**Machine Learning Assignment**

진 수 정

2017311974

통계학과

1. **Problem Definition**

주어진 문제는 각각의 비디오에 등장하는 사람의 음성, 시각 정보를 통해 해당하는 사람이 우울증을 가지고 있는지, 가지고 있지 않은지를 예측하는 Binary Classification 문제이다.

1. **Data**
   1. **Data Description**

2.1.1. train.csv

각종 기계학습 모델에 학습을 시키기 위한 데이터셋으로, 비디오로부터 추출된 음성, 시각 정보를 포함하고 있다. 405 \* 384 사이즈의 데이터프레임 형태이며, 피처들은 음성, 시각 정보의 평균, 분산, 최소값, 최대값 등의 numeric data로 구성되어 있다. 타겟은 우울증 여부를 나타내는 binary data이며, 타겟의 각 클래스의 비율이 크게 차이가 나지 않으므로 데이터 불균형은 존재하지 않는다 (class 0: 54%, class 1: 46%). 결측치는 없으므로 imputation 등의 작업은 필요하지 않다. 다만 피처의 개수가 383개로 많기 때문에 추후에 변수 선택이나 PCA를 통한 차원 축소 등의 작업이 필요할 수 있다.

2.1.2. X\_test.csv

학습시킨 모델을 이용하여 최종적으로 예측을 수행하고자 하는 데이터셋으로, 100 \* 383 사이즈의 데이터프레임 형태이다. 피처의 개수와 형태는 train 데이터셋과 동일하며, test 데이터셋에는 타겟 변수가 누락되어 있다.

* 1. **Data Pre-processing**
     1. Variable filtering by uniqueness

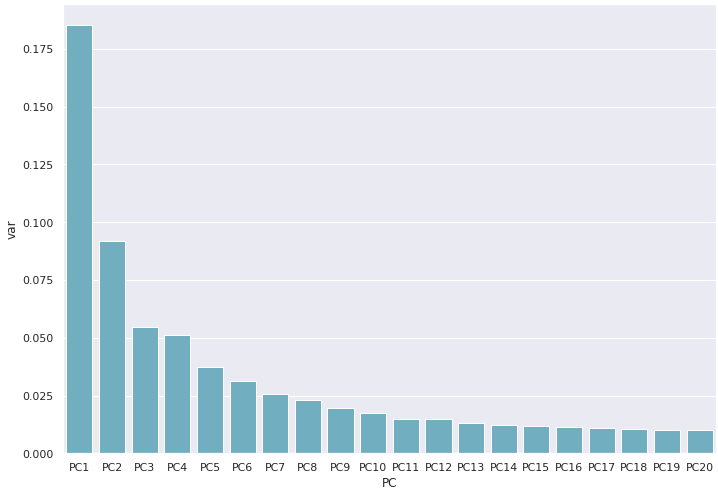
피처의 유니크 값 개수를 카운트하여, 하나의 값으로만 이루어진 7개의 피처를 확인하였다 (number\_faces\_var, number\_faces\_min, emotion\_disgust\_min, F0semitoneFrom27.5Hz\_sma3nz\_stddevRisingSlope\_min, StddevVoicedSegmentLengthSec\_min, StddevUnvoicedSegmentLength\_min, number\_faces\_max). 모든 행에 대해 같은 값을 갖는 해당 피처들은 아무런 정보를 포함하고 있지 않으므로 제거하였다.

* + 1. Variable filtering by correlation coefficient

피처들이 높은 상관관계를 가지고 있으면, 한 피처가 다른 피처에 의해 설명이 되므로 해당 피처를 삭제하여도 설명력에 크게 차이가 나지 않는다. 더불어, 로지스틱 회귀에서 피처들의 상관관계가 높으면 다중공선성 문제가 발생하여 예측력이 저하될 수 있다. 따라서 차원 축소와 성능 향상을 위해 높은 상관관계를 가지고 있는 일부 피처를 제거해주기로 하였다.

