אוניברסיטת בן-גוריון בנגב | המחלקה להנדסת תעשייה וניהול

פרויקט כריית נתונים - MLDM

איתור לקוחות לתוכניות חיסכון בבנק

עומר קידר | איל גינוסר

תקציר

אחד ממקורות ההכנסה העיקריים של בנקים הוא פיקדונות פיננסים. מאחר וכיום לקוחות כבר אינם ממהרים להפקיד את כספם בבנק כמו בעבר, הבנקים פועלים רבות על מנת לשווק מוצר זה ולמכור אותו ללקוחותיהם. אחת השיטות הנפוצות ביותר לביצוע משימה זו היא על ידי שיחות טלפון ללקוחות הבנק. אולם, מרכזי מכירות טלפונים עולים הרבה מאוד כסף לבנק, ואחוז הלקוחות שלבסוף מכניסים כספים לפיקדונות אינה גבוהה ביחס לכמות השיחות שמבוצעות. עובדה זו מעודדת את הבנקים למצוא פתרונות חדשניים לצמצום ההוצאות על מרכזי מכירות אלו.

בעבודה זו, פיתחנו מערכת המלצה אשר משתמשת בכלי לימוד מכונה בכדי להעריך אילו לקוחות יסכימו לבצע הפקדה בפיקדון בעקבות שיחת מכירה שיקבלו מנציג הבנק. את המערכת לימדנו והערכנו באמצעות נתונים על לקוחות של בנק פורטוגלי שנאספו במשך מספר חודשים. כדי למצוא את המודל הטוב ביותר עבורנו, ביצענו השוואה של מספר אלגוריתמי לימוד מכונה (ANN, RF, Xgboost, SVM) והשוונו ביניהם על פי הערכה של מדד הG-Mean, על מנת למצוא את המודל שיבטיח לנו איתור של מספר הלקוחות שיבצעו הפקדה בעקבות שיחה הגדול ביותר, תוך שמירה על כמות כוללת של שיחות שיעשו קטנה ככל האפשר.

תוצאות הניסוי שלנו מצאו כי מודל Random Forest (RF) הוא המודל האופטימאלי עבור מערכת ההמלצה שלנו, אשר הגיע לתוצאה של 93.1% במדד G-Mean. בנוסף, מצאנו כי שינוי של סף ההחלטה של המודל משפר רבות את התוצאות כאשר הדאטה אינו מאוזן וקיים שוני גדול בהתפלגות בין שתי מחלקות הסיווג, כפי שהיה בנתונים שלנו.

עבודתנו תורמת עם מסקנות על אילו נתונים הם המשמעותיים ביותר עבור בניית מודל סיווג מסוג זה עבור בנקים, אשר נמצאו על ידי מודל הRF שבנינו ועל ידי ניתוח הנתונים שביצענו. בנוסף, פיתחנו מערכת המלצה אשר באמצעותה הבנק יכול לחתוך את כמות שיחות הטלפון שעליו לבצע בכ-80%, תוך שמירה על הגעה לכ-97% מהלקוחות אשר יבצעו הפקדה לפיקדון בעקבות שיחת מכירה.

תוכן עניינים

1. מבוא והבנת הבעיה העסקית

הבנקים השונים ברחבי העולם מספקים שירותים רבים ומגוונים ללקוחותיהם. אחת הפעולות המסורתיות והיסודיות ביותר המהווה את המקור העיקרי למימון בנקאי היא פיקדונות בנקאיים , כאשר חלק מסוגי הפיקדונות מהווים את מקור האשראי והרווח היציבים ביותר לבנקים [8] .

המשבר הכלכלי העולמי שפקד את העולם בשנת 2008 יצר משבר אמון בין הלקוחות לבנקים וגרם לחשדנות רבה אשר גרמה לצמצום הפיקדונות הקיימים והחדשים בבנקים בצורה משמעותית. בנוסף, ההתפתחות המהירה של שוק ההון, תיווך פיננסי ומכשירים פיננסיים המספקים אפיקי השקעה והזדמנויות חדשות ללקוחות, גרמו לכך שהם לא ממהרים להפקיד את כספם בידי הבנקים. סיבות אלו בנוסף ללחץ כלכלי ותחרות בין הבנקים, יוצרים את הצורך של הבנקים למצוא פתרונות איכותיים לשיפור האפקטיביות של הקמפיינים השיווקיים שלהם בכלל ושל הקמפיינים לשיווק הפיקדונות בפרט [9].

מטרת הבנקים היא ל"טרגט" לקוחות אשר ירצו להירשם לתוכנית פיקדון ולהפקיד את כספם באחת מתוכניות הבנק תוך שמירה על מודל כלכלי נכון , כלומר חיסכון בשעות עבודה וצמצום כמות השיחות ללקוחות , צמצום כמות העובדים , צמצום ההוצאות ומקסום ההכנסות מהפיקדונות. קיימים אתגרים בביצוע "טירגוטים" אלו כמו מחסור בנתונים , התיישנות נתונים וידע חסר לגבי הגורמים המשפיעים על לקוחות לבצע פיקדונות [5,10]. אנו נרצה בפרויקט זה לבנות מערכת תומכת החלטה (DSS) [6] אשר תוכל לעזור לבנקים לעמוד במטרות שהצגנו, כאשר המטרה הראשית היא לחזות אילו לקוחות צפויים לבצע פיקדון ולצמצם את כמות הלקוחות הפוטנציאליים שמתפספסים תוך כדי הבנת מאפייני הלקוחות אשר משפיעים על הכנסת כספים לפיקדון על מנת שהבנקים יוכלו להשקיע בתחומים אלו, אשר יעודדו בעתיד את הגדלת כמות הלקוחות שירצו לבצע פיקדונות.

2. הבנת הנתונים

בחלק זה נציג את סט הנתונים שלנו, נפרט על תכונותיו, איכותו ונסביר את הקשרים בין המשתנים המסבירים למשתנה המוסבר ובין המשתנים המסבירים לבין עצמם. שלב זה הוא שלב קריטי ביותר לקראת הכנת הנתונים לאימון ובחינה , זהו שלב בו אנו מאתרים בעיות בדאטה כמו נתונים חסרים, חוסר איזון או ערכים חריגים ומוצאים את הדרך הטובה ביותר להתמודד איתן.

2.1 תיעוד מקור הנתונים ומשמעותם

הנתונים נאספו ממאגר המידע של מוסד בנקאי פורטוגלי ומכיל מידע בין השנים 2008-2010. המידע מכיל 16 מאפיינים (רציפים וקטגוריאליים) והנתונים נועדו כדי להבין אם אדם יסכים להכניס כסף לפיקדון בבנק ואם כן , אילו מאפיינים משפיעים על כך. אוסף הנתונים שלנו מכיל משתנים כמו גיל, מצב משפחתי, מאזן כספי, הלוואות קיימות ועוד. משתנה המטרה שלנו מוגדר להיות '0' כאשר הלקוח לא רוצה להכניס כסף לפיקדון בבנק ו '1' כאשר הוא מעוניין להכניס כסף לפיקדון. סט הנתונים שברשותנו מכיל 45,211 רשומות בסט האימון ו 4521 רשומות בסט הבחינה.

