**דו"ח הכנת מידע**

**מגישים:**

**עומר עצמון – 208669457**

**ולאד קיל –**

**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.

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.

לאחר שכל הערכים ממולאים יכולנו כעת לתקן ערכים לא תקינים. זאת עשינו על ידי קטימת הקצוות של הגאוסיאן שלנו להיות בתחום בין משהו למשהו. 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 לפי אמות המידה של ה-train set.

ולבסוף העברנו את כל הפיצ'רים שהם קטגוריות ל-one\_hot כנלמד.

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 הפיצ'רים השונים וככה יכולנו לסנן בצורה המיטבית את הפיצ'רים המיותרים. עשינו זאת בעזרת חישוב ה-MI עבור כל זוג פיצ'רים והתבוננות בערכים שלהם על גבי heat map וכך יכולנו לזהות את הכפילויות בקלות.

את ה-SFS מימשנו בעצמנו(לבונוס) והשתמשנו בו עם KNN ו-RandomForest כאשר שניהם עם פרמטר 20.