למידה חישובית - תרגיל 2 תחרות **Kaggle**

מגישים: סהר בריבי, 311232730
אור אוקסנברג, 312460132

שם הקבוצה: Rage against the machine learning

תיאור קצר של השיטה:

עיבוד מקדים:

החלק הראשון בתהליך הוא השלב של העיבוד המקדים וניקוי הדאטה. שלב זה כולל מספר תהליכים:

1. הפרדה בין עמודות קטגוריאליות, נומריות ובינאריות.
2. השלמת הערכים החסרים באמצעות שתי שיטות - העמודות הקטגוריאליות והבינאריות מקבלות התייחסות של most frequent והעמודות הנומריות מקבלות התייחסות של mean value.
3. ביצוע one hot encoder על העמודות הקטגוריאליות.
4. נרמול הדאטה

לאחר התהליך הראשוני, ביצענו feature selection על מנת להשתמש בפיצ'רים הרלוונטים ביותר. על מנת לבחור בפיצ'רים אלו ביצענו שילוב בין מספר שיטות שונות:

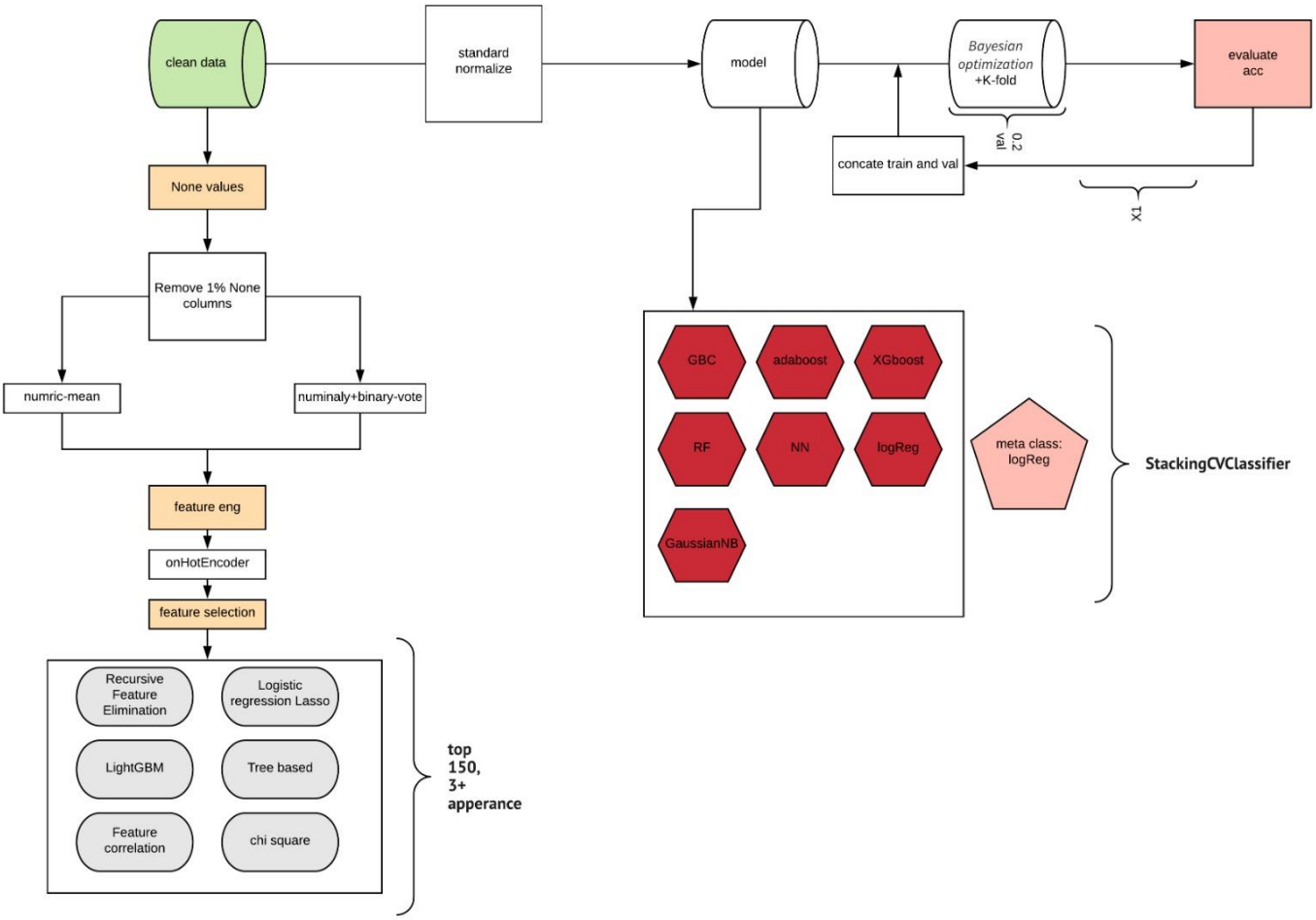
- אנחנו בודקים קוראלציות בין הפיצ'רים ללייבלים
- דירוג הפיצ'רים, 150 הפיצ'רים בעלי הדירוג הגבוה ביותר, לפי מדדים שונים ביניהם:
  - chi square
  - Logistic regression Lasso
  - Tree based
  - LightGBM
  - Recursive Feature Elimination - שימוש בlogistic regression, כשהמטרה היא להוריד כל פעם קצת פיצ'רים ולאמן מודל עד שמגיעים לתנאי עצירה שזה כמות הפיצ'רים שהגדרנו.
- לאחר דירוג הפיצ'רים אנחנו בוחרים את הפי'צרים שהיו בעלי הדירוג הגבוה ביותר בלפחות שלושה פרמטרים מהמפורטים לעיל.

במודל אנחנו משתמשים בStacking שעושה Ensemble למודלים הבאים:

- Logistic Regression
- XGBoost
- Adaboost
- GradientBoosting Classifier
- Random Forest
- NN

הMetaClassifier שלנו הוא Logistic Regression.

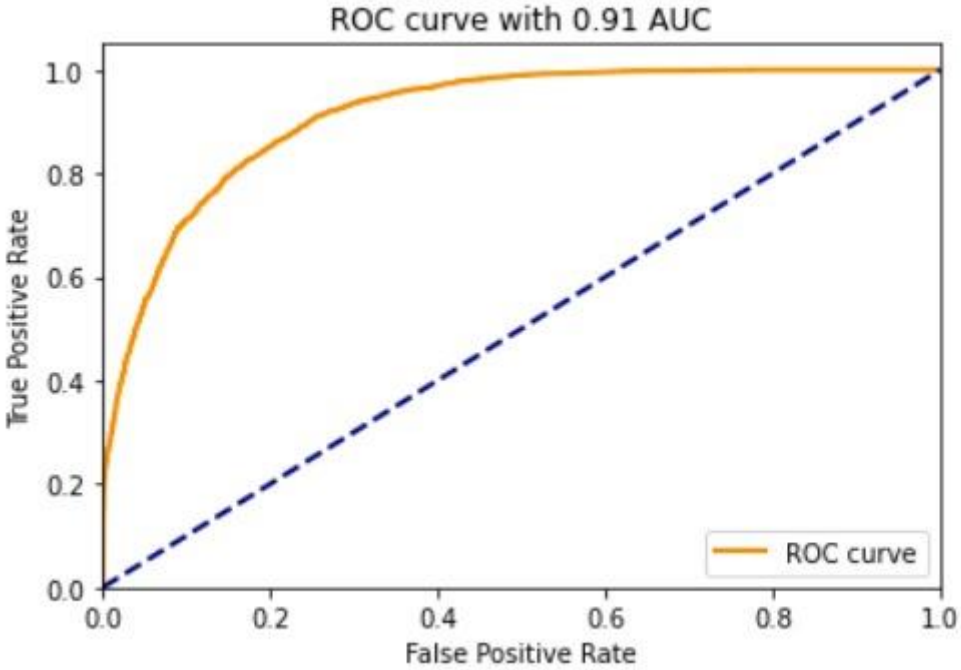
אנחנו מבצעים בחירה של היפר-פרמטרים בעזרת BayesianSearch, כאשר אנחנו מריצים 10 איטרציות ו10 CV. בתמונה מוצג הpipeline של השלבים שביצענו :



תוכן הNotebook שהניב את התוצאה הטובה ביותר מצורף, אך מאחר ואנחנו משתמשים בBayesianSearch, נציין גם את ההיפר פרמטרים שהביאו לתוצאה הטובה ביותר :

- Logistic Regression : C = 0.1
- XGBoost: booster = 'gbtree', eta = 0.1, gamma = 0.072600932, max_depth = 6, n_estimators = 100
- Adaboost : learning_rate = 0.4, n_estimators = 40
- GradientBoosting Classifier : max_depth = 7, min_samples_split = 8, n_estimators = 100
- Random Forest: min_samples_split = 9, n_estimators = 100
- NN : learning_rate = 0.001, layers1 = 3, layers2 = 3, layers3 =2, nodes1 = 700, nodes2 = 400, nodes3 = 300
- Meta classifier (Logistic Regression): C = 0.1

גרף הAUC של המודל הנבחר:



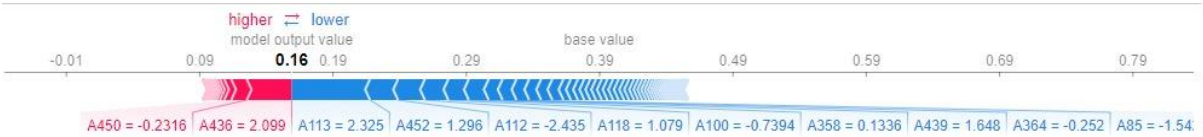
ניתוחי המודל בעזרת **SHAP**:

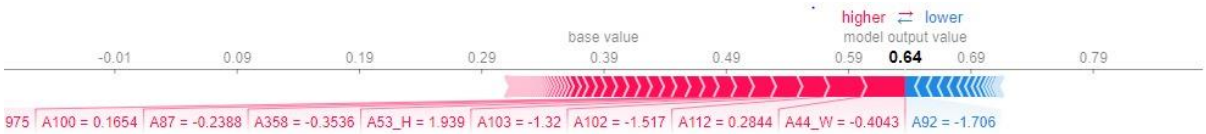
מאחר והרצנו Stacking, אין ביכולתנו להריץ SHAP על כל המודל. לכן, בחרנו להריץ SHAP על המסווג הכי מדויק מבין המודל, והרצנו זאת על רשת הנוירונים בStacking.

התנהגות המודל:

הגרפים הבאים מנותחים באופן הבא: הגורמים בכחול הם הגורמים שהורידו את הסיכוי להצלחת הטיפול בעוד הגורמים באדום הם עלו שהעלו את ההסתברות. הערך המצוין הוא הערך שהמודל נתן עבור רשומה זו, וזו ההסתברות להצלחת הטיפול לפי המודל.

להמחשה, בדוגמא הראשונה ניתן לראות שהערך שהמודל נתן הוא 0.35 כלומר, זוהי ההסתברות להצלחת הטיפול.

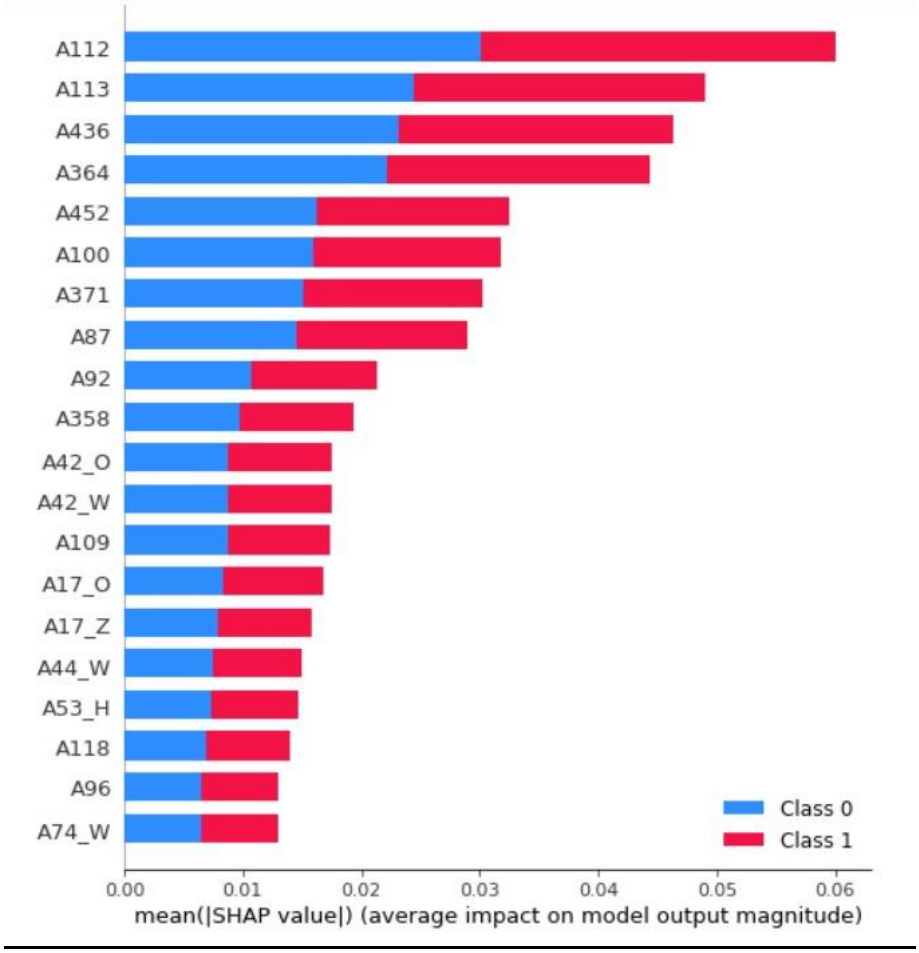




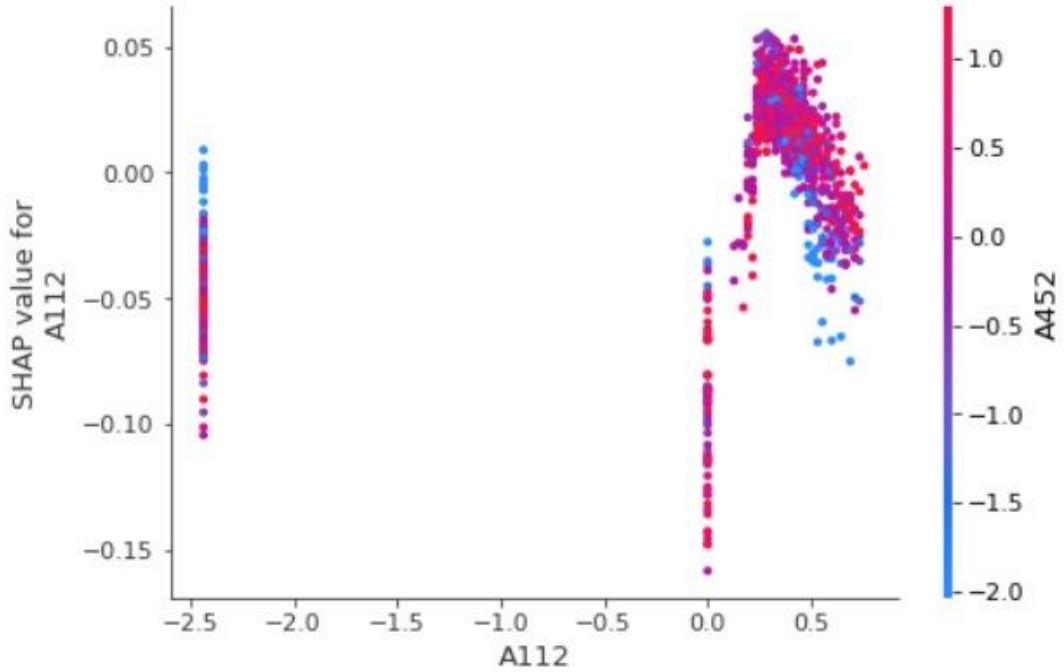
