State University of New York at Buffalo

Predictive Model for Password Strength Assessment

MSBA Practicum

Mr. Nicholas R. Hartnett

MGS-649

Professor Dominic Sellitto

5/12/2023

Table of Contents

Introduction:

In the modern day, advancements are constantly being made to bolster security and incorporate machine learning and predictive modeling to optimize organizations' various functionalities. With more and more personal information being stored online, password security has become an increasingly critical concern. Everyday cyber security, IT, and IoT teams combat bad actors and unethical hackers attempting to exploit organizations using intricate attacks such as brute-force attacks, keyboard sniffing, and DOS to gain hypersensitive data. With recent innovations like ChatGPT, becoming a bad actor does not require the coding abilities of the past. Any one person with the ambition to exploit an organization poses a threat financially to the organization as well as the private security of the individuals within the organization and their data. Once an organization has been breached, the passwords within that leak may be used to attack the individual on platforms outside of the organization itself. Weak passwords pose a significant threat to online security. With this information on the vulnerability of password security in an organization, there is a growing need for more intensive methods to evaluate and optimize password strength.

This report presents the results of my Master of Science practicum which focuses on solving the threat presented by weak passwords. This project is an analysis of passwords using predictive analytics. I utilized the “rockyou.txt” file, which is a well-known compilation of leaked passwords stolen from the company RockYou in 2009. The file of passwords provided by Kali Linux provides a password dictionary file as part of its standard installation. Kali Linux is an open-source project that is maintained and funded by Offensive Security, a provider of world-class information security training and penetration testing services. Because the “rockyou.txt” file contains 14,341,564 unique passwords I used systematic sampling without replacement to obtain a sample size of 500 passwords. The systematic sample was completed using Python programming and file handling. With the sample size selected, I will then continue to data pre-processing. I then extracted traits of the passwords in the form of counting unique characters. I have extracted the counts of lower-case, upper-case, special characters, total digits, total string length, the count of vowels, the count of consonants, and a binary variable that determines if the string is a palindrome.

By using the passwords from RockYou.txt I aimed to develop a machine-learning model for password strength prediction using predictive analytics techniques. The project involved followed the life cycle of a data analysis project as discussed in my predictive analytics course which abides by the following steps: the business issue understanding, data understanding, data preparation, exploratory analysis and modeling, validation, and visualization and presentation of the data. In this report, I covered the visualization and presentation steps within the graphical user interface development section. The purpose of this report is to provide an overview of the project, discuss the methodologies used, and present the findings and conclusions. The report is organized as follows:

- <u>Business Issue Understanding:</u> define business objectives, gather information required, determine appropriate analysis method, clarify the scope of work, and identify deliverables.

- Tools Utilized: A comprehensive list of all tools used for the project and their definitions or a comparable explanation of what the tool does and how it operates.
- Data Understanding: Collect initial data, identify data requirements, determine data availability, and explore data and characteristics.
- Data Preparation: Cleanse, format, blend, and sample.
- Exploratory Analysis and Modeling: Develop methodology, determine important variables, build a model, and assess the model.
- Validation: Evaluate results, review process, and determine next steps.
- Graphical User Interface: explain the tools used to implement the GUI and the steps to launch and run the application on any device.
- Summary: This section summarizes the main findings, takeaways, lessons learned, and conclusions of the project.
- Appendix: This section provides supplementary information, including screenshots from my code, a data dictionary, and additional analyses.

Overall, this report aims to provide a comprehensive account of the project, including its goals, methods, and outcomes. By developing a machine learning model for password strength prediction, this project contributes to the growing field of predictive analytics and offers practical solutions to the problem of password security.

Business Issue Understanding:

The primary objective of this project was to develop a machine learning model for password strength prediction that would help improve password security beyond the standard “minimum characters, one special character, one capitalized letter” basic requirements. To achieve my overall objective, I declared the following sub-objectives:

- Gather data containing the passwords (“RockYou.txt”) and then create password characteristics with the variables aforementioned in the introduction.
- Train a machine learning model on this data to accurately predict password strength based on the data at hand.
- Develop a deployable graphical user interface that allows users within an organization to interact with the model and assess the strength of their passwords.

The appropriate analysis method for this project was predictive analytics using machine learning algorithms. This method allowed me to train a model on the gathered data and use it to make accurate predictions about the strength of potential passwords before an employee within an organization using it as their password. The scope of work for this project was limited to developing a machine learning model for password strength prediction as well as creating a graphical user interface to make the model deployable and interactable for the organization. Other aspects of online security, such as network security and secure data storage, were outside the scope of this project.

Tools Utilized:

- Python version 3.9.13: An interpreted, object-oriented, high-level programming language with dynamic semantics.

- Gradio: a free and open-source Python library that allows you to develop an easy-to-use customizable component demo for your machine learning model.
- Pickle: A generic object serialization module that can be used for serializing and deserializing objects for reloading trained machine learning models.
- Microsoft Excel: a software program created by Microsoft that uses spreadsheets to organize numbers and data with formulas and functions.
- Orange Data Mining Application: open-source data processing suite. It works as a canvas where the user can drag widgets for specific data operations, and then load their data into the program.
- Microsoft Word: Microsoft word is a word processing software application primarily used for creating documents in different formats.
- University of Illinois at Chicago Password Strength Test: This is a publicly available strength test located at URL https://www.uic.edu/apps/strong-password/.
    - This website was used to create the "PasswordStrength" target variable for my machine learning model.
- ChatGPT: an AI chatbot that uses natural language processing to create humanlike conversational dialogue.
    - As a recent student of Python this tool was used to assist in understanding the code and assisting in grammatical errors within the report.

Data Understanding:

To collect initial data for this project I needed to obtain the text file "RockYou.txt". To accomplish this, I went to the website www.Kaggle.com and downloaded the text file.

The “RockYou.txt” has millions of leaked passwords from the year 2009 with all passwords being stored as String values. The simplest type of attack that an unethical hacker would perform on an organization to acquire passwords by only manipulating the complexity of those passwords is by performing a brute force attack. A brute force attack occurs when a hacker attempts to guess a user's login credentials either manually or by SQL injections most commonly. This is typically through standard password combinations from existing databases or personal identification numbers. Logically, the key to successfully guessing a password to gain unauthorized access is by either entering the exact password of the user or by entering a different password that becomes hashed into the same values as a different password. The latter is only possible if the hacker can determine collisions from the hash algorithm. In cybersecurity, a hash algorithm collision is when two pieces of data in a hash table share the same hash value. In my case, the hash is irrelevant to my project as it is a process performed after the password is created. So, the only logical way to ensure password strengthening was by improving the complexity of the password string itself.

A string is composed of individual characters that may be individually analyzed and assessed to indicate a password’s strength. For my project, I chose to focus on the following categories: lower-case, upper-case, special character, total digits, total string length, the count of vowels, the count of consonants, and a binary variable that determines if the string is a palindrome. To obtain all these categories I used Python file handling to take 500 passwords using a systematic sample to count their fields and then store them into a .csv file for analysis in Orange Data Miner. The file “SampleGenerator.py” is where all coding to acquire the variable counting was performed. These categories are the variables that the machine learning model used to predict password strength.

With the predictors selected, I needed a variable to determine if the given variables resulted in a good or bad password. To acquire a target variable, I used the publicly available Strength test website provided by the University of Illinois at Chicago. With all variables selected, I created my data dictionary (Appendix: **Image 1**) and moved forward to the next step of data preparation.

Data Preparation:

With all variables selected, before the actual modeling, I had to ensure that the data was properly cleaned and formatted with no missing data. To accomplish this, I used the Excel feature “Sort and Filter” and “Delete row” functions. When my code from the Python file “SampleGenerator.py” was executed, it stored all the variables in a .csv, but it skipped every other row, leaving many blank spaces. To get rid of the blank spaces and format the file properly I used the “delete row” function after highlighting all empty rows. Next, there was no missing data, but from manually transcribing the password strength from the website to my .csv I needed to ensure that all variables were uniform so that Orange Data Miner could decipher the data. To accomplish this, I used the Excel feature “Sort and Filter” (Appendix: **Image 2**).

With all the passwords from the text file coming from the year 2009, the password’s overall strength is lackluster. In the last 14 years password strength requirements have increased exponentially. When the 500 passwords were placed in the University of Illinois at Chicago Password Strength Test, they did not receive high scores. The University of Illinois at Chicago Password Strength Test grades passwords as very weak, weak, good, strong, and very strong. Of the variables, I had selected from “RockYou.txt” the strength values varied between only very weak, weak, and good. No variables from my sample of the original text file had a strength score of strong or very strong. Ultimately this impacted the model’s ability to predict a password strength because it was only able to train off of data that was very weak, weak, or good. With all data preparation completed, I moved forward to the next step, exploratory analysis, and modeling.

Exploratory Analysis and Modeling:

To conduct my exploratory analysis and modeling I used Orange Data Miner. Upon selecting a file prior to the actual modeling, I needed to firstly ensure that the data was following the format from my data dictionary (Appendix: **Image 1**). Variables must have the correct type, role, and values for Orange Data Miner to accept them into a model. The variables were defined as the having the following types and roles:

- “PasswordLength”: Type = numeric, Role = feature
- “VowelCount”: Type = numeric, Role = feature
- “ConsonantCount”: Type = numeric, Role = feature
- “NumberCount”: Type = numeric, Role = feature
- SpecialCharacterCount”: Type = numeric, Role = feature
- “IsPalindrome”: Type = categorical, Role = feature
- “lowercaseLetterCount”: Type = numeric, Role = feature

- “uppercaseLetterCount”: Type = numeric, Role = feature
- “PasswordStrength”: Type = categorical, Role = target
- “Password”: Type = text, Role = skip

With all variable's roles and types defined, I could then logically deduce which algorithms would be feasible for my model. Before actually implementing a predictive model, I needed to create a data sample from my 500 rows of data. I decided to use a fixed proportion of data using deterministic sampling with 70% of the data for training and 30% going to the test data.

Next, I selected four predictive model algorithms. Knowing that my target variable was categorical, and my predictors were primarily numeric with one as a categorical I narrowed my model selections to Logistic Regression, Forest, Random Forest, and Naïve Bayes. After creating the models with Orange Data Miner, I used the Test and score feature along with a confusion matrix to assess the models. The highest scoring model was Random Forest. The random forest predictive model operates by drawing multiple random samples with replacements from the data then using a random subset of predictors, fitting a classification tree to each sample, the predictions are then combined from the individual trees to obtain improved predictions. The scores for my model using the random forest model for each category of the password are as follows:

- Evaluation results for target “Very Weak” or “1” (Appendix: **Image 3**):
  - AUC = 1.0
  - CA = 0.994
  - F1 = 0.996

  - Precision = 0.996
  - Recall = 0.996
- Evaluation results for target “Weak” or “2” (Appendix: **Image 4**):
  - AUC = 0.999
  - CA = 0.989
  - F1 = 0.973
  - Precision = 0.973
  - Recall = 0.0973
- Evaluation results for target “Good” or “3” (Appendix: **Image 5**):
  - AUC = 0.998
  - CA = 0.994
  - F1 = 0.800
  - Precision = 0.800
  - Recall = 0.800

With the exploratory analysis and modeling complete I moved forward to the next step, validation.

Validation:

With the provided scores I evaluated them and found them to be acceptable. As stated previously, the passwords are from 2009 and thus are not nearly as complex or standardized as the current day. With a lack of passwords defined as good and no passwords meeting the criteria of strong or very strong, the model can only train to be exceptionally good at distinguishing between weak and very weak passwords. While on the surface this appears to undermine the construct of creating an application that classifies password strength, the scores are more than acceptable given the nature of the dataset. To further breakdown and understand how well the model is deciphering the passwords I analyzed the results by setting up a confusion matrix to confirm my suspicions of the dataset (Appendix: **Image 6**). The confusion matrix confirmed my suspicions that the nature of the dataset resulted in a bias in distinguishing between "Very Weak" and "Weak" passwords. With my validation complete and model optimized, I moved forward to the deployment phase in the step visualization and presentation.

Graphical User Interface:

To create my graphical user interface, I first needed to mobilize the model by transforming it from the Orange Data Miner .ows file to a functional pickle file that could interact with an application. My model of the random forest was saved as "PasswordModel.pkcls". Using open-source code acquired from my predictive analytic class I decided to use Gradio and Orange3 libraries with Python 3.9.13 to create my GUI. The GUI contains a simple textbox labeled "Enter Password:" to insert a potential password. When the user selects to submit the form sends them to the function that determines individual variable counts and then sends those variables to the pickle model and returns a score in the password into a textbox labeled "Password Strength:". The model may be visualized in (Appendix: **Image 7**). The Python GUI is executed by running the file labeled "PasswordStrengthApp.py". The steps and requirements to run the application are as follows:

- Have a Python version of 3.9.

- Install Gradio and Orange3 in an accessible location for the file.
- At the location of the “PasswordStrengthApp.py” you must execute the file in command prompt by running the command: “py PasswordStrengthApp.py”.
- The app may be viewed under localhost at: “http://127.0.0.1:7860/”.

Despite my best efforts, as a recent student of Python and Gradio the application will run, but upon submission of the password the password strength field will result in an “Error”. I am unsure if the model is timing out when I compute the variable field counts, but after all research transpired, I was unable to find the source of the error.

Summary:

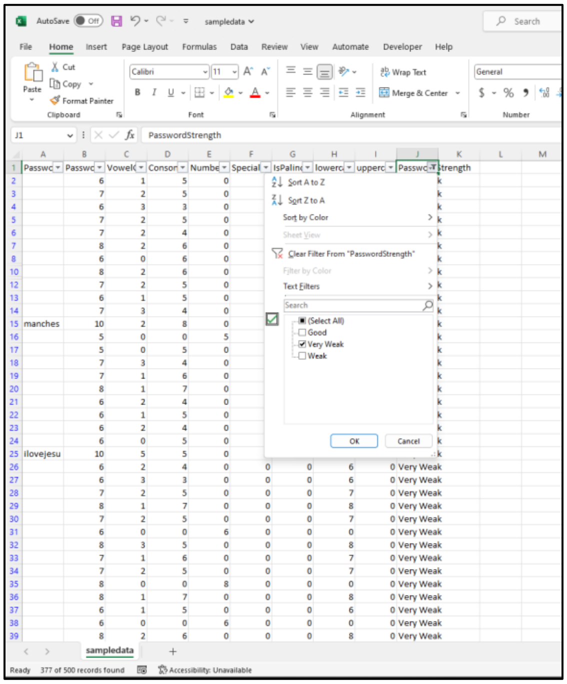
Data analytics has become an essential tool in today's data-driven world, and the surplus of data and progress in technology and the security fields have made data analytics streamlined and more impactful than ever. Ensuring that data is accurate and free from errors is crucial to obtaining meaningful insights from data analysis. In addition, it is important to choose the appropriate data analysis techniques and methods based on the nature of the data and the research question at hand. The growing importance of data analytics has led to high demand for skilled professionals in this field. To succeed in this field, it is obligatory to have a combination of technical skills, including programming and statistical analysis, as well as soft skills such as critical thinking, problem-solving, and communication. This project allowed me to practice such skills and develop a product. Despite the flaws of my GUI, I have learned lessons and look forward to applying what I have learned to my career. In conclusion, data analytics is an essential tool in today's world, and its importance will only continue to grow as businesses and organizations seek to gain insights from the vast amounts of data available to them.

Appendix

Image 1: Data Dictionary.

Variable Name:	Character Length:	Data Type:	Role:	Values:	Description:
Password	128	String/Text	Meta	Characters	This field holds the stored passwords.
PasswordLength	128	Numeric	feature	Any Whole Number	This field indicates the length of the password as an integer.
VowelCount	128	Numeric	feature	Any Whole Number	This field indicates the count of the vowel letters in the password as an integer.
ConsonantCount	128	Numeric	feature	Any Whole Number	This field indicates the count of the consonant letters in the password as an integer.
NumberCount	128	Numeric	feature	Any Whole Number	This field indicates the count of the numbers in the password as an integer.
SpecialCharacterCount	1	Categorical	feature	(0,1)	This field indicates the count of the special characters in the password as an integer.
IsPalindrome	1	Categorical	feature	(0,1)	This field indicates if the password is a palindrome.
lowercaseLetterCount	128	Numeric	feature	Any Whole Number	This field indicates the count of the lowercase letters as an integer.
uppercaseLetterCount	128	Numeric	feature	Any Whole Number	This field indicates the count of the uppercase letters as an integer.
PasswordStrength	1	Categorical	Target	(1,2,3)	This field indicates the strength of the password: 1 = Very Weak, 2 = Weak, and 3 = Good.

Image 2: Sort and Filter in Excel.



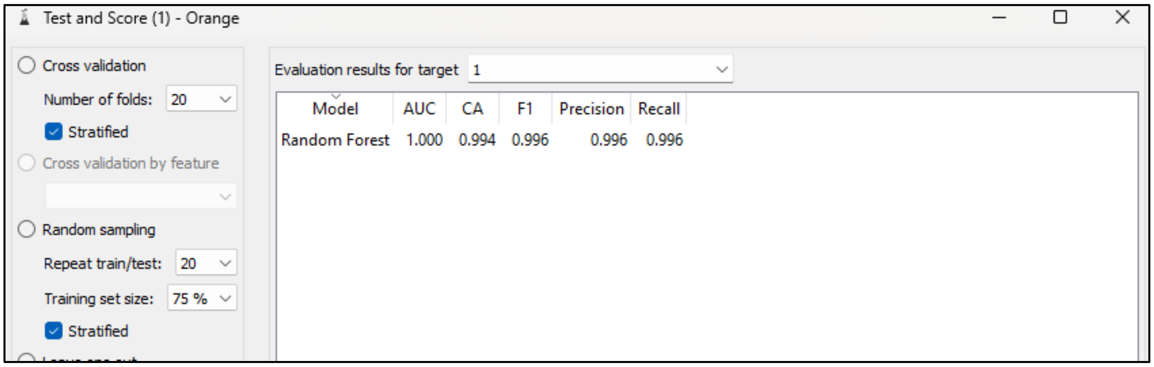
J1 | PasswordStrength

■ (Select All)
□ Good
☑ Very Weak
□ Weak

	A	B	C	D	E	F	G	H	I	J	K
1	Passwc	Passwc	Vowel(	Conson	Numbe	Special	IsPalin(	lowerc	upperc	Passwc	trength
2		6	1	5	0					k	
3		7	2	5	0					k	
4		6	3	3	0					k	
5		7	2	5	0					k	
6		7	2	4	0					k	
7		8	2	6	0					k	
8		6	0	6	0					k	
10		8	2	6	0					k	
12		7	2	5	0					k	
13		6	1	5	0					k	
14		7	3	4	0					k	
15	manches	10	2	8	0					k	
16		5	0	0	5					k	
17		5	0	5	0					k	
18		7	3	4	0					k	
19		7	1	6	0					k	
20		8	1	7	0					k	
21		6	2	4	0					k	
22		6	1	5	0					k	
23		6	2	4	0					k	
24		6	0	5	0					k	
25	ilovejesu	10	5	5	0					k	
26		6	2	4	0	0	0	6	0	Very Weak	
27		6	3	3	0	0	0	6	0	Very Weak	
28		7	2	5	0	0	0	7	0	Very Weak	
29		8	1	7	0	0	0	8	0	Very Weak	
30		7	2	5	0	0	0	7	0	Very Weak	
31		6	0	0	6	0	0	0	0	Very Weak	
32		8	3	5	0	0	0	8	0	Very Weak	
33		7	1	6	0	0	0	7	0	Very Weak	
34		7	2	5	0	0	0	7	0	Very Weak	
35		8	0	0	8	0	0	0	0	Very Weak	
36		8	1	7	0	0	0	8	0	Very Weak	
37		6	1	5	0	0	0	6	0	Very Weak	
38		6	0	0	6	0	0	0	0	Very Weak	
39		8	2	6	0	0	0	8	0	Very Weak	

377 of 500 records found

Image 3: Evaluation Results for “Very Weak” or “1”.

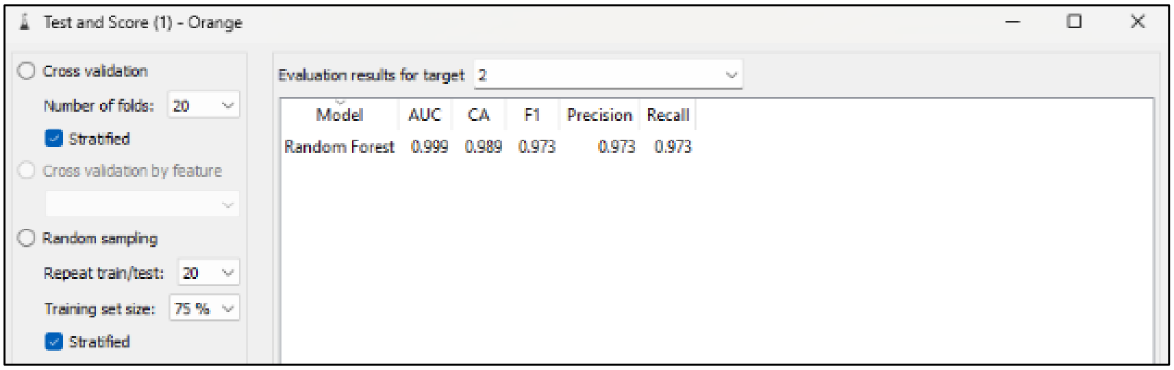

Test and Score (1) - Orange

○ Cross validation
Number of folds: 20
☑ Stratified
○ Cross validation by feature
○ Random sampling
Repeat train/test: 20
Training set size: 75 %
☑ Stratified

Evaluation results for target 1

Model	AUC	CA	F1	Precision	Recall
Random Forest	1.000	0.994	0.996	0.996	0.996

Image 4: Evaluation Results for “Weak” or “2”.



Test and Score (1) - Orange

○ Cross validation

Number of folds: 20

☑ Stratified

○ Cross validation by feature

○ Random sampling

Repeat train/test: 20

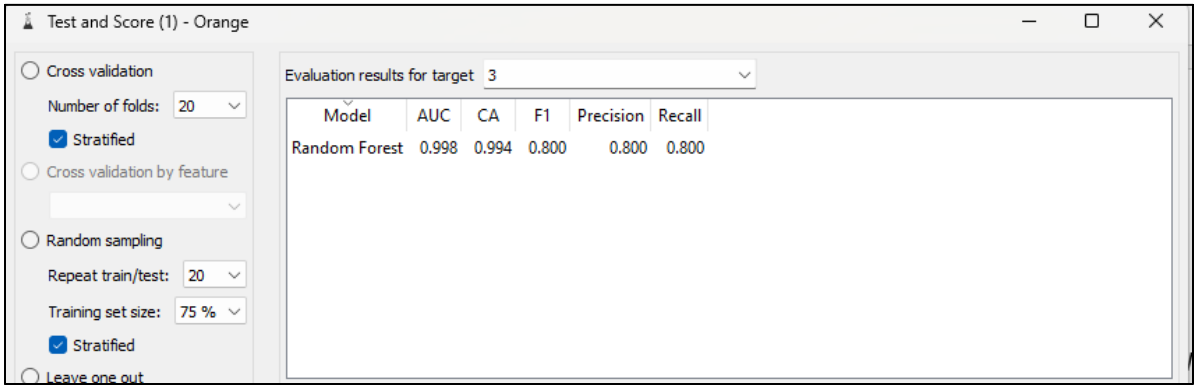
Training set size: 75 %

☑ Stratified

Evaluation results for target 2

Model	AUC	CA	F1	Precision	Recall
Random Forest	0.999	0.989	0.973	0.973	0.973

Image 5: Evaluation Results for “Good” or “2”.



Test and Score (1) - Orange

○ Cross validation

Number of folds: 20

☑ Stratified

○ Cross validation by feature

○ Random sampling

Repeat train/test: 20

Training set size: 75 %

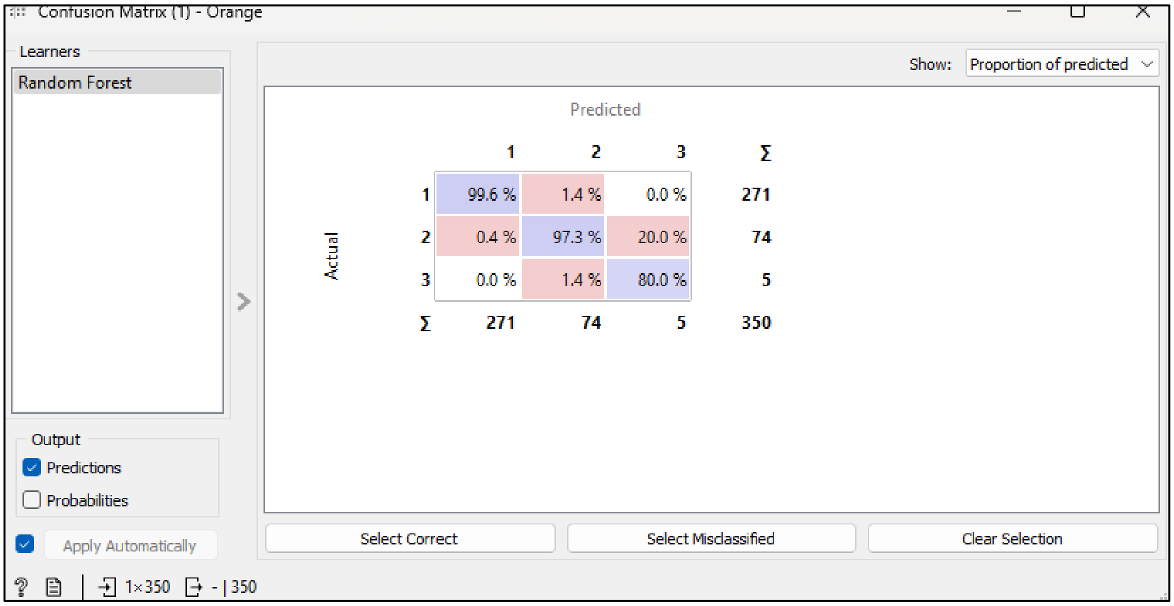
☑ Stratified

○ Leave one out

Evaluation results for target 3

Model	AUC	CA	F1	Precision	Recall
Random Forest	0.998	0.994	0.800	0.800	0.800

Image 6: Confusion matrix for random forest model.



Confusion Matrix (1) - Orange

Learners

Random Forest

Show: Proportion of predicted

Predicted

Actual

	1	2	3	Σ
1	99.6 %	1.4 %	0.0 %	271
2	0.4 %	97.3 %	20.0 %	74
3	0.0 %	1.4 %	80.0 %	5
Σ	271	74	5	350

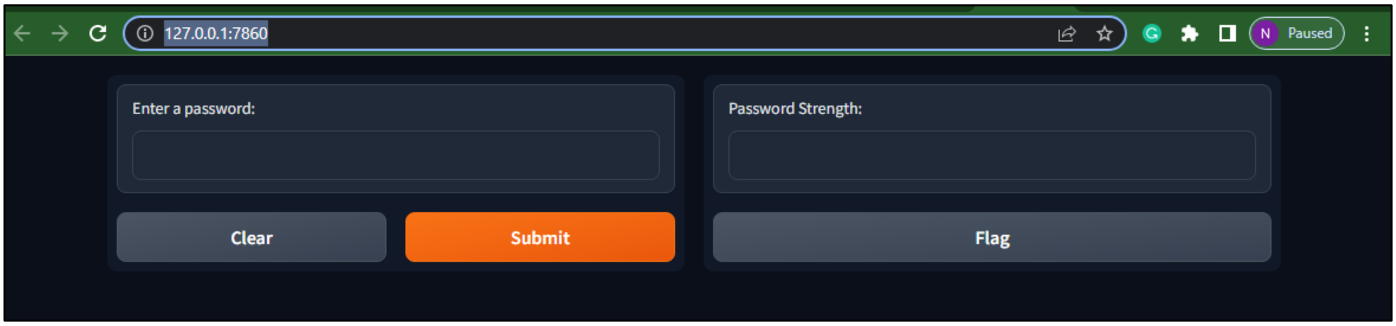
Output

☑ Predictions

☐ Probabilities

☑ Apply Automatically

Image 7: Screenshot of GUI example.


127.0.0.1:7860
Paused
Enter a password:
Password Strength:
Clear
Submit
Flag