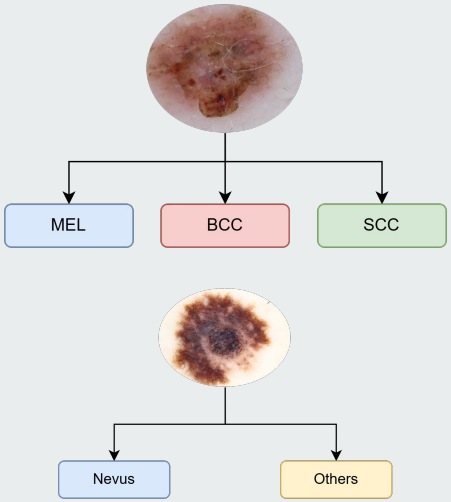
Project 1: Classical Machine Learning for Skin Lesion Classification.

Muhammad Kabir Hamzah
Faisal Farhan

Erasmus Mundus Joint Masters in Medical Imaging and Applications



Universitat de Girona

MAIA

Introduction

Objective: Develop a computer aided algorithm for the diagnosis in dermoscopic images.

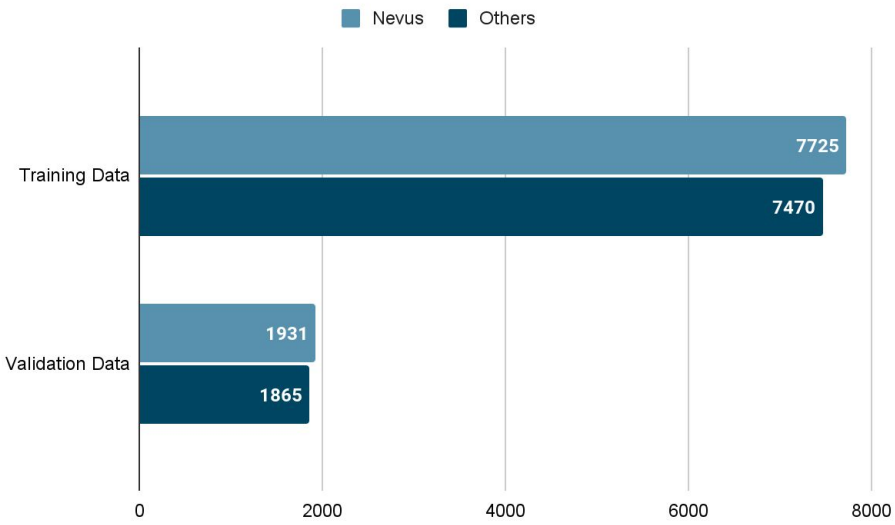
Challenges:

#1. The binary problem of classifying Nevus images vs all the others.

#2. A three-class problem consisting on the classification of cancers: melanoma vs basal cell carcinoma vs squamous cell carcinoma.

