IT incident management event log analysis

Abstract

IT incident management systems are key productivity tools that help companies manage digital workflows for enterprise operations. Incident management systems enable effective and structured workflows for technical support, otherwise very difficult to achieve in environments with a medium to large customer base. The operation of such systems produces rich information about the support operating model, which companies typically find valueable for audit purposes. Such audit data contains valuable information for the analysis and optimization of the support operating model. In this study, we analyse an audit dataset of an IT incident management system in an attempt to find insights about the support operating model, and build a predicting model of the incidents time to resolution.

Dataset

The dataset object of study is the Incident management process enriched event log Data Set, from the UC Irvine Machine Learning Repository. This is an event log of an incident management process extracted from the audit system of an instance of the ServiceNow platform used by an IT company.

The dataset contains information of 141.712 events, corresponding to 24.918 incidents. Each event is an instance with 36 attributes: 1 case identifier, 1 state identifier, 32 descriptive attributes, 2 dependent variables.

Attributes used to record textual information of the incidents and events are not included in the dataset.

Missing values are considered as unknown information.

Preprocessing

The raw data is available in a structured `csv` file. Basic preprocessing is done to:

- Map categorical textual values to numerical values representing categories
- Ensure data types control
- Build derived variable time_to_resolution. Instances with negative values of this variable are discarded, considered bad data quality. For convenience in terms of data type, this is eventually expressed in seconds.

A support file with metadata for preprocessing, `preprocmd.json`, is provided for reproducibility.

Data segmentation

Train and test datasets have been created from the preprocessed data, with a 80-20 random split.

Objectives

A priori, the objectives of the study are:

- Define a prediction model for the time to resolution
- Select features that play an important role in the time to resolution
- Perform unsupervised classification of the events, with the aim of supporting the feature selection and/or defining prediction submodels per group that are more precise than an overall model

Exploratory analysis

After preprocessing, the descriptive variables have either numerical or boolean values. The target variable time_to_resolution is expressed as a timedelta.

	number	incident_state	active	reassignment_count	made_sla	caller_id	...	time_to_resolution
0	INC0000045	0	True	0	True	2403	...	0 days 10:13:00
1	INC0000045	6	True	0	True	2403	...	0 days 10:13:00
2	INC0000045	6	True	0	True	2403	...	0 days 10:13:00
3	INC0000045	7	False	0	True	2403	...	0 days 10:13:00
24	INC0000062	0	True	0	True	3765	...	0 days 08:53:00

Table 1. Incident Event log excerpt

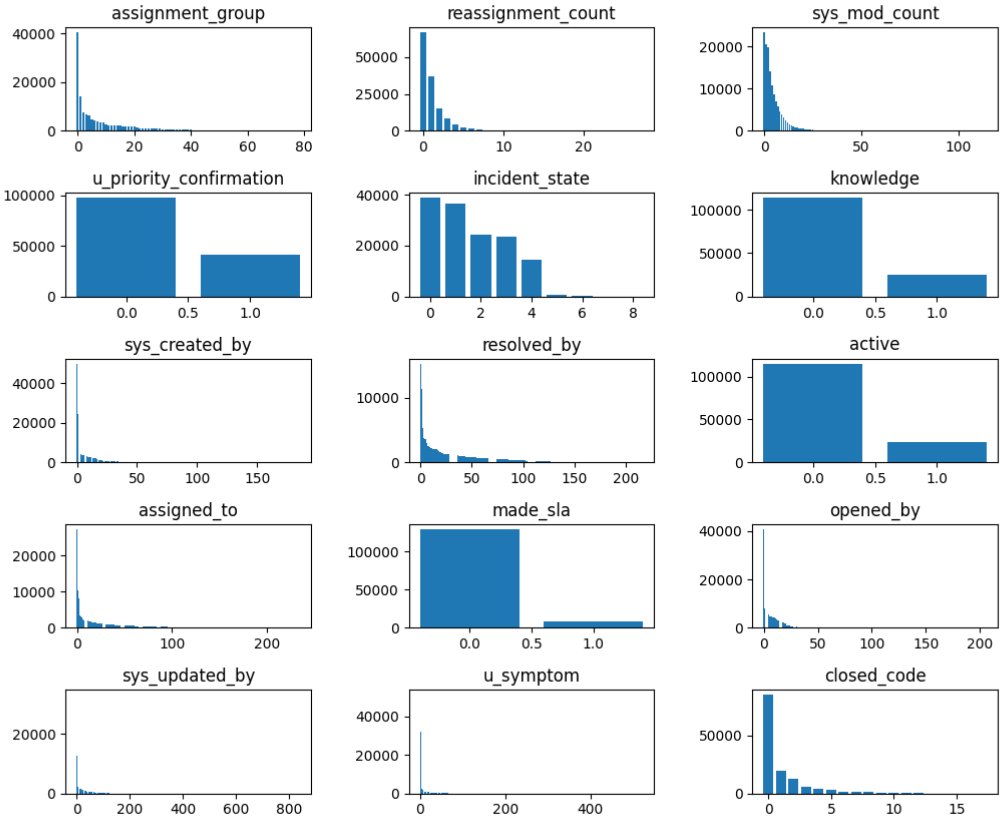
The correlations of each descriptive variable with the dependent variable can give an initial idea of what features may be driving the time to resolution.

variable	corr(., time_to_resolution)
assignment_group	-0.157981
reassignment_count	0.155982
sys_mod_count	0.136565
u_priority_confirmation	-0.124181
incident_state	-0.120124
knowledge	0.1145
sys_created_by	0.107133
resolved_by	0.0992228
active	0.0878472
assigned_to	0.0708714
made_sla	-0.0620198
opened_by	0.0533508
sys_updated_by	-0.0530903
u_symptom	-0.0511794
closed_code	-0.0395071

Table 2. Descriptive variables with the highest correlations to the dependent variable

The distribution of the values of the descriptive variables may provide some insights on variables that may not be too informative.

Figure 1. Frequencies of the descriptive variables most correlated with the time_to_resolution



We can observe that variables `sys_updated_by` and `u_symptom` concentrate heavily on one or few values, they are probably not very relevant.

First predictor and feature selection: Random forest regressor

A random forest regressor model has been fit to predict the time_to_resolution, with 100 trees and mean squared error as the criterion to measure the quality of the split.

After fitting the model using all original features, we pay attention at the features importance. We will try to reduce the model dimensionality by reducing the number of features. The approach will be to fit a random forest with the same parameterization and train data set, varying the number of features considered in the model. The number of features will vary from 15 to the total number of 30, and we will evaluate the evolution of the model score.

The feature selection approach described requires of some criterion for ranking the features. Two criteria will be compared: pair-wise correlation with the target variable and features importance of the random forest fit with all features.

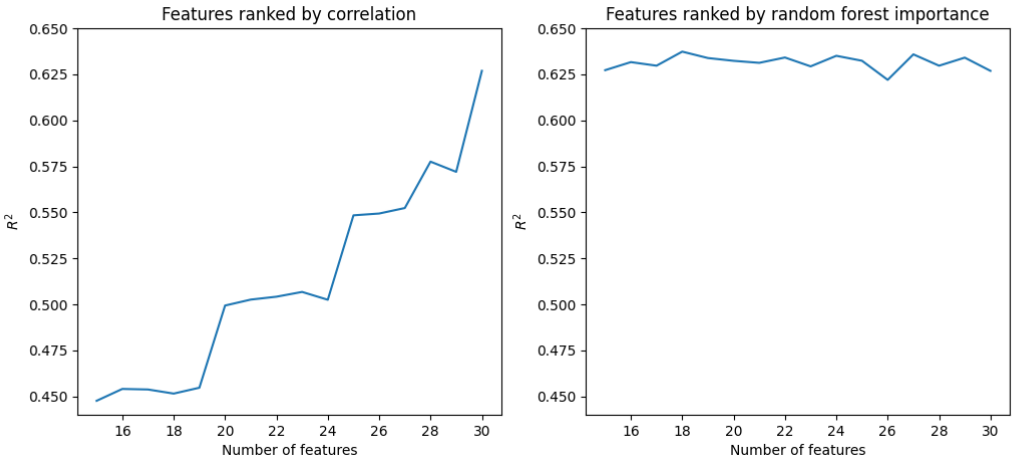
Most correlated with target variable	**Most important random forest features**
assignment_group	caller_id
reassignment_count	resolved_by
sys_mod_count	location
u_priority_confirmation	sys_mod_count
incident_state	subcategory
knowledge	assignment_group
sys_created_by	opened_by
resolved_by	incident_state

Most correlated with target variable	Most important random forest features
active	category
assigned_to	u_symptom
made_sla	sys_updated_by
opened_by	assigned_to
sys_updated_by	closed_code
u_symptom	sys_created_by
closed_code	reassignment_count

Table 3. Descriptive variables with the highest correlations to the dependent variable and features with highest importance in random forest built with all features (top 15)

The following figures show the evolution of model scores as the number of features of the model increase, comparing the two criteria for ranking the importance of the features.

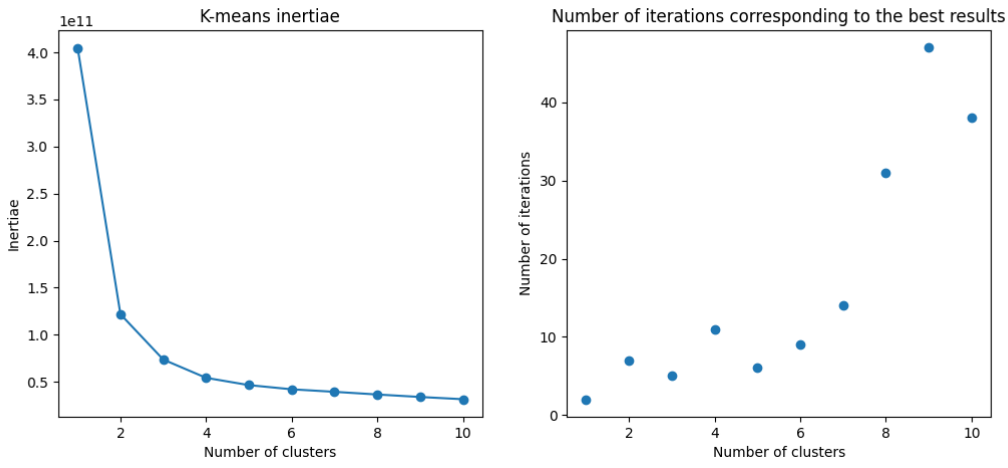
Figure 2. Evolution of the random forest scores



Unsupervised classification: k-means clustering

Last, an unsupervised classification model was applied to the data with the aim of finding patterns that could possibly describe the data and help defining more accurate local predictors. A k-means clustering model was run on the complete dataset, varying the number of clusters and looking into the model inertiae to determine the optimal number of clusters to classify the data into.

Figure 3. K-means metrics



The inertiae graph presents the expected elbow shape. After classifying into four clusters, the inertiae curve flattens,

showing that the average distance to the cluster centroids does not improve significantly by increasing the number of clusters. Four is therefore the natural choice of number of clusters to classify the data into.

Once the data points are classified into four groups, we now fit a random forest regressor in each of the four clusters and look at the importance of the features on each of the clusters. The expectation is to find different important features across different clusters. However the results were not really conclusive and this analysis was not pursued further.

Conclusion

The pair-wise correlation of features to target variable is a poor and naive approach for feature selection. This is expected in a highly dimensional dataset. The fact manifests clearly when comparing the quality of the random forest predictors with reduced dimensionality, when the features have been selected based on the correlation to the target variable, as compared to the case when the features are selected based on the features importance of the maximal random forest.

The highest scores of the random forest predictions are not achieved when using all the descriptive features in the model. Some features have little variability, and values repeat over events related to the same incident or correlate significantly with other features. As a result, there are cases where features are not informative and may just introduce noise in the prediction models.

It is therefore meaningful trying to reduce the dimensionality of the random forest regressor models, not just for economy, but also to reduce noise that affects the quality of predictions.

References

Incident management process enriched event log Data Set
UC Irvine Machine Learning Repository
code repository
Amaral, C. A. L., Fantinato, M., Peres, S. M., Attribute Selection with Filter and Wrapper: An Application on Incident Management Process. Proceedings of the 14th Federated Conference on Computer Science and Information Systems (FedCSIS 2018), pp. 679-682, 2018. [Web Link]