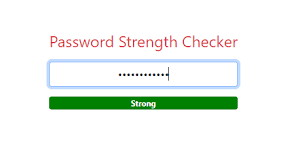
**Password Strength Checker**

Input : password (combination of letters, numbers, characters or either of them)

Output : It will predict whether given password is strong or weak or medium



**Abstract:**

A password strength checker works by understanding the combination of digits, letters, and special symbols you use in your password. It is created by training a machine learning model on a labelled dataset of different combinations of letters and special symbols people use in passwords. The model learns from data about what combinations of letters and symbols can be classified as a solid or weak password.

**Dataset:**

To create an application to check the strength of passwords, we need to have a labelled dataset about different combinations of letters and symbols. I found a dataset on Kaggle to train a machine learning model to predict the strength of a password.

Link: [**dataset**](https://www.kaggle.com/datasets/bhavikbb/password-strength-classifier-dataset)

**Required Libraries:**

1. Pandas
2. Numpy
3. Sklearn

**1 .Pandas :**

* + - * It is a python library.
      * It s a high level manipulation data
      * and used for analysing the data.

**2 .Numpy:**

* + - * + Numpy stands for numerical python.
        + It working with the arrays

**3.Sklearn:**

* Scikit-learn is a free machine learning library for Python.
* It features various algorithms like support vector machine, random forests, and k-neighbours, DecisionTreeclassifers.

**Algorithm Used:**

*DecisionTreeClassifier:*

A decision tree is a type of supervised machine learning used to categorize or make predictions based on how a previous set of questionswereanswered.

**About the Dataset:**

The dataset contains two columns **: password and strength**.

In strength Column:

* 0 : It specifies the password strength is WEAK
* 1 : It specifies the password strength is MEDIUM
* 2 : It specifies the password strength is STRONG

Program:-

import pandas as pd

import numpy as np

data=pd.read\_csv(r"C:\Users\'\Downloads\data.csv",on\_bad\_lines="skip")

print(data.head())

Output:

password strength

0 kzde5577 1

1 kino3434 1

2 visi7k1yr 1

3 megzy123 1

4 lamborghin1 1

**EXPLANATION:**

I will convert 0, 1, and 2 values in the strength column to weak, medium, and strong.

data= data.dropna()

data["strength"]=data["strength"].map({0:"Weak",

1:"Medium",

2:"Strong"})

print(data.head())

Output:

password strength

0 kzde5577 Medium

1 kino3434 Medium

2 visi7k1yr Medium

3 megzy123 Medium

4 lamborghin1 Medium

**EXPLANATION:**

To train a machine learning model to predict the strength of the

password. Before we start preparing the model, we need to [tokenize](https://thecleverprogrammer.com/2022/01/04/sentence-and-word-tokenization-using-python/)

 the passwords as we need the model to learn from the combinations

of digits, letters, and symbols to predict the password’s strength. So

here’s how we can tokenize and split the data into training and test

sets :

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

def word(password):

character=[]

for i in password:

character.append(i)

return character

x=np.array(data["password"])

y=np.array(data["strength"])

tdif=TfidfVectorizer(tokenizer=word)

x=tdif.fit\_transform(x)

xtrain,xtest,ytrain,ytest=train\_test\_split(x,y,test\_size=0.2,random\_state=42)

**EXPLANATION:**

To train a classification model to predict the strength of the password:

from sklearn.tree import DecisionTreeClassifier

model= DecisionTreeClassifier()

model.fit(xtrain, ytrain)

print(model.score(xtest, ytest))

Output:

0.9246510562889513

**EXPLANATION:**

 We can check the strength of a password using the trained model:

import getpass

user=getpass.getpass("Enter Password: ")

data=tdif.transform([user]).toarray()

output=model.predict(data)

print(output)

Output:

Enter Password: ········

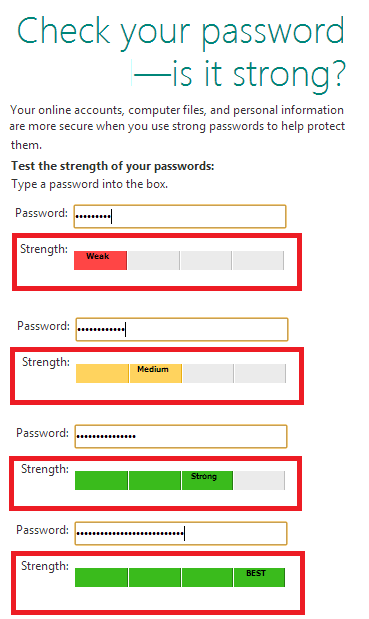
['Weak']

Enter Password: ········

['Medium']

Enter Password: ········

['Strong']



**Conclusion:**

So this is how you can use machine learning to create a password’s strength checker using the Python programming language. A password strength checker works by understanding the combination of digits, letters, and special symbols you use in your password.

Name: S.Sandhyakumari

ID: S180666

Class: CSE-2C