חיזוי של דירוג מסעדה חדשה   
לפי מסעדות קיימות

A white chef hat with a black background

Description automatically generated

**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)

**תוכן עניינים:**

שער1

תוכן עניינים2

מבוא3

תיאור הפתרון המוצע לבעיה4

איסוף הנתונים4

מקורות המידע4

אופן איסוף וקבלת הנתונים6

הכנת מסד הנתונים המרכזי8

חיבור מסד הנתונים הראשי עם שאר מסדי הנתונים8

השלמת ערכים חסרים במסד הנתונים המרכזי9

אופן חישוב הדירוג המשוקלל10

הסרת תכונות במסד נתונים המרכזי12

התכונות במסד הנתונים המרכזי13

האלגוריתם14

שדרוג פעולת האלגוריתם14

חישוב ערך MSE 16

חיזוי דירוג המסעדה16

תיאור המערכת17

שרטוט המערכת18

תהליך האימון18

תהליך החיזוי18

תיאור הניסויים19

היפר-הפרמטרים19

המדדים19

ביצוע הניסויים19

תוצאות הניסויים של דיוק תוצאות החיזוי עבור קבוצת המבחן20

תוצאות הניסויים עבור קבוצת המבחן24

תוצאות הניסויים של דיוק תוצאות החיזוי עבור קבוצת האימון25

תוצאות הניסויים עבור קבוצת האימון29

איכות האלגוריתם30

סיכום31

קשיים במהלך הפרויקט31

כיוונים להמשך המחקר 31

ביבליוגרפיה32

**מבוא:**

נכון לשנת 2020, בממוצע נסגרות בישראל 3000 מסעדות בשנה. המחזור השנתי של תעשיית המסעדות עמד על 30 מיליארד שקל (בשנת 2019), והוא מעסיק 203 אלף עובדים. לרוב, המסעדות ובתי הקפה לא "שורדים" את השנה הראשונה, פעמים רבות זה נגרם עקב חוסר התאמה של המסעדה לאזור או לאוכלוסייה. דבר זה גורם לבעלי מסעדות הפסדים כספיים רבים.

לפי הכתבה: *"תוך חצי שנה: ממסעדה לא כשרה למסעדה כשרה – ובחזרה"*, ראינו שהמסעדה "פדרו" באילת שינתה את אופי המסעדה (כשרות) שלה ולאחר חצי שנה החזירה חזרה את אופי המסעדה. "פדרו" החזירו את אופי המסעדה הקודם מכיוון שמסעדה כשרה לא התאימה לאופי האנשים באזור. המסעדה "פדרו" אופיינה במספר סוגי היין הרב שיש במסעדה והיווה חלק ניכר מהכנסות המסעדה.   
*"****אז נכון שהשקענו*** *במהלך של ההפיכה לכשרים* ***כסף****, אבל זה הביא לנו קהל שלא הייתי נחשף אליו אם לא הייתי כשר."   
"...אבל אחרי הכל, קהל אוכלי הכשר זה קהל שרובו לא ישקיע כסף בארוחות כאילו אין מחר.* ***בקהל הלא כשר יש אנשים שאוהבים לשתות ולאכול ולשתות יין יקר, ובקהל הכשר – פחות****"*.   
*"...היה לנו במתכונת הלא כשרה בערך 70-80 סוגי יין. אבל פתאום הרגשנו שגם מגוון היין ירד וגם* ***הקהל שמגיע לא ממש מתעניין בזה****."*

כאמור בכתבה המסעדה לא נסגרה בסופו של דבר, אך איבדה כסף רב, משאבים ואולי גם אנשים שאיבדו עניין במסעדה עקב החלטה שגויה.

כתבה נוספת: *"שגב משנה קונספט: ממסעדה כשרה למסעדה לא כשרה".  
"רק שנה מאז שפתח את המסעדה הכשרה הראשונה שלו בבאר שבע, משנה השף שגב משה את הקונספט והופך אותה למסעדה לא כשרה."  
"המתחם בקושי עובד באמצע השבוע, אנחנו עסק כלכלי ואין שום היגיון להפעיל שם מסעדה כשרה"  
"גם עם זה שאנחנו אמנים אנחנו אנשי עסקים, ואנחנו מבינים* ***שההחלטה הייתה שגויה****."  
"העסק צריך להתקיים בזכות עצמו במובן הכלכלי, ואתה כמסעדן* ***צריך להתאים את עצמך לשטח****."*

נפתור בעיה זו על ידי חיזוי דירוג המסעדה לפני פתיחתה או בזמן קיומה של המסעדה, על פי מאפיינים שידועים מראש. 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 רגרסיה. הרעיון להשתמש בעץ החלטה הוא שיש המון שאלות שניתן לשאול על מנת למצוא מכנה משותף בין מסעדות מצליחות למסעדות שאינן מצליחות. על ידי שאלות אלו ניתן לבנות עץ החלטה וכך נוכל לסווג בצורה מדויקת את המסעדות לפי המאפיינים שלהם. היתרון של עץ החלטה על פני מסווגים אחרים הוא שהמסווג ניתן לפירוש על ידי בני אדם, ניתן לראות את החלטות האלגוריתם בכל שלב ולהסיק את מידת החשיבות של כל תכונה. בנוסף, בעץ החלטה ניתן להשתמש בדוגמאות עם תכונות חסרות.

**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 לא מגוון אז המסווג שיצא לנו יהיה לא מוצלח.

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

מקורות המידע:

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 (הסיווג) של המסעדות:

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

בעזרת שני תיוגים אלו אנו יוצרים את הדירוג המשוקלל (יתואר בהמשך) של המסעדה.

1. Google Places – ממקור זה אנו מקבלים את 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 המיקומים של כל 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 שונות 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 שקיבלנו 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 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 מספר רחובות) יש קוד שנקרא "סמל אזור סטטיסטי".  
   במהלך השנים, שונו סמלי האזור הסטטיסטיים, לכן יש צורך לבצע המרה ביניהם. הנתונים שאנו משיגים ממקור זה מכילים את סמל האזור הסטטיסטי הישן והאקסל שמכיל את המעבר מסמל אזור סטטיסטי לרחובות משתמש בסמל האזור הסטטיסטי החדש. השתמשנו באקסל 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 סגנון המסעדה לפי הכתובת והשם שלה. עבור כל מסעדה במסד הנתונים של גוגל, אנו בודקים את כתובתה. נחפש את המסעדות עם כתובת דומה במסד הנתונים של Rest, ולאחר מכן נבדוק אם יש מסעדה עם שם זהה. במידה ולא מצאנו מסעדות עם כתובת דומה, נבצע חיפוש רק לפי השם.   
   במידה ויש התאמה אנו מוסיפים את סוג המסעדה והאם היא כשרה.  
   ביצענו התאמה בדרך זו מכיוון ששמנו לב שישנם שמות מעט שונים עבור אותה מסעדה בין מסדי הנתונים.  
   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 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 משפרת 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 על 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 עושה משלוחים?" אם המסעדה אכן הייתה עושה משלוחים, ככל הנראה 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 הערך הממוצע של כל המסעדות.

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

הערה: חלק מהתכונות עדיין מכילות ערכים ריקים ולא נשלים אותן, והן "פתוח בשבת" כפי שהוסבר לעיל, "סוג המסעדה" ו- "האם כשר". בחרנו לא להשלים ערכים ריקים של תכונות אלו מכיוון שאנו לא יודעים מהו הערך האופטימלי והעדפנו לא למלא ערכים שגויים מכיוון שאלו פוגעים באיכות המודל ובתוצאותיו. נתמודד עם ערכים ריקים באלגוריתם עצמו כפי שיתואר בהמשך.

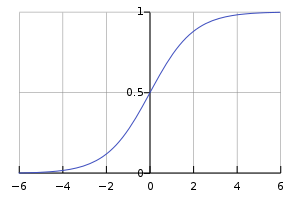
אופן חישוב הדירוג המשוקלל (הסיווג):

כאמור, יש לנו את התכונות הבאות (ממסד הנתונים של 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 (עבור הנתונים שאספנו).  
רצינו לשקלל את שתי התכונות כדי להבדיל, למשל, בין מסעדות עם דירוג 5 ומספר מדרגים גבוה לבין מסעדות עם דירוג 5 ומספר מדרגים נמוך, דבר שמעיד על אמינות הציון ופופולאריות המסעדה.  
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 אמין, כי יש מעט דירוגים, אז 50 אחוז שהדירוג מייצג את המסעדה ו- 50 אחוז שהוא לא).

נסמן:

*כאשר* n *הוא מספר המדרגים.*

*נשים לב שהערכים החיוביים הראשונים של הסיגמואיד הם:*

*קיבלנו שעבור מספר מדרגים נמוך יחסית אנחנו מגיעים כמעט לאחד (שמייצג 100% אמינות).*

*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 מהפונקציה הקודמת) 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, כך שבמקום לתת יותר דגש לתכונות עם קורלציה גבוהה, תהיה יותר התחשבות בתכונות עם מידע רלוונטי יותר. בנוסף, מכיוון שאנו משתמשים בעץ החלטה, יש משמעות גבוהה להסרת תכונות מכיוון שזה יגרום לשיפור בחירת השאלות (הפיצול). כמו כן, הסרת תכונות עם קורלציה גבוהה תגרום 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 | [10 - 100] |

**האלגוריתם:**

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

שדרוג פעולת האלגוריתם:

כפי שראינו בשלב בניית מסד הנתונים, קיימים ערכים חסרים בתכונות מסוימות. במהלך האלגוריתם, אנו מצפים שלא יהיו 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. במקום איבוד דוגמאות של מסעדות שחסרים להן ערכים בתכונות מסוימות, ניסינו להתמודד עם ערכים ריקים תוך ריצת האלגוריתם.

במהלך בניית עץ ההחלטה, כאשר בוחרים תכונה שלפיה מפרידים את הנתונים (בעזרת שאלה בינארית), אם תהיה דוגמה שערך התכונה שלה הוא ריק אנו נוסיף את הנתונים של אותה דוגמה לשני הצמתים (לצומת שעונה "כן" ולצומת שעונה "לא"). אחרת, נוסיף את הנתונים של אותה דוגמה 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 שנובעים מצומת שאלה זה.

כעת, אנו נוסיף בכל שלב משקלים המייצגים את כמות השאלות שנענו לכל דוגמה במהלך ריצת העץ. 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 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 לדירוג 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, לכן 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, כשיש ערכים ריקים אנו מכניסים את הדוגמה לשני הצמתים הבנים, אך בעת ביצוע חישוב ה-* MSE*, אנו מכניסים לחישוב את הערכים הלא ריקים בלבד. כלומר, רק הדוגמאות שענו על השאלה הבינארית. ננסה להבין מדוע דרך זו עדיפה על ידי דוגמה. נחשב את ערכי ה-* MSE *של הבנים של הצומת השמאלי (*Serves Beer?*) מהדוגמה לעיל בשתי 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 שהשאלה 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 דירוג המסעדה:*

*ראשית, אנו מוסיפים תכונות על סמך הנתונים שהמשתמש הזין, כגון אחוז החרדים שיש באזור המסעדה שלו, כמה מסעדות יש באזור המסעדה וכו'. לאחר מכן, על מנת לחזות את דירוג המסעדה, אנו מריצים את עץ ההחלטה שיצרנו עם נתוני המסעדה החדשה. לבסוף כשמגיעים לעלה אנו מחזירים את הערך שרשום בו. במידה ויש ערך ריק בתכונה שהעץ משתמש בה בריצת האלגוריתם, אנו מחשבים את שני המסלולים של הצומת ולבסוף מחזירים ממוצע של שני ערכי העלים שמתקבלים.*

***תיאור המערכת:***

*המערכת שבנינו 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 וחיזוי דירוג המסעדה של הלקוח.*

*בניית מסדי הנתונים:*

* Google *– אספנו את הנתונים של גוגל בעזרת הספרייה googleplaces, 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 כל מסדי הנתונים עם מסד הנתונים של גוגל 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 הדירוג המשוקלל והסרנו תכונות כפי שתואר בחלק הקודם.*

*למידה:*

*בשלב זה למדנו את ההיפר-פרמטרים (יתואר בהמשך) הנותנים את תוצאות הדיוק הטובות ביותר. לאחר מכן, השתמשנו בהיפר-פרמטרים שמצאנו ביצירת המסווג (עץ רגרסיה).*

*יצירת עץ רגרסיה:*

*יצרנו אלגוריתם הבונה את עץ ההחלטה בהתאם למה שכתוב בחלק הקודם (שימוש במשקלים עבור תכונות עם ערכים חסרים). כמו כן, השתמשנו בחישוב* MSE *כפי שתואר.*

*חיזוי דירוג המסעדה של הלקוח:*

*יצרנו אלגוריתם המריץ את הנתונים של הלקוח על עץ ההחלטה שיצרנו, לבסוף מחזיר את הדירוג של המסעדה (כפי שתואר בחלק הקודם).*

*אפשרויות נוספות:*

* *כאשר בעל המסעדה ממלא את נתוני המסעדה שלו, הוא יכול לבחור באופציה מיוחדת של "מצא את סוג המסעדה המתאים ביותר". במצב זה, האפליקציה מריצה את עץ ההחלטה מספר פעמים, בכל פעם עם אפשרות אחרת לסוג המסעדה. לבסוף, מחזירה ללקוח את סוג המסעדה עם הציון הטוב ביותר (בהתאם לשאר הנתונים של הלקוח).*
* *כאשר בעל המסעדה מקבל את הציון של המסעדה שלו, הוא מקבל בנוסף מידע סביב אזור המסעדה שלו. למשל, מצב סוציו-אקונומי, אחוז הדתיים, מספר תחנות אוטובוס ועוד. כמו כן, האפליקציה מציגה לבעל המסעדה את מיקום דירוג המסעדה שלו ביחס למסעדות באזור וכן כמה מסעדות, עם סוג כמו שלו, יש באזור.*

***שרטוט המערכת:***

*תהליך האימון:*

השלמת ערכים חסרים ומחיקת תכונות מיותרות

יצירת עץ החלטה רגרסיבי

איחוד כל המידע למסד נתונים מרכזי

למידת היפר-פרמטרים עבור המודל

איסוף נתוני Cbs

איסוף נתוני Gov

איסוף נתוני Rest

איסוף נתוני Google

תהליך החיזוי:

הכנסת נתונים של מסעדה חדשה

הצגת ציון המסעדה (סיווג)

הרצת עץ החלטה רגרסיבי

השלמת נתונים של הלקוח

**תיאור הניסויים:**

בשלב זה ביצענו ניסויים וכך למדנו את היפר-הפרמטרים הטובים ביותר עבור עץ ההחלטה שלנו. הניסויים שעשינו בוצעו על ידי *5-fold cross validation*. בעזרת שיטה זו נוכל לקבוע את היפר הפרמטרים הטובים ביותר עבור המודל וכן להעריך את הביצועים שלו.

