EC 9560 – DATA MINING LAB 03

DARMILA.T

2020/E/027

SEMESTER 07

30th OCTOBER 2024

Data Preprocessing

```
0
      1 print(train_data[train_data['PCIAT_PCIAT_Total']<=30].sii.value_counts())</pre>
      2 print(train data[(train data['PCIAT-PCIAT Total']>30)
            & (train_data['PCIAT_PCIAT_Total']<50)].sii.value_counts())</pre>
      4 print(train_data[(train_data['PCIAT_PCIAT_Total']>=50)
            & (train_data['PCIAT_PCIAT_Total']<80)].sii.value_counts())</pre>
      6 print(train data[train_data['PCIAT_PCIAT_Total']>=80].sii.value counts())
      8 train_data.sii.value_counts()
<del>_</del>y sii
    0.0
            1594
    Name: count, dtype: int64
    1.0
            730
    Name: count, dtype: int64
    2.0
            378
    Name: count, dtype: int64
    3.0
            34
    Name: count, dtype: int64
          count
     sii
          1594
      0.0
      1.0
            730
      2.0
      3.0
             34
    dtype: int64
```

This sii values are correctly distributed within the PCIAT-PCIAT_Total score intervals.

To ensure the sii target variable distribution aligns with the expected categories of PCIAT_Total score ranges, verify that each range of scores corresponds correctly to its designated sii value.

Define the Expected Ranges: Based on the original criteria:

- PCIAT-PCIAT Total between 0-30 should have sii = 0.
- PCIAT-PCIAT Total between 31–49 should have sii = 1.
- PCIAT-PCIAT Total between 50–79 should have sii = 2.
- PCIAT-PCIAT Total between 80-100 should have sii = 3.

Removes any rows in the train DataFrame where the sii column has missing (NaN) values.

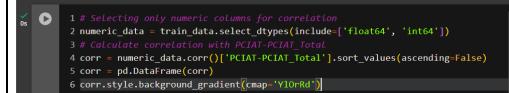
```
✓
2s
```

```
[20] 1 train_data = train_data.dropna(subset='sii')
```

Here, I clean the data by ensuring all rows in train have valid values in the sii column, which is likely important if sii is used as a target variable or feature in further analysis.

Correlations

Given the large number of available features, I decided to perform feature selection to assess its impact on the model. Here, I selected features with the strongest correlation to the PCIAT total score, discarding those with weaker correlations.



	PCIAT-PCIAT_Total
PCIAT-PCIAT_Total	1.000000
sii	0.899681
Physical-Height	0.420765
Basic_Demos-Age	0.409559
PreInt_EduHx-computerinternet_hoursday	0.374124
Physical-Weight	0.353048
Physical-Waist_Circumference	0.327013
FGC-FGC_CU	0.287494
BIA-BIA_BMI	0.248060
Physical-BMI	0.240858
SDS-SDS_Total_T	0.237718
SDS-SDS_Total_Raw	0.234432
PAQ_A-Season	0.219292
FGC-FGC_PU	0.196006
BIA-BIA_Frame_num	0.193631
FGC-FGC_GSD	0.160472
Physical-Systolic_BP	0.147081
FGC-FGC_GSND	0.146813
FGC-FGC_TL	0.136696
PAQ_C-Season	0.115316
BIA-BIA_FFMI	0.109694
BIA-BIA_FMI	0.085863
BIA-BIA_Activity_Level_num	0.084548

BIA-BIA_LST	0.075623
Physical-Diastolic_BP	0.069321
FGC-FGC_PU_Zone	0.056973
BIA-BIA_DEE	0.053094
BIA-BIA_SMM	0.052912
BIA-BIA_ICW	0.052593
BIA-BIA_TBW	0.043015
BIA-BIA_Fat	0.038548
BIA-BIA_BMR	0.037009
BIA-BIA_FFM	0.037009
BIA-BIA_ECW	0.035568
Preint_EduHx-Season	0.033788
Basic_Demos-Enroll_Season	0.029793
BIA-BIA_LDM	0.025885
SDS-Season	0.025112
Physical-Season	0.021411
FGC-FGC_GSD_Zone	0.006861
Fitness_Endurance-Time_Sec	-0.000373
CGAS-Season	-0.003344
FGC-FGC_CU_Zone	-0.004454
BIA-BIA_BMC	-0.008870
FGC-FGC_GSND_Zone	-0.009525
PAQ_C-PAQ_C_Total	-0.021943
PAQ_A-PAQ_A_Total	-0.026854

FGC-Season	-0.030890
FGC-FGC_TL_Zone	-0.037214
Physical-HeartRate	-0.037594
BIA-Season	-0.040922
Fitness_Endurance-Max_Stage	-0.041720
Fitness_Endurance-Time_Mins	-0.052376
CGAS-CGAS_Score	-0.070542
FGC-FGC_SRR	-0.077836
FGC-FGC_SRL	-0.091221
Basic_Demos-Sex	-0.093648
Fitness_Endurance-Season	-0.097788
FGC-FGC_SRR_Zone	-0.109682
FGC-FGC_SRL_Zone	-0.148850

```
1 selection = corr[(corr['PCIAT-PCIAT_Total']>.1) | (corr['PCIAT-PCIAT_Total']<-.1)]
      2 selection = [val for val in selection.index]
      3 selection.remove('PCIAT-PCIAT_Total')
      4 selection.remove('sii')
      5 selection.remove('Physical-BMI')
       6 selection.remove('SDS-SDS_Total_Raw')
[25] 1 selection
 → ['Physical-Height',
      'Basic_Demos-Age',
      'PreInt_EduHx-computerinternet_hoursday',
      'Physical-Weight',
      'Physical-Waist_Circumference',
      'FGC-FGC_CU',
'BIA-BIA_BMI'
      'SDS-SDS_Total_T',
      'PAQ_A-Season',
      'FGC-FGC_PU',
      'BIA-BIA_Frame_num',
      'FGC-FGC_GSD',
      'Physical-Systolic_BP',
      'FGC-FGC_GSND',
      'FGC-FGC_TL',
      'PAQ_C-Season',
      'BIA-BIA_FFMI',
       'FGC-FGC_SRR_Zone',
      'FGC-FGC_SRL_Zone']
```

