pDSC 680 Applied Data Science

Chitramoy Mukherjee, T301 T302 2253 Winter 2025

Project-2 Milestone-3

Wildlife Protection of Endangered Species

Description:

An endangered species is a type of organism that is threatened by extinction. Species become endangered for two main reasons: loss of habitat and loss of genetic variation. The conservation of endangered and rare wildlife is a primary part of preserving biodiversity. The purpose of this project is to intelligently monitor endangered and rare wildlife species by using Machine Learning models with animal data and computational resources from various parks in United States.

Problem Statement:

The rapid decline in populations of endangered species globally poses a critical threat to biodiversity and the stability of ecosystems. Traditional methods of monitoring and protecting these species are often labor-intensive, time-consuming, and prone to inaccuracies. There is an urgent need for innovative, scalable, and efficient solutions to identify, monitor, and protect endangered species effectively. Leveraging advancements in Artificial Intelligence (AI) and Machine Learning (ML), we propose developing a sophisticated AI/ML-based system to enhance the efforts of wildlife conservationists, researchers, and policymakers in safeguarding endangered species.

Objectives:

1. **Identification:** Develop AI/ML models capable of accurately identifying endangered species in their natural habitats using camera traps, drones, and satellite imagery.
2. **Monitoring:** Implement AI-powered real-time monitoring systems to track the movements, behaviors, and population dynamics of endangered species, providing valuable data for conservation efforts.
3. **Threat Detection:** Utilize machine learning algorithms to predict and identify potential threats to endangered species, such as poaching, habitat loss, and climate change impacts, allowing for timely and targeted interventions.
4. **Data Integration:** Create a centralized platform to integrate and analyze data from various sources, including field observations, sensor networks, and citizen science initiatives, to provide comprehensive insights into the status of endangered species.

5. **Policy Support:** Generate actionable insights and recommendations for policymakers and conservation organizations to formulate and implement effective conservation strategies based on data-driven evidence.

By harnessing the power of AI/ML, this project aims to revolutionize wildlife protection efforts, ensuring the survival and flourishing of endangered species for future generations.

Datasets and Data Dictionary:

Here I am using 3 datasets as below:

File: final_species.csv

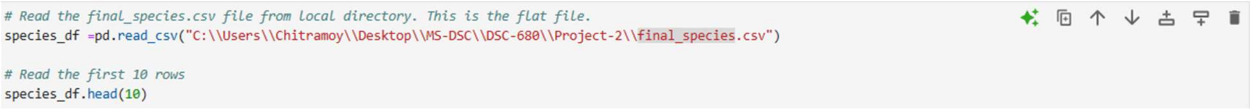
Source: Abigail Larion(n.d.), https://www.kaggle.com/datasets/nationalparkservice/park-biodiversity/data?select=species.csv

In the final_species.csv we have added 2 additional derived column is_float and first_common_name for the project purpose.

final_species.csv

The species dataset contains plants and animal species information from different national parks of United States. This dataset contains 13 columns.

Column Name	**Data Type**	**Description**
Species ID	String	Unique ID. National Park service code for each species
Park Name	String	Park Name in which the species appears.
Category	String	Category of species like mammal, bird etcetera.
Order	String	The scientific order the species belongs to.
Family	String	The scientific family the species belongs to.
Scientific Name	String	Full scientific species name.
Common Names	String	Usual common name(s) for the species. Comma-delimited.
Record Status	String	record status from park
Occurrence	String	Current presence of that species
Nativeness	String	Whether the species is native to the area or a non-native/invasive.
Abundance	String	Commonality of sightings.
Seasonality	String	When the species can be found in the park. Blank if the species is found there year-round.
Conservation Status	String	IUCN species conservation status.



```
# Read the final_species.csv file from local directory. This is the flat file.
species_df =pd.read_csv("C:\\Users\\Chitramoy\\Desktop\\MS-DSC\\DSC-680\\Project-2\\final_species.csv")

# Read the first 10 rows
species_df.head(10)
```

Website: National Park locations and areas

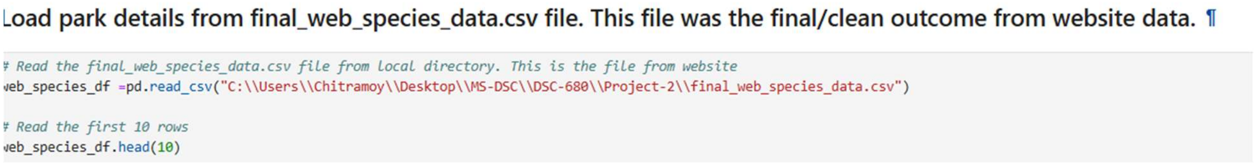
Source: https://irma.nps.gov/NPSpecies

Created the final_web_species_data with required columns from park_name, state, location, date_established, area, recreation_visitors, description and established_date.

final_web_species_data.csv

This data source will be used to get parks information as below. Further information is provided under "Project Approach" section on how this data will be extracted.

Column Name	Data Type	Description
Park Code	String	Unique value. National Park code
Park Name	String	Name of the park.
State	String	US state(s) in which the park is located. Comma-separated.
Acres	Integer	Size of the park in acres.
Latitude	Integer	Latitude of the park (centroid).
Longitude	Integer	Longitude of the park (centroid).



Load park details from final_web_species_data.csv file. This file was the final/clean outcome from website data. ¶

```
# Read the final_web_species_data.csv file from local directory. This is the file from website
web_species_df =pd.read_csv("C:\\Users\\Chitramoy\\Desktop\\MS-DSC\\DSC-680\\Project-2\\final_web_species_data.csv")

# Read the first 10 rows
web_species_df.head(10)
```

API: Animals API

Source: [Animals API - API Ninjas (api-ninjas.com)](https://api-ninjas.com)

The Animals API provides interesting scientific facts on thousands of different animal species.

/v1/animals

HTTP GET: Returns up to 10 results matching the input name parameter.

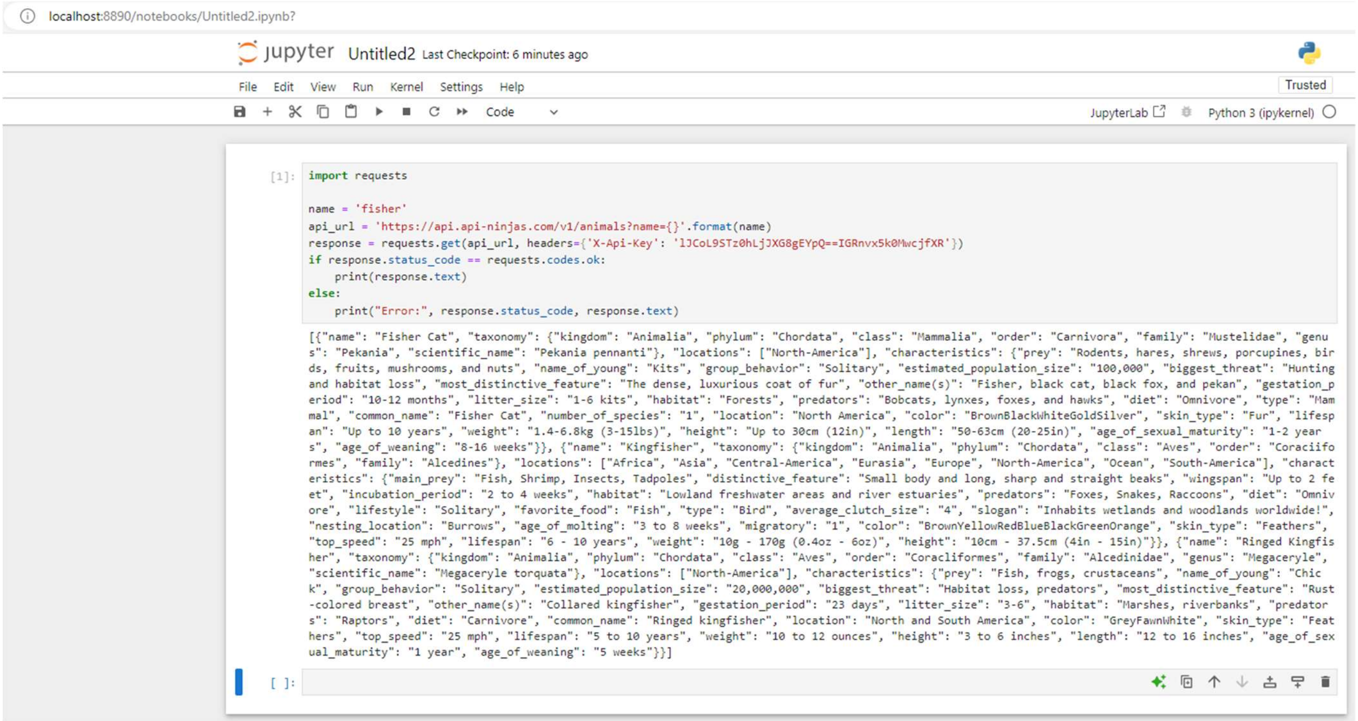
Parameters: name (required) - common name of animal to search. This parameter supports partial matches (e.g. fox will match gray fox and red fox)

Headers: X-Api-Key (required) - lJCoL9STz0hLjJXG8gEYpQ==IGRnvx5k0MwcjfXR

Example API call:

Using Python script in Jupyter Notebook.

```
import requests
name = 'fisher'
api_url = 'https://api.api-ninjas.com/v1/animals?name={}'.format(name)
response = requests.get(api_url, headers={'X-Api-Key':
'lJCoL9STz0hLjJXG8gEYpQ==IGRnvx5k0MwcjfXR'})
if response.status_code == requests.codes.ok:
    print(response.text)
else:
    print("Error:", response.status_code, response.text)
```



```
import requests

name = 'fisher'
api_url = 'https://api.api-ninjas.com/v1/animals?name={}'.format(name)
response = requests.get(api_url, headers={'X-Api-Key': 'lJCoL9STz0hLjJXG8gEYpQ==IGRnvx5k0MwcjfXR'})
if response.status_code == requests.codes.ok:
    print(response.text)
else:
    print("Error:", response.status_code, response.text)
```

```
[{"name": "Fisher Cat", "taxonomy": {"kingdom": "Animalia", "phylum": "Chordata", "class": "Mammalia", "order": "Carnivora", "family": "Mustelidae", "genu
s": "Pekania", "scientific_name": "Pekania pennanti"}, "locations": ["North-America"], "characteristics": {"prey": "Rodents, hares, shrews, porcupines, bir
ds, fruits, mushrooms, and nuts", "name_of_young": "Kits", "group_behavior": "Solitary", "estimated_population_size": "100,000", "biggest_threat": "Hunting
and habitat loss", "most_distinctive_feature": "The dense, luxurious coat of fur", "other_name(s)": "Fisher, black cat, black fox, and pekan", "gestation_p
eriod": "10-12 months", "litter_size": "1-6 kits", "habitat": "Forests", "predators": "Bobcats, lynxes, foxes, and hawks", "diet": "Omnivore", "type": "Mam
mal", "common_name": "Fisher Cat", "number_of_species": "1", "location": "North America", "color": "BrownBlackWhiteGoldSilver", "skin_type": "Fur", "lifesp
an": "Up to 10 years", "weight": "1.4-6.8kg (3-15lbs)", "height": "Up to 30cm (12in)", "length": "50-63cm (20-25in)", "age_of_sexual_maturity": "1-2 year
s", "age_of_weaning": "8-16 weeks"}}, {"name": "Kingfisher", "taxonomy": {"kingdom": "Animalia", "phylum": "Chordata", "class": "Aves", "order": "Coraciifo
rmes", "family": "Alcedines"}, "locations": ["Africa", "Asia", "Central-America", "Eurasia", "Europe", "North-America", "Ocean", "South-America"], "charact
eristics": {"main_prey": "Fish, Shrimp, Insects, Tadpoles", "distinctive_feature": "Small body and long, sharp and straight beaks", "wingspan": "Up to 2 fe
et", "incubation_period": "2 to 4 weeks", "habitat": "Lowland freshwater areas and river estuaries", "predators": "Foxes, Snakes, Raccoons", "diet": "Omniv
ore", "lifestyle": "Solitary", "favorite_food": "Fish", "type": "Bird", "average_clutch_size": "4", "slogan": "Inhabits wetlands and woodlands worldwide!",
"nesting_location": "Burrows", "age_of_molting": "3 to 8 weeks", "migratory": "1", "color": "BrownYellowRedBlueBlackGreenOrange", "skin_type": "Feathers",
"top_speed": "25 mph", "lifespan": "6 - 10 years", "weight": "10g - 170g (0.4oz - 6oz)", "height": "10cm - 37.5cm (4in - 15in)"}}, {"name": "Ringed Kingfis
her", "taxonomy": {"kingdom": "Animalia", "phylum": "Chordata", "class": "Aves", "order": "Coracliformes", "family": "Alcedinidae", "genus": "Megaceryle",
"scientific_name": "Megaceryle torquata"}, "locations": ["North-America"], "characteristics": {"prey": "Fish, frogs, crustaceans", "name_of_young": "Chic
k", "group_behavior": "Solitary", "estimated_population_size": "20,000,000", "biggest_threat": "Habitat loss, predators", "most_distinctive_feature": "Rust
-colored breast", "other_name(s)": "Collared kingfisher", "gestation_period": "23 days", "litter_size": "3-6", "habitat": "Marshes, riverbanks", "predator
s": "Raptors", "diet": "Carnivore", "common_name": "Ringed kingfisher", "location": "North and South America", "color": "GreyFawnWhite", "skin_type": "Feat
hers", "top_speed": "25 mph", "lifespan": "5 to 10 years", "weight": "10 to 12 ounces", "height": "3 to 6 inches", "length": "12 to 16 inches", "age_of_sex
ual_maturity": "1 year", "age_of_weaning": "5 weeks"}}]
```

final_api_species_data.csv

Load individual species details from final_api_species_data.csv file. This file was the final/clean outcome from API data.

```
# Read the final_api_species_data.csv file from local directory. This is the file from API
api_species_df =pd.read_csv("C:\\Users\\Chitramoy\\Desktop\\MS-DSC\\DSC-680\\Project-2\\final_api_species_data.csv")

# Read the first 10 rows
api_species_df.head(10)
```

Methods/Steps:

Using the provided sources, here are specific analysis methods to complete the Wildlife Protection of Endangered Species project:

1. Data Collection and Integration

- **National Park Locations and Areas (Website):** Scrape or extract data on national park locations, areas, and relevant attributes. This data will provide important contextual information about the habitats and protected areas for endangered species.
- **Animals API:** Access and retrieve animal-related data such as species names, characteristics, habitats, and conservation statuses. This API data can be integrated with other sources to enhance the overall dataset.
- **Species.csv (File):** Import and preprocess the CSV file to extract species-specific data. This file may contain information on species names, scientific classifications, population sizes, threats, and conservation statuses.

2. Geospatial Analysis

- **Habitat Mapping:** Using GIS tools, map the locations of national parks and overlay them with species distribution data from the Animals API and Species.csv. This helps in identifying critical habitats and conservation areas.
- **Habitat Suitability Modeling:** Apply machine learning algorithms like MaxEnt to predict suitable habitats for endangered species based on environmental variables (e.g., climate, vegetation) extracted from the National Park data and Animals API.

3. Species Identification and Monitoring

- **Species Classification:** Develop and train machine learning models to classify different species based on attributes from the Animals API and Species.csv. Use this to identify species captured in images and videos from camera traps or drones within national parks.
- **Population Monitoring:** Utilize data from the Species.csv file to track population trends over time. Apply time-series analysis to detect patterns and predict future population changes.

4. Threat Detection and Prediction

- **Poaching and Illegal Activities:** Implement anomaly detection algorithms on data from the national parks to identify unusual activities or potential threats. Combine this with real-time monitoring using AI-powered surveillance systems.

- **Environmental Changes:** Analyze satellite imagery and remote sensing data to detect habitat changes and deforestation within national parks. Use this data to assess the impact of environmental changes on endangered species.

5. Data Visualization and Reporting

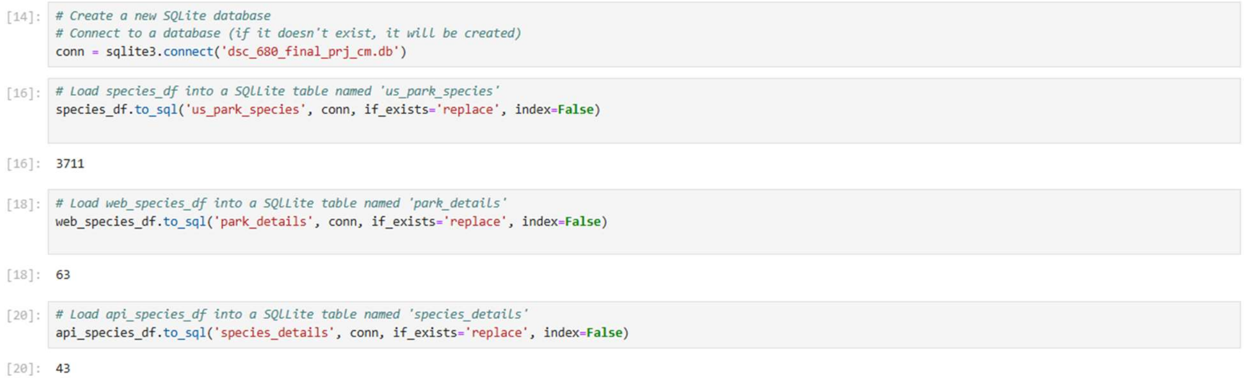
- **Interactive Maps:** Create interactive maps to visualize species distributions, habitats, and threats. Integrate data from the National Park locations, Animals API, and Species.csv to provide a comprehensive view.
- **Dashboards:** Develop dashboards to display key metrics and trends related to endangered species protection. Use data visualization tools to present insights and recommendations to conservationists and policymakers.

By utilizing these analysis methods, the project can effectively integrate data from various sources to monitor, protect, and conserve endangered species within national parks and their surrounding areas. If you have any specific requirements or need further details, feel free to let me know!

Analysis:

In below steps, create a new SQLLite database named as 'dsc_680_final_prj_cm.db'. If the database doesn't exist then it will get created otherwise connection will get established for existing database. Once connection gets established, load species_df into a SQlLite table named 'us_park_species'. Similarly, load web_species_df into a SQlLite table named 'park_details' and load api_species_df into a SQlLite table named 'species_details'.

If the table already exists, then it will be replaced. Here table 'us_park_species' contains 3711 rows, 'park_details' table contains 63 rows and 'species_details' table contains 43 rows.



```
[14]: # Create a new SQLite database
      # Connect to a database (if it doesn't exist, it will be created)
      conn = sqlite3.connect('dsc_680_final_prj_cm.db')

[16]: # Load species_df into a SQLLite table named 'us_park_species'
      species_df.to_sql('us_park_species', conn, if_exists='replace', index=False)

[16]: 3711

[18]: # Load web_species_df into a SQLLite table named 'park_details'
      web_species_df.to_sql('park_details', conn, if_exists='replace', index=False)

[18]: 63

[20]: # Load api_species_df into a SQLLite table named 'species_details'
      api_species_df.to_sql('species_details', conn, if_exists='replace', index=False)

[20]: 43
```

After loading the Dataframes, fetch all the column names from all the 3 tables for better visibility of columns. Here we will use the PRAGMA command to show the structure of columns.

