Beginner Problem

Gautam Pai(Kaggler)

27 January 2020

Contents

1 Introduction

1.1 Problem Statement

There will be various issues logged by users through disparate forums. The problem is to tag the issue with the relevant category so that appropriate team can address the issue. There are 12 unique category tags, based on the tagging done on the past data, we need to tag for the unseen new issues logged by the users.

1.2 Sample Data

1.2.1 Training Dataset

Table 1.1: Training Sample Data

Serial_ID	type	Keywords	Prominent Keywords	title	body	description	Category
400078	SLN	turn, rotate, stuck, XPS 18, XPS18 won't rotate		Dell XPS 18 Screen May Not Rotate as Expected	Dell XPS 18 Screen May Not Rotate as Expected…	Explanation of the limitation of screen rotation on your Dell XPS 18.	Display & Audio Devices

As you can see in the table above there are 7 predictor variables and one highlighted is the dependent variable.

1.2.2 Validation Dataset

Table 1.2: Validation Sample Data

Serial_ID	type	Keywords	Prominent Keywords	title	body	description
500001	SLN	Inspiron N5050, Inspiron N4050, Inspiron N5110, Inspiron N311z, Vostro 1550, Vostro 1450, Vostro 3550, Vostro 3350, Vostro 131	inspiron, vostro	Computer Boots to a Black Screen After Windows 8 Upgrade	website and enter your Service Tag to get the latest version of the BIOS ..	Resolve situation where Dell computers boot to a black screen after Windows 8 upgrade has completed

The above table represents the validation (scoring) dataset where we need to tag the relevant categories based on the predictor variables

1.2.3 Dictionary Dataset

Table 1.3: DictionarySet Sample Data

RecordId	Title	Description	Article Type
833902	Are Metrosync replication session automatically failed over on peer SP if Replication Ports are failing?	According to the nas capabilities document a Nas server will only failover in the event of a SP failure or reboot. Disabling a switch port or a link going down on a port is not considered a "SP fault" or Array issue and would not cause a failover..	How To

The above table provides a dictionary set which gives the semantic relationship of the keywords.

Table 1.4: Category Values

Category
OS & Drivers
Drive & Storage
System Board & Related
Display & Audio Devices
Power Related
3rd Party & Applications
Input Devices
Others
Imaging Devices
Network & WiFi
Virus/Spyware
Diagnostics Related

2 Exploratory Data Analysis

2.1 Distribution of categories

Any predictive modeling requires that we look at the data before we start modeling. However, in data mining terms *looking at data* refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as **Exploratory Data Analysis**. In Figure2.1 we have plotted the distribution of the categorical variable *Category*.

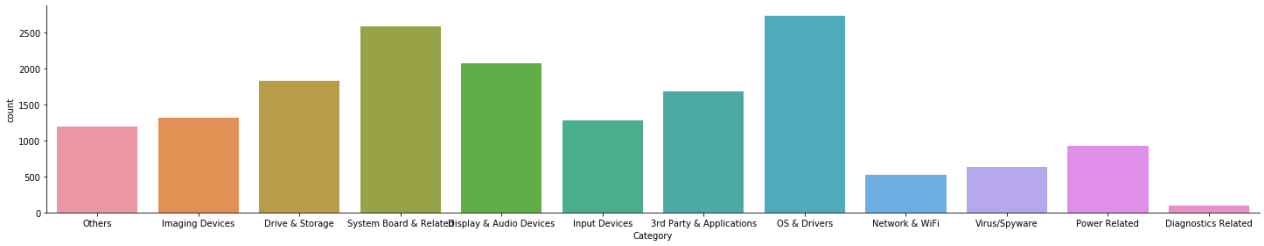


Figure 2.1: Catplot of the dependent variable

From the above plots it looks like the dependent variable distribution is not balanced and large number of observations are dominated for the category “OS & Drivers” & “System Board & Related”

2.2 Missing Value Analysis

In this stage we identify the missing values in each and every feature and gather the count of all the variables. Missing values can arise due to many reasons like business unable to capture the data due to privacy or error in capturing the data. Missing values are present as NA/NAN or empty values. The rule of thumb is if the missing values are more than 30%, it is advisable to drop that variable. If it is below to some percentage based on business scope, we can either drop those samples or else impute those values with either mean, median, knnimputation, random sampling imputation.

