Password Strength Prediction

Avinash Raval, Freya Modi, Freyal Shah, Zalak Shah

School of Engineering and Applied Science, Ahmedabad University, Ahmedabad, Gujarat 380013 Email: [avinash.r, freya.m, freyal.s, zalak.s]@ahduni.edu.in

***Abstract*—In this paper, we try to reproduce the machine learning-based models to predict the strength of a password. The objective of the project is to develop a tool that can accurately classify a password into one of three categories: weak, medium, or strong. The model is trained on a large data-set of passwords, with labels indicating their strength, and various features such as length, complexity, and usage of special characters, numbers, and capital letters. We use multiple machine learning algorithms to identify the best performing model. Logistic Regression Classifier is used for analysis. Thus, we even figure out the balance between accuracy and efficiency by implementation of big data.**

***Keywords*—Accuracy, Strength, Regression, Classification, Password strength**

1. INTRODUCTION

Passwords are widely used for securing online accounts and sensitive information. They are designed to protect user data from unauthorized access by requiring an authentication process to gain access to a particular account. However, the use of weak passwords can make accounts vulnerable to hacking, posing a significant risk to users' privacy and security. Weak passwords are generally easy to guess or crack using various methods, including brute force attacks or dictionary attacks. As such, there is an urgent need to develop an effective and reliable technique for measuring the strength of a password to ensure that users are not using weak passwords that can be easily guessed or cracked. The development of an efficient password strength measuring tool can assist users in generating more robust passwords that are more difficult for attackers to break. This, in turn, can help to reduce the likelihood of security breaches, thereby safeguarding users' confidential data. A reliable password strength measuring technique can also help organizations to enforce stronger password policies, ensuring that their employees use robust passwords that are less susceptible to attacks. Therefore, the development of an efficient password strength measuring tool is crucial in enhancing online security and ensuring that users' privacy and security are not compromised.

1. LITERATURE SURVEY

Umar Farooq in the proposed paper, “Real Time Password Strength Analysis on a Web Application Using Multiple Machine Learning Approaches”, intends to investigate how different machine learning models implemented in the context of a certain application perform. We assess the precision of the Decision Tree, Naive Bayes, Linear Regression, Random Forest, and Neural Network models in detail.

" Machine Learning Based Password Strength Analysis" by Sony Kuriakose, G Krishna Teja, Sravan Duggi, A Harshel Srivatsava, Venkat Jonnalagadda: The article intends to enable users to increase the password strength they choose to input for their accounts in accordance with the analysis offered in the application before setting the password. This will assist the user in avoiding being a victim of cyber password attacks, which are quite likely to occur when using weak or obvious passwords.

Jamuna KS, Karpagavalli S, and Vijaya MS in the paper, “A Novel Approach for Password Strength Analysis through Support Vector Machine”, extracted features from a set of 10,000 passwords and evaluated the performance of a model using 10-fold cross-validation. With an accuracy of 89%, the RBF kernel-trained SVM classifier was found to perform well. The study's findings underline the significance of password security in the digital era and show how feature extraction techniques and machine learning algorithms are helpful in evaluating password strength.

1. IMPLEMENTATION

The proposed model is implemented using Python and various libraries such as scikit-learn, pandas, and NumPy.

**Finding the dataset:** The first task involves finding the appropriate dataset from a reliable source so that the model accuracy is not at stake. A sample of the dataset is given below:

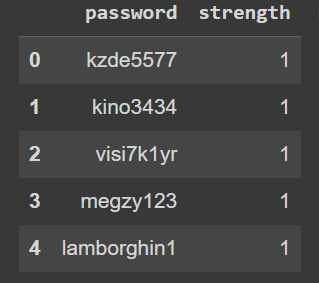


Figure 1: Password Strength Detection Dataset

**Performing EDA and preprocessing data:** It is possible that data samples have null values or duplicate values in the dataset in which they are loaded. This may happen when the data is not captured properly. So, they need to be removed. It is done by using inbuilt python functions. Other than this, the data is normalized to bring all the passwords into a common scale for identification of patterns and features which are indicators of strong password and hence improve the accuracy. On performing EDA, it was found that the data was highly imbalanced with 70.16% of data having their password strength as 1 (medium strength).

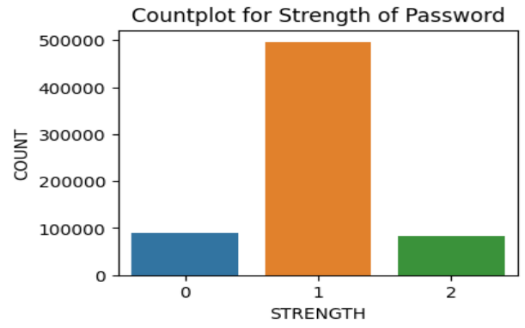


Figure 2: Count plot for Strength of Password

To overcome the problem of imbalanced dataset, oversampling needs to be done. Oversampling is performed using SMOTE algorithm. SMOTE stands for Synthetic Minority Oversampling Technique, which generates synthetic samples of minority class by creating new instances which are similar to existing instances.

**Training and testing:** The model is a Logistic Regression Classifier which uses sigmoid function to model the relationship of input and output variables. The logistic function maps any input value to a probability value between 0 and 1. Here, the probability can be used to predict the probability that the input belongs to the positive class. Here, the positive class is ‘strong password’. For training and testing purposes, data is split into two parts: 80% of the data is used for training and 20% of the data is used for testing the model. After the model has been successfully trained, it is tested against the test data in order to check for its accuracy.

1. RESULTS

The initial results included prediction using a Decision Tree Classifier. The results show that our model can predict the strength of a password with an accuracy of 97%, meaning that 97 percent of the total passwords were correctly classified by the model. However, Decision Tree models easily overfit the data, leading to poor generalization performance. Also, the decision tree models are sensitive to the new data, so they do not work when new data requiring a different decision tree, is added. Thus, an alternative approach that uses the method of logistic regression using gradient descent is applied.

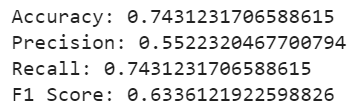


Figure 3: Performance measure on the logistic regression model

The current model for prediction is Logistic Regression Classifier. The results show that our model can predict the strength of a password with an accuracy of 74.30%, meaning that approximately 74.3 percent of total passwords are correctly classified by the model. However, logistic regression outputs a probability distribution over the three classes, indicating the likelihood of the password being weak, medium, or strong.

The logistic regression model was also checked for overfitting by calculating the accuracy for the training dataset as well the validation dataset. Both the accuracies are approximately same meaning that the model is non overfitting.



On plotting a graph between training and validation loss, it was found that training and validation loss lines overlap. From this it can be said that although the model is not overfit but it might be an underfit model.

V. CONCLUSIONS

To conclude the discussion, it can be said that the Logistic Regression Classifier with an accuracy rate of 74.30% can serve as an effective solution for predicting password strength, which is a crucial factor in ensuring online security. However, it is important to note that relying solely on one method is not advisable, as there may be limitations to its effectiveness in certain situations. Moreover, while the Logistic Regression is efficient and makes it easier to interpret the relationship between input and output variables, it falls short in accurately predicting data with minimal variability as Decision Tree Classifier Model gave an accuracy of 97% which is much higher than that of Logistic Regression Model. The difference in accuracy might be due to the reason that the model is underfit in case of Logistic Regression and overfit in case of Decision Tree Classifier. As a result, we conclude that we should not rely only on one model for password strength detection but should opt for a mixture of models which would have more accuracy complexity to capture complex relationships in the data.

1. REFERENCES

[1] Bansal, B. (2019, june 27). *password strength classifier dataset*.Kaggle.com[.https://www.kaggle.com/datasets/bhavikbb/password-strength-classifier-dataset](https://www.kaggle.com/datasets/bhavikbb/password-strength-classifier-dataset)

[2] *Building a Password Strength Classifier Model Using Machine Learning*. (2022, January 26). Section.io. Retrieved March 11, 2023, from [https://www.section.io/engineering-education/building-a-password-strength-classifier-model-using-machine-learning/#adding-features-and-labels](https://www.section.io/engineering-education/building-a-password-strength-classifier-model-using-machine-learning/" \l "adding-features-and-labels)

[3] AdityaBagad/Password-Strength-Analysis: Password Strength Analysis using Machine Learning Classification Algorithms. (n.d.). GitHub. Retrieved March 11, 2023, from <https://github.com/AdityaBagad/Password-Strength-Analysis>

[4] U. Farooq, “Real Time Password Strength Analysis on a Web Application Using Multiple Machine Learning Approaches,” International Journal of Engineering Research & Technology (IJERT), ISSN: 2278- 0181 Vol. 9 Issue 12, December 2020

[5] Sarkar, S., & Nandan, M. (2022). Password Strength Analysis and its Classification by Applying Machine Learning Based Techniques. 10.1109

[6] *Password Strength Prediction Using Supervised Machine Learning Techniques*. (2009, December 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5376606?denied=

Github\_Link: <https://github.com/freyal0811/CSE523-Machine-Learning-2022-Predictors_4.0.git>