שם הקורס: מבוא ללמידת מכונה

מרצה: דור בנק

מתרגל: אילן וסילבסקי

פרויקט מסכם – רכישות באתר E-Commerce

שמות המגישים: ירדן וולף - 209423284 וגלעד שפיר 319092235

תאריך הגשה: 09.06.22

תקציר מנהלים:

מטרת הפרויקט הייתה להעריך את ההסתברות של סשנים באתר קניות E-Commerce באינטרנט להסתיים ברכישה או בלי. בפרויקט בנינו מערכת המנבאת את הסיכוי של משתמש מסוים, תחת תנאים מסוימים, לבצע רכישה בזמן הגלישה באתר.
הפרויקט עוסק בבעיית קלסיפיקציה בינארית, שבה כל סשן יכול להסתיים ב-2 דרכי סיווג בלבד. קיבלנו מספר פיצ'רים, עבור חלקם ידענו את המשמעות ואחרים היו אנונימיים. כדי למקסם את השימוש בכל פיצ'ר, ניסינו להגיע למסקנות לגבי משמעותם ופרשנותם של כל פיצ'ר.
בפרויקט השתמשנו רבות בחומר הנלמד בקורס וביצענו לפיו הנחות רבות שעזרו לנו במימושו.
כאמור, לצורך חיזוי מיטבי, ביצענו תהליך זהה על נתוני המבחן ונעזרנו בעיבוד המקדים של נתוני האימון על מנת לעבדם.
חשוב לנו לציין את מעגליות התהליך. ביצענו מספר איטרציות של 4 השלבים הראשונים, כאשר בכל איטרציה הסתמכנו על תוצאות האיטרציות הקודמות, וביצענו שיקולים כגון אילו ערכים כדאי להסיר או איך כדאי לנרמל את הנתונים כדי לתת בסופו של דבר את התחזיות המיטביות מהמודלים שלנו.
אמנם לא התעמקנו בכך בפרויקט, אך לדעתנו ישנה גם חשיבות גדולה מאוד להבנה עסקית של אתר מכירות באינטרנט כתוצאה מהפרויקט שעשינו לשיפור ביצועי העסק שלו.

חלק 1 – אקספלורציה

תהליך האקספלורציה נעשה במטרה לחקור את ה-data, לקבל הבנה על האופי שבו כל פיצ'ר מתפלג, האם קיימת התנהגות קורלטיבית בין הפיצ'רים השונים וניתוח נתונים סטטיסטים על כל פיצ'ר.

התחלנו בקריאת ה- data והצגתו בטבלה. מטרת ההצגה הייתה לקבל הרגשה וידע לגבי הנתונים ואיך הם נראים בפועל. בדקנו את סוג הנתונים (int, float, object, bool) תחת כל משתנה על מנת שנוכל בסוף תהליך העיבוד המקדים לקבל data אחיד ומספרי לצורך הרצת המודלים.

באמצעות מספר שיטות ויזואליזציה הצגנו כיצד כל הפיצ'רים מתפלגים (לצורך זיהוי חריגים). עבור משתנים קטגוריאליים בדקנו כיצד הנתונים מתפלגים עבור כל קטגוריה (לצורך מילוי ערכים חסרים בשלב הבא). ביצענו ויזואליזציה נוספת עבור משתנים שחשבנו שיתאימו להיות משתני דמי על מנת לראות אילו ערכים הכי נפוצים.

הוספנו ויזואליזציות של heatmap שיראו לנו את הקורלציה בין הפיצ'רים השונים לצורך מילוי ערכים חסרים וכן גם הבנה טובה יותר של ה-data. בנוסף, בדקנו את כמות הערכים החסרים עבור כל אחד מהפיצ'רים.

חלק 2 - עיבוד מקדים

לאחר קבלת הבנה מעמיקה של ה-data, נוכל לבצע את תהליך העיבוד המקדים ולהתאים אותו עבור הרצת המודלים בחלק הבא.

