Predicting Repository Upkeep with Textual Personality Analysis

by

Alexander Sachs

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Masters of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2019

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Michael Godfrey
Professor, Dept. of Mathematics, University of Waterloo

Supervisor(s): Jesse Hoey
Professor, Dept. of Mathematics, University of Waterloo

Internal Member: Meiyappan Nagappan
Professor, Dept. of Mathematics, University of Waterloo

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

GitHub is an excellent democratic source of software. Unlike traditional work groups however, GitHub repositories are primarily anonymous and virtual.

Traditional strategies for improving the productivity of a work group often include external consultation agencies that do in-person interviews. The resulting data from these interviews are then reviewed and their recommendations provided. This is one such claim of a group of strategies called group dynamics. In the online world however where colleagues are often anonymous and geographically dispersed, it is often impossible to apply such approaches.

We developed experimental methods to discern the same information that one would normally obtain through in-person interviews through automated means. Here we provide this automated method of data collection and analysis that can later be applied for the purposes of recommendation agents.

Comments from individual developers were collected via various GitHub APIs. That data was then converted into personality traits for each individual through textual persona extraction and mapped to a personality space called SYMLOG. The resulting dynamics between each of the personalities of the developers of each repository are analyzed though SYMLOG to predict how successful each project is likely to be. These predictions are compared against valid preexisting success metrics.

Acknowledgements

I would like to thank Jesse Hoey for his guidance and support over these past couple of years.

I would also like to thank Meiyappan Nagappan for his ongoing input and feedback through the Themis project.

I would also like to thank Jailton Coelho, Kenny Hancock, Rahul Iyer, Josh Jung, Andrew Li, Wenkai Li, Aarti Malhotra, Mina Nouredanesh, Neda Paryab, Deepak Rishi, Elizabeth Sachs, Eric Sachs, Harald Sachs, Selma Sachs, Nolan Shaw, Zahra Sheikh, and Ronghao Yang whose support and input all helped to make this thesis possible.

Dedication

This is dedicated to my friends and family whose support made this thesis possible.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Open source efforts have become increasingly important to the software industry. According to Black Duck Software, 78% of the 1,300 respondents surveyed reported the use of open source software in their daily business and only 3% did not use open source software in any way [37].

Open source software can be sourced through a variety of methods. One increasingly popular method is through the use of a coding platform and site called GitHub.com. GitHub.com is a website that provides a GIT version controlled environment to house proprietary and open source projects. It provides functionality for common development practices (bug reporting, code alteration recommendations also known as pull requests, feature discussion, etc.) all of which is available to the public for open source projects. As of Sep 30, 2018 GitHub hosted over 96 million projects and provided a medium for collaboration for over 31 million developers [13].

Despite the growing acceptance of open source products there are still a number of significant drawbacks to consider. One major issue unveiled in the 2017 OpenSourceSurvey was interpersonal conflict. Of the top six issues highlighted in the survey, five of them were behavioural: unwelcoming language or conduct, dismissive responses, unexplained rejection, conflict, and unresponsiveness. Additionally, 18% of respondents reported experiencing negative behaviour from others in their groups while 50% report witnessing such behaviour in general. Evidently personality conflicts in this collaborative space is a genuine concern and its resolution relevant to the software community as a whole.

To address these behavioural issues we analyzed each project as a team of developers. One approach to predict a project's stability was to assess the health of the team that was developing it. To this aim we used group dynamics to gauge the stability of the team to predict which groups would succeed. One of the most well-known group dynamics model is SYMLOG (SYstematic, Multiple Level, Observation of Groups) [20]. SYMLOG is a body of social psychology theories brought together by Robert F. Bales to better describe the underlying mechanisms behind the interaction of individuals within a group [2]. Bales postulated that behaviour indicative of group dysfunction (such as the behavioural issues highlighted in the OpenSourceSurvey) is a symptom of an underlying personality conflict within the group. This tension within a group is what Bales called "group polarization" and often led to a significant loss of productivity.

SYMLOG has over 40 years of research to support its theories. It has also been applied in industry to improve team efficiency for over 15 years in U.S. companies. Through this consultation, over a million surveys have been accumulated to further corrobarate SYMLOGs assertions [2].

While SYMLOG itself has been used in industry to assess group cohesion, a more established method of personality assessment itself, is the Five Factor Model (also known as the Big 5) which researchers across specialties concur is fundamental [29]. The Five Factor Model asserts the five characteristics that describe an individual's personality are:

- Openness: Artistic, curious, imaginative, insightful, original, wide interests
- Conscientiousness: Efficient, organized, planful, reliable, responsible, thorough
- Extraversion: Active, assertive, energetic, enthusiastic, outgoing, talkative
- Agreeableness: Appreciative, forgiving, generous, kind, sympathetic, trusting
- Neuroticism: Anxious, self-pitying, tense, touchy, unstable, worrying

SCG (SYMLOG Consulting Group) even proposed a mapping between the SYMLOG space and the Five Factor Model [15].

In this thesis I tested the effectiveness of utilizing SYMLOG theories to predict which GitHub repositories were actively maintained and compared it to a model developed by Coelho, et al. To attain the personalities for assessment, I used a chain of tools to derive individual SYMLOG personalities from the raw GitHub comments that are publicly visible. Coelho in contrast, utilized various metadata as his features (number of forks, number of issues, number of pull requests, etc.) to train their models.

In its traditional form however, SYMLOG is a predominantly manual task. It has been primarily applied in the industry through in-person interviews and questionnaires of the entire team of interest [2]. This worked well for corporate groups where participants are physically present and are compelled to cooperate. In the online collaborative domain of GitHub however where users have the option of anonymity and are geographically dispersed across vast regions, this approach is infeasible. As such, these interpersonal approaches to the SYMLOG process needed to be approximated in a way that could be automated and scaled. That is what was explored here.

The main contributions of this thesis were:

1. provided an automated method of approximating the SYMLOG process in the context of GitHub comments

2. applied multiple models in analyzing and comparing SYMLOG predictions to those of various machine learning techniques executed by Jailton Coelho, et al. [7].

3. compared the relative impact of SYMLOG personality interpretations to aggregated Big 5 metrics.

All code and data associated with this thesis can be found at https://github.com/jessehoey/THEMIS.C under SYMLOG_Thesis.

Chapter 2

Related Work

2.1 The "Big 5" Personality Model

The Big 5 is the predominant personality model. Psychologists across specialties concur that the Big 5 attributes are fundamental and are the minimum factors required to adequately describe the personality of an individual across a variety of circumstances (self-reports, ratings, language, etc.) [29]. These five factors are:

- Openness: artistic, curious, imaginative, insightful, original, wide interests
- Conscientiousness: efficient, organized, planful, reliable, responsible, thorough
- Extraversion: active, assertive energetic, enthusiastic, outgoing, talkative
- Agreeableness: appreciative, forgiving, generous, kind, sympathetic, trusting
- Neuroticism: anxious, self-pitying, tense, touchy, unstable, worrying

The Big 5 (a.k.a. the Five-Factor Model) is used in a variety of use cases. As the predominant personality model it is used (among others) to:

- Predict behaviour: Such as the study done by Barlett et al. whereby the Big 5 was used to predict aggression [3]
- Predict interpersonal relationship outcomes: As done in a study by Seidman et al. whereby relationship activities were correlated to the Big 5 traits [36]

- Predict susceptibility: Wall et al. correlated Big 5 characteristics to persuasion susceptibility [40]

2.2 SYMLOG

Systematic, **M**ultiple **L**evel, **O**bservation of **G**roups (SYMLOG) is a body of social psychology theories originally developed by Robert F. Bales, as well as a consulting group (SYMLOG Consulting Group (SCG)), that looks at the interactions between individuals as a group. Instead of adopting the Big 5 as their personality model, SYMLOG developed a personality model of its own specifically suited for evaluating individuals in the context of a group. SYMLOG uses this determined personality to unveil potential emotional conflicts in a group. The primary tenet of SYMLOG is the *theory of unification and polarization* in which similar cognitive representations are grouped together and the distance between cognitive clusters are dilated [2]. The main consequence of this theory is that individuals tend to see "good" people as better than they really are and "bad" people as worse than they really are. It is this theory of cognitive clustering that forms the foundation from which the SYMLOG method is based.

SYMLOG was born out of small-group interaction studies done by Bales. Initially, in order to annotate the actions of a group, a discrete set of 12 actions that captured the majority of group dynamics was developed. These 12 actions came to be known as the Interaction Process Analysis (IPA) categories. Through the study of thousands of groups with the use of these IPA categories for annotations, three personality axes were developed that were able to explain the majority of behaviour of individuals within a group. Hence, SYMLOG personality is measured based on those three dimensions. The dimensions are friendly versus unfriendly, acceptance versus rejection of authority, and dominance versus submissiveness. Each of these three dimensions are further subdivided into their opposing poles: friendly becomes **P**ositive, unfriendly becomes **N**egative, acceptance of authority becomes **F**orward, rejection of authority becomes **B**ackward, dominance becomes **U**pward, and submissiveness becomes the **D**ownward direction (see Fig 2.1 for a visualization). The SCG proposes that these poles are related to the Big 5 dimensions in the following way [15]:

- Extraversion: U/D
- Agreeableness: PF
- Conscientiousness: NF

- Neuroticism: NB
- Openness: PB

The coupling of Poles (ex. PF) represents the intermediate direction between these two poles.

In order to assess where an individual lies in this space Bales divided the area into 26 subspaces by utilizing intermediate directions (see Fig 2.1 for a visualization, each cube is a subspace).

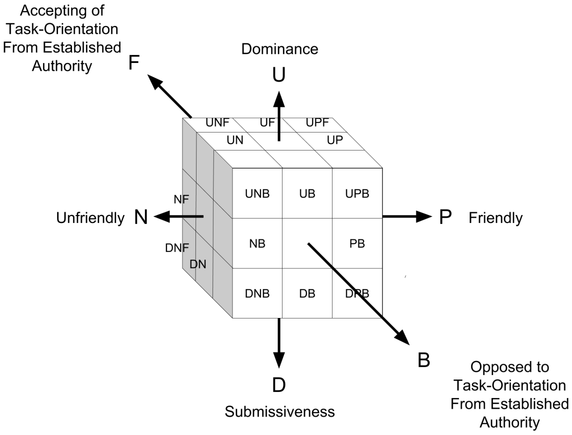
Accepting of Task-Orientation From Established Authority
F
Dominance
U
UNF
UF
UPF
UN
UP
UNB
UB
UPB
NF
Unfriendly
N
P
Friendly
NB
PB
DNF
DN
DNB
DB
DPB
B
D
Submissiveness
Opposed to Task-Orientation From Established Authority

Figure 2.1: SYMLOG Space: SYMLOG has 6 poles (**U**pward, **D**ownward, **P**ositive, **N**egative, **F**orward, and **B**ackward) that are combined to form 26 cubic subspaces where an individual's personality can lie. Bales visualizes an individual's personality as a force/vector originating at the center of this cube and terminating in one of these 26 subspaces [2].

Bales then created a 26 question survey such that each question would assess an individual's propensity towards each of these 26 subspaces of personality (see Fig 2.1). Each participant rates every other participant (as well as an ideal *wish* persona that represents what an optimal team member would behave like and a vilified *reject* persona that represents the exact opposite) and the scores (0 for rarely, 1 for sometimes, and 2 for often) are tallied up and averaged for each individual (see Table 2.2 for an example).

SYMLOG Survey

In general, what kinds of values does this person show in his or her behavior?

Code	Description	Response
U	Individual financial success, personal prominence and power	0/1/2
UP	Popularity and social success, being liked and admired	0/1/2
UPF	Active teamwork toward common goals, organizational unity	0/1/2
UF	Efficiency, strong impartial management	0/1/2
UNF	Active reinforcement of authority, rules and regulations	0/1/2
UN	Tough-minded, self-oriented asstertiveness	0/1/2
UNB	Rugged, self-oriented individualism, resistance to authority	0/1/2
UB	Having a good time, releasing tension, relaxing control	0/1/2
UPB	Protecting less able members, providing help when needed	0/1/2
P	Equality, democratic participation in decision making	0/1/2
PF	Responsible idealism, collaborative work	0/1/2
F	Conservative, established, correct way of doing things	0/1/2
NF	Restraining individual desires for organizational goals	0/1/2
N	Self-protection, self-interest first, self-sufficiency	0/1/2
NB	Rejection of established procedures, rejection of conformity	0/1/2
B	Change to new procedures, difference values, creativity	0/1/2
PB	Friendship, mutual pleasure, recreation	0/1/2
DP	Trust in the goodness of others	0/1/2
DPF	Dedication, faithfulness, loyalty to the organization	0/1/2
DF	Obedience to the chain of command, complying with authority	0/1/2
DNF	Self-sacrifice if necessary to reach organizational goals	0/1/2
DN	Passive rejection of popularity, going it alone	0/1/2
DNB	Admission of failure, withdrawal of effort	0/1/2
DB	Passive non-cooperation with authority	0/1/2
DPB	Quiet contentment, taking it easy	0/1/2
D	Giving up personal needs and desires, passivity	0/1/2

Table 2.1: SYMLOG Survey: The questionnaire utilized to gauge the personality of an individual through peer review (codes are shown here for reference). The response had a value of Rarely (0), Sometimes (1), and Often (2) in the original SYMLOG survey used by SCG [2].

These scores are then aggregated to a single 3 dimensional point for each individual on the dimensions P/N, F/B, and U/D. The method used to achieve this is for each participant is:

1. Obtain the average score for each question from all the raters (see Table 2.2)
2. For each pole (P,N,F,B,U,D) sum up the scores (see Table 2.3):
    - Find all the questions that have that pole's letter in their code (UPF and UNF for instance both have U)
    - Sum up the scores across the relevant questions (9 for each pole)
3. For each dimension, take the difference between the opposing poles. For the P/N dimension for instance you would take the difference between P and N to obtain the final position for the individual along that axis (see Table 2.4)

An example of this method can be found in Tables 2.2, 2.3, and 2.4.

Once these datapoints are attained, they are plotted in the 3 dimensional space with the X,Y,Z axes being P/N, F/B, U/D. The Z dimension is represented by the size of the circle. In order to interpret the meaning of these datapoints with relation to one another, a set of referential markers (two large circles and a bi-directional arrow going through the center) is placed over top of this plot. This set of referential markers is known as the Overlay (see Fig 2.2).

The Overlay is essentially the social psychological lens through which the personalities of a group are interpreted. The Overlay sits at the centre of the plot (or close to it). While Bales states that no fit of the Overlay is "absolute", he does mention that good tentative fit is to move it about the origin until the major clusters reside in one or both of the large circles [2]. The line of polarization in a good fit typically passes through the wish personality and the reject personality which incidentally tend to be on the exact opposite side of the plot. These trends lead to an Overlay fit in which the line of polarization passes through (or close to) the origin.

The Overlay divides the plot into five distinct regions that have implications on those found within each region (see Fig 2.2 for a visualization). The *Inner Circle* is typically where the core contributors are found. These are the people that are the most productive in the group and act as an anchor for all the rest. In Fig 2.2 Steve and Anne are in this region. Since these personalities are so tightly clustered they are purported to have a strong psychological connection to one another [2].

SYMLOG Peer Ratings for Fred

Code	Colleague 1	Colleague 2	Colleague 3	Average
u	0	0	1	0.333
up	0	0	2	0.667
upf	0	1	2	1.000
uf	1	1	1	1.000
unf	0	1	2	1.000
un	2	1	1	1.333
unb	2	1	1	1.333
ub	1	2	1	1.333
upb	1	0	0	0.333
p	2	2	1	1.667
pf	0	0	2	0.667
f	2	0	2	1.333
nf	0	0	1	0.333
n	2	0	2	1.333
nb	2	0	2	1.333
b	0	1	0	0.333
pb	1	0	1	0.667
dp	2	0	1	1.000
dpf	1	2	1	1.333
df	2	2	2	2.000
dnf	2	1	2	1.667
dn	1	2	1	1.333
dnb	2	2	2	2.000
db	1	2	1	1.333
dpb	2	0	1	1.000
d	1	1	1	1.000

Table 2.2: SYMLOG example ratings for Fred

Intermediate Calculations for Fred

Dimension	Formula	Result
P	up + upf + upb + p + pf + pb + dp + dpf + dpb = 0.667 + 1.0 + 0.333 + 1.667 + 0.667 + 0.667 + 1.0 + 1.333 + 1.0	8.334
N	unf + un + unb + nf + n + nb + dnf + dn + dnb = 1.0 + 1.333 + 1.333 + 0.333 + 1.333 + 1.333 + 1.667 + 1.333 + 2.0	11.665
U	u + up + upf + uf + unf + un + unb + ub + upb = 0.333 + 0.667 + 1.0 + 1.0 + 1.0 + 1.333 + 1.333 + 1.333 + 0.333	8.332
D	dp + dpf + df + dnf + dn + dnb + db + dpb + d = 1.0 + 1.333 + 2.0 + 1.667 + 1.333 + 2.0 + 1.333 + 1.0 + 1.0	12.666
F	upf + uf + unf + pf + f + nf + dpf + df + dnf = 1.0 + 1.0 + 1.0 + 0.667 + 1.333 + 0.333 + 1.333 + 2.0 + 1.667	10.333
B	unb + ub + upb + nb + b + pb + dnb + db + dpb = 1.333 + 1.333 + 0.333 + 1.333 + 0.333 + 0.667 + 2.0 + 1.333 + 1.0	9.665

Table 2.3: SYMLOG example intermediate calculations for Fred

Final SYMLOG Profile for Fred

Axis	Formula	Result
P/N	P - N = 8.334 - 11.665	-3.331
U/D	U - D = 8.332 - 12.666	-4.334
F/B	F - B = 10.333 - 9.665	0.668

Table 2.4: SYMLOG example final personality for Fred

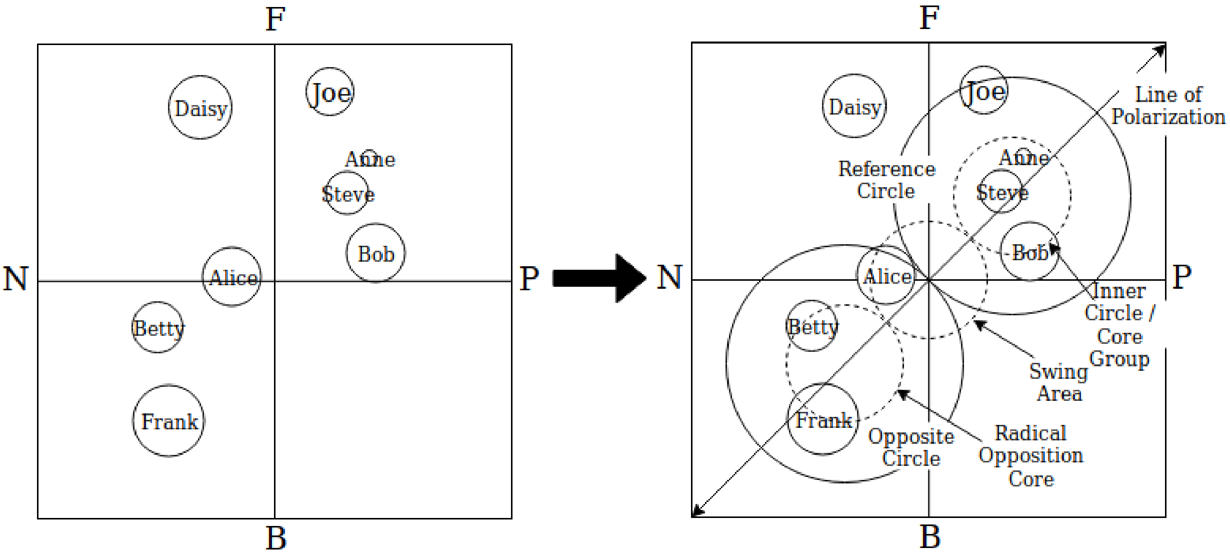

Figure 2.2: SYMLOG Personalities with Overlay (the size of the circle corresponds to the U/D dimension)

Moving outward there is the *Reference Circle*. In a productive team this region is typically where the rest of the team can be found. In our example Steve and Anne are also in this region as well as Bob and Joe. These individuals still contribute to the group but are far enough away that they only experience a mild attraction to those in the Inner Circle. Should conflicts arise between individuals on opposite sides of the Reference Circle, a mediator from the Inner Circle is typically required to resolve the conflict. Opposite the Reference Circle there is the *Opposite Circle*. In this circle are housed the degenerates. Frank, Betty, and Alice are those individuals in our example. Typically individuals found here are counter-productive to the efforts of the group. Conflict between these individuals and those in the Reference circle is frequent and difficult to resolve. The *Radical Opposition Core* is the inner core of the *Opposite Circle*. In our example, only Betty resides here. These individuals reject those who are friendly, group-centric, or equalitarian. These are the rebels against the status-quo. The *Swing Area* is located between the Reference Circle and the Opposite Circle. Here essentially lies the outcasts. Alice can be found here. Individuals in this region often times chose to align with the Reference Circle individuals for some issues and with the Opposite Circle group for others. These individuals have the potential to become either mediators or scapegoats [2].

Bales proposes that the most productive teams have most (if not all) of their members in the reference circle and that the overlay ends up being oriented such that the reference circle gets placed in the PF sector. Bales also dictates that in this ideal situation that nobody should be in the Opposite Circle region (which would be the NB region in this case). Bales proposes that the NB region is typically the most unproductive space for any individual to reside.

For the purposes of completeness, Bales provides an explanation for each conceivable position that a persona can reside (see Fig 2.3). These regions are:

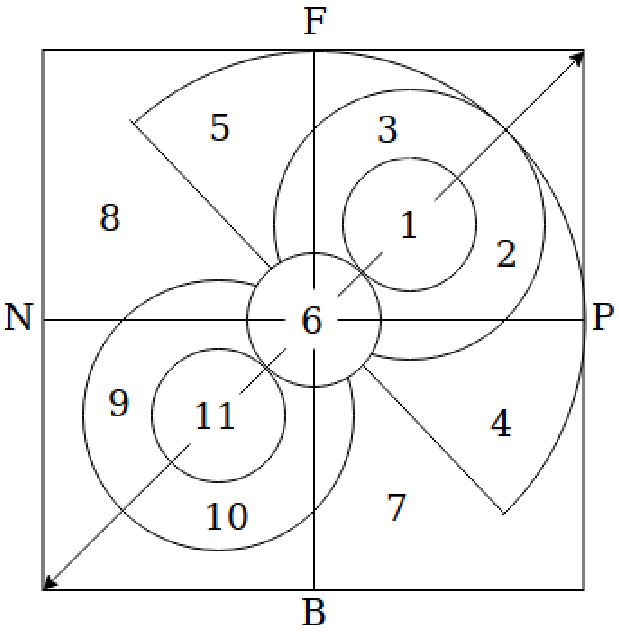

Figure 2.3: Comprehensive SYMLOG Overlay Regions

1. Most Effective Teamwork Core: This is where the leaders of the team reside. These are the individuals that are the most productive and keep the team together. It is chosen to be 9 units in diameter as the maximum distance for mutual acceptance [2].

2. Liberal Teamwork Side: Productive members of the team that may have some conflict with those in area 3.

3. Conservative Teamwork Side: Likewise, productive members that may require occasional mediation.

4. Group-centered Wing: A more relaxed, but friendly fringe.

5. Authority-centered Wing: A more authority centered fringe.
6. Swing Area: Individuals here may vote with the core members on some issues and with opposition members on other issues.
7. Libertarian Fringe: Relatively unproductive members that favour working less.
8. Individualistic Fringe: Unproductive members that are focused on authority and their own needs.
9. Anti-group Opposition: An area of active conflict against the core group that are relatively negative.
10. Anti-authority Opposition: The most resistant to authority can be found here.
11. Radical Opposition Core: The most resistant area to the efforts of the group as a whole. Members here offer some mild rebellion leadership (although not as strong as those in the core group) and resist the efforts of the most effective teamwork core.

2.2.1 SYMLOG in Online Environments

While a number of studies have used various personality paradigms to attain personas from online content, one study in particular done by Berdun et al. utilized an online cooperative game to attain the situational behaviour of each player to estimate their personality [4]. Each of the 98 participants did a traditional SYMLOG questionnaire to establish a ground truth. Then participants played a Lord of the Rings board game simulated in an online environment. Each of their actions was mapped to a SYMLOG direction and an aggregated profile was developed over time for each player. This study was able to accurately attain the profiles of participants along the P/N and U/D SYMLOG dimensions thus setting a precedent for applying SYMLOG to online group interactions.

2.2.2 Competing Group Analysis Models

Affect Control Theory (ACT) is a complementary model of behaviour that can be extended to group dynamics. ACT is an affective framework that, like SYMLOG, is based on 3 axes. For ACT these axes are Evaluation, Potency, and Activity (EPA) [30]. Evaluation is essentially a person's assessment of whether something is good or bad (similar to SYMLOG's Positive vs Negative dimension), Potency is an individual's perception of an object's power (not too dissimilar from SYMLOG's Dominance vs Submissiveness), and Activity measures how active a thing or person is. The latter is the only axis that does not fit particularly well with the SYMLOG space.

ACT models a situation in which there are two individuals interacting with one another. Each individual is assigned an identity that will be used in the calculations. To see how one of these individuals "feels" about an interaction, you give one of these agents (known as the *actor*) an action to perform. Every action has an EPA vector that denotes the sentimental meaning of that action. If one agent performs an action that is unusual given its prescribed identity, then the recipient of that action (known as the *object* of the interaction) feels discomfort. This discomfort can be measured as the Euclidean distance between the EPA of the performing agent and the EPA of the action performed. This difference is known as *deflection*. ACT proposes that individuals tend to act in such a way as to minimize this deflection. This results in a scenario in which people behave in the manner that is expected of them. As such, ACT can do a reasonable job at predicting the success of individual interactions but suffers from an issue of scalability.

GroupSimulator developed by David R. Heise was an innovative attempt to merge ACT and SYMLOG. This piece of software utilizes the IPA actions that are the conceptual foundation upon which SYMLOG was built. It then maps these 12 actions to the EPA space (through the use of affective dictionaries). Then the group is modelled according to a set of user provided configurations to determine actor and object selection protocols [17]. These protocols include selecting:

- the agent that has the potential to produce the least amount of deflection
- the agent that is feeling the greatest amount of deflection
- the agent that is feeling the least amount of deflection
- the agent that is evaluated by the group as the most esteemed (the E of EPA)
- the agent that is evaluated by the group as the most potent
- the agent that is evaluated by the group as the most active
- the agent that is evaluated the most highly along a specified SYMLOG dimension (one of the 26 directions in the SYMLOG space)
- a random agent

Additionally, the group itself is included in the model as an agent. As such, actors can select the "group" as an object to act upon. Any action taken towards the group is simulated as that action being taken on each actor in turn.

The flow of the simulation is as follows:

1. Select an actor and object based on user specified criterion
2. Calculate the optimum action to take (determined as the action that will minimize deflections of the actor and object according to traditional ACT calculations)
3. Proceed with ACT evaluations to deduce the transient impressions of each actor involved
4. Override the transient impressions of the agents involved to the impressions produced by the current event
5. Repeat

For example, if a user opts to configure the simulator such that:

- the actor is the agent that feels the greatest deflection
- the object is the agent that has the greatest potential to produce the least amount of deflection

with three agents of prescribed identities:

1. student
2. professor
3. teacher

If a situation arises such that the student is yelling at the group (which is simulated as the student yelling at the professor and the student yelling at the teacher), then a great deal of deflection is created (as this is a rather surprising event). The professor is a highly esteemed individual that would not expect to be yelled at by a student (even more so than a teacher), thus feels more deflection than the teacher does and as a result, will be selected (based on the given configuration) for the next iteration. To resolve this conflict, the professor would choose the student as the object of interaction, as this is the individual that is acting most uncharacteristically and hence has the greatest potential to reduce deflection if the group's impression of him can be "corrected". In order to restore the impressions of the agents involved back to what would be expected of the prescribed identities, the professor may choose to "rebuke" the student for acting inappropriately. This would reduce deflection of the professor and the student as the professor is restoring his status over the student. The simulation runs until the user decides to end it.

This simulation can be run to get an estimate regarding what actions will be taken in the group and by whom. However, there is no specific "successful" combination of actions and thus falls short of predicting the success or failure of the group as a whole.

An extension to the ACT model was the ACT-S. This variant of ACT includes the notion of a self that is selected dynamically given a situation as opposed to being set in advance [16]. The "true" self is a vector average of the calculated self-sentiments generated with every interaction. The distance between this true self and the current situational self-sentiment is known as inauthenticity. Instead of fixing a self-identity at the start of a scenario, the identity displayed by an actor is selected such that it will minimize feelings of inauthenticity in the agent (in other words provide it with an opportunity to act in a way that is in-line with its "true" self).

BayesACT, another extension of ACT, is a dyadic interaction framework. As an extension of ACT, it too takes into account deflection. However, it also takes into account the variability of the sentiments (how universal one word or personality is compared to another) by utilizing Bayesian statistics to model uncertainty [34]. In this way, BayesACT can model how the identities of individuals are formed over time by acting in a probabilistic fashion (as opposed to the deterministic fashion of ACT). This captures a more realistic process of identity formation.

BayesACT-S, like ACT-S, is an extension of its predecessor that adds the notion of a dynamically fluctuating self-identity. The difference between ACT-S and BayesACT-S is that while ACT-S uses a single vector average to represent the fundamental self, BayesACT-S instead uses a multimodal distribution to represent the fundamental self of an agent [19]. This allows for greater flexibility as an agent can experience multiple identities at once in a more detailed way.

2.3 GitHub as an Open Source Collaborative Environment

GitHub is an online collaborative coding platform based on the Git framework. In this paradigm, any user can copy the codebase to a local machine and make changes to it there. If this user wishes to share these changes to the community, then they can propose these changes in the form of a *pull request*. This will notify the maintainer (a user with write permissions) of the changes proposed and provides an opportunity to make comments and changes to the pull request before deciding to accept or reject the changes that the pull request proposes.

GitHub boasts hosting over 96 million repositories with over 31 million developers and over 200 million pull requests [13]. It also offers free use for public repositories which are often utilized by open-source developers. On this platform, each pull request automatically starts a publicly available conversation in which everybody in the project can participate. The messages exchanged over GitHub for these public repositories are also publicly available thus making it an ideal source of research data for those interested in this cohort.

Social Factors in GitHub

Social factors play an influential role in the dynamics of their respective GitHub repositories. GitHub's Open Source Survey showed that 5 of the top 6 issues in open source projects are social. These issues are unresponsiveness, dismissive responses, conversational conflict, unexplained rejection, and unwelcoming language and content [12].

Historically, it has been shown that more than technical factors influence the acceptance or rejection of a pull request [39]. It has even been shown that a complex social structure exists around any given pull request [9]. Additionally, social connections play a stronger role in the acceptance of a pull request than technical factors do [39]. If one were to focus on just the technical qualities of a pull request, and ignore the social context surrounding it, there is a strong likelihood that such a pull request would be rejected [9].

While one study on closure rates reported that desirable social network characteristics did not have a significant effect on closure rates, the same researchers also conceded that they did have a significant impact on the number of commits [23]. One caveat that they provide was that the social features were highly inter-correlated, thus making it difficult to discern which were impactful. When modeling for pull requests, it was found that the strongest factor in the model was the social distance between the pull requester and the project owner (does the pull requester follow the maintainer/reviewer) [39]. This could be an indication of trust between the requester and the reviewer that could lower the evaluation cost of the pull request for the reviewer [39].

Other trends include number of comments, social network density, and prior interaction. Ducheneaut found that for every comment made in a pull request conversation, the likelihood of acceptance is reduced by 54.6% (in other words the likelihood would be halved with every comment) [39]. Jarczyk, et al. found that having a highly dense social network for the repository had a negative impact on short-term issue closure rates (defined by Jarczyk as the percentage of issues that were closed within three or fewer days of being created) [23]. However, they also found that long-term issue closure rates (issues closed within 365 days) were unaffected. This led them to the conclusion that extensive discussion did not seem to hurt issue-closure rates of long outstanding issues [23]. So while extensive discussion seems to hurt pull request acceptance it does not seem to hurt long-term issue closure rates. Additionally, for pull requests, having prior interaction within the repository (previously accepted pull requests for instance) was a strong predictor of acceptance (35.6% increase in likelihood per interaction) [39].

Sentiment in GitHub Text

Deriving sentiment from text has been a long-standing challenge. However, techniques such as sentiment analysis or opinion mining have been developed in the last decade to address this challenge [27, 32]. These techniques can be applied to the domain of GitHub in monitoring emotional features. In a study by Jurado, et al. they obtained issue comments for a few popular repositories and analyzed their emotional features. Their method was lexicon-based and utilized affective dictionaries (in particular ANEW) [24]. The packages NLTK was used for text processing (tokenizing words, etc.) and Snowball was used for word stemming. For each issue they did a frequency count of the emotional words (as defined by the WordNet-Affect lexicon). Emotional labels were assigned according to frequented emotions. However, few messages in GitHub issues appeared to contain any kind of sentiment. Despite this, Jurado does still assert that NLP can be applied to developer written text and that this could be utilized in assessing the development process of a repository.

Later work began utilizing more sophisticated machine learning techniques. Rishi utilized SVMs and deep learning in his architecture to improve upon the more rudimentary lexicon-based approaches [33]. He used Mechanical Turk emotional annotations on pull request sentences as his ground truth. Each annotation was one of the IPA categories. 3000 pull request comments from GHTorrent were annotated (as opposed to issue comments which Jurado used) [24]. Rishi did a 1-vs-all classification for each of the emotional labels across 5 folds of validation achieving a max F1 score of 0.64 for the IPA category of *agrees* [33]. Rishi concluded that subjective emotional and social interaction were indeed significant. This conclusion concurs with the work of Jurado (that sentiment can be informative analyzing the software development process) [24]. Rishi also concluded that automated detection poses a considerable challenge to classification tasks. This is an unsurprising conclusion given that Jurado's findings were that the vast majority of comments were void of sentiment [24]. It is conceivable that this would make training any kind of machine learning algorithm challenging.

IBM's Personality Insights service takes the machine learning idea further by scaling up the training set. This Personality Insights is a cloud-based service that allows the client to send text to IBM's servers where it is processed through proprietary algorithms to infer a personality from the textual input [22]. IBM asserts that behaviour/emotion is linked to personality [22] and that text is closely linked with personality [10, 11, 14, 18, 41]. IBM purports that they utilize an open-vocabulary approach (they do not rely on a prebuilt dictionary, but rather create it dynamically [35]) to infer personality from the text provided [22].

With regard to the specific method that Personality Insights utilizes there are some similarities between these works. Personality Insights uses NLTK to tokenize their inputs into a n-dimensional space [22] whereas Jurado used the same library to isolate their words for their lexicon-based approach [24]. In order to represent their words in a vector space Personality-Insights utilizes GloVe [22]. In order to infer the big 5 given this representation they use a neural network (details undisclosed) [22]. Similarly, Rishi also used various deep architectures to predict sentiment [33]. For ground truth, both IBM and Rishi utilized survey data. They then utilized their deep architectures to predict what a survey result for a given input might look like [22, 33]. IBM utilized twitter data while Rishi trained on content from GitHub, IBM involved 1500-2000 participants per predicted characteristic while Rishi utilized 6 participants [22, 33]. For IBM's surveys they used:

- 50-Item Big 5 international Personality item Pool (IPIP)
- 120-Item Facet extended from IPIP with Neuroticism, Extraversion, and Openness (IPIP-NEO)
- 52-Item Fundamental Needs (Developed by IBM)
- 26-Item Basic Values (Developed by Shwartz)

In the literature, correlations above 0.2 between the predicted and actual personalities are considered acceptable performance for such text-to-personality tools which puts Personality-Insights in a comparable position with an average correlation of 0.31 [22]. Other researchers that have created their own textual personality extraction include:

- Golbeck who achieved a 10-18% error with respect to ground truth (Big 5 survey results) with Twitter data [14]

- Sumner who also utilized Twitter data attained 65% accuracy with respect to personality surveys [38]

- Walker who obtained 60-70% accuracy for the Big 5 attributes using the essays of psychology students [28]

Predicting Individual Success

Over the years a number of researchers have looked into the impact that an individual developer can have on others. For these studies various metrics have been used including varying definitions of influence [26, 21], issue closure rates [23], and peer review assessment [2]. These researchers sought to find what makes a good developer and how they influence the team.

HITS and PageRank are two such metrics borrowed from web page assessment for implementation in the social influence space. Traditionally, HITS is a PageRank algorithm that utilizes hubs and authorities. It is presumed that high quality hubs link to high quality authorities and that high quality authorities link to high quality hubs. Thus, the relative importance of any web page can be calculated by a mutually reinforcing algorithm that depends on the quality of the web pages that reference it. Similarly, PageRank postulates that the relative importance of a web page is also dependent on what websites refer to it. The PageRank of a website is calculated by simulating a random walk through the web, the more times a website gets visited in the simulation, the higher its relative importance becomes. Unlike HITS however, PageRank normalizes the weights of these links so that the "votes" that a single page can provide to the overall calculation is the same (if a website references only one website then that "vote" is more impactful than a single reference to that site from a web page that has many references) [8].

Liao et al. crafted an experiment to determine how to predict who will be a "popular" developer and what will be a popular repository in the future [26]. The motivation behind this work was to improve upon previous methods for doing this prediction, primarily HITS and PageRank. For this experiment they defined "popular" as the developer/project's ability to attract followers/stars in the future. HITS utilizes a linked graph and defines the score of a hub as the summation of the scores of the authorities that they refer to and the scores of the authorities as the summation of the scores of the hubs that refer to them thus iteratively converging to some global state of the graph [8]. PageRank utilizes the hyperlink structure of the web to score a web page based on the relative importance of each hyperlink that is referring to it [31]. In this application to GitHub, the nodes were made to be the developers and the links to be the "follows" relationship [26].

Liao et al.proposed that utilizing both HITS and PageRank together in a cooperative fashion could offer improvements over using either one on its own. This new proposed method was named DevRank [26]. In this new method there are two networks. One has the developers as nodes and the follower-followee relationship as the links and each would be ranked through the PageRank algorithm. The second is a bipartite network where the nodes would be both developers and projects and the links would be the "commit" relationship. This second network is evaluated through the HITS method to transfer influence between the project and the developer. In the final implementation the algorithm does a single iteration on the follower network, updating the influence of its the developers, then it does one iteration on the commit network to transfer influence between project and developer, and so on [26]. Using this inclusive approach the authors expected to achieve gains over using either network separately as this approach incorporated the role that repositories play in transferring influence to developers and vice versa [26].

By using GHTorrent to attain data on 1,047,550 developers over 1,320 projects leading up to 2012 Liao, et al. ranked the developers according to their algorithm as well as with HITS and PageRank separately (among a handful of other measures) [26]. Their DevRank method did indeed perform better than any other metric used. DevRank achieved 75% precision in determining which developers (the top 30) would become the most influential over the next year of data (they compared against the data on these developers during 2013) as opposed to the 62% achieved by the next best metric [26]. The correlation between the DevRank score and the number of followers attained in the next year was near perfect. Project influence prediction while still better than the others tested (PageRank, HITS, etc.) attained only 60% precision [26]. Additionally, DevRank converged over fewer iterations than either PageRank or HITS.

Hu's alternative to HITS and PageRank on the other hand, was a following-star-fork-activity model for developer influence [21]. In this model, they define "influence" as the variation ratio (original followee number - current followee number)/(original followee number + current followee number) rather than just the raw count of attained followers in the next year [21, 26]. Hu et al.obtained their data through a GitHub crawler as opposed to

GHTorrent. In addition, they tested a larger number of methods including:

- UserRank
- HITS
- H-Index
- Borda Count

UserRank is a modified PageRank algorithm utilizing the follower-followee relationship [21]. This UserRank is essentially identical to Liao's PageRank algorithm [26]. Likewise, both Liao and Hu implemented HITS using identical methodologies (also follower-followee relationship) [26, 21]. H-Index was implemented as an individual that has created h projects that have been starred or forked (each implemented as its own metric) a minimum of h times. Borda Count is the metric that Hu and colleagues adapted for the GitHub context and is defined as the sum of scores of each evaluative factor. In this context, the evaluative factors are:

- User Activity (number of commits, projects created, number of opened issues, sent pull requests)
- Authority Value Prescribed by HITS
- User Follower Number
- User Repository Forked H-Index
- User Repository Star H-Index
- The User Rank prescribed by UserRank

Based on change ratio between Jan 19, 2017 and Apr 3, 2018 Hu showed that Borda Count does indeed offer the most accurate ranking of the ranking schemes tested (variation ratio of 15.67% compared to the 10.11% of the next best metric) [21].

Alternatively, SYMLOG predicts individual success somewhat indirectly. While SYMLOG is primarily based on group success and the interaction of individuals within a group, they do propose a "best" member for productive teamwork [2]. This member is defined as the SCG optimum (see Table 2.6)

Table 2.5: SCG Optimum Personality

Vector	Score	Vector	Score
U	17	N	7
UP	21	NB	10
UPF	30	B	22
UF	27	PB	22
UNF	20	DP	24
UN	18	DPF	29
UNB	9	DF	24
UB	19	DNF	22
UPB	23	DN	9
P	24	DNB	6
PF	25	DB	3
F	18	DPB	8
NF	18	D	9

Table 2.6: Values range from 0 to 33 [2]

This standard was developed through thousands of surveys and significant deviation from this standard is considered by SCG to be an indicator of concern [2]. This standard is used as a guide for altering individual behaviour for the betterment of the group [2].

Jarczyk et al. investigated individuals through the dynamics of group performance in the developer environment with respect to short and long-term issue closure rates [23]. In particular, he discovered that relying on a select few people led to faster issue closure rates as opposed to a distributed workload, thus emphasizing the impact of individual performance on the group [23]. Additionally, emphasizing individual responsibility by assigning issues to specific individuals was positively correlated with faster closure rates [23].

2.4 Predicting Group Success

2.4.1 Predicting Success

One approach to take in analyzing groups is to gather as much historical data as possible and attempt to utilize that in making predictions about the future. A couple of studies including those done by Coelho, et al. [7] and one done by Borges et al. [5] did exactly that.

Borges and colleagues wanted to use historical GitHub data to predict how popular each repository would become [5]. To do this, they used trends in the star count to predict what the star count would be in a certain number of weeks into the future. For this experiment they used the star count at every week for a given repository as a independent variable in a multiple linear regression model. They fitted their model based on 26 weeks of data and were accurately (mRSE 0.432 ± 0.257, 95% confidence interval) able to predict the star count of a repository 6 months into the future. They validated their test results using 10-fold cross-validation across various week ranges [5].

Instead of doing a trend analysis of a particular feature to predict what that feature would be in the future, Coelho and colleagues wanted to utilize mineable characteristics to predict a ground truth that could not be readily determined through any automated means [7] (as opposed to one that could be simply mined at a later date). In particular, these researchers wanted to automate the process of identifying which repositories were actively maintained and which were not. In order to do any kind of training, they first needed to establish a ground truth. To achieve this they labelled 1002 repositories using various techniques up to and including actually contacting the repository owners to confirm unambiguously what the status of the repository was. They labelled these repositories as either active or inactive and used them to train their models accordingly [7].

For their machine learning models (they used 10 different variations) they utilized a large number of mineable variables. These included number of forks, various issue characteristics, pull request metadata, commit relationships, contributor data, as well as various social network data governing the dynamics of the group, and many others. Once the algorithms were trained on this set of repositories, they ran them on a set of 5,783 other repositories. To validate the results, they surveyed the developers of a subset of these repositories.

After comparing these 10 machine learning algorithms, they found that random forest performed the best. It achieved a kappa of 0.78 (a kappa greater than 0.6 is considered significant [25]) which is a measure of the accuracy taking into consideration the bias of the dataset. This is especially important in this scenario as the algorithms were trained on an unbalanced set (754 active repositories, 248 inactive). Random forest attained a recall of 96% for identifying which repositories were unmaintained.

2.4.2 Key Factors in Successful Groups

Another approach to analyzing the dynamics of a group is to attempt to isolate what characteristics actually cause a group to succeed or fail in an attempt to draw actionable conclusions. Two of these studies were done along this vein, one was done by Jarcyz et al.in studying GitHub groups and their closure rates [23], the other (which actually involved a series of studies) was done through SYMLOG in isolating what makes an effective team [2].

Jarczyk et al. wished to see how various group-centric features could explain the success or failure of a GitHub repository [23]. To achieve this they created a number of GLMs (Generalized Linear Models) to model issue closure rates, stars, and number of commits in an attempt to ascertain which objectively measured aspects of the group explain these dynamics.

For data, they selected approximately 10,000 repositories to study. Each repository needed to be at least 2 years old, have at least 100 commits, and have 5 or more team members. They then selected their feature set to include the Gini coefficient for workload distribution. The Gini coefficient is defined as:

$$G = \frac{\sum_{i=1}^{n} (2i - n - 1) y_i}{n^2 \bar{y}}$$

where n is the total number of project members, y_i is the number of commits for member i and $\bar{y}$ is the average number of commits per member. The smaller the Gini coefficient, the more distributed the workload is. Their features set also included a number of social network characteristics regarding cooperation, discussion, skill similarity, and more. For these social network attributes, they averaged those qualities of the individual members in order to get team-level characteristics (which was the level of interaction they wished to study).

From these experiments, Jarczyk and colleagues derived a number of conclusions. They concluded that desirable social structure generally led to more commits. However, these social features were also highly inter-correlated making it difficult to uncover a single driving factor. The traditional metric of popularity (number of stars) was predictive of faster issue closure rates. They also established that larger team size and greater distribution of workload led to worse long-term closure rates. Additionally, specific assignment of issues to specific developers led to much better closure rates for the team.

SYMLOG approached investigating group dynamics by observing in-person groups and collecting thousands of peer review surveys from successful and unsuccessful teams. While a full consultation is recommended, they do provide a few objective indicators of success for a group:

1. The most effective teams usually have nobody in the Opposite circle (see Fig 2.2)
2. Effective teams usually have a positive correlation between their U, P, and F poles.
3. An optimum fit usually entails:
    - Members to be in the reference circle and for that circle to coincide with the PF quadrant
    - That individuals tend to fit closely to the SCG Optimum Profile on the bar graph

2.5 Summary

Considering the personality models available in the literature, SYMLOG was the paradigm best suited for assessing personality dysfunction in the context of a work group. While SYMLOG has been primarily applied in industry through in-person settings, prior work has successfully used SYMLOG in online environments. GitHub is an online environment that is ideal for researching the effects of interactions amongst developers. Utilizing the ground truth labels of maintenance levels provided by Coelho et al. SYMLOG can be used on GitHub commentary to assess the impact of personality on group performance.

Chapter 3

Data Extraction and Interpretation

3.1 Data Extraction

In order to gauge the impact of personality on the maintainance level of a repository we needed a textual data source of user dialogue to ascertain the personalities of individuals in the context of a GitHub project. To this end, we utilized the GitHub API to extract the conversations for any given user in a project. Once this data stream was set up, we experimented with two methods of personality interpretation: (1) third-party annotation of existing conversations using an established SYMLOG questionnaire with the potential for scaling up to mass dissemination via Mechanical Turk, and (2) an automated tool for textual personality extraction to bypass the manual annotation step altogether.

3.2 Interpretation of the Data

3.2.1 Pilot

In order to assess the impact of personality on performance, we first endeavoured to utilize human capital in the interpretation of GitHub comments to best address the potential ambiguity in the data (for example, assessing emotional sentiment). To attain the personality of the individuals involved for a project we filled out a series of SYMLOG surveys for a subset of the members that were involved in the project. To design for Mechanical Turk (where ideally the task for human annotators should be fairly simple) we simplified the traditional SYMLOG rating scale (0 = rarely, 1 = sometimes, 2 = often) to a binarized version to indicate whether the behaviour was or was not seen in this particular conversation. Over a large enough sample of conversations, the average would approximate the ratio for which each behaviour is exhibited.

To achieve this, two judges [1] annotated three projects by focusing on the top 10 developers by number of commits and choosing ten conversations for each of these developers, totalling 100 conversations per project (where possible). These conversations were then annotated in accordance with the binarized SYMLOG survey. For one project in particular (Oni, https://github.com/onivim/oni), 91 conversations were annotated by both Judges (as not all 10 of the top 10 developers contributed to 10 conversations). Comparing the character assessments submitted by these judges we attained a Linearly Weighted Kappas of 0.027, -0.039, and 0.208 for the P/N, F/B, and U/D dimensions respectively. As such, inter-rater agreement was deemed to be low. Additionally, it took approximately 5 minutes to annotate a single conversation for a single person. This meant that for a single project (which would be a single datapoint in our analysis) it would take about 8 hours and 20 minutes for a single skilled annotated to label the data (to annotate 100 conversations). This was prohibitively expensive. Due to this fact and the poor inter-rater agreement, we decided against pursuing the Mechanical Turk avenue of data annotation.

3.2.2 Automation

A more economically viable option was to automatically deduce personas through a machine learning algorithm. To this end we utilized IBM's Personality-Insights cloud service to extract the Big 5 personality from all the comments that a particular individual had made. To assess the validity of the tool, we compared its' evaluation of SYMLOG personas (as attained through an SCG mapping described in section 4) to those derived from the annotations done by our human judges. Collectively the three judges (using Personality-Insights as one of the "judges") obtained average Kappas of 0.09, -0.019, and 0.206 for the P/N, F/B, and U/D dimensions respectively which is not significantly different from what the human judges were able to achieve on their own (0.028, -0.019, 0.208). As such, it was deemed that the Personality-Insights service could achieve a comparable accuracy to that of a human when it came to assessing personality from textual inputs. Thus, we chose to move forward with Personality-Insights in interpreting personality from our extracted GitHub comments.

[1] Alexander Sachs and Undergrad Research Assistant Andrew Li

3.3 Experiment Data and Interpretation

The ground truth for our experiment was the labels from Coelho's *Identifying Unmaintained Projects in GitHub* maintained/unmaintained training data. This dataset had 754 projects that were deemed maintained and 248 that were deemed unmaintained.

Given this list of 1002 repositories, 916 of them still exist (724 maintained, 192 unmaintained). From these repositories we retrieved our raw textual data (issue and review comments) through GitHub's API. We then filtered this data by:

- excluding any comments made after November 2017 (in order to imitate the conditions of Coelho's study)
- excluding individuals whose word count did not exceed 300 (as Personality-Insights cannot deduce a personality for these individuals)

After retrieving and filtering all of the issue and review comments for all repositories with GitHub's API, the raw text was aggregated by project, by individual, and concatenated together chronologically.

The text was then stripped of all markup text and fed into the Personality-Insights cloud service where the Big 5 attributes were attained for each individual. These personalities were transformed into SYMLOG vectors and analyzed at the group level to attain the SYMLOG scores for the repository as a whole (explained in detail in chapter 4). We hoped to see a correlation between SYMLOG's predictions of success (the SYMLOG scores) and Coelho's labels of projects being maintained or unmaintained. The assumption was that "successful" SYMLOG groups would correlate to Coelho's maintained status.

For example, in the Oni project, Akin909 had the following aggregated text from all pull request conversations he was involved with:

```
"Currently image layers pop up whenever a split is made, or rather I observe
this using any plugins that create splits. This PR aims to fix this which I
raised as #1673, using the filter functionality of the add bufferLayer metho
d, which is very cool re api design \ud83d\udc4d.\r\n\r\nI added a filter to
only open the image layer for ‘jpg‘ and ‘png‘ formats (the only 2 that came
to mind at the time) Actually thinking about it rather than having a list de
fined by oni it might be useful to some users to have the list of files the
image layer opened for editable aka via their config or (*not for this PR* a
means of toggling it on or off) the reason I suggest this is that in one sce
nario I know of a user might want to edit the raw file, or process it in som
e way, this can definitely be the case with something like ‘svg‘ but may als
o be the case for some other file format for some other reason @bryphe added
in the config setting for this now \ud83d\udc4d  Currently image layers pop
up whenever a split is made, or rather I observe this using any plugins that
create splits. This PR aims to fix this which I raised as #1673, using the f
ilter functionality of the add bufferLayer method, which is very cool re api
design \ud83d\udc4d.\r\n\r\nI added a filter to only open the image layer fo
r ‘jpg‘ and ‘png‘ formats (the only 2 that came to mind at the time) Actuall
y thinking about it rather than having a list defined by oni it might be use
ful to some users to have the list of files the image layer opened for edita
ble aka via their config or (*not for this PR* a means of toggling it on or
off) the reason I suggest this is that in one scenario I know of a user migh
t want to edit the raw file, or process it in some way, this can definitely
be the case with something like ‘svg‘ but may also be the case for some othe
r file format for some other reason @bryphe added in the config setting for
this now \ud83d\udc4d  @bryphe when you say potentially deprecate the ‘buffe
rs‘ setting does this mean by default all users will have to work with ‘tabs
‘, definitely not wanting to restart what have been many discussions, but it
seems like anyone with a primarily buffer based workflow coming from vim wou
ld only be able to use tabs? \ud83d\ude1e\r\n\r\nMight just wait to see what
the change ends up looking like since I interact very little/never with tabs
\ud83d\ude1f  @bryphe when you say potentially deprecate the ‘buffers‘ setti
ng does this mean by default all users will have to work with ‘tabs‘, defini
tely not wanting to restart what have been many discussions, but it seems li
ke anyone with a primarily buffer based workflow coming from vim would only
be able to use tabs? \ud83d\ude1e\r\n\r\nMight just wait to see what the cha
nge ends up looking like since I interact very little/never with tabs \ud83d
\ude1f  @bryphe when you say potentially deprecate the ‘buffers‘ setting doe
s this mean by default all users will have to work with ‘tabs‘, definitely n
ot wanting to restart what have been many discussions, but it seems like any
one with a primarily buffer based workflow coming from vim would only be abl
e to use tabs? \ud83d\ude1e\r\n\r\nMight just wait to see what the change en
ds up looking like since I interact very little/never with tabs \ud83d\ude1f
Add a 20s expiry time to all notifications that **AREN’T** errors aka info n
otifications and warn notifications, discussed in #1618 \r\n\r\nThis PR will
conflict with #1636 but will push a conflict fixed version depending on whic
```

```
h goes first @CrossR the max width on the notifications is set to 30% of the
available space which since I work primarily of a laptop seemed to always be
squarish but I guess on a hi-res large monitor 30% could easily be enormous
might be worth me using ‘30rem‘ instead which should roughly equate to 30cha
racters wide based on the current font size, can add that to this once I pus
h the merge fixes Updated \ud83d\udc4d, dont actually have a large screen to
hand but 25rem should be a more constant value than the percentage @bryphe m
anaged to get it working with the observables \ud83d\ude04 \ud83c\udf87 and
added the api method as well as removed the component side effect Thanks @br
yphe was nice to get the opportunity to use ‘redux-observable‘, seems I took
down one of the unit tests but just fixed it Add a 20s expiry time to all no
tifications that **AREN’T** errors aka info notifications and warn notificat
ions, discussed in #1618 \r\n\r\nThis PR will conflict with #1636 but will p
ush a conflict fixed version depending on which goes first @CrossR the max w
idth on the notifications is set to 30% of the available space which since I
work primarily of a laptop seemed to always be squarish but I guess on a hi-
res large monitor 30% could easily be enormous might be worth me using ‘30re
m‘ instead which should roughly equate to 30characters wide based on the cur
```

and so on...

This text was then sent to the Personality-Insights, a response JSON was returned, and the extracted traits were:

- Openness: 0.822737151733451
- Conscientiousness: 0.566430416016414
- Extraversion: 0.456057406009756
- Agreeableness: 0.616118627814123
- Neuroticism: 0.487888758636614

From these we obtained the SYMLOG dimensions using the equations from section 4.1:

$$pn = P - N = 9(Openness + Agreeableness) - 9(Conscientiousness + Neuroticism) = 3.461$$

$$fb = F - B = 9(Agreeableness + Conscientiousness) - 9(Neuroticism + Openness) = -1.153$$

$$ud = U - D = [(extraversion - 0.5) \times 2 \times 18] - [0] = -1.582$$

Where pn, fb, and ud are the P/N, F/B, and U/D dimensions respectively.

These SYMLOG personalities form the foundation of our analysis.

3.4 Summary

In this section we laid out the raw features that were used in our data pipeline for the experiments. We showed our initial exploratory pilot study to assess the feasibility of manually annotating conversations with the SYMLOG survey. The shortcomings of this approach was our motivation for creating the automated system that was used for the experiments. We also detailed our ground truth data set that was used for training and testing.

Chapter 4

Methodology

To process the extracted Big 5 attributes, we took the following steps:

1. convert the Big 5 attribute values into SYMLOG values
2. plot out the personalities for each repository and orient the compass for interpretation, and
3. interpret the personalities relative to the compass in order to calculate the various SYMLOG scores for the repository

Each of these steps had a number of viable alternatives that were considered. These alternatives are described in the section below.

4.1 Attaining the SYMLOG Personalities

In order to test the explanatory power of SYMLOG in describing the success of a group, we needed to first attain the personalities to see how they fit into the larger context of the group. Given the Big 5 attribute scores, there were a few options:

1. Use the SCG linear mapping implied by their article [15]
2. Use manifold embedding to map the higher dimensional Big 5 profiles to the lower dimension SYMLOG profiles and use manual annotation as ground truth to discern an interpretable mapping from one space to the other

For the purposes of this thesis, we chose to explore the linear mapping suggested by SCG.

SCG Linear Mapping from Big 5 to SYMLOG

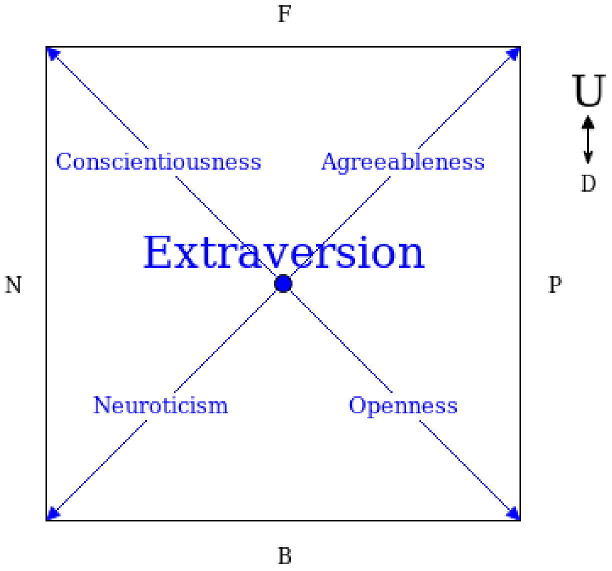

Figure 4.1: SCG Mapping for Relating Big 5 to SYMLOG

In order to decompose a Big 5 attribute into its component SYMLOG parts, we borrowed a trigonometry equation:

$$cos\theta = \frac{Adjacent}{Hypotenuse}$$

For example, consider the F direction as a single vector that is contributed to by both the *Conscientiousness* and *Agreeableness* vectors (see Fig 4.2).

To determine what portion of the Conscientiousness vector is contributing to the F vector we create a triangle between the F vector, the Conscientiousness vector, and a line connecting their terminal points (which would also be the bounding square of the plot in this scenario). Since this forms a right triangle we can utilize $cos\theta = \frac{Adjacent}{Hypotenuse}$ in the form of $cos\phi = \frac{F}{Conscientiousness}$ so then the portion of conscientiousness that contributes to F is:

$$F_C = cos\phi \times Conscientiousness$$

Likewise, the Agreeableness portion will become:

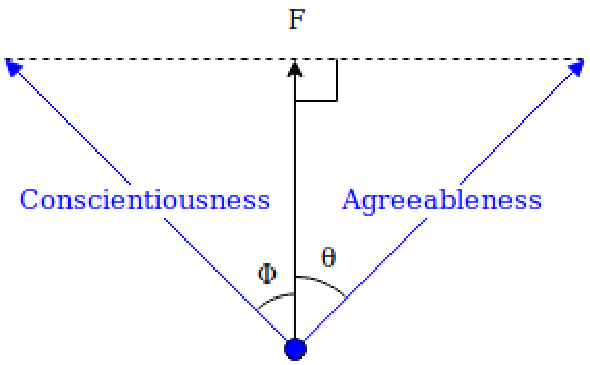

Figure 4.2: SCG Mapping: F Vector Decomposition

$$F_A = cos\theta \times Agreeableness$$

Therefore, F becomes:

$$F_0 = F_A + F_C = \cos\phi \times Agreeableness + \cos\theta \times Conscientiousness$$

In this case, $\phi = \theta = \frac{\pi}{4}$ and $\cos\frac{\pi}{4} = \frac{\sqrt{2}}{2}$. Which simplifies the above expression to:

$$F_0 = \frac{\sqrt{2}}{2} \times (Agreeableness + Conscientiousness)$$

Since each of the Big 5 attributes can attain a value ranging from 0 to 1 and each of the SYMLOG metrics can attain a value from 0 to 18, we need a normalizing constant C in the above expression to scale the value appropriately such that when Agreeableness and Conscientiousness are their maximum value (1), then F too will be at its maximum value (18).

$$F = C\frac{\sqrt{2}}{2} \times (Agreeableness + Conscientiousness)$$

So then C becomes:

$$C = 18/(\frac{\sqrt{2}}{2} \times (1+1)) = 12.728$$

Thus, the final equation for F becomes:

$$F = 9(Agreeableness + Conscientiousness)$$

The SYMLOG vectors P,N,B are calculated in a similar fashion and thus their equation are:

$$N = 9(Conscientiousness + Neuroticism)$$
$$B = 9(Neuroticism + Openness)$$
$$P = 9(Openness + Agreeableness)$$

The one exception to this method, is the U and D poles as the only Big 5 attribute that describes these two poles is the extraversion dimension. In order to calculate these we map the extraversion attribute (range from 0 to 1) onto the U/D axes (which range from -18 (down/submissive) to +18 (up/dominant)) and then attribute the value to the pole that it lies on. In other words:

$$U = \begin{cases} 0 & 0 \leq extraversion < 0.5 \\ (extraversion - 0.5) \times 2 \times 18 & 0.5 \leq extraversion \leq 1 \end{cases}$$

$$D = \begin{cases} (extraversion \times 2) \times 18 & 0 \leq extraversion < 0.5 \\ 0 & 0.5 \leq extraversion \leq 1 \end{cases}$$

Here *extraversion* is the big 5 attribute from 0 to 1, we multiply by 2 here because we separate out the range into two equal sections. Thus we need to scale up the partial measure (ranging from 0 to 0.5) up to the full measure (ranging from 0 to 1) and then again to the full range of the SYMLOG vector (0 to 18).

Validating SCG Mapping

In order to validate that the SCG mapping of the Big 5 space to the SYMLOG space provided a reasonable approximation we manually annotated a single repository and compared this manually derived SYMLOG personality set with those predicted by IBM's Personality Insights. The results indicate little change between using another human as a judge as opposed to using Personality Insights (average pairwise Kappa of 0.07 across the dimensions as an assessment of agreement between two humans versus an average pairwise Kappa of 0.09 when adding Personality Insights as a third judge).

4.2 Orient the Compass

To draw any meaningful conclusion from a plot of SYMLOG personalities, we needed to place the overlay over the plot (as in Fig 2.2). This needed to be done in order to see how groups of individuals are positioned relative to each other and relative to the Overlay. This was done to see what forces could conceivably be acting upon them. To place this compass over the plot, there were a few logical options:

1. Use Bale's ideal for where a compass *should* line up in an optimal group (then measure all the success metrics relative to this orientation, thus deviance from this ideal would lead to worse scores)
2. Utilize Bale's idea of a good tentative fit (a placement such that most of the members are encompassed in the overlay) by doing a linear regression of the personalities in the SYMLOG space (as a heuristic for a reasonable fit) and then align the line of polarization along this line of best fit. Then the reference circle can be selected in one of two ways:
   (a) choose the circle that is closest to Bale's ideal region (PF) as the reference circle. In this method we implement the most productive interpretation.
   (b) choose the circle that contains the highest aggregate level of dominance as the reference circle. In this method we implement the most influential interpretation as Bales' interpretations propose that dominant individuals tend to attain positions of power (increased chance of becoming a leader and/or mediator).

In this thesis, we compared all options.

4.2.1 Bale's Ideal Group Compass Placement

According to Bales the best groups typically end up with a compass placement of best fit that situates itself over the PF quadrant (for the Reference Circle). Thus we can synthesize a score that approximates how closely a sample group conforms to this "best" group by placing the reference side of the compass at the center of this ideal region and then measuring our metrics from this idealized norm.

4.2.2 Regression Line Compass Placement

In the SYMLOG guide, it is recommended to place the compass along the plane of the P/N-F/B axes such that the majority of the members coincide with the Reference and Opposite Circle [2]. In order to approximate this placement of best fit in an automated fashion we can do a regression analysis along the P/N-F/B plain (we essentially just ignore the dominance/submissiveness portion of the SYMLOG personality). This will provide us with a fit for the compass that minimizes the deviation from the center of the Reference and Opposite circles (as these are located along the line of polarization which will be aligned with this regression line),

4.2.3 Calculating the Repository's SYMLOG Score

In order to do meaningful experiments explicit scoring procedures were defined. For SYMLOG there were a variety of interpretations that could be implemented and tested to see which provided the best explanation for the data. These scoring schemes were:

1. Use the SCG optimum profile for a team member as the idealized average for the group (deviance from this norm should be predictive of failure)
2. The positive intercorrelation of U, P, and F directions (higher positive intercorrelation should be indicative of success)
3. The proportion of members that attain a PF-type personality
4. The proportion of the team members that end up being encompassed by the Opposition Circle (higher proportions should be indicative of failure)
5. The degree to which the actual compass approximates Bale's placement of an ideal group (Regression Line Compass Placement only)

For the first metric, the SCG optimum profile is an established benchmark that has been established over thousands of surveys and used in the industry as an indicator for what team members need to change and in what way [2]. As such, variance using this idealized norm as an average can give us a measure of dissidence that should be predictive of unproductive practices. The exact formula used in this experiment is:

$$dissidence = \frac{\sum_{i=1}^{n} \sqrt{(pn_i - pn_{ideal})^2 + (fb_i - fb_{ideal})^2 + (ud_i + ud_{ideal})^2}}{n-1}$$

Where n denotes the number of members in this particular repository, pn, fb, and ud represent the P/N, F/B, and U/D axes respectively, and *ideal* denotes the specific personality described by Table 2.6.

This *ideal* identified in the formula above is an aggregated form of the personality described in Table 2.6. Ideally, we would want to take the variance over all of the 26-dimensions as Bales emphasized the importance of such granularity [2]. Unfortunately, the SCG mapping that we have used to convert the Big 5 attributes to the SYMLOG attributes does not afford us this luxury. This mapping only converts to the pole level of the SYMLOG vector space (P,N,F,B,U, and D). As such, we use this aggregated approximation to get a proxy to what the "real" dissidence would be.

The second metric, likewise is an observation from Bales. He proposes that the U, P, and F directions should be positively correlated with one another in productive groups and that orthogonality is actually a significant indicator of instability in the group [2]. As such, for our measure we used:

$$upf\_corr = \frac{Corr(U,P) + Corr(U,F) + Corr(P,F)}{3}$$

Where $Corr$ is the Pearson R statistic between the two provided poles. The hypothesis here is that a higher upf_corr should be indicative of higher performance.

Similarly, the third metric is derived from Bales' observation that most productive teams tend to have their members aggregated in the PF region of the plot. As such, using the proportion of members that are located in this region as a metric can be indicative of success (with a higher proportion having the potential to lead to productive groups). In this context:

$$pf\_prop = \frac{number\ of\ members\ in\ the\ PF\ quadrant}{total\ number\ of\ members}$$

Conversely, the fourth metric is a simplistic heuristic based on Bales' observation that most productive groups tend to have no members in the Opposite Circle. Thus, this can be indicative of failure if there are individuals in this region. In this context:

$$opp\_prop = \frac{number\ of\ members\ in\ the\ NB\ quadrant}{total\ number\ of\ members}$$

Where a higher score here would be predictive of failure.

The fifth metric is based on Bales' observation that most productive teams (when analyzed through the SYMLOG process), tend to have their compass placed about the origin such that their reference circle line up over the PF region of the plot. Hence, using deviation from this idealized rotation as a metric can be indicative of issues within the group. As such, the degree difference between the actual placed compass and this idealized version, can be used as a metric predictive of success (the smaller the degree, the greater the chance of success).

$$rot\_regret = \mid \theta_{actual} - \theta_{ideal} \mid$$

Where a higher *rot_regret* would be indicative of failure.

Metrics included for the purposes of completeness (but were not particularly emphasized by Bales) are:

- One metric for each of the 11 regions in Fig 2.3 where the metric is calculated as number of members in that particular region divided by the total number of members
- 0th, 25th, 50th, 75th, and 100th percentile for each of the 6 poles totalling an additional 35 metrics

4.2.4 Predictions

These metrics were compared against the labels of the 916 existing repositories provided by Coelho et al. with "success" being attributed to an *active* project, while "failure" was attributed to the *inactive* label. Each of the five SYMLOG metrics presented above as well as the raw Big 5 group scores (as determined by averaging their members) were analyzed through a set of models to see which interpretation of the GitHub repository members was predictive of whether or not that project was actively maintained.

4.3 Summary

In this chapter we discussed how the SYMLOG profiles were attained from the Big 5 personas. We detailed the results of our validation experiments regarding the IBM Personality-Insights tools. These results indicated that the tool had similar performance to that of a human judge for this task. We also showed how Bales' observations regarding SYMLOG were interpreted and quantified as input features for the purposes of the experiments in this thesis.

Chapter 5

Experiments

5.1 Models

We used a number of standard models using Python 3 to test the validity of using group personality to predict if a repository was maintained or unmaintained. For the purposes of classification a 0 corresponded to the maintained status while a 1 corresponded to the unmaintained status. The models we used were:

1. Linear: A simple multiple-linear regression from sklearn.linear_model.LinearRegression where a threshold t is chosen. If the predicted value y is less than t then a 0 is considered to be the classification, otherwise a 1 is the classification.

2. Logistic: A logistic regression from sklearn.linear_model.LogisticRegression(solver = 'liblinear', multi_class = 'ovr'). A logistic regression was chosen as it was a model better suited for binary classification than simple linear regression. *liblinear* was chosen as it is the most flexible type of solver. *ovr* or one-vs-all was chosen as we are only required to make a binary classification.

3. Xgboost: A model that balanced the use of multiple decision trees. This model was implemented with the xgboost package with a $max\_depth = 5$, an objective function of "binary:logistic", and a *eta* of 1 (also known as the learning rate).

4. SVM: A statistical machine learning model designed to be flexible and can be used with smaller datasets. This model was implemented with sklearn.svm.SVC(kernel = 'rbf', gamma = 'scale', probability = True, random_state = RANDOM_SEED).

The “rbf” kernel was chosen here as it is the most accurate of the options. Gamma was set to “scale” which sets the gamma to 1 / (number of features X variance of features). The higher the gamma, the less impact each individual training case has on the learned function. As such, this scaled gamma offers a reasonable compromise between overfitting and generalizability. The probability was set to True so that we would receive a probability from 0 to 1 that we could then interpret using our own threshold into a binary classification.

5. Neural Network: A simple neural network was implemented through PyTorch. This network had the same number of in-nodes as the number of input metrics, it has 5 hidden neurons, and a single output neuron.

All models utilized a random seed (either explicitly through the given arguments or through environment settings) to allow for replicability.

5.2 Training

To train each model for each experiment the dataset was initially divided into 11 equal sets such that each set had approximately the same number of positive and negative samples. As to which set a particular sample was assigned to was determined by a random seed. The first 10 sets were used for 10-fold cross-validation. The probabilities resulting from these training sets were then collected and plotted into a PRC (Precision Recall Curve). From this PRC, a best threshold was determined by measuring the Euclidean distance from each point to the reference point (1,1: a perfect model). With this best threshold determined, the 11th set was then utilized for a final test of the model using this threshold. This process was then repeated 10 more times so that each set was used once as the final test set.

5.3 Results

To see the impact of just the SYMLOG metrics that we defined we trained each of our models on just those metrics. The resulting PRCs can be found in Fig 5.1 (where AUC is the area under the curve).

The best F1 score average achieved with these models on the final test runs was with Xgboost at 0.52. To see if the features that Bales specified as the most impactful (rot_regret,

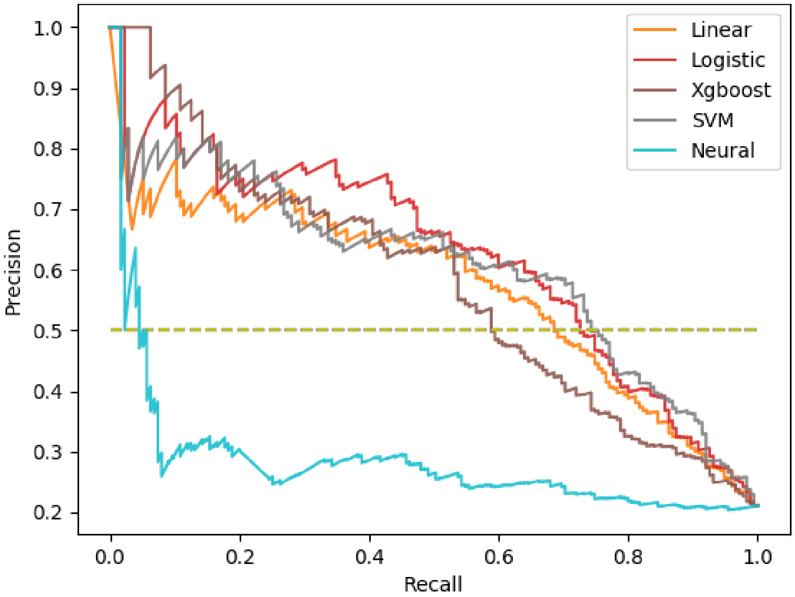

Figure 5.1: Best AUC PRCs for SYMLOG with multiple models

Model	Average F1	Average Precision	Average Recall
Linear	0.445	0.683	0.341
Logistic	0.450	0.723	0.341
Xgboost	0.523	0.605	0.479
SVM	0.410	0.680	0.305
Neural	0.000	0.000	0.000

Table 5.1: Averages for the final test runs based on the PRCs for SYMLOG with multiple models

opp_prop, *pf_prop*, *upf_corr*, and *dissidence*) were significant, we used all the features (including the proportion of individuals in each of the 11 regions as can be seen in Fig 2.3) and compared that against excluding them from the model (see Fig 5.2). When we exclude the factors that Bales emphasizes as imperative, our average F1 score dropped from 0.52 to 0.47.

To see how SYMLOG in general compares against using metrics from other personality

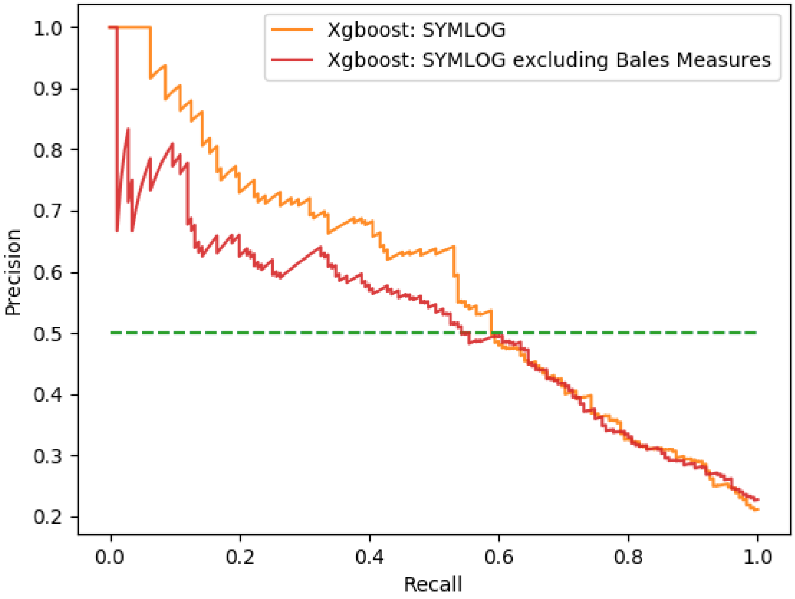

Figure 5.2: Best AUC PRCs for SYMLOG with and without Bales' preferred measures

Model	Average F1	Average Precision	Average Recall
Xgboost: SYMLOG	0.523	0.605	0.479
Xgboost: SYMLOG excluding Bales Measures	0.476	0.558	0.421

Table 5.2: Averages for the final test runs based on the PRCs for SYMLOG with and without Bales' preferred measures

paradigms, we looked at the influence of the Big 5 attributes without the SYMLOG interpretations to see what value the SYMLOG interpretations add. To visualize the difference, we plotted the PRCs of the Xgboost model (as this was the best model for SYMLOG and still a top-performer for Big 5 (see Fig 5.3)) for the median Big 5 attributes, the percentiles of the Big 5 (0th, 25th, 50th, 75th, 100th percentiles to capture distribution information), and the model with all variables mentioned so far (including SYMLOG).

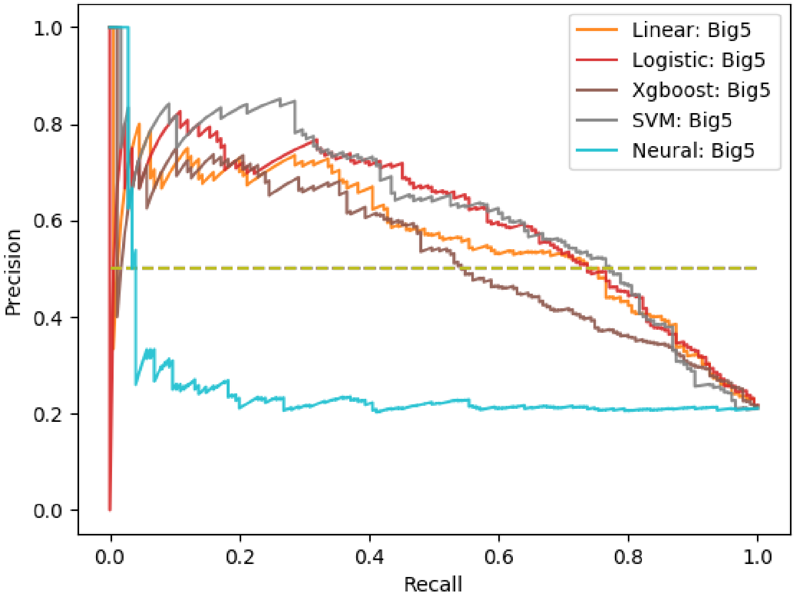

Figure 5.3: Best AUC PRCs for the Big 5

Model	Average F1	Average Precision	Average Recall
Linear: Big5	0.453	0.692	0.352
Logistic: Big5	0.490	0.706	0.388
Xgboost: Big5	0.476	0.554	0.425
SVM: Big5	0.488	0.705	0.384
Neural: Big5	0.022	0.039	0.015

Table 5.3: Averages for the final test runs based on the PRCs for Big 5 for various models

The results can be seen in Fig 5.4. Looking at the final test runs we saw that using all of the personality information from both paradigms achieves an average F1 of 0.52 while excluding the SYMLOG metrics lowers that F1 to 0.48. This difference of 0.04 indicates that Bales' group-level indicators do indeed provide predictive information that the naive Big 5 group-level aggregations do not include.

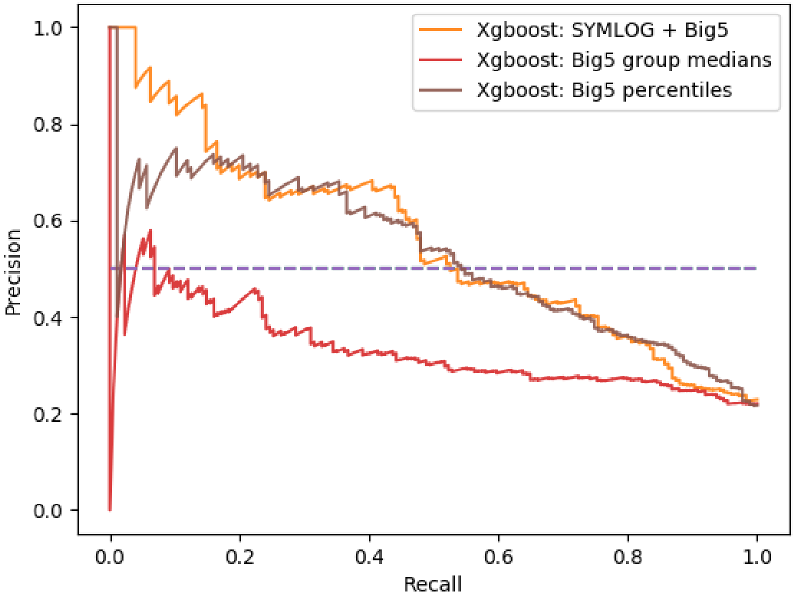

Figure 5.4: Best AUC PRCs for SYMLOG and Big 5

Model	Average F1	Average Precision	Average Recall
Xgboost: SYMLOG + Big5	0.515	0.606	0.463
Xgboost: Big5 group medians	0.277	0.341	0.239
Xgboost: Big5 percentiles	0.476	0.554	0.425

Table 5.4: Averages for the final test runs based on the PRCs for SYMLOG and Big 5

The next stage of the analysis, was to test if personality data provided useful information to the data that Coelho, et al. used in their experiments (pull requests, number of commits, etc.). Looking at only using Coelho's data in our models we achieved our best average F1 score of 0.79 with the Xgboost model. The testing results can be seen in Fig 5.5 for all of the models we utilized.

Going forward with the Xgboost model, we compared how Coelho's data fared on its own and how it did with the addition of SYMLOG (the results of the training can be seen in Fig 5.6).

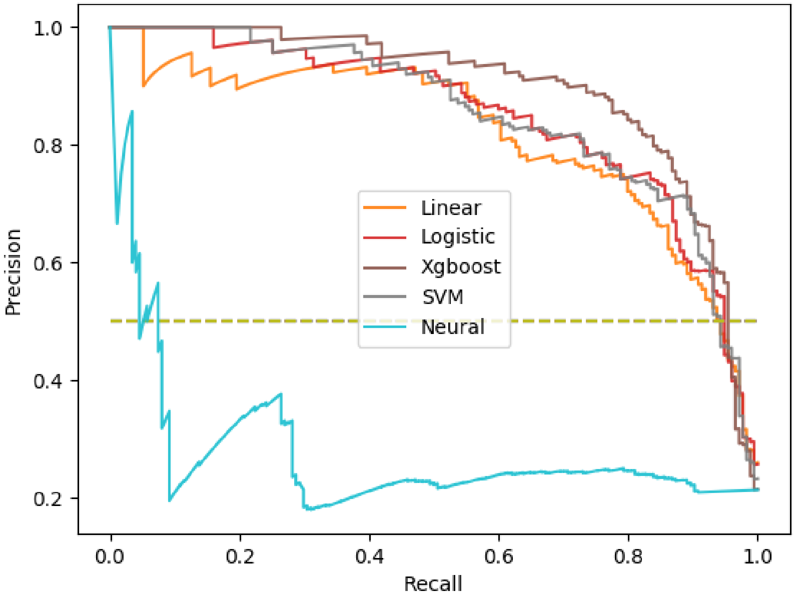

Figure 5.5: Best AUC PRCs for Coelho's training data with multiple models

Model	Average F1	Average Precision	Average Recall
Linear	0.749	0.767	0.734
Logistic	0.752	0.807	0.708
Xgboost	0.795	0.826	0.771
SVM	0.750	0.809	0.703
Neural	0.000	0.000	0.000

Table 5.5: Averages for the final test runs based on the PRCs for Coelho's training data with multiple models

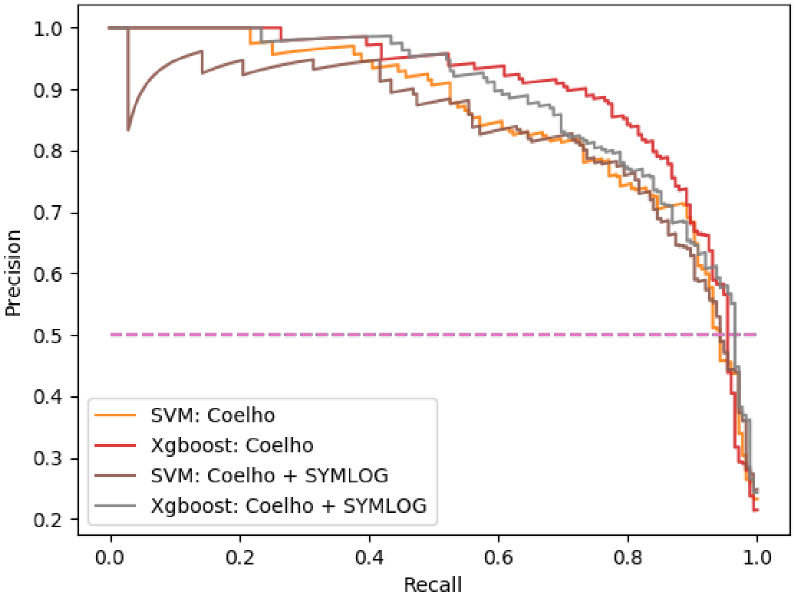

Figure 5.6: Best AUC PRCs for Coelho's training data with and without SYMLOG metrics

Model	Average F1	Average Precision	Average Recall
SVM: Coelho	0.750	0.809	0.703
Xgboost: Coelho	0.795	0.826	0.771
SVM: Coelho + SYMLOG	0.765	0.798	0.739
Xgboost: Coelho + SYMLOG	0.778	0.817	0.755

Table 5.6: Averages for the final test runs based on the PRCs for Coelho's training data with and without SYMLOG metrics

Using the best model for both Coelho's data and SYMLOG (Xgboost) we saw a minor decline in the average F1 score from 0.79 to 0.78. However, the Mann-Whitney test showed this difference to be insignificant (produced labels attained a P value of 0.477 Which is a great deal higher than the accepted statistically significant value of 0.05, see Table 34).

Chapter 6

Discussion and Conclusion

6.1 Discussion

Plotting out the SYMLOG personalities for each project, we realized that the majority of the personas were placed in the PF region between the origin of the plot and the center of the PF region.

Looking at just the SYMLOG personality metrics and their ability to predict the success of a project we can see that intentionally excluding the measures that Bales highlights as important induces a moderate drop in performance (as can be seen in Fig 5.2) and can be seen in the F1 test scores of 0.52 and 0.48 with and without Bales' imperative metrics respectively.

Looking at just the Big 5 metrics, we saw that using just the medians is a oversimplification of the personality data (as can be seen by its comparatively abysmal training performance in Fig 5.4). However, when we take an approximation of the distribution of Big 5 personalities (by using percentiles) in the project, we saw a substantial improvement that made the Big 5 metrics competitive with the SYMLOG model (a testing F1 score of 0.48 as opposed to the 0.52 when we include the SYMLOG metrics in the model).

The most substantial test of validity for the personality data was using it in parallel with Coelho's data. Using Coelho's data only, the highest average F1 score came to 0.79 with Xgboost. Adding in all of the personality metrics from both SYMLOG and Big 5 produced an F1 of 0.78, an actual *decline*. These minor differences in F1 scores and the insignificant P value from the Mann-Whitney test (0.477 from Table 34) indicate that we cannot refute the null hypothesis (that personality does not have a consequential impact on the maintenance of a project).

There are a number of possible reasons behind this null result. One possibility is that personality simply does not have a consequential effect on the maintenance done on a project. According to Coelho et al., the most common reason for open source project failure is the creation of a new competitor on GitHub [6]. This is an eventuality that cannot be predicted by personality. Another possibility, is that the emotional expression of individual members is repressed. Of the 10,829 issues annotated across 9 projects by Jurado et al., 81-93% of the issues in each project contained no sentiment whatsoever [24]. One reason for the emotionally reserved nature of GitHub is that it is a dataset in which the observed individuals know that they are being observed and that anything said will become permanent public record. Since a reasonable proportion of individuals use their GitHub profile as a professional development tool to demonstrate their abilities (a.k.a. "self-marketing") they need this profile to be linked to their true identities in order to receive the desired benefits. Self-marketing as defined by Hars, et al. was found in 36.7% of survey respondents [1]. Since many GitHub developers are trying to build a desirable reputation, they are more likely to be reserved in their commentary. Thus, the anonymity aspect typically associated with online commentary does not apply here. Anything written can be traced back to its author.

Another contributor to the emotionally reserved nature of this dataset, is the fact that it is textual. There is a great deal of involuntary communication that simply cannot be conveyed through this mode of communication. A couple of examples include body language and pitch nuance commonly associated with the rich and vibrant communication medium of in-person conversation. The closest substitute to these nuances in textual communications is emojis. Unfortunately these still do not allow for the same granularity of emotional communication that real conversation can offer. As a result of this limitation of textual communication, a null sentiment of a comment is often assumed by the reader. In an identical in-person conversation however, a moderate insight into the emotional state of the speaker could have been gleaned from non-verbal cues. This is what is lost in the reduced dimensionality of textual communication. As such, there is not as much sentiment from which to derive a proper personality and thus proper causality to project success from emotion.

In addition, there was little to no personality variability between the individuals of any given project. This has the potential of making it difficult for a model to discern a causal effect between the data and the result due to the apparent homogeneity between the projects.

Another reason for the null result could be the mitigation of the typical negative effects of undesirable personality traits. In a traditional work environment, an individual is expected to be productive and approachable eight hours a day, five days a week. Under such consistently emotionally demanding conditions it can be difficult to conceal emotional instability. Outbursts from such instability can negatively impact the immediate work environment and hinder productivity. On GitHub however, the flexibility of a volunteer-based, decentralized governance typically afforts its members the convenience of working when they want to work. This has the potential to smooth out the emotional volatility associated with personality conflicts in a group.

Any individual at any time can take a break from an emotionally charged situation (on GitHub it makes little difference if you respond to a query in one hour as opposed to one day after that individual has had a chance to calm down). This provides otherwise emotionally volatile individuals the opportunity to present the best version of themselves by taking strategic breaks from heated discussions (an option that would be infeasible in a typical corporate scenario). Additionally, if an individual is not feeling up to working due to any kind of personal distress, they are not obligated to do so. This similarily allows an individual to opt out of interacting with others during a time of frustration where they might otherwise lash out inappropriately. For these reasons the impact of personality conflicts that would otherwise be nearly unavoidable in corporate circumstance can be partially mitigated in the flexible context of the GitHub workspace.

Another possibility for this null result is that a certain amount of detail could have been lost in translation (from data through to interpretation). In this thesis, three fundamental strategies for attaining the personality of various individuals have been presented:

1. Traditional SYMLOG approach: the in-person meetings and peer-review questionnaires

2. Pilot Study Annotation approach: the third-party commentary annotation of exhibited tendencies

3. Personality-Insights Automation approach: the automatic conversation of textual commentary into personality

While the most accurate approach was the questionnaires originally developed with SYMLOG in mind, getting everyone from a GitHub repository to do these questionnaires is infeasible. As such, the Pilot Study Annoation and Personality-Insights Automation approaches were developed in order to provide a feasible approximation. However, these each come with their respecitve shortcomings that may have contributed to attaining a null result.

The Pilot Study Annotation approach has a number of weaknesses. The first weakness is in the step whereby each conversation is annotated by a third-party judge in order to gauge which personality traits were being exhibited. While the survey used for this annotation is essentially identitcal to the original SYMLOG survey, the individual actually giving an opinion is one that is external to the group itself. This is in contrast to the traditional approach in which only individuals who are themselves involved in the group are permitted to give feedback on those in the group (only peers can rate peers). It is possible that this difference could add a bias to the personalities perceived. The second weakness is that the survey itself was binarized (as opposed to the inital ordinal rating scale of 0, 1, or 2 in an attempt to simplify the questionnaire for potential scalability to less-qualified annotators). This binarization could have forced the rankings from the judges into extremes that may not otherwise have been there had the scale remained intact. Indeed, when the judges were compared, there was a very low level of inter-rater agreement (0.027, -0.02, and 0.24 for the P/N, F/B, and U/D dimensions respectively).

The Personality-Insights automation technique also has its disadvantages. One disadvantage is that this technique only has an average correlation of 0.31 with the Big 5 dimensions when compared against ground truth (defined here as the traditional Big 5 personality questionnaires) [22]. Additionally, these retrieved personalities then need to undergo a transformation into the SYMLOG space before they can be utilized. This transformative mapping has yet to be formally validated.

In order to address these shortcomings in future works, there are a number of precautions that could be implemented. One precaution would be a thorough curating process to eliminate the projects that failed due to causes other than personality. This has the potential to eliminate the mixed effects of multiple causes so that a definitive answer as to whether or not personality has a causal impact on maintenance could be achieved. Another alternative, would be to take a model (such a Coelho's) that focuses on causes other than personality, and use the projects misclassified by this model as a dataset of interest. The fact that these projects were misclassified by a model that addresses the non-personality based causes of failure alludes to the possibility that personality had an impact on these projects. This would allow for an efficient procurement of projects for research where personality has a reasonable opportunity at having an observable impact (as alternative hypotheses have been partially eliminated).

With regard to addressing the emotional reservation of project members, this could be addressed by filtering out individuals from analysis that are likely to withhold their true sentiment from observation. The individuals that are most likely to audit their comments before posting them are those whose profiles can be linked to their true identities. If an individual is using a profile that can not be linked back to their true identity, there is less incentive to filter out negative commentary from their posts. Due to this effect, if we limit our analysis to just these less self-conscious individuals, we have an opportunity to witness a broader range of emotions. To achieve this, we can focus on only those individuals that appear to be using fictitious usernames. This would allow a researcher to focus on the individuals for whom they are able to attain a true personality for (as opposed to a self-audited one).

To address the lack of personality variability, there are a couple of solutions. One is to use a different technical dataset that is more opinion based. An example, would be a forum such as Gitter. In this way it is possible to obtain more commentary that is indicative of personality as opposed to content that is strictly reporting objective results (which tends to be void of sentiment). Alternatively, textual personality inference could be dropped altogether in favour of traditional personality assessment devices (questionnaires) in order to attain the true personalities of the individuals involved. These could then be used to circumvent the issue of emotionally reserved commentary and its natural counterpart; homogeneous personalities.

6.2 Conclusion

We set out to assess the SYMLOG paradigm in inferring useful information about the GitHub projects that we looked at. To this end we used Coelho's data as a baseline for predicting the success or failure of a project (by using the maintained/unmaintained labels as a proxy). We then added in personality data in an effort to improve results. What we found is that personality data gleaned from GitHub issues and reviews did not provide predictive information that was not already captured by the more objective metrics used by Coelho in their study.

Individuals on GitHub however are aware that every comment is public. This led to a fairly polite and reserved persona online, likely suppressing a great deal of emotional volatility that would otherwise be seen in more private interactions. This suppressed emotional range made it difficult for the Personality Insights service to accurately deduce the true personality of the individual it was tasked with assessing and thus inhibited our model's ability to correlate personality with results.

In addition, we assessed the relative explanatory power of SYMLOG's interpretation of the Big 5. These assessments showed that while SYMLOG adds a modest amount of explanatory information, the majority of SYMLOG's predictive power can be captured by utilizing distribution metrics of the Big 5 attributes for the various projects. While this does not allow for the numerous suggestions for corrective action that SYMLOG does, it can serve as a useful, more simplistic, proxy for whether or not SYMLOG's interpretations would add value to a model.

6.3 Further Work

To extend this work there are a number of directions to explore. One direction, would be to seek out a more emotionally informative datasource. Looking at the distribution of the personalities deduced from a prospective source can be an informative filter for assessing a dataset's viability.

Additionally, this work established the potential for textual inference of personality. This work can be used as a foundation in informing an emotionally sensitive agent about the social health and needs of a group.

References

[1] Shaosong Ou Alexander Hars. Working for free? motivations for participating in open-source projects. *International Journal of Electronic Commerce*, 6(3):25–39, 2002.

[2] Robert Freed Bales. *Social Interaction Systems: Theory and Measurement*. Routledge, 2 Park Square, Milton Park, Abingdon, Oxon OX14 4RN 711 Third Avenue, New York, NY 10017, USA, 2017.

[3] Christopher P Barlett and Craig A Anderson. Direct and indirect relations between the big 5 personality traits and aggressive and violent behavior. *Personality and Individual Differences*, 52(8):870–875, 2012.

[4] Franco D Berdun, Marcelo G Armentano, Luis S Berdun, and Matías Cincunegui. Building symlog profiles with an online collaborative game. *International Journal of Human-Computer Studies*, 127:25–37, 2019.

[5] Hudson Borges, Andre Hora, and Marco Tulio Valente. Predicting the popularity of github repositories. In *Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering*, page 9. ACM, 2016.

[6] Jailton Coelho and Marco Tulio Valente. Why modern open source projects fail. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, pages 186–196. ACM, 2017.

[7] Jailton Coelho, Marco Tulio Valente, Luciana L Silva, and Emad Shihab. Identifying unmaintained projects in github. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, page 15. ACM, 2018.

[8] Chris Ding, Xiaofeng He, Parry Husbands, Hongyuan Zha, and Horst Simon. Pagerank, hits and a unified framework for link analysis. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 249–253. SIAM, 2003.

[9] Nicolas Ducheneaut. Socialization in an open source software community: A socio-technical analysis. *Computer Supported Cooperative Work (CSCW)*, 14(4):323–368, 2005.

[10] Lisa A Fast and David C Funder. Personality as manifest in word use: correlations with self-report, acquaintance report, and behavior. *Journal of personality and social psychology*, 94(2):334, 2008.

[11] Alastair J Gill, Scott Nowson, and Jon Oberlander. What are they blogging about? personality, topic and motivation in blogs. In *Third International AAAI Conference on Weblogs and Social Media*, 2009.

[12] GitHub. Open source survey, 2017.

[13] GitHub. The state of the octoverse, 2018.

[14] Jennifer Golbeck, Cristina Robles, Michon Edmondson, and Karen Turner. Predicting personality from twitter. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*, pages 149–156. IEEE, 2011.

[15] SYMLOG Consulting Group. Overview of big five personality inventory, 2019.

[16] David Heise and N MacKinnon. *Self, identity, and social institutions.* Springer, 2010.

[17] David R. Heise. Groupsimulator, 2012.

[18] Jacob B Hirsh and Jordan B Peterson. Personality and language use in self-narratives. *Journal of research in personality*, 43(3):524–527, 2009.

[19] Jesse Hoey and Tobias Schröder. Bayesian affect control theory of self. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[20] Jesse Hoey, Tobias Schröder, Jonathan Morgan, Kimberly B Rogers, Deepak Rishi, and Meiyappan Nagappan. Artificial intelligence and social simulation: Studying group dynamics on a massive scale. *Small Group Research*, 49(6):647–683, 2018.

[21] Yan Hu, Shanshan Wang, Yizhi Ren, and Kim-Kwang Raymond Choo. User influence analysis for github developer social networks. *Expert Systems with Applications*, 108:108–118, 2018.

[22] IBM. The science behind the service, 2018.

[23] Oskar Jarczyk, Szymon Jaroszewicz, Adam Wierzbicki, Kamil Pawlak, and Michal Jankowski-Lorek. Surgical teams on github: Modeling performance of github project development processes. *Information and Software Technology*, 100:32–46, 2018.

[24] Francisco Jurado and Pilar Rodriguez. Sentiment analysis in monitoring software development processes: An exploratory case study on github's project issues. *Journal of Systems and Software*, 104:82–89, 2015.

[25] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[26] Zhifang Liao, Haozhi Jin, Yifan Li, Benhong Zhao, Jinsong Wu, and Shengzong Liu. Devrank: Mining influential developers in github. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[27] Bing Liu et al. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2(2010):627–666, 2010.

[28] Franc Mairesse, Marilyn Walker, et al. Words mark the nerds: Computational models of personality recognition through language. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 28, 2006.

[29] Robert R McCrae and Oliver P John. An introduction to the five-factor model and its applications. *Journal of personality*, 60(2):175–215, 1992.

[30] Charles Egerton Osgood, William H May, Murray Samuel Miron, and Murray S Miron. *Cross-cultural universals of affective meaning*, volume 1. University of Illinois Press, 1975.

[31] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[32] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

[33] Deepak Rishi. Affective sentiment and emotional analysis of pull request comments on github. Master's thesis, University of Waterloo, 2017.

[34] Tobias Schröder, Jesse Hoey, and Kimberly B Rogers. Modeling dynamic identities and uncertainty in social interactions: Bayesian affect control theory. *American Sociological Review*, 81(4):828–855, 2016.

[35] H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791, 2013.

[36] Gwendolyn Seidman. The big 5 and relationship maintenance on facebook. *Journal of Social and Personal Relationships*, 36(6):1785–1806, 2019.

[37] Black Duck Software. 2015 future of open source survey results, 2015.

[38] Chris Sumner, Alison Byers, Rachel Boochever, and Gregory J Park. Predicting dark triad personality traits from twitter usage and a linguistic analysis of tweets. In *2012 11th International Conference on Machine Learning and Applications*, volume 2, pages 386–393. IEEE, 2012.

[39] Jason Tsay, Laura Dabbish, and James Herbsleb. Influence of social and technical factors for evaluating contribution in github. In *Proceedings of the 36th international conference on Software engineering*, pages 356–366. ACM, 2014.

[40] Helen J Wall, Claire C Campbell, Linda K Kaye, Andy Levy, and Navjot Bhullar. Personality profiles and persuasion: An exploratory study investigating the role of the big-5, type d personality and the dark triad on susceptibility to persuasion. *Personality and Individual Differences*, 139:69–76, 2019.

[41] Tal Yarkoni. Personality in 100,000 words: A large-scale analysis of personality and word use among bloggers. *Journal of research in personality*, 44(3):363–373, 2010.

APPENDICES

Table 1: Supporting data for Table 5.1 for model Linear

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.571429	0.800000	0.444444	0.319797
1	0.709677	0.846154	0.611111	0.286353
2	0.444444	0.666667	0.333333	0.321263
3	0.370370	0.555556	0.277778	0.323560
4	0.645161	0.769231	0.555556	0.303296
5	0.384615	0.555556	0.294118	0.284100
6	0.100000	0.333333	0.058824	0.259781
7	0.500000	0.857143	0.352941	0.312886
8	0.320000	0.500000	0.235294	0.308550
9	0.454545	1.000000	0.294118	0.275917
10	0.400000	0.625000	0.294118	0.325989

Table 2: Supporting data for Table 5.1 for model Logistic

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.642857	0.900000	0.500000	0.320711
1	0.733333	0.916667	0.611111	0.286644
2	0.384615	0.625000	0.277778	0.276213
3	0.307692	0.500000	0.222222	0.291116
4	0.687500	0.785714	0.611111	0.255233
5	0.384615	0.555556	0.294118	0.305473
6	0.272727	0.600000	0.176471	0.281028
7	0.500000	0.857143	0.352941	0.263143
8	0.320000	0.500000	0.235294	0.302370
9	0.300000	1.000000	0.176471	0.297481
10	0.416667	0.714286	0.294118	0.262248

Table 3: Supporting data for Table 5.1 for model Xgboost

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.484848	0.533333	0.444444	0.211711
1	0.666667	0.733333	0.611111	0.068576
2	0.482759	0.636364	0.388889	0.229856
3	0.533333	0.666667	0.444444	0.189851
4	0.486486	0.473684	0.500000	0.040176
5	0.482759	0.583333	0.411765	0.131666
6	0.588235	0.588235	0.588235	0.117841
7	0.461538	0.666667	0.352941	0.197367
8	0.424242	0.437500	0.411765	0.532646
9	0.480000	0.750000	0.352941	0.095209
10	0.666667	0.590909	0.764706	0.287793

Table 4: Supporting data for Table 5.1 for model SVM

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.518519	0.777778	0.388889	0.174492
1	0.620690	0.818182	0.500000	0.172164
2	0.370370	0.555556	0.277778	0.164763
3	0.466667	0.583333	0.388889	0.173952
4	0.571429	0.800000	0.444444	0.179825
5	0.384615	0.555556	0.294118	0.165265
6	0.285714	0.750000	0.176471	0.168547
7	0.416667	0.714286	0.294118	0.164465
8	0.250000	0.428571	0.176471	0.166008
9	0.300000	1.000000	0.176471	0.161527
10	0.320000	0.500000	0.235294	0.166141

Table 5: Supporting data for Table 5.1 for model Neural

Run	F1	Precision	Recall	Best Threshold from PRC
0	0	0	0.0	0.218530
1	0	0	0.0	0.254368
2	0	0	0.0	0.254541
3	0	0	0.0	0.245875
4	0	0	0.0	0.261333
5	0	0	0.0	0.258383
6	0	0	0.0	0.227158
7	0	0	0.0	0.234747
8	0	0	0.0	0.200755
9	0	0	0.0	0.277222
10	0	0	0.0	0.244435

Table 6: Supporting data for Table 5.2 for model Xgboost: SYMLOG

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.484848	0.533333	0.444444	0.211711
1	0.666667	0.733333	0.611111	0.068576
2	0.482759	0.636364	0.388889	0.229856
3	0.533333	0.666667	0.444444	0.189851
4	0.486486	0.473684	0.500000	0.040176
5	0.482759	0.583333	0.411765	0.131666
6	0.588235	0.588235	0.588235	0.117841
7	0.461538	0.666667	0.352941	0.197367
8	0.424242	0.437500	0.411765	0.532646
9	0.480000	0.750000	0.352941	0.095209
10	0.666667	0.590909	0.764706	0.287793

Table 7: Supporting data for Table 5.2 for model Xgboost: SYMLOG excluding Bales Measures

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.344828	0.454545	0.277778	0.317781
1	0.628571	0.647059	0.611111	0.284519
2	0.516129	0.615385	0.444444	0.398022
3	0.388889	0.388889	0.388889	0.149201
4	0.588235	0.625000	0.555556	0.099758
5	0.482759	0.583333	0.411765	0.065941
6	0.307692	0.444444	0.235294	0.154898
7	0.689655	0.833333	0.588235	0.066895
8	0.451613	0.500000	0.411765	0.194836
9	0.428571	0.545455	0.352941	0.147320
10	0.413793	0.500000	0.352941	0.099302

Table 8: Supporting data for Table 5.3 for model Linear: Big5

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.666667	1.000000	0.500000	0.303264
1	0.727273	0.800000	0.666667	0.298356
2	0.250000	0.500000	0.166667	0.318675
3	0.444444	0.666667	0.333333	0.283982
4	0.580645	0.692308	0.500000	0.295907
5	0.444444	0.600000	0.352941	0.293686
6	0.190476	0.500000	0.117647	0.277192
7	0.480000	0.750000	0.352941	0.288656
8	0.370370	0.500000	0.294118	0.326156
9	0.380952	1.000000	0.235294	0.290469
10	0.444444	0.600000	0.352941	0.311360

Table 9: Supporting data for Table 5.3 for model Logistic: Big5

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.620690	0.818182	0.500000	0.247232
1	0.774194	0.923077	0.666667	0.248741
2	0.444444	0.666667	0.333333	0.238615
3	0.413793	0.545455	0.333333	0.253196
4	0.625000	0.714286	0.555556	0.243345
5	0.461538	0.666667	0.352941	0.223025
6	0.200000	0.666667	0.117647	0.307724
7	0.538462	0.777778	0.411765	0.257238
8	0.357143	0.454545	0.294118	0.311112
9	0.434783	0.833333	0.294118	0.287059
10	0.518519	0.700000	0.411765	0.313951

Table 10: Supporting data for Table 5.3 for model Xgboost: Big5

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.606061	0.666667	0.555556	0.219437
1	0.666667	0.666667	0.666667	0.105023
2	0.484848	0.533333	0.444444	0.095354
3	0.437500	0.500000	0.388889	0.100464
4	0.606061	0.666667	0.555556	0.147248
5	0.571429	0.727273	0.470588	0.130812
6	0.333333	0.571429	0.235294	0.065015
7	0.387097	0.428571	0.352941	0.136632
8	0.258065	0.285714	0.235294	0.070250
9	0.482759	0.583333	0.411765	0.048191
10	0.400000	0.461538	0.352941	0.332143

Table 11: Supporting data for Table 5.3 for model SVM: Big5

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.592593	0.888889	0.444444	0.231209
1	0.727273	0.800000	0.666667	0.235397
2	0.444444	0.666667	0.333333	0.226062
3	0.370370	0.555556	0.277778	0.209634
4	0.645161	0.769231	0.555556	0.253186
5	0.480000	0.750000	0.352941	0.220024
6	0.272727	0.600000	0.176471	0.164531
7	0.480000	0.750000	0.352941	0.216999
8	0.344828	0.416667	0.294118	0.257436
9	0.434783	0.833333	0.294118	0.208651
10	0.571429	0.727273	0.470588	0.191101

Table 12: Supporting data for Table 5.3 for model Neural: Big5

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.00	0.000000	0.000000	0.216254
1	0.00	0.000000	0.000000	0.294412
2	0.24	0.428571	0.166667	0.281605
3	0.00	0.000000	0.000000	0.292968
4	0.00	0.000000	0.000000	0.299909
5	0.00	0.000000	0.000000	0.257164
6	0.00	0.000000	0.000000	0.276123
7	0.00	0.000000	0.000000	0.275366
8	0.00	0.000000	0.000000	0.278266
9	0.00	0.000000	0.000000	0.292608
10	0.00	0.000000	0.000000	0.291853

Table 13: Supporting data for Table 5.4 for model Xgboost: SYMLOG + Big5

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.533333	0.666667	0.444444	0.062739
1	0.526316	0.500000	0.555556	0.056072
2	0.600000	0.750000	0.500000	0.147235
3	0.551724	0.727273	0.444444	0.094296
4	0.571429	0.588235	0.555556	0.060482
5	0.500000	0.636364	0.411765	0.084442
6	0.466667	0.538462	0.411765	0.036776
7	0.562500	0.600000	0.529412	0.083118
8	0.470588	0.470588	0.470588	0.200512
9	0.416667	0.714286	0.294118	0.145872
10	0.470588	0.470588	0.470588	0.180275

Table 14: Supporting data for Table 5.4 for model Xgboost: Big5 group medians

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.206897	0.272727	0.166667	0.026780
1	0.275862	0.363636	0.222222	0.017839
2	0.222222	0.333333	0.166667	0.042479
3	0.486486	0.473684	0.500000	0.023541
4	0.242424	0.266667	0.222222	0.036367
5	0.148148	0.200000	0.117647	0.030719
6	0.193548	0.214286	0.176471	0.032649
7	0.142857	0.181818	0.117647	0.008101
8	0.370370	0.500000	0.294118	0.034203
9	0.240000	0.375000	0.176471	0.019562
10	0.516129	0.571429	0.470588	0.031684

Table 15: Supporting data for Table 5.4 for model Xgboost: Big5 percentiles

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.606061	0.666667	0.555556	0.219437
1	0.666667	0.666667	0.666667	0.105023
2	0.484848	0.533333	0.444444	0.095354
3	0.437500	0.500000	0.388889	0.100464
4	0.606061	0.666667	0.555556	0.147248
5	0.571429	0.727273	0.470588	0.130812
6	0.333333	0.571429	0.235294	0.065015
7	0.387097	0.428571	0.352941	0.136632
8	0.258065	0.285714	0.235294	0.070250
9	0.482759	0.583333	0.411765	0.048191
10	0.400000	0.461538	0.352941	0.332143

Table 16: Supporting data for Table 5.4 for model Xgboost: SYMLOG + Big5

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.533333	0.666667	0.444444	0.062739
1	0.526316	0.500000	0.555556	0.056072
2	0.600000	0.750000	0.500000	0.147235
3	0.551724	0.727273	0.444444	0.094296
4	0.571429	0.588235	0.555556	0.060482
5	0.500000	0.636364	0.411765	0.084442
6	0.466667	0.538462	0.411765	0.036776
7	0.562500	0.600000	0.529412	0.083118
8	0.470588	0.470588	0.470588	0.200512
9	0.416667	0.714286	0.294118	0.145872
10	0.470588	0.470588	0.470588	0.180275

Table 17: Supporting data for Table 5.4 for model Xgboost: Big5 group medians

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.206897	0.272727	0.166667	0.026780
1	0.275862	0.363636	0.222222	0.017839
2	0.222222	0.333333	0.166667	0.042479
3	0.486486	0.473684	0.500000	0.023541
4	0.242424	0.266667	0.222222	0.036367
5	0.148148	0.200000	0.117647	0.030719
6	0.193548	0.214286	0.176471	0.032649
7	0.142857	0.181818	0.117647	0.008101
8	0.370370	0.500000	0.294118	0.034203
9	0.240000	0.375000	0.176471	0.019562
10	0.516129	0.571429	0.470588	0.031684

Table 18: Supporting data for Table 5.4 for model Xgboost: Big5 percentiles

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.606061	0.666667	0.555556	0.219437
1	0.666667	0.666667	0.666667	0.105023
2	0.484848	0.533333	0.444444	0.095354
3	0.437500	0.500000	0.388889	0.100464
4	0.606061	0.666667	0.555556	0.147248
5	0.571429	0.727273	0.470588	0.130812
6	0.333333	0.571429	0.235294	0.065015
7	0.387097	0.428571	0.352941	0.136632
8	0.258065	0.285714	0.235294	0.070250
9	0.482759	0.583333	0.411765	0.048191
10	0.400000	0.461538	0.352941	0.332143

Table 19: Supporting data for Table 5.5 for model Linear

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.800000	0.823529	0.777778	0.460781
1	0.833333	0.833333	0.833333	0.498431
2	0.764706	0.812500	0.722222	0.426589
3	0.666667	0.666667	0.666667	0.407420
4	0.742857	0.764706	0.722222	0.435147
5	0.705882	0.705882	0.705882	0.466158
6	0.750000	0.800000	0.705882	0.468412
7	0.848485	0.875000	0.823529	0.372257
8	0.727273	0.750000	0.705882	0.491391
9	0.625000	0.666667	0.588235	0.411705
10	0.777778	0.736842	0.823529	0.454764

Table 20: Supporting data for Table 5.5 for model Logistic

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.800000	0.823529	0.777778	0.385421
1	0.857143	0.882353	0.833333	0.344477
2	0.764706	0.812500	0.722222	0.338186
3	0.666667	0.733333	0.611111	0.346995
4	0.750000	0.857143	0.666667	0.398775
5	0.774194	0.857143	0.705882	0.311099
6	0.774194	0.857143	0.705882	0.355012
7	0.838710	0.928571	0.764706	0.337690
8	0.727273	0.750000	0.705882	0.325058
9	0.600000	0.692308	0.529412	0.347421
10	0.722222	0.684211	0.764706	0.321410

Table 21: Supporting data for Table 5.5 for model Xgboost

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.787879	0.866667	0.722222	0.620246
1	0.941176	1.000000	0.888889	0.433287
2	0.722222	0.722222	0.722222	0.181512
3	0.717949	0.666667	0.777778	0.468489
4	0.848485	0.933333	0.777778	0.304283
5	0.800000	0.923077	0.705882	0.338074
6	0.812500	0.866667	0.764706	0.273220
7	0.857143	0.833333	0.882353	0.486441
8	0.764706	0.764706	0.764706	0.481713
9	0.687500	0.733333	0.647059	0.180365
10	0.800000	0.777778	0.823529	0.332302

Table 22: Supporting data for Table 5.5 for model SVM

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.800000	0.823529	0.777778	0.290228
1	0.888889	0.888889	0.888889	0.162095
2	0.764706	0.812500	0.722222	0.331835
3	0.562500	0.642857	0.500000	0.213825
4	0.764706	0.812500	0.722222	0.269982
5	0.733333	0.846154	0.647059	0.320915
6	0.733333	0.846154	0.647059	0.297001
7	0.866667	1.000000	0.764706	0.320607
8	0.727273	0.750000	0.705882	0.157188
9	0.645161	0.714286	0.588235	0.311217
10	0.764706	0.764706	0.764706	0.293813

Table 23: Supporting data for Table 5.5 for model Neural

Run	F1	Precision	Recall	Best Threshold from PRC
0	0	0	0.0	0.350153
1	0	0	0.0	0.332685
2	0	0	0.0	0.364847
3	0	0	0.0	0.387751
4	0	0	0.0	0.346743
5	0	0	0.0	0.361893
6	0	0	0.0	0.360139
7	0	0	0.0	0.358704
8	0	0	0.0	0.386791
9	0	0	0.0	0.317719
10	0	0	0.0	0.373900

Table 24: Supporting data for Table 5.6 for model SVM: Coelho

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.800000	0.823529	0.777778	0.290228
1	0.888889	0.888889	0.888889	0.162095
2	0.764706	0.812500	0.722222	0.331835
3	0.562500	0.642857	0.500000	0.213825
4	0.764706	0.812500	0.722222	0.269982
5	0.733333	0.846154	0.647059	0.320915
6	0.733333	0.846154	0.647059	0.297001
7	0.866667	1.000000	0.764706	0.320607
8	0.727273	0.750000	0.705882	0.157188
9	0.645161	0.714286	0.588235	0.311217
10	0.764706	0.764706	0.764706	0.293813

Table 25: Supporting data for Table 5.6 for model Xgboost: Coelho

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.787879	0.866667	0.722222	0.620246
1	0.941176	1.000000	0.888889	0.433287
2	0.722222	0.722222	0.722222	0.181512
3	0.717949	0.666667	0.777778	0.468489
4	0.848485	0.933333	0.777778	0.304283
5	0.800000	0.923077	0.705882	0.338074
6	0.812500	0.866667	0.764706	0.273220
7	0.857143	0.833333	0.882353	0.486441
8	0.764706	0.764706	0.764706	0.481713
9	0.687500	0.733333	0.647059	0.180365
10	0.800000	0.777778	0.823529	0.332302

Table 26: Supporting data for Table 5.6 for model SVM: Coelho + SYMLOG

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.800000	0.823529	0.777778	0.380553
1	0.914286	0.941176	0.888889	0.384487
2	0.777778	0.777778	0.777778	0.355043
3	0.628571	0.647059	0.611111	0.434960
4	0.823529	0.875000	0.777778	0.338887
5	0.774194	0.857143	0.705882	0.410532
6	0.750000	0.800000	0.705882	0.281526
7	0.774194	0.857143	0.705882	0.336161
8	0.764706	0.764706	0.764706	0.369584
9	0.666667	0.769231	0.588235	0.329820
10	0.736842	0.666667	0.823529	0.370749

Table 27: Supporting data for Table 5.6 for model Xgboost: Coelho + SYMLOG

Run	F1	Precision	Recall	Best Threshold from PRC
0	0.705882	0.750000	0.666667	0.486957
1	0.914286	0.941176	0.888889	0.293830
2	0.777778	0.777778	0.777778	0.495266
3	0.684211	0.650000	0.722222	0.274267
4	0.777778	0.777778	0.777778	0.372584
5	0.785714	1.000000	0.647059	0.301120
6	0.714286	0.909091	0.588235	0.332267
7	0.833333	0.789474	0.882353	0.228535
8	0.787879	0.812500	0.764706	0.323699
9	0.687500	0.733333	0.647059	0.132830
10	0.888889	0.842105	0.941176	0.418245

Table 28: U tests of significance (p values) for Table 5.1

Model	Linear	Logistic	Xgboost	SVM	Neural
Linear	0.500	0.408	0.0	0.348	0.0
Logistic	0.408	0.500	0.0	0.438	0.0
Xgboost	0.000	0.000	0.5	0.000	0.0
SVM	0.348	0.438	0.0	0.500	0.0
Neural	0.000	0.000	0.0	0.000	0.5

Table 29: U tests of significance (p values) for Table 5.2

Model	Xgboost: SYMLOG	Xgboost: SYM...
Xgboost: SYMLOG	0.500	0.244
Xgboost: SYMLOG excluding Bales Measures	0.244	0.500

Table 30: U tests of significance (p values) for Table 5.3

Model	Linear: Big5	Logistic: Big5	Xgboost: Big5	SVM: Big5	Neural: Big5
Linear: Big5	0.500	0.276	0.000	0.301	0.0
Logistic: Big5	0.276	0.500	0.002	0.471	0.0
Xgboost: Big5	0.000	0.002	0.500	0.002	0.0
SVM: Big5	0.301	0.471	0.002	0.500	0.0
Neural: Big5	0.000	0.000	0.000	0.000	0.5

Table 31: U tests of significance (p values) for Table 5.4

Model	Xgboost: SYM...	Xgboost: Big...	Xgboost: Big...
Xgboost: SYMLOG + Big5	0.500	0.098	0.40
Xgboost: Big5 group medians	0.098	0.500	0.15
Xgboost: Big5 percentiles	0.400	0.150	0.50

Table 32: U tests of significance (p values) for Table 5.4

Model	Xgboost: SYM...	Xgboost: Big...	Xgboost: Big...
Xgboost: SYMLOG + Big5	0.500	0.098	0.40
Xgboost: Big5 group medians	0.098	0.500	0.15
Xgboost: Big5 percentiles	0.400	0.150	0.50

Table 33: U tests of significance (p values) for Table 5.5

Model	Linear	Logistic	Xgboost	SVM	Neural
Linear	0.500	0.187	0.430	0.157	0.0
Logistic	0.187	0.500	0.238	0.452	0.0
Xgboost	0.430	0.238	0.500	0.202	0.0
SVM	0.157	0.452	0.202	0.500	0.0
Neural	0.000	0.000	0.000	0.000	0.5

Table 34: U tests of significance (p values) for Table 5.6

Model	SVM: Coelho	Xgboost: Coelho	SVM: Coelho ...	Xgboost: Coe...
SVM: Coelho	0.500	0.202	0.237	0.219
Xgboost: Coelho	0.202	0.500	0.453	0.477
SVM: Coelho + SYMLOG	0.237	0.453	0.500	0.477
Xgboost: Coelho + SYMLOG	0.219	0.477	0.477	0.500