**CS 591 Programming Project 3**

**Predicting Burnout in GitHub Discussions**

**Grade**: 100 marks in total (Program: 70%, Team Review: 30%)

**Type**: Group work (Team size cannot exceed 6)

**Programming language & Framework**: Python, TensorFlow or any other Neural Network frameworks.

**Hints**: Review Jupyter Notebook files (.ipynb) files on Canvas to build your own Neural Networks with TensorFlow framework

**Project Overview:** This assignment aims to know how to run predictions over a dataset using NLP and sentiment analysis.

**MAIN GOAL**: We want to identify whether burnout is present in this project.

**Tasks:**

# Part 1: Dataset and Sentiment Analysis

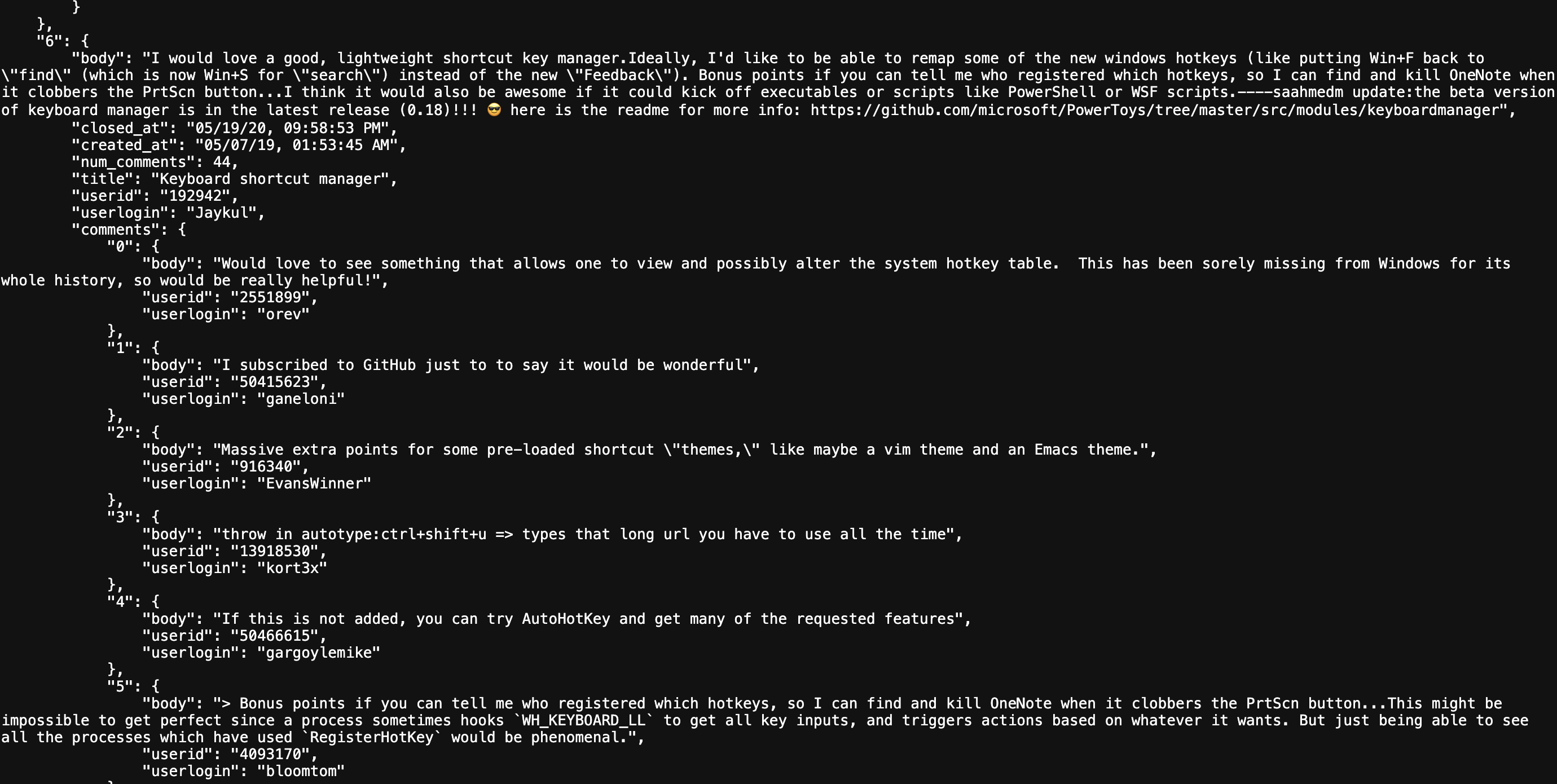
The dataset contains a range of issues and pulls requests from the PowerToys project in JSON format (<https://github.com/microsoft/PowerToys>).

The fragment below shows an example: Issue 6 has a title, body (text), user that posted it, dates of creation and close, and 44 comments. The comments are numbered in sequence with the user and body.

Goal 1: run a sentiment analysis to identify positive, neutral, and negative comments.

Obs:

* You can discard issues/pull requests without comments.
* You can convert the JSON into a CSV if you want.



**Figure 1**

Goal 2:

* Using the definition for burnout: <https://www.thesaurus.com/browse/burnout>

Identify every comment with burnout. Be creative! Label each comment with “Burnout” when at least one occurrence of the burnout words is present; it may not be precise! For example, the word collapse is present, but the sentiment analysis is positive, or when an antonym is present. There may be a library with burnout defined as **Intent Analysis or Emotion Detection (what are those concepts?). This process may need some tunning and group discussions.**

The team may disagree with the automatic detection (like in the case described in the previous paragraph), and when that happens, the team may fix it.

A list of every updated comment should be created to have scientific validity, and every team member may agree or disagree. Ex:

Issue/PR #1:

Comment #1: Burnout = Yes

Team Members Agreement/Disagreement:

Joe = No, Phillip = No, Smith = No …

Comment #2: Burnout = No

Team Members Agreement/Disagreement:

Joe = Yes, Phillip = Yes, Smith = Yes …

The list above should contain only the issues/PR comments you changed manually.

**Your report should include:**

* A new “sentiment” information with values “Positive”, “”Neutral, and “Negative” for each comment.
* A new “emotion” information with values like (happiness, frustration, anger, sadness, etc.)
* A new “intention” information (values can be wide-open).
* A new “burnout” information with values “Yes,” and “No” for every comment.

Ex (JSON):

…

“1”: {

“body”: “bla bla bla”, “userid”: “111”, “userlogin”: “aaa”, “sentiment”: “positive”, “emotion”: “happy”, “intent”: “welcoming”, “burnout”: “No”.

…

* The set of words (found in your dataset) contributed the most to the sentiment analysis and burnout detection.
* The list of manually classified burnout comments and the agreement/disagreement of the team members.

Obs: see examples here:

<https://www.kaggle.com/code/andradaolteanu/covid-19-sentiment-analysis-social-networks>

<https://www.analyticsvidhya.com/blog/2021/06/nlp-sentiment-analysis/>

<https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17>

# Part 2: Neural Network/NLP

Predict burnout using deep learning/machine learning methods. Now you have a new dataset with burnout labels. We want to predict when an issue/pull request has burnout signs.

Use any NLP technique (TF-IDF, Word2Vec, Doc2Vec, BERT) to transform your issues/PR text into features. Tip: you may concatenate text from the issue title and issue body with comment bodies to have only one corpus!

Report precision, recall, and f-measure.

**Elements that need to be in the report:**

The report should contain the following information:

* Detail the motivation and significance of the burnout detection and prediction.
* Include relevant references (well cited, high impact factor journals and conferences, etc.).
* Describe and discuss the final methodology employed by the team to solve the problem.
* Present and discuss the results achieved by the team.
* Discuss the strengths and limitations of the proposed methodology or application.

Each of the components of the final project (report & presentation, source code, peer review) will be given a score between 0 and 100 and will be weighted accordingly to obtain the final grade.

# Submission Instructions

You will need to turn in the following:

1. A **report** in **PDF format**. The report should briefly describe your implemented solution and fully answer the questions above. **Remember: You will not get credit for any solutions you have obtained that are not included in the report.** **Your report should allow me to understand your solution without running your source code.**

**Paper in Overleaf (LaTex)**: a project report may include at least the following items: **abstract, introduction** (**problem description and research question(s)), related work (how previous papers tried to solve this and why your solution is different/better), method (think about the architecture to answer the research question(s) – Ex: if you want to improve the accuracy – RQ1: To what extend our approach improves the accuracy? Include algorithm (pseudocode)**, and **implementation details – A picture helps a lot)**, **running results and discussion (analysis, co,pare with the previous work and answer the RQ(s))**, **conclusions on what you have learned in the project, future work (things that would be great but are out of the scope or unfeasible to do now)** , and **references**.

**OBS**: A report written in conference format using new NLP techniques and good precision/recall/f-measure is an excellent candidate for submission. I can help review projects for possible submission!

(Suggestion: <https://conf.researchr.org/home/icsme-2024> )

1. At the **beginning** of the report, the report must contain all the team members’ name, ID, graduate or undergraduate (e.g., Juefei Yuan, S0000000, graduate)
2. Your **source code** (in a GitHub repository). The code should be well commented, and it should be easy to see the correspondence between what's in the code and what's in the report. You need to include executables and supporting files (e.g., utility libraries). I will contact you if I can’t run your code to evaluate your solution. Don’t forget the dataset.

Please **zip** both the report and source code into one package and name it based on the following rule:

“**Team leader\_First\_Name\_ Team leader \_Last\_Name\_P3.zip**”

(i.e., Juefei\_Yuan\_P3.zip).

**Submission venue:** Submit your files through Canvas.

**Late policy:** You lose 25% of the points for every day the assignment is late. If you have a compelling reason for not being able to submit the assignment on time and would like to make a special arrangement, you must let me know **at least a week before the due date** (any genuine emergency situations will be handled on an individual basis).

**Academic integrity:** Feel free to discuss the assignment with each other in general terms, and to search the Web for general guidance (not for complete solutions). Coding should be done individually. If you make substantial use of some code snippets or information from outside sources, be sure to acknowledge the sources in your report.