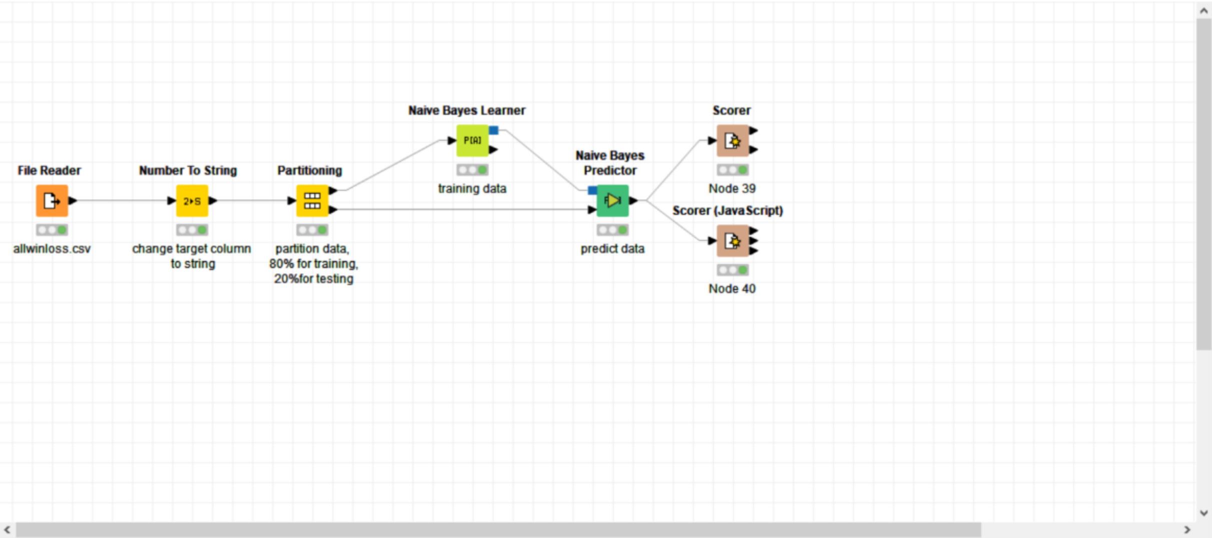
Overall workflow:



I read the allwinloss.csv file since we are doing classification for Naïve Bayes. Naïve Bayes needs the classification column as string type, so I used number to string node to change target column from number to string. Then, I select 80% as training data and 20% as testing data. In class, it says 1/3 of them should be testing data. I will have a try for it then make comparison. I am training the data by Naïve Bayes learner node and predict by predictor. There are 2 nodes that is useful for scoring methods, including classification accuracy, confusion matrix, F-value, and summaries.

Confusion matrix - 3:39 - Scorer

File Edit Hilite Navigation View

Table "spec_name" - Rows: 2 | Spec - Columns: 2 | Properties | Flow Variables

Row ID	I 1	I 0
1	156	45
0	18	174

confusion matrix

File Edit Hilite Navigation View

Table "default" - Rows: 3 | Spec - Columns: 11 | Properties | Flow Variables

Row ID	I TruePo...	I FalsePo...	I TrueNe...	I FalseN...	D Recall	D Precision	D Sensitivity	D Specificity	D F-meas...	D Accuracy	D Cohen'...
1	156	18	174	45	0.776	0.897	0.776	0.906	0.832	?	?
0	174	45	156	18	0.906	0.795	0.906	0.776	0.847	?	?
Overall	?	?	?	?	?	?	?	?	?	0.84	0.68

Accuracy statistics, including data for true positive, false positive, true negative, false negative, accuracy, F-measurement.

Scorer View

Confusion Matrix

Rows Number : 393	0 (Predicted)	1 (Predicted)	
0 (Actual)	174	18	90.63%
1 (Actual)	45	156	77.61%
	79.45%	89.66%	

Class Statistics

Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
0	174	45	156	18	90.63%	79.45%	90.63%	77.61%	84.67%
1	156	18	174	45	77.61%	89.66%	77.61%	90.63%	83.20%

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
83.97%	16.03%	0.680	330	63

A better view for the report, including confusion matrix, F-measure, summary for overall statistics includes accuracy, error, number of correct and incorrect classifications.

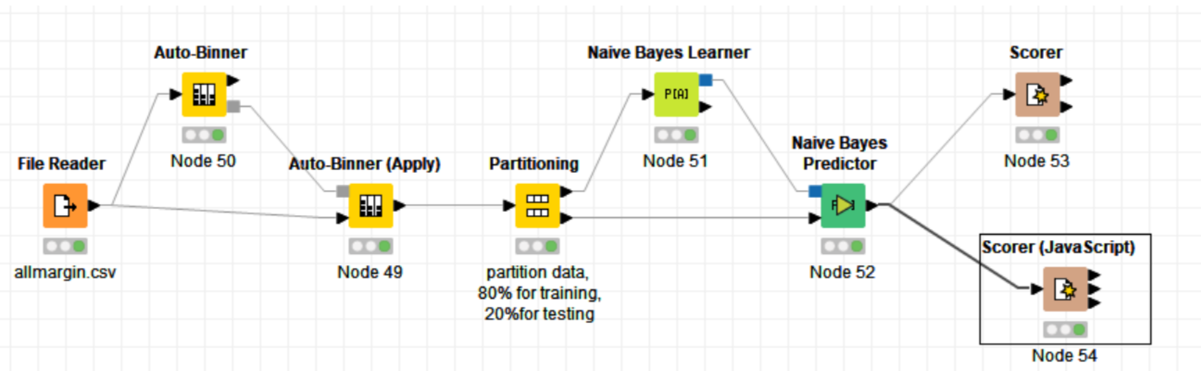
File Edit Hilite Navigation View

Table "default" - Rows: 1 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	D Overall Accuracy	D Overall Error	D Cohen's kappa	I Correctly Classified	I Incorrectly Classified
Overall	0.84	0.16	0.68	330	63

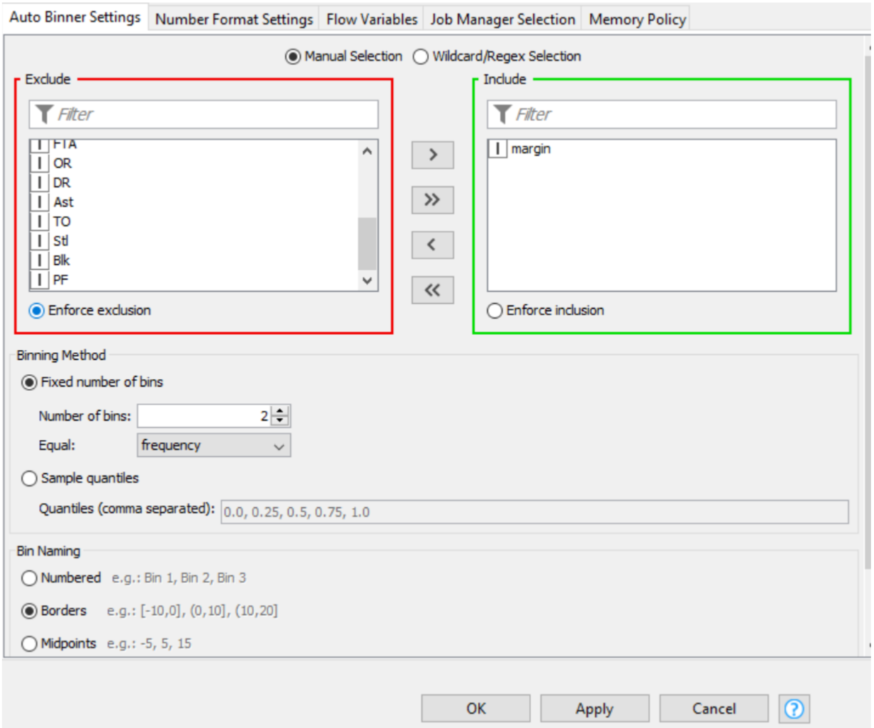
An overall summary for the result.

For the other dataset with numeric result, I am binning the target column into 5 groups, then do the similar thing as above. Here is the workflow:

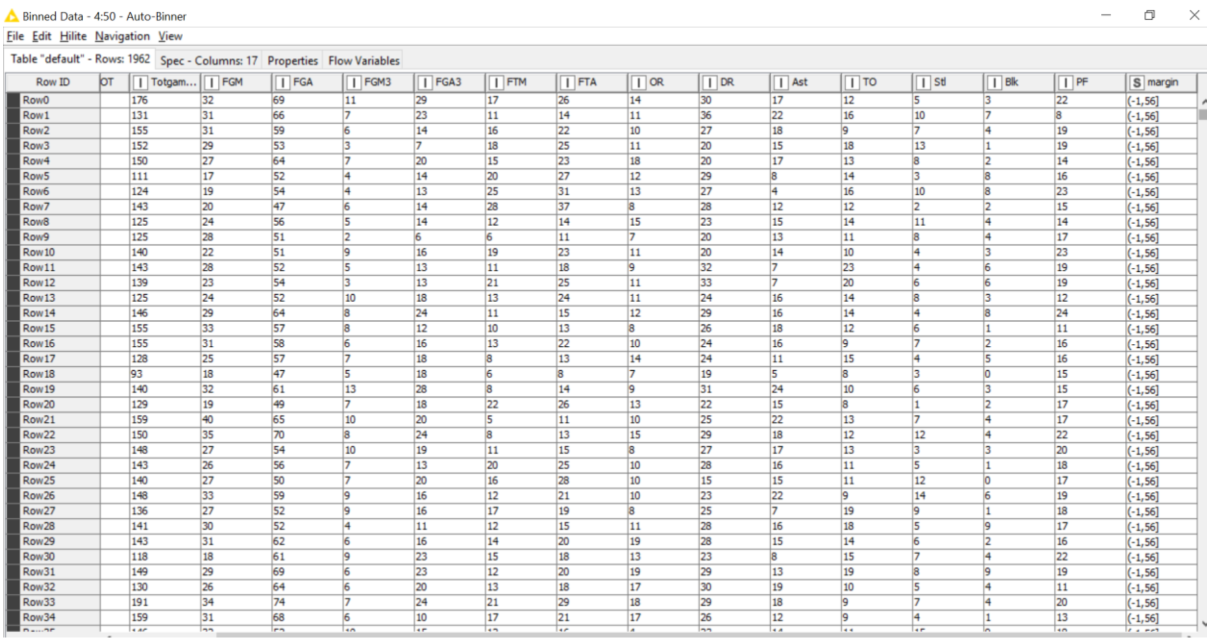


I used auto-binner for binning the target column into 2 bins, since we need to compare with win/loss classifier. Then I set the binning by frequency, naming them by border. Later we can change the naming by win / loss (suppose larger margin group is win, lower is loss). Then apply auto-binner into the dataset.

(auto binner setting)



(binning dataset, look at last column)


Binned Data - 4:50 - Auto-Binner

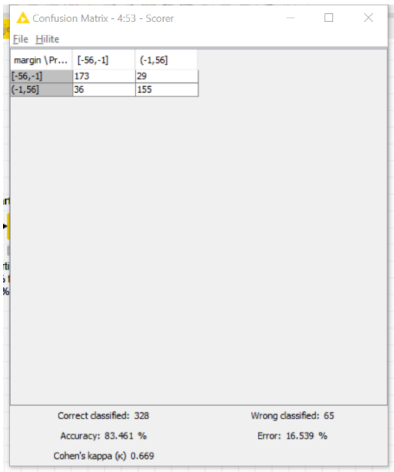
File Edit Hilite Navigation View

Table "default" - Rows: 1962 | Spec - Columns: 17 | Properties | Flow Variables

Row ID	OT	Totgam...	FGM	FGA	FGM3	FGA3	FTM	FTA	OR	DR	Ast	TO	Stl	Blk	PF	margin
Row0		176	32	69	11	29	17	26	14	30	17	12	5	3	22	(-1,56]
Row1		131	31	66	7	23	11	14	11	36	22	16	10	7	8	(-1,56]
Row2		155	31	59	6	14	16	22	10	27	18	9	7	4	19	(-1,56]
Row3		152	29	53	3	7	18	25	11	20	15	18	13	1	19	(-1,56]
Row4		150	27	64	7	20	15	23	18	20	17	13	8	2	14	(-1,56]
Row5		111	17	52	4	14	20	27	12	29	8	14	3	8	16	(-1,56]
Row6		124	19	54	4	13	25	31	13	27	4	16	10	8	23	(-1,56]
Row7		143	20	47	6	14	28	37	8	28	12	12	2	2	15	(-1,56]
Row8		125	24	56	5	14	12	14	15	23	15	14	11	4	14	(-1,56]
Row9		125	28	51	2	6	6	11	7	20	13	11	8	4	17	(-1,56]
Row10		140	22	51	9	16	19	23	11	20	14	10	4	3	23	(-1,56]
Row11		143	28	52	5	13	11	18	9	32	7	23	4	6	19	(-1,56]
Row12		139	23	54	3	13	21	25	11	33	7	20	6	6	19	(-1,56]
Row13		125	24	52	10	18	13	24	11	24	16	14	8	3	12	(-1,56]
Row14		146	29	64	8	24	11	15	12	29	16	14	4	8	24	(-1,56]
Row15		155	33	57	8	12	10	13	8	26	18	12	6	1	11	(-1,56]
Row16		155	31	58	6	16	13	22	10	24	16	9	7	2	16	(-1,56]
Row17		128	25	57	7	18	8	13	14	24	11	15	4	5	16	(-1,56]
Row18		93	18	47	5	18	6	8	7	19	5	8	3	0	15	(-1,56]
Row19		140	32	61	13	28	8	14	9	31	24	10	6	3	15	(-1,56]
Row20		129	19	49	7	18	22	26	13	22	15	8	1	2	17	(-1,56]
Row21		159	40	65	10	20	5	11	10	25	22	13	7	4	17	(-1,56]
Row22		150	35	70	8	24	8	13	15	29	18	12	12	4	22	(-1,56]
Row23		148	27	54	10	19	11	15	8	27	17	13	3	3	20	(-1,56]
Row24		143	26	56	7	13	20	25	10	28	16	11	5	1	18	(-1,56]
Row25		140	27	50	7	20	16	28	10	15	15	11	12	0	17	(-1,56]
Row26		148	33	59	9	16	12	21	10	23	22	9	14	6	19	(-1,56]
Row27		136	27	52	9	16	17	19	8	25	7	19	9	1	18	(-1,56]
Row28		141	30	52	4	11	12	15	11	28	16	18	5	9	17	(-1,56]
Row29		143	31	62	6	16	14	20	19	28	15	14	6	2	16	(-1,56]
Row30		118	18	61	9	23	15	18	13	23	8	15	7	4	22	(-1,56]
Row31		149	29	69	6	23	12	20	19	29	13	19	8	9	19	(-1,56]
Row32		130	26	64	6	20	13	18	17	30	19	10	5	4	11	(-1,56]
Row33		191	34	74	7	24	21	29	18	29	18	9	7	4	20	(-1,56]
Row34		159	31	68	6	10	17	21	17	26	12	9	4	1	13	(-1,56]

Then, I select 80% as training data and 20% as testing data. I am training the data by Naïve Bayes learner node and predict by predictor. Then I scored them by classification accuracy, confusion matrix, F-value, and summaries. They are the same steps as win/loss.

Confusion matrix



Better view for confusion matrix, class statistics, overall statistics.

Scorer View

Confusion Matrix

Rows Number : 393	(-1,56] (Predicted)	[-56,-1] (Predicted)	
(-1,56] (Actual)	155	36	81.15%
[-56,-1] (Actual)	29	173	85.64%
	84.24%	82.78%	

Class Statistics

Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
(-1,56]	155	29	173	36	81.15%	84.24%	81.15%	85.64%	82.67%
[-56,-1]	173	36	155	29	85.64%	82.78%	85.64%	81.15%	84.18%

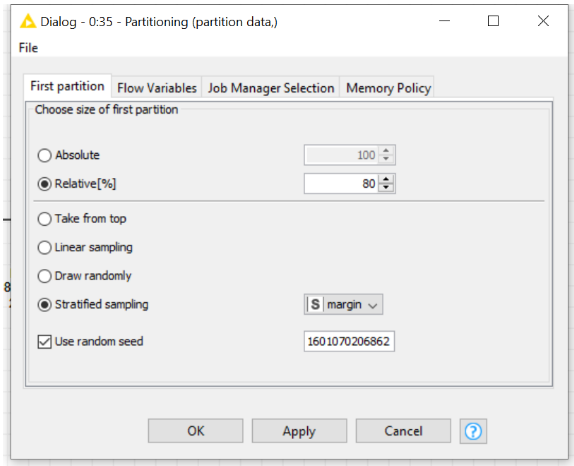
Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
83.46%	16.54%	0.669	328	65

Compare with win/loss:

By the steps above, if I run the same setting of model again, it will give me different result, since I partition data randomly. So, at the step of partition, I need to make sure these two model's data partitions to be the same. (since I am comparing them)

So, for comparison, when partition data, I choose 'Stratified sampling' instead of draw randomly, and use the same random seed, to make sure these two model's data partitions are the same.



Now, other things are equal, but difference with target column, do the following comparison:

Result for margin model (with grouping by width):

Scorer View

Confusion Matrix

Rows Number : 393	(0,56] (Predicted)	[-56,0] (Predicted)	
(0,56] (Actual)	145	51	73.98%
[-56,0] (Actual)	21	176	89.34%
	87.35%	77.53%	

Class Statistics

Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
(0,56]	145	21	176	51	73.98%	87.35%	73.98%	89.34%	80.11%
[-56,0]	176	51	145	21	89.34%	77.53%	89.34%	73.98%	83.02%

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
81.68%	18.32%	0.633	321	72

Result for margin model (with grouping by frequency):

Scorer View

Confusion Matrix

Rows Number : 393	(-1,56] (Predicted)	[-56,-1] (Predicted)	
(-1,56] (Actual)	145	51	73.98%
[-56,-1] (Actual)	21	176	89.34%
	87.35%	77.53%	

Class Statistics

Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
(-1,56]	145	21	176	51	73.98%	87.35%	73.98%	89.34%	80.11%
[-56,-1]	176	51	145	21	89.34%	77.53%	89.34%	73.98%	83.02%

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
81.68%	18.32%	0.633	321	72

We can see binning is a little bit different, but the result is the same. Also, same result when binning by percentiles (0,0.5,1).

Result for win/loss model

Scorer View

Confusion Matrix

Rows Number : 393	0 (Predicted)	1 (Predicted)	
0 (Actual)	176	21	89.34%
1 (Actual)	51	145	73.98%
	77.53%	87.35%	

Class Statistics

Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
0	176	51	145	21	89.34%	77.53%	89.34%	73.98%	83.02%
1	145	21	176	51	73.98%	87.35%	73.98%	89.34%	80.11%

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
81.68%	18.32%	0.633	321	72

So, we can see the prediction is the same, between directly using model by dataset with win/loss or binning continuous target columns into two groups by dataset with margin.

Though from the steps above, they did the same things, but binning continuous target can tell us more information if we divide it within more groups. For example, we can divide margin into 4 groups, by descending sequence, with win absolutely, just win several points, just loss several points, loss absolutely, according to 4 intervals.

Also, sometimes the accuracy between binning continuous target and categorical variable is different (other things are equal), since this example is evenly distributed, 50% percent of them are loss, 50% of them are win. When it is win, data is above 0. So, in this example, it should be like this (the predictions are the same). In the real life, most datasets are not like this.