Aircraft Wildlife Strikes

Amrita Chandrasekar

NH53017

Outline

- Introduction
- Problem statement
- Data collection
- Data preprocessing
- Exploratory data analysis
- Machine learning models
- Model evaluation results
- Challenges
- Future scope
- Conclusion
- Key takeaways

Introduction

Aim

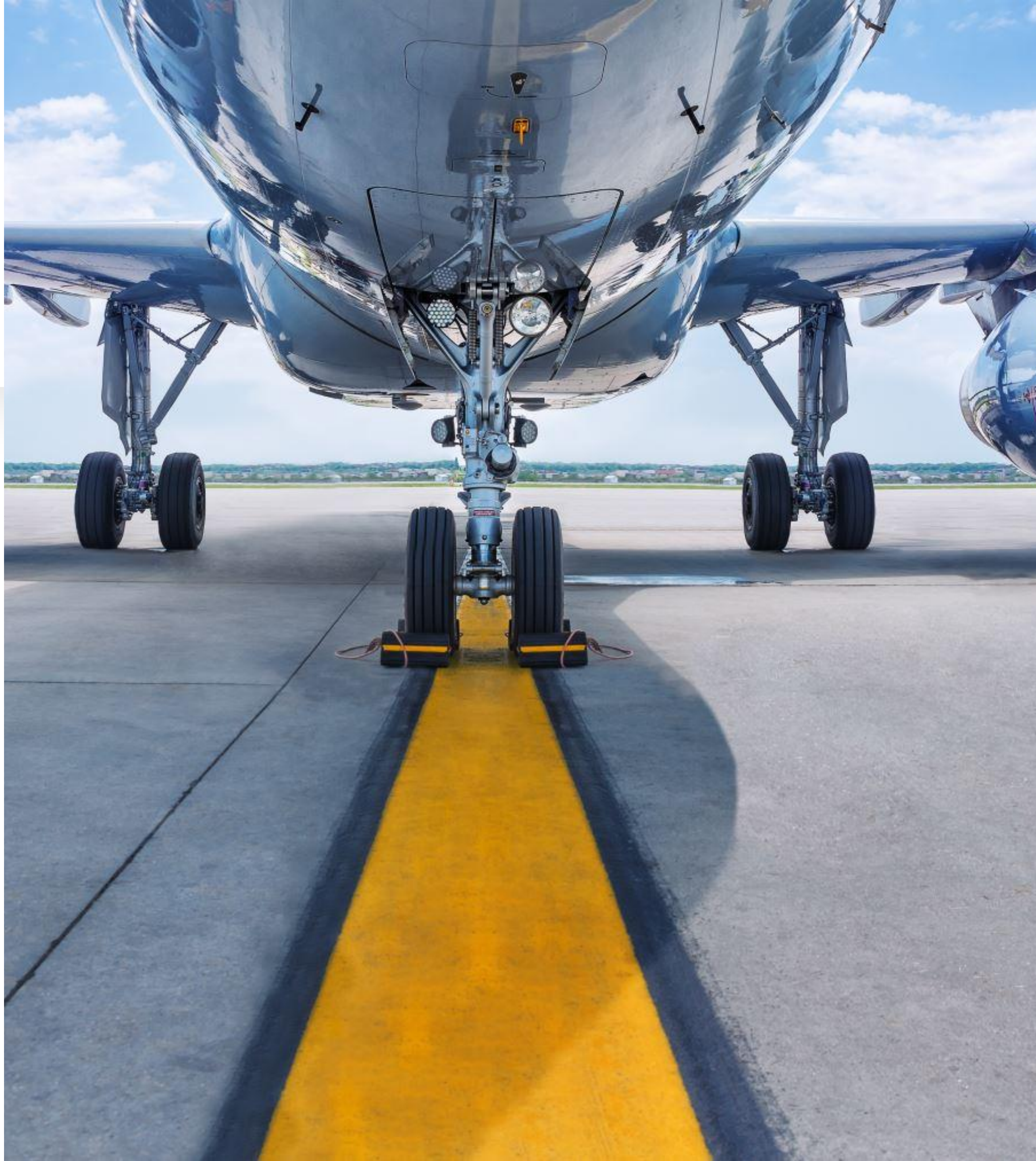
This project investigates the significant global issue of wildlife strikes on aircraft, which poses a risk to both aviation safety and wildlife conservation. The analysis seeks to understand patterns in these incidents to inform better preventative measures.

Goals and Objectives

- **To understand the frequency and patterns of wildlife strikes on aircraft**:
  - Analyze historical data to determine trends over time.
  - Identify peak times of the year, day, or specific conditions when wildlife strikes are most likely to occur.
- **To identify the species that are most commonly involved in strikes:**
  - Determine which species pose the greatest risk to aircraft.
  - Understand the behavior and habitats of these species to inform mitigation strategies.
- **To develop predictive models that can estimate the likelihood and potential severity of wildlife strikes:**
  - Utilize machine learning techniques to predict the damage level based on historical data.
  - Create a model to forecast the cost implications of wildlife strikes on aircraft repairs.

Problem Statement

Wildlife strikes pose risks such as aircraft damage, passenger safety hazards, and potential loss of life. This project aims to analyze and predict these incidents to enhance aviation safety measures.



Data Source and Collection

- **Kaggle**

https://www.kaggle.com/datasets/dianaddx/aircraft-wildlife-strikes-1990-2023

About Dataset

- Size:195MB
- Time frame:1990-2023
- Rows:288810
- Columns:100

DataPreprocessing

The preprocessing involved the following:

- Removing duplicates
- Filling missing values with
    - Medians for numerical columns, and
    - Modes for categorical columns

In [9]:
```
strike_data.isna()
```
Out[9]:

	INCIDENT_DATE	INCIDENT_MONTH	INCIDENT_YEAR	TIME	TIME_OF_DAY	AIRPORT_ID	AIRPORT	LATITUDE	LON
0	False	False	False	True	True	False	False	False	
1	False	False	False	True	True	False	False	False	
2	False	False	False	True	True	False	False	False	
3	False	False	False	True	True	False	False	False	
4	False	False	False	True	True	False	False	False	
...	...	...	...	...	...	...	...	...	
288805	False	False	False	False	True	False	False	True	
288806	False	False	False	False	True	False	False	True	
288807	False	False	False	False	False	False	False	False	
288808	False	False	False	False	False	False	False	False	
288809	False	False	False	False	True	False	False	False	

288810 rows × 35 columns

In [10]:
```
strike_data.isna().sum()
```
Out[10]:
```
INCIDENT_DATE            0
INCIDENT_MONTH           0
INCIDENT_YEAR            0
TIME                125177
TIME_OF_DAY         122416
AIRPORT_ID               6
AIRPORT                  0
LATITUDE             35501
LONGITUDE            35502
RUNWAY               69958
STATE                35501
LOCATION            254871
ENROUTE_STATE       283773
OPID                    26
OPERATOR                 0
AIRCRAFT                 0
PHASE_OF_FLIGHT     110967
HEIGHT              140003
SPEED               194573
DISTANCE            100576
SKY                 149428
PRECIPITATION       277635
AOS                 274614
COST_REPAIRS        283770
INDICATED_DAMAGE         0
DAMAGE_LEVEL        101495
BIRD_BAND_NUMBER    288420
SPECIES_ID               4
SPECIES                  1
REMARKS              24505
SIZE                 25272
NR_INJURIES         288534
NR_FATALITIES       288786
COMMENTS             69976
REPORTED_NAME            0
```


In [12]:
```
strike_data.drop_duplicates(inplace=True)
```

In [13]:
```
strike_data.duplicated().sum()
```
Out[13]:
```
0
```

In [14]:
```
# Fill missing values in numeric columns with the mean
numeric_columns = strike_data.select_dtypes(include='number').columns
strike_data[numeric_columns] = strike_data[numeric_columns].fillna(strike_data[numeric_colu

# Fill missing values in categorical columns with the most frequent value
categorical_columns = strike_data.select_dtypes(exclude='number').columns
strike_data[categorical_columns] = strike_data[categorical_columns].apply(lambda x: x.filln
```

In [15]:
```
strike_data.isna().sum()
```
Out[15]:
```
INCIDENT_DATE       0
INCIDENT_MONTH      0
INCIDENT_YEAR       0
TIME                0
TIME_OF_DAY         0
AIRPORT_ID          0
AIRPORT             0
LATITUDE            0
LONGITUDE           0
RUNWAY              0
STATE               0
LOCATION            0
ENROUTE_STATE       0
OPID                0
OPERATOR            0
AIRCRAFT            0
PHASE_OF_FLIGHT     0
HEIGHT              0
SPEED               0
DISTANCE            0
SKY                 0
PRECIPITATION       0
AOS                 0
COST_REPAIRS        0
INDICATED_DAMAGE    0
DAMAGE_LEVEL        0
BIRD_BAND_NUMBER    0
SPECIES_ID          0
SPECIES             0
REMARKS             0
SIZE                0
NR_INJURIES         0
NR_FATALITIES       0
COMMENTS            0
REPORTED_NAME       0
dtype: int64
```

EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis

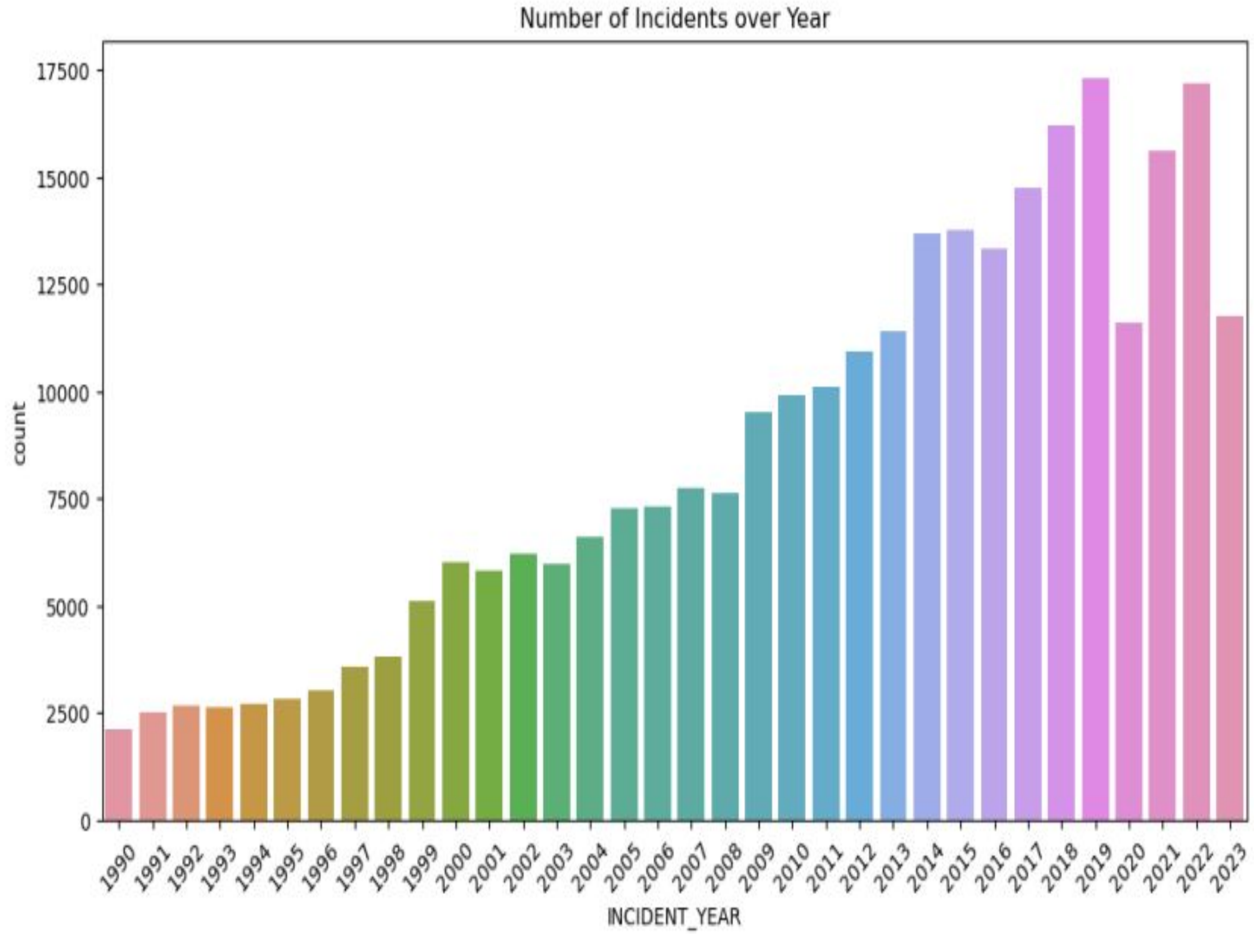
The EDA provides the below insights:

- Incidents per year (Bar Chart)
- Distribution of wildlife strikes by part of the day (Pie Chart)
- Incidents by state (Bar Chart)
- Top 10 species involved in strikes (Bar Chart)
- Number of injuries and fatalities by aircraft type (Bar Charts)
- Correlation matrix for numerical columns (Heatmap)

Exploratory Data Analysis

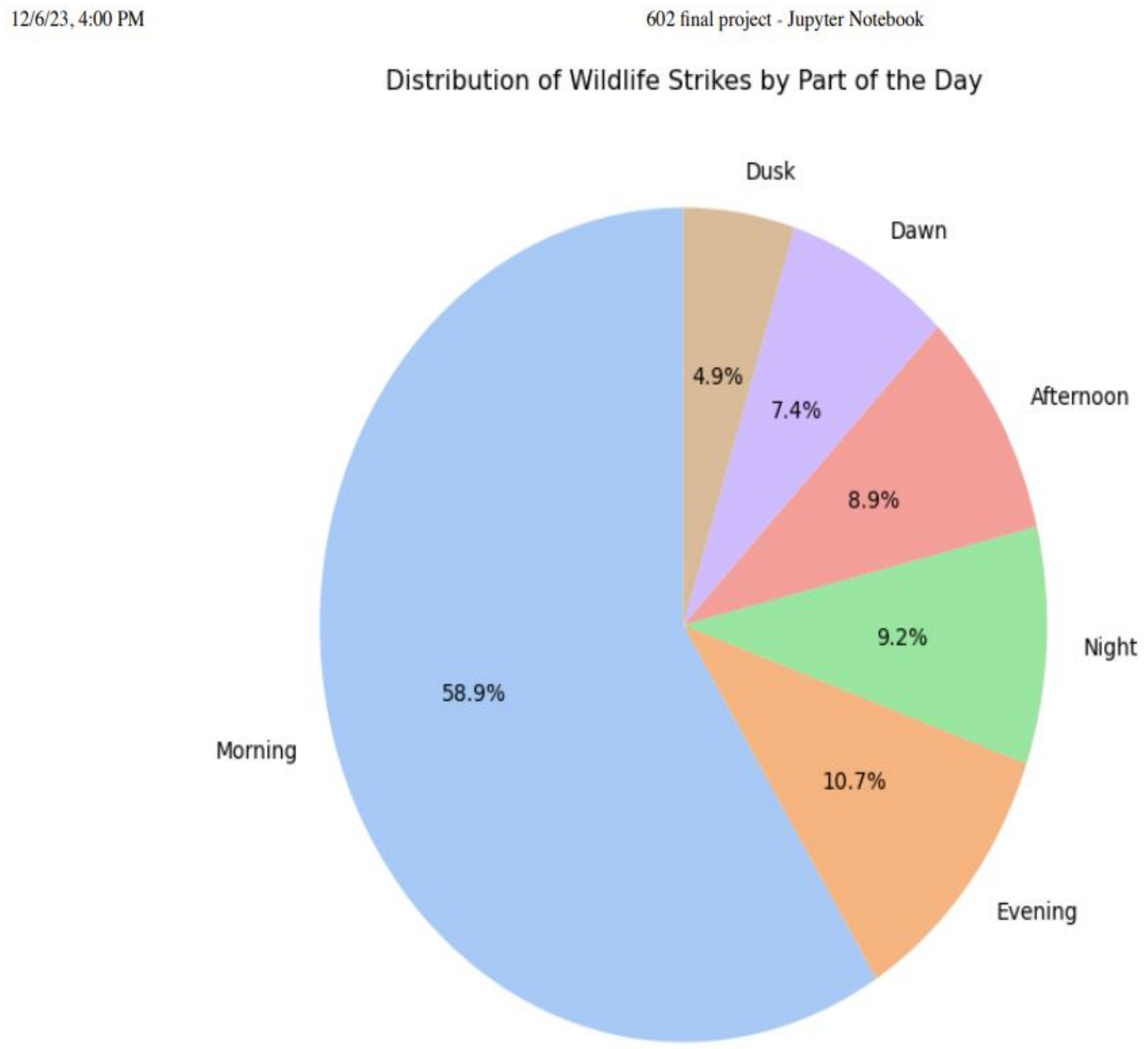
- Incidents per year (Bar Chart)

In [16]:
```
#Plotting the number of incidents per year
plt.figure(figsize=(12, 6))
sns.countplot(x='INCIDENT_YEAR', data=strike_data)
plt.xticks(rotation=45)
plt.title('Number of Incidents over Year')
plt.show()
```



Exploratory Data Analysis

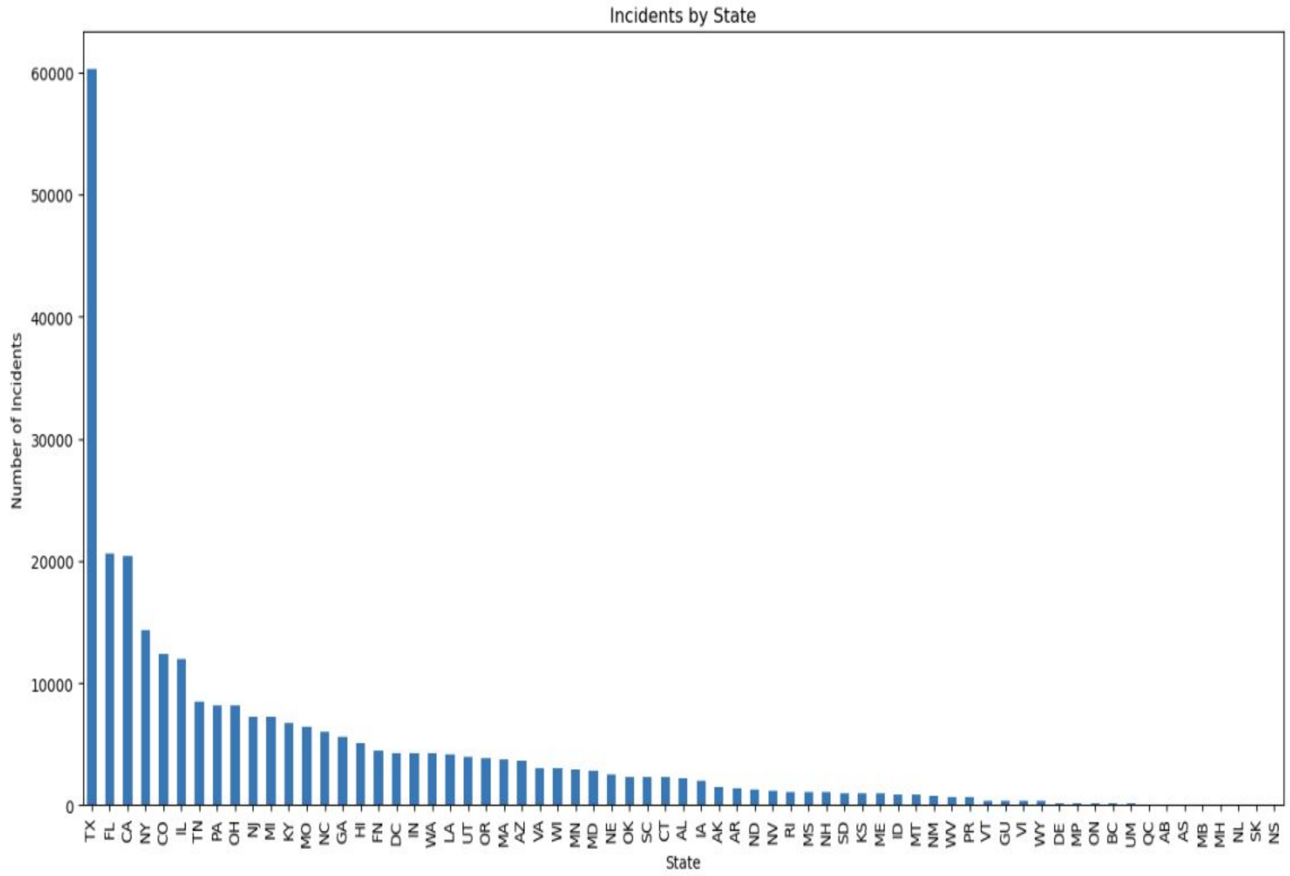
- Distribution of wildlife strikes by part of the day (Pie Chart)



Exploratory Data Analysis

- Incidents by state (Bar Chart)

In [18]:
```
# Aggregating incidents by state
state_incidents = strike_data['STATE'].value_counts()
# Increase the figure size
plt.figure(figsize=(16, 8))
state_incidents.plot(kind='bar')
plt.title('Incidents by State')
plt.xlabel('State')
plt.ylabel('Number of Incidents')
plt.show()
```



