Checking Password Strength Using Machine Learning and Artificial Intelligence

A
Project Report
Submitted in the partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering
by

Aman Kumar Singh (1805210007)
Pushpa Devi (1805210040)
Kishan Rana (1805210025)

Under the supervision of
Ms. Pratibha Pandey
Prof. D.S. Yadav

Department of Computer Science and Engineering
Institute of Engineering and Technology, Lucknow
Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh

CONTENTS

Declaration

We hereby declare that this submission is our own work and that, to the best of our belief and knowledge, it contains no material previously published or written by another person or material which to a substantial error has been accepted for the award of any degree or diploma of university or other institute of higher learning, except where the acknowledgement has been made in the text. The project has not been submitted by us at any other institute for the requirement of any other degree.

Submitted by: - Date: 06/06/2022

[1] Name: Aman Kumar Singh
Roll No.: 1805210007
Branch: Computer Science And Engineering
Signature:

[2] Name: Pushpa Devi
Roll No.: 1805210040
Branch: Computer Science And Engineering
Signature:

[3] Name: Kishan Rana
Roll No.: 1805210025
Branch: Computer Science And Engineering
Signature:

Certificate

This is to certify that the project report entitled **Checking Password Strength Using Machine Learning and Artificial Intelligence** presented by **Aman Kumar Singh** and **Kishan Rana** and **Pushpa Devi** in the partial fulfillment for the award of **Bachelor of Technology** in **Computer Science and Engineering** is a record of work carried out by them under my supervision and guidance at the Department of **Computer Science and Engineering** at **Institute of Engineering and Technology, Lucknow**.

It is also certified that this project has not been submitted at any other Institute for the award of any other degrees to the best of my knowledge.

Prof. D.S. Yadav
Ms. Pratibha Pandey
Department of Computer Science and Engineering
Institute of Engineering and Technology, Lucknow

Acknowledgement

We would like to express our sincere regards and appreciation to all the individuals who gave us the opportunity to complete our project. First, we wish to express our sincere gratitude to Head, CSE Department, IET Lucknow, and our supervisor **Prof. Diwakar Singh Yadav.** and **Ms. Pratibha Pandey** for their insightful comments, beneficial information, and realistic recommendation advice, which have helped us at all times in our research work and the making of this project. If it were not for their help and supervision, we would not have been able to complete this project.

We'd like to thank all of our friends who encouraged us and assisted us at every stage of the project's completion.

We'd also want to extend our heartfelt gratitude to all of the authors of the references and other literary works cited in this effort.

.

Name: Aman Kumar Singh
Roll No.: 1805210007
Signature:

Name: Pushpa Devi
Roll No.: 1805210040
Signature:

Name: Kishan Rana
Roll No.: 1805210025
Signature:

Abstract

Passwords are a lively component of method security. The password is the more destructively authenticating the correspondence in many requests. Password strength meters present a natural and immediate ability to be seen with eye response for that reason constitutes a forceful identification. When background a new account or changeful passwords, a limited beat indicates by virtue of how forceful a projected password is thought-out expected.

Life nowadays has emerged as in large part depending on passwords. A regular pc consumer might also additionally require passwords for plenty functions inclusive of logging in to pc accounts, retrieving email from servers, moving funds, buying online, having access to programs, databases, networks, internet sites, or even analyzing the morning newspaper online.

As the internet is growing the cyber crime world is also becoming more advanced and the number of security loopholes are increasing day by day. As the number of cyber attacks are increasing , choosing a good strength password has become an essential action for ourselves.

Our goal is to create a machine learning model which can predict the strength of passwords so that weak passwords can be detected and avoided to secure our social and personal information on the internet.

List of Figures

List of Tables

Chapter - 1
Introduction

Life nowadays has emerged as in large part depending on passwords. A regular pc consumer might also additionally require passwords for plenty functions inclusive of logging in to pc accounts, retrieving email from servers, moving funds, buying online, having access to programs, databases, networks, internet sites, or even analyzing the morning newspaper online.

So the passwords are the walls behind which we are keeping our valuable personal information on the internet. We all know that slowly everything around us is going to be digitized so we are going to have a lot of passwords for preserving various personal actions and information.

Sometimes our personal information is not meant to be known by any other person on the internet because having our valuable or critical information in someone else's hand can result in some really big problems.

As the internet is growing the cyber crime world is also becoming more advanced and the number of security loopholes are increasing day by day. As the number of cyber attacks are increasing , choosing a good strength password has become an essential action for ourselves.

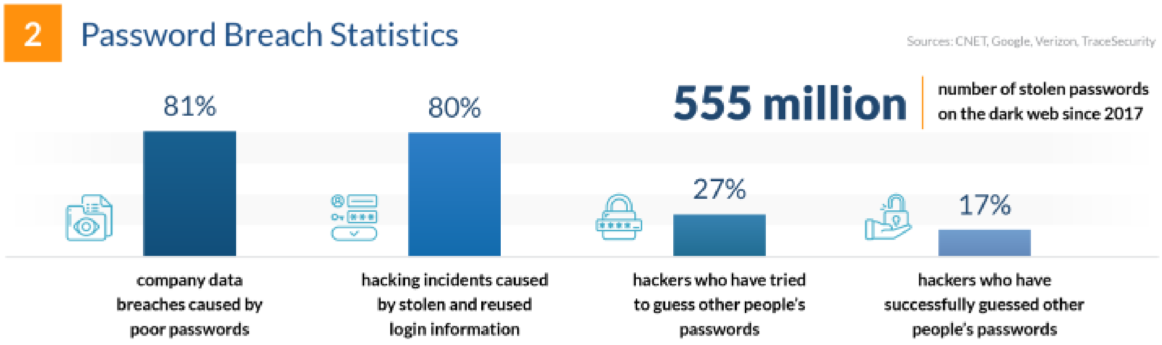


Fig. 1 - Password Breach Data Statics since 2017

The hassle of choosing and the use of true passwords is turning into extra critical each day. The wide variety and the significance of offerings which are supplied thru computer systems and networks grow dramatically and in lots of instances such offerings require passwords or different styles of consumer identification.

For exclusive reasons, together with apparent safety concerns, customers ought to use exclusive passwords for exclusive structures or offerings, making it extra hard to not forget and shield one's password. Finally, passwords are wanted for defensive mystery records that cannot

Be remembered through the consumer (e.g. non-public keys) in authentication and encryption software programs this is turning into crucial to many applications.

safety. Many safety experts propose password control software programs because of the pleasant manner to create and keep robust passwords.

1.1 Motivation

Cyber security and machine learning are one of the top trending tech stacks in today's world. Almost everything which is connected to the internet somehow connects with these two domains of computer science.

Whether we talk about our you tube , social media feed or doing online transactions everyday through Paytm, Gpay etc. we are somehow using features of cyber security and machine learning. These two technologies are core of this project so having a good understanding of these two will help us to understand and operate the software industry in a good manner.

1.2 Objective

Our objective is to create a machine learning model which can predict the strength of passwords with high accuracy.

And we are also opposed to getting a good introduction of cyber security and machine learning so that we can apply these learnings in our real life software engineering use cases.

Chapter-2
Literature Review

Egelman, S., Sotirakopoulos, A., Muslumov, I., Beznosov, K., & Herley, C described [11] : That consumer-preferred passwords constitute bound patterns has existed well recorded. Morris and Thompson erect that an oversized a part of passwords on a operating system whole were for sure guessable. 3 decades later, Florˆencio and Herley raise that netting users be disposed the feeblest passwords admitted. Many current leaks of huge identification datasets have indicated that normal alternatives, like “123456,” are terribly normal. whereas abundant toil is going on dedicated to bright customers to pick forceful passwords, the concept of word substance remains remarkably difficult to outline. The self-generated measures, within the means that technologist deterioration or guessing deterioration need awareness of the statistical distribution of passwords. Early works to live word substance paralleled the measures of cryptographical substance: Associate in Nursing identification of your time N, drawn from part of capability C, would have substance N log2 C bits. NIST directions gift a distinction regarding this approach, place strength may be a perform solely of your time and temperament arrangement.Weir et al.. showed that neither of those measures offers an honest guide to the opposition of a guessing attack , a finding storied by chaise et al.. word Cracking tools, to a degree John The murderer, bound discussion-lists and attain success so much in addition what the NIST deterioration calls. Some passwords that perform sturdy below the first deterioration measures fall nearly quickly to breaking forms. Probabilistic context free grammars are a unit inclined to vanquish even high-quality usage primarily based on results. whereas the concept of substance grant permission be fuzzy, it performs clear that a perfect strength of a identification hopeful Associate in Nursing growing perform of the difficulty it presents to trendy breaking forms.We calm the subsequent traits of members recent and new passwords to examine in what means or manner passwords changed established meter response :

- Length
- Levenshtein rewrite distance
- Number of lowercase messages
- Number of capitalization replies
- Number of digits
- Number of symbols

We acted not to check the new passwords of colleagues in the control condition cause their strength did not considerably change. Likewise, the habits at which point substance increased 'tween the EM and PPM environments acted not observably disagree. Thus, we merged two together exploratory environments and acted a Wilcoxon Signed Ranks test to equate the characteristics filed in Table two, between shareholders’ premature and adjusted passwords.We used the “Holm-Sidak" adjustment and lift that incidental to meters, passwords changed in three statistically important habits. First, time raised from a middle of nine to ten characters . Second, use of “distinctive” characters raised from nothing to seven participants. Third, little replies raised from a middle of six to seven letters individual-tailed). Thus, the meters stirred parties to style a lot of prolonged passwords through the inclusion of characters and supplementary little notes.Password meters could also be classified into two types: not connected to pc or network meters and wired meters. If not connected to pc or network meters, the live program is downloaded to the client and runs domestically. Thus, the campaigner passwords don't seem to be unprotected to some system that supports the concealment of the contestant passwords. On the other hand, it provides a flash for attackers to simply acquire the language secondhand from each one meter and to resolve by suggesting what passwords are calculated. It will influence the safety of forms. Maintaining current meters may be a lot of hard task in not connected to pc or network metering, as customers have to be compelled to check whether or not the beat is current on any occasion they have to update their passwords. On-line meters are sleek to uphold as a result of the most elements of meters are within the server. The gloss secondhand needn't be created public, that provides less probability to the attackers to realize the words within the gloss. Thus, if adequate computing capability is supported within the attendant, wired live appearance is favored.

Giancarlo Ruffo and Francesco Bergadano from Dipartimento di Informatica described that [1] :

Password Filters and Proactive Checking - Spafford plans that choked with enthusiasm examining will impose upon Bloom filtersDavies e Ganesan adopt a Markovian model in Bapasswd , and Nagle suggests a comprehensible, however persuasive take a look at established a linguistics analyzer in depoxius. With ProCheck,full of enthusiasm, examining is lowered to a Pattern Recognition drawback, place the task draw and confirm the foundations to categorize passwords pretty much as good or distressing. These rules is also given by method of call seedlings, buxom by classic initiation algorithms ID3-like; extremely, ProCheck uses helimnosn to make a resolution timber from a “crack” lexicon (upper category of models of distressing passwords), and from a carelessly manufacture file of “good” passwords.These approaches square measure distinguished on the action of the condensation rate of the given language and moment of truth captured for one classifiers to settle on if a identification is nice or distressing. Of course, another main parameter is probably going from each one categorization mistake portion (the total of the speed of faux negatives and wrong still image gaga a camera).As explicit within the following divisions, Enfilter uses a “Dictionary Filter” placed on conclusion wood categorization. The implementation of the aforesaid percolate is associate degree labile reasoning of ProCheck. despite the fact that ProCheck continues to be allotted to supply because the final adept answer w.r.t. classification periods and scope necessary for concealment of the compacted language, education part is refined so more corrects the condensation rate of the by-product resolution timber.

Chapter-3
Methodology

3.1 Theory of Algorithm

The password strength analyser is intended with numerous filters to reason the passwords that square measure unremarkably chosen by the user, by considering human behavior and tendency to pick passwords that square measure straightforward, short and straightforward to recollect.

Our model works on filtering logic. We'll apply multiple filters which are able to extract some properties from our model to coach.
Filters square measure nothing, however some logical conditions that we have a tendency to place earlier than knowledge and once knowledge passes through it then few potential square measure generated support the results.

We will be having a lot of data which we collected from various resources. Also the data we collected from various resources was not properly formatted so we did some formatting over our data. Our model will take input in the format of key value pairs where key will be password and value can be any value among 0, 1, 2 here 0 means weak password , 1 means medium password and 2 means strong password.

Since our model is based on supervised learning our model will classify these passwords in three categories and will extract patterns on the basis of category of password and when we will entered any test password our model will tell us on the basis of patterns our model extracted from previous training data that the password is weak, medium or strong.

The filters we are applying here are basically features selected from few research papers and our own analysis and these features are aligned in a particular order so that during decision making our model should not make bad decision ex. we should not put feature of two length password as strong because if it happens our whole model will train it self in bad manner and for every test data we will be getting some results which will be pruned to errors.

In this model explanation we are basically using three filters :

1. **Filter-1** - Empty passwords
2. **Filter-2** - Commonly used passwords
3. **Filter-3** - Dictionary words

Filter 1: Filter 1 check is basically the simplest check where the check is if the password entered by the user is empty or not , if empty it will be simply a weak password.

Filter 2: Now after successful completion of the first filter here Filter no 2 verifies the identification in the list of most usually secondhand passwords. A list of various conversations is asserted and are secondhand for proof. The words are composed from web pages. If the likely identification counterparts accompany the list, it is top-secret as a very feeble identification and appropriate alerts will open or fan out. Figure Shows the foundation of the full enthusiasm identification substance analyst.

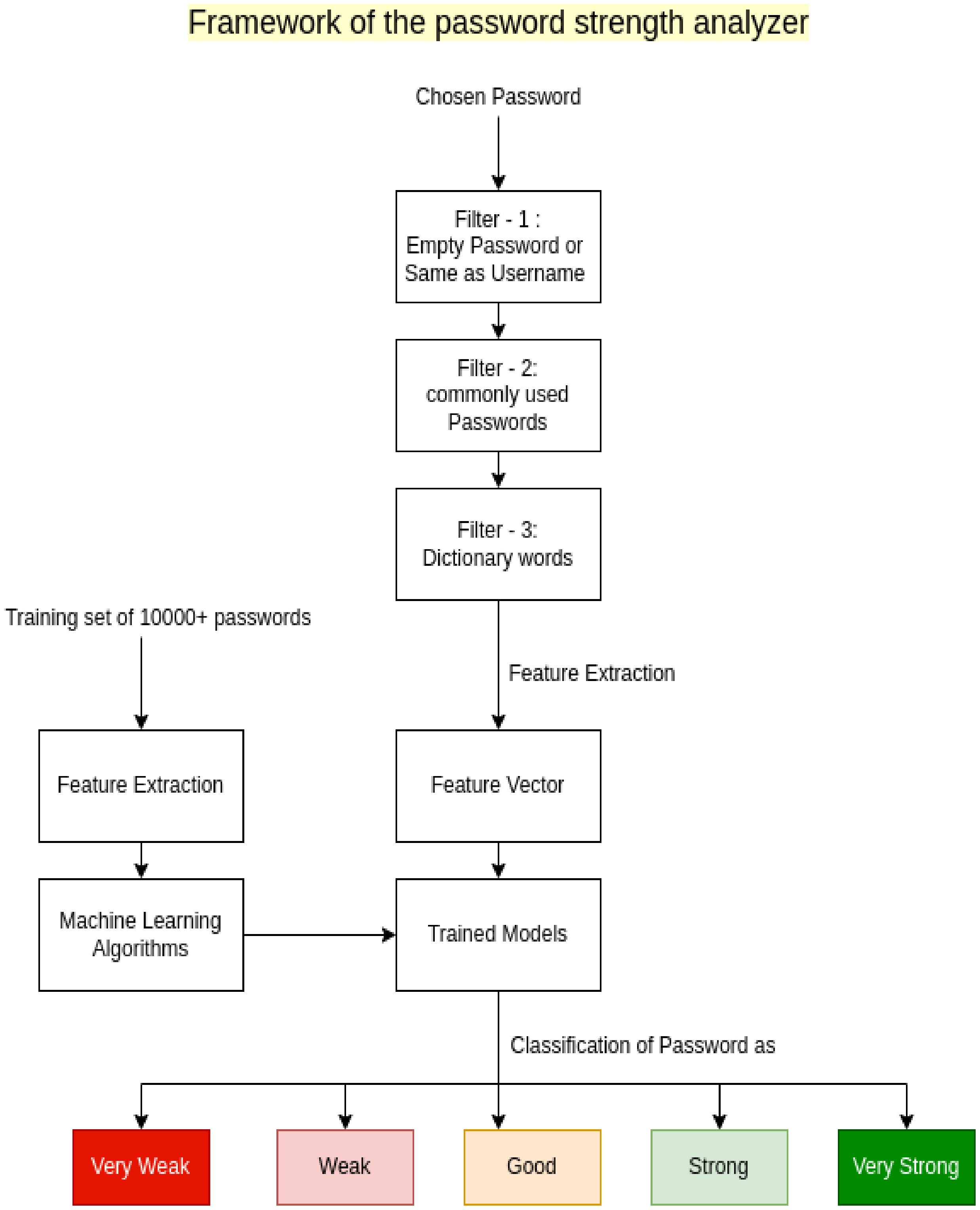

Framework of the password strength analyzer
Chosen Password
Filter - 1 : Empty Password or Same as Username
Filter - 2: commonly used Passwords
Filter - 3: Dictionary words
Training set of 10000+ passwords
Feature Extraction
Feature Extraction
Feature Vector
Machine Learning Algorithms
Trained Models
Classification of Password as
Very Weak
Weak
Good
Strong
Very Strong

Fig. 3.1.1 - Filtering logic Diagram of Algorithm

Filter 3: After filter no 1 check and filter no 2 check here filter no three is to envision the exposure of a feeble word that's applicable for a language attack, associate upper crust of 40k language words are composed. The word intensity grazing from 5 to eight varieties is organized on the ordering of alphabets. The words uphold indifferent files. The identification most well-liked each user is inspected for attracting substance against a group of language speech communication. If the word could be a word from vocabulary,, it's thought-out as a feeble identification and a few custom warnings are going to be shown to the shopper. Once hunting all of the filters we tend to apply, the doubtless identification is analyzed and attractive options are culled and proved against the Support Vector Machine classifier that's erected earlier by a coaching set of additionally to ten thousand, eight figure time passwords. Those models classify the identification chosen for one client as terribly feeble, feeble, good, forceful, and powerful in accordance with attractive substance.

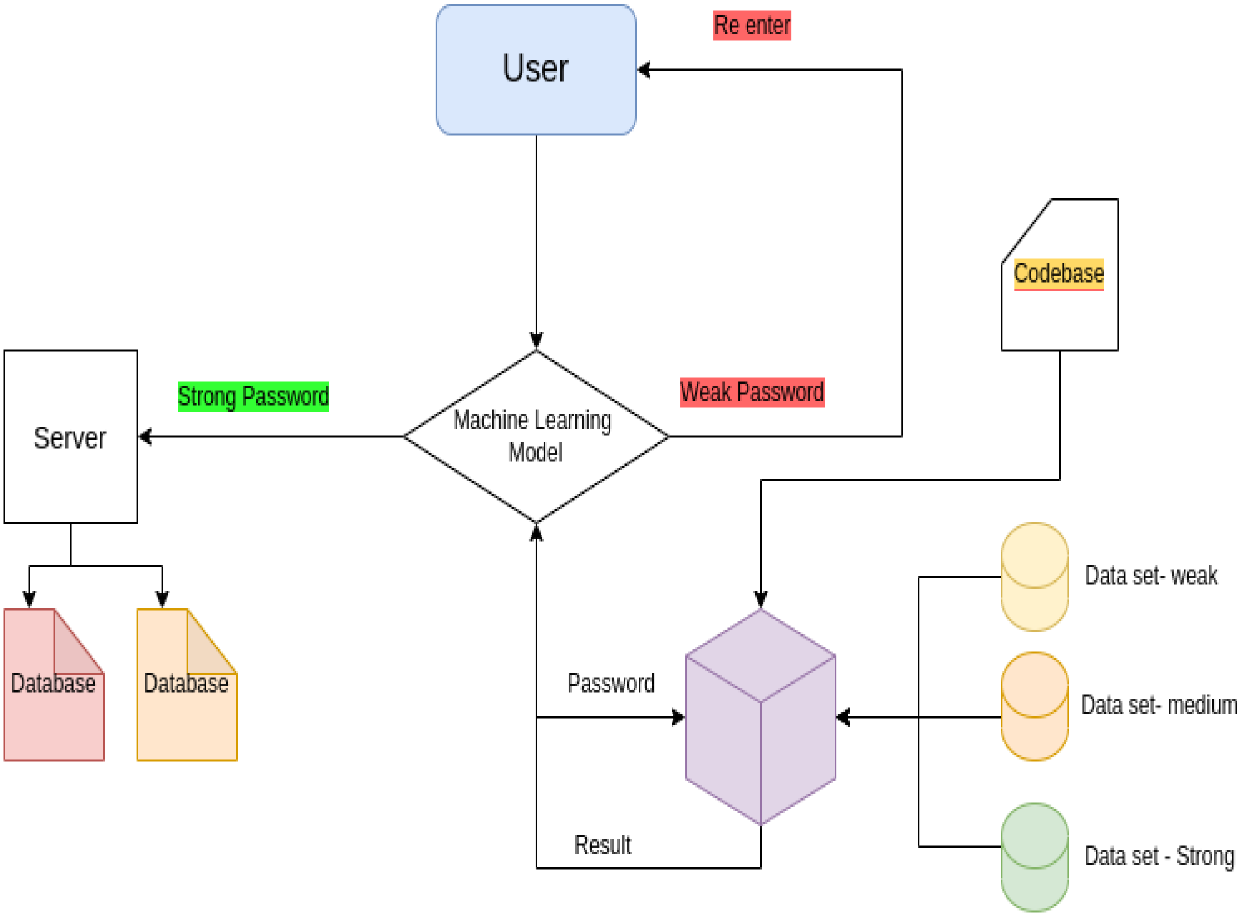


Fig. 3.1.2 - End to end user, system interaction diagram

3.2 Dataset

In a dataset, a coaching set is enforced to make up a model, whereas a check (or validation) set is to validate the model engineered. Knowledge points within the coaching set are excluded from the check (validation) set. Usually, a dataset is split into a coaching set, a validation set (some individuals use ‘test set’ instead) in every iteration, or divided into a coaching set, a validation set and a check set in every iteration.

In Machine Learning, we tend to essentially try and produce a model to predict the check knowledge. So, we tend to use the coaching knowledge to suit the model and testing knowledge to check it. The models generated are to predict the results unknown that are called because of the check set. As you noticed, the dataset is split into a train and check set so as to envision accuracies, precisions by coaching and testing it thereon.

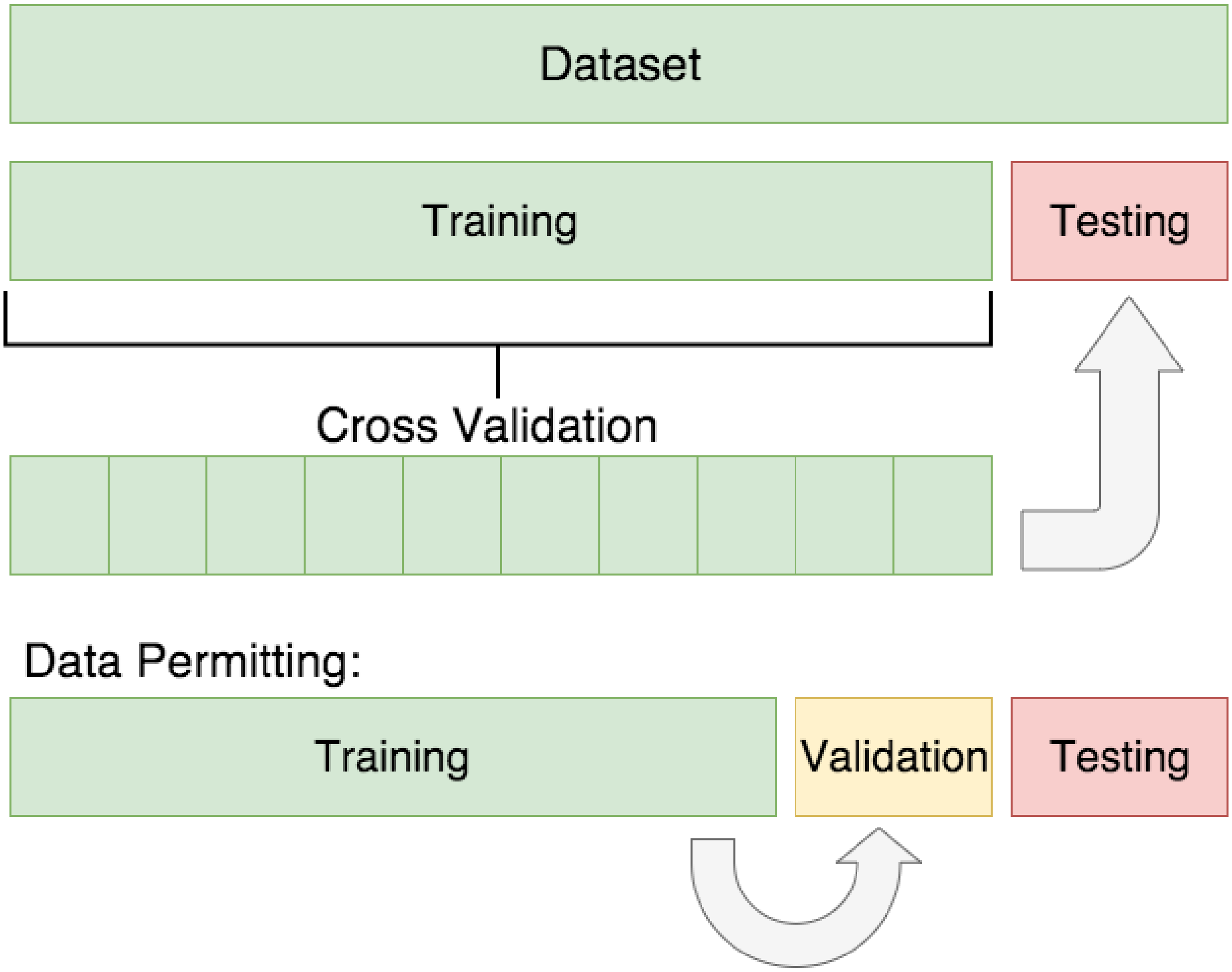


Fig. 3.2.1 - Dataset types diagram

3.2.1 Data Collection

Data collection is one of the most important and initial steps in data preparation for training any machine learning model.

We have used multiple data resources to gather the scale of data which is required to train our model to have a good efficiency on predictions.

The most common data sources we used to collect data :

