חיזוי של דירוג מסעדה חדשה   
לפי מסעדות קיימות

יקיר אוזן – 315009019 –[uzan.yakir@campus.technion.ac.il](mailto:uzan.yakir@campus.technion.ac.il)

שרי יצחק – 319011490 –[sarayyitzhak@campus.technion.ac.il](mailto:sarayyitzhak@campus.technion.ac.il)

**תיאור הבעיה:**

בכל יום בישראל נפתחות מסעדות רבות ונסגרות באותה מהירות. פעמים רבות זה נגרם עקב חוסר התאמה של המסעדה לאזור או לאוכלוסייה. דבר זה גורם לבעלי מסעדות הפסדים כספיים רבים.

נפתור בעיה זו על ידי חיזוי דירוג המסעדה לפני פתיחתה, כך יוכל הבעלים להחליט האם כדאי לו לפתוח את המסעדה, או האם הוא נדרש לעשות שינויים בסגנון המסעדה, מיקום המסעדה וכו'.

**תיאור הפתרון המוצע לבעיה:**

בפרויקט זה נתמודד עם הבעיה בתור בעיית למידה. בחרנו להשתמש בעץ החלטה רגרסיה.

**איסוף הנתונים:**

החלק החשוב ביותר בפרויקט מסוג זה הוא המידע שאנו משתמשים, מסד הנתונים שעליו אלגוריתם הלמידה שלנו מסתמך. במידה ומקור הנתונים שלנו יהיה לא אמין או שקרי אז המסווג שיצא לנו יהיה לא מוצלח.

את הנתונים אספנו עבור הערים: חיפה, הקריות, נשר וטירת הכרמל.

האיסוף נעשה מכמה מקורות מידע:

1. Google – עיקר המידע שלנו, ממקור זה אנו מקבלים מידע שימושי שיש למסעדות.   
   ממקור זה אנו מקבלים את התכונות הבאות:

* מזהה המסעדה לפי גוגל (place id).
* שם המסעדה (name).
* רמת מחיר (price level).
* כתובת (city, street, street number).
* מיקום גיאוגרפי (geo location).
* שעות פתיחה לפי ימים (activity hours).
* האם ניתן לאכול במקום (dine in).
* האם ניתן לבצע משלוחים (delivery).
* האם ניתן לבצע איסוף עצמי (takeout).
* האם ניתן להזמין מקום (reservable).
* האם מגישים בירה (serves beer).
* האם מגישים יין (serves wine).
* האם מגישים ארוחות בוקר (serves breakfast).
* האם מגישים בראנץ' (serves brunch).
* האם מגישים ארוחות צהרים (serves lunch).
* האם מגישים ארוחות ערב (serves dinner).
* האם מגישים אוכל טבעוני (serves vegetarian food).
* האם יש נגישות לנכים (wheelchair accessible entrance).
* האם ניתן לבצע איסוף בצד המדרכה (curbside pickup).
* האם פתוח בשבת (open on Saturday).
* האם יש אתר (website).

בנוסף לתכונות אלו, אנו מקבלים גם את הדירוגים של המסעדות:

* דירוג (rating).
* מספר מדרגים (user ratings total).

בעזרת שתי תכונות אלו אנו יוצרים את הדירוג המשוקלל (יתואר בהמשך) של המסעדה.

1. Google Places – ממקור זה אנו מקבלים את כל המיקומים של המסעדות והחנויות.

ממקור זה אנו מקבלים את התכונות הבאות:

* מזהה המקום לפי גוגל (place id).
* מיקום גיאוגרפי (geo location).
* סוג, מסעדה/חנות (type).

1. Rest – ממקור זה אנו מקבלים את סוג המסעדה והאם היא כשרה.

ממקור זה אנו מקבלים את התכונות הבאות:

* + מזהה המקום לפי rest (id).
  + שם המסעדה (name).
  + סוג המסעדה (type).
* האם היא כשרה (kosher).
* רמת מחיר (price level).
  + כתובת (city, address).

1. Gov – מאגרי המידע הממשלתיים – ממקור זה אנו מקבלים את כל המיקומים של תחנות האוטובוס.

ממקור זה אנו מקבלים את התכונות הבאות:

* + מזהה התחנה לפי gov (station id).
  + מיקום גיאוגרפי (geo location).

1. Cbs – הלשכה המרכזית לסטטיסטיקה – ממקור זה אנו מקבלים את המצב הסוציו-אקונומי ואת אחוז החרדים לפי רחובות.

ממקור זה אנו מקבלים את התכונות הבאות:

* + עיר (city).
  + רחוב (street).
  + אחוז החרדים (percent of religious).
  + מצב סוציו-אקונומי (socio-economic cluster, rank, value).

אופן איסוף וקבלת הנתונים:

מכל מקור מידע אספנו את הנתונים בצורה שונה, דרכי התמודדות שונים וכן קשיים שנאלצנו לפתור עבור מקרים מסוימים.

1. Google – השתמשנו ב- API של גוגל, שבו מבקשים מסעדות/חנויות לפי מיקום גיאוגרפי. הקושי שנתקלנו בו היה שלכל שאילתת API, מספר המקומות שהיינו מקבלים מהשאילתה היה מוגבל (60), לכן לא יכולנו לבקש את כל המסעדות שיש בחיפה (למשל) בשאילתה אחת. התמודדנו עם קושי זה על ידי מספר רב של בקשות. יצרנו רשת של נקודות על פני המפה ולכל נקודה ביצענו שאילתת API. המרחק בין הנקודות היה נמוך יחסית וקבוע (1 קילומטר).



1. Rest – את הנתונים אנחנו משיגים על ידי web scraping. לכל עיר אנו רושמים בשורה של כתובת ה- URL את הקישור המתאים לאתר Rest, משם אנו לוקחים את תוצאות החיפוש ומחלצים את הנתונים שאנו רוצים לאסוף.
2. Gov – את הנתונים אנחנו משיגים על ידי בקשות HTTPS ל- API של האתר Gov. לכל עיר (בעזרת הקוד של העיר שהוא המזהה ש- Gov נתנו) אנו מבצעים שאילתת HTTPS, מקבלים את הנתונים הרלוונטיים של אותה העיר ומחלצים את הנתונים שאנו רוצים לאסוף.
3. Cbs – את הנתונים אנו מורידים מהאתר של Cbs בצורת xlsx. אנו מבצעים 2 מיפויים על מנת לאסוף את הנתונים.   
   לכל אזור (שמכיל כמה רחובות) יש קוד שנקרא "סמל אזור סטטיסטי".  
   הנתונים שאנו משיגים ממקור זה מכילים סמל אזור סטטיסטי ישן והאקסל שמכיל את המעבר מסמל אזור סטטיסטי לרחובות משתמש בסמל אזור סטטיסטי חדש. לכן יש אקסל ביניים שמכיל מיפוי בין סמל אזור סטטיסטי ישן לאזור סטטיסטי חדש.   
   כלומר סה"כ יש 4 אקסלים:
   * אחוז החרדים לפי סמל אזור סטטיסטי ישן.
   * מצב סוציו-אקונומי לפי סמל אזור סטטיסטי ישן.
   * טבלת המרה מסמל אזור סטטיסטי ישן לסמל אזור סטטיסטי חדש.
   * רחובות לפי סמל אזור סטטיסטי חדש.

דוגמה:   
  
תמונה שמכילה טקסט, צילום מסך, קו, גופן

התיאור נוצר באופן אוטומטי  
  
לפי השורה לעיל ניתן לראות שבאזור הסטטיסטי 811 שבעיר חיפה יש 26.4 אחוז חרדים.



ניתן לראות, שסמל אזור סטטיסטי 811 (הישן) השתנה ל- 721 (החדש).



לסיום, הרחובות שנמצאים באזור הסטטיסטי 721 הם דורי יעקב רא"ל, שד טרומפלדור, לוחמי חרות ישראל, הגליל ומימון הרב.

לסיכום קיבלנו שברחובות הללו יש 26.4 אחוז חרדים.

**הכנת מסד הנתונים המרכזי:**

עד כה, יצרנו 5 מסדי נתונים: Google, Google Places, Rest, Gov, Cbs.

כעת, נרצה לחבר את כולם למסד נתונים אחד מרכזי.

חיבור מסד הנתונים הראשי (גוגל) עם שאר מסדי הנתונים:

1. Google Places – אנו מוסיפים לכל מסעדה את מספר החנויות ואת מספר המסעדות שיש לה באזור לפי רדיוס (100 מטרים ו- 500 מטרים). לכל מסעדה, אנו סופרים את המקומות שהמרחק שלהם מהמסעדה קטן מ- 100 מטרים ו- 500 מטרים בעזרת המיקום הגיאוגרפי.  
   אחוז התאמה: 100%
2. Rest – אנו מוסיפים לכל מסעדה האם היא כשרה ואת סוג המסעדה לפי כתובת ושם המסעדה. לכל מסעדה, אנו מחפשים לפי הכתובת של המסעדה את השורות המתאימות (במידה וקיימות) במסד נתונים הנ"ל ולפי שורות אלו אנו מחפשים את המסעדה עם התאמה לפי שם המסעדה (התאמה של השם שלה לפי גוגל והשם שלה לפי Rest) הכי גבוהה. במידה ואין שורות מתאימות לפי כתובת, אנו מחפשים התאמה לפי שם המסעדה בלבד. במידה ויש התאמה אנו מוסיפים את סוג המסעדה והאם היא כשרה.  
   אחוז התאמה של האם היא כשרה: 55%  
   אחוז התאמה של סוג המסעדה: 43.3%
3. Gov – אנו מוסיפים לכל מסעדה את מספר תחנות האוטובוס שיש לה באזור לפי רדיוס (100 מטרים ו- 500 מטרים). לכל מסעדה, אנו סופרים את תחנות האוטובוס שהמרחק שלהם מהמסעדה קטן מ- 100 מטרים ו- 500 מטרים בעזרת המיקום הגיאוגרפי.  
   אחוז התאמה: 100%
4. Cbs – אנו מוסיפים לכל מסעדה את אחוז החרדים ואת המצב הסוציו-אקונומי שיש באזור לפי כתובת. לכל מסעדה, אנו מחפשים לפי הכתובת של המסעדה את השורה המתאימה במסד הנתונים הנ"ל ולפי זה אנו מתאימים את אחוז החרדים ואת המצב הסוציו-אקונומי.  
   אחוז התאמה של אחוז החרדים: 27.7%   
   אחוז התאמה של המצב הסוציו-אקונומי: 39.6%

השלמת ערכים חסרים במסד הנתונים המרכזי:

השלמת ערכים חסרים הוא שלב חשוב בתהליך עיבוד המידע, ערכים חסרים פוגעים בשלמות הנתונים ומקשים על ניתוח התוצאות. בנוסף, ערכים חסרים עלולים לגרום להטיה של תוצאות האלגוריתם. כמו כן, השלמת ערכים חסרים משפרים את הביצועים של המודל בתהליך האימון.

כפי שניתן לראות, אחוז ההתאמה של מסדי הנתונים Cbs ו- Rest הם מועטים יחסית, נרצה להשלים את הערכים החסרים בעזרת ידע כללי של התכונות.

1. Rest –

* כידוע, מסעדה שפתוחה בשבת בהכרח לא כשרה. לכן, ניתן לדעת בעזרת התכונה, open on Saturday, האם המסעדה בהכרח לא כשרה. כלומר, אם ערך תכונה זו הוא True אז ערך התכונה Kosher יהיה False.  
  אחוז התאמה של האם היא כשרה: 74% (19%+)
* לפי שם המסעדה ניתן לדעת, בחלק מהמקרים, את סוג המסעדה. למשל מסעדות שיש להם בשם את המילה "בורגר" (כנראה) יהיו מסעדות המבורגרים. מילים נוספות ששייכנו לסוגים שונים:
  + "בורגר" – מסעדת המבורגרים
  + "פיצה" – פיצריות
  + "קפה" – בתי קפה
  + "חומוס" – חומוסייה

אחוז התאמה של סוג המסעדה: 51% (7.7%+)

