INSTITUTE FOR DEFENSE ANALYSES

Evaluating and Predicting Contract Performance Using Machine Learning: A Feasibility Study

Gregory A. Davis, Project Leader
Travis L. DePriest
Brian G. Gladstone
Laura A. Hildreth
Miranda G. Seitz-McLeese

September 2020
Approved for public release; distribution is unlimited.
IDA Document D-14275
H 20-000263

INSTITUTE FOR DEFENSE ANALYSES
4850 Mark Center Drive
Alexandria, Virginia 22311-1882

The Institute for Defense Analyses is a nonprofit corporation that operates three Federally Funded Research and Development Centers. Its mission is to answer the most challenging U.S. security and science policy questions with objective analysis, leveraging extraordinary scientific, technical, and analytic expertise.

About This Publication

This work was conducted by the IDA Systems and Analyses Center under contract HQ0034-19-D-0001, Project AY-7-4626, "Machine-Assisted High-Volume Contract Evaluation Feasibility Study," for the Office of the Under Secretary of Defense (Acquisition and Sustainment/Acquisition Analytics & Policy). The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

Acknowledgments

This project came to IDA's Cost Analysis and Research Division because of our expertise in acquisition, but the project also required expertise in text analytics, which resides in the Information Technology and Systems Division. We are grateful to Laura A. Odell who helped us find two great teammates. This document piloted a new process for version control using Bitbucket, and we thank R. Abraham Holland for inventing the workflow and helping us through it. We also thank Michael S. Nash, Brian Q. Rieksts, and Nicholas A. Wagner for performing a technical review of this document.

For More Information

Gregory A. Davis, Project Leader
gdavis@ida.org, (703) 575-4698

David E. Hunter, Director, Cost Analysis and Research Division
dhunter@ida.org, (703) 575-4686

Copyright Notice

Rigorous Analysis | Trusted Expertise | Service to the Nation

INSTITUTE FOR DEFENSE ANALYSES

IDA Document D-14275

Evaluating and Predicting Contract Performance Using Machine Learning: A Feasibility Study

Gregory A. Davis, Project Leader
Travis L. DePriest
Brian G. Gladstone
Laura A. Hildreth
Miranda G. Seitz-McLeese

Executive Summary

The Office of Acquisition Analytics and Policy, which is part of the Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)), tasked IDA to assess the feasibility of using machine leaning to analyze contracts for major defense acquisition programs (MDAPs). The goal of the analysis was to extract data from contracts and predict program performance. The study was divided into three stages: crawling, walking, and running.

Crawling

The crawling stage consisted of building a dataset. During this stage of the analysis, contracts were collected and processed.[1] The contracts chosen were listed in Selected Acquisition Reports (SARs) between December 1997 and December 2018, and were from MDAPs that were no longer reporting as of November 2019. Examining contracts from this period ensured that each program was more than 90 percent complete. Additionally, the dataset was limited to this period so that program performance outcomes were known which is necessary when using machine learning algorithms for predictive purposes. We collected 24,364 contract files in PDF format spanning 149 contract numbers and 34 MDAPs. (The MDAPs and their associated contract numbers are in Appendix B.) Finally, we used the Institute for Defense Analyses Text Analytics (IDATA) capability to turn the collected files into a machine-readable dataset.

Walking

In the walking stage, contract data were evaluated by training machine learning algorithms on our data to answer relatively simple questions. This activity ensured that the dataset was of reasonable quality, the machine-learning algorithms were functioning properly, and reasonable answers were produced. During this stage, word clouds were generated for each program. The following figures show the word clouds for two programs, CH-47F and ATACMS-APAM, respectively.

[1] Data were obtained from Electronic Data Access (EDA), which is part of the Procurement Integrated Enterprise Environment (PIEE), and included most Army MDAP contracts. We found that including contracts from the other Services in our dataset required more resources and time than our feasibility study allowed.

Word Cloud for the CH-47F Program

Word Cloud for the ATACMS-APAM Program

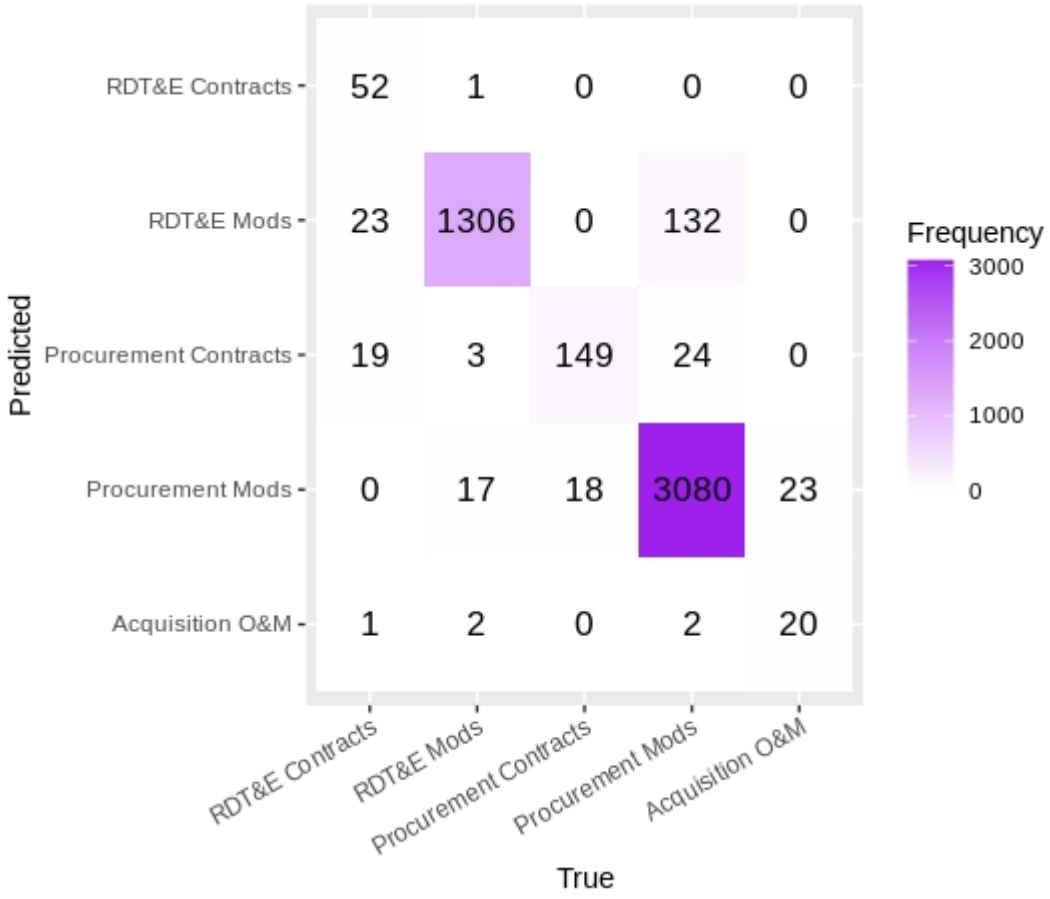
The word clouds show the frequency of various words in each program's contracts, with common words like "and" and "the" removed. These graphics were useful for revealing the most frequently used words in the documents for each program and helped ensure that we collected reasonable data.

Next, a naïve Bayes classifier was trained on a training set from these data and asked to differentiate each contract by one of five categories: (1) Research, Development, Test, and Evaluation (RDT&E); (2) RDT&E Modifications; (3) Procurement; (4) Procurement Modifications; and (5) Acquisition Operations and Maintenance (O&M) Modifications. These categories could be readily determined by humans, and it was expected that machine learning would produce the same results. The distribution of contract categories and number of contract types in each are shown in the following table.

Number of Documents by Type

Document Type	Number of Documents
RDT&E	472
RDT&E Modifications	6,646
Procurement	835
Procurement Modifications	16,191
Acquisition O&M Modifications	214
Total	24,358

The algorithm was trained on 80 percent of the files and then used to predict the category of the remaining 20 percent. The following confusion matrix shows how well the algorithm predicted contract type.



Confusion Matrix for File Type Identification

The diagonal elements, which are the largest numbers, show where the algorithm correctly identified the contract type in the test sample. Overall, the algorithm correctly classified 4,607 of the 4872 files, leading to an overall accuracy rate of 94.6 percent. The accuracy rate depends on sample size. For example, the algorithm predicted that 52 of the files in the test data were RDT&E contracts while 95 of the files were RDT&E contracts leading to an accuracy rate of nearly 55 percent while 3080 of the 3238, or just over 95 percent, of the procurement mods were correctly classified.

Other models, described later, indicated that contracts were successfully transformed into data. Consequently, this indicated it was possible to apply our algorithms to this dataset to ask relatively simple questions and obtain logical answers.

Running

In the running stage, more difficult forecasting questions were asked to test the ability of machine-learning algorithms to use contract datasets to predict program performance. We used Q-ratio as a metric for quantity growth, quantity-adjusted program acquisition unit cost as a metric for cost growth, and program end date. Support vector machine (SVM) models were trained using 70 percent of the programs and the performance metrics were predicted for the remaining 30 percent of programs. The SVMs were unable to predict performance any better than random guessing. We also examined the use of clustering to identify similar programs. Though we could identify similar programs, it was difficult to identify why the programs were similar indicating that more research is needed in this area.

Conclusions

We found that text analytics and machine-learning algorithms were well suited for extracting information from contracts and converting this information into a structured dataset. Although our analyses, which used several different metrics, indicated that the extracted data were useful for descriptive purposes, we were unable to ascertain whether machine-learning algorithms could predict program performance. This result, however, does not mean that forecasting program performance with a contract dataset is unfeasible. It may mean that a more complete (or different) set of contracts, other performance metrics, or alternative algorithms would improve predictive outcomes. Furthermore, to improve forecasting, it may be necessary to combine contract data with data from other sources.

Contents

1. Introduction

Because government contracts can be dense and voluminous, reading them can be a time-consuming and laborious process. Despite the effort required to read them, contracts constitute a large body of regularly formatted text that contains potentially useful information that increases during an acquisition program's life cycle. With current computation capabilities, these lengthy documents can be processed rapidly; thus, it is of interest to examine how the information from contracts can be used. To that end, the Office of Acquisition Analytics and Policy, which is part of the Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)), tasked the Institute for Defense Analyses (IDA) to assess the feasibility of applying text analytics and machine learning to contracts and modifications for major defense acquisition programs (MDAPs). The goals of the study are to:

- Review the literature on the use of text analytics to evaluate contracts
- Extract useful data from contracts and construct a usable dataset from DoD acquisition contracts and modifications
- Assess the feasibility of using machine learning to understand and predict the performance of MDAPs

A. Construction of Our Study

We divide this feasibility study into three stages that we describe as crawling, walking, and running. The crawling stage consists of using text mining to convert the information from MDAP contracts and modifications into a structured dataset. Next, the walking stage entails constructing basic descriptive measures and conducting basic analyses on the dataset created in the crawling stage. The goal of the walking stage is to demonstrate the viability of using the data in more complicated analyses. Finally, the running stage involves more complicated analyses to assess whether the information from contracts and modifications can be used to evaluate MDAP performance.

B. Data: Major Defense Acquisition Programs and Their Contracts

The term *Major Defense Acquisition Program* is defined formally in section 2430 of Title 10 of the United States Code. For a program to be considered an MDAP, it must meet high thresholds for development or procurement cost. For example, new aircraft (and sometimes their modifications), ships, combat vehicles, munitions, tactical vehicles, and

electronic systems can all be classified as MDAPs.[1] MDAPs have specific reporting requirements, including releasing a Selected Acquisition Report (SAR) at least once per year. Each SAR lists the active large contracts in the program, and we used this listing to find the MDAP contracts for our dataset.

In addition to reporting requirements, large government contracts differ from most contracts because they are modified routinely. These modifications can arise from funding or other program changes, including new requirements, new rules, new opportunities, contractor performance, and several other causes.

Over time, and often over several years, these large contracts transform from single documents to long strings of documents that are repeatedly amended. The purpose of this project is to explore whether this constantly growing set of documents can provide a useful dataset that, when combined with machine learning and text analytics, can be used to predict MDAP performance.

C. Literature Review of Text Analytics and Their Use with Acquisition Contracts

Due to frequent cost growth and schedule breaches, ways to improve cost and schedule estimates for MDAPs have gained considerable attention. In these analyses, researchers first identify a set of programs of interest (e.g., by commodity type, by military department, and so on). Next, the researchers collect data on the programs. Data collection can take several forms depending on the goal of the analysis and the programs of interest.

When evaluating MDAPs, a common data source is selected acquisition reports (SARs). SARs have been used frequently in previous studies such as the DoD's *Performance of the Defense Acquisition System* annual reports;[2] the Government Accountability Office's annual reports on *Defense Acquisitions: Assessments of Selected Weapon Programs*;[3] and a number of other reports (for example, McCormick, Hunter, and

[1] Other programs can be designated as MDAPs by Congress or by DoD if they are of interest for other reasons.

[2] "Performance of the Defense Acquisition System," Department of Defense, 2013–2016, https://libguides.nps.edu/acqprog/acqreports.

[3] "Defense Acquisitions: Assessments of Selected Weapons Programs," Government Accountability Office, 2011–2018, https://libguides.nps.edu/acqprog/acqreports.

Sanders, 2018;[4] McNicol et al., 2016; [5] McNicol and Wu, 2014[6]). Previous studies used a variety of statistical techniques and methods, and several used descriptive displays and regression analysis.

Although these studies have proven useful for identifying potential causes of cost growth, schedule growth, and other outcomes, they have also caused concern among some researchers. One concern is that these studies use easily quantifiable program characteristics, such as the percent of the budget spent prior to a specific milestone, whether the program is a joint program, or what type of weapon systems was analyzed. This approach is problematic because it limits the information used in the studies. Potential evidence of this are the low R-squared values (often 0.10 or lower) observed in the studies using regression analysis, indicating that a small proportion of variability in outcomes of interest are explained by the included factors.[7] Incorporating other information, such as the qualitative information found in contracts and similar documents, could lead to models that better predict cost and schedule growth.

Another concern is the prevalent use of SARs data. As explained in a 2006 RAND report,[8] using these data presents several drawbacks, including that SARs are available only for certain programs, typically MDAPs that are not highly classified. Further, SARs data are reported at a highly aggregated level, changes are often not well documented, and reported values lack clarity and consistency. For these reasons, other data sources, such as the original contracts, should be used. Other data sources used in previous studies include historical memoranda; summaries of program data from published sources; government documents, including contracts from the Federal Procurement Data System[9] and DoD

---

4 Rhys McCormick, Andrew Hunter, and Gregory Sanders, "Preliminary Findings: Is the Ratio of Investment between R&D to Production Experiencing Fundamental Change?" Naval Postgraduate School, Monterey, California, 2018, https://calhoun.nps.edu/handle/10945/58752.

5 David L. McNicol, David M. Tate, Sarah K. Burns, and Linda Wu, "Further Evidence on the Effect of Acquisition Policy and Process on Cost Growth of Major Defense Acquisition Programs," IDA Paper P-5330-REVISED, (Alexandria, VA: Institute for Defense Analyses, 2016).

6 David L. McNicol, and Linda Wu, "Evidence on the Effect of DoD Acquisition Policy and Process on Cost Growth of Major Defense Acquisition Programs," IDA Paper P-5126, (Alexandria, VA: Institute for Defense Analyses, 2014).

7 Thomas Light, Robert S. Leonard, Julia Pollak, Meagan L. Smith, and Akilah Wallace, "Quantifying Cost and Schedule Uncertainty for Major Defense Acquisition Programs (MDAPs)," Rand Corporation, 2017, https://www.rand.org/pubs/research_reports/RR1723.html.

8 Obaid Younossi, Lionel A. Galway, Bernard Fox, and John C. Graser, "Impossible Certainty: Cost Risk Analysis for Air Force Systems," Vol. 415. Rand Corporation, 2006, https://www.rand.org/pubs/monographs/MG415.html.

9 The Federal Procurement Data System, https://www.fpds.gov/fpdsng_cms/index.php/en/. From this website, users can download contract data reports that are publicly available.

Defense Budget documents;[10] and field research (see, for example, Berteau et al., 2010;[11] Lorell, Payne, and Mehta 2017;[12] Tyson, Harmon, and Utech, 1994[13]). These more varied data sources allow researchers to conduct analyses that may provide greater insights into programs, such as root cause analyses and case studies.

Although using additional data is beneficial, collecting it from contracts and modifications is time-consuming and prone to human error. An alternative approach is the use of machine-learning techniques, specifically text mining and analytics, to extract data from contracts, modifications, and other documents. Text mining allows researchers to take unstructured text, such as that found in contracts, and extract useful information that can be used in further analyses. Compared to studies solely using SARs data to evaluate defense acquisition, the number of studies incorporating data obtained via text mining is small. An overview of text analytics, including the techniques we use in this study, is found in Appendix A.

Of the studies using text analytics, several use text mining on the format 5 portions of contractor performance reports (CPRs) from the Defense Cost and Resource Center. In these portions of CPRs, contractors provide a written statement explaining cost and schedule variances and other contract issues. Based on the information gleaned from these write-ups, Freeman (2013)[14] used a naïve Bayes classifier to identify high-risk programs, defined as those at risk of a 6-month cumulative change in estimate at complete (EAC) greater than 5 percent. Additionally, Miller (2012)[15] conducted several regression analyses to predict changes in EAC at 4-, 5-, and 6-month milestones. The thesis of McGowin

[10] DoD budget requests for fiscal years 1998–2021 are available from the Under Secretary of Defense (Comptroller) at https://comptroller.defense.gov/Budget-Materials/.

[11] David Berteau, Joachim Hofbauer, Gregory Sanders, and Guy Ben-Ari, "Cost and Time Overruns in Major Defense Acquisition Programs," Center for Strategic and International Studies, 2010, https://www.csis.org/analysis/cost-and-time-overruns-major-defense-acquisition-programs-2011-0.

[12] Mark A. Lorell, Leslie Adrienne Payne, and Karishma R. Mehta, "Program Characteristics That Contribute to Cost Growth: A Comparison of Air Force Major Defense Acquisition Programs," Rand Corporation, 2017, https://www.rand.org/pubs/research_reports/RR1761.html.

[13] Karen W. Tyson, Bruce R. Harmon, and Daniel M. Utech, "Understanding Cost and Schedule Growth in Acquisition Programs," IDA Paper P-2967, (Alexandria, VA: Institute for Defense Analyses, 1994).

[14] Charlton E. Freeman, "Multivariate and Naïve Bayes Text Classification Approach to Cost Growth Risk in Department of Defense Acquisition Programs," PhD diss., Air Force Institute of Technology, 2013, https://apps.dtic.mil/dtic/tr/fulltext/u2/a583708.pdf.

[15] Trevor P. Miller, "Acquisition Program Problem Detection Using Text Mining Methods," PhD diss., Air Force Institute of Technology, 2012, https://apps.dtic.mil/dtic/tr/fulltext/u2/a557568.pdf.

(2018)[16] and the related paper by Ritschel, Fass, and Boehmke (2018)[17] also used text mining to identify trends in 5 major legislative reforms for acquisition and in a compendium of documents containing the views of experts on the legislative acts. This research compared the trends in the data sources and found that the trends differed, indicating that reforms did not address the concerns raised by experts. Brown (2017)[18] used text analytics to extract word frequencies from the annual National Defense Acquisition Act (NDAA) from 1998 through 2017 to show an increase in the use of cost-estimating terms. Lastly, Algarín (2016)[19] calculated word frequencies of 3 subcategories of terminology related to human systems integration (HSI) in documents for 16 MDAPs from the Defense Acquisition Management Information Retrieval database and the Acquisition Decision Memoranda website. The word frequencies of the 3 HSI subcategories were used as independent variables in logistic regression models to predict whether a program would experience a schedule breach or cost overrun.

The text analytics techniques (such as word frequencies) in these studies were relatively simple, as were the statistical analyses using the text data. However, these studies showed the potential utility of pursuing text analytic approaches for defense acquisition documents because more information can be incorporated into analyses. Given this preliminary evidence, researchers should investigate more sophisticated text analytics methods as well as how to use the data obtained via text mining.

---

16 Amanda L. McGowin, "An Analysis of Major Acquisition Reforms Through Text Mining and Grounded Theory Design," PhD diss., Air Force Institute of Technology, 2018, https://apps.dtic.mil/dtic/tr/fulltext/u2/1056519.pdf.

17 Jonathan D. Ritschel, Robert D. Fass, and Bradley C. Boehmke, "A Text Mining Analysis of Acquisition Reforms and Expert Views," *Defense AR Journal* 25 (3): 288–323, 2018.

18 G. E. Brown, "Measuring the Increasing Relevance of Cost Estimating Through Text Analytics," *ICEAA World* 2: 32–33, 2017.

19 Liana Algarín, "Human Systems Engineering and Program Success—A Retrospective Content Analysis," *Defense Acquisition Research Journal* 23 (1): 78–101, 2016.

2. Crawling: From Contracts to Data

Our study required taking a collection of contracts and their modifications and turning them into a machine-readable dataset. To accomplish this goal, we used the DoD Electronic Data Access (EDA) system and the IDATA capability. EDA allowed us to collect the contracts and modifications, while IDATA allowed us to convert the PDF versions of the documents into a machine-readable dataset.

A. EDA

EDA is a web-based system that allows users to access and retrieve a variety of documents, including contracts and modifications, and is part of the DoD's Procurement Integrated Enterprise Environment (PIEE). We received access to EDA and were allowed to use its Bulk Download tool to transfer contracts and modifications for Army MDAPs. This tool allows users to transfer multiple files at once, such as all files associated with a given contract number, and can be searched by contract number, Commercial and Government Entity (CAGE) code, or contract number and Department of Defense Activity Address Code (DoDAAC). We used contract numbers to ensure that we downloaded complete contract documents.

EDA returns a single file for each contract or contract modification under a specified number. For example, inputting the procurement contract number for the Stryker program, W56HZV07DM112, returns 2,154 files, of which 2,083 were modifications and the remaining 71 were either the initial contract or new purchase orders. In total, we obtained contracts and modifications for 149 contract numbers. The contract numbers, the programs with which the numbers are associated, and the number of files are found in Appendix B.

B. Institute for Defense Analyses Text Analytics (IDATA)

IDATA is IDA's in-house suite of capabilities specifically designed to facilitate discovery and extraction of information from government documents. IDATA facilitates text analysis in three phases: preprocessing and data acquisition, application of algorithms for data analysis, and data visualization. IDATA consists of a variety of modules to support this broad collection of use cases. A detailed discussion of all modules is outside the scope

of this report, but more information about other modules and IDATA in general is found in the October 2018 issue of *IDA Research Notes*.[20]

For this project, we used IDATA primarily for two tasks. First, we converted the documents from PDF format into plain text files. Nearly all documents were digital files, so converting them was relatively easy. For files that were not originally in digital format, such as scans of physical documents, IDATA performed optical character recognition. Second, after all documents were in plain text format, IDATA used extractors to "pull" particular pieces of data from documents and tag the documents with that information. In this project, documents were tagged with dates, CAGE codes, and dollar amounts, among other information. These tags could then be placed in a spreadsheet, thereby turning the unstructured data into a structured dataset.

C. Our Collection of Contracts and Its Associated Data Set

For this analysis, we used EDA to download files for Army programs no longer issuing SARs to the Congress as of November 2019. These programs should have spent at least 90 percent of the total funds they were expected to spend in their acquisition phases. As a result, many programs had completed their planned buys or had been cancelled. We restricted our study to completed MDAPs because we needed to know the outcomes of these programs for our analyses (e.g., quantity purchased, amount spent, whether the program was canceled). In total we downloaded 24,364 files, which spanned 149 contract numbers and 34 MDAPs. The MDAPs and their associated contract numbers are in Appendix B.

Using IDATA, we then extracted the text from each file as well as relevant information, such as CAGE codes and projected end date, that may be useful in later analyses.[21]

---

[20] Michelle Albert and Arun Maiya, "IDATA Overview," IDA Research Notes, (Alexandria, VA: Institute for Defense Analyses, 2018).

[21] In our file-level dataset, each row (24,364 in total) represents a file, and the columns contain relevant information extracted from the file as well as text information obtained using a bag-of-words model. For several of the subsequent analyses, we calculated *term frequency-inverse document frequency* (tf-idf) values, which we used in an LSA. (More details on tf-idf are in Appendix A.) The extracted LSA components or features were then used in several of our analyses. The contract-level dataset contained 149 rows, with each row representing a contract. The files associated with each contract were combined into one file with the columns containing relevant information, similar to the file-level dataset. The program- or MDAP-level dataset contained 34 rows, one for each MDAP. All files for a given program were combined into one file. As with the file-level and contract-level datasets, the columns contained information regarding the 34 MDAPs. We used the file-level dataset in the walking stage of our analysis and the program-level and contract-level datasets in the running stage.

3. Walking: Validating the Dataset

The purpose of the walking stage is to validate the dataset we created in the crawling stage. The walking stage is necessary for ensuring that the dataset contains the information we intended and for conducting simple analyses to assess whether we can answer questions using these data. If so, investigation into more sophisticated analyses is warranted. Before validating our dataset, we preprocessed these data to make them useful for later analyses.

A. General Description of Data Processing of Model Features

The first step in preprocessing the text from our documents is tokenization, in which text is split into tokens, which are the words in the document. We then normalize the tokens through stemming and lemmatization, convert all text to lowercase, and remove stop words. For each document, we then count the number of times each token occurs in our corpus, resulting in a bag-of-words that can be used in subsequent analyses.

B. Descriptive Analysis: Word Clouds

The first step in validating our dataset is to construct descriptive graphics to ensure that the text is reasonable and to gain further insights on our data. One common graphic is a word cloud where the tokens (words) in a document are written so that the font size is proportional to the frequency of the word. We created word clouds at the MDAP level. That is, for each of the 34 MDAPs, we created a bag-of-words using all documents for that MDAP. We then created a word cloud using word frequency for each MDAP. For brevity, we display the word clouds for the CH-47F and the ATACMS-APAM programs in Figure 1 and Figure 2, respectively.

Figure 1. Word Cloud for the CH-47F Program

Figure 2. Word Cloud for the ATACMS-APAM Program

These word clouds show that for the CH-47F program, the most frequent words or abbreviations are “Saudi,” “trainer,” “brake,” “hoist,” and “bdcbm.” For the ATACMS-APAM program, the most common words are “guidance,” “software,” and “Korea.” Note that these words are lower case due to preprocessing and that the colors are for visual

differentiation only. These results are reasonable and logical for each program. For example, Saudi Arabia operates CH-47Fs and South Korea operates ATACMS-APAMs, so it is not unexpected that these words would appear frequently. Names are also expected to occur frequently because contracts require points of contact.

C. Linear Regression Model to Predict End Date

We next use relatively simple predictive models to validate our dataset further and evaluate whether our data can be used to develop more sophisticated predictive models. For our simple models, we conducted our analysis at the document, or file, level. Conducting analyses at the file level leads to "overly optimistic" evaluations of performance. By treating each file as its own observation, we ignored that files associated with the same contract or program were related to one another. This approach leads to higher R-squared values for regression models and lower error rates for classification models than if we conducted analyses that accounted for the relationships between documents. For this stage, we considered analyses at the file level because poor results from this stage would render more complicated analyses irrelevant.

We used a linear regression model to predict the latest date of a program (a proxy for "end date"). In this analysis each file was an observation. In total, there were 835 files for procurement contracts, 28 of which were excluded because the end dates could not be extracted, leading to a sample size of 807 files. The 807 files were then randomly assigned to one of 5 groups, resulting in approximately 20 percent of the files in each group. We considered these 5 groups to be our "holdout" (or test) dataset and considered the remaining 80 percent of the data as training data. Using the training data, a bag-of-words was created using tf-idf. Then, to reduce the dimensionality of our bag-of-words, we used latent semantic analysis (LSA), which is described earlier. Using an LSA, we extracted features (or patterns) from the text. Note that an LSA was performed using all files so that it identified the most common features of our corpus. For each file, a score for each feature was calculated, and we then used these scores in our predictive models. Using the scikit-learn tool in Python,[22] we performed an LSA and extracted 100 features, which is the recommended number according to the Python user guide.

A linear regression model using the scikit-learn tool in Python[23] with end date as the response and the LSA scores of the first 100 features as explanatory variables. Using the results of the model, we predicted the response (i.e., the latest date) for the test dataset. We then calculated the R-squared value between the predicted latest date and the actual latest

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-Learn: Machine Learning in Python," *Journal of Machine Learning Research* 12: 2825–30, 2011.

[23] Ibid.

date. This process was repeated for the remaining 4 groups. The technical details of this analysis are found in Appendix C.

The five R-squared values of 0.89, 0.89, 0.90, 0.91, and 0.93 indicated that the predicted and actual dates were highly correlated. Figure 3 displays a scatterplot of the actual and predicted latest date for the five test sets combined. This plot visually shows the strong correlation between the two datasets. The R-squared value associated with this plot is 0.91. Although the R-squared values are high, caution must be used when interpreting them because they are not a good measure of model fit in this analysis as we know the observations are correlated. However, to validate our data, these R-squared values indicate that we extracted useful information from the documents.

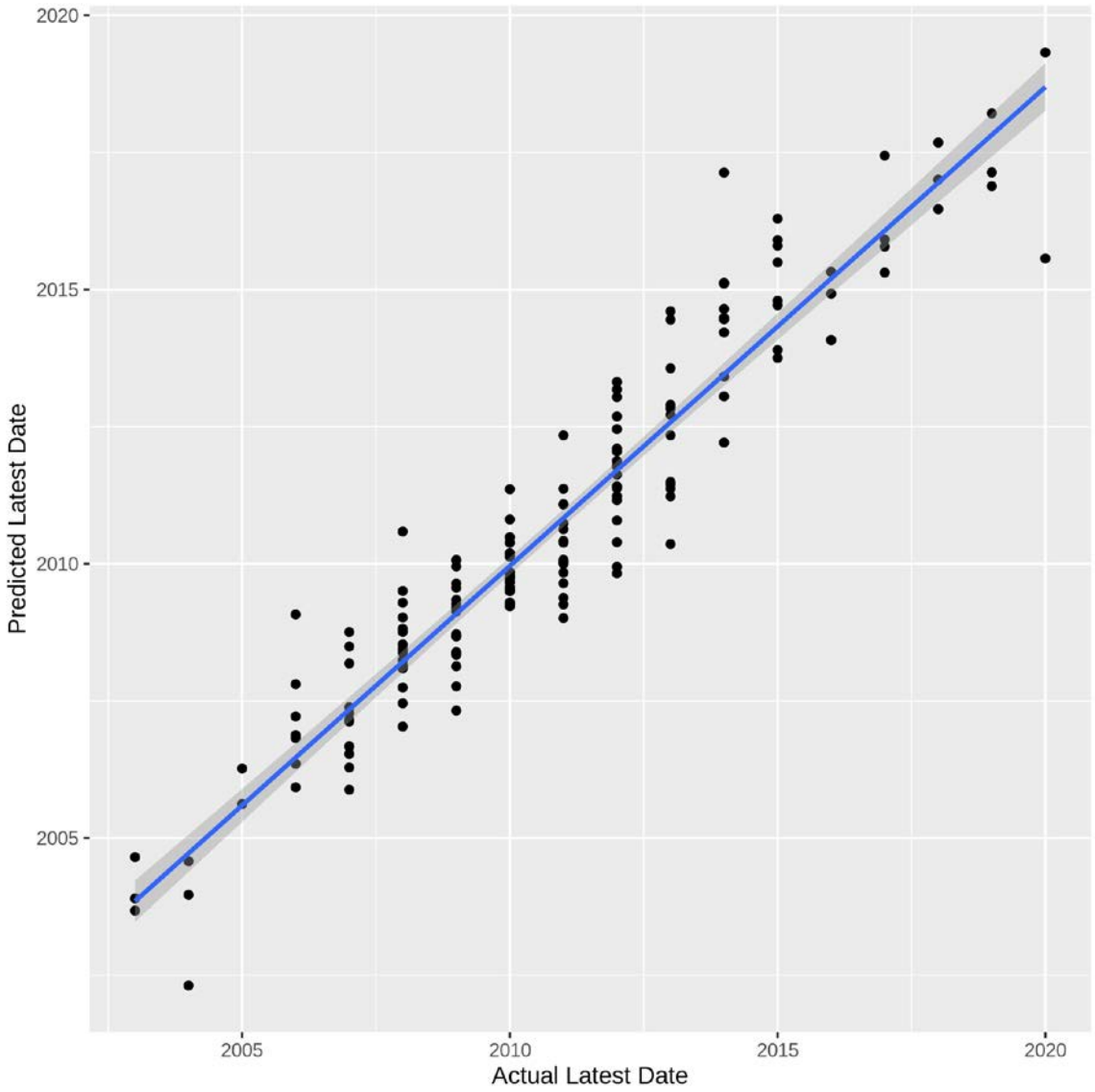


Figure 3. Predicted vs. Actual Latest Date

D. Naïve Bayes Classifier to Predict Document Type

Each document in our analysis is one of five types: RDT&E contracts, RDT&E modifications, procurement contracts, procurement modifications, or acquisition O&M modifications. The next model we consider is a naïve Bayes classifier, which categorizes each document into one of the five types. We conducted this analysis to determine whether our data could be used to classify documents. The dataset consists of 24,358 documents, and the breakdown of the number of documents by type is provided in Table 1.

Table 1. Number of Documents by Type

Document Type	Number of Documents
RDT&E	472
RDT&E Modifications	6,646
Procurement	835
Procurement Modifications	16,191
Acquisition O&M Modifications	214
Total	24,358

To train the naïve Bayes classifier, approximately 80 percent of the files were randomly selected as training data and the remaining 20 percent used as test data. Using the training data, a naïve Bayes classifier was trained with the scores of the first 100 LSA features as explanatory variables and document type as the response. The trained classifier was then used to predict document type for each file of the test data.

Figure 4 displays the confusion matrix associated with our classifier. For example, the test data contain 1,328 RDT&E modifications, of which 1,306 (in yellow), or 98.3 percent, were correctly classified. Of the 23 that were misclassified (in red), 1 was classified as an RDT&E contract, 3 as procurement contracts, 17 as procurement modifications, and 2 as acquisition O&M contracts.

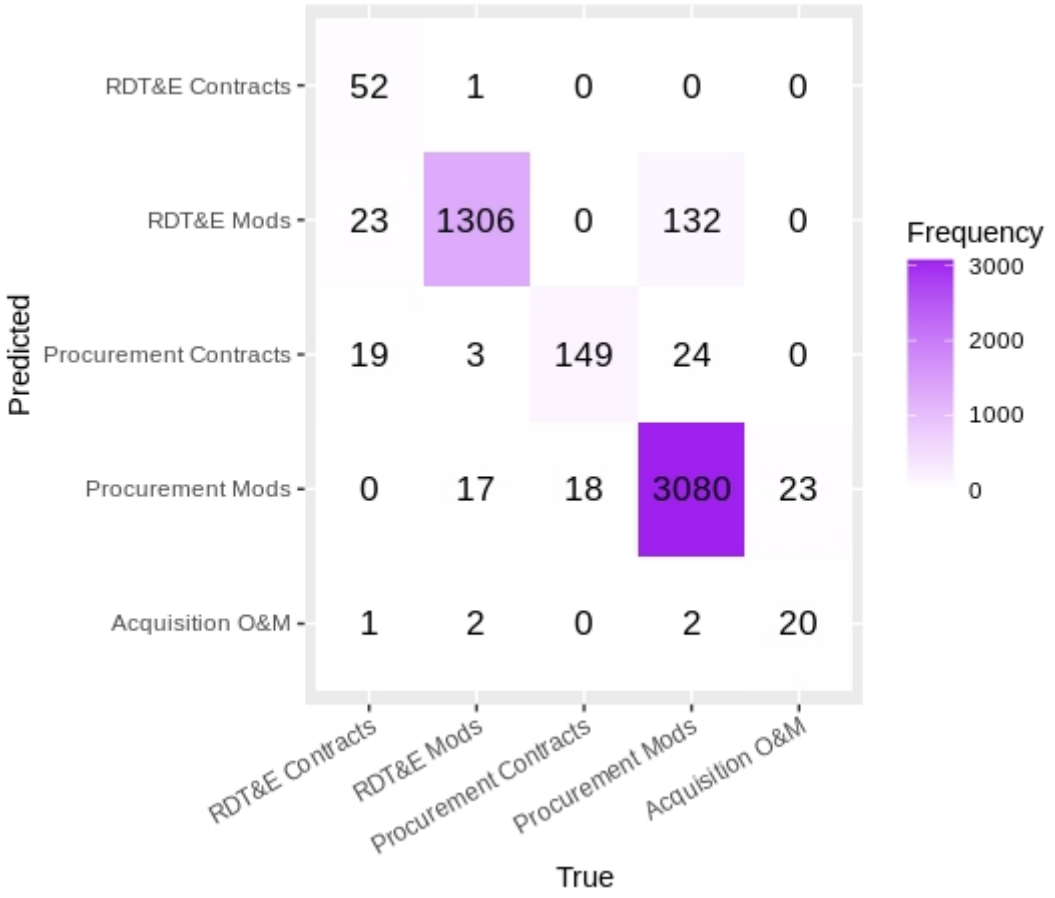


Figure 4. Confusion Matrix for Actual and Predicted Document Type

These results indicate that the naïve Bayes classifier correctly classified documents most of the time. In our analysis, 94.6 percent of documents were correctly classified, which is noticeably higher than the classification rate of 51.79 percent we would expect with random guessing. The classification rate and the performance of the naïve Bayes classifier vary across document type compared to random guessing. Document types with smaller sample sizes show lower classification accuracy. For example, RDT&E contracts were correctly classified 54.7 percent of the time, and acquisition O&M contracts 47.6 percent of the time. For both document classes, these classifications rates were less than the classification accuracy of 96.21 percent and 98.26 percent, respectively, that we would expect under random guessing. In contrast, procurement modifications were correctly classified 95.2 percent of the time compared to a classification accuracy of 55.45 percent under random guessing. With larger sample sizes, the naïve Bayes classifier was better able to "learn" the features of those document types, allowing for higher accuracy. These results indicated that we correctly classified documents and that we extracted useful information from the contracts for use in later analyses.

