Lab-1 [Ensemble Methods]

Out date: August 1, 2022
Due date: August 7, 2022 at 11:59PM

Submission

1. Prepare your solution in Orange and save the workspace for Problem 1 (e.g., Badge3_Lab-1_LastName.ows) **[20 points]**
2. Complete the tables given below and save the file (e.g., Badge3_Lab-1_LastName.docx). **[80 points]**
3. Upload the files to the Canvas.

Objective: To understand how to use Ensemble Methods based ML algorithms such as Random Forest, Ada Boost, Stacking and how to handle class imbalance problem.

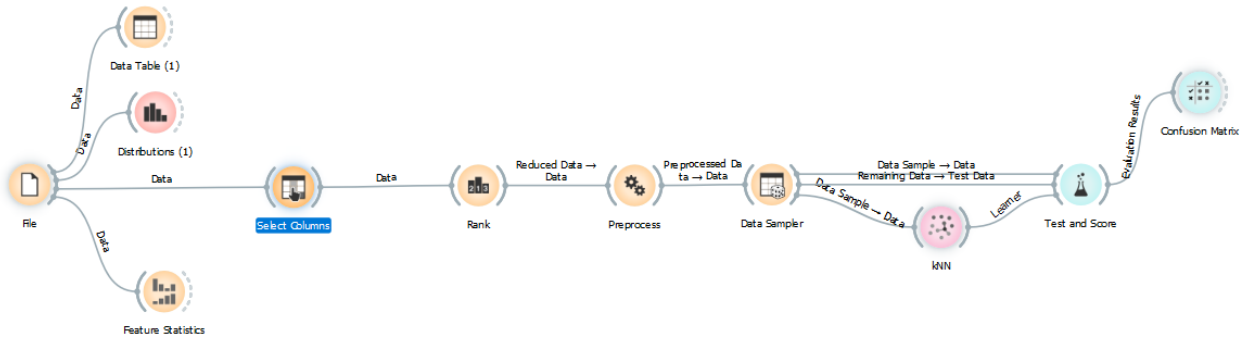
Problem 1 [100 points]

Data: For this lab, please download *GOMFields_Reserves_Processed.csv* and *Badge3_Lab1_Start.ows* files from Canvas to your folder.

(**Data Source:** https://www.data.bsee.gov/Main/FieldReserves.aspx#ascii)
Oil & BOE reserves and production in MMbbl
Gas reserves and production in Bcf
Field GOR in SCF/STB

Lab Instructions

1. Open the *Badge3_Lab1_Start.ows* file using Orange. Your pipeline should look as below:

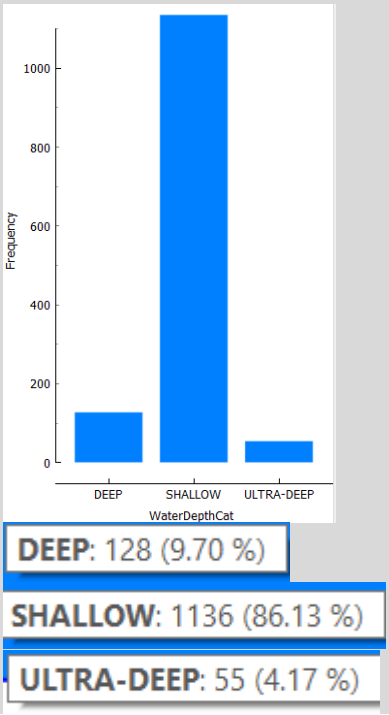


2. Open **File** widget and load the data. Complete the table below: (10 points)

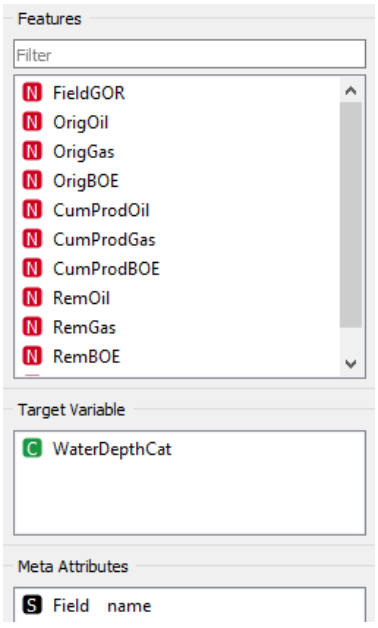
# of instances/rows	1319
# of features, attribute types and roles	12 features, 1 meta (13 columns)
What is the target variable and what are its class labels?	WaterDepthCat C categorical target DEEP, SHALLOW, ULTRA-DEEP

3. Open **Data Table, Feature Statistics and Distributions** widget and inspect features. Complete the table below: (15 points)

List 3 observations based on inspecting the features	No missing values so no need to worry computing, not a balanced target, you can see some history as field name MC

	807 was discovered in 1989, had 1741.9 mil barrels of oil and etc.
Do you see class imbalance issue for the target variable? What are the % of the labels?	Yes, Shallow is more than the other two 

4. Open **Select Columns** widget and ensure selection of variables is as below:



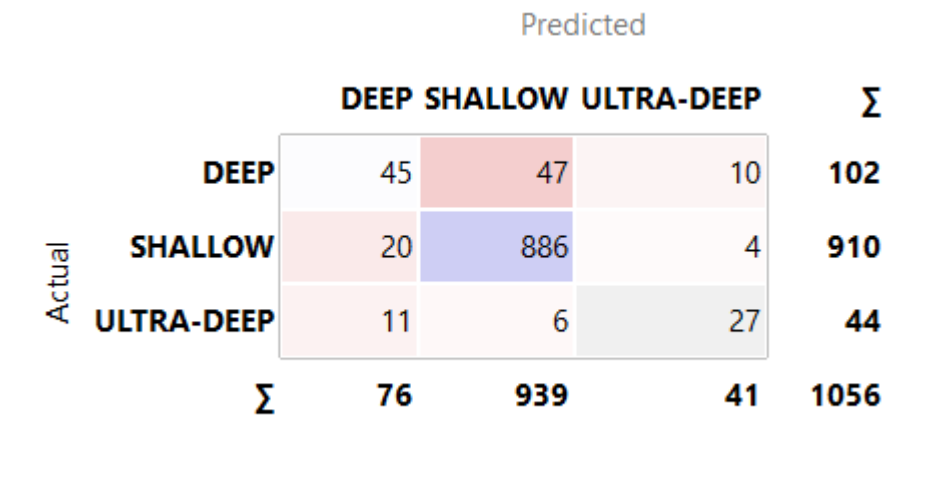
5. Inspect the rest of the pipeline. Open **Test and Score** & **Confusion Matrix** widgets.

   Selecting Cross validation (10 folds) for *Sampling* and observe the Evaluation Results for each of the class and the Confusion Matrix (average over class). (10 points)

What are your observations regarding the model performance?

Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN	0.096	0.047	0.872	0.907	0.901	0.683

Since we have a class imbalance, we would need to resolve this by class weighting/balance to improve specificity of the model; imbalances can affect this



Three class problem;

only 45 of deep fields are correctly predicted by the model, 10 is being misclassified as ultra-deep, remaining 47 as shallow fields;

Shallow looks okay since majority is classified as Shallow (886);

Ultra-Deep, only 27 is correctly classified

Complete the table below: (15 points)

Model / Class	AUC	CA	F1	Specificity
kNN / Average				
kNN / Deep				
kNN / Shallow				

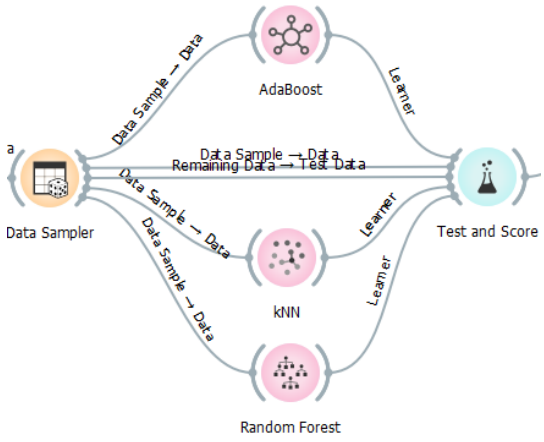
Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN	0.096	0.047	0.872	0.907	0.901	0.683

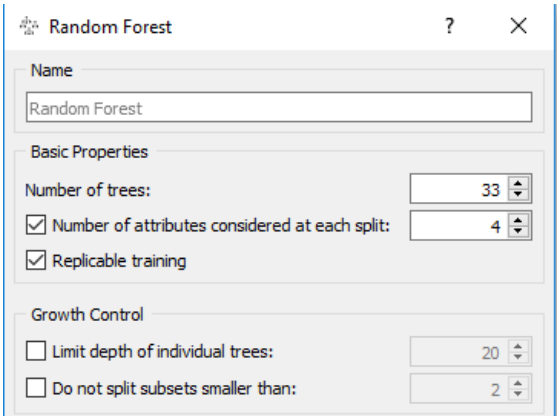
Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN	0.096	0.047	0.831	0.917	0.506	0.968

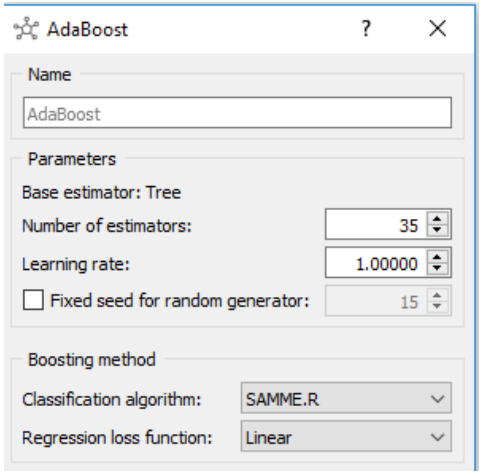
Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN	0.096	0.047	0.898	0.927	0.958	0.637

kNN / Ultra-Deep	Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
	kNN	0.096	0.047	0.893	0.971	0.635	0.986

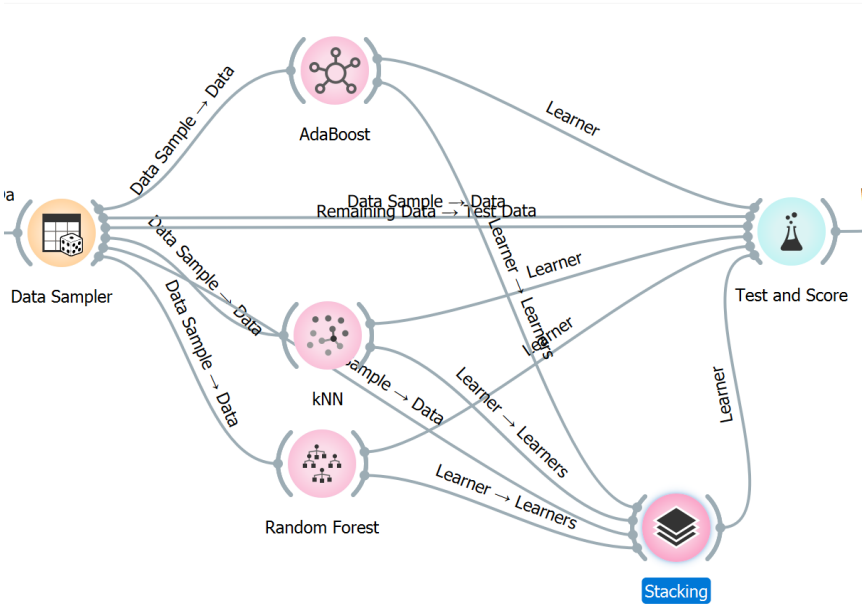
6. Add **Random Forest** and **AdaBoost** models to your pipeline as shown below.







7. Add **Stacking** models to your pipeline as shown below.



8. Select **Cross validation (10 folds)** as *Sampling* in the **Test and Score** widget. Observe Evaluation Results and the Confusion Matrix. (10 points)

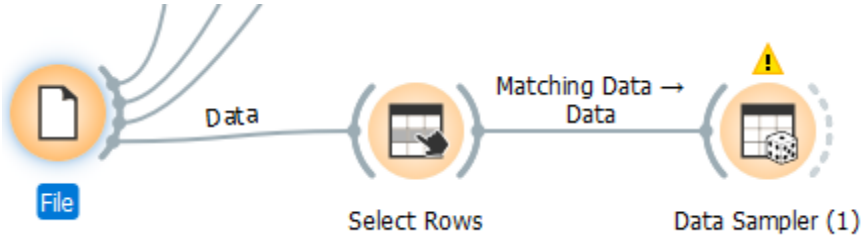
Model / Class	AUC	CA	F1	Specificity
kNN / Average over classes				
Random Forest / Average over classes				
AdaBoost / Average over classes				
Stacking / Average over classes				

Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN	0.096	0.047	0.872	0.907	0.901	0.683
Random Forest	0.826	0.098	0.907	0.904	0.896	0.677
AdaBoost	0.125	0.023	0.765	0.885	0.885	0.734
Stack	6.012	0.182	0.915	0.914	0.906	0.690

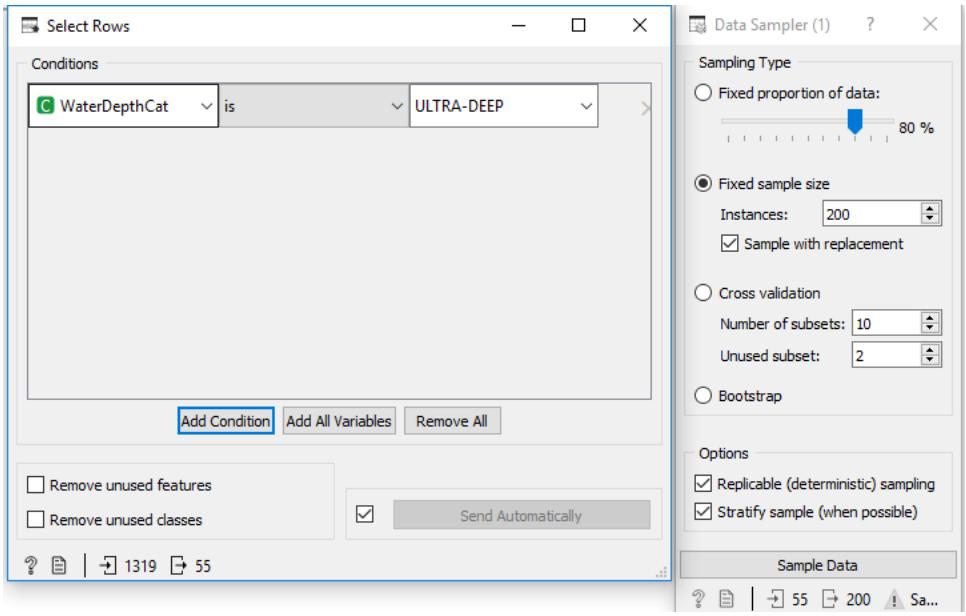
What are your observations on the performance of the 4 models?	Stacking gave highest AUC and CA and F1 but second highest Specificity. AdaBoost highest Specificity

What can be done to improve model performance?	Tune AdaBoost, over sample and under sample (select rows, data sampler to fixed data size…)

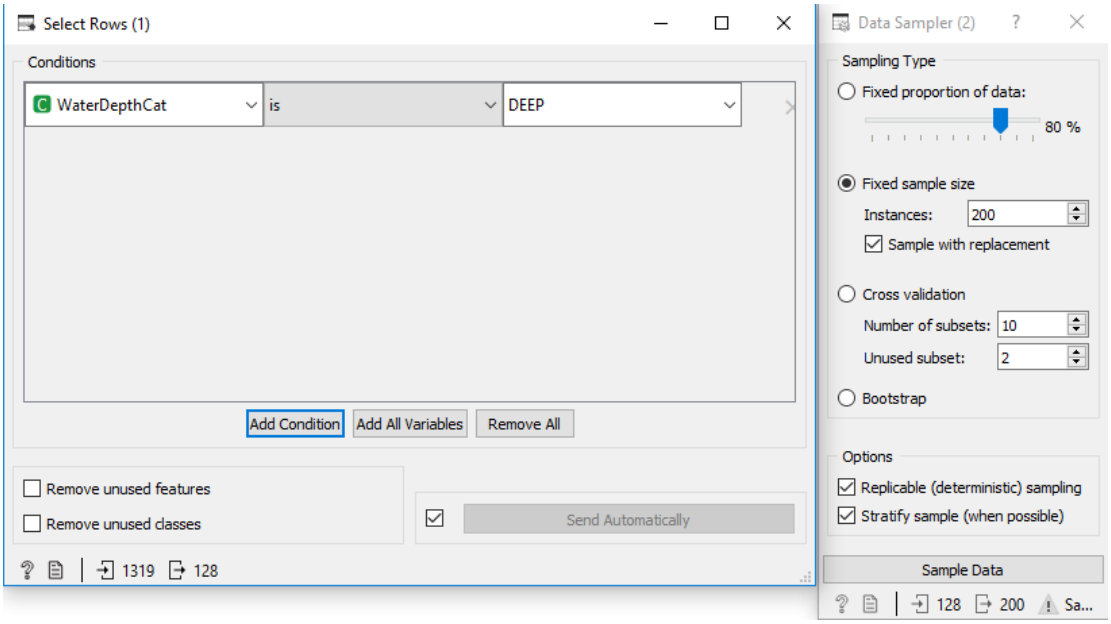
9. Let us oversample the Ultra-Deep class label. Add **Select Rows** and **Data Sampler** widgets and connect them to **File** widget as shown below.



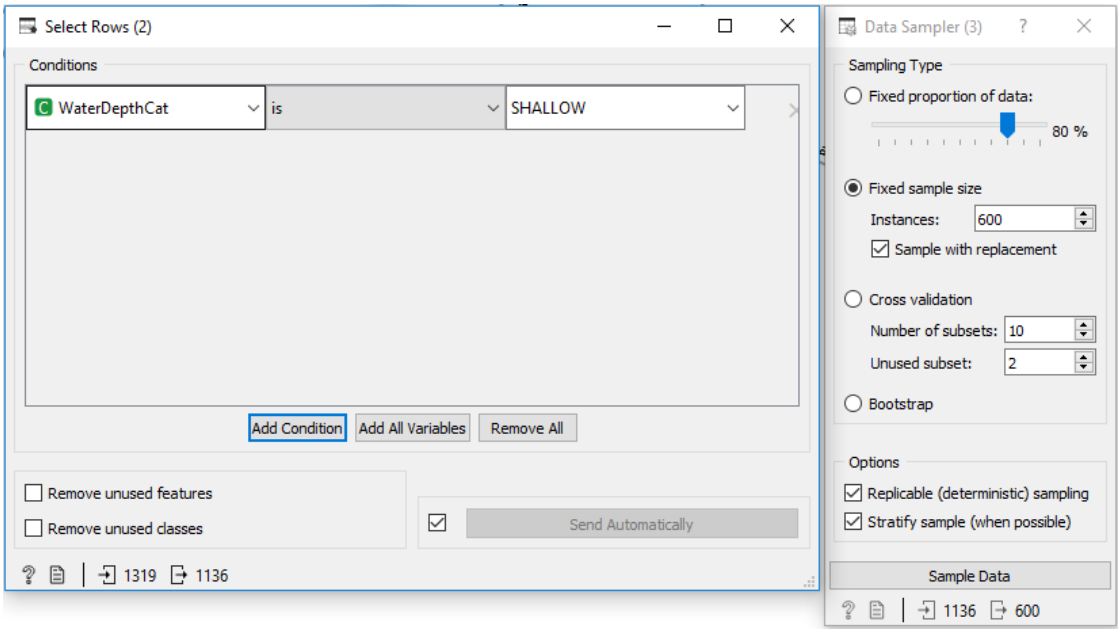
10. Edit **Select Rows** and **Data Sampler** widgets as shown below.



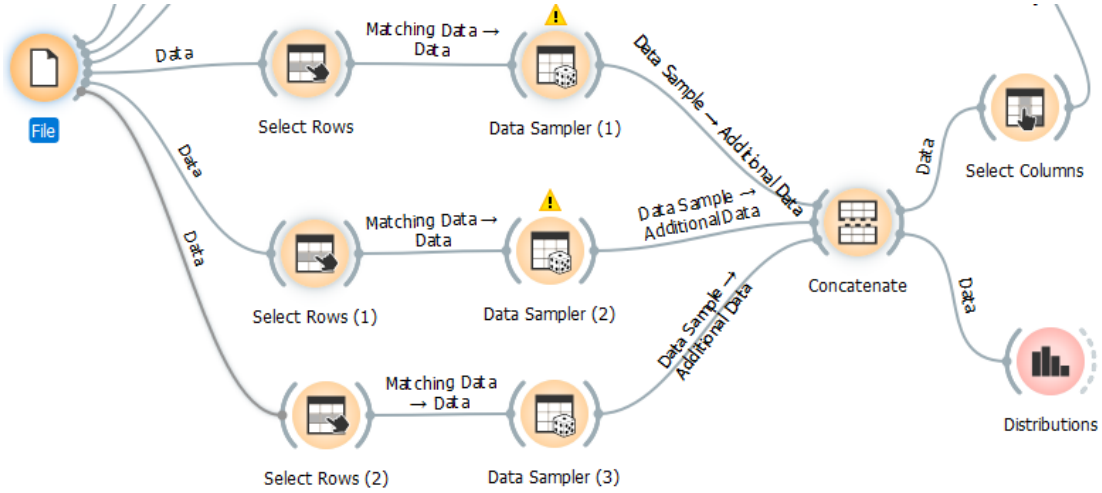
11. Let us oversample the Deep class label. Add another **Select Rows** and **Data Sampler** widgets and connect them to **File** widget. Edit them as below:

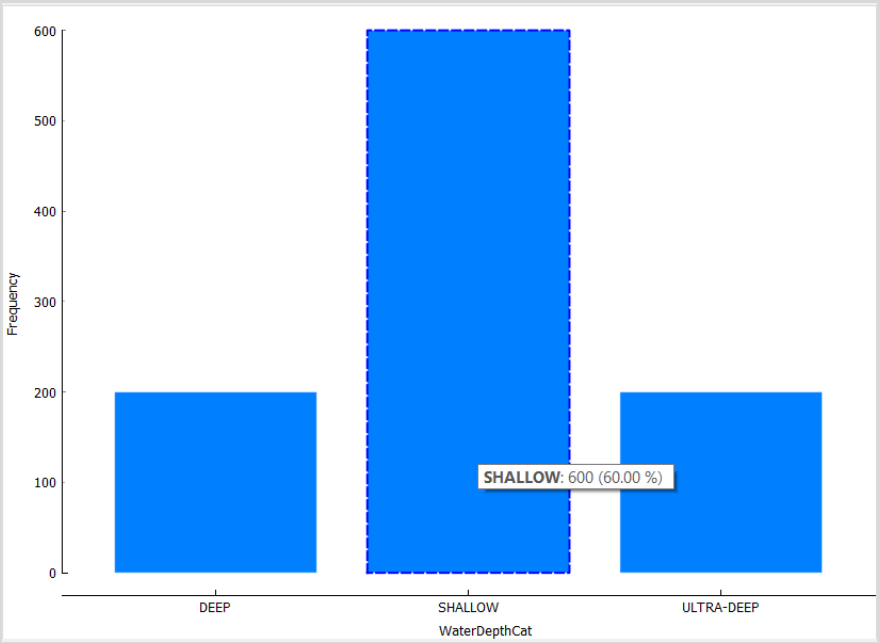


12. Let us under sample Shallow class label. Add another **Select Rows** widget and **Data Sampler widget** and connect them to **File** widget. Edit them as below:

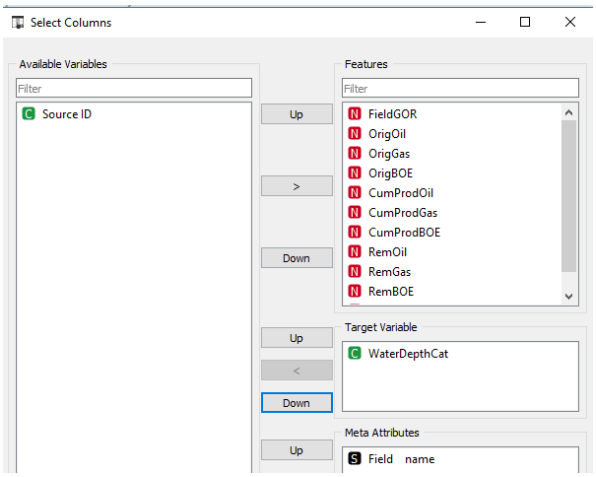


13. Add **Concatenate** widget and connect the 3 **Data Sampler** widgets. Connect **Concatenate** widget to **Select Columns** widget. Add **Distributions** widget to observe class label distribution.
    Pipeline should now look like this:



How does the class label distributions look now for each of the 3 class labels? (15 points)	Still imbalanced but improved. 

14. Open **Select Columns** widget and ensure variable selections are as below:



15. Open **Test and Score** and **Confusion Matrix** widgets. Select **Cross validation (10 folds)** for *sampling* in the **Test and Score** widget. Observe Evaluation Results and the Confusion Matrix. (10 points)

What are your observations on the performance of the 4 models?

Much definitely improved

Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN	0.096	0.047	0.872	0.907	0.901	0.683
Stack	6.012	0.182	0.915	0.914	0.906	0.690
Random Forest	0.826	0.098	0.907	0.904	0.896	0.677
AdaBoost	0.125	0.023	0.765	0.885	0.885	0.734

Vs

Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN	0.095	0.051	0.974	0.946	0.946	0.960
Stack	8.020	0.186	0.986	0.958	0.957	0.964
Random Forest (1)	0.781	0.071	0.983	0.954	0.953	0.967
AdaBoost (1)	0.152	0.018	0.953	0.946	0.946	0.965

Seems stacking is not too essential since it increases train time, and can be costly if huge dataset.

Has the model performance improved after over / under sampling of class labels?	Yes!

16. Change *sampling* to **Test on test data** in the **Test and Score** widget. Complete the table below: (15 points)

Model / Class	AUC	CA	F1	Specificity
kNN				
Random Forest				
AdaBoost				
Stacking				

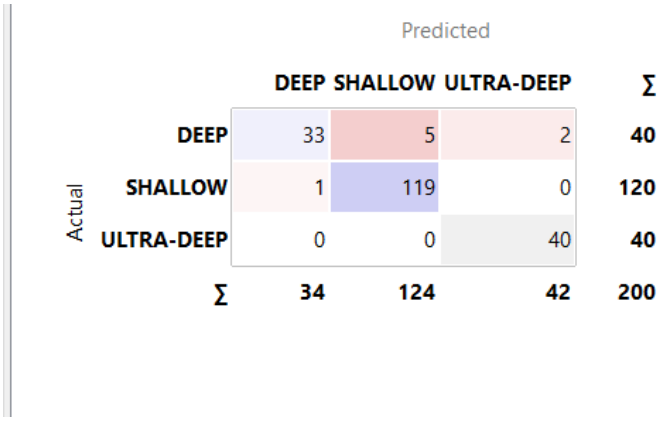
Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN	0.027	0.006	0.974	0.960	0.959	0.959
Random Forest (1)	0.080	0.009	0.975	0.935	0.936	0.952
AdaBoost (1)	0.015	0.003	0.953	0.945	0.945	0.955
Stack	0.594	0.019	0.979	0.965	0.965	0.960

2:23:XX

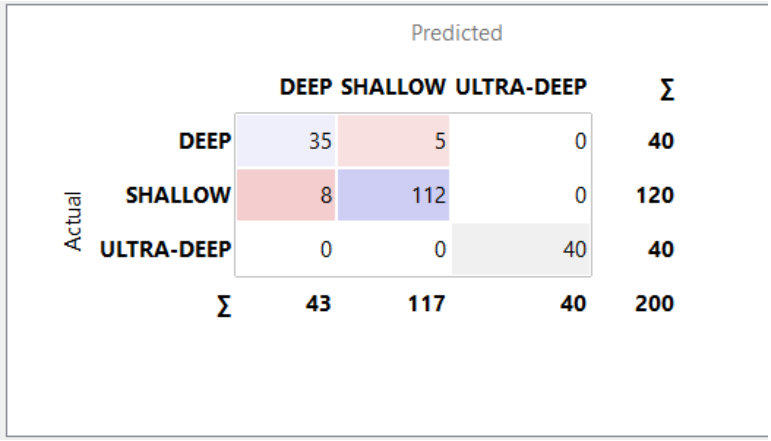
Observe the **Confusion Matrix** widget for predictions of each class label.

Which model would you select as your final model?	Stack actually did good, not considering train time

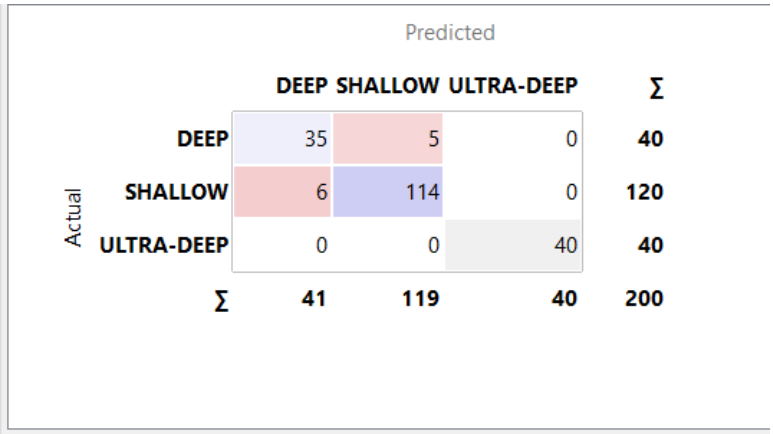
kNN:



Random Forest:



AdaBoost:



Stack: