Vorwissenschaftliche Arbeit

Die Manipulation der Frontpage von Reddit mithilfe von Bots

David Mikan, 8A

betreut von Theresa Hemedinger

Akademisches Gymnasium Wien

Beethovenplatz 1, 1010 Wien

Februar 2021

Inhaltsverzeichnis

Abstract

This paper sheds light on Reddit's front page and the inner workings of its community-based systems, especially in relation to attempted manipulation by bots. Firstly, Reddit's dynamics as social network, as well as its structure and popularity systems, are examined, determining that in fact only a small group of users plays a deciding role in a post's chances of becoming popular. Then the platform's rules and measures against malicious bots are evaluated, followed by a code demonstration of the ease with which one such malicious bot can be created and instructed to automatically cast votes.

Finally, a dataset of 95k datapoints, collected on the subreddit r/memes over a span of 42 hours, is introduced. This is used to analyse the development of posts on the subreddit. The insight emerges that the closed-source algorithms used by reddit to calculate which posts to show on its front page can be put into boundaries. The relatively small sample collected on r/memes enabled predictions about a posts future popularity and showed that a post must gain a certain amount of upvotes in designated timespans to be able to get to the front page. If a post's score is artificially inflated shortly after the post was created, it will show up in Reddit as rising to popularity, which increases its visibility and thus its chances of becoming popular. Contrastingly, small amounts of downvotes can bring down the ratio of upvotes, consequently keeping a post from getting to the front page which indicates the potential possibility of censorship caused by bots.

1. Einleitung

Die konstant zunehmende Macht sozialer Medien als politischer Meinungsbildner und Inhaltsfilter für die Massen macht sie zu weit mehr als bloß einer weiteren im Internet angebotenen Dienstleistung oder Kommunikationsplattform. Mit mehr als 52 Millionen täglich aktiven Nutzerinnen ist Reddit die aktuell neunzehnt-größte Website auf der Welt nach Alexas *The top 500 sites on the web*[1]. Das soziale Medium ist stark am Wachsen und dient vielen seiner Benutzerinnen als Haupt-Nachrichtenquelle. (Kastrenakes, 2020)

1.1. Motivation und Forschungsleitende Fragen

Ich bin seit 2 Jahren als aktives Mitglied der Community auf Reddit aktiv. In dieser Zeit bemerkte ich oft *Bots* als allseits präsente Entitäten, die mit Benutzerinnen interagieren und die Plattform beleben[2]. Inmitten dieser beständigen Aktivität fallen von Zeit zu Zeit Benutzerinnen auch bösartige Bots auf und werden oft auf der Seite diskutiert. Sie nutzen Reddits Systeme gezielt aus, um Inhalte und Meinungen zu bewerben, und andere zu zensieren. Diese Systeme interagieren auf besondere Weisen miteinander, die sich von anderen sozialen Netzwerken unterscheiden, jedoch durch ihre ungewöhnliche Beschaffung die Manipulation der Sichtbarkeit von Inhalten erlauben. Aus Interesse wollte ich diese Systeme und Algorithmen näher kennenlernen und ihr Verhalten quantifizieren, um mögliche Vorgehensweisen von bösartigen Bots aufzuzeigen.

Die forschungsleitenden Fragen, an deren Beantwortung sich die Arbeit orientiert, lauteten wie folgt:

Q1: Wie funktioniert das Popularitätssystem auf Reddit und durch welche Eigenschaften zeichnet es sich aus?

Q2: Wie können Reddits Systeme durch Bots zur Manipulation des Informationsflusses ausgenutzt werden? Wie sieht so ein Bot aus?

Q3: (Wie) kann man relevante Parameter im Zusammenhang mit dem Popularitätsalgorithmus quantifizieren, um für die Manipulation wichtige Stellen zu identifizieren?

1.2. Zu verwendeten Definitionen und Mitteln

Bedingt durch die technische Natur sind in dieser Arbeit viele Anglizismen und Fachausdrücke anzutreffen. Auf den Seiten 32-34 befindet sich deshalb ein Glossar; Wörter, die hier nachzuschlagen sind, sind bei erster Verwendung *kursiv* geschrieben.

[1] www.alexa.com/topsites/, zugegriffen am 6.2.2021

Das Literaturverzeichnis ist in APA-Zitation angegeben und befolgt die Richtlinien der 7. Auflage. Die Ausnahme sind Links, die nicht auf (direkt oder indirekt wiedergegebene) Inhalte, sondern Einzelnachweise zu verwendeten Daten sind, eigenständig erarbeitetes Material beinhalten oder erweiternde Informationen bieten, die in der Arbeit nicht thematisiert werden. Diese Links sind in Fußnoten, meist mit beigefügten Erklärungen, vermerkt.

Eben angesprochenes eigenständig erarbeitetes Material befindet sich in einem Github-Repository unter https://github.com/itsMik4n/Reddit-Frontpage-Data-Collection. Es enthält zur Datenerhebung und -auswertung eingesetzte Skripts, die erhobenen Daten, außerdem auch generierte Plots und Skripts mit Beispielen und Versuchen zur Herleitung verwendeter statistischer Mittel. Hochgeladen wurde das Repository am 22.2.2021, zuletzt bearbeitet am 23.2.2021. Besucht man den Link, findet man eine Ordnerstruktur vor, darunter befindet sich eine Übersicht mit Links zu den jeweiligen Dateien/Ordnern.

In dieser Arbeit wird aus Gründen der erleichterten Lesbarkeit das weibliche Substantiv verwendet, es bezieht sich sowohl auf die männliche als auch die weibliche Form.

2. Reddits Dynamiken als soziales Netzwerk und Nachrichtenquelle

2.1. Der Aufbau von Reddit

Reddit ist ein community-basiertes soziales Netzwerk. Es ermöglicht seinen Benutzerinnen, in diversen Foren, genannt Subreddits, Beiträge und Kommentare in Form von Texten, Links, Bildern und GIFs, Videos u.ä. zu verfassen, und von anderen Benutzerinnen erstellte Inhalte zu konsumieren. Reddit unterscheidet sich von anderen sozialen Netzwerken durch zwei seiner Struktur zugrundeliegende Ideen. Zum einen ist dies die vorherrschende Pseudonymität, Accounts werden prävalent unter Pseudonymen, mit keiner direkten Verbindung zur Person dahinter, erstellt (Van der Nagel & Frith, 2015). Zum anderen konzentriert sich die Reddit Frontpage (siehe 2.2.1) auf Themen, nicht auf einzelne Benutzerinnen. Statt dass Inhalte nach einzelnen Benutzerinnen gefiltert werden (will heißen, man folgt einzelnen Personen, deren Inhalte einem dann angezeigt werden), werden sie vielmehr durch die Einteilung in Subreddits thematisch gefiltert.

2.1.1. Subreddits

Es gibt mehr als 10 000 aktive Subreddits, jeder widmet sich einem bestimmten Thema oder Themenbereich. Die Bezeichnung von Subreddits formt sich aus r/, gefolgt vom einzigartigen Namen des Subreddits[4]. Die enorme Diversität und Granularität an aktiven Subreddits werden am besten durch Beispiele dargestellt:

Auf dem Subreddit r/ProgrammerHumour werden Witze unter Programmierern geteilt, während r/MovieDetails kleine, oft übersehene Details in Filmen findet. Auf r/Avoid5 findet man keinen Post mit dem Buchstaben E und r/HoldMyCatnip sammelt Videos von Katzen, die sich wie berauscht benehmen.

Doch nicht alle Subreddits sind rein amüsanter Natur: r/RelationshipAdvice hilft bei Beziehungskrach, in r/RBI tummeln sich (teils Hobby-)Privatdetektive die Anderen Rat geben und Investigationen in realen Situationen durchführen. r/News und r/WorldNews verlinken auf Zeitungsartikel zu aktuellem politischen Geschehen und sind in der Debatte um die mögliche Manipulation der Sichtbarkeit besonders wichtig.

2.1.2. Das Karma-System

Das Karma-System ist das zentrale Mittel zur Bewertung von Posts und Kommentaren. Userinnen können ihre Stimme nutzen, um den *Score* von Posts um 1 zu erhöhen oder zu senken; man spricht von Up- und Downvotes.

[4] Diese Konvention bildete sich daraus, dass Links auf Subreddits immer mit www.reddit.com/**r/** beginnen. Die Benutzernamen werden durch das Präfix u/ ausgezeichnet.

Durch auf Posts und Kommentare erhaltene Up-/Downvotes bekommen die Posterinnen Karmapunkte. Diese sind die einzige quantitative Vergütung für das Erstellen von Inhalten und besitzen keinen realweltlichen Wert. Die Motivation zur Partizipation im Subreddit-Verkehr liegt demnach in anderen Gründen; A. Richterich unterscheidet in ihrem Artikel (2014) zwischen intrinsischer und extrinsischer Motivation:

> "[...] on the one hand, there are users who emphasize the importance of topical variety and content quality [...] On the other hand, users reveal an interest in maximising their Karma-points."

Dieses gesamte System hat auf Userinnen zwei bemerkenswerte Auswirkungen:

Die Userin hat das Gefühl, als Teil der Community mit ihren Stimmen direkte Kontrolle auf die Sichtbarkeit von Posts zu nehmen. Daraus entsteht ein Gefühl der Obligation, qualitativ hochwertige Inhalte mit Upvotes zu versehen, und umgekehrt jene, die gegen Regeln verstoßen oder unter geringem Aufwand erstellt wurden, zu downvoten. (Richterich, 2014)

2.1.3. Popularität

Votes von Benutzerinnen bilden den Score eines Posts, ein Wert zur Quantifizierung der gesamten Votes[5]. Dieser Score spielt eine vitale Rolle in der auf Reddit verwendeten Form der Filterung von Post (reddit.com: api documentation, o. D.). *Welche* Posts *wo* und *wann* den Userinnen angezeigt werden, wird von einem algorithmisch gesteuerten demokratieähnlichen System entschieden. Der eingesetzte *Algorithmus* bewertet die Relevanz von Posts basierend auf einer Menge an Variablen. Diese können quantifizierbare Parameter wie der Score, aber auch einem Betrachter unbekannte Variablen sein, da der Algorithmus[6] closed-source ist; die zur Berechnung verwendeten Funktionen sind nicht öffentlich.

Die Popularität eines Posts besitzt keine handfeste Definition; sie steht im Zuge dieser Arbeit synonymisch für die Sichtbarkeit und Relevanz eines Posts. Das heißt, ein Post ist umso populärer, je mehr Userinnen ihn sehen, ihn up-/downvoten, oder ihn kommentieren[7].

2.2. Der Lebensweg eines Posts

Folgend wird in der Arbeit die Entwicklung eines Posts in drei Phasen eingeteilt, die jeder populäre Post durchläuft. Diese Phasen können als Abstraktion gesehen werden, welche die Dynamiken der folgend erklärten Sortiermethoden verständlich machen soll.

[5] Auf den Score wird in 4.2 näher eingegangen.
[6] Wird in dieser Arbeit „der Algorithmus" erwähnt, beziehe ich mich fortan auf ebendieses Filtersystem.
[7] Zur Sichtbarkeit findet sich Weiteres in 2.2.2 und 4.4.1. Die Popularität wird in 4.2 der Einfachheit und Messbarkeit zugunsten in drei Stufen unterteilt.

2.2.1. Die Frontpage

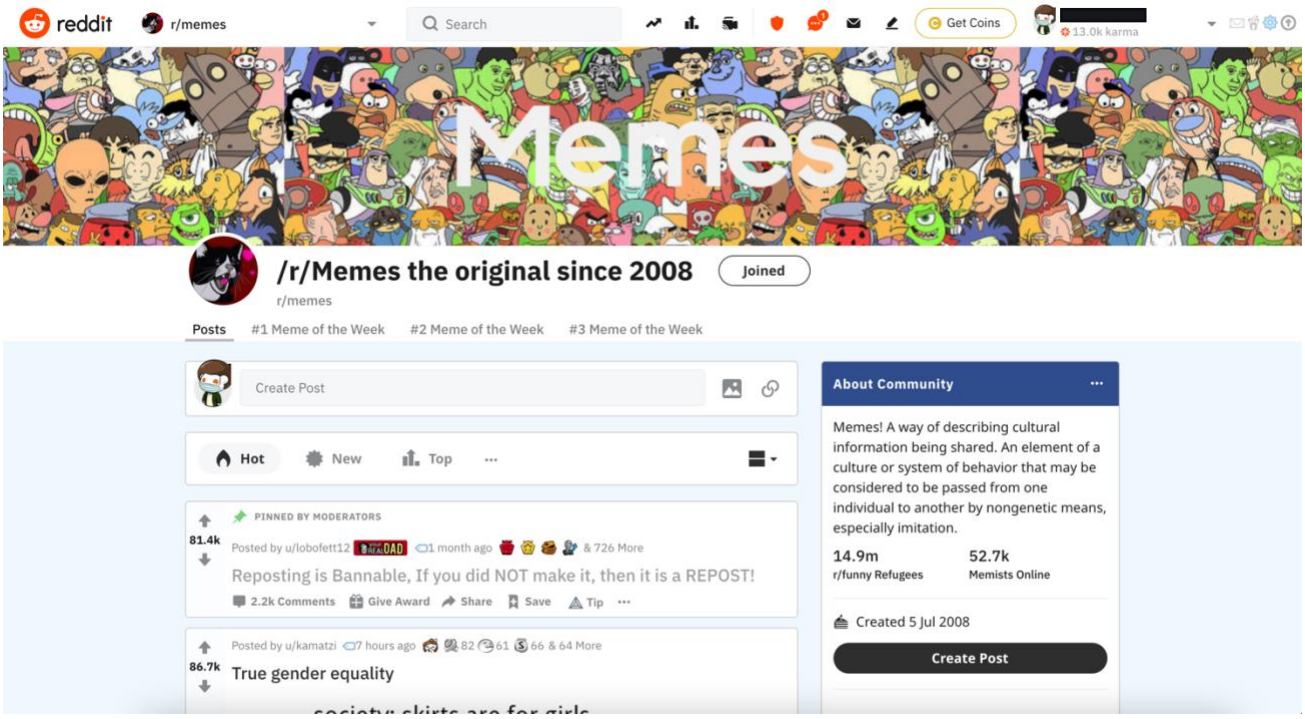

Fig 1 Das Layout von Reddits Website am Beispiel der Frontpage von r/memes, Stand 24.2.2021

Die Frontpage ist ein mehrschichtiger Überbegriff für eine Seite, auf der Posts aufgelistet sind. In der grundsätzlich akzeptierten Verwendungsweise steht die Frontpage für den Haupt-*Feed* einer Userin, wenn er nach Hot sortiert ist. In Kongruenz kann sie aber auch für den Feed einer unangemeldeten Person stehen, oder jenen eines einzelnen Subreddits.

2.2.2. Von New, über Rising, zu Hot

Möchte man sich nicht nur Posts in Hot (=auf der Frontpage) ansehen, kann man nach anderen Kriterien sortieren. Wählt man die Sortiermethode New, werden zuletzt gepostete Inhalte zuerst angezeigt. Ein soeben erstellter Post ist ausschließlich hier auffindbar – „er ist in New“. (reddit.com: api documentation, o. D.)

Ein erstellter Post ist also zunächst nur in New. Erhält ein gegebener Post in New genug Aufmerksamkeit, kommen erst die anderen Sortiermethoden des Feeds ins Spiel. Es steht Userinnen eine Vielzahl an Optionen zur Wahl, für diese Arbeit von Belang sind das schon besprochene New, Rising und Hot. Unter Rising finden sich Posts, die sozusagen „am Aufstieg zur Popularität“ sind und in Hot sind jene, die populär sind. Sie verzeichnen die meisten Upvotes, Kommentare und sonstigen Interaktionen. Ein Post, der es aus New herausgeschafft hat und nun in Rising aufscheint, muss hier wiederum erfolgreich genug sein, um nach Hot zu kommen. (reddit.com: api documentation, o. D.)

Die Algorithmen hinter Rising und Hot sind, wie in 2.1.3 erläutert, nicht öffentlich zugänglich. Wichtig anzumerken ist, dass technisch gesehen jeder Post eines Subreddits auch auf seiner Frontpage zu finden ist, es werden durch lediglich alle Posts absteigend nach unterschiedlichen Parametern wie Alter, Score und Kommentaranzahl sortiert. Es muss also arbiträr eine Grenze gezogen werden[8], in dieser Arbeit wurde diese für den Subreddit r/memes bei 100 festgesetzt, also ist ein Post „in Hot" oder „auf der Frontpage"[9], wenn er mit dieser Sortiermethode in den ersten 100 Posts auftaucht. Die weiteren Dynamiken dieses Systems werden in 3.1 und 3.2 beleuchtet.

2.3. Nachrichten und Politik

In einer von Reddits wichtigsten Funktionen agiert es als Aggregator für Nachrichtenartikel. Auf mehreren zu diesem Zweck eingerichteten Subreddits werden durch Posts Artikel verlinkt, die dann durch das Popularitäts-System sortiert werden, so entstehen Frontpages mit öffentlich bezogenen Nachrichten verschiedenster Quellen aus dem Internet. Nicht nur zur Akkumulation, sondern auch zur Bewertung und Diskussion wird die Plattform genutzt; unter jedem Post entstehen Unterhaltungen und Debatten um dasjenige Thema (Leavitt & Robinson, 2017, S. 1–3).

Auf diese Art werden Nachrichten aus den Federn professioneller Journalisten von der Community nach öffentlichem Interesse und empfundener Qualität aggregiert, unabhängig der veröffentlichenden Zeitung oder anderer Faktoren. Die Sortieralgorithmen wirken dabei als Inhaltsfilter und zeigen so die populärsten Artikel auf den Frontpages an. Auf diese Art ist Reddit die primäre Nachrichtenquelle von etwa 5% der amerikanischen Staatsbürgerinnen (Barthel et al., 2016).

Neben der personalisierten Frontpage findet sich in der mobilen Applikation von Reddit auch ein „News"-Tab. Im Verhalten ähnlich zur Frontpage, werden hier Frontpages einer festgesetzten Gruppe von nachrichtenbezogenen Subreddits angezeigt. Zwar minimal personalisierbar, ist dieser Inhalt größtenteils für jede gleich, Unterschiede nach Region gibt es auch nicht.

2.3.1. Die Echokammer für alle

Eine Echokammer[10] entsteht, wenn sich Individuen in virtuellen und reellen Welten nur mit Medien und Inhalten umgeben, die ihren schon gebildeten Meinungen und Überzeugungen entsprechen (Bruns, 2017). Dies führt zu einer Segregation zu einzelnen Filterblasen, in

[8] Grenzen wie diese können sehr wohl auch innerhalb von Reddits Algorithmus festgelegt sein. Erkenntnisse in 4.4.1 zeigen, dass z.B. die Sortiermethode Rising eine „harte" Altersgrenze für Posts hat, nach der keine Posts mehr in an der Spitze von Rising sind.
[9] Diese Terminologie wird fortan verwendet.
[10] Der Begriff Echokammer ist sehr lose definiert und wird in öffentlichen Debatten eher als konzeptuelle Idee verwendet, deshalb versucht sich diese Arbeit an einer diese Idee umfassenden Definition.

welchen die eigenen Ideologien nicht angezweifelt und jegliche anderen zensiert und als irrelevant empfunden werden. In Reddit entsteht eine solche Echokammer durch die in diesem Kapitel beschriebenen Dynamiken und Systeme. Was viel geupvotet wird, wird populär und sichtbar, das Andere bleibt der Masse unsichtbar und wird somit zensiert. Dadurch entsteht eine Echokammer, bei der sich die Benutzerinnen nicht völlig bewusst sind, dass ihnen eine von der populären Meinung gefilterte Landschaft an Inhalten gezeigt wird.

3. Die (manipulierte) Verbreitung von Inhalten auf Reddit

3.1. „lost in new“ – wer macht einen Post populär?

Unter dem Getümmel an Posts findet sich immer wieder einer, dessen Titel etwas wie „I hope this doesn’t get lost in new“ (Ich hoffe, dies geht nicht in New unter) aussagt. Die Post-Erstellerin bezieht sich mit dieser Aussage auf den allgemein angenommenen Umstand, dass ein Post, unabhängig von seiner Qualität oder Relevanz, in einer kurzen Zeit nach dem Posten eine gewisse Mindestmenge an Upvotes generieren muss, damit er überhaupt eine Chance hat, eine größere Zahl an Menschen zu erreichen. Geschieht dies nicht, verschwindet er im ständigen Zulauf neuer Posts - geht in New unter.[12] Das Gefühl vieler Userinnen, dass so auch qualitätsvolle Inhalte verlorengehen, wurde bewiesen durch den Nachweis einer Unterversorgung mit Userinnen, die Upvotes geben (Gilbert, 2013).

Falls der Post jedoch genug Upvotes bekommt, taucht er sodann in Rising auf, wo sich deutlich weniger Posts, aber auch mehr Userinnen befinden, deshalb werden qualitative Inhalte hier gefiltert und erhalten mehr Upvotes, was sie nach Hot bringt (populär macht) (reddit.com: api documentation, o. D.).

Als allgemein akzeptierte These gilt also, dass neben purem Zufall auch gutes Timing und eine Dose Glück notwendig sind, damit ein Post populär wird und Tausende bis Millionen an Menschen erreicht. All diese Elemente sind in der Phase nach dem Posten, in der sich der Post ausschließlich in New befindet, am wichtigsten und entscheidendsten. Wenn man demnach das richtige Timing kannte, und das Element des Glücks eliminieren könnte, würden sich die Chancen, einen populären Post zu erstellen, drastisch erhöhen.

3.1.1. Die Rolle der Sortiermethode New

Posts, die sich nur in New befinden, erscheinen ausschließlich jenen Benutzerinnen, die ihre Frontpage nach New sortieren. Nach New zu sortieren ist jedoch größtenteils unattraktiv für Benutzerinnen, da Posts vollkommen ungefiltert erscheinen, Spam und subjektiv Uninteressantes dominieren die New-Frontpage. Hinzu kommt, dass die Reddit Frontpage in der Grundeinstellung immer nach Hot sortiert ist und manuell zu New umgestellt werden müsste. Somit ist der Großteil der Benutzerinnen nie in New unterwegs, eine relativ kleine Gruppe benutzt diese Anzeigemethode. Da jedoch Votes in New den weiteren Lebenslauf eines Posts bestimmen, hat diese kleine Gruppe eine ungleich große, dominierende Macht. Eine Handvoll Upvotes kann dafür sorgen, dass ein Post es nach Rising schafft, und ihm somit Chancen geben, populär zu werden.

[12] In 4.4.1 wird dieses Phänomen weiter analysiert und für r/memes quantifiziert.

Eine noch kleinere Menge an Downvotes hingegen veranlasst, dass er in New stirbt. Ein Blog Post von Young (2013) zeigt mittels des damals noch öffentlich zugänglichen Quellcodes von Reddit, wie ein Angreifer mit verhältnismäßig kleinen „Bomben“ an Downvotes einen Post daran hindern kann, jemals in Rising, und somit auch Hot, zu erscheinen. Der Quellcode ist seitdem nicht mehr öffentlich zugänglich, Reddit ist stark gewachsen und hat sich verändert, dennoch wird in 4.4.3 gezeigt, dass dieselbe Behauptung immer noch wahr ist.

3.1.2. Relevanz für Versuche der Manipulation

In New kann eine relativ geringe Zahl an manipulierten Votes dafür sorgen, dass Posts sich aus dem Strom neuer Inhalte erheben und größeren Menschenmengen sichtbar werden, oder genau das verhindern, weshalb es der effektivste Ansatz zur Manipulation ist und fortlaufend in der Arbeit im Fokus steht. Dass eine künstliche Vote-Inflation kurz nach dem Posten garantiert, dass ein Post in Rising landet, wird in 4.4.1 argumentiert.

3.2. Bots auf Reddit und deren Einsetzung zur Manipulation von Posts

Bots sind Computerprogramme, die, ohne menschliche Eingaben, Aufgaben automatisiert erledigen. Die in dieser Arbeit referenzierten Bots nennen sich Social Bots, sie interagieren mit sozialen Netzwerken unter Accounts genauso, wie ein Mensch dies tun würde, ihre Handlungen sind jedoch vorprogrammiert. Bots, wie ebenjene, die sich an der Manipulation von Inhalten auf Reddit versuchen, sind sogenannte bösartige Bots. 24% der im Internet verzeichneten Aktivität stammt von bösartigen Bots (Imperva Research Labs, 2020).

Reddit bietet eine *API* spezifisch für Bots, die von der Community erstellt werden. Diese API macht es möglich, mit nur wenigen Zeilen[13] an Code Bots aufzustellen, die auf Reddit Posts abrufen, mit ihnen interagieren, Kommentare hinterlassen, etc. Gekoppelt an spezifisch formulierte Bedingungen können so automatisierte Scripts erstellt werden, die, beispielsweise, neue Posts eines gewissen Subreddits archivieren, oder nach Kommentaren zu einem bestimmten Thema suchen und auf diese mit vorprogrammierten Nachrichten antworten.

3.2.1. Bottiquette

Die Reddiquette[14] definiert Verhaltensregeln für die Userinnen auf der Plattform. Unter ihnen finden sich Regeln, wie:

> „Please don’t follow those who are rabble rousing against another redditor without first investigating both sides of the issue that's being presented”

[13] Wie in 3.2.4 demonstriert
[14] https://reddit.zendesk.com/hc/en-us/articles/205926439-Reddiquette, zugegriffen am 16.2.2021

Es besteht auch ein Gegenstück für Bots, die Bottiquette[15]. Sie enthält gleich zwei Mal die Aufforderung, Bots das Vergeben von Up- und Downvotes nicht zu erlauben:

> „Please don't allow your bot to vote"
>
> „Please don't create bots for the purposes of voting, votes must be cast by humans"

Dennoch ist das Voten für Bots durch die offizielle API möglich, es besteht auch keine bekannte Strafe oder Beschränkung für das Nutzen dieser, die Bottiquette ist vielmehr ein Leitfaden für gutwillige Akteure.

3.2.2. Account-Erstellung

Jede Userin/jeder Bot interagiert mit Reddit über einen Account. Solch ein Account besteht aus einem Benutzernamen, einem Passwort, und ist an eine E-Mail geknüpft. Im Gegensatz anderen sozialen Netzwerken muss diese E-Mail jedoch nicht über einen zugesendeten Link bestätigt werden, was die Verwendung von Wegwerf-E-Mail-Adressen zur automatisierten Account-Erstellung erleichtert. Dass Accounts von einem Menschen erstellt werden, garantiert ein *Captcha*, somit ist eine automatisierte Erstellung von Accounts stark erschwert[16]. Sich für Bots Accounts en masse zu beschaffen, ist dennoch nicht schwer: auf Internetseiten wie www.accsmarket.com (zugegriffen am 17.2.2021) wird behauptet, Tausende von Reddit Accounts zu niedrigen Preisen wie $0.24 pro Stück zu verkaufen.

3.2.3. Maßnahmen gegen Bots

Innerhalb Reddits sind natürlich Mechanismen zur Einschränkung bösartiger Bots eingerichtet. Reddit behauptet[15], es gäbe Algorithmen, welche versuchen, Bot-Accounts aufzuspüren. Da ein entdeckter Bot einfach auf einen anderen Account wechseln könnte, werden diese Accounts nicht gebannt, sondern „shadow-gebannt". Dies bedeutet, dass der Benutzerin (dem Bot) nichts angezeigt wird, aber deren Up- und Downvotes sowie Kommentare und Posts nicht mehr registriert werden. Diese sogenannten Shadow Bans sind interessant, da sie definitiv das Gefühl vermitteln, es wird etwas gegen Bots unternommen, sie sind aber kaum effektiv:

Ein Bot könnte nun einfach den Score eines Posts abrufen, dann ein Upvote vergeben, und schauen, ob sich der Score um 1 erhöht hat. Falls nicht, wäre der Account shadow-gebannt. Um dies zu verhindern existiert sogenanntes Vote Fuzzing; die API überträgt für den Score von Posts keine präzisen Werte, sondern wandelt sie zufällig leicht ab.

---

[15] https://www.reddit.com/wiki/bottiquette, zugegriffen am 16.2.2021

[16] Dennoch ist es realistisch möglich, sogenannte „captcha solver" Dienste gibt es zu sehr billigen Preisen https://www.perimeterx.com/resources/blog/2020/captchas-hard-for-humans-easy-for-bots/, es wäre alternativ auch eine technische Leichtigkeit, den restlichen Prozess zu automatisieren und die Captchas per Hand zu lösen; Sie werden in IT-Sicherheits-Kreisen zunehmend als veraltet gesehen.

Dass dieses Fuzzing noch ineffektiver ist als die Shadow Bans liegt daran, dass der Bot einfach von der API seinen jeweiligen Account abrufen kann. Liegt ein Shadow Ban vor, antwortet die API mit einem http-Statuscode 404, sodass das Vote Fuzzing irrelevant ist und höchstens Verwirrung bei legitimen Bots verursachen würde.

Wie erfolgreich erwähnte Algorithmen zur Erkennung regelbrechender Bots wirklich sind, oder wie sie vorgehen, ist völlig unbekannt. In einem Konferenzbeitrag von Carman et al. (2018, S.185) wurden jedoch erfolgreich politische wie apolitische Posts mit „Vote-Injektionen“ von 10 Upvotes versehen. Die gemessenen Auswirkungen sind beträchtlich und signalisieren, dass Vote-Manipulation schon mit einer kleinen Menge an Bots möglich und effektiv ist, was die Nützlichkeit und Komplexität jener Anti-Manipulations-Algorithmen infrage stellt.

3.2.4. Beispiel eines bösartigen Bots[17]

```
import praw

# Achtung: Dieses Skript dient nur edukativen Zwecken. Vote Manipulation
durch Bots
# ist auf Reddit verboten und wird, falls nachgewiesen, mit (Shadow-)Bans
bestraft.

bot = praw.Reddit(
    # Die Anmeldedaten sind mit jenen des jeweiligen Accounts
auszufüllen.
    # Wie man sich API-Zugang verschafft, wird in
    #
https://praw.readthedocs.io/en/latest/getting_started/quick_start.html
demonstriert
    # (zugegriffen am 22.2.2021)
    client_id = '###',
    client_secret = '###',
    password = '###',
    username = '###',
    user_agent = '###'
)
target = bot.subreddit('memes')
disliked_keywords = [
    # Eine Liste mit Schlüsselwörtern, die nicht erwünscht sind,
    # diese wird der Bot mit Downvotes versehen
    'dog',
    'puppy',
    'pitbull',
    'good boy'
]
liked_keywords = [
    # Enthält der Post eines diese Schlüsselwörter, gibt der Bot ihm
    # ein Upvote
```

[17] Die verwendeten Technologien werden in 4.1 genauer beschrieben.

```
    'cat',
    'kitten',
    'meow'
]

for post in target.stream.submissions():
    # Der Stream agiert als Endlosschleife: jeder neu erstellte Post
    # läuft durch durch eine Iteration und sein Titel wird sodann mit
    # den Schlüsselwörtern abgeglichen, und die passende Handlung
durchgeführt
    name = post.name.lower()
    if any(k in name for k in disliked_keywords):
        post.downvote()
    elif any(k in name for k in liked_keywords):
        post.upvote()
```

Code Block 1 https://github.com/itsMik4n/Reddit-Frontpage-Data-Collection/blob/main/tests/vote_bot_demonstration.py (hochgeladen am 22.2.2021)

Code Block 2 demonstriert einen primitiven bösartigen Bot[18], geschrieben in der Programmiersprache Python unter Verwendung der *Programmbibliothek* praw.

Zuerst meldet sich der Bot über einen gegebenen Account an, danach wird der Ziel-Subreddit in der Variable `target` definiert. `liked_keywords` und `disliked_keywords` sind zwei Listen, welche jeweils erwünschte und unerwünschte Schlüsselwörter definieren. Folgend kann mittels einer simplen Schleife der Bot angewiesen werden, auf neue Posts im Subreddit zu warten und jenen mit erwünschten Wörtern im Titel ein Upvote zu erteilen, und wiederum jenen mit unerwünschten Inhalten ein Downvote zu geben.

Der im Beispiel verwendete Bot würde, falls ausgeführt, auf r/memes jegliche neuen Posts mit Titeln, die von Katzen sprechen, upvoten und alle Memes über Hunde downvoten, und somit Katzen-Memes mehr Sichtbarkeit verschaffen (et vice versa). Es ist offensichtlich, wie dies, statt für Memes über Katzen und Hunde, zu politischen oder kommerziellen Zwecken genutzt werden könnte, und, statt mit einem Account, mit einem Netzwerk aus Bot-Accounts mit einer viel größeren Macht versehen werden könnte. Das Besorgniserregende ist jedoch die Leichtigkeit, mit der ein solcher Bot programmiert werden kann. Das Beispiel in Code Block 2 zeigt, dass ein paar Dutzend Zeilen[19] an Code ausreichen, um gezielt Inhalte aus New zu filtern und zu manipulieren.

3.3. Anwendungsbereiche der Manipulation

[18] Als bösartiger Bot versteht sich in diesem Kontext ein solcher, der gegen die Regeln Reddits verstößt.
[19] Das Code-Beispiel dient demonstrativen Zwecken und ist stark simplifiziert – das Kernstück ist jedoch die mühelose Interaktion mit der API von Reddit. Die verwendete Bibliothek erlaubt sogar eine sichere Verbindung über das Tor-Netzwerk – eine Maßnahme, die es einem potenziellen Angreifer leicht macht, seine Identität zu maskieren.

Die Anwendungsbereiche solch bösartiger Bots sind endlos. Zwei zeichnen sich jedoch dadurch aus, dass sie im großen Stil, teils auf nationaler Ebene, angewendet werden. Es handelt sich um die Verbreitung von Fake News, hauptsächlich zur Radikalisierung, aber auch einfach zur Disruption gedacht, und die Verbreitung von Werbung für Produkte oder Marken (Hurtado et al., 2019) (Rosenberg et al., 2019).

3.3.1. Fake News

Das Problem der Fake News ist mit der Ära der sozialen Medien und Peer-News Aggregation auf neue Höhen geschossen. Der Cambridge Analytica-Skandal war eng verbunden mit gezielter Verbreitung von Falschinformationen an Menschen, die zuvor von Algorithmen als empfänglich dafür eingestuft wurden (Guess et al., 2018).

4. Statistische Analyse 95 000 erhobener Datenpunkte

Im Zuge dieser Arbeit wurde eigenständig ein *Datensatz* aus grob 95000 Datenpunkten gesammelt. Es wurden Posts auf ihrem Weg zur (Nicht-)Popularität begleitet und dabei wurden regelmäßig relevante Attribute abgespeichert, sodass ein Datensatz entstand, der von insgesamt 6330 Posts die Entwicklung der ersten anderthalb Stunden katalogiert. Folgend werden in der Arbeit diese Daten auf interessante Eigenschaften untersucht, die potenziell mehr Aufschluss zu versteckten Faktoren, welche die Popularität eines Posts beeinflussen, liefern. Somit werden sowohl zeitliche Stellen identifiziert, an denen effektive Manipulationen eines Posts stattfinden könnten, als auch beeinflussbare Werte des Posts aufgezeigt, die anfälliger oder weniger anfällig für Manipulationen der Sichtbarkeit sind. Zunächst werden jedoch wichtige Grundlagen der statistischen Auswertung betrachtet, die in der Analyse angewendet wurden, sowie verwendete Methoden/Scripts erläutert.

Alle für diesen praktischen Teil der Arbeit verwendeten Skripts, Dateien, und Visualisierungen sind in einem *Github-Repository*, abrufbar unter https://github.com/itsMik4n/Reddit-Frontpage-Data-Collection, am 23.2.2021 hochgeladen worden.

4.1. Verwendete Technologien

Folgend werden der Prozess der Datenerhebung und Speicherung von Posts sowie die Durchführung der statistischen Auswertung der Daten beschrieben. Der Code wird in allgemeiner Sprache beschrieben; Auszüge aus verwendeten Skripts (in dieser Arbeit als *Code Blocks* beschriftet) sind lediglich der Vollständigkeit halber enthalten und nicht für das Verständnis der Arbeit notwendig.

Der Datensatz wurde in der Programmiersprache Python mithilfe der Bibliothek *praw* erhoben, die als *Wrapper* zur API von Reddit fungiert. Ein simples Beispiel eines Bots wurde in 3.2.4 demonstriert.

4.1.1. Datenbeschaffung

Das Ziel war, einem Subreddit eine vollständige Stichprobe an erstellten Posts zu entnehmen und aus dieser Rückschlüsse auf die generalisierte Entwicklung der Posts, getrennt nach erreichter Popularität, zu ziehen. Der gewählte Subreddit war r/memes; die Gründe für die Wahl von r/memes lagen einerseits in der hohen Frequenz, mit der neue Posts von Userinnen erstellt wurden, andererseits in der unpolitischen Natur des Subreddits (wie sogar in seinen Verhaltens- und Postregeln klar gemacht wird: „Rule 11 - NO MEMES ABOUT POLITICS“[22] [23]).

[22] Die Regeln eines jeden Subreddits – falls es sie gibt – befinden sich auf seiner Frontpage.
[23] https://reddit.com/r/memes, Stand 23.2.2021

Über den Zeitraum von 42 Stunden wurden im Intervall von 10 Minuten jeweils die neuesten Posts der vergangenen 10 Minuten gesammelt und grundlegende Attribute einmalig abgerufen; dazu zählen Titel des Posts, Erstellungszeitpunkt, ein permanenter Link zu diesem Post und die Information, ob es sich um einen Textpost handelt; dieser Schritt wird fortan *Collection* genannt.

Unmittelbar nach der Collection einer Post-Gruppe wurde ein *Update* dieser Gruppe durchgeführt. Updates begannen damit, dass die ersten 100 Posts aus jeweils Rising und Hot von r/memes abgerufen und gespeichert wurden (diese zwei Sets nennen sich im Code Update-Filter, mit ihnen kann kontrolliert werden, ob ein gewisser Post sich „in Rising“ oder „in Hot“ befindet). Anschließend wurde jeder Post der Gruppe abgerufen und seine variablen Attribute wurden abgespeichert[24]. Schlussendlich wurde der Post mit den Update-Filtern abgeglichen. Code Block 2 enthält die Methode *update* der Post-Gruppen-Klasse *PostGroup*.

```
    def update(self, reddit):
        if self.rem_saves <= 0:
            raise Exception("No further saves remaining")
        log(f"updating posts, updates left: {self.rem_saves}", self.id)
        timestamp = datetime.timestamp(datetime.now(timezone.utc))
        try:
            sub = reddit.subreddit(self.from_sub)
        except Exception:
            log('couldn\'t be saved, ' + self.jsonify(), self.id)
            return
        filters = [ # gets the x first posts from hot and rising as sets
            {_.id for _ in sub.rising(limit=MAX_FILTER)},
            {_.id for _ in sub.hot(limit=MAX_FILTER)}
        ]
        self.update_filters[timestamp] = [list(_) for _ in filters]
        for p_id, post in self.posts.items():
            try:
                submission = reddit.submission(p_id)
                update = {}
                for attr in ATTR_VARIABLE:
                    update[attr] = getattr(submission, attr)
                # check if submission is rising/hot (ergo interesting)
                update['rising'] = p_id in filters[0]
                update['hot'] = p_id in filters[1]
                if update['rising']: log(f"{p_id} is in Rising! Upvotes:
{update['score']}, {update['upvote_ratio']}")
                if update['hot']:
                    log(f"{p_id} is in Hot!!! Upvotes: {update['score']},
Ratio: {update['upvote_ratio']}")
                post['updates'][timestamp] = update
            except Exception as e: # if an error occurs and a post can't
be updated, drop the whole post
                # (most likely the post was deleted)
                log(f"failed to update {p_id}, error: {e}", self.id)
```

[24] Alle abgespeicherten Attribute werden in 4.3 erklärt und sind als Teil des Codes auch in https://github.com/itsMik4n/Reddit-Frontpage-Data-Collection/blob/main/scraper/config.py dokumentiert.

```
                self.posts.pop()
        self.rem_saves -= 1
        log(f"finished updating posts", self.id)
```

Code Block 2 Die Methode update aus der Klasse PostGroup, https://github.com/itsMik4n/Reddit-Frontpage-Data-Collection/blob/main/scraper/pg.py, Z.31-64

Über die folgenden 140 Minuten werden 14 weitere Updates erstellt, sodass letzten Endes von jedem Post 15 Zustände über 150 Minuten hinweg gespeichert wurden.

