Documentation and customer support cases

By Kliton Andrea
Mentor: Matt Fornito

Introduction

GOAL:

Improve technical documentation in MuleSoft products

Client:

MuleSoft Product Management department

Data sources:

- Documentation in Ascii format published in Github
- Documents triggering support cases data source

Overview

1. Data cleaning and data wrangling
2. Exploratory Data Analysis
3. Prediction models
4. Conclusion and recommendations

Data Cleaning

1. Removed duplicated documents. Retained only the last version of documents.
2. Excluded images and code examples
3. Working only with Ascii formatted files

Data Cleaning

Created Pandas dataframe - each row is a document file.

Content of each document is in Ascii format.

Necessary to remove the formatting tags and other text elements.

Data Cleaning

Removing:

- Ascii doc format tags
- Web references (http://www.*)
- Code snippets (xml configuration, java code, etc.)
- Spaces and new line markers
- Replace periods(.) with underscores (_) if period is part of a token, i.e. 3.8.4 to 3_8_4.

Text normalization

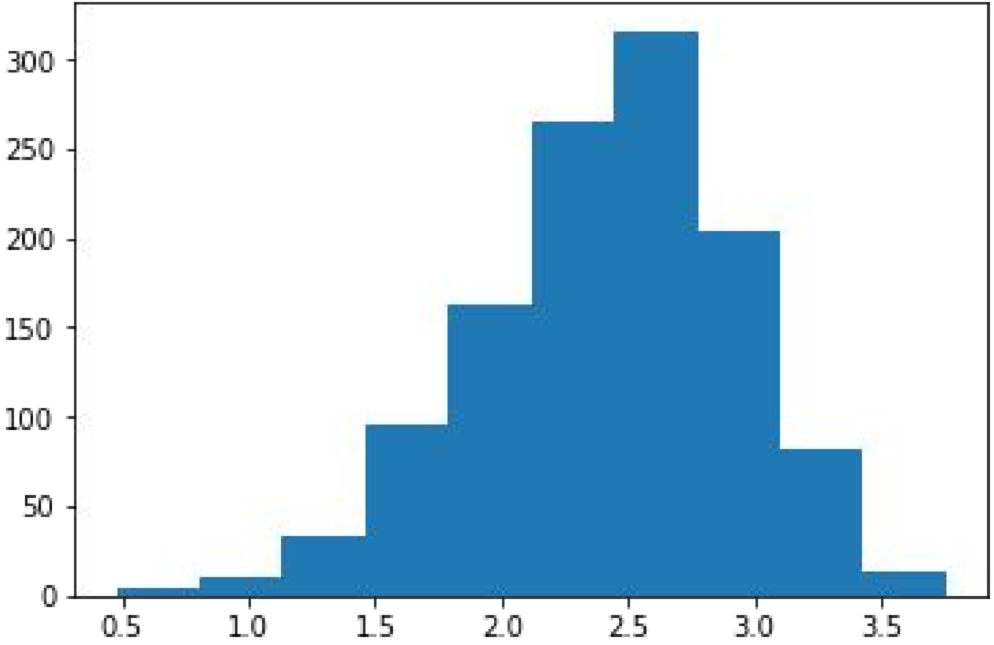
- Remove stop words
- Text stemmers
  - NLTK
  - sklearn
- Text lemmatization
  - spaCy (https://spacy.io/)

Used spaCy to retrieve word lemmas.

Retrieved bi-grams and tri-grams by applying Gensim phrase models

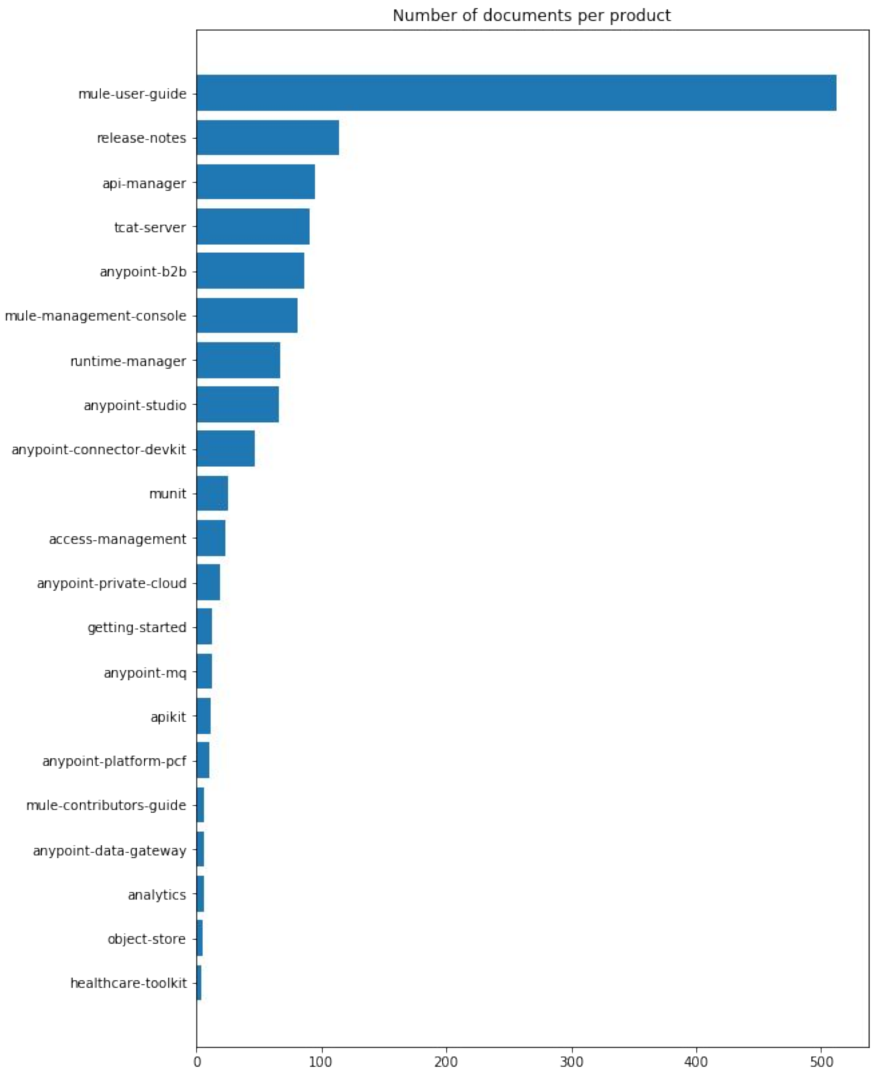
Exploratory data analysis

- 1187 documents
- 20 products
- Document length histogram from normalized text.
- Most of the document have a length of around 100s of tokenized words.
- Removed documents with less than 10 words.

