Badge-2 Lab-5 [Naïve Bayes and Handling Class Imbalance]

Out date: Jul 25, 2022
Due date: Jul 31, 2022 at 11:59PM

Submission

1. Prepare your solution in Orange and save the workspace for Problem 1 (e.g., Lab-5_LastName.ows) **[10 points]**
2. Prepare your solutions in Orange and save the workspace for Problem 2 (e.g., Lab-5_LastName.ows) **[10 points]**
3. Complete the tables given below and save the file (e.g., Lab-5_LastName.docx). **[80 points]**
4. Upload the files to the Canvas.

Objective(s):

To apply Naïve Bayes classifier for a classification problem and compare its performance with other machine learning algorithms.
Learn a way to address class imbalance and see its effect on model performance.

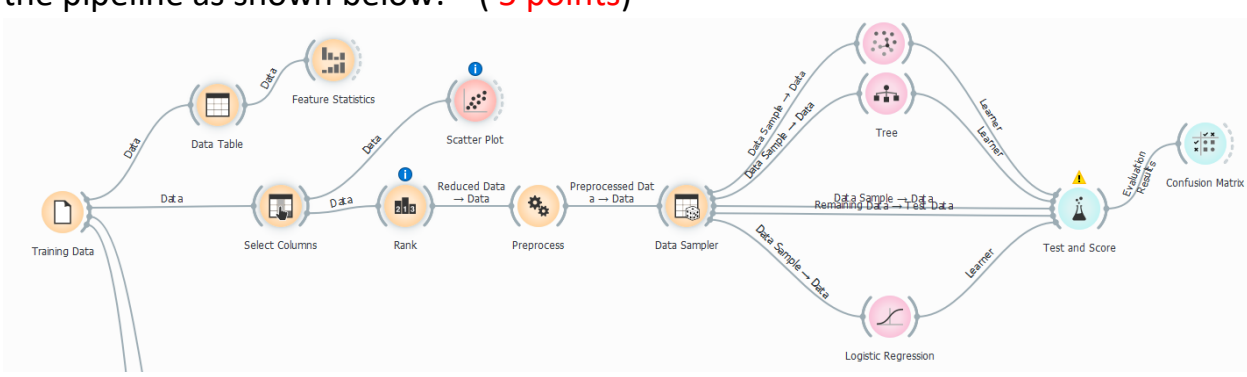
Problem 1.

Data: For this lab, please download *Train.csv* and *Lab5_Start.ows* files from Canvas to your folder.

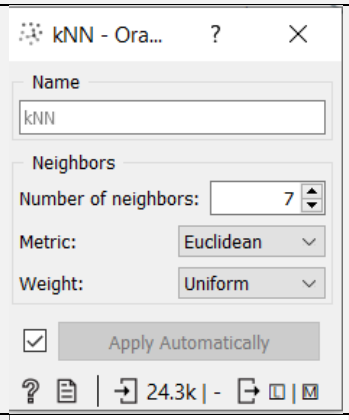
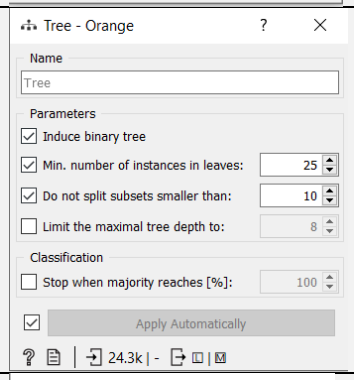
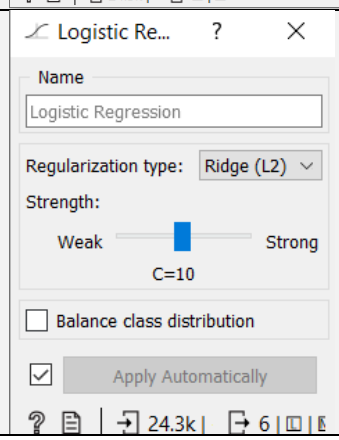
Data Source: https://gdr.openei.org/submissions/1111

Lab Instructions

1. Launch Orange. Open *Lab5_Start.ows, Train.csv* and verify that you can see the pipeline as shown below: (5 points)



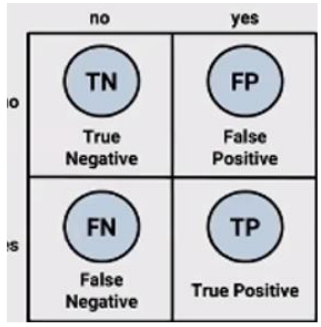
2. Inspect the pipeline and complete the table below: (10 points)

Data Set	Model Parameters	Features Used	CA	F1	Specificity
Training Set **CV, 5 folds**	kNN		0.990 0.984	0.990 0.984	0.959 0.937
Training Set **CV, 5 folds**	Tree		0.990 0.986	0.990 0.986	0.958 0.944
Training Set **CV, 5 folds**	Log Reg		0.905 0.905	0.895 0.895	0.481 0.479

3. Open **Confusion Matrix** widget. Examine this widget and complete the table below. Consider **Granitiod** as the positive class: (10 points)

Model Parameters	TP	TN	FP	FN
KNN, 7 Neighbors, Euclidean, Uniform	**21206**	2726	205	191
Tree (Bin Tree- Yes Min Instances: 25	21229	2747	184	168

Subsets smaller than: 10 Majority %: Unchecked)				
Logistic Regression C=10	20812	1206	1725	585

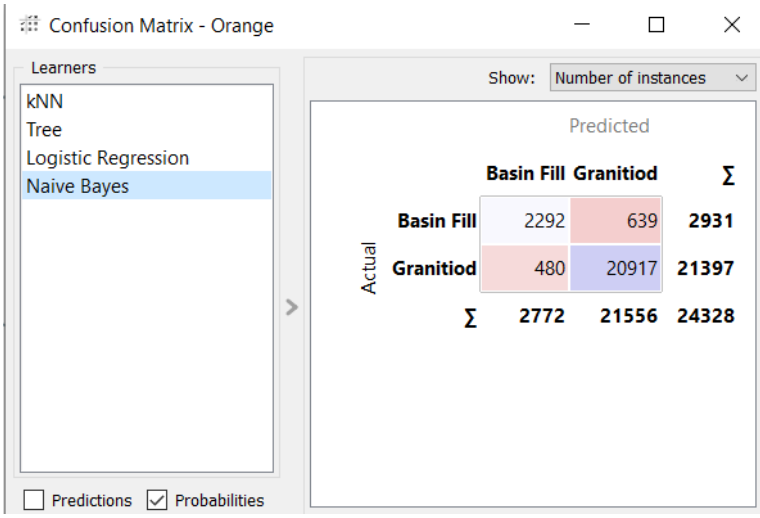


4. Add a **Naïve Bayes** model to the above pipeline.

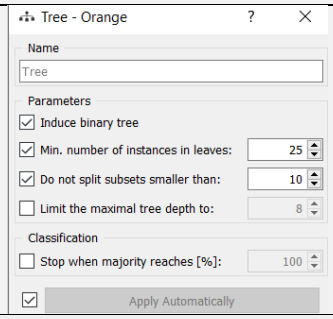
Complete the table below: (15 points).

Data Set	Model Parameters	Features Used	CA	F1	Specificity
Training Set **CV, 5folds**	kNN	Consistent to above	0.990 0.984	0.990 0.984	0.959 0.937
Training Set **CV, 5folds**	Tree	Consistent to above	0.990 0.986	0.990 0.986	0.958 0.944
Training Set **CV, 5 folds**	Naïve Bayes	Consistent to above	0.954 0.954	0.954 0.953	0.803 0.806
Training Set **CV, 5 folds**	Log Reg	Consistent to above	0.905 0.905	0.895 0.895	0.481 0.479

What is your observation of the Naïve Bayes model performance?	It has almost the fastest training and test time for both Training and CV-5 fold sets



Complete the table below for the best performing model on test data using **Test on test data**:

Data Set	Model Parameters	Features Used	CA	F1	Specificity
Test Data	Tree		0.987	0.987	0.943

Evaluation Results

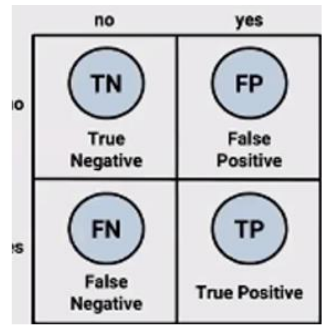
Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN	0.059	0.350	0.993	0.985	0.985	0.933
Tree	0.098	0.001	0.993	0.987	0.987	0.943
Naive Bayes	0.021	0.003	0.972	0.955	0.955	0.808
Logistic Regression	0.083	0.003	0.932	0.909	0.900	0.494

Confusion Matrix results for the best model on test data, **Granitiod** is the positive class:

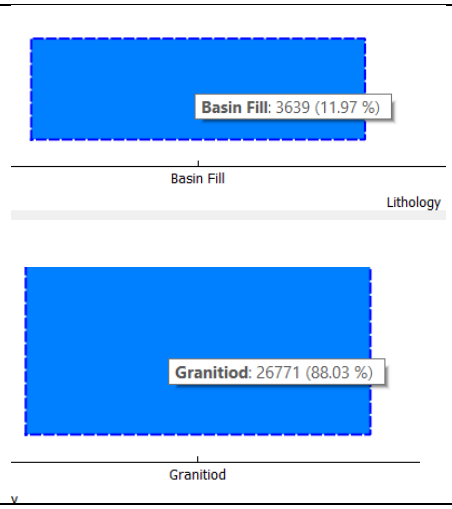
TP	TN	FP	FN
5341	663	45	33

Predicted

Actual	Basin Fill	Granitiod	∑
Basin Fill	663	45	708
Granitiod	33	5341	5374
∑	696	5386	6082



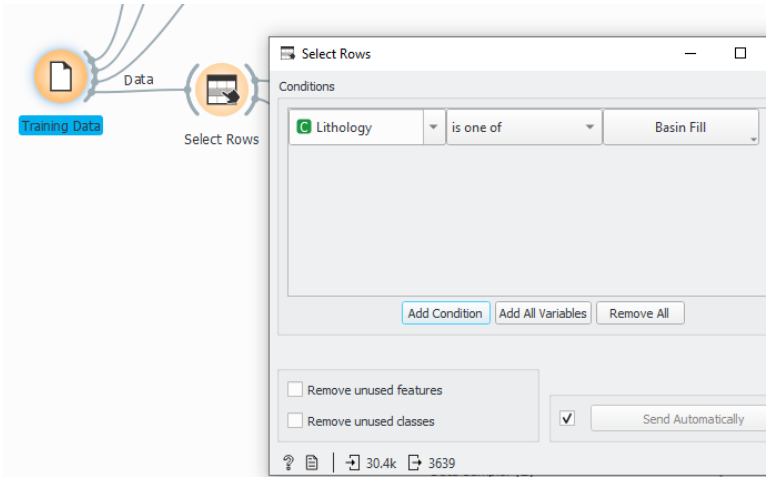
5. Target class distribution in the dataset: (4 points)

What is the distribution of the 2 target classes? Use the **Distributions** widget.	

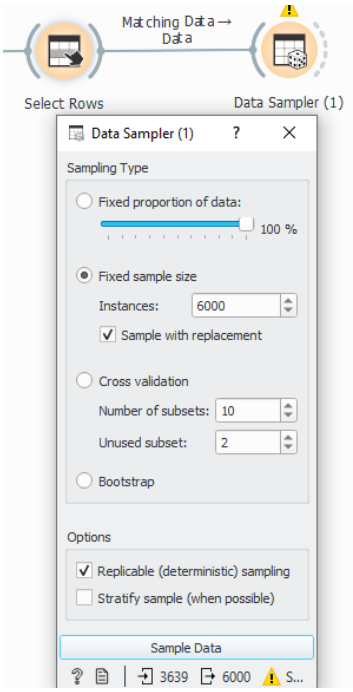
2:00:00

6. Add **Select Rows** widget to the dataset as shown below:

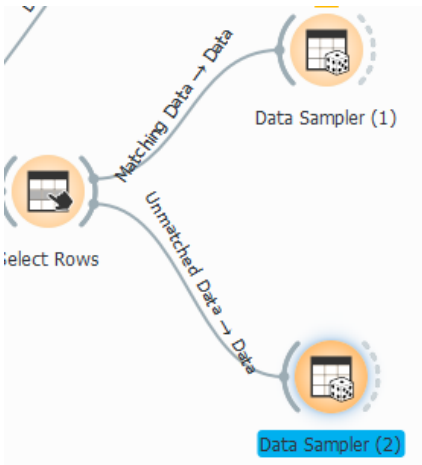
Filter **Basin Fill** values. (6 points)

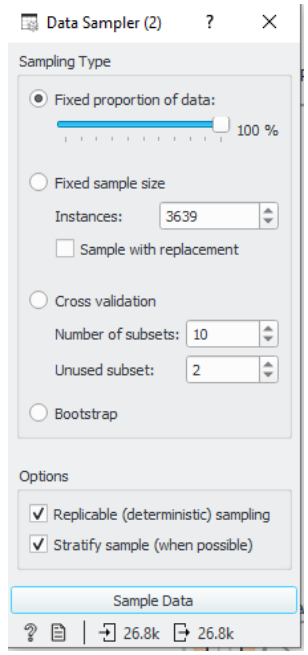


7. Add **Data Sampler** widget to **Select Rows** widget as shown below: (5 points)

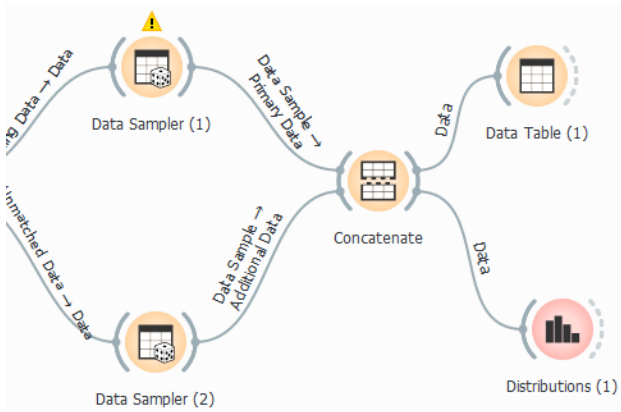


8. Add another **Data Sampler** widget to the **Select Rows** widget as shown below: (5 points)

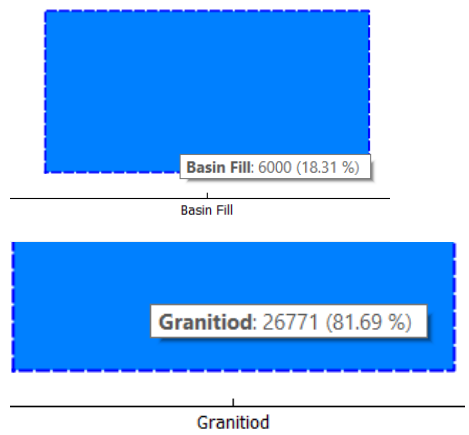




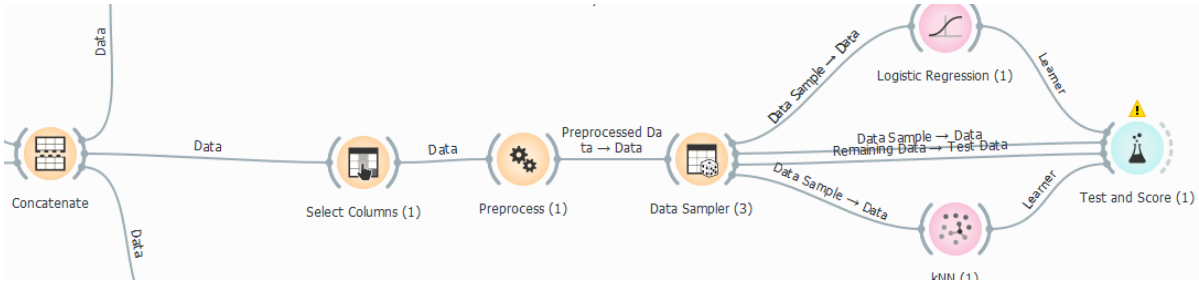
9. Combine the two datasets using the **Concatenate** widget as shown below. Add a **Data table & Distributions** widgets as well: (5 points)



10. What is the Distribution of the class labels now? Use **Distributions** widget to answer this question: (5 points)

What is the distribution of the 2 target classes? Use the **Distributions** widget.	

11. Complete the rest of the pipeline as shown below. Use the same parameters as in the earlier pipeline (Step 1). (10 points)



12.Open **Test and Score** widget and complete the table below. (5 points)

Data Set	Model Parameters	Features Used	CA	F1	Specificity
Training Set	Evaluation Results				

Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN (1)	0.068	1.593	0.999	0.990	0.990	0.981
Tree (1)	0.136	0.003	0.999	0.990	0.990	0.982
Naive Bayes (1)	0.027	0.011	0.974	0.945	0.946	0.903
Logistic Regression (1)	0.098	0.006	0.935	0.881	0.877	0.667

