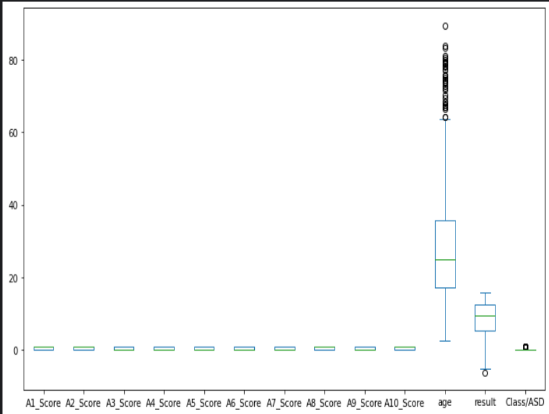
***Code-Autism Prediction Dataset***

Libraray:

* Numpy : for matrix and mathematics statmant
* Pandas: data analysis and processing
* Seaborn & Matplotlib : plotting & data exploration

Code🡪

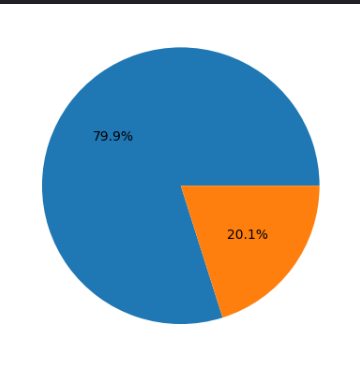
1. Which the data is NUM and which is CAT.
2. Drop for unimportant columns.
3. Solve the missing values.



1. Check to missing values.
2. Counted the number of values for (Class\ASD) to check whether the data is balanced or not ?!

0 : 639

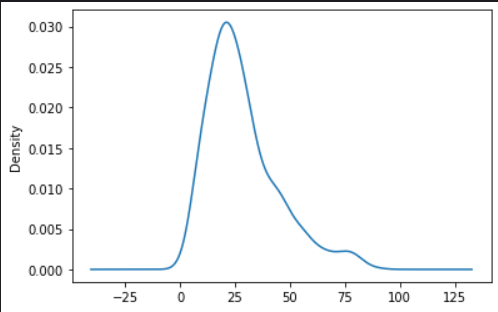
1 : 161

* 639/(639+161) =0.79

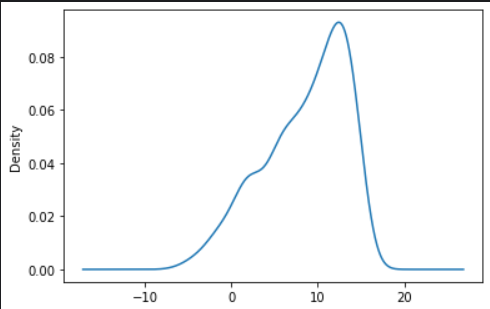
Note: so %80 from data belong to a certain category

The dataset it is unbalanced

1. I will use ordinalEncoder from preprocessing library to transformation the data category numeric values .
2. Check for outliers in numerical feature: By subplot.



1. Plotting for important numerical feature: By subplot.



Add an functions🡪

* LogisticRegresstion.

random\_state=234: Ensures reproducibility of results when running the code again

max\_iter=3000: Sets the maximum number of iterations for the model during training.

Cross validation: 0.905

* SVC

Cross validation: 0.868

* K-NN

Cross validation: 0.811

* Randomfor, adaboosting, gradboosting, voting

Cross validation score for the RandomForestClassifier: 0.916

Cross validation score for the AdaBoostClassifier: 0.876

Cross validation score for the GradientBoostingClassifie: 0.896

* CrossValidation

Voting to [('lr',lr), ('svc', svc), ('rfc', rfc), ('knc', knc)], voting='soft') to concatenation

Cross validation : 0.938

Voting to [('lr', lr), ('rfc', rfc), ('adab', adab), ('gradb', gradb)], voting='soft')

Cross validation: 0.965

StackingClassifier:

Cross validation score for the stacking classifier: 0.954

**When K=10**

Cross validation score for the LogisticRegression(max\_iter=3000, random\_state=234): 0.905

Cross validation score for the SVC(probability=True, random\_state=567): 0.868

Cross validation score for the KNeighborsClassifier(): 0.811

Cross validation score for the RandomForestClassifier(max\_depth=3, n\_jobs=-1): 0.915

Cross validation score for the AdaBoostClassifier(n\_estimators=100, random\_state=32389): 0.876

Cross validation score for the GradientBoostingClassifier(random\_state=34990): 0.896

**When K=7**

Cross validation score for the LogisticRegression(max\_iter=3000, random\_state=234): 0.907

Cross validation score for the SVC(probability=True, random\_state=567): 0.866

Cross validation score for the KNeighborsClassifier(): 0.816

Cross validation score for the RandomForestClassifier(max\_depth=3, n\_jobs=-1): 0.914

Cross validation score for the AdaBoostClassifier(n\_estimators=100, random\_state=32389): 0.884

Cross validation score for the GradientBoostingClassifier(random\_state=34990): 0.894

**When K=5**

Cross validation score for the LogisticRegression(max\_iter=3000, random\_state=234): 0.908

Cross validation score for the SVC(probability=True, random\_state=567): 0.867

Cross validation score for the KNeighborsClassifier(): 0.802

Cross validation score for the RandomForestClassifier(max\_depth=3, n\_jobs=-1): 0.917

Cross validation score for the AdaBoostClassifier(n\_estimators=100, random\_state=32389): 0.886

Cross validation score for the GradientBoostingClassifier(random\_state=34990): 0.899

**When K=3**

Cross validation score for the LogisticRegression(max\_iter=3000, random\_state=234): 0.907

Cross validation score for the SVC(probability=True, random\_state=567): 0.868

Cross validation score for the KNeighborsClassifier(): 0.793

Cross validation score for the RandomForestClassifier(max\_depth=3, n\_jobs=-1): 0.916

Cross validation score for the AdaBoostClassifier(n\_estimators=100, random\_state=32389): 0.864

Cross validation score for the GradientBoostingClassifier(random\_state=34990): 0.903

**NOTE:**The best K-fold when ( K = 3 )