4. Running: Assessing Programs

In the running stage, we examine whether our dataset can be used to conduct more sophisticated analyses, including advanced modeling techniques and assessment metrics as outcomes in these models. A change in this stage is that our analyses are conducted at the program or MDAP level rather than at the file level as in the walking stage. The walking stage ignores the fact that documents for the same MDAP share information, leading to "overly optimistic" performance evaluations when training and testing models on random file samples. To address this issue, we consider all documents for an MDAP as a time series and use this series in our analyses.

A. Support Vector Machines to Predict Quantity and Cost Growth

To investigate whether we can use our data to predict quantity and cost performance, we use support vector classifiers (SVCs) and support vector regressions (SVRs). These methods use a training set of data to develop algorithms either to predict the class of an observation (SVCs) or to predict a quantitative response (SVRs). Because this analysis is conducted at the program/MDAP level, we randomly selected 70 percent of programs (21 in total) and used their associated files as the training data. We used the files associated with the remaining 30 percent of programs (10 in total) as the test data.

Before conducting our analyses, we first preprocess our training data at the file/document level as in the walking stage. We create a bag-of-words, calculate tf-idf values, use these values to conduct an LSA, and retain the first 50 components. At the program level, this analysis yields a multivariate time series, meaning that we measure multiple variables (in this case, the LSA components) at multiple points over time (i.e., the dates of the documents).

SVCs and SVRs require time series of equal lengths. However, our data include programs of varying durations, from 6 to more than 20 years, necessitating a method that can be applied to unequal time series. To meet this goal, we use the global alignment kernel (GAK) in our SVCs and SVRs. Essentially, a GAK can measure the similarity of the shapes of time series of unequal lengths. A detailed discussion of the algorithm is found in Cuturi (2011).[24] In our final analysis, we used only the first 5 years of data so that the time series of each program was the same length. However, we may not always have the first 5 years

[24] Marco Cuturi, "Fast Global Alignment Kernels," *Proceedings of the 28th International Conference on Machine Learning*, ICML 2011, 929–36.

of information or may want to use time series of varying lengths, and ultimately chose to use a GAK to allow for this flexibility.

We first consider using SVMs to predict a metric of quantity that we call the Q-ratio. The Q-ratio is the quotient of two different quantities: the number of units purchased for a given program divided by the number that was projected at milestone B, when the program formally became an MDAP. If the Q-ratio is 1, then the program executed as planned. Values less than 1 indicate that the program bought fewer units than projected, and values greater than 1 indicate that the program bought more units than projected. This approach answers the oversight question "Will the program ultimately procure more or fewer units than initially estimated?" *before knowing the final outcome*. This metric does not necessarily measure program success because DoD may procure more or fewer units for a variety of reasons. However, because this outcome is known, relatively simple, and quantifiable, it is of interest to examine whether text information in early program documents can predict the program's Q-ratio at the end of the program.

Our analysis uses Q-ratio information on 31 programs. We use an SVC to predict whether the Q-ratio is greater than or equal to 1 and an SVR to predict the final Q-ratio. Neither model could predict program outcomes better than random guessing. In our dataset, 12 of the 31 programs, or 38.7 percent, met or exceeded expectations; the remaining 61.3 percent produced fewer units than initially expected. Under random guessing, we would estimate that a program meets or exceeds expectations 38.7 percent of the time and produces fewer units than originally anticipated 61.3 percent of the time, leading to an error rate of 52.6 percent (the sum of 38.7 percent squared and 61.3 percent squared). This result implies that the error rate for the model is 52.6 percent or higher. Under random guessing for the SVR, we would estimate the Q-ratio to be equal to the average Q-ratio. The models did not predict the Q-ratio better than we would had we guessed the Q-ratio to be equal to the mean for all programs.

Possibly, the poor performance of this model is due to overfitting and the presence of noisy data, which are data that contain a large amount of meaningless information about the predictors of the Q-ratio. Overfitting occurs when a model uses the noise in the training data to construct models that do not generalize to new data (in this case, the test data). With noisy data, overfitting frequently occurs because machine-learning algorithms cannot detect the signal—that is, the actual relationship between the Q-ratio and text information—above the noise.

To address this issue, we adjusted the regularization parameter C to avoid overfitting. As this parameter was adjusted, however, there was no point at which the model worked; it either overfit to the data or found nothing. The combination of small sample size and noisy data makes it difficult to construct models to predict the Q-ratio with this dataset.

We also considered using an SVR as used for the Q-ratio to predict the quantity-adjusted program acquisition unit cost (PAUC) growth. The PAUC for a program is equal to the total acquisition cost of the program including development, procurement, and construction divided by the total number of units produced by the program. PAUC growth is measured as the difference between the PAUC reported in the final SAR of a program and the PAUC reported at Milestone B. This difference is then divided by the PAUC reported at Milestone B in order to calculate growth. This growth is then adjusted to reflect quantity. Specifically, the PAUC at the end of the program is calculated using the quantity reported at Milestone B instead of the actual quantity produced at the end of the program. This metric is of interest because being able to predict programs that may experience high levels of cost growth would be beneficial. Our analysis consisted of 25 programs for which quantity-adjusted PAUC growth data are available (data were taken from McNicol, 2018[25]). We used 70 percent of the programs as training data and the remaining 30 percent as test data. As with the Q-ratio, we obtained the scores associated with the first 50 LSA features for the training data. In our SVR, quantity-adjusted PAUC growth was the response and the 50 LSA components were the explanatory variables. This model also performed poorly and is likely due to overfitting and noisy data.

In addition to the SVCs and SVRs, we considered several alternatives. The first was to use cumulative data. In our previous analyses, we considered each document for a contract as a time point. However, when using cumulative data, we considered all documents up to a given time as a time point. For example, at the time point of 1 year, we combined all of the contracts and modifications for a program into one document and treated this "document of documents" as our data point. For year 2, we combined all of the documents associated with a given program for years 1 and 2. We then followed a similar process for all years of a program so that we had 1 data point for each year of a contract. We then created a bag-of-words, calculated tf-idf values, performed an LSA, and retained the first 50 components. Next, we performed the same analyses described earlier using the 50 LSA components from the cumulative data. The models using the cumulative data performed equally poorly as those using the original time series data.

We also considered a simpler model whose data consisted of the documents associated with a program during its first 5 years of operation. We then created a bag-of-words, calculated tf-idf values, performed an LSA, and retained the first 50 LSA components. The 50 LSA components were used in a Lasso regression, which is a regression method that attempts to improve prediction accuracy and model interpretability. Unlike the previous models we considered, this model did not account for the time series nature of our data, and therefore did not perform better than those considered previously.

[25] David L McNicol, "Acquisition Policy, Cost Growth, and Cancellations of Major Defense Acquisition Programs," IDA Report R-8396, (Alexandria, VA: Institute for Defense Analyses, 2018).

B. SVR to Predict Program End Date

In the models to predict Q-ratio and quantity-adjusted PAUC growth, we conducted our analyses at the MDAP level. This approach led to a small sample size (n = 31 for the Q-ratio analysis and n = 25 for the quantity-adjusted PAUC growth analysis) because the number of Army MDAPs that fit our inclusion criteria was small.[26] One way to compensate for the small sample size is to conduct analyses at the contract level. For each contract, we use information associated with the base contract and subsequent modifications. As with the analyses on the program level, the result is a multivariate time series for each contract with 50 LSA components measured at each time point, which is the date of the base contract or modification.

The outcome of this analysis is the end date of the contract, including all modifications. Though this outcome is the same as in the walking section, the current analysis differs as it is conducted on the contract level rather than the file level, and we account for the time series nature of our data. We did not have ground truths (the true actual end date) for the contracts, but we were able to extract a collection of dates from each document from which we could infer the end date. Each base contract or modification was associated with several dates, such as the date signed, the effective date, and delivery dates. We wrote a script that found all dates in a document and returned the latest date associated with it. We then combined the dates from all documents associated with a contract and wrote a script to return the latest date from the collection of latest dates. This date is used as a proxy for the contract's end date. The latest date was then compared to the human schedule projection in the first contract. Though this model performed better than the models for Q-ratio or quantity-adjusted PAUC growth, it still performed poorly and did not predict better than random guessing.

As for the Q-ratio and quantity-adjusted PAUC growth analyses, we considered other models, including one that used cumulative data in a Lasso regression where the first 5 years of contract information served as explanatory variables. The Lasso regression performed better than previous analyses but still performed poorly. In fact, none of the models we considered performed better than human predictions at the beginning of the contract. To evaluate our models' performance, we compared the predicted end date with the extracted end date. The predicted end date is the year of the latest date in the earliest document associated with the contract, presumably the base contract. Figure 5 compares the original end date with the latest end date for each program. These values are highly correlated with a correlation of 0.86. This plot also illustrates that the latest end date is

[26] One could argue that adding MDAPs from the other Services and Agencies could improve the results. Although this result is possible, it not likely in this case because the total number of MDAPs is still small for this type of analysis.

always the same as or later than the original end date, indicating that none of the programs ended earlier than expected, though many ended later than expected.

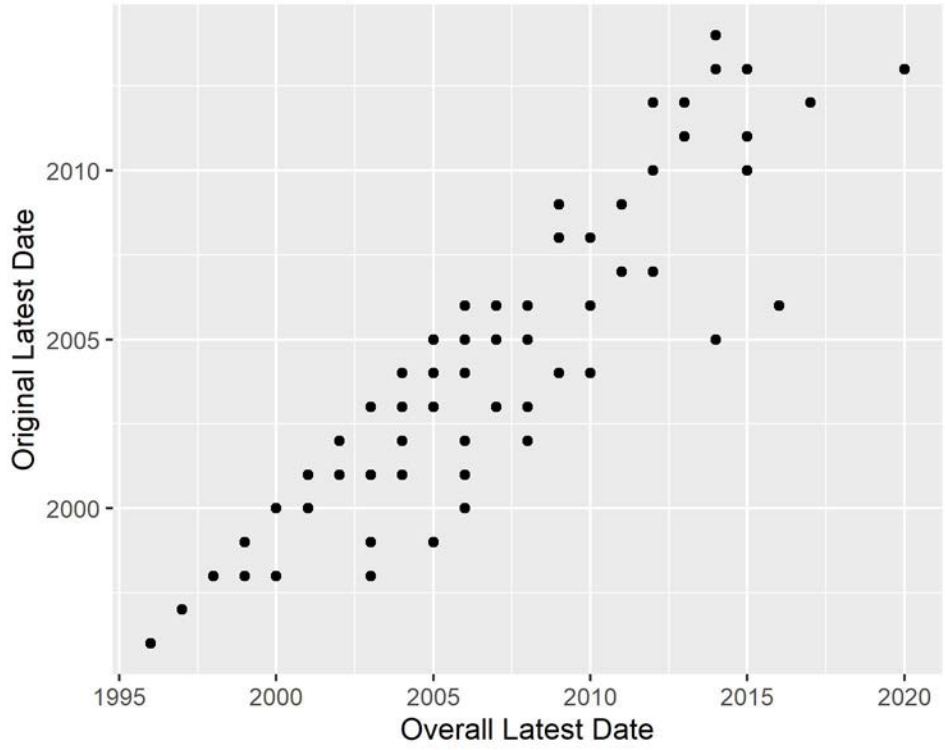


Figure 5. Contracting Officer Prediction Accuracy

In light of the success we had with data extraction and simpler models at the crawling and walking stages, this result suggests that the most cost-effective use of AI and natural language processing is to improve the predictions of humans by extracting structured data from documents as opposed to predicting program outcomes directly.

C. Cluster Analysis to Identify Similar Programs

The methods used previously are known as *supervised learning algorithms*. Their purpose is to learn information from data; the learned information is then used to predict outcomes in new data. This approach requires the data to be "labelled," meaning that we know the outcome. For example, in our Q-ratio analysis, one label for the SVC was whether the program met or exceeded expectations or produced fewer units than expected, and the actual value of the Q-ratio was the label for the SVR. Another set of machine-learning methods is *unsupervised learning algorithms*. The purpose of these methods is to learn patterns in the data. One of the most frequently used methods is clustering, which identifies similar clusters or groups in the data. In our case, we could use information extracted from the documents to identify similar programs.

Though there are several common clustering methods, their purpose is the same: to group similar observations. The main difference among algorithms is how similarity and dissimilarity are measured. One of the most frequently used clustering methods, which we use in our analysis, is k-means clustering from the scikit learn library in Python.[27] More detailed information about k-means clustering and other clustering methods is found in the blog *Towards Data Science*.[28]

To conduct a cluster analysis, we first complete the preprocessing steps described earlier using the documents of 34 MDAPs. Next, rather than separate the data into test and training sets, we used all programs because we have no identified outcomes for prediction. We then used a GAK with k-means clustering to group similar programs into clusters. Due to the small number of programs, we conducted our analysis with two or three clusters. The results of the analysis were mixed, so we focused on the results for two clusters. The programs assigned to each cluster for the two- and three-cluster solutions are found in Appendix D.

Our results indicate that each cluster contains approximately the same number of programs. Based on the programs in each cluster, there is no "obvious" reason for this grouping. An "obvious" grouping would be, for example, all aircraft MDAPs in one cluster and all other programs in another. To further investigate how these clusters differed, we created density plots of the Q-ratio, start date, and quantity-adjusted PAUC growth by cluster. As shown in Figure 6 and Figure 7, the densities for the Q-ratio and start date in both clusters are very similar as indicated by the similar red and blue lines. This result suggests that the two clusters do not differ substantially in terms of their Q-ratios and start dates. The plot of the clusters for quantity-adjusted PAUC growth, shown in Figure 8, indicates that the clusters differ mainly by quantity-adjusted PAUC growth. For the first cluster (in red), there is a large peak around 0, while this pronounced peak does not appear for the second cluster (in blue). This result suggests that the two clusters may differ based on cost growth, with one cluster having minimal growth and the other having greater cost growth. It is necessary to note that we considered only three factors that may differ among clusters, and that the differences between the clusters may be associated with other factors or combinations of factors.

---

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-Learn: Machine Learning in Python," *Journal of Machine Learning Research* 12: 2825–30, 2011.

[28] George Seif, "The 5 Clustering Algorithms Data Scientists Need to Know," Towards Data Science, accessed February 5, 2018, https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68.

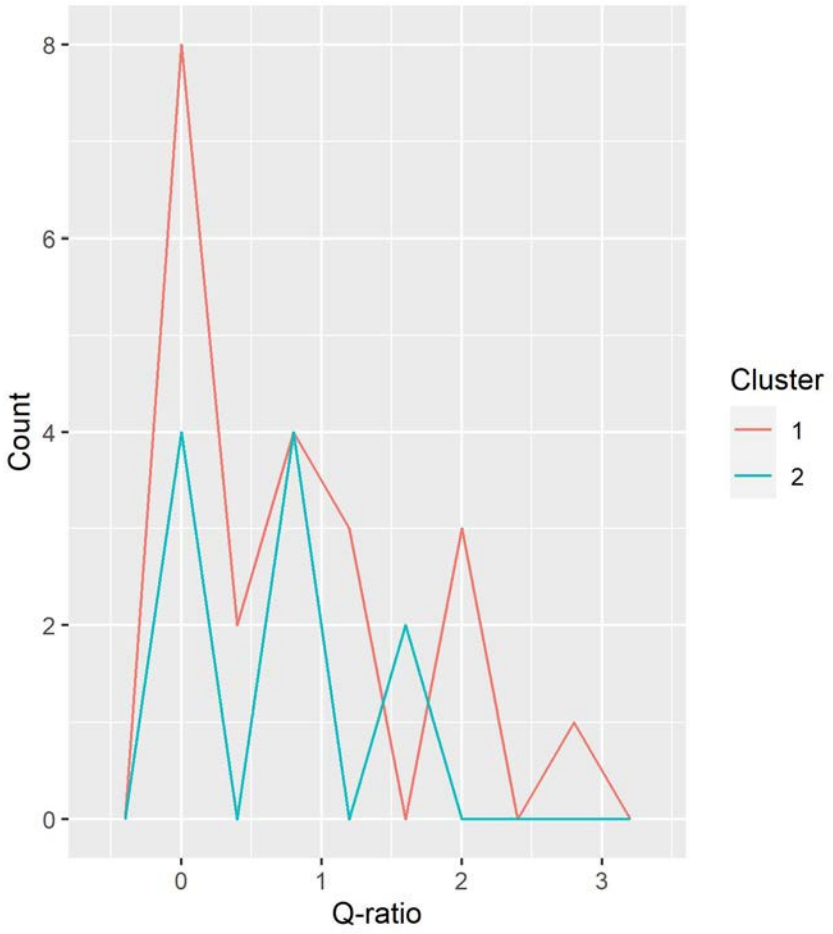


Figure 6. Cluster Distribution Comparison: Q-ratio

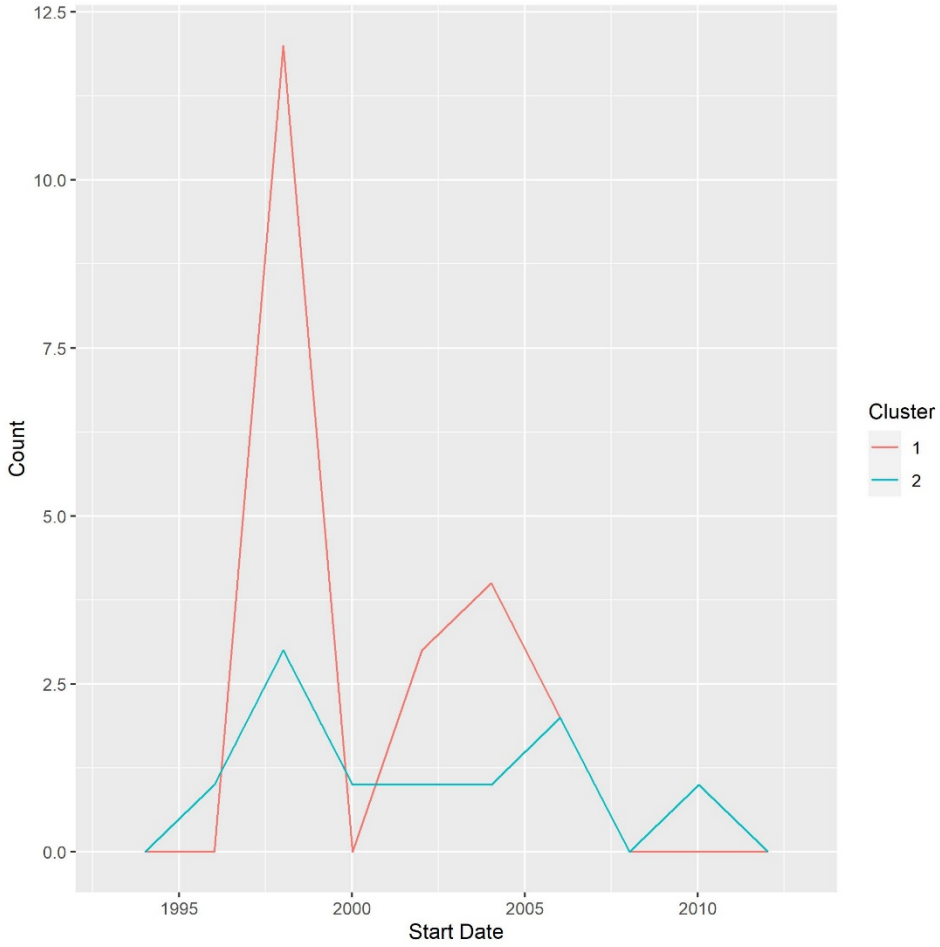


Figure 7. Cluster Distribution Comparison: Start Date

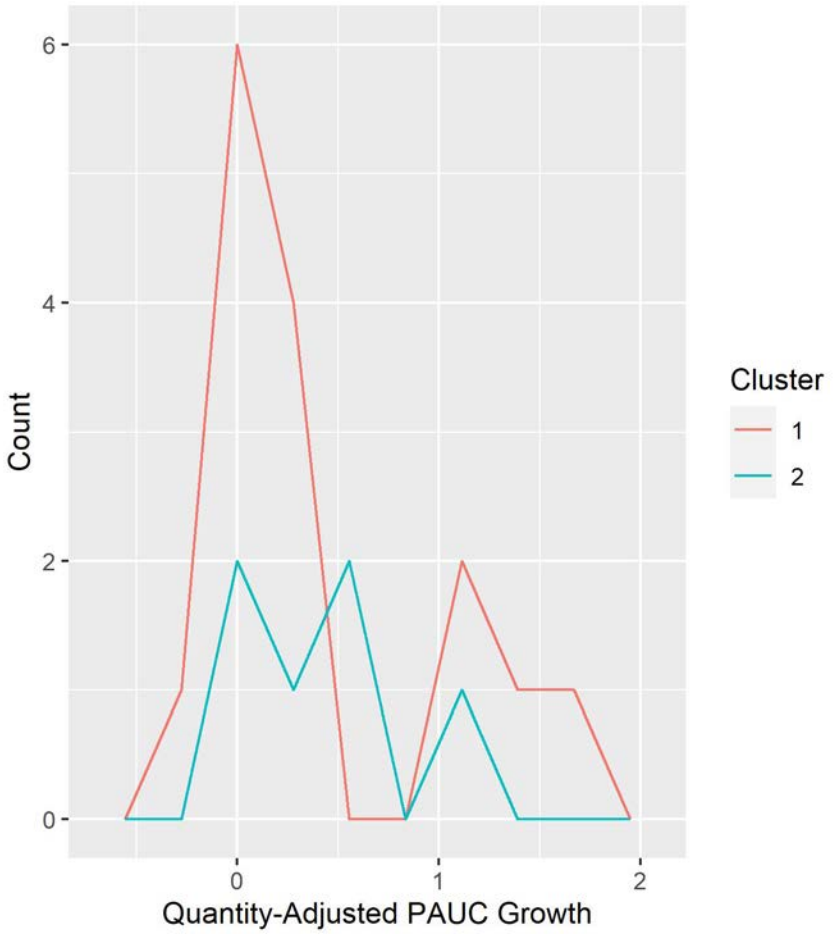


Figure 8. Cluster Distribution Comparison: Quantity-Adjusted PAUC Growth

To further investigate whether a combination of factors may create differences between the clusters, we used a scatterplot, which is a visual tool that uses colored points to denote cluster membership of potential factors. Figure 9 shows a scatterplot of the Q-ratio compared to start date. We obtained start dates using an approach similar to that for obtaining end dates, except that we used the earliest date instead of the latest. In this plot we do not see distinction between the two clusters (i.e., one distinct group of points for cluster 1 and another distinct group for cluster 2) as we would expect. Instead, we see overlap in the programs of the two clusters. Though this plot does not indicate that Q-ratio and start date together lead to differences between clusters, it does provide useful information for another purpose.

In this plot, we see considerable variation in the Q-ratio for programs with start dates between 1997 and 2002 while the Q-ratio is 0 for all but three programs with a start date after 2002. This result suggests a few possible explanations.

One explanation identifies a concern about data that may affect our analyses. Namely, our data for programs that start after 2002 may not be typical of all programs that started

after 2002. Because we include only programs for which the outcomes are known (complete or cancelled), programs that started after 2002 are not included in our analysis because they have not concluded. Consequently, cancelled programs may be overrepresented in our analyses so we must interpret the results cautiously.

Another possibility is that some change in acquisition management and policy changed procurement, or estimating habits, of MDAPs. Although interesting for further study, we did not investigate these possibilities because they are outside the scope of our study.

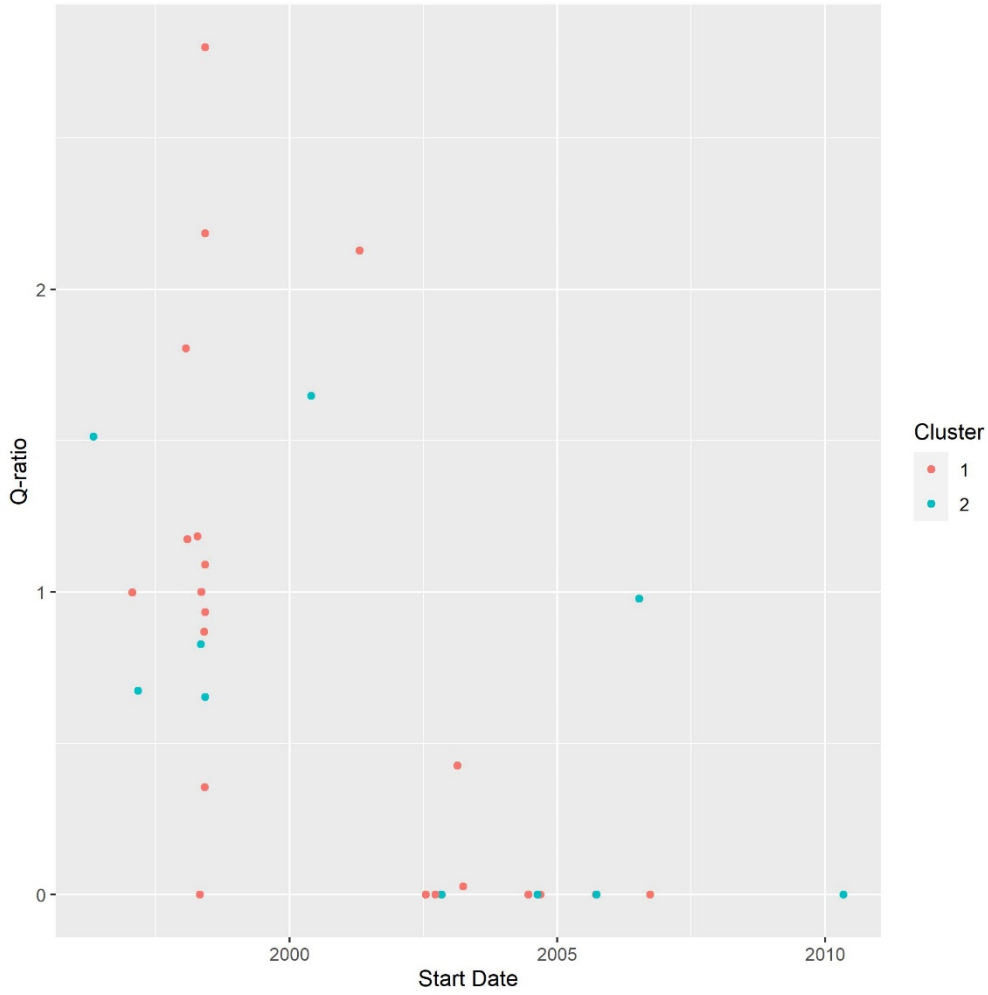


Figure 9. Q-ratio vs. Start Date by Cluster

5. Summary of Findings

The purpose of this study was threefold: (1) to conduct a literature review on the use of text analytics with machine learning to evaluate contracts and similar documents, (2) to obtain current DoD acquisition contracts and to convert these documents to a usable dataset, and (3) to assess the feasibility of using machine learning to understand and predict the performance of acquisition programs. Below we summarize the findings of our study.

A. Literature Review Findings

- In previous studies that evaluated MDAPs, data sources included SARs data, original contracts, historical memoranda, government documents from the Federal Procurement Data System, and DoD budget documents. For studies that used memoranda and other data sources with text, data were often extracted manually, which is time-consuming and prone to error.
- Previous studies evaluating MDAPs have typically used relatively simple statistical and machine-learning algorithms, such as descriptive displays, linear regression, and logistic regression, and have not investigated more complex methods such as SVMs or cluster analysis. However, the predictive ability of the methods used in these studies is low. Because these analyses included only quantifiable information and did not consider qualitative information, it was unclear whether combining qualitative and quantitative variables could improve predictions of program outcomes.
- The number of studies that used text mining and text analytics was limited. These studies typically use information from the format 5 portions of contractor performance reports for data. These studies also used naïve Bayes classifiers to identify at-risk programs and linear regressions to predict EAC. Other studies were more descriptive and identified trends in legislative reforms for acquisition. These studies also used word frequencies to investigate increases in cost-estimate terms in NDAAs and to identify whether a program experienced a schedule breach or cost overrun.

B. Document Collection and Dataset Construction Findings

- We used the EDA system to download contracts and associated modifications for 34 Army MDAPs that began after 1997 and stopped issuing SARs as of

November 2019. In total, we downloaded 24,364 files representing 149 contracts.

- IDATA easily extracted data from the 24,364 files and placed them in a spreadsheet for analysis. This activity showed that we could convert contracts and similar documents into a structured format suitable for data analysis.

C. Feasibility Findings

- The extracted data were well-suited for constructing descriptive visualizations such as word clouds. When examining the word clouds, the most frequent words were as expected. This suggested that we extracted correct and useful information from the documents.
- Initially, we conducted relatively simple analyses at the file level to validate our data and investigate the feasibility of exploring more complex models with these data. We used a naïve Bayes classifier to predict document type and a linear regression model to predict end date. The naïve Bayes classifier correctly predicted document type for more than 95 percent of unseen (or "new") documents while the linear regression led to predicted end dates that were highly correlated with actual end dates ($r = 0.95$). These results provided further evidence that we extracted useful information from the documents, and that we could use this information for analysis at the file level.
- We used SVMs to investigate whether we could predict the Q-ratio, a metric of changes in procurement quantity over a program's life cycle. In this analysis, we used an SVC to predict whether a program would have a Q-ratio of less than 1 or greater than or equal to 1, and an SVR to predict the Q-ratio. The predictive ability of this model was poor, likely due to the small sample size and noisy data. These results indicated that some machine-learning methods may not be well-suited for predicting certain acquisition outcomes with our limited dataset. This outcome also may be due to the low number of MDAPs evaluated, limiting the number of datasets available to build and train the models. The number of datasets we used was below the level traditionally required for this type of analysis. Additionally, these outcomes are inherently hard to predict by any previously used analytical methods.
- We used an SVR model to predict quantity-adjusted PAUC growth, which is a metric of cost growth. This model also had poor predictive ability, likely for similar reasons as the Q-ratio analysis.
- We used k-means clustering to group similar programs. The Q-ratio or start date of the clusters did not differ, but we observed potential differences in the quantity-adjusted PAUC growth: one group had near zero growth and the other

group had varying levels of growth. Further investigation into the use of clustering is warranted.

- Based on our findings, we ultimately concluded that text mining could easily be used to extract meaningful information from contracts and similar documents. The data extracted from contracts were well-suited for descriptive purposes; for example, we could use these data to construct a dashboard that tracked the progress of MDAPs and contracts. Our results indicated that the dataset we used was not well-suited for predictive purposes for the variables we evaluated.
- Predicting outcomes of interest, such as cost growth, schedule delays, and quantity changes, is problematic because factors that impact these outcomes may be difficult to observe and measure (e.g., political climate) and because there are high levels of uncertainty. Further compounding these issues is the relatively small number of MDAPs that we included in our dataset.[29] The use of noisy data and small sample size makes prediction difficult even when using sophisticated, machine-learning algorithms.
- Our results suggest that further research into unsupervised learning methods is warranted. This research could allow characteristics of successful programs to be identified and could be useful for inferring whether current or future programs are likely to succeed. We recommend that further research use data extracted from contracts in conjunction with other data sources to characterize and predict program outcomes. The combination of data sources may enhance predictive ability by using more information.

[29] There are not orders of magnitude of additional MDAPs that fit the criteria for this dataset. At most, there are 3–4 times the number of MDAPs used in this study.

Appendix A.
Overview of Text Analytics

Text analytics encompasses a broad set of techniques that convert unstructured data found in texts, such as contracts and their modifications, into structured data that can be used in subsequent analyses. This process is also commonly referred to as *text mining* and *machine learning from text*, depending on the field of study, so we use these terms interchangeably. In this section, we provide an overview of the process of text analytics.

Identify and Convert Documents

The first step is to identify the set of documents of interest, referred to as the *corpus*. These documents can be from different source formats, such as PDF files, websites, or comma-separated values files. Our corpus consists of PDF versions of the contracts and modifications of the Army MDAPs described in section 1.B. After the corpus is identified, the files are imported into a software application that converts the files into machine-readable formats (in our case, text (.txt) files). For this conversion, we used the IDATA capability, which was developed at IDA for text analytics. More details on IDATA are provided in Chapter 2.

Preprocess the Documents

After the corpus has been converted to machine-readable formats, the text must be preprocessed to allow for future analysis. Preprocessing often consists of three components: tokenization, normalization, and substitution.

Tokenization

Tokenization consists of breaking the text into pieces, called tokens, which are contiguous sequences of characters that have meaning (typically words). For example, a sentence with 12 words would have 12 tokens. Although this process sounds simple, it is quite complicated due to the complexities of the English language: hyphenated words, apostrophes, periods in abbreviations, and words that contain combinations of letters and numbers (such as 12^{th}) or numbers and punctuation marks (such as 14,053).

Normalization

Once tokenized, the data are then normalized by a process that converts all text to a standard format, such as lowercase letters. This process typically involves stemming and

lemmatization. Stemming removes affixes (such as suffixes and prefixes) to obtain the stem or root of a word. For example, "walking" becomes "walk." Lemmatization is similar to stemming in that both techniques reduce words to a common base or root, but lemmatization considers context while stemming does not. For example, with lemmatization, "better" becomes "good," but under stemming "better" does not change.

Substitution

In addition to tokenization and normalization, other steps, collectively referred to as *substitution*, are needed to ensure text is in a standardized format. These steps typically involve the substitution or removal of words. Common steps include the removal of numbers or their conversion to text; removal of punctuation (this often occurs during tokenization but may occur during substitution); deletion of white spaces (also often done during tokenization); and the removal of stop words. Stop words, which commonly include "the," "and," and "a," occur frequently in a language and thus add little to the understanding of a document.

Convert the Clean Text Using Predictive Analytics

After preprocessing, the clean text must be converted into a format that is usable in statistical and machine-learning models.

Bag-of-Words Model

A frequently used model is the bag-of-words, in which grammar and word order are ignored and the frequency of each word is recorded. The resulting dataset lists each unique word in one column and its frequency in another. Each column represents a word found in at least one document of the corpus and each row represents one document. Each cell indicates either the frequency of a given word for a specific document or whether the word is present. A bag-of-words is obtained for each document in the corpus, and these are combined into a single dataset.

To illustrate this model, consider an example from the blog *Analytics Vidhya*.[1] Suppose there are three movie reviews:

Review 1: This movie is very scary and long.

Review 2: This movie is not scary and is slow.

Review 3: This movie is spooky and good.

---

[1] Purva Huilgol, "Quick Introduction to Bag-of-Words (BoW) and TF-IDF for Creating Features from Text," *Analytics Vidhya*, accessed February 28, 2020, https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/.

Each review is a document and the collection of the three reviews is the corpus. In total, there are 11 unique words in the corpus: “this,” “movie,” “is,” “very,” “scary,” “and,” “long,” “not,” “slow,” “spooky,” and “good.” The resulting bag-of-words is shown in Table A-1.

Table A-1. Sample Bag-of-Words

	this	movie	is	very	scary	and	long	not	slow	spooky	good	Length of review
Rev. 1	1	1	1	1	1	1	1	0	0	0	0	7
Rev. 2	1	1	2	0	0	1	1	0	1	0	0	8
Rev. 3	1	1	1	0	0	0	1	0	0	1	1	6

Because common words such as “the” and “and” occur with high frequency, frequencies are normalized through weighting so that more important words receive greater weight. (Note that normalization here is not the same as normalization via stemming and lemmatization.) The most common weighting method is *term frequency-inverse document frequency* (*tf-idf*), in which the tf-idf value of a word reflects its importance in a document. The value is calculated as the ratio of term frequency to inverse document frequency. Term (or word) frequency is the number of times a word occurs in a document divided by the total number of words in the document. Inverse document frequency of a word is calculated as the natural log of the ratio of the total number of documents in the corpus divided by the number of documents that contain the given word. Large tf-idf values indicate words that are relevant to a particular document while small values indicate that the words are not relevant.

N-gram Model

Because the bag-of-words model does not consider word order, other models are used to preserve order information. One such model is the *n*-gram, where an *n*-gram represents a sequence of *n* words (for example, San Francisco is a 2-gram word, time of day is a 3-gram, and so on). Similar to a bag-of-words, the frequency of each *n*-gram is recorded, and these frequencies may also be weighted so that larger scores reflect more important *n*-grams.

Visualizations

Bags-of-words and *n*-grams can both be used in further descriptive and predictive analyses of text datasets. Descriptive analyses typically involve visualizations, the simplest of these being the *wordle*: The words in a document are displayed so that the size of the text is proportionate to word frequency. Other graphics that display similarities among

words are dendrograms and word clouds, which identify clusters of similar words, and network graphs, which display relationships between words.

Latent Semantic Analysis (LSA)

In predictive analytics, information from bags-of-words and *n*-grams can be incorporated into regression models, classification and regression trees, Bayesian networks, and other methods. The major issue with using bags-of-words and *n*-grams is that they are high-dimensional, meaning that they contain a large number of words or sequences of words. One method frequently used to reduce the dimensionality while maintaining as much original data as possible is latent semantic analysis (LSA).

LSA consists of four steps. In the first step, the corpus is represented using a bag-of-words (described previously) with each cell denoting the frequency of a given word in a specific document. In the second step, the frequencies are transformed so that more important words receive higher weight. Step three involves using singular value decomposition on the matrix. This method allows the original matrix to be approximated using fewer dimensions than the original by decomposing the matrix into its constituent elements (or latent topics). Lastly, in step four, a dimension-reduced matrix is used in subsequent analyses such as identifying similar documents, classifying new and unseen documents, and retrieving information. A useful overview of LSA and its applications is found in Dumais (2004);[2] Landauer, Foltz, and Laham (1998)[3] provide greater mathematical detail of this method.

Support Vector Machines

An additional application of LSA, seen in our later analyses, is to use the topics identified in the singular value decomposition in subsequent analyses. Frequently used in text analytics are *support vector machines* (SVMs), which can be used for both classification and regression. SVMs were originally developed as binary classifiers, though they have been extended to allow classification for more than two groups and for predicting continuous variables. For clarity, we call SVMs used for classification *support vector classifiers* (SVCs) and SVMs used for regression *support vector regressions* (SVRs).

The goal of both SVCs and SVRs is to use a set of explanatory variables to make predictions. For SVCs, we predict the class of an observation. For example, we may use the information from our documents to predict whether the document is for a procurement or RDT&E contract. SVRs, on the other hand, predict continuous outcomes such as the program acquisition unit cost (PAUC) or the end date of a program. The explanatory

---

[2] Susan T. Dumais, "Latent Semantic Analysis," *Annual Review of Information Science and Technology* 38 (1), Wiley Online Library: 188–230, 2004.

[3] Thomas K. Landauer, Peter W. Foltz, and Darrell Laham, "An Introduction to Latent Semantic Analysis," *Discourse Processes* 25 (2-3), Taylor & Francis: 259–84, 1998.

variables used in SVCs and SVRs can include both continuous and categorical variables. A more technical and mathematical introduction to SVMs is found in Cristianini, Shawe-Taylor, and others (2000),[4] while a less technical explanation is found in the *Monkey Learn Blog*.[5] When evaluating texts, one option for the explanatory variables is to use the tf-idf values associated with the words found in the text corpus. Another option, which we use later in this study, is to use the scores associated with the LSA topics as explanatory variables.

Text analytics can also be used to extract structured information from unstructured text, a process known as information extraction or retrieval. For example, we may want to extract information on the projected completion date of an MDAP or the anticipated quantity to be produced. This information can then be formatted for use in subsequent analyses.

[4] Nello Cristianini, John Shawe-Taylor, et al., *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, 2000.

[5] Bruno Stecanella, "An Introduction to Support Vector Machines," 2017, https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/.

Appendix B.
MDAPS and Their Contracts

Table B-1 provides the MDAPs used in this study and the number of contracts associated with each MDAP by contract type (procurement, RDT&E, or acquisition O&M).

Table B-1. MDAPs and Number of Contracts

MDAP Short Name	Number of Each Contract Types		
	Procurement	RDT&E	Acq. O&M
ABRAMS UPGRADE	4	1	
ACS		1	
AFATDS (ATCCS)		1	
ARH		1	
ASAS (ATCCS)		1	
ATACMS-APAM	3		
ATACMS-BAT	2	3	
ATIRCM-CMWS	3	2	
BLACK HAWK (UH-60A/L)	3		
BRADLEY UPGRADE	3		
CH-47F	3	3	
Comanche		4	
FAAD C2I (ATCCS)	1		
FBCB2	3	3	
FCS		1	
FMTV	5		
HIMARS	4		
INCREMENT 1 E-IBCT	1	1	
JAVELIN	4		
JOINT COMMON MISSILE		1	
JTRS GMR		1	
LAND WARRIOR	1	1	
LONGBOW APACHE	11	1	
LONGBOW HELLFIRE	3		
LUH		1	

MDAP Short Name	Number of Each Contract Types		
	Procurement	RDT&E	Acq. O&M
PAC-3	16	5	
Patriot/MEADS CAP	1	2	
SINCGARS	7		
SMART-T	2		
STRYKER	2		1
WIN-T		1	
WIN-T INC 1	2		
WIN-T INC 2	1		
WIN-T INC 3		2	
Total	**85**	**37**	**1**

Table B-2 provides information for each contract including the program with which the contract is associated, the contract number, appropriation category, and number of files.

Table B-2. Contract Numbers by Program and Number of Files

Program	Contract Number	Appropriation Category	Number of Files
ABRAMS UPGRADE	DAAE0700CN044	Procurement	49
ABRAMS UPGRADE	DAAE0701CN040	Procurement	131
ABRAMS UPGRADE	DAAE0794C0727	RDT&E	55
ABRAMS UPGRADE	DAAE0795C0292	Procurement	207
ABRAMS UPGRADE	DAAE0798C0033	Procurement	18
ACS	W15P7T04CJ409	RDT&E	19
AFATDS (ATCCS)	DAAB0790CE708	RDT&E	101
ARH	W58RGZ05C0234	RDT&E	55
ASAS (ATCCS)	DAAB0794CA515	RDT&E	153
ATACMS-APAM	DAAH0192C0038	Procurement	161
ATACMS-APAM	DAAH0194C0002	Procurement	33
ATACMS-APAM	DAAH0198C0093	Procurement	199
ATACMS-BAT	DAAH0101C0133	Procurement	26
ATACMS-BAT	DAAH0195C0001	RDT&E	166
ATACMS-BAT	DAAH0198C0105	RDT&E	57
ATACMS-BAT	DAAH0199C0121	Procurement	111
ATACMS-BAT	DAAH0199C0154	RDT&E	90
ATIRCM-CMWS	DAAB0702CB213	RDT&E	19
ATIRCM-CMWS	DAAB0795CD606	RDT&E	115

Program	Contract Number	Appropriation Category	Number of Files
ATIRCM-CMWS	W15P7T04CJ404	Procurement	9
ATIRCM-CMWS	W15P7T04D0055	Procurement	114
ATIRCM-CMWS	W15P7T06DT206	Procurement	1,066
BLACK HAWK (UH-60A/L)	DAAJ0994C0044	Procurement	9
BLACK HAWK (UH-60A/L)	DAAJ0997C0005	Procurement	335
BLACK HAWK (UH-60A/L)	DAAJ0997D0196	Procurement	293
BRADLEY UPGRADE	DAAE0700CM002	Procurement	71
BRADLEY UPGRADE	DAAE0701CM016	Procurement	186
BRADLEY UPGRADE	W56HZV05G0005	Procurement	716
CH-47F	DAAH2301C0028	Procurement	24
CH-47F	DAAH2303C0022	Procurement	39
CH-47F	DAAH2398C0069	RDT&E	78
CH-47F	W58RGZ04C0012	RDT&E	90
CH-47F	W58RGZ04G0023	RDT&E	1,484
CH-47F	W58RGZ13C0002	Procurement	122
Comanche	DAAH2300CA001	RDT&E	1
Comanche	DAAH2302C0122	RDT&E	25
Comanche	DAAJ0991CA004	RDT&E	2
Comanche	DAAJ0992C0453	RDT&E	66
FAAD C2I (ATCCS)	DAAH0194CS199	Procurement	129
FBCB2	DAAB0700DE501	Procurement	386
FBCB2	DAAB0701DE502	RDT&E	400
FBCB2	DAAB0795DE604	RDT&E	669
FBCB2	W15P7T04DG204	Procurement	104
FBCB2	W15P7T04DG205	RDT&E	466
FBCB2	W15P7T06DJ405	Procurement	270
FCS	W56HZV05C0724	RDT&E	340
FMTV	DAAE0703CS023	Procurement	373
FMTV	DAAE0792CR001	Procurement	159
FMTV	DAAE0798CM005	Procurement	222
FMTV	W56HZV08C0460	Procurement	202
FMTV	W56HZV09D0159	Procurement	627
HIMARS	DAAH0103C0005	Procurement	163
HIMARS	W31P4Q06C0001	Procurement	93
HIMARS	W31P4Q08C0001	Procurement	118
HIMARS	W31P4Q11C0101	Procurement	36
INCREMENT 1 E-IBCT	W31P4Q04C0059	RDT&E	248

Program	Contract Number	Appropriation Category	Number of Files
INCREMENT 1 E-IBCT	W56HZV09C0452	Procurement	107
JAVELIN	DAAH0100C0108	Procurement	207
JAVELIN	DAAH0196C0147	Procurement	39
JAVELIN	DAAH0197C0209	Procurement	109
JAVELIN	W31P4Q04C0136	Procurement	154
JOINT COMMON MISSILE	W31P4Q04C0094	RDT&E	57
JTRS GMR	DAAB0702CC403	RDT&E	116
LAND WARRIOR	DAAB0703CN001	RDT&E	170
LAND WARRIOR	W15P7T05CF201	Procurement	64
LONGBOW APACHE	DAAH2300C0001	Procurement	335
LONGBOW APACHE	DAAH2301C0092	Procurement	104
LONGBOW APACHE	DAAH2303C0164	Procurement	50
LONGBOW APACHE	DAAH2398C0008	Procurement	62
LONGBOW APACHE	DAAJ0995CA001	Procurement	167
LONGBOW APACHE	DAAJ0995CA002	Procurement	16
LONGBOW APACHE	DAAJ0996C0114	Procurement	15
LONGBOW APACHE	DAAJ0997C0124	Procurement	64
LONGBOW APACHE	W58RGZ04C0302	Procurement	59
LONGBOW APACHE	W58RGZ05C0274	RDT&E	52
LONGBOW APACHE	W58RGZ06C0093	Procurement	82
LONGBOW APACHE	W58RGZ06C0169	Procurement	275
LONGBOW HELLFIRE	DAAH0191C0057	Procurement	51
LONGBOW HELLFIRE	DAAH0197C0082	Procurement	79
LONGBOW HELLFIRE	DAAH0199C0086	Procurement	129
LUH	W58RGZ06C0194	RDT&E	1,082
PAC-3	DAAH0102C0050	Procurement	73
PAC-3	DAAH0102C0075	Procurement	44
PAC-3	DAAH0103C0017	Procurement	49
PAC-3	DAAH0103C0191	Procurement	62
PAC-3	DAAH0189C0458	RDT&E	11
PAC-3	DAAH0191C0602	RDT&E	5
PAC-3	DAAH0195C0021	RDT&E	117
PAC-3	DAAH0195C0022	RDT&E	36
PAC-3	DAAH0195C0446	Procurement	182
PAC-3	DAAH0196C0018	RDT&E	2
PAC-3	DAAH0198C0062	Procurement	141
PAC-3	W31P4Q05C0051	Procurement	58

Program	Contract Number	Appropriation Category	Number of Files
PAC-3	W31P4Q06C0180	Procurement	103
PAC-3	W31P4Q09C0002	Procurement	107
PAC-3	W31P4Q10C0002	Procurement	74
PAC-3	W31P4Q11C0001	Procurement	132
PAC-3	W31P4Q12C0002	Procurement	80
PAC-3	W31P4Q12C0100	Procurement	55
PAC-3	W31P4Q12G0001	Procurement	166
PAC-3	W31P4Q13C0068	Procurement	57
PAC-3	W31P4Q14C0034	Procurement	213
Patriot/MEADS CAP	DAAH0103C0164	RDT&E	134
Patriot/MEADS CAP	W31P4Q07G0001	RDT&E	226
Patriot/MEADS CAP	W31P4Q12C0001	Procurement	26
SINCGARS	DAAB0794CC401	Procurement	1
SINCGARS	DAAB0794CC402	Procurement	1
SINCGARS	DAAB0795CC502	Procurement	8
SINCGARS	DAAB0795CC503	Procurement	15
SINCGARS	DAAB0796CC501	Procurement	21
SINCGARS	DAAB0796CC502	Procurement	16
SINCGARS	DAAB0797CC600	Procurement	46
SMART-T	DAAB0702DD010	Procurement	107
SMART-T	DAAB0796CA757	Procurement	158
STRYKER	DAAE0700DM051	Procurement	1,889
STRYKER	DAAE0702CB001	Acquisition O&M	215
STRYKER	W56HZV07DM112	Procurement	2,154
WIN-T	DAAB0702CF403	RDT&E	41
WIN-T INC 1	W15P7T06DL219	Procurement	224
WIN-T INC 1	W15P7T07DK001	Procurement	691
WIN-T INC 2	W15P7T10DC007	Procurement	1,108
WIN-T INC 3	DAAB0702CF404	RDT&E	249
WIN-T INC 3	W15P7T14D0002	RDT&E	122
Total			**24,364**

Appendix C.
Technical Details for the Walking Stage

This appendix provides technical details for the following analyses conducted in the walking stage of our study: (1) linear regression to predict end date and (2) Naïve Bayes classifier to predict document type.

Linear Regression to Predict End Date

In our analysis, each PDF file is converted to raw text through the PDF2TXT utility on the Linux operating system. The scikit-learn library in the Python[1] programming language is used for the linear regression model. The sparse matrix is created from raw text from 807 documents (identified as either procurements or contracts) with the class **sklearn.feature_extraction.text.TfidfVectorizer**. The following parameters deviate from their default values:

- stop words = inc, the, and
- max_df = 0.99
- min_df = 2
- ngram_range = (1,2)
- sublinear_tf = True

The **token_pattern** parameter is set to split the raw text into tokens when it detects a pattern of three or more alphabetical characters with a non-alphanumeric character on each side (e.g., punctuation, white space, new line character, and so on).

The **min_df** parameter ignores tokens that occur in less than 2 percent of the document's raw text, while **max_df** ignores tokens that occur in more than 99 percent of the document's raw text.

The **stop_words** parameter ignores the tokens **inc**, **the**, and **and**. The **ngram_range** parameter uses unigrams and bigrams, resulting in more features (e.g., **Our model uses bigrams** is tokenized into **Our**, **model**, **uses**, **bigrams**, **our model**, **model uses**, and **uses bigrams**). The **sublinear_tf** parameter adds 1 to the tf-idf normalized scores.

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-Learn: Machine Learning in Python," *Journal of Machine Learning Research* 12: 2825–30, 2011.

An LSA is performed on the matrix using the class **sklearn.decomposition.TruncatedSVD**. The one parameter that deviates from its default value is **n_components**, which is set to 100. This parameter compresses the matrix to 100 features and is the LSA value recommended by the scikit-learn library documentation.[2]

The compressed matrix is used to predict the latest date of each document using a linear regression model from the class **sklearn.linear_model.LinearRegression** with all default values. The latest date is extracted, or mined, from the document's raw text using regular expressions to match patterns for 11 of the most commonly formatted dates.

The features and predictors are split into 5 training and test sets using the class **sklearn.model_selection.StratifiedKFold** with all default values, except the **shuffle** parameter is set to True. This parameter randomly shuffles the data before splitting them into 5 groups. Each group has the same distribution of data, or as close to the same distribution as possible. The model trains on 80 percent of the data and tests on 20 percent for each split.

Naïve Bayes Classifier to Predict Document Type

The text is transformed into a bag-of-words sparse matrix with the same specifications described in the previous section. The **TruncatedSVD** function is not performed on the sparse matrix as is not recommended with Naïve Bayes models.

The class **sklearn.naive_bayes.ComplementNB**, a Naïve Bayes model optimized for imbalanced datasets, is used to classify the documents with one of 5 labels. All default values are used except for **alpha**, which is set to 0.005.

[2] Ibid.

Appendix D.
Cluster Membership for Two- and Three-Cluster Solutions

Table D-1 presents cluster membership for the two-cluster solution, which groups similar programs in each cluster.

Table D-1. Cluster Membership for a Two-Cluster Solution

Cluster 1	Cluster 2
ABRAMS UPGRADE	ACS
AFATDS	ARH
ATACMS-APM	ASAS
ATACMS-BAT	BRADLEY UPGRADE
ATIRCM-CMWS	CH-47F
FCS	FAAD C2I
HIMARS	FBCB2
JAVELIN	FMTV
JOINT COMMON MISSILE	INCREMENT 1 I-EBCT
LONGBOW APACHE	JTRS GMR
PAC-3	LAND WARRIOR
WIN-T INC 1	LONGBOW HELLFIRE
	LUH
	SINCGARS
	SMART-T
	STRYKER
	WIN-T
	WIN-T INC 2
	WIN-T INC 23

Table D-2 provides the results for the three-cluster solution. Programs within the same cluster are considered to be similar to one another, while programs in different clusters are considered dissimilar to one another.

Table D-2. Cluster Membership for a Three-Cluster Solution

Cluster 1	Cluster 2	Cluster 3
ARH	AFATdS	ABRAMS UPGRADE
ASAS (ATCCS)	FAAD C2I (ATCCS)	ACS
ATACMS-APAM	Increment 1 E-IBCT	ATACMS-BAT
BRADLEY UPGRADE	JOINT COMMON MISSILE	ATIRCM-CMWS
FBCB2	LUH	CH-47F
FCS	SINCGARS	FMTV
HIMARS	STRYKER	JAVELIN
JTRS GMR	WIN-T INC 1	LAND WARRIOR
LONGBOW APACHE		LONGBOW HELLFIRE
PAC3		SMART-T
WINT		WIN-T INC 2
		WIN-T INC 3

Illustrations

Figures

Table

References

Albert, Michelle, and Arun Maiya. “IDATA Overview.” IDA Research Notes. (Alexandria, VA: Institute for Defense Analyses, 2018).

Algarín, Liana. “Human Systems Engineering and Program Success—A Retrospective Content Analysis.” *Defense Acquisition Research Journal*, 23 (1): 78–101, 2016.

Berteau, David, Joachim Hofbauer, Gregory Sanders, and Guy Ben-Ari. “Cost and Time Overruns in Major Defense Acquisition Programs.” Center for Strategic and International Studies, 2010. https://www.csis.org/analysis/cost-and-time-overruns-major-defense-acquisition-programs-2011-0.

Brown, G.E. “Measuring the Increasing Relevance of Cost Estimating Through Text Analytics.” *ICEAA World* 2: 32–33, 2017.

Cristianini, Nello, John Shawe-Taylor, et al. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

Cuturi, Marco. “Fast Global Alignment Kernels.” *Proceedings of the 28th International Conference on Machine Learning*, ICML 2011, 929–36.

Dumais, Susan T. “Latent Semantic Analysis.” *Annual Review of Information Science and Technology* 38 (1). Wiley Online Library: 188–230, 2004.

Freeman, Charlton E. “Multivariate and Naïve Bayes Text Classification Approach to Cost Growth Risk in Department of Defense Acquisition Programs.” PhD diss., Air Force Institute of Technology, 2013. https://apps.dtic.mil/dtic/tr/fulltext/u2/a583708.pdf.

Huilgol, Purva. “Quick Introduction to Bag-of-Words (BoW) and TF-IDF for Creating Features from Text.” *Analytics Vidhya*, 28 February 2020. https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/.

Landauer, Thomas K., Peter W. Foltz, and Darrell Laham. “An Introduction to Latent Semantic Analysis.” *Discourse Processes* 25 (2-3). Taylor & Francis: 259–84, 1998.

Light, Thomas, Robert S. Leonard, Julia Pollak, Meagan L. Smith, and Akilah Wallace. “Quantifying Cost and Schedule Uncertainty for Major Defense Acquisition Programs (MDAPs).” Rand Corporation, 2017. https://www.rand.org/pubs/research_reports/RR1723.html.

Lorell, Mark A., Leslie Adrienne Payne, and Karishma R. Mehta. “Program Characteristics That Contribute to Cost Growth: A Comparison of Air Force Major Defense Acquisition Programs.” Rand Corporation, 2017. https://www.rand.org/pubs/research_reports/RR1761.html.

McCormick, Rhys, Andrew Hunter, and Gregory Sanders. “Preliminary Findings: Is the Ratio of Investment Between R&D to Production Experiencing Fundamental Change?” Naval Postgraduate School. Monterey, California, 2018. https://calhoun.nps.edu/handle/10945/58752.

McGowin, Amanda L. “An Analysis of Major Acquisition Reforms Through Text Mining and Grounded Theory Design.” PhD diss., Air Force Institute of Technology, 2018. https://apps.dtic.mil/dtic/tr/fulltext/u2/1056519.pdf.

McNicol, David L. “Acquisition Policy, Cost Growth, and Cancellations of Major Defense Acquisition Programs.” IDA Report R-8396. (Alexandria, VA: Institute for Defense Analyses, 2018).

McNicol, David L., David M Tate, Sarah K Burns, and Linda Wu. “Further Evidence on the Effect of Acquisition Policy and Process on Cost Growth of Major Defense Acquisition Programs.” IDA Paper P-5330-REVISED. (Alexandria, VA: Institute for Defense Analyses, 2016).

McNicol, David L., and Linda Wu. “Evidence on the Effect of DoD Acquisition Policy and Process on Cost Growth of Major Defense Acquisition Programs.” IDA Paper P-5126. (Alexandria, VA: Institute for Defense Analyses, 2014).

Miller, Trevor P. “Acquisition Program Problem Detection Using Text Mining Methods.” PhD diss., Air Force Institute of Technology, 2012. https://apps.dtic.mil/dtic/tr/fulltext/u2/a557568.pdf.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. “Scikit-Learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12: 2825–30, 2011.

Ritschel, Jonathan D., Robert D. Fass, and Bradley C. Boehmke. “A Text Mining Analysis of Acquisition Reforms and Expert Views.” *Defense AR Journal* 25 (3): 288–323, 2018.

Seif, George. “The 5 Clustering Algorithms Data Scientists Need to Know.” *Towards Data Science*, 5 February 2018. https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68.

Stecanella, Bruno. “An Introduction to Support Vector Machines.” *Monkey Learn*, 2017. https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/.

Tyson, Karen W., Bruce R. Harmon, and Daniel M. Utech. “Understanding Cost and Schedule Growth in Acquisition Programs.” IDA Paper P-2967. (Alexandria, VA: Institute for Defense Analyses, 1994).

Younossi, Obaid, Lionel A. Galway, Bernard Fox, and John C. Graser. “Impossible Certainty: Cost Risk Analysis for Air Force Systems,” Vol. 415. Rand Corporation, 2006. https://www.rand.org/pubs/monographs/MG415.html.

Abbreviations

CAGE	Commercial and Government Entity Code
CPR	Contractor Performance Report
DoD	Department of Defense
DoDAAC	Department of Defense Activity Address Code
EAC	Estimate at Complete
EDA	Electric Data Access
GAK	Global Alignment Kernel
HIS	Human Systems Integration
IDA	Institute for Defense Analyses
IDATA	Institute for Defense Analyses Text Analytics
LSA	Latent Semantic Analysis
MDAP	Major Defense Acquisition Program
NDAA	National Defense Authorization Act
O&M	Operations and Maintenance
OUSD(A&S)	Office of the Under Secretary of Defense for Acquisition and Sustainment
PAUC	Program Acquisition Unit Cost
PIEE	Procurement Integrated Enterprise Environment
RDT&E	Research, Development, Test, and Evaluation
SAR	Selected Acquisition Report
SVC	Support Vector Classifier
SVM	Support Vector Machine
SVR	Support Vector Regression
tf-idf	Term frequency-inverse document frequency

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
xx-09-2020	Final	May 2019 - July 2020

4. TITLE AND SUBTITLE

Evaluating and Predicting Contract Performance Using Machine Learning: A Feasibility Study

5a. CONTRACT NUMBER

HQ0034-19-D-0001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Davis, Gregory, A.
DePriest, Travis, L.
Gladstone, Brian, G.
Hildreth, Laura, A.
Seitz-McLeese, Miranda, G.

5d. PROJECT NUMBER

AY-7-4626

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311-1882

8. PERFORMING ORGANIZATION REPORT NUMBER

IDA Document D-14275
H 20-000263

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of the Under Secretary of Defense (Acquisition and Sustainment)/Acquisition Analytics and Policy
Room 3B948A, The Pentagon
Washington, DC 20301

10. SPONSOR/MONITOR'S ACRONYM(S)

OUSD(A&S) / ASD(A) / DASD(AE) / AAP

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Office of Acquisition Analytics and Policy, which is part of Office of the Undersecretary of Defense for Acquisition and Sustainment (OUSD(A&S)), tasked the Institute for Defense Analyses with assessing the feasibility of using text analytics and machine learning to evaluate contracts and predict program performance. Although the machine-learning algorithms we evaluated could not predict program performance using the metrics we applied to the contracts, we were able to use text analytics to extract useful information from these contracts and similar documents. This report describes machine-learning algorithms that are feasible for exploratory and descriptive contract analysis. However, further research is necessary to establish whether it is reasonable to use this information, with additional data sources, to predict program performance.

15. SUBJECT TERMS

Major Defense Acquisition Programs; MDAP; oversight; contracts; procurement; machine learning

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Joseph, Brian
Unclassified	Unclassified	Unclassified	Same as Report	53	19b. TELEPHONE NUMBER *(Include area code)* (703) 697-8076

Reset