데이터사이언스를 위한 머신러닝과 딥러닝 I: 과제 2

2021-29725 주상원

문제 1.

(a)

- Using (1) multinomial logistic regression, (2) SGDClassifier, (3) support vector machines (linear and rbf), (4) k-nearest neighbors, (5) random forests, (6) gradient boosting model, (7) xgboost, (8) linear and quadratic discriminant analysis, I tried to implement 3-class classification model to predict the user clicks.
- **With stratified 5-fold Cross-validation (with 5 repetition), parameters were tuned using two metrics**

Firstly, f1_weighted score.

The F1 score can be interpreted as a harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

$$\mathrm{F1} = 2 * \frac{\mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

Weighted F1: Calculate metrics for each label, and find their average weighted by support (the number of true instances for each label). This alters 'macro' to account for label imbalance; it can result in an F-score that is not between precision and recall.

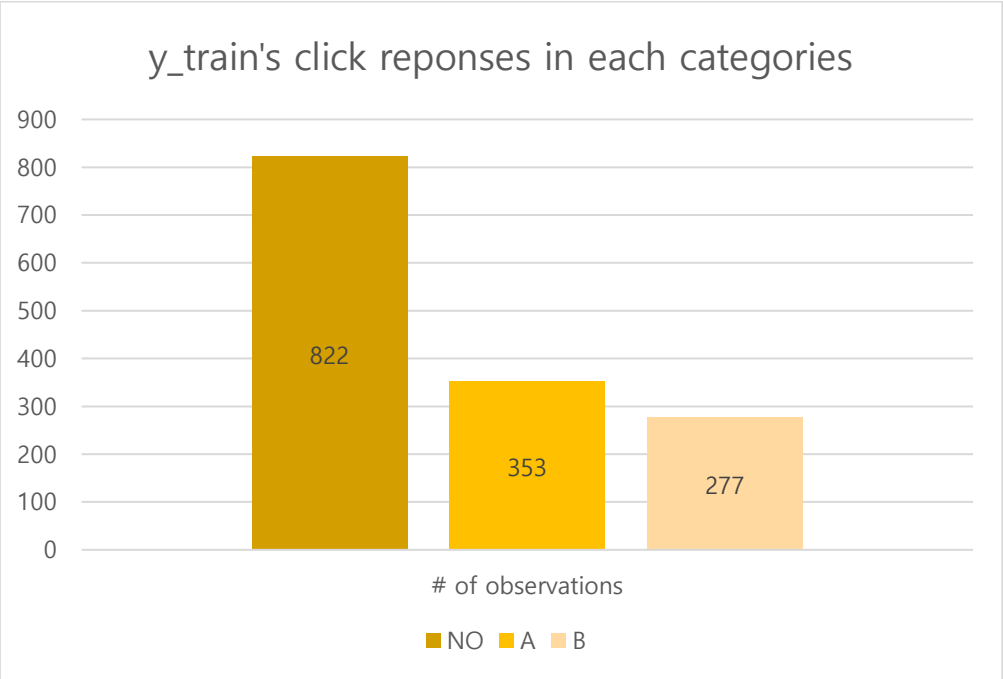
citation: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

Secondly, (if needed) Accuracy Score

Accuracy Score is "the set of labels predicted for a sample must exactly match the corresponding set of labels in y_true".

citation: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html#sklearn.metrics.accuracy_score

- **The distribution of y_train: ({0(no): 822, 2(A): 353, 1(B): 277})**



- Since the number of responses are not equally distributed, only resorting to the accuracy of the model might result in skewed training. That is, the model makes prediction of single value which has highest proportion of the training responses (y_train).
- Additionally, since the assignments' score will be calculated based on the test misclassification, the False Negatives and False Positives are crucial for this classification problem.
- Even though accuracy is widely used, I primally resort to **f1_weighted score** since this assignment is more focused on the False Negatives and False Positives rather than True Positives and True Negatives. Furthermore, in classification problems with imbalanced class distribution, F1_weighted score is a better metric to evaluate our model on.

