April 28, 2023

ML Software Vulnerabilities

Prepared BY: GROUP [TEAM 15]

Harshita Jain

Namrata Gaur

Shreya Reddy Bolla

Tapaswi Reddy Busireddy

Vinita Maloo

Sponsor: Foteini Argiropoulos

Table of Contents

Project Description

Static code analyser tools are used to detect the code smells in a source code. However, they do not assess the vulnerabilities present in the source code. Vinci Tool is a standalone machine learning application that can be used to predict vulnerabilities in a source code. The tool is a machine learning model that is trained based on a static code analyser report(either in XML or CSV format) provided to it. After successfully training the model, the tool makes its predictions of the vulnerabilities and assesses whether an entity in the static code analyser tool is a vulnerability or not. In addition, the tool can also be used to assess the risk(threat level) of software vulnerabilities. The tool helps in finding and avoiding vulnerabilities in software which can help prevent cyber attacks.

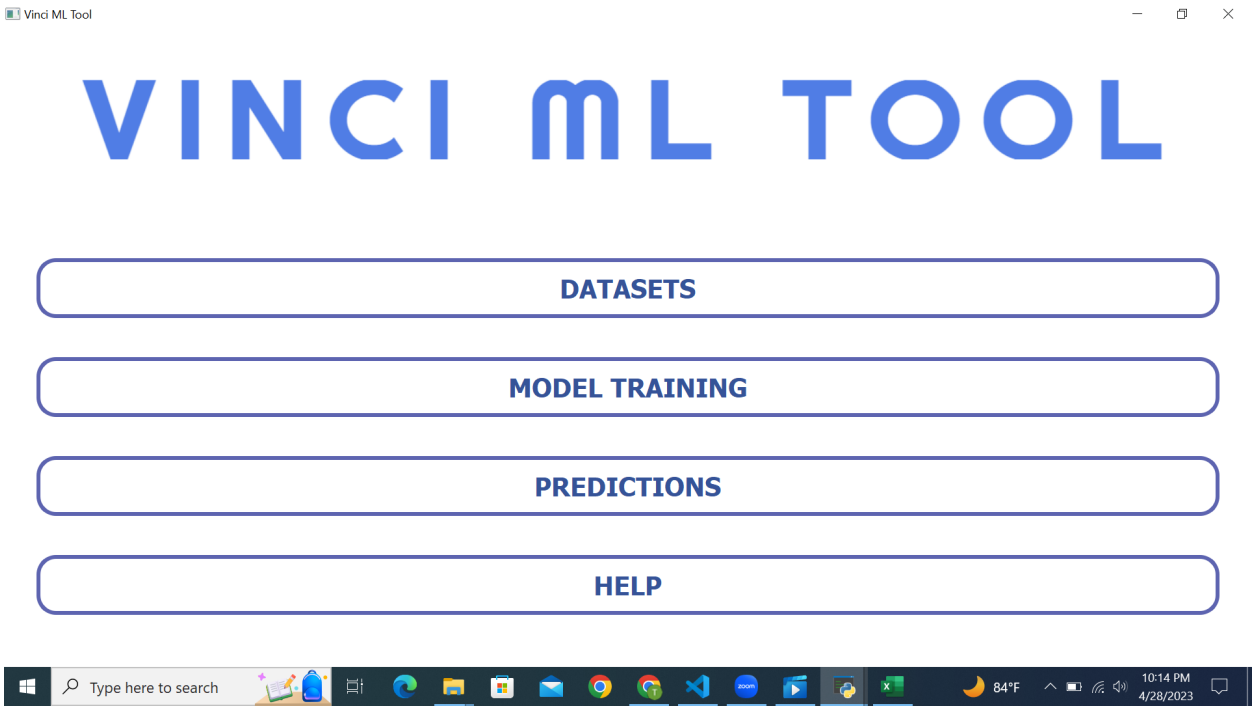
User Manual

The several features that were implemented are:

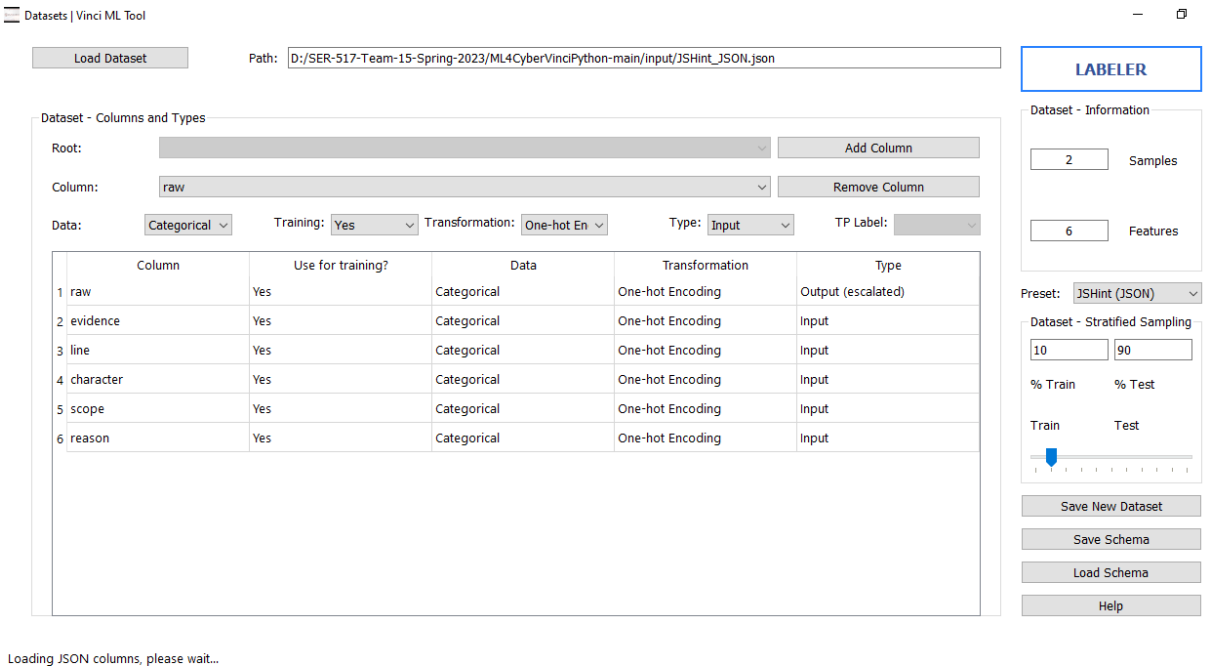
1. **Static code analyser tool support:**
   a. **Description:**

      As users and companies prefer to use varied programming languages, it is essential to provide support to more than one static code analyser tool in the Vinci tool. Therefore, we provided initial support to reports from various static code analysers, such as PMD(for Java), PHP_Codesniffer(for PHP), Pylint(for Python), ESLint, and JSHint(for Javascript). This will help users from several backgrounds to prefer the tool.

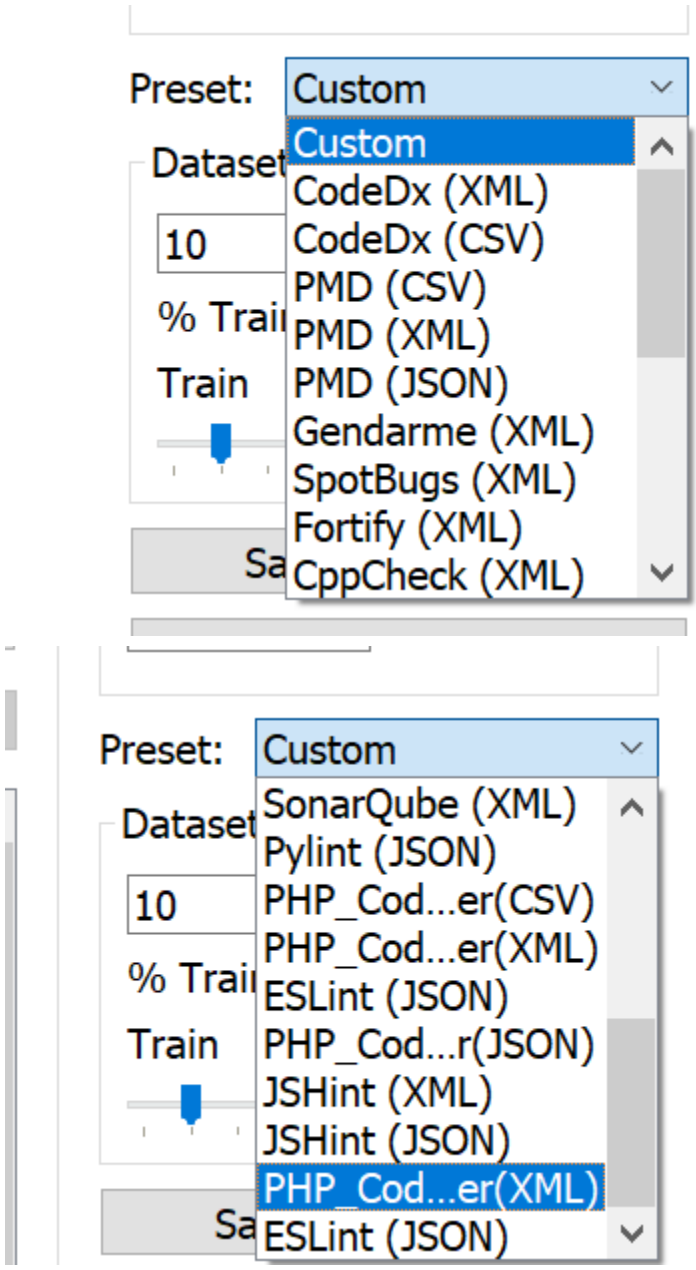
   b. **Feature user guide:**
      i. Run the tool.
      ii. On the landing page, choose the ‘Datasets’ button.



iii. In the ‘Datasets’ window, select the ‘Preset’ option on the right side.



iv. The ‘Preset’ dropdown shows the various static code analyser tools that are supported by the Vinci tool.



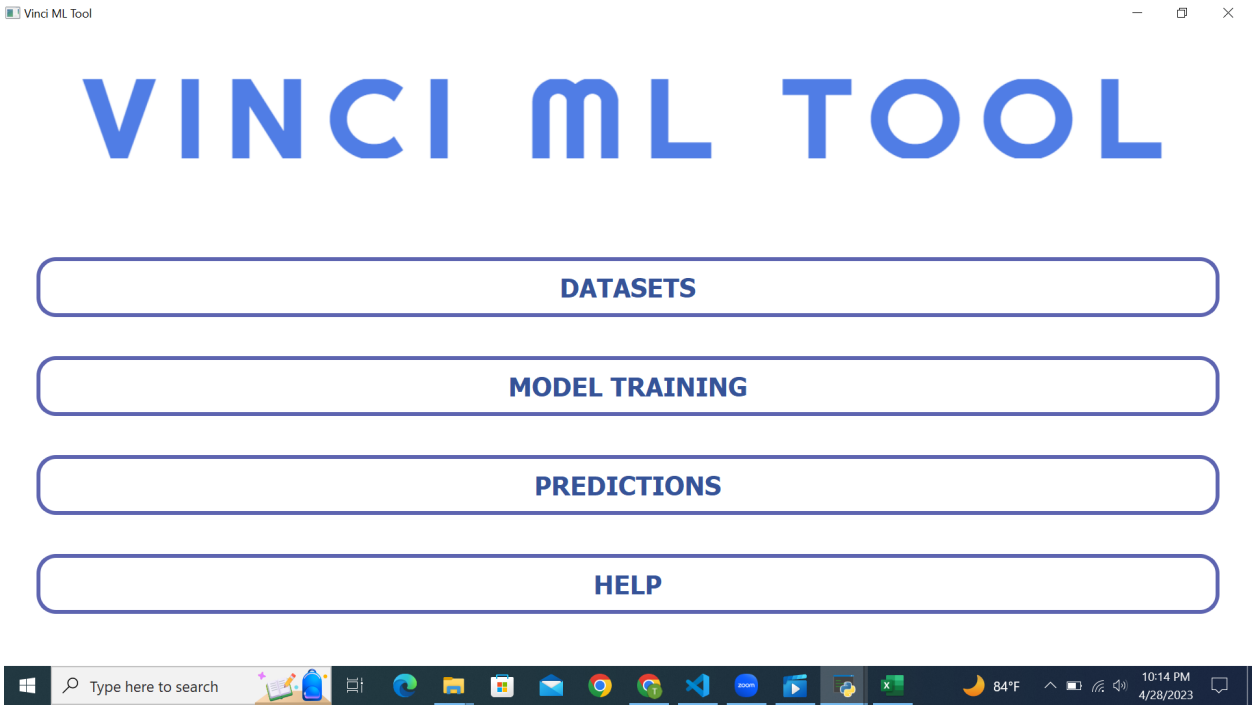
v. In the ‘Datasets’ window, click on the ‘Load Dataset’ button to load a dataset of your choice.

c. **Sponsor requirement:** Yes

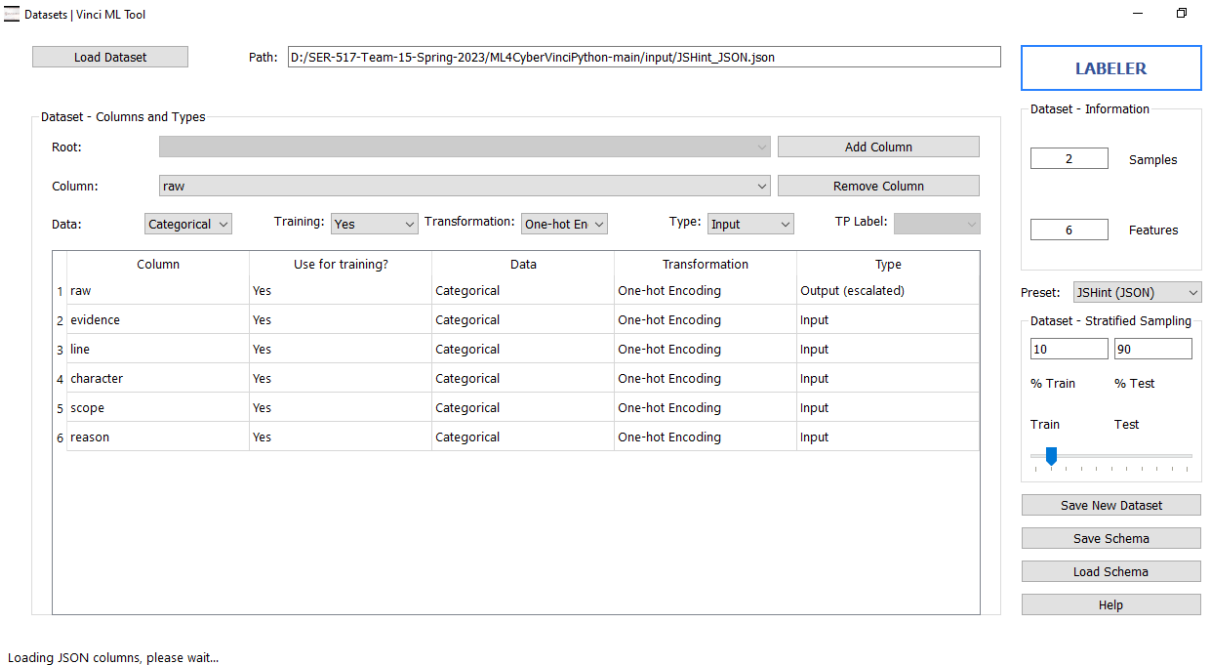
2. **Different file formats support:**
    a. **Description:**

        Initially the tool only supported the XML, and CSV file formats for the report. So, we have added JSON file format as well to load and label the data in the tool. Now, the tool supports XML, and CSV to the full extent with all the datasets and the reports but with JSON we could only load and label the data.

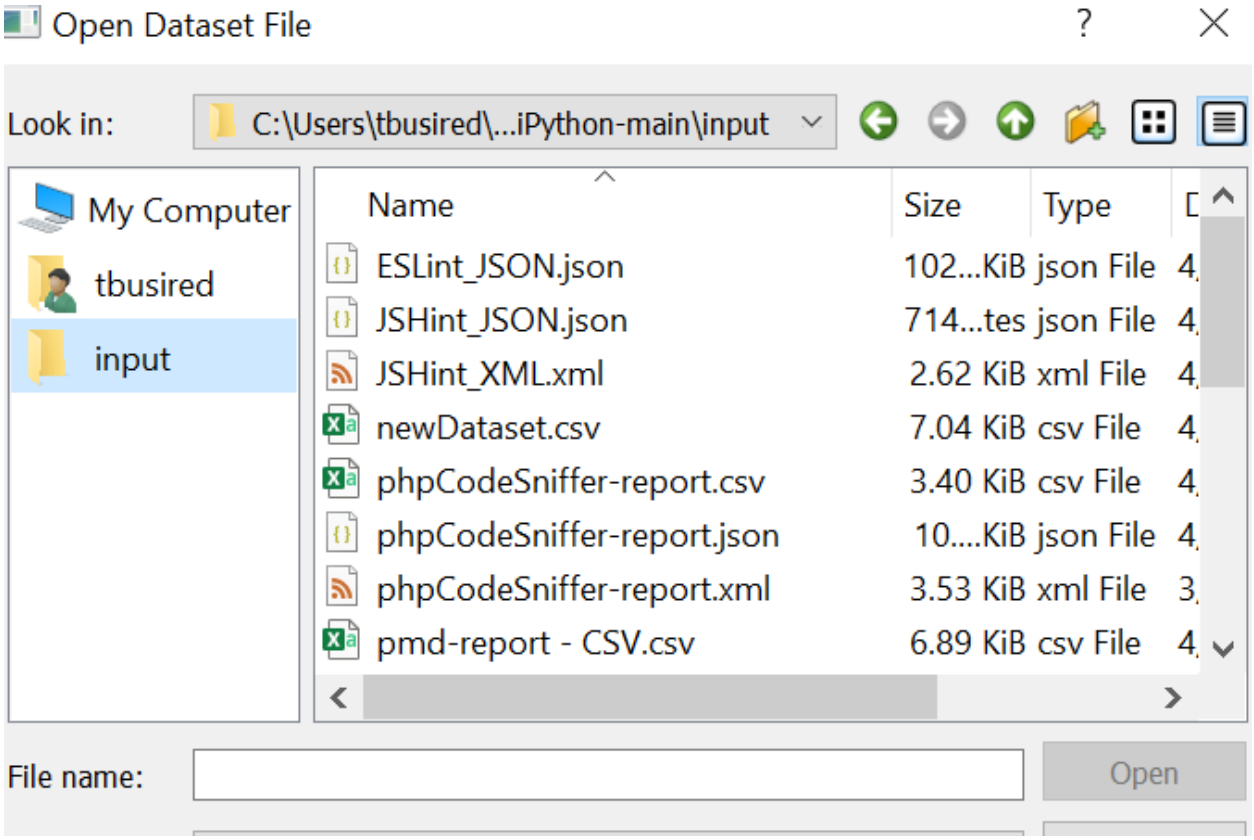
    b. Feature user guide:
        i. Run the tool.
        ii. On the landing page, choose the 'Datasets' button.



iii. In the ‘Datasets’ window, select the ‘Load Dataset’ option on the left side.



iv. The ‘Load Dataset’ options let you select the file from the file formats that are supported by the tool.



c. **Sponsor requirement:** Yes

3. Dynamic risk analysis:

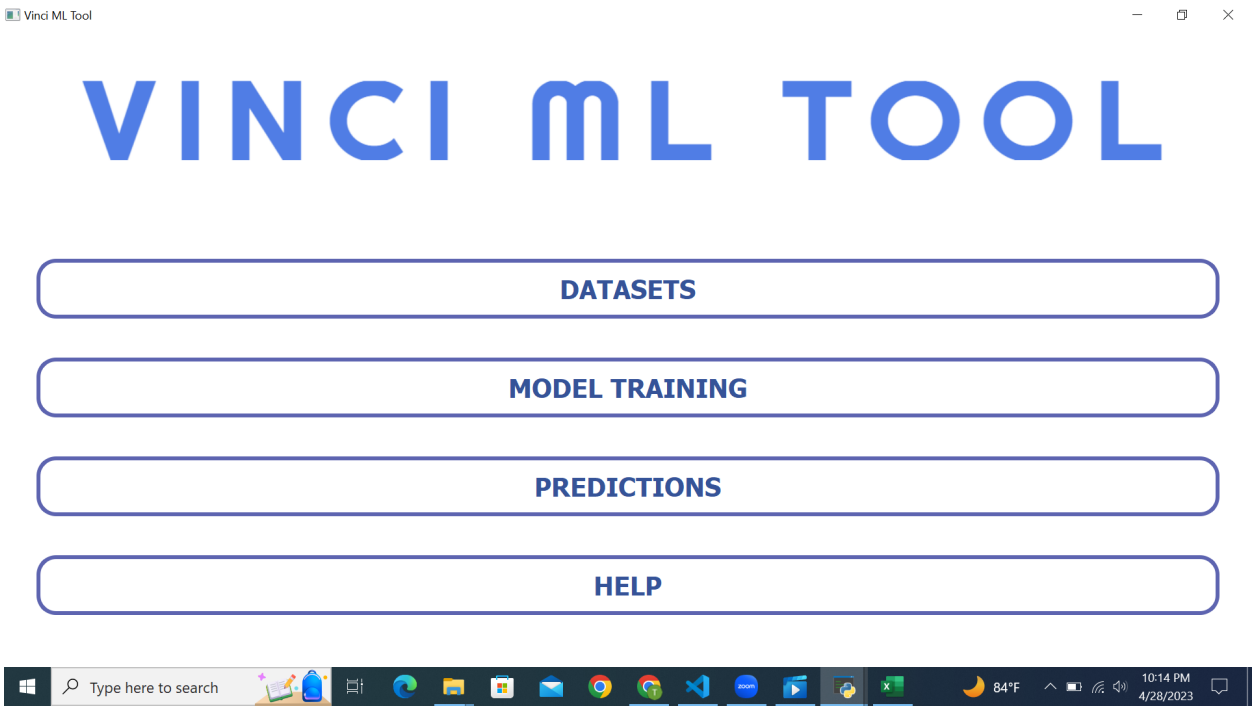
a. Description:

The tool initially had the option to assess the risk of each entity from the input report separately. To facilitate dynamic risk analysis, a logistic regression model was used. The user can label a few of the entities and save the results which will be considered as the training set for the model. A JSON-formatted schema is generated based on the training data which is then fed to the model. Then, a model is generated based on the schema and training data. Finally, the saved dataset, the schema, and the model are used to make predictions and the risk level results are saved in a CSV file.