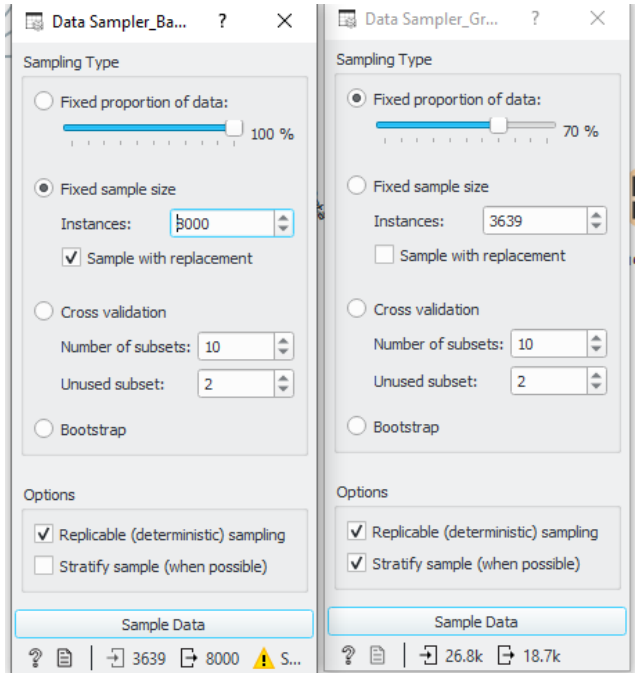
CV, 5 folds

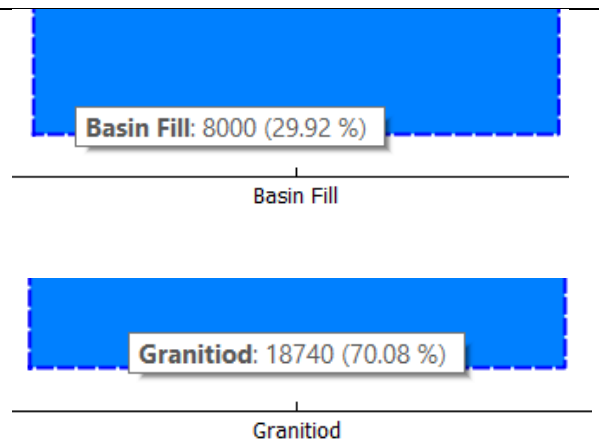
Evaluation Results

Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN (1)	0.202	1.488	0.997	0.983	0.983	0.967
Tree (1)	0.525	0.005	0.996	0.984	0.984	0.966
Naive Bayes (1)	0.110	0.015	0.973	0.940	0.941	0.897
Logistic Regression (1)	0.395	0.015	0.935	0.881	0.877	0.666

How do the results compare with the earlier models where class imbalance was not addressed?	The model is performing better now than before; but be mindful that over sampling has to be done with caution and that sampling distribution may not be reality when actual put the model in production (you are just trying to get the model to perform better, but at what cost)

13. Make the following changes to the 2 **Data Sampler** widgets used for over / under sampling of the class labels. (15 points)



What is the distribution of the 2 target classes? Use the **Distributions** widget.	

Data Set	Model Parameters	Features Used	CA	F1	Specificity
Training Set					

Evaluation Results

Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN (1)	0.061	1.212	1.000	0.988	0.988	0.990
Tree (1)	0.121	0.003	1.000	0.990	0.990	0.989
Naive Bayes (1)	0.025	0.008	0.967	0.916	0.917	0.912
Logistic Regression (1)	0.089	0.007	0.938	0.890	0.891	0.867

CV, 5 folds

Evaluation Results

Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN (1)	0.177	1.183	0.997	0.982	0.982	0.982
Tree (1)	0.539	0.003	0.997	0.984	0.984	0.980
Naive Bayes (1)	0.075	0.007	0.967	0.915	0.916	0.910
Logistic Regression (1)	0.347	0.010	0.938	0.889	0.890	0.867

How do the results compare with the earlier models where class imbalance was not addressed?	Improved the performance for the most parts.

Old CV-5 folds:

Evaluation Results

Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN	0.174	1.378	0.993	0.984	0.984	0.937
Tree	0.412	0.000	0.994	0.986	0.986	0.944
Naive Bayes	0.100	0.011	0.976	0.954	0.953	0.806
Logistic Regression	0.363	0.011	0.936	0.905	0.895	0.479

Old Training Set:

Evaluation Results

Model	Train time [s]	Test time [s]	AUC	CA	F1	Specificity
kNN	0.062	1.527	0.999	0.990	0.990	0.959
Tree	0.110	0.003	0.999	0.990	0.990	0.958
Naive Bayes	0.023	0.008	0.976	0.954	0.954	0.803
Logistic Regression	0.089	0.005	0.936	0.905	0.895	0.481