# eBirdCleanorgFinal

May 1, 2024

### 0.0.1 The Data Sleuths


| Name | Email | Github |
|---|---|---|
| Takashi Osanai | tosana2@uic.edu | TakashiOsa |
| Arka Pal | apal7@uic.edu | ArkaPal-uic |
| Raunak Singh | rsing76@uic.edu | raunaksingh1497 |
| Lokesh Manideep | lbogg@uic.edu | lokeshmanideepb |


**Repository** : link
**Dataset Source** : link **Initial Dataset** : link **Final Dataset** : link

**Chicago Luxury Effect -> Chicago Bird Diversity** : The inital idea was to check for luxury effect(A pattern of higher biodiversity in neighborhoods with higher socioeconomic status) in Chicago. Due to lack of a large concrete dataset for socioeconomic factors which is essential for training machine learning models, we chose to analyse only eBird data.

[2]:
```
import eda, ml, data_cleaning,constants,pandas as pd, numpy as np, geopandas as␣
↪gpd, warnings, textwrap; warnings.filterwarnings("ignore")
```

**Note** : The initial dataset was too large(unzipped version : **>13 GB**). Hence the dataset was split into several small files(xaa,xab,xac) for easy processing using split command. The split files can be found at link. Download the zip file, extract it to the data folder in the project directory before running Section 1.Keep in mind that the Section 1 execution takes approximately 1 hour on normal laptops.If you prefer to bypass the final working dataset creation process(**Section 1**), download the dataset directly using link and run from the **Section 2**

## 0.1 Section 1. Data

### 0.1.1 1.1 Read and Filter eBird Dataset

[ ]:
```
data_cleaning.create_cook_county_dataset()
df = data_cleaning.filter_dataset()
```

### 0.1.2 1.2 ebird Dataset Transformation

[ ]:
```
df = data_cleaning.data_transformation(df)
```

### 0.1.3 1.3 Aggregate eBird data based on neighborhood

Combine the eBird dataset and neighborhood geo dataframe so that each bird observation is tagged with the community in which the bird has been observed.

[ ]:
```
ebird_gdf = data_cleaning.agg_neighborhood(df)
```

### 0.1.4 1.4 Transform and Save the final dataset

Modify the dataframe with new columns(*day, year, month*) for easy processing during next stages and save the dataset.The granularity of the final dataset is eBird data of Chicago starting from the year 1948 to 2024. - Number of rows - 3902730 - Number of columns - 10 (COMMON NAME', 'SCIENTIFIC NAME', 'NATIVE', 'COUNT', 'OBSERVATION DATE', 'geometry', 'community', 'OBSERVATION MONTH', 'OBSERVATION DAY','OBSERVATION YEAR', 'Location')

[ ]:
```
save_final_dataset(ebird_gdf)
```

## 0.2 Section 2: Vizualizations

[3]:
```
import matplotlib.pyplot as plt, matplotlib.patches as mpatches, seaborn as␣
↪sns, plotly.express as px, altair as alt
```

[4]:
```
ebird_gdf=pd.read_csv('data/final_dataset.tsv', sep='\t')
ebird_gdf = ebird_gdf[(ebird_gdf["OBSERVATION YEAR"] >= 2014) &␣
↪(ebird_gdf["OBSERVATION YEAR"] <= 2023)]
```

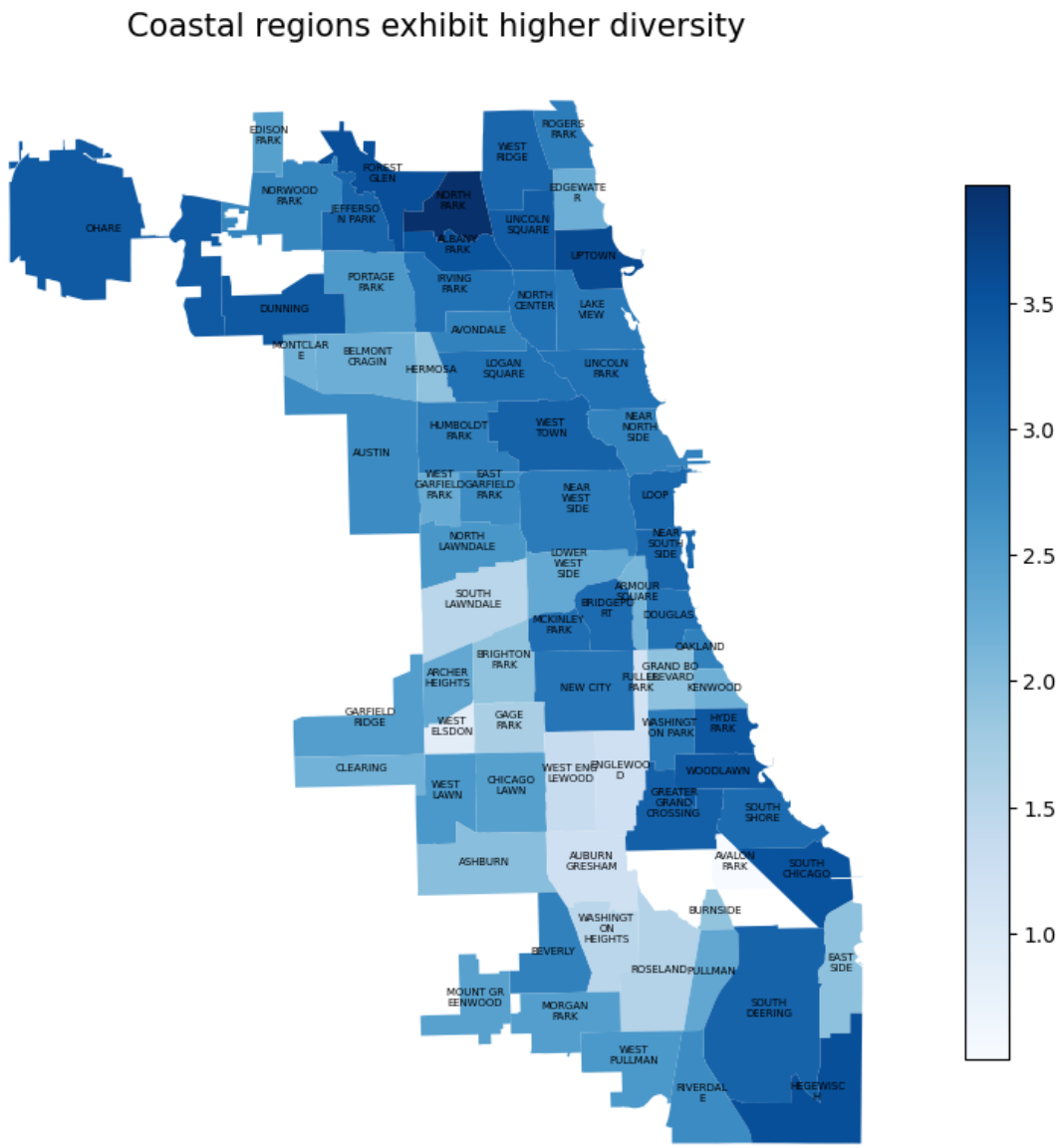
## 0.3 2.1 (*Which communities have higher bird diversity?*)

(*The graph depicts bird diversity across Chicago's communities in the year 2023.It is evident from the graph that the northern regions of Chicago have higher bird diversity than the south.Also,the areas adjacent to the coast demonstrate notably higher bird diversity, potentially due to their proximity to water bodies. This closeness to the coast likely plays a significant role in shaping the migration patterns of birds, resulting in distinct variations in diversity across different regions*)

[5]:
```
com_areas = gpd.read_file('data/neighborhoods/
↪geo_export_f5325bf0-9c6d-49a5-a5d9-0e5bf24fa856.shp')
df_2023 = ebird_gdf[ebird_gdf["OBSERVATION YEAR"] == 2023]
viz_df = eda.aggregate_data(df_2023,["community","COMMON NAME"])
viz_df["COUNT"] =  viz_df["COUNT"].apply(lambda x: sum(x))
viz_df = eda.aggregate_data(viz_df,["community"])
viz_df["shannon_index"] = viz_df["COUNT"].apply(eda.shannon_index)
viz_df = viz_df[["community","shannon_index"]]
gdf = com_areas.merge(viz_df, on='community')
```

[6]:
```
fig, ax = plt.subplots(figsize=(10, 10))
gdf.plot(column='shannon_index', cmap='Blues', linewidth=0.8,␣
↪ax=ax,legend=True,legend_kwds={'shrink': 0.75})
for idx, row in gdf.iterrows():
```

```
    wrapped_text = textwrap.fill(row['community'], width=8)
    plt.annotate(text=wrapped_text, xy=row.geometry.centroid.coords[0],␣
 ↪ha='center', fontsize=5)
ax.set_title('Coastal regions exhibit higher diversity', fontdict={'fontsize':␣
 ↪'15', 'fontweight' : '3'})
ax.set_axis_off()
plt.show()
```

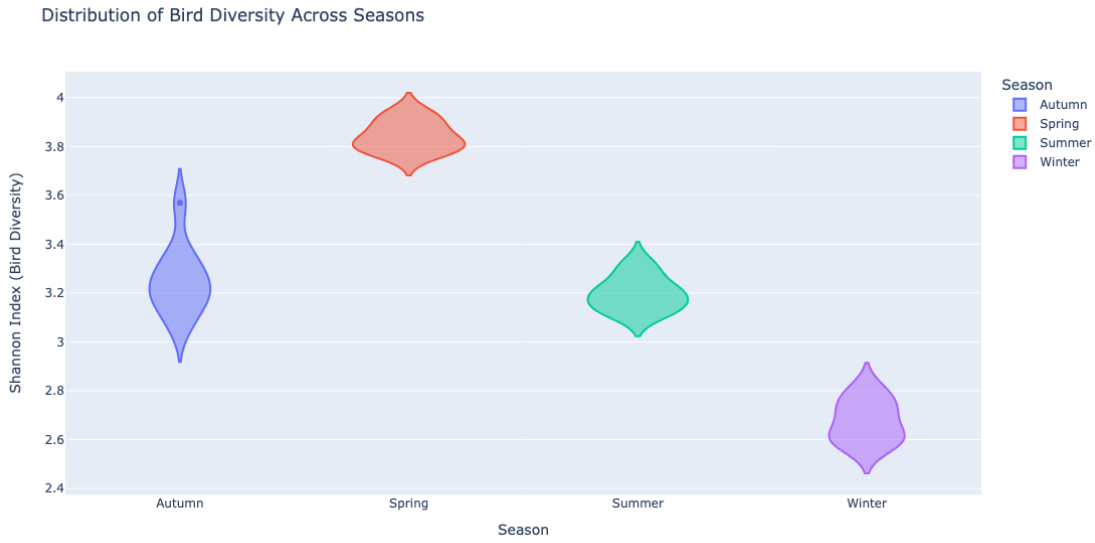


### 0.3.1 2.2 (Which Season has more bird diversity?)

*The violin plot illustrates the distribution of bird diversity values(Shannon index) across seasons. It reveals that the highest diversity tends to occur in spring. Autumn and summer exhibit comparable diversity levels, while winter tends to have the lowest diversity among the seasons.Spring exhibits higher bird diversity due to the abundance of resources like insects and budding plants, supporting breeding activities. Conversely, winter's harsh conditions and food scarcity limit bird diversity.*

[7]:
```
ebird_gdf['Season'] = ebird_gdf['OBSERVATION MONTH'].apply(eda.
 ↪categorize_season)
viz_df = eda.aggregate_data(ebird_gdf,["Season","OBSERVATION YEAR","COMMON␣
 ↪NAME"])
viz_df["COUNT"] =  viz_df["COUNT"].apply(lambda x: sum(x))
viz_df = eda.aggregate_data(viz_df,["Season","OBSERVATION YEAR"])
viz_df["shannon_index"] = viz_df["COUNT"].apply(eda.shannon_index)
```

[8]:
```
px.violin(viz_df, x='Season', y='shannon_index', color='Season',␣
 ↪title='Distribution of Bird Diversity Across Seasons', labels={'Season':␣
 ↪'Season', 'shannon_index': 'Shannon Index (Bird Diversity)'}, height=600).
 ↪update_layout(hovermode='x').show()
```



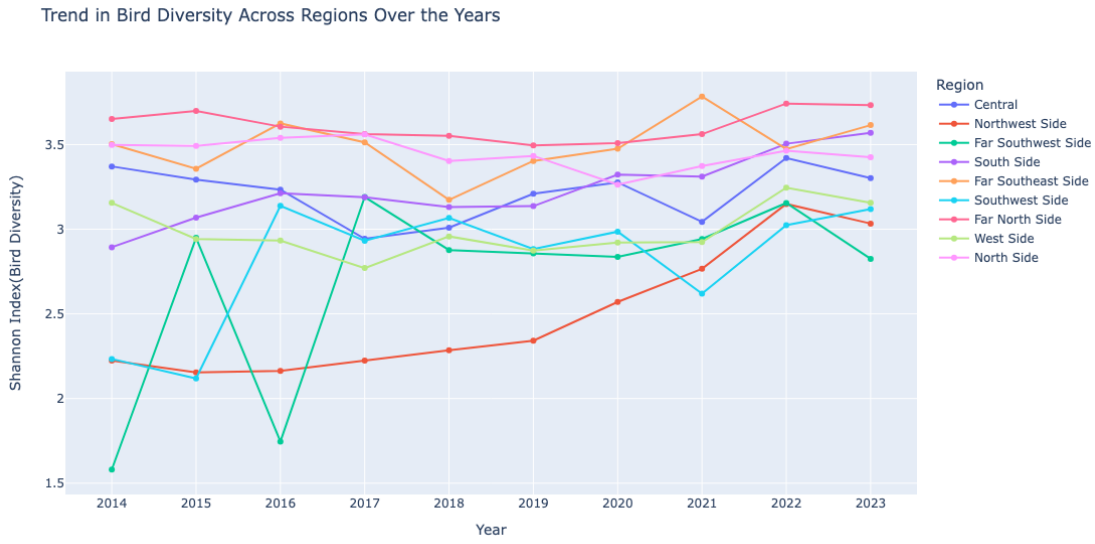

### 0.3.2 2.3 (Bird Diversity by Region)

*The line plot illustrates notable fluctuations in bird diversity across Chicago's regions over the years. Regions like the Northwestern, far South, and South areas displayed significant increases in diversity. Conversely, most other regions demonstrated relatively stable diversity levels with minor fluctuations. However, 2023 showed a general decline in bird diversity compared to the previous year 2022, suggesting possible environmental or ecological shifts impacting avian populations across*

*Chicago.*

[9]:
```
newmap = {}
for key,value in constants.community_location_map.items():
    newmap[key.upper()] = value
ebird_gdf['Region'] = ebird_gdf['community'].map(newmap)
viz_df = eda.aggregate_data(ebird_gdf,["Region","OBSERVATION YEAR","COMMON␣
↪NAME"])
viz_df["COUNT"] =  viz_df["COUNT"].apply(lambda x: sum(x))
viz_df = eda.aggregate_data(viz_df,["Region","OBSERVATION YEAR"])
viz_df["shannon_index"] = viz_df["COUNT"].apply(eda.shannon_index)
```

[10]:
```
fig = px.line(viz_df.sort_values(by="OBSERVATION YEAR"), x='OBSERVATION YEAR',␣
↪y='shannon_index', color='Region',
              title='Trend in Bird Diversity Across Regions Over the Years',
              labels={'OBSERVATION YEAR': 'Year', 'shannon_index': 'Shannon␣
↪Index(Bird Diversity)'},markers=True, height = 600)
fig.update_layout(xaxis=dict(tickmode='linear',␣
↪dtick=1),yaxis=dict(tickmode='linear', dtick=0.5))
fig.show()
```




## 0.4 Section 3: Machine Learning Analysis

[14]:
```
from sklearn.model_selection import train_test_split
```

### 0.4.1 3.1 Aggregate data by month and community

[11]:
```
final_df = eda.aggregate_data(ebird_gdf,["OBSERVATION MONTH", "OBSERVATION␣
↪YEAR", "community","COMMON NAME"])
final_df["COUNT"] =  final_df["COUNT"].apply(lambda x: sum(x))
final_df = eda.aggregate_data(final_df,["OBSERVATION MONTH", "OBSERVATION␣
↪YEAR", "community"])
final_df["shannon_index"] = final_df["COUNT"].apply(eda.shannon_index)
```

[12]:
```
community_counts = final_df['community'].value_counts()
communities_to_remove = community_counts[community_counts == 1].index.tolist()
final_df = final_df[~final_df['community'].isin(communities_to_remove)]
```

### 0.4.2 3.2 Set features and output for ML analysis

[15]:
```
features = ['OBSERVATION MONTH', 'OBSERVATION YEAR', 'community']
target = 'shannon_index'
X = final_df[features]
y = final_df[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)
ml_results = pd.DataFrame(columns=["Algorithm","MSE","RMSE","MAE","R^2"])
```

### 0.4.3 3.3 Run ML models and Store the Metrics

[16]:
```
def run_model(str):
    model = ml.MLFactory.get_instance(str)
    model.fit(X_train,y_train)
    y_pred = model.predict(X_test)
    mse,rmse,mae,r2 = model.calulate_metrics(y_test,y_pred)
    ml_results.loc[len(ml_results.index)] = [str, mse,rmse,mae,r2]
```

[17]:
```
# Baseline with Mean as the prediction value
run_model("Baseline")
```

[18]:
```
# Decision Tree Regressor
run_model("DecisionTree")
```

[19]:
```
# Random forest regressor
run_model("RandomForest")
```

[20]:
```
# Support Vector regressor Linear Kernel
run_model("SVR")
```

[21]:
```
# Gradient Boosting Regressor
run_model("GradientBoosting")
```

[22]:
```
# MLPRegressor(Neural Network) with hidden_layer_sizes=(100, 50) and activation␣
↪function activation='relu'
run_model('NeuralNetwork')
```

[23]:
```
ml_results
```

[23]:

| | Algorithm | MSE | RMSE | MAE | R^2 |
|---|---|---|---|---|---|
| 0 | Baseline | 0.717444 | 0.847020 | 0.680178 | -0.001866 |
| 1 | DecisionTree | 0.533931 | 0.730706 | 0.498759 | 0.254399 |
| 2 | RandomForest | 0.343903 | 0.586433 | 0.407998 | 0.519760 |
| 3 | SVR | 0.720204 | 0.848648 | 0.673786 | -0.005720 |
| 4 | GradientBoosting | 0.398130 | 0.630976 | 0.480008 | 0.444036 |
| 5 | NeuralNetwork | 0.438946 | 0.662530 | 0.533919 | 0.387039 |

## 0.5 Section 4: Results

### 0.5.1 Vizualizations

**Chicago Geographic Map(Geoplot)** - The northern regions have higher bird driversity compared to south.The costal regions also exhibit higher diversity.
**Season Comparison Map(Violin plot)** - The months of **Spring**(March,April,May) witness higher bird observations followed by Summer.
**Region Comparison Map(Line plot)** - Certain areas, such as the Northwestern, far South, and South regions, have experienced particularly pronounced increases in diversity.

### 0.5.2 Machine Learning

**What are we doing in ML?** We are training the models with the input features **OBSERVATION YEAR, OBSERVATION MONTH, COMMUNITY** and output value as **Shannon Index(Bird Diversity)** so that we can predict the bird diversity of these communities in the future.

The **ml_results** dataframe provides a comprehensive overview of the performance metrics for various machine learning algorithms applied to the dataset. Each algorithm's effectiveness is evaluated based on Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2) values.
The Baseline model, serving as a reference point,with its negative R^2 value suggests that it's not a suitable predictor.
Random Forest model provides the most accurate predictions compared to the other algorithms.Decision Tree performs better than baseline but less than the Random Forest. Support Vector Regression (SVR) exhibits a similar negative R^2 value to Baseline which shows the performance is worse than the Baseline

Gradient Boosting and Neural Network models also demonstrate improved performance over the Baseline, but slightly lower than the Random Forest in terms of RMSE and MAE. In summary, based on the above metrics, **Random Forest** emerges as the most promising model for predicting bird diversity.