- Parameter Tuning results:

(1) multinomial logistic regression:
multi_class='multinomial', solver='lbfgs',max_iter=1000

(2) SGDClassifier
max_iter=30000, average=True, alpha=0.01, l1_ratio=0.3, loss='log_loss', penalty='l2'

(3) support vector machines (linear and rbf)
Linear: kernel='linear', *C*=0.01, random_state=2022
Kernel: kernel='rbf', *C*=10,gamma=0.01,random_state=2022

(4) k-nearest neighbors
metric='euclidean', n_neighbors=6, weights='uniform'

(5) random forests
random_state=2020,criterion='entropy', max_features='sqrt', n_estimators= 200, max_depth=10

(6) gradient boosting model
random_state=2022,loss='log_loss', learning_rate=0.1, max_depth=1, max_features=90, n_estimators=75

(7) xgboost
random_state=2022, objective='multi:softmax',num_class=3, booster="gbtree", learning_rate=0.05, alpha=0.95, gamma=0.01, reg_lambda=0.01, n_estimators=125, max_depth=1

(8) linear and quadratic discriminant analysis
linear: shrinkage=0.5,solver='lsqr'
quadratic: reg_param=0.9

- The training result was calculated using same metrics as used for the parameter tuning in advance. (f1_weighted and accuracy score)

	(a) mlogit	(b) Linear SVM	(c) Kernel SVM	(d) SGD Classifier	(e) *k*-nn	(f) Random Forest	(e) GBM (boosting Tree)	(g) XGBoost	(h) LDA	(i) QDA
F1_weighted	0.72	0.57	0.87	0.64	0.69	1.00	0.64	0.62	0.65	0.62
Accuracy	0.73	0.66	0.87	0.69	0.70	1.00	0.68	0.66	0.65	0.62

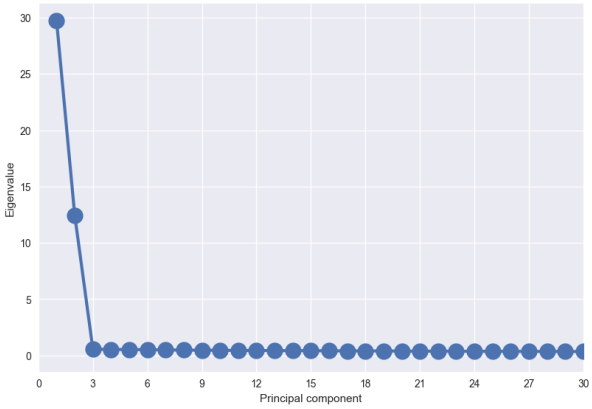
(b)

Dimension reduction on features or feature selection might provide better accuracy score, since models training with a subset of features (such as gradient boosting classifier, random forest etc.) showed better mean CV f1_weighted score (or accuracy score) compared to other models without feature selection. There might be a possibility of model specification issue on including non-informative or redundant predictors (at least 1753 combinations of variables showed linear correlation with higher than 0.5). Therefore, using dimension reduction on features or feature selection to improve prediction might be desirable.

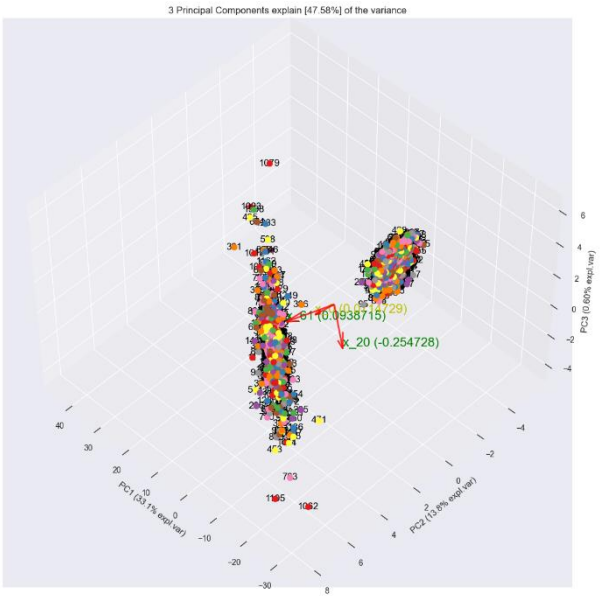
1) PCA

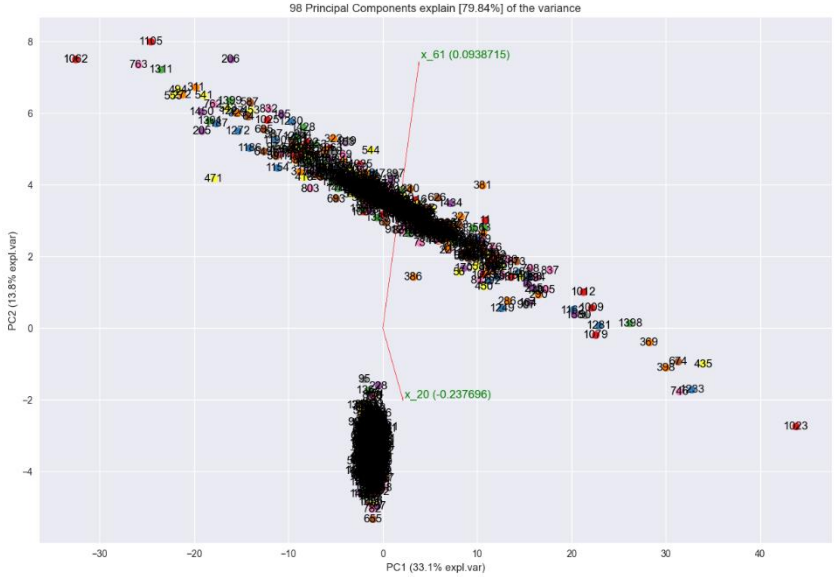
$$proportion = specific\ \ component's\ \ proportion$$
$$= \frac{specific\ component's\ eigenvalue\ (주성분\ 분산)}{sum\ of\ all\ eigenvalues\ (모든\ 주성분\ 분산의\ 합)}$$

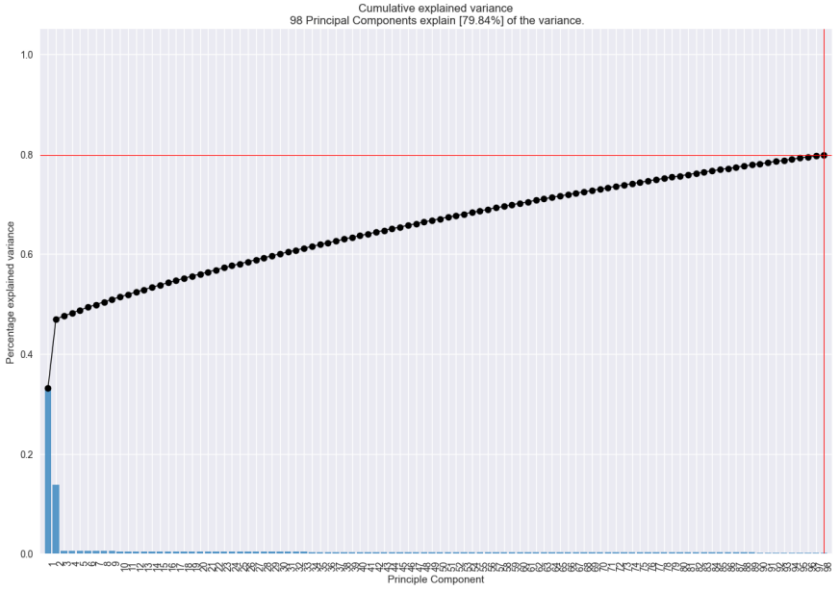
	Eigenvalue	Proportion	Cumulative
pca1	29.741754	0.331111	0.331111
pca2	12.455516	0.138666	0.469776
pca3	0.546720	0.006087	0.475863
pca4	0.535468	0.005961	0.481824
pca5	0.522446	0.005816	0.487640
pca6	0.507087	0.005645	0.493286
pca7	0.484598	0.005395	0.498681
pca8	0.473509	0.005272	0.503952
pca9	0.466847	0.005197	0.509150
pca10	0.446134	0.004967	0.514116



According to screeplot there is not much difference on eigenvalue after 3, therefore the number of components were set to three.







However, 3 components cannot be used since the variance explained by 3 components is only 48% of the total variance. If the variance explained by the components needs to be over 80%, then at least 98 components need to be set. This discrepancy between significant eigenvalues and very small variance explained by marginal components makes it difficult to use PCA in the analysis. Therefore, the PCA was not used in this analysis as a dimension reduction technique.

2) L1-based feature selection / Tree-based feature selection

(source: https://scikit-learn.org/stable/modules/feature_selection.html)

First, l1-regularized linear SVM and logistic regression was used to select features. In order to minimize the possibility of model overfit, cross-validation was adopted to choose best lambdas for l1-regularization.

- Linear SVM - original dataset: (1452, 251) → feature selected: (1452, 60)
- Multinomial Logistic Regression - original dataset: (1452, 251) → feature selected: (1452, 45)

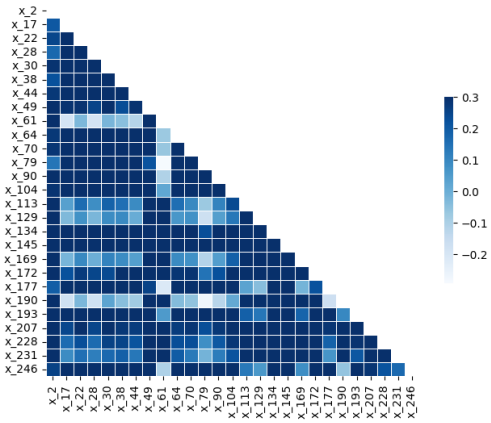
Second, tree- based feature selection was used to select features. Tree-based estimators can be used to compute impurity-based feature importances, which in turn can be used to discard irrelevant features

- Extra Tree - original dataset: (1452, 251) → feature selected: (1452, 45)

The features that are selected in at least two of the each selection methods were included in the final feature selection.

original dataset: (1452, 251) → **features chosen: (27, 1)**

selected features: *[2 17 22 28 30 38 44 49 61 64 70 79 90 104 113 129 134 145 169 172 177 190 193 207 228 231 246]*



(there aren't sets of variables that shows correlation higher than 0.3)

- Using selected features, the hyperparameter tuning (using 5-fold cross validation with f1_weighted) were conducted and 5-fold mean cross-validation scores of two metrics calculated again (f1_weighted and accuracy score)

- Parameter Tuning results:

(1) multinomial logistic regression:
C=0.01,multi_class='multinomial', max_iter=10000, dual=False, random_state=2020

(2) SGDClassifier
max_iter=30000, average=True, alpha=0.1, l1_ratio=0.5, loss='perceptron', penalty='l2'

(3) support vector machines (linear and rbf)
Linear: C= 0.01 , penalty="l1", dual=False, random_state=2020, max_iter=20000
Kernel: kernel='rbf',C=600,gamma=0.0004, random_state=2022

(4) k-nearest neighbors
metric='manhattan', n_neighbors=20, weights='distance'

(5) random forests
random_state=2020,criterion='entropy', max_features='sqrt', n_estimators= 200, max_depth=10

(6) gradient boosting model
random_state=2022,loss='log_loss', learning_rate=0.05, max_depth=2, max_features=15, n_estimators=175

(7) xgboost
random_state=2022, objective='multi:softmax',num_class=3, n_estimators=25, learning_rate=0.1, max_depth=3

(8) linear and quadratic discriminant analysis
linear: shrinkage=0.5,solver='lsqr'
quadratic: reg_param=0.8

문제 2.

(a) Estimated test performance for the models

- **stratified 5-fold mean cross-validation scores (with 5 repetition) of two metrics** is used as an estimated test performance of the model. Compared to validation set approach, cross-validation can use more data to train (which helps reduce overfitting) and can use all data to test the model even though cross-validation takes more time.
  (The last column is the weighted average of two mean metrics (by 6:4) to make model selection decision easy)
- Even though the highest mean f1_weigted score from cross-validation is quadratic discriminant analysis, I selected linear discriminant analysis as a model. Even though the mean score discrepancy is only 0.01, there is much higher discrepancies in mean accuracy score (and LDA showed better accuracy score). Therefore, linear discriminant analysis was used a final model of the classification.

5-fold CV / feature selected (n=27)	Mean F1 weighted	Std F1 weighted	Mean Accuracy Score	Std Accuracy Score	weighted mean (6:4)
linear discriminant analysis	0.625	0.022	0.633	0.023	0.628
quadratic discriminant analysis	0.626	0.022	0.627	0.023	0.626
Gradient boosted tree	0.602	0.022	0.642	0.021	0.618
xgboost	0.597	0.024	0.645	0.018	0.616
knn	0.605	0.024	0.635	0.02	0.617
ramdom forest	0.594	0.023	0.64	0.017	0.612
rbf svm	0.569	0.019	0.651	0.018	0.602
sgd classifier	0.572	0.019	0.644	0.017	0.601
multinomial logistic	0.568	0.018	0.647	0.016	0.6
linear svm	0.563	0.018	0.648	0.015	0.597

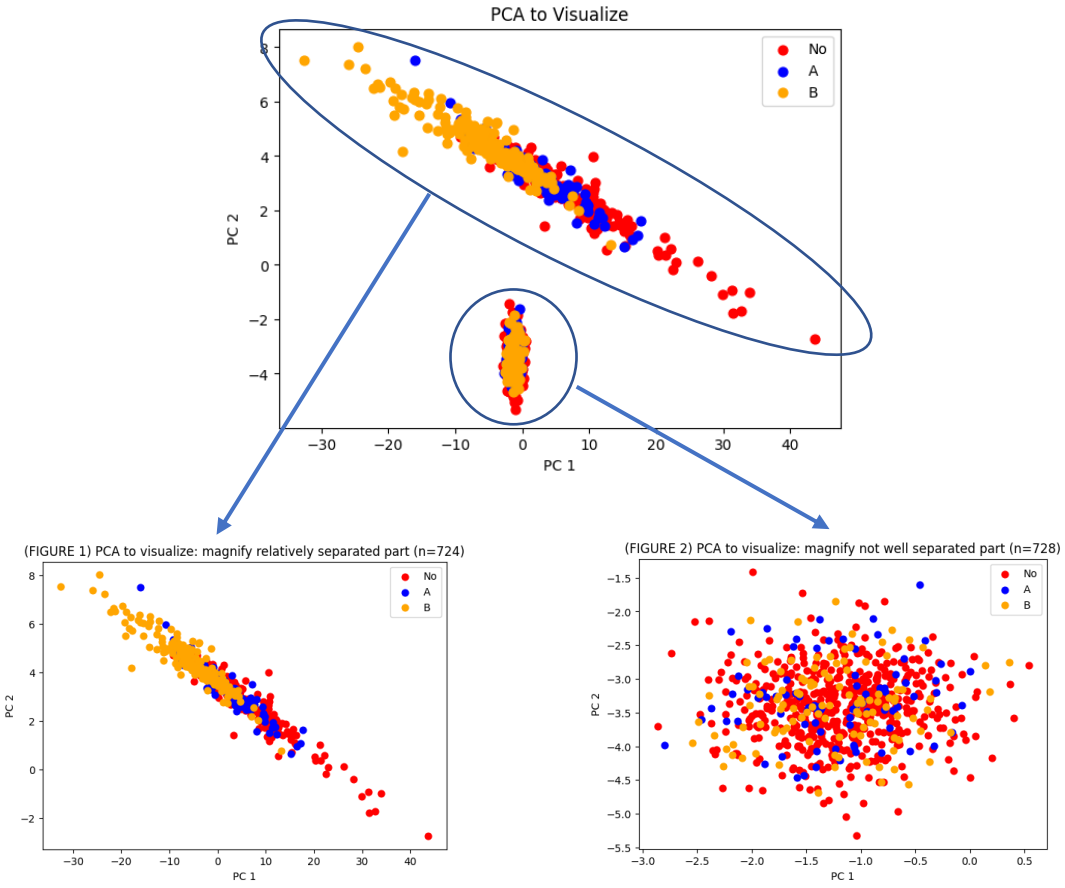
- **y_train vs. y_pred (by LDA)**

Classification	No	A	B
y_train	822	277	353
y_pred (by LDA)	188	47	65

Test of homogeneity in distribution: (Chi-squared)

$$\chi^2(df = 2) = 3.873, \quad p - value = 0.144$$

- It cannot be rejected that real clicks and the classifications are from different distributions (only if the tested data is randomly selected from the same distribution of training data)



- The reason why cross-validation score is comparatively low is since the data is not easily separable. Using dimension reduction to plane, I visualized the click responses with two principal components. (46% of variances are explained by two components)
- The dots are largely separated into two parts. One is like narrow oval-shaped, and the other one is like sphere shaped.
- If we apply distinct prediction models to two separate parts, the prediction score might be improved.

(b) Submitted.