מטלת מנחה 21

איתמר יורן

# כריית מידע – חיזוי מחלת כליות כרונית

### שאלה 1

## מטרות כריית המידע: בפרוייקט זה נשתמש בכריית מידע על מנת לחזות התפרצות של מחלת כליות כרונית עפ״י נתונים בריאותיים של פציינטים. אנו מסתמכים על מאגר רשומות בן 400 רשומות שונות שנאספו בארה״ב. ההנחות עליהן אני מסתמך במחקר זה, הן בראש ובראשונה שהמידע הניתן לי מהימן. נוסף על כך, הרשומות מגוונות מספיק כדי שתוצאת המחקר והסיווג המתקבל, יוכל לחזות בצורה איכותית מקרים שמחוץ למאגר מידע זה.

## הגדרת הנתונים: מצורפת הדפסת python עבור כל עמודות הטבלה.

| Index | Tag | Name | Data Type |NaN Values| Values (category)| Mean | Median | STD |

|---------|-------|-------------------------|-------------|----------|------------------|---------|----------|---------|

| | | | | | | | | |

| 1 | age | age | Numeric | 9 | - | 51.48 | 55.0 | 17.17 |

| 2 | bp | blood pressure | Numeric | 12 | - | 76.47 | 80.0 | 13.68 |

| 3 | sg | specific gravity | Numeric | 47 | - | 1.02 | 1.02 | 0.01 |

| 4 | al | albumin | Numeric | 46 | - | 1.02 | 0.0 | 1.35 |

| 5 | su | suger | Numeric | 49 | - | 0.45 | 0.0 | 1.1 |

| 6 | rbc | red blood cells | Category | 152 | normal 201 | - | - | - |

| | | | | | abnormal 47 | - | - | - |

| 7 | pc | pus cells | Category | 65 | normal 259 | - | - | - | | | | | | | abnormal 76 | - | - | - |

| 8 | pcc | pus cells clumps | Category | 4 | notpresent 354| - | - | - |

| | | | | | present 42| - | - | - |

| 9 | ba | bacteria | Category | 4 | notpresent 374| - | - | - |

| | | | | | present 22| - | - | - |

| 10 | bgr | blood glucose random | Numeric | 44 | - | 148.04 | 121.0 | 79.28 |

| 11 | bu | blood urea | Numeric | 19 | - | 57.43 | 42.0 | 50.5 |

| 12 | sc | serum creatinine | Numeric | 17 | - | 3.07 | 1.3 | 5.74 |

| 13 | sod | sodium | Numeric | 87 | - | 137.53 | 138.0 | 10.41 |

| 14 | pot | potassium | Numeric | 88 | - | 4.63 | 4.4 | 3.19 |

| 15 | hemo | hemglobin | Numeric | 52 | - | 12.53 | 12.65 | 2.91 |

| 16 | pcv | packed cell volume | Numeric | 71 | - | 38.88 | 40.0 | 8.99 |

| 17 | wbcc | white blood cells count | Numeric | 106 | - | 8406.12 | 8000.0 | 2944.47 |

| 18 | rbcc | red blood cells count | Numeric | 131 | - | 4.71 | 4.8 | 1.03 |

| 19 | htn | hypertension | Category | 2 | no 251 | - | - | - |

| | | | | | yes 147 | - | - | - |

| 20 | dm | diabetes mellitus | Category | 2 | no 261 | - | - | - |

| | | | | | yes 137 | - | - | - |

| 21 | cad | coronary artery disease | Category | 2 | no 364 | - | - | - |

| | | | | | yes 34 | - | - | - |

| 22 | appet | appetite | Category | 1 | good 317 | - | - | - |

| | | | | | poor 82 | - | - | - |

| 23 | pe | pedal edema | Category | 1 | no 323 | - | - | - |

| | | | | | yes 76 | - | - | - |

| 24 | ane | anemia | Category | 1 | no 339 | - | - | - |

| | | | | | yes 60 | - | - | - |

| 25 | class | class | Category | 0 | ckd 250 | - | - | - |

| | | | | | notckd 150 | - | - | - |

## KDD

## 1. מטרות כריית המידע: מטרת כריית המידע בפרויקט זה היא לחזות האם נבדק מסויים יחלה או לא במחלת כליות כרונית (CKD), בהתבסס על נתונים ומידע רפואי של 400 נבדקים.

## 2. איסוף הנתונים: את מאגר הנתונים המלא קיבלנו מאתר בשם “UCI” ומקורו ב”Enam medical college” הנתונים הגיעו בשלמותם ועל כן לא נזקקנו לאינטרגציות בין מאגרי נתונים

## 3. ניקוי הנתונים: חלק מהנתונים חסרים, אין נתונים חריגים. בהתאם לשיטה שתבחר ננקה את הנתונים. נבצע זאת בעזרת מילוי ערכים מסויים ומחיקת שורות בהן חסרים נתונים רבים.

## 4. טרנספורמצית נתונים: בפרויקט זה השיטות בהן השתמשתי לא דרשו טרנספורמציות נתונים כנרמול או דיסקרטיזציה.

## 5. כלים ושיטות לכריית המידע: בפרוייקט זה, השתמשתי בpython בלבד. למען הנוחות השתמשתי בJupyter notbook ובספריות כpandas,numpy,matplotlib וsickitlearn על מנת להציג ולנתח את הנתונים.

## 6. התאמת ייצוג הנתונים: התאמת הנתונים לשיטת כריית המידע שנבחרה. הספרייה scikit-learn נוחה לשימוש וכמעט ואינה מבקשת שינוי בהצגת הנתונים.

## 7. הרצת שיטות כריית המידע: נריץ אץ האלגוריתמים הנבחרים על מאגר המידע המותאם שיצרנו.

## 8. ניתוח תוצאות: נבדוק את מידת הדיוק של האלגוריתמים בהם השתמשנו ע״י בדיקת החלק במאגר הנתונים אותו ייעדנו למבחן. כלים כמו מטריצת ערפול ועקומת רוק יעזרו לנו בניתוח סטטיסטי של הצלחת האלגוריתם בסיווג נכון.

## שיטות כריית המידע העומדות בפנינו

## רגרסיה ליניארית שיטה זו מתאימה לחיזוי ערכים רציפים, שיטה זאת מתאימה כאשר ישנו קשר חזק בין חלק מהמשתנים הבלתי תלויים למשתנה התלוי. השיטה מנסחת קשר בין המשתנים הבלתי תלויים לתלויים ועל כל קלט שיקבל המודל, יפלוט ערך בתחום הערכים של המטרה. יתרונות השיטה הם מימוש קל ונח בפייתון. תצוגה גרפית קלה חסרונות הברורים של השיטה היא שבפרויקט שלנו איננו עובדים עם ערכים רציפים. המודל שאנו מבקשים הוא מודל המסווג בין שתי קבוצות.

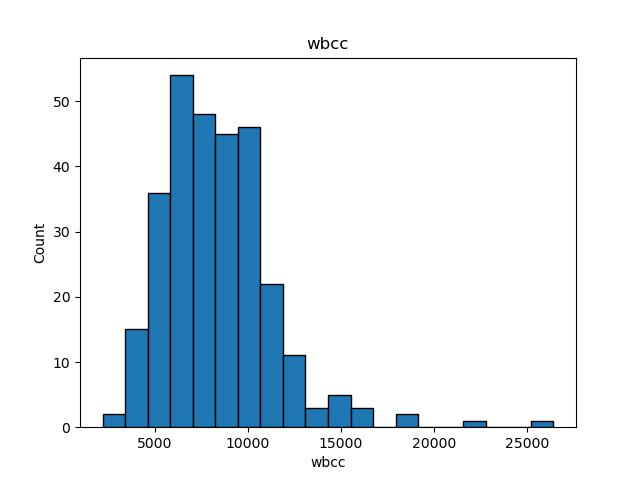
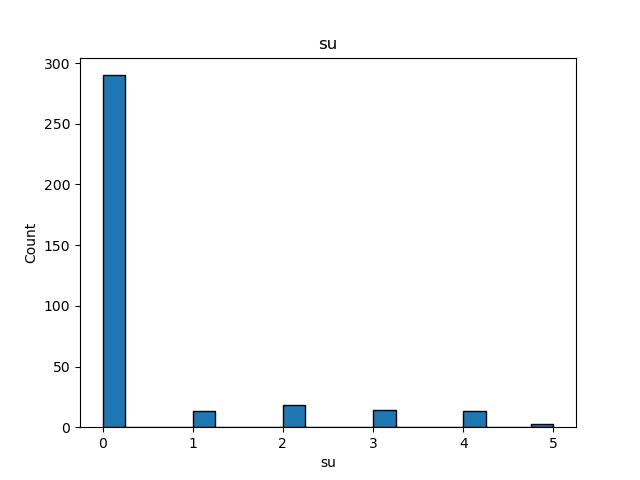
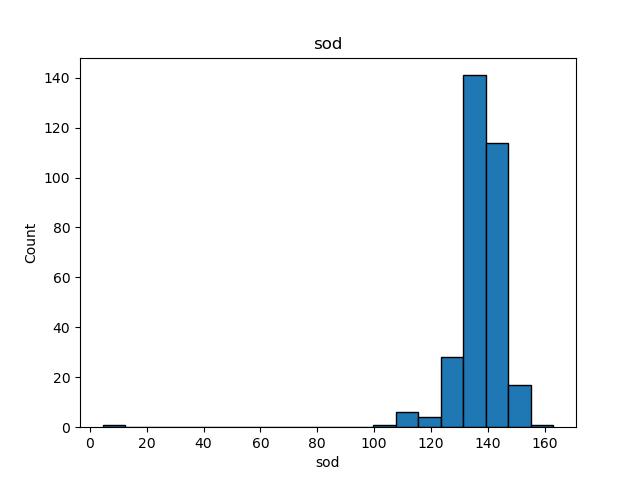
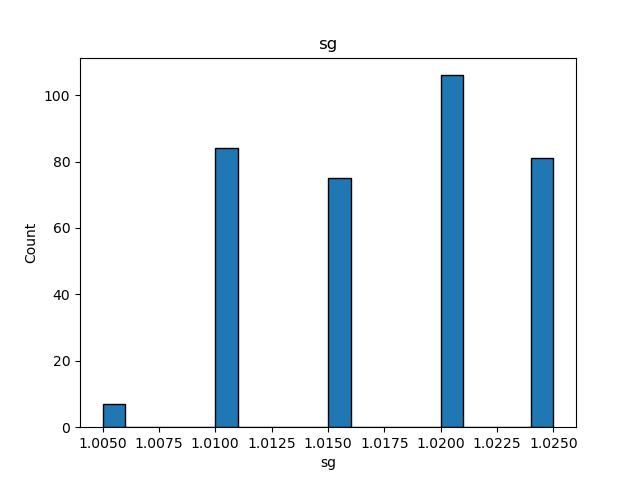
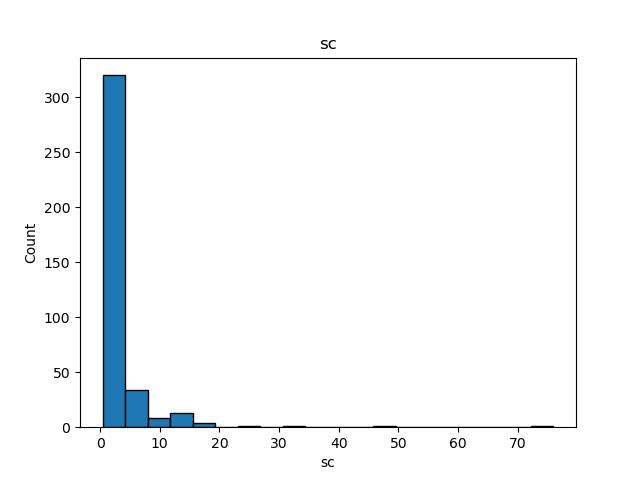
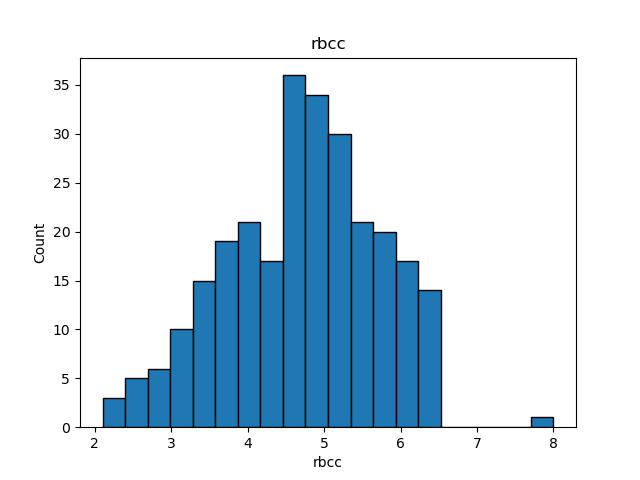
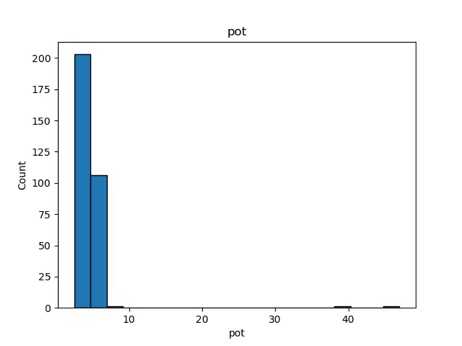
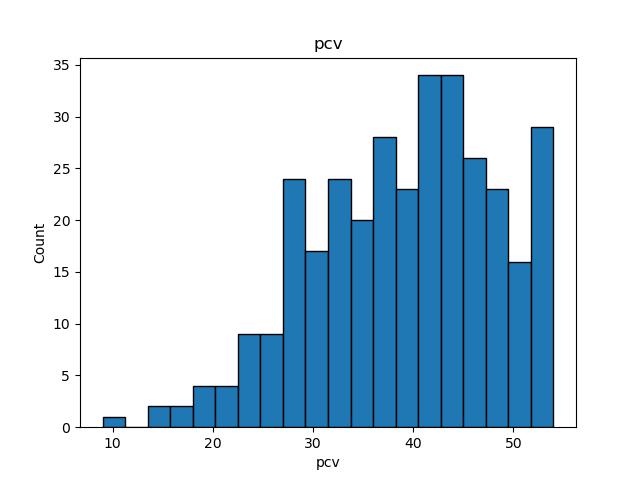
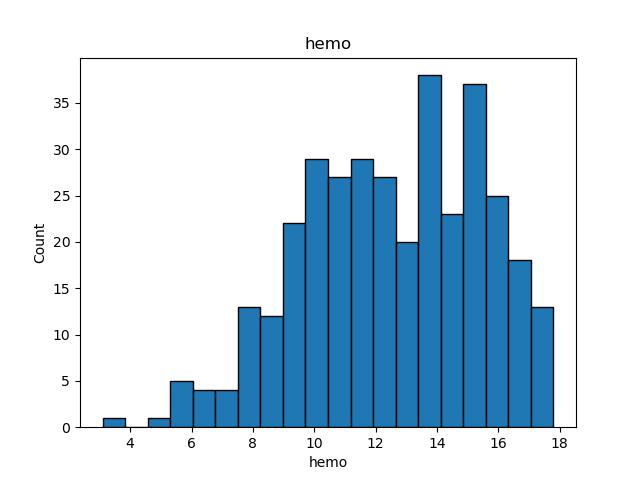
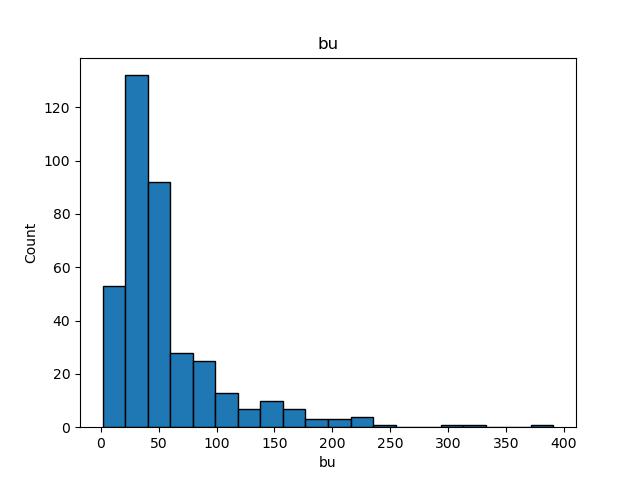
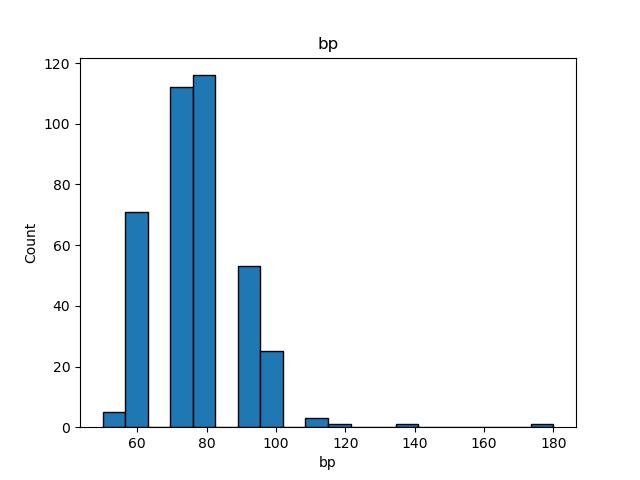
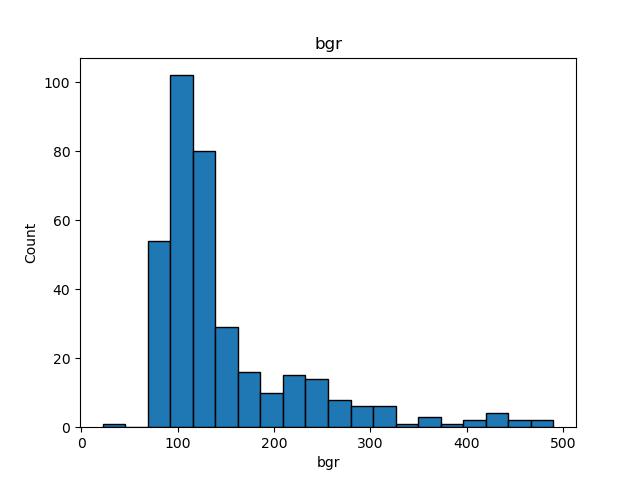
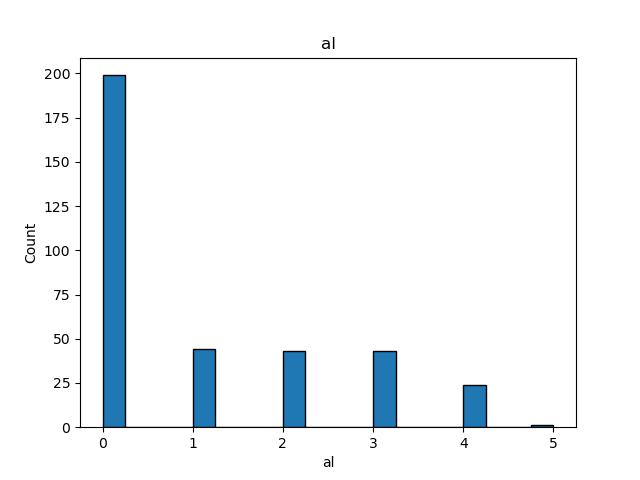
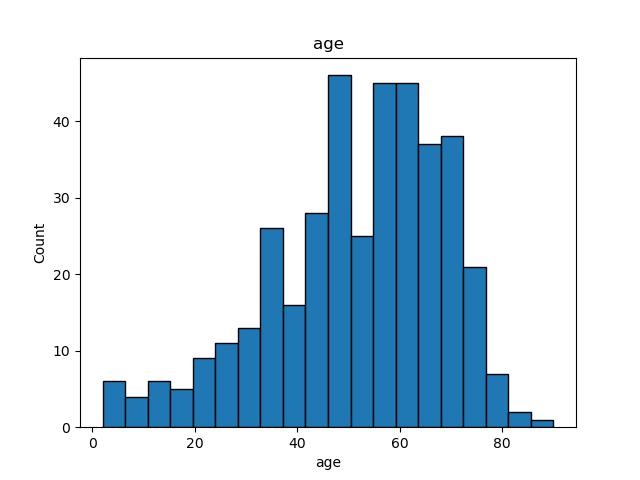
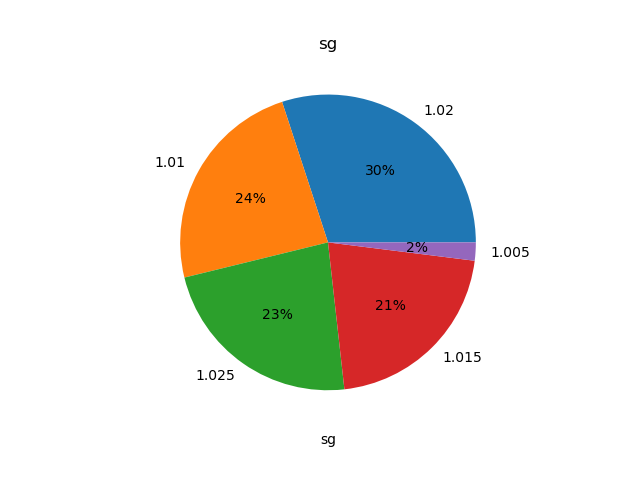
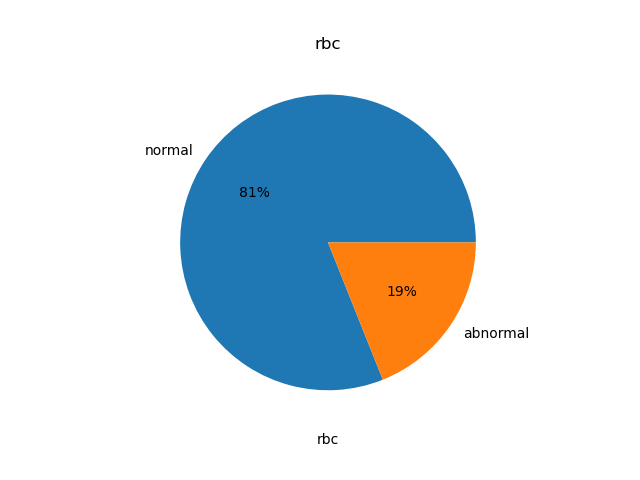
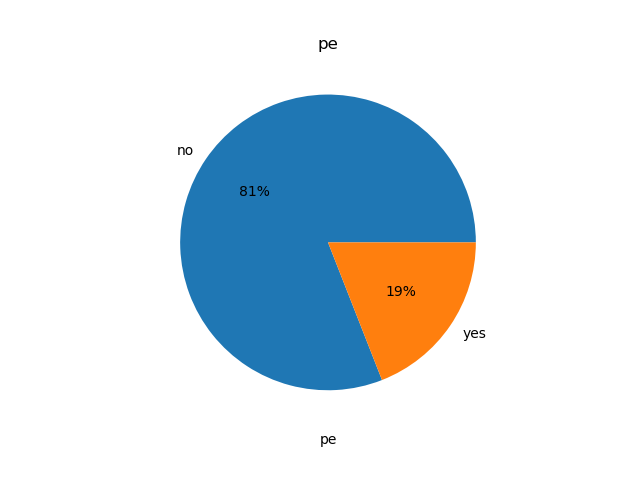
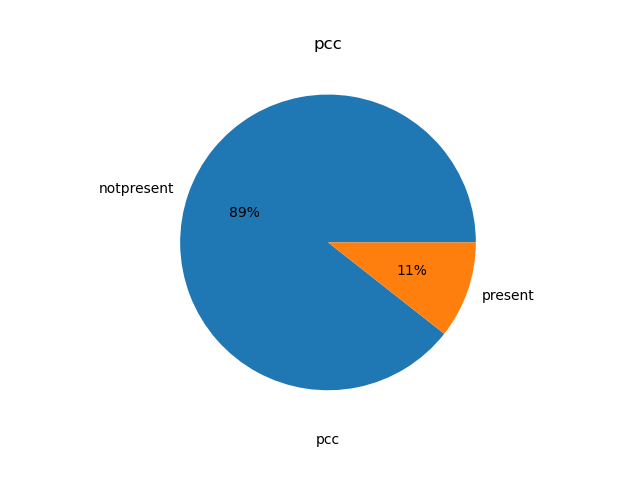
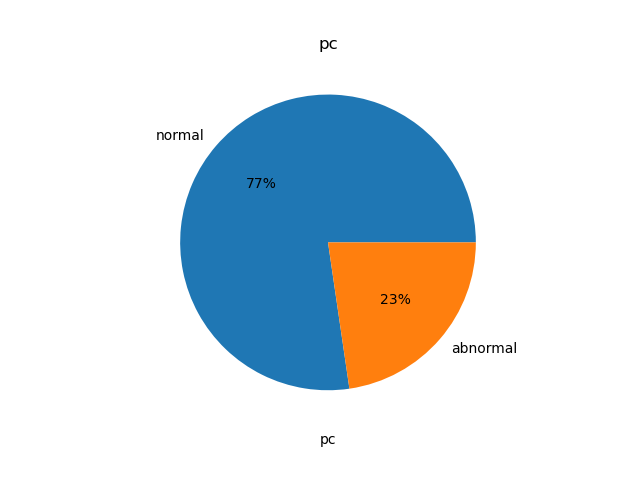
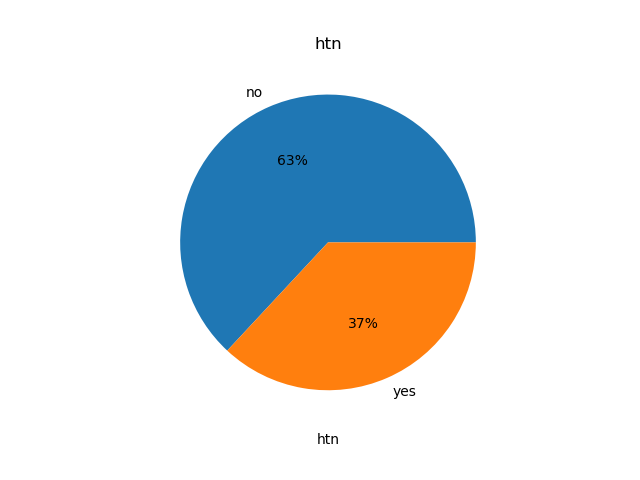
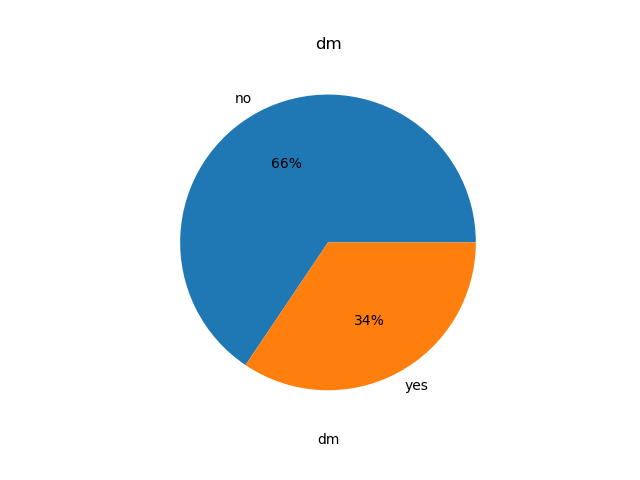
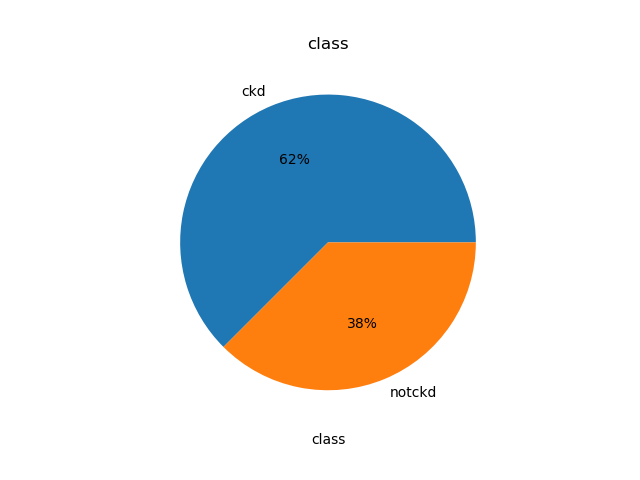
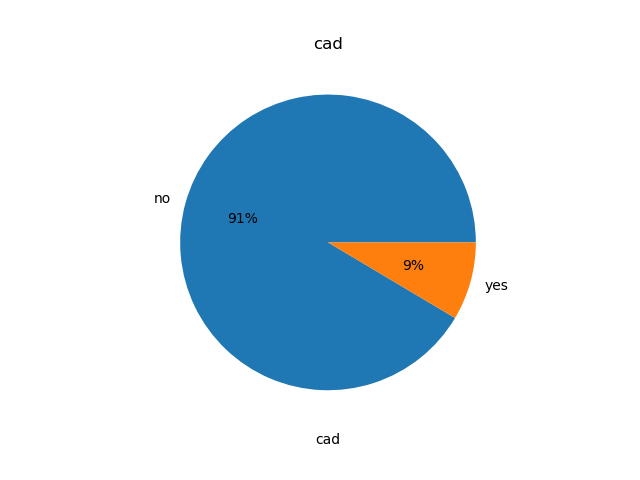
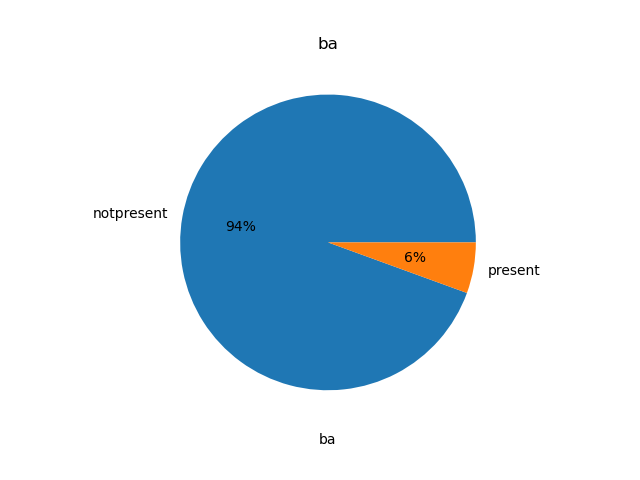
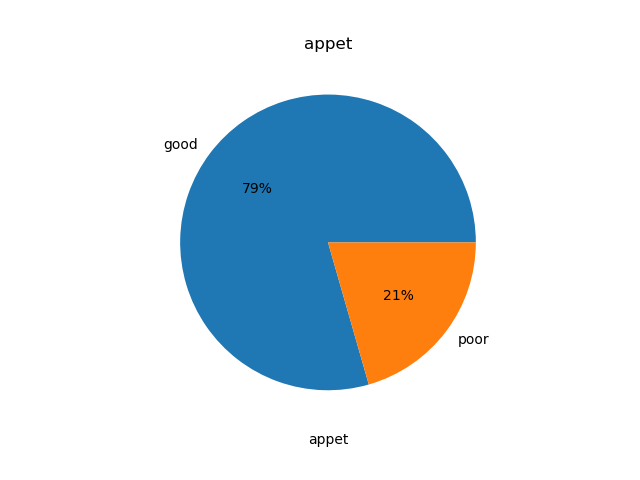
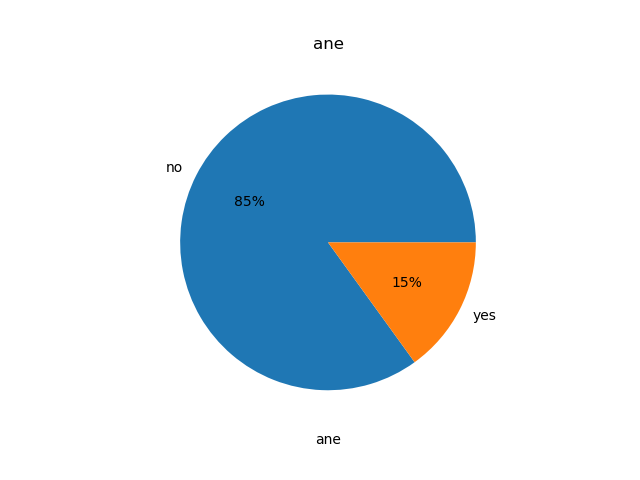
## רגרסיה לוגיסטית הרגרסיה הלוגיסטית כמובן קשורה קשר עמוק לרגרסיה הליניארית שתיארתי קודם לכן. התהליך בשניהם דומה. ההבדל הוא שרגרסיה לוגיסטית הוא מודל מסווג. הוא אינו מחזיר ערך רציף אלא קבוצת סיווג עבור כל רשומה. יתרון השיטה הוא כרגרסיה ליניארית, יישום קל ונח. יכולת תצוגת נתונים נוחה. החסרון הגדול של השיטה, הוא שהמודל מניח כי קיים קשר ליניארי בין המשתנים. אין למודל יכולת לתאר קשר מתמטי עמוק או נסתר יותר.

## עץ החלטה מבוסס רווח אינפורמטיבי (ID3) עץ החלטה זה הוא בסיסי ביותר, האלגוריתם שלו פועל בצורה רקורסיבית חמדנית (מתכנן רק את הצעד הקרוב) בכל שלב בוחר הלאגוריתם את התכונה שפיצולה יתרום הכי הרבה Information Gain. יתרונות השיטה הן, שאלגוריתם זה נחשב פשוט, אך עם הכלים הקיימים היום איני רואה יתרון המתעדף אותו על מודלים אחרים. חסרונותיו הם, עבודה רבה על הנתונים טרם הפעלת האלגוריתם. בנוסף התבססות על רווח אינפורמטיבי יכולה לתעדף תכונות מרובות ערכים, ובכך ליצור מצב של overfitting.

## עץ החלטה מבוסס Gini Index (Cart) אלגוריתם זה מייצר עץ החלטה איכותי, מבוסס על gini index. האלגוריתם יכול לעבוד עם ערכים בדידים ורציפים. פעולתו של האלגוריתם היא מציאת התכונה בכל איטרציה המפצלת את המאגר לשתי קבוצות השואפות להיות שוות בגודלן. יתרונות העץ הם, שהוא מייצר עץ קומפקטי ומאוזן יחסית. חסרונותיו של האלגוריתם הוא שנחשב בזבזני.

## ה. הכנת הנתונים מאגר הנתונים איתו אנו עובדים הוא מאגר בן 400 שורות שנאספו בהודו, והגיעו אלינו מאתר UCI. במאגר הנתונים אין נתונים החורגים בהרבה מטווח הנתונים לכן התעסקתי בעיקר בנתונים החסרים. תחילה, ביצעתי תיקונים על מנת להתאים את המאגר לעבודה עם python,pandas וכו׳, פעולות כגון הורדת פסיקים, גרשיים, שינוי סוג האובייקט לסוג המתאים ועוד. לאחכ מכן הורדתי מן המאגר שורות בהן חסרים מעל לשלושים אחוז מן הקריטריונים, על מנת שלא יווצר מצב שמילויין ישפיע על אלגוריתמי הסיווג. לאחר סינון זה נשארתי עם 369 שורות. בpython עבור חלק גדול מהאלגוריתמים שראיתי אין צורך בהשלמת הערכים החסרים אך חלק כן דורשים זאת. לשם כך פעלתי בדרך הבא – עבור קריטריונים קטגוריליים מילאתי ערך חסר באחת מן הקטגוריות בהסתברות השווה להסתברות למציאתן בשורה. עבור ערכים נומריים לקחתי את הממוצע והצבתי אותו במקום הערך החסר.

## לא נדרשתי עבור מאגר הנתונים הנ״ל, והכלים בהם השתמשתי, לטרנספורמציות כגון דיסקרטיזציה או נורמליזציה. חלק מהאלגוריתמים דרשו ערכים נומריים בלבד ולשם כך המרתי כל אחת מן הקטגוריות לערך מספרי. מצורפים מטה הנתונים לאחר השמטת הערכים החסרים.



שאלה 2

## השיטות בהן בחרתי הן עץ החלטה, ורגרסיה ליניארית ולוגיסטית. עץ החלטה CART הוא אלגוריתם המתאים למצב של סיווג שתי קבוצות שונות הוא מניב תוצאות איכותיות. רגרסיה לוגיסטית גם היא שיטה המתאימה לסווג בינארית מאגר של תכונות. יישום שני האלגוריתמים נח יחסית דרך ספריית scikit learn.

## תיאור שלבי השיטות עץ החלטה: השתמשתי בספריית scikit-learn על מנת לבנות עץ החלטה מבוסס מדד Gini המשתמש באלגוריתם CART. האלגוריתם פועל בצורה איטרטיבית כל שבכל איטרציה הוא בוחר את התכונה הרלוונטית ביותר. זאת נקבעת עפ״י חישוב מדד הגיני של כל עמודה וחישוב איזה תכונה תוריד את מדד הגיני כמה שיותר. האיטרציות עבור ענף מסויים בעץ נפסקות כאשר מדד הגיני של קבוצה שווה לאפס, המשמעות היא כמובן שהצומת מייצג רשומות של קבוצה אחת בלבד. האלגוריתם יכול גם לסיים את עבודתו כאשר הגיע לעומק המקסימלי שהוגדר לו, עובדה זאת רצויה על מנת שלא יווצר מצב של overfitting. רגרסיה לוגיסטית: רגרסיה לוגיסטית, מחשבת את הסיכוי, של קבוצת פרמטרים, להיות בקבוצות סיווג מסויימות. במקרה שלנו האם אותו אדם חולה במחלת הכליות או לא. האלגוריתם בצורה איטרטיבית מנסח ומשפר פונקציית הסתברות מהצורה (( P(y=1|x) = 1 / (1 + exp(-z) כאשר z הוא קומבינציה ליניארית של מקדמים, אשר כל אחד מהמקדמים מייצג תכונה מסויימת במאגר המידע, וכאשר y=1 or y=0 הן קבוצות הסיווג האפשריות. האלגוריתם בעת שהוא ״מתאמן״ בעצם מחפש את הקומבינציה הליניארית המתאימה ביותר לסיווג כך שבהכפלת המקדמים בנתוני הטבלה תספק הנוסחה ערך השואף ל1 או 0.

## עבור עץ ההחלטה התקבל העץ הבא: Timeline Description automatically generated with medium confidence

## קיבלנו עץ החלטה מעומק מירבי של חמש מספר הצמתים בו הוא 19 מתוכם מספר העלים הינו עשר. העץ יחסית קומפקפטי והמשמעות של כך היא שהעץ מסווג יחסית בקלות בין הקבוצות ולא משתמש בכמות גדולה של פרמטרים. כפי שניתן לראות העץ משתמש בתכונות בודדות מהטבלה בלבד כגון, Hemoglobin, blood urea,specific gravity ועוד.

## עבור רגרסיה לוגיסטית התקבלה טבלת המקדמים הבאה: Chart Description automatically generated

## המשמעות היא מידת ההשפעה של הפרמטר על החלטת הסיווג. ערך מוחלט גבוהה יחסית מצביע על קשר חזק יותר של המשתנה הבלתי תלוי על התוצאה הסופית.

## בין שתי השיטות ניתן לראות קשר חזק במקדמים שהיוו משקל גדול יותר בסיווג המשתנה.

## ד. מידת הדיוק של האלגוריתמים

Decision Tree

Confusion Matrics: [[46 3]

[ 0 62]]

Accuracy: 0.972

Precision: 0.953

Recall: 1.0

F1: 0.976

Auc: 0.969

Cross-validation scores: [0.959 0.945 0.945 0.986 0.972]

Logistic Regression

Confusion Matrics: [[44 5]

[ 8 54]]

Accuracy: 0.883

Precision: 0.915

Recall: 0.871

F1: 0.893

Auc: 0.884

Cross-validation scores: [0.932 0.932 0.824 0.945 0.945]

Linear Regression

Confusion Matrics: [[49 0]

[ 4 58]]

Accuracy: 0.964

Precision: 1.0

Recall: 0.935

F1: 0.967

Auc: 0.968

Cross-validation scores: [0.932 0.932 0.824 0.945 0.945]

## Confusion matrics - מטריצת הערפול מראה לנו את לנו את מספר ה- True positive,True negative,False positive,False negative Accuracy – הוא סך כל המופעים שסווגו נכון חלקי כל המופעים במאגר המידע. Precision - Recall - F1 – ממוצע הרמוני בין precision ו recall Cross validation score - מדד זה מעריך את איכות המודל שלנו מבחינת overfitting. במשך חמש איטרציות חולק המודל לקבוצת אימון וקבוצת מבחן ונבחן דיוק המודל עבור חלוקה זאת. ממוצע גבוה של בדיקות יכול להעיד על מהימנות המודל, ועל יכולתו לסווג נכונה בסבירות גבוהה גם פריטים שאינם מהמאגר, שכן המודל לא התאים את עצמו יתר על המידה להתנהגות הפריטים במאגר.

## ה.

## ניתוח והשוואת השיטות ניתן לראות כי אף על פי שרגרסיה ליניארית פחות מתאימה לכריית מידע מסוג זה, שכן מספקת ערך רציף, היא תוצאותיה איכותיות יותר מאשר הרגרסיה הלוגיסטית המסווגת קטגוריאלית. התאמה זאת עשיתי באמצעות בחירת הערך השלם הקרוב יותר. אתייחס מעתה לרגרסיה הליניארית בלבד אל מול עץ ההחלטה. שני האלגוריתמים נתנו תוצאות דומות ואיכותיות. הדיוק של שניהם כמעט זהה, 96%~. ההבדל העיקרי בין שני המודלים הוא הprecision אל מול הrecall. שני המודלים טעו במידה שווה, אך במקומות שונים. Precision משמעותו אחוז הtrue positive בעוד שrecall מדבר על true negative. לפי צרכים שונים נעדיף לצמצם את הטעויות במקומות שונים ולכן ניתן לקבוע כי מבחינת הדיוק שני המודלים כמעט זהים בינהם, בחירת אחד על פני האחר תהיה עפ״י העדפתנו לזהות חוסר הדיוק.

## 

# 

## 