אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

Faculty of Engineering Sciences

Department of Industrial Engineering and Management

Credit Card Frauds Classification Using Machine Learning

Tsoof Bar-Or and May Hakim

Prof. Boaz Lerner

23.2.2022

תקציר

בפרויקט זה, בחרנו לפתור בעיית סיווג בינארית בנושא הונאות כרטיסי אשראי. הונאת אשראי היא מעשה פלילי הרווח מאוד ברחבי העולם כיום, בייחוד עם מגמת הקניות ברשת שהפכה לפופולרית בשנים האחרונות.

בעזרת מאגר נתונים סינתטי שבחרנו מאתר Kaggle המתעד עסקאות אשראי אימנו מודלים של עץ החלטה, יער אקראי, XGBOOST, רשת נוירונים ורגרסיה לוגיסטית כדי לפתור את השאלה- האם עסקה X היא הונאה?

במסגרת העבודה על המאגר התמודדנו עם חוסר האיזון של מאגר הנתונים שתואם את חוסר האיזון של הבעיה במציאות, עם כמות גדולה של מאפיינים ועם מאפיינים קטגוריאליים שכללו קטגוריות מרובות.

לאחר צמצום משמעותי של כמות המאפיינים במאגר הרצנו חיפוש מקיף על המודלים: עץ החלטה, יער אקראי ו-XGBoost בו בדקנו עבור כל מודל קומבינציות שונות בגדלים מאפיינים של מאפיינים עם כוונון היפר-פרמטרים עבור כל בדיקה כזו.

שמרנו את תוצאות החיפוש כולל רמת הדיוק במדד f1 של כל בדיקה ובעזרתן מצאנו את המאפיינים הדומיננטיים במודלים המדויקים ביותר שהתקבלו.

לבסוף, השתמשנו במאפיינים הנבחרים לאימון נוסף של כל אחד מחמשת המודלים שפורטו לעיל, הפעם עם כוונון פרמטרים משמעותי יותר, כדי לבדוק איזה מודל הוא הטוב ביותר לפתרון הבעיה שלנו.

המודל שנבחר כטוב ביותר הוא XGBoost, עם דיוק f1 של 71.67%, ו-ROC AUC של 88.96%.

ראוי לציין שביצועיו של עץ ההחלטה היו קרובים מאוד (f1 של 64.94% ו-ROC AUC של 89.84%) ואף מדויקים יותר בגילוי הונאות (True Positive), אך זאת על חשבון התראות שווא (False Positive).

תוכן עניינים

הקדמה

על העבודה

העבודה עוסקת בהונאות אשראי. עבור כל רשומה נתונים רבים, בפרט נתונים קטגוריאלים רבי קטגוריות. הבעיה במהותה בעיה לא מאוזנת, רוב העסקאות בעולם הם עסקאות תקינות, ורק מיעוט קטן הן הונאות. כדי להתמודד עם הבעיה, נדרשות מערכות לומדות המטפלות היטב במאפיינים קטגוריאליים ובמידע לא מאוזן, לכן בחרנו להשוות בין עץ החלטה, יער אקראי ו-XGBoost. בפרט, עם אלגוריתם XGBoost כבר בוצעו ניסיונות מוצלחים לחיזוי הונאות אשראי. לצורך ההשוואה, ביצענו את החיזוי גם בעזרת רגרסיה לוגיסטית ורשתות נוירונים.

נציג את העבודה על בעיית הסיווג עפ"י השלבים של מתודת CRISP-DM.

על האלגוריתמים

עץ החלטות הוא אלגוריתם חמדני, המפצל את בסיס הנתונים לפי המאפיין שמפריד את המחלקות בצורה הטובה ביותר לפי המטריקה הנבחרת עד לתנאי העצירה. הסיווג מתבצע ע"פ קבוצת הרוב שנמצאת בתת הפיצול הרלוונטי. העץ מייצג סדרה של שאלות (האם שייך לקטגוריה A? האם ערך מעל 100? וכדומה) שבסופן ניתן לסווג כל תצפית לאחת המחלקות. **יער אקראי** הוא מודל מכלול (ensemble) של מספר רב של עצים שנבנים בצורות שונות שבסופו של דבר מבצעים "הצבעה" על הסיווג הנכון לתצפית. **XGBoost** הוא וריאציה של יער אקראי המשתמשת בגרדיאנטים על מנת למדל את העצים והקשר ביניהם, ונחשבת לאחת הוריאציות הפופולריות והמוצלחות של יער אקראי. **רגרסיה לוגיסטית** היא שיטה לשערוך הסתברות פוסטריורית להשתייך לכל אחת מהמחלקות בעזרת קומבינציה ליניארית של המאפיינים ושימוש בפונקציה ייעודית לנרמול. **רשת נוירונים** היא שיטה לשערוך הסתברות פוסטריורית בעזרת קומבינציה לא ליניארית של המאפיינים על ידי פעולות חיבור, כפל ופונקציות לא לינאריות כמו סיגמואיד, tanh ועוד.

