DRIVER DROWSINESS DETECTION

Delivery #7 Final Presentation

CSE4062 – Data Science, Spring2020, Group7

▶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 BALCI	abacinurettin@gmail.com	150715035

Problem Statement

- According to the US National Highway Traffic Safety Administration, approximately 100,000 crashes occur in US each year due to drowsiness of the drivers [1].
- Reasoning from this fact, the development of a robust and practical drowsiness detection system is a crucial step.
- We are aiming to design a system that checks a driver's facial behaviour, mainly eyes and mouth, in real-time to be able to detect drowsiness status of the driver.

Aims of the Project

Detecting Driver Drowsiness with High Accuracy

The main aim of this project is to detect driver drowsiness with high precision while using a low-cost camera.

Adaptivity to the Subject

Building a system that will be able to detect the doziness of all people from different ethnicities and personal characteristics. This project will be able to adjust itself by each person's facial feature standards.

Related Work

- **A Practical Driver Fatigue Detection Algorithm Based on Eye State (Liu et al., 2010) [2]**
  - In this paper authors aim to detect driver drowsiness by calculating PERCLOS.
  - The algorithm developed in this article is capable of detecting eye closure in high-speed after locating the eye.
  - We used this article to extract the PERCLOS feature from NTHU dataset.
- **Drowsiness Detection with Machine Learning (Zhong et al., 2019, Online Blog) [3]**
  - In this article, authors developed a system to detect driver drowsiness with some facial features.
  - They used Dlib for landmark detection and extract facial features. Then they implement feature importance and compared lots of classification algorithms.
  - This article was a base method for our project. We used this article for landmark detection, feature importance and some classification algorithms.

Scope

The project consists of 6 phases:

1. Processing videos and extracting frame-based features
2. Preprocessing and normalization for subject-wise and column-wise
3. Exploring the **NTHU** dataset
4. Running appropriate classifiers on the dataset
5. Running some cluster experiments on the dataset
6. Proposing an accurate driver drowsiness detection system

Scope: Constraints and Assumptions

Constraints:

- Insufficient illumination of the face according to time of the day,
- If the driver is wearing sunglasses or a hat and may have facial hair,
- There might be obstacles in front of the driver’s eyes e.g. the driver’s hand.

Assumptions:

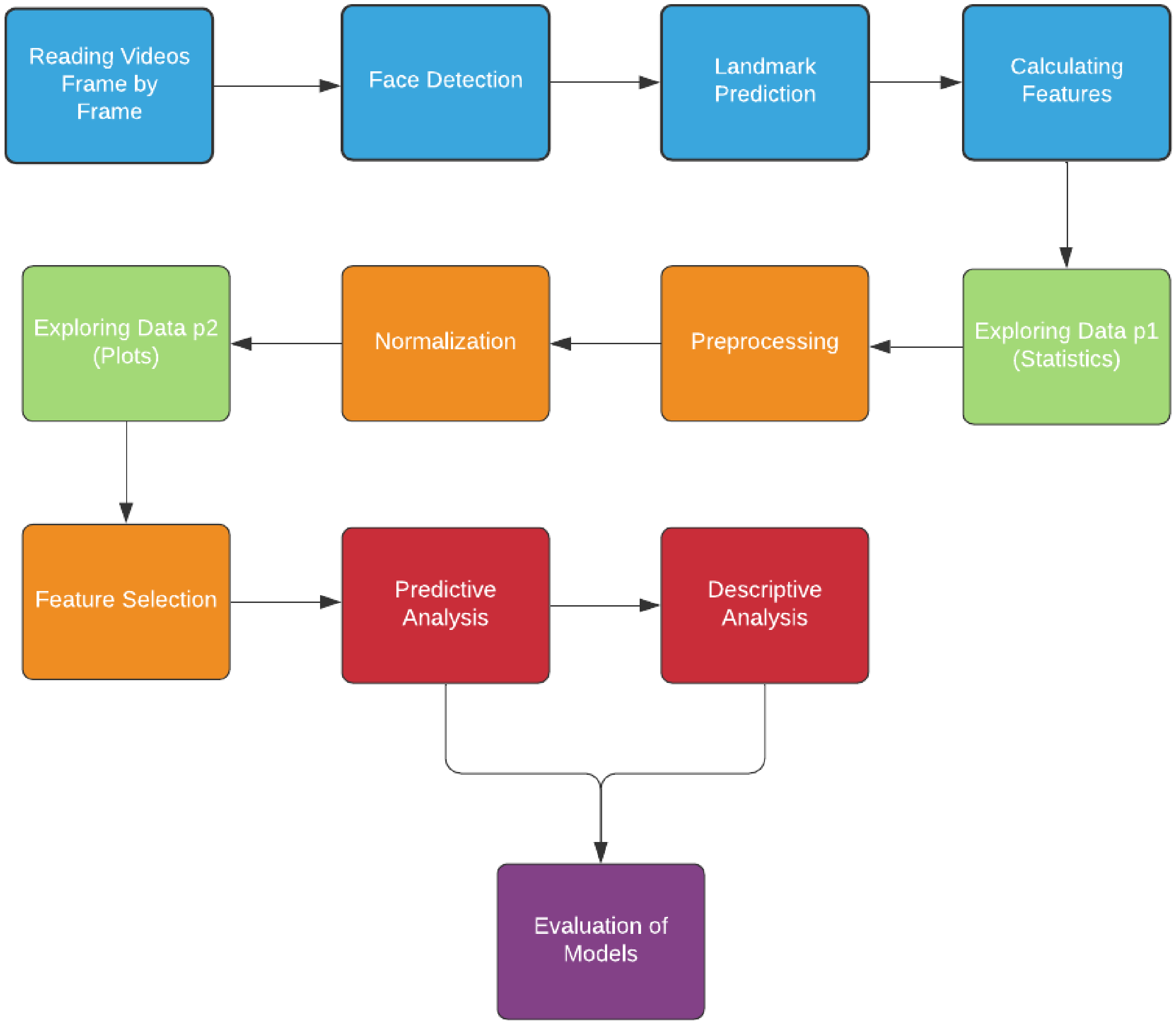
- It is assumed that all datasets that are used in this project are correctly annotated.
- The images are collected with sufficient illumination of the face.
- The camera is directly placed in front of the driver and nearly has one arm length distance.
- The driver’s head is up and his face fits in the camera.