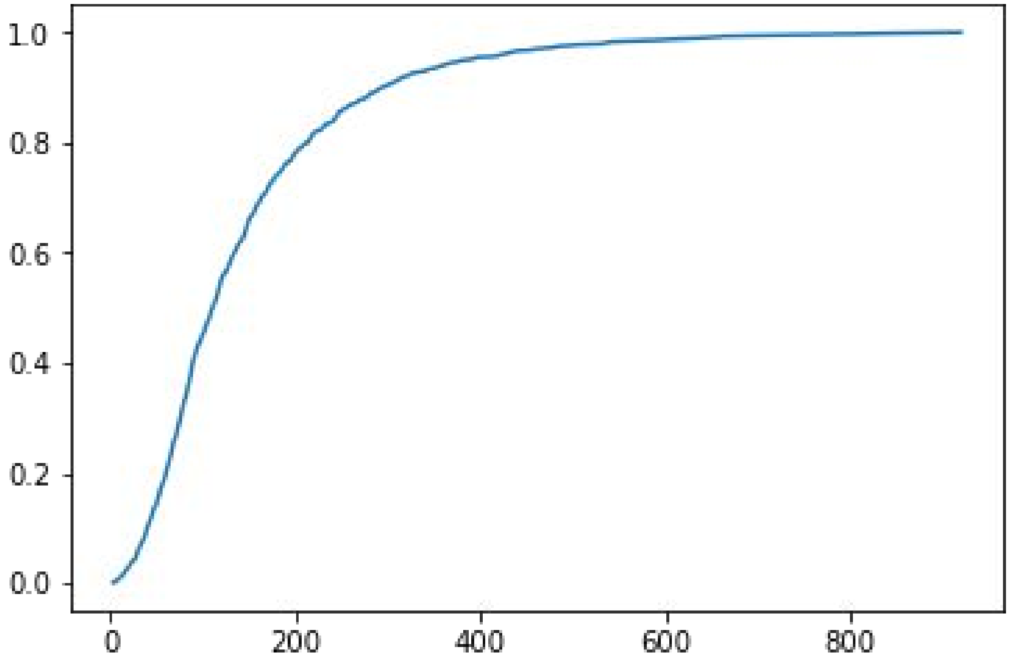


EDA

- Distribution of documents by product.
- Mule ESB is the core product. Its documentation spans at ⅓ of all the content



