

---

## COMP 421 – Final: SMS Spam Detection

---

**Date Assigned: April 19, 2021**

**Completion Date: May 4, 2021 11:55pm**

### Description

#### PART I

In this project, you will create a program that automatically predicts whether an SMS message is spam or ham (not ham).

This project will be split into your midterm and final.

For Part I, you will create a PHP program that reads SMS messages from a database table, converts the SMS messages into features, and put those features in a CSV file.

The SMS messages along with their class label (spam, ham) are in a file named **spam.sql**. The first column is the class label spam or ham. Ham means the text message is not spam. The second column is the text message.

### Specification

1. Create a MySQL database named `predict_spam`.
2. Use the SQL statements in the `spam.sql` file to create a table named `spam`. The statements in the SQL file will also insert data into the `spam.sql` table.
3. Create a PHP file named **`compute_features.php`**.
4. The PHP file should read each row in the database, compute features, and print out a csv file with each feature.

You should compute the following features:

- **`doesHaveLinks`**
  - This feature is True if a SMS message has links and False if a SMS message doesn't have links.
- **`doesHaveSpammyWords`**
  - This feature is True if the SMS contains spammy words and False if a SMS message does not have spammy words
  - To determine what words are spammy, look through the dataset and pay close attention to curse words and any words that are in the spam category but not in the ham category.
  - You should choose 10 spammy words.
- **`lengthOfText`**

- The number of characters including spaces in the text message.

These features should be printed to a csv file named **features.csv**.

The file should have the following columns: **doesHaveLinks**, **doesHaveSpammyWords**, **lengthOfText**, **class label**.

**The class label is SPAM or HAM.**

At a minimum, your program should have:

- A main function
- A function for each feature
  - doesHaveLinks(email)
  - doesHaveSpammyWords(email)
  - lengthOfText(email)
- A function for writing the features to the CSV

You can create more functions if you want to.

The name of your PHP file should be **compute\_features.php**.

## PART II

For Part II, you will:

1. Use Weka to read features from the csv file created in Part I to create a decision tree diagram.
2. Convert the decision tree diagram into PHP using conditionals (if/elif/else).

### Specification

At a minimum, your program should have:

- A main function
- A function named `make_prediction`
  - o This function takes as input the features and returns the prediction (this is where the decision tree code goes )
- The functions from `compute_features.php` that compute features

