BÁO CÁO THỰC HÀNH

**Môn học: IDPS**

**Lab 5: Học máy trong IDS**

*GVHD: Đỗ Hoàng Hiển*

1. **THÔNG TIN CHUNG:**

Lớp: NT204.N21.ATCL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Hoàng Văn Anh Đức | 20520890 | 20520890@gm.uit.edu.vn |
| 2 | Nguyễn Mạnh Cường | 20520421 | 20520421@gm.uit.edu.vn |
| 3 | Lê Quang Minh | 20520245 | 20520245@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Công việc** | **Thành viên thực hiện** | **Kết quả tự đánh giá** |
| 1 | All | Đức + cường | 100% |

**BÁO CÁO CHI TIẾT**



TÌM HIỂU VỀ TẬP DỮ LIỆU KDD CUP 1999

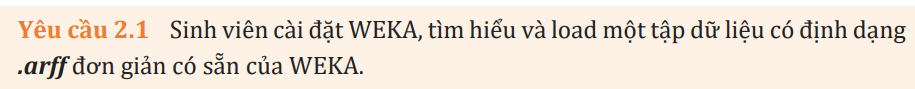
Dữ liệu trong bộ dữ liệu KDD Cup 1999 là lựu lựợng mạng đã đựợc thu thập, phân tích, xử lý để lấy các thuộc tính và từ đó gán nhãn tựơng ứng với loại tấn công hoặc dữ liệu bình thựờng.

1. **Số nhóm tấn công: 4**
   1. Kể tên các nhóm tấn công:
      1. **Tấn công DoS (Denial of Service):** 
         1. 'smurf.',
         2. 'teardrop.',
         3. 'pod.',
         4. 'land.',
         5. 'neptune.'
      2. **Tấn công R2L (Remote-to-Local):**
         1. 'guess\_passwd.',
         2. 'warezclient.',
         3. 'ftp\_write.',
         4. 'imap.',
         5. 'phf.',
         6. 'multihop.'
         7. 'spy.'
         8. 'warezmaster.',
         9. back.
      3. **Tấn công U2R (User-to-Root):**
         1. 'buffer\_overflow.',
         2. 'loadmodule.',
         3. 'perl.',
         4. 'rootkit.'
      4. **Tấn công Probing:**
         1. 'satan.',
         2. 'ipsweep.',
         3. 'portsweep.',
         4. 'nmap.'
2. **Số kiểu tấn công : 22 kiểu**
   1. Kể tên các kiểu tấn công đựợc gán nhãn :  
       'buffer\_overflow.', 'loadmodule.', 'perl.', 'neptune.', 'smurf.', 'guess\_passwd.', 'pod.', 'teardrop.', 'portsweep.', 'ipsweep.', 'land.', 'ftp\_write.', 'back.', 'imap.', 'satan.', 'phf.', 'nmap.', 'multihop.', 'warezmaster.', 'warezclient.', 'spy.', 'rootkit.'
3. **Mỗi instance trong tập dữ liệu KDD Cup 1999 bao gồm 40 thuộc tính , cụ thể gồm các thuộc tính :**

* 'duration','protocol\_type','service','flag','src\_bytes','dst\_bytes','land','wrong\_fragment','urgent','hot', 'num\_failed\_logins','logged\_in','num\_compromised','root\_shell', 'su\_attempted','num\_root','num\_file\_creations','num\_shells', 'num\_access\_files','num\_outbound\_cmds','is\_host\_login','is\_guest\_login','count','srv\_count','serror\_rate','srv\_serror\_rate', 'rerror\_rate','srv\_rerror\_rate','same\_srv\_rate','diff\_srv\_rate','srv\_diff\_host\_rate','dst\_host\_count','dst\_host\_srv\_count', 'dst\_host\_same\_srv\_rate','dst\_host\_diff\_srv\_rate','dst\_host\_same\_src\_port\_rate','dst\_host\_srv\_diff\_host\_rate','dst\_host\_serror\_rate', 'dst\_host\_srv\_serror\_rate','dst\_host\_rerror\_rate','dst\_host\_srv\_rerror\_rate'

A screenshot of a computer

Description automatically generated with medium confidence



A screenshot of a computer

Description automatically generated

* Attribute : 5 cột (features)
* Instance : 150 hàng (records)
* Sum of weights : tổng của các trọng số được gán cho từng mẫu dữ liệu trong tập dữ liệu.
* "Mean" (trung bình) - tổng giá trị của thuộc tính trong tập dữ liệu và chia cho số lượng mẫu dữ liệu
* "Stdev" (độ lệch chuẩn)- đại lượng đo lường mức độ biến thiên của dữ liệu từ giá trị trung bình
  + Độ lệch chuẩn là căn bậc hai của phương sai (variance).
  + Nó cho biết độ phân tán của các giá trị quan sát so với giá trị trung bình. Giá trị Stdev càng lớn, độ biến động của dữ liệu càng cao.
* Maximum : max của giá trị trong cột
* Min : min value trong cột



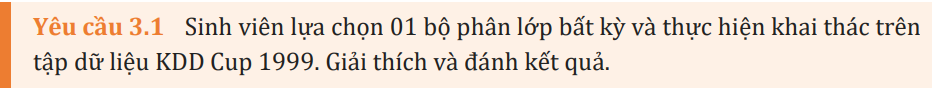
A screenshot of a computer

Description automatically generated

A screenshot of a computer

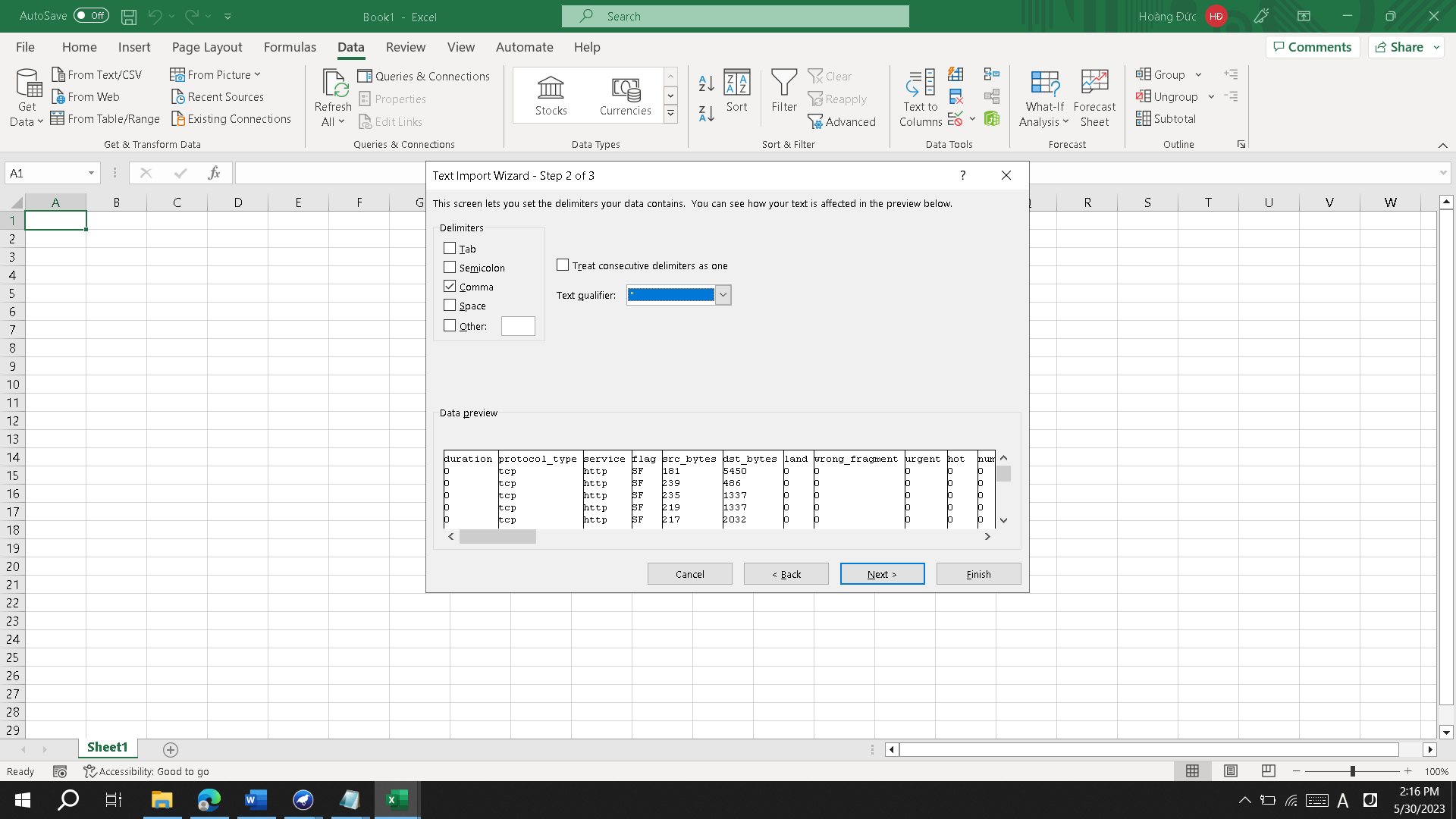
Description automatically generated

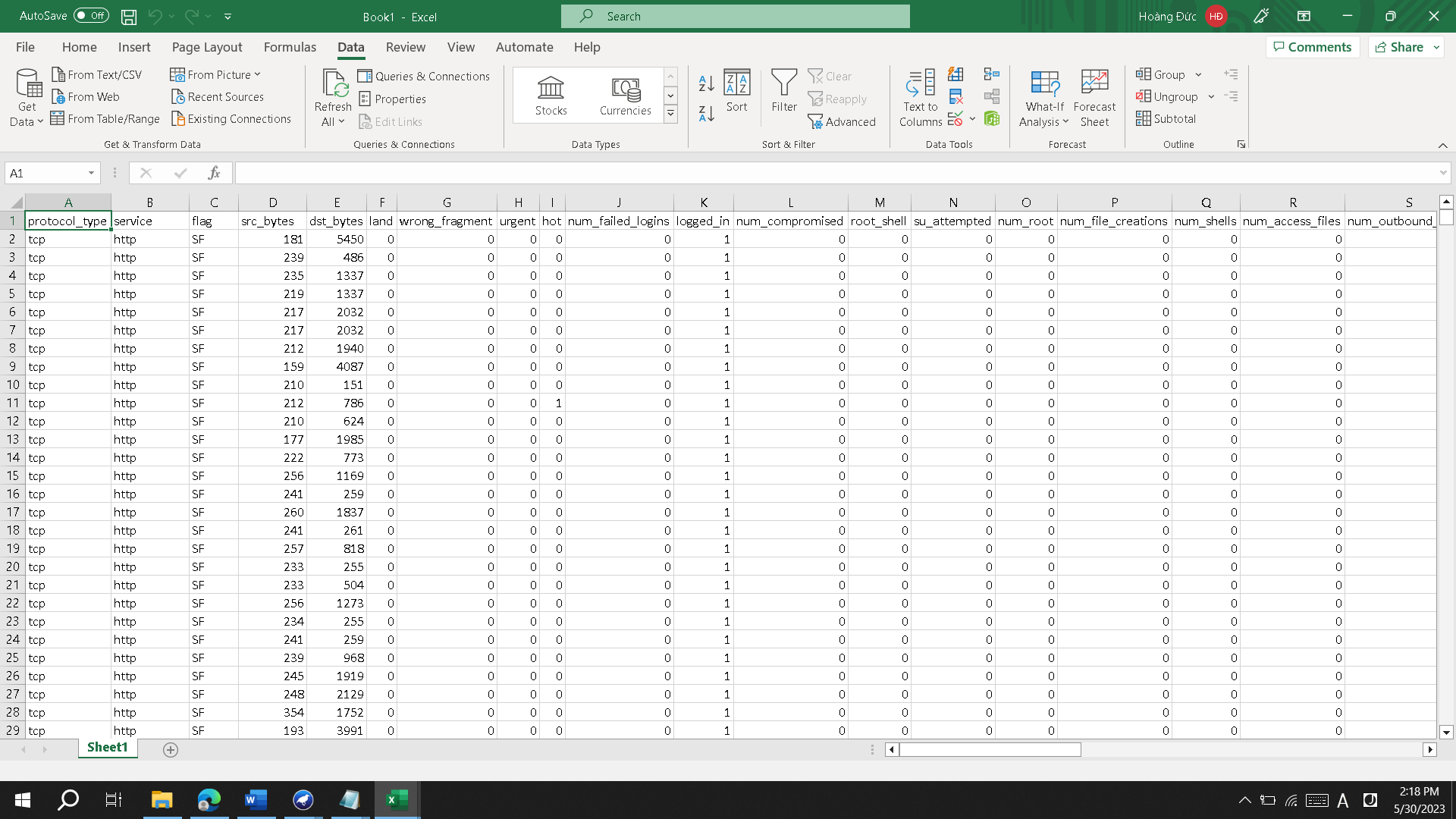
* **Size of the tree (Kích thước của cây):** Đây là số lượng nút (node) trong cây quyết định sau quá trình xây dựng. Trong trường hợp này, cây có 9 nút.
* **Time taken to build model (Thời gian xây dựng mô hình):** Đây là thời gian mất để xây dựng mô hình, được tính bằng giây. Trong trường hợp này, thời gian xây dựng mô hình là 0.01 giây.
* **Stratified cross-validation (Kiểm định chéo theo phân lớp):** Đây là phương pháp kiểm định mô hình được sử dụng để đánh giá hiệu suất của mô hình trên tập dữ liệu. Kết quả của quá trình kiểm định chéo được hiển thị sau đó.
* **Correctly Classified Instances (Số lượng mẫu được phân loại chính xác):** Đây là số lượng mẫu dữ liệu được phân loại đúng bởi mô hình. Trong trường hợp này, có 144 mẫu được phân loại đúng (96%).
* **Incorrectly Classified Instances (Số lượng mẫu được phân loại sai):** Đây là số lượng mẫu dữ liệu bị phân loại sai bởi mô hình. Trong trường hợp này, có 6 mẫu được phân loại sai (4%).
* **Kappa statistic (Thống kê Kappa):** Đây là một thống kê sử dụng để đo độ đồng nhất giữa các dự đoán của mô hình và nhãn thực tế. Giá trị Kappa gần 1 cho thấy mô hình có độ chính xác cao.
* **Mean absolute error (Sai số trung bình tuyệt đối):** Đây là giá trị trung bình của sai số tuyệt đối giữa dự đoán và nhãn thực tế.
* **Root mean squared error (Sai số bình phương trung bình căn):** Đây là sai số trung bình của dự đoán so với nhãn thực tế, tính bằng cách lấy căn bậc hai của sai số bình phương trung bình.
* **Relative absolute error (Sai số tuyệt đối tương đối):** Đây là tỷ lệ giữa sai số tuyệt đối trung bình và giá trị trung bình của nhãn thực tế.
* **Root relative squared error (Sai số bình phương tương đối căn):** Đây là tỷ lệ giữa sai số bình phương trung bình căn và giá trị trung bình của nhãn thực tế.
* **Total Number of Instances (Tổng số mẫu dữ liệu):** Đây là tổng số mẫu dữ liệu trong tập dữ liệu. Trong trường hợp này, có tổng cộng 150 mẫu.
* **Detailed Accuracy By Class (Độ chính xác chi tiết theo lớp):** Đây là đánh giá chi tiết về độ chính xác, độ nhạy (recall), độ chính xác dự đoán (precision), F-Measure, và các độ đo khác theo từng lớp của bộ dữ liệu. Đây là để xác định hiệu suất của mô hình trên từng lớp riêng biệt.
* **Confusion Matrix (Ma trận nhầm lẫn):** Đây là confusion matrix, hiển thị số lượng mẫu thực tế được phân loại vào từng lớp, so sánh với lớp dự đoán của mô hình. Nó giúp đánh giá sự khác biệt giữa dự đoán và nhãn thực tế.
* **Precision (Độ chính xác):** Precision đo lường tỷ lệ các dự đoán đúng về một lớp cụ thể so với tổng số các dự đoán dương (positive) cho lớp đó.
* **Recall (Độ phủ):** Recall đo lường tỷ lệ các dự đoán đúng về một lớp cụ thể so với tổng số các mẫu thực tế dương thuộc lớp đó.
* **F1-score:** F1-score là một chỉ số kết hợp giữa precision và recall để đánh giá hiệu suất tổng thể của mô hình.



A screenshot of a computer

Description automatically generated





Reading success

A screenshot of a computer

Description automatically generated

A diagram of a tree

Description automatically generated with medium confidence

**Trình bày nguyên lý hoạt động củà bộ phân lớp đã chọn. Giải thích lý do chọn bộ phân lớp này.**

**Nguyên lý hoạt động** :

* Random Forest là một thuật toán học máy **sử dụng nhiều cây quyết định** để tạo ra **dự đoán** cuối cùng.
* Quá trình hoạt động của Random Forest bao gồm **tạo ra các tập con ngẫu nhiên** **từ dữ liệu huấn luyện**, **xây dựng** các **cây quyết định** độc lập **trên các tập con** này, và **kết hợp dự đoán** của các cây **thông qua bỏ phiếu** hoặc kết hợp dự đoán.
* Random Forest giúp **giảm phương sai** và **overfitting**, **tăng tính tổng quát** **và ổn định của mô hình,** và có **khả năng xử lý dữ liệu lớn và các thuộc tính không tuyến tính.**
  + **Vd : 100 mẫu với 10 thuộc tính và 2 lớp. Chúng ta muốn xây dựng một Random Forest gồm 5 cây quyết định.**
    - mỗi cây 50 mẫu , mỗi cây có thể sử dụng 5 thuộc tính ngẫu nhiên
    - 3 cây dự đoán lớp A và 2 cây dự đoán lớp B
      * kết quả cuối cùng sẽ là lớp A với sự đa số.

**Vì sao :**

* Random Forest là một phương pháp ensemble kết hợp nhiều cây quyết định độc lập.
* Nó **sử dụng random feature selection** để g**iảm sự phụ thuộc giữa các cây và nguy cơ overfitting**.
* Bằng cách **sử dụng kỹ thuật Bagging**, nó **lấy mẫu ngẫu nhiên từ tập dữ liệu** và **giảm ảnh hưởng của dữ liệu nhiễu** và ngoại lai.
* Khi thực hiện dự đoán, **Random Forest sử** dụng **cơ chế vote** **từ** tất cả **các cây** để **đưa ra quyết định cuối cùng**. Phương pháp này **tạo ra dự đoán ổn định** và **đáng tin cậy.**

**Giải thích các test option**

* **Cross-validation (numFolds):** Tùy chọn này xác định số lượng fold trong quá trình chia dữ liệu thành các tập train và test.
  + Ví dụ, nếu đặt numFolds = 10, dữ liệu sẽ được chia thành 10 fold, và mô hình sẽ được huấn luyện và đánh giá trên từng fold, lần lượt sử dụng mỗi fold làm tập test và các fold còn lại làm tập train.
* **Percentage split (percents):** Tùy chọn này cho phép bạn chỉ định tỷ lệ phần trăm dữ liệu được sử dụng cho huấn luyện (training) và kiểm tra (testing). Ví dụ, nếu bạn đặt percents = 70, 70% dữ liệu sẽ được sử dụng để huấn luyện và 30% sẽ được sử dụng để kiểm tra.
* **Use training set**: sử dụng tập train để test
* **Supplied test set**: cung cấp tập test

**Giải thích và đánh giá các kết quả thu đựợc**

* **Kappa statistic** có giá trị từ 0 đến 1, với 1 đại diện cho sự hoàn hảo giữa dự đoán và nhãn thực tế. Trong trường hợp này, Kappa statistic là 0.9995, cho thấy sự tương đồng cao giữa dự đoán của mô hình và nhãn thực tế.
* **Mean absolute error**: Đây là trung bình giá trị tuyệt đối của sai số giữa dự đoán và nhãn thực tế.
  + Giá trị càng thấp, chất lượng dự đoán càng tốt.
  + mean absolute error là 0.0001, cho thấy mức độ chính xác cao của mô hình.
* **Root mean squared error**: Đây là căn bậc hai của trung bình của bình phương sai số giữa dự đoán và nhãn thực tế.
  + Giá trị càng thấp, chất lượng dự đoán càng tốt.
  + Trong trường hợp này, root mean squared error là 0.0046, cho thấy mức độ chính xác cao của mô hình.
* **Relative absolute error**: Đây là mean absolute error được chia cho trung bình giá trị tuyệt đối của nhãn thực tế, và được tính toán dưới dạng phần trăm.
  + Trong trường hợp này, relative absolute error là 0.1232%, cho thấy độ chính xác cao của mô hình.
* **Root relative squared error**: Đây là root mean squared error được chia cho trung bình giá trị tuyệt đối

**Câu hỏi của thầy**

**Giải thích tại sao Cross-validation : 10 folds nhưng Total number of instances là 150k thay vì 50k tại vì tổng là 500k instances**

* Thật ra nếu K-folds = 10 thì nó sẽ chia dataset thành 10 phần , mỗi lần train là 1/10 = 50k và test là phần còn lại , nhưng nó lại ra 150k tại vì khi train em chọn option là Percentage Split 70 nhưng khi xem kết quả em lại **bấm lộn** vào phần Cross-validation : 10 nên không thể giải thích tại sao chọn 10 folds mà Total number of instances là 150k
  + 150k là kết quả khi chọn Percentage Split : 70%

A screenshot of a computer

Description automatically generated with medium confidence

**Giải thích cơ chế hoạt động của J48**

* Mô hình J48 là một thuật toán học cây quyết định trong Weka, được dựa trên thuật toán C4.5 (Decision trees). Mô hình J48 sử dụng dữ liệu huấn luyện để xây dựng cây quyết định, một mô hình phân loại có cấu trúc cây dựa trên quyết định logic. Quá trình hoạt động của mô hình J48 được mô tả như sau:
  + **Bước 1:** Xác định node gốc: Ban đầu, toàn bộ tập dữ liệu huấn luyện được sử dụng làm node gốc của cây quyết định.
  + **Bước 2:** Xác định các node con: Dựa trên thuộc tính của dữ liệu, mô hình J48 chọn thuộc tính tốt nhất để phân chia dữ liệu thành các nhánh con. Mô hình sử dụng một số độ đo như độ thông tin (information gain) hoặc tỉ lệ thông tin (gain ratio) để xác định thuộc tính phân chia tốt nhất.
  + **Bước 3**: Xây dựng cây con: Với mỗi nhánh con, quá trình xây dựng cây được lặp lại bằng cách áp dụng các bước trên trên các tập dữ liệu con tương ứng với thuộc tính đã chọn. Quá trình này được lặp lại cho đến khi không còn thuộc tính để phân chia hoặc các nút lá đã đạt được đủ độ tinh khiết.
  + **Bước 4**: Xử lý các nút lá: Khi cây đã được xây dựng, các nút lá được gán nhãn dựa trên lớp phổ biến nhất trong tập dữ liệu huấn luyện tương ứng với nút lá đó.
  + **Bước 5**: Kiểm tra và dự đoán: Sau khi cây quyết định được xây dựng, nó có thể được sử dụng để dự đoán lớp của các dữ liệu mới. Quá trình dự đoán dựa trên quá trình đi xuống cây từ node gốc đến node lá tương ứng với dữ liệu đầu vào và trả về lớp được gán cho nút lá đó.

Mô hình J48 có thể cung cấp cả phân loại (classification) và dự đoán (prediction) dựa trên cây quyết định được xây dựng. Nó cung cấp một cách trực quan và diễn giải để hiểu quyết định và quy luật được học từ dữ liệu.

* + - * Khác với Random forest là nhiều cây quyết định , Decision tree chỉ có 1 và sẽ dựa vào kết quả phân loại của 1 cây để đánh giá hiệu suất

A picture containing text, diagram, screenshot, font

Description automatically generated

(Hình ảnh minh họa 1 cây quyết định)

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)