**כריית ידע - פרויקט מסכם**

**מרצה**: ד"ר אייל קולמן

**עוזר הוראה**: מר איתי מרגולין

**מגישים**:

|  |  |
| --- | --- |
| **שם** | **ת.ז.** |
| נעם תור | 203302021 |
| תומר ינאי | 305187296 |
| יוגב מטלון | 201390408 |

תוכן עניינים

[תקציר מנהלים 3](#_Toc503963270)

[שלבי העבודה 3](#_Toc503963271)

[שלב מספר 1 – Data Exploration 3](#_Toc503963272)

[שלב מספר 2 – Pre-Processing 4](#_Toc503963273)

[שלב מספר 3 – Feature Selection 5](#_Toc503963274)

[PCA 5](#_Toc503963275)

[LDA 5](#_Toc503963276)

[Decision tree 5](#_Toc503963277)

[שלב מספר 4 – בניית מודלים "פשוטים" 6](#_Toc503963278)

[KNN 6](#_Toc503963279)

[Logistic Regression 6](#_Toc503963280)

[שלב מספר 5 – בניית מודלים מתקדמים 6](#_Toc503963281)

[Random Forest 6](#_Toc503963282)

[Feed Forward Neuron Network 7](#_Toc503963283)

[שלב מספר 6 – השוואה למודל שאיננו מבוסס Machine Learning 7](#_Toc503963284)

[שלב מספר 7 – השוואה בין המודלים ובחירת מודל 8](#_Toc503963285)

[סיכום 9](#_Toc503963286)

[נספחים 10](#_Toc503963287)

# תקציר מנהלים

בפרויט זה יישמנו את החומר הנלמד בקורס "כריית ידע" על סט נתונים אנונימי אשר נלקח מאתר Kaggle. תהליך כריית הידע התבצע במספר שלבים:

1. Data Exploration – תהליך המחקר המקדים אשר כלל תובנות על מבנה ואופי הדאטה.
2. Pre-processing – תהליך העיבוד המקדים לרצת מודלי הלמידה הנבחרים.
3. Feature Selection –הורדות מימד בשיטות PCA, LDA ו-Feature Importance.
4. מודלים "פשוטים" – המודלים הנבחרים הינם KNN, ו-Logistic Regression. התהלך כלל חיפוש היפר-פרמטרים עבור המודלים, והרצת המודלים על הנתונים (במבנה הנבחר)
5. מודלים "מורכבים" – המודלים הנבחרים הינם Random Forest , ו-Feed Forward ANN. גם עבור המודלים הללו בוצע חיפוש אחר היפר פרמטרים והרצה בדומה לשלב 4.

סיכום תוצאות שלבים 4 ו-5:

|  |  |  |
| --- | --- | --- |
| **Model** | **AUC** | **Accuracy** |
| **KNN** | 0.7756 | 0.7405 |
| **Logistic Regression** | 0.7659 | 0.7471 |
| **Random Forest** | 0.7856 | 0.7497 |
| **Feed Forward ANN** | 0.7794 | 0.7438 |

1. השוואה למודל שאינו מבוסס פתרון Machine Learning – יצירת אלגוריתם Rule-Based, המבצע סיווג לתוצאות (ללא תהליך "למידה"). האלגוריתם החזיר תוצאות קרובות למודל סיווג רנדומאלי.
2. ביצוע השוואה בין המודלים ובחירת מודל להרצה על סט נתוני ה-Test (הלא מתויגים). המודל הנבחר שהורץ – **Random Forest**.

# שלבי העבודה

## שלב מספר 1 – Data Exploration

1. **בחירת נפוצים**– מבין תשעת הקלאסים הקיימים בדאטה בחרנו לסווג את המוצרים לשתי הקטגוריות הנפוצות ביותר: קטגוריה 2 ו-6. בעקבות בחירת קטגוריות אלו נותרנו עם 22,719 רשומות (מתוך 46,408).
2. **בדיקה האם התגיות מאוזנות** – האיזון חשוב מאוד כדי לוודא שהמודלים לא מסיקים מסקנות שגויות על סמך תגיות שהם לא מאוזנות (Data Label Distribution במחברת). ניתן לראות כי הלייבלים אכן מאוזנים (53% לקטגוריה 2 והשאר של 6).

**"מבט על" עבור הפיצ'רים** – ביצענו בדיקה עבור סוג (Type) כל פי'צר ובדיקה האם ישנם ערכים שאינם מספריים (או ערכי Null כמובן), תיאור סטטיסטי של כל אחד מהפיצ'רים (ממוצע, סטיית תקן, מינימום, מקסימום ורבעונים), ותרשים Box Plot לכל פי'צר. מניתוח ה-Box Plot ניתן לראות מעיכה דרמטית של הנתונים סביב ה-0 עבור כל הפיצ'רים.

1. **בדיקת התנהגות (התפלגות) הנתונים** – ביצענו בדיקות של התפלגויות הנתונים השונים (עבור כל feature) באמצעות גרפים העוזרים להבין את התנהגות הנתון בהשוואה לכל קטגוריה. בדיקות אלו מראות כי בכל אחד מהפיצ'רים הערך 0 הוא הנפוץ ביותר ובנוסף יכולות להעיד על קורלציה כפי שנפרט בהמשך. בנוסף ההתפלגויות יכולות לעזור לנו להבין אילו פיצ'רים יהיו הכי יעילים באבחנה בין שני הקלאסים- הפיצ'רים בהם השוני הגדול ביותר בין ההתפלגויות. במקרה זה ההתפלגויות דומות יחסית.

בדקנו עבור כל פי'צר את מידת ה"ספארסיות" שלו – מה אחוז המדויק של הערכים שאינם אפס בכל פיצ'ר, ניתן לראות כי בפיצ'ר 70 כמות המידע הינה הגדולה ביותר, ואילו פיצ'ר 51 כמעט לא כולל רשומות שערכם אינו 0.

בשלב זה זיהינו כי עלולה להיות "חפיפה" בין המחלקות בנתונים, כלומר רשומות אשר זהות לחלוטין אחת לשניה אך מסווגות למחלקות שונות. החלטנו לבדוק חפיפה זאת בכך שנבדוק את הרשומות בהן כל הפיצ'רים מאופסים, זיהינו כי קיימות **8581** רשומות כאלו, כאשר כמות הרשומות הכוללת היא 27,700. מתוך הרשומות המאופסות 6196 שייכות למחלקה 2 ו-2385 שייכות למחלקה 6.

מובן מאליו שההחלטה הטובה ביותר שמודל יכול לקבל בהינתן הנתונים ורשומה מאופסת היא לסווג את הרשומה למחלקה 2. אך נשים לב שבמצב זה המודל יטעה בסווג של לפחות 2385 רשומות - כלומר אנו מתחילים מחסם עליון של 89.5% דיוק. וזאת מבלי לבדוק חפיפות נוספות בנתונים, אנו מניחים שחסם זה הוא אף נמוך יותר.

1. **קורלציה** – בחלק זה בדקנו האם ישנה קורלציה בין הפיצ'רים השונים. קורלציה גבוהה תעיד כי עלינו לשקול הוצאת פיצ'רים, שכן אינן מסבירות מידע חדש. לאחר הבדיקה ראינו כי הפיצ'רים אינם קורלטיביים זה לזה, הקורלציות שנמצאו נמוכות מ0.4

## 

## שלב מספר 2 – Pre-Processing

1. **הסרת חריגים** – לא ביצענו הסרת חריגים מהנתונים. הסיבה לכך היא שמרבית הנתונים מאופסים כך שכפי שראינו בשלב הData exploration אפילו האחוזון ה75% בכלל הפיצ'רים הינו 0. כלומר הסרת חריגים תוריד רשומות חשובות בהם קיים מידע שאיננו 0, אך בנתונים אלו כל מידע שאיננו מאופס יקר ולכן חשוב לשמור רשומות אלו.
2. **Feature Extraction**- בחרנו להוסיף 2 עמודות לנתונים. עמודה “zeros\_number” – מכילה את כמות ה0 הקיימת ברשומה, לדוגמא רשומה בה כל הפיצ'רים הינם 0 תכיל את המספר 10 בעמודת ה’zeros\_number”. הרעיון מאחורי עמודה זאת הוא לסכם בפיצ'ר אחד את כמות הפיצ'רים שאינם מאופסים ברשומה, זאת משום שלהערכתנו זהו מידע חשוב בסיווג הקלאסים.

עמודה “smart\_multy”- מכילה הכפלה של כלל המספרים ברשומה אחד בשני, אך משום שבכל רשומה קיים 0 אחד לפחות, הוספנו החרגה שהכפלה ב0 למעשה מהווה הכפלה ב1.

לדוגמא : רשומה המכילה בפיצ'ר מסויים 5 בפיצ'ר אחר 6 והשאר אפסים תקבל את הערך 30.

רשומה המכילה 10 אפסים (כלל הפיצ'רים) תקבל את הערך 1.

הרעיון מאחורי עמודה זאת הוא לתת משמעות חזקה ולא לינארית לכמות ולערך המספרים שאינם 0 ברשומה.

הערה: בחרנו להוסיף שתי עמודות בלבד משום שלדעתנו עצם הספארסיות של הנתונים גורמת לכך שמניפולציות מתמטיות על הנתונים הקיימים לא יביאו הרבה תועלת- זאת משום שעיקר המידע כבר קיים בהאם נתון מסויים הוא 0 או לא, לכן לוגריתמים ופולינומים למיניהם יביאו תועלת מועטה.

1. **נרמול הנתונים-** השתמשנו בשני סוגי נרמול ושמרנו את גם את הפיצ'רים המקוריים וגם את הפיצ'רים המנורמלים על מנת להחליט בהמשך באיזה סוג נרמול, אם בכלל, ברצוננו להשתמש.

נרמול MinMax. הנרמול ממיר את הנתונים לאותה הסקלה – [0,1]. הנרמול מזיז את תוחלת ההתפלגות המקורית אך שומר על צורתה.

נרמול Z – הנרמול ממיר את הנתונים לסקאלה של "סטיות תקן", סקאלה שאיננה חסומה. כלל הנתונים מנורמלים לתוחלת 0 וסטיית תקן 1. היתרון של נרמול זה הוא שהוא מתאים יותר לשימוש ב-PCA (משום שסטיות התקן והתוחלות שוות).

1. **השלמת מידע חסר-** כפי שניתן לראות בסיכום הנתונים, לא קיימים נתונים חסרים בדאטה. וכפי שציינו נתייחס ל-0 כנתון בעל משמעות ולא כמידע חסר.
2. **הסרת עמודות-** לא הסרנו עמודות משום שאף על פי שיש עמודות בהם קיימים מעט מאוד ערכים, כל מידע חשוב בשל הספארסיות של הנתונים. וגם המעט מידע הקיים בעמודות אלו עלול להיות חשוב.
3. **שינוי Type** – לצרכי נוחות החלפנו את עמודת הtarget לעמודה בינארית כאשר 1 מייצג שהרשומה שייכת לclass\_2 ו0 מייצג שהרשומה איננה שייכת לclass\_2, כלומר שייכת לclass\_6

בנוסף הפרדנו עמודה זאת לווקטור נפרד בשם Y, ולעמודות הפיצ'רים נתנו את השם X.

