

TEL AVIV אוניברסיטת
UNIVERSITY תל אביב

הפקולטה להנדסה

פרויקט גמר- תחזית גשם

במסגרת הקורס "מבוא ללמידת מכונה"

המרצה: מר' דור בנק

מתרגל: מר' יעד טובלי

מגיש: אביתר וייס 308481324

Evyatarweiss1@gmail.com

תאריך: 16.7.20

הצהרת הבעיה

גשמים הם חלק חיוני מחיינו. השירות המטאורולוגי הישראלי היא יחידת סמך במשרד בתחבורה האחראית לחיזוי מזג האוויר. תפקיד זה הינו חשוב ביותר, מאחר ולחיזוי מזג האוויר ישנן השלכות כלכליות רבות בנושאים מגוונים מעבר לאיך עלינו להתלבש. מכאן שבפרויקט זה, אני אישם תאוריות ומודלים שלמדנו במהלך הקורס באמצעות Python-Pandas ו-Scikit-Learn, ואבנה מסווג בכדי לחזות אם יורד גשם מחר או לא לפי נתונים מסויימים.

מבנה העבודה כולל ארבעה מרכיבים עיקריים: חקר וניתוח נתונים, עיבוד נתונים, הטמעת מודלים והערכת מודלים.

תקציר

בפרויקט זה השתמשתי במודלים רבים מלמידת מכונה על מנת לחזות האם ירד גשם או לא זאת לצד סקירה וחקירה מעמיקה של הנתונים ובחירת המודלים הנכונים לנתונים בשביל תחזית בעלת AUC מקסימלי.

חקר וניתוח נתונים

חשוב לציין כי את חקירת הנתונים התחלנו עם 26 פיצרים, חלקם עם שם ידוע וחלקם אנונימיים. אם כן הפיצרים מתחלקים לקטגוריאליים וחלקם משתנים רציפים. מבדיקה מעמיקה נראה כי (Feature_14) הינו רציף אך מתנהג כקטגוריאלי וכן אליו מוצמד תו המידות MM. למרות היותו פיצר בעל שם אנונימי החלטתי לשנות אותו להוריד את הממ מכל איבר ולהפוך אותו לרציף בגלל הממ קשור למידה של כמות מים. באופן זה גם החלטה שרירותית להפוך את פיצר מספר 9 למספרי שכן הוא בבנוי ממספרים אך מתויג כקטגוריאלי. בהמשך בחקר מעמיק עוד יותר על הדאטה גיליתי שישנו קשר לא לינארי בין פיצר 13 ל14. כלומר יצרתי עמודה חדשה ולה עשיתי פונקציית למבדה (נספח 3). הפונקציה הנ"ל מראה כי האיבר בפיצר 13 הוא 1 אם ורק אם האיבר באותה שורה בעמודה של פיצר 14 הוא גדול מ1, ואם קטן מ1 הוא 0. מה שמאושש את זה הוא שלשניהם יש את אותם מספר ערכים חסרים. לכן ככל הנראה פיצר 13 הוא מניפולציה מתמטית של 14 מפה שהחלטתי להסיר את פיצר מספר 14.

ביצעתי בדיקה של כמות האיברים החסרים בכל עמודה, נראה כי לעמודת ה sunshine ישנם 1871 איברים חסרים שמהווים 8.4% מכלל העמודה, מאחר המספר לא גבוה נבחר בצעד זה לא להסיר אף עמודה שכן היא מביאה יותר תועלת מאשר חוסר בנתונים.

משתנים רציפים וקטגוריאליים

מעתה והלאה בוצעה הפרדה מלאה בין משתנים רציפים לקטגוריאליים על מנת לבצע מניפולציות שונות בהן. בראייה למרחק לרגע בו אצטרך להשלים את הנתונים החסרים למשתנים הקטגוריאליים ביצעתי ויזואליזציה לכל פיצר שכזה שמראה את ההתפלגות שלו וכמה מתוכו מתויג בלייבל 1 וכמה 0. כלומר מתוך זה ניתן להסיק את החשיבות שלו על החיזוי, דוגמה על פיצר 19 בנספח ב' שנראה שתחזית הגשם מתפלגת אצלו בהתאם תדירות האיבר. באופן זה עשיתי על המשתנים הרציפים. ששם ניתן לראות בבירור שלפיצר 13 (נספח ה'), ולפיצר Sunshine (נספח ד') ישנן עמודות שבהן ישנה נציגות גדולה של לייבל 1, ומפה שגם מעיד על טובות על חשיבות הפיצר לחיזוי.

עיבוד נתונים

לאחר חקר הנתונים נוכחנו לדעת כי ישנם לא מעט נתונים חסרים בהרבה עמודות. אם כן עלינו להשלים אותם בצורה מסוימת. לפי התפלגות הלייבל ברוב הפיצורים הנחתי כי אין חשיבות לבחירת ערך ספציפי כמו הממוצע או החציון שכן זה לא יתרום אלא רק יהרוס את המאזן. מפה שבחרתי שלהשלים את הנתונים החסרים ע"י כך שהם יקבלו ערך רנדומלי מתוך הערכים הקיימים. כמובן שלמשתנה רציף יש טווח של ערכים אינסופי ולמשתנה קטגוריאלי יש מספר סופי של אופציות.

מטריצת קורלציות

כרגע אנחנו עומדים על 25 פיצורים שכן בנספח ו' ניתן לראות את מטריצת הקורלציות שמסווגת לפי צבעים שכן צבע כהה מעיד על קורלציה שלילית וכהה על חיובית. ממטריצה זו ניתן לראות בבירור כי קיימים קשרים ליניאריים שגבוהים מ-0.8 למשל עם feature_0 1 and evaporation.

בשביל לראות את הקשרים הליניאריים בצורה ויזואלית ביצעתי pairplot (נספח ז'). כדי שהמודל יהיה יציב מספיק, השונות צריכה להיות נמוכה. וכן קורלציה של שני פיצורים מגדילה את השונות שמפה בחרתי להסיר מספר פיצורים (Year,0,1,16,17,11)

איברים חריגים

במבט מהיר, BOXPLOT מספקת אינדיקציות לסימטריה בתוך הנתונים. כלומר קל לזהות חריגים באמצעות השיטה הזו שכן חריגים שכאלו יכולים לפגוע קשות בתוצאות הניבוי. (נספח ח') לפי הנספח ניתן לראות כי ישנה כמות גדולה של חריגים במספר פיצורים שכן נניח והתפלגות הינה נורמלית, ולכן נסיר אותם באמצעות שיטת SCORE-Z הסרה של החריגים משמעותה, מסירים את השורות שבהם קיימים החריגים. אם כך, בנספח ט' ניתן לראות את ה boxplot לאחר ההסרה.

Dummy variable

משתנה זה הוא משתנה מלאכותי שנוצר לייצג פיצור קטגוריאלי באמצעות מספרים. כלומר זה תלוי במספר האיברים הייחודיים שיש לכל פיצור קטגוריאלי. על מנת שהמחשב יוכל לנבא באמצעות פיצורים קטגוריאליים ולהסיק מהם דברים, עלי לפצל אותם לפיצורים נפרדים באמצעות פונקציה זו.

Data scaling

כפי שראינו הדאטה שלנו מכיל פיצורים רבים מסוגים שונים וכן הכיל חריגים רבים (עכשיו פחות). מאחר ואלגוריתמים רבים בלמידת מכונה משתמשים במרחק אוקלידי זו בעיה מאחר והם מזניחים את כל עניין היחידות כלומר פיצורים עם טווח גדול יותר יחשב משמעותי יותר לפי המרחק האוקלידי. ומפה שבאמצעות Min_max_scaling הפכתי את כל הנתונים להיות בטווח של 0-1 והפרופורציות נשמרות.

בחירת פיצורים – קללת המימדים

לאחר פונקציית dummy variable יש בידנו 96 פיצורים מה שמביא אותנו לדבר על קללת המימדים. ריבוי הנתונים שמתרחש כאשר עברנו למימד (מספר פיצורים) גבוה יותר, משפיע על נפח המרחב המיוצג. כלומר עם עלייה במימדים הנתונים ממלאים פחות ופחות את מרחב. וזה מחזיר אותי לבעיה שצינתי ב data scaling, שאלגוריתמים משתמשים במרחק אוקלידי, ומפה שחישב של מרחב למספר עולה ועולה של פיצורים

הופל להיות לא יעיל גם אל מבחינת כוח חישוב וגם מבחינת תוצאות בפועל זה גורם ל over fitting. ניתן לזהות ממדיות גבוהה בהבדל ה AUC בניבוי על ה- train וה- test, אם ההבדל ביניהם גדול כלומר ה test נמוך הרבה יותר אז יש over fitting.

אני שילבתי בין 3 שיטות להקטנת הממדיות: PCA, Sequential Forward Selection, KBEST.

בעצם עבדתי על 2 DATA SETS במקביל לראות מה הוא השילוב הטוב ביותר.

1. בהתחלה ביצעתי את select_kbest, בבחירת ההיפר פרמטרים בחרתי להוריד את הממדיות מ 95 ל 20 פיצרים בידיעה שאני הולך לבצע אח"כ PCA. ניסיתי מספר score-function ביניהם f_regression and chi2, הטוב מביניהם היה f_regression וזאת על כך שהביא ליותר שונות מוסברת בביצוע ה PCA לאחריו וגם תוצאות טובות בהרבה במודלים שאח"כ. כפי שאמרתי לאחר בחירת ה 20 פיצרים ביצעתי PCA עם 4 קומפוננטות שכן זאת גם מסיבה שבדקתי מהו מספר הקומפוננטות הטוב ביותר כלומר בהתאם לתוצאות המודלים.

2. DATASET נוסף הוא באמצעות "Sequential Forward Selection", שכן בהיפר-פרמטרים בחרתי גם כן להוריד את הממדיות ל 20 פיצרים.

הטמעת מודלים

*בכל אחד מהמודלים אציג את ההיפר-פרמטרים הנבחרים עבור כל אחד מה Data-Sets.

SFFS.1

PCA+KBEST.2

לאחר מכן אציג את ה AUC במודל הספציפי עם הפרמטרים.

מודלים ראשוניים:

Naïve Bayes Classifier – זהו מודל ללא בחירת היפר-פרמטרים.

Model AUC score - sffs with: 0.8253430

Model AUC score - pca with: 0.8236270

K-Nearest-Neighbors – בבחירת ההיפר פרמטרים ביצעתי חיפוש חמדני מקיף על הפרמטרים הטובים ביותר עבור כל אחד מהדאטה-סט מהסעיף הקודם. שמתי דגש עבור $p=1$ שאומר שדובר במרחק מנהטן ולא אוקלידי. עבור ווקטורים בממדים גבוהים מנהטן לרוב עובד טוב יותר מהמרחק האוקלידי.

Model AUC score of feature selection model pca: 0.845513

leafsize = 9, n_neighbors:70, p = 1

Model AUC score of feature selection model sffs: 0.843634

leafsize 50, n_neighbors:50, p = 1

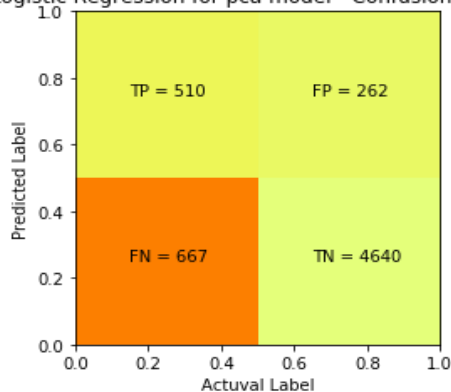
Logistic Regression - מאחר ומדובר בממדיות גבוהה יחסית גם אחרי ה dimensionality reduction, בחרתי לעשות גריד-סרץ' עם העדפה ל $l1$ penalty בזהה לסיבה בסעיף הקודם ו גם C נמוך שיהווה רגולריזציה חזקה בכדי למנוע overfitting אבל לא קטן מספיק בשביל למנוע למידה.

ניסיון נוסף היה לנסות thresholds שונים ל LR בגלל חוסר האיזון של הלייבלים אבל זה הערך הדיפולטיבי 0.5 היה הטוב ביותר.

Model AUC score - sffs with hyperparameter C 0.1 : 0.86765930

Model AUC score - pca with hyperparameter C 0.1 : 0.84563615

Logistic Regression for pca model - Confusion matrix



-Confusion matrix

במקרה הנ"ל מדובר על CM שמתייחס למודל רגרסיה לוגיסטית בהתייחס ל dataset שעבר מניפולציה של PCA. True positive (TP): אלה מקרים שניבאנו שכן ירד גשם ואכן בפועל ירד גשם.

True Negative (TN): ניבאנו שלא ירד גשם ובפועל לא ירד גשם.

False positive (FP): חזינו שירד גשם אבל לא ירד גשם.

False negative (FN): חזינו שלא ירד גשם אבל בפועל ירד גשם.

ישנם חישובים רבים וחשיבות רבה לCM, למשל חקלאי מסוים מכסה את הצמחיה שלו אם וירד גשם, ואם קרה FN כלומר חשב שלא ירד גשם אבל ירד זה פגע לו בצמחיה. כלומר אפשר לכנוון מודל שיעדיף לחזות שירד גשם ויטעה לפעמים כלומר יותר FP מאשר FN.

מודלים מתקדמים:

Multi-Layer Perceptron (ANN) – הדבר החשוב ביותר היה למנוע overfitting ולכן ביצעתי גריד-סרץ עם ערכי אלפא שונים, ולצד זה גם טווח ערכים לכמות שhidden layers.

hidden_layer_size = 20 alpha = 0.1 solver = "sgd", max_iter=200
ההיפר-פרמטרים יצאו זהים לשני הדאטה-סטס.

Model AUC pca test score: 0.8441970697029665

Model AUC sffs test score: 0.86520907492893

Support Vectors Machine – חיפוש מקיף של היפר – פרמטרים. החל מC רגולריזציה חלשה או חזקה, סוגים שונים של "קופסאות קסם"- קרנלים וגאמה שבעצם סותר את הרגולריזציה.

Model pca with kernel sigmoid and gamma=0.1 and C=1 AUC: 0.84500

Model sffs with kernel rbf and gamma 0.01 and C 100 AUC: 0.86840

Random Forest – כאן עשיתי לולאות בתוך לולאות ובחרתי את ההיפר-פרמטרים שנותנים את ה AUC הגבוה ביותר.

Model AUC score of feature selection model sffs: 0.8721382

n_estimators = 30, max_features = auto, max_depth = 20, min_samples_split = 10, min_samples_leaf = 4, bootstrap = False

הערכת המודלים

עבור כל dataset, הרצתי את כל ששת המודלים שהטמעתו מקודם, כל אחד עם ההיפר-פרמטרים המתאימים לו. וכן לכל מודל ישנו גרף שבתוכו מוצגת עקומת ROC עבור כל fold (10), אלו נמצאים במחברת.

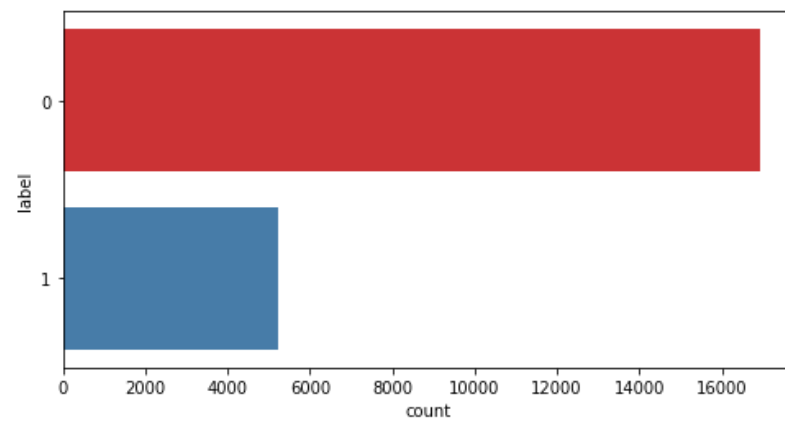
לאחר שלאורך כל הפרויקט רצתי עם שני DS במקביל בכדי לדעת מי מנבא הכי טוב, בחרתי בdata set של SFFS זאת כי הוא הראה יציבות AUC גבוהה יותר מהדאטה סט של PCA. כעת עלי לבחור מודל שאיתו אחזה את נתוני מבחן, לאור תוצאות המודלים גם בשלב ההטמעה וגם בשלב ההערכה בחרתי במודל Random Forest.

בכדי לוודא שאין overfitting ביצעתי למידה על דאטה מפוצל ובדקתי את הישגי המודל עליו וגם על הטסט. ההפרשים לא היו גדולים ומפה שאין overfitting מלבד העובדה שביצועי המודל על test זהים להטמעת המודלים. על מנת להגדיל את יכולת ההכללה בחרתי היפר-פרמטרים בשלב ההכללה שימנעו מצב כזה.

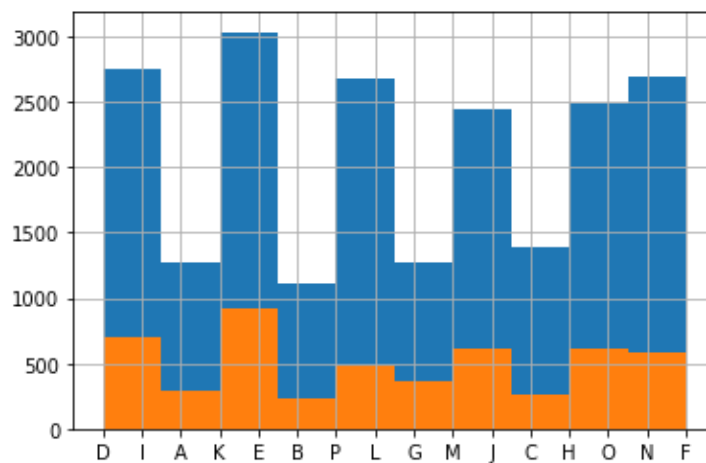
סיכום

אם כך לסיום אסקור את כל מה שעברנו עליו, בהתחלה קיבלנו דאטה עם המון רעשים, בצענו מניפולציות על הפיצורים בכדי להתאים אותם לקריאת מחשב, הסקנו מסקנות על הדאטה ועל חשיבויות כאלה ואחרות של פיצורים. יצרתי שני Data Sets באחד דיללתי את המימדיות באמצעות PCA+ KBEST והשני באמצעות SFFS. לאחר מכן בחנתי שישה מודלים שונים על כל אחד מה DS ובכל אחד מהם חיפשתי את ההיפר-פרמטרים האידיאליים לו בהתאמה ל DS וה AUC הגבוה ביותר. לאחר מכן בצעתי 10-fold-cross-validation, ולמדתי את המודלים על כלל הדאטה בפולדים השונים. ראיתי לנכון שהמודל הטוב ביותר לביצוע הפרדיקציה הוא Random Forest ובאמצעותו חזיתי את סט test.

נספח א'



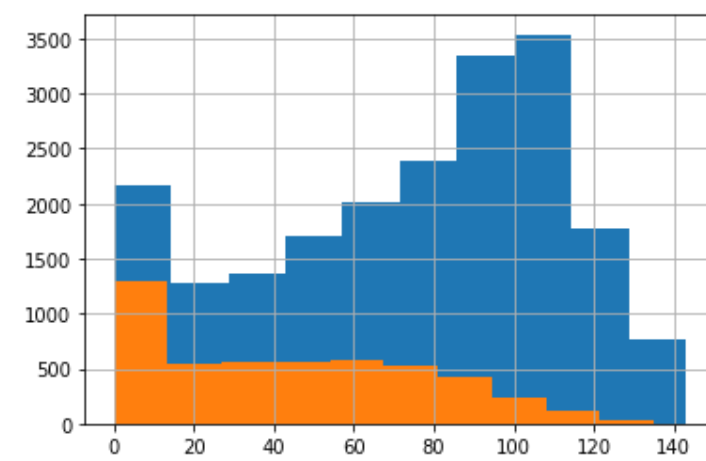
נספח ב'



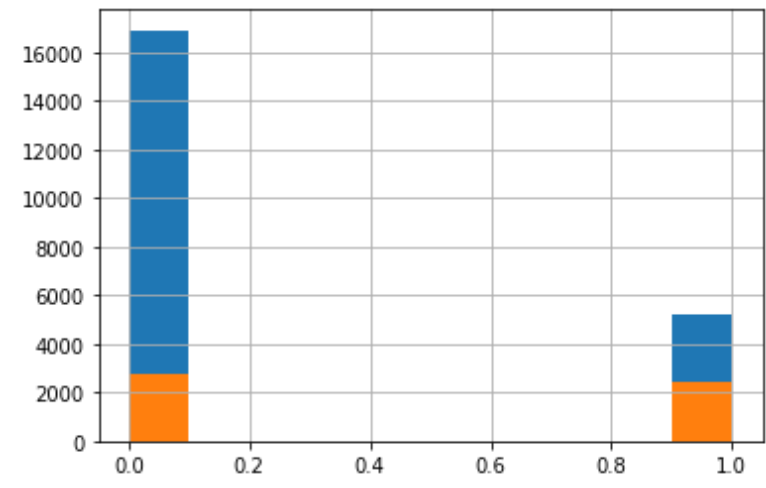
נספח ג'

```
df['check'] = df.Feature_14.apply(lambda x: "1" if x > 1 else ("0" if x <= 1 else np.nan )) df['check'].equals(df['Feature_13'])#
```

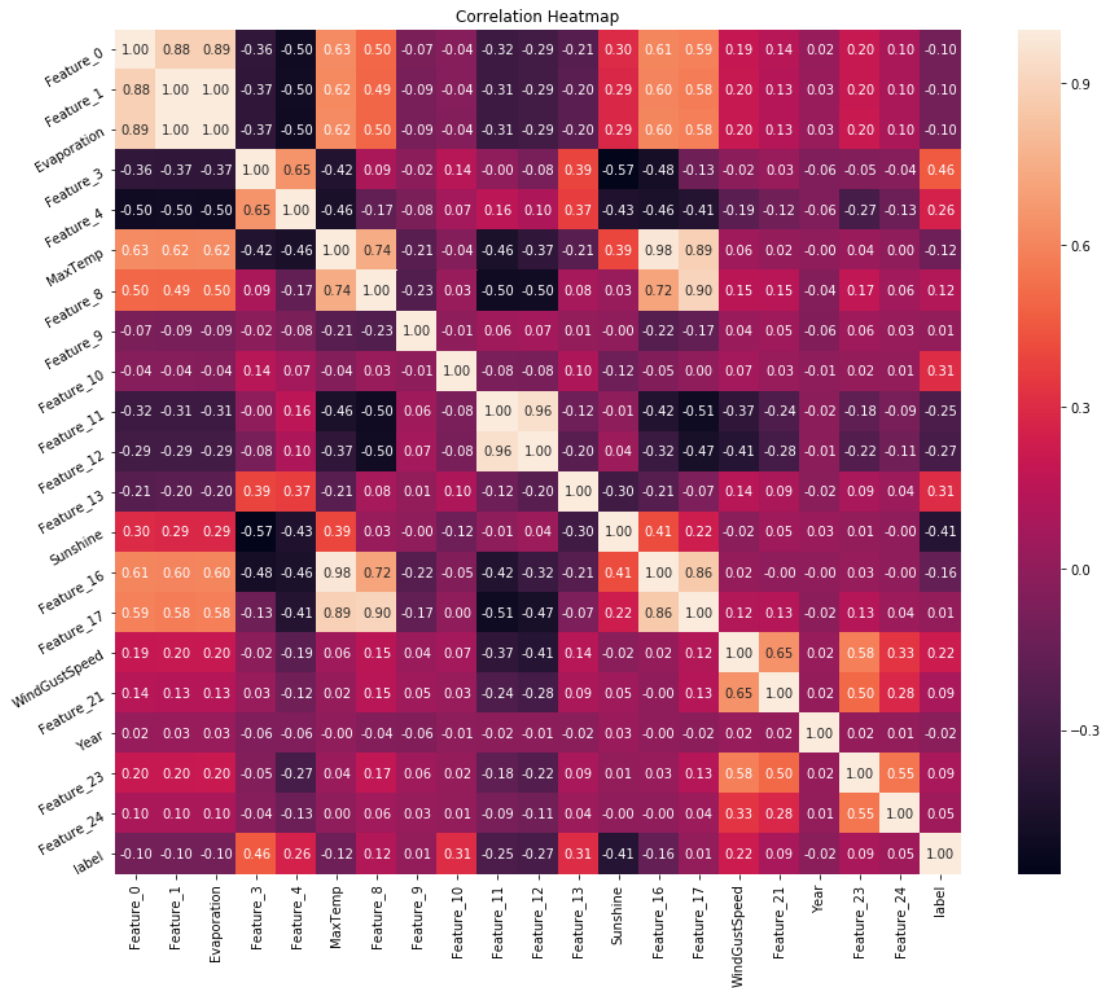
נספח ד'

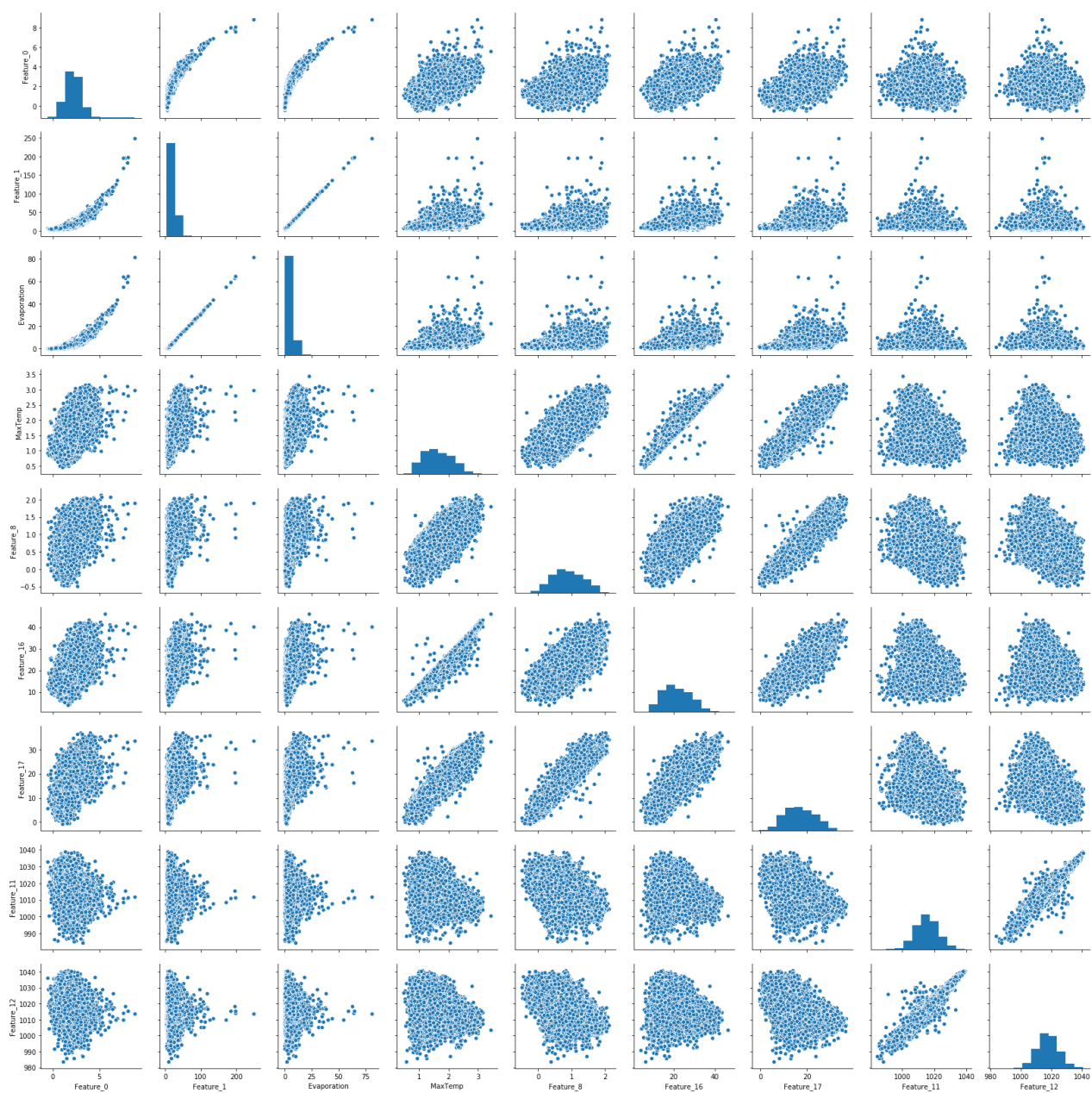


נספח ה'

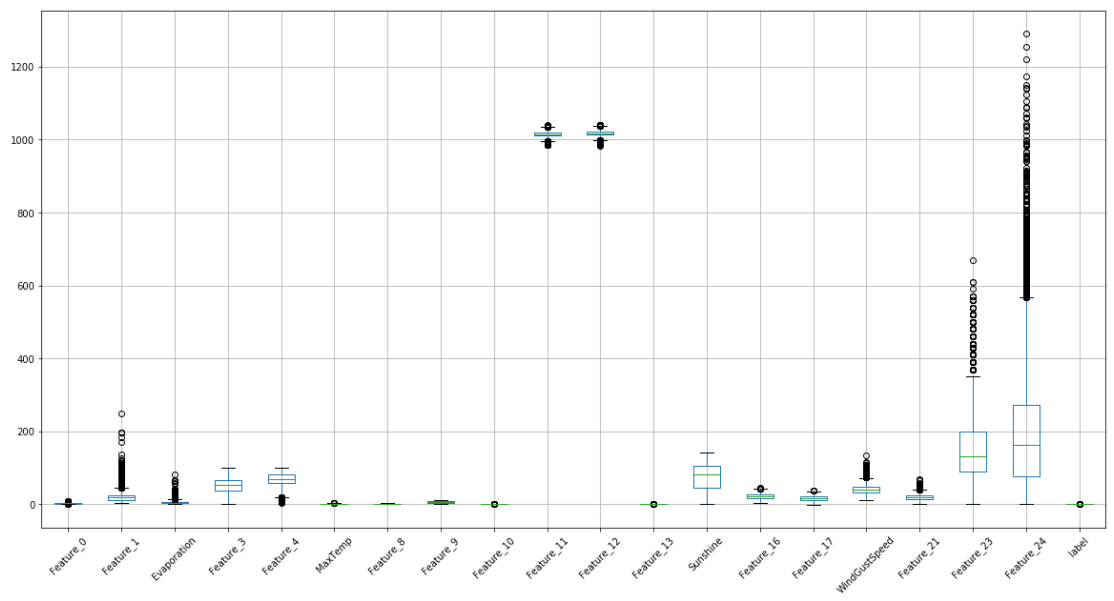


נספח ו'





נספח ח'



נספח ט'

