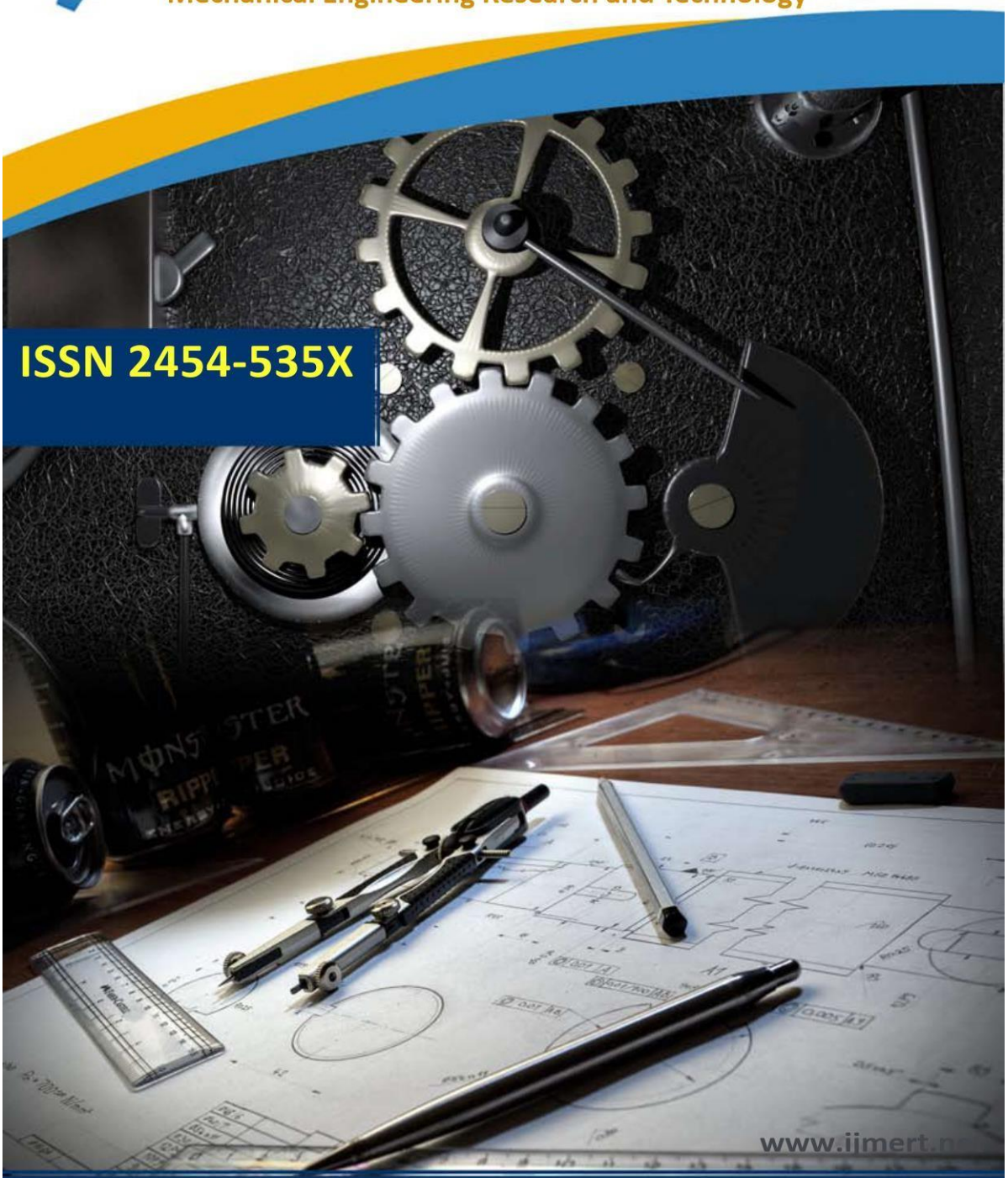
International Journal of Mechanical Engineering Research and Technology

ISSN 2454-535X



Email ID: info.ijmert@gmail.com or editor@ijmert.net

International Journal of Mechanical Engineering Research and Technology

ISSN 2454 – 535X www.ijmert.com
Vol. 15, Issue. 3, sep 2023

A Comparison Study of Random Forest and Logistic Regression for Password Strength Classification

To what extent is Random Forest (RF) favourable over Logistic Regression (LR) based on performance in password strength classification?

By: Ananya Jain

Abstract

Context: Passwords have become the universal method of authentication due to their simplicity and compatibility across a wide range of systems. However, due to their wide-spreadness, they have become vulnerable to external attacks like password cracking. Users are infamously poor at maintaining entropy in their passwords due to their tendency of including dictionary words, names, places, dates, keyboard patterns and so on in passwords, making them predictable. Password strength classifiers developed using machine learning algorithms like Random Forest (RF) and Logistic Regression (LR) can efficiently prevent attacks by coercing users to create strong passwords.

Subjects and Methods: In this study, I trained two machine learning models to detect the strength of different passwords. The two models use Random Forest and Logistic Regression respectively to classify passwords strengths as 0,1 or 2 with 0 being weak and 2 being strongest. I tested the model on 669,643 independent passwords retrieved from Kaggle and evaluated the models' classification against password standards.

Results: Random Forest has higher prediction accuracy whereas Logistic Regression has better time performance

Keywords: *password strength modelling, Random Forest, Logistic Regression, machine learning*

1. Introduction

Password Strength Classifiers are algorithms used to assess the strength or effectiveness of passwords. They are designed to analyse characteristics of a password like length, use of diverse character classes, and complexity to compute a score indicating the password's level of security. Higher score means the password is more effective against external attacks.
The usefulness of password classifiers however lies in their ability to give real-time feedback to users regarding the strength of the password they are creating. Through this, users are forced to use more secure passwords for their accounts[1].

Password strength classification is used across a variety of websites during the user registration process for enforcing strong passwords. These are also extended by password managers like LastPass[2] who use it for assessing the safety of users' saved passwords and for suggesting strong passwords. Additionally, password classifiers are paving their way into education with services like Password Monster that are being used to teach users about good password habits[3].