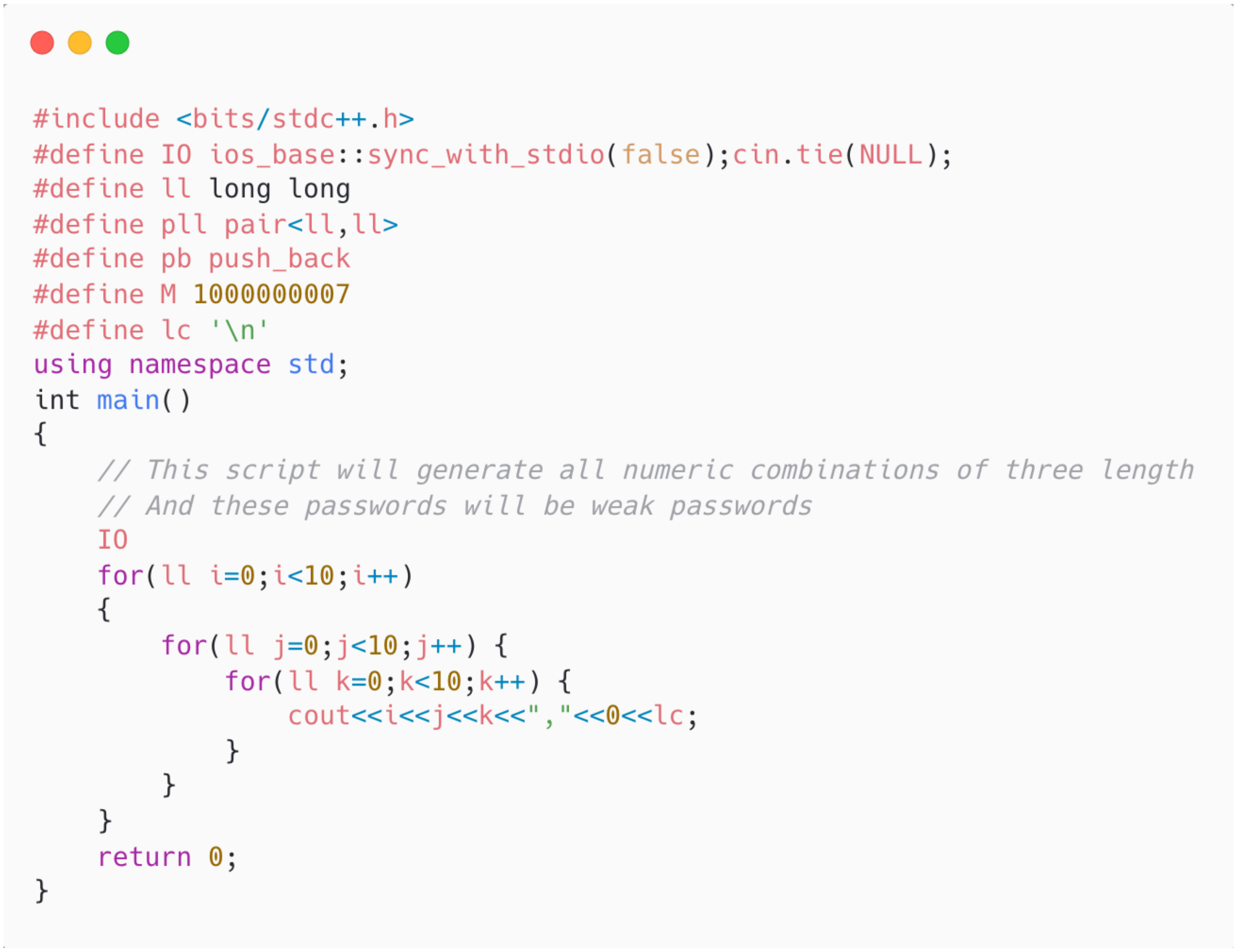
Manual Data Generation

Open Source Datasets

Web Crawling

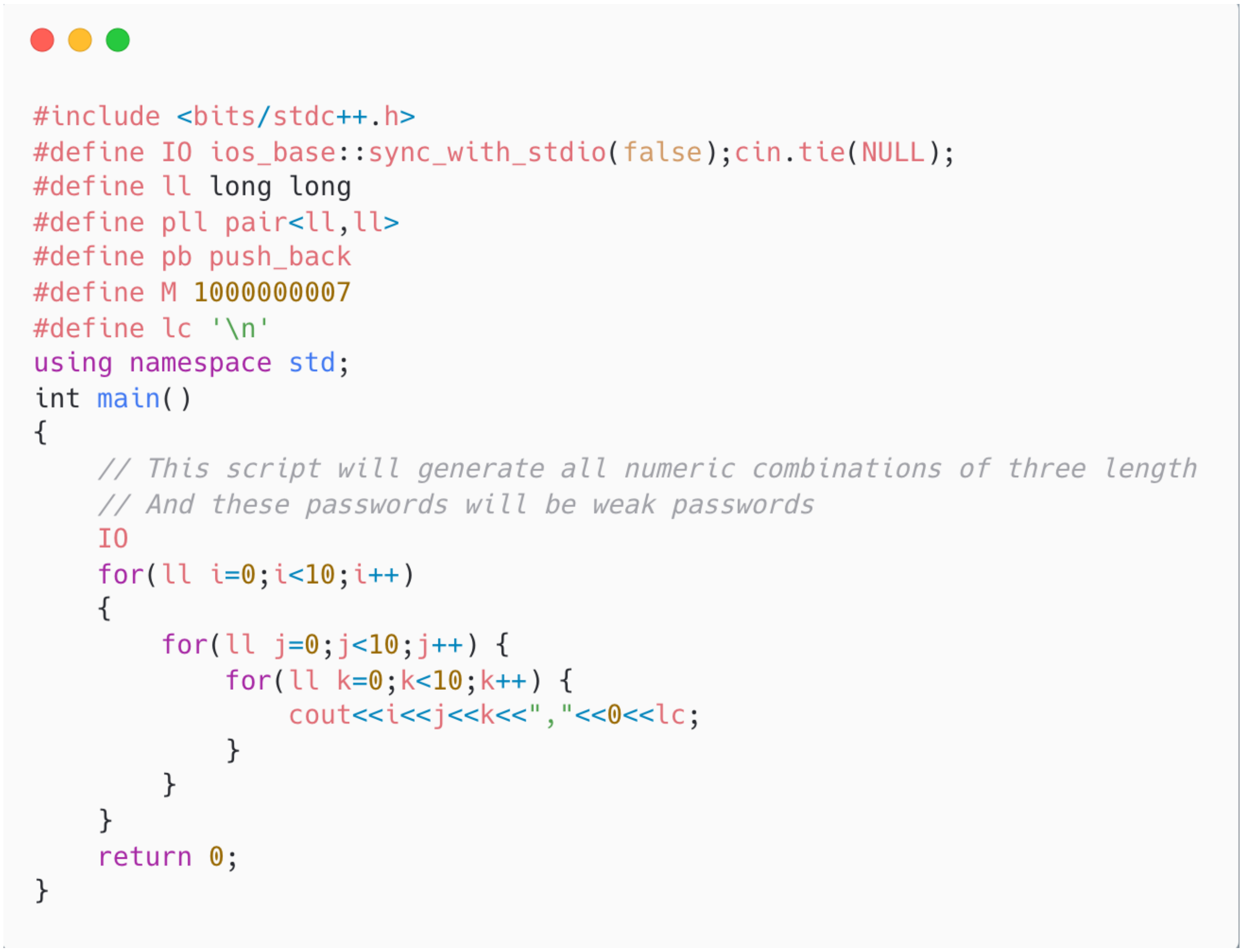
1. Manual Data Generation

For manual data generation we have used multiple web scripts to generate naive and easy passwords. We have produced a scale of almost 10000 data sets using manual data generation.



```
#include <bits/stdc++.h>
#define IO ios_base::sync_with_stdio(false);cin.tie(NULL);
#define ll long long
#define pll pair<ll,ll>
#define pb push_back
#define M 1000000007
#define lc '\n'
using namespace std;
int main()
{
    // This script will generate all numeric combinations of three length
    // And these passwords will be weak passwords
    IO
    for(ll i=0;i<10;i++)
    {
        for(ll j=0;j<10;j++) {
            for(ll k=0;k<10;k++) {
                cout<<i<<j<<k<<","<<0<<lc;
            }
        }
    }
    return 0;
}
```

Fig. 3.2.1.1 - Script to generate numeric weak passwords of three length



```
#include <bits/stdc++.h>
#define IO ios_base::sync_with_stdio(false);cin.tie(NULL);
#define ll long long
#define pll pair<ll,ll>
#define pb push_back
#define M 1000000007
#define lc '\n'
using namespace std;
int main()
{
    // This script will generate all numeric combinations of three length
    // And these passwords will be weak passwords
    IO
    for(ll i=0;i<10;i++)
    {
        for(ll j=0;j<10;j++) {
            for(ll k=0;k<10;k++) {
                cout<<i<<j<<k<<","<<0<<lc;
            }
        }
    }
    return 0;
}
```

Fig. 3.2.1.2 - Script to generate alphanumeric weak passwords of two length

2. Open source datasets

We have collected the majority of our data sets from open-source resources. On the internet there are thousands of coding snippets, open source datasets available to use free of cost and also these data sets are easy to find and are very effective to use.

Although there are few cons of using these data sets such as these data sets are quite messy and need too much cleaning and sometimes these data sets are too detailed and finding content which is of our use is difficult.

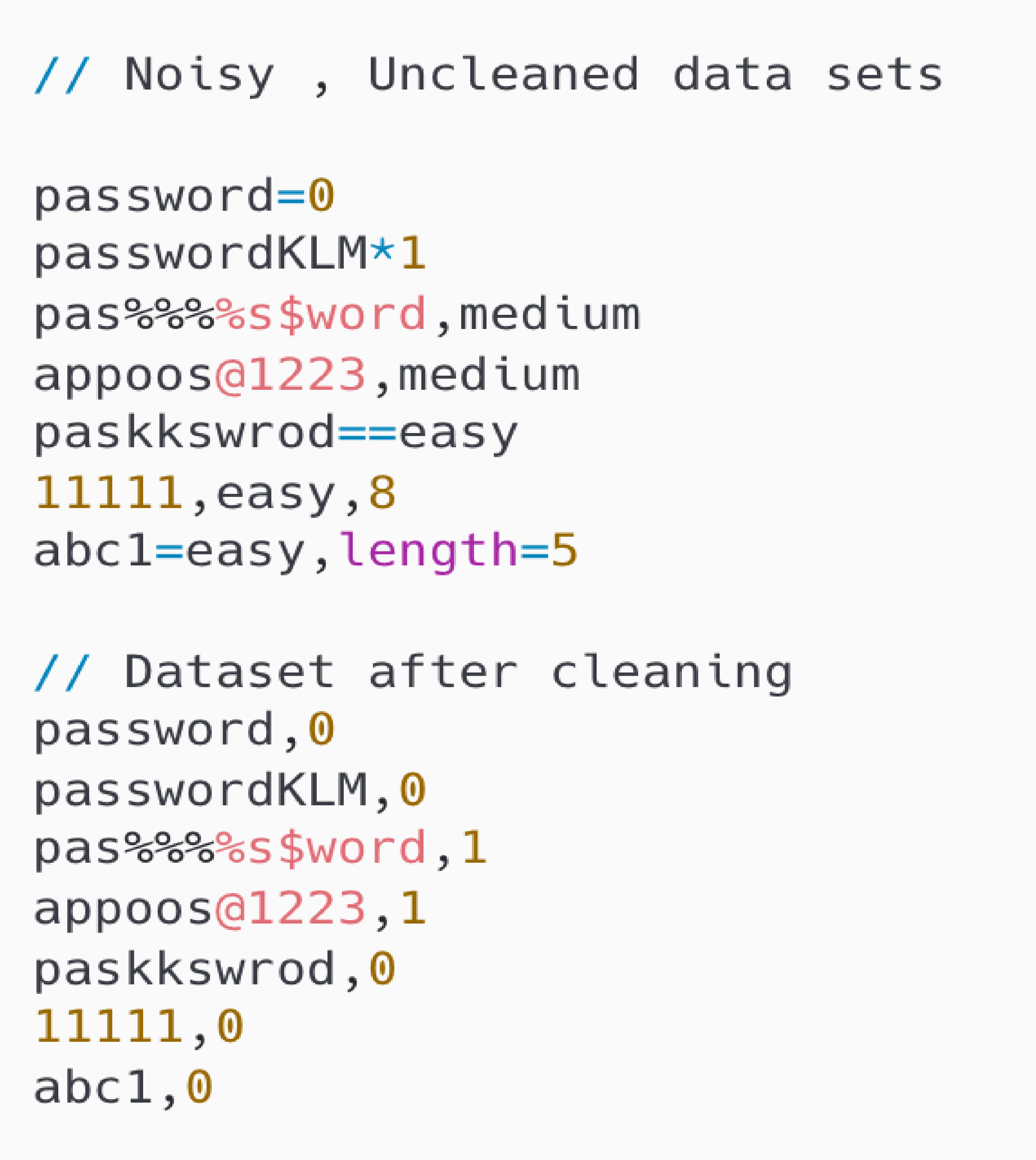
3. Web crawling

Let’s say we would like to extract information like product descriptions and costs from Amazon. We tend to achieve this through repetitive writing or copy-pasting. However clearly, there are approaches to several merchandise on Amazon to try to do this and their costs are modified all of the time. This can be what internet scraping tools are used for. They extract any reasonably info from websites. Plus, these tools search for new information mechanically or manually, attracting the new or updated information and storing it for your quick access.

3.2.2 Data Cleaning

Because the quality of the insights and results you produce is only as good as the data you have, data cleaning is a critical stage in the data science pipeline. Garbage in, garbage out, as the saying goes.

Using stale data to do analysis may result in inaccurate predictions, which will lead to poor decisions and potentially disastrous outcomes. Not only that, but most machine learning algorithms only operate when your data has been cleansed and is ready to be modeled.



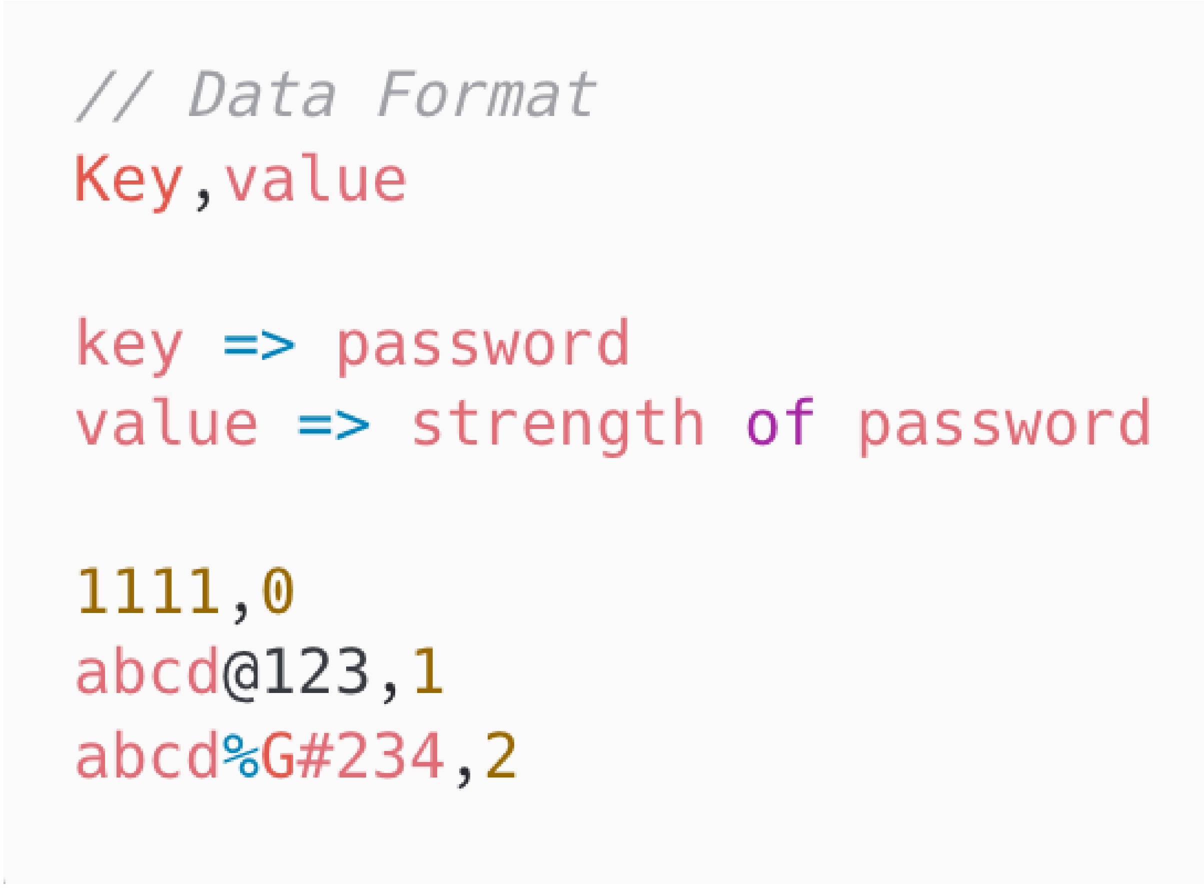
```
// Noisy , Uncleaned data sets

password=0
passwordKLM*1
pas%%%%s$word,medium
appoos@1223,medium
paskkswrod==easy
11111,easy,8
abc1=easy,length=5

// Dataset after cleaning
password,0
passwordKLM,0
pas%%%%s$word,1
appoos@1223,1
paskkswrod,0
11111,0
abc1,0
```

Fig. 3.2.2.1 - Messy and Unclean Data set example

3.2.3 Data Format

For our model we are using key value pairs as input data where key will be our password and value will be strength of our password.

As input we provide a string in which our password and strength of password is separated by a comma and we do this parsing inside our code and we treat first parsed value and second parsed value differently and order of these value should be same



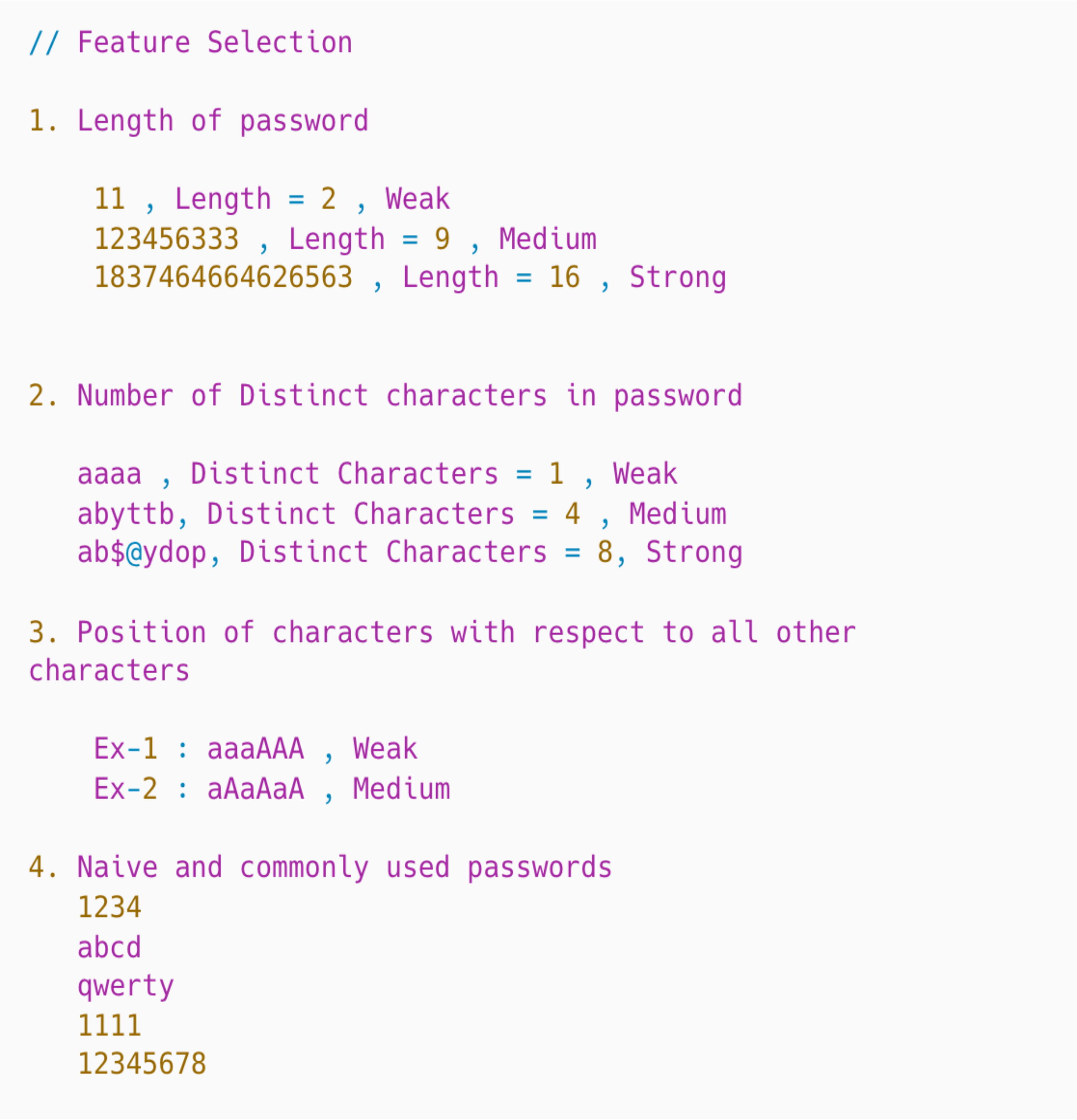
```
// Data Format
Key,value

key => password
value => strength of password

1111,0
abcd@123,1
abcd%G#234,2
```

Fig. 3.2.3.1 - Data Format for Our Model

3.3 Feature Selection

Selecting correct features is one of the most critical steps in ml model training because the whole behavior of our model depends on how our features inforce our model to make decisions.

1. Length of password
2. Number of Distinct characters in password
3. Number of numeric , alphabetic and other characters in password
4. Position of characters with respect to all other characters
5. Number of uppercase and lowercase alphabetic characters in password
6. Naive and commonly used passwords



```
// Feature Selection

1. Length of password

    11 , Length = 2 , Weak
    123456333 , Length = 9 , Medium
    1837464664626563 , Length = 16 , Strong


2. Number of Distinct characters in password

   aaaa , Distinct Characters = 1 , Weak
   abyttb, Distinct Characters = 4 , Medium
   ab$@ydop, Distinct Characters = 8, Strong

3. Position of characters with respect to all other
characters

    Ex-1 : aaaAAA , Weak
    Ex-2 : aAaAaA , Medium

4. Naive and commonly used passwords
   1234
   abcd
   qwerty
   1111
   12345678
```

Fig. 3.3.1 - Feature selection examples

3.4 Working of Algorithm

Decision Tree :

For creating decision trees the resolution shrub is one of the timber-located algorithms in the machine learning rule. It is well instinctive and easy to understand that form it super valuable in answering few of the classic machine learning questions.

1. Classification tree (Yes/No types) :

Such a tree is constructed through a system referred to as binary recursive partitioning. This is an iterative system of splitting the statistics into partitions, after which splitting it up in addition on every of the branches.

2. Regression shrubs (Continuous dossier types) :

Decision trees where the aim changing can take constant values (usually legitimate numbers) are named regression timbers. (for example the price of an apartment, or a patient's distance of stay in a nursing home).

3. Creation of Decision Tree :

In this order a set of preparation models is shabby into tinier and tinier subsets while as long as a mixed conclusion seedling catches incrementally grown. At the end of the knowledge process, a conclusion timber top the preparation set is restored.The key plan search out use a resolution sapling to partition the dossier scope into cluster (or thick) domains and empty (or scanty) domains.In Decision Tree Classification a new instance is top-secret by presenting it to a succession of tests that decide the class label of the model. These tests are arranged in a hierarchic building named a conclusion shrub. Decision Trees trail Divide-and-Conquer Algorithm.

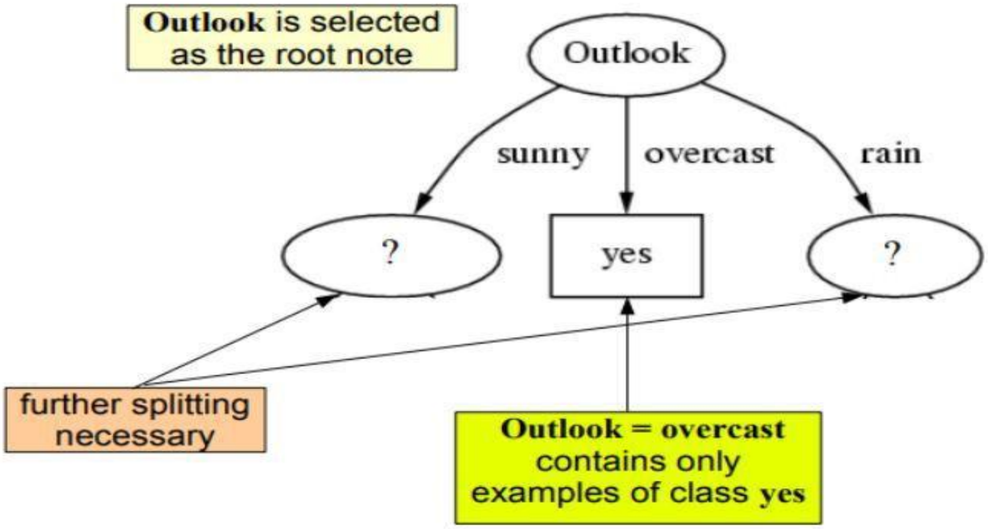


Fig. 3.4.1 - Decision tree generic working

Example of working of algorithm for weak , medium and strong passwords

1. 11

For the password **11** our feature selection condition length will be countered and this password will fall in the **weak** passwords category since this is a very short password.

2. aaaaaa

For the password **aaaaaa** our feature selection condition length will be passed but the features condition of having at least x number of distinct characters will fail and this password will fall into the **weak** passwords category.

3. abcd123

For the password **abcd123** our feature selection condition length will be passed but the feature condition of having at least x number of distinct characters will also pass and this password will fall in **medium** password category.

4. abcd#12@3

For the password **abcd123** our feature selection condition length will be passed but the feature condition of having at least x number of distinct characters will also pass and this password will fall in **medium** password category.

5. aaaaAAA

For the password **aaaaAAA** our feature selection condition length will be passed but the feature condition of having at least x number of distinct characters will fail and the feature condition of regex expressions where two same characters should maintain some distance will also fail. So this password will fall into the **weak** password category.

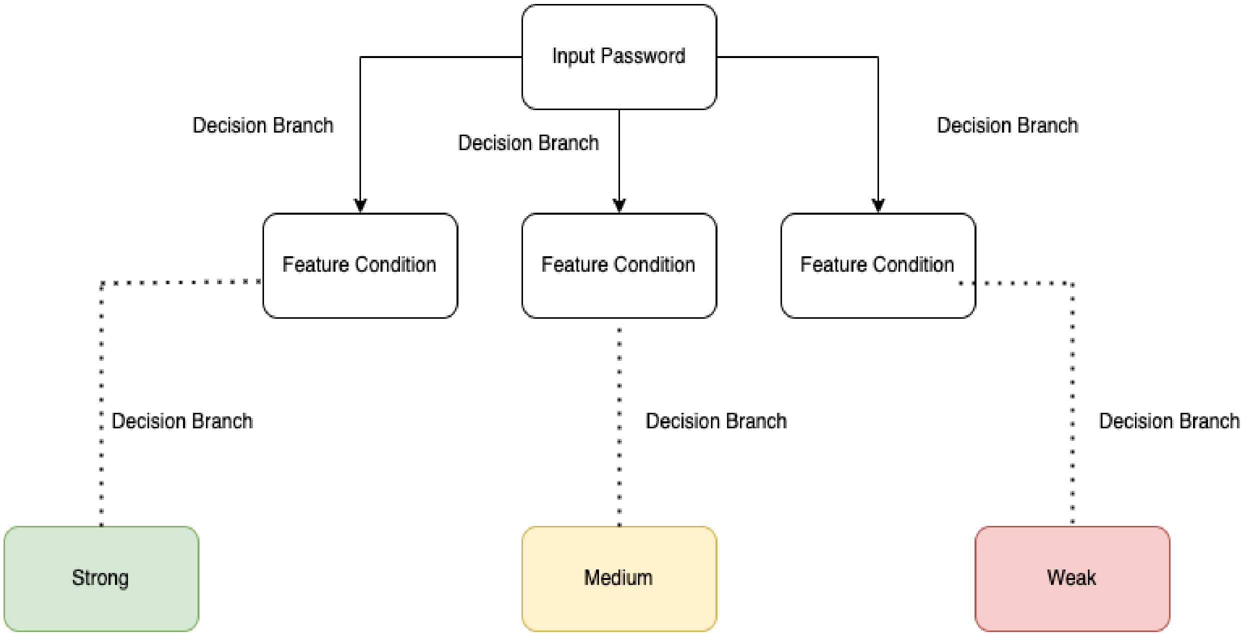


Fig. 3.4.2 - Algorithm flow diagram

Chapter-4
Results

We have trained our model on almost 7 lakh inputs and also we have tested our model on almost 200 inputs some of the statics of our training and testing of models are mentioned below :

Title	**Size**
Weak Passwords	89669
Medium Passwords	496749
Strong Passwords	82929
Total Passwords	**6669347**

Table 4.1 Training Dataset Size

Title	**Size**
Weak Passwords	1000
Medium Passwords	1000
Strong Passwords	500
Total Passwords	**2500**

Table 4.2 Testing Dataset Size

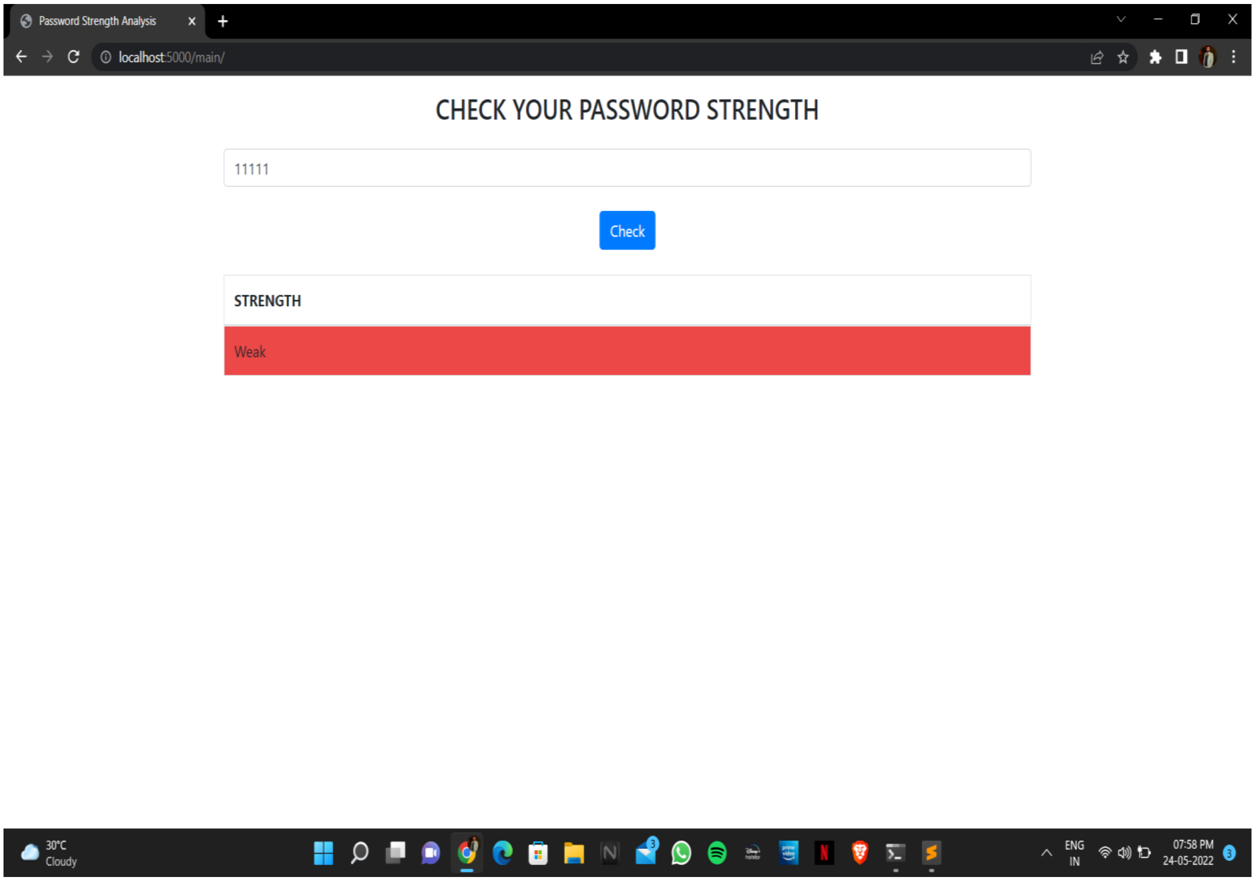
Title	**Percentage**
Correct Weak Passwords	**99**
Medium Passwords	**95**
Strong Passwords	**97**

Table 4.2 Accuracy of Model

4.1 Weak Password

Features :

1. **Medium Length passwords (<=5))**
2. **Passwords having many distinct characters (<=5)**
3. **Passwords having non repetitive characters (>=3)**
4. **Regex checks for having a optimal distance between same characters**



CHECK YOUR PASSWORD STRENGTH

11111

STRENGTH
Weak

Fig. 4.1.1 - Model Result for weak password

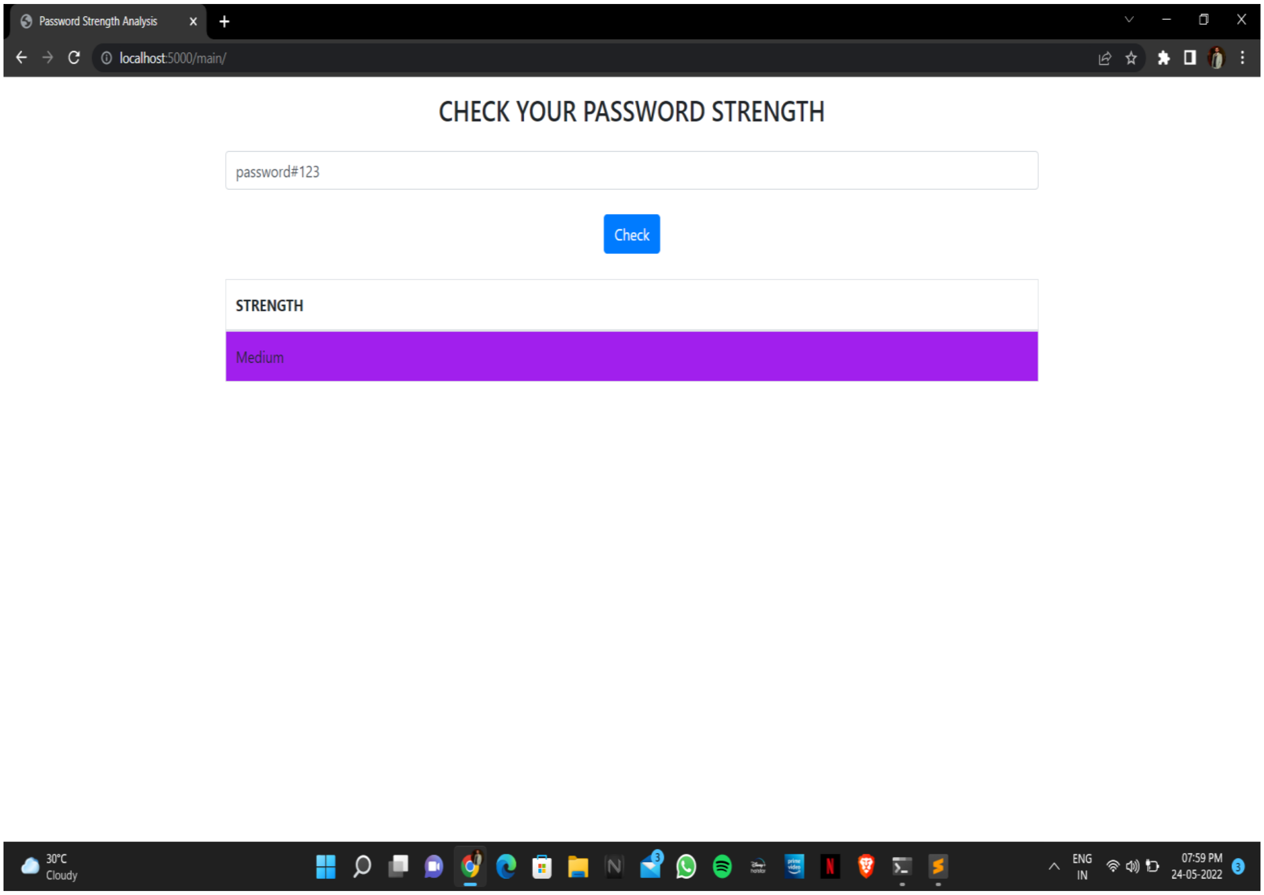
Examples :

1. **111111**
2. **abcd**
3. **ac**
4. **1234**
5. **4321**
6. **0000**
7. **9999**

4.2 Medium Password

Features :

1. **Medium Length passwords (length - [6-8])**
2. **Passwords having many distinct characters (>=5)**
3. **Passwords having non repetitive characters (<=1)**
4. **Regex checks for having a optimal distance between same characters**



CHECK YOUR PASSWORD STRENGTH

password#123

Check

STRENGTH

Medium

Fig. 4.2.1 - Model Result for medium password

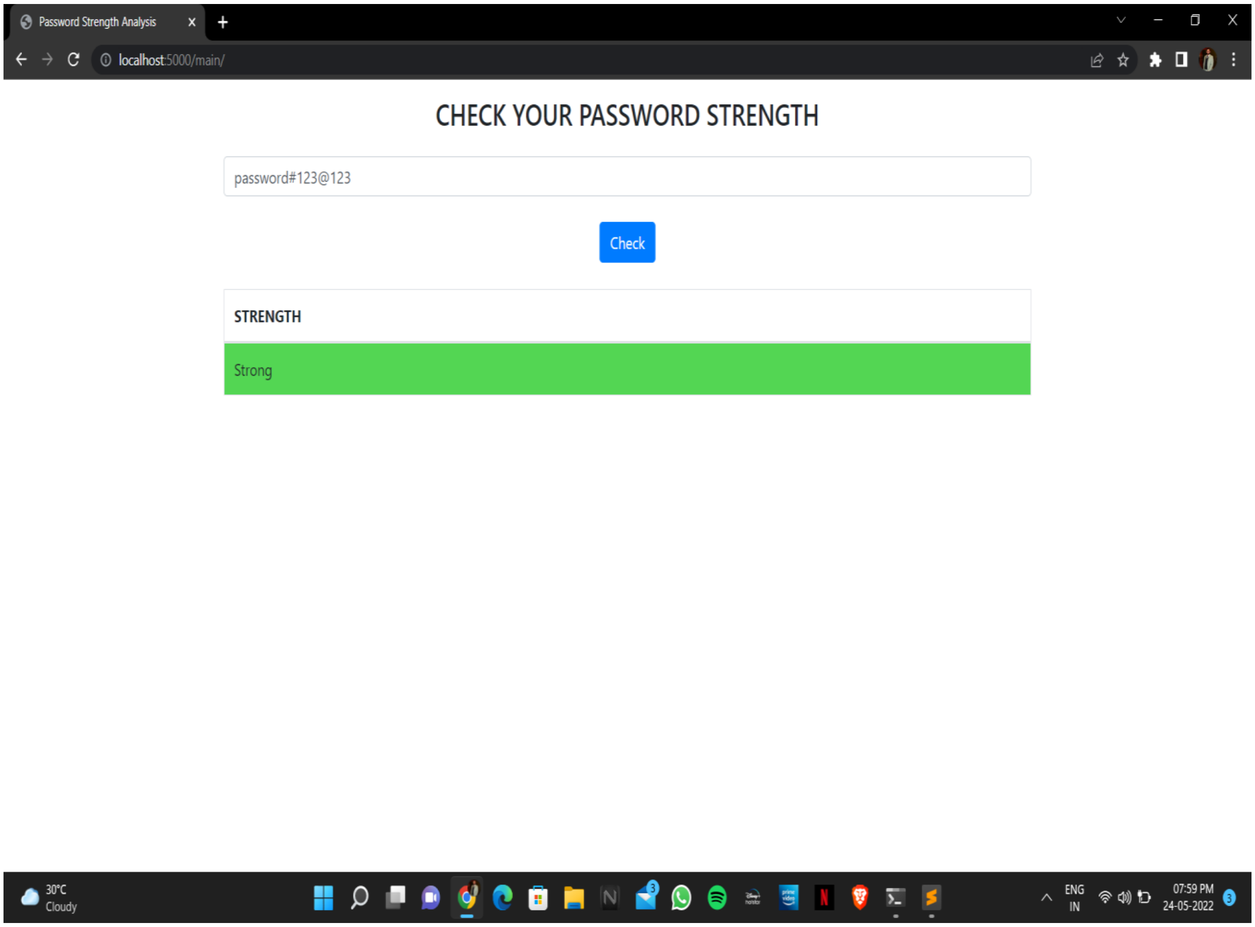
Examples :

1. **ok_jshshd**
2. **12873637**
3. **kdkd)***
4. **ksksk&$**
5. **ab12cd#**
6. **koskdjdh**

4.3 Strong Password

Features :

1. **Lengthy passwords (length >= 8)**
2. **Passwords having many distinct characters (>=6)**
3. **Passwords having non repetitive characters (<=2)**
4. **Regex checks for having a optimal distance between same characters**



CHECK YOUR PASSWORD STRENGTH

password#123@123

Check

STRENGTH
Strong

Fig. 4.3.1 - Model Result for strong password

Examples :

1. **anhdhdj$#%8K**
2. **kjskfhsdkfhs&@SD**
3. **,sjfnscjfsdhfjufhsf**
4. **ksjknauhubmjd$#%**
5. **dkajfns000778GGSS**
6. **kfnucbhfyex@@#$$$**
7. **skjd36464747JKSL**
8. **00oonsjfhaxjhhywegr**

Chapter-5
Conclusion

We have successfully built a model which is predicting passwords with a very good accuracy. We are trying to train our model on more robust data sets so that we can improve our accuracy.
The model we have built works on the basis of a decision tree algorithm where decision steps and conditions are features of weak , medium and strong passwords.
These features are hardcoded conditions and regex based conditions which a password may or may not fall in.

5.1 Advantages

We are using trie for pattern searching which is quite fast to predict password strength than running the whole algorithm again and again so for any password first our model will search if our current password is prefix or suffix of any existing result if such password exist then our model do not have to run again and we will be able to predict passwords directly from input.
But this trie technique can be applied only for weak passwords.

Example :
If **abcd1234** is a weak password then **abcd**, **abc**, **abcd12**, **abcd123** will also be weak passwords.

5.2 Limitations

For now we can not predict password strengths on a personalized basis, which means some password strength for a person can be a weak password for another person.
And also we don't have too many good feature conditions which can identify complex regex patterns in passwords.

Example :
If a person is having the name "**Group3**" then the password "**Group3@123**" will be a weak password for him/her but "**Group15@123**" might be a relatively strong password for him.
So for now our model can not predict personalized passwords strengths.

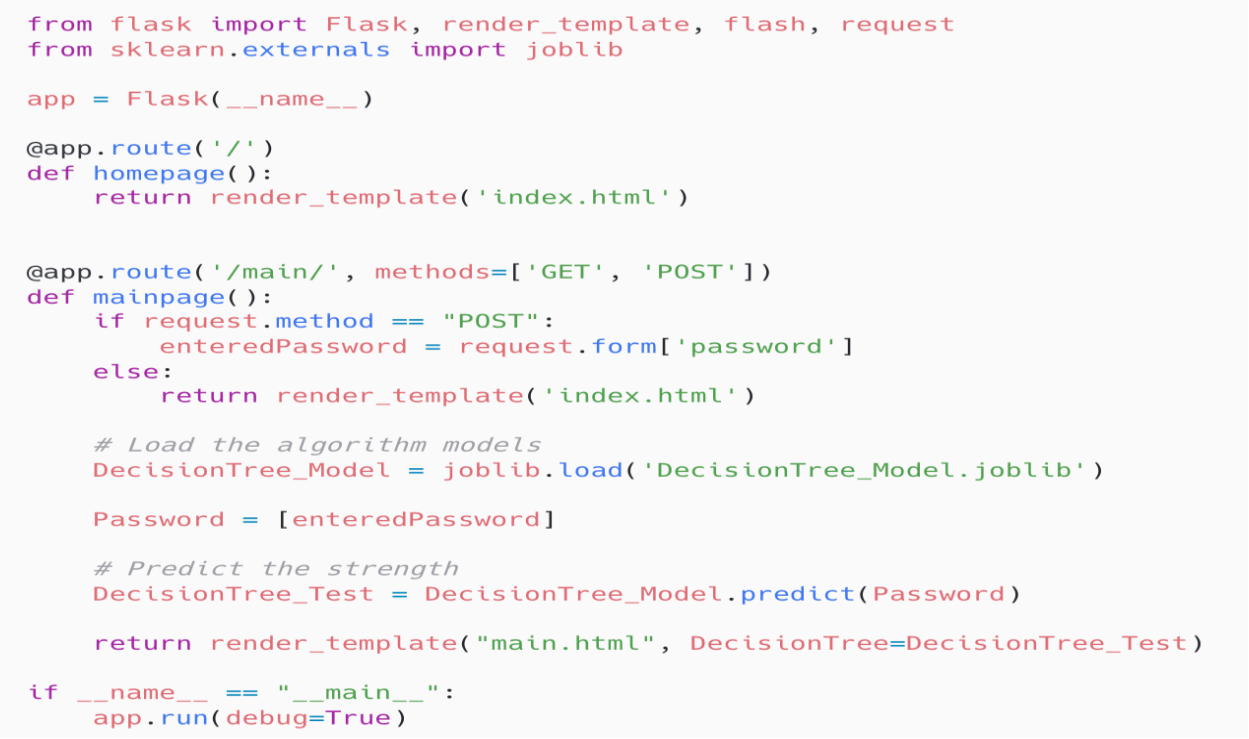
5.2 Future Scope

In the future scope of this project we have two new features

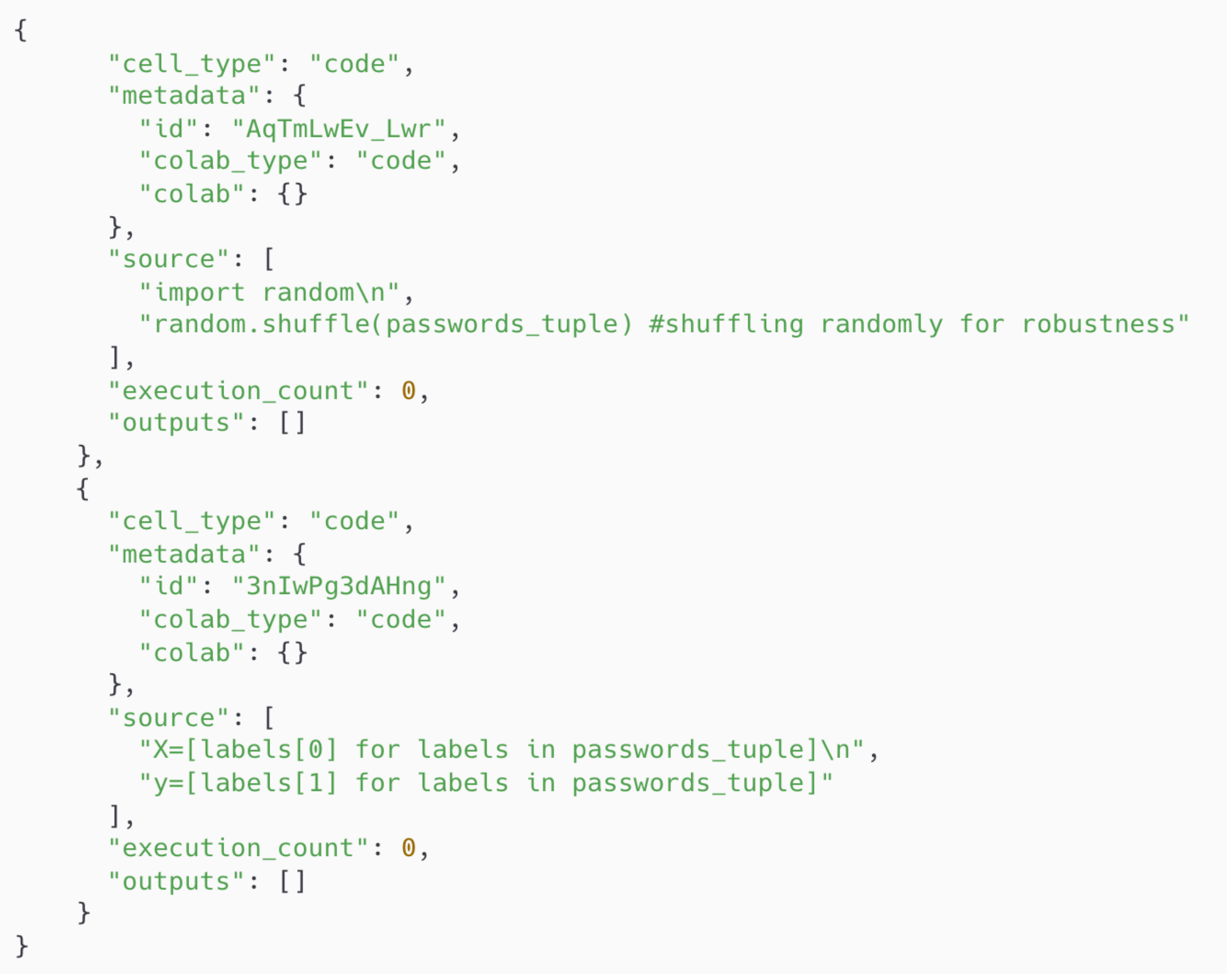
1. Memorable password suggestion
2. Personalized password strength prediction

Nowadays having a strong password which should be memorable is quite a tough task and generating these types of passwords will also require quite complex feature conditions so for now we are putting this feature in our future scope.
And personalized password suggestion means Let's say we have two people A and B then some password A@123 will be easy for person A and the same password will be strong for person B so we want to work on personalized password prediction in our future scope.

Appendix-A Model Code Snippets



```
from flask import Flask, render_template, flash, request
from sklearn.externals import joblib

app = Flask(__name__)

@app.route('/')
def homepage():
    return render_template('index.html')


@app.route('/main/', methods=['GET', 'POST'])
def mainpage():
    if request.method == "POST":
        enteredPassword = request.form['password']
    else:
        return render_template('index.html')

    # Load the algorithm models
    DecisionTree_Model = joblib.load('DecisionTree_Model.joblib')

    Password = [enteredPassword]

    # Predict the strength
    DecisionTree_Test = DecisionTree_Model.predict(Password)

    return render_template("main.html", DecisionTree=DecisionTree_Test)

if __name__ == "__main__":
    app.run(debug=True)
```



```
{
      "cell_type": "code",
      "metadata": {
        "id": "AqTmLwEv_Lwr",
        "colab_type": "code",
        "colab": {}
      },
      "source": [
        "import random\n",
        "random.shuffle(passwords_tuple) #shuffling randomly for robustness"
      ],
      "execution_count": 0,
      "outputs": []
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "3nIwPg3dAHng",
        "colab_type": "code",
        "colab": {}
      },
      "source": [
        "X=[labels[0] for labels in passwords_tuple]\n",
        "y=[labels[1] for labels in passwords_tuple]"
      ],
      "execution_count": 0,
      "outputs": []
    }
}
```

References

[1] Ruffo, G., & Bergadano, F. (2005, September). Enfilter: a password enforcement and filter tool based on pattern recognition techniques. In the International Conference on Image Analysis and Processing (pp. 75-82). Springer, Berlin, Heidelberg.

[2] Cristianini, N., & Shawe-Taylor, J. (2000). An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.

[3] Bergadano, F., Crispo, B., & Ruffo, G. (1997, April). Proactive password checking with decision trees. In Proceedings of the 4th ACM Conference on Computer and Communications Security (pp. 67-77).

[4] Soman, K. P., Loganathan, R., & Ajay, V. (2009). Machine learning with SVM and other kernel methods. PHI Learning Pvt. Ltd..

[5] https://en.wikipedia.org/wiki/Password_strength

[6] https://en.wikipedia.org/wiki/Brute-force_attack

[7] https://it.ufl.edu/it-policies

[8] https://medium.com/rangeforce/password-cracking-6d9612915f03

[9] https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

[10] https://medium.com/analytics-vidhya/tagged/data-extraction

[11] Egelman, S., Sotirakopoulos, A., Muslumov, I., Beznosov, K., & Herley, C. Does my password go up to eleven? The impact of password meters on password selection. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.