ASSIGNMENT 3: PRESENTATION AND INTERVIEW

COSC2789 - Practical Data Science with Python

Abstract

This report analyzed the incident logging to identify the key features and build the machine learning model to assist the process to make the process execution efficiently. Moreover, some interesting ideas are pointed out for further investigation

Bao Le

S3595082@rmit.edu.vn

19-01-2023

Table of Contents

ABSTRACT SUMMARY

The report analyses the attributes represented the incident management process event log and use multiple machine learning algorithms to detect the most useful model to get a better background process of IT Company. The report studies the classification and clustering method to detect the best feature and best tuning algorithm's parameter for rating the priority of the incident when it comes to the pipeline. The comparisons will be given to state which is the highest score model and the best to be chose. On the other hand, it will provide some idea and interesting flow to improve the background process in further investigation.

INTRODUCTION

One of the most expensive processes of IT companies this day is incident management. Nowadays, IT Companies are mostly manual their incident solving steps which are reporting, logging and resolving. As a result, there are lots of tools, methods and services built to reduce those time-consuming manually steps [1]. The target of the report is to use the UCI ML repository to build machine learning model for assisting operators in the IM process. The ideas are to predict:

1. Incident's Priority
2. Closed code of the incident
3. Incident's completion time

For the time being, the report is focusing in predicting the incident's priority so that the operators can know the priority of the incident when it comes to place and assign to the right person.

METHODOLOGY

Data Exploration

There are 36 attributes. The attributes are divided into 3 groups which is the incident, the user and the problem. The incident group contains the attributes that describe the incident itself. the identification group contains the attributes that describe the incident, and the support group contains the attributes that describe the problem. The dependent variable is the resolved_at and closed_at. The independent variables are the rest of the attributes. There are 141712 instances in the dataset with 24918 different values of incident identifier. At first, there is no null variables but there are lots of question marks value in the dataset which can be detected as null.

This section will explain why the closed code can be predicted via the relationship among dataset's features. From the correlation matrix, the active status, made_sla and reassignment_count is highly correlated with each other. Hence, there is an idea that the dataset can be used to predict the closed code, which will help the operators when new case come, they will know what to do, choose what automation process is the best for the situation.

From the dataset, apart from divided into three groups, it can also separate into two groups, those who come first and those who the incident resolved. Because we have detected the closed code potential predict model, let focus on the attributes that available when a ticket is created. Those are:

1. caller_id: identifier of the user affected.
2. opened_by: identifier of the user who reported the incident;

3. sys_created_by: identifier of the user who registered the incident;
4. contact_type: categorical attribute that shows by what means the incident was reported.
5. location: identifier of the location of the place affected.
6. category: first-level description of the affected service.
7. subcategory: second-level description of the affected service (related to the first level description, i.e., to category).
8. impact: description of the impact caused by the incident (values: 1 High; 2 Medium; 3 Low).
9. urgency: description of the urgency informed by the user for the incident resolution (values: 1 High; 2 Medium; 3 Low),
10. notify: categorical attribute that shows whether notifications were generated for the incident.

Those attributes can be used for priority predicting model, but we don't know which one is used yet. This section studies and find the importance features for the model.