על הקוד

הקוד כולו מורכב מארבעה קבצי פייתון. קובץ ה-Main מאחד את כולם לסקריפט יחיד שניתן להריץ בשלמותו. הסקריפט ב-Main <u>לא מבצע את הדגימה לבחירת המאפיינים באופן דיפולטיבי</u>, רעיון שמוסבר בגוף העבודה, אלא משתמש כבר בתוצאות הדגימה מקבצי CSV שגם הם מצורפים לקוד. במידה ונדרש להריץ את תהליך הדגימה שוב (תהליך של 16 שעות) ישנו משתנה בוליאני בשם sample שניתן לשנות את ערכו ל-True בתוך הקוד. קבצי הפייתון הנוספים הם plot_functions- אוסף פונקציות ששימשו אותנו לויזואליזציה של הנתונים והתוצאות, training_functions- מעטפת למודלים השונים שאימנו במהלך העבודה ו-util_functions, אוסף של פונקציות עזר לשינויים הנדרשים להכנת המידע להכנסה למודלים (לשלב ה-Pre-Process).

Business Understanding

הונאות אשראי הן בעיה נפוצה בכל מדינות העולם, בה אדם משתמש בכרטיס אשראי לא שלו לביצוע תשלום עבור מוצר או שירות, כך שבפועל גונב את הכסף. הונאת אשראי יכולה להתבצע לאחר גניבת כרטיס פיזי ושימוש בו, או בגניבת הפרטים עצמם ושימוש בהם ברשת.

במקרים בהם ישנו שימוש פיזי בכרטיס, ישנן מדינות בהן עבור סכומים גדולים יש צורך להקיש קוד סודי המהווה שכבת ביטחון נוספת. מנגנון כזה לא קיים בקניות ברשת ברוב המקרים, בהן גניבת פרטי אשראי יכולה להתבצע במהלך כל רכישה בה מוכנסים הפרטים המלאים של הכרטיס. הונאת אשראי יכולה להתרחש גם כאשר חשבון של אדם, ללא ידיעתו, משלם על פעולות המתרחשות בחשבון של עבריין.

ישנן שיטות רבות לאיתור הונאות אשראי, ביניהן בניית פרופיל מידע לכל משתמש על מנת להבין כיצד הוא מתנהג ביום יום. כאשר מזוהה התנהגות חריגה, החברה יכולה לחשוד שמדובר בהונאה ולשהות את פעולת התשלום עד אישור בעל פה של בעל האשראי. פעילות חריגה יכולה להיחשב כ:

- הוצאה גדולה, או רצף של הרבה הוצאות קטנות
- קניות מרובות אצל אותו מוכר
- קניות במדינה או באזור גיאוגרפי חריג למשתמש
- קניות בזמנים לא אופיינים

ישנם מודלים רבים שנבנו לזיהוי הונאות אשראי. הבעיה העיקרית בכולם היא חוסר האיזון הקיצוני של המידע, שכן כ99.99% מהעסקאות לא מדובר בהונאה (Woolston, S. E. 2017). בנוסף, המידע על עסקאות האשראי הוא מידע פרטי ורגיש ולכן לרוב לא נחשף לציבור כלל, ואם בוחרים לחשוף אותו מבצעים עליו מניפולציות כמו PCA כדי שלא יוכלו להסיק את הפרטים המדויקים של העסקאות.

כיום, ישנו שימוש נרחב בשיטות של לימוד מכונה כדי לזהות דפוסים בעסקאות אשראי וזיהוי הונאות. הבעיה היא בעיית סיווג בינארית, כאשר המידע הוא מידע טבלאי.

Data Understanding

מאגר הנתונים בו בחרנו הוא מאגר סינתטי מ-Kaggle שכל רשומה בו מתארת טרנזקציה בכרטיס אשראי. אנו נתייחס אליו כאל מאגר אמיתי עליו נעשה את המחקר. כאמור, מדובר בבעיית סיווג בינארית (0- עסקה אמיתית, 1- הונאה). המאגר מגיע מחולק מראש לסט אימון וסט בחינה. בסט האימון ישנן 1,296,675 רשומות, ובסט הבחינה 555,719 רשומות. בסט האימון ישנן 0.58% רשומות שמסווגות כהונאה, ובסט הבחינה 0.39%. כלומר, הסטים אינם מאוזנים בדומה לבעיה האמיתית.
בסיס הנתונים כולל את השדות הבאים:

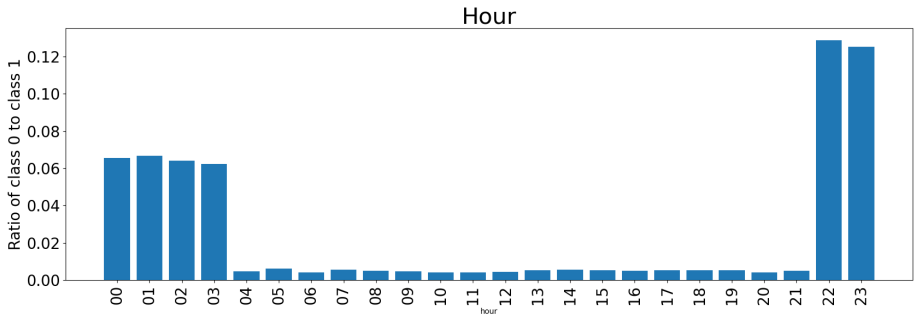
- trans_date_trans_time: תאריך ושעה של הטרנזקציה, **תאריך ושעה**
- cc_num: מספר כרטיס האשראי, **מזהה ייחודי אקראי (לא תואם מספרי אשראי אמיתיים)**
- merchant: שם המוכר שקיבל את התשלום, **קטגוריאלי (693 קטגוריות)**
- category: סוג השירות אותו נותן המוכר, **קטגוריאלי (14 קטגוריות)**
- amt: סכום הטרנזקציה בדולרים אמריקאיים, **ערך רציף**
- first: שם פרטי של הקונה
- last: שם משפחה של הקונה
- gender: גבר או אישה, **קטגוריאלי**
- street: הרחוב בו הקונה מתגורר, **קטגוריאלי (924)**
- city: העיר בה הקונה מתגורר, **קטגוריאלי (894)**
- state: המדינה בארה"ב בה הקונה מתגורר, **קטגוריאלי (51)**
- zip: קוד הדואר של הקונה, **רציף**
- lat: קו הרוחב בו גר הקונה, **רציף**
- long: קו הגובה בו הקונה גר, **רציף**
- city_pop: כמות האנשים הגרים בעיר בה הקונה גר, **רציף**
- job: העבודה של הקונה, **קטגוריאלי (494)**
- dob: תאריך הלידה של הקונה, **תאריך**
- trans_num: מזהה יחודי של הטרנזקציה
- unix_time: הזמן בו התקיימה הטרנזקציה בפורמט Unix, **רציף, כפילות עם trans_date_trans_time**
- merch_lat: קו גובה של המוכר, **רציף**
- merch_long: קו אורך של המוכר, **רציף**
- is_fraud: משתנה המטרה, האם הטרנזקציה היא רמאות או לא, **בינארי**

Data Preparation

תהליך הכנת הנתונים כלל שני מישורים- צמצום כמות המאפיינים וצמצום כמות הרשומות. אם מסתכלים על כל מאפיין קטגוריאלי כעל K מאפיינים בינאריים לפי הקטגוריות השונות, נקבל לא פחות מ-3,056 מאפיינים (עמודות) במאגר. על כן, היה עלינו לצמצם משמעותית את כמות המאפיינים כדי לקבל מודל פשוט ויעיל שניתן להסבירו. בנוסף, בסיס הנתונים מאוד לא מאוזן. הן משיקול זה והן משיקול כוח מחשוב (כמות הרשומות במאגר גררה זמני ריצה ארוכים מאוד), בחרנו לצמצם את הרשומות בקבוצת העסקאות הלגיטימיות (is_fraud=0) מכמות של מעל מיליון ל-250,000, בדגימה אקראית.