4.1.2. Datenauswertung

Für die statistische Auswertung wurde ein Skript geschrieben, welches den Datensatz in eine Variable ausliest und die Architektur zur Visualisierung eines Graphen bietet.

Vor der eigentlichen statistischen Auswertung der folgend gezeigten Plots und Datenverhältnisse wurden zahlreiche Versuche, Messungen, und Visualisierungen durchgeführt, um ein tieferes und vollständigeres Verständnis der Daten zu erhalten. Das erwähnte Skript für die Visualisierungen und partielle Einblicke in diesen Prozess finden sich im Repository unter https://github.com/itsMik4n/Reddit-Frontpage-Data-Collection/tree/main/plotter/plots.

4.2. Der Datensatz

	ID	Time	Score	Ratio	Comments	Stickied	Rising	Hot	State
0	0	0.2	2	1.0	0	False	False	False	rising
1	0	9.7	23	0.96	0	False	False	False	rising
2	0	19.2	31	0.97	0	False	True	False	rising
...	...	...	...	...	...	...	...	...	...
94949	2592	159.61	23	0.87	9	False	False	False	dead

Table 1 Auszug aus https://github.com/itsMik4n/Reddit-Frontpage-Data-Collection/blob/main/dataset/datapoints_complete.csv

Der verwendete Datensatz wurde aus den Rohdaten im *JSON-Format* generiert und in einer Datei im *CSV-Format* abgespeichert.[25]

Table 1 gibt einen Ausschnitt des Datensatzes wieder. Jede Reihe steht für einen einzelnen Datenpunkt, jeder dieser Datenpunkte repräsentiert einen Post an einem gewissen Zeitpunkt nach seiner Erstellung. Jedem Post können genau 15 Datenpunkte zugeordnet werden, die alle im Abstand von etwa 10 Minuten voneinander liegen[26]. Insgesamt verzeichnet der verwendete Datensatz 94949 Datenpunkte gehörig zu 6330 Posts.

[25] Alle Rohdaten befinden sich in https://github.com/itsMik4n/Reddit-Frontpage-Data-Collection/tree/main/raw_data, auf den Datensatz kann über https://github.com/itsMik4n/Reddit-Frontpage-Data-Collection/blob/main/dataset/datapoints_complete.csv zugegriffen werden.

[26] Durch kleine Verzögerungen im Ablauf des Codes, als auch den in der Funktion *mainloop* in der *Scheduler*-Klasse eingebauten Intervall von 60 Sekunden (Minimierung der Prozessorauslastung), wurden Updates nicht im perfekten 10-Minuten-Takt erstellt, es besteht eine durchschnittliche Abweichung von 28 Sekunden. Diese ist zu vernachlässigen, da die abgerufenen Attribute sich in dem Zeitraum nicht bemerkenswert ändern.

Ein Datenpunkt besitzt jeden der folgenden Werte[27]:

ID	Zum selben Post gehörige Datenpunkte haben dieselbe ID (Identifikationsnummer).
Time	Die Zeit in Minuten, die seit der Erstellung des zum Datenpunkt gehörigen Posts vergangen ist.
Score	Der Score eines Posts ist die Differenz der gesamt erhaltenen Upvotes zur Anzahl der erhaltenen Downvotes.
Ratio	Die Upvote-Ratio stellt den Anteil der Upvotes an den insgesamt erhaltenen Stimmen dar.
Comments	Die Anzahl aller unter dem Post erstellten, auch gelöschten, Kommentare.
Stickied	*True*, wenn der Post zu dem Zeitpunkt an der Frontpage des Subreddits gepinnt ist[28].
Rising	*True*, wenn der zugehörige Post zum Zeitpunkt des Updates in Rising zu finden ist.
Hot	True, wenn der Post im Moment in Hot (auf der Frontpage) zu finden ist.
State	*dead*[29], falls der zugehörige Post im gesamten Messzeitraum weder in Rising noch in Hot war; *rising*, falls er in Rising, jedoch nie in Hot war; *hot*, falls er an mindestens einem Updatezeitpunkt in Hot, und somit auf der Frontpage, war

4.3. Grundlegendes zur Auswertung

Wie in 3.2 beschrieben, basiert der Algorithmus auf nicht öffentlichen Funktionen. Die Parameter, nach denen sortiert wird, und deren Gewichtung in der Entscheidung des

[27] per https://praw.readthedocs.io/en/latest/code_overview/models/submission.html, zugegriffen am 23.2.2021
[28] Wurde nur erhoben, um einen Bias zu vermeiden (von Moderatoren angepinnte Posts werden unverhältnismäßig hoch geupvotet sein und sind automatisch „in Hot“) Keiner der gesammelten Posts war im Messzeitraum *stickied*.
[29] Die gewählte Bezeichnung *dead* für einen Post, der weder in Rising, noch in Hot war, bezieht sich, mangels einer allgemeingültigen Bezeichnung, auf die verbreitete Aussage „the post died in new“ („Der Post ist in New gestorben“) und hat keine weitere Relevanz

Algorithmus, können nicht mit Sicherheit vollständig dokumentiert und bewiesen werden. Durch die Beobachtung einer Stichprobe an Posts in einem Subreddit in ihrer Entwicklung vom Erstellungszeitpunkt an kann jedoch ein repräsentativer Datensatz erstellt werden, der sodann nach einzelnen variablen Werten ausgewertet werden kann. Die Visualisierung von Abgrenzungen verschiedener Parameter nach Zeit, Quantität, Verteilungsdichte u.ä., die vom Algorithmus gezogen werden, erlaubt ein tieferes Verständnis davon, was einen Post auf die Frontpage bringt und wie dies zur Manipulation des Algorithmus eingesetzt werden könnte.

4.3.1. Kausalzusammenhang – Ursache und Wirkung

Fest steht mit Sicherheit, dass beobachtete starke Korrelationen der erhobenen Parameter zur Popularität der Posts auch eine Kausalität indizieren (=Parameter ist Ursache für Popularitätseinstufung), da bekannt ist, dass genau diese Werte vom Algorithmus in Betracht gezogen werden, und er aus ihnen die Popularität errechnet. Was in dieser Arbeit analysiert wird, sind sowohl die einzelnen Zusammenhänge und Rollen der Parameter, also welche unter ihnen wirkungsvoller sind als andere, als auch deren Entwicklung über Zeit.

4.3.2. Unsichtbare Grenzen und wo man sie findet

Neben (negativ und positiv) korrelierten Werten kristallisieren sich bei der folgenden Analyse auch, teils sehr klar gezogene, Grenzen heraus, die entstehen und das Leben eines Posts in einen gewissen Rahmen fassen. Beispielsweise erkennt man, dass sich nach 104 Minuten kein einziger Datenpunkt mehr in Rising befindet. Dadurch, dass bis zu diesem Punkt die Werte regelmäßig verteilt liegen und es keine Ausreißer gibt, liegt der Schluss nahe, dass hier eine arbiträre Grenze liegt, und es lässt sich sagen: „Wenn ein Post älter als 104 Minuten ist, wird man ihn nicht mehr in Rising finden“[30] Vorsicht ist bei der Wort- und Sinnwahl der Rückschlüsse geboten; Ursache – Wirkung sind nicht immer deutlich, beispielsweise wäre der Schluss, der Algorithmus schließe ältere Posts aus Rising aus, falsch, da ein anderer Parameter mit dieser Beobachtung verbunden sein könnte.

[30] Hier spielen der spezifische Subreddit und die gewählte Definition, wann ein Post in Hot oder Rising ist, eine entscheidende Rolle. Die für diese Arbeit gewählten Definitionen sind in 3.2 nachzulesen.

4.4. Visualisierungen und Auswertung der Daten

4.4.1. Score-Entwicklung über Zeit

Fig 2 visualisiert die Entwicklung der Scores über den gesamten Zeitraum, in welchem die Posts beobachtet wurden. Die drei *Streudiagramme* sind nach Post-Popularität (dem State) getrennt, die Farbe indiziert, wo sich der zugehörige Post bei der Aufnahme des jeweiligen Datenpunktes befindet[31].

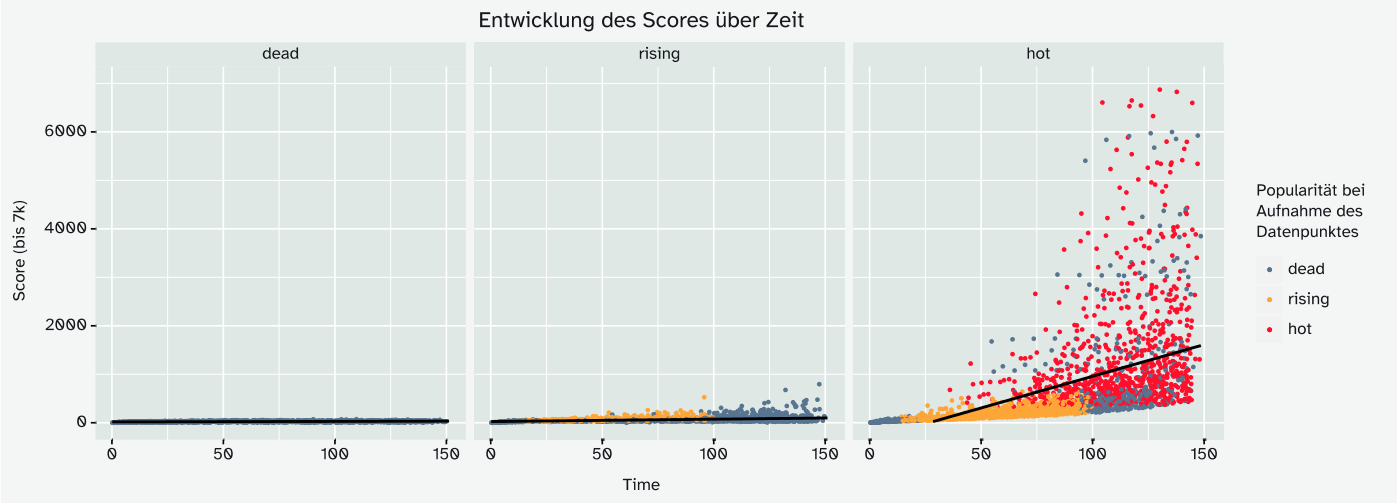

Fig 2 Die Entwicklung des Scores über Zeit, getrennt nach Post-Popularität; schwarze Linien entsprechen der linearen Regression der Datenpunkte

Posts, die in New verstorben sind, liegen, wie erwartet, konstant nahe bei 0, und entwickeln sich über die Zeit kaum nach oben. Ein Umstand, auf den hingewiesen werden sollte, ist die starke Disparität der Unterschiede zwischen den States dead-rising und rising-hot. Posts mit dem State rising zeigen zwar einen leichten Trend nach oben, jedoch ist der Unterschied von rising zu dead im Vergleich zu jenem zu hot (den populären Posts) verschwindend gering. Werden die Durchschnitte der letzten Datenpunkte jedes Posts betrachtet (also die Scores, die die Posts jeweils im letzten beobachteten Update aufwiesen), wird dieser noch klarer: der Durchschnitt des letzten verzeichneten Scores 30 für Posts mit dem State dead, 90 für jene aus rising, und 2776 für populäre Posts[32]. Dies zeigt, wie wichtig es für die Sichtbarkeit eines Posts tatsächlich ist, auf der Frontpage zu landen. Nur in Hot erreichen Inhalte die breiten Benutzerinnenmassen.

