Marmara University Faculty of Engineering

CSE4062 – Data Science, Spring2020

Group7

"DRIVER DROWSINESS DETECTION"

Delivery #3 - Report

CSE	Mahmut AKTAŞ	aktasmahmut97@gmail.com	150115010
CSE	Mustafa Abdullah HAKKOZ	mustafa.hakkoz@gmail.com	150117509
ME	Ozan Berke YABAR	ozanberkeyabar@gmail.com	150416822
MME	Ece HARPUTLU	harputlue@gmail.com	150515038
EE	Nurettin ABACI	abacinurettin@gmail.com	150715035

Submitted to: Assoc. Prof. Murat Can Ganiz

- 16.05.2020 -

EXPLORE YOUR DATA part-2

1. Introduction

In previous iteration, **23 columns** are concluded after processing videos and constructing database. To reduce the number of features some preprocessing and feature selection steps are included.

2. Preprocessing and Feature Selection

- 4 of them; “**Subject”**, “**Factors List”**, “**Facial Actions”** and **“Frame_no”** are selected as hierarchical indexes and dropped from columns of dataframe (23-4=19 columns left).
- By selecting only values **“Face_Detected”==1** , we get rid off some missing values (-1) due to multipe face and non face situations. After that **“Face_Detected”** column is also droppped from dataframe (18 columns left).
- Also first two frames of the videos in test set (16 videos in total) were also missing and marked as -1 while processing video so, we also dropped these rows from database by using **“DROWSINESS”** column.
- **“PERCLOS”** values of first 150 frames of all videos were also missing. So we changed them to 0 from -1 by using **“reserved_for_calibration”** column. So there’s no missing value left in all of fields in the dataframe. After that, **“reserved_for_calibration”** is also dropped from dataframe (17 columns left).
- Also there is abnormal 0’s in left and right eye features. I.e when subject turn to his right, his right eye is not detected and all of it’s values become 0. So, to detect this scenario, the condition of **(“RIGHT_EAR” ==0 and “LEFT_EAR” !=0)** is searched in dataframe and **“LEFT_EAR”** value is selected and copied in to newly defined **“EAR”** column instead of **“AVG_EAR”.** Reverse scenario is also checked and **“RIGHT_EAR”** is used to update **“EAR”** this time. For rest of values **“AVG_EAR”** is used as as usual. Then three columns **“LEFT_EAR”, “RIGHT_EAR”** and **“AVG_EAR”** are dropped from database. These steps are repeated for EC, SOP and LEB values, since they are eye features also. After that 4*3 = 12 columns dropped from dataframe and new 4 columns are added (17-12+4=9 columns left).
- After previous step, **“MOE”** column is updated by using new **“EAR”** column, since it was using ear values to scale **“MAR”** column.
- So after preprocessing we dropped all of the columns that won’t be needed anymore, and concluded 8 columns **"EAR"**, **"MAR"**, **"MOE"**, **"EC"**, **"LEB"**, **"SOP"**, **"PERCLOS"**, **"CLOSENESS"** as features and 1 column **“DROWSINESS”** as class label.

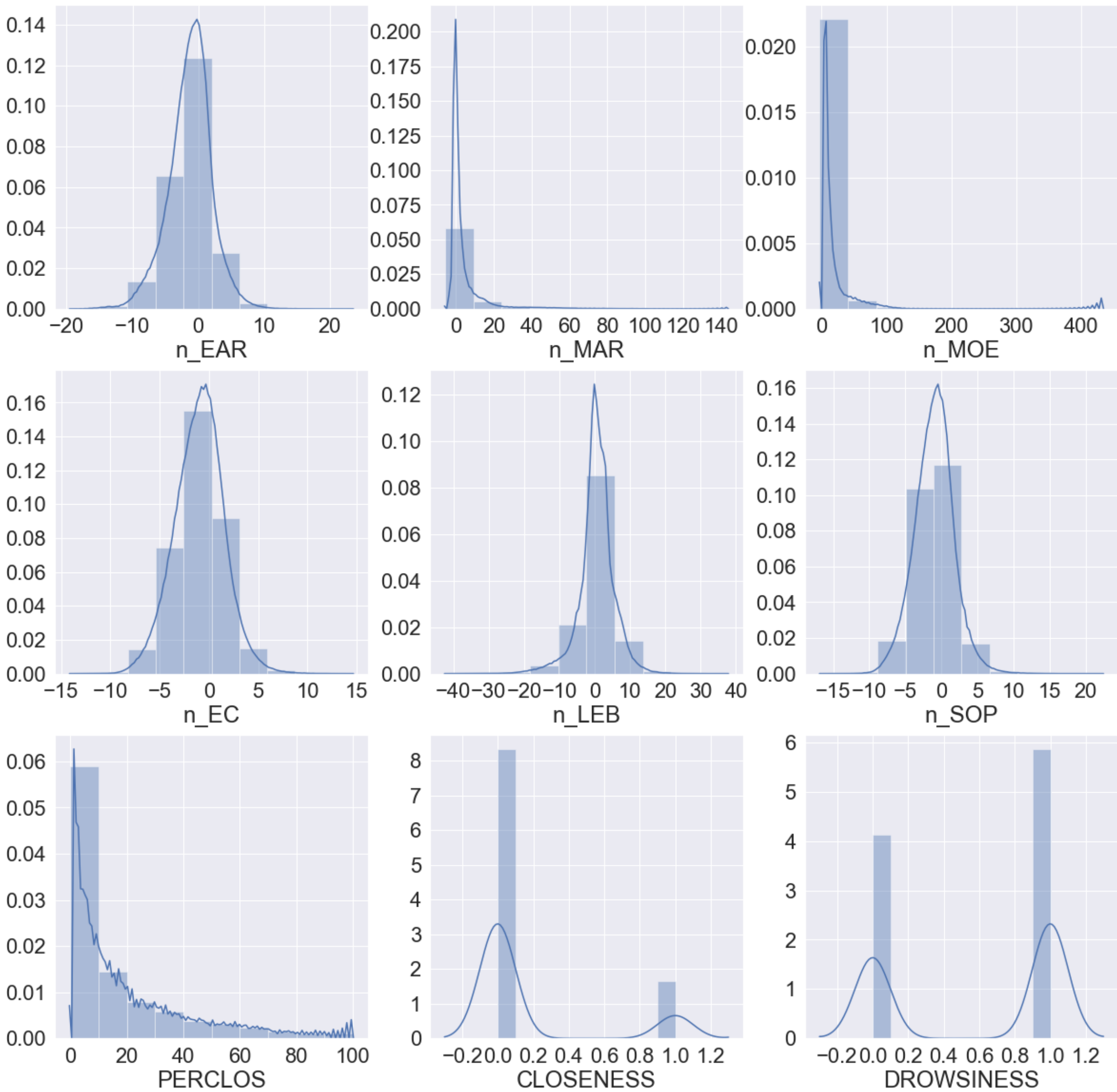
3. Normalization

In normalization step, we calculated mean and standart deviation of first 90 frames of alert videos and using these values we normalize relevant subject’s all videos. So this way, all of the values in the dataframe become adaptive to each subject.

We did not normalize **“PERCLOS”**, **“CLOSENESS”**and **“DROWSINESS”**. Other than this, we recalculate **“CLOSENESS”** column with using normalized thresholds for each subject. So in the end, we have 6 normalized columns **"n_EAR"**, **"n_MAR"**, **"n_MOE"**, **"n_EC"**, **"n_LEB"** and **"n_SOP"** and 3 other columns **“PERCLOS”**, **“CLOSENESS”**and **“DROWSINESS”**.

4. Distribution Charts

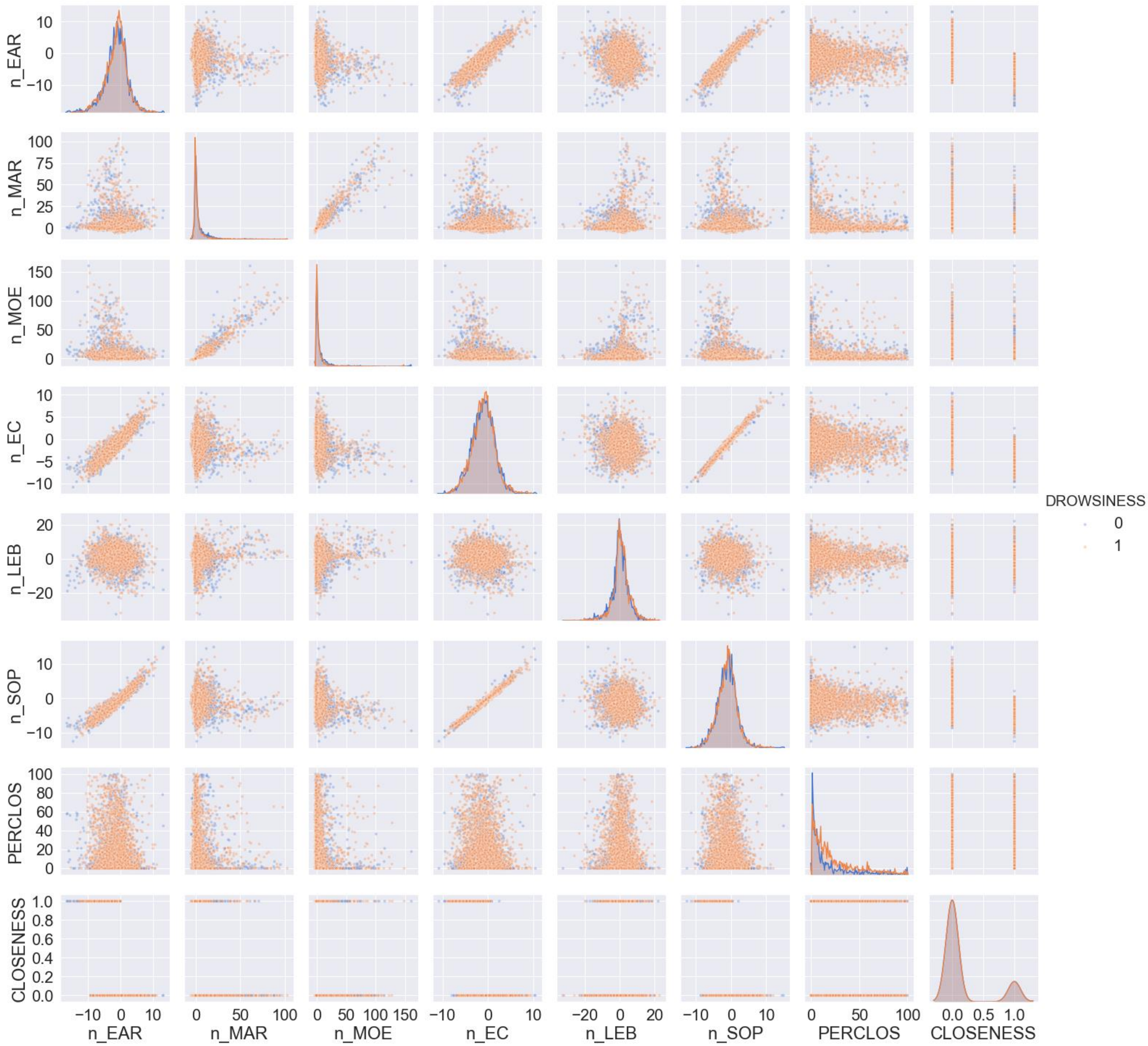
Distributions of values in each column are plotted below:



We can see that eye features like **n_EAR**, **n_EC**, **n_LEB** and **n_SOP** have normal distribution and mouth features **n_MAR** and **n_MOE** have skewed distribution. **CLOSENESS** and **DROWSINESS** are binary values and their class distributions don’t seem correlated here.

5. Scatter Plot Matrix

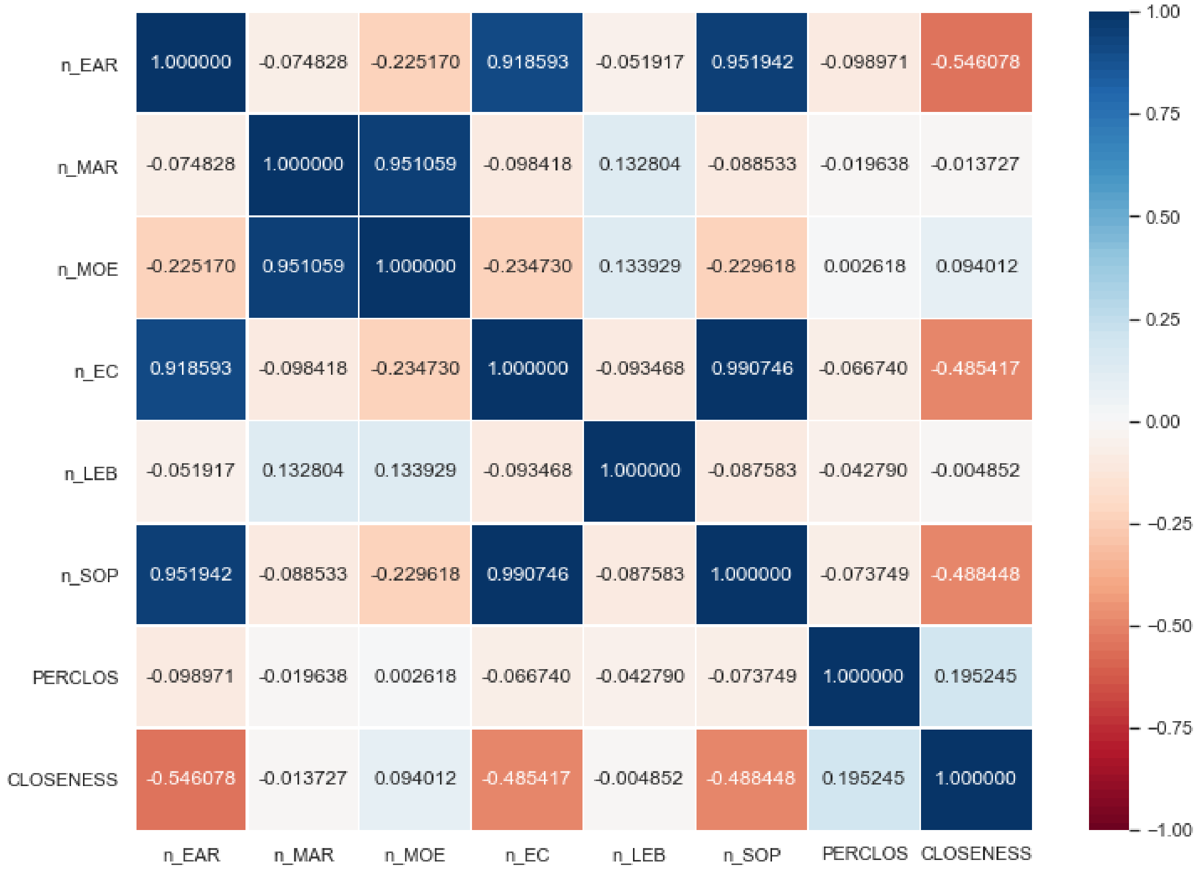
Scatter plots of each pair of features are plotted below:



In all cells in the matrix, classes are overlapped to each other even though we sampled (n=5000) our data to increase readability. Probablity distributions on diagonal cells also support this claim. So it seems like there's no way to seperate classes accordingly, if we choose only two features as subset of all of the features.

6. Correlation Matrix

To see the correlation of each pair of features, we plotted pearson correlation matrix below:



Darker cells show high correlation (blues are positive correlations, reds are negtive correlations) and lighter cells shows low correlations (white is 0).

3 of eye features **n_EAR**, **n_EC** and **n_SOP** are highly correlated to each other. On the other hand **n_LEB** isn't directly related to eye since it defines level of eyebrows. And mouth features **MAR** and **MOE** don't seem to correleated to eye features but they are again highly correlated to each other.

7. Conclusion

In this iteration of the project we reduced our features to 8 from 23 after using most of the features in preprocess and normalization steps. By adding class column to it, in its final form our dataframe has 9 columns.

We also used distribution plots, scatter plot matrix and correlation matrix to analyze our features and tried to understand their nature. It's concluded that, if we choose a feature subset which contains only 2 features, classes will not be seperated accordingly. Thus, in the next iteration we will try to increase the number of features in subsets and try them in some classifier models.