עבור פיצ'רים מטיפוס object, הורדנו מילים לא רלוונטיות והפכנו אותם למספריים, קטגוריאליים או בוליאנים בהתאם לאופיים.
לאחר מכן התקדמנו להתמודדות עם ערכים חסרים ב-data. החלטנו להסיר פיצ'רים בעלי מספר גבוה מאוד של ערכים חסרים (למעלה מ-90% מהנתונים ריקים).
בהתאם לקורלציה שהבחנו בה באקספלורציה בין פיצ'רים מסוימים, השלמנו ערכים בהתבסס על ממוצע או חציון בהתאם לאופיים, לאחר שחילקנו את הפיצ'ר שלפיו מילאנו את הערכים ל-2 קטגוריות נפרדות (למשל חילקנו לכמות דפי מוצרים גדולה וקטנה לפי חציון כאשר מילאנו את total duration לפי num of product pages). הרצנו תהליך זה עבור כל זוגות הפיצ'רים בעלי קורלציה הגבוהה מ-0.6 ביניהם.
עבור חלק מהפיצ'רים שזיהינו שצריכים להיות קטגוריאליים באקספלורציה, השלמנו ערכים חסרים לפי הערך הנפוץ ביותר באותו הפיצ'ר. כאשר היו מספר ערכים בעלי אותה שכיחות גבוהה בנתונים, מילאנו ערכים לפי "מילוי קדימה/אחורה" כדי שלא לשנות יותר מדי את התפלגות הקטגוריות בפיצ'ר.
לאחר מכן עברנו להתמודדות עם outliers. בחלק זה הסתמכנו רבות על התפלגות הפיצ'רים באקספלורציה. עבור פיצ'רים שהתפלגו בצורה גאוסיאנית או מעריכית, בחרנו להוריד את מאית הסשנים בעלי הערכים הקיצוניים ביותר. ההנחה הייתה שדרך זו לא תקטין לנו את ה-data באופן משמעותי, אך כן תשפר משמעותית את ביצועי המודלים בהמשך באמצעות התעלמות מערכים בעייתיים שעלולים לעוות את התוצאות שיתקבלו מהם.

השלב הבא היה הפיכת משתנים קטגוריאליים למשתני דמי. הסיבה שהפכנו חלק מהמשתנים הקטגוריאליים למשתני דמי הייתה שלא רצינו לתת משמעות שונה לערכים שונים (למשל, לא רצינו שסשנים באתר שהתבצעו בחודש דצמבר יקבלו משקל גדול בהרבה במודלים מאשר סשנים שקרו בחודש ינואר). בנוסף, עבור משתנים שלקטגוריות שלהם אין ערכים מספריים, רצינו לפצל כל סוג סשן לקטגוריה המתאימה לו (למשל סוג הדפדפן שממנו נכנס הלקוח לאתר). כמו כן, היו כמה משתנים קטגוריאליים בעלי

מספר קטגוריות גבוה מאוד. עבור פיצ'רים אלו, צמצמנו את כל הקטגוריות שהופיעו מספר נמוך של מופעים אל תוך קטגוריה אחת שאותה כינינו other (למשל עבור internet browser, כל דפדפן שנספרו בו פחות מבערך 200 כניסות נכנס לקטגוריית other browser).

לאחר תהליך זה, חילקנו את ה-data ל-X_train ו - y_train על מנת להוריד מימדים של המשתנים המסבירים בהמשך וביצוע תחזיות בהסתמך על ה-Train data.
כאמור, השלב הבא היה נרמול המשתנים המסבירים. בחרנו לנרמל בשיטת MinMaxScalar אשר מטרתה היא נרמול ה-data בין 0 ל-1. הסיבה שבחרנו בשיטה זו היא משום שרצינו שהנתונים יראו דומים לכל פי'צר. תהליך זה מסיר data שלא מעובד היטב או שכפולים כדי להבטיח שרק נתונים הגיוניים לוגית יישמרו.

משם עברנו להורדת המימדיות של הבעיה. לאחר תהליך העיבוד המקדים עד עתה, הגענו ל -61 פיצ'רים. בקורס למדנו שהבעיה המרכזית בעולם למידת המכונה היא Bias-Variance TradeOff. מספר רב של פיצ'רים עלול לגרור עימו עלייה משמעותית בשונות שלא תפוצה ע"י הורדת ה-bias שבאה בעקבותיו. לכן, החלטנו לצמצם את כמות הפיצ'רים שבחרנו תוך כדי שאנו שומרים על כמה שיותר מיכולת ההסבר של הפיצ'רים על המשתנה שאותו אנו רוצים לחזות.
בקורס למדנו שתי שיטות מרכזיות לעשות זאת: PCA ו-Feature Selection. ביצענו את שתיהן והשווינו ביניהן. ע"ס התוצאות שקיבלנו (בחינת ה-MSE שהתקבל על ה-Train Data), ראינו כי השיטה המוצלחת יותר הייתה Forward Feature Selection אשר הובילה אותנו ל-20 פיצ'רים. בסוף תהליך הורדת המימדיות, שמרנו רק את הפיצ'רים הנבחרים.