1. Cbs – במסעדות שאנו לא יודעים את אחוזי החרדים/המצב הסוציו-אקונומי, השלמנו את החסר בעזרת ממוצע הערכים של המסעדות שאנו יודעים את הנתונים הנ"ל לפי מרחק (בדקנו את המסעדות במרחק של 500 מטרים).  
   אחוז התאמה של אחוז החרדים: 68.4% (40.7%+)  
   אחוז התאמה של המצב הסוציו-אקונומי: 74.4% (34.8%+)
2. Google – במסעדות שאנו לא יודעים את רמת המחיר לפי גוגל, אנו משלימים את החסר בעזרת רמת המחיר שיש לנו לפי Rest, לכן אם מצאנו התאמה של מסעדה עם Rest ויש עבורה את התכונה (רמת המחיר) אז אנו מוסיפים את הערכים הללו.  
   אחוז הערכים המלאים לפני: 36.3%  
   אחוז הערכים המלאים אחרי: 42.5% (6.2%+)

כעת, נשלים חלק מהערכים החסרים הנותרים על ידי שיטות מוכרות:

1. Google –

ערכים בוליאנים אנו נשלים עם הערך הקבוע False. אנו מניחים שערכים בוליאנים ריקים הם ככל הנראה False, אחרת בעל המסעדה (או לקוחות שמזינים את הערכים הללו בגוגל) היה משלים את זה עם הערך True. למשל עבור השאלה "האם המסעדה מגישה אוכל טבעוני?" אם המסעדה באמת הייתה מגישה אוכל טבעוני אז בעל המסעדה היה רושם True, אחרת או שהוא רושם False או שהוא ישאיר ערך זה ריק.

הערה: את הערך "פתוח בשבת" אנו יצרנו בעזרת התכונות של שעות פתיחה/סגירה של המסעדה. למעשה, חישבנו את הערך הזה בעזרת שעות הסגירה ביום שישי ושעות הפתיחה ביום שבת. במידה ואין שעות פתיחה/סגירה למסעדה, השארנו את הערך הזה ריק. לכן את הערך (הבוליאני) הזה אנו לא נשלים עם הערך הקבוע False כי אנו לא יודעים את שעות הפתיחה/סגירה (נתמודד עם זה בהמשך).

* + רמת מחיר ושעות פתיחה/סגירה אנו נשלים בעזרת ערך הממוצע של כל המסעדות.

1. Cbs – אחוז החרדים ומצב סוציו-אקונומי אנו נשלים בעזרת ערך הממוצע של כל המסעדות. לאחר השלמת הערכים לפי מרחק, כעת, את הערכים החסרים הנותרים נשלים לפי ממוצע של כל המסעדות.

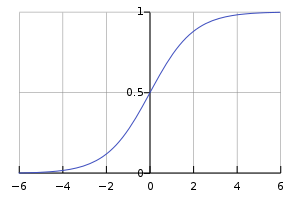
לבסוף, חלק מהתכונות עדיין מכילות ערכים ריקים ולא נשלים אותם והם "פתוח בשבת" כפי שהוסבר לעיל, "סוג המסעדה" ו- "האם כשר" בחרנו לא להשלים ערכים ריקים של ערכים אלו מכיוון שאנו לא יודעים איזה ערך הכי כדאי לשים והעדפנו לא לשים ערכים שגויים מכיוון שערכים שגויים פוגעים באיכות המודל ובתוצאות שלו. נתמודד עם ערכים ריקים בתכונות (בכללי) באלגוריתם עצמו כפי שיתואר בהמשך.

אופן חישוב הדירוג המשוקלל (הסיווג):

כאמור, יש לנו את התכונות הבאות (בעזרת גוגל):

* דירוג
* מספר מדרגים

בעזרתן אנו יוצרים את הדירוג המשוקלל של המסעדה.   
הדירוג המשוקלל יהיה מספר בין 10 ל- 100.

הדירוג נע בין 1 ל- 5.  
מספר המדרגים נע בין 1 לבין 10,000 (בערך).  
לכן רצינו להבדיל בין מסעדות עם דירוג: 5 ומספר מדרגים: 1000 (גבוה) לבין מסעדות עם דירוג: 5 ומספר מדרגים: 1 (נמוך).   
עשינו זאת על ידי נירמול מספר המדרגים וכך יצרנו ערך שמבטא את אחוז האמינות של הדירוג. פונקציית הנירמול שבחרנו היא סיגמואיד מכיוון שרצינו שערכים גבוהים יהיו קרובים לאחד וערכים נמוכים (חיוביים) יהיו קרובים לחצי (מכיוון שאנו לא יודעים האם הדירוג נכון, כי יש רק בן אדם אחד שדירג, אז 50 אחוז שהדירוג נכון ו- 50 אחוז שהוא לא נכון). נסמן:

*כאשר* n *הוא מספר המדרגים.*

*נשים לב שהערכים החיוביים הראשונים של הסיגמואיד הם:*

*קיבלנו שעבור מספר מדרגים נמוך יחסית אנחנו מגיעים כמעט לאחד.*

*תמונה שמכילה עלילה, קו, תרשים, טקסט

התיאור נוצר באופן אוטומטינפתור בעיה זו על ידי הקטנת המקדם שיש באקספוננט וכך "נשטח" את הסיגמואיד, נקבל:*

*כאשר* n *הוא מספר המדרגים.*

*נשים לב לכמה ערכים שיש לסיגמואיד החדש שיצרנו:*

*כלומר קיבלנו שעבור ערכים נמוכים אנחנו קרובים לחצי (יותר ממוקדם) ועבור ערכים גבוהים (יחסית) אנחנו קרובים לאחד. בנוסף, נשים לב, שעבור 80 מדרגים ועבור 1000 מדרגים, אחוז האמינות כמעט 1 בשני המקרים. זה מאוד הגיוני, כלומר אם יש 80 מדרגים אז כנראה שהדירוג של המסעדה אמין בדומה לדירוג של מסעדה עם 1000 מדרגים. כלומר בחרנו את המקדם* -0.05 *באופן כזה, שנקבל את ההתנהגות הזו, עבור ערכים בין 1 ל- 100 נקבל התנהגות של סיגמואיד ועבור ערכים שגבוהים מ- 100 נקבל כמעט את הערך 1.*

כעת נציג את הפונקציה שמחשבת את הדירוג המשוקלל של מסעדה:

כאשר x הוא הדירוג ו- n הוא מספר המדרגים.

נשים לב שהדירוג נע בין 1 ל- 5, לכן כאשר נכפיל ב- 20 נקבל ערך שנע בין 20 ל- 100. לאחר הכפלה בסיגמואיד נקבל ערך שנע בין 10 ל- 100.

הסרת תכונות במסד נתונים המרכזי:

אנו בחרנו להסיר תכונות עם קורלציה גבוהה. הסרת תכונות אלו יכולה למנוע Overfitting על ידי כך שנותנים דגש גבוה יותר לתכונות עם מידע רלוונטי מאשר לתכונות עם קורלציה גבוהה. בנוסף, מכיוון שאנו משתמשים בעץ החלטה, יש משמעות גבוהה להסרת תכונות מכיוון שזה יגרום לשיפור בחירת השאלות (הפיצול). במידה ויש 2 תכונות עם קורלציה גבוהה, אלגוריתם יכול לבחור את שני התכונות ולגרום לשני ענפים להתנהג בצורה דומה. כמו כן, הסרת תכונות עם קורלציה גבוהה יגרום לשיפור ביצועים ויעילות חישובית.

1. שעות פעילות – נשים לב לפי הטבלאות, שיש קורלציה גבוהה בין שעות הפעילות של ימים שני, שלישי, רביעי וחמישי, לכן נרצה לבחור רק אחד מהימים הללו (נבחר את יום רביעי, מכיוון שערך הקורלציה שלו עם כל אחד בממוצע גבוה יותר משאר הימים).

תמונה שמכילה טקסט, צילום מסך, מספר

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, מספר

התיאור נוצר באופן אוטומטי

1. יש קורלציה גבוהה (0.97) בין שתי התכונות הבאות:
   1. האם מגישים בירה (serves beer).
   2. האם מגישים יין (serves wine).

כלומר בהרבה מסעדות שמגישים בירה מגישים גם יין ולהיפך.   
לכן נבחר רק תכונה אחת (נבחר שרירותית את התכונה: האם מגישים בירה).

1. מצב סוציו-אקונומי – ניתן לראות לפי הטבלה שיש קורלציה גבוהה בין התכונות השונות שקשורות למצב הסוציו-אקונומי. לכן נבחר בתכונה: socio-economic rank מכיוון שערך הקורלציה שלו עם השאר בממוצע גבוה יותר משאר התכונות.

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

*Fun Fact: גילינו שיש קורלציה יחסית גבוהה (0.5) בין מספר האוטובוסים באזור לבין אחוז החרדים באזור.*

התכונות במסד הנתונים המרכזי:

|  |  |  |
| --- | --- | --- |
| שם | הסבר מילולי | סוג/ערכים אפשריים |
| dine\_in | האם ניתן לאכול במקום | [True, False] |
| delivery | האם ניתן לבצע משלוחים | [True, False] |
| reservable | האם ניתן להזמין מקום | [True, False] |
| serves\_beer | האם מגישים בירה | [True, False] |
| serves\_breakfast | האם מגישים ארוחות בוקר | [True, False] |
| serves\_brunch | האם מגישים בראנץ' | [True, False] |
| serves\_dinner | האם מגישים ארוחות ערב | [True, False] |
| serves\_lunch | האם מגישים ארוחות צהרים | [True, False] |
| serves\_vegetarian\_food | האם מגישים אוכל טבעוני | [True, False] |
| takeout | האם ניתן לבצע איסוף עצמי | [True, False] |
| wheelchair\_accessible\_entrance | האם יש נגישות לנכים | [True, False] |
| curbside\_pickup | האם ניתן לבצע איסוף בצד המדרכה | [True, False] |
| website | האם יש אתר | [True, False] |
| price\_level | רמת המחיר | [1, 2, 3] |
| sunday\_open | שעת פתיחה בראשון (בדקות) | [0 - 1440] |
| sunday\_close | שעת סגירה בראשון (בדקות) | [0 - 1440] |
| wednesday\_open | שעת פתיחה ברביעי (בדקות) | [0 - 1440] |
| wednesday\_close | שעת סגירה ברביעי (בדקות) | [0 - 1440] |
| friday\_open | שעת פתיחה בשישי (בדקות) | [0 - 1440] |
| friday\_close | שעת סגירה בשישי (בדקות) | [0 - 1440] |
| saturday\_open | שעת פתיחה בשבת (בדקות) | [0 - 1440] |
| saturday\_close | שעת סגירה בשבת (בדקות) | [0 - 1440] |
| open\_on\_saturday | האם פתוח בשבת | [True, False] |
| geo\_location | מיקום גיאוגרפי | (float, float) |
| percent of religious | אחוז החרדים באזור | [0 - 100] |
| socio-economic rank | מצב סוציו-אקונומי באזור | float |
| store\_100 | מספר חנויות באזור ברדיוס של 100 מטרים | int |
| store\_500 | מספר חנויות באזור ברדיוס של 500 מטרים | int |
| rest\_100 | מספר מסעדות באזור ברדיוס של 100 מטרים | int |
| rest\_500 | מספר מסעדות באזור ברדיוס של 50 מטרים | int |
| bus\_station\_100 | מספר תחנות אוטובוס באזור ברדיוס של 100 מטרים | int |
| bus\_station\_500 | מספר תחנות אוטובוס באזור ברדיוס של 500 מטרים | int |
| type | סוג המסעדה (35 אופציות) | str |
| kosher | האם היא כשרה | [True, False] |
| grade | הדירוג המשוקלל (הסיווג) של מסעדה | [0 - 100] |

**האלגוריתם:**

האלגוריתם שאיתו בחרנו לעבוד הוא עץ החלטה רגרסיה. בחרנו בעץ החלטה מכיוון שהתכונות שלנו מקיימות את הנדרש עבור שאלה בינארית ולכן קל לפצל אותן. בנוסף, קל להבין את תוצאות הפלט של האלגוריתם (העץ), אינטואיטיבי לפרש אותו. כמו כן, עץ החלטה בוחר מטבעו את התכונות החשובות ביותר בראש העץ.

אופן פעולת האלגוריתם באופן כללי:

1. אתחול העץ על ידי הנתונים (הערכים של התכונות והערכים של ציון המסעדה).
2. בוחרים את התכונה (עם ערך) שמפרידה את הנתונים על סמך שאלה בינארית, כך שנקבל את ערך ה- MSE הנמוך ביותר בכל שלב.
3. מפרידים את הנתונים לפי השאלה שבחרנו בשלב הקודם, כל תת קבוצה כזו מייצגת צומת בעץ החלטה.
4. חוזרים לשלב 2 עד שמגיעים לתנאי עצירה (עומק העץ או מינימום של ערכים או שגיאה נמוכה).
5. כאשר מגיעים לתנאי עצירה, הצומת הנ"ל נקרא עלה. בעלה יש את הערך שמוגדר להיות הציון של המסעדה (הציון נקבע לפי ממוצע הערכים שיש בתוכו).

