#### Designing and building smart privacy assistants for improving data dashboards

Bachelor Thesis Project-II report submitted to Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of Bachelor of Technology

in

Chemical Engineering

###### by

###### Saamira Yasmin

**(19CH10039)**

###### Under the supervision of Professor Mainack Mondal



**Department of Computer Science and Engineering Indian Institute of Technology Kharagpur**

###### Spring Semester, 2022-23

**May 01, 2023**

#### DECLARATION

I certify that

1. The work contained in this report has been done by me under the guidance of my supervisor.
2. The work has not been submitted to any other Institute for any degree or diploma.
3. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
4. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: May 01, 2023 (Saamira Yasmin)

Place: Kharagpur (19CH10039)

i

#### [DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING](http://www.iitkgp.ac.in/department/CE) [INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR](http://www.iitkgp.ac.in/) KHARAGPUR - 721302, INDIA



##### CERTIFICATE

This is to certify that the project report entitled “**Designing and building smart pri- vacy assistants for improving data dashboards**” submitted by **Saamira Yasmin** (Roll No. 19CH10039) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Chemical Engineering is a record of bona fide work carried out by her under my supervision and guidance during Spring Semester, 2022-23.

Professor Mainack Mondal

Date: May 01, 2022 Department of Computer Science and

Engineering

Place: Kharagpur Indian Institute of Technology Kharagpur Kharagpur - 721302, India

ii

## Abstract

Name of the student: **Saamira Yasmin** Roll No: **19CH10039**

Degree for which submitted: **Bachelor of Technology**

Department: **Department of Chemical Engineering**

###### Thesis title: Designing and building smart privacy assistants for improving data dashboards

Thesis supervisor: **Professor Mainack Mondal**

Month and year of thesis submission: **May 01, 2023**

**Abstract**

Voice assistants like Amazon’s Alexa, Google’s Voice Assistant (GVA), Microsoft’s Cortana and Apple’s Siri have become a part of our daily life. While these voice assistants provide great features and save our time, they also collect a good amount of user data which is a privacy concern for many. Here comes data dashboard in the picture. Google provides its own data dashboard ‘My Activity Dashboard’ where a user’s personal data is stored. This data can be of three types, namely, textual data (transcripts), voice recordings and ambient location data. The data that is stored in these dashboards is huge and can have privacy issues if the user’s system is compromised. Although the My Activity Dashboard provides options to users to delete any saved data element, the process is tedious and not user- friendly. There arises a need for a more usable and user-specific dashboard design. We bring the idea of using Machine Learning to classify data elements as sensitive or not by using classification algorithms and improve the efficacy of dashboards.

iii

## Acknowledgements

Firstly, I would like to thank my BTP supervisor, Prof. Mainack Mondal for providing me the opportunity to work on this project and always guiding me. I am grateful to him for his constant support and motivation. He has always been there to answer my queries and give valuable feedback.

I would also like to extend my gratitude to Vandit Sharma, my senior, for his suggestions and insights. I express my gratitude to Shashvat Gupta, my fellow researcher in this project. It has been a great experience collaborating with him.

iv

# Contents

[**Declaration**](#_bookmark0) **i**

[Certificate](#_bookmark1) ii

[Abstract](#_bookmark2) iii

[Acknowledgements](#_bookmark3) iv

[Contents](#_bookmark4) v

[List of Figures](#_bookmark5) vii

[Abbreviations](#_bookmark6) viii

1. [Motivation](#_bookmark7) 1
   1. [Introduction](#_bookmark8) 1
   2. [Problem Statement](#_bookmark9) 2
2. The ML Model

2.1 The ML Model ……………………………………………………………………………….

* + 1. The Training Data
    2. Continual Learning 7
  1. Feedback System 15
     1. Working of the feedback system 15
     2. Participatory Design 16

1. The Plug-in
   1. [Motivation](#_bookmark30) 19
   2. [Study Design](#_bookmark31) 19
   3. [Results](#_bookmark32) 20
   4. [Future Work](#_bookmark35) 22
      1. [Effect of Continual Learning](#_bookmark36) 22

v

1. [Challenges and Future Work](#_bookmark38) 23
   1. [Challenges](#_bookmark39) 23
      1. [Time Optimisation](#_bookmark40) 23
      2. [Audio Scraping](#_bookmark41) 24
      3. [ML model in browser](#_bookmark42) 25
   2. [Future Work](#_bookmark43) 25
      1. [User feedback to improve model](#_bookmark44) 25
      2. [Possible removal of back-end component](#_bookmark45) 26

[Bibliography](#_bookmark46) 27

# List of Figures

* 1. [Data flow between components](#_bookmark12) 3
  2. [Class diagram for Scraperlib](#_bookmark14) 4
  3. [GAD Dialogue for a data item](#_bookmark15) 5
  4. [Class Diagram for ModelAPI](#_bookmark18) 10
  5. [Plugin Dashboard](#_bookmark21) 13
  6. [Broad workflow of system](#_bookmark24) 15
  7. [Participatory Design](#_bookmark29) 18
  8. [Old Design of plugin dashboard](#_bookmark33) 20
  9. [New design of plugin dashboard](#_bookmark34) 21

vii

# Abbreviations

**GVA G**oogle **V**oice **A**ssistant **ML M**achine **L**earning

**GAD G**oogle **A**ctivity **D**ashboard **JS J**ava**S**cript

**DOM D**ocument **O**bject **M**odel **UI U**ser **I**nterface

**\*** To be implemented

viii

**Chapter 1**

# Motivation

### Introduction

Collection of data by companies and big organizations is very common nowadays. Voice assistants like Google's Voice Assistant collects data which is stored in its My Activity Dashboard. Users have the option of going through this dashboard and deleting items that may appear sensitive to them. However, given the volume of data saved each day, it is inconvenient for users to manually read through each data element. The paper, Sharma et al. [1] suggests the need for a ML model that can classify data items as sensitive or not and the requirement of a user-friendly and a more personalized interface. User privacy preferences vary with demographics. In this thesis project, we use the XGBoost classification algorithm to assess the sensitivity of data in dashboards. Our goal is to provide users with the sensitivity percentages for their dashboard data and create a human in the loop framework which improves with user feedback and continually learns.

1

Chapter 1. *Motivation* 2

### Problem Statement

Our aim is to build smart privacy assistants for improving data dashboards. Data dashboards store huge amount of data collected from voice assistants. Although these dashboards provide options to users to delete any or all data stored in them, they fail to provide satisfactory usability and privacy to the users. Our goal is to render a more usable interface which caters to the privacy concerns of users. We build a Chrome extension which scraps data from Google’s My Activity Dashboard and then rank the elements based on the percentage of sensitivity. We build a continual learning based XGBoost Machine Learning model that classifies data elements based on the risk factor and improves usability of data dashboards for the end user.

Studies from the paper Sharma et al. [1] suggest the need for a recommender system that recommends users so as to what data is sensitive on their dashboard and improves both usability and security.

**Chapter 2**

# Ranking sensitivity of elements on GVA

### The ML Model

Predicting the sensitivity of data dashboard elements requires a ML model that can accurately predict the risk percentages for each element on the dashboard. Studies from the paper Sharma et al. [1] show that XGBoost classifier performs best among other supervised ML algorithms like Support Vector Machines (SVM), Logistic Regression, Random Forest, Multi-Layer Perceptron (MLP).

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient- boosted decision tree (GBDT) Machine Learning library. It is an ensemble ML algorithm meaning it combines multiple machine learning algorithms to obtain a better model. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems. Supervised machine learning uses algorithms to train a model to find patterns in a dataset with labels and features and then uses the trained model to predict the labels on a new dataset’s features.

3

#### Training Data

###### We collected My Activity Dashboard data of 79 participants. This data included transcripts, voice recordings and location details. The collected data was transformed into a set of features to maintain user anonymity and prevent user data leakage. Our XGBoost model accepts a feature vector as input for training and testing purposes. To prevent data leak and privacy issues, the task of feature extraction was done in the chrome browser itself.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| **Unigram** | Unigram feature extraction takes as input a sentence and returns each  individual word from it |
| **Bigram** | Bigram feature extraction takes a sentence and gives us sets of two con-  secutive words in the sentence. |
| **Trigram** | Trigram feature extraction takes a sentence and gives us sets of three  consecutive words in the sentence. |
| **Swear** | We borrow a list of swear words with the intuition that sensitive tran- scripts might contain swear words. We then construct a feature vector of  length 72. |
| **Sentiment** | We obtain the sentiment for each word in a transcript. This information  is then used to construct a feature vector of length 4. |
| **Emotion** | We constructed an emotion feature having 8 fields representing the count of anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. This  feature was constructed using the NRC emotion lexicon. |
| **Regret** | The obtain regret words of eight categories- cursing, drug, health, rela- tionship, racial and religion, sex, violence, and work. We then construct  a feature vector of length 8 using these. |
| **Sent2Vec** | We obtain 525-dimensional spaCy word embeddings corresponding to  each word in the transcript. |
| **LIWC** | We use the 64 semantic categories in the Linguistic Inquiry and Word  Count dictionary to construct a feature vector of length 64. |

###### Table 2.1.1. a) Text features

Most of the text features are easily extract-able using lexicons extracted by previous works. We also used javascript implementation of [spaCy](https://spacy.io/universe/project/spacy-js) and [N-gram](https://www.npmjs.com/package/n-gram) to extract some text features. For less enthusiastic users of Google Voice Assistant, most collected data consists of texts only in form of search text on google and youtube, browsing history etc.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| **Soundnet** | Soundnet [(Aytar et al.](#_bookmark47) [(2016))](#_bookmark47) learns rich natural sound repre- sentations by capitalizing on large amounts of unlabeled sound data collected in the wild. We use sound representations from the last layer of the SoundNet network as a feature vector of length  1024. |
| **MFCC** | Mel-frequency cepstral coefficients [(Hossan et al.](#_bookmark49) [(2010))](#_bookmark49) of a  signal are a small set of features (in our case 26) which concisely describe the overall shape of a spectral envelope. |
| **Spectral contrast** | Spectral contrast [(Jiang et al. (2002))](#_bookmark50) considers the spectral peak,  the spectral valley, and their difference in each frequency sub- band. We use it as a feature vector of length 7. |
| **Tempo** | Tempo is the rate or speed of the beat. |

###### Table 2.1.1. b) Audio features

These features are extracted from voice recordings collected by Google Voice Assistant (GVA). We use the library [Meyda](https://meyda.js.org/) to extract most of the audio features described above.

###### Table 2.1.1. c) User features

|  |  |
| --- | --- |
| **Feature** | **Description** |
| **Age** | Age of the user is taken during post-installation phase of the extension. |
| **Gender** | Gender of the user is taken during post-installation phase of the exten-  sion. It is a feature vector of length 2. |
| **Major** | Any association with a computer science major is asked from the user in  the post-installation phase. It is a feature vector of length 2. |
| **Duration** | Duration of usage of google activity dashboard asked with other user  features. |
| **Account** | Duration of the existence of the user’s google account corresponds to this  feature. |
| **Frequency** | Frequency of usage of google assistant corresponds to this feature. |

Additionally, we feed in a set of user features to the model. The paper Sharma Et al. [1] suggests that user-based features are most important for sensitivity prediction. We find that they are also useful for personalization of the model and creating user-specific ML models.

###### 

###### We use a scraper library to scrape user data from Google’s My Activity Dashboard.

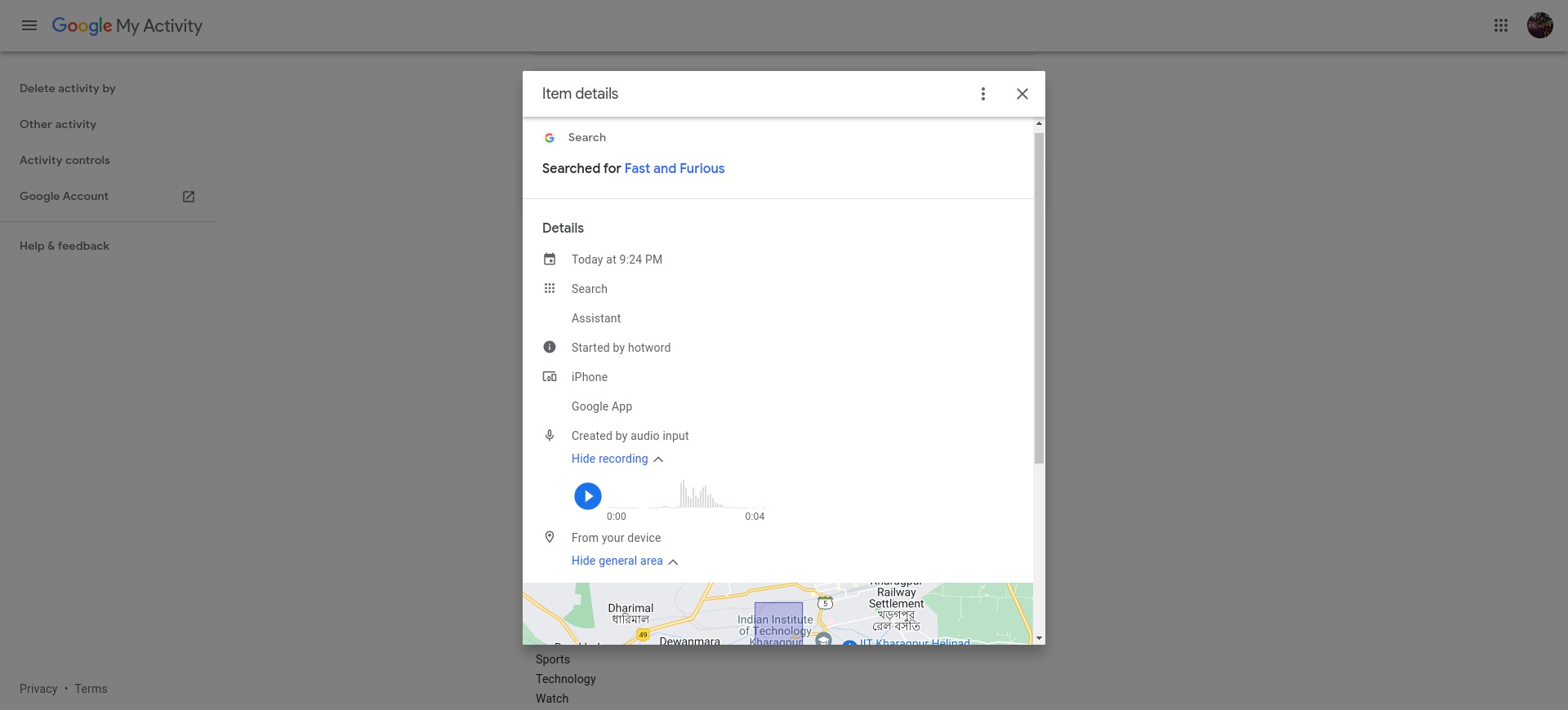


Figure 2.3: GAD Dialogue for a data item

For each data item, the dashboard provides a detailed dialogue box. This dialogue box contains all necessary information to be scraped. The scraper bot first opens the dialogue box of a data element. From this dialogue box, the scraper bot scans nodes with particular information type e.g. hyperlinks, voice recordings, location data etc.

#### Continual Learning

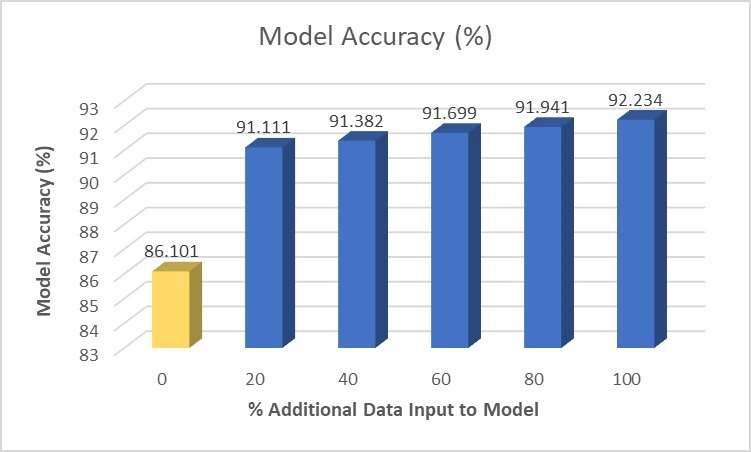
Continual Learning is a Machine learning concept wherein the model learns continuously and evolves based on the input of increasing amounts of data while retaining previously-learned knowledge. Our base XGBoost model predicted results on the test dataset with an accuracy of 86.1%. We wanted to improve the accuracy of our predictions and thought of applying continual learning to our model. Using additional training data sets we retrained the base model to perform better on unseen data.

Fig 2.1.2: Impact of Continual Learning on XGBoost model predictions

Fig. 3.2.1 shows the impact of adding more user data (features and labels) to our model and the increased performance. The yellow histogram with 0 additional data represents the base XGBoost model which was trained on 65% of the total dataset. We observed an increase of nearly 6.1% in the accuracy of our sensitivity results when using continual learning.

#### Feedback System

#### We incorporate user feedback in our model for continual learning purposes. User feedback helps us reduce the false positives and false negatives and aims to increase the model performance. We implement the concept of human-in-the-loop architecture to increase the accuracy of sensitivity predictions of our model.

#### 2.2.1 Working of the feedback system

#### The first set of predictions are given by the model and displayed in the interface of the plug-in. Thereafter, the user specifies the correctness of each predicted label against each data item on the dashboard. The user selects either ‘yes’ or ‘no’ as per the user’s understanding of sensitivity and risk. These corrected labels are then sent to the feedback model for re-training of the continual learning model. This iterative process of re-training based on user feedback should increase model performance and prediction results. A feedback system also ensures personalization based on user’s sense of privacy.

#### Figure 2.2.1: Schematic of the overall workflow

#### 2.2.2 Participatory Design

After the plugin was created, we needed to test the usability of the plugin’s interface so as to see if the plugin was in a user-friendly format as we needed to improve over the baseline defined by the Google My Activity Dashboard. So, we performed a **Co-design study** on the plugin’s dashboard to evaluate usability of plugin dashboard with respect to Google’s Activity Dashboard and incorporate design suggestions collected from users in the plugin dashboard.

#### 

#### 

#### Figure 2.2.2: Participatory Design

With the recruited participants, 1:1 meetings were scheduled on Google Meet, which was recorded and transcribed using the tl;dv extension. During these interviews, we asked a few demographic questions such as age, gender, and education qualification.

After this, the participants were familiar with the Google My Activity Dashboard, and several tasks were given to them to perform on the dashboard itself. After completion of the tasks, the participants were asked about the difficulty of the tasks and some about the dashboard interface.

The results from the survey were as under:

The users expressed a profound need for the functionality to filter the data according to their needs. At the same time, there was a need for a guide/manual to use the plugin and all features we get through the plugin. There were many buttons that were unintuitive for the users to click or what they meant. This helped us in modifying the user interface of our plug-in and create a more user-friendly extension with the feedback received from our participants.

**Chapter 3**

# The Chrome Extension

* 1. WORKFLOW

We integrate our back-end server with the front-end and create a Chrome plug-in which the user can install in his browser. Upon successful installation of the extension, the workflow typically looks like:

* + 1. The user is prompted to fill a form which asks for the user’s personal information e.g. age, gender etc. This information is required to train the classification model. As stated in Sharma and Mondal (2022), user features tremendously improve the model’s accuracy. The user may opt out of providing this information and in that case the model is trained using some default user features.
    2. After the user fills the form, the information is sent to the back- end application by the extension’s background script via a POST request.
    3. The back-end uses this information to train a XGBoost classification model with premium accuracy. This model is saved, compressed, and returned as response to the extension’s background script.
    4. On receiving the model, the background script stores the model. This model will be used to predict the risk probability of data elements while the extension is generally used.
    5. The user sends periodic feedback to the classifications made by the model and the model re-trains for improved accuracy
  1. **The Interface**

The interface is aimed to be more usable than the Google Activity Dashboard. In the least, it must include the most usable and important features provided by GAD. Some must-need features (implemented and required) are as follows:

* View all data elements in an ordered fashion
* View description of data elements in depth
* Take feedback regarding classification of data items
* Allow user to change the ordering fashion according to their convenience
* Nudge user about sensitive data items
* Allow user to delete an activity
* Allow user to delete all sensitive-suggested elements
* Provide interface to safely change settings of the plugin
* Allow user to edit their personal information
* View privacy and terms of the plugin
* Provide a help guide for the plugin

