National College of Ireland

Configuration Manual

MSc Research Project
Msc in Cybersecurity

Sandeep Thottukadavil Laji
StudentID:x22112936

School of Computing
National College of Ireland

Supervisor:Dr.Imran Khan

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sandeep Thottukadavil Laji
Student ID:	22112936
Programme:	Msc in Cybersecurity
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr.Imran Khan
Submission Due Date:	31/01/2024
Project Title:	Analyzing the Distribution and Interpretation of Neural Network Output for Password Scoring
Word Count:	1000
Page Count:	11

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sandeep Thottukadavil Laji
Date:	31st January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sandeep Thottukadavil Laji

x22112936

Section 1

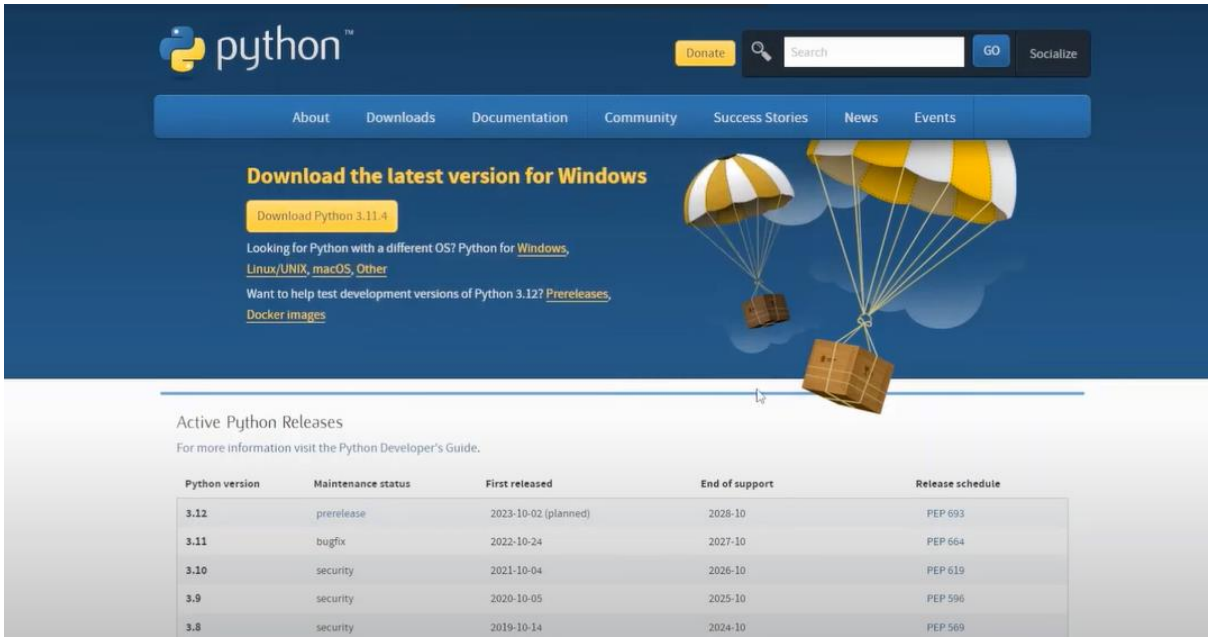


Figure 1: Download Python in Anaconda

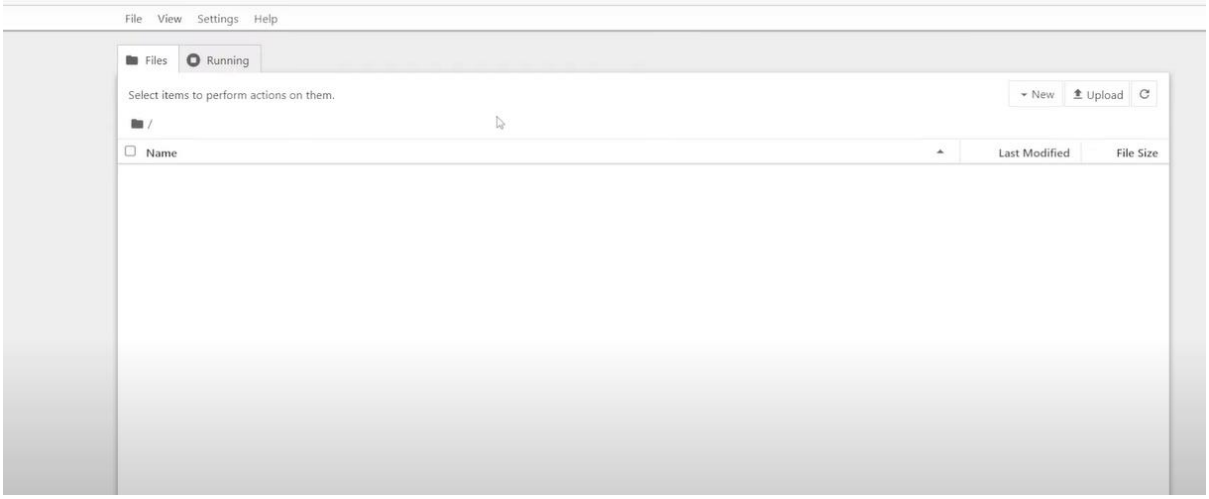


Figure 2: Opening the Python 3

Here, it is required to open the python 3 from anocoda. After opening, it is required to upload the dataset which will be used for the analysis.

Section 2

```
[2]: pip install pandas

Collecting pandas
  Obtaining dependency information for pandas from https://files.pythonhosted.org/packages/11/17/fb1a34f3e73debbc2fd15a01ea17eaab3717943d08463ff4979a4f024b3f/pandas-2.1.4-cp311-cp311-win_amd64.whl.metadata
  Downloading pandas-2.1.4-cp311-cp311-win_amd64.whl.metadata (18 kB)
Collecting numpy<2,>=1.23.2 (from pandas)
  Obtaining dependency information for numpy<2,>=1.23.2 from https://files.pythonhosted.org/packages/da/3c/3ff05c2855eee52588f489a4e607e4a61699a0742aa03ccf641c77f9eb0a/numpy-1.26.2-cp311-cp311-win_amd64.whl.metadata
  Downloading numpy-1.26.2-cp311-cp311-win_amd64.whl.metadata (61 kB)
     ---------------------------------------- 0.0/61.2 kB ? eta -:--:--
     ------ --------------------------------- 10.2/61.2 kB ? eta -:--:--
     ---------------------------------------- 61.2/61.2 kB 1.1 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\asus\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\asus\anaconda3\lib\site-packages (from pandas) (2023.3.post1)
Collecting tzdata>=2022.1 (from pandas)
  Using cached tzdata-2023.3-py2.py3-none-any.whl (341 kB)
Requirement already satisfied: six>=1.5 in c:\users\asus\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Downloading pandas-2.1.4-cp311-cp311-win_amd64.whl (10.6 MB)
   ---------------------------------------- 0.0/10.6 MB ? eta -:--:--
    --------------------------------------- 0.1/10.6 MB 2.8 MB/s eta 0:00:04
   - -------------------------------------- 0.5/10.6 MB 5.8 MB/s eta 0:00:02
   --- ------------------------------------ 1.0/10.6 MB 8.0 MB/s eta 0:00:02
   ------- -------------------------------- 2.0/10.6 MB 13.0 MB/s eta 0:00:01
   ----------- ---------------------------- 3.3/10.6 MB 16.3 MB/s eta 0:00:01
   ----------------- ---------------------- 4.6/10.6 MB 19.8 MB/s eta 0:00:01
   ---------------------- ----------------- 6.0/10.6 MB 21.4 MB/s eta 0:00:01
   ---------------------------- ----------- 7.3/10.6 MB 22.3 MB/s eta 0:00:01
   ---------------------------------- ------- 8.6/10.6 MB 24.0 MB/s eta 0:00:01
   ------------------------------------ --- 9.6/10.6 MB 23.7 MB/s eta 0:00:01
   ---------------------------------------  10.6/10.6 MB 28.5 MB/s eta 0:00:01
   ---------------------------------------- 10.6/10.6 MB 25.2 MB/s eta 0:00:00
```

Fig 3: Pandas Installation

- Installing Pandas Library with the “pip install command”

```
[4]: pip install seaborn

Collecting seaborn
  Obtaining dependency information for seaborn from https://files.pythonhosted.org/packages/7b/e5/83fcd7e9db036c179e0352bfcd20f81d728197a16f883e7b90307a88e65e/seaborn-0.13.0-py3-none-any.whl.metadata
  Downloading seaborn-0.13.0-py3-none-any.whl.metadata (5.3 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\asus\anaconda3\lib\site-packages (from seaborn) (1.26.2)
Requirement already satisfied: pandas>=1.2 in c:\users\asus\anaconda3\lib\site-packages (from seaborn) (2.1.4)
Requirement already satisfied: matplotlib!=3.6.1,>=3.3 in c:\users\asus\anaconda3\lib\site-packages (from seaborn) (3.8.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.3->seaborn) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.3->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.3->seaborn) (4.46.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.3->seaborn) (1.4.5)
Requirement already satisfied: packaging>=20.0 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.3->seaborn) (23.1)
Requirement already satisfied: pillow>=8 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.3->seaborn) (10.0.1)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.3->seaborn) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib!=3.6.1,>=3.3->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\asus\anaconda3\lib\site-packages (from pandas>=1.2->seaborn) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\asus\anaconda3\lib\site-packages (from pandas>=1.2->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\asus\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.3->seaborn) (1.16.0)
Downloading seaborn-0.13.0-py3-none-any.whl (294 kB)
   ---------------------------------------- 0.0/294.6 kB ? eta -:--:--
   - -------------------------------------- 10.2/294.6 kB ? eta -:--:--
   -------- ------------------------------- 61.4/294.6 kB 812.7 kB/s eta 0:00:01
   -------------------- ------------------- 153.6/294.6 kB 1.3 MB/s eta 0:00:01
   ---------------------------------------- 294.6/294.6 kB 1.8 MB/s eta 0:00:00
Installing collected packages: seaborn
Successfully installed seaborn-0.13.0
Note: you may need to restart the kernel to use updated packages.
```

Fig 4 :- Seaborn Installation

- Installing seaborn library by performing the “pip install seaborn” command

```
[7]: pip install scikit-learn

Collecting scikit-learn
  Obtaining dependency information for scikit-learn from https://files.pythonhosted.org/packages/4e/ba/ce9bd1cd4953336a0e213b29cb80bb11816f2a93de8c99f88ef0b446ad0c/scikit_learn-1.3.2-cp311-cp311-win_amd64.whl.metadata
  Downloading scikit_learn-1.3.2-cp311-cp311-win_amd64.whl.metadata (11 kB)
Requirement already satisfied: numpy<2.0,>=1.17.3 in c:\users\asus\anaconda3\lib\site-packages (from scikit-learn) (1.26.2)
Collecting scipy>=1.5.0 (from scikit-learn)
  Obtaining dependency information for scipy>=1.5.0 from https://files.pythonhosted.org/packages/43/d0/f3cd75b62e1b90f48dbf091261b2fc7ceec14a700e308c50f6a69c83d337/scipy-1.11.4-cp311-cp311-win_amd64.whl.metadata
  Downloading scipy-1.11.4-cp311-cp311-win_amd64.whl.metadata (60 kB)
     ---------------------------------------- 0.0/60.4 kB ? eta -:--:--
     ---------------------------------------- 0.0/60.4 kB ? eta -:--:--
     ------ --------------------------------- 10.2/60.4 kB ? eta -:--:--
     ------------------ --------------------- 30.7/60.4 kB 325.1 kB/s eta 0:00:01
     ---------------------------------------- 60.4/60.4 kB 456.0 kB/s eta 0:00:00
Collecting joblib>=1.1.1 (from scikit-learn)
  Obtaining dependency information for joblib>=1.1.1 from https://files.pythonhosted.org/packages/10/40/d551139c85db202f1f384ba8bcf96aca2f329440a844f924c8a0040b6d02/joblib-1.3.2-py3-none-any.whl.metadata
  Downloading joblib-1.3.2-py3-none-any.whl.metadata (5.4 kB)
Collecting threadpoolctl>=2.0.0 (from scikit-learn)
  Obtaining dependency information for threadpoolctl>=2.0.0 from https://files.pythonhosted.org/packages/81/12/fd4dea011af9d69e1cad05c75f3f7202cdcbeac9b712eea58ca779a72865/threadpoolctl-3.2.0-py3-none-any.whl.metadata
  Downloading threadpoolctl-3.2.0-py3-none-any.whl.metadata (10.0 kB)
Downloading scikit_learn-1.3.2-cp311-cp311-win_amd64.whl (9.2 MB)
   ---------------------------------------- 0.0/9.2 MB ? eta -:--:--
    --------------------------------------- 0.1/9.2 MB 3.6 MB/s eta 0:00:03
   - -------------------------------------- 0.3/9.2 MB 3.8 MB/s eta 0:00:03
   -- ------------------------------------- 0.5/9.2 MB 3.9 MB/s eta 0:00:03
   --- ------------------------------------ 0.7/9.2 MB 4.0 MB/s eta 0:00:03
   ---- ----------------------------------- 1.0/9.2 MB 4.4 MB/s eta 0:00:02
   ------- -------------------------------- 1.6/9.2 MB 6.6 MB/s eta 0:00:02
```

Fig 5:scikit-learn

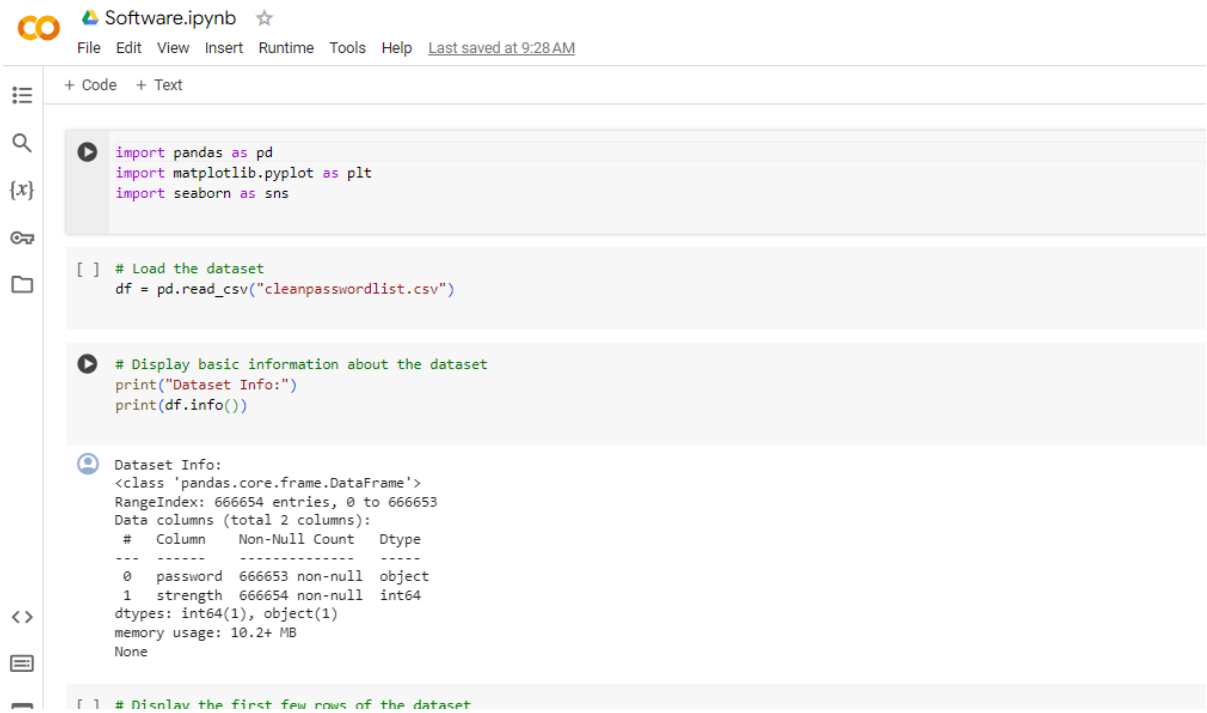
- Installing scikit-learn machine learning library by performing the “pip install scikit learn” command

```
[3]: pip install matplotlib

Collecting matplotlib
  Obtaining dependency information for matplotlib from https://files.pythonhosted.org/packages/26/5a/27fd341e4510257789f19a4b4be8bb90d1113b8f176c3dab562b4f21466e/matplotlib-3.8.2-cp311-cp311-win_amd64.whl.metadata
  Downloading matplotlib-3.8.2-cp311-cp311-win_amd64.whl.metadata (5.9 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Obtaining dependency information for contourpy>=1.0.1 from https://files.pythonhosted.org/packages/ca/2a/d197a412ec474391ee878b1218cf2fe9c6e963903755887fc5654c06636a/contourpy-1.2.0-cp311-cp311-win_amd64.whl.metadata
  Downloading contourpy-1.2.0-cp311-cp311-win_amd64.whl.metadata (5.8 kB)
Collecting cycler>=0.10 (from matplotlib)
  Obtaining dependency information for cycler>=0.10 from https://files.pythonhosted.org/packages/e7/05/c19819d5e3d95294a6f5947fb9b9629efb316b96de511b418c53d245aae6/cycler-0.12.1-py3-none-any.whl.metadata
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Obtaining dependency information for fonttools>=4.22.0 from https://files.pythonhosted.org/packages/c4/d8/a68085e2da45aad6acebf512fe2d9fe783d6af3efad4a3b0d708783f891b/fonttools-4.46.0-cp311-cp311-win_amd64.whl.metadata
  Downloading fonttools-4.46.0-cp311-cp311-win_amd64.whl.metadata (159 kB)
     ---------------------------------------- 0.0/159.4 kB ? eta -:--:--
     --------- -------------------------- 41.0/159.4 kB 991.0 kB/s eta 0:00:01
     ---------------------------------------- 159.4/159.4 kB 1.9 MB/s eta 0:00:00
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Obtaining dependency information for kiwisolver>=1.3.1 from https://files.pythonhosted.org/packages/1e/37/d3c2d4ba2719059a0f12730947bbe1ad5ee8bff89e8c35319dcb2c9ddb4c/kiwisolver-1.4.5-cp311-cp311-win_amd64.whl.metadata
  Downloading kiwisolver-1.4.5-cp311-cp311-win_amd64.whl.metadata (6.5 kB)
Requirement already satisfied: numpy<2,>=1.21 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib) (1.26.2)
Requirement already satisfied: packaging>=20.0 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib) (23.1)
Requirement already satisfied: pillow>=8 in c:\users\asus\anaconda3\lib\site-packages (from matplotlib) (10.0.1)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Obtaining dependency information for pyparsing>=2.3.1 from https://files.pythonhosted.org/packages/39/92/8486ede85fcc088f1b3dba4ce92dd29d126fd96b0008ea213167940a2475/pyparsing-3.1.1-py3-none-any.whl.metadata
```

Fig 6:-Matplotlib installation

- Installing the Matplot library by performing the “pip install matplotlib command”



Figre 7: Code Implmentation

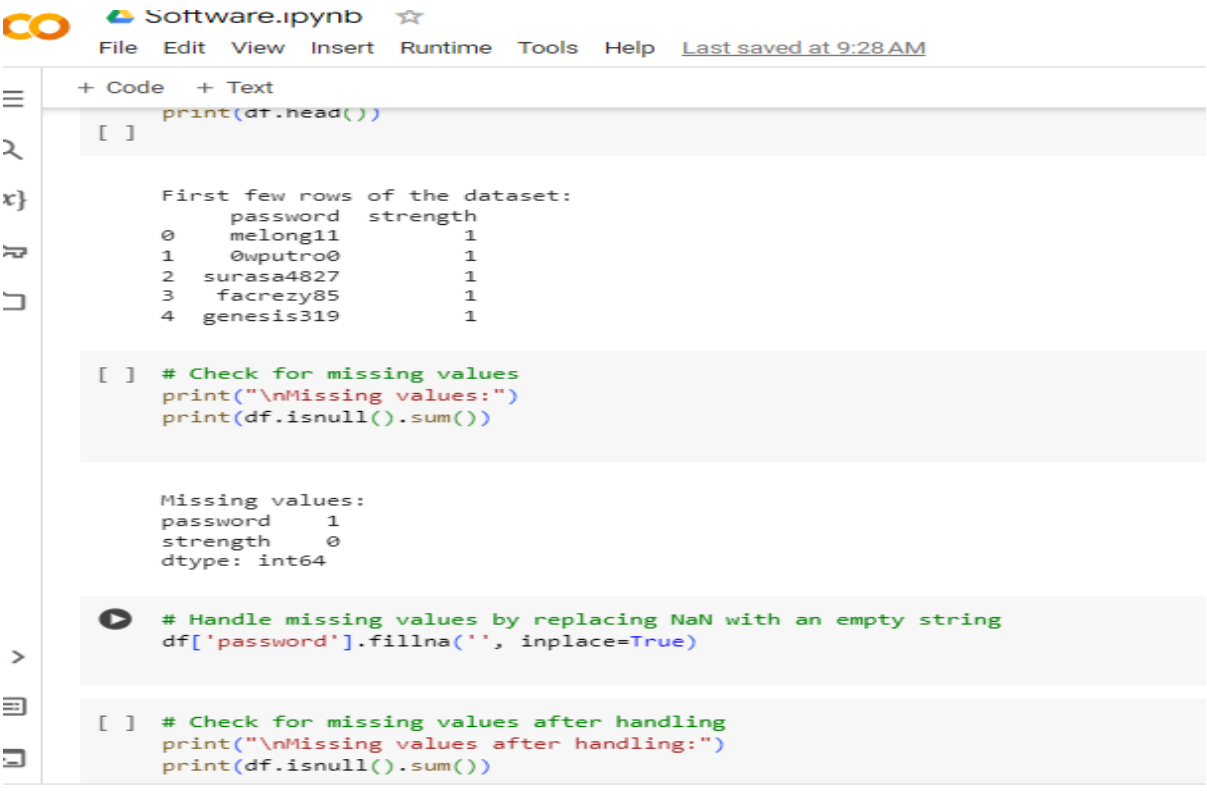


Figure 8: Code for data pre-processing

```
std         0.507934
min         0.000000
25%         1.000000
50%         1.000000
75%         1.000000
max         2.000000
```

Password Strength Distribution

```
[ ]
plt.figure(figsize=(8, 6))
sns.countplot(x='strength', data=df)
plt.title('Password Strength Distribution')
plt.xlabel('Strength')
plt.ylabel('Count')
plt.show()
```

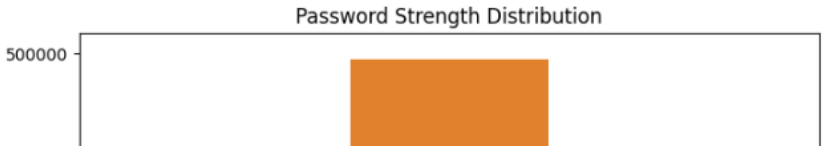


Figure 9: EDA

Section 3

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
```

```
[ ]
X = df[['password']]
y = df['strength']
```

```
[ ]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
[ ]
vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train['password'])
X_test_vectorized = vectorizer.transform(X_test['password'])
```

Figure 10: Model Implementation

They cover a wide range of topics, and these are just a few examples. Data exploration and visualization using Seaborn, data extraction with feature engineering, and predictive analysis with machine learning models like XGBoost and logistic regression are some of the methodologies and tools used. These are only a few of the various methods and tools used. The strategies and instruments mentioned before are only a few of the many that are used. The tools and strategies

shown below are only a few examples. Doing so may help the individual comprehend the inner workings of brain networks. It would do you well to grasp this concept.

These models can derive label predictions from vectorized training data, and the effectiveness of various tactics may be estimated by the accuracy measures of the models. Research that incorporates machine learning techniques may provide a comprehensive understanding of the tasks performed by the neural network's internal nodes. It is possible to do this. In this way, linkages and patterns that may not be immediately apparent from visuals are uncovered.

