Homework 2

2022-24790 Sungwoo PARK

For this homework I discussed with several classmates from GSDS: 곽남주, 연소정, 유지상, 윤용상, 이다예, 이동영, 이세라, 임상수, 최광호, though I swear that I never copied any code lines.

Problem 1.

(a)

(a)-1. Baseline

- For this project, I used the models listed below for baseline.

Logistic regression	RBF kernel SVM
Nearest neighbors	Decision tree classifier
Linear discriminant analysis	Bagging classifier
Quadratic discriminant analysis	Random forest classifier
Linear Support Vector Classifier	Gradient boosting classifier

- Also, I used two metrics: accuracy and ROC-AUC score for multiclass. The `scikit-learn` supports the ROC-AUC score for multiclass, by passing an argument `multi_class`.
- Furthermore, I tested the performance of each baseline model with two approaches: One-versus-one and One-versus-rest. The result is shown in the table below.

model	OVO_h	OVO_c	OVR_h	OVR_c
LogisticRegression	0.532646	0.519251	0.546392	0.530942
KNeighborsClassifier	0.577320	0.517182	0.484536	0.469665
LinearDiscriminantAnalysis	0.591065	0.543372	0.587629	0.556457
LinearSVC	0.532646	0.520621	0.542955	0.517168
QuadraticDiscriminantAnalysis	0.553265	0.555085	0.556701	0.547510
SVC	0.635739	0.643237	0.649485	0.641168
BaggingClassifier	0.646048	0.615691	0.615120	0.616381
DecisionTreeClassifier	0.536082	0.531677	0.453608	0.453824
RandomForestClassifier	0.639175	0.642541	0.639175	0.634262
GradientBoostingClassifier	0.628866	0.628755	0.656357	0.631516

    - The small letter `h` and `c` denotes hold-out validation and cross-validation, respectively.
    - Although I tried to use accuracy and ROC-AUC both, since the Linear SVC and SVM does not support the probability prediction, I only used accuracy for model comparison. While I used ROC-AUC score for model optimization. I'll explain it later.
    - We can see that for overall models, CV score was lower than the corresponding holdout validation score.

(a)-2. Model optimization

- I chose my best-model candiates, which are:
    - Logistic regression
    - Linear discriminant analysis
    - Bagging classifier
    - Random forest classifier
    - Gradient boosting classifier
- I'll explain how I chose these models in (b).
- With these models, I implemented the hyper-parameter tuning using the GridSearchCV provided by the `scikit-learn`. I chose two final candidates after

grid search: one is the Gradient boosting classifier, which achieved the highest score in terms of accuracy of 65.4%, and the other is the Logistic regression, with the highest ROC-AUC score of 0.782.

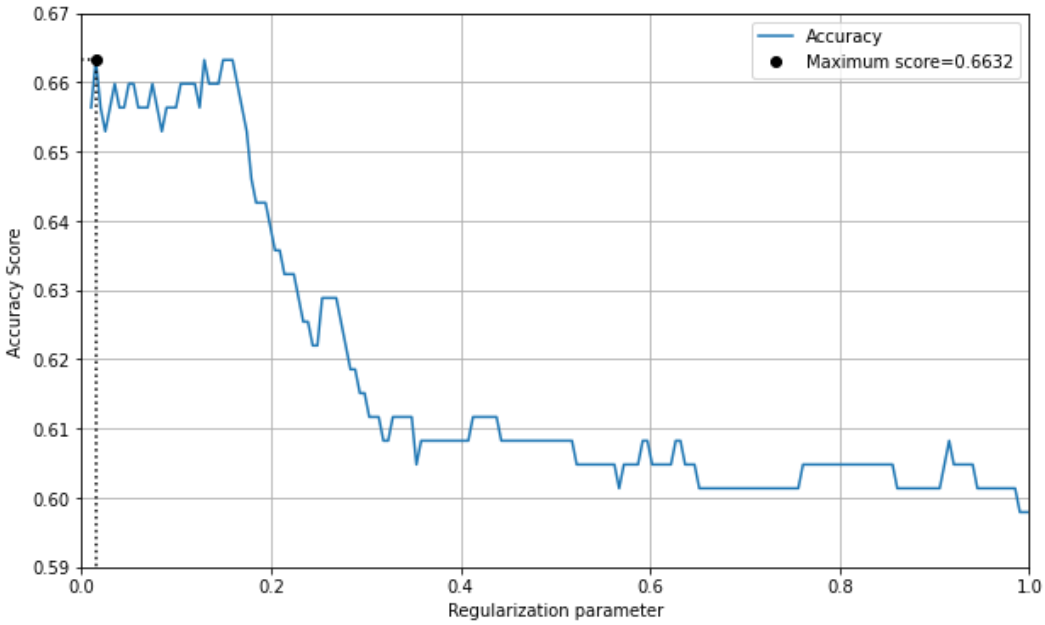
(b)

- I implemented three feature selection methods: PCA, shrinkage(regularization), and stepwise selection(especially the forward selection).
- Dimension reduction using PCA

	1	2	3	4	5	6	7	8	9	10	11	12
0	0.334324	0.334324	0.334324	0.334324	0.334324	0.334324	0.334324	0.334324	0.334324	0.334324	0.334324	0.334324
1	0.000000	0.118988	0.118988	0.118988	0.118988	0.118988	0.118988	0.118988	0.118988	0.118988	0.118988	0.118988
2	0.000000	0.000000	0.006341	0.006250	0.006365	0.006351	0.006358	0.006358	0.006364	0.006359	0.006359	0.006365
3	0.000000	0.000000	0.000000	0.006202	0.006184	0.006209	0.006218	0.006215	0.006226	0.006217	0.006224	0.006226
4	0.000000	0.000000	0.000000	0.000000	0.006064	0.006079	0.006001	0.006073	0.006077	0.006055	0.006061	0.006074

Explained variance dataframe of PCA

  - From the result, it is seen that the explained variance ratio drops dramatically from the third PC, implying that the first two PCs are significant.
  - However, the first two PCs only explains less than 50% of the total variance. Therefore, it seems that the dimension reduction is not useful here.
- Shrinkage



Accuracy plot for various regularization parameter

- I implemented the shrinkage by assigning the parameter of the logistic regression `penalty='l1'`, and checked the performance of each model under various regularization parameter, λ.
- The maximum accuracy achieved from shrinkage is 66.3%, and the λ at the maximum accuracy is 0.015.
- With this λ, 16 features are selected, since their coefficients are nonzero.

- Forward selection
  - I also implemented the forward selection using the package `SequentialFeatureSelection`. Here, I made my own early-stopping rule, which is that the selection is stopped when the maximum performance score - measured by accuracy - does not increase at least 3 times.

- After the selection for each model, I compared the performance of the shrinkage method and the selection method.

model	OVO_hL	OVO_cL	OVR_hL	OVR_cL	OVO_hS	OVR_hS
LogisticRegression	0.670103	0.649437	0.673540	0.648743	0.652921	0.652921
KNeighborsClassifier	0.615120	0.614990	0.580756	0.600531	0.639175	0.618557
LinearDiscriminantAnalysis	0.663230	0.642543	0.666667	0.645991	0.642612	0.639175
LinearSVC	0.659794	0.649435	0.666667	0.649437	0.649485	0.656357
QuadraticDiscriminantAnalysis	0.587629	0.581910	0.601375	0.583979	0.615120	0.618557
SVC	0.639175	0.644621	0.649485	0.645302	0.632302	0.649485
BaggingClassifier	0.621993	0.633599	0.591065	0.630812	0.628866	0.611684
DecisionTreeClassifier	0.498282	0.517192	0.487973	0.487567	0.532646	0.463918
RandomForestClassifier	0.642612	0.643910	0.625430	0.645278	0.632302	0.639175
GradientBoostingClassifier	0.670103	0.632897	0.652921	0.633587	0.656357	0.652921

  - The figure above is a table I used to compare models, and each score is measured by accuracy.
  - The capital letter `L` and `S` denotes regularization(LASSO) and forward selection, respectively.
  - The performance with the features selected from regularization scored higher than with those selected from forward selection for overall models.

    - After comparison, I picked the best-model candidates and implemented hyper-parameter tuning for these candidates, where the results are shown in (a):
        - Logistic regression
        - Linear discriminant analysis
        - Bagging classifier
        - Random forest classifier
        - Gradient boosting classifier
- Moreover, using the fact that the tree-based model computes the feature importance, the metric showing how much each element decreases the loss function, I also tried the feature selection based on the feature importance.
    - Specifically, I selected the top twenty features whose feature importance were higher than any others, then fit the model again with those selected features.
    - However, it turned out that validation score from the variable selection based on the feature importance was lower that that from the selection based on shrinkage.

Problem 2.

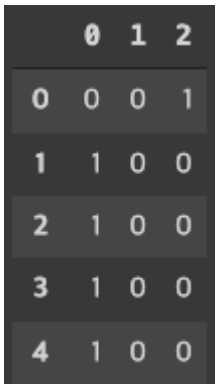
(a)

- As I mentioned above, I chose two final candidates, and I compared them by estimating the test performance using hold-out validation.
    - Gradient boosting classifier scored 64.6% accuracy and 0.756 ROC-AUC.
    - Logistic regression scored 67.7% accuracy and 0.787 ROC-AUC.
- I chose the logistic regression as my best model, and the reasons are:
    - It scored better performance than the boosting model in terms of both metrics, of course.
    - Also, the target values are imbalanced: almost 56.6% of the targets are no-clicks; 19.1% clicked the ad `A` and 24.3% clicked the ad `B`. Since the higher ROC-AUC means that the model classifies much more clearly, I thought that

the ROC-AUC would be more critical for this problem, than the accuracy itself.

(b)

- After test prediction, I restructured the target prediction, which was consolidated into one column at the preprocessing step, into the 300×3 matrix. The part of the test result is shown in the figure below.



	0	1	2
0	0	0	1
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0