Badge-3 Lab-3 [Neural Network]

Out date: Aug 8, 2022
Due date: Aug 14, 2022 at 11:59PM

Submission

1. Prepare your solution in Orange and save the workspace for Problem 1 (e.g., Lab-3_1_LastName.ows) **[10 points]**
2. Prepare your solutions in Orange and save the workspace for Problem 2 (e.g., Lab-3_2_LastName.ows) **[10 points]**
3. Complete the tables given below and save the file (e.g., Lab-3_LastName.docx). **[80 points]**
4. Upload the files to the Canvas.

Objective(s):

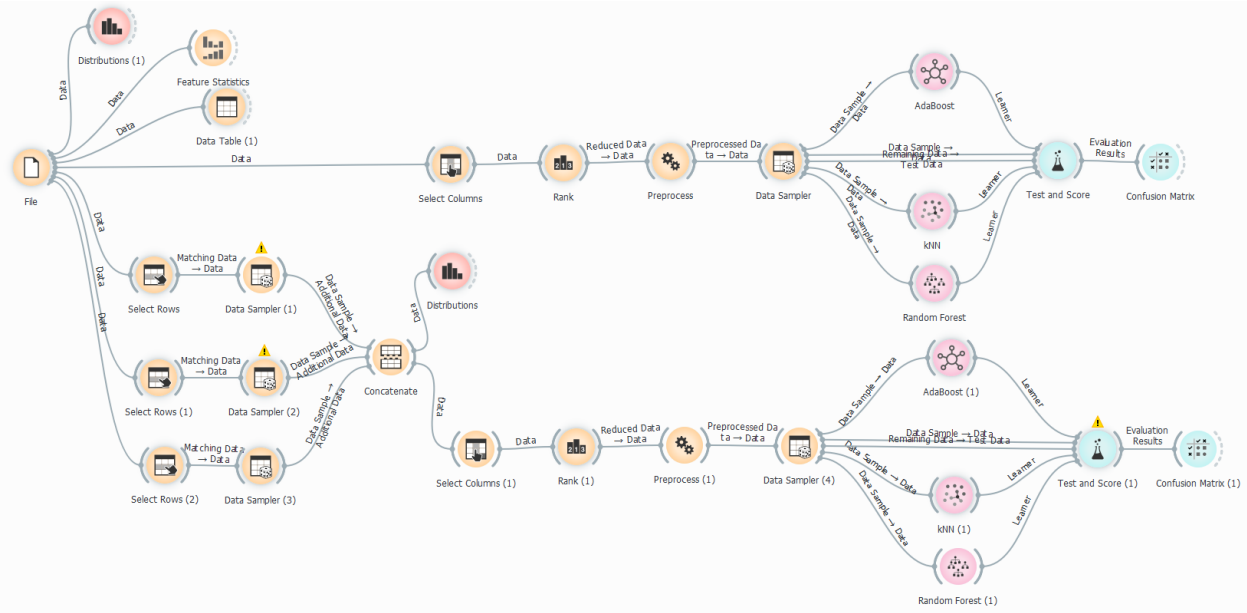
To apply Neural Network algorithm for classification and regression problems and compare its performance with other machine learning algorithms.

Problem 1/2. [50 points]

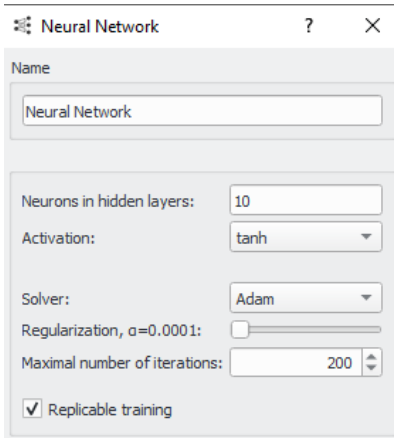
Data: For this lab, please download *GOMFields_Reserves_Processed.csv* and *WaterDepthCatPrediction_Start.ows* from Canvas to your folder.

Lab Instructions

1. Open the *WaterDepthCatPrediction_Start.ows* file in Orange. Your pipeline should look as below. Confirm the *GOMFields_Reserves_Processed.csv* dataset is loaded by opening the **File** widget.



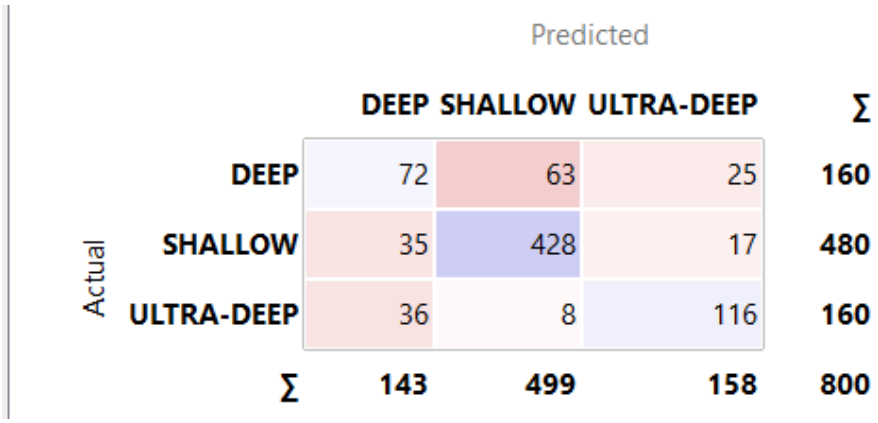
2. Inspect the pipeline. Add **Neural Network** widget to the second workflow with over/under sampling, with parameters as shown below, connect it to the Data Sampler and **Test and Score** widgets. Complete the table below for **Target Class as Average over classes** and **Cross Validation, 5 folds** for sampling: (20 points)



Model	AUC	CA	F1	Specificity

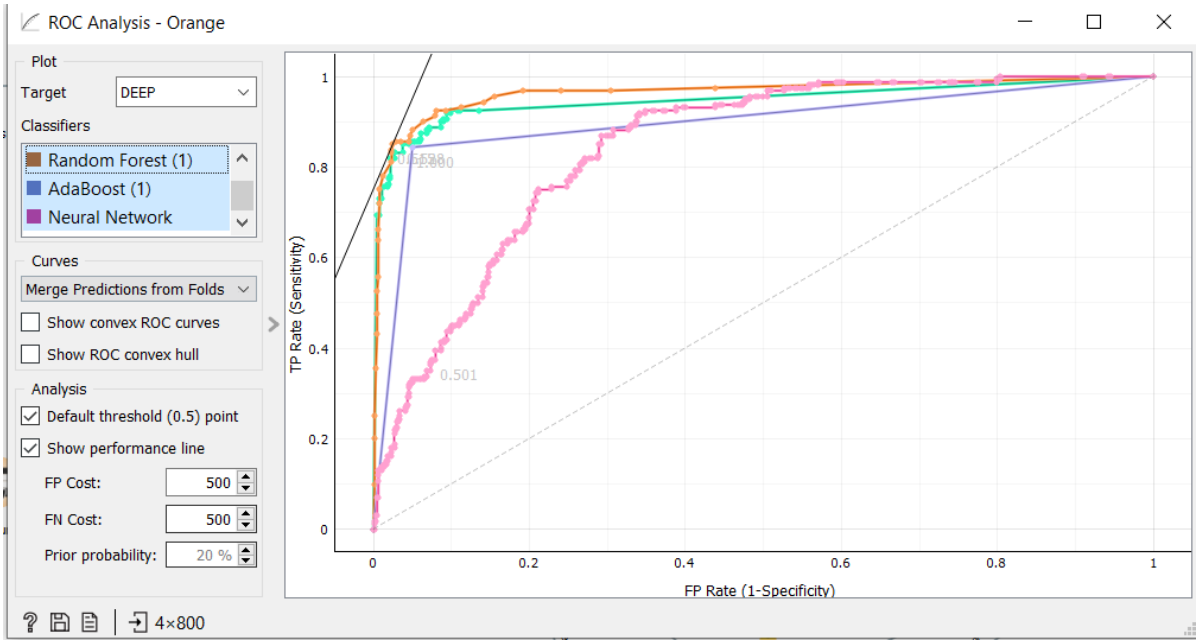
Model	Train time [s]	Test time [s]	AUC	CA	F1	Precision	Recall	Specificity
kNN	0.057	0.032	0.968	0.925	0.925	0.926	0.925	0.953
Random Forest (1)	0.417	0.052	0.982	0.945	0.945	0.945	0.945	0.964
Neural Network	2.464	0.014	0.914	0.770	0.766	0.762	0.770	0.832
AdaBoost (1)	0.077	0.014	0.935	0.916	0.917	0.919	0.916	0.957