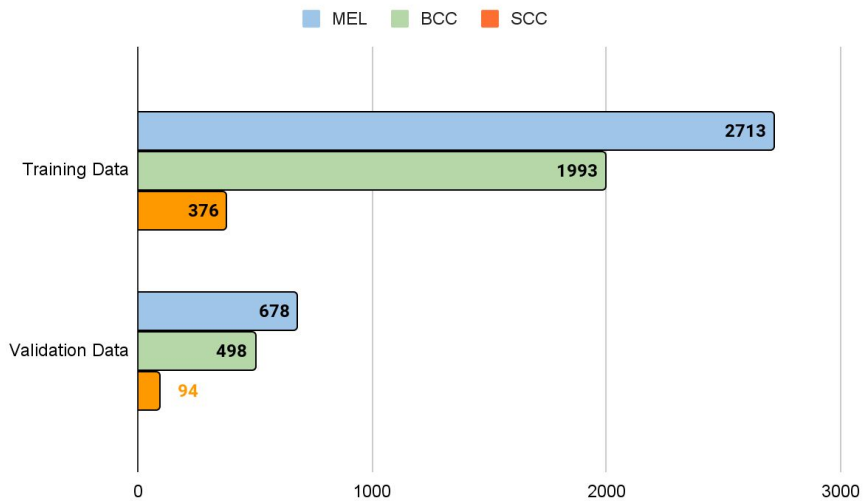
Dataset

#Challenge 1



Total Training Data: 15195
Total Validation Data: 3796
Total Test Data: 6340

#Challenge 2



Total Training Data: 5760
Total Validation Data: 1270
Total Test Data: 2121

Classification Pipeline: #Challenge 1

Dataset

Image Preprocessing

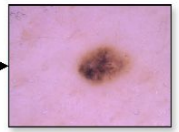
Hair Removal
ROI
Color Normalization

Feature Extraction

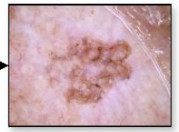
Color Moments
Color Histogram
Texture: GLCM and LBP

Model Training and Selection
Random Forest
KNN
XGBoost
LightGBM
SVM
CatBoost

Prediction



Nevus



Others

Classification Pipeline: #Challenge 2

Dataset

Image Preprocessing

Hair Removal
ROI
Color Normalization

Class Weights

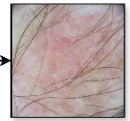
Feature Extraction

Color Moments
Color Histogram
Texture: GLCM and LBP

Model Training and Selection

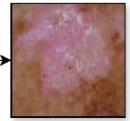
Random Forest
KNN
XGBoost
LightGBM
SVM
CatBoost

Prediction



BCC

MEL



SCC

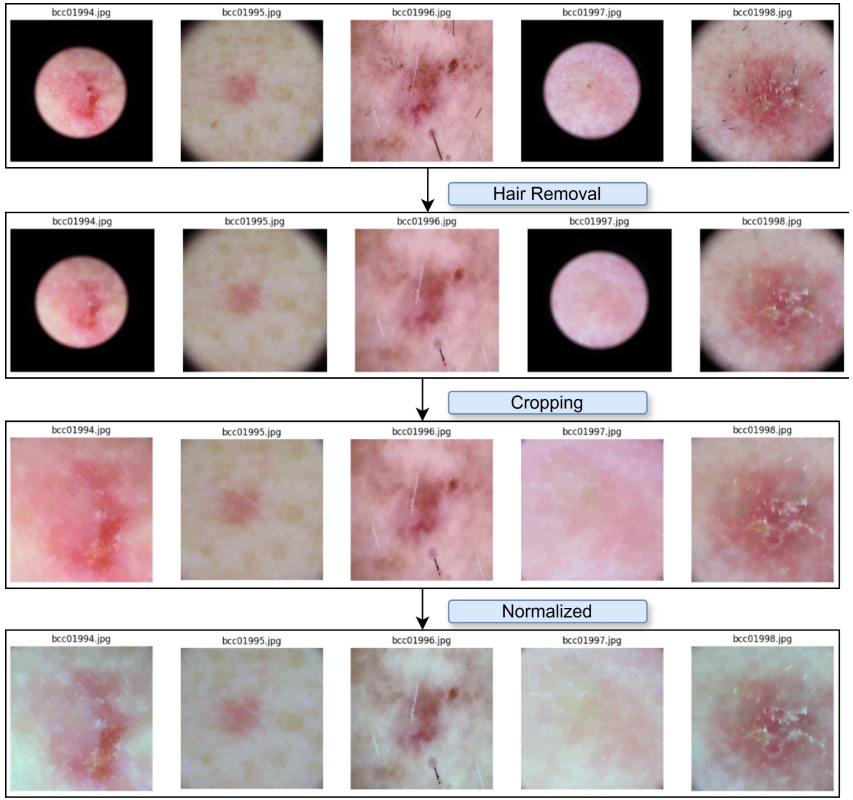
Image Preprocessing

- **Hair Removal:** Remove hair-like structures from an image using morphological operations and inpainting.
- **Region of interest:** Crop the image to get the region of interest. Remove the vignette frame.
- **Color Normalization:** Normalize the colors of the images using the Shades of Gray algorithm with gamma correction.

Preprocessors

Hair Removal
ROI
Normalization

Image Preprocessing



Data Imbalance

Augmentation

Zoom
Horizontal Flip

Class Weights

mel: 0.142
bcc: 0.104
scc: 0.753

We used Class Weights to deal with the data imbalance problem in #Challenge 2 as it gave better accuracy than augmentation.

