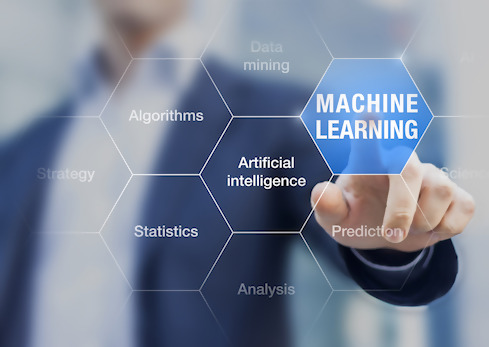
דו"ח פרויקט

Machine Learning Project



מנחה הקורס: ד"ר רוני הורביץ

שמות חברי הצוות:

מתן חובר: 203314752

אלירן סרור: 305404337

אמיר מסללאתי:203875166

חן שליו: 203205984

קישור ל- bitbucket: <https://bitbucket.org/Matanch/r-studio-project/src/master/>

**מבוא**

בפרויקט זה, במסגרת הקורס "כריית ידע ולמידת מכונה", נדרשנו לחקור DataSet המכיל נתונים גולמיים, המציגים שלל מידע פיננסי של 5910 חברות פולניות. המידע שלנו מכיל ציונים פיננסיים מהשנה הראשונה של החיזוי ולאחר 5 שנים.

עמודת המטרה- מה קרה בפועל לחברה, התבקשנו לבצע ניתוח לנתונים וכריית מידע על ה-DATA בכדי לחזות מי מהחברות עלולה לפשוט רגל על סמך הנתונים שבידינו.

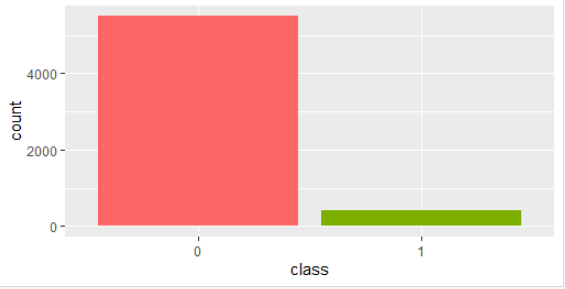
**מטרות הפרויקט**

* הרחבת הידע בחומר הנלמד, תוך שימוש בלמידת מודלים ונושאים חדשים.
* הפקת מודלים אשר באמצעותם יתקבלו תוצאות חיזוי אופטימאליות שבאמצעותן נצליח לחזות האם חברה פולנית תפשוט רגל או להיפך.
* זיהוי עמודות (תכונות ) אשר משפיעות בצורה מובהקת על עמודת המטרה.
* מתן סיכום אשר אמור לייצר לבעלי העניין יכולת השפעה גבוהה בקבלת החלטות.

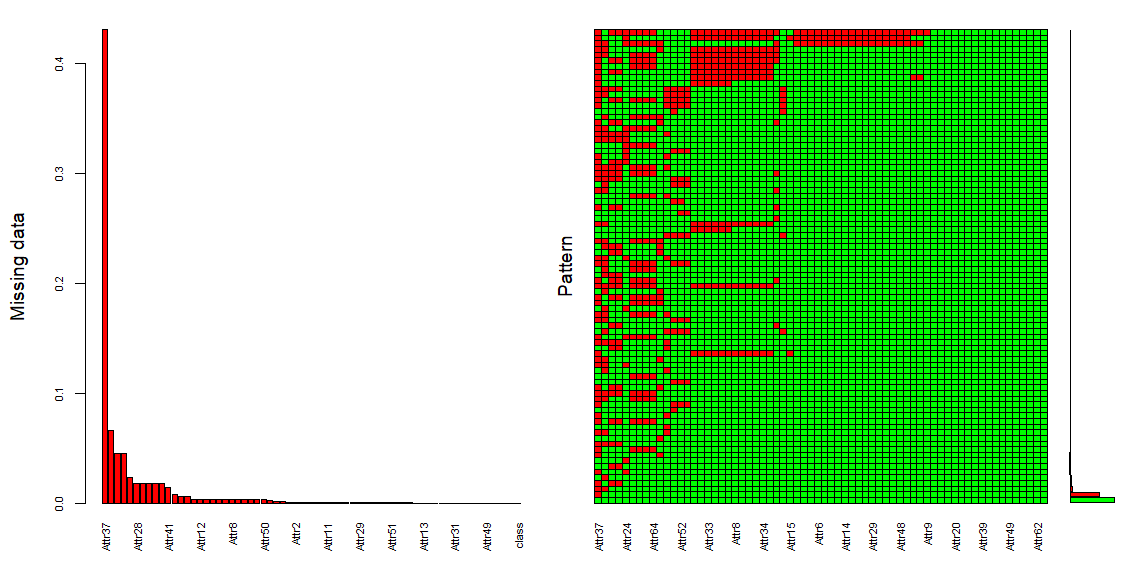
**תהליך העבודה**

תחילה בחנו את הנתונים שקיבלנו בקובץ והבנו שמדובר במאגר מידע גדול הכולל בתוכו המון מספרים הקשורים לניתוחים פיננסים, על מנת לנתח אותם נצטרך לבצע מספר פעולות הכוללות, ניקיון וסידור של הנתונים במטרה להגיע למידע הכי נכון ומדויק.

ביצענו בדיקת ראשונית לראות את היחס בעמודת המטרה – כבר בשלב זה זיהינו כי ישנה בעיה

 משמעותית של נתונים לא מאוזנים , אשר עלולים לפגוע באמינות המודלים.

בכדי לסדר את הנתונים, השתמשנו בכלים שניתנו לנו במהלך הקורס ואף ביצענו מחקר מעמיק באתרי אינטרנט אשר הסבירו לנו כיצד עלינו להתמודד בכל שלב.

**ניקיון הנתונים בוצע בצורה הבאה:**  
תחילה, טענו את כל הספריות שבהם נעזרנו לבנות את הגרפים והמודלים בהם בחרנו להשתמש.  
שנית , בדקנו את כמות ה-NA בנתונים שלנו, וגילנו כי ישנו מספר רב של נתונים חסרים.   
יצרנו ווקטור אשר נותן לנו אינדיקציה באילו עמודות קיימים ערכי NA. בנוסף, בחרנו לבצע פונקציה בשם aggr(df) על מנת לראות באופן גרפי את אחוז הנתונים החסרים בכל עמודה.

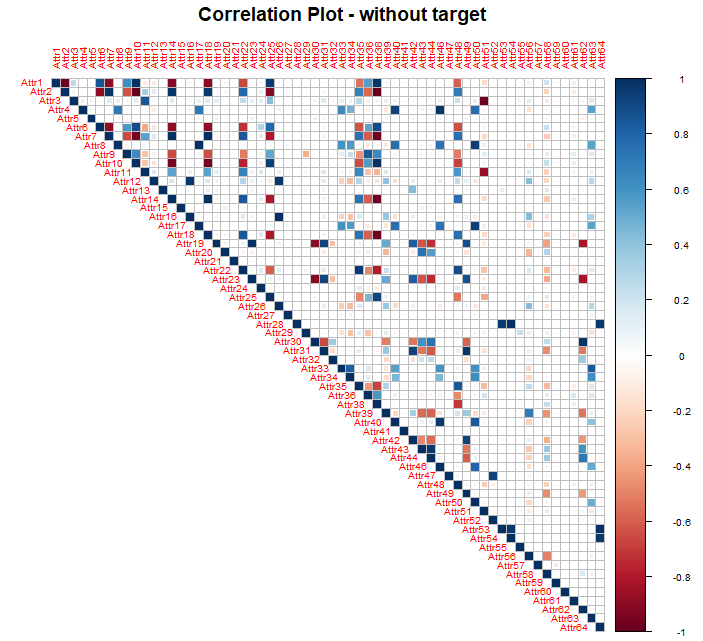
בעקבות בדיקה זו, החלטנו למחוק את תכונה-37Attr מכיוון שקיימים המון ערכים חסרים (כמעט 45%) ועל כן, ניתן להסיר את העמודה מכיוון שלא נוכל להסיק באמצעותה דבר בהקשר לתכונת המטרה בנוסף, החלטנו לטפל גם בתוכנה Attr21 אשר היא מתארת את היחס בין המכירות לשנה הנוכחית לעומת השנה הקודמת. ישנם מקרים בהם חברה נפתחה במהלך השנה ולכן אין נתונים משנים קודמות, דבר זה מוביל לערכים חסרים בעמודה זו ולכן, את כל הערכים החסרים בעמודה זו נשנה ל-0.

בנוסף, החלטנו לטפל גם בתוכנה Attr27 אשר היא המתארת את היחס בין הרווחים השוטפים לבין ההוצאות הפיננסיות של החברה (קרי הלוואות), לכן במצב בו קיים ערך חסר סביר להניח שזה משום שלחברה אין הלוואות או שזו טעות בהזנה של הנתון. לכן נשנה את הערכים החסרים ל-0.

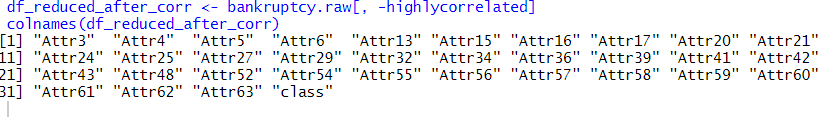
את העמודות "id" , "Attr45" נסיר.

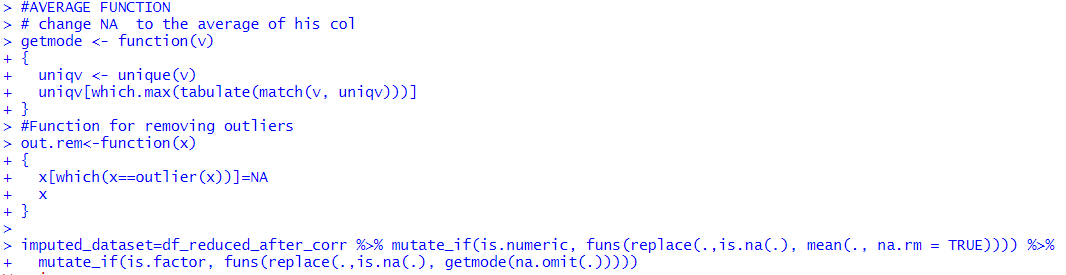
עמודת "id" הוסרה וזאת מכיוון שלא תורמת לחיזוי משתנה המטרה ואף עלולה לייצר מצב של .verfitting בנוסף, החלטנו להסיר גם את Attr45 מכיוון שהיא דומה מאוד ל-Attr60 כך שנוצרת כאן כפילות מיותרת של נתון.

לאחר מכן ביצענו בדיקת קורלציה(התאמה של מעל 80%), בדיקה שמראה את הקשר בין העמודות השונות במספר צורות, כלומר אם השתנות של תכונה מסוימת גוררת השתנות של תכונה אחרת. על מנת שנוכל לחזות נכון את התנהגות משתנה המטרה על סמך התכונות השונות עליהן להיות בלתי תלויות זו בזו. לשם בדיקה זו השתמשנו בגרף ה corrplot,אשר בודק התאמה בין עמודות, לצורך בדיקה זו הורדנו את עמודת ה-Target אשר לא תורמת לבדיקה זו.

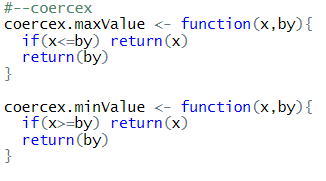
להלן תרשים ה- corrplot:

לאחר מכן, יצרנו וקטור אשר מכיל את המספרים של העמודות אשר נמצאה בהם התאמה גבוהה ובסוף הורדנו אותם מהמודל שלנו.  
Due to high correlation, these <<columns>> are recommended to be **removed:before** Treatment of NA values:

# 1 2 7 8 9 10 11 12 14 18 19 22 23 26 28 30 31 33 35 37 39 43 44 45 47 48 49 51 62  
 **טיפול בעמודות הנוספות עם ערכי NA:**לאחר מחקר שביצענו באינטרנט בנושא טיפול בערכי NA- נמצאו מספר שיטות אשר נותנות פתרון לבעיית הערכים החסרים(שיטת אמצע הקטע, מספר ייחודי או הזנת נתונים במודל MICE.   
אנו הגענו להחלטה שהדבר הטוב ביותר לעשות (וזה גם מה שנלמד בכיתה),להכניס אליהם את הממוצע של אותה עמודה .לבסוף זה המודל שגם יישמנו מכיוון, שהוא שיקף את המידע בצורה הכי מדויקת.  
 **בדיקות EDA**

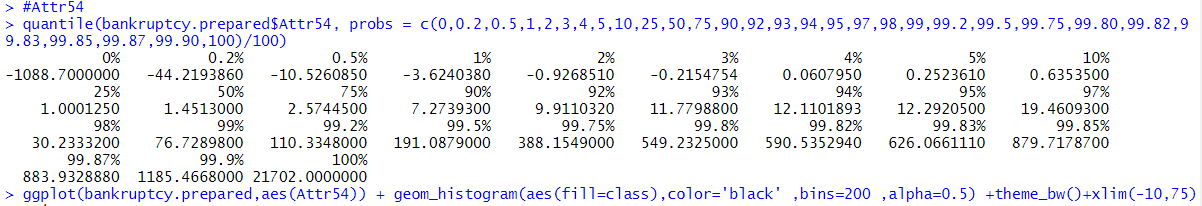
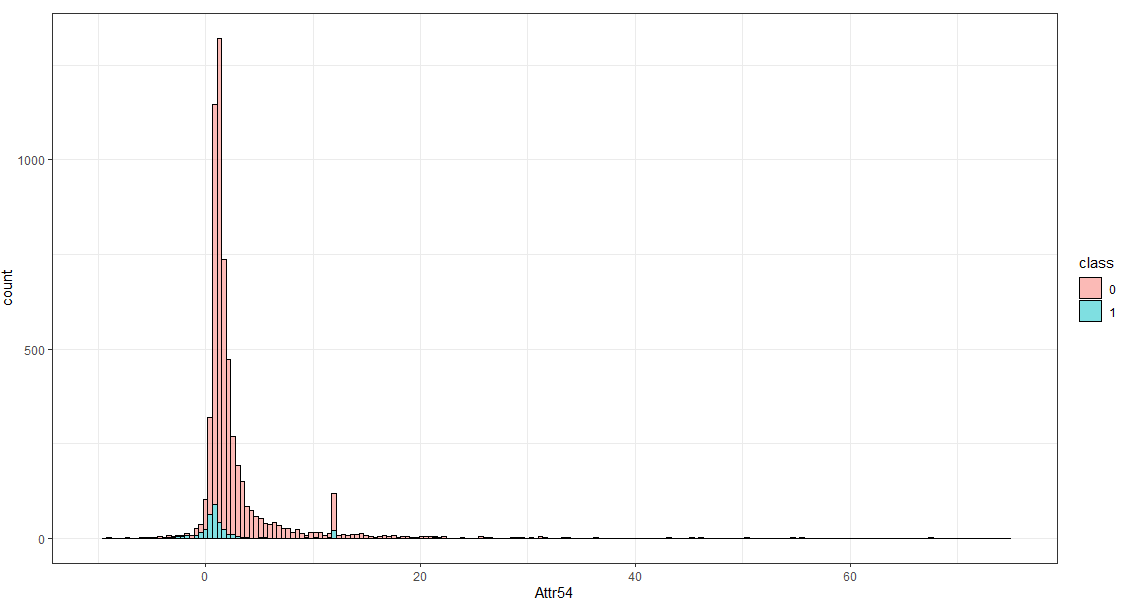


בשלב הראשון בדקנו בעזרת פונקציית היסטוגרם אם יש לשדות חריגים (זנבות). במצבים רבים המידע התקבץ באזור מסוים והיו מדגמים עם ערכים גבוהים או נמוכים בפער משמעותי מאוד משאר המידע, ולכן החלטנו להוריד את החריגים באמצעות הפונקציות אשר בנינו במהלך ההרצאה.

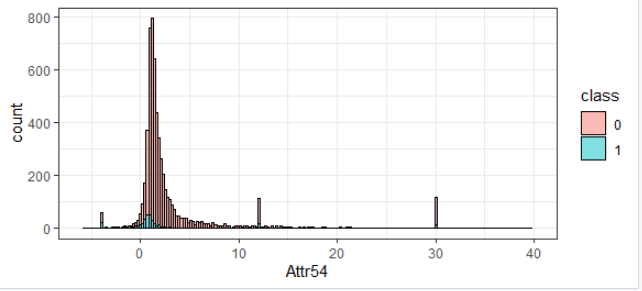


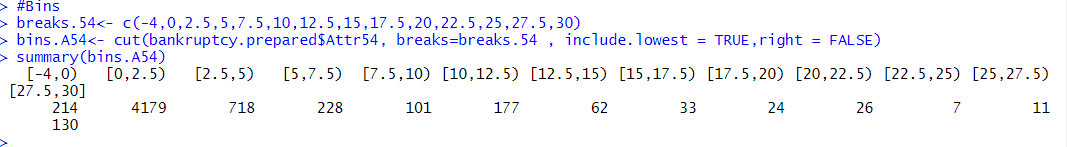
כל תכונה אשר החלטנו להשאיר במודל (לאחר תרשים הקורלציה) ביצענו קיבוץ ערכים חריגים(במידה וקיימים) ,לבסוף בדיקה באמצעות **Bining** בכדי לבדוק האם עמודה זו רלוונטית למודל.

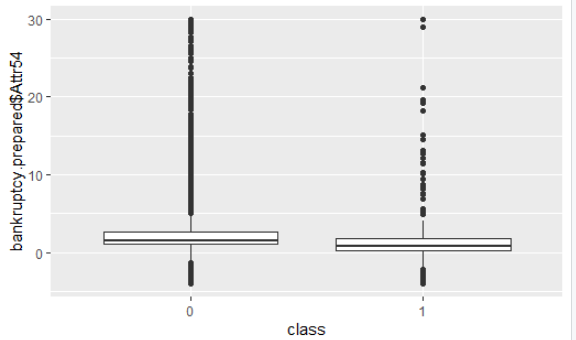
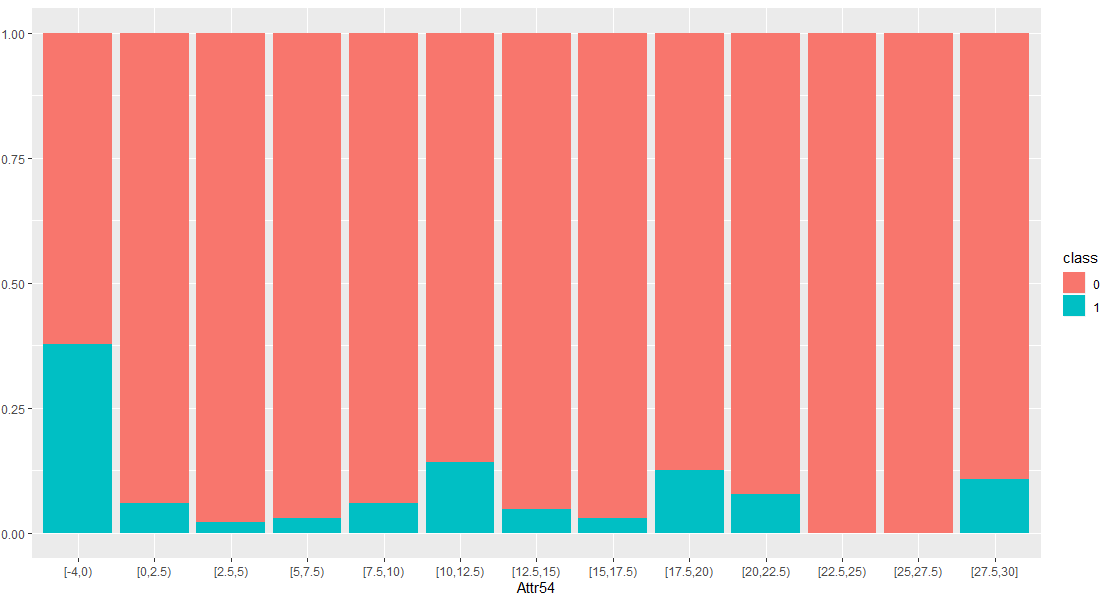
**להלן דוגמא לשימוש בתהליך ה-EDA**

בחרנו להתבונן בתכונה Attr54-

הוצאת חריגים  
הוצאנו חריגים באמצעות הפונקציות שיצרנו (מופיעות בתרשים מלעיל).גרף לאחר ניקוי החריגים



ביצוע Bins

ניתוח העמודה

זיהוי

זיהינו חוסר מגמה בעמודה זו ובנוסף מספר גבוהה של חריגים , לכן מחקנו אותה מהמודל.

**מודלי חיזוי**

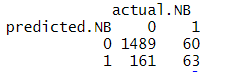
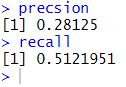
בשלב האחרון הרצנו שבעה מודלים ולצורך כך חילקנו את מסד הנתונים Train וTest-, ביחס של 70:30. להלן המודלים:

* Naive Bayes

בבדיקות הנ"ל החלטנו שההסתברות שחברה תפשוט רגל היא כאשר תוצאות ה-prediction הן מעל 0.5.

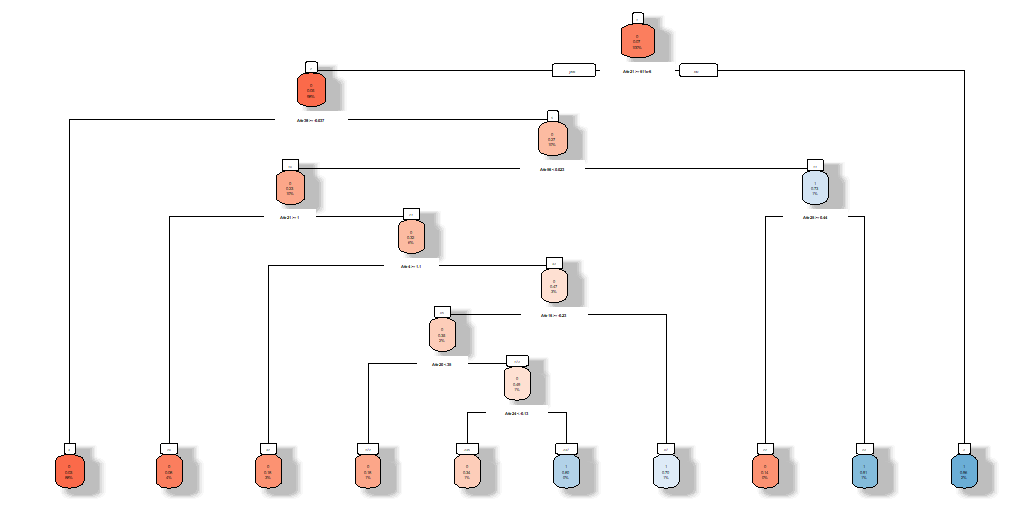
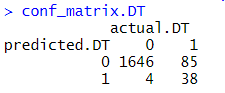
בנוסף החלטנו לייחס חשיבות גדולה יותר לrecall כיוון שמדובר בנתוני פשיטות רגל.

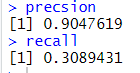
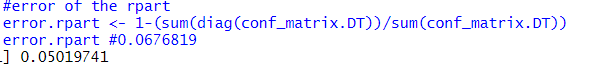
* Decision Tree
* Random Forest
* KNN
* Logistic Regression
* AdaBoost
* Tree Bag
* Naive Bayes after balance
* Neural nets

**Naive Bayes model**  
 confusion matrix: precsion and recall:

accuracy

מסקנות המודל: ב28.1% מהמקרים שהמודל חוזה שאכן העסק יפשוט רגל הוא אכן צודק.  
בנוסף, המודל מצליח לחזות רק ב-51.2% הצלחה עסקים שאכן יפשטו את הרגל.

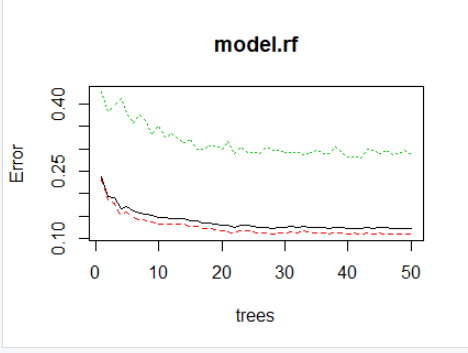
**Decision Tree**

  
:שגיאת התרשים  
  
  
  
precsion and recall

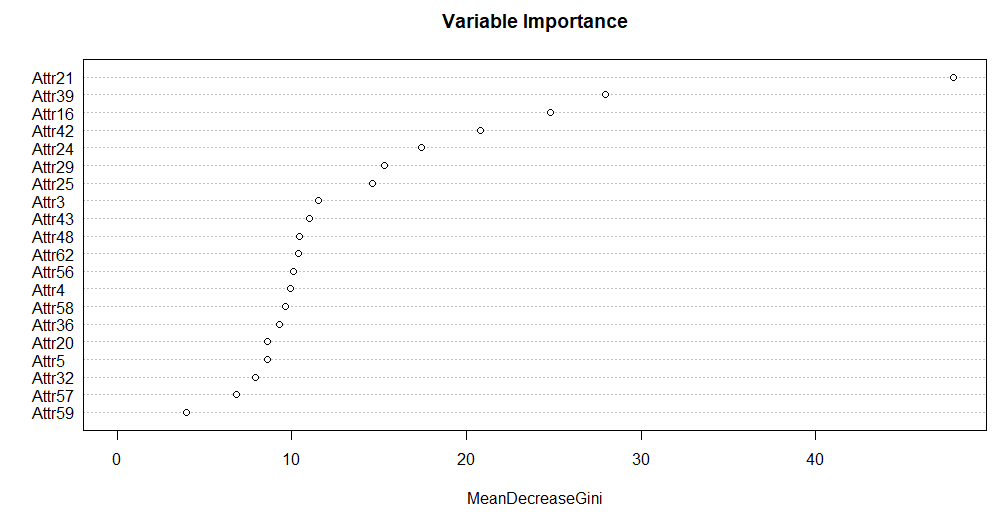
מסקנות מהמודל: ב90% מהמקרים שהמודל חוזה שאכן העסק יפשוט רגל הוא אכן צודק,  
בנוסף המודל מצליח לחזות רק ב-30% הצלחה עסקים שאכן יפשטו את הרגל אמנם אחוז הדיוק בפונקציית החיזוי של העץ היא גבוהה, אך עדין אנו מצליחים למצוא רק 30%   
מהמקרים זה עדיין לא מספיק.