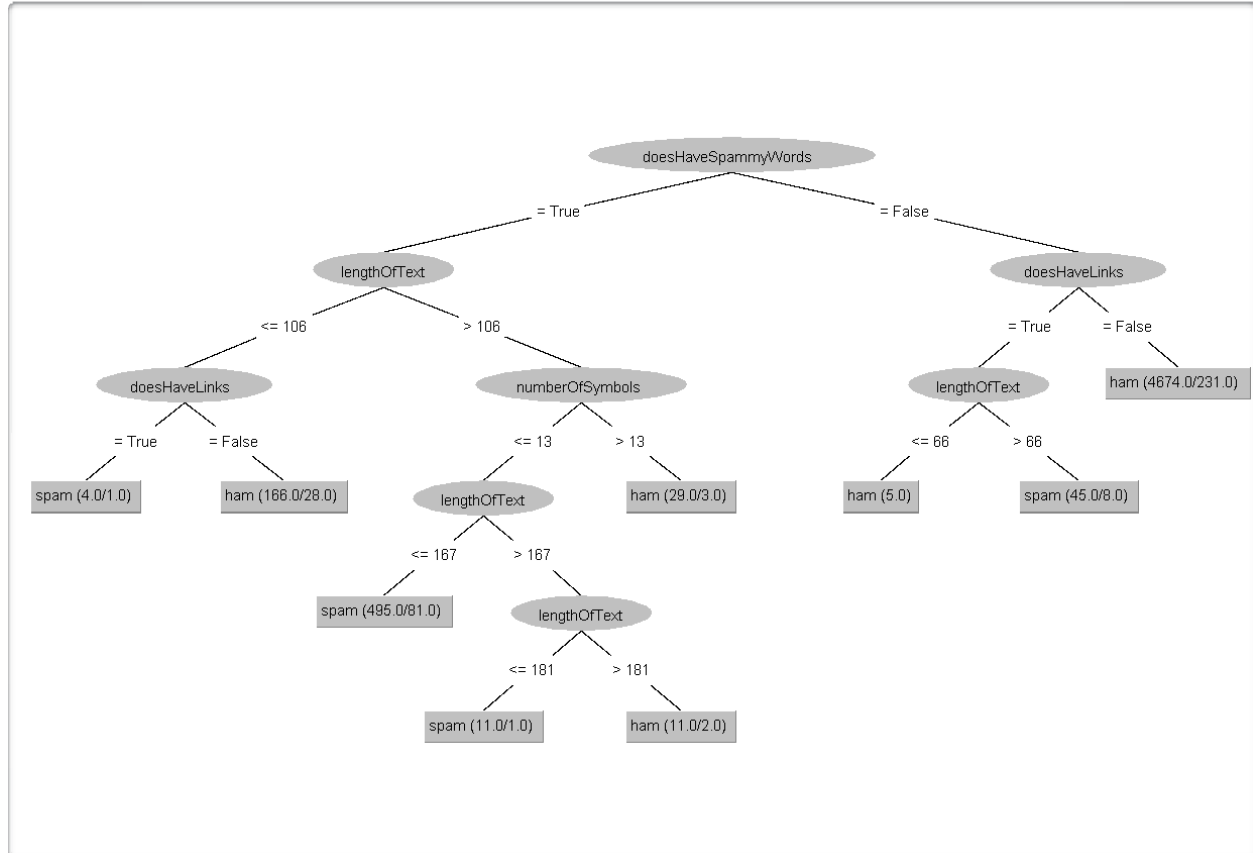
Create a html form that contains a textarea. This textarea is where you can copy and paste email messages. The html form should also contain a button. Place these input controls inside of an html form.

The name of the html file should be **predict\_spam.html**. The html form should **POST** to a file named **predict\_spam.php**.

`predict_spam.php` will get the email message, convert the email message into features (using the functions you created in **compute\_features.php**), use the if/else statements from the decision tree to predict whether the email is spam or ham. The program should print spam or ham.

You can copy and paste the functions to create features from **compute\_features.php** into **predict\_spam.php**.

