Ensemble Bathymetry Classification

Introduction

We are tasked with building a classifier on bathymetry data for the labels ‘good’, ‘bad’, or ‘I don’t know’, each with their own associated gain and penalty costs. The bathymetry data is depth-mapping data collected by cruises from different regions or institutions. For training data, there are four institutions, the amount of data of which after filtering is listed in the table below.

Institution	JAMSTEC + JAMSTEC2	NGDC	SIO	US_multi
Data	38M + 6M	35M	35M	32M

Table 1: Distribution of data across different institutions

The data is stored in tab-separated format, each data point consists of 35 columns of features and 1 column composed of the true labels. The proportion of the data by label for each institution is shown in Figure 1.

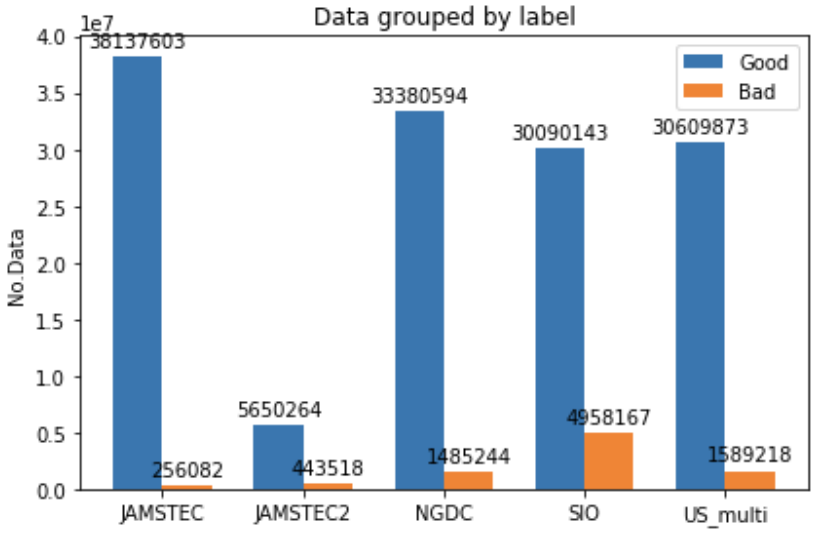

Figure 1: True data label distribution grouped by institution

Overall, we aim to build a stable and accurate classifier that maximizes the classification accuracy on the bathymetry data when tested on a test set composed of only one institution’s data. Accuracy requires that the predicted labels are the same as the true labels while stability means that small changes to inputs should result in small changes to output labels. To enforce stability, instead of having a simple binary classifier, we define a third label 'idk' which signifies

that the network is not confident in its prediction. Intuitively, one can see that having a third label enhances stability by instituting a buffer zone between the 'good' and 'bad;' thus preventing wide swings in the label with small perturbations of the input. For the scope of this project, we focus on the following three approaches: Gradient Boosting Trees on all institution data available, Neural Network on data sampled from all institutions, and an ensemble classifier that takes in the output of the two aforementioned classifiers and produces a final prediction. To quantify our confidence in each classifier we calculate the Accuracy, as (i) the area under the Precision/Recall curve; and (ii) the area under the Receiver Operating Characteristic (ROC) curve.

In our initial explanation of the data, we hypothesized that longitude and latitude are important for accurate depth mapping. To explore this hypothesis, we sampled data from each institution and plotted longitude and latitude to see if each institution's data had any overlap with another. Figure 2 shows inter-institutional overlap in location of the data, thus making it difficult for us to draw a clear pattern on how institutions vary on a geographical basis.

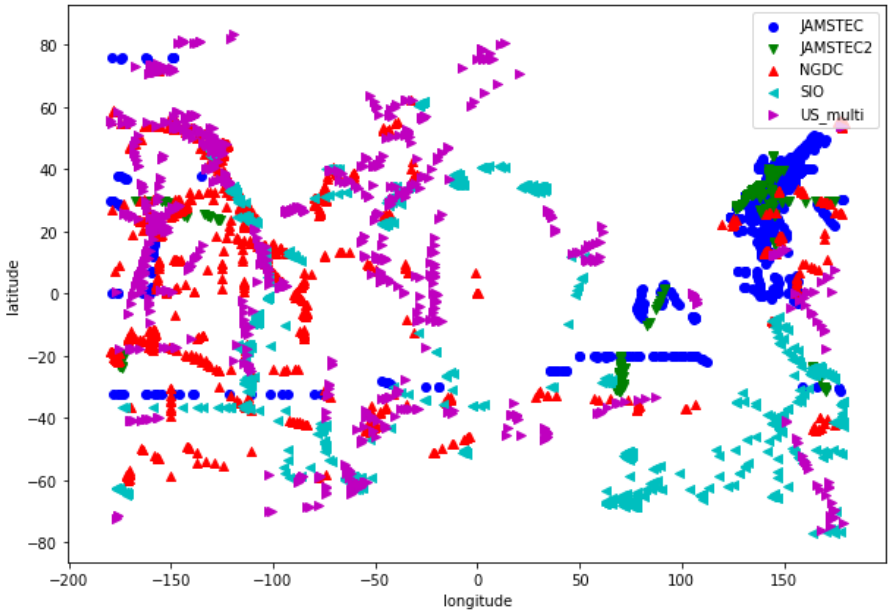

Figure 2: Geographical distribution of longitude and latitude of sampled data by institution

Approach #1: Boosting using LightGBM

For the first approach, we choose from the boosting algorithm family which tends to have good performance on classification tasks. Considering the huge data amount, we choose LightGBM, a high-performance gradient boosting framework based on the decision tree algorithm. The hyper-parameters were set as follows: number of leaves: 31; min data in one leaf: 1; max number of bins for feature bucketing: 255. We trained 1000 gradient boosted trees with a learning rate of eta=0.01.

For data utility, we combined data from all 4 institutions and split into training and validation set by 8:2. It takes 1~2hr to train the gradient boosting model on a m5ad.12xlarge AWS instance. The overall validation accuracy for this approach is 94.6%.

Approach #2: Neural Network

As a second approach to building a classifier, we built a neural network (NN) composed of four hidden layers with relu activation, using binary cross entropy loss and an Adam optimizer. The number of nodes in each layer from beginning to end was 30 (input layer), 128, 64, 32, 16, and 1 (output layer). The classifier was trained on sampled data from each region for five epochs, batch size of 256, with a learning rate of 0.001. We trained the model on the datasets from the various regions in the following order: JAMSTEC2, JAMSTEC, NGDC, SIO, and US_multi.

We then examined the ROC curve for the trained neural network on samples of data from each region (Figure 3). We saw that the NN performed the best on the JAMSTEC dataset and second best on the SIO dataset. The ROC curve for the NGDC dataset is still above the 45 degree diagonal line, meaning our classifier still performs better than random. However, the ROC curve for the US_multi data lies along the 45 degree diagonal line except for at the tail ends, meaning our classifier performs very poorly on the US_multi region dataset.

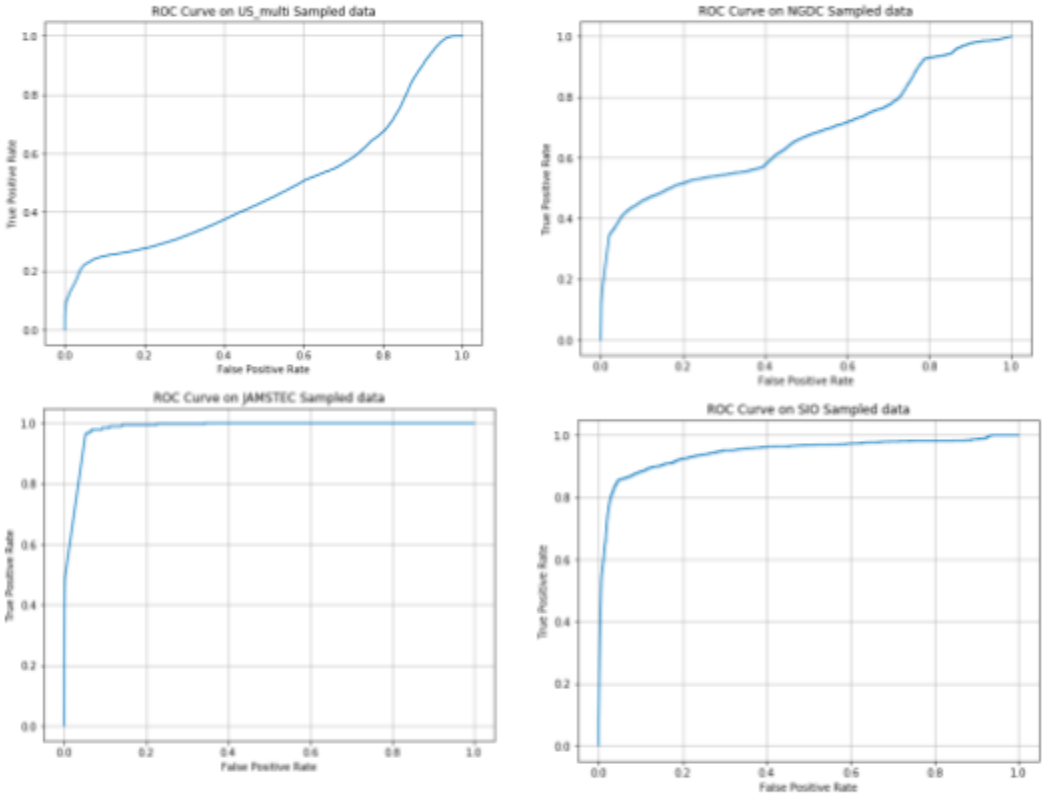

Figure 3: ROC curves of trained NN of sampled data by region.The closer the curve is to the left-hand border, and top-most border of the ROC space, the more accurate the

classifier is. The closer the curve is to the 45 degree diagonal of the ROC space, the less accurate the classifier is.

For comparison, we calculate the accuracy of the XGBoost model and the NN on the same validation dataset, which is randomly picked from the entire dataset. The accuracy of our neural network model is 95.3% which is higher than that of boosting trees as 94.6%. Following, we further tried to ensemble these two models to obtain a more robust model.

Approach #3: Ensemble Classifier: Boosting Trees + Neural Network

In order to combine the boosting trees and the neural network model, we use logistic regression to train a linear combination of the predictions. First, we randomly select 20% data from the entire dataset (regardless of region). Then, we combine the prediction probabilities of the two models into an $L \times 2$ array as the input, where L is the size of the validation set. We fit this with the true labels of the validation set and get the ensemble model. Finally, we use the ensemble model to generate final predictions on the test dataset.

The validation accuracy of the ensemble model is 97.4%, classifying 23,781,621 points as "good" and 1,016,900 as "bad". Analyzing the precision/recall curve, we found that the area under curve (AUC) for the ensemble classifier is 0.99. Precision measures what proportion of positive identifications were actually correct i.e., measures the number of true positives (TP) divided by the sum of TP and false positives (FP). Recall measures the proportion of TP identified correctly i.e., Recall = TP/ (TP + FN). The precision recall curve shows the tradeoff between precision and recall with an AUC closer to 1 showing a better classifier. Since our classifier's AUC is 0.99, our classifier is robust.

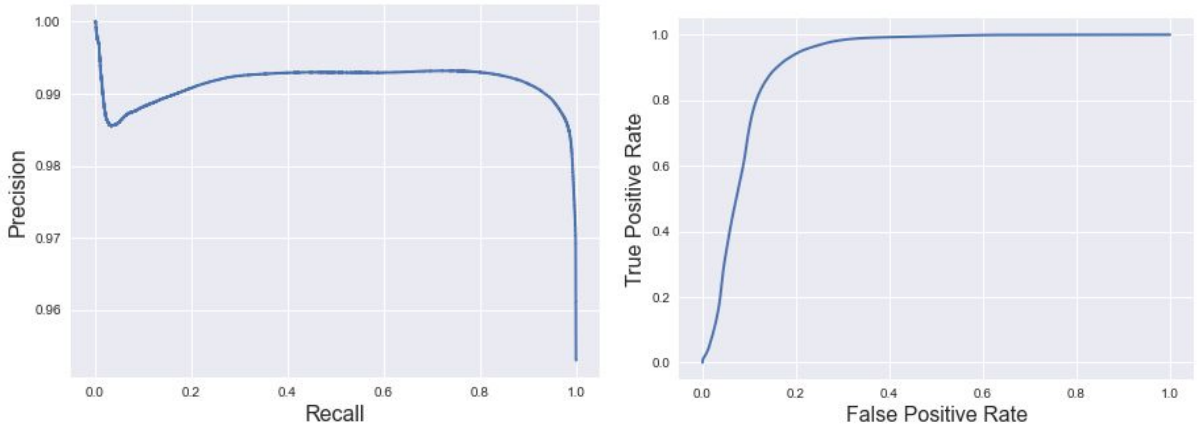

Figure 4: The performance of the ensemble classifier on the test set. *Left: Precision-Recall curve, where Area Under the Curve is 0.99. Right: ROC curve of True Positive vs False Positive Rate, where Area Under the Curve is 0.91.*

The Receiver Operating Characteristic (ROC) curve denotes the number of TP vs FP as the threshold for the classifier is varied. The area under the ROC curve is 0.91 which is another indicator that the classifier is robust.

To complete our classifier, we had to determine the thresholds for each label. To find the lower and upper thresholds, we plotted the distribution of correctly and incorrectly classified scores (see Figure 4). Here, we see the distribution of correctly and incorrectly predicted scores for the ensemble classifier with a threshold of 0.5. The black dashed lines indicated the upper and lower thresholds for the optimal region for saying the classifier is not confident in its predictions. As expected as the score approaches 1 the number of correct to incorrect increases dramatically. The distribution on the negative is rather more interesting. The spike in the number of incorrectly classified values around 0.2 is caused entirely by one region (JAMSTEC 2). It is unclear why this is the case.

Next, we determine the final thresholds for each label by maximizing a reward function defined by the following weights: +1 for correct predictions, -3 for incorrect predictions, and 0 for predictions that opted for neither a good nor bad label. We select the thresholds that maximize the reward function over the 20% randomly selected data across all institutions. We show the distribution of correct and incorrectly classified scores in Figure 5. The black vertical dashed lines indicate the lower and upper thresholds that divide the labels ‘bad’, ‘I don’t know’ and ‘good’, respectively.

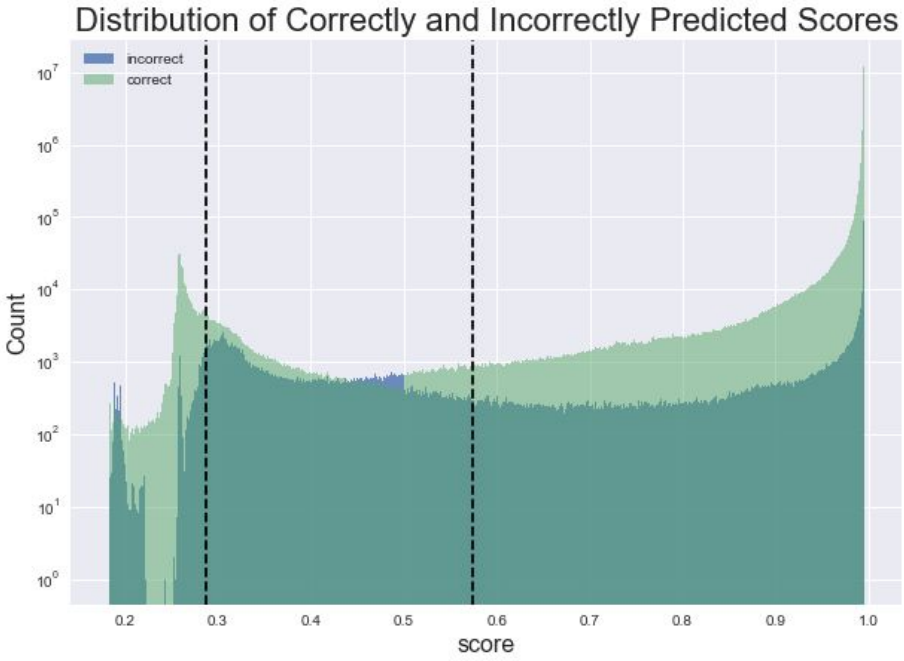

Figure 5: Distribution of correct and incorrectly classified predicted scores. The vertical black dashed lines indicate the lower (left) and upper (right) thresholds that determine the ‘bad’, ‘I don’t know’ and the ‘good’ labels.

We then produce predictions for our 20% randomly sampled across all institutions dataset and examine the distribution of these labels (Figure 6 (left)). Notice that the number of datapoints labeled ‘bad’ is much lower than ‘I don’t know’ or ‘good’. This means that the classifier labels good scores with high confidence but is not as certain when it comes to classifying low scores. This isn’t surprising since the proportion of ‘bad’ labelled data points in our training set is much smaller relative to the proportion of ‘good’ data points (Figure 1).

Finally, to analyze the performance of our ensemble model, we plot a histogram of the predicted labels on the test set (where the true labels are unknown) (Figure 6 (right)). The test set is drawn from a single, unknown institution, whereas our validation dataset was drawn from data across all institutions. Here, it is interesting to note that we find that the distribution of the labels in our test set is different than that of our validation set, predicting more ‘I don’t know’ labels than in our validation set.

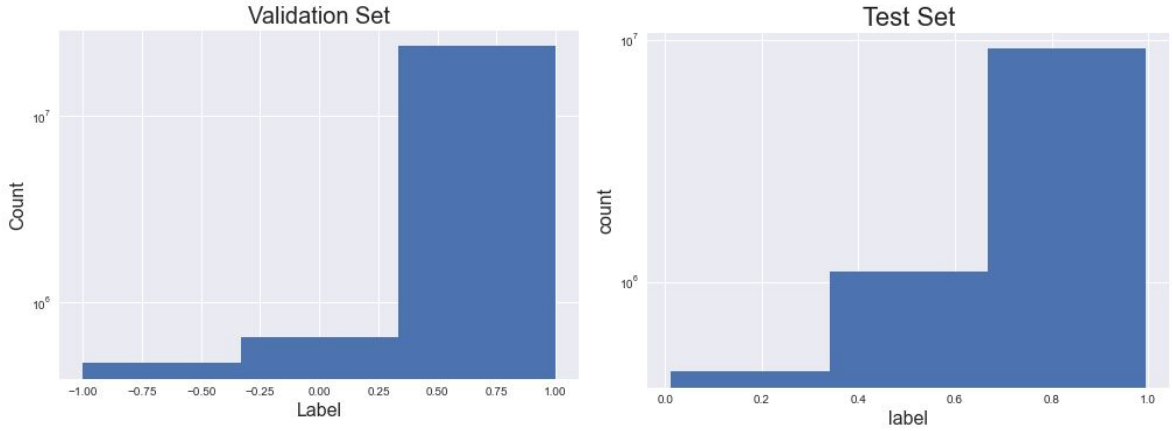

Figure 6: Distribution of data points by predicted label on the validation set (left) and on the test set (right).

Discussion of Thresholds:

Thresholds are chosen based on maximization of the reward function, as has been previously discussed. We state that we have chosen these thresholds globally in the sense that the reward function is calculated over the whole dataset. In order to make sure that we have a consistent classifier i.e., one that won't do too badly in any one region, we check the scores by region for the optimal threshold values for that region vs the optimal over the whole dataset. The average reward among all data points given by region is [0.994, 0.82 , 0.907, 0.854, 0.896] for JAMSTEC, JAMSTEC 2, NGDC, SIO, and US_multi respectively (data presented in the same order throughout). The reward values closer to +1, the reward for a correct prediction, the better. Thus we see that JAMSTEC 2 is the worst performer followed by SIO.

To see if we would have a higher overall gain by choosing the thresholds based on a particular dataset, we calculate the thresholds for each of the datasets individually: [0.36, 0.63], [0.17, 0.62], [0.28, 0.55], [0.38, 0.65], [0.36, 0.58], compared to [0.29 0.58] thresholds optimized across data from all institutions. Here, we see the tradeoff between stability and accuracy, since

increasing the width of the thresholds enhances the stability of the classifier but takes a toll on accuracy. Thus, to measure the extent of the tradeoff, we examine the cost of imposing each of the locally optimal solutions globally. We found that the net cost of setting each locally optimal region threshold over the whole dataset is: [-0.007, -0.015, -0.001, -0.008, -0.007]. These are not large costs but given that the classifier was 97% accurate it is still significant. The benefit of using local optimal region thresholds to the region it was optimal on was [0.0004, 0.0044, 0.0012, 0.0164, 0.0072], mostly less than the cost - even when you don't account for the fact that these are average scores among all data points and regional datasets have many fewer points individually than the total. Thus, in the tradeoff between accuracy and stability, we choose a slightly less stable but significantly more accurate classifier by keeping the globally optimal thresholds.

Conclusion:

In conclusion, the Ensemble Classifier does better than each of component classifiers: Boosting Trees and Neural Networks. It performs better at a 97% level of accuracy compared to the 94.6% accuracy of Boosting and 95.4% accuracy of the neural network classifier. While there are differences in bathymetry across the cruises based on the regions that the cruises visited, all cruise data was classified similarly, except JAMSTEC2 which had a lot of discrepancies. These discrepancies might be because it has the least amount of data points (see Figure 1) and the data points are collected over a more varied geographical area (see Figure 2). This might also be attributed to the decision to use global thresholds rather than JAMSTEC2 specific trained thresholds.