Brown University

DATA 1030

Fall 2020

If we introduce an animal species of specific nativeness, occurrence and seasonality to a National Park - can we predict its abundance?

Author
Vidushi Shukla

GitHub
https://github.com/vidushi-shukla/project-data-1030-vidushi-shukla

December 2, 2020

Contents

1 Introduction

Until recent centuries, ecosystems around the world developed and evolved in relative isolation. In an increasingly globalized world with extensive trade and passenger movement, the introduction of non-native species into ecosystems has evolved into a serious cause for concern [1]. A 2005 study found that invasive species in the US caused losses to the tune of $120 billion with over 42% of species classified as Threatened or Endangered under the Endangered Species Act directly threatened by them [2]. This threat comes in the form of competition for resources as well as the introduction of novel zoonotic diseases. Since their introduction is far more frequent and often accidental, much study and coordinated action has gone into dealing with invasive plant and insect species in National Parks in the US [3]. Meanwhile, invasive animal species have a far reaching impact which a 2020 review reported as minimally addressed by the National Parks Service (NPS) [4]. Of the 1409 reported populations of 311 invasive animal species in US National Parks, a 2018 study found that only 23% had park-wide management plans and only 11% were reported to be "under control" [5].

In this project we use an NPS database of animal and plant species from individual national parks and build a model that aims to solve a classification problem for animal species [6].

1) Target variable: Abundance of the species

2) Number of data points and features: 119,248 (each a specific species in a specific National Park) and 18 respectively; After preparatory exclusion: 49,735 data points and 15 features.

3) Description of features: Species ID (Continuous): Unique ID for each species; Park Name (Categorical): 58 different National Parks; Category (Categorical): Category of species, eg. mammal; Order (Categorical): Scientific order of the species, eg. Carnivora; Family (Categorical): Scientific family of the species, eg. Canidae; Scientific name (Categorical): eg. Canis lupus; Common Names (Categorical): eg. Gray wolf; Record Status (Categorical): Approved or In Review according to internal NPS arrangement; Occurrence (Categorical): Present or Not Confirmed; Nativeness (Categorical): Native or Not Native; Seasonality (Categorical): Resident, Breeder, Migratory, etc.; Conservation Status (Categorical): eg. Endangered; Park Code (Continuous): eg. ACAD; State (Categorical): eg. ME; Acres (Continuous): Area of National Park in acres; Latitude (Continuous): Latitude of center of the park in degrees; Longitude (Continuous): Longitude of the center of the park in degrees; and Abundance (Categorical): Abundant, Common, Uncommon, Occasional, Rare (in decreasing order), Unknown.

4) Since this is a Kaggle database it has been used in other studies. A study by Jonathan Bouchet asked if there were more species in danger in a given park/location. They studied the distribution of species along with their categories across different states and found that there were maximum species under threat in the parks in California and Washington.

2 Exploratory Data Analysis

Preparatory work: Removed the columns: "Species ID", "Common Names" and "Park Code" since they are not predictive of the "Abundance". Converted NaN cells in the DataFrame to the String "Missing" to create a "Missing" category for all the categorical features. Removed all data points for Category "Vascular Plant" and "Non Vascular Plant" since we are only interested in animal species. ∼30 data points were removed due to a data entry error in the original dataset where the categorical values from the "Nativeness" column had been incorrectly assigned to the "Abundance" column (i.e. assigned values of "Native" or "Not Native" in the "Abundance" column).

Left with 49,735 data points and 15 features, including "Abundance".

Fig. 1. shows the Abundance column (our target variable) as a bar plot. The largest category is "Missing" - the user-defined category we made for empty cells. This is followed by Unknown, Common, Uncommon, Rare, Occasional and Abundant in decreasing order of counts.

Fig. 2. summarizes the biodiversity of animal species across the parks through a stacked bar-plot of "Category" vs. "Park Name". We find that the greatest fraction of biodiversity in almost all the parks comes from Birds, with 2 exceptions - the Hawaii Volcanoes National Park and the Haleakalā National Park, together home to the extremely active Kīlauea and Mauna Loa, and the dormant Haleakalā volcanoes. The frequent volcanic activity means that the major fraction of biodiversity here comes from Insects. Similarly, the cold Alaskan parks: Gates of the Arctic and the Kobuk Valley - where few living creatures thrive - show high fractions of biodiversity due to Fungi.

Fig. 3. shows the relationship between biodiversity by Category and Area of the Park through a Box-Plot. The scale of the Area of Park is in 10^6 acres. It is unsurprising that the data for all 12 categories of "Category" cluster towards the lower end of the range between 0 and $2\text{x}10^6$ acres with a few outliers towards the higher end of the scale, since only 2 parks have $> 6\text{x}10^6$ acres and both are in Alaska with a relatively moderate number of species in the dataset. Both are very cold, explaining the absence of outliers for algae, crab/lobster/shrimp and spider/scorpion species.

Fig. 4. is a set of Categorical Histograms for Nativeness vs. Longitude. There are many missing and unknown categories across the range but it is very interesting that there are high spikes for "Not Native" counts on both ends of the Longitude range. This is explained by the fact that the two ends of the Longitude range roughly represent the 2 coasts of Continental US, and there is far greater invasion of non-native species near shorelines due to high human traffic as well as sea-borne goods traffic [7].

3 Methods

3.1 Data Preprocessing

The dataset is not IID since it follows group structure based on which National Park the data has been collected from. The same species could be present in 2 different parks and have different occurrence, nativeness, abundance, seasonality and conservation status. In order to train the model in the best way to be able to predict the Abundance of a species (with a specific set of features) the model hasn't seen before, the data would have to be split to include a representative sample from every National Park. To that end, we use Stratified Split for a 60:20:20 split of the data into training, validation and test sets, with the stratifying column set as "Park Name". OneHotEncoder was applied to categorical features since for nearly all the columns it is not possible to order the categories. StandardScaler was used for the continuous features since they all have a tailed distribution. LabelEncoder was used for the target variable "Abundance" so the classification labels are integers between 0 and (number of classes in Abundance)-1.

The OneHotEncoder generated 17,596 features for the categorical features. LabelEncoder uses 7 classes to generate integers from 0 to 6.

At the end the preprocessed data comprised of 49,735 rows and 17,655 features.

3.2 Evaluation Metrics

Carrying out a unique value count on our target variable, we get:

Missing	0.440676
Unknown	0.239610
Common	0.097316
Uncommon	0.091565
Rare	0.062592
Occasional	0.048839
Abundant	0.019403

Showing that we have a fairly imbalanced multi-class dataset. To that end a good metric for this problem is the f1 score. In this project two "average" parameters for sklearn's f1 score function were used: weighted and micro. Micro calculates metrics globally by counting all true positives, false negatives and false positives, while weighted calculated metrics for each label and finds the average weighted by support, thus accounting for label imbalance.

To calculate the weighted average baseline f1-score we calculate confusion matrices for random assignment for each class and then find the weighted mean (Detailed calculations [8], References [9] [10]). Baseline weighted f1 score = 0.407504.

The micro average f1-score is equal to the overall accuracy of the classifier [10] [11], thus the baseline is = 0.4407 (fraction of points that make up the highest class in the target variable).

3.3 Machine Learning Algorithms

Each pipeline (including splitting, ML algorithm and GridSearchCV) was run using 10 random states.

1. Logistic Regression
   (a) The logistic regression based pipeline:
      i. Employed GridSearchCV
      ii. Stratifying column: "Park Name"
      iii. class weight = 'balanced', solver = 'saga', and parameter grid with penalties: ['l1', 'l2'] and C values: [1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3, 1e4]
      iv. f1-score (average = "weighted")
      v. Output showed terrible scores (averaged for best model over 10 random states):
         A. Mean of train scores: 0.0335 , SD: 0.0797
         B. Mean of validation scores: 0.0329, SD: 0.0783
         C. Mean of test scores: 0.0330 , SD: 0.0773
2. RandomForestClassifier
   (a) The RandomForestClassifier Pipeline
      i. Employed GridSearchCV
      ii. Stratifying column: "Park Name"
      iii. parameter grid with max features: [0.5, 0.6, 0.7, 0.8, 0.9, 1] and max depth: [1, 3, 10, 30, 70, 100]
      iv. f1-score (average = "weighted")

v. Output had scores better than baseline (averaged for the best model over 10 random states):