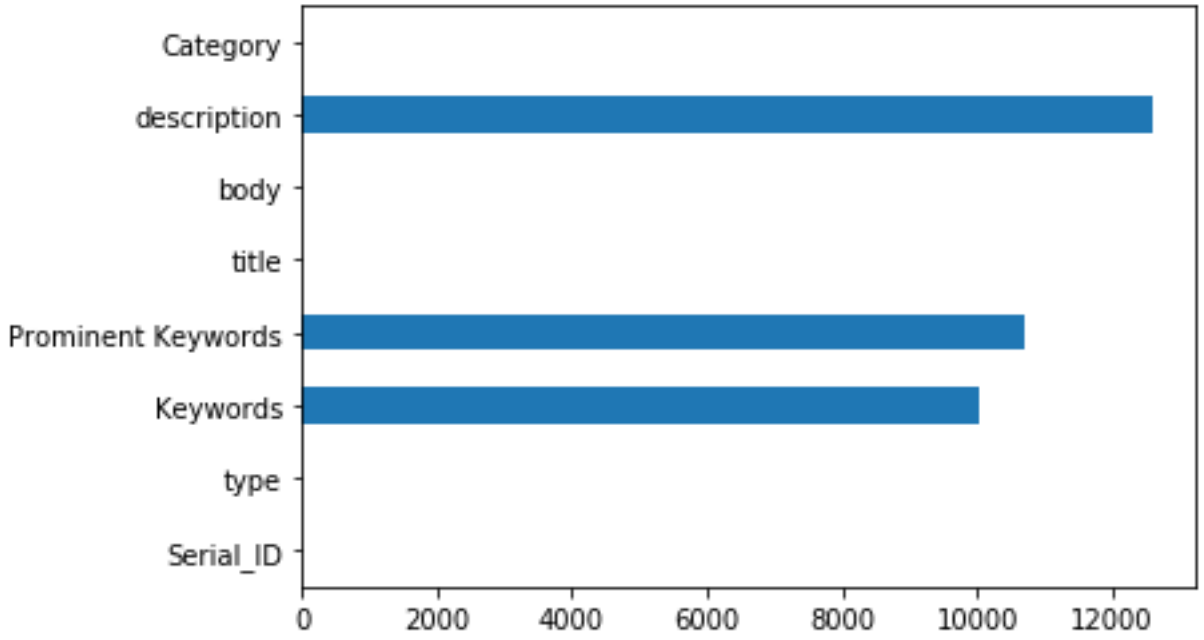
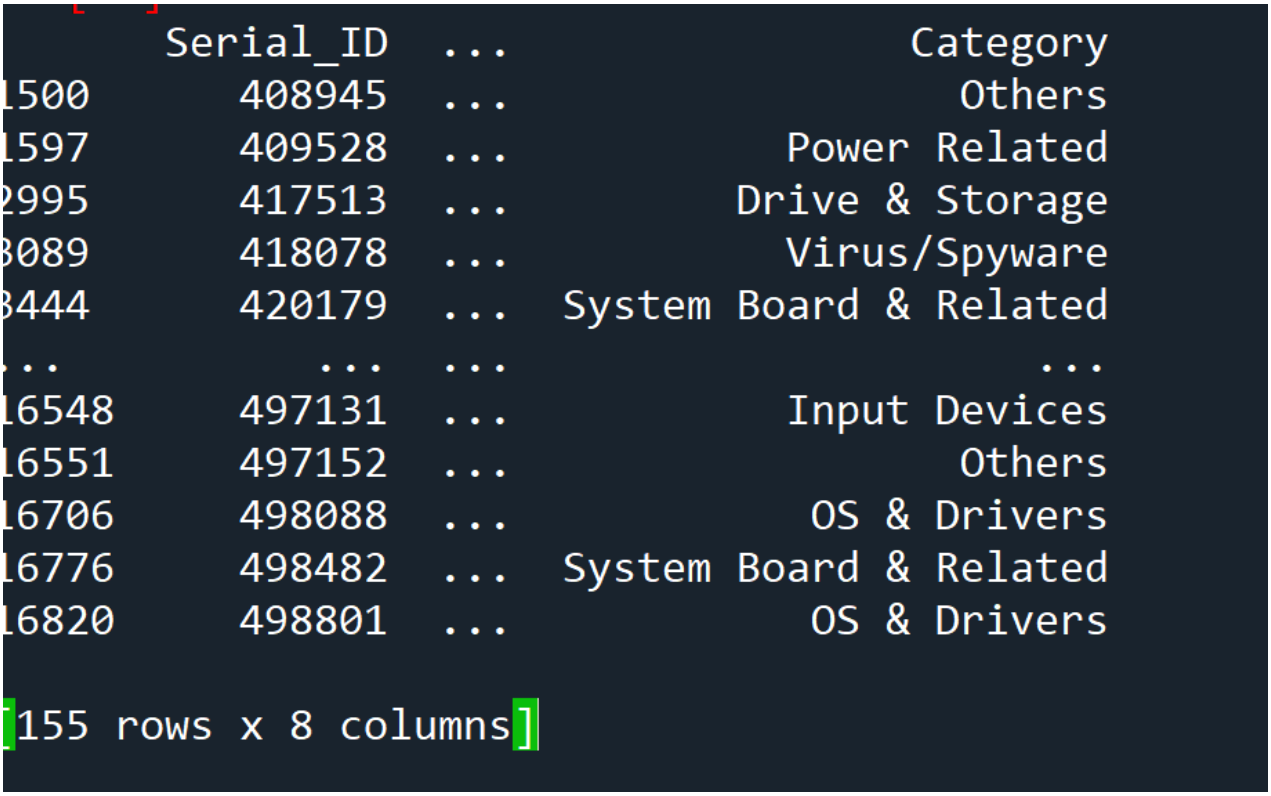


Fig 2.2: Missing Values for the variables in the training set

From the above plot it looks there are huge number of missing values in “description”, “Prominent Keywords” and “Keywords”

2.2 Finding out Dupes

Having duplicate records in the dataset gives improper accuracy in fitting the classifier and also it increases the complexity of the model. We tried to eliminate duplicate records by clubbing fields like 'title', 'description" & "title", "description", "body" and we found around 155 rows which are dupes on 3 column set as shown in figure below



	Serial_ID	...	Category
1500	408945	...	Others
1597	409528	...	Power Related
2995	417513	...	Drive & Storage
3089	418078	...	Virus/Spyware
3444	420179	...	System Board & Related
...	...	...	...
16548	497131	...	Input Devices
16551	497152	...	Others
16706	498088	...	OS & Drivers
16776	498482	...	System Board & Related
16820	498801	...	OS & Drivers

[155 rows x 8 columns]

3 Data Preprocessing

For any machine learning or data science projects, data preprocessing is a major step and typically constitute around 80% of the project work. The success of any project depends upon the quality of data preprocessing done.

For the current problem statement, the texts contained in all the columns need to be lowercased, tokenized, removal of stopwords, punctuation, numerals and then they have to be lemmatized i.e. transforming the text to its base form. Once all of the preprocessing steps are done, the token needs to be converted into numeric forms known as vectorization so that any statistical models are able to take as input and provide inferences. After preprocessing some of the rows in training dataset turned out to be completely missing (18records). Eliminated those records.

Once the preprocessing step completes, we persist the data into relevant csv files.

Sample data after preprocessing is shown in

Table 3.1 Sample data after preprocessing

title	description
metrosync replication session automatically failed peer replication port failing	according na capability document na server failover event failure reboot disabling switch port link going port considered fault array issue would cause failover since na server failover due network loss replication session would failed well

4 Modeling

The problem statement we are dealing with is a classification problem. We tried numerous algorithms starting from most basic of Logistic Regression, Multinomial NB, SVC, Ridge Classifier, Random Forests and AdaBoost Classifier. The following were tried and tested during modeling

1. Initially we used only title and description column vectors and found out that the accuracy was not going beyond 25%.
2. We used only body column vector and found to have a good accuracy but not over 50%
3. We used all the three column vectors and the accuracy was not better than 2

The vectorization of the texts in the above approaches was done using both CountVectorizer and TfIdfVectorizer

CountVectorizer: The vectorization is based on the frequency of the terms appearing in the corpus (group of documents constitutes a corpus)

TfIdfVectorizer: This vectorization penalizes the terms which are occurring frequently in the topics like a “sky” term is a common term within the “astronomy” topic. Term frequency is the percentage share of the word compared to all the tokens in the document whereas inverse document frequency is the logarithm of the total number of documents in a corpora divided by the number of documents containing the term.