Here, I select a subset of features that have a moderate correlation (either positive or negative) with PCIAT_PCIAT_Total, while excluding certain specified columns. The goal is to narrow down the list of features to those most relevant for analysis or modeling, based on their correlation with PCIAT-PCIAT_Total, and to exclude target or potentially redundant features.

Find the missing values in the train_data

```
1 null = train_data.isna().sum().sort_values(ascending = False).head(46)
2 null = pd.DataFrame(null)
3 null = null.rename(columns= {0:'Missing'})
4 null.style.background_gradient(cmap='YlOrRd')
```

	Missing	FGC-FGC_PU_Zone	861
PAQ_A-PAQ_A_Total	2373	FGC-FGC_SRL_Zone	859
Physical-Waist_Circumference	2253		
Fitness_Endurance-Time_Sec	2008	FGC-FGC_SRR_Zone	857
Fitness_Endurance-Time_Mins	2008	FGC-FGC_CU_Zone	852
Fitness_Endurance-Max_Stage	2005	FGC-FGC_TL_Zone	851
FGC-FGC_GSD_Zone	1872		
FGC-FGC_GSND_Zone	1872	FGC-FGC_PU	827
FGC-FGC_GSD	1865	FGC-FGC_SRL	825
FGC-FGC_GSND	1864	FGC-FGC_SRR	823
PAQ_C-PAQ_C_Total	1296		020
BIA-BIA_Activity_Level_num	923	FGC-FGC_CU	817
BIA-BIA_BMC	923	FGC-FGC_TL	817
BIA-BIA_BMI	923	CGAS-CGAS_Score	394
BIA-BIA_TBW	923	CGAS-CGAS_SCORE	394
BIA-BIA_DEE	923	Physical-Systolic_BP	258
BIA-BIA_ECW	923	Physical-Diastolic_BP	258
BIA-BIA_FFM	923	Division Head Date	250
BIA-BIA_FFMI	923	Physical-HeartRate	250
BIA-BIA_FMI	923	SDS-SDS_Total_T	211
BIA-BIA_Fat	923	Physical-BMI	209
BIA-BIA_Frame_num	923		
BIA-BIA_ICW	923	SDS-SDS_Total_Raw	209
BIA-BIA_LDM	923	Physical-Height	206
BIA-BIA_LST	923	Physical-Weight	164
BIA-BIA_SMM	923		
BIA-BIA_BMR	923	PreInt_EduHx-computerinternet_hoursday	82

Here, Identifies the column that more than one value is missing in the data frame.

Then, I check is any selected features have more than half value is missing, If I Found remove that in from the selection list features.

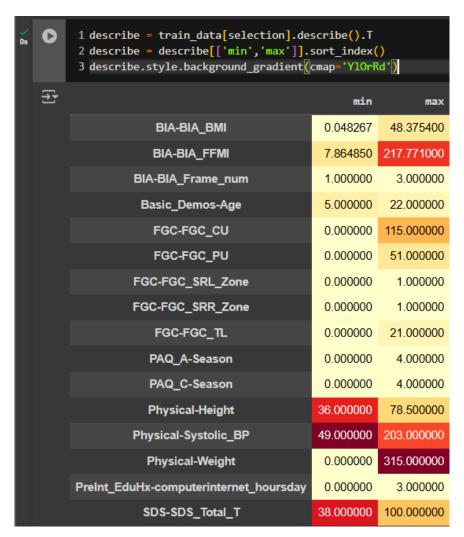
```
1 selection = [i for i in selection if i not in half_missing]
O
      2 selection
→ ['Physical-Height',
     'Basic_Demos-Age',
      'PreInt_EduHx-computerinternet_hoursday',
      'Physical-Weight',
      'FGC-FGC_CU',
      'BIA-BIA_BMI',
      'SDS-SDS_Total_T',
      'PAQ_A-Season',
      'FGC-FGC_PU',
      'BIA-BIA_Frame_num',
      'Physical-Systolic_BP',
      'FGC-FGC_TL',
      'PAQ_C-Season',
      'BIA-BIA_FFMI',
      'FGC-FGC_SRR_Zone',
      'FGC-FGC_SRL_Zone']
```

I have now identified 16 selected features based on two criteria:

- a) their correlation with the target variable and
- b) the presence of relatively few missing values.

Here, I want to create a model that makes accurate predictions by using only the most useful information while avoiding distractions from irrelevant data.

In the following section, I display the minimum and maximum values of the selected features in the training data.



Now, I scaled scale the selected features in your train_data DataFrame using MinMaxScaler.



Split the dataset

```
[35] 1 X = train_data[selection]
2 test = test_data[selection]
3 y = train_data.sii
```