REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM USING IoT

Submitted by

KARTHICK	**(713519CEIT013)**
SUBRAMANI	**(713519CEIT040)**
AKSHAYA	**(713519CEIT002)**
POOJA RANI	**(713519CEIT026)**

BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY

Performance Testing

Team ID	PNT2022TMID07826
Project Name	Real-time river water quality monitoring and control system

Model Performance

Classification Accuracy

Classification accuracy is the number of correct predictions made as a ratio of all predictions made.

This is the most common evaluation metric for classification problems, it is also the most misused. It is really only suitable when there are an equal number of observations in each class (which is rarely the case) and that all predictions and prediction errors are equally important, which is often not the case.

```
#include <ESP8266WiFi.h>
#include "DHT.h"
#include <ArduinoJson.h>
#include <PubSubClient.h>
// Watson IoT connection details
#define MQTT_HOST
"xpb9eu.messaging.internetofthings.ibmcloud.com"
//Organization
ID.messaging.internetofthings.ibmcloud.com
//change 3xr4l4
#define MQTT_PORT 1883
#define MQTT_DEVICEID "d:xpb9eu:ESP8266:dev1"
//d:Organization ID:Device Type:Device ID
//change 3xr4l4
#define MQTT_USER "use-token-auth"
#define MQTT_TOKEN "karthikproject" // change your
auth_id :
#define MQTT_TOPIC "iot-2/evt/status/fmt/json"
#define MQTT_TOPIC_DISPLAY "iot   2/cmd/display/fmt/json"
```

Note: given the stochastic nature of the algorithm or evaluation procedure, or differences in numerical precision. Consider running the example a few times and compare the average outcome.

This can be converted into a percentage by multiplying the value by 100, giving an accuracy score of approximately 77% accurate.

2. Log Loss

Logistic loss (or log loss) is a performance metric for evaluating the predictions of probabilities of membership to a given class.

The scalar probability between 0 and 1 can be seen as a measure of confidence for a prediction by an algorithm. Predictions that are correct or incorrect are rewarded or punished proportionally to the confidence of the prediction.

Below is an example of calculating log loss for Logistic regression predictions on the Pima Indians onset of diabetes dataset.

```
 // Add GPIO pins used to connect devices
#define DHT_PIN 2 // GPIO pin the data line of the DHT
sensor is connected to
// Specify DHT11 (Blue) or DHT22 (White) sensor
#define DHTTYPE DHT11
// Add WiFi connection information
char ssid[] = "karthick"; // your network SSID (name)
char pass[] = "87654321"; // your network password
DHT dht(DHT_PIN, DHTTYPE);
void sendSMS(String msg)
{
 Serial.print("AT"); //Start Configuring GSM Module
 delay(1000); //One second delay
 Serial.println();
 Serial.println("AT+CMGF=1");
 delay(1000);
 Serial.println("AT+CMGS=\"+916385808140\"\r");
 delay(1000);
 Serial.println(msg);
 delay(100);
 Serial.println((char)26);
 delay(1000);
}
void setup() {
// Start serial console
 Serial.begin(115200);
 Serial.setTimeout(2000);
 while (!Serial) { }
 Serial.println();
 Serial.println("ESP8266 IBM Cloud Application");
 // Start WiFi connection
 WiFi.mode(WIFI_STA);
 WiFi.begin(ssid, pass);
 while (WiFi.status() != WL_CONNECTED) {
 delay(500);
 Serial.print(".");
 }
 Serial.println("");
 Serial.println("WiFi Connected");
 // Start connected devices
 dht.begin();
```

```
  // MQTT objects
void callback(char* topic, byte* payload, unsigned int
length);
WiFiClient wifiClient;
PubSubClient mqtt(MQTT_HOST, MQTT_PORT,
callback, wifiClient);
// variables to hold data
StaticJsonDocument<100> jsonDoc;
JsonObject payload = jsonDoc.to<JsonObject>();
JsonObject status = payload.createNestedObject("d");
static char msg[50];
float h = 0.0;
float t = 0.0;
void callback(char* topic, byte* payload, unsigned int
length) {
 // handle message arrived
 Serial.print("Message arrived [");
 Serial.print(topic);
 Serial.print("] : ");
 payload[length] = 0; // ensure valid content is zero
terminated so can treat as c-string
 Serial.println((char *)payload);
}
```

Smaller log loss is better with 0 representing a perfect log loss.

Note: given the stochastic nature of the algorithm or evaluation procedure, or differences in numerical precision. Consider running the example a few times and compare the average outcome.

As mentioned above, the measure is inverted to be ascending when using **AD**

3. Area Under ROC Curve

Area Under ROC Curve (or ROC AUC for short) is a performance metric for binary classification problems.

The AUC represents a model's ability to discriminate between positive and negative classes. An area of 1.0 represents a model that made all predictions perfectly. An area of 0.5 represents a model as good as random.

A ROC Curve is a plot of the true positive rate and the false positive rate for a given set of probability predictions at different thresholds used to map the probabilities to class labels. The area under the curve is then the approximate integral under the ROC Curve.

The example below provides a demonstration of calculating AUC.

Note: given the stochastic nature of the algorithm or evaluation procedure, or differences in numerical precision. Consider running the example a few times and compare the average outcome.

You can see the the AUC is relatively close to 1 and greater than 0.5, suggesting some skill in the predictions.

4. Confusion Matrix

The confusion matrix is a handy presentation of the accuracy of a model with two or more classes.

The table presents predictions on the x-axis and accuracy outcomes on the y-axis. The cells of the table are the number of predictions made by a machine learning algorithm.

For example, a machine learning algorithm can predict 0 or 1 and each prediction may actually have been a 0 or 1. Predictions for 0 that were actually 0 appear in the cell for prediction=0 and actual=0, whereas predictions for 0 that were actually 1 appear in the cell for prediction = 0 and actual=1. And so on.

```
// Check if any reads failed and exit early (to try
again).
 if (sensorValue<50) {
 Serial.println("WATER POLLUTED");
 sendSMS("WATER POLLUTED");
 delay(3000);
 }
 if (isnan(h) || isnan(t)) {
 Serial.println("Failed to read from DHT sensor!");
 } else {
 // Send data to Watson IoT Platform
 status["temp"] = t;
 status["humidity"] = h;
 status["TURBIDITY"] = sensorValue;
 serializeJson(jsonDoc, msg, 50);
 Serial.println(msg);
 if (!mqtt.publish(MQTT_TOPIC, msg)) {
 Serial.println("MQTT Publish failed");
 }
 }
 // Pause - but keep polling MQTT for incoming
messages
 for (int i = 0; i < 10; i++) {
 mqtt.loop();
 delay(1000);
 }
}
```

```
void loop() {
 mqtt.loop();
 while (!mqtt.connected()) {
 Serial.print("Attempting MQTT connection...");
 // Attempt to connect
 if (mqtt.connect(MQTT_DEVICEID, MQTT_USER,
MQTT_TOKEN)) {
 Serial.println("MQTT Connected");
 mqtt.subscribe(MQTT_TOPIC_DISPLAY);
 mqtt.loop();
 } else {
 Serial.println("MQTT Failed to connect!");
 delay(5000);
 }
 }
```

on a test set.

Note: Your results may vary given the stochastic nature of the algorithm or evaluation procedure, or differences in numerical precision. Consider running the example a few times and compare the average outcome.

Although the array is printed without headings, you can see that the majority of the predictions fall on the diagonal line of the matrix (which are correct predictions).

5. Classification Report

Scikit-learn does provide a convenience report when working on classification problems to give you a quick idea of the accuracy of a model using a number of measures.

The *classification_report()* function displays the precision, recall, f1-score and support for each class.

The example below demonstrates the report on the binary classification problem.

```
     // Connect to MQTT - IBM Watson IoT Platform
 if (mqtt.connect(MQTT_DEVICEID, MQTT_USER,
MQTT_TOKEN)) {
 Serial.println("MQTT Connected");
 mqtt.subscribe(MQTT_TOPIC_DISPLAY);
 } else {
 Serial.println("MQTT Failed to connect!");
 ESP.reset();
 }
}
int sensorValue = analogRead(A0);
 Serial.println(sensorValue);
 delay(1000);
 h = dht.readHumidity();
 t = dht.readTemperature(); // uncomment this line for
centigrade
```

```
 Serial.print("Current humidity = ");
 Serial.print(h);
 Serial.print("% ");
 Serial.print("temperature = ");
 Serial.print(t);
 Serial.println("C ");
 // t = dht.readTemperature(true); // uncomment
this line for Fahrenheit
```

Note: given the stochastic nature of the algorithm or evaluation procedure, or differences in numerical precision. Consider running the example a few times and compare the average outcome.