Assignment 2

Learning Objectives:
In this assignment you will explore input and output representations for machine learning.

1. Gain intuition about how differences in feature space representation affect the rule that is learned.
2. Analyze the effect of noisy data on the machine learning model
3. Gain experience transforming data from logfile format into a tabular form that can be used for machine learning.

Part A:
For this part you will use the three versions of “titanic.arff“ that are given in the Assignment 2 folder. Basically this data contains information about the passengers in the titanic. The attributes are crew, age, sex and survived (or not). In the original “titanic.arff” file, age is represented as a nominal attribute with values of child or adult. In titanic-age-numeric.arff, age is represented instead as a numeric attribute representing the actual age of the passenger. The final file “titanic_noise.arff” is a version of “titanic.arff” with a little bit of noise added to one of the attributes in two instances.

1. Using Weka, Visualize/Draw the tree generated for each of the the three sets of data. In order to do this for one file, load it on the Preprocess tab, then go to the classify tab. Choose J48 as the learner. Run the experiment. Right click in the results buffer to generate the tree. Paste an image of your tree in your assignment write up. You need to show three trees in total.

2. Now compare the results from the original file where age was a nominal attribute with the results where age was a numeric attribute. How are the results different? How is the tree different? Considering that it would be possible for the exact same tree to be learned (because a split at age 18 would produce the same child/adult distinction), why do you think J48 settled on a different model?

3. Now compare the results from the original file with the file that has noise added. Describe the effect of noisy data on the generated machine learning model. Based on the differences between models, which kinds of instances do you think the model with the noisy data would consistently make mistakes on?

4. Turn in your answers to questions 1-3

Part B:
For this part you will use Python to transform clickstream data into a tabular form that is appropriate for machine learning. Getting some familiarity with Python now will open up opportunities for you to play around with machine learning code later – even as soon as assignment 3! The raw data files you will use are in the zip file included with the assignment called searchlog.zip. That zip file also contains a Python script called SetupSearchData.py.

The data for this assignment is just a portion of the data that is included with the Emerging Internet Users project in the projects folder. That dataset includes logs of search behavior from students at a university in a rural area in the south of India. There is one log per student, identified by the LogID. What we will do in this assignment is aggregate information from students from their search log and create one instance per student in an output file. The python script called SetupSearchData.py that you were given creates for each student one instance where four variables are extracted: the LogID, Number of pages clicked on (NumPages), Number of accesses to wikipedia (WikiAccess), and the number of words copied from web pages into the survey form they were given (NumWords). The instructions below explain how to create this csv file by loading the script into python to execute it. You will then modify the script to create a different output file.

For a more expansive description of the available data in that data set and the context in which it was collected, see the Emerging Internet Users subfolder in the projects folder on blackboard. These files came from an SQL database, but in this assignment we will just process the exported files, with one table per file.

S.No.	Table Name	Purpose/Description
	tblLogPages	Details of each page viewed by the clients.
	tblLogAction	Details of each Client Action like Click, Copy, AddTab, CloseTab, Scroll.
	tbltblPageCopyEvents	List of all Copy Actions with CopyText. Not being Used

1. tblLogPages

S. NO.	NAME	TYPE	DESC	RANGE/Sample
	LogPageID	bigint(20)	Unique ID for each page viewed	Autoincrement from 1
	LogID	bigint(20)	FK to tblLogs	Autoincrement from 1
	PageURL	varchar(2048)	URL of page viewed	
	PageLoadTimestamp	Datetime	Page Loaded Timestamp	
	TimeOnPage	Int(11)	Total Time spent on this page in seconds	
	NumScrollEvents	Int(11)	Number of scrolling actions in a page	

PRIMARY KEY (`LogPageID `)
FORIEGN KEY (‘LogID) – **tblLogs**

2. tblLogAction

S. NO.	NAME	TYPE	DESC	RANGE/Sample
	ActionID	bigint(20)	Unique ID for each user action	Autoincrement from 1
	LogID	bigint(20)	FK to tblLogs	Autoincrement from 1
	ActionTypeID	bigint(20)	FK to tblLogActionType	Autoincrement from 1
	LogPageID	bigint(20)	FK to tblLogPages	Autoincrement from 1
	ActionTimestamp	Datetime	Action Timestamp	

PRIMARY KEY (`ActionID `)
FORIEGN KEY (‘LogID) – **tblLogs**
FORIEGN KEY (‘ActionTypeID) – **tblLogActionType**
FORIEGN KEY (‘LogPageID) – **tblLogPages**

3. tblPageCopyEvent

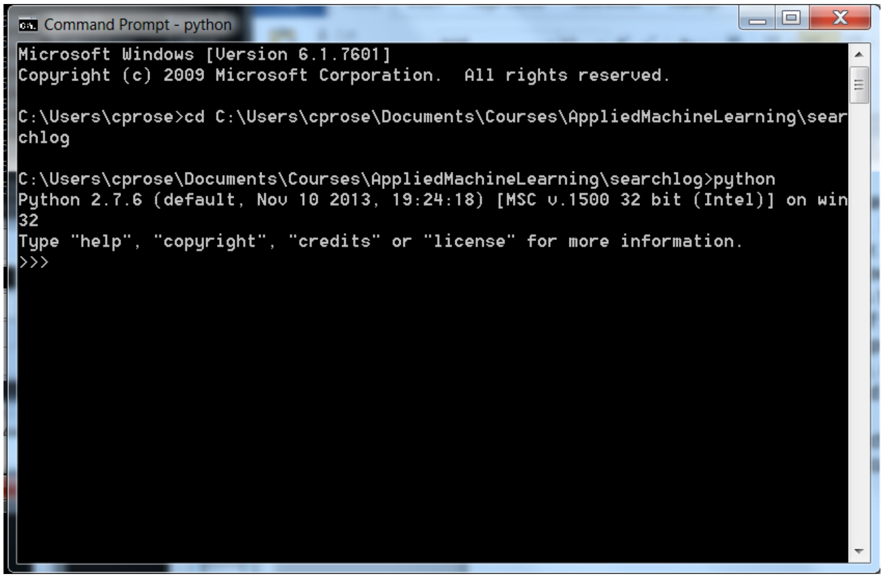
S. NO.	NAME	TYPE	DESC	RANGE/Sample
	CopyEventID	bigint(20)	Unique ID for each Copy Event	Autoincrement from 1
	ActionID	bigint(20)	FK ID for each Action	Autoincrement from 1

	CopyText	Text	Copied Text	

PRIMARY KEY (`CopyEventID `)
FORIEGN KEY (‘ActionID’) – **tblLogAction**

Here is a step by step guide:

1. First, if you do not have a version of Python running on your computer you will need to install it. I suggest you install version 2.7. You can get to the download page here: http://www.python.org/download/releases/2.7/. If you are using Windows, I suggest you use one of the Windows installers that is appropriate for your operating system version. Recent versions of Mac OS X come with Python pre-installed. Feel free to ask the TAs if you need help installing the software. First **cd** into the directory you created from the zip file and then launch python. If python is installed properly, you should be able to open a command line prompt, type **python**, and launch the interactive interpreter environment. Note that I have included some Python cheat sheets in the assignment 2 folder. You can also search for answers to your questions on google easily, especially the StackOverflow website. That is a great resource to get used to using.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\cprose>cd C:\Users\cprose\Documents\Courses\AppliedMachineLearning\sear
chlog

C:\Users\cprose\Documents\Courses\AppliedMachineLearning\searchlog>python
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

2. Now run the python script you were given using the command **execfile(‘SetupSearchData.py’)** at the python prompt. (Alternatively, you should just be able to run **python SetupSearchData.py** directly from the command line, without launching the interactive interpreter first) It will take a few moments to run, but then you should have a file called “output.csv” in the directory. This output file could be read into weka to run experiments. You can feel free to

experiment with that, but it is not required for this assignment. You can compare your file with “outputReference.csv” to make sure you created the file properly.

3. Now modify the script to add an additional variable. This variable will be analogous to WikiAccess, but instead of tabulating the number of accesses to Wikipedia, it will tabulate the number of accesses to google related pages (i.e., web pages with “google” in the URL). Please add the new column between WikiAccess and NumWords when you write your output. Once you have made the modifications, run the script to create the updated “output.csv” file. What you will turn in is your modified SetupSearchData.py script and your new “output.csv” file.