An approach to creating these algorithms is through machine learning. Random Forest (RF) and Logistic Regression (LR) are two machine learning models to classify password strength. However, it is unclear which model is better based on prediction accuracy and prediction time. Additionally, variation in performance between RF and RS could differ for different application and development budgets.

[1]"Real Time Password Strength Analysis on a Web Application Using Multiple Machine Learning Approaches – IJERT." International Journal of Engineering Research & Technology, 24 December 2020, https://www.ijert.org/real-time-password-strength-analysis-on-a-web-application-using-multiple-machine-learning-approaches. Accessed 2 July 2023.
[2] LastPass. "How Secure is Your Password?" LastPass | Something Went Wrong, lastpass.com/howsecure.php.

International Journal of Mechanical Engineering Research and Technology

ISSN 2454 – 535X www.ijmert.com
Vol. 15, Issue. 3, sep 2023

Accessed 2 July 2023.
[3] PasswordMonster, 3 Mar. 2022, www.passwordmonster.com. Accessed 3 July 2023.

Therefore, ***this paper seeks to investigate whether Random Forest (RF) outperforms Logistic Regression (LR) for password strength classification, as measured by prediction accuracy and prediction time.*** This research could help developers make comprehensive decisions on whether a Random Forest or Logistic Regression should be used, which eliminates investing unnecessary time, computer resources and labour attempting to fit data into an 'unsuitable' machine learning algorithm when a better alternative exists, especially considering machine learning models can require weeks to develop for real world accuracy.[4]

2. Theoretical Background

2.1 Password Strength

Password Strength is the measure of a password's resistance against brute-force attacks and password cracking. The strength lies in the following characteristics of the password: complexity, length and unpredictability[5]. To fulfil these criteria, the following guidelines must be adhered:

- Password length to be at least 12 characters, preferably more
- Use of both uppercase and lowercase characters
- Avoid dictionary words, keyboard patterns, repetitive characters, and letter sequences
- Avoid using names, addresses, phone numbers, or other personal details
- Avoid common passwords like (password, qwerty, 123456)
- Use a random combination of letters, numbers and symbols[6]

For this paper, combinations of the above guidelines are grouped together to form the basis of weak, medium and strong passwords which are numerically represented as 0, 1 and 2 respectively. The nature of each password strength level is described in the table below

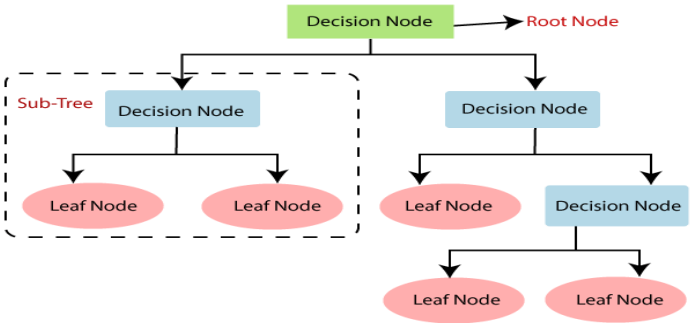
Figure 1: Password Strength Classification

Password	Strength	Reason
(i) patty94	Weak (0)	+ Combination of letters and numbers − Less than 12 characters − Use of name − Only lowercase letters − No symbols
(ii) alimagik1	Medium (1)	+ No identifiable dictionary words + Combination of letters and numbers − No uppercase letters − No symbols − Less than 12 characters
(iii) TyWM72UNEex8Q8Y!	Strong (2)	+ More than 12 characters + Mix of uppercase and lowercase letters + Combination of letters and numbers + Random arrangement, no identifiable sequences + Symbol present

2.2 Drawbacks of Decision Tree

Although decision trees can be used for password strength classification, they may be deemed unsuitable due to the problem of overfitting, especially when dealing with high-dimensional data like passwords.

Figure 2: Decision Tree[7]



A decision tree is generated by recursive splitting of data, based on the most informative feature, into decision nodes that are used to make decisions and leaf nodes that determine the result.[8] Overfitting occurs during this recursive process when the decision tree catches random fluctuations or noise in the data instead of learning the underlying patterns. This overfitting issue can be solved by Random Forest.[9] For this reason, Random Forest has been

International Journal of Mechanical Engineering Research and Technology

ISSN 2454 – 535X www.ijmert.com
Vol. 15, Issue. 3, sep 2023

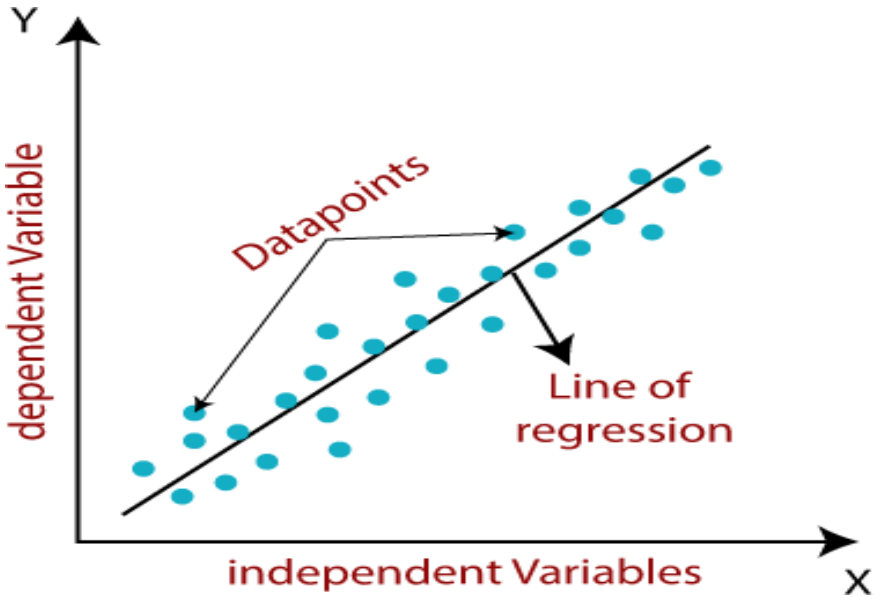
chosen instead.

2.3 Drawbacks of Linear Regression

Linear regression cannot be used because it lacks suitability for password strength classification due its inherent nature of predicting only numerical continuous outcomes, whereas password classification involves categorisation into discrete classes.

"Decision Tree Classification Algorithm." Www.javatpoint.com,
www.javatpoint.com/machine-learning-decision-tree-classification-algorithm. Accessed 10 July 2023.
[8]"What is a Decision Tree." IBM, https://www.ibm.com/topics/decision-trees. Accessed 10 July 2023.
[9]"Decision Tree Algorithm in Machine Learning." Javatpoint,
https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm. Accessed 10 July 2023.

Figure 3: Linear Regression[10]



In Linear Regression, a linear regression is assumed between the inputted feature and output. In the case of password classification, a complex relationship exists between password feature and password strength, therefore, the linear regression model will not accurately

capture the relationship[11]. For this reason, Logistic Regression is used instead, which is better suited to such classification problems.

2.4 Random Forest Algorithm

Random Forest is a supervised machine learning algorithm built on the concept of ensemble learning, where multiple classifiers are combined to solve complex problems[12]. Like the name suggests, a random forest is a collection of multiple decision trees that are trained on various data subsets through a technique known as bootstrapping. Multiple decision trees enhance accuracy as instead of relying on a single decision tree, the algorithm takes the prediction from each tree. In order to compute a result, a majority voting is done with each tree's prediction[13].

[10] "Linear Regression in Machine Learning - Javatpoint." Www.javatpoint.com, www.javatpoint.com/linear-regression-in-machine-learning. Accessed 10 July 2023.
[11]"Linear Regression in Machine Learning - Javatpoint." Www.javatpoint.com, www.javatpoint.com/linear-regression-in-machine-learning. Accessed 10 July 2023.
[12] "Random Forest Algorithm." Www.javatpoint.com, www.javatpoint.com/machine-learning-random-forest-algorithm. Accessed 10 July 2023.

Since each tree is generated independently with different data and attributes, Random Forest allows parallelisation, meaning the CPU can fully be used to create random forests. Moreover, as majority voting is carried out, the model's performance is not heavily affected by minor changes in the dataset, improving stability of the model. The majority vote also has the added advantage of resistance to overfitting.

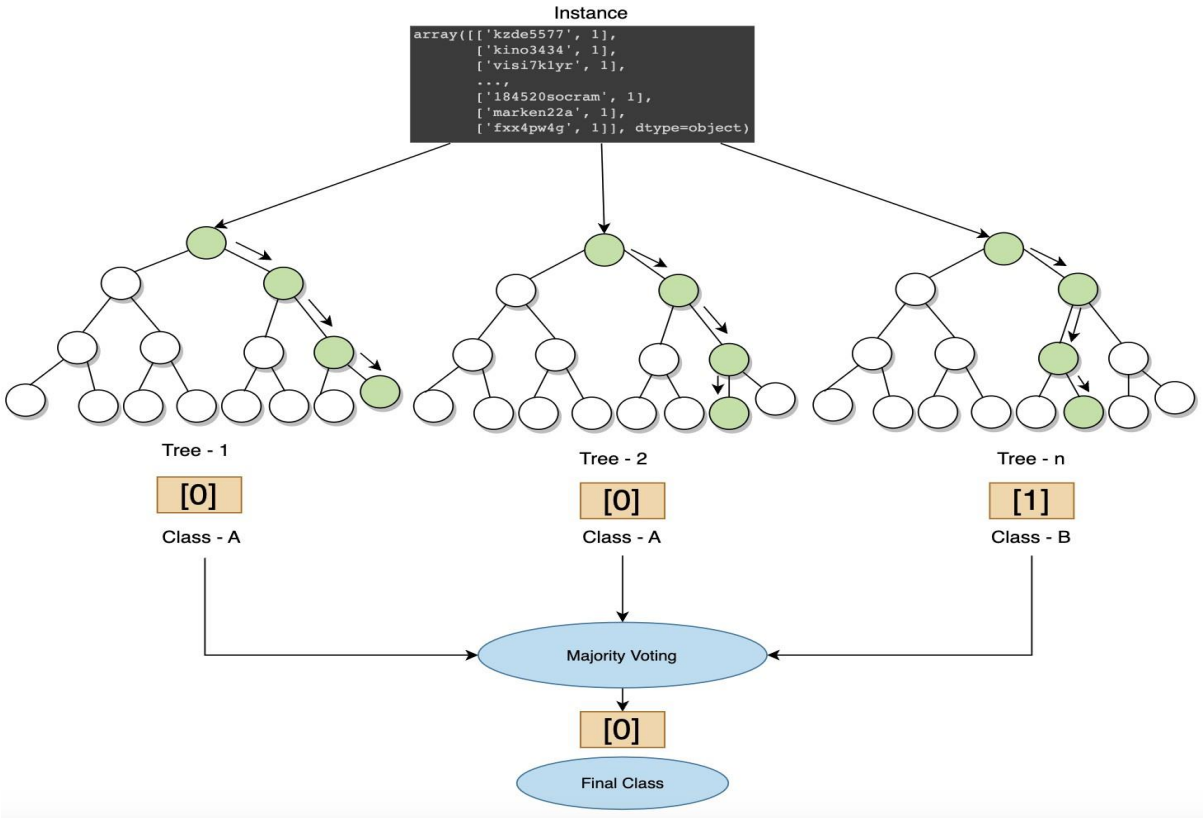
2.4a Working of Random Forest for Password Strength Classification

Figure 4: Random Forest for Password

International Journal of Mechanical Engineering Research and Technology

ISSN 2454 – 535X www.ijmert.com

Vol. 15, Issue. 3, sep 2023



[13]R, Sruthi E. "Random Forest | Introduction to Random Forest Algorithm." Analytics Vidhya, 21 June 2022, www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/. Accessed 10 July 2023.

During Training: Random data points are selected without replacement from the training data to build individual decision trees. During construction of the trees, at each node the algorithm partitions the data until a certain stopping criteria is met.

During Testing: For the same password instance from the training set, every tree individually predicts based on its individual criteria, as seen in the image above, different trees give different predictions. These predictions are then considered as votes and the class with the most votes is then declared as the final predicted class.

