This dataset contains information about drug classification based on patient general information and its diagnosis. Machine learning model is needed in order to predict the outcome of the drugs type that might be suitable for the patient.

Objectives of Notebook

This notebook aims to:

* Dataset exploration using various types of data visualization.
* Build various ML models that can predict drug type.

The machine learning models used in this project are:

* Logistic Regression
* Support Vector Machine (SVM)
* K Neighbours
* Decision Tree
* Random Forest
* AdaBoostClassifier
* GradientBoostingClassifier
* XGBClassifier

| **Variable Name** | **Description** | **Sample Data** |
| --- | --- | --- |
| **Age** | Patient Age | 23; 47; ... |
| **Sex** | Gender of patient (male or female) | F; M; ... |
| **BP** | Levels of blood pressure (high, normal, or low) | HIGH; NORMAL; LOW; ... |
| **Cholesterol** | Levels of cholesterol (high or normal) | 1.4; 1.3; ... |
| **Na\_to\_K** | Sodium to potassium ratio in blood | 25.355; 13.093; ... |
| **Drug** | Type of drug | DrugY; drugC |

**Data sample:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Age | Sex | BP | Cholesterol | Na\_to\_K | Drug |
| 0 | 23 | F | HIGH | HIGH | 25.355 | DrugY |
| 1 | 47 | M | LOW | HIGH | 13.093 | drugC |
| 2 | 47 | M | LOW | HIGH | 10.114 | drugC |
| 3 | 28 | F | NORMAL | HIGH | 7.798 | drugX |
| 4 | 61 | F | LOW | HIGH | 18.043 | DrugY |

**Exploratory Data Analysis:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Column** | **Non-Null Count** | **Dtype** |
| 1 | Age | 200 non-null | int64 |
| 2 | Sex | 200 non-null | object |
| 3 | BP | 200 non-null | object |
| 4 | Cholesterol | 200 non-null | object |
| 5 | Na\_to\_K | 200 non-null | float64 |
| 6 | Drug | 200 non-null | object |

**Analysis:**

* We have 200 records
* We have 6 features
* We don’t have null values
* Out of 6 features 4 categorical and 2 continuous in nature.

**Descriptive Statistics:**

|  |  |  |
| --- | --- | --- |
|  | Age | Na\_to\_K |
| count | 200 | 200 |
| mean | 44.315 | 16.08449 |
| std | 16.54432 | 7.223956 |
| min | 15 | 6.269 |
| 25% | 31 | 10.4455 |
| 50% | 45 | 13.9365 |
| 75% | 58 | 19.38 |
| max | 74 | 38.247 |

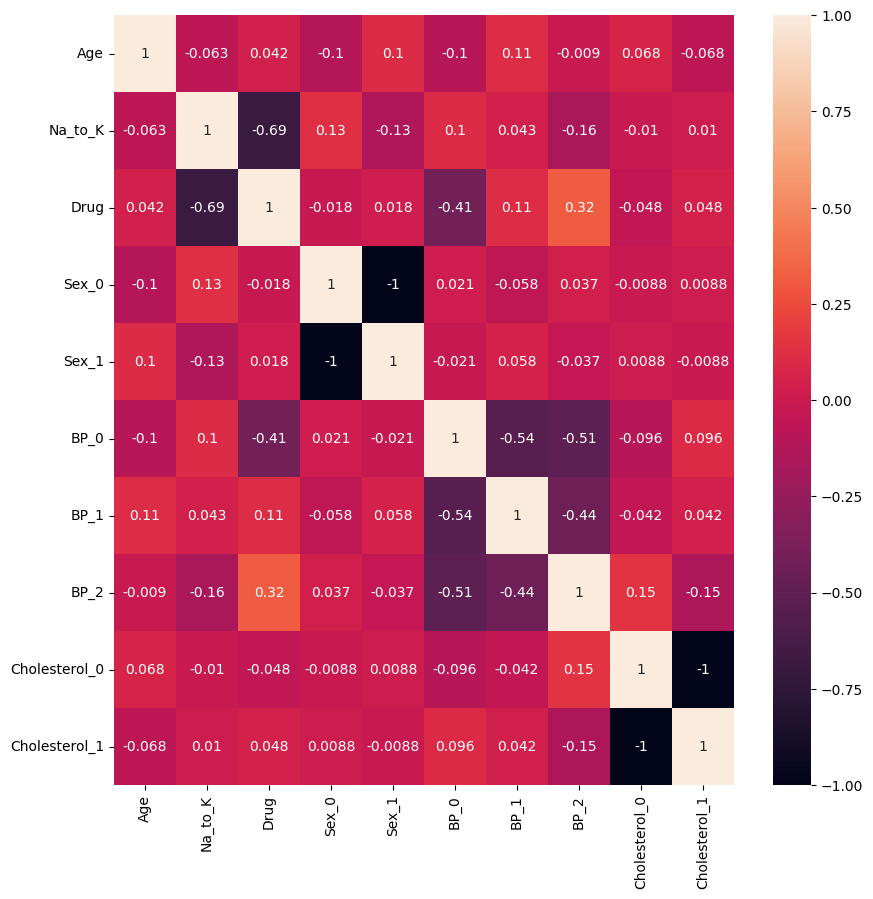
* As mean and median are almost same outliers may not exist.

**Data after encoding:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Age | Na\_to\_K | Drug | Sex\_0 | Sex\_1 | BP\_0 | BP\_1 | BP\_2 | Cholesterol\_0 | Cholesterol\_1 |
| 0 | 23 | 25.355 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 47 | 13.093 | 3 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 2 | 47 | 10.114 | 3 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 3 | 28 | 7.798 | 4 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 4 | 61 | 18.043 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |

**Multi-collinearity-Check**

* **Stage 1:Correlation Heatmap**



* Na\_to\_K, BP\_0 has higher -ve correlation with target feature.

**Multicollinearity test**

* Stage 2: Variance Inflating Factor (VIF)

formula for VIF = 1/(1-R2)

1. Regress every independent variable with each other and find the R2 score

2. find out VIF using above formula

3. if VIF is more than 5 for any independent variable we can conclude that multi-collinearity exists.

|  |  |  |
| --- | --- | --- |
|  | VIF | Independent variables |
| 2 | inf | Sex\_0 |
| 3 | inf | Sex\_1 |
| 4 | inf | BP\_0 |
| 5 | inf | BP\_1 |
| 6 | inf | BP\_2 |
| 7 | inf | Cholesterol\_0 |
| 8 | inf | Cholesterol\_1 |
| 1 | 1.046964 | Na\_to\_K |
| 0 | 1.032008 | Age |

* Except for Na\_to\_K and Age all other features have high VIF score.

**Correlation with target feature:**

|  |  |  |
| --- | --- | --- |
|  | independent variables | correlation |
| 6 | BP\_2 | 0.319533 |
| 5 | BP\_1 | 0.112195 |
| 8 | Cholesterol\_1 | 0.048415 |
| 0 | Age | 0.041856 |
| 3 | Sex\_1 | 0.018239 |
| 2 | Sex\_0 | -0.01824 |
| 7 | Cholesterol\_0 | -0.04842 |
| 4 | BP\_0 | -0.40703 |
| 1 | Na\_to\_K | -0.68905 |

* BP\_2, BP\_1, BP\_0, Na\_to\_K have higher correlation with target feature

**Feature engineering**

Applying Principal component analysis and adding target feature

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | PC\_1 | PC\_2 | PC\_3 | PC\_4 | PC\_5 | Drug |
| 1 | 0.414693 | 2.088055 | -1.61567 | 1.281071 | 1.037403 | 0 |
| 2 | -1.84372 | -0.97972 | 0.87641 | 1.226368 | 0.195012 | 3 |
| 3 | -1.90259 | -1.02359 | 0.899764 | 1.081141 | 0.004731 | 3 |
| 4 | -1.05803 | 2.248831 | 0.218861 | -1.53314 | 0.348587 | 4 |
| 5 | -0.66565 | 1.419629 | 1.592935 | 1.673441 | -0.71809 | 0 |

Analysis:

* Total number of features were reduced to 11 from 5
* Above features are considered for model building and evaluation.
* Explained variance ratio after PCA is:

[0.23734945, 0.4659659, 0.63792352, 0.79625888, 0.90321265]

**Model building:**

**1. Train-test split –** Splitting the data into train and test sets

**2. Cross-validation -** Evaluating machine learning models by training several ML models on subsets of the available input data and evaluating them on the complementary subset of the data.

**3. Hyperparameter tuning-** Hyperparameter tuning consists of finding a set of optimal hyperparameter values for a learning algorithm while applying this optimized algorithm to any data set. That combination of hyperparameters maximizes the model's performance, minimizing a predefined loss function to produce better results with fewer errors.

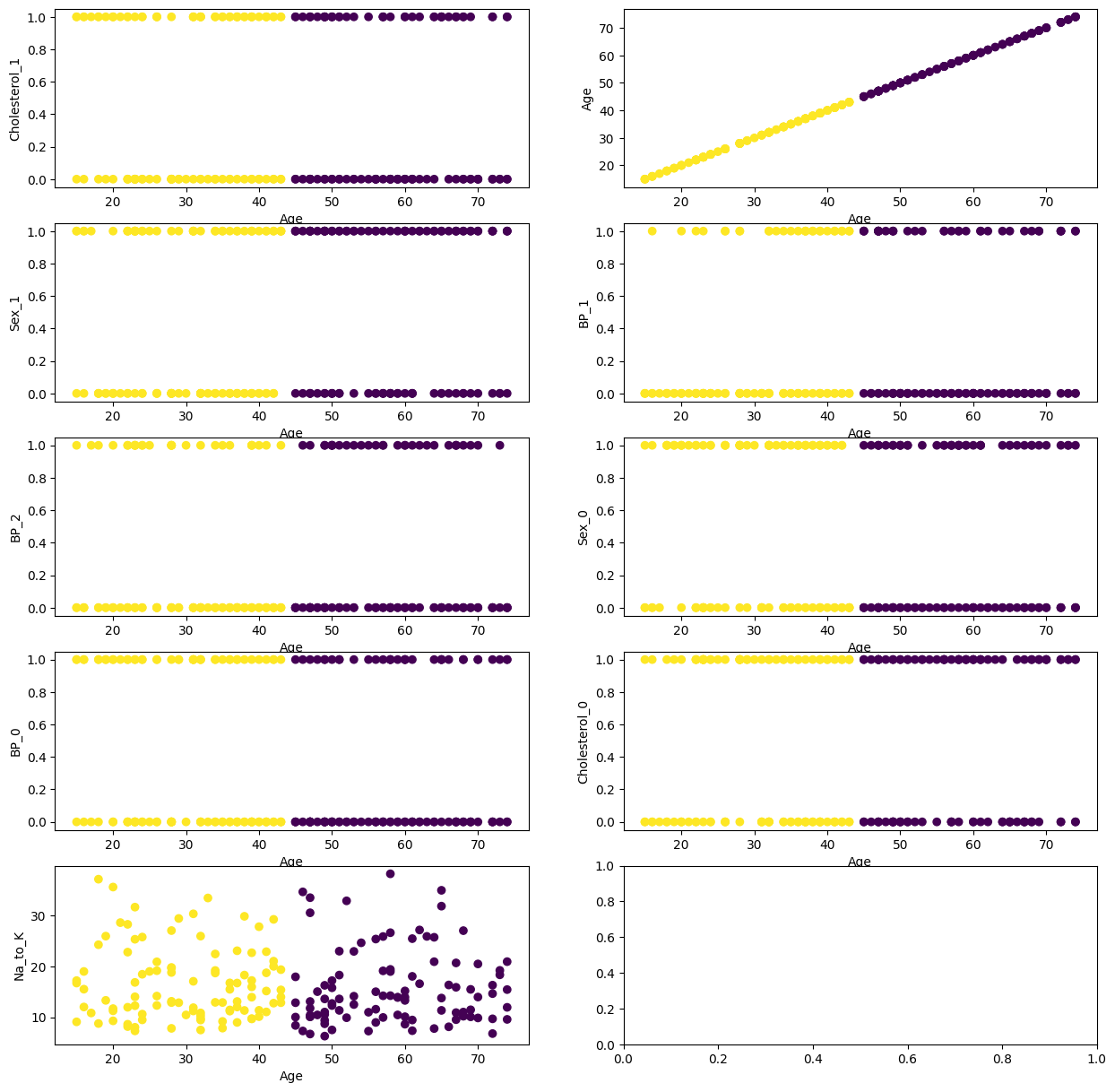
|  |
| --- |
| model: RF |
| Best\_params: {'max\_depth': 9, 'max\_features': 7, 'n\_estimators': 150} |
| model: XB |
| Best\_params: {'alpha': 0, 'eta': 1, 'gamma': 0, 'max\_depth': 3, 'reg\_lambda': 5} |
| model: GB |
| Best\_params: {'learning\_rate': 0.1, 'n\_estimators': 100} |
| model: KNN |
| Best\_params: {'algorithm': 'auto', 'weights': 'distance'} |
| model: SVR |
| Best\_params: {'C': 1.5, 'gamma': 'auto'} |
| model: DTR |
| Best\_params: {'criterion': 'gini', 'max\_depth': 11, 'max\_features': 4} |
| model: AB |
| Best\_params: {'learning\_rate': 0.5, 'n\_estimators': 200} |

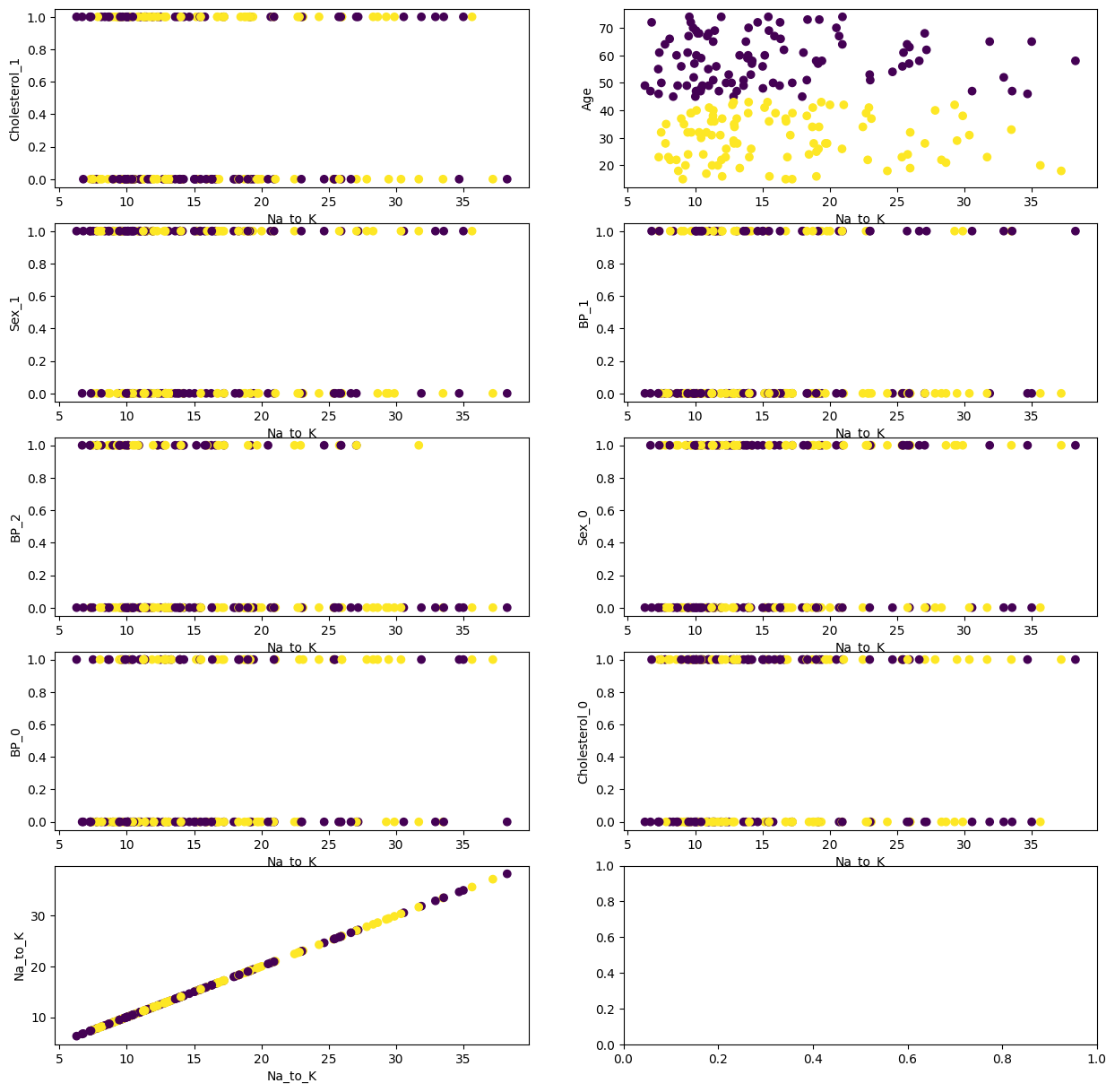
**Using clustering to check if it can help us improve accuracy**

Checking features with target to see which one makes clear clusters

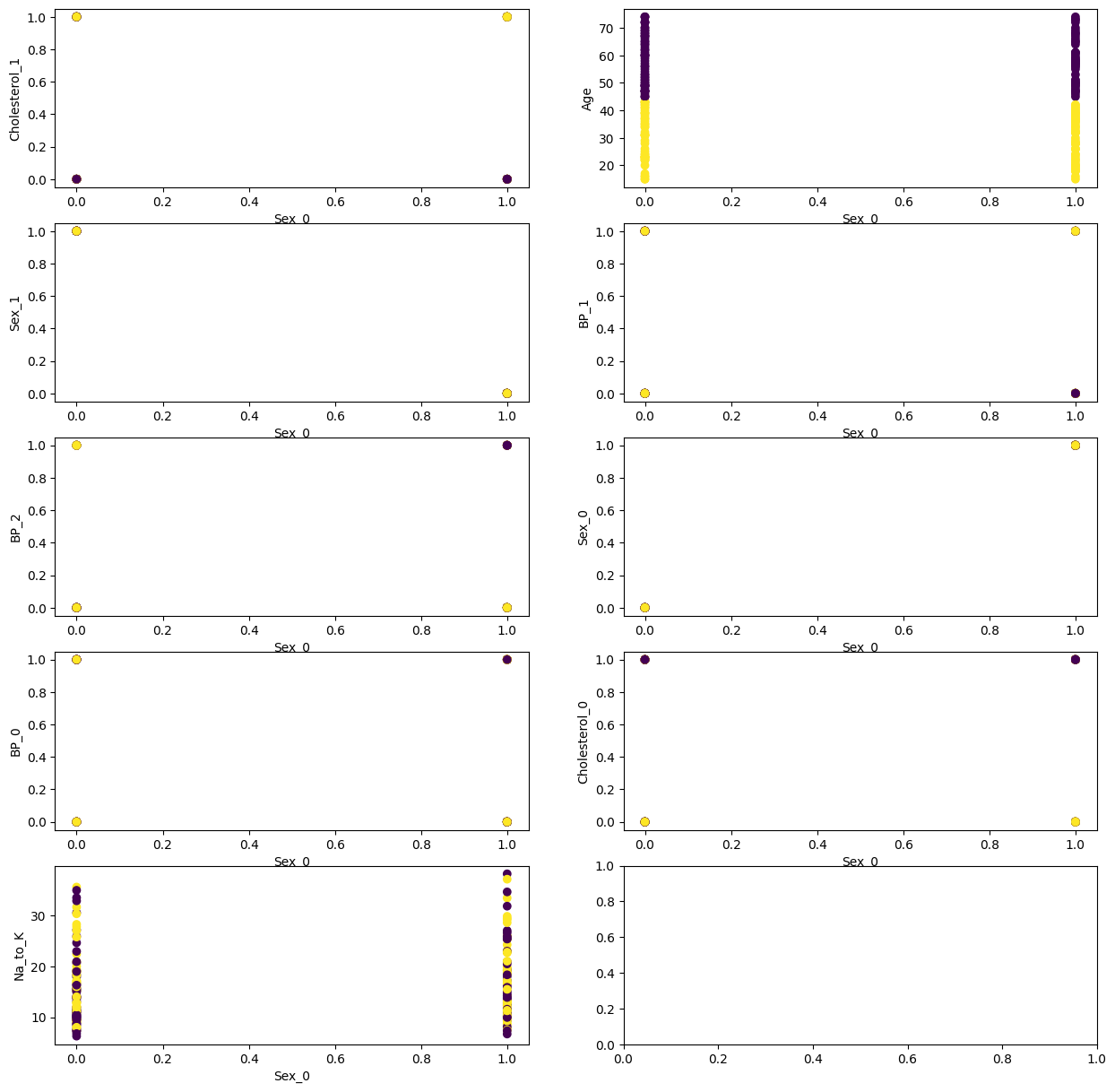
We are trying to use independent variables to see if there is any clusters being formed in the data so that we can use that as another independent variable in evaluation.

Age:

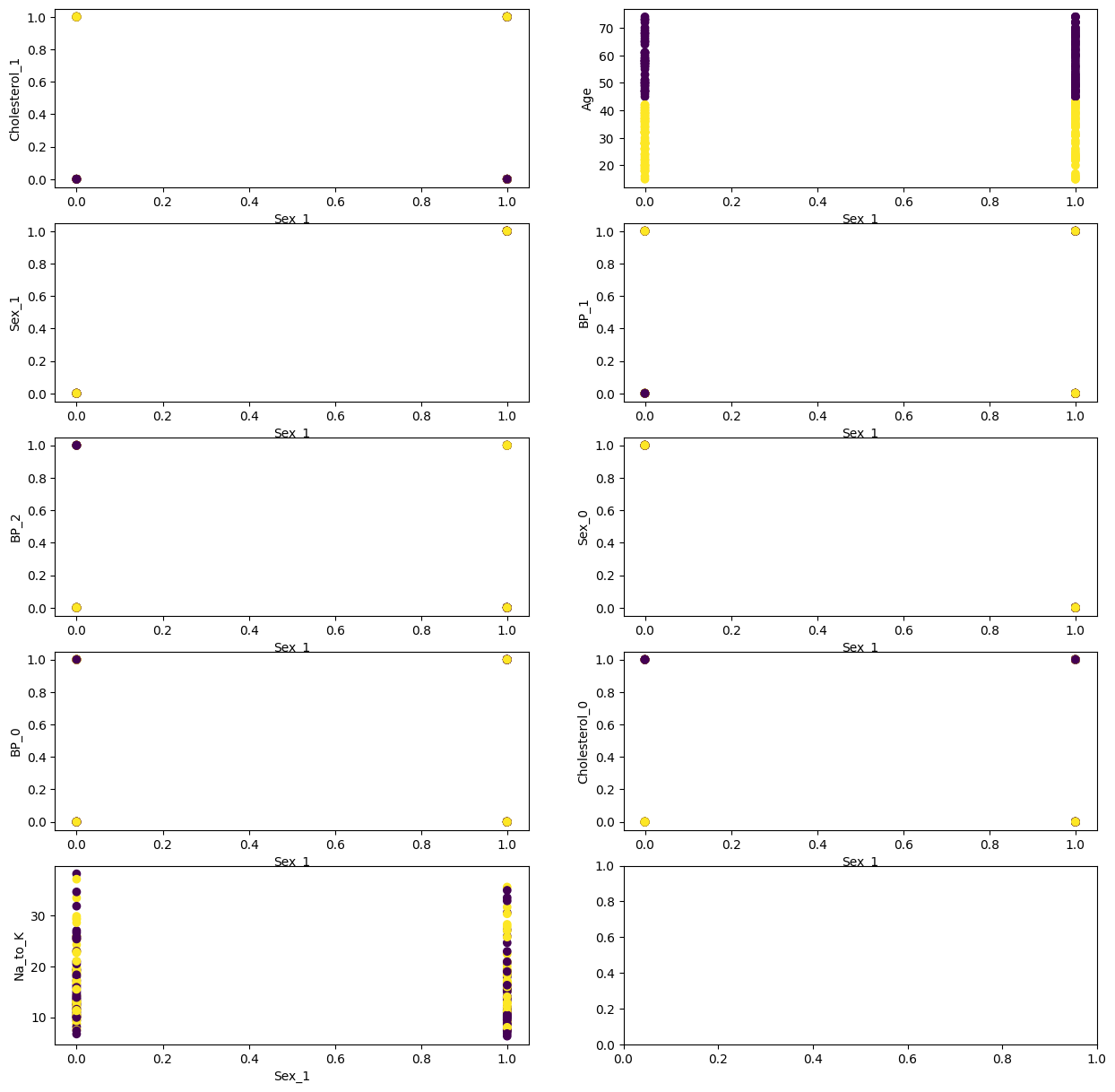


Na\_to\_K:

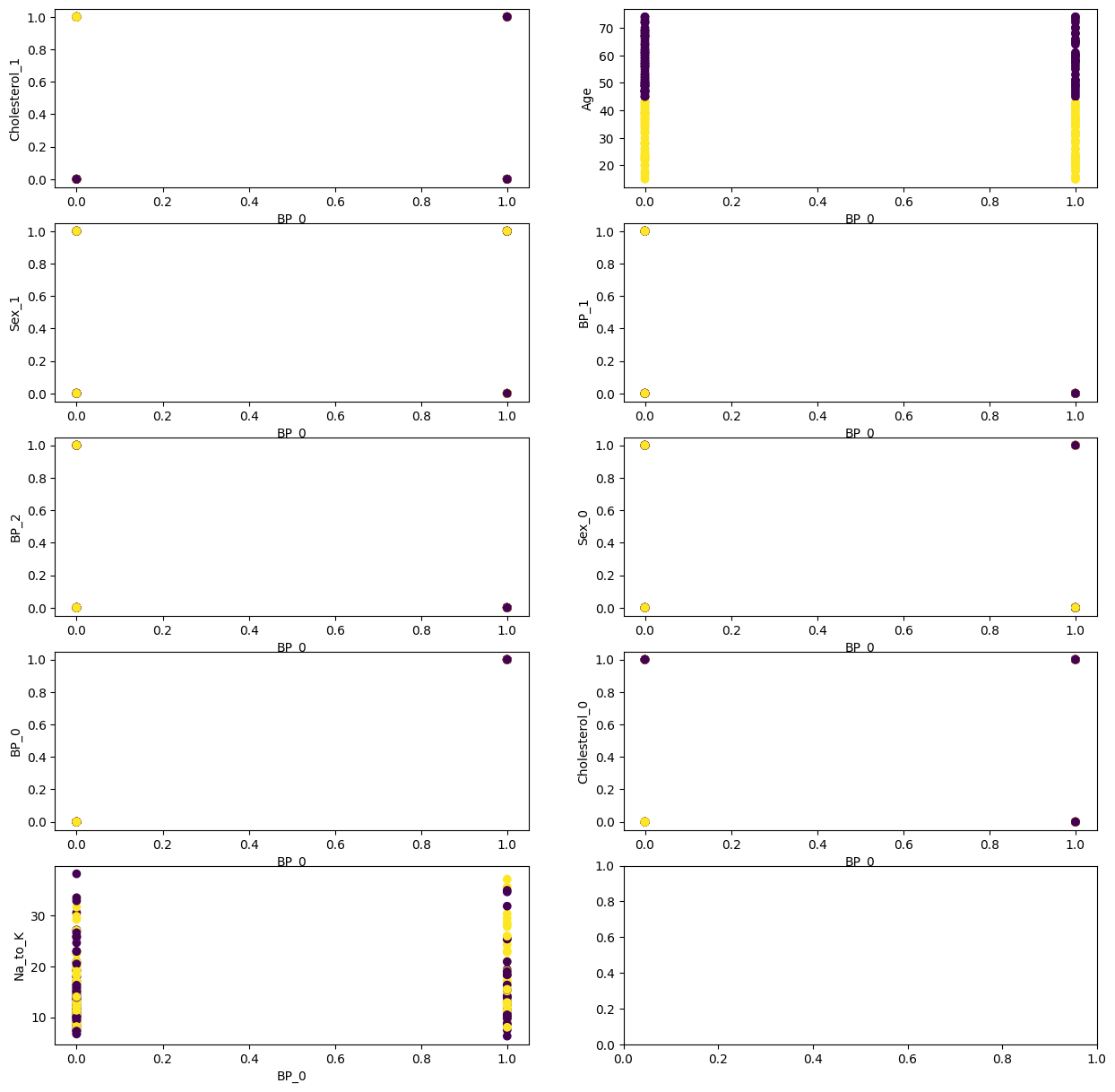
Sex\_0



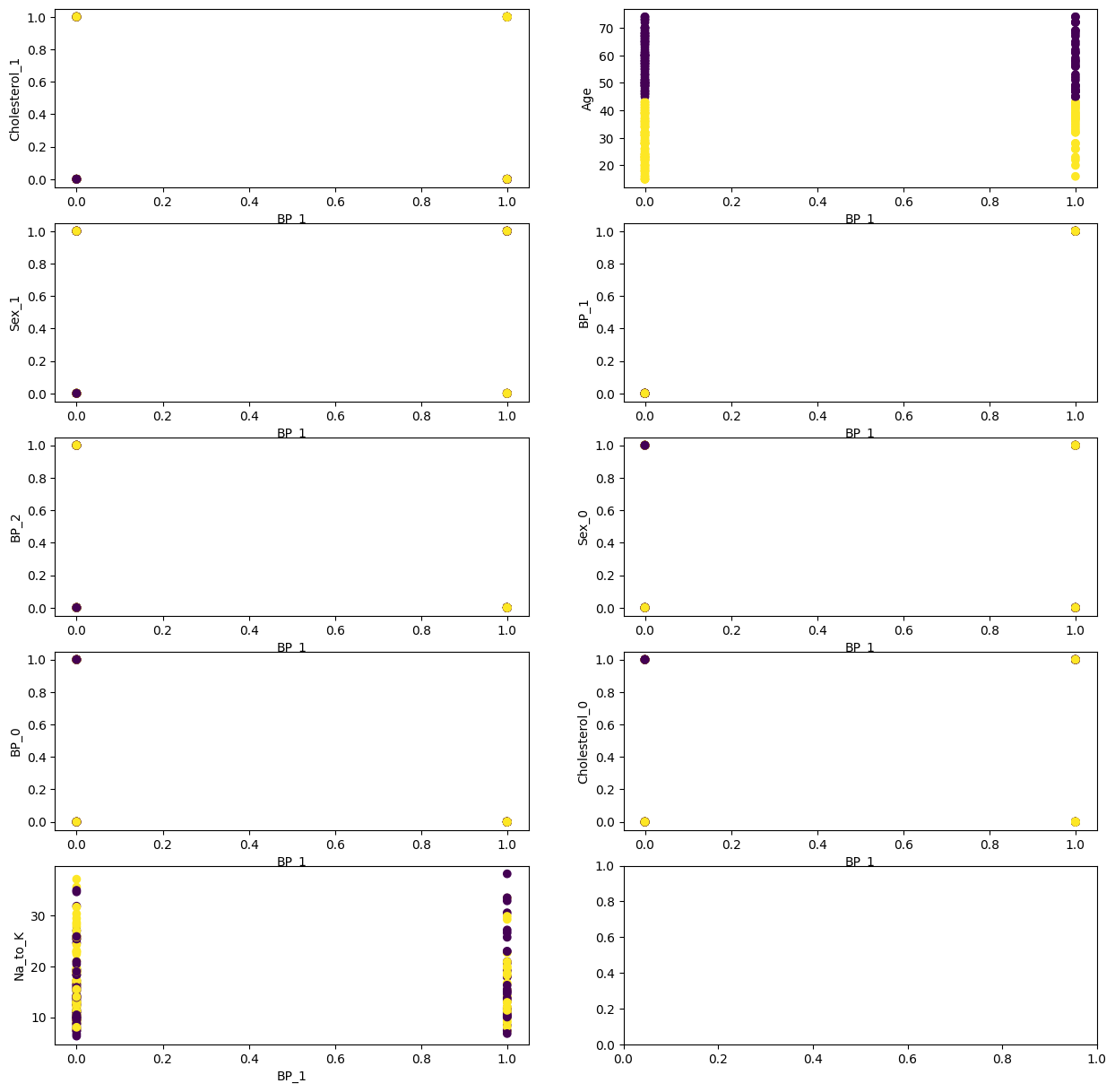
Sex\_1:



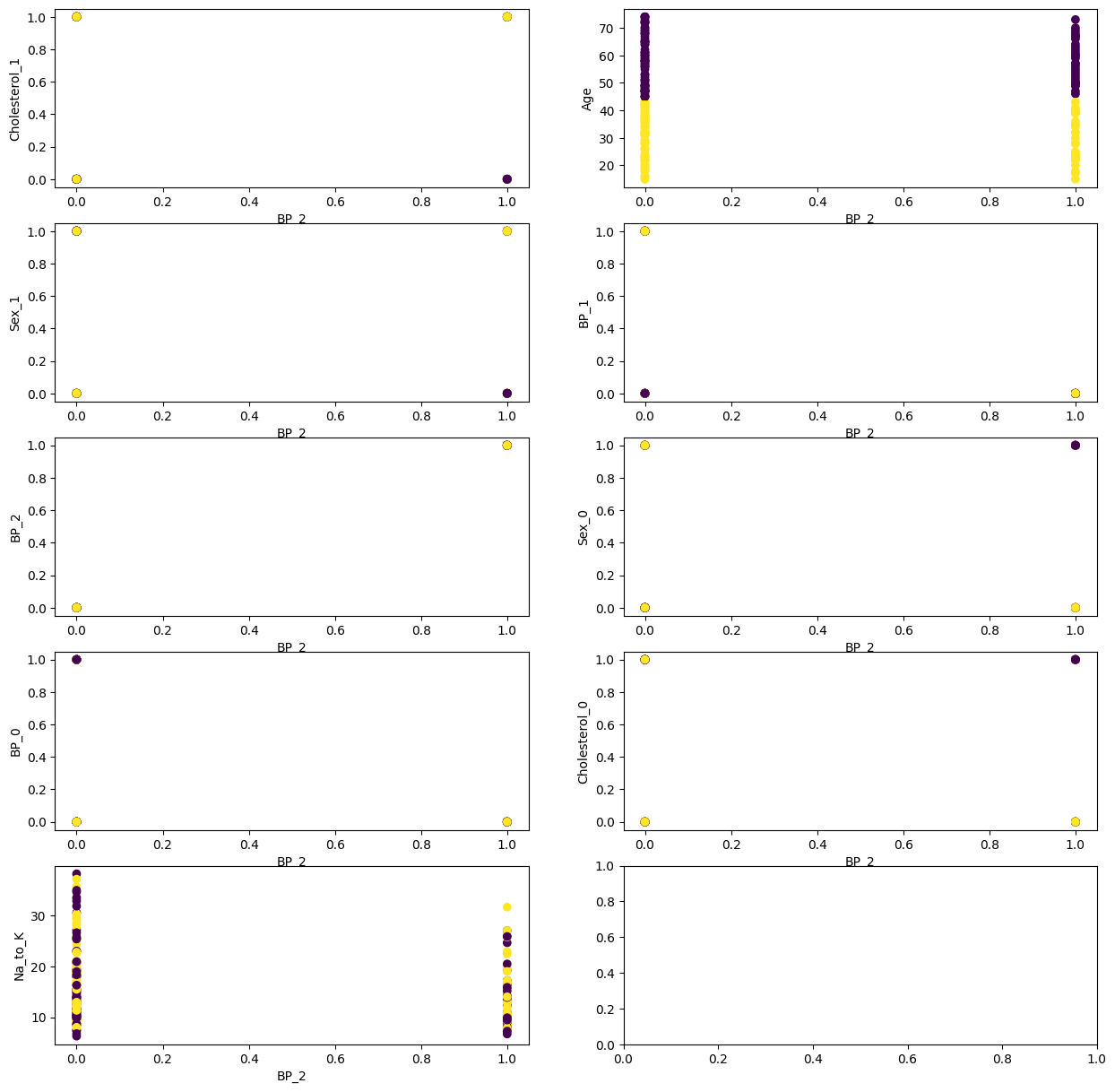
BP\_0:



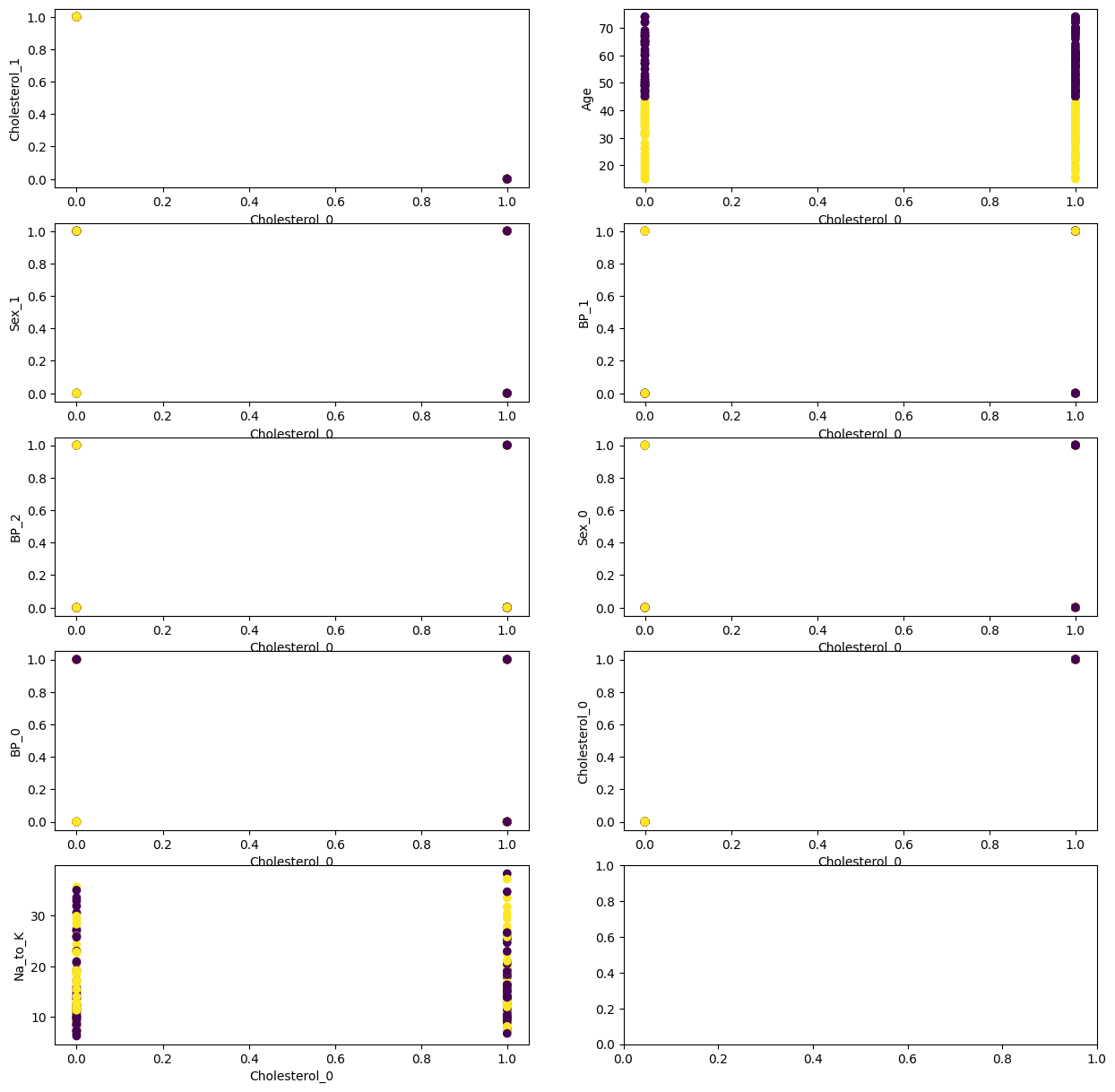
BP\_1



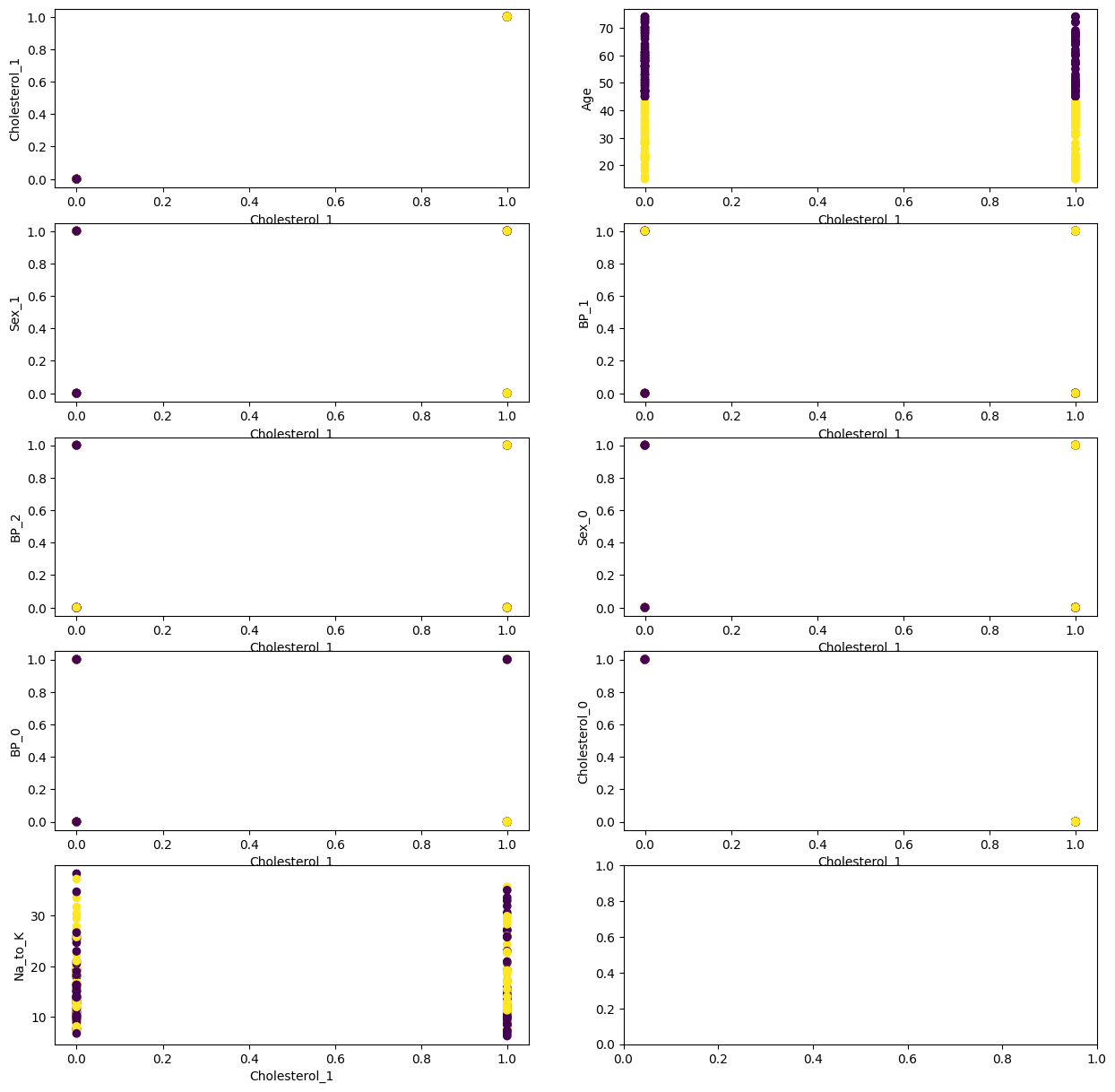
BP\_2:



Cholesterol\_0:



Cholesterol\_1:



**Analysis:**

* Age is forming clear clusters using all the independent variables.
* In column cluster, 1 represents data points belonging to cluster 1, 0 represents data points belonging to cluster 0.
* As values of cluster are less in scale compared to other values in different columns, we are trying to come up with a value which will we in range with respect to other value.
* In this case as cement is forming clear cluster, we are using that column.
* For this purpose, we have grouped cluster column with cement column and found mean and median of their respective cluster.

**Model building and evaluation: PCA transformed data**

**Normal model:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | model\_name | accuracy\_score | RMSE |
| 3 | RandomForestClassifier | 0.85 | 1.335415 |
| 6 | GradientBoostingClassifier | 0.85 | 1.278019 |
| 7 | XGBClassifier | 0.816667 | 1.538397 |
| 1 | DecisionTreeClassifier | 0.8 | 1.543805 |
| 0 | LogisticRegression | 0.733333 | 1.81659 |
| 4 | SVC | 0.633333 | 1.962142 |
| 2 | KneighborClassifier | 0.583333 | 1.966384 |
| 5 | AdaBoostClassifier | 0.483333 | 2.366432 |

**Applying Cross validation:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | model\_names | cv\_score | cv\_std |
| 3 | RandomForestClassifier | 0.875 | 0.033541 |
| 6 | GradientBoostingClassifier | 0.86 | 0.076811 |
| 7 | XGBClassifier | 0.845 | 0.04717 |
| 1 | DecisionTreeClassifier | 0.825 | 0.064226 |
| 0 | Logisticregression | 0.695 | 0.068739 |
| 2 | KNeighborsClassifier | 0.685 | 0.080777 |
| 4 | SVC | 0.675 | 0.078262 |
| 5 | AdaBoostClassifier | 0.47 | 0.078102 |

**Cross Validation post hyperparameter:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | model\_names | cv\_score | cv\_std |
| 1 | RandomForestClassifier | 0.86 | 0.053852 |
| 7 | XGBClassifier | 0.86 | 0.05831 |
| 6 | GradientBoostingClassifier | 0.855 | 0.065 |
| 2 | DecisionTreeClassifier | 0.82 | 0.064031 |
| 3 | KNeighborsClassifier | 0.81 | 0.07 |
| 0 | LogisticRegression | 0.695 | 0.068739 |
| 4 | SVC | 0.695 | 0.075664 |
| 5 | AdaBoostClassifier | 0.575 | 0.095525 |

**Model building and evaluation: Cluster data**

**Normal model:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | model\_name | accuracy\_score | RMSE |
| 3 | RandomForestClassifier | 0.983333 | 0.129099 |
| 1 | DecisionTreeClassifier | 0.966667 | 0.182574 |
| 6 | GradientBoostingClassifier | 0.966667 | 0.182574 |
| 7 | XGBClassifier | 0.95 | 0.316228 |
| 0 | LogisticRegression | 0.9 | 0.774597 |
| 5 | AdaBoostClassifier | 0.833333 | 0.408248 |
| 2 | KneighborClassifier | 0.683333 | 1.176152 |
| 4 | SVC | 0.533333 | 2.19089 |

**Applying Cross validation:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | model\_names | cv\_score | cv\_std |
| 6 | GradientBoostingClassifier | 0.86 | 0.076811 |
| 3 | RandomForestClassifier | 0.855 | 0.041533 |
| 7 | XGBClassifier | 0.845 | 0.04717 |
| 1 | DecisionTreeClassifier | 0.815 | 0.063443 |
| 0 | Logisticregression | 0.695 | 0.068739 |
| 2 | KNeighborsClassifier | 0.685 | 0.080777 |
| 4 | SVC | 0.675 | 0.078262 |
| 5 | AdaBoostClassifier | 0.47 | 0.078102 |

**Cross Validation post hyperparameter:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | model\_names | cv\_score | cv\_std |
| 1 | RandomForestClassifier | 0.99 | 0.02 |
| 6 | GradientBoostingClassifier | 0.985 | 0.022913 |
| 7 | XGBClassifier | 0.985 | 0.022913 |
| 2 | DecisionTreeClassifier | 0.94 | 0.08 |
| 0 | LogisticRegression | 0.925 | 0.068007 |
| 5 | AdaBoostClassifier | 0.835 | 0.032016 |
| 4 | SVC | 0.76 | 0.10198 |
| 3 | KNeighborsClassifier | 0.735 | 0.063443 |

**Model building and evaluation: RFE**

**Normal model:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | model\_name | accuracy\_score | RMSE |
| 1 | DecisionTreeClassifier | 0.966667 | 0.182574 |
| 3 | RandomForestClassifier | 0.966667 | 0.182574 |
| 6 | GradientBoostingClassifier | 0.966667 | 0.182574 |
| 7 | XGBClassifier | 0.95 | 0.316228 |
| 0 | LogisticRegression | 0.883333 | 0.866025 |
| 5 | AdaBoostClassifier | 0.833333 | 0.408248 |
| 4 | SVC | 0.716667 | 1.36626 |
| 2 | KneighborClassifier | 0.633333 | 1.30384 |

**Applying Cross validation:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | model\_names | cv\_score | cv\_std |
| 1 | DecisionTreeClassifier | 0.99 | 0.02 |
| 3 | RandomForestClassifier | 0.99 | 0.02 |
| 7 | XGBClassifier | 0.99 | 0.02 |
| 6 | GradientBoostingClassifier | 0.98 | 0.024495 |
| 0 | Logisticregression | 0.935 | 0.045 |
| 5 | AdaBoostClassifier | 0.835 | 0.032016 |
| 4 | SVC | 0.72 | 0.033166 |
| 2 | KNeighborsClassifier | 0.71 | 0.073485 |

**Cross Validation post hyperparameter:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | model\_names | cv\_score | cv\_std |
| 1 | RandomForestClassifier | 0.99 | 0.02 |
| 2 | DecisionTreeClassifier | 0.985 | 0.022913 |
| 6 | GradientBoostingClassifier | 0.985 | 0.022913 |
| 7 | XGBClassifier | 0.985 | 0.022913 |
| 0 | LogisticRegression | 0.935 | 0.045 |
| 5 | AdaBoostClassifier | 0.835 | 0.032016 |
| 4 | SVC | 0.74 | 0.076811 |
| 3 | KNeighborsClassifier | 0.725 | 0.064226 |

**Model building and evaluation**: Using XGBClassifier.

**Normal model:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | model\_name | accuracy\_score | RMSE |
| 1 | DecisionTreeClassifier | 0.966667 | 0.182574 |
| 3 | RandomForestClassifier | 0.966667 | 0.182574 |
| 6 | GradientBoostingClassifier | 0.966667 | 0.182574 |
| 7 | XGBClassifier | 0.95 | 0.316228 |
| 0 | LogisticRegression | 0.85 | 1.072381 |
| 5 | AdaBoostClassifier | 0.833333 | 0.408248 |
| 4 | SVC | 0.716667 | 1.36626 |
| 2 | KneighborClassifier | 0.65 | 1.24499 |

**Applying Cross validation:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | model\_names | cv\_score | cv\_std |
| 1 | DecisionTreeClassifier | 0.99 | 0.02 |
| 3 | RandomForestClassifier | 0.99 | 0.02 |
| 7 | XGBClassifier | 0.99 | 0.02 |
| 6 | GradientBoostingClassifier | 0.98 | 0.024495 |
| 0 | Logisticregression | 0.93 | 0.045826 |
| 5 | AdaBoostClassifier | 0.835 | 0.032016 |
| 4 | SVC | 0.72 | 0.033166 |
| 2 | KNeighborsClassifier | 0.715 | 0.089582 |

**Cross Validation post hyperparameter:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | model\_names | cv\_score | cv\_std |
| 1 | RandomForestClassifier | 0.99 | 0.02 |
| 6 | GradientBoostingClassifier | 0.985 | 0.022913 |
| 7 | XGBClassifier | 0.985 | 0.022913 |
| 2 | DecisionTreeClassifier | 0.96 | 0.037417 |
| 0 | LogisticRegression | 0.93 | 0.045826 |
| 5 | AdaBoostClassifier | 0.835 | 0.032016 |
| 4 | SVC | 0.755 | 0.075664 |
| 3 | KNeighborsClassifier | 0.72 | 0.074833 |