Yan Zichu 300476924

Comp309 Assignment 3

Core

I used the old dataset before update.

Business understanding:

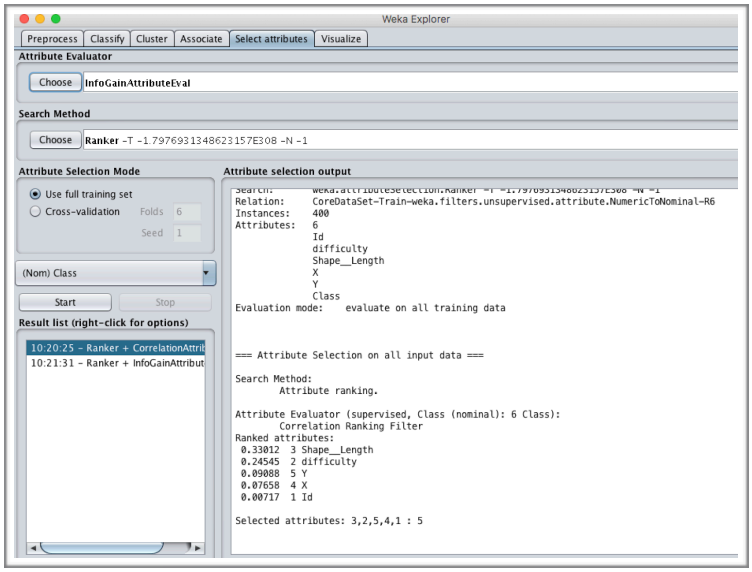
Data mining goal: The overall aim of this assignment is to develop the best possible machine learning system to predict the completion times of given tracks.

Core: The model I build need to be able to predict the class of future test data.

Data understanding:

In the core part, we have given training set and test set which have 5 attributes (id, difficulty, X, Y, Class). Difficulty represents the difficulty of finishing the track, the value of Class is based on difficulty, X, Y these 3 attributes.There are not too much data(400) in the training set but every data is high quality so it is good enough for the aim of Core part. And there is only one missing value exist in the training set under difficulty attribute.

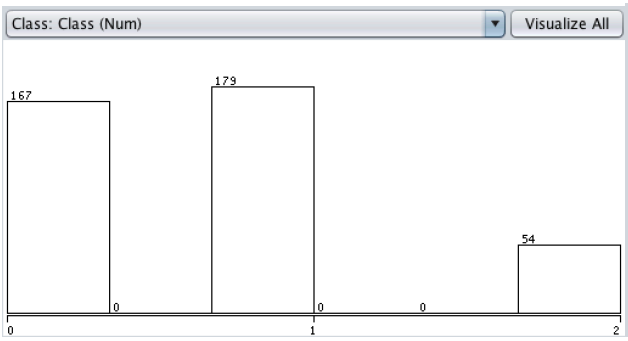
Data preparation:

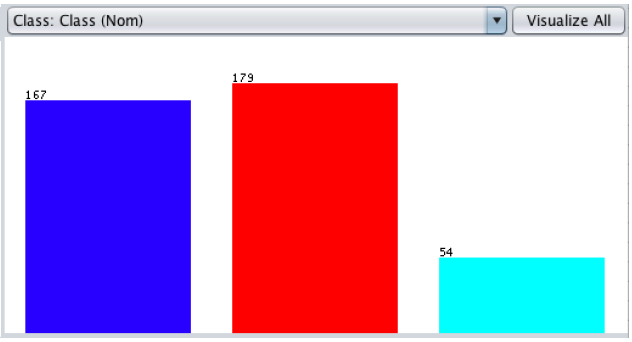


1.Select Data:
Rationale for Exclusion attributes: I decide to remove id because this attribute is irrelevance with predicting a category because of the small number of Ranked value. Rationale for Inclusion attributes: X,Y,Shape_Length and difficulty these 4 attributes will influence value of Class a lot because values of Ranked are big which shows they have relation with Class. **But actually I was not sure with difficulty cause it seems like a attributes made by human which is not a objective attribute.**

2.Clean data: because difficulty have a missing value so I use filter RemoveWithValue to remove that missing value.

3.Format data: Due to the fact that Class attribute use number (from 0 to 2)to represent degree of difficulty, while Weka will regard it as a numeric attribute but in the Core part the aim is to predicting a category(0,1,2) in testset through training machine by learning from training set. So Class attribute should be nominal type, then I used unsupervised preprocessor NumericToNominal to transfer Class attribute to nominal.





before NumericToNominal after NumericToNominal

4.Build Class attribute in test set: The purpose of this core part is to get the result of predicting categories in Class attribute.And at same time , I need to fill in instances into the Class attribute avoid Null problem when I import it in Weka.

C	D	E
Y	Class	
-39.792327	?	
-44.658424	?	
-38.752476	?	
-43.424834	?	
-38.975784	?	
-44.936069	?	
-44.369277	?	
-39.150284	?	
-39.858388	?	
-38.118776	?	
-35.611549	?	
-41.333022	?	
-44.799972	?	
-44.137355	?	
-38.922318	?	
-39.330399	?	
-45.893582	?	

Modeling:

Select modeling technique: The aim of Core is classification so I need an algorithm can classify nominal variables in Class by using some numeric data from X,Y and Shape__Length.So I tried to run Logistic algorithm.

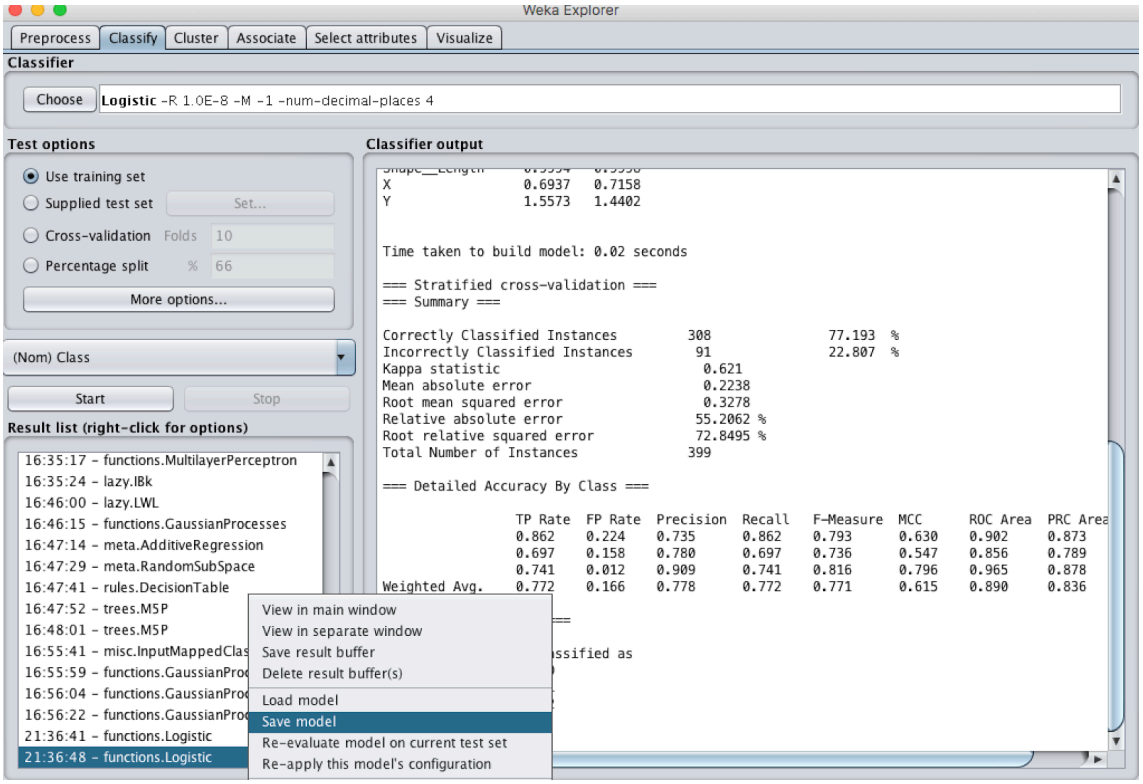
Logistic regression is different from linear regression. The essence of logistic regression algorithms is actually a classification algorithm, and it is not intended to predict a certain value. So the result is in the same form as the classification. Logistic regression is used for the classification of discrete variables, that is, the range of its output y is a discrete set, mainly used for class discrimination, and its output value y represents the class belonging to a certain class. Logistic Regression is mainly used to classify

problems. It is often used to predict probabilities. For example, knowing a person's age, weight, height, blood pressure and other information, predicting the probability of suffering from heart disease. The classic Logistic Regression is used for the two-category problem (only 0, 1 and 2).

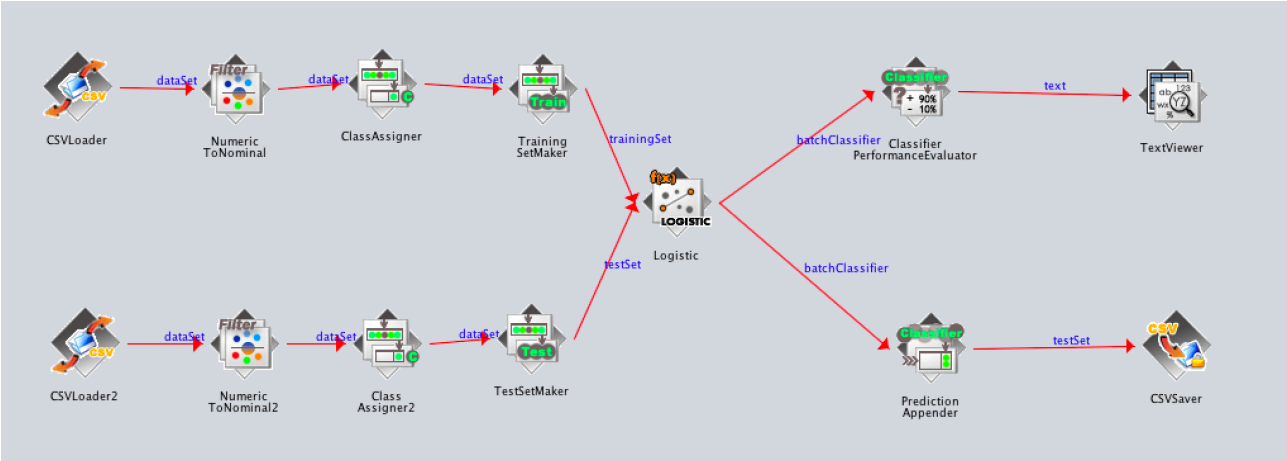
For any x value, the corresponding y value is within the interval (0, 1).

The function formula is : $$f(z) = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}},$$

Build model:



Assess model:



Evaluation :

```
=== Classifier model (full training set) ===

InputMappedClassifier:

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                   Class
Variable               0       1
================================
Shape__Length    -0.0006  -0.0002
X                -0.3657  -0.3344
Y                 0.443    0.3648
Intercept        87.4417  77.1828


Odds Ratios...
                   Class
Variable               0       1
================================
Shape__Length     0.9994   0.9998
X                 0.6937   0.7158
Y                 1.5573   1.4402

Attribute mappings:

Model attributes                    Incoming attributes
------------------------            ----------------
(numeric) Shape__Length         --> 1 (numeric) Shape__Length
(numeric) X                     --> 2 (numeric) X
(numeric) Y                     --> 3 (numeric) Y
(nominal) Class                 --> 4 (nominal) Class

=== Re-evaluation on test set ===

User supplied test set
Relation:     CoreDataSet-Test
Instances:     unknown (yet). Reading incrementally
Attributes:   4

=== Summary ===

Total Number of Instances                0
Ignored Class Unknown Instances                123

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Ar
                 ?        ?        ?          ?        ?          ?        ?         ?
                 ?        ?        ?          ?        ?          ?        ?         ?
                 ?        ?        ?          ?        ?          ?        ?         ?
Weighted Avg.    ?        ?        ?          ?        ?          ?        ?         ?

=== Confusion Matrix ===

 a b c   <-- classified as
 0 0 0 | a = 0
 0 0 0 | b = 1
 0 0 0 | c = 2
```

And then I use the result be saved to upload to the Kaggle and I got this result:

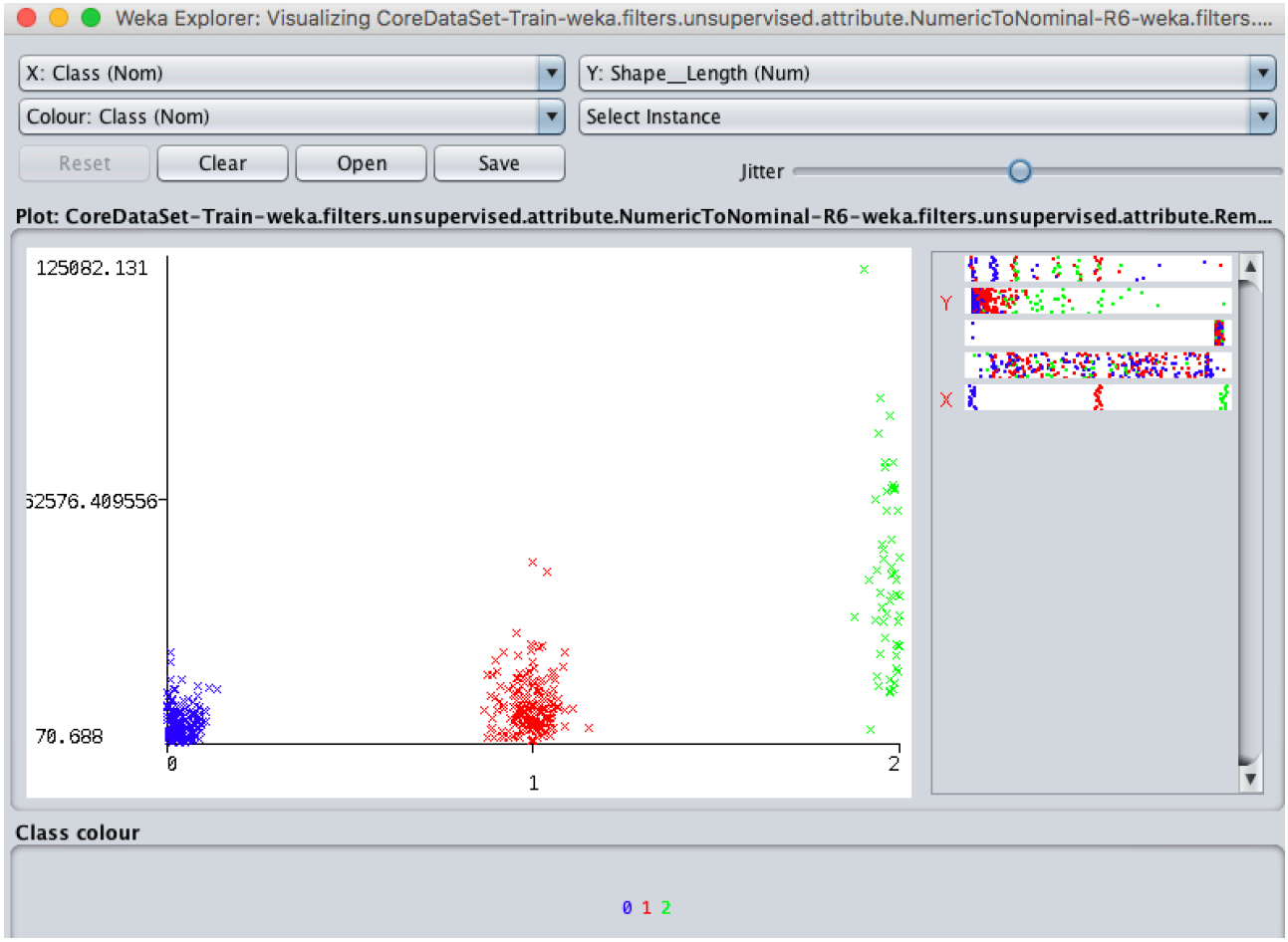
final2result.csv **0.91666**
9 minutes ago by Patrick Yan
add submission details

After I got this result I tried to remove difficulty as I mentioned before “But actually I was not sure with difficulty cause it seems like a attributes made by human which is not a objective attribute.” Then I got this result :

result5.csv **0.94444**
2 days ago by Patrick Yan
add submission details

This result proves that my guess is correct: difficulty is an manmade attribute and it should be removed when we predict objective facts, we should avoid using artificial subjective data although it is valuable.

Visualisation :



From this scatter plot we can see that the larger the shape length, the larger the value in the corresponding class. It shows that the shape length is positively correlated with the class.

Completion

Initial System

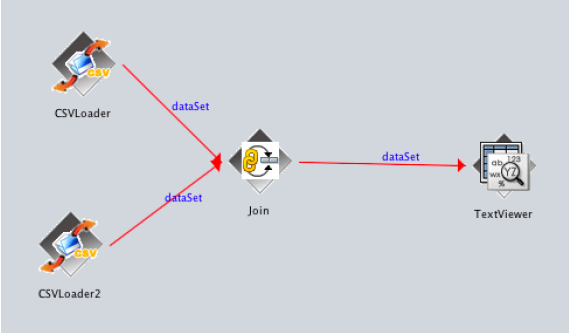
Business understanding:

Data mining goal: The overall aim of this assignment is to develop the best possible machine learning system to predict the completion times of given tracks.

Completion request: The model I build need to be able to predict the time how long we can finish the track.

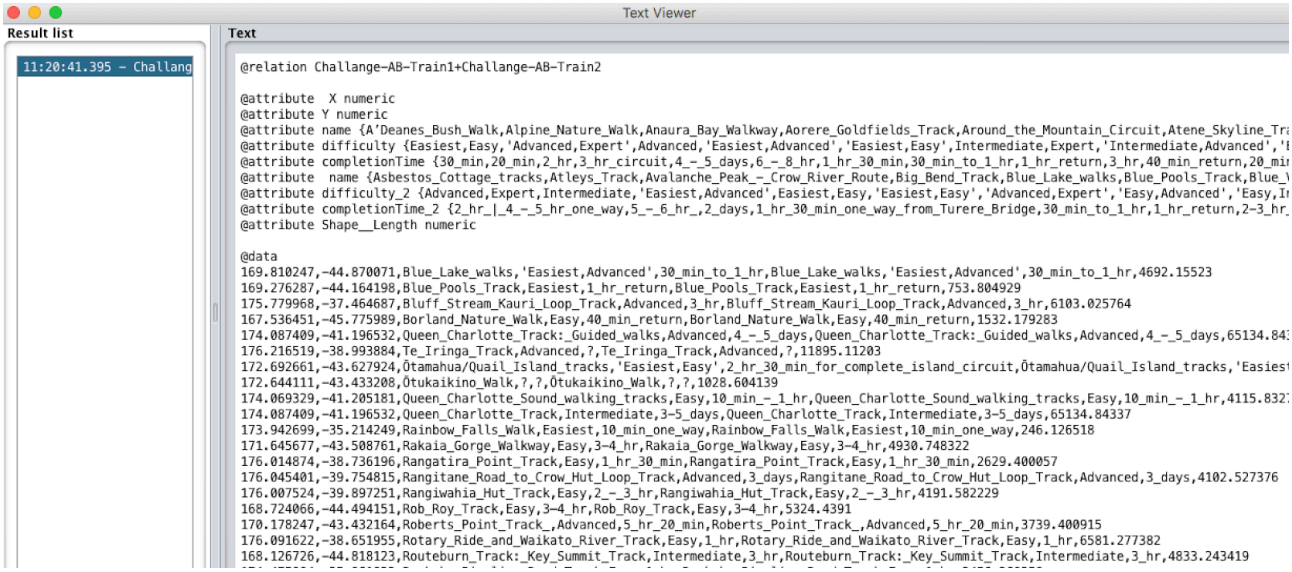
Data understanding:

At the beginning of the design, before uploading Kaggle, I refer to the algorithm of Core part.Although the purpose of these two parts are that develop the best possible machine learning system to predict the completion times of given tracks. However, since the dataset of the completion part becomes richer, the actual required algorithms are not the same. Since Time is a numeric type, logistic regression is no longer applicable. What we want to do here is to predict the Time required by known data. It is not a classification problem, so firstly I think of the linear regression algorithm to find the relationship between the independent time and the dependent variable.



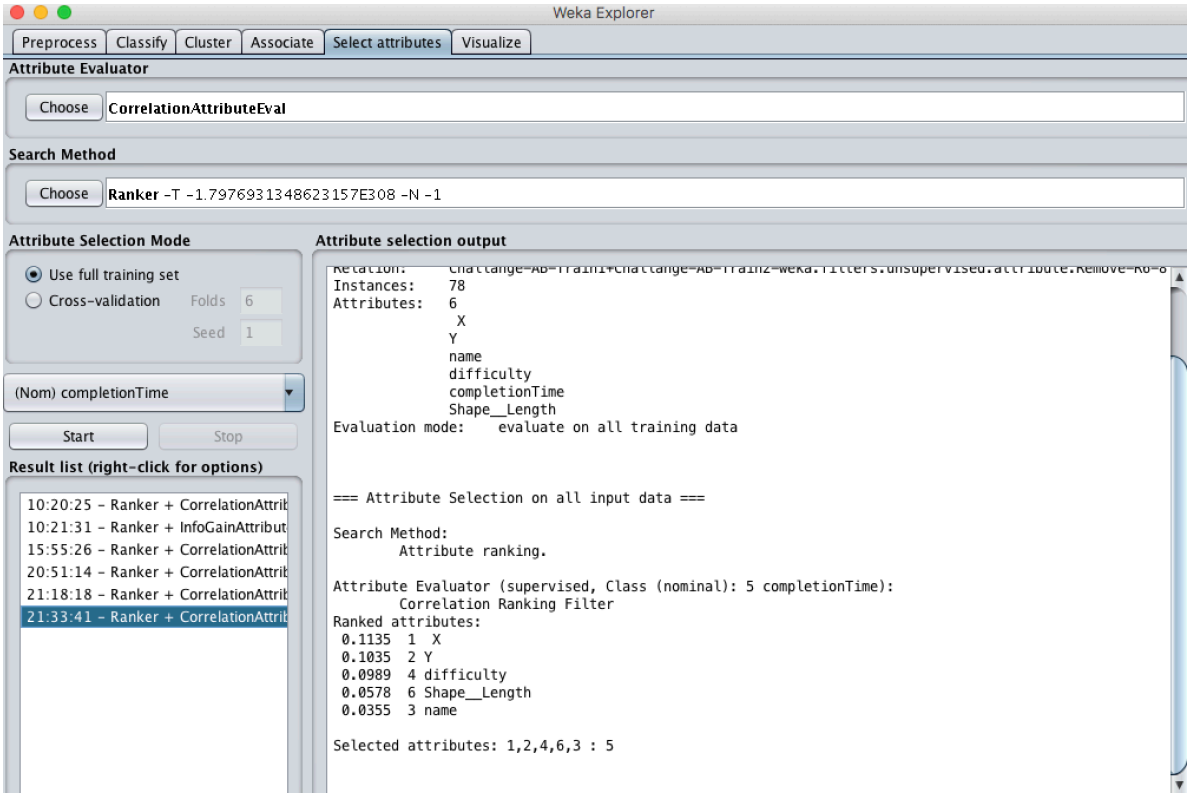
Data preparation:

1.I merge two training set together for gathering the useful data together (X , Y, completionTime, shape_Length).And as shown below, there are duplicate attributes and I would remove them.



2. After merge, I remove name, introduction ,OBJECTID ,hasAlters, walkingAndtramping, dateloadToGIS these attributes are not useful for predicting time so I remove them.

3. Then I used select attributes to find which attributes are irrelevant attributes after I remove those literal attributes.(But the number of correlation are pretty low.)



4.Remove missing value by Weka.

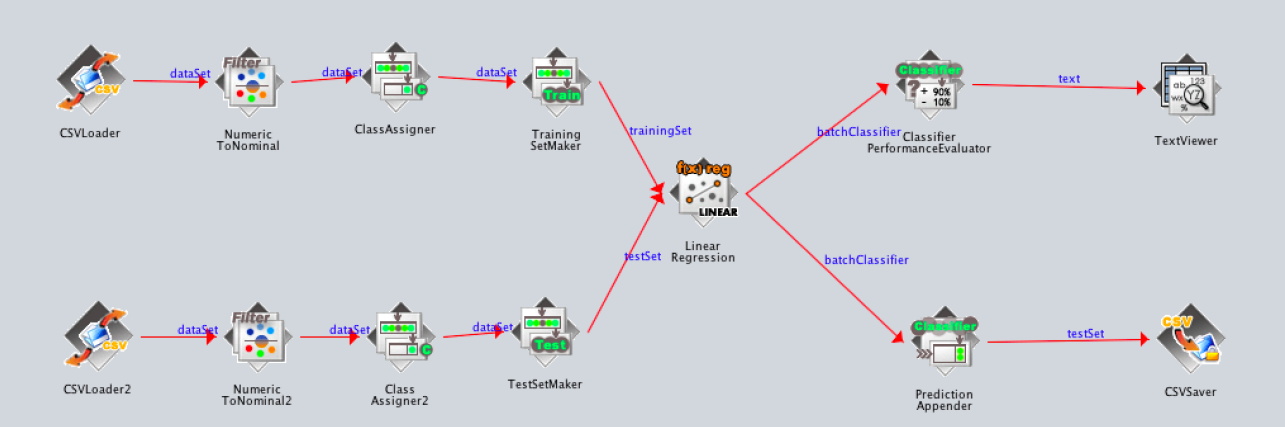
5.Use StringToNumeric for test set. Because I added a new attribute completionTime is test set and instance of it are ?(missing value).So weka regard them as String but I need it be numeric to match my training set.

6.Due to the Dummy file what minute be the unit of time so I transfer time into minutes by using excel.

A	B	C	D	E	F	G
Shape__Len	X	Y	difficulty	completionTime		
59.1001946	174.866523	-41.255969	Easy	10		
70.688113	174.071833	-35.319954	Easy	30		
81.6902574	173.857523	-35.715147	Easy	10		
89.8950159	173.084357	-34.72311	Easiest	20		
100.49515	171.327566	-42.115168	Easiest,Easy	20		
116.099977	167.713095	-45.430106	Easy	10		
125.59567	173.371187	-35.53639	Easiest	5		
137.869322	176.346294	-38.175303	Easiest	10		
139.071821	176.215495	-38.401156	Easiest	5		
140.472352	172.26327	-42.530454	Advanced,E	10		
143.563654	168.010272	-45.029872	Easiest	5		
151.515708	174.39006	-36.06467	Easy	30		
169.014441	169.374993	-46.607301	Easy	5		
169.623794	173.963922	-35.218832	Easiest	15		
176.019142	174.166833	-35.231833	Easy	10		

Modelling:

Because I want to predict time so I tried Linear Regression first.



lg-result.csv 469.71347

2 days ago by Patrick Yan

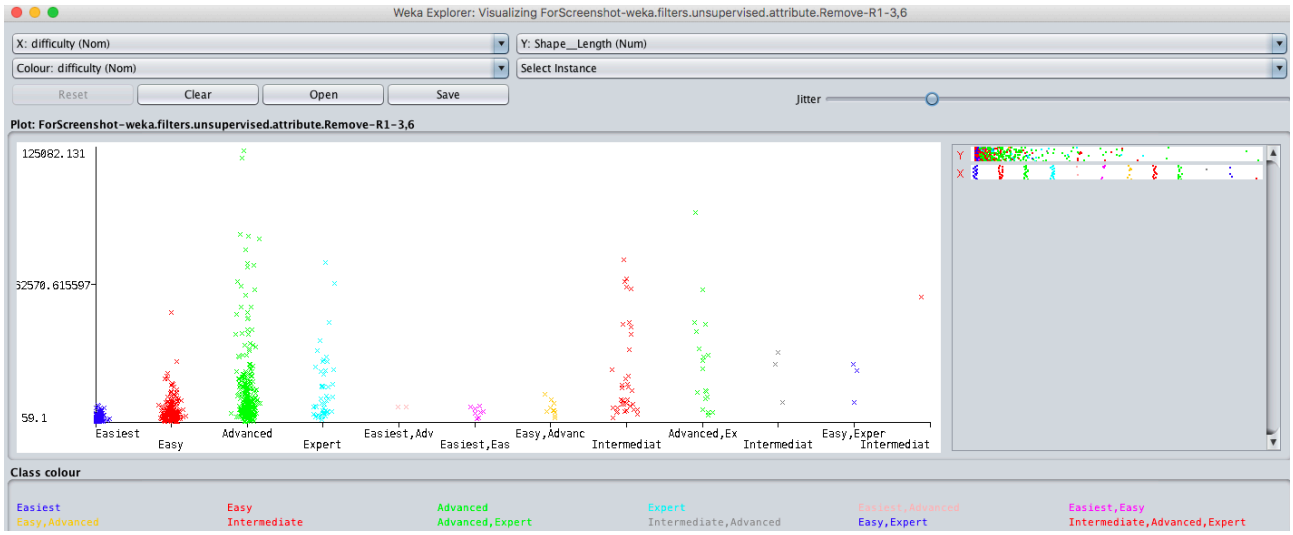
add submission details

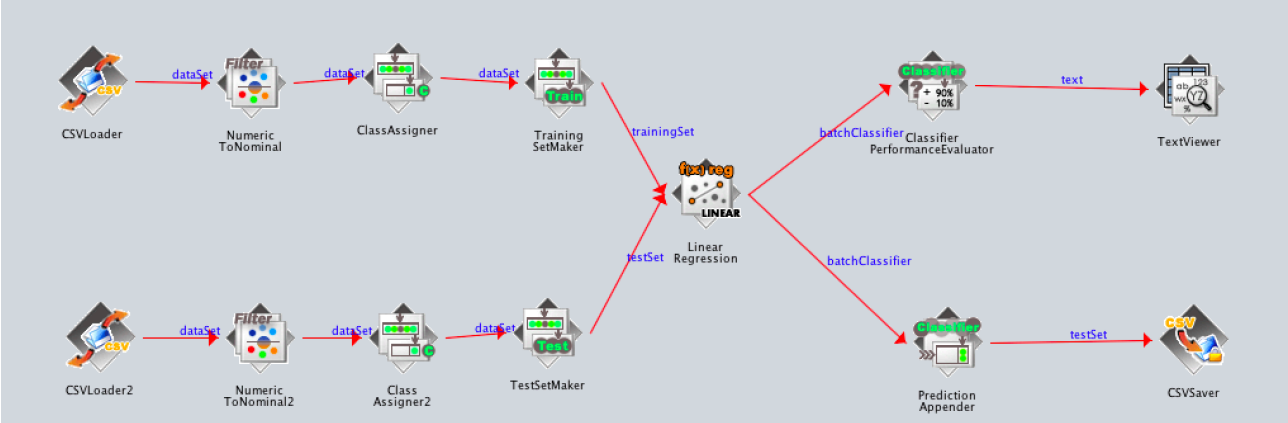
The result is not good. So I think maybe there are some issue happened when I converted those time in to minutes.It means I need to do data understanding again.

Intermediary system

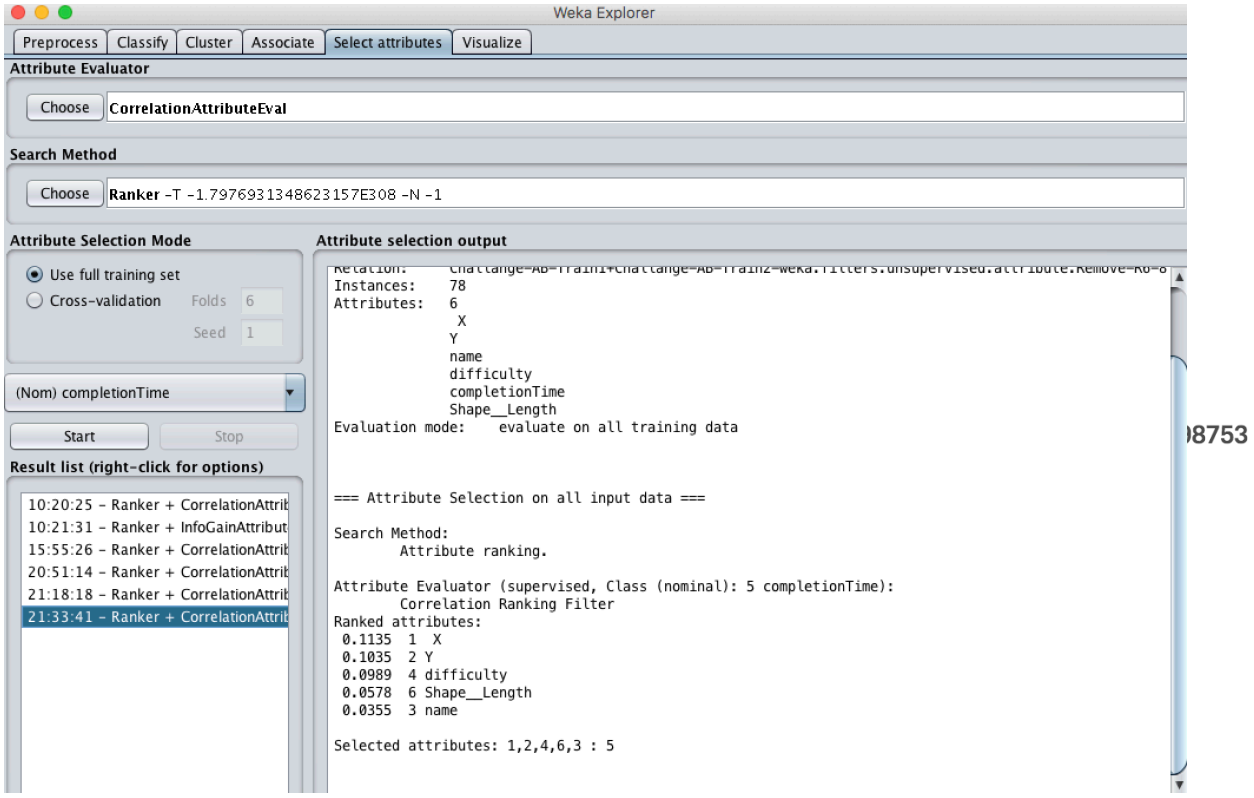
Data understanding:

1. At the very beginning , I converted 1 day as 24 hours. However ,it's impossible for a human being to walk whole day and it is unnecessary for a walker to do that.So I think I should regard 1 day as 10 hours.
2.And I removed difficulty attribute, because this attribute is influenced by shapeLength:





However, the error of this result is still too large, but I think data is reasonable now, so it means the algorithm I chosen is not very proper. At the same time, I also remembered the correlation values that were very

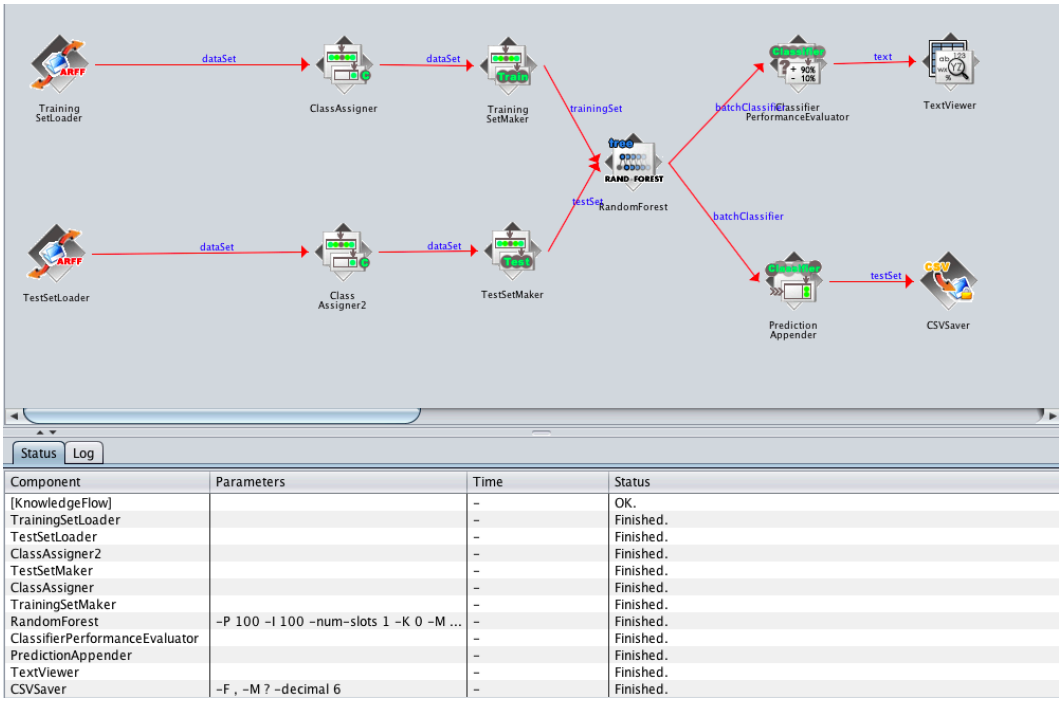


low.

Final System

Modelling:

Finally, I chose Random Forest because of those low value of correlation. (pre-request of using Random Forest)



Component	Parameters	Time	Status
[KnowledgeFlow]		-	OK.
TrainingSetLoader		-	Finished.
TestSetLoader		-	Finished.
ClassAssigner2		-	Finished.
TestSetMaker		-	Finished.
ClassAssigner		-	Finished.
TrainingSetMaker		-	Finished.
RandomForest	-P 100 -I 100 -num-slots 1 -K 0 -M ...	-	Finished.
ClassifierPerformanceEvaluator		-	Finished.
PredictionAppender		-	Finished.
TextViewer		-	Finished.
CSVSaver	-F , -M ? -decimal 6	-	Finished.

A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

And I would like to chose this as final system, because dataset and classifier technique is more reasonable : The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don't constantly all err in the same direction). While some trees (prediction of time) may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction.

Challenge

High accuracy and good results are important, but machine learning is for humans, so sometimes objective data cannot reflect the real results we want.

The model I finally chose is based on random forest, which is not complicated to explain :Decision Trees is the fundamental of random forest because they are the building blocks of the random forest model. Fortunately, they are pretty intuitive.

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.For decision tree, in these tree

structures, leaves represent class labels and branches .At each node, it will ask —What feature will allow me to split the observations at hand in a way that the resulting groups are as different from each other as possible (and the members of each resulting subgroup are as similar to each other as possible)?

The low correlation between models is the key which is satisfy the result I had when I do select attributes.When algorithm running, some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. In my training set each tree can calculate different time ,some of time are accurate ,some of time are not proper, when algorithm combine those result the time will be more accurate.So this is why Random forest have good performance on my dataset.

However, based on my result ,I do not think it is fine to use it directly because the error of my final system is still quite high, 177(root mean squared error). So it is unreliable to use this system, everyone have his own preference of speed ,so it is hard to suitable for everyone ,especially, if wheelchair users use this system it will bring them a terrible experience because the dataset actually does not contain the data about time need to spend for using wheelchair.However it does not mean it can not be a reference of walker, it still can be use as a predict system but when track is short we need to pay more attention to the prediction because error is high which means we may spend much less time to finish the track rather than the time be predict by my system.