Spammer Classifier Report

1. **Abstract**

ABC offers paid platform for sending millions of mails. This platform is abused by some of the spammer accounts. It is important to eliminate these accounts observing it’s mailing practices and responses to emails associated to these accounts. This report offers a solution to automate the process of classifying such accounts as spammers and assist in eliminating such accounts in a more efficient way.

1. **Model Selection**

This case is a classic example of classification. XGBoost classifier was used to classify spammer accounts. XGBoost classifier was preferred over other classifiers as it is ensemble method and offers robust solutions which can learn complex non-linear decision boundaries as well. Logistic regression was another method that can be used in such cases. But since the data had collinearity logistic regression was eliminated. Neural network is also a successful classifier but with limited data set it can easily overfit the data sets.

1. **Data Set**
2. Data comprises of 46 attributes and 21328 records.
3. Out of 46 attributes main 21 attributes are mentioned in terms of percent.
4. ‘Terminated’ is the label column (dependent variable) with two categories viz True & False.
5. **Data Exploration and Preparation**
6. Checked the data type of all the attributes maintaining uniformity across the data set. All the attributes are either float64 or integer except the label attribute which is Boolean type.
7. Removed *percent* attributes from the dataframe as these are redundant attributes. These attributes were not used as null values are converted as ‘0’ under percent attributes which is misleading conversion. Also, XGBoost model’s performance is not affected or boosted by monotonic transformations.
8. Checked the data imbalance to make sure higher accuracy of classifier reflects correct classification of both the categories of account holders. Data is balanced i.e. number of both categories of accounts were approximately equal.
9. Removed the attribute ‘smart\_send\_suppression\_bounce’ since this attribute doesn’t carry any information. All the values in this attribute were null.
10. Checked correlation among the attributes. Although XGBoost can handle correlated features only very highly correlated attributes like ‘other\_bounce’, ‘soft\_bounce’, ‘total\_injection\_count’ and ‘total\_injection\_count\_tracked’ were dropped.
11. Although data set has missing values, missing values were not imputed as XGBoost model can handle the missing values.
12. Again, Outliers were also not treated as XGBoost are robust to outliers in independent variables.
13. **Building Statistical Model – XGBoost**
14. The dataset was divided into Training and Validation Datasets with a ratio of 75% to 25% randomly.
15. Based on the training data DMatrix was initiated. DMatrix is internal structure used by XGBoost for memory efficiency and training speed.
16. Four different combination of hyperparameters were set specifying 4 different models. Hyperparameters were modified to achieve best model avoiding under or overfit. Target was to achieve minimum test error with lower train error.
17. Following hyperparameters were tried-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Learning rate** | **# Tree estimators** | **Maximum depth of tree** | **# subsamples** | **Train Error** | **Test Error** |
|  |  |  |  |  |  |  |
| Model1 | 0.01 | 100 | 7 | 0.8 | 0.1456 | 0.1619 |
| Model2 | 0.01 | 100 | 9 | 0.7 | 0.1235 | 0.1566 |
| Model3 | 0.01 | 100 | 10 | 0.7 | 0.1137 | 0.1579 |
| Model4 | 0.01 | 100 | 9 | 0.8 | 0.1326 | 0.1571 |

The best model selected is **Model2** as for other models test error increases with decrease in train error suggesting higher variance in other models.

1. **Interpreting the result of the XGBoost**

**1. Accuracy**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1. | Train Accuracy | 0.8899 |
| 2. | Test Accuracy | 0.8592 |

Since the data is balanced overall accuracy of the model can be trusted. This implies classifier classifies the overall account as spammers and genuine accounts with 85.9% accuracy.

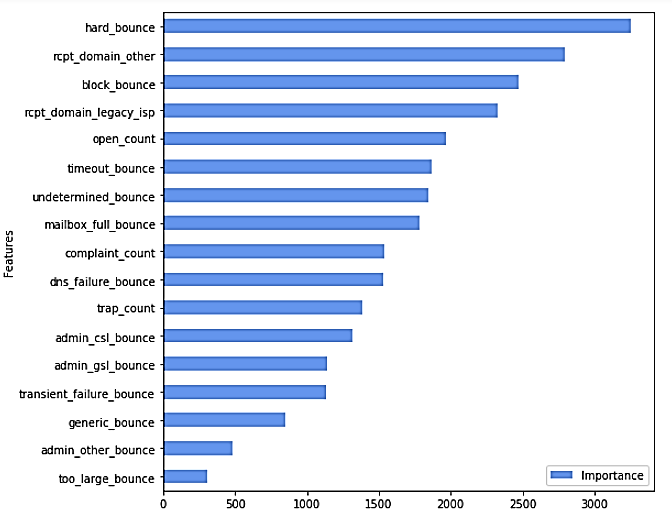
**2. Confusion Matrix**

|  |  |  |
| --- | --- | --- |
|  | **Predicted No** | **Predicted Yes** |
| **Actual No** | 2195 | 535 |
| **Actual Yes** | 216 | 2386 |

|  |  |  |
| --- | --- | --- |
|  | **Precision** | **Recall** |
| **FALSE** | 0.91 | 0.81 |
| **TRUE** | 0.82 | 0.91 |
| **Avg/Total** | 0.87 | 0.86 |

* From the result mentioned above, it was concluded that the classifier is predicting the spammers account correctly with the accuracy of 86%.
* Precision and Recall of the classifier are 87% and 86%. Precision is the fraction of relevant instances among retrieved instances. And recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances.
* 0.82 Precision for TRUE class implies, out of all the accounts predicted to be terminated by the classifier 82% are predicted correctly.
* 0.91 Recall for TRUE class implies, out of all the spammer account classifier is predicting 91% correctly.

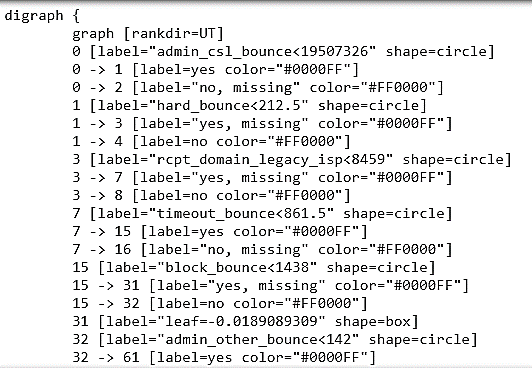
1. **Feature Importance**



Feature importance was plotted to identify important attributes to distinguish between spammers and genuine account holders. The attributes identified may aid to eliminate spammers accounts more efficiently based on business logic and intuition.

1. **Decision Tree**

* Decision tree was also plotted to visualize and interpret the logic to identify spammers more efficiently. Findings from the decision tree aligned with the important attributes identified in feature importance plot mentioned above.
* First few rules of the decision tree were observed, and the important attributes identified were used in framing initials rules to classify the accounts. For example, attributes like ‘hard\_bounce’, rcpt\_domain\_legacy\_isp’ were used initially to build the decision tree logic.



1. **Recommendation**

* Based on the findings of the report, it was concluded that *it is possible to automate the classification process which can assist the compliance team to mark and terminate the spammer accounts more efficiently*. Although the process cannot be fully automated as the classifier accuracy is about 86%.
* For the better business design, eliminating genuine accounts is bigger loss. Therefore, genuine accounts should not be marked for termination at any cost as losing a genuine customer is a bigger loss. To avoid this further manual intervention is required because classifier classifies few genuine accounts as spammers.
* Manual intervention can further be aided. Misclassified accounts marked for termination should be compared against the correctly classified accounts to identify the important distinguishing attributes. Manually cross checking the misclassified accounts across those attributes will speed up the process to classify all the accounts more efficiently.