The Academic College of Yezreel Valley
Department of Information Systems

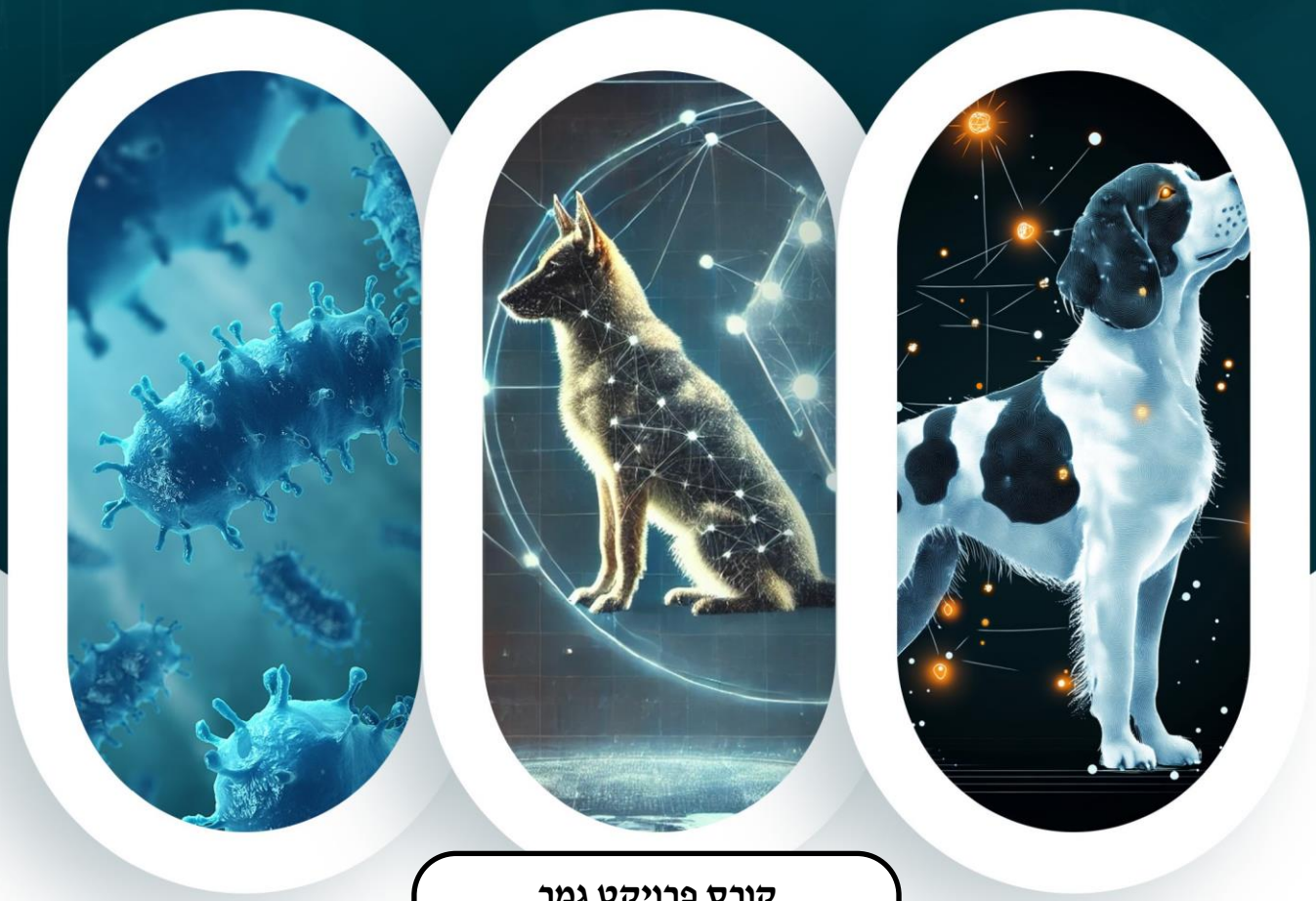
המכללה האקדמית עמק יזרעאל
חוג למערכות מידע

Data modeling report

פרויקט גמר מסמך מידול נתונים

מערכת לחיזוי אזורים ומועדים בסיכון לחשיפה
ולהתפרצות מחלת הכלבת והצגת מפת סיכונים.

סטודנטיות מגישות:
אל קוזלי אימאן, ת.ז 212175582
רם דניאל, ת.ז 208220509



קורס פרויקט גמר
שנת לימודים: תשפ"ה
תאריך הגשה: 10.5.2025
שם המרצה: מר זכאי אבי
שם המנחה: גב' גוטפריד ג'ניה

תוכן עניינים

1 בחירת טכניקות מידול (Selecting Modelling Techniques)

בשלב המידול, אחרי שהכנו את הנתונים ועיבדנו אותם בצורה שתאפשר למודלים לעבוד בצורה אופטימלית, הגיע הזמן לבחור את הטכניקות והמודלים המתאימים ביותר עבור הבעיה שלנו. המטרה כאן היא לחזות שני משתנים , אזור וחודש, במודל אחד. הבחירה במודלים המתאימים נעשתה תוך התחשבות בכמה גורמים מרכזיים.

הבנת סוג הנתונים:

הנתונים שלנו כוללים שני סוגים עיקריים של משתנים: נומריים וקטגוריאליים. המשתנים הנומריים כוללים טמפרטורה, גשם וימים גשומים בעוד שהמשתנים הקטגוריאליים כוללים את הסוג של החיה, היישוב, החודש, האזור וכו'. ישנם מודלים שמבצעים עבודה טובה יותר עם משתנים נומריים, ויש כאלו שמתאימים יותר לעבודה עם משתנים קטגוריאליים. לכן, היה חשוב לבחור מודלים שיכולים להתמודד עם שני סוגי הנתונים הללו בצורה אופטימלית.

הצורך בחיזוי שני משתנים בו זמנית:

המטרה שלנו היא לחזות אזור (Region) וחודש (Month) בו זמנית, מה שמחייב שימוש במודלים שיכולים לבצע סיווג מרובה-מטרות (Multi-output classification) כלומר, המודלים לא צריכים רק להניח את המיקום או את החודש בנפרד, אלא גם להבטיח שהם יכולים להתמודד עם חיזוי שני משתנים בצורה מדויקת בו זמנית. הבחירה במודלים כמו MultiOutputClassifier כעטיפה עבור מודלים כמו Random Forest או XGBoost נבעה מתוך הצורך הזה.

1.1 בחירה בטכניקת המידול המתאימה (Choosing the Right Modelling Techniques)

למרות שייתכן ויש מודלים נוספים שיכולים להתאים, בחרנו בשלב זה לנסות כ10 מודלים מקצועיים שמתאימים ביותר בתיאוריה לפרויקט שלנו כאשר מדובר בבעיות סיווג על נתונים מגוונים ומורכבים כמו שלנו.

המודלים שנבחרו:

- Logistic Regression: מודל פשוט יחסית שמניח קשר לינארי בין המשתנים, מתאים לשימוש על נתונים עם קשרים פשוטים וברורים בין התכונות ליעדים. היינו רוצים לראות אם הוא מצליח להתמודד עם הבעיה למרות הפשטות של המודל.
  - יתרונות: הוא מודל פשוט וקל להבנה. הדיוק שלו טוב כאשר מדובר בבעיות סיווג שבהן המידע נפרס על פני מספר קטגוריות, המודל מהיר באימון ובחיזוי, במיוחד כאשר יש כמות גדולה של נתונים.
  - חסרונות: המודל מוגבל כאשר הקשרים בין המשתנים אינם לינאריים. אם יש קשרים מורכבים יותר או אם הנתונים לא מייצגים קשר לינארי, הדיוק עלול להיפגע. ולפעמים לא מצליח להתכנס ולהגיע לתוצאה המתאימה.

- Decision Tree: מבצע חיזוי על ידי יצירת עץ החלטה. בכל צומת של העץ, המודל שואל שאלה לגבי אחד מהתכונות ומחליט איזה כיוון לקחת, עד שמגיעים להחלטה הסופית. הוא מאפשר לנו לראות בצורה ברורה איך כל תכונה משפיעה על התוצאה הסופית, אך יש לו נטייה לאוברפיטינג כאשר העץ עמוק מדי.

- יתרונות: היתרון המרכזי הוא שהמודל מאוד אינטואיטיבי וקל להבנה. כל עץ החלטה בונה סדרה של שאלות שגורמות למודל להחליט על הקטגוריה הסופית זה מאפשר לראות בצורה ברורה איך כל תכונה משפיעה על תוצאת החיזוי. הוא גם לא דורש הנחות קודמות על הנתונים, כלומר הוא יכול להתמודד עם נתונים שאינם לינאריים, או עם נתונים קטגוריאליים או נומריים. יתרון נוסף הוא שהמודל מצוין בהבנה של תהליכים דינמיים.
- חסרונות: הוא נוטה לעיתים לאוברפיטינג כאשר העץ עמוק מדי, כלומר הוא מתאים את עצמו מאוד לנתונים הספציפיים של האימון ולא כל כך טוב בנתוני בדיקה חדשים. בנוסף, קשה להבין את התמונה הכללית כשהעץ מאוד עמוק, מה שמקשה לפענח את התהליך שהוביל לחיזוי.

❖ Random Forest: כל עץ עובר על קבוצה שונה של נתונים ובסוף המודל מצרף את התוצאות. היתרון כאן הוא שנוצרת גיוון בין העצים, מה שמפחית את הסיכון לאוברפיטינג. כמו כן, הוא מתפקד בצורה טובה כאשר יש הרבה תכונות עם קשרים מורכבים.

- יתרונות: מפחית את הסיכון לאוברפיטינג. הוא עובד בצורה מאוד טובה כאשר יש הרבה תכונות שונות או נתונים לא לינאריים. יתרון נוסף שהוא מצוין להתמודד עם נתונים חסרים ויכול להפיק תוצאות טובות גם כאשר הנתונים לא נפרסים בצורה אידיאלית.
- חסרונות: החיסרון של Random Forest הוא שהוא לא תמיד מהיר במיוחד, במיוחד כאשר יש הרבה עצים ומאפיינים. בנוסף, קשה לפרש את התוצאה הסופית, כי מדובר בקבוצה של עצים ואין דרך פשוטה להבין מה בדיוק קרה מאחורי כל חיזוי.

❖ K-Nearest Neighbours (KNN): מודל זה מבוסס על הקשרים בין הדוגמה החדשה לדוגמאות קיימות. הוא בודק איזה דוגמאות הכי קרובות לדוגמה החדשה ומחזיר את התוצאה לפי הקטגוריות הנפוצות ביותר. מדובר במודל פשוט וקל להבנה, אך יכול להיות איטי בחישוב כאשר יש הרבה נתונים.

- יתרונות: מודל פשוט ונוח, שמבצע חיזוי על פי דמיון בין דוגמאות. הוא מצוין כאשר המידע לא לינארי או כאשר קשה למצוא קשרים ברורים בין המשתנים. המודל גם לא מצריך הרבה זמן לאימון כי הוא מבצע את החיזוי בזמן אמת, כלומר אין צורך בהכנה מראש של המודל.
- חסרונות: החיסרון הגדול שהוא יכול להיות מאוד איטי כאשר יש כמות נתונים גדולה. ככל שיש יותר דוגמאות, כך הזמן לחישוב הדמיון בין הדוגמאות יגדל באופן משמעותי. בנוסף, אם הנתונים רועשים או יש הרבה תכונות לא רלוונטיות, המודל עלול להניב תוצאות פחות טובות.

❖ SVC (Support Vector Classifier): מודל חיזוי שמייצר גבול החלטה אופטימלי בין קטגוריות שונות, אפילו כאשר הנתונים לא לינאריים. מדובר במודל חזק במיוחד שמספק ביצועים טובים מאוד כאשר יש גבול ברור בין הקטגוריות.

- יתרונות: הוא מודל חזק ויציב שמספק חיזוי מדויק, במיוחד כאשר יש גבול ברור בין הקטגוריות. הוא יכול להתמודד עם בעיות סיווג לא לינאריות על ידי שימוש בגרסה מרחבית גבוהה, כלומר הוא מוצא גבול החלטה אופטימלי שמפצל את הנתונים בצורה הטובה ביותר.

- חסרונות: לא תמיד מהיר כשיש הרבה נתונים. מודל כזה דורש זמן חישוב גבוה, במיוחד כשיש כמות גדולה של דוגמאות. בנוסף, הגדרות פרמטרים נכונות דורשות ידע מקצועי על מנת למנוע בעיות כמו אוברפיטינג.

❖ Naive Bayes: מודל זה מבוסס על תיאוריה הסתברותית שמניחה שכל תכונה תורמת בנפרד לסיווג. הוא יעיל במיוחד כאשר הנתונים הם קטגוריאליים ויכול להיות פתרון טוב בעבודה עם טקסטים או נתונים שבהם יש תלות בלתי תלויה בין התכונות.

- יתרונות: הוא מודל הסתברותי מאוד יעיל כאשר הנתונים קטגוריאליים, כמו בעיות סיווג טקסטים, ושונה מהממדים האחרים בכך שהוא מהיר מאוד ומבצע חיזוי כמעט מידי. בנוסף, מודל זה עובד בצורה טובה גם כאשר יש נתונים רועשים.
- חסרונות: הבעיה העיקרית היא ההנחה שהוא עושה על התכונות, שהיא כל תכונה בלתי תלויה. במקרים רבים, זה לא נכון, ובמצבים שבהם יש תלות בין תכונות, המודל עשוי להניב ביצועים פחות טובים.

❖ XGBoost: מודל מבית Gradient Boosting שמבצע תיקונים לחיזוי על ידי יצירת עצים נוספים שמתקנים את טעויות העצים הקודמים. מדובר במודל חזק מאוד, במיוחד כאשר מדובר בכמויות גדולות של נתונים או נתונים רועשים, והוא מציע דיוק גבוה מאוד.

- יתרונות: מספק ביצועים גבוהים מאוד. הוא מצוין עבור בעיות סיווג ורגולציה, במיוחד כאשר מדובר בנתונים רועשים או לא לינאריים. יתרונו הגדול הוא היכולת להתעדכן ולשפר את הביצועים באופן אוטומטי על ידי תיקון השגיאות של העצים הקודמים. הוא גם מאוד מהיר בזכות עיבוד מקבילי.
- חסרונות: החיסרון העיקרי של שהוא דורש הרבה זמן והגדרות פרמטרים כדי לקבל תוצאות טובות. כמו כן, הוא רגיש לאוברפיטינג כאשר לא מכוונים את הפרמטרים בצורה נכונה.

❖ LightGBM: מודל זה מבצע גם הוא Gradient Boosting אך הוא שונה ביכולת לבצע חיזוי בצורה מהירה ויעילה יותר, במיוחד עם נתונים גדולים. היתרון כאן הוא מהירות עיבוד גבוהה.

- יתרונות: הוא גרסה מהירה של GB המתמקדת בשיפור ביצועים על נתונים גדולים. המודל מבצע חיזוי בצורה מאוד מהירה ויעילה, במיוחד כאשר יש הרבה נתונים. יתרון נוסף הוא שהמודל צורך פחות זיכרון מהמודלים הקודמים ומספק ביצועים גבוהים גם עם נתונים פחות מעובדים.
- חסרונות: החיסרון העיקרי שהוא פחות מתאים לעבודה עם נתונים קטנים או כאשר יש הרבה תכונות קטגוריאליות שלא עברו עיבוד מתאים.

❖ Gradient Boosting: מודל נוסף המבוסס על טכניקת הבוסטינג בו כל עץ חדש מבצע תיקון של טעויות העצים הקודמים. זהו מודל חזק שמספק ביצועים טובים מאוד בעבודות סיווג ורגרסיה.

- יתרונות: הוא מודל חזק מאוד המבצע תיקונים בעבודת המודלים הקודמים ומספק דיוק גבוה. זהו מודל נהדר כאשר יש בעיות סיווג מורכבות או בעיות עם נתונים לא לינאריים, והוא עובד בצורה מצוינת עם נתונים לא מעובדים.

- חסרונות: החיסרון העיקרי של Gradient Boosting הוא זמן החישוב. כאשר יש כמות נתונים גדולה, המודל עשוי להיות איטי מאוד לאימון. בנוסף, כמו מודלים אחרים מבוססי בוסטינג הוא עלול להיות רגיש לאוברפיטינג אם לא מכוונים אותו כראוי

❖ Extra Trees: מודל דומה לRandom Forest אך הוא עושה יותר שימוש באקראיות בעת בניית העצים. בדרך כלל, הוא מבצע חיזוי מהיר ויעיל יותר, אבל זה יכול גם להוביל לתוצאות קשות לפירוש.

- יתרונות: מבצע חיזוי על ידי מספר עצים אקראיים מאוד, מה שמפחית את השפעת השגיאות ומספק חיזוי יציב. הוא מהיר במיוחד במקרים של נתונים גדולים ומספק ביצועים טובים גם כאשר יש הרבה תכונות.
- חסרונות: החיסרון העיקרי הוא שיכולת הפירוש של המודל קשה מאוד, מאחר והתוצאה מתקבלת מכמה עשרות עד מאות עצים. כמו כן, במקרים של נתונים רועשים מאוד, הוא עשוי לא להצליח לזהות את הדפוסים בצורה הטובה ביותר.

❖ CatBoost: הוא מודל למידת מכונה המבוסס על Gradient Boosting ומיועד במיוחד לטיפול בנתונים קטגוריאליים. המודל מבצע אופטימיזציה בצורה יעילה ומהירה, ומספק ביצועים טובים במיוחד גם כאשר יש כמות רבה של נתונים קטגוריאליים.

- יתרונות: היתרון העיקרי הוא יכולתו להתמודד בצורה טבעית עם נתונים קטגוריאליים, ללא צורך בהמרות ידניות. בנוסף, המודל מציע ביצועים מהירים במיוחד, ומספק תוצאות טובות גם כשיש הרבה תכונות קטגוריאליות.
- חסרונות: הקושי בהבנת המודל. קשה לפענח איך הוא הגיע לתוצאה הסופית. כמו כן במקרים של נתונים רועשים מאוד המודל עשוי להיתקל בקשיים בזיהוי דפוסים בצורה אופטימלית.

בחרנו במודלים אלו כיוון שהם מציעים יתרונות בולטים בטיפול בנתונים כמו שלנו, שמשלבים משתנים קטגוריאליים ונומריים.
בעת קבלת החלטה אילו מודלים ליישם, נבחנו שיקולים שונים שקשורים למאפייני הדאטה, יכולות האלגוריתמים, מגבלות טכניות והיקף המשימה. בחירה זו נעשתה מתוך מטרה לזהות את המודלים שיכולים להתמודד בצורה האפקטיבית ביותר עם בעיית הסיווג הדו-יעדי (Multi-output classification) שניצבה בפנינו – חיזוי של אזור גיאוגרפי וחודש הדיווח של מקרי כלבת בישראל.

- האם המודל דורש חלוקה של הנתונים לסט אימון וסט בדיקה?

רוב המודלים שנבחרו – כולל Logistic Regression, Decision Tree, Random Forest, XGBoost, LightGBM ו־Gradient Boosting – מצריכים חלוקה של הדאטה לשני סטים עיקריים: סט אימון (training set) וסט בדיקה (test set). חלוקה זו נועדה למנוע התאמת יתר (overfitting) ולוודא שהמודל מסוגל לבצע הכללה טובה על דאטה חדש שלא נחשף אליו במהלך הלמידה.
בפרויקט זה, כדי לשפר את מהימנות התוצאות ולהפחית תלות בפיצול יחיד, יושמה טכניקה מתקדמת של קרוס-ולידציה מסוג Repeated K-Fold (5 קיפולים ו־5 חזרות). טכניקה זו איפשרה הרצה של כל מודל במספר רב של תתי-מדגמים, כך שניתן להעריך ביצועים ממוצעים תוך מזעור השפעה של חריגים.

גם מודלים כמו K-Nearest Neighbors ו־SVC, אשר רגישים יותר לחלוקה לא מאוזנת או לשינויים במבנה הדאטה, הרוויחו מהשימוש ב־K-Fold בכך שנמנעו תופעות של חוסר עקביות בתוצאות או שגיאות חיזוי כתוצאה מהשפעה של נקודות קצה.

- האם יש כמות מספקת של נתונים להפקת תוצאות אמינות?

נפח הנתונים בפרויקט זה היה מספק וכלל מאות תצפיות עם מגוון משתנים רלוונטיים: משתנים קטגוריאליים כמו חודש, אזור וסוג חיה, לצד משתנים כמותיים כגון טמפרטורה, כמות משקעים ומספר ימי גשם.

המודלים המתקדמים מסוג Boosting (כגון XGBoost, LightGBM ו־Gradient Boosting) מותאמים במיוחד להתמודדות עם דאטה מגוון ורועש, והוכיחו ביצועים טובים גם כאשר קיימת מורכבות בין תכונות. יתרה מזו, מבנה הנתונים אפשר למודלים ללמוד קשרים לא טריוויאליים בין משתנים גיאוגרפיים, אקלימיים וזמניים.

יחד עם זאת, מודלים כמו KNN ו־SVC מצריכים משאבי עיבוד גבוהים כאשר כמות הנתונים עולה, ולכן נבדקה מראש ההיתכנות החישובית של הפעלתם. לא נרשמו בעיות זיכרון או קריסות בזמן הרצה, מה שמצביע על כך שהיקף הדאטה תאם את מגבלות המודלים שנבחרו.

- האם המודל דורש רמת איכות מסוימת של נתונים? האם הנתונים עומדים בדרישה זו?

איכות הנתונים היא קריטית להצלחת תהליך המידול. כל המשתנים הקטגוריאליים הומרו לערכים מספריים באמצעות LabelEncoder, והמשתנים הכמותיים עברו סטנדרטיזציה (StandardScaler) – במיוחד עבור מודלים רגישים להבדלי סקאלה כמו Logistic Regression, SVC ו־KNN. תהליך זה נועד למנוע הטיות עקב שונות גבוהה בין משתנים ולהבטיח תנאי למידה שווים.

ערכים חסרים טופלו מראש באמצעות הסרה של שורות בודדות או מילוי בערכי חציון, כך שנשמרו עקביות פנימית ואמינות בין התכונות השונות. תהליך ניקוי הנתונים איפשר לכל המודלים לפעול על בסיס דאטה איכותי, יציב ונטול עיוותים.

- האם סוג הנתונים מתאים למודלים? ואם לא – האם בוצעו המרות נדרשות?

סוג הנתונים התאים לרוב המוחלט של המודלים. לדוגמה, מודלים מבוססי עצים (כמו Random Forest ו־XGBoost) אינם דורשים סטנדרטיזציה ומתמודדים היטב עם תכונות קטגוריאליות ומספריות. עבור מודלים רגישים יותר כמו SVC ו־KNN, הושם דגש על סקלת המשתנים, נורמליזציה והמרות הכרחיות כדי לשפר את רמת הדיוק ולהבטיח למידה תקינה.

נוסף לכך, בוצעו התאמות קלות לייצוג החודשים (לדוגמה: המרה למספרים או לקטגוריות מסודרות), מה שאפשר למודלים להבין טוב יותר את ההקשר העונתי ולהבחין בין חודשים סמוכים. יישום של טכניקות עיבוד מקדים היה חיוני לתפקוד תקין של כלל המודלים, ושיפר את הביצועים בפועל.