Ein interessantes Phänomen zeigt sich auch bei der Betrachtung der jeweiligen Popularität der Posts bei Aufnahme der Datenpunkte. Dieser Parameter führt nicht an, welche Popularitätsstufe der dem Datenpunkt zugehörige Post im gesamten Beobachtungszeitraum

[31] Ermittelt aus den Parametern *Rising* und *Hot;* falls ein Punkt sowohl in Rising als auch in Hot ist, wird er zu Hot dazugezählt.
[32] Der Code zum Berechnen dieser Durchschnitte findet sich unter https://github.com/itsMik4n/Reddit-Frontpage-Data-Collection/blob/main/tests/calc_proofs.py.

erreicht, sondern für jeden Datenpunkt wird überprüft, ob der zugehörige Post sich bei der Durchführung des jeweiligen Updates in Rising oder Hot befindet (verdeutlicht durch die Färbung der Datenpunkte, siehe Legende in Fig 1 und 2). Hier zeigt sich, dass ab einer bestimmten Zeit nach der Posterstellung abrupt jegliche Posts aus Rising verschwinden. Bei näherer Betrachtung (in Fig 2) zeigt sich, dass dieser Zeitpunkt bei 104 Minuten liegt. Aus mehr als 20 000 Datenpunkten, die bei ihrer Aufnahme in Rising waren, gibt es keinen einzigen Ausreißer. Dies legt den Schluss auf eine der in 4.3.2 besprochenen Grenzen des Sortier-Algorithmus sehr nahe. Ein Post, der älter als 104 Minuten ist, wird nicht mehr in Rising sein. Dies bedeutet einerseits, dass ein Post, welcher noch nicht in Rising war, nach 104 Minuten mit Sicherheit einem Tod in New geweiht ist, andererseits könnte jedoch auch der Schluss gezogen werden, dass ein Post, der schon den Schritt, in Rising aufzutauchen,

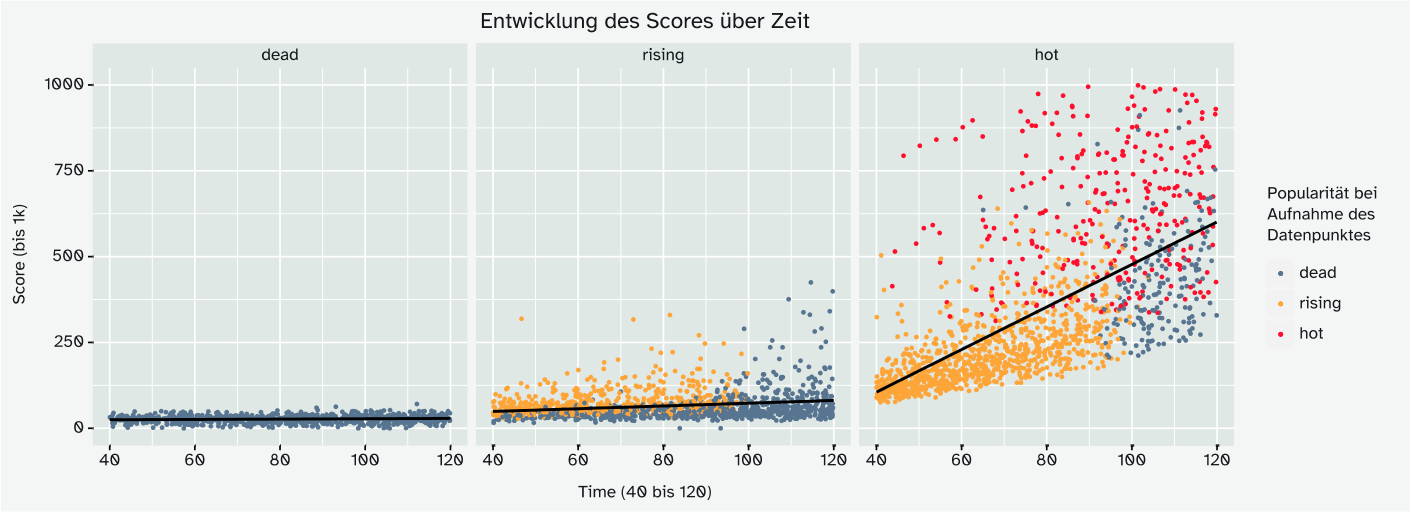

Fig 3 Die Entwicklung des Scores über Zeit, getrennt nach Post-Popularität. Gezeigt wird der Ausschnitt 40-120 Minuten nach der Post-Erstellung, die y-Achse geht nur bis 1000, um einen besseren Eindruck der Punkteverteilung zu geben

geschafft hat, nur eine begrenzte Zeit hat, um nach Hot zu kommen. Denn fällt er von Rising, wird er niemandem mehr angezeigt, und kann so keine Upvotes mehr erhalten.

4.4.2. Die konvexe Hülle als mächtiges Mittel der Entwicklungsvorhersage

Von einer Menge an Punkten lässt sich die konvexe Hülle zeichnen, indem die kleinste Teilmenge an Punkten gefunden wird, die zu einem Polygon verbunden alle anderen Punkte beinhaltet. Wenn man sich jeden Punkt am Graphen als Nagel im Brett vorstellt, ist die konvexe Hülle jene Gestalt, die ein Gummiband annehmen würde, wenn man es um alle Punkte spannte. Einzelne Ausreißer können großen Einfluss auf die konvexe Hülle einer Punktemenge haben, aber sie ist dennoch ein gutes Werkzeug, um die Gestalt von Ansammlungen an Punkten zu visualisieren und so die Zugehörigkeit eines Punktes zu einer Gruppe abschätzen zu können. Irgendeine Quelle angeben

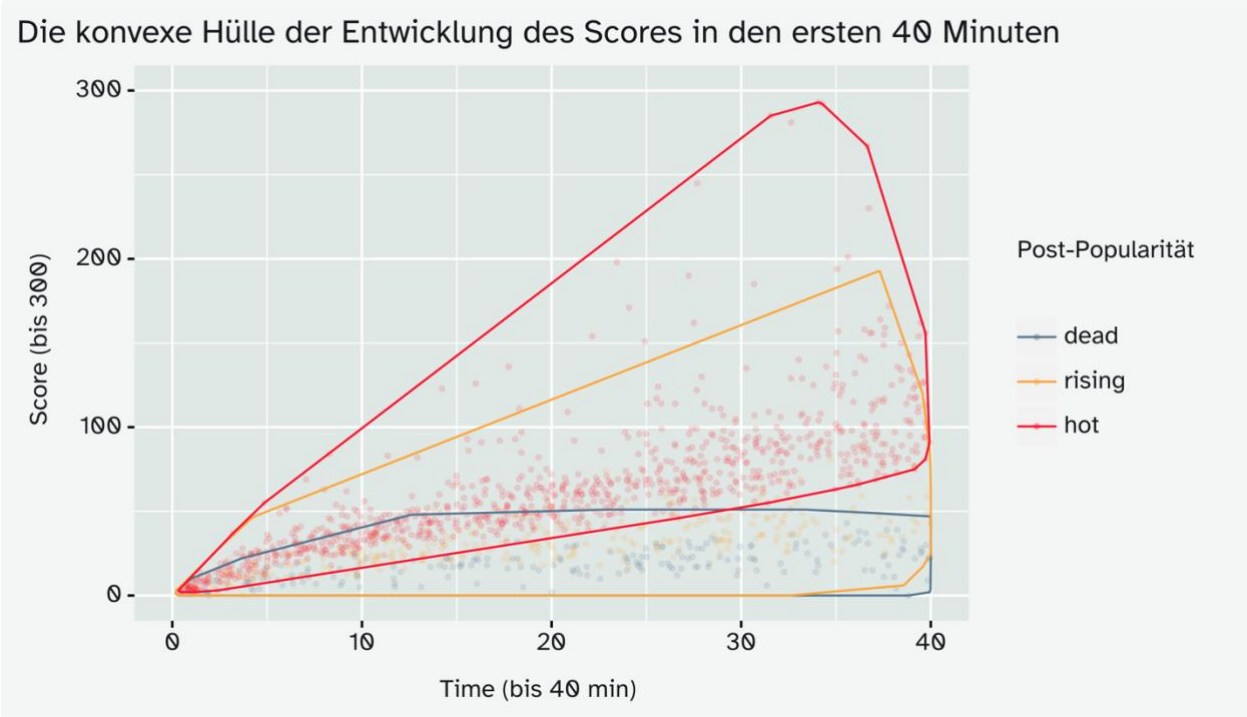

Fig 4 Die konvexen Hüllen getrennt nach Post-Popularität

In Fig 4 Die konvexen Hüllen getrennt nach Post-Popularität wurden die erhobenen Datenpunkte der ersten 40 Minuten nach der Popularität, die ihr zugehöriger Post insgesamt erreichte, getrennt, um mögliche Abspaltungen der verschiedenen Gruppen im unmittelbaren Zeitbereich nach dem Erstellungszeitpunkt zu analysieren. Tatsächlich zeigt sich schon eine klare Unterscheidung der konvexen Hülle der populären Posts (die rot umzeichnete Fläche in Fig 4 Die konvexen Hüllen getrennt nach Post-Popularität, fortan *populäre Hülle* genannt).

Grundsätzlich kann gesagt werden, dass ein Punkt, der außerhalb der konvexen Hülle einer der Gruppen liegt, mit ziemlicher Sicherheit aus dieser Gruppe ausgeschlossen werden kann. Das heißt, wenn ein neuer Punkt eines beobachteten Posts nicht in der populären Hülle liegt, dann wird dieser Post in seinem Leben vermutlich niemals in Hot sein (Überprüfen kann man dies, indem man aus einer Stichprobe von 80% der Punkte die konvexe Hülle errechnet, und

dann für den Rest der Punkte kontrolliert, ob sie in dieser Hülle liegen. Beim Versuch lagen von 614 Punkten 611 in der konvexen Hülle, nur 3 nicht[33]). Man kann also schon sehr früh in der Entwicklung eines Posts mit hoher Sicherheit[34] Prognosen ob der zukünftigen Entwicklung treffen. In der Manipulation durch Script-gesteuerte Bots bieten diese Prognosen eine ausgezeichnete Möglichkeit, den Fokus auf Posts zu legen, die sich wünschenswert entwickeln, und so die Macht des Botnets auf vielversprechende Posts zu bündeln, um bessere Ergebnisse zu erreichen.

Die konvexe Hülle der verstorbenen (State *dead*, siehe 4.2) Posts zeichnet sich dadurch aus, dass ihr Wachstum entlang der y-Achse schon nach ca. 12 Minuten beim Score von etwa 50 stagniert. Obwohl jeder Post bei einem Score von 1 anfängt und nur in New auftaucht, teilen sich die Hüllen der verstorbenen und der populären Posts sehr schnell, schon nach 25 Minuten findet kein einziger Punkt, der zu einem verstorbenen Post gehört, in der populären Hülle. Das bedeutet, dass die erste der in 2.2.2 beschriebenen kritischen Phasen im Leben eines populären Posts maximal 25 Minuten dauert. Hat der Post in diesem Abschnitt nicht genug Upvotes gesammelt, wird er es nicht mehr in Hot schaffen. Korreliert könnte dies vermutlich damit sein, dass der Post, selbst wenn er hiernach noch in Rising auftaucht (was möglich ist, da die konvexe Hülle der Posts mit State *rising* in den ersten 40 Minuten ja alle verstorbenen Punkte umfasst), er nicht hoch genug in Rising abgebildet wird, und so nicht von genug Nutzern gesehen wird, um die erforderliche Menge an Upvotes zu erhalten, um nach Hot zu gelangen.

Die konvexen Hüllen einzelner Gruppen geben also schon eine sehr gute Vorstellung davon, zu welcher Gruppe ein Post gehört, wie er sich entwickeln könnte, wann er überwältigende Chancen hat, populär zu werden und wann keine mehr.

[33] Dieser Versuch ist als Python-Script auf https://github.com/itsMik4n/Reddit-Frontpage-Data-Collection/blob/main/tests/convex_hull_affiliation_test.py verfügbar und durch Kommentare näher erläutert.
[34] Hier ließe sich nun die Konfidenz mittels von Konfidenzintervallen berechnen, doch dies überschreitet leider den Rahmen dieser Arbeit. Genaueres zu Konfidenzintervallen findet man unter [https://tu-dresden.de/gsw/phil/iso/mes/ressourcen/dateien/prof/lehre/sem/folder-2008-10-21-40971359000/Konfidenzintervalle.pdf?lang=en]

4.4.3. Die Upvote-Ratio visualisiert im Violin-Plot

Violin-Plots sind eine nützliche Alternative zu Box-Plots, sie können als Kreuzung zwischen Box- und Density-Plot gesehen werden. Zu jeglichen im Box-Plot dargestellten Informationen kommt die Verteilungsdichte der Werte hinzu, bemessen durch die Breite des Violin-Plots. Werden die Datenpunkte, getrennt nach State, als Violin-Plots dargestellt, wie in *Fig* 5, zeigt sich ein erwartetes Muster: die in New verstorbenen Posts haben die niedrigsten Upvote-Ratios, ihr Median liegt bei 0.92, jener derjenigen Posts, die in Rising geblieben sind, beträgt 0.95 und die populären Posts haben einen Median von 0.97. Eindeutig spielt die Upvote-Ratio eine relativ wichtige Rolle in den Entscheidungen der Algorithmen.

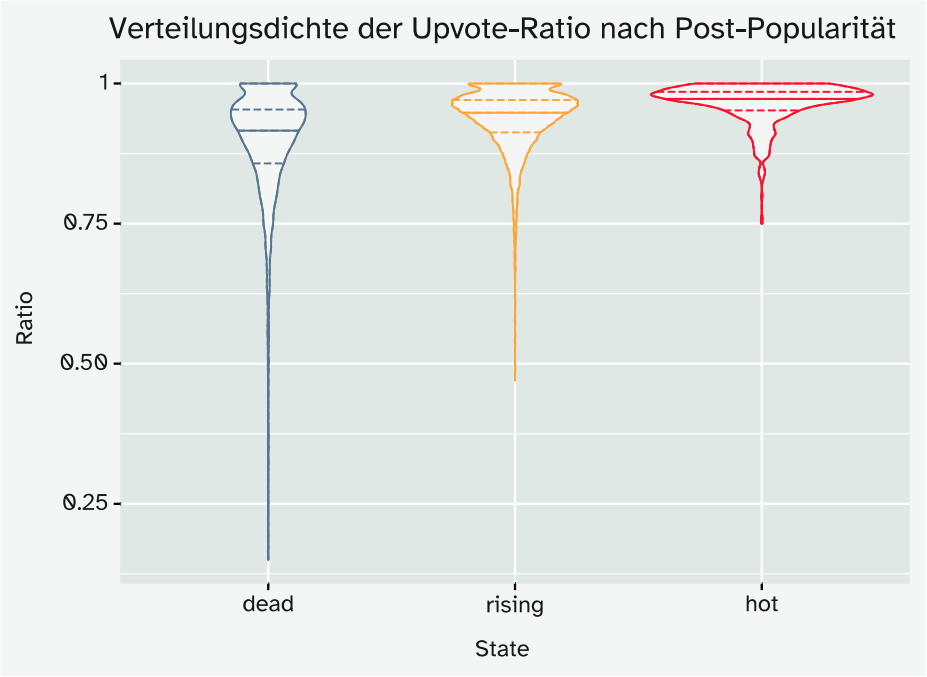

Fig 5 Violin-Plots der Upvote-Ratio getrennt nach Post-Popularität

Ein herausstechendes Merkmal der populären Posts ist die Absenz jeglicher Ausreißer mit Werten unter 0.75. Kein einziger Datenpunkt liegt unter diesem Wert, es kann hier also wieder eine jener Grenzen des Algorithmus vermutet werden, welche, falls unterschritten, einem Post verschwindende Chancen zur Popularität gibt. Somit ist auch die Upvote-Ratio von großer Wichtigkeit für die Prognose und Manipulation von Posts. Angreifer könnten sie zur Zensur ungewünschter Inhalte nutzen; denn in New senken schon kleine Mengen an Downvotes die Upvote-Ratio beträchtlich. Dies kommt daher, dass ja erst wenige Upvotes vorhanden sind, und somit ein einzelnes Downvote mehr wiegt.

Die Upvote-Ratio ist vermutlich derjenige Parameter, der am meisten von der inhaltlichen Qualität eines Posts abhängt, wie auch in 2.1.2 angesprochen: was als Inhalt von niedriger Qualität gesehen wird, wird gedownvotet, es sinkt die Upvote-Ratio. Deshalb kann die Upvote-

Ratio wohl auch als Indikator für die Qualität gesehen werden. Konkret heißt das für böswillige Akteure, dass Posts, welche mittels Bot-Accounts schon mit Upvotes gepusht wurden und dennoch niedrige Upvote-Ratios aufweisen, vermutlich inhaltliche Probleme haben. Dadurch können also diese unbeeinflussbaren, subjektiven Parameter angepasst werden, sodass die Manipulation der beeinflussbaren Parameter leichter gelingt.

5. Fazit

Reddits popularitätsbasierte System filtert von Benutzerinnen erstellte Inhalte und präsentiert populäre Posts auf der Frontpage. Die dafür eingesetzten Algorithmen sind nicht öffentlich bekannt, können jedoch gemessen und in Regeln und Bedingungen gefasst werden. Posts müssen nach der Erstellung eine gewisse Menge an Upvotes erhalten, damit ihnen eine Chance zur Popularität geboten wird. Dabei entscheidet eine relativ kleine Gruppe, ob dies geschieht oder der Post „in New verloren geht“.

Bots – automatisierte Skripts – können diese besonderen Eigenschaften Reddits ausnutzen, um Inhalten, wie beispielsweise Fake News, zur Popularität zu verhelfen und große Massen an Menschen zu erreichen. Dies ist definiert als Manipulation der Frontpage und auf Reddit verboten. Bösartige Bots müssen nicht sehr komplex sein und sind in primitiver Form relativ leicht zu erstellen und auszuführen.

Nur populäre Posts erreichen große Mengen an Benutzerinnen. Damit ein Post populär wird, muss er in Rising auftauchen und hier genug Upvotes sammeln. Die Grenze dafür liegt bei 104 Minuten, ist der Post nach dieser Zeit noch nicht populär, wird er es vermutlich nie sein. Ob der Post überhaupt eine Chance besitzt, populär zu werden und auf der Frontpage zu landen, und wie wahrscheinlich er dies erreichen wird, lässt sich schon sehr früh in der Entwicklung durch statistische Mittel vorhersagen, zur Demonstration dessen wurde die konvexe Hülle verwendet. Downvotes haben vor allem anfangs große Auswirkungen, die Upvote-Ratio zeigt, dass Posts, um populär zu werden, mehr als 75% geupvotet sein müssen. Ausnutzen lässt sich dies vor allem anfangs, wenn ein Post sehr wenige Upvotes erhalten hat und ein Downvote die Upvote-Ratio stark beeinflusst.

Literaturverzeichnis

Barthel, M., Stocking, G., Holcomb, J. & Mitchell, A. (2016, 25. Februar). *Seven-in-Ten Reddit Users Get News on the Site*. Pew Research Center's Journalism Project. https://www.journalism.org/2016/02/25/seven-in-ten-reddit-users-get-news-on-the-site/, zugegriffen am 23.2.2021

Bruns, A. (2017). *Echo Chamber? What Echo Chamber? Reviewing the Evidence*. Future of Journalism conference.

Carman, M., Koerber, M., Li, J., Choo, K. R. & Ashman, H. (2018). *Manipulating visibility of political and apolitical threads on Reddit via score boosting*. 184–190. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00037, zugegriffen am 10.2.2021

Fiesler, C., Jiang, J. A., McCann, J., Frye, K. & Brubaker, J. R. (2018). *Reddit Rules! Characterizing an Ecosystem of Governance*. 72–81. https://cmci.colorado.edu/~cafi5706/icwsm18-redditrules.pdf, zugegriffen am 2.1.2021

Forsyth, D. (2018). *Probability and Statistics for Computer Science*. Springer International Publishing.

Gilbert, E. (2013). Widespread underprovision on Reddit. *Proceedings of the 2013 conference on Computer supported cooperative work - CSCW '13*, 0. https://doi.org/10.1145/2441776.2441866, zugegriffen am 25.2.2021

Guess, A., Nyhan, B. & Reifler, J. (2018). Selective Exposure to Misinformation: Evidence from the consumption of fake news during the 2016 U.S. presidential campaign. *European Research Council*, 0. http://www.ask-force.org/web/Fundamentalists/Guess-Selective-Exposure-to-Misinformation-Evidence-Presidential-Campaign-2018.pdf, zugegriffen am 13.2.2021

Hintze, J. L. & Nelson, R. D. (1998). Violin Plots: A Box Plot-Density Trace Synergism. *The American Statistician*, *52*(2), 181–184. https://doi.org/10.2307/2685478, zugegriffen am 4.2.2021

Hurtado, S., Ray, P. & Marculescu, R. (2019). Bot Detection in Reddit Political Discussion. *Proceedings of the Fourth International Workshop on Social Sensing - SocialSense'19*, 0. https://doi.org/10.1145/3313294.3313386, zugegriffen am 22.2.2021

Imperva Research Labs. (2020). *Bad Bot Report*. https://www.imperva.com/resources/resource-library/reports/2020-Bad-Bot-Report/, zugegriffen am 20.2.2021

Kastrenakes, J. (2020, 1. Dezember). *Reddit reveals daily active user count for the first time: 52 million*. The Verge. https://www.theverge.com/2020/12/1/21754984/reddit-dau-daily-users-revealed, zugegriffen am 29.1.2021

Leavitt, A. & Robinson, J. J. (2017). *The Role of Information Visibility in Network Gatekeeping: Information Aggregation on Reddit during Crisis Events*. The 20th ACM Conference on Computer-Supported Cooperative Work and Social Computing. https://doi.org/10.1145/2998181.2998299, zugegriffen am 14.2.2021

reddit.com: api documentation. (o. D.). www.reddit.com., von https://www.reddit.com/dev/api/, zugegriffen am 22.2.2021

Richterich, A. (2014). 'Karma, Precious Karma!' Karmawhoring on Reddit and the Front Page's Econometrisation. *Journal of Peer Production*, *4*(1), 0. http://peerproduction.net/issues/issue-4-value-and-currency/peer-reviewed-articles/karma-precious-karma-2/, zugegriffen am 2.1.2021

Rosenberg, M., Confessore, N. & Cadwalladr, C. (2019, 19. März). *How Trump Consultants Exploited the Facebook Data of Millions*. The New York Times. https://www.nytimes.com/2018/03/17/us/politics/cambridge-analytica-trump-campaign.html?module=inline, zugegriffen am 13.2.2021

Shao, C., Ciampaglia, G. L., Varol, O., Flammini, A. & Menczer, F. (2017). The spread of fake news by social bots. *Indiana University, Bloomington*, 0. https://arxiv.org/abs/1707.07592v1, zugegriffen am 22.2.2021

The top 500 sites on the web. (2021, 8. Februar). Alexa. https://www.alexa.com/topsites/, zugegriffen am 6.2.2021

Van der Nagel, E. & Frith, J. (2015). Anonymity, pseudonymity, and the agency of online identity: Examining the social practices of r/Gonewild. *First Monday*, 0. https://doi.org/10.5210/fm.v20i3.5615, zugegriffen am 17.2.2021

Young, I. G. (2013, 9. Dezember). *Reddit's empire is founded on a flawed algorithm - Ian's Tech Notes*. Iangreenleaf Tech Notes. http://technotes.iangreenleaf.com/posts/2013-12-09-reddits-empire-is-built-on-a-flawed-algorithm.html, zugegriffen am 22.2.2021

Glossar[35]

Algorithmus	„Eine präzise, d.h. in einer festgelegten Sprache abgefasste, endliche Beschreibung eines allgemeinen Verfahrens unter Verwendung elementarer Verarbeitungsschritte zur Lösung einer gegebenen Aufgabe."[36]
API	Ein programmatischer Zugangspunkt zur Interaktion mit einem Programm.[37]
Bot	auf sozialen Medien eingesetzte automatisierte Skripts, welche Handlungen normaler Benutzerinnen durchführen können und bösartig zur Amplifikation von Nachrichten genutzt werden können.[38]
Captcha	Ein digitaler Test, der menschliche Benutzerinnen von Bots unterscheiden soll.[39]
CSV-Format	Ein vielfältiges Dateiformat zum Austausch von Daten, welches primär für Tabellen und Datensätze verwendet wird; kurz für Comma-Separated Values.[40]
Datensatz	Eine Ansammlung an Sets von Daten, die in einer Datei abgespeichert sind.[41]
Feed	„Eine Internet-Seite, deren gezeigte Information sich regelmäßig updatet, um die neueste Information zu zeigen" (übersetzt ins Deutsche)[42]
Github-Repository	Ein Verzeichnis zur Speicherung von Dateien, Skripts und Ordnerstrukturen, gehostet auf www.github.com,
JSON-Format	Ein Dateiformat zum Austausch von Daten, welches leicht für Menschen zu lesen und leicht für Maschinen zu verstehen sein soll; kurz für Javascript Object Notation.[43]

[35] Auf alle Quellen wurde zugegriffen am 25.2.2021
[36] https://wirtschaftslexikon.gabler.de/definition/algorithmus-27106
[37] https://dictionary.cambridge.org/de/worterbuch/englisch/api?q=API
[38] https://wirtschaftslexikon.gabler.de/definition/social-bots-54247
[39] https://dictionary.cambridge.org/de/worterbuch/englisch/captcha
[40] https://tools.ietf.org/html/rfc4180#section-2
[41] https://wirtschaftslexikon.gabler.de/definition/datensatz-28503
[42] https://dictionary.cambridge.org/de/worterbuch/englisch/feed
[43] https://www.json.org/json-en.html

lineare Regression	„Bei der Linearen Regression handelt es sich um eine spezielle Form der Regressionsanalyse, bei der nur solche Zusammenhänge betrachtet werden, bei denen die abhängigen Variablen eine lineare Kombination der Regressionskoeffizienten aufweisen.“[44]
Programmbibliothek (kurz Bibliothek)	„Eine Programmbibliothek bezeichnet in der Programmierung eine Sammlung von Programmfunktionen für zusammengehörende Aufgaben. Bibliotheken sind im Unterschied zu Programmen keine eigenständig lauffähigen Einheiten, sondern Hilfsmodule, die Programmen zur Verfügung gestellt werden.”[45]
Streudiagramm	„Ein Streudiagramm (engl. Scatterplot) ist die graphische Darstellung von beobachteten Wertepaaren zweier statistischer Merkmale“[46]
Wrapper	(im verwendeten Kontext) eine Programmbibliothek, die ein anderes Programm (in dem Fall Reddits API) umschließt, und erleichterte Interaktion mit diesem bietet.
	Im Datensatz vorkommende Werte
ID	Zum selben Post gehörige Datenpunkte haben dieselbe ID (Identifikationsnummer).
Time	Die Zeit in Minuten, die seit der Erstellung des zum Datenpunkt gehörigen Posts vergangen ist.
Score	Der Score eines Posts ist die Differenz der gesamt erhaltenen Upvotes zur Anzahl der erhaltenen Downvotes.
Ratio	Die Upvote-Ratio stellt den Anteil der Upvotes an den insgesamt erhaltenen Stimmen dar.
Comments	Die Anzahl aller unter dem Post erstellten, auch gelöschten, Kommentare.

[44] https://www.bwl-lexikon.de/wiki/lineare-regression/
[45] https://deacademic.com/dic.nsf/dewiki/1133576
[46] https://deacademic.com/dic.nsf/dewiki/1339088

Stickied	*True*, wenn der Post zu dem Zeitpunkt an der Frontpage des Subreddits gepinnt ist[47].
Rising	*True*, wenn der zugehörige Post zum Zeitpunkt des Updates in Rising zu finden ist.
Hot	True, wenn der Post im Moment in Hot (auf der Frontpage) zu finden ist.
State	*dead*[48], falls der zugehörige Post im gesamten Messzeitraum weder in Rising noch in Hot war; *rising*, falls er in Rising, jedoch nie in Hot war; *hot*, falls er an mindestens einem Updatezeitpunkt in Hot, und somit auf der Frontpage, war.

[47] Wurde nur erhoben, um einen Bias zu vermeiden (von Moderatoren angepinnte Posts werden unverhältnismäßig hoch geupvotet sein und sind automatisch „in Hot“) Keiner der gesammelten Posts war im Messzeitraum *stickied*.
[48] Die gewählte Bezeichnung *dead* für einen Post, der weder in Rising, noch in Hot war, bezieht sich, mangels einer allgemeingültigen Bezeichnung, auf die verbreitete Aussage „the post died in new“ („Der Post ist in New gestorben“) und hat keine weitere Relevanz

Eigenständigkeitserklärung

Ich, David Mikan, erkläre, dass ich diese vorwissenschaftliche Arbeit eigenständig angefertigt habe und nur die in Literaturverzeichnis und Fußnoten angegebenen Quellen verwendet habe.

Datum Unterschrift