Nongjie Feng, Rhys Leahy, and Carter Rogers

Prof. Amir Jafari

Intro to Data Mining

06 December 2021

The Greatest Lakes: Using Data Mining Algorithms and Techniques to Analyze Physical Properties of the Great Lakes

The five Great Lakes of North America are some of the most prominent and essential bodies of water on our planet. Lake Superior, Lake Michigan, Lake Huron, Lake Ontario, and Lake Erie combine to form the largest surface of freshwater in the world with over 94,000 square miles, larger than the entire United Kingdom (Beeton). Huge population centers such as Chicago, Cleveland, Detroit, and Milwaukee are scattered across the coastlines of these lakes. Providing a natural boundary between the United States and Canada, the geographic location of the Great Lakes plays a key role in their use as shipping thoroughfares. The lakes have a notable impact on the climate of the nearby landscape; summers are cooler within the Great Lakes region, and precipitation for the nearby areas is increased, especially in the winter (Beeton). Our goal for this project is to utilize data mining techniques and algorithms to study the physical properties of the Great Lakes. Specifically, we aimed to answer the following questions:

1. Based on surface temperature, can you predict whether each of the Great Lakes reach maximum ice coverage with respect to maximum ice cover average threshold?

2. Based on surface temperature, ice concentration, and physical properties, can you predict which lake a set of characteristics most likely belongs to?

To address the first question, we developed a logistic regression model to classify whether ice coverage exceeds a maximum ice coverage threshold in each of the Great Lakes. Such a model could help predict how percentage ice concentration might change in the Great Lakes as surface temperatures fluctuate with climate change. Additionally, to address the second question, we developed a KNN model to predict which lake a scientific team is most likely examining based on observed surface temperature, ice concentration, and a given lake's physical properties. Theoretically, this model could potentially aid scientists analyzing unlabeled Great Lake data to cluster their observations by lake.

DESCRIPTION OF DATA SET, WRANGLING, AND PREPROCESSING

To answer our key questions, we assembled multiple datasets collected and published by NOAA's CoastWatch program and The Great Lakes Environmental Research Laboratory (GLERL). This division within NOAA "obtains, produces, and delivers environmental data and products for near real-time observation of the Great Lakes to support environmental science, decision making, and supporting research" (Great Lakes CoastWatch Program Overview). NOAA, GLERL, and Michigan Sea Grant, collect, aggregate, and publish "near real-time satellite observations and in-situ data" including daily observations of surface temperature, ice concentration, water level, wind speeds, as well as physical lake characteristics like depth, coastline length, water area, and elevation going back to 1995 (ibid). In this project, we only aggregated data from and trained our models on data going back to 2008 because that's the oldest published ice concentration observations.

Since our questions primarily deal with surface temperature and ice concentration, we

focused on querying, cleaning, and joining these two main datasets, which were spread across 26 distinct .dat files. Furthermore, since physical lake characteristics like depth, volume, and surface area can affect ice formation, we also wrangled and merged our datasets with a statistical summary table outlining physical properties of the Great Lakes (NOAA, Great Lakes Physical Characteristics).

To integrate the ice and surface temperature data, we leveraged the Python Requests library to query the 26 .dat tables (Chandra). Additionally we built-in string operations to clean headers and white space from the raw files. Then, we used the Pandas library to merge the tables and transform the DataFrame into a long and tidy format that could best train both a KNN clustering model as well as a regression model (McKinney). The tidy long format specifically assisted in model development by including lake as a feature in each observation of surface temperature and ice concentration. The final clean and formatted DataFrame contained 31,126 rows and 16 variables. Since our most important features were surface temperature and ice concentration, in model preprocessing we dropped rows where ice concentration appeared as NaN, leaving 8,855 observations. This small percentage emerges because surface temperature is observed all year, but ice only emerges for about a quarter of the year during the winter months. In addition to ice concentration and surface temperature, our final dataset also featured physical properties such as depth, volume, and shoreline length.

To ensure that our data aggregation, cleaning, and preprocessing could be easily replicated across any computer, we hosted a copy of our final dataset on a Google Cloud Platform storage bucket. The surface temperature .dat files were prone to connection and timeout errors during requests, but cloud storage ensures reproducibility and facilitates a streamlined

modeling process across the project. A bucket with the full dataset can be accessed here.

ALGORITHMS USED

The first modeling technique utilized in this project is K-Nearest Neighbors. A common machine learning algorithm for classification, the goal achieved by the K-Nearest Neighbors algorithm is to classify data points based on the category of points surrounding it. The "K" from the name is chosen as the number of data points to consider during classification; for example, a model using k = 5 will observe the 5 nearest data points and use their properties to classify the given data point. Euclidean distance is most commonly used to determine the proximity of data points, although there are other distance metrics available. The benefit to using KNN as a modeling technique is that there are very few parameters to tune and no real model training is required; the downside is that KNN can become very computationally expensive, especially with higher-dimensional data (Joby). KNN is an attractive modeling option for our purposes because we are only using two numerical features, ice concentration and surface temperature, to classify the data points into the observed Great Lake from which they originated.

Linear regression is a very straightforward simple linear approach for predicting a quantitative response Y on the basis of a predictor variable X. We interpret coefficient as the average effect on Y of a one unit increase in variable X, holding all other predictors fixed. Linear Regression fits a linear model with coefficients beta to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

Rather than modeling this response Y directly, logistic regression models the probability that Y belongs to a particular category. In logistic regression, we use the logistic function and we use

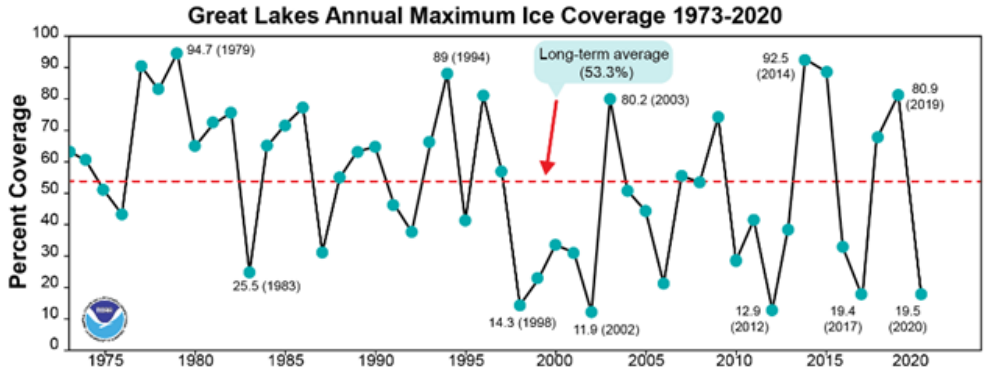
a method called maximum likelihood to fit the model. If p is a probability, then p/(1-p) is the corresponding odds; the logit of the probability is the logarithm of the odds. The logistic function will always produce an S-shaped curve. In a logistic regression model, increasing X by one unit changes the log odds by coefficient beta. Since the relationship between logistic function and X in is not a straight line, the amount that function changes due to a one-unit change in X will depend on the current value of X.

EXPERIMENTAL SETUP

The implementation of the K-Nearest Neighbors modeling process is fairly straightforward. The goal for this model is to classify the correct lake the data point originated from using its ice concentration and surface temperature recordings. KNN excels as a modeling technique for lower-dimensional data, and the only real tuning required for the model is to select a k-value that minimizes error. After normalizing the ice concentration and surface temperature columns, we selected an optimal k-value by looping through k-values between 1-50 to find the k-value for which the error is smallest. We defined error in this case as the percentage of points that were incorrectly classified relative to the observed category. Using this technique, we found the optimal k-value of k = 23 and fit a model using this value.

Rows in the dataset with missing values are excluded. The Lake variable is a categorical variable, and we need to encode the labels with values between 0 and (n-1) classes: Lake Erie:0, Lake Huron:1, Lake Michigan:2, Lake Ontario:3, Lake Superior:4. We use all the variables excluding time variables year, day and id. The ice percentage is our target variable. Split the data into a training set and a testing set. We will train our model on the training set and then use the

test set to evaluate the model. Rescale input variables using standardization or normalization will make more reliable predictions. Set parameter ‘normalize’ true and the regressors X will be normalized before regression by subtracting the mean and dividing by the L2-norm. There are three common evaluation metrics for regression problems. Mean Absolute Error (MAE) is the mean of the absolute value of the errors; Mean Squared Error (MSE) is the mean of the squared errors and is more popular than MAE, because MSE punishes larger errors, which tends to be useful in the real world; Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors and is also popular because RMSE is interpretable in the Y units.



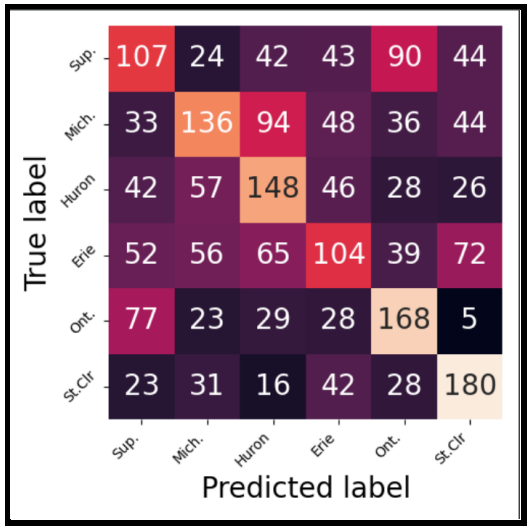
NOAA scientists project the long-term maximum Great Lakes ice cover average for 1973 to 2020 is 53.3%. We will be building a classification model to determine whether they reach maximum ice coverage with respect to maximum ice cover average threshold. In order to map this probability value to a discrete class (true/false), we select a threshold value of 0.533. Mathematically, it can be expressed as: When p larger than 0.533 and class is 1, they reach maximum ice coverage; When p less than 0.533 and class is 0, they do not reach maximum ice coverage. We change the categorical variable Lake to a dummy variable in the logit model. ‘MinMaxScaler’ scales all the data features in the range [0, 1]. Next, we train the logistic model,

we use default ‘C’ equal to 1 and threshold 0.533. Then, we can increase C to 100 and fit a more flexible model. We can also use a more regularized model than the default value of C=1, by setting C=0.01. We use the ‘predict_proba’ method to predict and it gives the probabilities for the target variable (0 and 1). We use a confusion matrix for summarizing the performance of a classification algorithm. Four types of outcomes are possible while evaluating a classification model performance. True Positives (TP): True Positives occur when we predict an observation belongs to a certain class and the observation actually belongs to that class. True Negatives (TN): True Negatives occur when we predict an observation does not belong to a certain class and the observation actually does not belong to that class. False Positives (FP): False Positives occur when we predict an observation belongs to a certain class but the observation actually does not belong to that class. This type of error is called Type I error. False Negatives (FN): False Negatives occur when we predict an observation does not belong to a certain class but the observation actually belongs to that class. This is a very serious error and it is called Type II error. Classification report is another way to evaluate the classification model performance. It displays the precision, recall, f1 and support scores for the model. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall is the ratio of correctly predicted positive observations to all observations in actual class. F1 Score is the weighted average of Precision and Recall. Another tool to measure the classification model performance visually is Receiver Operating Characteristic (ROC) Curve. The ROC Curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various levels. The Geometric Mean or G-Mean is a metric for imbalanced classification that, if optimized, will seek a balance between the sensitivity and the specificity. One approach would be to test the model

with each threshold returned from the call ‘roc_auc_score’ and select the threshold with the largest G-Mean value. Given that we have already calculated the Sensitivity (TPR) and the complement to the Specificity when we calculated the ROC Curve, we can calculate the G-Mean for each threshold directly. ROC-AUC is a single number summary of classifier performance. The higher the value, the better the classifier.

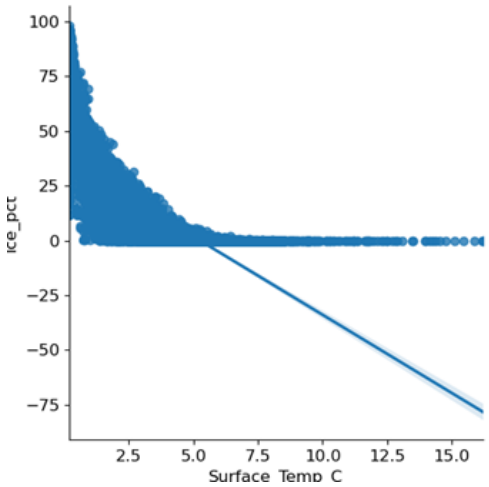
RESULTS

We can interpret the results of the KNN modeling with a heatmap. This visualization depicts where the model excelled in classifying the data points as well as where it struggled. We can see that the model had difficulty distinguishing between Lake Superior and Lake Ontario, for example; of the 350 observations from Lake Superior, 107 were correctly classified while 90 were incorrectly labelled as Lake Ontario. The model performed best at classifying points from Lake St. Clair, which isn’t surprising given its stark differences from the other lakes in the data set. Overall, the model was 39% accurate in classifying the data. This is fairly low, but it is important to consider that a naive model of randomly classifying the data would be about 16.7% accurate. Our model significantly outperforms this theoretical naive model.



From the line plot between surface temperature and ice percentage, we can see that there is a negative relationship between two variables. The lower the temperature, the greater the

percentage of ice. The MSE of the test and train set is around 320 and the R-square is around 0.44. The intercept of the model is 15, which means when the surface temperature is 0, there is 15 percent of ice cover. Holding all other features fixed, a 1 unit decrease in surface temperature. Ice percentage is associated with an increase of 15.



The logistic regression model accuracy score is 0.9345. So, the model does a very good job in predicting whether or not it will reach maximum ice coverage in the Great Lakes. The training-set accuracy score is 0.9365 while the test-set accuracy is 0.9345. These two values are quite comparable. So, there is no question of overfitting. We can see that, C=100 results in higher accuracy. Training-set accuracy score is 0.9771 and test-set accuracy is 0.9752. So, we can conclude that a more complex model should perform better. When C=0.01, Training-set accuracy score is 0.8964 and test-set accuracy is 0.8989. Both the training and test set accuracy decreased. We can see that our model accuracy score is 0.9345 but null accuracy score is 0.9076. So, we can conclude that our Logistic Regression model is doing a very good job in predicting the class labels. True Positives(TP) is 1582; True Negatives(TN) is 73; False Positives(FP) is 10; False

Negatives(FN) is 106. Precision is 0.9937 and it identifies the proportion of correctly predicted positive outcomes. It is more concerned with the positive class than the negative class.; Sensitivity is 0.9372 and it identifies the proportion of correctly predicted actual positives; Specificity is 0.8795. ROC-AUC of our model is 0.7008 and we can conclude that our classifier does a fair job in predicting whether it will reach maximum ice coverage or not.

SUMMARY AND CONCLUSIONS

From the result, a small number of observations predict that there will be maximum ice coverage. Majority of observations predict that there will be no maximum ice coverage. Logistic regression is a process of modeling the probability of a discrete outcome given input variables. Usually, we use logistic regression as a binary outcome, which fits our project very well. Also, logistic regression is considered safe and robust, according to what we learnt from class. The ROC-AUC is not close to 1, the G-Mean method suggests that a set threshold near to 1 may yield better results. The linear regression has a fair R-square score. From the plot in linear regression result, the dataset contains a lot of data points that surface temperature is larger than 5. However, the minimum ice percentage is 0. These points affect the overall performance of the linear model. We can delete these observations, but it will lose a lot of data. We may also try some other models like lasso regression and elastic net.

Our modeling process helps us gain a better understanding of the Great Lakes and their geographic features. Through our use of K-Nearest Neighbors to classify the data based on ice concentration and surface temperature, we observed that the lakes are very similar in terms of their physical properties, making it difficult to determine from which lake a data point was

obtained. Although our model outperformed the baseline naive model, there is room for potential improvement through the use of a different classification technique such as Support Vector Machine or Gradient Boosting.

Works Cited

Beeton, Alfred M.. "Great Lakes". Encyclopedia Britannica, Nov 23rd, 2020, https://www.britannica.com/place/Great-Lakes. Accessed 3 December 2021.

Chandra, R. V., & Varanasi, B. S. Python requests essentials. Packt Publishing Ltd, 2015.

Great Lakes CoastWatch Program Overview. https://coastwatch.glerl.noaa.gov/overview/cw-overview.html. Accessed 5 Dec. 2021.

Great Lakes Physical Characteristics. https://coastwatch.glerl.noaa.gov/statistic/physical.html. Accessed 5 Dec. 2021.

Great Lakes Statistics. https://coastwatch.glerl.noaa.gov/statistic/statistic.html. Accessed 5 Dec. 2021.

Joby, Amal. "What Is K-Nearest Neighbor? A ML Algorithm to Classify Data." *Learn Hub*, 19 July 2021, https://learn.g2.com/k-nearest-neighbor.

McKinney, W., et al. Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56), 2010.

scikit-learn. 2021. *sklearn.linear_model.LinearRegression*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html> [Accessed 6 December 2021].

scikit-learn. 2021. *sklearn.linear_model.LinearRegression*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html> [Accessed 6 December 2021].

En.wikipedia.org. 2021. *Logit - Wikipedia*. [online] Available at: https://en.wikipedia.org/wiki/Logit [Accessed 6 December 2021].

Hastie, T., Tibshirani, R. and Friedman, J., n.d. *The elements of statistical learning*.

James, G., Witten, D., Hastie, T. and Tibshirani, R., n.d. *An introduction to statistical learning*.

Kaggle.com. 2021. *Linear Regression House price prediction*. [online] Available at: <https://www.kaggle.com/faressayah/linear-regression-house-price-prediction> [Accessed 6 December 2021].

Kaggle.com. 2021. *Logistic Regression Classifier Tutorial*. [online] Available at: <https://www.kaggle.com/prashant111/logistic-regression-classifier-tutorial> [Accessed 6 December 2021].

Kaggle.com. 2021. *Heart Disease Prediction using Logistic Regression*. [online] Available at: <https://www.kaggle.com/neisha/heart-disease-prediction-using-logistic-regression> [Accessed 6 December 2021].

pandas?, A., Troy, A. and Zwinck, J., 2021. *Any way to get mappings of a label encoder in Python pandas?*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/42196589/any-way-to-get-mappings-of-a-label-encoder-in-python-pandas> [Accessed 6 December 2021].

Research.noaa.gov. 2021. *NOAA projects 30-percent maximum Great Lakes ice cover for 2021 winter - Welcome to NOAA Research*. [online] Available at: <https://research.noaa.gov/article/ArtMID/587/ArticleID/2706/NOAA-projects-30-percent-average-Great-Lakes-ice-cover-for-2021-winter> [Accessed 6 December 2021].

Brownlee, J., 2021. *A Gentle Introduction to Threshold-Moving for Imbalanced Classification*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/threshold-moving-for-imbalanced-classification/> [Accessed 6 December 2021].