**Group 370 : INaturalist Observation Data Prediction Model**

Group 370 : Making a prediction model using past observation data to determine the month a species is most likely observed during.

Your submissions:

* 370\_Report.pdf
* 370\_Codes.ipynb (with necessary comments)

Notes

* No extension to the deadline
* Each team can only submit one copy by a single member, just list all of your members in the report
* use RED font for the parts that you revised according to the feedbacks in your presentation

|  |  |  |
| --- | --- | --- |
| First Name | Last Name | Email address |
| Piotr | Rajchel | prajchel@hawk.iit.edu |

Table of Contents

[**1. Introduction** 2](#_Toc56029461)

[**2. Data** 2](#_Toc56029462)

[**3. Problems and Solutions** 2](#_Toc56029463)

[**4. KDD** 2](#_Toc56029464)

[**5. Evaluations and Results** 2](#_Toc56029467)

[**6. Conclusions and Future Work** 2](#_Toc56029470)

# **1. Introduction**

Last semester I made a basic Neural Network model to identify different bird species based on images. This semester, I’m keeping the environmental theme and using INaturalist Species Observation data. After thinking about applications and uses for this data in machine learning, I realized AI can’t randomly generate meaningful observation data, so instead I’ll be using it to determine what month a given species is most likely to be found during. I’ll enter a random species that resides in the states of Illinois, Michigan, Wisconsin, and Indiana, which means it should already exist in the training set, and it will determine what month was most likely for me to find it during. In other words, this information could be used to help you know what month the best month is to go searching for this species.

Another possible idea was to determine an average location or range, but locational data predicted by an ai isn’t that useful in a scenario like this.

# **2. Data**

My dataset was gathered through INaturalist’s Export to CSV function in their observation search. The database is massive, so I decided to narrow down the data to between 2015-2022, to only include amphibians and reptiles, and only in the states of Illinois, Michigan, Wisconsin, and Indiana.

When it comes to preprocessing the data, my first step before even downloading the data was deciding which columns I needed. The database provides a huge number of rows and columns, but many are for storage purposes, so rather than removing them after downloading the database, I did it ahead of time on the website. I also did not include data below “research grade”, meaning that each observation has been reviewed by peers. These are the columns I chose to leave:

A screenshot of a computer

Description automatically generated with medium confidence

# In the code, additional pre-processing included dropping the columns of id, URL, and image URL, as they were reference columns that were unnecessary in creating the data model. Additional dropped or modified rows included:

* **Captive\_Cultivated:** This row is only for when species are recorded that are raised by people, and not wild observations. Not only did I not want that data, but there was also not a single instance of a TRUE row in all the 71k+ rows, so the column was dropped.
* **Positional\_Accuracy:** Geospatial data isn’t too important when I’m looking specifically at observation month, and even if I was using it more, I’m keeping latitude and longitude, and I don’t need exact map location, so this column was removed.
* **Observed\_on\_String:** This column was the same as observed\_on, except with extra time info added onto the dates, so I removed it to reduce redundancy.
* **Time\_Observed\_at:** This column had more info than was useful, and to reduce redundancy, I used regex to remove anything but the 00:00:00 format timestamps and used fillna method ffill to fill missing values.
* **Time\_Zone:** Time values aren’t crucially important, so I used fillna method ffill as well to fill missing values.
* **Place\_Guess:** Replaced missing values with “None”, as there was no guess for the location.
* **Species\_Guess:** Replaced missing values with “None”, as there was no guess for the species.

Those were the initial column changes, but I ended up reducing the data down to only 3 columns of observed\_on, time\_observed\_at, and common\_name, because many of the other columns don’t contain useful information for prediction. For example, the scientific and taxon names are going to be the same as long as the common\_name is the same, so there’s no reason to keep them around as extraneous data.

After presenting, I decided to try adding back the columns just mentioned, as seen in the python book. I’ll also modify the presentation’s images to include the new recordings. I added on precision recall and f1 scores to all 3 as well, though I was unable to run KNN on n-fold cross validation, as it seemed to just freeze after completing hold-out. The Naïve Bayes results actually seemed the best afterwards, and the values for Accuracy, precision, recall, and f1 were all different this time, so I think the results there were correct, if not in the other sections. Here’s the screenshot I’ll include in the presentation as well:

Text

Description automatically generated

# **3. Problems and Solutions**

My problems were to figure out what methods were best suited for creating a prediction model to determine what month a species is most likely to be found in. I was only able to set up the model and run it properly in 3 models.

# **4. KDD**

Much of the initial data processing was just fitting the data to pass through KNN and Naïve Bayes algorithms, and trying whatever I could from what we’ve previously applied to get a prediction classifier working. I ended up using Naïve Bayes, KNN, and Decision Tree to get some prediction results for training and testing sets.

# **5. Evaluations and Results**

The evaluations I used were just N-Fold Cross validation and Hold-out. I wanted to focus on using N-Fold Cross validation, but I couldn’t get other models to work in time, and was having a ton of general trouble. What I provided in the Jupyter notebook pretty much details what I tried within the time frame. Therefore, I also included Accuracy, Precision and Recall using Hold-out evaluation for each of the 3 methods.

# **6. Conclusions and Future Work**

I’m condensing this into one section, because it all ties together. I’m including screenshots of the scores in the presentation, but based purely off of accuracy scores, KNN seemed to do the best.

My limitations included mainly time, and there was also the limit of how much you can truly do with observation data. No prediction is every truly correct, but all we can do is create estimations, which was my aim for the project. I got hung up on many issues, and ended up returning to what we’ve done in class as a base and just pulls values from that in the end. In the future, I’d obviously like to improve this project and put more work into it, as it’s a topic I appreciate and wasn’t able to commit as much time and effort as I’d like into it.

I was happy with my preprocessing work and decisions, even if some of it ended up being scrapped by my decision to only use 3 columns, but that was the extent of what I was happy with coming out of this project. I don’t know if I am able to improve or iterate upon the project any more at this point, due to personal time constraints and issues.