Part A: Authorship attribution: Who Controls this Account

<u>עיבוד מקדים</u>: בתהליך ה-preprocessing שלנו התמקדנו הן בהכנת הדאטה והן בהבנה שלה תוך יצירת פיצ'רים שראינו לנכון להוסיף/ליצור לשימוש במודלים שלנו. תחילה הסרנו על כל השורות עם הערכים החסרים בכדי לוודא תקינות ולהישאר עם דאטה כמה שיותר מדויקת ללא השלמות סינטטיות. להבנתנו הציוצים הישנים של טראמפ עם מכשיר האנדרואיד שלו היו עם הסבירות הכי גבוהה להיות שלו ככה שהנחנו שהמודלים ילמדו בעיקר מהם לזהות את טראמפ. לכן, משיקולים של הפשטת הבעיה החלטנו שכל שורה עם מכשיר אנדרואיד שייכת לטראמפ (0) וכל שורה אחרת היא לא (1). חילצנו פיצ'רים כמו מספר האותיות הגדולות כפי שהוצע בהנחיות העבודה, שימוש בסימני פיסוק, מספר התיוגים וכו'. בנוסף נעזרנו במודל GloVe כדי לייצג את הציוצים במרחב אמבדינג ממנו יצרנו פיצ'רים נוספים כמו ממוצע וקטורי האמבדינג לכל ציוץ וכו'. לבסוף יצרנו סט של פיצ'רים נומריים/וקטוריים שאיתם נוכל להשתמש עם סוגי המודלים השונים. כמו כן, הפקנו פיצ'רים נומריים מוקטורי האמבדינג ככה שנוכל להשתמש בהם גם במודלים שעובדים עם דאטה טבלאית.

<u>על המודלים</u>: נציג לכל אלגוריתם עם איזה דאטה הוא עבד, מה הפיצ'רים הנבחרים ולבסוף נציג ונשווה את מדד הדיוק של כל מודל בטבלה. לכל מודל ביצענו Grid Search במרחב חיפוש של כמה היפר-פרמטרים:

Algorithm	Data	Best Parameters	Accuracy
LogisticRegression	Tabular (scalar feats')	{'C': 100, 'solver': 'newton-cg'}	0.83
SVC	Tabular (scalar feats')	{'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}	0.83
FFNN	Vectors(scalars+embeds)	{'num_epochs':32, 'lr'=0.001, 'batch':32}	0.78
RandomForestClassifier	Tabular (scalar feats')	{'criterion': 'entropy', 'max_depth': 4, 'n_estimators': 30}	0.83
BERT	Text	{'epochs':10, 'lr':1e-4, 'batch_size':64, 'max_length':128}	0.92

<u>תוצאות</u>:

נשים לב כי רוב המודלים בעלי ביצועים דומים וגם לא רעים בסך הכל, מה שמצביע על אחידות הפרדיקציות ואיכות הדאטה. יתכן שביצועי FFNN מעט נמוכים יותר מכיוון שמדובר במודל מורכב יותר שדורש הרבה יותר fine tuning. המודל עם הביצועים הטובים ביותר הוא מודל ה-BERT של פרצוף-מחבק. מדובר במודל שאומן במיוחד על ציוצי טוויטר הקשורים לבחירות 2020 ועבר תהליך fine tuning באותו הקשר לטראמפ גם כן. המודל הרבה יותר עוצמתי ממה שהיינו יכולים ליצור במשאבים ובזמן שהיו נתונים לנו ולכן אנו לא מופתעים מהתוצאות. לכן מודל הBERT נבחר כמודל הטוב ביותר עם סט הפרמטרים שציינו בטבלה.

Part B: Text Classification - Vaccine Hesitancy

<u>עיבוד מקדים</u>: בתהליך ה-preprocessing שלנו התמקדנו הן בהכנת הדאטה והן בהבנה שלה תוך יצירת פיצ'רים שראינו לנכון להוסיף/ליצור לשימוש במודלים שלנו. תחילה הסרנו על כל השורות עם הערכים החסרים בכדי לוודא תקינות ולהישאר עם דאטה כמה שיותר מדויקת ללא השלמות סינטטיות. במקרה זה נעזרנו במודל alephBert (כלומר מודל של ברט שמאומן על השפה העברית) מאומן מHuggingFace כדי לייצג את הציוצים במרחב אמבדינג ממנו יצרנו פיצ'רים נוספים כמו ממוצע וקטורי האמבדינג לכל וסכום וקטורי האמבדינג.

הערה – העיבוד המקדים עם ברט צורך המון זכרון RAM ולפעמים גורם לקולאב לקרוס – התמודדנו עם זה בכך שעבדנו בצ'אנקים וכל פעם כתבנו חלקים לדיסק בתצורה של קבצי PKL שהוספנו גם לקבצי ההגשה.

<u>על המודלים</u>: נציג לכל אלגוריתם עם איזה דאטה הוא עבד, מה הפיצ'רים הנבחרים ולבסוף נציג ונשווה את מדד הדיוק של כל מודל בטבלה. לכל מודל ביצענו Grid Search במרחב חיפוש של כמה היפר-פרמטרים:

Algorithm	Data	Best Parameters	Accuracy
LogisticRegression	Tabular (scalar feats')	{'C': 0.1, 'solver': 'newton-cg'}	0.36
SVC	Tabular (scalar feats')	{'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}	0.50
FFNN	Vectors(scalars+embeds)	{'num_epochs':32, 'lr'=0.001, 'batch':32}	0.50
RandomForestClassifier	Tabular (scalar feats')	{'criterion': 'gini', 'max_depth': 1, 'n_estimators': 10}	0.50
AlephBERT	Text	{'epochs':10, 'lr':1e-4, 'batch_size':64, 'max_length':128}	0.72

<u>תוצאות</u>:

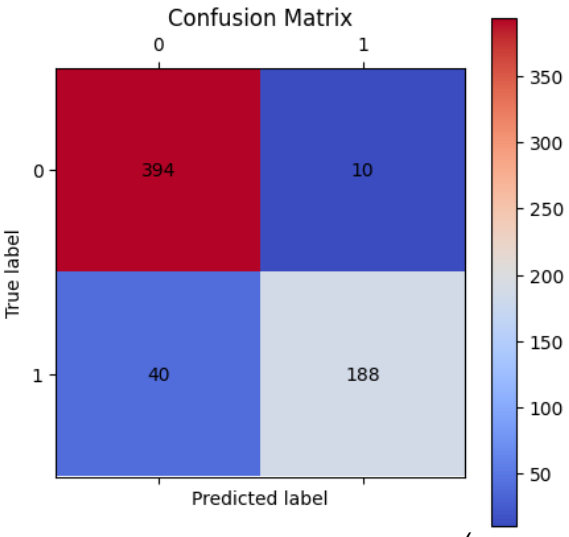
נשים לב כי מלבד ברט, כל המודלים מגיעים לתוצאות ACCURACY של לכל היותר 50%, המשימה הזאת קשה בשביל המודלים היותר פשוטים – חלקם מחזירים תמיד 0 (כלומר הם לא יותר טובים מגישה נאיבית שמחזירה לפי הmajority) המודל עם הביצועים הטובים ביותר הוא מודל ה-BERT של hugging-face שאומן על טקסט בעברית ממספר מקורות שונים כולל ויקיפדיה (אבל לא טוויטר). המודל הרבה יותר עוצמתי ממה שהיינו יכולים ליצור במשאבים ובזמן שהיו נתונים לנו ולכן אנו לא מופתעים מהתוצאות. לכן מודל הBERT נבחר כמודל הטוב ביותר עם סט הפרמטרים שציינו בטבלה.

discussing the different performance of the various algorithms on the different datasets

מסקנות ותובנות:

בקובץ המחברת IPYNB מפורטים הביצועים עבור כל מודל confusion matrix, precision, recall, accuracy, בעקבות הגבלת המקום של הדוח נתבונן בביצועים של המודלים הטובים ביותר עבור המשימות (השאר מודפסים בקובץ המחברת):

מודל ברט עבור בעיית הauthorship attribution :




```
Report:
              precision    recall  f1-score   support

           0       0.91      0.98      0.94       404
           1       0.95      0.82      0.88       228

    accuracy                           0.92       632
   macro avg       0.93      0.90      0.91       632
weighted avg       0.92      0.92      0.92       632
```

ניתן לראות כי המודל מצליח לסווג בצורה נכונה במרבית המקרים

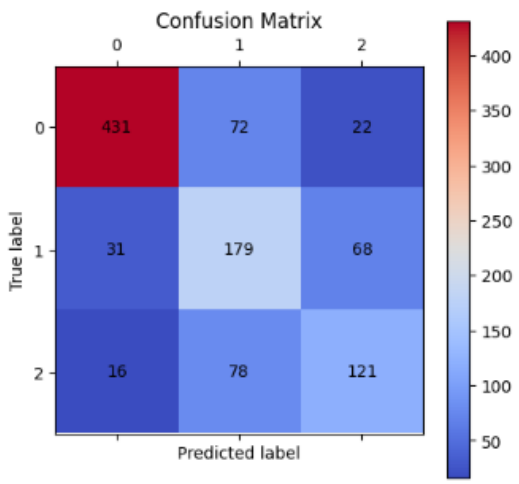
הטעות הנפוצה ביותר היא FN, זה הגיוני כי הדאטה הוא מעט

IMBALANCED – יש כפי 2 יותר מקרים בהם הקלאס 0.

עושה רושם שהמודל מצליח למדל בצורה יחסית טובה את ההתפל
הפשוטים יותר (זה טרנספורמר שאומן על דאטה של טוויטר לכן זה גם הגיוני)

המודלים האחרים גם הצליחו למדל את הדאטה ולהפריד בין המחלקות בחלק מהמקרים אבל הטרנספורמר עוקף אותם בהרבה, ומראה שכנראה באמת attention is all you need..

מודל אלף ברט עבור בעיית Text classification :




```
Report:
              precision    recall  f1-score   support

           0       0.90      0.82      0.86       525
           1       0.54      0.64      0.59       278
           2       0.57      0.56      0.57       215

    accuracy                           0.72      1018
   macro avg       0.67      0.68      0.67      1018
weighted avg       0.73      0.72      0.72      1018
```

כאן ניתן לראות כי המודל כבר הרבה יותר מתקשה בסיווג

יש ביצועים נאים עבור המחלקה הראשונה (ציוץ ניטרלי)

אך עבור 2 המחלקות האחרות הביצועים פוחתים, אם מסתכלים גם על הביצועים של שאר המודלים עבור אותה משימה ניתן לראות שהוא היחיד שמצליח להגיע להפרדה בין הקלאסים (השאר מחזירים כמעט תמיד 0).

הטעויות הנפוצות ביותר הן כאשר חוזים את המחלקה 1 והלייבל האמיתי הוא 0 או 2 (עם התפלגות דומה), גם פה יש IMBALANCED – יש כפי 2 יותר מקרים בהם הקלאס 0.

המסקנה שלנו היא שהמשימה הזאת היא הרבה יותר קשה מהמשימה הקודמת ולכן רק עם מודל מורכב הצלחנו להגיע לתוצאות שעוקפות בייסליין נאיבי (סביר שאם היה לנו עוד הרבה זמן ל hyper parameter tuning ולנסות עוד מודלים פשוטים אז היינו מגיעים לתוצאות יותר טובות מגישה נאיבית גם עם מודלים פחות מורכבים מטרנספורמר אבל בכל מקרה זה פחות straight forward ממשימת הסיווג הבינארית (הראשונה)