**Project Title:**

Predicting Areas of Interest in Code Reading

**Student:**

Jenna Wise

**Mentors:**

Dr. Bonita Sharif, Dr. Alina Lazar

**Institution:**

Youngstown State University

**Relevant URLS (Project website, blog, etc.)**

Blog:

<http://ysu-creu2015-2016.blogspot.com/>

**Poster Presentation:**

**Wise, J.**, **Whitely, J.,** Husain, A., Lazar, A., Sharif, B., “Predicting Expertise from Eye Gazes on Source Code”, ACM Richard Tapia Celebration of Diversity in Computing, Austin, Texas, 2016

**Goals and Purpose of the Project:**

We are focusing on analyzing eye gaze data gathered in our Software Engineering and Empirical Studies Lab at Youngstown State University. The goal of this project is to determine whether we can predict if a software developer is a novice or expert based on their eye gaze data. We are also interested in determining which features extracted from the code, such as lexical, syntactic, and semantic features are best for this prediction, and which machine learning algorithms give us the most accurate prediction. Another goal of this project is to compare the scan paths of novices and experts, using sequential analyses.

**Process Used in Completing the Research:**

In order to perform expertise prediction on our eye gaze data, we used five different machine learning algorithms: two-class boosted decision tree, the two-class decision jungle, the two-class decision forest, the two-class local deep SVM, and the two-class neural network machine learning algorithms in Microsoft Azure Studio. In order to run our data in Microsoft Azure, we followed the following process.

An initial master file, containing 106,143 rows and 20 columns, including all of the fixation data across all participants and across all tasks is first created. Three features were selected from the initial dataset: fixation duration, fully qualified name (FQN) and expertise. The expertise is determined based on a background questionnaire. The industry participants were labeled as professionals and the students were considered amateurs as they did not have much experience programming in Java.

 During the next pre-processing step all the instances in the dataset with missing values are removed leaving us with 60,941 rows. Quantizing the duration field into 10 bins makes the column appropriate to be used as weights for the feature column. To build a binary classification model to predict expertise based on the FQN feature and duration around 10% of the data is selected for cross-validation.

 The training set consists of 70% of the data, of which 28% was used for a 10-fold cross validation. The other 30% of the data was used for scoring and testing. The model generated in Microsoft Azure is then used to predict instances in a testing set. Accuracy, precision and recall results are reported for each of the models trained. Finally, predicted results are aggregated for each participant in the study. Majority voting is used to decide based on all predicted instances for the specific participant and task to determine if they are a professional or an amateur.

 We started working on our first sequential analysis using an open source tool called ScanMatch. We mapped our eye gaze data into sequences, in order to run them within ScanMatch. It pairwise compares participants based on weighted Areas of Interest. Each Area of Interest is one line in the source code used in the data collection experiment. A weighted matrix was created in order to weight each Area of Interest (AOI); the weight is higher if the two AOIs being compared are closer to each other in the source code. Once the weight matrix was created and our data turned into sequences, we compared each participant to every other participant within each task using ScanMatch.

**Conclusions and Results:**

We find that for SVM, Neural Network (NN), and Boosted Decision Tree models, in 19 out of 22 participants the prediction was made correctly giving an accuracy of 86.36%. In Decision Jungle and Decision Forest, 18 out of 22 were predicted correctly giving an accuracy of 81.81%. Note that we had a total of 22 participants. We then look at these values individually in each task (our data consists of three separate tasks per participant). We find that for task 2, the NN and SVM predict correctly with 100% accuracy but the accuracy drops to 90% and 80% for tasks 3 and 4 respectively. We believe the numbers are still reasonable high i.e., 80% or over.

 We see from the above results that in a majority of the cases, we get a good prediction based on the fixation data on source code elements with SVM and NN, and find that eye gaze fixations weighted by their durations are a good indicator of developer expertise.

 The preliminary results from our sequential analysis are somewhat promising. The comparison scores amongst novices are higher than the comparison scores of novices to experts and the comparison scores amongst experts are high than the comparison scores of experts to novices for each task. These results come from an initial glance at the resultant scores. Although, the normalized scores overall are very low. No score reaches over 0.5. We would like to perform statistical analyses on the resultant scores in the future and we would like to consider a different weight matrix.