AdaBoost Classifier

Subham Roy
Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260

1 Overview

An ensemble is a composite model, combining a series of low performing classifiers with the aim of creating an improved classifier. Here, individual classifier votes and final prediction label returned that performs majority voting. Ensembles offer more accuracy than individual or base classifier. Ensemble methods can parallelize by allocating each base learner to different-different machines. Finally, you can say Ensemble learning methods are meta-algorithms that combine several machine learning methods into a single predictive model to increase performance. Ensemble methods can decrease variance using bagging approach, bias using a boosting approach, or improve predictions using stacking approach.

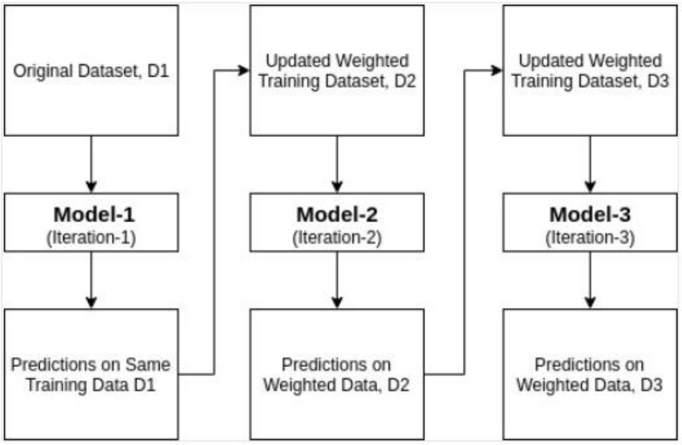
2 About AdaBoost Classifier

Boosting algorithms are a set of low accurate classifiers to create a highly accurate classifier. Low accuracy classifier offers accuracy better than the flipping of a coin. A highly accurate classifier offers error rate close to 0. Boosting algorithm can track the model who failed the accurate prediction. Boosting algorithms are less affected by the overfitting problem. The following three algorithms have gained massive popularity in data science competitions.

- AdaBoost (Adaptive Boosting)
- Gradient Tree Boosting
- XGBoost

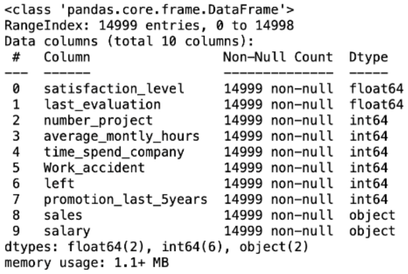
Let us understand how Adaptive Boosting or AdaBoost works. It works in the following steps:

- Initially, AdaBoost selects a training subset randomly.
- It iteratively trains the AdaBoost machine learning model by selecting the training set based on the accurate prediction of the last training.
- It assigns the higher weight to wrong classified observations so that in the next iteration these observations will get the high probability for classification.
- Also, it assigns the weight to the trained classifier in each iteration according to the accuracy of the classifier. The more accurate classifier will gain high weight.
- This process iterates until the complete training data fits without any error or until reached to the specified maximum number of estimators.
- To classify, perform a "vote" across all the learning algorithms you built.

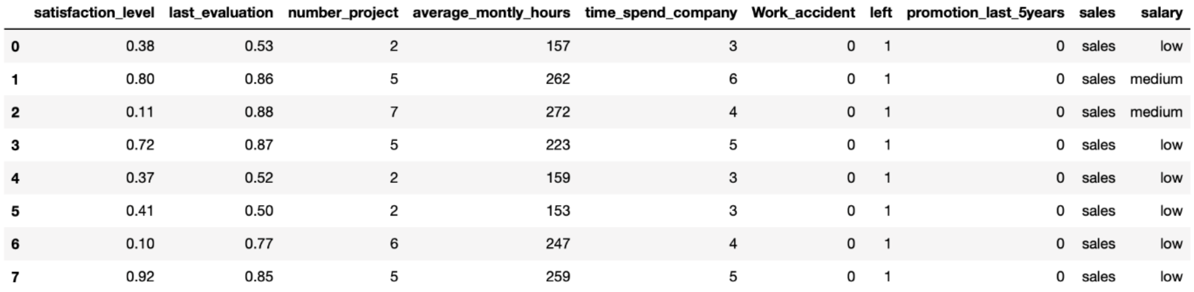

Original Dataset, D1
Model-1
(Iteration-1)
Predictions on Same Training Data D1
Updated Weighted Training Dataset, D2
Model-2
(Iteration-2)
Predictions on Weighted Data, D2
Updated Weighted Training Dataset, D3
Model-3
(Iteration-3)
Predictions on Weighted Data, D3

3 Experiment

The dataset used is Employee dataset which is used to group the employees having Low, Medium, and High salary. So, in this dataset, Salary is the target column. The dataset has 14999 rows and 10 columns. Out of 10 columns, the salary column is target variable and the others are predictors, which is also known as independent variables. This dataset does not have any null values and consists of float, object, and integer type columns. The target column is string type column. Data description consists of matrices like count, mean, max, min, std of every column in the dataset. For object type columns there will be no values in the data description.