שדרוג פעולת האלגוריתם:

כאמור, יש לנו ערכים חסרים במסד הנתונים. במהלך האלגוריתם, אנו מצפים שלא יהיה לנו ערכים ריקים מכיוון שלא ניתן לענות על שאלה עם ערך ריק. השלמנו חלק מהערכים בעזרת ידע כללי על התכונות, ממוצעים וערכים קבועים. אך עדיין יש לנו ערכים ריקים. ניסינו להתמודד עם ערכים ריקים במהלך האלגוריתם.

במהלך בניית עץ ההחלטה, כאשר בוחרים תכונה שלפיה מפרידים את הנתונים (בעזרת שאלה בינארית), אם תהיה שורה שערך התכונה שלה הוא ריק אנו נוסיף את הנתונים של אותה שורה לשני הצמתים (לצומת שעונה "כן" ולצומת שעונה "לא"). אחרת, נוסיף את הנתונים של אותה שורה לצומת הרלוונטי. לדוגמה: נניח שהתכונה/שאלה שלפיה מפרידים את הנתונים כעת היא "האם המסעדה כשרה?". עבור מסעדה שערך הכשרות שלה הוא ריק, נשים את המסעדה הנ"ל בשני הצמתים של שאלה זו.

כעת, אנו נוסיף בכל שלב משקלים המייצגים את כמות התשובות שנענו במהלך הדרך. במידה וענינו "כן" או "לא" נוסיף אחד למשקל של מסעדה זו, אחרת (הערך ריק) לא נוסיף משקל. לבסוף, כשנגיע לעלה, נחשב את ערך העלה על ידי ממוצע משוקלל בעזרת המשקלים שיש לכל מסעדה. נדגים:

נניח שנתון לנו מסד הנתונים הנ"ל:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Index\Feature** | **Kosher** | **Serves Beer** | **Delivery** | **Grade** |
| 1 | True | (blank) | True | **74** |
| 2 | True | True | True | **82** |
| 3 | (blank) | False | (blank) | **40** |
| 4 | False | True | (blank) | **95** |
| 5 | (blank) | True | False | **89** |
| 6 | True | (blank) | True | **65** |
| 7 | False | (blank) | True | **48** |

תמונה שמכילה טקסט, גופן, צילום מסך, מלבן

התיאור נוצר באופן אוטומטינניח שהעץ שנוצר הוא:

כאשר צמתים פנימיים בצבע כחול ועלים בצבע ירוק. השאלות מופיעות בשורה הראשונה והמשקל של כל מסעדה מופיע בתוך סוגריים מרובעים לצד האינדקס של המסעדה. הערכים בעלים מחושבים לפי ממוצע משוקלל בהתאם לציון המסעדה והמשקל שלה.

נסתכל על חישוב ערך העלה השמאלי ביותר: בהתחלה המשקל של כל מסעדה הוא 0 כמו שניתן לראות בראש העץ, לאחר השאלה הראשונה, "האם המסעדה כשרה?", המסעדות באינדקס 1,2,6 ענו "כן" והמסעדות באינדקס 3,5 עם ערך ריק. לכן, המשקל של מסעדות 1,2,6 עלה ב-1 והמשקל של מסעדות 3,5 נשאר 0. לאחר השאלה השנייה, "האם המסעדה מגישה בירה?", המסעדות שענו "כן" הם באינדקס 2,5 ולכן המשקלים שלהם הם 2,1 בהתאמה והמסעדות 1,6 עם ערך ריק עבור שאלה זו ולכן המשקל שלהם נשאר 1. כעת, נחשב את ערך העלה הנ"ל בהתאם למשקלים: *.*

*חישוב ערך* MSE*:*

*החישוב מבוסס על ממוצע מרחק הערכים מהממוצע בריבוע. כאמור, כאשר יש ערכים ריקים אנו מכניסים את הערך לשני הצמתים, אבל עבור חישוב ה-* MSE*, אנו מחשבים את הערכים הלא ריקים. כלומר, רק השורות שענו על השאלה הבינארית. ננסה להבין מדוע עדיף כך על ידי דוגמה. נחשב את ערכי ה-* MSE *של הבנים של הצומת השמאלי (*Serves Beer?*) מהדוגמה לעיל בשני דרכים:*

