

Adult Census Income Classification with EBM&SVM



Overview

The prediction task is to determine whether a person makes over 50K a year according to other explanatory variables. The data is sourced from 'UCI ML Repository — adult dataset'. People are divided into two groups by the salary (≤ 50 and > 50). The influence on each variable on the salary is explored by applying EBM and SVM and then the performance of both methods are evaluated by time of training, confusion matrix as well as accuracy score.

The performance comparison result of these two methods is: From the aspect of training time, with same training data, the SVM is faster than EBM. While in terms of accuracy, EBM has better performance than EBM.

Motivation

The salary prediction is a quite classic classification problem. While the goal of this project is to explore the performance of 'Explainable Boosting Machine (EBM)' with 'Support-Vector Machines (SVM)' as a comparing method. On the base of same dataset and same way of splitting dataset, the time of training and prediction accuracy will serve as two key indicators to compare these two methods.

Data Description and methods Summary(EBM&SVM)

Data set description:

The data source is '<https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data>'. The extraction was done by Barry Becker from the 1994 Census database. A set of reasonably clean records was extracted using the following conditions: $((\text{AAGE} > 16) \ \&\& \ (\text{AGI} > 100) \ \&\& \ (\text{AFNLWGT} > 1) \ \&\& \ (\text{HRSWK} > 0))$.

The data set includes different explanatory variables of 32561 adult ("Age", "WorkClass", "fnlwgt", "Education", "EducationNum", "MaritalStatus", "Occupation", "Relationship", "Race", "Gender", "CapitalGain", "CapitalLoss", "HoursPerWeek", "NativeCountry", "Income"), among those variables, the object variable is "Income", which is classified into two conditions: ' ≤ 50 ' and ' > 50 '

Methods:

- **Description of EBM:**

InterpretML is an open-source python package for training interpretable machine learning models and explaining blackbox systems. One of the training method concerned in the project is 'Explainable Boosting Machine(EBM)' from the python package 'interpret.glassbox', which simplifies the interpretation of training result and analysis of impact of variables on the result.

EBM has both high accuracy and interpretability. EBM uses modern machine learning techniques like bagging and boosting to breathe new life into traditional GAMs (Generalized Additive Models). This makes them as accurate as random forests and gradient boosted trees, and also enhances their intelligibility and editability.

- **Description of SVM:**

In machine learning, support-vector machines (SVMs, also support-vector network) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting).

Models

Procedures:

Setup a classification experiment;

Replace the string value by numeric value (only for SVM method);

Dividing the data set into train data set and test dataset(test_size=0.2);

Explore the dataset

Train the Machine with train dataset;

Global Explanations: What the model learned overall(only for EBM method);

The schema below shows the importance of each explanatory parameter.

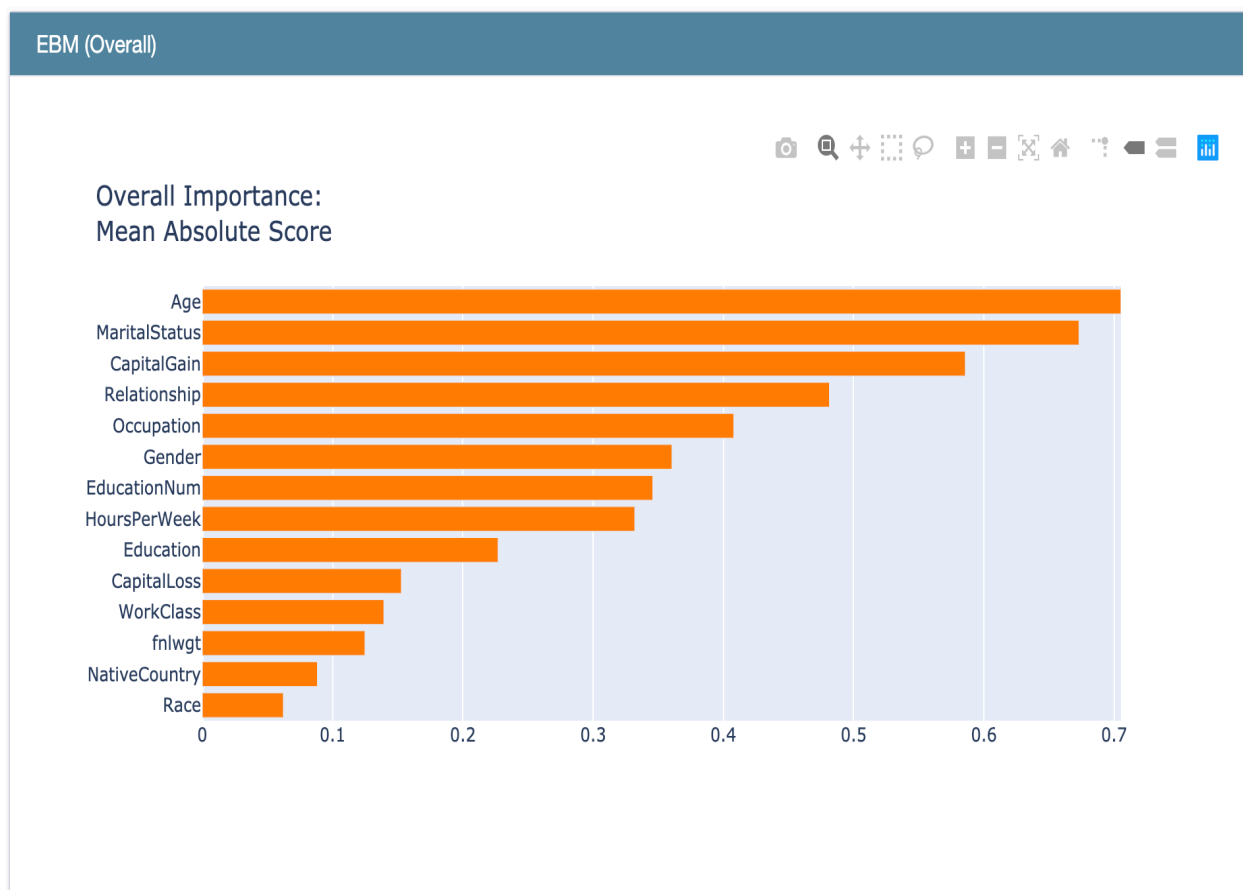


Fig1. Importance of each explanatory variable

Evaluate the performance in three ways

Time of Training; Confusion matrix; Accuracy score

The python scripts in detail in shown here: <https://github.com/Jinn42/Classifier-comparison-SVM-EBM/tree/master>

Summary of Results

For better visuality of the performance of each method, EBM and SVM, the result is shown in the following comparison table.

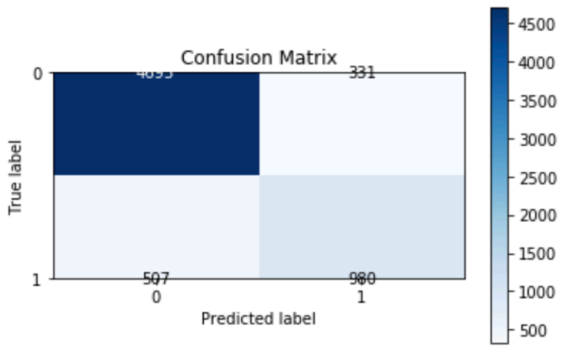
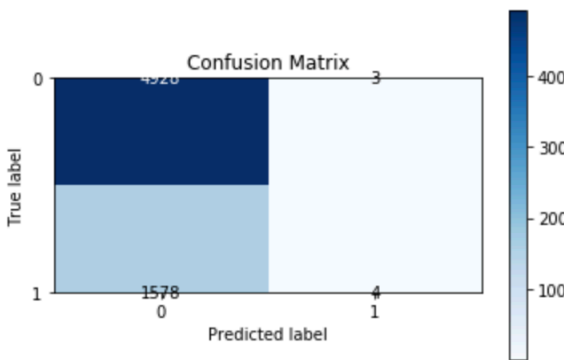
	Training time (s)	Accuracy Score	Confusion Matrix
EBM	136	0.87	 <p>Confusion Matrix</p> <p>True label</p> <p>Predicted label</p>
SVM	58	0.76	 <p>Confusion Matrix</p> <p>True label</p> <p>Predicted label</p>

Fig2. Performance result table