Human and Wildlife Coexistence in Canadian National Parks:

Discovering trends in reported incidents and identifying target areas for promoting health and safety of humans and wildlife and mitigating negative incidents in 35 Canadian National Parks

Emma Billard, Student Number: 501114915
Toronto Metropolitan University
CIND820 XJH – Big Data Analytics Project – W2023
Professor Tamer Abdou, PhD
February 20, 2023

Ryerson University

Table of Contents

Abstract **4**

Data Description **6**

Figure 1: Data Types **6**
Figure 2: Numerical Data Descriptive statistics **7**
Figures 3 & 4: Categorical Data Descriptive Statistics **8**

Methodology and Initial Observations **9**

Figure 5: Graph of overall methodology **9**

Data Collection 10

Data Processing 11

Exploratory Data Analysis 12

Figure 6: Total Number of Incidents by National Park and by Incident Type **13**
Figure 7: Total Number of Incidents per Year, by Incident Type **14**
Figure 8: Total Number of Incidents per Month, by Incident Type **14**

Modeling 15

Splitting Data 16
Dealing with Missing Values 17
Encoding the Data 18
Encoding Numeric to Categorical: **18**
Encoding Categorical to Numeric: **19**
Feature Selection 19
Figure 9: SelectKBest: Selected Features across 5 folds **21**
Figure 10: Forward Elimination: Selected Features across 5 folds. **21**
Figure 11: Forward Elimination Accuracy Scores per Fold. **22**
Figure 12: Forward Elimination vs. SelectKBest in Multinomial Logistic Regression model **23**
Figure 13: Forward Elimination vs. SelectKBest vs. intrinsic Entropy in Decision Tree Classifier model. **24**
Figure 14: Forward Elimination vs. intrinsic Entropy in Random Forest Classifier model, Plot 1 ..... **25**
Figure 15: Forward Elimination vs. intrinsic Entropy in Random Forest Classifier model, Plot 2 ..... **25**
Figure 16: Table of Selected Features using Forward Elimination and Random Forest's intrinsic Gini Index **26**

Dealing with Imbalanced Data 26
Models 27
Figure 17: Multinomial Logistic Regression, Plot 1: Performance Metrics Across 5 Folds 28
Figure 18: Multinomial Logistic Regression, Plot 2: Stability in Performance Metrics Across 5 Folds 29
Figure 19: Decision Tree Classifier, Plot 1: Performance Metrics Across 5 Folds 30
Figure 20: Decision Tree Classifier, Plot 2: Stability in Performance Metrics Across 5 Folds 30
Figure 21: Decision Tree Classifier, Plot 1: Performance Metrics Across 5 Folds 31
Figure 22: Decision Tree Classifier, Plot 2: Stability in Performance Metrics Across 5 Folds 32
Evaluation 32
Figure 23: Model Comparison, Plot 1: Comparing Average Metrics for each Model 33
Figure 24: Model Comparison, Plot 2: Comparing Overall Metrics for each Model 34
Figure 25: Model Comparison, Plot 3: Comparing Time Used in seconds for each Model 35
Figure 26: Model Comparison, Plot 4: Comparing Memory Used in seconds for each Model 36

Findings and Results: Answering the Research Questions 37
Research Question 1: 37
Figure 27: Confusion Matrix from Fold 5 of Random Forest Model: 38
Research Question #2: 39
Figure 28: Augmented Dickey-Fuller Test of Stationarity, p-value results 40
Research Question #3: 40
Figure 29: Results of Friedman Test for Significance on All Three Models..... 41
Figure 30: Results of Wilcoxon Test for Significance on Top Two Models 42

Shortcomings of the Work 43

Concluding Remarks 44

References 47

Abstract

Time spent in nature is wondrous. Whether you're drawn to witness epic mountainscapes, giant old growth forests, or wildlife in their natural habitat, there is always wonder to be found in the wild. But we must not forget that being able to witness our beautiful natural world is a privilege and a gift. We must care for our natural planet so that it continues to thrive year after year.

In Canada, our National Parks are maintained by the Parks Canada Agency. According to Government of Canada (2022), the Agency's mandate includes acting as guardians of the national parks and protecting our natural places to ensure they remain healthy and whole. Under this mandate, Gummer & Nicholl (2022) indicate that between 2010 – 2021, Parks Canada compiled four datasets of incidents of human-wildlife coexistence in 35 National Parks for the evaluation of trends to inform Parks Canada policies and to ensure safe visitor experiences while conserving wildlife and integrity of our ecosystems.

The four databases contain 70,000+ records of Incident Types, Animals Involved, Human Activities, and Responses related to reported human-wildlife coexistence incidents. I combined these three datasets using the unique Incident Number's associated with each record. Of the three datasets I use, I use the English version. I also reference the bilingual "Data Dictionary" and "Header Description" datasets provided with the data in the Canada's Open Government Portal.

For my data analytics project, I use all four datasets for the exploratory analysis phase. Then for the modeling phase, I remove the "Responses" data and use only the other three datasets related to Incidents Types, Animals Involved, and Human Activities (respectively). The reason for removing the Responses data from the modeling phase is because the Responses are inherently dependent on Incident Type and only independent variables should be used when predicting a target variable. I also do not use the several other datasets included in the same

Open Record and which contain compiled summaries of the incidents as I conduct my own pattern mining and summarizing.

The main problem I seek to address is to determine what features are most correlated with the various Incident Types with the goal of identifying target areas for promoting health and safety of humans and wildlife and for mitigating negative incidents in our National Parks.

I used the theme of predictive analysis, specifically pattern mining and causality. The goal is to find patterns and correlations that will hopefully allow me to make predictions on when, where, and why various incidents occur. This information will allow me to develop recommendations for park visitors and park employees to help ensure the health and safety of both humans and wildlife. I will tackle this by focusing on the following research questions:

1. What patterns can be visually observed in location and time of year for each of the following variables: human activities, animals involved, cause, and incident type. How do these patterns differ and across National Parks and over time?
2. Are the statistical properties of the data stationary over time?
3. What variables are most correlated with the occurrence of each incident type and what prediction model performs best in predicting “Incident Type”.

A brief summary of the results of my analysis for each question are laid out below:

1. Most incidents are occurring in the Banff and Jasper National Parks of Canada. The most prevalent Incident Types are “Human Wildlife Interaction” and “Rescued/Recovered/Found Wildlife”. Incidents have tended to increase over the years, and increase in frequency during the warmer months of May – October.
2. The statistical properties of the data are stationary over time.

3. The features most correlated with Incident Type (looking at the Features used in both the Random Forest and Decision Tree models) are: Total Staff Hours, Incident Month, Species Common Name, Total Staff Involved, Total Staff Hours, Protected Heritage Area, Field Unit, Latitude Public, and Activity Type_Railway. The Decision Tree Model and Random Forest models perform statistically the same in predicting Incident Type; however the Random Forest model is more efficient in that it requires less time and computer memory.

My Github repository including all the code for this project can be found here: https://github.com/ebillard06/human-wildlife-coexistence-data-analysis.

Data Description

The combined dataset contains 73658 rows and 170 columns and has a combination of numeric (float64) and categorical (object) data. The target variable we're looking at represents a multi-class classification problem as it consists of 9classes). I've included some figures here that depict summary statistics and data types for the dataset I am using. I also have a complete Exploratory Data Analysis Report available (along with the Complete_HWC_Data.csv data) in my GitHub repository which can be accessed here: https://github.com/ebillard06/human-wildlife-coexistence-data-analysis.

Figure 1: Data Types

The data types for attributes in columns 0-19 are listed in Figure 1. The attributes in columns 20-170 are one-hot encoded columns generated from the categorical data from the "Activity Type" and "Response Type" attributes from the datasets before they were merged. All data types for columns 20-170 are float64.

```
UniqueID                          object
Incident Number                   object
Incident Date                     object
Field Unit                        object
Protected Heritage Area           object
Incident Type                     object
Latitude Public                  float64
Longitude Public                 float64
Within Park                       object
Total Staff Involved             float64
Total Staff Hours                float64
Species Common Name               object
Sum of Number of Animals         float64
Animal Health Status              object
Cause of Animal Health Status     object
Animal Behaviour                  object
Reason for Animal Behaviour       object
Animal Attractant                 object
Deterrents Used                   object
Animal Response to Deterrents     object
dtype: object
```

A note about the “Field Unit” and “Protected Heritage Area” features. While it often appears as though the “Field Unit” and “Protected Heritage Area” (i.e. Canadian National Park plot) variables overlap, both are interesting to look at. The header description csv document included with the data describe “Protected Heritage Area” as directly reflecting the 35 Canadian National Parks and “Field Unit” as reflecting the “name of the administrative unit of Parks Canada Agency that is responsible for management of the incident based on its location”.

Figure 2: Numerical Data Descriptive statistics

Contains descriptive summary statistics for the attributes in columns 0-19 of the numerical data “float64” type.

```
Complete_HWC_Data[Complete_HWC_Data.columns[0:20]].describe()
```

	Latitude Public	Longitude Public	Total Staff Involved	Total Staff Hours	Sum of Number of Animals
count	73624.000000	73624.000000	73658.000000	73658.000000	73655.000000
mean	51.484498	-114.770930	1.477789	2.331069	2.728776
std	1.900213	9.784865	1.055412	14.361819	14.389458
min	41.902015	-140.297738	0.000000	0.000000	0.000000
25%	51.168223	-118.063343	1.000000	0.500000	1.000000
50%	51.286676	-116.165634	1.000000	1.000000	1.000000
75%	52.872448	-115.551471	2.000000	2.000000	1.000000
max	73.998028	-52.637169	32.000000	2400.000000	2000.000000

Figures 3 & 4: Categorical Data Descriptive Statistics

Combined, these figures contain descriptive statistics for the attributes in columns 0-19 of the categorical data “object” type. These figures are split in two to help with readability.

```
Complete_HWC_Data[Complete_HWC_Data.columns[11:20]].describe(include='object')
```

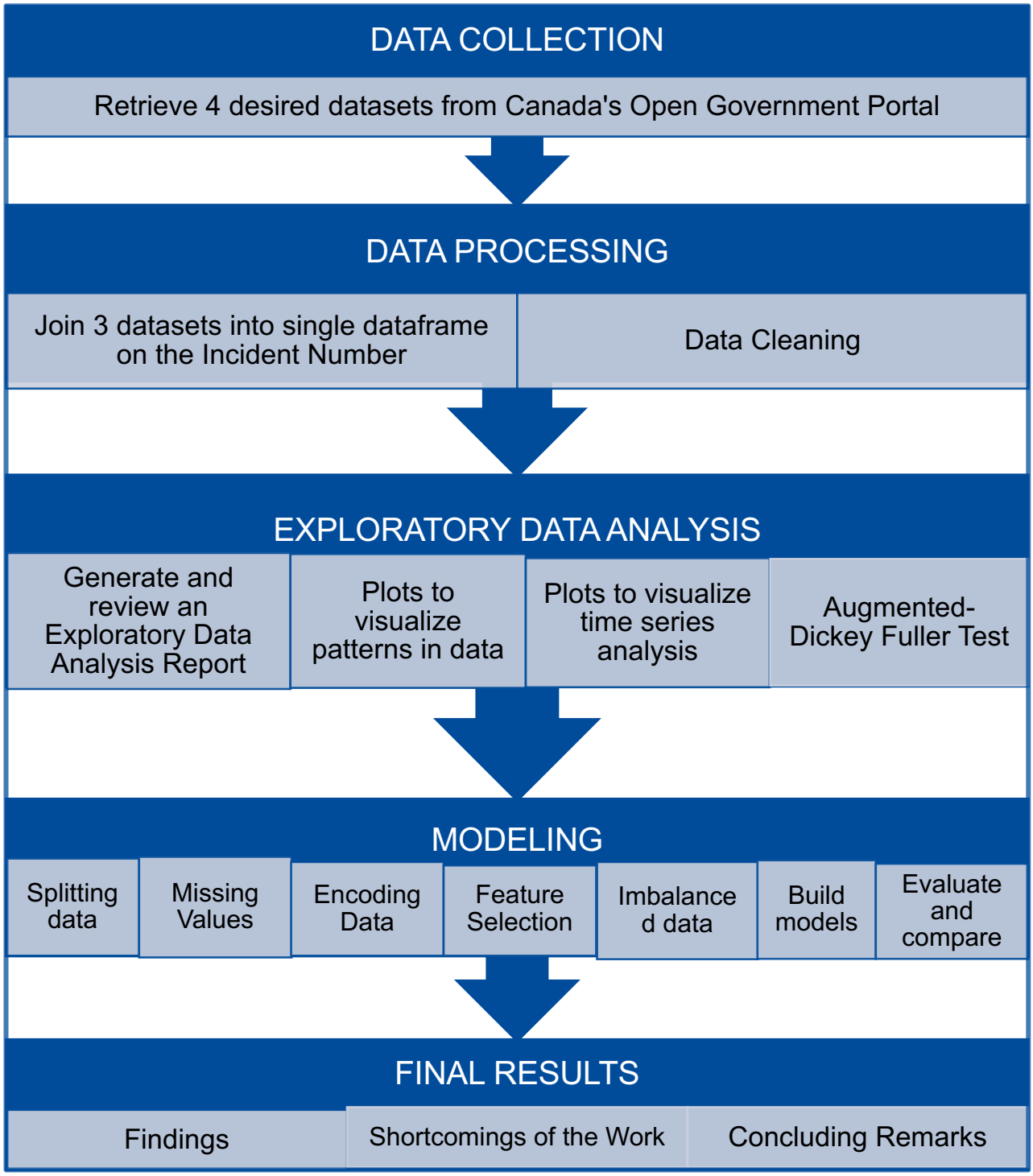
	Species Common Name	Animal Health Status	Cause of Animal Health Status	Animal Behaviour	Reason for Animal Behaviour	Animal Attractant	Deterrents Used	Animal Response to Deterrents
count	73655	41485	13197	45675	26138	24820	19544	10502
unique	322	9	17	23	17	21	26	10
top	Black Bear	Healthy	Collision	Presence - Wildlife Exclusion Zones	Habituation	Vegetation (natural)	Noise - Voice	Retreat - Run
freq	20898	25719	5752	17003	15559	9661	2474	4686

```
Complete_HWC_Data[Complete_HWC_Data.columns[0:11]].describe(include='object')
```

	UniqueID	Incident Number	Incident Date	Field Unit	Protected Heritage Area	Incident Type	Within Park
count	73658	73658	73658	73658	73658	73658	73618
unique	73658	64258	4299	19	35	9	2
top	BAN2010-0003.3	BAN2013-1151	2021-05-26	Jasper Field Unit	Banff National Park of Canada	Human Wildlife Interaction	Yes
freq	1	11	96	25982	27030	48673	72755

Methodology and Initial Observations

Figure 5: Graph of overall methodology.



