Driver Drowsiness Detection

Mustafa Abdullah HAKKOZ
Computer Engineering
Marmara University
Istanbul, Turkey
mustafa.hakkoz@gmail.com

Mahmut AKTAŞ
Computer Engineering
Marmara University
Istanbul, Turkey
aktasmahmut97@gmail.com

Ozan Berke YABAR
Mechanical Enginnering
Marmara University
Istanbul, Turkey
ozanberkeyabar@gmail.com

Ece HARPUTLU
Metallurgical and Materials Engineering
Marmara University
Istanbul, Turkey
harputlue@gmail.com

Nurettin ABACI
Electrical and Electronic Engineering
Marmara University
Istanbul, Turkey
abacinurettin@gmail.com

***Abstract*—Driver drowsiness is a massive issue threatening road safety, for this reason development of a robust and practical driver drowsiness detection system is a crucial step. In order to reduce the number of drowsiness-caused accidents, various researches have been conducted with the aim of finding practical and non-invasive drowsiness detection systems by using behavioral measuring techniques. Many of the previous works on behavioral measuring techniques have mainly focused on the analysis of eye closure and blinking of the driver. New facial features from mouth, eye pupil, eyebrows etc. through the single-frame method. We have implemented the algorithms using the NTHU-DDD dataset. Experimental results verified the effectiveness of the proposed method compared to the state-of-art results.**

Keywords

Drowsiness detection, facial expression, NTHU-DDD, kNN, Decision Tree, Naïve Bayes, AGNES, Mutual Information, DBSCAN.

I. INTRODUCTION

Driver fatigue is a persistent danger for drivers and road safety which is the dominant reason for road accidents. According to the KGM (Karayolları Genel Müdürlüğü), last year 3704 people have died in road accidents in Turkey and 92.65% of accidents were caused by driver defects [1]. Driver fatigue, more specifically drowsiness, is one of the major reasons of the traffic accidents [2]. Most of drivers who had an accident before due to the sleepiness (94%), mentioned themselves to be alone in the vehicle [3]. According to a survey between drivers, 24% of subjects were having a sleep attack regularly, during their driving experiences even if they don't cause any accidents [3]. On this basis, the development of an accurate and practical drowsiness detection system is a necessity.

In the last decade, most of major car companies developed their own practical and robust drowsiness detection systems. But the recent quick rise of machine learning requires to be updated with newer technologies. So, for this project, we want to build our own visual drowsiness detection system and to find ways of implementing machine learning classifiers that we learned in the class lectures.

In this paper, we propose a system that checks a driver's facial behavior, mainly eyes and mouth and detect the drowsiness status of the driver. In order to achieve this goal, firstly we detected the driver's face and extract the facial features. After extracting these features, we trained a model with drowsy non-drowsy test data using the NTHU-DDD dataset. By using this model, we did classification experiments and got accuracy results.

In terms of experimental dataset, National Tsing Hua University released a public driver drowsiness dataset in ACCV2016 competition [4]. The public NTHU-DDD dataset is composed of 360 training videos and 20 testing videos, involving different scene condition (i.e., glasses, no-glasses or sunglasses under variant illumination), where all videos were captured by an infrared camera. We exclude the subjects with sunglasses. In addition, some drowsiness related information was labeled frame-by-frame in NTHU-DDD, such as eye state, mouth state and head state. The subject in this dataset have acted the state of drowsiness. In order to measure the accuracy of the proposed approach, we test the several classifiers such as Decision Tree, Naïve Bayes, kNN on NTHU-DDD and report the experiment results in Section IV.

II. RELATED WORK

2.1 Drowsiness Detection Methods

There are various ways to determine the drowsiness level:

- Physiological methods: In these methods, systems are detecting drowsiness with collecting data through various electrophysiological sensors like Electrocardiogram (ECG) to listen electrical activity of human heart [5], Electroencephalogram (EEG) to record the activity of human brain [6] and Electrooculogram (EOG) to observe human eye [7]. They are reliable and precise methods but placing sensors on the driver's body and collecting data intrusively reduces the driver's comfort.
- Vehicle-based methods: These methods include observing steering wheel movements [8] and lane deviation [9]. They are non-invasive methods but highly correlated with driver's skill so they are subjective methods.
- Behavioral methods: This category observes behavioral movements of drivers with cameras and predicts his drowsiness level. Since it is non-intrusive and produces high accuracies due to the novel developments in

machine learning and computer vision domains, their popularity increases among researchers and commercial solutions. Some of the behavioral signals that may give a cue on driver's fatigue are: Head position [10, 11], yawning [12], blinks [13], or other facial actions like state of eyebrow, lip or jaw [14].

2.2 Feature Extraction Methods

After the detection of face and facial members, it's necessary to produce some meaningful numerical values to predict the drowsiness of the subject. Some methods that can be used for it are listed below.

- Eye Closure Analysis: There are some metrics that can be used to determine eye state, which are PERCLOS (Percentage of Eye Closure) [15, 16, 17] and EAR (Eye Aspect Ratio) [18].
- Yawning Analysis: Yawning is one of the first indicators that comes to mind when detecting drowsiness, so some metrics which are tracking mouth openness are used in literature. One example of them is MAR (Mouth Aspect Ratio) [19, 20, 21].
- Facial Expression Analysis: It's also possible to use some other facial features like wrinkles by detecting them with Laplacian Filter [14].
- Head Position Analysis: Another algorithm, POSIT is used in drowsiness detection [22] in the literature along with some other Head Pose Estimation techniques [10, 19].

2.3 Classifying Methods

After extracting features from raw data and constructing training datasets, there are also many choices of classifiers to predict the drowsiness level. Some of them are listed below:

- Basic Thresholding: It's rare to see using thresholds for drowsiness detection like [23] but there are some other usages of thresholds in blink detection [18] or nodding detection [8].
- Conventional Machine Learning Tools: While it's possible to use simpler approaches like Logistic Regression, Decision Tree, k-NN (K-Nearest Neighbors), NB (Naïve-Bayes) and get satisfactory results [14, 24, 20, 21], most researchers likely prefer more complex tools i.e. SVM (Support Vector Machine) [7, 19, 10, 11, 25, 16, 18, 5], Random Forest [19, 10, 6], HMM (Hidden Markov Model) [26, 9, 4, 27, 22], AdaBoost (Adaptive Boosting) [28, 27, 11] and even XGBoost (Extreme Gradient Boosting).

2.4 State-of-art Results

The primary challenge in DDD literature is each of researches using different datasets [29] and the absence of a standard, large and realistic datasets that can be used as benchmarks. There is an example of the effort in 2017, about comparing different works on different datasets by using meta-analysis approach which indicates CNN as a most successful classifier against SVM and HMM [29]. Yet, it's still not enough to predicate state-of-art results because some of the databases are not open to public access and open ones are mostly consist of actor subjects or non-realistic environments and these are the main reasons shadowing accountability of works in the literature.

