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 מסוימים; 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
  - (0-1)_Financial_balance_score
  - Of_Household_Income%
  - Avg_size_per_room
  - Garden_sqr_meter_per_person_in_residancy_area
  - Avg_Residancy_Altitude
  - YearlyExpensesK
  - 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
  - YearlyIncomeK
  - 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, שאפשרו 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.

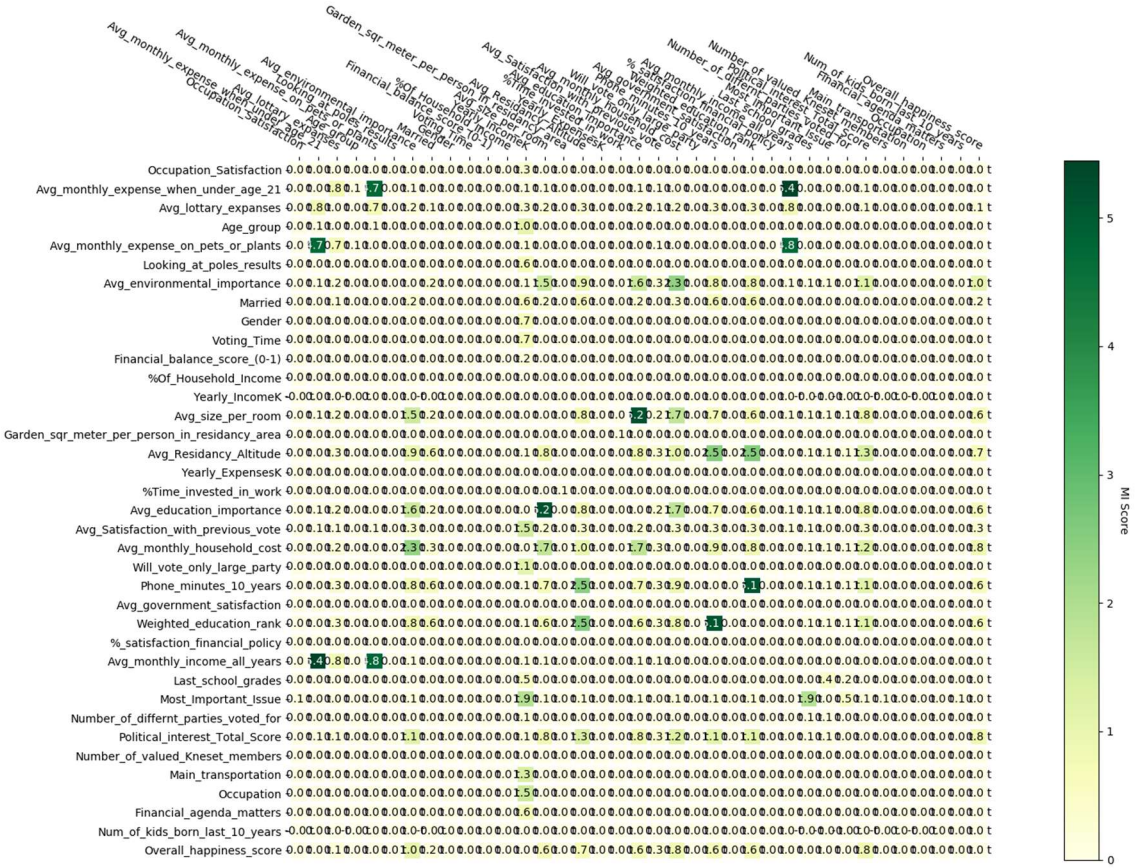
לאחר שכל הערכים ממולאים יכולנו כעת לתקן ערכים לא תקינים. זאת עשינו על ידי קטימת הקצוות של הדגימות לפי boxplot. דבר זה הכניס את כל הערכים לטווח מוגבל ונתן לנו את האופציה להמשיך לשלב הבא וזהו הנרמול.

בשלב הנרמול העברנו את תחום הערכים להיות בין 1- ל- 1 או 1-0 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 כ Heat map. מאחר והציון של MI אינו חסום מלמעלה, ובשלב ראשון לא ברור אילו ציוני התאמה בין שני מאפיינים מרמזים על יתירות ואילו לא, הוויזואליזציה אפשרה לנו לבחון זאת בקלות



Sequential Forward Search

מימשנו אלגוריתם חמדן לבחירת מאפיינים כפי שהוצג בכיתה. נראה של-SFS יתרון משמעותי על פני SBS במקרה הזה מפני אי התאמה של מאפיינים לתיוג, ויתירות גדולה של מאפיינים. הפעלנו את האלגוריתם עד להתכנסות עבור $\epsilon \leq 10^{-8}$ 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 הוולידציה על קבוצה זו היה טוב יותר.

Feature	Score
Weighted_education_rank	0.4661377776872545
Overall_happiness_score	0.6673326761330604
Avg_size_per_room	0.8194906712663
Last_school_grades	0.8542750319184256
Number_of_differnt_parties_voted_for	0.8568014667560828
Avg_monthly_expense_on_pets_or_plants	0.8620102176411223
Phone_minutes_10_years	8652095436200573
Avg_education_importance	0.8680079755598336
Political_interest_Total_Score	0.8713380337726354
Most_Important_Issue	0.8728059476339087
Avg_environmental_importance	0.8737408045977885
Married	0.8754730015041806

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 שייתן K מאפיינים בהתחשב בכך. תוצאות ההרצה בכל מקרה היו נחותות משמעותית מSFS.