היפר-הפרמטרים:

1. min for pruning – המספר המינימלי של דוגמאות הנדרש לפיצול צומת פנימי. אם נגיע לצומת שיש בו מספר דוגמאות שקטן ממספר זה, נהפוך צומת זה לעלה בעץ.  
   ערכים: [3, 10, 25, 50, 65].
2. max depth – העומק המירבי של העץ.  
   ערכים: [3, 5, 8, 10, 12, 15].
3. min samples leaf – האחוז המינימלי של דוגמאות הנדרש כדי שצומת יהיה עלה בעץ. כלומר, תהיה נקודת פיצול בצומת מסוים רק אם בכל צד בפיצול (ימין ושמאל) יהיו לפחות האחוז הנ"ל ממספר הדוגמאות בצומת.  
   ערכים: [0.1, 0.15, 0.2, 0.25].

המדדים:

בדקנו את אחוז הדיוק - accuracy (בעזרת MAE) של תוצאות החיזוי עבור קבוצת המבחן ועבור קבוצת האימון. בעזרת מדד זה נוכל לדעת את טיב האלגוריתם (כלומר כמה עץ ההחלטה צודק עם היפר-הפרמטרים שנתונים לו).   
בעזרת דיוק האימון נוכל להעריך את טיב הלמידה שהאלגוריתם ביצע על מסד הנתונים. בעזרת דיוק המבחן נוכל להעריך את ביצועי האלגוריתם עבור דוגמאות חדשות, דיוק מבחן גבוה מעיד על הכללה טובה. אם נקבל דיוק אימון גבוה מאוד לעומת דיוק המבחן, נדע שהאלגוריתם מבצע overfitting (או underfitting להיפך). לעומת זאת, אם שניהם גבוהים נדע שההכללה טובה. במידה ונקבל ערכים נמוכים של אחוז דיוק, נדע שהיפר-פרמטרים אלו לא טובים עבור המודל.

ביצוע הניסויים:

מכיוון שיש 3 היפר-פרמטרים חילקנו את הגרפים בצורה הבאה: לכל ערך שלmin samples leaf יצרנו גרף בפני עצמו, כאשר לכל גרף יש עקומה לכל ערך של min for pruning, לבסוף ערכי ציר x הם הערכים של max depth וערכי ציר y הם הערכים של המדדים (דיוקים או MSE).

צבעי העקומות של הערכים של min for pruning בכל הגרפים הם:

תמונה שמכילה גופן, צילום מסך, טקסט, מספר

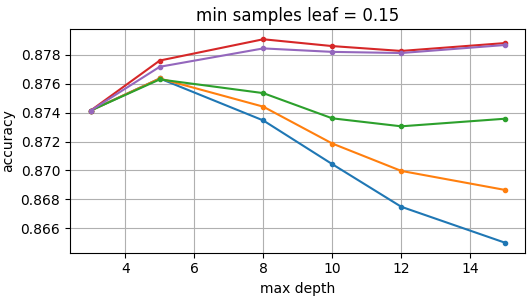
התיאור נוצר באופן אוטומטי

תוצאות הניסויים של דיוק תוצאות החיזוי עבור קבוצת המבחן:

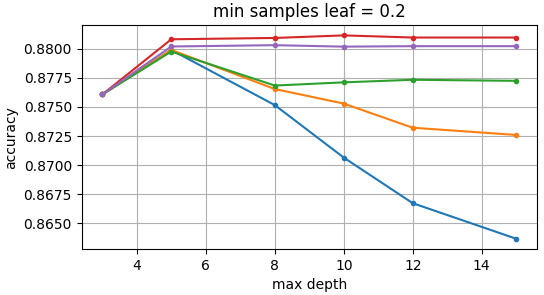
תמונה שמכילה קו, עלילה, תרשים, טקסט

התיאור נוצר באופן אוטומטי

|  |  |
| --- | --- |
| **min for pruning = 3** | **min for pruning = 10** |
| |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.874325** | | 5 | **0.874693** | | 8 | **0.870867** | | 10 | **0.866854** | | 12 | **0.864333** | | 15 | **0.861346** | | |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.874325** | | 5 | **0.874699** | | 8 | **0.87222** | | 10 | **0.86893** | | 12 | **0.867997** | | 15 | **0.867672** | |
| **min for pruning = 25** | **min for pruning = 50** |
| |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.874325** | | 5 | **0.874645** | | 8 | **0.872707** | | 10 | **0.87102** | | 12 | **0.871925** | | 15 | **0.873219** | | |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.874325** | | 5 | **0.875453** | | 8 | **0.874974** | | 10 | **0.874256** | | 12 | **0.87507** | | 15 | **0.876249** | |
| **min for pruning = 65** | **היפר-פרמטרים הטובים ביותר** |
| |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.874325** | | 5 | **0.87529** | | 8 | **0.874913** | | 10 | **0.87474** | | 12 | **0.875587** | | 15 | **0.876658** | | **Min samples leaf = 0.1**  **Min for pruning = 65**  **Max depth = 15**  **Valid accuracy = 0.8766** |