But recently there is one novel work by Yaocong Hu, Mingqi Lu, Chao Xiez, and Xiaobo Lu [30], which aims to fill the gap of benchmark datasets and uses NTHU-DDD dataset with a baseline method on it. They propose 3D Conditional GAN and Two-level Attention Bi-LSTM model and declare their results on NTHU-DDD dataset as in the Table 1 and Table 2. Hence, these can be used for future researchers to compare their works.

Table 1. the ablation analysis of the 3dcgan network on NTHU-DDD testing dataset.

Model	DR(%)	FAR(%)	AR(%)
3DDIS	74.1	29.0	72.6
3DDIS-A	74.7	27.7	73.6
3DDIS-B	77.6	23.9	76.9
3DGAN	76.4	25.6	75.4
3DcGAN-A	76.9	24.5	76.2
3DcGAN-B	81.9	18.6	81.7
3DcGAN	82.3	16.5	82.8

Quantitative result in Table 1 shows that the 3DcGAN network achieves the total accuracy rate of 82.8% with detection rate 82.3% and false alarm rate 16.5% amongst other types of generative models.

Then they improved these results with using LSTM-based techniques. The model inputs consecutive short-term drowsiness-related representation, captures temporal dependencies and outputs the long-term drowsiness score of each frame.

Table 2. The ablation analysis of the several LSTM-based techniques on NTHU-DDD testing dataset.

Model	DR(%)	FAR(%)	AR(%)
3DcGAN+LSTM	83.8	15.6	84.1
3DcGAN+BiLSTM	85.5	15.0	85.3
3DcGAN-ALSTM-A	86.2	14.1	86.0
3DcGAN-ALSTM-B	86.0	14.9	85.6
3DcGAN-TLALSTM	86.9	14.0	86.5
3DcGAN-BiALSTM-A	86.5	13.8	86.3
3DcGAN-BiALSTM-B	86.6	14.4	86.1
3DDIS+TLABiLSTM	82.1	18.8	81.7
3DcGAN+TLABiLSTM	87.5	13.3	87.1

In addition to the above article, there is another work that is highly related to our proposed method as mentioned in the introduction [21]. This work done by Grant Zhong, Rui Ying, He Wang, Aurangzaib Siddiqui and Gaurav Choudhary. We took the most benefit from this work in this paper. They have the same workflow with this paper. We used the features like EAR, MAR, MOE (Mouth Over Eye), EC (Eye Circularity) from this article. Apart from this paper, they used UTA-RLDD dataset for their work.

III. Approach

The whole pipeline of the project comprises eight steps as they can be seen in Fig. 3. "Processing Videos" phase to reading and processing videos, "Feature Extraction" phase to construct ad-hoc features, "Preprocessing" phase to handling missing values and normalization, two "Explorotary Data

Analysis" steps for data statistics and visual analysis, "Predictive Analysis" for classification and finally "Descriptive Analysis" to implement clustering models along with evaluation of models to measure the success of the both classification and clustering models proposed.

A. Processing Videos

Implementation of the project starts with "Processing Videos" phase which includes steps;

1. Reading video frames, either can be done from a dataset or a camera in a real-time manner. For this step, opencv-python [31], a python version of OpenCV library, is used.

2. Detecting faces with Dlib's get_frontal_face_detector [32] method. It uses a pre-trained "Histogram of Oriented Gradients + Linear SVM" pipeline for face detection. There's also another CNN-based method in Dlib but it isn't suitable for real-time purposes.

3. Predicting facial landmarks with Dlib's shape_predictor [33] method which is an implementation of the paper Kazemi and Sullivan (2014) [34]. This method also uses a pre-trained model of an ensemble of regression trees and predicts 68 facial landmarks which can be seen in Fig. 1.

All features will be used in later phases, are extracted from these positional landmarks.

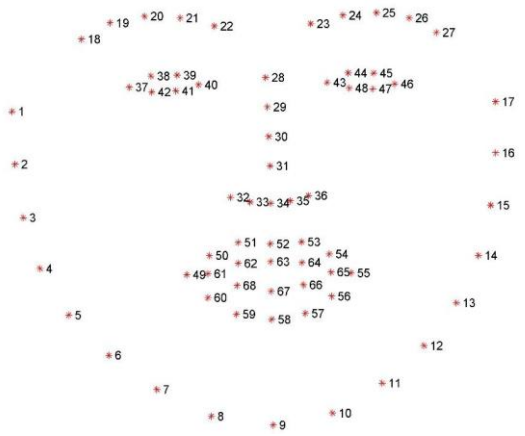


Fig. 2. 68 facial landmark coordinates of Dlib's shape_predictor method.

B. Feature Extraction

After the processing video phase, implementation continues with the feature extraction phase. For the frame-based model we proposed, without detecting any facial action (blink, yawning etc.), all features are computed for every frame and all of this information will be used for classification.

- **Eye Aspect Ratio (EAR):** It is proposed by the paper [18] and investigates eye closeness using a simple mathematical formula for real-time purposes.

$$EAR(i) = \frac{\|p_{38} - p_{42}\| + \|p_{39} - p_{41}\|}{2\|p_{37} - p_{40}\|} \quad (1)$$

- **Mouth aspect ratio (MAR):** This formula resembles EAR (1), in the context of using 68 facial landmarks (see Figure-12). It uses inner landmarks of the mouth (61, ..., 68) and calculates a ratio just like EAR. Therefore, it can be useful for detecting yawning behavior [35].

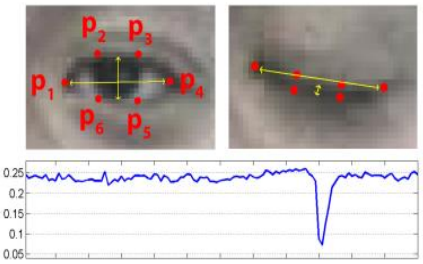


Fig. 1. An original figure from the 2016 EAR paper [35]. Eye landmarks are used in the calculation of EAR with open/closed eye scenarios.

$$MAR(i) = \frac{\|p_{63} - p_{67}\|}{\|p_{61} - p_{65}\|} \quad (2)$$

- **Eye Circularity (EC):** It's a measure like EAR but it puts greater emphasis on the pupil area. [21]

$$EC(i) = \frac{4 \times \pi \times Pupil\ Area}{(Eye\ Perimeter)^2} \quad (3)$$

$$Pupil\ Area = \left(\frac{\|p_{38} - p_{41}\|}{2}\right)^2 \times \pi \quad (4)$$

$$Eye\ Perimeter = \|p_{37} - p_{38}\| + \|p_{38} - p_{39}\| + \|p_{39} - p_{40}\| + \|p_{40} - p_{41}\| + \|p_{41} - p_{42}\| + \|p_{42} - p_{37}\| \quad (5)$$

- **Mouth over Eye (MOE):** It's an additional feature which can be interpreted as true drowsiness, since some facial actions like smiling and talking may produce some fake yawning MAR values.

$$MOE(i) = \frac{MAR(i)}{EAR(i)} \quad (6)$$

- **PERCLOS**: Indicates the frequency of closed eyes up until that moment. [15] We are planning to use it with a small-time window like t = 90 seconds.

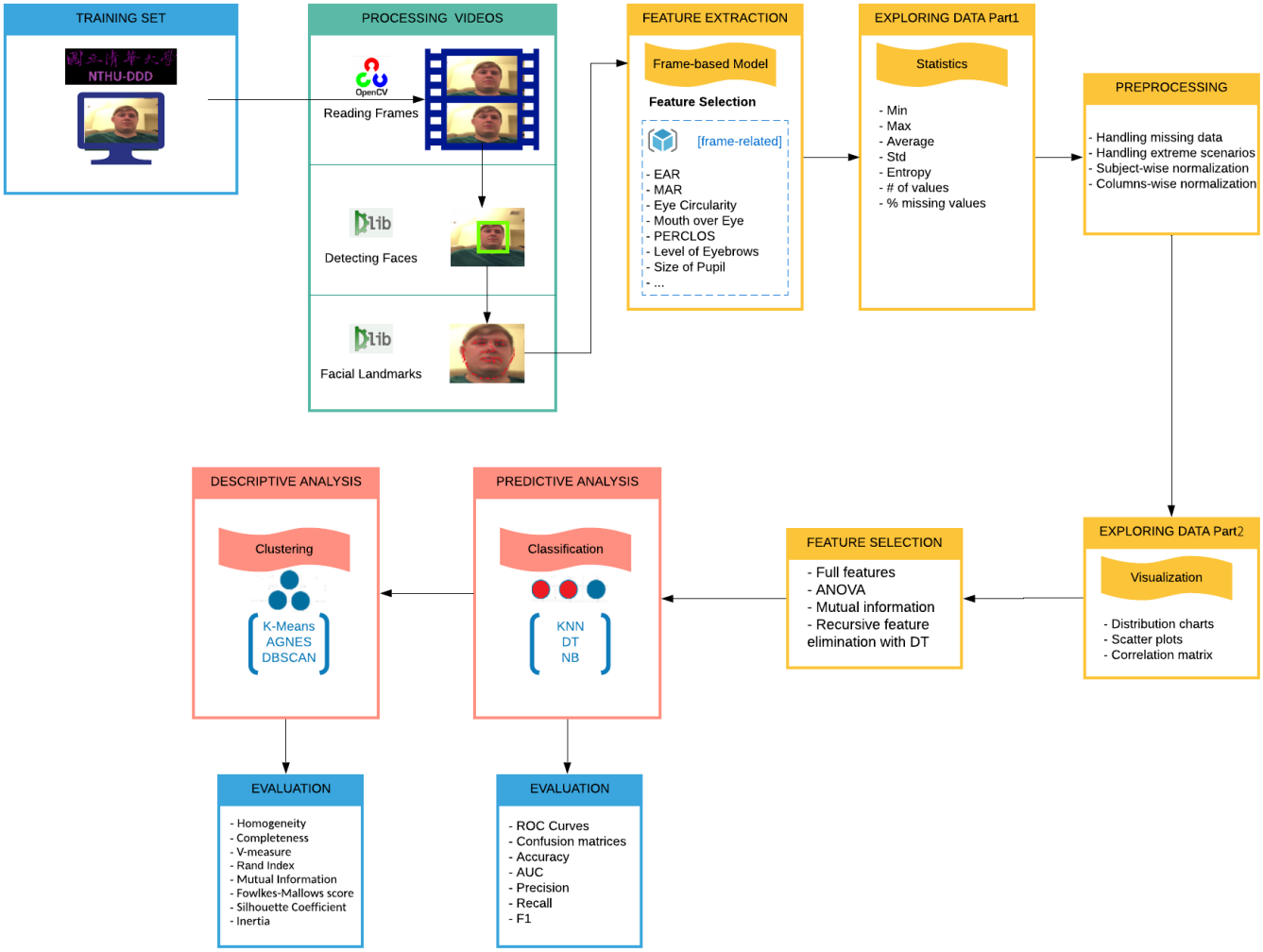
$$PERCLOS(i) = \frac{count\ of\ frames\ when\ the\ eyes\ are\ closed}{total\ count\ of\ frames\ up\ until\ that\ moment} \quad (7)$$

- **Level of Eyebrows (LEB):** Average distance between first two of inner points of eyebrows and inner corner of an eye. Other points of eyes are ignored due to their moving nature.

$$LEB(i) = \frac{\|p_{21} - p_{40}\| + \|p_{22} - p_{40}\|}{2} \quad (8)$$

- **Size of Pupil (SOP):** This is not a directly related to drowsiness but fluctuations of pupil size may be related to the fatigue of a subject [36]. So, the defined formula below measures the ratio of pupil diameter and eye width.

Fig. 3. Conceptual diagram of the pipeline.



- $SOP(i) = \frac{\|p_{38} - p_{41}\|}{\|p_{37} - p_{40}\|}$ (9)

C. *Exploring Data part-1*

After feature extraction phase, we explore the data with some statistical tools like average, missing values, standard deviation and entropy (i.e. Appendix 1, Table 1).

D. *Preprocessing*

In the preprocessing phase, we handle the missing values by dropping them, in such scenarios as no face detected, multiple face detected, time-lapses in videos and annotation mismatching.

Also, two normalization steps are included. First is subject-wise normalization, To do it, we calculated mean and standard 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.

$$normalized\ feature_{n,m} = \frac{feature_{n,m} - \mu_{n,m}}{\sigma_{n,m}} \quad (10)$$

Where, $\mu_{n,m}$ and $\sigma_{n,m}$ are mean and standard deviation of the feature n of the subject m.

And for columns-wise normalization, mean and standard deviation is calculated for each feature n by using the similar formula in (10).

E. *Exploring Data part-2*

After preprocessing phase, we continue to explore the dataset with using some plots such as distribution charts, scatter plots and correlation matrix. They can be seen in Fig. 1, 2, 3 in Appendix 2.

F. *Feature Selection*

For feature selection phase, we tried four different feature subsets (i.e. Fig. 1 in Appendix 3):

1. Full sets of features,
2. Feature selection with ANOVA,
3. Feature selection with Mutual Information,
4. Recursive feature elimination with Decision Tree Classifier.

G. *Predictive Analysis*

We chose 5 different classifying models to experiment with:

1. K-NN with 5 neighbors,
2. K-NN with 25 neighbors,
3. Decision Tree Classifier with gini,
4. Decision Tree Classifier with entropy,

5. Gaussian Naïve-Bayes.

After implementing the models with four different subsets of features in total of 20 experiments, we evaluate them with using 5 different metrics: Accuracy, precision, recall, F1 and roc-auc score. And finally, significant models are determined with using t-test (i.e. Appendix 4).

H. Descriptive Analysis

Before clustering experiments, we apply Principal Component Analysis (PCA) to our data so number of features are reduced to 9 to 3 for visualization purposes.

We choose 3 different clustering models: K-means, AGNES and DBSCAN. K-means runs without any problem on our dataset which has a size of (610K x 3) but other two models give memory errors. So, we use resample method to choose random (and stratified) 10K sample to work on manageable data.

For hyperparameter tuning of the clustering models, we used elbow technique for K-means, silhouette plots for AGNES and silhouette scores with grid-search for DBSCAN.