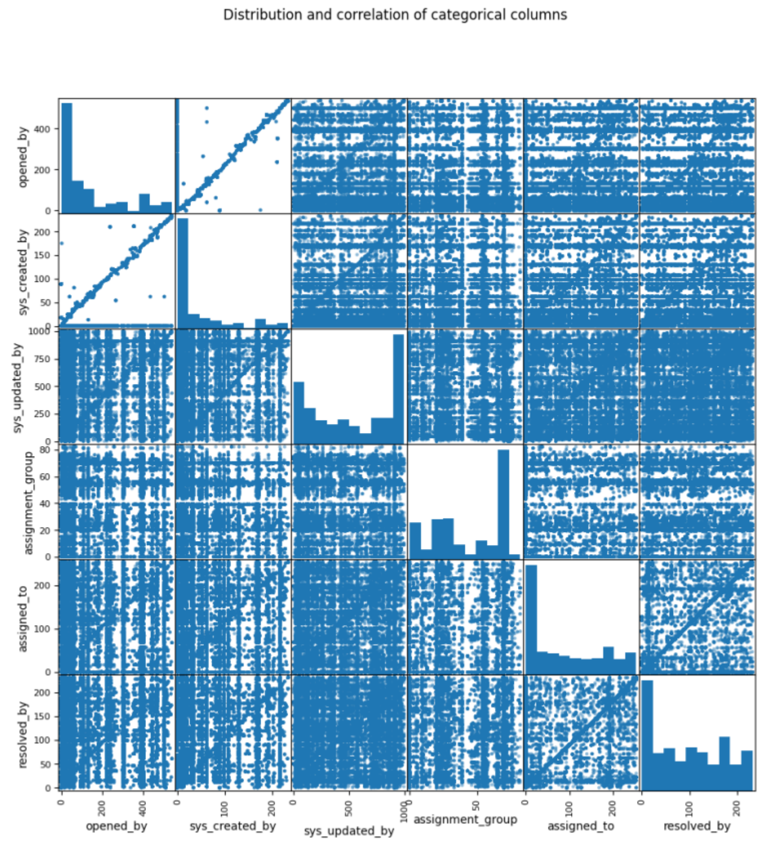


Figure 2: Distribution and scatter matrix of categorical columns

From the figure 2, we can see that:
1. The resolved by distribution diagram is normal distributed, however, the rest of the attributes are not normal distributed, they are imbalanced
2. The relationship between sys_created_by and opened_by is fully linear, seem like who open the ticket is the same person who create the ticket

3. As same as the sys_created_by and opened_by, the assigned_to and resolved_by are nearly linear, not 100% but 70%. It means that the person who resolve the ticket is the same person who assigned to the ticket. Let see other categorical attributes

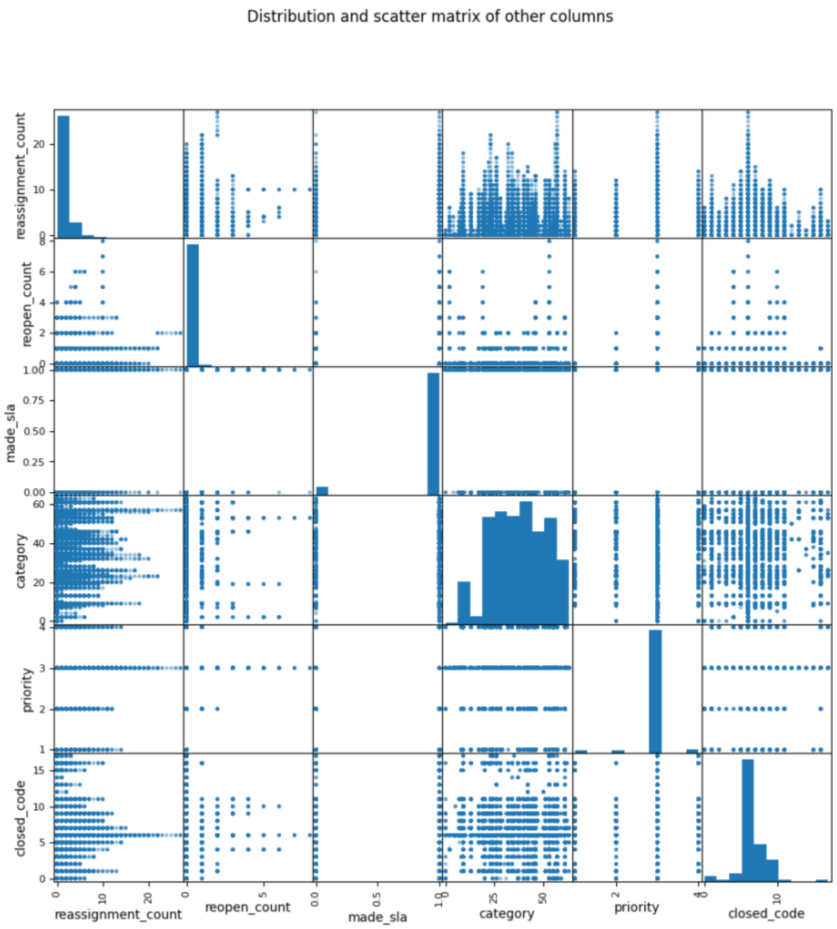


Figure 3: Distribution and scatter matrix of other categorical columns

As can be seen from figure 3, we can say that the data is imbalanced, most of the priority is 3. The made_sla distributed mostly for value 1. The reassignment_count and reopen_count is similar with each other. The priority will be studied next.

