עבודת בית מסכמת - למידה חישובית

מגישות: הלל דודיאן 318593720 יהודית פרל 311596852

מטבע הדברים, כסף הוא תחום המעסיק את רוב האנשים בחיי היומיום, אנשים שואפים להרוויח יותר במטרה לחיות בנחות וללא דאגות עלכן בחרנו להתמקד בשאלה מה הרבדים המשפעים על הכנסותיו של האדם.

הקובץ נתונים שאנחנו בחרנו נלקח מאתר UCI ,שבו מאגרי נתונים ללימוד למידה חישובית . הקובץ מכיל מאגר מידע על אנשים ממדינות שונות. כולל בתוכו 14 מאפיינים ו עמודת תוצאה 32562 שורות. נרצה לבנות מודל שיחזה האם אדם מסוים ירוויח יותר או פחות 50 אלף בשנה.

בחרנו בנושא זה כי נושא שמעניין את רובנו, מכיל מידע רב, שברובו נתונים מורכבים המאפשר ביצוע של פילוחים שונים ומגוונים.

קבצי הקוד בפרוייקט:

- test – הקובץ הראשי אותו צריך להפעיל
- lgReg_handle – מכיל את הפונקציות הקשורות למודל LogisticRegression
- decision_tree_handle - מכיל את הפונקציות הקשורות למודל DecisionTree type
- csv_handle – מכיל את הפונקציות השורות לסידור קובץ הנתונים

בנוסף הקובץ economic_data מכיל את כל הנתונים

בחרנו להתמקד ב2 אלגוריתמים המבצעים classification :

1. logistic regression
2. decision tree

כיוון שמדובר בהמון שורות הפעלנו את האלגוריתמים בחלוקה למדינות

בעת ההרצה:

- יוצג כפלט כל מדדי השגיאות של המדינות עבור שני האלגוריתמים
- גרף להמחשת השגיאה על נתוני ה test על λ -ות שנבחרו
- מתוך כל ה λ מצאנו λ אופטימלית וחשבנו עבורו ערך השגיאה של train set ו test set על מס דוגמאות שונה
- תווצר תיקייה בשם countries_tree המכילה את הגרפים של עצי ההחלטה של כל המדינות

זאת בתנאי שהקובץ graphiz מוגדר במחשב
(נא להסתכל בנספח – "תוסף התקנה")

עבור כל שיטה ביצענו את ערכות השגיאה הבאות :

לצורך החישובים הגדרנו -

טבלה 1: המתייחסת למתחת/שווה 50 אלף

חזוי / אמיתי	מתחת\שווה ל 50 אלף	מעל 50 אלף
מתחת\שווה ל 50 אלף	TP	FN
מעל 50 אלף	FP	TN

TP- הכנסה שווה\מתחת ל50 והמודל צודק

FP- הכנסה מעל 50 והמודל טועה

TN- הכנסה מעל 50 והמודל צודק

FN – הכנסה שווה\ מתחת ל50 והמודל טעה

טבלה2: המתייחסת למעל 50 אלף

חזוי / אמיתי	מעל 50 אלף	מתחת\שווה ל 50 אלף
מעל 50 אלף	TP	FN
מתחת\שווה ל 50 אלף	FP	TN

TP- הכנסה מעל 50 והמודל צודק

FP- הכנסה שווה\ מתחת ל50 והמודל טועה

TN- הכנסה שווה\ מתחת ל50 והמודל צודק

FN – הכנסה מעל 50 והמודל טעה

וחשבנו את ממדי השגיאה (על סמך הטבלה השנייה) :

- recall- מבין אלו שמרוויחים מעל 50 אלף כמה זיהינו
- precision– מבין אלו שהמודל חזה מעל 50 אלף באיזה אחוז הוא צודק
- F-score
- Support – מתוך הy_test שנשלח כמה באמת נחזו כמו הערך האמיתי ב y_test

ממדי השגיאה חושבו פעם אחת ביחס לטבלה 1 (סימון 0 בקוד) ופעם אחת לטבלה 2 (סימון 1 בקוד)

בנוסף חושב (שיטה מובנית) :

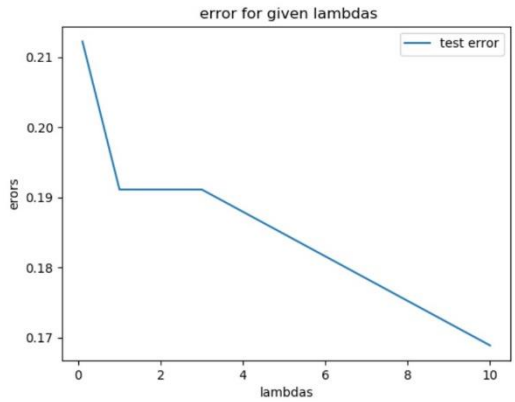
- micro avg- הממוצע הכולל של TP, FN, FP
- macro avg – ממוצע של ממוצע ללא משקל פר תווית
- weighted avg- ממוצע של support פר תווית.

אלגוריתם logistic regression

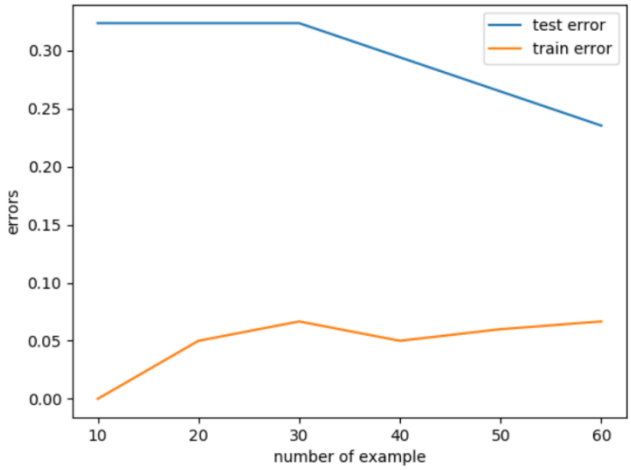
- קובץ הנתונים מכיל 14 עמודות של מאפיינים מתוכם 8 עמודות (לא כולל עמודת המדינות) כללו מחרוזות של תתי קטגוריות שדרשו פיצול עמודות ושינוי לערכים מספרים – סך הכל נוספו 40 עמודות
- לאחר הטיפול במחרוזות בוצע נרמול של כל הנתונים
- באמצעות אלגוריתם kfold הגדרנו חלוקה של הדאטא ל10 חלקים (K=10) כאשר 9 חלקים יהיו train set וחלק נוסף עבור ה test set הגדרנו K איטרציות כאשר בכל פעם חלק אחר מהחלקים שהוגדרו מהווה את ה test set
- הגדרנו מערך של למדות אפשריות ועבור כל למדה חשבנו את ערך השגיאה הממוצעת שהתקבלה
- לאחר מכן הפעלנו את האלגוריתם:
  - חשבנו את הערכות השגיאה שלמדנו
  - הגדרנו ערכי למדה (λ) וחשבנו עבורם את ערך השגיאה בtest
  - מתוך כל הלמדות מצאנו למדה אופטימלית וחשבנו עבורו ערך השגיאה של train set ו test set על מספר דוגמאות שונה

מסקנות

עבור הרצה עם המערך הבא: C_param_range = [np.inf, 10, 1, 0.5,(1/3),0.1]
(כאשר הלמדות זה 1 חלקי כל תא במערך זה)
ציירנו את הגרף של השגיאה לכל ערך של למדה
הלמדה האופטימלית היא 10 (בגרף רואים שלמדה זו נותנת את השגיאה הנמוכה ביותר)



עבור הלמדה הכי טובה ציירנו גרף שבו ציר ה-x הוא מספר הדוגמאות
וציר ה-y הוא השגיאה על קבוצת הלמידה והשגיאה על קבוצת test קבועה
מהגרף ניתן לראות שהמודל שהשגיאות עבור הנתונים שהוא מכיר נמוכות בהרבה מהשגיאות על הנתונים שהוא לא מכיר
כלומר overfitting



אלגוריתם decision tree

רצינו לחזות בשיטה זו כיון ששיטה זו מציגה באופן מוחשי ומובן יותר את החלוקה ומאפשרת הסקת מסקנות יעיה ומדויקת יותר

