	Logistic Regression	Random Forest,	XGBoost	Neural Networks
Filter method	- accuracy = 0.569620253164557 - precision = 0.5185185185185185 - recall = 0.7777777777777778 - f1 = 0.6222222222222222	- accuracy = 0.5443037974683544 - precision = 0.5 - recall = 0.5277777777777778 - f1 = 0.5135135135135136	- accuracy = 0.5569620253164557 - precision = 0.5128205128205128 - recall = 0.5555555555555556 - f1 = 0.5333333333333333	0.5569620132446289
Forward feature selection	- accuracy = 0.569620253164557 - precision = 0.5185185185185185 - recall = 0.7777777777777778 - f1 = 0.6222222222222222	- accuracy = 0.5443037974683544 - precision = 0.5 - recall = 0.5277777777777778 - f1 = 0.5135135135135136	- accuracy = 0.5569620253164557 - precision = 0.5128205128205128 - recall = 0.5555555555555556 - f1 = 0.5333333333333333	0.5569620132446289
Backward feature selection	- accuracy = 0.5822784810126582 - precision = 0.5306122448979592 - recall = 0.7222222222222222 - f1 = 0.611764705882353	- accuracy = 0.45569620253164556 - precision = 0.425531914893617 - recall = 0.5555555555555556 - f1 = 0.48192771084337355	- accuracy = 0.5443037974683544 - precision = 0.5 - recall = 0.6388888888888888 - f1 = 0.5609756097560975	0.5696202516555786

Recursive feature selection	- accuracy = 0.569620253164557 - precision = 0.52 - recall = 0.7222222222222222 - f1 = 0.6046511627906976	- accuracy = 0.5569620253164557 - precision = 0.5121951219512195 - recall = 0.5833333333333334 - f1 = 0.5454545454545454	- accuracy = 0.5949367088607594 - precision = 0.5476190476190477 - recall = 0.6388888888888888 - f1 = 0.5897435897435898	0.6202531456947327
Tree-base Method	- accuracy = 0.5569620253164557 - precision = 0.5102040816326531 - recall = 0.6944444444444444 - f1 = 0.588235294117647	- accuracy = 0.5822784810126582 - precision = 0.5365853658536586 - recall = 0.6111111111111112 - f1 = 0.5714285714285715	- accuracy = 0.5443037974683544 - precision = 0.5 - recall = 0.6388888888888888 - f1 = 0.5609756097560975	0.58227849006652
PAC	- accuracy = 0.4810126582278481 - precision = 0.4528301886792453 - recall = 0.6666666666666666 - f1 = 0.5393258426966292	- accuracy = 0.5189873417721519 - precision = 0.48 - recall = 0.6666666666666666 - f1 = 0.5581395348837209	- accuracy = 0.5189873417721519 - precision = 0.48 - recall = 0.6666666666666666 - f1 = 0.5581395348837209	0.5316455960273743

Principal Component Analysis (PCA) คือ เทคนิคเชิงเลขอันหนึ่งที่เรานิยมนำมาใช้ในการทำ decomposition ข้อมูล ในกระบวนการเตรียมและคลีนข้อมูล เพื่อลด features redundancy และช่วยบีบอัด feature space ให้เล็กลง ตัดตัวแปรที่มีความสำคัญน้อยออกไป โดยไม่ทำให้สูญเสีย information หรือข้อมูลสำคัญไป หลักการเบื้องต้นของ PCA เริ่มจากการทำ Normalize ข้อมูล ด้วย Standard deviation ได้ จากนั้นนำค่าที่ได้มาคำนวณหา Covariance Matrix จากนั้นนำไปคำนวณ Eigen vector และ Eigen value เพื่อใช้เป็นค่าในสกัดคุณสมบัติเด่นออกมา แล้วทำการจัดเรียงลำดับค่าตาม Eigen vector เราก็จะได้ Feature ที่สำคัญตัว Top 3 หรือ Top 5 เพื่อนำมาใช้ในการกำหนดเป็น Feature หลัก หรือเป็นตัวแปรสำคัญในการวิเคราะห์ต่อไป โดย PCA จะให้ความสำคัญกับ Correlation ซึ่งเป็นอีกกระบวนการที่ช่วยเราในงาน Feature Selection และ Feature extraction ในการทำ Machine Learning ต่อไป

Monfreda, M. (2012). Principal Component Analysis: A Powerful Interpretative Tool at the Service of Analytical Methodology. Principal Component Analysis. doi:10.5772/36929