

Disclaimer: *The insights and analysis expressed in this paper are those of the authors and do not necessarily reflect the views, official policy, or position of Georgetown University or the Federal Government.*



GEORGETOWN UNIVERSITY
School of Continuing Studies

Historical Federal Procurement Record Analysis for Unsupervised Clustering Algorithms

GitHub: <https://github.com/georgetown-analytics/acquisitioners>

Georgetown University School of Continuing Studies
Cohort 26 — Capstone Project

Authors: Michelle McAllister, Richard Schreiber
Capstone Advisor: Adam Morris

1. Abstract

The Federal Government acknowledges the value of advanced analytics like machine learning and has a dedicated Data Science team to further capabilities for contract analysis. However, historical procurement data reporting is poor, data availability is limited, and not much groundwork has been completed. Therefore, the Acquisitioners team decided to run clustering models on government contract data to surface meaningful trends, patterns and/or segmentation within the existing historical record. The resulting models found clearly identifiable market segments that will be valuable to business development teams and strategic decision-makers in the product development offices.

2. Introduction

The goal of this project was to identify patterns in government spending habits through insights found from unsupervised clustering algorithms. The use of unsupervised learning was the best choice for this data because it lacked deep and clear categorization of different types of purchases. Several models were tested, but KMeans and natural language processing models proved to be the most effective.

A very specific question or problem is needed for supervised learning. The data set is general, and there were not any feasible questions to answer with supervised learning. Some supervised learning classification projects could include deeper category level tagging but are complicated by regulatory changes and relationships. The collected subset of FPDS data focuses on data specifically about the Federal Acquisition Service's Information Technology Category (aka FAS-ITC).

Due to these limitations, the team decided to use unsupervised learning models through two routes:

1. using models with one-hot encoded data
2. using text NLP models

Our hypothesis was that:

Unsupervised learning models will clearly identify spending trends and market segments in FAS-ITC's federal contracting activity.

3. Definitions

Obligations / Spend:

denotes a specified amount of money that is expected to be used towards a contract. Also called “spend.”

Department / Agency:

A federal agency of the executive branch of the United States. Agency can sometimes be used interchangeably with the word department. However, it is important to note that a department is the highest level of the hierarchy. An agency can be a sub-organization of a department and an office can be a sub-organization of an agency and a division can be a sub-organization of an office. In this experiment, the research team only focuses on the department level of the hierarchy, but will use agency and department interchangeably.

Vendor:

A private-sector business that is transacting with an agency.

Government Contracting

Local, State, and Federal Governments buy most of their supplies from the private-sector, a.k.a businesses. Just like you have credit cards, debit cards, cash, Venmo, and bitcoin wallets — agencies have different contract vehicles they can use. Several agencies have created contract vehicles that can be used by other agencies. This is similar to how a credit card agency makes one type of credit card that many of us can choose to use with our own spending limits.

Each row in the data will denote a time that an agency obligated money towards a purchase. This analysis will cover a subset of contracting data. The subset focuses on contract vehicles that were specifically made by the Federal Acquisition Service’s Information Technology Category (aka FAS-ITC). ITC is in charge of making contract vehicles that other agencies can use to purchase IT products and services.

Contract Vehicle

A contract vehicle is a product made by FAS-ITC. When program or product management is mentioned, it is referring to the management of the contract vehicle. FAS-ITC wants federal agencies/departments to utilize the contract vehicles they create. The more money a federal agency spends using a FAS-ITC contract vehicle, the better.

4. Architecture & Design

FIGURE 1. Original Design Schematic

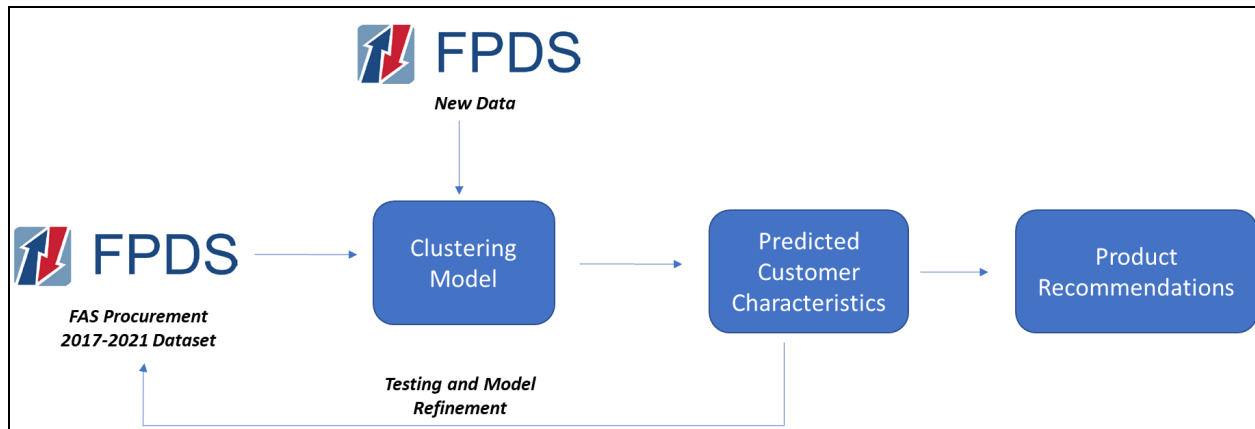
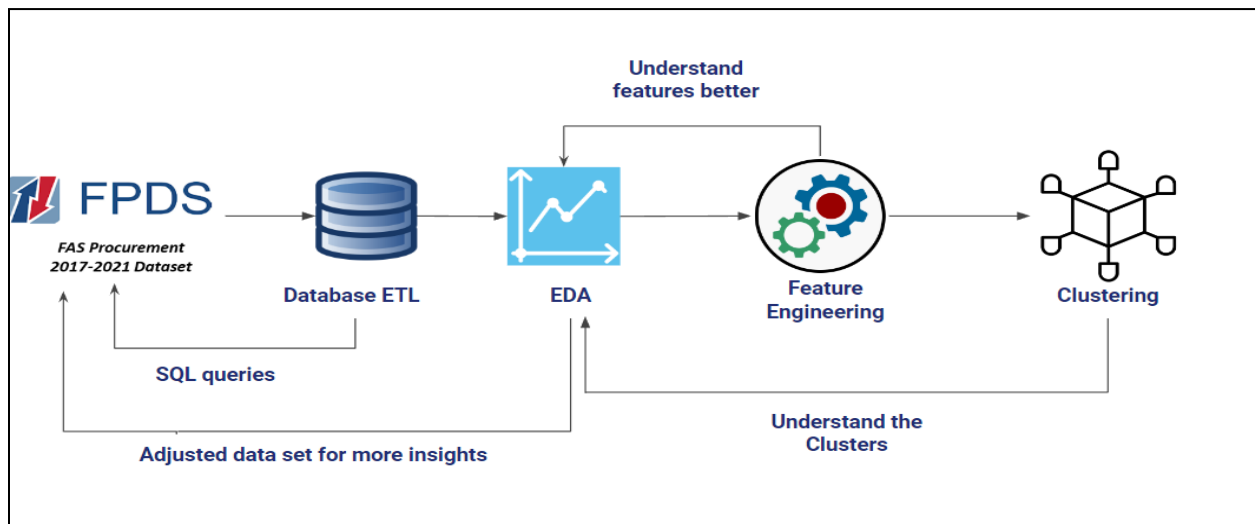


FIGURE 2. Adjusted Design Schematic



5. Data Source

I. Federal Procurement Data Set (FPDS)

What is reported to FPDS?

Contracts whose estimated value is over the micro-purchase threshold of \$10,000

Who reports data into FPDS?

Contract officers (federal employees) input the data during the acquisition process for the federal agency they work for.¹²

¹ https://www.fpds.gov/wiki/index.php/FPDS_FAQ

² https://www.fpds.gov/help_V1_1/Reportable_Nonreportable_Contract_Actions.htm

How can you access FPDS?

Civilians can utilize FPDS EZSearch and Sam.gov. Federal Employees have their own enriched versions of the data set stored internally through cloud services like AWS Redshift.

There is a lack of numeric data in the data set. Dollars Obligated is the only continuous variable. Naics_code is a categorical ID number. Everything else is categorical text data. This posed an issue for us when attempting to find statistical relationships between data elements.

II. ETL: Redshift, PostgreSQL, Datagrip

The team extracted a subset of data that only includes activity from ITC contract vehicles. The data is FPDS but is pulled from the Federal Acquisition Service internal redshift servers using SQL. The next challenge was setting up our own database so we could have an environment to transform the data and create new tables. We used AWS to set up a PostgreSQL server. The team used Datagrip to run SQL queries to the PostgreSQL database, transform and clean the data, and create tables for EDA.

6. Exploratory Data Analysis

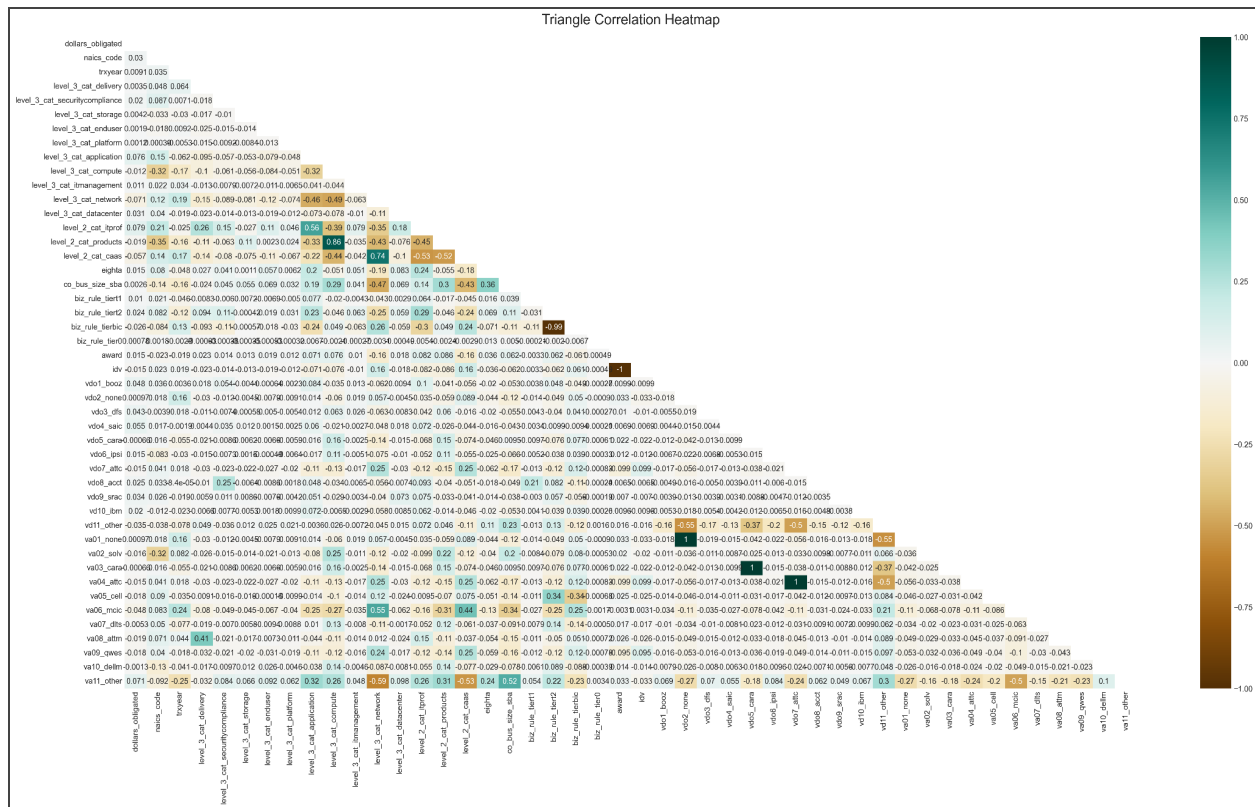
Each team member took two major steps to perform EDA.

- Analyze the total data set
- Analyze a specific agency

We did this because we wanted a general understanding of the data relationships first. Then, we wanted to see how specific agencies behaved based on spending habits (shown by the dollars_obligated column/field). The final value from these two approaches would show us which columns/fields seemed to be the most impactful/significant.

I. Correlation of Data Elements

FIGURE 3. Correlation Matrix Heat Map of Preliminary Feature Columns



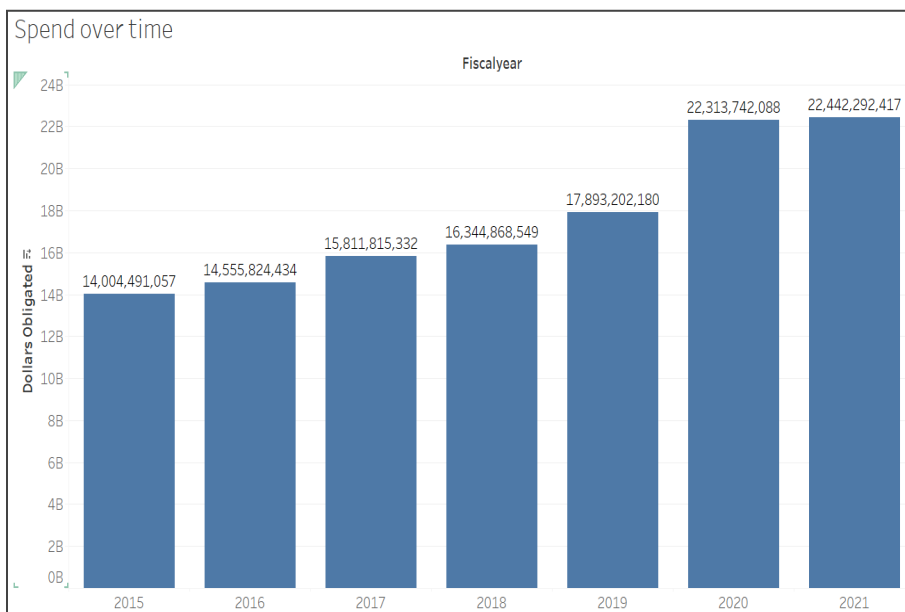
The correlation matrix (heatmap) shows that there are very few strongly correlated features. This led the team to focus on understanding the relationships between the columns/features with methods other than statistical analyses. Due to the lack of numeric data, the other categorical feature columns were paired up with the only continuous numeric column (dollars obligated) to identify trends.

II. EDA on Total Data Set

FIGURE 4. FAS-ITC's Contract Vehicles by Highest Dollars Obligated Overall Fiscal Years

Contract Name	F
Sched 70 HW SW	40,614,185,536
SCHEDULE 70 - INFORMATION TECHNOLOGY	34,430,180,596
ALLIANT	19,382,359,835
8(a) STARS II	6,346,750,916
Network/EIS	5,458,363,386
ALLIANT 2	5,083,442,926
ALLIANT SB	4,804,645,741
(PREDECESSOR) 8ASTARS2	1,956,443,856
CONNECTIONS II	1,054,372,834
MAS	982,394,012
COMSATCOM	896,180,210
AGENCY BPA / ORDER AGAINST SCHEDULE 70	803,491,674
CS2 CUSTOM SATCOM SERVICES	420,877,585
VETERAN TECHNOLOGY SERVICES II (VETS II)	392,822,535
VETERANS TECHNOLOGY SERVICES (VETS)	299,023,047
Wireless Mobility Solutions	104,864,668
AGENCY BPA / ORDER AGAINST SCHEDULE 8..	103,900,602
SCHEDULE 70: IT SERVICE DESK	75,588,291
CS2SB CUSTOM SATCOM SERVICES	66,956,866
TWO MULTI-ROLE TACTICAL COMMON DATA ..	66,447,062
8(a) STARS III	45,571,822
Salesforce	37,994,972
SCHEDULE 70 - INFORMATION TECHNOLOGY ..	14,623,386
UHF SATCOM ANTENNA SUBSYSTEM.	6,806,700
INFORMATION TECHNOLOGY SERVICES: SAL..	4,786,448
Enterprise Infrastructure Solution (EIS)**	2,424,898
SATCOM	-141,067
8ASTAR	-5,188,126
Millennia Lite	-6,105,910
Millennia	-20,774,828
CONNECTIONS	-57,054,416

FIGURE 5. FAS-ITC's Total Dollars Obligated by Fiscal Year



There are many contract vehicles and FAS-ITC wants to make sure they get the maximum number of people to use theirs. The graphs show FAS-ITC'S contract vehicles and the increase in dollars obligated over time. Figure 5 shows the total amount of dollars obligated per year. The total amount of dollars obligated for that year is generated from the sum of all the dollars obligated from each contract vehicle shown in Figure 4. Being able to identify a trend from these two fields led the team to choose contract_name (the data element depicting a contract vehicle) as a viable feature.

Contract vehicles are a lot like how people use credit cards or debit cards to purchase something. Not in the way a credit card uses debt but because it is the relationship between a

bank who issued the card and the customer that uses the card to purchase something from a vendor. ITC (and other agencies) have created similar products that federal agencies, offices, and departments can use to purchase things. Just like credit cards and banks, different contract vehicles have different pros and cons. Some have better-vetted vendors while others have faster customer service. This has created competition within the federal acquisition space between agencies that build contract vehicles (purchasing methods) for others to use. The value here is knowing that the `contract_name` field is important to the data set.

FIGURE 6. FAS-ITC’s Top Funding Departments Based On Dollars Obligated

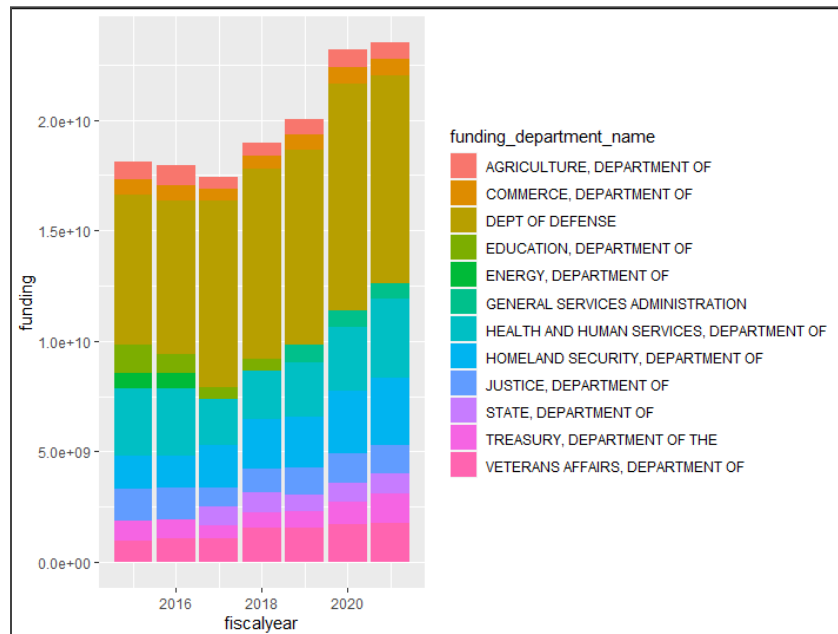


Figure 6 can show us the trends of which federal agencies and departments spend the most money with a FAS-ITC contract vehicle. The graph clearly shows that the Department of Defense (DoD) spends more with a FAS-ITC contract vehicle every single year. It is safe to say that the DoD is FAS-ITC's top customer year after year. Further analysis can lead to similar insights about what agencies spend more or less using a FAS-ITC contract vehicle.

III. EDA ON DEPARTMENT OF COMMERCE

FIGURE 7. Department of Commerce’s Top NAICs Through a FAS-ITC Contract Vehicle by Dollars Obligated

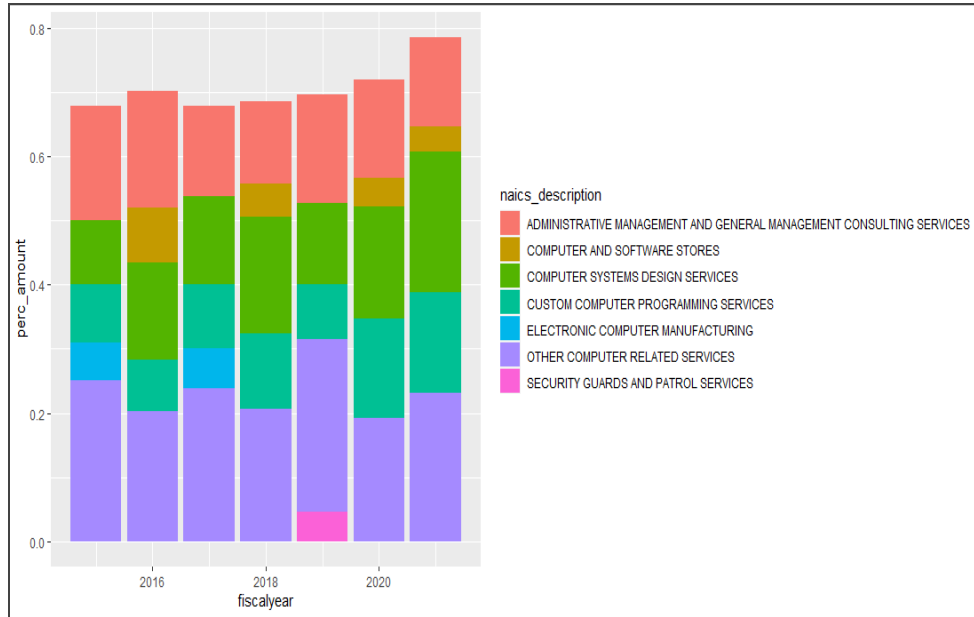
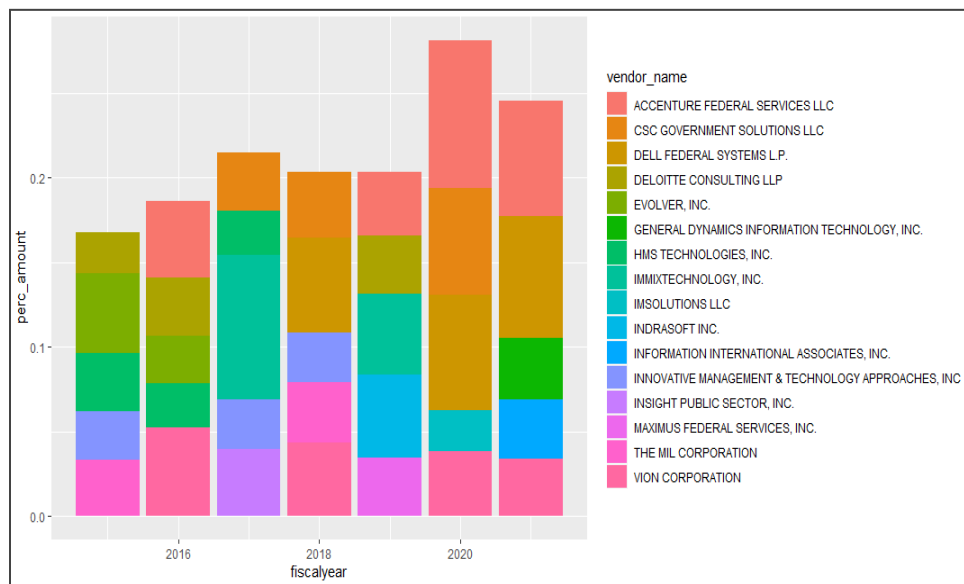


FIGURE 8. Department of Commerce’s Top Vendors Through a FAS-ITC Contract Vehicle by Dollars Obligated



Analyzing specific agencies is important because the agency is the customer of FAS-ITC. FAS-ITC wants an agency like the Department of Commerce to use its contract vehicles. Trends can be found by pairing the dollars obligated field with other fields. When trends like these can be found it helps FAS-ITC’s business development and program management teams understand their customers better. It is important to know what their top category of spending is and what private-sector business they buy from the most. With this information, FAS-ITC can create a strategy to capture more business with the customer agency and improve customer experience through its programs (contract vehicle).

These bar graphs show that these columns are significant when looking for trends.

- funding_department_name
- naics_description
- vendor_name

IV. EDA On Department Of Treasury

FIGURE 9. Treasury Spending Dashboard 1

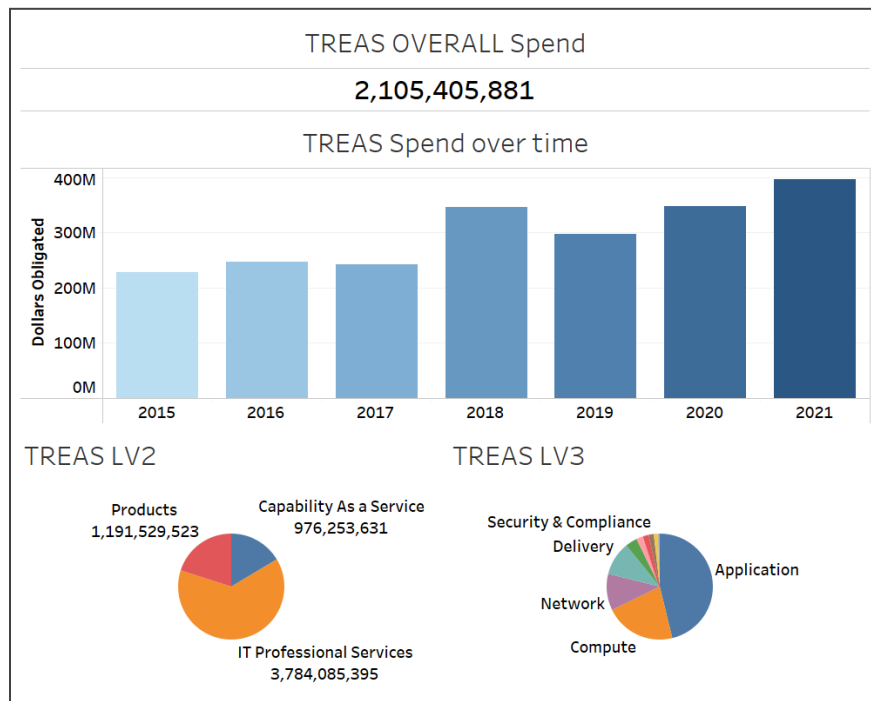


FIGURE 10. Treasury Spending Dashboard 2

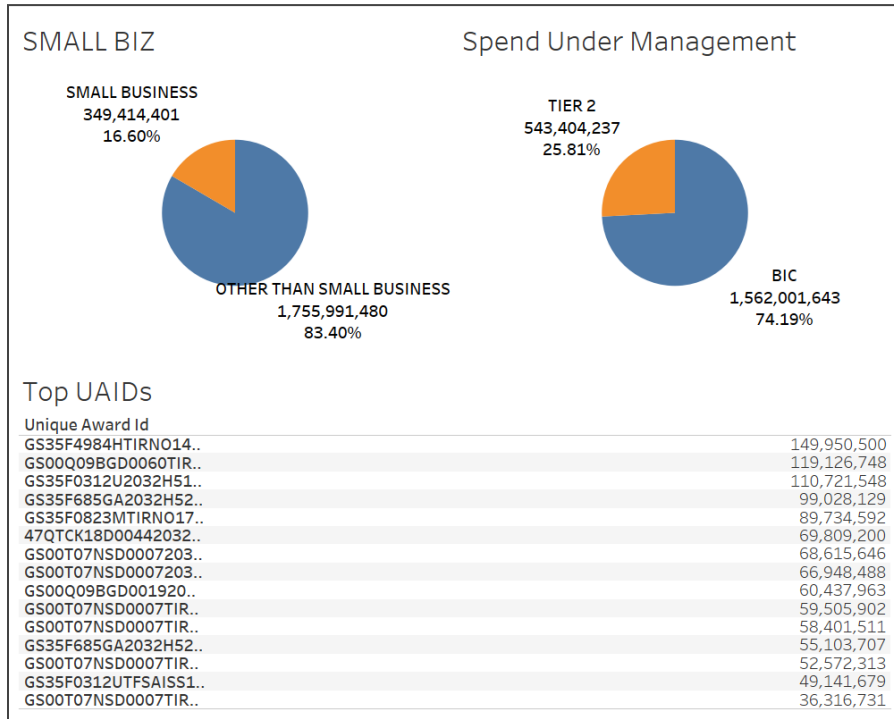
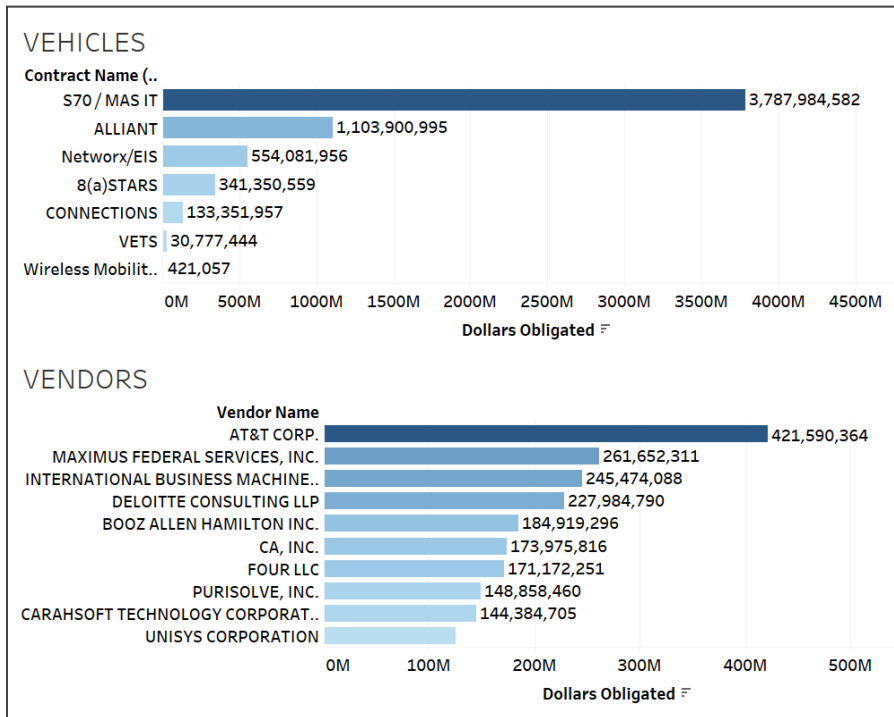


FIGURE 11. Treasury Spending Dashboard 3



These dashboards show a typical analysis of an agency that would take place in FAS-ITC. Overall spend, spend over time, level 2 and 3 categories, how much small biz, what contract tier, top unique award IDs, and what vehicles and vendors they use the most. The value here is showing how pairing dollars obligated with the categorical columns can paint a picture of how an agency spends its money. One agency may spend its money in one direction while the other agency spends it in another way. Knowing which data fields can paint that picture for us is important when choosing feature columns later on for machine learning.

V. Conclusion of EDA

The value from the EDA is what columns had the most significant and telling relationships. Below are all the columns/fields that we decided to focus on as potential features.

Column Name	Description	Data Type
funding_department_name	The name of the federal department obligating money (aka “spend”) towards a purchase.	object
naics_description	Description of the North American Industry Classification System code	object
fiscalyear	The fiscal year in which the purchase or contract action took place.	object
dollars_obligated	The dollar amount used by a funding department to purchase something from a vendor.	int64
co_bus_size_determination	Denotes if the vendor was a small business or not.	object
business_rule_tier	The tier that the contract vehicle falls under. Tier 0, Tier 1, Tier 2, and BIC are the possible values.	object
unique_award_ID	An ID that signifies a contract action. Usually deals with the addition or subtraction of dollars obligated. Can sometimes denote a change in the contract terms with no change in dollars obligated.	object
contract_name	The name of the contract vehicle used by a funding department/agency to purchase	object

After analysis, we decided to gather a revised dataset:

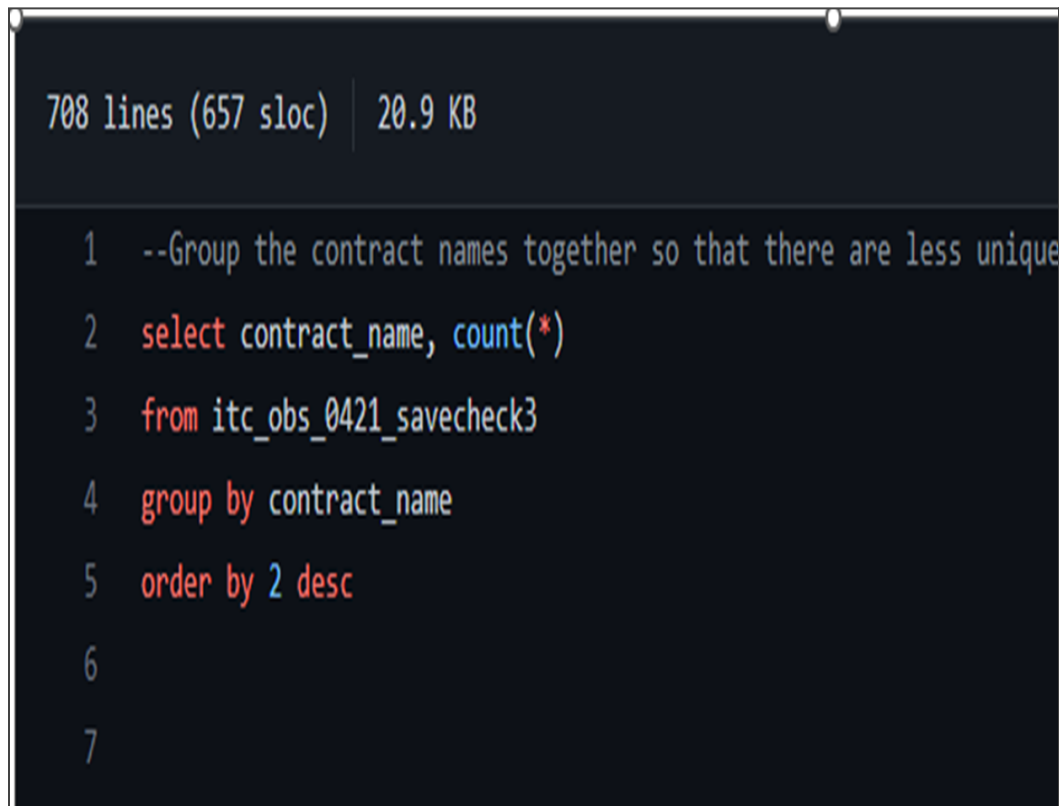
- Add contract transaction descriptions
- Limit transactions to those related to IT expenditures

The second data set posed several challenges while going through a second ETL process:

- New characters from the transaction descriptions caused the import process to fail and generate 117MB of error log files.
- Log files were inspected and error sources were identified and resolved.

Date data types were spoiled if the CSV was opened and closed. This was discovered after import to the database, so a copy of the original file was wrangled to keep the dates intact. One column in the CSV continued to fail the import process, so that data was imported to the database alone. It was joined in the database as a second step through a row id created in Excel.

FIGURE 14. DataGrip SQL Query Snapshot



```
708 lines (657 sloc) | 20.9 KB

1  --Group the contract names together so that there are less unique
2  select contract_name, count(*)
3  from itc_obs_0421_savecheck3
4  group by contract_name
5  order by 2 desc
6
7
```

Over 708 lines of code went into meeting the challenges described on the last slide. It gave us better data tables for feature engineering.

I. Feature Engineering: One-Hot Encoding

FIGURE 15. Snapshot of One-Hot Encoded Dataframe

level_3_cat_delivery	level_3_cat_securitycompliance	level_3_cat_storage	level_3_cat_enduser	...	Cn
0	0	0	0	...	
0	0	0	0	...	
0	0	0	0	...	
0	0	0	0	...	
0	0	0	0	...	

One-hot encoding was done in SQL and Pandas. Columns with few unique values are easily transformed into a boolean column with the original column dropped from the dataset within Pandas.³ Some of the data was a bit more complex and so the columns were created with SQL. For example, assigning top vendors to records required some pattern matching and validation that was easiest to accomplish in SQL.

The top 10 vendors for the highest number of unique awards and highest dollars obligated were identified and tagged with a series of integer columns containing 0 for false and 1 for true. This approach was used because the vendor_name column contained over 1,400 unique values so it was inappropriate for one-hot encoding in a dataframe or SQL.

The final result was a one-hot encoded data set to use for various clustering models. We intended to try many models.

Feature Selection:

Unsupervised clustering presents a few challenges for machine learning and one of them is feature selection. This project leveraged institutional knowledge to select features, but a mathematical method was also used to select features based on their silhouette score.⁴ This allowed us to explore a completely unbiased group of features based on statistical analysis. See Table “Highest Scoring Features based on Silhouette Score”.⁵

³

See: https://github.com/georgetown-analytics/acquisitioners/blob/main/GTSegment_Feature_selection.ipynb

⁴“Feature selection for K-means”.medium.com

<https://medium.com/analytics-vidhya/k-means-algorithm-in-4-parts-4-4-42bc6c781e46>

⁵ See also:

<https://github.com/georgetown-analytics/acquisitioners/blob/main/KMeans%20Feature%20Selection.ipynb>

FIGURE 16. 3D Model of Silhouette Scores of Feature Columns from One-Hot Encoded Data Set

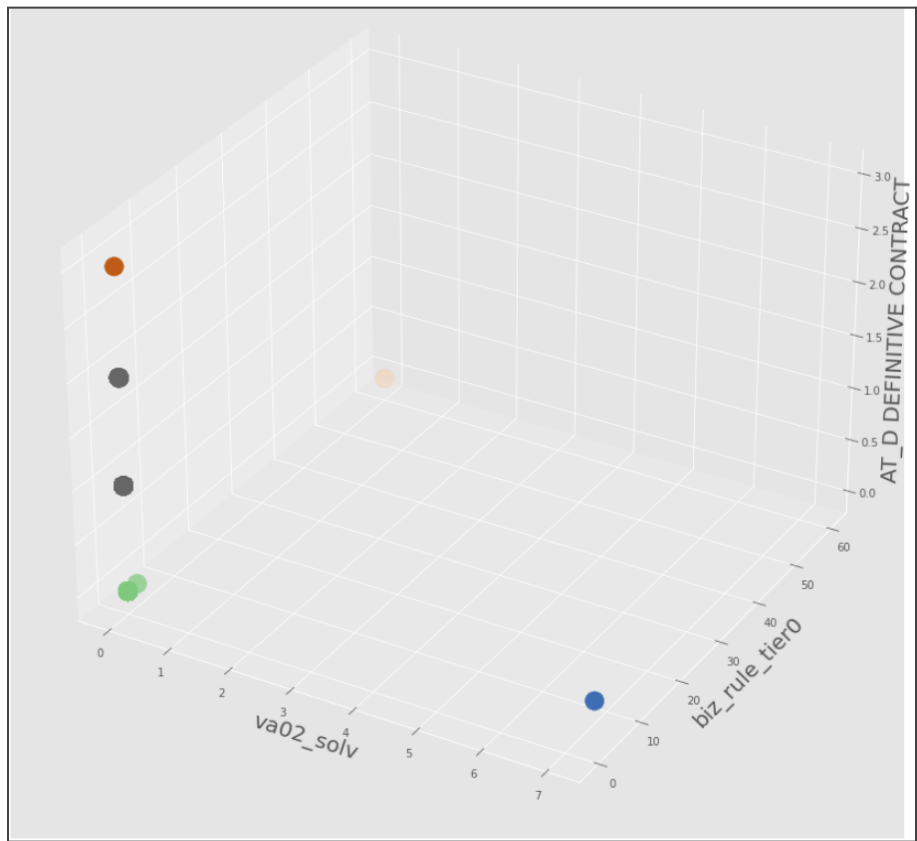


FIGURE 17. Highest Scoring Features based on Silhouette Score

feature	K	Best 3	K	Best 3	K	Best 3	K	Best 3
'naics_code'	5		6		7		8	✓
'dollars_obligated'	5		6		7		8	✓
'biz_rule_tier0'	5	✓	6		7		8	
'vdo9_srvc'	5		6		7	✓	8	
'va02_solv'	5	✓	6		7		8	
'va10_dellm'	5		6		7	✓	8	
'AT_BIDC'	5		6		7		8	✓
'AT_BPURCHASEORDER'	5		6	✓	7		8	
'AT_DDEFINITIVECONTRACT'	5	✓	6		7		8	
'Cnr_MILLENIAL'	5		6	✓	7		8	
'Cnr_SALESFORCE'	5		6	✓	7		8	
'Cnr_VETTECHSERV'	5		6		7	✓	8	

II. Feature Engineering: NLP

FIGURE 18. Snap Shot of Code Creating Requirements Column

```
df = df.astype({'level_2_category': str})
df = df.astype({'level_3_category': str})
df = df.astype({'co_bus_size_determination': str})
df = df.astype({'contract_name': str})
df = df.astype({'level_3_category': str})
df = df.astype({'description_of_requirement': str})

df["req"] = df[['description_of_requirement', 'level_2_category', 'level_3_category', 'co_bus_size_determination',
               'business_rule_tier', 'contract_name']].apply(lambda x: ' '.join(x), axis = 1)

df['req'].dtypes

dtype('O')
```

```
#Convert Column to all lower case, strip punctuation marks
df["req"] = df.req.str.replace(',', '')
df["req"] = df.req.str.replace('.', '')
df["req"] = df["req"].str.lower()
```

FIGURE 19. Snap Shot of Resulting Data Frame with Requirements Column

df.head()							
	dollars_obligated	description_of_requirement	level_2_category	level_3_category	co_bus_size_determination	business_rule_tier	contract_name
0	0.0	DHS BULK CLOSE OUT	Capability As a Service	Network	OTHER THAN SMALL BUSINESS	TIER 2	SCHEDULE 70 - INFORMATION TECHNOLOGY
1	0.0	ITAS SUPPORT SERVICES	IT Professional Services	IT Management	SMALL BUSINESS	TIER 2	SCHEDULE 70 - INFORMATION TECHNOLOGY
2	0.0	WIRELESS SERVICE	IT Professional Services	Network	OTHER THAN SMALL BUSINESS	TIER 2	SCHEDULE 70 - INFORMATION TECHNOLOGY
3	0.0	PERFORMANCE PERIOD: 10/24/18-09/30/19DOJ FBI ...	Capability As a Service	Network	OTHER THAN SMALL BUSINESS	TIER 2	SCHEDULE 70 - INFORMATION TECHNOLOGY
4	0.0	IGF::OT:IGF VERIZON WIRELESS FY19 RMB/ IOD	Capability As a Service	Network	OTHER THAN SMALL BUSINESS	TIER 2	SCHEDULE 70 - INFORMATION TECHNOLOGY

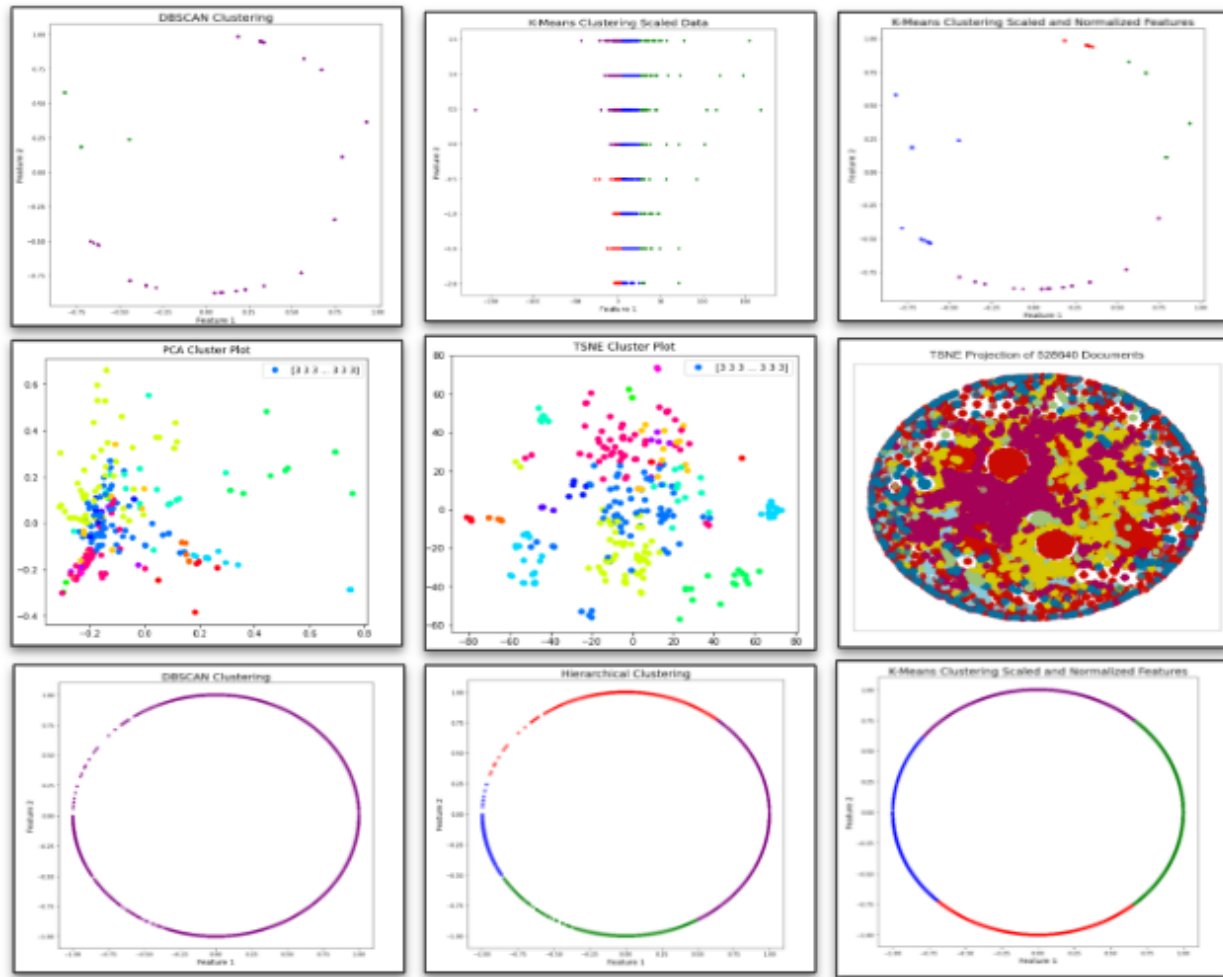
This code creates one column that contains all the normalized text from the feature columns in the entire row. This column called “req” will be used for the NLP models the team attempted. When using natural language processing and a TFIDF-Vectorizer, there is a need to tokenize specific groupings of texts called “documents.” Each row in this data frame now has a cell under the “req” field that acts as the document needed for tokenization. Once the documents are tokenized, they are in the proper format to be used in machine learning models.

This was a unique way to get around the lack of numeric and continuous data. The team could have chosen to use only the description_of_requirement field as the document for tokenization. However, this would leave out all the rest of the chosen feature columns. This

solution allows the strings from all the feature columns to be included in the model. The team believed this would create a more accurate and robust result in the end.

8. Machine Learning

FIGURE 20. Cluster Visualization Matrix



These visualizations represent almost all the models created. The middle visuals were the results of NLP while the others are from one-hot encoding.

I. NLP Models

FIGURE 21. Snap Shot of Code Creating a Pipeline For TFIDF-Vectorization

```
pipeline = Pipeline([
    ('vect', TfidfVectorizer(tokenizer = word_tokenize, stop_words=stop))
])
```

FIGURE 22. Snap Shot of Code Fit and Transforming the Req Field

```
# Assign the column I want to use as req then run it through the pipeline,  
  
req = df['req']  
X = pipeline.fit_transform(req)  
X.shape
```

FIGURE 23. Elbow Plot Determining Ideal Number for “K”

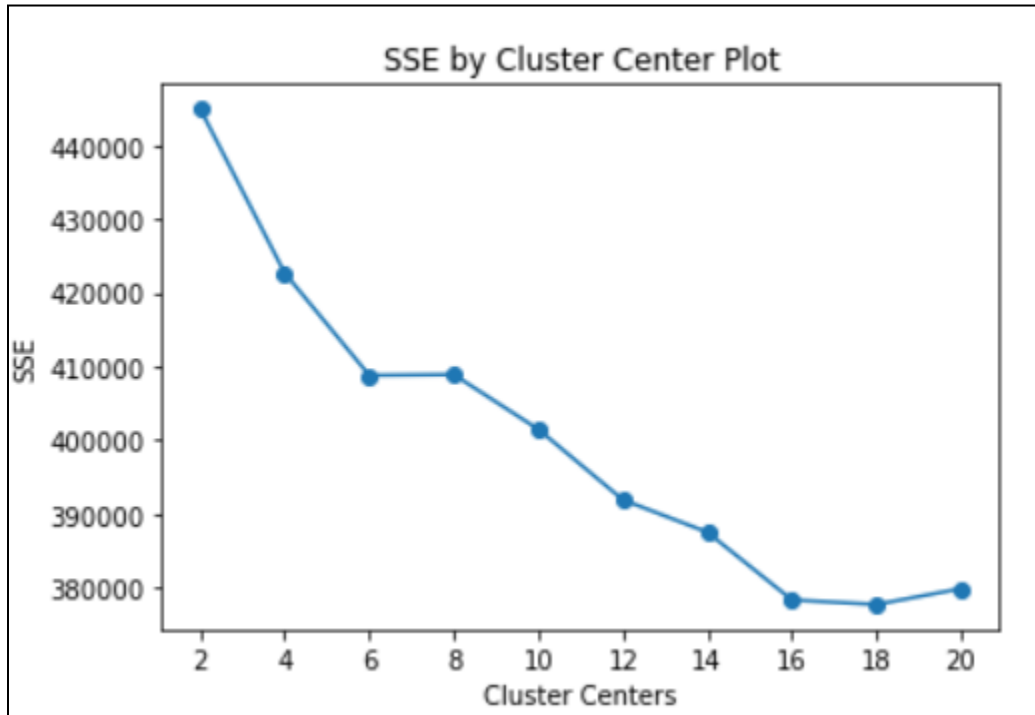


FIGURE 24. Snapshot of MiniBatch Kmeans Partial Fitting

```
# manually fit on batches
kmeans = MiniBatchKMeans(n_clusters=10,random_state=0,batch_size=60000)

kmeans = kmeans.partial_fit(X[0:60000,:])
kmeans = kmeans.partial_fit(X[60000:120000,:])
kmeans = kmeans.partial_fit(X[120000:180000,:])
kmeans = kmeans.partial_fit(X[180000:240000,:])
kmeans = kmeans.partial_fit(X[240000:300000,:])
kmeans = kmeans.partial_fit(X[300000:360000,:])
kmeans = kmeans.partial_fit(X[360000:420000,:])
kmeans = kmeans.partial_fit(X[420000:480000,:])
kmeans = kmeans.partial_fit(X[480000:540000,:])

print(datetime.datetime.now() - now)

0:01:29.063973

labels = kmeans.predict(X)
labels

array([4, 5, 4, ..., 5, 5, 5])
```

FIGURE 25. Snap Shot of Code That Attaches Resulting Cluster Label to Corresponding Row

```
# Add clusters label to DF
df['clusters'] = labels
df.head()
```

ars_obligated	description_of_requirement	level_2_category	level_3_category	co_bus_size_determination	business_rule_tier	contract_name	req	clusters
0.0	DHS BULK CLOSE OUT	Capability As a Service	Network	OTHER THAN SMALL BUSINESS	TIER 2	SCHEDULE 70 - INFORMATION TECHNOLOGY	dhs bulk close out capability as a service net...	4
0.0	ITAS SUPPORT SERVICES	IT Professional Services	IT Management	SMALL BUSINESS	TIER 2	SCHEDULE 70 - INFORMATION TECHNOLOGY	itas support services it professional services...	5
0.0	WIRELESS SERVICE	IT Professional Services	Network	OTHER THAN SMALL BUSINESS	TIER 2	SCHEDULE 70 - INFORMATION TECHNOLOGY	wireless service it professional services netw...	4
0.0	PERFORMANCE PERIOD: 10/24/18-09/30/19DOJ FBI ...	Capability As a Service	Network	OTHER THAN SMALL BUSINESS	TIER 2	SCHEDULE 70 - INFORMATION TECHNOLOGY	performance period: 10/24/18-09/30/19doj fbi ...	5
0.0	IGF::OT::IGF VERIZON WIRELESS FY19 RMB/ IOD	Capability As a Service	Network	OTHER THAN SMALL BUSINESS	TIER 2	SCHEDULE 70 - INFORMATION TECHNOLOGY	igf::ot::igf verizon wireless fy19 rmb/ iod ca...	4

The code to create the NLP models required fit and transforming the requirement column through the TFIDF-Vectorizer, using the tokens to create an elbow plot for the optimal number for K (since the

model used for the NLP models is Mini Batch Kmeans), and then applying them to which ever model sequence was being tested.

The first model tested was Mini Batch Kmeans on its own. The next interaction used PCA and T-SNE visualizations with the Mini Batch Kmeans. The final iteration used Yellowbrick's T-SNE Visualizer models with Mini Batch Kmeans. The resulting clusters of the Yellowbrick T-SNE Visualizer model are analyzed later in this project.

II. *One-Hot Encoded Models*

FIGURE 26. Snap Shot of Code Showing Scaling of Data

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(x1)

fmX = pd.DataFrame(X_scaled)

ax = sns.boxplot(data=fmX)
ax
```

FIGURE 27. Snap Shot of Code Showing Normalization of Scaled Data

```
# Normalizing the data so that
# the data approximately follows a Gaussian distribution
X_normalized = normalize(X_scaled)

# Converting the numpy array into a pandas DataFrame
df_X_normalized = pd.DataFrame(X_normalized)
```

FIGURE 28. Plot of Scaled Data

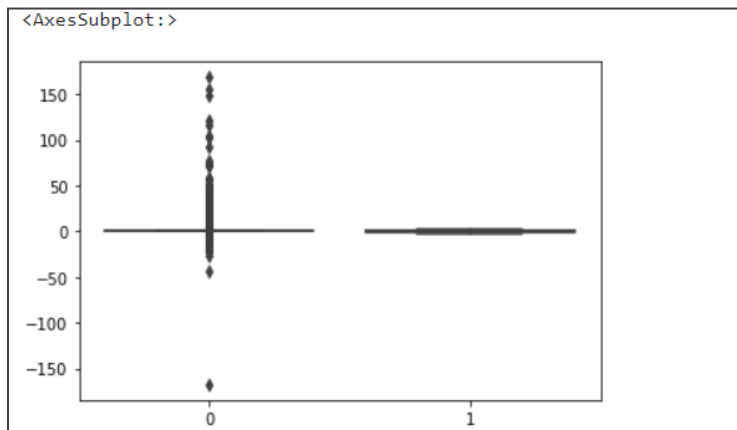


FIGURE 29. Plot of Normalized and Scaled Data

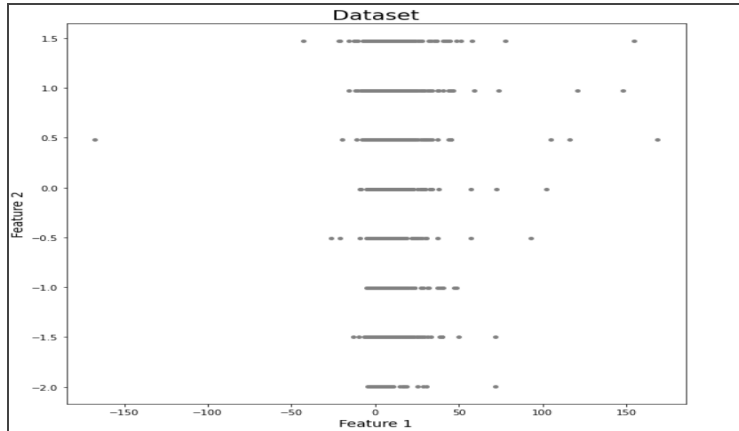


FIGURE 30. Snap Shot of Code Using Normalized and Scaled Data in Kmeans Model

```
#start with KMeans cluster of the 2
k_means_norm=KMeans(n_clusters=4,random_state=42)
k_means_norm.fit(df_X_normalized[[0,1]])
df_X_normalized['KMeans_labels']=k_means_norm.labels_
```

FIGURE 31. Snap Shot of Code Using Hierarchical Clustering Model

```
#Look at agglomerative - selected a very small sample since memory error
#running this took 80% of memory -any more pings my pc
from sklearn.cluster import AgglomerativeClustering
model = AgglomerativeClustering(n_clusters=4, affinity='euclidean')
model.fit(df_samp[[0,1]])
df_samp['HR_labels']=model.labels_
```

FIGURE 32. Snap Shot of Code Using DBSCAN Clustering Model

