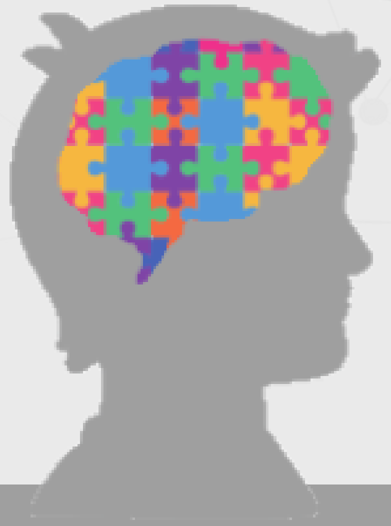
# AUTISM DETECTOR



# Problem discussion

Autism spectrum disorder (ASD) is a developmental disability caused by differences in the brain. Some people with ASD have a known difference, such as a genetic condition. Other causes are not yet known. Scientists believe there are multiple causes of ASD that act together to change the most common ways people develop.

People with ASD may behave, communicate, interact, and learn in ways that are different from most other people. There is often nothing about how they look that sets them apart from other people.

- Help to diagnose ASD in an automatic and faster way
- Faster screening
- Improve diagnose accuracy by including in the diagnose similar charateristics between patient

# Dataset

- 704 people
- 10 behavioral features
- Patient descriptive features

AQ10 test for adult

a1_score, a2_score, a3_score.....

- Age
- Gender
- Ethnicity
- Born with jaundice
- Presence of immediate family member with PDD

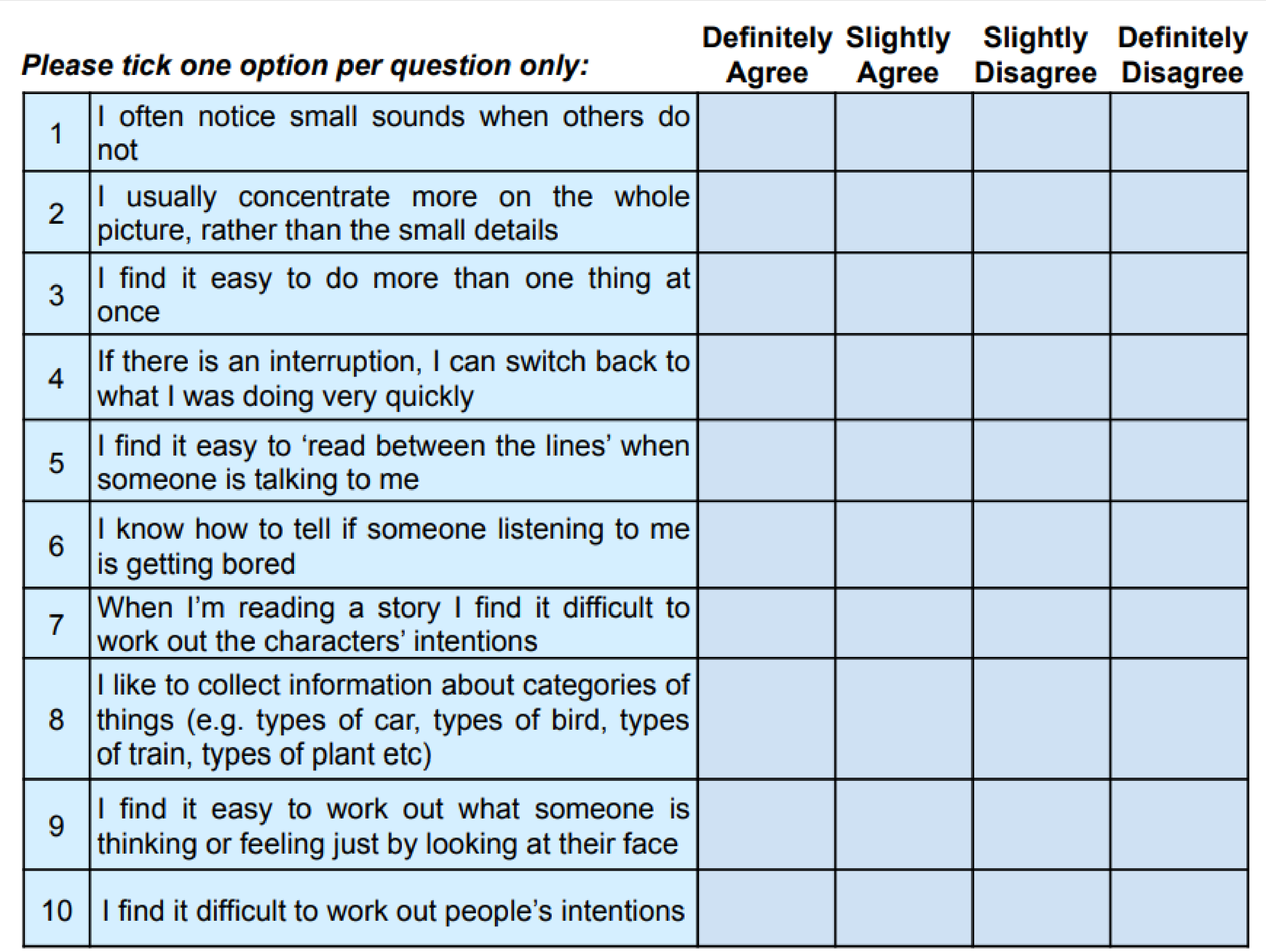
**Are all proved to be effective in detecting ASD cases**

*https://www.kaggle.com/datasets/faizunnabi/autism-screening*

# Dataset

Autism Spectrum Quotient (AQ)
Test for adult

1 = Definitely or Slightly Agree
0 = Definitely or Slightly Disagree

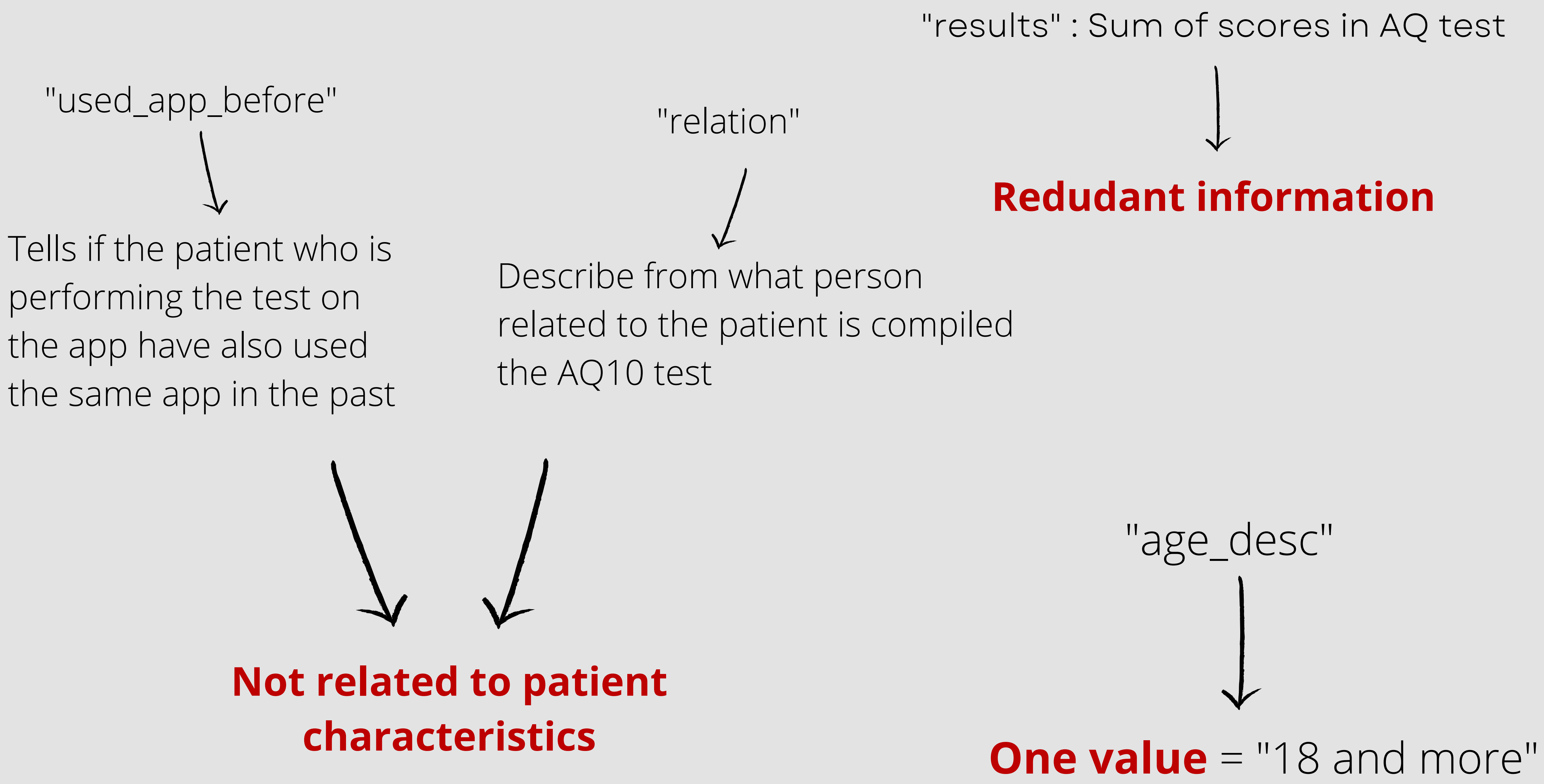


| *Please tick one option per question only:* | | Definitely Agree | Slightly Agree | Slightly Disagree | Definitely Disagree |
|---|---|---|---|---|---|
| 1 | I often notice small sounds when others do not | | | | |
| 2 | I usually concentrate more on the whole picture, rather than the small details | | | | |
| 3 | I find it easy to do more than one thing at once | | | | |
| 4 | If there is an interruption, I can switch back to what I was doing very quickly | | | | |
| 5 | I find it easy to 'read between the lines' when someone is talking to me | | | | |
| 6 | I know how to tell if someone listening to me is getting bored | | | | |
| 7 | When I'm reading a story I find it difficult to work out the characters' intentions | | | | |
| 8 | I like to collect information about categories of things (e.g. types of car, types of bird, types of train, types of plant etc) | | | | |
| 9 | I find it easy to work out what someone is thinking or feeling just by looking at their face | | | | |
| 10 | I find it difficult to work out people's intentions | | | | |

# Preprocessing

Features selection method is not a good way to preprocess this data

If we inspect the dataset manually we can see that there are some attributes which are useless



"used_app_before"

Tells if the patient who is performing the test on the app have also used the same app in the past

"relation"