## שלב מספר 3 – Feature Selection

ביצענו 3 בדיקות על מנת להבין האם ביכולתנו להוריד את מימד הנתונים וכיצד לעשות זאת:

### PCA

את ה-PCA ביצענו על העמודות המנורמלות בשיטת Z, זאת על מנת ששוני אפשרי בסקאלות של הפיצ'רים לא ישפיע על הקומפוננטות. בשרטוט של שתי הקומפוננטות הראשונות ניתן לראות שלא נוצרת הפרדה טובה בין המחלקות אותם ברצוננו לסווג - נקודות שתי המחלקות חופפות אחת לשניה.

בבחינת הקומפוננטות שנוצרו אנו רואים שגרף מצטבר של אחוז השונות המוסברת עולה בצורה איטית ונזדקק לכמעט 10 קומפוננטות על מנת להסביר 95% מהשונות בנתונים. זאת כאשר המימד המקורי הינו 12. זוהי איננה הורדת מימד משמעותית ולכן לא נשתמש ב-PCA להוריד מימד בבעיה.

### LDA

את ההתאמה של קומפוננטות ה-LDA ביצענו על העמודות המנורמלות בשיטת MinMax, אך בגרף שנוצר ניתן לראות שהקומפוננטה שנוצרה לא יוצרת הפרדה טובה בין המחלקות- יש חפיפה משמעותית בין הנקודות. לכן הורדת מימד בשיטת LDA איננה מתאימה במקרה זה.

### Decision tree

ביצענו אימון של עץ החלטה על הנתונים המנורמלים בשיטת MinMax, על מנת לראות מה הפיצ'רים שהעץ יבחר כמשמעותיים להפרדה בין הקלאסים. אימנו את העץ כאשר העומק המקסימלי הוא 24 (פעמיים מספר הפיצ'רים) והמדד הנבדק הוא Gini. גם בשיטה זו ניתן לראות שאם ברצוננו להגיע למעל 95% במדד הfeature importance של העץ עלינו לבחור עשרה פיצ'רים. הורדת מימד לא משמעותית.

בסיכום שיטות Feature selection בהם השתמשנו ניתן להגיד כי לא נמצאה שיטה שהראתה אפשרות מובהקת להורדת מימד, תוך כדי שמירת המידע הכלול בהם. לכן הגדרנו את סט ה-train default להיות סט הנתונים הכולל את כל 12 הפיצ'רים. למרות זאת אנו רוצים להשאיר לעצמנו בהמשך אפשרות לבדוק ביצועים של מודלים על הנתונים במימד נמוך יותר (ראה דוגמה לכך במודל KNN).

נבחר את הפיצ'רים שהיו משמעותיים בשיטת "עץ ההחלטה" ונגדיר בעזרתם דאטה-פריים חדש בו יכללו רק 5 מימדים- “feat\_61”, “feat\_20”, “feat\_70”, “feat\_51”, “smart\_multi”

כלל המימדים מנורמלים בשיטת MinMax, סכום הפיצ'רים במדד הfeature importance של העץ המאומן הוא 82.55%

## שלב מספר 4 – בניית מודלים "פשוטים"

כשלב מקדים לשלב אימון המודלים ביצענו שתי פעולות:

1. חלוקת הנתונים בשיטת K-fold כאשר k=5, ושמירת הסטים המחולקים במערך לשימוש עתידי בשיטת Cross Validation (CV) בזמן אימון המודלים.
2. כתיבת פונקציה להצגת גרף ROC עבור תחזיות של מודל מסויים. חשוב לציין שהפונקציה נלקחה במלואה מתוך הדוקומנטציה של sklearn, רפרנס קיים בתוך מחברת הפרוייקט.

### KNN

אימנו מודל KNN על סט הנתונים train\_defualt, ביצענו חיפוש בעזרת K-fold cross validation אחר ה-k המתאים ביותר לסט הנתונים שלנו. המדדים שנבדקו על מנת להשוות בין הערכי ה-k השונים הינם “Accuracy” ו"AUC" והחיפוש בוצע על ערכי K שגדלים בצורה אקפוננציאלית בחזקות של 2- כלומר K = 2,4,16….

ניתן לראות שכבר בk=8 המדדים עולים בצורה חדה ולאחר מכן מתקיימות עליות מינוריות יותר עד לk=64. בבחירת הk האופטימלי התלבטנו בין 64 ל32 משום שבk=64 מדד ה”AUC” גבוה יותר אך מדד ה"accuracy" נמוך יותר.

לבסוף החלטנו לבחור ב-k=32 משום שלמרות שההבדלים מאוד מינוריים הם טיפה משמעותיים יותר במדד ה”accuracy"

בk- שנבחר המדדים שחושבו בשיטת הk-fold הם:

יש לציין שהמדדים חושבו בשיטת K-fold וכאשר נאמן את המודל על 100% מהנתונים הם עשויים להשתפר.

בנוסף לאימון מודל זה בדקנו גם בצורה דומה מודל שאומן על train\_reduce , דאטה במימד נמוך יותר שכלל רק את חמשת הפיצ'רים הטובים ביותר על פי הfeature importance, מודל זה הגיע לתוצאות פחות טובות מהמודל שאומן על כלל הפיצ'רים (ערכים מופיעים במחברת הקוד).

### Logistic Regression

אימנו מודל רגרסיה לוגיסטית על סט הנתונים train\_defualt, וגם פה השתמשנו ב- CV, הפעם על מנת למצוא את ה-c המתאים ביותר למודל שלנו, השתמשנו ב-c מסדרי גודל שונים בערכים:

C = 0.01, 0.1, 0.5, 0.75, 1. הערך C=1 השיג את הביצועים הטובים ביותר, כאשר המדדים שחושבו הם:

מודל זה לא השיג ביצועים טובים יותר ממודל ה KNN והחלטנו לעבור לבדוק מודלים מתקדמים יותר.

## שלב מספר 5 – בניית מודלים מתקדמים

בשלב זה אימנו שני סוגי מודלים – Random Forest ורשת נוירונים.

משום שבשני המודלים יש מספר היפר-פרמטרים רב יחסית אשר יש להתאים לסוג הנתונים נעזרנו במחלקה מתוך sklearn בשם GridSearchCV אשר מריצה מספר רב של פעמים את המודל הנבחר, בכל פעם בשילוב אחר של פרמטרים אפשריים שהוגדרו. כל הרצה עובדת בשיטת k-fold בערך דיפולטיבי של k=3. מתודות המחלקה מאפשרות למצוא את הפרמטרים המביאים לתוצאה הטובה ביותר של המדד שהוגדר, ושומרות מודל מאומן בפרמטרים אלו בו ניתן להשתמש.

### Random Forest

את המודל אימנו על סט הנתונים train default, כאשר החיפוש במרחב ההיפר-פרמטרים התבצע על האפשרויות הבאות:

min\_samples\_split = 4, 8, 16, 20

max\_depth = 5,8,10,20,30

criterion = 'gini’ or 'entropy’

'n\_estimators = 50

random\_state = 1

המדד שנבדק הינו AUC

הפרמטרים שהובילו לתוצאות הטובות ביותר הינם:

המדדים שהתקבלו מהמודל:

כלומר, זהו המודל בעל התוצאות הטובות ביותר עד לנקודה זאת.

### Feed Forward Neuron Network

גם במודל זה השתמשנו במחלקה GridSearchCV . חשוב לציין שזמן הריצה של הפונקציה היה ארוך משמעותית ולכן צמצמנו את אפשרויות החיפוש בצורה ניכרת.

כל השכבות הינן מסוג fully Contact וכוללות פונקציית אקטיבציה זהה שתיבחר בחיפוש במרחב הפרמטרים של המודל. באפשרויות לשכבות השונות השווינו בין 3 אפשרויות שונות בארכיטקטורה אך זהות במספר הפרמטרים (100). בנוסף הגדלנו בכל הריצות את כמות האיטרציות המקסימלית ל-1500 על מנת להגדיל את הסיכוי להתכנסות.

החיפוש במרחב ההיפר-פרמטר כלל האת האפשרויות הבאות:

המדד שנבדק הינו AUC.

הפרמטרים שהובילו לתוצאות הטובות ביותר הינם:

כאשר פרמטרים אלו הגיעו ל-AUC של 0.784 בעת החיפוש.

לאחר מכן לא הצלחנו לשחזר תוצאה זאת (כנראה בשל הסטוכסטיות המעורבת באימון הרשתות) ובאימון בשיטת CV עם הפרמטרים שנבחרו הושגו התוצאות:

ואלו ערכים טובים פחות מאשר של מודל ה-Random Forest.

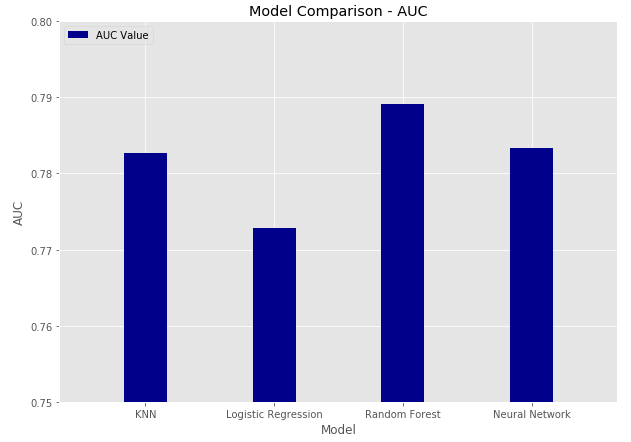
## שלב מספר 6 – השוואה למודל שאיננו מבוסס Machine Learning

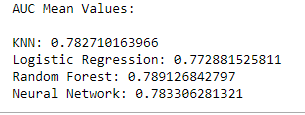
על מנת להשוות את התוצאות לשיטות מסורתיות יותר, בנינו אלגוריתם שמבוסס על חוקים סטטיסטיים על פי התפלגות כל פיצ'ר במחלקות. עבור כל פיצ'ר חושב ממוצע לכל מחלקה. כאשר האלגוריתם בודק לאיזו מחלקה לשייך את הרשומה הוא משווה בין ערך הפיצ'ר ברשומה לערכו הממוצע של הפיצ'ר בכל מחלקה. הפיצ'ר מקבל את ערך המחלקה אליה הוא יותר קרוב, ולבסוף התיוג הסופי מתבצע על פי שיטת הרוב קובע (עבור כלל הפיצ'רים)

אלגוריתם זה לא הצליח להגיע לתוצאות טובות, ואף לא עבר בהרבה את התוצאות שאלגוריתם רנדומלי היה משיג. ערך ה-Accuracy של המודל:

## שלב מספר 7 – השוואה בין המודלים ובחירת מודל

ביצענו השוואה בין ה AUC של המודלים השונים על מנת לבחור את המודל עם הביצועים הטובים ביותר- ה AUC שהשווה הוא חישוב ממוצע חמשת ההרצות של ה-CV:





כמו כן, במחברת הקוד, מופיעות השוואות והצגות בין גרפי ה-ROC הממוצעים של כל המודלים.

ניתן לראות שהמודל שהשיג את הביצועים הטובים ביותר הוא ה Random Forest לכן השתמשנו במודל זה על מנת לבצע פרדיקציה על הטסט.

הפרדיקציות ייוצאו לקובץ CSV בשם " predictions\_on\_test.csv".

