Data Preparation

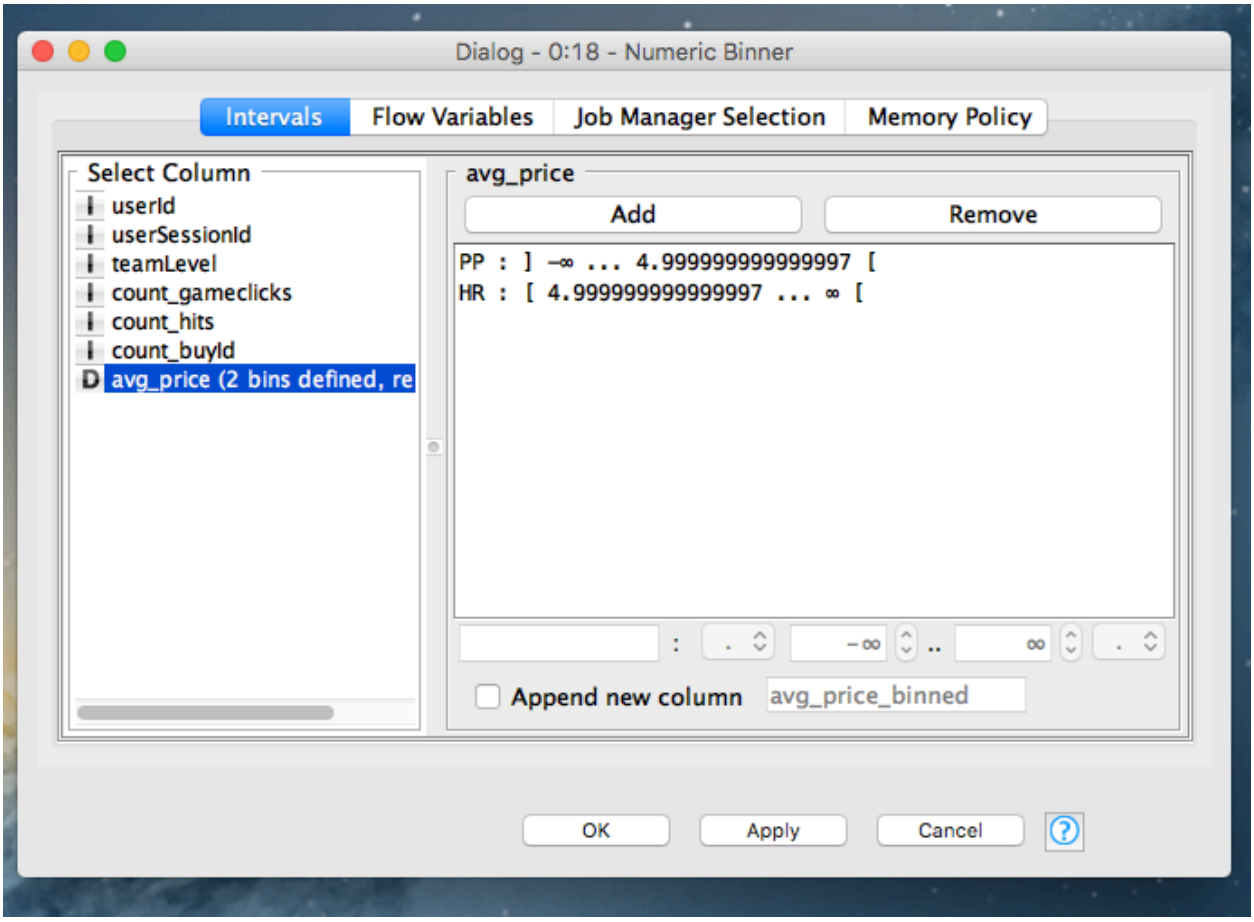
Analysis of combined_data.csv

Sample Selection

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

Attribute Creation

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:



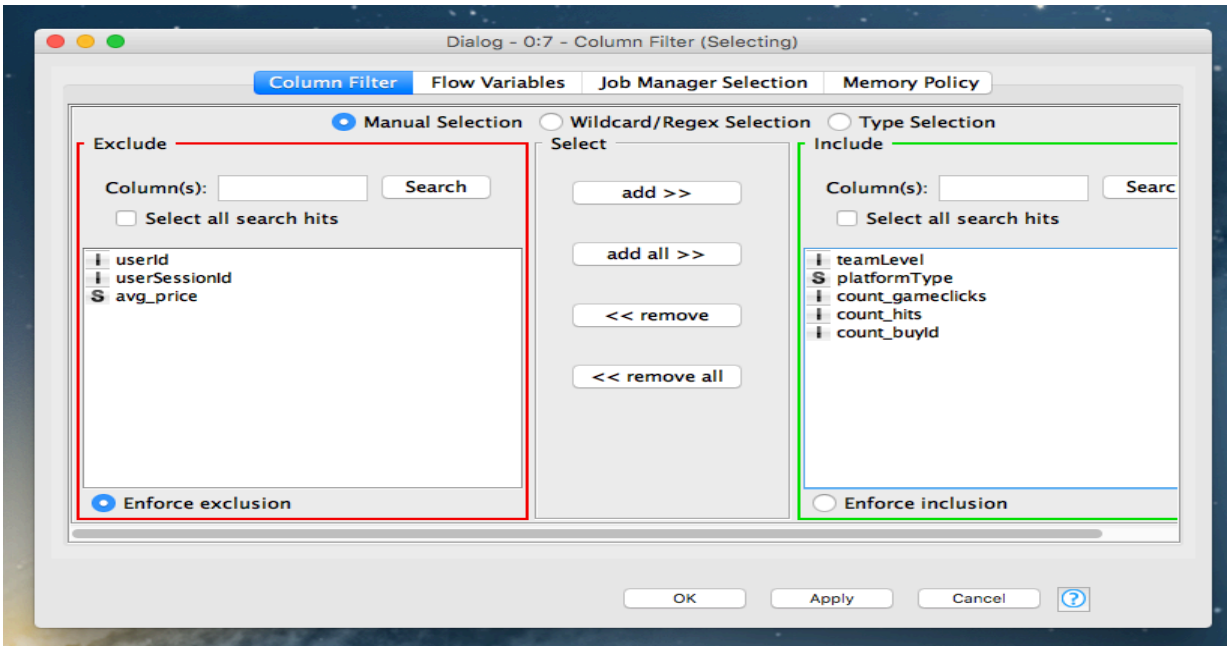
Description for how attribute was designed:
I uploaded the data and filtered those rows where count_buyId is nonzero. Then I applied Math formula (Avg_price >=5) to create binary attribute variable called HighRollers. The attribute value =1 represents HighRolles and value 0 represents PennyPinchers

The creation of this new categorical attribute was necessary because <Fill in 1-2 sentences>.

Attribute Selection

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

Attribute	**Rationale for Filtering**
userId, userSessionId and avg_price	The filters that do not help accurately predict the target should be removed.
userId	Highly-branching attribute. Decision tree bias toward high information gain, so this attribute can prevent generalized pattern.
userSessionId	Highly-branching attribute. Decision tree bias toward high information gain, so this attribute can prevent generalized pattern.
avg_price	The categorical attribute using binning was created based on this attribute. So if this attribute is included than there is nothing to predict, the model will have all the information and will have 100% accuracy.

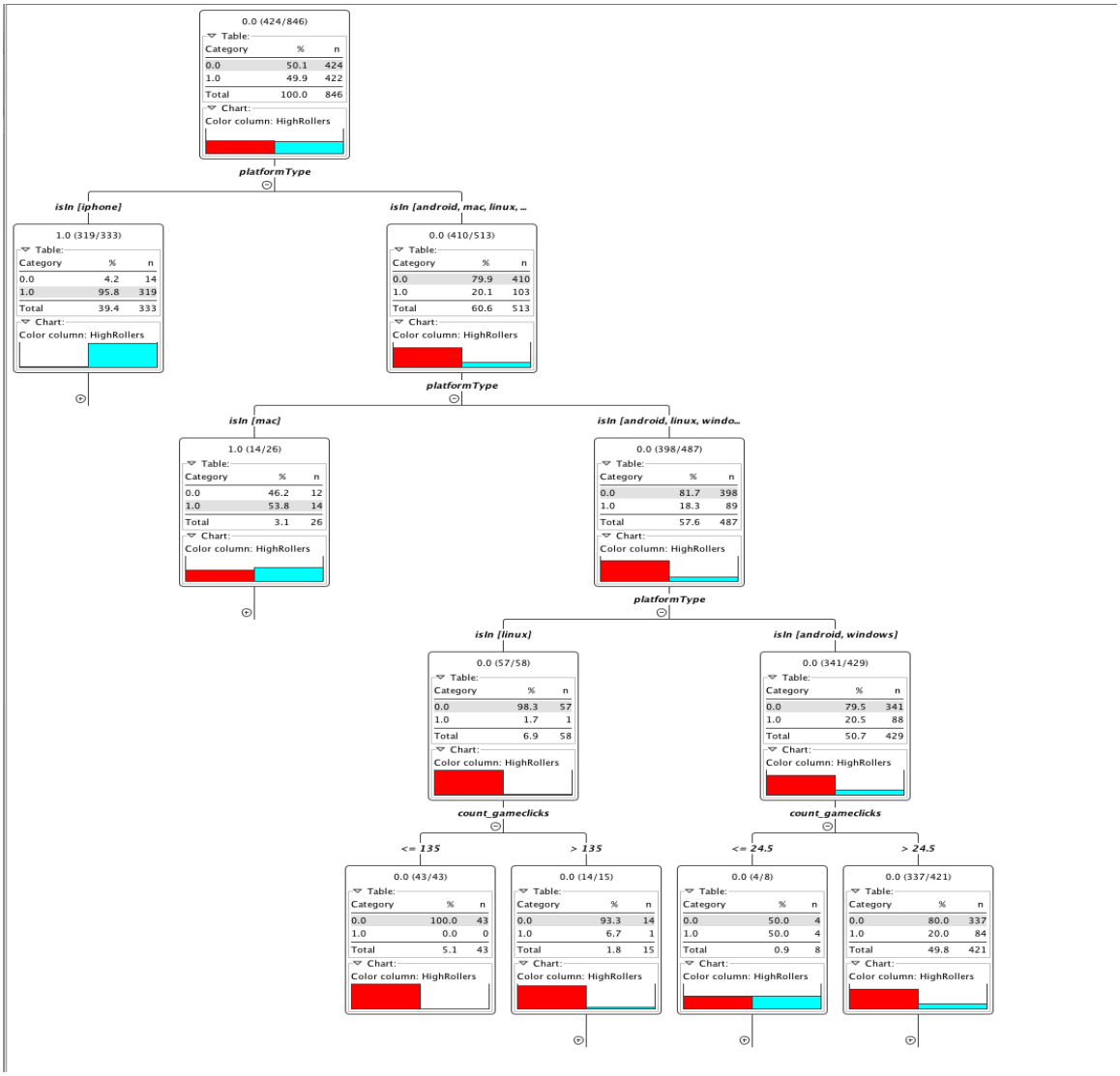


Data Partitioning and Modeling

The data was partitioned into train and test datasets.
The **train data (60%)** data set was used to create the decision tree model.
The trained model was then applied to the **test data (40%)** dataset.
This is important because we need to check accuracy of our model and improve the model performance by testing it on test set of data that was not used to train a model.

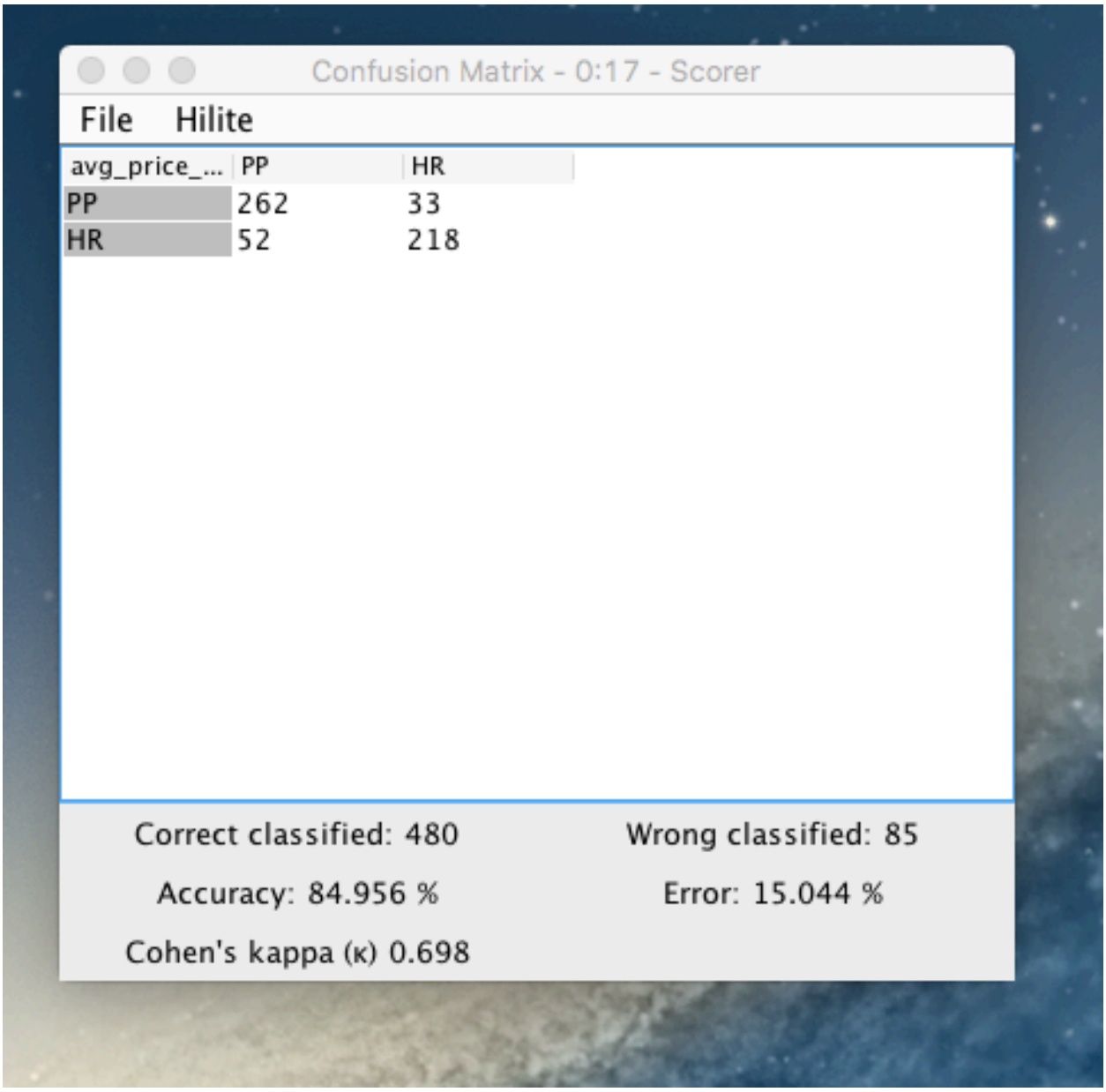
When partitioning the data using sampling, it is important to set the random seed because this ensures that you will get the same partitions every time you execute this node. This is important to get reproducible results.

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



Evaluation

A screenshot of the confusion matrix can be seen below:



As seen in the screenshot above, the overall accuracy of the model is 100**%**

Confusion Matrix

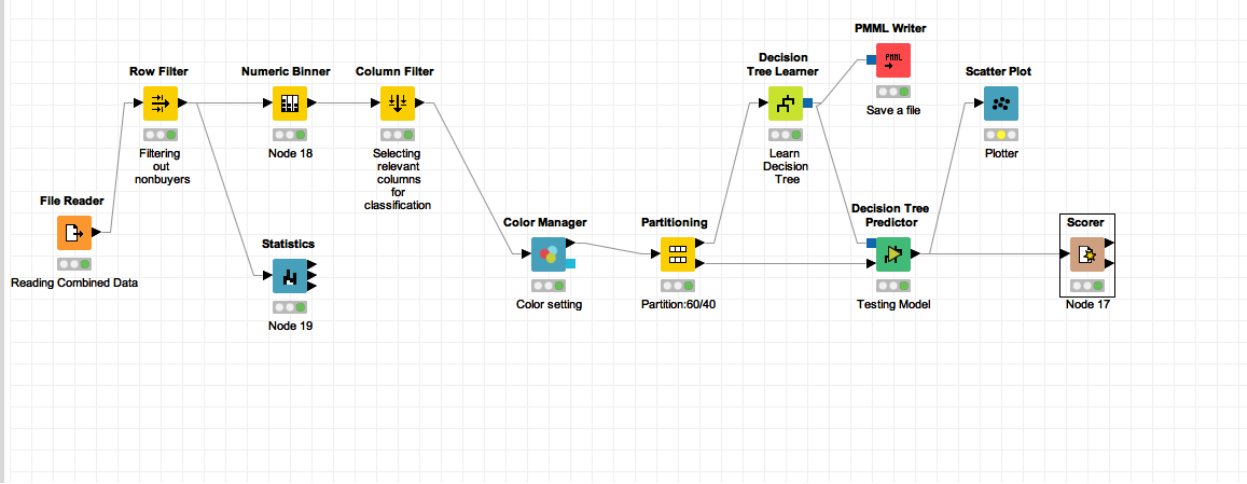
	PennyPitchers	HighRollers
PennyPitchers(PP)	True Positive TP (262)	False Negative (33)
HighRollers(HR)	False Positive (52)	True Negative (218)

So in this case, the
Correctly predicted, PennyPitchers = 262 and Correctly predicted HighRollers = 218
Incorrectly predicted, PennyPitchers = 52 and incorrectly predicted HighRollers = 33

Accuracy = (TP + TN) / (TP +TN + FP + FN)
= 84.956

Analysis Conclusions

The final KNIME workflow is shown below:



What makes a HighRoller vs. a PennyPincher?

The PlatformType is the main decision factor on deciding who wiis going to be HighRoller Vs. PennyPincher. From the prediction we see that users who uses iphone spends more amount and are HighRoller vs the users who play game using devices on other operating system (i.e. android, window, mac or linux etc).

Specific Recommendations to Increase Revenue
1. Since iphone users are HighRollers, the game offers should be high values if platformType is iphone, since that will be more appealing to them and are more likely to buy them.
2. Since other operating system device (i.e. android, mac, windows etc) and linux are generally PennyPinchers, they should be given low values. It is possible that they will purchase low value items but in more volume, so there should be some other strategy.

References:

1. https://www.youtube.com/watch?v=RHsO10q7e2Y