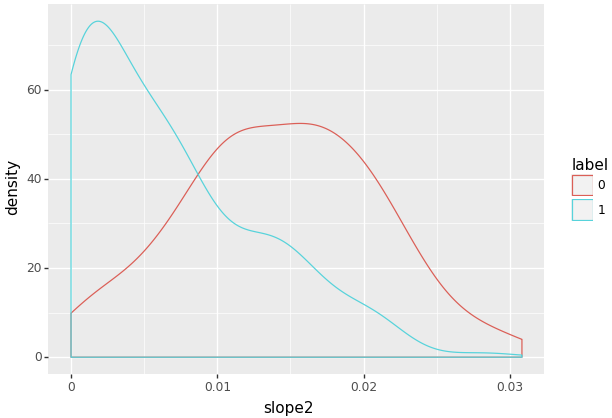
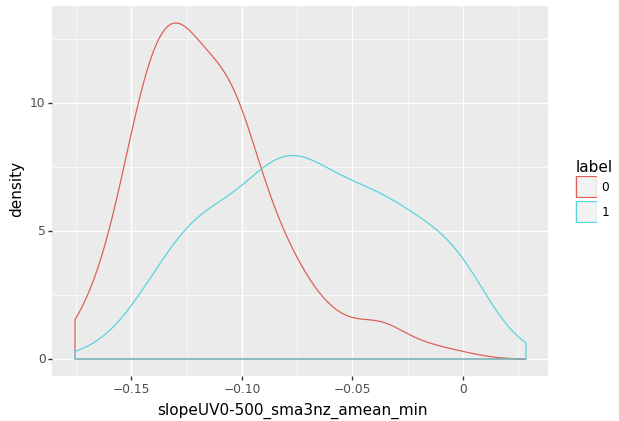
주어진 데이터에서 모든 피처들이 수치형 데이터이므로 모든 피처들 간의 피어슨 상관계수를 계산하였다. 상관계수 절댓값이 0.9보다 높은 변수쌍에 대하여, 라벨과의 상관계수가 더 낮은 변수를 제거하였다. 그 결과 피처의 개수가 376개에서 192개로 줄었으며, 이를 통해 모델링 시간 감축 또한 기대해볼 수 있다.

* + 1. Feature engineering by PCA

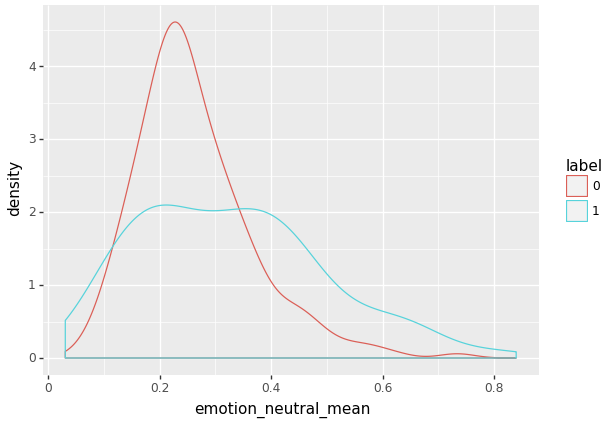


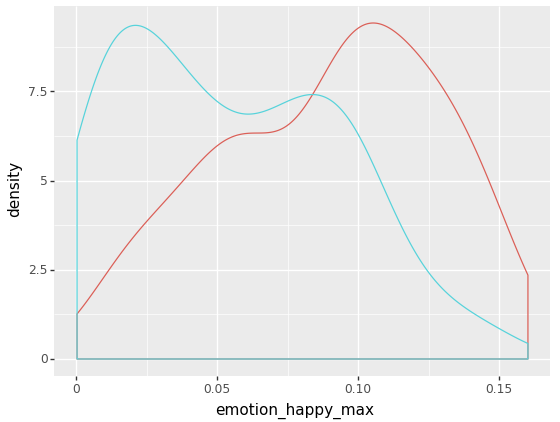
[Plot 1] Scree plot for PCA on the Feature Dataset (Train+Test)

Train set과 Test set의 피처들에 대하여 정규화를 시켜준 다음 PCA를 수행하였다. Scree plot에서 Elbow point가 2이므로, 2개의 주성분만 사용하였다. 이 때 2개의 주성분으로 설명되는 데이터의 분산이 28% 정도밖에 되지 않으므로, 데이터를 2개의 주성분으로 축소하기에는 무리가 있었다. 하지만 2개의 주성분을 통해 기존의 피처로는 설명할 수 없었던 데이터의 일부가 설명될 수도 있다고 판단하여, 2개의 주성분(PC1, PC2)을 기존 데이터셋에 새롭게 추가하였다.

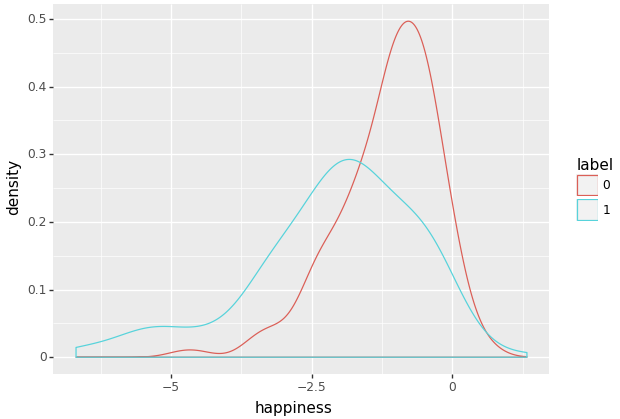
1. **Analysis**
   1. Feature engineering by variable transformation

[Plot 2] Density plot of 1st order (left) and 2nd order (right) of ‘slopeUVO-500\_sma3nz\_amean\_min’ by Label class

라벨과의 상관관계가 높았던 피처 'slopeUV0-500\_sma3nz\_amean\_min’에 대하여 라벨별로 'slopeUV0-500\_sma3nz\_amean\_min’의 분포를 시각화 해보았다 (왼쪽). 시각화 결과, 라벨별로 피처의 분포가 잘 구분되므로 해당 피처는 라벨 분류에 효과적인 피처임을 파악하였다. 또한 해당 피처를 제곱한 값에 대해서도 시각화를 진행해본 결과 (오른쪽) 라벨별로 피처의 분포가 잘 구분되므로, 라벨 분류에 도움이 될 수 있을 거라 판단해서 기존 데이터셋에 추가하였다(slope2).



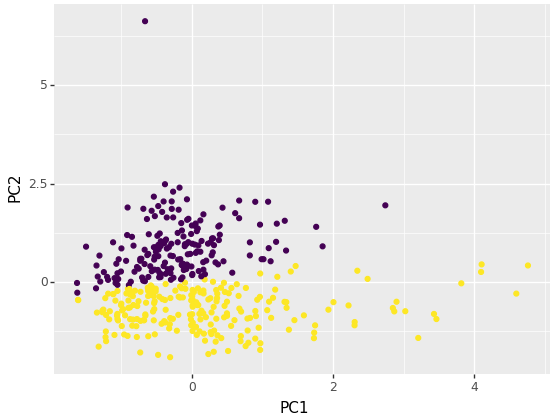
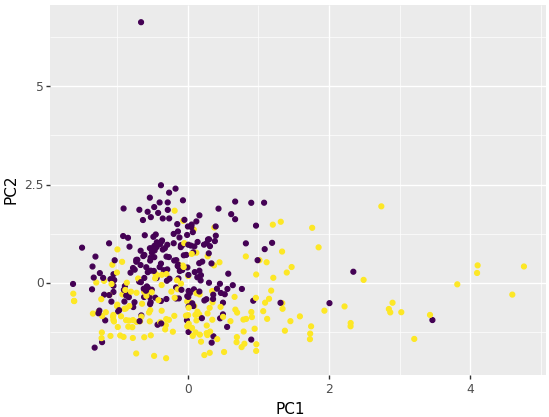
[Plot 4] Density plot of ‘emotion\_happy\_max’ (left) and ‘emotion\_neutral\_mean’ (right) by Label class



[Plot 5] Density plot of log( ‘emotion\_happy\_max’ / ’emotion\_ neutral\_mean’) by Label class