1.2 הנחות המודל (Model Assumptions)

כאשר מצמצמים את מגוון כלי המידול האפשריים לקראת בחירה סופית של אלגוריתמים, ישנה חשיבות רבה לתיעוד שיטתי של תהליך קבלת ההחלטות. תיעוד זה כולל הן את ההנחות שנעשו בנוגע למבנה הנתונים ולמאפייניהם, והן את המניפולציות שבוצעו לצורך התאמתם לדרישות הספציפיות של כל מודל. הבנה ברורה של ההנחות והמניפולציות מאפשרת לא רק לשחזר את שלבי העבודה, אלא גם להסביר את ההבדלים בביצועים בין המודלים השונים.

הנחות שבוצעו במהלך עיבוד הנתונים:

- איכות הנתונים:

הנחנו כי הנתונים שנמסרו בשלב הקודם עברו תהליך ניקוי ראשוני וכי הם תקינים ברובם. כלומר, לא אמורות להופיע שגיאות גסות, כפילויות משמעותיות או חריגות בוטות. יחד עם זאת, במהלך עיבוד הנתונים זוהו ערכים חסרים בעמודות כגון מיקום גיאוגרפי וזן החיה. ערכים אלו טופלו או הוסרו באופן מבוקר, תוך שמירה על שלמות הדאטה והימנעות מהטיית התפלגות הנתונים.

- טיב הקשרים בין המשתנים:

חלק מהמודלים (כגון Logistic Regression, SVC ו־KNN) מניחים קשרים לינאריים או מבנה נרחב שניתן ללמוד רק לאחר עיבוד מתאים של המשתנים. לכן הונח כי נדרשת המרה של משתנים קטגוריאליים לערכים נומריים. לעומת זאת, מודלים כמו XGBoost, LightGBM ו-Random Forest אינם דורשים קשרים לינאריים ויכולים להתמודד היטב עם מבנים מורכבים – ולכן הותאמו להם שיטות עיבוד אחרות.

- סטנדרטיזציה של משתנים נומריים:

הונח כי טווחי ערכים לא אחידים (למשל טמפרטורה בין 10- ל־40 מעלות) עלולים לגרום למודלים כמו KNN או SVC להעניק משקל לא מידתי לתכונות מסוימות. לפיכך, בוצעה סטנדרטיזציה בעזרת StandardScaler על מנת לאזן בין טווחי המשתנים ולהבטיח השפעה אחידה של כל תכונה על המודל.

- טיפול בשונות בין סוגי משתנים:

כיוון שהמשימה כללה חיזוי של שתי מטרות שונות (אזור וחודש), בהתבסס על תכונות קטגוריאליות ונומריות כאחד, הונח כי יש לוודא שהמודלים מסוגלים להתמודד עם מידע הטרוגני. בהתאם לכך, הותאמו שיטות קידוד שונות לכל מודל לפי הצורך, במטרה לשמר את האינפורמציה הרלוונטית ולשפר את ביצועי התחזיות.

מניפולציות שבוצעו על הנתונים:

- קידוד משתנים קטגוריאליים לנומריים:

בוצעה המרה באמצעות LabelEncoder למשתנים כמו חודש, אזור, יישוב וסוג החיה. פעולה זו הייתה קריטית עבור מודלים שאינם תומכים במשתנים טקסטואליים, ומאפשרת חישוב מתמטי תקני על כלל התכונות.

- סטנדרטיזציה לתכונות נומריות:

המאפיינים הנומריים הותאמו לטווחים נורמליים על ידי שימוש ב־StandardScaler, מתוך מטרה לשמור על פרופורציות שוות בין התכונות. הדבר סייע במיוחד עבור מודלים כמו SVC ו-KNN, אשר רגישים לטווחים שונים של ערכים.

- המרת החודש לערך טקסטואלי:

לשם שיפור הקריאות והמשגת התכונה החודשית, הומרו ערכי החודש מ־1–12 לשמות החודשים (January עד December). צעד זה לא נדרש לצורך ההרצה המעשית, אך שימש ככלי עזר לפרשנות אנליטית וייצוג גרפי.

- טיפול בערכים חסרים:

שורות בהן נמצאו ערכים חסרים שאינם ניתנים להשלמה (למשל שם יישוב חסר או סוג חיה חסר) הוסרו מהדאטה. ההחלטה הושפעה מכך שמדובר במידע קריטי לחיזוי מדויק, ולא ניתן היה לבצע השלמה סבירה מבלי לפגוע באמינות הנתונים.

- זיהוי ואיפוס ערכים חריגים (Outliers):

ערכים קיצוניים, דוגמת טמפרטורת קלט של מאות מעלות, זוהו והוסרו מהמערך. פעולה זו בוצעה כדי למנוע השפעות בלתי רצויות על תהליך הלמידה של המודלים.

- הסרת עמודות לא רלוונטיות:

מתוך כלל הנתונים הוסרו עמודות שלא תרמו לחיזוי, כגון Date, War Name, Event Per Year ו־Index Event ID. הסרת תכונות אלה נועדה למנוע רעש מיותר ולהתמקד בתכונות המועילות לתהליך המידול.

לסיכום ההנחות שתוארו והמניפולציות שבוצעו נועדו להבטיח כי מערך הנתונים תואם לדרישות המבניות והטכניות של כלל המודלים שנבחרו, תוך שיפור משמעותי ביציבות, דיוק וחיזוי כללי. שלב זה היווה נדבך מרכזי ביצירת מודלים אמינים וברי הכללה.

---

2. עיצוב בדיקה (Test Design)

לפני שמתחילים בהרצה של המודלים, יש לתכנן מראש כיצד תיבחן הצלחתם. עיצוב בדיקה נכון ומעמיק מהווה שלב קריטי בתהליך המידול, שכן הוא מבטיח שהערכת המודלים תהיה עקבית, אובייקטיבית ומבוססת על קריטריונים שהוגדרו מראש. תהליך זה כולל שני רכיבים מרכזיים: הגדרת קריטריונים ברורים להערכת איכות המודל (Goodness of Model) והחלטה על מערך הנתונים שישמש לבדיקה.

במסגרת זו, הוגדר שהקריטריונים המרכזיים לבחינת הצלחת המודלים יהיו דיוק הסיווג (Accuracy) של כל אחד משני משתני היעד – האזור הגיאוגרפי (Region) וחודש הדיווח (Month). Accuracy נבחר כמדד עיקרי מכיוון שהוא מספק דרך ישירה ופשוטה למדוד את שיעור החיזויים הנכונים של כל מודל עבור כל משתנה יעד. כמו כן, נלקחה בחשבון האפשרות להשתמש במדדים משלימים (Precision, Recall, F1-Score), במיוחד במידה שיתגלה חוסר איזון בין הקטגוריות – למשל, חודשים או אזורים שמיוצגים בדאטה בתדירות שונה.

כדי לאמוד את המודלים בצורה אחידה ואמינה, הוחלט מראש להשתמש בשיטת Repeated K-Fold Cross-Validation – חלוקה חוזרת של הנתונים ל־5 קיפולים ו־5 חזרות. בשיטה זו, הנתונים מחולקים בכל איטרציה ל־5 חלקים: בכל פעם 4 משמשים כסט אימון והחמישי כסט בדיקה. כל קטע מקבל תור שווה לשמש כסט בדיקה, כך שבסוף התהליך מתקבלת תמונה ממצעת ואובייקטיבית של ביצועי המודלים על פני כלל חלקי הדאטה. הגישה הזו נבחרה מראש כדי לצמצם הטיה הנובעת ממדגם ספציפי, ולאפשר למודלים להיבחן על כל חלקי הדאטה ללא חפיפות מיותרות.

הנתונים שנעשה בהם שימוש בתהליך הבדיקה כוללים הן משתנים קטגוריאליים (כמו חודש, אזור, סוג חיה) והן משתנים נומריים (כמו טמפרטורה, כמות משקעים וימי גשם). כבר בשלב התכנון הוגדרו פעולות עיבוד מקדים – כמו המרת משתנים קטגוריאליים לנומריים באמצעות LabelEncoder וסטנדרטיזציה של משתנים נומריים באמצעות StandardScaler – כדי לוודא שהנתונים מותאמים לדרישות של כל אלגוריתם שנבחר.

מאחר שמדובר במודלים מסוג Supervised Learning בלבד, תהליך ההערכה יתבסס כולו על התאמת התוצאה החזויה לתווית הנכונה של כל משתנה יעד. תכנון מראש של מדדי הערכה אלו מאפשר להשוות בין המודלים על בסיס אחיד, גם כאשר כל אחד מהם משתמש באסטרטגיה שונה (עצים, חיזוקים, מרחקים או הסתברויות).

בהקשר זה, תוכננה גם גישה איטרטיבית לבחינת המודלים. עבור כל מודל, תתבצענה עד חמש הרצות עם התאמות פרמטרים מתונות – למשל התאמת עומק עץ, מספר איטרציות או ערכי למידה. במידה והביצועים לא משתפרים לאחר חמש איטרציות, תיבחן האפשרות לעבור לאלגוריתם חלופי. כך ניתן לאזן בין חיפוש אחר פרמטרים אופטימליים לבין יעילות תהליך הפיתוח.

עיצוב בדיקה זה מבטיח שהמודלים ייבחנו באופן שיטתי, מדויק וללא תלות בחלוקה ספציפית של הדאטה, תוך שמירה על סטנדרט גבוה של תכנון מראש – שהוא אבן יסוד בפרויקטים של למידת מכונה וניתוח נתונים.

---

3. תיאור המודל (Model Description)

בשלב זה, לאחר הכנת הנתונים, תהליך הבחירה של משתנים רלוונטיים, וטיפול בערכים חסרים, בוצע תהליך מידול מבוסס השוואה בין מספר אלגוריתמים ללמידה מונחית (Supervised Learning). מטרת המידול הייתה לבנות מערך תחזיות דו-יעדי (Multi-output classification), המנבא הן את האזור הגיאוגרפי (Region) והן את חודש הדיווח (Month) של מקרי כלבת בישראל.

בהתאם להנחיות, כל מודל נבחן באמצעות קרוס-ולידציה מסוג Repeated K-Fold עם 5 קיפולים ו־5 חזרות, מתוך מטרה לייצר הערכה יציבה וריאלית לביצועים לאורך תתי-מדגמים שונים של הדאטה. לכל אלגוריתם שנבדק נשמרו רמות הדיוק (Accuracy) החציוניות של תחזיות האזור והחודש, וכן תועדה כל בעיה שצצה במהלך ההרצה, לצד מאפיינים בולטים או חריגים בהתנהגות המודל.

במהלך תהליך המידול, נעשה שימוש במגוון ספריות מתקדמות כגון scikit-learn, XGBoost, LightGBM, ו־CatBoost. חלק מהמודלים תמכו באופן טבעי בתחזית דו-יעדית באמצעות MultiOutputClassifier, בעוד אחרים דרשו התאמות והפרדה בין מטרות החיזוי. תיעוד שיטתי נשמר לכל ריצה, כולל תצורת המודל, ביצועים, בעיות טכניות, ותובנות איכותיות שהופקו.

3.1 הגדרות פרמטרים (Parameter Settings)

במרבית האלגוריתמים נעשה תחילה שימוש בפרמטרים ברירת מחדל (default settings), וזאת במטרה לבסס קו השוואה ראשוני. לאחר מכן, נערכו התאמות ממוקדות במודלים שנמצא כי יש להם פוטנציאל גבוה לשיפור ביצועים או שנדרשו להתאמה טכנית (למשל במקרה של חוסר תמיכה מובנית ב־multi-output).

להלן תיאור עיקרי של הגדרות הפרמטרים וההתאמות שבוצעו עבור המודלים השונים:

- Gradient Boosting: הוגדר עם מספר n_estimators=100 ועומק מרבי max_depth=3. נעשה שימוש באובייקט GradientBoostingClassifier כבסיס ל־MultiOutputClassifier, שהוכיח את עצמו כמודל המוביל במבחני הדיוק.
- LightGBM ו־XGBoost: עטופים בתוך MultiOutputClassifier, תוך התאמת מספר איטרציות (iterations) ושימוש במנגנון השתקת אזהרות והפחתת פלטים על ידי verbosity=0. השיפור בפלט עזר לזהות בעיות מהותיות בלבד, מבלי להציף את סביבת העבודה.
- CatBoost: לא תומך באופן מובנה בתחזית רב-יעדית multi-output ולכן הוגדרו שני מודלים נפרדים: אחד עבור תחזית האזור ואחד עבור תחזית החודש. נעשה שימוש בפרמטרים כגון iterations=100 מבלי להפעיל Grid Search בשלב זה. תצורה זו אפשרה בדיקה שקופה של ביצועי המודל לכל יעד בנפרד.
- K-Nearest Neighbors (KNN): הוגדר עם k=5 כערך התחלתי, אך התוצאות לא הצדיקו כוונון נוסף. הביצועים הנמוכים הצביעו על חוסר התאמה של שיטה זו לאופי הנתונים.
- SVC ו־Logistic Regression: הוגדרו עם פרמטרים פשוטים תוך שימוש ב־error_score="raise" לצורך איתור מיידי של בעיות חישוביות וכדי לזהות שגיאות מידיות. כמו כן, הוגדרה השתקה של אזהרות כלליות נפוצות והפחתת הדפסות מיותרות באמצעות פרמטרים כגון verbose=0 כדי לשפר את יעילות הקוד, קריאות הפלט וזמן הריצה של הקוד.

בנוסף, במהלך השיפור האחרון של הקוד, הוסרו רווחים משמות עמודות שהפריעו לריצת מודלים כמו LightGBM ו־CatBoost, והוטמעו מנגנוני ניהול שגיאות חכמים שאפשרו להריץ את כל המודלים באחידות. התיעוד המפורט של כל פרמטר ותצורה אפשר להבין את השפעתם על תוצאות המודלים, ומהווה תשתית לעבודה חוזרת או הפעלה אוטומטית על דאטה חדש בעתיד.

3.2 תיאור תוצאות המודלים (Descriptions of Model Results)

בשלב זה בוצעה הרצה שיטתית של עשרה מודלים מסוג classification מתוך ספריות בפייתון, במטרה לזהות את האלגוריתם שמניב את הביצועים הטובים ביותר עבור תחזית כפולה: גם של האזור הגאוגרפי שבו התרחש אירוע הכלבת, וגם של החודש שבו התרחש. תהליך ההערכה התבסס על שיטת קרוס-ולידציה

מסוג RepeatedKFold עם 5 קיפולים ו־5 חזרות (סה"כ 25 ריצות לכל מודל), שנועדה להבטיח יציבות בתוצאות על פני חלוקות שונות של הדאטה.
שני מדדי הביצוע המרכזיים שהוערכו היו:

- Region Accuracy – דיוק החיזוי של האזור הגאוגרפי.
- Month Accuracy – דיוק החיזוי של החודש שבו התרחש האירוע.

יכולת הפקת מסקנות מהממצאים:
הניתוח העלה באופן ברור את יתרונות המודל Gradient Boosting, שהשיג את הביצועים הגבוהים ביותר מבין כל המודלים, עם דיוק של כ־95.84% בתחזית האזור ו־87.08% בתחזית החודש. התוצאות הגבוהות והעקביות שלו משקפות לא רק את האפקטיביות של האלגוריתם אלא גם את ההתאמה שלו למורכבות הנתונים – בהם מופיעים משתנים עונתיים, גאוגרפיים ומאפייני מזג אוויר.
מודלים מתקדמים נוספים כמו XGBoost ו־LightGBM השיגו ביצועים דומים בדיוק תחזית האזור (כ־95.78%) אך ירדו לרמות של 84.05% ו־85.08% בהתאמה עבור תחזית החודש, מה שמרמז על כך שמודל ה־Gradient Boosting מצליח ללכוד טוב יותר את הדפוסים הזמניים של נתוני הכלבת – יכולת קריטית כאשר נדרש חיזוי לפי עונה וחודש.

דפוסים, חריגות ותובנות:
בחלק מהמודלים הפשוטים יותר נצפתה ירידה חדה ביכולת החיזוי של החודש. מודלים כמו Logistic Regression ו־SVC הציגו דיוקים נמוכים במיוחד (28% ו־10.5% בהתאמה), דבר שמחזק את ההשערה שהמבנה הפנימי של הדאטה אינו מאפשר הפרדה ליניארית פשוטה, במיוחד בממד הזמן.
מודל K-Nearest Neighbors, שמבוסס על מרחקים במרחב מאפיינים, הראה ביצועים חלשים של כ־18% בלבד בחיזוי החודש. ככל הנראה, המרחקים הגיאוגרפיים והעונתיים אינם מספקים הבחנה טובה בין הקטגוריות כאשר הנתונים בעלי רעש, חפיפות או חוסר ייצוג שווה.
בניגוד לכך, Naive Bayes הציג יתרון בזמן הריצה אך הפגין ביצועים בינוניים. הוא הגיע ל־88.6% בדיוק האזור, אך כשל בתחזית החודש (כ־35%), כנראה בשל ההנחות הסטטיסטיות שהוא מניח לגבי אי־תלות בין המשתנים – הנחות שלא מתקיימות במלואן במאגר הנתונים הזה.
מודל CatBoost דרש טיפול נפרד מאחר והוא לא תומך ב־MultiOutputClassifier בצורה ישירה. לכן הופרדו החיזויים לשני תהליכים שונים (אחד לאזור ואחד לחודש), מה שגרם לירידה מסוימת בדיוק תחזית החודש (כ־77.95%) לעומת תוצאה מרשימה באזור (95.64%).

תקלות טכניות ופתרונן:
בגרסאות מוקדמות של הקוד, זוהו בעיות בתצוגת הפלט ובמנגנוני הרצה, בעיקר במודלים כמו LightGBM. הודעות אזהרה מרובות מסוג No further splits with positive gain שיבשו את הקריאות של תהליך ההרצה ויצרו עומס מיותר בתיעוד.
בגרסה המשופרת טופלה בעיה זו באופן יסודי:

- רווחים הוסרו משמות עמודות שהפריעו לפרסינג פנימי של המודלים.
- פרמטר verbose הוגדר כ־0 או -1 במודלים הרלוונטיים, מה שצמצם את הפלט הלא רלוונטי.
- נעשתה השתקה של אזהרות לא מהותיות במיוחד עבור מודלים כמו LightGBM.

כך נוצר תהליך הרצה הרבה יותר יציב ונקי, שמאפשר לנתח את התוצאות בצורה יעילה וברורה.

איכות נתונים וחישובים:

לא נמצאו בעיות חמורות באיכות הנתונים. ערכים חסרים טופלו מראש באמצעים סטטיסטיים כגון מילוי חציוני, והנתונים עברו תהליך נרחב של קידוד וסטנדרטיזציה. כמו כן, לא זוהו חישובים שגויים או חריגות מתמטיות במהלך הקרוס־ולידציה.

---

4. הערכת המודל (Model Assessment)

לאחר שלב ההרצה וההשוואה בין כלל המודלים, בוצעה הערכה שיטתית ומבוססת נתונים שמטרתה להכריע אילו מודלים היו מדויקים, עקביים ויעילים מספיק כדי להיחשב כמודלים הסופיים לצורך המשך שימוש או יישום. ההערכה הסתמכה הן על קריטריונים אובייקטיביים של דיוק והן על קריטריונים סובייקטיביים, כמו פשטות השימוש, אמינות הריצה וקלות הפירוש של התוצאות.

(1) אובייקטיבי – ממוצע דיוק של חיזוי Region ו־Month;

(2) סובייקטיבי – יציבות תוצאות, רמת הפשטות של המודל, כמות ההתאמות שנדרשו.

במהלך ההרצות, כל מודל עבר חמישה ניסויים חוזרים תוך שימוש בקיפול צולב (cross-validation), דבר שאיפשר לבחון לא רק את הביצועים הממוצעים אלא גם את עקביות התוצאה לאורך ריצות שונות. שני המדדים שנבחנו עבור כל מודל היו: דיוק ממוצע בחיזוי האזור (Region Accuracy) ו־דיוק ממוצע בחיזוי החודש (Month Accuracy).

במהלך ההשוואה ניכרה עליונות עקבית של מודל Gradient Boosting. מודל זה הציג את הביצועים הגבוהים ביותר בקרב כלל המודלים, עם דיוק ממוצע של כ־0.9584 עבור חיזוי האזור, ודיוק של 0.8708 עבור חיזוי החודש. יתר על כן, מודל זה הפגין יציבות מרשימה בין הרצות, והצליח לשמר רמות ביצוע גבוהות גם בתתי-קבוצות של הנתונים. מעבר לביצועים החישוביים, מודל Gradient Boosting בלט בפשטות יחסית בהפעלה, ללא צורך בכוונון יתר או שינוי פרמטרים פרט להגדרות ברירת המחדל. לא נרשמו אזהרות או הודעות שגיאה, זמן הריצה היה סביר מאוד, והפלט הסופי היה נקי, תמציתי וברור.

לשם השוואה, מודלים נוספים שהתקרבו לביצועים של Gradient Boosting היו XGBoost ו-LightGBM, אך אלו הצריכו טיפול נוסף בהתראות מערכת (כמו התראות על שמות עמודות, אינסוף ב-gain וכו') או הפיקו פלט ארוך יחסית עם אזהרות חוזרות במהלך ההרצה הראשונית. אמנם לאחר שיפורים פונקציונליים בקוד חלק מהבעיות נפתרו, אך השיפור המשמעותי ביותר באיזון שבין ביצועים, פשטות ושקט תפעולי התקבל במודל Gradient Boosting.

טבלת ההשוואה סידרה את המודלים מהטוב לפחות טוב (בדיוק), כאשר Gradient Boosting דורג ראשון, ואחריו XGBoost ו־LightGBM. המודל בעל הביצועים הנמוכים ביותר היה SVC. ההערכה מראה כי מודלים מבוססי עצים (Boosting / Forest) מתאימים במיוחד למשימת סיווג זו, הן מבחינת ביצועים והן מבחינת יכולת ניתוח.

טבלת ההשוואה הסופית בין כלל המודלים לפי שני מדדי הדיוק מוצגת להלן:

Model	Average Region Accuracy	Average Month Accuracy
Gradient Boosting	0.958406	0.870836
XGBoost	0.957847	0.840485
LightGBM	0.957847	0.850862
Random Forest	0.956440	0.813507
CatBoost (Separate)	0.956437	0.779531
Extra Trees	0.944933	0.845825
Decision Tree	0.937351	0.786544
Naive Bayes	0.886193	0.350346
Logistic Regression	0.837928	0.284864
K-Nearest Neighbors (KNN)	0.655627	0.180327
SVC	0.523940	0.105049

לסיכום, הבחירה הסופית במודל Gradient Boosting נתמכת הן על ידי התוצאות הכמותיות והן על ידי שיקולים איכותיים הקשורים לנוחות שימוש ואמינות. המודל מספק ביצועים גבוהים בשני יעדי התחזית, קל להפעלה ולתחזוקה, ומתאים להטמעה בשלבים הבאים בתהליך של הפרויקט גמר.

לצורך הגשה, מצורפים מספר קבצים:

1.קובץ קוד בפורמט Google Colab הכולל את הקוד המלא וההרצות – לצפייה בלינק:

https://colab.research.google.com/drive/1q--hAeKdSvedGbSAdJ6puLrcnyINbGqz?usp=sharing

2.קובץ Python בפורמט .py עם עותק של הגרסה העדכנית של הקוד.

3.קובץ Word מסכם הכולל עותק של הקוד, ואת הפלט המלא שהתקבל כולל דירוג תוצאות המודלים.

4.קובץ הנתונים העדכני לשם הרצת הקוד.

5. קובץ וורד זה וגרסת PDF שלו לצורך קריאה ברורה.

בברכה,

דניאל ואימאן.