2.5 Logistic Regression Algorithm

Logistic Regression is a supervised learning classification algorithm capable of predicting a target variable's probability based on dependent variables. The main purpose of the model is

to find the best fitting model to describe a relationship between an independent variable and a dependent variable[14]. A logistic function is used to model the dependent variable, hence, the name logistic regression. This logistic function is represented the sigmoid function as below:

$$S(x) = \frac{1}{1 + e^{-x}}$$

Logistic Regression calculates its output using this equation to return a probability value (between 0 and 1) determining its classification[15]. Although logistic regression is traditionally used for binary classification, it can be extended further to classify three or more classes, known as Multinomial Binary Classification[16], which is used in this case.

---

[14] "Logistic Regression in Machine Learning." Www.javatpoint.com, www.javatpoint.com/logistic-regression-in-machine-learning. Accessed 10 July 2023.

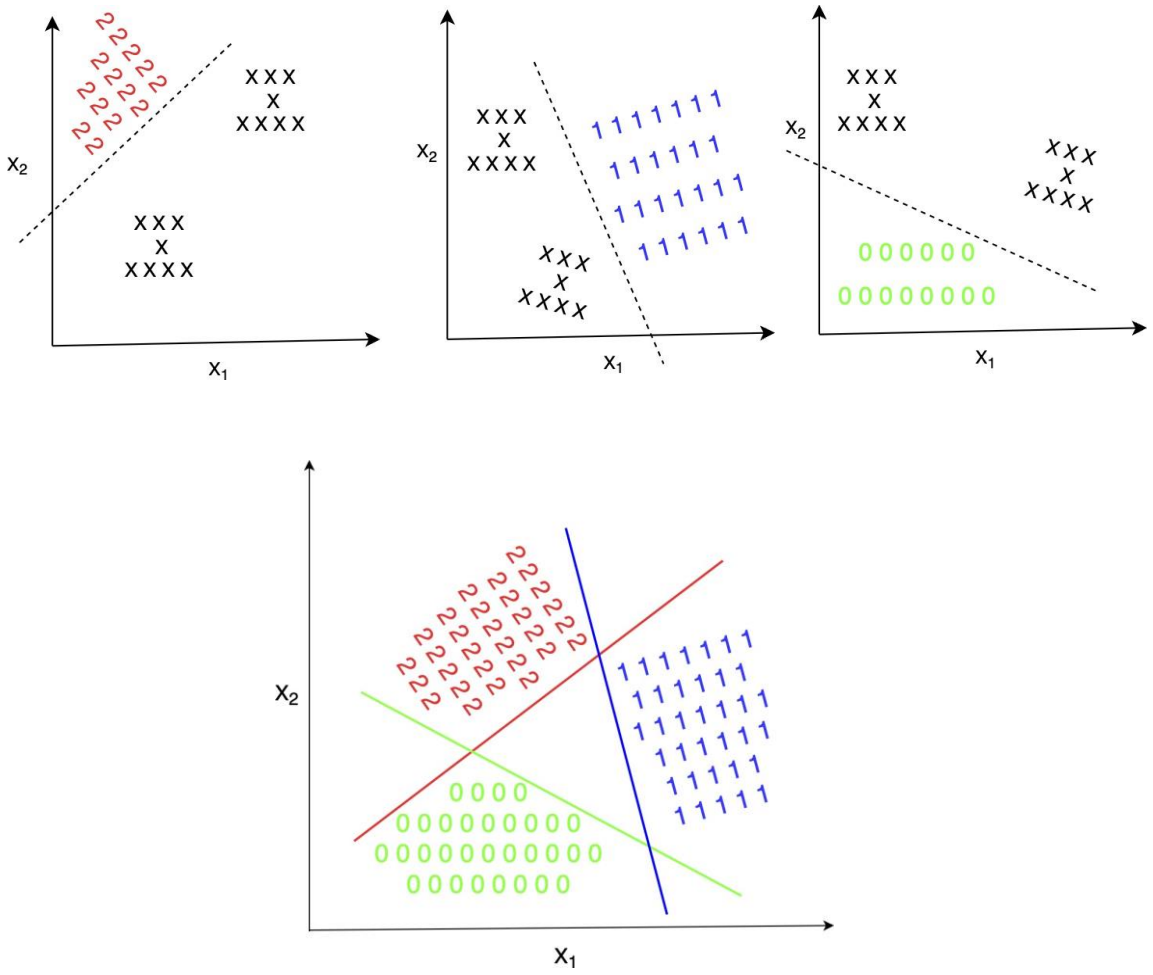
[15] Pant, Ayush. "Introduction to Logistic Regression." Medium, 22 Jan. 2019, towardsdatascience.com/introduction-to-logistic-regression-66248243c148#:~:text=Logistic%20regression%20transforms%20its%20output,to%20return%20a%20probability%20value. Accessed 10 July 2023.

[16] "Just a Moment..." Just a Moment.., machinelearningmastery.com/multinomial-logistic-regression-with-python/#:~:text=Logistic%20regression%20is%20a%20classification,to%20as%20binary%20classification%20problems. Accessed 10 July 2023.

As Logistic Regression carries out relatively simple calculations in comparison to other models, it is extremely quick at classifying unknown records. Moreover, as it draws linear boundaries between the different classes, the risk of overfitting is minimised.

2.5 a Working of Logistic Regression for Password Strength Classification

Figure 5: Logistic Regression for Password Strength Classification



In order to classify the passwords, a One-Vs-Rest classification strategy has been used[17].

[17]GeeksforGeeks | A Computer Science Portal for Geeks, www.geeksforgeeks.org/one-vs-rest-strategy-for-multi-class-classification/. Accessed 10 July 2023.

During Training: Three different binary classifiers are trained for each output class (Weak, Medium and Strong). All three models are trained on the same test data, however the labels of positive or negative differ in each case. Through this, a threshold is calculated of 'Weak' or 'Not Weak', 'Medium' or 'Not Medium' and 'Strong' or 'Not Strong', as seen in the image above.

During Testing: The password is passed through all three models and each model gives out an output score indicating the probability of the password belonging to that class. The password is then predicted to belong to the class with the highest probability.

3. Methodology

3.1 Dataset and Pre-processing

The two models were developed with Google Collaboratory[18] using Python. To train and test the models, the Password Strength Classification dataset from Kaggle was used[19]. The dataset contains a total of 669643 passwords. The CSV file contains the password along with a strength measure equal to 0, 1 or 2, with 0 being weak. There are 496,801 medium (1) passwords, accounting for about 74% of all the passwords in the dataset.

Figure 6: Distribution of Dataset

Class	Labelled As	Count	Ratio
Weak	0	89,702	13.395%
Medium	1	496,801	74.189%
Strong	2	83,137	12.415%
Total		669643	

[18]"Welcome To Colaboratory - Colaboratory." Google Research, https://colab.research.google.com. Accessed 20 July 2023.
[19]"Password Strength Classifier Dataset." Kaggle: Your Machine Learning and Data Science Community, www.kaggle.com/datasets/bhavikbb/password-strength-classifier-dataset. Accessed 20 July 2023.

As the data set is unbalanced with an approximately 1:6:1 ratio, the the accuracy is not enough to establish which algorithm performs better due to the problem of overfitting, hence, the confusion matrix is further used to calculate the F1-score in order to accurately evaluate the models' performances. Additionally, each model's testing time was also noted.

During the pre-processing, the CSV file is loaded into a DataFrame. Then, the number of missing values (NaNs) is identified and hence all missing values are deleted. The remaining data is now converted into a numpy array and shuffled to avoid any biases that may exist in the ordering of the dataset. Using a function for tokenization and the 'fit_transform' method, the list of passwords is converted into a matrix of TF-IDF features.

Figure 7: Data preprocessing code

```
pswd_data = pd.read_csv("/content/data.csv",error_bad_lines=False)
pswd_data.head()

p  Run cell (⌘/Ctrl+Enter)
   cell has not been executed in this session
pswd_data.dropna(inplace=True)
pswd_data.isnull().sum()


pswd = np.array(pswd_data)

random.shuffle(pswd)

ylabels = [s[1] for s in pswd]
allpasswords = [s[0] for s in pswd]

def createTokens(f):
    tokens = []
    for i in f:
      tokens.append(i)
    return tokens
vectorizer = TfidfVectorizer(tokenizer=createTokens)
X = vectorizer.fit_transform(allpasswords)
```

3.2 Programming the Models

For this experiment, the training/testing split had to be the same for a fair comparison. Hence, the train/test split was 80/20% for both models, meaning 535,711 passwords were used for training and the remaining 133,928 for testing.

While programming the Random Forest model, the number of decision trees used in this investigation is 10, specified through the 'n_estiamtors' parameter. The criterion is set as 'entropy', meaning that the algorithm will be using information gain based on entropy of class labels to evaluate splits. With 'random_state = 0' the algorithm sets the random seed to ensure reproducibility.

Figure 8: Programming the Random Forest model

```
classifier = RandomForestClassifier(n_estimators=10, criterion ='entropy', random_state = 0)
```

Similarly, an instance of the 'LogisticRegression' class from scikit-learn[20] is used to develop the Logistic Regression model. This instance, named as 'log_class', is the logistic regression classifier used for predictions

Figure 9: Programming the Logistic Regression model

```
log_class = LogisticRegression ()
```

In each model's program, I measured the performance as follows:

[20]scikit-learn: machine learning in Python — scikit-learn 1.3.0 documentation, https://scikit-learn.org/stable/. Accessed 21 July 2023.

1) **F1 Score:** calculated so that a single value can be used for comparison of the algorithms to take into account a good average of the precision and recall[21]. Hence, the F1 Score indicates a good balance between correctly identifying positive cases and avoiding false positives. A higher F1 Score means better performance[22]. It is calculated for each password strength classification for both the models

$$F1\,score = \frac{2 \times (precision \times recall)}{precision + recall}$$

2) **Prediction Time (seconds):** Measured by the difference between the end and start time for each model's program. The values were calculated through the use of Python's time library, using the time.time() method[23]. The start time was computed after the training and the end time was computed after the testing. Therefore, the lower the prediction time in seconds, the better the performance. The prediction time was calculated as whole for both the models.

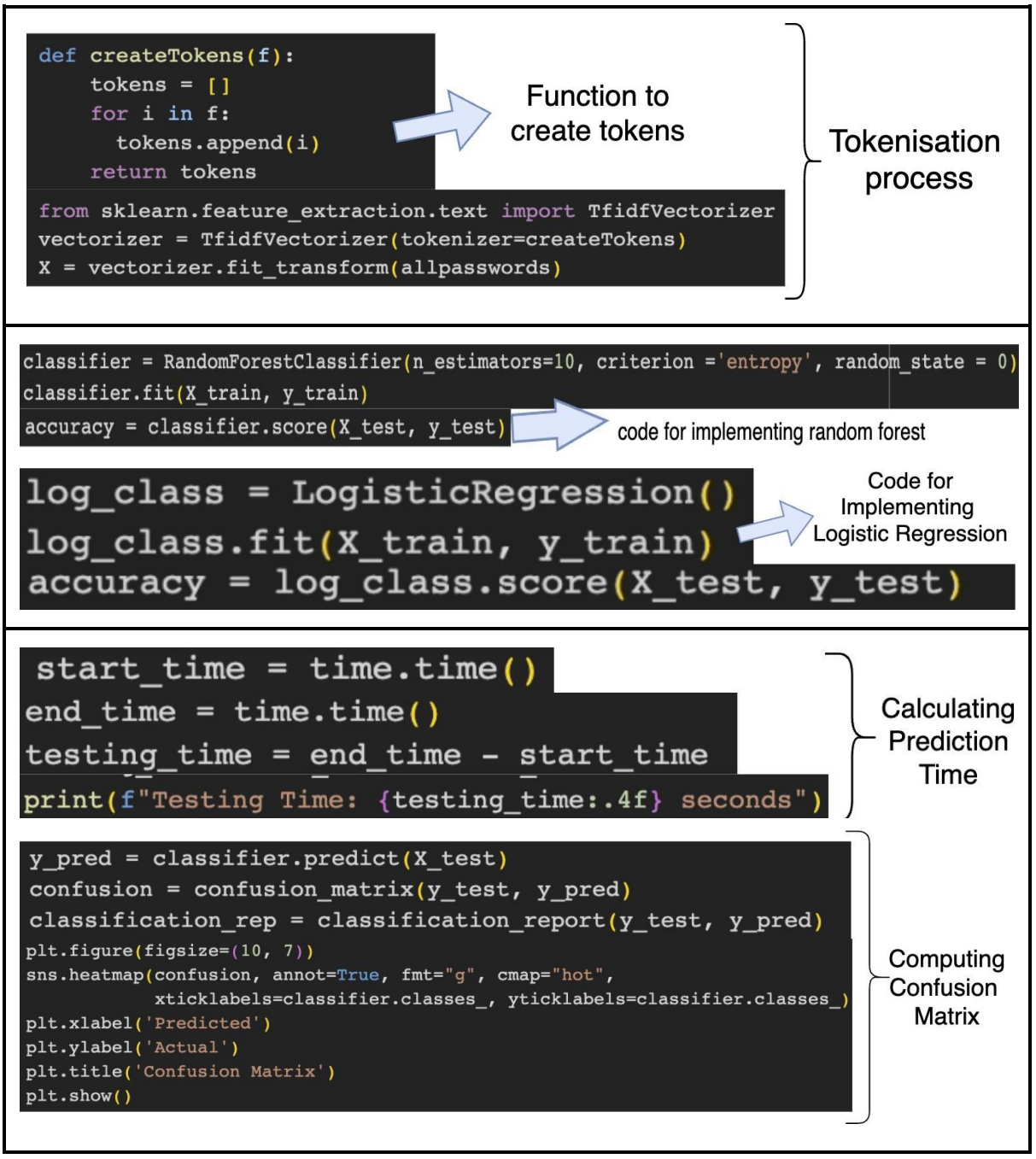
3.3 Experimental Procedure

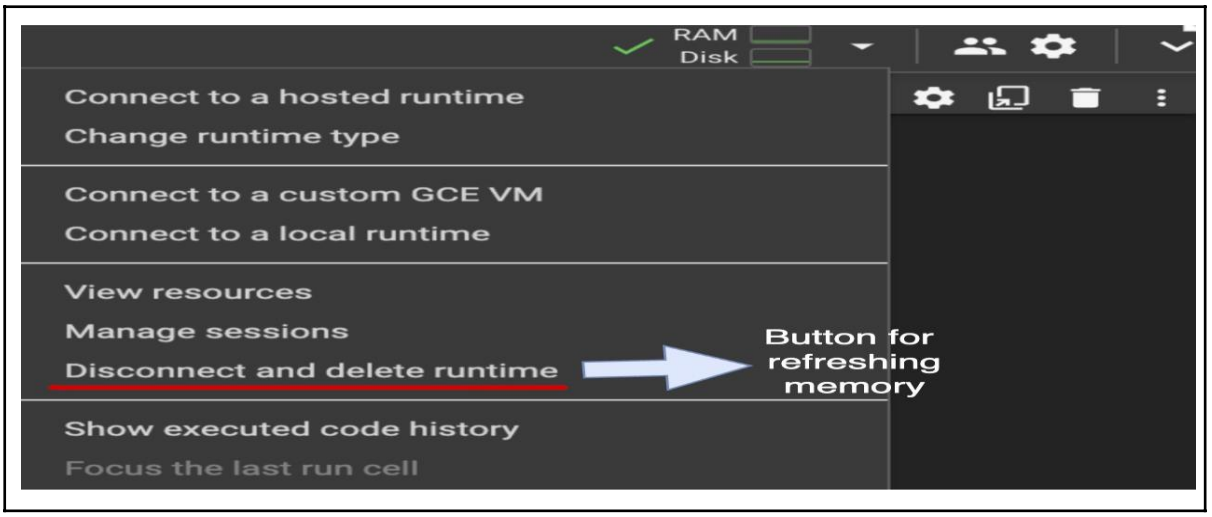
1. Complete TF-IDF vectorisation to transform passwords into a numerical matrix.
2. Execute the code for Random Forest, listing the resultant prediction time and confusion matrix in the observation tables
3. Clear the Google Compute Engine's memory to prevent lower time values in later trials by having the training data in memory
4. Repeat steps 2-3 two more times for trial 2 and trial 3

[21]"F1 Score in Machine Learning: Intro & Calculation." V7 - AI Data Platform for Computer Vision, 1 2023, www.v7labs.com/blog/f1-score-guide. Accessed 20 July 2023.
[22]Allwright, Stephen. "What is a good F1 score? Simply explained (2022)." Stephen Allwright, 20 April 2022, https://stephenallwright.com/good-f1-score/. Accessed 20 July 2023.
[23]"time — Time access and conversions — Python 3.11.4 documentation." Python Docs, https://docs.python.org/3/library/time.html. Accessed 21 July 2023.

5. Calculate the F-1 score for each password strength using the confusion matrix from the three trials and list in the observation tables
6. Repeat steps 2-5 for the Logistic Regression programme

Figure 10: Annotated code corresponding to particular steps in my procedure





4. Experimental Results

4.1 F-1 Score

The table below compares the classification performances of the Random Forest and Logistic Regression with respect to the password strength. All values observed from the program were rounded off to 4 decimal places for higher precision. Three trials were conducted because both the models have randomness in their initialisation and training process; therefore conducting three trials ensures that the evaluation is representative and not influenced by a specific randomisation. This averaged result will, thus, be more indicative of expected real-world performance.

Figure 11: F-1 Score Results

Strength	Random Forest				Logistic Regression			
	Trial 1	Trial 2	Trial 3	Avg.	Trial 1	Trial 2	Trial 3	Avg.
Weak	0.9492	0.9484	0.9511	0.9495	0.3825	0.3963	0.3893	0.3893
Medium	0.9864	0.9856	0.9867	0.9862	0.8854	0.8844	0.8860	0.8852
Strong	0.9703	0.9685	0.9699	0.9695	0.7530	0.7389	0.7560	0.7493

4.2 Prediction Time

The table below compares the prediction time of the Random Forest and Logistic Regression. Each value in the table has been recorded in seconds elapsed during the program execution, these lower values are more favourable as they indicate quicker predictions by the model. For the same reasons as stated under 'Prediction Accuracy', four decimal places have been used and three trails have been conducted.

Figure 12: Prediction Time Results

Time	Random Forest				Logistic Regression			
	Trial 1	Trial 2	Trial 3	Avg.	Trial 1	Trial 2	Trial 3	Avg.
	0.4374	0.6834	0.9799	0.7002	0.0443	0.0440	0.0436	0.0439

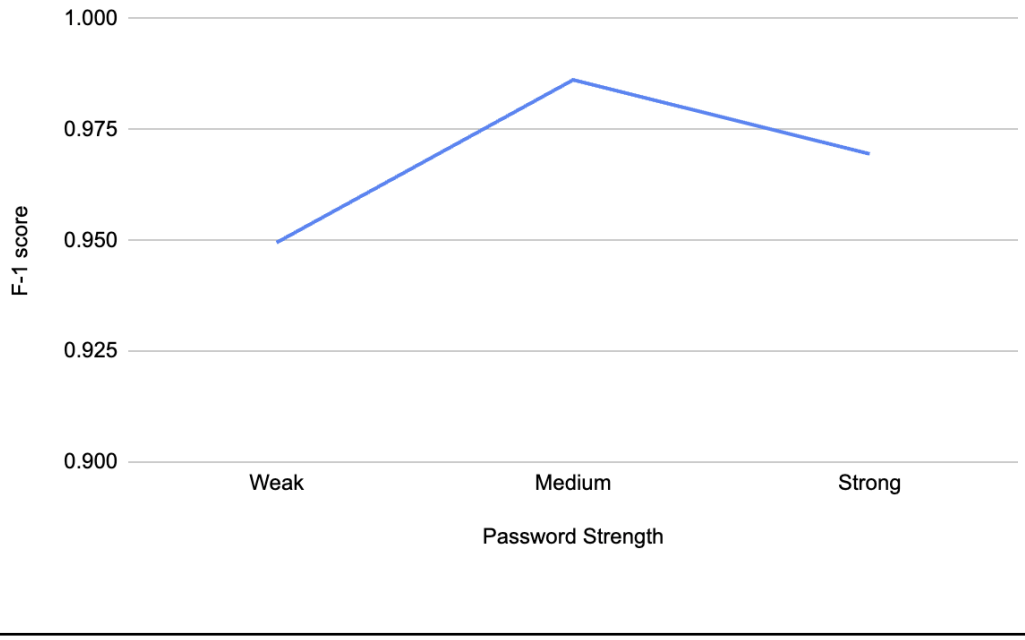
4.3 Consolidated Performance Results

Figure 13: Consolidated results

Random Forest		Logistic Regression	
Macro F-1 Score	Time/s	Macro F-1 Score	Time/s
0.9684	0.7002	0.6746	0.0439

4.4 Analysing Random Forest Performance

Figure 14: Random Forest Performance



Prediction Time: 0.7002 seconds

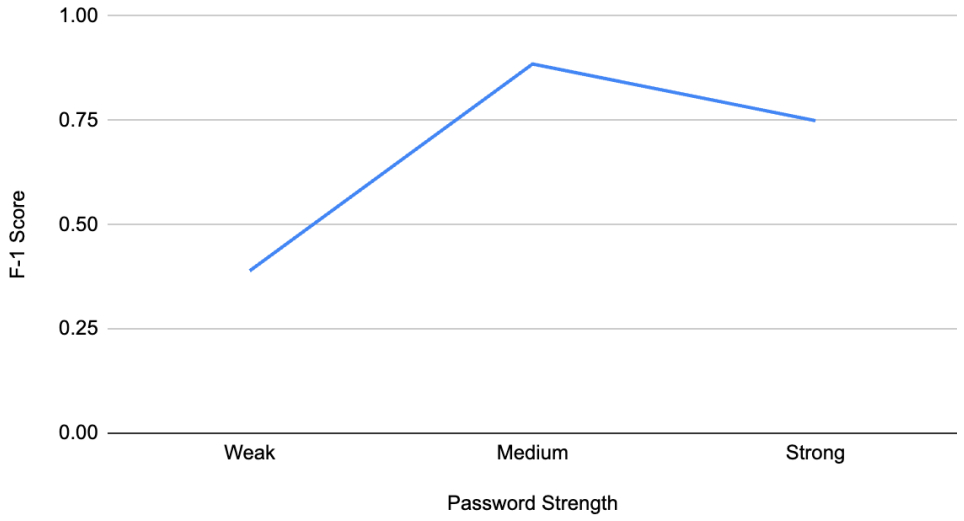
My experiment indicates that Random Forest can predict password strength with very good accuracy with a Macro F1 Score of 0.9684. The accuracy seems to increase as password strength increases, however after medium strength there seems to be diminishing return in accuracy, this may be due to the imbalance in the dataset. Nevertheless, the overall accuracy from weak to strong increases from 0.9495 to 0.9695, therefore, the impact of the unbalanced dataset on the model is minor. This may be the case as Random Forest can effectively handle imbalanced data by assigning higher weights to the minority class[24] during training, thus improving classification of weak passwords. This property combined with its ensemble nature are likely the reason for high accuracy.

[24]"Surviving in a Random Forest with Imbalanced Datasets | by Kyoun Huh | SFU Professional Computer Science." Medium, 13 February 2021, https://medium.com/sfu-cspmp/surviving-in-a-random-forest-with-imbalanced-datasets-b98b963d52eb. Accessed 3 August 2023..

However, the pros of the Random Forest are also likely the reason for its poor time performance[25] of 0.7002 seconds, a 16 times slower performance in comparison to Logistic Regression. Its ensemble nature constructs 10 decision trees, each which requires time for optimisation and growth followed by a voting process.

4.5 Analysing Logistic Regression Performance

Figure 15: Logistic Regression Performance



Once again, the results indicate the same pattern of increasing and decreasing accuracy, however the variations in Logistic Regression are much more significant. With a macro F1 score of 0.6746, the Logistic Regression model has an average accuracy. This may be due to the fact Logistic Regression handles categorical features by encoding them[26], however, in the case of passwords, the categorical variables have complex relationships, making it difficult for the model to encode them. The unbalanced dataset may also be a huge contributor to diminishing performance as unlike Random Forest, Logistic Regression can be biassed

[25] "Random forest Algorithm in Machine learning: An Overview." Great Learning, https://www.mygreatlearning.com/blog/random-forest-algorithm/. Accessed 3 August 2023.
[26]Roy, Baijayanta. "All about Categorical Variable Encoding | by Baijayanta Roy." Towards Data Science, https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02. Accessed 3 August 2023.

towards the majority class[27] and struggle to predict the minority classes as seen with the lower F1 score for weak (0.3893) and strong passwords (0.7493). Despite its accuracy limitations, Logistic Regression took the lead in prediction time with just 0.0439 seconds.

