MACHINE LEARNING

CSI 5155

Supervised Learning and Evaluation of Learning

Assignment 2

Authors
Alexandra SKLOKIN (300010511)

Professor
Dr. Herna VIKTOR

November 2, 2021

Contents

1 Methodology (1,2,4)

For this assignment I used *jupyter notebooks*, Excel spreadsheets, and the *sklearn Python* library. Please find all of the code for this assignment in my git repository[1].

1.1 Preprocessing

Preprocessing led to better accuracies based on some preliminary testing. The following pipeline was applied to each dataset for preprocessing:

Algorithm 1 Preprocessing Pipeline ($dataframe$)

```
# Cleaning the Data
Fill_Empty_Entries(dataframe)
Transform_To_Count(dataframe[dates])
# Feature Selection
sets ← []
accuracies ← []
for k = 1 to length(dataframe.features) do
    set ← SelectKBest(dataframe, k)
    sets.append(set)
    model ← GradientBoostingEnsemble(set.X_train, set.y_train)
    accuracies.append(accuracy(set.y_test, model.predict(set.X_test)))
end for
best_features = sets[index_of_max(accuracies)]
dataframe ← dataframe[best_features]
# Feature Transformation
1_of_C_Encoding(dataframe[categorical_features])
Standardization(dataframe[numerical_features])
# Resampling
Oversample(dataframe)
return dataframe
```

This is the final results, after preprocessing, of the datasets:

	Online Shopping	Marketing Campaign	Heart
Entries	20,844	2,316	330
Features	8	16	11

Table 1: Sizes of Datasets after Preprocessing

[1] https://github.com/alexandrasklokin/CSI5155/tree/main/Assignment2

1.2 Training

Hyperparameter tuning was performed in an attempt to improve results. Then, I used Stratified 10-Fold Cross Validation in an attempt to reduce variance among folds, since the distribution of train/test set will always attempt to reflect that of the full set. The following describes the algorithm used for training:

Algorithm 2 Training with 10-CV ($dataframe$)

```
accuracies ← []
params ← []
models ← [knn, svm, dt, rf, mlp, gbe]
# Hyperparameter Tuning
for model in models do
    params[model] = GridSearch(dataframe, model)
end for
# Cross Validation Training
cv_10 ← StratifiedKFold(n_splits = 10, shuffle = True)
for training, testing in cv_10 do
    for model in models do
        trained ← model.fit(training.X, training.y, params[model])
        accuracy ← accuracy(test.y, trained.predict(test.X))
        accuracies.append(accuracy)
    end for
end for
return accuracies
```

2 Experimental Results

2.1 Accuracy on Shopping Intention Data (3)

Fold	Accuracy					
	KNN	SVM	DT	RF	MLP	GBE
1	0.88564	0.88159	0.89457	0.90754	0.90349	0.87835
2	0.88808	0.89051	0.90024	0.90754	0.90673	0.89457
3	0.88564	0.88402	0.89457	0.89538	0.89781	0.66910
4	0.88808	0.87835	0.89294	0.89781	0.89132	0.87186
5	0.90105	0.89213	0.91565	0.91646	0.92214	0.89700
6	0.88240	0.88078	0.89376	0.89943	0.89619	0.85726
7	0.88808	0.87916	0.89294	0.89862	0.88727	0.87997
8	0.88564	0.88078	0.89943	0.90511	0.90187	0.88159
9	0.88402	0.88808	0.89132	0.89700	0.89457	0.87997
10	0.89051	0.89538	0.89781	0.90998	0.90430	0.84347
avg	0.88792	0.88508	0.89732	0.90349	0.90057	0.85531
std	0.00490	0.00570	0.00672	0.00653	0.00923	0.06388

Table 2: Accuracy on Shopping Intention Dataset

2.2 Paired T-Test (3)

Model 1	Model 2	Avg Diff.	t-value	p-value	reject H_0?
KNN	SVM	0.00284	1.65503	0.13230	False
KNN	DT	-0.00941	-8.58279	0.00001	True
KNN	RF	-0.01557	-10.77813	0.00000	True
KNN	MLP	-0.01265	-5.82999	0.00025	True
KNN	GBE	0.03260	1.55469	0.15444	False
SVM	DT	-0.01225	-6.09221	0.00018	True
SVM	RF	-0.01841	-10.33599	0.00000	True
SVM	MLP	-0.01549	-6.76745	0.00008	True
SVM	GBE	0.02976	1.39899	0.19532	False
DT	RF	-0.00616	-4.87875	0.00087	True
DT	MLP	-0.00324	-2.37171	0.04179	True
DT	GBE	0.04201	2.01368	0.07488	False
RF	MLP	0.00292	1.95750	0.08197	False
RF	GBE	0.04818	2.36139	0.04250	True
MLP	GBE	0.04526	2.15987	0.05908	False

Table 3: T-Test for Significant Difference of Algorithms

2.3 Accuracy on Three Datasets (5)

Refer to Tables 2, 7, and 8 for 10-fold cross validation accuracies. Average accuracies are shown in Table 4.

Dataset	Avg. Accuracy over 10-Fold CV					
	KNN	SVM	DT	RF	MLP	GBE
Online Shopping	0.88792	0.88508	0.89732	0.90349	0.90057	0.85531
Marketing Campaign	0.87563	0.89247	0.87951	0.91320	0.88989	0.93133
Heart	0.86970	0.56364	0.82424	0.85455	0.76364	0.86061
avg	0.87775	0.78040	0.86702	0.89041	0.85137	0.88242
std	0.00759	0.15330	0.03111	0.02567	0.06219	0.03466

Table 4: Average Accuracy on Three Datasets

2.4 Friedman's Test (5)

Dataset	Ranking					
	KNN	SVM	DT	RF	MLP	GBE
Online Shopping	4.0	5.0	3.0	1.0	2.0	6.0
Marketing Campaign	6.0	3.0	5.0	2.0	4.0	1.0
Heart	1.0	6.0	4.0	3.0	5.0	2.0
avg	3.7	4.7	4.0	2.0	3.7	3.0
std	2.055	1.247	0.816	0.026	1.247	2.160

Table 5: Ranking of Algorithm Accuracy on Three Datasets

Equation 1	3.5
Equation 2	12.5
Equation 3	3.5
Friedman Stat	3.57143
P-Value	0.61261
Reject H_0?	False

Table 6: Friedman Test for Significant Difference of Algorithms [1]

3 Discussion (6)

3.1 Accuracies

In Section 2.1 we obtain the accuracies of six models over ten folds of the *Online_Shopping* dataset. We note that for each model the accuracies are very similar among folds, in comparison to accuracies for *Marketing_Campaign* and *Heart* datasets where there was significantly higher variance of accuracies across folds (refer to Tables 7 and 8).

In Section 2.3 we can also note that the average accuracies for the *Online_Shopping* and *Marketing_Campaign* datasets are usually higher than for the *Heart* dataset.

These are likely due to the sizes of the datasets, 20844, 2316, and 330, respectively. It is possible that the *Marketing_Campaign* and *Heart* datasets do not have enough entries to make as good predictions as on *Online_Shopping*. In fact, since we are using K-Fold Cross Validation, the folds will contain even less data diversity. This can lead to the higher variance of accuracies among folds for those two datasets.

In addition, during preprocessing, it was noted that the correlation between features and the target were lower in the *Marketing_Campaign* and *Heart* datasets. This might also contribute to the lower accuracies, since the features may not be sufficient to make accurate predictions.

3.2 T-Test

In Section 2.2 there was significant evidence of model difference for:

- KNN $\longleftrightarrow$ DT, RF, MLP
- SVM $\longleftrightarrow$ DT, RF, MLP
- DT $\longleftrightarrow$ RF, MLP

and lack of significant evidence of model difference for:

- KNN $\longleftrightarrow$ SVM
- RF $\longleftrightarrow$ MLP
- GBE $\longleftrightarrow$ KNN, SVM, DT, RF, MLP [2]

From this experiment, we can tell that most of the models are significantly different on the *Online_Shopping* dataset. Each model architecture is

[2] KNN= K-Nearest Neighbours, SVM= Suport Vector Machine, DT= Decision Tree, RF= Random Forest, MLP= Multilayer Perceptron, GBE= Gradient Boosting Emsemble

theoretically different- implying different classification decisions and results. There are two pairs of algorithms which are not significantly different which are KNN/Linear-SVM, and RF/MLP. It does seem that simpler geometric models (KNN, SVM) did not differ. It is also evident that the GBE model is not significantly different from any other model.

However, similar to the 'No Free Lunch' Theorem, it cannot be said that the above mentioned model similarities/dissimilarities will hold on other datasets.

3.3 Friedman-Test

Overall, different models performed better over different datasets, as expected by the 'No Free Lunch' theorem. There did not seem to be one model which performed better or worse than others over all datasets (refer to Table 5). For example, both the *KNN* and *GBE* models were the best over one dataset, and then worst over another. Although it can not be explicitly explained why some models will perform better over certain datasets than others, it may have to do with size of the datasets. Again, using the same models, GBE is prone to overfitting, and KNN is prone to underfitting.

Using the Friedman test, in section 2.4, there is not significant evidence of model difference. Since there was no significant difference, we did not need to perform the Nemenyi Post-Hoc Test to determine the critical difference.

4 References

[1] Herna Viktor. *Topic5 Experiments (CSI5155 Lecture Notes)*. Oct. 2021.

A Accuracy on Marketing Campaign Data (5)

Fold	Accuracy					
	KNN	SVM	DT	RF	MLP	GBE
1	0.87069	0.84052	0.84483	0.87500	0.86638	0.91379
2	0.87931	0.90948	0.90517	0.93966	0.91810	0.96983
3	0.90086	0.88793	0.87069	0.91810	0.89655	0.93966
4	0.90948	0.92672	0.93534	0.95259	0.90517	0.96983
5	0.87931	0.90086	0.85345	0.88793	0.88793	0.90517
6	0.86638	0.91810	0.92672	0.92672	0.87500	0.92672
7	0.85281	0.88312	0.86147	0.88745	0.87879	0.90043
8	0.84848	0.86147	0.87446	0.88312	0.87013	0.91775
9	0.88312	0.88312	0.84848	0.92641	0.88312	0.93939
10	0.86580	0.91342	0.87446	0.93506	0.91775	0.93074
avg	0.87563	0.89247	0.87951	0.91320	0.88989	0.93133
std	0.01825	0.02547	0.03050	0.02603	0.01785	0.02291

Table 7: Accuracy on Marketing Campaign Dataset

B Accuracy on Heart Data (5)

Fold	Accuracy					
	KNN	SVM	DT	RF	MLP	GBE
1	0.84848	0.54545	0.84848	0.84848	0.84848	0.84848
2	0.84848	0.42424	0.81818	0.78788	0.75758	0.84848
3	0.93939	0.45455	0.78788	0.90909	0.51515	0.87879
4	0.78788	0.54545	0.84848	0.84848	0.81818	0.84848
5	0.84848	0.66667	0.84848	0.84848	0.84848	0.90909
6	0.87879	0.60606	0.81818	0.81818	0.51515	0.84848
7	0.90909	0.48485	0.81818	0.87879	0.81818	0.84848
8	0.81818	0.60606	0.75758	0.78788	0.75758	0.84848
9	0.90909	0.54545	0.81818	0.93939	0.87879	0.87879
10	0.90909	0.75758	0.87879	0.87879	0.87879	0.84848
avg	0.86970	0.56364	0.82424	0.85455	0.76364	0.86061
std	0.04505	0.09506	0.03264	0.04655	0.13055	0.02010

Table 8: Accuracy on Heart Dataset