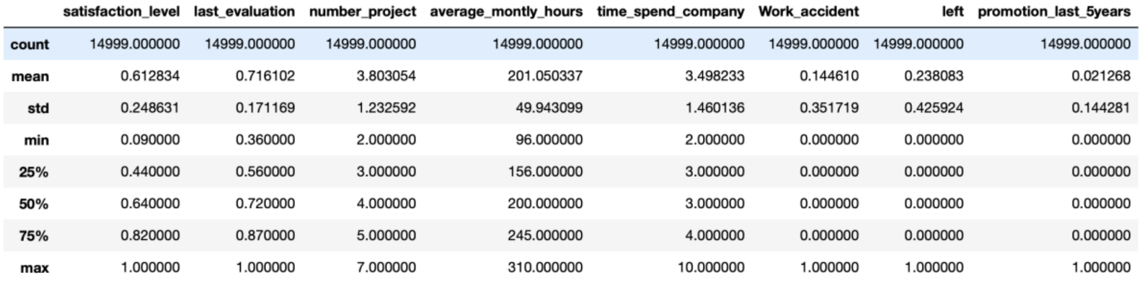


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   satisfaction_level     14999 non-null  float64
 1   last_evaluation        14999 non-null  float64
 2   number_project         14999 non-null  int64
 3   average_montly_hours   14999 non-null  int64
 4   time_spend_company     14999 non-null  int64
 5   Work_accident          14999 non-null  int64
 6   left                   14999 non-null  int64
 7   promotion_last_5years  14999 non-null  int64
 8   sales                  14999 non-null  object
 9   salary                 14999 non-null  object
dtypes: float64(2), int64(6), object(2)
memory usage: 1.1+ MB
```



	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low
5	0.41	0.50	2	153	3	0	1	0	sales	low
6	0.10	0.77	6	247	4	0	1	0	sales	low
7	0.92	0.85	5	259	5	0	1	0	sales	low

Fig: Few records in the dataset



	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years
count	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000
mean	0.612834	0.716102	3.803054	201.050337	3.498233	0.144610	0.238083	0.021268
std	0.248631	0.171169	1.232592	49.943099	1.460136	0.351719	0.425924	0.144281
min	0.090000	0.360000	2.000000	96.000000	2.000000	0.000000	0.000000	0.000000
25%	0.440000	0.560000	3.000000	156.000000	3.000000	0.000000	0.000000	0.000000
50%	0.640000	0.720000	4.000000	200.000000	3.000000	0.000000	0.000000	0.000000
75%	0.820000	0.870000	5.000000	245.000000	4.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	7.000000	310.000000	10.000000	1.000000	1.000000	1.000000

Fig: Dataset Description

The dataset has three types of features object, float, and integer type. Since we train the model on independent features only so, we split the dataset into two parts. We stored all the independent variables in X and target variable in y. For this analysis, we have taken 80% data for training and 20% for test. There are no null values in the dataset. So, I applied model on training data and checked the accuracy on test data. My model got an accuracy close to 89% when used AdaBoost classifier with Decision trees. Below are the images of plots observed during exploratory data analysis. With respect to the correlation matrix between the predictors and response, we drop the highly correlated variables. We can see that only columns 'salary_low' and 'salary_medium' are correlated because these are created from same column. So, we drop either 'salary_low' or 'salary_medium' from further analysis.

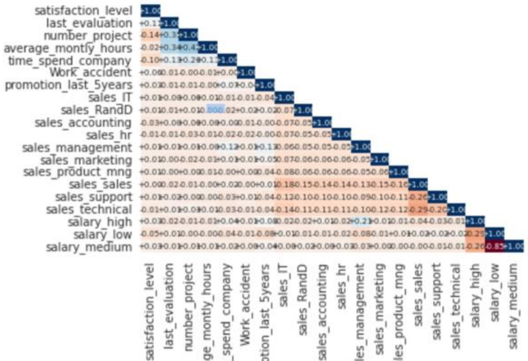
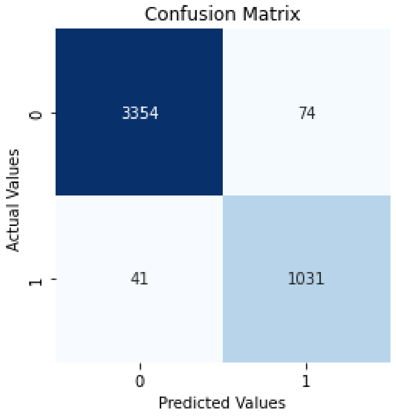


Fig: Correlation plot for all features

4 Analysis

Below are the figures shows the confusion matrix by the model. Model which we created in this analysis has got the accuracy 97%.



References

[1] Gareth James [at], Daniela Witten [aut], Trevor Hastie [aut, cre], Rob Tibshirani [aut], Balasubramanian Narasimhan [ctb], *Introduction to Statistical Learning, Second Edition*.

[2] Tom M. Mitchell, *Machine Learning: A multistrategy approach, 1997 Edition*.

[3] URL - *https://github.com/cchangyou/adaboost-implementation*; *https://www.datacamp.com/tutorial/adaboost-classifier-python*