Dataset Specifications

國立清華大學
NTHU-DDD

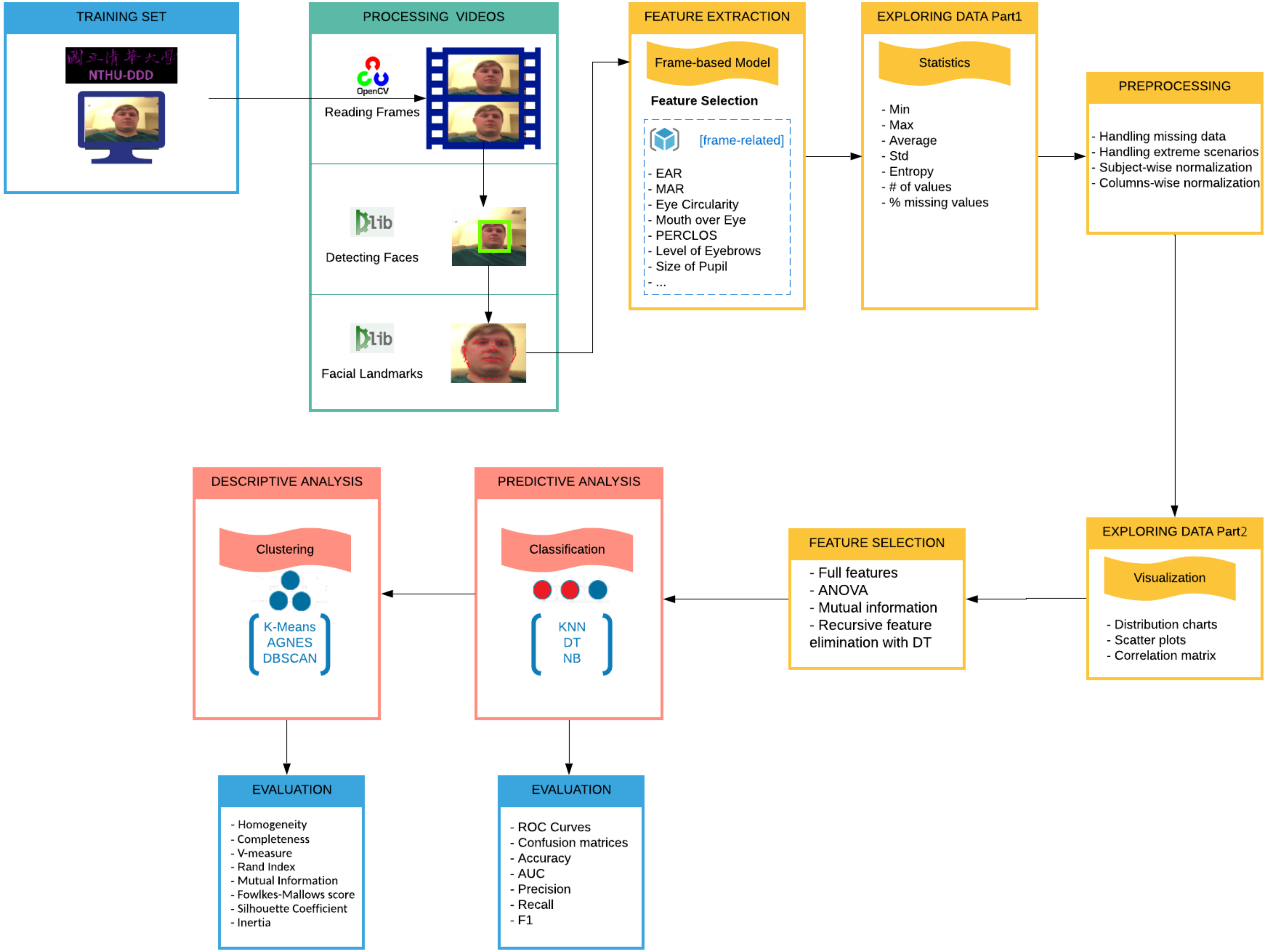
NTHU Driver Drowsiness Dataset [4]**:**

- National Tsing Hua University
- 36 people, 90 videos, 10 hours
- Several ethinicities, driving scenarios and accesories
- **Acting** videos
- Only **frame labels** are provided: 0, 1 for eye closeness, nodding and yawning behaviours



Methodology: Flowchart

Methodology: Conceptual Diagram



Processing Videos

Implementation of the project starts with **Processing Videos Phase** which includes steps;

1) Reading Video Frames

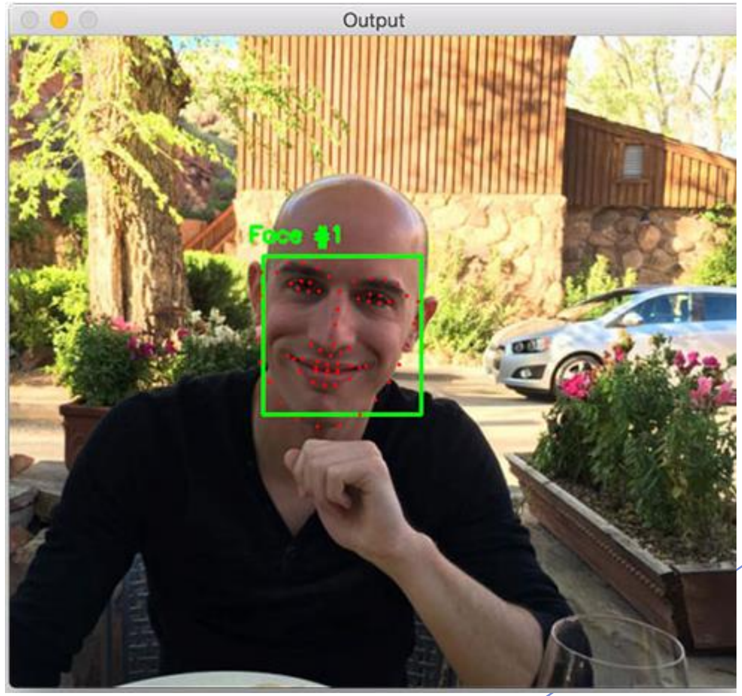
- Either can be done from a dataset or a camera in a real-time manner.
- For this step, **opencv-python**, a python version of OpenCV library, is used.

Processing Videos

Implementation of the project starts with **Processing Videos Phase** which includes steps;

2) Detecting Faces

- **Detecting faces** with **Dlib's get_frontal_face_detector** 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.

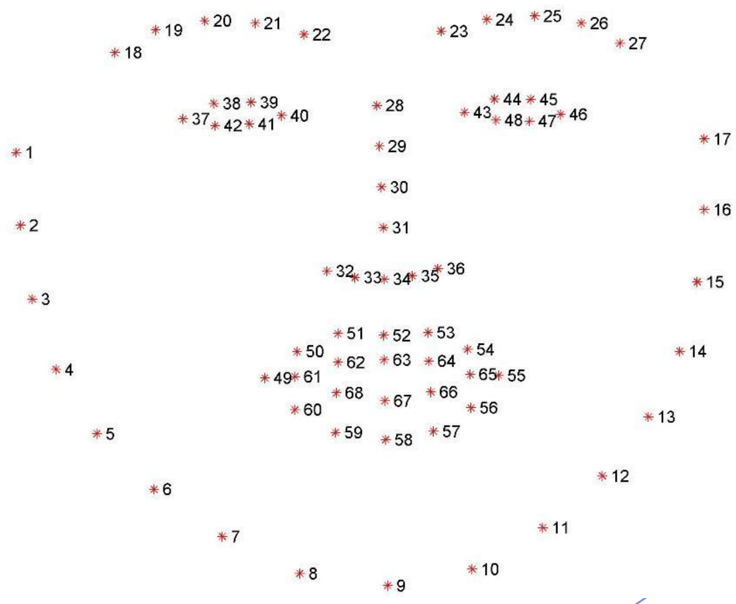


Processing Videos

Implementation of the project starts with **Processing Videos Phase** which includes steps;

3) Predicting Facial Landmarks

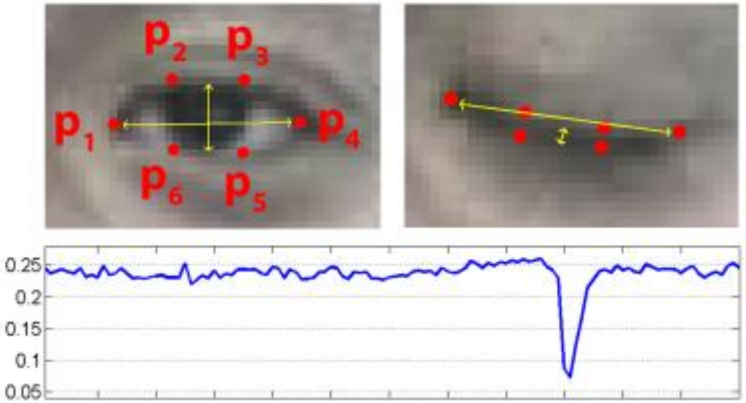
- **Predicting facial landmarks** with **Dlib's shape_predictor** method which is an implementation of the paper Kazemi and Sullivan (2014) [5].
- This method also uses a pre-trained model of an ensemble of regression trees and predicts 68 facial landmarks which can be seen in **figure on right.**
- All features will be used in later phases, are extracted from these positional landmarks.



Feature Extraction Phase

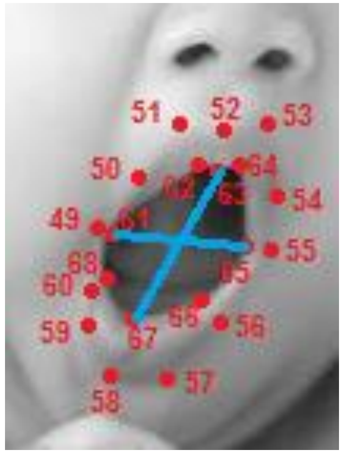
All of the features are calculated for every frame in the video and will be handed into the classifier.

- **<u>Eye Aspect Ratio(EAR)</u>**



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

- **<u>Mouth Aspect Ratio(MAR)</u>**



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

Feature Extraction Phase

- **<u>Eye Circularity(EC)</u>**

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

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

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

Feature Extraction Phase

- **Mouth over Eye (MOE)**

It is basically EAR over MAR equation.
It's an additional feature which can be interpreted as true drowsiness.

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

- **PerClos(PC)**

Indicates the frequency of closed eyes up.

$$PC = \frac{count\ of\ frames\ when\ the\ eyes\ are\ closed}{total\ count\ of\ frames\ up\ until\ that\ moment} \times 100\%$$

Feature Extraction Phase

All of the features are calculated for every frame in the video and will be handed into the classifier.

- **Level of Eyebrows(LEB)**

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

- **Size of Pupil(SOP)**

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

Blue line represents Eye Width [p37, p40] and orange line represents Pupil Diameter [p38, p41]. The ratio of them is called Size of Pupil (SOP).

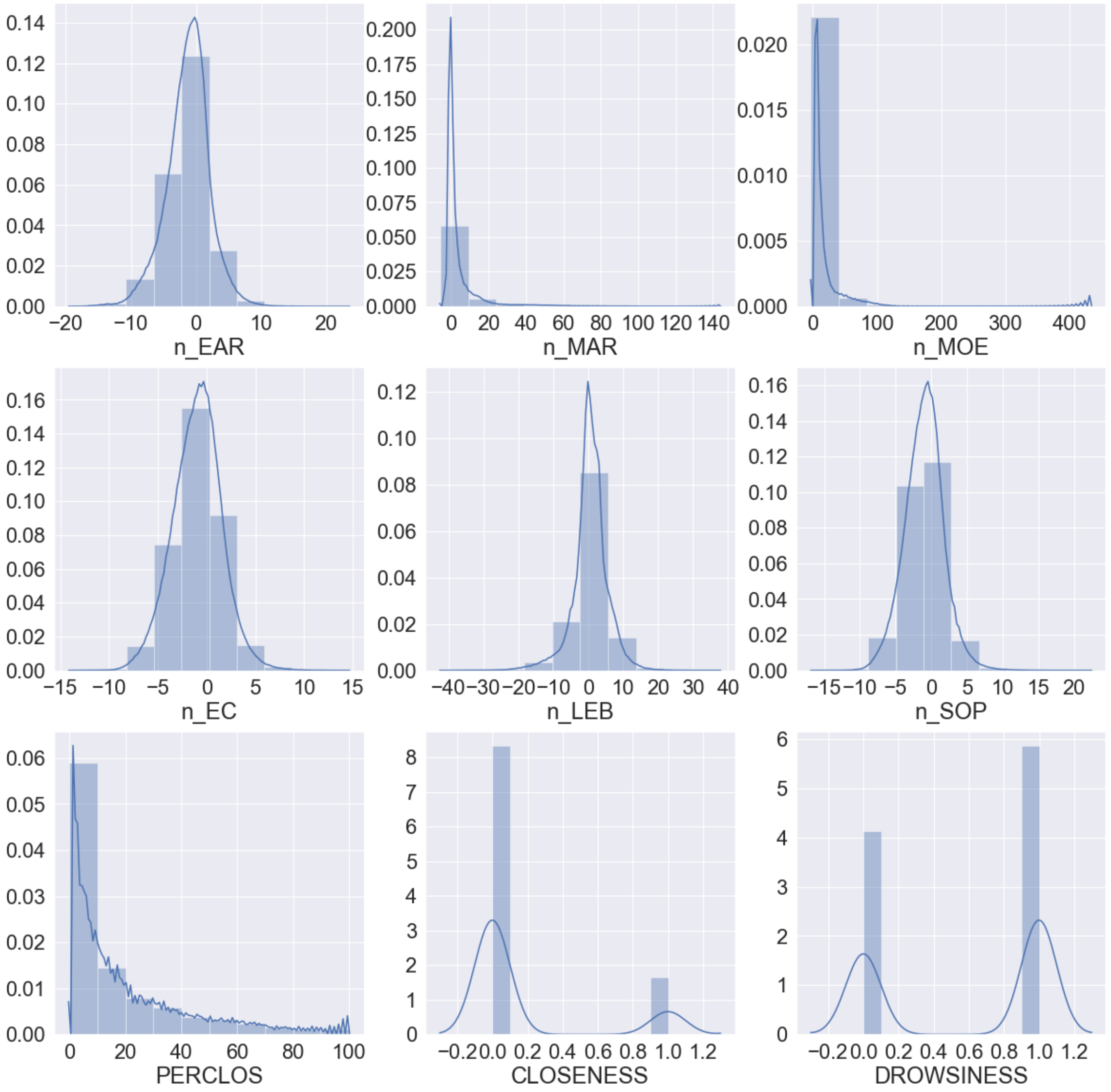
Exploring Data p1: Statistics on Resulting Dataframe

Index	n_EAR	n_MAR	n_MOE	n_EC	n_LEB	n_SOP	PERCLOS	CLOSENESS	DROWSINESS
296	0.261608	-0.403644	-0.403713	-0.305519	0.226616	-0.161908	-0.775884	-0.446668	0
297	0.416698	-0.255756	-0.286058	0.199038	0.442506	0.25655	-0.775884	-0.446668	0
298	0.147575	-0.250873	-0.268838	0.33316	0.0163402	0.262167	-0.775884	-0.446668	0
299	-0.0172533	-0.283602	-0.289363	-0.00302558	0.236438	-0.0344702	-0.775884	-0.446668	1
300	0.00289108	-0.399373	-0.394028	-0.0723458	0.298181	-0.0419824	-0.775884	-0.446668	1
301	-0.270607	-0.519511	-0.501613	-0.290815	0.00494504	-0.2727	-0.775884	-0.446668	1

	n_EAR	n_MAR	n_MOE	n_EC	n_LEB	n_SOP	PERCLOS	CLOSENESS	DROWSINESS
count	614,593.00	614,593.00	614,593.00	614,593.00	614,593.00	614,593.00	614,593.00	614,593.00	614,593.00
mean	-0.00	-0.00	0.00	0.00	-0.00	-0.00	0.00	-0.00	0.59
std	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.49
min	-5.45	-0.88	-0.59	-5.13	-8.51	-5.70	-0.78	-0.45	0.00
25%	-0.57	-0.43	-0.42	-0.65	-0.43	-0.63	-0.71	-0.45	0.00
50%	0.07	-0.31	-0.31	0.03	0.02	0.03	-0.43	-0.45	1.00
75%	0.61	-0.05	-0.06	0.64	0.52	0.62	0.35	-0.45	1.00
max	7.41	12.37	30.59	6.11	7.37	8.47	3.89	2.24	1.00

