Classification of Users based on Familiarity with Browser Functions

Classes:

1. Beginner
2. Intermediate
3. Expert

Platform Used:

Chrome Extension - Extensions allow you to add functionality to Chrome without diving deeply into native code. The most important file required for the extension to function is the manifest.json file. This manifest is nothing more than a metadata file in JSON format that contains properties like the extension's name, description, version number. Chrome comes packed with a lot of developer tools, one of which is the Chrome Extension API set. The use of APIs to access browser data can raise certain security issues, hence appropriate permissions are required to access the data. These permissions are also part of the manifest.

Training Data:

The training data was collected by creating a quiz which tests the familiarity of the user with the functionality of the browser. The data was collected from users belonging to the age group 15-60. The users were tested for the following:

1. Adding a Bookmark

API used – chrome.bookmarks

Permission - bookmarks

1. Creating a Bookmark Folder

API used – chrome.bookmarks

Permission - bookmarks

1. Changing the Font Size

API used – chrome.fontSettings

Permission - fontSettings

1. Changing the following Privacy Settings:

API used – chrome.privacy

Permission - privacy

* 1. Autofill Settings
  2. Spelling-Error Detection
  3. Third-Party Cookie Settings
  4. Password Saving
  5. Translation Services

1. Removing a Bookmark

API used – chrome.bookmarks

Permission – bookmarks

1. Deleting a History Item

API used – chrome.history

Permission - history

The time taken by the user to perform the above tasks was timed individually. The user action is tracked and on completion of a particular task, the next task is displayed. The chrome.tabs API is used to switch the browser tabs, and requires the permissions “tabs” and “activeTab”.

Apart from the above tasks, the users were asked about the following:

1. Purpose:
   1. Searching
   2. Surfing
   3. Development of Web Applications
   4. Two of the above options
   5. All of the above options
2. Usage:
   1. Almost none or lesser than five minutes
   2. Lesser than fifteen minutes
   3. Lesser than half an hour
   4. An hour or two
   5. More than two hours

Storage:

The data was stored locally in the system. The localStorage functionality of the browser was used to temporarily store the intermediate test results for a user. To store the final results of the users we use the chrome.storage.local API provided by Chrome. The permission required to access storage is “storage”.

Classifier:

Decision Trees - Decision Trees were used for the classification process. A decision tree is an algorithm that contains only conditional control statements. It is a flowchart-like structure in which each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. In this case, the internal nodes are the tasks to be performed by the user, the branches are the outcomes of the task, and the leaf nodes represent the experience-level of the user (Expert, Intermediate, Beginner).

After the user has been classified, we modify the home page of the user so that it suits his/her needs. To change this, we add the “chrome\_url\_overrides” object to the manifest. This enables us to set a page that replaces the “New Tab” page.

The home pages of different users have the following features:

1. Expert
   1. Search Engine
   2. Frequently visited sites
   3. Recommendations
2. Intermediate
   1. All features of the Expert User
   2. Links to various Browser Functions
3. Beginner
   1. All features of the Intermediate User
   2. Description of various Browser Functions