Exploring Password Complexity: A Comparative Analysis of Supervised and Unsupervised Learning Approaches for Password Strength Classification

Brianna Schuh • Advised by Professor Priya Panda • Yale University • Department of Statistics and Data Science • May 8th, 2023

Motivation

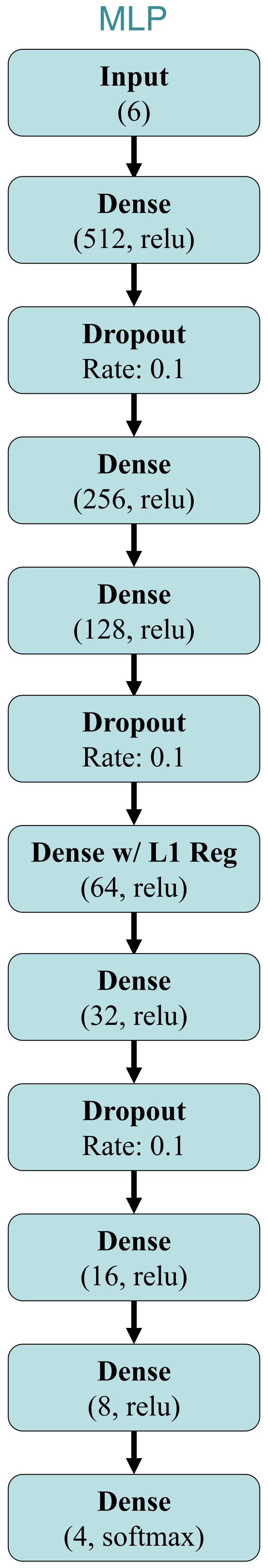
So far in 2023, 17 major companies have reported data breaches. Although many companies store password encryptions instead of raw passwords, these breaches still pose a threat to account holders. Attackers can access their personal information if they are able to crack the password. Companies use algorithms to determine password strength and derive their password policy accordingly. In 2012, Dropbox released their zxcvbn[1], their own algorithm for classifying passwords. In this project, we examine password complexity as defined by zxcvbn through the use of supervised and unsupervised methods.

Dataset and Algorithm

Zxcvbn is an algorithm that classifies passwords into one of five categories: very weak, weak, medium, strong, and very strong. These classifications are based on the presence of dictionary words, patterns, sequences, and entropy. For our dataset, we randomly sampled 700,000 passwords from the 2015 000webhost data breach and classified each password strength by zxcvbn. Less than 3,000 passwords from our dataset were labeled as very weak, so we merged the very weak and weak classes. We extracted the following features: Shannon entropy, n-gram frequency (n = 2, 3, 4), Levenshtein distance, character repetition weight sum, most common character set count, character frequency ratio, and password length to unique ratio.

Supervised Models

Architecture of Supervised Models


MLP

Input (6)

Dense (512, relu)

Dropout Rate: 0.1

Dense (256, relu)

Dense (128, relu)

Dropout Rate: 0.1

Dense w/ L1 Reg (64, relu)

Dense (32, relu)

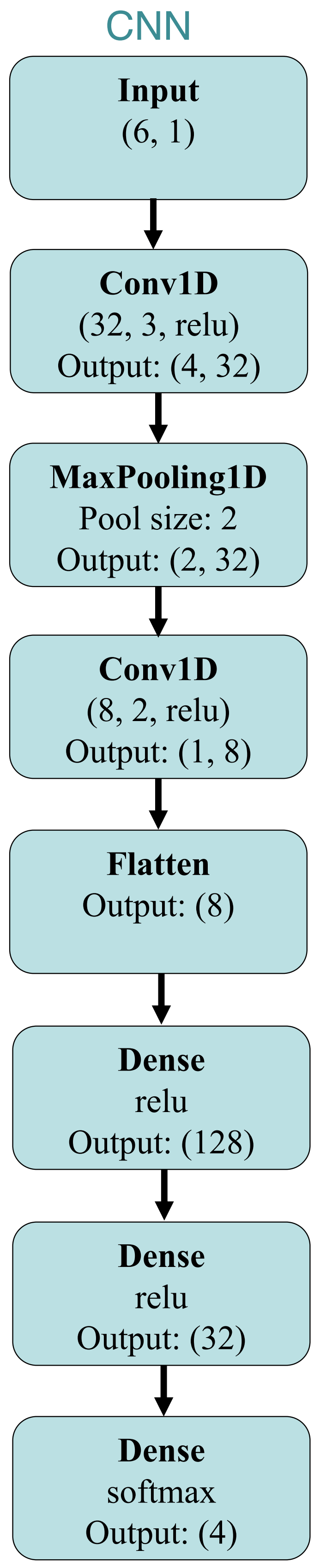
Dropout Rate: 0.1

Dense (16, relu)

Dense (8, relu)

Dense (4, softmax)

Figure 1. MLP Architecture


CNN

Input (6, 1)

Conv1D (32, 3, relu) Output: (4, 32)

MaxPooling1D Pool size: 2 Output: (2, 32)

Conv1D (8, 2, relu) Output: (1, 8)

Flatten Output: (8)

Dense relu Output: (128)

Dense relu Output: (32)

Dense softmax Output: (4)

Figure 2. CNN Architecture

Shannon entropy and Levenshtein distance hold a lot of importance in both models.

Results of Supervised Models

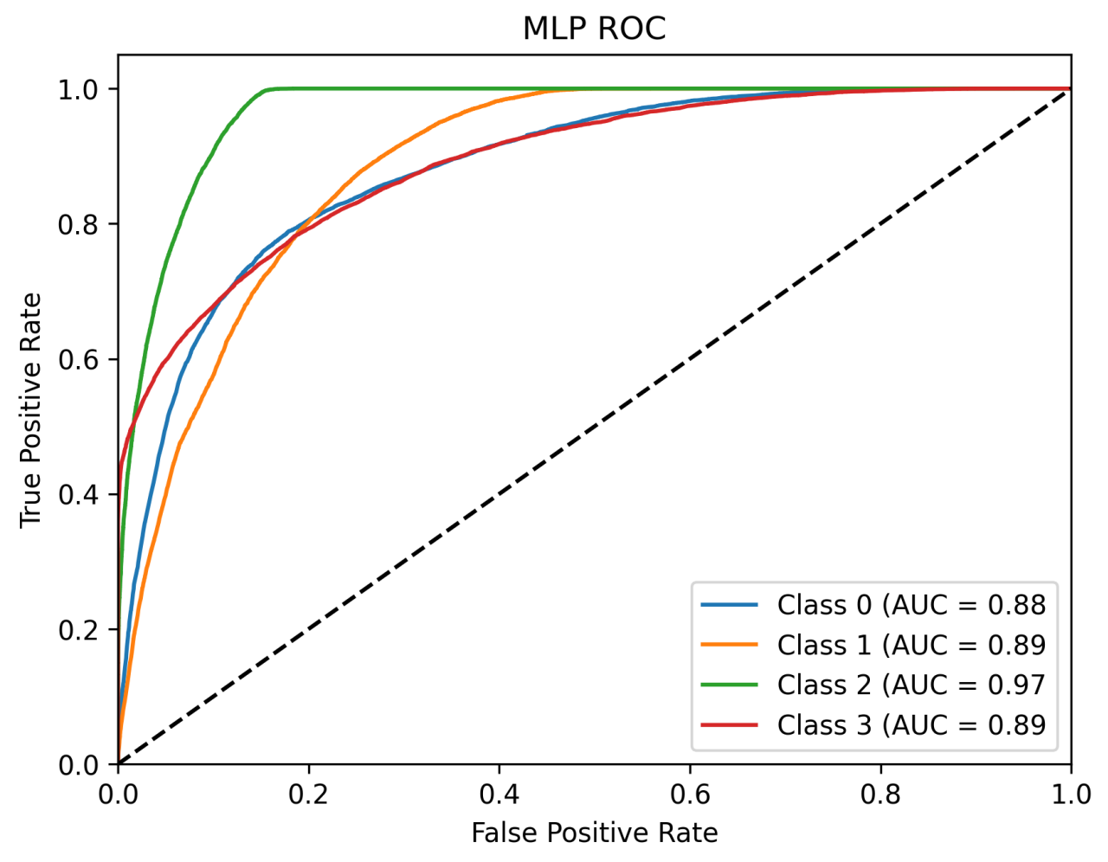


Figure 3. MLP ROC

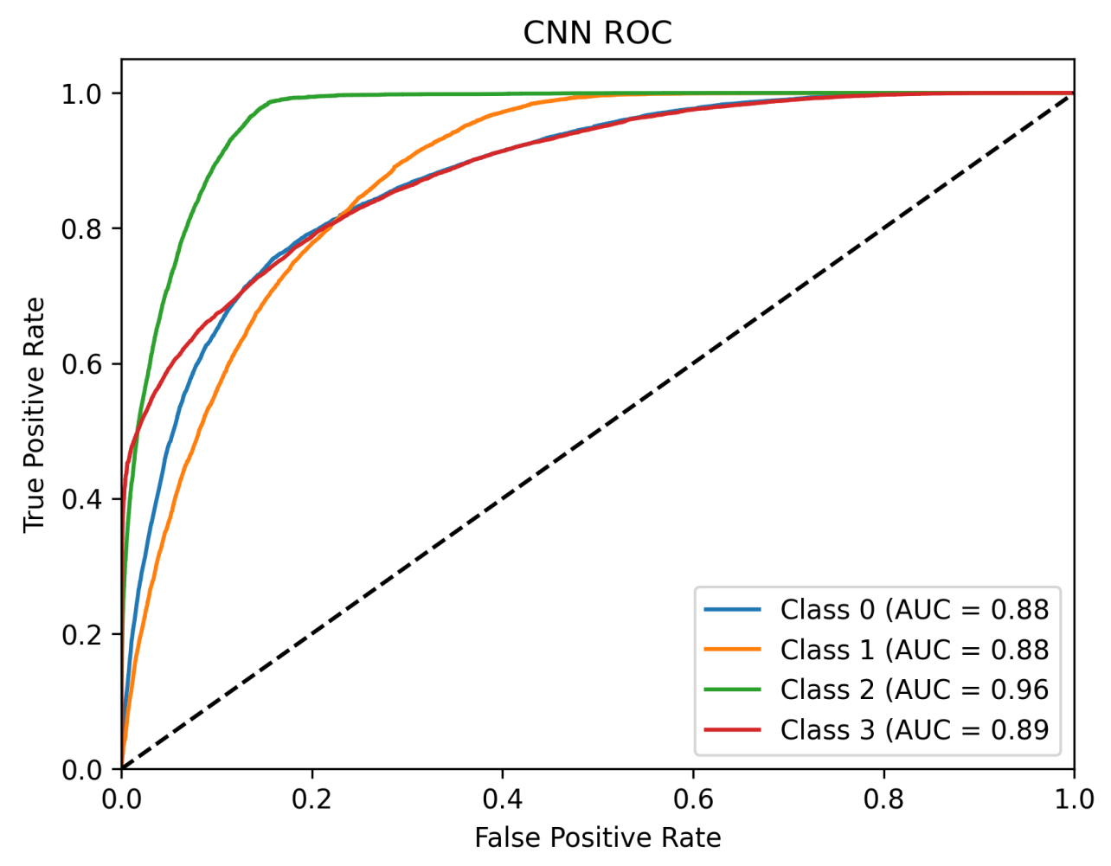


Figure 4. CNN ROC

Both models had a macro-average AUC of 0.90, which means that both can effectively discriminate between different password strengths.

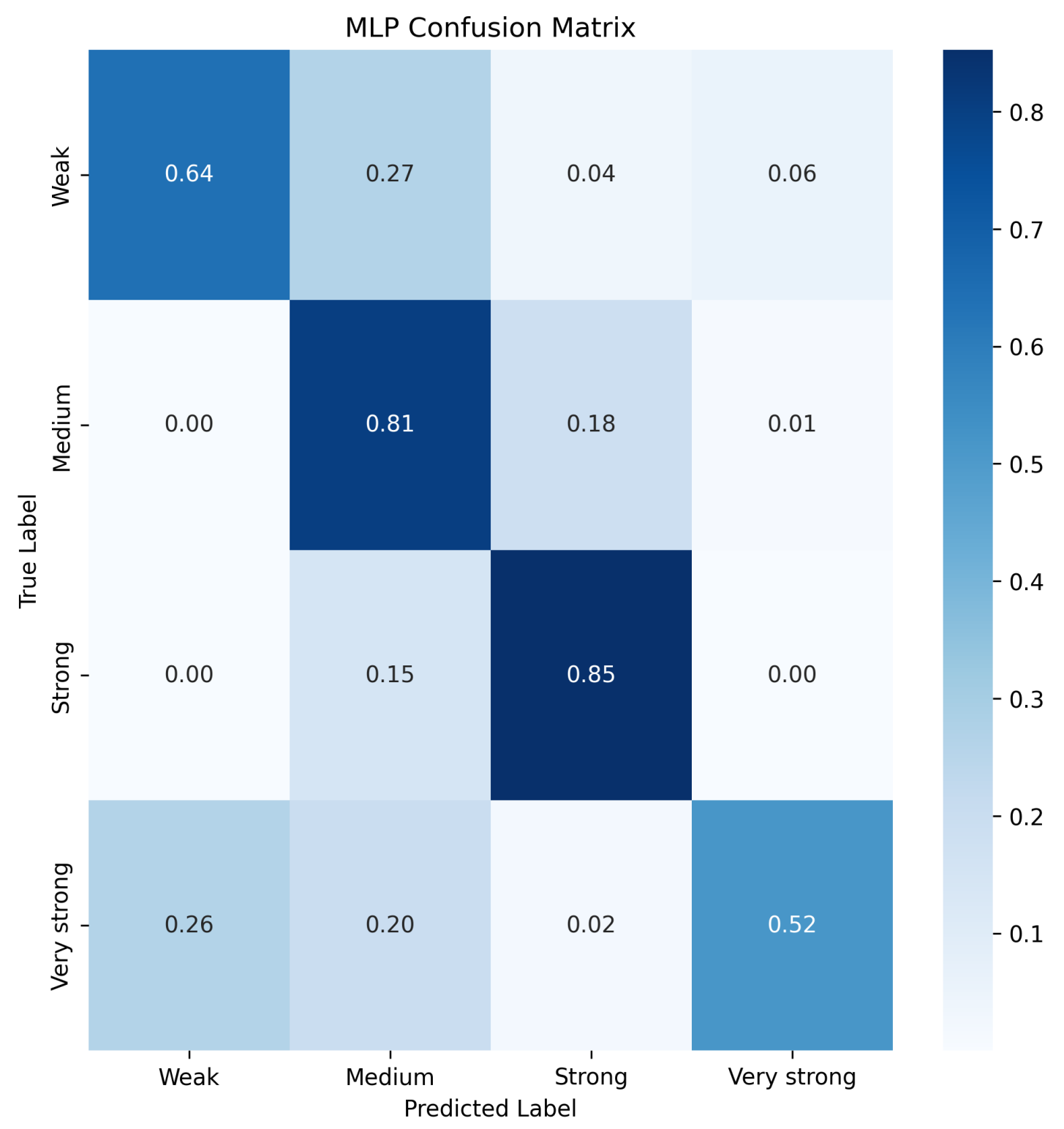


Figure 5. MLP Confusion Matrix

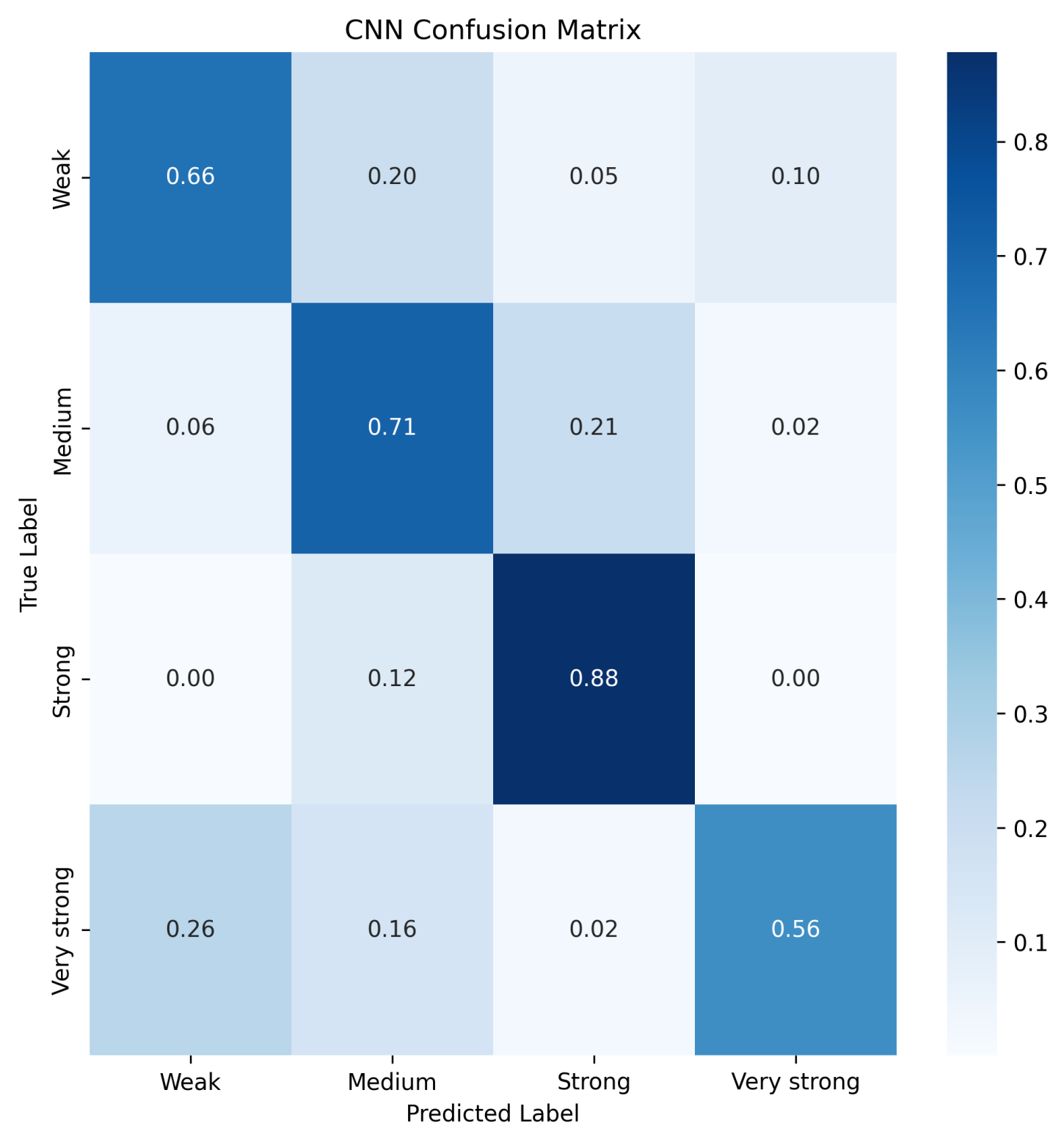


Figure 6. CNN Confusion Matrix

Both models showed a similar classification trend in their confusion matrices. The most interesting observation is that both models often misclassified very strong passwords as weak. However, the converse does not hold.

Rank	Feature	Importance
1	Four-gram frequency	125.99
2	Levenshtein distance	123.45
3	Trigram frequency	115.98
4	Shannon entropy	106.79
5	Password length to unique ratio	106.11
6	Bigram frequency	72.76
7	Character frequency ratio	46.56
8	Most common character type count	41.93
9	Character repetition weight sum	25.43

Figure 7. MLP Feature Rankings

Rank	Feature	Importance
1	Shannon entropy	0.293
2	Character frequency ratio	0.152
3	Most common character type	0.125
4	Levenshtein distance	0.100
5	Character repetition weight sum	0.038
6	Bigram frequency	0.034
7	Trigram frequency	0.007
8	Four-gram frequency	0.002
9	Password length to unique ratio	0.000

Figure 8. CNN Feature Rankings

Unsupervised Models

We performed k-means so we can better understand the underlying patterns in our data. To determine the optimal number of clusters, k, we used the elbow method. We computed the sum of square errors (SSE) for each k, and plotted them against their respective cluster. Our scree plot revealed two elbow points, one at k = 2, and another at k = 4. We performed k-means at both points.

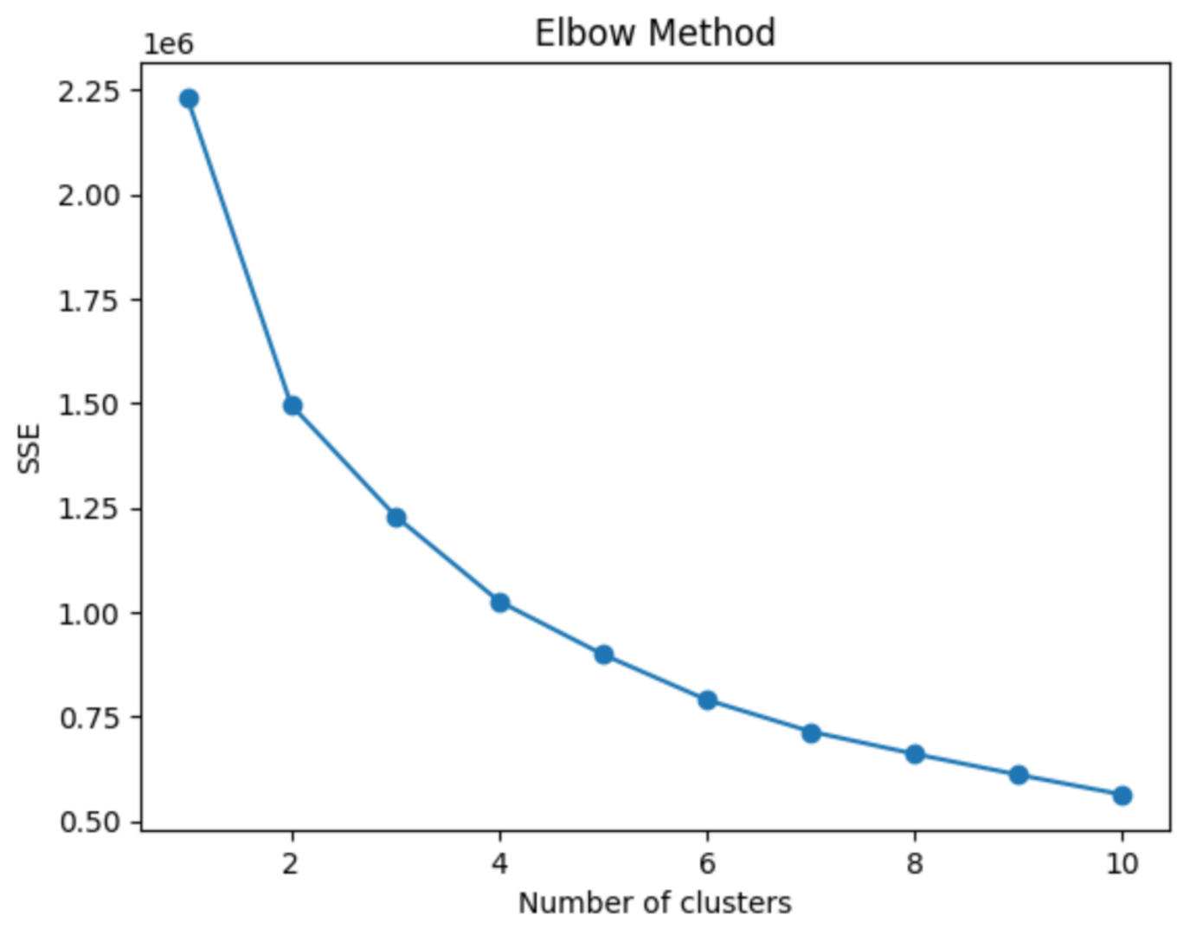


Figure 9. Scree Plot of K-means

K-Means (k = 2)

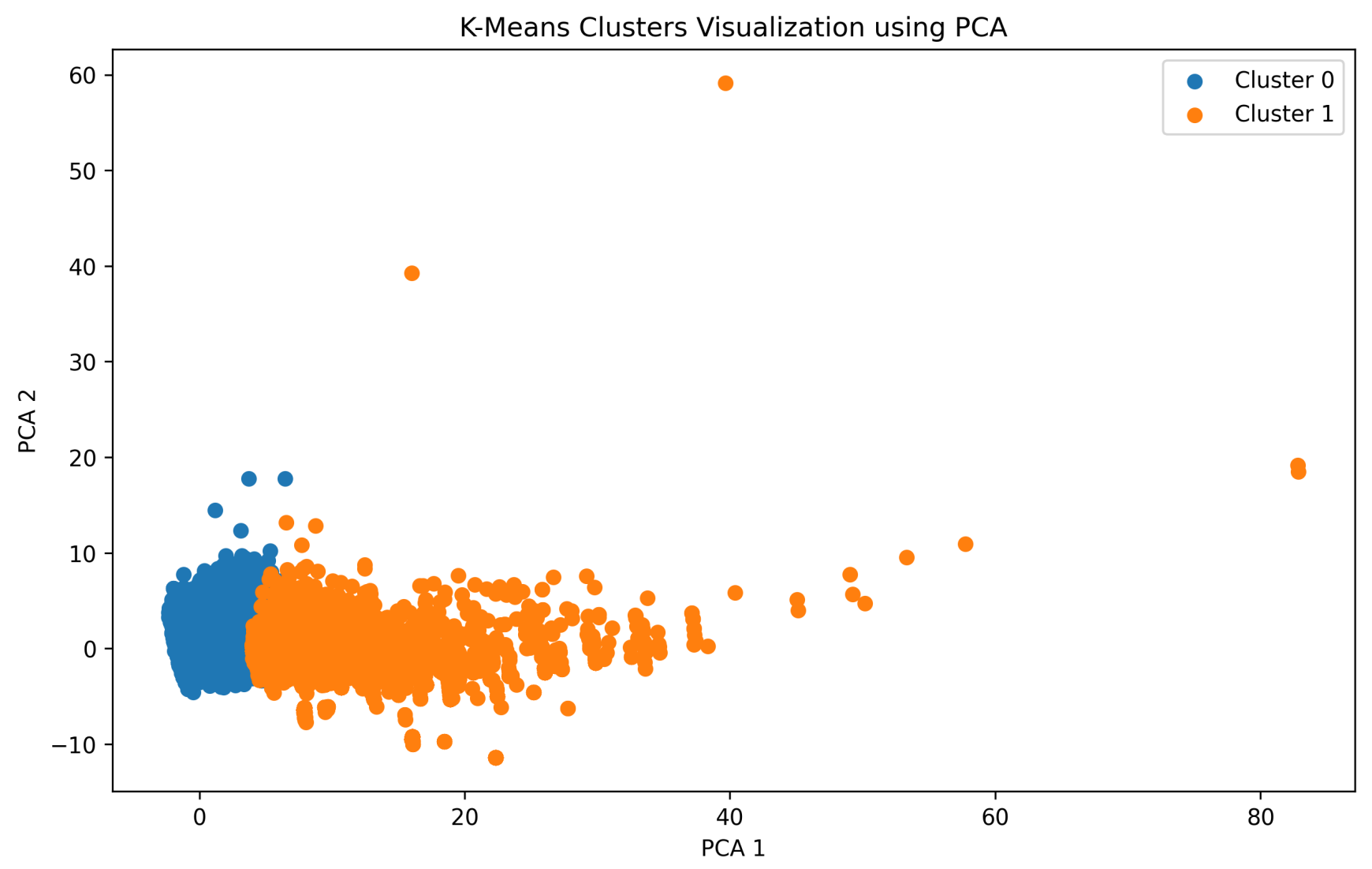


Figure 10. K-means Clusters Using PCA (k = 2)

Cluster	Weak	Medium	Strong	Very Strong
0	109681	114497	114905	115078
1	6869	2053	1645	1472

Figure 11. Distribution of zxcvbn classes within each cluster (k = 2)

Metric	Score
Silhouette Score	0.7321
Davies-Bouldin Score	0.9503
Adjusted Rand Index (ARI)	0.0004
Normalized Mutual Information (NMI)	0.0088
Adjusted Mutual Information (AMI)	0.0088

Figure 12. K-means Cluster Quality Metrics (k = 2)

Figure 10 reveals that the clusters do not appear to be distinct, at least in a two-dimensional space. Moreover, Figure 11 suggests that the clusters are not well-defined. Cluster 1 contains mostly weak passwords, but that cluster is significantly smaller in size compared to Cluster 0. Cluster 0 contains a variety of password strengths. The Silhouette score and the Davies-Bouldin score suggest that the clusters are well separated and cohesive. However, the ARI, NMI, and AMI reveal that the clusters are not well-defined, which aligns with Figure 11.

K-Means (k = 4)

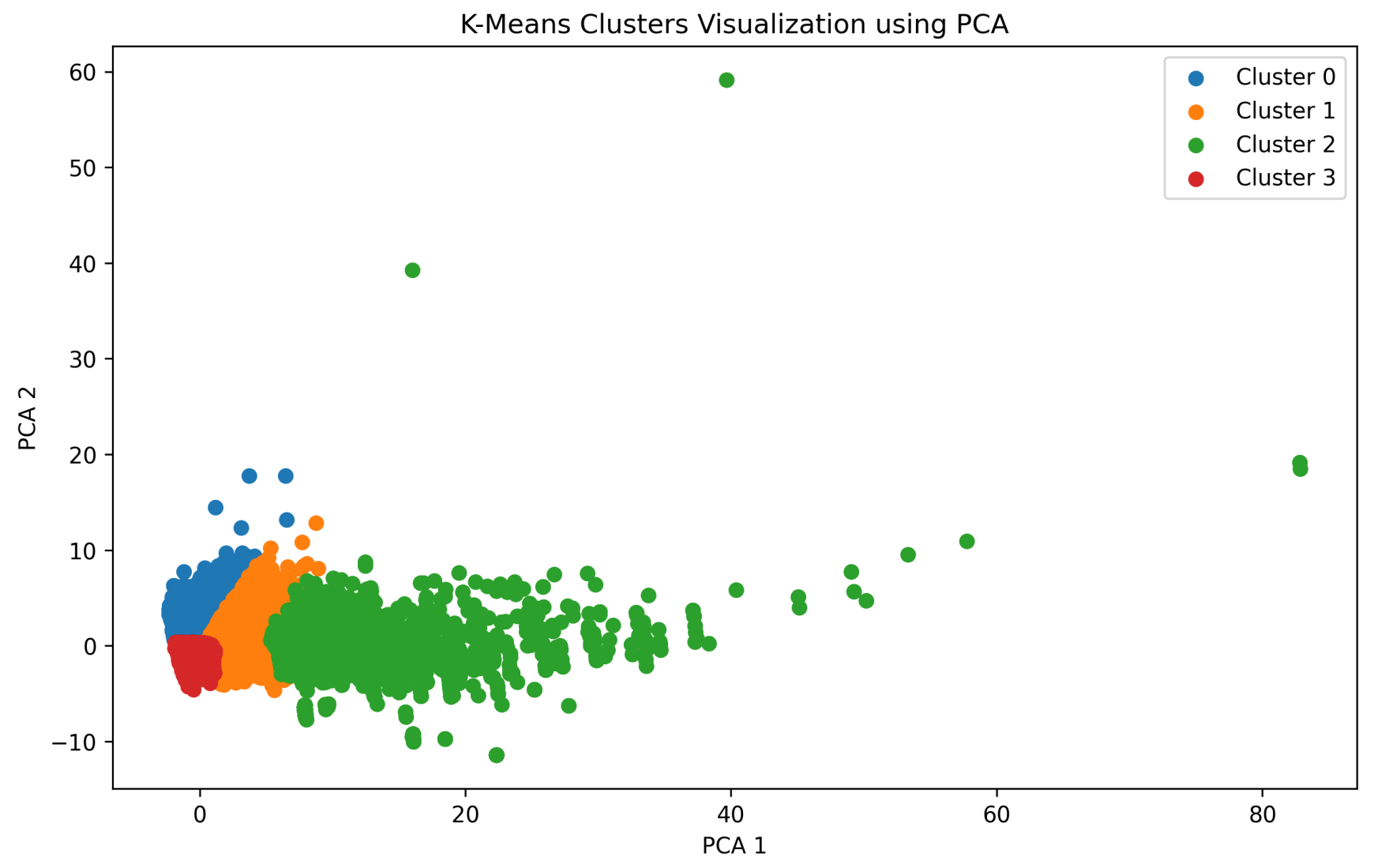


Figure 13. K-means Clusters Using PCA (k = 4)

Cluster	Weak	Medium	Strong	Very Strong
0	6519	9310	41703	98096
1	14200	13791	15421	11152
2	4671	902	801	636
3	91160	92547	58625	6666

Figure 14. Distribution of zxcvbn classes within each cluster (k = 2)

Metric	Score
Silhouette Score	0.2351
Davies-Bouldin Score	1.2822
Adjusted Rand Index (ARI)	0.2077
Normalized Mutual Information (NMI)	0.2227
Adjusted Mutual Information (AMI)	0.2227

Figure 15. K-means Cluster Quality Metrics (k = 2)

Figure 13 reveals that the clusters appear to be less distinct than in the case of k = 2, at least in a two-dimensional space. However, Figure 11 suggests that the clusters are more well-defined. Cluster 0 contains mostly strong and very strong passwords. Cluster 1 contains a variety of password strengths. Cluster 2 contains primarily weak passwords. Cluster 3 mostly has weak and medium passwords. The Silhouette score and the Davies-Bouldin score suggest that the clusters are not well separated. However, the ARI, NMI, and AMI reveal that the clusters are well-defined, which makes sense given that there are four classes in the actual dataset, so when k = 4, the clusters are able to better represent the distribution of the classes.

Conclusion

The insights derived from both supervised and unsupervised models complement each other. The supervised methods provided more interpretable results; it was clear to see which features were seen as more important for either one. Unsupervised approaches provided a more exploratory understanding of the data. Patterns within the clusters could be detected, which can be used to better understand the similarities and differences across password strengths. Based on the findings of k-means, there was not a linear relationship between the centroids and password strengths. This implied that the patterns are no—trivial, which suggested that deep learning could be appropriate since they are generally used for higher dimensional data. Further research should be done to investigate why both supervised models often misclassified very strong passwords as weak. Although the converse did not hold, misclassifications could cause a serious security threat. Also, more research should be done on CNNs. CNNs are typically used for image classification since they can work with sequential data. However, this research suggested that CNNs could perhaps be used as a more sophisticated password classification tool.

References

[1] Wheeler, Dan. "Zxcvbn: Realistic Password Strength Estimation." Dropbox, https://dropbox.tech/security/zxcvbn-realistic-password-strength-estimation.