Term End Project Milestone

Madhavi Ghanta

DSC, Bellevue University

DSC550-T301 Data Mining

Professor Brett Werner

11/15/2023

ABSTRACT:

This term-end project aims to evaluate the use of Data Mining techniques. The objective of this project is to illustrate the use of learned techniques to mine and analyze large datasets to discover useful knowledge. Text mining, unstructured data, social networks, and other types of unsupervised data mining methods for data science are included.

INTRODUCTION

Analyze data to predict the traits to detect Autistics disease among toddlers

Problem:

Autistic Spectrum Disorder (ASD) is a neurodevelopmental condition associated with significant healthcare costs, and early diagnosis can significantly reduce these. Unfortunately, waiting times for an ASD diagnosis are lengthy and procedures are not cost-effective. The economic impact of autism and the increase in the number of ASD cases across the world reveals an urgent need for the development of easily implemented and effective screening methods.

Solution:

Therefore, a time-efficient and accessible ASD screening is imminent to help health professionals and inform individuals whether they should pursue a formal clinical diagnosis. The rapid growth in the number of ASD cases worldwide necessitates datasets related to behavior traits.

Data Source: https://www.kaggle.com/fabdelja/autism-screening-for-toddlers?select=Toddler+Autism+dataset+July+2018.csv

Description of Dataset:

The dataset was developed by Dr Fadi Fayez Thabtah (fadifayez.com) using a mobile app called ASDTests (ASDtests.com) to screen autism in toddlers. we can use it to estimate the predictive power of machine learning techniques in detecting autistic traits.

Screen print showing the used raw dataset:

In [2]:
```
# 1.Load the data from the "Toddler Autism dataset July 2018.csv" file into a DataFrame.
addr1 = "D:/MS_DataScience/DSC550/Milestone-1/Toddler Autism dataset July 2018.csv"
df_todd = pd.read_csv(addr1)
df_todd.head()
```

Out[2]:

	Case_No	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Age_Mons	Qchat-10-Score	Sex	Ethnicity	Jaundice	Family_mem_with_ASD	Who completed the test	ASD_Traits
0	1	0	0	0	0	0	0	1	1	0	1	28	3	f	middle eastern	yes	no	family member	No
1	2	1	1	0	0	0	1	1	0	0	0	36	4	m	White European	yes	no	family member	Yes
2	3	1	0	0	0	0	0	1	1	0	1	36	4	m	middle eastern	yes	no	family member	Yes
3	4	1	1	1	1	1	1	1	1	1	1	24	10	m	Hispanic	no	no	family member	Yes
4	5	1	1	0	1	1	1	1	1	1	1	20	9	f	White European	no	yes	family member	Yes

Summary of the Process:

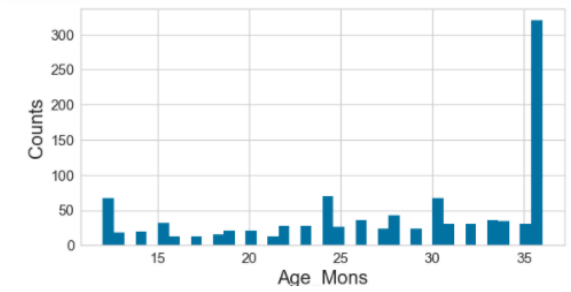
Please find below Steps involved in creating the ASD Screener Model.

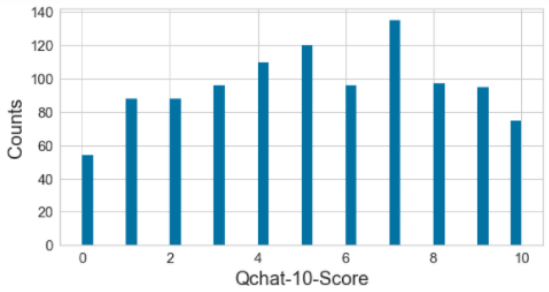
Begin Milestone 1: Completed the graphical analysis of data by creating a minimum of four graphs.

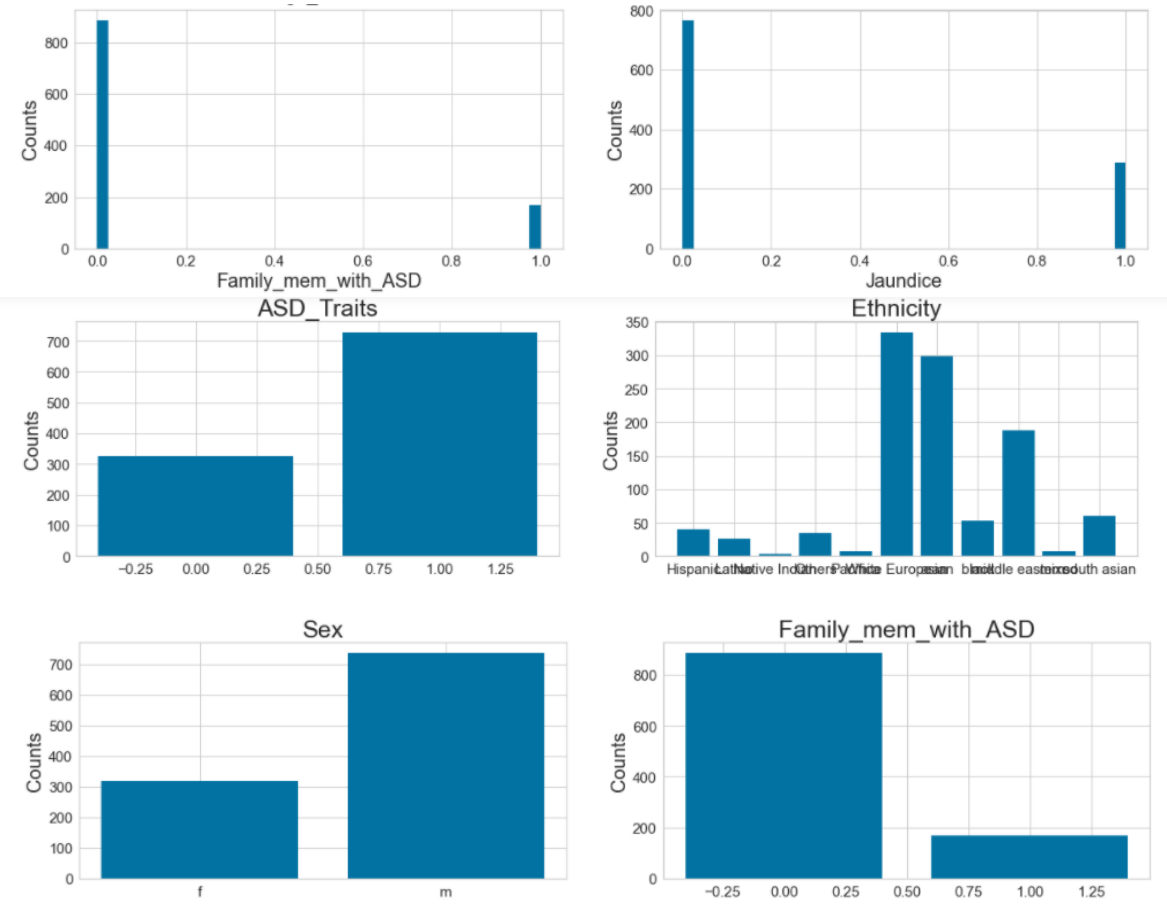
Summarized Data

	Sex	Ethnicity	Jaundice	Family_mem_with_ASD	\
count	1054	1054	1054	1054	
unique	2	11	2	2	
top	m	White European	no	no	
freq	735	334	766	884	

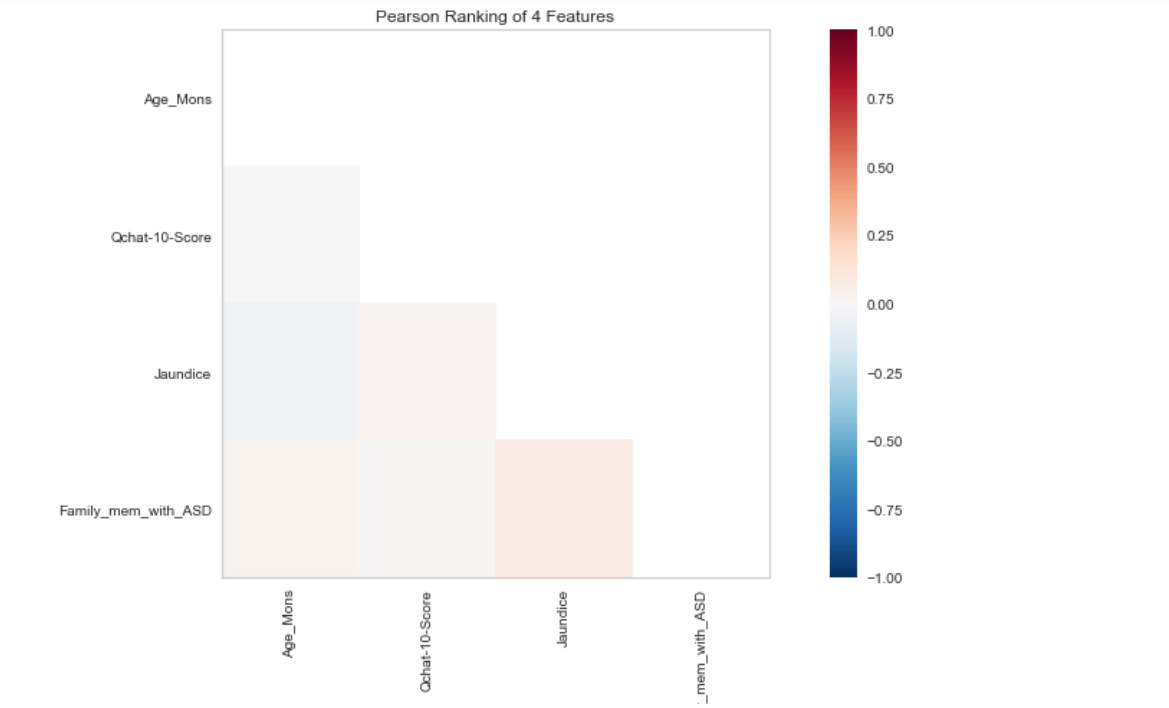
	Who completed the test	ASD_Traits
count	1054	1054
unique	5	2
top	family member	Yes
freq	1018	728





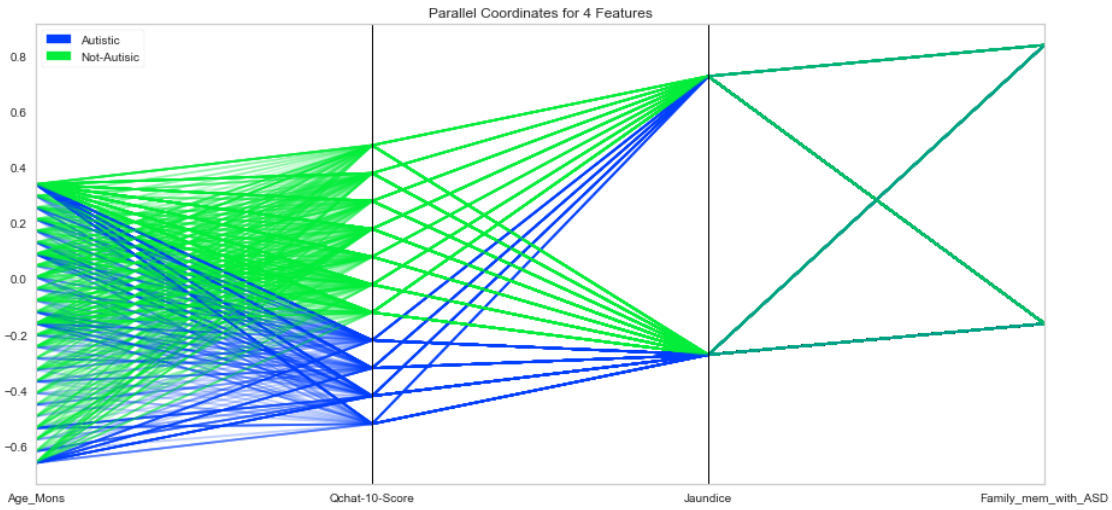


Determine the Pearson Ranking among features:



The correlation between the variables is low (1 or -1 is high positive or high negative, 0 is low or no correlation)
Here These results show there is positive correlation between 'ASD_Traits' & 'Qchat-10-Score', but there's not a high correlation among other variables.

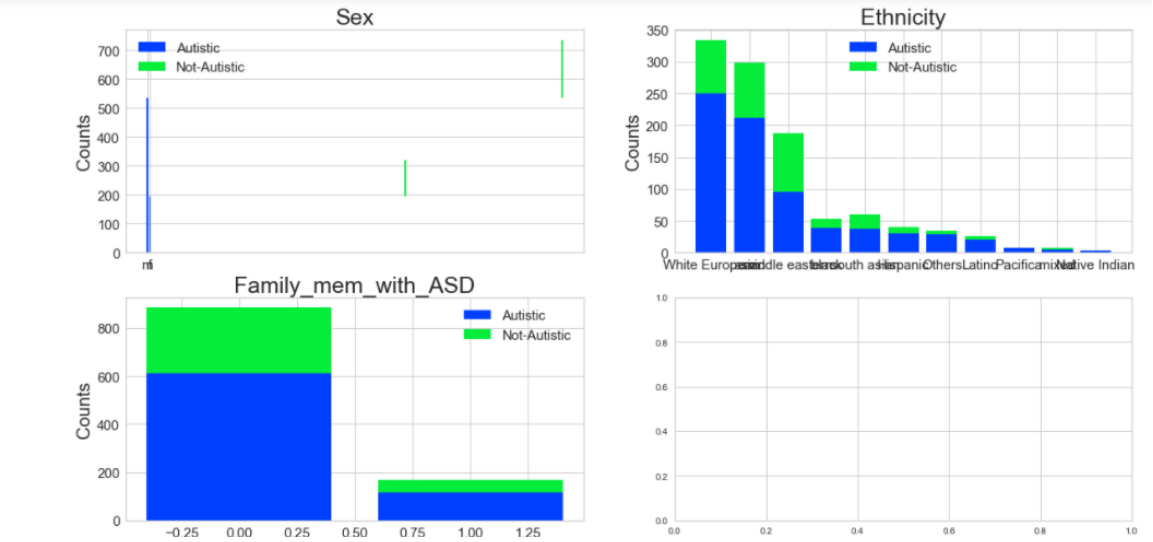
Finding out the Parallel correlation among the selected features:



Parallel coordinate for 4 features shows below information:
For Autistic patients : we have a seen relationship between age ,score and jaundice variables,however family men with ads feature is not having any relation with other features

For Non Autistics: we have seen relationship between all listed 4 features

Used Stack Bar Charts to compare toddlers who is having ASD & who didn't have ASD based on the other variables.



less females have ASD as compared to MEN, white europian is having more rate for impact sue to ASD family member history, as compared with Non-ASD history.

Milestone 2

1) Drop any features that are not useful for your model building. You should explain and justify why the feature dropped is not useful.

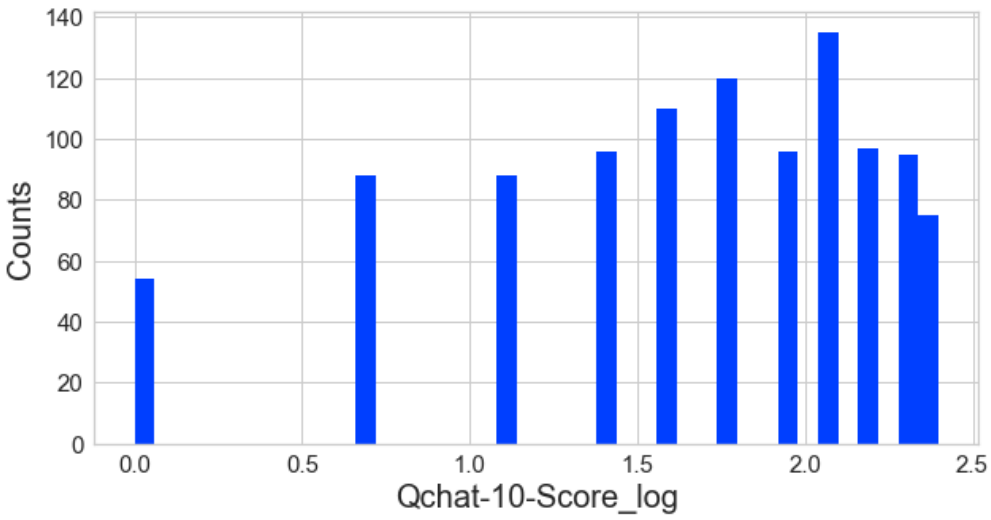
2) Address any missing data issues.

3) Build any new features that you need for your model, e.g., create dummy variables for categorical features if necessary. Explain your process at each step. You can use any methods/tools you think are most appropriate.

```
#fill the missing data with median value
# num_features = ['Age_Mons', 'Qchat-10-Score', 'Jaundice', 'Family_mem_with_ASD']
def fill_na_median(df_todd, inplace=True):
    return df_todd.fillna(df_todd.median(), inplace=inplace)

fill_na_median(df_todd['Qchat-10-Score'])

# check the result
print(df_todd['Qchat-10-Score'].describe())
```

```
count    1054.000000
mean        5.212524
std         2.907304
min         0.000000
25%         3.000000
50%         5.000000
75%         8.000000
max        10.000000
Name: Qchat-10-Score, dtype: float64
```

log transformation is showing high skewed values & and its counts in histogram :



Milestone 3, Build and evaluate at least one model.

Selected the features for the model

```
#create a whole features dataset that can be used for train and validation data splitting
# here we will combine the numerical features and the dummie features together
features_model = ['Jaundice', 'Age_Mons', 'Qchat-10-Score_log']
data_model_X = pd.concat([df_todd[features_model], data_cat_dummies], axis=1)
```

splitting the datasets into training & test datasets

```
# create a whole target dataset that can be used for train and validation data splitting
#data_model_y = df_todd.replace({'autism': {1: 'Autistic', 0: 'Not_Autistic'}})['ASD_Traits']
data_model_y = df_todd['ASD_Traits']
# separate data into training and validation and check the details of the datasets
# import packages
from sklearn.model_selection import train_test_split

# split the data
X_train, X_val, y_train, y_val = train_test_split(data_model_X, data_model_y, test_size =0.3, random_state=11)
```

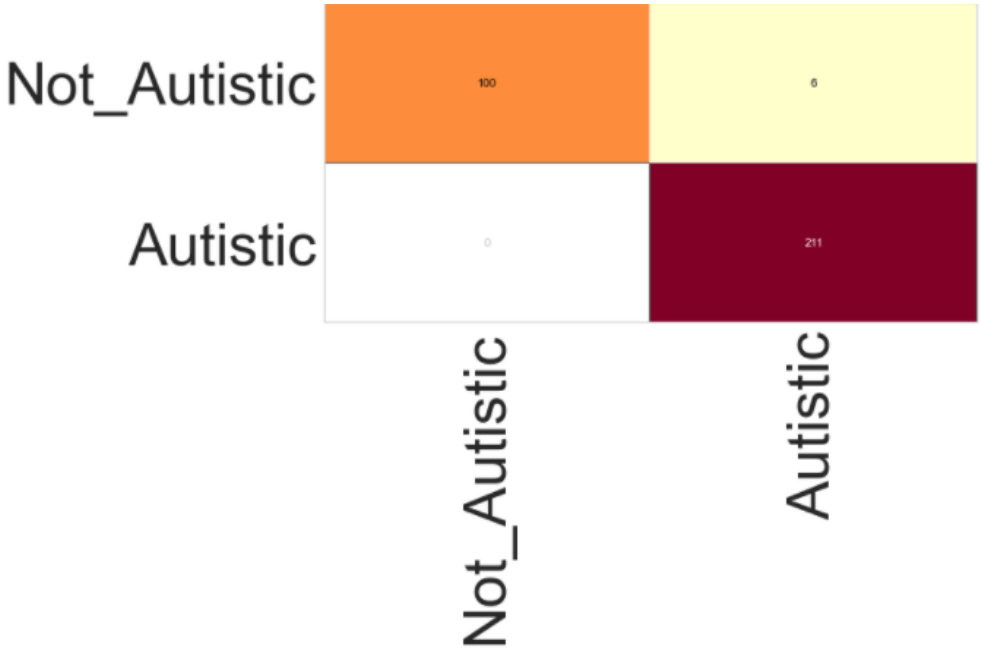
```
# number of samples in each set
print("No. of samples in training set: ", X_train.shape[0])
print("No. of samples in validation set:", X_val.shape[0])

# Autistic and not-autistic
print('\n')
print('No. of autistic and not-autistic in the training set:')
print(y_train.value_counts())

print('\n')
print('No. of autistic and not-autistic in the validation set:')
print(y_val.value_counts())
```

```
No. of samples in training set:  737
No. of samples in validation set: 317


No. of autistic and not-autistic in the training set:
1    517
0    220
Name: ASD_Traits, dtype: int64


No. of autistic and not-autistic in the validation set:
1    211
0    106
Name: ASD_Traits, dtype: int64
```

Here i have used Classification technique, since we are categorizing data into a given number of classes like 'Not_Autistic', 'Autistic'.

The main goal of a classification problem is to identify the category/class to which a new data will fall under.

Eval Metrics by using Confusion Matrix



There are two possible predicted classes: "Autistic" and "Not_Autistic". If we were predicting the presence of a disease, for example, "yes" would mean they have the disease, and "no" would mean they don't have the disease.

The classifier made a total of 317 predictions.

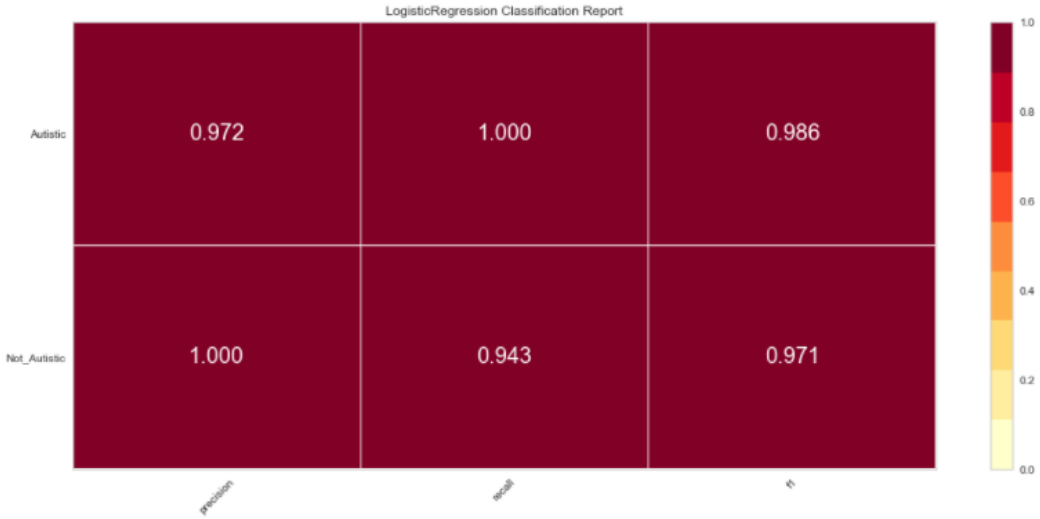
Out of those 317 cases, the classifier predicted "yes" 217 times, and "no" 100 times.

In reality, 211 patients in the sample have the disease, and 106 patients do not.