Exploratory Data Analysis

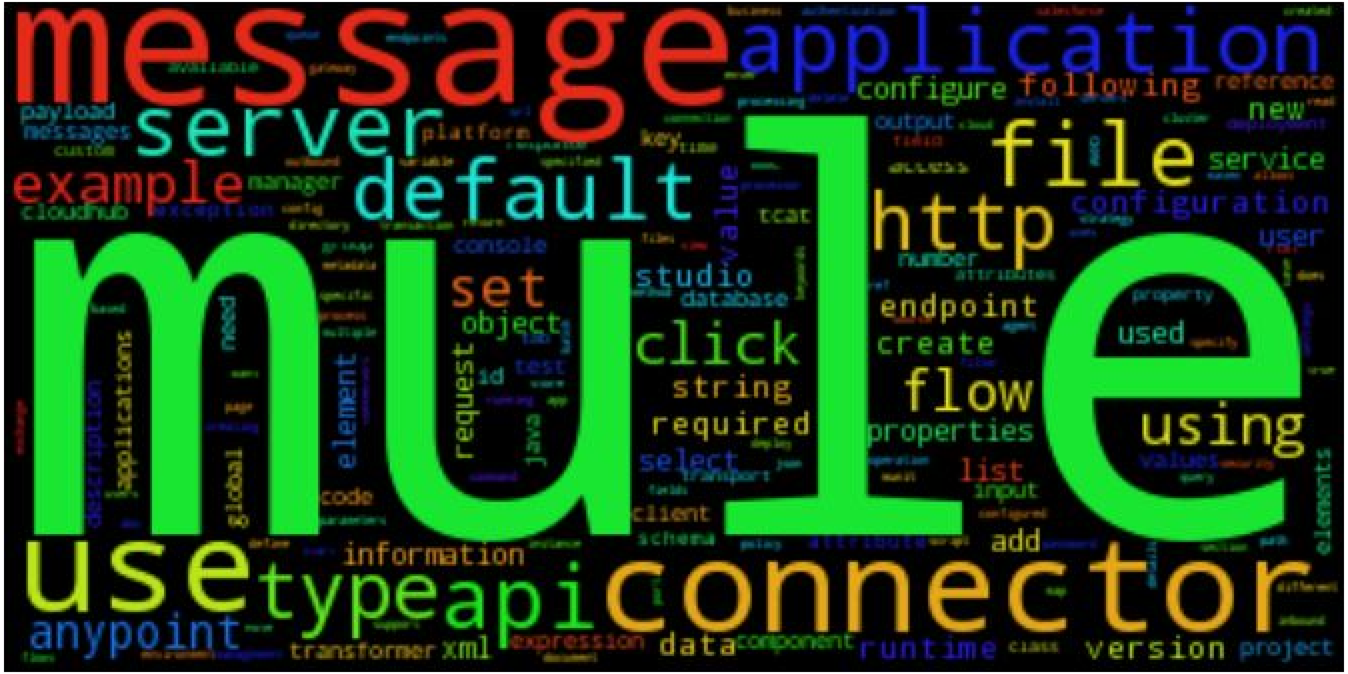


Word frequency proportion in documents.

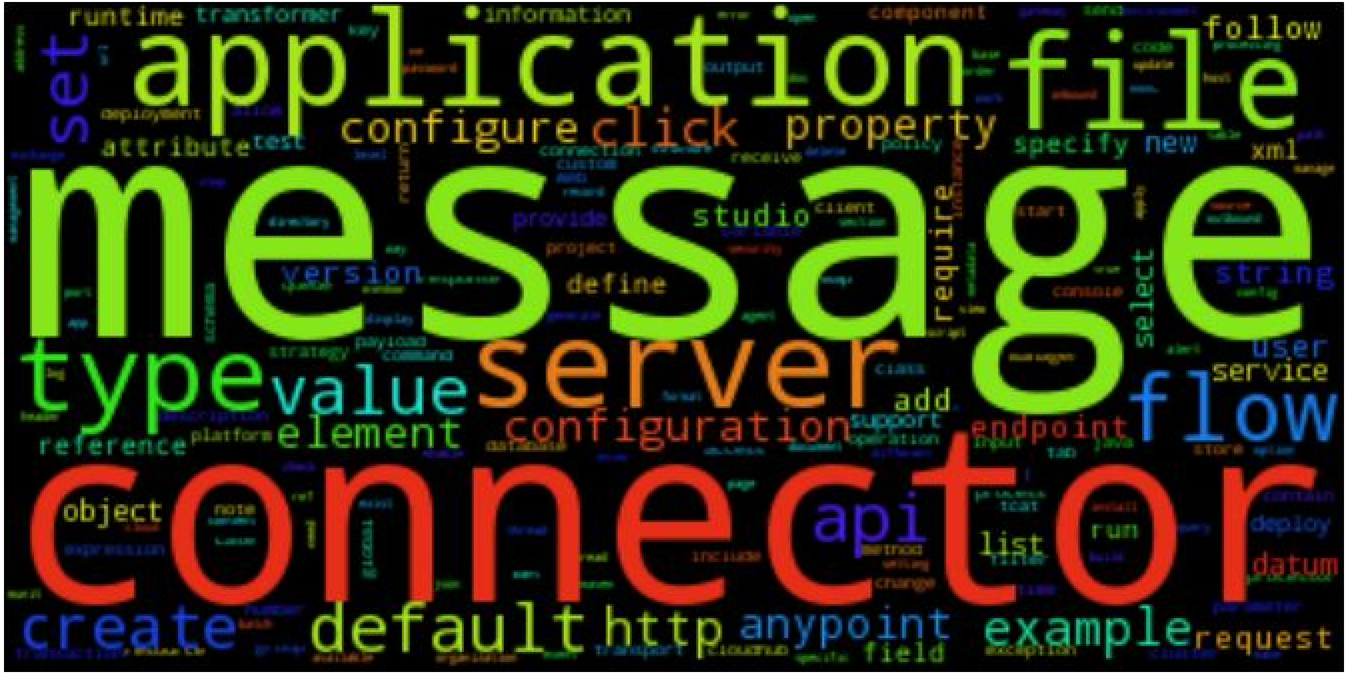
Y - proportion of words

X - number of word seen in x documents and less

Word cloud build on word counts



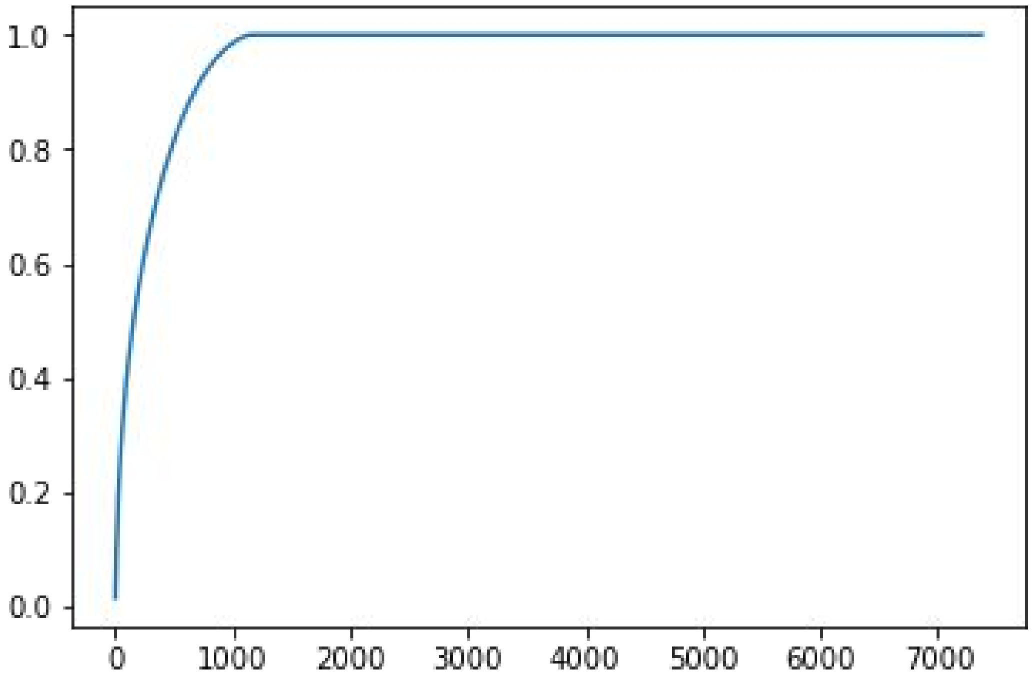
Word cloud based on tf-idf matrix



Exploratory Data Analysis

Tf-idf matrix with 7372 words and 1176 documents.

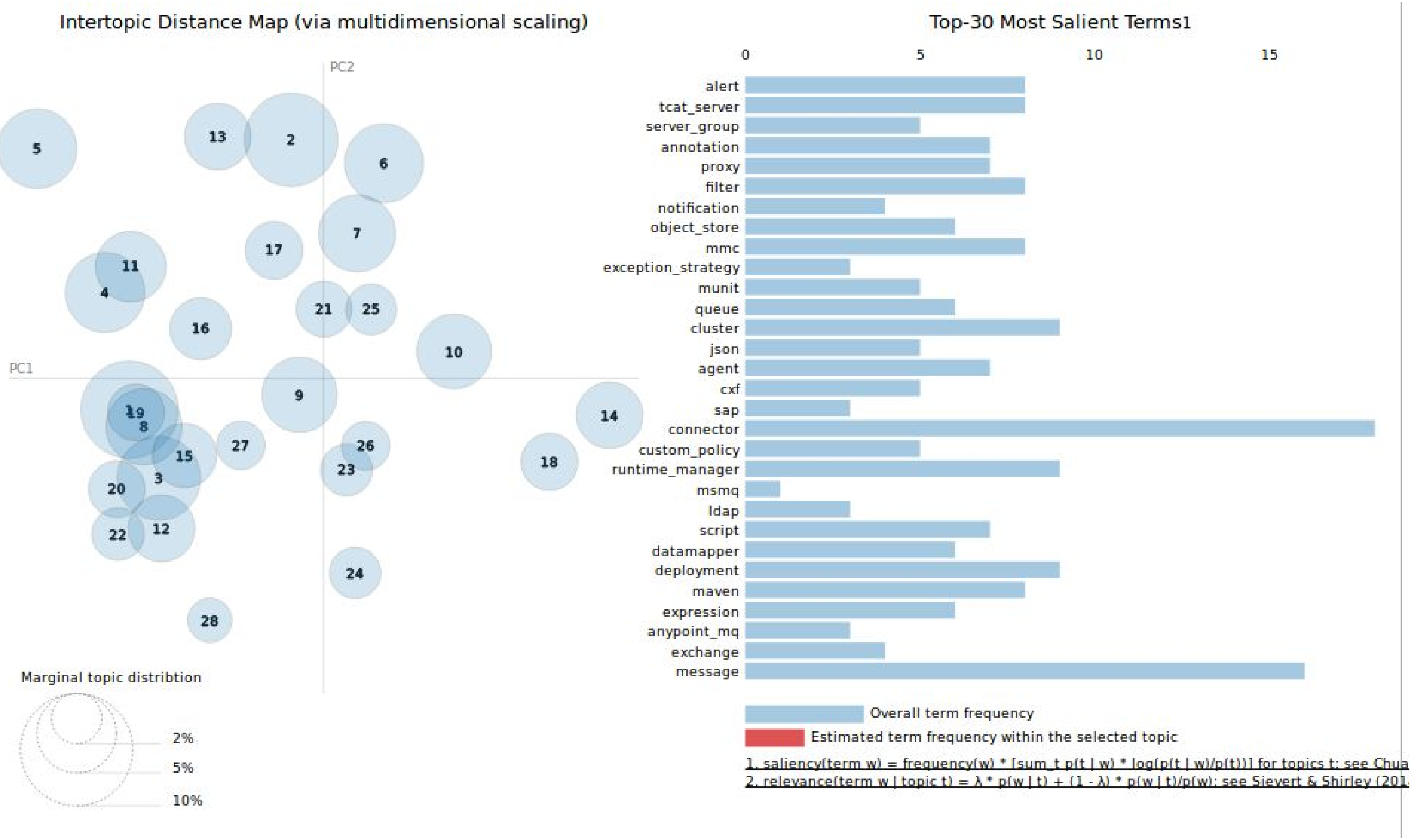
Applying PCA reduces the number of features (words) to 562.