Data Collection

Retrieve the three desired datasets and two reference documents fom Canada's Open Government Portal: "Human-wildlife coexistence incidents in selected national parks from 2010 to 2021", available at this link: https://open.canada.ca/data/en/dataset/cc5ea139-c628-46dc-ac55-a5b3351b7fdf. The names of the three datasets used in this project:

1. "Human-wildlife coexistence incidents detailed records 2010-2021 – Parks Canada" (English)
2. "Human-wildlife coexistence animals involved detailed records 2010-2021 – Parks Canada" (English)
3. "Human-wildlife coexistence activities detailed records 2010-2021 – Parks Canada" (English)
4. "Human-wildlife coexistence responses detailed records 2010-2021 – Parks Canada" (English). *Note*: *The responses dataset was merged with the others for the purposes of the EDA report and exploratory analysis phases, but is removed from Modeling steps because Modeling is seeking to predict "Incident Type" and the "Response" data is inherently dependent on "Incident Type" and therefore not independent values.*

The names of the resources pertaining to these datasets that I also used and are available at the above link:

- "Human-wildlife coexistence data dictionary – Parks Canada
- "Human-wildlife coexistence header descriptions – Parks Canada

Data Processing

This phase has been started by dealing with duplicate “Incident Number” observations in each of the 3 datasets. In the “Animals” dataset, I created a “UniqueID” column based on the “Incident Number” value and a running count of occurrences of that “Incident Number” so each observation had a unique identifier. In the “Incident” dataset, each duplicate occurrence of the “Incident Number” held an NA value as the “Incident Type” (and held no other new information) so I simple dropped those rows. In the “Activities” dataset, the “Activity Type” the Activity types were encoded using one-hot encoding so that each distinct category has a column with binary values of “0” for no and “1” for yes. I decided on this approach after discussing the options with Professor Abdou and I decided encoding these two variables was the best way to maintain the information in the dataset and be able to merge the 3 datasets together.

I then conducted data cleaning by examining the categorical feature in each dataset and checking their values for validity. I did this by comparing the unique values in the categorical features with the unique values for those features that were listed in the Data Dictionary. For any variables that did not match the valid values in the Data Dictionary, I either replaced them with the correct value (if the incorrect value just had a spelling error and the correct value was obvious), replaced with “Unknown” (if “Unknown” was a valid entry referenced in the Data Dictionary), moved the entry to the correct feature (where the value was valid in another feature and that row’s feature value was missing), or marked them as a missing value.

I then merged the three datasets together using the “UniqueID” value generated in the “Animals” dataset to ensure there was unique identifier for each observation in the newly combined dataset, named Complete_HWC_Data.

Exploratory Data Analysis

I generated an Exploratory Data Analysis Report using the Panda's ProfileReport. I used this report to get a general sense of the variance within variables, correlation between variables, missing values, distinct values per variable, etc. I used the datetime python module to convert my date into various formats, including creating a year column, a year and month column, and month column to my dataset so that I would be able to visualize patterns over those periods of time.

Using the matplotlib python library, I created various histograms and plots to further visualize the variables that I was most interested in examining. I first created histograms for each of the independent variables looking at frequency of each unique value (i.e. total incidents that were recorded for each unique value. For most attributes, the discrepancies in frequency counts across unique values were quite extreme so the y-axis were scaled logarithmically to help visualize the information (without applying the 'log' scale, it was not possible to see and understand the variations in frequencies for the values that had lower frequency counts.

Next, I created some plots to better visualize how many total incidents occurred in each National Park, to visualize the total number of incidents and separately, to visualize the incidents grouped by "Incident Type", in each of the National Parks. I then visualized the total number of incidents per year, grouped by "Incident Type", and on a separate plot, grouped by "National Park". I also visualize the total number of incidents per year, grouped by "Incident Type".

My biggest observations and take-aways based on the various plots I generated are:

- The majority of the incidents occurred in the "Banff", "Jasper", Field Units with over 20000 incidents, with the "Lake Louise, Yoho, and Kootenay Field Unit" being next highest but with significantly less incidents at just under 10,000 (see Figure 6).

- Looking at Incident Types, “Human Wildlife Interactions” is the most frequent at near 50,000 and the next highest is “Rescued/Recovered/Found Wildlife” at over 10,000. These will likely be the two Incident Types that are best predicted in my prediction model because there is so much data on them (see Figure 6).
- Most Incident Types increased over the years (particularly dramatic for “Human Wildlife Interaction”) (see Figure 7).
- All Incident Types increased during the summer months and decreased during the winter months (see Figure 8).

Figure 6: Total Number of Incidents by National Park and by Incident Type

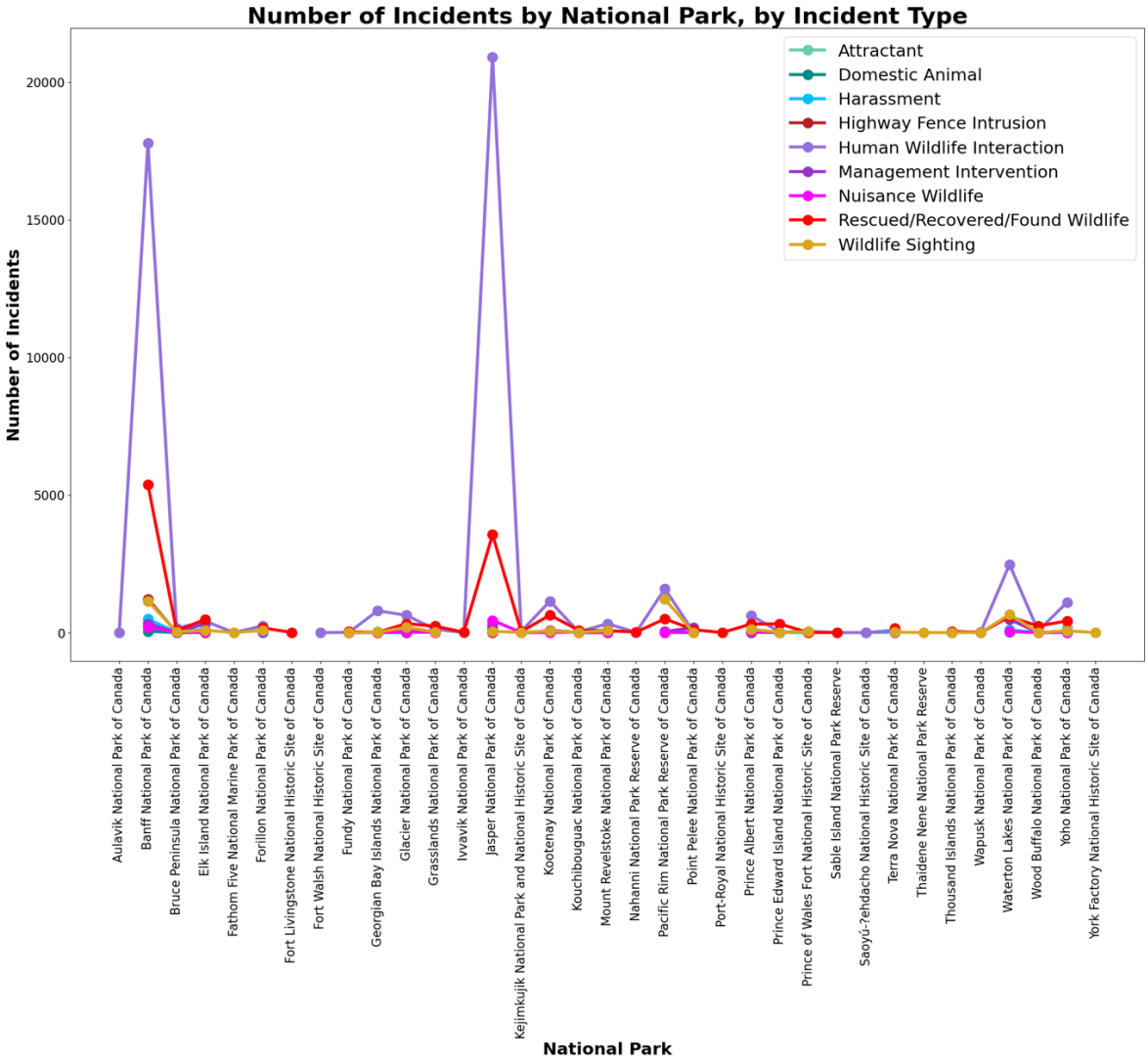


Figure 7: Total Number of Incidents per Year, by Incident Type

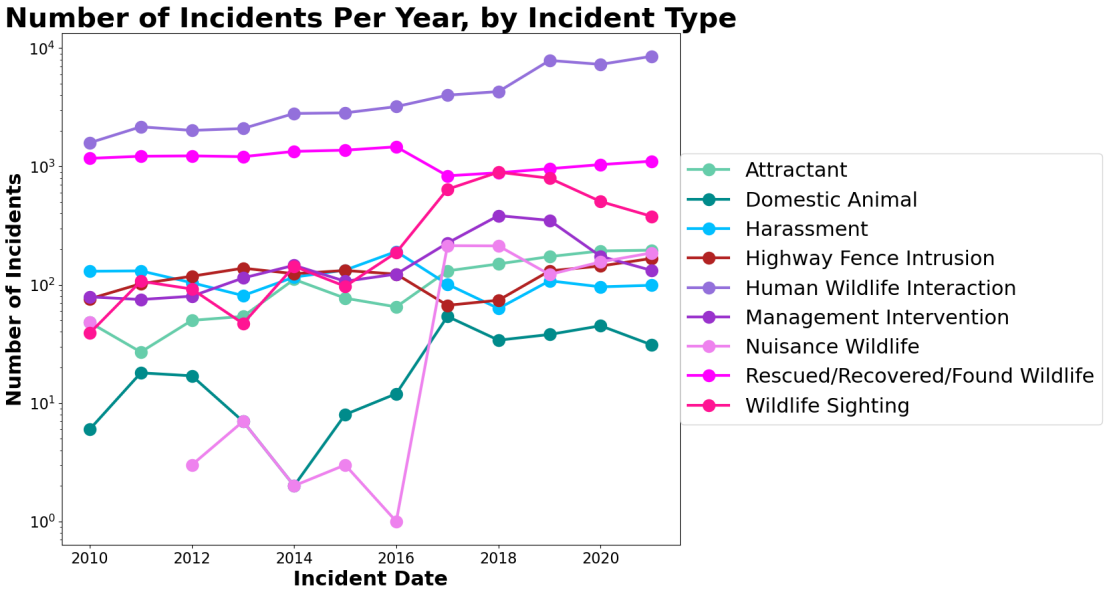
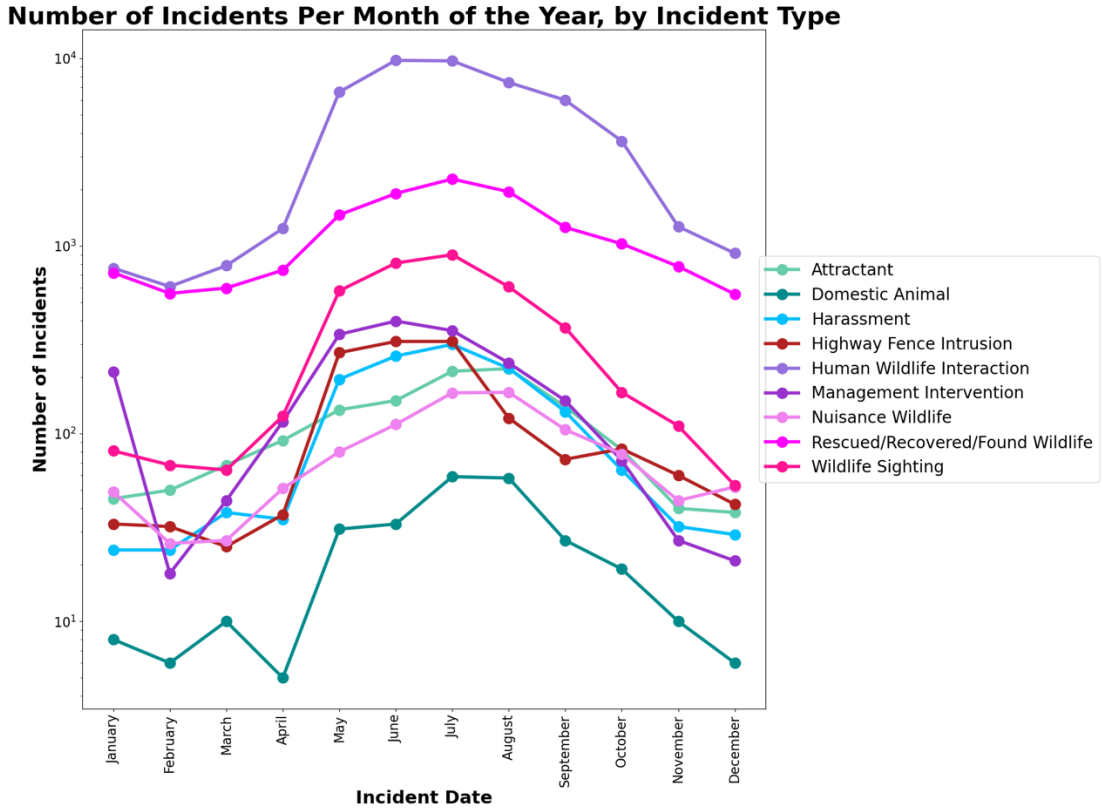


Figure 8: Total Number of Incidents per Month, by Incident Type



The last part of my Exploratory analysis was to run the Augmented Dickey-Fuller test of stationarity on the data using the python statsmodel, adfuller stat tool. I ran the test on all independent features that would be used in the modeling (excluding the “Activity Types” because that data was one-hot-encoded during the merge of datasets and would not work well with the test because of so many 0 values). The data was found to be stationary and would therefore work well for my analysis.

Modeling

Before getting to modeling there are some preprocessing and other steps that needed to be conducted on the data to prepare it for modeling. Because these steps will be dealt with in the creation of the model (after the splitting of data), I’ve included this information as part of the Modelling phase of the analysis.

The first thing I do before even splitting the data, is to create a subset of my entire dataset that removes the Response Type data. The reason for removing the Responses data from the modeling phase is because the Responses are inherently dependent on “Incident Type” and only independent variables should be used in the modeling when predicting a target variable.

Next, I sort all my data by “Incident Date”, reset the index, and then set “Incident Date as the index. I also add a column to my data set for “Incident Month” (extracting the month from the “Incident Date” because I want that extracted information to be included as a feature (looking at the plots I visualized for number of incidents per month, Month appears to be useful information). I drop the “Unique ID” and “Incident Number” columns from the data because those columns should not be used in the modeling. I decided to use a threshold of 10% to determine whether I would impute missing values or drop them. I was also able to tell from my EDA report, that there are several features that contain too many missing values (over 35%). Those features are as

follows and were dropped from my dataset at this stage (before splitting): "Animal Health Status", "Cause of Animal Health Status", "Animal Behaviour", "Reason for Animal Behaviour", "Animal Attractant", "Deterrents Used", "Animal Response to Deterrents".

Splitting Data

When it came time to split my data into training and test sets, I thoroughly investigated the TimeSeriesSplit function of the scikit-learn python package which is typically used for splitting and cross-validating data into train/test sets based on their date/time. As described by Howell (2023), doing a TimeSeriesSplit “means our test [data] is always in the future compared to the data our model is fitted on.”

Through all my research online, I was unable to find a clear way to use the TimeSeriesSplit function to split and cross-validate my data into train and test sets and also, then be able to apply various functions to those train sets before applying the model. For instance, after splitting, I need to be able to impute missing data in the training sets, encode the data in all sets (by fitting to train and applying to both train and test), apply a couple different feature selection methods, and balance the training data. It was important that these steps all occurred after the split (and not before) to avoid any data leakage between the train and test sets. The resources that I was able to find on using the TimeSeriesSplit, did not conduct any other step between split and modeling and therefore I was not able to figure out how to apply my interim steps accordingly when using that package.

Instead of using the TimeSeriesSplit function, I instead decided to manually split my data into 5 folds of train/test sets. I chose 5 folds as that was found to be a common approach in the literature, and would provide me with a lot of information for comparison sake, while also being manageable time and efficiency wise. Additionally, because my dataset spans 11 years, using 5

folds worked neatly (increasing Training set by 2 years with each fold), and I was able to split as follows:

- **Fold 1** → Training set (2010-2012) and Test set (2013);
- **Fold 2** → Training set (2010-2014) and Test set (2015);
- **Fold 3** → Training set (2010-2016) and Test set (2017);
- **Fold 4** → Training set (2010-2018) and Test set (2019);
- **Fold 5** → Training set (2010-2020) and Test set (2020).

At this stage, I only split the data into train and test sets, and did not split X from y. I will be splitting X and y after imputing missing values, and encoding to categorical, but before encoding to numeric (see below).

Dealing with Missing Values

The dataset had a few variables under the 10% threshold for imputation that needed to be dealt with. I applied different methods of imputation depending on the feature. Each of the 5 Train set folds were imputed individually. These methods are outlined as follows:

For the features “Species Common Name”, “Sum Number of Animals”, and “Within Park”, I chose to impute these values with the mode of the feature, in other words, with the most frequently occurring value for that column. I imputed these values using .fillna and .value_counts functions.

For the Latitude and Longitude values, because these features represent a geographical location which could be important for predicting “Incident Type”, I wanted to maintain the integrity of this feature as much as possible. In other words, I didn’t want to simply impute with the most frequently occurring value, because that Latitude and/or Longitude value might represent a

geographical area on the other side of Canada. To facilitate this, I chose to impute the missing values with the mean value for that given National Park (i.e. Protected Heritage Area). So for example, if the missing Latitude and/or Longitude was in a row for the "Grasslands National Park of Canada", I imputed the missing value with the mean "Latitude" or "Longitude" value for the "Grasslands National Park of Canada". I imputed the values using .fillna and referencing dictionaries to .map the imputations.

There was also one "Activity Type" row that had missing values for each "Activity Type" feature. Because there are about 90 "Activity Type" columns, all of which were missing 1 value for this single row, I decided to simply drop the affected row rather than impute those values.

Encoding the Data

Encoding Numeric to Categorical:

The methods I use for balancing the data, feature selection, and modeling, all require the data to be encoded to numeric form. Most of my data was categorical; however, a few features were already numeric. Before encoding the dataset, I first wanted to normalize the numeric data by converting it to categorical (creating groups/bins of similar data). These bins would then be encoded back to numeric along with all the other features in the next step. The features that needed to be converted to categorical were: "Latitude Public", "Longitude Public", "Total Staff Involved", "Total Staff Hours", and "Sum of Number of Animals".

Of course, the best practice when encoding data is to fit the encoding to the training data, and then apply that fit to both the train and test data. I did this step manually by investigating the distribution of data for each feature in the largest (i.e. 5^{th}) fold of training data sets. Based on this distribution, I set the bin boundaries and then applied those bins to all folds of the training data and test data sets. For Latitude and Longitude values, the 10 bins were automatically generated

on an even split. I chose to do 10 bins to best be able to maintain the general geographic location from east to west Canada. For “Total Staff Involved”, “Total Staff Hours”, and “Sum of Number of Animals”, I selected the bins based on frequency rather than an even split of the distribution. This is because in all three of these features, by far the most common value was “1”, with fewer and fewer instances of the other values as the number increased.

Encoding Categorical to Numeric:

At this stage, and before converting the entire data sets to numeric, I first split each train/test set into X, y sets by sub-setting on “Incident Type”.

Using recommendations and code examples from Brownlee (2020), I apply scikit-learn package’s OrdinalEncoder class to encode the input features (X datasets) and the LabelEncoder (which is designed to be used on a single feature) to encode the target feature (y datasets). I fit both the OrdinalEncoder and LabelEncoder onto the train data set for each of the 5 folds, and then apply the encoding to both the train and test sets for that fold.

Feature Selection

In this section, I apply and compare three different techniques (one each of filter based, wrapper, and intrinsic) for feature selection and compare the resulting ranked features. I refer to the filter based and wrapper based methods as “independent” to distinguish them from the intrinsic method used.

The first independent technique I use is the Chi-Squared, filter-based technique and I apply this using SelectKBest method from the scikit-learn library. I chose Chi-Squared as filter-based technique as that was the method used by Baral et al. (2021) and Naha et al. (2020) when conducting similar analysis (as I discovered during my literature review).

The second independent technique I use is the Forward Elimination, wrapper method using the SequentialFeatureSelector function of the .feature_selection method from the mlxtend python library. I decided to use Forward Elimination after comparing results from Backward Elimination and finding the Forward method to perform slightly better.

Each of the two independent feature selection techniques were applied to each of the 5 folds of train and test data. For each technique, the features selected on each fold were then compared, and the average top 6 features across all folds were taken as the “Top 6” features identified by that technique. I chose to select and work with the “Top 6” features in each method after examining the scores for each feature found by these techniques and observing that the after the Top 6, the scores for further features were quite low.

The 6 features found across the 5 folds selected by SelectKBest using Chi-squared can be seen in Figure 9. As shown, the “Top 6” features across all folds for this technique are: 'Species Common Name', 'Protected Heritage Area', 'Field Unit', 'Total Staff Hours', 'Sum of Number of Animals', and 'Incident Month'.

The 6 features found across the 5 folds selected by Forward Elimination can be seen in Figure 10. Whereas with SelectKBest, I chose the Top 6 features by looking at the average score for each feature across all folds, with Forward Elimnation, I chose the selected features for the fold that performed best. The reason I did this, is the Forward Elimination technique ranks the accuracy for the selected set of features, whereas the SelectKBest, ranks the accuracy for features individually. As shown in Figure 11, the “Top 6” features for the Forward Elimination technique were found in Fold 3. Those features relate to: 'Species Common Name', 'Field Unit', 'Total Staff Hours', 'Sum of Number of Animals', 'Activity Type_Railway', 'Latitude Public'.

Figure 9: SelectKBest: Selected Features across 5 folds

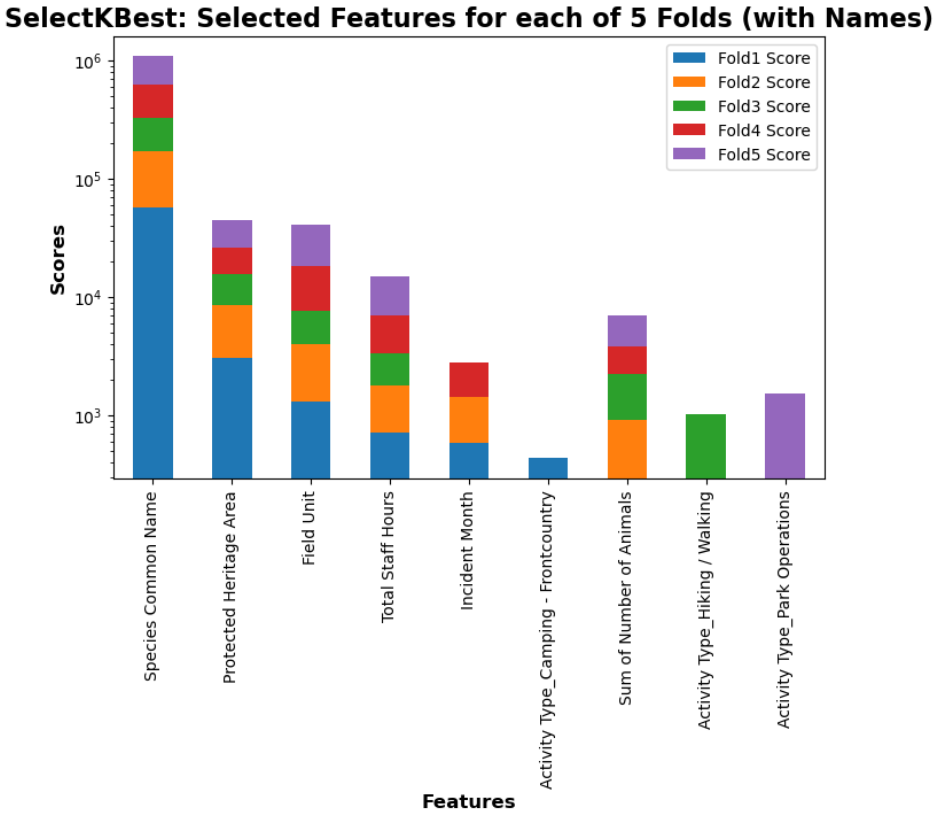


Figure 10: Forward Elimination: Selected Features across 5 folds.

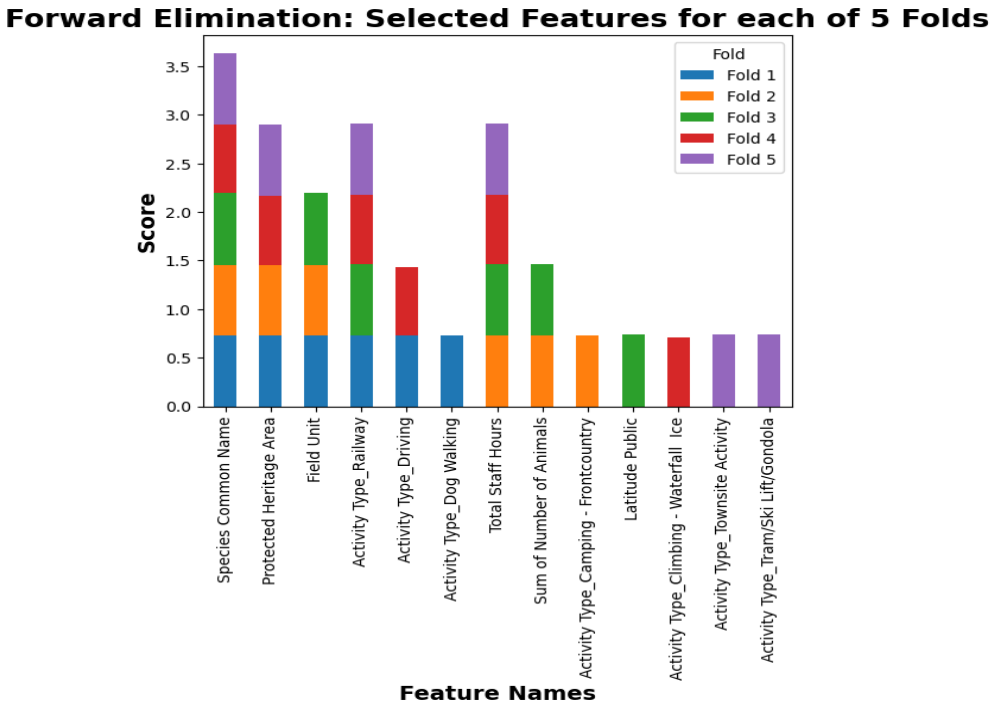
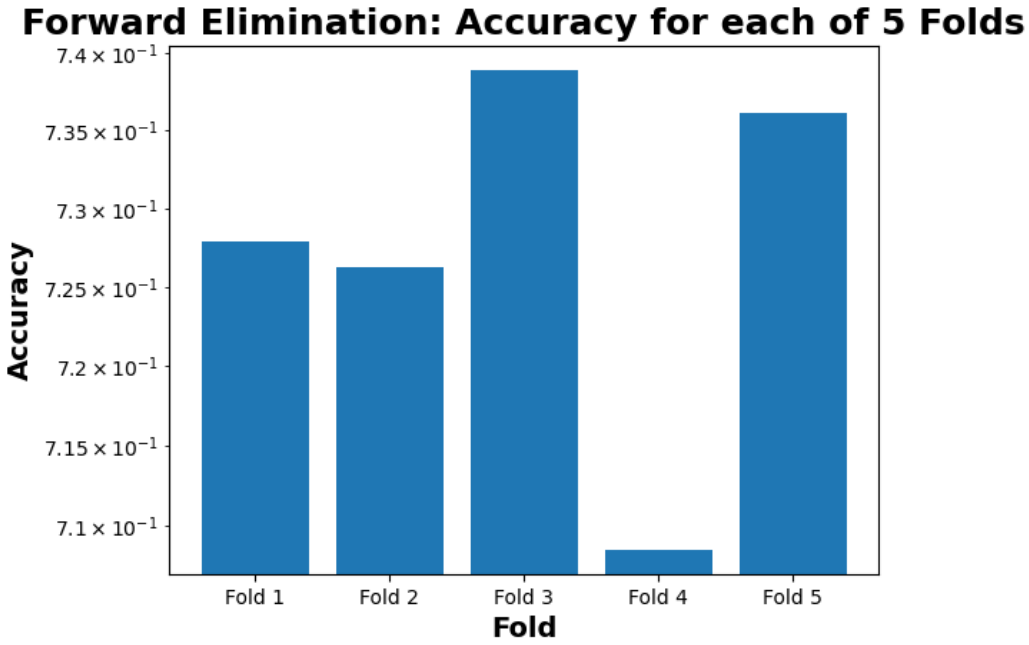
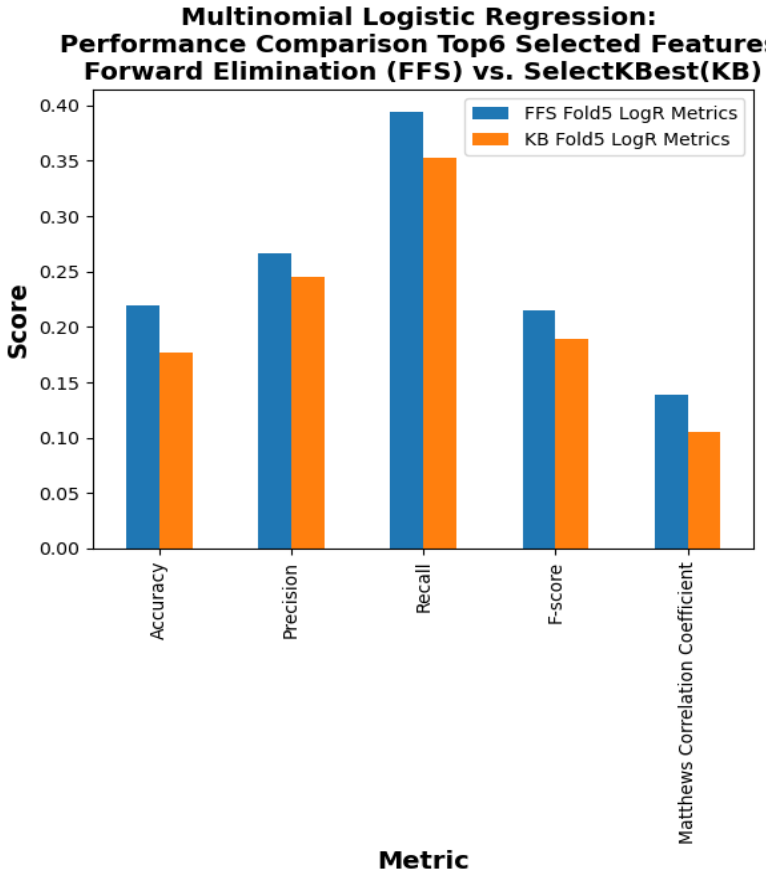


Figure 11: Forward Elimination Accuracy Scores per Fold.



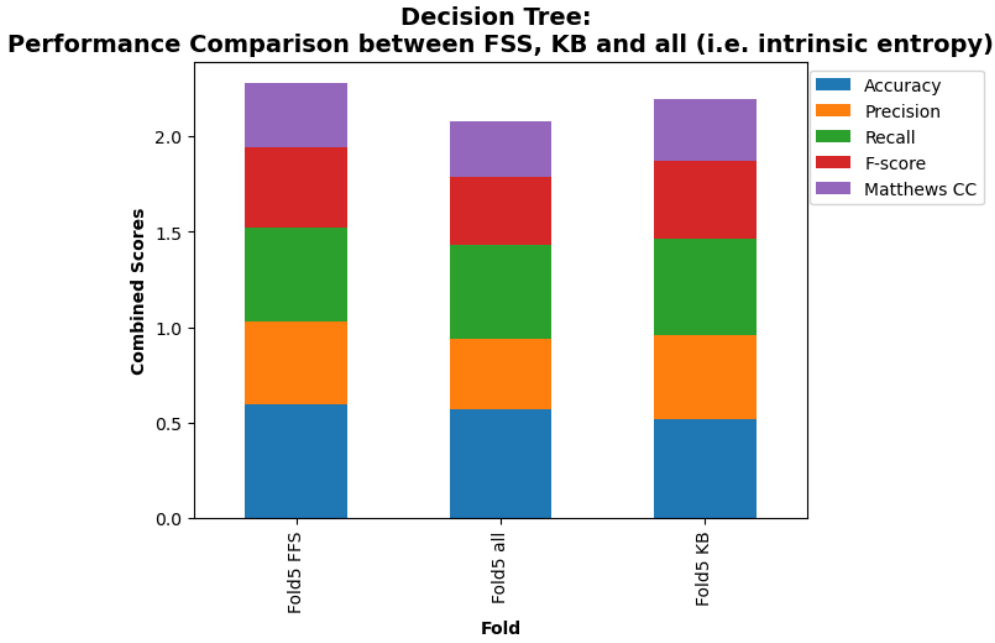
To determine which feature selection technique provides the best set of features, I used my Multinomial Logistic Regression model, and the largest (5^{th}) fold of train and test data, and ran the model two times. For the first model run, I use the Top 6 features identified by the SelectKBest chi-squared feature selection technique. For the second model run, I use the Top 6 features identified by the Forward Elimination technique. I did not use an intrinsic method with the Multinomial Logistic Regression model as when I attempted to run the model on all features, the code would run for too long and Jupyter Notebook would time out. I compared the model results for each subset of features used and found that the features identified by Forward Feature Selection performed better overall than those identified by SelectKBest (see Figure 12).

Figure 12: Forward Elimination vs. SelectKBest in Multinomial Logistic Regression model.



I also ran a similar test to compare the selected features using the Decision Tree Classifier model. On this one, I again used the largest (5^{th}) fold of the train test data, and ran the model three times. For the first model run, I use the Top 6 features identified by the SelectKBest chi-squared feature selection technique. For the second model run, I use the Top 6 features identified by the Forward Elimination technique. And for the third mode, I use the intrinsic feature selector of "entropy". I compared the model results for each subset of features used and found that while the performances were pretty close, the features identified by Forward Feature Selection performed better overall than those identified by SelectKBest, or intrinsic Entropy (see Figure 13).

Figure 13: Forward Elimination vs. SelectKBest vs. intrinsic Entropy in Decision Tree Classifier model.



With the features selected with Forward Elimination performing better with both Multinomial Logistic Regression and Decision Trees, when I got to my third model using Random Forest Classifier, I decided to use not use the SelectKBest subset of features and only compare Forward Elimination to the Random Forest intrinsic Gini index selector. The results of this comparison were extremely close (as visualized in Figure 14 and Figure 15. The Random Forest's intrinsic feature selection performs very slightly better than using the Forward Feature Selection Top 6 Features. They are so close, I could probably have used either; however, considering that the Random Forest Model takes a bit more time (regardless of whether I use the Forward Elimination subset of features or all features), and the Forward Elimination process itself took long enough to process, I decided to use the intrinsic feature selection method here to reduce the overall processing time for this model.

Figure 14: Forward Elimination vs. intrinsic Entropy in Random Forest Classifier model, Plot 1

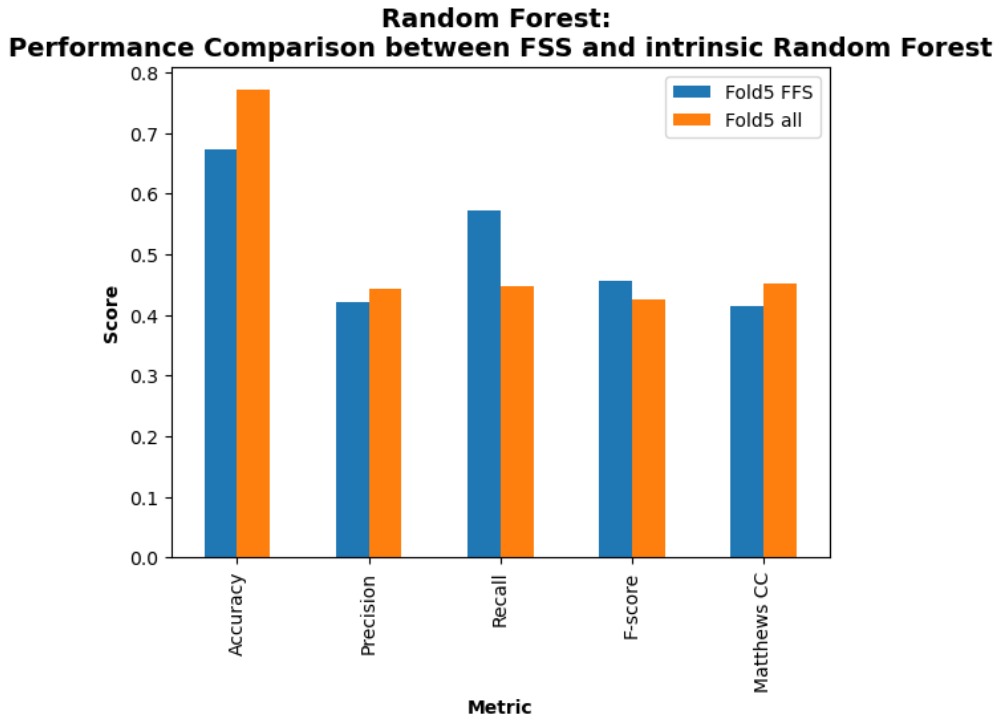
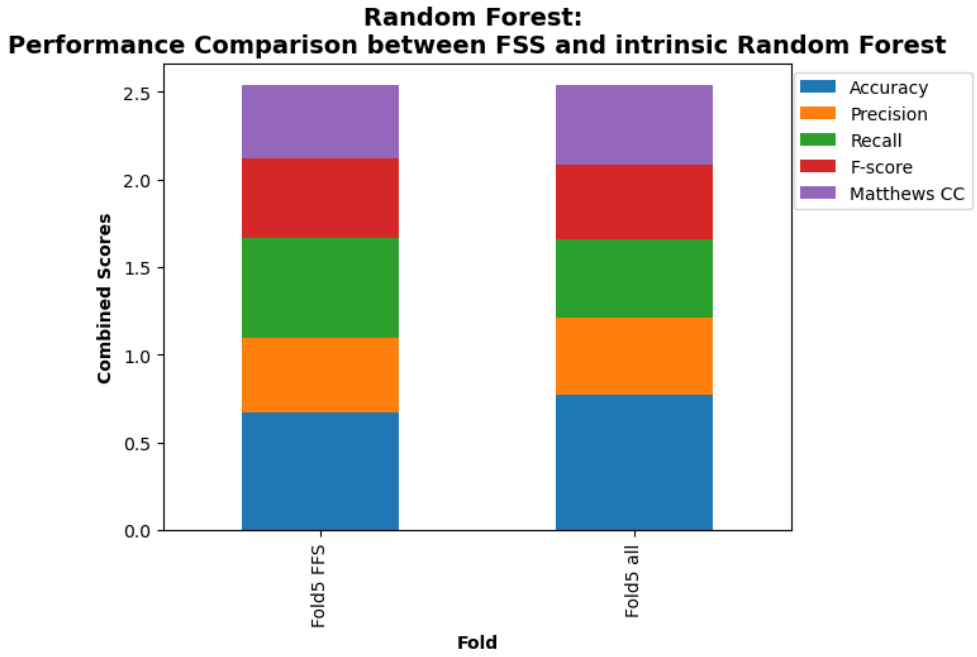


Figure 15: Forward Elimination vs. intrinsic Entropy in Random Forest Classifier model, Plot 2



When using the Random Forest intrinsic feature selector (Gini index), the Top 6 features the model uses (for comparison against the features selected by SelectKBest and Forward Elimination) are: “Total Staff Hours”, “Incident Month”, “Species Common Name”, “Total Staff Involved”, “Protected Heritage Area” and “Field Unit”. An important note is that the Random Forest model does use other features as well, these are just the top 6 ranked by that model.

The selected features used by the Forward Elimination method (used in the Multinomial Logistic Regression and Decision Trees models) vs. the intrinsic feature selection of Gini index used in the Random Forest models are summarized in the table at Figure 16.

Figure 16: Table of Selected Features using Forward Elimination and Random Forest’s intrinsic Gini Index

Forward Elimination	Random Forest intrinsic (Gini Index)
Species Common Name	Total Staff Hours
Total Staff Hours	Incident Month
Activity Type_Railway	Species Common Name
Field Unit	Total Staff Involved
Sum of Number of Animals	Protected Heritage Area
Latitude Public	Field Unit

Dealing with Imbalanced Data

Note: the above comparisons of feature selectors using the 3 models were conducted after the data was balanced.

My dependent variable, Incident Type, is very imbalanced. Boyle (2019) succinctly addresses the problem with imbalanced classes as “Most machine learning algorithms work best when the number of samples in each class are about equal. This is because most algorithms are designed to maximize accuracy and reduce error.”

I chose to use the Synthetic Minority Oversampling Technique (SMOTE) oversampling technique to deal with my imbalanced data. Brownlee (2021) describes "SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line."

I chose to conduct over-sampling to balance the data instead of under-sampling because the number Incident Types in the minority class is so low that if I were to under-sample and bring my majority class down to the same frequency as my minority class, I would no longer have enough observations remaining in my dataset.

All 5 folds of training data were oversampled using the same technique.

Models

I chose to build three different models to compare the results. All three models were applied to each of the 5 folds of training and test data and the performances across these 5 folds were compared and averaged. I then took the average performance of each model and compared it to the other models to make an observation regarding the best performer. The models I used are summarized below.

1. **Multinomial Logistic Regression**: I chose to use a Multinomial Logistic Regression model because I noticed it was the model often chose by others (Baral et al. (2021) and Naha et al. (2020) doing similar research when I conducted my literature review. I used the scikit-learn library's linear_model.LogisticRegression function. With this model, as described under the "Feature Selection" section above, I used the "Top 6" features identified using the Forward Elimination feature selector. See Figure 17 and Figure 18 for visualization of performance metrics on this model. Figure 17 shows

each of the performance metrics and how each fold scored, and Figure 18 shows the total scores for each fold and represents the stability of the model across folds.

Figure 17: Multinomial Logistic Regression, Plot 1: Performance Metrics Across 5 Folds

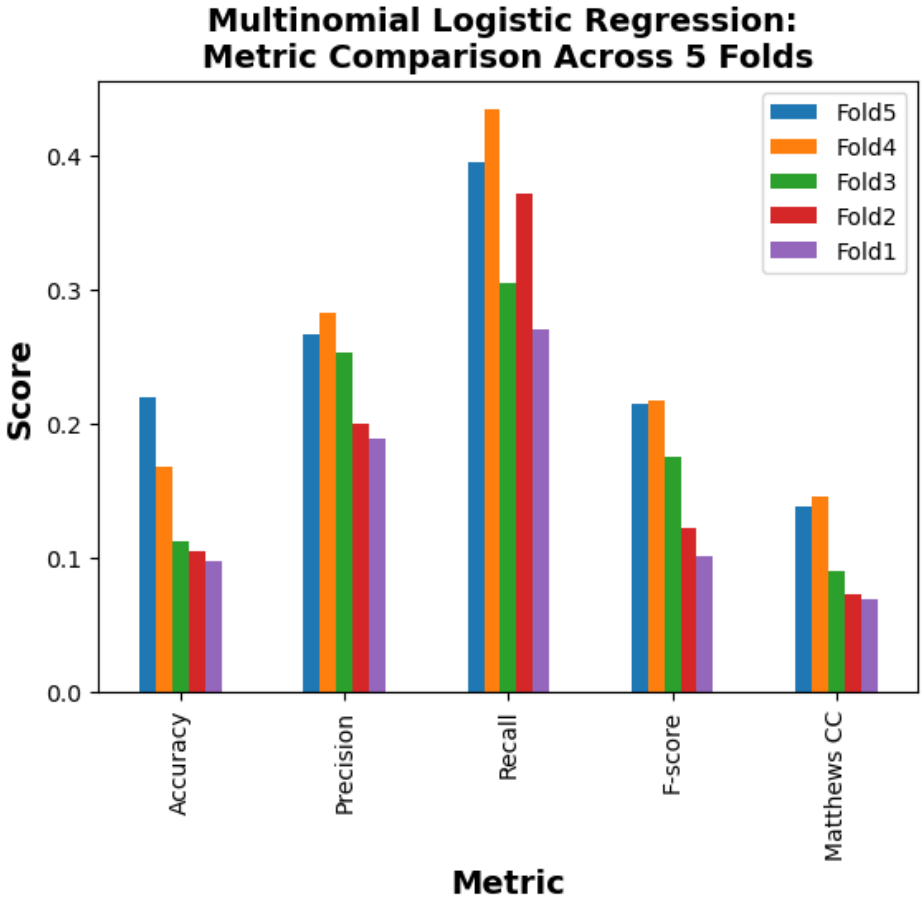
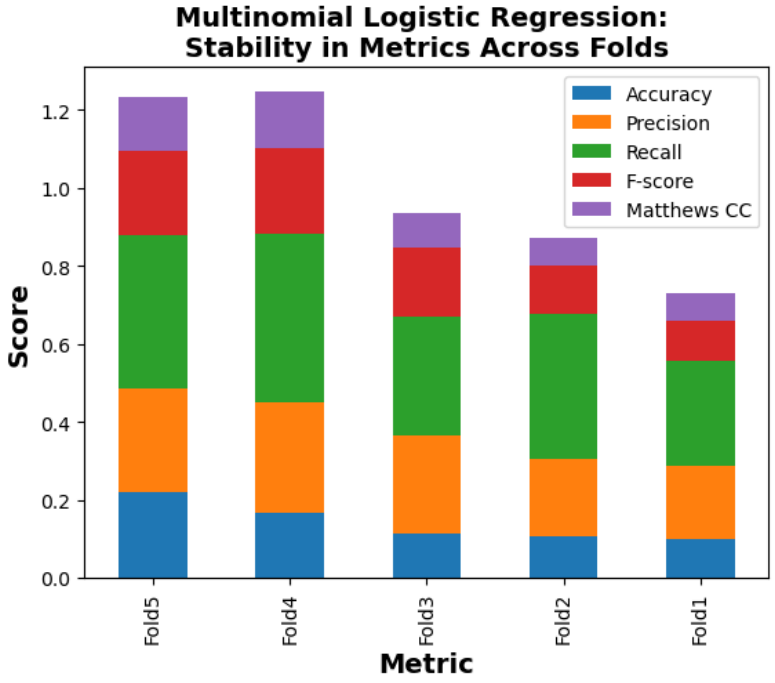


Figure 18: Multinomial Logistic Regression, Plot 2: Stability in Performance Metrics Across 5 Folds



2. **Decision tree classifier**: I chose one model to be the decision tree classifier because it works well with multiple classification problems and is a good model to use when the target variables are categorical. For my decision trees, I used scikit-learn python package and DecisionTreeClassifier function. With this model, as described under the "Feature Selection" section above, I used the "Top 6" features identified using the Forward Elimination feature selector. See Figure 19 and Figure 20 for visualization of performance metrics on this model. Figure 19 shows each of the performance metrics and how each fold scored, and Figure 20 shows the total scores for each fold and represents the stability of the model across folds.

Figure 19: Decision Tree Classifier, Plot 1: Performance Metrics Across 5 Folds

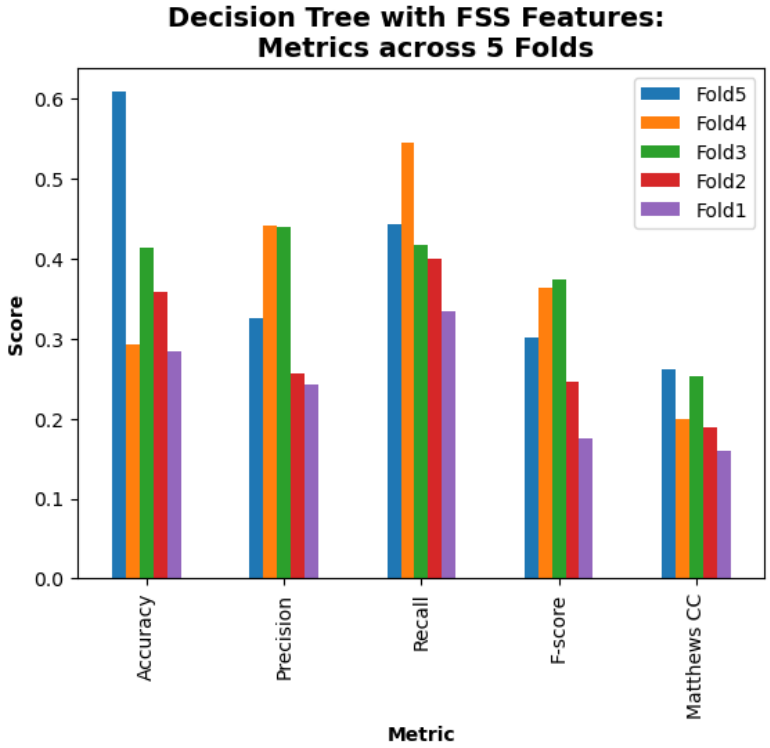
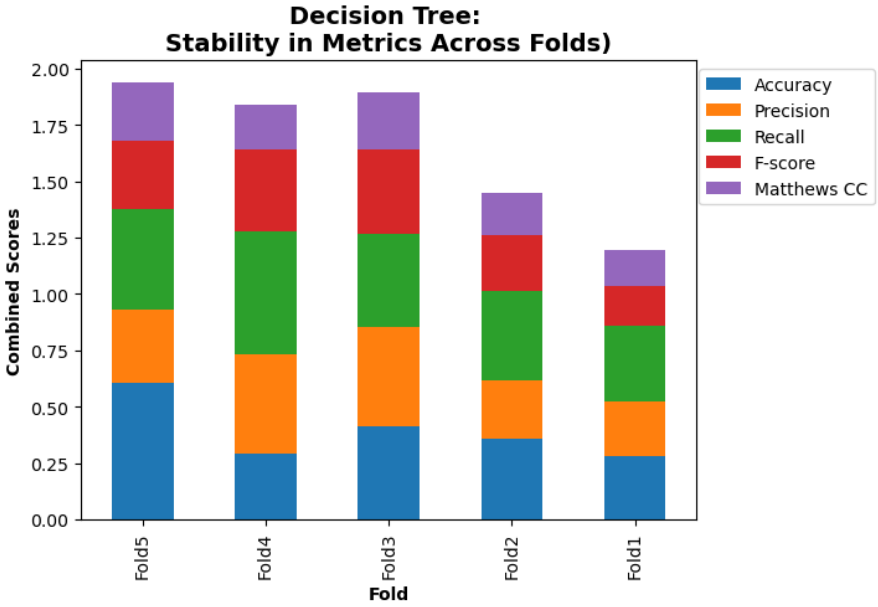


Figure 20: Decision Tree Classifier, Plot 2: Stability in Performance Metrics Across 5 Folds



3. **Random Forest Classifier**: I chose one model to be the random forest classifier because it also works well with multiple classification problems and is a good model to use when the target variables are categorical. I also recall both from our lectures, and from the research I conducted, that Random Forest is said to typically outperform Decision Trees and I wanted to see the comparison for myself. For my random forest models, I used scikit-learn python package and ensemble RandomForestClassifier function. With this model, I used the default number of trees (100) and the "gini index" criterion for feature selection. I did run a few comparisons to see if increasing the number of trees to 500 and/or using "entropy" as my feature selection criterion would perform better, but I found the model scores to be equivalent and chose to stick with the default values for the sake of model run time. See Figure 21 and Figure 22 for visualization of performance metrics on this model. Figure 21 shows each of the performance metrics and how each fold scored, and Figure 22 shows the total scores for each fold and represents the stability of the model across folds.

Figure 21: Decision Tree Classifier, Plot 1: Performance Metrics Across 5 Folds

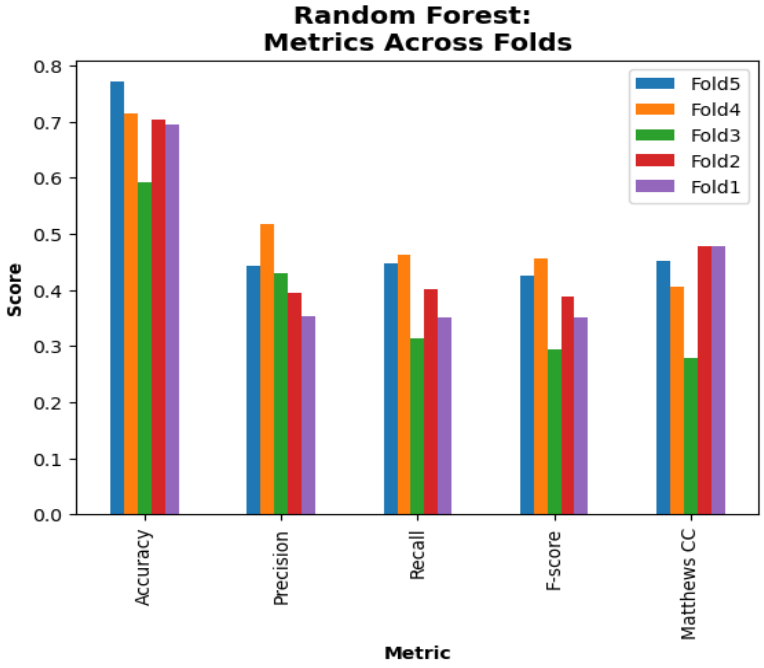
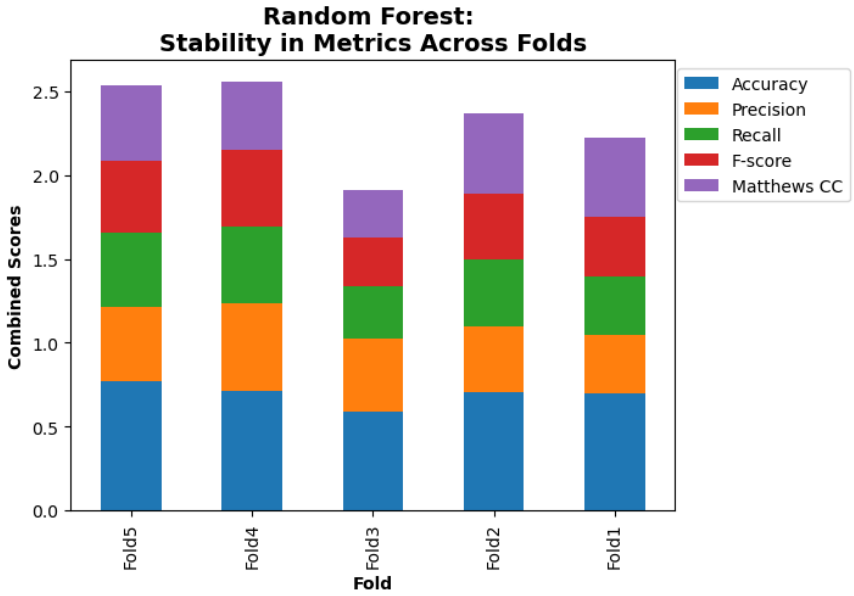


Figure 22: Decision Tree Classifier, Plot 2: Stability in Performance Metrics Across 5 Folds



Evaluation

The common evaluation metrics are accuracy, precision, recall, and F-score. Accuracy represents the fraction of predictions our model predicted correctly and is often not the best metric to use when evaluating imbalanced datasets. Precision is a good metric when your main aim is to minimize false positives and recall is a good metric when you want to maximize the true positive rate. In this situation, I see it as more important to maximize the true positive rate (successfully predict incident types) than it is to minimize false positive rates so while I will be considering all the performance metrics in my evaluation, I will consider high recall as more important than high precision. F-score combines the precision and recall metrics.

I've also chosen to measure the Matthews correlation coefficient (MCC) which takes into account true positives, true negatives, false positives and false negatives. Because the MCC takes into account all categories of the confusion matrix, it can be particularly useful (more so

than the accuracy score) for imbalanced datasets. I will consider a high MCC score as the most insightful metric.

To evaluate and compare the results of the three models, I took the average of each metric across all 5 folds for each model. Figure 23 and Figure 24 plot that data. In Figure 23, you'll see that the Decision Tree model's precision score was much higher than that of the other models; however the MCC score is much higher with the Random Forest model than the other two. In Figure 24, you can see that combined, the Random Forest model scores highest.

Figure 23: Model Comparison, Plot 1: Comparing Average Metrics for each Model

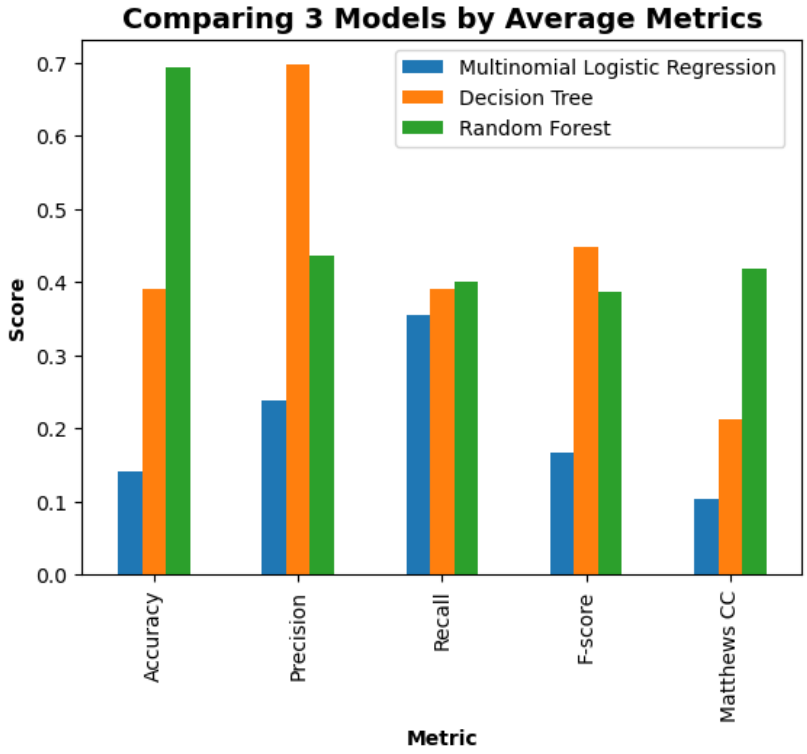
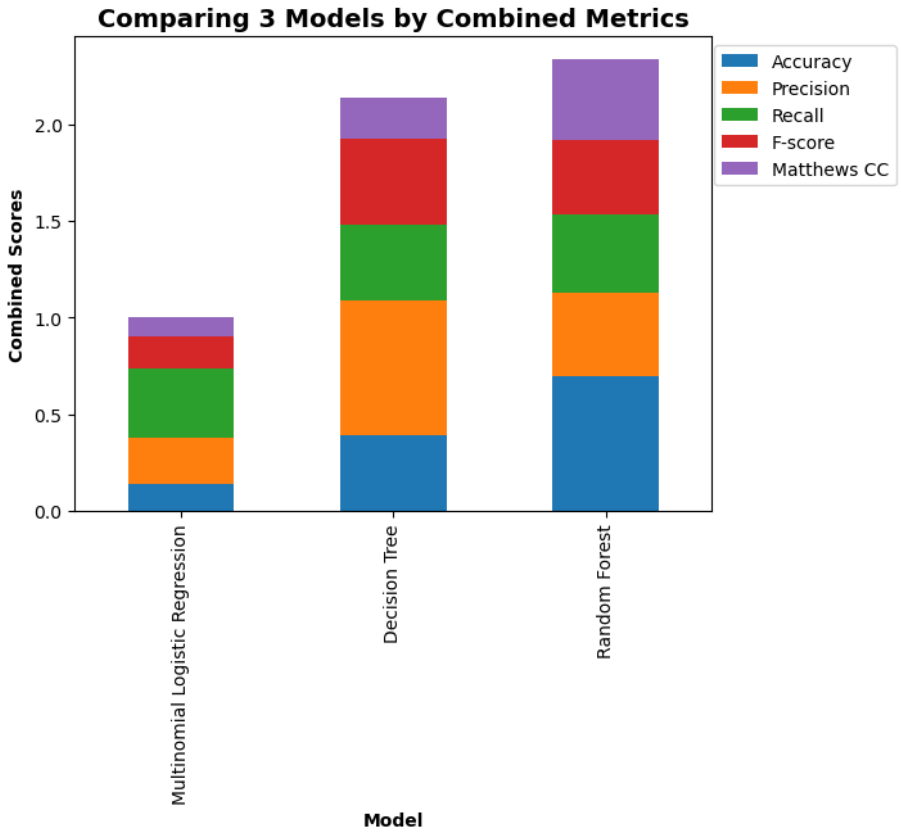


Figure 24: Model Comparison, Plot 2: Comparing Overall Metrics for each Model



I also looked at the time taken (measured in seconds) and computer memory used (measured in gigabytes (GBs)) to run each model, including time and memory consumption for feature selection (if the feature selection was done separately from the model). I plot those comparisons separately from the other metrics for better visuals. Figure 25 and Figure 26 plot this data. You can see in these plots that the Random Forest models take less time and memory to run.

Figure 25: Model Comparison, Plot 3: Comparing Time Used in seconds for each Model

Note: *the time taken on the Decision Tree model was so negligible that it does not appear on the plot and you only see the time for the feature selection.*

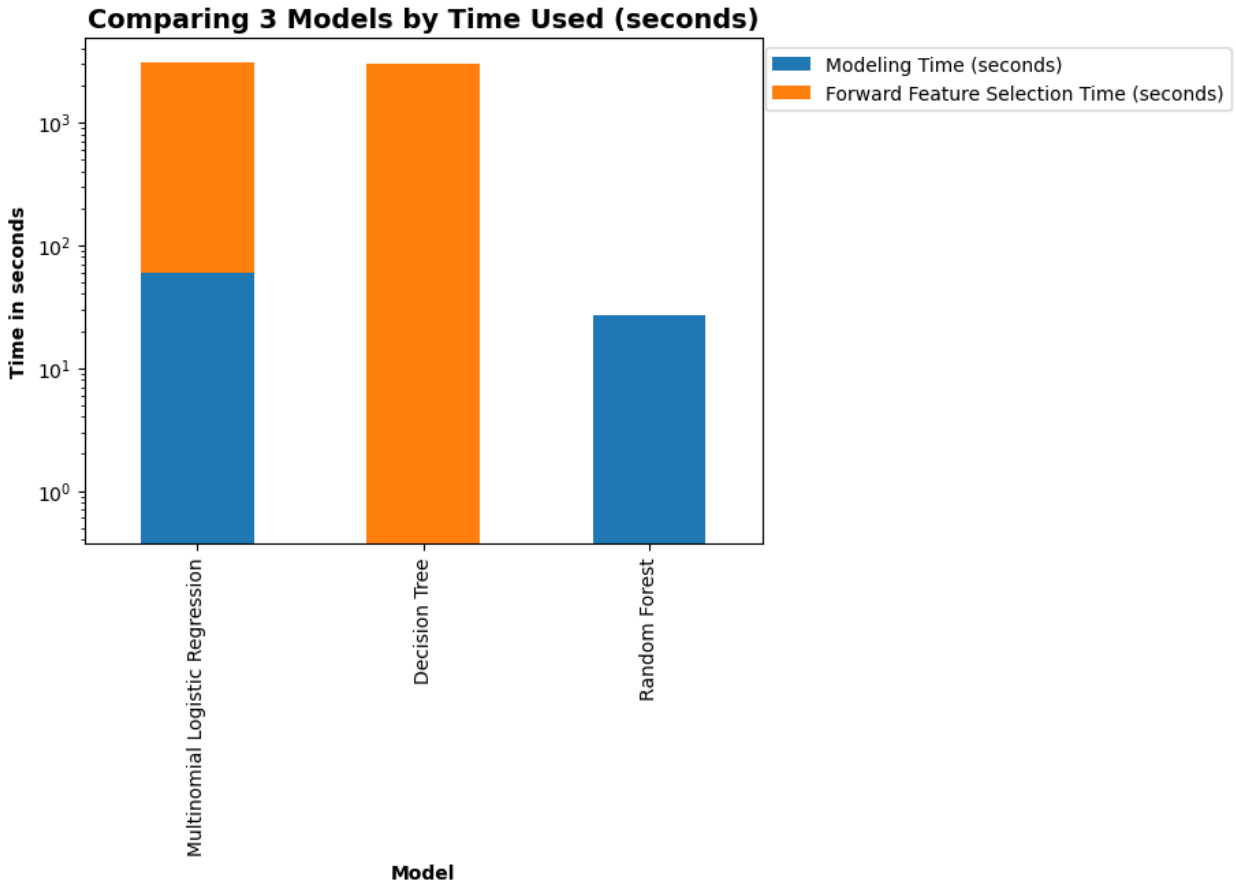
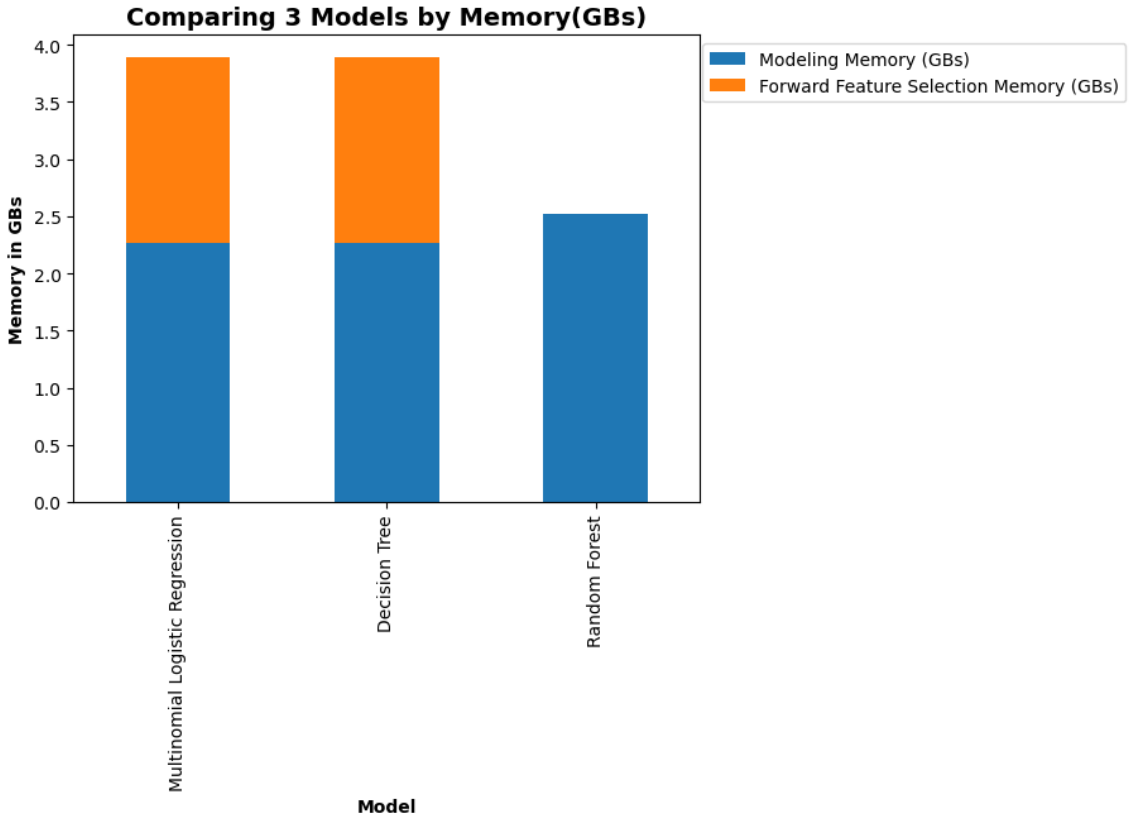


Figure 26: Model Comparison, Plot 4: Comparing Memory Used in seconds for each Model



Based on these plot comparisons, I observe that the Decision Tree and Random Forest are pretty comparable to in terms of performance metrics, with the Random Forest classifying coming in with slightly higher metrics, particularly on the Matthews Correlation Coefficient which is an excellent metric for overall performance. The Random Forest models also take significantly less time and memory and are therefore more efficient to run. I look further at the differences between the models and test whether the difference is statistically significant in the “Findings and Results” section below.

Findings and Results: Answering the Research Questions

Research Question 1:

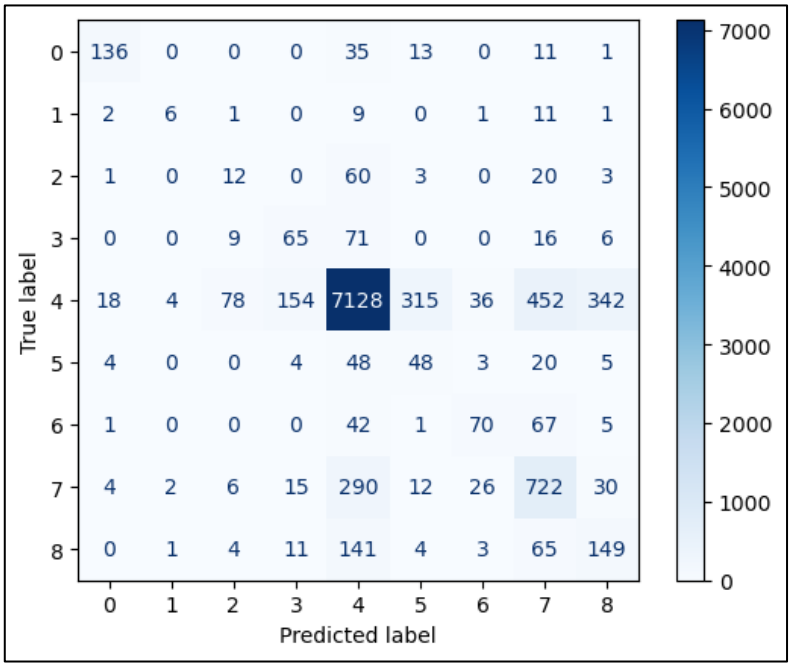
Question: What patterns can be visually observed in location and time of year for each of the following variables: human activities, animals involved, cause, and incident type. How do these patterns differ and across National Parks and over time?

The prevalent patterns to be observed in the data are as follows. These patterns are visualized in Figures 6-8 which can be found under the "Exploratory Analysis" section above.

- The Parks with the most incidents are Banff and Jasper.
- Almost half the number of Parks have fewer than 100 Incidents
- The majority of Incident Types are Human Wildlife Interaction, and the next highest is Rescued/Recovered/Found Wildlife. When observing this, I assumed that these two Incident Types will likely have more correct predictions than the other Incident Types because they have more data. Interestingly, if you look at Figure XX below, you can see that this assumption was correct and Incident Type 4 and 7 ("Human Wildlife Interaction" and "Rescued/Recovered/Found Wildlife" had the highest number of correct predictions). I did not want to include confusion matrices from every model I ran (3 models X 5 folds = 15 confusion matrices), but Figure 27 portrays the confusion matrix for the largest (5^{th}) fold from the Random Forest model.
- Year Trend for Incident Types: the Human Wildlife Interaction Incident Type has Increased significantly, Rescued/Recovered/Found Wildlife has slightly decreased over time. Wildlife Sighting and Management Intervention both seemed to spike around 2018.

- Year Trend for Parks: Jasper and Banff both increased in number of incident types over the years, with Jasper going up in 2021 and Banff going down in 2021. All other parks seems to have mainly remained at a similar level each year, with Waterton and Pacific Rim having some increase over the years.
- Month trend for Incident Types: All Incidents increase during the warmer months between May to October.

Figure 27: Confusion Matrix from Fold 5 of Random Forest Model:



0 → Attractant
1 → Domestic Animal
2 → Harassment
3 → Highway Fence Intrusion
4 → Human Wildlife Interaction
5 → Management Intervention
6 → Nuisance Wildlife
7 → Rescued/Recovered/Found Wildlife
8 → Wildlife Sighting

Research Question #2:

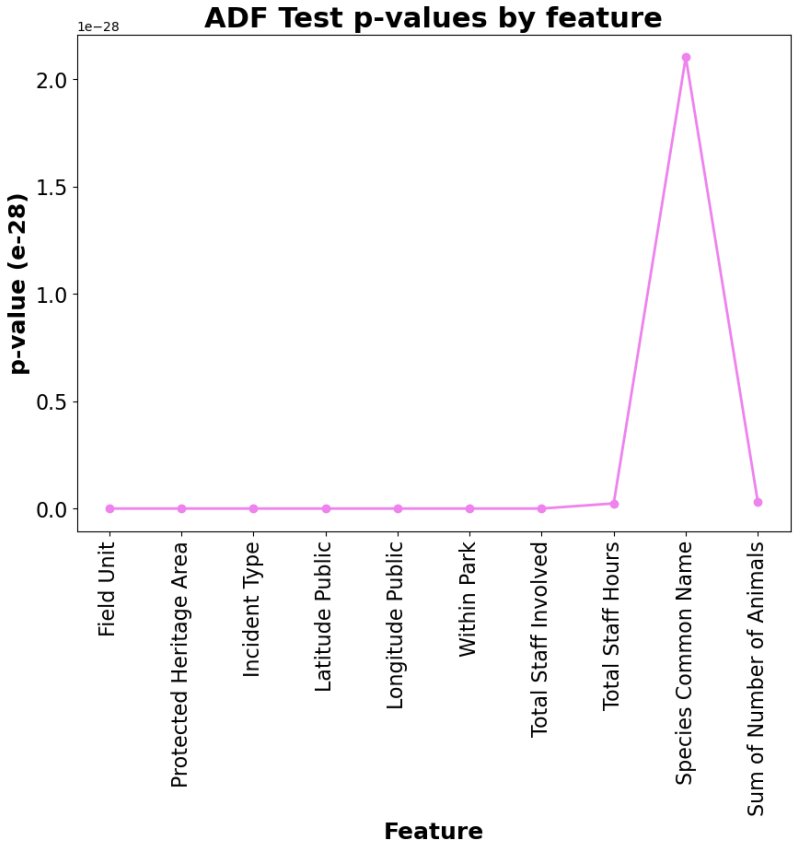
Question: Are the statistical properties of the data stationary over time?

I wanted to determine stationarity because "stationarity is assumed by most of the algorithms" (Radečić, 2020) and because my data is time based, I wanted to confirm that it was stationary before proceeding to further analysis and modeling. As described by Bex T. (2021), if the data was found to be non-stationary, we would need to take steps to address it and transform the data to stationary (for example, by using differencing) before proceeding to further analysis and modeling.

As part of my Exploratory Analysis, I ran the Augmented Dickey-Fuller test of stationarity on the data using the python statsmodel, adfuller stat tool. I ran this test on the independent features that I would be using in my model (excluding the "Activity Types" because that data was one-hot-encoded during the merge of datasets and would not work well with the test because of so many 0 values). I chose a significance level of 0.05%.

According to Statology (2021), the null hypothesis with this test is that the time series is non-stationary and the alternative hypothesis is that the time series is stationary. The p-values for each feature were less than 0.05 so we can reject the null hypothesis and conclude that the time series is stationary. For reference, the p-values are depicted in Figure 28 below, please note, the y-axis values are on a scale of e-28, so even the higher point for "Species Common Name" is below 0.05 with a p-value of around 2.0e-28.

Figure 28: Augmented Dickey-Fuller Test of Stationarity, p-value results



Research Question #3:

Question: What variables are most correlated with the occurrence of each incident type and what prediction model performs best in predicting “Incident Type”.

I noted my observations regarding which model performed best above, under the “Modeling: Evaluation” section. To truly understand which model performed best, I conducted statistical tests for significance in the variation between models. My data is not normally distributed, so I am using non-parametric tests. Further, because my scores are all achieved independently on different train/test sets, I applied the Friedman test which looks at paired samples for more than 2 groups. I ran the Friedman Test on each Metric separately, so Accuracy, for example, will have 3 groups (models) and 5 pairs (folds) for each model. I have 5 main metrics

(not including the time and memory metrics) and will run the Friedman Test on each of those 5 (that is, Accuracy, Precision, Recall, Fscore, and Matthews Correlation Coeffient).

The null hypothesis for these tests is that the mean score for all three models is the same. The alternative hypothesis for these tests is that at least one mean score for all three models is different. I have selected to use a 5% test for significance (which is the typical standard), so if the p-value is less than 0.05, we can reject the null hypothesis that the mean score is the same for all three models. The results for my Friedman Test can be found in Figure 29.

Figure 29: Results of Friedman Test for Significance on All Three Models

Friedman Test Results on All Three Models

Accuracy findings: The p-value (0.0067) is less than 0.05 so we **reject** the null hypothesis that the mean Accuracy for all three models is the same, i.e. at least one of the mean accuracies is significantly different than the others.

Precision findings: The p-value (0.8187) is greater than 0.05 so we **accept** the null hypothesis that the mean Precision for all three models is the same, i.e. there is no significant difference between the Precision scores across the three models.

Recall findings: The p-value (0.0067) is less than 0.05 so we **reject** the null hypothesis that the mean Recall for all three models is the same, i.e. at least one of the mean recalls is significantly different than the others.

Fscore findings: The p-value (0.0067) is less than 0.05 so we **reject** the null hypothesis that the mean Fscore for all three models is the same, i.e. at least one of the mean Fscores is significantly different than the others.

Matthews Correlation Coefficient findings: The p-value (0.0067) is less than 0.05 so we **reject** the null hypothesis that the mean Matthews Correlation Coefficient for all three models is the same, i.e. at least one of the mean Matthews Correlation Coefficients is significantly different than the others.

From my initial observations of the model performance, I knew that the Multinomial Logistic Regression model was the poorest performer and that the Decision Tree and Random Forest models were quite comparable in terms of performance metrics. Because of this, I decided to also test for

statistical significance in the variation between my two top performing models (Decision Tree and Random Forest).

For this test, I used the Wilcoxon Signed Rank non-parametric test (which is good for comparing 2 groups (my 2 models). Again, my null hypothesis for these tests is that the mean score for the two models is the same. The alternative hypothesis for these tests is that at least one mean score for the two models is different. I use the same 5% test for significance, so if the p-value is less than 0.05, we can reject the null hypothesis that the mean score is the same for the two models. The results for my Wilcoxon Test can be found in Figure 30.

Figure 30: Results of Wilcoxon Test for Significance on Top Two Models

Wilcoxon Test Results on Two Models (Decision Tree and Random Forest)

Accuracy findings: The p-value (0.0625) is greater than 0.05 so we **accept** the null hypothesis that the mean Accuracy for the two models is the same, i.e. there is no significant difference between the Accuracy scores between the two models.

Precision findings: The p-value (1.0) is greater than 0.05 so we **accept** the null hypothesis that the mean Accuracy for the two models is the same, i.e. there is no significant difference between the Accuracy scores between the two models.

Recall findings: The p-value (0.0625) is greater than 0.05 so we **accept** the null hypothesis that the mean Accuracy for the two models is the same, i.e. there is no significant difference between the Accuracy scores between the two models.

Fscore findings: The p-value (0.0625) is greater than 0.05 so we **accept** the null hypothesis that the mean Accuracy for the two models is the same, i.e. there is no significant difference between the Accuracy scores between the two models.

Matthews Correlation Coefficient findings: The p-value (0.0625) is greater than 0.05 so we **accept** the null hypothesis that the mean Accuracy for the two models is the same, i.e. there is no significant difference between the Accuracy scores between the two models.

I therefore conclude that the Decision Tree and Random Forest models perform equally well at predicting my target variable (“Incident Type”). That said, if you are taking run time and memory usage into consideration, the Random Forest model does perform better because of its

use of the intrinsic feature selector (compared to the Decision Tree model which relies on the independent Forward Elimination feature selector which is quite heavy for time and memory use).

The features most correlated with Incident Type (looking at the Features used in both the Random Forest and Decision Tree models are: Total Staff Hours, Incident Month, Species Common Name, Total Staff Involved, Total Staff Hours, Protected Heritage Area, Field Unit, Latitude Public, and Activity Type_Railway.

Shortcomings of the Work

Overall, with my Random Forest model having an average Matthews Correlation Coefficient score of 0.42, even my "best" models is not doing a great job of accurately predicting the Incident Types. Most likely, this is related to the dataset being quite imbalanced - with two Incident Types occurring much more often than the others. While I did balance the data before modeling, the confusion matrices and performance metrics make it clear that most of the successful predictions are for those two Incident Types (namely "Human Wildlife Interaction", and "Rescued/Recovered/Found Wildlife"). I would have liked to test different methods of balancing the data to see if I could get a model that would better predict the minority classes; however, due to time constraints, those comparisons could not be included in the scope of this project.

I was also very interested in looking further at what incidents and other attributes result in injury and/or death to the animal. Unfortunately, the sheer volume of missing data in the fields related to animal Health Status and Causes meant those variables had to be dropped from the general data. It would really interesting to instead only use those dropped variables (i.e. subset the data on the rows that contain values for those fields) and run the prediction model again to see if we can predict those. Again, due to time constraints, that examination could not be included in the scope of this project.

It was unfortunate that the Activity Type field needed to be one-hot-encoded when the datasets were joined. The reason for the one-hot-encoding was because the Activity data did not have a unique identifier for each observation, in other words, there were multiple “Activity Types” listed for a single incident report. In order to merge the datasets, I needed to either create lists of “Activity Types” for a single incident, which would have led to more unique values for that feature (with each list being a unique value), or apply one-hot-encoding to the data. After speaking with Professor Abdou about the pros and cons, I decided to apply one-hot-encoding. I don’t think there is a better solution that could have been applied; however, having done the one-hot-encoding, I believe the “Activity Type” data lost some importance and if there had been a single “Activity Type” for each incident report, then that feature likely would have been much more useful in the prediction models.

I also found it difficult to work with so many categorical features, and to work with a multi-class classification problem. From my experience in our course work and assignments, I had never dealt with a dataset containing so many missing values, such imbalanced data, multi-class target variable, and categorical data. Because of this, I spent a lot of time learning and figuring things out and put an incredible amount of time into this analysis. I have absolutely learned so much through this invaluable experience.

Concluding Remarks

In a personal communication (February 3, 2023), D. Gummer, Wildlife Management Specialist from Parks Canada, shared that so far, this data has been used to build an internal system that staff and management of the Parks can use to view and analyze incident data for their day-to-day operations. “The data and ongoing analyses are also helping to inform new national policy/guidance that [they] are working on and many more that [they] propose for [the] future;” however there is not yet a public report that has been written to summarize the results of

this data collection or how it has been applied in policies. Gummer shared that individual parks also use the data to develop their own local policies/reports, but there was no clear line from the data to the specific policies it has addressed.

My research aimed to discover and highlight trends in coexistence incidents over time and across geographic locations and aimed to find correlations between attributes that can predict incidents and identify target areas where improved protection is needed. Unfortunately, with the data available, finding accurate correlations and predictions has proven difficult. With that said, my recommendations for Parks Canada relate to the data collection (so we can obtain more information that could assist in predicting and therefore preventing incidents), increased resources in high incident areas, and public education. My recommendations are here:

1. As we've seen, the most common incident is "Human Wildlife Interaction". This incident is defined in the data description .csv file (provided with the data) as "A negative interaction between wildlife and people and/or their property; whether major or minor; with or without physical contact". I would love to see this incident divided into a few more specific incident types such as "Human Wildlife Interaction – People - Physical Contact", "Human Wildlife Interaction – People – No Physical Contact", and "Human Wildlife Interaction – Property". Perhaps even including a separate feature for "Major" or "Minor" incident that could be used for all Incident Types. I believe splitting up the major class of incidents would be extremely useful for differentiating between these possible interactions and would provide a better base for prediction and prevention.

2. Have each incident report relate to a single Activity. If more than one activity was conducted in which incidents occurred, have each one reported as a distinct Activity.

3. With most incidents occurring in Banff National Park of Canada and Jasper National Park of Canada, it would be prudent to focus attention to those Parks and Field Units (designated Parks Canada offices). Place more Park Staff in those areas and enforcing further restricted areas where most incidents occur within those parks during peak months.
4. Create additional educational resources for the public and Park Staff regarding best practices on how to avoid negative encounters with wildlife, and what to do if they do occur (particularly for the 4 species who are involved in the most incidents: Black Bear, Elk, Grizzly Bear, and Mule Deer).

I've spent the last 7 years working for an environmental charity and this topic is very close to home for me. There is a constant push-and-pull between the desire to be in nature exploring and witnessing wildlife and the need protect and preserve these spaces. Most people will say it's the experiences they've had out in nature that have truly helped them understand the great importance of protecting the natural world. This is why it's so important to ensure that we are able to balance these two needs – being able to enjoy the natural world, while also taking every opportunity given to us to protect it. We must continue strive to improve our coexistence with wildlife for all of our benefits.

I feel grateful to have been able to choose a dataset that is close to my hear. While I was not able to go as far as I'd hoped with identifying concrete target areas for mitigating incidents and identifying causes, I believe this analysis has successfully identified the target areas for the data that we are missing and that would help us be able to better predict and mitigate incidents in the future.

References

Baral, K., Sharma, H. P., Rimal, B., Thapa-Magar, K., Bhattarai, R., Kunwar, R. M., Aryal, A.,& Ji, W. (2021). Characterization and management of human-wildlife conflicts in mid-hills outside protected areas of Gandaki province, Nepal. Plos One. https://doi.org/10.1371/journal.pone.0260307

Bex, T. (2021, July 17). How to Remove Non-Stationarity in Time Series Forecasting. Towards Data Science. https://towardsdatascience.com/how-to-remove-non-stationarity-in-time-series-forecasting-563c05c4bfc7

Boyle, T. (2019, February 3). Dealing with Imbalanced Data. Towards Data Science. https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18

Brownlee, Jason. (2020, August 18). How to Perform Feature Selection with Categorical Data. Machine Learning Mastery. https://machinelearningmastery.com/feature-selection-with-categorical-data/

Brownlee, Jason. (2021, March 17). SMOTE for Imbalanced Classification with Python. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

Government of Canada. (2022, November 19). *The Parks Canada mandate and charter*. https://parks.canada.ca/agence-agency/mandat-mandate

Gummer, D., & Nicholl, S. (2022, September 15). *Human-wildlife coexistence incidents in selected national parks from 2010 to 2021*. Government of Canada. https://open.canada.ca/data/en/dataset/cc5ea139-c628-46dc-ac55-a5b3351b7fdf

Howell, Egor. (2023, January 10). How To Correctly Perform Cross-Validation For Time Series. Towards Data Science. https://towardsdatascience.com/how-to-correctly-perform-cross-validation-for-time-series-b083b869e42c

Naha, D., Dash, S. K., Chettri, A., Chaudhary, P., Sonker, G., Heurich, M., Rawat, G. S. & Sathyakumar, S. (2020). Landscape predictors of human-leopard conflicts within multi-use areas of the Himalayan region. Scientific Reports: nature research, 10(11129). | https://doi.org/10.1038/s41598-020-67980-w

Radečić, Dario. (2020, January 11). What is Stationarity in Time Series and why should you care. Towards Data Science. https://towardsdatascience.com/what-is-stationarity-in-time-series-and-why-should-you-care-f3b45082356b

Statology. (2021, May 25). Augmented Dickey-Fuller Test in Python (With Example). Statistics Simplified: Statology. https://www.statology.org/dickey-fuller-test-python/