3/9/2020

Applied Machine Learning

Assignment 2

Rupal Shrivastava
RXS190017

Table of Contents

Introduction 2

Dataset Description 2

Data Preprocessing 2

Algorithms 3

Support Vector Machine (SVM) 3

Linear Kernel 3

Polynomial Kernel 3

Radial Basis Function Kernel 3

Learning Curve 3

Decision Trees 4

Entropy 4

Gini 4

Learning Curve 4

Pruning 4

Boosting 5

Gradient Boosting 5

Adaboost 5

Learning Curve 5

Pruning 5

Conclusion 6

Citation 6

Variable Description 6

Introduction

In this assignment, we will implement Support Vector Machine, Decision Tree and Boosting Algorithm on the GPU runtime and Bank Marketing dataset. We are performing Binary Classification on the target variable by implementing k-fold cross validation on the training dataset for all the three algorithm and identifying the best way to classify each dataset.

Dataset Description

The GPU dataset used can be found at: https://archive.ics.uci.edu/ml/datasets/SGEMM+GPU+kernel+performance
The dataset consists of 14 features (independent variables) and 241,600 rows. First 10 features are ordinal and last 4 as binary variables. There were 4 runs performed on this dataset, which corresponds to 4 runtime variables in the dataset. More details on the dataset can be found in the link mentioned above.

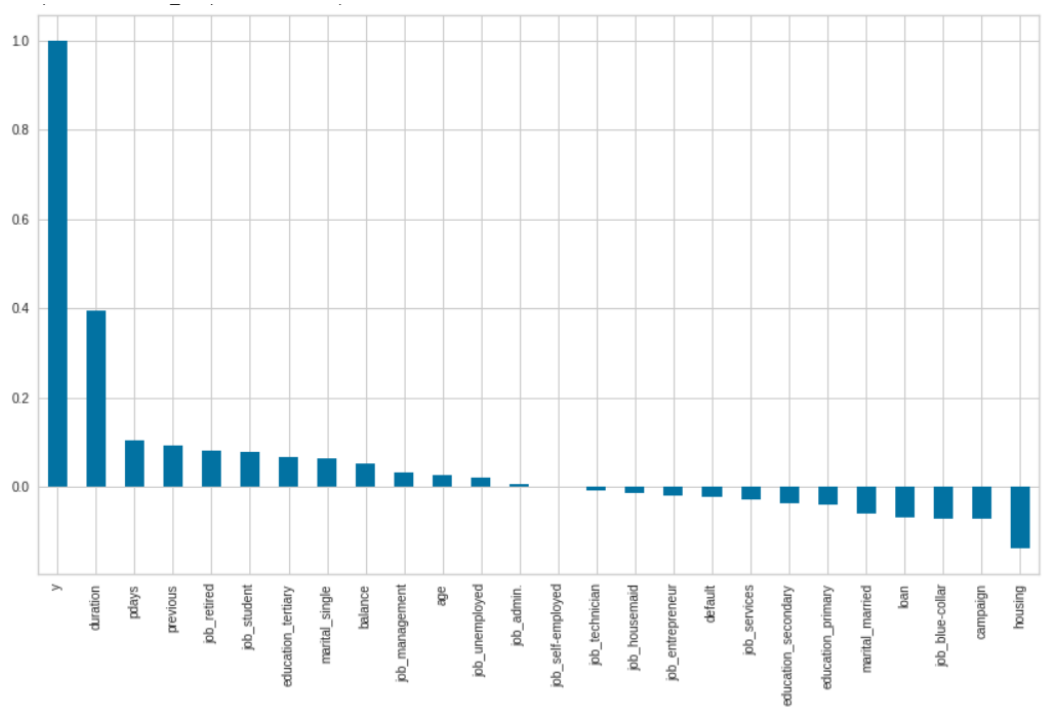
The Bank Marketing dataset can be found at: http://archive.ics.uci.edu/ml/datasets/Bank+Marketing#
This dataset contains Direct Marketing (cold calling) campaign information of a Portuguese Bank. The goal is to predict if the client will subscribe to a term deposit. I chose this dataset because it is interesting to see how many conversations does cold calling bring to the business.
This dataset contains 15 features, and 45211 rows. It has a combination ordinal, nominal and interval variables.

The assignment is divided in three parts: Data Preprocessing, Algorithms and Experiments.

Data Preprocessing

Preprocessing for the GPU runtime remains the same as performed for the first project.
For Bank Marketing dataset, we check for the following:

1. *Null Values*: The data has no null datapoints
2. *Feature selection:* We drop the features which do not impact the classification like: day, month, contact and poutcome. (Variable description is at the end of the Report).
3. *Outliers*: There are no outliers in the dataset.
4. *Variable treatment*: The dataset has 4 binary variables with yes and no as the values, which were converted to zeros and ones.
   It also has 3 nominal variables for which dummies were created.
5. *Correlation Matrix*: Checked the correlation among features. None of the features are highly correlated with each other, as can be seen by the graph below:



6. *Split the dataset:* The dataset is split into test and train, with test size as 30% of the dataset.
7. *Feature Scaling*: For the purpose of making bringing all the feature on the same scale, we perform feature scaling on all the features. Features are scaled using: $$z = \frac{x - \mu}{\sigma}$$

Algorithms

I have applied the following 3 classification algorithms on the 2 datasets. In each algorithm, I have performed 5-fold cross-validation for each experiment to create a generalized model for the test dataset, increase efficiency and reduce overfitting. For the best algorithm, I have also created learning curves – accuracy vs train size for test and validation dataset.

Support Vector Machine (SVM)

We are applying SVM on both the dataset to classify the target variable. Since we cannot judge which functions provide a better fit for each dataset, we use different kernels functions in SVM to find the best way to classify the data. I have experimented with 3 kernels for both the datasets and the results are as follows:

Linear Kernel

This kernel tries to classify the target into 2 part with a linear decision boundary. The accuracy of this kernel for GPU dataset is 84.89% and that for Bank Marketing dataset is 88.24%

Polynomial Kernel

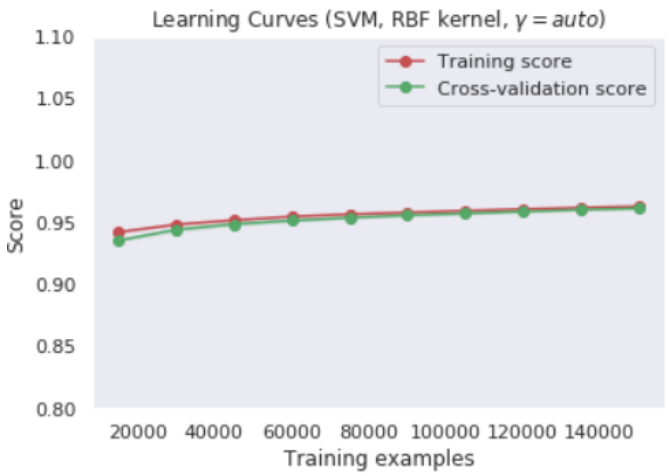
This kernel tries to classify the target into 2 part with a 3rd degree polynomial decision boundary. The accuracy of this kernel for GPU dataset is 90.95% and that for Bank Marketing dataset is 88.75%

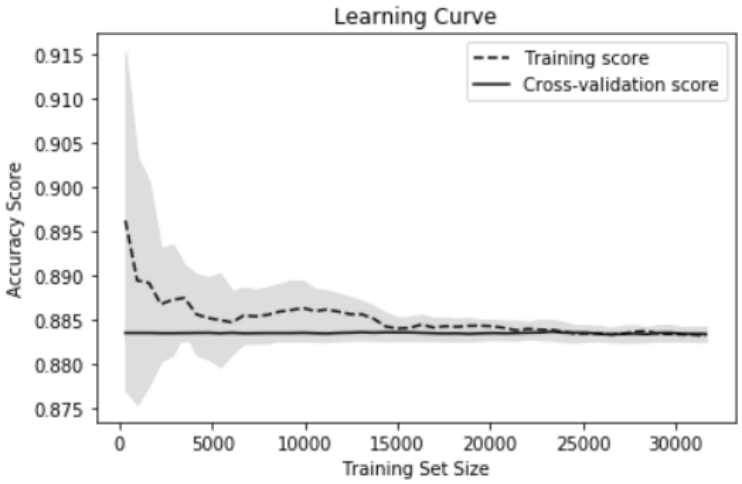
Radial Basis Function Kernel

This kernel tries to classify the target into 2 part with a radial decision boundary. The accuracy of this kernel for GPU dataset is 95.94% and that for Bank Marketing dataset is 88.78%. This kernel provides the best accuracy of the three, and better so for the GPU runtime dataset.

Learning Curve

Left learning curve is for GPU Runtime Dataset and right one is for Bank Marketing Dataset.





Decision Trees

Decision tree is another classification algorithm, where the dataset features is split into groups and a hierarchical tree is generated with nodes and leaves, which predicts the target variable. I have experimented with 2 criteria for both the datasets and the results are as follows:

Entropy

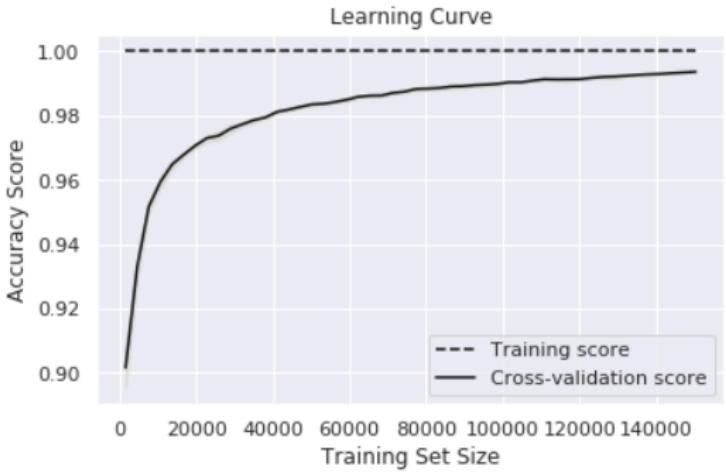
In this criterion we try to identify the node to maximize the information gain. The accuracy of this criteria for GPU dataset is 99.32% and that for Bank Marketing dataset is 85.29%

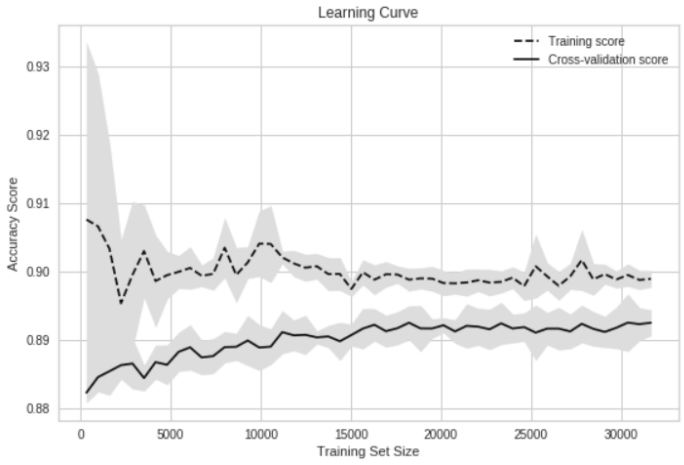
Gini

This is very similar to entropy but is faster to compute. The accuracy of this criteria for GPU dataset is 99.32% and that for Bank Marketing dataset is 85.14%. Both the criteria performance is almost similar to one another. Decision tree seem to be overfitting in case of GPU dataset, but works well for the Bank Marketing dataset.

Learning Curve

Left learning curve is for GPU Runtime Dataset and right one is for Bank Marketing Dataset.





Pruning

Pruning is the technique of reducing the tree complexity by removing the features which do not contribute much to the classification. I applied grid search method to identify when to stop. I have pruned the best criteria model for each dataset.

The accuracy of Gini criteria for GPU dataset is:

```
Accuracy:  0.9877246684443914
Best Parameters:  {'criterion': 'gini', 'max_depth': 20, 'min_samples_leaf': 5}
```

Since the tree was overfitting, the accuracy reduces after pruning, in case of GPU runtime dataset.

And for entropy criteria of Bank Marketing dataset is:

```
Accuracy:  0.8959776853743637
Best Parameters:  {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 100}
```

The accuracy increases in case of Bank Marketing dataset, as pruning provides better fit of the model.

Boosting

Boosting is the algorithm which is an iterative process to train weak learners to generate the prediction rule. After many iterations, the boosting algorithm combines these weak learners to produce a strong learner for classifying data. I have performed two types of boosting as follows:

Gradient Boosting

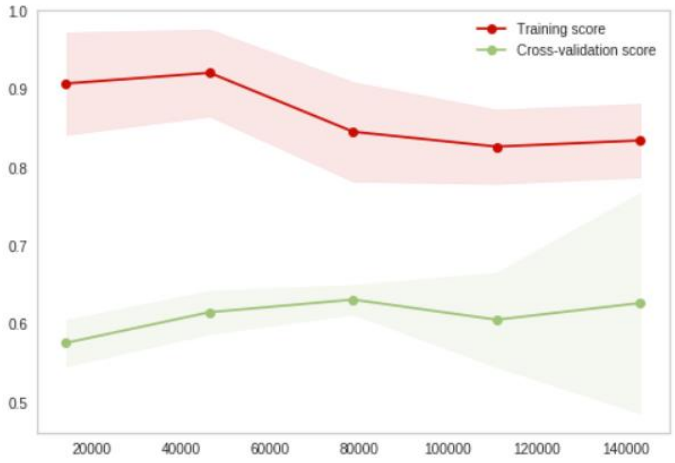
This is the generic algorithm to provide approximate solutions to the weak learners' addition. The accuracy of this criteria for GPU dataset is 77.71% and that for Bank Marketing dataset is 88.24%

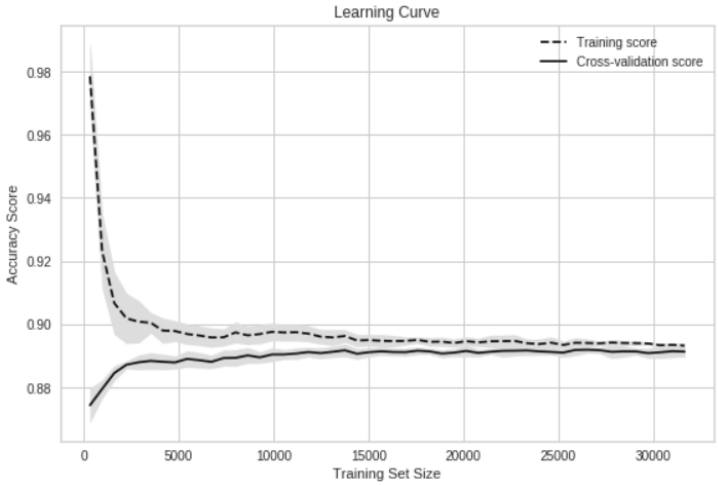
Adaboost

This algorithm allows the users to assign weights to the weak learners while training the model. The accuracy of this criteria for GPU dataset is 80.03% and that for Bank Marketing dataset is 89.52%

Learning Curve

Left learning curve is for GPU Runtime Dataset and right one is for Bank Marketing Dataset.





Pruning

I again applied grid search method to identify when to stop. I have performed pruning on Adaboost algorithm, as it is the better of the two.

The accuracy for GPU dataset is:

```
Accuracy:  0.8033488361390411
Best Parameters:  {'n_estimators': 50}
```

And for Bank Marketing dataset is:

```
Accuracy:  0.8952824428853032
Best Parameters:  {'n_estimators': 1500}
```

Conclusion

After performing all the three classification algorithms on both the datasets, we can conclude that for:

1. GPU Runtime dataset, even though Decision Tree after pruning provides better accuracy rates than any other model(~98%), I will still prefer RBF SVM(~95%) to classify the data as it avoid any chances of overfitting.
2. Bank Marketing dataset, Decision tree after pruning provides better accuracy than any other model(~89%), so is the best way to classify Bank Marketing dataset.

In all the learning curves we can see that on increasing the training dataset size, the training accuracy goes down and validation accuracy goes up, showing our model is good and will be able to predict more datapoints accurately. But if the gap between the train and validation curve if large, this mean our dataset has high variance, which is the case in GPU runtime dataset, but not in the Bank Marketing dataset. But the issues of high bias and variance can be tackled by hyperparameter tuning in all the algorithms.

Additionally, I could improve the results by doing feature selections, experimenting with different folds of k-fold cross-validation, experiment with more kernels and hyperparameters and create validation curve and MSE curves for these models.

Citation

[Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

Variable Description

bank client data:

1 - age (numeric)

2 - job : type of job (categorical: "admin.","unknown","unemployed","management","housemaid","entrepreneur","student", "blue-collar","self-employed","retired","technician","services")

3 - marital : marital status (categorical: "married","divorced","single"; note: "divorced" means divorced or widowed)

4 - education (categorical: "unknown","secondary","primary","tertiary")

5 - default: has credit in default? (binary: "yes","no")

6 - balance: average yearly balance, in euros (numeric)

7 - housing: has housing loan? (binary: "yes","no")

8 - loan: has personal loan? (binary: "yes","no")

related with the last contact of the current campaign:

9 - contact: contact communication type (categorical: "unknown","telephone","cellular")

10 - day: last contact day of the month (numeric)

11 - month: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")

12 - duration: last contact duration, in seconds (numeric)

other attributes:

13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)

15 - previous: number of contacts performed before this campaign and for this client (numeric)

16 - poutcome: outcome of the previous marketing campaign (categorical: "unknown","other","failure","success")

Output variable (desired target):

17 - y - has the client subscribed a term deposit? (binary: "yes","no")