<u>חשיבות המשתנים לפי SHAP :</u>

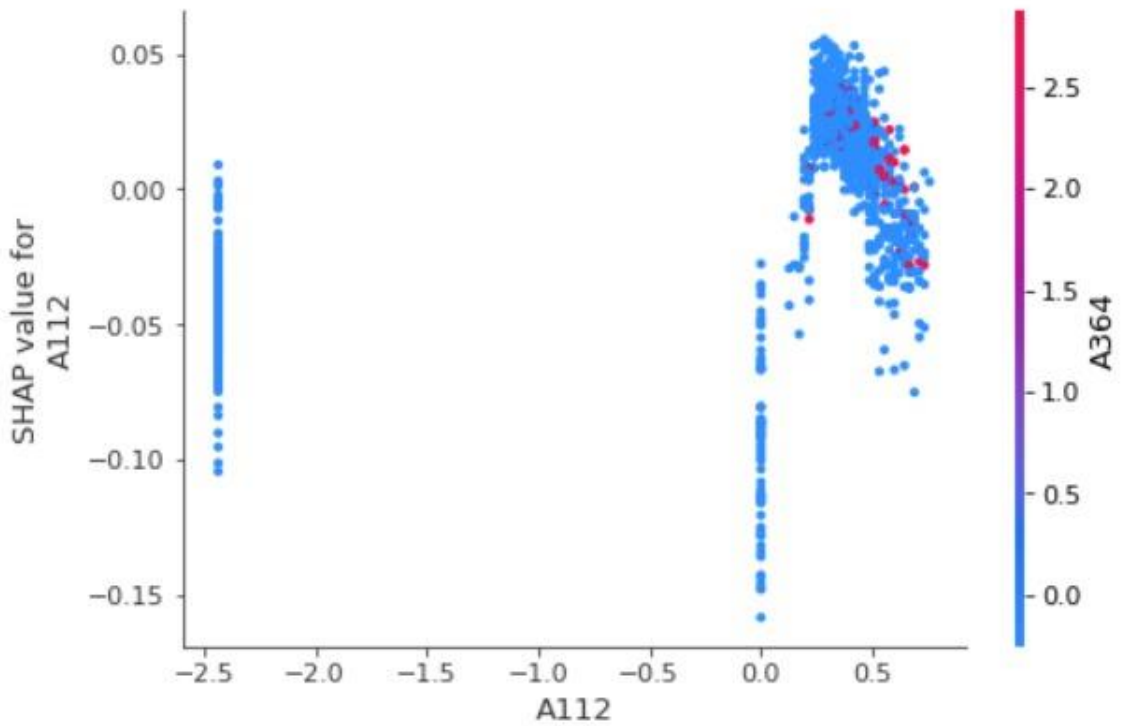
בחלק זה נריץ SHAP על Random Forest מנת להציג הן את הSummary plot והן את ה Dependence Contribution Plots. הסיבה שאנחנו מריצים על Random Forest ולא על רשת הנוירונים כמו בסעיף הקודם, היא שההרצה על רשת הנוירונים דורשת שימוש בKernalExplainer של SHAP ולוקחת יותר מדי זמן (56 שעות) ולא היה ברשותנו את המשאבים להריץ זאת.

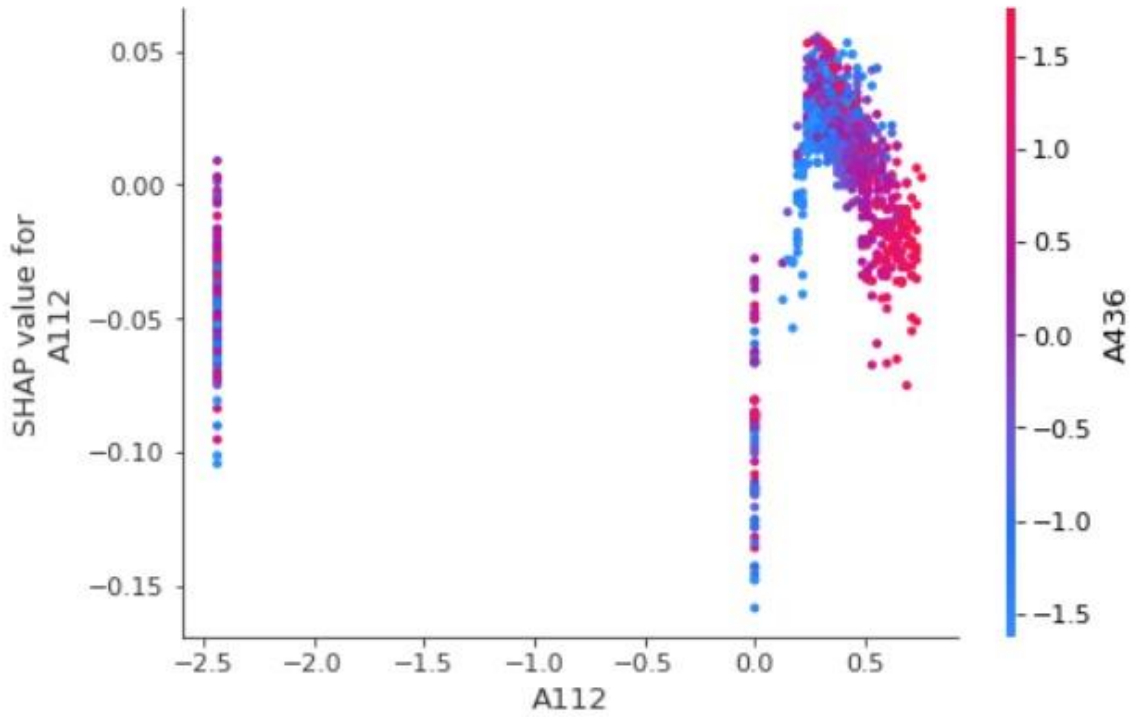
<u>Plot Summary</u>

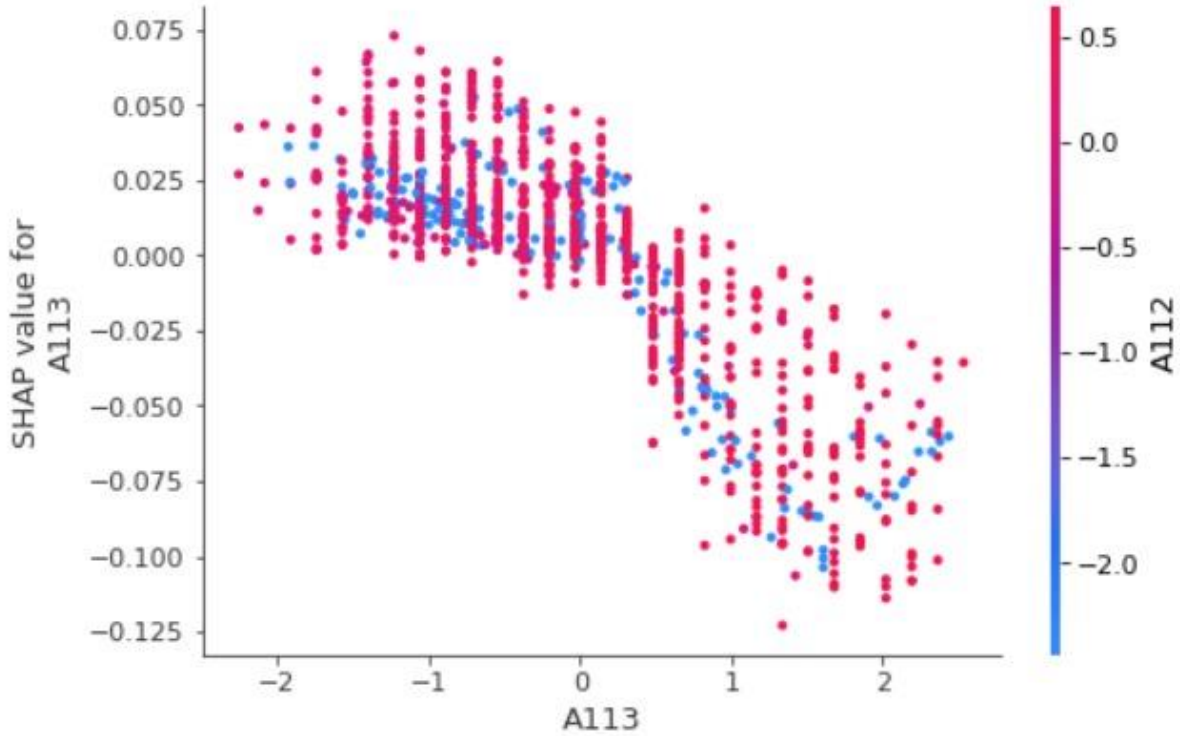


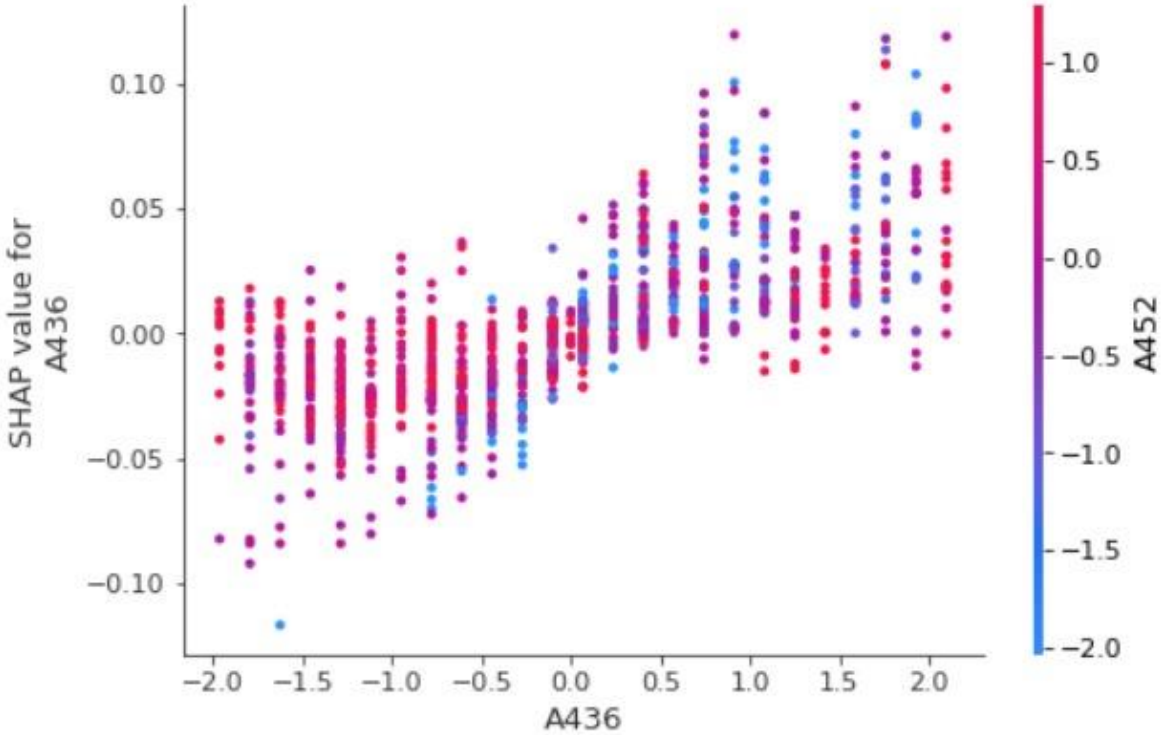
SHAP Dependence Contribution Plots

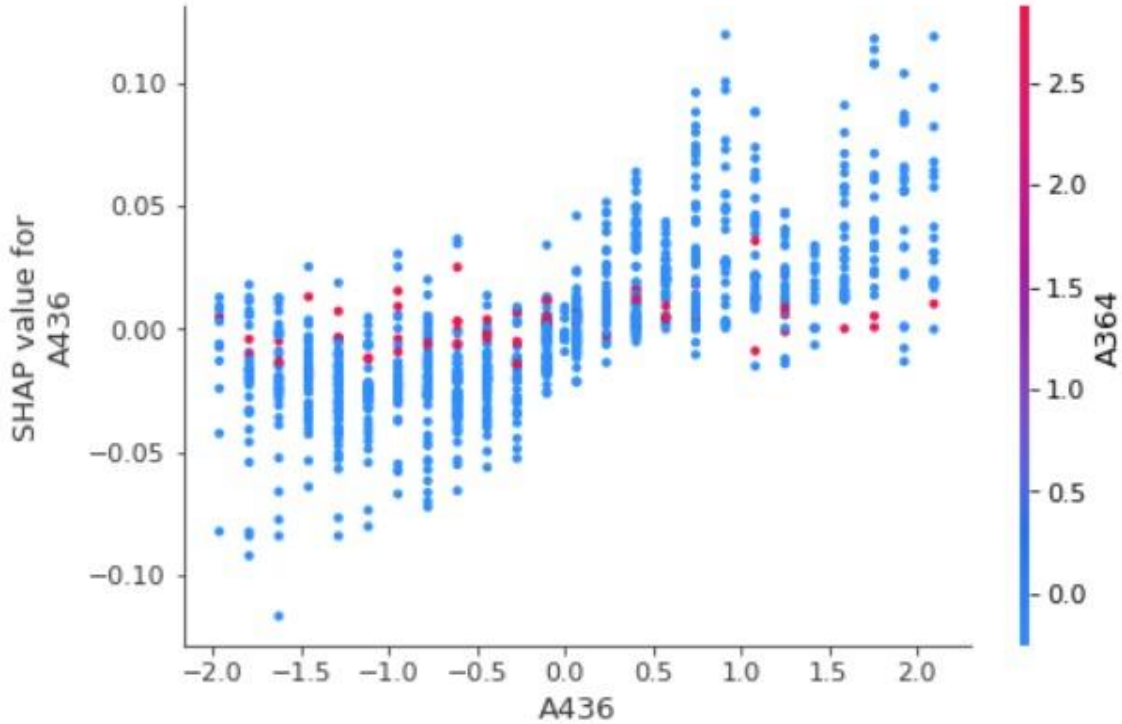


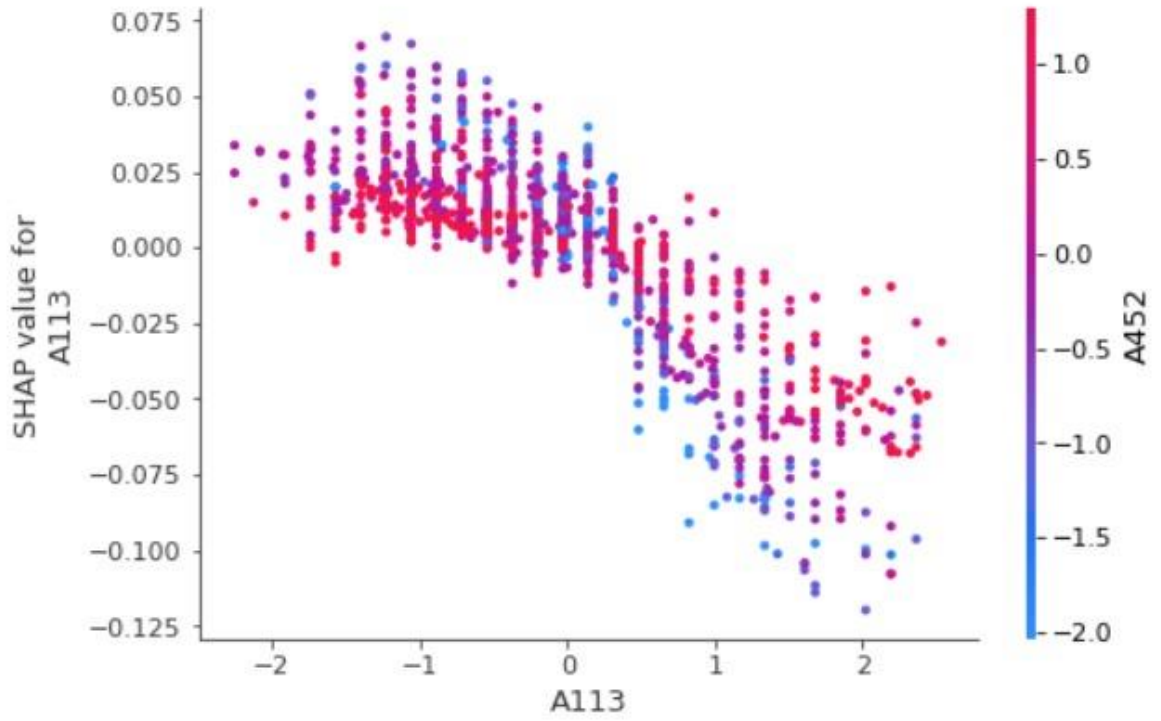


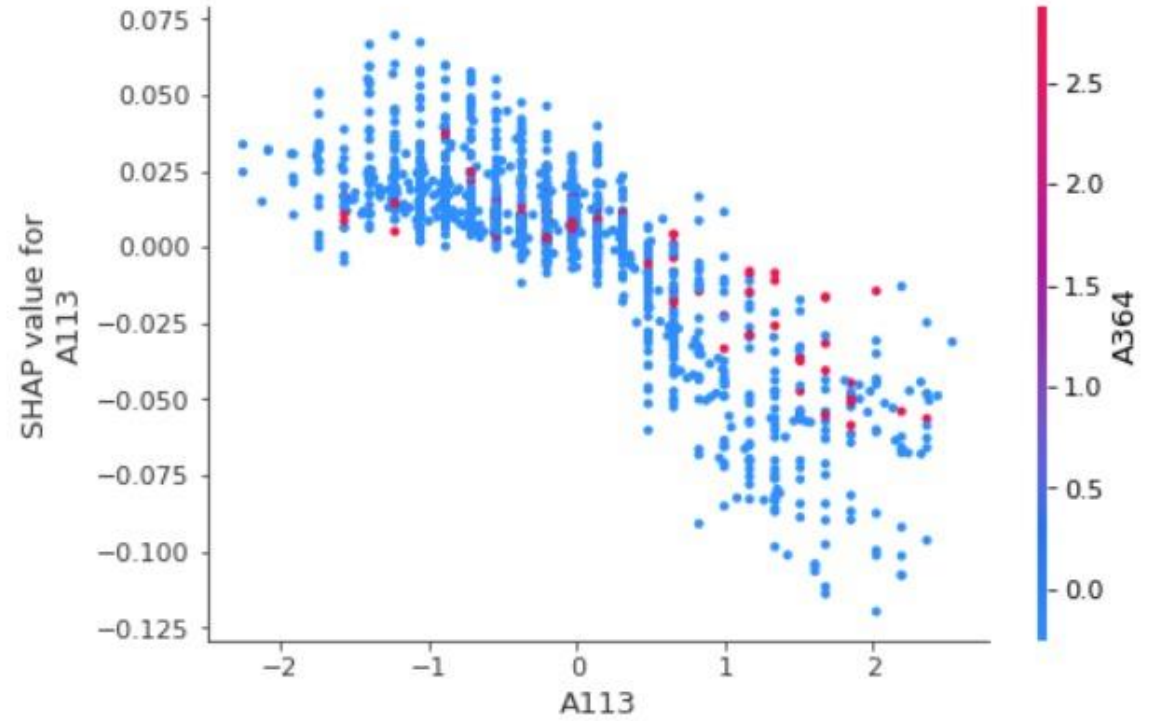

SHAP value for A112
A112
A436
0.05
0.00
−0.05
−0.10
−0.15
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5

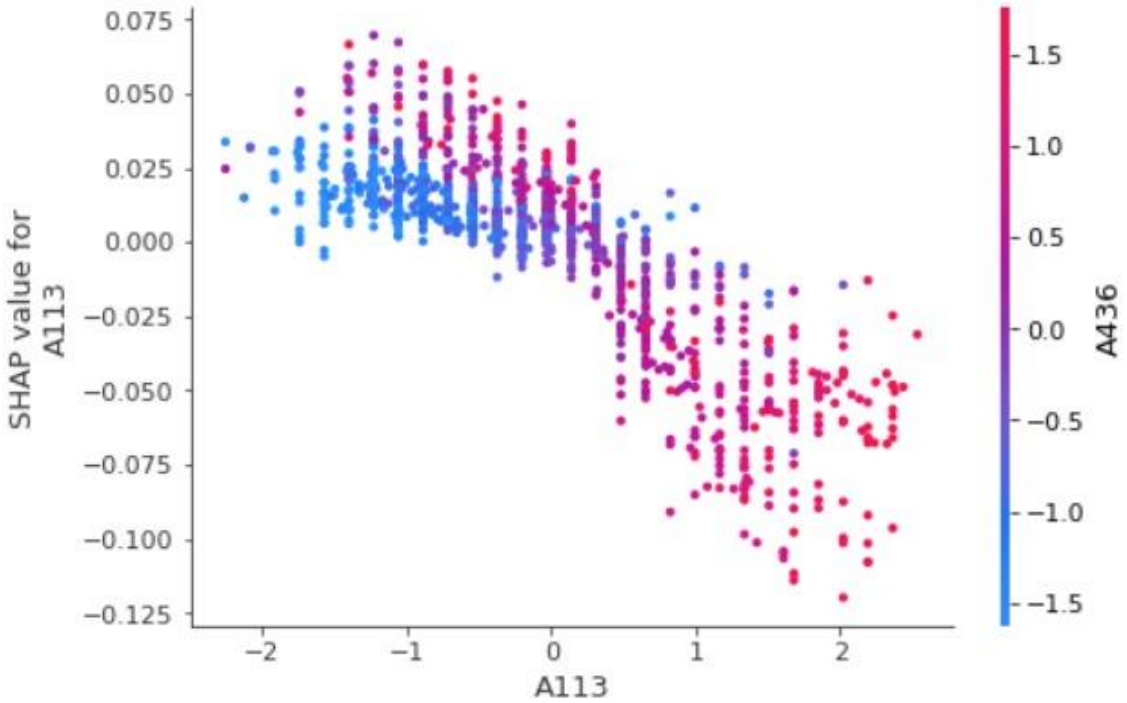

SHAP value for A113
A113
A112
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
−2
−1
0
1
2
0.5
0.0
−0.5
−1.0
−1.5
−2.0

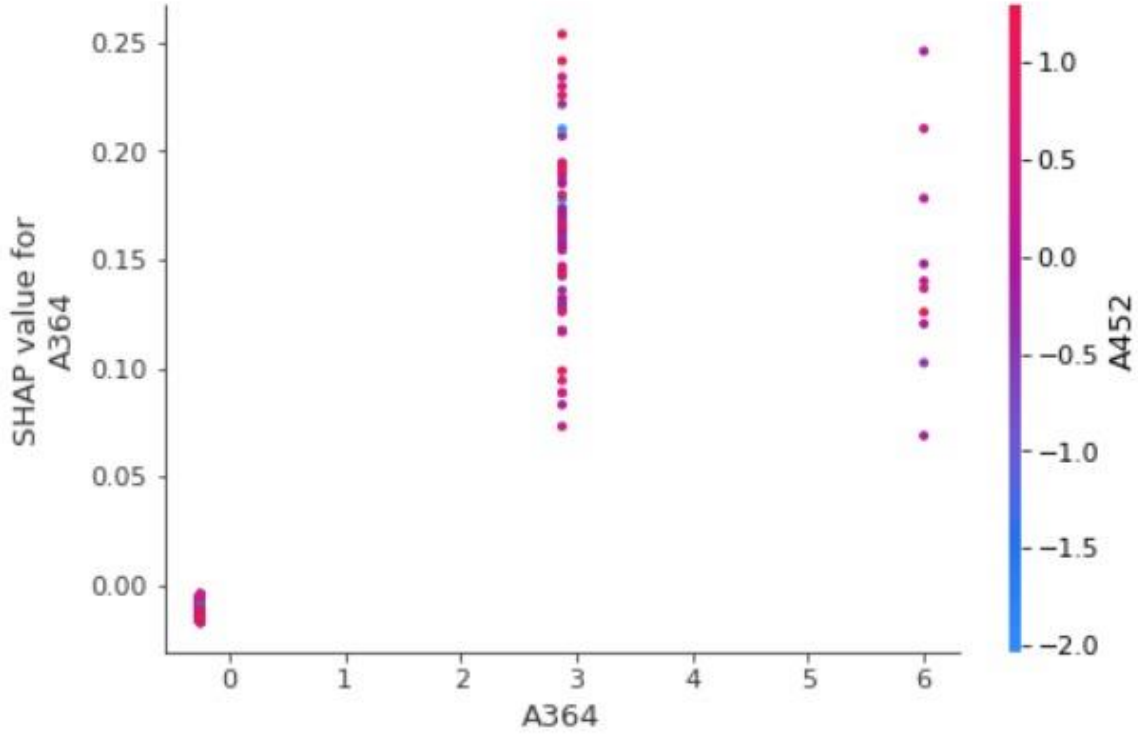

SHAP value for A436
A436
A452
0.10
0.05
0.00
−0.05
−0.10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


SHAP value for A436
A436
A364
0.10
0.05
0.00
−0.05
−0.10
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5


SHAP value for A113
A113
A452
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
−2
−1
0
1
2
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0


SHAP value for A113
A113
A364
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
−2
−1
0
1
2
2.5
2.0
1.5
1.0
0.5
0.0


SHAP value for A113
A113
A436
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
−2
−1
0
1
2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5


SHAP value for A364
A364
A452
0.25
0.20
0.15
0.10
0.05
0.00
0
1
2
3
4
5
6
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0