**Artificial Intelligence Project 1: Design Document**

**Alex Abrahamson, Wilson Harris, Matthew Nitschke**

1. **Description of the Problem**
2. **ARFF Converter**

Our ARFF converter is a simple cli application written in javascript and run via the javascript runtime environment Node.js. The parameters of the program ask for the filename of the .cvs file, and a list of attribute types. Once run, the program parses out the filename and attribute types. It then loops through each attribute type looking for types of “enum” or “date”, where if encountered asks the user for enum properties or date format. After the arguments have been parsed, arffConvert reads the desired .csv file and extracts the header line. This line is split into an array by its commas and looped through appending its name and correlating attribute type at each iteration. Finally, the data of the csv file is processed by splitting each line into an array. Each column is formatted by encoding any invalid characters, and ensuring that if the columns type is a string, that the data is wrapped in quotes. After the data processing is complete the file is written to the user's current directory.

When designing this application the first decision to be made was to figure out how to get the attributes data types. We went through two iterations of this design, one used type inference by looking at the actual data of the csv file trying to guess what data type the column was, and the other simply required the user to enter each type as an argument when the program was called. We decided on the second option due to the lack of consistency in the csv files which caused for faulty type inference. Another problem which had to be fixed was dealing with commas within quotes. Csv files usually denote a string with a comma in it by surrounding the string with quotes. This breaks splitting each column by commas because the algorithm only looked for singular commas. A simple regex selector, found on stackoverflow, which ignores commas inside quotes was used to fix this problem. Finally, WEKA requires all strings to be wrapped in quotes. A simple format column function solved this problem by adding quotes to lines which needed them. Although small problem surfaced when this was implemented, because single quotes were used to wrap the data, and text with a single quote in it (such as i’m, that’s, it’s) broke the quote and henceforth broke the parse. To solve this, a before removing wrapping quotes, a find and replace is done on the data replacing any single quotes with their escaped counterpart: \’.

1. **Experimental Design**
   1. **Data Sets**

We will use five different data sets. When deciding what data sets to use, we considered two characteristics, the number of instances and attributes. We expect these characteristics to be useful in detecting which algorithms take the longest to process data with different types of data sets. We classified a low number of instances to be less than 500 and a high number to be greater than 1,000,000. Attributes were classified as a low if less than 50 and high if greater than 100,000. All of our data was retrieved from the UC Irvine Machine Learning Repository. The first data set, Pems-SF, had a low number of instances and a high number of attributes. The data described the occupancy of car lanes on the freeways in San Francisco. Hemp mass, our second data set, had a high number of instances and a low number of attributes. The data was produced by simulations of collisions that produce specific particles and their decay products. Next, we used a Parkinson Disease spiral drawing data set which had a low number or instances and attributes. Subjects, some with Parkinson Disease, had their handwriting tested to see if Parkinson Disease can be identified through handwriting. Our fourth data set had a high number of instances and attributes. It is a set of URLs and their attributes with the goal of classifying malicious websites. The final data set is a smartphone-based recognition of human activities data set. It had a moderate number of instances and attributes.

* 1. **Machine Learning Algorithms**

We will be testing five learning algorithms:

1. K-nearest neighbor
2. Naïve Bayes
3. Logistic regression
4. Decision tree with pruning
5. Support vector machine with a nonlinear kernel

These algorithms are already written on Weka, so the only editing of the algorithms will be tuning.

* 1. **Evaluation Measures**

We will evaluate the algorithms using two Weka analysis methods: Percent correct and elapsed time training. Accuracy and speed are two common evaluations of machine learning algorithms. (I need to try and find an article that can support this claim). Percent correct is a useful evaluation for understanding the overall accuracy of an algorithm with a single number. We chose elapsed time training because in our tests there will be more training data than testing data. Therefore, the difference between times will likely be more significant. K-nearest neighbor is an exception, because it is a lazy algorithm, but elapsed time training will provide valuable information on computation time for the other algorithms.

**References**

Notes:

Pems-SF Data Set: Low number of instances, high number of attributes (440, 138672) The data describes the occupancy   
rate, between 0 and 1, of different car lanes of San Francisco bay area freeways. The measurements cover the period from Jan. 1st 2008 to Mar. 30th 2009 and are sampled every 10 minutes.

HEMPMASS Data Set: High number of instances, low number of attributes (10500000,28)These signatures are learned from Monte Carlo simulations of the collisions that produce these particles and the resulting decay products. In each of the three data sets here, the goal is to separate particle-producing collisions from a background source.

Parkinson Disease Spiral Drawings Using Digitized Graphics Tablet Data Set: Low number of instances, low number of attributes(77,7) The PD and control handwriting database consists of 62 PWP (People with parkinson) and 15 healthy individuals who appealed at the Department of Neurology in Cerrahpasa Faculty of Medicine, Istanbul University. From all subjects, three types of handwriting recordings (Static Spiral Test (SST), Dynamic Spiral Test (DST) and Stability Test on Certain Point (STCP)) are taken.

URL Reputation Data Set(2396130,3231961): High number of instances, high number of attributes. List of URL’s with their attributes to try and identify malicious websites.

Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set. (10929, 561). Moderate number of instances and attributes. The experiments were carried out with a group of 30 volunteers within an age bracket of 19-48 years. They performed a protocol of activities composed of six basic activities: three static postures (standing, sitting, lying) and three dynamic activities (walking, walking downstairs and walking upstairs). The experiment also included postural transitions that occurred between the static postures. These are: stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand. All the participants were wearing a smartphone (Samsung Galaxy S II) on the waist during the experiment execution. We captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz using the embedded accelerometer and gyroscope of the device. The experiments were video-recorded to label the data manually. The obtained dataset was randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.