Using Acoustic Telemetry Data and Environmental Variables to Predict White Shark Presence in Southern California

Final Report

Burns, Echelle
Jan 2020

1. INTRODUCTION

1.1 Background

Acoustic telemetry is a common tool used in research projects that focus on the movement patterns of species that are resident to seawater environments. Individuals are tagged with small, acoustic transmitters (hereby transmitters) that emit an ultrasonic signal that can be heard by acoustic receivers (hereby receivers) that are deployed nearby (within 0 m to 1 km away, depending on the environmental conditions). Similar to the FasTrak lane on the highway, receivers act as a toll booth, listening for transmitters that come into its detection radius, and transmitters act as the FasTrak beeper in your car, sending out a signal to be picked up by the toll booth (receiver). When a transmitter is within the detection range of a receiver, the receiver then records the date, time, and transmitter ID code of the tagged animal in the area. This method of acoustic telemetry is termed "passive" acoustic telemetry, because it does not require researchers to actively follow individuals in real-time. Instead, the receivers remain stationary in a single area for as little as a few days and to as much as an entire year, collecting data streams as they come. When receivers are recovered and downloaded, the resulting dataset tells researchers when a particular individual was within a specific area.

1.2 The Problem

Research studies that focus on large, predatory species like sharks, typically attract more media attention and public enthusiasm than other, less charismatic species. White Sharks (*Carcharodon carcharias*), in particular, appear to be one of the more popular species, as they are commonly addressed by news broadcasters and the entertainment industry (e.g., Jaws). In southern California, juvenile White Sharks show seasonal and sometimes annual fidelity to particular coastal beach habitats, meaning that individuals return to the same beaches year after year. These beaches are often coined 'hotspots' for juvenile White Sharks, and individuals may stay in these hotspot areas for up to a few months before moving to another region. Researchers hypothesize that these movements are primarily temperature driven, as juvenile White Sharks are not yet capable of fully controlling their internal body temperatures[1]; the sharks move towards warmer waters when it becomes too cold. However, no studies have explored whether this hypothesis is correct, or how other environmental parameters affect White Shark movement. By determining which factors are most influential in White Shark movement, we may be able to predict when and where future hotspots will arise. Because juvenile White

[1] Dewar, H., Domeier, M., & Nasby-Lucas, N. (2004). Insights into young of the year white shark, Carcharodon carcharias, behavior in the Southern California Bight. *Environmental Biology of Fishes*, *70*(2), 133-143.

Sharks frequently use coastal beaches that are shared with beach-goers, predicting White Shark movements may help to reduce the probability of shark-human interactions by giving lifeguards advanced notice of when White Sharks are likely to be present. Therefore, the purposes of this project are 1) to determine which combinations of environmental conditions are most likely correlated with juvenile White Shark presence in southern California, and 2) to predict the ranges and categories of conditions that are most likely to result in the presence of a White Shark.

1.3 Specific Clients

This project is geared towards researchers who may be studying similar species or may be trying to answer similar research questions. In addition, this project may be useful for lifeguards or other public safety personnel who may benefit from advanced knowledge of potential threats within their jurisdictional region.

2. MATERIALS AND METHODS

2.1 Dataset Sources and Cleaning Procedures

A receiver deployment log was provided by the CSULB Shark Lab. This file included information on when and where acoustic receivers were in the water, with parameters such as Receiver SN (serial number), Latitude, Longitude, DateTime In (local time), DateTime Out (local time), and Station Name. Receivers that were lost or still deployed in the water (receivers with NAs in either time column) were removed from the dataset and the datetimes were converted to UTC. The count of receivers per grid cell per day were calculated and saved as Receiver Density. In addition, the geospatial range from this dataset was calculated and used to gather environmental datasets from NOAA CoastWatch.

Acoustic receiver detection data were also provided by the CSULB Shark Lab and included fields such as Receiver ID, Transmitter Code, DateTime (UTC), Latitude, and Longitude. These data were pre-processed by the Shark Lab to include only individuals that were detected by an acoustic receiver two or more times per day. This dataset was cleaned to only include data during times that receivers were known to be in the water, in order to reduce the probability of false detection, when a receiver suggests an animal was present when it really was not. In addition, only sharks that were on the CSULB Shark Lab's tag deployment datasheet were included in the final dataset. The tag deployment datasheet was merged with detection data to provide information on the tagging cohort (year) of the individual, as well as its weight (kg) at tagging, total length (cm) at tagging, sex, and tagging location (city). These parameters were not used for the current project, but may be useful for future analyses.

Daily environmental datasets for sea surface temperature, chlorophyll-A and sea surface salinity, and overall seafloor depth gradient data were gathered from NOAA CoastWatch's ERDDAP and were downloaded to a local server. Each of these environmental datasets have been hypothesized to influence the behavior of other marine species. Due to the large spatial extent, and resulting large file size, of these datasets, data were saved in one-year increments. Moon phases were also calculated using the *pylunar* python library.

2.2 Data Wrangling

See: SpringBoard-Capstone1/Data_Wrangling

A shapefile of grid cells (0.01 x 0.01 degrees) was created in ArcGIS in order to standardize and merge the environmental datasets with the shark detection data. The grid ID number over which individual points were located was saved to each dataset. This allowed for the comparison of data across grid cells, as opposed to across individual latitude/longitude combinations. Several environmental datasets did not extend far enough into shallow waters to overlap with grid cells that contained receivers or shark detections. In addition, each individual dataset had its own unique set of latitude and longitude values. Therefore, the mean latitude and longitude for each zone were calculated as the average latitude and longitude across all geospatially-referenced datasets. Once averaged, missing data were filled using values from the nearest grid cell for that parameter during that day. In order to accomplish such filling, the data had to be ordered by date, mean latitude, and mean longitude. Additionally, in order to calculate moon phase, mean latitude and mean longitude coordinates had to first be converted from decimal degrees to tuples of (degrees, minutes, seconds) and datetime had to first be converted to tuples of (year, month, day, hour, minute, second). The appropriate conversions were conducted before pylunar was called.

Due to the high abundance of zeros in the dataset, a jackknifing technique was performed in order to produce a dataset with a relatively equal number of data points between times when sharks were present and when sharks were absent. This was done in two separate ways. The first method was used prior to testing different machine learning algorithms (Sections 2.3, 3.1, 3.2). Zone IDs in which sharks were detected each year were gathered and a list of nearby zones (± 10 IDs away) was generated. A random sample of data with no sharks detected (non-shark dataset) was then taken. The random sample only collected data from zones that were within the zone list that was generated. The size of random samples varied across years, but were equal to the number of rows in the dataset in which sharks were present. Sometimes, the non-shark dataset with zones within the zone list was not large enough to perform the random sampling. In this case, all data within the non-shark dataset with relevant zones were gathered, and the remaining rows of data were randomly grabbed from the non-shark dataset with non-relevant zones. This method did not filter for rows of data in which receivers were present (receiver densities of 0 were present in this dataset). The second method of jackknifing took place after machine learning had begun (Section 3.2). This is because the models used early in the machine learning portion were exhibiting predicted values that were highly accurate, although preliminary data analysis showed that most environmental predictors had no significant relationship to shark presence or absence. One hypothesis for this outcome was that the machine learning algorithms were using receiver density as a hard predictor for shark presence; if no receivers were deployed, there were no sharks in the area. Although technically true, this increases the probability of a false negative, in which sharks are present, but there is no technology in the water to detect it. Therefore, the second method of jackknifing took all rows of data that had receiver density values greater than 0, regardless of whether a shark was present or not. Making sure the data were gathered across similar zones and zones in which receivers were present was important to reduce biases that may arise from comparing shark data that primarily occurs in nearshore waters with non-shark data that may extend for several hundred miles offshore.

Finally, after jackknifing, the data were grouped by zone and date, and the total number of sharks were calculated. Therefore, the final dataset included the following columns: Zone, Date, Month, Year, Number of Sharks, Receiver Density, Sea Surface Temperature, Sea Surface Salinity, Chlorophyll-A, Seafloor Depth Gradient, and Lunar Phase.

2.3 Exploratory Data Analysis and Visualization

See: SpringBoard-Capstone1/Data_Story, Prelim_Data_Analysis

All data were visually assessed for outliers and descriptive statistics were conducted. If found, outliers were removed from the dataset. Prior to analyses, zeros were temporarily removed in order to determine whether there was a relationship between the total number of sharks detected within a zone per day and each parameter. For continuous datasets (sea surface temperature, sea surface salinity, chlorophyll-A, depth gradient) histograms were plotted and the mean and all four quartiles were calculated. However, there was high variation (0 - 20) in the number of sharks that were detected at each environmental value. Therefore, a new, categorical column was added that indicated simply shark presence (1) or shark absence (0). For each continuous environmental parameter, a t-test was conducted in order to determine whether the range of values for the parameter were significantly different when sharks were present or absent. In addition, the variances of each variable were plotted over scales of days, weeks, and months in order to identify any substantial trends that may bias results.

For categorical data (year, zone, receiver density, lunar phase), a barplot was created and the counts in each category were calculated. Boxplots and One-Way ANOVAs were also created and conducted in order to determine whether significantly more sharks were present at different levels.

For data that were associated with time (number of sharks detected, receiver density, Sea Surface Temperature, Sea Surface Salinity, Chlorophyll-A), time series plots were also created in order to visualize potential annual and seasonal trends.

2.4 In-Depth Analysis Using Machine Learning

All categorical variables (Zone, Month, Year, Lunar Phase) underwent one-hot encoding via *pandas* get_dummies() and data were split into random training and testing data prior to being analyzed by machine learning algorithms. Decision Tree, Gradient Boosting, Stochastic Gradient Descent, K Nearest Neighbors, Naive Bayes, and Gaussian Process classifiers via sklearn were all attempted to analyze this dataset (*SpringBoard-Capstone1/Machine_ Learning/Machine_Learning)*. Originally, machine learning techniques were run using the jackknifed dataset of values that were generated using the first jackknifing method. Data were not balanced prior to initial machine learning runs, which allowed for parameter testing of balanced and unbalanced sample weights. Grid searches and *for* loops with several potential hyperparameter values were used to optimize the hyperparameters of the models. Whether grid searches or *for* loops were used depended on the computational time required to conduct a grid search with five folds. Precision and recall estimates were used to identify the best model.

After initial models were run, models were refined using a dataset that was created using the second jackknife technique that only kept rows of data in which receivers were present (receiver density > 0, *SpringBoard-Capstone1/Machine_Learning/Machine_Learning_2 -Refined_Models*). This helped to reduce the probability of false negatives within the dataset. Even after the new jackknifing technique was implemented, there were nearly twice as many rows without sharks present compared to rows with sharks present. Therefore, instances in which sharks were present were sampled with replacement until there was an equal number of rows with sharks present and sharks absent. This technique was chosen because the entire dataset had less than 10,000 rows. Decision Trees, Gradient Boosting, and K Nearest Neighbor Classifiers were used to analyze this dataset.

A Decision Tree was also used to determine the minimum required sample size for optimal machine learning results by randomly jackknifing the dataset for 2^8 … 2^n … 2^{14} random samples and running the Decision Tree Classifier. Model output precision values were calculated using cross_val_score from *sklearn* with a total of three folds to ensure that each category (sharks vs no sharks) had at least 5 samples in each model run.

3. RESULTS

3.1 Exploratory Data Analysis and Visualization

See: SpringBoard-Capstone1/Prelim_Data_Analysis/Preliminary_Data_Analyses and Presence v Absence

Sea Surface Temperature values ranged between 10.99ºC and 25.02ºC. Temperature values when sharks were present showed a mean (± SD) of 18.31 ± 2.81ºC, and values of 18.39 ± 2.65ºC when sharks were absent (Figure 1). T-test results indicated no significant difference in temperature when sharks were present compared to when they were absent (t_{stat} = -1.824, p = 0.068). There were some trends in daily and weekly temperature variance, in which days and weeks appeared to alternate between high and low variance values (Figure 2). However, this relationship was not as clear on a monthly cycle, indicating that grouping the data by month might be appropriate to add to machine learning algorithms without introducing forms of bias.

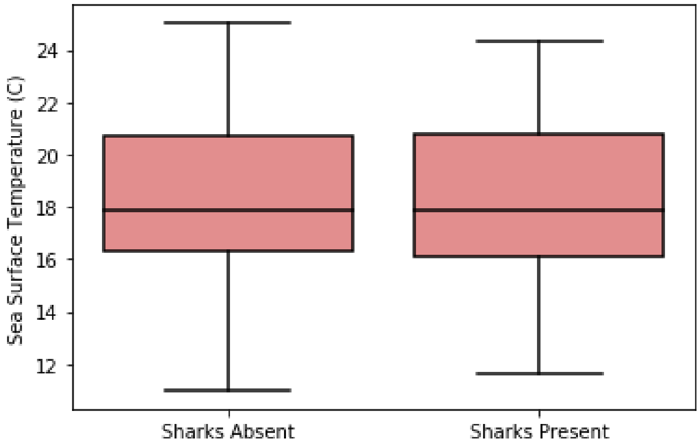

Figure 1. Sea Surface Temperature when sharks are present vs absent.

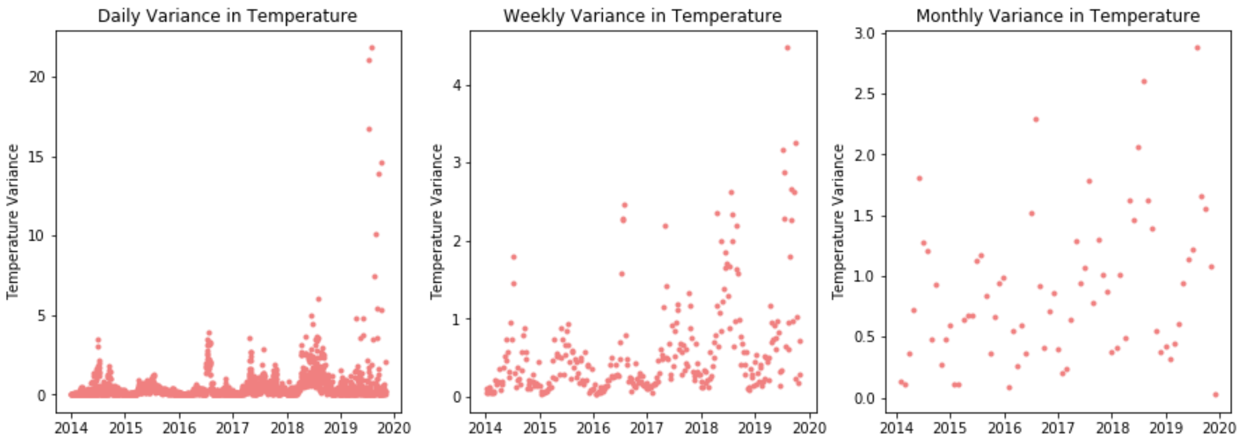

Figure 2. Sea Surface Temperature variance on a daily, weekly, and monthly time scale.

Sea Surface Salinity values ranged between 30.00 and 39.44 psu. Salinity values when sharks were present showed a mean (± SD) of 33.29 ± 1.27 psu, and values of psu when sharks were absent (Figure 3). T-test results indicated no significant difference in salinity when sharks were present compared to when they were absent (t_{stat} = 0.199, p = 0.842). No clear trends were observed between daily, weekly, or monthly salinity variances (Figure 4).

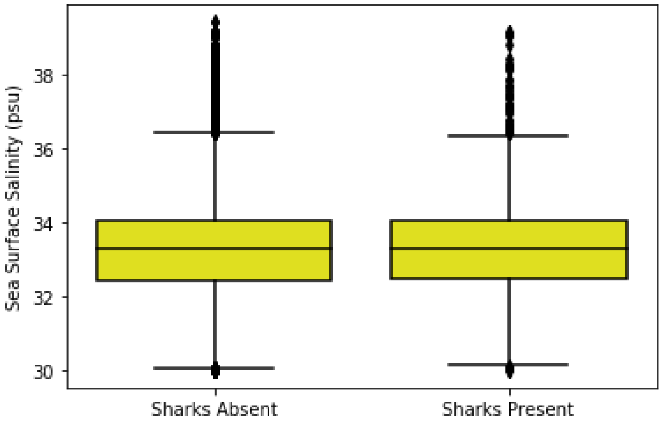

Figure 3. Sea Surface Salinity when sharks are present vs absent.

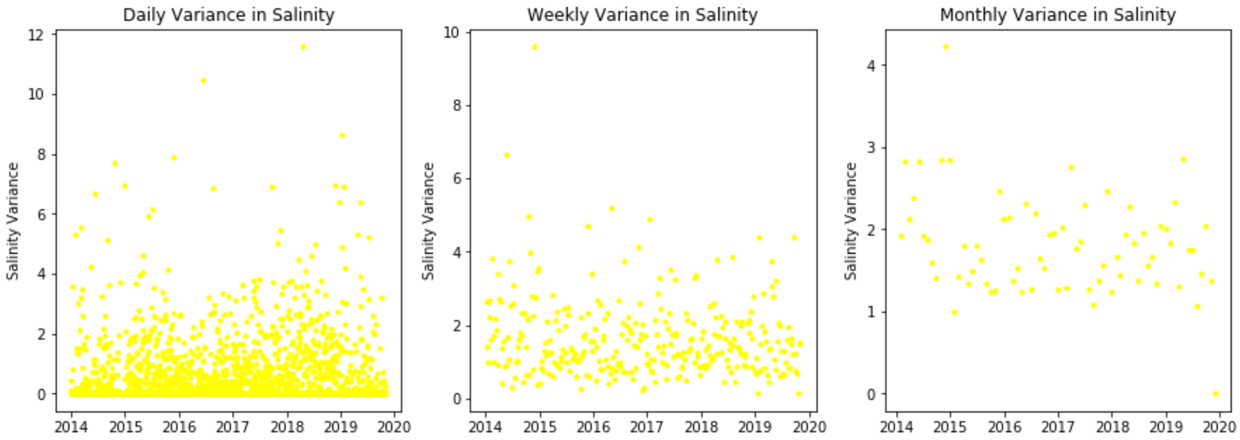

Figure 4. Sea Surface Salinity variance on a daily, weekly, and monthly time scale.

Chlorophyll-A values ranged between 2.27e-13 and 359.25 mg/m^3. Chlorophyll levels when sharks were present showed a mean (± SD) of 0.72 ± 5.46 mg/m^3, and values of 0.86 ± 7.50 mg/m^3 when sharks were absent (Figure 5). T-test results indicated no significant difference in chlorophyll values when sharks were present compared to when they were absent (t_{stat} = -1.119, p = 0.263). When looking at the variance of chlorophyll values across days, weeks, and months, there were some instances in which variance values were abnormally high (Figure 6). However, this did not occur too often (~ 17 days out of the dataset) so values were kept in the dataset to pass to machine learning algorithms.

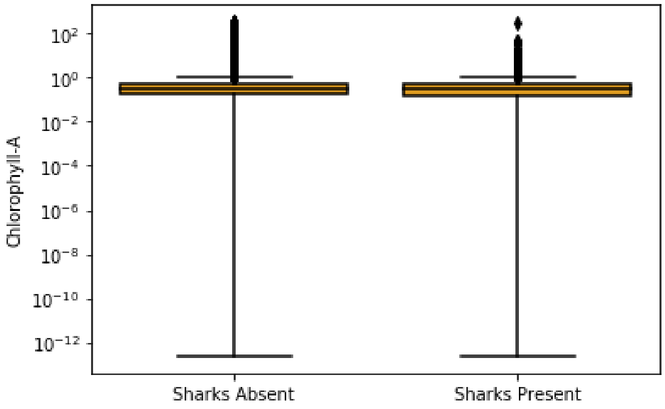

Figure 5. Chlorophyll-A when sharks are present vs absent

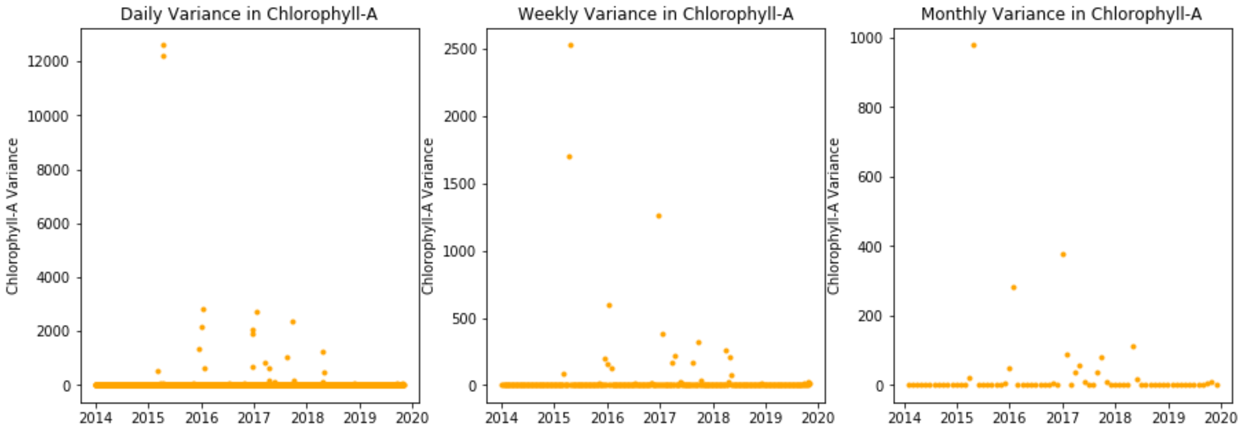

Figure 6. Chlorophyll-A variance on a daily, weekly, and monthly time scale.

Depth gradient values ranged between 0.0009 and 0.0233 m. Gradient levels when sharks were present showed a mean (± SD) of 0.015 ± 0.004 m and values of 0.009 ± 0.005 m when sharks were absent (Figure 7). T-test results indicated a significant difference in depth gradient values when sharks were present compared to when they were absent (t_{stat} = 84.668, p < 0.01). There was an interesting trend between depth gradient variance across days, weeks, and years, during which there is a higher variance in more recent years, beginning in 2016 (Figure 7). However, this is unexpected, because seafloor depth gradients were stationary over time (only one set of values was used for the entire study period). Therefore, it may be possible that these variances are an artifact in the dataset of regions in which receivers are placed. More receivers have been deployed in recent years, so there may be more coverage in regions of steeper sloping seafloors.

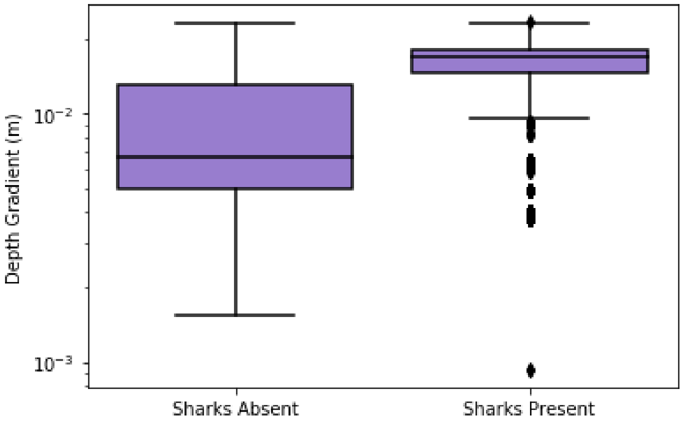

Figure 7. Depth gradient when sharks are present vs absent

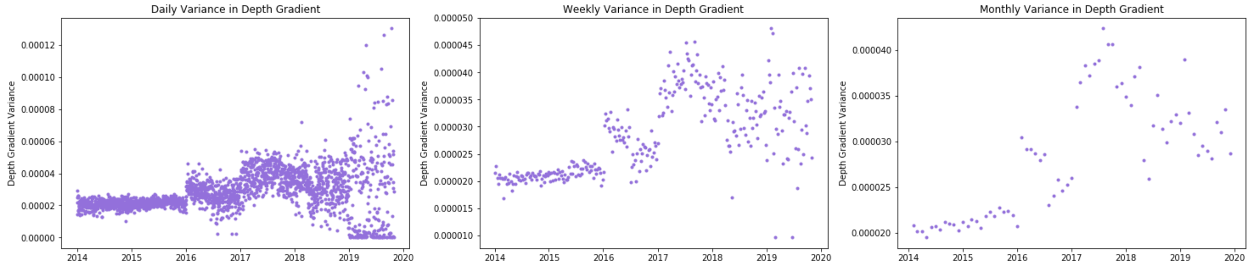

Figure 8. Depth gradient variance on a daily, weekly, and monthly time scale.

Most data were present for years 2015 and 2017, and the least amount of data was present for 2019 (Figure 9). ANOVA results indicated that years 2015, 2017, and 2018 showed significantly higher shark densities compared to 2014 ($p < 0.01$), that 2016 showed statistically similar shark density values compared to 2014 ($p = 0.153$) , and that 2019 showed significantly smaller shark density values compared to 2014 ($p = 0.028$).

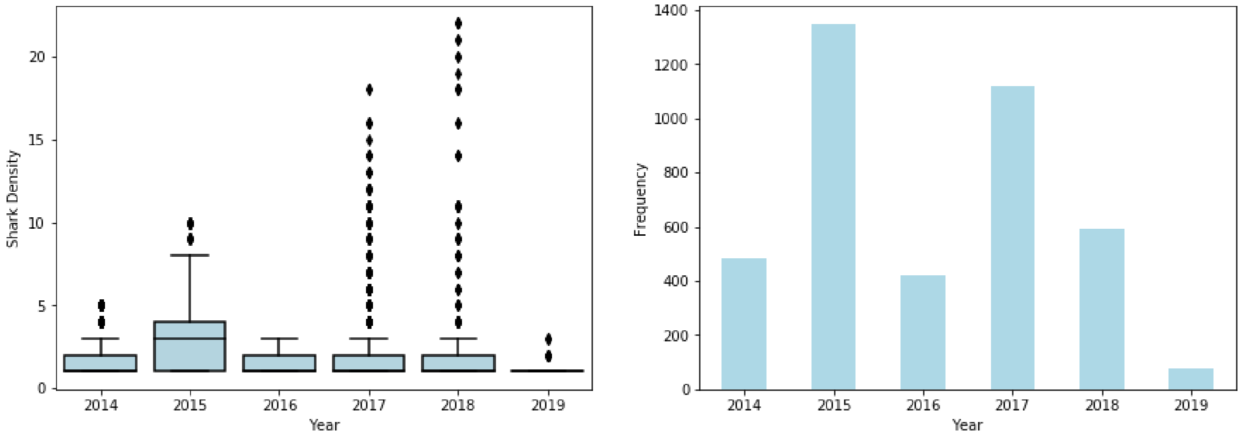

Figure 9. Left: Shark density across years, **Right:** Frequency of data points corresponding to different years.

Only some zones showed high shark density values, which may indicate that some zones are favored by juvenile White Sharks (Figure 10. Of the 50 zones that contained sharks, zones corresponding to smaller regions in Santa Barbara, Santa Monica Bay, Long Beach, and Dana Point showed significantly higher shark densities compared to other regions ($p < 0.01$).

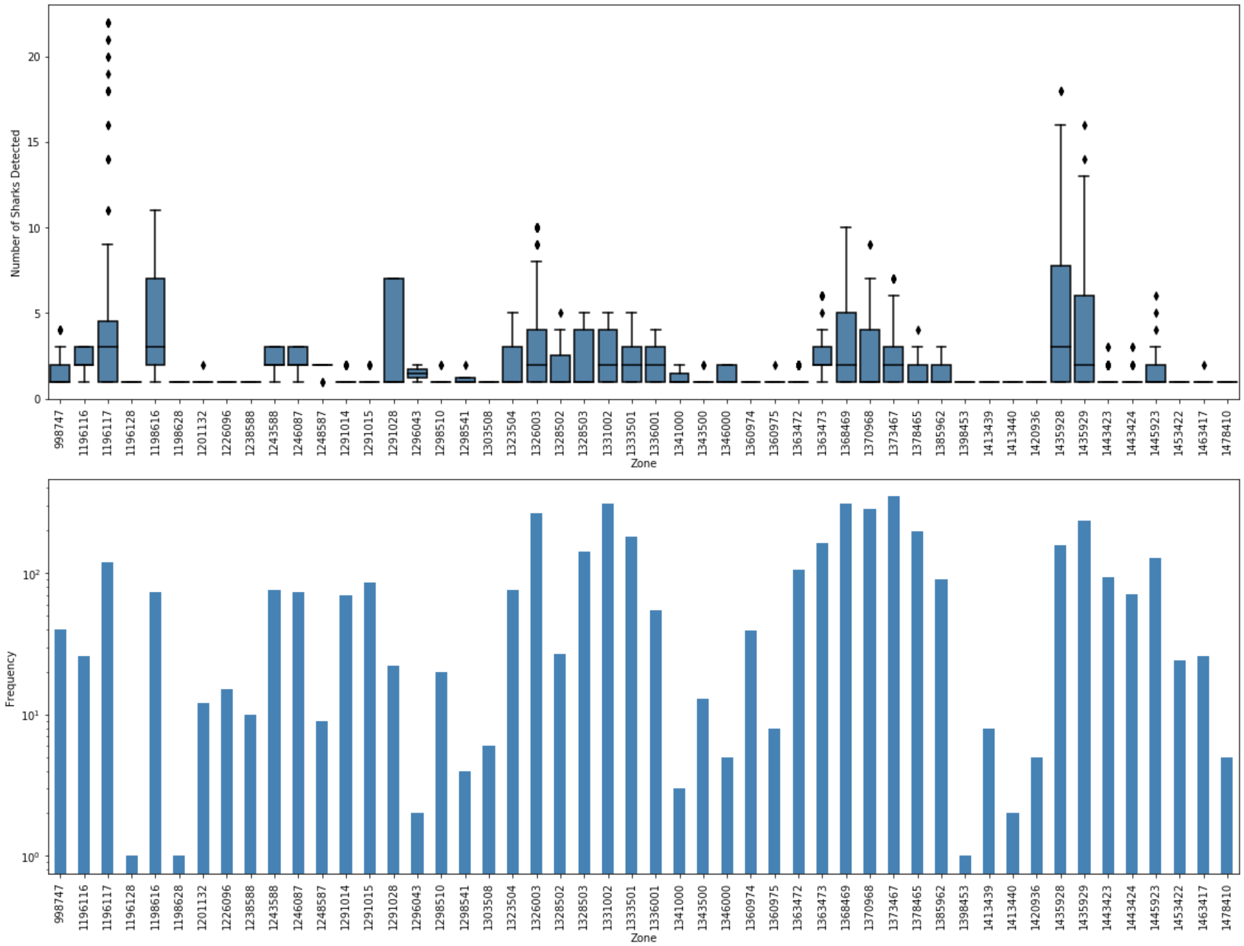

Figure 10. Left: Shark density across zones, **Right:** Frequency of data points corresponding to different zones.

Receiver density values ranged between 0 receivers and 3 receivers per grid cell per day. Because it is impossible to detect a tagged shark within a grid that has no receivers, only data that included one or more receiver were used for preliminary analyses. ANOVA results indicated that there were significantly higher shark densities in regions with two receivers deployed compared to regions with one receiver deployed ($p < 0.01$), but that there were similar shark densities in regions in which three receivers were deployed compared to regions with one receiver deployed ($p = 0.625$). This difference between the two and three receivers deployed and shark density is likely due to fewer instances of three receivers being deployed in a single grid cell (Figure 11).

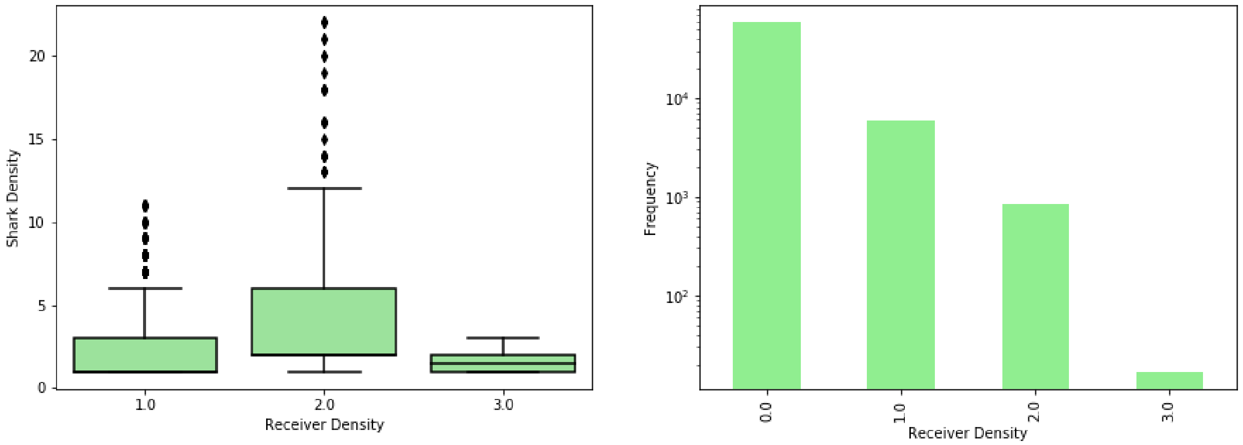

Figure 11. Left: Shark density across receiver densities, **Right:** Frequency of data points corresponding to different receiver densities.

Lunar phases that included the most shark detection data were the waxing and waning crescent and gibbous phases (Figure 12). Alternatively, the new moon, first quarter, full moon, and last quarter phases occurred less frequently in the dataset. Despite differences in the difference of distributions across lunar phases, an ANOVA found no significant difference in shark density across all lunar phases (p > 0.1)

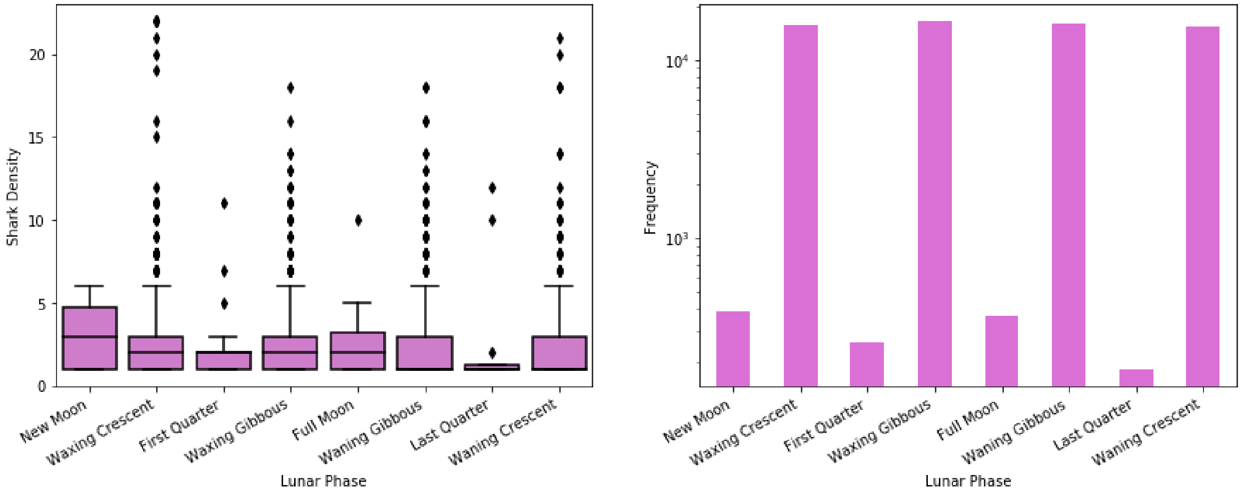

Figure 12. Left: Shark density across lunar phases, **Right:** Frequency of data points corresponding to different lunar phases..

3.2 Preliminary In-Depth Analysis Using Machine Learning

See: SpringBoard-Capstone1/Machine_Learning/Machine_Learning

Class weights were incorporated into a Decision Tree Classifier as ‘balanced’ or ‘None’ for a range of max depth parameters. These models produced two very different trends in accuracy plots. The non-balanced weight accuracy values appeared to plateau at max depth values around 8, while the balanced weight accuracy values appeared to show a consistent, positive linear relationship as max depth values increased. Using a grid search showed that the best model using a Decision Tree Classifier is one without class weights and with a maximum depth of 10. This model led to precision and recall values of 0.996 and 0.985 for the testing dataset, respectively.

Precision, recall and accuracy values that were calculated for a Gradient Boosting Classifier with different levels of learning rates indicated higher scores at learning rates less than 0.5, and much lower scores at learning rates larger than 0.5. However, all scores were greater than 0.90. Using a grid search that incorporated a variety of values for learning rate, max depth and number of estimators indicated that the best model using a Gradient Boosting Classifier had a learning rate of 0.075, a max depth of 3 and a number of estimators equal to 100.

A grid search was attempted for a Stochastic Gradient Descent Classifier for several parameters. However, due to the large size of the dataset (once one-hot encoding took place) and the large lists of hyperparameters that were possible, the code took more than 12 hours to run and was, therefore, terminated. However, running a Stochastic Gradient Descent Classifier with the default hyperparameters produced a model with 0.99 precision and 0.99 recall for the testing dataset.

Models were assessed using K Nearest Neighbor Classifiers with a possible number of neighbor values ranging from 1 to 30 and using uniform and distance weighting. Precision and recall values for uniformly weighted data points found that a value of 10 nearest neighbors produced sufficiently high recall and precision (~0.991) for the testing dataset, while the precision and recall values for distance weighted points found a value of around 9 nearest neighbors produced a sufficiently high recall around 0.9925 and accuracy around 0.9945 for the testing dataset. A grid search was attempted using a variety of more neighbor estimates, but the code was running for five hours before it was terminated.

Gaussian Naive Bayes Classifier precision and recall values were calculated using the default hyperparameters of the classifier and produce a precision of 0.9999 and a recall of 0.9337 for the testing dataset.

The Gaussian Process Classifier was also run using default hyperparameters, but the dataset appeared to be too large for the classifier to handle, even after taking precautions suggested by other model users, such as adding 'copy_X_train = False' into the model before running. To combat this, the data were split into groups of 5000 points and sequentially fitted to the classifier. This model resulted in a precision value of 0.95 and a recall value of 0.998 for the testing dataset.

3.3 Refined Machine Learning Classification

See: SpringBoard-Capstone1/Machine_Learning/Machine_Learning_2-Refined_Models

Upon further inspection of the data, it became clear that the model was performing better than expected, perhaps because there were data that included regions in which no receivers were present. Because of this, it appeared that all models were automatically assigning 0s to rows that had no receiver present. Therefore, data were re-jackknifed using the second method in order to include only data in which receivers were present.

Using these data, the Decision Tree Classifier grid search was run using a range of max depths between 3 and 10, and results found that the best model had a hyperparameter of max depth equal to 10. However, upon plotting the precision scores for this model, it appeared that all models showed precision values that ranged between approximately 0.65 and 0.85, which was much lower than the models using the previous dataset. By looking at the plotted precision values, it appeared as though there was a large difference in precision between maximum depths of 4 and 5, but that the spacing between precision became slightly smaller at higher max depths. In order to keep a model that had a decent precision (0.72) and was relatively easy to understand, a decision tree was built using a max depth of 6 (Figure 13). When predicting values for the training and testing data using this model, precision values were 0.928 and 0.931, respectively. The most important features in this model appeared to be depth gradient, the year 2019, particular zones, temperature, and receiver densities of 2.

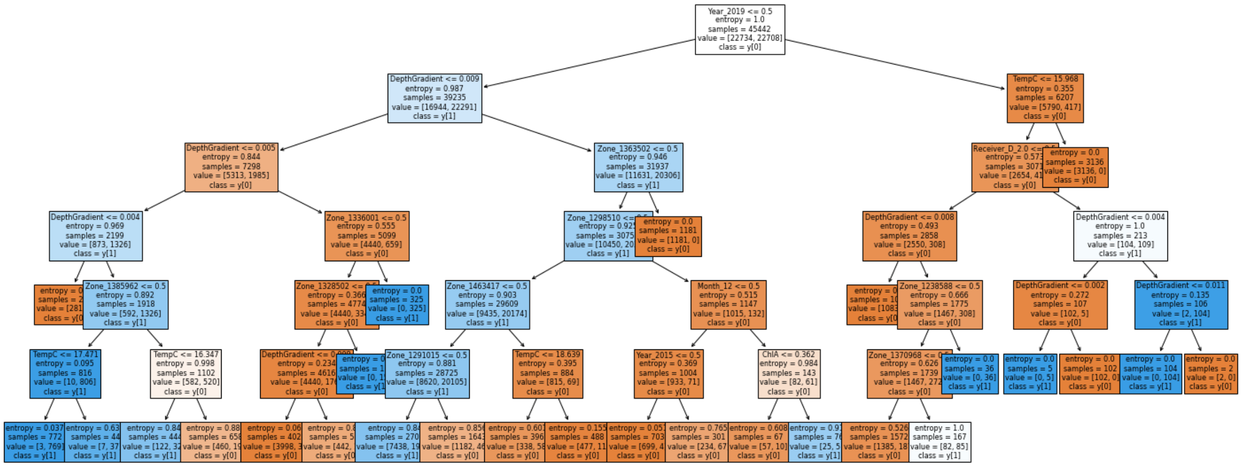
Year_2019 <= 0.5 entropy = 1.0 samples = 45442 value = [22734, 22708] class = y[0]
DepthGradient <= 0.009 entropy = 0.987 samples = 39235 value = [16944, 22291] class = y[1]
TempC <= 15.968 entropy = 0.355 samples = 6207 value = [5790, 417] class = y[0]
DepthGradient <= 0.005 entropy = 0.844 samples = 7298 value = [5313, 1985] class = y[0]
Zone_1363502 <= 0.5 entropy = 0.946 samples = 31937 value = [11631, 20306] class = y[1]
Receiver_D_2.0 <= entropy = 0.573 samples = 3071 value = [2654, 41 class = y[0]
entropy = 0.0 samples = 3136 value = [3136, 0] class = y[0]
DepthGradient <= 0.004 entropy = 0.969 samples = 2199 value = [873, 1326] class = y[1]
Zone_1336001 <= 0.5 entropy = 0.555 samples = 5099 value = [4440, 659] class = y[0]
Zone_1298510 <= entropy = 0.925 samples = 3075 value = [10450, 20 class = y[1]
entropy = 0.0 samples = 1181 value = [1181, 0] class = y[0]
DepthGradient <= 0.008 entropy = 0.493 samples = 2858 value = [2550, 308] class = y[0]
DepthGradient <= 0.004 entropy = 1.0 samples = 213 value = [104, 109] class = y[1]
entropy = 0 samples = 2 value = [281 class = y[0
Zone_1385962 <= 0.5 entropy = 0.892 samples = 1918 value = [592, 1326] class = y[1]
Zone_1328502 <= entropy = 0.366 samples = 4774 value = [4440, 33 class = y[0]
entropy = 0.0 samples = 325 value = [0, 325] class = y[1]
Zone_1463417 <= 0.5 entropy = 0.903 samples = 29609 value = [9435, 20174] class = y[1]
Month_12 <= 0.5 entropy = 0.515 samples = 1147 value = [1015, 132] class = y[0]
entropy = 0 samples = 10 value = [1083 class = y[0
Zone_1238588 <= 0.5 entropy = 0.666 samples = 1775 value = [1467, 308] class = y[0]
DepthGradient <= 0.002 entropy = 0.272 samples = 107 value = [102, 5] class = y[0]
DepthGradient <= 0.011 entropy = 0.135 samples = 106 value = [2, 104] class = y[1]
TempC <= 17.471 entropy = 0.095 samples = 816 value = [10, 806] class = y[1]
TempC <= 16.347 entropy = 0.998 samples = 1102 value = [582, 520] class = y[0]
DepthGradient <= 0 entropy = 0.234 samples = 4616 value = [4440, 17 class = y[0]
entropy = 0 samples = 1 value = [0, 1 class = y[1
Zone_1291015 <= 0.5 entropy = 0.881 samples = 28725 value = [8620, 20105] class = y[1]
TempC <= 18.639 entropy = 0.395 samples = 884 value = [815, 69] class = y[0]
Year_2015 <= 0.5 entropy = 0.369 samples = 1004 value = [933, 71] class = y[0]
ChlA <= 0.362 entropy = 0.984 samples = 143 value = [82, 61] class = y[0]
Zone_1370968 <= entropy = 0.626 samples = 1739 value = [1467, 272 class = y[0]
entropy = 0.0 samples = 36 value = [0, 36] class = y[1]
entropy = 0.0 samples = 5 value = [0, 5] class = y[1]
entropy = 0.0 samples = 102 value = [102, 0] class = y[0]
entropy = 0.0 samples = 104 value = [0, 104] class = y[1]
entropy = 0.0 samples = 2 value = [2, 0] class = y[0]
entropy = 0.037 samples = 772 value = [3, 769] class = y[1]
entropy = 0.63 samples = 44 value = [7, 37 class = y[1]
entropy = 0.84 samples = 444 value = [122, 32 class = y[1]
entropy = 0.88 samples = 658 value = [460, 19 class = y[0]
entropy = 0.06 samples = 402 value = [3998, 3 class = y[0]
entropy = 0.8 samples = 5 value = [442, class = y[0
entropy = 0.84 samples = 270 value = [7438, 19 class = y[1]
entropy = 0.856 samples = 1643 value = [1182, 46 class = y[0]
entropy = 0.601 samples = 396 value = [338, 58 class = y[0]
entropy = 0.155 samples = 488 value = [477, 11 class = y[0]
entropy = 0.051 samples = 703 value = [699, 4 class = y[0]
entropy = 0.765 samples = 301 value = [234, 67 class = y[0]
entropy = 0.608 samples = 67 value = [57, 10] class = y[0]
entropy = 0.91 samples = 76 value = [25, 5 class = y[1]
entropy = 0.526 samples = 1572 value = [1385, 18 class = y[0]
entropy = 1.0 samples = 167 value = [82, 85] class = y[1]

Figure 13. Decision Tree output from a model with a max depth of 6.

A similar approach was conducted for the Gradient Boosting Classifier. A grid search used a variety of possible values for learning rate and max depth and the best model was suggested to have a learning rate of 0.75, a max depth of 8, and 100 estimators. However, upon plotting the precision scores, it was found that all scores ranged between 0.65 and 0.99 and most scores were greater than 0.80 (Figure 14). Therefore, a model that showed about 0.85 precision was chosen to predict the testing dataset. This model used a learning rate of 0.075, a max depth of 3, and 100 estimates. The precision for the training and testing data for this model were 0.904 and 0.903, respectively. The most important features in this model included Depth Gradient, the year 2019, 2015 and 2018, a receiver density of 2, temperature, and particular zones.

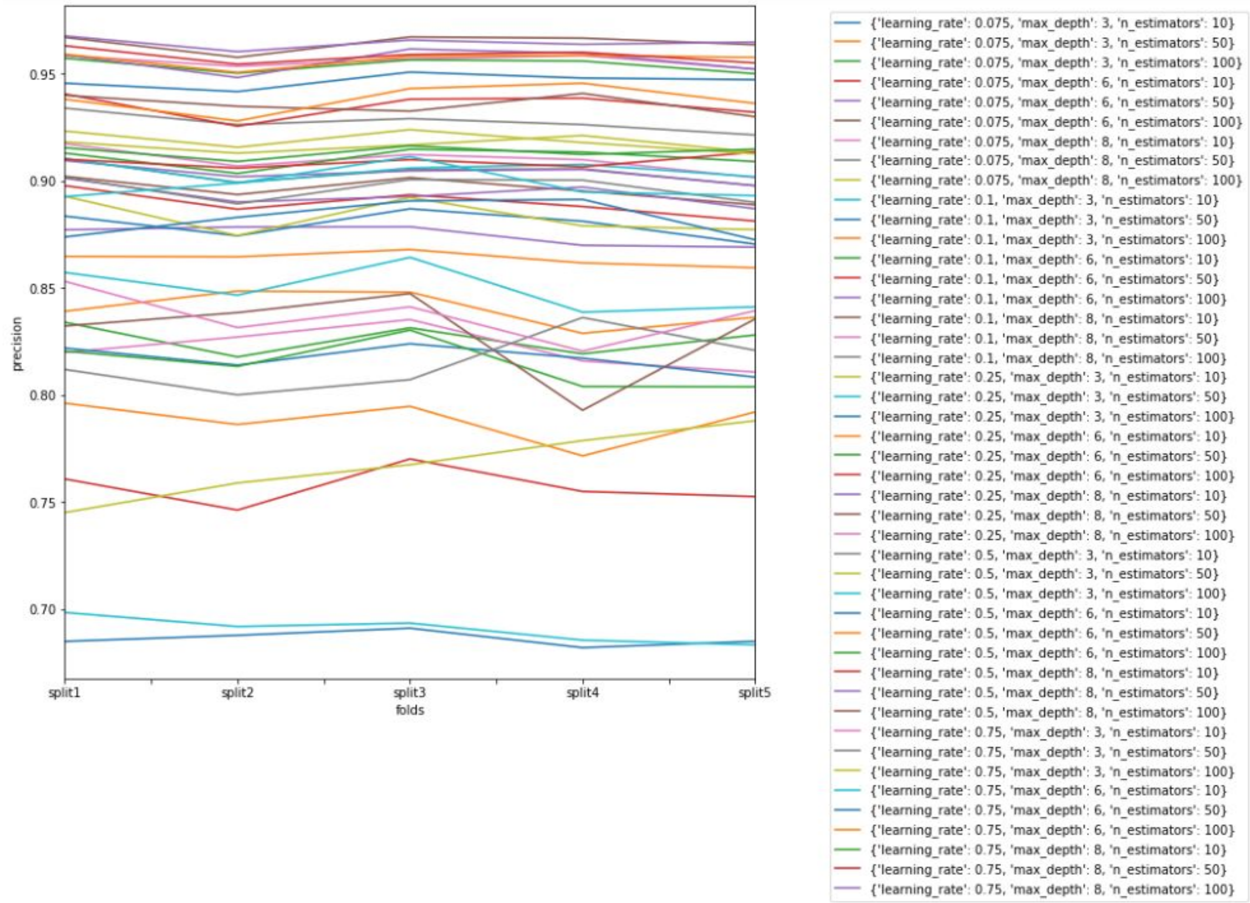

Figure 14. The precision results from a grid search using five folds for a Gradient Boosting Classifier with a variety of options for learning rate, max depth, and number of estimators.

Although these results are beneficial when receivers are in the water, the model should also perform well when receivers are not present. Therefore, another Gradient Boosting Classifier was run without the receiver density predictor. Results from a grid search indicated the best model should have the hyperparameters of a learning rate of 0.75, a max depth of 8, and 100 estimators. However, a similar trend was seen in which most model runs were between 0.65 and 0.99 accuracy. The model with a predicted precision score of about 0.85 was again chosen (learning rate = 0.075, max depth = 3, number of estimators = 100). This model produced precision values of 0.916 when predicting the training data and 0.9156 when predicting the testing data. The most important features here included depth gradient, years 2019, 2015 and 2018, temperature, and particular zones.

A K Nearest Neighbor Classifier was also run on the dataset for which the receiver density factor was removed. A grid search for this model indicated that the best model would have one nearest neighbor with uniform weights. Using this model to predict the testing dataset yielded a precision of 0.999.

Finally, an optimal sample size was calculated by running 100 iterations for each potential sample size and precision estimates for each model were predicted using cross validation with three folds. All values were averaged for each sample size range and results were plotted for precision on the training and testing datasets. From these plots, it appears that the precision values begin to plateau around a precision of 0.70 at approximately 4000 samples.

4. DISCUSSION

4.1 Choosing the Best Model

For the purposes of training a machine learning model to be able to accurately classify the presence of absence of a juvenile White Shark in southern California, it is important to use training data for which one can be confident that there are no false negatives. Originally going into this project, the accuracy scores for these models were expected to be low, but were very high during the initial machine learning stage. Once identifying and fixing the problem (the presence of potential false negatives in regions where receivers were not present), the results were more consistent to what was expected, with precision values in the 70-80% range.

The best model to answer the question at hand is likely a series of different models, some of which may be more useful to some demographics compared to others. For example, if lifeguards are interested in which particular thresholds of values may increase the likelihood of shark presence, a Decision Tree Classifier, such as the one conducted in Section 3.3 might be best. From this example, a key could be developed that would ask lifeguards a series of questions in order to determine shark presence. For example: *Is the year 2019? If no, is the depth gradient in your beach region less than or equal to 0.004 meters? If no, there is likely not a shark present. However, if yes, is your beach located within zone 123? If not, there is likely a shark present.* Although such a model does not produce as exact results as Gradient Boosting or K Nearest Neighbors, it would give lifeguards more confidence in how the results were generated and what parameters appear to be most important to shark presence.

Alternatively, some lifeguards may simply be okay with getting a “sharks are likely present” or “sharks are likely not present” alert for each day, based on yesterday’s environmental information. For such scenarios, a model that tended to perform better, such as the Gradient Boosting Classifier or K Nearest Neighbors Classifier might be more suited. Since these models performed better (particularly K Nearest Neighbors), one would be able to provide relatively instant answers to the questions that lifeguards are most interested in by quickly predicting the outcome from one, new row of data. When these results are taken in combination with the precision estimate, lifeguards may would not have to predict the outcome themselves and might have more time to deal with more pressing matters.

Using a model without receiver density present as a predictor might also be useful for researchers who are interested in finding out where a new acoustic receivers could be placed. The best model for this would likely be the Gradient Boosting Classifier or K Nearest Neighbors Classifier. After being fit to the training data, the researcher can then plug in a few months worth of environmental data for potential zones in which a receiver would be easy to place. Such datasets do not have to be taken by the researcher, but could simply be data grabbed from NOAA CoastWatch from the most recent 6 month time period. By testing on this dataset, the researcher could then see how many times the model predicted a shark to be present in each zone and decide to put a new receiver (or set of receivers) in the zone with the highest number of predicted sharks. Although K Nearest Neighbors would likely provide a more precise assessment of shark presence, a Gradient Boosting Classifier might be more useful for the researcher to explain to their funding sources why they chose the region they did. Perhaps the important factors, like temperature and depth gradient would be more convincing to those who are not comfortable or familiar with machine learning algorithms.

Additionally, these three models, particularly K Nearest Neighbors, are ideal for attempting to fill in gaps in historical data where receivers may not have been present. Researchers using acoustic telemetry often require instances in which they try to recover deployed receivers and cannot find them. K Nearest Neighbors has the potential to be very useful in these instances to backlog the data. Although not 100% precise, such an estimate may help researchers to get a better idea of what the juvenile White Shark population is doing in a particular area, even when the nuances of field work get in the way.

Finally, these techniques could be appropriate for other researchers who are using similar technology but have different degrees of data available and are studying different species. According to the optimal sample size analysis, researchers can implement a similar machine learning method to their data with files as few as 4000 rows long without compromising predictive power.

4.2 Finding the Best Parameters

Most of these models found that depth gradient was the most important predictor, which originally seemed surprising, due to the small range of depth gradient values that were present in the dataset. However, the t-test in the preliminary data analyses also indicated that significantly different depth gradients corresponded to when sharks were present compared to when they were absent. From the Decision Tree Classifier and the t-test results, it appears that sharks may be present more often in regions with slightly steeper seafloor depths. This may be attributed to the feeding mechanisms of juvenile White Sharks, who tend to feed on soft bodied prey items in the sand[2]. Surfaces with small slopes may make hunting easier for juvenile White Sharks by forcing prey to swim uphill, against the natural slope of the seafloor. In addition, sloping surfaces may also be indicative of the habitats that most prey items frequently use, like Round Stingrays (*Urolophus halleri*), that tend to move between deeper and shallower waters based on environmental conditions, such as water temperature and wave action[3].

Other parameters that seemed to be important was the year 2019. This is likely because when this Capstone project started, the year 2019 was not yet over. Therefore, there were likely significantly fewer data points and fewer shark detections in 2019 compared to historical years. Nonetheless, it appears that year itself, regardless of the year being 2019, 2018, 2015, or beyond is a key driver in how well the model performs. This could be incredibly useful for backlogging potential shark presence events in previous years, but presents an interesting challenge in predicting shark presence in real-time data. Each year is accompanied by unique differences in sea surface temperature, storm events, and other environmental parameters that

[2] Estrada, J. A., Rice, A. N., Natanson, L. J., & Skomal, G. B. (2006). Use of isotopic analysis of vertebrae in reconstructing ontogenetic feeding ecology in white sharks. *Ecology*, *87*(4), 829-834.

[3] Lowe, C. G., Moss, G. J., Hoisington, G., Vaudo, J. J., Cartamil, D. P., Marcotte, M. M., & Papastamatiou, Y. P. (2007). Caudal spine shedding periodicity and site fidelity of round stingrays, Urobatis halleri (Cooper), at Seal Beach, California: implications for stingray-related injury management. Bulletin, Southern California Academy of Sciences, 106(1), 16-27.

may influence the precision of these models. However, if all other parameters are collected for testing data, the model is expected to still work relatively well (> 0.70 precision).