Venu Malireddy

MACHINE LEARNING ENGINEER NANODEGREE
SERVICE REQUEST ANALYSIS

June 1, 2018

PROPOSAL

1. Domain Background

The goal of every IT Service Management framework is to ensure that the right processes, people and technology are in place so that the organization can meet its business goals.
ITSM caters to

- **Change Management** - standard methods and procedures for effective managing of all changes
- **Release Management** - testing, verification, and release of changes to the IT environment
- **Incident Management** - the day-to-day process that restores normal acceptable service with a minimal impact on business
- **Problem Management** - the diagnosis of the root causes of incidents in an effort to proactively eliminate and manage them

Because onset of digital workplace, cloud, IOT devices, smart machines, bots can also submit tickets online. Smart cars on the roads can send signal data of future failures and issues and also can notify ensuing maintenance schedule which can be treated as service request, there would be a need to handle numerous requests.
So, help desk should be able to tackle, volume, velocity and variety of requests flooding in to helpdesk. In order to Create better customer experiences at lower cost, help desk needs help.

ITSM solution providers like ServiceNow, a SAS company is catering to 80% of global 2000 companies. The growth of ITSM and Gartner statements below has motivated me to think of a solution that could help any Service request processing applications.

2. Problem Statement

IT Incident Management is comprised of a set of processes and practices designed to return a service(s) to normal functioning as quickly as possible with as little negative impact as possible.

Addressing the issue tickets to appropriate person or unit in the support team has critical importance in order to provide improved end user satisfaction while ensuring better allotment of support recourses. The assignment of help ticket to appropriate group is still manually performed. Especially at large organizations, the manual assignment is not applicable sufficiently. It is time consuming and requires human efforts. There may be mistakes due to human errors. Wherever you observe, workloads are increasing. And humans are often the bottleneck in clearing them.

Gartner says

Whenever a service request is raised, the request needs to be categorized, assigned and processed. 43% of IT service desk respondents had more than 100 different assignment groups to choose from and nearly a quarter of IT respondents faced a choice from more than 300 groups

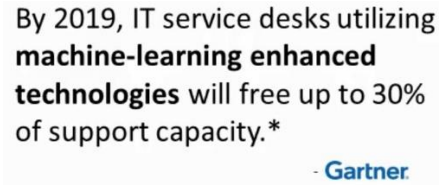
Improving service response times though automated processing allows

- customer service to scale with increased digital interactions,
- Tackling an overwhelming volume of messages from customer service
- increasing customer satisfaction and
- Lowering the overall cost of providing customer service

The main objectives are

- Reduce violation of service agreement levels (SLA) by assigning tickets to Subject Matter Experts who have handled similar tickets in the past
- Reducing problem determination effort by recommending relevant solutions from similar previously solved incidents
- Reduce occurrence of prevalent failure types by classifying incident failure types into a prior known failure class.
- To prioritize root cause analysis for large-volume types

Gartner predicts



Main challenge

- Main challenge that needs to be resolved is to create a model for automatic support tickets classification
- The ticket should be assigned to correct support teams / personnel.

Service tickets of around 60K data are available for public usage (https://www.ibm.com/communities/analytics/watson-analytics-blog/it-help-desk/). This is extracted to CSV / excel file and some sensitive information if any will be striped and enriched further to make it suitable for further analysis This is classified and assigned to support person with details of KB (knowledge base) article that is used to resolve the issue. The service tickets can be classified into the following categories like type, Business service, Category, Assignment group, Sub category, impact, urgency, resolution code etc..

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. Given a class variable $y$and a dependent feature vector $x_1$through x_n, Bayes' theorem states the following relationship:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

I am planning to use Naïve Bayes algorithm for multinomially distributed data which is one the best algorithm in text classification. I will try to optimize for best performance using best hyperparameters utilizing GridSearchCV for improved precision and recall.

3. Datasets and Inputs

- Data is extracted from IBM public community database to collect historical data (six months old) to fetch around 60 k classified support tickets with original messages from users and already assigned labels and KB articles.
  Helpdesk tickets: https://www.ibm.com/communities/analytics/watson-analytics-blog/it-help-desk/
- These labels are categorical features like Type, IT owner, Business service, Category, Assignment group, impact, urgency, resolution code etc.
- These defined labels and assigned KB articles serves to train the model and tune it with training data.
- 20% of unseen test data is used to test the model and to evaluate performance of the model.
- I can also use API to extract data to CSV

Process Data

- Clean and prepare text data and features to make it valuable for machine learning scenarios
- To strip the data from any sensitive information
- Look for proper sampling and tackle for unbalanced data

Training data Consists of historical tickets that are used to generate training data for creating the machine learning model. This data is stored temporarily in a dedicated repository. Prediction data Consists of new tickets, for example IT tickets, and is used for creating machine learning proposals.
The sample data set of extracted and cleaned data set looks like:

ticket	Requestor	Seniority	IT Owner	Category	Ticket Type	Severity	Priority	Days Open
1	1929	1 - Junior	50	Systems	Issue	2 - Normal	0 - Unassigned	3
2	1587	2 - Regular	15	Software	Request	1 - Minor	1 - Low	5
3	925	2 - Regular	15	Access/Login	Request	2 - Normal	0 - Unassigned	0
4	413	4 - Management	22	Systems	Request	2 - Normal	0 - Unassigned	20
5	318	1 - Junior	22	Access/Login	Request	2 - Normal	1 - Low	1
6	858	4 - Management	38	Access/Login	Request	2 - Normal	3 - High	0
7	1978	3 - Senior	10	Systems	Request	2 - Normal	3 - High	9
8	1209	4 - Management	1	Software	Request	2 - Normal	0 - Unassigned	15

4. Solution Statement

Solution recommendation is based on historical service tickets that have been classified into different categories like ticket type, ticket category, priority level, impact, department name etc..

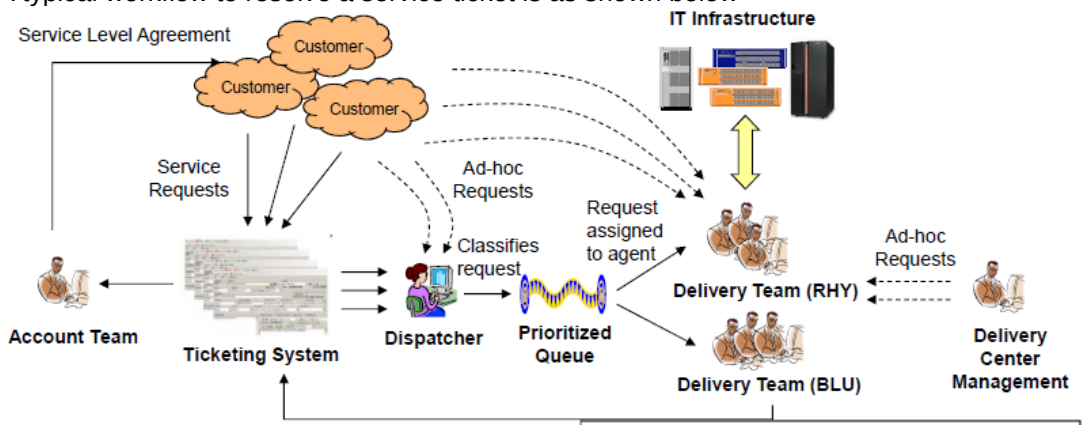
Ticket categorization:

Once a service ticket arrives, the machine learning algorithm automatically categorizes it into a certain category with a certain confidence level. Also, the support team assigned and the level of urgency is determined based on the input of the customer request. The system decides where the tickets need to be routed.

Thus, a list of tickets is assigned to different functional teams and the customer service agents can immediately start with the most urgent ones.

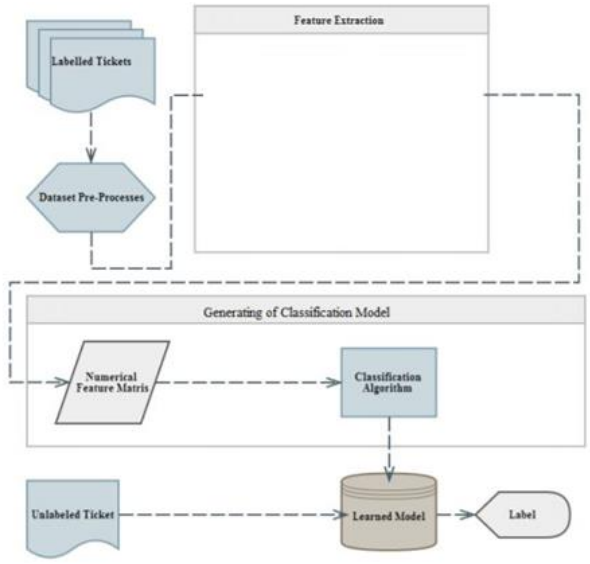
The machine learning algorithms can explore unobservable and unexplainable patterns and connections based on huge volume of data feeds.

A typical workflow to resolve a service ticket is as shown below



Source: Y. Diao et al., WSC 2011 and CNSM 2011

Training the model for Feature extraction and Ticket Categorization:



Solution recommendation

Solution recommendation is one of the functionalities of Service Ticket Intelligence
The training data contains labels as well as knowledge base articles numbers (KB Article).
The KB article is the source or knowledge where the steps for issue resolution are provided.

After a ticket arrives, the machine learning algorithm analyzes the unstructured text and pulls out most similar tickets and closest solution articles.
Our training model will learn the suitable KB article for specific problem category. The model provides solution recommendation for customer service agent.
Customer service agents go through them and pick up the most relevant one. Then they respond to the customer without searching through knowledge databases manually.

Get the labeled data:

Split data to train/test sets with 80:20 ratio

Create Model:

Dataset consisting of previously categorized tickets are used to train classification algorithms. Bag of words approach is utilized to extract features vectors.

There are various algorithms which can be used for text classification. We will start with the simplest one ‘Naive Bayes (NB)’ model. Naïve Bayes is a statistical classification algorithm based on Bayes theorem. It provides quite well performance when the training data consists of low amount of data. Also, the classifier relates with features rather than instances.

Evaluate the model

The **GridSearchCV** process will then construct and evaluate one model for each combination of parameters. Cross validation is used to evaluate each individual model and the default of 3-fold cross validation is used

Activate the model. After the model is activated, you can send unstructured questions to receive recommended answers.

5. Benchmark Model

Each ticket for which the model's prediction matches the actual result is considered a 'success'.
Any mismatch (in other words, any prediction that was subsequently changed by the desk agent) is considered an 'error'.
These errors are then used in the scheduled retraining of the model, which takes place at customer-defined intervals. Over time, the model can be refined by the supervisor to increase both the accuracy and the coverage (that is, the types of scenarios it can accurately predict).

Bench Mark in Domain:

Free form text is separated by line units, then conditional random fields (CRFs) is used to assign label to each line in order to indicate the information type of the unit. The proposed method demonstrated 82% accuracy

Textual description of web services which might be in the form of Web Service Description Language (WSDL) documents are used to classify and 83% accuracy is achieved (Bruno et al, 2005).

6. Evaluation Metrics

we modeled a scenario for a typical front-line customer support function.

Improved Customer Service:

Employee productivity will be enhanced since service representative is relieved of routine time consuming tasks.

Increased productivity:

Less time wasted on incorrect ticket routing, Improved customer experience

SLA Compliance :

A faster, more accurate process Increased SLA compliance

Strategic goal:

Configure a Service Level agreement (SLA) for top priority (priority 1) incidents for each pilot service

Evaluate with Confusion Matrix

Score and evaluate model on test data using model without hyperparameter tuning

With out grid search

- Evaluate model using Confusion Matrix
- Find out Mean

With grid search

- Evaluate model using Confusion Matrix
- Find out Mean

Calculate

- Precision
- Recall
- F1 score

- **Recall = # of correct positive predictions / # of positive examples**
- **Precision = # of correct positive predictions / # of positive predictions**

7. Project Design

Get data

Get an access token

The token is used to pass the authentication check of the backend service. After receiving the access token, you can upload training data which is used for training the machine learning model.

You need to specify the input columns and output columns.

- Collect around 50k classified support tickets

Process Data

- Clean and prepare text data and features to make it valuable for machine learning scenarios
  - Check Email addresses, phone numbers and urls, image references
  - Parse data Parse descriptions based on bag of words

Split the data into Training and test data.

Feature selection

Extract features from text

- Ticket_Type
- Bsiness_service
- Category
- Impact
- Urgency
- Sub category

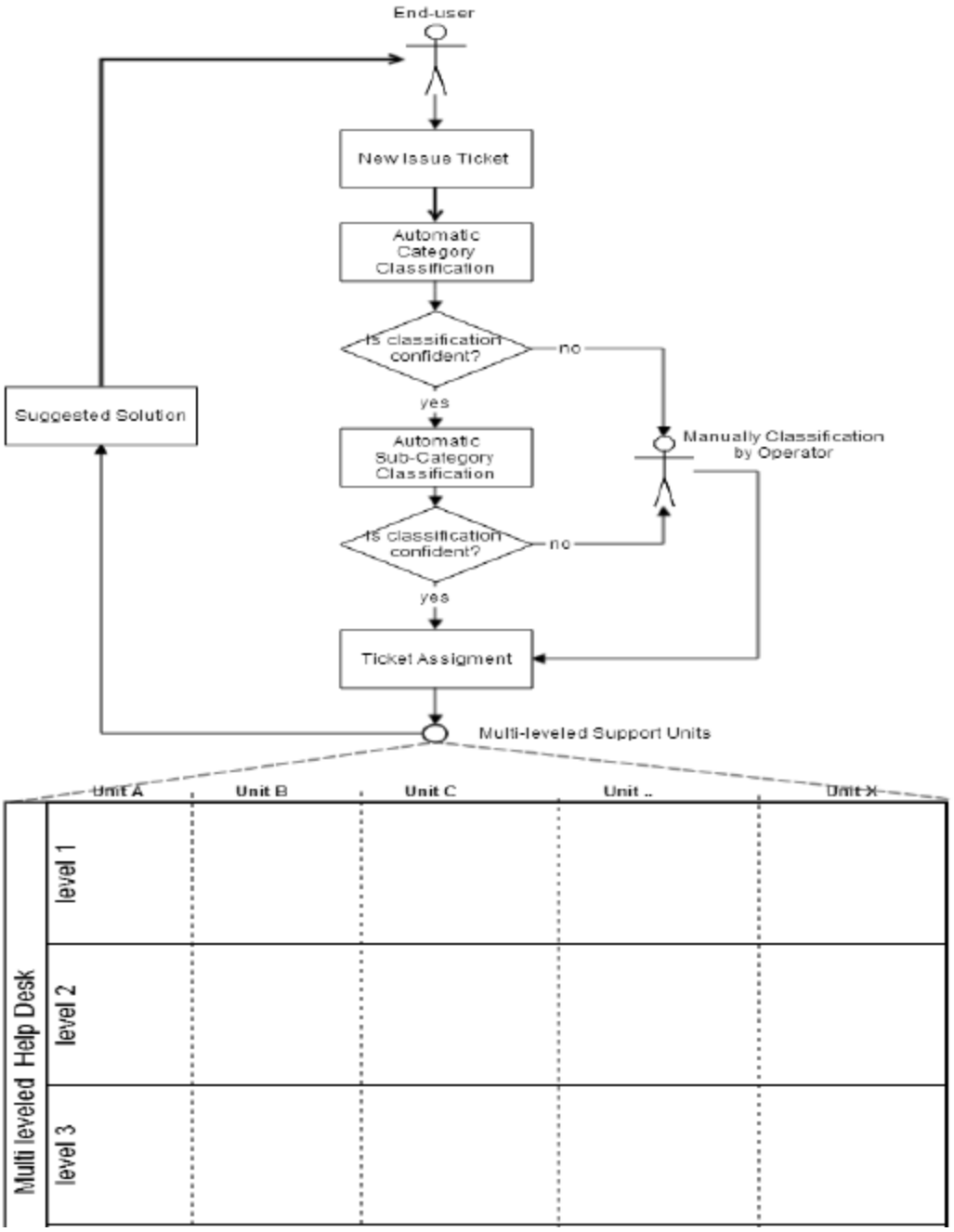
Train Model:

Use Naïve Bayes method

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features

Test Model

- The model is tested against a different data set and a report is generated showing the accuracy of predictions compared to real data



Tune the model

The model can be tuned using different data, confidence thresholds or other solutions for comparison – once deployed for use, service desk agents can accept or over-ride the predictions

Use GridSearchCV

Fine tune parameters

Plot confusion matrix

After the training data has been passed to the server, you can trigger the training process. The machine learning algorithm will start training the model.

- Evaluate model using Confusion Matrix
- Find out Mean

Calculate

- Precision
- Recall
- F1 score

Improve Model

Scheduled retraining at predefined intervals and incorporating manually changed predictions allows the model to improve accuracy and coverage quickly over time

Name	Title	Date

Approved By Date Approved By Date

References:

- Proceeding of the International Conference on Artificial Intelligence and Computer Science (AICS 2014), 15 - 16 September 2014, Bandung, INDONESIA. **(e-ISBN978-967-11768-8-7).** Organized by http://WorldConferences.net
- McCallum and K. Nigam (1998). A comparison of event models for Naive Bayes text classification. Proc. AAAI/ICML-98 Workshop on Learning for Text Categorization, pp. 41-48.
- https://www.aaai.org/Papers/Workshops/1998/WS-98-05/WS98-05-007.pdf
- https://www.ibm.com/communities/analytics/watson-analytics-blog/it-help-desk/
- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.46.1529&rep=rep1&type=pdf
- http://scikit-learn.org/stable/modules/grid_search.html#grid-search