Describe from what person related to the patient is compiled the AQ10 test

**Not related to patient characteristics**

"results" : Sum of scores in AQ test

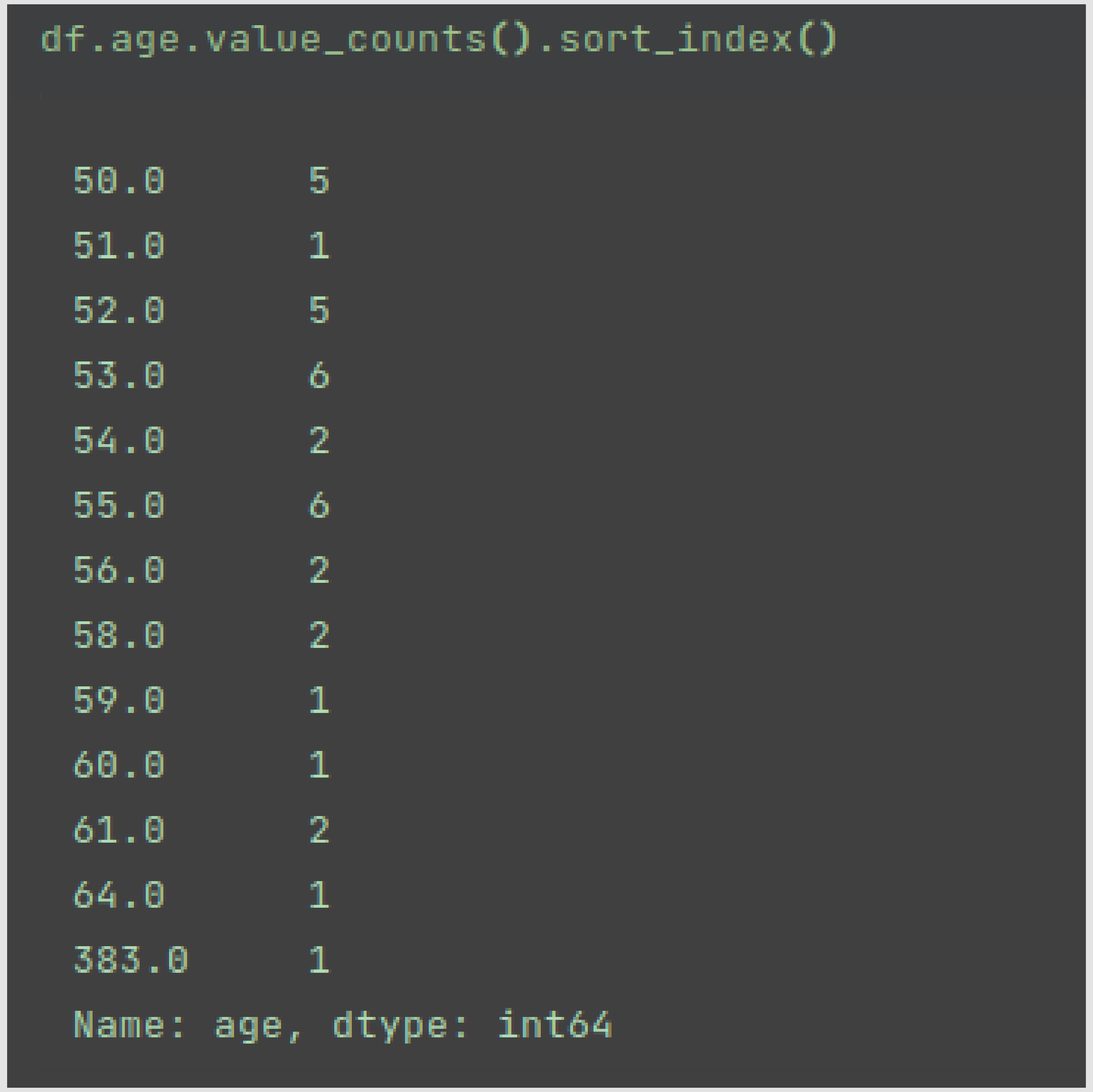
**Redudant information**

"age_desc"

**One value** = "18 and more"

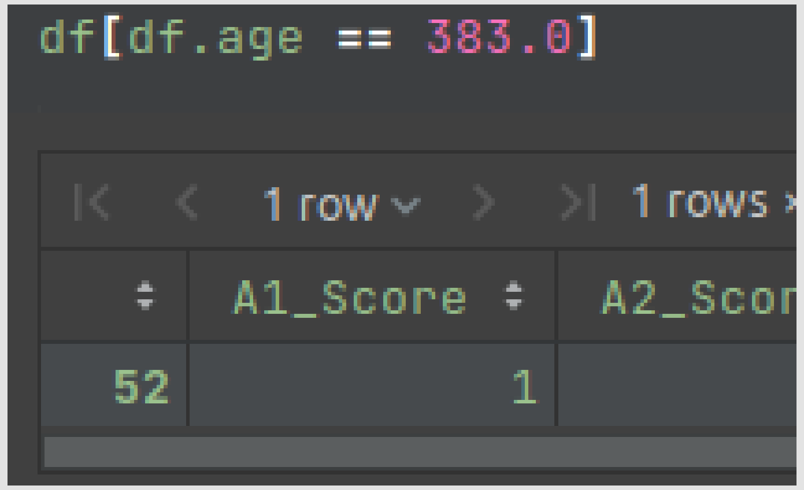
# Preprocessing

By sorting and counting values of age we can see that there is an anomalous value.



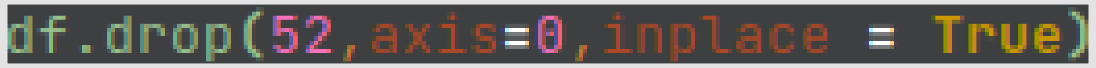
```
df.age.value_counts().sort_index()


 50.0      5
 51.0      1
 52.0      5
 53.0      6
 54.0      2
 55.0      6
 56.0      2
 58.0      2
 59.0      1
 60.0      1
 61.0      2
 64.0      1
 383.0     1
 Name: age, dtype: int64
```



```
df[df.age == 383.0]
```

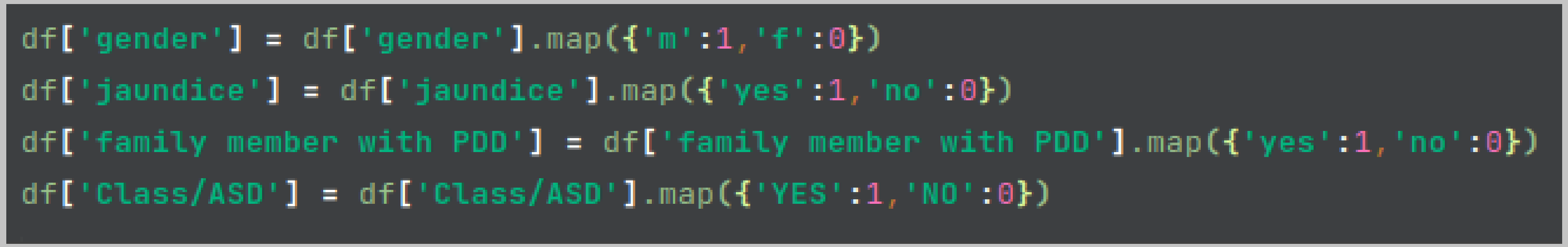
| | A1_Score | A2_Scor |
|---|---|---|
| 52 | 1 | |



```
df.drop(52,axis=0,inplace = True)
```

# Preprocessing

Encode the value of “jaundice”,”gender”,”family member with PDD”,”Class/ASD” to 0 and 1 given that they can assume only two values



```
df['gender'] = df['gender'].map({'m':1,'f':0})
df['jaundice'] = df['jaundice'].map({'yes':1,'no':0})
df['family member with PDD'] = df['family member with PDD'].map({'yes':1,'no':0})
df['Class/ASD'] = df['Class/ASD'].map({'YES':1,'NO':0})
```

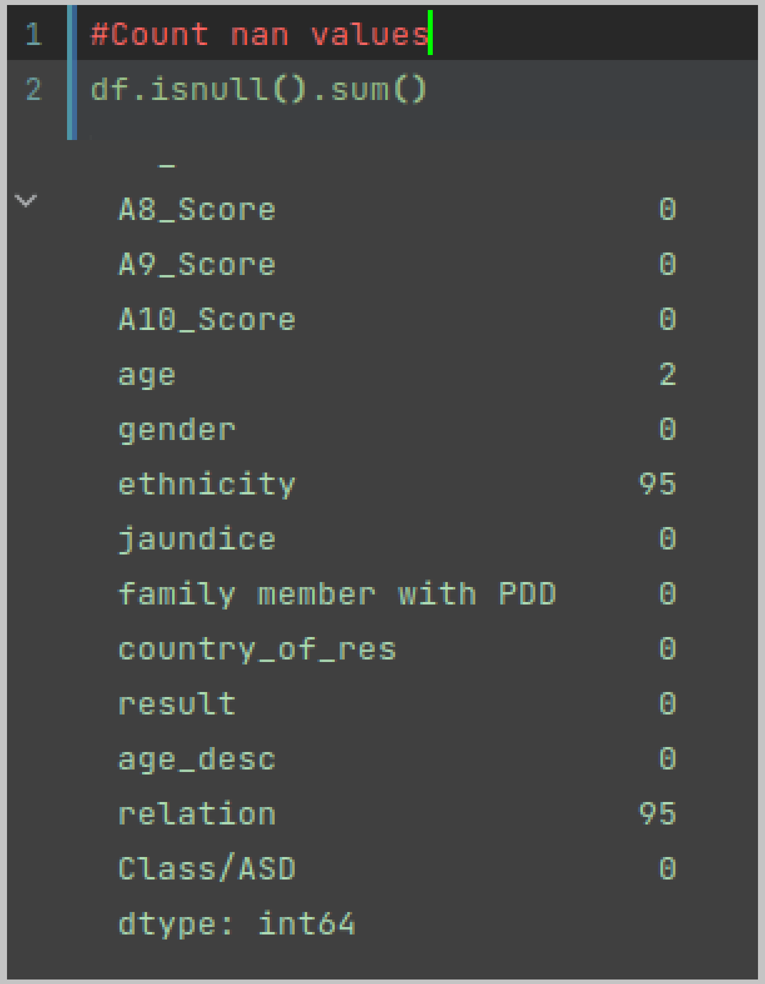
The question of the AQtest change  based on the age category of the patient but in the dataset are only present patient that the test categorize as "adult"

↓

For this reason, it was decided to not categorize age, so giving the classifier the possibility to use the age as a continuous valued attribute

# Preprocessing

Inspecting missing values



```
#Count nan values
df.isnull().sum()
```

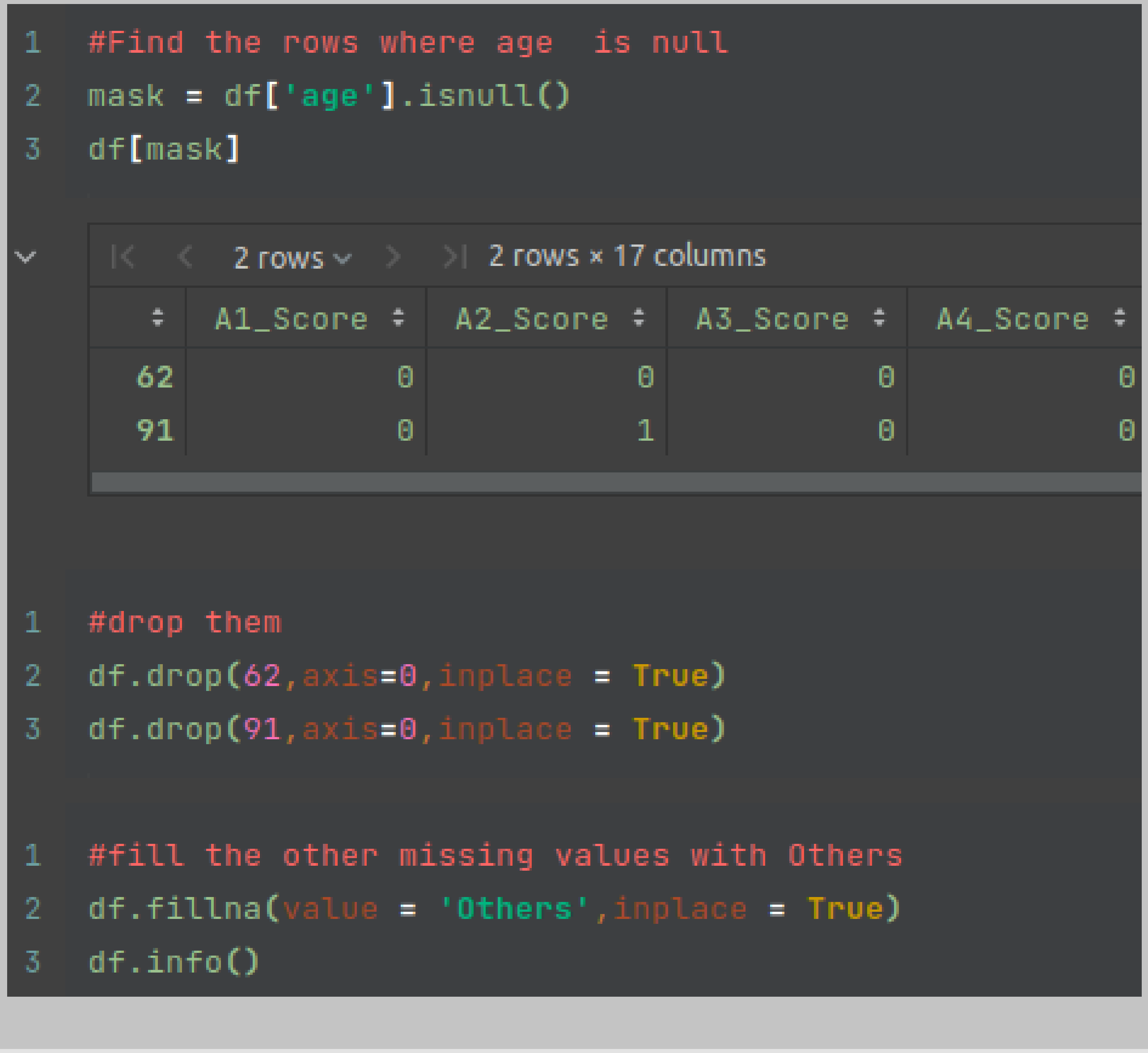
```
A8_Score                0
A9_Score                0
A10_Score               0
age                     2
gender                  0
ethnicity              95
jaundice                0
family member with PDD  0
country_of_res          0
result                  0
age_desc                0
relation               95
Class/ASD               0
dtype: int64
```

Age: 2 null values

Ethnicity: 95 null values

# Preprocessing

Inspecting null values of age



```
#Find the rows where age  is null
mask = df['age'].isnull()
df[mask]
```

| | A1_Score | A2_Score | A3_Score | A4_Score |
|---|---|---|---|---|
| 62 | 0 | 0 | 0 | 0 |
| 91 | 0 | 1 | 0 | 0 |

```
#drop them
df.drop(62,axis=0,inplace = True)
df.drop(91,axis=0,inplace = True)
```

```
#fill the other missing values with Others
df.fillna(value = 'Others',inplace = True)
df.info()
```

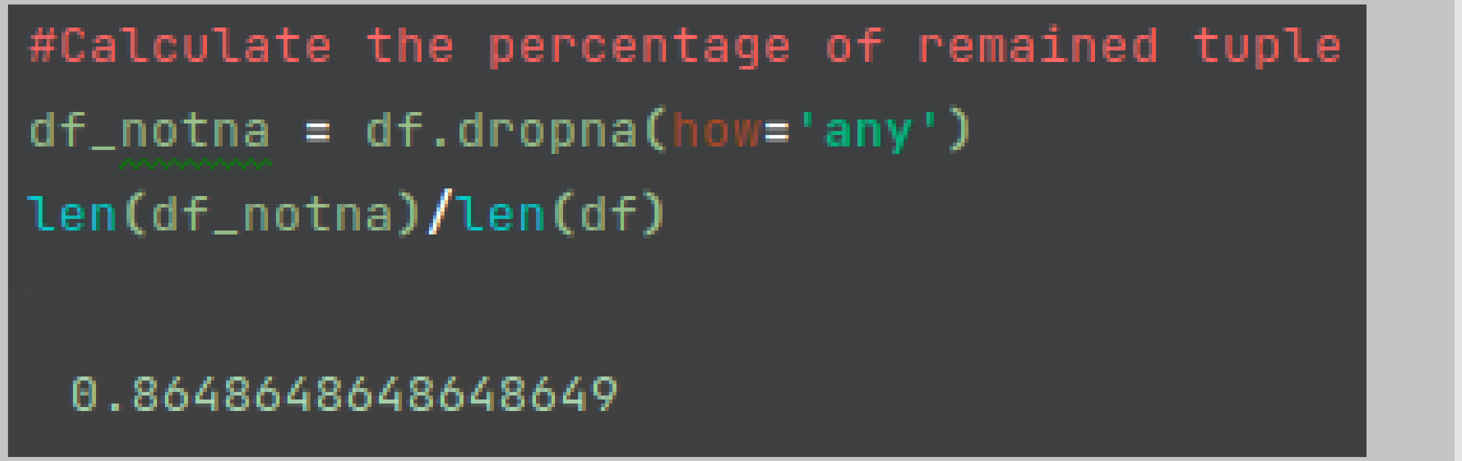
Rows are dropped because they have two descripitive charateristics as null

# Preprocessing

## Handling missing values

We need to choose to eliminate tuples with missing values or to keep them by filling missing values with a NotSpecified value

By ignoring the tuple we can see that we lose the 13,82% of tuples that appears to be not so much, in fact, there are no very big differences in the distribution of the class.

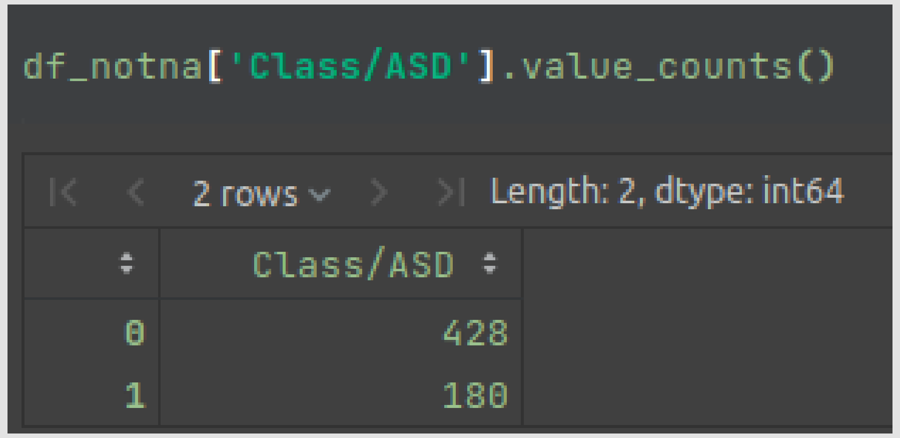


```
#Calculate the percentage of remained tuple
df_notna = df.dropna(how='any')
len(df_notna)/len(df)

 0.8648648648648649
```

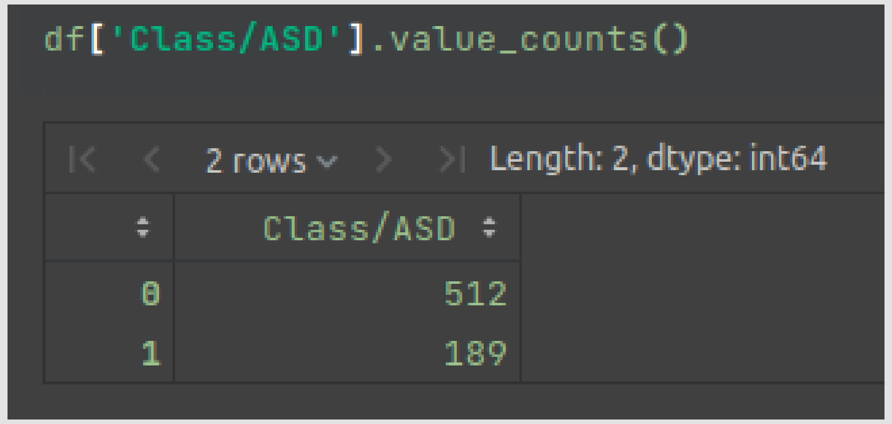
# Preprocessing

We can see that in both cases classes are not so imbalanced but when we ignore the null tuple we balance a little bit the classes because we remove a lot of negative tuple with respect to positive tuple.



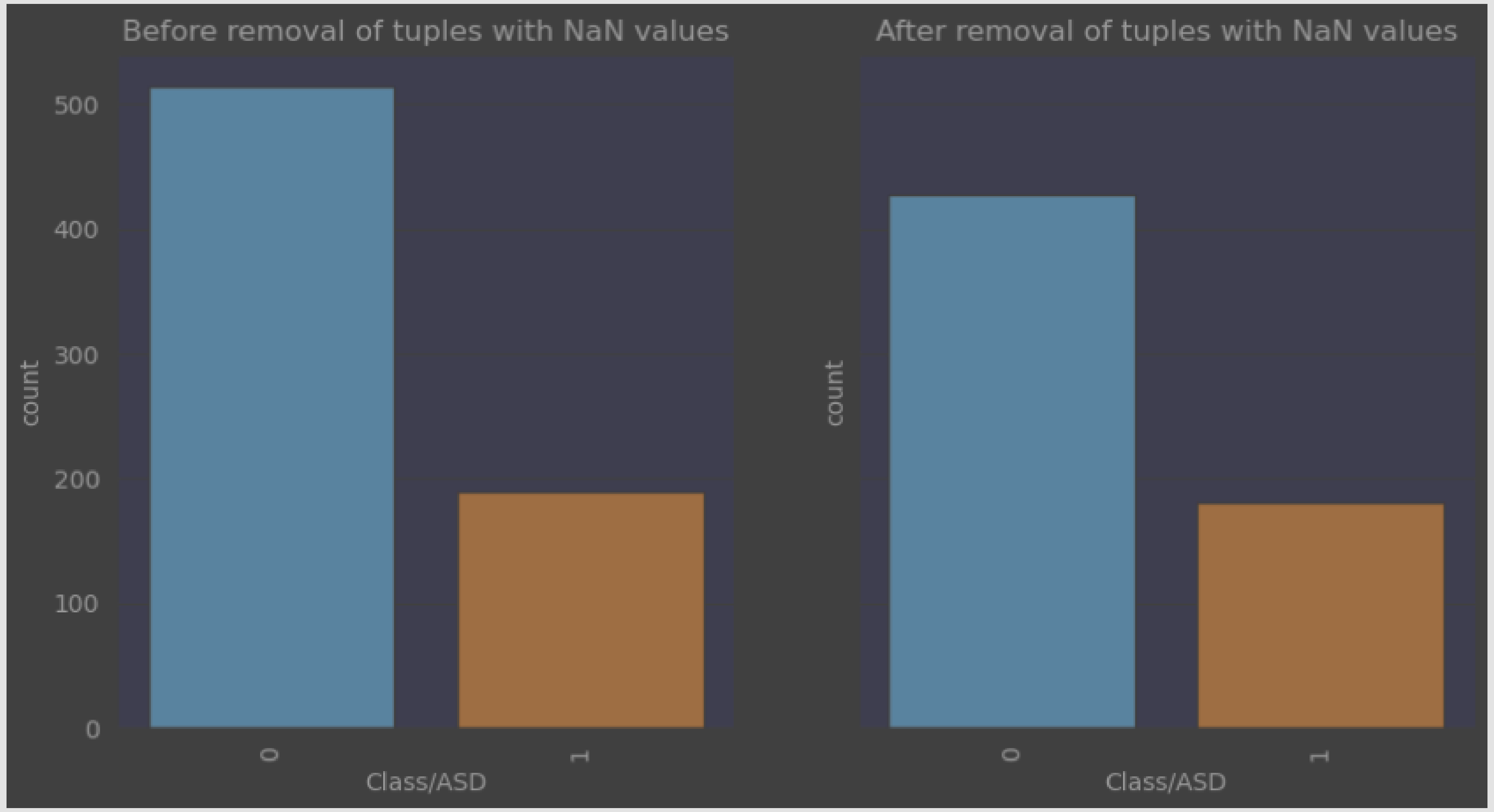
```
df_notna['Class/ASD'].value_counts()
```

```
|<  <  2 rows v  >  >|  Length: 2, dtype: int64
      Class/ASD
0           428
1           180
```



```
df['Class/ASD'].value_counts()
```

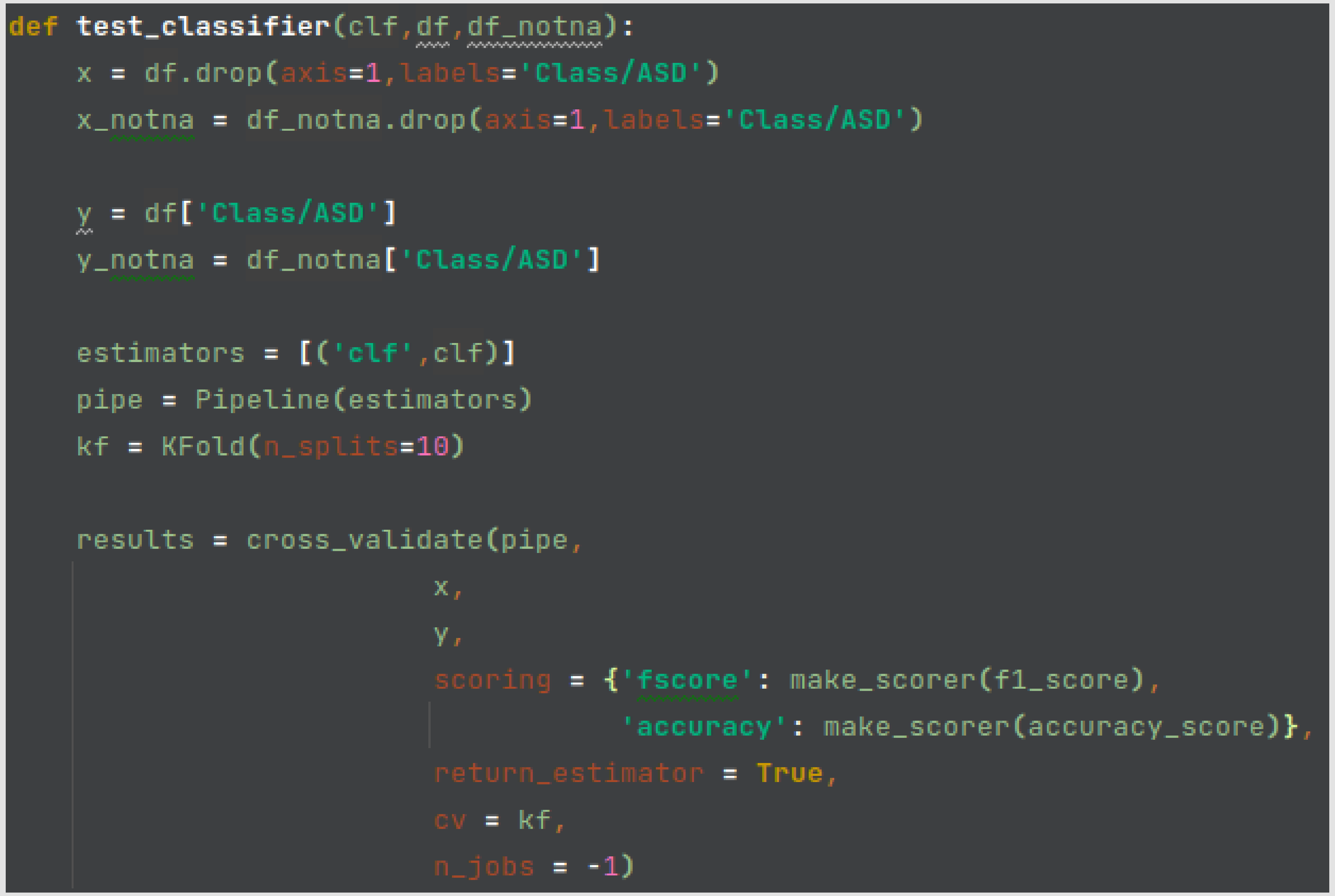
```
|<  <  2 rows v  >  >|  Length: 2, dtype: int64
      Class/ASD
0           512
1           189
```



# Evaluation and comparison of classifiers

How to compare classifier?

- Cross validation method for both dataset
- F-score measure to include recall
- Use t-test to asses if the two results set are different due to chance or there is a statistical significant difference



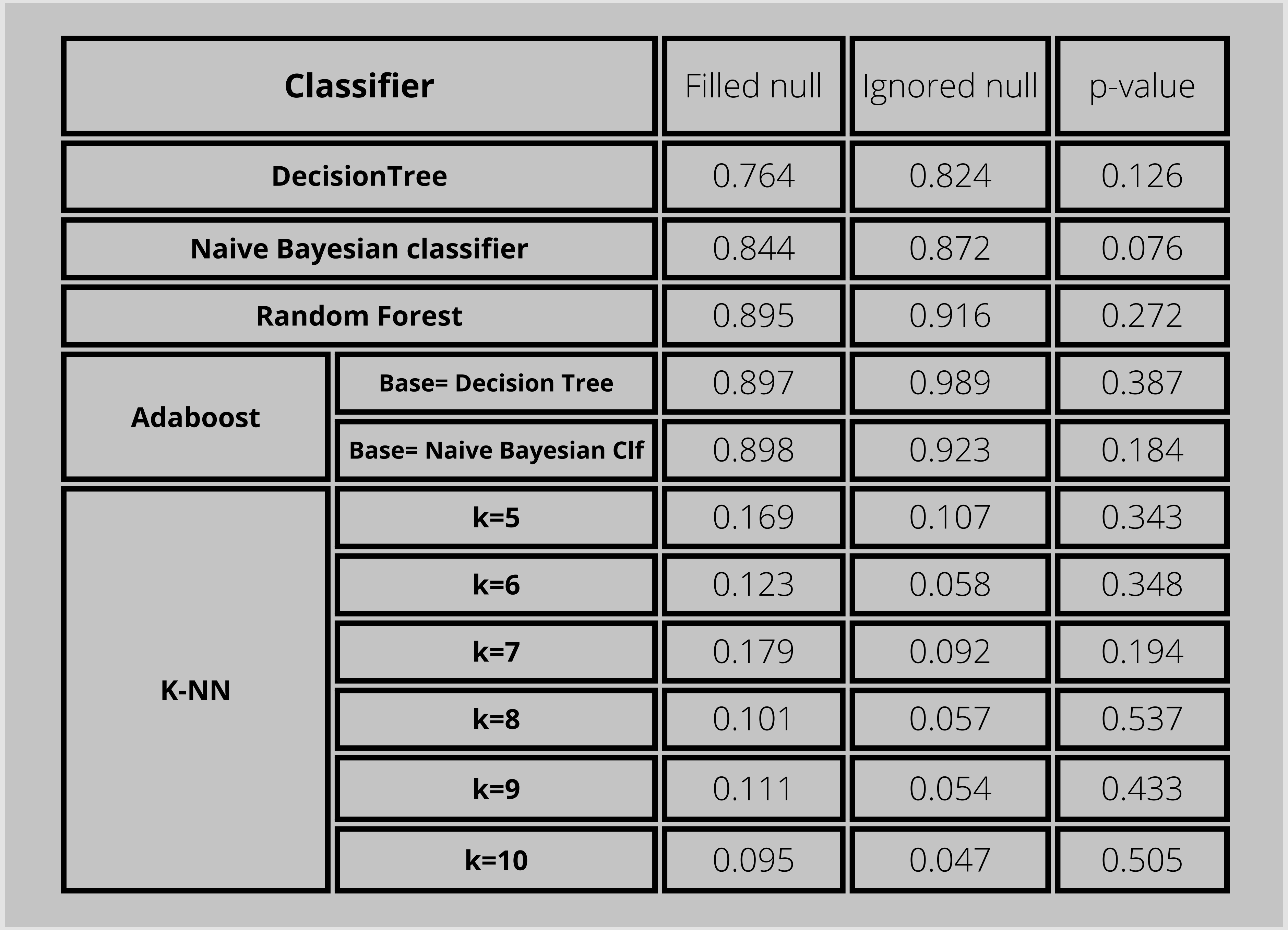
```
def test_classifier(clf,df,df_notna):
    x = df.drop(axis=1,labels='Class/ASD')
    x_notna = df_notna.drop(axis=1,labels='Class/ASD')

    y = df['Class/ASD']
    y_notna = df_notna['Class/ASD']

    estimators = [('clf',clf)]
    pipe = Pipeline(estimators)
    kf = KFold(n_splits=10)

    results = cross_validate(pipe,
                         x,
                         y,
                         scoring = {'fscore': make_scorer(f1_score),
                                    'accuracy': make_scorer(accuracy_score)},
                         return_estimator = True,
                         cv = kf,
                         n_jobs = -1)
```