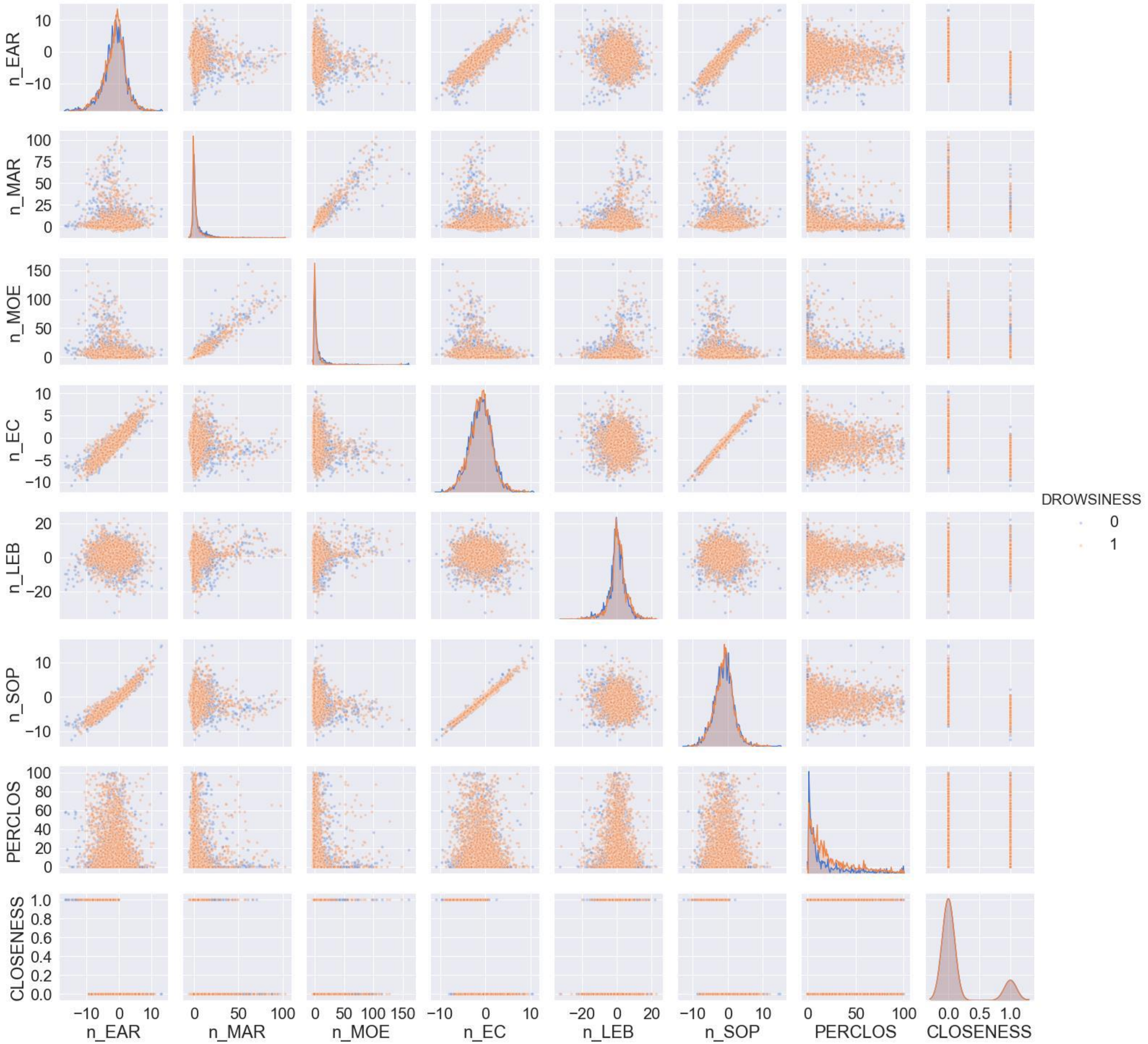
Exploring Data p2:
Distribution Charts

- 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.



Exploring Data p2:
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.



Exploring Data p2:
Correlation Matrix

- 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.

	n_EAR	n_MAR	n_MOE	n_EC	n_LEB	n_SOP	PERCLOS	CLOSENESS
n_EAR	1.000000	-0.074828	-0.225170	0.918593	-0.051917	0.951942	-0.098971	-0.546078
n_MAR	-0.074828	1.000000	0.951059	-0.098418	0.132804	-0.088533	-0.019638	-0.013727
n_MOE	-0.225170	0.951059	1.000000	-0.234730	0.133929	-0.229618	0.002618	0.094012
n_EC	0.918593	-0.098418	-0.234730	1.000000	-0.093468	0.990746	-0.066740	-0.485417
n_LEB	-0.051917	0.132804	0.133929	-0.093468	1.000000	-0.087583	-0.042790	-0.004852
n_SOP	0.951942	-0.088533	-0.229618	0.990746	-0.087583	1.000000	-0.073749	-0.488448
PERCLOS	-0.098971	-0.019638	0.002618	-0.066740	-0.042790	-0.073749	1.000000	0.195245
CLOSENESS	-0.546078	-0.013727	0.094012	-0.485417	-0.004852	-0.488448	0.195245	1.000000

Predictive Analysis: Feature Selection

#	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

We first determined 4 different feature subsets:

1. Full sets of features
2. Feature selection with SelectKBest method of SkLearn. As scoring function we choosed f_classif (ANOVA) .
3. Feature selection with SelectKBest method of SkLearn. As scoring function we choosed f_ mutual_info_classif (Mutual Information) .
4. Feature selection with RFECV (Recursive feature elimination on CV). As estimator parameter we chosoed DecisionTreeClassifier.

Predictive Analysis: Classification Experiments

We determined 5 different classifying models:

1. KNeighborsClassifier model of Sklearn with 5 neighbours
2. KNeighborsClassifier model of Sklearn with 25 neighbours
3. DecisionTreeClassifier model of Sklearn with gini
4. DecisionTreeClassifier model of Sklearn with entropy
5. GaussianNB model of Sklearn.

#	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

Predictive Analysis: Confusion Matrices and t-test

- **Accuracy T_Test between KNN-5 & KNN-25:**
  T Va lue = [-6.9592847], P Value = [0.00018816]
  p-value<=0.05 so <u>there's a significant difference</u> between models.

- Precision T_Test between KNN-5 & KNN-25:
  T Value = [2.06812435], P Value = [0.07486675]
  p-value>0.05 so <u>there's NO significant difference</u> between models.

- Recall T_Test between KNN-5 & KNN-25:
  T Value = [-34.54848005], P Value = [9.0674e-08]
  p-value<=0.05 so <u>there's a significant difference</u> between models.

- F1 T_Test between KNN-5 & KNN-25:
  T Value = [-11.94455852], P Value = [6.8344e-06]
  p-value<=0.05 so <u>there's a significant difference</u> between models.

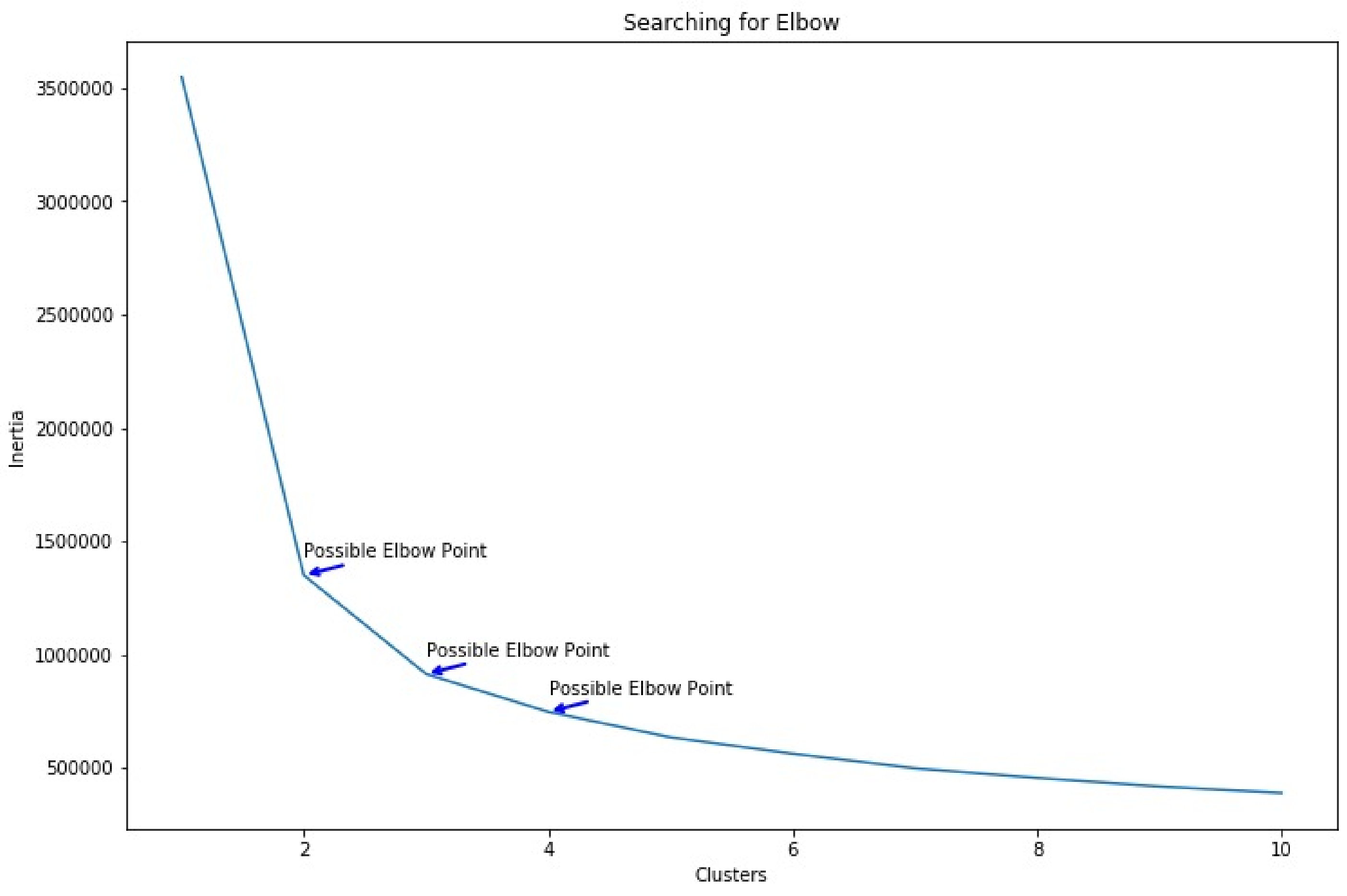
- AUC T_Test between KNN-5 & KNN-25:
  T Value = [-3.56333437], P Value = [0.00841836]
  p-value<=0.05 so <u>there's a significant difference</u> between models.

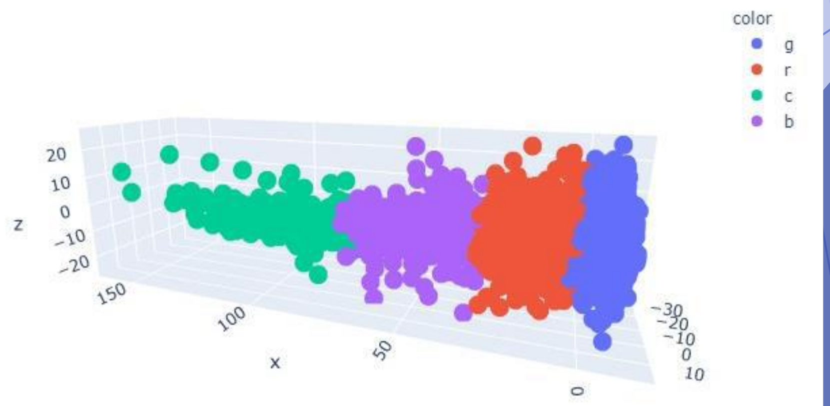
Descriptive Analysis: Setup

- After loading data, we apply **PCA** to our data so **we reduced feature number to 9 to 3 for visualization purposes.** So, we named our new 3 features as "PCA_feature1", "PCA_feature2" and "PCA_feature3".
- We choose 3 different clustering models of Sklean: **KMeans, AgglomerativeClustering and DBSCAN**. Kmeans 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.

Descriptive Analysis: K-means Elbow Method

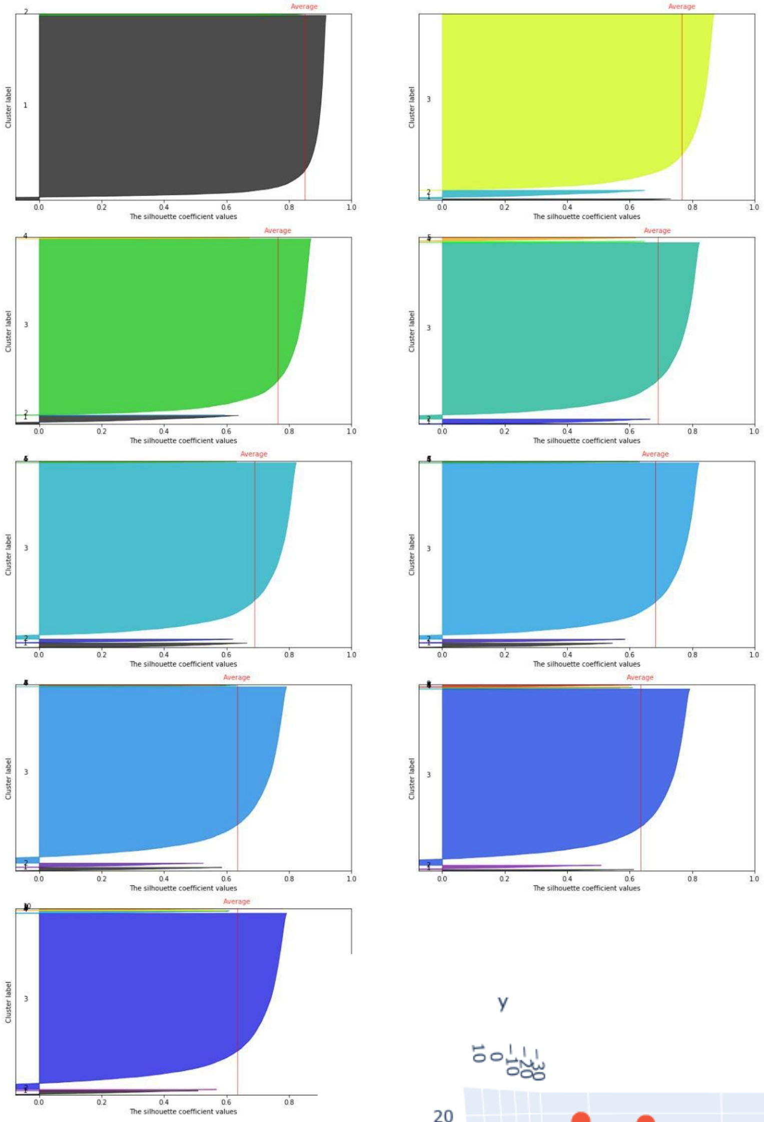
- "**inertia_**" attribute of **KMeans class of SKLearn** provides sum of squared distances of samples to their closest cluster center. We plot it and decide on the parameter "n_clusters" by using **elbow method.**
- We plot candidate parameters seperately and decide on **n_clusters = 4**

