**Data 670 Data Analytics**

**Amir Ronaghy**

**Professor: Dr. Steve Knode**

**Assignment 6**

**August 12, 2018**

**Executive Summary**

This project involved the development of a document clustering application for NASA that could automatically cluster its policy documents together by topic. Although NASA stores its policy documentation in an online repository (i.e., the NODIS Library), it lacks the ability to query the documents by policy topic in a complete and consistent manner. This could potentially inhibit the awareness of NASA policies among the agency’s personnel and could also possibly lead to the expenditure of resources on the documentation of policies that have already been documented. The difficulties in retrieving NASA policies might thus result in the waste of taxpayer money and the violation of NASA and federal mandates, rules, and regulations. They could also undermine the agency’s ability to achieve its mission and ultimately compromise our economy and national security. The project’s objective was to develop an application capable of extracting the policy documents from the NODIS Library and clustering the documents together by topic to make it easier for the end users to locate the desired policies. The technical approach to achieving this object will involve a multitude of cutting-edge programming and data science techniques – webscraping, text mining, data visualization, natural language processing (NLP), and machine learning – using Python, which has become the preeminent language within the data science community for machine learning and artificial intelligence. The project’s ultimate goal will be to help NASA achieve its mission in serving our country’s public interests, protect our economic interests, and defend our national security by promoting the awareness of and identifying any gaps in the agency’s policies.

The project team developed three distinct clustering models using different combinations of NLP and machine learning techniques. The champion model involved the use of the k-means method, an unsupervised machine learning technique used for clustering, and the Latent Dirichlet Allocation (LDA) method, an unsupervised probabilistic modeling technique that can be used to identify documents topics among other purposes. This model outperformed the other two models in terms of the clustering quality by a wide margin and performed reasonably well as per the metrics defined in the key performance indicators (KPI). Furthermore, the document clustering application was able to extract 95 percent of the web entries associated with the policy documents from the NODIS Library, representing a substantial improvement over the current webscraping functionality. From this perspective, the project achieve a successful outcome. However, a qualitative review of the model’s textual and graphical outputs revealed clustering performance issues serious enough to warrant concern about whether the application would be fit for deployment to production. In its current state, the application could very well cause more issues for the project stakeholders than it resolves when it comes to enabling them to retrieve the policy documentation with greater ease and efficiency. Thus, while the use of the k-means clustering method combined with the LDA technique shows promise in terms of the ability to power a useful document clustering application, it would be best to delay the deployment of the application until it has been further refined. In particular, it is recommended that a future project team evaluate the possibility of 1) improving the quality of the LDA model’s outputs to allow for the identification of a more accurate set of policy topics, 2) determining whether a hierarchical or k-means clustering model would perform better when combined with a document-topic matrix created using the LDA technique, and 3) using deep learning techniques rather than the traditional text mining methods as the basis for engineering new features to be using in the clustering model.

Table of Contents

[Project Scope 5](#_Toc521873879)

[Problem Description 5](#_Toc521873880)

[Business Understanding 5](#_Toc521873881)

[Organization 6](#_Toc521873882)

[Stakeholders 6](#_Toc521873883)

[Define Business Area 7](#_Toc521873884)

[Business Objectives 7](#_Toc521873885)

[Business Success Criteria 7](#_Toc521873886)

[Background 7](#_Toc521873887)

[Research 8](#_Toc521873888)

[Gaps in this Problem Resolution 8](#_Toc521873889)

[Proposed Project 8](#_Toc521873890)

[Key Performance Indicators 9](#_Toc521873891)

[Project Insights of your Data Analysis 9](#_Toc521873892)

[Project Milestones 9](#_Toc521873893)

[Completion History 10](#_Toc521873894)

[Lessons Learned 11](#_Toc521873895)

[Data Set Description 14](#_Toc521873896)

[Data Set Description 14](#_Toc521873897)

[High-Level Data Diagram 15](#_Toc521873898)

[Data Definition/Data Profile 15](#_Toc521873899)

[Data Preparation/Cleansing/Transformation 16](#_Toc521873900)

[Data Preparation 16](#_Toc521873901)

[Data Cleansing 16](#_Toc521873902)

[Data Transformation 16](#_Toc521873903)

[Data Analysis 17](#_Toc521873904)

[Data Visualization 19](#_Toc521873905)

[Data Visualization 1 19](#_Toc521873906)

[Data Visualization 2 20](#_Toc521873907)

[Data Visualization 3 21](#_Toc521873908)

[Proposed Visualizations 22](#_Toc521873909)

[Predictive Models 24](#_Toc521873910)

[Predictive Model 1 24](#_Toc521873911)

[Predictive Model 2 27](#_Toc521873912)

[Predictive Model 3 30](#_Toc521873913)

[Predictive Model Review 33](#_Toc521873914)

[Final Results 34](#_Toc521873915)

[Analysis Justification 34](#_Toc521873916)

[Findings 34](#_Toc521873917)

[Review of Success 35](#_Toc521873918)

[Recommendations for Future Analysis 36](#_Toc521873919)

[References 38](#_Toc521873920)

# Project Scope

# Problem Description

NASA reportedly lacks any reliable means of helping its personnel locate policies by topic. Although NASA’s NODIS Library provides a document search feature, the functionality has been described by the NASA project lead as problematic. Specifically, it is unable to consistently deliver search results that are complete and accurate. An examination of the search feature has confirmed that it is indeed highly suspect. For example, queries for policies on information security and enterprise architecture did not return any results although both documents are clearly listed in the repository. An attempt to manually locate all relevant policies associated with a given topic within a list of 275 policies would be too inefficient to serve as a viable alternative.

These technical issues undoubtedly make it difficult for the NASA end users to locate all the relevant policies for a given topic. This could potentially inhibit the awareness of NASA policies among the agency’s personnel. In other words, a large portion of the NASA workforce might be uninformed about the particulars of NASA’s policies, or might be unaware that a given policy exists at all, if they cannot find the relevant documentation in the first place. This lack of awareness could also lead to the use of resources on the development of policies that have already been documented. Based on the example provided above, a scenario in which NASA stakeholders lead a redundant effort to craft the agency’s enterprise architecture policies even though they have already been published. The duplicative work in this example could be prevented simply by improving NASA’s ability to classify and organize its policies.

The difficulties NASA personnel would have in locating the agency’s policies might thus result in the waste of NASA resources and taxpayer money on unnecessary policy-related tasks and projects. The lack of awareness that would surely be caused by these document retrieval issues could lead to the violation of NASA and federal mandates, rules, and regulations. In fact, these violations are likely to occur if the employees are unable to reference the associated policies. Indeed, the NASA Inspector General’s report from September 2017 identified a number of management and performance challenges related to matters such as deep space exploration, the science portfolio, IT governance and security, contracting and grants, making it imperative for NASA to improve its personnel’s compliance with the associated policies (NASA Office of the Inspector General, 2017). Furthermore, this lack of awareness could potentially undermine NASA’s ability to achieve its mission with respect to the promotion of our country’s aeronautics, space exploration, and scientific discoveries. Given the multitude of NASA policies related to IT governance and security, human capital management, research and development, cooperation with third-party institutions, acquisitions and other matters related to NASA’s mission, the inability of NASA personnel to reference these policies could adversely impact not only the agency but also our country at large given the broad implications of the NASA mission in multiple respects.

# Business Understanding

This project will fall within the intersection of aeronautics, space exploration, science, national security, and economics. This project ultimately aims to help NASA in pursuing its mission with respect to serving our country’s national interests and the interests of the world at large. According to the NASA 2018 Strategic Plan, the agency’s mission is to:

1. Promote the exploration of space and the creation of new knowledge and opportunities for humanity;
2. Support the growth of our economy and improvement of our technologies in space and aeronautics;
3. Enhance our understanding of the universe and our place in it; and
4. Advance American leadership (NASA, 2018).

Without the sufficient documentation and awareness of the policies that align with NASA’s mission, it would be very difficult if not impossible for NASA to pursue its mission. Furthermore, the project seeks to ensure the proper awareness and documentation of NASA IT governance and security policies intended to protect the agency’s data and infrastructure from cyberattacks. Our space technological superiority is increasingly being challenged by the growing expertise among other countries as well as by nation states seeking to attack and defeat our space assets (Oleson, Silsby, & Skelly, 2012). Such an attack or defeat would not only compromise NASA’s ability to achieve its mission with respect to the advancement of our collective knowledge and understanding of space, science, and aeronautics. It could also potentially damage our economy and jeopardize our competitive advantages in space and aeronautical technologies while simultaneously bolstering the economics and technologies of adversarial nation states. Furthermore, given the potential military and intelligence dimensions of NASA’s assets and activities (David, 2015), a breach or destruction of our space assets could also have grave implications for our national security.

### Organization

The primary stakeholder organizations will be the NASA offices that are responsible for the development of the agency’s policies.

### Stakeholders

The first group of stakeholders consists of the NASA senior managers and executives, who are responsible for developing and approving the agency’s policies. The project would benefit these stakeholders by making it easier for them to identify any potential policy gaps. Currently, it would be difficult for these stakeholders to find and remediate any such gaps due to the deficiencies in the existing document search features. By creating a document clustering application that improves on the current functionality, these stakeholders would be better able to search for policies related to a specified topic and determine whether the given policies have been documented. This will enable them to resolve any potential policy games in a timelier and more efficient manner and help prevent any duplicative efforts in developing policies that have already been documented.

The second group of stakeholders are the NASA employees who would search for NASA policies for their own reference or those of their colleagues. Due to the aforementioned inadequacies of the document search feature, these stakeholders may encounter difficulties in locating the policies for a given topic. A robust document clustering application would increase the likelihood of these stakeholders finding the policies they need. This in turn would expand the awareness of NASA’s policies among the agency’s personnel and hence minimize the likelihood of policy violations due to a lack of understanding. The third group of awareness would be the members of the American public who are interested in referencing NASA’s policies. The project would benefit these stakeholders by enabling them to retrieve the policies with greater ease in pursuit of their commercial, academic, or public policy interests.

# Define Business Area

Please refer to the Business Understanding section above for a description of the impacted business areas.

### Business Objectives

Based on the rationale described in the Business Understanding section above, the following business objectives have been defined for this project:

1. Support NASA’s mission by promoting the awareness of and identifying any gaps in the policies that ensure the alignment of IT with business objectives, the governance and security of IT assets, the proper management of the agency’s human capital, the regulation of research and development efforts both within NASA and between NASA personnel and third-party institutions.
2. Help to preserve our country’s national security by promoting the awareness of and identifying any gaps in the policies that safeguard NASA’s infrastructure and data, whose breach or destruction could adversely impact our country at large given the reported military and intelligence aspects of our space assets.
3. Help to preserve our country’s economic and technological advantages vis a vis other nation states by promoting the awareness of and identifying any gaps in the policies that safeguard NASA’s infrastructure and data, whose breach or destruction could damage our economy and our space and aeronautical technologies while strengthening those of our adversaries.

### Business Success Criteria

To achieve a successful outcome, the project should result in the delivery of a complete and functional analytical pipeline capable of clustering the NASA policy documents, both the ones that currently reside in the NODIS Library as well as those that will be posted there in the future. Specifically, this means the application developed by the project should be capable of producing the results summarized below.

1. The application should improve on the current webscraping functionality. This means it should extract a higher percentage of the policy document web entries from the NODIS Library and place them into a corpus for analysis than the existing webscraping functionality, which is described in Research below. The current application can scrape just under 50 percent of the policy document web entries from the NODIS Library. The future application should be capable of scraping a substantially higher percentage of the associated web entries than the current one.
2. The application should separate the policy documents into the appropriate clusters based on their respective topics. The documents within the clusters should be reasonably close to the cluster centers, which would indicate that there is in fact a high degree of similarity among the documents within the clusters. Furthermore, the clusters should be well-separated from one another, meaning that the clusters should be distinguishable to enable the stakeholders to easily search for the policies based on their topics.

# Background

Please refer to the Problem Description and Business Understanding sections above for the project’s background information.

### Research

In December 2017, Jonathan Riley, who was a student in the UMUC MS in Data Analytics program at the time, developed three separate Python applications that could be used to extract the policy documents from the online NODIS Library and place the document contents into a format suitable for analysis. The first application is intended to perform a webscrape of the NODIS Library, put the URLs and text of the documents into a data frame, and load the data frame into an Excel file. The second is supposed to convert the policies documented as PDF files to text files. The third is meant to remove any extraneous characters (e.g., formatting metadata) from the text files. Taken together, these applications provide a solid and useful foundation on which this project can build with respect to the extraction of the documents from the NODIS Library.

### Gaps in this Problem Resolution

As mentioned in the Business Success Criteria section above, the current webscraping functionality can scrape just under 50 percent of the policy documents residing in the NODIS Library. This translates into a total of 136 documents that can be extracted and placed into a data frame in Excel format for analysis. While this quantity of data should suffice for the development of a functional predictive model, the model should ideally be trained and tested on the majority of the policy documents posed online. An increase in the volume of data available for training and testing will most likely translate into greater accuracy for the model. In addition, the Excel file generated by the current webscraping functionality for the data frame contains a large amount of formatting metadata, which could potentially compromise the quality of the predictive model and create additional work due to the need to remove the extraneous characters from the data frame.

# Proposed Project

The project’s objective will be to develop a document clustering application that can automatically pull policy documents from the NODIS Library and separate the documents into clusters based on their respective topics. This will enable the stakeholders to locate existing policies and identify any potential policy gaps more easily, thus mitigating the risks to NASA’s mission and the country’s economy and national security described in the preceding sections. The following is a summary of the technical approach that will be followed in achieving the project’s objective:

1. The existing webscraping functionality will be improved so that it can extract a higher portion of the policy documents from the NODIS Library and place their contents into a data frame in Excel format.
2. The documents’ text will be parsed, tokenized, and cleaned to ensure the terms will be relevant to the analysis to be performed.
3. Create visualizations of the text data to gain insights into the policy documents and identify any necessary adjustments to the technical approach.
4. New features will be engineered through the use of text mining and natural language processing (NLP) techniques to enhance the accuracy of the clustering model that will be developed.
5. Clustering models will be developed through the use of unsupervised machine learning techniques – k-means and hierarchical clustering – to group the documents together into distinct clusters based on their respective policy topics. The model that exhibits the best clustering performance will be selected as the champion model.

The application will be developed in Python due to its suitability for software development, text mining, NLP, machine learning, and data visualization. R will also be used to generate certain visualizations where necessary. In addition, the application script will be placed into a Jupyter notebook so it can be run in Spark, Anaconda or other cloud-hosted platforms suitable for the ingestion and processing of big data.

### Key Performance Indicators

The following KPI’s have been defined for the document clustering the application to be developed by the project:

1. **The predictive model should be trained on no less than 60 percent of the policy documents scraped from the NODIS Library.** As mentioned above, the current webscraping functionality is able to extract less than half of the documents in the NODIS Library and place them into a data frame that can be used for the analysis. A production grade application should be capable of scraping and clustering the clear majority of the policies published online at a minimum. Thus, the project team will seek to achieve a substantial improvement over the current webscraping functionality in this respect.
2. **The clustering model should achieve a mean silhouette coefficient greater than 0.5 when applied to the text data.** A score above this threshold would indicate that the model has identified a reasonable structure when clustering the documents whereas a score below this threshold would suggest that the structure is deficient and likely to be artificial (Spector, n.d.). Please see the Predictive Models section for details on the mean silhouette coefficient.

### Project Insights of your Data Analysis

The project team expects to meet the performance thresholds defined for each of the KPI’s described in the preceding section. As mentioned in this section, these expectations are based on the team’s first-hand experience with prior machine learning projects. The attainment of results beneath these thresholds in terms of the predictive model’s accuracy or its clustering of the data would indicate that the document clustering application to be created by the project is not fit for production. In addition, the project team expects that Python will be highly suitable as a programming language for webscraping, text mining, NLP, and machine learning given the extent to which it has been used in other projects for these purposes. Furthermore, it is expected that the use of Jupyter Notebook to run the application in Spark or other cloud-hosted analytics platforms will prove to be a superior option that the use of Python IDLE desktop client locally for the ingestion and processing of the document text data.

# Project Milestones

The following milestones have been defined for this project:

1. Select, describe, and analyze dataset – 6/18/18
2. Prepare and clean data – 7/8/18
3. Create visualizations of data – 7/15/18
4. Develop predictive models and analyze results – 7/29/18
5. Complete final report – 8/12/18

# Completion History

|  |  |
| --- | --- |
| **Week 1** | NA |
| **Week 2** | NA |
| **Week 3** | NA |
| **Week 4** | * Held kickoff meeting with the NASA project lead. * Obtained and performed initial discovery of NASA policy documents from NODIS Library. * Tested the existing webscraping and PDF converter applications to gauge their efficacy. * Performed research on the use of text mining, NLP, and machine learning for document clustering. * Completed project scope briefing document. * Completed Project Scope and Data Set Description reports. |
| **Week 5** | * Worked on developing Presentation 1 to present the proposed project to the DATA 670 class. |
| **Week 6** | * Completed and delivered Presentation 1 to the DATA 670 class. * Began development of the data preprocessing script. |
| **Week 7** | * Debugged and refined the data preprocessing script. * Completed the Data Preparation report. |
| **Week 8** | * Improved the data preprocessing script as necessary. * Created visualizations to generate insights into the text data. * Completed the Data Visualization report. * Completed and delivered Presentation 2 to the DATA 670 class. |
| **Week 9** | * Worked on developing the webscraping functionality and cluster model. |
| **Week 10** | * Completed the first iteration of the webscraping functionality and cluster model created using k-means method and TF-IDF matrix as a feature. |
| **Week 11** | * Completed the three candidate models and evaluated their performances based on the mean silhouette coefficient metric.   + Model 1 – K-means clustering model with TF-IDF matrix as a feature.   + Model 2 – Hierarchical clustering model with TF-IDF matrix and document similarity matrix as features.   + Model 3 – K-means clustering model with TF-IDF matrix and document-topic matrix as features. * Completed the Predictive Models report. * Completed and delivered Presentation 3 to the DATA 670 class. |
| **Week 12** | * Improved quality of the topic-term matrix created for Model 3 to enable identification of topics based on their associated terms. * Identified nine distinct topics within the document-topic matrix in Model 3 based on the topic-term matrix. * Revised Model 3 parameters to improve its performance and accuracy. * Completed analysis on potential reasons why cluster models identified the optimal number of clusters for the text data as 2. * Completed project status brief and sent it to NASA project lead for review. * Completed the Final Results report. |

# Lessons Learned

|  |  |
| --- | --- |
| **Week 1** | NA |
| **Week 2** | NA |
| **Week 3** | NA |
| **Week 4** | 1. The use of a webscraping functionality to extract the documents from an online repository can be technically challenging. Thus, it would be best to focus initially on the development of a workable model that can accurately classify and cluster the documents rather. This will help prevent the team from spending an inordinate amount of time on enhancing the webscraping functionality, which is of a lower priority than the predictive model development. 2. Due to the technical challenges in converting the PDF documents to a text format to enable them to be analyzed, the PDF documents should be excluded from the project scope given the scheduling constraint. Any attempts to convert the PDF documents to a text format should be undertaken only if time permits. |
| **Week 5** | 1. The project team had initially defined a KPI stating that the predictive model should separate the documents into the correct clusters based on their respective policy topics with an accuracy rate of no less than 80 percent when applied to the test data. This KPI was subsequently nullified based on the recommendation of the NASA project lead, who advised that each document would need to be labeled manually in order for this KPI to be tracked. Given the volume of documentation that needs to be processed by the model, the manual labelling of the documents would be highly impractical. |
| **Week 6** | NA |
| **Week 7** | 1. It should be kept in mind that the Python NLTK plain text corpus reader, which will be used to read the text files into a corpus, automatically tokenizes the text once it has been loaded into the corpus. Thus, there is no need to apply the NLTK tokenizer function to the text. Doing so could potentially create technical issues with the preprocessing steps, which would thus result in the use of precious time on debugging efforts that could be spent on other tasks more productively. |
| **Week 8** | 1. Certain visualizations such as the TF-IDF word cloud and network of terms graph can be generated more efficiently in R than in Python. Numerous technical difficulties were encountered when trying to create them in Python, which resulted in an inordinate amount of time spend on troubleshooting and research. For future projects, it would probably be best to create them in R rather than attempting to do so in Python. |
| **Week 9** | 1. The project’s original intent was to measure the performance of the clustering models based on the ratio between the total within sum of squares for all clusters and the total sum of squares between the clusters. The underlying assumption was that only the k-means method would be used to cluster the documents together. Because the hierarchical clustering method will be used as well, and the aforementioned ratio is more appropriate for the k-means method, this metric was deemed to be inappropriate for the project. The mean silhouette coefficient described in the Predictive Models section was selected as an alternative performance measurement technique that can apply to both clustering methods. |
| **Week 10** | 1. Originally, the planned technical approach was to focus on developing a workable cluster model before attempting to enhance the existing webscraping functionality. However, the refined approach selected by the project team required that a more effective webscraping functionality be created before the preprocessing, feature engineering, or model development steps could be completed. The following project served as a guide to the approach ultimately chosen for this project: https://beckernick.github.io/law-clustering/. |
| **Week 11** | NA |
| **Week 12** | 1. Regarding Model 3, a TF-IDF matrix was originally created to serve as the basis for the Latent Dirichlet Allocation (LDA) model that generated the document-term matrix feature for the k-means clustering model. However, the TF-IDF matrix needed to be replaced with a document-term matrix produced using the bag-of-words method. The reason for this was that LDA model only works well with integers rather than continuous values as inputs, meaning that a TF-IDF matrix should not have been used as the basis for the LDA model. 2. When using the LDA technique, one must be careful in setting the LDA model’s parameters to avoid performance issues. Within Model 3, the LDA model’s maximum number of iterations was originally set to 10,000 based on an example provided in an online document clustering tutorial, which resulted in an excessively long application runtime (i.e., about 45 minutes). When the maximum number of iterations was lowered to the default value of 10, the runtime was reduced to only 3 minutes. Thus, Model 3’s performance issue could have been avoided simply by accepting the LDA model’s default value for this parameter. |

# Data Set Description

|  |  |  |
| --- | --- | --- |
| **Complexity** | Student completes plan components.  High: 12-15 points More than 30,000 rows, more than 15 attributes, 3 or more data sets  Med: 8-11 points: Between 10,000 – 30,000 rows, between 10 and 15 attributes, 2 data sets  Low: 0-7 points: Less than 10,000 rows, less than 10 attributes, 1 data set. | 15 |

### Data Set Description

The dataset that the project team will use to develop the model is a collection of policy documents posted to the NODIS Library, which contains a total of 275 policy documents. Based on a review of several randomly selected documents, it appears that the vast majority of the documents consist of a header, footer, and the text of the policy. For more details on each of these elements, refer to the Data Definition/Data Profile section below. Because the project will involve the clustering of the policy documents based on their respective topics, the data of greatest interest will be the policy text. The documents vary greatly in terms of the volume of text, ranging from 76 words to over 40,000 words. The median word count appears to be around 1,500, which means there will be a great deal of text data to be mined and analyzed.

The text data will be gathered from the web entries for each policy document. In this respect, two caveats need to be mentioned. First, the web entries for certain policies appear to consist of little more than a link to a PDF file. Because of the technical challenges in converting PDF's to text files to enable them to be parsed and tokenized, the NASA project lead has recommended that the PDF documents be excluded from the project scope, at least during the project’s initial phase. Second, certain web entries consist of little more than a hyperlinked table of contents for the associated policy document. Due to the technical challenges in extracting the text data from each hyperlinked section in this scenario, the text data will initially be scraped from the web entry pages only. As time permits, the project team will evaluate the options for pulling the text data from the hyperlinked sections of the policy documents. Nevertheless, given the volume of the text contained in the text files, there should be enough data to enable the project team to develop a viable document clustering model. One possible approach for working around these two obstacles would be to manually copy and paste the contents of all the policy documents into text files that could be used to generate a corpus for the clustering model. However, in addition to being inordinately time consuming and inefficient, this approach would defeat one of the key parts of the project’s objective, which is to develop a robust application capable of automatically clustering new policies as they get posted to the NODIS Library.

Due to the unstructured textual nature of the dataset, the data will need to be parsed, tokenized, and cleaned so that each document will be converted into a collection of individual terms relevant to the analysis. In turn, text mining and visualizations will be used to analyze the term frequencies, associations, and similarities among the words in the documents. NLP techniques will be employed to engineer features that identify the most important terms in the corpus, the document topics, and the similarities among the documents. The k-means and hierarchical clustering methods will be used to create models capable of separating the documents into distinct clusters that each represent a given topic. Finally, the mean silhouette coefficient metric described in the Predictive Models section will be used to assess the clustering performances of the models and select the champion model.

### High-Level Data Diagram

Because the dataset to be used by the project will be a corpus consisting of unstructured text files, this section is not applicable to the project.

### Data Definition/Data Profile

The following is a summary of the structure of the typical NASA policy document posted to the NODIS Library. Based on a review of several of the policy documents, this summary would appear to apply to the vast majority of the documents.

1. There is a header that includes the document title as well as extraneous characters such as the document ID and date, which will need to be removed. Certain documents contain a header at the top of each page, which means the header text appears throughout those documents.
2. The body of the document contains the text of the policies, which would naturally be the focus of the analysis.
3. There is a footer that contains extraneous characters such as page numbers, URLs, and document ID that will need to be removed. Certain documents contain a footer at the bottom of each page, which means the footer text appears throughout those documents.
4. Each policy document posted to the NODIS Library has a unique URL. Though the URLs would be irrelevant from an analytical perspective, their use as inputs to the webscraping functionality will be critical to ensuring the policy documents can be scraped from the NODIS Library.
5. Each policy document posted to the NODIS Library contains HTML tags that were used to format and structure its associated web entry page. The HTML tags will also be immaterial from an analytical perspective but crucial to ensuring the successful extraction of the policy documents from the NODIS Library since the tags will need to be used to enable the webscraping functionality to scrape the text data from the web entries for the policy documents.

# Data Preparation/Cleansing/Transformation

### Data Preparation

The following the steps will be used to prepare the NASA policy documentation for the analysis. All tasks described below will be performed in Python.

1. All the web entries representing the NASA policy documents will be scraped from the NODIS Library and assembled into a dictionary consisting of the titles and text extracted from the documents.
2. Extraneous characters such as numbers, single letters, punctuation marks, non-ASCII characters, and HTML tags will be removed from the text since they are irrelevant to the analysis.
3. All terms will be converted to lower case to prevent terms that are capitalized differently from being improperly treated as separate terms by the algorithms to be used later on.
4. All extraneous whitespaces will be removed to prevent them from affecting the tokenized words.
5. The policy document text will be tokenized, meaning that the text will be broken up into a vector of individual terms. This step will ensure the text can be analyzed through text mining and NLP and the documents can be clustered through machine learning.
6. Lemmatization will be applied to the terms to convert groups of words into their dictionary forms. This accounts for factors like parts of speech, the meaning of the word in the sentence, the meaning of the word in the nearby sentences, etc. For example, “good,” “better,” and “best” would all be lemmatized to “good” (NSS, 2017). This step will be performed to ensure the model treats groups of terms with the same meaning as a single term.
7. Stop words will be removed from the text. Stop words are those that add no value to the analysis either because they are common to the language generally (e.g., I, you, we, is, are) or because they will obviously occur frequently throughout the corpus given the nature of the text. For example, "NASA" would be a stop word in this context because the text pertains to the NASA policies, and thus the word "NASA" would obviously be very common in this case. Both the standard list of English stop words in the Python NLTK library and a custom list of stop words will be used for this purpose.
8. The preprocessed corpus will be stored in a new dictionary consisting of the titles and text extracted from the documents.

### Data Cleansing

Please refer to the “Data Preparation” section above for a description of the steps to be followed for cleaning the data.

### Data Transformation

The first new feature that will be created is the term frequency-inverse document frequency (TF-IDF) measure, which will be represented as vectors of term-document pairs in the form of a matrix. TF-IDF measure is a useful feature for text mining and NLP since it gives a higher weighting to terms that appear frequently in certain documents but do not occur frequently throughout the corpus (Rose, n.d.). In contrast, if a given term has a high absolute frequency in the corpus but appears in a large number of documents throughout the corpus, this will diminish the term’s TF-IDF measure. On the other hand, the traditional document-tem matrix is based on the absolute frequencies of the terms throughout the corpus and thus might not provide as much insight into the meaning of a given document as the TF-IDF matrix could.

The second new feature that will be engineered is the document similarity measure, which is a metric that will be built on top of the TF-IDF measure and can be used to determine the degree of similarity between a pair of documents. The document similarity feature will be represented as a matrix consisting of document pairs and similarity scores for each one. These similarity scores will reflect the cosine similarity between the document pairs and fall within a range of zero to one. The closer the score is to one, the greater the similarity between the documents and vice versa (Sarkar, 2018-a). The document similarity feature will serve as an input to the hierarchical cluster model that will be developed by the project.

The third new feature to be generated is the document-topic measure, which will be displayed for each document-topic pair in a matrix. This metric will be built on top of a document-topic matrix to be created beforehand using the bag-of-words method. The document-topic matrix will be produced by the LDA method, which is an unsupervised probabilistic modeling technique that can be used to identify the topics of the documents (Kumar, 2017-a). The document-topic measure also falls within a range of zero to one, with one representing the strongest association possible between a document and topic. The main drawback of the document-topic matrix is that the topics are not labeled explicitly but rather are referenced through a series of ID numbers. Hence, another output of the LDA method, the topic-term matrix, will be used to identify the topics through an evaluation of the most frequent terms associated with each topic ID (Sarkar, 2018-a). Subsequently, the document-topic measures will be incorporated into the k-means clustering model to be created by the project. The outputs of the k-means and hierarchical cluster models will be compared to determine which model performed better in separating the documents into the appropriate cluster. The model that exhibits the superior performance will be selected for deployment into production.

