ASSESSING PASSWORD SECURITY USING MACHINE LEARNING FOR CYBERSECURITY

19th Annual City Tech Poster Session

Ryjll Morris, Michael Bennett, Aparicio Carranza

Computer Engineering Technology

ABSTRACT

Password cracking is a method used to obtain or discover a secure password. It can be used in cases of a forgotten password or a locked-out account. However, it is widely used for malicious intent. This includes gaining unauthorized access to a system to obtain private/personal information. To prevent these attacks and properly secure an account, a strong, unique password is necessary. We demonstrate how Machine Learning can be used to assess password security. We employ the scikit python library (MB) to read a dataset of passwords into the model's data frame, and the Python data package pandas (RM) to train the model to recognize and analyze basic combinations of characters. We create four passwords, with varying combinations and use Natural Language Processing (NLP), to compare them to the data base and determine their strength.

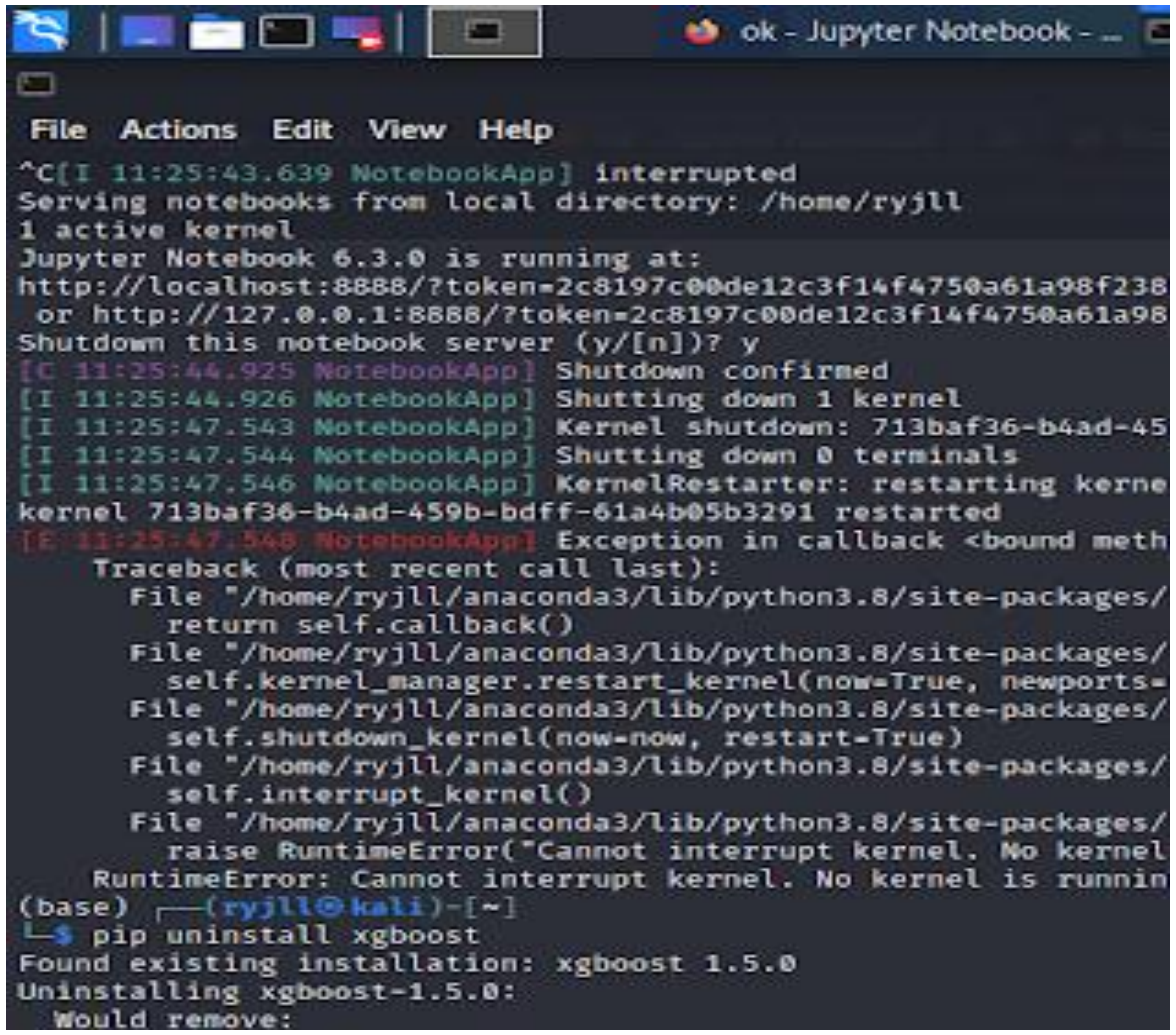
INTRODUCTION

- **Using Machine Learning, a password can be assessed for its strength by comparing it to a list of passwords in a dataset and documenting how hard it was for the system to crack.**
- **We employed the Machine Learning Library Scikit-Learn (aka sklearn) to retain the ability to incorporate the simplicity and power of Python.**
- **Pandas, also known as 'Python Data Analysis Library', is also a python library that equips the user with high-performance data manipulation and analysis tools as well as powerful data structures.**
- **Various Python commands allow for the libraries and data packages mentioned above to be accessed and manipulated in order to achieve accurate password assessments.**
- **The ideal result of this project is the model's prediction of how strong each password is.**

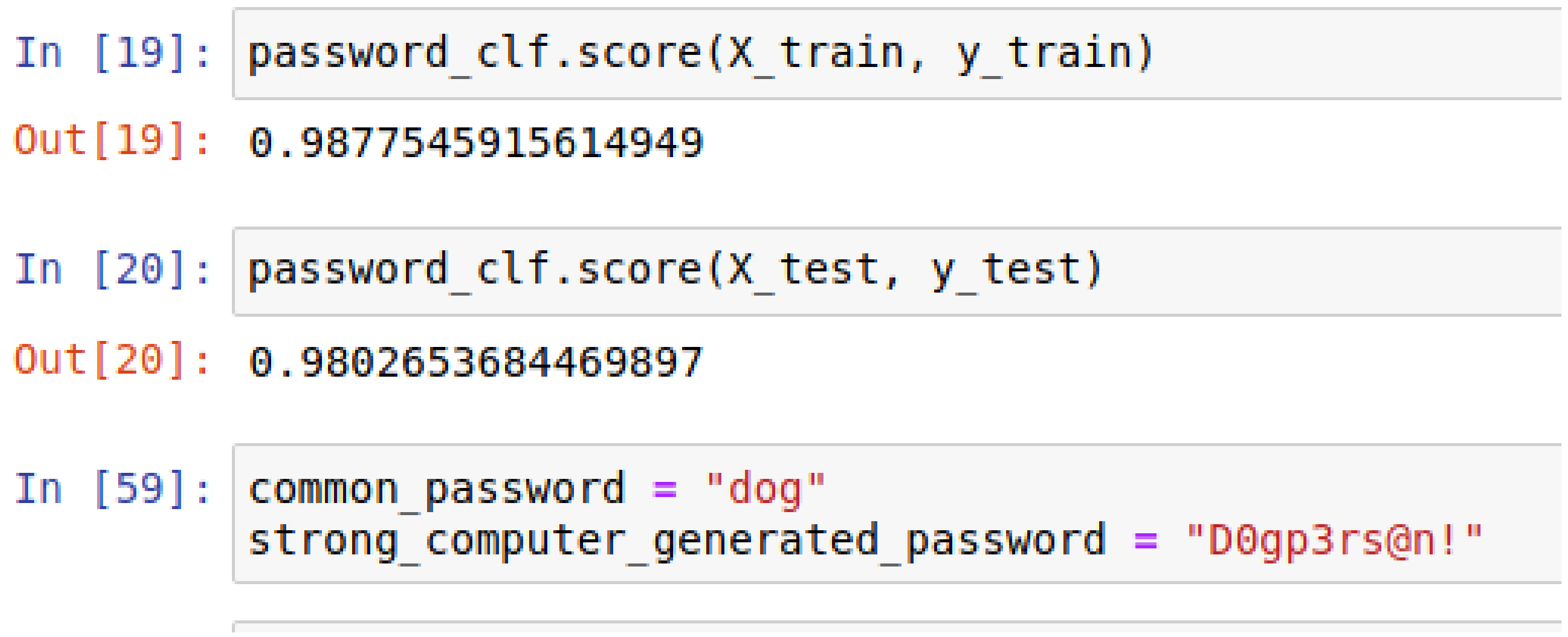
COMMANDS

- **import pandas as pd**
- **df = df.sample(frac=1)**
- **password_clf = Pipeline(**
  [("vect", TfidfVectorizer(tokenizer=character_tokens)), ("clf",
  XGBClassifier()),])
- **common_password = "qwerty"**
  strong_computer_generated_password =
  "c9lCwLBFmdLbG6iWla4H"
- **password_clf.predict([common_password,**
  strong_computer_generated_password])

RESULTS



```
^C[I 11:25:43.639 NotebookApp] interrupted
Serving notebooks from local directory: /home/ryjll
1 active kernel
Jupyter Notebook 6.3.0 is running at:
http://localhost:8888/?token=2c8197c00de12c3f14f4750a61a98f238
 or http://127.0.0.1:8888/?token=2c8197c00de12c3f14f4750a61a98
Shutdown this notebook server (y/[n])? y
[C 11:25:44.925 NotebookApp] Shutdown confirmed
[I 11:25:44.926 NotebookApp] Shutting down 1 kernel
[I 11:25:47.543 NotebookApp] Kernel shutdown: 713baf36-b4ad-45
[I 11:25:47.544 NotebookApp] Shutting down 0 terminals
[I 11:25:47.546 NotebookApp] KernelRestarter: restarting kerne
kernel 713baf36-b4ad-459b-bdff-61a4b05b3291 restarted
[E 11:25:47.548 NotebookApp] Exception in callback <bound meth
    Traceback (most recent call last):
      File "/home/ryjll/anaconda3/lib/python3.8/site-packages/
        return self.callback()
      File "/home/ryjll/anaconda3/lib/python3.8/site-packages/
        self.kernel_manager.restart_kernel(now=True, newports=
      File "/home/ryjll/anaconda3/lib/python3.8/site-packages/
        self.shutdown_kernel(now=now, restart=True)
      File "/home/ryjll/anaconda3/lib/python3.8/site-packages/
        self.interrupt_kernel()
      File "/home/ryjll/anaconda3/lib/python3.8/site-packages/
        raise RuntimeError("Cannot interrupt kernel. No kernel
    RuntimeError: Cannot interrupt kernel. No kernel is runnin
(base) ┌──(ryjll㉿kali)-[~]
└─$ pip uninstall xgboost
Found existing installation: xgboost 1.5.0
Uninstalling xgboost-1.5.0:
  Would remove:
```



```
In [19]: password_clf.score(X_train, y_train)
Out[19]: 0.9877545915614949

In [20]: password_clf.score(X_test, y_test)
Out[20]: 0.9802653684469897

In [59]: common_password = "dog"
         strong_computer_generated_password = "D0gp3rs@n!"
```

HARDWARE TOOLS

- **Laptop**
- **Wifi Adapter**

SOFTWARE TOOLS

- **Pandas Python Library**
- **VMware Workstation Pro**
- **Scikit Python Library**
- **XGBoost**
- **Jupyter Notebook**
- **Python(Installed on Kali OS)**

CONCLUSION

- **In this project we were able to assess the strength of a number of passwords by using the predictions of a trained model.**
- **In one example we were able to discover the strength of a common password "dog" to be 0[weak] compared to an enhanced version of it like "D0gp3rs@n!", where the strength was determined to be 2[strong].**

REFERENCES

- **"TutorialsPoint," [Online]. Available: https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.htm.**
- **S.-k. Developers, "scikit-learn.org," 2007-2021. [Online]. Available: https://scikit-learn.org/stable/tutorial/basic/tutorial.html.**
- **M. Analytics, "mode.com," 2021. [Online]. Available: https://mode.com/python-tutorial/libraries/pandas/.**
- **A. Bronshtein, "towardsdatascience.com," 17 April 2017. [Online]. Available: https://towardsdatascience.com/a-quick-introduction-to-the-pandas-python-library-f1b678f34673.**