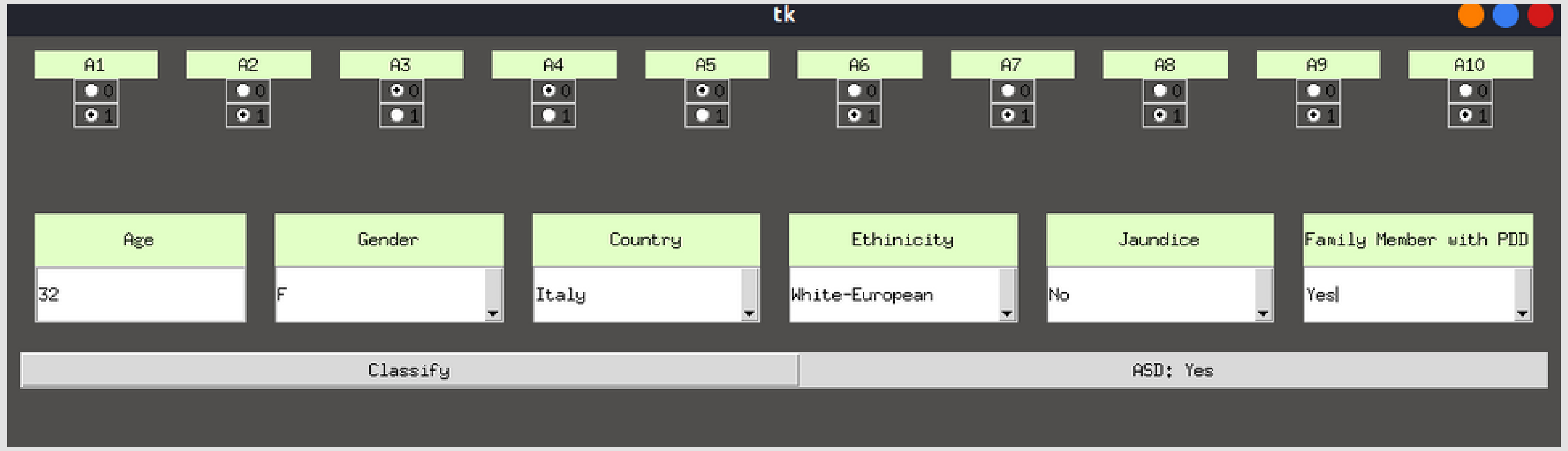
# Result



| **Classifier** | | Filled null | Ignored null | p-value |
|---|---|---|---|---|
| **DecisionTree** | | 0.764 | 0.824 | 0.126 |
| **Naive Bayesian classifier** | | 0.844 | 0.872 | 0.076 |
| **Random Forest** | | 0.895 | 0.916 | 0.272 |
| **Adaboost** | **Base= Decision Tree** | 0.897 | 0.989 | 0.387 |
| | **Base= Naive Bayesian Clf** | 0.898 | 0.923 | 0.184 |
| **K-NN** | **k=5** | 0.169 | 0.107 | 0.343 |
| | **k=6** | 0.123 | 0.058 | 0.348 |
| | **k=7** | 0.179 | 0.092 | 0.194 |
| | **k=8** | 0.101 | 0.057 | 0.537 |
| | **k=9** | 0.111 | 0.054 | 0.433 |
| | **k=10** | 0.095 | 0.047 | 0.505 |

# Implementation

GUI : Tkinter



tk

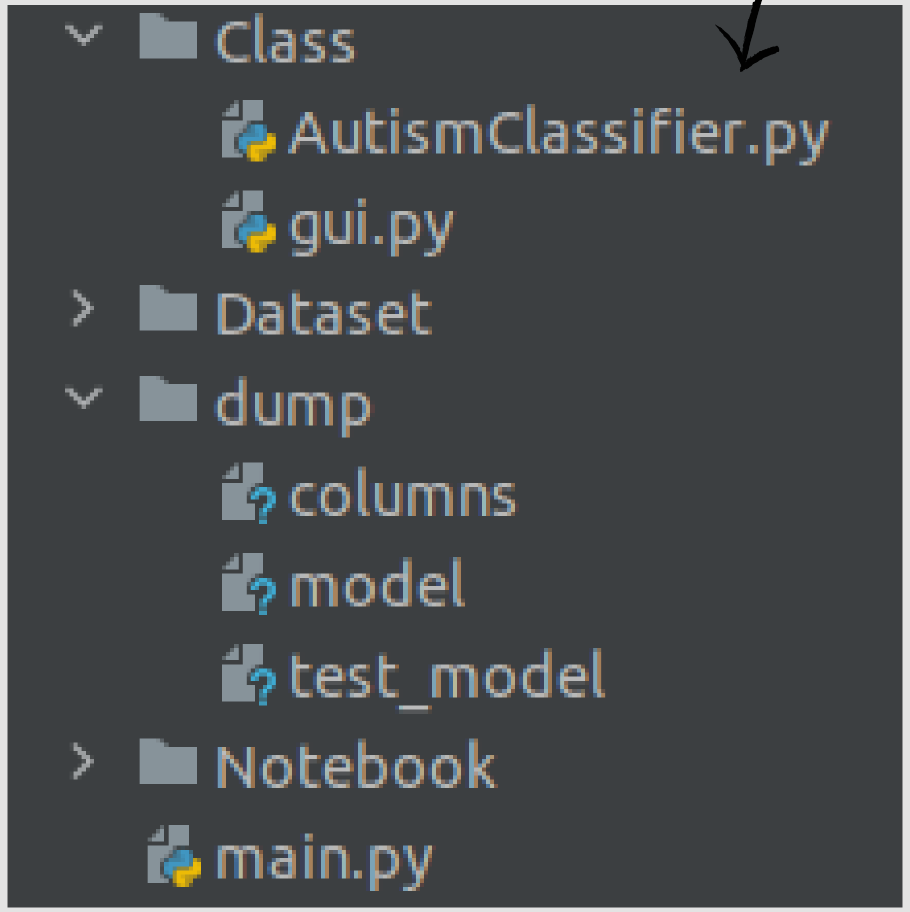
| A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 |
|---|---|---|---|---|---|---|---|---|---|
| ○ 0 | ○ 0 | ◉ 0 | ◉ 0 | ◉ 0 | ○ 0 | ○ 0 | ○ 0 | ○ 0 | ○ 0 |
| ◉ 1 | ◉ 1 | ○ 1 | ○ 1 | ○ 1 | ◉ 1 | ◉ 1 | ◉ 1 | ◉ 1 | ◉ 1 |

| Age | Gender | Country | Ethinicity | Jaundice | Family Member with PDD |
|---|---|---|---|---|---|
| 32 | F | Italy | White-European | No | Yes |

Classify

ASD: Yes

A class that represent the classifier and a class for the gui which use the method of the classifier



```
Class
    AutismClassifier.py
    gui.py
Dataset
dump
    columns
    model
    test_model
Notebook
main.py
```

Pickle module to serialize the trained model

# Thanks for the attention