כדי להתאים את ה-Test Data למודלים שנבצע בשלבים הבאים, הרצנו עליו את כל העיבוד המקדים שעשינו עד כה לפי ה-Train Dta.

חלקים 3+4 - הרצת והערכת המודלים:

ראשית, חילקנו את ה-Train Data לחמישית Validation וארבע חמישיות Train. זאת לצורך ביצוע Confusion Matrix על המודל האופטימלי שבחרנו.

עבור כל אחד מ-4 המודלים שבחרנו להציג, התחלנו מבחירת ההיפרפרמטרים הטובים ביותר באמצעות שימוש ב-K-Fold ובפונקציית Grid Search כאשר הכנסנו מספר אפשרויות בחירה לתוך כל היפרפרמטר עיקרי שמקבל המודל בעת הרצתו.

לאחר שקיבלנו את ההיפרפרמטרים הטובים ביותר, שמרנו אותם וביצענו עליהם K-Fold Cross Validation כדי להציג את ערך ה-auc של כל אחד מהפיצולים בכל אחד מהמודלים שבחרנו.
כמו כן, במטרה לבדוק האם מתקיים OverFitting במודלים, הרצנו את המודל עם ההיפרפרמטרים הנבחרים על ה-Train Data וביצענו את החיזוי על ה-Train. הסיבה לכך היא שהנחנו שפערים משמעותיים בתוצאות שבין החיזוי על ה-Train וה-Test יכולים להעיד על בעיית OverFitting במודל. על כל מודל הוספנו אילוסטרציה כיצד הוא מסווג את ה-Test Data לאחר הרצת המודל באמצעות בחירת 2 הפיצ'רים המשמעותיים ביותר לכל מודל.

המודל הראשון שבחרנו הוא SVM. ההיפרפרמטרים שנבחרו על הנתונים הם: {'C': **0.1**, 'gamma': **'auto'**, 'kernel': **'rbf'**}. לאחר ביצוע K-Fold Cross Validation קיבלנו ערך AUC ממוצע של 0.88. מביצוע בדיקת OverFitting על מודל זה, קיבלנו כי אין פער משמעותי בין התוצאות המתקבלות על ה-Train ועל ה-Test ומכך הסקנו כי אין בהכרח בעיית OverFitting במודל.

המודל השני שבחרנו הוא Random Forest. ההיפרפרמטרים שנבחרו הם: {'criterion': **'gini'**, 'max_depth': **70**, 'min_samples_leaf': **2**, 'min_samples_split': **3**, 'n_estimators': **200**}. לאחר ביצוע K-Fold Cross Validation קיבלנו ערך AUC ממוצע של כמעט 0.9. עבור מודל זה הוספנו בדיקת חשיבות הפיצ'רים להחלטות המודל. קיבלנו 2 פיצ'רים משמעותיים במיוחד (PageValues ו - ExitRates) המהווים יחד 90% מיכולת ההסברה של המודל על המשתנה החזוי מתוך 20 הפיצ'רים שנבחרו ע"י הורדת המימדיות. מביצוע בדיקת OverFitting על מודל זה, קיבלנו כי קיים פער של 0.08 ב-AUC המתקבל מה-Train לעומת ה-Test. הדבר מצביע על כך שייתכן ויש בעיית OverFitting במודל.

המודל השלישי שבחרנו הוא Logistic Regression. ההיפרפרמטרים שנבחרו הם: {'penalty': **'l1'**, 'solver': **'saga'**}. לאחר ביצוע K-Fold Cross Validation קיבלנו ערך AUC ממוצע של 0.88. עבור מודל זה הוספנו את המשקולות המתאימות לכל פיצ'ר במודל. קיבלנו ש-2 הפיצ'רים בעלי המשקולות המשמעותיות ביותר היו PageValues ו- A_c_9. מביצוע בדיקת OverFitting על מודל זה, קיבלנו כי אין פער משמעותי בין התוצאות המתקבלות על ה-Train ועל ה-Test ומכך הסקנו כי אין בהכרח בעיית OverFitting במודל.