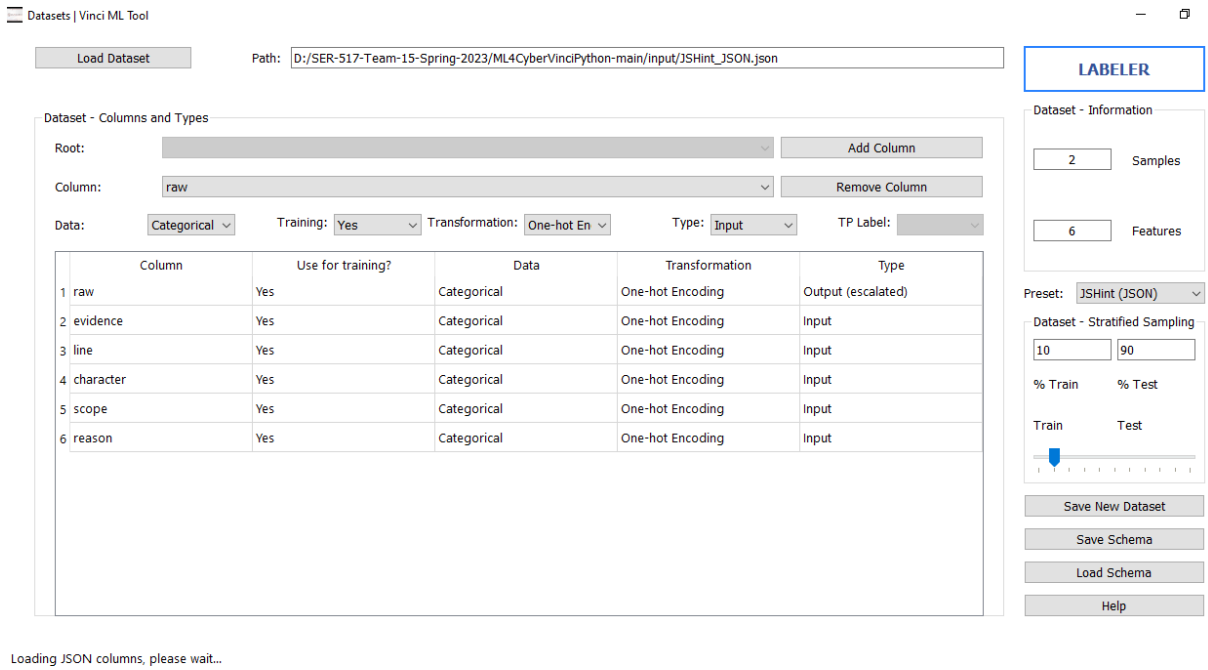
b. Feature user guide:

i. Run the tool.

ii. On the landing page, choose the ‘Datasets’ button.

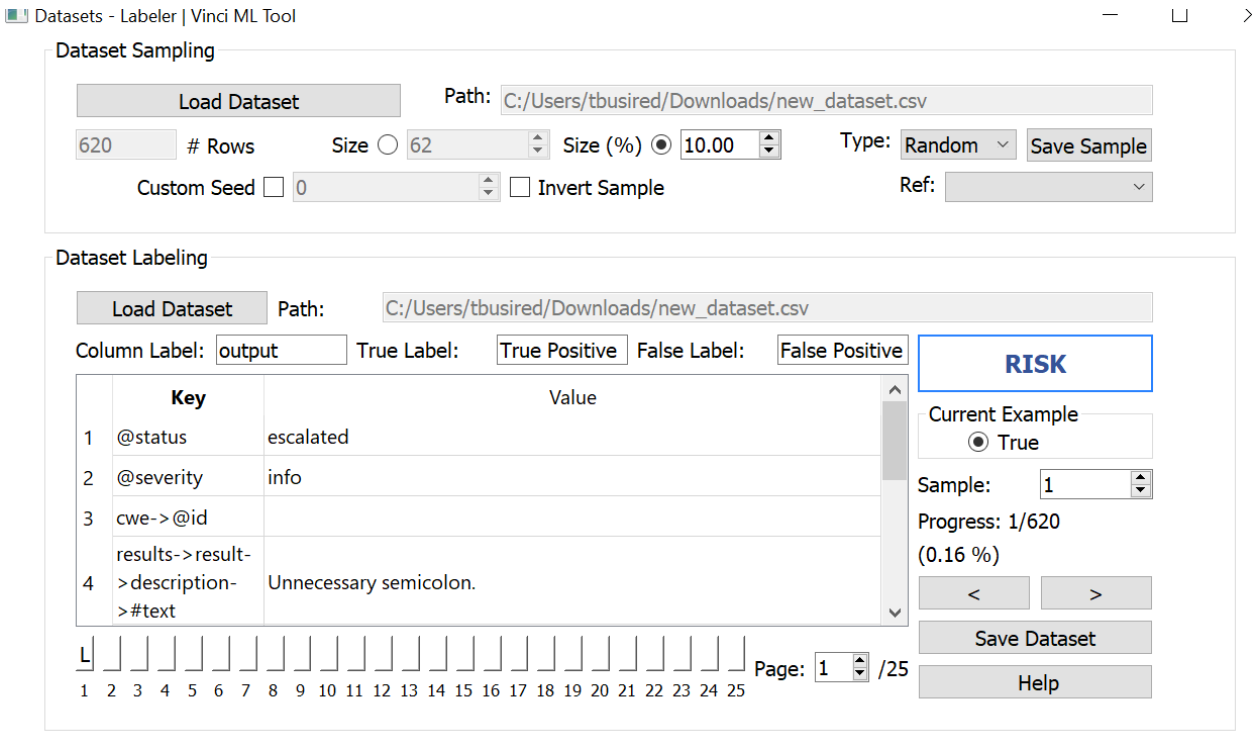


iii. In the 'Datasets' window, click on the 'Load Datasets' button and load a CodeDx CSV dataset.

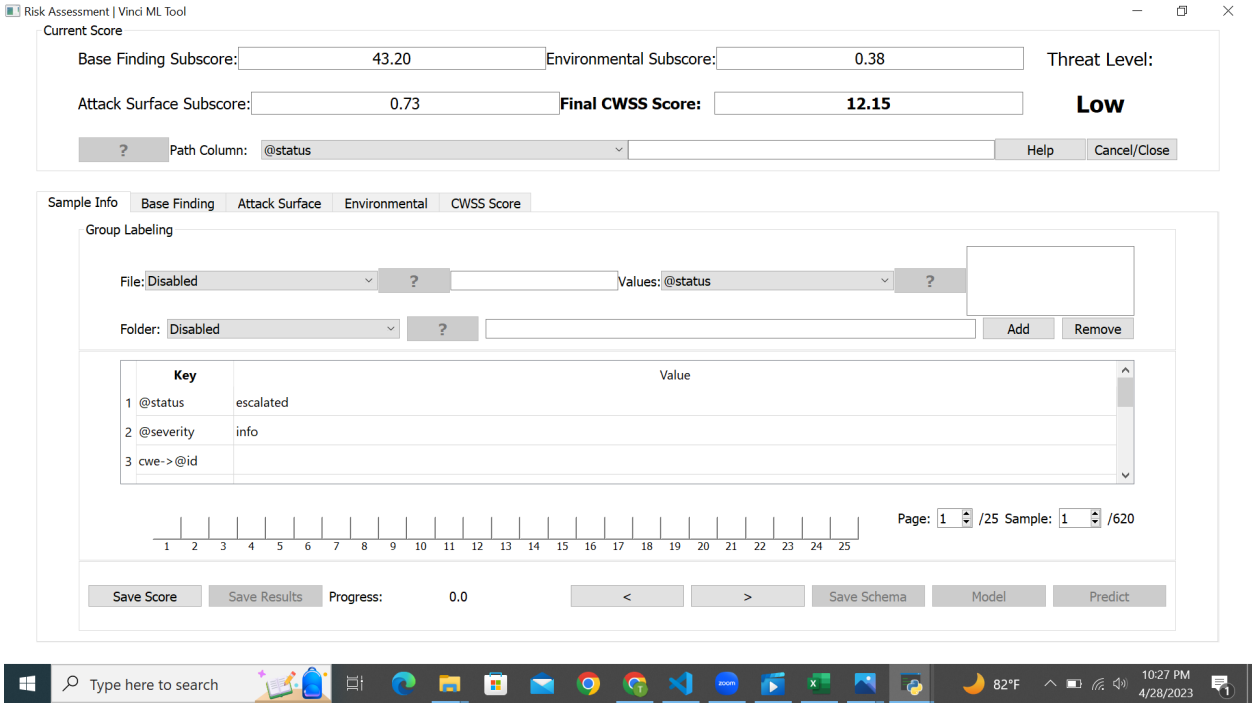


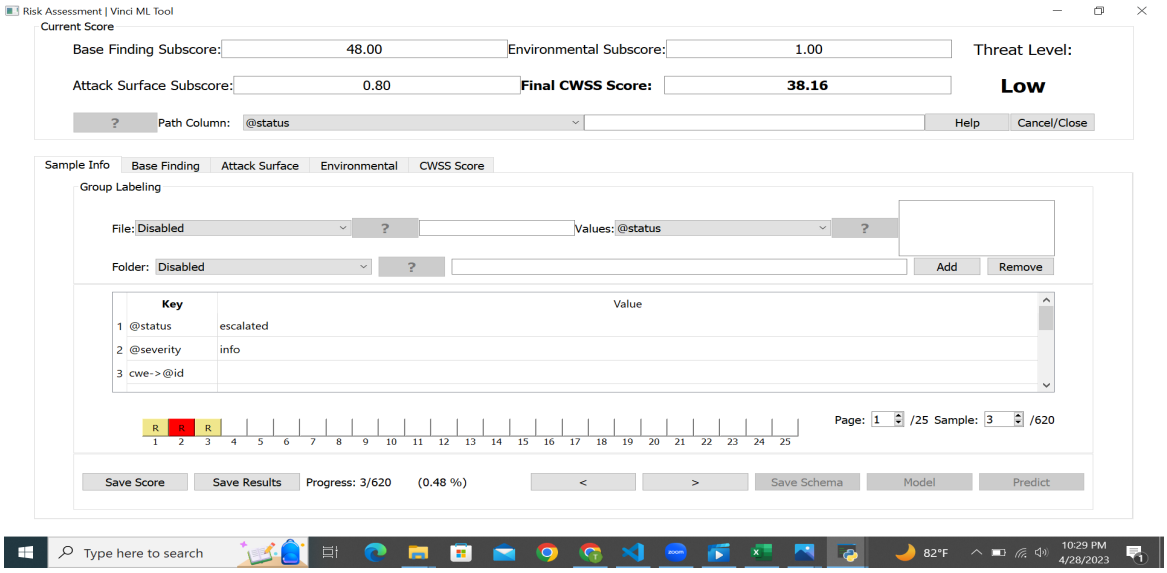
iv. Click on the ‘Labeler’ button.

v. In the ‘Labeler’ window, click on the ‘Risk’ button.

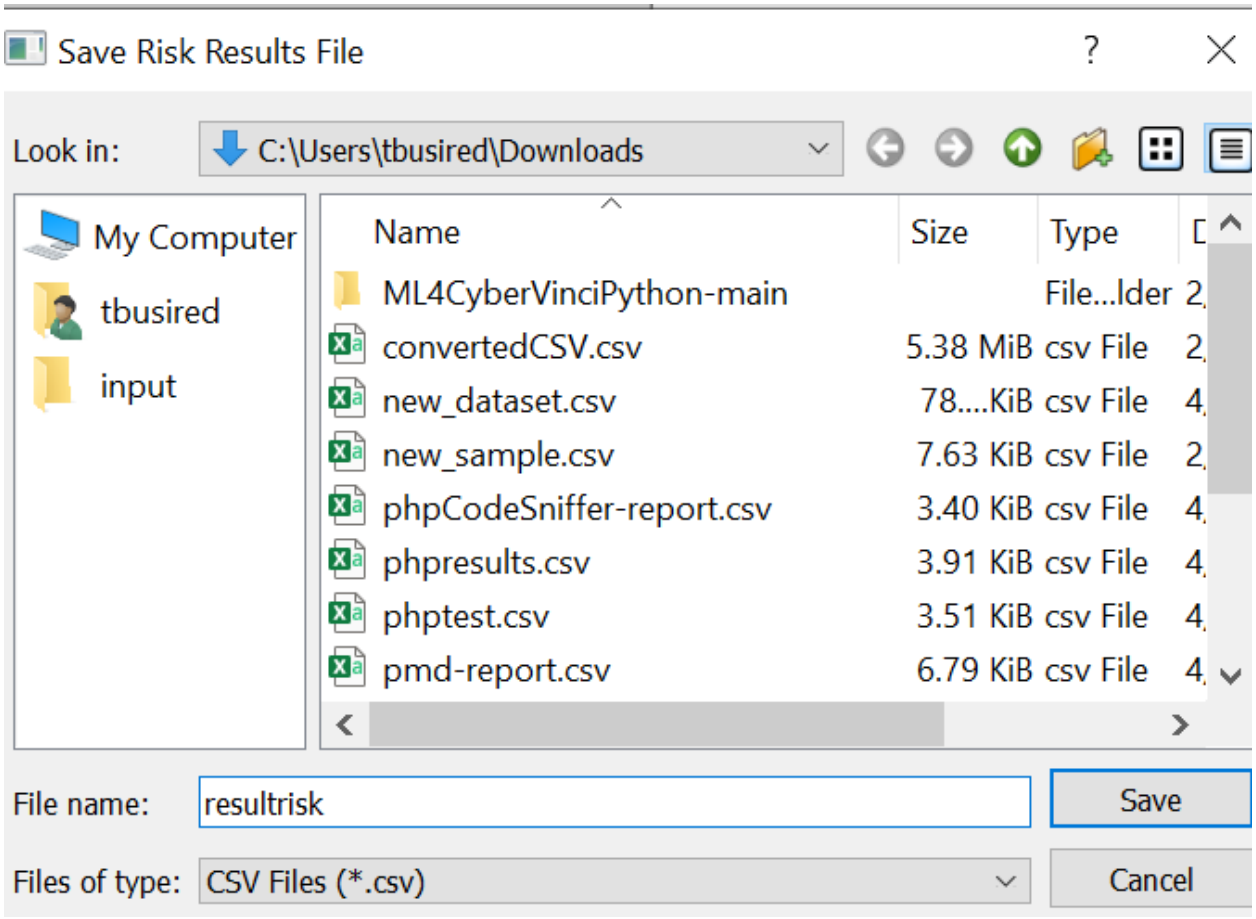


vi. In the ‘Risk’ window, change the values of the scores as necessary and click on ‘Save Scores’ to save the threat level for that entity.

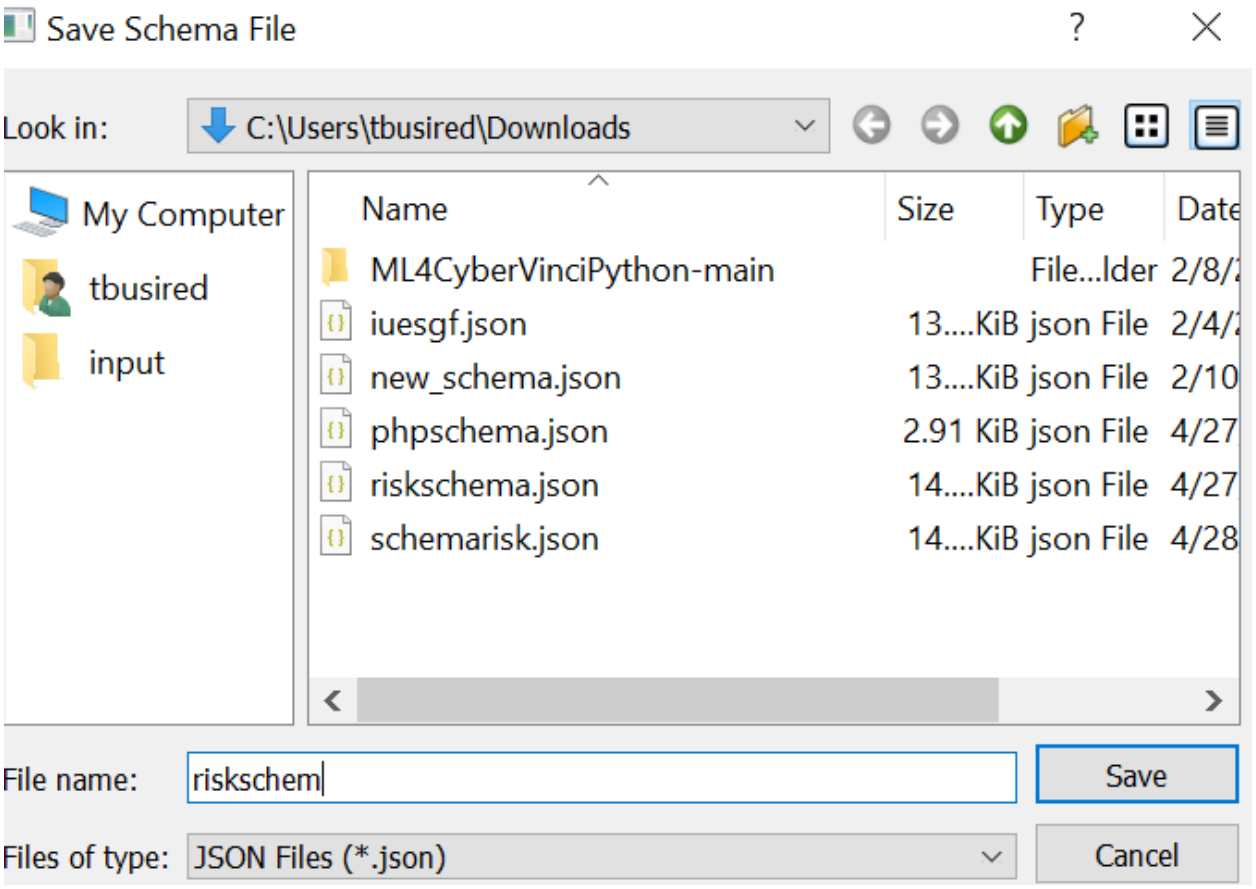




vii. After sufficient entries have been labeled, click on the 'Save Results' button and save the CSV file.

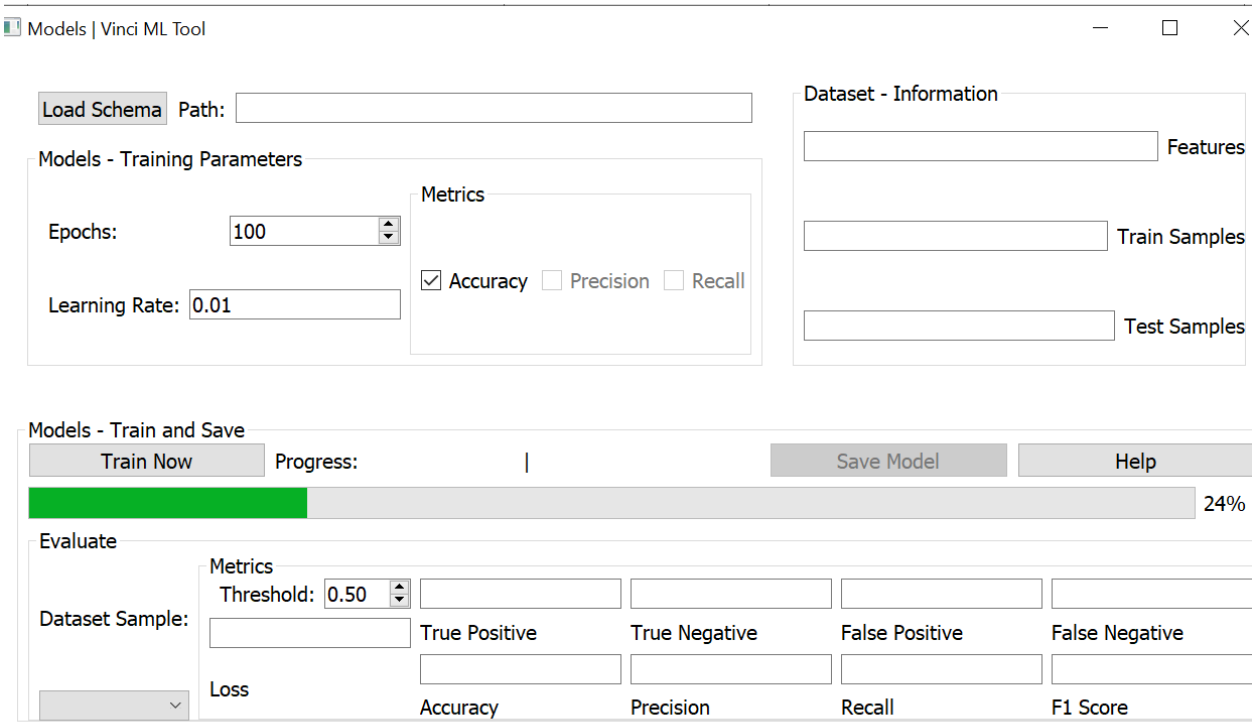


viii. Click on ‘Save Schema’ and save the JSON formatted schema file.

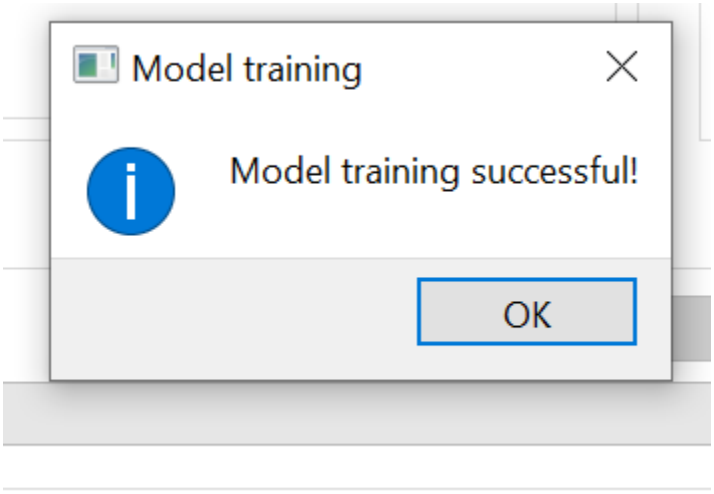


ix. Now, click on the ‘Model’ button.

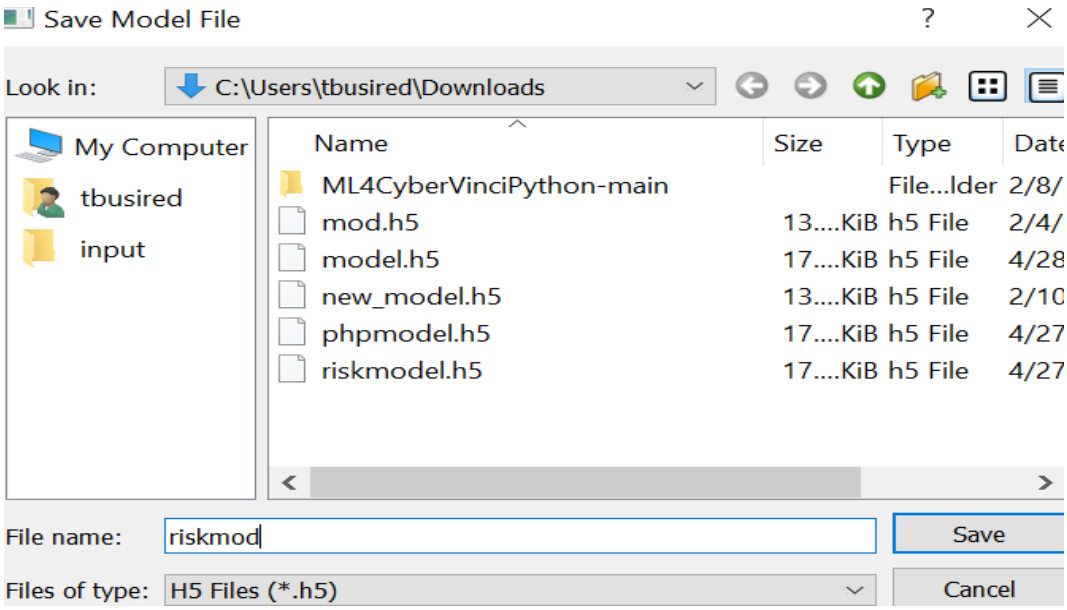
x. In the ‘Model’ window, upload the previously saved schema file and change the epochs, learning rate, and threshold as necessary.



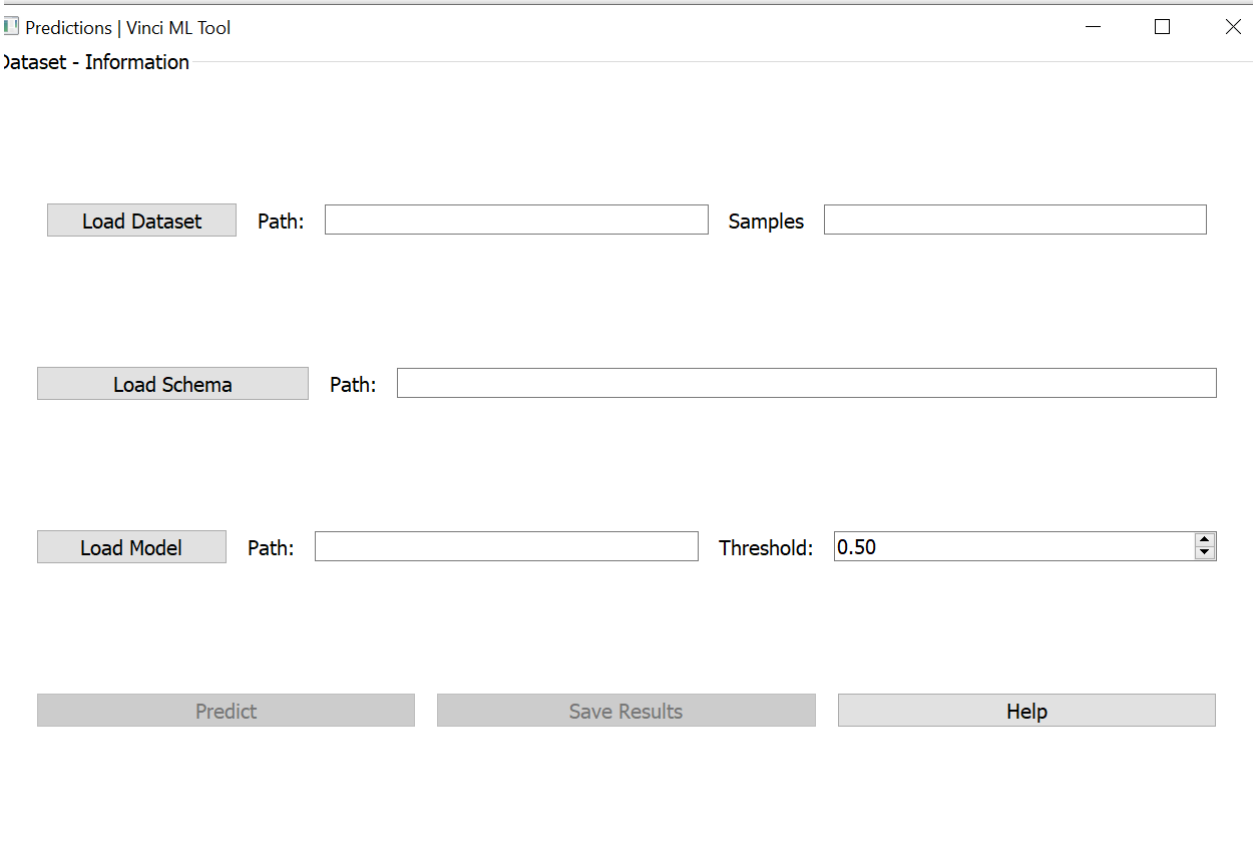
xi. Click on ‘Train Now’ and wait for the model to be trained.



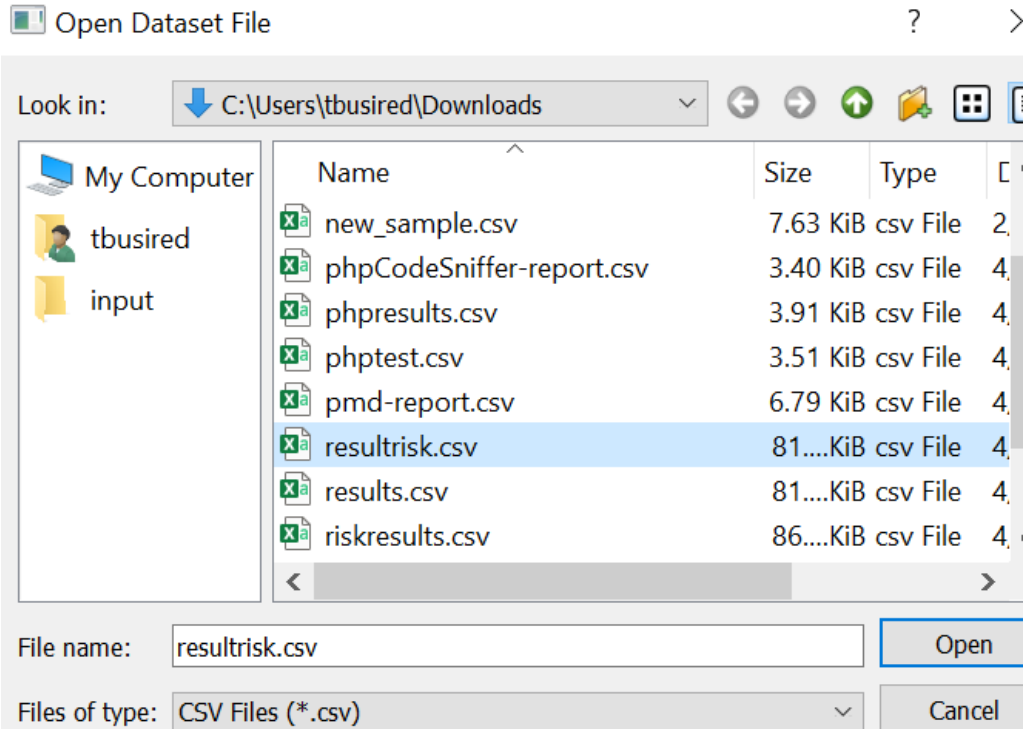
xii. When the training is finished, click on ‘Save Model’ and save the model.



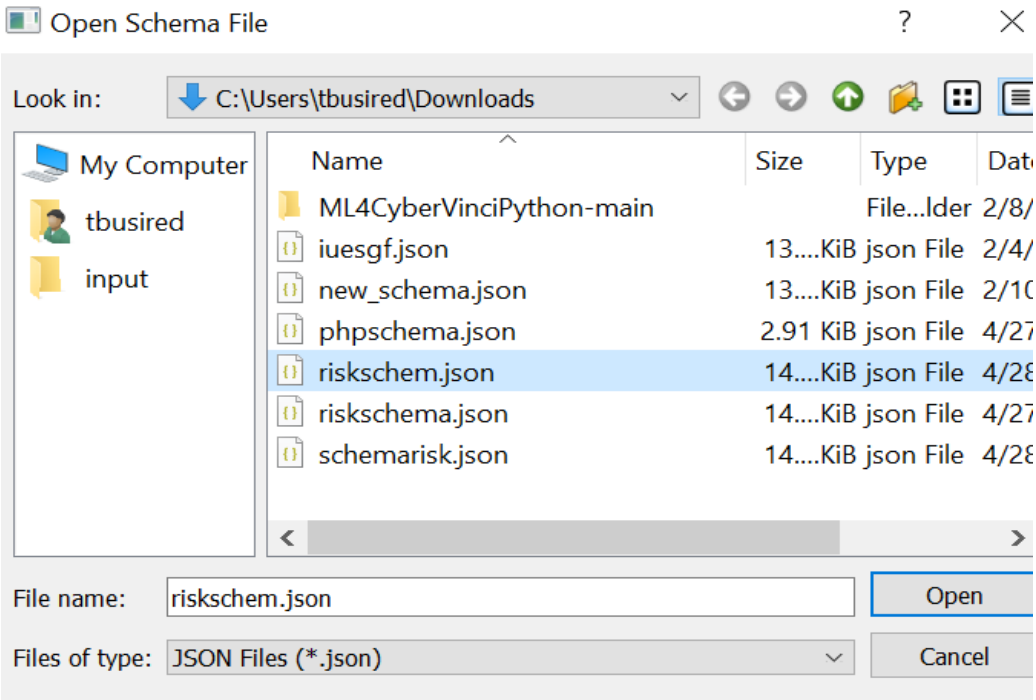
xiii. Now click on the ‘Predict’ button in the ‘Risk’ window.

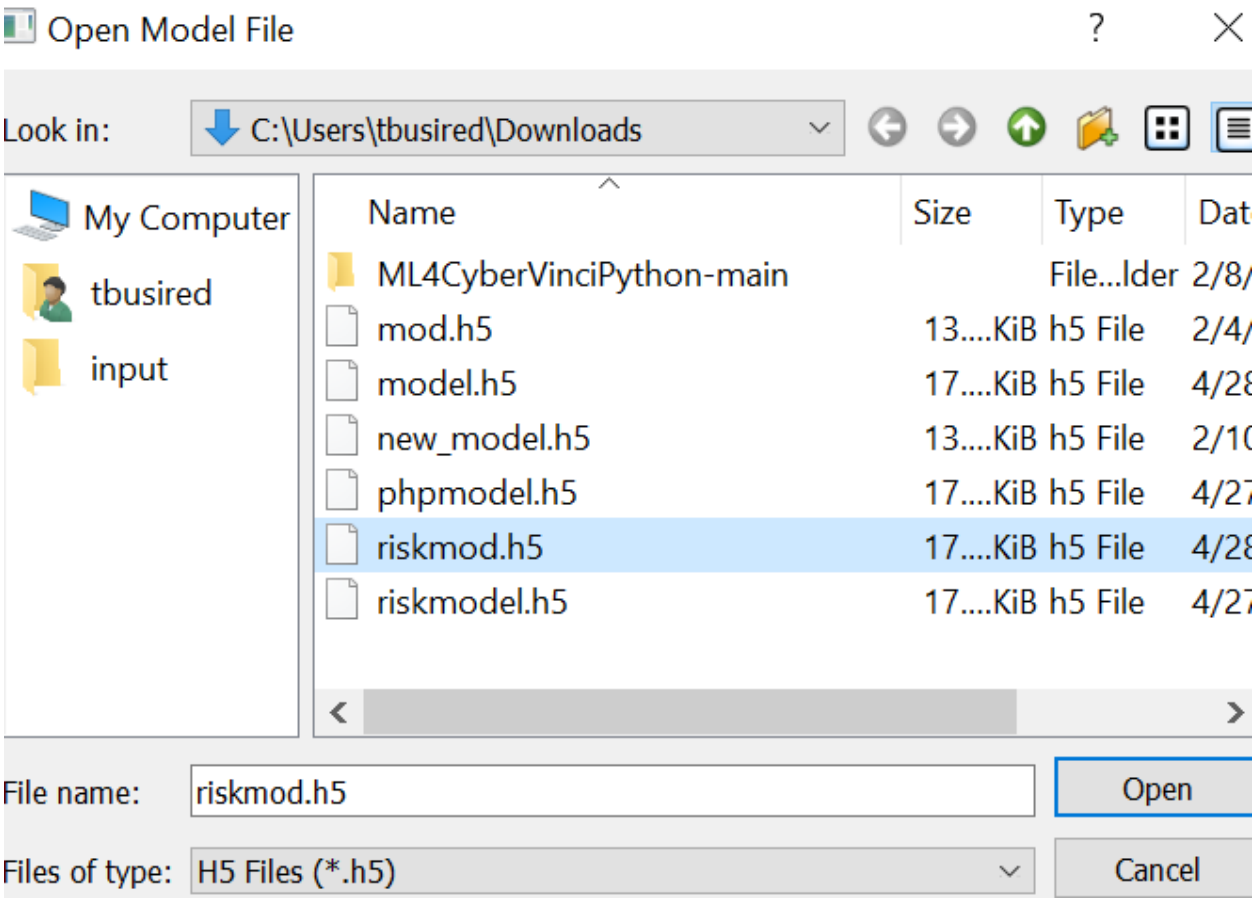


xiv. In the 'Prediction' window, click on 'Load Dataset' and load the saved dataset from step vii.

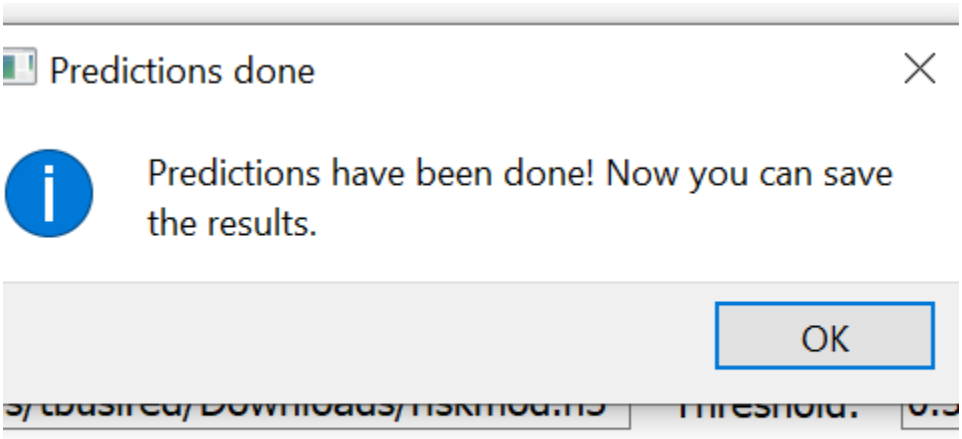


xv. Click on 'Load Schema' and load the schema from step viii and click on 'Load Model' and load the model from step xii.

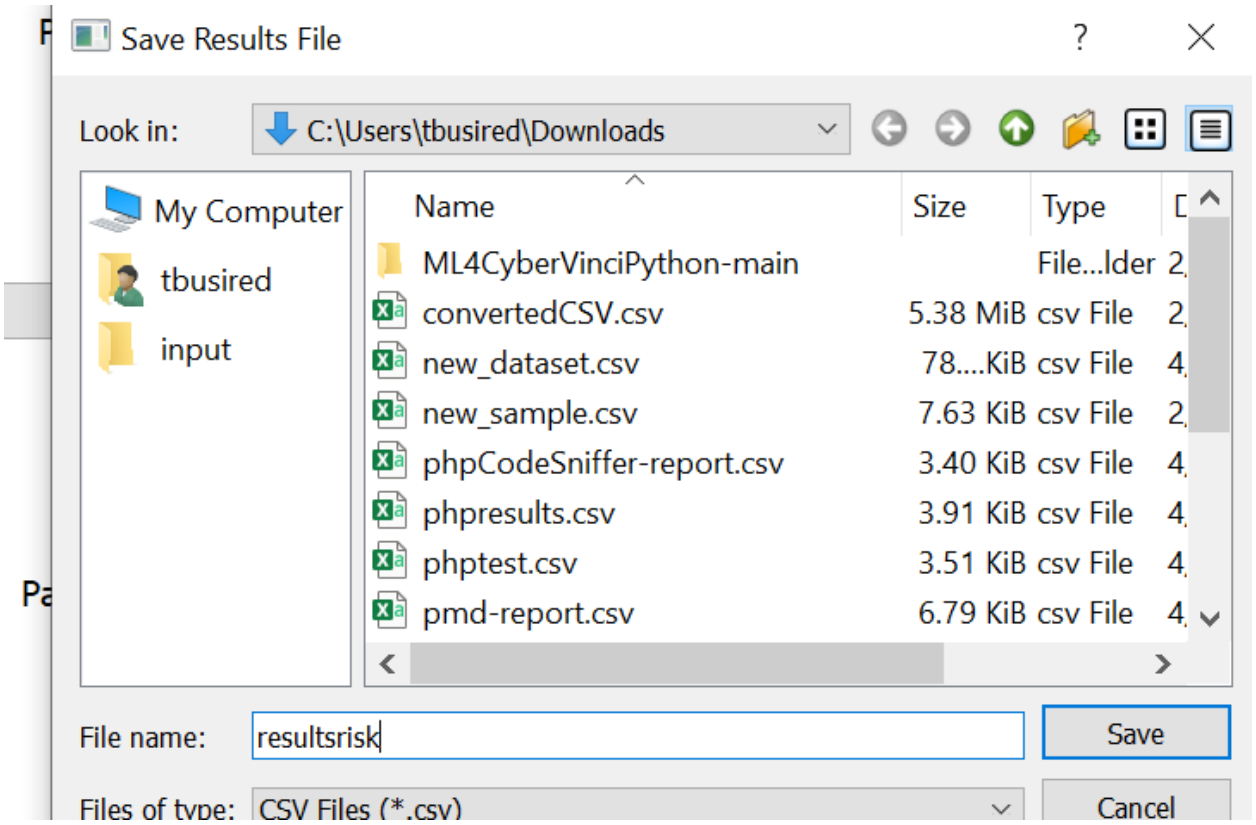




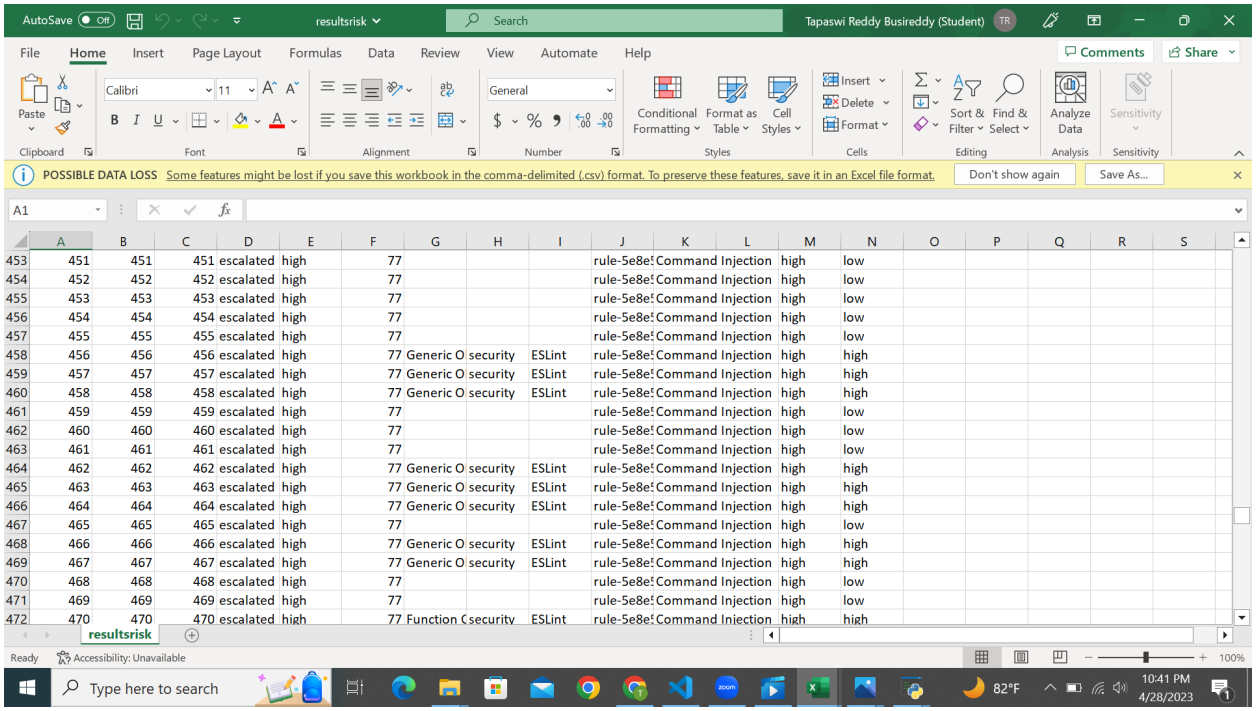
xvi. Click on 'Predict' and wait for predictions to be done.



xvii. Click on ‘Save Results’ to save the results.



xviii. A new column with the predictions will be visible in the saved CSV file.

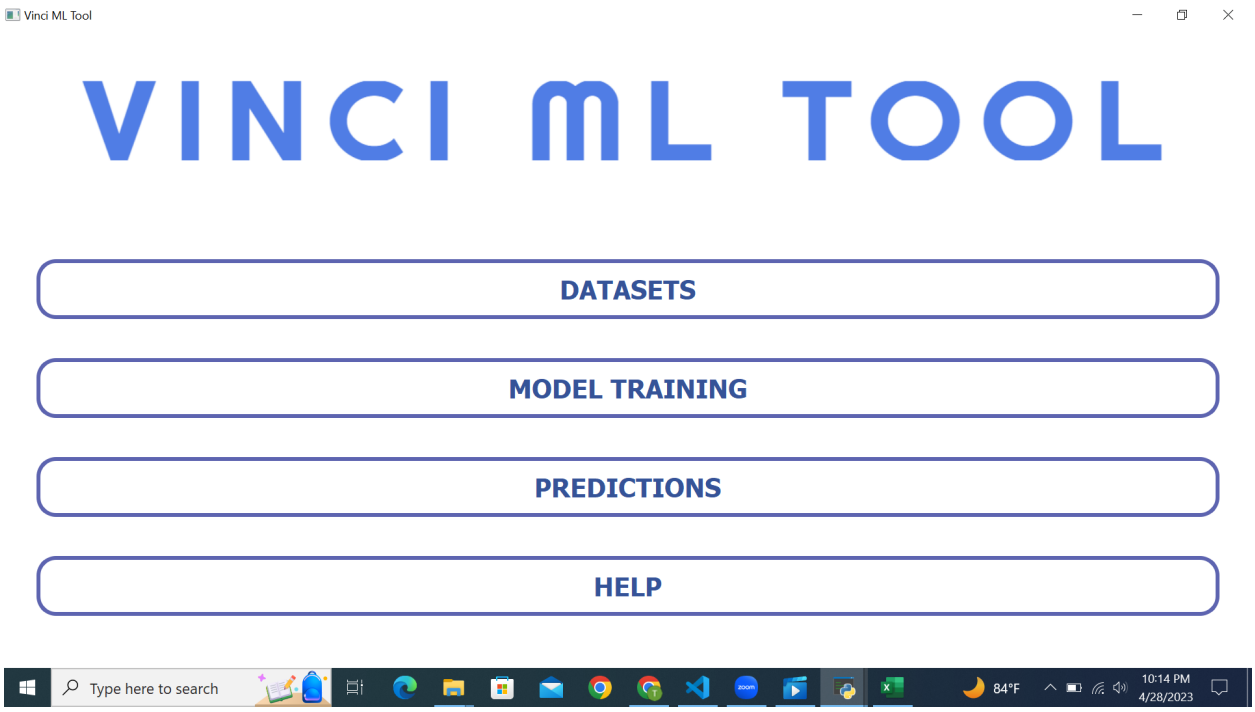


c. **Sponsor requirement:** Yes

4. **Dynamic window resizing:**
   a. **Description:**

      Initially the windows were very fixed and the components in the window were very congested and wouldn't resize while maximizing. So, we added a dynamic resizing to the window which expands whenever the user clicks on the maximized window. In, this way the components look larger for the windows when maximize and fit right according to the laptop display screen.

   b. **Feature user guide:**
      i. Run the tool.
      ii. Click on maximize to view the window size



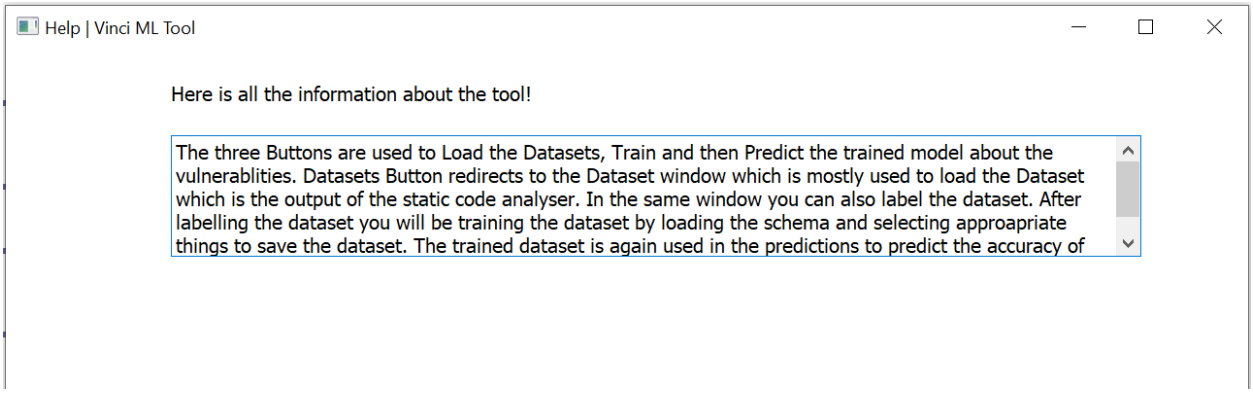
      iii. Click on the 'Datasets' options and click on maximize to view the window size with dynamically resized components.
      iv. Click on the 'Labeler' option and click on maximize to view the window size with dynamically resized components.
      v. Click on the 'Risk' option and click on maximize to view the window size with dynamically resized components.
      vi. Click on the 'Model Training' option and click on maximize to view the window size with dynamically resized components.
      vii. Click on the 'Predictions' option and click on maximize to view the window size with dynamically resized components.
      viii. Click on the ' Help' option and click on maximize to view the window size with dynamically resized components.

c. **Sponsor requirement:** Yes

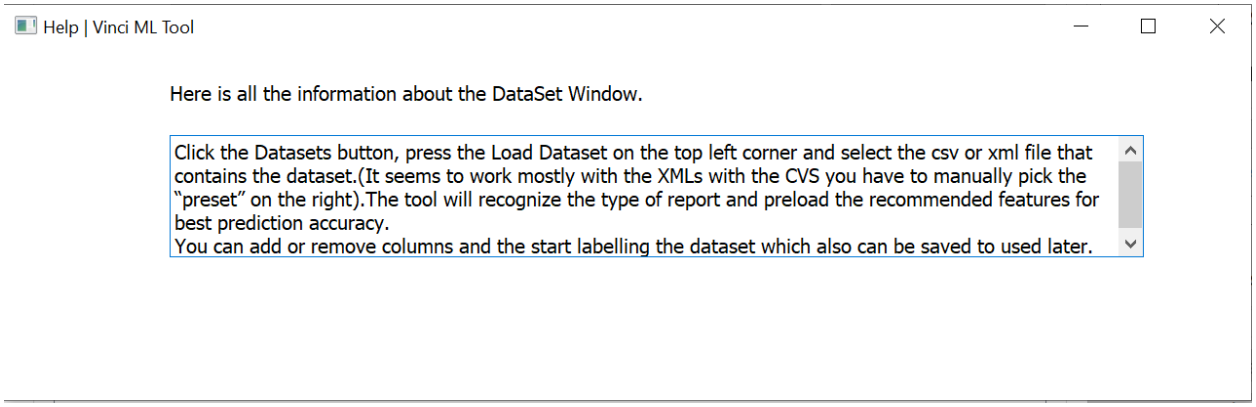
5. **Help option for each screen:**
   a. **Description:**

      There is a lot of information in the tool which a user may or may not understand the fields and what they do. So, we decided on adding the help screen in every window which helps the user to identify the fields and work on the datasets in the tool. The help gives all the information about the windows and their functionalities.

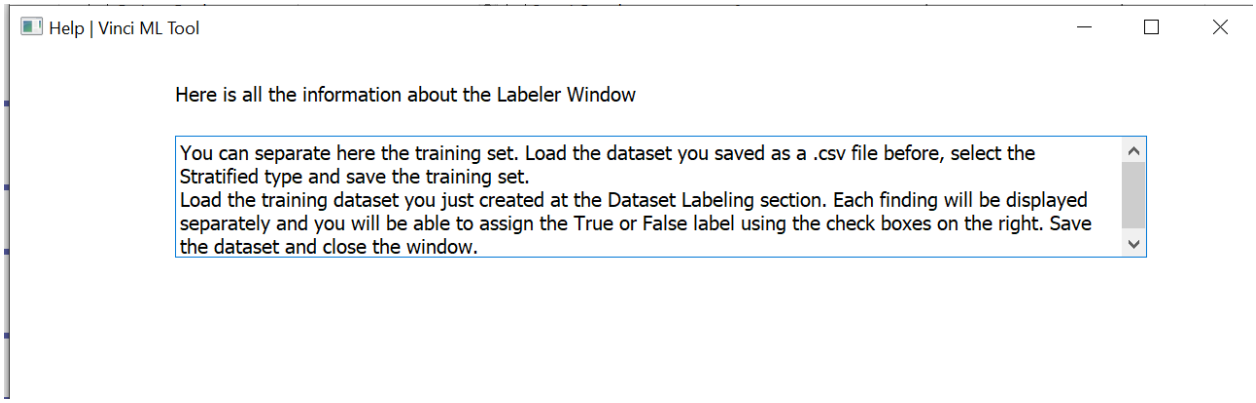
   b. **Feature user guide:**
      i. Run the tool
      ii. Click on 'Help' to know about each of the features displayed



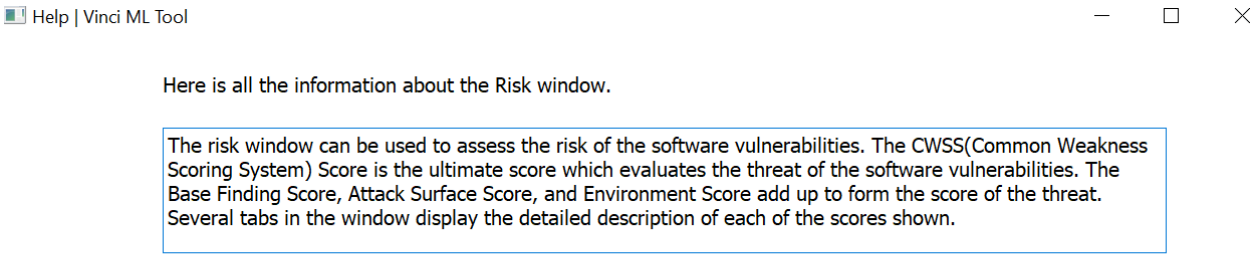
      iii. Click on 'Datasets' and click on 'Help' to know more about the 'Datasets' window and the functionalities involved.



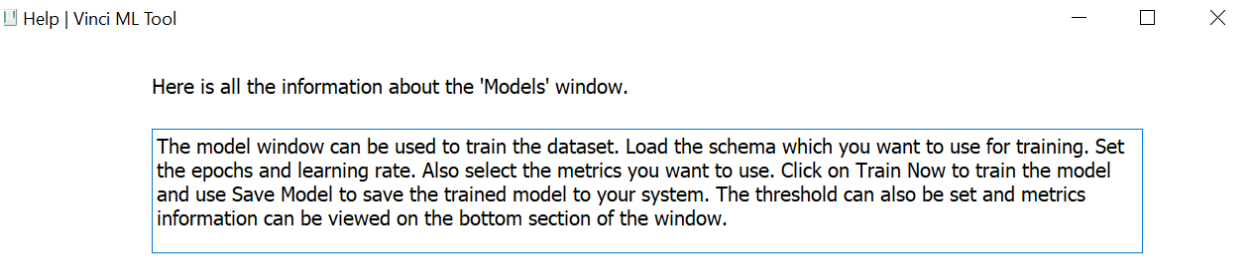
iv. Click on ‘Labeler’ and click on ‘Help’ to know more about the ‘Labeler’ window and the functionalities involved.



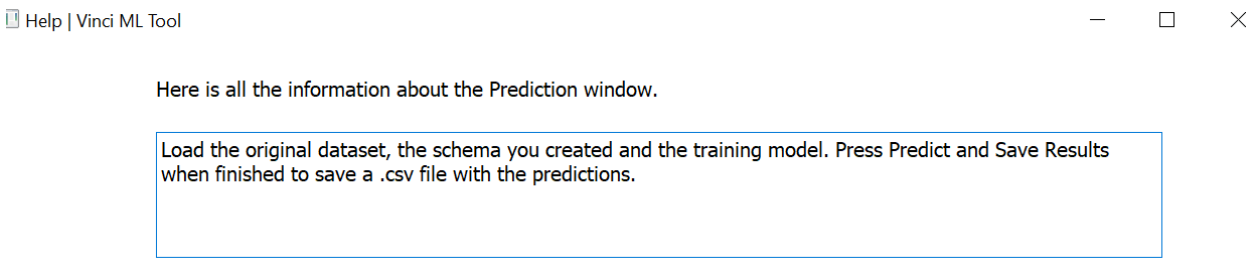
v. Click on ‘Risk’ and click on ‘Help’ to know more about the ‘Risk’ window and the functionalities involved.



vi. Click on ‘Model Training’ and click on ‘Help’ to know more about the ‘Model Training’ window and the functionalities involved.

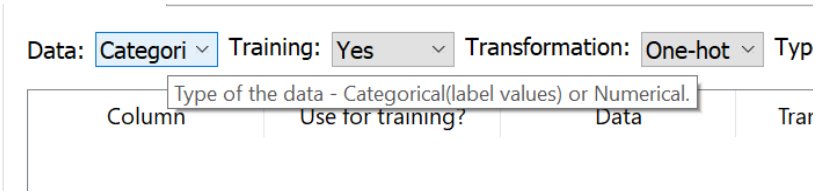


vii. Click on ‘Predictions’ and click on ‘Help’ to know more about the ‘Predictions’ window and the functionalities involved.

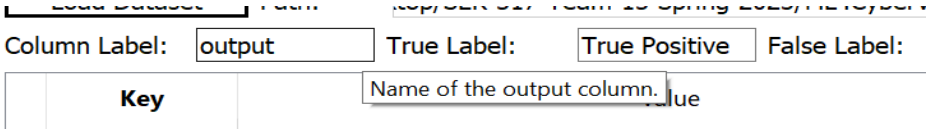


c. **Sponsor requirement:** Yes

6. **Tooltips for all screens:**
   a. **Description:**
      i. Tooltips are added to every label on all the available screens. This will help the user get an understanding of the functionality of that particular widget or textbox and helps enhance the overall UI of the tool.
   b. **Feature user guide:**
      i. Run the tool.
      ii. Click on the ‘Datasets’ window.
      iii. Hover over any of the widgets or textboxes to display a tooltip.



      iv. Click on the ‘Labeler’ button.
      v. Hover over a textbox in the window to display a tooltip.



      vi. Navigate to the ‘Risk’ window and hover over a textbox to display a tooltip.
      vii. Navigate to the ‘Model Training’ window and hover a textbox to display a tooltip.
      viii. Navigate to the ‘Prediction’ window and hover a textbox to display a tooltip.
   c. **Sponsor requirement:** Yes

Installation Guide

The following steps are the guidelines to successfully install the project into the system and to create an executable for the same.

- Installation:
  - Make sure you are using Python 3.8
  - To install dependency packages, issue the following command from your project folder:

    $ pip install -r requirements.txt

  - After this, you should be able to launch the App with the following command:

    $ python gui/app.py

- Creating the executable file:
  - To generate an executable file, run the following command:

    $ cd gui

    $ pyinstaller --clean --onefile --windowed --icon=app.ico app.py

  - To save executable to another folder, use --distpath:

    $ pyinstaller --clean --onefile --windowed --icon=app.ico --distpath=distUbuntu/ app.py

Issues

There are no major pending issues in the tool. A few of the minor issues that could be improved in the tool are:

- The prediction made is less accurate so some of the Risk predictions made by the tool are wrong.
  - Possible Solution: This could be improved by using a different type of logistic regression modeling algorithm for risk prediction.
- The Risk Prediction can only be performed after the Labeler screen is visible. The Risk feature can be migrated to the Landing screen so that the user can perform the prediction without going to the Labeler screen.
- Some of the screens are redundant like modeling and prediction of Risk as well as vulnerabilities. This redundancy could be removed.
- Some of the UI screens follow different patterns than others. In the future, all the screens can be made in such a way that they follow the same pattern.

Outlook

The tool can be further enhanced in the following ways:

- **Static code analyser tool support:**

  Although the tool supports several static code analyser tools and their reports(such as PMD, PHP_Codesniffer, CodeDx, etc.), the support of a static code analyser tool for each

programming language would facilitate developers to choose Vinci tool to find the vulnerabilities in their software. Providing support for tools, such as Understand(for COBOL, FORTRAN, and PASCAL) will increase the reachability and usability of the tool.

- **Enhanced risk model:**

  The current model for predicting the risk of software vulnerabilities has an accuracy of over 85%. However, enhancing the model will help increase the accuracy of the predictions and in addition, the several parameters used to calculate the risk can also be used to make the predictions as opposed to the use of only threat level in the current model.

- **Data transformation support:**

  Currently the tool only supports the transformation of data from categorical to numerical using One Hot Encoding. Integration of the use of several other data transformation techniques will help in working with other types of data that are not compatible with the One Hot Encoding technique.

- **Remove redundant elements from GUI:**

  The tool currently has a large number of screens and buttons which are quite redundant and unnecessary. Removing these redundant elements and screens will make the navigation seamless for the users and also improve the user experience. Screens currently do not flow sequentially. To access the screen for each functionality, the user needs to go back to the landing page and redirect to other screens from there on. Instead, the screens should follow a sequential order so that it is easy for first-time users to use the tool.

Video Presentation

- Youtube Link: SER-517-Team-15-Final-Deliverable