**Báo cáo bài tập lớn**

**Đề tài: Phát hiện tấn công web-base dựa trên điểm bất thường.**

1. **Tập dữ liệu đào tạo.**
2. **Tập dữ liệu thô.**

Ví dụ cho 1 trong n dữ liệu thô chúng ta thu thập được từ website.

GET http://localhost:8080/tienda1/publico/anadir.jsp?id=2&nombre=Jam%F3n+Ib%E9rico&precio=85&cantidad=%27%3B+DROP+TABLE+usuarios%3B+SELECT+\*+FROM+datos+WHERE+nombre+LIKE+%27%25&B1=A%F1adir+al+carrito HTTP/1.1

User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8 (like Gecko)

Pragma: no-cache

Cache-Control: no-cache

Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5

Accept-Encoding: x-gzip, x-deflate, gzip, deflate

Accept-Charset: utf-8, utf-8;q=0.5, \*;q=0.5

Accept-Language: en

Host: localhost:8080

Cookie: JSESSIONID=B92A8B48B9008CD29F622A994E0F650D

Connection: close

1. **Tập dữ liệu để đào tạo.**

Sau khi thu thập dữ liệu bước tiếp theo phân tích dữ liệu. Dữ liệu sau phân tích được lưu lại dưới file .csv và lưu theo các trường sau:

**id**: rời rạc, số dòng yêu cầu HTTP thô

**method**: rời rạc, phương pháp được thực hiện trên tài nguyên được xác định bởi Yêu cầu-URI

**length\_path**: rời rạc, chiều dài đường dẫn của một URI

**printable\_characters\_ratio\_path**: liên tục, tỷ lệ phần trăm các ký tự có thể in trong đường dẫn

**non\_printable\_characters\_ratio\_path**: liên tục, phần trăm các ký tự không thể in trong đường dẫn

**letter\_ratio\_path**: liên tục, tỷ lệ chữ cái trong đường dẫn

**digit\_ratio\_path**: liên tục, tỷ lệ chữ số trong đường dẫn

**symbol\_ratio\_path**: liên tục, phần trăm symbol trong đường dẫn

**num\_segment**: rời rạc, số đoạn path được phân tách bằng dấu gạch chéo "/"

**is\_file**: rời rạc, chỉ ra nếu URI là một tập tin

**file\_extension**: rời rạc, mở rộng tệp

**num\_parameters**: rời rạc, số tham số trong truy vấn

**length\_query**: rời rạc, chiều dài truy vấn

**printable\_characters\_ratio\_query**: liên tục, phần trăm các ký tự có thể in trong truy vấn

**non\_printable\_characters\_ratio\_query**: liên tục,phần trăm các ký tự không thể in trong truy vấn

**letter\_ratio\_query**: liên tục, tỷ lệ chữ cái trong truy vấn

**digit\_ratio\_query**: liên tục, tỷ lệ chữ số trong truy vấn

**symbol\_ratio\_query**: liên tục, phần trăm biểu tượng trong truy vấn

**num\_headers**: rời rạc, number of headers

**standard\_headers\_ratio**: liên tục, phần trăm tiêu đề chuẩn trong yêu cầu HTTP

**non\_standard\_headers\_ratio**: liên tục, phần trăm tiêu đề không chuẩn trong yêu cầu HTTP

**length\_header*\_${header-name}:*** rời rạc, độ dài của giá trị của tiêu đề

**printable\_characters\_ratio\_header*\_${header-name}***: liên tục,phần trăm các ký tự có thể in trong header***\_${header-name}***

**non\_printable\_characters\_ratio\_header*\_${header-name}***: liên tục,phần trăm các ký tự không thể in trong ***header\_${header-name}***

**letter\_ratio\_header*\_${header-name}***: liên tục, tỷ lệ chữ cái trong header***\_${header-name}***

**digit\_ratio\_header*\_${header-name}***: liên tục, tỷ lệ chữ số trong header***\_${header-name}***

**symbol\_ratio\_header*\_${header-name}***: liên tục, phần trăm biểu tượng trong header***\_${header-name}***

**is\_standard\_header*\_${header-name}***: rời rạc, cho biết header***\_${header-name}*** có phải là tiêu chuẩn hay không

**is\_persistent\_connection**: rời rạc, cho biết nếu là kết nối liên tục sử dụng giá trị của tiêu đề "Kết nối"

**content\_type**: rời rạc, Loại MIME của nội dung yêu cầu HTTP

**length\_body**: rời rạc, length of body

**printable\_characters\_ratio\_body**: liên tục, phần trăm các ký tự có thể in trong nội dung

**non\_printable\_characters\_ratio\_body**: liên tục, phần trăm các ký tự không thể in trong nội dung

**letter\_ratio\_body**: liên tục, percentage of letters in body/ tỉ lệ phần trăm của chữ cái trong body

**digit\_ratio\_body**: liên tục, percentage of digits in body / tỉ lệ phần trăng của chữ số trong body

**symbol\_ratio\_body**: liên tục, percentage of symbols in body / tỉ lệ phần trăm biểu tượng trong body

**num\_line**: rời rạc, number of line in body

**num\_word**: rời rạc, number of word in body

**label**: : rời rạc, loại yêu cầu HTTP.

**Trong đó**:   
**-** val **StandardHeaders** = **Seq**("Accept", "Accept-Charset", "Accept-Datetime",

"Accept-Encoding", "Accept-Language", "Access-Control-Request-Method",

"Access-Control-Request-Headers", "Authorization", "Cache-Control", "Connection",

"Content-Length", "Content-MD5", "Content-Type", "Cookie", "Date", "Expect", "From", "Host",

"If-Match", "If-Modified-Since", "If-None-Match", "If-Range", "If-Unmodified-Since",

"Max-Forwards", "Origin", "Pragma", "Proxy-Authorization", "Range", "Referer", "TE",

"User-Agent", "Upgrade", "Via", "Warning"

);

- ***${header-name}*** là một phần tử trong **StandardHeaders** .

1. **Phân tích dữ liệu.**
2. **Toolbox cho thống kê cơ bản**

* / \*\*

     \* Lấy tổng số ký tự không in được của **sequence** của **String**

     \*

     \* @param seq **sequence** của **String**

     \* @return tổng số ký tự không in được của **sequence** của **String**

     \* /

   def nonPrintableCharRatio (seq: Seq [String]): Double

* / \*\*

     \* Đếm số ký tự không thể in trong **String**

     \*

     \* @param str **String**

     \* @ trả về số ký tự không thể in trong **String**

     \* /

   def nonPrintableCharCount (str: String): Int

* / \*\*

     \* Đếm số ký tự có thể in trong **String**

     \*

     \* @param str **String**

     \* @ trả về số ký tự có thể in trong **String**

     \* /

def printableCharCount(str: String): Int

* / \*\*

     \* Gets tổng số ký tự in được của **sequence** của **String**

     \*

     \* @param seq **sequence** của chuỗi

     \* @return tổng số ký tự có thể in của **sequence** của **String**

     \* /

def printableCharRatio(seq: Seq[String]): Double

* / \*\*

     \* Gets total symbols ratio of sequence of string

     \*

     \* @param seq **sequence** của **String**

     \* @quay tổng tỷ lệ ký hiệu của **sequence** của **String**

     \* /

   def symbolRatio (seq: Seq [String]): Double

* / \*\*

     \* Đếm số biểu tượng trong **String**

     \*

     \* @param str **String**

     \* @quay số biểu tượng trong **String**

     \* /

   def symbolCount (str: String): Int

* / \*\*

     \* Nhận được tổng số tỷ lệ chữ cái của **Sequence** của **String**

     \*

     \* @param seq **Sequence** của **String**

     \* @ trả lại tổng số tỷ lệ chữ cái của **Sequence** của **String**

     \* /

   def letterRatio (seq: Seq [String]): Double

* / \*\*

     \* Đếm số chữ số trong chuỗi

     \*

     \* @param str chuỗi

     \* @quay số chữ số trong chuỗi

     \* /

   def digitCount (str: String): Int

1. **Phân tích RawHttpRequest**

* / \*\*

     \* Nhận giá trị của một Tiêu đề

     \*

     \* @param headerName tên của tiêu đề

     \* @ return giá trị của Tiêu đề, "no\_content" nếu giá trị trống

     \* /

   def getHeaderValue (headerName: String): String

* / \*\*

     \* Nhận tỷ lệ Tiêu đề không chuẩn trong tiêu đề của yêu cầu HTTP thô

     \*

     \* @ return tỷ lệ Tiêu đề không chuẩn trong tiêu đề của yêu cầu HTTP thô

     \* /

   def getNonStandardHeaderRatio: Double

* / \*\*

     \* Đạt được tỷ lệ Tiêu đề chuẩn trong tiêu đề của yêu cầu HTTP thô

     \*

     \* @retturn Tỷ lệ tiêu đề chuẩn trong tiêu đề của yêu cầu HTTP thô

     \* /

   def getStandardHeaderRatio: Double

* / \*\*

     \* Đại diện cho RawHttpRequest ở định dạng chuỗi

     \*

     \* @return RawHttpRequest ở định dạng chuỗi

     \* /

  override def toString: String

* / \*\*

     \* Thay thế các Tiêu đề hiện có trong danh sách Tiêu đề chuẩn

     \*

     \* @return danh sách Tiêu đề chuẩn với Tiêu đề yêu cầu HTTP

     \* /

   private def replaceExistingHeaders: Seq [Header]

* / \*\*

     \* Phân tích cú pháp truy vấn của yêu cầu HTTP thô

     \*

     \* @param truy vấn toàn bộ dòng truy vấn

     \* @return danh sách các tham số trong truy vấn

     \* /

   private par parseQuery (query: String): Seq[Parameter]

* / \*\*

     \* Phân tích tham số của yêu cầu HTTP thô

     \*

     \* @param tham số toàn bộ dòng của tham số

     \*@return Tên và thông số của tiêu đề

     \* /

private def parseParameter(parameter: String): (String, String)

1. **Tuning Model**

Một nhiệm vụ quan trọng trong ML là lựa chọn model, hoặc sử dụng dữ liệu để tìm mô hình hoặc thông số tốt nhất cho một tác vụ cụ thể. Điểu này được gọi *tuning* . Tuning có thể được thực hiện cho các *Estimator* như LogisticRegression, hoặc cho toàn bộ *pipeline* bao gồm nhiều thuật toán và các bước khác. Người dung có thể điều chỉnh toàn bộ *pipeline* cùng một lúc thay vì điều chỉnh từng phần tử trong *pipeline* một cách riêng biệt.

Mllib hỗ trợ lựa chọn model bằng cách sử dụng các công cụ như CrossValidator và TrainValidationSplit. Những công cụ này yêu cầu các mục sau:

* Estimator: thuật toán hoặc Pipeline để điều chỉnh.
* Thiết lập ParamsMap: các tham số để chọn, đôi khi được gọi là “parameter grid” để tìm kiểm.
* Evaluator: Chỉ số để đo lường mức độ phù hợp của model được thực hiện với dữ liệu thử nghiệm

Với một mức độ cao hơn, các công cụ chọn mẫu hoạt động như sau.

* Chia dữ liệu đầu vào thành hai phần riêng biệt: training và test
* Với mỗi cặp (training,test) , chúng lặp lại qua bộ PramsMapS
  + Đối với mỗi ParamMapS, chúng fit với Estimator  bằng cách sử dụng các tham số đó, lấy mẫu được chuẩn bị và đánh giá hiệu suất của mô hình bằng Evaluator.
* Chúng chọn model được đào tạo bởi bộ thông số hoạt động tốt nhất

Evaluator có thể là một:

* RegressionEvaluator cho vấn đề regression
* BinaryClassificationEvaluator cho vấn đề binary data.
* MulticlassClassificationEvaluator cho vấn đề multiclass

Trong phạm vi bài toán chúng ta sử dụng TrainValidationSplit. TrainValidationSplit tạo ra một cặp dữ liệu (training, test) duy nhất, chỉ đánh giá kết hợp các tham số một lần.

* Ưu điểm đó là ít tốn kém.
* Nhược điểm đó là không tạo ra kết quả đáng tin cậy khi tập dữ liệu đào tạo không đủ lớn.

1. **Evaluation metrics**

mllib đi kèm với một số thuật toán học máy có thể được sử dụng để học hỏi và đưa ra các dự đoán về dữ liệu. Khi các thuật toán này được áp dụng để xây dựng các mô hình học máy, cần phải đánh giá hiệu suất của mô hình trên một số tiêu chí, phụ thuộc vào ứng dụng và các yêu cầu của nó mllib cũng cung cấp một bộ chỉ số cho mục đích đánh giá hiệu suất của các mô hình học máy.

* + - * Đánh giá mô hình phân loại

Mặc dù có rất nhiều thuật toán phân loại khác nhau, việc đánh giá các mô hình phân loại đều chia sẻ các tài nguyên tương tự nhau. Trong vấn đề phân loại được giám sát, có tồn tại một đầu ra thực và một đầu ra dự đoán do model tạo ra cho mỗi điểm dữ liệu. Vì vậy, kết quả cho mỗi điểm dữ liệu có thể dược gán cho một trong bốn loại:

* True Positive(TP): nhãn là positive và dự đoán cũng positive.
* True Negative(TN): nhãn là negative và dự đoán cũng negative
* False Positive(FP): nhãn là negative nhưng nhãn là positive
* False Negative(FN): nhãn là positive nhưng nhãn là negative

Bốn số này là các khối xây dựng cho hầu hết các chỉ số đánh giá trình phân loại. Điểm cơ bản khi xem xét đánh giá trình phân loại là độ chính xác thuần túy (tức là dự đoán chính xác hoặc không đúng) thường không phải là chỉ số tốt. Lý do cho điều này là do một tập dữ liệu có thể bị mất cân bằng cao.

* + - * + Phân loại nhị phân

Threshold tuning: nó có nghĩa là nhiều mô hình phân loại thực sự tạo ra một điểm cho mỗi lớp, tron đó điểm số cao hơn cho thấy khả năng cao hơn. Trong trường hợp nhị phân, mô hình có thể xuất ra xác suất cho mỗi lớp P(Y=1|X) và P(Y=0|X) . Thay vì chỉ đơn giản lấy xác suất cao hơn, có thể có một số trường hợp mô hình có thể cần được điều chỉnh để nó chỉ dự đoán một lớp khi xác suất rất cao (ví dụ: chỉ chặn giao dịch thẻ tín dụng nếu mô hình dự đoán gian lận với> 90 % xác suất). Do đó, có một ngưỡng dự đoán xác định những gì lớp dự đoán sẽ dựa trên xác suất mà mô hình đầu ra. Việc điều chỉnh ngưỡng dự đoán sẽ thay đổi độ chính xác và thu hồi của mô hình và là một phần quan trọng của tối ưu hóa mô hình. Để hình dung độ chính xác, số lần gọi lại và các chỉ số khác thay đổi như một hàm của ngưỡng, thực tế phổ biến là vẽ các chỉ số cạnh tranh với nhau, được tham số hóa theo ngưỡng.