Feature Extraction iter 1

Features

Gray Level Co-occurrence Matrix (GLCM)
Distances: 1-5 Angle: 0 - 135°
Statistics: contrast, correlation, energy, homogeneity, dissimilarity

Local Binary Pattern (LBP)
Radius: 5 Number of points: 8 & 16

Feature Extraction iter 2

Features

Gray Level Co-occurrence Matrix (GLCM)
Distances: 1-5 Angle: 0 - 135°
Statistics: contrast, correlation, energy, homogeneity, dissimilarity

Local Binary Pattern (LBP)
Radius: 5 & 10 Number of points: 8 & 16

Feature Extraction iter 3

Features

Gray Level Co-occurrence Matrix (GLCM)
Distances: 1-5 Angle: 0 - 135°
Statistics: contrast, correlation, energy, homogeneity, dissimilarity

Local Binary Pattern (LBP)
Radius: 5 & 10 Number of points: 8 & 16

Gabor Features
theta: 0 - 180° Sigma: (1, 3) Frequency: (0.05, 0.25)

Feature Extraction Final Pipeline

Color Features

Color Moments: Extracting Color Moments of the images.

Color Histogram: Extracting Color Histogram of the images.

RGB: color image
- mean, variance, standard deviation, skewness for each channel.
- histogram for each channel.

HSV: color image
- mean, variance, standard deviation, skewness for each channel.
- histogram for each channel.

LAB: color image
- mean, variance, standard deviation, skewness for each channel.
- histogram for each channel.

Feature Extraction Final Pipeline

Texture Features

Gray Level Co-occurrence Matrix (GLCM)
Distances: 1-5 Angle: 0 - 135°
Statistics: contrast, correlation, energy, homogeneity, dissimilarity

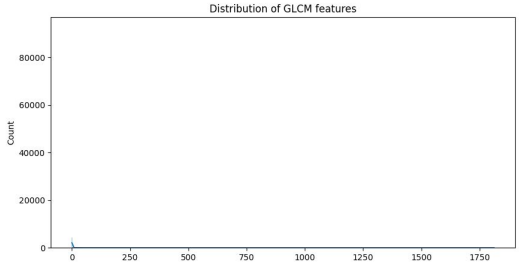
Local Binary Pattern (LBP)
Radius: 5 & 10 Number of points: 8 & 16

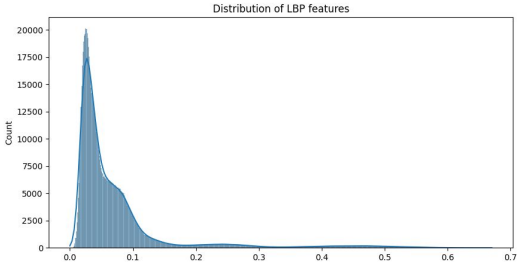
Summary of Feature Extraction Iterations

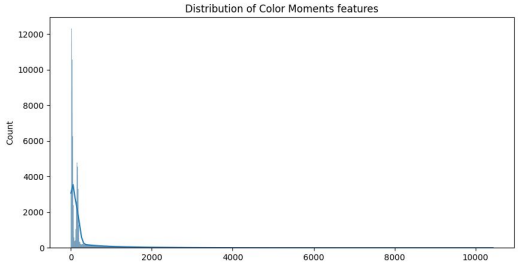
Trials	Features	Best Accuracy
Iteration 1	GLCM, LBP	0.78
Iteration 2	GLCM, LBP	0.795
Iteration 3	GLCM, LBP, GABOR	0.795
Final Pipeline	**GLCM, LBP, Color Features**	**0.84**

We implemented Feature Extraction Final pipeline for both #Challenge 1 and #Challenge 2.

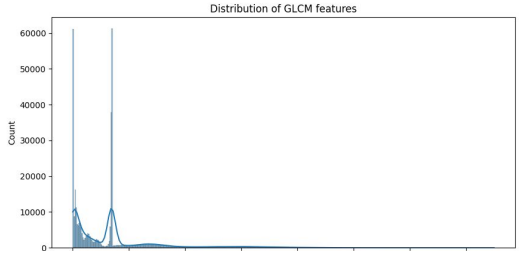
Feature Visualization (#Challenge 1)

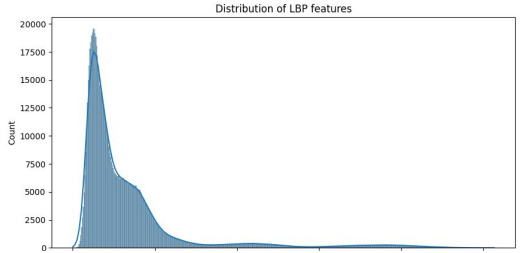


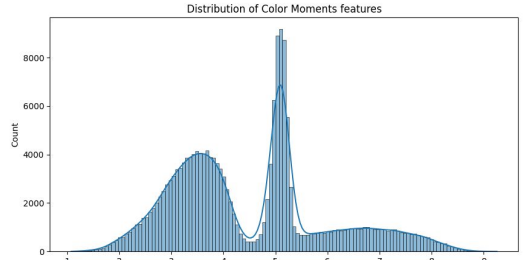




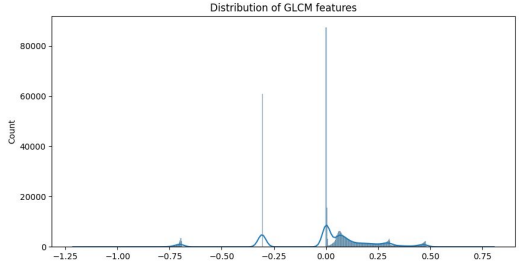
Extracted features

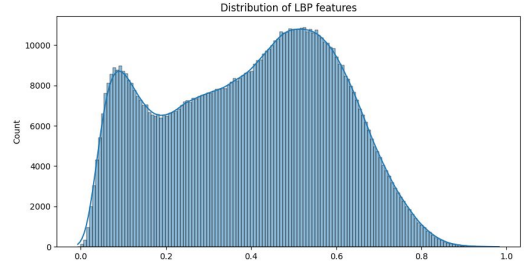


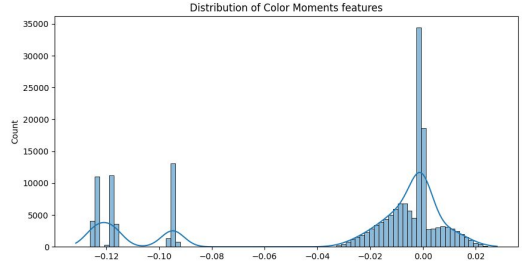




After log-transformation

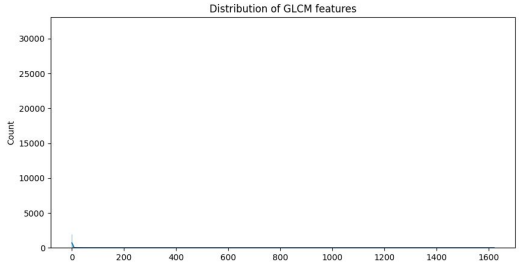


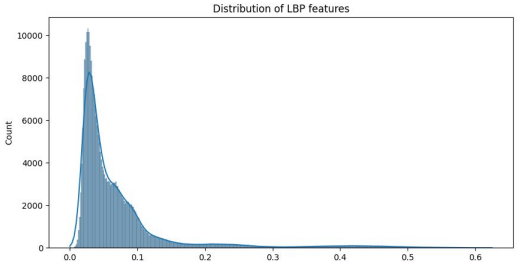


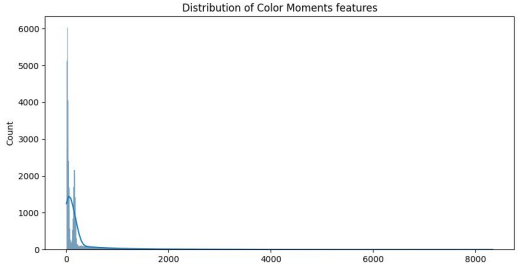


After min-max scaling

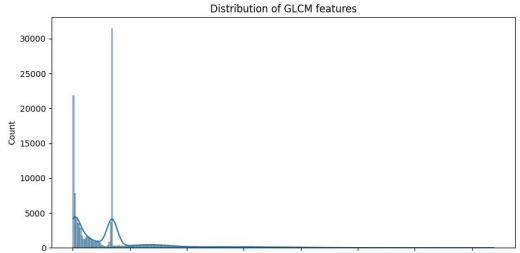
Feature Visualization (#Challenge 2)

