**דו"ח הכנת מידע**

**מגישים:**

**עומר עצמון – 208669457**

**ולאד קיל – 312866775**

**הכנת המידע**

בשלב הכנת המידע העברנו את המידע שלנו תחת מספר פילטרים ומשני מידע שונים. כל פילטר סידרנו ככה שמורכב משני שלבים fit and transform. בשלב ה-fit אנו מסדרים את המידע בעזרת ה-train set ובשלב ה-transform אנו מפעילים את הפילטר לפי הפרמטרים שקבענו קודם על שלושת הסטים השונים. חשוב לציין שבמהלך דוח זה נציין מספר מספרים של הגדרות שקבענו אשר כולם נבחרו לאחר ניסויים רבים וחשיבה על איך ניתן למקסם את דיוק הפיצ'רים הסופיים שנבחר.

הדבר הראשון שעשינו היה לחלק את המידע לשלושת הסטים המבוקשים: train, val and test. לאחר מכן חילקנו כל סט ל-x ול-y כאשר ה-y הכיל את ה-label(vote) וה-x הכיל את כל הפיצ'רים השונים מהם אנו אמורים להסיק את המסקנה. כלי העזר לטעינת הנתונים, חלוקת המדגם, ויזואליזציה וכו' מוגדרים בקובץ utils.

בנוסף לכך, לאחר בחינה ידנית של המדגם הנתון ובחינת המאפיינים השונים כתבנו מספר הגדרות קשות עבור מאפיינים מסויימים; אילו מהם קטגוריים, מה הם הערכים האפשריים שלהם וכו'. הגדרות אלו נכתבו בקובץ globals.

זיהינו כי המאפיינים מתחלקים בצורה הבאה:

* מאפיינים רציפים:
  + Avg\_monthly\_expense\_when\_under\_age\_21
  + Avg\_lottary\_expanses
  + Avg\_monthly\_expense\_on\_pets\_or\_plants
  + Avg\_environmental\_importance
  + Financial\_balance\_score\_(0-1)
  + %Of\_Household\_Income
  + Avg\_size\_per\_room
  + Garden\_sqr\_meter\_per\_person\_in\_residancy\_area
  + Avg\_Residancy\_Altitude
  + Yearly\_ExpensesK
  + %Time\_invested\_in\_work
  + Avg\_education\_importance
  + Avg\_Satisfaction\_with\_previous\_vote
  + Avg\_monthly\_household\_cost
  + Phone\_minutes\_10\_years
  + Avg\_government\_satisfaction
  + Weighted\_education\_rank
  + %\_satisfaction\_financial\_policy
  + Avg\_monthly\_income\_all\_years
  + Political\_interest\_Total\_Score
  + Overall\_happiness\_score
* מאפיינים בדידים:
  + Occupation\_Satisfaction
  + Yearly\_IncomeK
  + Last\_school\_grades
  + Number\_of\_differnt\_parties\_voted\_for
  + Number\_of\_valued\_Kneset\_members
  + Num\_of\_kids\_born\_last\_10\_years
* מאפיינים קטגוריים:
  + Age\_group
  + Will\_vote\_only\_large\_party
  + Most\_Important\_Issue
  + Main\_transportation
  + Occupation
* מאפיינים בוליאניים:
  + Looking\_at\_poles\_results
  + Financial\_agenda\_matters
  + Gender
  + Voting\_Time
  + Married

בהתאם לנתונים הללו הגדרנו מחלקות למאפיינים, שאיפשרו לנו לשמר נתונים סטטיסטיים והגדרות שונות לגבי המאפיינים. אלו מוגדרות בקובץ features.

הדבר השני שעשינו היה להשלים חורים במידע, זאת עשינו בעזרת KNNs אחד לכל פיצ'ר בעלי פרמטר של 20 שכנים. לכל פיצ'ר יצרנו את ה-KNN שלו לפי ה-train set ולאחר מכן הפעלנו אותו על שלושת הסטים השונים. בחרנו דווקא ב-KNN כיוון שהוא מהיר לביצוע וגם יחסית למשימה נותן ביצועים טובים. תחת ההנחה שקיים מתאם רב-מאפיינים בין נדגמים דומים, שמשתקף במידה מסויימת גם בתוצאות הMutual Information, השלמה של מידע חסר ע"י סיווג/רגרסיה משכנים קרובים עדיפה על פני השמת נתונים קבועים שמשנים את התפלגות המדגם.

לאחר שכל הערכים ממולאים יכולנו כעת לתקן ערכים לא תקינים. זאת עשינו על ידי קטימת הקצוות של הדגימות לפי boxplot. דבר זה הכניס את כל הערכים לטווח מוגבל ונתן לנו את האופציה להמשיך לשלב הבא וזהו הנרמול.

בשלב הנרמול העברנו את תחום הערכים להיות בין 1- ל- 1 או 0-1 בהתאם לנתון לפי אמות המידה של ה-train set. נשים לב כי נרמול בוצע אך ורק למאפיינים רציפים או מספריים בדידים, ולא למאפיינים קטגוריים או בוליאניים. את אלו העברנו לייצוג בוליאני במספר עמודות ע"י קידוד one-hot.

כלל המחלקות והפונקציות לעיבוד מקדים של המדגם מוגדרות בקובץ manipulators.

Mutual Information

לאחר שסיימנו לארגן את המידע התחלנו לחפש אחר הפיצ'רים אותם אנו נבחר להציג. בחרנו להשתמש בפילטר המבוסס על Mutual Information ועל wrapper המבוסס על SFS. את MI בחרנו כיוון שהוא מזהה הכי טוב את הקשרים בין המאפיינים השונים וככה יכולנו לסנן בצורה המיטבית את הפיצ'רים המיותרים. חישוב MI בוצע איטרטיבית כך שבכל שלב נבחר מאפיין אחד וכנגדו נבחנים שאר המאפיינים. את התוצאות שמרנו במטריצה (סימטרית, מתלות הדדית של המשתנים) שייצגנו כHeat map. מאחר והציון של MI אינו חסום מלמעלה, ובשלב ראשון לא ברור אילו ציוני התאמה בין שני מאפיינים מרמזים על ייתירות ואילו לא, הויזואליזציה איפשרה לנו לבחון זאת בקלות

A screenshot of a computer

Description automatically generated

Sequential Forward Search

מימשנו אלגוריתם חמדן לבחירת מאפיינים כפי שהוצג בכיתב. נראה שלSFS יתרון משמעותי על פני SBS במקרה הזה מפני אי התאמה של מאפיינים לתיוג, ויתירות גדולה של מאפיינים. הפעלנו את האלגוריתם עד להתכנסות עבור שהסתיים לאחר 12 איטרציות. בשלב הראשון ניסינו לממש אלגוריתם BDS אבל החיפוש לאחור התברר כמאוד כבד מבחינת זמן ריצה ולכן נזנח לטובת SFS טהור.

המאפיינים שנבחרו יחד עם Cross Validation accuracy score. עבור כל מאפיין מופיע score של הרצת מסווג עם המאפיינים עד אליו וכולל אותו. ריצת SFS הסתימה למעשה כאשר לא ניתן היה לשפר יותר את הציון המצטבר הזה.

בחרנו בתוצאות ההרצה באמצעות מסווג KNN מאחר וציון הואלידציה על קבוצה זו היה טוב יותר.

|  |  |
| --- | --- |
| Score | Feature |
| 0.4661377776872545 | Weighted\_education\_rank |
| 0.6673326761330604 | Overall\_happiness\_score |
| 0.8194906712663 | Avg\_size\_per\_room |
| 0.8542750319184256 | Last\_school\_grades |
| 0.8568014667560828 | Number\_of\_differnt\_parties\_voted\_for |
| 0.8620102176411223 | Avg\_monthly\_expense\_on\_pets\_or\_plants |
| 8652095436200573 | Phone\_minutes\_10\_years |
| 0.8680079755598336 | Avg\_education\_importance |
| 0.8713380337726354 | Political\_interest\_Total\_Score |
| 0.8728059476339087 | Most\_Important\_Issue |
| 0.8737408045977885 | Avg\_environmental\_importance |
| 0.8754730015041806 | Married |

Select K best

ניסינו להשתמש באלגוריתם נוסף, select\_k\_best מחבילת feature selection של sklearn. מאחר והמאפיינים בקוד שלנו מוגדרים באמצעות מחלקות שכתבנו והיחס למאפיינים קטגוריים מקודדים one-hot הוא כמאפיין יחיד קשה היה להגדיר את האלגוריתם כך שיתן K מאפיינים בהתחשב בכך. תוצאות ההרצה בכל מקרה היו נחותות משמעותית מSFS.