A. Mean of train scores: 0.5661 , SD: 0.0012

B. Mean of validation scores: 0.5671, SD: 0.0044

C. Mean of test scores: 0.5651, SD: 0.0032

3. XGBoostClassifier

(a) The first XGBoostClassifier Pipeline

i. Due to a memory overflow issue associated with using 17,655 features, one column (Scientific Name) was dropped from the original feature matrix and this brought the number of features down to 2000. This now allowed us to employ XGBoost Classification

ii. Stratifying column: "Park Name"

iii. parameter grid with tunable max depth: [1, 3, 10, 30, 100] and reg alpha: [0, 1e-2, 1e-1, 1e0, 1e1, 1e2]

iv. f1 score (average = "weighted")

v. Output had scores almost equal to baseline (averaged for the best model over 10 random states):

A. Mean of train scores: 0.2694, SD: 0.0016

B. Mean of validation scores: 0.2702, SD: 0.0037

C. Mean of test scores: 0.2709, SD: 0.0037

(b) The second XGBoostClassifier Pipeline

i. Similar to above, one column was dropped from the feature matrix

ii. Stratifying column: "Park Name"

iii. parameter grid with tunable reg-alpha: [0, 1e-2, 1e-1, 1e0, 1e1, 1e2]

iv. f1-score (average = "micro")

v. Output had good scores (averaged for the best model over 10 random states):

A. Mean of train scores: 0.7615, SD: 0.0377

B. Mean of validation scores: 0.7620, SD: 0.0391

C. Mean of test scores: 0.7614, SD: 0.0357

4 Discussion of Results

Comparing with the baseline scores from 3.2, we determine that the RandomForestClassifier and the second XGBoostClassifier pipelines gave better model performance than the baseline cases.

Model	Mean test score	No. of SDs above baseline
RandomForestClassifier (stratified by Park Name)	0.5651	49
XGBoostClassifier (used f1 score average = 'micro')	0.7614	14

It is worth noting that the XGBoostClassifier used average parameter: "micro", which implies both a different baseline and a different method of calculation. Since this dataset is imbalanced it is far more appropriate to have used average: "weighted" as is done in the first XGBoostClassifier pipeline. The pipeline with f1-score(average = "micro") was carried out to gauge if the choice of metric has an effect on performance of the model. The XGBoostClassifier that did use the f1 score with average "weighted" showed scores almost equal to the baseline, suggesting it is not a good choice for our dataset.

The best model is the RandomForestClassifier model. The hyperparameter tuning for the best RandomForestClassifier model gave max-depth = 1, max-features = 0.6.

4.1 Feature importance

As Fig. 5 shows, checking the Permutation Feature importance of the test set for the RandomForestClassifier suggests that only 2 features have a considerable impact on the test score: Occurrence and Nativeness. This is strongly in agreement with what we hypothesized at the start of this project with our question, where we asked if we could predict the abundance of a species in a park, knowing its seasonality, nativeness and occurrence. It also explains why when using GridSearchCV, so many hyperparameter pairings gave identical or very similar scores. It makes logical sense that answering the questions: "Is this species actually present in this park?" and "Is this species native to this park?" have the greatest impact on how abundant the species is in the park.

5 Outlook

1. Perhaps the best way to improve this model would have been to have greater computational power so XGBoost (and SVC, not featured here) could have been run on the full dataset of 17,655 features.

2. The RandomForestClassifier gave the best performance for the feature matrix with 17,655 features but has scope for improvement in its test scores.

3. More elaborate hyperparameter tuning would likely have improved the scores since many pairings of hyperparameter gave identical scores for the same split of data, especially for RandomForestClassifier.

4. Collecting better data would lead to improved classification. There are many missing values in the categorical features, including the target variable, which has 40% missing values.

5. Since occurrence and nativeness are highly predictive of the abundance of a species, it would be useful to have more detailed data for these 2 categories.

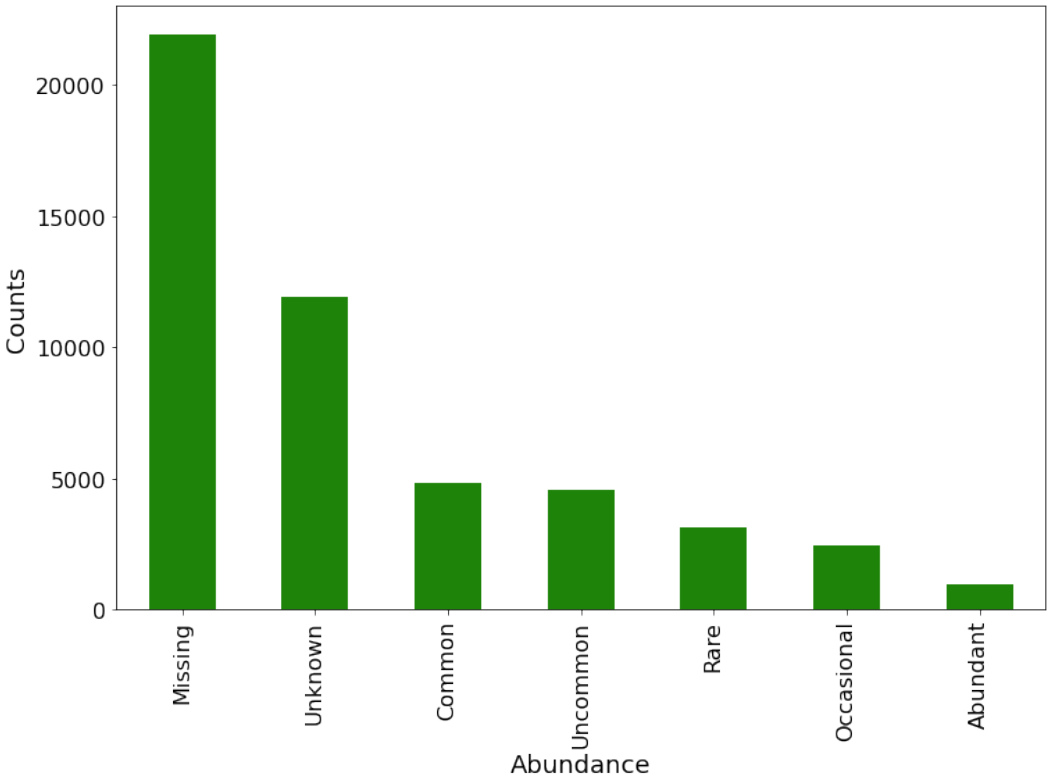

Fig. 1: Bar plot for "Abundance"

