Proposal for a Modernized Flight Risk Assessment Tool for General Aviation Pre-Flight Planning

Hunter Walden
Kathleen Hill
Chi Quinn
Erick Torres
Dr. Isaac Gang, Project Advisor

George Mason University

Executive Summary

This project sought to research machine learning methods that could be used to create a modernized Flight Risk Assessment Tool (FRAT). The tool could be made available to General Aviation (GA) pilots, which could increase pre-flight hazard identification and subsequently decrease accident risk and improve safety for the GA community. This would be an improvement on existing FRATs through the incorporation of predictive modeling, developed through research on historical accident data, utilizing algorithms that predict a pre-defined risk category for a given flight profile. Data was sourced from the National Transportation Safety Board (NTSB) to include 40 years of aircraft accident and incident information. The models evaluated during this project were specifically designed for pre-flight risk assessment and, as such, utilized only information available during pre-flight analysis. This data is similar to topics covered in legacy FRAT tools and the "Pilot, Aircraft, enVironment, and External pressures" (PAVE) checklist for hazard identification. The models used in this project did not consider pilot errors, lapse of judgment, air traffic controller errors, other in-flight human errors, or mechanical component failure anomalies as those data points are not known prior to takeoff. The models explored showed the ability to predict risk categories for flight profiles based on pattern recognition through the classification of historical accident/incident records and hazard reports and showed the potential to be more robust than legacy FRATs with several models achieving moderately high accuracy in predicting risk categories. This method of utilizing a model to predict the risk category for the flight reduces pilot-induced bias, decreasing the likelihood that a pilot may not properly identify hazards, or be overconfident in their abilities. This modern application of risk analysis could be superior to legacy tools as it is capable of: 1) continuously updating risk profiles based on additional data provided by each flight record, 2) identifying a risk that the pilot may otherwise not identify, and 3) providing a customized a risk score based on a specific flight profile.

Table of Contents

Problem Statement and Background

Prior to every flight, it is important that a pilot identify hazards, assess risks, and mitigate risks associated with a given flight profile. They should consider many factors before determining whether to fly, such as current and forecasted weather, personal expertise, familiarity with departure and destination airport, currency and proficiency, hours in aircraft model, etc. The Federal Aviation Administration (FAA) General Aviation Joint Steering Committee published a fact sheet describing the use of Flight Risk Assessment Tools, which enables proactive hazard identification prior to flights. A pilot with the FAA Safety Team's (FAAST) FRAT can produce a risk score to make the decision on whether to fly and which hazards to mitigate (General Aviation Joint Steering Committee, n.d.). This project sought to improve upon this tool by researching and testing predictive models that could compare planned flight profiles to historical records of accidents, incidents, and flight hazards to identify potential hazards a pilot might otherwise not identify or even be aware of. This project utilized advanced machine learning algorithms to examine historical flight accidents and incident data to determine risk categories for flights. For each flight, risk categories were predicted based on outcome severity.

The FAA described the process for conducting risk assessments in its Risk Management Handbook. Reviewing the PAVE checklist allows a pilot to deliberately assess areas of concern during preflight planning. This acronym describes the following: pilot, aircraft, environment, and external pressures. The "pilot" portion of this checklist generally includes a self-assessment including a subjective evaluation of a pilot's "experience, currency, physical, and emotional condition" (*Risk Management Handbook*, 2009, p. 3-3). The models developed in this project include a quantitative evaluation of the pilot's experience based on available flight hour metrics. Future work might include physical evaluation via smart wearable technology integrated with pre-flight planning software, or the utilization of surveys to gather data. This project incorporated the "aircraft" and "environment" areas of risk assessment with the available historical accident, airport, and weather data. While there are other application-based FRAT tools available, there are not currently any known tools that utilize predictive models to predict risk categories for flights based on historical accident and incident data.

Expanding FRAT accessibility and effectiveness was worth pursuing due to the heavy cost of lapses in pilot judgment and poor preflight planning. According to Robert Wright's article in *Aviation Safety Magazine*, "poor risk management was a root cause of nearly 50 percent of the fatal business aircraft accidents" of those he analyzed as a part of his research (2018, para. 13). According to Boyd's 2017 literature review of safety in GA, civilian aviation operations excluding paid passenger transport, accounts for 94% of civilian aviation fatalities (p. 657). Boyd goes on to discuss the financial cost of these accidents, describing a loss of $1.6-4.6 billion in expenses "associated with injury (inclusive of hospital costs) and/or loss of life, accident investigations, loss of pay with a fatal accident, and loss of the aircraft" (2017, para. 2). This clearly illustrates the need for improved, accessible, and effective preflight risk management tools within GA.

Many studies have examined what might lead to an increase in risk for a pilot. Bazargan & Guzhva (2011) compare the effect of gender, age, and experience on a pilot's likelihood to be involved in an accident in a study of over 40,000 accidents from 1982-2002 (see Figure 2). They found that age and gender do not appear to affect pilot error up to the age of 60. They also suggest that more experienced pilots are less likely to exhibit pilot errors leading to accidents. Fultz & Ashley discuss the degree to which weather contributes to accident fatality in GA accidents. They state that 60% of fatal accidents occurred while flying in instrument meteorological conditions

(IMC), most frequently "between October and April, on weekends, in early morning and evening periods, and along the West Coast, Colorado Rockies, Appalachian Mountains, and the Northeast" (2016, p. 291). Boyd (2017) describes IMC as increasing the likelihood of accidents by requiring a pilot to control the aircraft by reference to instruments in the absence of outside visual cues. Though only 9% of general aviation mishaps occur during these conditions, they account for a significant portion of fatalities. Additional factors which Boyd discusses as increasing the likelihood of an accident in GA are geographic location (see Figure 3), planned flight distance, and time of day. Those flights taking place in mountainous areas, with a longer flight distance, or conducted at night carry a significantly higher likelihood of fatal outcomes (para. 7-8).

Solution Definition

The solution for this project was defined as a modernized preflight risk assessment tool that is specifically for Part 91 pilots and is meant to increase accessibility, increase hazard identification, and increase safety records in the GA community. The scope of this project was only to conduct the foundational research required to validate the concept, upon which further testing, validation, and applications can be built in the future. The modernized preflight risk assessment tool should utilize advanced machine learning and artificial intelligence solutions, as detailed in this report, to predict risk classification scores utilizing data entered by a pilot regarding their proposed flight profile. This project used historical NTSB accident and incident data that was hand-labeled based on event outcome severity to train different machine learning models. Since the tool is for preflight risk assessment, only data available to a pilot during preflight planning was used to train predictive classification models.

This modernized FRAT solves many issues with traditional tools and worksheets. Limitations of FRATs currently available to pilots include poor accessibility, incomplete or subjective survey questions, the excessive time required to complete worksheets and lack of regulation requiring usage. The FAAST FRAT is neither user-friendly nor easy to find, download, or utilize. While the manual completion of a FRAT can be a way of ensuring a pilot seriously examines criteria relevant to pre-flight risk, this process is only helpful if the pilot takes time to deliberately conduct a flight risk assessment. Wright describes the shortcomings of traditional FRATs in his previously mentioned article, asserting that a) they do not consider the probability of a certain hazard occurring, b) they allow pilot bias and overconfidence to mark higher risk events as lower risk, and c) they tend to be incomplete in their evaluation criteria (2018). Wright goes on to discuss a shortcoming in the typically included threshold for go/no-go determination. He states that numerical risk assessment should consider outcome severity and probability of occurrence (2018).

This project proposed solutions to many of these shortfalls of traditional FRATs in the following ways. First, accessibility is improved if the risk prediction model is integrated into a preflight planning program or incorporated into an accessible web application. Second, the predictive model reduces potential bias from the pilot or tendencies to adjust input in manually completed FRATs to obtain lower risk scores (keeping in mind there is bias in predictive models). This is done through a comparison of the flight profile against similar historical flight profile data producing an objective risk score and identifying hazards that the pilot could then choose to mitigate. Third, it considers the severity of hazards. The probability of occurrence was not utilized or calculated in this project due to a lack of non-accident/incident-related flight data. Finally, this

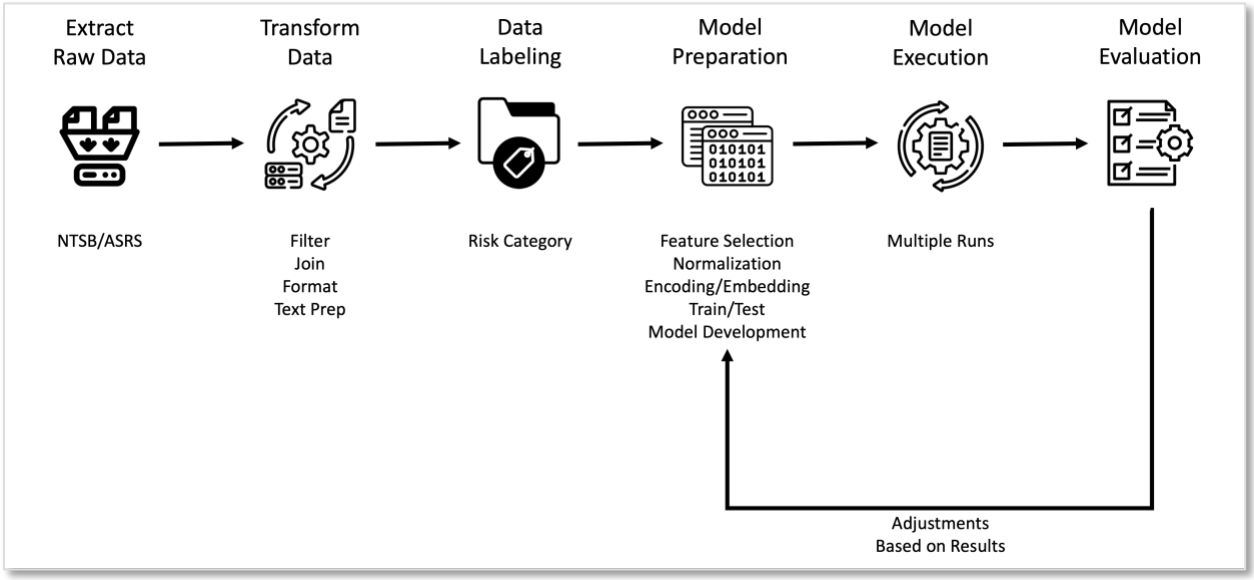
project provides the pilot an opportunity to review and mitigate hazards that they otherwise may have missed with incomplete tools or a rushed analysis.

Methodology

Figure 1 summarizes the methodology and workflow for the project. In general, a standard extract, transform, and load workflow was used to collect raw data, manipulate it to a useful form, and load it into a flat file for ingestion into predictive machine learning models. Several standard practices were utilized as part of the methodology. After data collection, data cleansing was conducted. Data cleansing included correcting erroneous data, standardizing text data, altering data types, and handling missing values, amongst others. Once the data was cleaned, records were labeled for classification purposes. Data distributions were analyzed and assessed for resampling. Multiple different models were run, including XGBoost, random forest, Bidirectional Encoder Representations from Transformers (BERT), and a feed-forward neural network (FFNN). Each model was run initially as a baseline, followed by multiple iterations and tests on manipulating data inputs, hyperparameters, and sample sizes. Once a suitable result was obtained, n-fold cross-validation testing was conducted to ensure models were not overfitting and could generalize well.

Figure 1

Workflow Diagram Utilized in FRAT Project



Data Extraction and Collection

The primary data source was the continuous set of incident and accident records compiled from NTSB data sets (National Transportation Safety Board, 2021). NTSB data consists of accidents and incidents that were investigated and reported on by the National Transportation Safety Board, including over 40 years of accident and incident records dating back to 1980. Over the course of the project, multiple other data sources were collected, cleaned, and evaluated for suitability and modeling.

The records included in the various data sets were initially examined to explore potential trends across events. When examining the NTSB data set, a few trends were noted. First, it

appeared there was an increase in event occurrences across the summer months (see Figure 2). Second, Figure 3 shows that most events occurred at elevations below 1000 feet. After examining the data as a density plot across the continental United States, high-density accident clusters were seen in multiple regions (See Figure 3). To truly evaluate these trends, it would be necessary to incorporate information from all GA flights to calculate rates by which to compare disparate geographic locations. This limitation is discussed in further detail in the Conclusions section.

Figure 2

Event Occurrence by Month (NTSB Data)

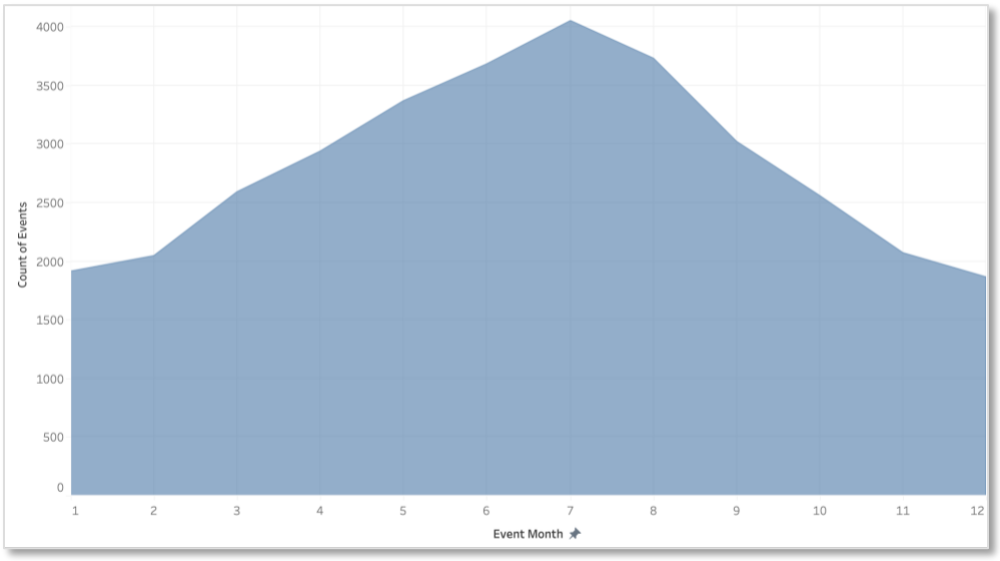


Figure 3

Event Occurrence by Airport Elevation (NTSB Data)

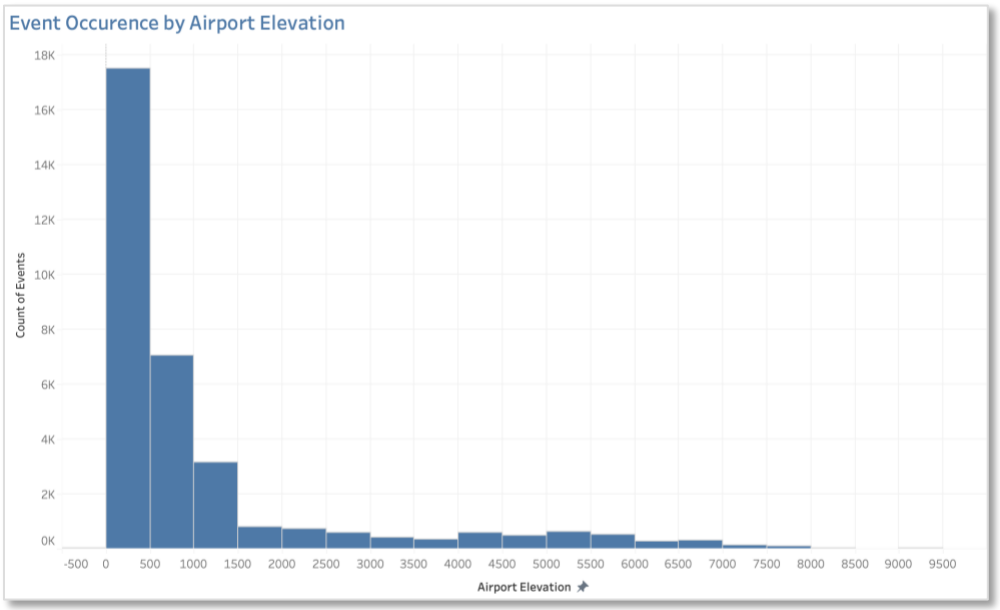
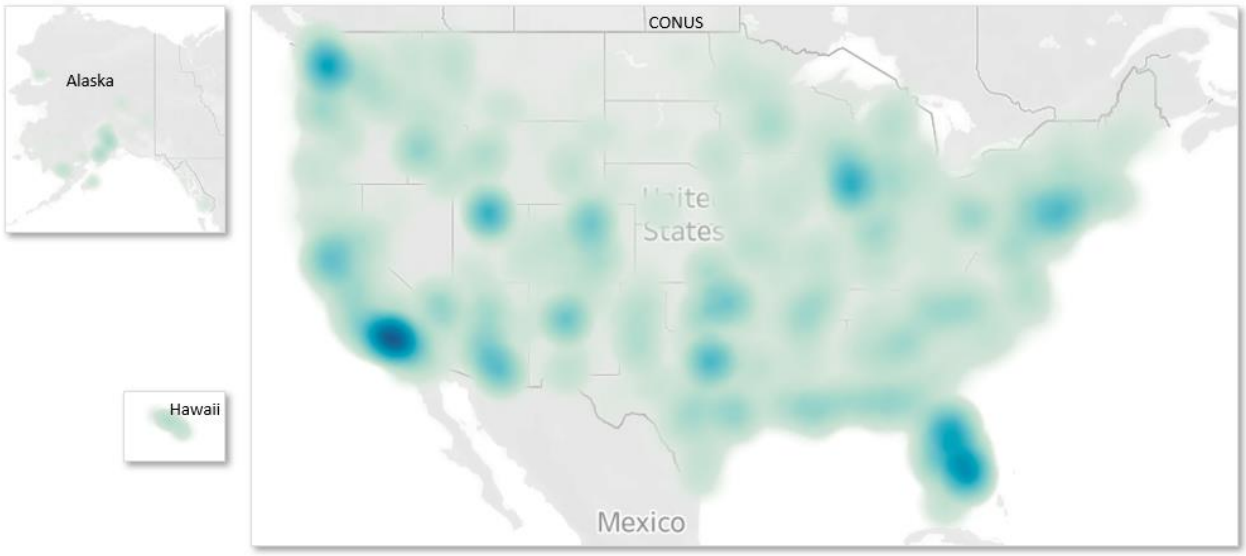


Figure 4

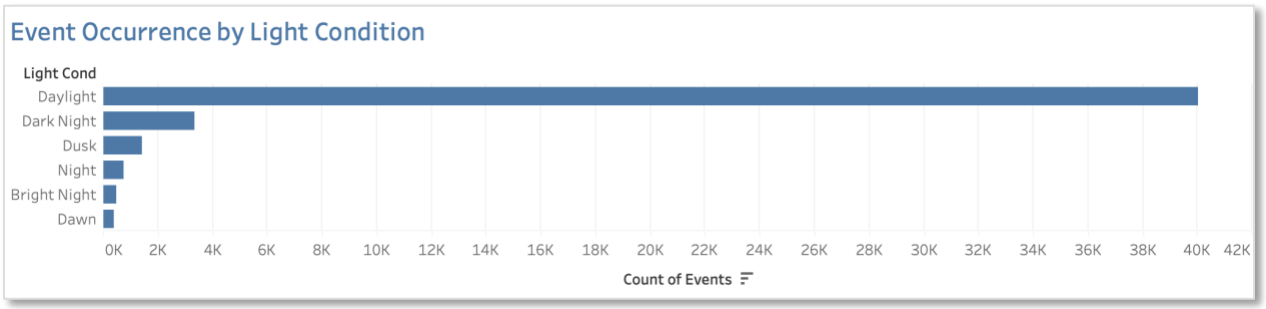
Heatmap of NTSB General Aviation Accidents 2008-2022



Note. This heat map illustrates the distribution of accident records in the NTSB data set. High-density areas include metropolitan areas, mountainous areas, and the eastern and western seaboard where aviation traffic is high. It is important to understand how the geographic area affects flight risk.

Figure 5

Event Occurrence by Light Conditions (NTSB Data)



Data Transformation

Standardizing and normalizing data for this project was a very complex task. Much of the information contained in the NTSB records was a heterogeneous mixture of text-based nominal and ordinal categorical data, and discrete and continuous numeric data. Basic cleaning tasks included standardizing naming conventions, handling missing values, filtering data, joining data, deduplicating event records, and ensuring consistency throughout the data set.

Filtering required for this project began with ensuring data only included GA records. The data originally contained records from all Federal Aviation Regulation (FAR) parts. As our research was focused on improving safety for GA, records from Parts 121, 125, and 129 were removed as these commercial air carrier flight profiles are vastly different from those of Part 91 and would not necessarily provide relevant data to the model.

Accident causal factors were analyzed, and the following factors were found to be present in the data:

Table 1
Causal Factors for Incidents in NTSB and ASRS Data sets

Causal Factor	Description of Events Included
Weather	Unintentional flight into IMC, lightning strikes, downdrafts, precipitation, icing, hail, strong or gusty winds, etc.
Environmental	Non-weather-related anomalies include controlled flight into terrain (CFIT), elevation, failure to avoid obstructions, inadequate runway, icy or wet runway, bird strikes, etc.
Pilot Related	Information about the pilot and their craft. This data includes flight time and aircraft information and includes wrong control input, loss of control, lack of situational awareness, human error, failure to follow assigned instructions, etc.
Aircraft	Component malfunction, engine failure, loss of power, power available exceeding power required

Accident records were not filtered based on causal factors. It is important to paint a picture of the environment in which any accident or incident occurs to identify patterns that lead to accidents, regardless of the causal factor. Primary and secondary causal factors tend to be identified as factors that occur at or around the time of the event. For example, an engine failure might be cited as the primary cause of a crash, and a secondary cause might be cited as an oil leak. What may not be captured explicitly is that the engine maintenance schedule was not followed properly, and the pilot did not conduct proper preflight records inspection causing the pilot not to notice that an oil seal exceeded its lifespan and was susceptible to failure. This example accident very well could have been mitigated through pre-flight risk assessment.

Once the data was cleaned and filtered, a risk category was assigned to each record, based on the result of each accident or incident. Each record contained an injury field and an aircraft damage field with categories of none, minor, severe, or fatal/destroyed. The risk labeling scheme is as follows:

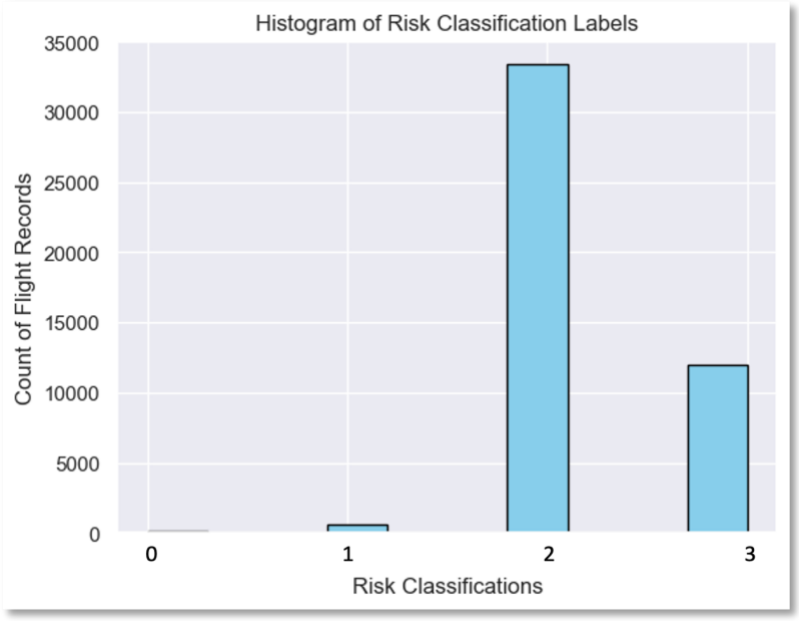
Table 2
Proposed Risk Categories

Risk Score	Consequence Level	Description
0	Low	Flight resulted in an incident or accident with no aircraft damage AND no minor injury.
1	Medium	Flight resulted in an incident or accident with either minor aircraft damage OR minor injury, AND no fatality.
2	High	Flight resulted in an incident or accident with either severe aircraft damage OR severe injury, AND no fatality.
3	Catastrophic	Flight resulted in an incident or accident with either fatality or destruction of the aircraft.

Note: When comparing aircraft damage and injury, the highest of the two dictates the classification. For example, a record with minor aircraft damage and severe injury would be classified as category 2, High. Numerical risk scores are only utilized for modeling purposes, and it is preferred that an end user see an output of "Low" or "High" rather than 0 or 2. The risk score of 0 does not imply a no-risk flight.

Figure 6

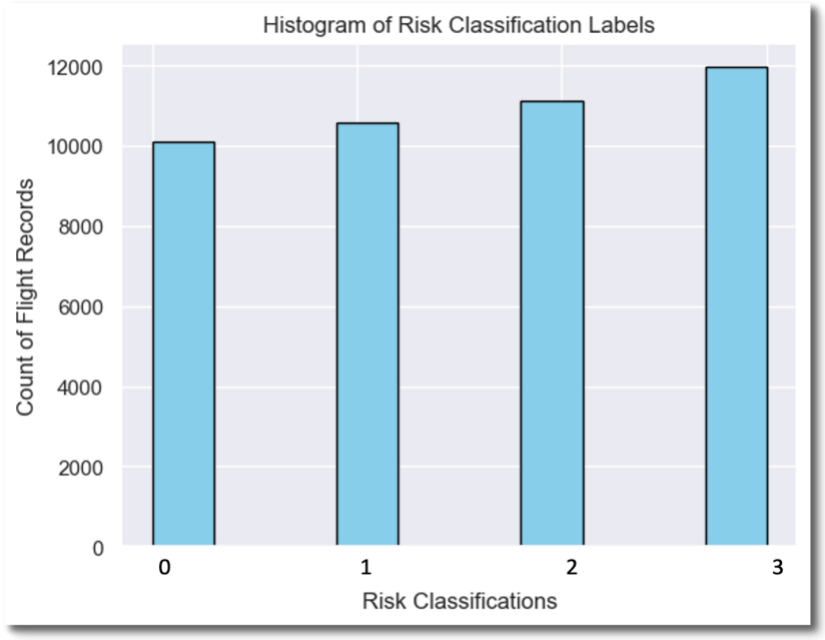
Histogram of Risk Classification Labels



The data set was heavily skewed toward those records which resulted in higher risk categories. This stands to reason, as the data was sourced from the NTSB data set including incident data as previously described. Flights conducted without incident or with only minimal negative outcome would either not be included or only included on a limited basis if the event triggered an NTSB investigation (see Figure 6). An imbalanced data set is not conducive to good machine learning results because the models would tend to overfit on a majority class. This problem is solved through resampling techniques to balance the data set. To rebalance the labeled data set, a combination of under and oversampling techniques was utilized. Those records labeled as risk values 2 (the majority class) were randomly undersampled to reduce their count, and those labeled as 0 or 1 (minority classes) were oversampled by random duplication to increase their count. While these techniques can introduce bias into the data, it is necessary to prevent overfitting across all categories and increase accuracy and result scores appropriately. Further discussion can be found in the conclusions section of this report. Figure 7 shows the distribution of labeled classes following the rebalancing of the labeled data set, resulting in records between 10,000 and 12,000 for all labels.

Figure 7

Histogram of Risk Classification Labels for Balanced Data set



Modeling

To predict risk categories using pre-flight data, machine learning algorithms were used to classify records in the data set. Multiple models were implemented and tested with the intent of determining which might predict risk categories most accurately. To start, features were selected for inclusion in the model. Once features were selected, data was manipulated into usable formats specific to each model's specifications. Only data available to pilots during pre-flight planning was included in the models to keep the model aligned with the intent of the project. Retained features are outlined Table 3.

Table 3

Feature selection for predictive modeling

Pilot Data	Aircraft Data	Environment Data	Weather Data
Total Flight Hours	Aircraft Make	Dept/Dest Airport	Temperature / Dewpoint
Actual Instrument Time	Aircraft Model	Airport Location	Wind Direction / Speed
Flight Hours in Make	Airframe Hours	Airport Elevation	Meteorological conditions
Flight Hours at Night	Certified Max Gross Weight	Departure Time	Ceilings
Rotary Wing Flight Hours	Airframe Inspection Timeline	Runway length	Visibility
Single Engine Flight Hours	Engine hours	Runway width	Weather Observation Time
Multi-Engine Flight Hours	No. Engines	Day of week	Light Conditions
Simulated Instrument Hours	Engine Manufacturer	Month	Wind Gust Speed

Note. These features contain information that is available and typically used during pre-flight planning.

XGBoost Model (Baseline)

XGBoost was selected as the baseline classification model for this project for the following reasons. First, no clear patterns presented themselves in the data in terms of correlation. Second,

the data was a mix of numeric data and categorical text data. Third, the data set had a significant number of missing values. Boosted decision tree models are well-equipped to handle the characteristics of the data set. Specifically, the Histogram-based Gradient Boosting Classification Tree from the SciKit Learn library was utilized (Pedregosa, 2011).

The process for training the model included final data preparation, running a baseline model on the imbalanced data set, then training subsequent models to test the effects of feature selection, data set balancing, hyperparameter tuning, and labeling schemes. Once a final configuration was selected, based on evaluation metrics such as confusion matrix, F1 score, precision, accuracy, and recall, a ten-fold cross-validation test was run to check for overfitting and generalization.

To establish a baseline, the initial model was trained on the original imbalanced data set. This model showed a modest ability to accurately predict risk categories for accidents and incidents producing an average F1 score of 0.76. With a baseline established, multiple follow-on tests were conducted. The process and results are enumerated below. The F1 scores represented in the "Results" are a weighted average of F1 scores for each class for easy comparison between tests. More detailed results are included in the appendix.

Table 4

XGBoost modeling results

Test Iteration	Configuration	F1 Score Results
Baseline	Imbalanced	0.76
1	Balanced	0.85
2	Category reduction, imbalanced	0.77
3	Category reduction, balanced	0.79
4	Normalized, balanced	0.84
5	Category reduction, normalized, balanced	0.79
6	10-fold cross-validation, balanced*	0.85

Note: * indicates the preferred model configuration

Referencing Table 4 above, test 1 involved training the model on the balanced data set. This showed an overall increase in performance and a tendency for the model to "miss high". In other words when the model was wrong, it tended to predict a higher risk category than the true category, which is a favorable characteristic. Test 2 involved category reduction to observe the effect that different categories had on the results. In this test, risk categories 0 and 1 were combined into a single risk category, in the original imbalanced data set. The results of this test showed a decrease in overall performance. Test 3 expanded on test 2 and maintained the category reduction, but this time on the balanced data set. Here, there was a slight increase in performance over test 1. This observation demonstrates that having more categories and a balanced data set allows the model to discriminate between higher risk (2 and 3) and lower risk (0 and 1) categories more accurately. Tests 4 and 5 used similar configurations as tests 2 and 3 respectively, with the exception that normalization was implemented in tests 4 and 5. There were no appreciable differences in performance with normalization.

Overall, test iteration 6 provided the best results. This test was conducted with four category labels, a balanced data set, unnormalized data, and default hyperparameter settings for the model. Observing the individual label F1 scores as well as the confusion matrix showed that overfitting occurred on labels one and two. This is expected due to the method by which the data

set was balanced through over-sampling. Although the low-risk labels were overfitted, there was a positive benefit because it allows the model to reduce error when predicting high-level risk events.

The 10-fold cross-validation results were very consistent through each iteration of the validation tests. This provided evidence that the model should be able to generalize new data with similar accuracy. Another important factor is how the model makes errors. We observed that when the model incorrectly classifies a record, it tends to predict a higher risk category than the true classification, rather than a lower risk category. This is an important characteristic in risk prediction because it means the model is making more conservative estimations. If the model is wrong, it is better to err on the high-risk side of the scale.

Random Forest Model

Random Forest Classification is another ensemble model that was utilized due to its flexibility in handling missing values, large data sets, and high dimensionality (Johnston, & Mathur, I., 2019). Random forest uses bootstrap aggregating or bagging to avoid overfitting and decreasing variance (Bruce et al., 2020). It produces a collection of decision trees and predicts the output of the majority vote.

The process for training the model echoed the method of training and testing the XGBoost model. This included data preparation, producing a baseline model, and observing multiple models using different methods including rebalancing the data set and hyperparameter tuning. The best result was selected based on the F1 score and the confusion matrix. The model was initialized using the original, imbalanced data set using the parameter default values of the random forest classifier in the SciKit Learn library (Pedregosa et al., 2011). The baseline random forest model returned a fair score of 0.75. The data set was then balanced using similar over and under-sampling methods to the previous model. The F1 score for the balanced data set produced a better F1 score of 0.85.

Next, hyperparameter tuning was performed to evaluate the best values in the model to improve the performance. This procedure included performing a 10-fold cross-validation to avoid overfitting and ensure generalization (Bruce et al., 2020). Table 5 below displays the results of the three different methods.

Table 5

Random Forest Modeling Results

Test Iteration	Configuration	F1 Score Results
Baseline	Imbalanced	0.75
1	Balanced	0.85
2	10-fold cross validation, balanced*	0.86

Note: * indicates the preferred model configuration

The 10-fold cross validation, balanced model obtained the best results with an average F1 score of 0.86. The balanced configuration score in test iteration 1 is only scored .01 less. This implies the best result from hyperparameter tuning did not greatly improve the performance of the balanced model. However, observing the confusion matrix showed that the model correctly predicted each risk most of the time. While there were more values misclassified for the two highest risks, it is likely to predict one of the last two highest risk categories, rather than the first

two lowest risk classifications. This again was consistent with the results from the XGBoost models.

Feed-Forward Neural Network

When preprocessing the data set for training a neural network, there were two issues to be addressed. First, the data set included columns with both categorical and numerical data as discussed above. To correct for this, the categorical values were encoded into numerical values for model input. Second, the feed-forward neural network model that was selected for this project was not able to process missing values. Previous research had been published on different methods with which to address this issue, some focusing on the imputation of these values, and others accounting for the uncertainty inherent in missing data by including a probability density distribution in the place of any missing data points (Smieja et al., 2018).

In the interest of time, this project chose to implement the former method, imputing missing values, by using the Scikit Learn Multivariate feature imputation class IterativeImputer which "models the missing values as a function of other features and uses that estimate for imputation" (Pedregosa, 2011). At the point of pre-processing for the neural network, all categorical values had been encoded into numerical values, so this was possible. It is worth noting here that missing values were not imputed in the XGBoost or random forest models because those models are equipped to handle missing values, and imputation is another way to introduce bias in the training data. There are issues with imputation of missing values through the implemented method, as the numerical features convey a meaning which might be lost through imputation. For instance, if a flight record includes information from a particular airport, with a particular airplane and engine manufacturer, and particular weather conditions but the lighting condition is missing, there is a possibility that the lighting conditions would be imputed incorrectly, therefore skewing the resulting output of the model. Additionally, late in the project it was discovered that the imputation injected bias back into the model by re-skewing the dataset to elevated risk classification values.

The FFNN utilized three dense layers: two hidden layers with the rectified linear (ReLU) activation function and one output layer with the Softmax activation function allowing for the output to provide a probability distribution over the classes (Bala, n.d.). The ReLU activation function is the most widely used, and the Softmax activation function is typically utilized for a multi-class classification problem such as this.

Like the procedure outlined above for the XGBoost model, the initial FFNN was trained on the original imbalanced data set, resulting in a rather low average F1 score of 0.60, indicating that it only had a slight ability to accurately predict risk categories for accidents and incidents. With a baseline established, multiple tests were conducted, involving the imputation of missing values through different methods and the utilization of a balanced or imbalanced label set. The process and results are enumerated below in Table 6 with F1 scores utilized in the same manner as Table 4. All values included in Table 6 ultimately are based on an imbalanced data set due to the imputation issue described above. More detailed results are included in the appendix.

Table 6

Neural Network modeling results

Test Iteration	Configuration	F1 Score Results
Baseline	Imbalanced, whole data set imputation	0.60
1	Imbalanced, label-based imputation	0.71

2	Imbalanced, label-based imputation	0.96
3	Imbalanced, label-based imputation, 10-fold cross validated using KerasClassifier	0.93
4	Imbalanced, label-based imputation, 10-fold cross validated using StratifiedKFold*	0.97
5	Same as 2 following tuning of hyperparameters	0.94

Note: * indicates the preferred model configuration

The FFNN results following imputation returned average F1 scores in the .90-.97 range, though this should be evaluated with caution. Ultimately the highest-performing method was the 10-fold cross-validation method with the StratifiedKFold implementation using stratified sampling rather than random in order to ensure that every iteration samples from each of the classifications performing slightly better than the simpler non-cross validated model (GeeksforGeeks, 2023). Experiment 5 shows that tuning of hyperparameters of the model resulted in a slightly less accurate output than using the default. Hyperparameters tested through this process were the activation and optimizer methods for the neural network, results indicating that the utilized optimizer was optimal; our model utilized the Adam optimizer, which is "a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments" (Team, n.d.). The tuning indicated that the ReLU activation method used should be switched to Sigmoid, though this is typically used for binary classification problems (Choubey, 2023). This might account for a decrease in performance.

Bidirectional Encoder Representations from Transformer (BERT) for Classification

Another approach taken to model the data was to use a BERT model for multi-class classification. This approach was taken to understand if better performance over the benchmark and the feed-forward neural network could be obtained by utilizing advanced neural networks. The process for modeling the data with BERT was based on the research with similarly structured data by Chris McCormick (McCormick, 2021), and utilized models from the HuggingFace library (HuggingFace, n.d.) and Pytorch libraries (Paszke et al, 2019).

The data set was first balanced using over and under-sampling techniques consistent with the methods used in the benchmark XGBoost models. Each attribute was taken out of the standard data frame structure and combined into one body of text for each record. Additional wording was inserted to provide the model with context for each feature. For example, a string of data like: "rdu 1200 cessna" was transformed to "takeoff airport rdu takeoff time 1200 aircraft make cessna". This mimics a corpus that is traditionally used in natural language processing. Additional experiments were run to measure the effect the additional context had on the results. After transforming the data into combined strings of text, each record was then tokenized and encoded. After tokenizing and encoding, the data set was split into training, testing, and validation sets. Hyperparameters for the model were set. Finally, the model was trained and evaluated.

The evaluation method for this model was consistent with the baseline XGBoost models. F1, accuracy, precision, and recall were measured for each training and validation evolution with the BERT model. In addition to the evaluation metrics, training and validation loss were measured and plotted to check for overfitting. Due to heavy computation costs and training time, the BERT model was trained on two iterations each with a different configuration. Both iterations used

default hyperparameters and model configuration. The first iteration used the data set with additional context, the second iteration only used the raw data without additional context.

Table 7
BERT modeling results

Test Iteration	Configuration	F1 Score Results
1	Data with context	1.0
2	Data without context	1.0

Both iterations of the BERT model produced perfect scores. This is concerning from an overfitting and data leakage perspective. Overfitting indicates that the model has memorized the data set and if presented with unseen information, would not accurately be able to label the data. To check for overfitting the training and validation loss was plotted for each epoch. A well-fit model should have training loss and validation loss decreasing with each epoch and converging towards each other on each epoch. An overfit model would show validation loss increasing while training loss decreases with each epoch. The loss curves shown below in Figure 7 are from test iteration one and indicate that the BERT model has a good fit. This is a positive signal for the model.

In conclusion, the perfect scores for the BERT model must be further scrutinized to ensure the model is not overfitting and there is no data leakage. This model should not be considered for inclusion in a production system until the cause of the perfect scores can be identified and tested. Although the model appears to have a good fit, more research needs to be conducted to ensure that the model can generalize. This should be done with new data that the model has not seen before. Caution should be exercised to ensure a bad model is not put into production.

Figure 7
BERT model training and validation loss curves

Feature Importance

After testing was complete, feature importance was analyzed for both the XGBoost Model and the FFNN as these both have simple methods for doing so. Of note for the XGBoost Model,

the top features used to make decisions on classification included various types of pilot flight hours, airframe hours, airport elevation, temperature, date, and location information. Evaluation of the features of the FFNN was done via both weight magnitude and gradient-based methods. The first evaluates the weights with which a feature influenced a network's decision, while the gradient-based method calculates the loss function gradients in respect to the input features (Theiler, 2022). Each indicated different features that are important to the classification performance of the model, though both were a mix of meteorological, event, and aircraft specific information among the highest scored features. Visualizations of feature importance for both models can be found in Appendices B and C.

Conclusions

Overall, this project succeeded in solving the problem it initially set out to solve. Risk categories were identified, and models were trained with reasonable accuracy to predict those categories. Although reasonable outcomes were achieved, there is still much work that can be done to improve the foundational work. As it stands, the models presented in this project should not blindly be incorporated into a production-level application that predicts flight risk for pilots. First, additional work must be done to address the limitations described in the subsequent sections. Second, the work should be peer-reviewed to ensure the results are reproducible, and methodologies are validated.

Model Comparison

Each model proposed, trained, and tested in this report has its own strengths and weaknesses. Overall, when selecting a superior model, it is important to weigh the strengths and weaknesses of each model and configuration to ensure poorly fit or heavily biased models are not utilized in a production environment. To summarize, all the models that were trained during this project showed some degree of ability to accurately predict risk categories.

XGBoost and random forest models showed very promising results with room for improvement. Overall, both the XGBoost and random forest models were considered the strongest performers but in terms of quality of result metrics. Model improvement can most likely be achieved by addressing some of the limitations in the next section. For example, if more data was available in the low-risk categories, like categories 0 and 1, there would be less requirement for balancing the data set through over and under-sampling. This should also reduce the overfitting characteristic of the model on the low-risk categories.

The FFNN that was utilized showed high scores and results, but there were problems with imputation. After further evaluation the imputer injected too much bias and skewed both the data set distribution and the model output to the point where it almost always predicted risk category 2 exclusively. For these reasons, XGBoost and the random forest models are preferred over FFNN. Improved performance of FFNN models can be addressed through exploration of improved missing value imputation, the addition of unique low-risk category records, as well as expanding the data set in general as these models work better with larger data sets that are low in missing values. This would eliminate the need for heavy imputation.

XGBoost and random forest models are preferred to neural networks in multi-class classification problems such as this one for a few reasons. Neural networks require larger data sets, and higher computational resources, and cannot process missing values. With a larger data

set and limited missing values, the neural network might become more desirable through future iterations of this project. Additionally, because they are simpler models, the XGBoost and random forest models are easier to manipulate, train, and validate than the neural networks explored here such as FFNN and BERT.

The BERT model, although at first glance looks to be the best model based on F1 scores, should be considered the least desirable model to implement into a production application at the time of this report. This is because the model produced perfect scores and extremely minimal loss. This points to overfitting or data leakage, or a combination of the two. Due to the time and resource constraints on this project, it was not possible to identify and mitigate the reason for these scores, though this issue is still being investigated. This would be a good idea for future research because the other models utilized in this project showed promising results, perhaps BERT could outperform the other models if the problems are identified and appropriately solved.

Limitations

There are numerous limitations remaining hindering progress of this project, falling into two categories: data availability limitations, and model limitations.

Regarding data availability limitations, it is important to note that the data set utilized for training these models only includes records from accident and incident NTSB databases. There is a difference in data output and quality between preflight planning data for a non-incident related flight and investigation data. Investigation data is collected by an investigator who did not plan the flight and is collecting data after an event occurred. Often, in the case of fatal events, the person who planned the flight is deceased and unable to provide the investigator with flight data. This is an opportunity for biased data collection on the part of the investigator or incomplete and missing data. Due to the inability to collect general aviation flight records, or pre-flight data from pilots for this project, modeling was limited to predicting risk categories only on accident and incident records. Therefore, the research presented in this report more appropriately answers the problem: given an accident or incident, predict the risk category for the event. If regular pre-flight data were to be obtained and incorporated into modeling, new testing, and evaluation would be required for the new models. Adding this data would also reduce bias in the models and would allow for risk category prediction and risk probability prediction which would be very beneficial to the pilot.

It is important to recognize that all models have limitations, and no model can predict reality all the time. Each of the models utilized in this project was limited in some way. Regarding the FFNN, the model could not handle missing values, and therefore missing values had to be imputed. This can inadvertently inject bias into the model in the way that the imputation algorithms are built through the artificial insertion of conditions which might not have been relevant to each flight. The BERT model's primary limitation was that it is a very complex model to configure and train. Therefore, troubleshooting the model, in this case, trying to understand the reasons behind overfitting, became very difficult. The XGBoost and random forest models were limited because the data sets had to be balanced to increase the result scores. The methods by which the data set was balanced injected bias into the model because the low-risk categories were duplicated almost 100x. This caused overfitting in those categories. Although overfitting is typically avoided, in the case of XGBoost, it was not completely detrimental to the results. This overfitting creates a bias in the model to predict higher risk categories when it incorrectly classifies a label. It is better to have a more conservative model,

than a model that incorrectly labels a high-risk flight as low-risk. Most of these limitations were addressed and overcome by utilizing a method of testing, evaluating, reconfiguring, and retesting, until a suitable solution was obtained, or performance could no longer be increased. Finally, the results were cross validated to ensure final scores were not anomalies.

Future Model and Application Development

For a modernized FRAT application to be successful it must be accessible, simple to use, fast, and accurate. A modernized FRAT can be developed and implemented through the following steps. First, since the FAA already owns and manages the source data (NTSB incident reports), a simple extract, transform, and load (ETL) pipeline can be developed to periodically update new incident and accident data, transform the data for model preparation, and load it to an existing data warehouse. Next, a programmed script can be implemented to train the model with new information as it is added. Then, the model would be put into production and connected to a modern web application.

This interface would make the program accessible to pilots during preflight planning. Pilots should be able to set up accounts, where they can update their recent flight hours, their aircraft data, and their flight plan, so they do not have to input this data on each use of the application. Finally, the application would ask the pilot a series of questions to ensure all relevant features are available to the model to make a risk assessment prediction. After the pilot inputs the data requested by the application, the data is run against the recurrently trained machine learning algorithm and a risk score is produced with details highlighting which attributes lead to the elevated risk score calculation. This will alert the pilot to the most probabilistic and dangerous hazards present in each flight scenario, to which the pilot can apply the appropriate level of mitigation. One method for generating this output would be to run a K-Nearest Neighbor algorithm to identify the most similar flight records within the predicted risk category, and then extract the most common features from these records to indicate areas of potential risk mitigation.

The best way to interface through which to implement this model is to incorporate it into preflight planning software such as ForeFlight or through a custom phone application to allow real-time risk analysis. Through such software, much of the required information would already be made available, such as the pilot's historical flight data, the filed flight plan, and the current and forecasted weather based on the route. The ease of use and minimal pilot input required would save time and could improve the probability that a risk assessment is done in the first place.

Though the scope of this project only included foundational research into modeling risk and did not include application development, several areas for future development have been identified. Through the incorporation of this tool in preflight planning software, there is potential to integrate the use of health informatics via the integration of smartwatch data. Wingelaar-Jagt et al. (2021) examine the safety risks of fatigue in aviation in their Frontiers in Physiology publication, stating that it is an "important risk factor for aircraft incidents and accidents both in civil and military aviation. In the last two decades, it has been identified as the probable cause of 21–23% of major aviation [accident] investigations" (para. 3). A pilot might not be able to adequately judge their level of fatigue resulting from previous flights or other physiological issues, so wearable monitoring of physiological data could potentially enable a nonbiased assessment to be integrated into a predictive model. This would require a test group of pilots to wear such a device over a period to build an adequate data set, as well as further advancement in fatigue research. Adão

Martins et al. (2021) indicate that the currently available published literature does not indicate that this technology is ready.

For this application to be accepted and widely used in the GA community, several efficiencies should be built into the application for ease of use. First, pilots could have profiles where aircraft data and pilot flight hour data are tracked automatically as previously mentioned. Second, the pilots could save flight plans in their profiles and the model would pull aircraft, airport, and flight path data from those flight plans. Finally, the pilot would be prompted to fill out any additional data the model does not have. All these features would make the risk assessment process accessible and efficient for the pilot. The faster and easier the application is to use, the more likely the pilot is to use the application and apply the risk assessment to their flight.

According to Shappel et al. (2010) in the 20 years proceeding their study nearly 40,000 GA "aircraft were involved in accidents, roughly 20% of which were fatal" (p. 1). According to Wright, most accidents can be traced back to preflight planning (2018). Due to the high cost of property and life, and the lack of an easily accessible and accurate FRAT tool, this project sought to fill a critical need for GA pilots. It is our hope that this research can be built upon and improved to create a modernized flight risk assessment tool that enables a more robust preflight planning culture in the general aviation community and improves hazard identification and mitigation, improving the safety of flight for all who use it.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., & Zheng, X. (2016). TensorFlow: A System for Large-Scale Machine Learning. OSDI (p./pp. 265--283).

Adão Martins, N. R., Annaheim, S., Spengler, C. M., & Rossi, R. M. (2021). Fatigue Monitoring Through Wearables: A State-of-the-Art Review. Frontiers in Physiology, 12. https://doi.org/10.3389/fphys.2021.790292

AIDS System Information Page (n.d.). Retrieved January 9, 2023, from https://www.asias.faa.gov/apex/f?p=100%3A15%3A%3A%3ANO%3A%3AP15_REGION_VAR%3A1#:~:text=The%20FAA%20Accident%20and%20Incident,NTSB)%20definition%20of%20an%20accident.

Bala, P. C. (n.d.). Softmax Activation Function: Everything You Need to Know. Pinecone. https://www.pinecone.io/learn/softmax-activation/#:~:text=The%20softmax%20activation%20function%20simplifies,distribution%20over%20the%20input%20classes.

Bazargan, M., & Guzhva, V. S. (2011). Impact of gender, age and experience of pilots on general aviation accidents. Accident Analysis &Amp; Prevention, 43(3), 962–970. https://doi.org/10.1016/j.aap.2010.11.023

Boyd, D. D. (2017). A Review of General Aviation Safety (1984–2017). Aerospace Medicine and Human Performance, 88(7), 657–664. https://doi.org/10.3357/amhp.4862.2017

Bruce, P., Bruce, A. & Gedeck, P. (2020). Practical Statistics for Data Scientists (2nd ed.). O'Reilly Media, Inc.

CAROL Help. (n.d.). Www.ntsb.gov. Retrieved May 8, 2023, from https://www.ntsb.gov/Pages/CAROL.aspx

Choubey, V. (2023, February 6). Activation Functions in Neural Network: Steps and Implementation. Medium. https://medium.com/codex/activation-functions-in-neural-network-steps-and-implementation-df2e4c858c21

ForeFlight - Personal Aviation. (n.d.). https://foreflight.com/solutions/personal/

Fultz, A. J., & Ashley, W. S. (2016). Fatal weather-related general aviation accidents in the United States. Physical Geography, 37(5), 291–312. https://doi.org/10.1080/02723646.2016.1211854

GeeksforGeeks. (2023). Stratified K Fold Cross Validation. GeeksforGeeks. https://www.geeksforgeeks.org/stratified-k-fold-cross-validation/

General Aviation Joint Steering Committee. (n.d.). Flight Risk Assessment Tools. In www.FAASafety.gov (AFS-850 16_12). FAA Aviation Safety. https://www.faa.gov/news/safety_briefing/2016/media/SE_Topic_16-12.pdf

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. (2020). Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.

Herzmann, D. (n.d.). IEM :: Download ASOS/AWOS/Metar Data. Iowa Environmental Mesonet. Retrieved January 9, 2023, from https://mesonet.agron.iastate.edu /request/download.phtml?network=NC_ASOS

HuggingFace Inc. (n.d.). Transformers: AutoModelForPreTraining (model documentation). Retrieved May 7, 2023, from https://huggingface.co/docs/transformers/model_doc/auto#transformers.AutoModelForPreTraining

Hunter, J. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95.

Johnston, & Mathur, I. (2019). Applied supervised learning with Python : use scikit-learn to build predictive models from real-world datasets and prepare yourself for the future of machine learning (1st edition). Packt Publishing Ltd.

Lemaitre, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research, 18(17), 1-5.*

McCormick, C. (2021, June 29). Combining categorical & numerical features with BERT. Retrieved October 19, 2021, from https://mccormickml.com/2021/06/29/combining-categorical-numerical-features-with-bert/

McKinney, W. (2010). Data structures for statistical computing in Python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).

National Transportation Safety Board (2021). Aircraft Accidents Database (Version 1.0) [Data set]. NTSB Accident Database. Retrieved from https://data.ntsb.gov/carol-main-public/

NASA. (n.d.). ASRS database online - aviation safety reporting system. NASA. Retrieved January 9, 2023, from https://asrs.arc.nasa.gov/search/database.html

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32 (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, from https://scikit-learn.org/stable/about.html

Perneger, T. V. (2005). The Swiss cheese model of safety incidents: are there holes in the metaphor? BMC Health Services Research, 5(1). https://doi.org/10.1186/1472-6963-5-71

Risk Management Handbook (FAA-H-8083-2). (2009). Federal Aviation Administration. Retrieved January 11, 2023, from https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/media/faa-h-8083-2.pdf

Scikit-learn. (n.d.). 6.4. Imputation of missing values. https://scikit-learn.org/stable/modules/impute.html

Shappell, S., Hackworth, C., Holcomb, K., Lanicci, J., Bazargan, M., Baron, J., Iden, R., & Halperin, D. (2010). Developing Proactive Methods for General Aviation Data Collection. Retrieved from https://commons.erau.edu/publication/1221

Smieja, M., Struski, Ł., Tabor, J., Zieliński, B., & Spurek, P. (2018). Processing of missing data by neural networks. In Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18) (pp. 2724–2734). Red Hook, NY, USA: Curran Associates Inc.

Team, K. (n.d.). Keras documentation: Adam. https://keras.io/api/optimizers/adam/

Theiler, S. (2022, July 5). tf.GradientTape Explained for Keras Users - Analytics Vidhya - Medium. Medium. https://medium.com/analytics-vidhya/tf-gradienttape-explained-for-keras-users-cc3f06276f22

Waskom, M. L. (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021.

Wingelaar-Jagt, Y. Q., Wingelaar, T. T., Riedel, W. J., & Ramaekers, J. G. (2021). Fatigue in Aviation: Safety Risks, Preventive Strategies and Pharmacological Interventions. Frontiers in physiology, 12, 712628. https://doi.org/10.3389/fphys.2021.712628

Wright, R. (2019, October 29). Risk assessment tools. Aviation Safety. Retrieved January 11, 2023, from https://www.aviationsafetymagazine.com/features/risk-assessment-tools/

Appendix A: Detailed Best Model Results

XGBoost – Test Run 6 – 10-fold cross-validation on balanced data set

Metric	Class 0	Class 1	Class 2	Class 3	Average All Classes
F1	0.99	0.98	0.71	0.69	0.85
Precision	0.99	0.97	0.70	0.71	0.84
Recall	1.0	1.0	0.72	0.68	0.85

Random Forest – Test Run 2 – 10-fold cross-validation on balanced data set

Metric	Class 0	Class 1	Class 2	Class 3	Average All Classes
F1	1.0	0.99	0.71	0.69	0.86
Precision	1.0	1.0	0.75	0.65	0.85
Recall	1.0	0.99	0.68	0.74	0.85

Feed-Forward Neural Network – Test Run 4 – 10-fold stratified cross-validation on balanced data set using label-based imputation

Metric	Class 0	Class 1	Class 2	Class 3	Average All Classes
F1	NC	NC	NC	NC	0.97

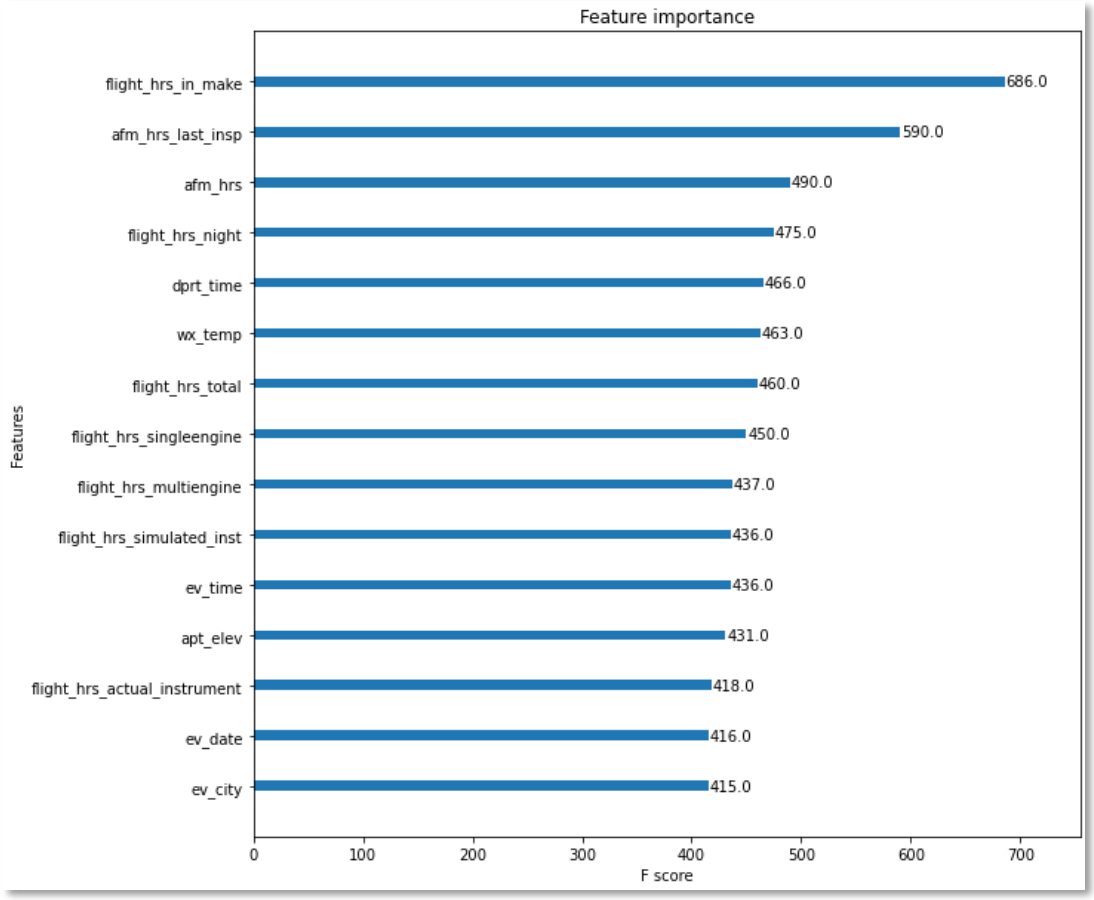
BERT – Test Run 1 – no context

Metric	Class 0	Class 1	Class 2	Class 3	Average All Classes
F1	NC	NC	NC	NC	1.0
Precision	NC	NC	NC	NC	1.0
Recall	NC	NC	NC	NC	1.0

Note: NC = not calculated

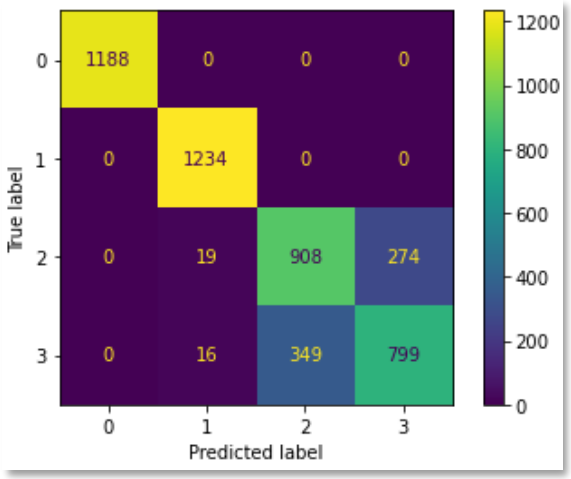
Appendix B: XGBoost Supplemental Visualizations

XGBoost Model Feature Importance:



This chart shows the top 15 most important features in the XGBoost decision trees. The top factors included various pilot flight hours, airframe hours, airport elevation, temperature, date, and location information.

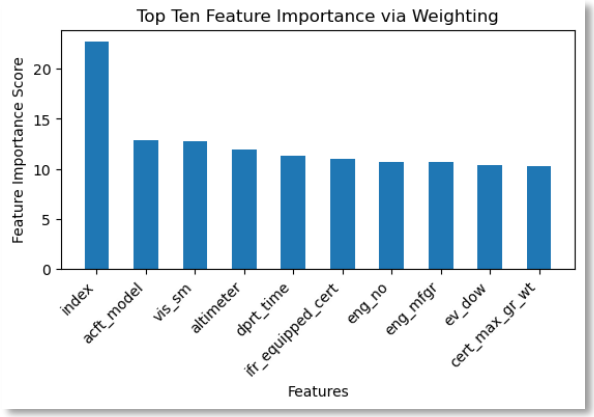
XGBoost Confusion Matrix:

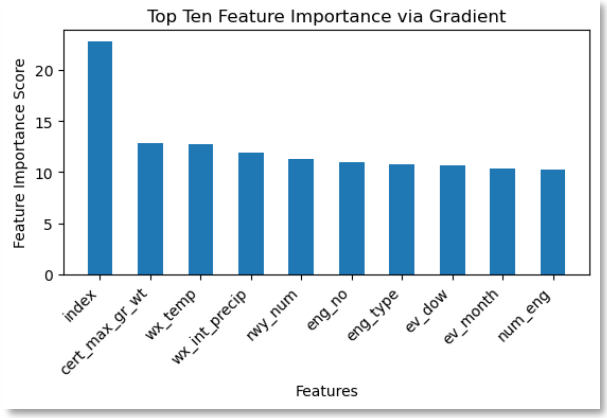


This is a sample of one of the confusion matrices that was produced in the output of the XGBoost 10-fold cross-validation testing. The y-axis represents True labels, and the x-axis represents Predicted labels. Overall, there are nearly perfect predictions for labels 0 and 1. This is due to the oversampling techniques used. It can be observed that the model predicts high, which compensates for the overfitting in the low-risk classes.

Appendix C: Feedforward Neural Network Supplemental Visualizations

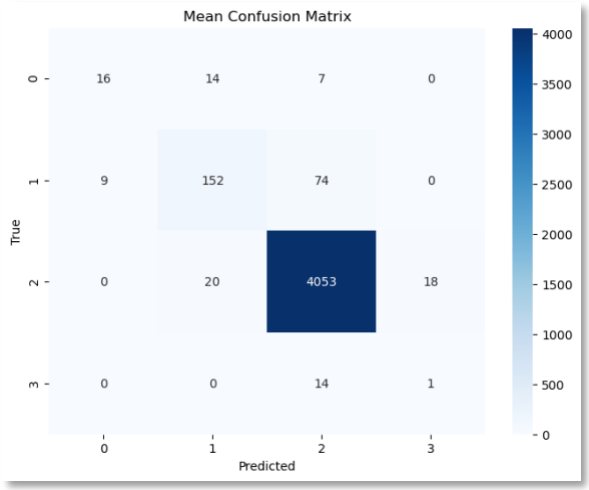
FFNN Model Feature Importance:





These charts shows the top 10 most important features to the neural network classification via either weight magnitude or gradient-based methods as described in the paper. The top factors included various meteorological, aircraft, and event specific (time/location) information. The "index" on the x axis refers to how the model is passing each record into the neural network, and as such can be ignored for feature importance evaluation.

FFNN Confusion Matrix:



This is a sample of the confusion matrix that was produced in the output of the FFNN 10-fold cross-validation testing. The y-axis represents True labels, and the x-axis represents Predicted labels. This confusion matrix shows the issue described in the model comparison section, with the classification predicted almost exclusively as label 2 due to a imbalanced training set.