המודל הרביעי והאחרון שבחרנו הוא KNN. ההיפרפרמטרים שנבחרו הם: {'algorithm': **'auto'**, 'n_neighbors': **5**, 'p': **1**}. לאחר ביצוע K-Fold Cross Validation קיבלנו ערך AUC ממוצע של כמעט 0.8. מביצוע בדיקת OverFitting על מודל זה, קיבלנו כי קיים פער של 0.12 ב-AUC המתקבל מה-Train לעומת ה-Test. הדבר מצביע על כך שייתכן ויש בעיית OverFitting במודל.

ראינו כי Random Forest מחזיר לנו את החיזויים הטובים ביותר ועל כן בו בחרנו להרצת קובץ ה-Test. ביצענו עליו Confusion Matrix, כאשר החיזוי נעשה על סט ה-Validation שיצרנו קודם לכן. התוצאות שהתקבלו הראו כי ציון ה-Accuracy שלנו היה קצת מעל 90%. קיבלנו כי מתוך 1952 תצפיות סיווגנו נכון כ-1764 תצפיות.

חלק 5 - ביצוע פרדיקציה:

התחלנו מהגדרת הפונקציות שבהן השתמשנו ב-pipeline. בחלק זה הצגנו את התהליך של טעינת הנתונים ועיבוד מקדים. שחזרנו את התוצאות האופטימליות שקיבלנו בחלקים הקודמים (שימוש במודל Random Forest עם ההיפרפרמטרים הנבחרים). לבסוף, ביצענו חיזוי על קובץ ה- Test data שבו הצגנו את ההסתברות של כל session להסתיים ברכישה באתר. את תחזיות המודל על ה-Test ניתן ייצאנו לקובץ csv כנדרש.

סיכום:

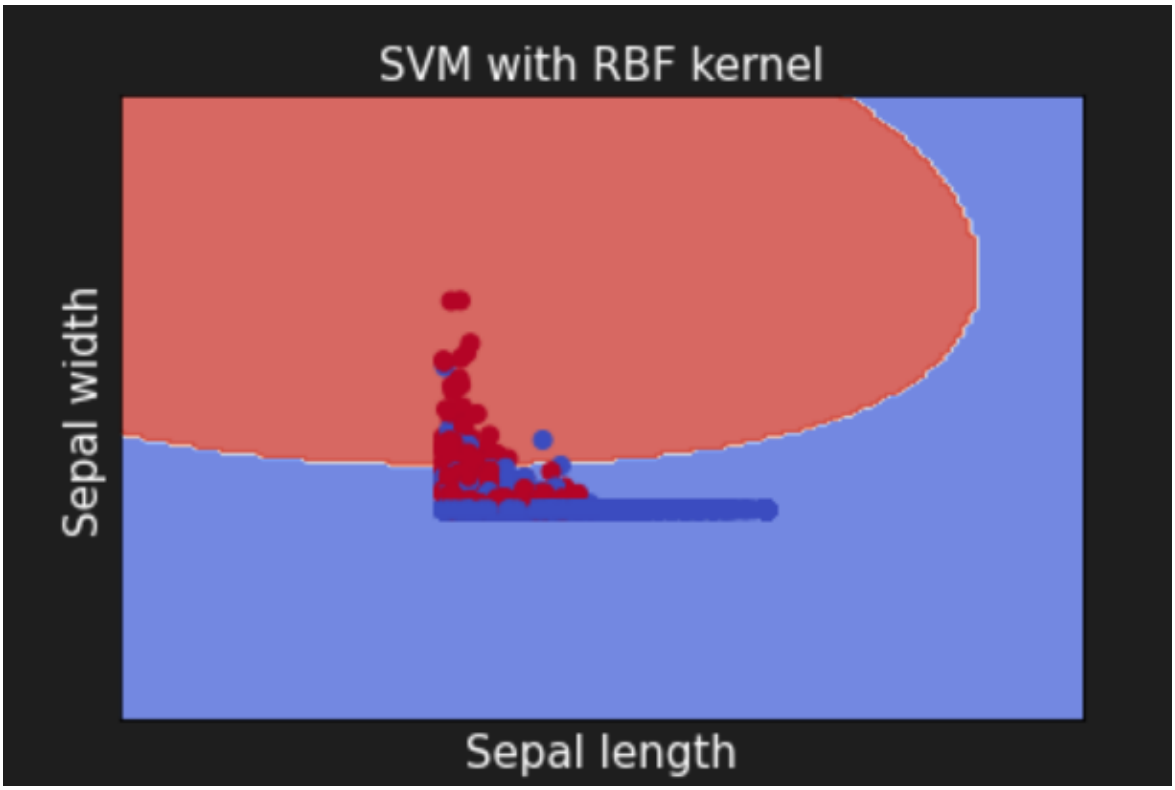
בפרויקט דנו במספר מודלים והגענו לכמה מסקנות במהלך ניתוח הנתונים.
לפני אמידת המודלים, הגענו למסקנה חשובה הקשורה בפתרון בעיית המימדיות של הבעיה. ראינו כי באחת האיטרציות הראשונות שביצענו, צמצום מימדים בשיטת PCA שבה ניסינו לשמור כ-99% משונות המודל החזיר לנו 2 מימדים! הבנו כי ייתכן ובחירה בשיטה זו תוכל להחזיר לנו מודל מאוד פשוט, אך נאבד את כל המידע הקשור במימדים השונים של הבעיה ולא נוכל לקבוע את ההשפעות שלהם בנפרד על סיווג הבעיה.

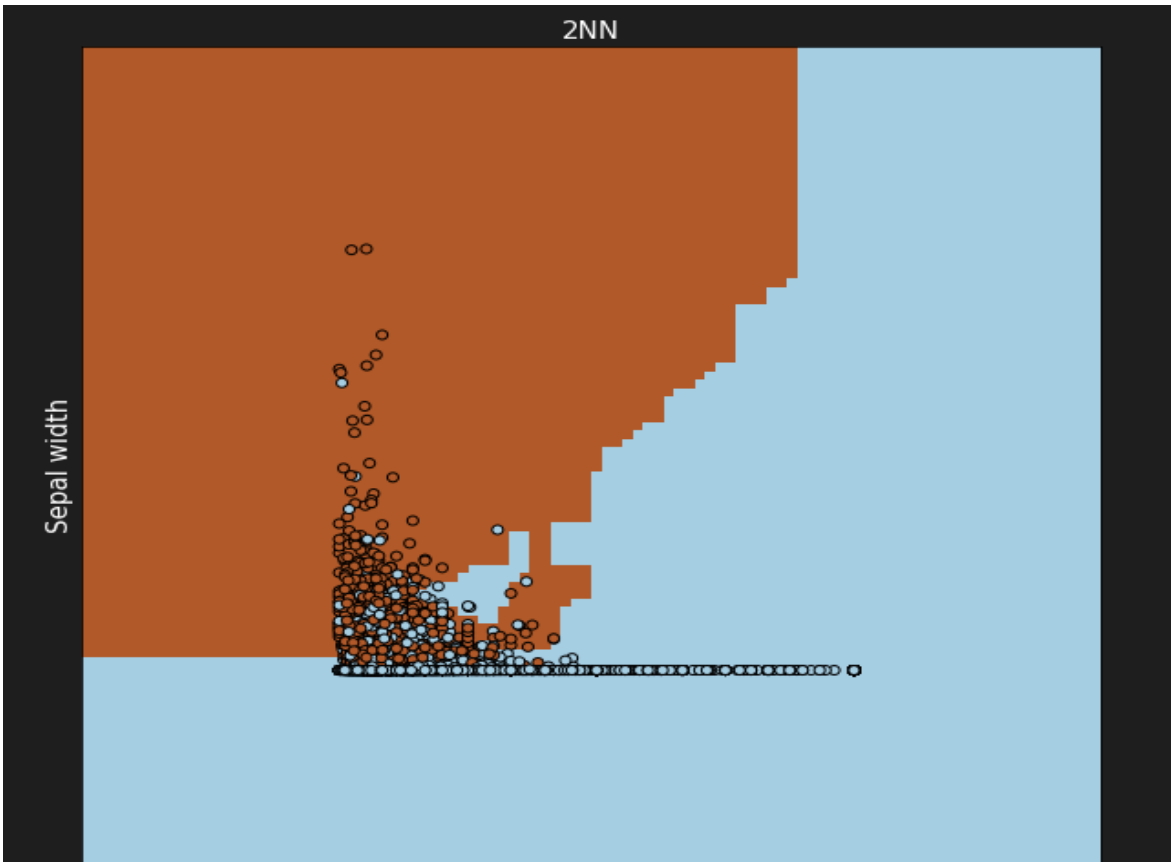
מסקנה נוספת שקיבלנו בעקבות העבודה על התהליך היא הפער הביצועי המתרחש לרוב בין המודלים הראשוניים שבחרנו למודלים המתקדמים. לרוב, המודלים המתקדמים נתנו תוצאות יותר טובות בצורה משמעותית. הדבר אינו בהכרח מפתיע, אך מעניין לראות שהדבר באמת קורה בפועל.

בפרויקט השתמשנו בשיטת מדידת תוצאות של AUC תחת ROC Curve. בעת עבודות חיפוש למען הפרויקט, מצאנו שיטות אמידת חיזוי נוספות. על אף שזו השיטה הנהוגה, ייתכן והיינו מגלים תוצאות מעניינות נוספות באמצעות שיטות scoring אחרות.

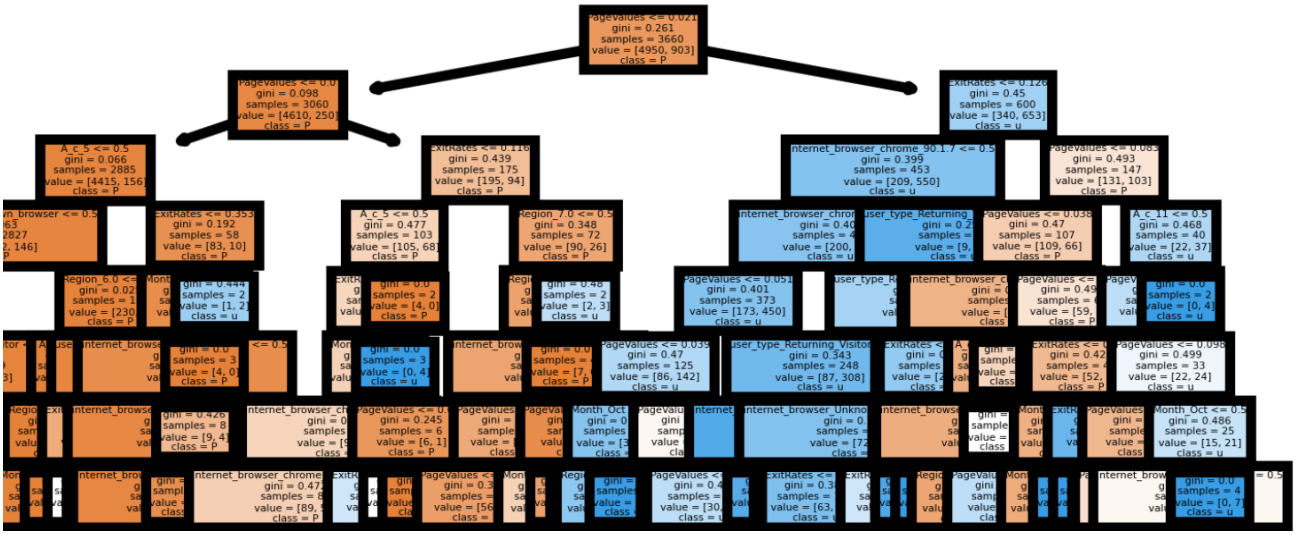
נספחים:

ויזואליזציה של SVM ו-KNN על הנתונים:





ויזואליזציה של הפיצולים ההתחלתיים על המודל הנבחר Random Forest על סט הנתונים:



(מופיעות בסוף כל הרצת מודל)

ויזואליזציות נוספות ניתן לראות בגוף הקוד (בשלבי האקספלורציה של הנתונים והערכת המודלים).

הסבר אחריות כל שותף ותרומתו לעבודה:

הפרויקט ברובו בוצע יחדיו, אך כל אחד הביא את נקודות החוזקה שבו עבור החלקים השונים. ירדן ניהל את חלק כתיבת הקוד שבוצע. את תהליך הסקת המסקנות וההסברים שביצענו במהלך הפרויקט ניהל ברובו גלעד.