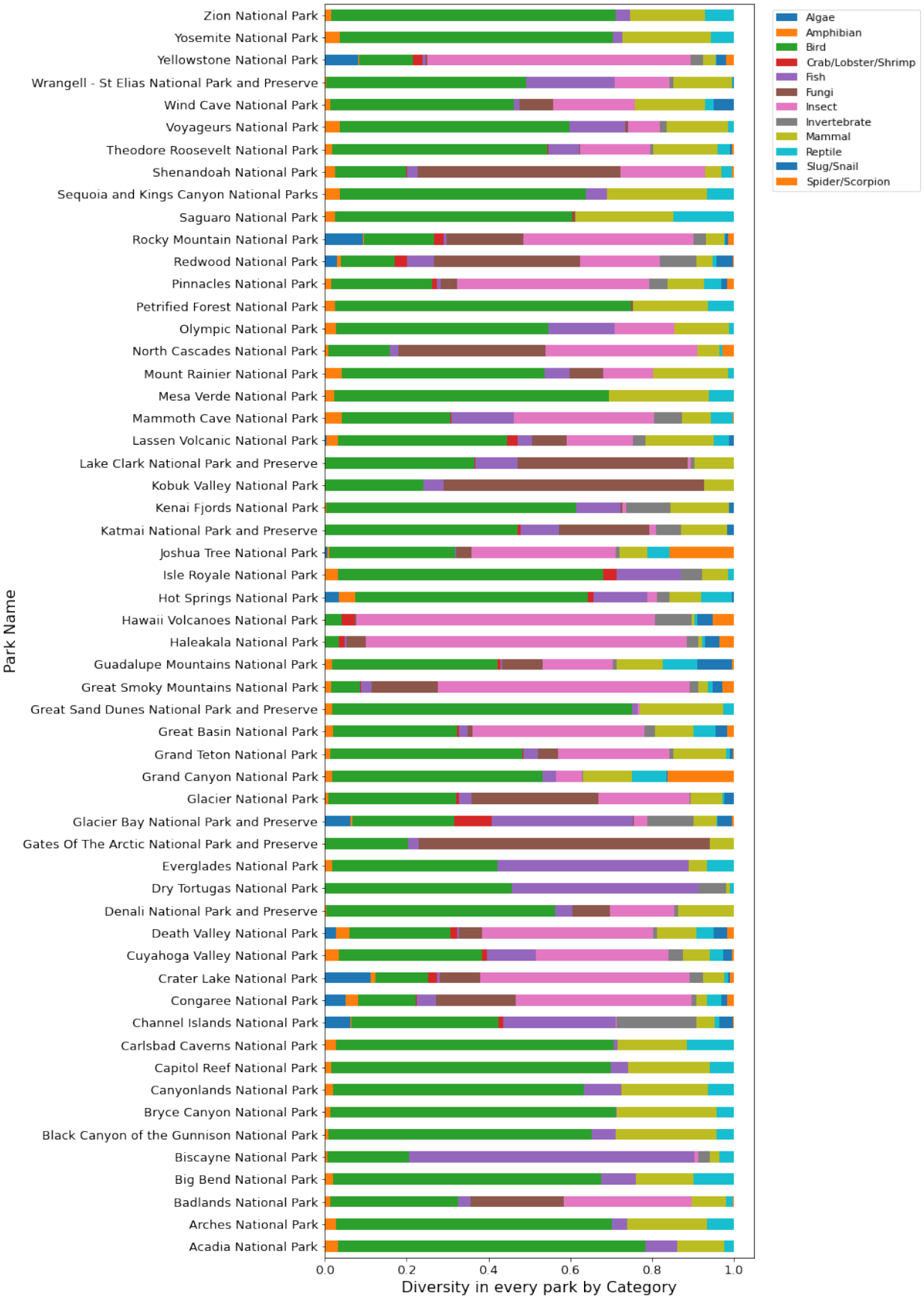

Fig. 2: Stacked bar plot of Category vs. Park Name

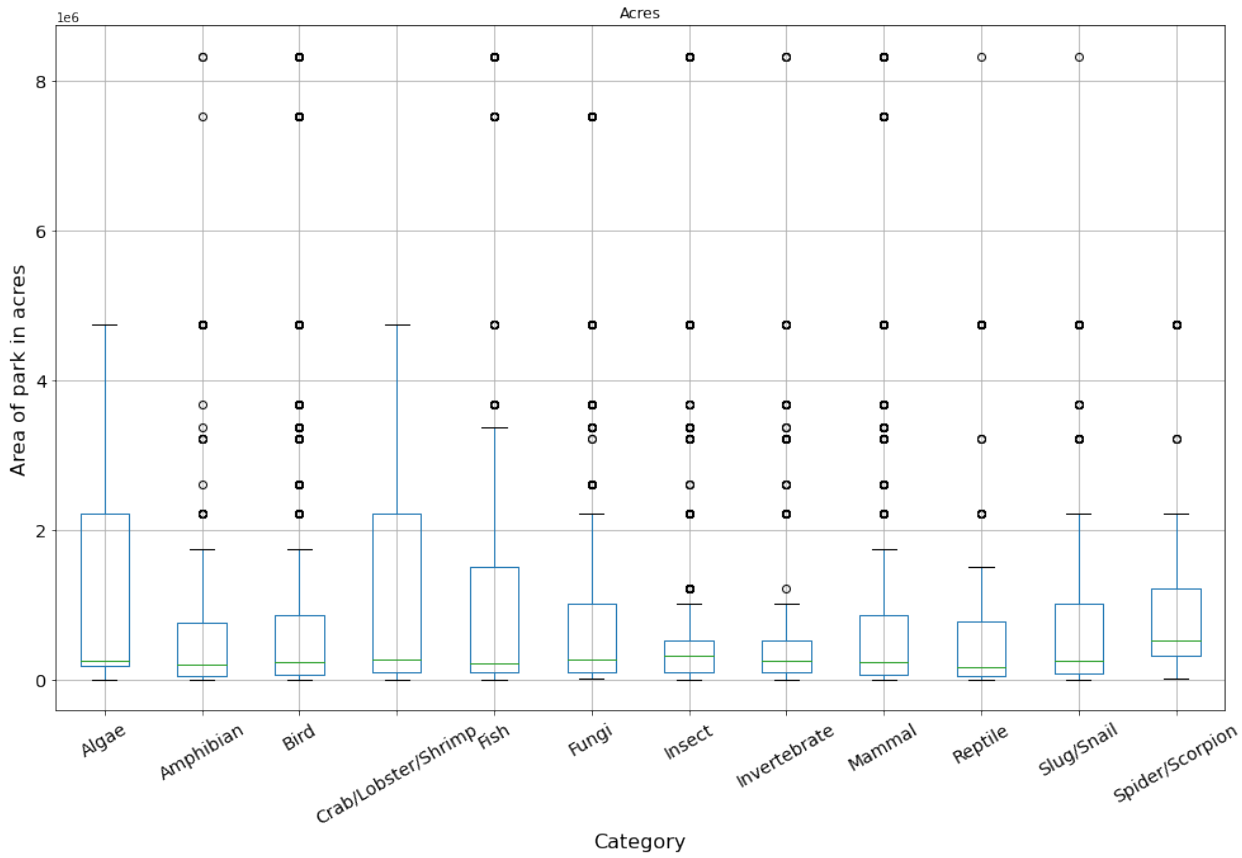

Fig. 3: Box plot of Category vs. Area in acres

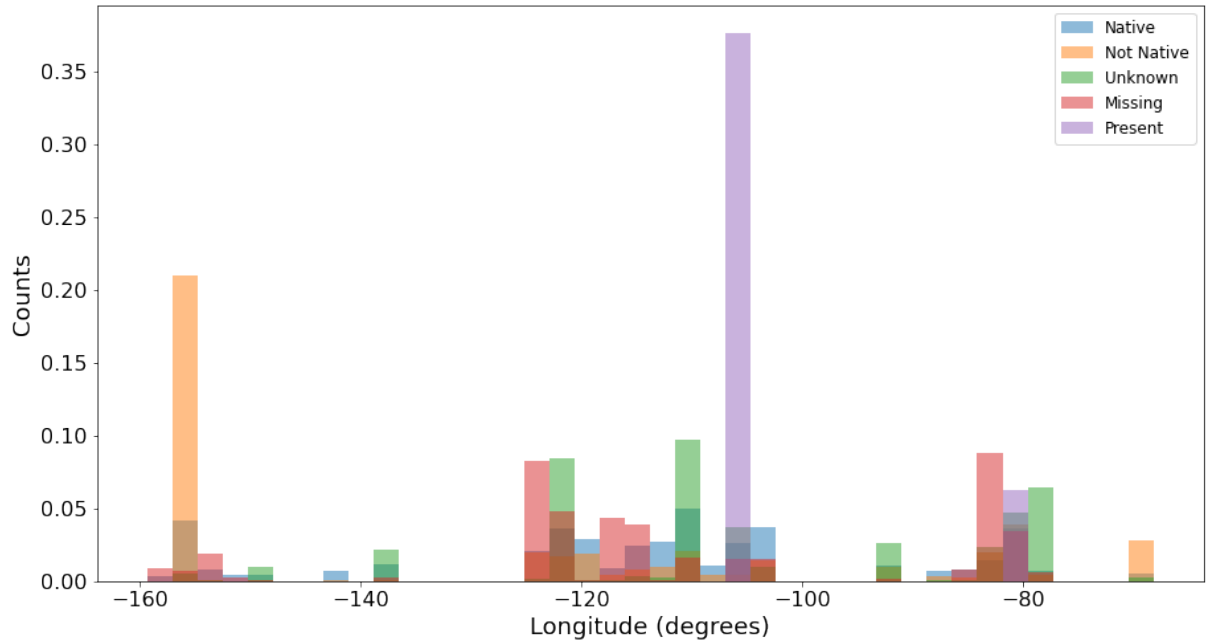

Fig. 4: Categorical histograms of Nativeness vs Longitude

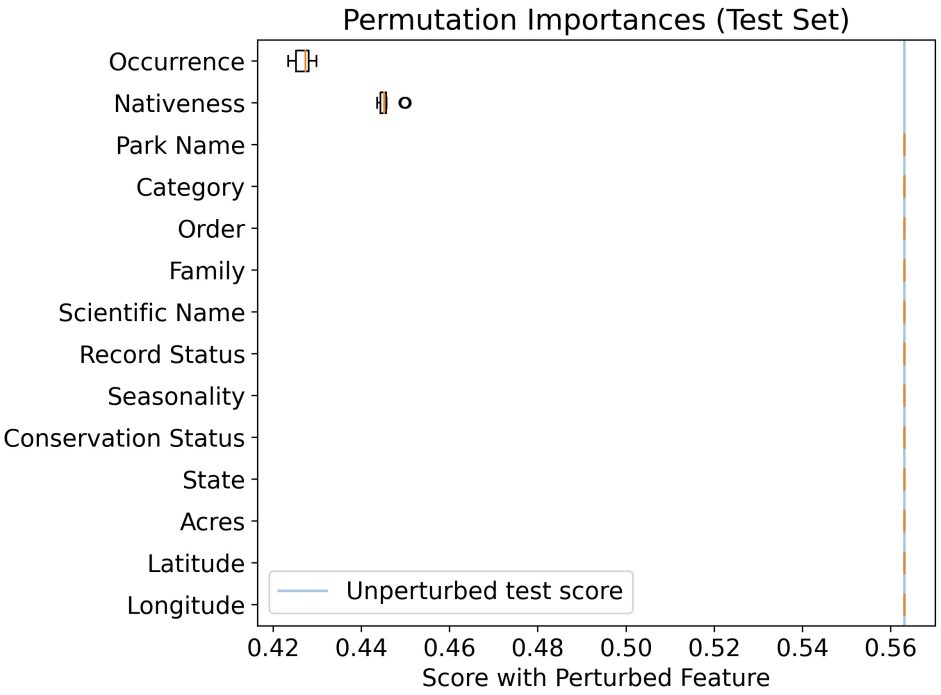

Fig. 5: Permutation Importances of Features for RandomForestClassifier

References

[1] S. Lowe, M. Browne, S. Boudjelas, and M. De Poorter, *100 of the world's worst invasive alien species: a selection from the global invasive species database*, vol. 12. Invasive Species Specialist Group Auckland, 2000.

[2] D. Pimentel, R. Zuniga, and D. Morrison, "Update on the environmental and economic costs associated with alien-invasive species in the united states," *Ecological economics*, vol. 52, no. 3, pp. 273–288, 2005.

[3] R. F. Hughes, G. P. Asner, J. Mascaro, A. Uowolo, and J. Baldwin, "Carbon storage landscapes of lowland hawaii: the role of native and invasive species through space and time," *Ecological Applications*, vol. 24, no. 4, pp. 716–731, 2014.

[4] A. A. Dayer, K. H. Redford, K. J. Campbell, C. R. Dickman, R. S. Epanchin-Niell, E. D. Grosholz, D. E. Hallac, E. F. Leslie, L. A. Richardson, and M. W. Schwartz, "The unaddressed threat of invasive animals in us national parks," *Biological Invasions*, vol. 22, no. 2, pp. 177–188, 2020.

[5] J. Resnik, "Biodiversity under siege, invasive animals and the national park service: a state of the knowledge report," *Natural Resource Report NPS/NRSS/BRD/NRR—2018/1679. National Park Service, Fort Collins, Colorado*, 2018.

[6] "Biodiversity in national parks," *National Park Service*, https://www.kaggle.com/nationalparkservice/park-biodiversity.

[7] C. Simkanin, I. Davidson, M. Falkner, M. Sytsma, and G. Ruiz, "Intra-coastal ballast water flux and the potential for secondary spread of non-native species on the us west coast," *Marine Pollution Bulletin*, vol. 58, no. 3, pp. 366–374, 2009.

[8] V. Shukla, "Calculation of baseline weighted f1 score," *https://github.com/vidushi-shukla/project-data-1030-vidushi-shukla/tree/main/data*.

[9] B. Shmueli, "Multi-class metrics made simple, part i: Precision and recall," *https://towardsdatascience.com/multi-class-metrics-made-simple-part-i-precision-and-recall-9250280bddc2*.

[10] B. Shmueli, "Multi-class metrics made simple, part ii: the f1-score," *https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1*.

[11] "Micro f1 score is equivalent to accuracy," *https://stackoverflow.com/questions/37358496/is-f1-micro-the-same-as-accuracy*.