Introduction to AI : Assignment 2

Classifying by Income Group

Seher Khan 05304

Contents

Introduction 2
Data Description 2
Data Cleaning 3
Applying and Testing Models 5
Decision Tree 5
Naïve Bayes 6
Artificial Neural Networks 6
Logistic Regression 7
Results 9
Decision Tree 9
Naïve Bayes 9
Artificial Neural Networks 9
Logistic Regression 10
Evaluation Metrics 10
Accuracy 10
Cohen's Kappa 11
Precision 13
Sensitivity and Specificity 14
F-measure 15
Conclusion 15

Introduction

Problem Statement: Predict whether income exceeds $50K/yr based on census data.

Classification Techniques employed:

1. Decision Tree Classifier
2. Naïve Bayes Classifier
3. Artificial Neural Networks
4. Logistic Regression

Data Source: https://archive.ics.uci.edu/ml/datasets/Adult

Data Description

The data was provided in two files: adult.data and adult.test, in a 2:1 ratio. The former was to be used for model training and the latter was to be used for model testing.

	Number of Entries	Percentage of Total
Training	32561	67%
Test	16281	33%
Total	48842	100%

	<=50K	>50K	Total
Training	24720	7841	32561
Test	12435	3846	16281
Total	37155	11687	48842

The data had 15 columns (Col0 to Col14), with Col14 being the class attribute of income group.

adult.data:

Row ID	Col0	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8	Col9	Col10	Col11	Col12	Col13	Col14
Row0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
Row1	50	Self-emp-no...	83311	Bachelors	13	Married-civ-...	Exec-manag...	Husband	White	Male	0	0	13	United-States	<=50K
Row2	38	Private	215646	HS-grad	9	Divorced	Handlers-cle...	Not-in-family	White	Male	0	0	40	United-States	<=50K
Row3	53	Private	234721	11th	7	Married-civ-...	Handlers-cle...	Husband	Black	Male	0	0	40	United-States	<=50K
Row4	28	Private	338409	Bachelors	13	Married-civ-...	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
Row5	37	Private	284582	Masters	14	Married-civ-...	Exec-manag...	Wife	White	Female	0	0	40	United-States	<=50K
Row6	49	Private	160187	9th	5	Married-spo...	Other-service	Not-in-family	Black	Female	0	0	16	Jamaica	<=50K
Row7	52	Self-emp-no...	209642	HS-grad	9	Married-civ-...	Exec-manag...	Husband	White	Male	0	0	45	United-States	>50K
Row8	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Female	14084	0	50	United-States	>50K
Row9	42	Private	159449	Bachelors	13	Married-civ-...	Exec-manag...	Husband	White	Male	5178	0	40	United-States	>50K
Row10	37	Private	280464	Some-college	10	Married-civ-...	Exec-manag...	Husband	Black	Male	0	0	80	United-States	>50K
Row11	30	State-gov	141297	Bachelors	13	Married-civ-...	Prof-specialty	Husband	Asian-Pac-Is...	Male	0	0	40	India	>50K
Row12	23	Private	122272	Bachelors	13	Never-married	Adm-clerical	Own-child	White	Female	0	0	30	United-States	<=50K

adult.test:

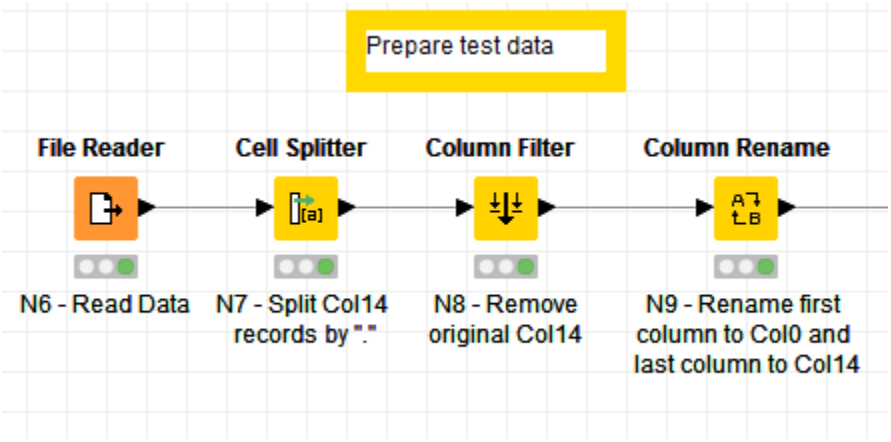
Row ID	|1x3 Cr...	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8	Col9	Col10	Col11	Col12	Col13	Col14
Row0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K.
Row1	38	Private	89814	HS-grad	9	Married-civ-...	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K.
Row2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-...	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K.
Row3	44	Private	160323	Some-college	10	Married-civ-...	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K.
Row4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States	<=50K.
Row5	34	Private	198693	10th	6	Never-married	Other-service	Not-in-family	White	Male	0	0	30	United-States	<=50K.
Row6	29	?	227026	HS-grad	9	Never-married	?	Unmarried	Black	Male	0	0	40	United-States	<=50K.
Row7	63	Self-emp-no...	104626	Prof-school	15	Married-civ-...	Prof-specialty	Husband	White	Male	3103	0	32	United-States	>50K.
Row8	24	Private	369667	Some-college	10	Never-married	Other-service	Unmarried	White	Female	0	0	40	United-States	<=50K.
Row9	55	Private	104996	7th-8th	4	Married-civ-...	Craft-repair	Husband	White	Male	0	0	10	United-States	<=50K.
Row10	65	Private	184454	HS-grad	9	Married-civ-...	Machine-op-inspct	Husband	White	Male	6418	0	40	United-States	>50K.
Row11	36	Federal-gov	212465	Bachelors	13	Married-civ-...	Adm-clerical	Husband	White	Male	0	0	40	United-States	<=50K.

Data Cleaning

Step 1

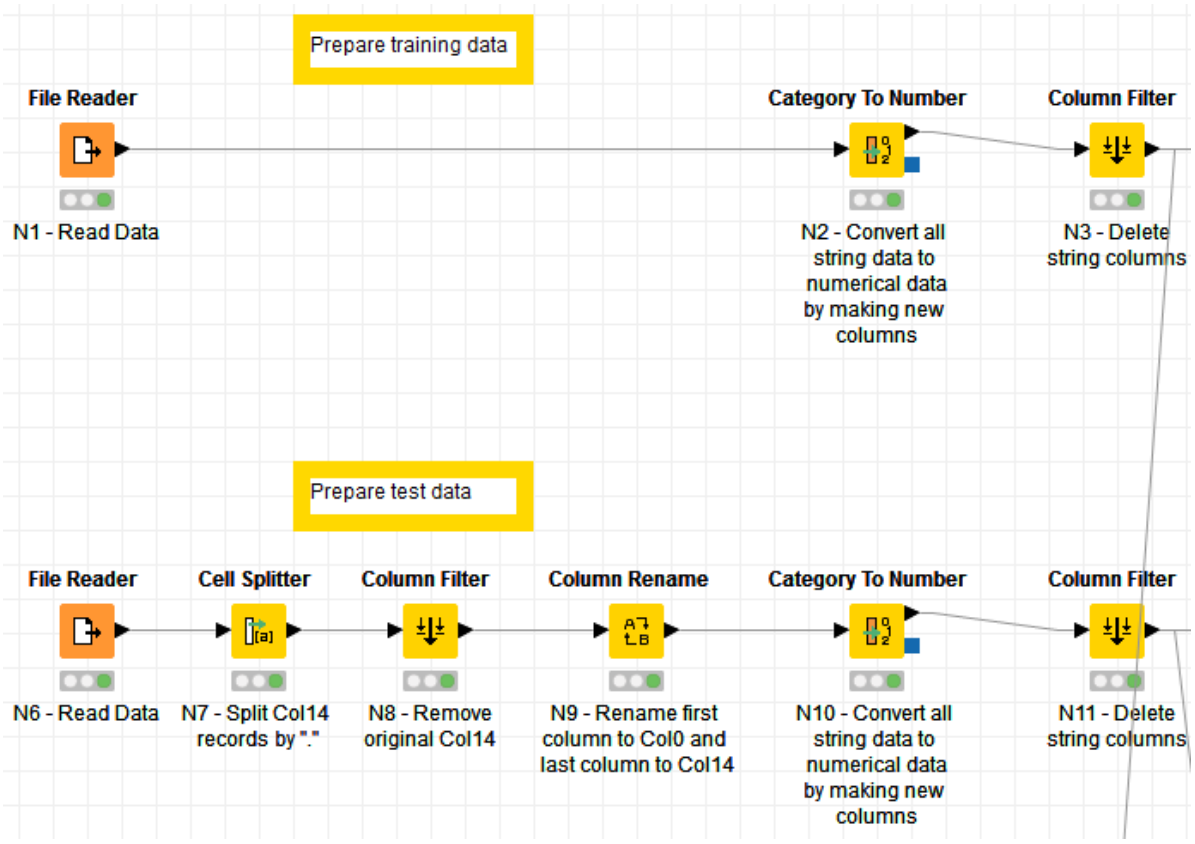
The names of the first column were different in both files. Moreover, in adult.test, entries of Col14 had a period at their end. These two problems meant that if a model was applied on adult.data, it could not be used with adult.test directly. Hence I tackled these issues in adult.test by:

1. Splitting Col14 at the period (N7), removing the old column (N8), and renaming the new column to Col14 (N9)
2. Renaming the first column to Col0 (N9)



Step 2

A precondition of applying Artificial Neural Networks and Logistic Regression is that, apart from the class attribute, all series in the data are normally distributed. However, string type data cannot be normalized and must first be converted to numerical data. Hence during my preprocessing process, in both the training and test data, I converted the columns with string type data (Col1, Col2, Col3, Col5, Col6, Col7, Col8, Col 9 and Col13) to numerical data (N2 and N10) and deleted the original string type columns (N3 and N11).



Training data:

Row ID	Col0	Col1 (t...	Col2	Col3 (t...	Col4	Col5 (t...	Col6 (t...	Col7 (t...	Col8 (t...	Col9 (t...	Col10	Col11	Col12	Col13 (...	Col14
Row0	39	0	77516	0	13	0	0	0	0	0	2174	0	40	0	<=50K
Row1	50	1	83311	0	13	1	1	1	0	0	0	0	13	0	<=50K
Row2	38	2	215646	1	9	2	2	0	0	0	0	0	40	0	<=50K
Row3	53	2	234721	2	7	1	2	1	1	0	0	0	40	0	<=50K
Row4	28	2	338409	0	13	1	3	2	1	1	0	0	40	1	<=50K
Row5	37	2	284582	3	14	1	1	2	0	1	0	0	40	0	<=50K
Row6	49	2	160187	4	5	3	4	0	1	1	0	0	16	2	<=50K
Row7	52	1	209642	1	9	1	1	1	0	0	0	0	45	0	>50K

Testing data:

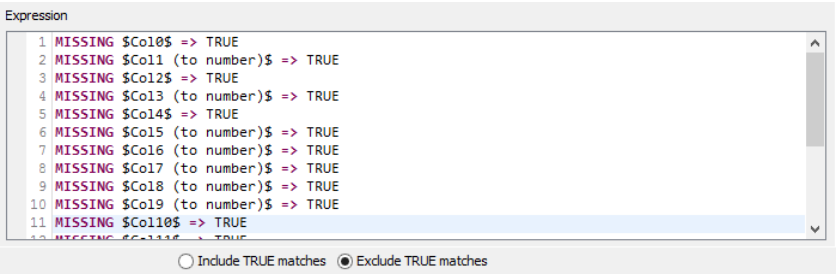
Row ID	Col0	Col1 (t...	Col2	Col3 (t...	Col4	Col5 (t...	Col6 (t...	Col7 (t...	Col8 (t...	Col9 (t...	Col10	Col11	Col12	Col13 (...	Col14
Row0	25	0	226802	0	7	0	0	0	0	0	0	0	40	0	<=50K
Row1	38	0	89814	1	9	1	1	1	1	0	0	0	50	0	<=50K
Row2	28	1	336951	2	12	1	2	1	1	0	0	0	40	0	>50K
Row3	44	0	160323	3	10	1	0	1	0	0	7688	0	40	0	>50K
Row4	18	?	103497	3	10	0	?	0	1	1	0	0	30	0	<=50K
Row5	34	0	198693	4	6	0	3	2	1	0	0	0	30	0	<=50K
Row6	29	?	227026	1	9	0	?	3	0	0	0	0	40	0	<=50K
Row7	63	2	104626	5	15	1	4	1	1	0	3103	0	32	0	>50K

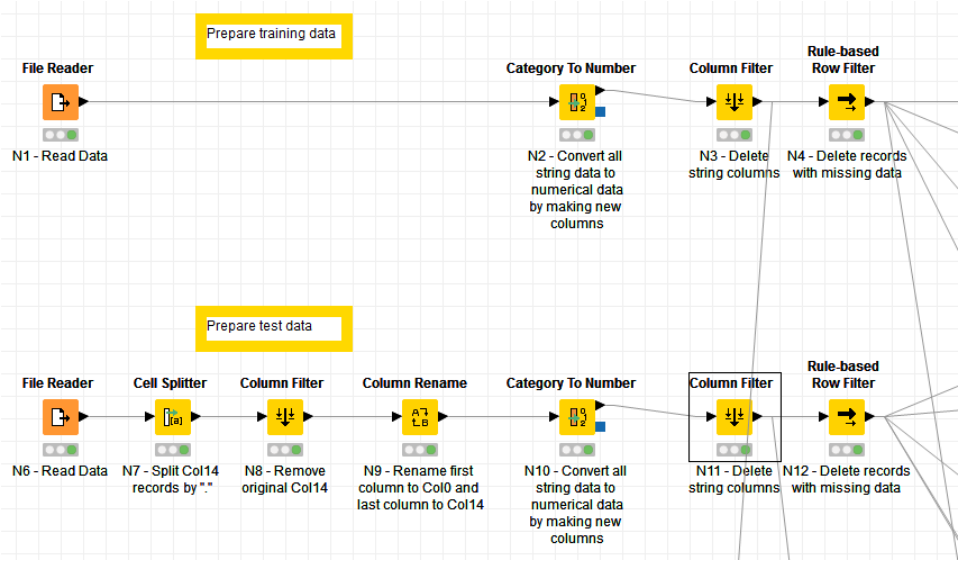
Step 3

Finally, several examples of both test and training data contained missing values of one or more columns. Rows which contain missing values cannot be used in fitting or testing a model.

	All	With missing values	Percentage of entries with missing values
Training	32561	2399	7.37%
Test	16281	1221	7.50%

I removed records with missing data using N4 and N12 by configuring each as follows:



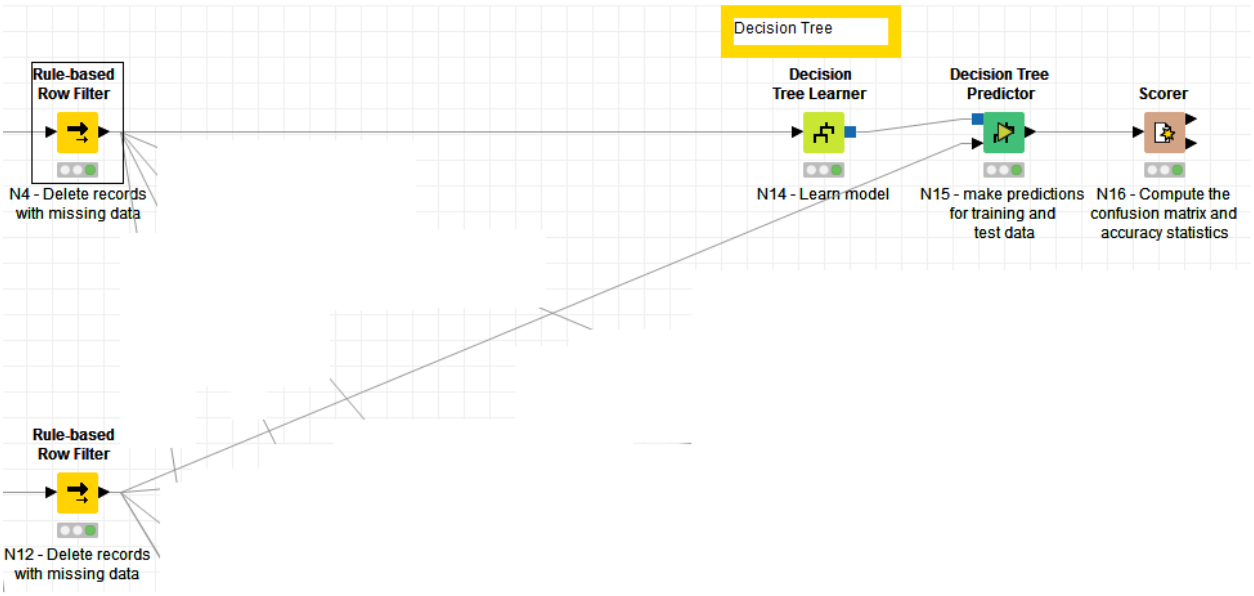


Applying and Testing Models

Each model was applied using the respective learner nodes. The models were then used to make predictions for testing data using the respective predictor nodes. Finally, scorer nodes were used to compute the confusion matrix and accuracy statistics.

Decision Tree

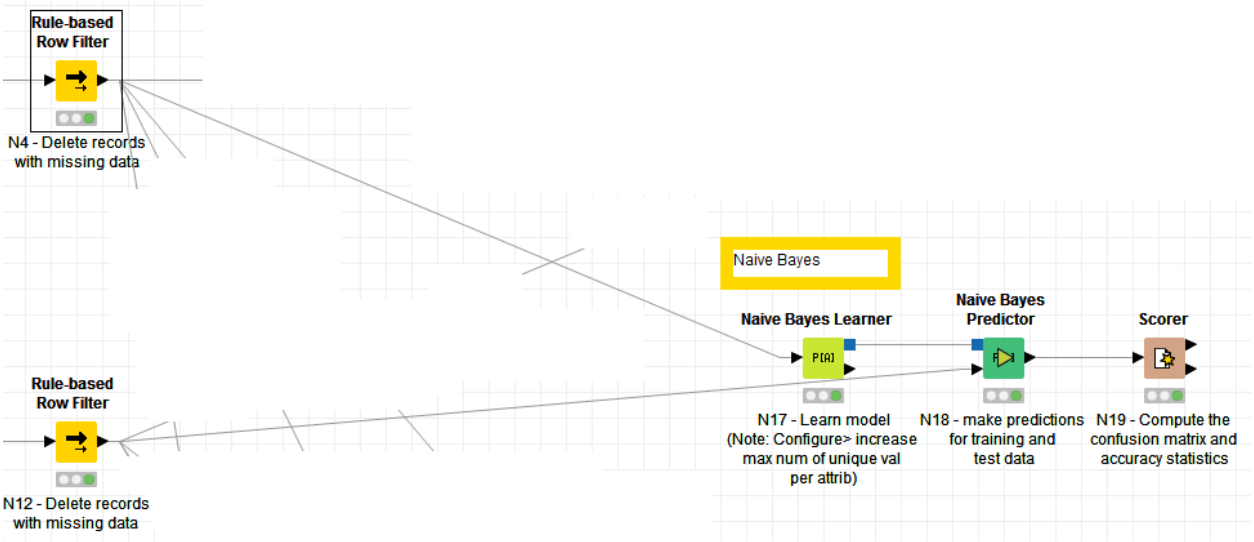
The model is learnt by calculating a quality measure for each attribute (apart from the class attribute). The attribute which turns out to be most homogenous is placed at the root of the tree. Branches which represent the possible values of this attribute are made and the process is repeated to select the root of each subtree. In my model I used the Gini Index to measure quality.



Naïve Bayes

This model is applied by calculating the conditional probabilities of each attribute given the value of the class attribute. This technique is called "Naïve" Bayes because it assumes that,

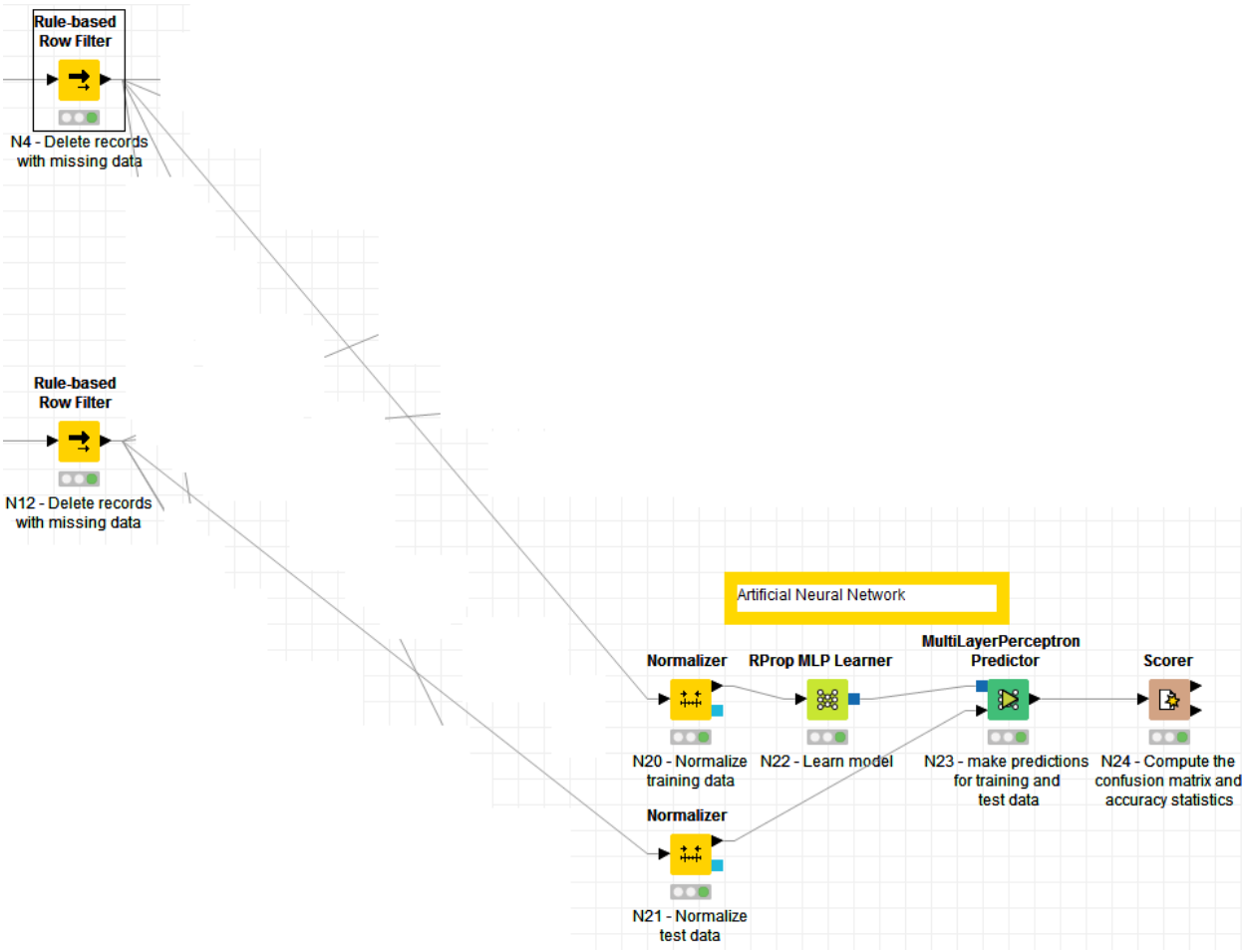
$$P(A_1 = a_1, A_2 = a_2 | C = c_0) = P(A_1 = a_1 | C = c_0) \times P(A_2 = a_2 | C = c_0)$$



Artificial Neural Networks

This technique works by passing the inputs (all attribute values other than that of the class attribute of one record) across layers of "perceptrons". As the inputs move across the branches that connect one layer of perceptrons to another, their values are changed as they are multiplied by the respective weights of these branches. Moreover as the inputs enter a perceptron, they are converted to the output of the perceptron through a "sigmoid function". This process is repeated for all records in the training data and the weights of the branches are fine tuned over time.

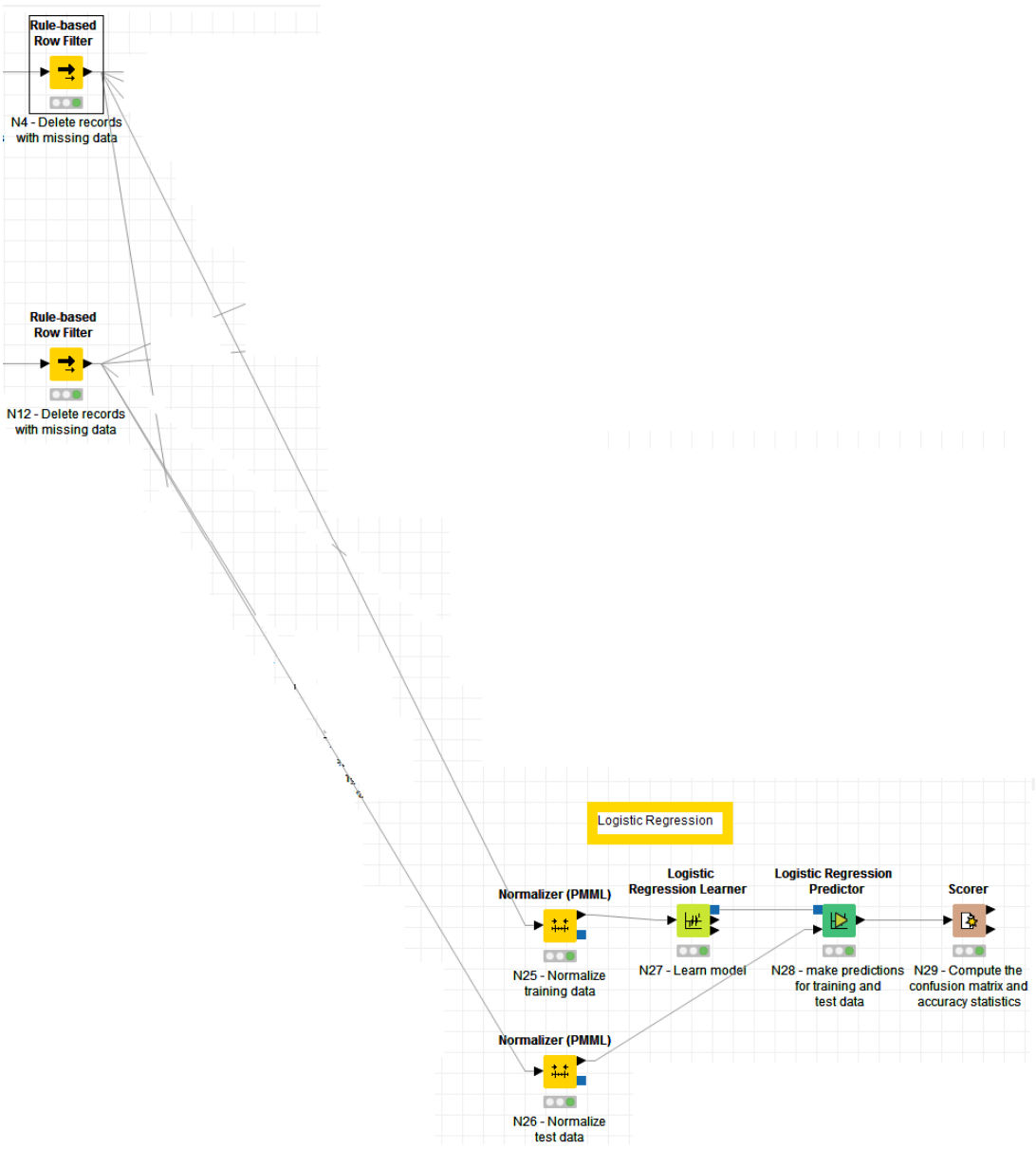
A pre-requisite of this technique is that all attributes other than the class attribute have normal data series. Hence I normalized Col0 to Col13 of both training and test data before learning the model and making predictions.



Logistic Regression

This technique works by applying a special kind of regression model which has a binary dependent variable. It works by learning coefficients using the training data.

A pre-requisite of this technique is that all attributes other than the class attribute have normal data series. Hence I normalized Col0 to Col13 of both training and test data before learning the model and making predictions.


Rule-based Row Filter
N4 - Delete records with missing data
Rule-based Row Filter
N12 - Delete records with missing data
Logistic Regression
Normalizer (PMML)
N25 - Normalize training data
Logistic Regression Learner
N27 - Learn model
Logistic Regression Predictor
N28 - make predictions for training and test data
Scorer
N29 - Compute the confusion matrix and accuracy statistics
Normalizer (PMML)
N26 - Normalize test data

Results

Decision Tree

Confusion Matrix		Predicted	
		<=50k	>50k
Actual	<=50k	9988	1372
	>50k	1617	2083

Accuracy Statistics						
	Recall	Precision	Specificity	F-measure	Accuracy	Cohen's kappa
<=50k	0.879225	0.860664	0.56297297	0.86984542	?	?
>50k	0.562973	0.602894	0.87922535	0.58225017	?	?
Overall	?	?	?	?	0.80152722	0.45229567

Naïve Bayes

Confusion Matrix		Predicted	
		<=50k	>50k
Actual	<=50k	10567	793
	>50k	2123	1577

Accuracy Statistics						
	Recall	Precision	Specificity	F-measure	Accuracy	Cohen's kappa
<=50k	0.930193662	0.832702916	0.426216216	0.878752599	?	?
>50k	0.426216216	0.665400844	0.930193662	0.519604613	?	?
Overall	?	?	?	?	0.80637450	0.405560113

Artificial Neural Networks

Confusion Matrix		Predicted	
		<=50k	>50k
Actual	<=50k	10707	653
	>50k	1998	1702

Accuracy Statistics						
	Recall	Precision	Specificity	F-measure	Accuracy	Cohen's kappa
<=50k	0.942518	0.842739	0.46	0.88984002	?	?
>50k	0.46	0.722718	0.94251761	0.56218002	?	?
Overall	?	?	?	?	0.82397078	0.45873975

Logistic Regression

Confusion Matrix		Predicted	
		<=50k	>50k
Actual	<=50k	10670	690
	>50k	2103	1597

Accuracy Statistics						
	Recall	Precision	Specificity	F-measure	Accuracy	Cohen's kappa
<=50k	0.939261	0.835356	0.43162162	0.88426636	?	?
>50k	0.431622	0.698295	0.93926056	0.53348923	?	?
Overall	?	?	?	?	0.81454183	0.42569172

Evaluation Metrics

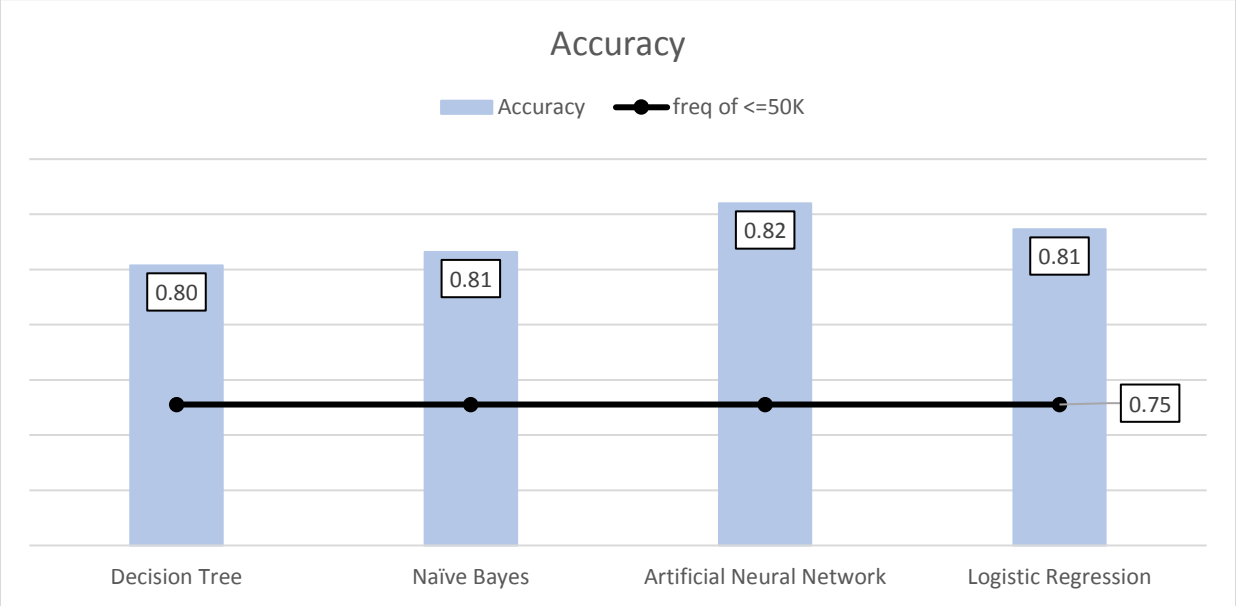
Note: Positive refers to <=50K and Negative refers to >50K

Accuracy

Accuracy is the ratio of correct predictions (true positive and true negative) to the total number of predictions,

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Judging by the accuracy, the Artificial Neural Networks model is the best.



However, only 25% of records are of the >50K income group. The data is therefore unbalanced, and all classifiers that are applied on this data will be biased toward the majority class (>50K). This makes accuracy a misleading measure of the classifiers' performances.

Training Data without missing values			
Value of Col14	**<=50K**	**>50K**	**Total**
Number of entries	**22654**	7508	30162
Percentage of entries	**75%**	25%	100%

Cohen's Kappa

Cohen's Kappa takes into account the accuracy which has resulted from random chance. It therefore gives the accuracy that is achieved due to the model only, and not by chance. Its values range from -1 to +1 where,

- -1 indicates total disagreement between the actual and predicted classes
- 0 indicates no agreement between the actual and predicted classes
- +1 indicates total agreement between the actual and predicted classes

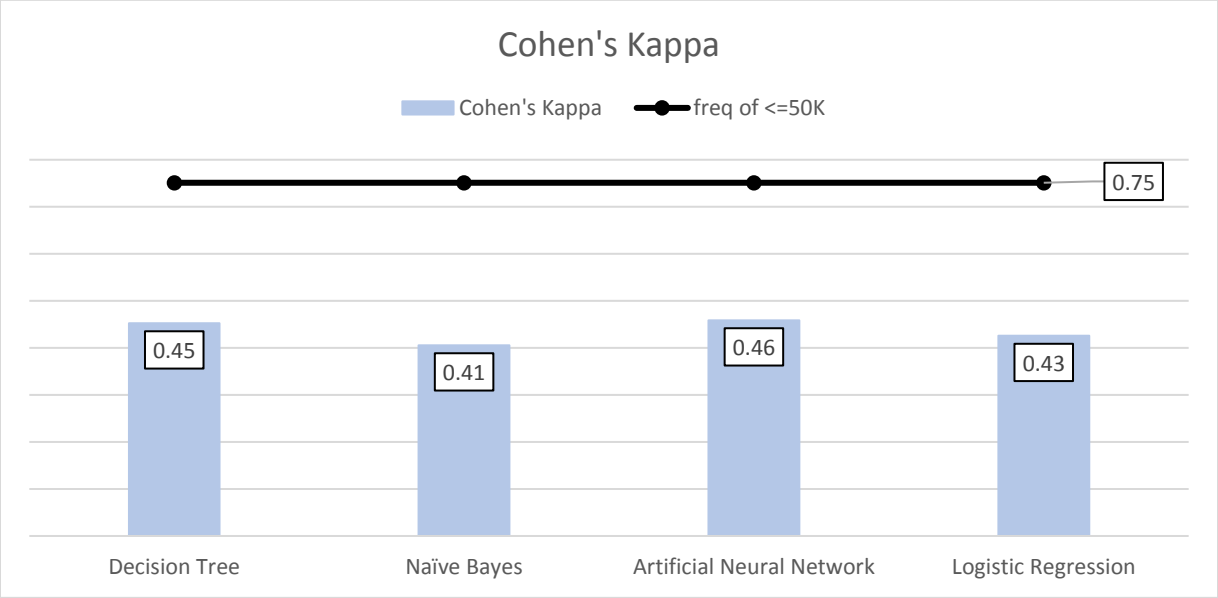
Cohen's Kappa is given by the formula,

$$Cohen's\ Kappa\ = \frac{Accuracy\ -\ Random\ Accuracy}{1\ -\ Random\ Accuracy}$$

Where,

$$Random\ accuracy\ = \frac{(TN + FP)(TN + FN) + (FN + TP)(FP + TP)}{(TP + TN + FP + FN)^2}$$

The Cohen's Kappa for all models indicated moderate agreement between predicted and actual classes for all models. The value was highest for the Artificial Neural Networks model at 0.46, closely followed by the Decision Tree model at 0.45.



Precision

Precision a measure of the "exactness" of the predictions. It is the ratio of correct true (false) predictions to the total true (false) predictions,

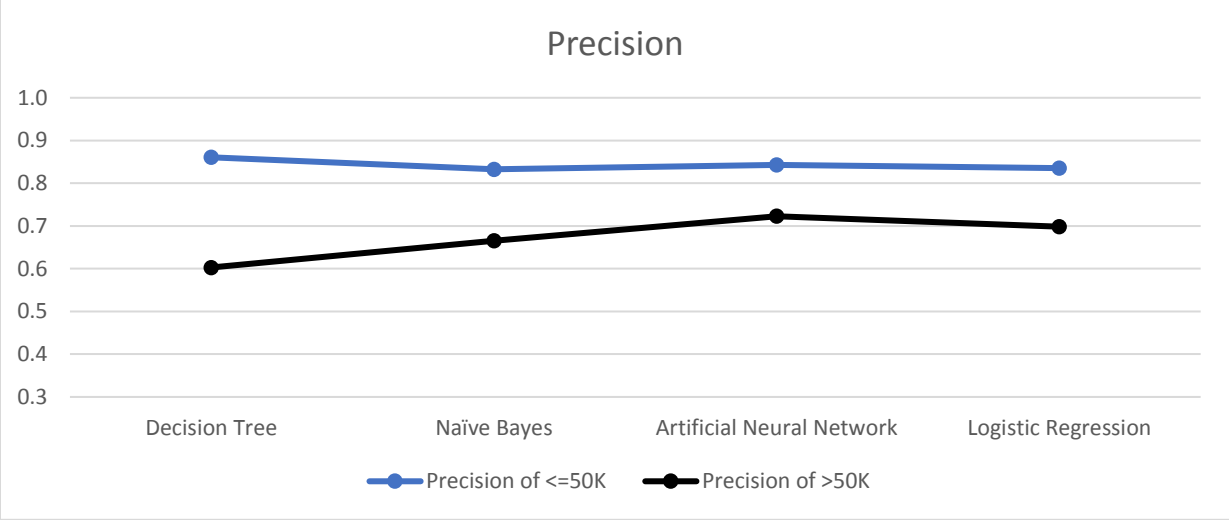
$$Precision(+) = \frac{TP}{TP + NP}$$

$$Precision(-) = \frac{TN}{TN + FN}$$

Studying the values of precision, I found that the precision of >50K was lower than >=50K in all techniques. This means makes sense considering the point made above: only 25% of the records were of >50K and therefore the classifiers were likely to be biased against this class.

It was also found that the Decision Tree model is most precise in predicting <=50K, while the Artificial Neural Networks model is most precise in predicting >50K.

Since the variation in precision of the <=50K is very low at 0.00012 compared to 0.00202 of >50K, I would choose the model with the best performance for >50K.



Sensitivity and Specificity

Sensitivity (also called Recall or the True Positive Rate) is the ratio of correct positive predictions to the actual positive values,

$$Sensitivity = \frac{TP}{TP + FN}$$

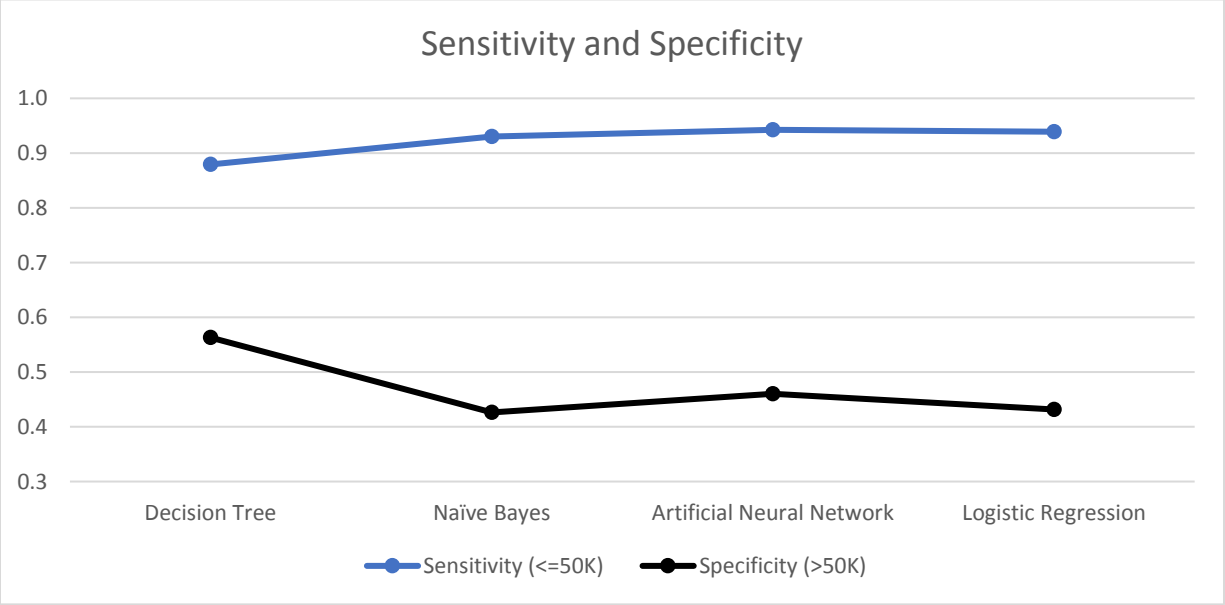
Specificity (also called the True Negative Rate) is the ratio of correct negative predictions to the actual negative values,

$$Specificity = \frac{TN}{TN + FP}$$

It was found that, for all classifiers, sensitivity was higher than specificity, i.e. performance of classifiers for >=50K was higher than that of <50K.

The Artificial Neural Networks model gave the best sensitivity (<=50k), while the Decision Tree model gave best specificity (>50K).

While the variation in sensitivity (>=50K) was lower than that of specificity (<50K), it was still substantial. Hence picking a model based on these measures is tricky. A tradeoff between the two exists. I would prefer a higher specificity since the classifiers are already biased against the >50K class.



F-measure

F-measure combines precision and sensitivity/specificity. When either of the two are zero, F-measure is zero and as they increase, F-measure also increases. It is given by the formulae,

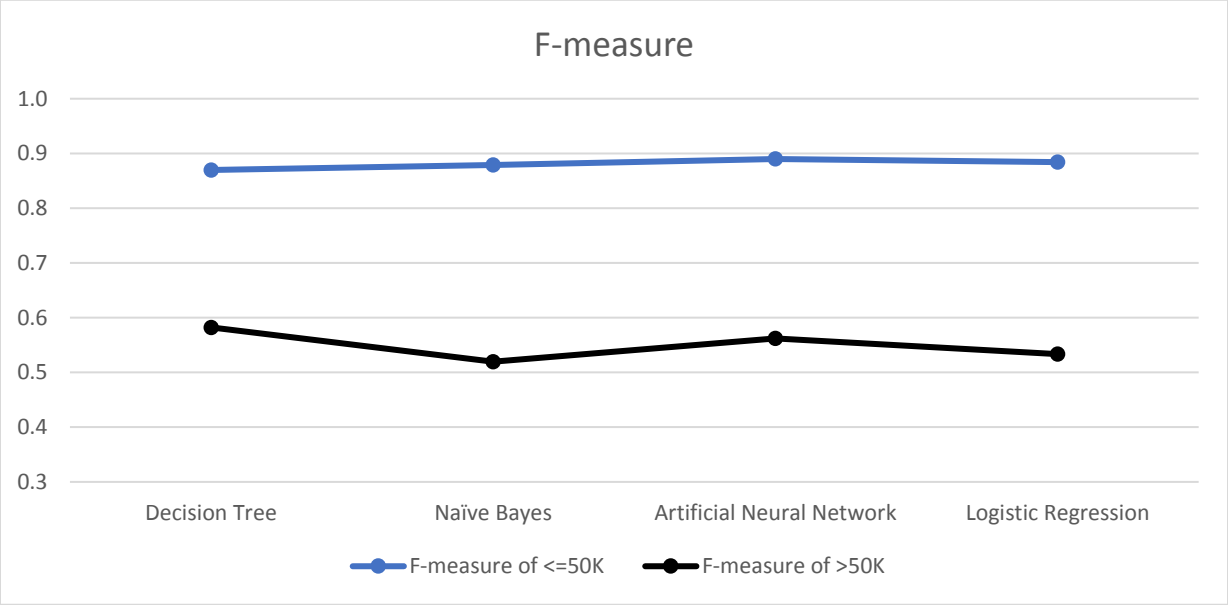
$$F - measure\ (+) = \frac{2 * Sensitivity * Precision(+)}{Sensitivity + Precision(+)}$$

$$F - measure\ (-) = \frac{2 * Specificity * Precision(-)}{Specificity + Precision(-)}$$

As expected (considering both sensitivity/specificity and precision) the F-measure of >50K was lower than <=50K in all techniques.

The Artificial Neural Networks model gave the highest F-measure for <=50K while Decision Tree model had the highest F-measure for >50K.

Since the F-measure of <=50K has a low variance of 0.00005 compared to 0.0006 of >50K, I would pick the classifier that performs best for >50K, which is the Decision Tree model.



Conclusion

After assessing the above evaluation metrics, I found that the decision tree classifier was the best classifier for the given data.