	משתנה	סוג	פירוט	טווח ערכים
1	Age	נומרי	גיל הלקוח	18-95
2	Job	קטגוריאלי	סוג העבודה של הלקוח. משתנה קטגוריאלי בעל 7 ערכים אפשריים	
3	Marital	קטגוריאלי	מצב משפחתי. רווק\ה, נשוי\ה, גרוש\ה	
4	Education	קטגוריאלי	רמת ההשכלה	
5	Default	בינארי	האם יש ללקוח קרדיט	
6	Balance	נומרי	מאזן שנתי ממוצע של הלקוח	-8019 - 102127
7	Housing	בינארי	האם יש ללקוח הלוואה על בית	
8	Loan	בינארי	האם יש ללקוח הלוואה אישית	
9	Contact	קטגוריאלי	סוג ההתקשרות עם הלקוח (לא ידוע\ טלפון קווי\ סלולרי)	
10	Day	נומרי	היום בחודש בו בוצעה ההתקשרות האחרונה עם הלקוח	1-31
11	Month	נומרי	החודש האחרון בו בוצעה התקשרות עם הלקוח	
12	Duration	נומרי	הזמן שערכה ההתקשרות האחרונה עם הלקוח	0-4918
13	Campaign	נומרי	מספר ההתקשרויות שבוצעו עם הלקוח במהלך קמפיין המכירות	1-63
14	Pdays	נומרי	מספר הימים שחלפו מאז ההתקשרות האחרונה עם הלקוח בקמפיין קודם	-1 -871
15	Previous	נומרי	מספר ההתקשרויות שבוצעו עם הלקוח בקמפיינים קודמים	0-275
16	poutcome	קטגוריאלי	תוצאה של הקמפיין הקודם עבור הלקוח (לא ידוע\הצלחה\כישלון)	

2.2 איכות הנתונים

בחנו את הנתונים באמצעות שימוש בויזואליזציות וביצענו בדיקות לכפילויות וחוסרים וראינו כי אין בדאטה שלנו נתונים חסרים או כפולים וכי אין נתונים חריגים בדאטה שלנו אך יש מספר משתנים בהם מופיע הערך המיוחד "Unknown" (נפרט על המשתנים וכמות הרשומות בטבלה בהמשך). יש מספר דרכים להתמודד עם ערכים חסרים או לא ידועים כמו למשל מחיקת רשומות או פיצ'רים , השלמה על-פי ממוצע או חציון, שימוש ב KNN להשלמת הנתונים ועוד. תיאור ודוגמאות של הויזואליזציות של הנתונים ניתן לראות בנספח 1.

2.3 הסתברויות אפריוריות

על מנת שנוכל להציג את ההסתברויות האפריוריות של הנתונים שלנו בצורה נוחה ומובנת , אשר תאפשר לנו לחלץ מידע רלוונטי , בחרנו לחלק את הפיצ'רים הנומרים שלנו לקבוצות על-פי היסטוגרמות ו Box-plots של הנתונים שהוצאנו. בתמונה למטה ניתן לראות דוגמה של חלק מן המשתנים. תמונה של שאר המשתנים ניתן לראות בנספח 2.

Age		Job		Marital		Education		Default		Balance		Housing		Loan	
Range	%	Class	%	Class	%	Class	%	Class	%	Range	%	Class	%	Class	%
18-30	15%	Admin	11%	Married	12%	Primary	15%	Yes	2%	[-8019-0)	16%	Yes	56%	Yes	16%
31-50	64%	Blue-collar	22%	Single	60%	Secondary	51%	No	98%	[0-10,000)	82%	No	44%	No	84%
51-70	19%	Enterpreneur	3%	Divorced	28%	Tertiary	29%			[10,000-20,000)	1%				
70-95	1%	Housemaid	3%			Unknown	4%			[20,000-30,000)	<1%				
		Management	2%							[30,000-40,000)	<1%				
		Retired	5%							[40,000-50,000)	<1%				
		Self-emloyed	3%							[50,000-60,000)	<1%				
		Services	9%							[60,000-70,000)	<1%				
		Student	2%							[70,000-80,000)	<1%				
		Technician	17%							[80,000-90,000)	<1%				
		Unemployed	3%							[90,000-100,000)	<1%				
		Unknown	<1%							[100,000-110,000)	<1%				

טבלאות אלו עזרו לנו רבות בשלב הכנת הנתונים המפורט בהמשך. אחד הדברים העיקריים שניתן לראות כאן הם הפיצ'רים אשר בהם היו ערכים של "Unknown". אותם ערכים לא ידועים לא מוסיפים לנו מידע חיוני ולכן התייחסנו אליהם כנתונים חסרים והחלטנו לבצע השלמה על-ידי שימוש ב KNN בפיצ'רים בהם אחוז הלא ידועים היה נמוך יחסית. בפיצ'ר Poutcome ניתן לראות שאחוז הלא ידועים מהווה את הרוב, לכן על-פי קריאה בספרות החלטנו כי לא יהיה נכון להשלים את הערכים הללו על פי מחלקות המיעוט של הפי'צר משום שהדבר יכול ליצור הטעיות ולכן החלטנו בכל זאת להשתמש בפיצ'ר זה כפי שהוא.

2.4 קורלציות

השלב האחרון בהבנת הנתונים היה לבדוק את רמת ההתאמה בין המשתנים. עניין אותנו לראות אילו משתנים מסבירים בצורה טובה את משתנה המטרה שלנו ואלו לא ובנוסף לבדוק האם יש משתנים מסבירים עם קורלציה גבוהה מאוד ביניהם. את הבדיקה הראשונית ביצענו על סט הנתונים המקורי ואת הקורלציות ריכזנו בטבלת מתאם (heatmap) אשר ניתן לראות בנספח 3.א. הקורצליות נבדקו בצורה הבאה: קורצליה של משתנים רציפים מול משתנים רציפים נבדקה על ידי מתאם פירסון. קורצליה של משתנים קטגוריאלים מול משתנים קטגוריאלים נבדקה על ידי מתאם קרמר (Cramer's V), וקורצליה בין משתנים רציפים לקטגוריאלים נבדקה על ידי יחס התאמה (Correlation Ratio). התוצאות מראות באופן כללי כי משתנה המטרה אינו מוסבר בצורה גבוהה על ידי המשתנים המסבירים.

