Code Readability Group Progress Report

Preston Ford, Rai Katsuragawa, Juneon Kim, YoungWoo Song, Jeffry Yoon

Project Description

This project aims to develop a machine learning model that predicts the readability score of Python code snippets. Code readability is crucial for software maintenance and collaboration among developers. By scoring the readability, our model can help identify code that may need refactoring to improve quality and maintainability.

Data Source and Gathering Method

We are going to collect data from open-source repositories like GitHub and coding platforms like LeetCode. We will mainly use GitHub as our data source since it has a variety of code snippets and projects. To retrieve the data from GitHub, we plan to use the API provided by GitHub to search repositories, issues, and files by keywords, languages, and other filters. Additionally, we are going to use a web scraper to collect the data from LeetCode to complement more instances and data variety. We will only gather the code data that has been written in Python due to the complexity of comparing code written in different languages. We will also use some convenient Python libraries that can interpret the dataset and output the relevant information that can be used for evaluating features.

Feature Explanation with Example Data

Ratio of Comments	Max Line Length	Complexity	Variable Name Quality	Style Guide Adherence	Readability

0%	270	High	Bad	15%	Low
20%	100	Medium	Good	50%	Medium
40%	60	Low	Good	80%	High

We plan on having five features. These five features are the ratio of lines of comments to the total number of lines of code, the length of the longest line, complexity of the code, the quality of variable names and adherence to the style guide. To measure complexity, we will be using the Radon library which outputs an integer with no upper bound and a lower bound of 1. Numbers 1-20 will be considered low complexity, 21 - 40 will be considered medium complexity and numbers higher than 41 will be considered high complexity. For style guide adherence, we will be comparing the code to the PEP 8 style guide using the pylint library.

Initial Machine Learning Models

Regression Models:

1) Linear Regression: This can be a baseline model to see if there’s a linear relationship between features and readability score. Fit linear regression model to the features and check the initial output of how each feature contributes linearly to readability.
2) Decision Trees: This can be a model to capture nonlinear relationships between features and readability score. Build a decision tree regressor, allowing the model to create splits based on features that may reveal deeper insights.

Schedule

Tuesday, Nov 12: Finish data collection

Tuesday, Nov 26: Finish training machine learning models

Monday, Dec 2: Finish presentation slides