Introduction

The goal of this project is to run machine learning analyses on oceanic features that might influence their geographical classification and assess these analyses' accuracy, without including any location-based data in the learning model. Several machine learning models are deployed in this repository and assessed based on their accuracy of classification. The first part of this paper runs through the process of finding, cleaning, and preparing the datasets for analysis. The second section goes through the k-nearest neighbors and decision tree models of supervised learning, discussing the reasoning behind choosing those methods of analysis and their accuracy scores. Finally, the third section of the paper discusses results generated from three unsupervised learning models: k-means clustering, agglomerative clustering, and DBSCAN clustering. Further research would involve applying this model to different values of depth and time for a continuous evaluation of changes in ocean classification.

Data Selection, Cleaning, and Preprocessing

The dataset used in this project originated from three separate datasets that were merged for analysis. Two of these were taken from NASA's free-access Earthdata website, while the third was taken from an open-access feature layer, created by the United Nations Environment Programme-World Conservation Monitoring Centre (UNEP-WCMC), on ArcGIS Online.

Feature Class Datasets - Description

Both datasets that were used as feature classes in this analysis originated from NASA's Estimating the Circulation and Climate of the Ocean (ECCO) project, where scientists attempt to generate global ocean datasets regarding their properties and changes over time. There are over 100 parameters measured by NASA within this endeavor, organized into more than 25 datasets available to the public for free with registration (ECCO - Home, 2025). Both daily and monthly datasets can be downloaded, as well as those developed in ½° latitude-longitude grids and ECCO's "lat-long-cap-90" native model grid (ECCO - Dataset Gallery, 2025). This Ocean Classification Model uses the latest monthly dataset available, which was recorded in December 2017. Additionally, for ease of analysis and understanding, both chosen datasets were organized in ½° latitude-longitude grids.

The first feature set includes variables regarding the ocean's surface: atmosphere surface specific humidity, atmospheric surface air temperature, zonal (east-west) wind speed, meridional (north-south) wind speed, overall wind speed, and location data (ECCO Consortium, 2021). The data also includes location bounds and time bounds, but because the time and location were already specified above, these variables were rendered redundant. The second feature set contains data regarding the ocean's water content at given depths: density stratification, hydrostatic pressure, and in-situ water density (ECCO Consortium, 2021). As described earlier, this data was also separated by location bounds and time bounds, but had an additional dimension of depth. For the purposes of this analysis, the data from the shallowest depth of -5.0m was chosen. As a result, the density stratification variable, which measures ocean water density at different depths, was rendered redundant as well.

Both datasets were organized in multidimensional arrays, with each variable measured in different dimensions. The datasets were flattened into two-dimensional arrays before merging and cleaning, using Python's Xarray package to read the datasets and Pandas to convert them to dataframes. Most Earth data is organized in multidimensional net-CDF files, which is a file type I was unfamiliar with. Learning how to work with higher dimensions and flatten them into a two-dimensional dataframe, especially when working with such large files that include global levels of data and millions of values, proved to be a challenging yet rewarding experience. The datasets were merged based on latitude and longitude columns to maintain the integrity of location values for each feature.

Target Class Dataset - Description

The dataset describing the target class for this analysis primarily focuses on the classification of oceans based on their locations. The original dataset includes 232 ecoregions and 62 provinces within the oceans, all included within 12 realms (Spalding, 2007). Since this project focuses on classification using machine learning, the 12 realms were chosen as the target class for this analysis. The dataset was read as a shapefile using Python's Geopandas package, with polygons and multipolygon geometries to designate latitude-longitude location points.

To conduct this analysis, Geopandas was used to merge the target class with the feature class datasets by conducting a spatial join of latitude-longitude points in ocean features with the ocean realm polygons. The final, merged dataset was then converted back to a regular Pandas dataframe for data cleaning and preprocessing to occur. Considering that this was my first time coding in Python, the learning curve of using Geopandas was difficult, but I was able to handle the process and generate a map of the classification dataset as well.

Data Cleaning and Preprocessing

Once the final datasets had been merged and converted into a dataframe, irrelevant columns were removed from the analysis. This included the time and location columns, as the time and date were the same for each row (only one unique value was found), and the goal of this project is to predict the ocean's realm, as a class, without knowing its location. Additionally, the density stratification variable was removed at this step for reasons explored previously. Finally, due to the difficulty of handling the large size of the dataset, rows with missing values were removed. The data was ready for preprocessing.

Python's Scikit-learn library was imported at the beginning to split the data into training and test sets, both of which were stratified by the target class in order to ensure an even distribution of classes between both. The target class was then encoded in both sets using scikit-learn's One-Hot Encoder library. Finally, the data from the training set was visualized for insights into variable distribution and skewness, along with the correlation between features.

Histograms of both target and feature classes were created to analyze distributions. A histogram of the target class, ocean realms, shows that the highest counts of values are concentrated in Atlantic Warm Water and Southern Cold Water. Meanwhile, the most skewed feature class sets are atmospheric surface temperature, atmospheric surface pressure, and hydrostatic pressure anomaly. Because the data for atmospheric surface temperature and pressure variables are left-skewed, both variables were squared. However, this seemed to make little difference in the distribution overall. The same was true for the hydrostatic pressure anomaly variable as well, except that the data is right-skewed – a logarithmic transformation was applied to the training set, but it seemed to make little difference in overall distribution. Since the transformations did not significantly change variable distribution, they were not conducted on the datasets later used in machine learning.

The final visualization includes a matrix of plots displaying the correlations between the distributions of all feature variables in the training data. Most variables were related to each other in some way, with the visualizations depicting a variety of shapes. Occasionally, there were a few variables that did not seem related to each other, with most of those graphs including one of the three available measurements of wind speed. Nevertheless, the data was prepared for machine learning at this point.

Supervised Learning Models

K-Nearest Neighbors and Decision Trees were chosen as the two models of supervised learning applied to this dataset, primarily because neither model makes assumptions about the distribution of the data. As explained in the previous section, the skewness in distributions did not change significantly despite transformation attempts. Both models were first fit to the training set of the data, then used to predict target classes in the test set.

K-Nearest Neighbors Model

The K-Nearest Neighbors supervised learning model makes predictions on datasets by choosing a random data point within the feature class's dimensional space, and selecting a certain number of "nearest neighbors" around that point. The model then classifies that group of points as belonging to a target feature by assessing which target class is most represented by that group of points. There are two parameters the user can manipulate in order to find the best k-nearest neighbors model for their dataset: the number of neighbors selected and the type of distance calculation the model will conduct. Lower numbers of nearest neighbors tend to fit models better, but run the risk of overfitting and predicting the classes of new data points poorly, as the model becomes specific to the training data. Increasing the number of neighbors generally worsens performance, but can reduce the risk of overfitting.

A grid search conducted on the ocean classification dataset revealed the best parameters for k-nearest neighbors to be 1 or 3, in sequence, nearest neighbors, calculated with the model's default measurement system of Euclidean distances. Euclidean distance is simply the direct linear distance between a randomly selected point in the space and a point within the dataset. The model was fit to the training dataset and then used to predict the ocean realms within the testing dataset. Fitting the model to 1 neighbor results in a roughly 87.8% accuracy score in class prediction in the test set, implying that the model correctly guesses around 88% of the classes present in the test set. Subsequently, 3 neighbors result in an accuracy score of around 66.2%.

Decision Tree Classification Model

A decision tree classification model in machine learning operates by asking a series of Boolean questions about a data point for classification. It runs through several iterations of questions for each point in the training dataset, and uses the information learned through this process to make predictions on the test set. Much like the k-nearest neighbors model, the decision tree classifier also has several parameters that can be adjusted to improve performance. The key metric that became relevant in this analysis, after a grid search was conducted, was the maximum depth of the tree. The maximum depth parameter determines how many Boolean questions are asked by the model of the data point; a higher value of maximum depth runs the risk of overfitting, as the model may become overly specific to the questions asked of the training set and struggle to classify new values. The grid search through parameters revealed that the best maximum depth value for this dataset, in order to maximize performance but minimize overfitting, is 9. The default value for the maximum depth parameter, when unspecified, is none. With no maximum depth, the decision tree model predicts around 94.5% of the classes present in the test set accurately. However, with the maximum depth of 9 applied, the accuracy score of the decision tree's predictions falls to roughly 83%.

Comparing the Models and the Effects of Principal Component Analysis

Both models perform relatively well, as assessed by their accuracy scores, on predicting the ocean realm class within the test set. The best model with k-nearest neighbors shows a higher overall accuracy score than the best-performing decision tree. The precision, recall, and F1 scores determined by the model also provide new insights into model performance. The recall and F1 scores for the k-nearest neighbors models are consistently higher than those of the decision tree. However, there are occasional inconsistencies in the precision scores for each model; the score is higher for some classes in the decision tree model than in the k-nearest neighbors model, meaning in those cases, the decision tree had a higher ratio of correctly-predicted positive values to the total number of predicted positive values in the dataset. Nevertheless, because the k-nearest neighbors had higher scores in almost every class and in three out of the four scores calculated, the model was reevaluated with Principal Component Analysis applied.

Principal Component Analysis (PCA), in the context of machine learning, is designed to reduce the number of features selected for analysis to those that impact the variations in the dataset the most. In this model, PCA was conducted to reduce the number of features such that 95% of the variance in the dataset remained explained. The result of this PCA was a reduction of the initial 9 features to just 5. This PCA was then applied to both the testing and training datasets, and the k-nearest neighbors model was run with 1 neighbor specified as a parameter. The performance of the model greatly increased after PCA, with a new accuracy score of 93.2% on the test set, and precision, recall, and F1 scores increased across the board. It seems that the additional features that were ignored after PCA were confusing the model with additional noise and variance. When reduced to the 5 most significant features, the model improved.

Unsupervised Learning Models

Three different unsupervised learning models were adapted to the training set of this data: k-means clustering, agglomerative/hierarchical clustering, and DBSCAN clustering. All models were run before and after clustering to compare the results for the models.

K-Means Clustering

K-means clustering operates by separating data into a given number of clusters based on how close the data points are to each other. In other words, data points with the least amount of space between them are clustered into a group, for which the number of groups of classification is pre-provided to the model. The ideal number of clusters in a dataset can be visualized using an elbow plot, where the inertia, or the compactness of points within clusters, is visualized against several numbers of clusters to assess the point at which the reduction of inertia begins to flatten. Less inertia is reduced with each additional cluster, meaning there is little benefit in adding clusters for analysis.

The k-means clustering model was tested on data before and after PCA was applied. The method of PCA applied was the same as that described in the Supervised Learning section of this paper. Prior to PCA, the ideal number of clusters, as visualized in the elbow plot, was around 8. However, after PCA, this number reduced to 6 clusters. A potential reason for this is that the more "noisy" features were removed, so there was less of a need for additional clusters. The scoring methods for models will be discussed later in the paper.

Agglomerative/Hierarchical Clustering

Agglomerative clustering runs distance calculations for each data point one by one, creating new clusters with new distances between them. This process continues to run until there are no clusters remaining; in other words, until all the clusters have joined into one. The agglomerative cluster model was not running on the initial training dataset due to its size. The amount of memory in Google Colab's servers was not able to handle the iterative process of clustering. To remedy this, a new, smaller training set was created by re-splitting the original, cleaned data into a new training and test set, with less data included in the training set (though the stratification of the data was the same, based on the target data, as described earlier). The model was then successful in running through the smaller set of ocean classification training data, both before and after PCA.

DBSCAN Clustering

The DBSCAN clustering model runs by assessing the dataset and grouping points together based on their distances, much like the previous models. However, it generally ignores more distant points and designates them as noise, which is often useful when dealing with real-world data that includes many noisy points. The DBSCAN model ran smoothly on the original training dataset, both before and after PCA.

Unsupervised Learning Results and Model Comparisons

The accuracy of the results from unsupervised learning was questionable. Two metrics of accuracy were applied to all three models for comparison purposes: Adjusted Rand Index (ARI) and silhouette scores. ARI assesses the accuracy of clustering predictions in unsupervised learning by comparing the similarities between clusters in the feature class, contrasted with those of the classes defined by the target class. It also attempts to correct for chance agreement between the clusters. The silhouette score, on the other hand, determines how tightly the points in the clusters are distributed and compares those distances to the distances between the clusters themselves. The tighter the cluster and the farther away it is from the next cluster, the higher the score. Both of these metrics range from values of -1 to +1, with 0 indicating complete randomness.

Because agglomerative clustering could not be run on the whole training dataset, the smaller version of the training set was applied to all scoring metrics for equal analysis. The analysis would have been unbalanced if two of the models used the larger training set, while the third used the smaller one. Both scores were first calculated on the smaller training set itself, then on the smaller training set with PCA applied.

DBSCAN was the worst-performing clustering model for this dataset, both before and after PCA. The ARI of the model, before PCA was applied, was nearly 0, and the silhouette score was -0.42. Agglomerative and k-means clustering performed similarly to each other before PCA – both their ARIs were 0.09, though k-means had a silhouette score of 0.30, and that of agglomerative clustering was 0.26. Given that a score of 0 indicates random chance, none of these models were particularly effective at clustering the data in an accurate way into their classes.

Scores after PCA did not show much improvement. DBSCAN clustering resulted in an ARI of 0.02, with a silhouette score of -0.25. The agglomerative and k-means clustering methods stayed similar as well. Agglomerative clustering has an ARI of 0.11 and a silhouette score of 0.28, while k-means had an ARI of 0.08 and a silhouette score of 0.32. Evidently, clustering methods were not quite as effective as expected. The ideal would be to have the models cluster the data into the 11 classes found in the initial dataset. However, this was not the case before or after PCA. The models likely need to be adjusted or improved with existing parameters for better fits.

Conclusions and Opportunities for Further Research

The purpose of this study was to attempt to classify oceans into their respective realms without latitude and longitude information, simply based on metrics like atmospheric temperature, humidity, pressure, wind directions and speeds, hydrostatic pressure, and water density. After cleaning, preprocessing, and visualizing each of the features and attempting to transform data that was not normally distributed, the two best supervised learning methods for this dataset were assessed to be k-nearest neighbors and decision trees. While both models' optimal parameters resulted in fairly accurate results, the k-nearest neighbors analysis showed higher accuracy, especially after principal component analysis was applied to the data. On the other hand, clustering models did not provide many insights into ocean classification by realm, as the best adjusted rand index, before and after PCA, was just 0.11 with the post-PCA agglomerative clustering model. The best silhouette score was of the k-means model after PCA, at 0.32. In future attempts at conducting this analysis, I would like to adjust more parameters in each of the models, especially unsupervised learning models. I would also like to explore more methods of transforming and normalizing the data, so that different models can be applied to the set. Additionally, because this model only includes data from December 2017, it may not capture year-round weather patterns that impact monthly values of the feature classes. Combining monthly data into year-long averages would be one way to check for better-fitting classification models. Finally, there is a discrepancy between the date of creation of the target class dataset, which was published in 2007, and the feature classes. Opportunities to expand this research would therefore include a time series analysis, where comparisons can be made between classifications in 2007 and those in 2017, or even in the present.

References

ECCO Consortium, Fukumori, I., Wang, O., Fenty, I., Forget, G., Heimbach, P., & Ponte, R. M.. 2021. ECCO Ocean Density, Stratification, and Hydrostatic Pressure - Daily Mean 0.5 Degree (Version 4 Release 4). Ver. V4r4. PO.DAAC, CA, USA. Dataset accessed 2025-12-22 at https://doi.org/10.5067/ECG5D-ODE44

ECCO Consortium, Fukumori, I., Wang, O., Fenty, I., Forget, G., Heimbach, P., & Ponte, R. M.. 2021. ECCO Atmosphere Surface Temperature, Humidity, Wind, and Pressure - Monthly Mean 0.5 Degree (Version 4 Release 4). Ver. V4r4. PO.DAAC, CA, USA. Dataset accessed [YYYY-MM-DD] at https://doi.org/10.5067/ECG5M-ATM44

ECCO - Dataset Gallery. (2025, November 13). NASA. https://ecco-group.org/datasets.htm

ECCO - Home. (2025, November 13). NASA. https://ecco-group.org/

Spalding, M. D., Fox, H. E., Allen, G. R., Davidson, N., Ferdaña, Z. A., Finlayson, M., Halpern, B. S., Jorge, M. A., Lombana, A., Lourie, S. A., Martin, K. D., McManus, E., Molnar, J., Recchia, C. A., & Robertson, J. (2007). Marine ecoregions of the world: A bioregionalization of coastal and shelf areas. *BioScience*, *57*(7), 573–583. https://doi.org/10.1641/B570707