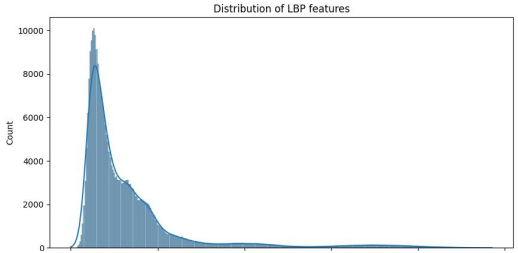


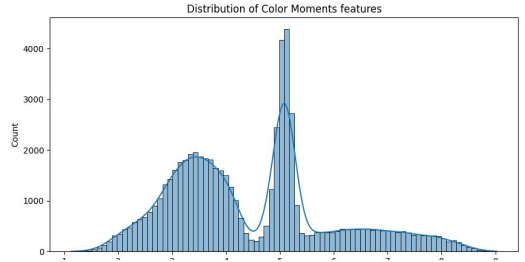




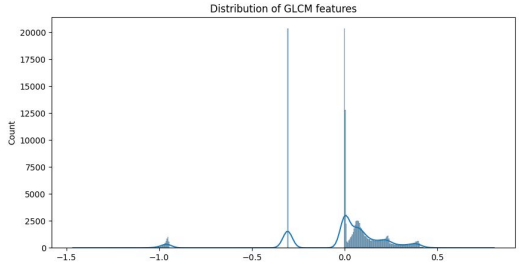
Extracted features

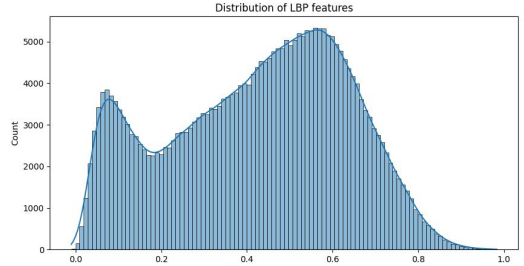


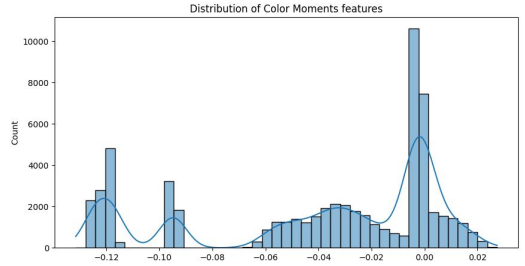




After log-transformation







After min-max scaling

Machine Learning Models

Single Classifiers
Random Forest KNN XGBoost LightGBM SVM CatBoost

Ensemble
Majority Voting among classifiers using the best parameters obtained from the grid search.

Hyperparameter Tuning

#Challenge 1

We used Grid Search to find the best parameters, using a subset of the training set as validation and the provided validation set as test set. The best model is selected based on accuracy.

Models	Best Parameters	Validation accuracy	Test accuracy
Random Forest	max_depth: 20, n_estimators: 300	0.831	0.824
KNN	metric: manhattan, n_neighbors: 9, weights: distance	0.692	0.669
XGBoost	max_depth: 30, n_estimators: 200	0.833	0.826
LightGBM	boosting_type: gbdt, learning_rate: 0.2, num_leaves: 50	0.833	0.822
SVM	c: 10, gamms: 0.1, kernel: rbf	0.510	0.498
CatBoost	depth: 10, iterations: 300	0.825	0.826

Hyperparameter Tuning

#Challenge 2

We used Grid Search to find the best parameters. The best model is selected based on accuracy and Kappa score.

Models	Best Parameters	Validation accuracy	Kappa Score	Test accuracy	Kappa Score
Random Forest	max_depth: 20, n_estimators: 200	0.798	0.620	0.795	0.623
KNN	metric: manhattan, n_neighbors: 11, weights: distance	0.752	0.535	0.735	0.508
XGBoost	max_depth: 30, n_estimators: 200	0.829	0.680	0.818	0.665
LightGBM	boosting_type: gbdt, learning_rate: 0.2, num_leaves: 50	0.830	0.682	0.831	0.690
SVM	c: 10, gamma: 0.001, kernel: rbf	0.697	0.409	0.675	0.385
CatBoost	depth: 10, iterations: 300	0.830	0.685	0.827	0.687

Classification Reports

#Challenge 1

Models	Class	Accuracy	Precision	Recall	F1-score
Random Forest	nevus	0.83	0.81	0.86	0.83
	others	0.83	0.84	0.79	0.82
KNN	nevus	0.70	0.70	0.71	0.71
	others	0.70	0.70	0.69	0.69
XGBoost	nevus	**0.84**	**0.83**	**0.86**	**0.84**
	others	**0.84**	**0.85**	**0.82**	**0.83**
LightGBM	nevus	0.83	0.82	0.85	0.84
	others	0.83	0.84	0.81	0.83
SVM	nevus	0.51	0.51	1.00	0.68
	others	0.51	1.00	0.01	0.02
CatBoost	nevus	0.83	0.82	0.85	0.84
	others	0.83	0.84	0.81	0.83

Classification Reports

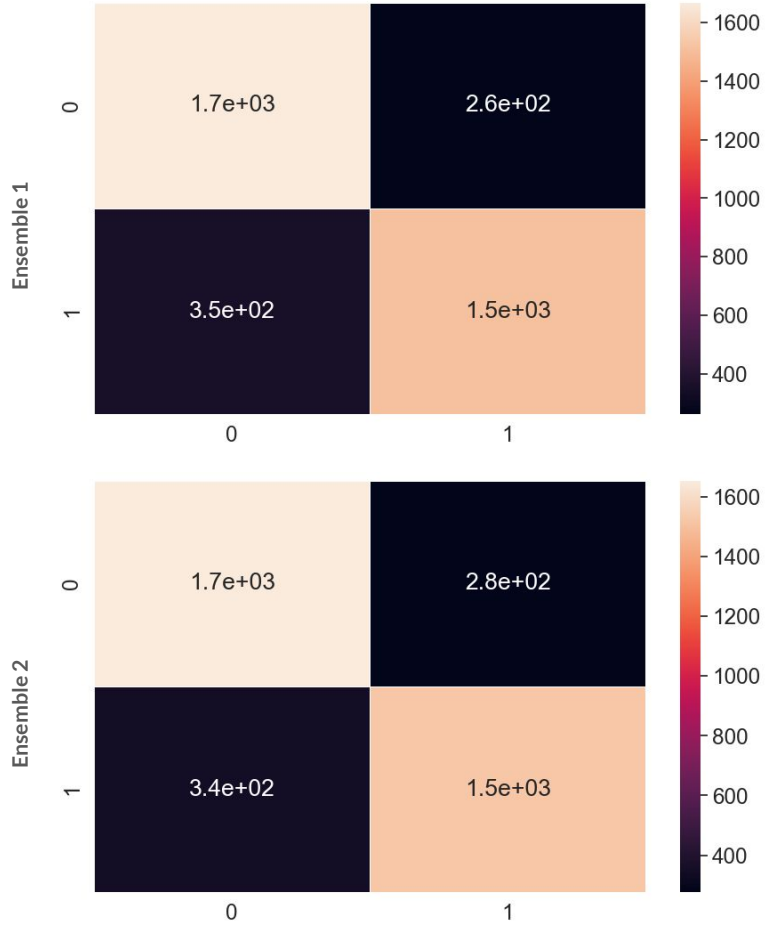
#Challenge 2

Models	Class	Accuracy	Precision	Recall	F1-score	Kappa
Random Forest	bcc	0.81	0.75	0.87	0.80	0.64
	mel	0.81	0.86	0.86	0.86	
	scc	0.81	0.67	0.09	0.15	
KNN	bcc	0.76	0.70	0.81	0.75	0.55
	mel	0.76	0.81	0.82	0.82	
	scc	0.76	0.40	0.04	0.08	
XGBoost	bcc	0.83	0.78	0.89	0.83	0.68
	mel	0.83	0.88	0.89	0.89	
	scc	0.83	0.61	0.12	0.20	
LightGBM	bcc	**0.85**	**0.80**	**0.89**	**0.84**	**0.72**
	mel	**0.85**	**0.89**	**0.89**	**0.89**	
	scc	**0.85**	**0.73**	**0.29**	**0.41**	
SVM	bcc	0.62	0.67	0.67	0.67	0.39
	mel	0.62	0.88	0.62	0.73	
	scc	0.62	0.11	0.34	0.16	
CatBoost	bcc	0.84	0.80	0.87	0.83	0.70
	mel	0.84	0.89	0.89	0.89	
	scc	0.84	0.66	0.31	0.42	

Ensemble of best models

#Challenge 1

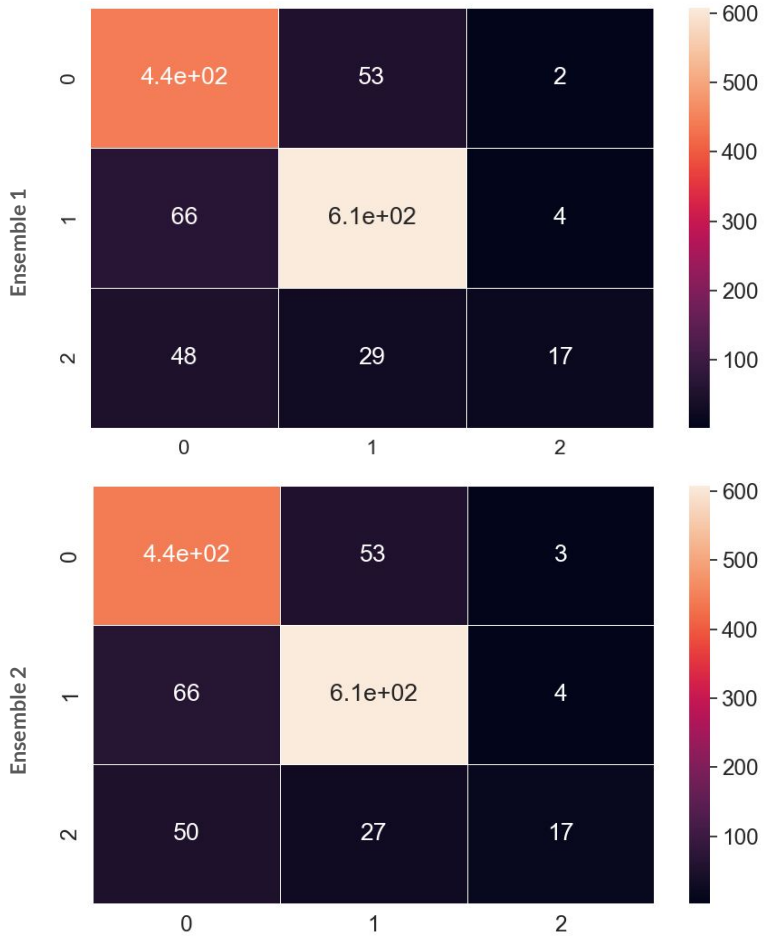
Majority Voting	Precision	Recall	F1 score	Kappa	Acc
LightGBM, XGB, RandomForest	0.8388	0.8382	0.8380	0.6761	0.84
XGB, LightGBM, GradBoost, AdaBosst	0.8369	0.8366	0.8365	0.6730	0.84



Ensemble of best models

#Challenge 2

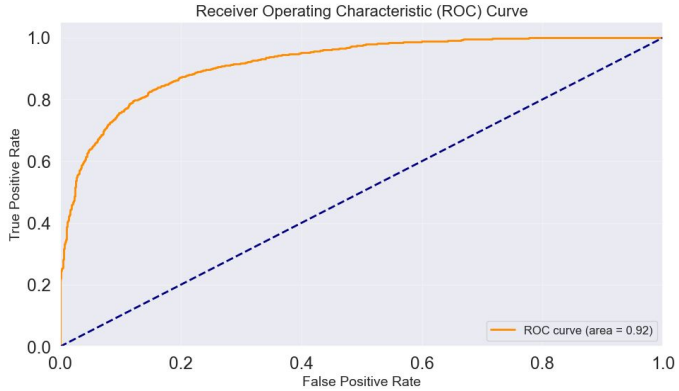
Majority Voting	Precision	Recall	F1 score	Kappa	Acc
LightGBM, XGB, RandomForest, CatBoost	0.8369	0.8409	0.8253	0.7036	0.84
XGB, LightGBM, RandomForest	0.8348	0.8401	0.8248	0.7023	0.84

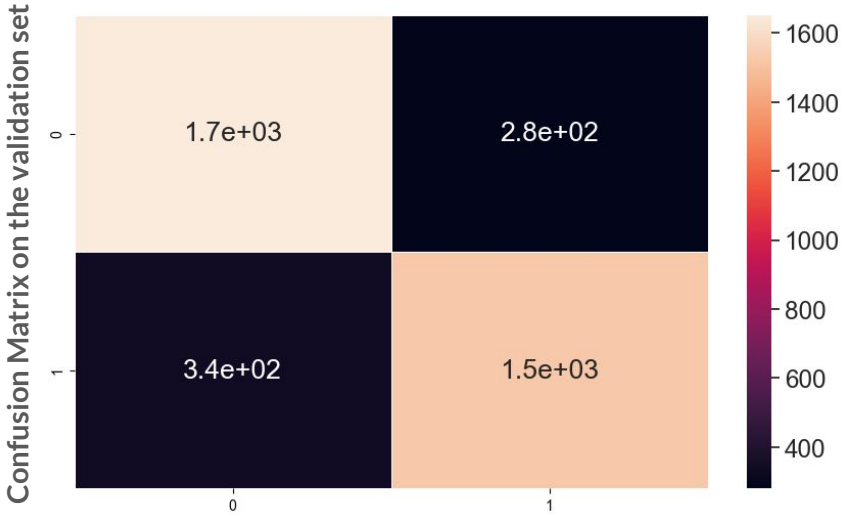


Best Classifier Result

#Challenge 1

XGBoost
Accuracy: 0.84

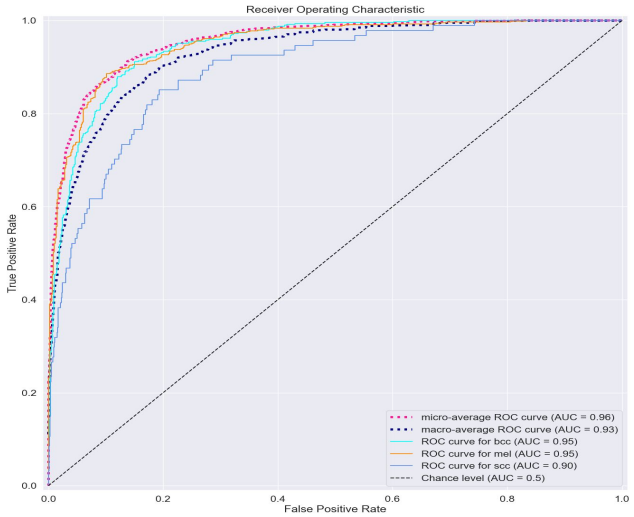


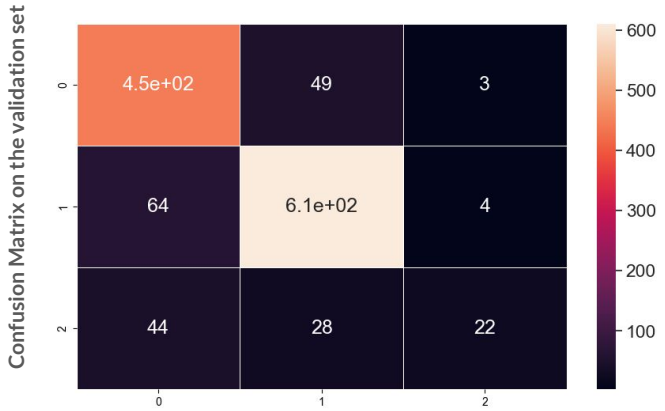


Best Classifier Result

#Challenge 2

LightGBM
Accuracy: 0.85
Kappa: 0.72






Thank U