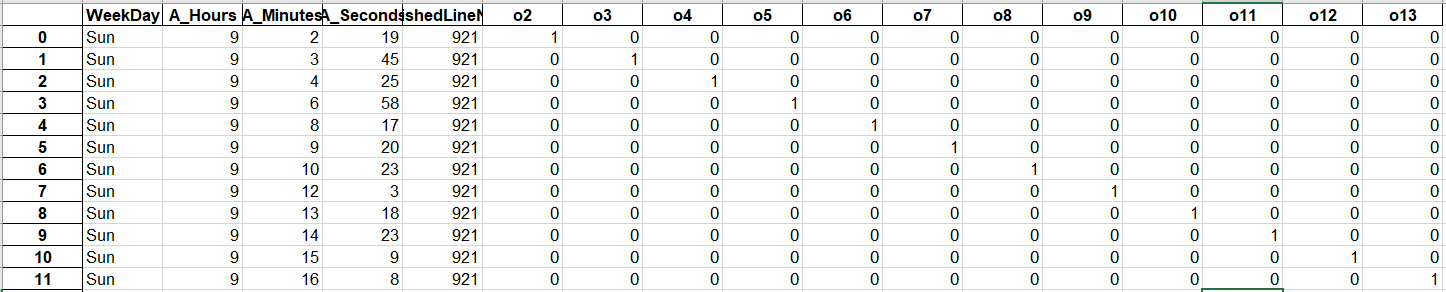
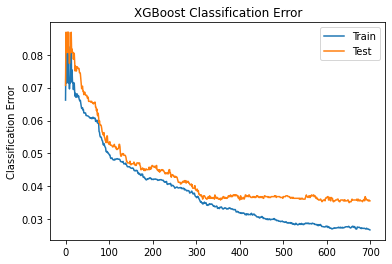
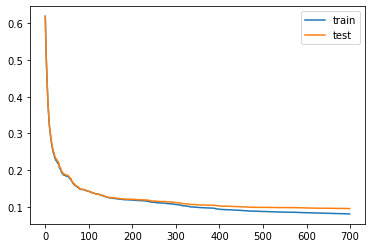
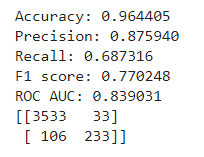
**מסמך הסבר:**

הסבר עבור קוד פייטון **XGBOOST\_Classifier.ipynb.** זה עוד מודל שהיה לי בראש והיא טיפה יותר קשה להבנה אבל בכל זאת אני אנסה להסביר איך המודל עובדת. הדבר המרכזי פה שהשתמשנו ב XGBOOST קלסיפיקציה, במלים אחרות מה אני אומר פה האם אוטובוס היה בשעה זאת בתחנה כן שהוא יקבל 1 לוגי לתחנה שכן היה ו 0 לוגי כאשר הוא לא היה בתחנה הזאת. דוגמה של טבלה מהסוג הזה:



אז בקלות אפשר לראות שיש לנו יום בשבוע, שעה מתי הוא היה בתחנה, השעה מפורקת לשעות, דקות ושניות, ומה שנאמר למלא האם הוא היה בשעה זאת מקבל 1 ושאר תחנות מקבלים 0. בסופו של דבר כל תחנה תהיה מודל והכניסות שלה הם שעות, דקות, שניות, מספר קו והיציאה היא 0 או 1 צדקנו בשעה או לא. בנינו מודל לכל תחנה והדיוק הוא ממש גבוה אבל אני חושב צריך להזהר מהדיוק כי בגדול מעניין אותנו נתון מתי אמרנו כן וצדקנו. כל תוצאות אפשר לראות בהרצת קוד פייטון. אפשר לראות שהתכנסות כל גרפים היא התכנסות טובה, באמת מודל לומדת ובאמת יש שיפור באחוזי שגיאה אפשר לראות דוגמה:



אז זה בעצם צורה של כל מודל עבור כל תחנה. אחרי זה שזה עבר כל התאליך של בניה מתחיל רגע שהוא טיפה קשה להבנה. אז איך בכל זאת אנחנו מקבלים זמן הגעה לתחנה מסוימת? בשביל זה בניתי בונקציה שהיא מקבלת מארך של כל המודלים(מודלים של תחנות), פונקציה מקבלת זמן נוכחי ואיזה תחנה(מספר תחנה) אנחנו רוצים לדעת. בתוך פונקציה יש לוגיקה שבעזרתה אני מתחיל לחפש את כל האפשרויות מתי שוטובוס בפעם הראשונה פוגש את תחנה מהשעה שהזנתי לו. במלים אחרות אני מחפש X שעבורו אני אקבל במודל ספיציפי 1 ביציאה. נקח דוגמה: אנו רוצים לדעת מתי אוטובוס תגיע לתחנה 7, השעה עכשיו 11:34,מה פונקציה תעשה, היא תתחיל לבדוק כל האפשרויות משעה 11:34 עד שתגיע בפעם הראשונה לתחנה 7 עבור שעות שאנו בודקים, ותתחיל למלא את כל האפשרויות עד שתסיים בדיקה עבור תחנה 7 במלים אחרות עד שנרא תחנה 8, ותחזיר את דאטה, ובדאטה אנו נבחר ראנדומלי שורה 1, כי רוב הדאטה הם ממש קרובים הבדל עד 3 דקות. הבעיה יחידה שדרוש הרבה דאטה, כאשר יש קצת דאטה אז קשה למודל ללמד אותה. אולי לא היתי אומר שזאת מודל הכי מושלמת אבל היתה פה חשיפה מחוץ לקופסה.

כל הקוד נבדק, ויש הסברים עבור כל קטעי קוד.