Introduction to Data Science

מטלה 4

סמסטר אביב 2020

מרצה:

ד"ר גייל גלבוע פרידמן

מתרגל:

מר עידן לופו

מר אפי פקאני

תאריך הגשה - 25/06/2020

מטרת התרגיל

מטרת התרגיל היא תרגול אלגוריתם Classification בשם Decision trees שלבי העבודה הם - הכנת הדאטה, תחקור סטטיסטי בסיסי של הנתונים, אימון המודל וניתוח תוצאות האלגוריתם.

התרגיל מתייחס לקובץ דאטה עם נתונים שנאספו ממערכת מובילאיי בת"א:

Mobileye_risk_TelAviv_collision_likelihood.csv

הגשה

- **הגש תיקיית zip ששמה S1_YOURNAME_YOURID.zip המכילה X קבצים (יודגש באדום בהנחיות): Word, ipynb.**
- יש להגיש Jupyter Notebook בשם: S4_YOURNAME_YOURID.ipynb.
- יש להגיש מחברת מסודרת המכילה:
  - תאי Markdown המחלקים את המחברת ל-sections בעזרת כותרות בהתאם לתרגיל (Part 1, Part 2 וכד').
  - תאי Code עבור שאלות תכנותיות.
  - במידה ותרצו להוסיף הערות לקוד ניתן להוסיף עם סימון הערה – "#".
  - **<u>לא</u> לרשום תשובות מילוליות בתוך המחברת/בתוך האקסל אלא רק ב-Word.**
- עבור **<u>כל שאלה</u>** רשמו את התשובה בקובץ Word והציגו תצלום מסך של הפלט כסימוכין.
  - עבור שאלות בהן רשום הסבר, **שימו לב** להוסיף תשובה מילולית ב-Word.
  - עבור שאלות בהן רשום קוד בלבד, ניתן לצרף צילום מסך של הקוד והפלט ל-Word.
  - דוגמא לתשובה תקינה:

ניתן לראות שבקובץ שלנו ישנם 2823 שורות ו12 עמודות:

```
rows, cols = data.shape

print('There are', rows, 'rows.')
print('There are', cols, 'cols.')
```

```
There are 2823 rows.
There are 12 cols.
```

בהצלחה !

(14 נק') חלק 1 - קריאת נתונים + סטטיסטיקה בסיסית

לתרגיל זה מצורף קובץ csv המכיל נתונים ממערכת מובילאיי על מקטעי דרך ברחבי תל אביב–

Mobileye_risk_TelAviv_collision_likelihood.csv

להזכירכם, מדובר בנתונים מחודש ינואר, כל תצפית מייצגת מקטע דרך אחר בעיר תל אביב,
לטובת התרגיל בוצעו שינויים בקובץ ועל כן יש לסקור מחדש את הנתונים ולבצע תהליך pre-processing בהתאם.

לנתונים נוספה עמודת "**collision_likelihood**" שתהווה עמודת הסיווג (lable) אותו נרצה לנבא. את מודל ה-DecisionTree נאמן בעזרת סט האימון ונבנה מודל עץ החלטה שינבא לנו את הסיווג עבור כל אחד מקטעי הדרך בסט הבחינה.

ראו את פירוט השדות מטה:

detection_drives_count – number of drives during the month.
avg_speed – average speed during the month.
near_miss_pedestrian_ratio – ratio of pedestrian collision warning during the month.
near_miss_bicycle_ratio – ratio of bicycle collision warning during the month
near_miss_vehicle_ratio – ratio of Forward collision warning during the month
avg_pedestrian_on_road_volume – average number of pedestrians detected on the drivable path during the month.
avg_bicycle_on_road_volume – average number of cyclists detected on the drivable path during the month.
braking_count – total number of braking during the month.
cornering_count – total number of cornering during the month.
harsh_braking_ratio – normalized ratio of harsh braking out of the total braking during the month.
harsh_cornering_ratio – normalized ratio of harsh cornering out of the total cornering during the month.
section_length – distance of the drivable path (in meters).
collision_likelihood – defined as 'high'/low' according to the probability to collide in this drivable path.

1. הורד את קובץ הנתונים הנ"ל.
2. קוד (2 נק') – ייבאו את הספריות: pandas, matplotlib.pyplot, numpy ותתי הספריות מתוך sklearn: DecisionTreeClassifier, train_test_split, preprocessing.
3. קוד (2 נק') – קראו את קובץ הנתונים Mobileye_risk_TelAviv_ collision_likelihood.csv לתוך DataFrame והריצו תיאור סטטיסטי על הנתונים.
4. הסבר (2 נק') – עברו על פלט התיאור הסטטיסטי שהצגתם (עמודה אחר עמודה) והסבירו לפחות ממצא אחד שעולה מהתיאור הסטטיסטי שלדעתכם יש להתייחס אליו לפני שנתחיל לעבד את הנתונים (preprocessing).
5. קוד (2 נק') – מהו אחוז המקטעים המסוכנים ואחוז המקטעים הלא מסוכנים עפ"י עמודה **collision_likelihood?**

6. קוד (2 נק') – הדפיסו את סכום התאים החסרים (NaN/NA) עבור כל אחת מהעמודות ב-DataFrame.
   a. הסבר (1 נק') – לאיזו עמודה חסר הכי הרבה ערכים?
   b. הסבר (3 נק') – הציעו דרך לעבד את הנתונים (preprocessing) עבור כל אחת מהעמודות כך שלא ישארו תאים ריקים ב-DataFrame שלנו.

חלק 2 – Preprocessing (22 נק')

על מנת להשתמש באלגוריתם DecisionTree עלינו לוודא שאין לנו תאים רקים בטבלת הנתונים:

7. קוד (2 נק') – עמודת 'harsh_braking_ratio' אינה מכילה מספיק מידע עבורנו, כמעט מחצית מערכיה ריקים והערכים שכן מלאים מרביתם '0'. מחקו עמודה זאת מה-DataFrame שלנו.
8. על מנת למלא את שאר הערכים הריקים, נבצע Imputing על בסיס המידע הקיים לנו בכל עמודה:
   a. קוד (3 נק') – עבור עמודה 'near_miss_pedestrian_ratio' בצעו impute עם הערך הממוצע של העמודה (mean).
   b. קוד (3 נק') – עבור עמודה 'near_miss_bicycle_ratio' בצעו impute עם הערך החציוני של העמודה (median).
   c. קוד (6 נק') (מאתגר) – עבור עמודה 'braking_count' בנו פונקציה בשם "fill_NaN_with_normalRand" אשר תקבל עמודה אחת מתוך ה-DataFrame (שיש בה ערכים ריקים) ותמלא אותם באופן אקראי (random) עפ"י ההתפלגות הנורמלית של הערכים <u>בעמודה זו</u>. (טיפ: בנו מסיכה (mask) עבור האינדקסים החסרים בעמודה ולאחר מכן מלאו אותה ע"י שימוש בפונקצית random.normal של numpy).
   d. קוד (2 נק') – הפעילו את הפונק' מהסעיף הקודם על העמודה 'braking_count'.
   e. קוד (2 נק') – וודאו שלא נשארו ערכים ריקים באף אחת מהעמודות ב-DataFrame שלנו.

9. קוד (2 נק') – השתמשו בפונק' train_test_split שיבאתם על מנת לחלק את הדאטה לסט אימון וסט בדיקה, שיעור קבוצת הבדיקה יהיה 20% מתוך כלל הנתונים ויבחר באופן רנדומלי (random_state=0) בעזרת הפונק'.
10. קוד (2 נק') – הדפיסו למסך את גודל (shape) של כל אחד מטבלאות הדאטה שיצרתם בחלוקה: X_train, X_test, y_train, y_test, וודאו שאכן החלוקה בוצעה ביחס של 20%.

חלק 3 – Decision Tree (64 נק')

בחלק זה ניצור את עץ ההחלטה שלנו שילמד כיצד לסווג מקטעי דרך לפי סט האימון, תחילה נרצה לבחון מהו עומק העץ האידיאלי לבניית המודל, לאחר מכן נבצע חישובי ג'יני (Gini Impurity) על מנת להבין מדוע העץ נבנה כמו שהוא.

חלק 3 א' - ביחנו את עומק העץ האופטימלי בעזרת מדד Accuracy (22 נק')

11. קוד (2 נק') – צרו 2 רשימות ריקות בשם: train_acc, test_acc עבור תוצאות מדד הדיוק (Accuracy) של סט הבדיקה וסט האימון.
12. קוד (2 נק') – הריצו לולאה בטווח של 20 ערכים וצרו מסווג עץ החלטה בעל עומק ענפים "d" עבור כל אחד מהערכים (1-20 כולל), עבור כל ערך התאימו את המסווג לסט האימון (fit).
13. קוד (2 נק') – בתוך אותה לולאה, הוסיפו לרשימות שיצרנו בשאלה 12 את ציון הדיוק עבור סט נתוני האימון וסט נתוני הבדיקה בהתאמה (כך שנקבל 2 רשימות עם 20 ציונים כל אחת, כל אחד מהערכים יהווה ציון דיוק עבור עומק עץ "d").
14. קוד (4 נק') – צרו גרף Scatter על בסיס 2 הרשימות שיצרתם. הגרף יציג את תוצאות מדד הדיוק עבור כל אחד מעומקי העץ השונים. שימו לב לסמן את תוצאות סט האימון בצבע שונה מתוצאות סט הבדיקה (כמו כן, יש לתת כותרות לצירים וכותרת לפירוש הצבעים, בדומה לנעשה בתרגול).

15. בהתבסס על הגרף שיצרתם בסעיף הקודם:

a. הסבר (3 נק') – האם אתם יכולים לזהות מגמה כלשהי בתוצאות מדד הדיוק של <u>סט האימון</u> בהתבסס על עומק העץ? אם כן, מהי המגמה? מדוע היא מתקיימת?

b. הסבר (3 נק') – האם אתם יכולים לזהות מגמה כלשהי בתוצאות מדד הדיוק של <u>סט הבדיקה</u> בהתבסס על עומק העץ? אם כן, מהי המגמה? מדוע היא מתקיימת?

c. הסבר (6 נק') – בהינתן הגרף, באיזה עומק עץ הייתם ממליצים להשתמש? מדוע?

<u>(30 נק') חלק 3 ב' – חישוב ג'יני (Gini Impurity)</u>

מחלק זה השתמשו בעץ החלטה בעל הפרמטרים:

DecisionTreeClassifier(max_depth=6, min_samples_leaf=10, random_state=0)

להזכירכם נוסחת חישוב Gini information gain:

$$\mathrm{I}_G(p) = \sum_{i=1}^{J}\left(p_i \sum_{k\neq i} p_k\right) = \sum_{i=1}^{J} p_i(1-p_i) = \sum_{i=1}^{J}(p_i - {p_i}^2) = \sum_{i=1}^{J} p_i - \sum_{i=1}^{J} {p_i}^2 = 1 - \sum_{i=1}^{J} {p_i}^2$$

או בקצרה:

$$I_G(p) = 1 - \sum_{j=1}^{N} \mathrm{p}_j^2$$

16. קוד (4 נק') – חשבו בעזרת הנוסחה לעיל את ערך הג'יני בסט האימון (ג'יני שלב ראשון).
17. קוד (6 נק') – חשבו את ערך הג'יני עבור המשתנה "near_miss_pedestrian_ratio " בהנחה ואלגוריתם עץ ההחלטה החליט לחלק את סט האימון בערך X <= 0.002 (ג'יני שלב שני, רמז: בנו שתי טבלאות עפ"י אם התנאי מתקיים או לא מתקיים כפי שעשינו בתרגול. הוציאו את הנתונים הנדרשים לנוסחה חשבו את הג'יני במחברת יופיטר והעתיקו את התשובה ל-Word).
18. קוד (6 נק') – חשבו את ערך הג'יני עבור המשתנה "section_length" בהנחה ואלגוריתם עץ ההחלטה החליט לחלק את סט האימון בערך X <= 23 (ג'יני שלב שני).
19. הסבר (5 נק') – סכמו בקצרה מהאינטרנט כיצד אלגוריתם עץ החלטה קובע מאיזה משתנה כדאי להתחיל לחלק את העץ (Feature Importance), בתשובתכם התייחסו לערך הג'יני (Gini Impurity) הנדרש עבור המשתנה הראשון ממנו מתחיל העץ.

20. הסבר (5 נק') – בהתאם לתשובה שמצאתם לשאלה 19,
מאיזה משתנה אלגוריתם עץ ההחלטה שלנו יבחר להתחיל את הפיצול הראשון מבין שני המשתנים שעבורם חישבנו ג'יני (ג'יני שלב שני בשאלות: 17,18).
21. קוד (4 נק') – הציגו בעזרת גרף בארים את החשיבות של כל אחד מהמשתנים במודל עבור עץ ההחלטה עם הפרמטרים שהוגדרו בתחילת חלק זה. הציגו את החשיבות בסדר עולה מהקטן לגדול, הקפידו להציג כותרת לגרף (בהתאם לנעשה בתרגול).

(12 נק') חלק 3 ג' – ויזואליזציה

22. קוד (4 נק') – באמצעות פונקציית export_graphiz צרו גרף של עץ ההחלטה בעל הפרמטרים שהוגדרו בתחילת חלק 3 ב'.
23. קוד (6 נק') – בהינתן העץ שציירתם בסעיף הקודם, תארו את המאפיינים של מקטע הדרך בו קיימת ההסתברות הגבוהה להיות מסווג כ-"high", כיצד ניתן לזהות בקלות מי הם קטעי הדרך האלו בעזרת צבעי העץ? ומהי ההסתברות שמקטע דרך זה יהיה מסווג כ-"high" ?
(רמז: קיימים 2 סוגי מקטעי דרך בעלי סבירות זהה שהיא הגבוהה ביותר להתנגשות).
24. קוד (2 נק') – שמרו קובץ png או pdf המכיל את גרף עץ ההחלטה שקיבלתם.

References:

- https://towardsdatascience.com/decision-tree-in-python-b433ae57fb93
- https://www.kdnuggets.com/2019/08/understanding-decision-trees-classification-python.html
- https://www.python-course.eu/Decision_Trees.php

בהצלחה!