Data Analysis for ITSM Incident Management Process

Akshaya Mohan
Information Management
Syracuse University
Syracuse
akmohan@syr.edu

Lavnish Talreja
Information Management
Syracuse University
Syracuse
lptalrej@syr.edu

Akhil Nair
Applied Data Science
Syracuse University
Syracuse
anair05@syr.edu

***Abstract*—The objective of this project is to correctly predict if a system or user issue is resolved, time taken to resolve those tasks . We will also be looking at important attributes that will help us predict the reasons behind the time taken for an incident to be resolved and closed.**

I. Introduction

ITSM defines an incident as an interruption of a service. IT firms usually work by using test management tools for requirement gathering and solving system related or user related problems using those tools. But these tools need to have a system where one could find what issues needed to be resolved, priority of those issues, system or user related issues and the most important parameter is when those issues are resolved. Incident management data set helps us in understanding what issues are prioritized and when they are resolved. The system can predict if the high priority issues are resolved before a medium or a low priority issue and so on. By predicting when those issues were supposed to be resolved and when they were, we can find whether employees are resolving those issues considering the urgency.

II. Domain Knowledge

An IT incident is a problem that blocks an organization from their IT services. Incident Management process helps the company in keeping track of their incidents/tickets so that they can be handled by their employees. An incident is created when an end user faces an issue and ends when the issue is handled by the service team.

Information about the incident is collected by the team when the ticket is created by the user. In the next stage, the incidents are distributed and assigned to employees for analysis and resolution. Without proper prioritization

the incidents are not prioritized, and the timeliness is not met. This can result in business outages particularly with higher priority issues.

III. Dataset Description

We have a separate collection for train and test data. The training dataset consists of a total of 36 assessment parameters. The group of independent variables is a mix of categorical and numerical type of data. The dataset consists of 1 case identifier, 1 state identifier, 32 descriptive attributes and 2 dependent variables. There are 28 categorical, 5 continuous numerical and 3 discrete numerical variables. Most of the independent variables in our data describe various specifications of the incident that happened such as “syscreated”, “sysupdated” and priority status of that incident to be resolved etc. Remaining attributes inform us about reasons the incident was caused by and whether it was resolved, if yes then when etc. This dataset contains no NA’s, but it has missing values. They are defined as a string value named “?”. 17 of those 33 independent attributes contain “?” string values which must be treated.

IV. Dataset Preprocessing

When you start working with a dataset it is important to preprocess the data in order to make sure that the analysis is accurate. It is important to scale and normalize the data before introducing them to the models.

Initially, our dataset had no missing or NA values. There were a few special characters which we dealt at first. We converted the ”?” special character into NA values so that they can further be substituted with the mean/mode of their respective column.

Heatmap is a data visualization technique which helps us understand the magnitude of data against each column. By plotting the heatmap we analyzed few columns having most number of NA values and hence didn’t contribute much to our analyses. So we dropped the following columns ’cmdb_ci’, ’problem_id’, ’rfc’, ’vendor’, ’caused_by’ from our dataset.

We detected abnormal values that differed too much from the other values in the dataset. We removed the outliers from the columns and moved to further analysis.’reopen_count’, ’sys_mod_count’ were the numerical attributes that had missing values. They were replaced with the mean of their

respective column. 'caller_id', 'opened_by', 'sys_created_by', 'sys_created_at', 'location', 'category', 'subcategory', 'u_symptom', 'assignment_group', 'assigned_to', 'closed_code', 'resolved_by', 'resolved_at' were the categorical attributes found to have missing values in our dataset. Since these attributes were essential for analysis we decided to replace them with mode of their column instead of dropping them.

The dataset now is preprocessed and is consistent without any missing values and outliers. The next step implemented in this process was LabelEncoding. The dataset contained categorical values but the algorithms we used expected numerical data in order to provide accurate prediction results. The labelencoding concept is used to convert the categorical values to numerical values where each category is assigned a numerical value based on the class of data.

V. Exploratory Data Analysis

Exploratory Data Analysis is one of the most important practices used in a data science project. It is all about statistical modeling and visualization practices used to reform and analyze data to get meaningful insights. Filtering reformed data also helps in getting out essential aspects of the data for further analysis.

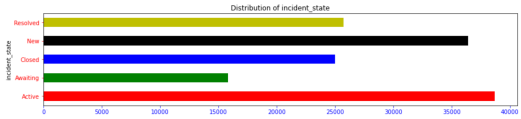

Fig. 1. Incident state

Fig. 2. Closed Code

First step which we did was analyzing the shape of the data frame and calculating NA's Next step was to find out the distribution of state of incidents in the given dataset. As there were no NA's but consisted of data inconsistency, we decided to remove it by replacing it with NA's and decided to remove columns that had high number of NA's. Drawing a heatmap provided us with those columns and we decided to remove those outliers. For categorical columns, we decided to mode values and for numerical columns, we decided to replace it with mean values. Filtering out only closed incidents for our analysis as the incident can be in other state for multiple time and closed will be the final state which will give us the unique incident. After filtering out the dataset, it could really help us in providing answers to questions such as what type of incidents were getting reopened? After performing further analysis, we could find that close code 6,7,8,9 are the

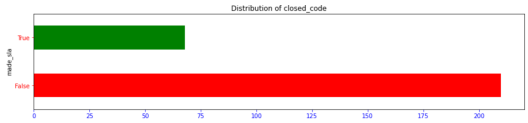

Fig. 3. Distribution of Closed Code

major ones with which the incidents are getting closed. We performed similar analysis with SLA to check if these reopened tickets had any similar patterns. Further analysis proved us that SLAs are missed in most of the reopened incidents so if the tickets are getting reopened then there is high chance that SLAs will be missed.

Next, we looked at what kind of incidents are missing the SLA's. Further Analysis revealed that round 37% incidents are missing SLAs so now let us see what kind of incidents usually miss SLAs. There was a very clear pattern that if SLAs mostly get missed for critical and high priority incidents whereas SLAs are met for Moderate and Low priority incidents. Next, we decide to see which features are important to predict the close code.

Next, we decide to look how closed incidents are impacted performs when it comes to priority. We could see that Critical and high impact incidents were solved first and low priority and impact incidents were resolved at lost which is a positive outcome.

VI. Machine Learning Algorithms

A. Decision Tree Algorithm

Decision Tree is a category of supervised learning algorithm. This algorithm works well for both continuous and categorical variables which is why we have implemented it as one of our models. Each of the leaf nodes represent a class label and branches are the features that lead to those label decisions.

We have implemented the algorithm to classify the priority level of the incidents based on the impact it has on the institution and the urgency to deal with it. In our model, the leaf nodes represent the incident state to check for urgency.

Gini impurity is used to predict the probability of an incorrect classification. We can predict if the instance is classified according to the distribution of class labels from the dataset. The likelihood of classification is 0 if the dataset is 0. If the training dataset has a sample of different mixtures, the likelihood will be high.

The dependent variable for our model is the 'priority' status and the X variables are used to determine the priority of the incident state. Hyperparameters were tuned to optimize the model architecture and reduce the loss associated with the model.

The decision tree model gave us the best accuarcy score of 98.65%. The best hyper parameters were : c-value=0.1, Penalty=l1,solver=liblinear.

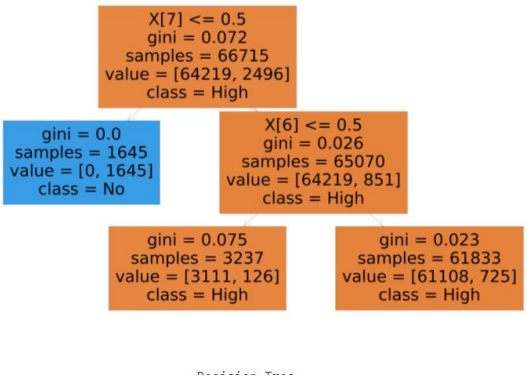

Fig. 4. Decision Tree model

B. Logistic Regression Model

Logistic Regression is a statistical model that incorporates logistic functions to model the dependent variable. The binary logistic model uses the dependent variable with two values 'High/Low priority'. The model takes in the input and predict an output between 0 and 1. We have used a receiver operating characteristic curve to depict the performance of a classification model. Our ROC curve provides two parameters namely True Positive Rate(TPR) and False Positive Rate(FPR).

TPR = TP/TP+FN

FPR= FP/FP+TN

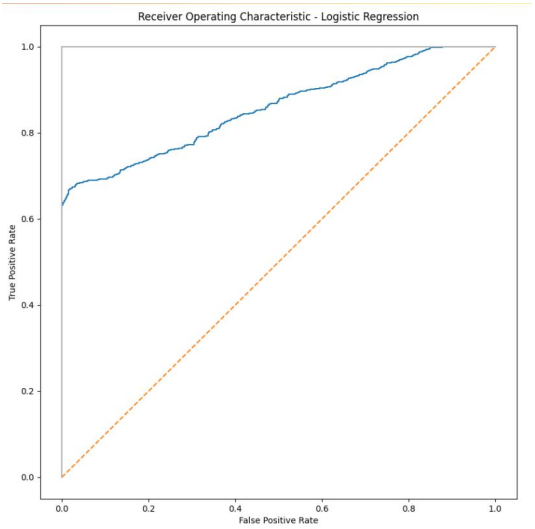

Fig. 5. Logistic Regression model

logistic regression also gave us the best accuracy score. The best hyper parameters were : c-value=0.1, Penalty=l1,solver=liblinear. The model gave us the accuracy score of 98.5%

C. Support Vector Machine

Support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

One of the advantages of using an SVM model for image classification is that it uses a kernel trick to map the input data onto a higher dimensional plane to establish decision boundaries that are not achievable in 2-dimensional space. This is great for a classification problem that is not linearly separable.

Feature extraction in the case of SVMs is

very important. The main objective in SVM is to find the optimal hyperplane to correctly classify between data points of different classes. The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameter is the inverse of the radius of influence of samples selected by the model as support vectors. The C parameter trades off correct classification of training examples against maximization of the decision function's margin. For larger values of C, a smaller margin will be accepted if the decision function is better at classifying all training points correctly. A lower C will encourage a larger margin, therefore a simpler decision function, at the cost of training accuracy. In other words, C behaves as a regularization parameter in the SVM. The behavior of the model is very sensitive to the gamma parameter. If gamma is too large the radius of the area of influence of the support vectors only includes the support vector itself and no amount of regularization with C will be able to prevent overfitting. In our model, we have chosen the kernel to be "RBF" with the C = [0.1,1,10] and gamma =[1,0.1, 0.01]. The accuracy obtained after testing the model with the above mentioned hyperparameters is 96%

D. Naive Bayes Model

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. It is easy and fast to predict class of test data set. It also performs well in multi class prediction. When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data. It performs well in case of categorical input variables compared to numerical variables. For numerical variables, normal distribution is assumed.

Also, it has some cons such as if categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as "Zero Frequency". To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation. Another limitation of Naïve Bayes is the assumption of independent predictors. In real life, it is almost

impossible that we get a set of predictors which are completely independent. We used cv=5, Multinomial as the estimator and the parameter grid consisted of 'alpha' and 'fit_prior'. The accuracy obtained after testing the model with the above mentioned hyperparameters is 92.98% with a Recall of 94.77

VII. Conclusion

There is very clear pattern that if SLAs mostly get missed for critical and high priority incidents whereas SLAs are met for Moderate and Low priority incidents. We could also forecast that high impact with critical priority incidents would be solved before any other incident and that is a positive outcome.

VIII. Reference:

- https://www.aismartz.com/blog/why-eda-is-crucial-for-any-data-science-project/
- https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/
- https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc
- https://www.lucidchart.com/blog/incident-management-process
- https://www.datacamp.com/community/tutorials/decision-tree-classification-python

IX. Heroku App Link:

https://obscure-atoll-74687.herokuapp.com/