###### Interface Design

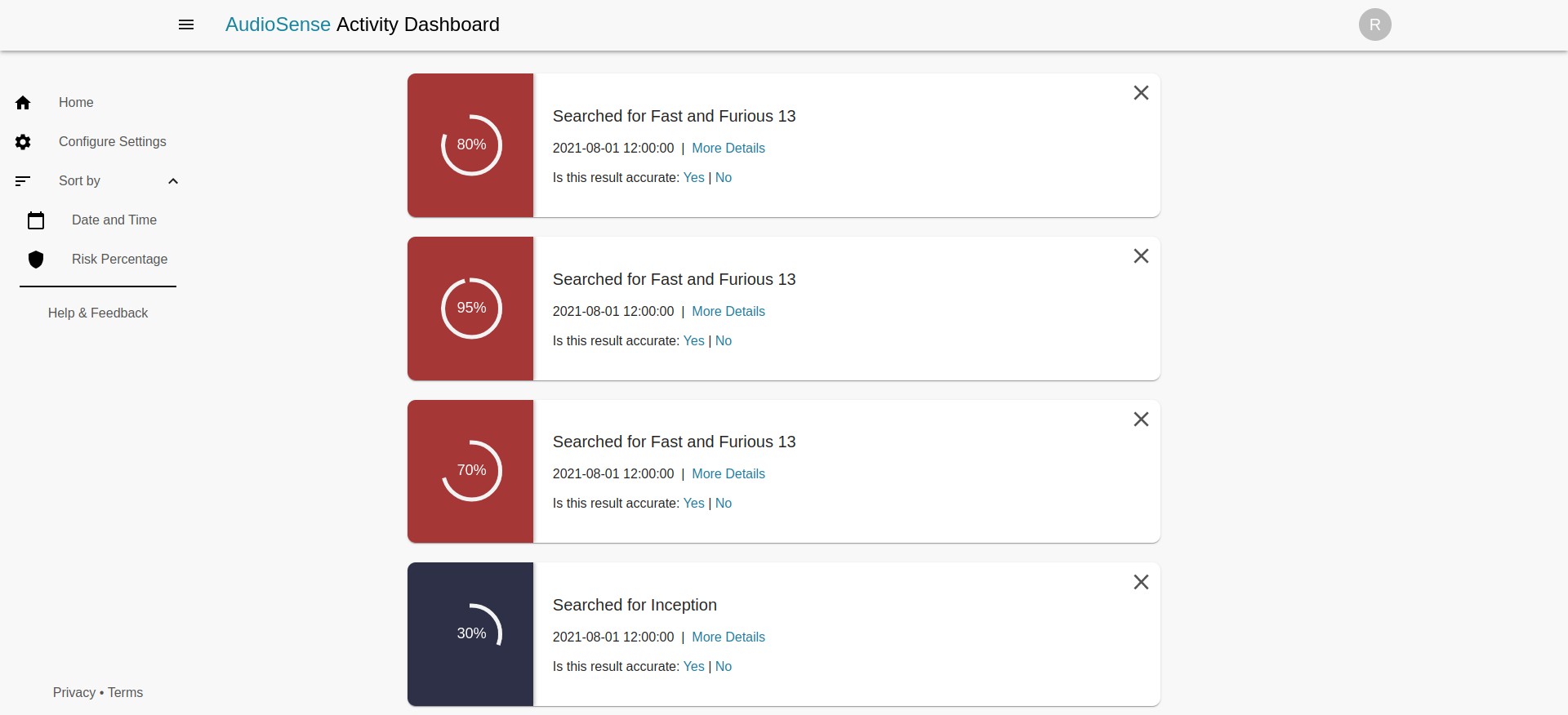


Figure 3.2: Plugin Dashboard

Feature was added to pipeline filters and add/remove these filters from the pipeline. The extension will launch itself once the Google Activity Dashboard is accessed; this reduces the number of clicks required by the user to access the extension and is aimed at increasing user retention on the plugin.

**Chapter 4**

**Future Work and Challenges**

#### Future Work

##### Testing the model with User Feedback

Currently, we have tested the extension based on only one-time input from the user which includes the user features. We now need to test our continual learning model based on user feedback. The user needs to input his feedback based on the displayed results in the front-end which will then be sent to the back-end server for retraining and improvised results.

##### Possible removal of back-end component

Currently, the back-end application fulfills three purposes:

* Trains the **XGBoost** model using user’s features
* Apply **continual learning** over the model to increase its accuracy
* Accept feedback from user and re-train model

All purposes could be solved if the **ml-xgboost** library had the support for them. As a future work, we plan to further improve the library by incorporating all necessary support for all of these uses. This would enable us to get rid of the whole back-end application component all together and perform all operations regarding training, continual retraining, loading and usage inside the extension itself. This would make the extension much more reliable, secure and fast.

**4.1.3. User Survey**

We plan to conduct a survey and recruit participants for the same. We wish to create personalized models for each user based on user’s privacy preferences. The survey shall have questions pertaining to the user’s privacy choices as well as general questions to understand the user’s mindset and knowledge about GVA.

We also plan to find the temporal effects on risk classification and how users’ privacy choices change over time.

#### Challenges

#### 

#### 4.2.1 ML model in browser

A vital challenge which aroused was to unpack and load the saved XGBoost model in the browser’s environment. This task was challenging due to a lot of reasons.

* + - 1. Lack of libraries for unpacking or using ML models in browser environment
      2. Existing libraries could only unpack basic ML models and not XGBoost model
      3. Format in which model was saved was **human-unreadable**, forbidding us to write our own interpreter
      4. After long search and finding the library **ml-xgboost**, the library was very much crude and had issues listed in the section [2.1.3.](#_bookmark19)

We improved upon the library **ml-xgboost** and used the library in our extension. Now, we can easily load, unpack and use our saved XGBoost model.

# Bibliography

* + - 1. Sharma, V. and Mondal, M. (2022). Understanding and improving usability of data dash- boards for simplified privacy control of voice assistant data. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3379–3395, Boston, MA. USENIX Associa- tion.
      2. Oh, the Places You've Been! User Reactions to Longitudinal Transparency About Third- Party Web Tracking and Inferencing. *CCS '19: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security: Ben Weinshel, Miranda Wei, Mainack Mondal, Euirim Shoi, Shawn Shan, Claire Dolin, Michelle Mazurek and Blasé Ur.*
      3. Aytar, Y., Vondrick, C., and Torralba, A. (2016). Soundnet: Learning sound representa- tions from unlabeled video. *CoRR*, abs/1610.09001.
      4. Jiang, D.-N., Lu, L., Zhang, H.-J., Tao, J.-H., and Cai, L.-H. (2002). Music type classifi- cation by spectral contrast feature. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 113–116 vol.1.