**Random forest**

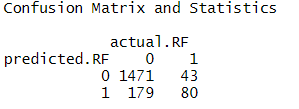
תחילה, ניתן לראות כי כאשר בחרנו ב150 עצים אחוז הטעות היה רק 6.16%.

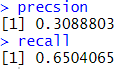
בנוסף, ניתן לראות מהגרף הבא כי כאשר אנו עוברים את ה50 עצים, אחוז הטעות נשאר קבוע באיזור ה5%.

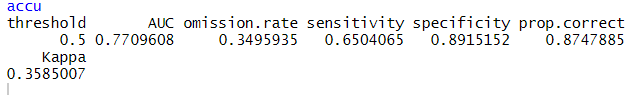
לכן נחתוך את היער ומראש נגדיר לאלגוריתם רק 50 עצים.

לאחר מכן בדקנו את חשיבות התכונות לפי סדר במודל הנ"ל:

ניתן לראות כי התכונה החשובה במודל לפי תרשים זה הינה Attr21 וכן הלאה .  
להלן תוצאות המודל:

Precsion and Recall



Accuracy

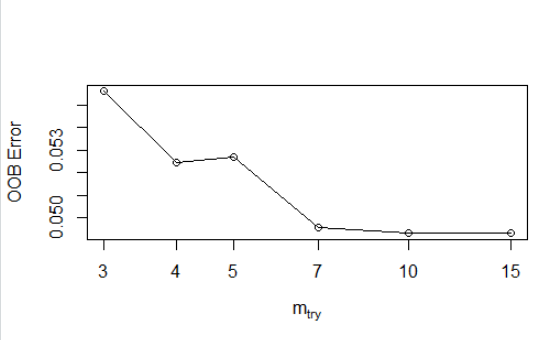
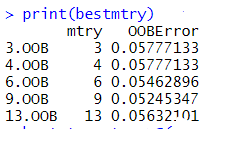
לאחר בחינה של תוצאות המודל ואף ביצוע מחקר מעמיק באינטרנט בכדי לנסות לשפר את

תוצאות המודל של ה- Random Forest מצאנו דרך אשר הובילה לשינוי בתוצאות.

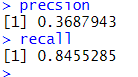
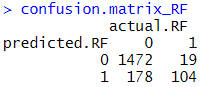
ביצענו בדיקה בכדי לגלות מהם מספר הענפים והפיצולים האופטימאלי בכל עץ,משמעות של

מספר הענפים האופטימלי הינו – מספר המשתנים שנדגמו באופן אקראי.

בבדיקה זו נמצא כי מספר פיצולים האופטימאלי הינו 10.



להלן תוצאות המודל:

 precsion and recall

  
accuracy **ניתן לראות כי בעקבות התהליך שבוצע על מנת להקטין את מספר הענפים והפיצולים תרם   
לשיפור תוצאות המודל באופן מובהק.**מסקנות מהמודל: ב-36% מהמקרים שהמודל חוזה שאכן העסק יפשוט רגל הוא אכן צודק  
בנוסף המודל מצליח לחזות רק ב-84.5% הצלחה שעסקים שאכן יפשטו את הרגל.

**KNN model ((אלגוריתם שכן קרוב**

**אלגוריתם השכן הקרוב** או  ***k*-Nearest Neighbors algorithm** (או בקיצור **k-NN** ) הוא אלגוריתם חסר [פרמטרים](https://he.wikipedia.org/wiki/%D7%A4%D7%A8%D7%9E%D7%98%D7%A8_%D7%A1%D7%98%D7%98%D7%99%D7%A1%D7%98%D7%99) ל[סיווג](https://he.wikipedia.org/wiki/%D7%A1%D7%99%D7%95%D7%95%D7%92_(%D7%A1%D7%98%D7%98%D7%99%D7%A1%D7%98%D7%99%D7%A7%D7%94)) ול[רגרסיה מקומית](https://he.wikipedia.org/wiki/%D7%A8%D7%92%D7%A8%D7%A1%D7%99%D7%94_%D7%9E%D7%A7%D7%95%D7%9E%D7%99%D7%AA). בשני המקרים הקלט תלוי ב-k התצפיות הקרובות במרחב התכונות . k-NN יכול לשמש לסיווג או לרגרסיה:

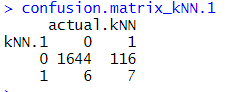
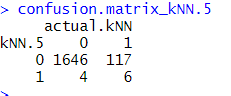
* **k-NN לסיווג** – בהינתן קלט של דוגמה חדשה, האלגוריתם משייכה לקבוצה. הדוגמה משויכת למחלקה הנפוצה ביותר בקרב k השכנים הקרובים (כאשר k מוגדר כמספר חיובי שלם, בדרך כלל מספר קטן). אם k=1 האובייקט משויך למחלקה של השכן הבודד הקרוב ביותר.
* **k-NN לרגרסיה** – בהינתן דוגמה חדשה, האלגוריתם מחזיר ערך מאפיין לדוגמה. ערך זה הוא ממוצע ערכים של ערכי k השכנים הקרובים ביותר.

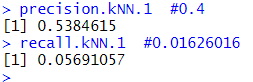
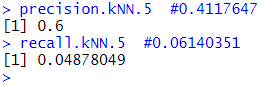
k-NN הוא אלגוריתם לימוד מבוסס מופעים, או [למידה עצלה](https://he.wikipedia.org/w/index.php?title=%D7%9C%D7%9E%D7%99%D7%93%D7%94_%D7%A2%D7%A6%D7%9C%D7%94&action=edit&redlink=1), שבו הפונקציה מקורבת באופן מקומי בלבד וכל החישובים נדחים עד סיווגה. אלגוריתם k-NN הוא מבין האלגוריתמים הפשוטים ביותר בתחום [למידת המכונה](https://he.wikipedia.org/wiki/%D7%9C%D7%9E%D7%99%D7%93%D7%AA_%D7%9E%D7%9B%D7%95%D7%A0%D7%94).

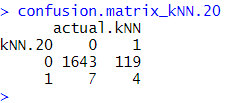
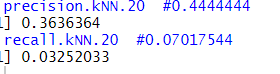
אופן שימוש האלגוריתם :השכנים נלקחים מתוך סדרת אובייקטים של מחלקה (עבור k-NN לסיווג ( או אפיון הערך  
)עבור k-NN לרגרסיה( ידועים.  
חסרון האלגוריתם: חיסרון בולט של האלגוריתם הוא רגישותו למבנה המקומי של הנתונים.

להלן תוצאות המודל**:** במודל שלנו בחרנו לחקור 3 סוגים של ערכים (שכנים ) – 1, 5, ו-20.

**Knn = 1 Knn = 5**



**Precision and Recall Precision and Recall**

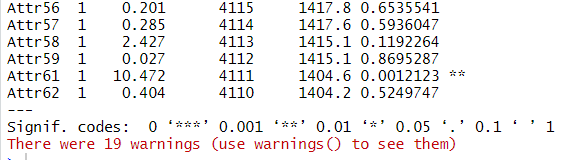
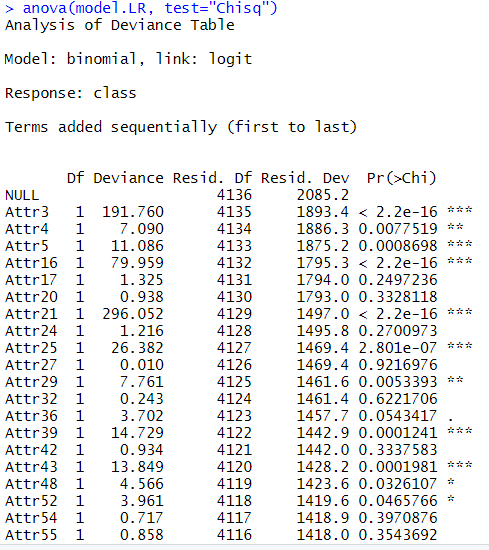
 **Knn = 20**

**Precision and Recall**

מסקנות ממודל שכן קרוב**:** ניתן לראות כי כאשר ערך ה-Knn שווה ל-5 , המודל נותן תוצאות טובות יותר.

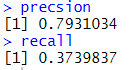
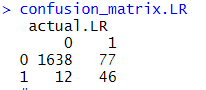
**Logistic Regression**

תחילה, ביצענו מבחן אנובה בכדי לראות האם קיימת התאמה בין העמודות.



ניתן לראות כי קיימות עמודות אשר תוצאותן זהות לעמודות אחרות.

להלן תוצאות המודל :

  **Precision and Recall**

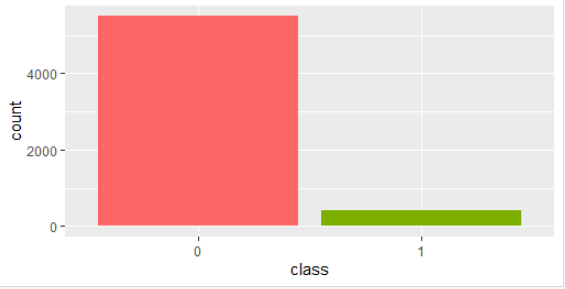
מסקנות מהמודל: ב79% מהמקרים שהמודל חוזה שאכן העסק יפשוט רגל הוא אכן צודק.  
בנוסף המודל מצליח לחזות רק ב-37.5% הצלחה עסקים שאכן יפשטו את הרגל.

**מודלים של איזון המידע**

בעקבות מחקר מעמיק שביצענו על המודלים שבוצעו עד כה , הגענו למסקנה כי אנו לא כל כך מרוצים מתוצאות המודלים. בכדי לנסות ולשפר את תוצאות המודלים, החלטנו לנסות ולאזן את נתוני המודל . כמו שראינו בתחילה , קיים הבדל ניכר בין ערכי עמודת ה-Target שלנו, מה שגורם מצב של חוסר איזון בנתונים.

להלן תרשים המשקף את איזון הנתונים לפני ואחרי:

**After Boosting Before Boosting**

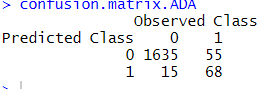


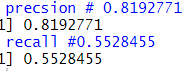
בכדי טפל במצב זה, ביצענו מחקר באינטרנט על מנת למצוא אילו מודלים מתאימים במצב של חוסר באיזון נתונים. בחרנו לבחון שלושה מודלים אשר לטעמינו נראו הכי מעניינים.  
ההמודלים שבחרנו היינם: AdaBoost , Tree Bag ו-)Naive Bayesשאותו אנו מכירים מההרצאות ומתרגילי הבית).

