Venu Malireddy

MACHINE LEARNING ENGINEER NANODEGREE
SERVICE REQUEST ANALYSIS

June 20, 2018

DEFINITION

1. Project Overview

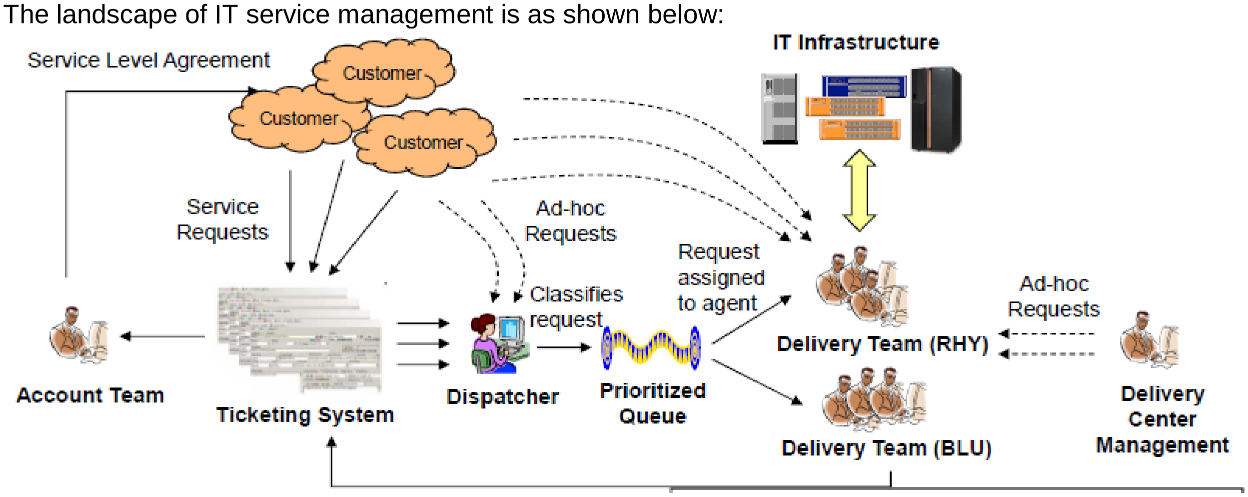
The goal of every IT Service Management framework is to ensure that the right processes, people and technology are in place so that the organization can meet its business goals.
ITSM caters to

- **Incident Management** - the day-to-day process that restores to 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 and bots, these devices 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.

The landscape of IT service management is as shown below:



[1] IT Service Management

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 of assigning 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 desk 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:

> "By 2018, things as customers will drive 6 billion support requests, requiring businesses to develop new customer support capabilities."
>
> gartner.com/SmarterWithGartner — Gartner

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:

> By 2019, IT service desks utilizing **machine-learning enhanced technologies** will free up to 30% of support capacity.*
>
> - Gartner

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.

3. Metrics

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 and utilizing GridSearchCV for improved precision and recall.

For a document *d* and a class *c* : $P(c \mid d) = P(d \mid c)\, P(c) / P(d)$

Precision : % of selected items that are correct
Recall : % of correct items that are selected

	Correct	**Not Correct**
Selected	TP	FP
Not Selected	FN	TN

Recall	Fraction of docs in class *I* classified correctly	TP/(TP+FP)
Precision	Fraction of docs assigned class *I* that are actually about class *i*	TP/(TP+FN)
Accuracy	(1 – error rate) Fraction of docs classified correctly	2*Recall*Precision/ (Recall +Precision)

ANALYSIS

4. Data Exploration

Each week, financial protection bureau receives complaints / requests regarding financial products and services to companies for response and resolution of the issue. Based on customer complaint narrative or description, the request is categorized based on products, sub-products, issues, and sub-issues are grouped.
Data is extracted from a public community database(data.gov) to collect historical data (six months old) to fetch around 60 k classified support tickets with original messages from users with already assigned labels.

Helpdesk tickets: https://catalog.data.gov/dataset/consumer-complaint-database

- These labels are categorical features like complaint type, Product, sub-product, company etc.
- These defined labels serve 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.

Following are the fields in the database

Date received
Product
Sub-product
Issue
Sub-issue
Consumer complaint narrative
Company public response
Company
State
ZIP code
Tags
Consumer consent provided?
Submitted via
Date sent to company
Company response to consumer
Timely response?
Consumer disputed?
Complaint ID

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 unbalanced data

Training data Consists of historical requests that are used to generate training data for creating the machine learning model..
These requests are categorized into one of the following product types

- Checking or savings account
- Credit card or prepaid card
- Credit reporting, credit repair services, or other personal consumer reports
- Debt collection
- Money transfer, virtual currency, or money service
- Mortgage
- Payday loan, title loan, or personal loan
- Student loan
- Vehicle loan or lease

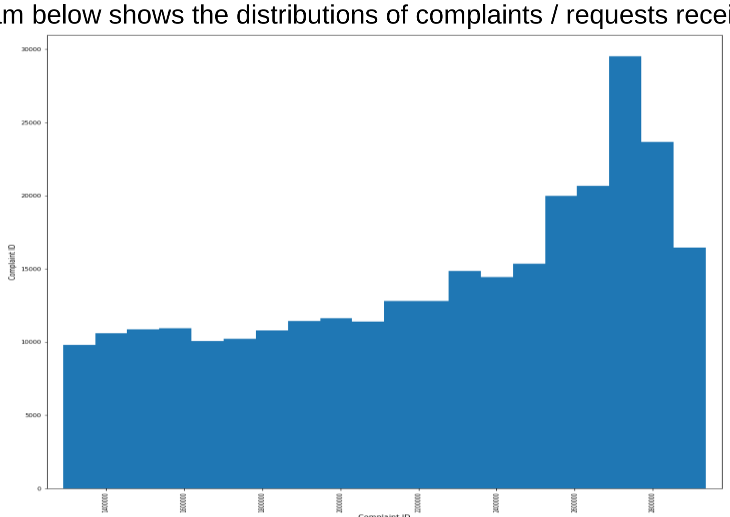
5. Exploratory Visualization

The characteristics of the service requests and the number of records in database are:

data frame details

<class 'pandas.core.frame.DataFrame'>	
Int64Index: 288291 entries, 1 to 1048574	
Data columns (total 18 columns):	
Date received	288291 non-null object
Product	288291 non-null object
Sub-product	236190 non-null object
Issue	288291 non-null object
Sub-issue	188048 non-null object
Consumer complaint narrative	288291 non-null object
Company public response	139357 non-null object
Company	288291 non-null object
State	287260 non-null object
ZIP code	285867 non-null object
Tags	49400 non-null object
Consumer consent provided?	288291 non-null object
Submitted via	288291 non-null object
Date sent to company	288291 non-null object
Company response to consumer	288290 non-null object
Timely response?	288291 non-null object
Consumer disputed?	163567 non-null object
Complaint ID	288291 non-null int64
dtypes: int64(1), object(17)	
memory usage: 41.8+ MB	

The histogram below shows the distributions of complaints / requests received from users



Complaint

Ticket categorization:

Once a service ticket arrives, the machine learning algorithm automatically categorizes it into a certain ‘product’ with a certain confidence level. The system decides where the tickets need to be routed. Thus, a list of tickets is assigned to different product 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.
For every product type, a numerical number is assigned as product-code to enable machine learning classifier to predict better.

```
pd.unique((df_cat[['Product_code','Product']]).values.ravel())
```

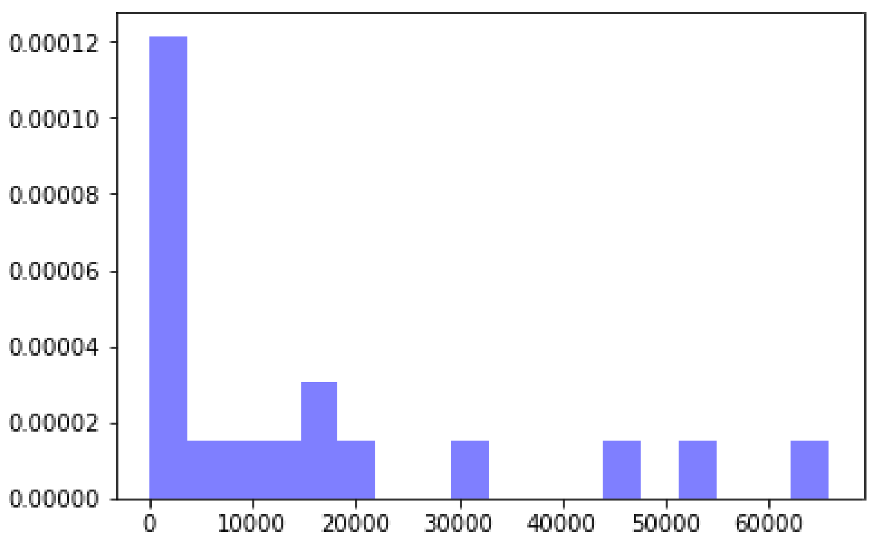
Product Code and Product
0: 'Bank account or service',
1: 'Checking or savings account',
2: 'Consumer Loan',
3: 'Credit card',
4: 'Credit card or prepaid card',
5: 'Credit reporting',
6: 'Credit reporting, credit repair services, or other personal consumer reports',
7: 'Debt collection',
8: 'Money transfer, virtual currency, or money service',
9: 'Money transfers',
10: 'Mortgage',
11: 'Other financial service',
12: 'Payday loan',
13: 'Payday loan, title loan, or personal loan',
14: 'Prepaid card',
15: 'Student loan',
16: 'Vehicle loan or lease',
17: 'Virtual currency'

The main categories required for machine learning analysis are shown below. The user request or complaint along with assigned product label and product code are shown below for visual analysis

	Product	complaint	Product _code
1	Credit reporting	I have outdated information on my credit repor...	5
2	Consumer Loan	I purchased a new car on XXXX XXXX. The car de...	2
7	Credit reporting	An account on my credit report has a mistaken ...	5

	Product	complaint	Product _code
12	Debt collection	This company refuses to provide me verificatio...	7
16	Debt collection	This complaint is in regards to Square Two Fin...	7
25	Mortgage	Started the refinance of home mortgage process...	10
26	Mortgage	In XXXX, I and my ex-husband applied for a ref...	10
28	Credit reporting	I have disputed several accounts on my credit ...	5
29	Mortgage	Mortgage was transferred to Nationstar as of X...	10
36	Credit card	Was a happy XXXX card member for years, in lat...	3
43	Credit card	Without provocation, I received notice that my...	3
49	Debt collection	I am writing to request your assistance in loo...	7
61	Credit reporting	I am disputing the inaccurate information the ...	5
64	Credit reporting	Checked my credit report after filing complain...	5
69	Mortgage	Need to move into a XXXX facility. Can no long...	10

The customer request is provided as a narrative or description. This will be the source for machine learning classifier to analyze. The distribution of complaints / requests are shown below in histogram



df_prod = df['Product'].value_counts()

df_prod

Following are the 18 types of labels that machine learning classifier needs to predict. The assigned labels and the count of label in training set is shown below .

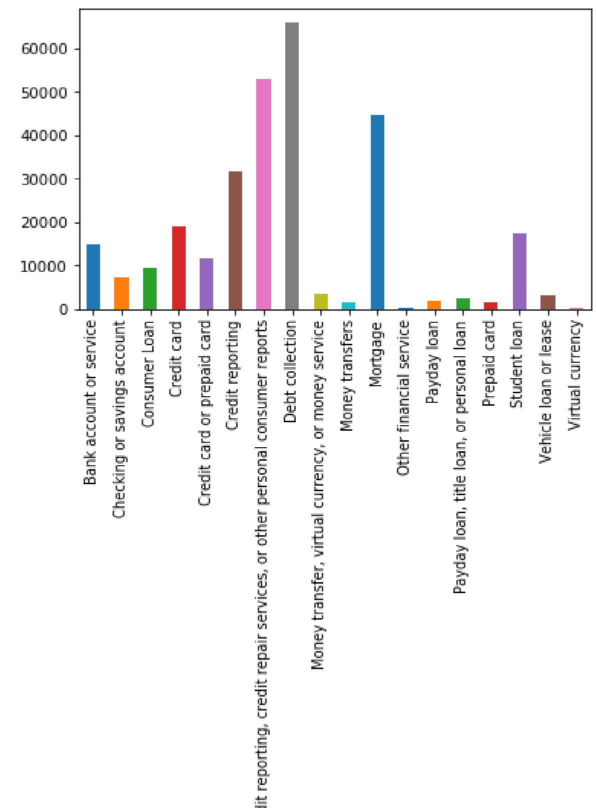
Debt collection	65848
Credit reporting, credit repair services, or other personal consumer reports	53109
Mortgage	44797
Credit reporting	31572
Credit card	18785
Student loan	17250
Bank account or service	14842
Credit card or prepaid card	11769
Consumer Loan	9444
Checking or savings account	7086
Money transfer, virtual currency, or money service	3281
Vehicle loan or lease	3114
Payday loan, title loan, or personal loan	2401
Payday loan	1744
Money transfers	1494
Prepaid card	1447
Other financial service	292
Virtual currency	16

Each product category is assigned with a numerical label so that it helps ML classifier to analyze and assign suitable category.

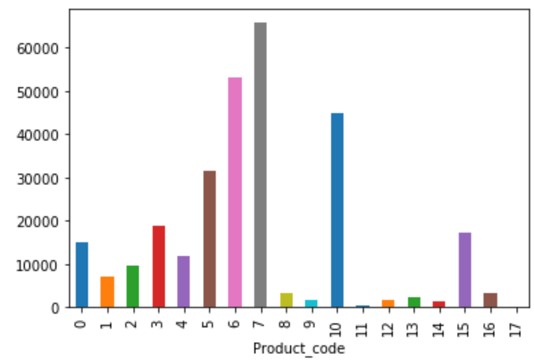
```
# Categories in data frame
df_cat.columns = ['Product', 'complaint', 'Product_code']
# Count the columns of the Product
df_cat.groupby('Product').complaint.count().plot.bar(ylim=0)
plt.show()
# Plot the columns of the Product code
df_cat.groupby('Product_code').complaint.count().plot.bar(ylim=0)
```

Name: Product, dtype: int64

Product



Product_code



6. Algorithms and Techniques

Consider a dataset that consists of 10000 requests or complaint descriptions.
These descriptions can sometimes be quite long. We need to count all the words in them and create a feature vector (vector with number of occurrences for all unique terms/words in all documents).

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 and does not contain all possibilities.

A method most frequently used is term frequency- inverse document frequency (tf-idf). The tf-idf value increases comparatively to the number of times a term appears in the document but is offset by the frequency of the term in the corpus. It is the most common used method in literature because of the performance with SVM learning algorithm. The distinctive terms for classification which have got bigger idf than a certain threshold for all documents are considered as stop words. Lots of these terms are conjunctions used as an independent word to the topic and misspelled words. The stop words have been removed from the feature vectors. In this way, feature vector size is reduced as much as possible and noise of the feature vector was eliminated.

4.3 Feature Extraction

Term Frequency–Inverse Document Frequency
TF: Just counting the number of words in each document has an issue: it will give more weightage to longer documents than shorter documents. To avoid this, we can use frequency (**TF - Term Frequencies**) i.e. #count(word) / #Total words, in each document.

TF-IDF: Finally, we can even reduce the weightage of more common words like (the, is, an etc.) which occurs in all document. This is called as **TF-IDF i.e Term Frequency times inverse document frequency.**

Term Frequency–Inverse Document Frequency (TF–IDF) is a numerical statistic that is used to determine the importance of a word in a document. This statistic can be used as a feature extraction method by simply computing it for each feature (word frequency) of each sample (document).

The value of TF–IDF statistic is computed as the product of two other statistics—term frequency and inverse document frequency. Inverse document frequency is a statistic that determines relative rarity of a term across many documents.
If, for example, a term (t) is always present in a set of documents (D), it conveys very little meaning and can be almost disregarded

7. 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 72% 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).

METHODOLOGY

8. Data Preprocessing

Get data
The helpdesk request data is available in public domain from https://catalog.data.gov/dataset/consumer-complaint-database. The data can be downloaded in CSV format and can be used as training data to train machine learning model.

- Collect around 50k classified support tickets

Process Data

- Clean and prepare text data and features to make it valuable for machine learning scenarios
- Data is already anonymized by removing all references to personal information
    - Analyze the data pattern. Check for null values.
    - All unnecessary dimensions are reduced.
    - Only required dimensions are chosen for machine learning classifier
    - Numerical codes are provided Product as product_code

	Product	complaint	Product_code
1	Credit reporting	I have outdated information on my credit repor...	5
2	Consumer Loan	I purchased a new car on XXXX XXXX. The car de...	2
7	Credit reporting	An account on my credit report has a mistaken ...	5
12	Debt collection	This company refuses to provide me verificatio...	7
16	Debt collection	This complaint is in regards to Square Two Fin...	7
25	Mortgage	Started the refinance of home mortgage process...	10
26	Mortgage	In XXXX, I and my ex-husband applied for a ref...	10
28	Credit reporting	I have disputed several accounts on my credit ...	5
29	Mortgage	Mortgage was transferred to Nationstar as of X...	10
36	Credit card	Was a happy XXXX card member for years, in lat...	3
43	Credit card	Without provocation, I received notice that my...	3
49	Debt collection	I am writing to request your assistance in loo...	7
61	Credit reporting	I am disputing the inaccurate information the ...	5
64	Credit reporting	Checked my credit report after filing complain...	5
69	Mortgage	Need to move into a XXXX facility. Can no long...	10

Numerical values are assigned to each product for machine learning classifier to analyze and assign to required category.

Product_code	Product
0	Bank account or service
1	Checking or savings account
2	Consumer Loan
3	Credit card
4	Credit card or prepaid card
5	Credit reporting
6	Credit reporting
7	Debt collection
8	Money transfer
9	Money transfers
10	Mortgage
11	Other financial service
12	Payday loan
13	Payday loan
14	Prepaid card
15	Student loan
16	Vehicle loan or lease
17	Virtual currency

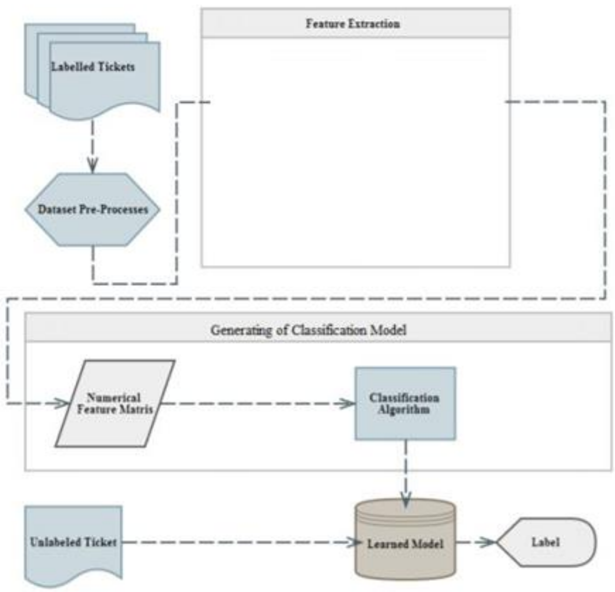
Split the model:

- All complaint / request data with request descriptions are considered as service requests
- The predefined labels are separated as train_labels
- This data is used to split into training and testing data set with 80:20 ratio

Feature selection

Extract features from text

The customer complaint description is parsed and features are extracted. Machine learning classifier is trained based on pre-assigned labels of product type and product code



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

Training the model for Feature extraction and Ticket Categorization:

Calculate TF-DF

Once we have a list of matching documents, they need to be ranked by relevance. Not all documents will contain all the terms, and some terms are more important than others. The relevance score of the whole document depends (in part) on the *weight* of each query term that appears in that document.

`tf(t in d) = √frequency`

Inverse document frequency

How often does the term appear in all documents in the collection? The more often, the *lower* the weight. Common terms like and or the contribute little to relevance, as they appear in most documents,

`idf(t) = 1 + log (numDocs / (docFreq + 1))`

Train Model:
Naive Bayes methods are a set of supervised learning algorithms based on 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:

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

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
- ➢ Some of the test conducted are shown below.

```
# Test the prediction of classifier
predict = clf.predict(vectorizer.transform(test_data))
predict1 = clf.predict(vectorizer.transform(['Why is this account in my credit report? It was paid in full in 2013. XXXX is ruini
print(" Product Code predicted =", predict1)
```

```
Product Code predicted = [6]
```

 # Test the prediction of classifier

predict = clf.predict(vectorizer.transform(test_data))

predict1 = clf.predict(vectorizer.transform(['Why is this account in my credit report? It was paid in full in 2013. XXXX is ruining my credit report!']))

print(" Product Code predicted =", predict1)

```
Product Code predicted = [6]
```

predict1 = clf.predict(vectorizer.transform(['I have paid all copays and just rec d a bill from a collection agency. I have never rec d anything in the mail before on this. I called the Dr offfice and they say it was because visits that are being billed didnt have a referral but I always had a referral and if they told me that I would have gotten XXXX. I have attached proof that my primary care physician did send referrals. I feel as though this is some sort of way for them to bill people almost 5 years later knowing they will pay or it will kill their credit. I am in the XXXX and will be forced to pay something that is not accurate almost 5 years after the services']))

print(" Product Code predicted =", predict1)

```
Product Code predicted = [7]
```

Tune the model

After selecting my initial model and obtaining baseline results, I tried to improve my model's performance and targeted metrics by tuning my model's features and parameters
Scikit learn's Pipeline and GridSearchCV classes allow me to accomplish this task and iterate quickly. With my pipeline setup, I can use GridSearchCV. GridSearchCV will take my pipeline and a range of numbers I specify for each parameter and then find the combination set that provides the best estimator. Once, I have created my GridSearch, I can output the details of the best model by using GridSearch's function and make predictions on the testing data

The model can be tuned using different data, confidence thresholds or other solutions for comparison

```
Pipeline(steps=[('vect', CountVectorizer(analyzer='word', binary=False, decode
                    _error='strict',dtype=<class 'numpy.int64'>, encoding='utf-8'
                    , input='content' lowercase=True, max_df=1.0,
                    max_features=None, min_df=1,ngram_range=(1, 1),
                    preprocessor=None, stop_words=None,
                     strip...inear_tf=False, use_idf=True)), ('clf',
                     MultinomialNB(alpha=1.0, class_prior=None,
                     fit_prior=True))])
```

Use GridSearchCV

Fine tune parameters
Plot confusion matrix

```
from sklearn.model_selection import GridSearchCV
#  penalty parameter of either 0.01 or 0.001 for the linear SVM
param = {'vect__ngram_range': [(1, 1), (1, 2)],
                'tfidf__use_idf': (True, False),
                'clf__alpha': (1e-2, 1e-3)}
```

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

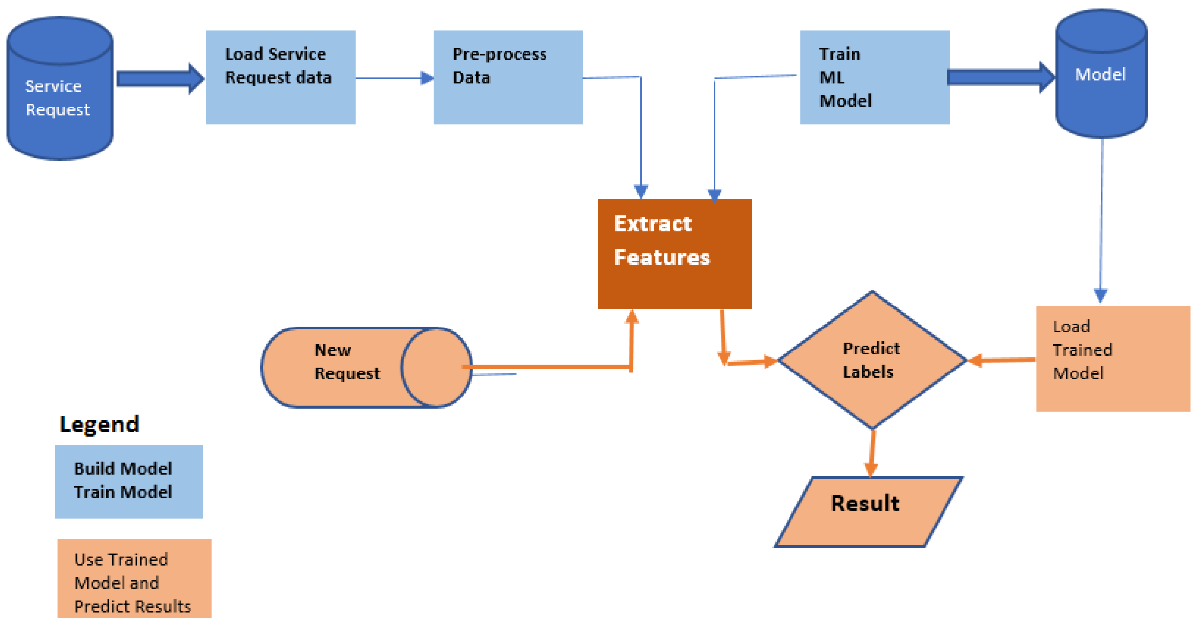
Scheduled retraining at predefined intervals and incorporating manually changed predictions allows the model to improve accuracy and coverage quickly over time

9. Implementation

The diagram below depicts the both training and testing process. Service requests will be downloaded to a CSV file. The data will be preprocessed and enriched so that it will be useful for machine learning purpose.

The customer narrative provided by user will be used to parse and extract the required features. The model will be trained using pre-defined labels. The model will be tested with unseen test data to measure the performance and accuracy.

The trained model can be used to predict product type for any new customer request.



Get the labeled data:

Historical data with predefined labels will be used to train the model, Data will be split into train/test sets with 80:20 ratio

Create Model:

Dataset consisting of previously categorized tickets are used to train classification algorithms. TF-IDF 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.

Fine tune and 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

Pipeline

Scikit-learn provides a pipeline utility to help automate machine learning workflows. Pipelines are very common in Machine Learning systems, since there is a lot of data to manipulate and many data transformations to apply. So we will utilize pipeline to train every classifier.

OneVsRest multi-label strategy

The Multi-label algorithm accepts a binary mask over multiple labels. The result for each prediction will be an array of 0s and 1s marking which class labels apply to each row input sample.

Naive Bayes

OneVsRest strategy can be used for multi-label learning, where a classifier is used to predict multiple labels for instance. Naive Bayes supports multi-class, but we are in a multi-label scenario, therefore, we wrap Naive Bayes in the OneVsRestClassifier.

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

RESULTS

10. Model Evaluation and Validation

In order to evaluate the used classification algorithms, several experiments have been conducted.

The model is built based on Multinomial Bayes classification method using DF-IDF. Then the model is again tested with pipe line. the model is evaluated using confusion matrix. The metrics of precision, accuracy and F1score are calculated. A heatmap is also plotted to check the behavior and performance. The model is again tried with SVM. The performance did not improve much. The accuracy is 77% approx.

Evaluate with Confusion Matrix

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

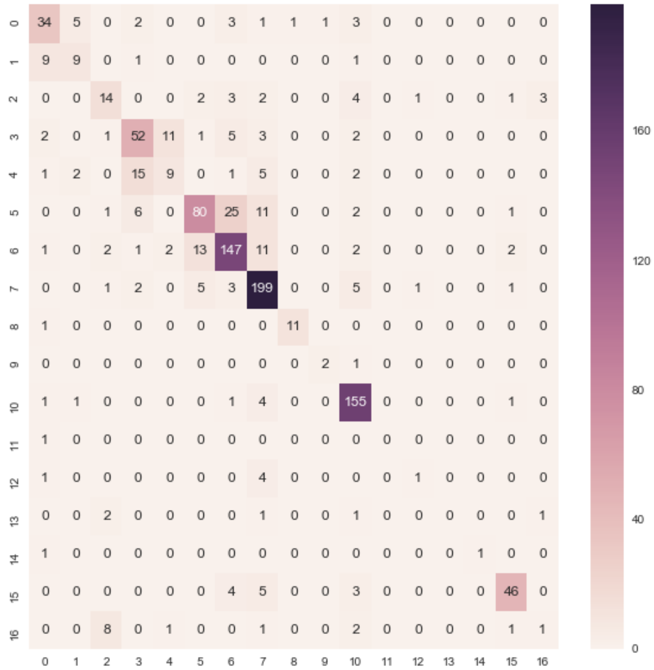
Without grid search

- Evaluate model using Confusion Matrix
- Find out Mean

With grid search

- Evaluate model using Confusion Matrix
- Find out Mean

The heatmap below shows the results of prediction on test set



Calculate

- Precision
- Recall
- F1 score

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

	precision	recall	f1-score	support
Credit reporting	0.65	0.68	0.67	50
Consumer Loan	0.53	0.45	0.49	20
Debt collection	0.48	0.47	0.47	30
Mortgage	0.66	0.68	0.67	77
Credit card	0.39	0.26	0.31	35
Other financial service	0.79	0.63	0.70	126
Bank account or service	0.77	0.81	0.79	181
Student loan	0.81	0.92	0.86	217
Money transfers	0.92	0.92	0.92	12
Payday loan	0.67	0.67	0.67	3
Prepaid card	0.85	0.95	0.90	163
Money transfer, virtual currency, or money service	0.00	0.00	0.00	1
Credit reporting, credit repair services, or other personal consumer reports	0.33	0.17	0.22	6
Checking or savings account	0.00	0.00	0.00	5
Vehicle loan or lease	1.00	0.50	0.67	2
Credit card or prepaid card	0.87	0.79	0.83	58
Virtual currency	0.20	0.07	0.11	14
avg / total	0.74	0.76	0.75	1000

```python
np.mean(predict_gsCV == test_labels[:2000])
```

0.761

Metric	Formula	Value
Precision	$\frac{TP}{(TP + FP)}$	0.74
Recall	$\frac{TP}{(TP + FN)}$	**0.76**
F1	$\frac{2 * Recall * Precision}{(Recall + Precision)}$	**0.75**

In confusion matrix,

- True Positive (TP) is the set of document that is correctly assigned to the given category,
- False Positive (FP) is the set of documents that incorrectly assigned to the category,
- False Negative (FN) is the set of documents that is incorrectly not assigned to the category and
- True Negative (TN) is the set of the set of documents correctly not assigned to the category

Then to improve performance of the model further, hyperparameters are tuned. some parameters such as use_idf in the Tfidf transformer are used for performance. Classifiers tend to have many parameters as well. MultinomialNB includes a smoothing parameter alpha and SGDClassifier has a penalty parameter alpha and configurable loss and penalty terms in the objective function

Then I tuned hyperparameter and used Gridsearch CV . this has substantially increased 77.5 %

To test the performance of model, some customer requests are selected, and results are predicted for product names

11. Justification

The model predicts the categories to extent of 76% accuracy. This will greatly enhance the performance of the teams trying to resolve helpdesk tickets. Initial bench mark accuracy for the model is 56%. Various tuning measures are applied to enhance accuracy of the classifier. The accuracy of the final model is increased to 76% which can be considered as reasonably good accuracy for classification model.
The following advantages are very much visible

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

CONCLUSION

Everyday financial protection bureau receives thousands of complaints / requests regarding various financial products and services. The department needs to analyze and assign each ticket based on customer complaint narrative or description to the required product type for resolution manually. A classification model based on supervised learning is proposed to assign the ticket to the required product type automatically.

Historical data set with predefined labels are considered to train classification model. Multinomial Naïve Bayes classifier-based model is used to classify the requests. Term-frequency-inverse document frequency (TF-IDF) methodology is used, where in words are given weight based on relevancy of the word.

In TF-IDF, it is the term weight which is represented in Vector space model. Thus, entire document is a feature vector. which points to a point in vector space such that there is an axis for every term in our bag.

I have used 2 different algorithms to evaluate the model like Naïve bates, SVM. Also used pipeline and hyperparameters to tune the model. With Multinomial Naïve Bayes, I could achieve 77.5% of accuracy.

The evaluation demonstrated the effectiveness of the proposed model. The model can be fine tuned still to improve performance and can be tried with convolution neural networks as well.

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
- [1]https://dl.acm.org/citation.cfm?id=3098113&dl=ACM&coll=DL (KDD'17, August 13–17, 2017, Halifax, NS, Canada)
- https://catalog.data.gov/dataset/consumer-complaint-database
- Source: Y. Diao et al., WSC 2011 and CNSM 2011
- http://docplayer.net/8898671-Machine-learning-predictive-analytics-for-it-services.html