**STUDY SCRIPT**

**Introduction:**

Hello, and thank you for agreeing to participate in our study today. My name is Himesh Nandani and I will be conducting the experiment with you. Before we begin, I would like to provide you with some information about the study and answer any questions you may have.

Purpose of the Study:

The purpose of this study is to validate if feedback loops improve the performance of deep learning models in the area of code smells detection. Code smells are code structures or design choices that are considered to be suboptimal and may indicate a deeper problem in the codebase. Examples of code smells include long method or class, duplicate code, and feature envy. We are interested in understanding if the machine learning model we use to predict code smells can improve if we pass the user feedback for predictions back to the model. The information we gather from this study will be used to improve code quality and help software developers make more informed decisions when refactoring their code.

Experimental Design:

For this study, we will be using a randomized controlled experiment. Participants will be randomly assigned to either the control group or the experimental group. The control group will be asked to provide feedback on code snippets using the IntelliJ plugin without the deep learning model, while the experimental group will be asked to use the IntelliJ plugin with the deep learning model. This will allow us to compare the effectiveness of the plugin with and without the deep learning model.

Procedure:

During the study, you will be asked to use an IntelliJ plugin that presents you with code snippets containing potential code smells. You will be asked to provide feedback on each snippet, indicating whether you believe it contains a code smell or not. The study will take 90 mins over two sessions of 45 mins each. The experiment will be conducted on a computer with IntelliJ installed. You will be provided with all the materials needed for the experiment.

The detailed procedure is as follows:

1. Open the IntelliJ IDE and navigate to the "Tools" menu.
2. Select "TagCoder" and then click on "Analyze Code".
3. Wait a few seconds for the plugin to analyze the project.
4. Open the classes in the project one by one.
5. In the left part of the editor, look for the icon of TagCoder in the gutter.
6. Click on the TagCoder icon to see if the plugin has detected any code smells.
7. Validate whether the predicted code smell by the plugin is correct.
8. Provide feedback using the dialog box that pops up, indicating whether you agree or disagree with the plugin's prediction.
9. Repeat this process for multiple classes in the project.
10. When finished, click on "Retrain Model" in the TagCoder menu to improve the performance of the deep learning model.

Note: Please make sure to follow the instructions carefully and provide accurate feedback as this will help us to improve the effectiveness of the deep learning model.

This procedure is also outlined in the fact sheets next to you.

Risks and Benefits:

There are no known risks associated with participating in this study. However, there may be potential benefits from participating, such as contributing to the development of better tools for software developers.

Confidentiality:

All information collected during this study will be kept confidential. Your name will not be associated with any of the results. Only the researchers involved in the study will have access to the data collected.

Voluntary Participation:

Participation in this study is voluntary. You are free to withdraw from the study at any time without any penalty. If you decide to withdraw, your data will not be included in the study.

Contact Information:

If you have any questions or concerns regarding this study, please feel free to contact us at [hnandani@dal.ca](mailto:hnandani@dal.ca).

Conclusion:

Thank you for taking the time to participate in this study. Do you have any questions or concerns before we begin?