To evaluate these 3 clustering models, we used several metrics:

- Estimated number of clusters
- Number of instances in cluster
- Estimated number of noise points
- Standard deviation of clusters
- **Homogeneity**: For perfect clustering, each cluster contains only members of a single class.
- **Completeness**: For perfect clustering, all members of a given class are assigned to the same cluster.
- **V-measure**: Harmonic mean of Homogeneity and Completeness.
- **Adjusted Rand Index**: Given the knowledge of the ground truth class, the adjusted Rand index is a function that measures the similarity of the two assignments, ignoring permutations and with chance normalization.
- **Adjusted Mutual Information**: Given the knowledge of the ground truth class, the Mutual Information is a function that measures the agreement of the two assignments, ignoring permutations.
- **Fowlkes-Mallows score**: Geometric mean of precision and recall.
- **Silhouette Coefficient**: If the ground truth labels are not known, the Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample by (b - a) / max(a, b).
- **Inertia (Sum of Squared Error)**: Inertia measures the internal cluster sum of squares (sum of squares is the sum of all residuals). Inertia is utilized to measure how related clusters are amongst themselves, the lower the inertia score the better. However, it is important to note that inertia heavily relies on the assumption that the clusters are convex (of spherical shape). DBSCAN and AGNES does not necessarily divide data into spherical clusters, therefore inertia is not a good metric to use for evaluating DBSCAN and AGNES models (which is why I did not include inertia in the code above). Inertia is more often used in other clustering methods, such as K-means clustering. Thus, inertia_ attribute is only provided for K-means in SKLearn.

IV. Experiment Setup

While preprocessing the data, after handling missing values, an additional optimization step is added to increase the success of the models. There are abnormal 0's in left and right eye features. When subject turn to his right, his right eye is not detected and all of its 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. These steps are repeated for **EC**, **SOP** and **LEB** values, since they are eye features also. **MOE** column is updated by using new **EAR** column, since it was using ear values to scale **MAR** column (6). So after preprocessing we dropped all of the columns that won't be needed anymore, and concluded eight columns **EAR**, **MAR**, **MOE**, **EC**, **LEB**, **SOP**, **PERCLOS**, **CLOSENESS** as features and one column **DROWSINESS** as class label.

After both of the normalization steps (subject-wise and column-wise) explained in (10), data is handed into classification experiments. To implement feature selection without causing bias on test-set, we concluded to use 5x2 cross-validation technique [37]. It's recommended to use an outer CV (i.e. 5 folds) to repeat the whole pipeline to produce an array of results which will be necessary to run t-test on. In every fold, we can do feature elimination without effecting test-set along with inner-cv to run standard experiments such as, feature elimination or hyperparameter tuning.

In our case we used 5 folds for outer CV and we did train-test splitting there. Then in every fold we first run our 3 feature elimination methods. For the fourth method (RFE), we needed another CV to run so we choose 2 folds for inner CV. We also transformed corresponding outer test-sets by using selected features. After that, we run our 5 different classifiers and run predictions on corresponding test-sets. And finally, we evaluate models by using 5 metrics we determined above. In total, we run 5 x 4 = 20 different experiments in each of 5 folds.

V. Experimental Results And Discussion

A. Classification Experiments

For t-test, we chose our top two experiments according to roc-auc metric in Table 3:

1. KNN-25 with full features: 80.71
2. KNN-5 with full features: 79.31

Then run scipy's ttest_ind method on their CV scores. We compared them by using 5 different metrics evaluation metrics (Accuracy, precision, recall, AUC, F1). According to precision metric there's no significant difference between them but for all other metrics, there is. So, we can conclude that KNN-25 with full features is our best experiment (i.e. Appendix 4).

We also observed that, in general our feature elimination methods didn't provide better results than full features unless Naïve-Bayes classifier. In the case of NB, feature elimination methods based Mutual Information and ANOVA resulted in higher accuracy.

Table 3. Comparison of results of clustering models.

#	1	2	3
Clustring Experiment	**K-Means**	**AGNES**	**DBSCAN**
# of Clusters	4	3	1
Number of Instances in Clusters	{0: 287, 1: 8260, 2: 1269, 3: 184, 'avg': 2500.0}	{0: 77, 1: 450, 2: 9473, 'avg': 3333.33}	{-1: 4, 0: 9996, 'avg': 5000.0}
Std. Dev. Of Cluster1	[11.85, 6.39, 7.24]	[16.05, 8.52, 5.79]	[3.20, 14.36, 8.04]
Std. Dev. Of Cluster2	[3.01, 4.65, 4.45]	[20.91, 6.56, 7.37]	[17.18, 5.13, 4.86]
Std. Dev. Of Cluster3	[6.84, 7.04, 6.22]	[6.61, 5.02, 4.69]	
Std. Dev. Of Cluster4	[20.26, 7.16, 6.05]		
SSE	746778	-	-
NMI	0,019	0,022	0
Silhouette Val.	0,516	0,767	0,877
RI	-0,002	-0,015	0
Est. # of Noise Points	0	0	4
Homogenity	0,018	0,015	0
Completeness	0,02	0,043	0,06
V-Measure	0,019	0,022	0,001
Fowlkes-Mallows Score	0,599	0,675	0,717

When we compare to our best model (KNN25-FULL, accuracy:0.74, recall: 0.80) to state-of-art results of generative models (Table 1), it is close to top model (recall: 82.3%) even though its less complex model comparing to the ones in the table. But when it comes to LSTM-models in Table 2, our results can't match them. Because sequential models like LSTM, takes the relationships between frames in to consideration, not just the sperate frames unlike our models.

B. Clustering Experiments

We compared best of 3 models by using several evaluation metrics in Table 4. Only metrics don't require ground-truth labels are Silhouette Coefficient and inertia. For Silhouette Coefficient, DBSCAN takes ahead, then comes AGNES and followed by K-MEAN. Its results are correlated with Fowlkes-Mallows score although they require different inputs.

But from the perspective of Homogeneity and Completeness principles, AGNES performs best. Following is K-MEANS and DBSCAN. The results are similar for Mutual Information.

And finally, for Rand Index, AGNES performs best again. Followings are K-MEANS and DBSCAN.

One important note, for best parameters of DBSCAN, the model produces only 1 cluster so most of metrics generates null values for it. Thus, it's better not taking DBSCAN into consideration in the evaluation phase.

Table 4. Five models with four different subsets of features

#	Experiment	Accuracy	Precision	Recall	AUC	F1
		Avg	Avg	Avg	Avg	Avg
1	KNN5 FULL	0,7378	0,7698	0,7931	0,7284	0,7813
2	KNN5 ANOVA	0,6879	0,7197	0,7636	0,6709	0,741
3	KNN5 MI	0,7029	0,7108	0,7542	0,6595	0,7319
4	KNN5 RFE	0,7286	0,7592	0,7813	0,7149	0,7701
5	CART-GINI FULL	0,7288	0,7698	0,7678	0,7211	0,7688
6	CART-GINI ANOV	0,6554	0,7074	0,7042	0,6455	0,7058
7	CART MI	0,6918	0,7132	0,7129	0,6531	0,713
8	CART RFE	0,7289	0,7693	0,767	0,7204	0,7681
9	NB FULL	0,5585	0,7341	0,3854	0,5937	0,5055
10	NB ANOVA	0,6326	0,6794	0,7031	0,6163	0,6911
11	NB MI	0,6039	0,6984	0,5635	0,6092	0,6237
12	NB RFE	0,5202	0,7222	0,267	0,5606	0,3899
13	KNN25 FULL	0,7429	0,7668	0,8071	0,7295	0,7864
14	KNN 25 ANOVA	0,7012	0,7209	0,7986	0,6801	0,7578
15	KNN25 MI	0,7159	0,7148	0,7901	0,6715	0,7506
16	KNN25 RFE	0,7351	0,7569	0,7963	0,7168	0,7761
17	CART-ENT FULL	0,7315	0,7711	0,7706	0,7231	0,7708
18	CART-ENT ANOVA	0,6568	0,7072	0,7046	0,6454	0,7059
19	CART-ENT MI	0,6939	0,7147	0,7142	0,6549	0,7144
20	CART-ENT RFE	0,7305	0,7712	0,7696	0,7229	0,7704

VI. Conclusion

For this project, we processed one of the common datasets used in driver drowsiness detection literature, NTHU-DDD and extracted eight hand-made features to feed several conventional machine learning algorithms such as KNN, NB and Decision Tree.

Rather than running complex classification algorithms, we mainly focused on developing advanced preprocessing techniques specialized just for NTHU dataset and to achieve that, we analyzed data with statistical tools and plots along with feature selection techniques like ANOVA, Mutual Information and Recursive Feature Elimination. We also introduced descriptive analysis with clustering algorithm such as K-means, AGNES and DBSCAN.

Consequently, we achieved acceptable scores (recall: 0.80) with our KNN model comparing to some of the state-of-art results explained in Table 1 (top recall: 0.82).

References

[1] Karayolları Genel Müdürlüğü, Trafik Kazaları Özeti. [Online]. Available: https://www.kgm.gov.tr/Sayfalar/KGM/SiteTr/Trafik/TrafikKazalariOzeti.aspx (Date of Access: 20 / 06 /2020)

[2] Abdulkerim Sönmez, "Ağır Vasıta Sürücüleri'nin Çalışma Koşulları ve Trafik Kazaları, Uzun Mesafe Yük ve Yolcu Taşımacılığı Yapan Sürücüler Üzerine Bir Çalışma", T.C. Emniyet Genel Müdürlüğü Trafik Hizmetleri Başkanlığı Trafik Araştırma Merkezi Müdürlüğü, 1999.

[3] Mahir Gökdağ, Fatih İrfan Baş, "The Effect of Fatigue and Sleepiness upon DriverBehaviors", Erzincan University Journal of Science and Technology, 2019.

[4] C.-H. Weng, Y.-H. Lai, and S.-H. Lai, "Driver drowsiness detection via a hierarchical temporal deep belief network," in Computer Vision – ACCV 2016 Workshops, C.-S. Chen, J. Lu, and K.-K. Ma, Eds. Cham: Springer International Publishing, 2017, pp. 117–133.

[5] W. Han, Y. Yang, G. Bin Huang, O. Sourina, F. Klanner, and C. Denk, "Driver Drowsiness Detection Based on Novel Eye Openness Recognition Method and Unsupervised Feature Learning," Proc. - 2015 IEEE Int. Conf. Syst. Man, Cybern. SMC 2015, no. September, pp. 1470–1475, 2016.

[6] M. Patel, S. K. L. Lal, D. Kavanagh, and P. Rossiter, "Applying neural network analysis on heart rate variability data to assess driver fatigue", Expert Syst. Appl., 38(6):7235–7242, June 2011

[7] S. Hu and G. Zheng, "Driver drowsiness detection with eyelid related parameters by Support Vector Machine", Expert Syst. Appl., 36(4):7651–7658, May 2009.

[8] D. McDonald, C. Schwarz, J. D. Lee, and T. L. Brown, "Real Time Detection of Drowsiness Related Lane Departures Using Steering Wheel

Angle," Proc. Hum. Factors Ergon. Soc. Annu. Meet., vol. 56, no. 1, pp. 2201–2205, 2012.
[9] Zer Customs, Volvo Driver Alert Control and Lane Departure Warning. [Online]. Available: http://www.zercustoms.com/news/Volvo-Driver-Alert-Control-and-Lane-Departure-Warning.html (Date of Access 20 / 06 /2020)
[10] A. Mittal, K. Kumar, S. Dhamija, and M. Kaur, "Head movement based driver drowsiness detection: A review of state-of-art techniques," Proc. 2nd IEEE Int. Conf. Eng. Technol. ICETECH 2016, pp. 903–908, 2016.
[11] A. Čolić, O. Marques, B. Furht, Driver Drowsiness Detection Systems and Solutions. Cham Heidelberg New York Dordrecht London: Springer, 2014.
[12] Centers for Disease Control and Prevention, Drowsy Driving: Asleep at the Wheel. [Online]. Available: https://www.cdc.gov/features/dsdrowsydriving/index.html (Date of Access 20 / 06 /2020)
[13] R. Ghoddoosian, M. Galib and V. Athitsos, A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.
[14] T. Nakamura, A. Maejima, and S. Morishima, "Detection of driver's drowsy facial expression," Proc. - 2nd IAPR Asian Conf. Pattern Recognition, ACPR 2013, pp. 749–753, 2013.
[15] Wierwille, Walter W. et al. "RESEARCH ON VEHICLE-BASED DRIVER STATUS/PERFORMANCE MONITORING; DEVELOPMENT, VALIDATION, AND REFINEMENT OF ALGORITHMS FOR DETECTION OF DRIVER DROWSINESS. FINAL REPORT", 1994.
[16] A. Dasgupta, A. George, S. L. Happy and A. Routray, "A Vision-Based System for Monitoring the Loss of Attention in Automotive Drivers", in: IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, pp. 1825-1838, Dec. 2013.
[17] A. Liu, Z. Li, L. Wang and Y. Zhao, "A practical driver fatigue detection algorithm based on eye state," 2010 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), Shanghai, 2010, pp. 235-238.
[18] T. Soukupová and Jan Cech. "Real-Time Eye Blink Detection using Facial Landmarks", 2016.
[19] A. Ramos, J. Erandio, E. Enteria, N. Carmen, L. Enriquez and D. Mangilaya, "Driver Drowsiness Detection Based on Eye Movement and Yawning Using Facial Landmark Analysis", International journal of simulation: systems, science & technology, 0.5013/IJSSST.a.20.S2.37., 2019.
[20] Q. Cheng, W. Wang, X. Jiang, S. Hou and Y. Qin, "Assessment of Driver Mental Fatigue Using Facial Landmarks", in IEEE Access, vol. 7, pp. 150423-150434, 2019.
[21] Zhong, G., Ying, R., Wang, H., Siddiqui, A., & Choudhary, G., Drowsiness Detection with Machine Learning. [Online]. Available: https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a (Date of Access 20 / 06 /2020)
[22] I. H. Choi, C. H. Jeong, and Y. G. Kim, "Tracking a driver's face against extreme head poses and inference of drowsiness using a hidden Markov model", Appl. Sci., vol. 6, no. 5, 2016.
[23] R. Prem Kumar, M. Sangeeth, K.S. Vaidhyanathan, A. Pandian. "TRAFFIC SIGN AND DROWSINESS DETECTION USING OPEN-CV", International Research Journal of Engineering and Technology (IRJET) vol. 06 issue 03, 2019.
[24] T. Ojala, M. Pietikainen and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions", Proceedings of 12th International Conference on Pattern Recognition, Jerusalem, Israel, 1994, pp. 582-585 vol.1.
[25] Naz, Saima et al. "Driver Fatigue Detection using Mean Intensity, SVM, and SIFT." IJIMAI 5: 86-93, 2019.
[26] R. Fu, H. Wang, W. Zhao, "Dynamic Driver Fatigue Detection Using Hidden Markov Model in Real Driving Condition", Expert Systems with Applications, vol.63, pp.397-411, 2016.
[27] E. Tadesse, W. Sheng and M. Liu, "Driver drowsiness detection through HMM based dynamic modeling," 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014, pp. 4003-4008.
[28] P. Viola and M. Jones, "Robust real-time object detection", International Journal of Computer Vision, 2001
[29] M. Ngxande, J-R. Tapamo, M. Burke, Driver Drowsiness Detection Using Behavioral Measures and Machine Learning Techniques: A Review of State-Of-Art Techniques, in: Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), 2017.
[30] Hu, Y., Lu, M., Xie, C., & Lu, X, Driver Drowsiness Recognition via 3D Conditional GAN and Two-level Attention Bi-LSTM. IEEE Transactions on Circuits and Systems for Video Technology, (2019)
[31] PyPi, opencv-python. [Online]. Available: https://pypi.org/project/opencv-python/ (Date of Access 20 / 04 /2020)
[32] Dlib, Classes. [Online]. Available: http://dlib.net/python/index.html#dlib.get_frontal_face_detector (Date of Access 20 / 04 /2020)
[33] Dlib, Classes. [Online]. Available: http://dlib.net/python/index.html#dlib.shape_predictor (Date of Access 20 / 04 /2020)
[34] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees", 2014 IEEE Conference on Computer Vision and Pattern Recognition 1867-1874, 2014.
[35] A. Ramos, J. Erandio, E. Enteria, N. Carmen, L. Enriquez and D. Mangilaya, "Driver Drowsiness Detection Based on Eye Movement and Yawning Using Facial Landmark Analysis", International journal of simulation: systems, science & technology, 10.5013/IJSSST.a.20.S2.37., 2019.
[36] B. Wilhelm, A. Widmann, W, Durst, C. Heine and G. Otto, "Objective and quantitative analysis of daytime sleepiness in physicians after night duties", International journal of psychophysiology: official journal of the International Organization of Psychophysiology. 72. 307-13. 10.1016/j.ijpsycho.2009.01.008., 2009.
[37] T. G. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms," in Neural Computation, vol. 10, no. 7, pp. 1895-1923, 1 Oct. 1998, doi: 10.1162/089976698300017197.

Appendix – 1: Exploratory Data Analysis part-1

Table 1. Statistical overview of the dataset.

#	Feature Name	Description	Type	Min.	Max.	Avg.	Std. Dev.	Entropy	# of Values	Missing Values %
1	Frame_No	Frame Number	Numerical	1	17460	2140,444	2935,9576	18,48939	681517	0 %
2	Face_Detected	# of Detected Faces	Numerical	0	2	0.9022	0.2976	19.2295	681517	0 %
3	AVG_EAR	Average Eye Aspect Ratio	Numerical	0.2251	0.9142	0.2836	0.0735	19.1795	681517	9.81 %
4	LEFT_EAR	Left Eye Aspect Ratio	Numerical	0.0192	0.7692	0.2577	0.0651	19.1817	681517	9.81 %
5	RIGHT_EAR	Right Eye Aspect Ratio	Numerical	0.0	1.3333	0.3095	0.0869	19.1717	681517	9.81 %
6	MAR	Mouth Aspect Ratio	Numerical	0.0	1.8804	0.1238	0.1802	18.2539	681517	9.81 %
7	MOE	Mouth Over Eye	Numerical	0.0	19.9546	0.4761	0.7393	18.1879	681517	9.81 %
8	PERCLOS	Percentage of Eye Closure	Numerical	0.0	100.0	16.8288	21.5928	18.0942	681517	16.33 %
9	AVG_LEB	Level of Eyebrows	Numerical	5.9732	67.6823	31.0206	6.9819	19.1927	681517	9.81 %
10	LEFT_LEB	Left Eye Brow Level	Numerical	6.4051	65.1325	31.4313	6.6788	19.1965	681517	9.81 %
11	RIGHT_LEB	Right Eye Brow Level	Numerical	4.1231	73.9968	30.6099	7.5747	19.1851	681517	9.81 %
12	AVG_SOP	Size of Pupils	Numerical	0.2324	1.0521	0.4686	0.0595	19.2177	681517	9.81 %
13	LEFT_SOP	Left eye pupil size	Numerical	0.215	1.4907	0.4826	0.0721	19.2136	681517	9.81 %
14	RIGHT_SOP	Right eye pupil eye	Numerical	0.2165	0.9235	0.4547	0.0556	19.2185	681571	9.81 %
15	AVG_EC	Eye Circularity	Numerical	0.132	0.9681	0.472	0.0883	19.204	681517	9.81 %
16	LEFT_EC	Left Eye Circularity	Numerical	0.1137	0.9846	0.4575	0.0888	19.2019	681517	9.81 %
17	RIGHT_EC	Right Eye Circularity	Numerical	0.113	1.225	0.4865	0.1036	19.197	681517	9.81 %
18	CLOSENESS	Eye Closure Status	Binary	0	1	0.1647	0.3709	16.6276	681517	9.81 %
19	DROWSINESS	Drowsiness Status	Binary	0	1	0.5717	0.4948	18.5718	681517	0.0046 %
20	Subject	Subject Number	Nominal	005	026	-	-	4.4521	22	-
21	Factors List	Factors List	Nominal	Night glasses	No glasses	-	-	1.9322	4	-
22	Facial Actions	Facial Actions	Nominal	Yawning	Nonsleepy Combination	-	-	2.3048	4	-
23	Reserved for Calibration	Reserved For Calibration	Nominal	True	False	-	-	0.35	681517	-

- All numerical and binary features are extracted from videos frame by frame.
- All nominal features are extracted from video and folder names. For later purposes, Subject will be used for subject-wise normalization; Factors List and Facial Actions can be used for sample elimination after performing some performance test on them. And finally Reserved for Calibration will be used as a label for calibration phase.
- Drowsiness is ground truth label which is extracted from text file annotations provided by NTHU-DDD.
- AVG_EAR, MAR, MOE, PERCLOS, AVG_LEB, AVG_SOP and AVG_EC are the main features that will be used for classification. Closeness can also be added here.
- Features with LEFT and RIGHT prefix represent left and right eye values, and features with AVG represent average of both eye values. LEFT and RIGHT values can be used for classification also instead of AVG when one of the both eyes' value is missing.

Appendix – 2: Exploratory Data Analysis part-2

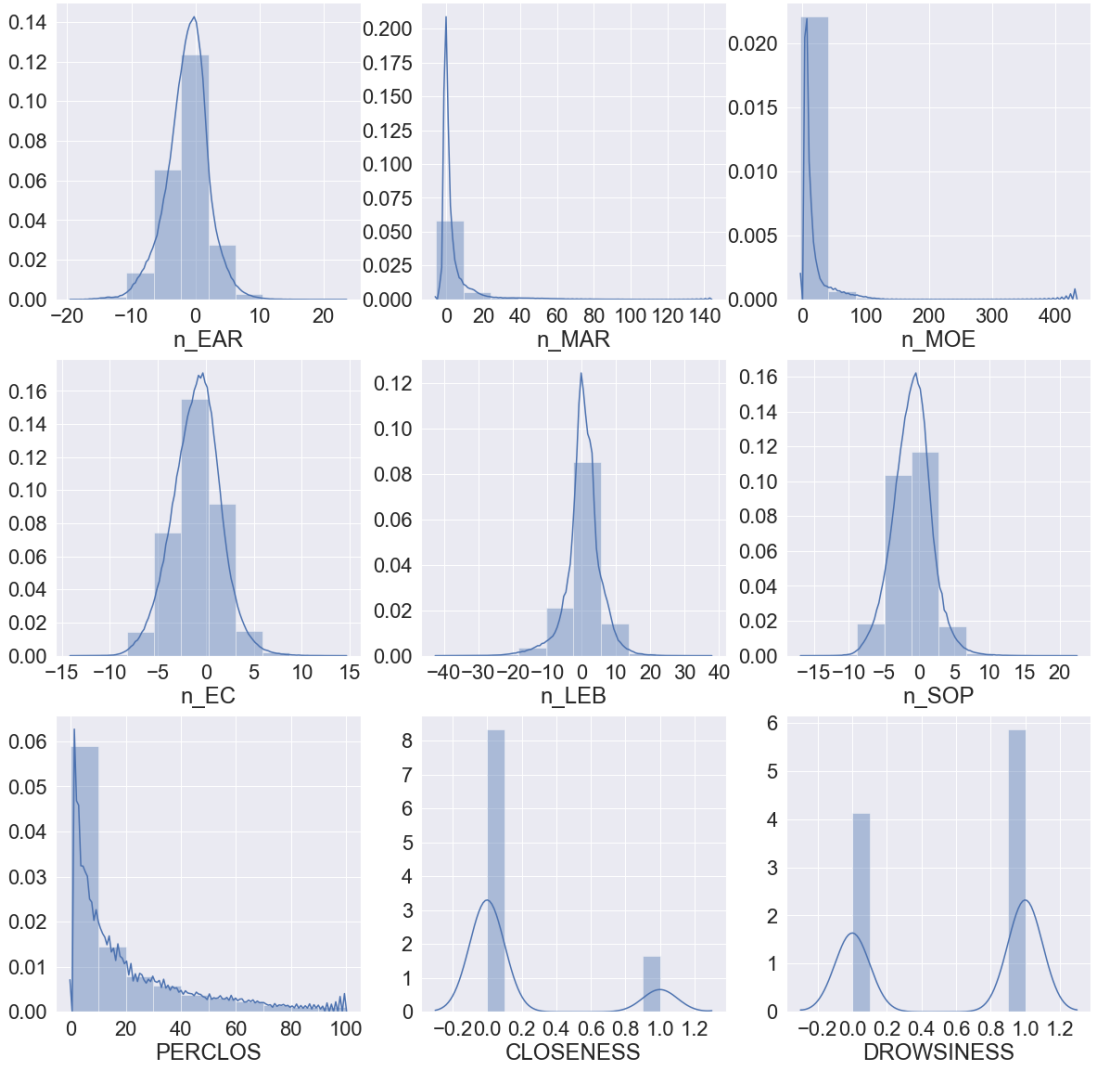


Fig. 1. Distribution charts.

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.

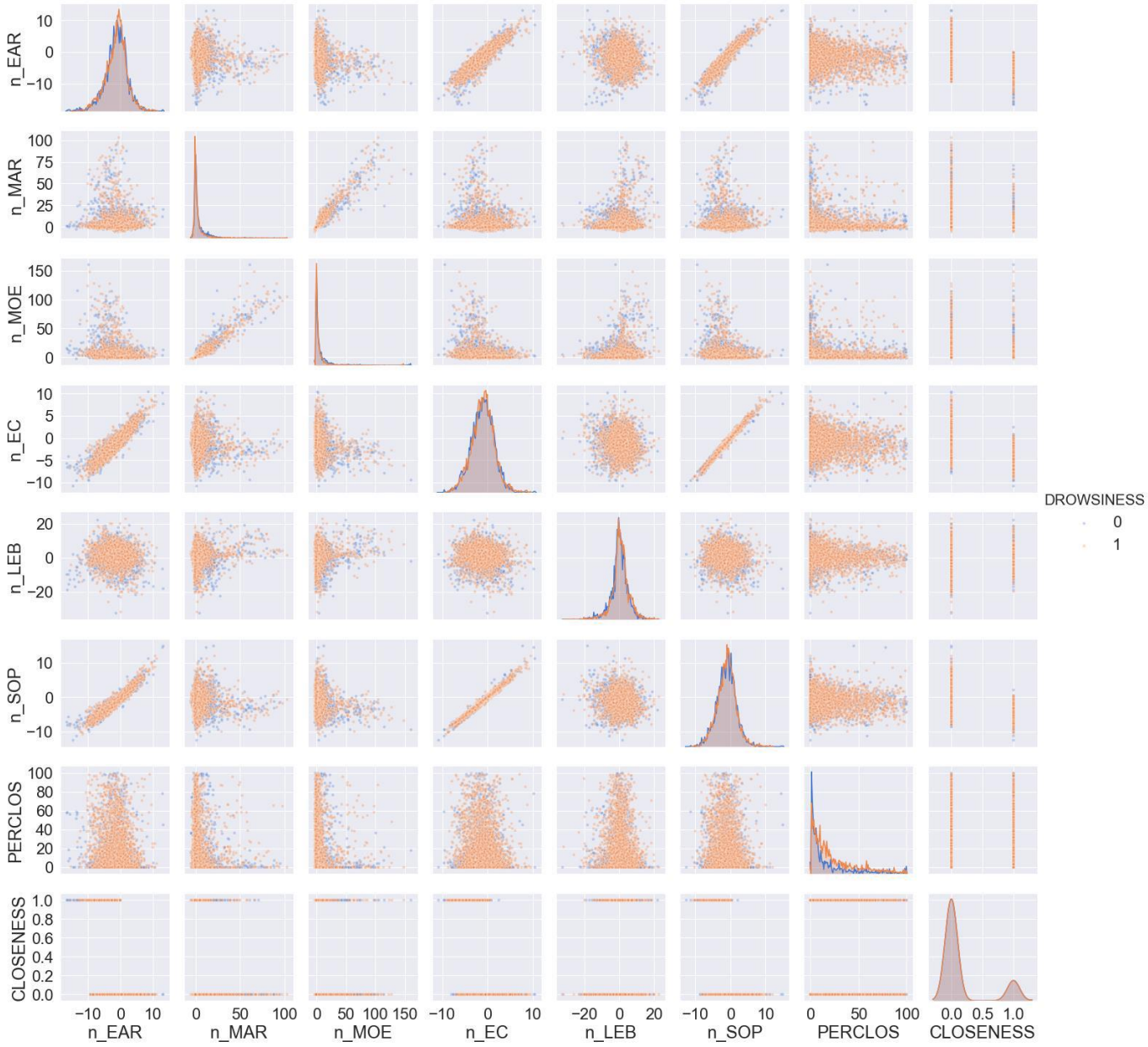


Fig. 2. Scatter plot matrix.

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.

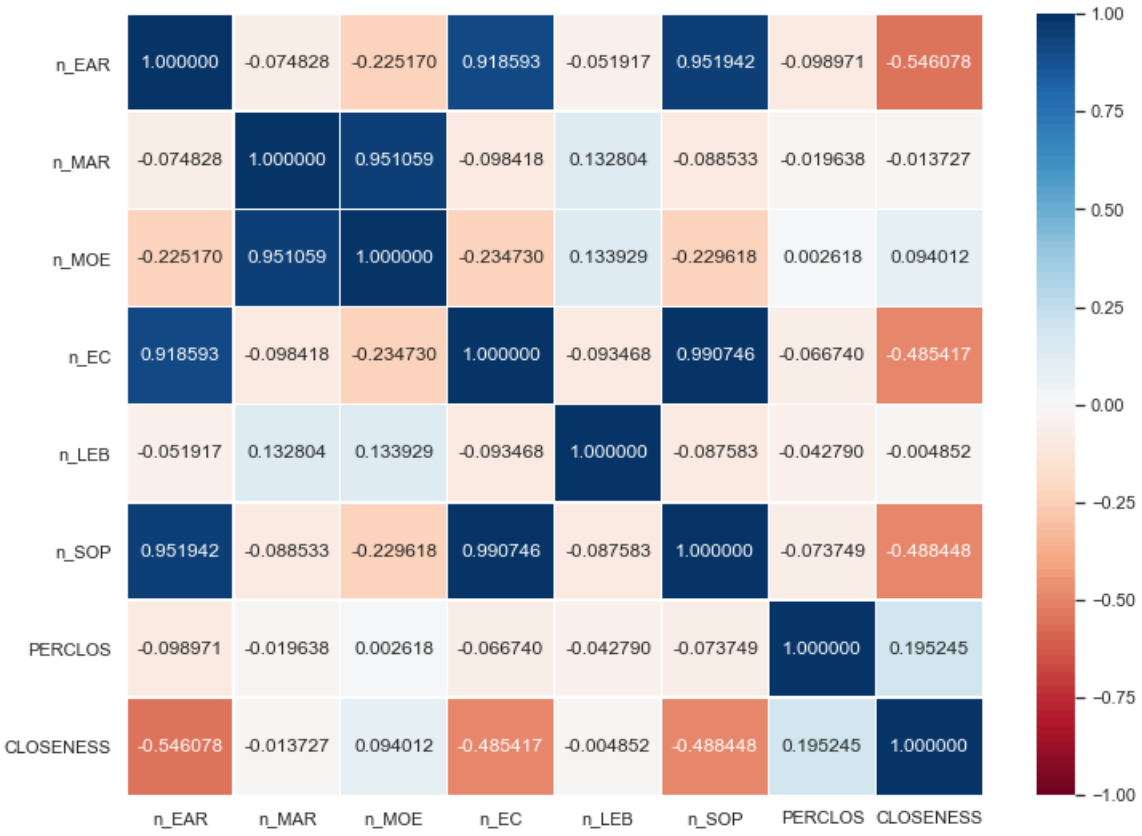


Fig. 3.. Correlation matrix.

Darker cells show high correlation (blues are positive correlations; reds are negative 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.

Appendix – 3: Feature Selection Results

#	Feature Name	Description	ANOVA						Mutual Info						Decision Tree					
			Fold1	Fold2	Fold3	Fold4	Fold5	Avg	Fold1	Fold2	Fold3	Fold4	Fold5	Avg	Fold1	Fold2	Fold3	Fold4	Fold5	Avg
1	EAR	Eye Aspect Ratio	27790	27687	27644	27617	28057	27759	0.0543	0.0540	0.0539	0.0535	0.0552	0,0541	1	1	1	1	1	1
2	MAR	Mouth Aspect Ratio	3759	3792	3783	3765	3775	3775	0.1314	0.1313	0.1301	0.1309	0.1321	0,1311	1	1	1	1	1	1
3	MOE	Mouth Over Eye	4846	4873	4857	4830	4869	4855	0.0243	0.0249	0.0234	0.0235	0.0236	0,0239	1	1	1	1	1	1
4	EC	Eye Circularity	22696	22459	22432	22512	22879	22596	0.0361	0.0360	0.0360	0.0357	0.0360	0,0359	2	2	1	2	2	1.8
5	LEB	Level of Eyebrows	14953	14904	14869	15080	15043	14970	0.0646	0.0652	0.0637	0.0663	0.0648	0,0649	1	1	1	1	1	1
6	SOP	Size of Pupil	26841	26594	26570	26647	27051	26741	0.0892	0.0897	0.0883	0.0882	0.0896	0,089	3	3	1	3	3	2.6
7	PERCLOS	Percentage of Eye Closure	17311	17022	17106	16943	17031	17083	0.0359	0.0353	0.0359	0.0359	0.0360	0,0358	1	1	1	1	1	1
8	CLOSENESS	Eye Closure Status	8448	8250	8343	8229	8388	8332	0.0268	0.0281	0.0264	0.0279	0.0278	0,0274	4	4	1	4	4	3.4

Fig. 1. Feature section with ANOVA, Mutual Info, Decision Tree.

Appendix – 4: T-Test Results

- **Accuracy T_Test** between KNN-5 & KNN-25:

  T Value = [-6.9592847], P Value = [0.00018816]

  p-value<=0.05 so there's a significant difference between models.

- **Precision T_Test** between KNN-5 & KNN-25:

  T Value = [2.06812435], P Value = [0.07486675]

  p-value>0.05 so there's no significant difference between models.

- **Recall T_Test** between KNN-5 & KNN-25:

  T Value = [-34.54848005], P Value = [9.06573874e-08]

  p-value<=0.05 so there's a significant difference between models.

- **F1 T_Test** between KNN-5 & KNN-25:

  T Value = [-11.94455852], P Value = [6.83443097e-06]

  p-value<=0.05 so there's a significant difference between models.

- **AUC T_Test** between KNN-5 & KNN-25:

  T Value = [-3.56333437], P Value = [0.00841836]

  p-value<=0.05 so there's a significant difference between models.