|  |  |
| --- | --- |
| **min for pruning = 3** | **min for pruning = 10** |
| |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.874144** | | 5 | **0.876359** | | 8 | **0.873457** | | 10 | **0.870435** | | 12 | **0.867487** | | 15 | **0.865007** | | |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.874144** | | 5 | **0.876359** | | 8 | **0.87441** | | 10 | **0.871862** | | 12 | **0.869967** | | 15 | **0.868654** | |
| **min for pruning = 25** | **min for pruning = 50** |
| |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.874144** | | 5 | **0.876295** | | 8 | **0.875347** | | 10 | **0.873604** | | 12 | **0.873051** | | 15 | **0.873574** | | |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.874144** | | 5 | **0.8776** | | 8 | **0.879072** | | 10 | **0.8786** | | 12 | **0.878264** | | 15 | **0.878806** | |
| **min for pruning = 65** | **היפר-פרמטרים הטובים ביותר** |
| |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.874144** | | 5 | **0.877169** | | 8 | **0.878439** | | 10 | **0.8782** | | 12 | **0.87812** | | 15 | **0.878673** | | **Min samples leaf = 0.15**  **Min for pruning = 50**  **Max depth = 8**  **Valid accuracy = 0.879** |



|  |  |
| --- | --- |
| **min for pruning = 3** | **min for pruning = 10** |
| |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.876066** | | 5 | **0.879884** | | 8 | **0.875157** | | 10 | **0.870652** | | 12 | **0.86674** | | 15 | **0.863692** | | |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.876066** | | 5 | **0.879884** | | 8 | **0.876534** | | 10 | **0.875284** | | 12 | **0.873213** | | 15 | **0.872593** | |
| **min for pruning = 25** | **min for pruning = 50** |
| |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.876066** | | 5 | **0.879748** | | 8 | **0.876831** | | 10 | **0.877105** | | 12 | **0.877327** | | 15 | **0.877235** | | |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.876066** | | 5 | **0.880799** | | 8 | **0.880914** | | 10 | **0.881132** | | 12 | **0.88095** | | 15 | **0.88095** | |
| **min for pruning = 65** | **היפר-פרמטרים הטובים ביותר** |
| |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.876066** | | 5 | **0.880181** | | 8 | **0.880294** | | 10 | **0.880168** | | 12 | **0.880209** | | 15 | **0.880209** | | **Min samples leaf = 0.2**  **Min for pruning = 50**  **Max depth = 10**  **Valid accuracy = 0.881** |

תמונה שמכילה קו, עלילה, תרשים, מדרון

התיאור נוצר באופן אוטומטי

|  |  |
| --- | --- |
| **min for pruning = 3** | **min for pruning = 10** |
| |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.879099** | | 5 | **0.879486** | | 8 | **0.869765** | | 10 | **0.864734** | | 12 | **0.861462** | | 15 | **0.85897** | | |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.879099** | | 5 | **0.879486** | | 8 | **0.871458** | | 10 | **0.869102** | | 12 | **0.869001** | | 15 | **0.868671** | |
| **min for pruning = 25** | **min for pruning = 50** |
| |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.879099** | | 5 | **0.879619** | | 8 | **0.874155** | | 10 | **0.87346** | | 12 | **0.873638** | | 15 | **0.873466** | | |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.879099** | | 5 | **0.880795** | | 8 | **0.879378** | | 10 | **0.879411** | | 12 | **0.879411** | | 15 | **0.879411** | |
| **min for pruning = 65** | **היפר-פרמטרים הטובים ביותר** |
| |  |  | | --- | --- | | max depth | valid acc | | 3 | **0.879099** | | 5 | **0.88123** | | 8 | **0.879513** | | 10 | **0.879538** | | 12 | **0.879538** | | 15 | **0.879538** | | **Min samples leaf = 0.25**  **Min for pruning = 65**  **Max depth = 5**  **Valid accuracy = 0.881** |

תוצאות הניסויים עבור קבוצת המבחן:

היפר-הפרמטרים הטובים ביותר שקיבלנו בסך הכל הם:

**Min samples leaf = 0.25**

**Min for pruning = 65**

**Max depth = 5**

**Min samples leaf = 0.2**

**Min for pruning = 50**

**Max depth = 10**

קיבלנו 2 אופציות של היפר-פרמטרים (הדיוק שלהם קרוב מאוד עד כדי מאית אחוז). אנו נבחר את האופציה עם עומק העץ הגבוה יותר על מנת לקבל עץ יותר מורכב, עם יותר עלים, יותר חיזויים. מכיוון שאנו בונים עץ רגרסיה אנו מעוניינים ביותר עלים כדי לקבל שונות גדולה יותר.

נשים לב שככל שאנו מגדילים את היפר-הפרמטר min for pruning אנו מקבלים תוצאות טובות יותר, כלומר אם גודמים ענפים בשלבים מוקדמים יותר בתהליך בניית העץ מקבלים אחוז דיוק גבוה יותר. עבור ערכים קטנים אנו מקבלים overfitting כי מנסים להגיע לעלים עם מעט דוגמאות בעוד עבור ערכים גבוהים אנו מתגברים על רעשים ומקבלים חלוקה טובה יותר. בנוסף, קיבלנו שככל שאנו מגדילים את היפר-הפרמטר max depth בשילוב עם min for pruning נמוך יחסית אנו מקבלים ירידה באחוז הדיוק. כמו קודם, יש נטייה לקבל overfitting כאשר עומק העץ גבוה ועלים עם מספר דוגמאות קטן יחסית.

