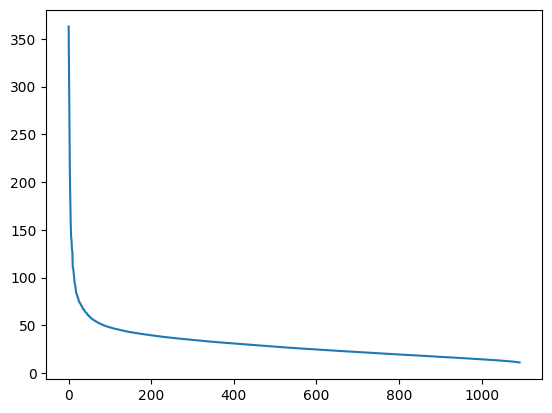
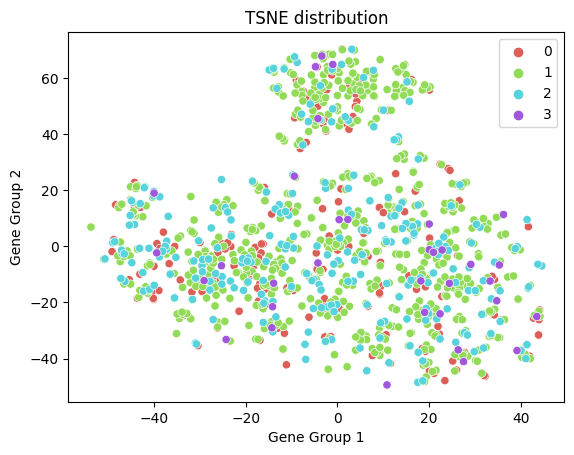
**ניקוי דאטה:**

* STD – הורדנו תכונות בעלי סטיית תקן נמוכה מאוד( > 0.05) ובכך צמצנו את מספר התכונות ל15115.
* SVD – הרצנו SVD לקבלת מספר מקסימלי של singular values(1094) ועשינו להם plot כדי לזהות מה הן התכונות החשובות ביותר. ראינו שהתכונות הכי חשובות ה100 תכונות הראשונות.



* TSNE – הרצנו TSNE בניסיון לצמצם את 17,000 התכונות ל2 ולצייר אותם בגרף כדי לראות אם ניתן לזהות בעין חלוקה למחלקות.



**תקנון דאטה**:

* Replace 0 with std – החלפה של כל 0 במטריצה עם הממוצע של התכונה שה0 אמור לייצג (מנקודת הנחה שהאפסים הם קריאות שגויות של התכונה בתהליך הזיהוי)
* Clipping by order – לקיחת 1% מהערכים הנמוכים ביותר של כל תכונה והחלפתם בערך הסף (כלומר הערך העליון ביותר של 1%).
* Clipping by value – קביעת ערך הסף כערך ה1% של כל תכונה והחלפת כל ערכים שקטנים מערך זה בכל תכונה בערך הסף.
* Stage union – בעקבות כך שאין הרבה דוגמאות לחלק מהמחלקות ביצענו איחוד מחלקות בצורה הבאה:

**Class 0 : stages 0, I, IA, IB**

**Class 1: stages II, IIA, IIB**

**Class 2: stages III, IIIA, IIIB, IIIC**

**Class 3: stages IV, X**

**[ 7. 90. 86. 6. 6. 358. 257. 2. 156. 27. 65. 20. 14.] ערכי המקור:**

**[189. 621. 250. 34.] לאחר האיחוד:**

**\* ניסיון נוסף היה איחוד הקבוצה מספר 3 עם קבוצה מספר 2 מאחר ובקבוצה 3 לא היו מספר דוגמאות רב מספיק להבנת הקבוצה בחלוקה זו**:

דבר זה שיפר את התוצאות ל 55-60% accuracy ו macro avg – 45-50%.

* Stage deletion – מחיקת כל החולים שמשתייכים לשלבים שאין בהם הרבה דוגמאות.
* Standard scaler – תקנון דאטה לפי התפלגות נורמאלית של קבוצת הTRAIN.
* Range scaler – תקנון דאטה לפי התחימה של קבוצת הTRAIN לתחום [0, 1]

**אלגוריתמי למידה**:

* Logistic regression
* LogisticRegressionCV
* Linear regression
* SVM
* Adaboost – מומש בשני דרכים – האחת – ידנית עם שימוש באלגוריתם בסיסי אחד מהאלגוריתמי ה classification הנמצאים ברשימה כבסיס, והשניה – באמצעות המודול של SKLEARN שמריץ את האלגוריתם samme שמבוסס על decision tree.
* GRIDSEARCH – ביצוע אופטימיזציה לפרמטים של האלגוריתמים הניתנים לו. (בעבודה בוצע על svc & logistic regression).

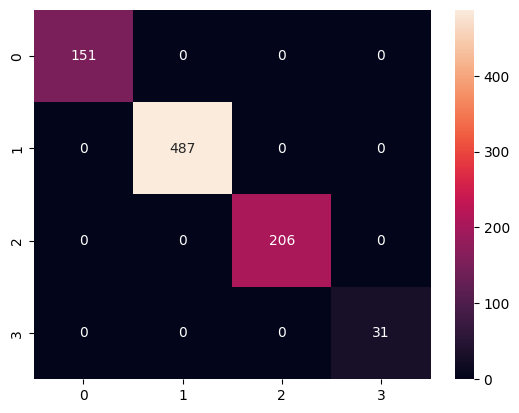
**כלי מדידה**:

* Confusion matrix
* Classification report
* Stability testing – נעשה על ידי הרצת האלגוריתם הנבחר 5 פעמים ובכל פעם מבצעים ערבוב של הדאטה וחלוקה מחדש לtrain ו test, הרצת המודל וחישוב משוקלל של הערך הממוצע, וסטיית תקן בין ההרצות של ה classification report.
* ROC curve + AUC per class – שרטוט גרף ה CURVE וחישוב ה AUC עבור כל קלאס בשיטת OVR כדי להמיר את הmultyclass שלנו למקרה בינארי שקל לשרטוט.

**תוצאות**:

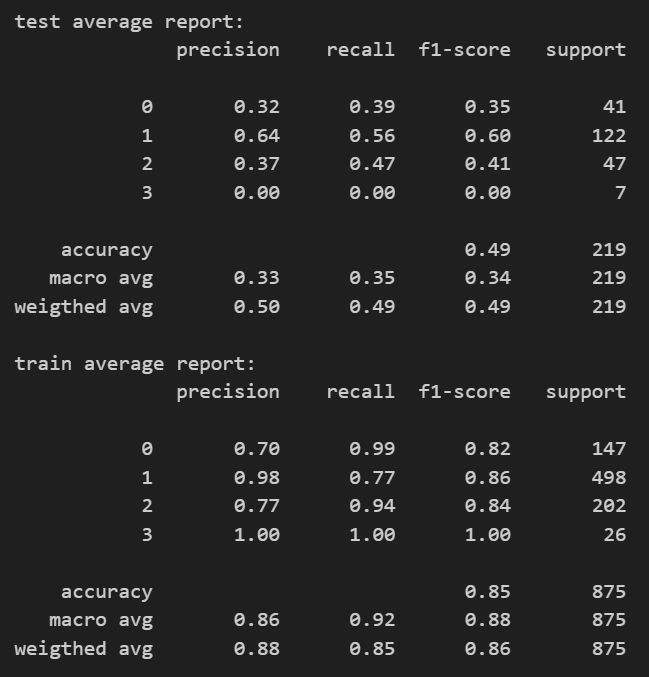
בריצות הבסיסיות accuracy לא עבר את ה 50% בSTABILTY TESTING MACROAVG F1 SCORE, לא עבר את ה 30% והיה overfitting על הtrain שקיבל 100% בכל המדדים האפשריים.

A screenshot of a computer screen

Description automatically generatedA screenshot of a grid

Description automatically generated

לכן הפעלנו את השיטות המצויינות למעלה במטרה עיקרית לצמצם את הoverfitting ולהעלות את הmacro avg.

תוצאות סופיות

A screenshot of a color chart

Description automatically generated:

A screenshot of a graph

Description automatically generated

A graph of a positive rate

Description automatically generatedA graph of a positive rate

Description automatically generated

A graph of a positive rate

Description automatically generatedA graph of a positive rate

Description automatically generated

**מסקנות**:

* נתקענו ב"חסם" עליון ללמידה על הדאטה, ככל הנראה מחוסר דוגמאות.
* השיפורים שביצענו שיפרו מעט את המודל הסופי אך בעיקר הגיעו למודל סופי מהר יותר(מודלים אחרי תקנון/ניקוי מתכנסים אחרי כמה שניות ומודלים ללא תקנון/ניקוי יכולים לרוץ גם מספר דקות)
* אפילו שלא הצלחנו לבנות מודל בעל כושר חיזוי טוב, ניתן לראות במטריצת הבלבול "מדרגות" כלומר המודלים טועים בעיקר על שלבים "קרובים" ולכן נראה שאכן יש למידה אך מחוסר דוגמאות היא לא מספיק טובה.