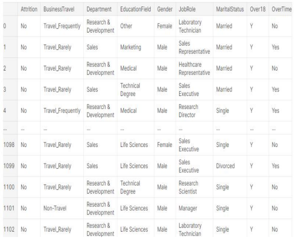
Ensemble Learning

Brief by Zizhe Wang

Introduction **- Random Forest (RF), GBDT** and **XGBoost** are all **Ensemble Learning** methods that improve the prediction performance of a single model by combining the predictions of multiple individual learners. Random Forest generates its individual learners in a parallel manner, while XGBoost generates them in a sequential manner. Both methods are widely used and effective in a variety of applications. Ensemble algorithms and big data are related because both can be used to improve the performance of machine learning models. Ensemble algorithms can be used to combine the predictions of multiple models trained on big data, and big data can be used to train more accurate and generalizable models using ensemble algorithms.

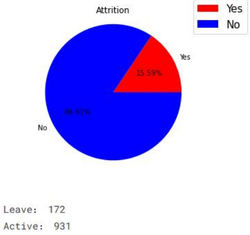
Methodology **–** The **Pre-processing:** (1)Check for missing and duplicate values. (2)Search and delete attributes with constant column values in the training set. (3)Output statistical values for numerical types. (4)Output statistical values for non-numerical types.

Feature engineering: (1)Feature selection: Select and remove the 5 least correlated features based on common sense and spearman correlation (generating a heatmap) to reduce noise and improve generalization ability (5 was the best choice tested). (2)Feature transformation: Numeric features are retained in their original state, ordinal categorical features are represented by a list of numbers with order-based representations, and unordered categorical features are first transformed into numbers using label encoding and then optionally represented using one-hot encoding based on subsequent requirements. (3)Feature dimensionality reduction: There are only 49 dimensions even with one-hot encoding, and fewer dimensions with label encoding, so there is no need for dimensionality reduction. Attempting to reduce the dimensions significantly decreased accuracy, so it was abandoned. (4)Feature standardization: Most tree models do not require standardization, but GBDT generally does. However, standardization had poor performance on this dataset, so this step was skipped.



	Attrition	BusinessTravel	Department	EducationField	Gender	JobRole	MaritalStatus	Over18	OverTime
0	No	Travel_Frequently	Research & Development	Other	Female	Laboratory Technician	Married	Y	No
1	No	Travel_Rarely	Sales	Marketing	Male	Sales Representative	Married	Y	Yes
2	No	Travel_Rarely	Research & Development	Medical	Male	Healthcare Representative	Married	Y	No
3	No	Travel_Rarely	Sales	Technical Degree	Male	Sales Executive	Married	Y	Yes
4	No	Travel_Frequently	Research & Development	Medical	Male	Research Director	Single	Y	Yes
...	...	...	...	...	...	...	...	...	...
1098	No	Travel_Rarely	Sales	Life Sciences	Female	Sales Executive	Single	Y	No
1099	No	Travel_Rarely	Sales	Life Sciences	Male	Sales Executive	Divorced	Y	Yes
1100	No	Travel_Rarely	Research & Development	Technical Degree	Male	Research Scientist	Single	Y	No
1101	No	Non-Travel	Research & Development	Life Sciences	Male	Manager	Single	Y	No
1102	No	Travel_Rarely	Research & Development	Life Sciences	Male	Laboratory Technician	Single	Y	No

Figure 1: non-numerical types

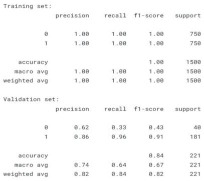


```
Leave:  172
Active:  931
```

Figure 2: pie chart of the trainData

Dataset: A pie chart of the traindata was used to view the distribution, and the dataset was split using train_test_split. Due to imbalanced data, SMOTE oversampling was used. After pre-processing, researcher use the **sklearn** library to call these algorithms and **visualize** the model tuning process by selecting the parameters that have the greatest impact on the model's performance. Then adjust these parameters to optimize the model's performance.

Results **-** After removing the five least correlated features, the **Random Forest** algorithm achieved the highest accuracy on the test set: 0.86274. The prediction results on the train and validation data sets are as follows:



```
Training set:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       750
           1       1.00      1.00      1.00       750

    accuracy                           1.00      1500
   macro avg       1.00      1.00      1.00      1500
weighted avg       1.00      1.00      1.00      1500

Validation set:
              precision    recall  f1-score   support

           0       0.62      0.33      0.43        40
           1       0.86      0.96      0.91       181

    accuracy                           0.84       221
   macro avg       0.74      0.64      0.67       221
weighted avg       0.82      0.84      0.82       221
```

Figure 3: prediction results based on RF

On the test set, the accuracy of both the GBDT and XGBoost algorithms can also reach 0.86274, but the number of n_estimators is significantly higher than the number of this parameter in RF.

Discussion

- Pro: Random Forests are relatively simple to use, requiring only the selection of a few parameters to begin training. They also have automatic feature selection, so manual feature selection is not required.
- Pro: GBDT and XGBoost are generally less sensitive to noisy data compared to Random Forests.
- Con: Random Forests are sensitive to noisy data and may overfit if the dataset contains a lot of noise.
- Con: GBDT and XGBoost are sensitive to the selection of hyperparameters, which can affect their performance. XGBoost has more hyperparameters to tune, making it more difficult to use for some users.

Conclusion **-** Ensemble learning methods, such as Random Forest, GBDT, and XGBoost, are useful tools for improving the performance of machine learning models. In this study, the Random Forest algorithm was found to be the best comprehensive performance at predicting outcomes on the test set. However, it is important to consider the pros and cons of each algorithm, as well as the characteristics of the dataset, when deciding which method to use. It is also important to carefully pre-process and engineer features in order to maximize the performance of the model.