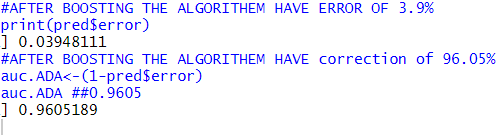
**AdaBoost Model**

,AdaBoost קיצור של [Boosting](https://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.com&sl=en&sp=nmt4&u=https://en.m.wikipedia.org/wiki/Boosting_(meta-algorithm)&xid=17259,15700021,15700186,15700191,15700256,15700259,15700262,15700265,15700271&usg=ALkJrhjNmNeUITnkMyIzZzFRulvqCDRIXA) Adaptive , הוא [מטא-אלגוריתם](https://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.com&sl=en&sp=nmt4&u=https://en.m.wikipedia.org/wiki/Meta-algorithm&xid=17259,15700021,15700186,15700191,15700256,15700259,15700262,15700265,15700271&usg=ALkJrhgaMM732xrxVUADZYKdajcaeAig0A) [למידת מכונה](https://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.com&sl=en&sp=nmt4&u=https://en.m.wikipedia.org/wiki/Machine_learning&xid=17259,15700021,15700186,15700191,15700256,15700259,15700262,15700265,15700271&usg=ALkJrhhcbbYQgb7OMOt9pBFct0hsuZu_9Q) . ניתן להשתמש בו בשילוב עם סוגים רבים אחרים של אלגוריתמי למידה כדי לשפר את הביצועים. הפלט של אלגוריתמי הלמידה האחרים ('הלומדים החלשים') משולב לסכום משוקלל המייצג את התפוקה הסופית של המסווג המוגבר. AdaBoost הוא אדפטיבי במובן זה שלומדים חלשים הבאים עוקבים לטובת המקרים המסווגים על ידי סיווגים קודמים. AdaBoost רגיש לנתונים [ולמחשבים](https://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.com&sl=en&sp=nmt4&u=https://en.m.wikipedia.org/wiki/Outlier&xid=17259,15700021,15700186,15700191,15700256,15700259,15700262,15700265,15700271&usg=ALkJrhi59Jg0TWXOuU9yKCvEZsc1MI6YJg) רועשים. בחלק מהבעיות זה יכול להיות פחות רגיש לבעיית [ההישג](https://translate.googleusercontent.com/translate_c?depth=1&hl=iw&prev=search&rurl=translate.google.com&sl=en&sp=nmt4&u=https://en.m.wikipedia.org/wiki/Overfitting_(machine_learning)&xid=17259,15700021,15700186,15700191,15700256,15700259,15700262,15700265,15700271&usg=ALkJrhiPx1ac4QFc6wtXzY_h_GCLuov2yQ) יתר מאשר אלגוריתמי למידה אחרים. הלומדים האינדיבידואליים יכולים להיות חלשים, אך כל עוד הביצועים של כל אחד מהם מעט טובים יותר מאשר ניחוש אקראי, ניתן להוכיח כי המודל הסופי יתכנס למלומד חזק.

**להלן תוצאות המודל:**

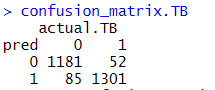
**Precision and Recall**

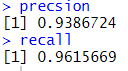


שגיאת האלגוריתם , ואחוז התאמת האלגוריתם:

מסקנות מהמודל: ניתן לראות כי כאשר מבצעים מניפולציות על מנת לאזן את הנתונים במודל , התוצאות גדלות. לכן כנראה קיימת בעיה של **חוסר איזון בנתונים** בDataSet שלנו.

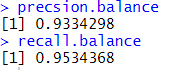
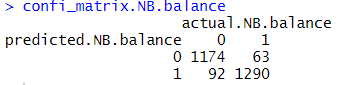
**Tree Bag Model**

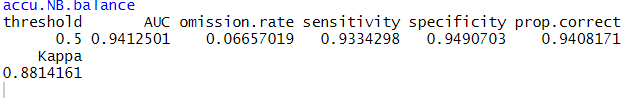
**Precision and Recall**



מסקנות מהמודל: ב93% מהמקרים שהמודל חוזה שאכן העסק יפשוט רגל הוא אכן צודק.  
בנוסף המודל מצליח לחזות רק ב-96% הצלחה עסקים שאכן יפשטו את הרגל.  
ניתן להבחין כי איזון הנתונים משפיע משמעותית על אמינות המודל.

**Naive Bayes (After Balacne)**

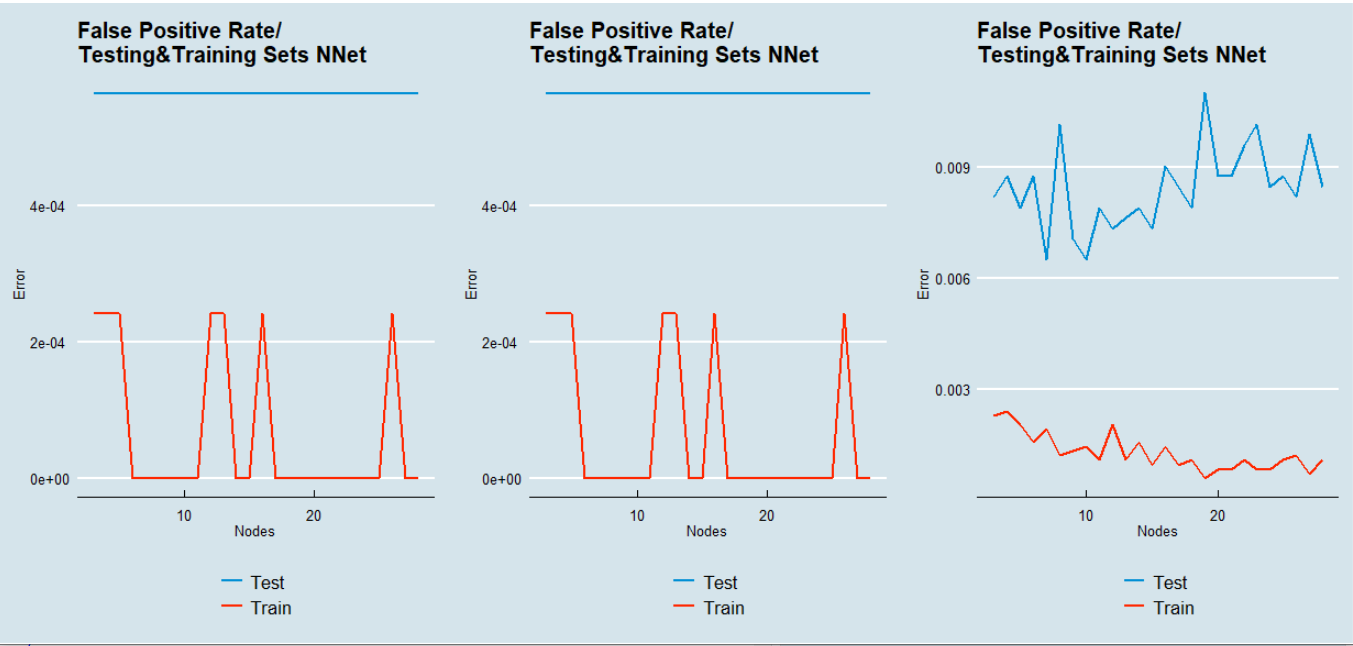
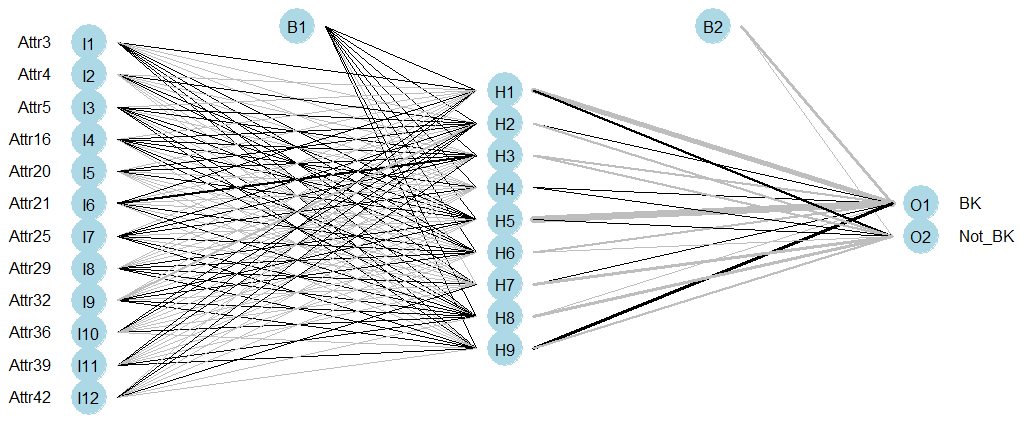
**Precision and Recall**

