Data Mining ASSIGNMENT 22

Eli Ackerman

השתמשתי בדאטה סט של מטלה 21, עם כל ה data preprocessing.

1. למידה בייסיאנית ולמידה מבוססת תצפיות

א. נבחר באלגוריתם *Naïve – Bayes*.

תיאור האלגוריתם:

האלגוריתם Naïve bayes הוא מסווג הסתברותי המבוסס על משפט בייס, מסווג זה מניח שתכונות המחלקה הן בלתי תלויות, במילים פשוטות, מסווג Naive Bayes מניח שהנוכחות של תכונה מסוימת במחלקה אינה קשורה לנוכחות של תכונה אחרת כלשהי.
גם אם תכונות אלה תלויות זו בזו או בקיומן של התכונות האחרות, מסווג בייס נאיבי מחשיב את כל המאפיינים הללו כתורמים באופן בלתי תלוי ועצמאי להסתברות.

האלגוריתם פועל באופן הבא:
1. המר את סט הנתונים ל – frequency table.
2. צור טבלת סבירות (likelihood table) ע"י מציאת ההסתברויות של תכונות נתונות.
3. כעת השתמש במשפט בייס כדי לחשב את ההסתברות ה – posterior (הסתברות posterior – הסתברות בדיעבד, לא ידועה מראש)

מתמטית משפט בייס מוצג בצורה הבאה:

$$P(A|B) = [P(B|A) * P(A)] / P(B)$$

כאשר,
P(A|B) היא ההסתברות ה – posterior של המנבא (תכונה) הנתון למחלקה (יעד).
P(A) היא ההסתברות ה – prior (הסתברות prior – הסתברות ידועה מראש) של מחלקה.
P(B) היא ההסתברות ה – prior של מנבא.
P(B|A) היא הסבירות (likelihood) שהיא ההסתברות למנבא נתון מחלקה.

ניתוח האלגוריתם Naïve Bayes תוך כדי התייחסות ל – יתרונות / חסרונות, למה עדיף על Bayesian network במקרה שלנו, והתייחסות לסיבוכיות זמן ריצה:

יתרונות:
1. יעילות - Naïve bayes קל ליישון ויעיל מבחינה חישובית, הוא scales עם הדאטה ומסוגל להתמודד עם כמויות גדולות של דאטה בצורה דיי טובה.
2. ביצועים – למרות הפשטות שלו וההנחה הנאיבית של אי תלות בין תכונות, naïve bayes לרוב בעל ביצועים מפתיעים ונמצא בשימוש לעיתים קרובות כי הוא הביצועים עולים על שיטות סיווג מתוחכמות ומסובכות יותר.
3. טיפול בנתונים קטגוריים – naïve bayes מתאים במיוחד לעבודה עם משתני קלט קטגוריים שאינם תלויים מספרית. עבור המקרה שלנו זאת בחירה טובה כי יש בדאטה סט שלנו תכונות קטגוריות מרובות.

חסרונות:
1. אי תלות בין תכונות – כמובן שזהו החיסרון המובהק של האלגוריתם שכן לרוב זה ממש לא המצב, כאשר ההנחה מופרת בגלל שאינה מתקיימת זה משפיע לרעה על ביצועי האלגוריתם.
2. תדירות אפס – אם למשתנה קטגורי יש קטגוריה ב- test set שלא נצפתה ב – training set אז המודל יקצה הסתברות 0 ולא יוכל לבצע חיזוי, כדי לפתור זאת ישנם כמה דרכים כמו m – estimator או Laplace estimation (מקרה פרטי של m – estimator כאשר m=1)

סיבוכיות זמן ריצה:
למסווג בייס נאיבי יש סיבוכיות זמן ריצה של O(nd + kd) כאשר:
- n הוא מספר הנקודות ב – training set.
- d הוא מספר התכונות של כל נקודה.
- k הוא מספר המחלקות.

סיבות לכך שבמקרה שלנו בחרתי ב- Naïve Bayes על פני Bayesian Network:

1. פשטות ויעילות – כפי שהוזכר למעלה, naïve bayes הוא פשוט ויעיל, לא דורש חישובים מורכבים או כמות גדולה של נתונים כדי לתפקד כמו שצריך, הדאטה סט שלנו ממש ממש קטן יחסית לבעיות "בעולם האמיתי" ולכן הוא יתמודד ביעילות עם זה.
2. תכונות קטגוריות – צוין למעלה למה Naïve bayes מעולה לתכונות קטגוריות ולנו יש הרבה כאלה.
3. בעיית סיווג בינארי – הבעיה שלנו היא בעיית סיווג בינארי (ckd vs notckd), naïve bayes עובד טוב במיוחד עבור בעיות סיווג שכאלה.
4. מורכבות הרשת – Bayesian networks מסובכות הרבה יותר מnaïve bayes .
5. סיבוכיות הרשת - Bayesian networks בדר"כ יותר יקרות לחישוב מאשר naïve bayes, החישוב גודל בצורה אקספוננציאלית עם מספר ההורים שעשויים להיות לצומת ברשת.
6. דרישת נתונים – Bayesian networks דורשות בדר"כ הרבה יותר נקודות כדי ליצור מודלק מדויק, במיוחד אם יש הרבה תכונות.

ב. אלגוריתם k-NN (k-Nearest Neighbors)

<u>תיאור האלגוריתם:</u>

אלגוריתם k-NN הוא סוג של אלגוריתם למידה מבוסס תצפיות, בשימוש נרחב לסיווג ורגרסיה. כאן, במקרה שלנו, אנו מתמקדים בשימוש בו בסיווג.
העיקרון מאחורי k-NN הוא די פשוט: הוא מניח שסביר להניח שלתצפיות דומות תהיה אותה מחלקה.

השלבים הכרוכים בשימוש באלגוריתם k-NN לסיווג הם כדלקמן:

- feature scaling: מכיוון ש k-NN מחשב את המרחק בין נקודות נתונים שונות, חיוני לשנות את קנה המידה של התכונות כך שכולן יתרמו באופן שווה לחישוב המרחק.

- בחר את המספר k של השכנים: המספר k מתייחס למספר השכנים הקרובים ביותר שיש לקחת בחשבון בעת ביצוע חיזוי. זה בדרך כלל מספר אי-זוגי כאשר הבעיה היא סיווג בינארי.

- עבור כל תצפית שלא נראתה:
חשב את המרחק מתצפית זאת לכל שאר התצפיות בtraining set (ניתן לחשב מרחק בדרכים שונות, אך המרחק האוקלידי הוא המרחק הנפוץ ביותר)
מיין את המרחקים בסדר עולה ובחר את k התצפיות הקרובות ביותר.
מחלקת הרוב בין k תצפיות אלה היא המחלקה החזויה עבור התצפית הבלתי נראת.

<u>ניתוח האלגוריתם:</u>

יתרונות:
1. קל להבנה ויישום – אלגוריתם k-NN הוא פשוט ואינטואיטיבי
2. אין הנחות לגבי הנתונים – k-NN אינו מניח הנחות לגבי התפלגות או מבנה הנתונים, מה שהופך אותו לבחירה טובה עבור בעיות שבהן הנחות כאלה קשה להניח או לאמת.

חסרונות:
1. יקר מבחינה חישובית – k-NN יכול להיות יקר מבחינה חישובית, במיוחד עבור דאטה סטים גדולים, זאת מכיוון שהוא דורש חישוב של המרחק של test sample נתונה לכל training samples.
2. רגיש לתכונות לא רלוונטיות או מיותרות – מכיוון שהאלגוריתם משתמש במרחק בין תצפיות לביצוע חיזויים, הוא יכול להיות רגיש לתכונות לא רלוונטיות או מיותרות שעלולות לעוות את מדידת המרחק.
3. בחירה של k – בחירת k מתאים יכולה להיות מאתגרת, k קטן עלול להוביל למודל שרגיש לרעש, בעוד ש – k גדול עשוי לכלול תצפיות ממחלקות אחרות.

סיבוכיות זמן ריצה:
במונחים של סיבוכיות זמן ריצה, תהליך האימון עבור k-NN הוא כמעט אפסי מכיוון שכל החישוב נדחה עד לחיזוי. עם זאת, שלב החיזוי יכול להיות יקר מבחינה חישובית, במיוחד עבור דאטה סטים גדולים, זאת מכיוון שהוא כרוך בחישוב

המרחק מה – test sample לכל ה – training samples ב – training set. לפיכך, זמן הריצה הוא O(Nd) כאשר N הוא מספר התצפיות ב training set ו- d הוא מספר הממדים (תכונות).

איך אלגוריתם k-NN מתאים לבעיה שלנו:

בהתחשב בבעיה שלנו – סיווג חולה חדש ל – ckd או notckd, k-NN יכולה להיות גישה מעשית מכיוון:

1. סיווג בינארי - k-NN עובד היטב עם בעיות סיווג בינארי.
2.קל ליישום – k-NN היא נקודת התחלה טובה מכיוון שהיא קלה ליישום והבנה.
3. קנה המידה של תכונות – מכיוון שהדאטה סט שלנו מכיל תכונות בקנה מידה שונה (למשל גיל, לחץ דם וכו'), feature scaling יהיה שלב חשוב בהכנת הנתונים בשביל שימוש ב- k-NN.

ג. בחירת גישה אחת מבין k-NN או Naïve Bayes

נבחר ב – Naïve Bays מהסיבות הבאות:

1. יעילות חישובית - Naïve bayes יעיל מבחינה חישובית, אין בו צורך לחשב מרחקים כמו k-NN, מה שהופך אותו למהיר יותר משמעותית, יעילות זו עושיה להיות חיונית במיוחד אם נצטרך להפעיל את המודל בלעתים קרובות וכן אם הדאטה סט יגדל בעתיד (כנראה שלא אבל זאת גישה יותר נכונה למרות זאת)
2. פחות רגישות לתכונות לא רלוונטיות - בעוד שגם k-NN וגם Naive Bayes יכולים להיות מושפעים מתכונות לא רלוונטיות או מיותרות, Naive Bayes, באופן כללי, מטפל בהן טוב יותר. ב k-NN תכונות לא רלוונטיות עלולות לעוות את מדידת המרחק באופן משמעותי, מה שעלול להוביל לסיווג שגוי. בדאטה סט שלנו, בזמן שעבדתי מראש וניקינו את הנתונים, עדיין עשויות להיות תכונות שהן פחות אינפורמטיביות עבור משימת הסיווג, וסביר להניח naïve bayes יטפל בהן טוב יותר.
3. פלט הסתברותי – naïve bayes מספק לא רק את תוצאת הסיווג אלא גם את ההסתברויות הposterior של תוצאה זו. זהו מידע שימושי שיכול לספק יותר הקשר, למשל, במצבים שבהם נרצה להבין את הביטחון של המודל בתחזיותיו.
4. ביצועים – naïve bayes נוטה לבצע היטב במצבים רבים בעולם האמיתי, כולל סיווג טקסט ותרחישי אבחון רפואי, הדומים לבעיה שלנו.

בעולם האמיתי הייתי מבצע cross validation לבחירת המודל ושוקל עוד מודלים.

ד. הרצת Naïve Bayes ודיווח התוצאות
התוצאות שהתקבלו לאחר הרצת המודל:

```
Naive-Bayes Confusion matrix:
 [[28 0]
 [ 1 51]]

Naive-Bayes  classification report:
precision    recall  f1-score   support

28       0.98      1.00      0.97       0
52       0.99      0.98      1.00       1

accuracy                           0.99      80
macro avg       0.98      0.99      0.99      80
weighted avg       0.99      0.99      0.99      80
```

ה. ניתוח תוצאות והסקת מסקנות

בהתבסס על התוצאות של Naive Bayes על הדאטה סט של מחלת כליות כרונית (CKD) אנו יכולים לבצע את הניתוח הבא ולהסיק מספר מסקנות:

1. **<u>ביצועי מודל:</u>** הסיווג Naive Bayes מפגין ביצועים מצוינים בדאטה סט הנתון, כפי שמעיד ציון הדיוק הגבוה של 0.99 ב test set. הוא סיווג בצורה נכונה כמעט את כל המקרים משתי המחלקות, והוכיח את יכולת הניבוי החזק שלו.
2. **<u>דיוק, ריקול וציון F1 :</u>** עבור מחלקה 0 (ללא CKD) דיוק, ריקול וציון F1 כולם מושלמים או כמעט מושלמים. באופן דומה, עבור מחלקה 1 (CKD) הדיוק מושלם והריקול כמעט מושלם, ומניב ציון F1 הקרוב ל-1. מדדים אלו מצביעים על כך שהמודל לא רק מדויק בתחזיות שלו, אלא גם בעל ציון גבוה רגישות לאיתור מקרים חיוביים.
3. **<u>איזון בסיווג:</u>** הציונים הגבוהים הן עבור דיוק והן עבור זכירה מצביעים על כך שהמודל מאוזן היטב ואינו מעדיף אף אחת מהמחלקות על פני האחרות. זה לא נוטה לניבוי יתר או חיזוי חסר של אף אחד מהמחלקות, וזה חיוני בתרחיש רפואי שכן הן שליליות שגויות (חולה עם CKD מתפספס) והן חיוביות שגויות (חולה בריא מאובחן בטעות עם CKD) עלולות להיות השלכות חמורות.
4. **<u>הכללה</u>**: בהתחשב בעובדה שהמודל מתפקד היטב בtest set, סביר לצפות ממנו להכליל היטב לנתונים חדשים שלא נראו. עם זאת, חשוב לציין שיש לנתר את האפקטיביות של המודל באופן מתמיד כאשר הוא נפרס על נתונים מהעולם האמיתי.
5. **<u>פרשנות:</u>** למרות הביצועים החזקים שלו, מגבלה אחת של Naive Bayes היא שהוא עשוי להיות פחות בר פרשנות בהשוואה לכמה מודלים אחרים. ההנחה של אי תלות בין תכונות עשויה להקשות על פירוש היחסים בין תכונות שונות לבין התוצאה החזויה.

לסיכום, Naive Bayes מציע פתרון עוצמתי ויעיל לבעיית חיזוי CKD. בהתחשב בביצועיו החזקים, הוא עשוי לשמש כדי לסייע באבחון מוקדם של המחלה, ובכך לאפשר טיפול בזמן .עם זאת, למרות התוצאות המבטיחות הללו, חשוב לאמת את ביצועי המודל כפי שהוא בעולם האמיתי.

<u>2. ניתוח אשכולות</u>

א. מדדי איכות לאשכולות

חלק מהמדדים שבדר"כ משתמשים בהם:

Silhouette Coefficient – זה ייתן מדד איזון למידת ההפרדה של האשכולות (הפרדה) ועד כמה כל אשכול קומפקטי (לכידות). זהו מדד נפוץ עבור משימות שבהן המספר האמיתי של אשכולות אינו ידוע או מתעלם מהן במהלך אשכול.

Homogeneity, Completeness, and V-Measure – אלו הם שלושה מדדים קשורים שמעריכים את איכות הקיבוץ על ידי השוואתו לאמת ידועה (קבוצות הסיווג). הומוגניות בודקת שכל אשכול מכיל רק חברים ממחלקה אחת. השלמות בודקת שכל החברים במחלקה נתונה מוקצים לאותו אשכול. V-Measure הוא הממוצע ההרמוני של הומוגניות ושלמות. אמצעים אלה יהיו שימושיים כדי לראות אם האשכולות המתרחשים באופן טבעי מתאימים לקבוצות הסיווג.

ננסה להשתמש בניתוח אשכולות בכדי לקבל תובנות שיעזרו לנו למשימת הסיווג, לכן בהרצת האלגוריתם נתעלם מתכונת הסיווג (שכן מדובר ב unsupervised learning method), נגדיר מספר אשכולות כמספר הערכים של תכונת הסיווג, ונשווה בין תכונת הסיווג לחלוקת האשכולות שתתקבל.

ב. בחירת גישה לניתוח האשכולות

נבחר בגישה פופולרית בגלל הפשטות והיעילות שבה – k-means.

<u>תיאור הגישה:</u>

K-Means הוא אלגוריתם חלוקה, המנסה לחלק את הנתונים למספר נתון של אשכולות (K), בהתאם לפונקציית מרחק (דמיון) המגדירה את המכנה המשותף המבוקש בין איברים באותו אשכול.

כל נקודת נתונים שייכת לאשכול עם הממוצע או המרכז הקרוב ביותר.

הרעיון המרכזי הוא כדלקמן:

1. **אתחול** – התחל בהצבת K centroids באופן אקראי במרחב של נתוני הקלט שלנו.
2. **שלב ההקצאה** - כל נקודת נתונים מוקצית למרכז הקרוב ביותר שלה, בהתבסס על מדידת מרחק (בדרך כלל מרחק אוקלידי). כתוצאה מכך נוצרים אשכולות, שכל אחד מהם מרוכז סביב ה-K centroids
3. **שלב עדכון** - ה-centroids מחושבים מחדש כממוצע של כל נקודות הנתונים שהוקצו לאותו אשכול בשלב ההקצאה.
4. **חזור על שלבים 2 ו-3** - שלבי ההקצאה והעדכון חוזרים על עצמם באופן איטרטיבי עד להתכנסות, שזה בדרך כלל כאשר ההקצאות כבר לא משתנות או שהשינוי נמצא מתחת לסף מוגדר.

נימוק לבחירת K-Means :

1. מספר אשכולות - אחד הפרמטרים העיקרים עבור K-Means הוא מספר האשכולות, k. במקרה שלנו, אנחנו יודעים שנרצה לחלק את הנתונים לשני אשכולות (ckd ו – notckd), כך שהידע הזה מתיישב היטב עם הדרישות של האלגוריתם.
2. יעילות - K-Means היא גישה פשוטה ומתאימה במיוחד למצבים שבהם יש מספר רב של תכונות. יעיל מבחינה חישובית בהשוואה לשיטות אשכולות אחרות, בנוסף מועיל עובדים עם דאטה סט גדול יותר.
3. פרשנות - הפלט של K-Means הוא פשוט יחסית לפירוש, מה שיכול להיות יתרון משמעותי כאשר מנסים להפיק תובנות מממצאי המודל.

ג. שלבי ניתוח האשכולות

שלב 1: עיבוד נתונים

ראשית, נציין שבניגוד לנתונים הסופיים בממן 21 בהם התכונות הנומריות עברו דיסקרטיזציה, נלך שלב אחד אחורה לפני שבוצעה דיסקרטיזציה בכדי לא להשפיע לרעה על חישוב המרחקים.
כעת, נבצע נורמליזציה לכל אחד מהערכים הנומריים לתחום [0,1] בכדי שחישוב המרחקים ייעשה באופן שווה לכל התכונות.

שלב 2: התעלמות מתכונת הסיווג

על מנת למנוע הטיה, יש להתעלם זמנית מתכונת הסיווג במהלך ניתוח האשכול .הסיבה לכך היא שהמטרה שלנו עם אשכולות היא לגלות דפוסים נסתרים בנתונים, שבאופן אידיאלי לא אמורים להיות מושפעים מתוויות היעד המוגדרות מראש שלנו.

שלב 3: החלטה על מספר האשכולות

מספר האשכולות (K) הוא פרמטר מפתח עבור אלגוריתם K-Means . עבור משימה זו, מכיוון שאנו מחפשים תובנות שיכולות לסייע במשימת הסיווג, ייתכן שיהיה רעיון טוב להגדיר את K להיות שווה למספר הערכים הייחודיים בתכונת הסיווג. זה עשוי לאפשר לנו להשוות את האשכולות שנוצרו עם תכונת הסיווג. לפיכך נגדיר K=2.

שלב 4: K-Means Clustering (main step)

זהו השלב העיקרי שבו אלגוריתם K-Means מופעל על הדאטה סט. השלבים המעורבים הם:

- אתחל באופן אקראי K centroids.
- הקצה כל נקודת נתונים לcentroid הקרוב ביותר, היווצרות של K אשכולות.
- חשב מחדש את ה centroid של כל אשכול בהתבסס על האיברים הנוכחיים (נקודות הנתונים) של האשכול.
- חזור על שני השלבים שלעיל עד שהמרכזים לא זזים יותר באופן משמעותי, או שהושג מספר מסוים של איטרציות

שלב 5: פירוש של האשכולות

לאחר הפעלת אלגוריתם K-Means, האשכולות המתקבלים מתפרשים. זה נעשה על ידי ניתוח המאפיינים של נקודות הנתונים בתוך כל אשכול והבנת מה הן עשויות לייצג. שלב זה יכול לתת תובנות חשובות לגבי הנתונים.

שלב 6: השוואת האשכולות עם תכונת הסיווג האשכולות שהתקבלו מושווים למחלקות בפועל שניתנו על ידי תכונת הסיווג. השוואה זו עשויה לחשוף כמה דפוסים או תובנות מעניינות שיכולות להיות שימושיות עבור משימת הסיווג. ניתן לעשות זאת על ידי בדיקת התפלגות המחלקות בתוך כל אשכול.

שלב 7: הערכה

ניתן להעריך את האשכולות באמצעות המדדים שהוגדרו בסעיף קודם (מרחק בין אשכולות, מרחק בין נקודות בתוך אותו אשכול, ציון צללית).

פרמטרים:

מלבד מספר האשכולות K, אלגוריתם K-Means בפועל עשוי לכלול פרמטרים נוספים, כמו המספר המרבי של איטרציות שיש להפעיל, השיטה לאתחול הcentroids, מדד המרחק שיש להשתמש בו (בדרך כלל מרחק אוקלידי), וכן את הtolerance לשקול אם הcentroids זזו באופן משמעותי.

ערכי הפרמטרים:

הערך של K יוגדר כמתואר לעיל(K=2, אני בדקתי עם ערכי K אחרים K=3,4,…,15 ולכולם היו ביצועים גרועים יותר). ניתן להגדיר את המספר המרבי של איטרציות לערך ברירת מחדל כמו 300, אבל זה עשוי להיות מוגדל אם האלגוריתם לא מתכנס. הcentroids הראשוניים נבחרים בדרך כלל באקראי מתוך הדאטה סט, אם כי ישנן שיטות מתוחכמות יותר כמו K-means++ . מדד המרחק יהיה בדרך כלל המרחק האוקלידי, וניתן להשתמש ב tolerance קטן כמו 0.0001 כדי להחליט אם הcentroids זזו באופן משמעותי.

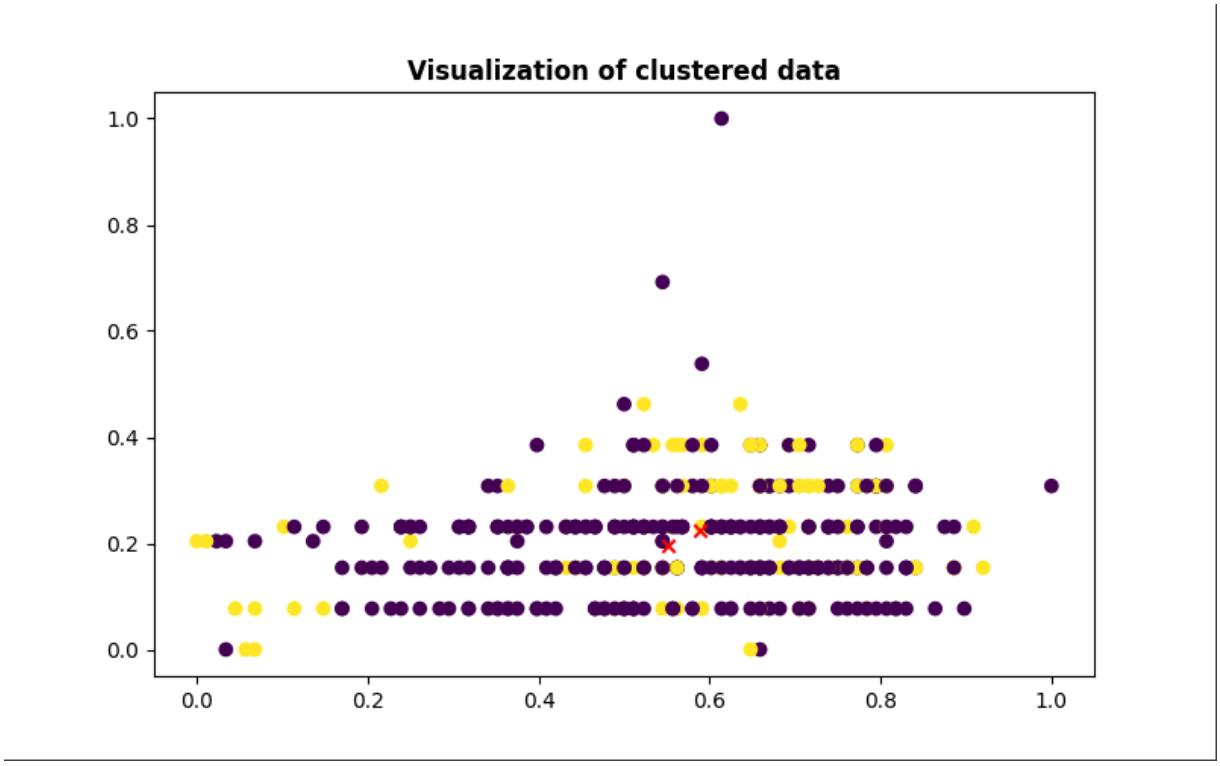
ד. ריצה ודיווח תוצאות

התוצאות שהתקבלו לאחר הרצה:

```
==== Clustering regular evaluation ====
For n_clusters = 2, average intra-cluster distance is : 1.1183606076279105
For n_clusters = 2, minimum inter-cluster distance is : 2.792443038197775
For n_clusters = 2, the average silhouette_score is : 0.5144471395704442

==== Clustering supervised evaluation ====
Accuracy: 0.6525
:Confusion matrix
[0   150]]
[[111 139]
```

ויזואליזציה:



ה. ניתוח תוצאות ומסקנות

האלגוריתם K-Means שימש למשימתנו לניתוח אשכולות על הדאטה סט של מחלת כליות כרונית, כאשר מספר האשכולות, K, מוגדר ל – 2 לאחר בדיקת כמה וכמה ערכי K שהביאו תוצאות גרועות יותר.
התוצאות הבאות התקבלו:

<u>הערכה רגילה (שימוש במדדי אשכולות):</u>

- המרחק הממוצע של נקודות בתוך אשכול עבור כל אשכול יצא סביב 1.118, מדד זה משקף את המרחק הממוצע בין כל נקודת נתונים למרכז האשכול שהוקצה לה, ככל שהערך הזה קטן יותר, כך קיבוץ האשכולות הדוק יותר, היות וקיבלנו יחסית ערך נמוך, זה מעיד על כך שנקודות הנתונים בכל אשכול קרובים למדי למרכזים המתאימים להם.
- המרחק המינימלי בין אשכולות היה כ – 2.792, מדד זה מודד את המרחק בין אשכולות שונים, כאשר ערך גדול יותר מעיד על הפרדת אשכולות טובה יותר, במקרה שלנו הערך מעיד על מידה סבירה של הפרדה בין שני האשכולות.
- ציון הצללית הממוצע היה סביב 0.514, ציון הצללית נע בין מינוס 1 ל – 1, כאשר ערך שקרוב יותר ל1 מצביע על כך שנקודת הנתונים מותאמת היטב לאשכול שלה ומתאימה בצורה גרועה לאשכולות אחרים. היות וקיבלנו ציון מעל 0.5 בממוצע, זה מצביע על כך שבממוצע, נקודות הנתונים היו מקובצות כראוי.

<u>הערכה מפוקחת (שימוש במדדי דיוק):</u>

- הדיוק של הקלאסטרינג בהשוואה לתוויות האמיתיות היה 65.25%, המשמעות היא ש – 65.25% מסך כל המופעים סווגו כהלכה כ- ckd או notckd.
- ה – confusion matrix מראה ש – 150 מופעים סווגו בצורה נכונה כ – ckd (true positives) ו – 111 מופעים סווגו בצורה נכונה כ – notckd (true negatives). עם זאת, היו 139 מופעים שסווגו בצורה שגויה, הם היו ckd אך סווגו כ – notckd, אין מופעי notckd שסווגו בטעות כ – ckd.

<u>מסקנות:</u>

גישת ה -K-means clustering הצליחה להפריד באופן סביר בין חולי CKD לבין מטופלים שאינם CKD עם ציון צללית ממוצע טוב למדי ומרחק בין אשכולות.

הדיוק של המודל, בהשוואה לתוויות בפועל, היה 65.25%. למרות שזה לא דיוק גבוה במיוחד, זה משמעותי בהתחשב בכך שמדובר באלגוריתם למידה לא מפוקח שלא הייתה לו גישה לתוויות היעד בשלב ההדרכה.

עם זאת, המספר הגבוה של false negatives יכול להיות מדאיג, שכן זה אומר שחולי CKD רבים סווגו באופן שגוי כלא CKD .
זה מצביע על כך שאולי יש מקום לשיפור המודל, אולי על ידי ניסיון אלגוריתמים שונים של אשכולות, כוונון hyperparameters או שימוש בתכונות שונות.
לבסוף, העובדה ש K=2 הביאה את המדדים הטובים ביותר לאחר בדיקת מספר ערכי K (עד 15) תומכת בשימוש בשני אשכולות. זה מתיישב עם הציפיות שלנו, בהתחשב בעובדה שיש שני סוגים של חולים 'ckd' ו 'notckd'.

לסיכום, בעוד שגישת האשכולות K-means השיגה הצלחה מתונה, ניתן לבצע שיפורים נוספים כדי להגביר את הדיוק ולהקטין את מספר הסיווגים השגויים. התוצאות בכל זאת מדגימות את הפוטנציאל של שיטות למידת מכונה ללא פיקוח כמו K-means כדי לסייע בזיהוי מצבי מחלה, גם כאשר נתונים מסומנים עשויים שלא להיות זמינים בקלות.

3. רשת נוירונים מלאכותית

א. ארכיטקטורת הרשת

בניתוח זה, ניישם feedforward neural network, המכונה גם Multilayer Perceptron (MLP) עבור בעיית סיווג בינארי.

- **שכבת הקלט:** בשכבה זו יהיו 17 נוירונים כמספר התכונות, לא כולל משתנה המטרה (כזכור, ביצעתי feature selection במטלה הקודמת וכתוצאה מכך נמחקו כמה מהתכונות, בדאטה סט המקורי היו 24 תכונות לא כולל משתנה המטרה)

- **שכבות נסתרות:** אני בוחר להשתמש בשכבה נסתרת אחת בלבד, מהסיבה שאינני רוצה שתהיה התאמת יתר ושכבה אחת בדר"כ מספיקה למגוון רב של משימות, במיוחד כשהמשימה לא כל כך מסובכת, כמו במקרה שלנו. ככלל אצבע לא פורמלי (נלקח מפה https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw) נבחר ב – 9 נוירונים (הממוצע של מספר הנוירונים בשכבת הקלט והפלט שזה יוצא 18=1+17 חלקי 2), כמובן יהיה עוד נוירון של bias אבל לא צריך להזכיר אותו בהגדרת הארכיטקטורה. פונקציית ההפעלה בשכבה הנסתרת תהיה ReLU, הפונקציה הזו מציגה אי-ליניאריות למודל, מה שמאפשר למודל ללמוד קשרים מורכבים בין משתנים, יתר על כך, היא יעילה מבחינה חישובית ומקלה על בעיות שונות. היא מוגדרת בצורה הבאה : $f(x) = \max(0,x)$.

- **שכבת הפלט:** שכבה זו תהיה בעלת נוירון בודד, מכיוון שהמשימה שלנו היא בעיית סיווג בינארי (ckd או notckd). פונקציית ההפעלה עבור שכבה זו תהיה הפונקציה Sigmoid, הפונקציה הזו מתאימה כאן מכיוון שהיא "מועכת" את הפלט של הנוירון לערך בין 0 ל – 1, שאותו ניתן לפרש כהסתברות של המחלקה החיובית (במקרה שלנו ההסתברות ל – ckd), בחירה נפוצה לבעיות סיווג בינארי כמו שלנו.

- **קשרים וזרימת מידע:** הנוירונים יהיו מחוברים באופן מלא בין השכבות, המשמעות היא שכל נוירון בשכבה מקבל קלט מכל הנוירונים של השכבה הקודמת ושולח את הפלט שלו לכל הנוירונים של השכבה הבאה, המידע בסוג זה של רשת, המכונה feedforward NN, זורם משכבת הקלט, דרך השכבה / שכבות הנסתרות, אל שכבת הפלט, מבלי לחזור אחורה.

ב. פרמטרים של תהליך האופטימיזציה

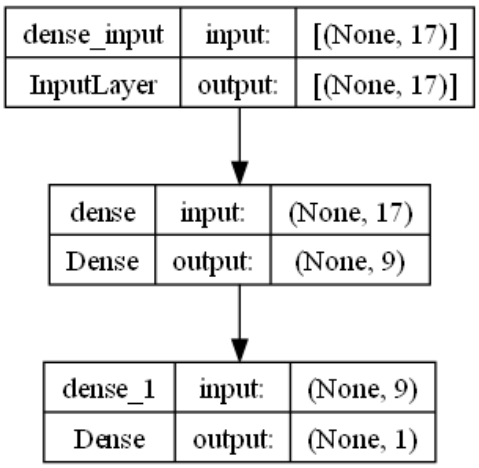
- **פונקציית שגיאה (פונקציית הפסד):** פונקציית השגיאה מכמתת עד כמה תחזיות המודל שלנו תואמות את הערכים בפועל. מכיוון שאנחנו מתמודדים עם בעיית סיווג בינארי (ckd או notckd), פונקציית ההפסד המתאימה כאן תהיה "Binary Cross-Entropy Loss", cross entropy loss או log loss מודד את הביצועים של מודל סיווג שהפלט שלו היא הסתברות בין 0 ל – 1. הפונקציה הזאת אידיאלית לביות סיווג בינארי.

- **Batch size:** גודל batch מתייחס למספר דוגמאות האימון (training examples) המשמשות באיטרציה אחת של אימון מודל (מעבר אחד קדימה ואחורה). גדלי batch קטנים יותר רועשים יותר ומציעים regularizing effect ושגיאת הכללה נמוכה יותר (generalization error), בעוד שגדלי batch גדולים יותר מתכנסים מהר יותר ומציעים יעילות חישובית. מכיוון שהדאטה סט שלנו מכיל 400 דוגמאות שתי אופציות טובות לגודל יהיו 32 או 64, נלך על 32, זה גודל מאוזן, קטן בשביל לא לגרום לבעיות זיכרון וללמידה מהירה וגדול מספיק להשערה סבירה של הגרדיאנט.

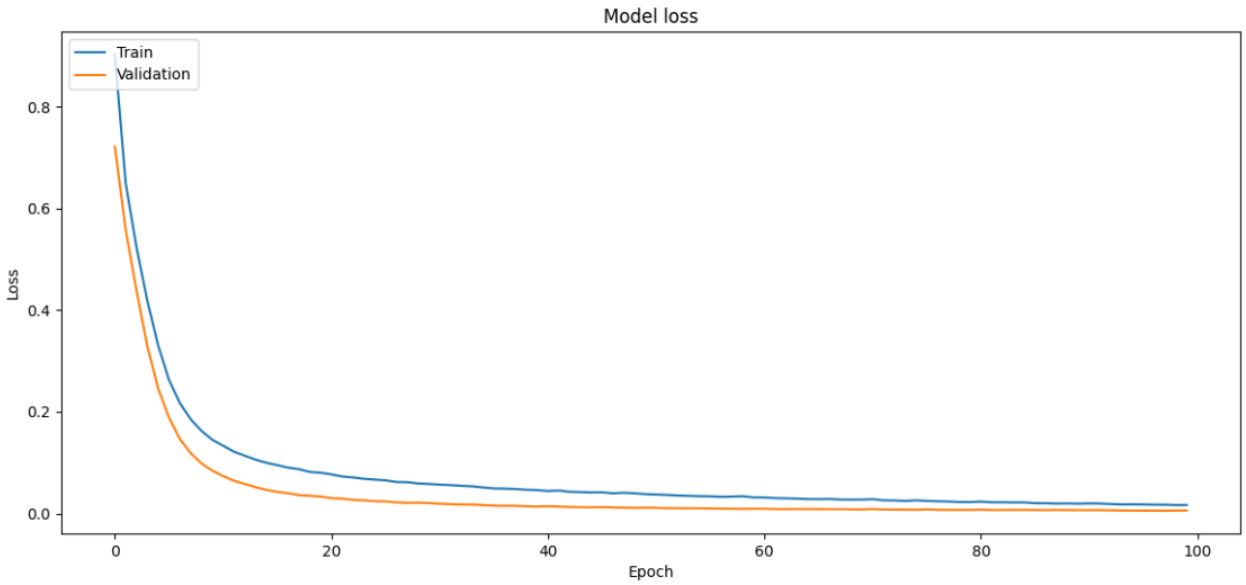
- **קצב למידה (learning rate):** קצב הלמידה קובע כמה להתאים את המודל בתגובה לשגיאה המשוערת בכל פעם שהמשקלים של המודל מתעדכנים. בחירת קצב הלמידה היא אולי המאתגרת ביותר שכן ערך קטן מדי עלול לגרום לתהליך אימון ארוך שעלול להיתקע ולא להתכנס, בעוד שערך גדול מדי עלול לגרום ללמידה של סט משקולות לא אופטימלי מהר מדי או תהליך אימון לא יציב. אני בחרתי בקצב למידה מתון, 0.01, זהו ערך דיפולטיבי נפוץ כשאין מידע מוקדם לגבי איזה קצת למידה לבחור.

ג. ריצה והערכת ביצועי הרשת לאורך ה Epochs של האימון

פלוט של הארכיטקטורה:



גרף השגיאה עבור נתוני האימון ועבור נתוני המבחן:



תוצאות:

Test accuracy 0.987500011920929

Confusion Matrix:
[[27 1]
 [0 52]]

Model: “sequential”

```
_________________________________________________________________
# Layer (type)                 Output Shape              Param
=================================================================
dense (Dense)                (None, 9)                 162

dense_1 (Dense)              (None, 1)                 10

=================================================================
Total params: 172
Trainable params: 172
Non-trainable params: 0
_________________________________________________________________
None
```

ד. מקרים חריגים בהם בוצע סיווג שגוי
המספרים הם אחרי נורמליזציה!
סווג כ- 1 (ckd) בזמן שהוא 0 (notckd)

```
Instance index 1 was misclassified
age      0.511
bp       0.231
sg       1.020
al       0.000
rbc      0.000
pc       0.000
pcc      0.000
ba       0.000
bu       0.081
pot      0.045
hemo     0.694
pcv      0.956
wc       0.244
cad      0.000
appet    1.000
pe       0.000
ane      0.000
Name: 280, dtype: float64
True label = 0, Predicted label = 1
```

ה. ניתוח תוצאות ומסקנות

התוצאות מצביעות על כך שהמודל מתפקד טוב מאוד, הניתוחים של כמה מההיבטים המכריעים:

- דיוק: הדיוק של הtraining וגם ה validation היו מאוד גבוהים, דיוק הtraining היה כמעט 100% (99%) (לפי הepochs בepoch האחרון) , דיוק ה validation הגיע ל100% (בepoch האחרון), וכן 98% על הtest. זה מצביע על כך שהמודל למד לבצע תחזיות נכונות על נתונים שנראו ובלתי נראים כאחד.
- התאמת יתר: התאמת יתר היא תופעה שבה המודל מתפקד טוב על נתוני האימון וגרוע על הvalidation, אצלנו נראה שאין בעיה כזאת, דיוק הvalidation עולה על נתוני האימון, מה שמצביע על כך שהמודל מכליל היטב לנתונים בלתי נראים.

לסיכום, נראה שמודל מותאם היטב ומספק תוצאות מעולות, עם זאת להמשך, כדאי באמת לנסות אותו בעולם האמיתי כדי לראות את הביצועים בעולם האמיתי, בהתאם לכך ישנה אפשרות לשנות את הארכיטקטורה / קצב הלמידה וכו'.

4. סיכום ומסקנות

בממ"ן 21 הגדרנו את הבעיה ואת המטרה העיקרית שלנו – בעיית סיווג בינארי, האם למטופל יש או אין מחלת כליות כרונית. ביצענו תהליך של עיבוד נתונים שכלל התעסקות עם הנתונים, הגדרתם, חקירתם, ניקוי של הנתונים, טרנספורמציה של הנתונים ועוד. סקרנו מודלים שונים והחלטנו להשתמש ב2 מודלים שמבוססים על עץ, Random forest ו CART, שני המודלים השיגו דיוק גבוה במיוחד, מעל 95%, מה שאומר שהם מודלים מצוינים לבעיה שלנו וכדאי מאוד להשתמש בהם בעולם האמיתי כדי להבין את ביצועם שם.

בממ"ן 22 המשכנו עם אותה הבעיה, עבדתי עם אותו בסיס נתונים שעבר עיבוד נתונים (כשהגענו למודלים מסוימים כמו k-means חזרנו קצת אחורה ללפני הדיסקרטיזציה וביצענו נורמליזציה) , התחלנו עם סיווג ביסיאני, שם נבחר האלגוריתם Naïve Bayes שהשיג תוצאות מעולות, כמעט מושלמות, בין היתר גם תיארנו וחקרנו את אלגוריתם k-NN , בהמשך עברנו לאלגוריתם מהתחום של למידה לא מפוקחת (unsupervised learning) שנקרא K-MEANS, למען ניתוח אשכולות, היות והדאטה שלנו הוא לייבלד, ומדובר באלגוריתם שלא לומד את הדאטה, התוצאות לא היו כמו שאר המודלים והיו קצת בינוניות. לבסוף התעסקנו עם המודל אולי החזק ביותר, רשת נוירונים, השתמשנו בגרסה הבסיסית ביותר, feedforward neural network, גם המודל הזה השיג ביצועים פנומנליים, מעל 98%, מה שמעיד על כך שכדאי מאוד להשתמש בו בעולם האמיתי.

לסיכום, עברנו על הרבה מודלים וחקרנו אותם, איך הם עובדים, מה הם השיגו והאם כדאי להשתמש בהם לבעיה שלנו, רוב המודלים השיגו תוצאות יוצאות דופן, עם דיוק state-of-the-art, ניתן, אם נשתמש בהם בעולם האמיתי, לגלות האם מטופל הוא חולה במחלת כליות כרונית בדיוק של מעל 98%, וזאת המסקנה החשובה ביותר בפרויקט הזה

הערה אחרונה:
גם בממן 21 וגם בממן 22 כתבתי בשפת התכנות python, יכולתי להשתמש בweka אבל היות וזה התחום בו ארצה לעסוק בעתיד תמיד אני אעדיף לשמור לעצמי פונקציות ותהליכים לעתיד, בפרויקט הזה יש הרבה כאלה. אשמח לשתף את כל הפרויקט (בקוד זה פרויקט אחד ולא מחולק ל2 שונים) והקוד במידת הצורך.

תודה על הקורס, היה מעניין בצורה יוצאת דופן, למדתי המון.