2.4.1 קשרים מעניינים בין המשתנים המסבירים למוסבר

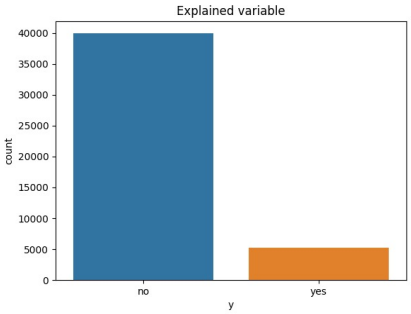
המשתנים המסבירים הטובים ביותר הם duration עם 0.39, poutcome עם 0.31 ו- month עם 0.26. שאר המשתנים הניבו תוצאות של 0.15 או פחות. המשתנים הגרועים ביותר מבחינת התאמה עם משתנה המטרה הם default עם 0.02, age עם 0.03 ו- day עם 0.03. לאור תוצאות אלו, החלטנו לבצע איחוד קטגוריות במשתנים על פי היסטוגרמות ולבחון את הקורצליות מחדש. הנחנו למשל, שצמצום המשתנה age לקבוצות גילאים תניב תוצאות טובות יותר, שכן יש השפעה גבוהה יותר לקבוצת הגיל בה אדם נמצא על מידת הסיכוי שאדם ינעל את כספו בפיקדון (למשל: צעיר או פנסיונר) לעומת הגיל המדויק. בדיקה חוזרת של טבלת הקורלציות לאחר השינוי הציגה שיפור משמעותי במשתנים age עם 0.13 לעומת 0.03, ו- pdays עם 0.24 לעומת 0.1 (מפת קורלציה לאחר איחוד קטגוריות ניתן לראות בנספח 3.ב).

2.4.2 קשרים מעניינים בין המשתנים המסבירים:

מבחינת הקורלציות בין המשתנים השונים, ראינו כי הקורצליה הגובה ביותר הייתה 0.61 בין בין poutcome לבין pdays ובכל שאר המשתנים הקורצליות היו נמוכות יותר וברובם נמוכות מאוד. החלטנו בשלב זה שקורלציות אלו לא גבוהות בצורה כזו שמספיקה לוותר על אחד המשתנים ובנוסף אין קשר הגיוני כל כך בניהם, לכן החלטנו שלא להסיר אף אחד מהם.

2.5 האם סט הנתונים מאוזן ומייצג את המציאות :

סט נתונים	y – משתנה המטרה	הסתברות אפריורית	כמות רשומות
Train	לא יכניס כסף לפיקדון	88.3%	39,922
	יכניס כסף לפיקדון	11.7%	5289
Test	לא יכניס כסף לפיקדון	88.47%	4000
	יכניס כסף לפיקדון	11.53%	521



סט הנתונים שלנו לא מאוזן כלל וכפי שניתן לראות, 88.3% מכלל הרשומות בסט הנתונים שייכות ללקוחות שלא ירצו להכניס את כספם לפיקדון ('0') ורק 11.7% מכלל הרשומות הם לקוחות אשר ירצו להכניס את כספם לפיקדון ('1'). ניתן לראות כי גם סט האימון וגם סט הבחינה לא מאוזנים בצורה כמעט זהה, דבר שלא ייצור לנו הטעיות בתוצאות כאשר נבצע בחינה למודל שלנו על סט הבחינה.
בנוסף, על פי הנתונים באתר FDIC - Federal Deposit Insurance Corporation מצאנו כי הנתונים משקפים את המציאות ולכן כפי שנרחיב בהמשך, בחרנו להשתמש בשיטת הורדת סף (Threshold) על מנת להתמודד עם חוסר האיזון בנתונים ולא בחרנו בשיטות כמו Upsampling ו Downsampling שלדעתינו ולפי מידע שמצאנו בספרות יובילו לתוצאות פחות טובות.

3. הכנת הנתונים

3.1 איחוד קטגוריות

בשלב הבנת הנתונים בחנו את מידת ההתאמה של המשתנים המסבירים למשתנה המוסבר. אחת הפעולות אשר ביצענו מעבר לכך הייתה איחוד קטגוריות של המשתנים הרציפים על מנת לבחון את ההשפעה על מידת ההתאמה, כיוון שהיא הייתה נמוכה עבור כל המשתנים. באופן כללי, איחוד הקטגוריות הטיב עם רוב המשתנים שלנו, למעט previous ו- campaign. לכן, עבור בניית המודלים שלנו, ביצענו איחוד קטגוריות עבור המשתנים: age, balance, day, month, duration, pdays.

3.2 השמטת נתונים

לאחר בחינת הנתונים וההתאמות בין המשתנים בחרנו להסיר מן הנתונים את המשתנים הבאים: **Default** - מידת ההתאמה שלו היא 0.02 בלבד. נתון זה אינו מפתיע כיוון שזהו משתנה בינארי עם התפלגות של 98% עבור 'לא' ו2% עבור 'כן', כלומר, הוא אינו נותן לנו הרבה מידע. **Day** - מידת ההתאמה עם משתנה המטרה הייתה 0.3 וגם על ידי איחוד קטגוריות עלתה רק ל 0.6. לעומת החודש בשנה אשר הראה התאמה טובה יחסית, היום בשבוע אינו מסביר טוב את משתנה המטרה. אנו מניחים כי לעומת חודשי השנה, אשר מציגים הבדלים גדולים ביניהם כמו עונות, תקופות של חגים ועוד, ליום בשבוע בו נעשית שיחת המכירה יש הרבה פחות משמעות.

3.3 ONE-HOT-ENCODING

בחרנו להשתמש בשיטת one-hot-encoding בכדי למדל את המשתנים הקטגוריאליים שלנו עם משתני דמה.

3.4 השלמת ערכים חסרים

על אף שסט הנתונים לא הגיע עם ערכים חסרים, בחרנו להתייחס אל הערך "Unknown" בתור ערך חסר. לראות עינינו, זו תהיה טעות לאמן מודל אשר ייתן משקולות לערך אינו ידוע, כיוון שדבר זה לא יתרום לאמינות המודל. בחרנו להשתמש במודל השלמת ערכים חסרים על ידי KNN כאשר n=5 כיוון שהוא מתאים ללמידה מונחת ונותן ביצועים טובים עבור דאטה שבו אין קורלציה גבוהה בין המשתנים (כמו במקרה שלנו) [1,3]. ביצענו השלמת ערכים עבור הפיצ'רים job, education, contact.

כיוון שב- poutcome אחוז הערכים החסרים היה גבוהה מאוד (מעל 80%), בחרנו לא לבצע עליו השלמת ערכים חסרים ולהשאיר בו את הערך "unknown".

3.5 חלוקת הנתונים

הנתונים שלנו הגיעו עם קובץ train וקובץ test. לצורך בניית המודלים וכיוונון הפרמטרים שלהם, ביצענו חלוקה של קובץ ה-train לסט אימון (train set) וסט בחינה (validation set). את החלוקה ביצענו על פי שיטת holdout, כאשר 80% מסט הנתונים הוא עבור סט האימון ו20% לסט הבחינה. בחרנו להשתמש בשיטה זו על פני שיטת Kfold כיוון שסט הנתונים שלנו מאוד לא מאוזן, ובמצב כזה שימוש בחלוקה על פי Kfold עלול ליצור הטעיה בתוצאות [4] . דבר זה עלול לקרות כתוצאה מכך שישנה סבירות גבוהה שבפולדים (fold) רבים לא יהיו סמפלים של מידע שמייצג את הקלאס הקטן (כיוון שהוא רק 12% מתוך כלל הנתונים). על אף ששימוש בחלוקה עם stratify מאפשר חלוקה של הדאטה לסט אימון וסט בחינה עם התפלגות זהה של הקלאסים בשני הסטים ופותר את בעיה זו, ביצענו השוואה בין ולידציה על פי Kfold וholdout וראינו כי התוצאות כמעט זהות לחלוטין. לכן, העדפנו להשתמש בשיטת holdout עם stratify כיוון שהוא עדיף מבחינת זמני ריצה.

4. מידול

4.1 מדדים

Confusion Matrix

Actual \ Predicted	0	1
0	TN	FP
1	FN	TP

הבעיה שלנו עוסקת בחיזוי של לקוחות פוטנציאלים אשר יסכימו להצעה להפקדת פיקדון כתוצאה משיחת מכירה מהבנק. ניתן לומר כי במצב הקיים (ללא מודל סיווג) על סמך ההתפלגות הנאמדת מן הנתונים שבידינו, אם לבנק יש לדוגמה 10,000 לקוחות פוטנציאלים, הבנק יאלץ לשלם עבור ביצוע של 10,000 שיחות טלפון שיניבו כ-1200 הפקדות בלבד. מטרתנו אם כך היא לאתר ככל הניתן את 1200 הלקוחות אשר יבצעו הפקדה (מקסימום עסקאות), תוך צמצום מספר השיחות הכולל שיש לבצע (שיפור יחס עלות-תועלת). במצב כמו שלנו בו הדאטה כלל לא מאוזן ואנו רוצים להגיע לדיוק גבוה על המחלקה הקטנה, מדד ה Accuracy אינו רלוונטי כלל. לכן, המדד העיקרי שלפיו החלטנו להעריך ולכוונן את המודלים הוא מדד Geometric Mean Score - $\sqrt{TPR * TNR}$ כאשר $TPR = \frac{TP}{TP+FN}$ ו $TNR = \frac{TN}{TN+FP}$. השאיפה שלנו היא שהמודל יחזה נכון מקסימום לקוחות שירצו להכניס את כספם לפיקדון (TP), תוך שמירה על כמה שפחות חיזויים שגויים (FP). זה בדיוק מה שערך גבוהה של מדד זה מבטיח לנו, מקסום של ערכי ה- TP תוך שמירה על ערכי FP נמוכים ככל שניתן. מעבר לכך בדקנו לאורך כל העבודה גם את מדדי הAccuracy, Recall, Precision,F1 לצורך הבנה עמוקה יותר, אך הערכנו את כל המודלים על-פי G-mean.

4.2 התמודדות עם דאטה לא מאוזן

כפי שכבר ציינו, סט הנתונים שלנו איננו מאוזן (כ- 88% מקלאס 0 ו 12% מקלאס 1). כדי להתמודד עם בעיה זו, ולאחר קריאה מעמיקה על הדרכים להתמודדות עם דאטה לא מאוזן, החלטנו לבצע התאמה של סף ההחלטה (threshold) בכל מודל על מנת להתאים אותו לדאטה הלא מאוזן שלנו[1]. כאשר מבצעים סיווג בינארי (כפי שאנחנו עושים בעבודתנו) המודלים משערכים את ההסתברות של כל תצפית להשתייך לכל אחד מהקלאסים. מרבית המודלים מתבססים על כך שהנתונים מאוזנים (50% קלאס 0, 50% קלאס 1) וכך סף ההחלטה של סיווג המודל נקבע על 0.5 (בקירוב). כלומר, אם על פי המודל הסיכוי להשתייך לקלאס 0 עבור תצפית מסוימת הוא מעל 0.5, המודל יסווג את התצפית כקלאס 0 ואחרת ל 1. עבור מצב בו הנתונים אינם מאוזנים, שינוי של סף ההחלטה יכול להוביל לתוצאות גבוהות יותר במדדים השונים, כך שלכל מדד יש את הסף אשר ימקסם את ערכו. לכן, בכל מודל לאחר כיוונן הפרמטרים חישבנו את סף ההחלטה אשר יוביל לתוצאות הטובות ביותר במדד G-mean על-ידי שימוש ב ROC Curve וביצענו הערכה של המודל על פי סף זה.

[1] https://machinelearningmastery.com/threshold-moving-for-imbalanced-classification/

4.3 מודלים

4.3.1 ARTIFICIAL NEURAL NETWORKS

בחרנו לאמן מודל רשתות נוירונים כיוון שהוא הראה תוצאות טובות עבור בעיות מהסוג שאנו רוצים לפתור [2]. ביצענו נרמול מסוג MIN-MAX כאשר Min=0, Max=1 , בכדי שכל הפיצ'רים יהיו בעלי טווח ערכים זהה. פעולה זו חשובה עבור הצלחת מודל מסוג זה. לאחר מכן ביצענו בחינה ראשונית של הקלאסייר Multi Layer Perceptron עם ערכי ברירת המחדל שלו. זוהי רשת בעלת שכבה חבויה אחת עם 100 נוירונים. המודל הניב תוצאות סבירות על train וה-validation אך לא מספקות. לכן ביצענו כיוונון לפרמטרים של המודל גם בכדי להקטין את השונות בין תוצאות האימון והולידציה ובנוסף כדי לשפר את תוצאות המודל ככל הניתן.

Train G_Mean	Validation G_Mean
0.754	0.624

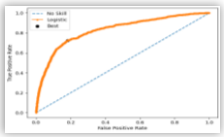
הפרמטרים שכוונו וערכיהם:

Hidden_layers	Solver	Activation	Learning_rate_init
(67,), (67, 67), (67, 67, 67), (140,), (140, 140), (140, 140, 140)	sgd, adam , lbfgs	tanh, relu , logistic	0.1, 0.01 , 0.001

המודל הטוב ביותר שהתקבל:

Hidden_layers	Solver	Activation	Learning_rate_init
(140,140)	adam	tanh	0.1

Train G_Mean	Validation G_Mean
0.768	0.751



ניתן לראות כי לאחר כוונון הפרמטרים הצלחנו לשפר את המודל בכ-12% עבור מדד geometric mean על סט הולדיציה, וכי השונות בין סט האימון והולידציה ירדה בצורה משמעותית. לאחר מכן חישבנו את ערך הסף (threshold) שיניב את ערך מדד G-mean הגבוה ביותר וביצענו בחינה של המודל על פי סף זה. ערך ה threshold שהתקבל הוא threshold=0.096420.

Train G_Mean	Validation G_Mean
0.778	0.77

Actual	Predicted 0	Predicted 1
0	6511	1474
1	289	769

תוצאות המודל שהתקבל לאחר כיוונון ה threshold מראות לנו כי יש שיפור במספר ה- TP שהמודל חוזה. שיפור זה מגיע על חשבון חיזוי גבוהה מאוד של FP ולכן ערכים כמו accuracy ו- precision יורדים ואנו מקבלים רק עליה קלה במדד ה- G-mean. פירוט מלא של תוצאות כל המדדים ומטריצות המבוכה לאורך תהליך כוונון המודל ניתן לראות בנספח 4.

4.3.2 RANDOM FOREST

מודל זה נפוץ מאוד לשימוש עם דאטה טבלאי אשר נותן תוצאות טובות לבעיות קלסיפיקציה. בנוסף, הוא מודל יחסית מהיר לאימון גם עם כמויות דאטה גדולות. מעבר לכך, הוא מתמודד היטב עם שוני בנתונים -אין לו בעיה עם משתנים קטגוריאלים והוא אינו מצריך נרמול של משתנים רציפים. תכונה חשובה נוספת של מודל זה הוא היכולת להבין מה הם הפרמטרים החשובים ביותר בהם המודל עשה שימוש , כלומר ניתן להבין מהם המאפיינים המשפיעים ביותר ועל-ידי כך לבצע שינויים עסקיים בהתאם לצורך וזה בדיוק נותן מענה לאחת הבעיות שאנו רוצים לפתור. בחינה ראשונית של המודל הניבה את התוצאות הבאות:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.926	0.576	0.909	0.419	0.86	0.345	0.964	0.534	0.98	0.888

ניתן לראות כי תוצאות המודל על סט האימון מצוינות אך לעומת זאת על בסט הולידציה יש ירידה משמעותית בדיוק המודל בכל המדדים. כלומר, המודל התאים את עצמו יותר מדי לסט האימון והגיע למצב של Overfitting ולכן יש לבצע כוונון פרמטרים כדי לשפר את התוצאות.

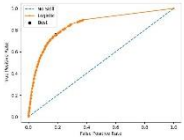
הפרמטרים שכוונו וערכיהם:

N_estimators	Criterion	Max_features
10 , 20 , 35 , 45 , 52 , 70 , 90 , 110 , 130 , 180 , 200	gini , entropy	15-45

המודל הטוב ביותר שהתקבל:

N_estimators	Criterion	Max_features
35	gini	26

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall
0.927	0.625	0.908	0.464	0.864	0.41



ניתן לראות שיפור בביצועי המודל על סט הולידציה בכ-7% במדד ה- G-mean ובהתאם גם מדד ה- recall השתפר. המודל עדיין מציג ביצועים ירודים על סט הבחינה ביחס לסט האימון. אנו מעריכים כי העובדה שסט הנתונים איננו מאוזן היא אחת הסיבות לתוצאות אלו ולכן ציפינו לראות שיפור משמעותי לאחר כיוונון של סף ההחלטה של המודל. לאחר בדיקה , סף ההחלטה אשר התקבל הוא threshold=0.118571.

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall
0.937	0.785	0.7	0.476	0.987	0.761

כפי שחשבנו, ניתן לראות כי כאשר אנו בוחנים את המודל עם סף החלטה שונה, ערך מדד G-mean עלה ב-16%. על אף שתוצאת הבחינה עדיין מעט רחוקה מתוצאת האימון, אנו מרוצים מביצועי המודל היות והם מעפילים על ביצועי רשת הנוירונים. פירוט מלא של תוצאות כל המדדים ומטריצות המבוכה לאורך תהליך כוונון המודל ניתן לראות בנספח 4.

4.3.3 XGBOOST

זהו מודל Gradient Boosting Machine. עקרון הפעולה של המודל הוא הרכבה של תוצאות חיזוי של מודלי עצי החלטה חלשים. בחרנו לבחון שימוש במודל זה כיוון שמודל RF שגם משתמש בעצי החלטה הגיע לתוצאות טובות ומודל XGboost לעיתים מעפיל על תוצאות Random Forest. בנוסף, מודל זה זוכה להערכה רבה כיום בבעיות data science והשתמשו בו בהרבה מודלים שניצחו בתחרויות קאגל שונות[2].

Train G_Mean	Validation G_Mean
0.714	0.6

ביצענו בחינה ראשונית למודל וראינו כי לעומת מודל RF, השונות בין תוצאות האימון לולידציה נמוכה יותר, אולם עדיין קיימת. כפי שראינו כבר במודלים הקודמים, כוונון הפרמטרים עוזר בהקטנת השונות ולכן ביצענו זאת גם פה.

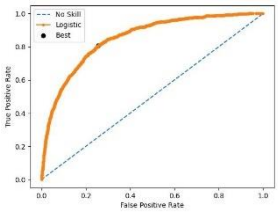
הפרמטרים שכוונו וערכיהם:

Booster	Eta	Max_depth	Min_child_weight
gbtree, dart	0.01, 0.1, 0.2 , 0.3 , 0.4 , 0.5 ,0.6 ,0.7 ,0.8 ,0.9 , 1	4, 6 , 8 , 10 , 12 , 14 ,16 ,18 ,20	1, 2, 3

המודל הטוב ביותר שהתקבל:

Train G_Mean	Validation G_Mean
0.877	0.62

Booster	Eta	Max_depth	Min_child_weight
gbtree	1	10	2



באופן מעט מפתיע, חל שיפור ב-2% בלבד עבור מדד G-mean על סט הולידציה, והשונות עלתה במקום להצטמצם. עם זאת, ראינו גם במודל הRF כי מה שמוביל לשינוי משמעותי יותר בתוצאות של מדד G-mean הוא דווקא שינוי הסף ולא כוונון הפרמטרים. תוצאת חישוב הסף האופטימאלי עבור מודל זה היא threshold=0.032809 .

[2] https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/.

תוצאות המודל הסופי, כפי שציפינו, בהחלט השתפרו. הן מבחינת טיב המדד והן מבחינת הקטנת השונות, מספר הTP עלה בצורה משמעותית (כמעט פי 2) , אך מספר ה FP עלה בצורה דרסטית , כמעט פי 6 ולכן המודל לדעתנו פחות טוב מ RF.

Train G_Mean	Validation G_Mean
0.873	0.775

		Predicted	
		0	1
Actual	0	5946	2039
	1	205	853

פירוט מלא של תוצאות כל המדדים ומטריצות המבוכה לאורך תהליך כוונון המודל ניתן לראות בנספח 4.

4.3.4 SVM

החלטנו לבחון מודל נוסף , מודל ה Support Vector Machine אשר ידוע כמודל טוב למשימות סיווג [7]. אלגוריתם זה מוצא גבול החלטה כקו או כמישור על מנת להפריד בין המחלקות השונות. בנוסף אלגוריתם זה ידוע כאלגוריתם שיודע להתמודד טוב עם כמויות גדולות של נתונים ולכן החלטנו להשתמש בו.

בדיקה ראשונית של המודל מראה כי התוצאות שהתקבלו נמוכות מאוד אם כי נראה כי אנו לא נמצאים במצב של Overfitting. כמו בשאר המודלים, ביצענו כיוונון פרמטרים כדי להגיע לתוצאות טובות יותר. הפרמטרים אשר כוונו וערכיהם:

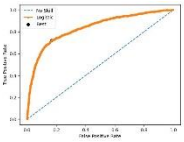
Train G_Mean	Validation G_Mean
0.558	0.546

Kernel	C
linear, poly , rbf , sigmoid	0.01, 0.025 , 0.1 , 0.5 , 1 , 3 , 10 , 100

המודל הטוב ביותר שהתקבל ותוצאותיו:

Kernel	C
rbf	100

Train G_Mean	Validation G_Mean
0.716	0.59



ניתן לראות שהמודל השתפר לאחר כיוונון הפרמטרים אך התוצאות על סט הולידציה עדיין היו נמוכות יחסית. רצינו לבצע שיפור נוסף ולכן ביצענו כיוונון לסף כפי שעשינו במודלים הקודמים. הסף שהתקבל הוא threshold = 0.100075.

בתוצאות המודל הסופי ניתן לראות כי היה שיפור של כ 18% בדיוק של פרמטר ה G-Mean ובנוסף שיפור משמעותי ב recall שכמובן בא על חשבון ה precision. שיפור זה תואם את השיפורים שקיבלנו גם במודלים הקודמים לאחר כיוונון הסף.

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.848	0.774	0.555	0.482	0.86	0.72	0.362	0.362	0.839	0.819

פירוט מלא של תוצאות כל המדדים ומטריצות המבוכה לאורך תהליך כוונון המודל ניתן לראות בנספח 4.

בסך הכל אנו רואים כי ברוב המודלים אנו רואים שיפור משמעותי על סט הולידציה כאשר עשינו שימוש בכיוונון הסף. כיוונון הסף הוכיח את חשיבותו הרבה במצב של דאטה לא מאוזן והניב תוצאות טובות באופן משמעותי.

5. הערכה

לאחר שהגענו למודל אופטימאלי של כל אחד מארבעת המודלים אשר אימנו, ביצענו השוואה ביניהם על סמך התוצאות על סט הבחינה (test). מודל ה random forest, אשר היה המודל עם הביצועים הטובים ביותר על סט הולידציה, הגיע לביצועים הטובים ביותר גם בסט הבחינה. מעבר לתוצאות העדיפות של המודל במדד G-Mean, הוא גם הגיע לתוצאות הטובות ביותר במדדי F1, Precision ו Accuracy. מעניין לראות כי התוצאות שהתקבלו על סט הבחינה גבוהות בצורה משמעותית מאלו שקיבלנו בסט הולידציה. אנו חושדים כי הסבר אפשרי לתופעה זו היא שסט הבחינה מכיל באופן מלא או חלקי נתונים אשר קיימים בסט האימון.

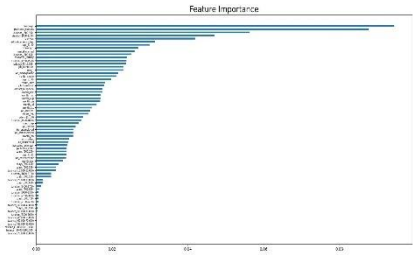
Test Scores					
Model	G_Mean	F1	Recall	Precision	Accuracy
ANN	0.794	0.471	0.793	0.335	0.795
Random Forest	0.931	0.693	0.973	0.538	0.9
Xgboost	0.865	0.512	0.99	0.345	0.782
SVM	0.833	0. 523	0.848	0.378	0.822

		Predicted	
		0	1
Actual	0	3564	436
	1	14	507

5.1 שיפור המודל

בחרנו לבחון אפשרות של שיפור המודל הנבחר כדי להגיע לתוצאות טובות יותר. ביצענו שתי שיטות: הכנת הנתונים מחדש ושימוש במודל ensemble.

5.1.1 הכנת הנתונים מחדש



השתמשנו בהיסטוגרמה של חשיבות המאפיינים שהוצאנו ממודל הRF כדי לראות אילו פי'צרים הם בעלי משמעות נמוכה למודל והסרנו אותם מהנתונים. אימנו מחדש את המודל וקיבלנו תוצאה של 0.917 במדד G-mean. כאמור, תוצאה זו אינה עדיפה על המודל המקורי.

5.1.2 מודל ENSSEMBLE

היות וניסיון השיפור הראשון שלנו לא צלח ניסינו לשפר את המודל על ידי יצירת מודל ensemble שמשתמש בהכרעת הרוב של כל ארבעת המודלים. בנינו את המודל כך שבמידה ושני מודלים לפחות חזו דגימה כ-1 היא תקבל 1, ואחרת 0. בצורה זו יצרנו תיעדוף להקטנת FN. למרבה הצער, מודל זה לא הצליח לשפר את תוצאות מודל הRF ונתן תוצאה של 0.898 בלבד.

6. סיכום, דיון, ומסקנות

מטרתנו בפרויקט הייתה לאתר לקוחות אשר יסכימו להפקיד את כספם לתוכניות חיסכון בבנק ולנסות להבין את המאפיינים המגדירים בצורה הטובה ביותר לקוחות כאלו. על-מנת לעשות זאת, עשינו שימוש במתודולוגיית CRISP-DM. בתחילה חקרנו את הבעיה העסקית על-ידי קריאת מאמרים ומחקרים בנושא איתור לקוחות, ביצענו ניתוחים שונים לדאטה ובנינו מודלים בהתאם ולבסוף הערכנו את המודלים וביצענו שיפורים על מנת להגיע לתוצאות הטובות ביותר.

בעבודתנו על הפרויקט נתקלנו בדאטה לא מאוזן בצורה חריגה, אשר הוביל אותנו לחקר מעמיק בנושא על מנת למצוא את הפתרון המתאים ביותר לבעיה שלנו שיאפשר לנו להתמודד עם חוסר האיזון בצורה הטובה ביותר. מצאנו כי הדרך הנכונה לנו ביותר היא שימוש בכוונון הסף (threshold). עשינו בכך שימוש בכל אחד מהמודלים שבנינו וראינו שיפור ניכר בכל אחד מהמודלים.

במהלך הפרויקט בחרנו לבחון ארבעה מודלים – Random Forest , ANN , Xgboost ו SVM. עבור כל אחד מהמודלים בוצע תחילה אימון בערכים דיפולטיביים של המודל, לאחר מכן בוצע כיוונון פרמטרים במטרה לשפר את המודלים ולבסוף בעזרת שימוש ב ROC Curve מצאנו את הסף הטוב ביותר למודל שיניב את התוצאות הטובות ביותר וביצענו אימון מחדש.

בגלל ההתמודדות עם הדאטה הלא מאוזן, הבנו כבר בהתחלה כי שימוש במדד Accuracy להערכת המודלים אינו רלוונטי כלל ולכן חקרנו ובדקנו מה המדד המתאים ביותר למטרות שלנו. גילינו כי המדד המתאים לנו ביותר הוא מדד ה G-mean score ועשינו בו שימוש על מנת להעריך את טיב המודלים. מצאנו כי המודל הטוב ביותר היה RF שהניב דיוק של 0.93 על סט הבחינה.

לאחר שמצאנו את המודל הטוב ביותר ניסינו לבצע מספר שיפורים כדי להגיע לתוצאות טובות יותר. ניסינו לבצע שינויים בסט הנתונים שלנו ולבצע בחינה מחודשת של המודלים על הסט החדש ובנוסף ניסינו לעשות שימוש ב Ensemble (הכרעת הרוב) בין ארבעת המודלים שבנינו בעצמינו. בשני המקרים הניסיון לשיפור כשל והגענו לתוצאות אומנם קרובות יחסית אך נמוכות מהמודל המקורי ולכן נשארנו איתו.

המסקנות שלנו מהפרויקט היו שעל מנת לשפר בצורה משמעותית את הדיוק על המחלקה הקטנה (קבוצת הלקוחות שירצו לבצע הפקדה לפיקדון) נדרש ניתוח מעמיק של המאפיינים שמגדירים לקוחות כאלו משום שראינו שאין קורלציות גבוהות בין הפיצ'רים שלנו למשתנה המטרה. לצרוך כך ניתן לעשות שימוש גם בפיצ'רים הטובים ביותר (על פי מודל ה RF) וגם לנתח את הלקוחות ולהוציא פיצ'רים חדשים שיוכלו לעזור להכין סט נתונים עדכני ואינפורמטיבי יותר שיעזור בשיפור התוצאות. בנוסף ראינו שכיוונון הסף הוא פתרון מעולה לבעיות עם דאטה לא מאוזן בצורה משמעותית ומניב שיפור משמעותי למודלים.

7. ביבליוגרפיה

[1] Gustavo E.A.P.A. Batista and Maria Carolina Monard. 2003. An analysis of four missing data treatment methods for supervised learning. *Appl. Artif. Intell.* 17, 5–6 (2003), 519–533. DOI:https://doi.org/10.1080/713827181

[2] Elsayad M. Alaa Elsalamony A. Hany. 2018. Bank Direct Marketing Based on Neural Network. *Adv. Energy Mater.* 8, 25 (2018), 1–9.

[3] Alireza Farhangfar, Lukasz Kurgan, and Jennifer Dy. 2008. Impact of imputation of missing values on classification error for discrete data. *Pattern Recognit.* 41, 12 (2008), 3692–3705. DOI:https://doi.org/10.1016/j.patcog.2008.05.019

[4] Yunqian Ma Haibo He. 2013. *Imbalanced Learning: Foundations, Algorithms, and Applications*. John Wiley & Sons. Retrieved July 15, 2021 from https://books.google.co.il/books?hl=iw&lr=&id=CVHx-Gp9jzUC&oi=fnd&pg=PT9&dq=Imbalanced+Learning:+Foundations,+Algorithms,+and+Applications&ots=2iMkHqAxak&sig=pWbgyV-IYJ7pwKJMvR9bUwnTM3E&redir_esc=y#v=onepage&q=Imbalanced Learning%3A Foundations%2C Algorithms%2C and Applications&f=false

[5] Vikas Jayasree and Rethnamoney Vijayalakshmi Siva Balan. 2013. A review on data mining in banking sector. *Am. J. Appl. Sci.* 10, 10 (2013), 1160–1165. DOI:https://doi.org/10.3844/ajassp.2013.1160.1165

[6] Stefan Lessmann, Johannes Haupt, Kristof Coussement, and Koen W. De Bock. 2021. Targeting customers for profit: An ensemble learning framework to support marketing decision-making. *Inf. Sci. (Ny).* 557, (2021), 286–301. DOI:https://doi.org/10.1016/j.ins.2019.05.027

[7] Sérgio Moro, Raul M.S. Laureano, and Paulo Cortez. 2011. Using data mining for bank direct marketing: An application of the CRISP-DM methodology. *ESM 2011 - 2011 Eur. Simul. Model. Conf. Model. Simul. 2011* Figure 1 (2011), 117–121.

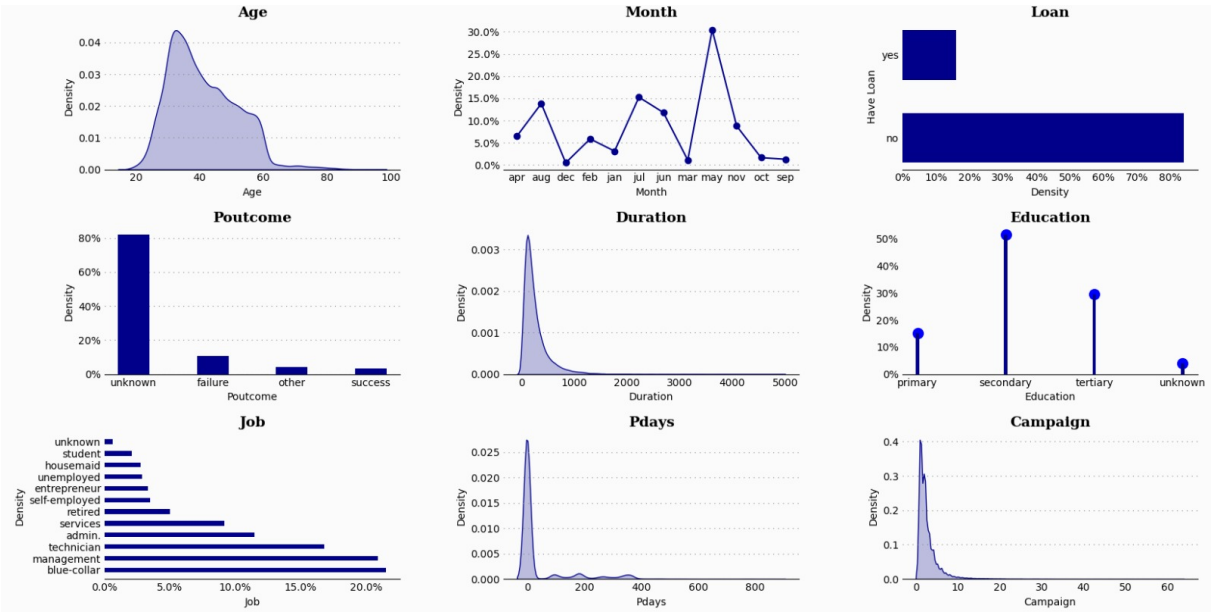
[8] J. Y. Shih, W. H. Chen, and Y. J. Chang. 2014. Developing target marketing models for personal loans. *IEEE Int. Conf. Ind. Eng. Eng. Manag.* 2015-January, (2014), 1347–1351. DOI:https://doi.org/10.1109/IEEM.2014.7058858

[9] Q. R. Zhuang, Y. W. Yao, and O. Liu. 2018. Application of data mining in term deposit marketing. *Lect. Notes Eng. Comput. Sci.* 2, (2018), 14–17.

[10] A Framework for Improving Find Best Marketing Targets Using a Hybrid Genetic Algorithm and Neural Networks.

8. נספחים

נספח 1 : דוגמאות לויזואליזציה של חלק מהנתונים שביצענו

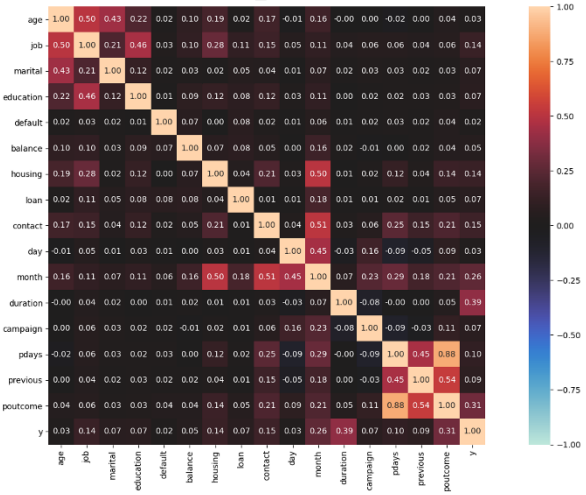


נספח 2 : התפלגויות של הנתונים לאחר קטגוריזציה

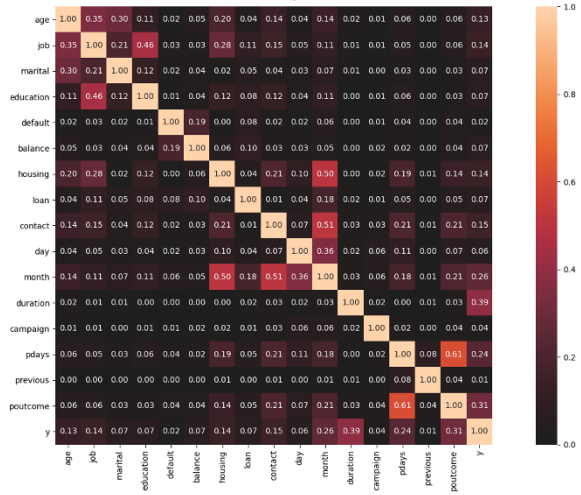
Contact		Day		Month		Duration		Campaign		Pdays		Previous		Poutcome	
Class	%	Range	%	Class	%	Range	%	Range	%	Range	%	Range	%	Class	%
Cellular	65%	[0-5)	13%	Jan	3%	[0-300)	73%	[1-10)	97%	-1	82%	[0-25)	99%	Success	3%
Telephone	6%	[5-10)	17%	Feb	6%	[300-600)	19%	[10-20)	2%	[0-100)	3%	[25-50)	<1%	Failure	11%
Unknown	29%	[10-15)	18%	Mar	1%	[600-900)	5%	[20-30)	<1%	[100-200)	6%	[50-75)	<1%	Other	4%
		[15-20)	22%	Apr	6%	[900-1200)	2%	[30-40)	<1%	[200-300)	3%	[75-100)	<1%	Unknown	82%
		[20-25)	11%	May	30%	[1200-1500)	<1%	[40-50)	<1%	[300-400)	5%	[100-125)	<1%		
		[25-31)	18%	Jun	12%	[1500-1800)	<1%	[50-60)	<1%	[400-500)	<1%	[125-150)	<1%		
				Jul	15%	[1800-2100)	<1%	[60-70)	<1%	[500-600)	<1%	[150-175)	<1%		
				Aug	14%	[2100-2400)	<1%			[600-700)	<1%	[175-200)	<1%		
				Sep	1%	[2400-2700)	<1%			[700-800)	<1%	[200-225)	<1%		
				Oct	2%	[2700-3000)	<1%			[800-900)	<1%	[225-250)	<1%		
				Nov	9%	[3000-3300)	<1%					[250-275)	<1%		
				Dec	<1%	[3300-5000)	<1%								

נספח 3 : מפות חום של התאמות בין משתנים

ב



א



נספח 4 : תוצאות האלגוריתמים לאורך תהליך המידול

ANN

Initial ANN:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.754	0.624	0.674	0.471	0.579	0.406	0.807	0.56	0.935	0.893

		Predicted 0	Predicted 1
Actual	0	7647	338
	1	628	430

Tuned ANN:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.768	0.751	0.517	0.502	0.674	0.641	0.419	0.413	0.853	0.851

		Predicted 0	Predicted 1
Actual	0	7022	963
	1	380	678

Tuned ANN with threshold:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.778	0.77	0.47	0.466	0.747	0.727	0.343	0.343	0.803	0.805

		Predicted 0	Predicted 1
Actual	0	6511	1474
	1	289	769

ANN Test:

Train G_Mean	Test G_Mean	Train F1	Test F1	Train Recall	Test Recall	Train Precision	Test Precision	Train Accuracy	Test Accuracy
0.801	0.794	0.484	0.471	0.802	0.793	0.347	0.335	0.8	0.795

		Predicted 0	Predicted 1
Actual	0	3181	819
	1	108	413

Random Forest

Initial RF:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.926	0.576	0.909	0.419	0.86	0.345	0.964	0.534	0.98	0.888

		Predicted 0	Predicted 1
Actual	0	7666	319
	1	693	365

Tuned RF:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.927	0.625	0.908	0.464	0.864	0.41	0.957	0.535	0.979	0.889

		Predicted 0	Predicted 1
Actual	0	7608	377
	1	624	434

Tuned RF with threshold:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.937	0.785	0.7	0.476	0.987	0.761	0.542	0.347	0.901	0.804

		Predicted 0	Predicted 1
Actual	0	6467	1518
	1	253	805

RF Test:

Train G_Mean	Test G_Mean	Train F1	Test F1	Train Recall	Test Recall	Train Precision	Test Precision	Train Accuracy	Test Accuracy
0.934	0.931	0.697	0.693	0.98	0.973	0.541	0.538	0.901	0.9

		Predicted 0	Predicted 1
Actual	0	3564	436
	1	14	507

Xgboost

Initial Xgboost:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.714	0.6	0.629	0.46	0.519	0.372	0.798	0.602	0.928	0.898

		Predicted 0	Predicted 1
Actual	0	7724	261
	1	664	394

Tuned Xgboost:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.877	0.62	0.85	0.46	0.774	0.404	0.944	0.536	0.968	0.889

		Predicted 0	Predicted 1
Actual	0	7615	370
	1	631	427

Tuned Xgboost with threshold:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.873	0.775	0.53	0.432	0.992	0.806	0.362	0.295	0.794	0.752

		Predicted 0	Predicted 1
Actual	0	5946	2039
	1	205	853

Xgboost Test:

Train G_Mean	Test G_Mean	Train F1	Test F1	Train Recall	Test Recall	Train Precision	Test Precision	Train Accuracy	Test Accuracy
0.861	0.865	0.511	0.512	0.988	0.99	0.344	0.345	0.779	0.782

		Predicted 0	Predicted 1
Actual	0	3021	979
	1	5	516

SVM

Initial SVM:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.558	0.546	0.425	0.407	0.319	0.306	0.636	0.606	0.899	0.895

		Predicted 0	Predicted 1
Actual	0	7774	211
	1	734	324

Tuned SVM:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.716	0.59	0.637	0.448	0.521	0.359	0.818	0.595	0.93	0.896

		Predicted 0	Predicted 1
Actual	0	7726	259
	1	678	380

Tuned SVM with threshold:

Train G_Mean	Validation G_Mean	Train F1	Validation F1	Train Recall	Validation Recall	Train Precision	Validation Precision	Train Accuracy	Validation Accuracy
0.848	0.774	0.555	0.482	0.86	0.72	0.362	0.362	0.839	0.819

		Predicted 0	Predicted 1
Actual	0	6641	1344
	1	296	762

SVM Test:

Train G_Mean	Test G_Mean	Train F1	Test F1	Train Recall	Test Recall	Train Precision	Test Precision	Train Accuracy	Test Accuracy
0.837	0.833	0.531	0.523	0.855	0.848	0.385	0.378	0.823	0.822

		Predicted 0	Predicted 1
Actual	0	3273	727
	1	79	442