GEORGE MASON UNIVERSITY

Data Analytics Engineering

Fall 2021

Determining Noise and Actuals in the Sightings/Incidence of UAS from FAA Databases

Samuel Razia

John Brzezinski

Joey Rainey

Lahari Tadepalli

Hermella Tessema

George Mason University

12/9/2021

DAEN 690 Project Report

This Page Intentionally Left Blank

Table of Contents

Abstract

The Federal Aviation Administration (FAA) regulates the US internal and adjacent civil aviation operations and safety. With the rapid growth of both private and commercial aviation critical to the US economic expansion, and the rapid technological growth of unregulated Airborne devices, aviation safety incidents and potential threats have increased. This project focused on the growing threats of Unmanned Aircraft systems (UAS) incidents. Specifically, the FAA seeks to leverage various data reporting various “noise” and related observations to discover UAS safety violations. We applied analytical and machine learning methods to merge, analyze, model, and visualize data on Tableau. We used an NLP similarity testing model to cross-validate datasets from multiple sources shared by the FAA and NASA. The similarity testing allowed us to score over 13,000 records between the two datasets and reasonably confirm 85 reports to be actual UAS incidents. We trained and tested predictive modeling with our cleaned and cross-validated data set, which allows for new reports to be tested and classified as actual or noise. We had less than 100 records out of over 13,000 that we decided to label as actual UAS incident reports. For our model training purposes, the remaining records would have all been labeled as noise. We expect this ratio severely underestimated numbers of actual UAS incidents to be reported. We recommend additional reports verification in the dataset and that the updated dataset be used in our predictive model pipeline with additional NLP models.

1 Introduction

1.1 Background

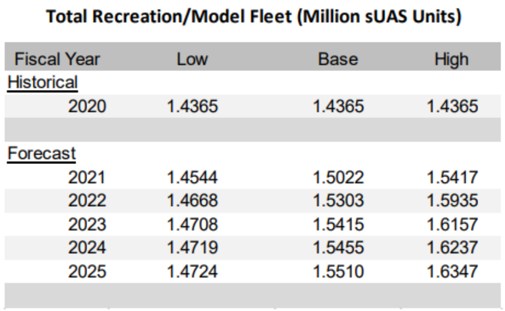
The Federal Aviation Administration (FAA), being the country's most prominent government/transport organization, was established on August 23, 1958, and is accountable for upholding guidelines regarding airplane tasks and support [1]. The exponential growth in Unmanned Aircraft Systems (UAS), or drones, used for commercial and recreational purposes, has posed significant risks, especially when sharing the open sky with manned aircraft. To prevent risks, the Federal Aviation Administration monitors the skies to ensure that manned aircraft and UAS can operate simultaneously without incident.

Figure 1-1. Example of vulnerability of manned aircraft to unmanned aircrafts systems (UAS).

The FAA categorizes drones into four classes: Recreational Flyer, Certificated Remote Pilot or Commercial Operator, Public Safety or Government User, and Educational User.

- Recreational Flyers fly without financial compensation or volunteer work, purely for leisure. These pilots must take the Recreational UAS Safest Test (TRUST).
- Certified remote pilots or commercial operators are flying UAS for compensation or indirect compensation. They must be certified under the FAA remote certification program, which is more rigorous than the TRUST certification.
- Public Safety/Government users are piloting for public safety and law enforcement agencies.
- Educational users fly under the same rules as recreational flyers [2].

According to the Aerospace Forecasting Trends of 2021 to 2041 regarding UAS trends, the FAA reported that almost 1.14 million recreational UAS owners had registered with the FAA. The agency reported the past known trends as shown in the graph of Figure 1-2 below and predicts these numbers will increase to the numbers found in the table of Figure 1-2 within the upcoming years [3].



Total Recreation/Model Fleet (Million sUAS Units)

Fiscal Year	Low	Base	High
Historical			
2020	1.4365	1.4365	1.4365
Forecast			
2021	1.4544	1.5022	1.5417
2022	1.4668	1.5303	1.5935
2023	1.4708	1.5415	1.6157
2024	1.4719	1.5455	1.6237
2025	1.4724	1.5510	1.6347

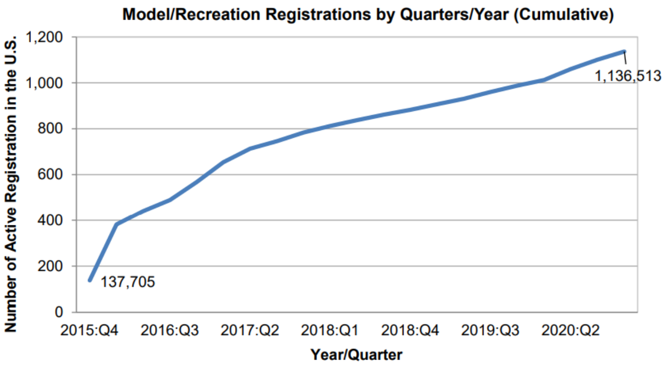
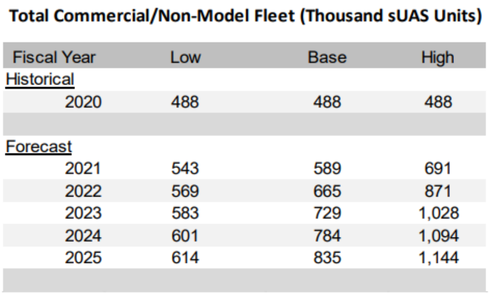


Figure 1-2. (left) Table depicts the forecasted number of recreational UAS models in millions on market from 2020 to 2025. (right) Line graph depicting the Number of Active Registration of recreational UAS in the U.S from 2015Q4 to 2020 Q2 [3].

The FAA also reported that 488,000 commercial UAS registered by the end of 2020. Figure 1-3 below shows FAA's forecast of an increase in the commercial UAS fleet from 2020 to 2025, similar to the past five years. [3]



Total Commercial/Non-Model Fleet (Thousand sUAS Units)

Fiscal Year	Low	Base	High
Historical			
2020	488	488	488
Forecast			
2021	543	589	691
2022	569	665	871
2023	583	729	1,028
2024	601	784	1,094
2025	614	835	1,144

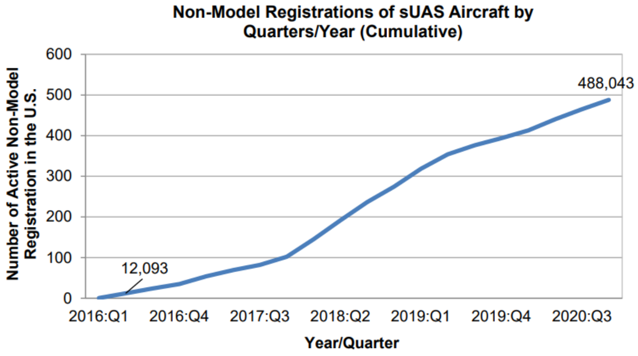


Figure 1-3. (left) Table depicts the forecasted number of commercial UAS models in thousands on market from 2020 to 2025. (right) Line graph depicting the Number of Active Registration of commercial UAS in the U.S from 2015Q4 to 2020 Q2 [3]

As of November 1, 2021, the FAA has reported 865,607 UAS as registered, where 339,794 are commercial drones, 522,226 are recreational, and 3,591 are paper registrations. The FAA has also reported that 251,322 UAS pilots are Remote Certified, while 144,182 have completed TRUST certifications issued by test administrators. [4]

UAS usage has seen exponential growth in the last few years in market size, and according to Business Insider [5], the UAS market size is expected to grow to $63.6 billion by 2025. Insider Intelligence also predicts that by 2023, total global shipments for enterprise UAS, those sold directly to a business for use, will reach 2.4 million, increasing at a 66.8% compound annual growth rate (CAGR). Along with the enterprise UAS market, UAS for recreational uses has also seen significant growth in the past few years, especially with new models having lower costs. Business Wire predicts the market for recreational UAS, also referred to as consumer UAS, is expected to reach $3.26 billion in 2025, at a CAGR of 9% [6]

The unprecedented growth in the UAS market with options for several recreational and commercial purposes led to growth in dangerous air-born incidents. There are several examples of UAS incidents in recent years; one such incident is the crash of a UAS on the lawn of the White House in January of 2015 [7], which was the result of a UAS flight operated by a government officer. Although the incident caused no damage, this did raise the issue of small UAS not being able to be detected by security radars, which were designed to detect aircraft and missiles at the White House.

A UAS strike on a commercial plane in Canada in 2017 is another example of a significant UAS incident that sparked safety concerns and conversations. A UAS hit a light engine aircraft carrying eight passengers at an altitude of 1500 feet while the aircraft was landing. The aircraft only sustained minor damage and could land successfully [8]. Although the damage was minor in this case, it is essential to note in collisions like these, there could have been more damage to the larger aircraft than the drones.

In August 2021, a Canadian flight school Cessna 172 aircraft reported a midair collision with police department operated UAS that damaged the aircraft, seen in Figure 1-4 below [9]

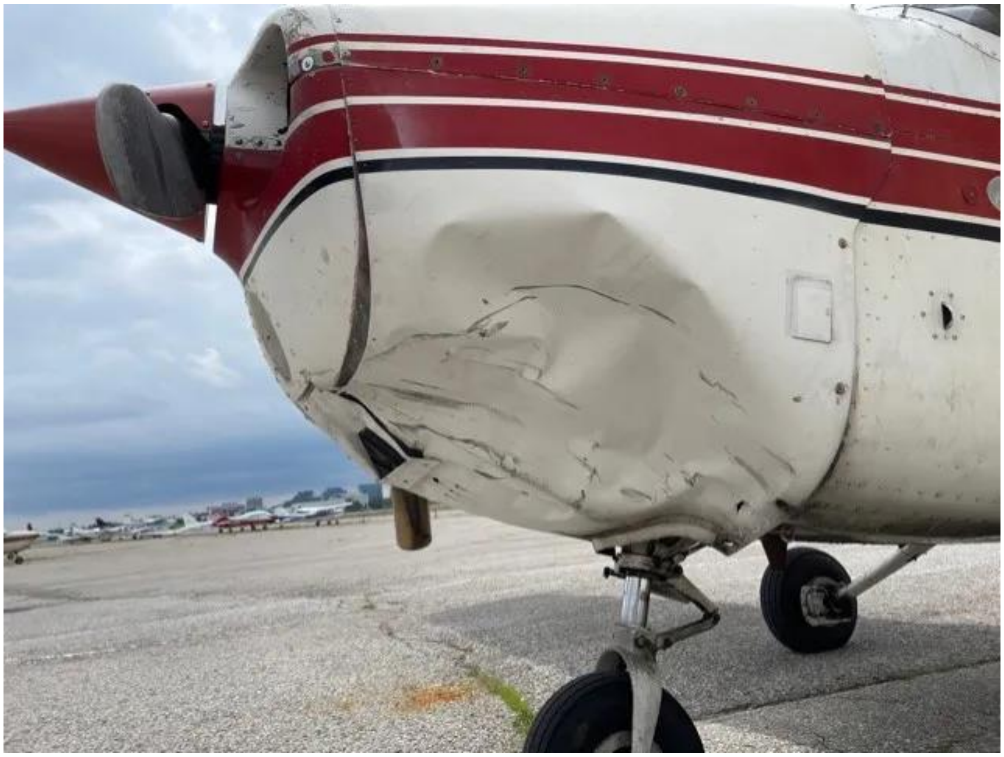


Figure 1-4. Cessna 172 damaged by police drone at Buttonville Airport in August 2021 [9].

According to research conducted in 2018 by the University of Dayton Research Institute [10], larger aircraft do not always have the upper hand in collisions with UAS. In 2018, a commercial helicopter pilot, John Marking, reported a collision between his Army Blackhawk helicopter and a UAS that caused minor damage and dents to the surface and rotor of the helicopter. Although the helicopter was able to return home safely, Marking highlighted how violent the collision was in shaking the helicopter and how dangerous it could have been had the UAS hit a critical part of the helicopter, such as the tail rotor blades, which may have caused the pilot to lose control [11]. The Dayton Research Institute study group in the 2018 study, following Marking's incident, conducted research where they launched a 2.1-pound DJI Phantom 2 UAS at the wing of an M20 aircraft which resulted in significant damage to the wing, tearing into the structure of the wing, while the group leader, Kevin Poorman, reported that the UAS did not shatter (see Figure 1-5 below). Concluding the study, Poorman emphasized the need to regulate UAS operations and the building material of UAS to make them more breakable to enhance safety [10].

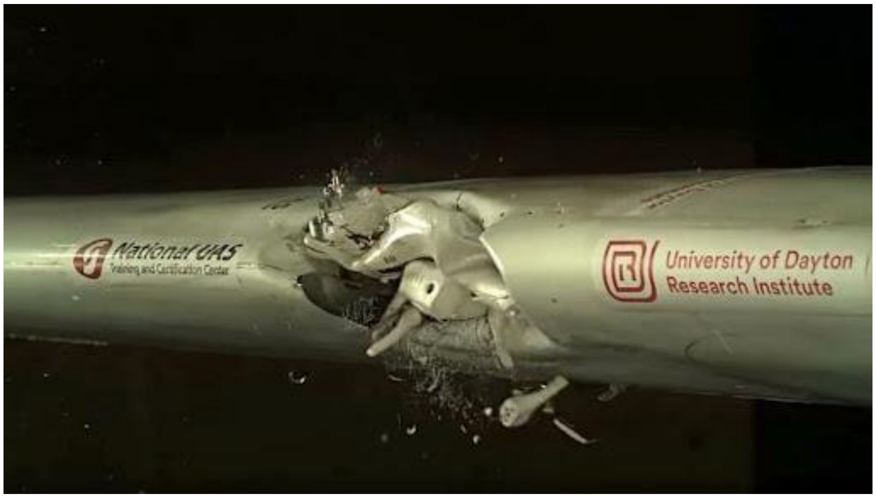


Figure 1-5. Dayton Research Institute's Impact Physics Lab show small drones pose a risk to manned aircraft. Impact tests prove large aircraft won't always win in collision with drones [10].

UAS sightings near airports create an even more vital threat to safety. In December 2018, London's Gatwick Airport was unable to operate correctly for more than a day due to reports of several UAS in the area. The appearance of UAS caused significant issues in flight schedules where over a thousand flights were canceled, affecting 140,000 passengers. A few weeks later, in January 2019, London's Heathrow Airport halted departures temporarily, for about an hour, after reports of UAS sightings. Gatwick Airport reported another incident in March 2020 regarding a near collision with a UAS so close to a Boeing 747, that a flight attendant could see and recognize the UAS model [12].

Another instance of UAS disruptions around airports occurred in 2020 as a near-miss collision with an aircraft carrying 186 passengers at Manchester Airport, later classified as a Category A incident, indicating a severe risk of collision. Other incident examples reported in 2020 include a UAS sighting at Frankfurt Airport, which caused the airport to stop operating for two hours, delaying several flights, and one in Perth Airport of Scotland, which reported a near-miss accident of a UAS that flew into a restricted area and almost collided with a light aircraft [12].

The FAA currently has several regulations to avoid such issues and regulate UAS operations in general. One such rule for regulating Small UAS is Part 107. Part 107 rules for commercial and government UAS are set in place for UAS weighing less than 55 pounds and has a list of requirements and rules set in place. UAS are allowed to be operated during daylight or in twilight if the UAS has anti-collision lights. A UAS is not allowed to operate more than 400 feet above ground or structure or with a speed of more than 100 mph. Some more of these operating requirements include:

- Avoiding manned aircraft, not operating carelessly
- Keeping UAS within sight
- Operating or visually observing only one UAS at a time
- Avoiding flying UAS over people unless they are directly involved in the operation
- Avoid operating UAS around a moving vehicle or aircraft unless operating in a not densely populated area.[13]

Commercial and Government UAS under Part 107 are also required to register each drone they operate. A registration number provided must be visible on the UAS. When it comes to UAS registration, the FAA mandated that all recreational UAS weighing more than 0.55 pounds but less than 55 pounds register online as of December 2015. Similarly, the FAA has made it mandatory for every commercial or non-model UAS to register online as of April 2016. Commercial UAS owners must register for each; however, recreational UAS owners do not need to register each UAS [13].

Under Part 107, a UAS operator must have a Remote Pilot Certificate that provides the operator with an understanding of regulations and safety procedures set by the FAA. Certificate holders must pass an initial knowledge exam and complete online training every two years to maintain aeronautical knowledge. Under Part 107, operators are required to make their UAS available for inspection by the FAA and report to the FAA any incidents that result in serious injury, damage, or loss of consciousness within ten days of occurrence [13].

An additional safety measure set in place by the FAA is the UAS Remote Identification or Remote ID, also known as a digital license plate, which assists the FAA in providing additional information on the utilization of airspace to national security agencies law enforcement, and other government institutions. With Remote ID, UAS operators in flight can provide identification and location information to other parties such as towers, aircraft pilots, and law enforcement. When a UAS looks to be flying in a dangerous manner or in an area where it is not allowed to fly, Remote ID assists the FAA, law enforcement, and other federal authorities locating the control station [14].

There are three methods for UAS operators to meet the requirements of the Remote ID rule, outlines in Figure 1-6 below. One is to operate a standard Remote ID UAS, which has a built-in Remote ID capability to broadcast the requirements of the Remote ID rule. With this method, a UAS in flight provides UAS ID, location, altitude, velocity, control station location, elevation, time stamps, and emergency status from take-off to the end of UAS operation. Another method is to use a UAS with a Remote ID broadcast module, a device that can be added to UAS and broadcasts the required information, including take-off location. With this method, UAS operators must always be able to observe their UAS visually, and broadcasting gives all the information from the first method except emergency status during the entire operation of the UAS [14].

The last method is to fly UAS without the remote ID equipment at FAA-recognized identification areas, or FRIAs, the only locations UAS can operate without the required broadcasting remote ID equipment. Under this method, anyone can operate drones in the specified area, but the areas can be requested only by community-based organizations and educational establishments. The Remote ID rule requires all UAS manufacturers to comply with the requirements by September 16th of 2022 and requires all UAS operators to comply by September 16th, 2023 [14].

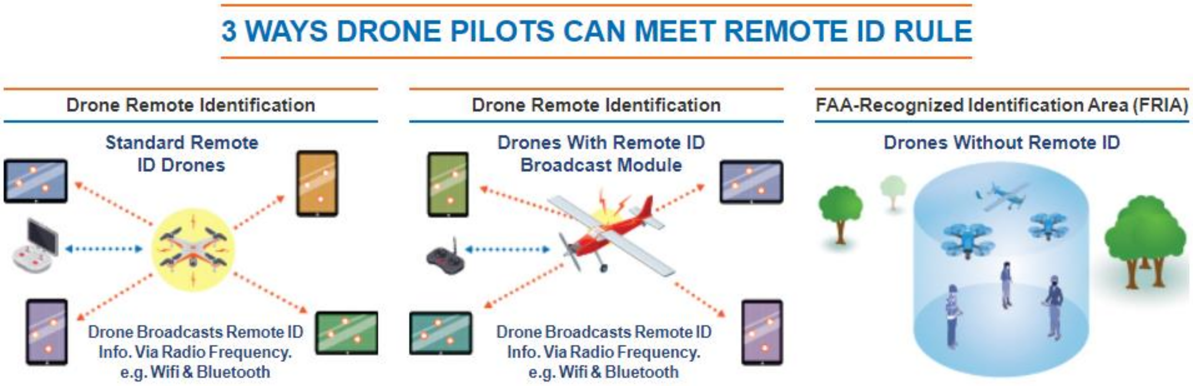

3 WAYS DRONE PILOTS CAN MEET REMOTE ID RULE
Drone Remote Identification
Standard Remote ID Drones
Drone Broadcasts Remote ID Info. Via Radio Frequency. e.g. Wifi & Bluetooth
Drone Remote Identification
Drones With Remote ID Broadcast Module
Drone Broadcasts Remote ID Info. Via Radio Frequency. e.g. Wifi & Bluetooth
FAA-Recognized Identification Area (FRIA)
Drones Without Remote ID

Figure 1-6. FAA's outlined three methods to satisfy Remote ID requirements [14]

The FAA has distributed such guidelines for UAS; however, operators do not always oblige with the rules while operating UAS. A concern with small UAS is that they are challenging to recognize entirely and are not effectively picked up by radars with current technology; the lack of automatically and accurately detecting UAS requires a manual step of detailing any unmanned aircraft infringing upon FAA guidelines. Currently, the FAA keeps a freely available report of Unmanned Aircraft sightings from pilots, residents, and law enforcement agencies to explore and instruct administrators operating UAS within unapproved limits [15]. The reports are available from sightings reported in November 2014 until June 2021 and include information on the date and location of the sighting and the textual message part from the reporting entity.

1.2 Problem Space

The increase in UAS usage for commercial and recreational purposes has led to an increase in safety issues in airspace and an increase in reports of such safety issues. The FAA has been receiving and recording UAS sighting reports since 2014. These reports can easily be found on the FAA website and are available to the public. The drone sightings reported to the FAA can come from different sources, as mentioned above, the most common source being pilots in flight. The report made by the pilot can offer a description of the UAS, including color, model, or size, and even the altitude in feet and direction of flight of the observed UAS. Small UAS being undetectable with the currently available radar technology leaves the UAS sighting reports received so far unverified, and the accuracy of the report dependent solely on the reporter.

Due to Congress and other parties expressing interest and raising questions about the extent of unsafe use of small Unmanned Aircraft Systems, the Government Accountability Office (GAO) in 2018 conducted a study of the FAA-distributed data set concerning the issue. GAO found there are several limitations to the FAA's information on the extent of such unsafe UAS use as FAA officials expressed to GAO that although the FAA receives such reports, the presence of UAS in the sightings cannot be verified. The FAA has also reported the ongoing technology development to detect and remotely identify UAS to better understand future reports received [16].

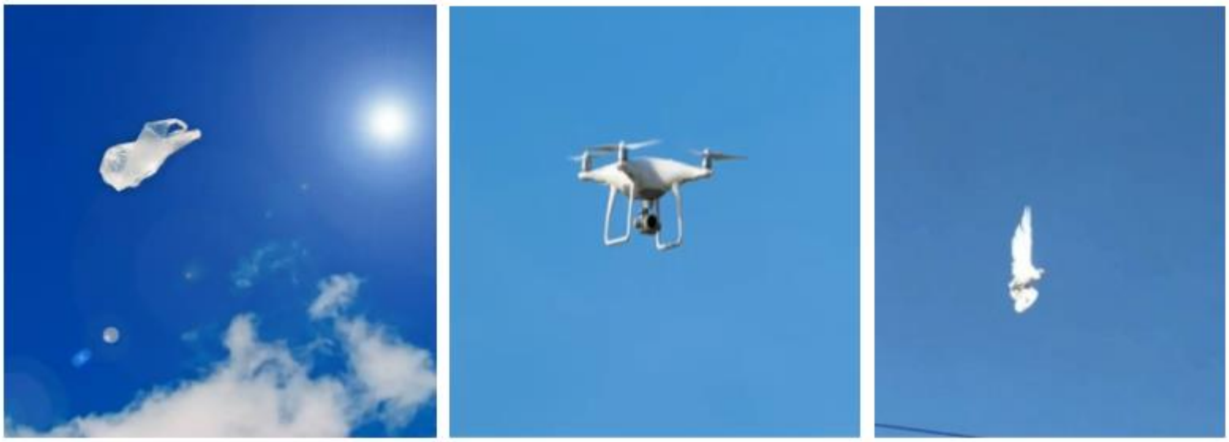


Figure 1-7. Comparison of similar looking objects possibly observed by pilots

The three images in Figure 1-7 above show what pilots could classify as a small white object observed during the flight, depending on the accuracy of visuals, closeness in the distance to objects, and ability to distinguish between objects. When considering these factors and the other demands of flying an aircraft, it can quickly be understood why a UAS incident report from pilots may not always be accurate. However, regardless of the report's accuracy, some data is better than no data, so learning how to utilize the available data is critical to the FAA's progress in UAS regulations. The available data will need to be utilized while the technology to automatically identify and verify UAS in flight is currently being developed and implemented. It is challenging to determine whether the reports are actual UAS sightings or 'noise' such as birds, plastic, balloons, or other objects in the airspace. It is also challenging to accurately understand the extent of noise in the reports sent to the FAA, making it difficult to gauge the degree of danger experienced over time. There are 13,000 records in the current FAA database for UAS sightings, but this does not correlate to 13,000 actual UAS incidents.

Another challenge with the reports submitted to the FAA is the inconsistency in recorded data. The records are in textual form, with most of the information chunked together in one text blob, making it challenging to analyze the raw data without heavy parsing and formatting. These restrictions on the report data can hinder the FAA's ongoing effort and progress in improving aviation safety and allowing a safe integration of manned and unmanned aircraft in the open sky.

Receiving an unformatted textual report is not ideal for processing large quantities of data for summarization or modeling. The FAA requires a data pipeline to extract essential variables into a formatted output for summary statistics and visualizations. These summary statistics and visualizations will allow the FAA to understand better the current UAS environment and how the UAS is being operated around manned aircraft. Our project aims to provide the FAA with historical insight into the contents of the reports submitted to them, which would allow the FAA to have a better understanding of the trends in the reports they receive. The team will utilize textual analysis and regular expression extracting methods to break down the reports and obtain critical information that would allow us to analyze the trends in the reports based on different factors. The team will consider factors such as location, time, and season of reporting, including relations to recent events such as the Covid-19 Pandemic, the reporting aircraft details, and the law enforcement agency involved when analyzing the trends in the reports received so far.

Since no current technology is being implemented to confirm UAS sightings, the FAA has years' worth of data that they cannot efficiently verify as confirmed sightings. The inability to label the data can limit their options for using it meaningfully. Our project will also be looking into validating the historical UAS incident reports from the FAA UAS Sightings dataset compared to other databases. A credible database that will be evaluated is the NASA Aviation Safety Reporting System, or ASRS, which records confidential reports of aviation traffic, including UAS sightings. Cross-validating the UAS sightings report with ASRS databases can give the FAA confidence in determining parts of the report that actual UAS incidents versus noise should be considered.

The team will work on the UAS Sightings Reports dataset, a paragraph-formatted textual report that includes various details such as location and time, a summary of the incident that comes directly from the reporting party, and any additional information detail the report receiver may have included such as aircraft type. We will work with textual analysis and mining methods to extract valuable details from the dataset to provide the FAA a constructed method of extracting data from their datasets, allowing them to continue to have further insight into other similar reports of future reports of the same kind.

1.3 Research

A big part of research and domain knowledge came from several meetings and discussions with our FAA partners that allowed us to understand the problem at hand better. With the continuing discussions, we could dig deeper into the UAS sighting dataset and the different values extracted from the reports. These discussions shaped an idea of what the final cleaned dataset would look like in preparing for further analysis.

The team was required to do much research to understand the problem's extent better. With extensive research, the team was able to understand the trends in drone usage, past and current events regarding drone sightings and incidents, and rules and regulations set in place by the FAA for operating drones. This allowed the team to gain better domain knowledge regarding aviation verbiage and a better understanding of terminology, descriptions, and abbreviations used in the reports from the datasets. One of the team's primary sources in better understanding the problem is the FAA site.

The FAA site provides the most information on current regulations and milestones the FAA has reached in regulating UAS operations and aviation safety. The UAS sightings reports are also available to the public on the site. The FAA site provides the following information on UAS:

- Current Regulations for UAS Operations
- Requirements for UAS operations
- UAS Registration information
- Aviation Terminology
- Remote Pilot and other certification information
- Research and Development
- Resources regarding safety and guidance
- Current trends and numbers of UAS usage

- Forecasting analysis and much more [2]

To better understand the extent of UAS sightings and incidents, the team researched recent news and studies on UAS sighting dangers. The team was able to identify trends of UAS usage utilizing market size predictions and evaluating reports dated back to 2015 to the present.

The next part of our research was textual analysis methods to extract details needed from the UAS sightings dataset. The team investigated several avenues for data extraction. The first one was AWS Glue Databrew, a user-friendly tool for data preparation that allows for easier data manipulation and cleaning, which is beneficial, especially when working with extensive textual data, such as in our case. AWS Databrew also provides column statistics and additional features that make the data cleaning and extracting process more manageable. AWS Databrew has a cost associated with running data manipulation tasks, making it impossible to use for this project without access from GMU.

After difficulties using AWS Databrew and further discussions with the FAA partner's preference in analytical tools, the team switched gears to using R and Python for data extraction using regular expressions.

R with Stringr: The Stringr package in R, a free software for statistical computing, allows for character manipulation and pattern searching functions with Regular Expressions [17].

Python with Pandas: Python is a high-level object-oriented programming language commonly used for data analysis. The Pandas package in Python allows for data manipulation and analysis utilizing data structure operations [18].

Another area of research was Natural Language Processing (NLP) techniques that can be used for cross-validation and modeling. The team looked at past capstone projects that worked with NLP as well as several online sources. For cross-validation and statistical modeling methods, the team looked into:

- Cosine similarity - cosine similarity is a method to measure the similarity in the text between two texts by measuring the cosine of the angle between the vector representation of the texts. The cosine similarity can have a value between 0 and 1. If the cosine similarity between the vectors is closer to 1, then the two vectors are considered similar. If the value is closer to 0, they are oriented differently, implying the compared texts are not similar [19].
- Word embedding – word embeddings are vector representations of words or text that represent semantically similar texts similarly. There are many pre-trained algorithms for word embeddings using Python and R. For this project, we looked more into the text2vec package in R that implements the GloVe algorithm for word embedding [20].
- Fuzzy matching algorithms - Fuzzy matching, also known as approximate string matching, is a type of search algorithm that will find matches between two texts considering misspells or partial entries for words, looking for strings that match approximately instead of an exact match [21].
- Topic modeling - A method used for classifying documents to find natural groupings and extract hidden topics from large documents or text [22].

The combinations in research of domain information, textual analysis tools, cross-validation techniques, predictive modeling methods, and visualization techniques allowed the team to explore the problem at hand and progress onto the solution space.

1.4 Solution Space

The approach to the solution consisted of textual analysis and NLP methods to work with the textual data. We used combinations of regular expressions to first extract details from the reports that would allow us to perform analysis based on several factors. The extraction process consisted of multiple cycles of trial and error to parse through the summary section of the FAA UAS Sightings dataset, which provided informative individual variables such as reporting aircraft type, which the incident was reported to, the time of day, whether the aircraft was commercial of general aviation and other generic details. This extracted data allows for summaries of the historical reports to better understand reported incidents. We also used the cleaned and extracted data for additional cross-validation against FAA internal dataset and for further analysis to provide insight.

Text similarity methods were used to cross-validate the FAA UAS sightings with the ASRS dataset to look for what would be considered actual sightings. Both data sets were grouped by the month, year, and state of the incident then cross-referenced against every report in the same group between the two datasets. Our similarity model scored the incident reports, and we chose the highest-scoring matches as confirmed sightings. Taking these newly confirmed reports, we labeled our dataset for use in predictive modeling to classify a report as an actual UAS sighting or noise.

An NLP workflow was utilized for the predictive modeling task. Every report was used to create a corpus of text specific to our project. The corpus was then used to run a Latent Dirichlet Allocation (LDA) model to uncover hidden topics across all of our reports. Once trained on our report text, the LDA model could ingest a report and assign the probability of the new report belonging to the topic. The topic probabilities for a report were then input to a logistic regression model to predict if the new report was an actual sighting or noise.

The data extraction, cross-validation process, predictive model workflow, and analysis of trends in reports across different factors will provide the FAA with the tools and confidence needed to continue data validation across several datasets and improve drone operation regulations promoting aviation safety.

1.5 Project Objectives

The main objective for this project is to understand the trends of UAS incident reports concerning various factors (location, aircraft, the agency notified, time of the day) and create a model that can accurately identify and predict 'noise' from the FAA UAS sightings dataset. The team plans to develop a data pipeline to process UAS incident reports into a format that can gather insights from analysis and modeling. The project's findings will help the FAA learn ways to identify and eliminate the noise from their datasets by using predictive modeling on new UAS incident reports as they are received to label a report as actual or noise.

Since the database is in a free-form text format, the data pipeline will require text mining and extraction work performed on the dataset. Once the data is adequately cleaned, parsed, and analyzed, the team hopes to provide the FAA with more insights about the historical records regarding multiple factors such as the source, the location, and the action reported in handling the sightings/incidents. The relevant information will be available in a format that the FAA can quickly summarize further. Our pipeline will also allow the FAA to process the relevant information on new reports as they become available.

The second part of our pipeline will utilize NLP predictive modeling to ingest a new UAS incident report and classify it as actual or noise. It will ingest data outputted from our processing pipeline used to extract summary data. This portion of the project will allow the FAA to receive a timely analysis of reports as they are made available.

The team hopes that providing deeper insight into the UAS sightings reports will determine trends to identify the difference in 'noise' versus 'actuals' within the reports. The FAA will be one step closer to having a clear and accurate vision of the extent of the ongoing issue with UAS sightings as reported by various sources, which will allow the FAA to assess current regulations and place appropriate safety and risk measures with drone operations as well as unmanned aircraft integration into open skies.

1.6 Primary User Stories:

As a user, I want to be able to identify false positives and understand the actuals of the database.

As an FAA forecaster, I want to be able to determine the quality of noise versus actual reports from the FAA UAS sightings data.

As an analyst, I want to find the best way to organize, clean and extract the UAS sightings report data.

As an FAA forecaster, I want to know what methods of text mining and analysis are best to understand the textual narratives in the UAS sightings report data.

As an analyst, I want to find cross validation points between the ASRS and UAS sighting datasets.

1.7 Product Vision - Sample scenarios (why would someone want to use this)

The recent increase in UAS usage and issues with safety makes the findings of this project valuable to the FAA and other involved parties looking to improve safety, reduce incidents relating to drones by putting stricter regulations in place, and allowing safer methods for manned and unmanned aircraft to exist in the open sky. This project is precious for:

- For: The FAA as well as other related agencies
- Who: work towards improving aviation safety and drone operation regulations
- The: analysis and cross-validation of the UAS dataset provide
- Is a: deeper insight of the UAS sightings dataset from the FAA
- That: that provides an understanding of trends in drone sighting reports and knowledge of the extent of 'noise' vs. 'actuals' for received reports
- Unlike: the current report format that provides no context or validity to the individual reports
- Our product: can take raw data, cross-validate, train predictive models for new reports, and process, clean, and format raw text into an analyze-friendly format.
- Caveats: the lack of any verified reports has limited the performance of the predictive modeling

Scenario #1

A pilot reports drone sighting during flight with specific details. Based on the reporting aircraft and reported details, identify whether the report can be considered viable.

Scenario #2

A team at FAA wants to look at historical UAS incidents to try and find trends. The team will need to be able to extract relevant data from the UAS incidents reports to develop a picture.

1.8 Definition of Terms:

Word	Abbreviation	Definition
Air Route Traffic Control Center	ARTCC	A facility responsible for controlling aircraft flying in the airspace of a given flight information region at high altitudes between airport approaches and departures. Also referred to as ’centers’.
Air Traffic Control Command Center	ATCCC	The ATCCC Team uses traffic management initiatives (TMIs) to manage the flow of air traffic and minimize delays. They are also referred to as 'Air Traffic Control System Command Center' (ATCSCC).
Aircraft	A/C	An aircraft is a vehicle or machine that can fly by gaining support from the air— also abbreviated as 'ACFT.'
Aircraft Type Designator		An alphanumeric identifier of no more than four characters used to identify aircraft uniquely.
Airport	ARPT	A place from which aircraft operate aerodrome with extended facilities, primarily for commercial air transport.
Airport Traffic Control Towers	ATCT	A tall building structure located strategically gives the best view of the entire airport area and its surroundings.
Aviation Safety Reporting System	ASRS	The National Aeronautics and Space Administration (NASA) designed a database system to capture confidential reports, analyze the resulting aviation safety data, and disseminate vital information to the aviation community.
Comma Seperated Values	CSV	Comma Separated Values
Cosine Similarity Modeling		NLP technique that allows for 2 text documents to be compared for similarity. It does so by converting the text into separate numerical representations which can be compared by their distances(differences) from each other. Smaller distances result in higher similarity scores.
CSV files	CSV	A file storage method that allows for quick access and sharing. The files contain all of the data separated by a comma.

Drone		aircraft that is not operated with direct human contact but instead with remote control. Also referred to as Unmanned Aircraft Systems (UAS).
Federal Aviation Administration	FAA	The largest transportation agency of the U.S. Department of Transportation is to provide a safe and efficient aerospace system.
Government Accountability Office	GAO	A branch government agency that provides auditing, evaluation, and investigative services for the United States Congress.
Inverse Document Frequency	IDF	A measure of how rare a term is, where the greater the score the rarer the term.
Latent Dirichlet Allocation Topic Modeling	LDA	An algorithm that uncovers a specified number of clusters in data. The number of desired clusters is determined by the user which leaves the final number subjective.
Lemmatization		Similar to stemming, process of grouping together the inflected forms of a word so they can be analysed as the actual root.
Logistic Regression		An algorithmic technique used for predictive modeling. Weights are assigned to input variables that can then be applied to make predictions on a binary outcome variable. These models can be tuned multiple ways, our project aimed to minimize the false positive classifications while maximizing the true positive classifications.
Manned Aircraft		An aircraft that is being operated by direct physical contact of a pilot or a human operator.
National Airspace System	NAS	A network of U.S airspace with controlled and uncontrolled airspace includes air navigation, equipment services, airport, landing areas, rules and regulations, workforce, material, and procedures.
Natural Language Processing		The branch of artificial intelligence concerned with giving computers the ability to understand text and spoken words in much the same way human beings can. Allows to analyze natural languages.
Near Mid Air Collision	NMAC	Occurs when two aircraft come within 100 feet vertically and 500 feet horizontally.
Regular Expression	Regex	A sequence or pattern of characters used to match/search for a string
Safety Management System	SMS	An organization within the FAA to managing safety risk and assuring the effectiveness of safety risk controls.
Stemming		Removing the suffix from a word and reduce it to its root word.

Stop word removal		Stop words are a set of commonly used words in a language or topic.
Term Frequency	TF	is a scoring of the frequency of the word in the current document. Since every document is different in length, it is possible that a term would appear much more often in long documents than shorter ones. The term frequency is often divided by the document length to normalize (D'Souza 2018).
Term Frequency - Inverse Document Frequency	TF-IDF	In TF-IDF, the weight assigned to each token not only depends on its frequency in a document but also how recurrent that term is in the entire corpora. TF-IDF weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus (D'Souza 2018).
Terminal Radar Approach Control	TRACON	FAA facility that houses aircraft traffic controllers and guides aircraft departing and arriving at airports.
Text Normalization		Text normalization is the process of transforming text into a single canonical form that it might not have had before.
Tokenization		Tokenization is a very common task in NLP. Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens (Christopher et al. 2008). Here is an example of tokenization
Unmanned Aircraft Systems	UAS	Aircraft that is not operated with direct human contact but instead with remote control. Also referred to as drones.

2 Data Acquisition

2.1 Overview:

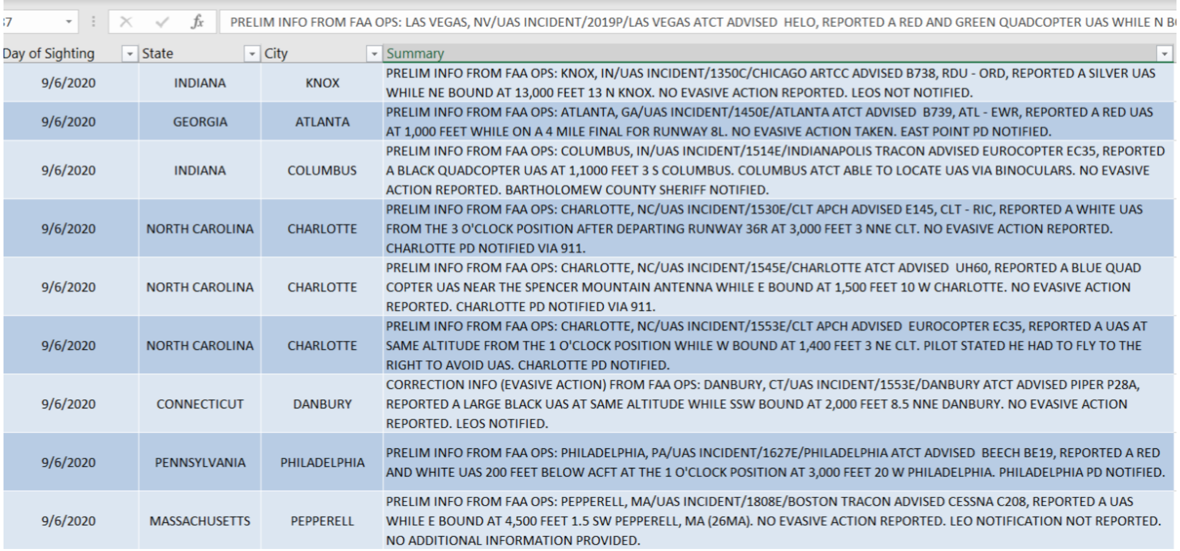
The primary datasets for the project, FAA's UAS sighting dataset, and NASA's ASRS dataset, are available to the public. The FAA provided an additional dataset for aircraft information for supplementary information on aircraft types.

2.1.1 UAS Sightings Report Dataset

The FAA UAS sightings report is the primary dataset used for this project. The dataset for UAS drone sightings is available for the public on the FAA site, where the sightings are posted in quarterly chunks, and there are reports of UAS sightings recorded from November 2014 until June 2021. The dataset has four columns and about 13000 records. The four columns are:

- Day of Sighting - date and time of reporting
- State - state where sighting was reported
- City – city where sighting was reported
- Summary – text blob which includes information from the actual report, as well as additional information regarding the report.

The summary of the UAS sightings report can include descriptions of reporting aircraft type, the direction of flight, altitude of flight of aircraft, and the distance of UAS in relation to aircraft, a color description of the UAS, evasive action taken, and law enforcement agency that was notified.



PRELIM INFO FROM FAA OPS: LAS VEGAS, NV/UAS INCIDENT/2019P/LAS VEGAS ATCT ADVISED HELO, REPORTED A RED AND GREEN QUADCOPTER UAS WHILE N B

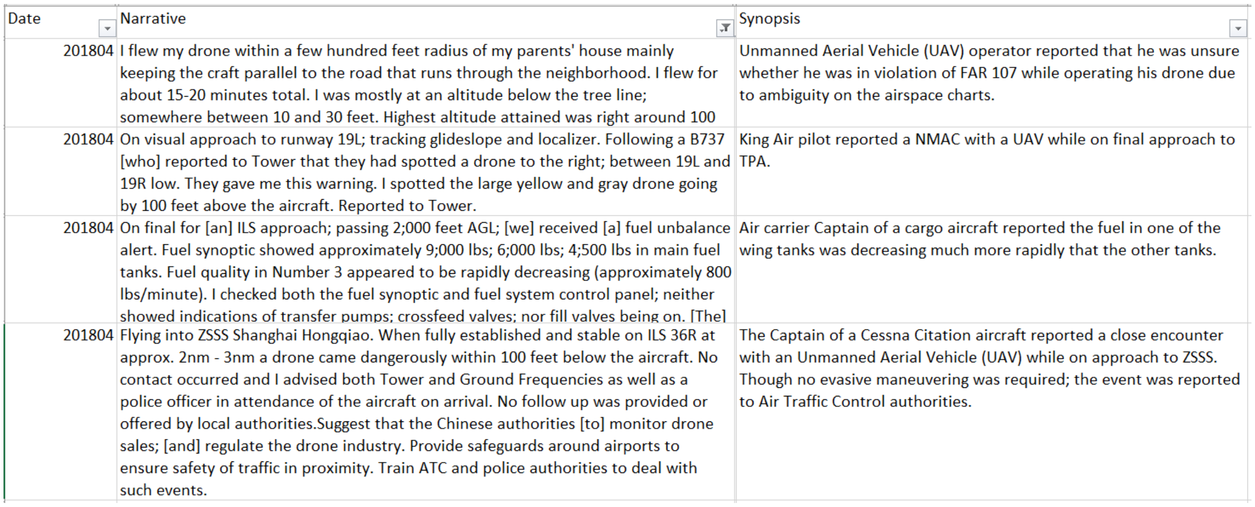
Day of Sighting	State	City	Summary
9/6/2020	INDIANA	KNOX	PRELIM INFO FROM FAA OPS: KNOX, IN/UAS INCIDENT/1350C/CHICAGO ARTCC ADVISED B738, RDU - ORD, REPORTED A SILVER UAS WHILE NE BOUND AT 13,000 FEET 13 N KNOX. NO EVASIVE ACTION REPORTED. LEOS NOT NOTIFIED.
9/6/2020	GEORGIA	ATLANTA	PRELIM INFO FROM FAA OPS: ATLANTA, GA/UAS INCIDENT/1450E/ATLANTA ATCT ADVISED B739, ATL - EWR, REPORTED A RED UAS AT 1,000 FEET WHILE ON A 4 MILE FINAL FOR RUNWAY 8L. NO EVASIVE ACTION TAKEN. EAST POINT PD NOTIFIED.
9/6/2020	INDIANA	COLUMBUS	PRELIM INFO FROM FAA OPS: COLUMBUS, IN/UAS INCIDENT/1514E/INDIANAPOLIS TRACON ADVISED EUROCOPTER EC35, REPORTED A BLACK QUADCOPTER UAS AT 1,1000 FEET 3 S COLUMBUS. COLUMBUS ATCT ABLE TO LOCATE UAS VIA BINOCULARS. NO EVASIVE ACTION REPORTED. BARTHOLOMEW COUNTY SHERIFF NOTIFIED.
9/6/2020	NORTH CAROLINA	CHARLOTTE	PRELIM INFO FROM FAA OPS: CHARLOTTE, NC/UAS INCIDENT/1530E/CLT APCH ADVISED E145, CLT - RIC, REPORTED A WHITE UAS FROM THE 3 O'CLOCK POSITION AFTER DEPARTING RUNWAY 36R AT 3,000 FEET 3 NNE CLT. NO EVASIVE ACTION REPORTED. CHARLOTTE PD NOTIFIED VIA 911.
9/6/2020	NORTH CAROLINA	CHARLOTTE	PRELIM INFO FROM FAA OPS: CHARLOTTE, NC/UAS INCIDENT/1545E/CHARLOTTE ATCT ADVISED UH60, REPORTED A BLUE QUAD COPTER UAS NEAR THE SPENCER MOUNTAIN ANTENNA WHILE E BOUND AT 1,500 FEET 10 W CHARLOTTE. NO EVASIVE ACTION REPORTED. CHARLOTTE PD NOTIFIED VIA 911.
9/6/2020	NORTH CAROLINA	CHARLOTTE	PRELIM INFO FROM FAA OPS: CHARLOTTE, NC/UAS INCIDENT/1553E/CLT APCH ADVISED EUROCOPTER EC35, REPORTED A UAS AT SAME ALTITUDE FROM THE 1 O'CLOCK POSITION WHILE W BOUND AT 1,400 FEET 3 NE CLT. PILOT STATED HE HAD TO FLY TO THE RIGHT TO AVOID UAS. CHARLOTTE PD NOTIFIED.
9/6/2020	CONNECTICUT	DANBURY	CORRECTION INFO (EVASIVE ACTION) FROM FAA OPS: DANBURY, CT/UAS INCIDENT/1553E/DANBURY ATCT ADVISED PIPER P28A, REPORTED A LARGE BLACK UAS AT SAME ALTITUDE WHILE SSW BOUND AT 2,000 FEET 8.5 NNE DANBURY. NO EVASIVE ACTION REPORTED. LEOS NOTIFIED.
9/6/2020	PENNSYLVANIA	PHILADELPHIA	PRELIM INFO FROM FAA OPS: PHILADELPHIA, PA/UAS INCIDENT/1627E/PHILADELPHIA ATCT ADVISED BEECH BE19, REPORTED A RED AND WHITE UAS 200 FEET BELOW ACFT AT THE 1 O'CLOCK POSITION AT 3,000 FEET 20 W PHILADELPHIA. PHILADELPHIA PD NOTIFIED.
9/6/2020	MASSACHUSETTS	PEPPERELL	PRELIM INFO FROM FAA OPS: PEPPERELL, MA/UAS INCIDENT/1808E/BOSTON TRACON ADVISED CESSNA C208, REPORTED A UAS WHILE E BOUND AT 4,500 FEET 1.5 SW PEPPERELL, MA (26MA). NO EVASIVE ACTION REPORTED. LEO NOTIFICATION NOT REPORTED. NO ADDITIONAL INFORMATION PROVIDED.

Figure 2-1. A sample of the UAS Sightings Dataset from the FAA

2.1.2 ASRS Dataset

ASRS captures confidential reports, analyzes the resulting aviation safety data, and disseminates vital information to the aviation community. Data from the ASRS database can be acquired by the public as well. For specific results on drone sightings, filtering the synopsis and narrative fields to include different variations and spellings of keywords such as 'drone', 'uas', 'unmanned' and 'quadcopter' can be beneficial. We initially used a complete data download for this project and the uncertainty that comes with using the specific keywords, primarily when the reports and summaries depend on the correct inclusion of appropriate words by the reporter or the report receiving agency. This resulted in a dataset with approximately 35,600 records and 125 columns, with a total size of about 9.5 GB. The dataset also had many null values and variables irrelevant to our project. For this project and cross-validating, we are focusing on the columns date, state, narrative, and synopsis:

- Date – the month and year of incident
- Narrative – Pilot's summary of the event
- Synopsis – NASA's summary of the Pilot's account
- State – state the incident happened in

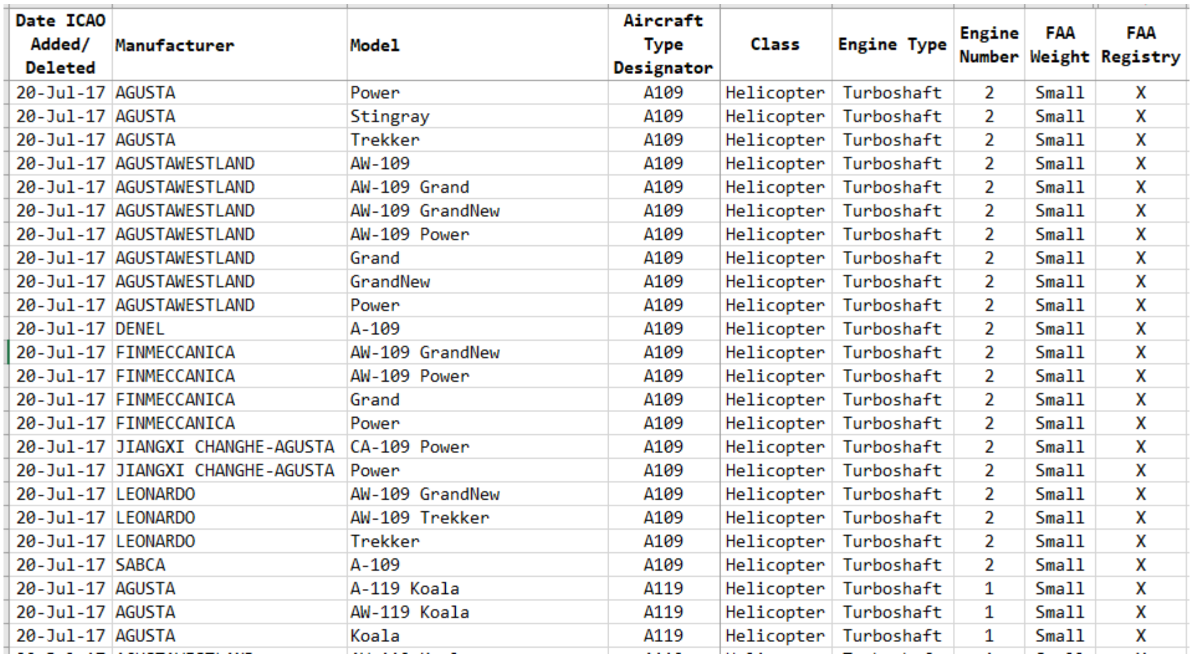


Date	Narrative	Synopsis
201804	I flew my drone within a few hundred feet radius of my parents' house mainly keeping the craft parallel to the road that runs through the neighborhood. I flew for about 15-20 minutes total. I was mostly at an altitude below the tree line; somewhere between 10 and 30 feet. Highest altitude attained was right around 100	Unmanned Aerial Vehicle (UAV) operator reported that he was unsure whether he was in violation of FAR 107 while operating his drone due to ambiguity on the airspace charts.
201804	On visual approach to runway 19L; tracking glideslope and localizer. Following a B737 [who] reported to Tower that they had spotted a drone to the right; between 19L and 19R low. They gave me this warning. I spotted the large yellow and gray drone going by 100 feet above the aircraft. Reported to Tower.	King Air pilot reported a NMAC with a UAV while on final approach to TPA.
201804	On final for [an] ILS approach; passing 2;000 feet AGL; [we] received [a] fuel unbalance alert. Fuel synoptic showed approximately 9;000 lbs; 6;000 lbs; 4;500 lbs in main fuel tanks. Fuel quality in Number 3 appeared to be rapidly decreasing (approximately 800 lbs/minute). I checked both the fuel synoptic and fuel system control panel; neither showed indications of transfer pumps; crossfeed valves; nor fill valves being on. [The]	Air carrier Captain of a cargo aircraft reported the fuel in one of the wing tanks was decreasing much more rapidly that the other tanks.
201804	Flying into ZSSS Shanghai Hongqiao. When fully established and stable on ILS 36R at approx. 2nm - 3nm a drone came dangerously within 100 feet below the aircraft. No contact occurred and I advised both Tower and Ground Frequencies as well as a police officer in attendance of the aircraft on arrival. No follow up was provided or offered by local authorities.Suggest that the Chinese authorities [to] monitor drone sales; [and] regulate the drone industry. Provide safeguards around airports to ensure safety of traffic in proximity. Train ATC and police authorities to deal with such events.	The Captain of a Cessna Citation aircraft reported a close encounter with an Unmanned Aerial Vehicle (UAV) while on approach to ZSSS. Though no evasive maneuvering was required; the event was reported to Air Traffic Control authorities.

Figure 2-2. A sample of columns in ASRS dataset used for project

2.1.3 FAA Aircraft Information Dataset

The FAA provided the aircraft information and included various details about different aircraft types. The dataset includes information on the aircraft manufacturer, the model, the aircraft designator, details about aircraft and aircraft parts, among many others. The detail in this dataset allows the team to have better domain knowledge of the nature of the reporting aircraft and allows for a deeper analysis of reporting aircraft as a factor. For this project, the FAA partners requested a focus on the specific columns Class, Engine Type, Engine Number, FAA Weight, and Air Carrier.



Date ICAO Added/ Deleted	Manufacturer	Model	Aircraft Type Designator	Class	Engine Type	Engine Number	FAA Weight	FAA Registry
20-Jul-17	AGUSTA	Power	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	AGUSTA	Stingray	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	AGUSTA	Trekker	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	AGUSTAWESTLAND	AW-109	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	AGUSTAWESTLAND	AW-109 Grand	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	AGUSTAWESTLAND	AW-109 GrandNew	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	AGUSTAWESTLAND	AW-109 Power	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	AGUSTAWESTLAND	Grand	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	AGUSTAWESTLAND	GrandNew	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	AGUSTAWESTLAND	Power	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	DENEL	A-109	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	FINMECCANICA	AW-109 GrandNew	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	FINMECCANICA	AW-109 Power	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	FINMECCANICA	Grand	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	FINMECCANICA	Power	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	JIANGXI CHANGHE-AGUSTA	CA-109 Power	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	JIANGXI CHANGHE-AGUSTA	Power	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	LEONARDO	AW-109 GrandNew	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	LEONARDO	AW-109 Trekker	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	LEONARDO	Trekker	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	SABCA	A-109	A109	Helicopter	Turboshaft	2	Small	X
20-Jul-17	AGUSTA	A-119 Koala	A119	Helicopter	Turboshaft	1	Small	X
20-Jul-17	AGUSTA	AW-119 Koala	A119	Helicopter	Turboshaft	1	Small	X
20-Jul-17	AGUSTA	Koala	A119	Helicopter	Turboshaft	1	Small	X

Figure 2-3. Sample of FAA Aircraft Information Dataset

2.2 Field Descriptions:

After parsing through the FAA dataset, extracting multiple columns from the summary section of the FAA UAS Sightings dataset, and combining information of the reporting aircraft with the aircraft information dataset provided from the FAA, we have the following as our current data fields and their description:

Field Name	Type	Description
Date	Datetime	A column with the date and time information of when the report was made. The date has 'MM/DD/YYYY' format followed by a military time version of the time of report. A sample record is '11/13/2014 3:30:00 PM' indicating the month, day, year as well as hour and minute of reporting time.
Event	String	This field is the original raw text blob found in the FAA UAS sightings data reports. It has several pieces of information filled in by the report receiving entity such as incident type, location, reporting time, the report details, law enforcement agency notified, and other details separated with punctuations in some cases or as a lump sentence in others. This field is further extracted in other columns.

Summary	String	The column in the FAA UAS sightings dataset that includes the report part of the summary that has the main details on the sighting. This was separated out from the raw ‘Event’ field in the UAS sightings data from all other information the original event field included such as preliminary details. It is the cleaned data field that was used in the NLP models.
City	String	This field is from the FAA UAS sightings data and has information on the city where the report was received.
State	String	This field is from the FAA UAS sightings data and has information on the State where the report was received.
Alert_For	String	This extracted field includes the additional information filled out by the FAA operator on some of the records where details such as FAA operator summarization of the report, possible follow-up information or details on aircraft information can be found. Due to inconsistencies in recording of reports, this can only be found on some records and results in null values for the rest.
Notified	String	This field has information extracted from the summary of the FAA UAS sightings data on who was notified of the sightings or incident if possible. The notified category in this field was further standardized in data cleaning to group similar descriptions of entities together. This field has null values depending on if the information is not available or due to not being picked up by the regular expression used for data extraction and cleaning.
Advise	String	This field has an extracted value from the main report section that has the details on the tower or control that is receiving the report from pilots. This field has null values either due to lack of details on the report receiving entity or the appropriate information not being picked up the by the regular expression used for data extraction and cleaning.
Time	String	This field has the time value extracted from the preliminary information of the event category of the FAA UAS sightings data. The time stamp is in military time, with just four digits (‘1617’ for 4:17 PM).
Timezone	String	The field has a one letter abbreviation/code of the military time zone attached to the four-digit time stamp in the preliminary information section of the FAA UAS sighting.
Aircraft	String	This field has an extracted value of the reporting aircraft type or designator that was mentioned in the FAA UAS sightings report data.

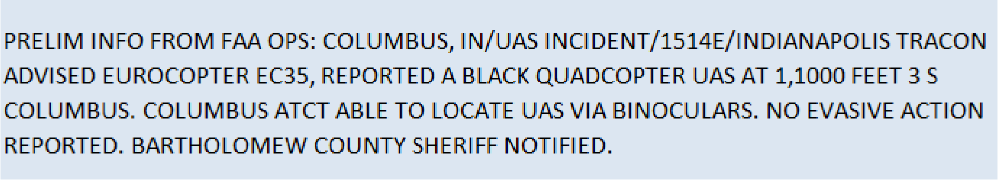
UAScolor	String	This field is an extracted value of a UAS color described in the report summary of the FAA UAS reports data. The field can be null due to lack of color description in the reports or inability to be picked up with the regular expression used to extract and clean data.
Altitude	String	This field has the altitude in feet mentioned in the FAA UAS Sightings data and was extracted from the report summary section.
Class	String	This field is merged from an FAA provided aircraft information dataset based on a lookup on the aircraft type that was extracted and mentioned earlier on this list. The 'Class' field gives additional information on the class of the reporting aircraft class; Landplane, Amphibian, Helicopter, Gyrocopter, Seaplane and Tiltrotor are among aircraft class found in the dataset.
Engine Type	String	This field is another merged from the FAA provided aircraft information dataset based on a lookup on the aircraft type that was extracted and mentioned earlier on this list. The 'Engine Type' field gives additional information on the engine type of the reporting aircraft type and the different classifications include Jet, Piston, Turboprop and Turboshaft.
Engine Number	String	This is another field merged from the FAA provided aircraft information dataset to provide additional information on the reporting aircraft type that was extracted earlier. The 'Engine Number' field gives additional information
FAA Weight	String	This field includes additional information on the FAA classified weight category of aircraft and this was merged with an FAA provided aircraft information dataset. The FAA weight categories different classifications associated with each aircraft type such as small, large, heavy, super and more.
Air Carrier	String	Designator for commercial or recreational flying.

2.3 Data Context:

Our dataset represents UAS sightings submitted by pilots to the FAA from November 2014 to June 2021. The final dataset is a combination of UAS incident and sightings reports collected by the FAA combined with the FAA Aircraft Information for the reporting aircraft, if available in the reports. The UAS sightings data is released quarterly; therefore, several reporting datasets from the FAA were merged to compose the final reporting data. The summary section of the reports was further broken down into multiple columns to compose several extracted characteristics such as altitude in feet of the observed UAS, the reporting aircraft manufacturer and craft designator, the notified law enforcement agency, and more.

2.4 Data Conditioning

One of the main challenges of this process was working with the summary section of the UAS sightings data. The summary column includes most of the essential information in a single text blob with inconsistent formatting. The raw text contains information such as the altitude of the drone or plane, the reporting aircraft or the tower that received the information if reported by a pilot, the police department or law enforcement agency notified, and so on. The majority of the summary column records in the dataset had the format below in Figure 2-4 with different variations of the details that were included.



PRELIM INFO FROM FAA OPS: COLUMBUS, IN/UAS INCIDENT/1514E/INDIANAPOLIS TRACON ADVISED EUROCOPTER EC35, REPORTED A BLACK QUADCOPTER UAS AT 1,1000 FEET 3 S COLUMBUS. COLUMBUS ATCT ABLE TO LOCATE UAS VIA BINOCULARS. NO EVASIVE ACTION REPORTED. BARTHOLOMEW COUNTY SHERIFF NOTIFIED.

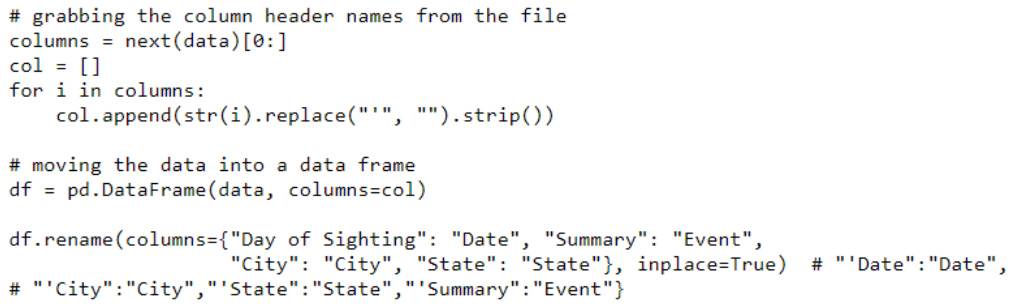
Figure 2-4. A sample record of a report found in the 'Summary' section of UAS sightings dataset

Another challenge for this project was to extract all such information from the summary section consistently. Since the nature of the data we are working on are reports created from verbal submissions, there were many inconsistencies in reporting where some of the reports were missing details, others included. These inconsistencies created a scenario where we had to utilize many regular expression(regex) patterns with error catching to process most of the reports accurately. The reports also span over seven years, creating formatting and encoding issues from one quarterly file to the next. We used regex to work through these issues, which allowed us not to drop any of the quarterly reports due to encoding differences. One of the methods the team utilized was extracting the specific details mentioned above using regular expressions in Python and R. Due to the inconsistency of the reporting method, the extracting and parsing process was a cycle of trying multiple regex patterns until the extraction method handled the majority of the formatting scenarios. The process also included manually fixing some issues that were not appropriately extracted or formed mistakes due to the regex pattern used.

Our first task was to merge all the quarterly FAA UAS incident reports and the ASRS reports for the same time-period.

2.4.1 Merging UAS Sightings Reports

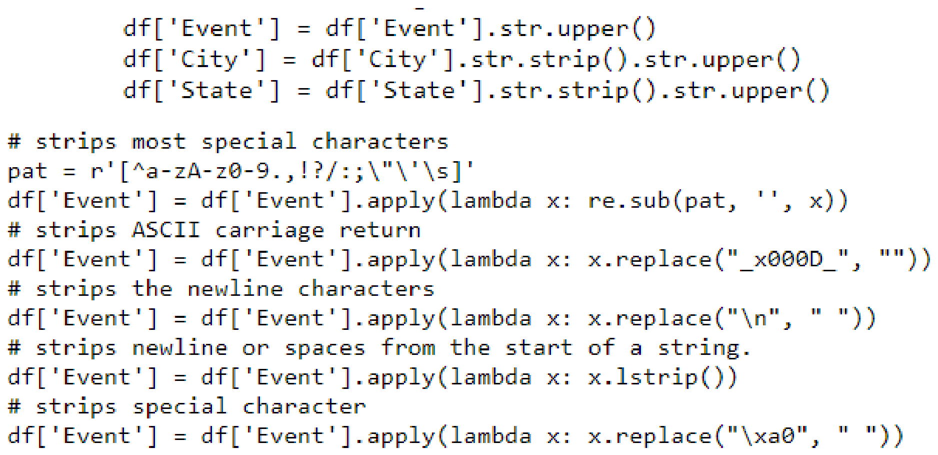
The multiple files of UAS sightings reports had to be merged into one dataset to prepare the FAA data for analysis and cross-validation. As previously mentioned, the FAA releases the reports every quarter. Starting from 2014 through June 2021, they were that 25 datasets needed to be merged. Python was used to merge all the datasets into one dataset used for the project. The merged dataset has columns Date, City, State, and Event, originally 'Summary' in the FAA UAS Sighting datasets. This was done by importing the files into a data frame in Python then appending each data frame to a single data frame containing all the reports. The master data frame was then exported to a CSV format.



```
# grabbing the column header names from the file
columns = next(data)[0:]
col = []
for i in columns:
    col.append(str(i).replace("'", "").strip())

# moving the data into a data frame
df = pd.DataFrame(data, columns=col)

df.rename(columns={"Day of Sighting": "Date", "Summary": "Event",
                   "City": "City", "State": "State"}, inplace=True)  # "'Date":"Date",
# "'City":"City","'State":"State","'Summary":"Event"}
```

Figure 2-5. Python Code – Importing and appending UAS sightings data

All string-type variables of the merged dataset were also standardized to uppercase letters in this process. The 'Event' field was further cleaned to remove all encoding irregularities.



```
    df['Event'] = df['Event'].str.upper()
    df['City'] = df['City'].str.strip().str.upper()
    df['State'] = df['State'].str.strip().str.upper()

# strips most special characters
pat = r'[^a-zA-z0-9.,!?/:;\"\'\s]'
df['Event'] = df['Event'].apply(lambda x: re.sub(pat, '', x))
# strips ASCII carriage return
df['Event'] = df['Event'].apply(lambda x: x.replace("_x000D_", ""))
# strips the newline characters
df['Event'] = df['Event'].apply(lambda x: x.replace("\n", " "))
# strips newline or spaces from the start of a string.
df['Event'] = df['Event'].apply(lambda x: x.lstrip())
# strips special character
df['Event'] = df['Event'].apply(lambda x: x.replace("\xa0", " "))
```

Figure 2-6. Python Code – Cleaning of 'Event' field

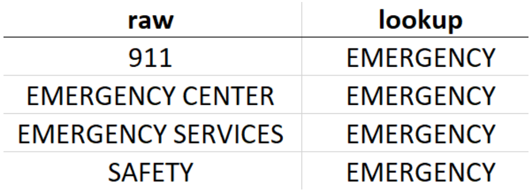
2.4.2 Merging ASRS Dataset

The ASRS data set was limited to 5000 records per export. The team had to download nine separate report queries to be merged into a single file. The merge encountered no encoding issues. The same technique for merging the UAS dataset was incorporated for this data set.

With our data set consolidated into two files, we looked to clean the data following to extract information that the FAA was interested in using.

2.4.3 Standardization of Text in UAS Dataset

The UAS incident reports included free form inputs that created inconsistencies in our data, e.g., a report location might be input as “California,” and then the following report location could be “Cali.” For analysis, we would consider those both California and had to process the UAS incident reports to extract data consistently. Standardization was required for the city, state, and law enforcement agency that was notified. We manually created lookup dictionaries that captured every unique use of a word assigned as the key and a standardized naming convention as the value. Below is an example of how we turned four different values into a single group. The team was able to standardize the final dataset using the lookup dictionaries.

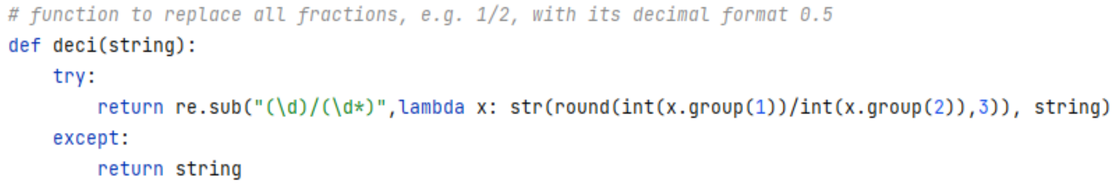


raw	lookup
911	EMERGENCY
EMERGENCY CENTER	EMERGENCY
EMERGENCY SERVICES	EMERGENCY
SAFETY	EMERGENCY

Figure 2-7. Example of lookup dictionary created

2.4.4 Removal of Fractions in UAS Dataset

Several of the reports included numbers written in fractions, e.g., ½. The team used regex to locate the fractions and convert them to decimal values.



```
# function to replace all fractions, e.g. 1/2, with its decimal format 0.5
def deci(string):
    try:
        return re.sub("(\d)/(\d*)",lambda x: str(round(int(x.group(1))/int(x.group(2)),3)), string)
    except:
        return string
```

Figure 2-8. Python Code – Replacing fraction values into decimal formats

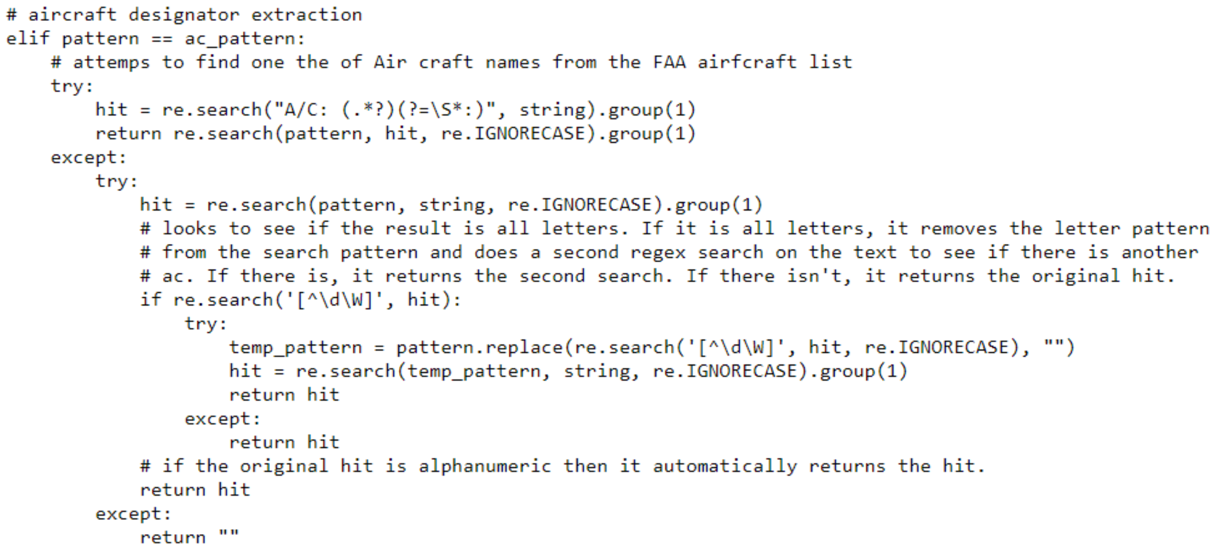
2.4.5 Extracting Details from UAS Dataset

The merged FAA UAS sightings data was first imported to python and standardized to all uppercases again to avoid issues when using regular expressions. We used the python package, Pandas, an open-source package in python used for data analysis, data science, and machine learning. Pandas can be used for data cleaning, manipulation, merging, wrangling, visualizations, statistical analysis, and more [18].

The FAA team provided us with a list of data categories they would be interested in seeing extracted for comment. These categories include the law enforcement agency that was notified, the initial location of the incident report, the time of the incident, the aircraft type reporting the incident, the altitude of the UAS, the location of the incident, and the color of the UAS. In addition to these data points extracted from the incident report, the FAA also wanted us to use the internal aircraft information dataset as a reference to look up details on the reporting aircraft and append it to the dataset for further analysis. The data that was added to the UAS incident reports includes the FAA class designation for the aircraft (Landplane, Amphibian, Gyrocopter, Helicopter, Seaplane, or Tiltrotor), the engine type of the aircraft (Jet, Piston, Turboprop, or Turboshaft), the number of engines, the FAA weight classification (Small, Small+, Large, Heavy, or Super), and if the aircraft is commercial or general aviation. These additions were made through a simple dictionary lookup on the aircraft name provided in the report. Extracting the information from the UAS incident reports was much more involved and complicated.

From the FAA-provided Aircraft Information dataset, we used the Aircraft Type Designator column as a lookup value for the reporting aircraft mentioned in the UAS incident report. After gathering the impressive list of aircraft designators in the aircraft information dataset, we created a regular expression string from the impressive list to extract any designator mentioned in the UAS incident reports. Aircraft or reporting entity not included in the aircraft type designator list from the

FAA-provided Aircraft Information Dataset, such as B757, was manually added to the regex string. Other reporting aircraft types that do not have aircraft designators are mentioned in the reports, which were described in the summary of the UAS sightings data as 'HELO,' 'HELI,' 'HELICOPTER' and 'HEL HELI,' 'HELICOPTER' and HAWK,' were also added to the regex string manually. The values for the extracted reporting aircraft information were placed in a new column, 'Aircraft.' The extracted aircraft designator was then used to lookup the internal FAA information requested to be added to the reports.

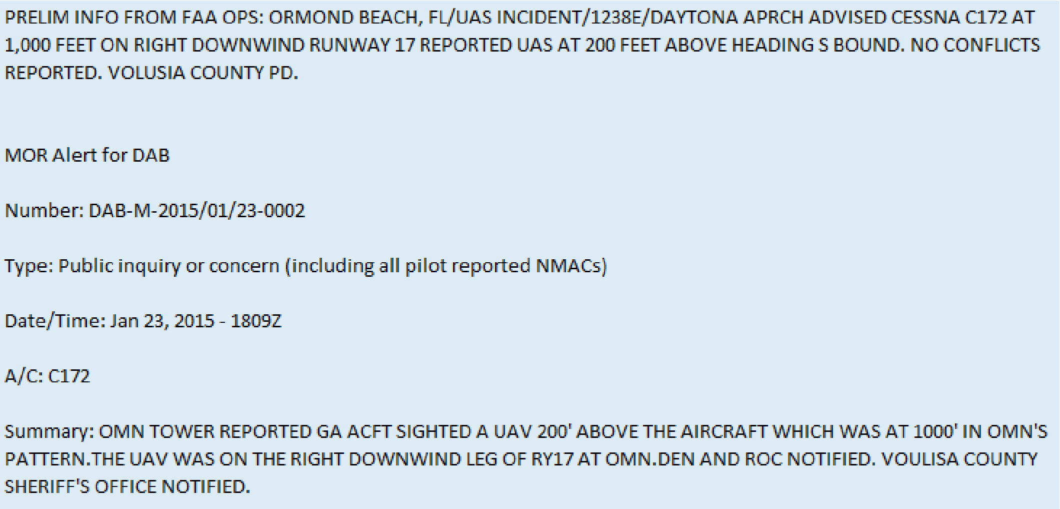


```
# aircraft designator extraction
elif pattern == ac_pattern:
    # attemps to find one the of Air craft names from the FAA airfcraft list
    try:
        hit = re.search("A/C: (.*?)(?=\S*:)", string).group(1)
        return re.search(pattern, hit, re.IGNORECASE).group(1)
    except:
        try:
            hit = re.search(pattern, string, re.IGNORECASE).group(1)
            # looks to see if the result is all letters. If it is all letters, it removes the letter pattern
            # from the search pattern and does a second regex search on the text to see if there is another
            # ac. If there is, it returns the second search. If there isn't, it returns the original hit.
            if re.search('[^\d\W]', hit):
                try:
                    temp_pattern = pattern.replace(re.search('[^\d\W]', hit, re.IGNORECASE), "")
                    hit = re.search(temp_pattern, string, re.IGNORECASE).group(1)
                    return hit
                except:
                    return hit
            # if the original hit is alphanumeric then it automatically returns the hit.
            return hit
        except:
            return ""
```

Figure 2-9. Python Code – Extracting Aircraft Designator

A list of colors of interest was constructed and used to search each record to extract the UAS color reported in the sighting. We used the colors: red, orange, yellow, green, cyan, azure, blue, violet, magenta, rose, grey, black, tan, white, purple, and silver as critical colors to be extracted. The color descriptions were extracted using regex searches to a new column, 'UAScolor .'

Due to the data collection and recording inconsistencies over the years, some of the reports have different formats. As shown in Figure 2-10 below, some of the reports had multiple internal headers *(PRELIM INFO FROM FAA OPS:, MOR Alert for, Number:, Type, Date/time:, A/C:, and Summary:)*. These allowed for simplified data extraction because we could use regex to search for these formatted patterns and easily access the data. However, the structure of the more recent reports did not include these consistent headers and usually only included the first section labeled PRELIM INFO.



PRELIM INFO FROM FAA OPS: ORMOND BEACH, FL/UAS INCIDENT/1238E/DAYTONA APRCH ADVISED CESSNA C172 AT 1,000 FEET ON RIGHT DOWNWIND RUNWAY 17 REPORTED UAS AT 200 FEET ABOVE HEADING S BOUND. NO CONFLICTS REPORTED. VOLUSIA COUNTY PD.

MOR Alert for DAB

Number: DAB-M-2015/01/23-0002

Type: Public inquiry or concern (including all pilot reported NMACs)

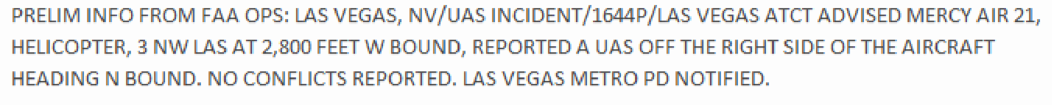
Date/Time: Jan 23, 2015 - 1809Z

A/C: C172

Summary: OMN TOWER REPORTED GA ACFT SIGHTED A UAV 200' ABOVE THE AIRCRAFT WHICH WAS AT 1000' IN OMN'S PATTERN.THE UAV WAS ON THE RIGHT DOWNWIND LEG OF RY17 AT OMN.DEN AND ROC NOTIFIED. VOULISA COUNTY SHERIFF'S OFFICE NOTIFIED.

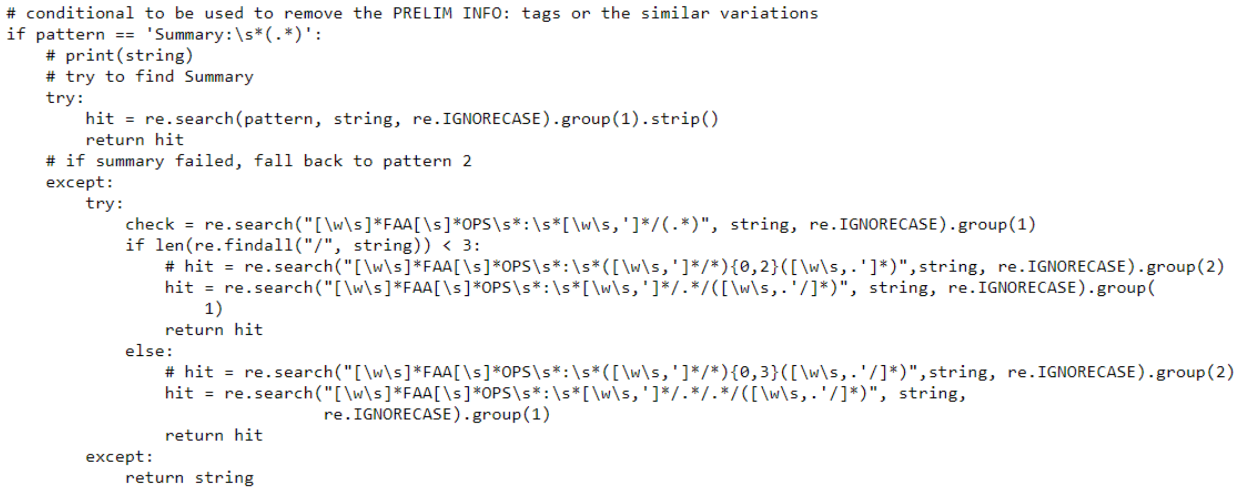
Figure 2-10. Additional details some of the UAS sighting report records include

In order to run an NLP model on the text, we had to extract text that only included information from the report that would add value to the model training. As seen in Figure 2-10 above, some files include a clean and easily accessible summary of the UAS sighting at the end of the record; this was extracted when it was available. Since this report formatting was deprecated, the team was required to use regex pattern searches created around how many forward slashes ("/") were contained in the text. A report of the sighting was in every UAS incident after the last forward slash; therefore, the number of forwarding slashes in the report text were counted, and all the text after the last slash was extracted as the UAS sightings report when a formatted summary was missing as Figure 2-11 below.



PRELIM INFO FROM FAA OPS: LAS VEGAS, NV/UAS INCIDENT/1644P/LAS VEGAS ATCT ADVISED MERCY AIR 21, HELICOPTER, 3 NW LAS AT 2,800 FEET W BOUND, REPORTED A UAS OFF THE RIGHT SIDE OF THE AIRCRAFT HEADING N BOUND. NO CONFLICTS REPORTED. LAS VEGAS METRO PD NOTIFIED.

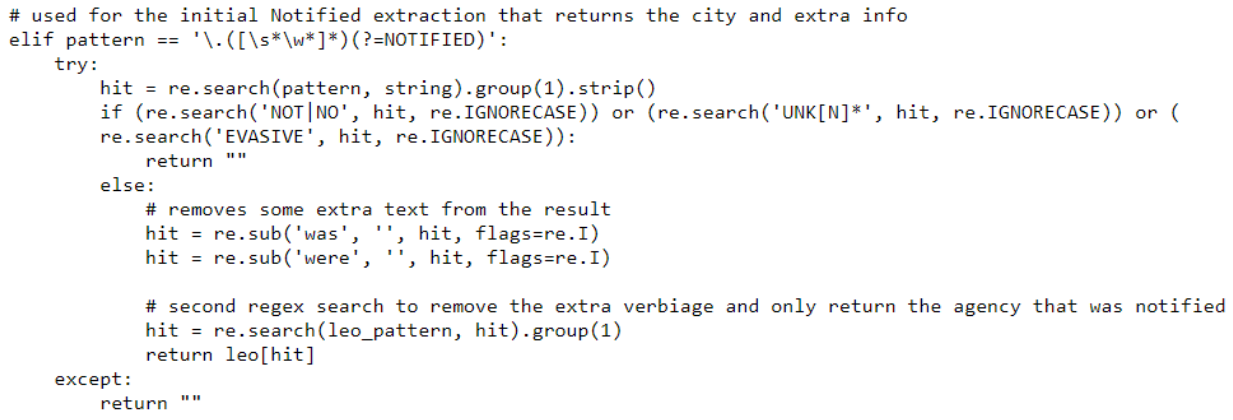
Figure 2-11. A UAS sighting report record without a formatted summary and additional details



```
# conditional to be used to remove the PRELIM INFO: tags or the similar variations
if pattern == 'Summary:\s*(.*)':
    # print(string)
    # try to find Summary
    try:
        hit = re.search(pattern, string, re.IGNORECASE).group(1).strip()
        return hit
    # if summary failed, fall back to pattern 2
    except:
        try:
            check = re.search("[\w\s]*FAA[\s]*OPS\s*:\s*[\w\s,']*/(.*)", string, re.IGNORECASE).group(1)
            if len(re.findall("/", string)) < 3:
                # hit = re.search("[\w\s]*FAA[\s]*OPS\s*:\s*([\w\s,']*/*){0,2}([\w\s,.']*)",string, re.IGNORECASE).group(2)
                hit = re.search("[\w\s]*FAA[\s]*OPS\s*:\s*[\w\s,']*/.*/([\w\s,.'/]*)", string, re.IGNORECASE).group(
                    1)
                return hit
            else:
                # hit = re.search("[\w\s]*FAA[\s]*OPS\s*:\s*([\w\s,']*/*){0,3}([\w\s,.'/]*)",string, re.IGNORECASE).group(2)
                hit = re.search("[\w\s]*FAA[\s]*OPS\s*:\s*[\w\s,']*/.*/.*/([\w\s,.'/]*)", string,
                                re.IGNORECASE).group(1)
                return hit
        except:
            return string
```

Figure 2-12. Python Code – Looking up internal headers and additional summary where available

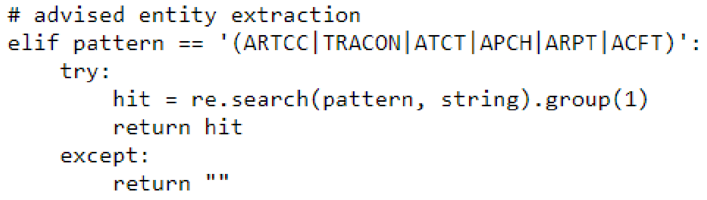
To extract the law enforcement agency involved or contacted, we used a look around with the word 'Notified' and extracted the details in that sentence before the word 'Notified.' A look around in regex searches the text for the pattern but only extracts it if the look around is also included in the text. In this scenario, our pattern was looking for any number of combinations of spaces and letters followed by the word "NOTIFIED". The values were extracted to the new column, 'Notified'. The extracted name was then looked up in our law enforcement standardization dictionary to provide consistency.



```
# used for the initial Notified extraction that returns the city and extra info
elif pattern == '\.([\s*\w*]*)(?=NOTIFIED)':
    try:
        hit = re.search(pattern, string).group(1).strip()
        if (re.search('NOT|NO', hit, re.IGNORECASE)) or (re.search('UNK[N]*', hit, re.IGNORECASE)) or (
        re.search('EVASIVE', hit, re.IGNORECASE)):
            return ""
        else:
            # removes some extra text from the result
            hit = re.sub('was', '', hit, flags=re.I)
            hit = re.sub('were', '', hit, flags=re.I)

            # second regex search to remove the extra verbiage and only return the agency that was notified
            hit = re.search(leo_pattern, hit).group(1)
            return leo[hit]
    except:
        return ""
```

Figure 2-13. Python Code – Lookup and extraction of law enforcement agency notified

The next portion that was extracted was the report receiving entities such as a TRACON or ATCT. Report receiving control towers were extracted out to a new column, 'Advice,' using a regex pattern containing the entities of interest in the same way that the color of the UAS was extracted. We created a specific list of names (ARTCC, TRACON, ATCT, APCH, ARPT, ACFT) and extracted them when they were included in a report.



```
# advised entity extraction
elif pattern == '(ARTCC|TRACON|ATCT|APCH|ARPT|ACFT)':
    try:
        hit = re.search(pattern, string).group(1)
        return hit
    except:
        return ""
```

Figure 2-14. Python Code - Extracting control tower receiving report

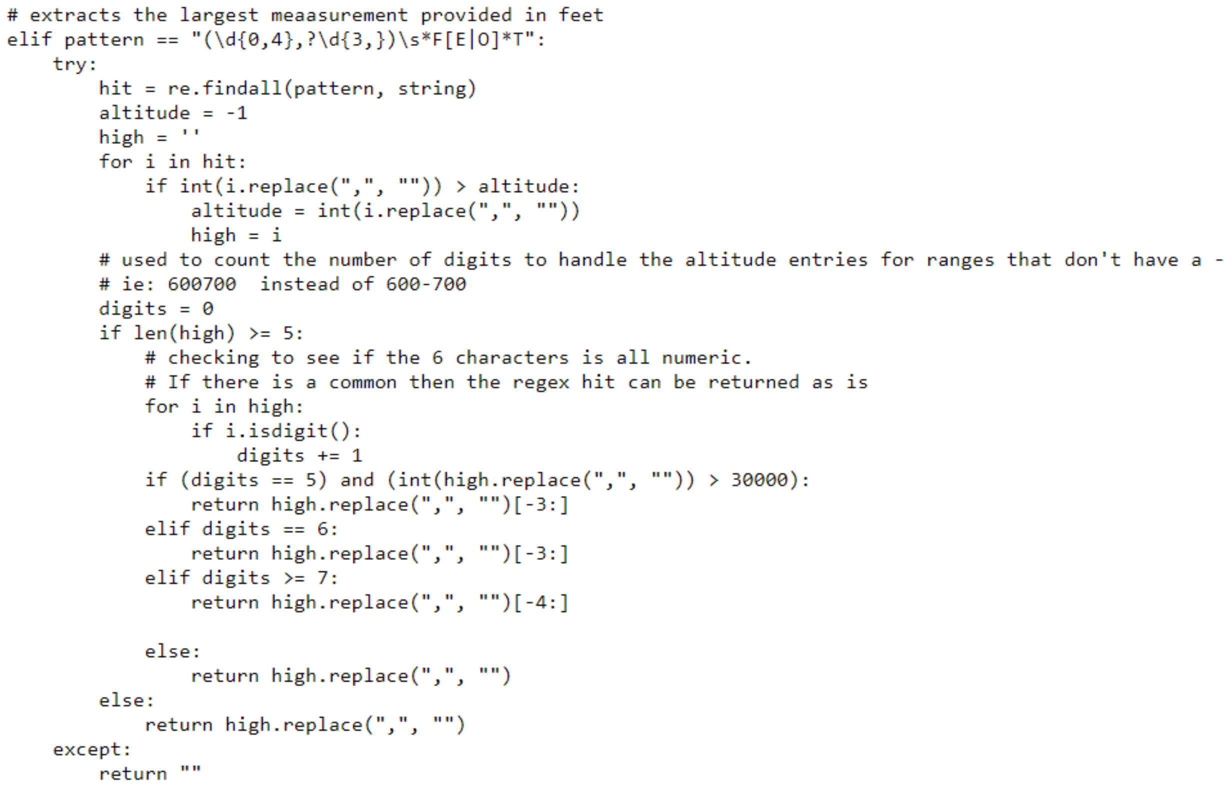
The time recorded in the reports and the time zone abbreviation were also extracted to two new columns, 'time' and ' timezone.' They were consistently formatted in the reports as a four-digit numerical value in the 24-hour format, followed by a letter representing the time zone. This made for easy extraction using a simple regex pattern.

'time': '(\d{4})\w'

'timezone': '\d{4}(\w)'

The last new column from the extraction of the summary section is the altitude in feet described in the reports. For this value, every numerical measurement in feet was extracted into a list, and the largest measurement was used as the reporting altitude. We decided to use this method for simplicity over absolute accuracy because the reported altitudes are only estimates provided by the pilots. The FAA advised that it was an acceptable method for the data they wanted to do.

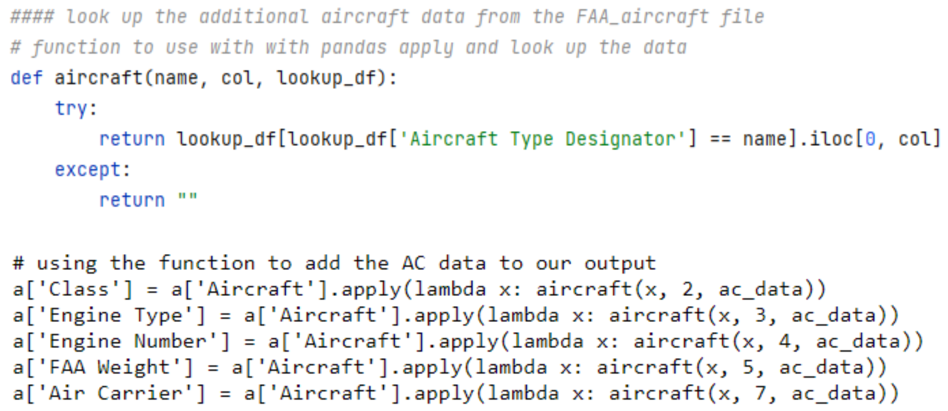
We encountered some edge cases when a range was provided for the altitude. The reports included instances of ranges, appearing as 600-700 feet but also 600700 feet, as an example, when a hyphen was not typed. We solved most of the edge cases by estimating where a hyphen would have been and extracting the maximum value of the range. For example, a UAS sighted over 20,000 feet is not accurate. Any altitude extracted at least five digits and more significant than 20,000 feet was assumed to be a range missing the hyphen. The expected location of the hyphen was calculated, and the upper limit of the range was used as the altitude measurement.



```
# extracts the largest meaasurement provided in feet
elif pattern == "(\d{0,4},?\d{3,})\s*F[E|O]*T":
    try:
        hit = re.findall(pattern, string)
        altitude = -1
        high = ''
        for i in hit:
            if int(i.replace(",", "")) > altitude:
                altitude = int(i.replace(",", ""))
                high = i
        # used to count the number of digits to handle the altitude entries for ranges that don't have a -
        # ie: 600700  instead of 600-700
        digits = 0
        if len(high) >= 5:
            # checking to see if the 6 characters is all numeric.
            # If there is a common then the regex hit can be returned as is
            for i in high:
                if i.isdigit():
                    digits += 1
            if (digits == 5) and (int(high.replace(",", "")) > 30000):
                return high.replace(",", "")[-3:]
            elif digits == 6:
                return high.replace(",", "")[-3:]
            elif digits >= 7:
                return high.replace(",", "")[-4:]

            else:
                return high.replace(",", "")
        else:
            return high.replace(",", "")
    except:
        return ""
```

Figure 2-14. Python Code - Extracting the altitude in feet

After extracting the needed data from the raw text summary of the UAS sightings dataset, the next step is to merge the additional details from the Aircraft Information dataset based on the aircraft name that was exported in the UAS report. The five columns Class, Engine Type, Engine Number, FAA Weight, and Air Carrier were added to the final dataset based on a lookup from the aircraft names. Since only aircraft names included in the FAA Aircraft Information data were extracted from the UAS reports, the team could easily append the requested data.



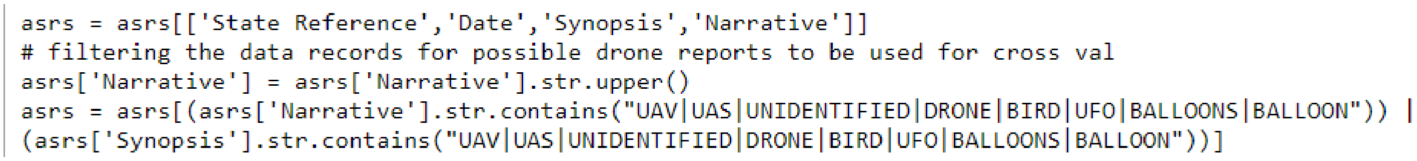
```
#### look up the additional aircraft data from the FAA_aircraft file
# function to use with with pandas apply and look up the data
def aircraft(name, col, lookup_df):
    try:
        return lookup_df[lookup_df['Aircraft Type Designator'] == name].iloc[0, col]
    except:
        return ""

# using the function to add the AC data to our output
a['Class'] = a['Aircraft'].apply(lambda x: aircraft(x, 2, ac_data))
a['Engine Type'] = a['Aircraft'].apply(lambda x: aircraft(x, 3, ac_data))
a['Engine Number'] = a['Aircraft'].apply(lambda x: aircraft(x, 4, ac_data))
a['FAA Weight'] = a['Aircraft'].apply(lambda x: aircraft(x, 5, ac_data))
a['Air Carrier'] = a['Aircraft'].apply(lambda x: aircraft(x, 7, ac_data))
```

Figure 2-15. Python Code - Merging details from the additional aircraft information dataset

2.4.6 Filtering ASRS Dataset

The initial data download of the complete ASRS dataset contained more than 100 fields, but the team only used the 'Narrative' and 'Synopsis' fields to keep records relevant to the project. The ASRS database collects reports regarding all sorts of aviation issues and sections, so the initial data download contents were too broad for cross-validation. Only a few reports were labeled under the UAS category, which required the team to look for other ways to build a cross-validation data set.

We decided to look at the actual raw text summary of the incident reports and use reports that contained anything we considered a possible UAS incident, even if the data was not labeled as one. The key words *UAV, UAS, UNIDENTIFIED, DRONE, BIRD, UFO, BALLOONS, and BALLOON* were used to filter such records. The team also retained the 'State' and 'Date of incident' from the original download. Additional standardization was done to the extracted 'State' field to match our naming format. Additionally, the year and month of reports were extracted out to two separate columns for cross-validation.



```
asrs = asrs[['State Reference','Date','Synopsis','Narrative']]
# filtering the data records for possible drone reports to be used for cross val
asrs['Narrative'] = asrs['Narrative'].str.upper()
asrs = asrs[(asrs['Narrative'].str.contains("UAV|UAS|UNIDENTIFIED|DRONE|BIRD|UFO|BALLOONS|BALLOON")) |
(asrs['Synopsis'].str.contains("UAV|UAS|UNIDENTIFIED|DRONE|BIRD|UFO|BALLOONS|BALLOON"))]
```

Figure 2-16. Python Code - Filtering initial ASRS database download

2.5 Data Quality Assessment:

2.5.1 FAA UAS Sightings Report Dataset

- Completeness – The dataset has no null values. Each record is complete with information on the date of the report, location information including city and state, and some version of the report made by the reporting party.

- Consistency– The dataset differs between the years regarding how data was recorded. Information was merged into one area after around 2019. The amount of information provided in the summary section of the report lacks consistency depending on the time and who recorded the report.
- Uniqueness– The dataset has few duplicate reporting with a slight language change. The dataset does not have a unique identifier, so there could be more duplicates.
- Conformity– The dataset had a few areas that were not standardized in dates. The summary section of the data included numerical as well as text data. This was dealt with through the data cleaning process.
- Accuracy– Values are accurate to the pilot is reporting, but the report is not verified to be true.

2.5.2 ASRS Dataset

- Completeness – The dataset has many null values. Not all records in the dataset downloaded apply to UAS or the purposes of this project. The records have missing information for fields relevant to the project, such as aircraft type.
- Consistency– The dataset is not consistent in the reporting as often even when they have a location in the narrative; the database reports it defaults as 'ZZZ' Airport tower with 'US' as the state. Many records do not tell the whole story; therefore, the team will rely mainly on the narrative and synopsis fields.
- Uniqueness– The dataset is unique as it has an identifier for the ASRS reports.
- Conformity– Dataset has multiple fields that could be considered duplicates. The columns used for this project are in a standard format.
- Accuracy– The accuracy of the dataset is slim since it has many default values being used in records, and all reports are submitted without verification.

2.6 Other Data Sources

The team considered using local law enforcement UAS incident reports in our cross-validation stage but could not incorporate any of them due to the time constraints of the course. We found access too late in the term. They would have increased the number of known actual sightings, which would be a possible improvement with future work. Through the NYPD, New York City and San Francisco, through their city government, provide UAS reports. These are both locations of high report activity in our UAS incident dataset.

The FAA also had a small data set of less than 100 records consisting of confirmed UAS sightings around the DFW airport. The team manually cross-checked the reports against the UAS incident data set, resulting in no validated reports.

3 Analytics and Algorithms

Natural Language Processing (NLP) is a field of computer science that focuses on the interaction between machines and human language, giving machines the ability to understand and extract meanings from human language [23]. NLP enables an automated approach to handle speech or text and allows for purposes such as spam detection, digital assistants, autocomplete, search engine, text summarization, sentiment analysis, and more. We utilized similarity testing and topic modeling for cross-validation and modeling in our workflow.

After cleaning our dataset, extracting the needed information, and adding the additional aircraft details, we move on to the following steps: cross-validation of the historical UAS sightings reports and predictive modeling. To set up a predictive modeling pipeline, we needed to have our reports labeled as actual sightings or noise. The FAA could not provide any data that allowed us to label our historical data as such. As a result, the team used NLP techniques to estimate actual sightings. We considered multiple techniques to cross-validate our two datasets and used NLP modeling for text similarity with word embedding. The R library text2vec was used to accomplish this conversion.

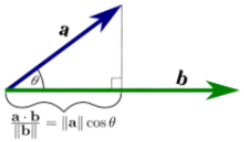
After cross-validation, we built a predictive modeling pipeline to predict if a report is a noise or an actual sighting. To accomplish this, we used the NLP technique of topic modeling. This allowed the team to develop a model which could automatically extract common topics and assign probabilities for an individual report belonging to each topic. The probabilities were then used as parameters in random forest and logistic regression models to predict if a report is an actual sighting or noise.

3.1 Cross Validation with Word Embedding and Cosine Similarity:

The NLP technique for similarity testing was used to avoid manually going through thousands of reports for cross-validation. Text similarity determines how similar two texts are to each other. This is accomplished with Word Embedding, which converts a text to a vector of numeric values for each text word. This vector can then be compared to another converted text vector by measuring the distance between the two. For this project, the differences are the distances of the two text vectors calculated using the cosine similarity [24].

Cosine Similarity is a method used to determine the similarity between two or more vectors, documents, or text. It can be calculated by measuring the cosine of the angle between two vector representations of text projected in a multi-dimensional space. The calculated measure for the cosine similarity testing has a value between 0 to 1. Each dimension in the multi-dimensional space represents a word in a given text group [19].

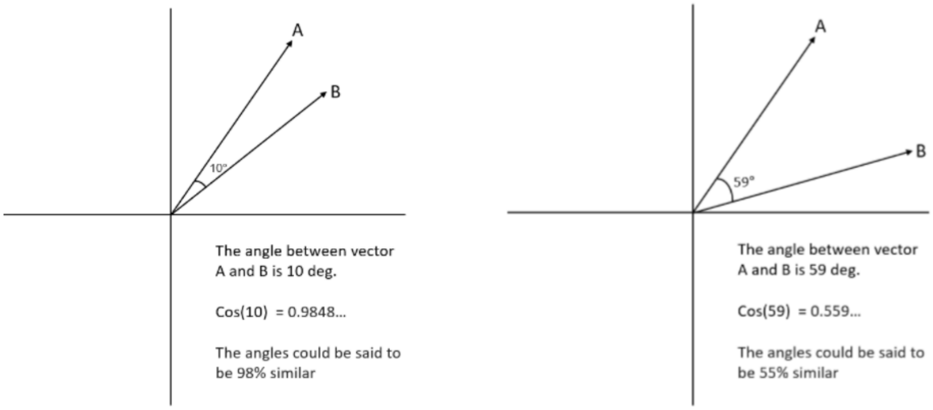
Mathematically, the cosine similarity of two vectors can be defined as the dot product of the vectors divided by the product of the length of each vector. The dot product of two vectors is the quantitative measure of how much one vector, vector A, is pointing in the same direction as another vector, vector B. Hence, the smaller the degree, meaning the more vector A is oriented similarly to vector B, the larger the value is. If the two vectors are perpendicular, meaning the angle between them is 90 degrees, the cosine value will be 0, indicating no similarity [25].


a
b
θ
$\frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{b}\|} = \|\mathbf{a}\|\cos\theta$

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Figure 3-1. Cosine Similarity formula and graphic

In Cosine Similarity, the closer the value is to 1, the smaller the angle between the two vectors and, therefore, the more similar the two vectors are [25].



The angle between vector A and B is 10 deg.

Cos(10) = 0.9848...

The angles could be said to be 98% similar

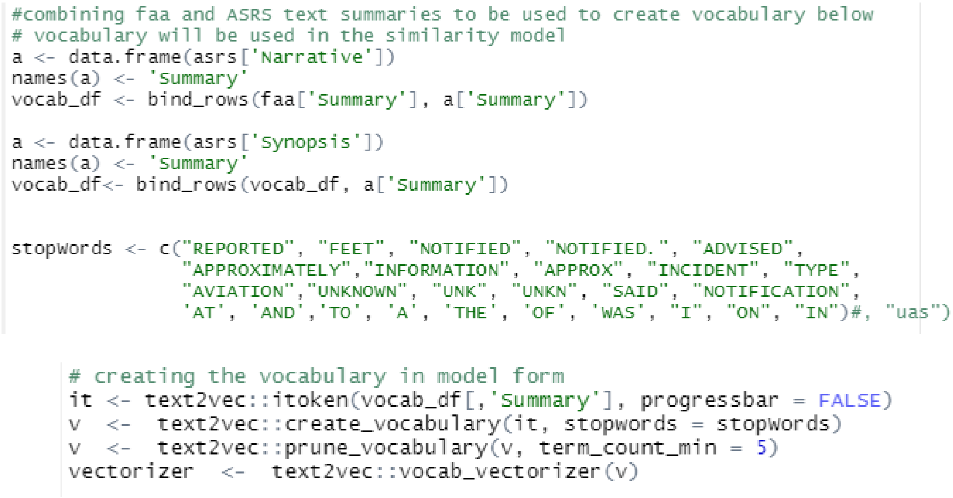
The angle between vector A and B is 59 deg.

Cos(59) = 0.559...

The angles could be said to be 55% similar

Figure 3-2. Cosine Similarity examples

We used the concepts of cosine similarity with the R package text2vec to cross-validate the FAA UAS sightings data and the ASRS dataset. The 'narrative' and 'summary' text from the filtered ASRS dataset and 'summary' from the cleaned FAA dataset was used to create a vocabulary in R. After removing stop-words from the data, the text's vocabulary was then vectorized using the text2vec function in R.

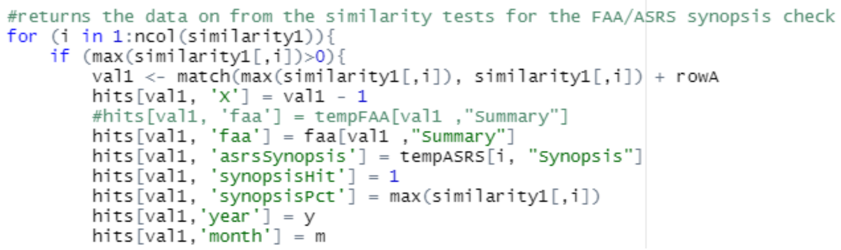


```
#combining faa and ASRS text summaries to be used to create vocabulary below
# vocabulary will be used in the similarity model
a <- data.frame(asrs['Narrative'])
names(a) <- 'Summary'
vocab_df <- bind_rows(faa['Summary'], a['Summary'])

a <- data.frame(asrs['Synopsis'])
names(a) <- 'Summary'
vocab_df<- bind_rows(vocab_df, a['Summary'])


stopwords <- c("REPORTED", "FEET", "NOTIFIED", "NOTIFIED.", "ADVISED",
               "APPROXIMATELY","INFORMATION", "APPROX", "INCIDENT", "TYPE",
               "AVIATION","UNKNOWN", "UNK", "UNKN", "SAID", "NOTIFICATION",
               'AT', 'AND','TO', 'A', 'THE', 'OF', 'WAS', "I", "ON", "IN")#, "uas")
```

```
# creating the vocabulary in model form
it <- text2vec::itoken(vocab_df[,'Summary'], progressbar = FALSE)
v  <-  text2vec::create_vocabulary(it, stopwords = stopwords)
v  <-  text2vec::prune_vocabulary(v, term_count_min = 5)
vectorizer  <-  text2vec::vocab_vectorizer(v)
```

Figure 3-3. R code – setting up Document-Term Matrix for similarity testing

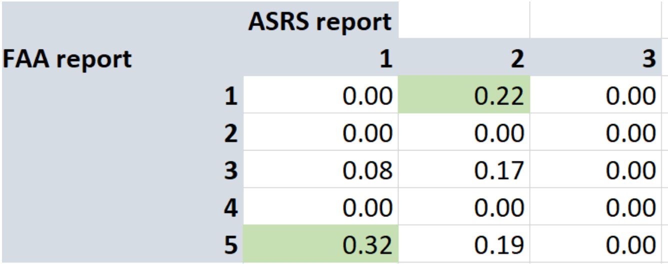
Since the ASRS dataset only had months and years provided for dates, the team grouped reports by month, year, and state in the UAS sightings dataset. The team then compared the text to every record in the same month, year, and state from the ASRS dataset 'narrative' and 'synopsis' texts. These comparisons are given a score between 0 to 1 depending on similarity. The highest match for each ASRS match was kept as an actual UAS report in the FAA dataset.



```
#returns the data on from the similarity tests for the FAA/ASRS synopsis check
for (i in 1:ncol(similarity1)){
    if (max(similarity1[,i])>0){
        val1 <- match(max(similarity1[,i]), similarity1[,i]) + rowA
        hits[val1, 'X'] = val1 - 1
        #hits[val1, 'faa'] = tempFAA[val1 ,"Summary"]
        hits[val1, 'faa'] = faa[val1 ,"Summary"]
        hits[val1, 'asrsSynopsis'] = tempASRS[i, "Synopsis"]
        hits[val1, 'synopsisHit'] = 1
        hits[val1, 'synopsisPct'] = max(similarity1[,i])
        hits[val1,'year'] = y
        hits[val1,'month'] = m
```

Figure 3-4. R code – storing of the cross-validation results

Figure 3-5 is an example of scoring and designation. There were five reports in the FAA data and three reports in the ASRS data for a matching month, state, and year between the two datasets. The similarity model found matches of varying similarity for ASRS reports 1 and 2 shown below, and the corresponding report from the FAA data with the highest score would be labeled as actual. Out of this group, there could be a maximum of three validated reports in the FAA data, and only two were labeled as actuals.



		ASRS report		
FAA report		**1**	**2**	**3**
	1	0.00	0.22	0.00
	2	0.00	0.00	0.00
	3	0.08	0.17	0.00
	4	0.00	0.00	0.00
	5	0.32	0.19	0.00

Figure 3-5. Example of similarity scores between a group of reports

The process for cross-validation outputs two files. The first file includes the hits for the cross-validation containing the narrative, synopsis, and summary texts, plus the month and year of the event and the value for the similarity score between 0 and 1. This allowed the team to review the reports that were matched as similar to make sure the matches were reasonable. Around 800 reports in the UAS incident report dataset received a match with an ASRS report. The team did not assume that they were all matches. After reviewing the matches, we decided that a similarity score of 0.25 or greater would be the threshold to label a UAS incident as actual. The threshold is subjective and was determined by the group after reviewing matches. A threshold value of 0.25 gave the team confidence that most of the modeled matches were true and included very few mislabels.

The second file output is the FAA cleaned and split data with an additional column added to the end, 'hit,' with a value of 0 for 'noise' and 1 for 'actual' indicating whether the specific report is a cross-validated hit or if it is noise. We ended up with 85 actual reports validated.

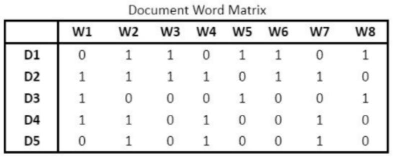
3.2 Topic Modeling with LDA

Once we had a cross-validation process completed, the following steps were to create models that would predict noise versus actuals in new sets of incoming reports. For this process, we used topic modeling, which is a type of modeling that allows for classifying documents into similar groups or topics.

- Topics, also referred to as themes, are statistically significant words in a corpus [26].
- Tokenization is the process of breaking text into smaller chunks or tokens, which can be words, sentences, or characters [27].
- N-gram in NLP refers to a sequence of words with the count of N. a bigram is a sequence of two words, the trigram is a sequence of three words, and n-gram is a sequence of n-words [28].

Topic modeling allowed us to create predictive models by extracting information from the reports. A popular example of topic modeling is Latent Dirichlet Allocation or LDA. LDA is used to identify the pattern of words that repeat together, occur together often, and are similar. It assigns a word a probabilistic score for a topic the word most potentially belongs to. LDA is used to group text to statistically significant topics. LDA assumes that topics are made up of words, and documents are made up of topics. The topics generate words using probabilistic distribution; therefore, documents are referred to as the probability distribution of topics, and topics are referred to as the probability distribution of words. [22] [26] [29]

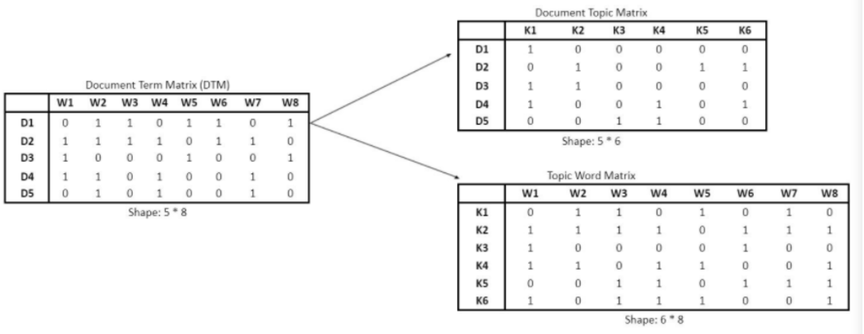
LDA uses a Document Term Matrix, or DTM, to represent the relationship between a document and a word after the documents have been cleaned and tokenized.



Document Word Matrix

	W1	W2	W3	W4	W5	W6	W7	W8
D1	0	1	1	0	1	1	0	1
D2	1	1	1	1	0	1	1	0
D3	1	0	0	0	1	0	0	1
D4	1	1	0	1	0	0	1	0
D5	0	1	0	1	0	0	1	0

Figure 3-6. A document- word matrix with the documents as rows and words as columns [29]



Document Term Matrix (DTM)

	W1	W2	W3	W4	W5	W6	W7	W8
D1	0	1	1	0	1	1	0	1
D2	1	1	1	1	0	1	1	0
D3	1	0	0	0	1	0	0	1
D4	1	1	0	1	0	0	1	0
D5	0	1	0	1	0	0	1	0

Shape: 5 * 8

Document Topic Matrix

	K1	K2	K3	K4	K5	K6
D1	1	0	0	0	0	0
D2	0	1	0	0	1	1
D3	1	1	0	0	0	0
D4	1	0	0	1	0	1
D5	0	0	1	1	0	0

Shape: 5 * 6

Topic Word Matrix

	W1	W2	W3	W4	W5	W6	W7	W8
K1	0	1	1	0	1	0	1	0
K2	1	1	1	1	0	1	1	1
K3	1	0	0	0	0	1	0	0
K4	1	1	0	1	1	0	0	1
K5	0	0	1	1	0	1	1	1
K6	1	0	1	1	1	0	0	1

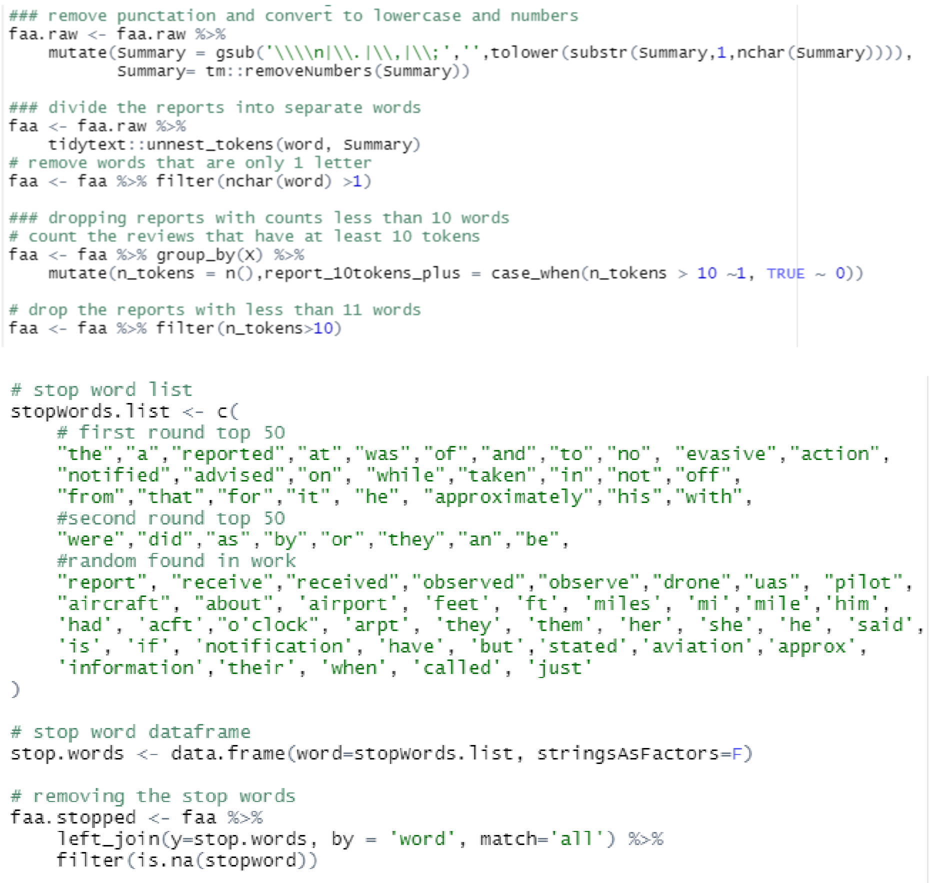
Shape: 6 * 8

Figure 3-7. A document-word matrix is converted to a document-topic matrix and a topic-word matrix [29]

The document-word matrix is further converted into two matrices, a document-topic matrix, which contains the topics in the document, and a topic-word matrix, which contains the topics. The goal of LDA is to optimize these two matrices to end up with the optimal distribution of both. This is an iterative process since documents, topics, and words are correlated. After many iterations for each word and each document, the goal is to come up with the most relevant topic for the current word in the iteration being evaluated, and the most probable topic is chosen using calculations for the proportion of the topic per document and word per topic. The product of the two proportions is used to estimate a new probability, and LDA classifies the new topic as the most relevant topic using product probability [29].

For our project, we used the topicmodels library in R to evaluate the topic models using LDA, the randomForest and caret package to train the predictive models, and the ggplot2 package to create visualizations of the model performance.

After loading the raw merged UAS sightings dataset, the dataset was further prepped for topic modeling by removing punctuations, standardizing all text to lowercase, removing reports with less than ten words, and removing stop words.

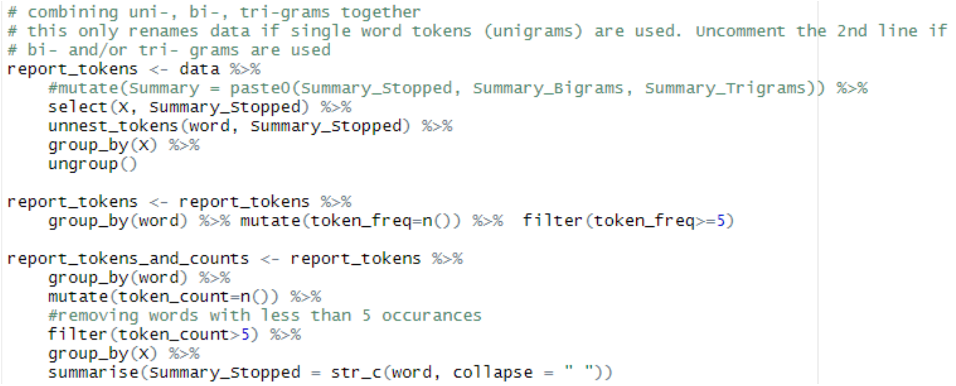


```r
### remove punctation and convert to lowercase and numbers
faa.raw <- faa.raw %>%
    mutate(Summary = gsub('\\\\n|\\.|\\,|\\;','',tolower(substr(Summary,1,nchar(Summary)))),
           Summary= tm::removeNumbers(Summary))

### divide the reports into separate words
faa <- faa.raw %>%
    tidytext::unnest_tokens(word, Summary)
# remove words that are only 1 letter
faa <- faa %>% filter(nchar(word) >1)

### dropping reports with counts less than 10 words
# count the reviews that have at least 10 tokens
faa <- faa %>% group_by(X) %>%
    mutate(n_tokens = n(),report_10tokens_plus = case_when(n_tokens > 10 ~1, TRUE ~ 0))

# drop the reports with less than 11 words
faa <- faa %>% filter(n_tokens>10)
```

```r
# stop word list
stopWords.list <- c(
    # first round top 50
    "the","a","reported","at","was","of","and","to","no", "evasive","action",
    "notified","advised","on", "while","taken","in","not","off",
    "from","that","for","it", "he", "approximately","his","with",
    #second round top 50
    "were","did","as","by","or","they","an","be",
    #random found in work
    "report", "receive","received","observed","observe","drone","uas", "pilot",
    "aircraft", "about", 'airport', 'feet', 'ft', 'miles', 'mi','mile','him',
    'had', 'acft',"o'clock", 'arpt', 'they', 'them', 'her', 'she', 'he', 'said',
    'is', 'if', 'notification', 'have', 'but','stated','aviation','approx',
    'information','their', 'when', 'called', 'just'
)

# stop word dataframe
stop.words <- data.frame(word=stopWords.list, stringsAsFactors=F)

# removing the stop words
faa.stopped <- faa %>%
    left_join(y=stop.words, by = 'word', match='all') %>%
    filter(is.na(stopword))
```

Figure 3-8. R code – topic modeling data cleaning code part 1

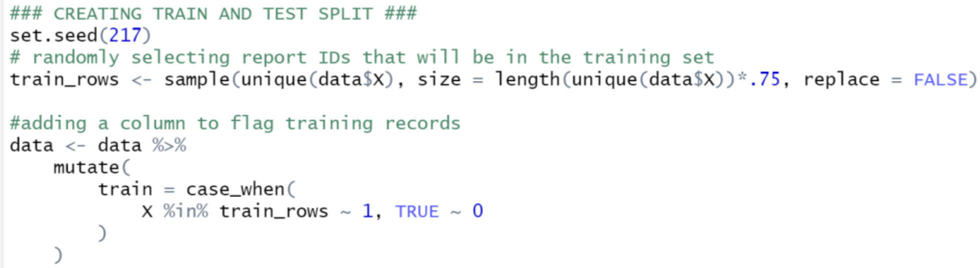
Next, we removed words that appeared in five or fewer reports across the over 13,000 total reports to limit words that may show up too infrequently to add value.



```
# combining uni-, bi-, tri-grams together
# this only renames data if single word tokens (unigrams) are used. Uncomment the 2nd line if
# bi- and/or tri- grams are used
report_tokens <- data %>%
    #mutate(Summary = paste0(Summary_Stopped, Summary_Bigrams, Summary_Trigrams)) %>%
    select(X, Summary_Stopped) %>%
    unnest_tokens(word, Summary_Stopped) %>%
    group_by(X) %>%
    ungroup()

report_tokens <- report_tokens %>%
    group_by(word) %>% mutate(token_freq=n()) %>%  filter(token_freq>=5)

report_tokens_and_counts <- report_tokens %>%
    group_by(word) %>%
    mutate(token_count=n()) %>%
    #removing words with less than 5 occurances
    filter(token_count>5) %>%
    group_by(X) %>%
    summarise(Summary_Stopped = str_c(word, collapse = " "))
```

Figure 3-9. R code – topic modeling data cleaning code part 2

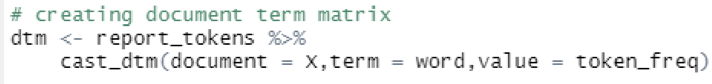
The data set was randomly split into training and test sets. We used a 75% training and 25% testing split. Given that we had such a rare outcome variable, we had to ensure it was represented in both training and testing sets. The training set had 63 actual reports, and the testing had 22 actual reports.



```
### CREATING TRAIN AND TEST SPLIT ###
set.seed(217)
# randomly selecting report IDs that will be in the training set
train_rows <- sample(unique(data$X), size = length(unique(data$X))*.75, replace = FALSE)

#adding a column to flag training records
data <- data %>%
    mutate(
        train = case_when(
            X %in% train_rows ~ 1, TRUE ~ 0
        )
    )
```

Figure 3-10. R code – splitting data set into training and testing sets

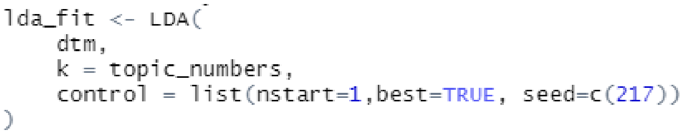
After the data was split into training and testing sets, the next step is to create a document term matrix that is used as input for building the LDA topic model. The DTM tracks the frequency of each term by document, in our case, by the report. The rows of the DTM represent each report, and the columns represent the relevant tokens.



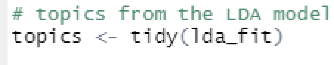
```
# creating document term matrix
dtm <- report_tokens %>%
    cast_dtm(document = X,term = word,value = token_freq)
```

Figure 3-11. R code – creating document-term matrix for topic modeling

The LDA model inputs the DTM created and the number of topics chosen as a tuning parameter. We explored topic groupings starting from two topics up to ten and determined that five topics provided the most distinct topics. This is a subjective decision, but the team was confident in our five groups.



```
lda_fit <- LDA(
    dtm,
    k = topic_numbers,
    control = list(nstart=1,best=TRUE, seed=c(217))
)
```



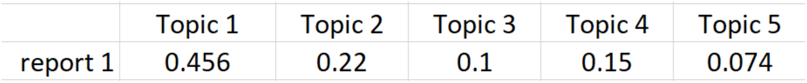
```
# topics from the LDA model
topics <- tidy(lda_fit)
```

Figure 3-12. R code – using LDA to create five topics from our training data

With our LDA topic model trained on our DTM, we could input individual reports and receive an output that was the probability the individual report belonged to each of the five topics. These five probabilities became our input parameters for a predictive model.

3.3 Predictive Modeling

The team has taken raw text reports, cleaned and parsed the data, created a model to determine five topics in our data, and can now utilize a predictive model using the topic model output. After inputting a report into a topic model, the output will be the probability that the report belongs to each topic.



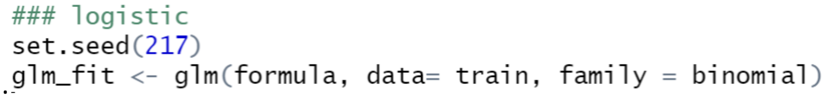
	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
report 1	0.456	0.22	0.1	0.15	0.074

Figure 3-13. Example of topic probabilities assigned to a single report after being processed through the trained LDA topic model

Every report in the training set was input into the topic model and received its topic probability assignments. These probabilities variables were then used to train predictive models as inputs. Two models were trained and tested - Logistic Regression and Random Forest.

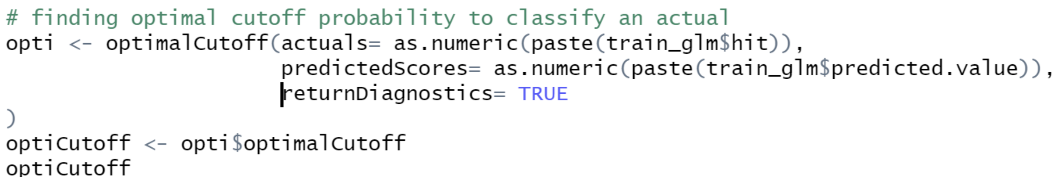
3.3.1 Logistic Regression

Logistic regression was utilized to perform predictive labeling. A commonly used linear regression outputs the probability of inputs belonging to a binary class. This serves our project ideally since we aim to classify a report as actual or noise. The output for our logistic regression will be a value between 0 and 1, and the training dataset was again used to train our logistic regression model



```
### logistic
set.seed(217)
glm_fit <- glm(formula, data= train, family = binomial)
```

After the model was trained, the optimal output threshold had to be determined for what would be labeled as actual or noise. We could not use model accuracy as our optimization guide because our output variable was rare. Accuracy over 99% could be achieved by classifying everything as noise with only 85 accurate reports of over 13,000 total records. It was determined that our goal for this predictive workflow would be to maximize the true positive classifications while minimizing the false positive classifications. R has a function called optimalCutoff in the InformativeValue library, which searches for that. The function is provided with the predictions from the training data and the actual outcomes for the same data, and it outputs the threshold to use on the logistic regression to minimize false positives while maximizing true positives.



```
# finding optimal cutoff probability to classify an actual
opti <- optimalCutoff(actuals= as.numeric(paste(train_glm$hit)),
                      predictedScores= as.numeric(paste(train_glm$predicted.value)),
                      returnDiagnostics= TRUE
)
optiCutoff <- opti$optimalCutoff
optiCutoff
```

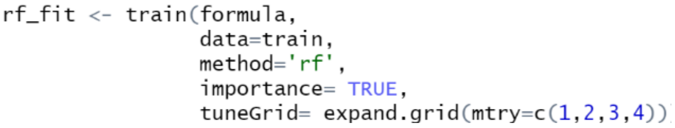
Figure 3-14. R code – optimization function used to determine the optimal classification threshold

The logistic regression model can now be trained and optimized for our goals. We can use the trained model on the test dataset and set the labeling threshold to our optimized cutoff.

3.3.2 Random Forest

Random forest was used as our second predictive model. Unlike the logistic regression model, random forest is a non-linear method. It creates a 'forest' by building many 'trees.' 'Trees' are created by splitting the data on a single variable into smaller subsets, e.g., topic 1 probability less than 0.2 would be group A and more significant than 0.2 would be group B. It then repeats splitting the data into the smaller subsets using another input variable. These splits are continued until the minimum number of records is in a group, so it cannot be split again, or the maximum number of splits (depth) is reached. These are both tunable model parameters. The number of trees, how many input variables to test at each split, and how the input variables are sampled are also tunable parameters.

For our random forest model, we held some tuning parameters constant - the number of trees was held constant at 500, the minimum number of records in a subset was held at one, and our maximum depth was equal to our input variables count. The number of input variables tested at each split was our tuning parameter. We saw the best performance testing four input variables at each split.



```
rf_fit <- train(formula,
                data=train,
                method='rf',
                importance= TRUE,
                tuneGrid= expand.grid(mtry=c(1,2,3,4)))
```

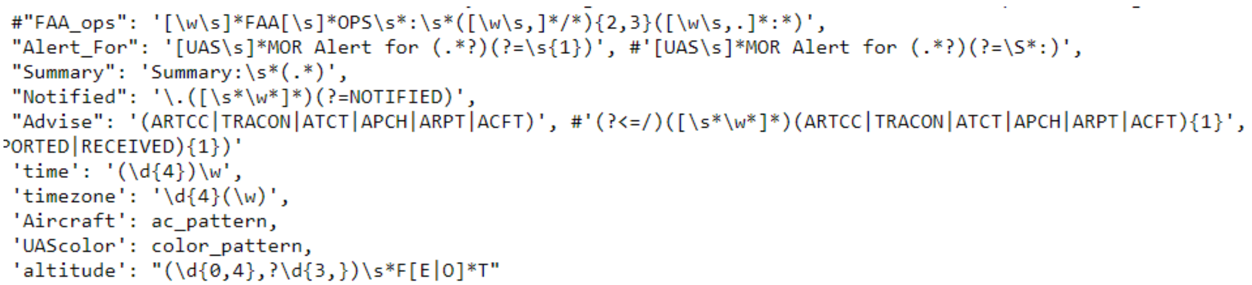
Figure 3-15. R code – function used to train the random forest model

We made our random forest output a probability of being an actual report and utilized the same threshold optimization process used on the logistic regression output.

3.4 Regular Expression

The team utilized regular expressions to handle the textual field of the FAA UAS Sightings dataset and extract essential details of the report. A regular expression is a sequence or pattern of characters used to match or search for a string. We used regular expressions specifically with the Python pandas package to extract essential details from the summary section of the UAS dataset that included the majority of the information about the reports [18].

We used regular expressions from the UAS sightings dataset to extract the additional summary details, the aircraft type, any color description mentioned, altitude in feet, the control tower advised, the notified law enforcement entity, and more.



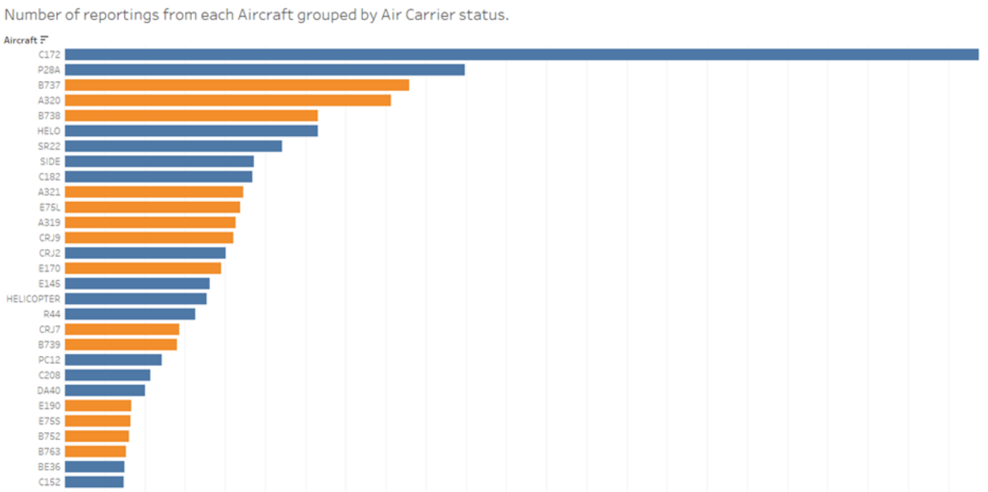
```
 #"FAA_ops": '[\w\s]*FAA[\s]*OPS\s*:\s*([\w\s,]*/*){2,3}([\w\s,.]*:*)',
 "Alert_For": '[UAS\s]*MOR Alert for (.*?)(?=\s{1})', #'[UAS\s]*MOR Alert for (.*?)(?=\S*:)',
 "Summary": 'Summary:\s*(.*)',
 "Notified": '\.([\s*\w*]*)(?=NOTIFIED)',
 "Advise": '(ARTCC|TRACON|ATCT|APCH|ARPT|ACFT)', #'(?<=/)([\s*\w*]*)(ARTCC|TRACON|ATCT|APCH|ARPT|ACFT){1}',
PORTED|RECEIVED){1})'
 'time': '(\d{4})\w',
 'timezone': '\d{4}(\w)',
 'Aircraft': ac_pattern,
 'UAScolor': color_pattern,
 'altitude': "(\d{0,4},?\d{3,})\s*F[E|O]*T"
```

Figure 3-16. Python code – regex patterns used to extract relevant data from the raw report text

4 Visualization

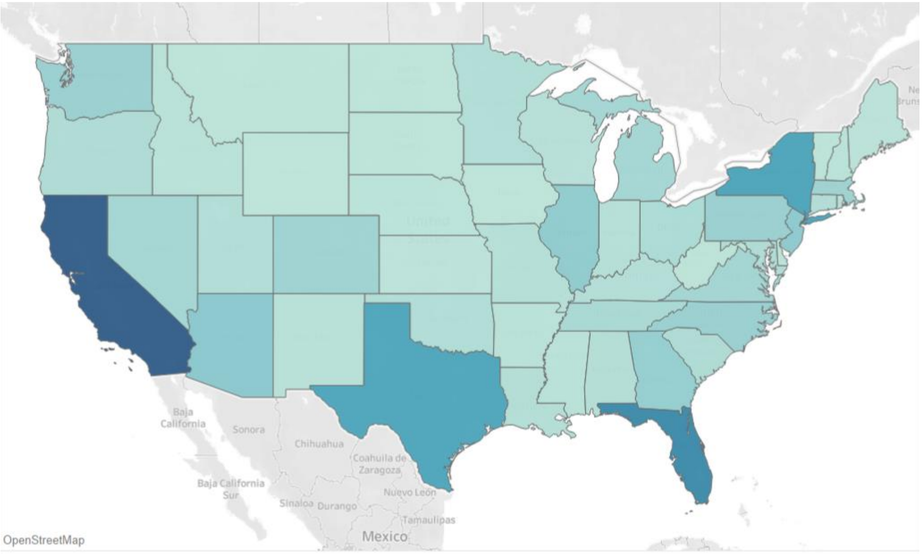
4.1 UAS Sightings Dataset Insights

For an initial analysis of our cleaner dataset, we have the following visualizations on aircraft type as well as heatmap distribution of reported sightings of drones.



Aircraft	
C172	1,138
P28A	498
B737	429
A320	407
B738	316
HELO	316
SR22	271
SIDE	236
C182	234
A321	223
E75L	219
A319	213
CRJ9	210
CRJ2	201
E170	195
E145	181
HELICOPTER	177
R44	163
CRJ7	143
B739	140
PC12	121
C208	107
DA40	100
E190	83
E75S	82

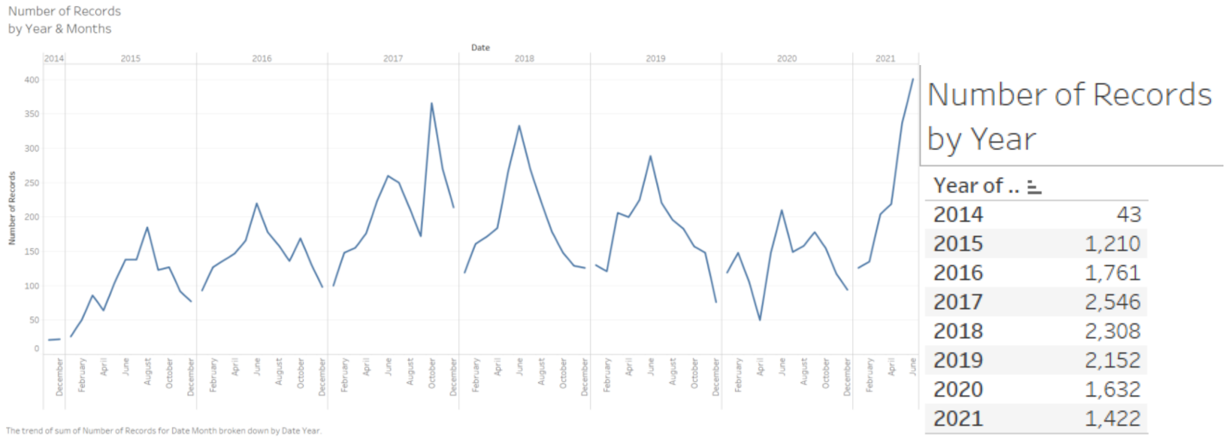
Figure 4-1. Aircraft counts from the FAA dataset. These aircraft names were extracted from the raw report text. They are segmented to show if the planes are operated by commercial airlines or used for general aviation. The aircraft type distribution of the reporting aircraft and the number of reports made by each with the most amount of reports made by Cessna C172, Piper P28A and Boeing B737.



Records Reported by State from 2014 to 2021

State	
CALIFORNIA	2,268
FLORIDA	1,540
NEW YORK	1,119
TEXAS	1,107
ARIZONA	498
NEW JERSEY	462
ILLINOIS	447
GEORGIA	400
NORTH CAROLINA	351
WASHINGTON	343
PENNSYLVANIA	342
MASSACHUSETTS	308
COLORADO	306
VIRGINIA	266
MICHIGAN	238
NEVADA	235
DC	206
TENNESSEE	195
OHIO	183
OREGON	143

Figure 4-2. A heatmap that shows the state distribution of reports made between 2015 and 2021 with California, Florida and New York having the highest number of reports.



Number of Records by Year

Year of ..	
2014	43
2015	1,210
2016	1,761
2017	2,546
2018	2,308
2019	2,152
2020	1,632
2021	1,422

Figure 4-3. Time series shows the trends in number of records reported each year by month.

4.2 NLP/Topic Model Pipeline Visuals

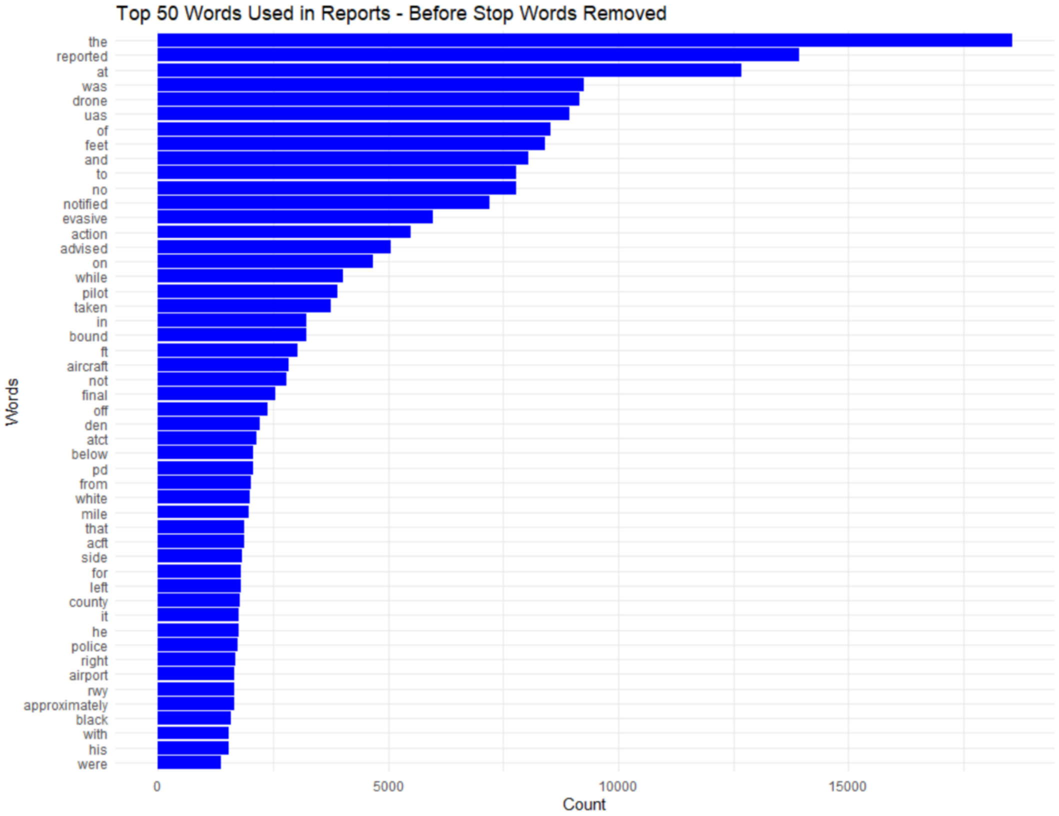


Figure 4-4. word count prior to removal of stop words

Figure 4.4 is the word count before removing stop words. There appear to be primarily words that will add no accurate information to the NLP models because they are either so frequent that it is indistinguishable from one report to the next, or they are words that do not add value to topic modeling – such as for, that, they, were, where, while, and other forms.

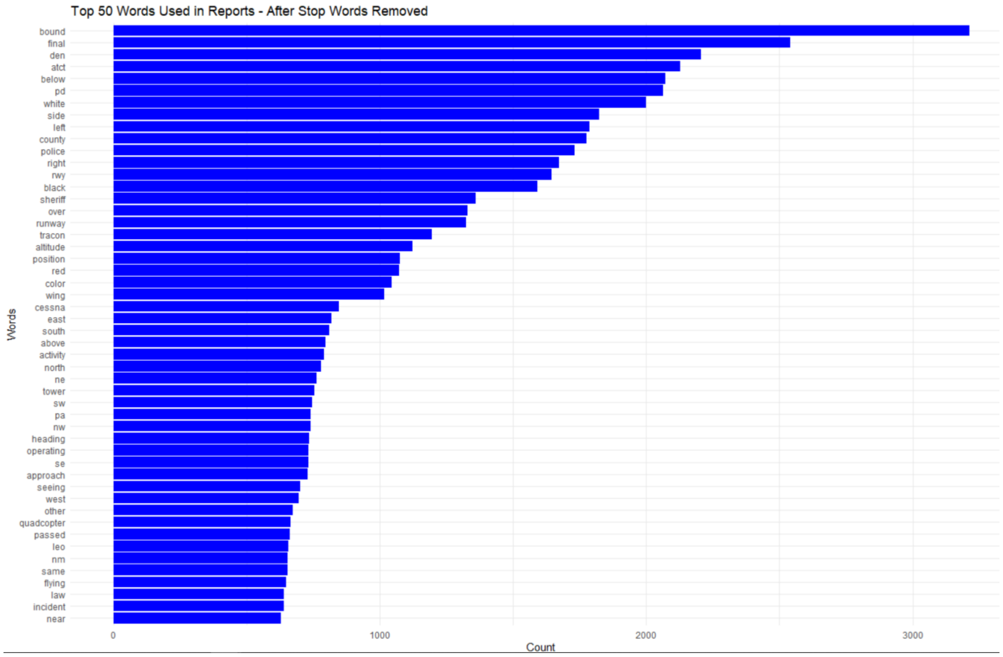


Figure 4-5. word count after stop word processing

The word count after stop-words are removed in Figure 4.5. There is a much tighter range in the top 50, and no word appears more than 4,000 times over the 13,000 reports, which should leave us with more impactful NLP models.

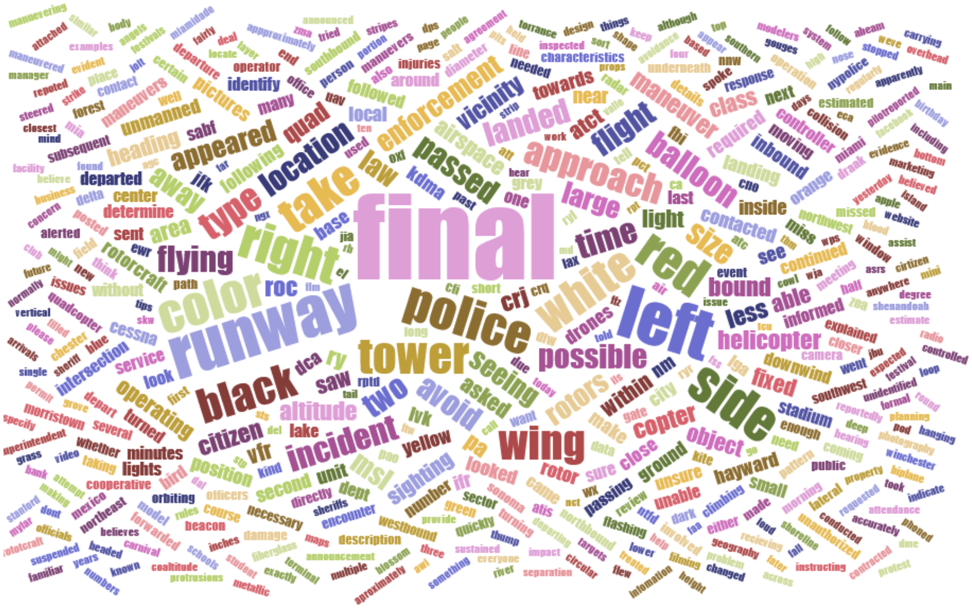


Figure 4-6. word cloud from the 85 actual reports

Figure 4.6 is the word cloud from the 85 reports determined to be true, and Figure 4.7 is the word cloud for the reports labeled as noise. These were created after stop words were removed and the data was processed for NLP. We utilized the word clouds to look for other possible stop words to remove.

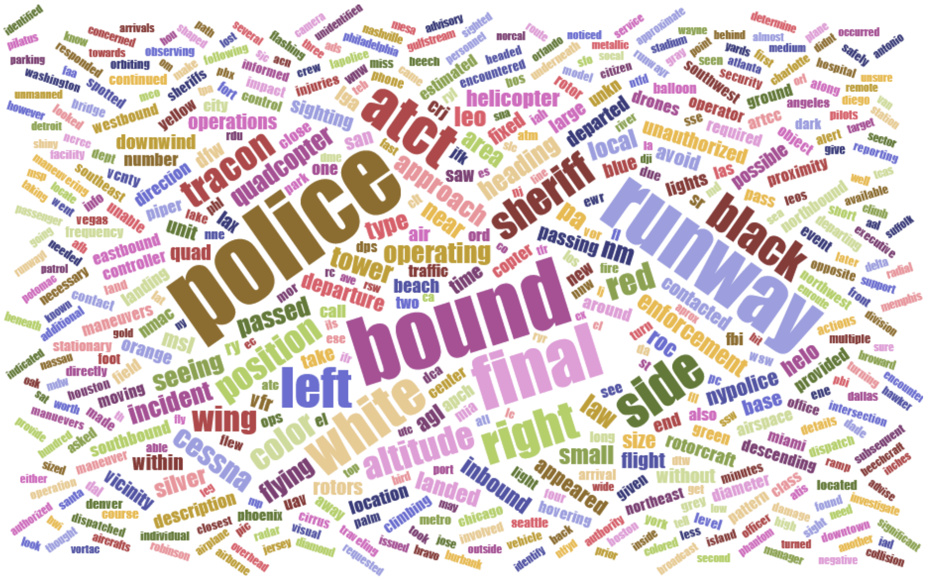


Figure 4-7. word cloud from the remaining reports labeled as noise

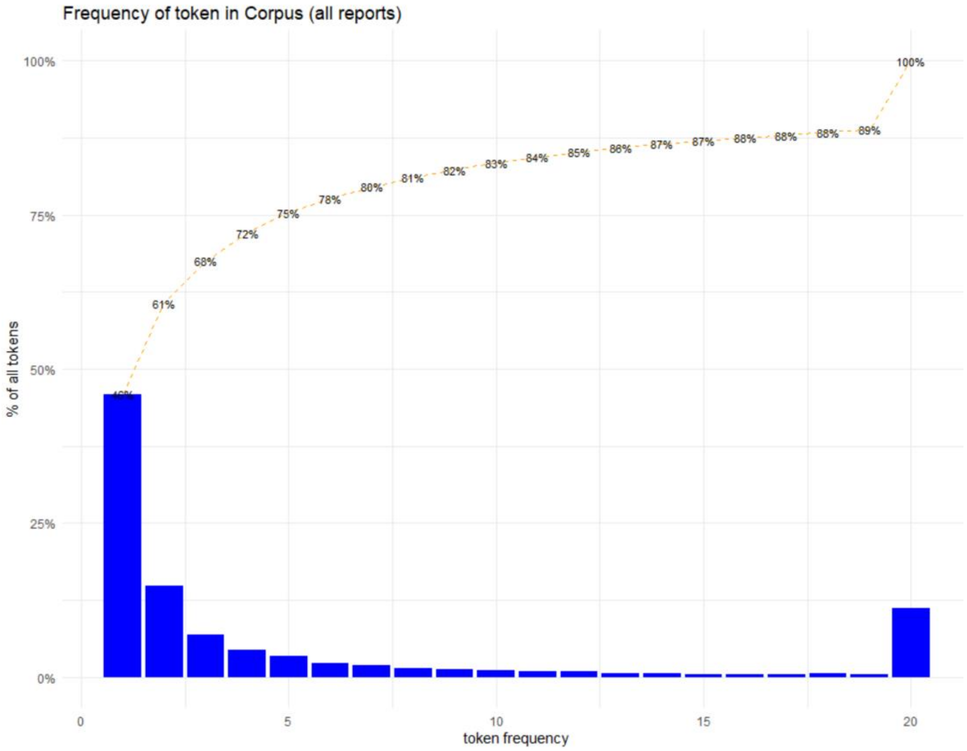


Figure 4-8. Bar chart shows the word frequency, and the line plot shows cumulative percentage of the word's occurrences

We explored the token (word) frequency across all the reports. 75% of the words appear less than six times in over 13,000 reports. Words that appear less than six times were removed from our NLP models to keep the included text relevant. Something that appears infrequently is not going to provide much info to the models.

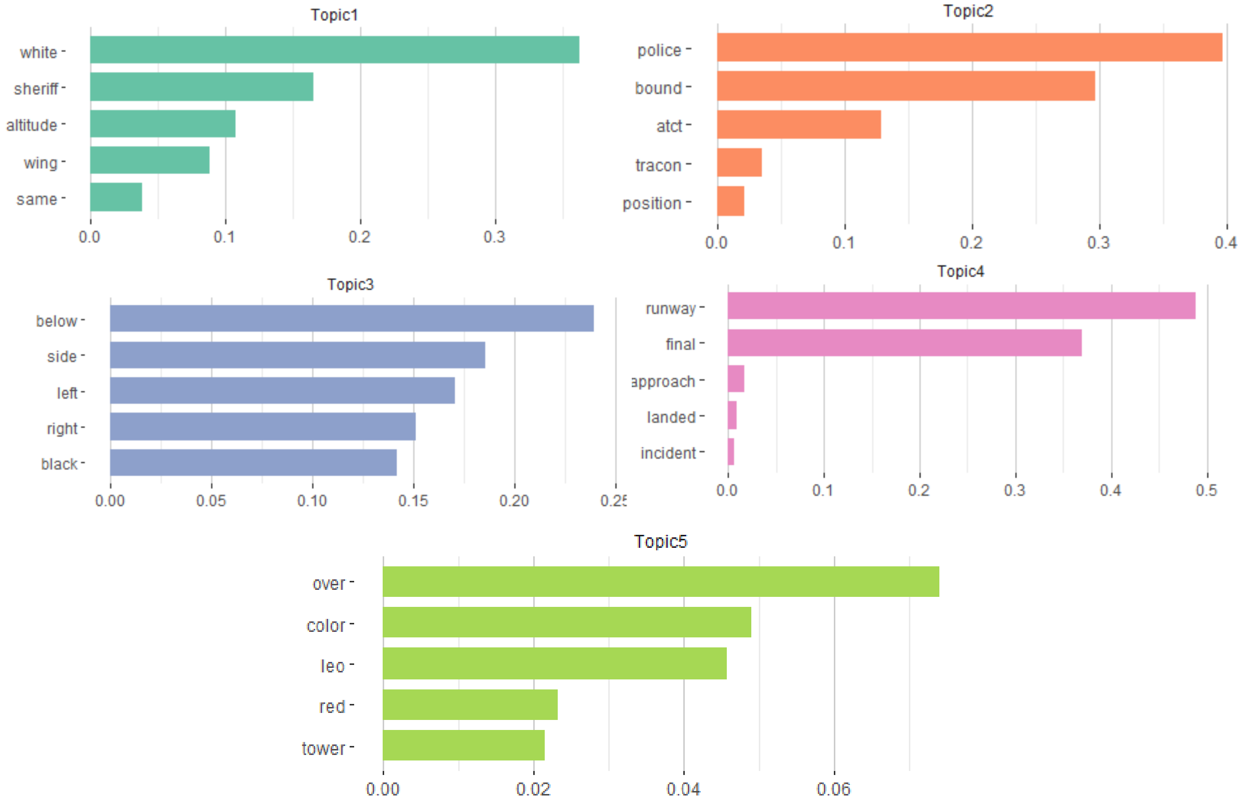


Figure 4-9. The top 5 words by probability of appearance in each of our 5 topics

Using text2vec, we created a topic modeling algorithm to uncover five separate topics. Figure 4.9 shows the top 5 tokens' probability of appearance in the respective topic. Topic 1 appears to be reported to the sheriff about UAS at the same altitude as the reporting aircraft. Topic 2 appears to be aircraft reporting to the police, air traffic control, or a Tracon. Topic 3 appears to be reports of UAS on either side of the aircraft. Topic 4 appears to be reports of UAS incidents as planes are on their final descent. Topic 5 appears to be reported to law enforcement or the tower about colored UAS over the aircraft. We explored different topics but felt that 5 captures the unique topics with the most negligible overlap.

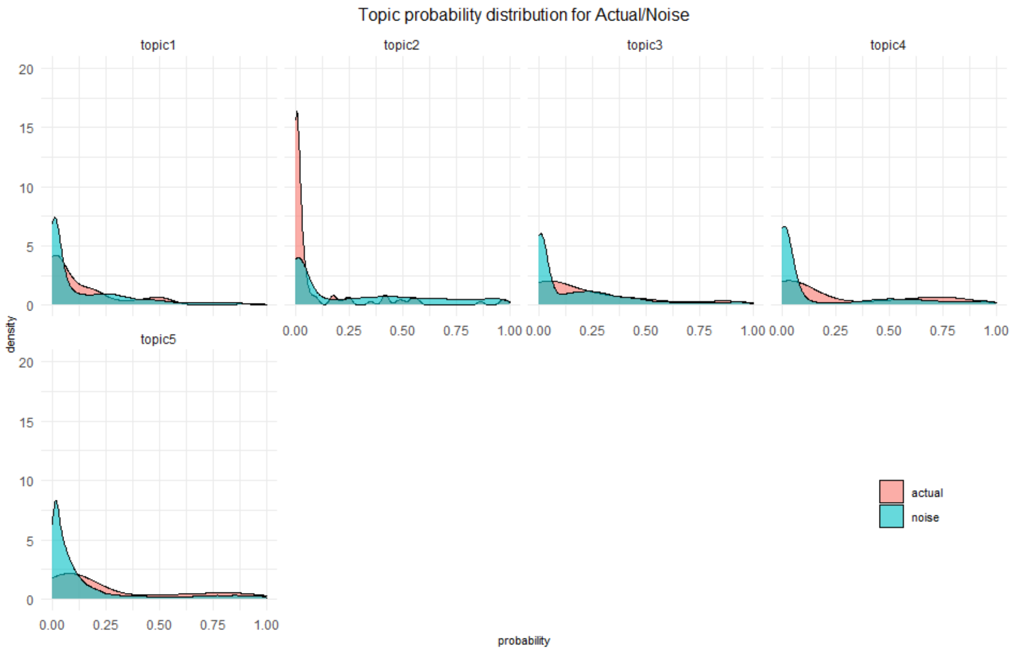


Figure 4-10. density plots for each topic showing the difference between actual reports and noise reports

Looking at the topic probability densities for actual vs. noise reports, topic 1, topic 2, and topic 3 have a higher probability of noise, while topic 4 and topic 5 have a little better chance of being an actual sighting. With our topics established, we were now able to move on to predictive modeling that can use the probability of a report belonging to a topic to predict if it is noise or actual.

4.3 Predictive Modeling Using Topic Modeling

Figure 4.11 is the ROC curve from our best-performing model, AUC = 0.66. This puts the true positive rate around 0.63 and the false positive rate around 0.32. The lack of reports known to be truly limited the team's ability to train a predictive model that was a solid all-around predictor. As a result of this rare outcome variable, Accuracy [(true positive + true negative) / (true pos. + true neg. + false pos. + false neg.)] is not a viable metric to use for model performance since we could predict everything to be noise and still almost hit 100% accuracy (3109/3132=99.3%; with all 3109 being true negatives). We decided to aim to minimize false positives while maximizing true positives.

The confusion matrix in Figure 4.12 is for our best logistic regression. Expanding on how we optimized our model, we believed that capturing the possible true positives while limiting the false positives would use the FAA. Our target was to get about 2/3 of the true positives while only missing 1/3 of the noise.

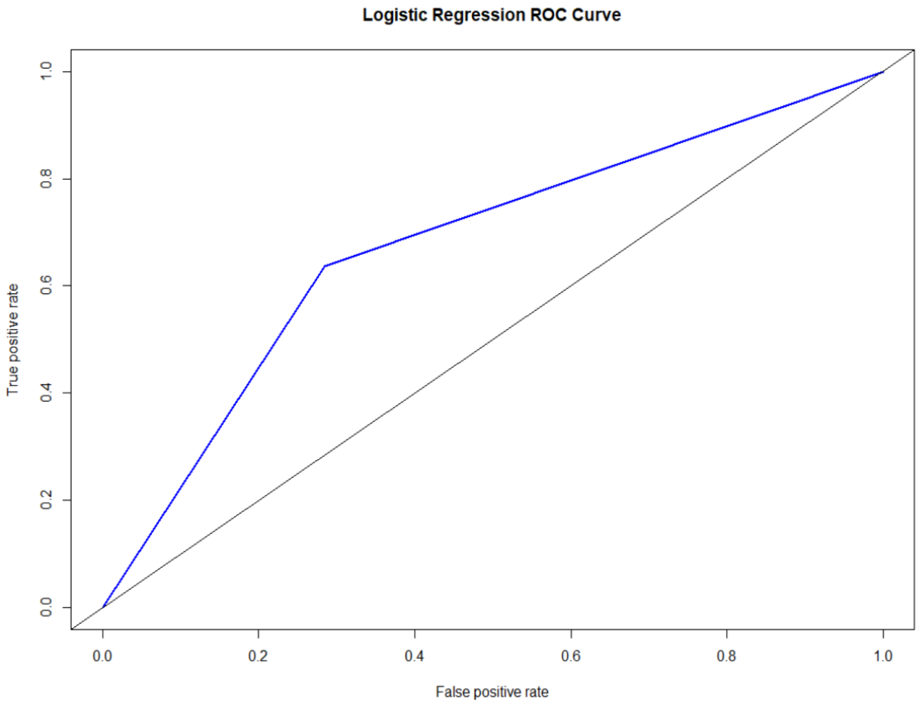


Figure 4-11. ROC curve produced from our optimal logistic regression.

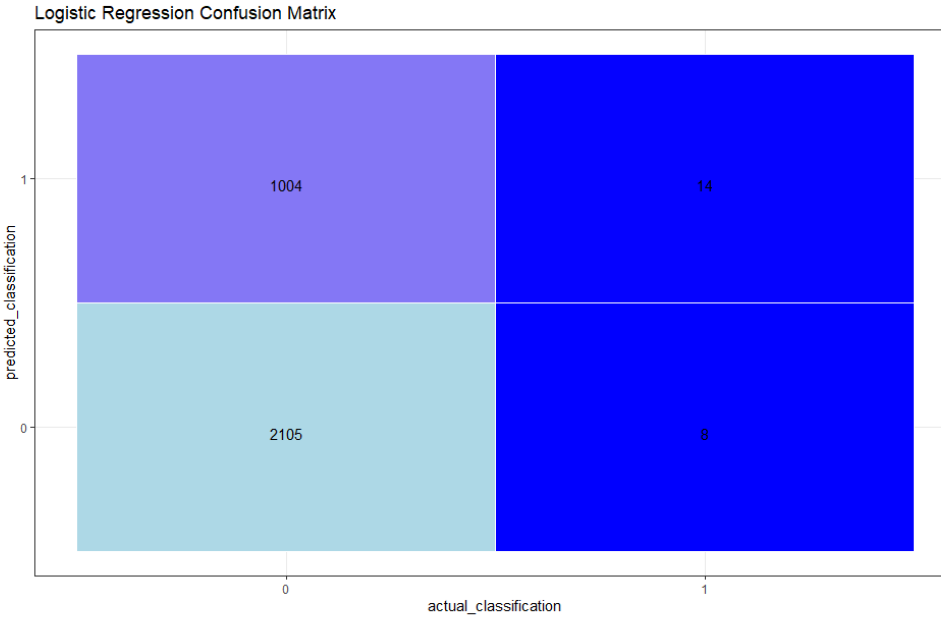


Figure 4-12. Confusion matrix from our logistic regression model

Next, the team looked to improve the logistic regression model by using a Random Forest model. This model performed much worse than the logistic regression model and could be considered nearly as valuable as a coin flip. The AUC is 0.52, represented in Figure 4.13, with a true positive rate of 0.50 and a false positive rate of 0.45. The confusion matrix representing these rates is Figure 4.14. The model was abandoned in favor of the Logistic Regression model due to the lack of utility.

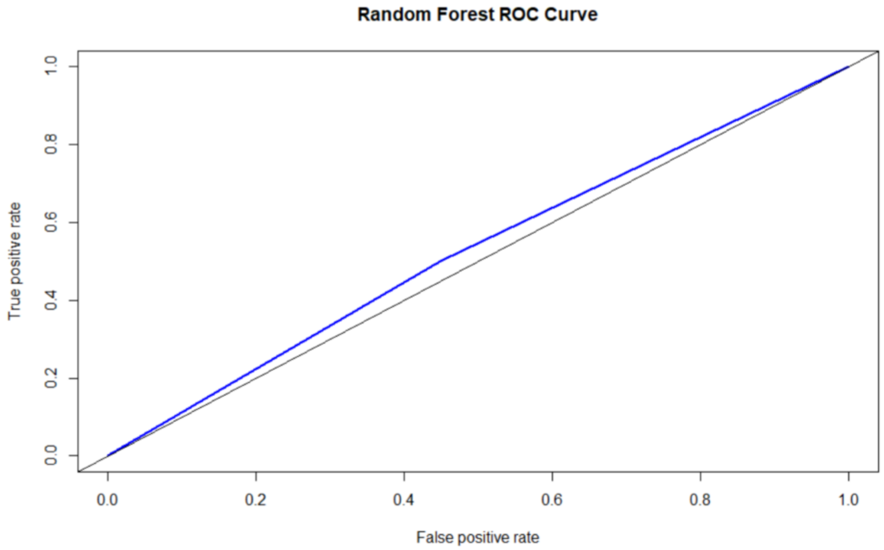


Figure 4-13. ROC curve produced from our random forest model

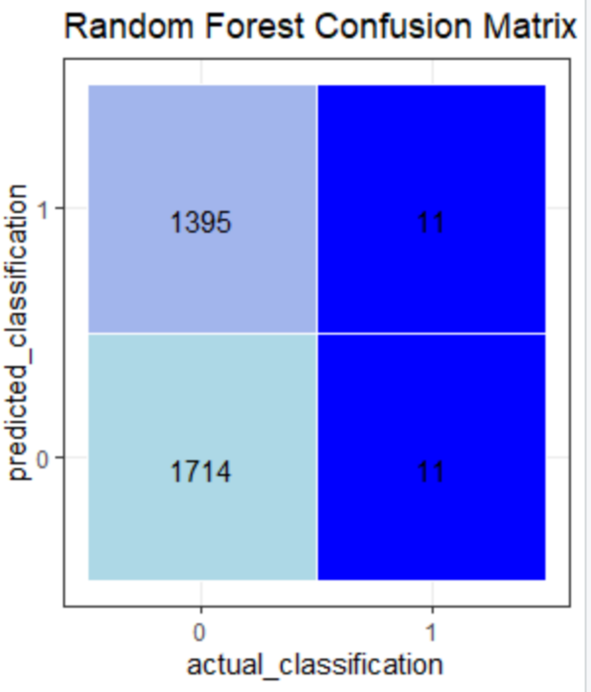


Figure 4-14. Confusion matrix from our random forest model

5 Findings

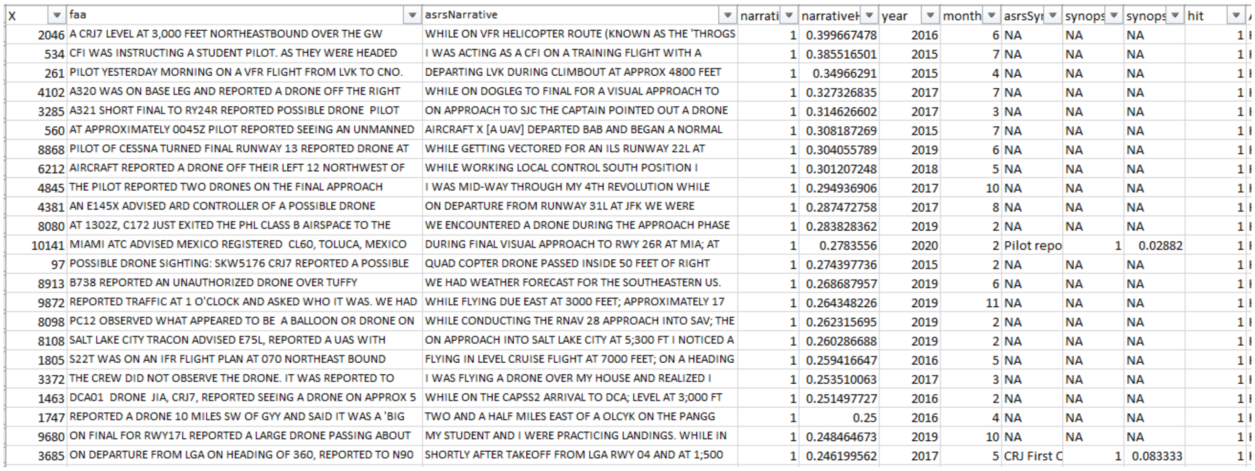
5.1 Data extraction for summary statistics and visualization:

The team was able to successfully create programming scripts that automate the workflow to turn the raw UAS incident reports into a formatted data set. The scripts extract data from the reports and append external data that is a crucial step for historical analysis and reporting. The FAA having access to this data will enable fast and reliable analysis of new reports as they become available.

The number of UAS incident reports has slowly declined since their peak in 2017, declining more during the start and duration of the Covid-19 pandemic lockdown. This would seem antithetical to the data that UAS usage is rapidly increasing year after year, however, it could mean that the certification requirements are creating an environment where UAS pilots are flying their UAS in a safe and regulated manner, while the rapid decrease in numbers during the end of 2019 and duration of 2020 could primarily be due to the Covid-19 lockdown and major inactivity during this period. We are not able to verify this claim, but it would seem plausible based off increased regulation on the pilots. The reports also have consistent high numbers in summer months, peaking in June for most years and have seen a recent increase in number of reports in 2021 with June 2021 having the highest number of reports to date.

5.2 Cross-Validation to label actual sightings:

Using the concept of cosine similarity and text2vec in R, the initial cross-validation between the UAS sightings dataset and the ASRS database dataset results in 737 hits, with the highest value for a measure of similarity being 0.433 and the majority of the hits falling below a 0.2 value.



X	faa	asrsNarrative	narrati	narrativeH	year	month	asrsSyı	synops	synops	hit
2046	A CRJ7 LEVEL AT 3,000 FEET NORTHEASTBOUND OVER THE GW	WHILE ON VFR HELICOPTER ROUTE (KNOWN AS THE 'THROGS	1	0.399667478	2016	6	NA	NA	NA	1
534	CFI WAS INSTRUCTING A STUDENT PILOT. AS THEY WERE HEADED	I WAS ACTING AS A CFI ON A TRAINING FLIGHT WITH A	1	0.385516501	2015	7	NA	NA	NA	1
261	PILOT YESTERDAY MORNING ON A VFR FLIGHT FROM LVK TO CNO.	DEPARTING LVK DURING CLIMBOUT AT APPROX 4800 FEET	1	0.34966291	2015	4	NA	NA	NA	1
4102	A320 WAS ON BASE LEG AND REPORTED A DRONE OFF THE RIGHT	WHILE ON DOGLEG TO FINAL FOR A VISUAL APPROACH TO	1	0.327326835	2017	7	NA	NA	NA	1
3285	A321 SHORT FINAL TO RY24R REPORTED POSSIBLE DRONE PILOT	ON APPROACH TO SJC THE CAPTAIN POINTED OUT A DRONE	1	0.314626602	2017	3	NA	NA	NA	1
560	AT APPROXIMATELY 0045Z PILOT REPORTED SEEING AN UNMANNED	AIRCRAFT X [A UAV] DEPARTED BAB AND BEGAN A NORMAL	1	0.308187269	2015	7	NA	NA	NA	1
8868	PILOT OF CESSNA TURNED FINAL RUNWAY 13 REPORTED DRONE AT	WHILE GETTING VECTORED FOR AN ILS RUNWAY 22L AT	1	0.304055789	2019	6	NA	NA	NA	1
6212	AIRCRAFT REPORTED A DRONE OFF THEIR LEFT 12 NORTHWEST OF	WHILE WORKING LOCAL CONTROL SOUTH POSITION I	1	0.301207248	2018	5	NA	NA	NA	1
4845	THE PILOT REPORTED TWO DRONES ON THE FINAL APPROACH	I WAS MID-WAY THROUGH MY 4TH REVOLUTION WHILE	1	0.294936906	2017	10	NA	NA	NA	1
4381	AN E145X ADVISED ARD CONTROLLER OF A POSSIBLE DRONE	ON DEPARTURE FROM RUNWAY 31L AT JFK WE WERE	1	0.287472758	2017	8	NA	NA	NA	1
8080	AT 1302Z, C172 JUST EXITED THE PHL CLASS B AIRSPACE TO THE	WE ENCOUNTERED A DRONE DURING THE APPROACH PHASE	1	0.283828362	2019	2	NA	NA	NA	1
10141	MIAMI ATC ADVISED MEXICO REGISTERED CL60, TOLUCA, MEXICO	DURING FINAL VISUAL APPROACH TO RWY 26R AT MIA; AT	1	0.2783556	2020	2	Pilot repo	1	0.02882	1
97	POSSIBLE DRONE SIGHTING: SKW5176 CRJ7 REPORTED A POSSIBLE	QUAD COPTER DRONE PASSED INSIDE 50 FEET OF RIGHT	1	0.274397736	2015	2	NA	NA	NA	1
8913	B738 REPORTED AN UNAUTHORIZED DRONE OVER TUFFY	WE HAD WEATHER FORECAST FOR THE SOUTHEASTERN US.	1	0.268687957	2019	6	NA	NA	NA	1
9872	REPORTED TRAFFIC AT 1 O'CLOCK AND ASKED WHO IT WAS. WE HAD	WHILE FLYING DUE EAST AT 3000 FEET; APPROXIMATELY 17	1	0.264348226	2019	11	NA	NA	NA	1
8098	PC12 OBSERVED WHAT APPEARED TO BE A BALLOON OR DRONE ON	WHILE CONDUCTING THE RNAV 28 APPROACH INTO SAV; THE	1	0.262315695	2019	2	NA	NA	NA	1
8108	SALT LAKE CITY TRACON ADVISED E75L, REPORTED A UAS WITH	ON APPROACH INTO SALT LAKE CITY AT 5;300 FT I NOTICED A	1	0.260286688	2019	2	NA	NA	NA	1
1805	S22T WAS ON AN IFR FLIGHT PLAN AT 070 NORTHEAST BOUND	FLYING IN LEVEL CRUISE FLIGHT AT 7000 FEET; ON A HEADING	1	0.259416647	2016	5	NA	NA	NA	1
3372	THE CREW DID NOT OBSERVE THE DRONE. IT WAS REPORTED TO	I WAS FLYING A DRONE OVER MY HOUSE AND REALIZED I	1	0.253510063	2017	3	NA	NA	NA	1
1463	DCA01 DRONE JIA, CRJ7, REPORTED SEEING A DRONE ON APPROX 5	WHILE ON THE CAPSS2 ARRIVAL TO DCA; LEVEL AT 3;000 FT	1	0.251497727	2016	2	NA	NA	NA	1
1747	REPORTED A DRONE 10 MILES SW OF GYY AND SAID IT WAS A 'BIG	TWO AND A HALF MILES EAST OF A OLCYK ON THE PANGG	1	0.25	2016	4	NA	NA	NA	1
9680	ON FINAL FOR RWY17L REPORTED A LARGE DRONE PASSING ABOUT	MY STUDENT AND I WERE PRACTICING LANDINGS. WHILE IN	1	0.248464673	2019	10	NA	NA	NA	1
3685	ON DEPARTURE FROM LGA ON HEADING OF 360, REPORTED TO N90	SHORTLY AFTER TAKEOFF FROM LGA RWY 04 AND AT 1;500	1	0.246199562	2017	5	CRJ First C	1	0.083333	1

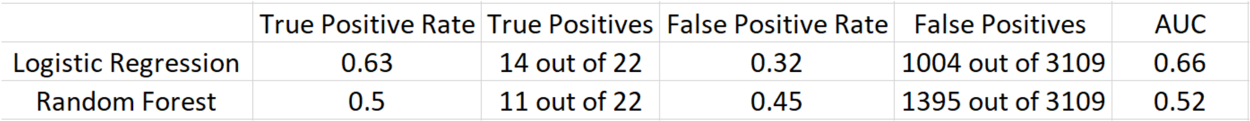
Figure 5-1. Example of our similarity testing output. This was used for cross validation.

The team went through the cross-validation hits manually to determine a substantial threshold for a similarity value that would be used to determine which similarity matches we kept as true UAS sightings. The 737 similarity hits were randomly sampled, and the team manually reviewed the two matched reports. The team individually found their thresholds and set the model threshold to average the individual reviews. 0.25 was determined as the cutoff value we were confident in for returning mostly accurate labels. Based on the new cutoff value, the number of similarity hits between the FAA UAS sightings and the NASA ASRS dataset was reduced from 737 to 85 records.

We were looking for more cross-validated records but could not reasonably include any below our cutoff. We proceeded with the 85 records we labeled as actuals and created a predictive modeling workflow trained on the cross-validated results. More sophisticated NLP modeling could potentially result in a higher match rate for validating reports.

5.3 Predictive Modeling:

The expectation for results of the predictive modeling performance was not substantial considering that there was such a rare outcome variable, 85 reports out of over 13,000. The logistic regression and random forest models' results confirmed the expectation. While they were both subpar results, the logistic regression outperformed the random forest model by a decent amount, as seen in Figure 5.2. With that being said, the logistic regression does not perform at a level that could be considered reliable either. Both models are about as useful as a coin flip, given that an AUC score of 0.50 is the equivalent of using a coin flip to label each record.



	True Positive Rate	True Positives	False Positive Rate	False Positives	AUC
Logistic Regression	0.63	14 out of 22	0.32	1004 out of 3109	0.66
Random Forest	0.5	11 out of 22	0.45	1395 out of 3109	0.52

Figure 5-2. comparison of results from the logistic regression model and random forest model. The team aimed to maximize True Positive rate while minimizing False Positive rates.

6 Summary

The most impactful output created from our project is the raw text processing pipeline. The team was able to effectively clean and extract the vital information from the FAA UAS sightings reports and merge it with external data sources. The workflow for this text processing has been developed for new reports as they are made available to the FAA. This workflow for extracting and standardizing data will aid the FAA in having a streamlined process to take raw reports and turn them into a data source which allows them to analyze the UAS incidents through visualizations and summary statistics. This analysis will allow for identifying the location of high UAS activity and a better understanding of the types of aircraft involved.

The team was also able to create a model to cross-validate the reports in the UAS sightings dataset against similar reports from the NASA ASRS dataset. With limited records in the ASRS dataset, the team successfully validated only 85 reports. Also, while validating 85 reports is relatively small, it does show improvement over the no-validated reports that the team started with. It will be crucial to validate more reports moving forward if a reliable predictive model needs to be deployed.

With the cross-validated results, the team trained and tested a predictive model. While the predictive results were subpar, the entire data pipeline that was created lays a solid foundation for future work. The entire data pipeline also shows that the UAS incident reports have valuable data contained within them, and there is a path to discovering even more insights through NLP modeling. This foundation can be built on to improve results using more sophisticated NLP methods and predictive modeling techniques. Newer models can be compared with the baseline models created in this project to determine their utility once they are developed.

7 Future Work

The team has recognized multiple paths that could improve upon the foundation we have laid. The most impactful and necessary work would be to label the historical reports as actual sightings or noise accurately. A more balanced dataset regarding actual reports versus noise reports will enable more robust predictive models. We could only cross-validate 85 reports out of over 13,000 as actual sightings. This created a weak predictive model, and additional confirmed reports should lead to a more potent performing model.

If actual reports cannot be confirmed, the following preferred option would be to explore more sophisticated methods for matching text similarities and incorporating the techniques with the ASRS dataset. In addition to using the improved models on the ASRS dataset, additional UAS sighting reports can be used in the cross-validation process to find more validated reports. Several agencies have UAS sightings data; NYPD and the City of San Francisco are recommended to explore since the data is known to exist, and the cities have high report activity in the FAA UAS sightings dataset.

If there is no way to increase the number of accurately labeled reports, synthetic data generation techniques can create more actual report samples that can then be used to train predictive models. Our cross-validation produced 85 reports that appeared to be actual reports likely. They can be used as guides for creating synthetic reports to balance the data set.

More sophisticated techniques than Topic Modeling are available for NLP pipelines that feed into a predictive model. We chose to start with Topic Modeling to see if it could provide satisfactory results. We were satisfied with the topics the LDA model uncovered, but we could not truly test the effectiveness of using topic probabilities as parameters in a predictive model due to the small distribution of validated reports. The sophisticated NLP techniques would likely outperform the topic models, but we recommend exploring both new techniques on the existing data and predictive retraining models on a more balanced dataset.

Regarding predictive modeling, Neural Networks are an extremely powerful tool in the NLP toolkit. We would expect that if a Neural Network was substituted into our workflow, it could improve predictive performance.

8 Appendix A: Code References

GitHub Link - https://github.com/jrainey4-GMU/GMU-DAEN-690-Team-Noise

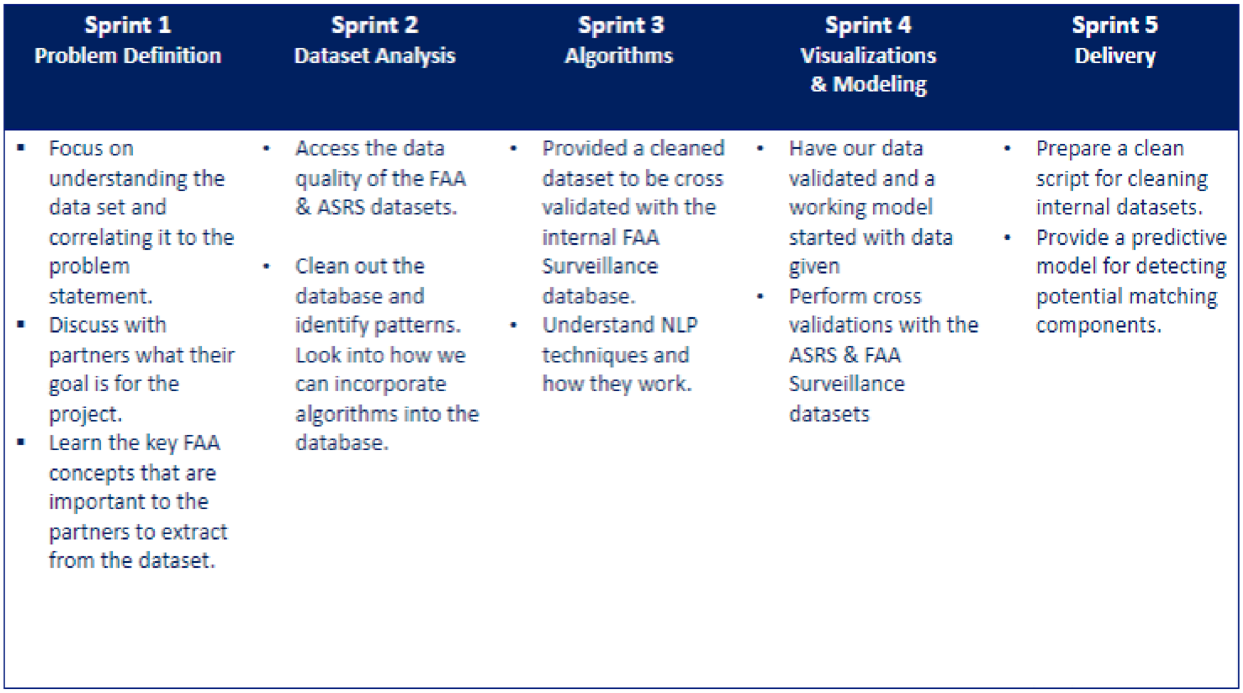
9 Appendix B: Risk Section

Risk Name	Description	Probability	Impact	Mitigation
ASRS Database observations are limited	Compares to the FAA UAS sightings dataset, the NASA ASRS database has significantly fewer records than anticipated	Medium	High	Work with FAA partners on other plans, possible internal other databases and clarification on requirements on cleaning data for efficient cross validation against limited records.
Understanding FAA notation and Aircraft Information	The reports include lots of terminology designed and used by pilots	High	Medium	Work as a team to understand terminology and work with FAA partners for better domain knowledge
Working with the 'Narrative'/'Summary' section of datasets	Most of the core information we are working with as a team is big chunks of text with most of the information needed.	High	High	We will need to parse out key information from the chunks of text and patterns, observe what we could standardize as key elements to reports
Using agile in a team setting and implementing user stories	The group would benefit more from working in epics	Medium	Low	We will organize our swim lanes into user stories and break down epics into story features we want to dive into
Redirecting of data cleaning and parsing techniques to R and Python	We will need to look at developing a flexible script in R or Python for data cleaning.	Medium	Medium	We will work on how to leverage current work on databrew and move to R or Python without losing progress
Cost of Using AWS Databrew	Using databrew with larger datasets and more recipes increased in price exponentially quickly	Medium	Medium	Redirect current progress to other languages based on discussion with FAA sponsors.
Anomalies in Dataset	This is what we need to explore in our datasets and cleaning	Medium	High	Try to cross reference with traffic control data since ASRS is limited

	besides the key indicators			
Having too much data to look into	There are 13000 records of inconsistent textual reports to look into	High	Medium	We will focus on smaller cities where applicable
Needing more valid data for analysis	Look for other data sources to cross validate	Medium	High	Identifying possible local police reports to look for any extra information on reports
NLP Techniques for textual analysis	Data cleaning and analysis process is significant to our progress and we need to find the best way	High	Medium	The team will do extensive research into past research projects as well as the methods to identify other understand the techniques needed for NLP analysis

10 Appendix C: Agile Development

- [Sprint 1] Discuss with partners what their goal is for the project.
- Learn the key FAA concepts that are important to the partners to extract from the dataset.
- [Sprint 2 & 4] Access the data quality of the FAA & ASRS datasets.
- [Sprint 3] Provided a cleaned dataset to be cross validated with the internal FAA Surveillance database.
- [Sprint 4] Perform cross validations with the ASRS dataset.



Sprint 1 Problem Definition	Sprint 2 Dataset Analysis	Sprint 3 Algorithms	Sprint 4 Visualizations & Modeling	Sprint 5 Delivery
▪ Focus on understanding the data set and correlating it to the problem statement. ▪ Discuss with partners what their goal is for the project. ▪ Learn the key FAA concepts that are important to the partners to extract from the dataset.	• Access the data quality of the FAA & ASRS datasets. • Clean out the database and identify patterns. Look into how we can incorporate algorithms into the database.	• Provided a cleaned dataset to be cross validated with the internal FAA Surveillance database. • Understand NLP techniques and how they work.	• Have our data validated and a working model started with data given • Perform cross validations with the ASRS & FAA Surveillance datasets	• Prepare a clean script for cleaning internal datasets. • Provide a predictive model for detecting potential matching components.

11 References

[1] "A Brief History of the FAA | Federal Aviation Administration." https://www.faa.gov/about/history/brief_history (accessed Nov. 02, 2021).

[2] "Unmanned Aircraft Systems (UAS)." https://www.faa.gov/uas/ (accessed Nov. 02, 2021).

[3] "FAA Aerospace Forecast Fiscal Years 2021–2041," p. 28.

[4] "UAS by the Numbers." https://www.faa.gov/uas/resources/by_the_numbers/ (accessed Nov. 02, 2021).

[5] I. Intelligence, "Drone market outlook in 2021: industry growth trends, market stats and forecast," *Business Insider*. https://www.businessinsider.com/drone-industry-analysis-market-trends-growth-forecasts (accessed Nov. 02, 2021).

[6] "Global Consumer Drones Market Report 2021: Hobbyist & Gaming; Aerial Photography; Others - Forecast to 2030 - ResearchAndMarkets.com," Jun. 21, 2021. https://www.businesswire.com/news/home/20210621005402/en/Global-Consumer-Drones-Market-Report-2021-Hobbyist-Gaming-Aerial-Photography-Others---Forecast-to-2030---ResearchAndMarkets.com (accessed Nov. 05, 2021).

[7] M. D. Shear and M. S. Schmidt, "White House Drone Crash Described as a U.S. Worker's Drunken Lark," *The New York Times*, Jan. 27, 2015. Accessed: Nov. 05, 2021. [Online]. Available: https://www.nytimes.com/2015/01/28/us/white-house-drone.html

[8] "Drone strikes commercial aircraft in Quebec: Garneau | CTV News." https://www.ctvnews.ca/canada/drone-strikes-commercial-aircraft-in-quebec-garneau-1.3633035 (accessed Nov. 05, 2021).

[9] "172 Substantially Damaged By Police Drone - AVweb." https://www.avweb.com/aviation-news/172-substantially-damaged-by-police-drone/ (accessed Nov. 05, 2021).

[10] "Risk in the Sky? : University of Dayton, Ohio." https://udayton.edu/udri/news/18-09-13-risk-in-the-sky.php (accessed Nov. 05, 2021).

[11] "Drones crashing into airplanes: Small quadcopters can seriously damage." https://www.usatoday.com/story/travel/nation-now/2018/10/17/drones-crashing-into-airplanes-quadcopters-damage-video/1657112002/ (accessed Nov. 05, 2021).

[12] R. Radar, "5 Biggest Drone Incidents at Airports in 2020." https://www.robinradar.com/press/blog/5-biggest-drone-incidents-at-airports-in-2020 (accessed Nov. 05, 2021).

[13] “Small Unmanned Aircraft Systems (UAS) Regulations (Part 107) | Federal Aviation Administration.” https://www.faa.gov/newsroom/small-unmanned-aircraft-systems-uas-regulations-part-107?newsId=22615 (accessed Nov. 11, 2021).

[14] “UAS Remote Identification Overview.” https://www.faa.gov/uas/getting_started/remote_id/ (accessed Nov. 05, 2021).

[15] “UAS Sightings Report.” https://www.faa.gov/uas/resources/public_records/uas_sightings_report/ (accessed Nov. 11, 2021).

[16] U. S. G. A. Office, “Small Unmanned Aircraft Systems: FAA Should Improve Its Management of Safety Risks.” https://www.gao.gov/products/gao-18-110 (accessed Nov. 29, 2021).

[17] “Simple, Consistent Wrappers for Common String Operations.” https://stringr.tidyverse.org/ (accessed Nov. 11, 2021).

[18] “10 minutes to pandas — pandas 1.3.4 documentation.” https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html (accessed Nov. 11, 2021).

[19] “Cosine Similarity - Understanding the math and how it works? (with python),” *Machine Learning Plus*, Oct. 22, 2018. https://www.machinelearningplus.com/nlp/cosine-similarity/ (accessed Nov. 29, 2021).

[20] R. Khandelwal, “Word Embeddings for NLP,” *Medium*, Dec. 28, 2019. https://towardsdatascience.com/word-embeddings-for-nlp-5b72991e01d4 (accessed Nov. 11, 2021).

[21] S. Li, “Natural Language Processing for Fuzzy String Matching with Python,” *Medium*, Dec. 06, 2018. https://towardsdatascience.com/natural-language-processing-for-fuzzy-string-matching-with-python-6632b7824c49 (accessed Nov. 11, 2021).

[22] S. Li, “Topic Modeling and Latent Dirichlet Allocation (LDA) in Python,” *Medium*, Jun. 01, 2018. https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24 (accessed Nov. 11, 2021).

[23] M. Raza, “Importance of NLP in Data Science,” *TheLeanProgrammer*, Jun. 12, 2021. https://medium.com/theleanprogrammer/importance-of-nlp-in-data-science-3e2fd2424b2d (accessed Nov. 11, 2021).

[24] “Text Similarities : Estimate the degree of similarity between two texts | by Adrien Sieg | Medium.” https://medium.com/@adriensieg/text-similarities-da019229c894 (accessed Nov. 29, 2021).

[25] “The dot product - Math Insight.” https://mathinsight.org/dot_product (accessed Nov. 29, 2021).

[26] “Latent Dirichlet Allocation (LDA) | LDA using Gensim,” *Analytics Vidhya*, Jun. 27, 2021. https://www.analyticsvidhya.com/blog/2021/06/topic-modeling-and-latent-dirichlet-allocationlda-using-gensim-and-sklearn-part-1/ (accessed Nov. 11, 2021).

[27] S. Chakravarthy, “Tokenization for Natural Language Processing,” *Medium*, Jul. 10, 2020. https://towardsdatascience.com/tokenization-for-natural-language-processing-a179a891bad4 (accessed Nov. 11, 2021).

[28] “Understanding Word N-grams and N-gram Probability in Natural Language Processing | by Sunny Srinidhi | Towards Data Science.” https://towardsdatascience.com/understanding-word-n-grams-and-n-gram-probability-in-natural-language-processing-9d9eef0fa058 (accessed Nov. 11, 2021).

[29] “Topic Modeling and Latent Dirichlet Allocation (LDA) using Gensim,” *Analytics Vidhya*, Jun. 28, 2021. https://www.analyticsvidhya.com/blog/2021/06/part-2-topic-modeling-and-latent-dirichlet-allocation-lda-using-gensim-and-sklearn/ (accessed Nov. 11, 2021).