IML - Hackaton Description - Task 2

11 ביוני 2020

משתתפים: עילי מליניאק, גל פיבלמן, יסמין סלע, אור גרשוני.

חלק א' - יצירת דוגמיות וחלוקה לסטים

1. הסוגיה הראשונה שנתקלנו בה הייתה לייצר מקבצי הקוד הגולמיים שכתבנו דוגמיות במבנה שיתאם את הדוגמיות עליהן יבחן המודל שלנו (כל דוגמית היא $string$ המייצג שורה עד 5 שורות מתוך קובץ הקוד).
2. טיפלנו בסוגיה זו בעזרת סריקה של כל קובץ קוד (סה"כ 7 קבצים) ובחירה רנדומית של מספר בין 1 ל־5. המספר שנבחר הוא מספר השורות שנלקחו לדוגמית.
3. בסיום הריצה החזרנו $DataFrame$ המכיל 2 עמודות (בזהה למקרה המבחן אליו התכוננו) ו־ $130K$ דגימות: (1) עמודת $sample$: $string$ המייצג בין שורה ל־5 שורות מתוך קובץ הקוד. (2) עמודת $label$: סיווג בין 0 ל־6 בהתאם לקובץ ממנו נלקח מקטע הקוד.
4. את הדאטא פיצלנו לפי היחס של הדאטא המקורית ל־ 20% סט מבחן ו־ 80% סט אימון.
5. גם את סט האימון חילקנו לשניים כך שעל 40% מהדאטא המקורית ביצענו $pre-processing$, התאמת מודלים, והתאמת פיצ'רים ועל 40% ביצענו $validation$.

חלק ב' - וקטוריזצייה ו־ $preprocessing$ לקבצי טקסט

1. $\underline{pre-proccessing}$:

(א) בדקנו את התפלגויות המילים השכיחות ביותר בכל $class$, (ר' גרף מטה).

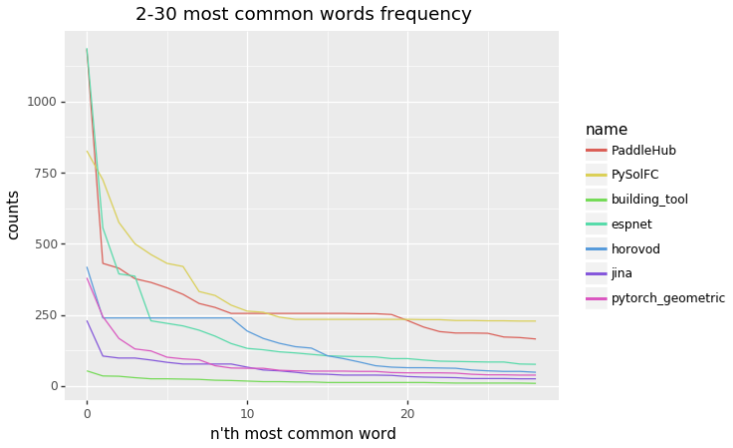
(ב) ניתן לראות כי ההתפלגויות הן אחידות בין המילה ה־10 השכיחה ביותר לביו המילה ה־20 השכיחה ביותר, ולכן עבור כל $class$ בחנו רק את עשר המילים השכיחות ביותר. מצאנו כי בנוסף לסימני הפיסוק ו־$newline$ (שהופיעו באופן תדיר בכל ה־$class$־ים), המילה $else$ נמצאה בין עשר המילים השכיחות ביותר עבור כל ה־$class$ים גם היא.

(ג) זיהינו בהמשך כי סימני הפיסוק ו־$newline$ אינם משפיעים באופן מכריע על ה־$accuracy$, אך הסרתם משפר באופן ניכר את זמן הריצה ולכן בחרנו להסירם. עם זאת, ראינו כי הורדת המילה $else$ מורידה במעט את ה־$accuracy$, ולבסוף החלטנו להשאיר אותה.

2. $\underline{Vectorization}$:

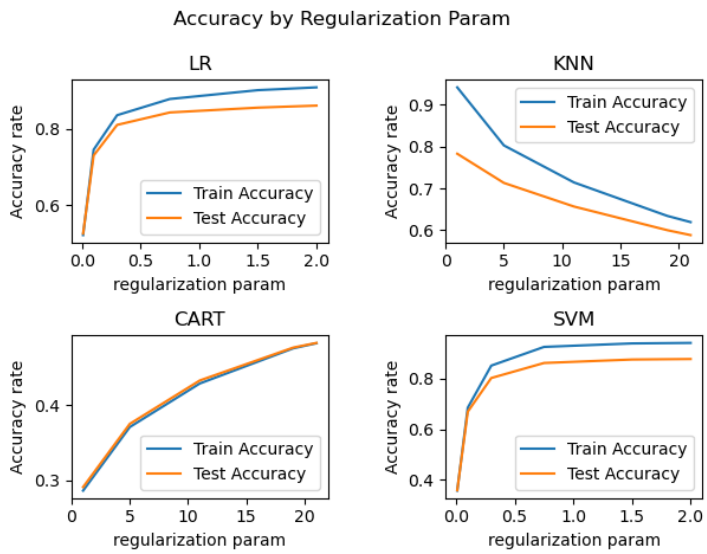
(א) לאחר בחינת ביצועיהם של מספר כלים לוקטוריזציה ($HashingVectorizer, CountingVectorizer$) נוכחנו כי ה־$TfidfVectorizer$ הוא הכלי האופטימלי עבור וקטוריזציה של הדגימות שלנו, הן מבחינת זמן ריצה והן מבחינת ביצועים.

(ב) בכדי לחדד את ביצועיו של ה־$vectorizer$ שלנו בחנו מספר פרמטרים שלו, ולבסוף גילינו כי הפרמטר המשפיע ביותר הוא פרמטר $maxfeatures$, ושהיחס האופטימלי הוא חצי ממספר הדגימות $train-set$.



חלק ג' - בחירת מודל ופיצ'רים

1. לסוגיית בחירת המודל ניגשנו עם 6 מסווגים סוגים.
2. מספר הדגימות הגדול $(130K)$ אפשר לנו לבצע סגמנטציה על הדאטא ללא צורך בשימוש ב- CV.
3. כדי לבחור מסווג מתאים פיצלנו את דאטת ה-$pre-proccesing$ לדאטת אימון ומבחן ואימנו כל אחד מ-6 המודלים עליה.
4. עבור מסווגים שיש להם פרמטר רגולציה הצענו טווח פרמטרים כאקט ראשוני של בחירת פיצ'רים (נרחיב בהמשך).
5. הצגנו את התוצאות על גרף (ראו מטה). שני המודלים שהציגו תוצאות טובות ביותר היו ה- SVM והלוגיסטיקה הלינארית.



6. החלטנו לבחור במודל ה-SVM משום שהוא הציג נתונים טובים יותר בזמן ריצה קצר באופן ניכר.
7. לאחר הבחירה ביצענו הרצה נוספת של בדיקות על SVM הן על דאטת ה-$pre-processing$ והן על דאטת ה- $validation$ במטרה לבחור פרמטרי רגולריזציה שיאפשרו טעות הכללה (=טעות על דאטת טסט שחילצנו באופן רדנומלי מדאטת $pre-proccesing$ וה-$validation$) מינימלית.
8. לאחר בחינת התוצאות החלטנו לבחור בפרמטרים הבאים: $\lambda = 1.5$. ה- λ הזה הוא אותו λ שממשקל את החריגה של הנקודה מ- $hyperplane$ מסוים.
9. בעזרת פרמטר $\lambda = 1.5$ על מודל SVM הגענו לרמת דיוק 86%.
10. בסיום הבדיקה אישרנו שאכן ה- $accuracy$ עולה עם עליית מספר הדגימות. את הבדיקה הזו ביצענו על ה- $train\ set$. זאת במטרה לכייל את עצמינו לקראת ביצוע ה-$train$ האחרון לפי ביצוע פרדיקציה על ה- $test\ set$.

חלק ד' - בחינת המודל

1. בסיום שלב בחירת המודל והפיצ'רים ביצענו אימון אחרון על כל דאטת ה- $train$ משמע 80% מהדאטא המקורית.
2. החלטנו לבצע אימון על מספר גדול של דגימות בהתאם לתובנה שלנו על ה-$sample\ complexity$. מדובר באימון על $90K$ דגימות.
3. בסיום שלב האימון האחרון ביצענו הרצה על ה-$test\ set$ ($40K$ דגימות) והגענו ל- 89.9% דיוק.