### Data Analysis

As mentioned previously, this project involves the use of NLP and machine learning in developing a document clustering application. Hence, Python will be used to develop the predictive model that will drive the application and also to perform the preprocessing, text mining, visualization tasks that need to be completed beforehand. The reason for this is that Python has emerged as the language of choice for artificial intelligence (AI) projects, including those involving the use of NLP, due largely to its enormous range of high-quality libraries (Gall, 2018; Pointer, 2018). In particular, it features a large deep learning ecosystem that is larger than that of R or any other language both in terms of the number of libraries and the frequency of their use among data scientists (Piatestky, 2017). The vast majority of the most widely used deep learning libraries are compatible with Python, and many of them are in fact Python-first projects (Allen & Li, 2017; Pointer, 2018; Wilder-James, 2017).

Because of its sizable deep learning ecosystem, the use of Python would benefit this project by providing a wide range of options for building a complete NLP pipeline that would power the document clustering application. In addition to offering myriad options for deep learning, Python offers two state-of-the art libraries for text mining and NLP – NLTK and spaCy – along with the scikit-learn and genism libraries that are suitable for the machine learning tasks that will need to be completed (Sarkar, 2018-a). In addition, the Word2Vec deep learning model, which is compatible with Python, could potentially be used for certain feature engineering tasks to help ensure the development of a robust document clustering application (Sarkar, 2018-b). For the visualization tasks such as building the word clouds, frequency plots, and network of terms graph, Python features a multitude of libraries such as NLTK, matplotlib, wordcloud, and NetworkX that would be useful (Grando, 2017; Kadam, n.d.; Ladd et al, 2017).

Placing the document clustering application into a Jupyter notebook will ensure it can be run in Apache Spark and other cloud-hosted analytics engines. As mentioned earlier, the use of Spark is expected to provide a superior option compared to the use of Python IDLE desktop client locally for the ingestion and processing of the text data, particularly due to Spark’s aptitude in the ingestion and processing of big data. It is also expected that Spark would be a more robust alternative to the legacy Hadoop MadReduce method of big data processing. Because Spark can ingest and process data in memory, it should offer a superior level of performance compared to MapReduce, which performs its jobs on disk instead. In addition, Spark enables the completion of a wide variety of operations such as selecting, transforming, and analyzing the data in a single job, whereas MapReduce must often be combined with other frameworks for this purpose (Scott, 2015). Furthermore, the Spark machine learning library, SparkMLlib, can be integrated with Python through Jupyter Notebooks and a series of APIs, which would thus make Spark a natural fit as an analytics engine for a Python project.

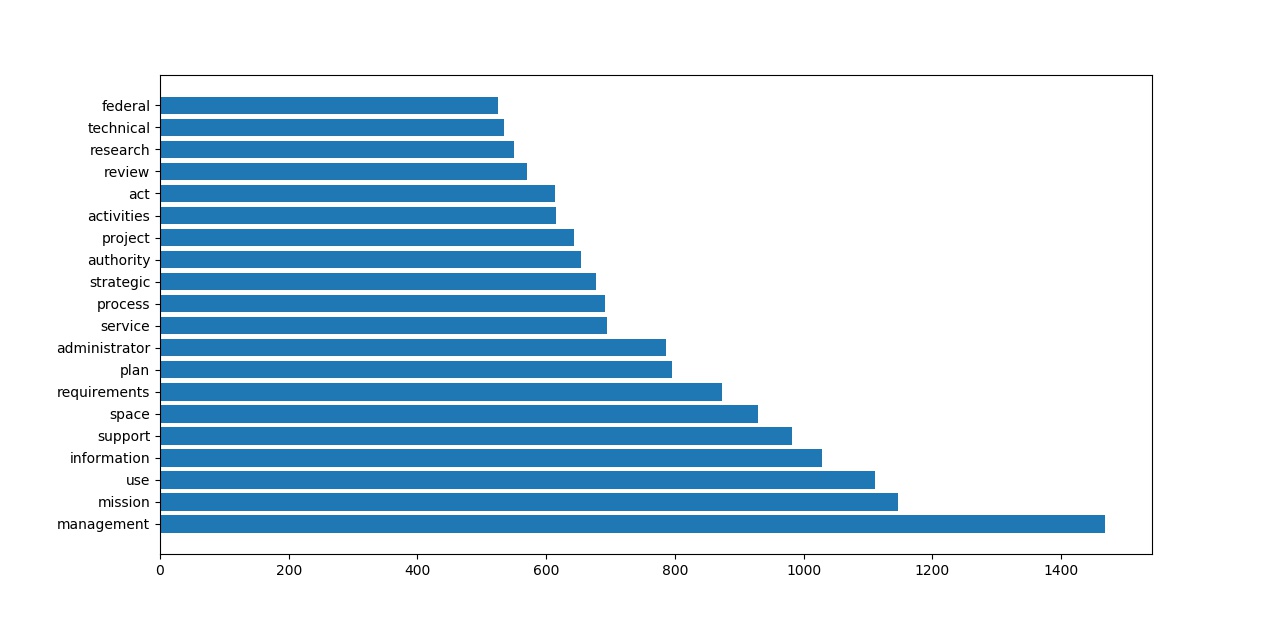
# Data Visualization

### Data Visualization 1

In the visualization below, the frequencies of the 20 most common terms in the corpus are arranged by frequency in ascending order. The visualization was created first by storing the frequency distributions of the terms in a variable using the NLTK library’s FreqDist function, then defining another variable to store the 20 most frequent terms, and finally by using the Matplotlib library’s plt function to generate the bar chart showing the term frequencies. The following is a summary of some of the most interesting insights gained from the frequently recurring words displayed in the visualization:

* “mission” and “strategic” – These terms suggest that supporting the NASA mission and its strategic goals is a recurring theme of the policies.
* “information” – This term indicates that the handling, protection, and use of information by NASA personnel is a frequent topic among the policy documents and points to the existing of a large number of documents dealing with information security and privacy.
* “space” – The recurrence of this term along with other terms such as “technical,” “research,” “review,” and “project” suggests that the documents commonly refer to the policies related to space technologies, research and development, and their underlying projects.
* “service” – The implications of this term are ambiguous since it could refer either to NASA’s public service to our country or perhaps to the services provided to its internal or external customers (e.g., IT services). Given that they both tend to be frequent topics among government policy documents, it is likely that the NASA policy documentation refers to both of these topics.
* “federal” – This term indicates that compliance with federal laws, rules, and regulations is a critical theme of the NASA policies.

These findings thus align with the project’s problem description, which highlighted the importance of this project in ensuring the advancement of NASA’s mission; federal laws, rules, and regulations; and preservation of our country’s national security and its economic and technological advantages through the protection of NASA’s data and infrastructure. In doing so, these findings would appear to have substantiated the business case for the project. At the same time, certain insights gathered from the visualization would also seem to confirm the value of using the TF-IDF measure to generate features to be used in the clustering models. As shown in the visualization, many of the frequent terms such as “management,” “use,” “support,” and “requirements” are rather generic and do not provide much insight into the contents of the policy documents. Therefore, a clustering model that uses the absolute terms frequencies as inputs would most likely compromise the quality of the outputs compared to a model that incorporates the TF-IDF measure.

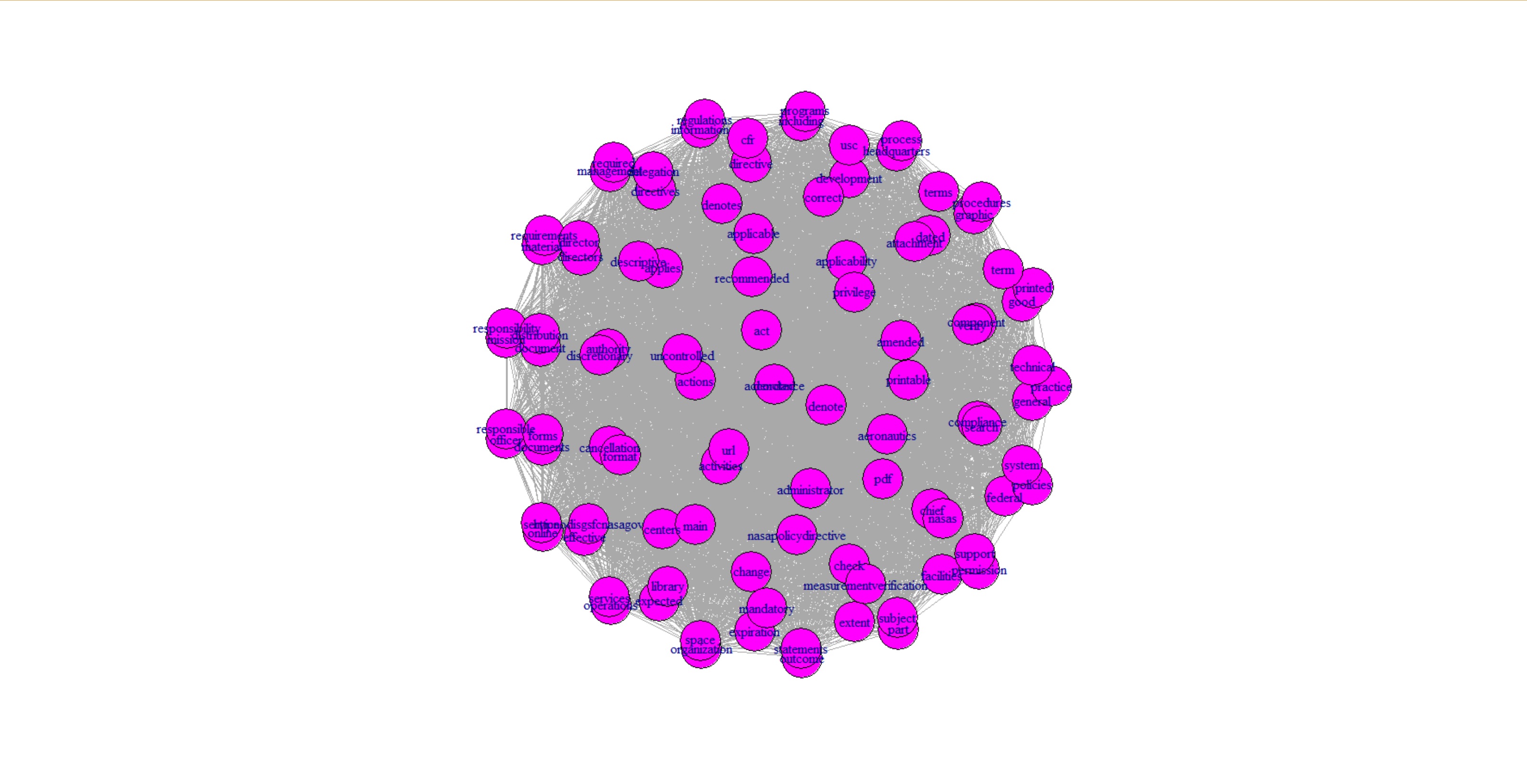


### Data Visualization 2

The visualization below is a network of terms graph, which depicts the relations among the words that appeared together most frequently in the corpus. This graph was created in R, first by generating a term-document matrix to store all words with a sparsity level of 0.4, and then by using the igraph library to produce the graph based on the term-document matrix with different combinations of layouts, sizes, and colors. After repeated trial and error, the optimal sparsity level for the term-document matrix and combination of layout, size, and color for the graph were selected. The following are some of the most interesting combinations of terms displayed in the graph:

* “general,” “technical,” and “practice” – This combination allude to the existence of documentation on policies governing the conduct of technical tasks and projects.
* “federal,” “system,” and “policies” – This combination suggests that certain policies deal with the need to comply with federal policies on the management and use of systems, most likely in an IT and information security context.
* “facilities,” “support,” and “permission” – This combination most likely points to the existence of policies controlling access to NASA facilities and the overall management of the facilities.
* “responsible,” “officer,” “forms,” and “documents” – This combination appears to reference the policies identifying the roles and responsibilities for the management of NASA forms and other documents.
* “information” and “regulations” – This combination indicates that policies aligning with federal and agency regulations on the use, management, and security of information by NASA personnel have been documented.

These findings do not impact the model development approach or expectations on what it will produce. However, the visualization did provide some additional insights into the types of topics that might be covered by the policy documents such as the governance of information systems, document management, and facilities management.



### Data Visualization 3

The visualization below is a word cloud illustrating the most important words in the corpus based on the TF-IDF measure. Word clouds are typically used to depict the importance of the in terms of their absolute frequencies. However, since the bar chart described above was already created for that purpose, the TF-IDF measure was used instead to produce the word cloud below. As explained in the Data Transformation section above, the TF-IDF measure gives a higher weighting to terms that appear frequently in certain documents but do not occur frequently throughout the corpus. Thus, the TF-IDF measure can be useful in providing insights into the distinctions in the contents of the various documents in a corpus. On the other hand, since the absolute term frequencies simply measures how often a word appears in the corpus without regard for the number of documents in which it appears, there might not be as much clarity in this respect.

To generate the word cloud, a document-term matrix leveraging the TF-IDF measure was first created in R. After repeated trial and error, any terms with a sparsity level of 0.75 or above was filtered out of the document-term matrix. Then, the wordcloud package in R was used to generate the word cloud. As shown below, the importance of the words is signified by both their sizes and colors, with the largest words representing the most important ones. According to the word cloud, “security” is the most important term in the corpus based on the TF-IDF measure, which makes sense given that one would expect a government agency to have an abundance of security policies in place, especially an agency as critical to our national security and interests as NASA. Among the other important words, “foreign” appears to be among the most critical, which is what one would expect given the sensitivity of NASA’s dealings with foreign governments and institutions. In addition, “agreement” suggests that policies have been documented to govern the conduct of agreements, perhaps those amongst internal NASA organizations, between NASA and external government and non-government organizations, or both. The importance of “employee” points to the existence of policies on employee conduct and human resources, which would exist across all types of organizations. Finally, “omb” alludes to the OMB standards, rules, and regulations with which federal agencies would need to comply. These finding thus provide additional insights into the topics that might be discovered amongst the policy documents and could potentially serve as the basis for clustering the documents.



### Proposed Visualizations

One proposed visualization would be a word bubble, which depicts the most frequent words in a corpus as multi-colored bubbles and displays the frequency of each word when the user hovers over that word’s bubble. By illustrating the words in a visually appealing manner while also displaying the word frequencies in numerical terms, a word bubble would embody the benefits of the traditional frequency plots and word clouds while resolving the deficiencies of each one (Combs & Roman, 2017). Hence, it would be beneficial for the project to evaluate the use of a word bubble to illustrate the absolute word frequencies, and perhaps and importance of the words based on the TF-IDF measure, if time permits.

Another proposed visualization would be a bigram frequency plot. A bigram is a sequence of two adjacent elements – usually letters, syllables, or words – in a string of tokens. A bigram frequency plot would therefore benefit the project by depicting the most frequent combinations of words appearing together in the corpus, which could thus provide additional information about the context in which the most frequent terms appear (Grando, 2017).

# Predictive Models

### Predictive Model 1

The Model 1 is a k-means cluster model in which only the TF-IDF matrix described in the preceding sections above has been included as a new feature. K-means clustering involves the partitioning of a data set into the specified number of clusters, represented as k, and the random assignment of each observation to one of the clusters. The k-means algorithm then iteratively reassigns the observations to the appropriate clusters based on the closeness of the cluster centroids to the observations until the results no longer change. At this point, a local optimum has been attained, meaning that the mean distance between the observations within each cluster has been minimized (Hastie et al, 2013). The elbow method is typically used to identify the optimal k value for a k-means model. However, in this case, the elbow method indicated that the optimal k value would be two, which would not be a viable option for this project given the multitude of policy topics that were uncovered by the visualizations. Thus, it was necessary to select a k value through repeated trial and error, which resulted in the selection of a k value of 12.

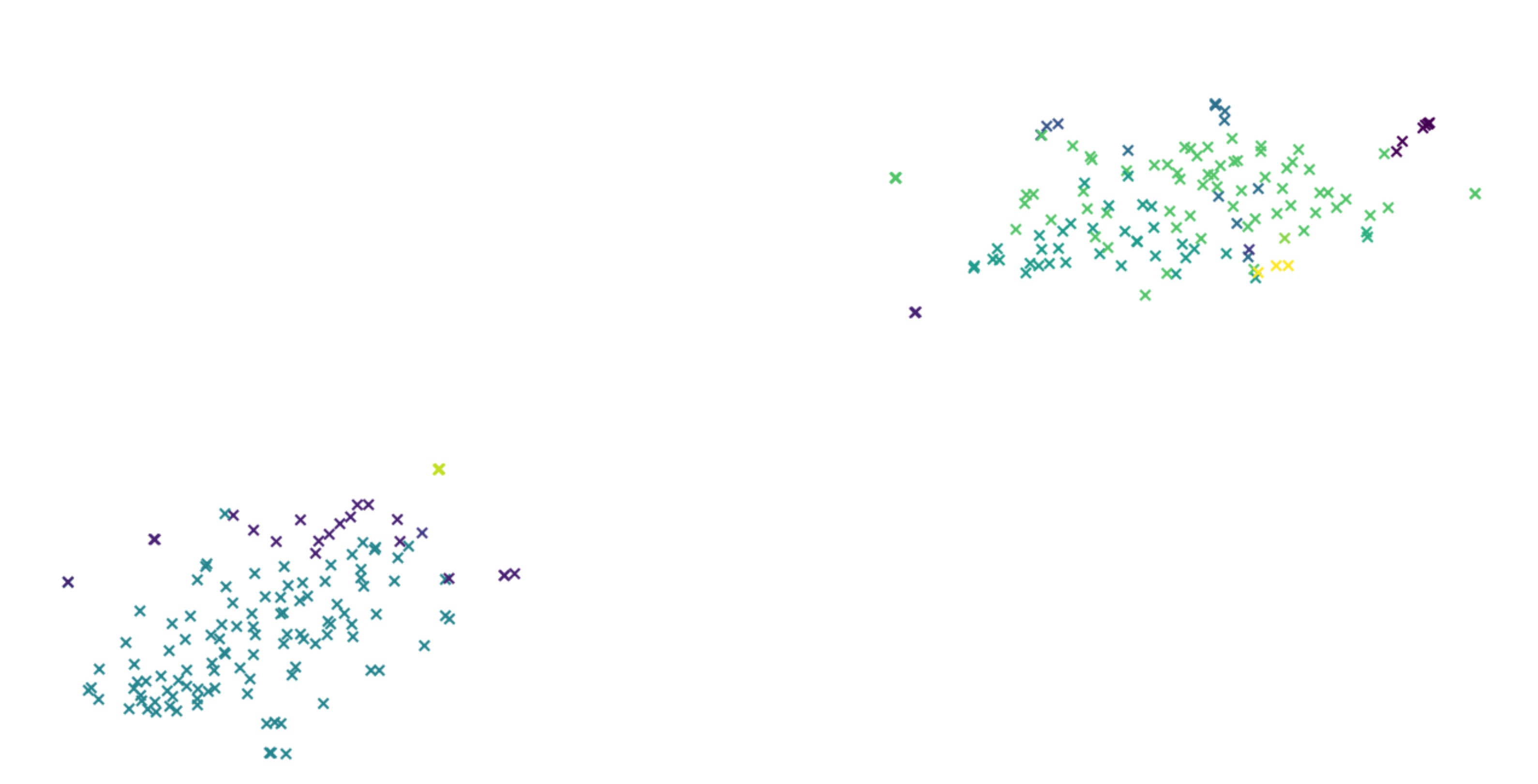
The graph for this model depicts the similarities among the TF-IDF vectors on a two-dimensional scatter plot with the similarities computed using the t-distributed Stochastic Neighbor Embedding (t-SNE) method. This technique involves:

1. Estimating the similarity between a pair of data points in a high-dimensional space as the conditional probability that the first data point would pick the second one as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at the first data point.
2. Estimate the similarity between the counterparts of the two data points in a low-dimensional representation based on a conditional probability similar to the one described in step 1 above.
3. Finding a low-dimensional data representation that minimizes the mismatch between the two conditional probabilities mentioned in steps 1 and 2 (Becker, 2016).

To create this graph, the TF-IDF matrix was first converted to a lower dimensional space to reduce the sparsity of the matrix. Second, a two-dimensional representation of the TF-IDF matrix was created using the t-SNE method. Finally, the vector embeddings from the two-dimensional matrix were plotted on a scatter plot to illustrate the groupings of the observations into the clusters (Becker, 2016).

The average silhouette method, which measures the clustering quality based on how well the observations lie within their respective clusters at different k values, was used to evaluate the performance of all three models. The primary advantage that this method has over the techniques for analyzing the performance of cluster models is that it can be used to assess both k-means and hierarchical clustering models (Boehmke, n.d.-a). The optimal k value is determined by the average silhouette width, which correlates positively with the clustering quality. In other words, the greater the average silhouette width for a given k value, the higher the clustering quality is at that k value (Boehmke, n.d.-b; Kassambara, 2017). In this context, the mean silhouette coefficient for all samples within the data would serve as the metric for assessing the models’ performance. This metric encompasses a range between -1 and 1, with the former representing incorrect clustering and the latter signifying a perfect clustering performance (scikit-learn developers, 2017-a).

The model’s performance was evaluated through both quantitative means (i.e., the mean silhouette coefficient) as well as qualitative means. First, a review of a sample of three clusters showed that the model performed poorly in grouping together the documents based on their policy topics. Although the model appeared to place documents of similar topics in the same clusters, it also grouped together documents with no apparent similarities to one another. For example, one of the clusters contained documents on policies related to communications and correspondence, real estate, classified information, and occupational health. The model’s suspect performance was confirmed by the mean silhouette coefficient of 0.11 for the optimal k value of 12, meaning the model was barely closer to the ideal score of 1 than to the worst possible score of -1. In other words, the model did not perform significantly better than one could have by randomly grouping the observations together. As shown in the graph below, there was a great deal of overlap among the clusters generated by the model. While one could detect identify clusters based on the manner in which the vectors with the same colors are grouped together, the graph shows that the clusters are not well separated from one another. Thus, one can only conclude that the model’s cluster assignments were highly unreliable. While it is certainly possible that the model’s performance could be improved through additional iterations, one could certainly question whether the k-means method combined with a TF-IDF matrix as a feature would be fit for a robust, production-grade document clustering application.

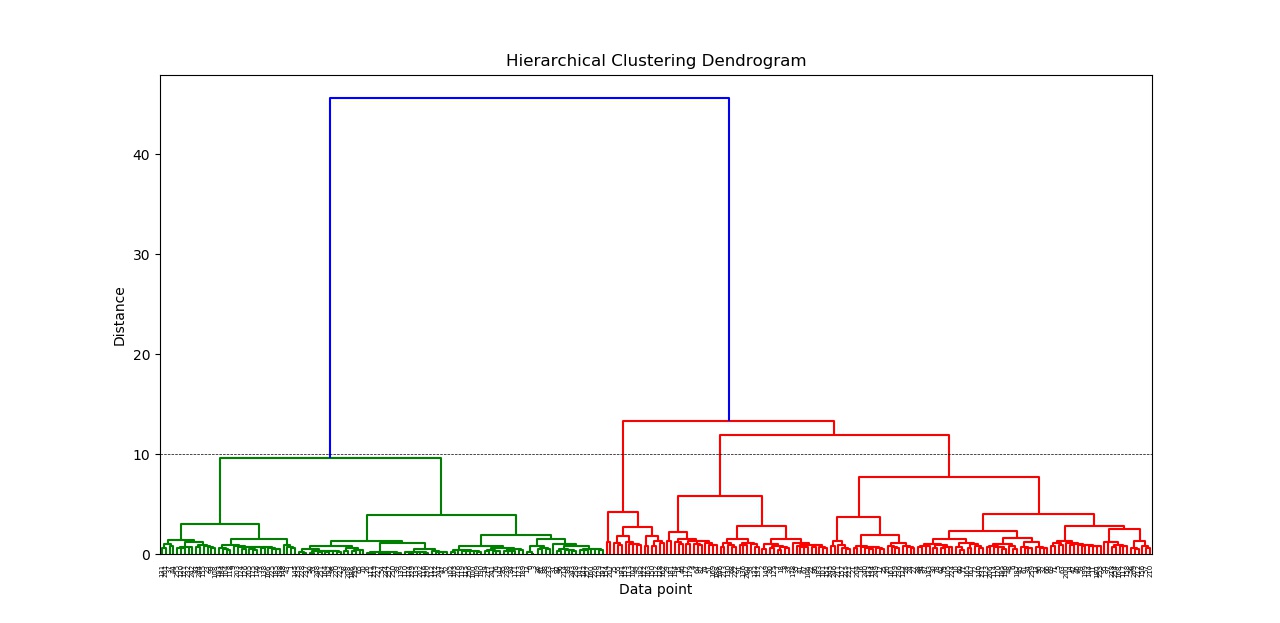


### Predictive Model 2

The Model 2 was created using the hierarchical clustering technique and incorporated a document similarity matrix as a feature. Hierarchical clustering offers two key advantages compared to the k-means method. The first is that the number of clusters (i.e., the k value) does not have to be pre-determined, which obviates the effort one would otherwise spend trying to determine the optimal k value through trial and error or the use of a technique such as an elbow plot. The second is that the output is conveyed in a tree-like graph called a dendrogram that tends to be more visually appealing and easier to interpret than a cluster plot (Hastie et al, 2013). The bottom-up, or agglomerative, method is the most popular hierarchical clustering technique. It derives its name from the fact that it initially treats each observation as a distinct cluster, represented as leaves in the dendrogram, and then merges the clusters together iteratively into branches until they have all been agglomerated into a single cluster at the top. The similarities among the observations are denoted by the closeness of the points where the leaves and branches merge. The closer the merging points are to one another, the greater the similarity among the corresponding observations (Hastie et al, 2013; Manning, Raghavan & Schütze, 2008).

The document similarity matrix that was engineered for use as a feature in this model consists of document pairs and the cosine similarities between each one. The cosine similarity represents the cosine of the angle between the feature vector representations of two observations. The lower the angle between the observations, the higher the cosine of the angle, and hence the greater the similarity of the observations. In other words, a cosine of 1 would represent a perfect match between the pair of documents, while a cosine of 0 means that the documents are as dissimilar as possible (Sarkar, 2018). The TF-IDF matrix described previously served as the foundation for the document similarity matrix in the sense that the features extracted from the documents using the former were incorporated into the latter.

As was the case with the Model 1, the second one’s performance was evaluated using the mean silhouette coefficient described in this section above along with qualitative means such as inspecting the cluster assignments and the graph generated by the model. Similar to the Model 1, the Model 2 exhibited a poor performance in clustering the documents. Its mean silhouette coefficient was only 0.12, meaning it only barely outperformed the Model 1 in this regard. In addition, a review of three sample cluster showed that several of the documents assigned to the same cluster appeared to have little or no relation to one another as was the case with the Model 1. For example, one of the clusters contained documents pertaining to policies on research and technology, export control, labor relations, and environmental compliance and restoration. Furthermore, the dendrogram produced by the model, which is displayed below, suggests that the optimal number of clusters in this case would be two based on the point in the graph where the distances among the clusters at the different levels are the highest (Hees, 2015). As mentioned in this section above, the notion that only two clusters would suffice to group the documents together is highly implausible given the number of potential topics identified in the visualizations. The one advantage presented by the dendrogram compared to the graphs generated by the other two models is that it could potentially provide a framework for organizing the documents in a hierarchical grouping structure. In other words, certain types of documents could be arranged into two broad categories and then broken down further into subcategories under each one as appropriate. On the other hand, it would be unfeasible to do so using the cluster plots produced by the other two models. Unfortunately, the Model 2’s performance was such that it would not be fit for use in production regardless of the benefits afforded by the use of the dendrogram.



### Predictive Model 3

The Model 3 was creating using the k-means clustering method and incorporated a document-topic matrix generated through the use of the LDA technique as a feature. As described in the Data Transformation section above, LDA is an unsupervised probabilistic modeling technique. A LDA model can be described as a collection of composites made up of parts, with the documents representing the composites and the words and/or phrases (i.e., n-grams) embodying the parts in a topic modeling context, (Lettie, 2018). LDA can be used to automatically discover the topics contained by documents by representing the documents as mixtures of topics and the topics as mixtures of words (Chen, 2011; Lettie, 2018). When applied to topic modeling, the underlying assumption of the LDA technique is that each document has been produced as follows:

1. Determine the number of words N the document will have according to a Poisson distribution.
2. Select the topic mixture for the document based on a Dirichlet distribution over a fixed set of K topics.
3. Generate each word in the document by first picking the topic based on the topic distribution specified in step 2 and then using the topic to generate the word.

Each document would subsequently be represented as a mixture of topics based on the words that were generated in this fashion (Chen, 2011).

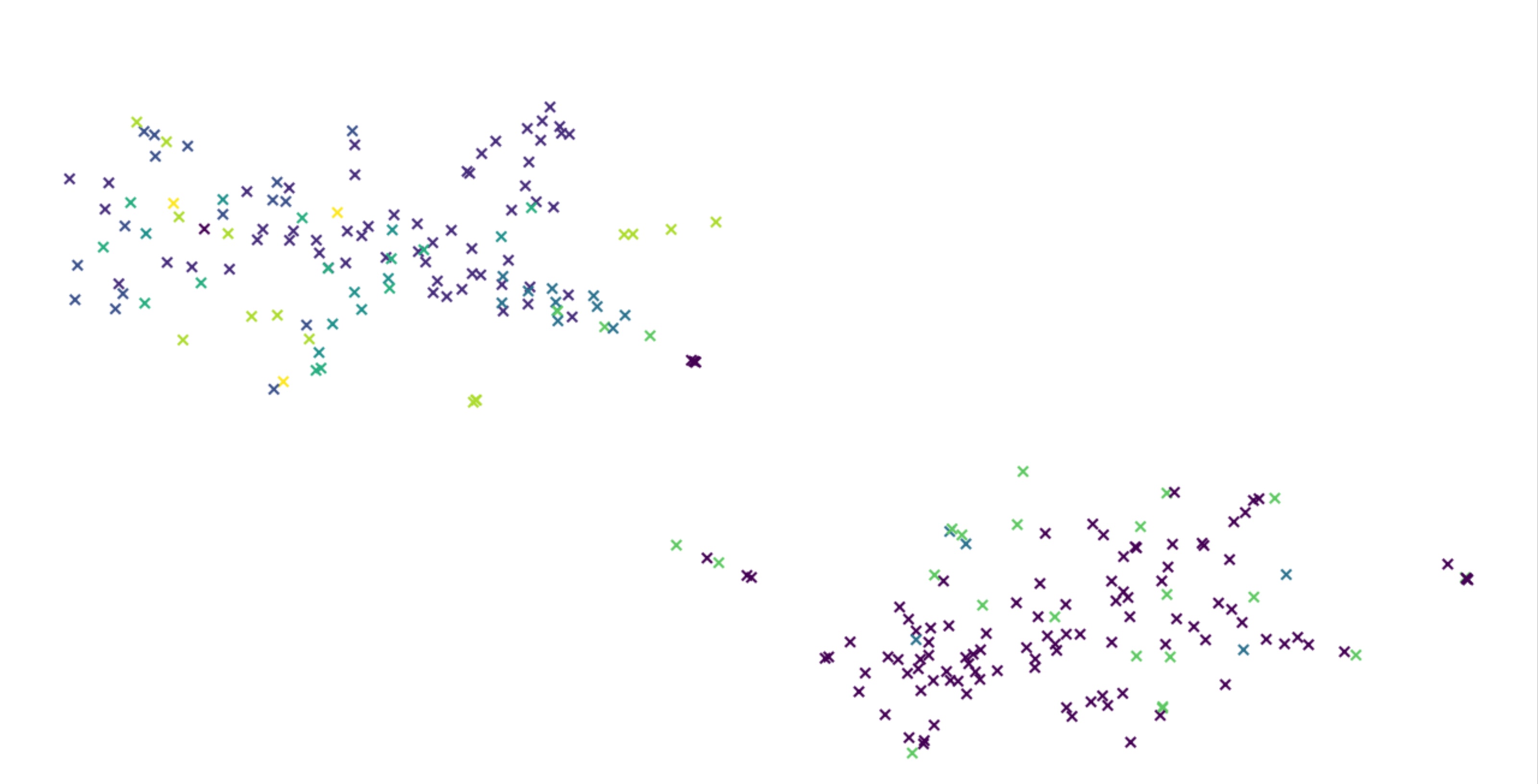
To enhance the performance of the k-means cluster model, LDA was used to generate a probabilistic topic model built on top of the features extracted from the documents through the document-term matrix created through the bag-of-words method. This probabilistic topic model consists of two features. The first, the topic-term matrix, describes the probability of selecting a given word when sampling a particular topic. The topic-term matrix can be viewed as a breakdown of how much each word contributes to the given topic. The second table, the document-topic matrix, describes the probability of selecting a given topic when sampling a particular document and would ultimately serve as a feature of the Model 3 because of its utility in measuring the strength of the relationships among the documents and topics (Lettie, 2018; Sarkar, 2018). Because these matrices reference the documents’ topics with ID numbers (e.g., T1, T2, etc.) rather than calling out the specific topic names, the topic-term matrix was created so that the topics could be identified through an assessment of which terms were most strongly associated with each topic ID. Using this technique, nine distinct topics were extracted from the text data:

1. Audit and accounting
2. Research and development
3. Security, legal and regulatory compliance
4. Public etiquette (e.g., treatment of flag and public officials)
5. Information management
6. Privacy
7. Space communications
8. Physical resource management
9. Intellectual property

Based on the total of nine topics, the k value of the k-means clustering model was changed from 12 to 9 to confirm whether aligning the number of clusters with the number of topics would improve the model’s clustering performance. Indeed, the Model 3’s mean silhouette coefficient improved from a score of 0.55 with a k value of 12 to 0.63 with a k value of 9. In other words, the revised k-means clustering model performed well above the mean silhouette coefficient threshold of 0.5 that was used to determine whether a model’s clustering performance would be acceptable.

In terms of the model’s performance, the Model 3 outperformed the other two by a substantial margin with respect to the mean silhouette coefficient. Whereas the Model 3 achieved a score of 0.63, the first and Model 2s attained scores of 0.11 and 0.12, respectively. This metric would thus suggest that the Model 3 performed fairly well in clustering the documents together based on their policy topics. However, the qualitative performance assessments would not appear to substantiate that conclusion. In particular, a review of the cluster assignments indicated that the model clustered together many dissimilar documents as was the case with the first two models. For example, one cluster consisted of documents on policies related to the conduct of agreements, human capital management, export control, and the management of information and communications. The results displayed by the cluster plot were also concerning. As shown in the graph below, the clusters represented by the groups of vectors of the same color overlap with one another to an extent that would make it difficult to distinguish amongst them. Thus, the despite the Model 3’s promising performance as measured by the mean silhouette coefficient, the lack of separation among the clusters in the cluster plot together with the inspection of the cluster assignments raise serious questions about whether this model is fit for use in a production-grade application.

Compared to the other two models, the Model 3 would appear to be the superior one based on the silhouette metric since its score was far closer to the ideal score of 1 than the scores of the other two models. Hence, the Model 3 should be selected as the champion model and should be the focus of the project team’s technical efforts for the duration of this project and perhaps beyond. Based on this result, one could conclude that the use of the LDA technique would significantly enhance the performance of a k-means clustering model when it comes to document clustering compared to the use of either cosine similarity measures or TF-IDF measures. However, the performance issues described in the preceding paragraph would need to be resolved before the model can be deployed to production. Otherwise, the use of this model as part of a production application could potentially end up causing more problems than it solves for the project stakeholders.



### Predictive Model Review

Please see the preceding section for a review of the comparative strengths and weaknesses of each of the three models and recommendations on how to proceed with the project.

# Final Results

### Analysis Justification

The soundness of the technical approach used for this project is reflected in the fact that project team leveraged the techniques recommended for document clustering problems by professional data scientists and developers who have published their works for the Python open-source user community. In particular, Nick Becker had used a webscraping and modeling approach similar to the one used for this project to develop an application capable of clustering together the laws passed by the U.S. Congress based on their respective topics. Based on the modeling results Becker had posted to his GitHub repository, one could conclude that his model achieved a reasonable degree of success in terms of its clustering performance (2016). In addition, the use of the cosine similarity measures and LDA model as features to be used in a document clustering model has been rationalized and demonstrated by a multitude of other data scientists such as Abhijeet Kumar (2017), Brandon Rose (n.d.), Dipanjan Sarkar (2018), as well as the developers of the Python scikit-learn library (2017-b).

This project’s outcome has further demonstrated the validity of the technical approach followed by the project team. Specifically, despite the quality issues described in the Predictive Models section above, the champion model (i.e., Model 3) was able to cluster the documents together based on their respective policy topics as expected and achieve a clustering performance result well above the threshold for acceptability as per KPI 2. Furthermore, the new webscraping functionality was able to extract a far higher percentage of the policy document web entries than the existing one and far exceeded the threshold for acceptability defined for KPI 1. (Please refer to the Review of Success section below for further details). These results thus demonstrate that the project team’s technical approach resulted in the development of a fully functional document clustering application that has the potential to operate in production once the clustering quality has been further refined. (Please see the Recommendations for Future Analysis section for the detailed recommendations.)

### Findings

Based on the project’s results, it would appear that neither a k-means nor a hierarchical clustering model alone would suffice to ensure the creation of a production-grade document clustering application. Even when new features engineered through the use of TF-IDF and cosine similarity measures were included in the clustering models, the models still could not cluster the documents with an acceptable level of quality. Thus, one would have to include that the use of these NLP techniques would not be apt for this use case despite the fact that they had been recommended by a multitude of professional data scientists for other document clustering problems as explained in the Analysis Justification section above. On the other hand, the development of a LDA model to generate a document-topic matrix for inclusion as a feature in the k-means clustering model resulted in a clustering performance that was not only far superior to that of either of the two models created using combinations of the aforementioned clustering and NLP techniques but also clearly exceeded the minimum threshold for this metric defined in KPI 2. The table below compares the three models created by the project team along with the results they produced.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Techniques** | **New Features** | **Silhouette Score** | **High Clustering Accuracy?\*** |
| Model 1 | K-Means Clustering. TF-IDF Measures | TF-IDF Matrix | 0.11 | No |
| Model 2 | Hierarchical Clustering, TF-IDF Measures, Cosine Similarity Measures | TF-IDF Matrix, Document Similarity Matrix | 0.12 | No |
| Model 3 | K-Means Clustering, Bag of Words, LDA | Bag-of-Words Matrix, Document-Topic Matrix | 0.63 | No |

\*Based on a manual inspection of randomly selected clusters

Thus, the use of the LDA technique combined with the k-means clustering technique shows promise when it comes to creating a document clustering application capable of satisfying the business objectives. Assuming that further refinements are made to the model, it could potential serve as the basis for an application that delivers long-term value to the project’s stakeholders.

With respect to the selection of a platform on which to run the application, the testing efforts demonstrated that the use of a desktop client such as the Python IDLE would not pose any significant performance issues. Once the models’ parameters had been adjusted as necessary, the application runtime lasted no longer than three minutes when any of the models were tested. This indicated that, contrary to the initial expectations, the use of a desktop client as the underlying platform should not hinder the application’s performance based on the current documentation volume. However, as the number of documents posted to the NODIS Library grows over time, it might become necessary to use a cloud-hosted platform such as Apache Spark as was originally recommended by the project team. To that end, the team will work on incorporating the application into a Jupyter notebook to ensure it can be run in the cloud in case it becomes necessary to do so.

### Review of Success

As discussed in the Business Success Criteria section, the project’s success would be determined based on whether it resulted in the delivery of a complete and functional analytical pipeline capable of clustering the NASA policy documents, both the ones that currently reside in the NODIS Library as well as those that will be posted there in the future. In this regard, the project has been largely successful. The document clustering application developed by the project has improved on the current webscraping functionality by extracting a much higher percentage of the policy document web entries from the NODIS Library and placing them into a corpus that can be processed by the clustering model. Whereas the current webscraping functionality can scrape just under 50 percent of the policy document web entries from the NODIS Library, the new application was able to scrape 95 percent of them. This is substantially higher than the minimum threshold of 60 percent defined in KPI 1, meaning that the project achieved a resounding success in terms of improving upon the current webscraping capability.

The document clustering application developed by the project has proven capable of separating the policy documents into the appropriate clusters based on their respective topics. As confirmed during testing, the application is able to consistently cluster documents with identical or similar topics together. With a mean silhouette coefficient of 0.63, the application has clearly exceeded the minimum threshold defined in KPI 2 to determine whether the model has generated a reasonable structure for the text data through its clusters. Based on the KPI, the project can be perceived as successful in this respect. Unfortunately, the qualitative analysis of the application’s clustering performance, involving examinations of the underlying model’s textual and graphical outputs, did not demonstrate that the clusters generated by the model were well-separated from one another. This means the application likely is unable to cluster the documents together by topic with a sufficiently high degree of reliability. In light of this fact, the likelihood that the application will cause more problems for the project stakeholders than it resolves is too high for it to be recommended for deployment to production. Therefore, the project cannot be deemed an unequivocal success when it comes to meeting the business objectives.

### Recommendations for Future Analysis

To determine whether the document clustering application produced by this project can be improved upon, the following analyses are recommended for a future project:

1. **Attempt to improve the quality of the topic-term matrix developed through the use of the LDA technique for Model 3.** While it is possible to identify the terms and their respective weights within each topic, the output contains inordinate numbers of numbers, HTML tags, and other characters that had been removed from the corpus during the preprocessing phase. The modeling outputs show a clear disconnect between the features that were incorporated into the k-means clustering model through the document-topic matrix and those that were included in the topic-term matrix. Unfortunately, the project team was unable to resolve this issue before the project deadline due to time constraints. The resolution of this issue could potentially enable a future project team to identify a more accurate set of topics and adjust the model’s parameters accordingly.
2. **Develop a hierarchical clustering model that includes a document-topic matrix created using the LDA technique.** This model’s outputs should then be compared with those of Model 3 to determine whether the hierarchical or k-means clustering method would result in a superior clustering performance when combined with the LDA technique. This could hence produce insights that lead to an improvement upon the new document clustering application.
3. **Evaluate options to use deep learning to engineer new features for the clustering model.** Although the traditional text mining techniques such as the TF-IDF and bag words methods can be effective for extracting new features from text data, they are unable to retain crucial information about the text such as the semantics, structure, sequence, and context. In addition, the use of these techniques results in the creation of sparse term vectors that could lead to the creation of models that demonstrate poor quality or overfit the text data if the vectors are not refined as necessary to reduce its sparsity (Sarkar, 2018-b). Thus, it would make sense to explore more advanced alternatives for feature engineering that do not suffer from these shortcomings. One possible alternative technique would be Word2vec, which was created by Google to automatically detect the semantical meaning of a given text. This would help improve the model’s ability to accurately cluster future documents containing terms not encountered during its development (Ameisen, 2018; McCormick, 2016). Hence, the use of Word2vec or other deep learning models could lead to the development of a more robust document clustering application.

# References

Ameisen, E. (2018). How to solve 90% of NLP problems: a step-by-step guide. Retrieved from https://blog.insightdatascience.com/how-to-solve-90-of-nlp-problems-a-step-by-step-guide-fda605278e4e

Becker, N. (2016). Clustering US Laws using TF-IDF and K-Means. Retrieved from https://beckernick.github.io/law-clustering/

Boehmke, B. (n.d.-a). Hierarchical Cluster Analysis. Retrieved from https://uc-r.github.io/hc\_clustering

Boehmke, B. (n.d.-b). K-means Cluster Analysis. Retrieved from https://uc-r.github.io/kmeans\_clustering

Combs, L. & Roman, P. (2017). Beyond the Word Cloud. Visualizing Text with Python. Retrieved from https://medium.com/district-data-labs/beyond-the-word-cloud-428e3c25b59c

David, J.E. (2015). NASA's Secret Relationships with U.S. Defense and Intelligence Agencies. Retrieved from https://nsarchive2.gwu.edu/NSAEBB/NSAEBB509/

Gall, R. (2018). Why is Python so good for AI and Machine Learning? 5 Python Experts Explain. Retrieved from https://hub.packtpub.com/python-machine-learning-expert-interviews/

Grando, U. (2017). Python: Count Frequencies with NLTK. Retrieved from https://inzaniak.github.io/pybistuffblog/posts/2017/04/20/python-count-frequencies-with-nltk.html

Hastie, H., James, G., Tibshirani, R., Witten, D. (2013). An Introduction to Statistical Learning. Retrieved from http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Seventh%20Printing.pdf

Hees, J. (2015). SciPy Hierarchical Clustering and Dendrogram Tutorial. Retrieved from https://joernhees.de/blog/2015/08/26/scipy-hierarchical-clustering-and-dendrogram-tutorial/

Kassambara, A. (2017). Determining The Optimal Number Of Clusters: 3 Must Know Methods. Retrieved from http://www.sthda.com/english/articles/29-cluster-validation-essentials/96-determining-the-optimal-number-of-clusters-3-must-know-methods

Kumar, A. (2017-a). Topic Modelling (Part 2): Discovering Topics from Articles with Latent Dirichlet Allocation. Retrieved from https://appliedmachinelearning.wordpress.com/2017/09/28/topic-modelling-part-2-discovering-topics-from-articles-with-latent-dirichlet-allocation/

Kumar, A. (2017-b). Topic Modelling (Part 3): Document Clustering, Exploration & Theme Extraction from SimpleWiki Articles. Retrieved from https://appliedmachinelearning.wordpress.com/2017/10/13/topic-modelling-part-3-document-clustering-exploration-theme-extraction-from-simplewiki-articles/

Kadam, S. (n.d.). Generating Word Cloud in Python. Retrieved from https://www.geeksforgeeks.org/generating-word-cloud-python/

Ladd, J., Otis, J., Warren, C.N., & Weingart, S. (2017). Exploring and Analyzing Network Data with Python. Retrieved from https://programminghistorian.org/en/lessons/exploring-and-analyzing-network-data-with-python

Manning, C.D., Raghavan, P., & Schütze, H. (2008). Hierarchical agglomerative clustering. Retrieved from https://nlp.stanford.edu/IR-book/html/htmledition/hierarchical-agglomerative-clustering-1.html

Mayo, M. (2018). Text Data Preprocessing: A Walkthrough in Python. Retrieved from https://www.kdnuggets.com/2018/03/text-data-preprocessing-walkthrough-python.html

McCormick, C. (2016). Google's trained Word2Vec model in Python. Retrieved from http://mccormickml.com/2016/04/12/googles-pretrained-word2vec-model-in-python/

National Aeronautics and Space Administration. (2018). NASA Strategic Plan 2018. Retrieved from https://www.nasa.gov/sites/default/files/atoms/files/nasa\_2018\_strategic\_plan.pdf

NSS. (2017). The Essential NLP Guide for data scientists (with codes for top 10 common NLP tasks). Retrieved from https://www.analyticsvidhya.com/blog/2017/10/essential-nlp-guide-data-scientists-top-10-nlp-tasks/

Oleson, G., Silsby, B., & Skelly, D. (2012). NASA is essential for national security. Retrieved from http://www.thespacereview.com/article/2210/1

Piatetsky, G. (2017). Emerging Ecosystem: Data Science and Machine Learning Software, Analyzed. Retrieved from https://www.kdnuggets.com/2017/06/ecosystem-data-science-machine-learning-software.html

Pointer, I. (2018). The 5 best programming languages for AI development. Retrieved from https://www.infoworld.com/article/3186599/artificial-intelligence/the-5-best-programming-languages-for-ai-development.html

Sarkar, D. (2018-a). Understanding Feature Engineering (Part 3) — Traditional Methods for Text Data. Retrieved from https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

Sarkar, D. (2018-b). Understanding Feature Engineering (Part 4) — A hands-on intuitive approach to Deep Learning Methods for Text Data — Word2Vec, GloVe and FastText. Retrieved from https://towardsdatascience.com/understanding-feature-engineering-part-4-deep-learning-methods-for-text-data-96c44370bbfa

scikit-learn developers. (2017-a). Silhouette Coefficient. Retrieved from http://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient

scikit-learn developers. (2017-b). Topic extraction with Non-negative Matrix Factorization and Latent Dirichlet Allocation. Retrieved from http://scikit-learn.org/stable/auto\_examples/applications/plot\_topics\_extraction\_with\_nmf\_lda.html#sphx-glr-auto-examples-applications-plot-topics-extraction-with-nmf-lda-py

Scott, J.A. (2015.) Getting Started with Apache Spark: From Inception to Production. Retrieved from http://info.mapr.com/rs/mapr/images/Getting\_Started\_With\_Apache\_Spark.pdf

Spann, T. (2018). Scala vs. Python for Apache Spark. Retrieved from https://dzone.com/articles/scala-vs-python-for-apache-spark

Spector, P. (n.d.). Cluster Analysis. Retrieved from https://www.stat.berkeley.edu/~s133/Cluster2a.html

Wilder-James, E. (2017). Understanding AI Toolkits. Retrieved from https://www.svds.com/understanding-ai-toolkits/