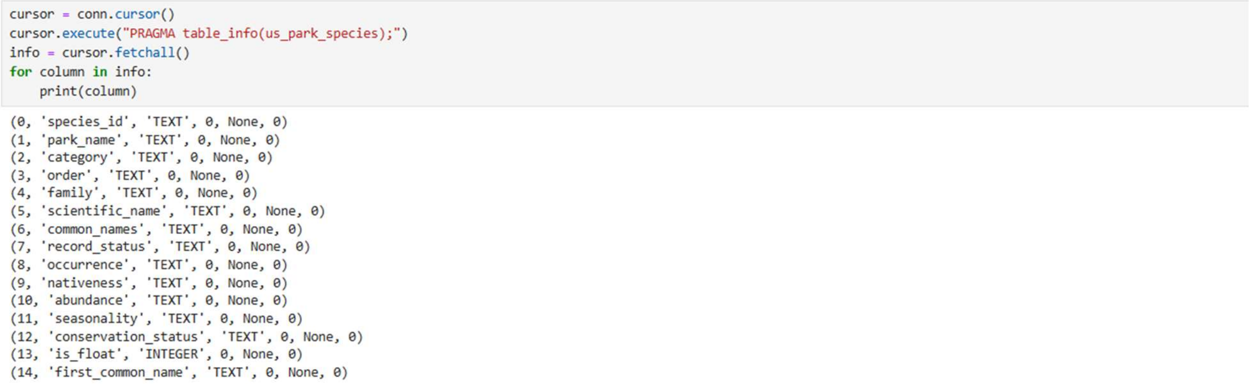
Brief introduction of tables

1. us_park_species : Contains information of which US park contains how many types of species and basic details

2. park_details : Contains US park information like area, lat/lon, totat recreation visitors etc.¶

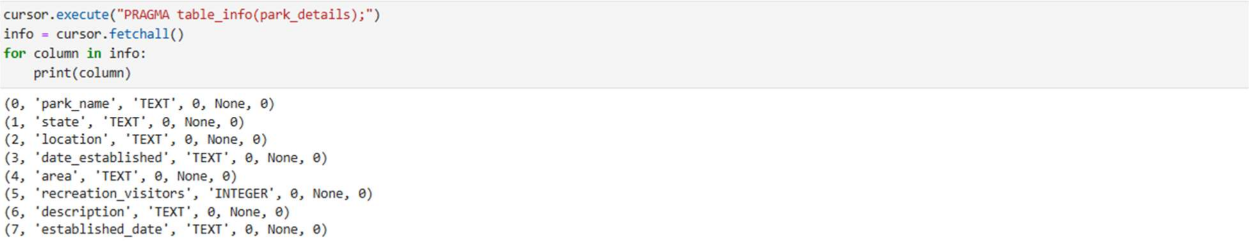
2. species_details : Contains individual species details like their kingdom, predators, habitat, biggest threar etc.

Create primary indexes on tables 'park_details' and 'species_details'. In 'park_details' table, park_name column can be primary. In 'species_details' table, species_name can be primary.



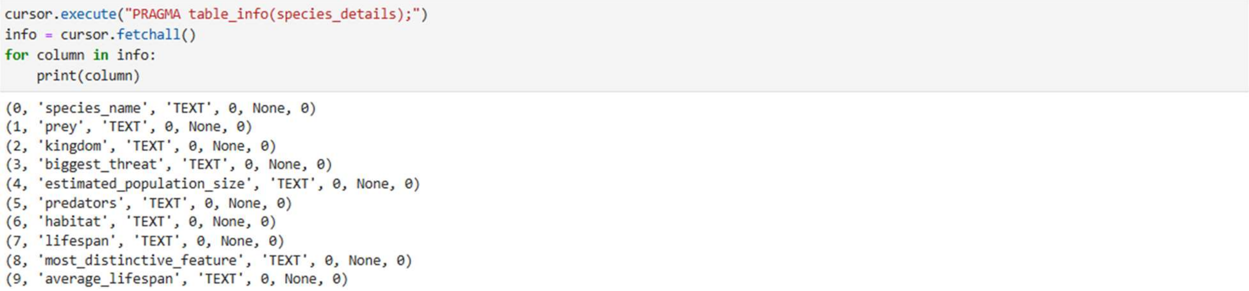
```
cursor = conn.cursor()
cursor.execute("PRAGMA table_info(us_park_species);")
info = cursor.fetchall()
for column in info:
    print(column)
```

```
(0, 'species_id', 'TEXT', 0, None, 0)
(1, 'park_name', 'TEXT', 0, None, 0)
(2, 'category', 'TEXT', 0, None, 0)
(3, 'order', 'TEXT', 0, None, 0)
(4, 'family', 'TEXT', 0, None, 0)
(5, 'scientific_name', 'TEXT', 0, None, 0)
(6, 'common_names', 'TEXT', 0, None, 0)
(7, 'record_status', 'TEXT', 0, None, 0)
(8, 'occurrence', 'TEXT', 0, None, 0)
(9, 'nativeness', 'TEXT', 0, None, 0)
(10, 'abundance', 'TEXT', 0, None, 0)
(11, 'seasonality', 'TEXT', 0, None, 0)
(12, 'conservation_status', 'TEXT', 0, None, 0)
(13, 'is_float', 'INTEGER', 0, None, 0)
(14, 'first_common_name', 'TEXT', 0, None, 0)
```



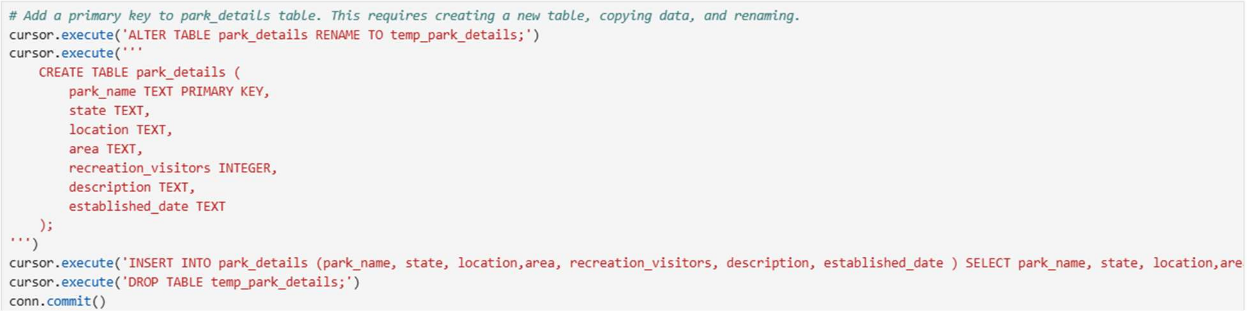
```
cursor.execute("PRAGMA table_info(park_details);")
info = cursor.fetchall()
for column in info:
    print(column)
```

```
(0, 'park_name', 'TEXT', 0, None, 0)
(1, 'state', 'TEXT', 0, None, 0)
(2, 'location', 'TEXT', 0, None, 0)
(3, 'date_established', 'TEXT', 0, None, 0)
(4, 'area', 'TEXT', 0, None, 0)
(5, 'recreation_visitors', 'INTEGER', 0, None, 0)
(6, 'description', 'TEXT', 0, None, 0)
(7, 'established_date', 'TEXT', 0, None, 0)
```

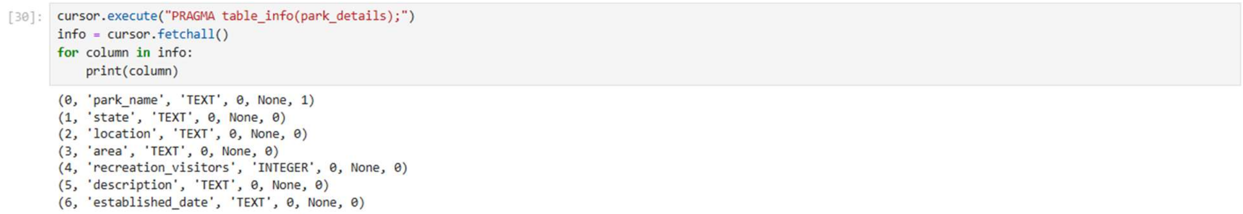


```
cursor.execute("PRAGMA table_info(species_details);")
info = cursor.fetchall()
for column in info:
    print(column)
```

```
(0, 'species_name', 'TEXT', 0, None, 0)
(1, 'prey', 'TEXT', 0, None, 0)
(2, 'kingdom', 'TEXT', 0, None, 0)
(3, 'biggest_threat', 'TEXT', 0, None, 0)
(4, 'estimated_population_size', 'TEXT', 0, None, 0)
(5, 'predators', 'TEXT', 0, None, 0)
(6, 'habitat', 'TEXT', 0, None, 0)
(7, 'lifespan', 'TEXT', 0, None, 0)
(8, 'most_distinctive_feature', 'TEXT', 0, None, 0)
(9, 'average_lifespan', 'TEXT', 0, None, 0)
```

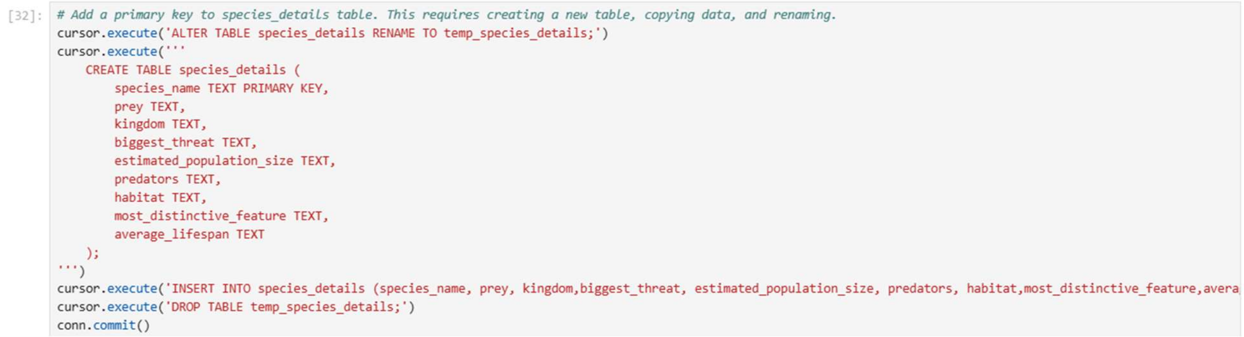


```
# Add a primary key to park_details table. This requires creating a new table, copying data, and renaming.
cursor.execute('ALTER TABLE park_details RENAME TO temp_park_details;')
cursor.execute('''
    CREATE TABLE park_details (
        park_name TEXT PRIMARY KEY,
        state TEXT,
        location TEXT,
        area TEXT,
        recreation_visitors INTEGER,
        description TEXT,
        established_date TEXT
    );
''')
cursor.execute('INSERT INTO park_details (park_name, state, location,area, recreation_visitors, description, established_date ) SELECT park_name, state, location,are
cursor.execute('DROP TABLE temp_park_details;')
conn.commit()
```

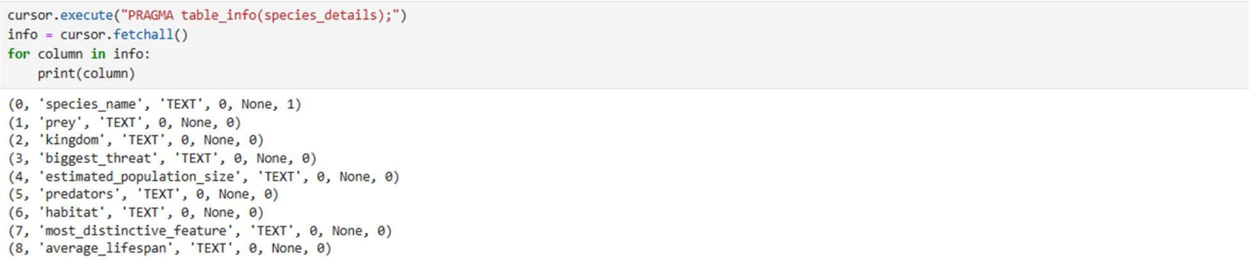


```
[30]: cursor.execute("PRAGMA table_info(park_details);")
      info = cursor.fetchall()
      for column in info:
          print(column)
```

```
(0, 'park_name', 'TEXT', 0, None, 1)
(1, 'state', 'TEXT', 0, None, 0)
(2, 'location', 'TEXT', 0, None, 0)
(3, 'area', 'TEXT', 0, None, 0)
(4, 'recreation_visitors', 'INTEGER', 0, None, 0)
(5, 'description', 'TEXT', 0, None, 0)
(6, 'established_date', 'TEXT', 0, None, 0)
```



```
[32]: # Add a primary key to species_details table. This requires creating a new table, copying data, and renaming.
      cursor.execute('ALTER TABLE species_details RENAME TO temp_species_details;')
      cursor.execute('''
          CREATE TABLE species_details (
              species_name TEXT PRIMARY KEY,
              prey TEXT,
              kingdom TEXT,
              biggest_threat TEXT,
              estimated_population_size TEXT,
              predators TEXT,
              habitat TEXT,
              most_distinctive_feature TEXT,
              average_lifespan TEXT
          );
      ''')
      cursor.execute('INSERT INTO species_details (species_name, prey, kingdom,biggest_threat, estimated_population_size, predators, habitat,most_distinctive_feature,avera
      cursor.execute('DROP TABLE temp_species_details;')
      conn.commit()
```

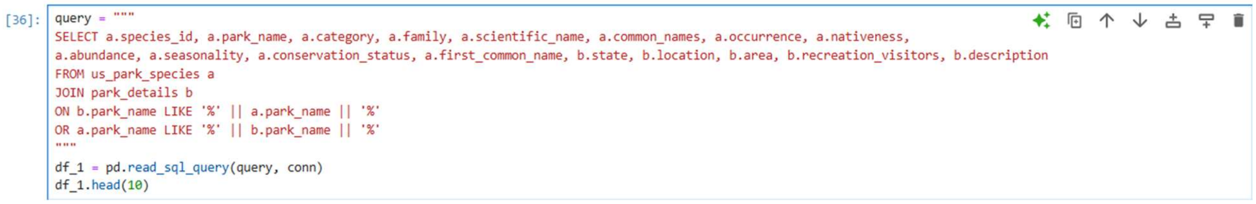


```
cursor.execute("PRAGMA table_info(species_details);")
info = cursor.fetchall()
for column in info:
    print(column)
```

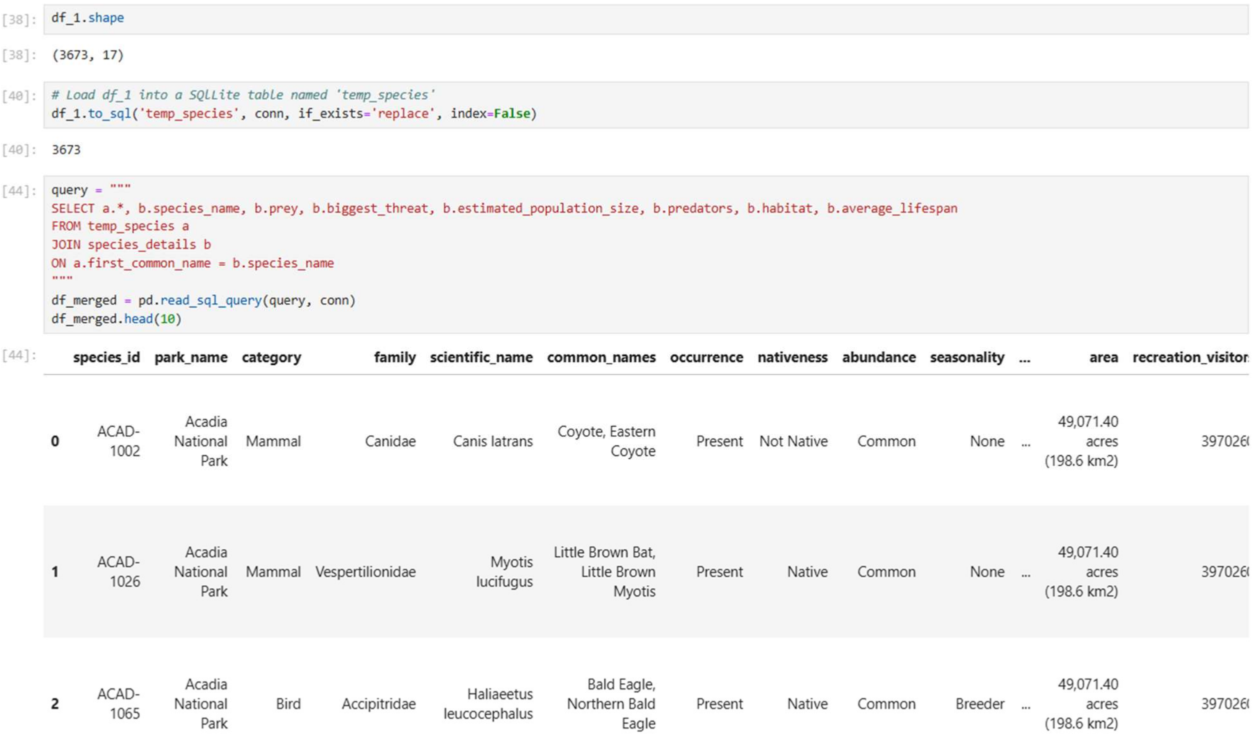
```
(0, 'species_name', 'TEXT', 0, None, 1)
(1, 'prey', 'TEXT', 0, None, 0)
(2, 'kingdom', 'TEXT', 0, None, 0)
(3, 'biggest_threat', 'TEXT', 0, None, 0)
(4, 'estimated_population_size', 'TEXT', 0, None, 0)
(5, 'predators', 'TEXT', 0, None, 0)
(6, 'habitat', 'TEXT', 0, None, 0)
(7, 'most_distinctive_feature', 'TEXT', 0, None, 0)
(8, 'average_lifespan', 'TEXT', 0, None, 0)
```

Here we will use a SQL query with the LIKE operator to perform a case-insensitive join on the park_name columns of both tables 'us_park_species' and 'park_details'. The % wildcard characters are used to match any sequence of characters.

Load the merged data into a dataframe and show top 10 rows.

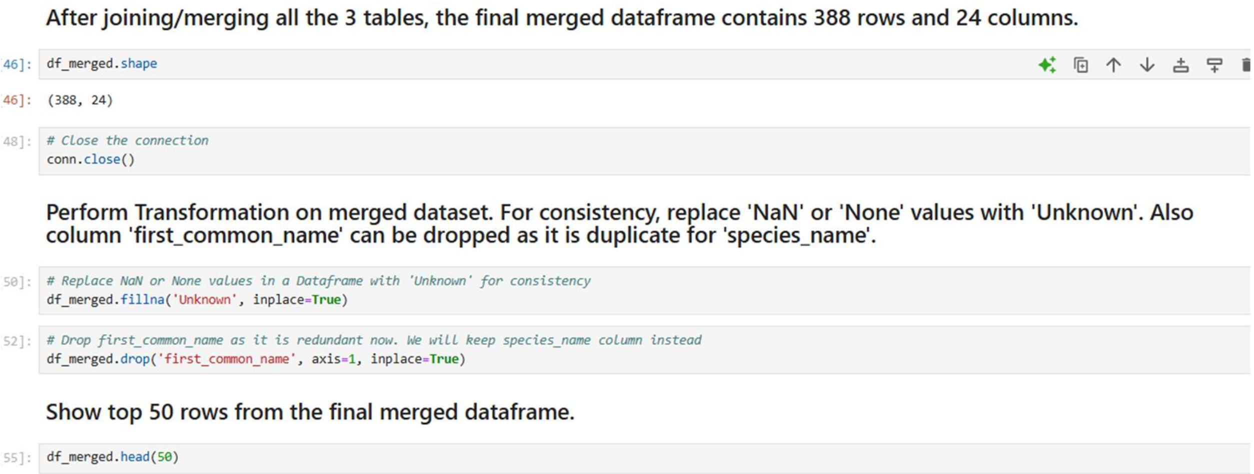


```
[36]: query = """
      SELECT a.species_id, a.park_name, a.category, a.family, a.scientific_name, a.common_names, a.occurrence, a.nativeness,
      a.abundance, a.seasonality, a.conservation_status, a.first_common_name, b.state, b.location, b.area, b.recreation_visitors, b.description
      FROM us_park_species a
      JOIN park_details b
      ON b.park_name LIKE '%' || a.park_name || '%'
      OR a.park_name LIKE '%' || b.park_name || '%'
      """
      df_1 = pd.read_sql_query(query, conn)
      df_1.head(10)
```



```
[38]: df_1.shape

[38]: (3673, 17)

[40]: # Load df_1 into a SQLLite table named 'temp_species'
      df_1.to_sql('temp_species', conn, if_exists='replace', index=False)

[40]: 3673

[44]: query = """
      SELECT a.*, b.species_name, b.prey, b.biggest_threat, b.estimated_population_size, b.predators, b.habitat, b.average_lifespan
      FROM temp_species a
      JOIN species_details b
      ON a.first_common_name = b.species_name
      """
      df_merged = pd.read_sql_query(query, conn)
      df_merged.head(10)

[44]:
```

	species_id	park_name	category	family	scientific_name	common_names	occurrence	nativeness	abundance	seasonality	...	area	recreation_visitor
0	ACAD-1002	Acadia National Park	Mammal	Canidae	Canis latrans	Coyote, Eastern Coyote	Present	Not Native	Common	None	...	49,071.40 acres (198.6 km2)	397026
1	ACAD-1026	Acadia National Park	Mammal	Vespertilionidae	Myotis lucifugus	Little Brown Bat, Little Brown Myotis	Present	Native	Common	None	...	49,071.40 acres (198.6 km2)	397026
2	ACAD-1065	Acadia National Park	Bird	Accipitridae	Haliaeetus leucocephalus	Bald Eagle, Northern Bald Eagle	Present	Native	Common	Breeder	...	49,071.40 acres (198.6 km2)	397026



After joining/merging all the 3 tables, the final merged dataframe contains 388 rows and 24 columns.

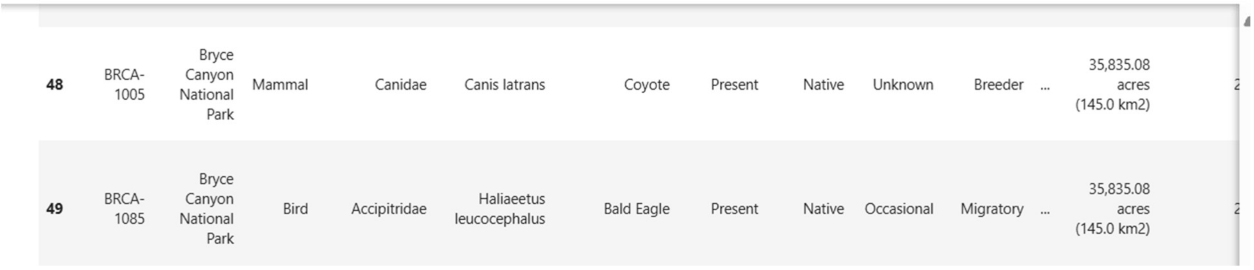
```
46]: df_merged.shape

46]: (388, 24)

48]: # Close the connection
     conn.close()
```

Perform Transformation on merged dataset. For consistency, replace 'NaN' or 'None' values with 'Unknown'. Also column 'first_common_name' can be dropped as it is duplicate for 'species_name'.

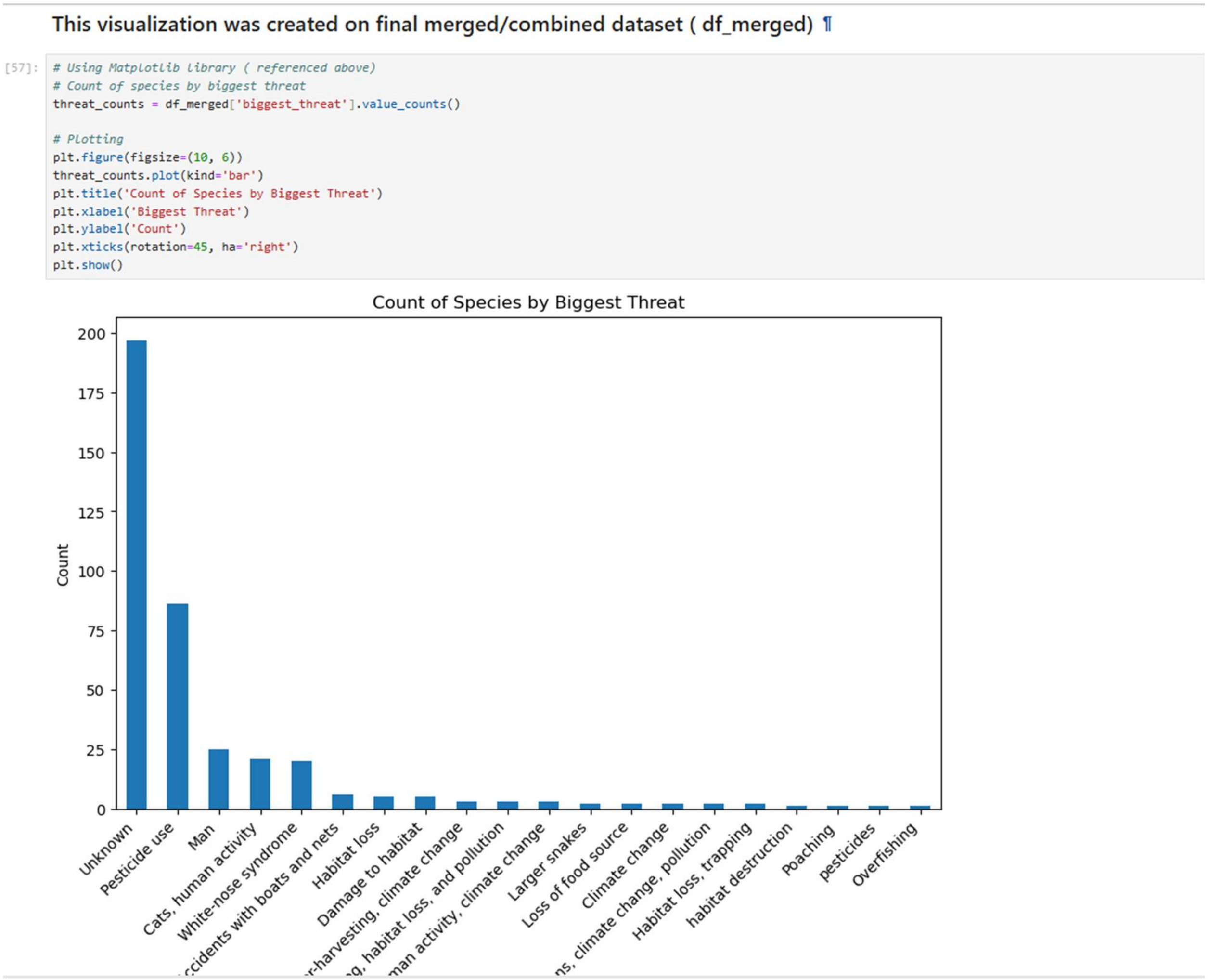
```
50]: # Replace NaN or None values in a Dataframe with 'Unknown' for consistency
     df_merged.fillna('Unknown', inplace=True)

52]: # Drop first_common_name as it is redundant now. We will keep species_name column instead
     df_merged.drop('first_common_name', axis=1, inplace=True)
```

Show top 50 rows from the final merged dataframe.

```
55]: df_merged.head(50)
```



48	BRCA-1005	Bryce Canyon National Park	Mammal	Canidae	Canis latrans	Coyote	Present	Native	Unknown	Breeder	...	35,835.08 acres (145.0 km2)	2
49	BRCA-1085	Bryce Canyon National Park	Bird	Accipitridae	Haliaeetus leucocephalus	Bald Eagle	Present	Native	Occasional	Migratory	...	35,835.08 acres (145.0 km2)	2



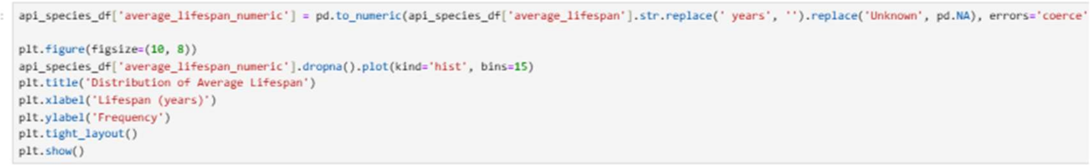
This visualization was created on final merged/combined dataset (df_merged) ¶

```
[57]: # Using Matplotlib library ( referenced above)
      # Count of species by biggest threat
      threat_counts = df_merged['biggest_threat'].value_counts()

      # Plotting
      plt.figure(figsize=(10, 6))
      threat_counts.plot(kind='bar')
      plt.title('Count of Species by Biggest Threat')
      plt.xlabel('Biggest Threat')
      plt.ylabel('Count')
      plt.xticks(rotation=45, ha='right')
      plt.show()
```

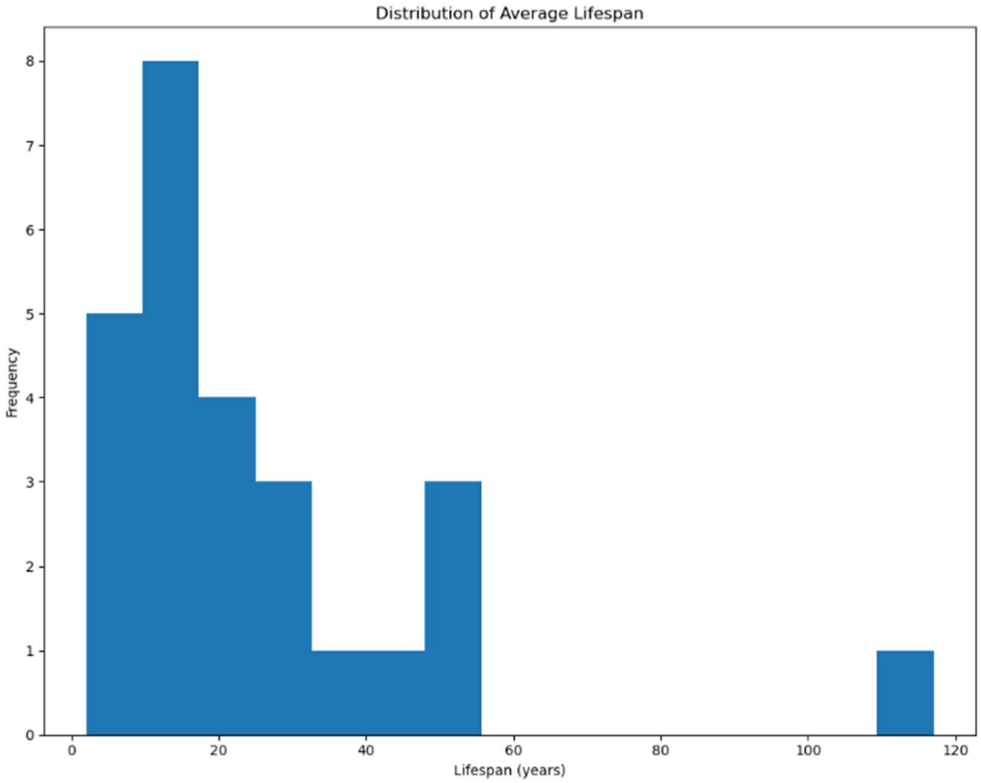
Visualization 2: Distribution of Average Lifespan of species in years

Average_lifespan is a numerical variable but contains some non-numeric values like "Unknown". We need to clean this data, convert it to numeric, and then we can use a histogram plot to visualize the distribution of lifespans. Majority of the species are falling under 20-30 years of average lifespan.

This visualization is created using individual dataset (api_species_df) which contains species details from API source.



```
api_species_df['average_lifespan_numeric'] = pd.to_numeric(api_species_df['average_lifespan'].str.replace(' years', '').replace('Unknown', pd.NA), errors='coerce'

plt.figure(figsize=(10, 8))
api_species_df['average_lifespan_numeric'].dropna().plot(kind='hist', bins=15)
plt.title('Distribution of Average Lifespan')
plt.xlabel('Lifespan (years)')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



Visualization 3: Generate visualization to show which park has biggest threat

Here a bar chart is used to show which park has biggest threat. This approach involves counting the occurrences of the biggest threat for each park and then plotting these counts. Filter out 'Unknown' values from biggest_threat and create a new dataframe 'new_df'.

1. Group the DataFrame by park_name and biggest_threat, and count the occurrences.

2. Find the biggest threat for each park by selecting the threat with the highest count for each park.

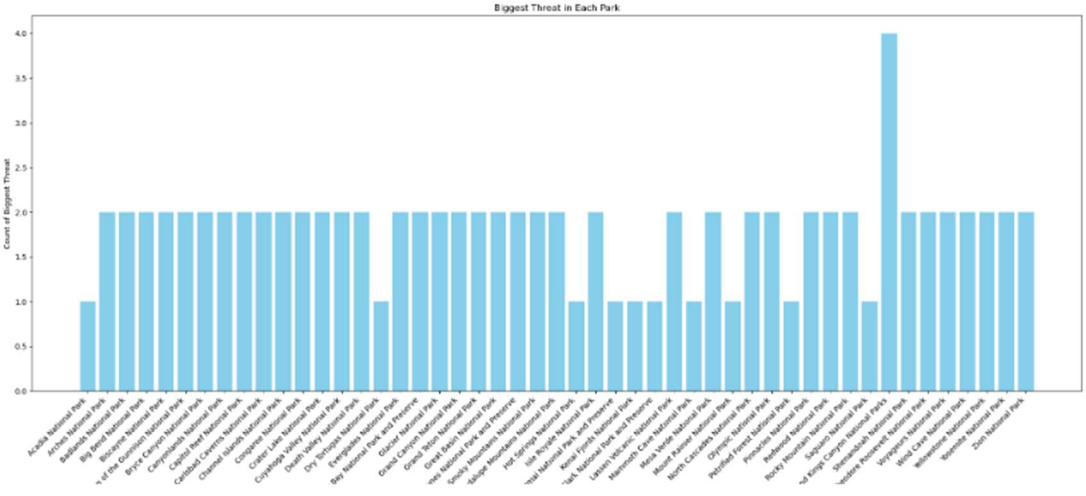
3. Plot a bar chart with parks on the x-axis and the count of the biggest threat on the y-axis.s.

Below visualization is created on merged/combined dataset (df_merged). Based on the generated graph, Sequioa and Kings Canyon National Park has the biggest threat.

```
[61]: # Remove 'Unknown' value rows for biggest_threat
      new_df = df_merged.loc[df_merged['biggest_threat'] != 'Unknown']
      new_df.shape
```

```
[61]: (191, 23)
```

```
[63]: # Step 1: Count occurrences of each threat for each park
      threat_counts = new_df.groupby(['park_name', 'biggest_threat']).size().reset_index(name='counts')

      # Step 2: Find the biggest threat for each park (the threat with the highest count)
      biggest_threats = threat_counts.loc[threat_counts.groupby('park_name')['counts'].idxmax()]

      # Step 3: Plotting
      plt.figure(figsize=(20, 10))
      plt.bar(biggest_threats['park_name'], biggest_threats['counts'], color='skyblue')
      plt.title('Biggest Threat in Each Park')
      plt.xlabel('Park Name')
      plt.ylabel('Count of Biggest Threat')
      plt.xticks(rotation=45, ha='right')
      plt.tight_layout()
      plt.show()
```



Visualization 4: Proportion of Biggest Threat for each Category of Species

This visualization involves grouping the data by category and biggest_threat, counting the occurrences, and then plotting the family with the highest count of the biggest threat.

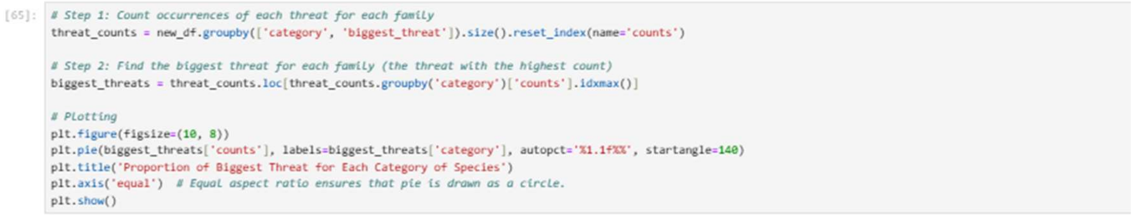
Here is a step-by-step approach

1. Group the DataFrame by category and biggest_threat, and count the occurrences.

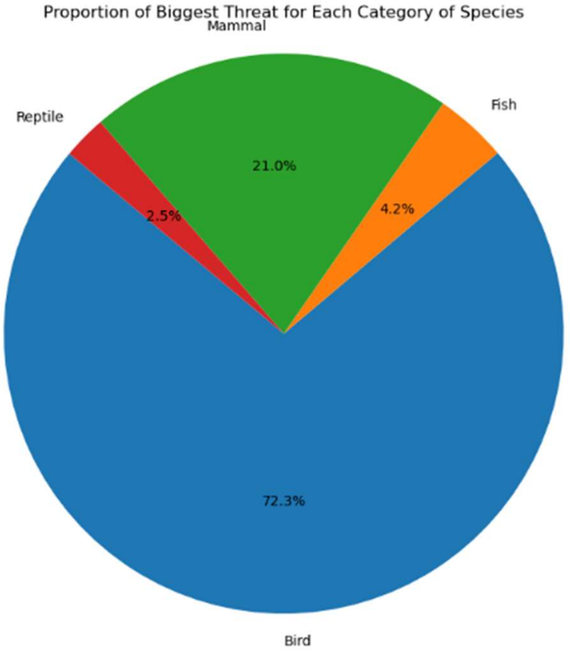
2. Find the biggest threat for each category by selecting the threat with the highest count for each category of spec

3. Plot a pie chart with color coded categories and percentage of threat.

This visualization is created on merged dataframe with filtered 'Unknown' values for biggest threat (new_df). Here category 'Bird' are under biggest threat.



```
[65]: # Step 1: Count occurrences of each threat for each family
      threat_counts = new_df.groupby(['category', 'biggest_threat']).size().reset_index(name='counts')

      # Step 2: Find the biggest threat for each family (the threat with the highest count)
      biggest_threats = threat_counts.loc[threat_counts.groupby('category')['counts'].idxmax()]

      # Plotting
      plt.figure(figsize=(10, 8))
      plt.pie(biggest_threats['counts'], labels=biggest_threats['category'], autopct='%1.1f%%', startangle=140)
      plt.title('Proportion of Biggest Threat for Each Category of Species')
      plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
      plt.show()
```



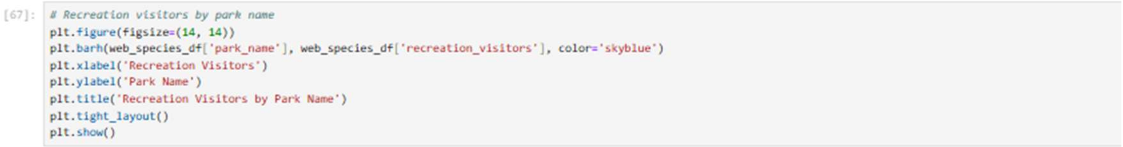
Visualization 5: Recreation Visitors by Park Name

This visualization is to show number of recreation visitors by Park Name. It is shown by bar graph and is generated from individual web_species_df which contains park details from website.

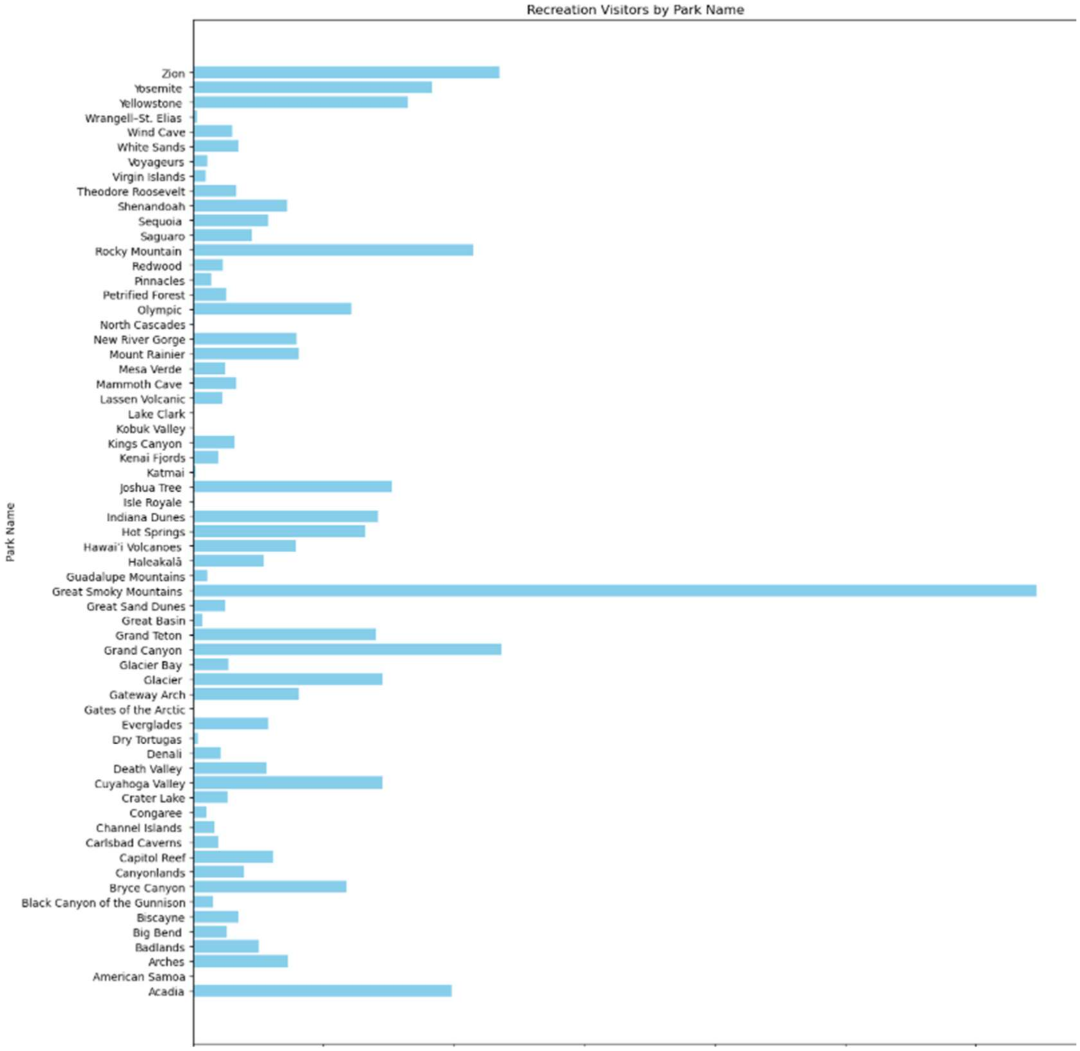
In previous visualizations, since the top reason of biggest threats are due to 'Pesticides' and 'Man', below visualization will provide information on the distribution of recreation visitors in each park which might have some impact.

Interpretation

In below visualization, 'Great Smoky Mountains' national park got exceptionally high count of recreation visitors. In previous visualizations, 'Sequioa and Kings Canyon National Park' shows highest threat but number of recreation visitor is comparatively low with repect to other parks. Based on that, number of Recreation visitors doesn't seem to be impacting the cause of biggest threat for Sequioa.

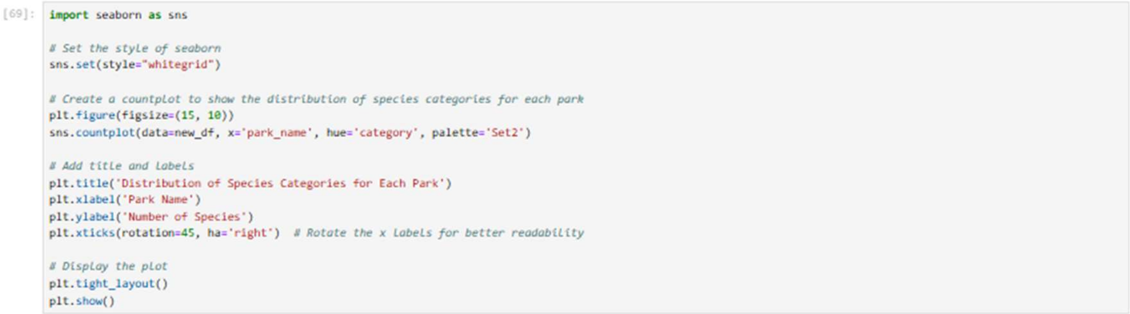


```
[67]: # Recreation visitors by park name
      plt.figure(figsize=(14, 14))
      plt.barh(web_species_df['park_name'], web_species_df['recreation_visitors'], color='skyblue')
      plt.xlabel('Recreation Visitors')
      plt.ylabel('Park Name')
      plt.title('Recreation Visitors by Park Name')
      plt.tight_layout()
      plt.show()
```

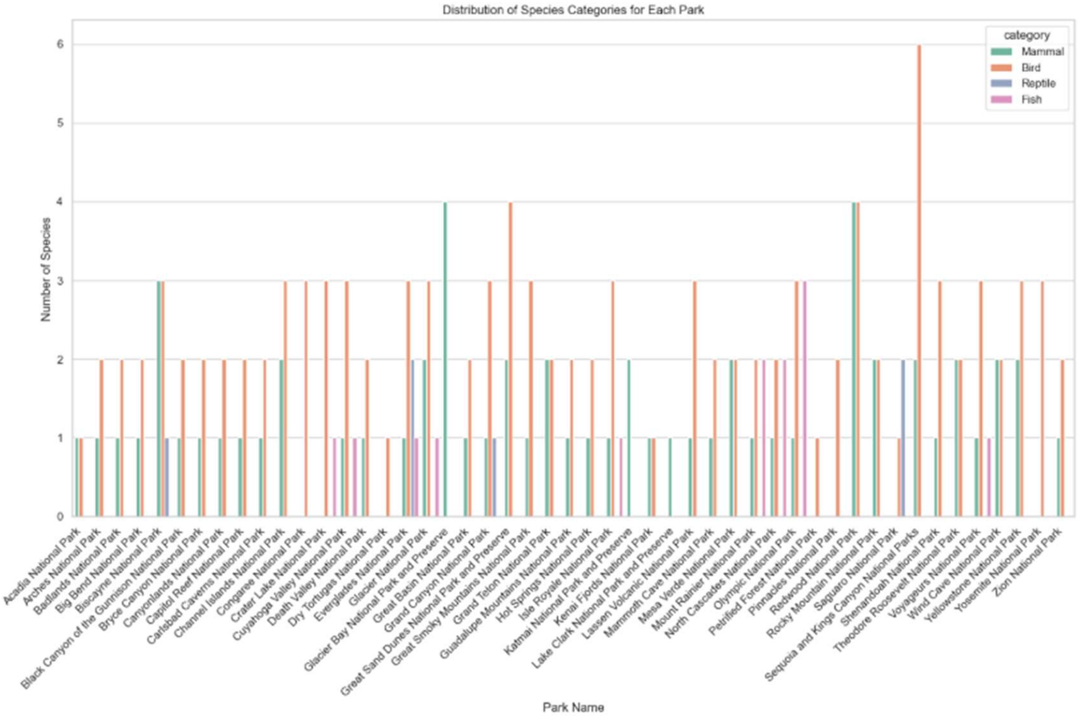


Visualization 6: Species category distribution for each park

This visualization is to identify which different categories of species are habitat for each park. The count plot is developed using Seaborn library. Based on the plot, for Sequioa and Kings Canyon National Park species category 'Bird' and 'Mammal' are more prevalent as compared to other parks. This visualization is showing relationship between Sequioa and Kings Canyon National Park and threat as Sequioa and Kings Canyon National Park is showing biggest_threat count across all US parks, also it got identified that 'Bird' category species is at highest threat based on previous plots and since Sequioa and Kings Canyon National Park got the 'Bird' as the most popular and highest count species, this park might be at a risk of endangered bird species.



```
[69]: import seaborn as sns

      # Set the style of seaborn
      sns.set(style="whitegrid")

      # Create a countplot to show the distribution of species categories for each park
      plt.figure(figsize=(15, 10))
      sns.countplot(data=new_df, x='park_name', hue='category', palette='Set2')

      # Add title and labels
      plt.title('Distribution of Species Categories for Each Park')
      plt.xlabel('Park Name')
      plt.ylabel('Number of Species')
      plt.xticks(rotation=45, ha='right')  # Rotate the x labels for better readability

      # Display the plot
      plt.tight_layout()
      plt.show()
```



Conclusion:

The purpose of this project is to intelligently monitor endangered and rare wildlife species by using Machine Learning models with animal data and computational resources from various parks in United States. For this project, data has been collected from flat file, website and API sources. The data extracted from flat file contains plants and animal species information from different national parks of United States. After applying few cleaning and transformation steps, final dataframe was generated and saved locally as final_species.csv. Next set of data which got extracted from Website contains park information which includes Location, Date established, Area of the park and Number of visitors information of all the national parks in United States. Post cleanup and transformation, this dataset was saved locally as final_web_species_data.csv. The 3rd set of data was extracted from Animal API which contains detail information for individual species like Average lifespan, Biggest Threat, Habitat, Predator etc. This dataset got saved locally as final_api_species_data.csv after cleanup and transformation.

The purpose of this project is to merge all three clean datasets and create visualizations on those datasets. The locally saved files got loaded as DataFrames first and then those DataFrames got loaded as SqlLite tables named as 'us_park_species', 'park_details' and 'species_details'. These tables are getting further combined/merged using matching keys/data like Park Name, Species Name and final dataset combining all the 3 datasets is getting created based on matched values.

In this project, the purpose of visualizations is to analyze and evaluate categories of species which are having biggest threat and if they are under the risk of endangered species. The 1st visualization (Bar Plot) is showing the count of species by biggest threat where 'Pesticide use' and 'Man/Human activity' are the topmost reasons. The 2nd visualization (Histogram Plot) is showing the distribution of Average Lifespan of species in years. It shows that average lifespan for majority of species is falling under 20-30 years of range. If those species are falling under biggest threat, then due to their low average lifespan there is a risk of possible extinction in near future. The 3rd visualization (Bar Plot) is showing which park has biggest threat and based on the generated graph, Sequioa and Kings Canyon National Park shows the highest. the 4th visualization (Pie Chart) is showing Proportion of Biggest Threat for each Category of Species, which interestingly shows that Birds are 72.3% victims of biggest threats (Pesticides, human activity).

The 5th visualization (Bar Plot) is generated to identify the number of recreation visitors for each park where we can assume that 'Sequioa and Kings Canyon National Park' shows highest threat, but number of recreation visitor is comparatively low with respect to other parks. Based on that, number of Recreation visitors doesn't seem to be impacting the cause of biggest threat for Sequioa. The 6th visualization (Count plot) is showing Species category distribution for each park where interesting fact to see is that 'Sequioa and Kings Canyon National Park' has majority 'Bird' category of species.

'Bird' category species is at highest threat based on previous plots and since Sequioa and Kings Canyon National Park is showing the highest count of species under Biggest Threat and got the 'Bird' as the most popular and highest count species, this park might be at a risk of endangered bird species.

Limitations:

Inconsistent Species Information: The final_species.csv and final_web_species_data.csv files have overlapping but not completely matching species-related data.
This inconsistency can lead to difficulties in accurately integrating and analyzing species information across the datasets.

Lack of Comprehensive Threat Data: The datasets provide some details about biggest_threat and predators in the final_species.csv file, but there is limited information on other critical threats, such as habitat loss or climate change, which are essential for creating effective protection strategies.

Data Completeness and Timeliness: The final_web_species_data.csv includes date_established and recreation_visitors information, which may not be frequently updated. Incomplete or outdated information can affect the accuracy of the AI/ML models in predicting current conservation status and population trends

Addressing these limitations can improve the effectiveness of the AI/ML project and lead to better protection strategies for endangered species.

Ethical Considerations:

Data Bias and Accuracy: Inaccurate or biased data can lead to incorrect predictions, potentially diverting resources away from species in critical need or misinforming conservation strategies.

Privacy Concerns: Some datasets may include sensitive locations of endangered species. If this information is not handled correctly, it could lead to increased poaching or habitat destruction.

Misuse of Predictions: Predictions could be misinterpreted or misused by stakeholders, leading to harmful decisions or policies against the very species they aim to protect.

Dependence on Technology: Over-reliance on machine learning predictions could overshadow traditional conservation knowledge and practices, which are also crucial for species preservation.

Resource Allocation: Machine learning models might prioritize some species over others, influencing funding and conservation efforts and potentially neglecting species not deemed "important" by the model's criteria.

Future Uses/Additional Applications:

Predictive Habitat Analysis: By integrating species occurrence and habitat information, the AI/ML models can predict potential habitats for endangered species. This can help conservationists identify areas that need protection and restoration efforts.

Visitor Impact Assessment: Analyzing recreation visitor data alongside species data can provide insights into how human activities affect wildlife. This can lead to the development of strategies to minimize human impact on endangered species and their habitats.

Conservation Prioritization: Using AI/ML to assess the conservation status, abundance, and threats of various species, the project can prioritize conservation efforts for species most at risk. This can help allocate resources more effectively and ensure the most critic al species receive the attention they need.

By leveraging these datasets, the AI/ML project can significantly enhance wildlife protec tion efforts and contribute to the preservation of endangered species.

Recommendations:

Data Enrichment: Enhance datasets by integrating additional external data sources, such as climate data, habitat maps, and human activity data. This can provide a more comprehensive view of the factors affecting endangered species and improve predictive accuracy.

RealTime Monitoring: Implement realtime data collection and monitoring systems to ke ep the datasets updated. This can involve using IoT sensors, satellite imagery, and citizen science contributions to ensure the AI/ML models have the most current info rmation for accurate analysis and decision-making.

Interdisciplinary Collaboration: Collaborate with ecologists, conservationists, and data scientists to refine AI/ML models and ensure they are grounded in ecological principles. This can help in developing targeted conservation strategies that are both scientifically sound and practically feasible.

By adopting these recommendations, the project can enhance its effectiveness and contribute significantly to the protection of endangered species.

Ethical Assessment:

Ensuring fairness in model predictions across different species, ecosystems, and demographics is crucial and efforts should be made to mitigate bias and promote equitable outcomes. During data transformation, some of the NaN or empty values got converted into 'Unknown' value for consistency or has been removed and filtered out . But there is a possibility that some crucial species information might have lost during this clean up as they did not have sufficient information recorded. If the data used for training is biased (e.g., underrepresents certain species or habitats), the model may produce biased predictions.

Based on visualizations, one of the observation came up that 'Man/Human Activity' is one of the topmost reasons of biggest threat for species. In that case, number of recreation visitors also play an important role because a species habitat can be impacted with increasing visitors so the data need to be accurate and should get updated with recent information for this project. Accurate information or missing data can lead to inappropriate results.

Another important ethical implication is Transparency on how the data collection process is taking place, including what data is being collected, how it will be used, and for what purpose considering all the guidelines for website and API calls. This data should not be used inadvertently which can cause intentional impact to tourism or any decision regarding national parks or any species.

Challenges/Issues:

When working with data for endangered species in the context of machine learning several concerns and challenges may arise:

Data Availability: API data on endangered species can be scarce or incomplete due to the limited number of individuals and the difficulty in tracking and monitoring these species in their natural habitats. After running the API for a few animals, not every output provides all the information required in the final dataset.

Temporal and Spatial Variability: Endangered species data can be highly variable over time and space, requiring complex models to understand patterns of behavior, migration, and population dynamics.

Impact of Climate Change: The effects of climate change on species habitats, migration patterns, and population dynamics add another layer of complexity to modeling efforts.

Integration with Conservation Strategies: The ultimate goal is to aid conservation efforts, which requires models not only to predict but also to prescribe actionable strategies. This requires a deep understanding of the ecological, social, and economic

References:

Here are some additional sources that can be used to validate results and support the Wildlife Protection of Endangered Species project using AI/ML:

1. **Scientific Journals and Publications**

   Sources: Journals such as "Conservation Biology," "Biodiversity and Conservation," and "Ecological Applications."

Description: These journals publish peer-reviewed research articles on wildlife conservation, habitat management, and the application of AI/ML in ecological studies. They provide credible scientific evidence and methodologies to support the project.

2. **International Union for Conservation of Nature (IUCN) Red List**
   Source: IUCN Red List of Threatened Species
   Description: The IUCN Red List is a comprehensive resource that provides information on the conservation status of species worldwide. It includes data on species populations, threats, and conservation measures, which can be used to validate species-specific results.
   Website: IUCN Red List
3. **World Wildlife Fund (WWF) Reports**
   Source: WWF
   Description: WWF publishes reports and white papers on various aspects of wildlife conservation, including endangered species protection, habitat preservation, and the impact of climate change. These reports offer valuable insights and real-world case studies.
   Website: World Wildlife Fund
4. **United Nations Environment Programme (UNEP)**
   Source: UNEP
   Description: UNEP provides reports, data, and policy recommendations on global environmental issues, including biodiversity and endangered species conservation. Their publications can help validate the project's methodologies and findings.
   Website: UNEP
5. **Convention on International Trade in Endangered Species of Wild Fauna and Flora (CITES)**
   Source: CITES
   Description: CITES maintains a database of species that are regulated under international trade agreements. It provides information on species protection measures and trade regulations, which can be used to support conservation efforts.
   Website: CITES
6. **National Geographic**
   Source: National Geographic
   Description: National Geographic publishes articles, documentaries, and research on wildlife conservation and endangered species. Their content

often includes expert insights and field research that can complement the project's data and analysis.
Website: National Geographic

These sources offer credible and comprehensive information that can validate the results and support the Wildlife Protection of Endangered Species project using AI/ML.

10 user questions regarding the Wildlife Protection of Endangered Species AI/ML project?

1. Whats the benefit of Wildlife Protection of Endangered to general public?

   Ans: Protecting endangered wildlife helps maintain biodiversity, ensuring balanced ecosystems. It promotes economic growth through ecotourism, creating jobs and supporting local communities. Additionally, healthy ecosystems provide vital services like clean air, water, and disease regulation, benefiting public health.

2. What changes were made to the data?

   Ans: No major changes has been done to data in this milestone project. All the 3 datasets got loaded into SqlLite tables and during joining some redundant columns got filtered out in the final dataset. For consistency, some NaN/None values got converted into 'Unknown' across the final dataset.

3. How do you handle missing or inconsistent data across the different datasets?

   Ans: To handle missing or inconsistent data, first, identify and fill in missing values using techniques like mean imputation or interpolation, or flag them for review. Next, standardize data formats, units, and naming conventions across datasets. Finally, cross-verify data to resolve any discrepancies, ensuring consistent and accurate information. These steps help maintain data integrity.

4. Are there any legal or regulatory guidelines for your data or project topic?

   Ans: Not required as the data is collected from a valid API source using secret API key and includes species information only.

5. How do you measure the effectiveness and accuracy of the AI/ML models in protecting endangered species?

   Ans: To measure the effectiveness and accuracy of AI/ML models in protecting endangered species, monitor key metrics such as prediction accuracy, precision, and recall. Additionally, track real-world outcomes like the reduction in poaching incidents and improvements in species population trends. Regularly validate model predictions against field data and adjust models as necessary for continuous improvement.

6. What risks could be created based on the transformations done?

   Ans: No major transformations have been done in this milestone project but for visualizations one big chunk of data has been filtered out for biggest threat as 'Unknown'. Since the visualizations got generated on limited data, there is a possibility of biased predictions or incomplete analysis.

7. Did you make any assumptions in cleaning/transforming the data?

   Ans: Yes, for visualization a big chunk of data got filtered out with 'Unknown' value as biggest threat, so all the assumptions were made on limited dataset. The limited dataset (new_df) is containing only 191 rows of species data.

8. Was your data acquired in an ethical way?

   Ans: The data has been acquired in ethical ways as all sources are trusted and park information is managed by National Park Services in United States. However, the time and effort spent on species inventories varies from park to park, which may result in data gaps. Species taxonomy changes over time and reflects regional variations or preferences; therefore, records may be listed under a different species name.

9. How would you mitigate any of the ethical implications you have identified?

   Ans: Since the final dataset has limited amount of data which can be used for training purposes of the model but further data collection from trusted sources can provide better results and more insights. In further analysis, if species data gets outdated then a mechanism need to be implemented where real time data on certain park information will get feeded from other credible sources.

10. How can stakeholders and conservationists’ access and use the insights generated by the AI/ML models?

    Ans: Stakeholders and conservationists can access insights from AI/ML models via user-friendly dashboards, reports, and data visualization tools. These platforms provide actionable recommendations, real-time alerts, and comprehensive analytics. By utilizing these insights, they can make informed decisions, prioritize conservation efforts, and monitor the effectiveness of their strategies.