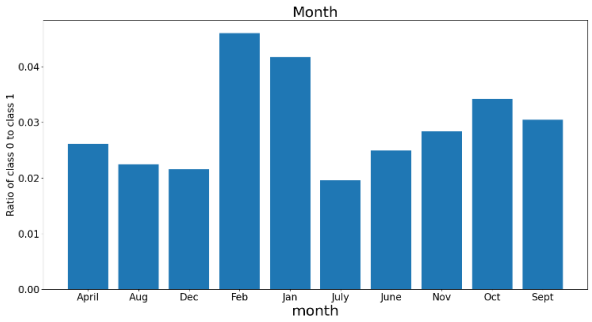
1. מיצוי מאפיינים

בשלב הראשון דווקא הגדלנו את כמות המאפיינים, מתוך הצורך למצות מאפיינים משמעותיים מתוך שדות שלא ניתן להשתמש בהם בצורה ישירה - תאריכים. מתוך מאפייני הזמן במאגר- זמן העסקה ותאריך לידת מבצע העסקה יצרנו שלושה מאפיינים חדשים: **שעת העסקה, חודש העסקה ושנת הלידה של מבצע העסקה**. בחרנו לעשות זאת מתוך מחשבה שיש שעות מסוימות ביום בהן יותר סביר שיתרחשו הונאות, חודשים בשנה בהם מתקיימות יותר הונאות ושבני גילאים מסוימים חשופים יותר להונאות. לשנת הלידה התייחסנו כאל משתנה רציף.
כדי למצוא מגמות בנתונים, הסתכלנו על היחס בין כמות ההונאות לכמות העסקאות הלגיטימיות במידע בכל אחד מהמאפיינים הנבדקים. עבור כמות כל כך גדולה של נתונים (יותר ממיליון רשומות), ראינו לנכון להניח שההסתברות למקרה של תת-דגימה או דגימת יתר של אחת הקטגוריות תצטמצם ונוכל לקבל מידע משמעותי מהיחס הזה.
לאחר מיצוי מאפיין השעה הסתכלנו על היחס הזה עבור כל שעה ביום, וראינו באופן ברור 3 קבוצות: 00-03 בלילה, 04-09 בבוקר ו-22-23 בערב. הפכנו את הקבוצות הללו לקטגוריות A, B, C בהתאמה כמתואר בתרשים 1.



תרשים 1: יחס הונאות\לא הונאות בכל שעה ביום

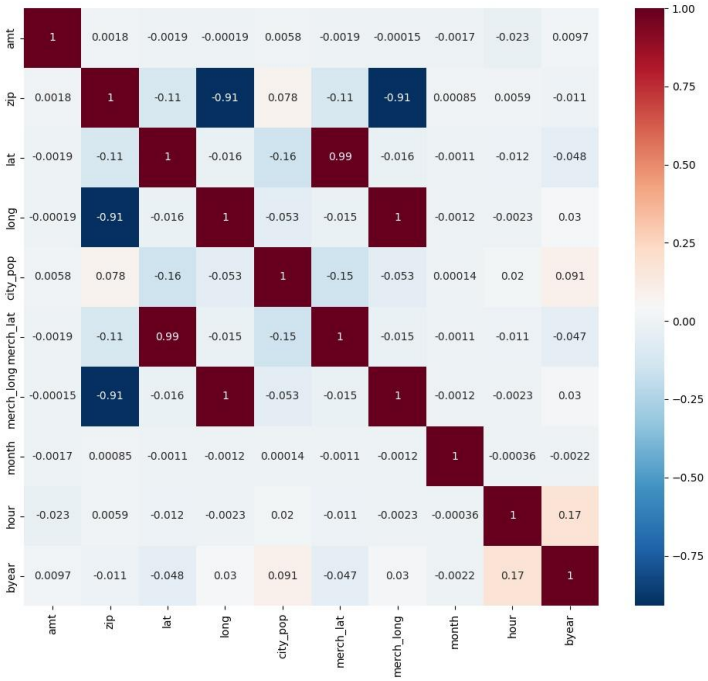
עבור המאפיין החדש של חודש העסקה לא ראינו דפוס ברור, והחלטנו להשאירו כפי שהוא ולהתייחס אליו כאל קטגוריאלי כמתואר בתרשים 2.



תרשים 2: יחס הונאות\לא הונאות לכל חודש בשנה

2. טיפול במאפיינים רציפים

לאחר יצירת המאפיינים החדשים, בדקנו קורלציות בין המאפיינים הרציפים השונים כמתואר בתרשים 3. ראינו קורלציה כמעט מוחלטת בין קואורדינטות האורך והרוחב של מבצע העסקה (long, lat) ושל העסק (merch_long, merch_lat), לכן בחרנו להסיר את משתני הקואורדינטות של העסק (כאשר ברור לנו שבמציאות יתכן שיש משמעות למיקום העסק והסיכוי לקורלציה כזו לא סביר, אך כיוון שהנתונים סינתטיים בחרנו להתעלם מכך). קורלציות דומות נצפו גם בין ה-ZIP Code של מבצע העסקה לקואורדינטות האורך של המוכר ושל מבצע העסקה, ועל כן הסרנו גם את המשתנה zip.

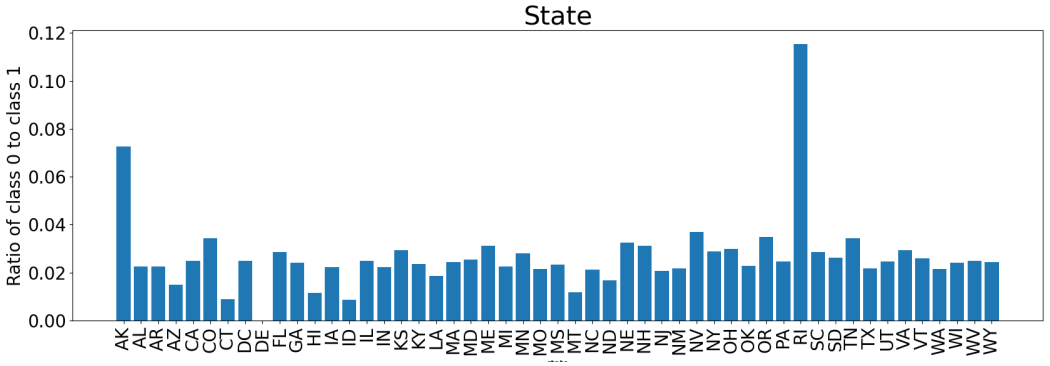


תרשים 3: מטריצת קורלציות בין המאפיינים הרציפים

3. צמצום קטגוריות

ביצענו את תהליך האלימינציה גם עבור המדינות השונות בארה"ב.

במדינה DE (דלאוור) היו סה"כ 9 רשומות, וכולן היו של הונאות. כיוון שהייצוג של המדינה היה מאוד לא משמעותי מבחינת כמות הרשומות וכיוון שכמות ההונאות מתוכן לא מייצגת את יחס ההונאות האמיתי, בחרנו שלא להכניס את הקטגוריה הזו למודל (בתרשים המוצג הכנסנו אותה כ-0).



תרשים 4: יחס הונאות\לא הונאות לכל מדינה בארה"ב

ניתן לראות בתרשים 4 כי ישנן שתי מדינות בהן היחס חריג, בעוד שבשאר הוא די דומה. מסיבה זו, החלטנו לבחור מתוך המאפיין הקטגוריאלי state שני מאפיינים בינארים- AK (אלסקה) ו-RI (רוד איילנד), כאשר ערך 0 בשניהם מעיד על כך שהרשומה שייכת לאחת מהמדינות האחרות בארה"ב.

בשלב הבא, הסתכלנו על המאפיינים Merchant, Job, City, Street. כולם מאפיינים קטגוריאליים עם מאות קטגוריות לכל אחד. כדי להתמודד עם כמות המאפיינים העצומה הזו, תחילה בדקנו את יחס ההונאות בסט החדש (לאחר הצמצום ממיליון רשומות ל-250,000), והתוצאה הראתה כי ההונאות מהוות כ-3% מכלל הרשומות. בשלב הבא, עבור כל אחד מהמאפיינים הקטגוריאליים שציינו, בדקנו את יחס ההונאות בכל הקטגוריות המרכיבות אותו. לכל רשומה נוסף מאפיין בינארי חדש המקבל 0 עבור יחס מתחת לממוצע (3%) ו-1 עבור יחס מעל הממוצע עבור כל אחד מהמאפיינים בנפרד, כך שבסוף התהליך נשארנו עם 4 מאפיינים בינאריים חדשים.

4. בחירת מאפיינים - דגימה

עד כה, ירדנו מ-3,056 מאפיינים ל-47 בלבד. את השלב האחרון בצמצום ביצענו בשיטת חיפוש מקיפה. לשם כך, בחרנו שלושה מודלים: **עץ החלטות, יער אקראי ו-XGBoost.**

בשיטה הזו אנחנו מניחים ששימוש במספר מודלים יתן לנו הערכה טובה יותר על איכות המאפיינים המתאימים לבעיה באופן כללי. המודלים נבחרו משיקולי זמני חישוב קצרים. מודל הרגרסיה הלוגיסטית נבחן גם הוא, אך לא השתמשנו בו בבדיקה הסופית עקב תוצאות דיוק נמוכות משמעותית משאר המודלים.

ביצענו תהליך דגימה כך לכל אחד מהמודלים בנפרד:

- נדגם מספר המאפיינים שיכנסו למודל: 8, 12 או 16.
- בוצע כוונון פרמטרים קצר שנבחן באמצעות CV-3.
- נשמר ציון ה-f1 הממוצע של כל מודל.

כל בדיקה כזו, עבור כל מודל, נמשכה 4 שעות. סה"כ אומנו 1,942 עצים, 70 יערות אקראיים ו-170 מודלים של XGBoost. כיוון שזמן החיפוש היה זהה עבור כל מודל, השוני נובע מזמני הריצה של כל אלגוריתם.

