Random Forest Algorithm for predicting Ripple Zones

# Objective

This project aims to use ripple index and granular data (mean size, sorting, skewness and kurtosis) to train a random forest model for predicting the tidal zone of said ripple.

# Method

1. All the data that we have was first loaded onto an excel sheet and converted to CSV format for use in MATLAB. The data also contained information related to the zones of each ripple which was to be used for training our model.
2. The data that we have for the different features mentioned above vary over a set range. Thus, we first use a MATLAB function to categorize the data of each feature into ranges (binning; Fig. 1).
3. This data was visualized in the form of a bar graph (Fig. 2).
4. After this we created a template tree ‘t’ to be used by our Random Forest Model in which we specified the parameters (features) to be used for predicting the output (zones, Fig. 3).
   1. Predictor variables was set as ‘all’ features
   2. Node selection method was selected as ‘interaction-curvature’
   3. Surrogate was set to ‘on’ for a more flexible model which can deal with missing data
5. Fitrensemble method was used to train our Random Forest Model (Fig. 4)
   1. The output/predicted variable was set to ‘Zone’
   2. The method for predicting the output was set as ‘Bag’
   3. The number of learning cycle iterations was set to 100
   4. The trees generated in the model were set to use the template created earlier ‘t’
6. We then permuted OOB (out of bag) iterations of our generated Random Forest Model to understand the effect that each predictor variable (feature) had on our model. A bar graph was generated for the same for better clarity (visualization) of the results (Fig. 5, 6).
7. The next step was to compare the accuracy of the above results using MSE (mean squared error) to understand which values were actually affecting the model’s prediction. A line graph for the same was created for better clarity (visualization; Fig. 7, 8).
8. The final step for understanding the effect of each predictor on the model was creating a gradient showing the interaction of the predictor variables (features) taken in a pair-wise manner (Fig. 9, 10).
9. The next part of our project involved partitioning the data into a training dataset (70%) and test dataset (30%). After this was done, we used the initial Random Forest Model specified above to train our Algorithm using the fitensemble method and the training dataset and predicting the zones of the test dataset (Fig. 11). A combined table of test dataset and predicted zones was then created (Fig. 12).

# Observations

1. From our Random Forest Model we observed that mean grain size and kurtosis had the most effect on determining which tidal zone a ripple mark belongs to.
2. Ripple Index had the least impact on determining the same.
3. The model accuracy for analyzing predictor importance came out to be 88.97% and that for predicting tidal zones of ripples came out to be 91.67%.

# Results

The Random Forest Model generated using above mentioned data was successful in accurately predicting which zone a specific dataset belongs to. Mean size and kurtosis had the most effect on determining the same.

# Future Work

In order to better visualize the obtained results (demarcation of zones), we can use methods like PCA (Principal Component Analysis) to generate weighted variables and set them as the X and Y axes of a mesh grid.

# Figures

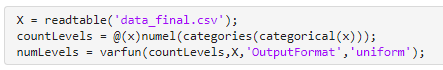


Fig. 1: MATLAB code for counting number of categories (levels/ranges) for each feature

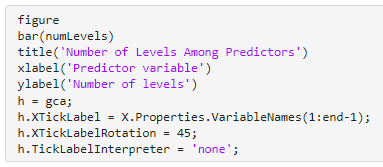


Fig. 2: MATLAB code for generating a bar graph visualizing number if categories for each feature

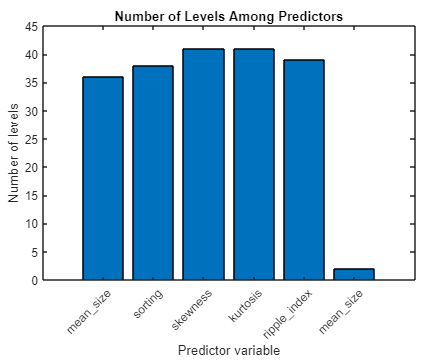


Fig. 3: Bar graph showing number of categories for each feature

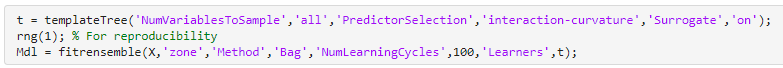


Fig. 4: MATLAB code for template Random Forest Tree and Model

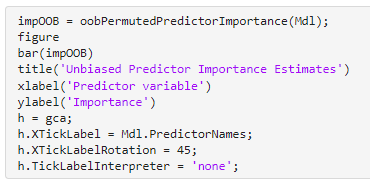


Fig. 5: MATLAB code for generating a bar graph visualizing the OOB importance of each feature

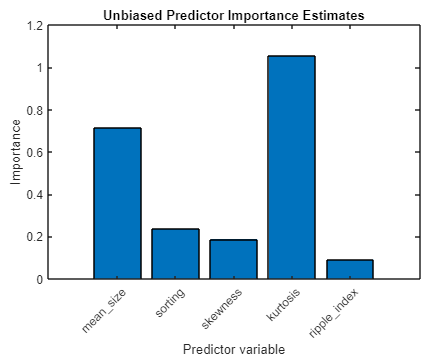


Fig. 6: Bar graph showing the OOB importance of each feature

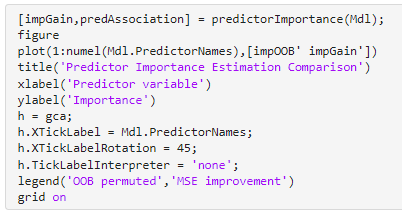


Fig. 7: MATLAB code for comparing OOB importance against MSE

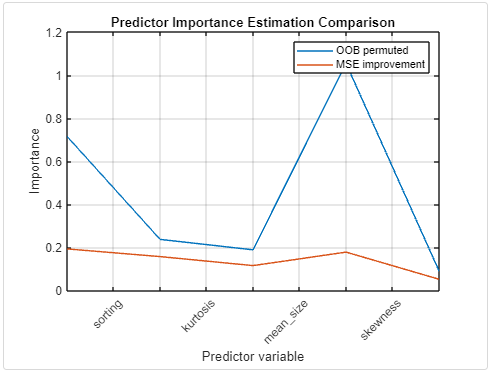


Fig. 8: Line graph showing the overall predictor variable importance (OOB & MSE)

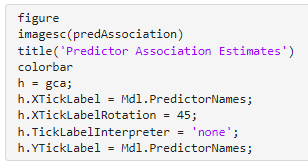


Fig. 9: MATLAB code for generating a gradient grid comparing pair-wise interaction of predictor variables

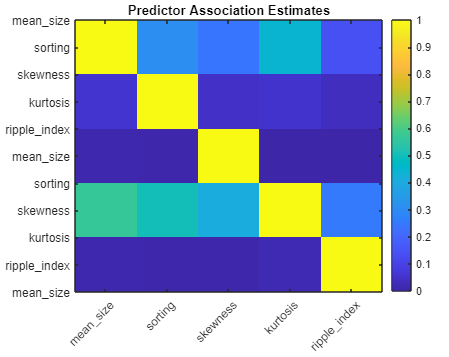


Fig. 10: Gradient grid visualizing pair-wise interaction of features

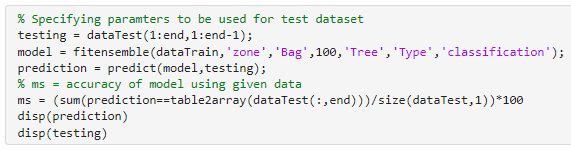


Fig. 11: MATLAB code for using training datasets to predict zones of training datasets

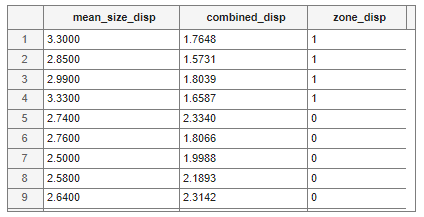


Fig. 12: Table displaying predicted zones of training datasets

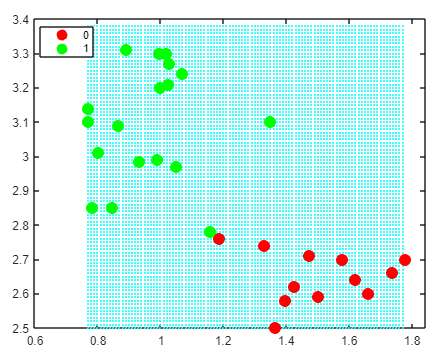


Fig. 13: Mesh grid - mean size vs kurtosis

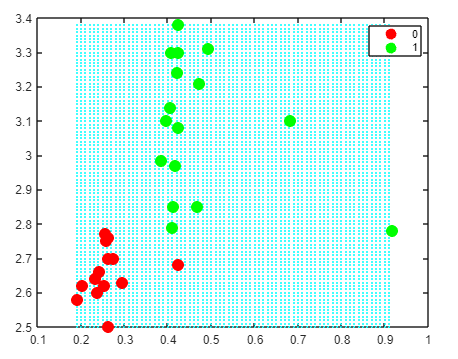


Fig. 14: Mesh grid - mean size vs sorting

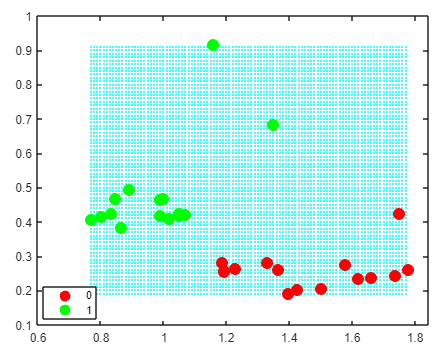


Fig. 15: Mesh grid - kurtosis vs sorting