**Final Project Report**

**IST 707: Applied Machine Learning**

**“Predicting the Aircrash Severity: NTSB Aviation/Airplane Accidents”**



**By: Mohammed Huzaif Kherani, Sai Krishna Bolla, Samar Zemeer, Shabib Khan**

## **I**ntroduction

Flying has long been the preferred method of transportation since it is quick, inexpensive, and so convenient. As of June 2019, the FAA reported that 2,781,971 people flew daily in the United States. Investigators from the Washington, D.C.-based National Transportation Safety Board (NTSB) are tasked with sifting through the wreckage, analyzing critical data from cockpit voice recorders and flight data recorders, the so-called “Black box” and perusing maintenance records, weather data, and communications with air traffic controllers. But even after painstaking probes, there isn’t always a clear answer to why these catastrophes occur. Instead, plane crashes often result from the interaction of a combination of factors, according to NTSB public affairs.

The NTSB focuses on initiatives to lower General Aviation accidents, redoubled with the Federal Aviation Administration's Safer Skies Initiative in 1998. The fatality rate in general aviation has declined over the past 22 years. Although there is a decline, accidents still occur, though, and there is some evidence that indicates an increase in the number of accidents that indicate hazard exposure. The accident data suggest that there is still more the aviation community needs to understand about the factors involved in an accident sequence. So, aviation has changed into a high level of operations technically, administratively, and even technologically. This has been made possible with the introduction and manufacture of larger and faster aircraft that incorporate advanced information management technologies. With an increase in the volume of traffic and the demand for air transportation comes the risk and rise of air crashes. Various researchers and bodies have studied the causes of air crashes.

This study examines the causes of air crashes in the United States. It raises the following pertinent business questions:

* To anticipate and classify the severity of any airplane accident based on past accidents?
* What are the different factors that affect the severity of an airplane crash?
  + Factors related to the aircraft
  + Factors in conjunction with human elements and other externalities.

The study results provide government, business, and pilots with valuable information specifically in the following recommended areas:

1. Enhance the quality and use of accident reports for machine learning applications
2. Redouble efforts to enhance flight abilities and counteract decision-making faults
3. Highlight the significance of weather briefings, preflight preparation, and weather-based risk management
4. Develop an aviation-specific corpus for text mining to enhance text analysis and transformation.
5. Determine methods for collecting and publishing more open-source flight data for safety modeling.

## Methods

***The Data***

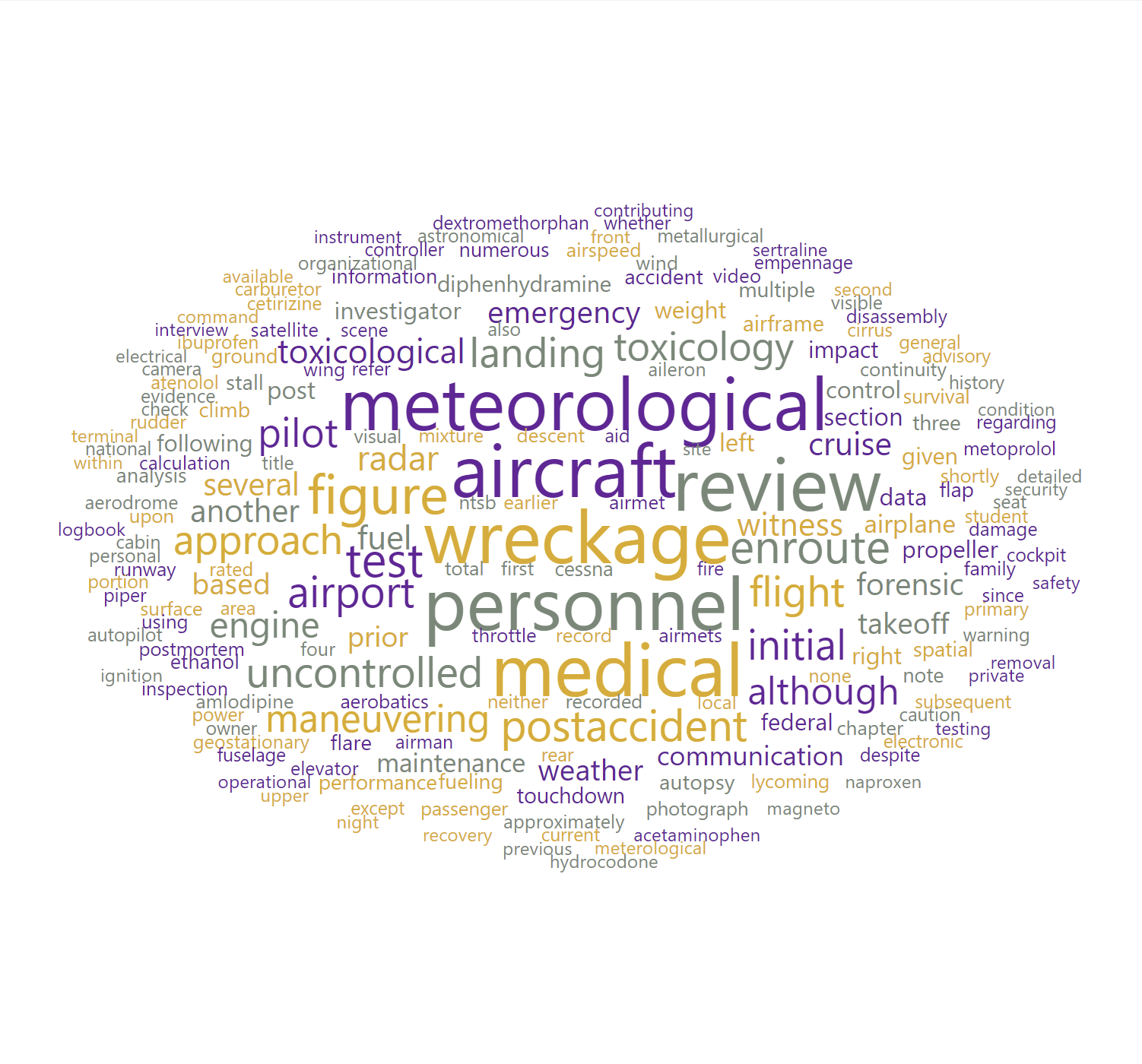
The National Transportation Safety Board (NTSB) analyses every accident involving civil aircraft in the United States and publishes a complete dataset for each year beginning from 1982, updated monthly in Microsoft Access 2000 MDB format. The dataset has been downloaded from the official [NTSB website](https://www.ntsb.gov/Pages/AviationQuery.aspx), and the following data preprocessing steps were carried out.

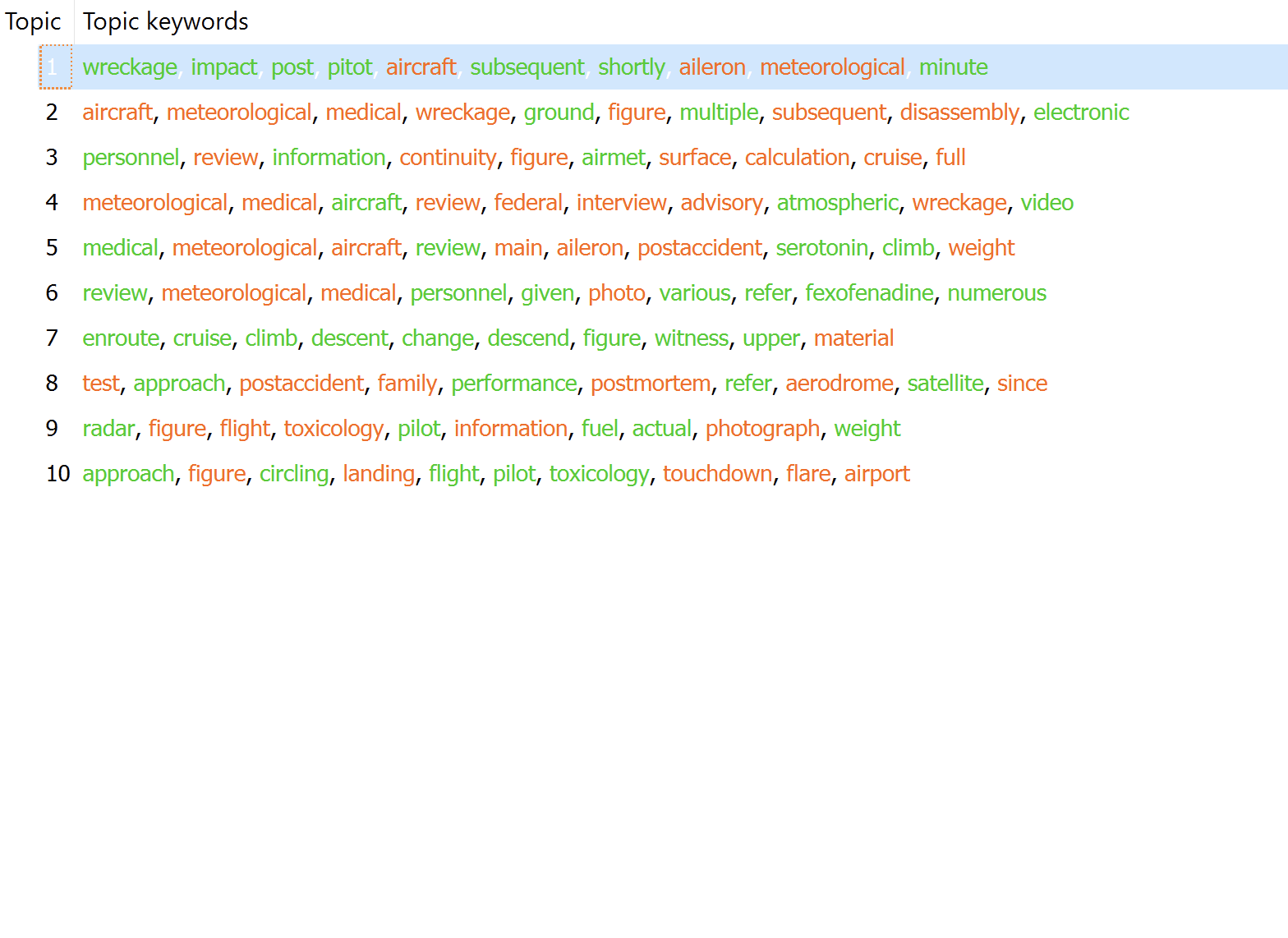
Extracting the data using a combination of SQL Queries and Excel has resulted in 12 sheets, with each individual data sheet containing data about aircraft, weather, pilot, events, etc., with a total of 150+ variables and a disproportionate amount of data (~25,000 rows in some and ~86,000 data pointers in others) in each sheet. However, every accident has been assigned a unique “event id” that distinctly helps to identify the data regarding each accident across all the sheets. But, given a large amount of data and data variables, it was proven to be computationally expensive to merge all the sheets as is, and the output dataset is chaotic.

To solve this problem, we have carried out text mining on the accident narratives data to understand the common variables causing air crashes. After following the below-mentioned steps to preprocess the text, a word cloud along with topic modeling has been carried out, and the results are as follows.

Preprocess Text:

* Transformation: turned all text into *lowercase*, removed the *accents and URLs, and*  parsed *HTML.*
* Tokenization: the text is then broken down into smaller components using *Regexp* of 4+ letters.
* Normalization: the text has been lemmatized using WordNet Lemmatizer, which applies a network of cognitive synonyms to tokens based on a large lexical database of English.
* Filtering: Stopwords, along with numbers and Regexp, were filtered out of the text. Building a word cloud for few times resulted in additionally filtering the words: *according | additional | examination.*

The resulting word cloud and topic models suggested that **aircraft, personnel,** and **meteorological**  factors played a crucial role in the majority of aircraft accidents, according to the accident report narratives.

*Figure (1): Output of Topic Modeling and Word cloud*

Based on the above output and referring to the NTSB [Official Aviation Query website](https://www.ntsb.gov/Pages/AviationQuery.aspx) and prior literature, a set of ~45 variables were selected relating to meteorological, aircraft, and personal factors - a brief description of which is provided in the appendix section.

Before proceeding with designing a final merged dataset for analysis, the following two factors/assumptions were made:

1. The NTSB database consisted of data about accidents of various types of aircraft (airplanes, helicopters, etc.). However, for the purposes of this analysis, only data pertaining to airplanes were considered.
2. Even though NTSB primarily investigates civil aviation accidents within the United States, the database has some information regarding accidents that happened on foreign soil too. For the purposes of this analysis, such data has been filtered out (a minor percentage), and only civil aviation accidents that happened in the United States were considered.

The aircraft and events sheets were merged using the inner join command with matching “*NTSB no*” rows. For the subsequent sheets pertaining to the flight crew, flight\_time (that consisted of data regarding Pilot time) and engines were merged, matching the rows in the “*NTSB event id*” column. (Other transformations have also been carried out in individual data workbooks as a preparatory step but were not discussed here due to report constraints). After the initial preprocessing step, the master dataset consisted of ~45+ variables and ~20000 data pointers.

Before proceeding further, we have divided the data into two subsets, wherein the first subset only focused on aircraft-related factors, and the second subset focused on meteorological, aircraft, and personal factors.

***First data subset***

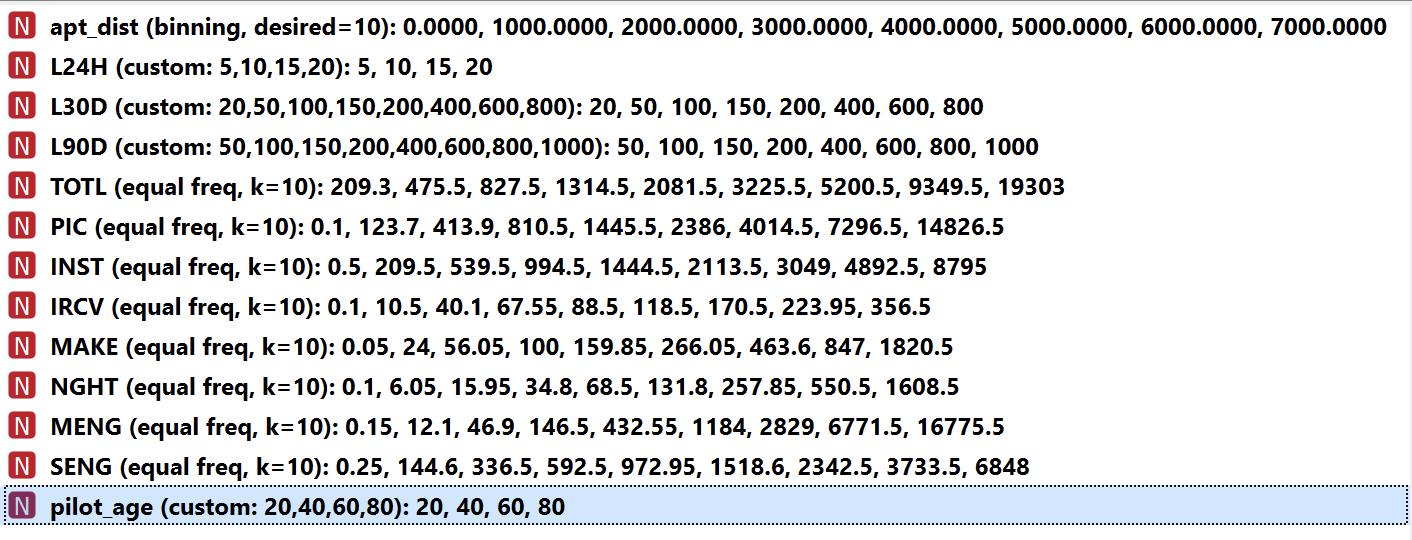
For the first dataset, there are a total of 10,000 values in the training dataset and 2000 values in the test dataset. The twelve variables in our first data set are:

| Severity | A description (4 level factor) on the severity of the crash |
| --- | --- |
| Safety Score | A measure of how safe the plane was deemed to be |
| Days since Inspection | How long the plane went without inspection before the accident |
| Total Safety Complaints | Number of complaints from mechanics prior to the accident |
| Control Metric | An estimation of how much control the pilot had during the accident given the factors at play |
| Turbulence in gforces | The recorded/estimated turbulence experience during the accidents. |
| Cabin Temperature | The last recorded temperature before the accident, measured in degrees fahrenheit. |
| Accident Type Code | The type of accident |
| Max elevations |  |
| Violations | Number of violations that the aircraft received during the inspections. |
| Adverse Weather Metric | An estimation of the weather |
| Accident ID | Unique ID assigned to each row |

***Second data subset***

**Initial steps (in R):** All the columns with only NA values were deleted, the rows with the ‘Incident’ investigation type were removed, all the numeric variables were set to type numeric, changed the data variable suitable for EDA, and the blank, "N/A,” "Unknown,” and similar responses were changed to NA.

In the final step, a new variable with the maximum injury outcome (Fatal, Injury, or None) for every record with at least one non-NA value for the number of fatalities and # injuries (serious, minor, or none) was added. The resulting dataset consisted of ~18,000 rows of 40 variables and was loaded into the Orange Data Miner.

**Further steps (in Orange):** After running the initial models, we found a need to discretize the following pilot-related factors, to improve the accuracy of the models further.

*Figure (2): Initial Discretization of “Pilot” factors in Orange Data Miner*

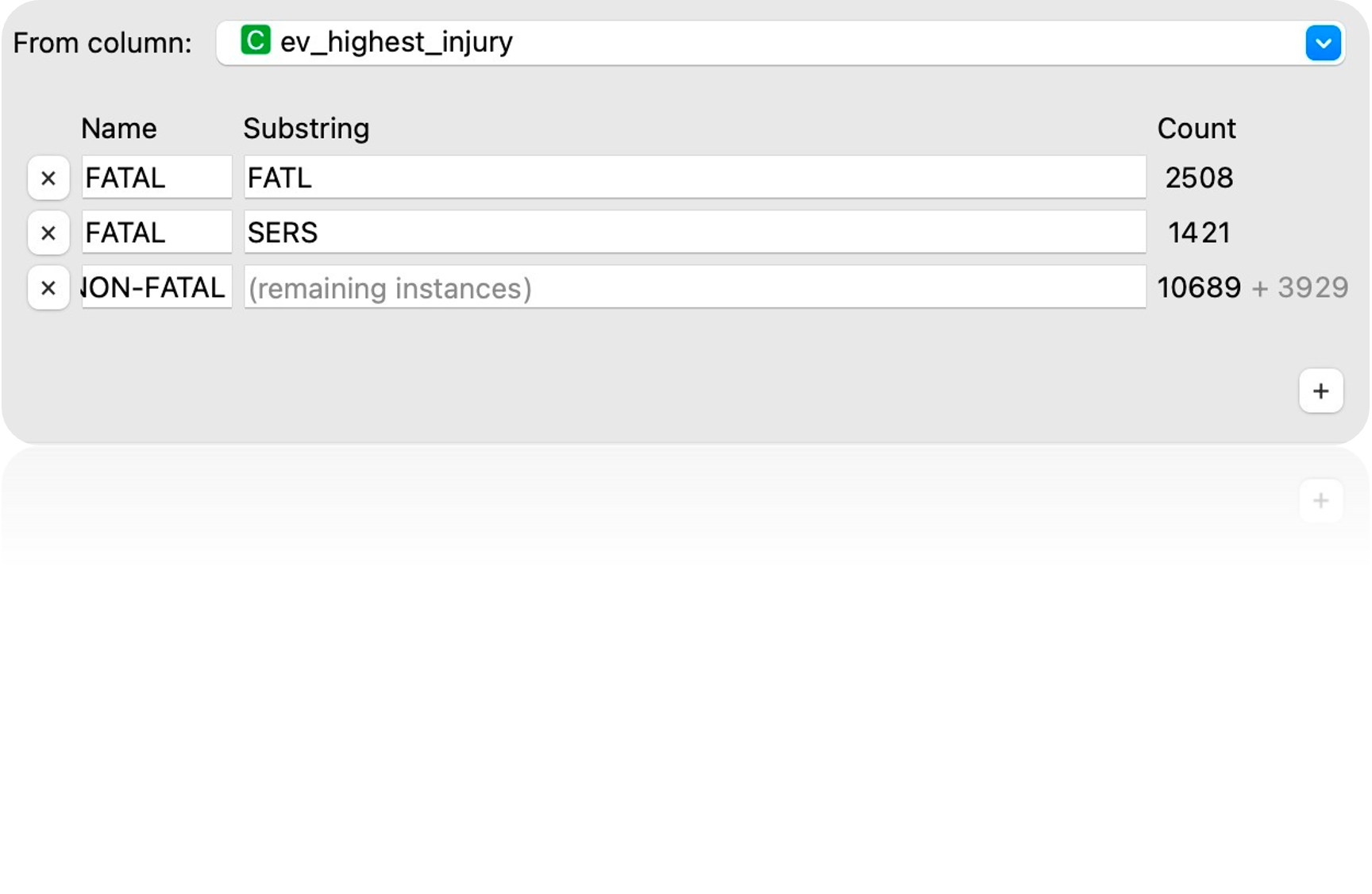
Building an initial decision tree, we achieved 98% accuracy. However, by diagnosing the tree output more closely, we understood that the following variables (erroneously) influenced the final output.

a. crew\_tox\_perf: Whether a toxicity check has been performed on the crew before flight.

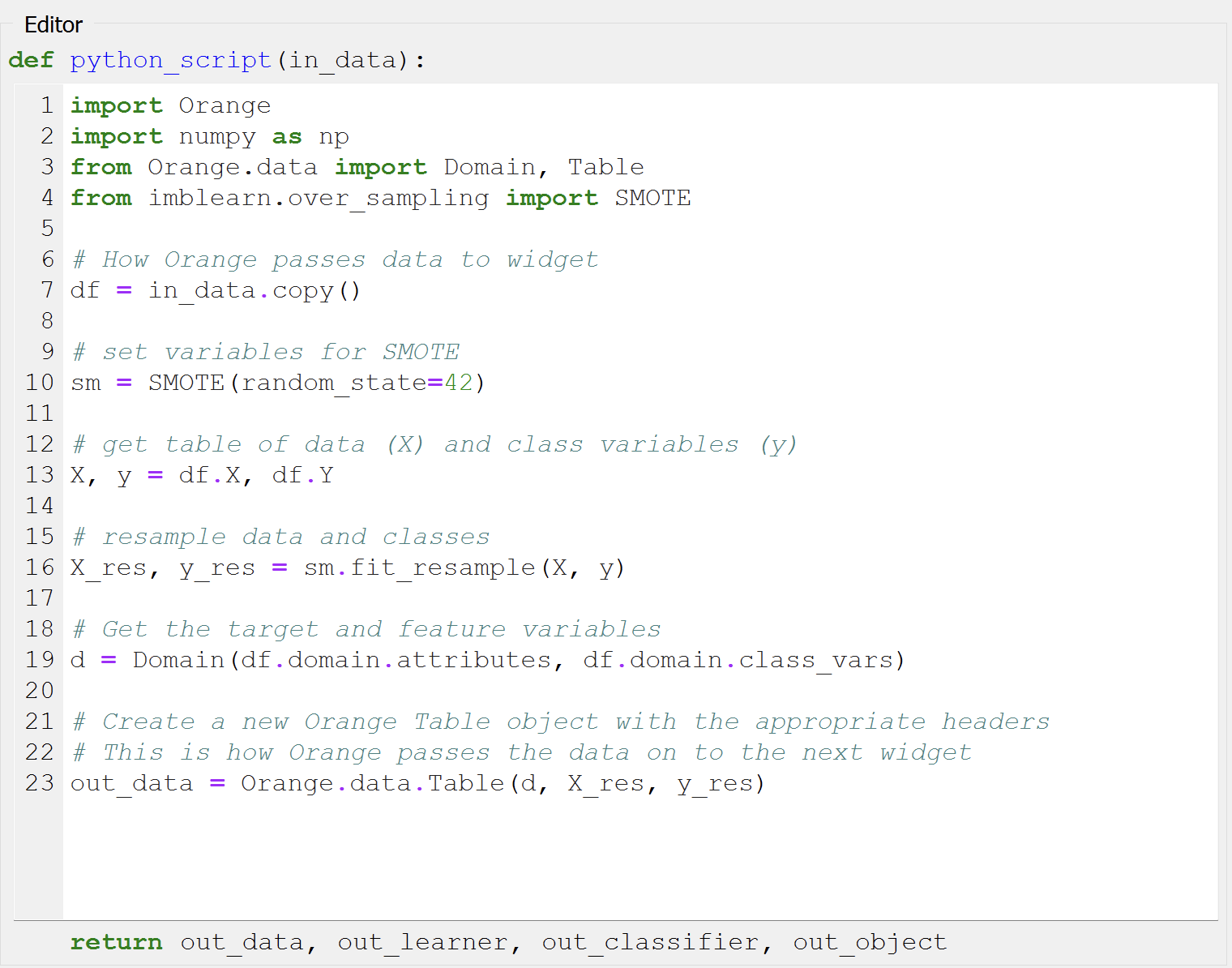
b. damage: Level of damage to the aircraft (Minor/Substantial/Unknown)

c. pilot\_inj\_level: Injury level of pilot resulting from the accident (Fatal, Minor, Serious).

The *target variable initially consisted of four values - None, Minor, Serious, and Fatal.* After building an initial decision tree, we understood a dichotomous categorical variable would help to increase the model’s accuracy better. Subsequently, a class has been created using the “Create Class” function, and the target variable is divided into two values: Fatal (/serious) and non-Fatal.



*Figure (3): Panel showcasing the ‘Create Class’ function and Class imbalance.*

It is evident from the above figure that the target variable has a disproportionate amount of values, i.e., a class imbalance. Initially, running the Machine Learning models with the dataset as is resulted in an ~75% accuracy but observing the confusion matrix, it became apparent that the non-Fatal value is predicted with better accuracy (at ~87%) than the Fatal value (~69% accuracy) due to this imbalance.

Hence, the data was first imputed using a model-based imputer (simple tree) which essentially, depending on the values of other characteristics, creates a model to forecast the missing value. After the data is imputed, a python script based on the addon “Imblearn” is run to balance the classes, essentially oversampling the data. This resulted in better accuracy, as evident from the below-discussed results section.

*Figure (4): Python Script utilizing “Imblearn” to balance the Class imbalance.*

**Train and Test Split**: The approach we have followed for building the train and test data is as follows:

1. The above imputed and over-sampled data is split into train and test data with 70/30 proportion using the Data Sampling function in Orange - with replicable deterministic sampling. All the original data before sampling is used to build the model, and the model validation is performed using the test split.
2. The original data (before imputing and oversampling) is split into train and test data in 80/20 proportion using the Data Sampling function in Orange - with replicable deterministic sampling. The 80% proportion is further imputed and follows the procedure mentioned above. The model is then tested on the remaining 20% dataset.

## Results

To reiterate, we have followed the following three-step approach to solve our business problem using data mining techniques:

* Approach - 1: Predicting the air crash severity using only aircraft-related factors
* Approach - 2: Predicting the air crash severity using only personnel (pilot) factors
* Approach - 3: Predicting the air crash severity using a combination of aircraft, personnel, and meteorological factors

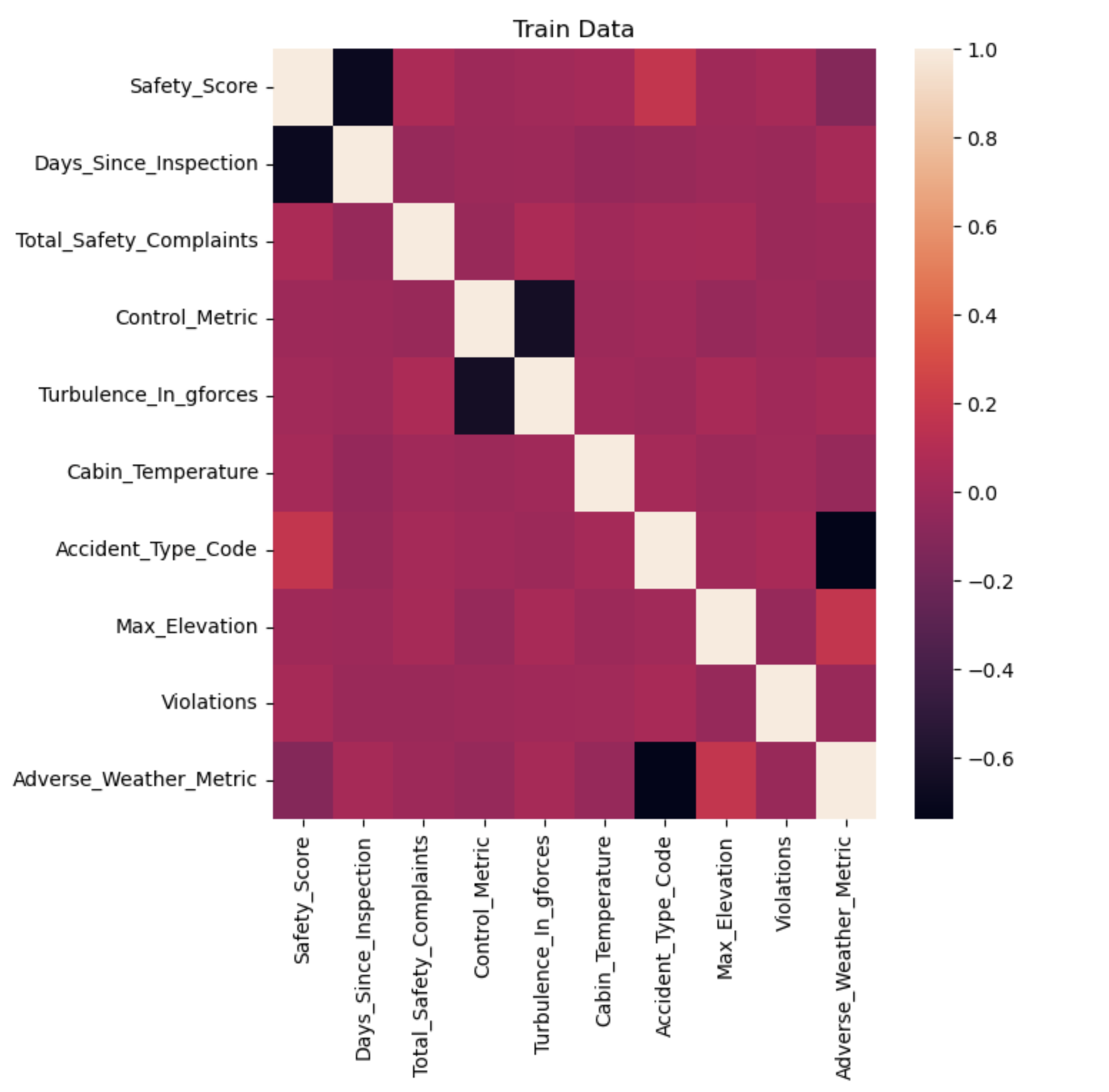
Given the categorical nature of our target variable, and a majority of the independent variable, the following data mining approaches were followed:

1. **Decision Tree**: Decision Tree refers to statistical modeling that uses a form of the decision tree, where node splits are decided based on an information metric. The data in the decision tree is split based on information from features.
2. **Random Forest:** Random Forest constructs a set of classification trees when given a set of labeled data. When creating individual trees, a random selection of attributes is selected (thus the term "random"), and the optimal attribute for the split is chosen from this subset. Classification is determined by the majority vote of the forest's independently built tree classifiers.
3. **Gradient Boosting**: Gradient Boosting is a machine learning technique for regression and classification problems that generate a prediction model as a collection of weak prediction models, often decision trees.
4. **Logistic regression**: Logistic regression is an algorithm for classifying data. It predicts a binary outcome using a collection of independent variables.
5. **Neural Network:** A neural network is a collection of algorithms that attempts to recognize underlying relationships in a data set by emulating how the human brain functions.

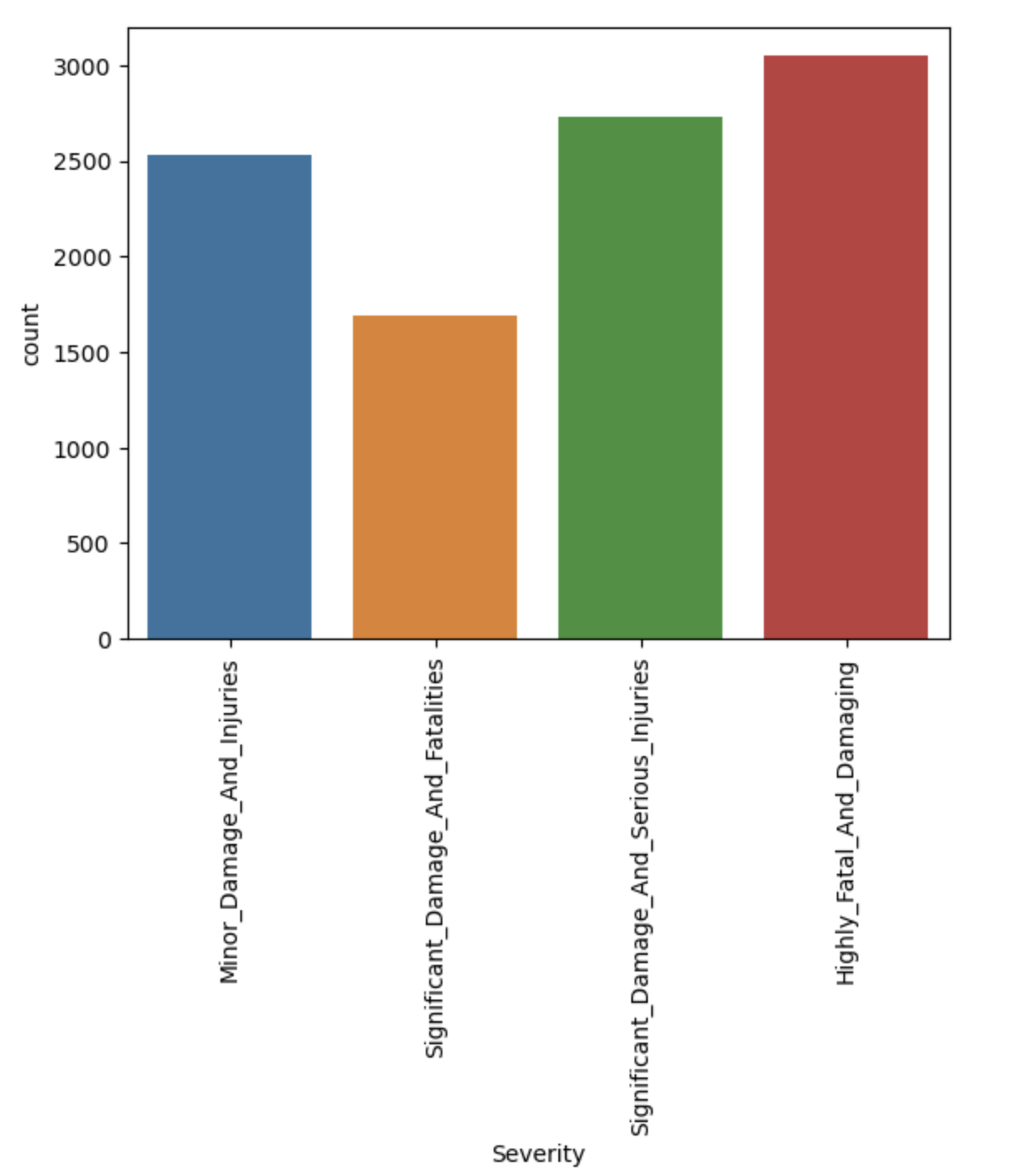
***Exploratory Data Analysis***

We first ran some exploratory analysis on the dataset to better understand our data. EDA enables us to provide greater data insights. In order to determine the connection between our variables, we first created a heatmap of the correlation matrix.

According to the below heatmap, there is a strong negative correlation between Days since the inspection and Safety Score, a strong negative correlation between Turbulence in g-forces and Control Metric, a weak positive correlation between Accident type code and Safety Score, a strong negative correlation between Adverse Weather Metric and Accident Type Code, and a weak positive correlation between Adverse Weather Metric and Max Elevation.

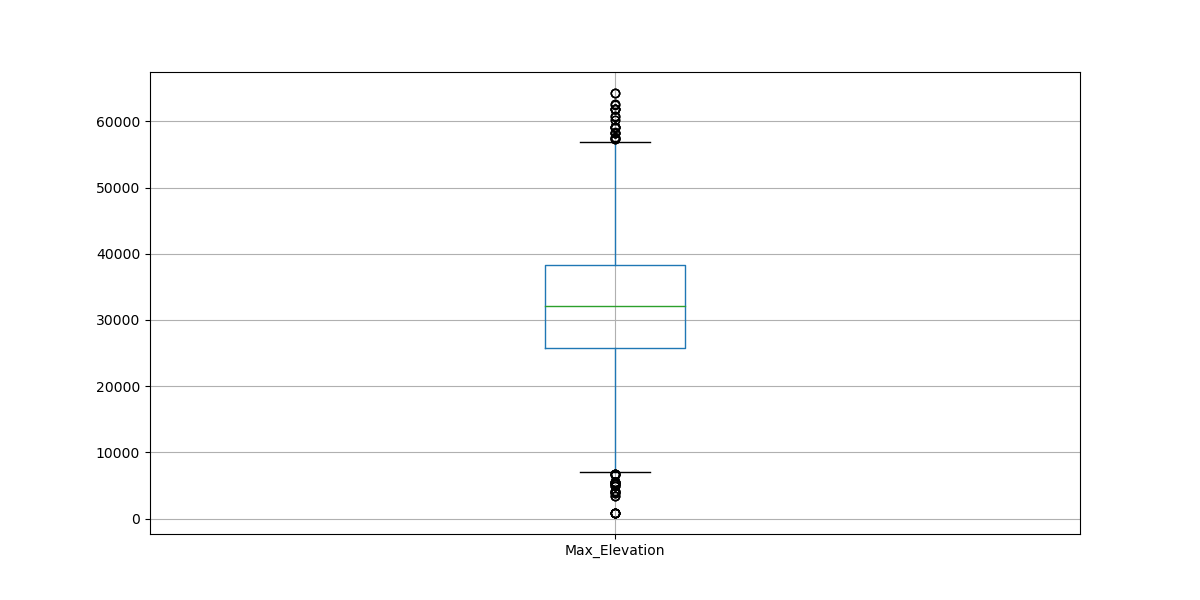
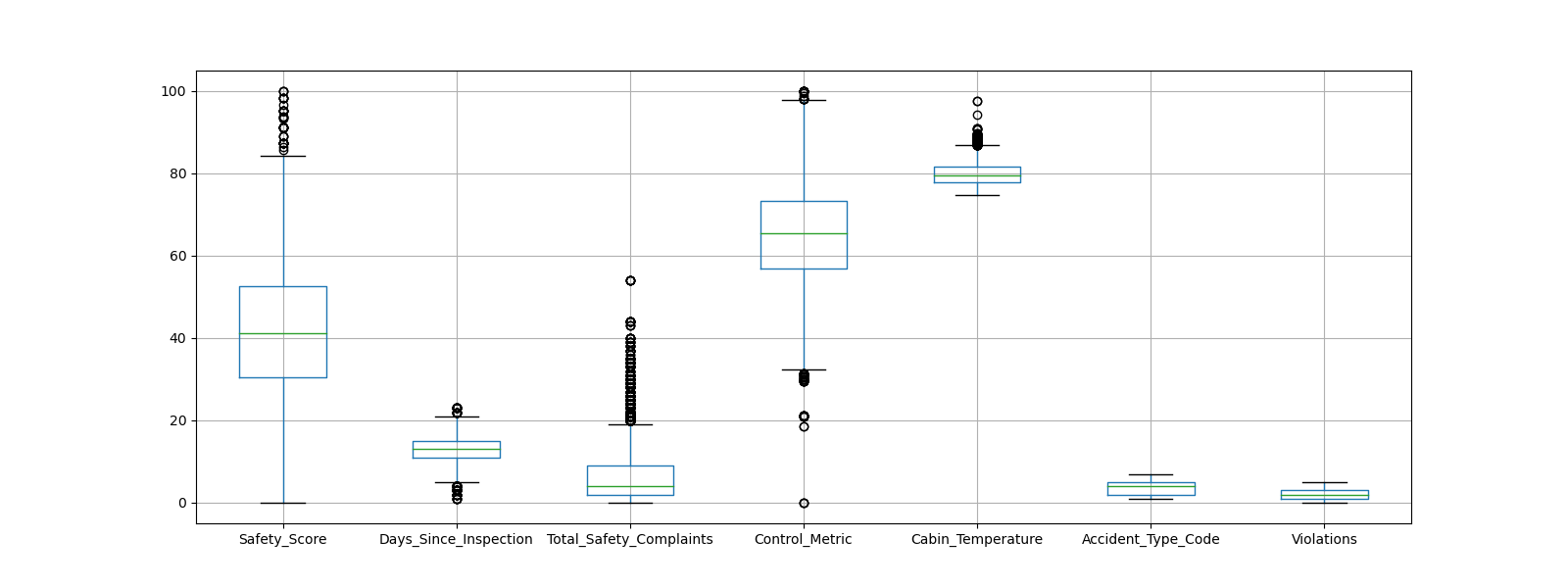
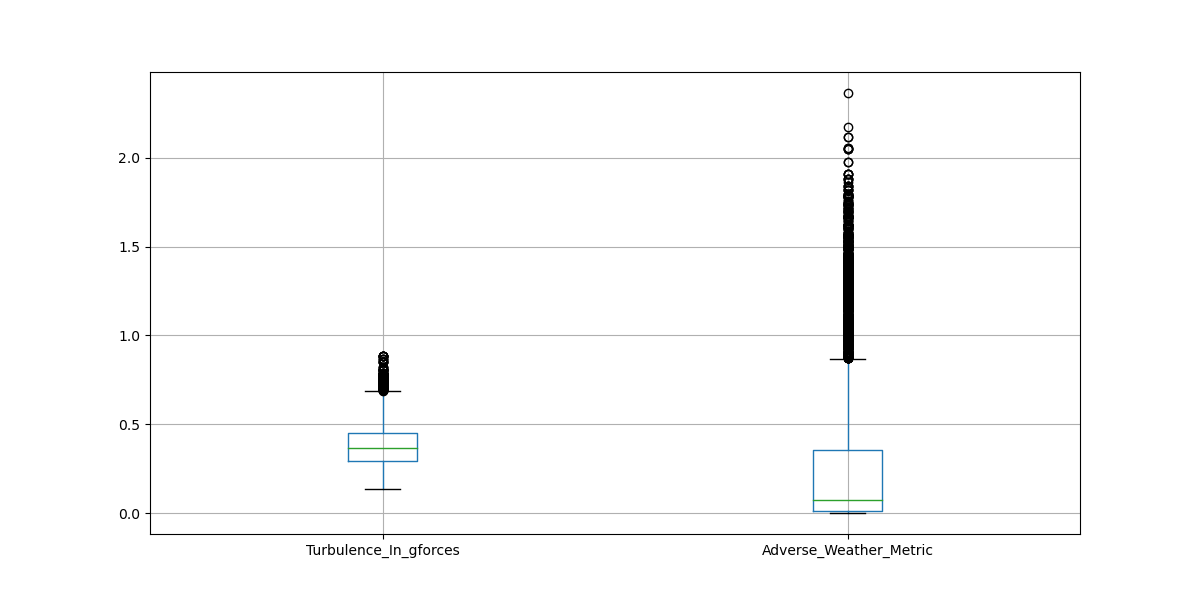


*Figure (5): Correlation Heatmap*



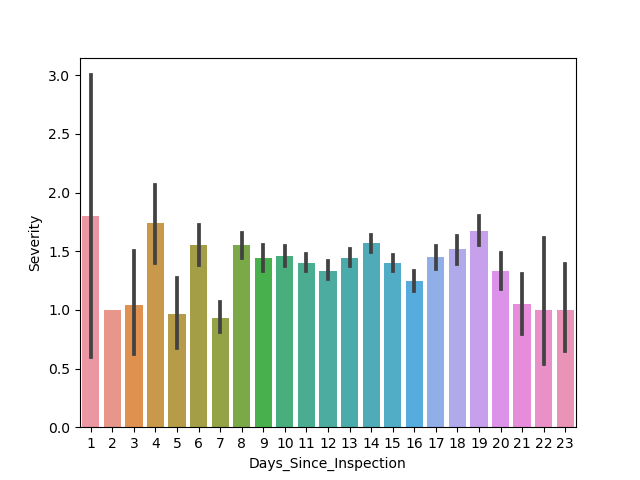
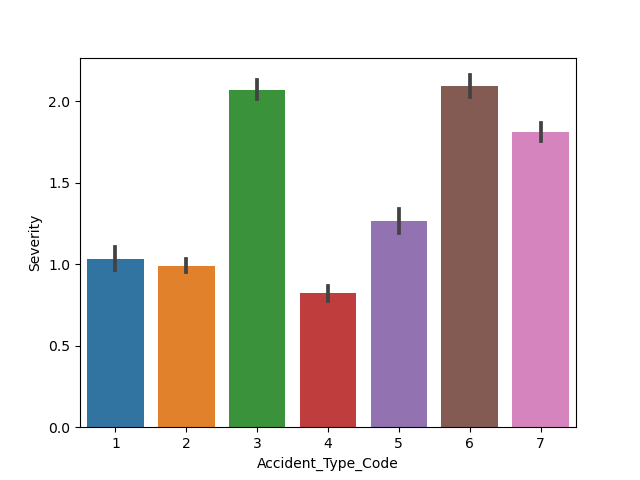
We attempted to create a graph showing the count of accident severities, which was our target variable, after understanding the correlation between the variables. This gave us an idea of the count of accident severities for all four types. The above graph shows that most accidents were extremely dangerous and fatal.

*Figure (6): Severity vs. Count graph*

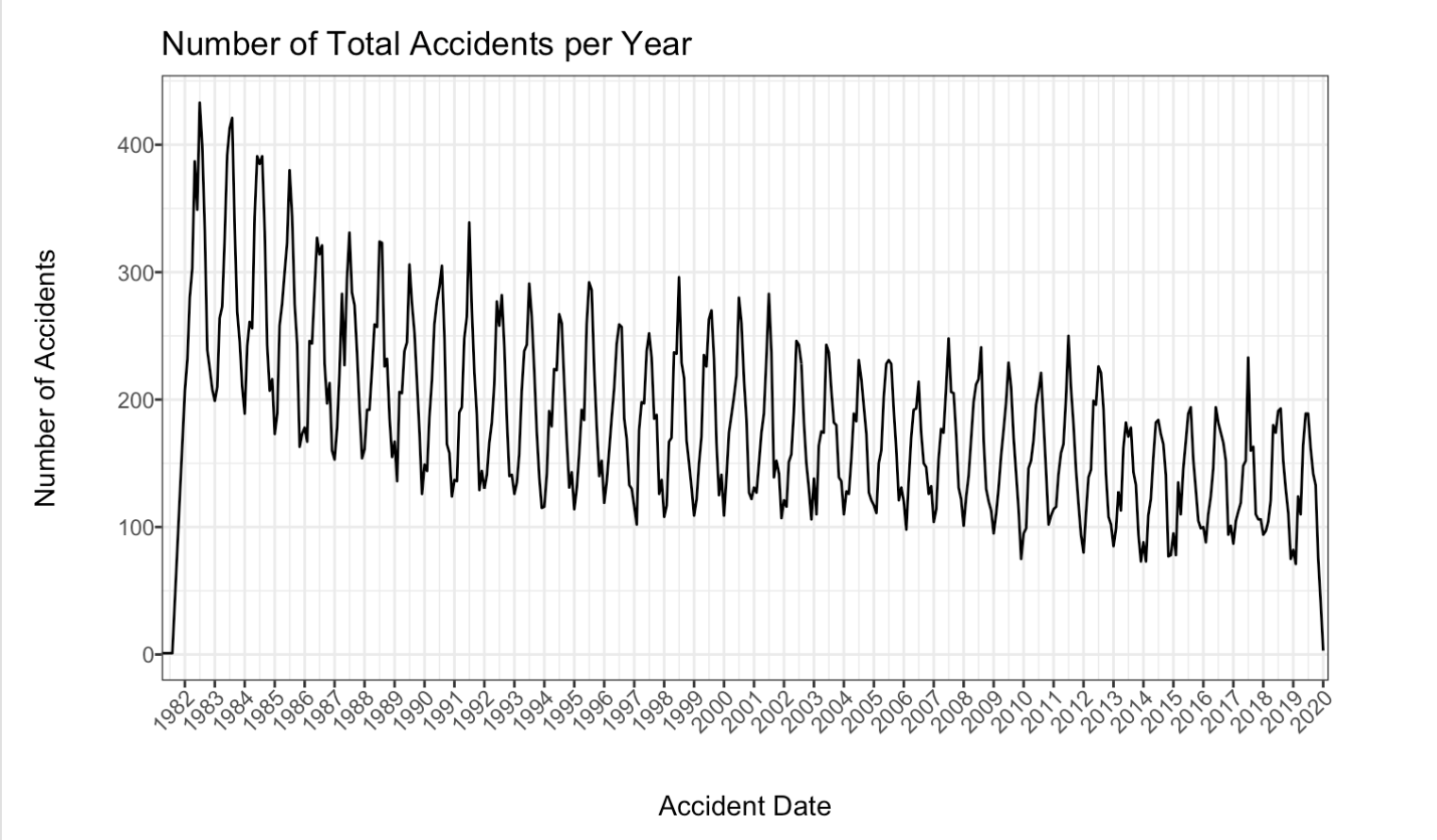
*Figure (7): Box plots of numerical variables*

We attempted to gain a deeper understanding of the numerical variables in our dataset using a box plot. First, since the "Accident Type Code" and "Severity" columns were categorical variables, we removed them from the box plot.

We can observe that the data has a large number of outliers. A z-score was used to normalize the numerical data. A z-score is a metric that quantifies how closely a value relates to the mean of a set of values. We eliminated all records having a z-score of 3 or less, or -3, respectively. The scaled columns were then blended into the original data frame after 493 records that were deemed outliers were eliminated. We used Label Encoding on the Severity target column.

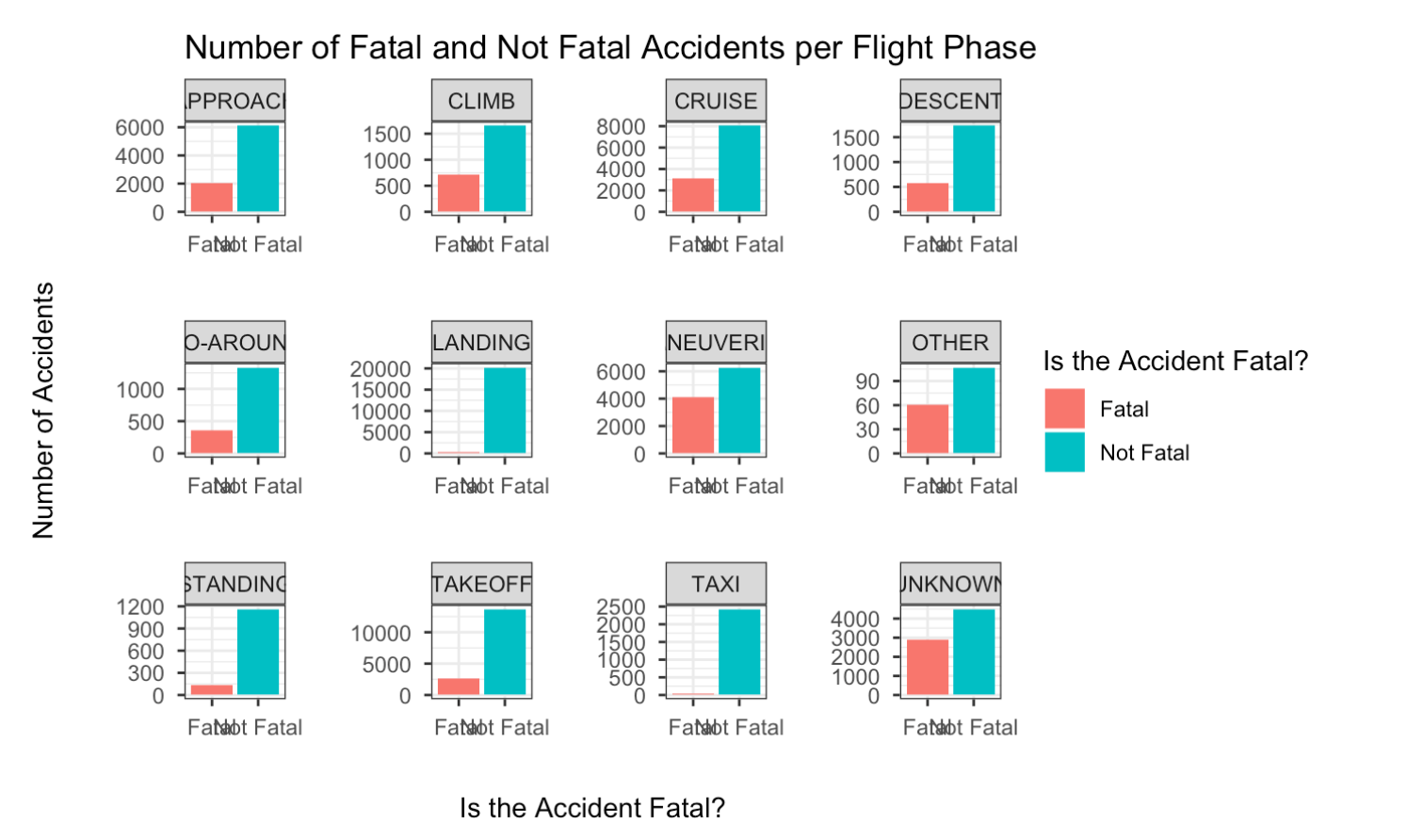
*Figure (8): Bar plots for days since inspection and accident type code vs. severity.*

We also plotted graphs for days since inspection and accident type code with severity to understand how the variables influence severity.



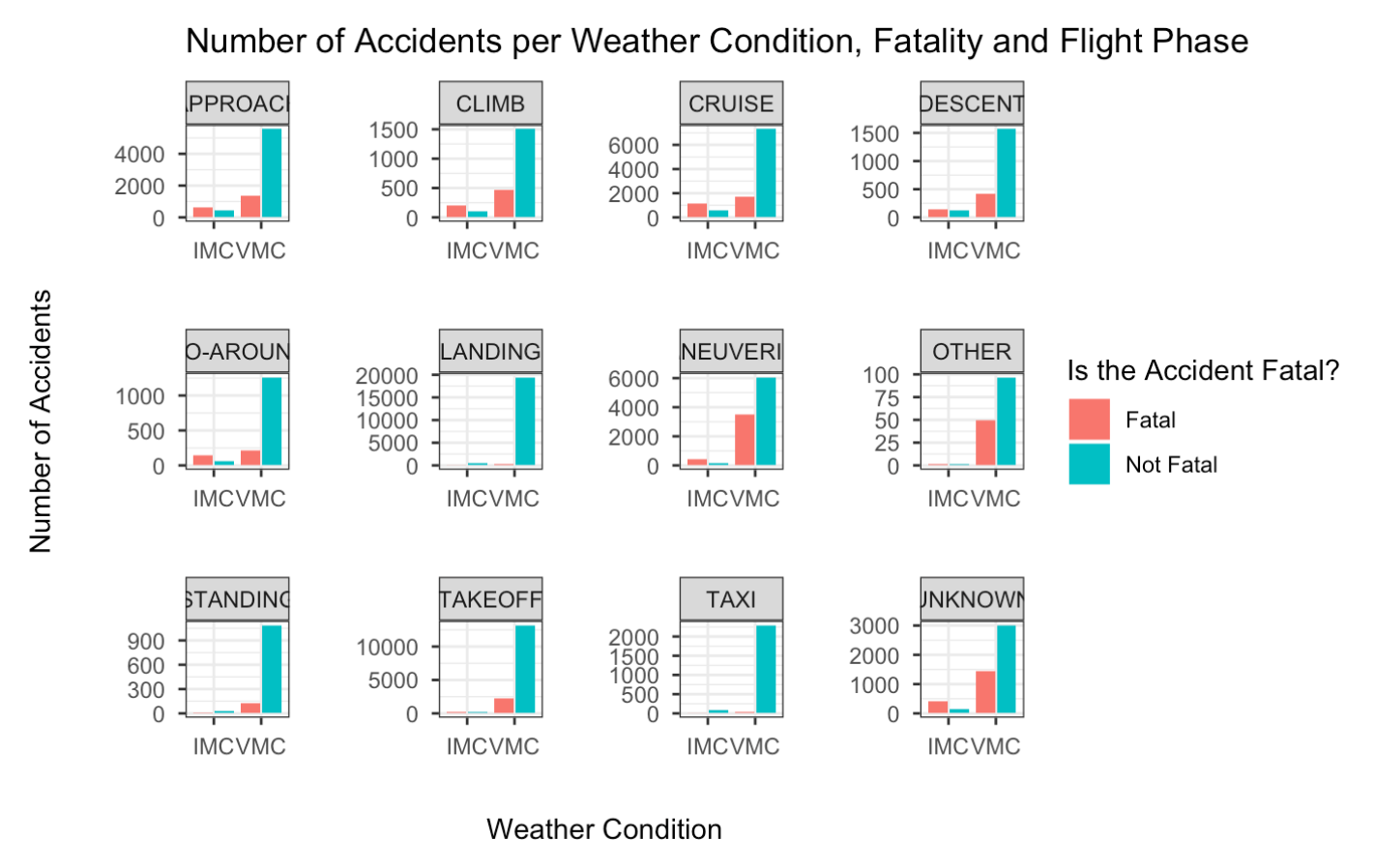
With the help of this graph, we can better comprehend the patterns in the number of accidents from 1982 to 2020. There was initially an increase in casualties, but that number has since decreased in recent years.

*Figure (9): Graph showing the number of accidents over time.*



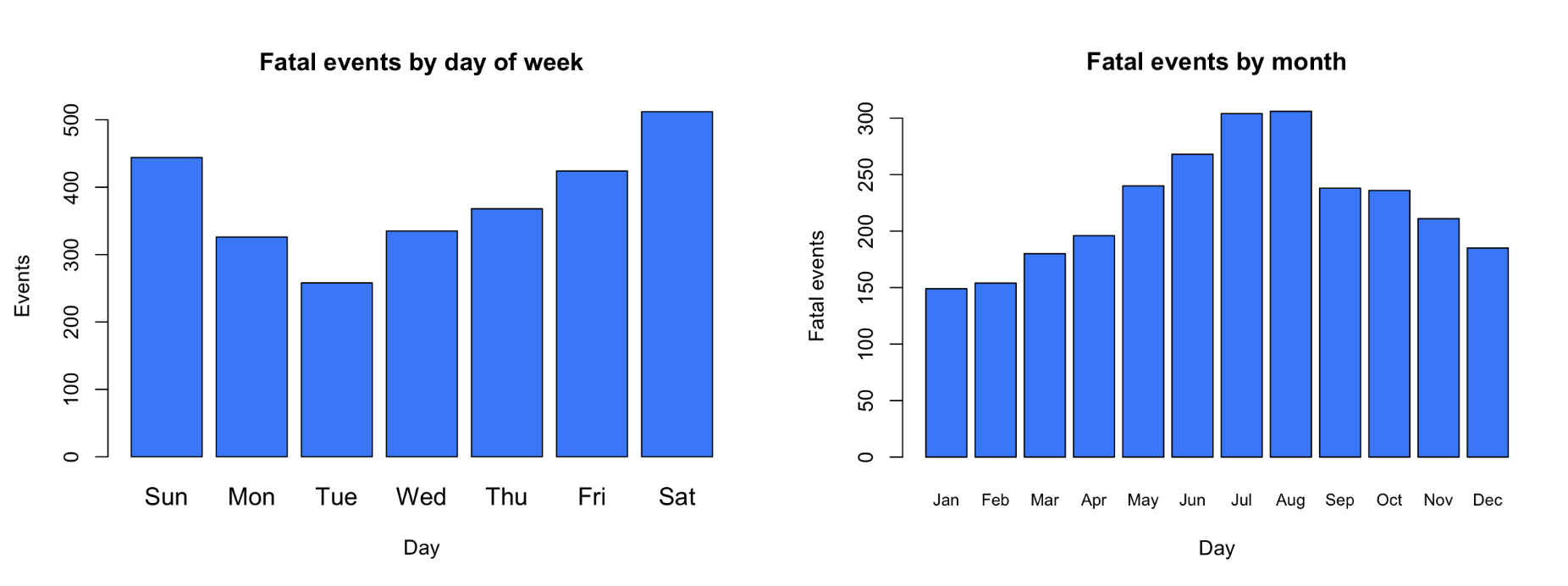
We could better grasp the accident fatalities during various flying phases because of these graphs. We can see that the landing and taxi incidents had no fatalities. There are fatal incidents in every other phase of flight.

*Figure (10): Graph showing the accident fatalities in different flight phases*

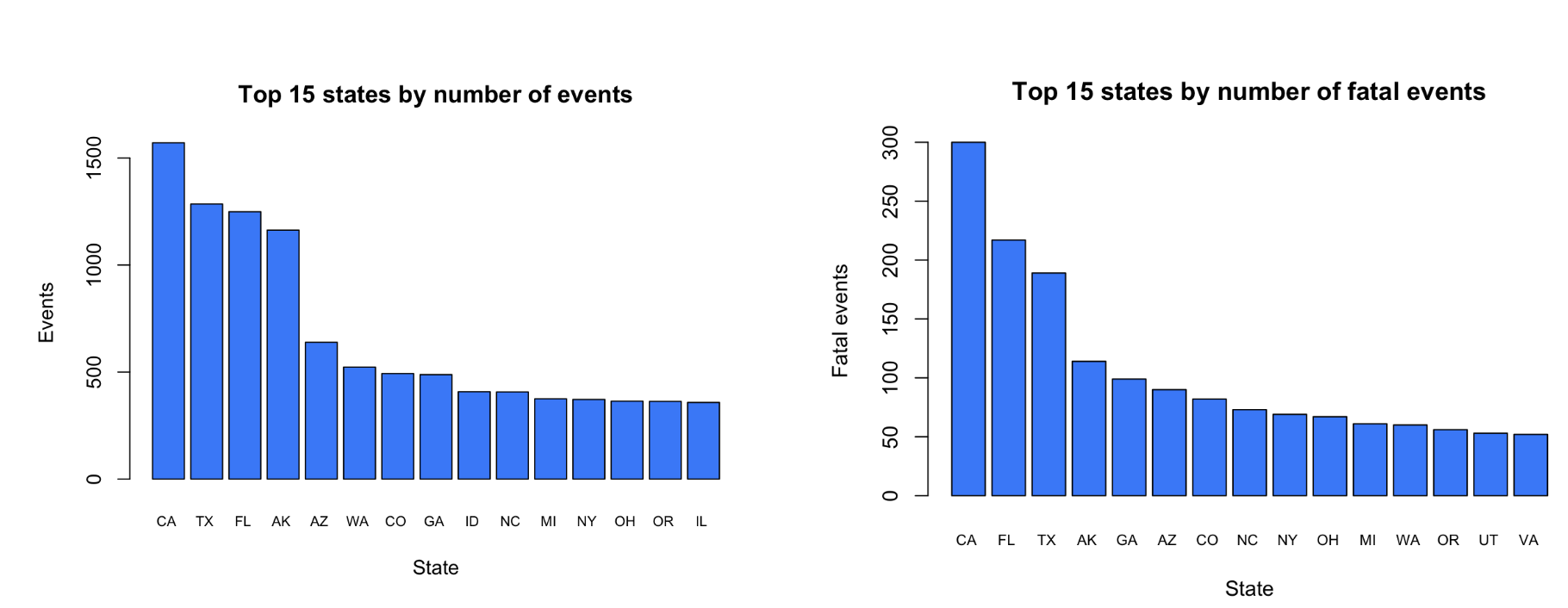


*Figure (11): Graph showing the accident fatalities in different flight phases depending on weather conditions.*

With the help of these graphs, we got an insight into accidents during different flight phases during different weather conditions. VMC and IMC are aviation terms used to describe meteorological conditions during flight. VMC stands for visual meteorological conditions, and IMC stands for instrument meteorological conditions. VMC generally refers to good weather. Accidents in IMC are usually not fatal, whereas more accidents are non-fatal in VMC.

*Figure (12): Graphs showing the accident fatalities by days and months.*

These graphs allow us to understand why there are more weekend accidents than weekdays. Since people often travel more on the weekends, there are more flights, which could be one of the causes of more accidents. Similar to the monthly analysis, we can see higher fatalities in the summer.



*Figure (13): Graph depicting the number of fatal accidents in each US state.*

The above graph helps us understand the fatalities state-wise. We can see that the maximum number of accidents happened in California.

**Machine Learning Models**

***Approach - 1: Predicting the air crash severity using only aircraft-related factors (First data subset).***

Label Encoding was performed on the Target Column. Basic Feature Engineering was performed in the interest of time and needed more domain expertise to create new features. Independent features and Target columns were split as X and Y. We have split the train data into train and validation data. We kept 90% data as training data and 10% as validation data.

**1. Decision Tree:** We tried to achieve maximum accuracy by tuning the default hyperparameters. We used Grid search to obtain the best combination of parameters. Model performance depends heavily on hyperparameters. Hyperparameters are customizable parameters that you can use to control the model training process. The parameter’s name serves as the key. The various values of the argument are the dictionary's values. By doing this, a table that represents different parameter values is created. A GridSearchCV object was created. The decision tree object, parameter values, and folds-per-fold are the inputs. Performance metrics for classification were employed. This is the standard system for scoring. We found the best values for the parameters using the attribute best estimator. We got the best accuracy for the model when the cross-validation was set to 5.

The best values for the parameters are as below:

**DecisionTreeClassifier(criterion='entropy', max\_depth=15, max\_features='auto', min\_samples\_leaf=5, min\_samples\_split=20,random\_state=1234).**

The output of the decision tree is as follows:

| Accuracy - 0.788 |
| --- |

|  | **Precision** | **Recall** | **F-1 Score** | **Support** |
| --- | --- | --- | --- | --- |
| Highly\_fatal\_And\_Damaging | 0.81 | 0.82 | 0.82 | 284 |
| Minor\_Damage\_And\_Injuries | 0.74 | 0.80 | 0.77 | 236 |
| Significant\_Damage\_And\_Fatalities | 0.82 | 0.79 | 0.80 | 160 |
| Significant\_Damage\_And\_Serious\_Injuries | 0.79 | 0.75 | 0.77 | 271 |

| accuracy |  |  | 0.79 | 951 |
| --- | --- | --- | --- | --- |
| macro avg | 0.79 | 0.79 | 0.79 | 951 |
| Weighted avg | 0.79 | 0.79 | 0.79 | 951 |

**Confusion Matrix**

| 233 | 24 | 6 | 21 |
| --- | --- | --- | --- |
| 17 | 189 | 9 | 21 |
| 12 | 11 | 126 | 11 |
| 24 | 32 | 13 | 202 |

**2. Random Forest:** We have defined a range for all the parameters for the Random Forest Classifier. Similar to the decision tree, we created a Grid search to obtain the best combination of parameters. With the help of n\_estimators, we determine the number of decision trees in the forest. In the best values, we got the n\_estimators as 5. Here also, after trying different values for cross-validation, we got the best values when cross-validation was set to 5.

The best values for the parameters of the Random Forest Classifier are as below:

**RandomForestClassifier(criterion='entropy', min\_samples\_split=7, n\_estimators=5, random\_state=1234)**

The output of the Random Forest is as follows:

| Accuracy - 0.895 |
| --- |

**Confusion Matrix**

| 260 | 9 | 7 | 8 |
| --- | --- | --- | --- |
| 5 | 222 | 3 | 6 |
| 5 | 14 | 135 | 6 |
| 20 | 14 | 2 | 235 |

|  | **Precision** | **Recall** | **F-1 Score** | **Support** |
| --- | --- | --- | --- | --- |
| Highly\_fatal\_And\_Damaging | 0.90 | 0.92 | 0.91 | 284 |
| Minor\_Damage\_And\_Injuries | 0.86 | 0.94 | 0.90 | 236 |
| Significant\_Damage\_And\_Fatalities | 0.92 | 0.84 | 0.88 | 160 |
| Significant\_Damage\_And\_Serious\_Injuries | 0.92 | 0.87 | 0.89 | 271 |

| accuracy |  |  | 0.90 | 951 |
| --- | --- | --- | --- | --- |
| macro avg | 0.90 | 0.89 | 0.89 | 951 |
| Weighted avg | 0.90 | 0.90 | 0.90 | 951 |

**3. Gradient Boosting**: We defined the parameters we want to pass through GridSearchCV to get the best parameters. We have n estimators, learning rate, subsample, and maximum depth parameters in the dictionary. We tried various values to get the utmost accuracy.

The best values for the parameters of the Gradient Boosting algorithm are as below:

**Best: 0.945333 using {'learning\_rate': 0.1, 'max\_depth': 9, 'n\_estimators': 500, 'subsample': 0.7}**

The output of the Random Forest is as follows:

| Accuracy - 0.945 |
| --- |

|  | **Precision** | **Recall** | **F-1 Score** | **Support** |
| --- | --- | --- | --- | --- |
| Highly\_fatal\_And\_Damaging | 0.94 | 0.95 | 0.95 | 284 |
| Minor\_Damage\_And\_Injuries | 0.95 | 0.96 | 0.95 | 236 |
| Significant\_Damage\_And\_Fatalities | 0.96 | 0.94 | 0.95 | 160 |
| Significant\_Damage\_And\_Serious\_Injuries | 0.96 | 0.95 | 0.96 | 271 |

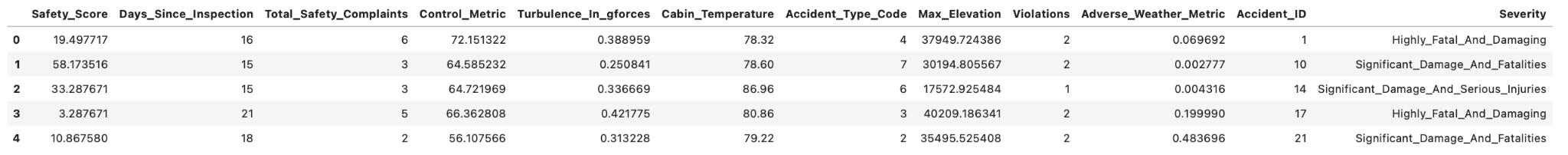
| accuracy |  |  | 0.95 | 951 |
| --- | --- | --- | --- | --- |
| macro avg | 0.95 | 0.95 | 0.95 | 951 |
| Weighted avg | 0.95 | 0.95 | 0.95 | 951 |

**4. Extreme Gradient Boosting:** Extreme Gradient Boosting is a specific implementation of the Gradient Boosting method which uses more accurate approximations to find the best tree model. Both Extreme Gradient Boosting and Gradient Boosting follow the principle of gradient boosting. There are, however, differences in modeling details. Specifically, Extreme Gradient Boosting uses a more regularized model formalization to control over-fitting, which gives it better performance. For Extreme Gradient Boosting, after trying multiple values for Learning rate, max depth, and cross-validation, we achieved an accuracy of 95.48% with an F1 score of 0.95.

The best parameters of Gridsearch obtained for Extreme Gradient Boosting are as follows:

**{'XGB\_\_max\_depth': 10, 'XGB\_\_learning\_rate': 0.2, 'XGB\_\_gamma': 0.1}**

Extreme Gradient Boosting was considered as the final model, and RandomSearchCV was used for hyper-parameterization. After training four different models, we used the Extreme Gradient Boosting model for our predictions on the Test dataset. First, with the help of inverse label encoding, we converted the numeric form back into labels and then predicted the severity of the airplane accidents on the test data set.

****

*Figure (14): Output prediction of Severity on the test data*

***Approach - 2: Predicting the air crash severity using only personnel factors​***

All the following machine learning models are built using Orange Data Miner - hence the following preprocessing steps are followed in addition to ones previously mentioned: *remove instances with unknown target values, Continuize categorical variables (with one-hot-encoding), remove empty columns, impute missing values with mean values.*

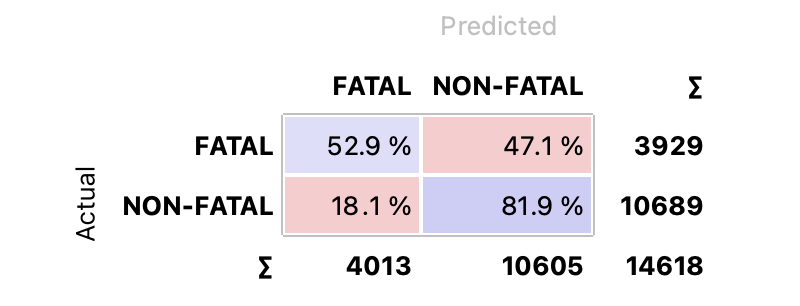
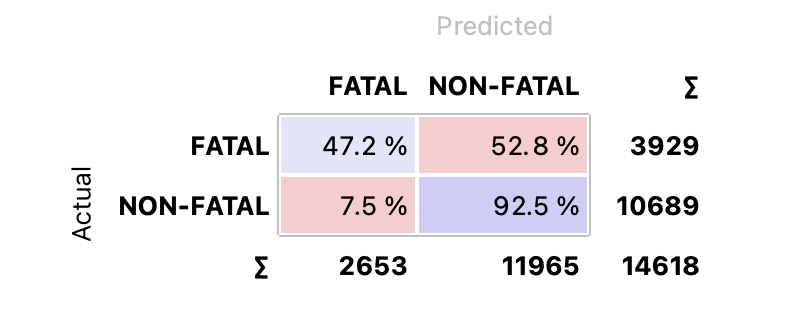
The final variables considered in this approach are Pilot hours (*Last 24 hours, 30 days, 90 days, Total, as Pilot-In-Command, on Single Engine, on Multi-Engine, Day & Night*), Age, Sex, Medical Certificate Validity, Pilot Profession, is Pilot Flying, and is Professional Pilot.

We achieved 98% accuracy by building an initial decision tree, as described earlier. However, we had to remove the variables: crew\_tox\_perf and pilot\_inj\_level, which erroneously influenced the final output.

We have achieved the best accuracy scores with the following parameters:

* Minimum Number of Instances - **3** (Increasing it did not show a drastic difference)
* Do not split subsets smaller than - **5** (Increasing it decreased the accuracy score)
* Limit maximal tree depth to - **50** (Increasing or decreasing it has no influence

| **Model** | **AUC** | **CA** | **F1** | **Precision** | **Recall** |
| --- | --- | --- | --- | --- | --- |
| Random Forest | 0.788 | 0.804 | 0.790 | 0.792 | 0.804 |
| Tree | 0.689 | 0.741 | 0.742 | 0.743 | 0.741 |



*Figure (15): Confusion matrix for Random Forest model (Left) and Decision Tree Model (Right)*

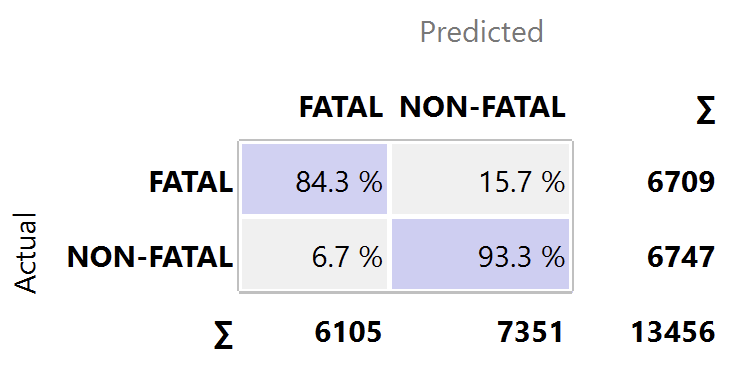
***Approach - 3: Predicting the air crash severity using a combination of aircraft, personnel, and meteorological factors***

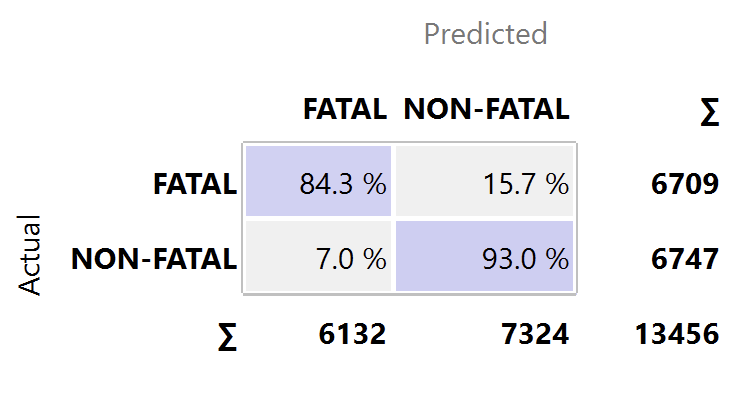
Step - 1: Hypertuning all the parameters for the five Machine Learning models discussed earlier and performing a five-fold cross-validation test, Logistic regression followed by Gradient Boosting produced the best accuracy (F-1) scores, as shown in the table below:

| **Model** | **AUC** | **CA** | **F1** | **Precision** | **Recall** |
| --- | --- | --- | --- | --- | --- |
| **Tree** | 0.811 | 0.768 | 0.768 | 0.769 | 0.768 |
| **Stack** | 0.947 | 0.886 | 0.889 | 0.888 | 0.889 |
| **Random Forest** | 0.934 | 0.875 | 0.874 | 0.876 | 0.875 |
| **Neural Network** | 0.921 | 0.853 | 0.853 | 0.853 | 0.853 |
| **Logistic Regression** | 0.943 | 0.887 | 0.889 | 0.890 | 0.889 |
| **Gradient Boosting** | 0.945 | 0.888 | 0.888 | 0.890 | 0.888 |

A sample of parametric tuning carried out with the two best models and the concerning F-1 scores reported are as follows:

| **Parametric Tuning** | **Logistic Regression (F-1 Score)** | **Gradient Boosting (F-1 Score)** |
| --- | --- | --- |
| Regularization Type - Lasso (or) Ridge | 0.889 |  |
| Regularization - **Lasso**, C - 1 Score (**1**, 0.6, 10) | 0.889, 0.885, 0.886 |  |
| Gradient Boosting Methods: Scikit-Learn, xgboost, Random Forest xgboost, **catboost** |  | 0.875, 0.883, 0.821, 0.888 |
| Method: **catboost**, Learning rate: 0.1, **0.3**, 0.5 |  | 0.887, 0.888, 0.885 |

Confusion matrices for the Logistic Regression and Gradient Boosting models are:



*Figure (16): Confusion matrix for Decision Tree Model (Left) and Gradient Boosting (Right)*

Step - 2: The models are then validated using a 20% test split by Data Sampler. The resulting F-1 scores reported are as follows:

| **Model** | **AUC** | **CA** | **F1** | **Precision** | **Recall** |
| --- | --- | --- | --- | --- | --- |
| **Tree** | 0.813 | 0.770 | 0.770 | 0.771 | 0.770 |
| **Stack** | 0.951 | 0.890 | 0.902 | 0.896 | 0.890 |
| **Random Forest** | 0.941 | 0.875 | 0.875 | 0.875 | 0.875 |
| **Neural Network** | 0.928 | 0.859 | 0.859 | 0.859 | 0.859 |
| **Logistic Regression** | 0.947 | 0.890 | 0.890 | 0.892 | 0.890 |
| **Gradient Boosting** | 0.941 | 0.876 | 0.876 | 0.877 | 0.876 |

Step - 3: The models are then validated using the test split from the original dataset (before imputing and oversampling). The resulting F-1 scores reported are as follows:

| **Model** | **AUC** | **CA** | **F1** | **Precision** | **Recall** |
| --- | --- | --- | --- | --- | --- |
| **Stack** | 0.836 | 0.828 | 0.820 | 0.820 | 0.828 |
| **Logistic Regression** | 0.811 | 0.829 | 0.822 | 0.822 | 0.829 |
| **Gradient Boosting** | 0.833 | 0.826 | 0.815 | 0.818 | 0.826 |

Utilizing the Rank widget, variables were score according to their correlation with the discrete target variable, based on applicable internal scorers and connected external models that supports scoring, such as logistic regression, random forest, SGD, etc.

| **Rank** | **Variable** | **Gain ratio** | **Gini** |
| --- | --- | --- | --- |
| **1** | **ev\_nr\_apt\_loc** | **0.146097** | **0.093735** |
| **2** | **L24H** | **0.076186** | **0.042901** |
| **3** | **L30D** | **0.061312** | **nan** |
| **4** | **PIC** | **0.055912** | **0.101144** |
| **5** | **NGHT** | **0.055513** | **0.089404** |
| 6 | L90D | 0.052919 | 0.05681 |
| 7 | wx\_cond\_basic | 0.050805 | 0.01601 |
| 8 | SENG | 0.050447 | 0.099868 |
| 9 | wind\_vel\_ind | 0.041037 | 0.009211 |
| 10 | INST | 0.039095 | 0.024991 |

## Discussion

To reiterate our entire approach in solving the business problem of anticipating and classifying the severity of any airplane accident based on past incidents, and finding out the relevant factors influencing the severity, we:

* First conducted text mining on accident narrative to understand the themes/topics around the aircraft crashes and found that three categories of factors played a crucial role: aircraft-related, personnel-related, and meteorological (weather)-related.
* Then performed the data extraction and preprocessing steps along with a literature review to arrive at two data subsets: a subset with only aircraft-related factors and a subset that contained all three factors with ~45 Variables.
* Given the categorical nature of our target variable, along with a majority of our independent variables, we set out to utilize five models: Decision Trees, Random Forest, Gradient Boosting, Logistic Regression, and Neural Networks. We have then devised the data mining (Machine Learning) process into a three-step approach:
  + Approach - 1: Predicting the air crash severity using only aircraft factors
  + Approach - 2: Predicting the air crash severity using only personnel factors
  + Approach - 3: Predicting the air crash severity using a combination of aircraft, personnel, and meteorological factors
* Built initial tree models to diagnose and fix erroneous variables and class imbalance (using imblearn and SMOTE method of Oversampling).

From our machine learning approaches, building and tuning machine learning models, we have seen that including only aircraft factors produced a 95% accuracy in predicting the severity of air crashes. However, when only the personnel-related factors were considered, the accuracy went down to 78% (69% before hyper-parameter tuning), which meant human factors played an important role in predicting the severity of air crashes.

In our final approach of including all the factors, the models were able to predict the severity of air crashes with a staggering ~89% accuracy. However, there is a small catch. When we reran the experiment to test the models on a test set based on original data (before imputing and class balancing), the accuracy in predicting the severity of air crash reached 82% - which given the real-world data, is reasonably good.

We have identified that the following variables/factors played a significant role in predicting the severity of air crashes:

| **Variable Name** | **Description** |
| --- | --- |
| ev\_nr\_apt\_loc | Indicate where the event took place (on or off an airport/airstrip) |
| L24H | Hours last 24 hours |
| L30D | Hours last 30 days |
| PIC | Pilot-In-Command |
| NGHT | Total hours at night |
| L90D | Hours last 90 days |
| wx\_cond\_basic | The basic weather conditions at the time of the event. |
| SENG | Total single-engine hours |
| wind\_vel\_ind | The local reported wind speed at the time of the event.  Reported wind speed is determined using the average speed over a 2-minute period. |
| INST | Total hours as an “Instructor” |

From the above table and aforementioned results, it becomes apparent that pilot (or) personnel-related factors, which could be controlled for, played a significant role in predicting the severity of air crashes. This means that by focusing more on pilot experience, training, medical certifications, etc., an air carrier can reduce the accident rate and fatality in the case of one. It also helps to prepare better/manage other externalities.

## Conclusion

When we first started, we had to extract data from a real-world database consisting of 12 data sheets containing information about each individual factor (aircraft, weather, events, flight crew, etc.) We had to deal with a cumulative of ~150+ variables and disproportionate data pointers. Even though we followed a reasonable approach in selecting our variables, we had to exclude many variables and factors that may play an important role in predicting the severity. By gaining domain expertise and experimenting with text mining to derive variables/words based on topics, we can create new features with which the model can interpret better.

Including the incident data, along with incorporating more variables representing the additional factors and utilizing data across different geographical locations, would be something we look forward to probing into in the future work.

## Appendix

| **1. Data Cleaning & Preprocessing** | Sai, Huzaif, Samar |
| --- | --- |
| **2. Descriptive Statistics** | Shabib, Samar |
| **3. Machine Learning Techniques (Clustering & Classification)** | Huzaif, Shabib, Sai |
| **4. Machine Learning Techniques (Regression)** | Sai, Huzaif |
| **5. Final Report** | Sai, Huzaif, Shabib, Samar |
| **6. Final Presentation** | Sai, Huzaif, Shabib, Samar |

List of 40 Variables included in the analysis:

| Mid-air | Basic weather conditions | Sightseeing flight | Med certificate validity |
| --- | --- | --- | --- |
| Ground collision | Flight plan type | Air-medical flight | Professional pilot |
| Airport location to crash | Homebuilt | Airspace | Highest certificate |
| Atmospheric lighting | Fixed-retractable gear | Crew position code | Total flight hours |
| Wind gusts indicated | Purpose of Flight | Age of the Pilot | Total flight hours |
| **Accident Injury level** | Second pilot on board | Sex of the Pilot | Hours last 90-days |
| Hours last 30-days | Total hours single-engine | Defining events | Report narrative |
| Hours last 24-hours | Total hours at night | Occurrences | Factual narrative |
| Total hours make | Engine type | Causes | Cause narrative |
| Total hours multi-engine | Multi-engine aircraft | Factors | Incident narrative |