```
#finally DBScan, though both heirarchical and KMeans did pretty well
from sklearn.cluster import DBSCAN
dbscan=DBSCAN()
dbscan.fit(df_samp[[0,1]])
df_samp['DBSCAN_labels']=dbscan.labels_
```

9. Results

I. Analysis of Clusters from Yellowbrick TSNE-Visualizer Model

The following dashboards and visuals are an analysis done on the resulting clusters created from the Yellowbrick TSNE-Visualizer model. Each visual shows how the break down of the clusters in different ways by pairing them in graphs with dollars obligated and other feature columns.

FIGURE 33. Dashboard of Clusters by Dollars Obligated by Feature Columns

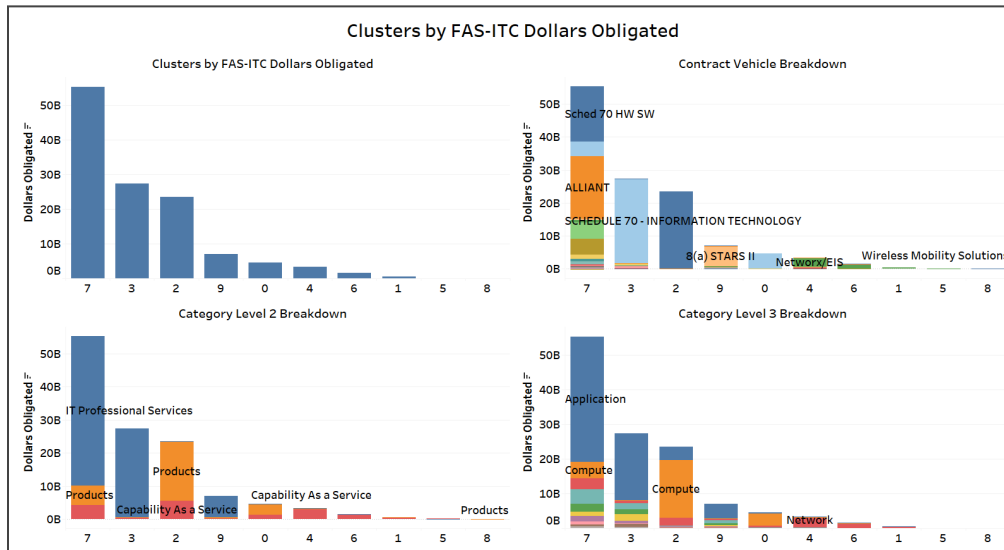


FIGURE 34. Dashboard of Dollars Obligated by Fiscal Year by Cluster Number

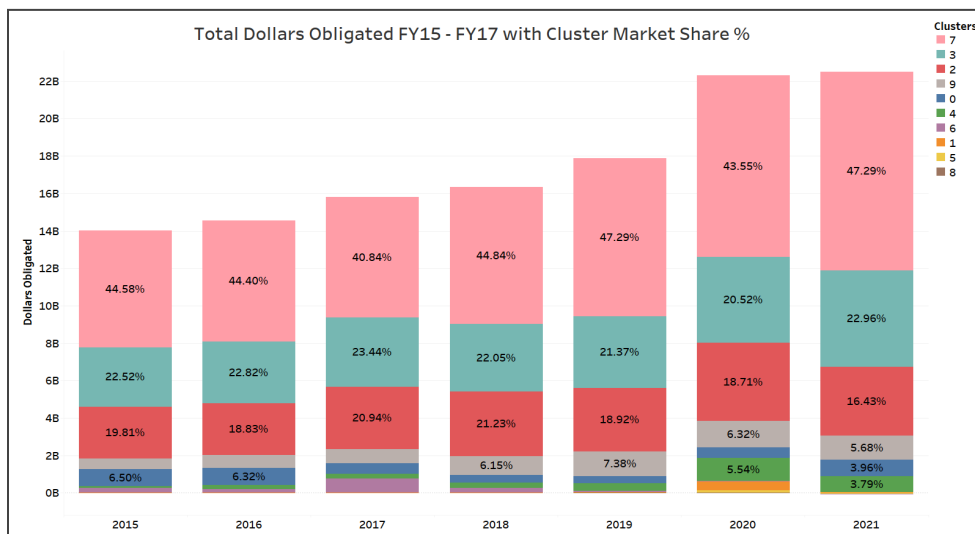
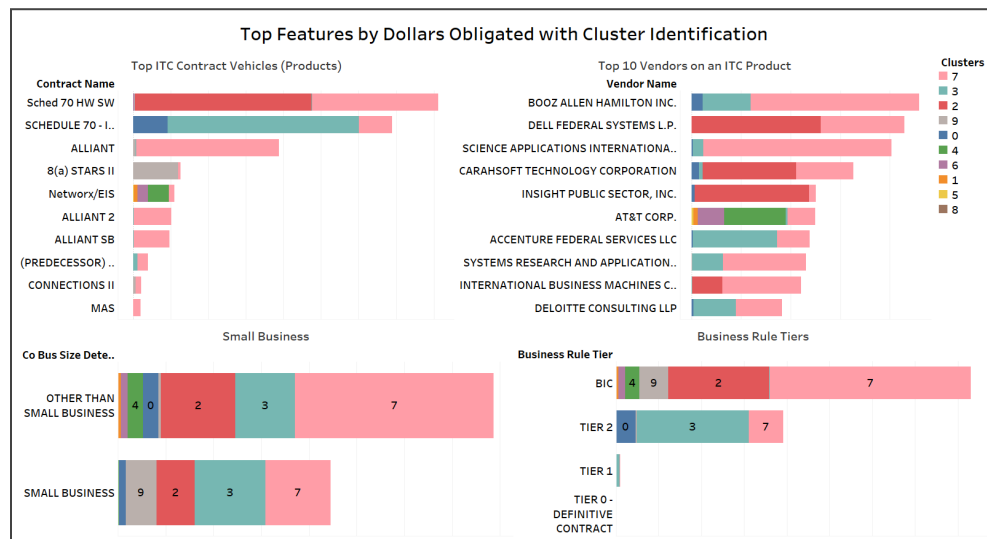


FIGURE 35. Dashboard of Feature Columns by Dollars Obligated by Cluster Number



II. Cluster Categories and Names Based on Analysis and Contents of Cluster

Market Movers

Cluster 7: ITC's Bread and Butter (BIC Alliant and S70 Spend)

- 85% BIC / 14% Tier 2
- 75% not SB/ 25% SB
- 52% Alliant / 30% a S70 HW/SW
- Well distributed vendors and agencies

Cluster 3: Tier 2 S70

- 96% Tier 2
- 93% through S70 - IT
- 98% IT Professional Services

Cluster 2: S70 Hardware

- 99% Schedule 70 HW/SW
- 76% Products / 23% CaaS
- 71% Compute, 16% Application

Telecom Clusters

Cluster 1 & 5: Telecom Ghostssector

- Fade in and out with low value
- All 517110
- Networx Contract

Cluster 6: Dead DoD Telecom

- Faded away from FY15-FY18, maybe a dead market
- 87% NAIC 517110
- Networx Contract

Cluster 4: Emerging Telecom

- noticeable growth over time, possible emerging market
- 83% in NAIC 517110
- Networx Contract

Random Clusters

Cluster 8: Standing Desktops

- Descriptions about height-adjustable, free standings desktops

Cluster 9: DoD SB Professional Services

- 99% Small Biz
- Uses mainly 8aSTARS II.
- Mainly IT Professional Services Lvl 2 Category

Cluster 0: S70 - IT Products/Compute

- tagged as lvl 2 Products and lvl 3 compute, but uses a S70 - IT (instead of S70 HW/SW like usual hardware purchases).

10. Conclusion

I. Hypothesis:

The unsupervised learning models were able to clearly identify spending trends and market segments in FAS-ITC's federal contracting activity.

II. Lessons learned

- Clustering is useful when attempting to understand chaotic data
- Be prepared to wrangle

III. Next Steps

- Deeper analyses of interesting clusters for business development
- Selecting product or customer sectors to run supervised learning on for deeper tagging or product recommendation