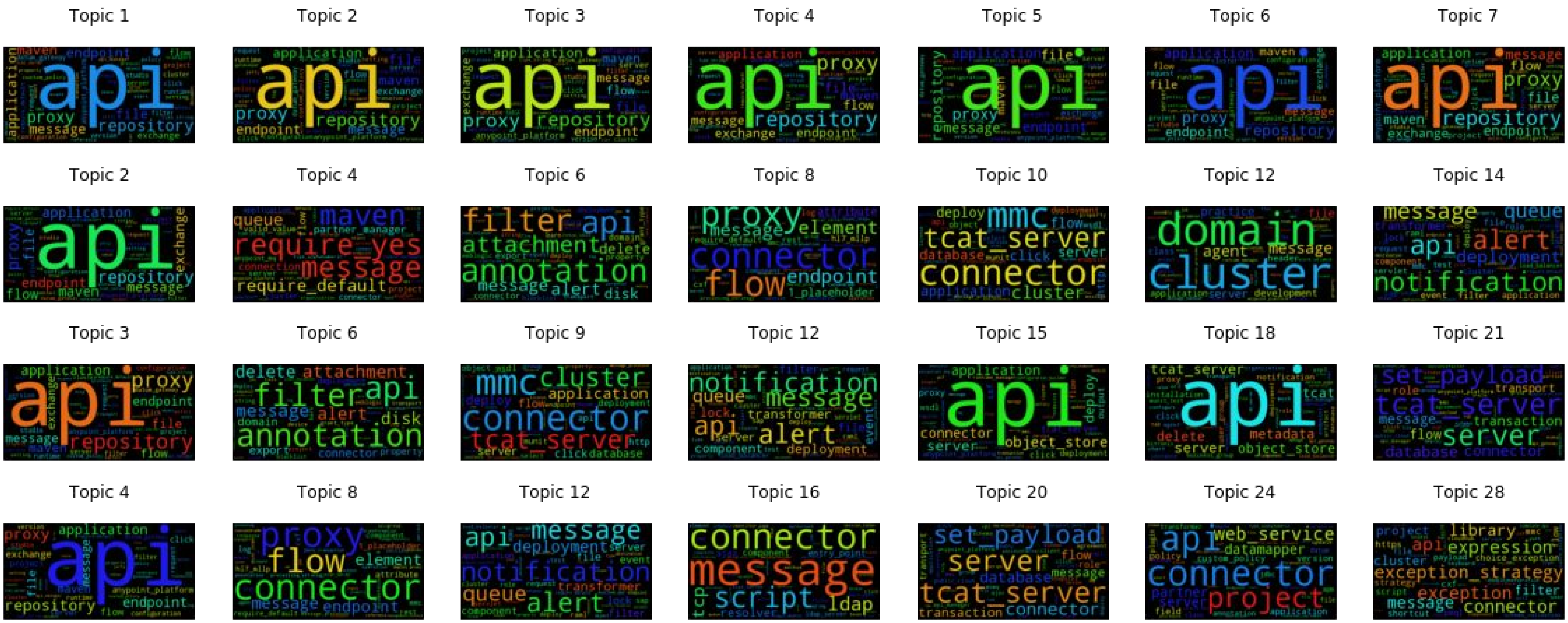
Topic Modeling

1. Latent Semantic Analysis
    a. No latent factors discovered
2. Hierarchical Dirichlet process
    a. Suggests an LDA topic model by defining alpha values
    b. Upper bound for number of topics is 56
3. Latent Dirichlet Analysis
    a. Minimum number of topics 24
    b. Maximum number of topics 56
    c. Chosen number of topics 24

LDA


Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
5
13
2
6
7
17
11
4
21
25
16
10
9
19
8
14
15
27
26
23
3
18
20
22
12
24
28
Marginal topic distribtion
2%
5%
10%
Top-30 Most Salient Terms1
0
5
10
15
alert
tcat_server
server_group
annotation
proxy
filter
notification
object_store
mmc
exception_strategy
munit
queue
cluster
json
agent
cxf
sap
connector
custom_policy
runtime_manager
msmq
ldap
script
datamapper
deployment
maven
expression
anypoint_mq
exchange
message
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chua
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (201

Word cloud for each of the topics



Prediction Models

1. Multinomial Naive Bayes
    a. Train score AUC = 0.56 and F1 = 0.28
    b. Test score AUC = 0.6 and F1 = 0.33
2. Support Vector Machines
    a. Train scores AUC = 0.6 and F1 = 0.33
    b. Test score AUC = 0.69 and F1 = 0.5
3. Random Forests
    a. Train scores AUC = 0.95 and F1 = 0.95
    b. Test scores AUC = 0.58 and F1 = 0.27
4. XGBoost
    a. Train score AUC = 1.0 and F1 = 1.0
    b. Test score AUC = 0.6 and F1 = 0.33

Conclusions

- Best prediction model is SVM
- Word features are important for predicting the documents that will trigger support cases.
- The complexity of the document is another important factor. It was established that long documents containing more code are prone to support issues.
- Reducing the number of support cases triggered by issues in documentation will result to reduced costs of $300K (as minimum).

Next Steps

- Create a corpus that can be utilized in a chatbot application to assist users with simple technical questions about the product.
- Include more resources like dedicated forums, Knowledge Base repository, Slack channels for NLP analysis.
- Use NLP methods to categorize different application configurations.