Ex: You want to calculate the tf-idf weight for the word "computer", which appears five times in a document containing 100 words. Given a corpus containing 200 documents, with 20 documents mentioning the word "computer", tf-idf can be calculated by multiplying term frequency with inverse document frequency as

TF-IDF = (5/100)*log(200/20)

Once the terms are vectorized, the dataset is transformed into a sparse matrix whereing the document containing only the texts will have values and the rest of the terms are set as 0. Also the dimensionality is huge (around 10000 feature vectors). We tried using Truncated SVD to find the best fit components having good explained variance. But still the accuracy was not improved

4.1.1 Leveraging the Dictionary Dataset

In Dictionary dataset we selected the 2 column vectors, “Title” and “Description” and tried to approach the problem in unsupervised learning. We tried to create bag of words model using genism and then tried to extract topics using LDA (Latent Dirichlet Allocation). For creating the genism dictionary we had taken both the “title” and “description” of the training dataset as well

```
docs = dict_processed_docs + train_processed_docs
dict_corpus = gensim.corpora.Dictionary(docs)
```

As in above code we built the dictionary considering both the training as well dictionary dataset fields (title+description)

Once we have the BOW corpus and the dictionary we use gensims **LdaMulticore** to categorize the data into some number of topics as in the code below.

```
lda_model =  gensim.models.LdaMulticore(bow_corpus_dict,
                                        num_topics = 15,
                                        id2word = dict_corpus,
                                        passes = 10,
                                        eval_every=1,
                                        workers = None)
```

The num_topics is a hyperparameter to tune where the Coherence of the model is more and perplexity less. To find the best number of topics we find the coherence values by giving from the lowest number of topics from 3 to 21 and we plotted the coherence value in each iteration. In Fig 4.1 shows the plot for the coherence for different topics and found that for 15 topics coherence is good

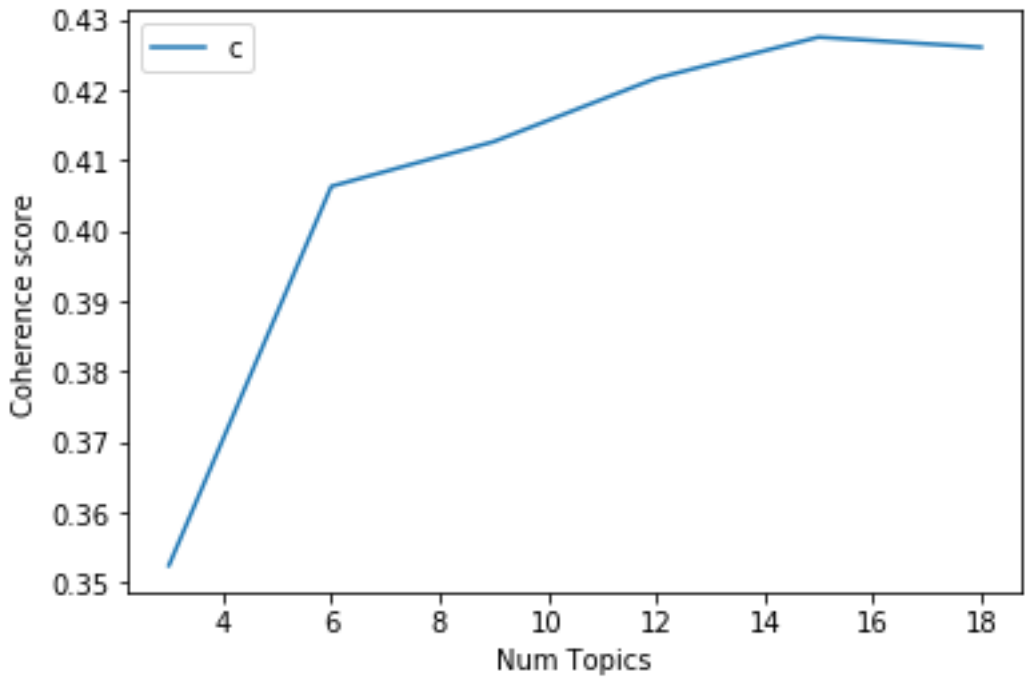


Fig 4.1 Plot of coherence of the LDA model with number of topics

We categorized the dataset into 15 topics and the top 10 keywords were extracted per topic. The same LDA model was used to categorize the training dataset(title+description) but with only the training corpus. We extracted “Topic” and “Topic_Keywords” in the training set. Fig 4.2 shows the intertopic distance map of each of the topics

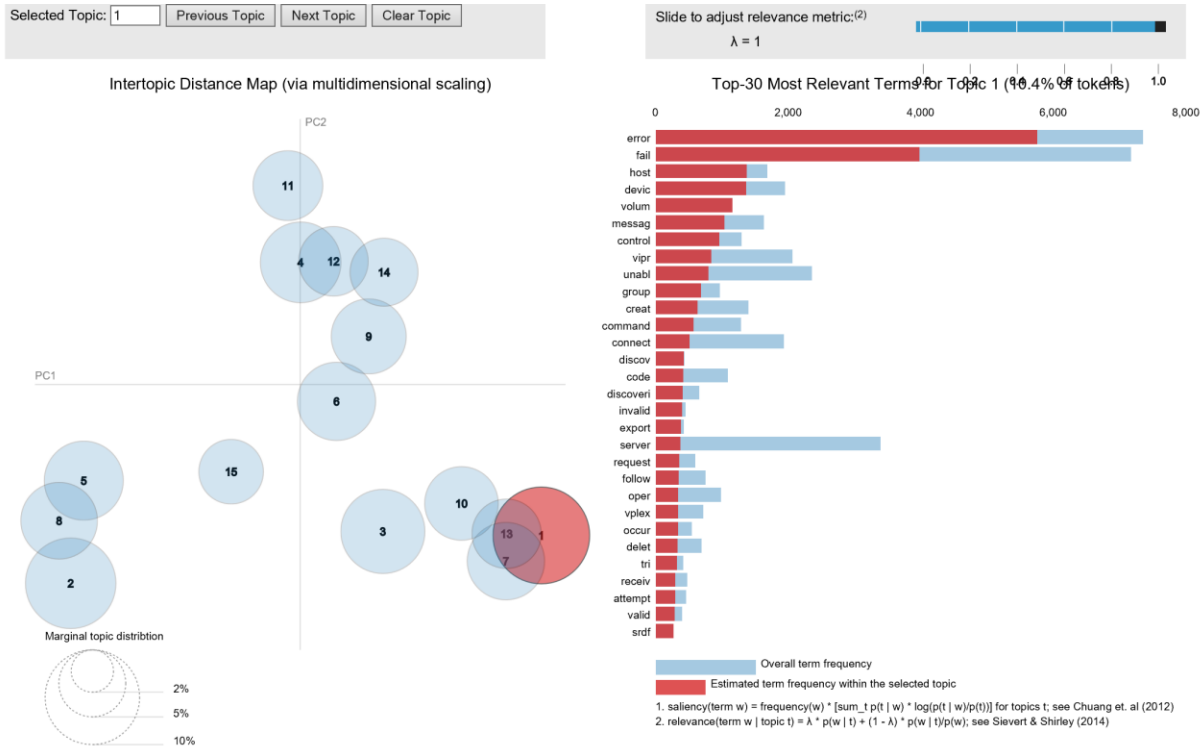


Fig 4.2 Top 30 Keyword frequency in each topic with their intertopic distance map

4.1.2 Building the baseline model

We selected “Topic_Keywords” and “body” fields of the training. We used RandomForestClassifier since that is the model giving the highest accuracy of 60%. The vectorizer used for the model was CountVectorizer.

```
classifier = RandomForestClassifier(n_estimators = 700,
                                    class_weight = 'balanced_subsample',
                                    #oob_score=True,
                                    max_features = "sqrt",
                                    n_jobs=-1,
                                    bootstrap=False,
                                    # min_samples_leaf = 4,
                                    # min_samples_split = 10,
                                    random_state=34)
```

```
classifier.fit(train_count,y_train)
```

Once we get a baseline model with a good accuracy, it is best to pickle the model for later use.

4.1.3 Boosting the model (AdaBoostClassifier)

We wanted to improve the model accuracy of the RF classifier. We used our baseline model as a baseline estimator in AdaBoostClassifier. This classifier requires a learning rate for optimal convergence we selected learning rate of 0.01 and also in each epoch, it reduces the loss and gives more weight to the misclassified sample in the next iteration thereby increasing the accuracy. This step takes a huge amount of time since the boosting process is sequential and more time taken for more number of estimators used.

Save the model

5 Submission of Validation

This is the final part of the competition. We select the validation dataset and apply preprocessing steps for all the fields done so far like “title”, “description”, “body”.
Apply lda model to title+description fields and extract keywords and topics.
With Topic_Keywords+body vectors as features, transform the data using countvectorizer
Used the model of the last step and perform prediction on the vectorized inputs.

6 References

David M. Blei, Andrew Y. Ng, Michael I. Jordan Latent Dirichlet Allocation

http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf