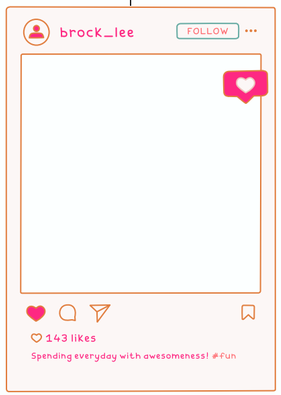
הצגת פרויקט גמר בלמידת מכונה-סיווג בין עמודי אינסטגרם פקטיביים לאמיתיים

מגישים:

נועם דוד ת"ז: 319073235
יוגב אופיר ת"ז: 322719881



תיאור תהליך

עמודים מזויפים זאת בעיה מרכזית בפלטפורמת המדיה ובפרט ברשת החברתית אינסטגרם. באמצעות למידת מכונה ניתן להבחין בין עמודים פקטיביים לעמודים אמיתיים. את המאגר שלנו בחרנו מאתר קאגל שהוצע ככלי בקורס, אתר המספק מגוון של דאטה סט לכל מיני בעיות הקשורות בניתוח נתונים. בחרנו 4 אלגוריתמים לסיווג אשר נלמדו בקורס לניתוח הבעיה.

המאגר שלנו

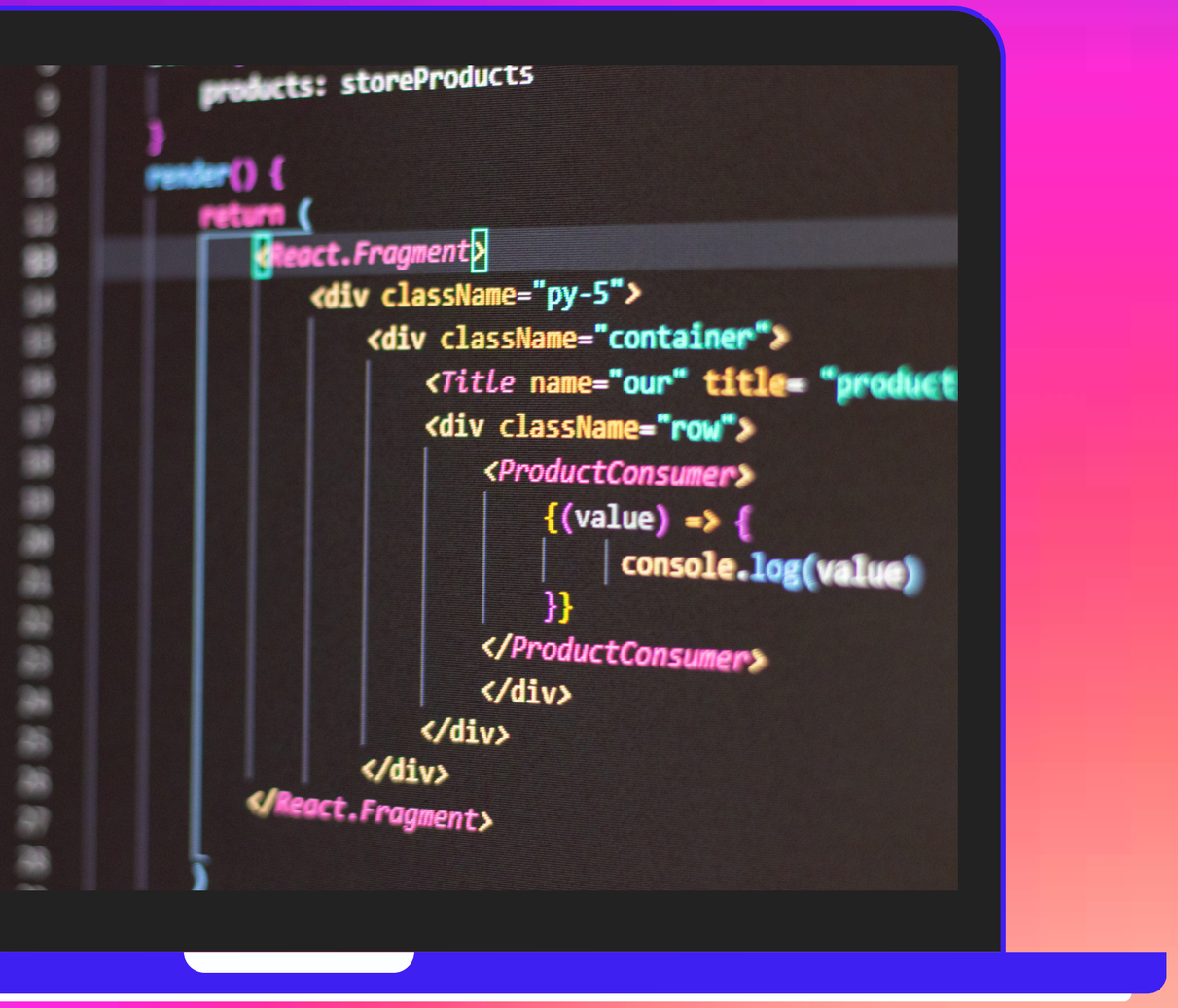
המאגר שלנו הוא 2 קבצי CSV- אחד לצורך קבוצת האימון והשני כקבוצת הבחינה.

הקבצים מכילים נתונים עבור מודל למידת מכונה. הנתונים מורכבים ממאפיינים שונים המתארים פרופילים של משתמשים ברשת החברתית INSTAGRAM כך שכל שורה מייצגת פרופיל ייחודי עם ערכי המאפיינים שלו.

קבצי CSV הם פשוטים ליצירה ולקריאה. קבצים אלה נתמכים על ידי רוב התוכנות לעיבוד נתונים ולמידת מכונה. זהו פורמט טקסטואלי פשוט המשמש לאיחסון נתונים בטבלאות כך שניתן לפרק את הנתונים לפי עמודות ושורות בקלות.

הנתונים המופיעים בכל עמודה הם:

1. PROFILE PIC האם יש תמונת פרופיל (1) או לא (0).
2. NUMS/LENGTH USERNAME יחס בין מספר התווים בשם המשתמש לאורך השם המשתמש.
3. FULLNAME WORDS מספר המילים בשם המלא.
4. NUMS/LENGTH FULLNAME יחס בין מספר התווים בשם המלא לאורך השם המלא.
5. NAME==USERNAME האם השם המשתמש תואם לשם המלא (1) או לא (0).
6. DESCRIPTION LENGTH אורך התיאור בפרופיל.
7. EXTERNAL URL האם יש קישור חיצוני (1) או לא (0).
8. PRIVATE האם הפרופיל פרטי (1) או לא (0).
9. POSTS מספר הפוסטים בפרופיל.
10. FOLLOWERS מספר העוקבים של הפרופיל.
11. FOLLOWS מספר הפרופילים שהפרופיל עוקב אחריהם ומקבל מהם עידכונים.
12. FAKE האם הפרופיל נחשב לפי הנתונים האלה לפרופיל מזויף (1) או לא (0).



האלגוריתמים בהם השתמשנו

KNN (K-NEAREST NEIGHBORS)

- אלגוריתם קרובי השכנים, הוא אלגוריתם פשוט ויעיל למשימות סיווג שמבוסס על עקרון של דמיון בין הנקודות.
- האלגוריתם לומד על ידי "זיהוי" של הנקודה הקרובה ביותר לנקודה חדשה על פי מרחק חיתוך נקודות מוגדר במרחב המאפיינים.
- בעת השימוש באלגוריתם, החישוב של הנקודה הקרובה ביותר נעשה על פי הנתונים הקיימים בקבוצת האימון.

ADABOOST (ADAPTIVE BOOSTING)

- אלגוריתם למידת מכונה שמשמש לבניית מודלים חזקים לבעיות סיווג. האלגוריתם מבוסס על העקרון של בניית מודלים חלשים ושיפורם באמצעות "חיזוק" של טעויות התחזיות.
- בכל צעד של האלגוריתם, הוא מתמקד בטווחים שבהם המודלים החלשים עדיין לא טובים ומשנה את המשקלים של הנתונים האלה כך שטעויות נפוצות יותר מסומנות עם משקל גבוה יותר, על מנת שהמודל הבא התמקד בתיקון השגיאות הללו.
- התוצאה הסופית היא דמות מורכבת של מודלים חלשים, המאורגנים לפי רמת החוזק שלהם.

SVM (SUPPORT VECTOR MACHINE)

- אלגוריתם לבעיות סיווג אשר עיקרו הוא למצוא את המסלול האופטימלי בין קבוצות הנתונים השונות ביותר. נקודת המוצא של אלגוריתם SVM היא יצירת "מישור הפרדה" המפריד בין נקודות הנתונים בצורה הטובה ביותר.

- בשלב האימון, המטרה היא למצוא את ההפרדה המקסימלית בין קבוצות הנתונים השונות, זאת בעזרת וקטורים תומכים שהם הנקודות הקריטיות ביותר שבמרחב המאפיינים. לאחר האימון, המודל נבדק ונבחן על נתוני אימות נפרדים על מנת לבדוק האם המודל יכול להתמודד בצורה יעילה עם נתונים חדשים שהוא לא נתקל בהם במהלך האימון.

- המטרה היא למצוא את הקו המירבי שיפריד בין הקבוצות באופן הטוב ביותר. זה נעשה על ידי מקסימיזצית המרחק בין הקו הזה לקבוצות הנתונים הקרובות אליו.

LOGISTIC REGRESSION

- אלגוריתם לפתרון בעיית הסיווג. אשר משתמש בפונקציה לוגיסטית לחיזוי הסבירות לקבלת שתי קטגוריות אפשריות.

- המטרה היא לחזות את הסיכוי של תוצאה ספציפית באמצעות פונקציה לוגיסטית זו. בדרך כלל, מדובר בסיווג של נתונים לשני קבוצות או קטגוריות. (פיקטיבי או אמיתי)

תוצאות השוואה בין אלגוריתמים

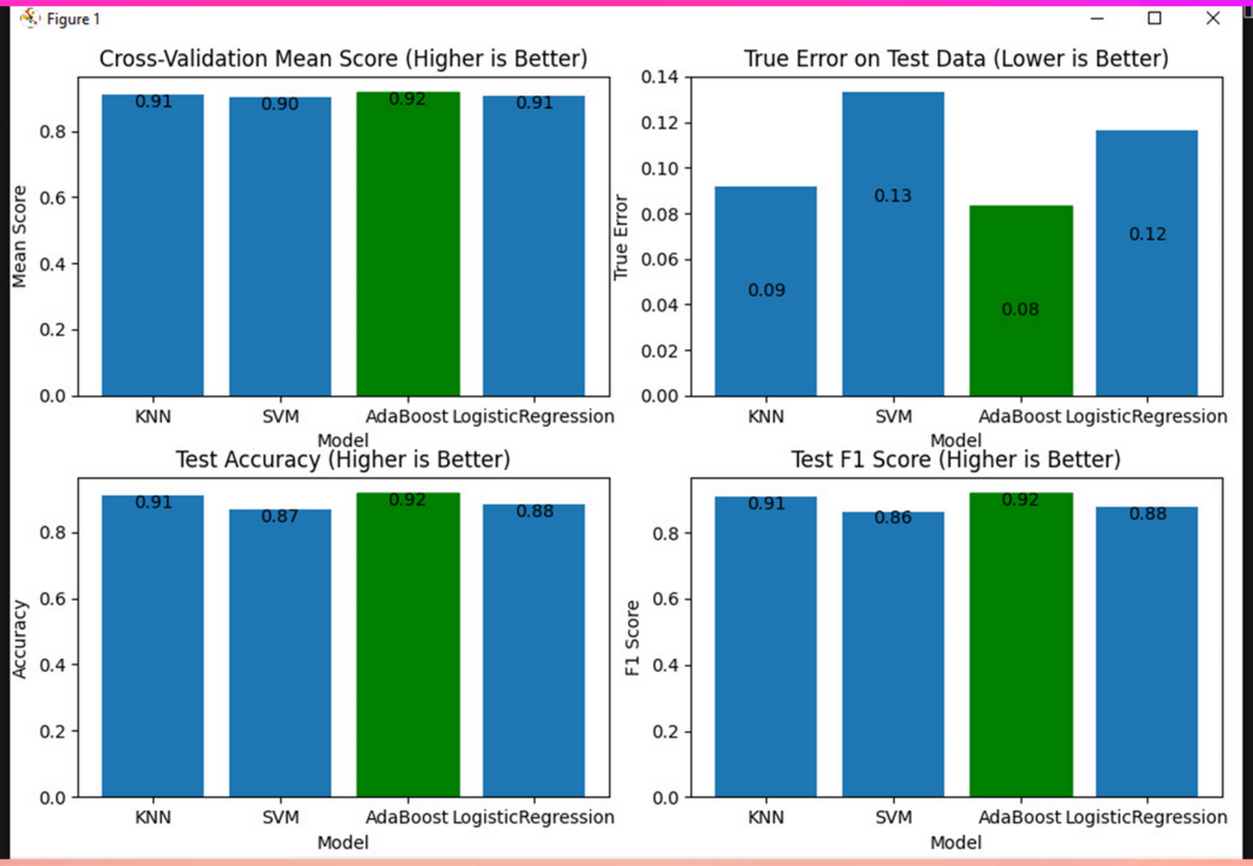
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\Yogev\Desktop\New folder> python.exe .\main.py
KNN results:
Best Parameters: {'n_neighbors': 5, 'p': 1}
Cross-Validation Mean Score: 0.91
True Error on Test Data: 0.09
Test Accuracy: 0.91
Test Precision: 0.90
Test Recall: 0.92
Test F1 Score: 0.91

SVM results:
Best Parameters: {'C': 10}
Cross-Validation Mean Score: 0.90
True Error on Test Data: 0.13
Test Accuracy: 0.87
Test Precision: 0.88
Test Recall: 0.85
Test F1 Score: 0.86

AdaBoost results:
Best Parameters: {'algorithm': 'SAMME', 'n_estimators': 100}
Cross-Validation Mean Score: 0.92
True Error on Test Data: 0.08
Test Accuracy: 0.92
Test Precision: 0.89
Test Recall: 0.95
Test F1 Score: 0.92

LogisticRegression results:
Best Parameters: {'C': 0.1, 'solver': 'liblinear'}
Cross-Validation Mean Score: 0.91
True Error on Test Data: 0.12
Test Accuracy: 0.88
Test Precision: 0.91
Test Recall: 0.85
Test F1 Score: 0.88
```



הבדלים בין תוצאות

כאשר מודל מקבל רמת דיוק גבוהה, זה אומר שהוא מסוגל לזהות ולסווג נתונים חדשים בדיוק גבוה. בניגוד, כאשר מודל קיבל קרוס ולידציה גבוהה ביותר, זה אומר שהמודל היה יעיל ביותר בזמן האימון, אבל אולי יהיה פחות יעיל ביכולתו לכלול את המידע החדש.

תוצאת F1 משקפת את יכולת המודל לזהות כמה שיותר מדויק ערכים חיוביים ושליליים, בעוד שקרוס ולידציה משקפת את הביצועים הכלליים של המודל על כל הקבוצות של הנתונים ביחד בזמן אימון.

דיוק גבוה ביותר מתמקד בכלליות של תפקוד המודל ולא באופן ספציפי ולכן יכול לפספס אי דיוק ספציפי, בעוד ש-F1 SCORE מתמקד ביכולת המודל לפעול באופן מאוזן ולהפחית את כמות השגיאות.

קישור לקוד באתר GITHUB

אתגרים וסיכום תהליך

למידת סיפריות חדשות כגון SKLEARN ו-MATPLOTLIB על מנת להיעזר בפונקציות ועל מנת ליצור תרשימים להצגת התוצאות והשוואת המודלים שבחרנו.

למידת טכניקת GRID-SEARCH אשר בוחרת מתוך קבוצת פרמטרים שהגדרנו את הפרמטר שהביא את התוצאות הטובות ביותר עבור כל מודל.
(לדוגמה: עבור מודל KNN איזה K_VALUE מתוך קבוצת הערכים הזו שהגדרנו נותן את הסיווג הטוב ביותר.)

בחירת פרמטרים לכל מודל ע"י ניסוי וטעייה. על מנת לקבל את התוצאה הטובה ביותר אבל האלגוריתם והדאטה. ביצענו מחקר על בחירת פרמטרים נפוצה עבור כל מודל וביצענו בדיקות על קבוצת פרמטרים זו אשר נתנה לנו ביצועים טובים

מהשוואת המודלים ניתן לראות שמודל הADABOOST נתן את התוצאות הטובות ביותר. ההערכה שלנו לכך היא מכיוון שהמודל מתמקד כל פעם על חלקים חלשים ומתקן אותם בזמן האימון מה שגורם לו להיות מאוזן יותר, להפחית שגיאות ובנוסף נותן CROSS VALIDATION טוב יותר.
בעקבות כל אלה המודל נותן סיווג בדיוק גבוה יותר עבור נקודת סיווג חדשה.