**1. Giới thiệu**

**1.1 Bối cảnh**

Trong thời đại số hóa hiện nay, vấn đề an ninh mạng ngày càng trở nên quan trọng khi mà thông tin cá nhân và tài sản của người dùng liên tục bị đe dọa. Một trong những cách thức bảo vệ cơ bản nhất là sử dụng mật khẩu để ngăn chặn truy cập trái phép vào tài khoản cá nhân. Tuy nhiên, việc lựa chọn mật khẩu yếu hoặc sử dụng cùng một mật khẩu cho nhiều dịch vụ đã tạo điều kiện thuận lợi cho tội phạm mạng khai thác. Mật khẩu yếu có thể bị tấn công dễ dàng thông qua các phương pháp brute-force hoặc từ điển (dictionary attacks), dẫn đến mất mát dữ liệu và vi phạm an ninh nghiêm trọng.

**1.2 Mục tiêu**

Dự án "Đánh giá độ mạnh của mật khẩu" nhằm áp dụng các phương pháp học máy để xây dựng hệ thống có khả năng tự động đánh giá và phân loại độ mạnh của mật khẩu. Bằng cách phân tích các đặc trưng như độ dài, số lượng ký tự in hoa, ký tự số và các ký tự đặc biệt, hệ thống sẽ giúp người dùng nhận biết mức độ an toàn của mật khẩu họ đang sử dụng, từ đó thúc đẩy họ tạo ra mật khẩu mạnh hơn để bảo vệ tài khoản cá nhân của mình.

**1.3 Cấu trúc báo cáo**

Báo cáo này được chia thành các phần như sau:

* **Cơ sở lý thuyết:** Trình bày các khái niệm về độ mạnh mật khẩu, các phương pháp đánh giá hiện tại và cách thức áp dụng học máy trong lĩnh vực này.
* **Phương pháp nghiên cứu:** Mô tả dữ liệu được sử dụng, quy trình tiền xử lý dữ liệu, các mô hình học máy được triển khai và các đặc trưng được trích xuất từ mật khẩu.
* **Kết quả và thảo luận:** Đánh giá hiệu quả của các mô hình thông qua các chỉ số đo lường và so sánh. Phân tích sâu về các đặc trưng quan trọng và hiệu năng của từng mô hình.
* **Kết luận và hướng phát triển:** Tóm tắt kết quả đạt được và đề xuất các hướng đi tiếp theo để cải tiến hệ thống đánh giá mật khẩu.

Với mục tiêu trên, dự án không chỉ dừng lại ở việc đưa ra một phương pháp đánh giá mật khẩu mà còn hướng tới việc cung cấp một công cụ hữu ích cho người dùng trong việc nâng cao nhận thức về an ninh mạng và bảo mật thông tin cá nhân.

**2. Cơ sở lý thuyết**

**2.1 Độ mạnh của mật khẩu**

Độ mạnh của mật khẩu là một thước đo về khả năng chống lại các cuộc tấn công truy cập trái phép. Một mật khẩu mạnh thường phải đáp ứng các tiêu chí như:

* **Độ dài mật khẩu:** Mật khẩu dài hơn thường khó đoán hơn so với mật khẩu ngắn. Một mật khẩu nên có ít nhất 8 ký tự để đảm bảo độ bảo mật cơ bản.
* **Sự đa dạng ký tự:** Mật khẩu nên bao gồm các ký tự in hoa, in thường, số và ký tự đặc biệt (như @, #, $). Sự đa dạng này làm tăng số lượng các khả năng có thể xảy ra, từ đó tăng tính bảo mật.
* **Tính ngẫu nhiên và tránh các mẫu dễ đoán:** Mật khẩu không nên chứa các mẫu dễ đoán như chuỗi ký tự liên tiếp (123456, abcdef), tên riêng hoặc ngày sinh, bởi những mẫu này dễ bị tấn công thông qua phương pháp từ điển.

Các phương pháp đánh giá độ mạnh của mật khẩu có thể dựa vào việc xem xét các tiêu chí trên. Tuy nhiên, cách tiếp cận truyền thống dựa trên quy tắc (heuristic) thường không đủ mạnh mẽ trong bối cảnh hiện đại, vì chúng chỉ xem xét một số yếu tố đơn lẻ mà không có khả năng tổng hợp toàn diện.

**2.2 Các phương pháp đánh giá hiện tại**

Phương pháp truyền thống sử dụng các quy tắc cố định như yêu cầu về độ dài tối thiểu và sự hiện diện của các ký tự đặc biệt. Mặc dù đơn giản và dễ triển khai, các phương pháp này dễ dàng bị lừa dối bởi những mật khẩu chỉ tuân thủ hình thức mà không thực sự an toàn. Các phương pháp cải tiến bao gồm:

* **Phương pháp heuristic:** Dựa trên các quy tắc đã được xác định trước, ví dụ như yêu cầu mật khẩu phải có ít nhất một ký tự đặc biệt và số. Phương pháp này không đủ linh hoạt và dễ dàng bị phá vỡ.
* **Phương pháp dựa trên từ điển:** Sử dụng một danh sách các mật khẩu phổ biến để kiểm tra mật khẩu đầu vào. Tuy nhiên, phương pháp này chỉ phát hiện được những mật khẩu phổ biến chứ không đánh giá được độ mạnh của các mật khẩu mới lạ.

**2.3 Học máy trong đánh giá mật khẩu**

Học máy (Machine Learning) là một lĩnh vực con của trí tuệ nhân tạo (AI) tập trung vào việc xây dựng các mô hình toán học để phân tích dữ liệu và đưa ra dự đoán hoặc quyết định mà không cần lập trình cụ thể cho từng tác vụ. Trong ngữ cảnh của đánh giá độ mạnh mật khẩu, học máy giúp xác định mối quan hệ phức tạp giữa các đặc trưng của mật khẩu và độ mạnh của chúng dựa trên các mẫu dữ liệu thực tế.

Trong đề tài này, chúng tôi sử dụng ba mô hình học máy chính: Naive Bayes, Logistic Regression, và Random Forest. Mỗi mô hình có những ưu điểm riêng và phù hợp cho việc đánh giá độ mạnh mật khẩu theo các cách khác nhau:

**2.3.1 Naive Bayes**

Naive Bayes là một thuật toán phân loại dựa trên định lý Bayes với giả định độc lập giữa các đặc trưng. Mặc dù giả định này có thể không luôn đúng trong thực tế, Naive Bayes vẫn hoạt động hiệu quả trong nhiều bài toán phân loại vì:

* **Đơn giản và nhanh chóng:** Naive Bayes là một trong những thuật toán nhanh nhất và ít tốn tài nguyên nhất, phù hợp cho các ứng dụng đòi hỏi xử lý tức thời.
* **Hiệu quả với dữ liệu lớn:** Do không yêu cầu tính toán phức tạp, mô hình này dễ dàng mở rộng cho các tập dữ liệu lớn.
* **Phù hợp với các bài toán phân loại:** Naive Bayes đặc biệt hiệu quả khi cần phân loại nhanh giữa các lớp, chẳng hạn như phân loại mật khẩu vào các nhóm "yếu" hoặc "mạnh."

**2.3.2 Logistic Regression**

Logistic Regression là một mô hình thống kê thường được sử dụng để dự đoán khả năng xảy ra của một sự kiện nhị phân. Trong bối cảnh này, Logistic Regression dự đoán khả năng một mật khẩu thuộc về lớp mật khẩu mạnh hoặc yếu dựa trên các đặc trưng của nó.

* **Khả năng diễn giải cao:** Logistic Regression cung cấp các hệ số trọng số cho từng đặc trưng, giúp người dùng hiểu rõ hơn về tác động của mỗi đặc trưng đối với độ mạnh mật khẩu.
* **Tính linh hoạt:** Mặc dù tên là "Logistic Regression," thuật toán này có thể áp dụng cho các bài toán phân loại đa lớp và có thể được mở rộng cho các lớp "yếu," "trung bình," và "mạnh" nếu cần.
* **Khả năng tổng quát tốt:** Logistic Regression có khả năng tránh overfitting tốt hơn các mô hình phức tạp khác, đặc biệt là với các dữ liệu có đặc trưng đơn giản như trong bài toán đánh giá mật khẩu.

**2.3.3 Random Forest**

Random Forest là một thuật toán học máy mạnh mẽ dựa trên việc xây dựng nhiều cây quyết định và kết hợp chúng để cải thiện độ chính xác của dự đoán. Đối với bài toán đánh giá mật khẩu, Random Forest có những ưu điểm vượt trội:

* **Xử lý tốt tính không đồng nhất của dữ liệu:** Random Forest không giả định bất kỳ mối quan hệ tuyến tính nào giữa các đặc trưng, điều này giúp nó hoạt động hiệu quả ngay cả khi các đặc trưng không tương quan tuyến tính với nhau.
* **Khả năng xử lý và đánh giá đặc trưng:** Random Forest có thể đánh giá mức độ quan trọng của từng đặc trưng, từ đó giúp chúng ta hiểu rõ hơn đặc trưng nào quan trọng nhất trong việc phân loại mật khẩu.
* **Khả năng tổng quát cao:** Nhờ kết hợp nhiều cây quyết định, Random Forest giảm thiểu nguy cơ overfitting so với các mô hình dựa trên một cây quyết định duy nhất.
* **Ứng dụng đa dạng:** Bên cạnh đánh giá độ mạnh của mật khẩu, Random Forest còn được ứng dụng rộng rãi trong các bài toán như phát hiện gian lận, dự đoán y tế, và phân loại văn bản.

Mỗi mô hình đều có các ưu điểm riêng và phù hợp cho các ngữ cảnh khác nhau. Trong đề tài này, việc so sánh hiệu năng của các mô hình Naive Bayes, Logistic Regression, và Random Forest giúp xác định phương pháp nào phù hợp nhất để đánh giá độ mạnh của mật khẩu. Kết quả từ mỗi mô hình được đánh giá dựa trên các tiêu chí như độ chính xác, khả năng dự đoán, và tính dễ dàng trong triển khai thực tế.

**3. Phương pháp nghiên cứu**

**3.1 Dữ liệu và nguồn dữ liệu**

Bộ dữ liệu cho nghiên cứu này bao gồm các mật khẩu thực tế đã được phân loại theo độ mạnh. Các mật khẩu được gán nhãn theo ba mức độ: **yếu** (0), **trung bình** (1), và **mạnh** (2). Nguồn dữ liệu được thu thập từ các bộ sưu tập mật khẩu công khai và được chuẩn bị sẵn với nhãn để phục vụ cho việc huấn luyện các mô hình phân loại.

Dưới đây là các hàng đầu tiên, các thông tin, kết quả kiểm tra các giá trị bị thiếu và thống kê mô tả của các cột số học của bộ dữ liệu

**password strength**

**0 kzde5577 1**

**1 kino3434 1**

**2 visi7k1yr 1**

**3 megzy123 1**

**4 lamborghin1 1**

**<class 'pandas.core.frame.DataFrame'>**

**RangeIndex: 669640 entries, 0 to 669639**

**Data columns (total 2 columns):**

**# Column Non-Null Count Dtype**

**--- ------ -------------- -----**

**0 password 669639 non-null object**

**1 strength 669640 non-null int64**

**dtypes: int64(1), object(1)**

**memory usage: 10.2+ MB**

**None**

**password 1**

**strength 0**

**dtype: int64**

**strength**

**count 669640.000000**

**mean 0.990196**

**std 0.507948**

**min 0.000000**

**25% 1.000000**

**50% 1.000000**

**75% 1.000000**

**max 2.000000**

**3.2 Tiền xử lý dữ liệu**

Quá trình tiền xử lý dữ liệu được thực hiện để đảm bảo rằng dữ liệu đầu vào của mô hình là đầy đủ và có chất lượng. Các bước tiền xử lý bao gồm:

* **Loại bỏ các giá trị trùng lặp và rỗng**: Để đảm bảo rằng mỗi bản ghi trong dữ liệu là duy nhất và có ý nghĩa.
* **Chuyển đổi mật khẩu thành chuỗi ký tự**: Đảm bảo rằng tất cả các mật khẩu được lưu trữ dưới dạng chuỗi ký tự và sẵn sàng cho việc trích xuất đặc trưng.

Dưới đây là kết quả của bộ dữ liệu sau khi tiền xử lý:

**<class 'pandas.core.frame.DataFrame'>**

**Index: 669639 entries, 0 to 669639**

**Data columns (total 2 columns):**

**# Column Non-Null Count Dtype**

**--- ------ -------------- -----**

**0 password 669639 non-null object**

**1 strength 669639 non-null int64**

**dtypes: int64(1), object(1)**

**memory usage: 15.3+ MB**

**None**

**password strength**

**0 kzde5577 1**

**1 kino3434 1**

**2 visi7k1yr 1**

**3 megzy123 1**

**4 lamborghin1 1**

**3.3 Trích xuất đặc trưng**

Từ các mật khẩu trong bộ dữ liệu, chúng tôi đã trích xuất chín đặc trưng bao gồm:

1. **Độ dài mật khẩu** - Tổng số ký tự.
2. **Số lượng chữ in hoa** - Đếm số ký tự viết hoa.
3. **Số lượng chữ thường** - Đếm số ký tự viết thường.
4. **Số lượng ký tự số** - Đếm số chữ số.
5. **Số lượng ký tự đặc biệt** - Đếm số ký tự không phải là chữ và số.
6. **Số ký tự lặp lại** - Đếm số ký tự lặp lại.
7. **Số cặp ký tự liền nhau** - Xác định các ký tự liền kề có cùng loại, ví dụ như số hoặc chữ cái.
8. **Số cặp số liền nhau** - Đếm các cặp chữ số liền nhau.
9. **Đa dạng ký tự** - Đánh giá sự kết hợp giữa chữ hoa, chữ thường, số và ký tự đặc biệt.

Dưới đây là kết quả của các đặc trưng đã được trích xuất:  
 **length uppercase\_count lowercase\_count digit\_count special\_char\_count \**

**0 8 0 4 4 0**

**1 8 0 4 4 0**

**2 9 0 7 2 0**

**3 8 0 5 3 0**

**4 11 0 10 1 0**

**repeat\_char\_count consecutive\_digits\_count consecutive\_letters\_count \**

**0 2 3 3**

**1 2 3 3**

**2 1 0 4**

**3 0 2 4**

**4 0 0 9**

**char\_variety strength**

**0 6 1**

**1 6 1**

**2 6 1**

**3 6 1**

**4 6 1**

Các đặc trưng này sau đó được chuẩn hóa để chuẩn bị cho quá trình huấn luyện mô hình.

**3.4 Lựa chọn và triển khai mô hình học máy**

Trong dự án này, chúng tôi đã sử dụng ba mô hình phổ biến để đánh giá và phân loại độ mạnh của mật khẩu:

* **Naive Bayes** - Đơn giản và hiệu quả với các bài toán phân loại cơ bản.
* **Logistic Regression** - Phù hợp với các dữ liệu có mối quan hệ tuyến tính và cung cấp khả năng diễn giải tốt.
* **Random Forest** - Mô hình phức tạp với nhiều cây quyết định, phù hợp với các dữ liệu có cấu trúc đa dạng và yêu cầu độ chính xác cao.

**3.5 Huấn luyện và đánh giá mô hình**

* **Chia dữ liệu**: Dữ liệu được chia thành 70% cho huấn luyện và 30% cho kiểm tra.
* **Chuẩn hóa dữ liệu**: Các đặc trưng đầu vào được chuẩn hóa để đảm bảo mô hình học hiệu quả.
* **Huấn luyện mô hình**: Mỗi mô hình được huấn luyện và sau đó đánh giá bằng tập kiểm tra để xác định độ chính xác và khả năng phân loại của chúng.
* **Đánh giá hiệu suất**: Độ chính xác, F1-score và các chỉ số khác được ghi nhận và so sánh để xác định mô hình nào hoạt động tốt nhất trên dữ liệu đã chuẩn bị.

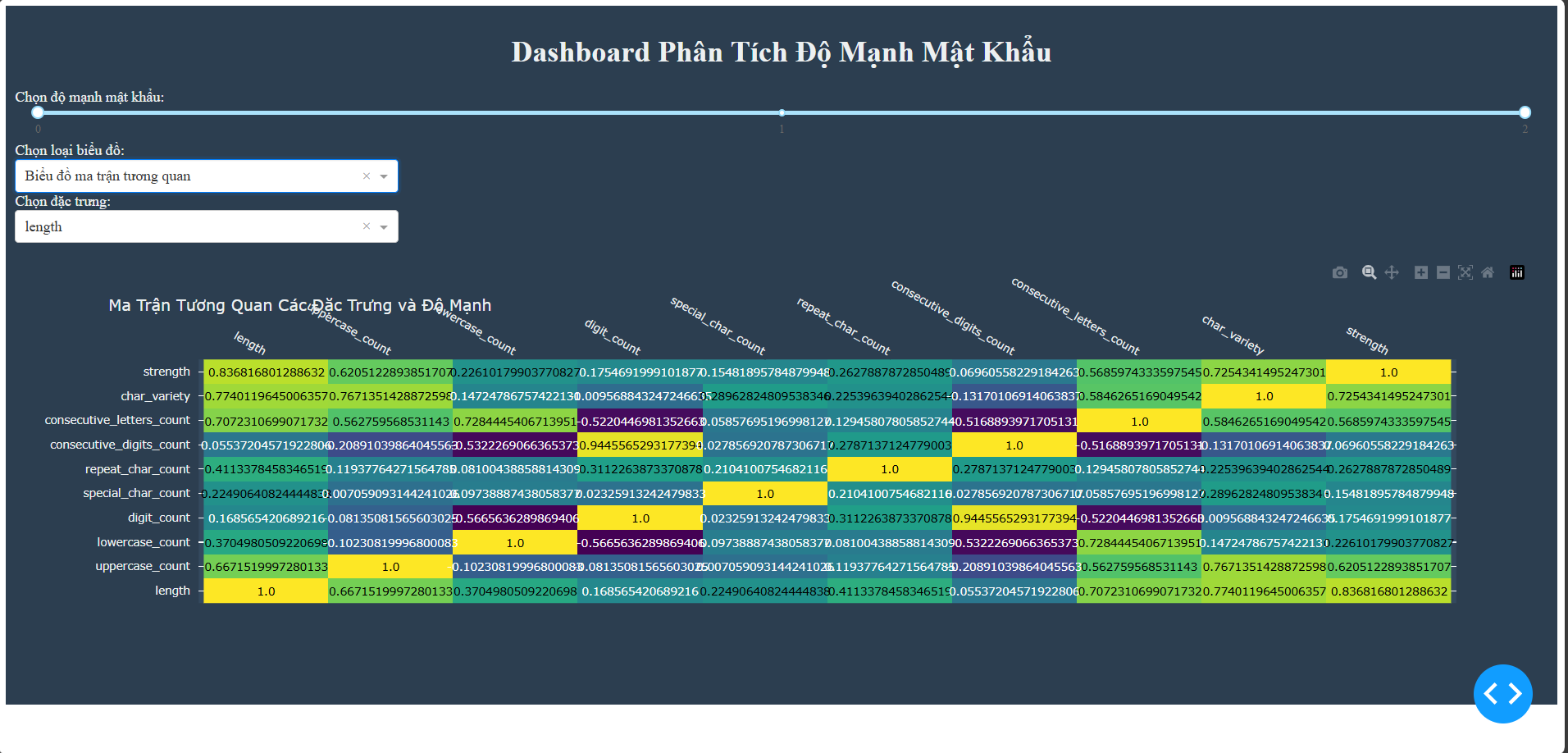
**3.6 Xây dựng Dashboard với Dash by Plotly**

Để trực quan hóa dữ liệu và hỗ trợ quá trình phân tích, chúng tôi đã xây dựng một dashboard sử dụng Dash by Plotly. Dashboard này cho phép người dùng tương tác với dữ liệu và xem xét các đặc trưng của mật khẩu thông qua các biểu đồ trực quan:

* **Biểu đồ đường**: Cho phép người dùng theo dõi các thay đổi trong một đặc trưng cụ thể theo độ mạnh của mật khẩu.
* **Biểu đồ phân tán**: Giúp phân tích mối quan hệ giữa các đặc trưng và độ mạnh mật khẩu.
* **Biểu đồ tròn (Pie chart)**: Cung cấp cái nhìn tổng quát về tỷ lệ các loại mật khẩu yếu, trung bình, và mạnh.
* **Ma trận tương quan**: Hiển thị mối quan hệ giữa các đặc trưng với độ mạnh mật khẩu, giúp nhận diện các đặc trưng có tương quan cao và đánh giá vai trò của chúng.

Dashboard này được xây dựng với giao diện màu sắc sống động, thân thiện với người dùng và cho phép điều chỉnh các tham số để có cái nhìn sâu hơn vào dữ liệu.

Giao diện Dashboard với ma trận tương quan làm ví dụ:



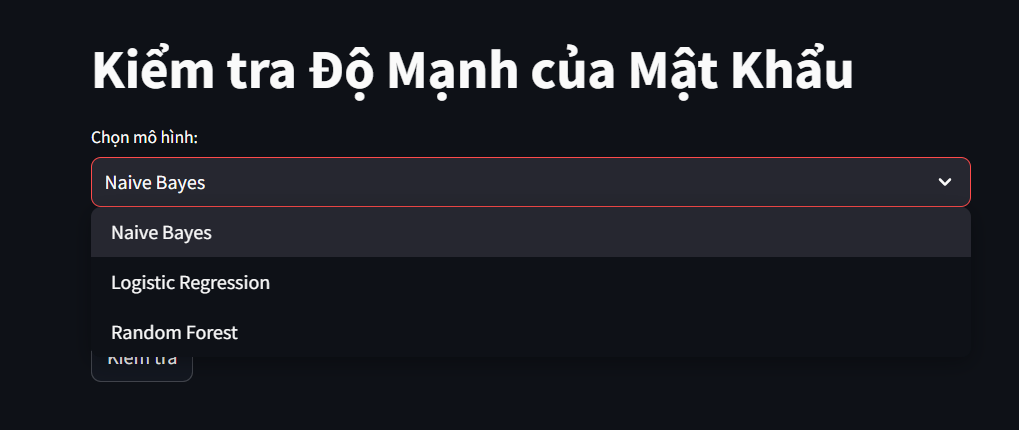
**3.7 Phát triển ứng dụng kiểm tra mật khẩu với Streamlit**

Ngoài dashboard, chúng tôi còn xây dựng một ứng dụng đơn giản sử dụng Streamlit để kiểm tra độ mạnh của mật khẩu. Ứng dụng này cho phép người dùng:

* **Chọn mô hình**: Naive Bayes, Logistic Regression hoặc Random Forest, tùy thuộc vào nhu cầu và hiệu suất mà người dùng mong muốn.
* **Nhập mật khẩu**: Người dùng có thể nhập mật khẩu của mình để kiểm tra.
* **Kết quả đánh giá**: Ứng dụng sẽ trích xuất các đặc trưng từ mật khẩu, chuẩn hóa dữ liệu và sau đó sử dụng mô hình đã chọn để dự đoán độ mạnh. Kết quả sẽ được hiển thị ngay lập tức.

Ứng dụng này có giao diện thân thiện, dễ sử dụng và cho phép người dùng dễ dàng đánh giá độ mạnh của mật khẩu mà không cần kiến thức chuyên môn sâu về máy học. Việc sử dụng Streamlit cũng giúp đơn giản hóa quá trình triển khai và tạo điều kiện thuận lợi cho người dùng tương tác với mô hình.

Giao diện ứng dụng:





**4. Kết quả và Thảo luận**

**4.1 Kết quả huấn luyện và đánh giá mô hình**

Dựa trên các mô hình đã huấn luyện, chúng tôi đã thu được các kết quả chi tiết về độ chính xác, precision, recall, và F1-score cho các mô hình Naive Bayes, Logistic Regression, và Random Forest:

**Model: Naive Bayes**

**Accuracy: 0.9770324353383908**

**Classification Report:**

**precision recall f1-score support**

**0 1.00 0.90 0.95 26903**

**1 0.98 0.99 0.99 148957**

**2 0.92 1.00 0.96 25032**

**accuracy 0.98 200892**

**macro avg 0.97 0.96 0.96 200892**

**weighted avg 0.98 0.98 0.98 200892**

**------------------------------------------------------------**

**Đã lưu mô hình Naive Bayes thành công vào naive\_bayes\_model.pkl**

**Model: Logistic Regression**

**Accuracy: 0.9962467395416442**

**Classification Report:**

**precision recall f1-score support**

**0 0.99 0.99 0.99 26903**

**1 1.00 1.00 1.00 148957**

**2 0.99 1.00 0.99 25032**

**accuracy 1.00 200892**

**macro avg 0.99 0.99 0.99 200892**

**weighted avg 1.00 1.00 1.00 200892**

**------------------------------------------------------------**

**Đã lưu mô hình Logistic Regression thành công vào logistic\_regression\_model.pkl**

**Model: Random Forest**

**Accuracy: 1.0**

**Classification Report:**

**precision recall f1-score support**

**0 1.00 1.00 1.00 26903**

**1 1.00 1.00 1.00 148957**

**2 1.00 1.00 1.00 25032**

**accuracy 1.00 200892**

**macro avg 1.00 1.00 1.00 200892**

**weighted avg 1.00 1.00 1.00 200892**

**------------------------------------------------------------**

**Đã lưu mô hình Random Forest thành công vào random\_forest\_model.pkl**

**Đã lưu scaler thành công.**

* **Naive Bayes**:
  + **Độ chính xác**: 97.7%
  + **Precision, Recall và F1-score**: Mô hình đạt được F1-score trung bình là 96% trên các loại mật khẩu.
  + **Phân tích**: Naive Bayes tuy đơn giản nhưng cho thấy hiệu suất khá tốt trong việc phân loại độ mạnh của mật khẩu.
* **Logistic Regression**:
  + **Độ chính xác**: 99.6%
  + **Precision, Recall và F1-score**: Mô hình này đạt F1-score trung bình là 99%, cho thấy khả năng phân loại gần như hoàn hảo trong dữ liệu được sử dụng.
  + **Phân tích**: Logistic Regression tỏ ra hiệu quả và ổn định, đặc biệt với dữ liệu có tính chất tuyến tính hoặc gần tuyến tính​.
* **Random Forest**:
  + **Độ chính xác**: 100%
  + **Precision, Recall và F1-score**: Mô hình này đạt được F1-score là 100% trên tất cả các loại mật khẩu.
  + **Phân tích**: Với khả năng xử lý phức tạp của các cây quyết định, Random Forest đã phân loại chính xác và cho thấy là mô hình hiệu quả nhất trong dự án này​.

**4.2 Phân tích các đặc trưng quan trọng**

Trong mô hình Random Forest, chúng tôi đã tiến hành phân tích độ quan trọng của các đặc trưng để hiểu rõ hơn về các yếu tố ảnh hưởng đến độ mạnh của mật khẩu.

**Accuracy: 1.0**

**Classification Report:**

**precision recall f1-score support**

**0 1.00 1.00 1.00 26903**

**1 1.00 1.00 1.00 148957**

**2 1.00 1.00 1.00 25032**

**accuracy 1.00 200892**

**macro avg 1.00 1.00 1.00 200892**

**weighted avg 1.00 1.00 1.00 200892**

**Feature Importances:**

**length 0.547215**

**char\_variety 0.187890**

**uppercase\_count 0.124460**

**consecutive\_letters\_count 0.048601**

**lowercase\_count 0.038850**

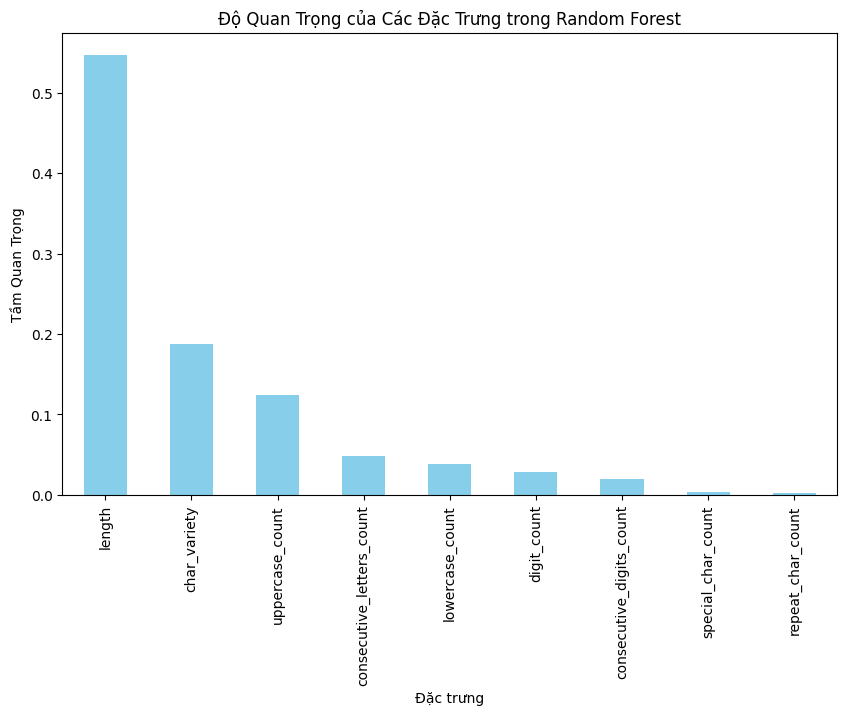
**digit\_count 0.028266**

**consecutive\_digits\_count 0.019749**

**special\_char\_count 0.003280**

**repeat\_char\_count 0.001689**

**dtype: float64**

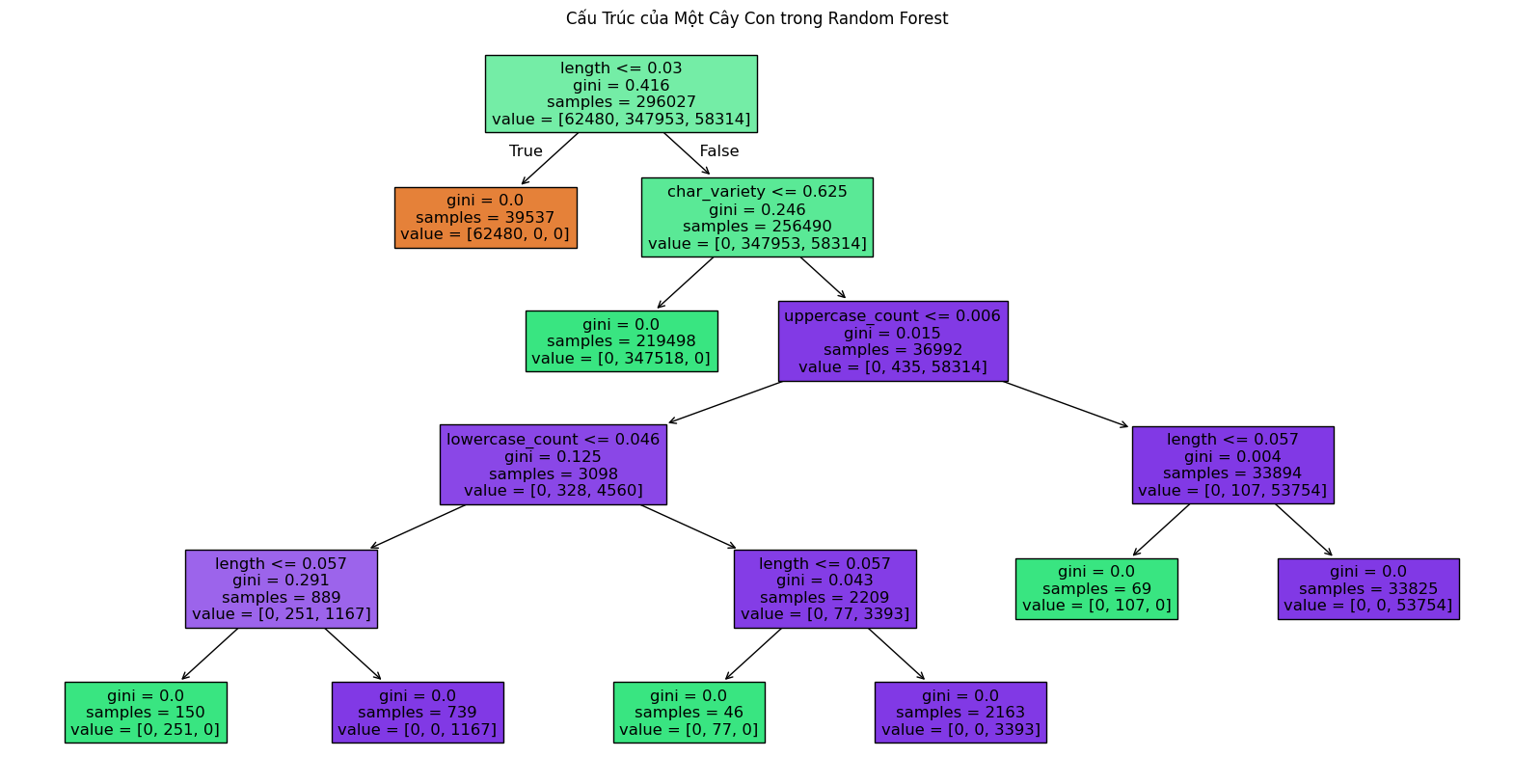
****

Kết quả cho thấy:

* **Độ dài mật khẩu** là đặc trưng quan trọng nhất với tỉ trọng là 54.7%.
* **Độ đa dạng ký tự** cũng đóng vai trò quan trọng, chiếm 18.8%.
* Các đặc trưng khác như số lượng chữ in hoa, số lượng ký tự liền nhau, và số ký tự lặp lại cũng đóng góp vào khả năng phân loại của mô hình​.

**4.3. Phân tích mô hình**

**4.3.1. Phân tích mô hình RandomForest**



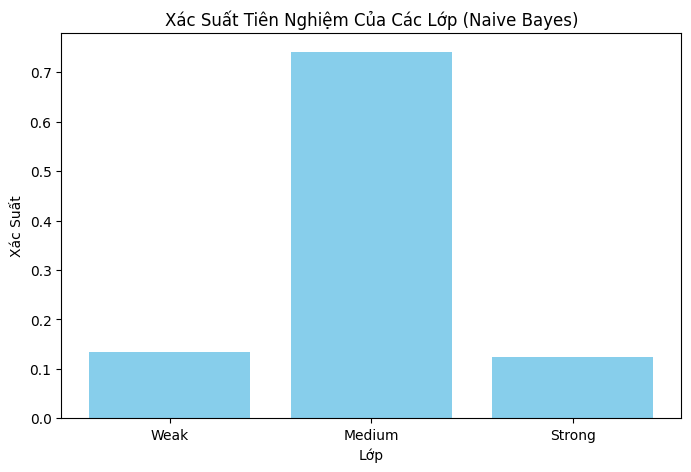
Trong quá trình phân tích Random Forest, chúng tôi đã trích xuất cấu trúc của một trong những cây con. Biểu đồ bên dưới thể hiện cấu trúc này, bao gồm các nút và ngưỡng phân chia dựa trên các đặc trưng của mật khẩu như length, char\_variety, uppercase\_count, v.v.

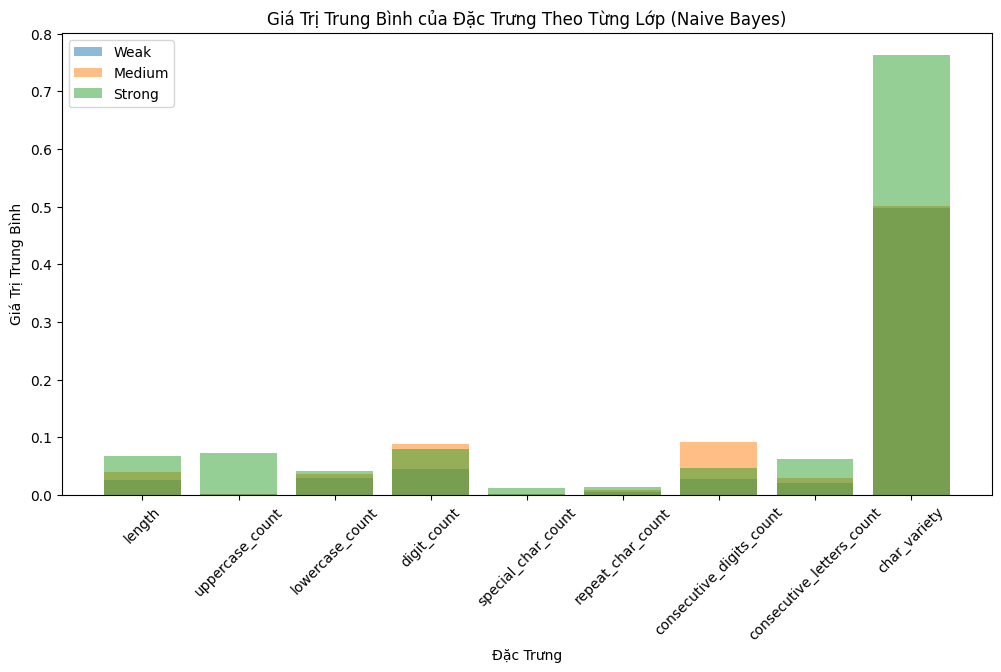
Cấu trúc cây này cho thấy cách mà Random Forest phân loại mật khẩu dựa trên các đặc trưng. Mỗi nút đại diện cho một điều kiện phân chia, và các lá cây cho biết lớp đầu ra (mạnh, trung bình hoặc yếu) cùng với số lượng mẫu ở mỗi lớp.

Nhìn chung, length là một đặc trưng quan trọng được sử dụng sớm trong quá trình phân chia, tiếp theo là char\_variety và uppercase\_count. Điều này phản ánh rằng độ dài và sự đa dạng trong loại ký tự là yếu tố quan trọng để xác định độ mạnh của mật khẩu.

Hình ảnh của cấu trúc cây này cung cấp cái nhìn rõ ràng về logic mà mô hình sử dụng để đưa ra quyết định, góp phần làm rõ quy trình đánh giá độ mạnh mật khẩu.

**4.3.2. Phân tích mô hình NaiveBayes**

****

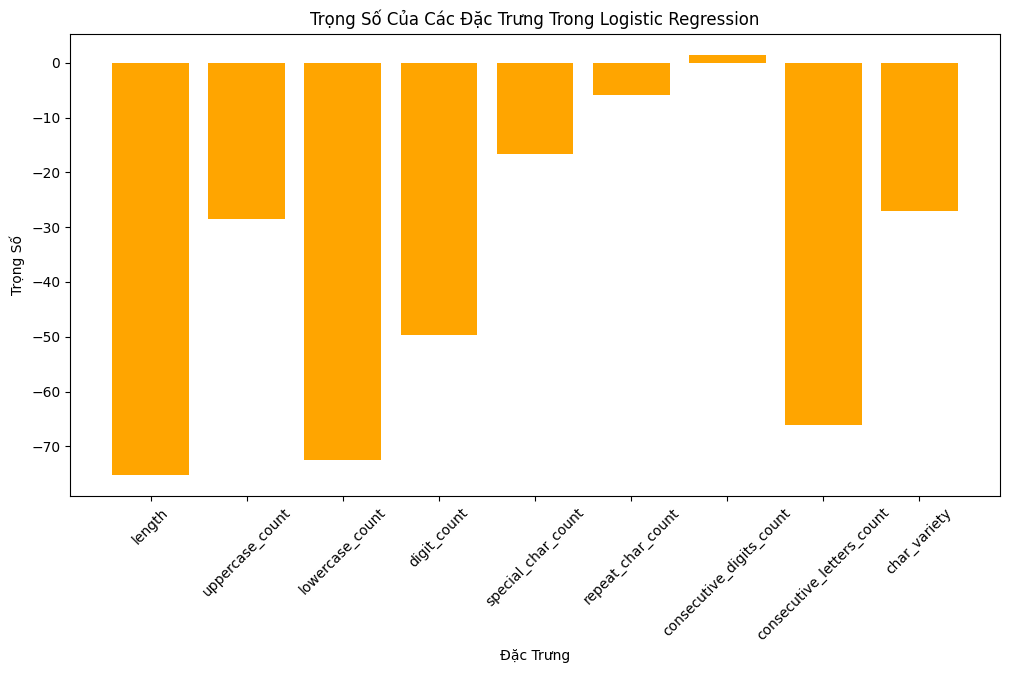
****

Xác suất tiên nghiệm của các lớp: Biểu đồ xác suất tiên nghiệm của các lớp mật khẩu cho thấy lớp 'Medium' chiếm tỷ lệ cao nhất, vượt qua 70%, trong khi các lớp 'Weak' và 'Strong' có xác suất tiên nghiệm thấp hơn. Điều này có thể phản ánh đặc điểm phân phối của tập dữ liệu mật khẩu với đa số mật khẩu ở mức trung bình.

Giá trị trung bình của đặc trung theo từng lớp: Các đặc trưng được Naive Bayes sử dụng cho thấy sự khác biệt về giá trị trung bình giữa các lớp 'Weak', 'Medium', và 'Strong'. Đặc trưng 'char\_variety' có giá trị trung bình cao nhất ở lớp 'Strong', cho thấy sự đa dạng trong các ký tự của mật khẩu đóng vai trò quan trọng trong việc xác định mật khẩu mạnh. Ngoài ra, các đặc trưng như 'length', 'uppercase\_count', và 'digit\_count' cũng góp phần tạo ra sự khác biệt giữa các mức độ mạnh của mật khẩu.

* + Mô hình Naive Bayes phù hợp với việc xử lý các đặc trưng độc lập và cung cấp cái nhìn tổng quan về cách mà mỗi đặc trưng ảnh hưởng đến độ mạnh của mật khẩu. Tuy nhiên, do đặc điểm giả định tính độc lập của các đặc trưng, mô hình này có thể không đạt độ chính xác cao như một số mô hình khác nhưng lại cho phép suy luận trực tiếp từ các xác suất tiên nghiệm.

**4.3.3. Phân tích mô hình Logistic Regression**



**Phần chặn (Intercept): [ 24.27439574 4.67450727 -28.948903 ]**

Trọng số của các đặc trưng: Biểu đồ trọng số của các đặc trưng trong Logistic Regression cho thấy các đặc trưng như 'length', 'uppercase\_count', và 'digit\_count' có tác động lớn trong việc phân loại mật khẩu. Các trọng số âm cho thấy khi giá trị của các đặc trưng này tăng, xác suất mật khẩu yếu cũng có thể tăng, điều này tùy thuộc vào mối quan hệ cụ thể giữa đặc trưng và độ mạnh của mật khẩu.

Phần chặn (Intercept): Logistic Regression có các phần chặn khác nhau cho mỗi lớp mật khẩu, với các giá trị [24.27, 4.67, -28.95]. Điều này thể hiện cách thức mà mô hình điều chỉnh dựa trên phần còn lại của đặc trưng để dự đoán lớp mật khẩu.

**4.3. Thảo luận**

Các mô hình Logistic Regression và Random Forest cho thấy hiệu suất rất cao, với Random Forest đạt độ chính xác tuyệt đối. Tuy nhiên, để tối ưu hơn nữa cho các ứng dụng thực tế, chúng tôi khuyến nghị thử nghiệm thêm với các thuật toán khác hoặc kết hợp các mô hình để kiểm tra sự cải thiện về hiệu suất và tốc độ xử lý.

**5. Kết Luận và Đề Xuất Phát Triển**

**5.1 Kết Luận**

Dựa trên các mô hình Naive Bayes, Logistic Regression và Random Forest, hệ thống đã đạt được những kết quả khả quan trong việc phân loại và đánh giá độ mạnh của mật khẩu. Cụ thể:

* **Naive Bayes**: Cho phép mô hình hóa mối quan hệ giữa các đặc trưng với giả định độc lập, giúp dự đoán nhanh chóng và dễ dàng triển khai. Tuy nhiên, hạn chế của mô hình này là có thể bỏ qua sự phụ thuộc giữa các đặc trưng, điều này có thể làm giảm độ chính xác.
* **Logistic Regression**: Mô hình này có khả năng giải thích rõ ràng mối quan hệ giữa các đặc trưng và xác suất phân loại mật khẩu thông qua trọng số của các đặc trưng. Với Logistic Regression, người dùng có thể hiểu sâu hơn về tác động của từng đặc trưng đến độ mạnh của mật khẩu.
* **Random Forest**: Mô hình này đạt hiệu suất cao hơn khi xử lý tập dữ liệu phức tạp nhờ vào cấu trúc cây ngẫu nhiên, cho phép mô hình học từ các đặc trưng khác nhau một cách hiệu quả. Bên cạnh đó, khả năng trực quan hóa qua biểu đồ cây cũng cung cấp cái nhìn trực quan về cách thức mà các đặc trưng ảnh hưởng đến độ mạnh của mật khẩu.

**5.2 Hạn Chế**

Mặc dù các mô hình này đã mang lại kết quả tốt, nhưng hệ thống vẫn còn một số hạn chế:

* **Tập dữ liệu**: Nếu tập dữ liệu không đủ lớn hoặc không đại diện, các mô hình có thể không đạt được độ chính xác cao khi dự đoán.
* **Đặc trưng của mật khẩu**: Một số đặc trưng có thể bị bỏ qua trong quá trình trích xuất, dẫn đến việc mô hình không khai thác hết tiềm năng thông tin.
* **Mô hình hóa đơn giản**: Các mô hình hiện tại chưa xem xét đến các yếu tố phức tạp như chuỗi kí tự liên tục hoặc chuỗi từ điển, mà có thể giúp phân biệt mật khẩu mạnh và yếu.

**5.3 Đề Xuất Phát Triển**

Dựa trên các kết quả và hạn chế đã được phân tích, các hướng phát triển tương lai có thể bao gồm:

* **Mở rộng tập dữ liệu**: Tăng cường kích thước và sự đa dạng của tập dữ liệu bằng cách thu thập thêm nhiều mẫu mật khẩu từ nhiều nguồn khác nhau, giúp mô hình hóa được các loại mật khẩu thực tế hơn.
* **Phát triển các đặc trưng mới**: Bổ sung các đặc trưng liên quan đến từ điển hoặc các mô hình chuỗi để cải thiện khả năng phân loại. Điều này có thể bao gồm việc phát hiện các chuỗi từ điển phổ biến hoặc phát hiện các mẫu chuỗi liên tục.
* **Sử dụng các mô hình học sâu (Deep Learning)**: Cải thiện độ chính xác và khả năng xử lý phức tạp của mô hình bằng cách áp dụng các kỹ thuật học sâu như mạng nơ-ron tích chập (CNN) hoặc mạng nơ-ron hồi quy (RNN) để phát hiện các mẫu mật khẩu phức tạp hơn.
* **Phát triển giao diện người dùng và API**: Tích hợp hệ thống với một giao diện thân thiện hơn cho người dùng cuối, đồng thời phát triển API để hệ thống có thể được sử dụng linh hoạt trên các ứng dụng và nền tảng khác nhau.

**5.4 Ứng Dụng Thực Tiễn**

Hệ thống đánh giá độ mạnh mật khẩu này có thể được ứng dụng trong nhiều lĩnh vực, bao gồm:

* **Ứng dụng quản lý mật khẩu**: Các công ty cung cấp dịch vụ quản lý mật khẩu có thể tích hợp hệ thống này để giúp người dùng tự động kiểm tra độ mạnh của mật khẩu.
* **An ninh mạng**: Các tổ chức và doanh nghiệp có thể sử dụng hệ thống để tăng cường bảo mật thông qua việc kiểm tra mật khẩu của nhân viên hoặc người dùng trước khi cho phép sử dụng trong hệ thống.
* **Giáo dục và đào tạo**: Các tổ chức giáo dục có thể sử dụng công cụ này để đào tạo và nâng cao nhận thức về an toàn mật khẩu cho học sinh, sinh viên và nhân viên.

Phần kết luận đã tổng hợp lại các ưu nhược điểm, cùng với đó là các hướng phát triển và ứng dụng thực tiễn cho hệ thống đánh giá độ mạnh của mật khẩu, từ đó giúp cải thiện và tối ưu hóa hệ thống trong tương lai.

**TÀI LIỆU THAM KHẢO**

 Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, 487-499.

 Bonneau, J., Herley, C., van Oorschot, P. C., & Stajano, F. (2012). The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. 2012 IEEE Symposium on Security and Privacy, 553-567. https://doi.org/10.1109/SP.2012.44

 Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

 D’Agostino, R. B., Belanger, A., & D’Agostino, R. B., Jr. (1990). A suggestion for using powerful and informative tests of normality. The American Statistician, 44(4), 316-321. https://doi.org/10.1080/00031305.1990.10475751

 Foroughi, H. J., Saberi, A., & Fathy, M. (2013). Robust password strength evaluation model using machine learning. 2013 5th Conference on Information and Knowledge Technology, 32-38. https://doi.org/10.1109/IKT.2013.6698160

 Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1). Springer.

 Gupta, A., & Dalal, K. (2021). Evaluation of machine learning algorithms for password strength estimation. International Journal of Computer Applications, 174(22), 30-36.

 Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

 James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: With applications in R. Springer. https://doi.org/10.1007/978-1-4614-7138-7

 Singh, A., & Singh, S. (2020). A comparative analysis of supervised learning algorithms for password strength estimation. International Journal of Advanced Computer Science and Applications, 11(9), 255-262. https://doi.org/10.14569/IJACSA.2020.0110930

 Von Ahn, L., Blum, M., Hopper, N. J., & Langford, J. (2003). CAPTCHA: Using hard AI problems for security. Advances in Cryptology—EUROCRYPT 2003, 294-311. https://doi.org/10.1007/3-540-39200-9\_18