Accuracy: Overall, how often is the classifier correct?

(TP+TN)/total = (211+100)/317 = 0.98

Classification Report:

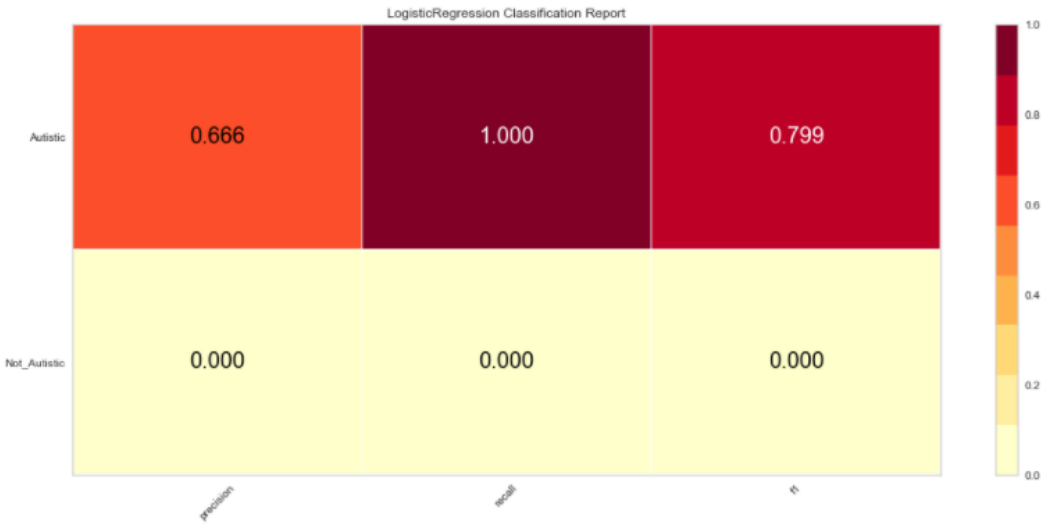


Found below reports from logistic regression classification report:

Precision – What percent of your predictions were correct? 97.2%
Recall – What percent of the positive cases did you catch? 100%
F1 score – What percent of positive predictions were correct? 98.6%

before concluding the final result, I have computed the classification reportes by using others features too

(['Jaundice', 'Age_Mons'])



Precision – What percent of your predictions were correct? 66.2%
Recall – What percent of the positive cases did you catch? 100%
F1 score – What percent of positive predictions were correct? .79%
data is biased it is not all combination autistic and not autistic datas
seems 'Jaundice', 'Age_Mons' are not good candidates as individuals predictors.

Conclusions/Recommendations:

The ASD Pre-Screening traits data is very useful, by using above-created model , health professionals can predict accurately the possibility of a toddler to be an ADS patient.

when we build the model by using ['Jaundice', 'Age_Mons', 'Qchat-10-Score_log'] features then the accuracy rate of the trained model is around 97% , looks like Qchat-10-Score_log score feature data is one of the major contributers of good accuracy.

Precision – What percent of your predictions were correct? 97.2%
Recall – What percent of the positive cases did you catch? 100%
F1 score – What percent of positive predictions were correct? 98.6%

However when i have dropped the Qchat-10-Score_log feature from the model and rebuild the model by using ['Jaundice', 'Age_Mons'] features, then its accuracy scores got decreased,

Precision – What percent of your predictions were correct? 66.2%
Recall – What percent of the positive cases did you catch? 100%
F1 score – What percent of positive predictions were correct? .79%

Recommendations: The toddler ASD prediction model is one of the useful tools, which is going to reduce the expenses related to unnecessary ADS diagnostics for all toddlers.
by using this model, the healthcare professional can accurately predict, which toddler requires ASD diagnostics and which doesn't require such type of diagnostics tests.
This Model will predict perfectly, only if it gets the accurate traits details from the toddler's parents.

References

Screening and Diagnosis of Autism Spectrum Disorder for Healthcare Providers | CDC

Chris Albon, Machine Learning with Python

Benjamin Bengfort, Applied Text Analysis with Python