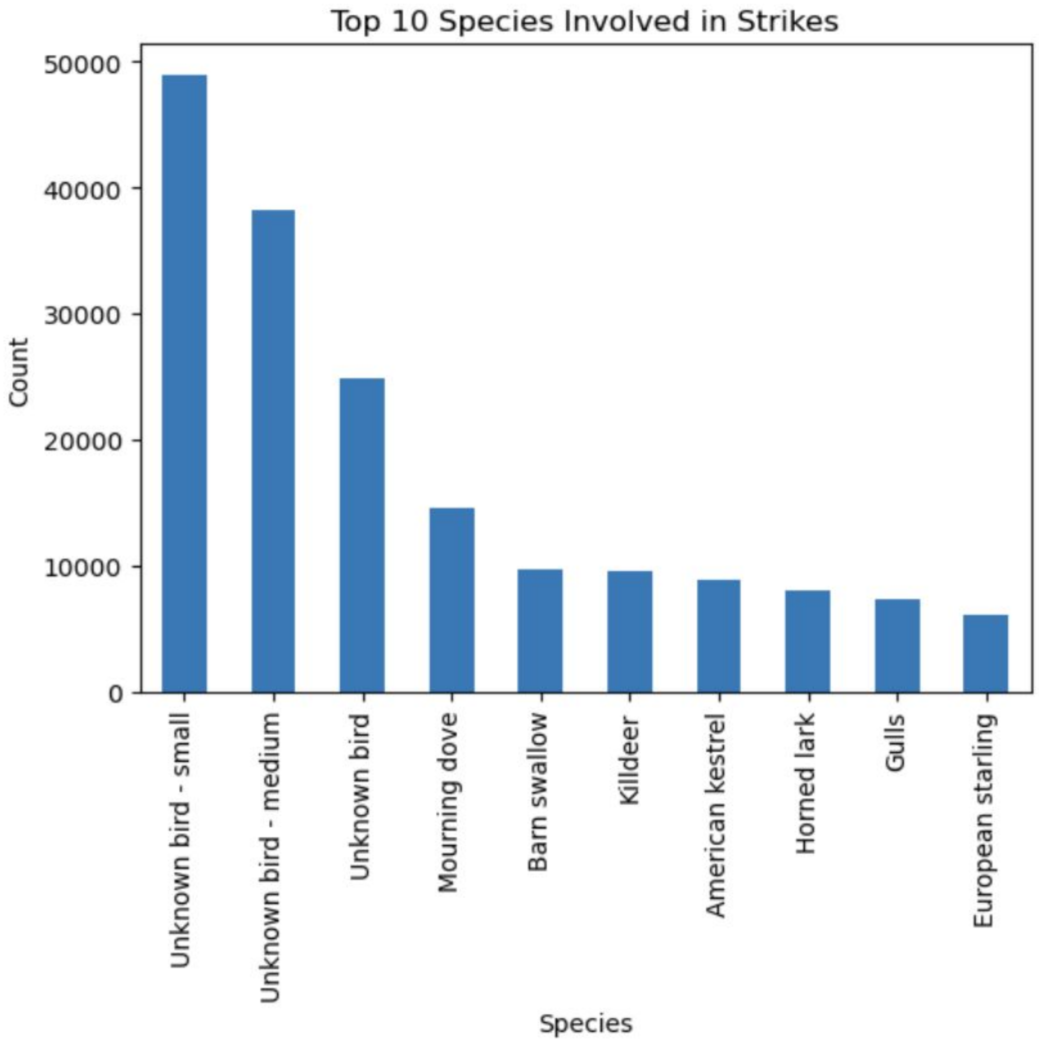
Exploratory Data Analysis

- Top 10 species involved in strikes (Bar Chart)

12/6/23, 4:00 PM 602 final project - Jupyter Notebook

In [22]:

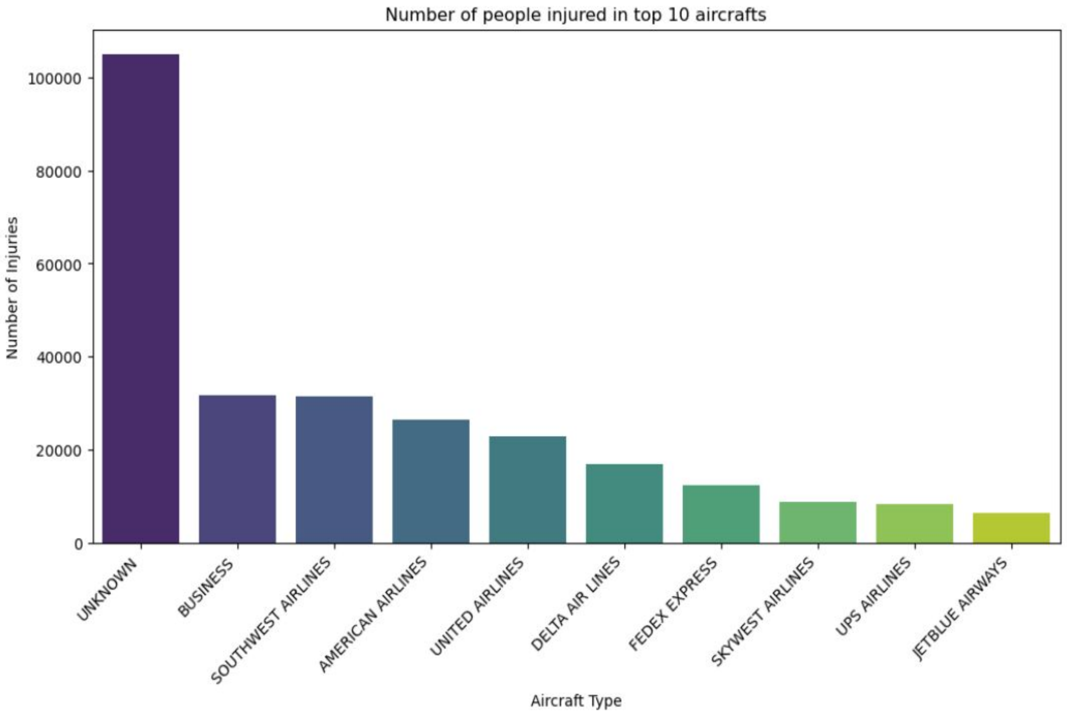
```
# Species involved in strikes
species_counts = strike_data['SPECIES'].value_counts().head(10)
species_counts.plot(kind='bar')
plt.title('Top 10 Species Involved in Strikes')
plt.xlabel('Species')
plt.ylabel('Count')
plt.show()
```



Exploratory Data Analysis

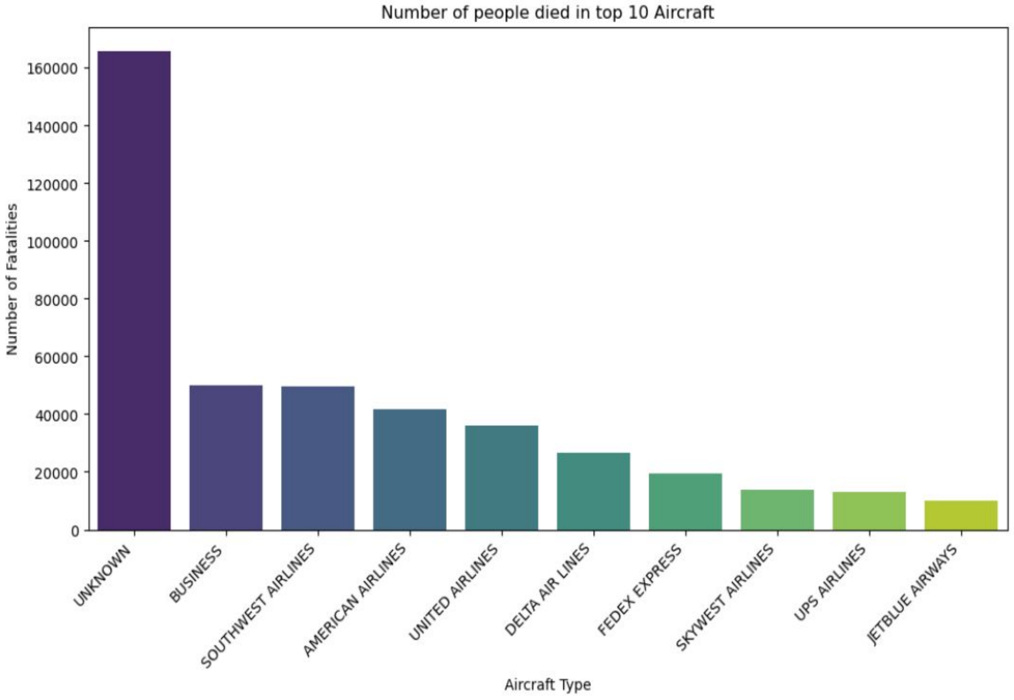
- Number of injuries and fatalities by aircraft type (Bar Charts)

```
plt.ylabel('Number of Fatalities')
plt.xticks(rotation=45, ha='right')
plt.show()
```



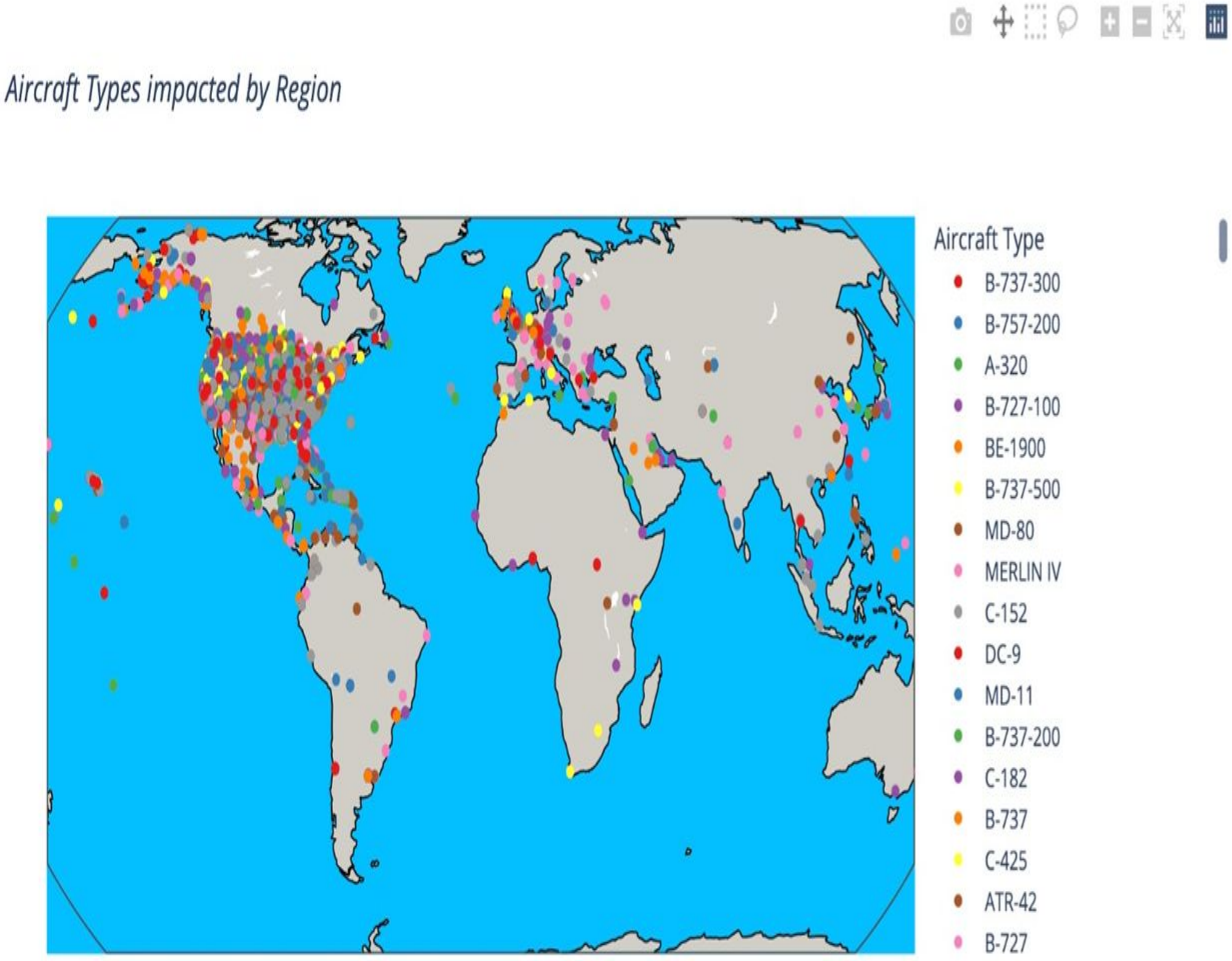
12/6/23, 4:00 PM

602 final project - Jupyter Notebook



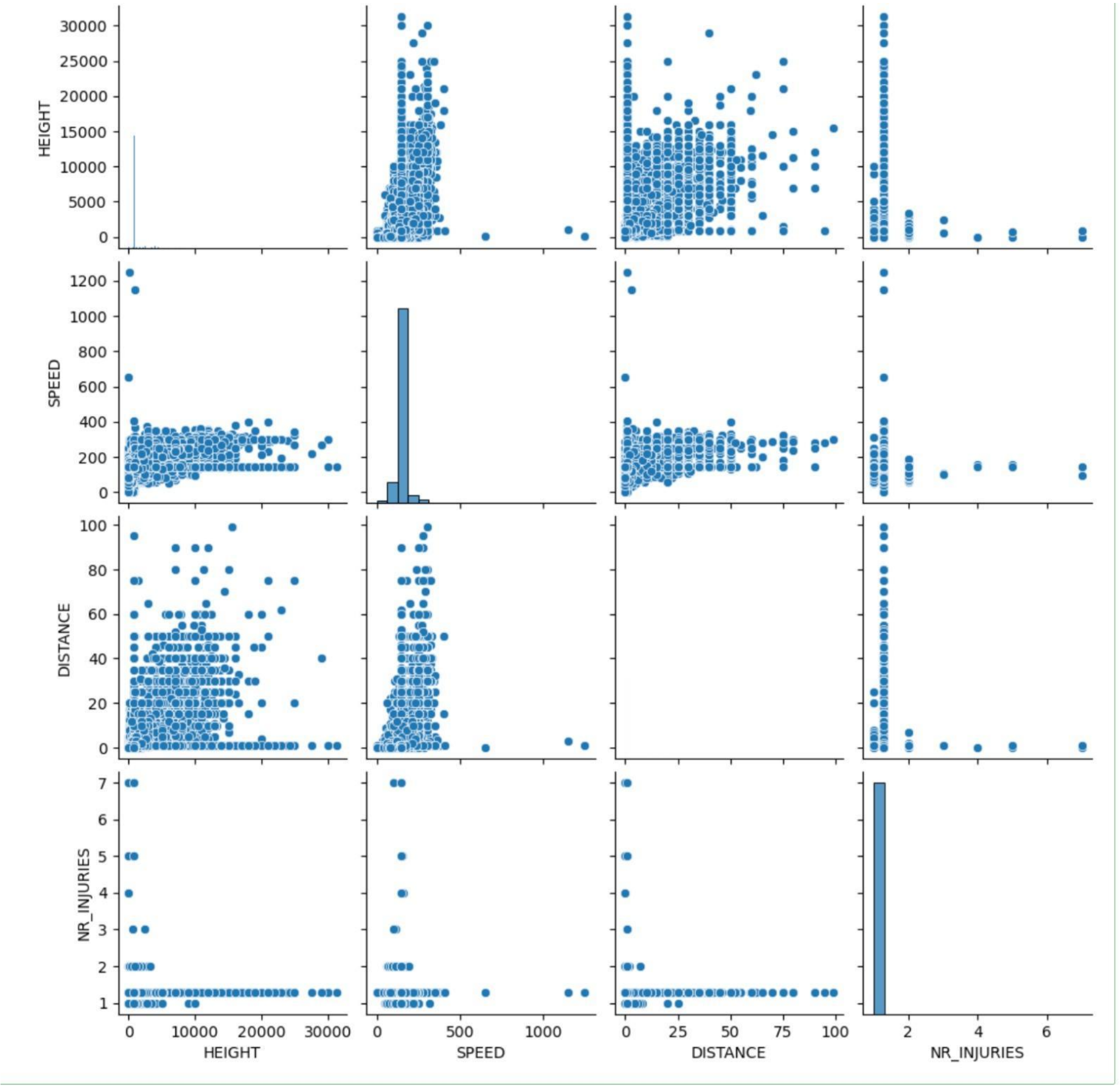
Exploratory Data Analysis

- Aircrafts impacted by region



Exploratory Data Analysis

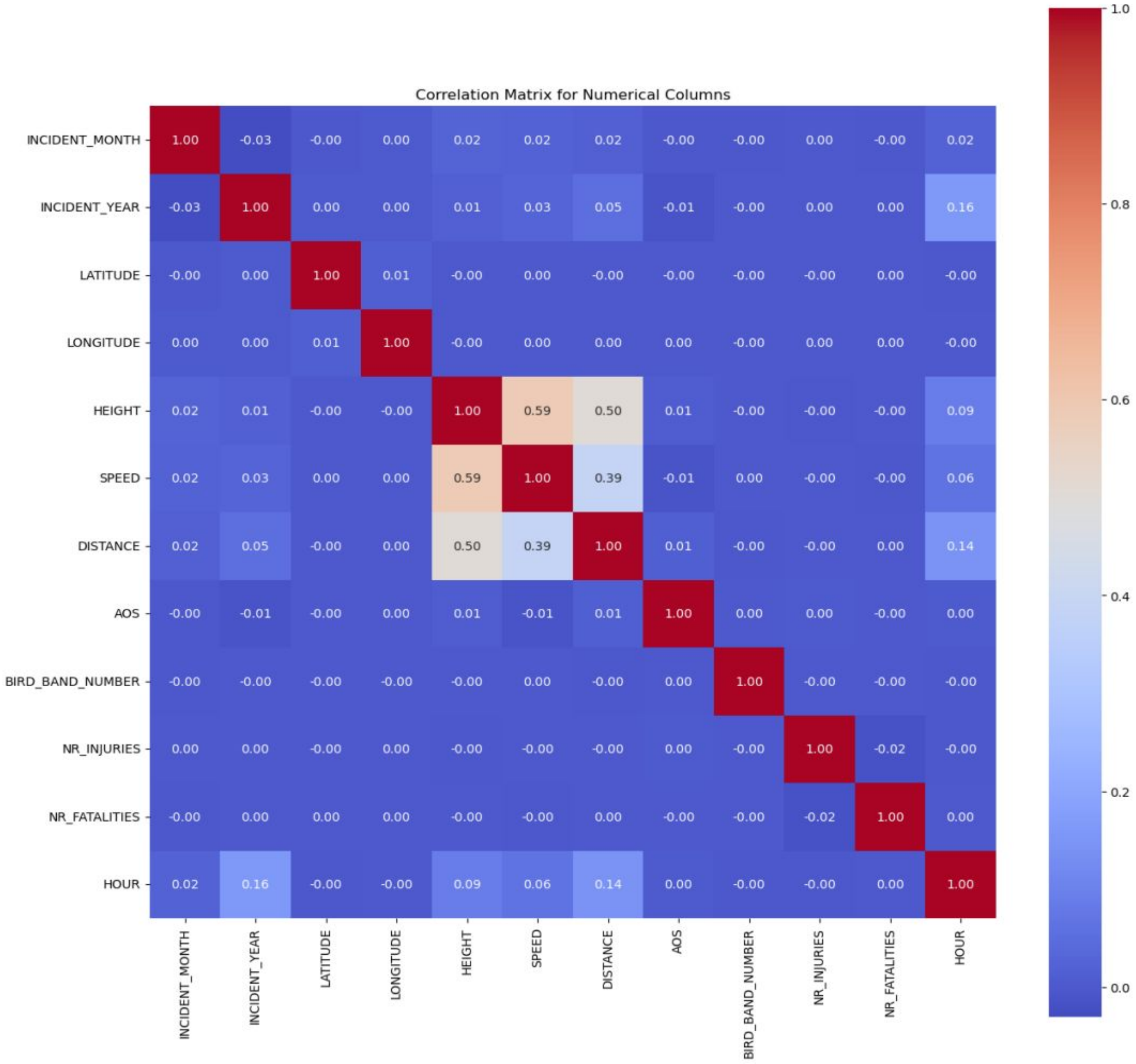
- Pair plot between numerical columns

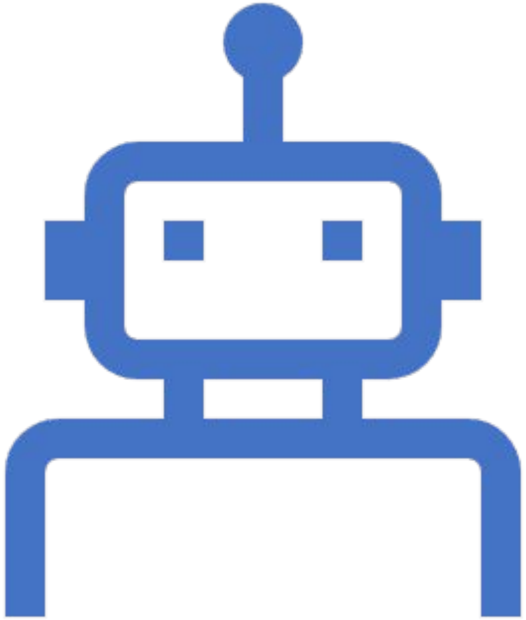


```
plt.show()
```

Exploratory Data Analysis

- Correlation matrix for numerical columns (Heatmap)





Machine Learning Models

Classification:[Predicting Damage Level]

- Random Forest Classifier
- XGBoost Classifier
- Logistic Regression Classifier

Regression:[Predicting Cost Repair]

- Gradient Boosting Regressor
- Linear Regression
- Decision Tree Regressor

Clustering:[Determining Relationship B/W Damage And Cost Repair]

- K-Means

Model Evaluation Results

- Classification model’ s performance was assessed using accuracy, precision, recall, and f1-scores
- Regression models were evaluated based on Mean Squared Error (MSE) and R-squared values.

Model Evaluation Results

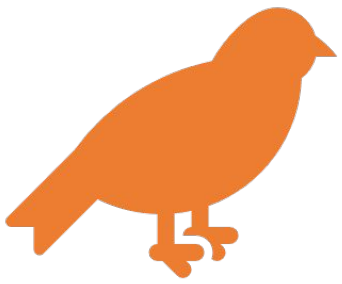
```
Random Forest Classifier:
Accuracy: 0.6049178499739797
              precision    recall  f1-score   support

           0       0.61      0.90      0.73     53798
           1       0.55      0.36      0.44     23419
           2       0.63      0.30      0.41     18561
           3       0.62      0.24      0.35     11586
           4       0.88      0.52      0.66       244

    accuracy                           0.60    107608
   macro avg       0.66      0.46      0.51    107608
weighted avg       0.60      0.60      0.57    107608

XGBoost Classifier:
Accuracy: 0.5614359527172701
              precision    recall  f1-score   support

           0       0.58      0.90      0.71     53798
           1       0.45      0.31      0.37     23419
           2       0.54      0.18      0.27     18561
           3       0.64      0.11      0.19     11586
           4       0.90      0.43      0.58       244

    accuracy                           0.56    107608
   macro avg       0.62      0.39      0.42    107608
weighted avg       0.55      0.56      0.50    107608

Logistic Regression Classifier:
Accuracy: 0.5016820310757565
              precision    recall  f1-score   support

           0       0.51      0.98      0.67     53798
           1       0.32      0.03      0.06     23419
           2       0.32      0.03      0.05     18561
           3       0.00      0.00      0.00     11586
           4       0.40      0.01      0.02       244

    accuracy                           0.50    107608
   macro avg       0.31      0.21      0.16    107608
weighted avg       0.38      0.50      0.36    107608
```

Model	MSE	R-squared
Linear Regression	5.44511e+09	0.0227
Decision Tree Regression	2.76398e+10	-3.9608
Gradient Boosting Regression	2.40268e+10	-3.3123

Challenges

- Numerical and categorical data
- Imbalanced classes in classification tasks- model sampling technique
- Low R-squared values in regression - hyperparameter tuning technique

Future Scope



Enhance bird control measures and prevent bird strikes

Integration with Weather Data

Conclusion

- Utilized machine learning to analyze and predict wildlife strikes on aircraft, enhancing aviation safety.
- Advanced data analysis revealed key patterns, informing strategies to mitigate future incidents.
- Provided valuable insights for wildlife strike prevention, demonstrating the effectiveness of data-driven approaches.

KEY TAKEAWAYS

- Machine Learning Efficiency: Showcased machine learning's capability in predicting wildlife strikes.
- Data-Driven Insights: Uncovered vital patterns for strategic strike prevention.
- Data Quality Emphasis: Stressed the importance of thorough data preprocessing.
- Room for Enhancement: Identified opportunities for improving data balance and model accuracy.
- Real-World Impact: Demonstrated practical applications in aviation safety and wildlife management.

References

(1)Altringer, L., Navin, J., Begier, M. J., Shwiff, S. A., & Anderson, A. (2021). Estimating wildlife strike costs at US airports: A machine learning approach. Transportation Research Part D: Transport and Environment, 97, 102907. https://doi.org/10.1016/j.trd.2021.102907

(2)Wei, K., Liu, W., Sui, Y., Guo, W., & Yu, G. (2022). Motion detection of bird based on optical radar. 2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT). https://doi.org/10.1109/iccasit55263.2022.9986743(Future Scope)

(3)Andrews, R., Bevrani, B., Colin, B., Wynn, M. T., Ter Hofstede, A. H. M., & Ring, J. (2022). Three novel bird strike likelihood modelling techniques: The case of Brisbane Airport, Australia. PLOS ONE, 17(12), e0277794. https://doi.org/10.1371/journal.pone.0277794(Future Scope)

Thank you