Tree View

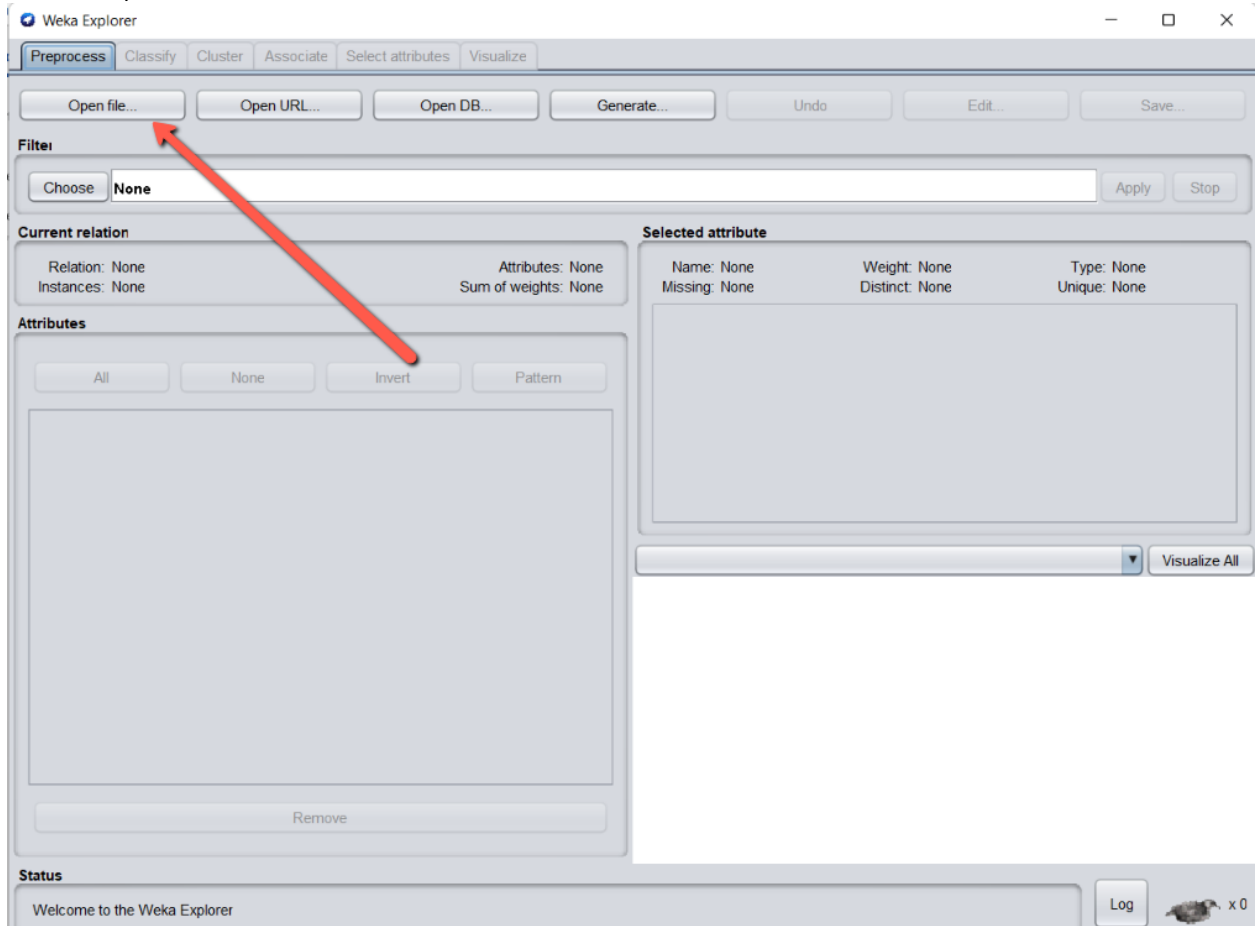


To generate the decision tree:

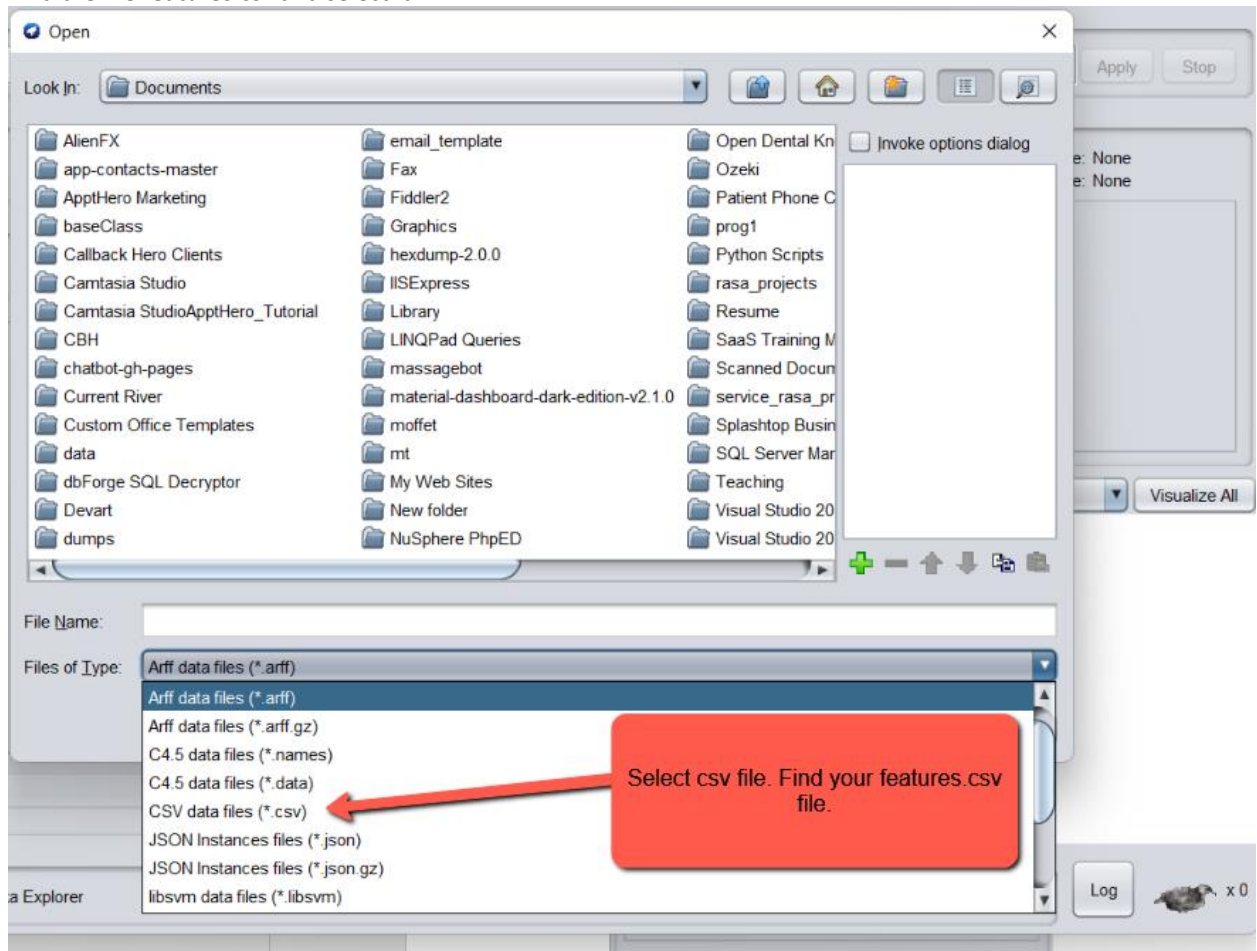
1. Open Weka
2. Click the Explorer button



3. Click the Open file button



4. Find the file features.csv and select it



Your Weka program should look similar to this, but with your features.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter: Choose **None** Apply Stop

**Current relation**

Relation: weather.symbolic Instances: 14 Attributes: 5 Sum of weights: 14

**Attributes**

All None Invert Pattern

No.	Name
1	<input checked="" type="checkbox"/> outlook
2	<input type="checkbox"/> temperature
3	<input type="checkbox"/> humidity
4	<input type="checkbox"/> windy
5	<input type="checkbox"/> play

Remove

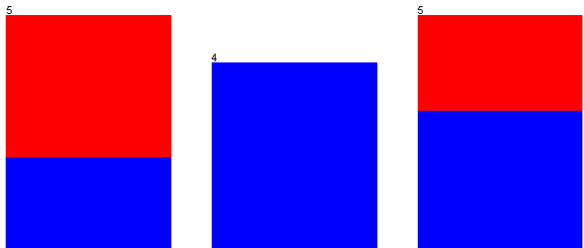
**Selected**

Left-click to edit properties for this object, right-click/Alt+Shift+left-click for menu

Name: outlook Missing: 0 (0%) Distinct: 3 Type: Nominal Unique: 0 (0%)

No.	Label	Count	Weight
1	sunny	5	5.0
2	overcast	4	4.0
3	rainy	5	5.0

Class: play (Nom) Visualize All

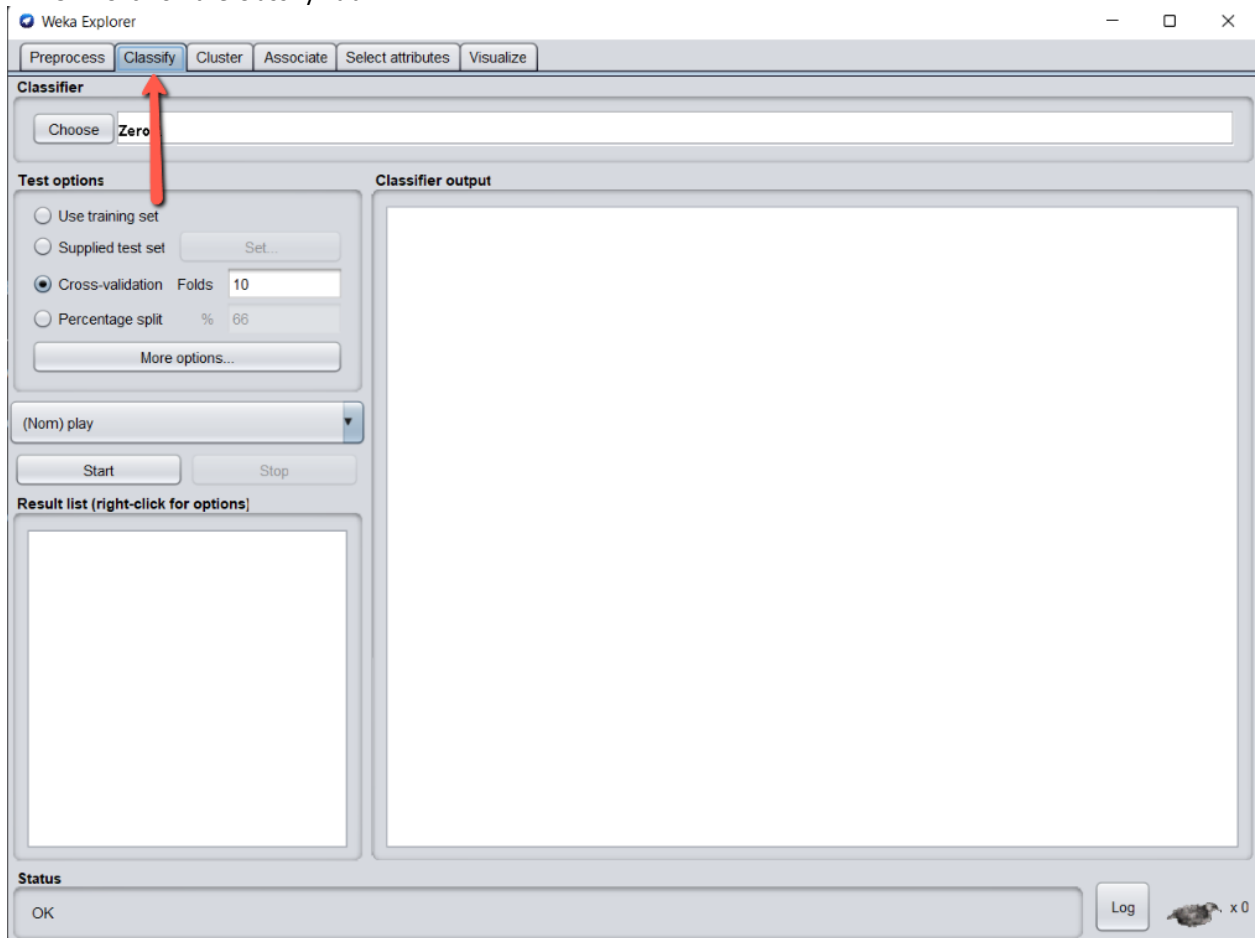


5 4 5

**Status**

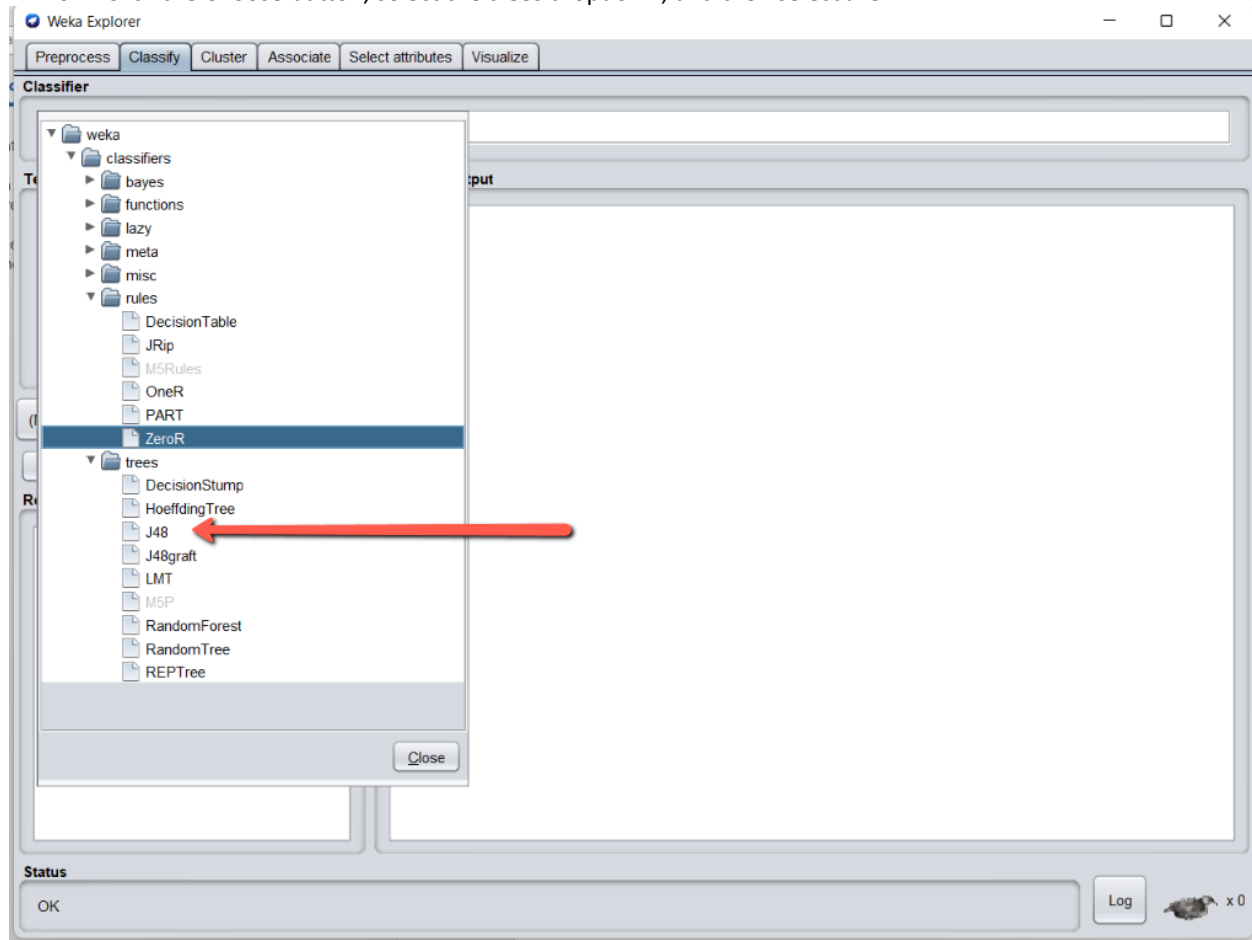
OK Log x 0

5. Click on the Classify Tab





6. Click the Choose button, select the trees dropdown, and then select J48



7. Make sure class\_label is selected (right above the Start button)

8. Press the Start button

9. In the results pane, right click a result, and select Visualize tree

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

☐ Use training set

☐ Supplied test set Set...

☒ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Nom) play

Start Stop

Result list (right-click for options)

08:23:13 - trees J48

- View in main window
- View in separate window
- Save result buffer
- Delete result buffer(s)
- Load model
- Save model
- Re-evaluate model on current test set
- Re-apply this model's configuration
- Visualize classifier errors
- Visualize tree
- Visualize margin curve
- Visualize threshold curve
- Cost/Benefit analysis
- Visualize cost curve

Classifier output

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	7	50	%
Incorrectly Classified Instances	7	50	%
Kappa statistic	-0.0426		
Mean absolute error	0.4167		
Root mean squared error	0.5984		
Relative absolute error	87.5	%	
Root relative squared error	121.2987	%	
Total Number of Instances	14		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	FRC Area	Cl
ye	0.556	0.600	0.625	0.556	0.588	-0.043	0.633	0.758	ye
no	0.400	0.444	0.333	0.400	0.364	-0.043	0.633	0.457	no
	0.500	0.544	0.521	0.500	0.508	-0.043	0.633	0.650	

Matrix ===

psifi

Status

OK

Log

x 0