MLDL: Assignment 2 Report

신성구
데이터사이언스학과, 2022-22047

1. Multiple Model Training

(a) Train multiple models that you learned in class. Clearly explain what models are used. Report the training results. What metric did you use? How do different models perform on the training data?

1) Used Models

Model Train에 사용된 모델들은 다음과 같다.

1. SVM (linear kernel)
2. SVM (radial kernel)
3. Random Forest
4. LDA
5. QDA
6. XGBoost
7. LGBM
8. Logistic Regression

Training은 다음 네 가지 Case로 진행된다.

1. NO Dimensionality Reduction (full features)
2. With Dimensionality Reduction (PCA)
3. With Feature Selection (No PCA)
4. With Feature Selection + PCA

다각적인 분석과 Best Model의 선택을 위해 Classification 문제를 해결할 수 있는 대부분의 Machine Learning 모델을 시도하였으며, 또한 변수 선택이나 차원 축소의 효과를 확인하기 위해 (X,X), (X,O), (O,X), (O,O)의 네 가지 경우에 대하여 후보 모델들을 모두 적합하였다.

방법론: 각각의 case마다 모델 별로 GridSearchCV를 통해 optimal한 hyperparameter를 구한다. 다만 StandardScaler나 PCA, Feature Selection을 "OnlindAd_X_train.csv" 데이터에 적용한 후에 Grid Search나 Cross-validation을 하면 data leakage가 발생하므로, 이를 handling하기 위하여 scikit-learn이 제공하는 Pipeline 라이브러리를 활용하였다. 차원 축소와 변수 선택의 구체적인 방법은 다음과 같다.

1. 차원 축소: PCA로 X_train을 변환하였을 때, 'explained_variance_ratio'의 누적 그래프는 n_components=2에서 elbow가 생긴다. 비록 설명되는 분산은 낮지만, 이후에는 n_components가 1 증가할 때마다 설명되는 분산이 미미하게 단조증가하므로 큰 분산을 가지는 시점의 n_components는 원래 feature의 수와 비슷하다. 따라서 tradeoff로서 n_components=2로 PCA를 수행하였다.

2. 변수 선택: scikit-learn이 제공하는 SelectFromModel(base estimator = RandomForestClassifier()) 을 활용하였다. 해당 Feature Selector는 base estimator인 랜덤포레스트 모델이 중요하게 간주하는 변수를 추려내는 역할을 한다.

각 Case에 대하여 Optimal한 파라미터들을 구한 이후에는 해당 파라미터들을 사용해 모델들을 적합하고, 그 때의 Train-Accuracy와 Cross-Validation을 통한 Estimated-Test-AUC, 그리고 Estimated-Test-Accuracy의 값을 저장한다. 저장한 값들을 기반으로 train과 test의 성능(추정)을 비교하며, 그 중 가장 Best Model을 선정하여 X_test 데이터에 대한 Prediction을 생성해 제출한다.

2) Metric & Training Results

Metric으로는 Accuracy 와 AUC Score를 사용하였다. Accuracy만으로는 recall이나 precision 등의 정보를 반영할 수 없기 때문에 AUC Score를 구하여 모델을 복합적으로 평가하였다.

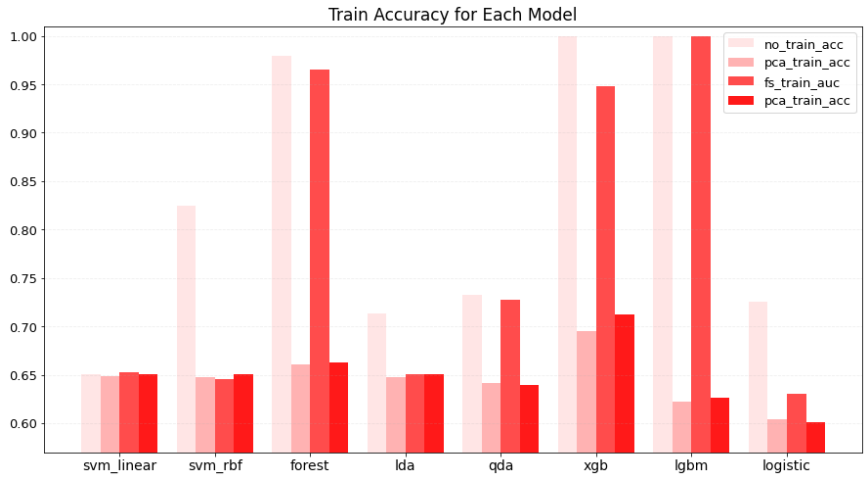
```
params_svm_l = {'svm_linear__C':np.logspace(-3,2,10).tolist()}
params_svm_rbf = {'svm_rbf__C':np.logspace(-3,2,10).tolist(),
                  'svm_rbf__gamma':np.logspace(-3,2,10).tolist()}
params_rf = {'forest__n_estimators':[100,500,1000], 'forest__min_samples_leaf':[5,10,15,20],
             'forest__criterion':['gini','entropy','log_loss']}
params_lda = {'lda__n_components':[1,2]}
params_qda = {'qda__reg_param':[0.1, 0.2, 0.3, 0.4, 0.5]}
params_xgb = {'xgb__n_estimators':[100,500,1000], 'xgb__learning_rate':[0.001,0.005,0.01,0.05,0.1]}
params_lgbm = {'lgbm__n_estimators':[100,500,1000], 'lgbm__learning_rate':[0.001,0.005,0.01,0.05,0.1],
               'lgbm__max_depth':[2,5,10,15]}
params_logistic = {'logistic__penalty':['l2','none']}

params = [params_svm_l, params_svm_rbf, params_rf, params_lda, params_qda, params_xgb, params_lgbm, params_logistic]
```

Picture 1. Hyperparameter Grid Settings

위의 Picture 1과 같이 Hyperparameter Grid를 설정하고, Grid Search를 통해 구한 optimal parameter로 적합한 모델들의 Training 결과는 아래와 같다.

Picture 2. Training Accuracy for Each Model & Case

Picture 2는 각 케이스 별 모델들의 Training Accuracy를 나열한 그래프이다. 우선 PCA를 했을 때와 하지 않았을 경우를 비교하면, PCA를 하지 않았을 때가 전반적으로 더 높은 Accuracy를 보인다(그래프의 y축 scale이 다른 것에 유의). 이는 PCA를 수행할 때 n_components=2로 설정하였고, 그 때 설명되는 분산이 높지 않아서 발생한 것으로 추정된다. 따라서 Training에 한해서는 PCA(n_components = 2)를 수행하지 않는 것이 더 유리하다고 할 수 있을 것이다. 다만, PCA를 적용하지 않은 경우 RandomForest, XGBoost, LGBM과 같은 트리 기반의 모델들은 accuracy가 1에 육박하는데, 이는 높은 정확도와는 별개로 Overfitting의 위험성이 있다는 점을 기억해야할 것이다. 또한 추가적으로는 Logistic Regression 모델의 training accuracy는 모든 case에서 낮은 값을 보이는 것도 확인할 수 있었다.



Picture 3. 모델 별 Training Accuracy 집계

위의 Picture 3는 모델 별 Training Accuracy를 집계한 그래프이다. Train Accuracy는 대체적으로 트리 기반의 모델들이 강세를 보였다. 또한 앞서 살펴보았듯 PCA(n_components=2)를 적용하지 않은 경우의 정확도가 대체로 높다는 것도 한눈에 확인할 수 있다. 또 추가적으로 Feature Selection을 적용했을 때에는 다른 경우들과 비슷한 정확도를 보이거나, 트리 기반의 모델들은 아무 처리도 하지 않은 경우(가장 정확도가 높은 경우)에 버금가는 높은 Training 정확도를 보이기도 하였다. 그 이유는 Feature Selector의 base_estimator가 RandomForestClassifier 모델이기 때문인 것으로 추정하였다.

(b) Do you think dimension reduction on features (or feature selection) is needed? If so, provide analysis on which features may be important. If not, please justify your answer.

Train 결과에 따르면, train 단계에서는 PCA(n_components=2)를 적용하지 않는 것이 전반적인 train 정확도를 높이고 있었다. 그 이유는 Picture 2에 대한 설명에서 설명하였듯이 n_components=2로서 약 250개에 달하는 큰 feature set을 2개의 축으로 축소시켰고, 또 해당 축들이 설명하는 분산이 높지 않기 때문이라고 추정하였다.

다만 Feature Selection의 경우에는 선택을 했을 때와 그렇지 않았을 때의 훈련 정확도가 전반적으로 비슷하였다. 따라서 비슷한 성능을 유지하면서 계산해야 할 feature의 개수는 줄이자는 본래의 목적에 부합하므로 Feature Selection은 필요하다는 결론을 내렸다.

변수를 선택하는 기준은 base estimator인 RandomForest 모델에 기반하고, 별도로 기준을 지정하지 않았으므로, scikit-learn이 default로 지정한 train data(Cross-validation의 경우, validation set을 제외한 훈련 데이터)에 대하여 tree를 split할 때 감소하는 Gini 불순도의 평균이 큰 변수들이 선택되었을 것이다.

2. Best Model Selection & Prediction

(a) Report the estimated test performance for your best model. Provide a reason for your choice of a model among the models you considered.

1) Model Selection

모델을 선택하기에 앞서, SVM(Linear)와 LDA 모델들은 Accuracy로 평가하면 상위권에 속하였으나, prediction의 클래스 값으로 0 or 2 만을 내보내는 경우가 많다는 것을 확인하였다. 이는 모델이 linear boundary를 가지거나, X 데이터의 클래스 분포가 불균형(클래스 0:약 800개, 클래스 1:약 200개, 클래스 3:약 300개) 하기 때문인 것으로 생각된다. 또한 PCA(n_components=2)를 통해 차원을 축소하지 않은 경우, QDA 모델을 fitting할 때 입력변수 간 공선성 문제로 인한 경고메시지를 확인할 수 있었다. 따라서 Model Selection 단계에서는 이들 모델을 후순위로 두고, 나머지 후보들 중에서 Accuracy와 AUC Score를 고려하여 모델을 선택하였다.

Picture 4. Estimated Test Accuracy, AUC score of each model for each case

Table 1. Train Accuracy

	svm_linear	svm_rbf	forest	lda	qda	xgb	lgbm	logistic
no_train_ACC	0.650138	0.824380	0.979339	0.713499	0.732094	1.000000	0.999311	0.725207
pca_train_ACC	0.648760	0.647383	0.661157	0.647383	0.641185	0.694904	0.622590	0.603994
fs_train_ACC	0.652204	0.646006	0.964876	0.650826	0.727273	0.947658	1.000000	0.630854
fs_pca_train_ACC	0.650826	0.650138	0.662534	0.650826	0.639118	0.712121	0.626033	0.600551

Table 2. Estimated Test Accuracy

	svm_linear	svm_rbf	forest	lda	qda	xgb	lgbm	logistic
no_cv_ACC	0.646679	0.648755	0.630831	0.578490	0.537182	0.642529	0.639090	0.508944
pca_cv_ACC	0.649432	0.647366	0.586743	0.646676	0.641844	0.630134	0.593656	0.602595
fs_cv_ACC	0.650120	0.647361	0.632203	0.623249	0.570890	0.639092	0.634947	0.548210
fs_pca_cv_ACC	0.650805	0.649435	0.594322	0.647366	0.641849	0.626690	0.618431	0.599836

Table 3. Estimated Test AUC Score

	svm_linear	svm_rbf	forest	lda	qda	xgb	lgbm	logistic
no_cv_AUC	0.716216	0.688368	0.724344	0.681148	0.655777	0.718046	0.720195	0.656963
pca_cv_AUC	0.709164	0.675215	0.688830	0.709316	0.709453	0.672863	0.699786	0.720416
fs_cv_AUC	0.717140	0.718591	0.731090	0.716206	0.704768	0.716331	0.698446	0.719965
fs_pca_cv_AUC	0.713021	0.684335	0.700732	0.713013	0.708261	0.697338	0.710490	0.721368

Train의 결과(Table 1)와 Picture 4와 Table 1, Table 2를 고려하자면, 트리 기반의 모델들은 1에 가까웠던 train accuracy에 비해 훨씬 낮은 estimated test accuracy, AUC score를 보였다. 이는 해당 모델들이 overfitting 되었음을 뒷받침하는 결과이다. 그에 반해 나머지 모델들은 train accuracy와 estimated test accuracy가 60% 내외를 유지하였는데, train과 test(estimated) 각각에 대하여 성능이 좋다고 할 수 없으므로, 이는 underfitting의 경우라고 할 수 있다.

따라서 필자의 입장에서 fitting된 모델들 가운데 가장 좋은 경우를 고르는 문제는 결국 'underfitting vs overfitting' 가운데서 결정을 내려야 하는 것으로 귀결되었다. 두 가지 경우 모두 피하고 싶은 것들이지만, overfitting보다는 underfitting이 그나마 낫다고 판단하였다. 그 이유는 overfitting의 경우에는 generalized performance가 감소하는 정도의 상한이 없지만, underfitting의 경우에는 그렇지 않기 때문이다. 즉 overfitting된 모델은 오히려 더 신뢰할 수 없는 모델이라는 것이다.

2) Best Model & Estimated Test Performance

결론적으로 지금까지 언급한 기준과 기피해야 할 점들을 토대로, 최종 모델로서 "Feature Selection + PCA를 한 case의 SVM (radial kernel) 모델을 선택하였다. Train과 Test(추정)의 accuracy가 비슷하고, Test(추정) accuracy는 약 64.94%로, 전체 Test(추정) accuracy 중에서 2위에 위치하였다(1위는 SVM -Linear kernel). Test(추정) AUC score 역시 다른 모델들과 비슷한 수준으로 나타났다.

선정된 모델의 Estimated Test Performance는 다음과 같다.

1. Accuracy: approximately 0.649435
2. AUC Score: approximately 0.684335