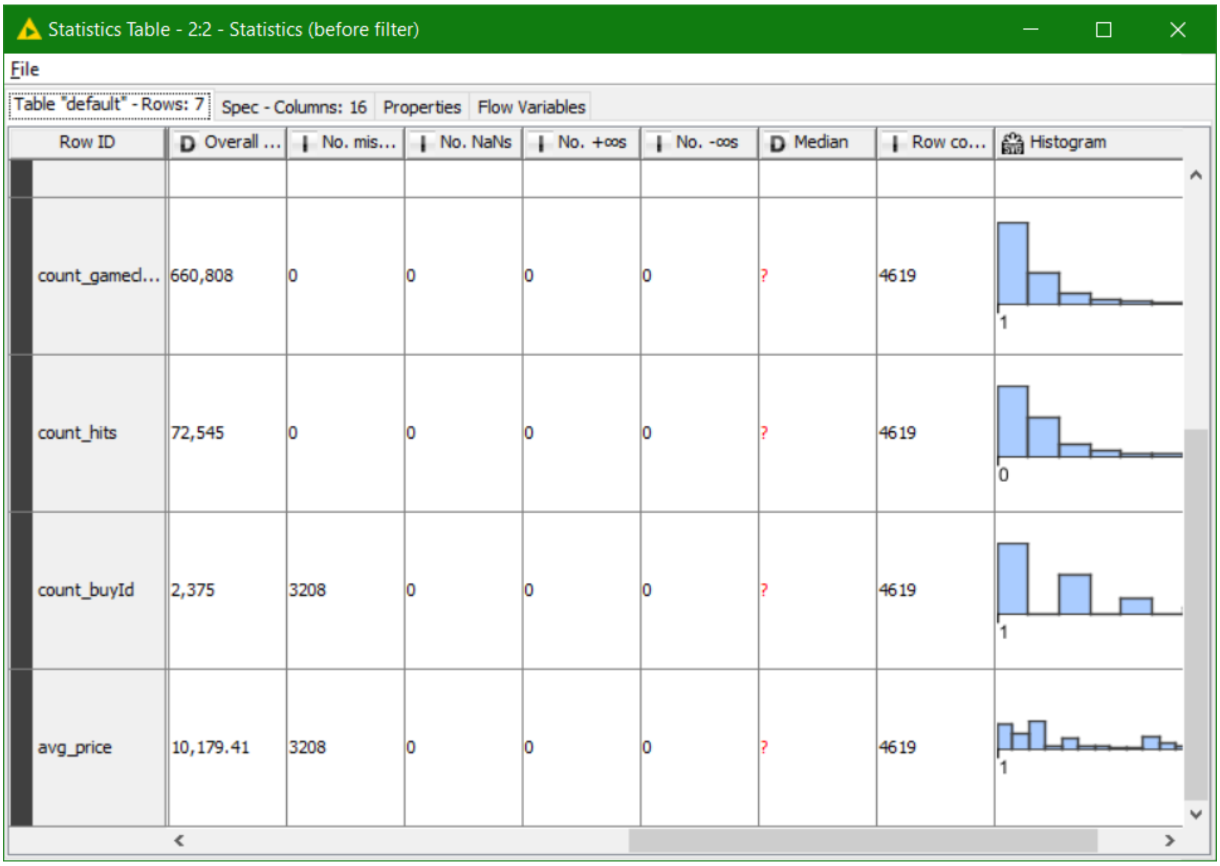
Data Preparation

Analysis of combined_data.csv

Sample Selection

Item	Amount
# of Samples	4619
# of Samples with Purchases	1411

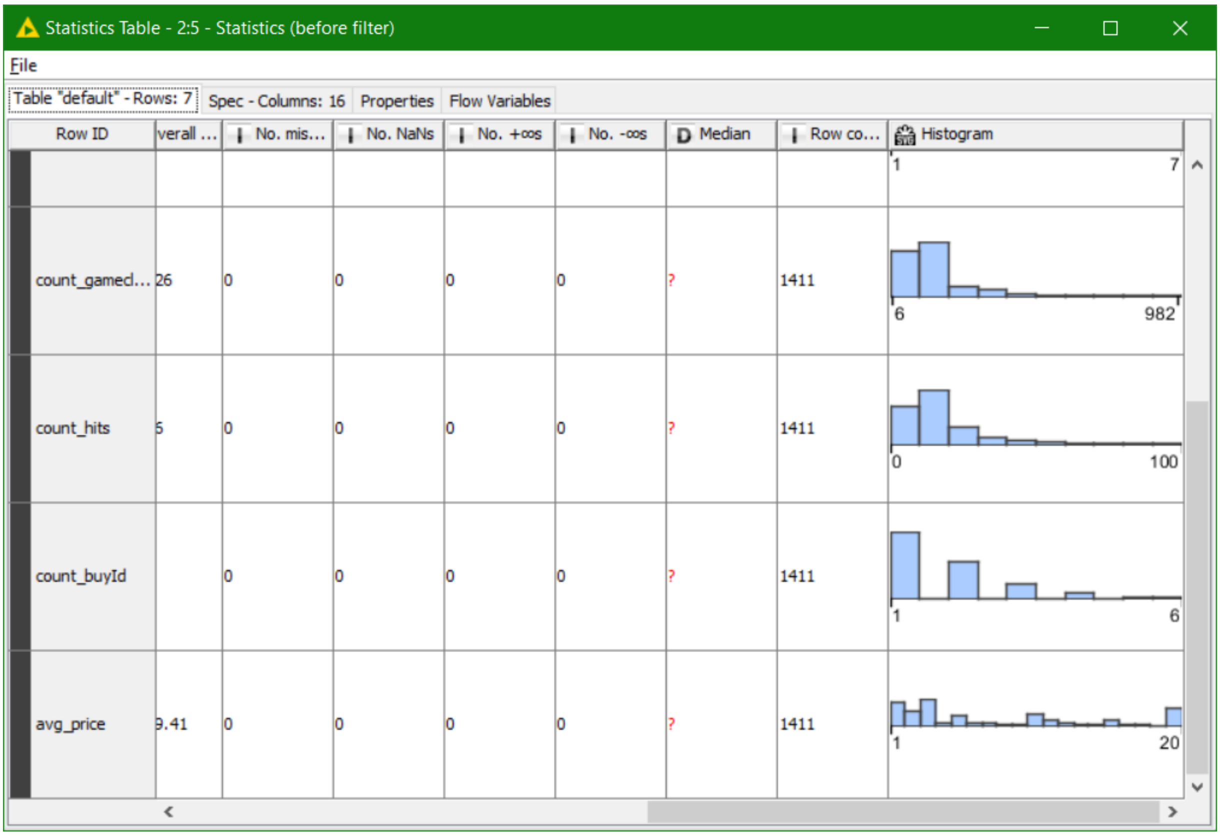


Statistics Table - 2:2 - Statistics (before filter)

File

Table "default" - Rows: 7 | Spec - Columns: 16 | Properties | Flow Variables

Row ID	Overall ...	No. mis...	No. NaNs	No. +∞s	No. -∞s	Median	Row co...	Histogram
count_gamecl...	660,808	0	0	0	0	?	4619	1
count_hits	72,545	0	0	0	0	?	4619	0
count_buyId	2,375	3208	0	0	0	?	4619	1
avg_price	10,179.41	3208	0	0	0	?	4619	1



Statistics Table - 2:5 - Statistics (before filter)

File

Table "default" - Rows: 7 | Spec - Columns: 16 | Properties | Flow Variables

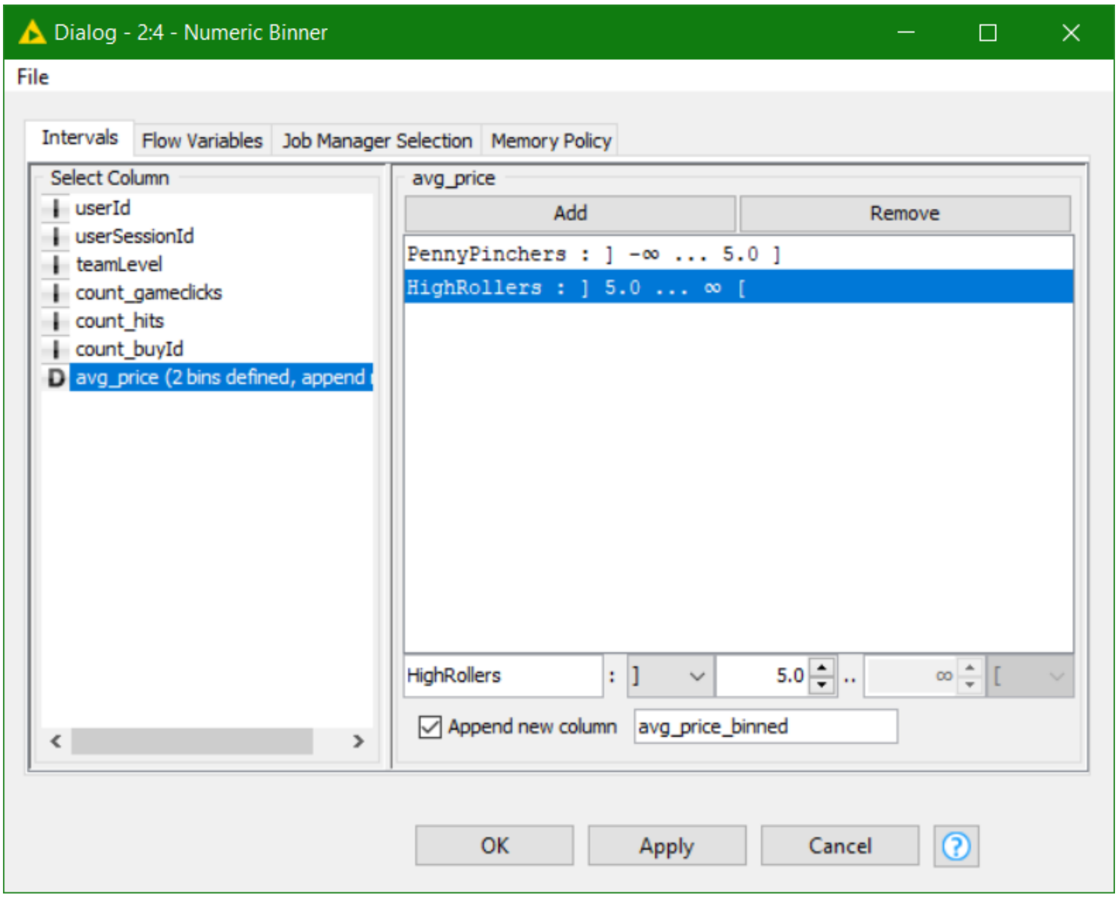
Row ID	verall ...	No. mis...	No. NaNs	No. +∞s	No. -∞s	Median	Row co...	Histogram
								1 7
count_gamecl...	26	0	0	0	0	?	1411	6 982
count_hits	5	0	0	0	0	?	1411	0 100
count_buyId		0	0	0	0	?	1411	1 6
avg_price	9.41	0	0	0	0	?	1411	1 20

v

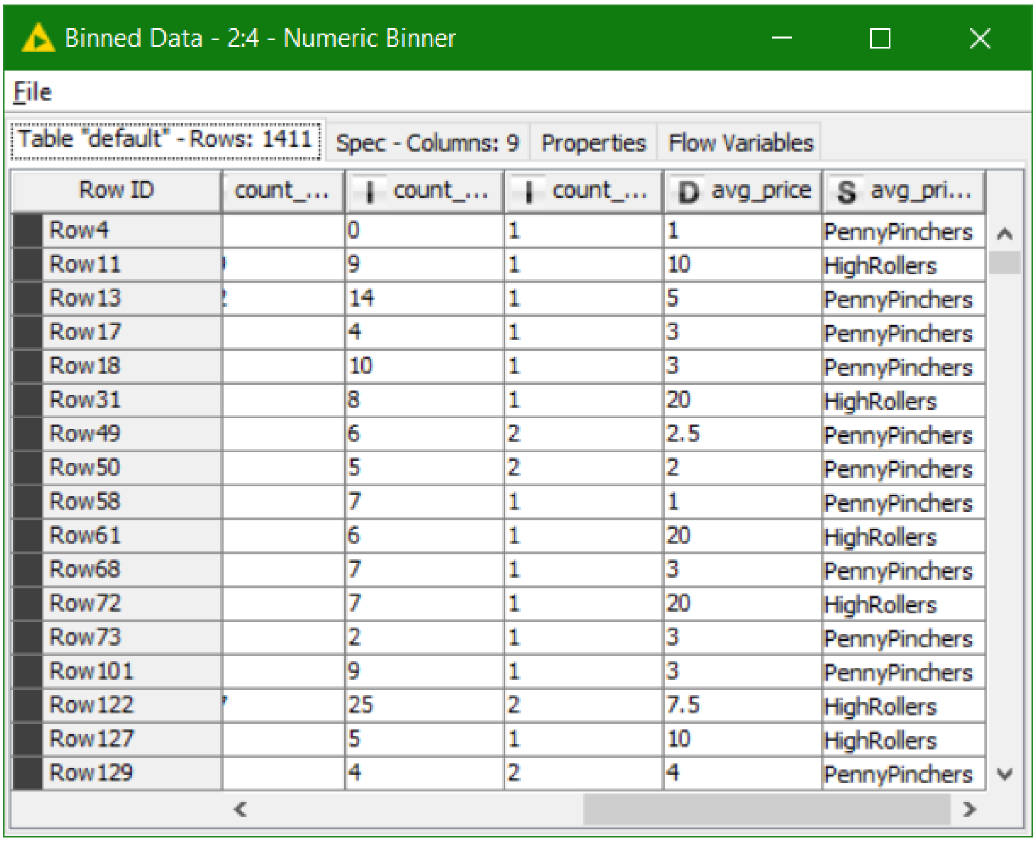
Here I have taken two snaps. First one is before applying filters (**4619 rows**) and the second one is after applying filters. Those rows which have NULL as a value are removed from the combined_data.csv file and then the statistics are observed similar to the first. After filtering we get **1411** rows and also the graph is quite similar to the previous one.

<u>Attribute Creation</u>

A new categorical attribute was created to enable analysis of players as broken into 2 categories (HighRollers and PennyPinchers). A screenshot of the attribute follows:



The numeric avg_price variable was redefined as a category variable with 2 values: PennyPinchers and HighRollers. Penny Pinchers were those who bought items costing $5.00 or less, and HighRollers are those users who bought items costing above $5.00. The design is shown above, **where "]" is inclusive**, and **"[" is exclusive**. The new category variable is named **"avg_price_binned."**



Row ID	count_...	count_...	count_...	avg_price	avg_pri...
Row4		0	1	1	PennyPinchers
Row11		9	1	10	HighRollers
Row13		14	1	5	PennyPinchers
Row17		4	1	3	PennyPinchers
Row18		10	1	3	PennyPinchers
Row31		8	1	20	HighRollers
Row49		6	2	2.5	PennyPinchers
Row50		5	2	2	PennyPinchers
Row58		7	1	1	PennyPinchers
Row61		6	1	20	HighRollers
Row68		7	1	3	PennyPinchers
Row72		7	1	20	HighRollers
Row73		2	1	3	PennyPinchers
Row101		9	1	3	PennyPinchers
Row122		25	2	7.5	HighRollers
Row127		5	1	10	HighRollers
Row129		4	2	4	PennyPinchers

The creation of this new categorical attribute was necessary because we will be using a decision tree algo. to determine the attributes responsible for **highroller** and **a pennyPincher**. It will also serve as the reference for training and subsequently scoring the Decision Tree model.

Avg-price only can not be used for classification task with a continuous-value.

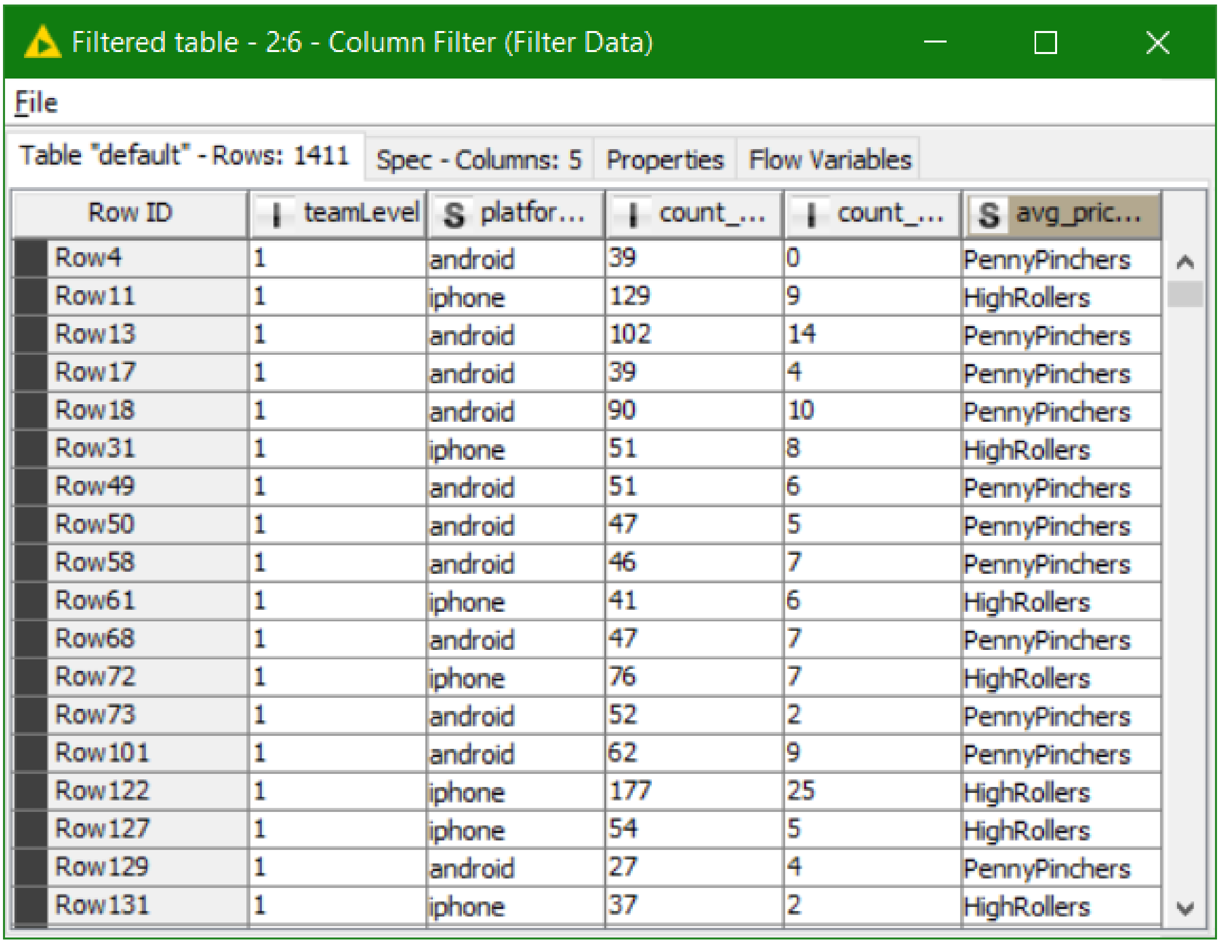
Attribute Selection

The following attributes were filtered from the dataset for the following reasons:

Attribute	**Rationale for Filtering**
userId	We are not interested in finding who exactly is a highRoller plus It doesn’t make any sense on deciding whether a player is highRoller or a PennyPencher.
userSessionId	This attribute is used to identify the session and the session does

	not contribute to a highRoller or a PennyPincher
count_buyId	The number of items purchased doesn't define a highRoller
avg_price	The numeric variable was replaced by the category variable(Binned). So it cannot be included within the training and testing dataset as the decision tree will then be giving us 100% Accuracy which defeat the original objective of analysing the behaviour and predicting the results.

After applying column filter.
Now if you observe the last column, their you can find that it has specified the PennyPinchers and the highRollers



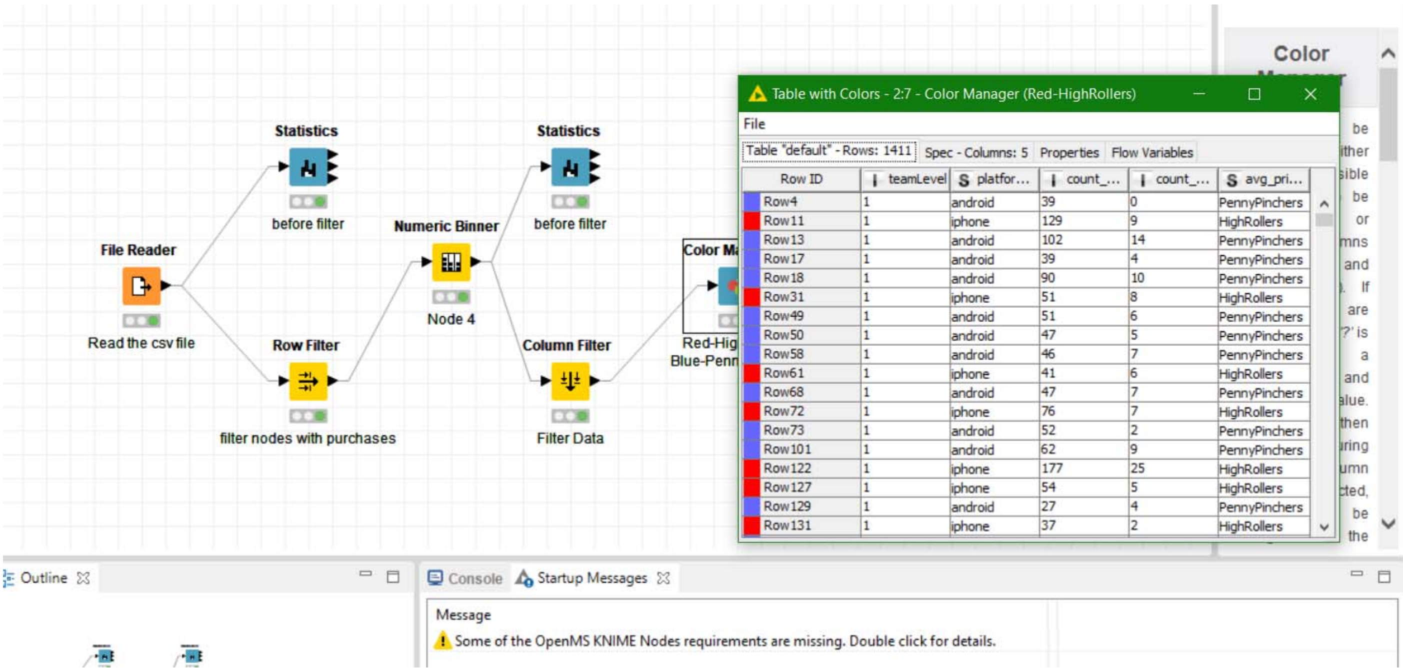
Filtered table - 2:6 - Column Filter (Filter Data)

File

Table "default" - Rows: 1411 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	teamLevel	platfor...	count_...	count_...	avg_pric...
Row4	1	android	39	0	PennyPinchers
Row11	1	iphone	129	9	HighRollers
Row13	1	android	102	14	PennyPinchers
Row17	1	android	39	4	PennyPinchers
Row18	1	android	90	10	PennyPinchers
Row31	1	iphone	51	8	HighRollers
Row49	1	android	51	6	PennyPinchers
Row50	1	android	47	5	PennyPinchers
Row58	1	android	46	7	PennyPinchers
Row61	1	iphone	41	6	HighRollers
Row68	1	android	47	7	PennyPinchers
Row72	1	iphone	76	7	HighRollers
Row73	1	android	52	2	PennyPinchers
Row101	1	android	62	9	PennyPinchers
Row122	1	iphone	177	25	HighRollers
Row127	1	iphone	54	5	HighRollers
Row129	1	android	27	4	PennyPinchers
Row131	1	iphone	37	2	HighRollers

The resulting table will be passed to the Color Manager node, where **High Rollers will be assigned a Red color** and **PennyPinchers a Blue.**



Data Partitioning and Modeling

The data was partitioned into train and test datasets.

The **Train** data set was used to create the decision tree model.

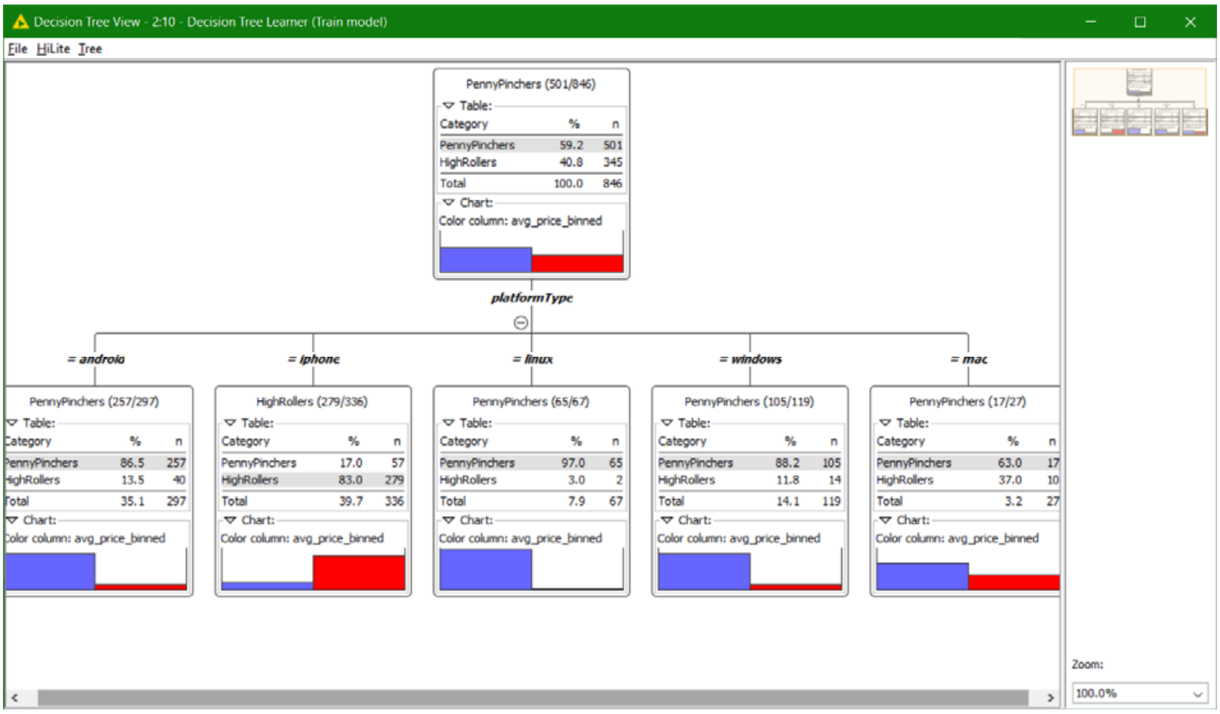
The trained model was then applied to the **Test** dataset.

This is important because the train data set consist of the records whose labels are already known and this will facilitate us to build the classification model(a decision tree) while the test data set would contain records with known labels, thus serving as an unbiased means of evaluating the performance of the trained model. Later on we will be comparing the results of the trained model(i.e accuracy, etc)

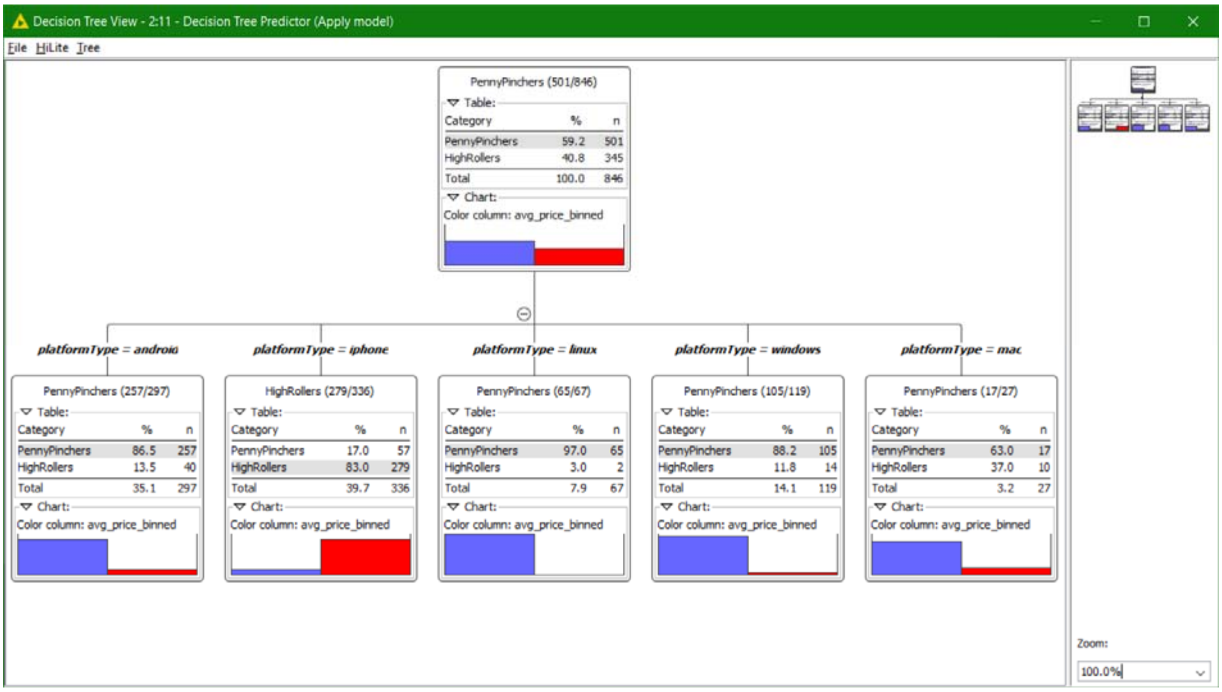
When partitioning the data using sampling, it is important to set the random seed because it ensures that I will get the same partitions every time I execute this node(i.e to replicate the same partitioned datasets for repeated executions of the training and scoring process). It is important to get reproducible results. Also, it is not set by default, so we will need to set it when we use this node.

A screenshot of the resulting decision tree can be seen below:

Zoom-in the image to view more clearly

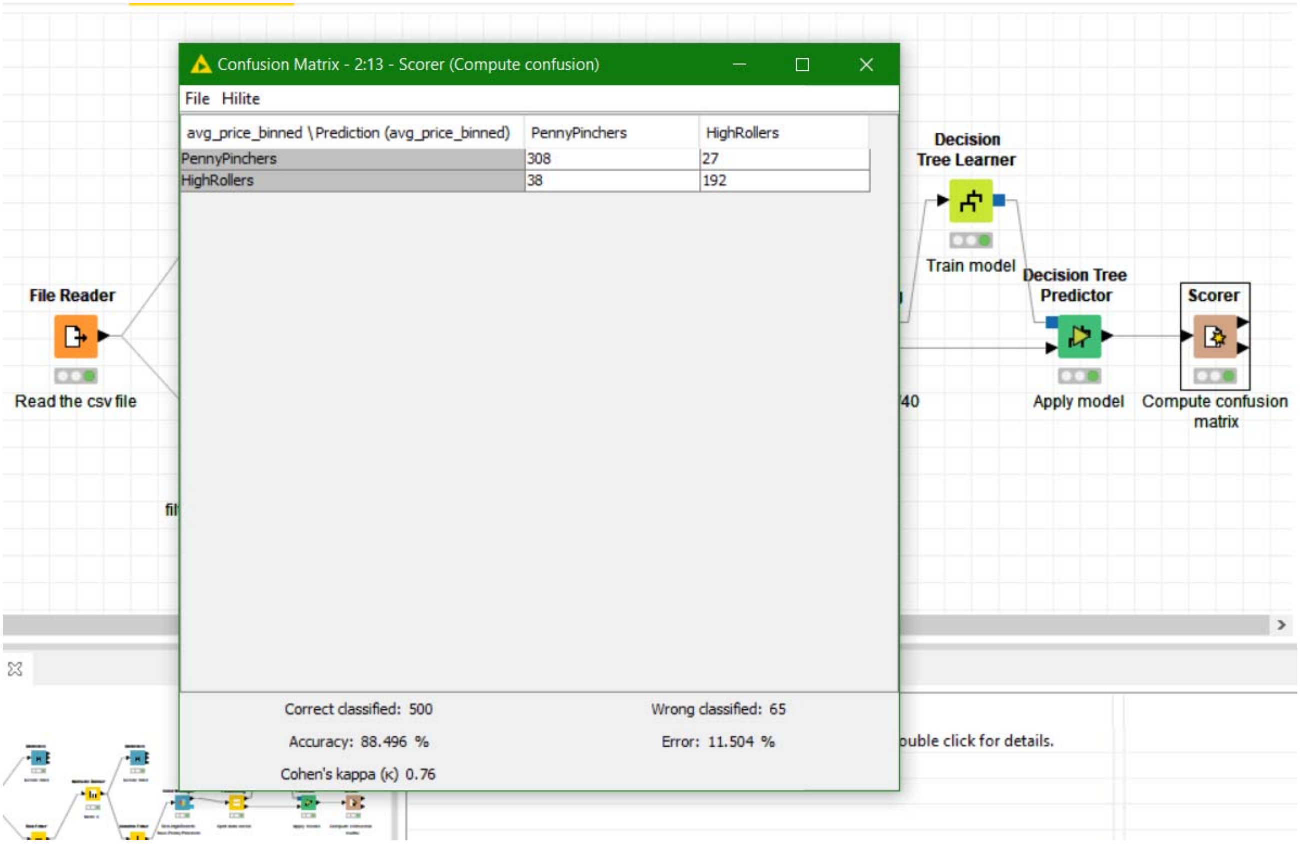


The following is the Decision Tree View of the Test Model



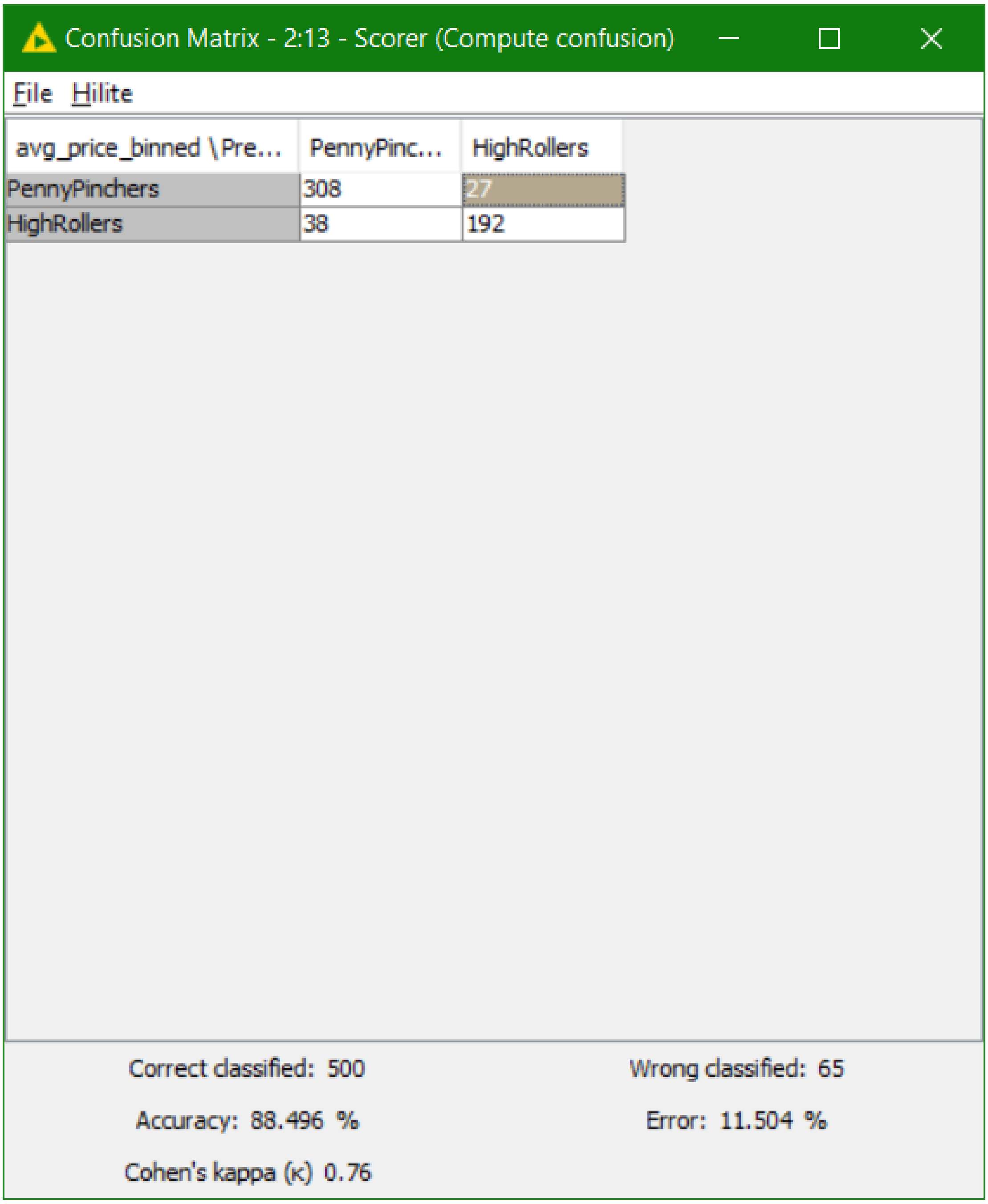
Evaluation

A screenshot of the confusion matrix can be seen below:



As seen in the screenshot above, the overall accuracy of the model is 88.496%.

Confusion Matrix –



This shows that there are total **65** Wrong classified predictions (**38+27 = 65**)

Of the 335 Penny Pinchers in the test data set >>
308 (91.9%) of them were correctly predicted as Penny Pinchers by decision tree model.
27 (8.1%) of these Penny Pinchers were incorrectly predicted as High Rollers.

Of the 230 High Rollers in the test data set >>
192 (83.5%) of them were correctly predicted as High Rollers by decision tree model.
38 (16.5%) of these High Rollers were incorrectly predicted as Penny Pinchers.

Row 150 is highlighted. Wrong Prediction (Predicted – PennyPinchers, Actual – HighRollers)

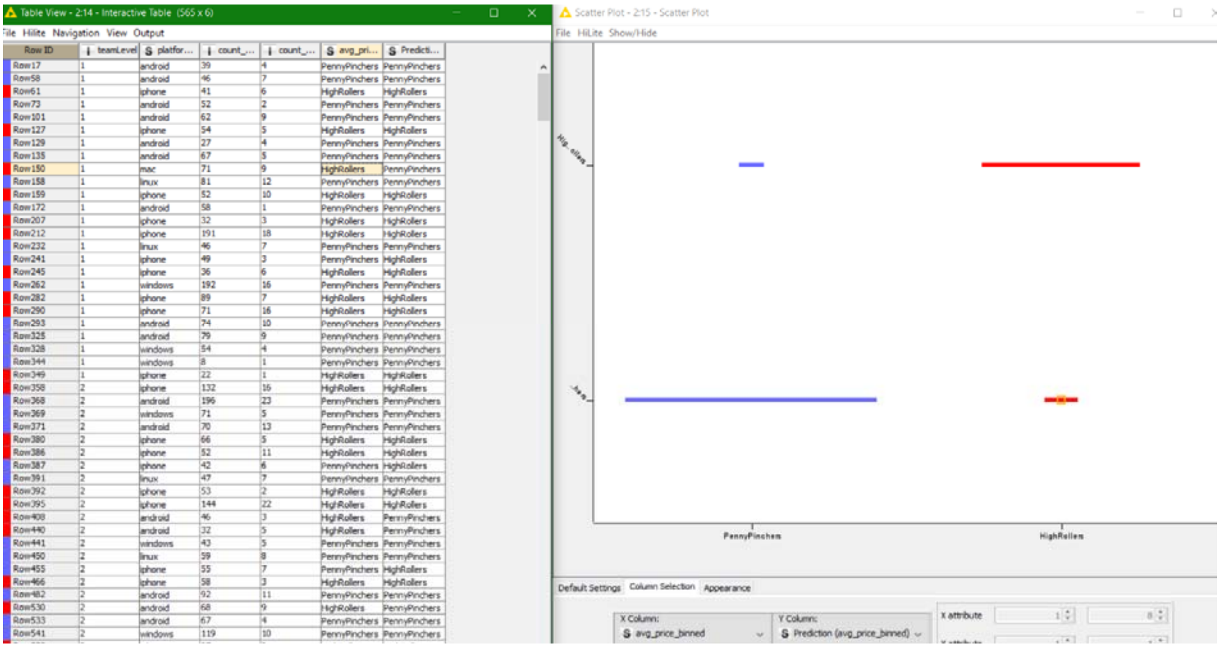


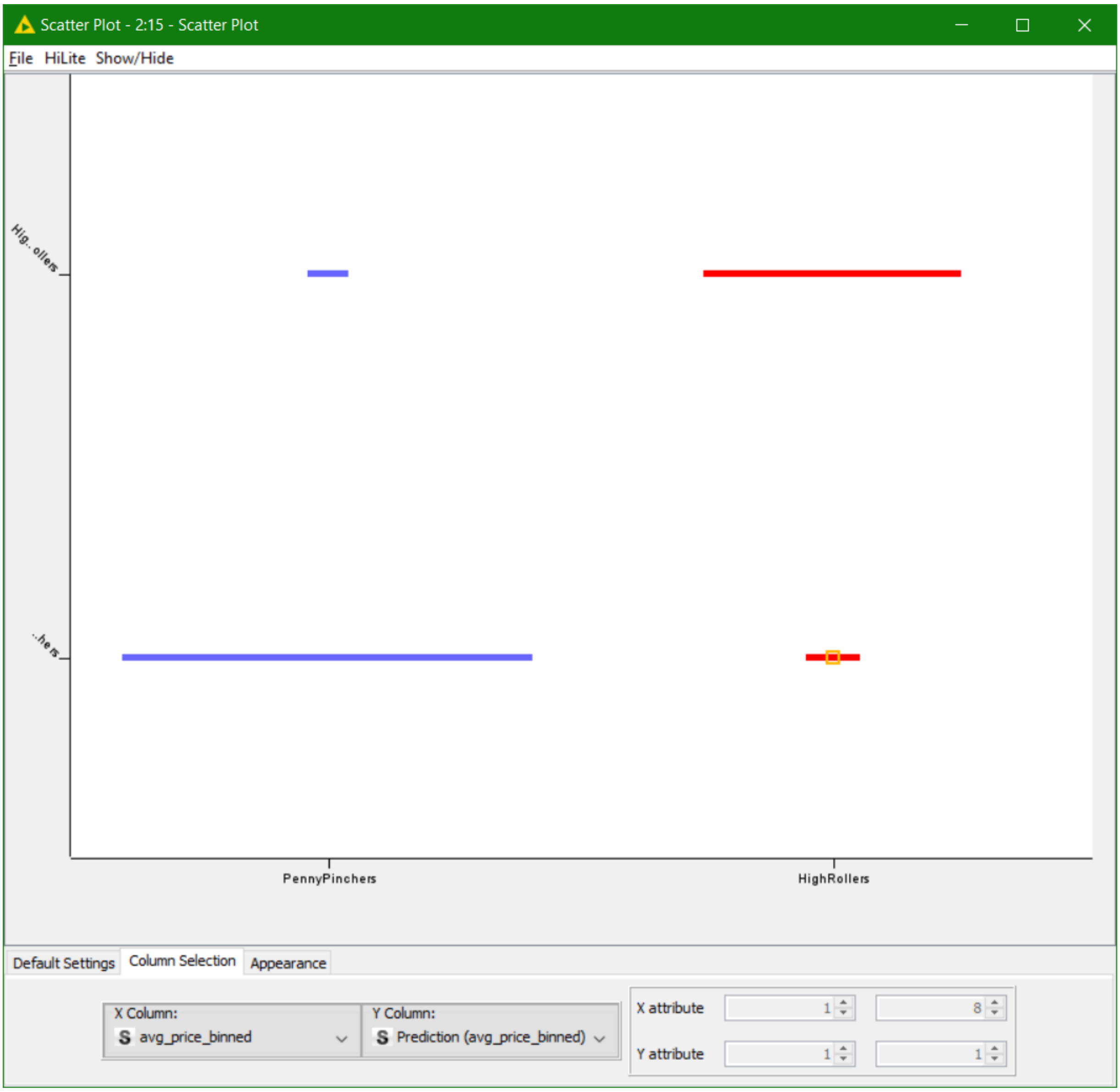
Table View - 2:14 - Interactive Table (565 x 6)

File Hilite Navigation View Output

Row ID	teamLevel	platfor...	count_...	count_...	avg_pri...	Predicti...
Row17	1	android	39	4	PennyPinchers	PennyPinchers
Row58	1	android	46	7	PennyPinchers	PennyPinchers
Row61	1	iphone	41	6	HighRollers	HighRollers
Row73	1	android	52	2	PennyPinchers	PennyPinchers
Row101	1	android	62	9	PennyPinchers	PennyPinchers
Row127	1	iphone	54	5	HighRollers	HighRollers
Row129	1	android	27	4	PennyPinchers	PennyPinchers
Row135	1	android	67	5	PennyPinchers	PennyPinchers
Row150	1	mac	71	9	HighRollers	PennyPinchers
Row158	1	linux	81	12	PennyPinchers	PennyPinchers
Row159	1	iphone	52	10	HighRollers	HighRollers
Row172	1	android	58	1	PennyPinchers	PennyPinchers
Row207	1	iphone	32	3	HighRollers	HighRollers
Row212	1	iphone	191	18	HighRollers	HighRollers
Row232	1	linux	46	7	PennyPinchers	PennyPinchers
Row241	1	iphone	49	3	PennyPinchers	HighRollers
Row245	1	iphone	36	6	HighRollers	HighRollers
Row262	1	windows	192	16	PennyPinchers	PennyPinchers
Row282	1	iphone	89	7	HighRollers	HighRollers
Row290	1	iphone	71	16	HighRollers	HighRollers
Row293	1	android	74	10	PennyPinchers	PennyPinchers
Row325	1	android	79	9	PennyPinchers	PennyPinchers
Row328	1	windows	54	4	PennyPinchers	PennyPinchers
Row344	1	windows	8	1	PennyPinchers	PennyPinchers
Row349	1	iphone	22	1	HighRollers	HighRollers
Row358	2	iphone	132	16	HighRollers	HighRollers
Row368	2	android	196	23	PennyPinchers	PennyPinchers

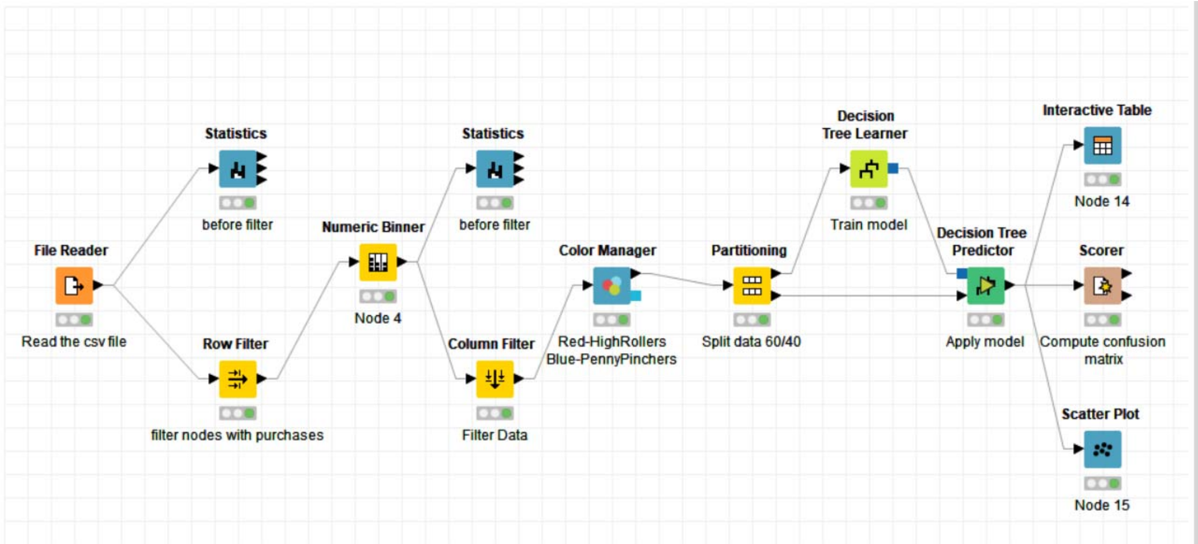
ROW 150 highlighted above is shown in the scatter plot below
The **wrong prediction are small horizontal blue line and the red line**. Both denote wrong Predictions.

The below graph means that if on both axis we have highRollers and Penny Pinchers then the prediction is good and since the success rate is 88.5% that's why we have long horizontal red and blue lines.


Scatter Plot - 2:15 - Scatter Plot
File HiLite Show/Hide
Hig...ollers
...hers
PennyPinchers
HighRollers
Default Settings
Column Selection
Appearance
X Column:
avg_price_binned
Y Column:
Prediction (avg_price_binned)
X attribute
1
8
Y attribute
1
1

Analysis Conclusions

The final KNIME workflow is shown below:



What makes a HighRoller vs. a PennyPincher?

Based on the Decision Tree that was formulated, it can be concluded that players using iOS devices has been classified as HighRollers, while Players of other platforms have been classified as PennyPinchers.

Linux being the first in terms of PennyPinchers (97% users) and just 3% highrollers
Mac being the second in terms of HighRollers (37% users).
Android on third with 13.5% highRollers and 86.5% PennyPinchers.

Specific Recommendations to Increase Revenue
1. We should be Targeting the iOS users more than other platform users. A separate budget should be kept for promotion of the game on iOS platform so that the user base will increase and so does the revenue as iOS users being the HighRollers.
2. Encouraging them(iOS players) by providing rewards to write the good review of the game, so that other platform users can be influenced by the rating and reviews of the iOS platform. Thus increasing the Downloads and the user Base
3. Providing Special Discount to the Other platform users to increase the tendency of the user to make the purchase.
4. The most popular and frequently used item should be priced so that the users will be forced to make the purchase in order to play the game with ease. This method is not ideal as it will lead the players to uninstall the game. So this could be a way but not the best way