- עבור המאפיינים שכללו מחרוזות של תתי קטגוריות עשינו lable לכל תכונה נתנו כותרת במקום פיצול עמודות
- ביצענו נרמול של הנתונים והפעלנו את האלגוריתם עבור כל מדינה
- עבור כל מדינה הצגנו עץ החלטה[1] לפי מדד entropy ולפי מדד Gini Index

כל צומת בעץ כוללת את המשתנים הבאים[2]:

1. שם המאפיין
2. הרווח של עמודה (Gain)על סמך מדד השוני שנבחר (Gini Index \ entropy) הGain מייצג את הכדאיות של העמודה להיבחר שלפיה יבוצע הפיצול בהעץ. Gain מחושב באופן שונה בין מדד למדד ץ ככל שערכו גדול יותר העמודה כדאית יותר, כשערכו 0 משמע שמדובר בעלה.
3. מס הדוגמאות (samples) - מציג את מספר הערכים\ מאפיינים שנמצאים בקטגוריה זו.
4. ערך (value)- זה מערך
   תא 1- מספר אנשים שהכנסתם מתחת/ שווה ל50 .
   תא 2- שמציג את מספר האנשים שהכנסתם מעל 50 אלף בשנה

מסקנות

- בכל המדינות מרבית האנשים מרווחים מתחת . שווה ל 50 אלף שקלים
- המאפיינים המשפעים ב**עיקר** לשאלת ההכנסה במדינות שנבדקו הן: מעמד עבודת, מערכת יחסים, עיסוק וגיל
- מדובר באלגוריתם חמדן לבחירת תכונות חשובות המבצע פיצול על סמך פיצול הטוב ביותר בשלב מסוים, במקום להסתכל קדימה לפני הפיצול שיגרום חיזוי טוב יותר בשלב הבא. *מהווה חסרון משמעותי וכתוצאה מכך עלול לגרום ל overfitting*
- מדד entropy מביא ניבוי מדויק יותר בהתאם למדדי השגיאה
- להפתעתנו אין התייחסות לעמודת המין או הגזע בעצי ההחלטה חשבנו שיהיה לכך יותר השפעה, ככה״נ נובע מהיותו אלגוריתם חמדן

השוואה בין אלגוריתם decision tree לעומת logistic regression

- הבנת מסקנות מוצגת באופן ברור יותר ב decision tree
- השוואנו בין ממדי השגיאה על אותה קבוצת test של נתונים וראינו כי רמת הדיוק הכללית של האלגוריתם (סך הפעמים שמודל צודק בייחס לכל כל הבדיקות)

---

[1] תמונות העצים מוצגים בתקיה בשם countries_tree בתוך קובץ הzip שהוגש – נוצרים באת הפעלת האלגוריתם

[2] מצורפת תמונה להמחשה כנספח

לפי מדד entropy בעץ החלטה זהה בין 2 האלגורתמים ולפי מדד Gini Index בעץ ההחלטה, רמת הדיוק גבוהה יותר logistic regression יחד עם זאת שאר מדדי השגיאות טובים יותר ב logistic regression לעומת עץ החלטה לפי שני המדדים לכן נעדיף להשתמש באלגוריתם זה. **מסקנה זו תקפה בכל המדינות**[3]

נספחים

1. השוואה בין ממדי השגיאות לפי עץ החלטה למול logistic regression

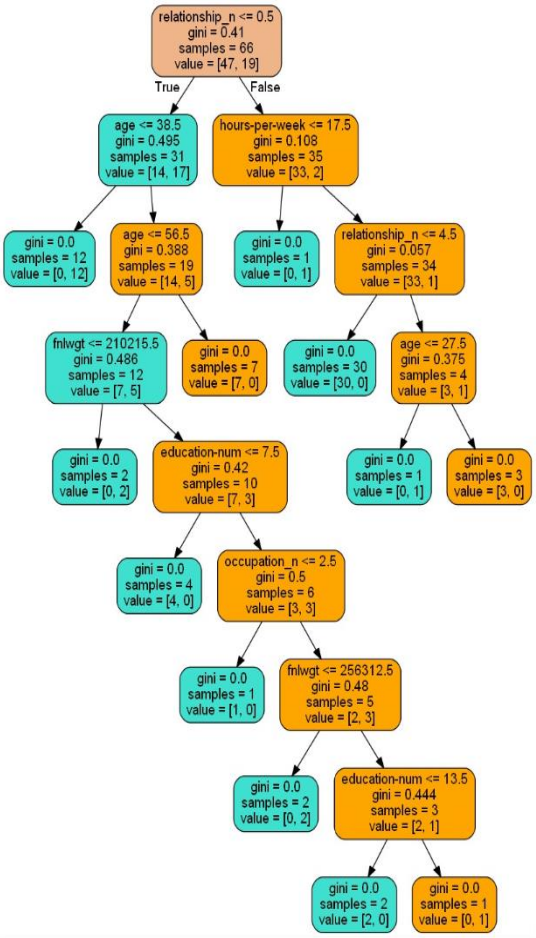
```
##############################################
#########     compare models    #############
##############################################
--------DecisionTree type entropy ---------------
Results of  DecisionTreeClassifier with  about the country  Cuba
accuracy 82.75862068965517
recall 50.0
precision 60.0
F_score 54.54545454545454
TPR 50.0
FPR 8.695652173913043
-------------------------------------------------
--------------LogisticRegression----------------
Results of  LogisticRegression  about the country  Cuba
accuracy 82.75862068965517
recall 95.65217391304348
precision 84.61538461538461
F_score 89.79591836734693
TPR 95.65217391304348
FPR 66.66666666666666
```

```
##############################################
#########     compare models    #############
##############################################
--------DecisionTree type gini ---------------
Results of  DecisionTreeClassifier with  about the country  Cuba
accuracy 75.86206896551724
recall 66.66666666666666
precision 44.44444444444444
F_score 53.333333333333336
TPR 66.66666666666666
FPR 21.73913043478261
-------------------------------------------------
--------------LogisticRegression----------------
Results of  LogisticRegression  about the country  Cuba
accuracy 82.75862068965517
recall 33.33333333333333
precision 66.66666666666666
F_score 44.44444444444444
TPR 33.33333333333333
FPR 4.3478260869565215
```

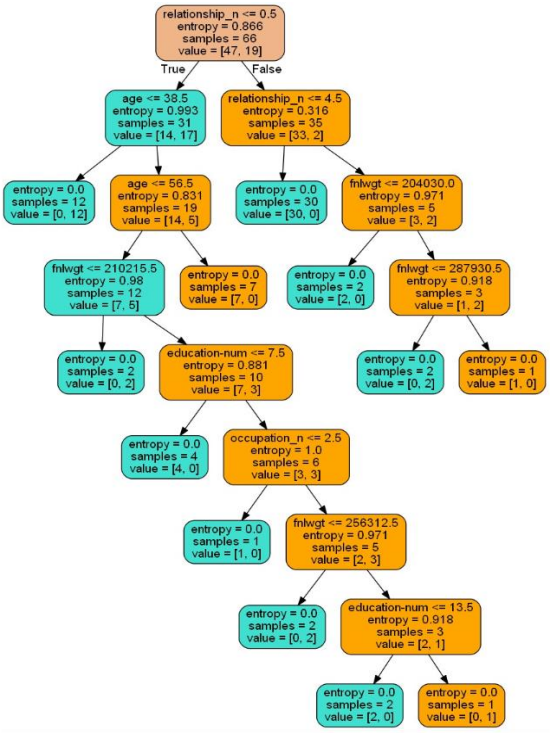
[3] מצורפת דוגמת מדינה כנספח, עבור כל המדינות מוצג קפלט בהרצת התוכנית

2. דוגמא לעץ החלטה של קובה

לפי מדד Gini Index



דוגמא עץ החלטה של קובה לפי מדד entropy



3. תוסף התקנה

על מנת להריץ את התוכנית צריך להוריד graphiz

ולחבר את path שלו

שלבים:

- הורדת graphiz מהאתר https://graphviz.gitlab.io/download/
עבור windows:
ttps://graphviz.gitlab.io/_pages/Download/Download_windows.html
- להוריד את הגרסאת msi

על מנת לחבר את Path:

- לחיצה על המאפפינים של "מחשב זה"
- לחיצה על הגדרות מערכת מתקדמות
- לחיצה על משתני סביבה
- לחיצה על משתני מערכת
- שם להוסיף את נתיב ה-bin של ההתקנה
- לצאת מpyCharm (או כל כתבן אחר) ולהיכנס