* *ללא ערכים ריקים:*

*צד שמאל מכיל את שורות 2,5 וצד ימין מכיל את שורה 3. נתחיל מצד שמאל: הממוצע של הערכים הוא 85.5. ערך ה-* MSE *הוא 12.25. צד ימין: יש רק ערך אחד לכן ערך ה-* MSE *הוא 0.   
ערך ה-* MSE *המשוקלל הוא 8.1667.*

* *עם ערכים ריקים:  
  צד שמאל מכיל את שורות 1,2,5,6 וצד ימין מכיל את שורות 1,3,6. נתחיל מצד שמאל: הממוצע של הערכים הוא 77.5. ערך ה-* MSE *הוא 88.25. צד ימין: הממוצע של הערכים הוא 59.67. ערך ה-* MSE *הוא 206.89.   
  ערך ה-* MSE *המשוקלל הוא 132.95.*

*ניתן לראות שכאשר הוספנו את הערכים הריקים, ערך ה-* MSE *קפץ מאוד ולכן יתכן ולא יבחר את השאלה הנ"ל בעקבות זאת, למרות שבלי הערכים הריקים ערך ה-* MSE *נמוך ולכן יתכן ששאלה זו תהיה השאלה הכי טובה בנקודה זו. בנוסף, כשמחשבים את ערך ה-* MSE *עם ערכים ריקים הוא עלול לעודד שאלות שמכילות הרבה ערכים ריקים (פחות אינפורמטיביות) לעומת שאלות עם פחות ערכים ריקים. לכן בחרנו בתהליך בניית העץ שלנו לחשב את ערך ה-* MSE *ללא ערכים ריקים.*

*חיזוי ציון המסעדה:*

*על מנת לחזות את ציון המסעדה, אנו מריצים את עץ ההחלטה שיצרנו עם נתוני המסעדה החדשה, לבסוף כשמגיעים לעלה אנו מחזירים את הערך שרשום בו. כמו כן, ישנם ערכים מסוימים שהמערכת מחשבת עבור המשתמש, למשל אחוז החרדים שיש באזור המסעדה שלו, כמה מסעדות יש באזור המסעדה וכו. במידה ויש ערך ריק אנו מחשבים את שני המסלולים של הצומת ולבסוף מחזירים ממוצע של שני ערכי העלים שמתקבלים.*

***תיאור המערכת:***

*המערכת שבנינו שחוזה את דירוג המסעדה של הלקוח נכתבה בשפת Python. במערכת זו מבוצעים עיקר הדברים שציינו בחלק "תיאור הפתרון המוצע לבעיה".*

*המערכת מתבססת על בניית מסדי הנתונים, הרכבת מסד נתונים מרכזי, למידה, יצירת עץ החלטה רגרסיה וחיזוי דירוג המסעדה של הלקוח.*

*בניית מסדי הנתונים:*

* Google *– אספנו את הנתונים של גוגל בעזרת הספרייה googleplaces. אספנו את הנתונים על ידי מספר שאילתות כפי שתואר בחלק הקודם.*
* Rest *– אספנו את הנתונים מהאתר rest.co.il על ידי* web scraping *בעזרת הספרייה selenium.*
* Gov *– אספנו את הנתונים מהאתר data.gov.il על ידי בקשות* HTTPS *בעזרת הספרייה urllib.*
* Cbs *– הורדנו את הנתונים מהאתר cbs.gov.il וחילצנו את הנתונים כפי שתואר בחלק הקודם.*

*הרכבת מסד נתונים מרכזי:*

*חיברנו את מסד הנתונים של גוגל עם כל השאר כפי שתואר בחלק הקודם נעזרנו בספרייה textdistance למדידת מרחק בין מילים ובספרייה mpu למדידת מרחק בין 2 נקודות גיאוגרפיות. בנוסף, השלמנו ערכים חסרים, חישבנו את הציון המשוקלל והסרנו תכונות כמו שתואר בחלק הקודם.*

*למידה:*