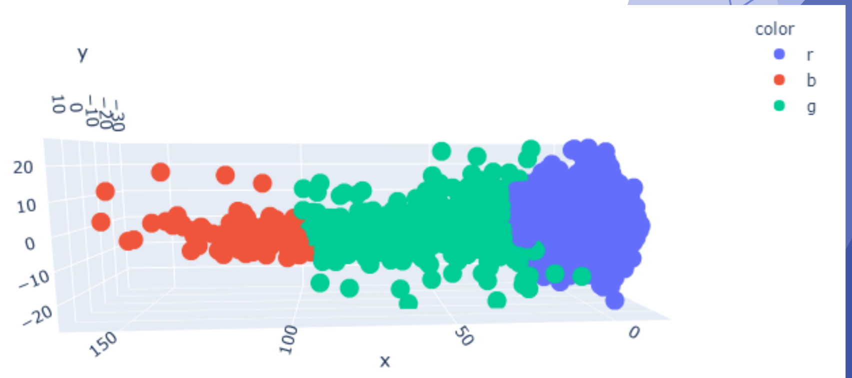




Descriptive Analysis: AGNES and Silhouette plots

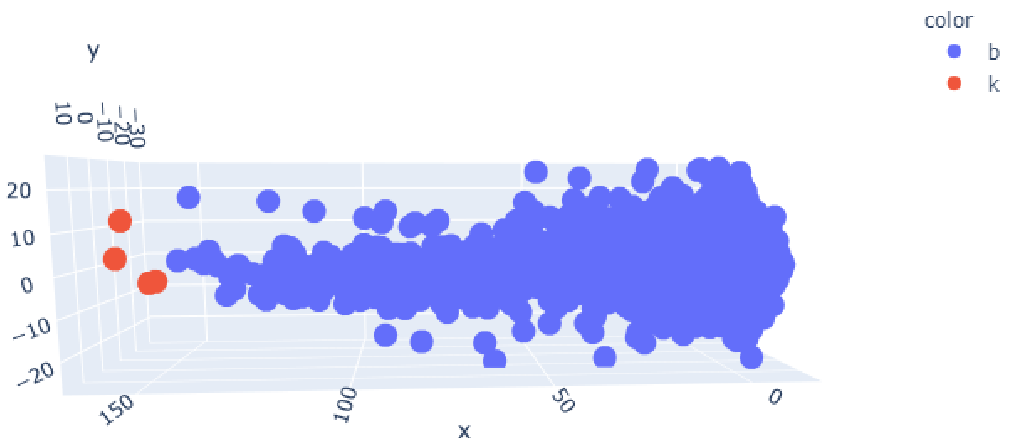
- There's no inertia (Sum of squared distances of samples to their closest cluster center) attribute of **AgglomerativeClustering** class so we used silhouette plots to select cluster number of AGNES.
- We plot candidate parameters seperately and decide on **n_clusters = 3**





Descriptive Analysis: DBSCAN and GridSearch

- **eps**: The maximum distance between two samples for one to be considered as in the neighborhood of the other (default: 0.5).
- **min_samples**: The number of samples (or total weight) in a neighborhood for a point to be considered as a core point. This includes the point itself (default: 5).

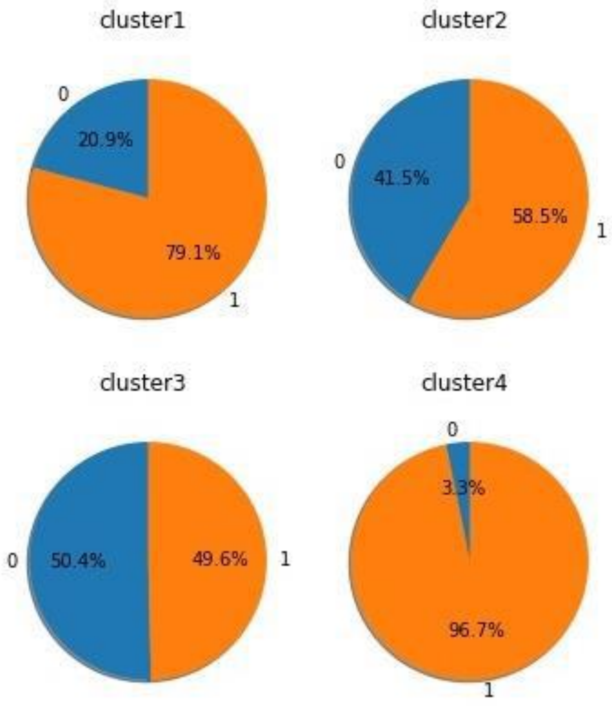


- There's also no inertia attribute of **DBSCAN** class in SKLearn so we used gridsearch on silhouette coefficient score to determine best parameters of DBSCAN.
- Apperantly best parameters (eps:5, min_samples=11) are resulting only 1 cluster. Red ones in the plot are just outliers.

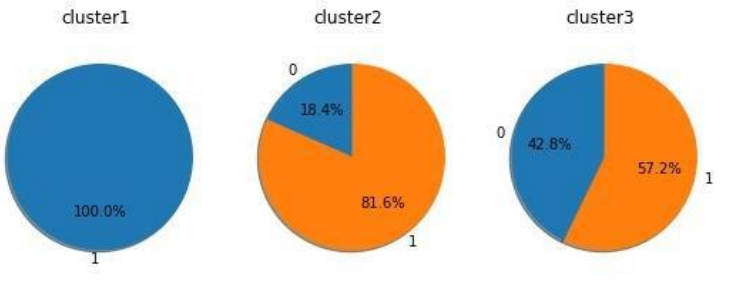
Descriptive Analysis: Pie Charts for Class Distributions

Homogeneity and Completeness principles:

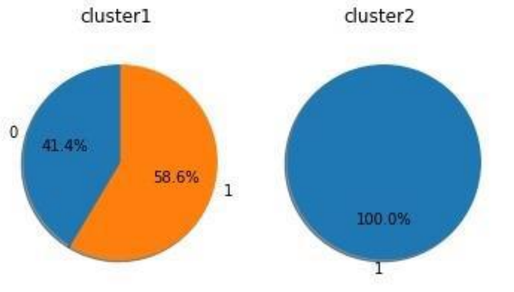
- **Homogeneity**: For perfect clustering, each cluster contains only members of a single class.
- **Completeness**: For perfect clustering, all members of a given class are assigned to the same cluster.
- **V-measure**: Harmonic mean of Homogeneity and Completeness.



K-means with 4 clusters



AGNES with 3 clusters



DBSCAN with 2 clusters

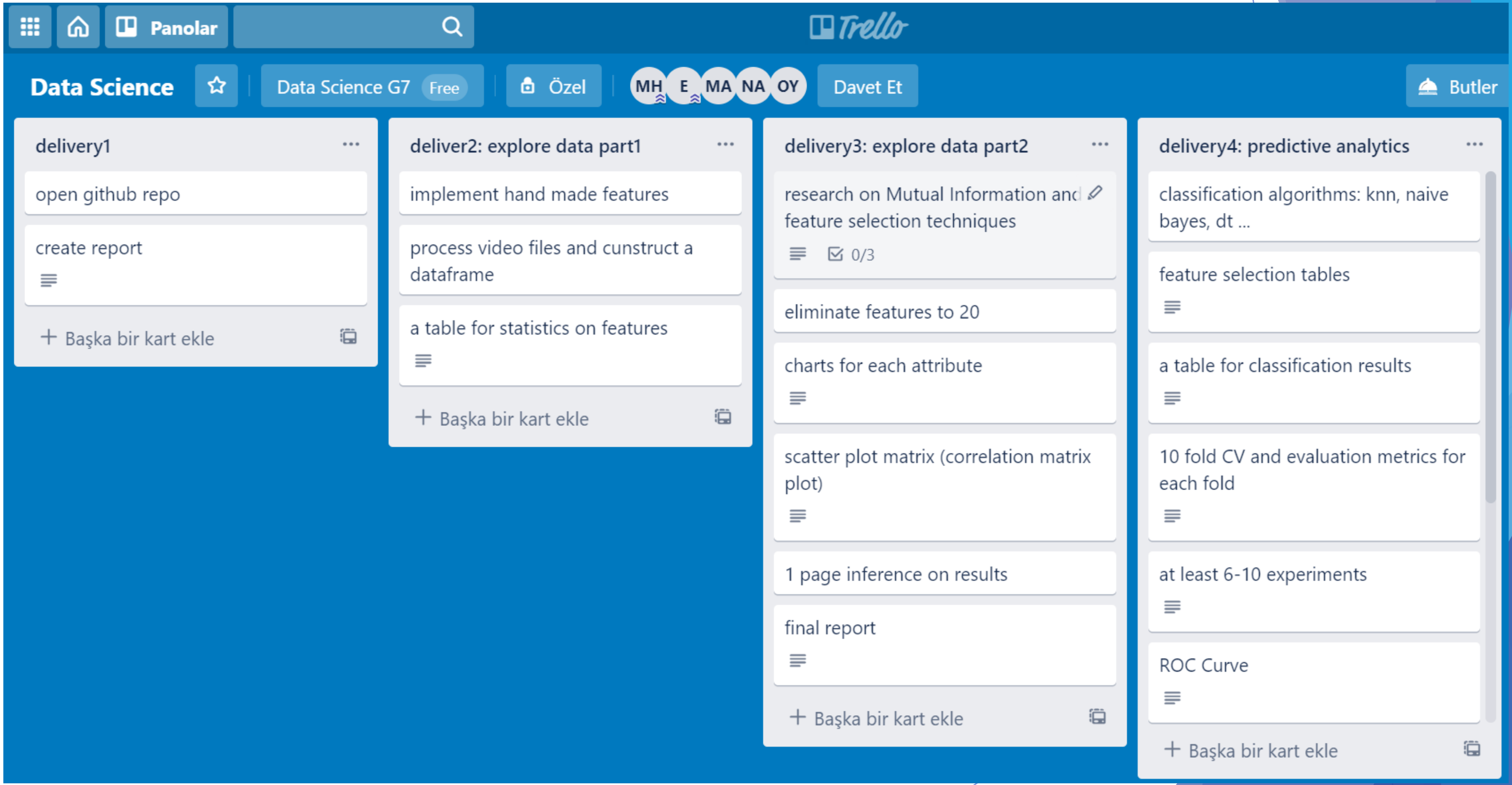
Descriptive Analysis:
Clustering Evaluation

- **Adjusted Rand Index** is a function that measures the similarity of the two assignments, ignoring permutations and with chance normalization, <u>given the knowledge of the ground truth class.</u>
- **Adjusted Mutual Information** is a function that measures the agreement of the two assignments, ignoring permutations, <u>given the knowledge of the ground truth class.</u>
- **Fowlkes-Mallows score** is geometric mean of precision and recall, <u>given the knowledge of the ground truth class.</u>
- **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), <u>without knowing the ground truth labels.</u>
- **Inertia (SSE)** 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.

Descriptive Analysis:
Clustering Evaluation

#	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

Project Colloboration Tool: Trello



Project Colloboration Tool: Completed Tasks

- Processing videos and constructing DataFrame
  1. Reading videos and extracting facial features
  2. Reading annotations from the dataset
  3. Concatenating dataframes by fixing matching errors
- Preprocessing
  1. Handling missing values
  2. Subject-wise normalization
  3. Column-wise normalization
- Exploring data with statistics and visualization
- Feature selection: ANOVA, MI, RFE w DT
- Trying some ML classifiers: kNN, Naive Bayes, Decision Tree
- Trying some clustering models: K-means, AGNES, DBSCAN
- Evaluation of models

Project Colloboration Tool: Difficulties and Solutions

- While processing videos and constructing the DataFrame
  1. **Multiple faces and no face detected**: frames are marked
  2. **Time lapses**: frames are discarded
  3. **Mismatching annotations (First two frames of NTHU-test):** frames are marked
- While preprocessing
  1. **Handling missing values**: Marked frames are discarded
  2. **Unconsistent thresholding for different subjects:** Subject-wise normalization
- While calculating feature importances
  1. Unexpectedly high values for some features: Column-wise normalization
  2. Preventing bias on test-set while using t-test with feature-importance techniques: 5x2 cross-validation
- While implementing clustering models
  1. Visualization in 3D: PCA to reduce dimensions to 3 from 9.
  2. Memory errors for AGNES and DBSCAN: Resampling 10k from 610k

References

- [1] 1-Hartman, K. and J. Strasser, Saving Lives ThroughAdvanced Vehicle Safety Technology:Intelligent Vehicle Initiative Final Report. 2005,Department of Transportation: Washington, DC.
- [2] 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.
- [3] 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 02 / 06 /2020)
- [4] Computer Vision Lab, National Tsuing Hua University, Driver Drowsiness Detection Dataset. [Online]. Available: http://cv.cs.nthu.edu.tw/php/callforpaper/datasets/DDD/ (Date of Access 20 / 04 /2020)
- [5] 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.