תוצאות הניסויים של דיוק תוצאות החיזוי עבור קבוצת האימון:

תמונה שמכילה קו, עלילה, תרשים, טקסט

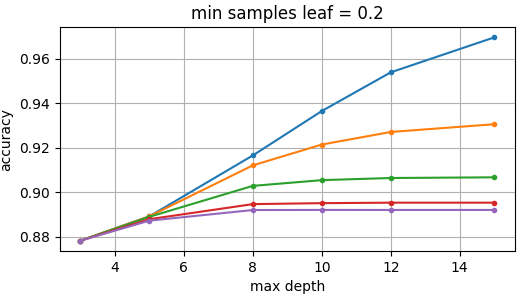
התיאור נוצר באופן אוטומטי

|  |  |
| --- | --- |
| **min for pruning = 3** | **min for pruning = 10** |
| |  |  | | --- | --- | | max depth | train acc | | 3 | **0.876367** | | 5 | **0.883193** | | 8 | **0.899283** | | 10 | **0.911891** | | 12 | **0.925102** | | 15 | **0.942817** | | |  |  | | --- | --- | | max depth | train acc | | 3 | **0.876367** | | 5 | **0.883186** | | 8 | **0.896452** | | 10 | **0.903985** | | 12 | **0.911477** | | 15 | **0.920625** | |
| **min for pruning = 25** | **min for pruning = 50** |
| |  |  | | --- | --- | | max depth | train acc | | 3 | **0.876367** | | 5 | **0.883122** | | 8 | **0.891478** | | 10 | **0.895138** | | 12 | **0.898381** | | 15 | **0.902709** | | |  |  | | --- | --- | | max depth | train acc | | 3 | **0.876367** | | 5 | **0.881839** | | 8 | **0.886616** | | 10 | **0.888938** | | 12 | **0.890632** | | 15 | **0.89341** | |
| **min for pruning = 65** | **היפר-פרמטרים הטובים ביותר** |
| |  |  | | --- | --- | | max depth | train acc | | 3 | **0.876367** | | 5 | **0.881099** | | 8 | **0.884444** | | 10 | **0.886305** | | 12 | **0.887613** | | 15 | **0.890035** | | **Min samples leaf = 0.1**  **Min for pruning = 3**  **Max depth = 15**  **Train accuracy = 0.942** |

תמונה שמכילה קו, עלילה, תרשים, טקסט

התיאור נוצר באופן אוטומטי

|  |  |
| --- | --- |
| **min for pruning = 3** | **min for pruning = 10** |
| |  |  | | --- | --- | | max depth | train acc | | 3 | **0.875822** | | 5 | **0.886496** | | 8 | **0.908498** | | 10 | **0.924497** | | 12 | **0.939363** | | 15 | **0.957036** | | |  |  | | --- | --- | | max depth | train acc | | 3 | **0.875822** | | 5 | **0.886496** | | 8 | **0.905378** | | 10 | **0.914094** | | 12 | **0.920287** | | 15 | **0.927385** | |
| **min for pruning = 25** | **min for pruning = 50** |
| |  |  | | --- | --- | | max depth | train acc | | 3 | **0.875822** | | 5 | **0.886366** | | 8 | **0.898371** | | 10 | **0.90215** | | 12 | **0.903708** | | 15 | **0.906039** | | |  |  | | --- | --- | | max depth | train acc | | 3 | **0.875822** | | 5 | **0.884977** | | 8 | **0.892011** | | 10 | **0.893332** | | 12 | **0.893682** | | 15 | **0.89445** | |
| **min for pruning = 65** | **היפר-פרמטרים הטובים ביותר** |
| |  |  | | --- | --- | | max depth | train acc | | 3 | **0.875822** | | 5 | **0.884321** | | 8 | **0.889435** | | 10 | **0.890114** | | 12 | **0.890216** | | 15 | **0.890792** | | **Min samples leaf = 0.15**  **Min for pruning = 3**  **Max depth = 15**  **Train accuracy = 0.957** |



|  |  |
| --- | --- |
| **min for pruning = 3** | **min for pruning = 10** |
| |  |  | | --- | --- | | max depth | train acc | | 3 | **0.87825** | | 5 | **0.889175** | | 8 | **0.916517** | | 10 | **0.936445** | | 12 | **0.953819** | | 15 | **0.969499** | | |  |  | | --- | --- | | max depth | train acc | | 3 | **0.87825** | | 5 | **0.889175** | | 8 | **0.912072** | | 10 | **0.921411** | | 12 | **0.92706** | | 15 | **0.930532** | |
| **min for pruning = 25** | **min for pruning = 50** |
| |  |  | | --- | --- | | max depth | train acc | | 3 | **0.87825** | | 5 | **0.888977** | | 8 | **0.902868** | | 10 | **0.905434** | | 12 | **0.906402** | | 15 | **0.90671** | | |  |  | | --- | --- | | max depth | train acc | | 3 | **0.87825** | | 5 | **0.887914** | | 8 | **0.894678** | | 10 | **0.895115** | | 12 | **0.89533** | | 15 | **0.89533** | |
| **min for pruning = 65** | **היפר-פרמטרים הטובים ביותר** |
| |  |  | | --- | --- | | max depth | train acc | | 3 | **0.87825** | | 5 | **0.887265** | | 8 | **0.892025** | | 10 | **0.892078** | | 12 | **0.892065** | | 15 | **0.892065** | | **Min samples leaf = 0.2**  **Min for pruning = 3**  **Max depth = 15**  **Train accuracy = 0.969** |