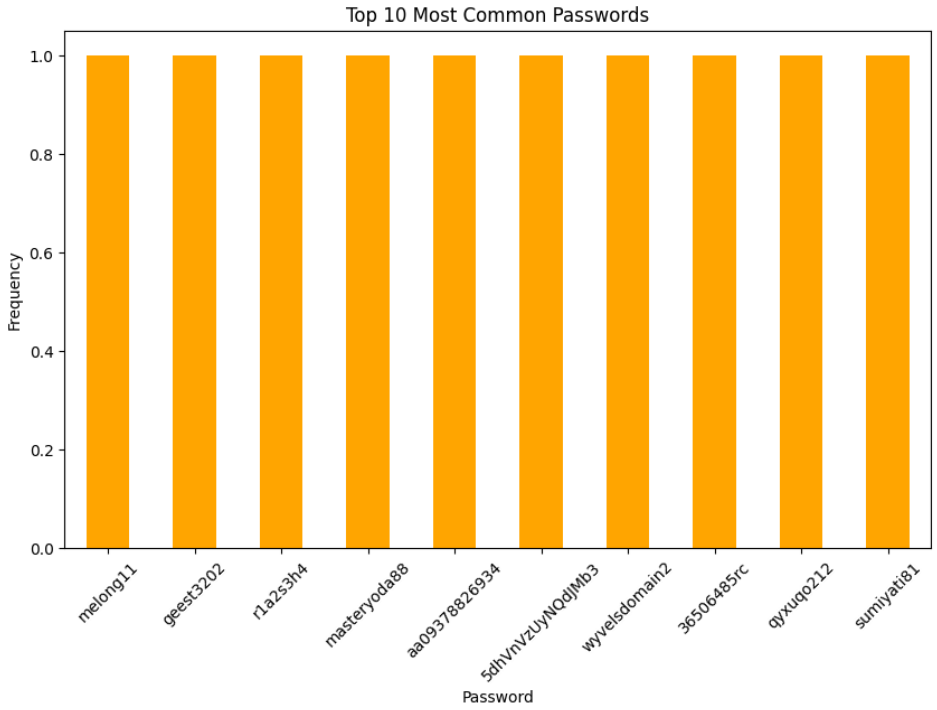


Figure 11: Top 10 Most Common Passwords

In the figure, the top 10 passwords that are found in the dataset the most frequently are shown. The recurrence dissemination of these passwords is shown by means of the utilization of a bar chart, which offers a reasonable and brief cognizance of the passwords that are used the most frequently. What's more, various passwords are shown along the horizontal axis of the chart, while the frequencies that are related with those passwords are portrayed along the upward axis. Along these lines, it is feasible to figure out which passwords are utilized the most frequently.

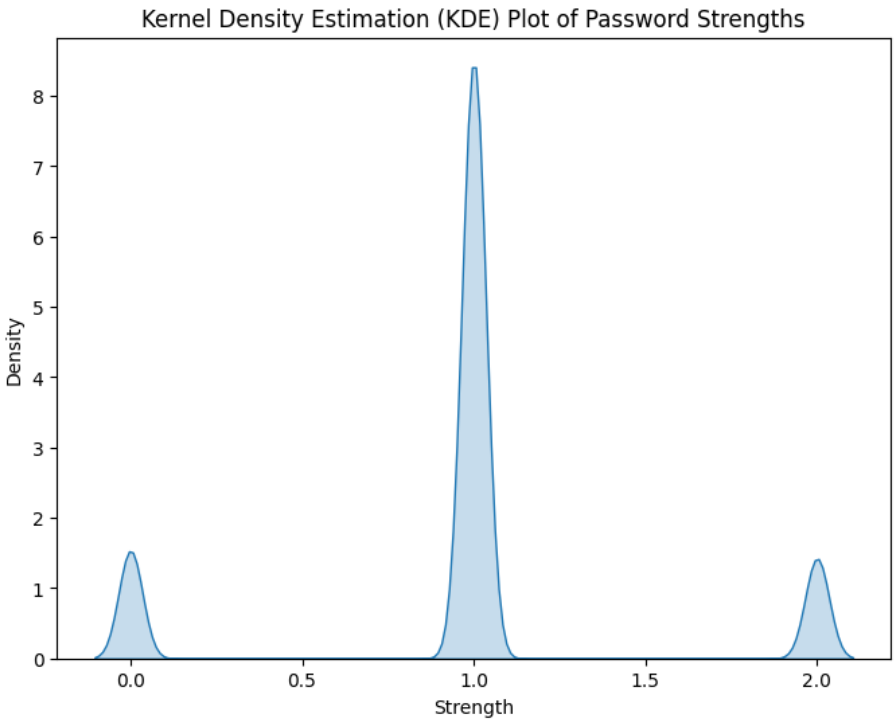


Figure 12: Kernel Density Estimation (KDE) Plot of Password Strengths

The Kernel Density Estimation (KDE) offers a visual portrayal of the circulation of password strengths inside the dataset. This model can be seen as above. It can give experiences on the concentration and fluctuation of password strengths by review the density of various strength levels. This is achieved by means of the utilization of a smooth bend. To acquire a superior understanding of the dispersion example of password strengths, the level hub is utilized to indicate the different strength classes, while the upward hub is used to represent the density of occasions.

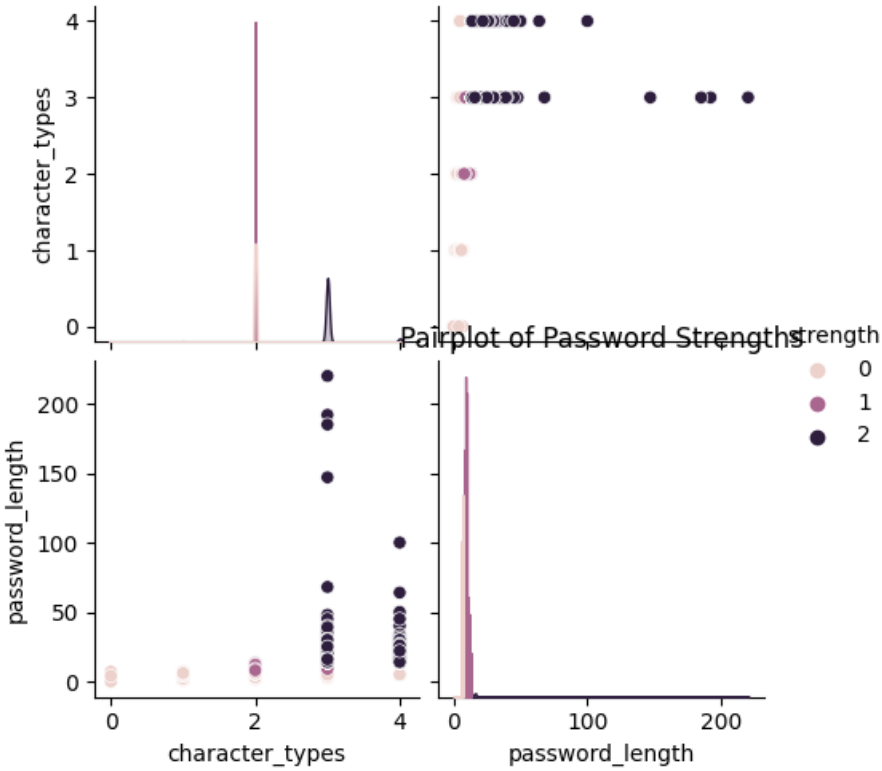


Figure 13: Pairplot of Password Strengths

A complete visual investigation of the correlations between variables is given by the pairplot that is given. The variables are differentiated from each other based on the strength of the password. To illustrate the pairwise joins that exist between features, it makes utilization of a large number of varieties to indicate varying levels of power. All in all, this graphical representation assists in tracking down possible correlations, patterns, and distributions among a large number of variables, which thusly advances a more mind boggling understanding of the interrelationships that exist between the variables that are incorporated inside the dataset.

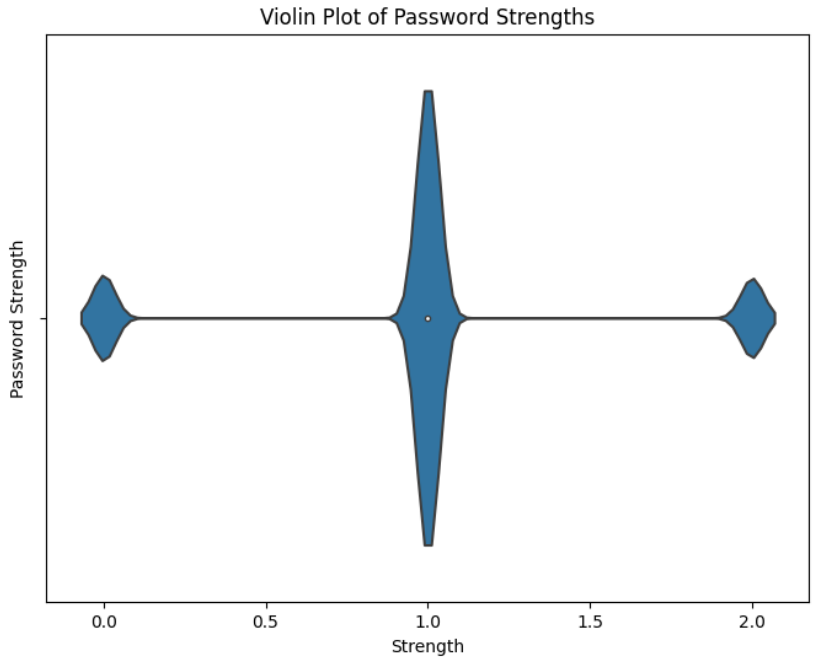


Figure 14: Violin Plot of Password Strengths

Violin plot illustrates the distribution of secret word strengths that are sorted by their different levels is displayed as a violin plot. By utilizing its width and structure, portraying the likelihood thickness of various different strength categories is conceivable. This gives a reasonable understanding of the distribution and concentration of these categories. With the guide of this graphical portrayal, one might have the option to get a more complete cognizance of the varieties and key examples that exist among the secret phrase strengths included inside the dataset.

References

Jung, D.H., Kim, H.S., Jhin, C., Kim, H.J. and Park, S.H., 2020. Time-serial analysis of deep neural network models for prediction of climatic conditions inside a greenhouse. Computers and Electronics in Agriculture, 173, p.105402.