# סיכום

בפרוייקט זה יישמנו את הידע שלמדנו בקורס על סט נתונים אשר נלקח מאתר Kaggle, ביצענו את התהליך במספר שלבים-

1. Data Exploration - בשלב זה ביצענו "היכרות" עם סט הנתונים, הפיצ'רים השונים והמחלקות אליהן יש צורך לסווג. בדקנו קורלציות, וזיהינו שהנתונים מכילים כמות גדולה של אפסים בכל העמודות, ואף כמות בה של רשומות המכילות כולן אפסים, ומתויגות בקלאסים שונים. ראינו כי לאור עובדה זו אנו מתחילים מחסם עליון של 89.5% דיוק, וזאת מבלי לבדוק חפיפות נוספות בנתונים.
2. Pre-processing - בשלב העיבוד המקדים ביצענו את כלל המניפולציות על הנתונים עבור הרצתם במודלים. ביניהם Feature Extraction, נרמול הנתונים, שינוי type, השלמת מידע חסר.
3. Feature Selection – בוצעו הורדות מימד בשיטות PCA, LDA ו-Feature Importance באמצעות Decision Tree.
4. בוצעו הרצה של מודלים KNN, ו-Logistic Regression. עבור כל אחד מהם בוצע חיפוש אחר ההיפר-פרמטרים (והשילוב עם הנתונים במימדים שונים משלב 3), המספקים את המודל בעל התוצאות הטובות ביותר (בשיטת CV).
5. בוצעו הרצה של מודלים Random Forest , ו-Feed Forward ANN. גם עבור המודלים הללו בוצע חיפוש אחר היפר פרמטרים בדומה לשלב 4.

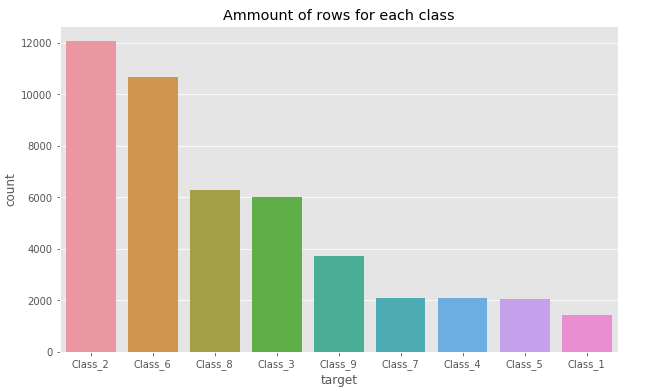
סיכום תוצאות שלבים 4 ו-5:

|  |  |  |
| --- | --- | --- |
| **Model** | **AUC** | **Accuracy** |
| **KNN** | 0.7756 | 0.7405 |
| **Logistic Regression** | 0.7659 | 0.7471 |
| **Random Forest** | 0.7856 | 0.7497 |
| **Feed Forward ANN** | 0.7794 | 0.7438 |

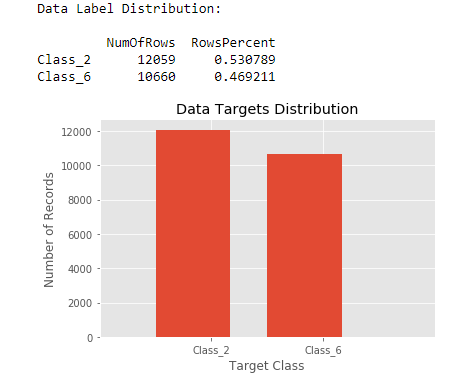
1. השוואה למודל שאינו מבוסס פתרון Machine Learning – יצרנו אלגוריתם Rule-Based, המבצע סיווג לתוצאות (ללא תהליך "למידה"). האלגוריתם החזיר תוצאות קרובות למודל סיווג רנדומאלי.
2. ביצוע השוואה בין המודלים ובחירת מודל להרצה על סט נתוני ה-Test (הלא מתויגים). המודל הנבחר שהורץ – Random Forest.

ברמה האישית, ניתן בהחלט לומר כי יישום תהליך כריית הידע אפשר לנו להתנסות, לחוות ולהפנים את חומר הקורס (ומעבר לו), ללמוד באופן אקטיבי ואפקטיבי, ולזכות בטעימה מעולמו של Data Scientist.

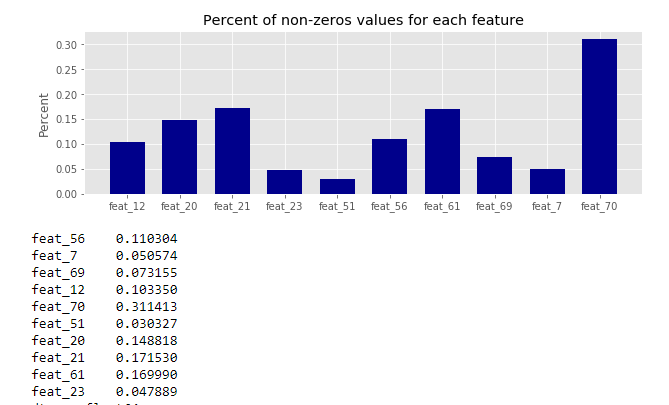
# נספחים

כמות הרשומות מכל מחלקה בדאטה הראשוני:

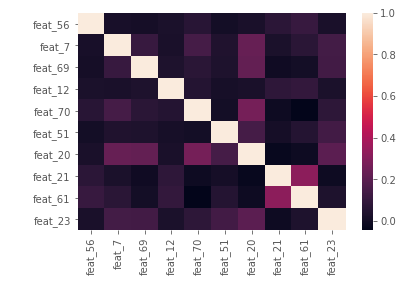
התפלגות הנתונים לאחר השארת שתי המחלקות העיקריות, מחלקה 2 ומחלקה 6, בלבד:



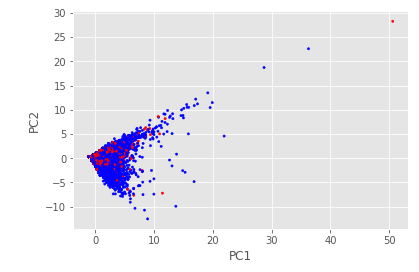
אחוז הערכים שאינם "אפס" עבור כל פיצ'ר:



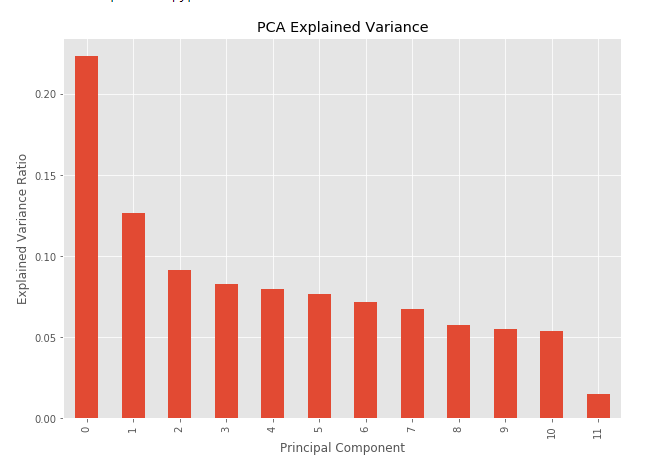
מטריצת הקורלציות בין הפיצ'רים:



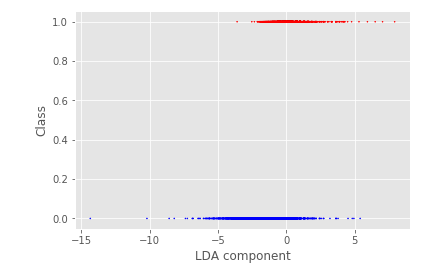
הטלת הרשומות לשתי הקומפוננטות הראשונות על מנת לזהות את התפלגות המחלקות במרחב לאחר PCA:



אחוז השונות המוסברת שמספקת כל קומפוננטה



ויזואליזציה של שתי המחלקות לאחר חישוב קומפוננטת LDA:



חשיבות הפיצ'רים לפי דירוג בשיטת עץ החלטה:

