Password Project

Kimberly Kokado

In [1]:
```
import numpy as np
import pandas as pd
import os
import re
```

First we need to import the dataframe:

In [2]:
```
df = pd.read_csv(r"C:\Users\juggl\OneDrive\Documents\rockyou.txt", delimiter= "\n",
                 header=None, names = ["Passwords"], encoding = "ISO-8859-1")
```

In [3]:
```
df.head() #gives the header
```

Out[3]:

	Passwords
0	123456
1	12345
2	123456789
3	password
4	iloveyou

In [4]:
```
#drop duplicates from the dataframe
df.drop_duplicates(subset=["Passwords"], keep=False, inplace =True)
```

In [5]:
```
df.info() #gives us info on the dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 14343279 entries, 0 to 14343475
Data columns (total 1 columns):
 #   Column     Dtype
---  ------     -----
 0   Passwords  object
dtypes: object(1)
memory usage: 218.9+ MB
```

In [6]:
```
#reset index after dropping duplicates
df = df.reset_index(drop=True)
print(df)
```

```
             Passwords
0               123456
1                12345
2            123456789
3             password
4             iloveyou
...                ...
14343274       xCvBnM,
14343275         ie168
14343276     abygurl69
14343277        a6_123
14343278  *7Â¡Vamos!
```

```
[14343279 rows x 1 columns]
```

In [7]:
```python
#function to check password
def password_check(password):

    # calculating the length
    length_error = len(password) < 8

    # searching for digits
    digit_error = re.search(r"\d", password) is None

    # searching for uppercase
    uppercase_error = re.search(r"[A-Z]", password) is None

    # searching for lowercase
    lowercase_error = re.search(r"[a-z]", password) is None

    # overall result
    password_ok = not ( length_error or digit_error or uppercase_error or lowercase_err

    if password_ok ==True:
        return "Password ok"
    if length_error == True:
        return "Length error"
    if uppercase_error == True:
        return "Uppercase error"
    if lowercase_error == True:
        return "Lowercase error"
    if digit_error == True:
        return "Digit error"
```

In [8]:
```python
def remove_specialchar(password):

    #search for special characters
    symbol_error = re.search(r"\W", password) is not None

    #search for spaces
    space_error = re.search(r" ", password) is not None

    if symbol_error == True:
        return "Symbol included"
    if space_error == True:
        return "Space included"
```

first half of the above function is from stackoverflow:
https://stackoverflow.com/questions/16709638/checking-the-strength-of-a-password-how-to-check-conditions

In [9]:
```python
df['Error'] = df['Passwords'].apply(remove_specialchar)
```

In [12]:
```python
print(df)
```

```
          Passwords      Error
0            123456       None
1             12345       None
2         123456789       None
3          password       None
4          iloveyou       None
...             ...        ...
```

```
14343274       ⍰xCvBnM,  Symbol included
14343275         ⍰ie168  Symbol included
14343276     ⍰abygurl69  Symbol included
14343277        ⍰a6_123  Symbol included
14343278  ⍰*⍰7Â¡Vamos!⍰  Symbol included

[14343279 rows x 2 columns]
```

In [13]:
```
#take the symbol and spaces out of the dataframe
clutter = df[(df['Error']=="Symbol included") | (df['Error'] == "Space included")].inde
df.drop(clutter, inplace=True)
```

In [14]:
```
df = df.reset_index(drop=True) #reset index after dropping clutter
print(df)
```

```
            Passwords Error
0              123456  None
1               12345  None
2           123456789  None
3            password  None
4            iloveyou  None
...               ...   ...
13481551  05438515170  None
13481552  themysthery  None
13481553         qq76  None
13481554     fuckyou_  None
13481555     ettena22  None

[13481556 rows x 2 columns]
```

In [15]:
```
#input error into a new column for password
df["Error"] = df['Passwords'].apply(password_check)
```

In [16]:
```
df.head()
```

Out[16]:

	Passwords	Error
0	123456	Length error
1	12345	Length error
2	123456789	Uppercase error
3	password	Uppercase error
4	iloveyou	Uppercase error

In [17]:
```
nist_pwd = df[ df['Error']== "Password ok"].index
print (nist_pwd)
```

```
Int64Index([    3475,     7456,     8114,     9828,    12130,    14128,
               16287,    18026,    18466,    21300,
            ...
            13479982, 13480099, 13480108, 13480754, 13481006, 13481007,
            13481336, 13481517, 13481526, 13481542],
           dtype='int64', length=301312)
```

In [18]:
```
print(len(nist_pwd))
```

```
301312
```

This is the number of passwords in the 'rockyou.txt' that pass the partial NIST parameters for

passwords.

In [ ]: