Predicting Cognitive Reflection from Digital Fingerprints

by

An Jimenez

Submitted to the Departments of Brain and Cognitive Sciences and Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computation and Cognition

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author .....
Departments of Brain and Cognitive Sciences and
Electrical Engineering and Computer Science
May 2nd, 2022

Certified by .....
David Rand
Erwin H. Schell Professor and Professor of Management Science and Brain and Cognitive Sciences at MIT, and the Director of the Human Cooperation Laboratory and the Applied Cooperation Team
Thesis Supervisor

Accepted by .....
Professor Mehrdad Jazayeri
Chair, Master of Engineering Thesis Committee

Predicting Cognitive Reflection from Digital Fingerprints

by

An Jimenez

Submitted to the Departments of Brain and Cognitive Sciences and
Electrical Engineering and Computer Science
on May 2nd, 2022, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computation and Cognition

Abstract

While social media is beneficial in facilitating social connections and spreading knowledge on a large scale, its negative impacts — the propagation of misinformation through networks and the emergence of echo chambers in particular — are consequential and dangerous, inducing a more divergent rather than cohesive society. What cognitive mechanisms are at play when users decide what to share and who to follow on social media? A recent study provides evidence that users with higher Cognitive Reflection Test (CRT) scores — a popular measure for reflective thinking — are more discerning in their Twitter behavior (Mosleh et al., 2021). While previous research sheds light on this relationship between cognitive reflection and Twitter behavior, there is an opportunity to generalize these correlations to larger populations and across different social media platforms by building a computational model to predict cognitive reflection from social media activity, which is the focus of my project. Applying machine learning techniques to the dataset used in Mosleh's study, I created a model that predicts CRT scores from Twitter features such as Tweet content and accounts followed (followees) and also determined which features and combinations of features are most predictive of cognitive reflection. Correlations between predicted and actual CRT scores are strongest when predicting with information related to followees ($r = 0.25$) and followee bios ($r = 0.24$). Combining followee features and applying different regression models improves prediction accuracy ($r = 0.29$). These conclusions help form a more complete picture of how cognitive reflection relates to social media activity, which has important implications for how we can encourage more intentional social media use and ultimately, reconnect divisive populations online.

Thesis Supervisor: David Rand
Title: *Erwin H. Schell Professor* and Professor of Management Science and Brain and Cognitive Sciences at MIT, and the Director of the Human Cooperation Laboratory and the Applied Cooperation Team

Acknowledgments

This project is the byproduct of both 18 months of personal work but also years of support and guidance from family, friends, and mentors alike. Amongst the sea of people who have helped me along the way, I extend my gratitude where it is absolutely due.

A recent shift towards Computation and Cognition and one cold email later, I was thrilled to be welcomed into the Human Cooperation Lab for my Master's of Engineering in the Spring of 2021 to perform cutting-edge work at the intersection of behavioral economics, psychology, and computation. To Professors David Rand and Mohsen Mosleh, my mentors for this project – for placing an immense amount of trust and freedom in me to experiment, explore, and take the development and direction of this project into my own hands.

I completed this thesis in a number of locations around Cambridge – from Hayden library overlooking the Boston skyline to Pepita coffee while savoring a hot drink – but I wrote most of what you are reading nestled in a cozy Central Square apartment. To Angela Cai and Anita Mokkapati – for welcoming me into their home with open arms as a haven for emotional support, productivity, and delicious home-cooked food.

Over the five-year marathon that was my MIT experience, I relied heavily on creative hobbies to keep me expressive, present, and grounded. To Anna Kohler and Tony Eng – for providing opportunities for me to step outside the technical bubble and express myself creatively – from acting to teaching to public speaking – which I believe is equally as important as my technical strengths.

Yet, the greatest support came from those cheering on the sidelines since the beginning. To my family – Jose Jimenez, Chinglee Chew, and Liang Jimenez – for being a sounding board of ideas and for providing perspective that life goes beyond a thesis. To Sarah Spector – for sharing MIT with me since Day 1 and for reminding me that "if we knew what we were doing, it wouldn't be research." Finally, to Jewon Sohn, Yun Gu, Ava Waitz, Anne Li, Soft Chaisuwan, and Geena Wang – for your unwavering support in me, no matter the distance nor time.

Contents

1 Introduction **13**

2 Background **15**

2.1 The Cognitive Reflection Test (CRT) 15

2.2 CRT Score Correlates with Twitter Behavior 16

2.3 A Computational Approach to Predict Traits 16

2.4 Prior Work on Twitter-Based Predictions 17

3 Methodology **19**

3.1 Project Goals 19

3.2 Machine Learning Task 20

3.3 Components of a Model 21

3.3.1 Train/Test Split 21

3.3.2 Vectorization for Text Features 21

3.3.3 Dimensionality Reduction for Text Features 24

3.3.4 Feature Selection 25

3.3.5 Prediction 27

3.4 Choices in Code Design 28

4 Dataset **31**

4.1 Target Features 31

4.2 Exclusions and Pre-Processing 33

4.3 Preliminary Statistics 34

4.4 Data Dimensionality 36
4.5 Umbrella Features 37

5 Pipeline **39**
5.1 Overview 39
5.2 Model 40
5.2.1 Transformation 40
5.2.2 Feature Selection 42
5.2.3 Prediction 42
5.3 Informative Features 44

6 Results **47**
6.1 Individual Features 47
6.2 Umbrella Features 49
6.3 Combined Features 50
6.4 Prediction with the Full Dataset 51
6.5 Informative Features 52

7 Conclusion **59**
7.1 Discussion 59
7.2 Ethics 60
7.3 Future Work 61

A Pipeline Schematic **63**

B Table of Target Features **65**

C Libraries, Packages, and Modules **69**

List of Figures

3-1 Screenshot of function from code providing function description, inputs, outputs, and data types. 30
3-2 Screenshot of top of the Jupyter notebook for running the pipeline. . 30

4-1 Distribution of participant age for Full and Complete dataset 35
4-2 Distribution of CRT scores for Full and Complete dataset. A CRT score of 1.0 means the user answered 7 out of 7 questions correctly. . 35
4-3 Umbrella feature categories. Text features are underlined. 38

5-1 High-level schematic of pipeline. 40
5-2 Transformation phase of pipeline 41
5-3 Selection phase of the pipeline 42
5-4 Individual and umbrella feature prediction 43
5-5 Combined feature prediction 44
5-6 Informative features schematic 45

6-1 Individual Feature prediction on Complete dataset 48
6-2 Umbrella feature prediction 49
6-3 Combined individual feature prediction on Complete dataset 51
6-4 Combined umbrella feature prediction on Complete dataset 51
6-5 Individual feature prediction on Full dataset 53
6-6 Informative Tweet/Retweet text with n_gram ranges (1, 1) and (2, 2) 57
6-7 Informative followee bios with n_gram ranges (1, 1) and (2, 2) 57
6-8 Informative follower bios with n_gram ranges (1, 1) and (2, 2) 57

6-9 Informative mentions 58
6-10 Informative domains 58
6-11 Informative followees 58
6-12 Informative bio features 58
6-13 Informative hashtags 58

A-1 High-level pipeline schematic, targeted for scientific and non-scientific audiences 63

List of Tables

2.1 Studies using Twitter digital fingerprints to predict Big Five personality traits 18

4.1 Preliminary statistics on profile features in Full and Complete dataset 35

4.2 Number of dimensions produced from TF-IDF vectorization on the Complete dataset using `min_df` = 1 versus `min_df` = 10, keeping `max_df` at 0.99 for both. The last column shows the percentage of the total dimensionality preserved when `min_df` = 10. 36

4.3 Explained Variance after TruncatedSVD with `n_components` = 50 on the Complete dataset 37

5.1 `TfidfVectorization` parameters for text feature transformation . . . 41

5.2 `TfidfVectorization` parameters for informative features 45

B.1 Target features with descriptions and examples 68

C.1 Modules used in code infrastructure 71

Chapter 1

Introduction

Social media, while a positive force in connecting individuals and increasing accessibility to information, is also a breeding ground for two of the greatest perils of our technology-reliant society: the spread of misinformation (Pennycook et al., 2020) and the existence of digital echo chambers (Du et al., 2016), in which users only encounter opinions that reinforce their own.

The duality of social media becomes extremely apparent during the height of major global events, such as the Covid-19 pandemic (Venegas-Vera, Colbert, and Lerma, 2020). Since the start of the pandemic, social media use has increased by 20-87% (Bruno Kessler Foundation, 2020). On one hand, social media has reduced health anxiety during isolation by connecting individuals to one another virtually (Stuart et al., 2021), improved health literacy and public awareness by sharing bite-sized scientific information online (Depoux et al., 2020), and allowing health care providers and experts to discuss and analyze literature in real-time (Gottlieb and Dyer, 2020). On the other hand, social media allows users to freely post content without prior vetting or peer review by health and medical professionals, which has led to a deluge of content and the spread of false health and virus-related information (Kouzy et al., 2020; Puri et al., 2020). Across any social media platform, these dangers are exacerbated by the instantaneous nature of social media, causing the rapid dissemination of information online and pushing quick-thinking behavior.

Understanding what cognitive mechanisms drive social media behavior — the con-

tent users share and the accounts users follow, in particular — is of high interest, given our heavy reliance on social media applications and the consequences from information spread. Recent work has explored the correlational relationships between social media features and cognitive reflection in a small subset of Twitter users (Mosleh et al., 2021); however, it is important to capture a more complete landscape of how cognitive reflection relates to social media activity across larger, more representative groups of users and social media applications beyond Twitter. To accomplish this, I created a machine learning model that predicts cognitive reflection of out-of-sample users with reasonable accuracy, using the widely-used Cognitive Reflection Test (CRT) as a proxy for measuring cognitive reflection. Using the model, I determine which individual and groups of Twitter features are most predictive for cognitive reflection.

My thesis lies at the intersection of computation and cognitive science. While the core of the work is computational — building a machine learning pipeline from scratch — the project applies the computational model to understand human thinking and behavior. Understanding what cognitive reflection is, how it is measured, and its implications for social media use at the population level informs the technical design and the conclusions drawn.

In this thesis, I discuss the following content related to the prediction pipeline for CRT score: relevant background information and prior work (Chapter 2), project goals and methodology (Chapter 3), the project dataset (Chapter 4), the prediction pipeline (Chapter 5), results (Chapter 6), and conclusions, including ethical considerations and suggestions for future work (Chapter 7).

Chapter 2

Background

2.1 The Cognitive Reflection Test (CRT)

Cognitive reflection is an individual's inclination to engage in analytical thinking, and it is commonly measured by behavioral economists using the Cognitive Reflection Test (Frederick, 2005). The test offers a series of questions which have intuitively compelling but incorrect answers, and individuals must override their intuitive response to arrive at the correct answer. For example, consider the following question:

"If you're running a race and you pass the person in second place, what place are you in?"

The correct answer is second place; however, it is easy to respond with the incorrect answer — first place — when responding automatically rather than analytically. The original CRT is a three-item test with numeric questions.

Using CRT score as a proxy, previous studies provide evidence that a greater propensity to engage in analytical reasoning improves our ability to assess the accuracy of information. Notably, one study shows that people who score higher on the CRT are less trusting of fake news content and less likely to share such content with their social networks, which is of particular importance to this project's wider implications for mitigating the spread of misinformation (Pennycook and Rand, 2019). Its pervasiveness and past success in measuring cognitive ability and mental heuristics

make the CRT a useful task to quantify one's ability to engage in reflective thinking.

2.2 CRT Score Correlates with Twitter Behavior

Another recent study explores the motivations behind social media use through a cognitive science lens using a seven-item CRT, comprised of the original three numeric questions and four non-numeric questions (Mosleh et al., 2021). Specifically, the study captured CRT scores and Twitter activity across 1,901 users to determine the correlational relationship between analytical thinking and social media behavior. Twitter users with higher CRT scores are more discerning in their social media use — that is, they are more selective in the accounts they follow, Tweet about weightier content, and share links from more reputable news sources. Additionally, groups of users with lower CRT scores tend to follow a set of accounts which are avoided by people with higher CRT scores regardless of political opinion, providing evidence for the existence of cognitive echo chambers.

One important limitation of this study is the size of the participant pool, which only represents a small subsection of the entire Twitter population. As such, the specific social media activity from a single user might significantly affect the strength of correlations found in the study. Generalizing these findings to larger groups and across different social media applications would contribute to a more complete understanding of how cognitive reflection drives social media activity, which is the focus of my thesis.

2.3 A Computational Approach to Predict Traits

One approach to extending the relationship between individual traits and social media activity to larger populations is by building a machine learning model that learns known relationships between traits and activities in order to predict an unseen user's traits, given what is called their digital fingerprint. Previous studies in this line of work show promising results for a computational-based approach, particularly for

predicting personality using the Big Five taxonomy for describing traits. The Big Five personality traits are openness, conscientiousness, extraversion, agreeableness, and neuroticism (Goldberg, 1990). One study conducted a series of meta-analyses to predict Big Five personality traits using digital fingerprints, which included pictures and text shared from a variety of social media platforms (Azucar et al., 2018). The predictive power of these digital fingerprints yields Pearson r correlations between predicted and actual personality traits ranging from 0.29 to 0.40 for the Agreeableness to Extraversion traits respectively, which align with the typical strength of relationship between personality and behavior (Roberts et al., 2007). Other studies have been able to predict personality using only one social media platform. One such study uses a combination of Facebook-specific profile features such as personal information, activities and preferences, and language information to predict the Big Five personality traits with a regression analysis, yielding correlations between actual and predicted personality scores between -0.23 and 0.26 (Golbeck et al., 2011).

In addition to observing a user's entire digital fingerprint, there is also work around isolating and comparing which umbrella features (i.e. groups of related individual features) are most predictive for personality. One recent study by Mori and Haruno investigates to what extent different types of Twitter information (network information, time information, word statistics, or bag of words) predicts individual traits and attributes most accurately. The authors conclude network information holds the highest predictive power ($r = 0.44$), followed by time information, word statistics, and lastly, bag of words (Mori and Haruno, 2020). Other studies use individual features alone for prediction. Using the individual feature of Facebook likes, Youyou et al. were able to predict the Big Five personality traits, generating a correlation coefficient of 0.56 for computer-based judgements (Youyou et al., 2015).

2.4 Prior Work on Twitter-Based Predictions

One of the most popular social media platforms today is Twitter, which is a content-rich, text-heavy social media application. On Twitter, individuals can share their

thoughts in a concise way — with 280 characters or less in the form of a Tweet. By nature, Twitter allows application users to get a glimpse into the thoughts, musings, and interests of individuals, which can be telling of a one's personality or other individual traits. Previous work shows that Twitter information is predictive of the Big Five personality traits (Azucar et al., 2018). The following studies use user's Twitter digital fingerprints to predict Big Five personality, generating the following r value ranges between predicted and actual personality.

Study	**Pearson's r value range**	**Number of participants**	**Twitter features**
Liu et al. 2016	[0.15, 0.19]	254	Text
Liu et al. 2016	[0.05, 0.19]	429	Profile image
Farnadi et al. 2016	[0.26, 0.46]	44	Demographics and text
Qiu et al. 2012	[0.16, 0.28]	142	Text
Sumner et al. 2012	[0.16, 0.25]	616	Text

Table 2.1: Studies using Twitter digital fingerprints to predict Big Five personality traits

Overall, these social media-based predictions are valuable for a variety of use cases, ranging from tailoring application features to enhance the user experience to improving user recommendations. My project adds to this line of work by building a social media-based predictive model for cognitive reflection using the dataset from Mosleh et al.'s study.

Chapter 3

Methodology

3.1 Project Goals

In this project, I will extend current research around cognitive reflection and social media behavior by building a machine learning model to predict CRT score given patterns of users' Twitter behavior. The goal of this project is five-fold:

1. To create a model that predicts CRT score with reasonable accuracy from Twitter digital fingerprints
2. To determine which individual Twitter features are the strongest predictors for CRT score
3. To determine which umbrella Twitter features are the strongest predictors for CRT score
4. To determine which combinations of Twitter features have the strongest predictive power for CRT score
5. To evaluate which words and phrases from Twitter profiles are most informative of CRT score

Accomplishing the goals above will reveal which aspects of social media profiles are the strongest indicators for cognitive reflection, both at the macro level (i.e. groups of Twitter features) and the micro level (i.e. words and phrases).

3.2 Machine Learning Task

The task of this project is a supervised machine learning task — that is, an algorithm is given both inputs and outputs and learns the relationship between the two in order to predict the output of an out-of-sample input. On the contrary, unsupervised learning is a task where only input data is provided, and the model finds patterns in the data on its own for prediction. Since the dataset of interest contains both inputs (Twitter features) and outputs (CRT score), I treat this task as a supervised machine learning task.

Drawbacks of Classification Problems

Under the umbrella of supervised machine learning tasks, this prediction task could be modeled as either a classification problem or a regression problem. Classification problems predict a discrete label from a set of predictors while regression problems predict a continuous value from a set of predictors.

If we take the task to be a classification problem, the model would need to predict CRT score by categorizing a predicted CRT score into one of 8 classifications (0 questions correct, 1 question correct, ..., 7 questions correct). Because there are eight possible labels, incorrect classifications are not very informative of how close our predictions are to the correct classification. Instead, we gain binary information of whether or not our model made a successful prediction, which is not helpful for improving the model since it learns and modifies its algorithm from the previous prediction losses.

Predicting CRT Score as a Regression Problem

Instead of asking, "Is the CRT score correctly classified?," we learn more about the accuracy of the prediction by asking "How close is our prediction to the actual score?" which is the basis for regression problems. In regression problems, we predict a continuous quantity and determine the error between the predicted and actual values, thus revealing not just whether we correctly predict CRT score but also by how

much our prediction was off by. Additionally, because our target follows a sequential ordering, this task lends itself nicely as a regression problem. By building a model to find the line of best fit between the Twitter features and CRT score (i.e. predicting continuous values), we can predict the output more accurately than if we treat the task as a discrete classification problem.

3.3 Components of a Model

At a high-level, the following pieces are standard components of a machine learning model and are integrated into my pipeline: (1) train/test split, (2) feature vectorization, (3) dimensionality reduction, (4) feature selection, and (5) prediction. In the following sections, I describe how each component works.

3.3.1 Train/Test Split

For any supervised machine learning task, it is necessary to split the data into a train and test set: the former is used to train the model, and the latter resembles out-of-sample observations for prediction. The model's accuracy is the model's prediction accuracy on the test data after training. For this project, the data is split into 90% training and 10% testing given the small size of the dataset; however, the user can tweak the split size as a parameter of the pipeline. Additionally, I stratify the CRT scores during data splitting to ensure there is a representative distribution of CRT scores in both the train and test sets.

3.3.2 Vectorization for Text Features

Machine learning models often take in vectors of numerical data as input. When dealing with text data such as Tweet content or domain names, it is necessary to convert the text into a numerical format in order to perform the machine learning task. There are many methods from Natural Language Processing (NLP) for converting text into numerical vectors, which is formally called vectorization. For this project,

I employ a widely-used feature vectorization method called Term Frequency-Inverse Document Frequency (TF-IDF), which builds on basic vectorization algorithms like bag-of-words by incorporating context and relative frequency of words.

TF-IDF is useful for extracting the most descriptive words from a text by measuring the relevance and uniqueness of the word to the specific text versus the entire text collection. The intuition behind the TF-IDF method is that words found more situationally are more likely to be representative of the individual document's content. More specifically, term frequency (TF) refers to the frequency of a particular term in an individual docment, and inverse document frequency (IDF) is how common a particular term is across the entire corpus of documents. Multiplying these two values together produces the TF-IDF value of each word in the corpus. A high TF-IDF value indicates that a word has a high frequency in a document (e.g. all Tweets from one user) and a low frequency in the entire corpus (e.g. all Tweets from all users in the study). For this project, I use the sklearn module `TfidfVectorizer` to perform TF-IDF vectorization.

Parameters: `min_df` and `max_df`

One of the most important parameters of `TfidfVectorizer` is the `min_df` parameter, which is used to remove words that appear too infrequently in the corpus. The `min_df` can be set to a a float between 0 and 1 or an integer value.

- **Float**: If `min_df` = 0.05, words that appear in less than 5% of the documents in the corpus are ignored.
- **Integer**: If `min_df` = 5, words that appear in less than 5 documents in the corpus are ignored.
- **Default**: The default `min_df` value is 1, which means no words are ignored.

The `min_df` parameter has a strong influences in the model's accuracy because words that are less common across different users are less helpful for overall prediction. For instance, the model is able to learn much more from a group of 100 low-CRT score

users who have all included the word "sweepstakes" in one of their Tweets versus one high-CRT score user who frequently Tweets "zozo," which is a made-up word. In the first example, it is helpful for the model to learn the word "sweepstakes" is associated with low CRT score since the word has a high frequency in the corpus. In the second example, it is not very helpful for the model to learn that "zozo" is associated with high CRT score since the word has a low frequency in the corpus. The `min_df` parameter takes care of these less predictive words by setting a threshold for the number or percentage of documents (the set of Tweets from one user) that must include the target word. Otherwise, the word is discarded from the vectorization.

For this project, I set the `min_df` to 10, which excludes any words that appear in less than 10 documents (about 1% of the Complete dataset containing $n = 926$, further described in Chapter 4). A complementary parameter, `max_df`, removes words that appear too frequently in the corpus. For example, a `max_df` set to 500 means words that appear in more than 500 documents in the corpus are ignored. I set `max_df` to 0.99, ignoring any words that appear in more than 99% of the documents in the corpus.

Parameter: `ngram_range`

The `ngram_range` parameter defines the lower and upper boundary of the range of n-values for different n-grams to be extracted from the documents. A `ngram_range` of (1, 1) means only unigrams are extracted (terms of 1 length) while a `ngram_range` of (2, 3) means bigrams and trigrams are extracted (terms of lengths 2 or 3). Since most of the text features are a string of distinct terms where sequential terms are not necessarily related, the n-gram range is set to the default (1, 1) for the model; however, I experiment with n-gram ranges of (2, 2) for text, follower bios, and followee bios in the informative features module, described in Chapter 5.

Parameter: `binary`

The `binary` parameter determines whether the term frequency output is binary. If `binary` is set to True, all non-zero term counts are set to one (i.e. 1 if the term

exists in the document, 0 otherwise). For sparse features such as domains, hashtags, followees, and mentions, I set `binary` to True. For more content-rich features such as text, followee bios, follower bios, and bio, I set `binary` to False, causing the term frequency output to represent the actual frequency of a particular term in a user's set of Tweets.

Parameter: `token_pattern`

The `token_pattern` parameter is a regular expression defining what a valid token (term) is. The default regular expression selects tokens of two or more alphanumeric characters. For this model, I use the default regular expression for bio, domains, hashtags, followees, and mentions. I use a custom regular expression for text, followee bios, and follower bios of two or more letters (A through Z) to focus the vectorization of these content-rich features primarily on words.

3.3.3 Dimensionality Reduction for Text Features

Once the text features are vectorized, they are in a regression-friendly format; however, the feature space created through vectorization is highly-dimensional. In other words, there is a large number of unique terms across each of the text features in the dataset. Each unique word yields a different TF-IDF score for each user. As such, the vectorized data has a large number of dimensions: one for every unique term in the entire corpus (e.g. all unique words Tweeted or all unique hashtags in the dataset).

High-dimensional data requires more compute power to process and a larger storage space. Notably, high-dimensional data can lead to over-fitting of the model. As the dimensionality of a dataset increases without adding more training data, the feature space becomes more and more sparse. Sparse data makes it easier for the model to find an optimal solution because the model fits to the noise of the train data. Since the model learns patterns during the training phase, a model overfit on train data will not be general enough for predicting on out-of-sample data (i.e. test data). One method to reduce the dimensions of the feature space is through dimensionality reduc-

tion: a process to condense data from a high-dimensional space to a low-dimensional space while still preserving meaningful properties of the data.

In this project, I use the TruncatedSVD method for dimensionality reduction to reduce the size of the feature space produced from TF-IDF vectorization. Regular Singular-Value Decomposition (SVD) uses matrix factorization to convert the original matrix into a lower-rank matrix and is useful for data with a large number of features (e.g. more features than participants). TruncatedSVD operates on a similar premise; however the user can specify the number of output columns of the new matrix whereas regular SVD always outputs a reduced matrix with n columns for a given n by m matrix. In this way, TruncatedSVD truncates the number of output columns to a user-specified number, giving the user more freedom in determining the final feature space. Instead of using regular SVD to produce matrices with a column space equivalent in size to the number of participants, which is still quite large ($n > 900$ for the Complete dataset), I use TruncatedSVD to reduce the dimensions of the column space even further to a value k where $k << n$. Importantly, any dimensionality reduction method is useful for sparse data, such as the vectorized output of TF-IDF, to improve efficiency and prevent the model from over-fitting to noise.

For this project, I use the `TruncatedSVD` module from sklearn for dimensionality reduction and set the `n_components` parameter to 50, which reduces the size dimensions of the TF-IDF matrix to 50 components. Additionally, I set the `random_state` to be 17 to maintain reproducibility across multiple function calls.

3.3.4 Feature Selection

Another way to reduce dimensionality of a dataset is through feature selection: a process to reduce a feature space by selecting a subset of features for the model. Similar to the benefits of TruncatedSVD, feature selection simplifies the model and reduces variance (i.e. noise) in the train data to improve the model's ability to generalize. There are three main methods for feature selection: the filter method, the wrapper method, and the embedded method.

1. **Filter method**: The filter method chooses a subset of features based on its correlational relationship to the target (e.g. Pearson's r correlation). This method, while straight-forward and fast, is a pre-processing step and ignores the interaction with the regression model. Additionally, the filter method does not deal with multicollinearity as each feature is considered independently, which is particularly relevant to the dataset of this project.

2. **Wrapper method**: The wrapper method passes a subset of features through the model and based on the performance of the model, adds or removes features from the subset. Since the algorithm works by searching through all sequential combinations of features, this method can be computationally expensive.

3. **Embedded method**: The embedded method combines the two methods by selecting features using algorithms with built-in feature selection — that is, using algorithms that naturally penalize (shrink) less influential features, which is called regularization. There are two main types of regularization methods: L2 (Ridge) and L1 (LASSO). In L2 regularization, the coefficients of less predictive features are minimized but never set to zero, which is useful to prevent over-fitting since all features are included. In L1 regularization, the coefficients of less predictive features are set to zero and dropped from the model, which is useful when there are a large pool of features to choose from.

Embedded Method with ElasticNet

For this project, I use the embedded method for feature selection and use a third form of regularization that gets the best of both regularizations: ElasticNet. ElasticNet is the model of choice for this project because it balances L1 and L2 regularizations by tuning the hyper-parameter α. ElasticNet performs L2 regularization in the case of $\alpha = 0$ and L1 regularization in the case of $\alpha = 1$. After the model penalizes features with less predictive coefficients, features are selected by dropping those with coefficients below a certain threshold. In order to select about 50% of the features, I set the threshold to be the median coefficient value. Typically, the embedded

method runs feature selection and model training in parallel such that the feature selection model is equivalent to the prediction model; however, I choose to to build two separate modules for feature selection and prediction in order to select features using ElasticNet but experiment using other models for final prediction (e.g. Random Forest Regressor). For this project, I use the `ElasticNetCV` module from sklearn which contains the ElasticNet model with cross-validation.

3.3.5 Prediction

After the data has been transformed and reduced, it enters the prediction phase where a regression model of choice is fitted to the transformed train data with a user-specified cross-validation strategy, and the fitted model predicts CRT score on the test data. To determine how close the model's prediction is to the actual CRT score, I calculate Pearson's r correlation coefficient and the respective p value between the two. Pearson's r values closer to -1.0 and 1.0 indicate a stronger correlation between predicted and actual CRT score and for r values closer to 0.0, a weaker correlation. Taking the absolute value of Pearson's r allows us to compare the strengths of correlations, since direction of correlation (i.e. positive or negative) is not the focus of this work. There are three models I explore in this project in the prediction phase.

1. **Ridge**: In this project, I use the Ridge regression model as the primary model for training and prediction. The data suffers from multicollinearity (i.e. there are high correlations between two or more of the independent variables) which can be problematic when explaining the relationships between variables. As such, the Ridge model is a standard choice to reduce the magnitude of those correlations by using L2 regularization which penalizes the square of the magnitude of the coefficients, reducing the impact of correlated variables.

2. **LASSO**: LASSO is another regression model that uses shrinkage of coefficients, similar to Ridge. Instead of L2 regularization, LASSO uses L1 regularization which penalizes the absolute value of the magnitude of coefficients. LASSO is especially useful in cases where there are a large number of features since less

predictive features will shrink to zero (e.g. in the case where $n >> m$ for n features and m observations, LASSO picks at most m predictors).

3. **Random Forest Regressor**: Random Forests is a regression model that builds decision trees from subsets of the data. A more robust version of regular Decision Trees regression, Random Forests averages the outputs of many decision trees to increase prediction accuracy and reduce over-fitting. Another key feature of Random Forests is that the algorithm considers features sequentially such that large forests of decision trees automatically capture interactions between features, unlike Ridge and LASSO*. At the cost of model fitting time, Random Forests is able to discover more complex relationships between variables, and its non-linear nature makes it a powerful option for this dataset.

For this project, I use the sklearn modules `Ridge`, `Lasso`, and `RandomForestRegressor` respectively for the models described above. Additionally, I choose to predict CRT score using 5-fold cross-validation for this project.

**While Ridge and LASSO do not naturally learn interactions between features, I explicitly compute polynomial combinations between features up to a specified degree using a sklearn module* `PolynomialFeatures`. *The polynomial combinations are passed as input data into the model for combined features prediction, described later in Chapter 5.*

3.4 Choices in Code Design

Because the main contribution of this thesis is computational, I intentionally designed the code with the following goals in mind to promote robust scientific practices.

1. **Goal 1: Provide clear documentation**

   To help users understand (1) what a section of the program does and (2) how to use the pipeline without having to read through and decipher the written code.

- Function descriptions, where each description contains: (1) A general explanation of what the function does, (2) all inputs and input types, and (3) all outputs and output types. Where helpful, code comments are added to explain sub-sections of functions (Figure 3-1).
- High-level descriptions that include the goal of the Jupyter notebook and directions on how to use the Jupyter notebook written as a heading at the top of each file (Figure 3-2).

2. **Goal 2: Ensure repeatability**

   To help users (1) reproduce the same outputs generated by the code creator and (2) catch mistakes more easily since long programs are segmented into manageable chunks of code.

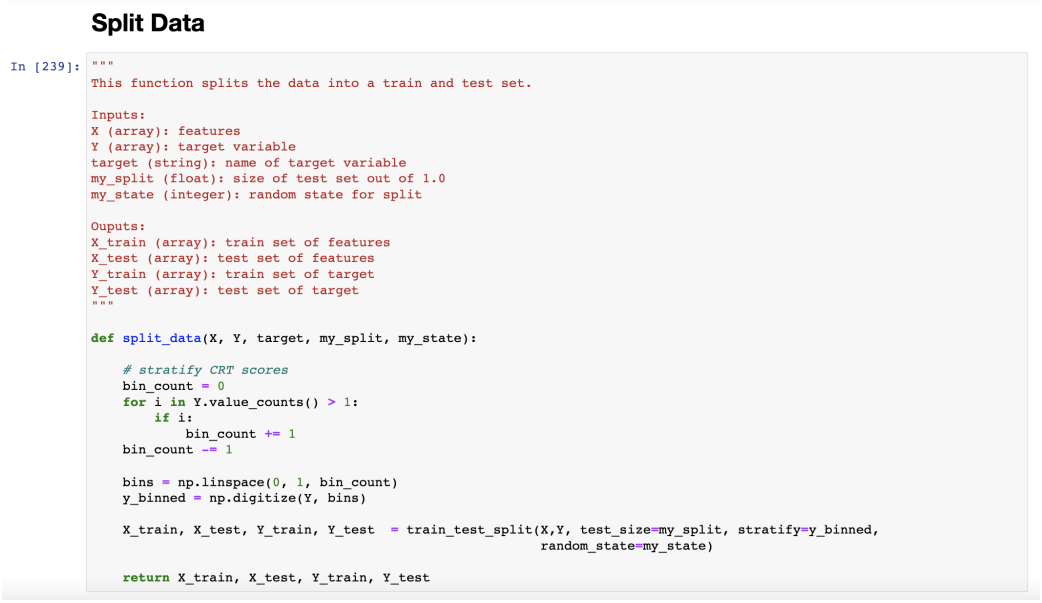
   - All code is written in Jupyter notebook, which is a web-based interactive environment to write, execute, and share code easily in the form of notebooks. Jupyter notebook allows the programmer to split up entire programs into snack-size scripts for the user.
   - When possible, random states are set to a fixed value for reproducible results across multiple function calls (e.g. `TruncatedSVD` and `Ridge` model).

3. **Goal 3: Cater to diverse usership**

   To help users gain (1) a descriptive overview of the end-to-end pipeline without diving into the code and (2) design freedom for specific pipeline modules.

   - Created a visual high-level diagram of pipeline that includes descriptions of modules, pipeline flow, key inputs and outputs, and key file names and locations (See Appendix A for diagram). *Target audience: scientific and non-scientific audiences.*
   - Created detailed, visual diagrams showing the inputs, outputs, and process of specific individual modules. *Target audience: scientific audiences.*

- Where possible, the module implementation is black-boxed, but the user can decide which parameters to apply in certain modules without knowing the specific code implementation.

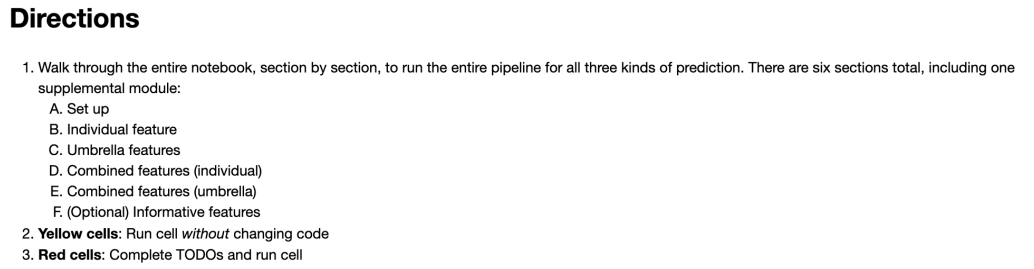


Split Data

In [239]:

```
"""
This function splits the data into a train and test set.

Inputs:
X (array): features
Y (array): target variable
target (string): name of target variable
my_split (float): size of test set out of 1.0
my_state (integer): random state for split

Ouputs:
X_train (array): train set of features
X_test (array): test set of features
Y_train (array): train set of target
Y_test (array): test set of target
"""

def split_data(X, Y, target, my_split, my_state):

    # stratify CRT scores
    bin_count = 0
    for i in Y.value_counts() > 1:
        if i:
            bin_count += 1
    bin_count -= 1

    bins = np.linspace(0, 1, bin_count)
    y_binned = np.digitize(Y, bins)

    X_train, X_test, Y_train, Y_test  = train_test_split(X,Y, test_size=my_split, stratify=y_binned,
                                                         random_state=my_state)

    return X_train, X_test, Y_train, Y_test
```

Figure 3-1: Screenshot of function from code providing function description, inputs, outputs, and data types.



Directions

1. Walk through the entire notebook, section by section, to run the entire pipeline for all three kinds of prediction. There are six sections total, including one supplemental module:
   A. Set up
   B. Individual feature
   C. Umbrella features
   D. Combined features (individual)
   E. Combined features (umbrella)
   F. (Optional) Informative features
2. **Yellow cells**: Run cell *without* changing code
3. **Red cells**: Complete TODOs and run cell

Figure 3-2: Screenshot of top of the Jupyter notebook for running the pipeline.

Chapter 4

Dataset

The raw data used for this project was graciously collected and shared with me by Mohsen Mosleh of the primary cited study (Mosleh et al., 2021). The complete suite of data merges the data pull from the original study and the latest data pull. The former is a merge of two older data pulls, one on August 18th, 2020 and one on April 12th, 2020; the latter contains the latest Twitter activity of the still-active participants from the original study, completed on November 18th, 2021. The data pulls were completed using the Twitter API and include user information such as Tweet content, basic user profile information, follower data, accounts followed data, domains shared, mention history, and hashtag history.

In this section, I describe the dataset, including the target features selected, exclusions applied, preliminary statistics, and pre-processing protocols.

4.1 Target Features

The raw data has a large feature space, containing hundreds of Twitter features and many more via self-generated companion features. Because the raw data is high-dimensional, it is important to reduce the feature space through a variety of methods to reduce information redundancy, eliminate less predictive features, and improve model efficiency and accuracy. In Chapter 3, I discussed three techniques for dimensionality reduction: TF-IDF (through choice of parameters), TruncatedSVD, and

feature selection using the embedded method with ElasticNet. Perhaps the least technical approach to reducing the feature space is by hand-selecting features from the raw data at the get-go. For this project, I choose features that have any of the following characteristics below.

1. Features which intuitively seem to be predictive of cognitive reflection (e.g. names of domain links shared)

2. Features which are information-rich (e.g. Tweet content)

3. Features which would be interesting to look at in regards to cognitive reflection (e.g. average number of followees of a user's followees)

From the raw data, I hand-select 49 features which encapsulate the variety in one's Twitter behavior and their overall digital fingerprint. The features are a combination of raw features pulled directly using the Twitter API and self-generated companion features, such as followee count and average length of hashtags. The following features are included in the set of selected features (see Appendix B for the full table of features).

- **Profile information**: Basic quantitative information from a user's profile (e.g. follower count, followee count, favorites count, URL count, etc.)

- **Emotional words**: Fraction of Tweets from a user that have at least one word from a standard word corpus like the Linguistic Inquiry and Word Count dictionary (e.g. positive words, negative words, etc.). *Pre-computed as part of Mosleh et al.'s study.*

- **Text**: Text from Tweets and Retweets the user has shared, capped at the most recent 3,200 Tweets by the Twitter API

- **Domains**: Domain names of websites a user has Tweeted or Retweeted

- **Followees**: Accounts a user follows

- **Hashtags**: Hashtags the user has Tweeted or Retweeted

- **Mentions**: Account names a user has mentioned in their Tweets or Retweets

- **Bios**: Twitter bio on user's profile (e.g. user bio, followees' bios, followers' bios)

4.2 Exclusions and Pre-Processing

Participants with no Twitter username or CRT score were excluded from the dataset as both are needed for the supervised machine learning task. Additionally, participants under 18 and participants with more than 7,000 followers were excluded from the data set: the former due to the COUHES's approved protocol and the latter due to some participants reporting Twitter screen names that were not theirs (i.e. accounts with an abnormally large number of followers). Only Tweets and Retweets in English were selected. The raw data is pre-processed before entering the pipeline in order to (1) reduce noise in the data and (2) generate companion features, as follows:

- **Text**: Tweets and Retweets are aggregated into a single string. Each item is separated by a space and pre-processed with simple regex under the following conditions adapted from the Tweet2Vec model: (1) remove HTML tags and hashtags, (2) replace mentions and URLs with special tokens, and (3) convert text to lowercase (Dhingra et al., 2016).

- **Mentions and hashtags**: Extracted item from token (@ for mentions and # for hashtags) and aggregated items into a single string. Each item is converted to lowercase, and items are separated by a space (keep repeats).

- **Domains**: Extracted domain name from full URL using the Python package `tldextract` and aggregated into a single string. Each domain is converted to lowercase, and items are separated by a space (keep repeats). Removed 'twitter' and 't' domain names, which redirect to another Tweet.

- **Followees**: Aggregated into a single string. Each item is converted to lowercase, and items are separated by a space.

- **Bio, followee bios, and follower bios**: Pre-processed under the same conditions as the Text feature.

- **Quantitative features**: (1) Write short scripts to generate companion features from the raw data (e.g. number of digits in screen name, average number of followers of a user's followees), (2) convert all boolean features to binary 0/1 (e.g. user has profile image, user has chosen to protect their Tweets), and (3) convert all quantitative features to floats.

The input to the pipeline is the pre-processed dataframe containing the 49 target features and the target (CRT score). The dataframe is created as a Pandas dataframe using the `pandas` package and stored as an Apache Parquet file for space efficiency using the `pyarrow` package.

4.3 Preliminary Statistics

The pre-processed dataset can be described with the following statistics:

- **Number of valid participants** (i.e. non-excluded participants who provided a Twitter handle and completed the CRT): 1,808 users.

- **Number of valid participants after dropping NaN values** (i.e. valid participants who have data for every target feature): 926 users.

- **Number of target features**: 49 total features (41 quantitative features and 8 text features).

For this project, I refer to the dataset with all valid participants as the **Full dataset** and the dataset with all valid participants after dropping NaN values as the **Complete dataset**. The figures below describe the age distribution, CRT score distribution, and basic statistics on profile features for the Full and Complete datasets.

The distribution of age follows a similar pattern for both the Full and Complete datasets with a peak at the 25-30 year range and a steady decrease in the 40-80 year range. The distribution of CRT score also follows a similar pattern for both datasets

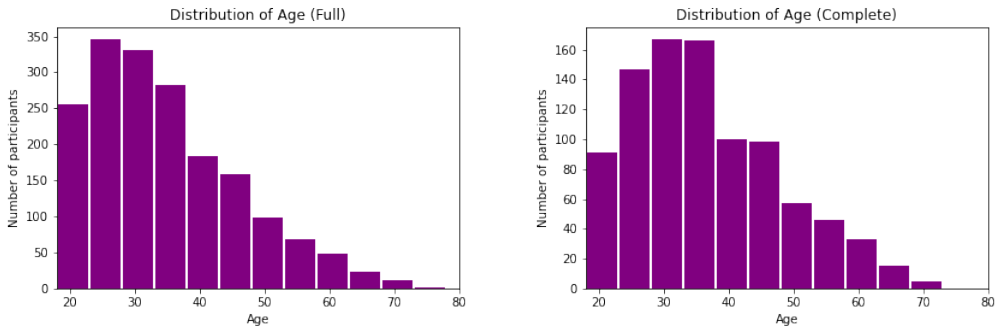


Figure 4-1: Distribution of participant age for Full and Complete dataset

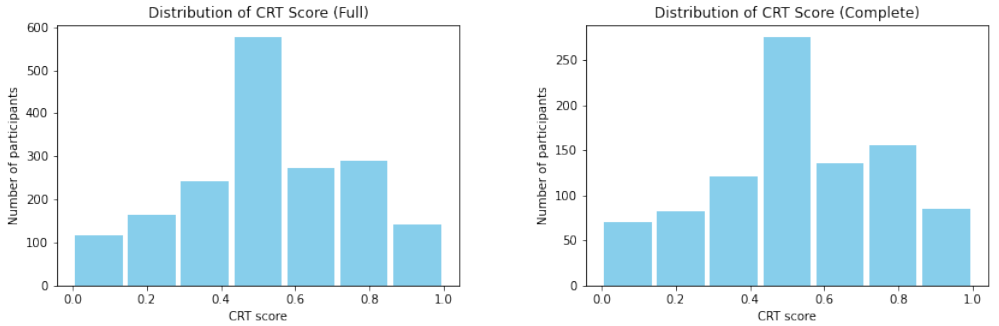


Figure 4-2: Distribution of CRT scores for Full and Complete dataset. A CRT score of 1.0 means the user answered 7 out of 7 questions correctly.

with a peak in the 0.4-0.6 score bucket. The mean values across each of the profile features listed in Table 4.1 are similar between the two datasets, with the Complete dataset means slightly higher than that of the Full dataset. The Full dataset contains more extreme minimum and maximums, likely due to containing double the number of participants.

	Full			**Complete**		
	Mean	Min	Max	Mean	Min	Max
CRT score	0.54	0.0	1.0	0.54	0.0	1.0
followees count	477.43	0.0	10,153.0	569.78	1.0	10,153.0
followers count	235.74	0.0	6,314.0	255.92	0.0	5,918.0
statuses count	4,304.61	0.0	513,248.0	4,796.77	0.0	513,248.0
hashtags count	830.37	0.0	27,974.0	1,159.59	0.0	27,974.0
domains count	312.87	0.0	9,377.0	406.03	0.0	9,377.0
mentions count	884.72	0.0	20,221.0	1,176.0	0.0	16,676.0
days on twitter	2,355.28	157.0	4,464.0	2,441.96	194.0	4,464.0

Table 4.1: Preliminary statistics on profile features in Full and Complete dataset

4.4 Data Dimensionality

In Chapter 3, I described the importance of TF-IDF vectorization and its parameters, such as `min_df` and `max_df` which I set to 10 and 0.99 respectively for the model. Setting the two parameters as such, I reduce the dimensions of the Complete dataset across the text features to the values listed in Table 4.2. The dimensions of the data are calculated by counting the number of output feature names after vectorization.

Text Feature	`min_df` = **1**	`min_df` = **10**	**% of Total**
Followees	327,341	5,372	1.6%
Domains	30,979	1,176	3.8%
Mentions	217,578	3,246	1.5%
Hashtags	138,352	3,946	2.9%
Follower Bios	82,248	8,911	10.8%
Followee Bios	164,975	21,781	13.2%
Text	194,227	31,410	16.2%
Bio*	2,513	180 (*`min_df` = 5)	7.2%

Table 4.2: Number of dimensions produced from TF-IDF vectorization on the Complete dataset using `min_df` = 1 versus `min_df` = 10, keeping `max_df` at 0.99 for both. The last column shows the percentage of the total dimensionality preserved when `min_df` = 10.

After TF-IDF vectorization, the dimensions of the text features are reduced by 84-98%. The text feature with the most dimensions prior to vectorization is followees, likely due to the sparsity of the followee network. The text features with higher density — text, followee bios, follower bios, and bios — are reduced the least from vectorization because of the repeated use of terms across users. Sparse features like followees, domains, mentions, and hashtags are reduced more aggressively after vectorization due to less term repetition across users.

TruncatedSVD is another method I use for dimensionality reduction. Setting the `n_components` parameter to 50 as described in Chapter 3, I achieve the explained variance totals for each text feature in the Complete dataset shown in Table 4.3. The explained variance is the percentage of variance explained by each of the components produced from TruncatedSVD and is calculated using an attribute of the `TruncatedSVD` module. The total explained variance is the summation of the ex-

plained variance percentage by each of the selected components.

Text Feature	Total Explained Variance
Followees	23%
Domains	38%
Mentions	25%
Hashtags	23%
Follower Bios	24%
Followee Bios	38%
Text	42%
Bio	64%

Table 4.3: Explained Variance after TruncatedSVD with `n_components` = 50 on the Complete dataset

The most variance is explained for the bio feature, which contains the fewest dimensions after vectorization. The rest of the text features have a total explained variance between 23-42%. Increasing the value of `n_components` increases the total explained variance; however, there is a trade-off in the number of components and the efficiency and accuracy of the model. First, increasing the number of components will increase the pipeline run-time. Second and more importantly, to create a model that generalizes for new, out-of-sample observations, it is important to choose a value of `n_components` that explains some of the variance in the current dataset but not all (100%) to prevent over-fitting.

4.5 Umbrella Features

Individual features can be combined to generate umbrella features: feature groups containing related features. For instance, hashtags, hashtags count, and hashtags average length all contain information related to hashtags and are grouped into one umbrella feature. The complete set of umbrella features is pictured in Figure 4-3 (See Appendix B for a complete description of each individual feature). For this project, I explore the predictive power of individual features, umbrella features, and combinations of features (individual and umbrella).

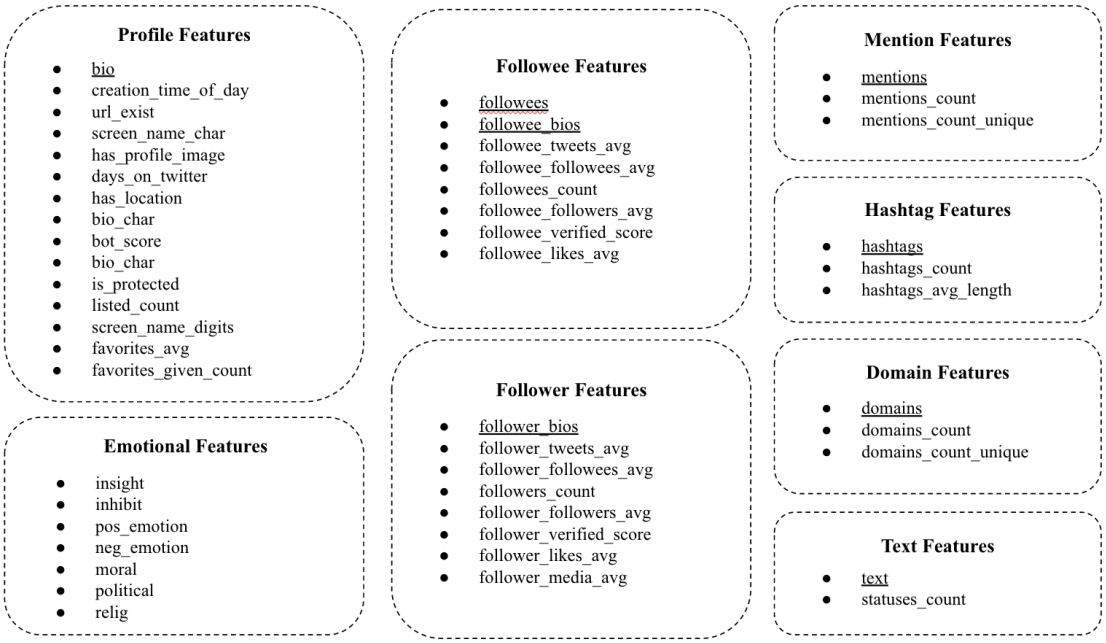


Figure 4-3: Umbrella feature categories. Text features are underlined.

Chapter 5

Pipeline

The core component of this project is the machine learning pipeline. The pipeline code can be found at the following GitHub repository:

`https://github.com/jimenezatmit/crt_prediction`.

In the following section, I describe the pipeline structure and implementation, which builds off the machine learning model components explained in Chapter 3.

5.1 Overview

At a high-level, the complete pipeline includes three major phases: transformation, selection, and prediction. First, the dataset is transformed through vectorization, dimensionality reduction, and scaling. Then, the data is split into a train and test set in order to resemble out-of-sample data as closely as possible. The train data then goes through feature selection, and the results from feature selection inform which features are selected for prediction. Finally, the data goes through the prediction phase where a regression model of choice learns patterns in the train data to predict CRT score in the test data. The prediction phase generates a Pearson's r correlation coefficient between predicted and actual CRT score, a p value, and a R^2 value. Individual features, umbrella features, and combined features prediction follow a similar high-level structure. The pipeline is written as a single Jupyter notebook such that the

user can run the entire pipeline by running each cell in the notebook start-to-finish.

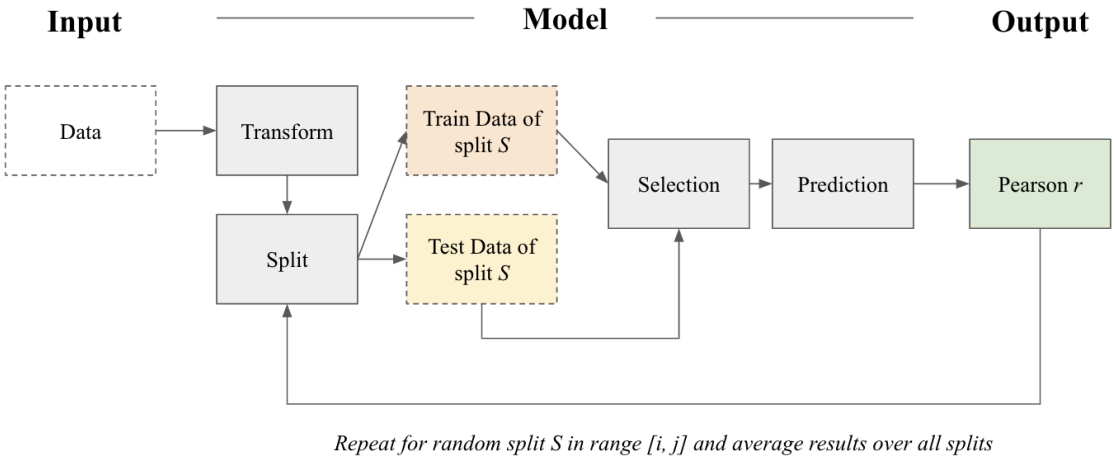


Figure 5-1: High-level schematic of pipeline.

Due to the small size of the dataset, there could be large variance in each split of train and test data. As such, I calcualte the r value from prediction over multiple splits and select the median r value to represent the overall prediction accuracy. For this project, I choose to run the pipeline over 50 unique splits of data.

5.2 Model

The core component of this pipeline is the model itself, which learns patterns and relationships between the target features and CRT score to predict an unseen user's CRT score. The model is built in a separate Jupyter notebook as the pipeline, and the pipeline imports all relevant model functions in order to black-box the specific code implementation frmo the user. In this section, I describe the three phases of transformation, selection, and prediction.

5.2.1 Transformation

In this phase, I transform all features and create a master dataframe. Numerical features with a high skew value (less than -0.5 or greater than 0.5) are log

transformed, and all numerical features are standardized using the sklearn package `StandardScalar`. Text features are transformed to numerical vectors using the sklearn module `TfidfVectorizer` (parameters listed in Table 5.1) and further reduced using sklearn's `TruncatedSVD` module.

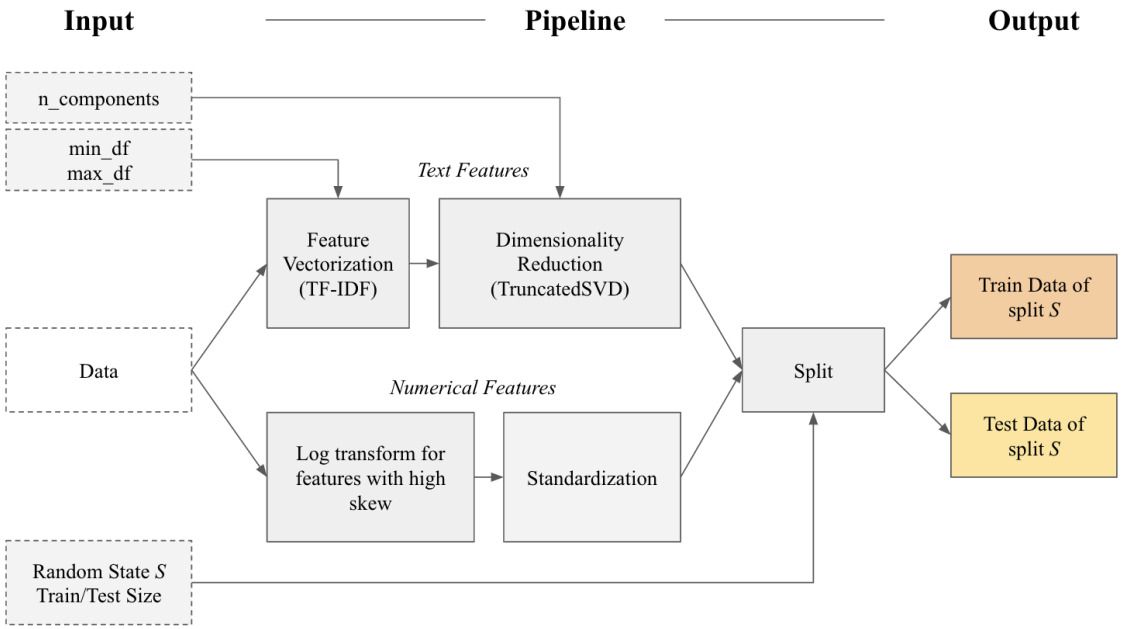


Figure 5-2: Transformation phase of pipeline

	Text, Follower Bios, Followee Bios	**Bio**	**Followees, Domains, Hashtags, Mentions**
`min_df`	10	5	10
`max_df`	0.99	0.99	0.99
`ngram_range`	(1, 1)	(1, 1)	(1, 1)
`binary`	False	False	True
`token_pattern` (RegEx)	2+ letters	2+ letters	2+ alphanumeric characters

Table 5.1: `TfidfVectorization` parameters for text feature transformation

5.2.2 Feature Selection

In this phase, the transformed features go through the embedded feature selection process using the sklearn module `ElasticNetCV`. During this phase, `ElasticNetCV` penalizes the coefficients of each feature depending on its predictive power on the train set and drops the features with coefficients below the median coefficient value in both the train and test set. The output is the train and test sets after feature selection, and the column dimensions of the train and test sets are equal (i.e. the same features are selected from both the train and test sets).

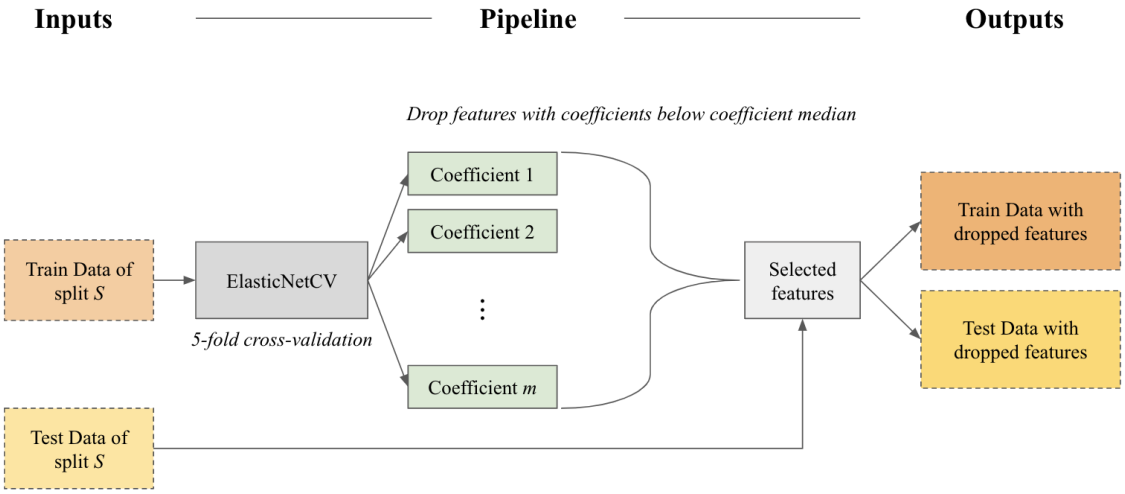


Figure 5-3: Selection phase of the pipeline

5.2.3 Prediction

In the prediction module, the user selects a regression model of choice — either Ridge, LASSO, or Random Forests — to fit and predict on the dataset using the following protocol.

1. Using the sklearn module `GridSearchCV` for hyperparameter tuning, fit the model of choice to the train data.

2. Predict CRT score of out-of-sample observations using the test data.

3. Compute Pearson's r between predicted and actual CRT score and the respective p value and R^2 score using the `pearsonr` package from scipy.

Individual Features

There are three types of prediction for this project. First, I predict CRT score on the test data using a single feature — that is, one of the 49 target features. I compute Pearson's r between predicted and actual CRT score for each split and select the median r value from all 50 splits to represent the r value for the individual feature. For clarity, the flow described in Figure 5-4 shows the prediction process for one split of data.

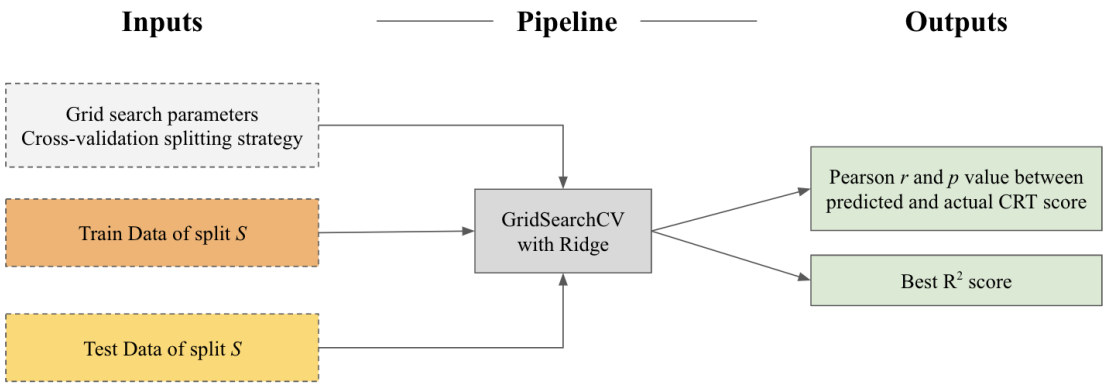


Figure 5-4: Individual and umbrella feature prediction

Umbrella Features

Umbrella feature prediction follows a very similar structure to that of individual feature prediction above, except that each input feature is a concatenation of all the individual features under each umbrella feature, described in Chapter 4.

Combined Features

While individual features hold some predictive power over CRT score, combining features can improve prediction accuracy. By combining features, I create the best

model for CRT score prediction given the current dataset. The following protocol is used to combine individual features.

1. Predict CRT score using all features.
2. Remove the individual feature with the highest median p value across splits.
3. Repeat Steps 1 and 2, removing one feature at a time, until one feature remains (i.e. feature with the lowest median p value).

Umbrella features are also combined using a similar protocol as above, but I remove the umbrella feature with the highest median p value with each iteration.

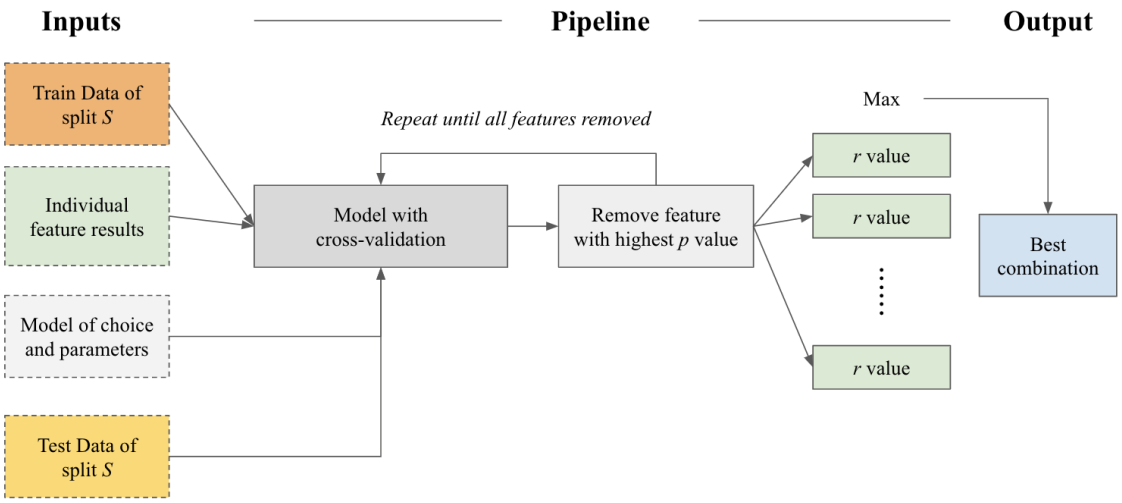


Figure 5-5: Combined feature prediction

5.3 Informative Features

This standalone module takes in any text feature as input and displays the top k informative features for high CRT scores and low CRT scores. This module applies to domains, mentions, followees, hashtags, text (Tweets and Retweets), bio, followee bios, and follower bios. The parameters listed in Table 5.2 are used for `TfidfVectorization` in this module.

	Text	Followees, Domains, Hashtags, Mentions	Follower Bios, Followee Bios	Bio
`min_df`	10	10	10	5
`max_df`	0.99	0.99	0.99	0.99
`ngram_range`	(1, 1), (2, 2), (3, 3)	(1, 1)	(1, 1), (2, 2), (3, 3)	(1, 1)
`binary`	False	True	False	False
`token_pattern` (RegEx)	3+ letters	2+ alphanumeric characters	2+ letters	2+ letters

Table 5.2: `TfidfVectorization` parameters for informative features

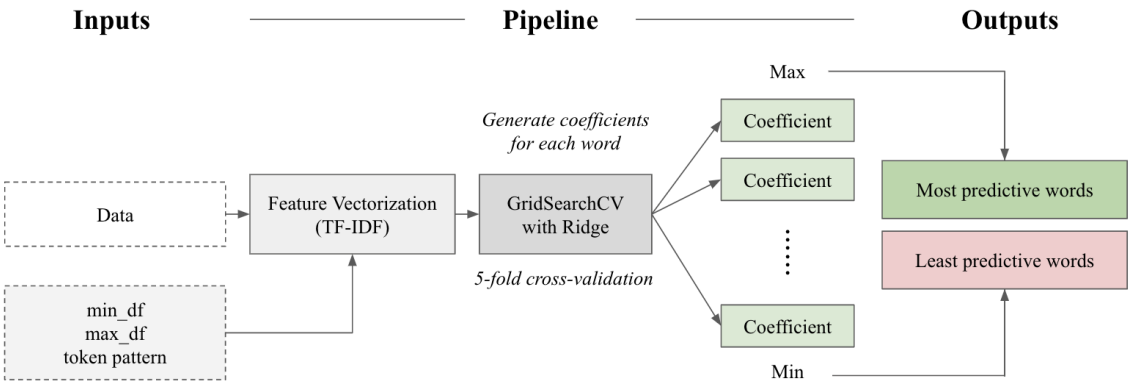


Figure 5-6: Informative features schematic

Chapter 6

Results

In this section, I describe the results generated from individual feature prediction, umbrella feature prediction, combined features prediction, and informative features using the Complete dataset which contains 926 users. For the individual, umbrella, and combined prediction results, the results are aggregated across the same 50 unique splits of data in order to compare performance across prediction categories. Additionally, I describe the results for individual feature prediction on the Full dataset, which contains all 1,808 users.

6.1 Individual Features

The plot below shows the results for the Pearson's r correlation coefficient between predicted and actual CRT score for each of the 49 individual features using the Ridge model with cross-validation. The r value shown is the median r value across the 50 splits, and the bars reflect the minimum r value and maximum r value across the splits (i.e. range). The features with an asterisk — followees and followee bios — are the features with statistically significant results (p median < 0.05). Text features are plotted in blue, and quantitative features are plotted in gray. The individual features with the strongest predictive power and statistical significance are followees (r median $= 0.25$, p median $= 0.015$) and followee bios (r median $= 0.24$, p median $= 0.022$). In general, the text features hold more predictive power than the quantitative features

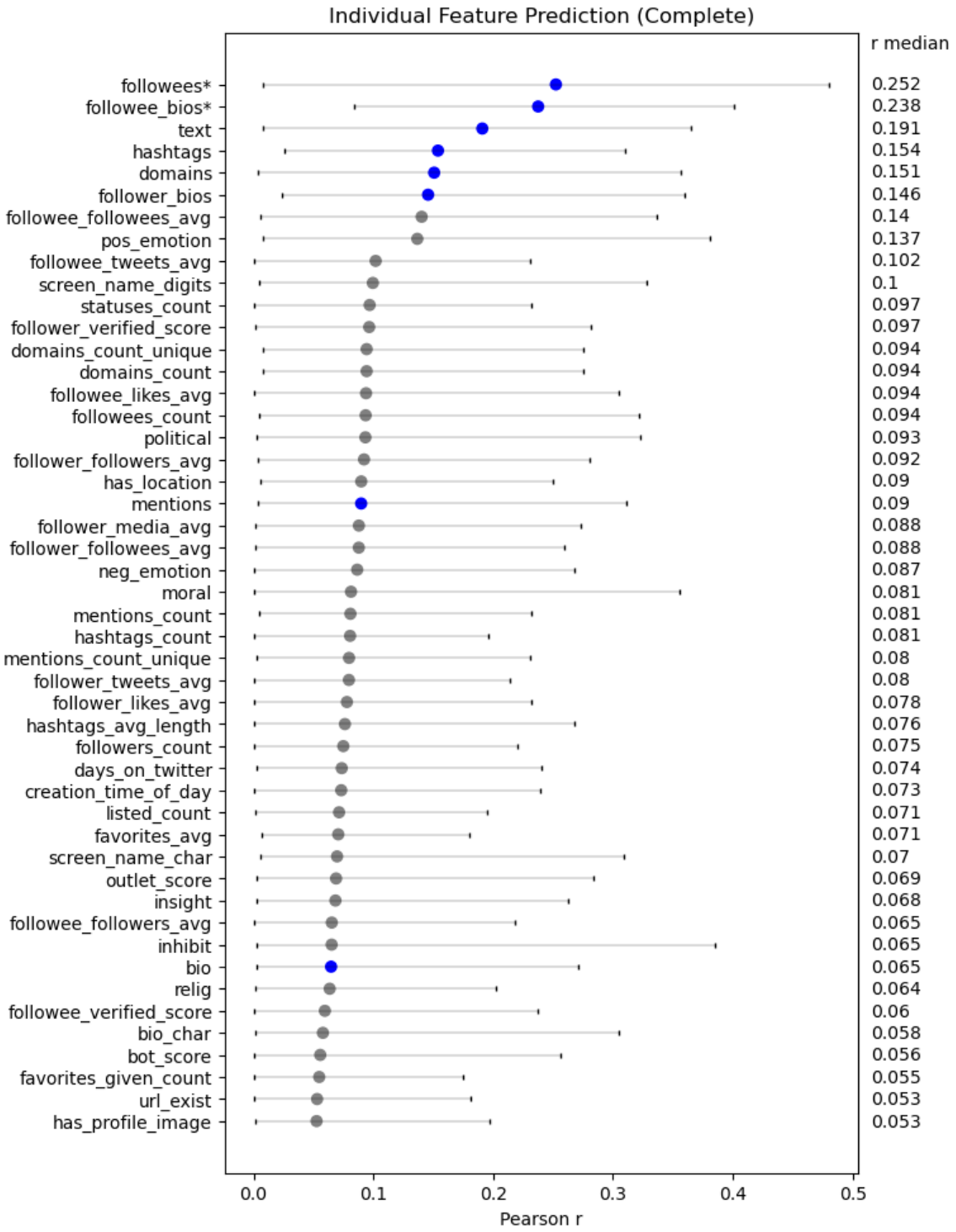


Figure 6-1: Individual Feature prediction on Complete dataset

because of their content density. Overall, it is expected that most of the features do not predict well individually since cognitive reflection is a complex behavior to capture given a single social media feature; however, although many of the individual features are not statistically significant, it does not necessarily mean these features cannot be predictive of cognitive reflection.

6.2 Umbrella Features

By combining individual features into umbrella features (i.e. related features), prediction accuracy increases. The plot below shows the results for Pearson's r correlation coefficient between predicted and actual CRT score for each of the 8 umbrella features using the Ridge model with cross-validation. The r value shown is the median r value across the 50 splits, and the bars reflect the minimum r value and maximum r value. The umbrella features with an asterisk — followee and text features — are the features with statistically significant results (p median < 0.05).

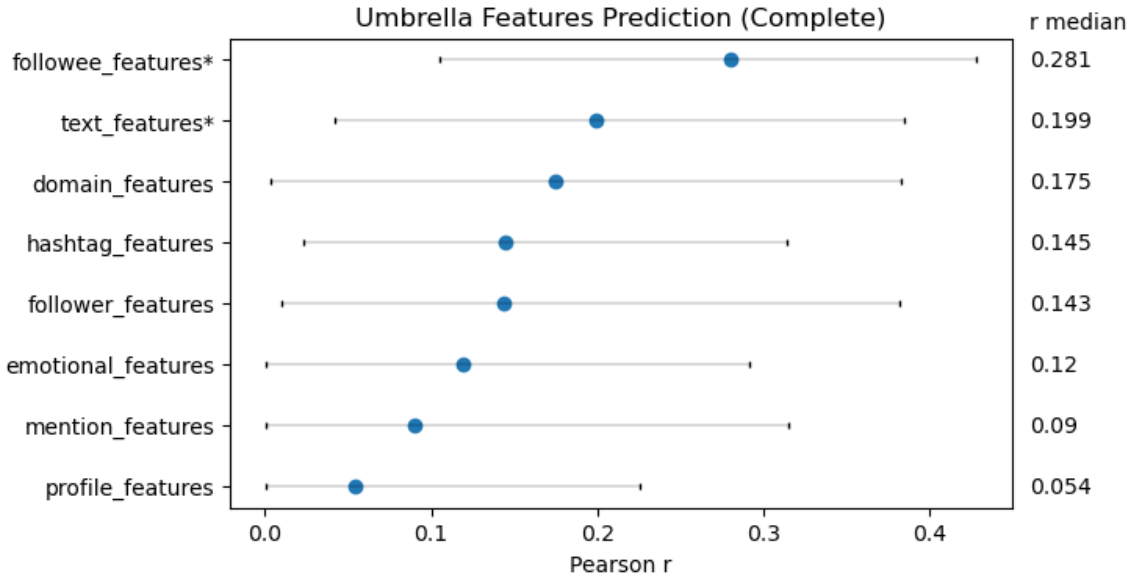


Figure 6-2: Umbrella feature prediction

The umbrella features with the strongest predictive power and statistical significance are followee features (r median = 0.28, p median = 0.006) and text features

(r median = 0.20, p median = 0.056). Although the rest of the umbrella features are not statistically significant, we can still determine their predictive power relative to one another. The remaining umbrella features from most predictive to least predictive are: domains features, hashtag features, follower features, emotional features, mention features, and finally, profile features.

6.3 Combined Features

Finally, I combine features and experiment with different models to generate the best-performing model. I combine features in two different ways: across individual features and across umbrella features. To determine the best combination of features, I run the combined features protocol described in Chapter 5.2.3 with the Ridge regression model and polynomial degree one. For the individual features, the best combination of features is followees and followee bios. For the umbrella features, the best combination of features is followee features, which contains the following individual features: followees, followee bios, followee Tweets average, followee followees average, followees count, followee followers average, followee verified score, and followee likes average. Next, I apply different models on the best combinations of features to determine which model yields the highest accuracy across 50 splits.

The plots below show the results for Pearson's r correlation coefficient for the best combination of features for individual and umbrella features using five models: Ridge with polynomial degree one (baseline), Ridge with polynomial degree two, LASSO with polynomial degree one, LASSO with polynomial degree two, and Random Forest Regressor. The r value shown is the median r value across the 50 splits, and the bars reflect the minimum and maximum r value. The model names with an asterisk are the runs with statistically significant results (p median < 0.05).

For combined individual features, the Random Forests model yields the highest accuracy with a r median of 0.28 (p median = 0.006). Depending on the split of data, the Random Forests model can yield a r value as high as 0.42 and as low as 0.11. For combined umbrella features, the LASSO model yields the highest accuracy with

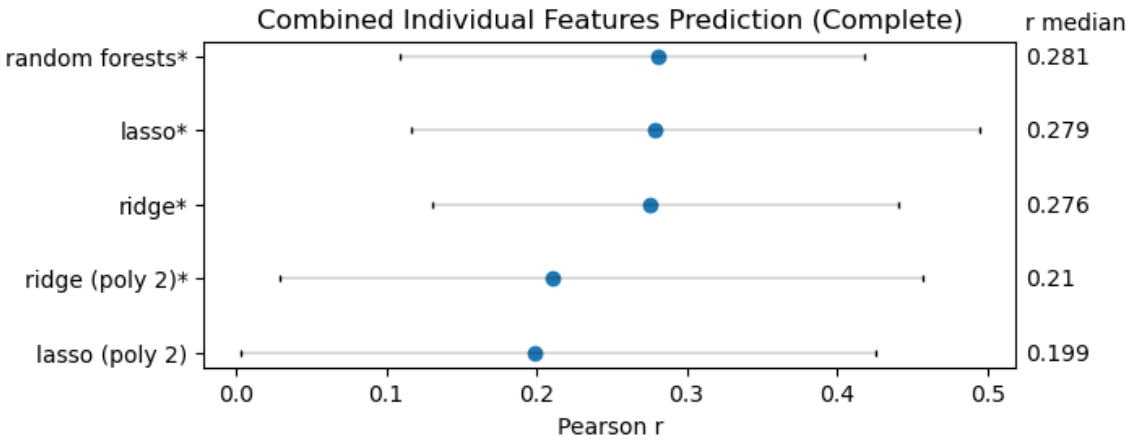


Figure 6-3: Combined individual feature prediction on Complete dataset

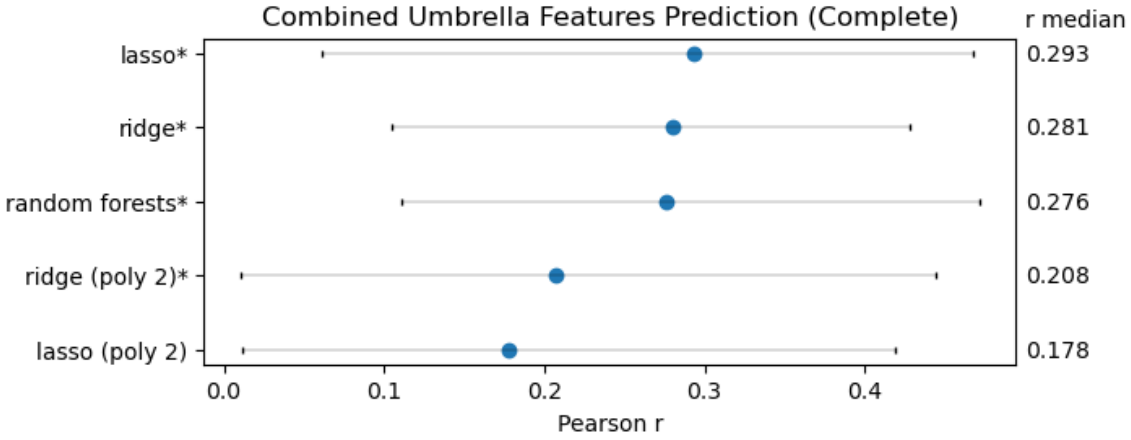


Figure 6-4: Combined umbrella feature prediction on Complete dataset

a r median of 0.29 (p median = 0.004). Depending on the split of data, the LASSO model can yield a r value as high as 0.47 and as low as 0.06. The large range indicates that the model accuracy is sensitive to the split of the data.

6.4 Prediction with the Full Dataset

The Complete dataset only contains about half the number of users in Full dataset (n = 926 vs. n = 1,808). To test the robustness of the model, I perform individual features prediction on the Full dataset. For each individual feature, I drop all the users with no value for that feature (i.e. NaN), which will be less than or equal 882:

the difference in number of users between the Full and Complete datasets. The results of individual features prediction with the Full dataset is shown below with a similar structure to that of individual features prediction on the Complete dataset.

For the Full dataset, the best-performing individual features with statistical significance are followee bios (r median = 0.25, p median = 0.002), followees (r median = 0.25, p median = 0.002), text (r median = 0.18, p median = 0.017), and domains (r median = 0.17, p median = 0.027). More individual features show statistical significance in the Full dataset than in the Complete dataset due to the greater number of observations in the Full dataset. The top two most predictive individual features are consistent between the Complete and Full datasets. To determine quantitatively whether adding new users affects accuracy, I calculate the Pearson's r correlation coefficient between (1) the number of new users added to the Full dataset for each feature and (2) the difference between the r median value in the Complete and Full datasets for each feature. The r value is 0.038, indicating there is no correlation between the number of added users and the change in r median value and that the model is robust to adding new data.

6.5 Informative Features

The results in Figures 6-6 to 6-13 show the most informative words across each of the text features in the Complete dataset — that is, the top fifteen words and phrases for high CRT score and low CRT score prediction. The coefficients listed are the term coefficients with the largest magnitudes after applying Ridge regression, which penalizes less predictive coefficients. Negative coefficients are associated with lower CRT scores (i.e. the CRT score decreases when the term is present), and positive coefficients are associated with higher CRT scores (i.e. the CRT score increases when the term is present). In the following subsections, I highlight some of the most predictive and most interesting informative terms and phrases across each of the text features.

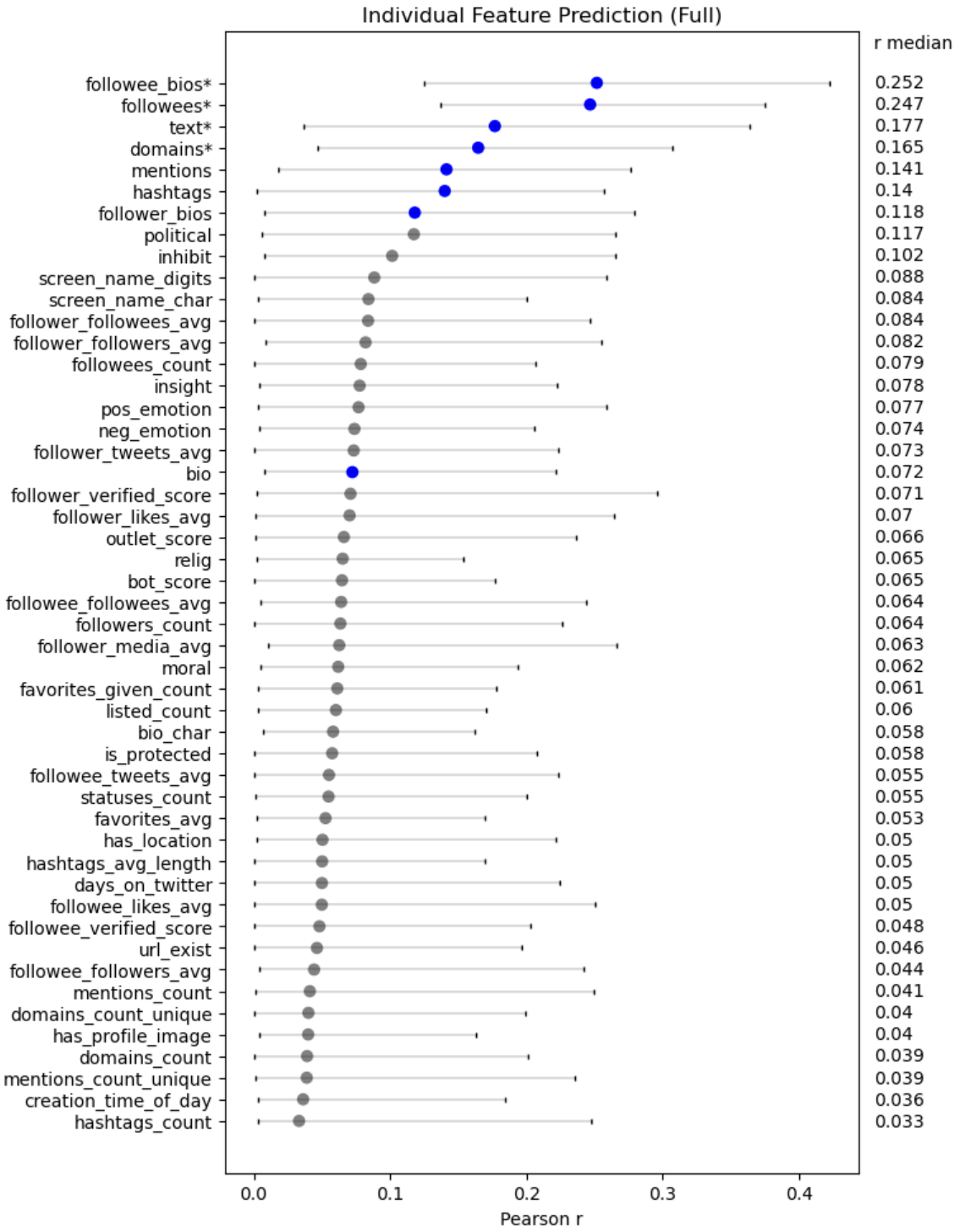


Figure 6-5: Individual feature prediction on Full dataset

Text

For the text feature, phrases related to giveaways like "entered giveaway," "simply follow," and "rewards" are predictive of lower CRT scores while phrases related to weightier subjects like "climate change," "brexit," and "autistic people" are predictive of higher CRT scores. Interestingly, "hillary clinton" is predictive of lower CRT scores and "international giveaway" is predictive of higher CRT scores.

Followee Bios

When observing the terms used in a user's followees' bios, less scholarly phrases like "producer," "instagram," "travel blogger," "surveys," "paid," and "opinionated" are predictive of lower CRT score while the more scholarly phrases "professor," "science," "nasa," "student," "author," and "cancer research" are predictive of higher CRT scores. Additionally, following accounts that include links or other social media accounts in their bio is predictive of lower CRT scores (e.g. "youtube https," "instagram user," "user instagram," "https https").

Follower Bios

Scholarly phrases found in follower bios are also predictive of higher CRT scores (e.g. "science," "student," "geek," "politics," "academic," "university," and "engineer"). Interestingly, "anti brexit" is the most predictive 2-word phrase for high CRT scores. Phrases associated with consumerism and blogging like "product," "online," "lifestyle blogger," "wedding," "social media," and "gift card" are predictive of lower CRT scores.

Mentions

For the mentions feature, mentions of "youtube" "hulu" "twitter" and "foursquare" are predictive of lower CRT scores while mentions of "reddit," "netflix," and "newyorker" are predictive of higher CRT scores. The most predictive mention for higher CRT scores is "hmrccustomers", which is the customer support account for the UK's tax,

payments and customs authority (HM Revenue & Customs). Public figure mentions such as "tedcruz," "jk rowling," "thatkevinsmith" (Kevin Smith), "neilhimself" (Neil Gaiman), "sethrogen," and "aoc" (Alexandria Ocasio-Cortez) are predictive of high CRT score.

Domains

Domains with the strongest predictive power for high CRT score include "wikipedia," "nature," "wsj," "theonion," and "bbc." Domains for sharing user-created content like "weebly" (personalized website), "gobranded" (surveys), "facebook" (social media), "woobox" (contests and giveaways), "soundcloud" (music), and "pscp" (live streaming) are predictive of lower CRT score.

Followees

The most informative followed accounts for high CRT score include "annakendrick47" (Anna Kendrick), "gretathunberg," "espn," "theonion," "actuallynph" (Neil Patrick Harris), "newscientist," "tedtalks," and "marscuriosity". Accounts with "get rich quick schemes" are more predictive of lower CRT scores. Some examples include "viewsbank" (bio states, "We pay you to be opinionated") and "lifepointspanel" (bio states, "Your opinion. Your Future. Take surveys, get paid."). Interestingly, public figures "katyperry," "barackobama," "shakira," and "jtimberlake" (Justin Timberlake) are also predictive of low CRT score.

Bio

The most informative words in a user's bio paint a picture of the kinds of users with lower versus higher CRT scores. For instance, the bio words "mummy," "artist," "designer," "follow," "assistant," and "music" are predictive of lower CRT score whereas "book", "manager," "university," "student," "football," and "engineer" are predictive of higher CRT score. Additionally, mentioning another user in one's bio (term "user") is also predictive of high CRT score.

Hashtags

The most predictive hashtags for high CRT score include "worldseries," "sherlock," "fallout4" (action role-playing video game), "openaccess," "txlege" (official hashtag for Texas Legislature), "phd," "nvidia," and "engvaus" (hashtag for England vs. Australia for the Cricket World Cup). The most predictive hashtags for low CRT score include "sponsored," "ad," "crypto," "follow," and "splendasavvies" (sponsored content for Splenda).

Informative Features for text

min_df: 10, max_df: 0.99, n_gram: (1, 1), r: 0.362, p value: 0.000

Terms (Low CRT Score)	Coefficients	Terms (High CRT Score)	Coefficients
playlist	-0.2844	original	0.1935
searching	-0.2207	retweet	0.1824
summer	-0.1912	international	0.1775
wowcher	-0.1906	thread	0.1634
thanks	-0.1900	awesome	0.1609
hattie	-0.1701	enough	0.1511
challenge	-0.1695	though	0.1467
gracias	-0.1675	brexit	0.1447
existing	-0.1632	earned	0.1422
family	-0.1619	excellent	0.1382
fantastic	-0.1612	winning	0.1364
rewards	-0.1570	autistic	0.1364
million	-0.1561	walked	0.1357
willoughby	-0.1533	shambles	0.1309
simply	-0.1520	tickets	0.1263

Informative Features for text

min_df: 10, max_df: 0.99, n_gram: (2, 2), r: 0.228, p value: 0.028

Terms (Low CRT Score)	Coefficients	Terms (High CRT Score)	Coefficients
playlist playlist	-0.1367	autistic people	0.1077
simply follow	-0.1166	retweet follow	0.0999
entered giveaway	-0.1154	thanks update	0.0973
another chance	-0.1088	nescaf original	0.0902
hillary clinton	-0.1018	international giveaway	0.0899
listen thanks	-0.0993	really matter	0.0859
fantastic fantastic	-0.0954	amazing please	0.0828
surveys guaranteed	-0.0933	climate change	0.0691
summer excited	-0.0898	daylight savings	0.0658
giveaway chance	-0.0897	colour lipsticks	0.0649
country trying	-0.0878	highly recommend	0.0632
website updated	-0.0864	amazon because	0.0624
please suffer	-0.0851	football league	0.0615
awkward moment	-0.0808	pretty awesome	0.0603
newcastle united	-0.0784	please please	0.0590

Figure 6-6: Informative Tweet/Retweet text with n_gram ranges (1, 1) and (2, 2)

Informative Features for followee_bios

min_df: 10, max_df: 0.99, n_gram: (1, 1), r: 0.258, p value: 0.012

Terms (Low CRT Score)	Coefficients	Terms (High CRT Score)	Coefficients
https	-0.2050	science	0.1159
news	-0.1851	nasa	0.1096
love	-0.1478	scotland	0.1013
surveys	-0.1439	impressum	0.0944
producer	-0.1275	terrier	0.0931
instagram	-0.1214	professor	0.0924
wedding	-0.1162	them	0.0904
blogger	-0.1130	baseball	0.0902
paid	-0.1117	alternative	0.0887
testing	-0.1017	brexit	0.0853
opinion	-0.1009	tweets	0.0839
ufficiale	-0.1006	author	0.0824
opinionated	-0.1005	student	0.0820
more	-0.0972	books	0.0803
twice	-0.0943	they	0.0771

Informative Features for followee_bios

min_df: 10, max_df: 0.99, n_gram: (2, 2), r: 0.396, p value: 0.000

Terms (Low CRT Score)	Coefficients	Terms (High CRT Score)	Coefficients
youtube https	-0.1585	they them	0.1638
creative producer	-0.1513	early access	0.1397
your opinion	-0.1432	border terrier	0.1270
opinionated powered	-0.1419	hier twittert	0.0919
video content	-0.1303	user user	0.0895
https https	-0.1254	university birmingham	0.0847
instagram user	-0.1244	fire emblem	0.0839
travel blogger	-0.1067	twitter account	0.0838
user find	-0.1048	cancer research	0.0801
twitter ufficiale	-0.1042	central park	0.0731
pink floyd	-0.1013	jobs events	0.0706
official twitter	-0.1000	assistant professor	0.0682
riot games	-0.0926	gadget geek	0.0680
user since	-0.0909	using social	0.0680
user instagram	-0.0892	fantasy football	0.0677

Figure 6-7: Informative followee bios with n_gram ranges (1, 1) and (2, 2)

Informative Features for follower_bios

min_df: 10, max_df: 0.99, n_gram: (1, 1), r: 0.193, p value: 0.064

Terms (Low CRT Score)	Coefficients	Terms (High CRT Score)	Coefficients
blogger	-0.1026	science	0.1190
online	-0.0959	brexit	0.0987
products	-0.0930	student	0.0948
family	-0.0926	interested	0.0897
make	-0.0915	orlando	0.0777
surveys	-0.0892	suis	0.0751
ática	-0.0837	scotland	0.0705
sono	-0.0799	geek	0.0682
wedding	-0.0790	politics	0.0664
https	-0.0785	academic	0.0661
will	-0.0769	celtic	0.0658
lifestyle	-0.0759	university	0.0653
youtube	-0.0754	engineer	0.0646
those	-0.0748	intelligent	0.0645
tips	-0.0739	impressum	0.0644

Informative Features for follower_bios

min_df: 10, max_df: 0.99, n_gram: (2, 2), r: 0.158, p value: 0.132

Terms (Low CRT Score)	Coefficients	Terms (High CRT Score)	Coefficients
life about	-0.1125	anti brexit	0.0914
might have	-0.1060	with that	0.0816
social media	-0.1045	than think	0.0714
official twitter	-0.0908	twin boys	0.0696
https youtube	-0.0884	life long	0.0694
will find	-0.0826	that your	0.0688
follow follow	-0.0764	game thrones	0.0667
gift card	-0.0760	data science	0.0641
twitter oficial	-0.0741	business with	0.0636
lifestyle blogger	-0.0723	student user	0.0630
writer poet	-0.0716	freelance artist	0.0617
your service	-0.0713	sense humor	0.0607
about your	-0.0700	like music	0.0595
travel blogger	-0.0684	many others	0.0588
over years	-0.0682	student nurse	0.0587

Figure 6-8: Informative follower bios with n_gram ranges (1, 1) and (2, 2)

Informative Features for mentions

min_df: 10, max_df: 0.99, n_gram: (1, 1), r: 0.269, p value: 0.009

Terms (Low CRT Score)	Coefficients	Terms (High CRT Score)	Coefficients
youtube	-0.0549	hmrccustomers	0.0406
hulu	-0.0368	reddit	0.0345
twitter	-0.0327	netflix	0.0320
foursquare	-0.0318	tedcruz	0.0308
watchmixer	-0.0317	space station	0.0307
viewsbank	-0.0315	thatkevinsmith	0.0261
att	-0.0310	neilhimself	0.0260
morrisons	-0.0301	jk rowling	0.0257
sharethis	-0.0293	eurovision	0.0247
sephora	-0.0293	newyorker	0.0237
po st	-0.0287	curryspcworld	0.0234
cnnbrk	-0.0277	sethrogen	0.0233
qmee	-0.0264	orangeuk	0.0232
potus	-0.0263	qikipedia	0.0232
gma	-0.0258	aoc	0.0231

Figure 6-9: Informative mentions

Informative Features for domains

min_df: 10, max_df: 0.99, n_gram: (1, 1), r: 0.292, p value: 0.004

Terms (Low CRT Score)	Coefficients	Terms (High CRT Score)	Coefficients
weebly	-0.1444	wikipedia	0.1738
gobranded	-0.1355	imgur	0.1450
soundcloud	-0.1255	nature	0.1128
pscp	-0.1204	politics	0.1102
people	-0.1199	gizmodo	0.0996
go	-0.1121	theonion	0.0945
facebook	-0.1033	nationaltrust	0.0929
socialmedialink	-0.0996	newyorker	0.0922
kentonline	-0.0925	wsj	0.0914
rspca	-0.0900	parliament	0.0824
woobox	-0.0892	newscientist	0.0816
playoverwatch	-0.0872	bbc	0.0804
sky	-0.0872	redbubble	0.0801
carrd	-0.0815	steamcommunity	0.0800
crunchyroll	-0.0796	theoatmeal	0.0752

Figure 6-10: Informative domains

Informative Features for followees

min_df: 10, max_df: 0.99, n_gram: (1, 1), r: 0.373, p value: 0.000

Terms (Low CRT Score)	Coefficients	Terms (High CRT Score)	Coefficients
viewsbank	-0.1552	annakendrick47	0.1697
coinbase	-0.1529	gretathunberg	0.1599
lifepointspanel	-0.1515	espn	0.1542
videogamedeals	-0.1487	theonion	0.1432
twitter	-0.1360	fjamie013	0.1348
toluna	-0.1236	actuallynph	0.1133
testingtime	-0.1177	gogcom	0.1118
youtube	-0.1141	nathanfillion	0.1114
hootsuite	-0.1081	wsbmod	0.1003
katyperry	-0.1061	adamschefter	0.0999
barackobama	-0.1056	newscientist	0.0924
tenergyofficial	-0.1003	theellenshow	0.0922
sharethis	-0.0987	seanhannity	0.0916
shakira	-0.0963	tedtalks	0.0906
jtimberlake	-0.0946	marscuriosity	0.0895

Figure 6-11: Informative followees

Informative Features for bio

min_df: 5, max_df: 0.99, n_gram: (1, 1), r: 0.030, p value: 0.772

Terms (Low CRT Score)	Coefficients	Terms (High CRT Score)	Coefficients
mummy	-0.0109	user	0.0137
loves	-0.0102	stuff	0.0084
music	-0.0094	from	0.0081
life	-0.0089	book	0.0077
artist	-0.0080	will	0.0065
social	-0.0080	manager	0.0060
designer	-0.0067	girl	0.0059
follow	-0.0066	university	0.0058
assistant	-0.0058	student	0.0058
your	-0.0054	this	0.0057
things	-0.0052	more	0.0056
professional	-0.0049	once	0.0054
london	-0.0047	account	0.0051
good	-0.0046	football	0.0049
right	-0.0046	engineer	0.0049

Figure 6-12: Informative bio features

Informative Features for hashtags

min_df: 10, max_df: 0.99, n_gram: (1, 1), r: 0.279, p value: 0.007

Terms (Low CRT Score)	Coefficients	Terms (High CRT Score)	Coefficients
sponsored	-0.1197	worldseries	0.0894
rt	-0.1006	sherlock	0.0816
mytwitteranniversary	-0.0993	fallout4	0.0809
ad	-0.0919	openaccess	0.0808
youtube	-0.0917	txlege	0.0736
splendasavvies	-0.0842	yolo	0.0734
crypto	-0.0762	steam	0.0725
soundcloud	-0.0762	gameofthrones	0.0716
in	-0.0753	engvaus	0.0677
orchardattesco	-0.0729	phd	0.0658
follow	-0.0728	dnd	0.0656
life	-0.0726	nvidia	0.0626
amreading	-0.0725	sweeps	0.0610
amazon	-0.0718	snl	0.0597
support	-0.0699	webdesign	0.0590

Figure 6-13: Informative hashtags

Chapter 7

Conclusion

7.1 Discussion

The primary goal of this thesis is to predict cognitive reflection within reasonable accuracy using social media behavior. By creating a machine learning model to learn patterns across Twitter features from Mosleh et. al's study to predict CRT score as a proxy for cognitive reflection, I generated the strongest Pearson's r correlation coefficient of 0.29 between predicted and actual CRT scores using followee information.

Looking at the individual, umbrella, and combined features prediction together, the results are consistent: the strongest predictors for CRT score contain information around accounts followed (e.g. followees, followee bios). Above all other Twitter features observed in this project, the accounts a user chooses to follow is most predictive of their inclination to engage in analytical thinking. Additionally, the model is robust when adding new data. Comparing the individual results from the Complete (n = 926) and Full datasets (n = 1,808), there is no correlation between the number of added users and the difference in r median value.

Comparing the results to previous work described in Chapter 2, my model generates a median r value larger than the maximum r value from previous studies that used Twitter digital fingerprints, with the exception of Farnadi et al.'s study, which uses demographic features (age and gender) in addition to Tweet text for prediction. My model predicts a novel individual trait — cognitive reflection — above the accu-

racy of previous studies that have used Twitter features to predict Big 5 personality.

The informative feature results corroborate the findings from Mosleh's study in that words, domains, accounts, and hashtags related "get rich quick" schemes are predictive of lower CRT scores (e.g. "giveaway," "sweepstakes," "paid"). In general, weightier and more scholarly terms are predictive of higher CRT score (e.g. "engineer," "student," "professor," "science"). Additionally, sharing platforms for user-created content are more predictive of lower CRT score.

What we learn from these results is that following a user on Twitter is powerful. By simply hitting the follow button, an individual chooses to allow the followed user's thoughts, content, opinions, and sharings (in the form of Retweets) into one's own timeline: the feed of posts an individual sees when they open the Twitter application. Who you follow inherently indicates what kinds of content and people are important to you — important enough that every post made by that account will appear on your timeline. If you use Twitter frequently enough, you will naturally dedicate attention and energy to their posts as you scroll through your feed. Not only can who you follow say a lot about who you are, the results from this study provide evidence that who you follow is predictive of your ability to engage in reflective thinking.

7.2 Ethics

My model is a tool to predict CRT score from Twitter fingerprints, and there are important ethical implications around knowing one's CRT score. Knowing an individual's CRT score implies knowing their propensity to engage in analytical thinking. In some sense, an individual's CRT score can be a window into their vulnerability and their ease to been manipulated, particularly for low CRT scorers. Knowing this kind of information can create huge ethical problems and can even exacerbate the echo chamber phenomenon across social networks. In general, models that use data and powerful algorithms to learn about individual traits without consent can be very dangerous if placed in the hands of the wrong people. Particularly for this model, knowledge of individuals' CRT scores can be used to classify individuals based off a

descriptive cognitive trait, which could be a form of discrimination if used with poor intention. While the model I built used individual data with consent, the model could be deployed as an active, dynamic tool that updates its algorithms and predictions by learning from more input data from individuals — whether or not they have provided consent.

To mitigate potential negative implications, policies should be in place to avoid any group to obtain full knowledge of CRT score and power over this tool. For instance, policies can require aggregation of data rather than individual data. Advertisement and political discourse should not target individuals with low CRT scores versus high CRT scores, but rather, CRT scores should be treated in aggregation like other individual identification traits such as ethnicity or gender.

7.3 Future Work

There are several directions to expand the current work. First, the dataset used to create the model is limited in observations. By acquiring more data through additional studies or other labs, the model can train on a larger corpus of data, which can lead to improved model accuracy and more generalized results. Additionally, by acquiring similar data from other labs and applying the same model from this project, we can test the robustness of the model on new datasets.

The best-performing model uses followee features to predict CRT score. A second direction for future work is to estimate CRT scores from papers that provide users' Twitter data. For example, one recent paper measures the spread of misinformation from public figures users choose to follow (Mosleh, 2021). One interesting application of this model would be to estimate CRT scores given the followee data from this study and see how the scores correlate with other user characteristics estimated from the user's timeline (e.g. ideology, toxicity, moral outrage).

The model is created to predict CRT score from Twitter features; however, a third extension is to apply the model to predict a different target using Twitter features. The model is designed such that the user can indicate the target variable at

the beginning of the pipeline. As long as the target variable is a numerical value, the model structure and methodologies can be easily applied for non-CRT prediction (e.g. predict a user's Twitter behavior to predict reward anticipation via their monetary incentive delay task score).

As a whole, my thesis helps to generalize what we know about the relationship between cognitive reflection and social media behavior to larger populations and across other social media platforms. It adds to previous work by predicting a new individual trait — cognitive reflection — from online behavior. The implications of this project will reverberate to future challenges in social media creation, moderation, and policy.

Appendix A

Pipeline Schematic

Figure A-1: High-level pipeline schematic, targeted for scientific and non-scientific audiences

Appendix B

Table of Target Features

Feature Name	***Description***	***Example***
domains	Domains the user has Tweeted and Retweeted, separated by a space and lowercased (keep repeats)	*"facebook youtube amazon amazon bbc facebook"*
mentions	Mentions the user has Tweeted and Retweeted, separated by a space and lowercased	*"potus amazonuk harry_styles potus potus"*
hashtags	Hashtags the user has Tweeted and Retweeted, separated by a space and lowercased	*"piano classical classicalmusic planets symphony symphony piano"*
text	Text from Tweets and Retweets the user has shared, separated by a space and pre-processed	*"i am looking forward to the weekend ! melodrama by lorde is one of the best albums of all time ."*
followees	Followees' screen names, separated by a space	*"POTUS ChaseSupport Oprah BBCNews"*
followers	Followers' screen names, separated by a space	*"rihanna Drake billyjoel"*
bio	Twitter bio after pre-processing	*"aspiring musician , composer , and experimenter . i write songs and play the piano ."*
follower_bios	Followers' Twitter bios, separated by a space and pre-processed	*"i produce new music every sunday . dm me any time . here to promote local artists from dallas . reach out to me with any questions"*

followee_bios	Followees' Twitter bios, separated by a space and pre-processed	*"discover your next favorite artist or find out about upcoming releases . president of the united states , husband to @user , proud dad and pop ."*
bio_char	Number of characters in bio	*22.0*
bot_score	Probability of account being a bot (API)	*0.88*
creation_time_of_day	Hour when the Twitter account was created in Coordinated Universal Time (UTC), out of 24	*17.0 (equivalent to 5 P.M. UTC)*
domains_count	Number of domains Tweeted or Retweeted (count repeats)	*12.0*
domains_count_unique	Number of unique domains Tweeted or Retweeted	*8.0*
favorites_avg	Average number of times a user's Tweets has been liked by other Twitter users	*177.2*
favorites_given_count	Total number of Tweets this user has liked in the account's lifetime	*288.0*
followees_count	Number of followees (Twitter accounts that follow the user)	*42.0*
followee_followees_avg	Average number of followees of a user's followees	*42.4*
followee_followers_avg	Average number of followers of a user's followees	*101.6*
followee_likes_avg	Average number of likes given to Tweets in the account's lifetime across user's followees	*25.6*
followee_media_avg	Average number of media shared in the account's lifetime across user's followees	*229.0*
followee_tweets_avg	Average number of Tweets shared across user's followees	*123.2*
followee_verified_score	Score out of 1.0 for percentage of verified followees	*1.0 = All followees are verified; 0.0 = No followees are verified*
followers_count	Number of Twitter followers	*248.0*
follower_followees_avg	Average number of followees of a user's followers	*32.4*
follower_followers_avg	Average number of followers of a user's followers	*87.6*
follower_likes_avg	Average number of likes given to Tweets in the account's lifetime across user's followers	*33.6*

follower_media_avg	Average number of media shared in the account's lifetime across user's followers	*41.0*
follower_tweets_avg	Average number of Tweets shared across user's followers	*22.2*
follower_verified_score	Score out of 1.0 for percentage of verified followers	*1.0 = All followers are verified; 0.0 = No followers are verified*
has_location	User has a user-defined location for this account's profile	*1.0 = True; 0.0 = False*
has_profile_image	User has a profile image	*1.0 = True; 0.0 = False (Twitter's default image)*
hashtags_avg_length	Average number of characters in user's hashtags (count repeats)	*8.4*
hashtags_count	Number of hashtags Tweeted or Retweeted (count repeats)	*144.0*
inhibit	Fraction of Tweets that have at least one word from the Linguistic Inquiry and Word Count dictionary for inhibit words	*0.54*
insight	Fraction of Tweets that have at least one word from the Linguistic Inquiry and Word Count dictionary for insight words	*0.17*
is_protected	User has chosen to protect their Tweets	*1.0 = True; 0.0 = False*
listed_count	Number of public lists that this user is a member of	*10.0*
mentions_count	Number of mentions Tweeted or Retweeted (count repeats)	*290.0*
mentions_count_unique	Number of unique mentions Tweeted or Retweeted	*170.0*
moral	Fraction of Tweets that have at least one word from the Linguistic Inquiry and Word Count dictionary for moral words	*0.23*
neg_emotion	Fraction of Tweets that have at least one word from the Linguistic Inquiry and Word Count dictionary for negative words	*0.77*
outlet_score	The average quality of content on the websites a user shares (Pennycook and Rand, 2019)	*5.0*

political	Fraction of Tweets that have at least one word from a dictionary of political words, suggested by reference (Preoţiuc-Pietro et al., 2017)	*0.12*
pos_emotion	Fraction of Tweets that have at least one word from the Linguistic Inquiry and Word Count dictionary for positive words	*0.23*
relig	Fraction of Tweets that have at least one word from the Linguistic Inquiry and Word Count dictionary for religious words	*0.44*
screen_name_char	Number of characters in user's screen name	*11.0*
screen_name_digits	Number of digits in user's screen name	*2.0*
statuses_count	Number of Tweets (including retweets) issued by the user	*1027.0*
url_exist	User provides a URL in association with their profile	*1.0 = True; 0.0 = False*

Table B.1: Target features with descriptions and examples

Appendix C

Libraries, Packages, and Modules

Name	Category	Link to Documentation
numpy	General	https://numpy.org/doc/stable/
pandas	General	https://pandas.pydata.org/docs/
math	General	https://docs.python.org/3/library/math.html
pickle	General	https://docs.python.org/3/library/pickle.html
matplotlib	General	https://matplotlib.org/
plotly	General	https://plotly.com/
statistics	General	https://docs.python.org/3/library/statistics.html
random	General	https://docs.python.org/3/library/random.html
pyarrow	General	https://arrow.apache.org/docs/python/parquet.html
os	General	https://docs.python.org/3/library/os.html

train_test_split	General	https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
Ridge	Prediction	https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html
Lasso	Prediction	https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html
RandomForestRegressor	Prediction	https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
pearsonr	Prediction	https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html
PolynomialFeatures	Prediction	https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html
GridSearchCV	Prediction	https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
Pipeline	Prediction	https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html

SelectFromModel	Selection	https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html
ElasticNetCV	Selection	https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNetCV.html
StandardScaler	Transformation	https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
TruncatedSVD	Tranformation	https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html
TfidfVectorizer	Transformation	https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Table C.1: Modules used in code infrastructure

Bibliography

1. Azucar, Danny, Davide Marengo, and Michele Settanni. "Predicting the Big 5 personality traits from digital footprints on social media: A meta-analysis." *Personality and individual differences* 124 (2018): 150-159.

2. Bruno Kessler Foundation COVID-19 and Fake News in the Social Media (10 March 2020). Available online: https://www.fbk.eu/en/press-releases/covid-19-and-fake-news-in-the-social-media/ (Accessed on April 14 2022).

3. Depoux, Anneliese, et al. "The pandemic of social media panic travels faster than the COVID-19 outbreak." *Journal of travel medicine* 27.3 (2020): taaa031.

4. Dhingra, Bhuwan, et al. "Tweet2vec: Character-based distributed representations for social media." *arXiv preprint arXiv:1605.03481* (2016).

5. Du, Siying, and Steve Gregory. "The Echo Chamber Effect in Twitter: does community polarization increase?." *International workshop on complex networks and their applications*. Springer, Cham, 2016.

6. Farnadi, Golnoosh, et al. "Computational personality recognition in social media." *User modeling and user-adapted interaction* 26.2 (2016): 109-142.

7. Frederick, Shane. "Cognitive reflection and decision making." *Journal of Economic perspectives* 19.4 (2005): 25-42.

8. Golbeck, Jennifer, Cristina Robles, and Karen Turner. "Predicting personality with social media." *CHI'11 extended abstracts on human factors in computing systems*. 2011. 253-262.

9. Goldberg, Lewis R. "An alternative" description of personality": the big-five factor structure." *Journal of personality and social psychology* 59.6 (1990): 1216.

10. Gottlieb, Michael, and Sean Dyer. "Information and disinformation: social media in the COVID-19 crisis." *Academic emergency medicine* (2020).

11. Kouzy, Ramez, et al. "Coronavirus goes viral: quantifying the COVID-19 misinformation epidemic on Twitter." *Cureus* 12.3 (2020).

12. Liu, Leqi, et al. "Analyzing personality through social media profile picture choice." *Tenth international AAAI conference on web and social media*. 2016.

13. Mosleh, Mohsen, et al. "Cognitive reflection correlates with behavior on Twitter." *Nature communications* 12.1 (2021): 1-10.

14. Mosleh, Mohsen, and David Rand. "Falsehood in, falsehood out: A tool for measuring exposure to elite misinformation on Twitter." (2021).

15. Pennycook, Gordon, and David G. Rand. "Fighting misinformation on social media using crowdsourced judgments of news source quality." *Proceedings of the National Academy of Sciences* 116.7 (2019): 2521-2526.

16. Pennycook, Gordon, and David G. Rand. "Lazy, not biased: Susceptibility to partisan fake news is better explained by lack of reasoning than by motivated reasoning." *Cognition* 188 (2019): 39-50.

17. Pennycook, Gordon, et al. "Understanding and reducing the spread of misinformation online." *ACR North American Advances* (2020).

18. Preoţiuc-Pietro, Daniel, et al. "Beyond binary labels: political ideology prediction of twitter users." *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017.

19. Puri, Neha, et al. "Social media and vaccine hesitancy: new updates for the era of COVID-19 and globalized infectious diseases." *Human vaccines & immunotherapeutics* 16.11 (2020): 2586-2593.

20. Qiu, Lin, et al. "You are what you tweet: Personality expression and perception on Twitter." *Journal of research in personality* 46.6 (2012): 710-718.

21. Roberts, Brent W., et al. "The power of personality: The comparative validity of personality traits, socioeconomic status, and cognitive ability for predicting important life outcomes." *Perspectives on Psychological science* 2.4 (2007): 313-345.

22. Stuart, Jaimee, et al. "Online social connection as a buffer of health anxiety and isolation during COVID-19." *Cyberpsychology, Behavior, and Social Networking* 24.8 (2021): 521-525.

23. Sumner, Chris, et al. "Predicting dark triad personality traits from twitter usage and a linguistic analysis of tweets." 2012 *11th international conference on machine learning and applications*. Vol. 2. IEEE, 2012.

24. Venegas-Vera, A. Verner, Gates B. Colbert, and Edgar V. Lerma. "Positive and negative impact of social media in the COVID-19 era." *Reviews in cardiovascular medicine* 21.4 (2020).

25. Youyou, Wu, Michal Kosinski, and David Stillwell. "Computer-based personality judgments are more accurate than those made by humans." *Proceedings of the National Academy of Sciences* 112.4 (2015): 1036-1040.