2. MinMax Scaler
3. Binarizer
4. Standard Scaler

```
# Apply Standard Scaler
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
SSX = scaler.fit_transform(X)
```

Step 6: Splitting the Data into Training data & Test Data

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3,
random_state=0)

print(X.shape, X_train.shape, X_test.shape)
```

Step 7: Building Classification Algorithm

7.1) Logistic Regression

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(solver='liblinear',multi_class='ovr')
lr.fit(X_train,Y_train)
```

7.2) K-Nearest Neighbors Classifier(KNN)

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(X_train,Y_train)
```

7.3) Naive-Bayes Classifier

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(X_train, Y_train)
```

7.4) Support Vector Machine (SVM)

```
from sklearn.svm import SVC
sv = SVC(kernel='linear')
sv.fit(X_train,Y_train)
```

7.5) Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(X_train,Y_train)
```

7.6) Random Forest

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=20, random_state=12,max_depth=6)
rf.fit(X_train,Y_train)
```