accuracy

מסקנות מהמודל: ב93% מהמקרים שהמודל חוזה שאכן העסק יפשוט רגל הוא אכן צודק.  
בנוסף המודל מצליח לחזות רק ב-95% הצלחה עסקים שאכן יפשטו את הרגל.  
ניתן להבחין כי איזון הנתונים משפיע משמעותית על אמינות המודל.

**Neural nets**

מודל נוסף שהחלטנו לחקור וללמוד עליו הוא רשת נוירונים, מודל זה הינו המודל האחרון והקשה מכולם, במודל זה ביצענו מחקר מעמיק אודות רשת הנוירונים אשר זכינו לראות בהרצאה הראשונה והסתקרנו מתכונותיה. במודל יצרנו 25 תרשימים להשוואה.

להלן דוגמא מהתרשיםתרשים מספר 7

ראינו כי בTrain set- קיימת שגיאה לא יציבה, ביצענו מספר אלגוריתמים בכדי לייצב את השגיאה ואכן השגנו מטרה זו (**להלן התרשים מלעיל** . רשת הנוירונים מזהה באופן מובהק את בעיית הפרויקט, האם יפשטו רגל או לא.

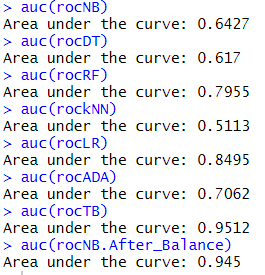
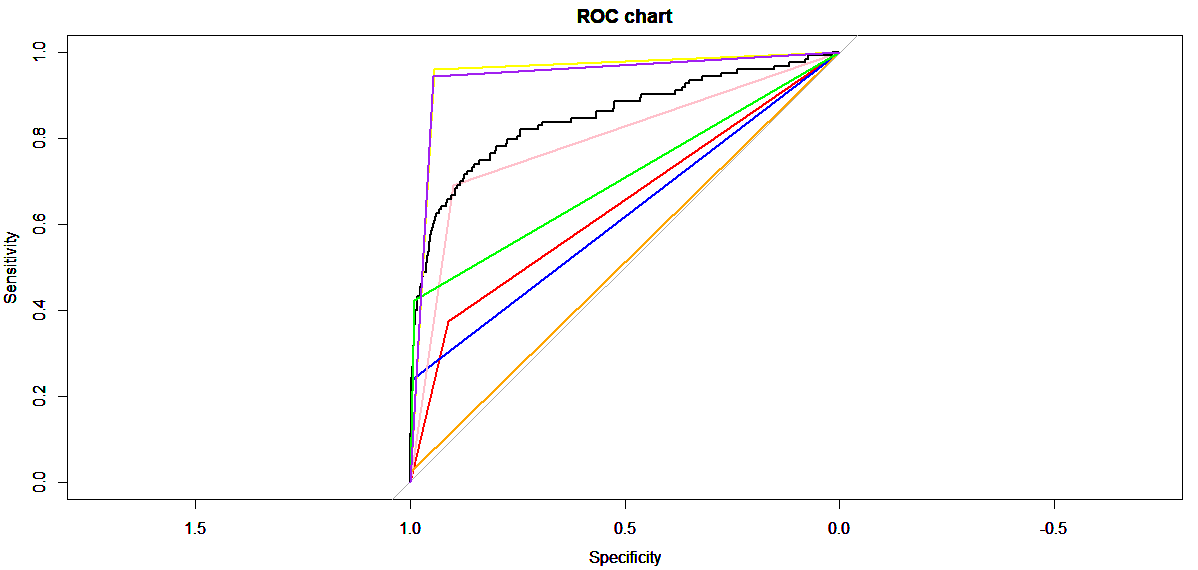
**1**

**2**

**33**

**מסקנות המודלים**

בשלב הבא ביצענו בדיקת **rock chart** לצורך השוואה בין המודלים.  
התוצאות שהתקבלו הן:



**אדום** – Naive Bayes **|**  **כחול**- Decision Tree **|** **שחור** – logistic regression **| ורוד**- RF

**כתום** – Knn  **|**  **ירוק** - ADAboost **|**  **צהוב** – TreeBag | **סגול** - Naive Bayes after balance

מתוצאות ההשוואה ע"פ התרשים הנ"ל ניתן לראות שהמודל הטוב ביותר לצורך בדיקה זו הוא:  
TreeBag , שבו השטח שמתחת לגרף הוא **95.12%.**  
מודל נוסף אשר נותן תוצאה גבוהה הינו : בייס נאיבי המאוזן (לאחר איזון נתוני עמודת המטרה) שבו השטח שמתחת לגרף הוא **94.5%**, בנוסף גם המודל של רגרסיה לוגיסטית שבו השטח מתחת לגרף היינו : **84.95%.**

**סיכום**

מעבודה זו הפקנו רבות על כריית ידע ולמידת מכונה, התמודדנו עם כמות רבה של נתונים פיננסיים, הגדרנו זמנים ויעדים מראש, ביצענו את העבודה בעזרת שיתוף פעולה בין חברי הצוות, ולמדנו רבות אודות ניתוח וניקיון נתונים, וכל זאת במטרה לבצע ניתוח מדויק של מידע השמור בדאטה פריים ואכן להגיע לחיזוי ותוצאות ברמת דיוק גבוהה לעמודת המטרה.  
בנינו מודלים אשר לא נלמדו במהלך הקורס על מנת לחקור את הנושא לעומק וזאת בכדי להשיג תוצאות אופטימליות. הגענו למסקנה כי איזון הנתונים משפיע משמעותית על אמינות המודל. לבסוף קיבלנו תוצאות המסבות לנו סיפוק רב באמצעות מחקר מעמיק ויצירת מודלים מעניינים.