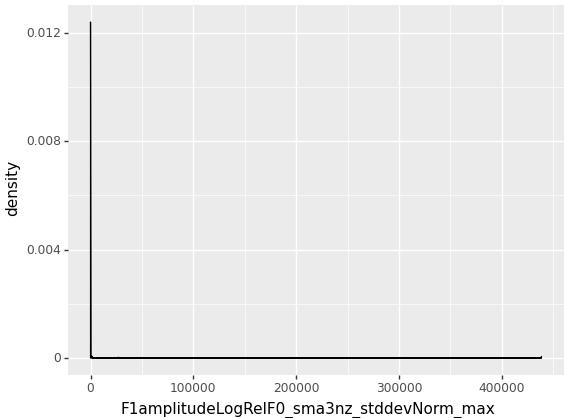
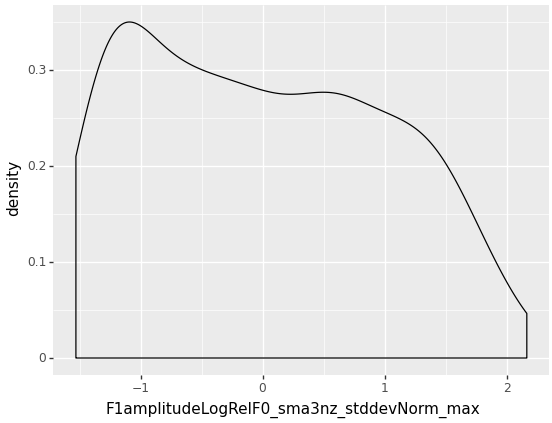
시각화 결과, 라벨 1은 라벨 0에 비해 emotion\_happy\_max 값이 작으며, emotion\_neutral\_mean 값이 큰 부분에 많이 분포하고 있는 것을 확인하였다 (Plot 4). 이를 이용하여 emotion\_happy\_max 값을 emotion\_neutral\_mean으로 나누어 로그를 취한 값에 대해서도 시각화를 진행해본 결과 (Plot 5), 라벨 0이 라벨 1에 비해 해당 피처의 값이 큰 부분에 집중해서 분포하고 있는 것을 확인할 수 있었다. 해당 피처 역시 기존 데이터셋에 추가해주었다(happiness).

* 1. Feature engineering by clustering



[Plot 5] Scatter plot coloring with Label class (left) and Cluster group (right)

PC1과 PC2 축에 대해 데이터의 라벨 값을 시각화 해본 결과, 데이터가 일부 overlapped 된 부분이 있지만 대체로 두 개의 그룹으로 잘 구분되는 것을 확인하였다 (왼쪽). PC1과 PC2를 이용하여 K-means 클러스터링을 진행한 후에 클러스터링 결과를 시각화 해본 결과 (오른쪽), overlapped 된 부분에서는 클러스터링 결과와 실제 라벨이 일치하지 않기도 하였지만, 클러스터링 결과가 라벨과 대략 72% 정도 일치하는 것을 확인할 수 있었다. 따라서 클러스터링 결과도 라벨 분류에 도움이 될 수 있을 거라 판단하여 하나의 피처로 추가하였다(clus).

* 1. Yeo-johnson Transformation

[Plot 6] Density plot of ‘F1amplitudeLogRelF0\_sma3nz\_stddevNorm\_max’ before (left) and after (right) Yeo-johnson Transformation

피처들의 4분위수 통계 정보와 시각화 결과를 통해 데이터에 highly skewed 된 피처들이 많은 것을 확인하였다 (왼쪽). 따라서 전체 데이터에 Yeo-johnson 변환을 시켜주어 피처들의 분포가 고르게 분포되어 있도록 변환시켰다 (오른쪽). 이어지는 모델링 파트에서 Yeo-johnson 변환 전후 데이터를 비교하여 더 높은 성능을 내는 데이터를 최종 데이터로 선정할 것이다.

1. **Model**
   1. **Classifier**
      1. Logistic Regression

선형 회귀 방식을 기반으로 하되 시그모이드 함수를 이용해 분류를 수행하는 모델이다. 세부적으로 보면, 시그모이드 함수의 최적선을 찾고 시그모이드 반환값을 확률로 간주해, 확률에 따라 분류를 결정하는 방법이다. 로지스틱 회귀는 가볍고 빠른 모델이지만, 복잡한 문제의 경우에는 다른 모델의 성능이 더 좋을 수 있다.

로지스틱 회귀의 경우 objective function에 L1 패널티를 추가한 라쏘를 적용할 수 있으며, 이를 통해 적절한 피처만을 로지스틱 회귀에 포함시키는 변수 선택을 수행할 수 있다. 그러므로 로지스틱 모델에 대해서는 라쏘, forward selection 두 가지 방법을 적용하여 변수 선택을 진행하였다.

* + 1. Random Forest

Bootstrap으로 여러 개의 subset data를 샘플링한 후에, 개별적인 트리가 각각의 데이터셋에 대해 학습을 수행한 뒤 보팅을 통해 최종적인 예측을 하는 알고리즘을 말한다. 전반적으로 다른 머신러닝 알고리즘보다 뛰어난 성능을 보이는 경우가 많다.

* + 1. LGBM

XGBoost의 느린 학습속도를 극복하기 위해 개발된 알고리즘으로, 학습에 걸리는 시간이 XGBoost보다 훨씬 적으며 메모리 사용량도 상대적으로 적으면서 XGBoost와 마찬가지로 뛰어난 예측 성능을 제공한다. 일반 GBM 계열의 트리 분할 방법과 다르게 leaf-wise 트리분할 방식을 사용함으로써 최대 손실값을 가지는 리프노드를 지속적으로 분할하기 때문에, 일반적인 균형 트리 분할 방식보다 예측 오류 손실을 최소화할 수 있다.

* + 1. SVC

training observation과 hyperplane 간의 최소거리인 margin을 최대로 하는 hyperplane을 찾는 문제이다. 보통 예측 정확도가 높으며 soft margin을 사용하여 non-separable한 케이스도 해결할 수 있지만, 연산이 많기 때문에 학습 속도가 느리다.

* + 1. GradientBoostingClassifier

여러 개의 개별적인 트리를 묶어 강력한 모델을 만드는 앙상블 기법이다. 이때, Random Forest와 달리 이전 트리의 오차를 보완하는 방식으로 트리를 만든다. 상대적으로 Random Forest보다 파라미터 설정에 민감한 편이지만, 적절한 파라미터를 잘 조정했을 때에 높은 정확도를 제공한다.

* 1. **Design Consideration**
     1. 데이터셋 선정

가장 컴퓨팅 속도가 빠른 로지스틱 모델로 Yeo-johnson 변환 전후 데이터셋을 학습시켜 성능을 비교해보았다. Yeo-johnson 변환 전후 데이터셋의 정확도는 각각 0.7377, 0.7541로, Yeo-johnson 변환을 한 데이터셋의 성능이 더 높았다. 따라서 모델링에 이용할 데이터셋을 Yeo-johnson 변환을 한 데이터셋으로 선정하였다.

* + 1. Feature selection

데이터셋에 피처들의 개수가 여전히 많은 것을 확인할 수 있는데, SVC나 Random Forest와는 달리 특히 로지스틱 모델의 경우 피처들이 highly correlated 되어 있을 때 예측이 제대로 이루어지지 않을 수 있다. 또한 SVC나 Random Forest의 경우에도 피처들을 적절히 선별하였을 때 더 높은 성능을 낼 수도 있다. 따라서 파라미터 튜닝을 마친 후에 RFE를 통한 변수 선택 과정을 거쳐 최종 데이터셋을 선정하였다.

후의 모델 앙상블 과정을 진행하기 위해 데이터의 통일성이 필요하므로, 메인으로 사용할 모델인 Random Forest의 RFE 변수 선택 결과를 최종 데이터셋으로 사용하였다.

1. **Experiments**
   1. **Settings**
      1. Split Train, Validation set

Train 데이터를 Train set과 Validation set으로 분할할 때, 라벨 클래스의 비율을 비슷하게 유지하기 위해서 층화추출을 통해 데이터를 분할하였다. 또한 Train set과 Validation set의 비율은 7:3으로 정하였다. 파라미터 튜닝 과정에서는 전체 데이터를 활용하였으며, validation accuracy를 산출할 때에만 나뉜 Train set, Validation set을 활용하였다.

* + 1. 파라미터 선정 방식

전체 데이터셋에 대해 5 fold cv를 진행하여 최적의 파라미터를 선정하였다. 튜닝할 파라미터가 많은 LGBM은 그리드 서치의 반복 횟수를 줄이기 위해, 먼저 모든 파라미터에 대해 넓은 범위에서 베이지안 최적화를 하여 최적 파라미터의 대략적인 범위를 파악하였다. 이후 그 근방에 대하여 보다 좁은 범위에서 그리드 서치를 진행하여 최적 파라미터 위치를 파악하였다. 나머지 모델(Logistic Regression, SVC, Random Forest, Gradient Boosting Classifier)에 대해서는 Logistic Regression, SVC)은 넓은 범위에서 그리드 서치를 진행하여 최적 파라미터 위치를 파악하였다.

* + 1. Feature selection

5.1.2에서 구한 최적의 파라미터로 적합시킨 Random Forest 모델에 대해 RFE(피처개수 20개, 40개, 60개, 80개, 100개)를 실행하여 가장 높은 성능을 내는 적절한 피처 개수, 최적의 피처 조합을 탐색하였다.

* + 1. 학습 프로세스

Yeo-johnson 변환 전후 데이터셋에 대해 가장 실행 속도가 빠른 Logistic Regression을 적용하여, 더 높은 성능을 낸 Yeo-johnson 변환 데이터셋을 모델링에 사용하였다. 해당 데이터셋으로 Grid search with 5 fold cv를 통해 Random Forest 모델의 최적의 파라미터를 파악하였다. 구한 최적의 파라미터로 RFE를 진행하여 가장 높은 성능을 보인 최적의 피처 조합 60개를 최종 데이터셋으로 결정하였다. 이후 나머지 4개의 모델(Logistic Regression, SVM, LGBM, Gradient Boosting Classifier)에 대해서도 최종 데이터셋에 5-fold cv를 통해 최적의 파라미터 조합으로 튜닝시켰다.

각각의 5가지 모델(Logistic Regression, Random Forest, LGBM, SVM, GradientBoostingClassifier)에 대해 최적 파라미터에서의 validation accuracy를 비교하여 가장 높은 정확도를 보인 Random Forest를 최종 모델을 결정하였다. 또한 1st level model을 5가지 모델로, 2nd level model을 Random Forest로 두고 stacking도 진행하였다. 그 후, 가장 높은 성능을 보인 모델로 전체 train set을 학습시켰다.

* 1. **Performance Metrics**
     1. Accuracy

전체 경우에 대하여 옳게 예측한 경우의 비율을 나타내는 값으로, 직관적이기 때문에 보편적으로 사용되는 지표이다. 추가적으로 클래스 불균형이 존재할 경우 large class를 잘 예측하는 방향으로 학습이 되며 small class는 잘 예측하지 못하는 경우가 생기기 때문에, F1-score 등 다른 평가 지표가 사용되기도 한다. 주어진 문제의 경우 클래스 불균형은 없었으므로 Accuracy는 평가지표로써 적절하다.

* 1. **Results**

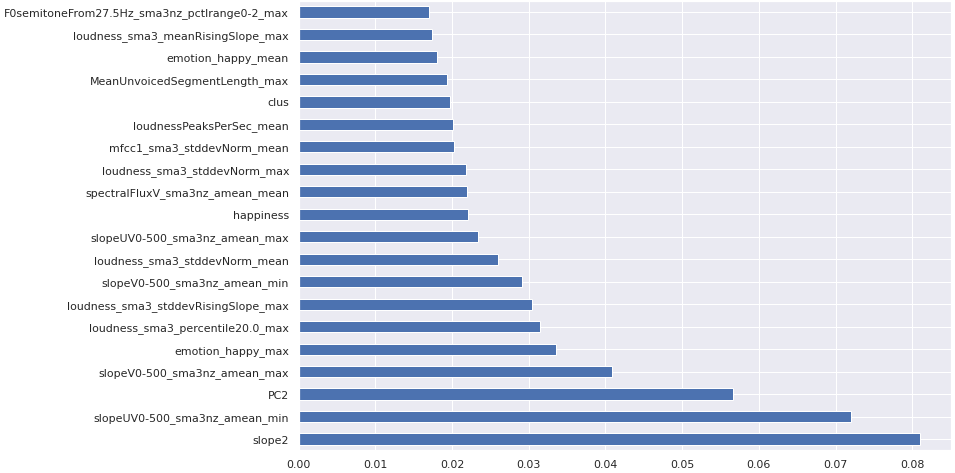
|  |  |  |
| --- | --- | --- |
| Best Model | Training Accuracy | Validation Accuracy |
| Logistic Regression | 0.823322 | 0.844262 |
| Random Forest | 0.961131 | 0.860656 |
| LGBM | 0.996466 | 0.803279 |
| SVC | 0.915194 | 0.860656 |
| GradientBoostingClassfier | 1.000000 | 0.827869 |
| Stacking | 0.795053 | 0.844262 |

[Table 1] Training Accuracy (split 70%) and Validation Accuracy (split 30%) obtained with 6 models(Logistic Regression, Random Forest, LGBM, SVC, Gradient Boosting Classifier, Stacking) with optimal parameters

LGBM, Gradient Boosting Classifier에서 max\_depth 등의 파라미터의 범위를 overfitting을 방지하기 위해 적절히 선택하였으나, overfitting 문제가 완전히 해결되지 않은 것을 짐작할 수 있다. 또한 model stacking의 결과 validation accuracy가 향상되지 않았으므로 단일 모델을 사용하는 것이 낫다고 판단하였다.

따라서 가장 높은 정확도를 보인 Random Forest를 최종 모델로 선정하였다. 그리드 서치 결과, 최적의 파라미터 조합은 (criterion: ‘gini’, max\_depth: 5, max\_features: ‘sqrt’, min\_samples\_leaf: 2, min\_samples\_split: 4, n\_estimators: 800)으로 구할 수 있었다. 전체 Train 데이터셋을 최적의 파라미터 조합에서 학습시킨 Random Forest 모델은 Test 데이터셋에 대해 85.1485의 정확도를 보여주었다 (리더보드 결과).

1. **Discussion and Limitation**



[Plot 7] Random Forest Feature Importances plot

* ‘slope2’, ‘slopeUVO-500\_sma3nz\_amean\_min’, ‘PC2’, ‘slopeVO-500\_sma3nz\_amean\_max’ 등의 피처가 Gini Impurity를 크게 줄임으로써 예측에 있어서 중요한 피처로 작용했다는 것을 알 수 있다.

- 활용방안: 우울증은 슬픔, 외로움 등의 감정에 빠지게 하며, 업무에 집중을 하지 못하도록 집중을 저하시키는 등 부정적인 영향을 미칠 수 있다. 따라서 개인이 우울증을 앓고 있는지 판별하는 작업은 중요한데, 위 과제에서 구현한 Depression Detection 알고리즘을 적용해볼 수 있을 것이다. 비디오를 시청하고 있는 사용자의 표정이나 음성 데이터를 웹캠이나 오디오를 통해 실시간으로 제공받고, Depression Detection 알고리즘을 적용해 봄으로써 우울증을 판별하고, 판별 결과와 여러 통계결과를 사용자에게 제공해주는 시스템으로 구현해볼 수 있을 것이다. 또한 Feature Importances 결과를 활용하여 우울증과 관련이 있는 요인에는 어떤 것들이 있는지 등과 같은 인사이트를 도출할 수 있을 것이라 예상한다.

- 한계점: 목표로 했던 정확도까지 끌어올리지 못했으며 85% 정도의 정확도에 그치는 결과를 냈다는 점이 아쉬웠다. 추가적으로 피처의 개수가 너무 많아 개별적인 피처에 대해 탐색하기에는 무리가 있었다.

1. **References**

권철민, 『파이썬 머신러닝 완벽 가이드』, 위키북스, 2019.