תמונה שמכילה קו, עלילה, תרשים, טקסט

התיאור נוצר באופן אוטומטי

|  |  |
| --- | --- |
| **min for pruning = 3** | **min for pruning = 10** |
| |  |  | | --- | --- | | max depth | train acc | | 3 | **0.88045** | | 5 | **0.8935** | | 8 | **0.922002** | | 10 | **0.943137** | | 12 | **0.958046** | | 15 | **0.968422** | | |  |  | | --- | --- | | max depth | train acc | | 3 | **0.88045** | | 5 | **0.8935** | | 8 | **0.915937** | | 10 | **0.923435** | | 12 | **0.926721** | | 15 | **0.928295** | |
| **min for pruning = 25** | **min for pruning = 50** |
| |  |  | | --- | --- | | max depth | train acc | | 3 | **0.88045** | | 5 | **0.893406** | | 8 | **0.90421** | | 10 | **0.905213** | | 12 | **0.905406** | | 15 | **0.905465** | | |  |  | | --- | --- | | max depth | train acc | | 3 | **0.88045** | | 5 | **0.890949** | | 8 | **0.893799** | | 10 | **0.893939** | | 12 | **0.893939** | | 15 | **0.893939** | |
| **min for pruning = 65** | **היפר-פרמטרים הטובים ביותר** |
| |  |  | | --- | --- | | max depth | train acc | | 3 | **0.88045** | | 5 | **0.890199** | | 8 | **0.891448** | | 10 | **0.891496** | | 12 | **0.891496** | | 15 | **0.891496** | | **Min samples leaf = 0.25**  **Min for pruning = 3**  **Max depth = 15**  **Train accuracy = 0.968** |

תוצאות הניסויים עבור קבוצת האימון:

היפר-הפרמטרים הטובים ביותר שקיבלנו בסך הכל הם:

**Min samples leaf = 0.2**

**Min for pruning = 3**

**Max depth = 15**

נשים לב שככל שאנו מגדילים את היפר-הפרמטר max depth אנו מקבלים תוצאות טובות יותר. מכיוון שככל שעומק העץ גבוה יותר כך אנו מקבלים יותר overfitting, ולכן כאשר אנו בודקים עבור קבוצת האימון אנו מקבלים אחוזי דיוק גבוהים יותר. בנוסף, ככל שהיפר-הפרמטר min for pruning קטן יותר כך אחוזי הדיוק גבוהים יותר. כמו קודם, אנו מקבלים פחות דוגמאות בעלים ולכן overfittingגבוה יותר.

המסקנה שלנו בסוף חלק זה היא שהמסווג עם היפר-הפרמטרים הללו:

**Min samples leaf = 0.2**

**Min for pruning = 50**

**Max depth = 10**

נותן את התוצאות הטובות ביותר (88% דיוק) מבין שאר האפשרויות ולכן אלו היפר-הפרמטרים שאיתם נעבוד בבניית עץ ההחלטה שלנו.

איכות האלגוריתם:

בגרפים הבאים שיערכנו את ביצועי האלגוריתם. בגרף הראשון ניתן לראות את ציון החיזוי לעומת הציון האמיתי של דוגמאות בקבוצת המבחן. בגרף השני ניתן לראות את ההפרש בין החיזוי לאמיתי.   
בגרפים ניתן לראות שהאלגוריתם לא נוטה לחזות "ציונים קיצוניים", כלומר מרבית הציונים הם בטווח 50-80. לכן סביר להניח שעבור מסעדה בעלת ציון גבוה/נמוך מאוד האלגוריתם יטעה.

אנו סבורים שהאלגוריתם מתנהג כך מכיוון שבמסד הנתונים שלנו אין הרבה דוגמאות עם ציונים קיצוניים ולכן (בשילוב עם העובדה של min for pruning גבוה יחסית), לא התקבלו עלים שמציינים ציונים כאלו.

A graph with red lines and green lines

Description automatically generated

A graph with red dots

Description automatically generated

**סיכום:**

מטרתנו בפרויקט זה הייתה ללמוד מסווג עץ רגרסיה אשר חוזה דירוג מסעדה חדשה. במהלך הפרויקט התמודדנו עם בניית מסדי נתונים מכמה מאגרי מידע שונים ומיזוגם למסד נתונים אחד גדול. בנוסף, בחרנו לסנן תכונות עם חשיבות נמוכה יותר ולהשלים מידע חסר על סמך מידע כללי על התכונות ובעזרת שיטות נוספות. כמו כן, למדנו את המסווג שבעזרתו אנו חוזים את הדירוג של המסעדה החדשה.

קשיים במהלך הפרויקט:

במהלך הפרויקט גילינו קשיים הן בשלבים הראשונים (בניית מסד הנתונים) והן בשלבים המתקדמים (יצירת המסווג):

* הבאת הנתונים דרך גוגל הייתה מאתגרת מכיוון שלכל שאילתה קיבלנו לכל היותר 60 מסעדות לכן היינו צריכים לפתור זאת על ידי מספר שאילתות ולאפיין זאת על ידי אזורים גיאוגרפיים.
* איחוד כל מקורות המידע למסד נתונים אחד גדול. לכל מקור מידע היה מזהה ייחודי משלו ללא קשר למזהים הייחודיים של המקורות האחרים (כאשר אפילו השם והכתובת לא בהכרח זהים במקורות שונים), לכן היינו צריכים למצוא פתרון איך לחבר את כולם למסד נתונים אחד. פתרנו זאת על ידי מיקום גיאוגרפי או על ידי שם מסעדה דומה וכתובת מקורבת.
* נתקלנו במסעדות עבורן היה מספר מדרגים נמוך, וקשה להסתמך על דירוג זה (לא נרצה להסתמך על חוויה ספציפית אלא על מכלול של דעות). רצינו לתת למסעדות אלו פחות חשיבות מאשר מסעדות עם הרבה מדרגים. פתרנו זאת על ידי חישוב ציון המסעדה עם פונקציה שמשקללת את מספר המדרגים ביחד עם הדירוג של המסעדה וכך קיבלנו סיווגים מדויקים יותר ומגוונים יותר.
* ערכים ריקים במסד הנתונים שלנו גרם לקושי בהרצת האלגוריתם (בניית העץ), השתמשנו במשקלים בזמן בניית העץ כדי להתאים את האלגוריתם לתכונות עם ערכים ריקים וכך ניתן לקבוע אילו שאלות מתאימות יותר לצמתים.

כיוונים להמשך המחקר:

* שימוש במסווגים נוספים – בפרויקט התמקדנו בעיקר במסווג של בניית עץ רגרסיה. ייתכן וקיימים מסווגים אחרים שיתנו תוצאות מדויקות יותר או בעל ביצועים גבוהים יותר מאשר המסווג שבו השתמשנו. לכן, רצוי מאוד בהמשך המחקר לבחון מסווגים אחרים ולבדוק אותם מול מסד הנתונים שלנו ולהשוות מול המסווג שהשתמשנו בו.
* שימוש באזורים גיאוגרפים נוספים – בפרויקט בחרנו לבנות את מסד הנתונים על בסיס אזור חיפה והסביבה. אבל, על ידי שינויים מינוריים ניתן להשתמש באזורים נוספים. כך נוכל לבנות מסד נתונים גדול יותר או לבנות לכל אזור את מסד הנתונים שלו וליצור לו מסווג אופטימלי לאזור שלו בלבד. בשני המקרים נשפר את יכולות הפרויקט ונקבל פרספקטיבה רחבה יותר.
* הכנסת תפריט למסד הנתונים – בפרויקט יש לנו תכונה של סוג המסעדה. אך תכונה זו כללית מידי ולא יורדת לפרטים של איזה סוגי מנות נוכל למצוא במסעדה. ניתן להשתמש במקור נתונים נוסף וכך להרחיב את מסד הנתונים שלנו לתכונות נוספות (למשל במסעדה בשרית נוכל לדעת האם מגישים קבב, פרגיות, המבורגר, סטייקים ו/או כנפיים וכו). בנוסף, נוכל לקבל היבט נוסף על מחירי המנות ולתת להם משמעות נוספת במסד הנתונים המרכזי. התמודדות נוספת במקרה זה היא חילוץ התפריט מאתרי מסעדות, כלומר יש הרבה מסעדות שהתפריט מופיע כתמונה או קובץ pdf לכן נאלץ לחלץ את התפריט ממנו (ניתן לעשות זאת למשל על ידי deep learning).
* שימוש בביקורות על מסעדה – לכל מסעדה שאנו מקבלים מגוגל אנו יכולים לחלץ את הביקורות שהשאירו לקוחות. בעזרת הביקורות הללו אנו יכולים לחלץ תכונות נוספות על המסעדה, למשל, האם מגישים מנות ללא גלוטן? האם יש כיסאות לתינוקות? רמת הניקיון? שירות העובדים? וכו'. אומנם המסעדן שישתמש באפליקציה לא יכול לדעת את התשובות לתכונות הללו, אבל הוא יכול לראות כמה תכונות אלו ישפיעו על דירוג המסעדה שלו ויוכל לשנות אותן בהתאם.

**ביבליוגרפיה:**

[תוך חצי שנה: ממסעדה לא כשרה למסעדה כשרה - ובחזרה](https://www.ynet.co.il/articles/0,7340,L-4941846,00.html) – כתבה מאתר ynet.

[שגב משנה קונספט: ממסעדה כשרה למסעדה לא כשרה](https://www.ynet.co.il/articles/0,7340,L-5433710,00.html) – כתבה מאתר ynet.

<https://www.google.com/maps> - Google Maps.