5. Conclusion

In this paper, the performance of Random Forest and Logistic Regression was compared using the prediction accuracy and prediction time for classification of password strength. While the inputted dataset for both the models was identical, differences in general trends for prediction accuracy and prediction time emerged.

Overall, my experiment shows that Random Forest performs with better accuracy whereas Logistic Regression performs with better prediction time. Therefore, a performance-time tradeoff exists in both the models for password strength classification. Hence, the selection of the model depends on the stakeholder's priority. In case of lower computational resources and large time constraints, Logistic Regression would be a better choice, whereas where accuracy is the deciding factor, Random Forest should be used. Hopefully, this paper will prove useful to developers who are looking to incorporate machine learning models for password strength classification in their projects.

[27]"Issues using logistic regression with class imbalance, with a case study from credit risk modelling." American Institute of Mathematical Sciences, https://www.aimsciences.org/article/doi/10.3934/fods.2019016. Accessed 3 August 2023.

6. Bibliography

“Real Time Password Strength Analysis on a Web Application Using Multiple Machine Learning Approaches – IJERT.” International Journal of Engineering Research & Technology, 24 December 2020, https://www.ijert.org/real-time-password-strength-analysis-on-a-web-application-using-multiple-machine-learning-approaches. Accessed 2 July 2023.

LastPass. "How Secure is Your Password?" LastPass | Something Went Wrong, lastpass.com/howsecure.php. Accessed 2 July 2023.

PasswordMonster, 3 Mar. 2022, www.passwordmonster.com. Accessed 3 July 2023.

"Mage." Give Your Data Team Magical Powers | Mage, www.mage.ai/blog/how-long-to-build-ml-model. Accessed 3 July 2023.

Educative. "Mage." Give Your Data Team Magical Powers | Mage, www.mage.ai/blog/how-long-to-build-ml-model. Accessed 5 July 2023.

Microsoft. "Create and Use Strong Passwords." support.microsoft.com/en-us/windows/create-and-use-strong-passwords-c5cebb49-8c53-4f5e-2bc4-fe357ca048eb. Accessed 5 July 2023.

"Decision Tree Classification Algorithm." Www.javatpoint.com, www.javatpoint.com/machine-learning-decision-tree-classification-algorithm. Accessed 10 July 2023.

“What is a Decision Tree.” IBM, https://www.ibm.com/topics/decision-trees. Accessed 10 July 2023.

“Decision Tree Algorithm in Machine Learning.” Javatpoint, https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm. Accessed 10 July 2023.

"Linear Regression in Machine Learning - Javatpoint." Www.javatpoint.com, www.javatpoint.com/linear-regression-in-machine-learning. Accessed 10 July 2023.

"Linear Regression in Machine Learning - Javatpoint." Www.javatpoint.com, www.javatpoint.com/linear-regression-in-machine-learning. Accessed 10 July 2023.

"Random Forest Algorithm." Www.javatpoint.com, www.javatpoint.com/machine-learning-random-forest-algorithm. Accessed 10 July 2023.

R, Sruthi E. "Random Forest | Introduction to Random Forest Algorithm." Analytics Vidhya, 21 June 2022, www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/. Accessed 10 July 2023.

"Logistic Regression in Machine Learning." Www.javatpoint.com, www.javatpoint.com/logistic-regression-in-machine-learning. Accessed 10 July 2023.

Pant, Ayush. "Introduction to Logistic Regression." Medium, 22 Jan. 2019, towardsdatascience.com/introduction-to-logistic-regression-66248243c148#:~:text=Logistic%20regression%20transforms%20its%20output,to%20return%20a%20probability%20value. Accessed 10 July 2023.

"Just a Moment..." Just a Moment.., machinelearningmastery.com/multinomial-logistic-regression-with-python/#:~:text=Logistic%20regression%20is%20a%20classification,to%20as%20binary%20classification%20problems. Accessed 10 July 2023.

GeeksforGeeks | A Computer Science Portal for Geeks, www.geeksforgeeks.org/one-vs-rest-strategy-for-multi-class-classification/. Accessed 10 July 2023.

“Welcome To Colaboratory - Colaboratory.” Google Research, https://colab.research.google.com. Accessed 20 July 2023.

"Password Strength Classifier Dataset." Kaggle: Your Machine Learning and Data Science Community, www.kaggle.com/datasets/bhavikbb/password-strength-classifier-dataset. Accessed 20 July 2023.

scikit-learn: machine learning in Python — scikit-learn 1.3.0 documentation, https://scikit-learn.org/stable/. Accessed 21 July 2023.

"F1 Score in Machine Learning: Intro & Calculation." V7 - AI Data Platform for Computer Vision, 1 2023, www.v7labs.com/blog/f1-score-guide. Accessed 20 July 2023.

Allwright, Stephen. “What is a good F1 score? Simply explained (2022).” Stephen Allwright, 20 April 2022, https://stephenallwright.com/good-f1-score/. Accessed 20 July 2023.

“time — Time access and conversions — Python 3.11.4 documentation.” Python Docs, https://docs.python.org/3/library/time.html. Accessed 21 July 2023.

“Surviving in a Random Forest with Imbalanced Datasets | by Kyoun Huh | SFU Professional Computer Science.” Medium, 13 February 2021, https://medium.com/sfu-cspmp/surviving-in-a-random-forest-with-imbalanced-datasets-b98b963d52eb. Accessed 3 August 2023..

“Random forest Algorithm in Machine learning: An Overview.” Great Learning, https://www.mygreatlearning.com/blog/random-forest-algorithm/. Accessed 3 August 2023.

Roy, Baijayanta. “All about Categorical Variable Encoding | by Baijayanta Roy.” Towards Data Science, https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02. Accessed 3 August 2023.

“Issues using logistic regression with class imbalance, with a case study from credit risk modelling.” American Institute of Mathematical Sciences, https://www.aimsciences.org/article/doi/10.3934/fods.2019016. Accessed 3 August 2023.