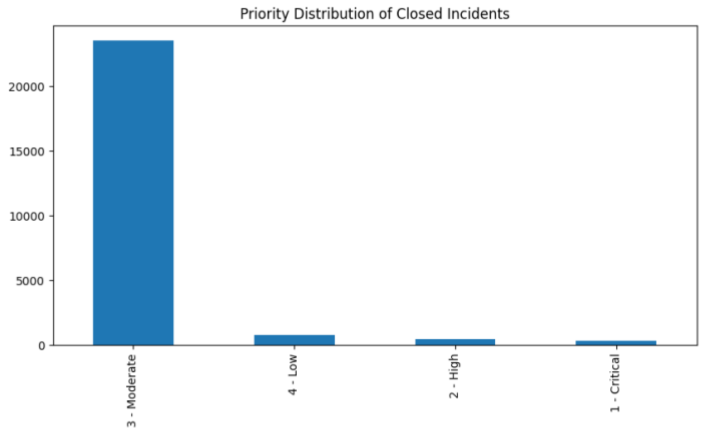
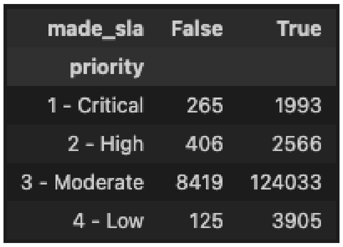


Figure 4: Priority Distribution of closed incident

From figure 4, it is clearly to state that the data is imbalanced, and the moderate priority of incident has the most closed case. As same as the closed state incident, the active incident also imbalanced, and most of them are moderate priority. One of the key aspects to detect the efficient of priority estimation is customer satisfaction.



made_sla	False	True
priority		
1 - Critical	265	1993
2 - High	406	2566
3 - Moderate	8419	124033
4 - Low	125	3905

Figure 5: Customer Satisfaction distribution among priority

As a conclude of choosing priority as predicting feature, from figure 5, the higher the priority, the less satisfied the customer. It can be stated that because of the complicated incident which need time and lots of work to completely solve the problem, so that it is difficult to satisfy the customer. On the second thought, the time-consuming might come from the wrong incident's solution or priority's estimation that makes a customer not happy at the beginning.

Classification

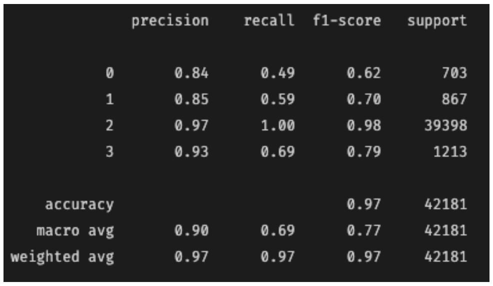
Select Features

As mentioned above, the priority needs to be predicted when the ticket comes, at that time, not all attributes are available. After investigating the features distribution, attributes use in the model are caller_id, opened_by, sys_created_by, contact_type, location, category, subcategory, u_symptom and made_sla. The made_sla is included to shown that priority is not related to the data after the process.

XGB Classifier

The XGBoost algorithm used for wide range of regression and classification offers a lot of hyperparameters to control the performance of the training model. It is used to deal with the imbalanced classification dataset by paying more attention to misclassification of the minority class for our priority. In this section, we will test and compare the normal XGBoost and Class Weight XGBoost output. Furthermore, a method for finding the best tuning parameter is also applied to improve and compare the XGBoost model [2].

After converting attributes to numeric and fit the xgboost model, the result is shown in figure 6. The accuracy score is very good with any machine learning model. However, because of the imbalanced dataset, it is not a good metric, the good metrics for imbalanced dataset is Macro F1 score which is a good all-around metric that balances precision and recall [3]. In this case, macro f1 score is 0.77. Based on the model, we have the confusion matrix (Figure 7), the accuracy of predicting moderate priority is the highest compared to the rest.



```
              precision    recall  f1-score   support

           0       0.84      0.49      0.62       703
           1       0.85      0.59      0.70       867
           2       0.97      1.00      0.98     39398
           3       0.93      0.69      0.79      1213

    accuracy                           0.97     42181
   macro avg       0.90      0.69      0.77     42181
weighted avg       0.97      0.97      0.97     42181
```

Figure 6: XGBoost Model result

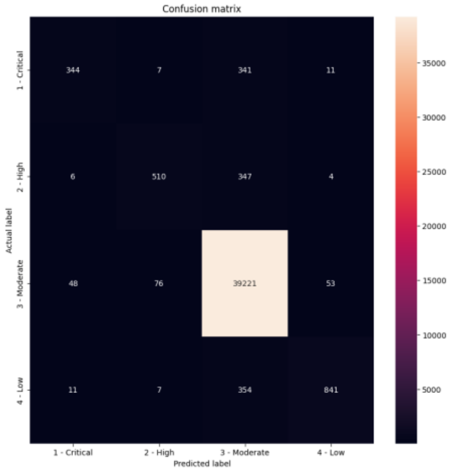
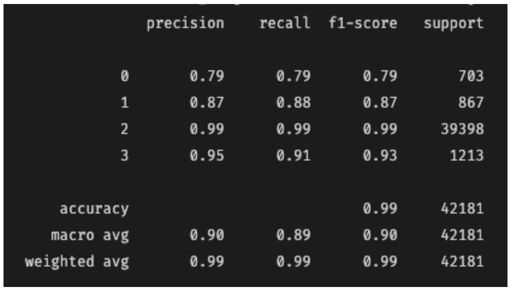


Figure 7: Confusion Matrix of XGBoost

In order to deal with the imbalanced dataset, the class weight method is applied to the training dataset to balance the output target. The result is not different from the old one, the reason might be the parameter for the model is not the best since basic parameter is applied without tuning. The tuning results are provided in the next section.

For the first tuning test, the result is illustrated in figure 8, the score is highly improved compared to the previous one and the best parameter found are max_depth is 8 and learning rate is 1.2. This is still the first tuning, there are plenty spaces to improve the model. Next the BayesSearchCV will be used to detect the best parameter. The BayesSearchCV is chose because the method reduces the finding time and minimize the resource use for wide range of tuning parameters. Figure 9 illustrates the result after using Bayesian for tuning XGBoost model, the score is higher than the grid search result. As a result, the best parameter for XGBoost is:

1. Learning Rate: 1
2. Max Depth: 34
3. N Estimators: 72

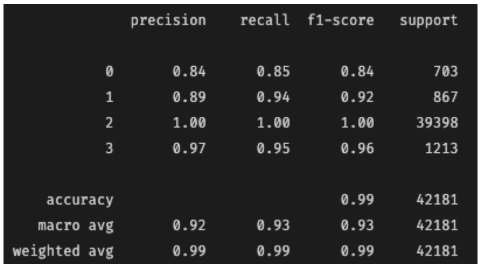


```
              precision    recall  f1-score   support

           0       0.79      0.79      0.79       703
           1       0.87      0.88      0.87       867
           2       0.99      0.99      0.99     39398
           3       0.95      0.91      0.93      1213

    accuracy                           0.99     42181
   macro avg       0.90      0.89      0.90     42181
weighted avg       0.99      0.99      0.99     42181
```

Figure 8: Grid Search tuning result



```
              precision    recall  f1-score   support

           0       0.84      0.85      0.84       703
           1       0.89      0.94      0.92       867
           2       1.00      1.00      1.00     39398
           3       0.97      0.95      0.96      1213

    accuracy                           0.99     42181
   macro avg       0.92      0.93      0.93     42181
weighted avg       0.99      0.99      0.99     42181
```

Figure 9: BayesSearch Tuning

At the end of the XGBoost model, the feature importance is exported to see the impact of choosing attributes with the target, priority.

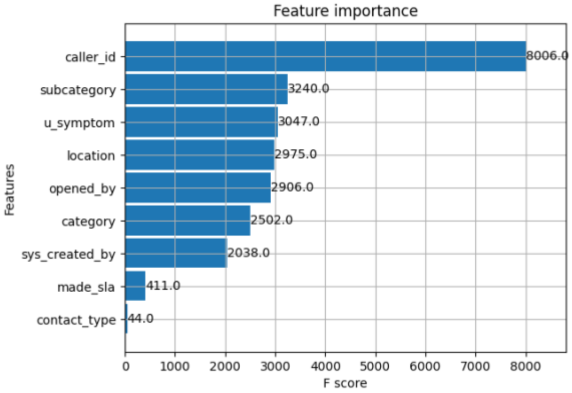


Figure 10: Feature Importance Score

Based on figure 10, the chosen features are corrected and the made_sla which used to show the unrelated features that are set after the incident reported is correct. Moreover, the caller_id is the most importance feature to detect the priority, which can assume that the user who detected the incident know how priority should the incident is, by this they can ranking the caller to prioritize their ticket since they know the priority of the incident best. The other feature is belonged to the type of the incident, the IT employee who receives is the second person know which priority the incident should be placed.

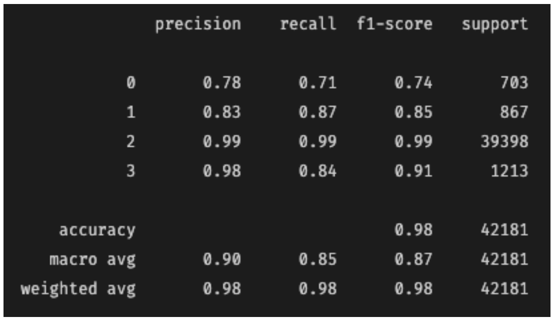
For future improvement, there are some steps can be done or studied:

1. Model Parameters Tuning
2. Use other method to implement imbalanced dataset such as oversampling or undersampling

Random Forest

Random forest uses many individual decision trees to operate as an ensemble. For each tree, the method splits out a class prediction and the most voted class will become the prediction model. In our case, because of the large dataset, it is reasonable to choose the random forest and use to compare with the XGB model.

At the first glance, the random forest gives better score than the XGB with no tuning parameters (Figure 11). As same as the XGB, Bayes Search will be applied to detect the best model for random forest. The result after tuning is not good as expected, even the weight score is high, the macro f1 score is the metric focusing on. It can be told that because of the imbalanced dataset, this is the best result of the random forest that can be given which shows the actual status of the dataset.

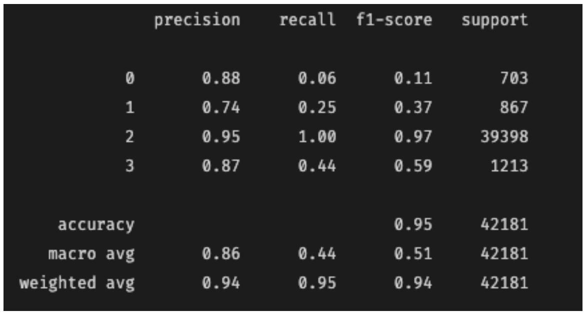


```
              precision    recall  f1-score   support

           0       0.78      0.71      0.74       703
           1       0.83      0.87      0.85       867
           2       0.99      0.99      0.99     39398
           3       0.98      0.84      0.91      1213

    accuracy                           0.98     42181
   macro avg       0.90      0.85      0.87     42181
weighted avg       0.98      0.98      0.98     42181
```

Figure 11: Random Forest Result



```
              precision    recall  f1-score   support

           0       0.88      0.06      0.11       703
           1       0.74      0.25      0.37       867
           2       0.95      1.00      0.97     39398
           3       0.87      0.44      0.59      1213

    accuracy                           0.95     42181
   macro avg       0.86      0.44      0.51     42181
weighted avg       0.94      0.95      0.94     42181
```

Figure 12: Random Forest after tuning

Comparison

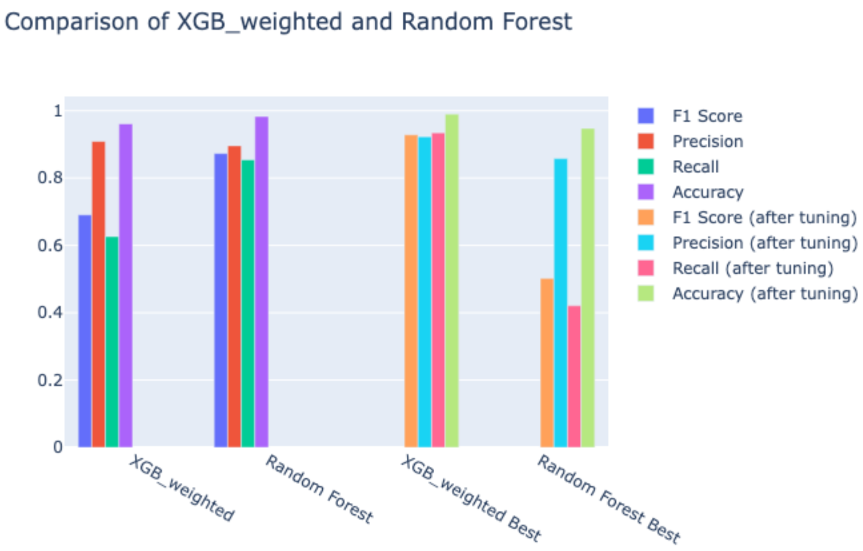


Figure 13: Comparison of XGB and Random Forest Model

Figure 13 shows the score of both model before and after tuning, the XGB using weight class with Bayes Search CV is the most balance model that gives acceptable value of metrics. It can be said that the XGB handling imbalanced data is good enough to cover the dataset. But there are spaces to improve the other models with other data handling method.

Clustering

Feature Selection

Based on the feature importance from the classification methods, the best 5 features of the dataset are chosen for the clustering method. Those are:

1. Caller_id
2. Opened_by
3. Subcategory
4. Sys_created_by
5. Category

However, there is an issue with the imbalanced dataset, the clustering method performs poorly with those type of data. The scoring might not as expect, but from the study, the weakness can be identified and discussed for future work.

For this section, the SMOTE will be used for handling imbalanced dataset.

K-Means

Before applying K-Means method, the K need to be detected for the best result, hence, the elbow chart is plotted to track the best K for the data training (Figure 14). From the Elbow method, the best number of clusters for the data training is 4.

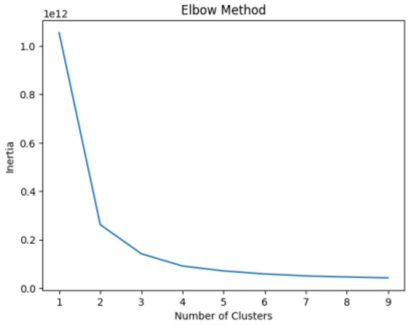


Figure 14: Elbow Method

Fitting the k=4 with the dataset after SMOTE, we have a result with 0.29108 accuracy when matching with the target. The result is poor, as a result, other clustering method is implemented to see if there is any mistake from the process or just a data itself.

DBSCAN

Unlikely the K-mean, now the train dataset included the target which is priority data and with the help of Nearest Neighbor method, the efficient parameter can be predicted (Figure 15). Eps is 0.4 and min_sample is 4 are chosen for the DBSCAN method. In this method, the silhouette score is used for detecting if it is good model or not. If the silhouette score is larger than 0.6, it is a good model. In our dataset, the score only 0.219 which can detect a bad one.

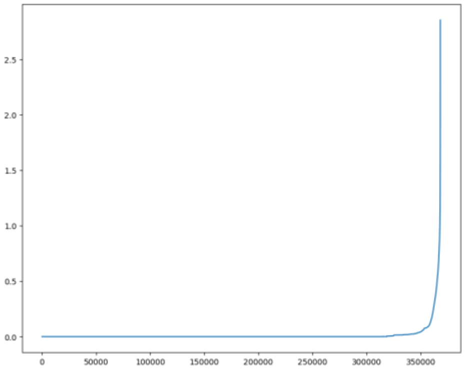


Figure 15: Nearest Neighbor

Future Solution

The result of kmean and dbscan are disappointed, even using methods for tracking best tuning parameter. The imbalanced dataset is one of the reasons and with SMOTE methods, it is not fully solved. A lot of handling imbalance class methods are investigated and applied. This section studies one of the undersampling method for imbalanced class which shown in figure 16 [4].

The method first use Silhouette value method to detect the K cluster and use the CLARA algorithm to cluster the negative class data. Then picking random sample and adding the positive class. After that using C4.5 classifier algorithm to evaluate confusion matrix. This can be used in the incident management dataset to improve the result of the clustering.

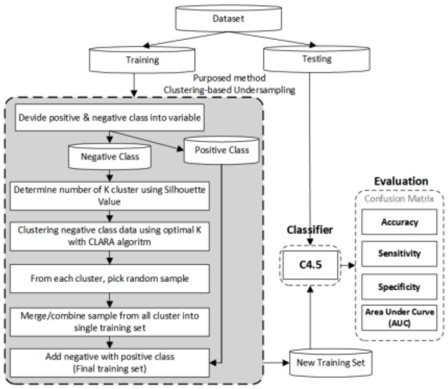

Dataset
Training
Testing
Purposed method
Clustering-based Undersampling
Devide positive & negative class into variable
Negative Class
Positive Class
Determine number of K cluster using Silhouette Value
Clustering negative class data using optimal K with CLARA algoritm
From each cluster, pick random sample
Merge/combine sample from all cluster into single training set
Add negative with positive class (Final training set)
New Training Set
Classifier
C4.5
Evaluation
Confusion Matrix
Accuracy
Sensitivity
Specificity
Area Under Curve (AUC)

Figure 16: Clustering-based undersampling procedure

RESULT

The classification model performance is better than the clustering. But with the future work, it is optimistic that the clustering can perform as same as or even better than the classification.

DISCUSSION

The distribution of the dataset is one of the most important keys for the performance of the machine learning model. In the most ideal case, the model is easy to implement and build with balance and well distributed dataset. In the worst case, there are necessary to apply some methods to handle imbalanced, right skewed or left skewed dataset. From the incident management event log dataset, some keys are pointed out:

1. The higher the priority, the lower satisfaction of the customer is.
2. Most of the satisfaction customers come from closed incident state.
3. The faster the case solved, the more satisfier the customer are

There are some ideas can implement machine learning model and improve IT process which are [5]:

1. Incident completion time
2. Closed Code
3. Relative between assigned to and resolved by attribute

For the first two ideas, the model will help estimation process and the customer does not have to wait so long for the solution of the incident. The relative between assigned to and resolved by can be used to detect the right individual to assign the problem to. Moreover, employee's performance can be ranked via this.

CONCLUSION

In conclusion, by comparing result of classification and clustering model, the XGBoost Classifier performs the best. It gives the best metrics for the incident management dataset. However, the clustering performance is improvable with the right handling imbalanced dataset method. Beside priority prediction, the IT company can apply more machine learning idea for the incident management process. It is strongly proved that machine learning is one of the important technique in industry.

REFERENCE

[1] A. S. Gillis, "Tech Target," August 2018. [Online]. Available: https://www.techtarget.com/searchitoperations/definition/IT-incident-management.

[2] J. Brownlee, "Machine Learning Mastery," 21 August 2020. [Online]. Available: https://machinelearningmastery.com/xgboost-for-imbalanced-classification/.

[3] J. Brownlee, "Machine Learning Mastery," 8 January 2020. [Online]. Available: https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/.

[4] W. Nugraha, "Clustering Based Undersampling for Handling Class Imbalance in C4.5 Classification Algorithm," *Journal of Physics: Conference Series,* 2020.

[5] Z. JIN, "Kaggle," 2021. [Online]. Available: https://www.kaggle.com/code/zhefeijin/incident-management/notebook.