המערכת תיצר קובץ csv שישמר בנתיב: backend/scrape_interviews/scrper_output

- הרצת מודל topic modeling -LDA ע"י קובץ LDA.py שנמצא בחבילת interview_topic_modeling.

לפני הרצת הקובץ יש לעדכן את הנתיב שמופיע ב-main על פי קובץ ה-csv החדש שנוצר (בשלב איסוף הנתונים).

- הרצת מודל Sentiment Analysis ע"י הרצת קובץ positive_negative.py שנמצא בחבילת interview_sentiment:
לפני הרצת הקובץ יש לעדכן את הנתיב שמופיע ב-main על פי קובץ ה-csv החדש שנוצר (בשלב איסוף הנתונים).

תוצאות

בהתאם למטרות המערכת שהצגנו קודם בחנו את תוצאות השימוש במערכת לפי מספר יעדים מרכזיים שהגדרנו:

- האם המילים שנבחרו בעזרת שימוש ב-Topic Modeling רלוונטיות ומאפשרות מיקוד בלמידה לקראת ראיון למשרה שהוגדרה במערכת.
- האם חוות הדעת שתויגו באמצעות שימוש בטכנולוגיית Semantic Analysis מסווגות בצורה הגיונית את חוות הדעת של מרואייני עבר.

<u>להלן התוצאות:</u>

Topic Modeling:

עבור משרת Software Engineer מודל ה-LDA לביצוע Topic Modeling עבור מאגר השאלות של החברות: Apple, Google ו-Microsoft הניב תחילה את התוצאות הבאות:

Apple- [String, S, LeetCode, List, Tree]

Google- [Coding, Question, Structure, Q, Find]

Microsoft- [List, 2, Linked, Answer, Task]

ניתן לראות שיש לא מעט מילים שחזרו שאינן קשורות כלל לתחום במתבקש ואפילו חלקם הם אותיות או מספרים שכנאה חזרו על עצמם בחלק מהשאלות.

כדי לנסות להתמודד עם הבעיה ולשפר את התוצאות הרצנו את המודל מספר פעמים וכאשר ראינו בתוצאות מילים, אותיות או מספרים שלדעתנו אין להן קשר ל -Software Engineer הוספנו למילים המסוננות וכך צמצמנו את המילים שלדעתנו מטות את התוצאות.

לאחר מכן הרצנו שוב וקיבלנו את התוצאות הבאות:

Apple- [String, Array, LeetCode, List, Tree]

Google- [Coding, Algorithm, Structure, Data, Find]

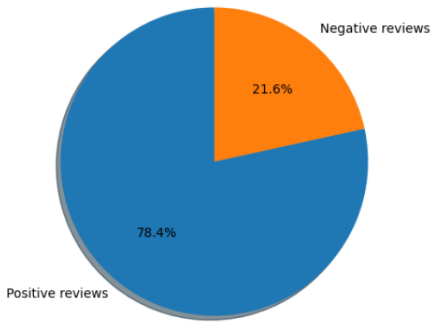
Microsoft- [List, Number, Linked, Array, Task]

ניתן לראות שכעת יש שיפור בתוצאות, והמילים שחוזרות עבור המידע על שאלות של כל אחת מהחברות אכן יותר רלוונטי ברובו וקשור לנושאים עבור משרת Software Engineer (למשל String, Array List, Tree, Linked). כמו כן יש מילים שעדיין האלגוריתם מחזיר שלא מגדירות באופן חד משמעי נושא שקשור Software Engineer כמו למשל המילה Find שאומנם יכולה לסייע לנחש מה הפעולה שצריך לעשות בשאלה מסוימת אבל לא ממקדת בנושא מסוים שזוהי מטרתו של האלגוריתם. אחרי השיפור שביצענו ניתן לומר שעל סמך התוצאות החדשות שהתקבלו מהמידע של שלושת החברות שבחרנו אפשר לומר שאכן המטרה הושגה והמודל עובד בצורה יחסית טובה.

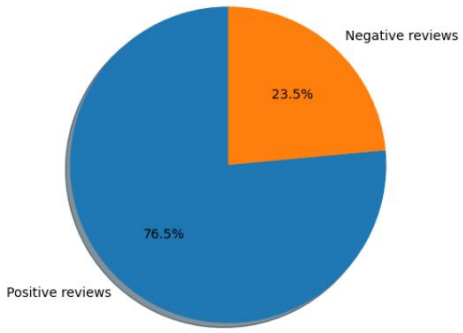
Sentiment Analysis:

עבור משרת Software Engineer השימוש בעזרת מילון של SentiWordNet לביצוע Sentiment Analysis עבור מאגר חוות הדעת של החברות: Apple, Google ו-Microsoft הניב את התוצאות הבאות:

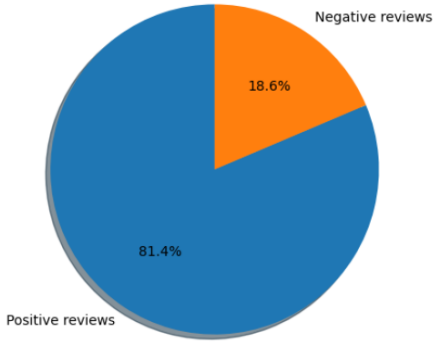
Apple



Google



Microsoft



עבור חברת Apple , מתוך 102 חוות דעת שחילצנו 80 חוות דעת סווגו כחיוביות ו- 22 חוות דעת סווגו כשליליות, 78.4% לעומת 21.6%.

עבור חברת Google , מתוך 102 חוות דעת שחילצנו 78 חוות דעת סווגו כחיוביות ו- 42 חוות דעת סווגו כשליליות, 76.5% לעומת 23.5%.

עבור חברת Microsoft , מתוך 102 חוות דעת שחילצנו 83 חוות דעת סווגו כחיוביות ו- 19 חוות דעת סווגו כשליליות, 81.4% לעומת 18.6%.

מכיוון שהמידע שחילצנו הגיע לא מתויג לא יכלונו לאמוד באמצעות מטריקה כלשהי את דיוק הסיווג לחוות דעת חיוביות ושליליות. לכן על מנת להבין אם האלגוריתם ביצע בצורה הגיונית את הסיווג קראנו את כל ה-100 חוות דעת של כל אחת מהחברות ועשינו התאמה לסיווג שהאלגוריתם בחר לתת לאותה חוות דעת. מתוך השוואה זו ראינו האלגוריתם שבחרנו לביצוע Semantic Analysis ביצע בצורה יחסית טובה את הבחירה לחוות חיוביות ושליליות. אומנם יש לציין שיש מקרים שבהם קראנו את חוות דעת כלשהי ולנו בתור קוראים לא היה ברור אם היא חיובית או שלילית (אפשר להגיד שהיא הייתה נטרלית) ולא ברור באופן חד משמעי על סמך מה בחר האלגוריתם בחר לסווג את אותה חוות דעת לאחת משתי הקטגוריות. יש לציין כי מאחר ונאספו נתונים רק ל-100 חוות דעת יכולנו לבצע ניתוח בעצמנו אבל במידה והיה מספר רב יותר של חוות דעת היה קשה להסיק באופן חד משמעי ומהיר אם הסיווג אכן נעשה בצורה הגיונית.

מסקנות

עמידה בציפיות הכלי

לאחר שסיימנו לבנות את הכלי, נתנו למשתמשים פוטנציאלים (מכרים שמחפשים עבודה) להשתמש בכלי.

חוות דעת של המשתמשים הכילה את הנקודות הבאות:

- הכלי חוסך זמן רב בהתכוננות לראיון.
- הנושאים שהכלי מפיק הינם רלוונטיים ברובם ואפשר על פיהם להתכונן טוב יותר (לחפש שאלות באינטרנט בנושא וכך לייעל את תהליך הלמידה).
- חוות הדעת הכללית שמוצגת כגרף מסייעת לקבל תמונה מקיפה של משתמשים רבים על חווית ההתמיינות לתפקיד בחברה המסוימת.

נראה כי הכלי עמד במטרותיו הכלליות שהוגדרנו.

שיפורים עתידיים אפשריים לכלי

- היינו רוצים להרחיב את הכלי כך שיתמוך במשרות וחברות נוספות וכמו כן לבנות ממשק משתמש שיתמוך בחיפוש רחב יותר.
- כפי שתיארנו בשלב מאגרי המידע, שלב איסוף המידע הוא צוואר הבקבוק שנתקלנו בו בהקשר של הרחבת הכלי למשרות נוספות.
- ע"י השקעת משאבים נוספים אנו סבורים כי ניתן להתגבר על האתגרים שהוצגו בשלב איסוף המידע וכך להרחיב את המערכת על מנת שהכלי יעבוד בצורה מקיפה יותר.
- הרחבת החיפוש לחיפוש משרות אחרות שמשתייכות לאותו התחום.
- שיפור מודלי הלמידה ע"י-בניה או הרחבת מילונים שמיועדים לביצוע ניתוחים על מידע הקשור לחיפושי עבודה להשתמש בהם ובכך אולי לשפר את התוצאות המודלים.
- שיפור מודלי הלמידה ע"י- הרצת מודלים על data גדול יותר ועל חומרה חזקה יותר.