לאחר מכן, בחרנו את 15 המודלים הטובים ביותר מכל אחת מהקבוצות (סה"כ 45) וקיבצנו אותם לניתוח התוצאות- עבור כל מאפיין, בדקנו בכמה מהמודלים הטובים ביותר הוא הופיע.

בחרנו לקחת את כל המאפיינים שהופיעו <u>ביותר משליש</u> מהמודלים:

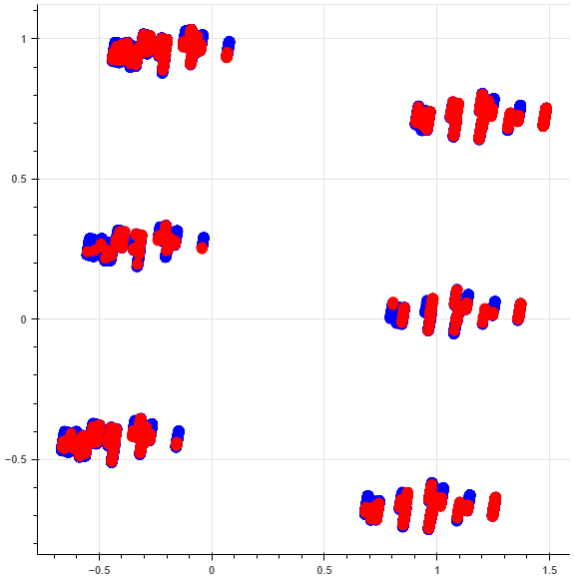
Feature	Amount	Gas Transport (C)	Groceries_pos (C)	Alaska (State)	Entertainment (C)	Group A (Hour)
Appearances	100%	66%	60%	46%	44%	42%
Feature	April (Month)	City Cutoff	Kids Pets (C)	Travel (C)	Group B (Hour)	Home (C)
Appearances	40%	40%	37%	37%	35%	35%
Feature	Shopping (C)	June (Month)	Merchant Cutoff	Birth Year	Groceries_net (C)	Street Cutoff
Appearances	35%	35%	35%	33%	33%	33%

טבלה 1: אחוז המקרים בהם מאפיין הופיע ב-45 המודלים הטובים ביותר, עבור כל מאפיין שנבחר למודלים הסופיים

בסוף, נשארנו עם 18 מאפיינים לשימוש במודלים הסופיים כפי שניתן לראות בטבלה 1. מאפיינים עם הסיומת Cutoff הם המאפיינים הבינארים של מעל או מתחת לממוצע היחס 3% שתוארו בסעיף 4, ומאפיינים שלידם (C) הם הקטגוריות מהמאפיין הקטגוריאלי Category.

5. ערכים חסרים או חריגים

מכיוון שהנתונים שלנו הם נתונים סינטטיים, אין בהם כלל ערכים חסרים. החלטנו לבצע את ניתוח החריגים רק לאחר בחירת המאפיינים, שכן באלפי מאפיינים קשה מאוד להגדיר מה הוא חריג. לאחר צמצום המאפיינים נשארנו עם 16 מאפיינים קטגוריאליים ו-2 מאפיינים רציפים. כדי לבצע הערכת חריגים, ביצענו PCA לכל התצפיות.



תרשים 5: PCA לבסיס הנתונים. באדום עסקאות הונאה, בכחול עסקאות לגיטימיות

ניתן לראות בתרשים 5 כי אין נקודות חריגות משמעותית שנמצאות מחוץ לצבירים העיקריים. גם כאשר מתמקדים בכל צביר בנפרד, לא ניתן לראות נקודות שנמצאות במרחק משמעותי מספיק כדי לסווג אותן כחריגות. מספר התצפיות גדול מאד (מעל 250 אלף), ולכן גם מספר קטן של נקודות חריגות ישפיעו בצורה מתונה יחסית על התוצאה. מסיבה זו, החלטנו לא להוציא אף תצפית מהסט. לפרויקט מצורף קובץ HTML אינטראקטיבי בו ניתן לבחון את התרשים ברזולוציות שונות.

Modeling

המודלים ביניהם בחרנו להשוות הם Decision Tree, Random Forest, XGBoost, MLP ו-Logistic Regression, כולם מהספרייה scikit learn. לאחר בחירת המאפיינים הסופית, ביצענו עבור כל מודל כוונון פרמטרים מקיף יותר.
להלן רשימת היפר-פרמטרים סופיים (בסוגריים ציינו את טווח החיפוש לכל היפר-פרמטר):

- **XGBoost** - קצב למידה 0.1 (0.05,0.1, 0.15, 0.2), עומק מקסימלי 8 (5-12), מספר עצים 50 (20-80 בקפיצות של 10).
- **יער אקראי** - מספר עצים 60 (20-80 בקפיצות של 10), ללא Bootstrap, עומק מקסימלי 9 (5-12) עם משקלי מחלקות 1:3.
- **עץ החלטות** - עומק מקסימלי 9 (5-12), קריטריון Gini (נבדקו Gini, Entropy, LogLoss) , משקלי מחלקות 1:2 (נבדקו 1:1 עד 1:5).
- **רשת נוירונים** במבנה של 2 שכבות חבויות בגדלים 64 ו-32, פונקציות אקטיבציה ReLu (נבדקו בנוסף גם Sigmoid ו-tanh), קצב למידה ראשוני 0.001 (0.0005,0.001, 0.0015, 0.002), אופטימייזר Adam ועצירה מוקדמת אל מול סט ולידציה בגודל 20% (נבדקו 10%,20%, 30%).
- **רגרסיה לוגיסטית** - מספר איטרציות 500 (200,500,700,1000), ללא קנסות (נבדקו l2 ,l1, ללא), משקלי מחלקות 1:4 (נבדקו 1:1 עד 1:5).

Train

Model	F1	Accuracy	ROC AUC
XGBoost	**0.9006**	**0.9945**	0.9246
Random Forest	0.8682	0.9927	0.9076
Decision Tree	0.8932	0.9939	**0.9307**
MLP	0.8588	0.9922	0.8971
Logistic Regression	0.6852	0.9830	0.8136

טבלה 2: מטריקות F1, דיוק ו-ROC AUC עבור כל מודל על סט האימון

בטבלה 2 ניתן לראות את המדדים השונים עבור כל מודל (נספח א'). התוצאות הן לאחר כוונון היפר פרמטרים לכל מודל. ה-XGBoost מציג את התוצאות הטובות ביותר מבחינת F1 ודיוק, אך עץ ההחלטות הוא המודל הטוב ביותר ע"פ ה-ROC AUC.

Evaluation

סט הבחינה נשמר בצד, ולא בוצע עליו שום ניסוי מעבר לבדיקה זו, לאחר בחירת המאפיינים וכוונון היפר-פרמטרים על סט האימון.

Test

Model	F1	Accuracy	ROC AUC
XGBoost	**0.7167**	**0.9976**	0.8896
Random Forest	0.6418	0.9966	0.8824
Decision Tree	0.6494	0.9966	**0.8984**
MLP	0.6391	0.9968	0.8602
Logistic Regression	0.3313	0.9915	0.7681

טבלה 3: מטריקות F1, דיוק ו-ROC AUC עבור כל מודל על סט הבחינה

ניתן לראות בטבלה 3 שמבחינת מדד ההערכה העיקרי שלנו, F1, מודל ה-XGBoost נותן את התוצאות הטובות ביותר בדומה לסט האימון. כשמסתכלים על מטריצות המבוכה כמפורט בהמשך העבודה, ניתן לראות שעץ ההחלטה מצליח לזהות מספר רב יותר של הונאות על חשבון מספר לא גדול באופן יחסי של זיהוי הונאות שגוי, מה שיכול להסביר את העובדה שערך ה-ROC AUC שלו הוא הגבוה ביותר.

מטריצות מבוכה

מטריצת המבוכה מחלקת את כל התצפיות ל-4 קטגוריות: הונאות שסווגו נכון (TP), הונאות שסווגו לא נכון (FP), עסקות לגיטימיות שסווגו נכון (TN), עסקאות לגיטימיות שסווגו לא נכון (FN). המטריקות השונות המשמשות להערכת המודל נבנות סביב ארבעת הערכים הללו.

		Predicted	
		Fraud	Non-Fraud
Real Values	Fraud	TP	FN
	Non-Fraud	FP	TN

XGBoost			
Train		Test	
6388	1118	1675	470
292	249708	854	552720

Random Forest			
Train		Test	
6134	1372	1646	499
489	249511	1338	552236

Decision Tree			
Train		Test	
6482	1024	1715	430
526	249474	1421	552153

MLP			
Train		Test	
5977	1529	1550	595
485	249515	1155	552419

Logistic Regression			
Train		Test	
4757	2749	1165	980
1620	248380	3722	549852

תרשים 6: מטריצות מבוכה לכל אחד מהמודלים על סט האימון והבחינה

מטריצות המבוכה מראות תמונה דומה למה שהמטריקות הראו בטבלאות 1 ו-2. עץ ההחלטות מצליח לזהות 35 הונאות יותר מה-XGBoost בסט הבחינה, אך על חשבון 567 עסקאות לגיטימיות שמסווגות כהונאות (False Alarm\Positive). באופן יחסי לכמות העסקאות הלגיטימיות בסט הבחינה, 553,574, מדובר על תוספת שיתכן ושווה את השיפור. נרחיב על כך בדיון.

סיכום, דיון ומסקנות

תהליך העבודה היה מורכב וכלל מספר רב של החלטות הקשורות בנתונים לא מאוזנים, מספר גדול מאוד של מאפיינים (בעלי קטגוריות מרובות) והרבה רשומות. גודל בסיס הנתונים הקשה על ביצוע ויזואליזציות שבאמצעותן ניתן להסיק מסקנות על טיב המאפיינים, לנתח חריגים ולבצע מספר גדול של ניסיונות אימון (בשל זמן האימון הארוך). על כן, השתמשנו ביוריסטיקות שונות על מנת לצמצם את הבעיה לסט מאפיינים ברור וקטן יחסית, שיאפשר גם הבנה בסיסית של קבלת ההחלטות של המודלים השונים, וצמצמנו את מספר הרשומות.

מבחינת התוצאות, למרות שנראה כי אלגוריתם XGBoost הוא בעל המדדים הטובים ביותר, אם נסתכל על מטריצות המבוכה נראה שמבחינת זיהוי הונאות (TP = True Positive) עץ ההחלטות הצליח במידה דומה ואף מעט טובה יותר, וזאת למרות שצורת הסיווג שלו פחות מורכבת. אך למרות זאת, ניתן לראות שהיו לו הרבה יותר התראות שווא (FP = False Positive). באחוזים, עץ ההחלטות מזהה 1.8% יותר הונאות (TP) על חשבון 0.1% תוספת של זיהויים שגויים של עסקאות (FP) לגיטימיות כהונאות, מה שעל פניו יכול להיחשב ל-Tradeoff משתלם, בהתחשב בעובדה שקבוצת המיעוט קטנה כל כך.

המצב מעלה את השאלה- האם f1 הוא הקריטריון המתאים ביותר להערכת איכות המודל בבעיה שלנו?

ובהקשר עסקי- האם אנו מוכנים לדייק יותר בזיהוי הונאות במחיר של התראות שווא ללקוחות האשראי?

במידה והיה מדובר בפרויקט אמיתי בשיתוף עם חברה שהייתה מספקת לנו נתוני אמת, כנראה שהיינו מציגים להם את הסוגיה (מבחינת השאלה העסקית שהצגנו), בוחרים את שני האלגוריתמים הללו וממשיכים לבדוק את דיוקם עם נתונים נוספים. זאת כדי לבדוק האם העובדה שהעץ סיווג טוב למרות שהוא מודל "פשוט" הופכת אותו למכליל יותר, שמזהה טוב יותר מקרים בהם יש הונאות. בנוסף, העץ יכול לתת הסבר ברור יותר לגורמים המתריעים על הונאה פוטנציאלית.

כיוון נוסף להמשך מחקר הוא אשכול. כפי שהצגנו ב-PCA בתרשים 5, יש חלוקה די ברורה לשישה אשכולות בעלי הפרדה משמעותית ביניהם. אפשר לשקול הוספה של מאפיין המציין את האשכול אליו העסקה משתייכת למודל, או להציע מסווגים שונים עבור האשכולות במידה ויש הבדלים בין המאפיינים הדומיננטיים בכל אשכול לבעיית הסיווג.

נספחים

נספח א' - נוסחאות לחישוב מטריקות

בפרוייקט השתמשנו בשלוש מטריקות להערכת המודלים. להלן החישוב שלהן.
הונאות שסווגו נכון (TP), הונאות שסווגו לא נכון (FP), עסקות לגיטימיות שסווגו נכון (TN), עסקאות לגיטימיות שסווגו לא נכון (FN).

$$f1 = \frac{2TP}{2TP+FP+FN}$$

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

חישוב ה-ROC AUC הוא חישוב שמתבצע בפונקציה המובנית של scikit, ומדובר על ביצוע אינטגרציה לעקומת ה-ROC.

ביבליוגרפיה

Abdulghani, A. Q., Uçan, O. N., & Alheeti, K. M. A. (2021, December). Credit Card Fraud Detection Using XGBoost Algorithm. In *2021 14th International Conference on Developments in eSystems Engineering (DeSE)* (pp. 487-492). IEEE.

Raj, S. B. E., & Portia, A. A. (2011, March). Analysis on credit card fraud detection methods. In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET) (pp. 152-156). IEEE.

Woolston, S. E. (2017). Machine-learning Methods for Credit Card Fraud Detection. California State University, Long Beach.