Observe the performance of the models by viewing the Confusion Matrix and ROC widgets. Compare the models and comment on the performance.	Neural taken the most time, not the best in terms of reach the best optimal accuracy, but more improvement needed. Other models are performing better. The ROC for Neural has a wide gap compared to the others. Confusion, Shallow and Ultra-Deep has low predicting correctly, there is larger misclassification with Ultra-deep; Shallow looks okay since majority is classified as Shallow.



Predicted

Actual	DEEP	SHALLOW	ULTRA-DEEP	∑
DEEP	72	63	25	160
SHALLOW	35	428	17	480
ULTRA-DEEP	36	8	116	160
∑	143	499	158	800



3. Tuning: Modify the Neural Network parameters as below and complete the table:
   Use **Cross Validation, 5 folds** as Sampling and Target Class as Average over classes. (30 points)

Model	AUC	CA	Sensitivity (Recall)	Specificity
NN (10, tanh, Adam, 0.001)	0.914	0.770	0.770	0.832
NN (100, tanh, Adam, 0.001)	0.939	0.830	0.830	0.867
NN (100, tanh, Adam, 0.01)	0.938	0.834	0.834	0.868
NN (100, tanh, Adam, 0.1)	0.937	0.828	0.828	0.863
NN (200, tanh, Adam, 0.01, 300 iterations)	0.941	0.834	0.834	0.869

NN (200, ReLu, Adam, 0.01, 300 iterations)	0.948	0.864	0.864	0.892
NN(50,100, ReLu, Adam, 0.01, 300 iterations)	0.955	0.876	0.876	0.907
NN(50,100,50 ReLu, Adam, 0.01, 300 iterations)	0.952	0.887	0.887	0.912

Compare the various NN models and compare its performance with RF and Logistic Regression.

Seems the last model did very well overall.

Neural Network	0.952	0.887	0.887	0.912
Logistic Regression	0.885	0.733	0.733	0.813

It did better than the LR.

Comment on the Train time requirements of NN compared to the other models

Model	Train time [s]	Test time [s]	AUC	CA	Recall	Specificity
kNN	0.057	0.032	0.968	0.925	0.925	0.953
Random Forest (1)	0.417	0.052	0.982	0.945	0.945	0.964
Neural Network	15.869	0.035	0.952	0.887	0.887	0.912
Logistic Regression	0.262	0.017	0.885	0.733	0.733	0.813
AdaBoost (1)	0.077	0.014	0.935	0.916	0.916	0.957

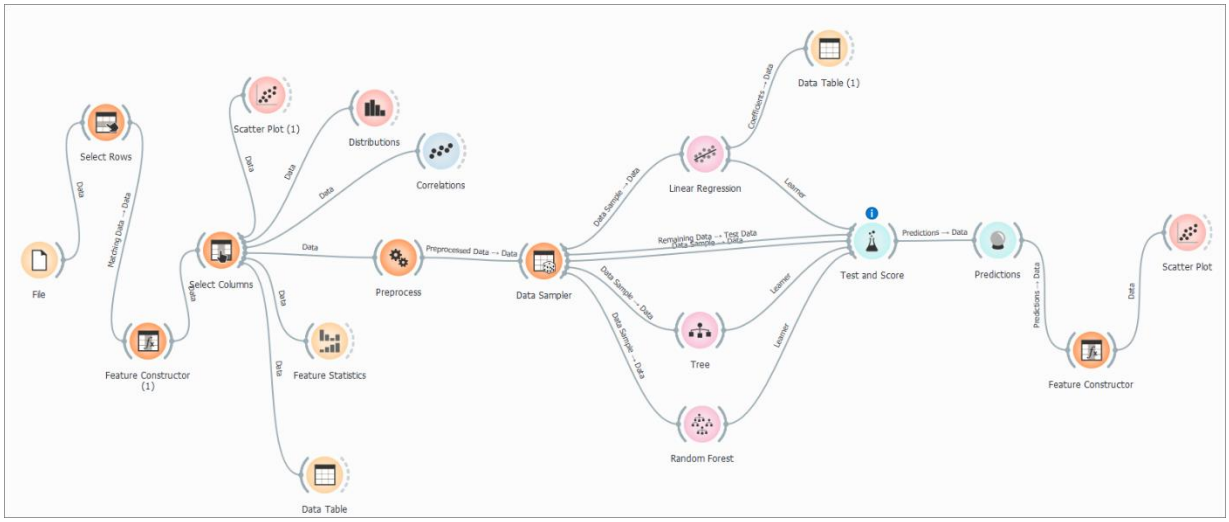
But the train time is very large.

Problem 2/2. [50 points]

Data: For this lab, please download *data.xlsx* and *gas_prod_pred_lab3start.ows* from Canvas to your folder.

Lab Instructions

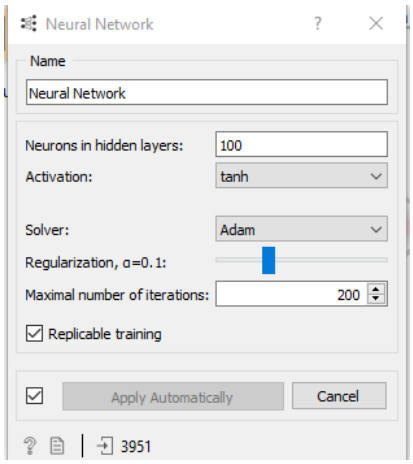
1. Open the *gas_prod_pred_lab3start.ows* file in Orange. Your pipeline should look as below. Confirm the *data.xlsx* dataset is loaded by opening the **File** widget.



2. Inspect the pipeline. Open **Test and Score** widget and complete the table below using **Cross Validation, 5 folds** for sampling: (6 points)

Model	RMSE	MAE	R2
Linear Regression	0.622	0.431	0.635
Tree	0.592	0.401	0.670
Random Forest	0.518	0.344	0.747

3. Add a **Neural Net** model to your pipeline. Using **Cross Validation, 5 folds** for sampling, complete the following table:



(9 points)

Model	RMSE	MAE	R2
Linear Regression	0.622	0.431	0.635
Neural Network	0.599	0.414	0.662
Random Forest	0.518	0.344	0.747
Tree	0.592	0.401	0.670

4. Try the following **Neural Network** parameters and complete the table below:
(35 points)

Model	RMSE	MAE	R2
Neural Network (Hidden Layers=100, Activation= tanh, Alpha = 0.5	0.600	0.414	0.660
Neural Network (Hidden Layers=50, 100,50 Activation= tanh, Alpha = 0.1	0.592	0.408	0.669
Neural Network (Hidden Layers=50, 100,50 Activation= tanh, Alpha = 0.005	0.601	0.418	0.659

Neural Network (Hidden Layers=50, 500,50 Activation= tanh, Alpha = 0.005	0.606 0.413 0.654

Comment on the performance of the various NN models.	Second seems to be the best, but the specificity of all varies slightly, even all the values.