IDENTIFYING INFLUENCER MARKET MANIPULATIONS AND RECOMMENDING ENGAGING ACCOUNTS

by
ÖZGÜN YARGI

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Master of Science

Sabancı University
December 2022

ABSTRACT

IDENTIFYING INFLUENCER MARKET MANIPULATIONS AND RECOMMENDING ENGAGING ACCOUNTS

ÖZGÜN YARGI

DATA SCIENCE M.A. THESIS, DECEMBER 2022

Thesis Supervisor: Asst. Prof. Onur Varol

Keywords: influencer, influencer market, engagement metric, fake engagement, bot accounts, recommendation system, similar content

Today, as consumers are channeled to social media platforms, the demands of companies and brands to advertise and promote on social media platforms are constantly increasing. Companies and brands that have been searching for different advertising and promotion approaches use influencers. They make an agreement with the influencer on a fee per story or per post to promote their products. The increasing number of social media users has led to the growth of the race within the influencer market. Some influencers have begun to use various methods that boost their engagement metrics artificially. Purchasing bot followers or automated engagement are examples of such manipulative efforts. As a result of this, although the engagement numbers of influencer seem high, they have blurred the organic engagement rate and misled the companies that hire influencers.

In this thesis, we present a new metric, the CRE (capture-recapture engagement) score, to the literature that can detect organic interactions more accurately than existing interaction metrics used in influencer marketing agencies. As a result of the evaluations made, it has been observed that the metric we presented offers better performance than the metrics used in the literature. In addition to this, we introduce an influencer recommendation system built by using the CRE score. The proposed system can identify influencers that have higher engagements while preserving the similarity of the profile content with the target user. This approach provides opportunities to select highly engaging but less popular influencers.

ÖZET

FENOMEN MARKET MANIPÜLASYONLARINI AÇIKLAMA VE İLGI ÇEKICI HESAPLAR ÖNERME

ÖZGÜN YARGI

VERI BİLİMİ YÜKSEK LİSANS TEZİ, ARALIK 2022

Tez Danışmanı: Dr. Öğr. Üyesi Onur Varol

Anahtar Kelimeler: fenomen, fenomen marketi, etkileşim metriği, sahte etkileşim, bot hesap, tavsiye sistemi, benzer tema

Günümüzde, tüketicilerin sosyal medya platformlarına kanalize olması ile birlikte firmaların ve markaların, sosyal medya platformları üzerinden reklam ve tanıtım yapma talepleri de sürekli olarak artmaktadır. Farklı reklam ve tanıtım arayışı içerisine girmiş olan firmalar ve markalar, bir metod olarak da fenomenleri kullanmaktadır. Fenomenlerle, hikaye başı veya paylaşım başı bir ücret üzerinden anlaşarak, ürünlerinin tanıtımını yaptırmaktadırlar. Sosyal medya kullanıcıların giderek artması, fenomen marketi içerisindeki yarışın da büyümesine yol açmıştır. Bazı fenomenler literatürde bahsedilen etkileşim metriklerini olduğundan daha yüksek gösteren çeşitli metodlar kullanmaya başlamışlardır. Bunlardan bir tanesi de bot hesaplara, kendi hesaplarını takip ettirmeleridir. Bunun sayesinde, her ne kadar fenomenlerin etkileşim sayıları yüksek gibi gözüküyor olsa da, organik etkileşim oranını bulanıklaştırmıştır.

Bu tezde, literatüre, organik etkileşimleri, kullanılmakta olan etkileşim metriklerine göre daha doğru tespit edebilecek, yeni bir metrik sunuyoruz. Sunduğumuz bu metrik, yapılan değerlendirmeler sonucunda, literatürde kullanılmakta olan metriklere göre daha iyi bir performans sunduğu gözlemlenmiştir. Bunun yanında, bu metriği kullanarak oluşturulmuş bir fenomen tavsiye sistemi tanıtıyoruz. Bu sistem sayesinde, hedef fenomene göre organik etkileşim miktarı daha yüksek ve aynı temayı konu alan başka bir fenomen seçebilmek mümkün kılınıyor.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to everyone who has helped and supported me during the course of my master's thesis.

First and foremost, I would like to thank my thesis supervisor, Onur Varol, for their invaluable guidance, support, and encouragement throughout this journey. Their expertise, insights, and constructive feedback have been instrumental in shaping the direction and quality of my research. I am deeply grateful for their mentorship and guidance.

I would also like to thank the members of my thesis committee, Ezgi Akpınar and Uzay Çetin, for their valuable feedback and suggestions, which have greatly contributed to the improvement of my thesis.

I am also grateful to Sabancı University for providing me with the resources and support needed to conduct my research.

Additionally, I would like to thank my friends and colleagues who have provided me with moral support, encouragement, and helpful advice throughout the duration of my thesis.

Finally, I would like to thank my family for their love and unwavering support, especially my dad, who supported me in any decision I made and encouraged me to do my best all the time. Their encouragement and belief in me have been a constant source of strength and motivation.

I am truly grateful to everyone who has contributed to this journey. Thank you all.

To my grandmother and grandfather

TABLE OF CONTENTS

LIST OF TABLES ..... **ix**

LIST OF FIGURES ..... **x**

1. INTRODUCTION **1**

2. Literature Review **4**

3. Methodology **7**

3.1. Dataset Collection 7

3.2. Capture Recapture 11

3.2.1. Definition 11

3.2.2. Usage Areas of Capture Recapture 12

3.2.3. Implementation of Capture Recapture to Social Enviroments . 12

3.3. Validation of Pretrained Models 15

3.3.1. Validating image based models 16

3.3.2. Validation text based models 19

3.4. Implementation 22

3.4.1. Feature Engineering 22

3.4.2. Recommendation System 26

3.5. Validation of Recommendation System 30

3.5.1. Validation Dataset Collection 30

3.5.2. Survey Creation 32

3.5.3. Performance Evaluation of Recommendation system 38

4. Discussion & Results **47**

5. Conclusion **50**

BIBLIOGRAPHY **52**

APPENDIX A **55**

LIST OF TABLES

Table 3.1. **Scraped Data Insight**: A sample of information scraped from Instagram that will further be used in similarity and engagement calculations. 10

Table 3.2. **F1 score comparison of SVM models trained on image embeddings**: F1 scores of SVM models in different sample size ranges, trained by using the embeddings of VGG16, Xception, and VGG19. Xception provides a higher score, while VGG16 embeddings outperform others on the complete dataset. 17

Table 3.3. **F1 score comparison of SVM models trained on text embeddings**: F1 scores' of each SVM model on different sample sizes. Outcomes of pre-trained models were used as an input to SVM. Embeddings taken from MiniLM-L6 led the best overall performance against Sentence-BERT and NLI. 20

Table 3.4. **Network Feature Explanations**: Features extracted from commenter, hashtag and tag networks. 25

Table 4.1. **AUC scores of compared metrics**: Performance comparison of used metrics. Annotator_3 contributes the best performance with AUC score 0.67 while Post number metric shows a lack of performance with an AUC score 0.453. The introduced metric takes 3^{th} place which outperforms all industry used metrics as well as annotator_4 and annotator_2 with AUC score 0.624. 47

Table 4.2. **Spearman Correlation Score Comparison of Used Presentations**: Performance comparison of used presentations Image based presentation of influencer contributes the best performance with Spearman score 0.2689 while System base representation shows a slightly lower performance with Spearman score 0.2449. 49

LIST OF FIGURES

Figure 3.1. **Follower Number Histograms Collected by Distinct Approaches**: Distribution of follower number counts collected by distinct approaches seem to be similar one another. The kurtosis in the collection created by using a heuristic is lower than the other approaches. 8

Figure 3.2. **Scrape Methodology** After executing the script, the program logs in and goes to the target account's page. While in page, it reaches the metadata by using *?___a=1* extension and scrapes. After the scraping operation finished for account page, the script iterates over first 36 posts one by one and scrapes the required information by using *?___a=1* extension one more time. After every required data is collected, the data is stored inside of the local drive. 9

Figure 3.3. **Capture Recapture Score Calculation in Social Media**: Visual explanation of how capture recapture score is calculated 13

Figure 3.4. **Cumulative vs Pairwise**: Figure 3.4a shows the distribution of calculated CR scores along follower numbers. Figure 3.4b shows the distribution of calculated CR scores along follower numbers. When the linear regression line is fit, the Spearman correlation of the cumulatively calculated scores is higher than the pairwisely calculated Spearman correlation, which led to the use of the cumulatively calculated scores in further experiments. 14

Figure 3.5. **UMAP Visualizations of VGG16 and Xception Embeddings** 3.5a shows distribution of Xception embeddings and 3.5b shows the distribution of VGG16 embeddings. Embeddings that was taken from 3.5b presents more distinguishable clusters compared to 3.5a..... 18

Figure 3.6. **UMAP Visualizations of Accounts Based on Embeddings of Xception and VGG16** Figure 3.6a shows the distribution of accounts using Xception embeddings, and figure 3.6b shows the distribution of accounts using VGG16 embeddings. As it can be seen, there is no big difference from the perspective of account representation. 19
Figure 3.7. **UMAP Visualizations of MiniLM-L6 and Sentence-BERT Embeddings** 3.7a shows the distribution of MiniLM-L6 embeddings, and 3.7b shows the distribution of Sentence-BERT embeddings. Embeddings that was taken from 3.7a presents similar clusters compared to 3.7b. 21
Figure 3.8. **UMAP Visualizations of Accounts Based on embeddings of MiniLM-L6 and Sentence-BERT** Figure 3.8a shows the distribution of accounts using MiniLM-L6 embeddings and figure 3.8b shows the distribution of accounts using Sentence-BERT embeddings. As it can be seen, there is no big difference from the perspective of account representation. 22
Figure 3.9. **Network Visualization** Figure 3.9a shows the network created by using commenters and figure 3.9b shows the network created by using hashtags and figure 3.9c shows the network created by using tags. 24
Figure 3.10. **Similarity comparison by data type** *Image vs Text* type shows correlation while *Image-Network* and *Text-Network* does not share any link. 26
Figure 3.11. **Piechart distribution of data type ratios** Image type data has influence on 76.2% of the dimension while metadata type data has only 0.1%. 26
Figure 3.12. **Follower Number vs CR Scores** Image type data has influence on 76.2% of the dimension while metadata type data has only 0.1%. 28
Figure 3.13. **Capture Recapture Score vs Engagement Rate** 3.13a shows the density of capture recapture scores per influencer type, and 3.13b shows the density of engagement rates per influencer type. Usage of engagement rate provides a better metric to use than capture recapture scores since generated rate does not distinct per influencer type 29
Figure 3.14. **Recommendation System Illustration** 29
Figure 3.15. **Follower Number Histogram**: Histogram of follower numbers separated by influencer type. 31

Figure 3.16. **Validation of Created Subsets**: The left-hand side figure shows the distribution of calculated engagement rate differences among chosen influencers. Each influencer type contains 45 samples. The figure on the right shows the calculated similarity scores between the influencers. Each influencer type contains 45 sample. 32
Figure 3.17. **Response Histogram**: Response counts of each annotator for given pairwise influencer comparison surveys..... 33
Figure 3.18. **Survey responses by question**: Figure 3.18a and 3.18b shows the counts of scores responded by each annotator. Most of the responses are covered by 7 and 8 for both histograms. Figure 3.18c shows the counts of scores responded by each annotator. Most of the responses are covered by 0 and 1..... 34
Figure 3.19. **Time Spent on Surveys per Annotator**: Survey response period histogram per annotator..... 35
Figure 3.20. **Internal consistency validation by using engagement rates**: Each sample represents the variance of engagement rate responses of an influencer. Each influencer is resolved at least 9 times by an annotator which allows to evaluate consistency by checking the variance of the resolved 9 responses..... 36
Figure 3.21. **Annotator Similarity Comparison by Engagement Similarity**: Figure 3.21a shows the agreement scores between annotators in terms of engagement rate by using Cohen's Kappa. Figure 3.21b shows the engagement rate similarity agreement between annotators calculated by continuous responses..... 37
Figure 3.22. **Annotator Similarity Comparison by Content Similarity**: Figure 3.22a shows the agreement scores between annotators in terms of content by using Cohen's Kappa. Figure 3.22b Shows the content similarity agreement between annotators calculated by continuous responses. 38
Figure 3.23. **Analogy Comparisons between Annotators and System**: Figure 3.23a shows the agreement level between annotators and the system from the engagement side. Figure 3.23b shows the agreement level between annotators and the system from similarity side..... 39
Figure 3.24. **Engagement Diversity**: Understanding engagement differences between compared influencers is crucial since the task is to identify the influencer that has the highest engagement rate. The figure shows the level of correlation in engagement diversity between annotators and the system..... 41

Figure 3.25. **Diversity Binary Comparisons**: Figure 3.25a shows the distribution of the system's engagement diversity scores for binary annotator decisions. Figure 3.25b shows the distribution of system's engagement diversity scores for binary annotator decisions. 42
Figure 3.26. **ROC Scores**: Figure 3.26a shows the receiver operating characteristics curve when annotator decisions are taken as ground truth. 3.26b shows the receiver operating characteristics curve when system decisions are taken as ground truth. 42
Figure 3.27. **ROC Curves of suggested metrics**: As annotator decisions are taken as ground truth, AUC scores of different literature metrics are compared with the newly introduced engagement rate metric. Introduced engagement rate metric appears to be outperforms others when annotators' decisions are taken as ground truth 43
Figure 3.28. **ROC Curves of annotators and the system**: Figure 3.28a shows the performances of annotators individually. Annotator_3 achieves the highest performance with an AUC 0.668, while annotator_2 achieves the worst performance with an AUC 0.519. The performance of the introduced metric achieves the 3^{th} best performance by overtaking annotator_2 and annotator_4. 44
Figure 3.29. **Fit Comparison per Data Type**: As annotator decisions are taken as ground truth, AUC scores of different literature metrics are compared with the newly introduced engagement rate metric. Introduced engagement rate metric appears to be outperforms others when annotators' decisions are taken as ground truth 45
Figure 3.30. **Internal Consistency of Influencers**: Scatter shows the distribution of variances and meanss of embedding cosine similarities per influencer. If an influencer shares similar content most of the time, we expect to see the mean close to 1 and variance close to 0. ... 46

Figure A.1. **Scraped Data Structure**: General look to the dataset directory tree before having data pre-processing 56
Figure A.2. **Directory Structure after Preprocessing**: The structure of the dataset Folder after preprocessing is finished. 58

1. INTRODUCTION

Consumers' high level of anxiety over traditional marketing corruption led to the search for new marketing trends. According to the current estimations, it is shown that, 95% of Generation Z, who is born between 1995 and 2015, owns a mobile phone while 93% of Generation Y, who is born between 1980 and 1994, owns and use mobile phones eagerly (Holland (2019)). This state of people accelerates the growth of social media platforms, and firms are impelled to take influencers under their wing as propulsion. By activating unique features of influencers, consumers are canalized target's offerings with concern of maximum profitability. In literature, this concept is called as "Online Influencer Marketing" (Leung, Gu & Palmatier (2022)) which elaborates on influencers' role in promoting firms. The relationship between firms and influencers is generally reciprocal since firms tend to nourish influencers to increase the engagement rate. This investment returns as higher perceptibility of offerings. The investment proportions of firms to influencers rapidly increase as the expected spending of marketers on influencer marketing is $16.4 billion by the end of 2022 (Leung, Gu, Li, Zhang & Palmatier (2022)).

The terminology of social influencer is defined as a person who managed to sway sizeable social networks that follow it (De Veirman, Cauberghe & Hudders (2017)). The expansion of the influencer market attracts attention that marketers start to appreciate the effect of content presentation (Akpinar & Berger (2017); Evans, Wojdynski & Grubbs Hoy (2019); Hughes, Swaminathan & Brooks (2019); Ki & Kim (2019); Lou & Yuan (2019)), with both influencer and firm characteristics (Breves, Liebers, Abt & Kunze (2019); De Veirman et al. (2017); Hughes et al. (2019); Valsesia, Proserpio & Nunes (2020)). Circumscribing the research field to understand the effect of influencer marketing on promoting products is inadequate. Although the developers of social media platforms may have seen their establishment as a step toward a more open marketplace of ideas, already strong organizations have consistently tried to increase their influence via the use of social media. This led bots to exist in miscellaneous places in different social media platforms for different motivations. A social bot is a computer algorithm that creates material automatically and engages

with users on social media in an effort to mimic and maybe change their behavior (Ferrara, Varol, Davis, Menczer & Flammini (2016)).

On Twitter, social bots have been forced to radiate propaganda to the benefit of falling regimes by state actors (Wirth, Menchen-Trevino & Moore (2019)). On streaming channel platform Twitch, bots are generally used to manage interactions between streamers and the audience (Seering, Flores, Savage & Hammer (2018)). On Instagram, social bots are used to deceive users and firms by reflecting obtained like numbers higher than actuality (Sen, Aggarwal, Mian, Singh, Kumaraguru & Datta (2018)). One metric that is generally used by firms while choosing influencer is like number (Spr (2019)). The quantity of the like number has an effect on the payment amount that the influencer demands. Because of this, understanding and detecting the manipulations on engagement metrics is must need to enhance firm performance and hoist offers.

In this thesis, we introduce a novel metric, the CRE (capture-recapture engagement) score, that is inspired by a frequently used estimation approach in ecology science called capture-recapture (Le Cren (1965)). Unlike other metrics used in literature such as follower number, like number, introduced metric standardizes engagement rate by eliminating manipulations of artificial engagement by statistically analyzing the odds of observing the existence of frequent engagement or a lack of regular followers. The introduced metric generates a score for an influencer by criticizing commenters' number of existences in posts. To measure the effectiveness of the metric, three separate datasets including 4,527 Instagram users varying from different follower number ranges were scraped by using various heuristics. The introduced metric's performance was compared with follower number, like number, comment number by using 4 independent Instagram users' annotations on a user study as ground truth. The AUC score of the introduced metric outperforms existing metrics in the literature.

Individual recommendations, which include tasks like suggesting familiar, similar, or intriguing people, have grown to be one of the most significant types of Recommender Systems (RSs) in social networks in recent years (Guy (2018)). Besides, by using the introduced metric, a recommendation system is created for Instagram, which provides content-based information of various types (image, text, network, etc.). We introduced a new similarity calculation approach for multi-type data platforms. Image and text-based features are generated by using pre-trained deep learning models that were trained on millions of samples. The performances of pre-trained models are evaluated on Instagram influencers as categories were pre-defined by experts. The new method allows users to adjust the effect of data types on the

similarity score. The recommendation system uses these to suggest a more engaged influencer than the target influencer which engagement is introduced as the number of interaction of an influencer on a post scaled by its follower number. By using the similarity score, the suggested influencer preserves the same content as the target influencer which allows users to reach similar social networks. The proposed system may have an effect on the influencer market since the introduced system reflects more reliable engagement rate scores against literature metrics. This system may greatly reduce the influencer cost as it tends to suggest influencers with a low follower number but a high engagement rate. After used models and features are validated by various experiments, the performance of recommendation system is measured by creating more than 700 survey to generate a ground truths for content similarity and engagement rate. These surveys were taken by 4 annotators who are Insagram users.

In upcoming sections, we will try to answer the following questions: What are the methodologies for scraping a social media platform? What heuristics can be used to create a dataset that has a negative follower number kurtosis? How can an influencer be represented by using its social media activities (images it shares, captions it writes, etc.) and how can the representations be used to find similarities between influencers? How can deep learning and classic machine learning algorithms be conducted to create an influencer representation that can be used for various tasks such as influencer classification or recommendation systems? How to evaluate the performance of deep learning models on social science tasks? What metrics are used to identify the engagement rate in literature? How to evaluate the performance of engagement rate metrics?

2. Literature Review

Influencers are content producers who have built up a loyal fan base through short-form content creation, vlogging, or blogging on social media platforms such as Twitter, Tiktok, Instagram, Facebook, Twitch, etc. They grant their followers access to information about their personal, daily lives, experiences, and viewpoints. Influencer marketing, a strategy in which firms work with influencers, aims to encourage them to promote their goods by providing test products and organizing events to enhance their reputation among the followers of these influencers, whose numbers are frequently quite large. Influencers, as opposed to conventional celebrities, are seen as more approachable, realistic, and intimate, making them easy to relate to since they communicate in person with their followers and disclose private, typically secretive portions of their lives (Abidin (2016); Jensen Schau & Gilly (2003)). This might lead to para-social contact, which has been defined as the appearance of a personal connection with a media actor and increases consumer receptivity to their viewpoints and actions (Knoll, Schramm, Schallhorn & Wynistorf (2015); Colliander & Dahlén (2011)). Influencer endorsements will probably be viewed as the influencer's impartial thoughts and may have the necessary persuasive power since they are deeply personal and integrated into the continuous stream of textual and visual narratives of their personal lives (Abidin, 2015). However, according to Belanche, Casaló, Flavián & Ibáñez-Sánchez (2021), influencer-brand collaboration has an affect on the credibility of an influencer. According to the test that had been conducted by using the followers of a celebrity, influencer-product harmony positively affects the relationship between the followers' and influencer. On the other hand, paid insight leads to negative congruence.

It is crucial for firms to seek an influencer who is popular with their target market to promote their goods. For instance, previous studies discovered favorable correlations between celebrity and brand attitudes (Amos, Holmes & Strutton, 2008; Silvera & Austad, 2004). Additionally, (Schemer, Matthes, Wirth & Textor, 2008) discovered that combining a brand with artists who are highly regarded leads to favorable sentiments about the brand. Because of this, influencers with very high follower

numbers, such as celebrities and mega influencers do not fit the description since creating appealing content that fits everyone in the follower echo-system becomes more and more challenging as the number of followers increases. Because of this, to reach domain-specific consumers, micro-influencers who have a follower number between 1,000 and 100,000 can be a better choice since they only create content about what they are passionate about (Sandra (2021); Lucy (2021)).

The interest in celebrities and well-known people keeps increasing as social platforms like microblogging services become a channel for disseminating news, viewpoints, and ideas (Kwak, Lee, Park & Moon (2010)). In order to solve the social overload problem and make the feed attractive and engaging for its user, it is crucial to choose the correct celebrities to follow (Guy (2015)). There are three main techniques that are used in industry to recommend people: graph-based, interaction-based, and content-based. Additionally, it is found that content-based approaches are typically better suited for comparable person recommendations since they are more speculative in nature and can take advantage of the wide-ranging yet noisy nature of content-based approaches (Guy (2018)). The celebrity or influencer recommendation system is highly related to the content-based approaches. Since the list of probable candidates in this recommendation task is not restricted to people the user knows, it can be substantially broader than in the other recommendation tasks. This kind of suggestion is predicated on the idea of homophily (love of the same), or people's propensity to associate and bond with those who share their interests (McPherson, Smith-Lovin & Cook (2001)). Because of this, platforms like Instagram, which provide both visual and textual data regarding the content, can be great fields to demonstrate a content-based recommendation system. Bertini, Ferracani, Papucci & Del Bimbo (2020) compared the the performance of an hybrid recommendation system which is the combination of visual features of users' photo collections and collaborative filtering, with a simple collaborative filtering. The results show that the recommendations made by using visual features outperforms traditionally used collaborative filtering.

The link between indegree and engagement has lately been the subject of research that takes the influencer marketing environment into consideration. Three papers use indegree as a control variable for explaining engagement, despite the fact that we are not aware of any fieldwork that explicitly explains this association. According to Hughes et al. (2019), there is a direct correlation between an influencer's indegree and the volume of Facebook and blog likes, comments their sponsored material receives. Valsesia et al. (2020) observe a favorable but waning impact of indegree on the quantity of likes and retweets a post receives using a sample of unpaid endorsements on Twitter. Both studies explore extremely low indegree influencers and

concentrate on content in the form of postings. On the Chinese social network Sina Weibo, sponsored tweets are investigated by Leung et al. (2022). They conclude that indegree improves the benefits of influencer marketing expenditures on engagement even if they do not establish a direct relationship between indegree and engagement.

Instagram is becoming an essential marketplace. Marketers and brands utilize it to connect with potential customers for advertising. The number of likes on posts is used as a proxy for a user's social reputation, and in certain situations, advertisers pay social media influencers with a large following to promote their goods. Due to the growing industry, people have started to fudge their likes in order to represent a higher social value. Even companies, advertising, and the underlying recommender algorithms of online social environments depend on the influencer and content popularity numbers supplied on these platforms. Users frequently artificially boost their content's popularity and engagement in a number of ways to gain greater rewards, such as by leveraging bots. Such an unnatural boost in popularity might result in brand losses (Zazat (2017)). To detect fake likes, Sen et al. (2018) developed a model that detects fake likes with 83.5% precision accuracy. There were also other research that tried to detect fraud, spam Benevenuto, Magno, Rodrigues & Almeida (2010) and fake Cao, Yang, Yu & Palow (2014) users.

3. Methodology

3.1 Dataset Collection

Instagram is the platform that we targeted due to its popularity among the influencer ecosystem. Instagram has a higher share of the influencer market than other social network services because it is a widely used social network service worldwide. Unfortunately, Instagram does not provide easy access to its API for researchers. Because of this, we have created an open source scraper for Instagram by using browser automation tools like Selenium, Requests, and Beautifulsoup. With the scraping system, we basically simulated the operations of a human browsing an Instagram page and collected information available on the website's content.

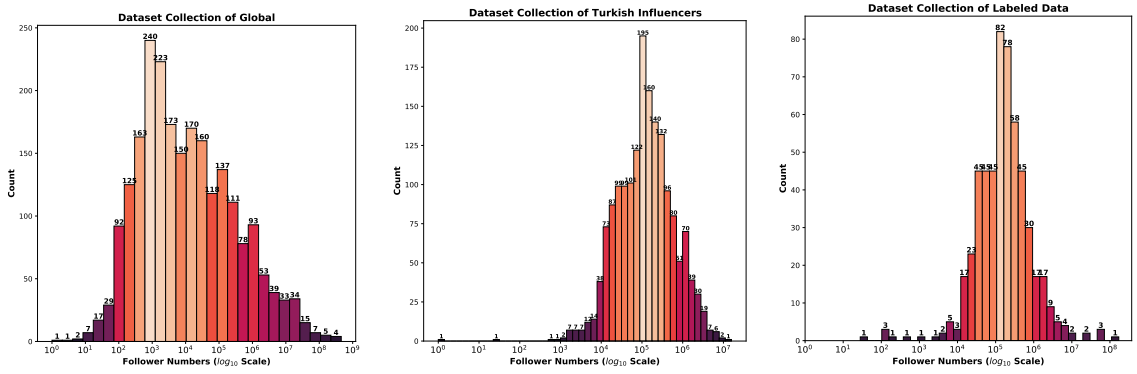
Different heuristic approaches were conducted on scraping operations for different tasks:

- **Heuristic 1**: This approach aims to collect accounts from different follower number spectrums, which have a negative kurtosis distribution. To collect accounts with a varied number of followers, the seed account was chosen as "Instagram" which has a high number of followers. According to our current hypothesis, accounts with high followers tends to follow accounts that have fewer followers. Considering this assumption, the "Instagram" official account was chosen as our seed account and we systematically collected followings of it for next scraping iterations. As the scraping operation progresses, accounts with lower followers than previous iteration will be scraped. At the end, a dataset with a wide spectrum of follower number was constructed. As it can be seen from the figure 3.1a, the number of samples from different levels is evenly distributed along the scale which provides a stable base for further analysis. Most frequently appeared accounts are the ones that have a follower

number around 10^3.

- **Heuristic 2**: This approach aims to collect accounts from different follower number spectrums. As a result, top Turkish influencer account lists were compiled from various sources. To detect the influencers on the Turkish influencer market, a website called *www.boomsocial.com* was scraped to get a list of influencers. This list contains 2022 influencers; each influencer was scraped using a custom system created for the task. Figure 3.1b shows the follower number histogram. The range scales between 1 and 10^7. However, ninety-seven percent of the data is located between 10^3 and 10^7. The most frequent accounts have 10^5 followers. This figure gives an insight into the distribution of Turkish influencers in the influencer market.

- **Heuristic 3**: This approach was used to generate a dataset for validating pretrained image and text processing models. A well-known influencer marketing platform, *"www.viralpitch.co"*, was used to collect categorized influencers, which are labeled as fashion, food, or tech-based influencers. By using this website, the usernames of 198 influencers from the fashion category, 156 influencers from the food category, and 195 influencers from the tech category were scraped. After obtaining the usernames of previously labeled influencers, related accounts were scraped using the newly created scraping tool. Figure 3.1c shows the histogram of follower numbers for the labeled dataset collection. Similar to the figure 3.1b, most frequent accounts exist throughout 10^5.



(a) Global accounts (b) Turkish influencers (c) Labeled accounts

Figure 3.1 **Follower Number Histograms Collected by Distinct Approaches**: Distribution of follower number counts collected by distinct approaches seem to be similar one another. The kurtosis in the collection created by using a heuristic is lower than the other approaches.

Instagram provides a variety of indicators to help users understand the visited account. A visitor can get an idea of how popular this account is by looking at the metadata, such as the number of followers, the number of followings, the number of

posts that have been shared so far, and the number of likes and comments for a post. By looking at the posts that are being shared, a visitor can understand the theme of the account. By looking at the captions of posts, a deep understanding of the post can be created, which fills in the parts where an image or video is not sufficient to explain the theme or concept. By looking at the tagged people or commenters, we may have an idea about the ecosystem of the account. Which people do follow this account more? Is the ecosystem big or not?

These indications may be used to figure out almost anything about the account. The main objective of the scraping operation was to gather all this relevant information to create a powerful recommendation system. Within Instagram, there are numerous places where you can scrape this information. One place that makes this operation easier is the URL extension. After the route of the account is written, such as "*https://www.instagram.com/ACCOUNTNAME/*", if you add "*?__a=1*" next to the route, it will send you to a page where relevant information regarding an account is stored in "json" format.

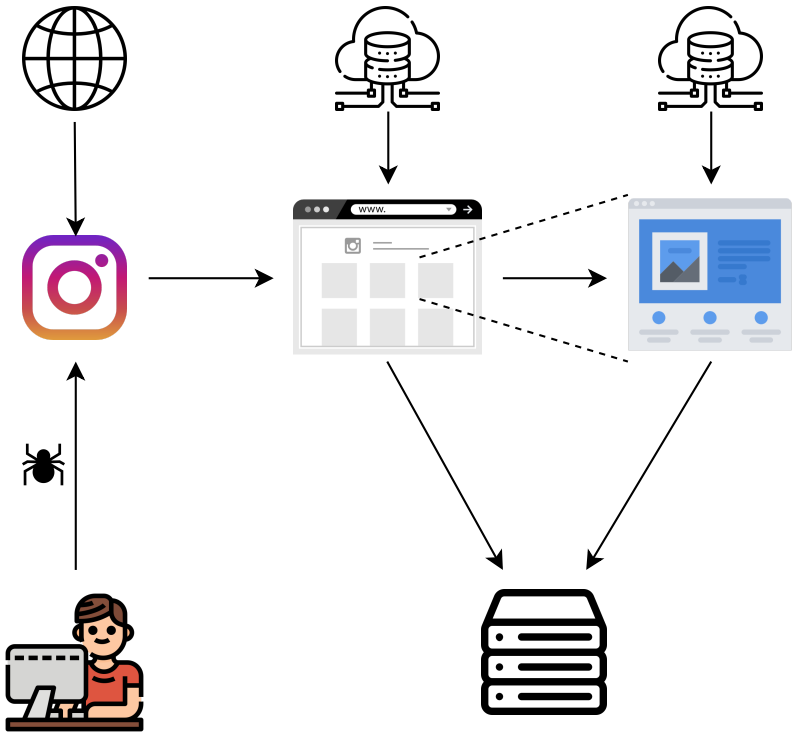


Figure 3.2 **Scrape Methodology** After executing the script, the program logs in and goes to the target account's page. While in page, it reaches the metadata by using *?__a=1* extension and scrapes. After the scraping operation finished for account page, the script iterates over first 36 posts one by one and scrapes the required information by using *?__a=1* extension one more time. After every required data is collected, the data is stored inside of the local drive.

The method used to scrape an Instagram account is as follows: After entering the target page, the first 10 followings' names are scraped to determine the next accounts

to scrape. Then, the first 20 posts' URL information is scraped by inspecting the HTML. Then, a new page is opened for the target account with *?__a=1* extension. On the new page, metadata-related information is scraped and stored in a local file. After the account page is scraped, the program visits each post URL one by one. Once again, by using the *?__a=1* extension, data related to the post is scraped, which includes the image, caption, comment information, and metadata related to the post such as like, comment number etc. At some point, Instagram made a change to its post HTML which resulted in a change to our scraping methodology. Rather than using T*?__a=1* extension, we parsed the HTML of post URL and scraped commenter related information from there. At most, 100 commenters are scraped from each post. The illustration of this approach can be observed by looking the Figure 3.2

Table 3.1 **Scraped Data Insight**: A sample of information scraped from Instagram that will further be used in similarity and engagement calculations.

Collected Data Information		
Location	**Data**	**Type**
Account Page	Id	`int`
	Posts	`int`
	Number of followers	`int`
	Number of followings	`int`
	Fullname	`string`
	Isverified	`boolean`
	Categorytype	`string`
Post	Number of likes	`int`
	Post id	`int`
	Number of comments	`int`
	Caption	`string`
	Timestamp	`int`
	Tag based info	`dict`
	Commenter based info	`dict`

Instagram provides various types of different data since users love to express their lives and thoughts by using different methods, such as sharing an image, a text that expresses their feelings, or a group of tagged people to show who is with whom. This wealth provided high-quality, unstructured data that could be scraped via Instagram. Table 3.1 shows a subset of what kind of information was scraped via Instagram. The scraped data can be separated into two sections, as one section mostly contains information about the account owner and the stats themselves, while the other section contains post-related information such as the number of likes, commenter-based information, etc. The richness of having different data types allows for the creation of a high-quality recommendation system.

Since Instagram does not provide an easy-to-use API, scraping the relevant information for this task was challenging. We used some frequently used web-based automation libraries, such as Selenium and Requests, to extract relevant information. These libraries extract information from any website by parsing the HTML, which makes them sensitive to HTML changes. Instagram also has some defense systems to overcome automated scraping operations, such as blocking the IP address, account verification systems, and blocking the account. To overcome this, we adjusted the scraping operation so that it behaves like a human being by adding randomness between each request. We were picking a number from a normal distribution with a mean of 2 and a variance of 1 that is used to determine the sleep duration between each request. This solution allowed for an increase in the number of accounts scraped in each execution. However, still, Instagram was able to detect automated scraping operation results with re-executing the scraping operation.

3.2 Capture Recapture

3.2.1 Definition

Capture-Recapture is a widely used method in ecology field (Le Cren (1965)) to estimate the size of a population. Traditionally, statisticians have classified capture-recapture models as being suited for closed or open populations (Pollock, Nichols, Brownie & Hines (1990)). In an open population, the size of the population changes with permanent additions and removals. In a close population, it is always fixed during the entire study and is easier to examine. The number of users of social media sites such as Instagram keeps changing, so they have a tendency to be an open population. However, we believed that the number of relevant followers for an account was fixed throughout the timeframe of our interest.

To estimate the size of any species, the number of captured animals is counted and marked daily. The next day, a number of animals that are already marked from the previous day as well as any that are yet to be seen are obtained and counted. Theoretically speaking, if marked animals are observed more frequently, this suggests that the general population size of the animal is small. On the other hand, if new

animals are more frequently observed, this suggests that the population size is larger. We were inspired by this approach and implemented it in the social media use cases. To estimate the engagement score of an account, we can make pairwise comparisons of the existence of commenters in both posts. Theoretically, if the number of unique commenters in each post is higher than the previously observed ones, we may expect a high engagement score, which would be close to the account's follower number. This would increase the sample size at an enormous rate.

3.2.2 Usage Areas of Capture Recapture

The capture-recapture methodology is applicable to various fields. Apart from being used on ecology to estimate the population (Manly (1970)), It was being used to estimate Botnet populations in which they track spamming behaviours. They used a capture-recapture variation on data from a big spam filtering service that gets 300 million email messages per day from over 8,000 different domains. They examined the population variations over a two-week period and compared estimations of the size of the Storm botnet to those obtained using more traditional estimation methods in order to validate their methodology. The approach can estimate the size of the population of spamming bots based on spam samples with just a 4–10% (Hao & Feamster (2008)).

3.2.3 Implementation of Capture Recapture to Social Enviroments

Figure 3.3 visualizes the implementation of capture-recapture on social network sites. We used comment information to calculate the engagement score. Because encountering a bot in the comment section is less common than encountering one in the "like" section. Moreover, the comment section provides more unique names for the commenters, which enables us to use this method.

To calculate capture and recapture scores, two distinct attitudes are followed. One approach generates a score by comparing post commenters with the commenters that were seen until now. By using this approach, more familiar commenters can be detected since we compare every post with a larger commenter pool as iteration increases. To illustrate, suppose an influencer shared 5 different posts. To generate a capture-recapture score using cumulative calculation, each post is compared to

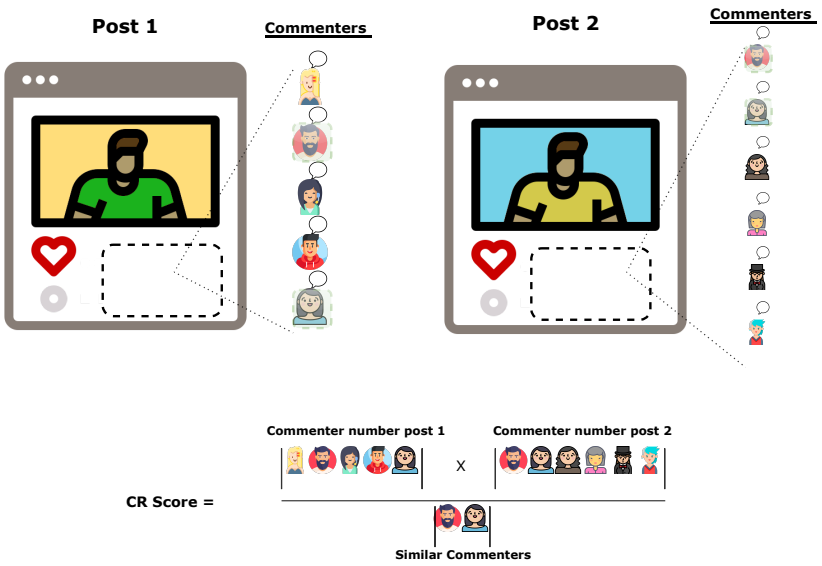
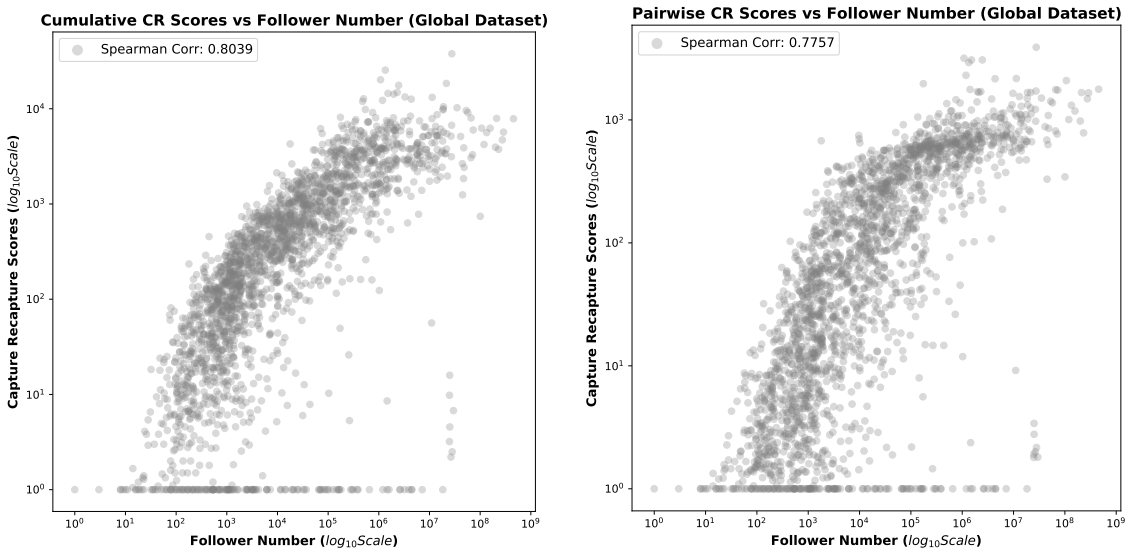


Figure 3.3 **Capture Recapture Score Calculation in Social Media**: Visual explanation of how capture recapture score is calculated

the others one by one, resulting in a total $\frac{5 \times 4}{2} = 10$ iteration. As the iteration of comparisons goes on, the memory that holds the names of commenters increases, and it generates a score for each comparison by using the commenters inside the memory. The mean of the 10 calculated capture-recapture scores is taken, which will reflect the general score of the influencer. The other method generates a score by only considering pairwise compared commenters. Thus, in each comparison, the memory is reset, which reduces the score of the capture-recapture scores compared to the cumulative calculation. However, the advantage of this calculation is that it reflects the size of the larger community for an influencer who comments on every post one by one.

(3.1)
$$Capture\ Recapture\ Score = \frac{\#\ commenters\ on\ post\ 1 \times \#\ commenters\ on\ post\ 2}{\#\ commenters\ exist\ in\ both\ post}$$

Both approaches are tested on the global collection dataset and generated scores. To understand the association between follower numbers and capture-recapture scores, distinct scatters were visualized by using both approaches. Figure 3.4 compares the distribution of both approaches along follower numbers. Both axes were scaled to log_{10} to get a better visual interpretation. After using both approaches, the capture-recapture scores calculated by the cumulative approach were distributed on



(a) Cumulative CR scores (b) Pairwise CR scores

Figure 3.4 **Cumulative vs Pairwise**: Figure 3.4a shows the distribution of calculated CR scores along follower numbers. Figure 3.4b shows the distribution of calculated CR scores along follower numbers. When the linear regression line is fit, the Spearman correlation of the cumulatively calculated scores is higher than the pairwisely calculated Spearman correlation, which led to the use of the cumulatively calculated scores in further experiments.

a larger scale than those calculated by the pairwise approach, which was expected. This makes the cumulative approach a better method to generate capture-recapture scores. Besides, to see the correlation between the capture-recapture score and follower number, the Spearman correlation score was calculated. It showed that capture-recapture scores calculated by cumulative methodology have a higher correlation between follower numbers than pairwise methodology, which makes the cumulative approach more useful. Because of these, for further experiments, capture-recapture scores will be calculated using the cumulative approach.

3.3 Validation of Pretrained Models

Social media platforms are used as tools to share moments with others. The most efficient way is to use visual materials and explanations regarding the content. This necessitates the use of visual and textual data types among the social media platforms. Since they are used to show the content of the post and give insight about the account holder, these data types are also the most efficient ones when finding similarities between content. Images are tensors that hold $3 \times imageWidthInPixel \times imageHeightInPixel$. The red, green, and blue (RGB) spectrum is represented by three different arrays. By using these tensors, computers visualize images accordingly. On the other hand, text data is stored by using ASCII table values, as each character has a corresponding integer value. Since the storage types of images and text are different, a common ground is needed to make both of them usable. Because of this, both text and image data have to be represented in the same manner. This can be accomplished with the help of deep learning models. Deep learning models may represent the given input structure in a fixed, vectorized, dimensional space. This feature of deep learning models makes them suitable for creating common ground. However, the best representations of image and text data can be kept from models that have been trained using large amounts of data. For those without adequate computational power, this can be quite costly. Luckily, some platforms, such as Huggingface or Python libraries, such as Keras provide models that have been trained on large amounts of data. These pre-trained models can later be used to create a common representation structure for image and text data.

We utilized classified influencer accounts in an attempt to validate the pretrained models. For each influencer, whether they fall within the tech, food, or fashion categories or not, pre-trained models are used to retrieve post-related information such as photo and caption embeddings. These embeddings are then used as input for influencer classification tasks in the tech, fashion, and food categories. Since the accuracy of a model is highly correlated with the given input, the model that has the best accuracy also reflects the best-working pre-trained model for social media data. Because of this, 6 different pre-trained models were chosen to compare, 3 of them for text data and the rest for visual data.

3.3.1 Validating image based models

Three pre-trained models were chosen to extract the embeddings of images, which are VGG16, Xception and VGG19. The embeddings taken from these models were used as features to represent images. These vectoral representations are later fed to the SVM classifier, which uses the RBF kernel. This trained model provides an environment for evaluating the quality of features extracted from pre-trained models, where a higher F1 score is the used metric. SVM with RBF kernels is the best choice for a given task because extracted vectorized image representations have a high dimensionality compared to sample size.

- **VGG16** is a 16-layer convolutional neural network structure. A pre-trained version of the VGG16 structure is used to extract embeddings, which was trained using the ImageNet dataset (Simonyan & Zisserman (2014)) which was proven to be a state-of-the-art model in the 2014 ImageNet Challenge. Its main construction contains 13 convolutional layers of various sizes (3x64, 3x128, 3x256, and 3x512) with 5 maxpool stages. The last phase of the convolutional layer is connected to a fully connected neural network with two hidden layers that contain 4096 nodes each. The last layer of the network contains 1000 nodes with a softmax activation function since the ImageNet dataset contains 1000 different layers. These models require 224x224 RGB images as an input. The only difference between VGG16 and VGG19 is that VGG19 contains three additional convolutional layers, which were also trained using ImageNet. To extract embeddings, the last layer of the convolutional network is flattened.

- **Xception** is a convolutional neural network architecture that heavily relies on depth-wise separable convolutional layers. In the construction stage of the architecture, the following hypothesis is used: that the mapping of cross-channel correlations and spatial correlations in the feature maps of convolutional neural networks can be entirely decoupled. The architecture has 36 convolutional layers, which are arranged in 14 modules, all of which have residual connections around them except for the first and last modules. It is sectioned into three stages: entry, middle, and exit flows. The data goes through the entry flow, the middle flow, which is repeated eight times, and finally the exit flow. The pre-trained version of the Xception architecture is used, which was trained using the ImageNet dataset (Chollet (2017)).

- **ImageNet** (Deng, Dong, Socher, Li, Li & Fei-Fei (2009)) is a dataset that contains 50 million cleaned and labelled, full resolution images. It has 12 subtrees. The major label groups inside the dataset are mammal and vehicles.

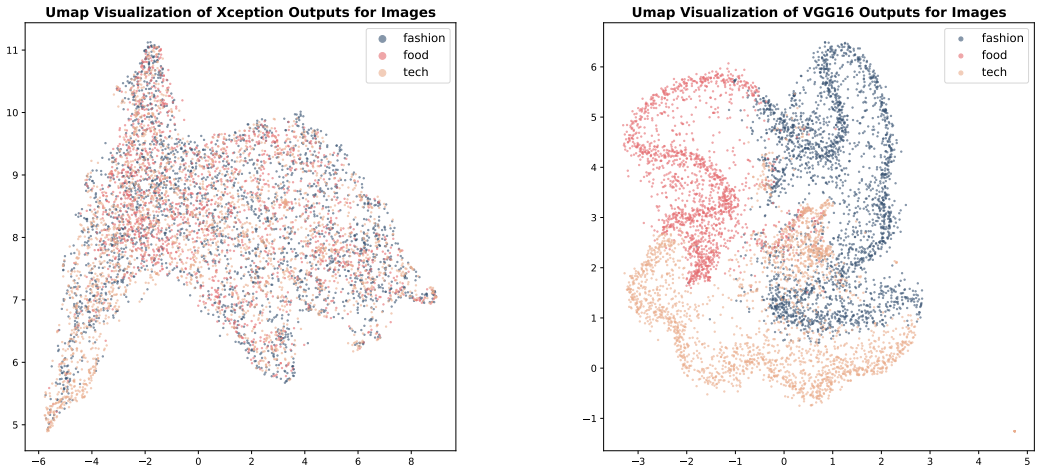
Table 3.2 **F1 score comparison of SVM models trained on image embeddings**: F1 scores of SVM models in different sample size ranges, trained by using the embeddings of VGG16, Xception, and VGG19. Xception provides a higher score, while VGG16 embeddings outperform others on the complete dataset.

Sample Size	VGG16	Xception	VGG19	Combined
300	0.683	**0.767**	0.731	0.721
2400	0.738	**0.760**	0.752	0.736
6408	**0.751**	0.749	0.732	0.740

Recall that a dataset was scraped from Instagram that contained influencers, and the content was known by using heuristic 3. This dataset contains influencers from the tech, food, and fashion categories. Since the content of each influencer is known in this dataset, to evaluate the quality of image embeddings extracted from pre-trained models, a model can be trained by using these embeddings and labels in a supervised manner. Although the count distribution of each category is almost the same (198, 195, 156), the F1 score is chosen as an evaluation metric for image embeddings. Table 3.2 shows the results of the experiment. As can be seen, Xception embeddings give the best results on smaller samples. On the other hand, VGG16 embeddings outperform others with high sample sizes. We have also concatenated the embeddings of VGG16, Xception and VGG19 embeddings and checked the performance of common representation. However, common representation shows a lack of performance when it is compared with VGG16. This could be observed by looking at the distribution of embeddings in 2D space as well. If distinguishable clusters can be observed by looking at the visual, it is fair to say that the pre-trained model works as expected on Instagram data.

There are various types of dimensionality reduction algorithms such as Principle Component Analysis, Singular Value Decomposition etc. In this research, the UMAP (Uniform Manifold Approximation and Projection) dimensionality reduction technique is used. It is based on Riemannian geometry and algebraic topology (McInnes, Healy & Melville (2018)) which provides great visualization quality with a low cost and short run time. Additionally, UMAP can be used as a general-purpose dimension reduction strategy for machine learning because it has no computational limits on embedding dimensions. UMAP creates a topological representation of the high-dimensional data by patching together its local fuzzy simplicial set representations and local manifold approximations.

Figure 3.5 shows the comparison of embedding distributions taken from Xception and VGG16. The main concern with these visualizations is to observe distinguishable clusters regarding the classes. As it can be seen from figure 3.5, figure 3.5b



(a) Embedding Distributions of Xception (b) Embedding Distributions of VGG16

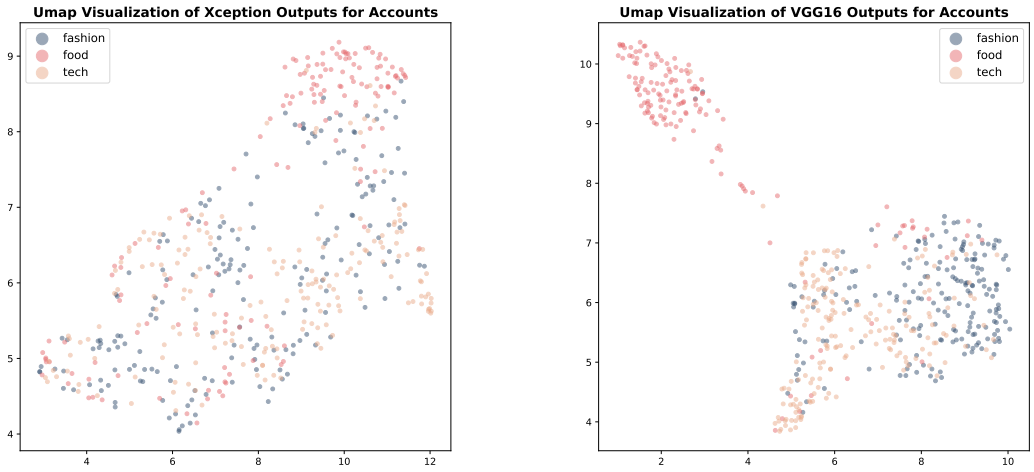
Figure 3.5 **UMAP Visualizations of VGG16 and Xception Embeddings** 3.5a shows distribution of Xception embeddings and 3.5b shows the distribution of VGG16 embeddings. Embeddings that was taken from 3.5b presents more distinguishable clusters compared to 3.5a.

presents highly distinguishable clusters compared to figure 3.5a.

The same comparison can also be applied to account-based embeddings. The evaluation accomplished so far was made by using the image embeddings, which means each image is represented by one embedding. However, influencers post more than one image most of the time. To represent an influencer with a vector embedding, the mean of each image embedding, which are the images posted by the influencer, is taken. For example, given an influencer with 10 posts, 10 image embeddings are extracted by using a pre-trained model. The mean of the 10 image embeddings is used to represent the influencer.

Although VGG16 outperforms Xception in image embeddings, when we compare the account-based cluster occurrences, both models performed similarly when Figure 3.6 is analyzed.

Since both models perform almost the same in terms of F1 scores, the UMAP visuals are taken into account and it is stated that VGG16 clustered better than Xception. Because of this, VGG16 was chosen as model to be used to extract image embeddings from posts.



(a) Account Distributions of Xception (b) Account Distributions of VGG16

Figure 3.6 **UMAP Visualizations of Accounts Based on Embeddings of Xception and VGG16** Figure 3.6a shows the distribution of accounts using Xception embeddings, and figure 3.6b shows the distribution of accounts using VGG16 embeddings. As it can be seen, there is no big difference from the perspective of account representation.

3.3.2 Validation text based models

As with images, text also plays a crucial role in content comprehension. Because of this, text embeddings are also needed to identify influencers, which can be achieved by using pre-trained models. The pre-trained models chosen for this task are Sentence-BERT, MiniLM-L6, and NLI. These models take a text as an input and output a vectoral representation of the text, which is called text embedding. These text embeddings can later be used for understanding content since texts store information regarding content same as images. Because of this, texts are crucial for understanding influencers' content. The embeddings taken from these models were used as features to represent texts. Similar to image embeddings, text embeddings are later fed to the SVM classifier, which uses the RBF kernel. To comprehend high-quality text embedding, the F1 scores of each model are compared.

The process of extracting text embeddings is different than extracting image embeddings in terms of the architecture of the model that is being used. Convolutional neural network architecture is not used in the text context, but rather a new mechanism called Transformers (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser & Polosukhin (2017)), which was evaluated from recurrence neural network architecture. Unlike, LSTM (Hochreiter & Schmidhuber (1997)), and convolutional

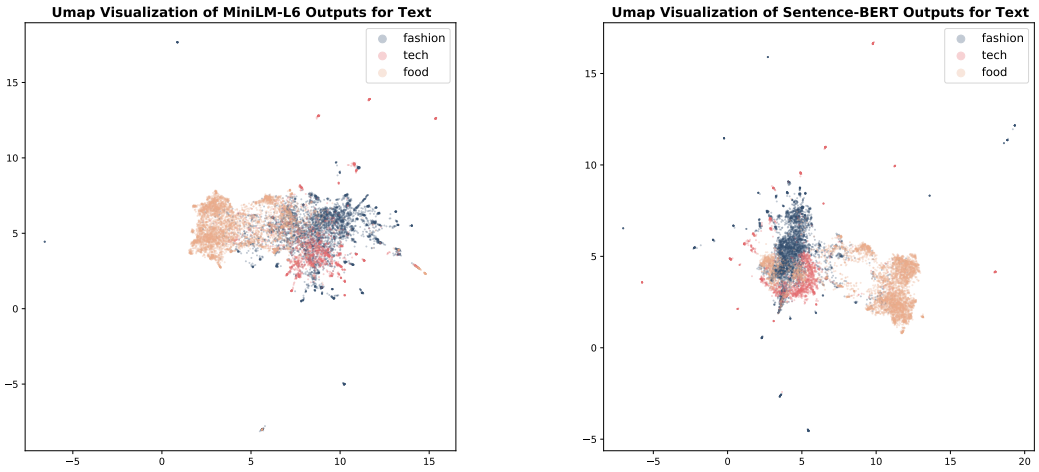
neural networks (LeCun, Bengio & Hinton (2015)), Transformers completely rely on attention mechanism to draw global dependencies between input and output. The attention mechanism can be described as a function that maps a query and a set of key-value pairs to an output. That is to say, the output is the weighted sum of the values, whose weights are computed by the compatibility function of the query with the corresponding key. This mechanism has been used in several places in the architecture. The architecture of the Transformers is composed of two stages, which are the encoder and decoder. The encoder is the part where input text is fed. It has an attention mechanism with position-wise fully connected feed-forward network. The decoder part contains 3 sub-layers, 2 of which are attention mechanisms, and the third is position-wise fully connected feed-forward network. The decoder is fed an input text pair. We will use the output of the position-wise fully connected feed-forward network located at the end of the encoder layer because we are only dealing with the vector representation of any text. All the models that are being used in this project are composed using this architecture, and the only differences are the datasets that are being fine-tuned and the feed-forward network node sizes.

Table 3.3 **F1 score comparison of SVM models trained on text embeddings**: F1 scores' of each SVM model on different sample sizes. Outcomes of pre-trained models were used as an input to SVM. Embeddings taken from MiniLM-L6 led the best overall performance against Sentence-BERT and NLI.

Sample Size	MiniLM-L6	Sentence-BERT	NLI	Combined
300	0.933	**0.967**	0.883	0.913
2400	**0.906**	0.892	0.885	0.889
6408	**0.871**	0.855	0.867	0.848

To evaluate the performance of text embeddings, the dataset created by using heuristic 3 is shared, which includes food, tech, and fashion content, as with image embedding experiments. To evaluate the quality of text embeddings extracted from pre-trained models, a SVM model can be trained by using these embeddings and labels in a supervised manner. Table 3.3 shows the results of the experiment. As can be seen, MiniLM-L6 embeddings give the best results on large samples. Similar to image embeddings, we concatenated MiniLM-L6, Sentence-BERT and NLI embeddings to create a common representation. However, the results show that it did not perform well when compared with the others. On the other hand, Sentence-BERT embeddings outperform others in small sample sizes. This could be observed by looking at the distribution of embeddings in 2D space as well. If distinguishable clusters can be observed by looking at the visual, it is fair to say that the pre-trained model works as expected on Instagram data.

Figure 3.7 shows the comparison of embedding distributions that were taken from



(a) Embeddings of MiniLM-L6 (b) Embeddings of Sentence-BERT

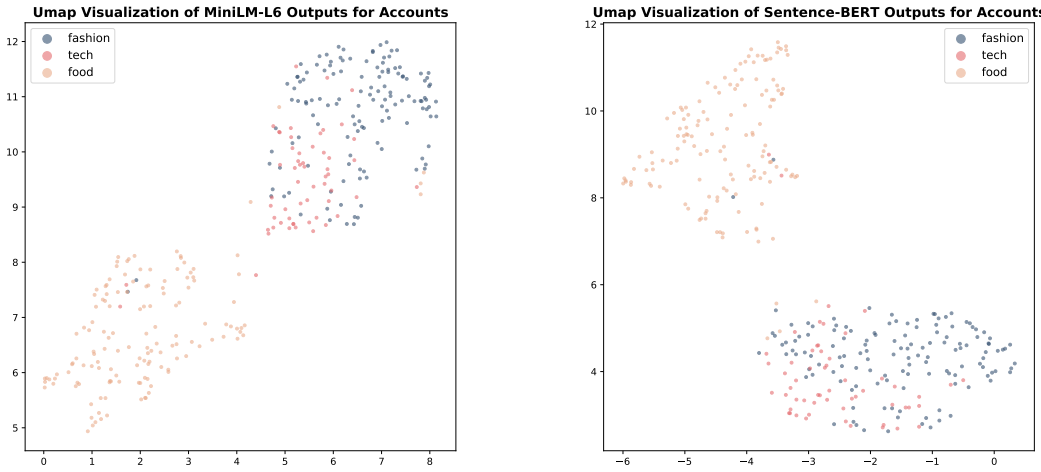
Figure 3.7 **UMAP Visualizations of MiniLM-L6 and Sentence-BERT Embeddings** 3.7a shows the distribution of MiniLM-L6 embeddings, and 3.7b shows the distribution of Sentence-BERT embeddings. Embeddings that was taken from 3.7a presents similar clusters compared to 3.7b.

MiniLM-L6 and Sentence-BERT. The main concern with these visualizations is to observe distinguishable clusters regarding the classes. As it can be seen from figure 3.7, figure 3.7a shares similar cluster distribution with figure 3.7b.

Recall that a fulfilled experiment emerged by using text embeddings of influencers. However, the distribution of influencers along UMAP using text content is still unknown. To accomplish this, the mean of each text embedding with a connection to the influencer is calculated and used to represent the influencer, just as it is in image embeddings. For example, given an influencer with 10 posts, 10 text embeddings are extracted by using a pre-trained model. The mean of the created 10 text embedding is used to represent the influencer.

Although MiniLM-L6 outperforms Sentence-BERT in text embeddings when table 3.3 is taken into account, the influencer-based cluster occurrences in figure 3.8 show no difference.

At the end of the experiment, MiniLM-L6 was chosen as the model to be used to extract caption and comment embeddings from posts. These embeddings would then be used as a feature of a recommendation system.



(a) Account Distributions of MiniLM-L6 (b) Account Distributions of BERT

Figure 3.8 **UMAP Visualizations of Accounts Based on embeddings of MiniLM-L6 and Sentence-BERT** Figure 3.8a shows the distribution of accounts using MiniLM-L6 embeddings and figure 3.8b shows the distribution of accounts using Sentence-BERT embeddings. As it can be seen, there is no big difference from the perspective of account representation.

3.4 Implementation

3.4.1 Feature Engineering

In this section, we will be talking about what features and how they were created that will later be used in the recommendation system.

After the scraping operation had finished, the features that are used in the recommendation system were created. These features are categorized into four different sections: image, text, network, and metadata features.

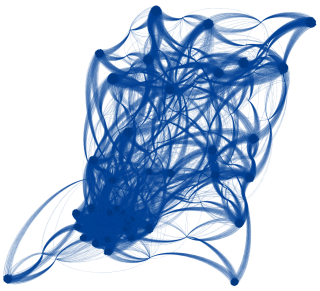
Image features are all features related to images or videos. At the moment, we only consider the first frame of the video as a thumbnail image, but this approach can be expanded with more computational resources. Recall that in section 3.3.1, we mentioned that a model trained on the ImageNet dataset with VGG16 structure is used to extract image embeddings. The convolutional layers of this model can be used to map images to their corresponding representations in multi-dimensional

space. That is to say, the last convolutional layer of the pre-trained VGG16 model is used to get vectoral representations of images, which gives us a kernel with 7x7x512 dimensions. These kernels are later reshaped to 1x25088. After extracting the embeddings of images, it could be possible to calculate statistical metrics regarding the influencers. The images with the same content would have similar weights in their extracted embeddings, allowing us to calculate a score for each influencer based on how consistently they share the same content by averaging the pairwise cosine similarity scores of each post. The cosine similarity is a metric that can be used to calculate the similarity between two sequences of numbers. In our case, we will be pairing every post that the influencer posted and calculating a cosine similarity score for every pairwise comparison. For example, if an influencer shared 3 different posts, 4 different cosine similarity scores would be calculated. If the calculated score is close to 1, compared images share similar content. With the help of this approach, if we want to know if the influencer keeps sharing similar content on its page most of the time, we can look at the mean and variance of the calculated cosine-similarity scores. If the calculated mean is closer to 1, then it can be understood that the influencer keeps sharing similar content most of the time. Variance may also give insight about the frequency of sharing the same content. So, at the end, there are $25088 + 2 = 25090$ image-related features that can be used in the recommendation system.

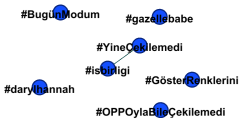
Text features are all features related to captions and comments. Recall that in section 3.3.2, we mentioned that a model trained on more than 1 billion training pairs is used to extract text embeddings and is called MiniLM-L6. The last position-wise fully connected feed-forward network can be used to map texts to their corresponding representations in multi-dimensional space. That is, the final hidden layer of the pre-trained MiniLM-L6 model is used to obtain vectoral representations of texts, yielding a dense vector with a 1×768 dimension. After extracting embeddings of text, it could be possible to calculate statistical metrics regarding the influencers. The texts that have the same content would share similar vectors in multidimensional space, which allows us to calculate a score for each influencer based on how consistently they share the same content by looking at the mean of the pairwise cosine similarity scores of each post. In our case, we will be pairing every post's caption with the one that the influencer posted and calculating a cosine similarity score for every pairwise comparison. If the calculated score is closer to 1, compared captions share similar content. With the help of this approach, if we want to know if the influencer keeps sharing similar content on its page most of the time, we can look at the mean and variance of the calculated pairwise cosine similarity scores. If the calculated mean is closer to 1, then it can be understood that the influencer keeps sharing similar

content most of the time. Variance may also give insight about the frequency of sharing the same content. In addition to this, this approach can also be implemented for comment-based features since they are also text. By generating a score for each post and taking the mean of the generated scores, we can determine whether the comments posted on the page share inner similarity or not. The generated score is the average cosine similarity score of pairwise comment comparisons in the post. At the end, there are $768 + 3 = 771$ text-related features that can be used in the recommendation system.

Network features are all features related to tags, commenters, and hashtags. By looking at these 3 types, we can create networks for each influencer, as nodes will be chosen based on type (commenter, hashtag, or tagged account), and the edge condition is whether two nodes showed up in the same post or not. If two nodes appeared in multiple posts, the weight of the edge would be higher than the rest. Figure 3.9 shows an example of how created networks are shown in each network type. It is observed that commenter networks are generally bigger networks than hashtag and tag networks.



(a) Commenter Network



(b) Hashtag Network

(c) Tag Network

Figure 3.9 **Network Visualization** Figure 3.9a shows the network created by using commenters and figure 3.9b shows the network created by using hashtags and figure 3.9c shows the network created by using tags.

By using the networks in figure 3.9, we can create features. The created features related with the network are shown in table

Table 3.4 shows the extracted features for each network type. However, some influencers may not have these networks. If an influencer does not have a network of the corresponding type, the features mentioned in the table 3.4 will be assigned a value of 0. Finally, we will have $11 \times 3 = 33$ in network-related features that can be used in a recommendation system.

Metadata features are all features that is related with the quantified statistics such as number of likes, comments, followers, followings, posts. At the end, we will have

Table 3.4 **Network Feature Explanations**: Features extracted from commenter, hashtag and tag networks.

Network Feature Descriptions	
Feature Name	**Description**
Number of nodes	Total number of nodes in the network
Number of edges	Total number of edges in the network (weight are not taken into account)
Total weight	Summation of weights of the edges
Density	$d = \frac{2m}{n(n-1)}$, where n is number of nodes and m is the number of edges
Average clustering coefficient	$C = \frac{1}{n}\sum_{v \epsilon G} c_v$ where n is the number of nodes in network
Radius	Returns the minimum eccentricity
Maximum clique size	Returns $O(\|V\|/(log\|V\|)^2)$ apx of maximum clique/independent set in the worst case.
Number of connected components	Returns the number of connected components.
Fraction of the largest connected component	Returns the highest number of connected components cluster over number of nodes.
Maximum core number	Returns the core number in the biggest cluster in the network.
Number of nodes that has maximum core number	Returns the number of nodes that has highest core number

a 5 metadata related features that can be used in recommendation system.

From the system perspective, each influencer now can be described by using the extracted image, text, network, and metadata features. So, each influencer is described in a $2590 + 771 + 33 + 5 = 3399$ dimensional space.

We wanted to investigate correlations between each data type to get a better understanding of which data types have a high influence on the identity of the account. The assumption was to observe a correlation between image and text data since they share knowledge in terms of the influencer's content. To prove this hypothesis, we created the figure 3.10 and used each data type to generate a 8613990 cosine similarity score between pairwise influencers individually and created scatter plots to show the correlations. We observed that there is a linear correlation between image and text data. For other comparisons, such as image versus network and text versus network, it appears that these comparisons do not share a common link within each other.

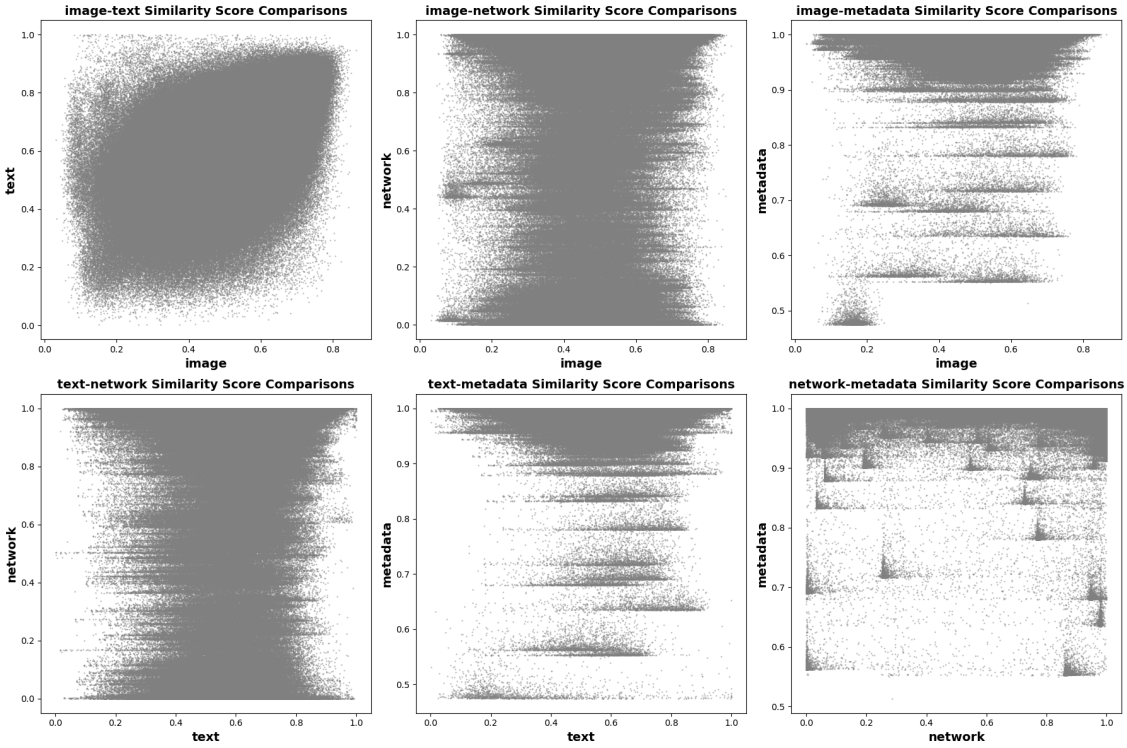


Figure 3.10 **Similarity comparison by data type** *Image vs Text* type shows correlation while *Image-Network* and *Text-Network* does not share any link.

3.4.2 Recommendation System

In this section, we will be talking about the structure of the system and the approaches we used to achieve a high engagement rate with users. The system is composed of 2 independent sections. One section is to be used to come up with similar accounts for a target account. The other section is to come up with accounts that have high engagement rates by using the capture-recapture score.

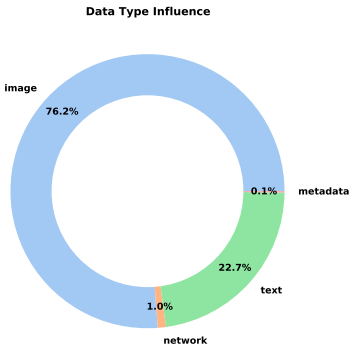


Figure 3.11 **Piechart distribution of data type ratios** Image type data has influence on 76.2% of the dimension while metadata type data has only 0.1%.

Recall that an influencer is described in 3399-dimensional space by using image, text, network, and metadata features, and figure 3.11 shows percentages of each data type along dimension. Image type data influences 2590 of them, accounting for 76.2% of the total size. This means that if we generate a cosine similarity score directly using the flat 3399 dimension, the resulting score will be heavily influenced by image type data, greatly reducing the effect of metadata or network. Since we also want to take them into account, we do not directly use combined 3399-dimensional representation but separate them using data types and calculate a score for each piece individually. Equation 3.2 shows how a similarity score is calculated.

(3.2)

$$
\begin{aligned}
similarityscore_{(inf1,inf2)} = \; & 0.7 \times cosineSimilarity_{(inf1,inf2)}(\left[Image\right]_{2590}, \left[Image\right]_{2590}) + \\
& 0.2 \times cosineSimilarity_{(inf1,inf2)}(\left[Text\right]_{771}, \left[Text\right]_{771}) + \\
& 0.05 \times cosineSimilarity_{(inf1,inf2)}(\left[Network\right]_{33}, \left[Network\right]_{33}) + \\
& 0.05 \times cosineSimilarity_{(inf1,inf2)}(\left[Metadata\right]_{5}, \left[Metadata\right]_{5})
\end{aligned}
$$

We have given different weights for different data types since we wanted some data types to have more control over similarity. According to figure 3.10, image and text type data contain more content-based information, which gives them a higher importance in the generation of similarity scores. However, network and metadata type data may still contain information regarding similarity, so we also took them into account with smaller weights.

Other sections of the system are used to find accounts that have high engagements. To rank influencers with high engagement, a score is generated for each influencer using the equation 3.1. After engagement rate scores are generated for each influencer, the correlation with follower numbers is checked.

Figure 3.12 shows that there is a high correlation between the capture-recapture score and follower number with *Pearson Correlation* 0.6159. This regression line can be used to estimate accounts that have a high engagement rate. If the capture-recapture score is higher than the expected score, which is calculated by using the regression line, then we may call these as high engagement rate accounts. Figure 3.13a shows the histogram of capture-recapture scores observed in 3.12 and figure 3.13b shows the histogram of engagement rates that is calculated by extracting the capture-recapture score from the corresponding expected value, which is observed by applying a linear regression line to the dataset. Histograms are separated by influencer type. As a result, the behaviors of two metrics between influencer types would be obvious. As it can be seen in figure 3.13a, the histograms for different

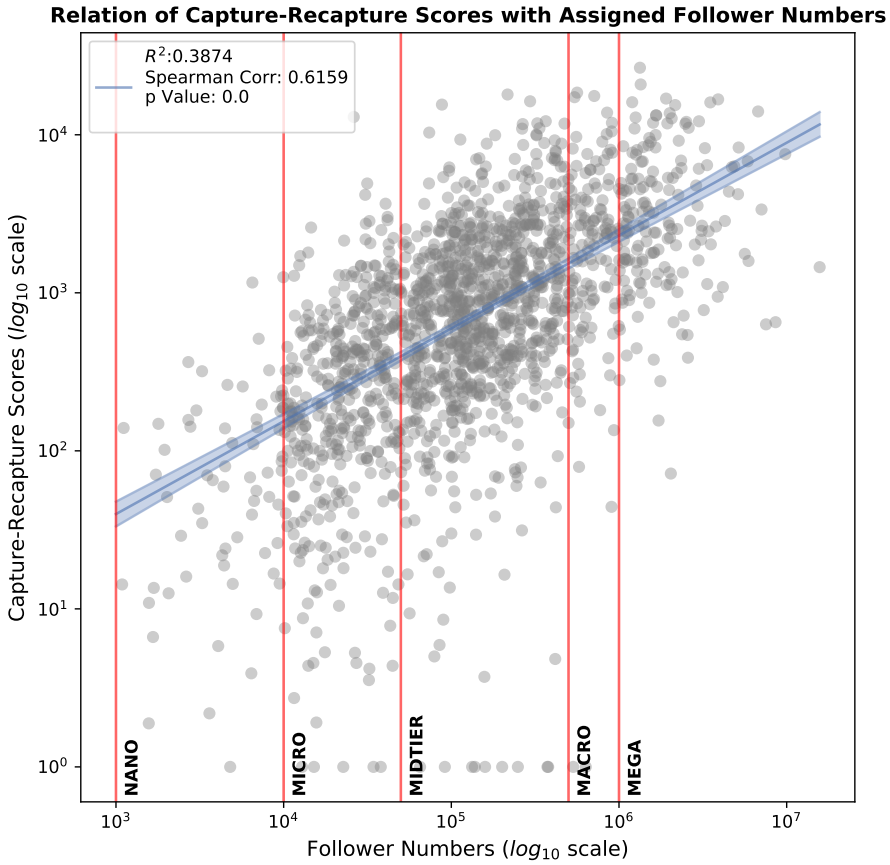
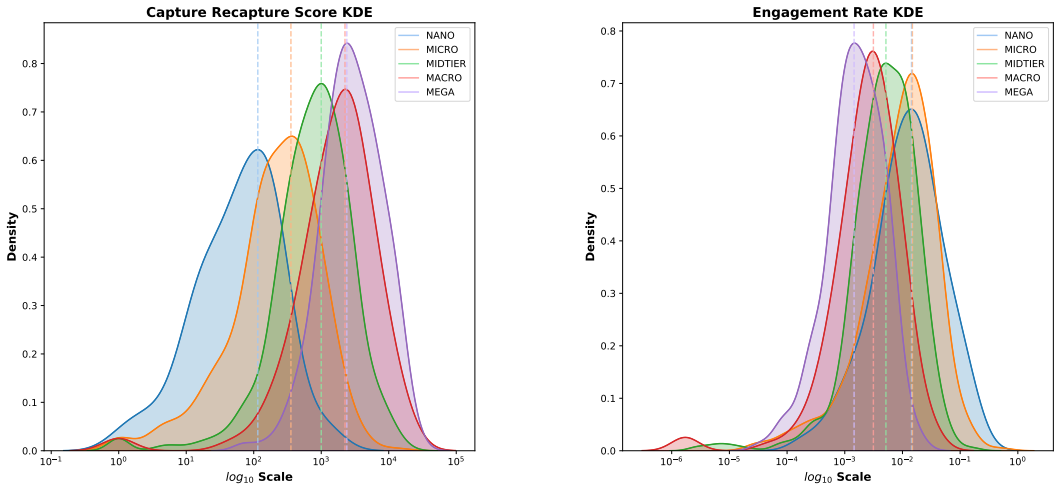


Figure 3.12 **Follower Number vs CR Scores** Image type data has influence on 76.2% of the dimension while metadata type data has only 0.1%.

influencer types do not perfectly overlap one another, which shows the effect of follower number on the capture-recapture score. However, in 3.13b the overlap accuracy among influencer types is higher than how the capture-recapture score fits. This shows that engagement scores are robust to follower numbers, which makes this metric useful to compare the engagements between different influencer types.

By using these two individual modules, we may come up with a system that will recommend an influencer that will share similar content with the target influencer but has a lower engagement rate than the recommended one. To make such a recommendation, a target account needs to be chosen by the user. After the target is chosen, the system finds the 20 most similar accounts in the dataset by using the equation 3.2. After most similar accounts are identified, capture-recapture scores are extracted from the vertically aligned expected value of the regression line, which returns a score about the rank of the engagement regardless of the follower number of the account. This normalization allows us to suggest influencers that have a lower follower number than the target influencer but have higher engagement rates. After scores are calculated, accounts that have a higher engagement rate than the



(a) KDE of Capture-Recapture Scores per Influencer Type

(b) KDE of Engagement Rate per Influencer Type

Figure 3.13 **Capture Recapture Score vs Engagement Rate** 3.13a shows the density of capture recapture scores per influencer type, and 3.13b shows the density of engagement rates per influencer type. Usage of engagement rate provides a better metric to use than capture recapture scores since generated rate does not distinct per influencer type

target account are suggested. The order of the suggestion is adjusted as the follower number will be in ascending order.

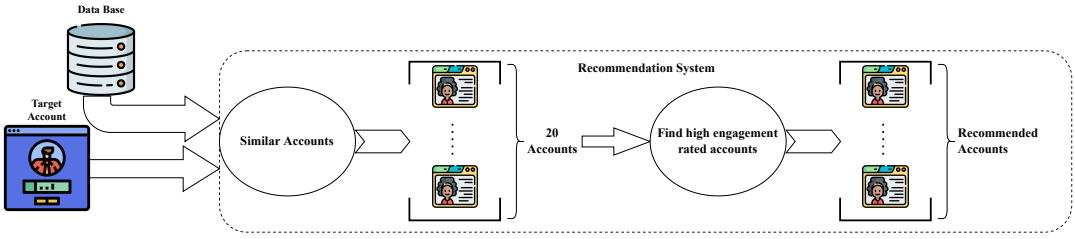


Figure 3.14 **Recommendation System Illustration**

3.5 Validation of Recommendation System

Although the system has been created, it is still unknown how good its performance is. Does the new capture-recapture score outperform metrics that are already in use? The system's evaluation will be discussed in this section, and the results will be compared to existing metrics.

In the evaluation of the system, we used 4 different coders who are undergraduate students that frequently use the social media platform Instagram. We created a subset of influencers that contains 10 of each influencer type for the evaluation of the system since perusing 2022 influencers would be time-consuming. After that, a survey is created that contains questions regarding the similarity and engagement rate of given influencers within a subset. We took the responses to these surveys as ground truth and evaluated the system based on these responses.

3.5.1 Validation Dataset Collection

Recall that the total number of Turkish influencers scraped from Instagram was 2022. The histogram of Turkish influencers by follower number is seen in figure 3.15. The majority of the scraped influencers are of the mid-tier variety, with only a few belonging to the macro variety. This results in a bias in the evaluation of the system, and using 2022 influencer would be time-consuming since each coder should make 2,043,231 comparisons. Since this is quite a large number to be considered, a smaller subset is created for coders to evaluate.

For evaluation, 10 influencers from each type (nano, micro, mid-tier, macro, and mega) were chosen randomly, which creates a subset with 50 influencers. After influencers have been determined, comparison pairs must be created. Each influencer that belongs to the same influencer type is compared individually. Since there are 5 different influencer types and each influencer type contains 10 influencers, there will be a $5 \times \frac{10 \times 9}{2} = 225$ comparison for each coder.

After influencers have been chosen for evaluation, the spectrum of engagement rate differences and similarity scores is needed to prove that the range is wide enough to get good results. Figure 3.16 shows the boxplot distributions of engagement rate differences and similarity types. What we expect to see from these figures is a wide

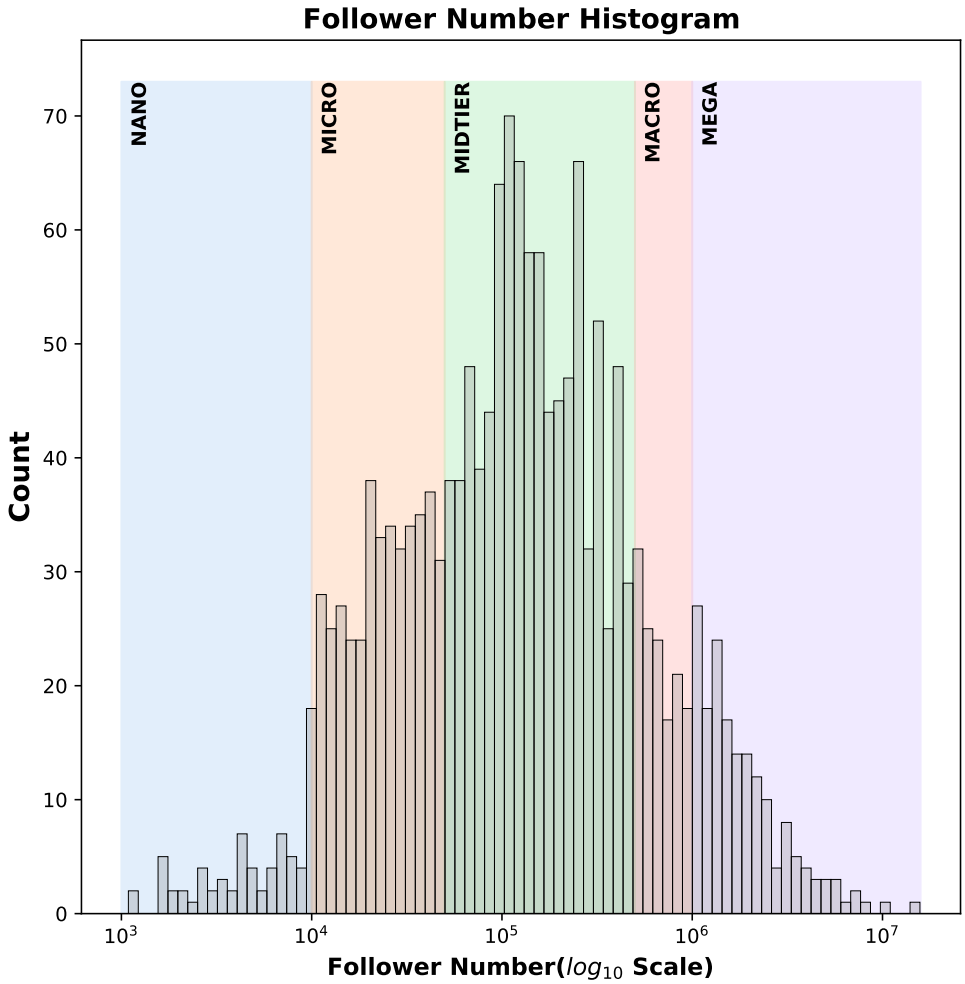


Figure 3.15 **Follower Number Histogram**: Histogram of follower numbers separated by influencer type.

enough spectrum. So that, for each influencer type, there will be samples that are way too similar to each other and samples that are too different. If two influencers have similar scores in terms of engagement or content similarity, then the difference between them is expected to be small. If they are different from each other, then the difference between influencers is expected to be big. If the created subset contains different values in the high spectrum, the comparison that will be made will be more accurate. After the boxplot visualization is made for both engagement rate differences and content similarity differences, it is seen that the spectrum is wide enough to get accurate results at the end of the validation of the recommendation system.

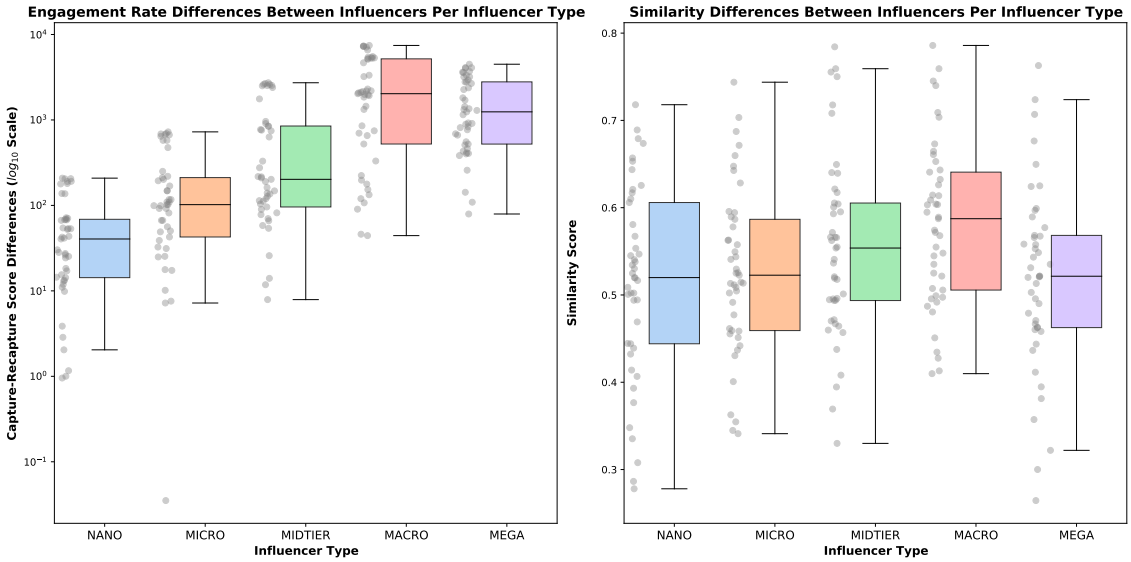


Figure 3.16 **Validation of Created Subsets**: The left-hand side figure shows the distribution of calculated engagement rate differences among chosen influencers. Each influencer type contains 45 samples. The figure on the right shows the calculated similarity scores between the influencers. Each influencer type contains 45 sample.

3.5.2 Survey Creation

After a subset that contains 10 influencers from each influencer type is collected, the survey needs to be created. Each coder's responses are based on the same pairwise comparisons, allowing for control over annotator agreements. Beside reciprocal comparisons of annotators, reciprocal comparisons of influencer types are also considered. Because of this, while creating pairwise comparison surveys, influencer pairs from cross-types such as nano-micro, micro-midtier, midtier-macro, and macro-mega are added. From each influencer type, 5 influencers are chosen to match with another 5 influencers from another type, which increases the pairwise comparison amount by $5 \times 5 = 25$ per annotator per cross-type. Since we have 4 cross-type categories, each annotator has to go over $4 \times 25 + 225 = 325$ survey. Recall that all annotators will resolve the same 325 pairs, which allows for performance comparisons between annotators. To increase the quality of the results, the order of the pairs was given differently for each annotator to prevent corrupted results based on long resolve sessions.

A couple of questions were asked to annotators that will shed light on the performance of the system. Since the system contains two different sections, the questions are asked based on understanding the performance of these two sections. After an annotator enters the survey, it is asked to examine the two given influencers by

entering the provided links. After learning about both influencers, the annotator is asked to assign a score based on their similarities. It is a linear scale question ranging from 0 to 10. If 0 is chosen, the compared influencers do not share any common content. If 10 is chosen, the compared influencers are identical. The same approach can also be used to analyze engagement rates among influencers. As a result, two additional questions are asked in order to better understand the engagement rates of the compared influencers. For both of the compared influencers, a linear scale ranging from 0 to 10 is created, as with the similarity question. If the value 0 is selected, the compared influencer does not share engagement. If 10 is chosen, compared influencer shares a high engagement. Each compared influencer is asked about their engagement rate individually. So, in order to understand the system's performance, three discrete questions are asked, one of which is related to similarity and the other two to engagement rate.

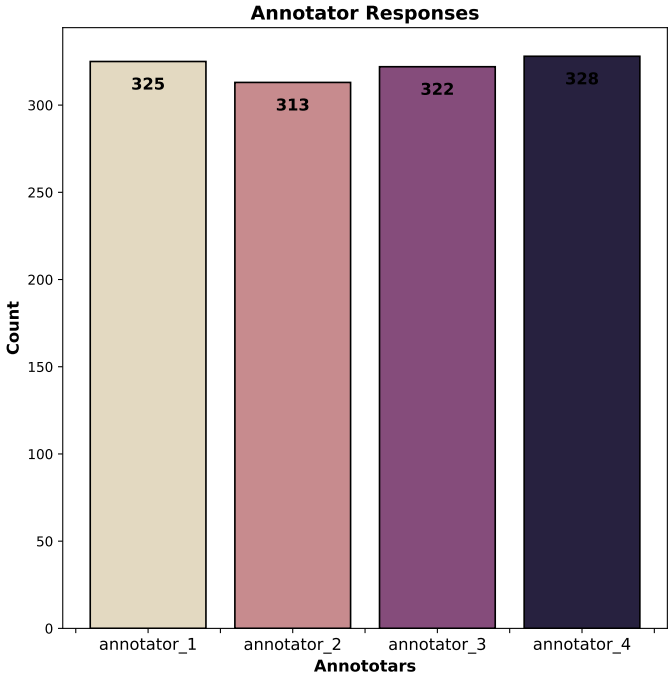
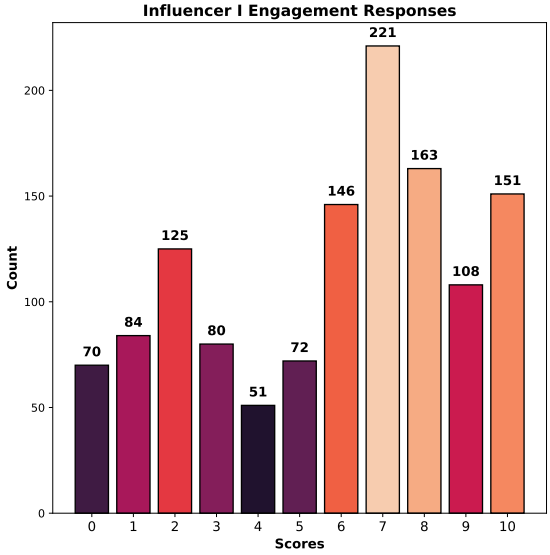


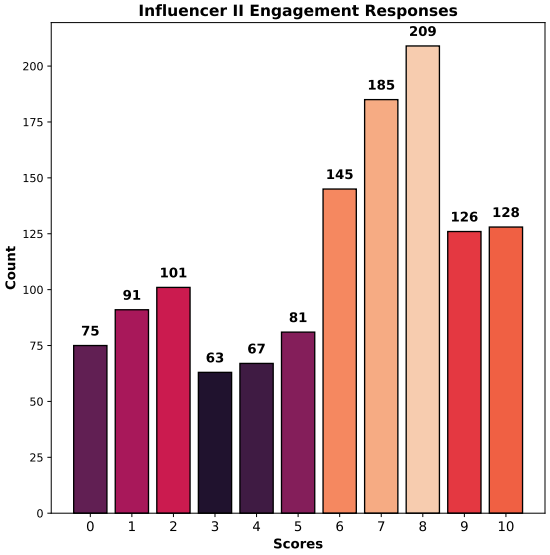
Figure 3.17 **Response Histogram**: Response counts of each annotator for given pairwise influencer comparison surveys.

After the expiration of the response entrance period, figure 3.17 shows the counts for each annotator. Although the total survey number for each influencer is 325, annotator_4 has sent responses 328 times. This shows that annotator_4 has responded to some surveys more than once. As a result, for surveys that had been resolved more than once, the most recent responses were used and the rest were ignored. Besides, it appears that some of the annotators were not able to complete all the surveys in the given time. For the comparisons that could not be resolved by any annotator, the responses of the rest will be taken into account. For example, if a

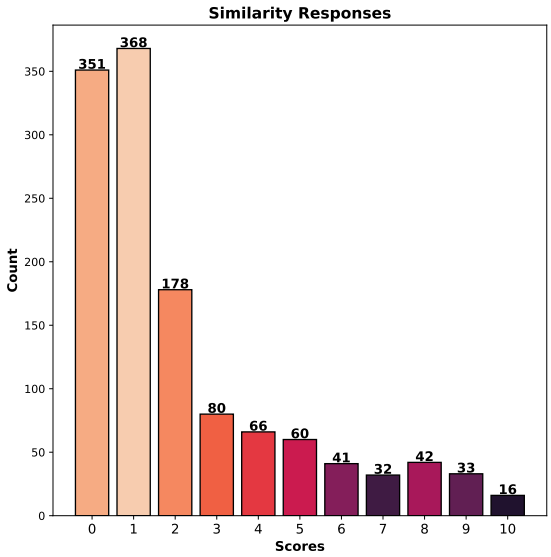
specific comparison is responded by all annotators except annotator_3, the further analysis will be conducted by using the responses of annotator_1, annotator_2, annotator_4.



(a) Engagement Response Histogram for Influencer I



(b) Engagement Response Histogram for Influencer II



(c) Similarity Response Histogram

Figure 3.18 **Survey responses by question**: Figure 3.18a and 3.18b shows the counts of scores responded by each annotator. Most of the responses are covered by 7 and 8 for both histograms. Figure 3.18c shows the counts of scores responded by each annotator. Most of the responses are covered by 0 and 1.

Figure 3.18c displays the total number of scores responded to by each annotator. According to the annotators, the majority of the responses are covered by 0 and 1, indicating that compared influencers do not share the same content in general. Figure 3.18a and 3.18b show the counts of engagement rate responses by score.

Both figures share similar properties since a specific influencer may appear in both sections. The scores of 7 and 8 cover most of the responses, which show that the examined influencers have a reasonable amount of engagement.

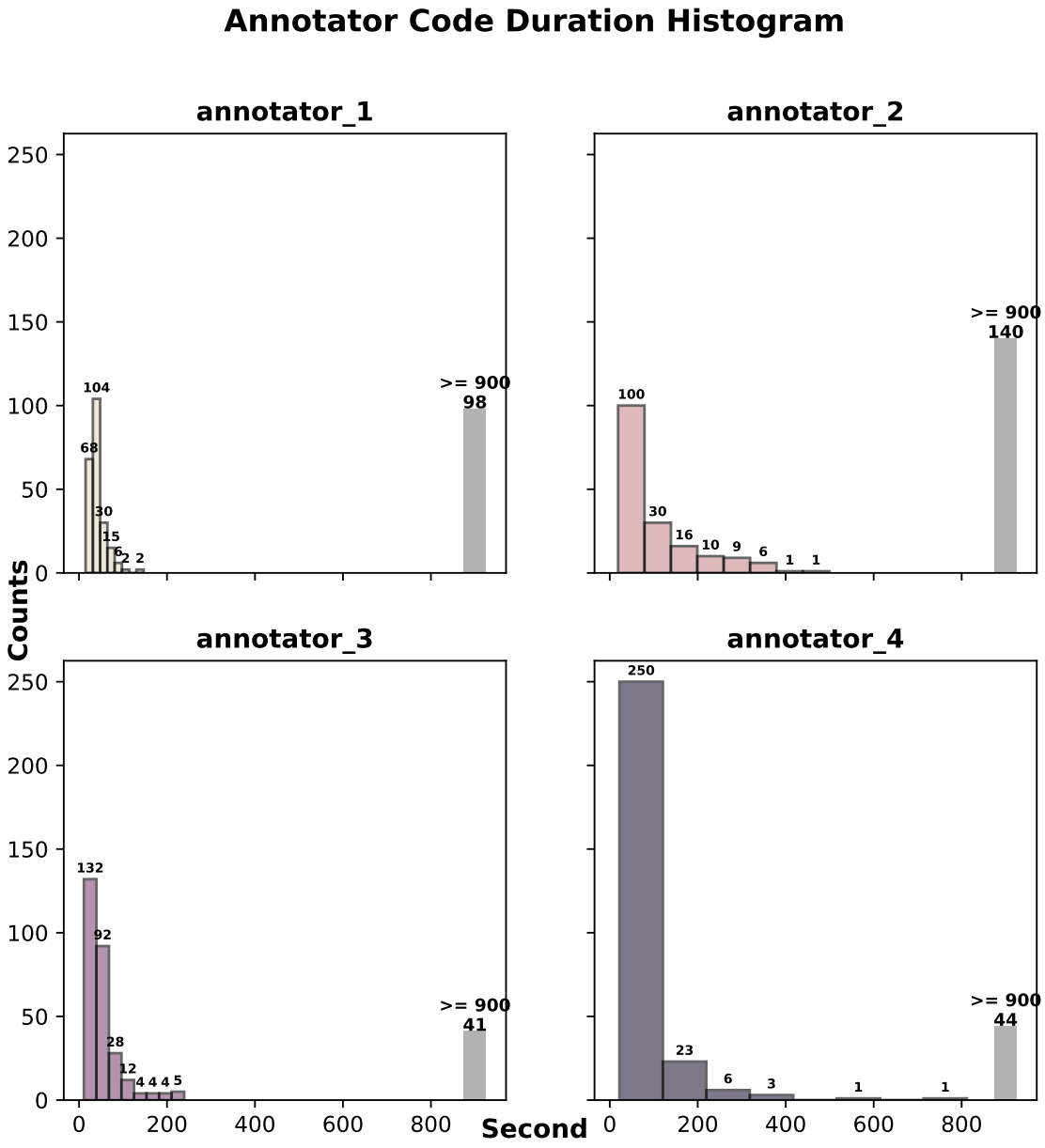


Figure 3.19 **Time Spent on Surveys per Annotator**: Survey response period histogram per annotator

To check the reliability of the responses, the times spent on each survey by annotators were analyzed in figure 3.19. It is believed that the amount of time spent on a survey has a significant impact on the reliability of the responses because determining which influencers share similar content and the rate of engagement takes time. It appears that most of the responses were sent in under 4 minutes. For the responses that take more than 15 minutes, they are assumed to be breaks. Annotators 2 and 4 spent more time than annotators 1 and 3.

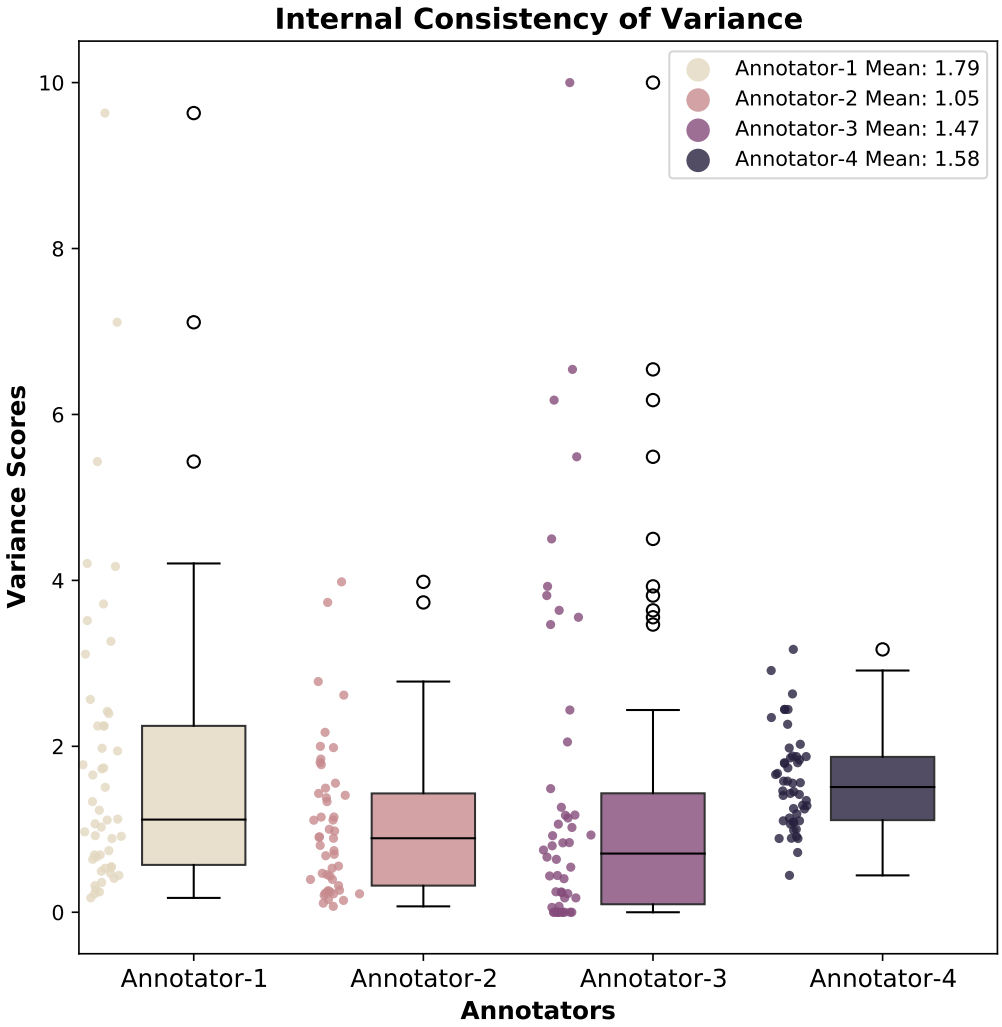
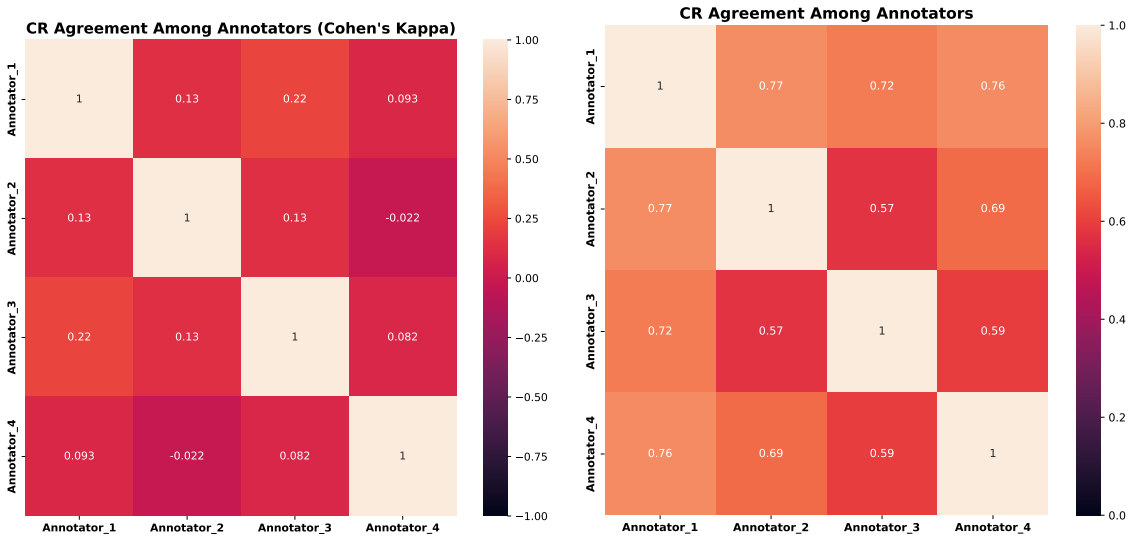


Figure 3.20 **Internal consistency validation by using engagement rates**: Each sample represents the variance of engagement rate responses of an influencer. Each influencer is resolved at least 9 times by an annotator which allows to evaluate consistency by checking the variance of the resolved 9 responses.

Recall that each influencer appears in at least 9 different survey due to the comparison approach. This means that an annotator responds to the engagement rate of the same influencer 9 times. This multiple assignment allows to measure how consistent an annotator is. The expectation from annotators is to see the variance scores for each influencer close to 0. This will show that an annotator does not randomly pick scores from a linear scale but shows consistency. When annotators are compared, figure 3.20 shows that annotator_4 has weak performance since the median score of variances calculated for every influencer is higher than the other annotators. Although outlier numbers are lower than the rest, the main objective is to keep the median score as low as possible. According to this, annotator_3 has the best performance among others.

These engagement scores can also be used to compare annotators reciprocally. One



(a) Heatmap of annotators calculated by using Cohen's kappa.

(b) Heatmap of annotators calculated by using linear scale differences

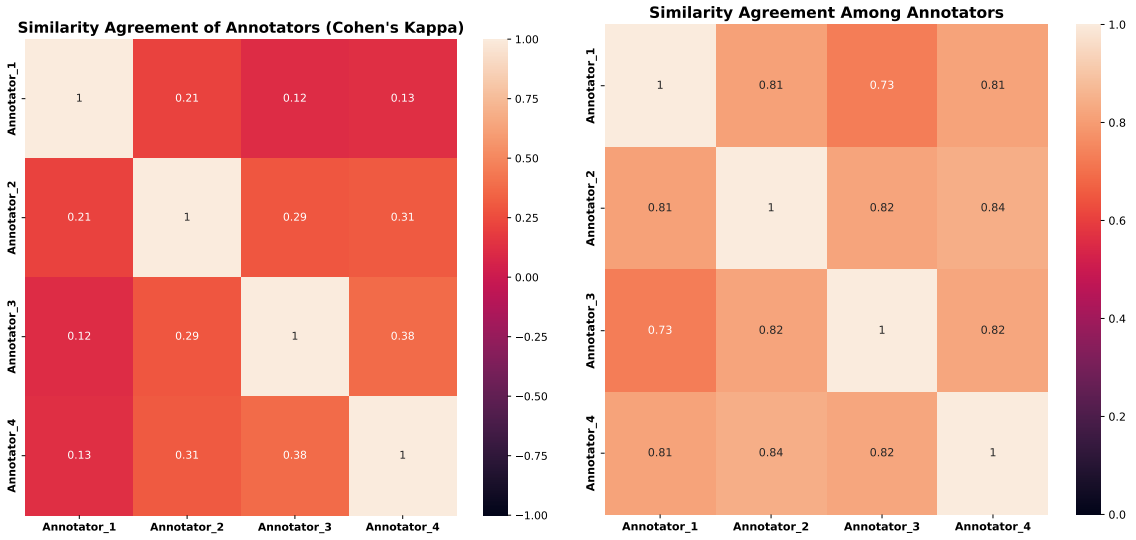
Figure 3.21 **Annotator Similarity Comparison by Engagement Similarity**: Figure 3.21a shows the agreement scores between annotators in terms of engagement rate by using Cohen's Kappa. Figure 3.21b shows the engagement rate similarity agreement between annotators calculated by continuous responses.

way to do this is to take the means of engagement rate responses resolved for each influencer by the annotator and compare the scores between annotators. In the creation of the similarity score for figure 3.21b, equation 3.3 is used, where x and y are vectors that contain engagement rates for a specific influencer, and n is the comparable influencer number. According to this calculation, agreement between annotators appears to be quite high. Another way to compare agreements based on engagement rate is to look at binary agreements, which means that, for a specific pairwise comparison, if both annotators responded with the same influencer as it contains a higher engagement rate, they agreed on this specific comparison. This approach can be applied to any pairwise comparison that is resolved by more than one annotator. The similarity scores observed in figure 3.21a are calculated by using Cohen's kappa in a binary manner. Although both heatmaps contain the same collocation, the scales of the generated similarity scores are different.

$$similarityscore_{x,y} = 1 - \frac{|\overline{x}_1 - \overline{y}_1| + |\overline{x}_2 - \overline{y}_2| + ... + |\overline{x}_n - \overline{y}_n|}{n} \tag{3.3}$$

The same approach can also be followed for similarity agreements. Two different approaches are implemented in figure 3.22 as binary and continuous agreement. The continuous agreement follows the same approach by using the equation 3.3. Binary

agreement is also almost the same for one exception. While deciding on binary agreements, if the annotator responded to a specific comparison by less than 2, then the decision of the annotator is labeled as not similar. The reason to choose this number is because of the count of similarity responses shown in figure 3.18c. Since most of the responses are lower than 2, it would be better to distinguish similar and not similar decisions by 2. Same as the engagement agreement comparison, although both heatmaps contain the same collocation, the scales of the generated similarity scores are different.



(a) Heatmap of annotators calculated by using Cohen's kappa.

(b) Heatmap of annotators calculated by using linear scale differences

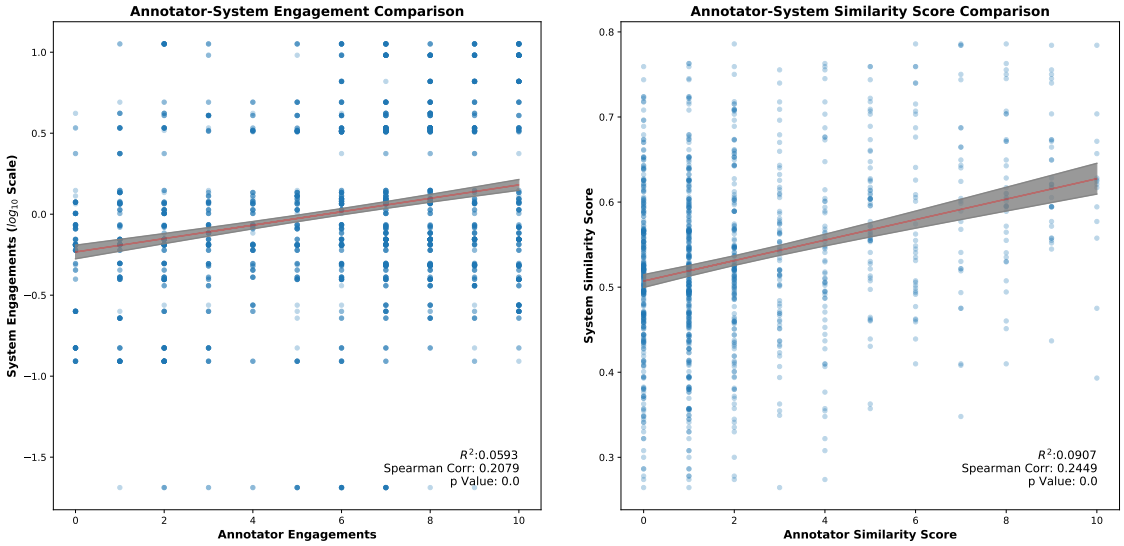
Figure 3.22 **Annotator Similarity Comparison by Content Similarity**: Figure 3.22a shows the agreement scores between annotators in terms of content by using Cohen's Kappa. Figure 3.22b Shows the content similarity agreement between annotators calculated by continuous responses.

After retrieving the responses and evaluating the annotators' performance, the system evaluation substructure is ready for implementation.

3.5.3 Performance Evaluation of Recommendation system

Since the dependability and evaluation of survey responses have been completed, it is safe to test the performance of the recommendation system using the responses. Recall that the recommendation system is composed of two sections as similarity filtering and engagement ranking. By executing the same pairwise comparisons on the system, we would have a chance to compare the results of annotators with the

system outputs. Figure 3.23 contains visuals for the similarity and engagement analogies. To get a better picture, regression lines are fitted to both of the analogies so that it can be better understood whether agreement between the system and annotators is significantly high or not. In figure 3.23a, each sample corresponds to a survey response on the x axis. To see the response of the system to a specific survey response, the y axis contains the system decision made for that sample. Since the responses of annotators were on a linear scale ranging from 0 to 10, samples are aligned vertically on the x axis, while the picture on the y axis is more distributed since these are the decisions of the system. The Spearman correlation between annotators and the system for engagement rate is 0.2079 which shows the rank of the linear correlation.



(a) Engagement Analogy between Annotators and System

(b) Similarity Analogy between Annotators and System

Figure 3.23 **Analogy Comparisons between Annotators and System**: Figure 3.23a shows the agreement level between annotators and the system from the engagement side. Figure 3.23b shows the agreement level between annotators and the system from similarity side.

From the similarity side, figure 3.23b shows the analogy between annotators and the system. Same as figure 3.23a, since decisions of annotators were made using a linear scale ranging from 0 to 10, the samples are vertically aligned with the x axis. The y-axis shows the similarity response of the system for a specific survey response. The Spearman Correlation between annotators and the system is calculated as 0.2449. By looking at these results, it can be said that there is a common ground between the system and annotators.

Looking at raw responses gives an insight about the performance of the system. However, since the aim of the recommendation system is to identify influencers that

have a higher engagement rate than the target influencer, the scale of engagement rate diversity is crucial to analyze. The linear agreement level between annotators and the system in the engagement rate diversity is depicted in figure 3.24. Each sample corresponds to a particular survey response, where the x axis of the response is the difference between influencer 1's engagement rate and influencer 2's engagement rate determined by the annotator, and the y axis of the response is the difference between influencer 1's engagement rate and influencer 2's engagement rate determined by the system. Notice that the difference is calculated by extracting the engagement of influencer 2 from influencer 1. So, if the engagement diversity has a negative value, it can be understood that influencer 2 has higher engagement than influencer 1 and vice versa. According to the figure 3.24, most of the responses entered by the annotators seem to reflect that compared influencers do not distinguish from each other in terms of the engagement rates. Because a significant portion of the calculated engagement diversity was 0. However, still, the level of linear correlation between the system and the annotators ends up with a 0.2271 Spearman correlation score.

Recall that, by looking at the engagement diversity differences, more engaged influencers may be understood easily. This functionality is crucial when suggesting influencers since a more engaged influencer is willing to be recommended. The performance of the system can also be observed by leaving one axis as a binary decision and counting the samples in each group. Figure 3.24 shows the two forms of this comparison, in which figure 3.25a uses the annotators' responses for a binary decision and figure 3.25b uses the system's responses for a binary decision. Although the distributions between two binary decisions look similar to each other, there are slight differences regarding the positions of the most frequent existing continuous scores.

The difference can be seen more easily if an agreement metric is generated. The best approach for this task would be to use the ROC curve, which is the receiving operating characteristics curve. The ROC curve uses true-positive and false-positive rates at different levels to create the curve. The area under this curve would indicate the degree of agreement between the ground truth and predictions. ROC curves make it possible to see the trade-off between sensitivity and specificity at all levels. Since they use a continuous-scale probabilistic distribution, the ROC curve helps to understand the performance of the system in a binary manner. To create the ROC curve, one needs to scale the continuous decisions between 0 and 1 since ROC curves look at probabilistic levels. Because of this, binary decisions will also be labeled as 0 and 1 where 1 means that influencer 1 has a higher engagement rate than influencer 2 and 0 means that influencer 2 has a higher engagement rate than

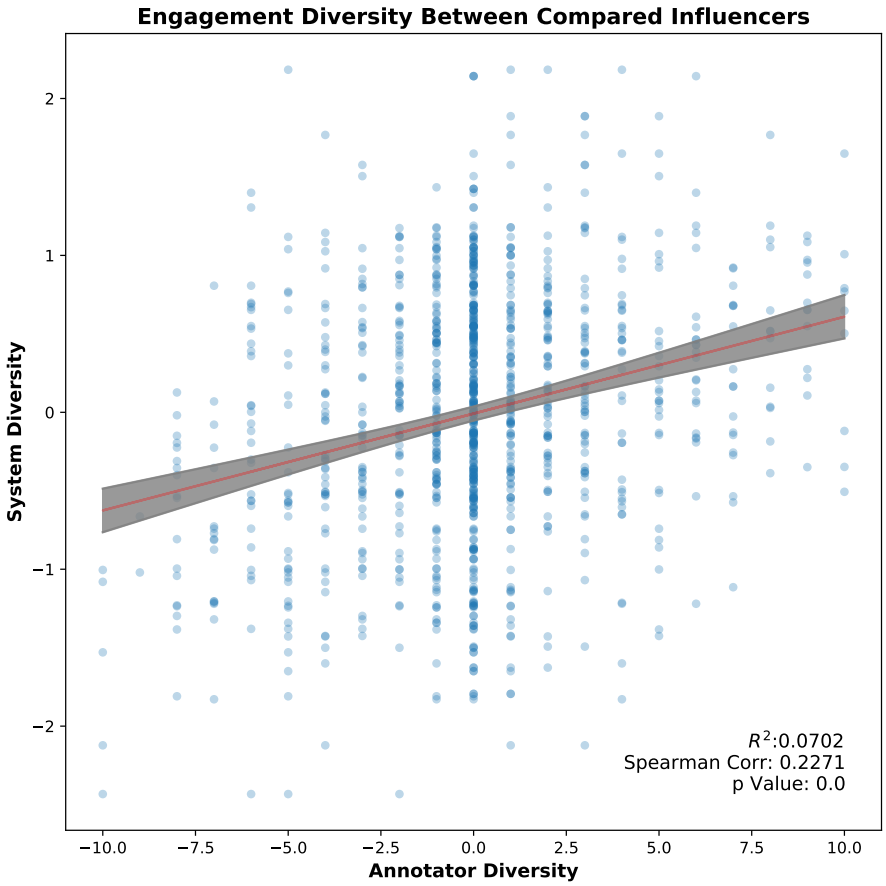
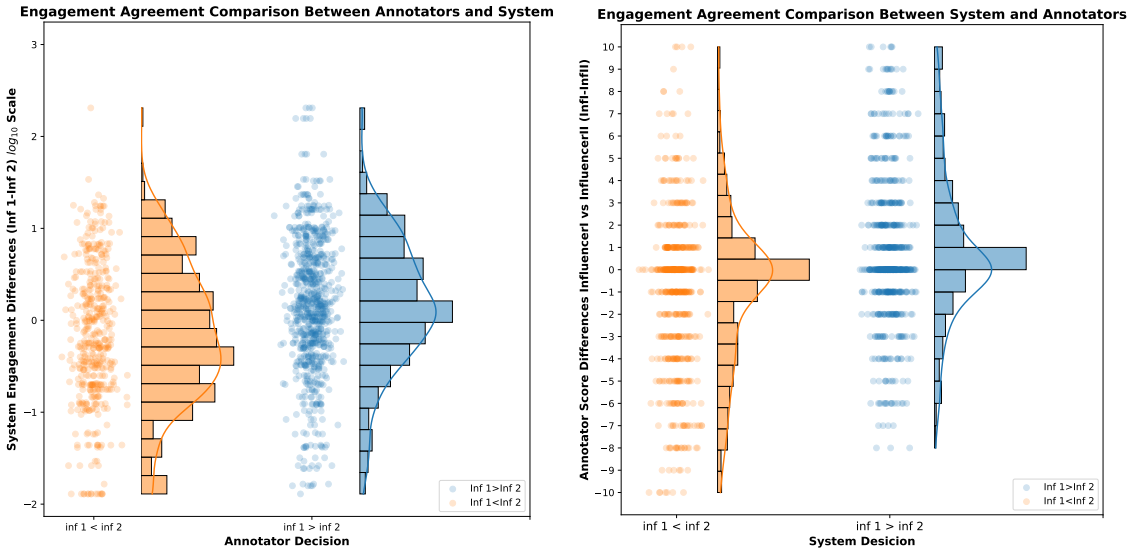


Figure 3.24 **Engagement Diversity**: Understanding engagement differences between compared influencers is crucial since the task is to identify the influencer that has the highest engagement rate. The figure shows the level of correlation in engagement diversity between annotators and the system

influencer 1. After normalization, the data is randomly divided into ten batches, with AUC scores calculated for each batch. The mean of the calculated AUC scores is taken to see the general picture.

This approach is applied to both system ground truth and annotator ground truth cases. Figure 3.26 shows the ROC curves of both cases. When figure 3.26 is compared with figure 3.26, the distinctness of decisions can be seen better with the help of AUC scores. As can be seen, when ground truth is swapped between the system and annotators, the mean AUC scores do not change dramatically. Recall that a random classifier is shown with red dashes whose AUC score is 0.5. If the AUC score is higher than 0.5, then, it can be said that the used model identifies the differences between classes. Since, in both cases, the mean AUC score is higher than the random classifier, it is safe to say that, the created model distinguishes the difference in engagement rates.

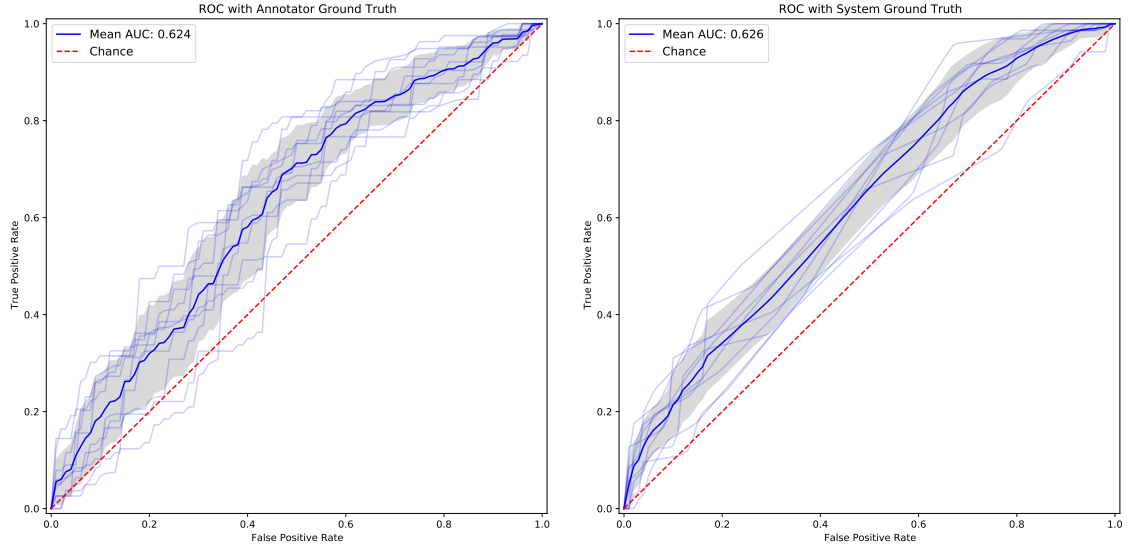
It is also useful to compare the used literature metrics to the introduced engage-



(a) Distribution of system engagement diversities by binary annotator decision groups.

(b) Distribution of annotator engagement diversities by binary system decision groups.

Figure 3.25 **Diversity Binary Comparisons**: Figure 3.25a shows the distribution of the system's engagement diversity scores for binary annotator decisions. Figure 3.25b shows the distribution of system's engagement diversity scores for binary annotator decisions.



(a) ROC with annotator ground truth

(b) ROC with system ground truth

Figure 3.26 **ROC Scores**: Figure 3.26a shows the receiver operating characteristics curve when annotator decisions are taken as ground truth. 3.26b shows the receiver operating characteristics curve when system decisions are taken as ground truth.

ment rate. To observe this, continuous scale decisions made by follower number, post number, following number, like number, and comment number are calculated and compared with the annotators' decisions, as the annotators' decisions will be

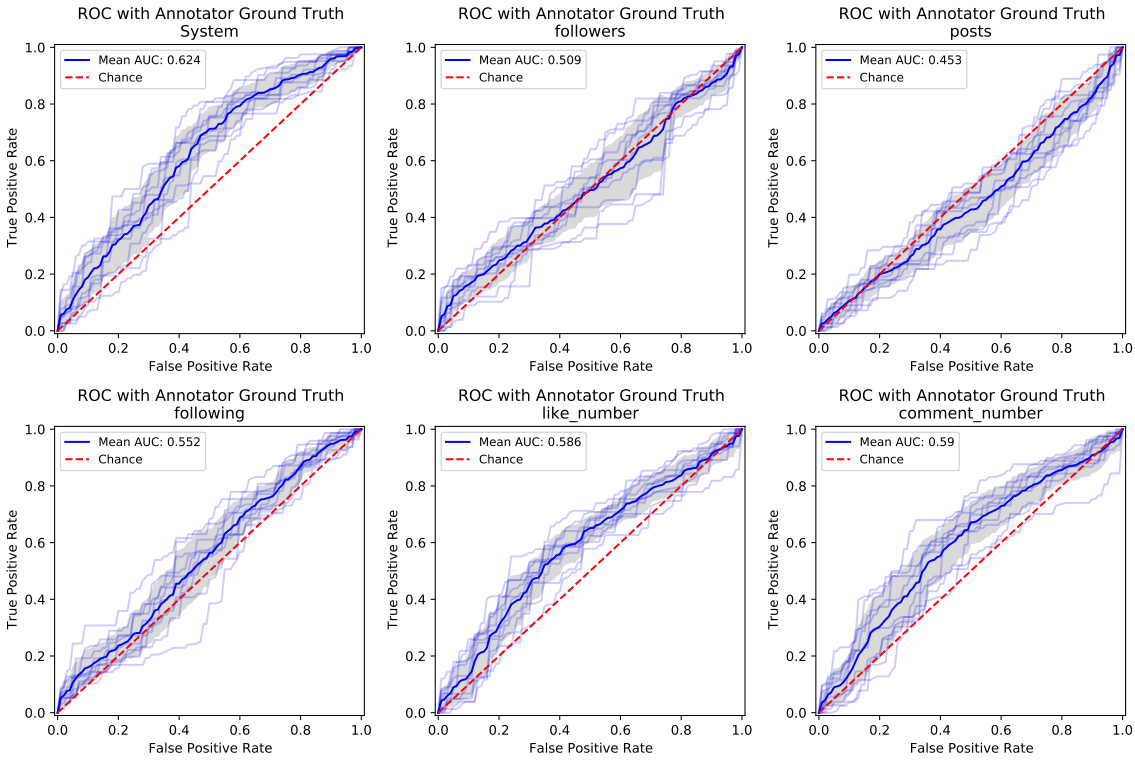
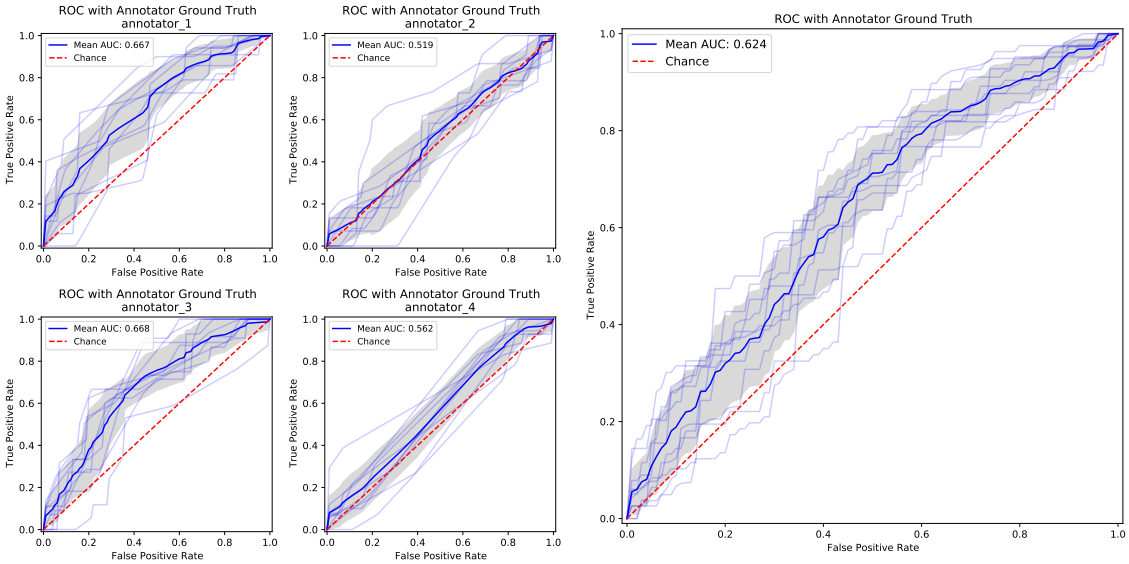


Figure 3.27 **ROC Curves of suggested metrics**: As annotator decisions are taken as ground truth, AUC scores of different literature metrics are compared with the newly introduced engagement rate metric. Introduced engagement rate metric appears to be outperforms others when annotators' decisions are taken as ground truth

the binary ground truth. If the AUC of the introduced engagement rate is greater than the AUC of the literature-proven metrics, it is safe to conclude that the introduced engagement rate outperforms others. Figure 3.27 shows the comparison of ROC curves with other literature metrics. As it can be seen, when the introduced engagement rate metric is used, agreement with annotators becomes higher than literature-based metrics. After the introduced engagement rate metric, the comment number catches the highest agreement with an AUC score of 0.59. The worst performance is provided when post numbers are used, which can be expected.

Figure 3.27 showed that the introduced metric outperforms the other literature metrics. However, it is also possible to compare the performance of the introduced metric with annotators one by one. To do this, annotators' performances against group decisions must be measured. This can be feasible by leaving one annotator out. For example, if we want to compare annotator_1's performance to the ground truth, which is the decision determined by annotator agreement, while ignoring annotator_1, annotator_1's responses will be used as predictive answers. This will let us understand the performance of the annotator. Figure 3.28a shows the ROC curves of individual annotators and the system by implementing the "leaving one

annotator out" approach. The best performance among annotators comes from annotator_3 with an AUC score of 0.668. This was expected since the internal consistency of the annotator_3 is the best among others, according to figure 3.20. The standing of the introduced metric is 3^{th} with an AUC 0.624. This proves that the introduced metric has the capability to make human-level decisions.



(a) ROC of annotators by leaving one annotator out

(b) ROC of System with annotator ground truth

Figure 3.28 **ROC Curves of annotators and the system**: Figure 3.28a shows the performances of annotators individually. Annotator_3 achieves the highest performance with an AUC 0.668, while annotator_2 achieves the worst performance with an AUC 0.519. The performance of the introduced metric achieves the 3^{th} best performance by overtaking annotator_2 and annotator_4.

So far, the analysis has shown that the introduced metric is adequate for determining engagement rates. The other leg of the system is to measure the performance of the system in similarity. Recall that, while calculating the similarity, the system uses multi-model data types to generate a similarity score. To measure the performance of the system, we may use the correlation score of the system with annotators' decisions, as we did in figure 3.23b. Since the system uses a combination of different data types such as images, text, network and metadata, calculating the correlation score of each component of the similarity system will provide a reference point for the system's performance. This comparison allows us to see if the similarity score calculated by using combined data types gives a better result than the similarity scores calculated by using each data type one by one.

Figure 3.29 shows the comparison of fits created by using different data types. Annotators' survey responses are taken as ground truth. The similarity score for the system is calculated using the equation 3.2. For others, the cosine similarity score

is calculated for the given data type. For instance, if a cosine similarity score for image-based data is to be calculated, the vectors of influencer 1 and influencer 2 that contain only image features will be used in the cosine similarity calculation. Calculated score is fed to regression line in y axis.

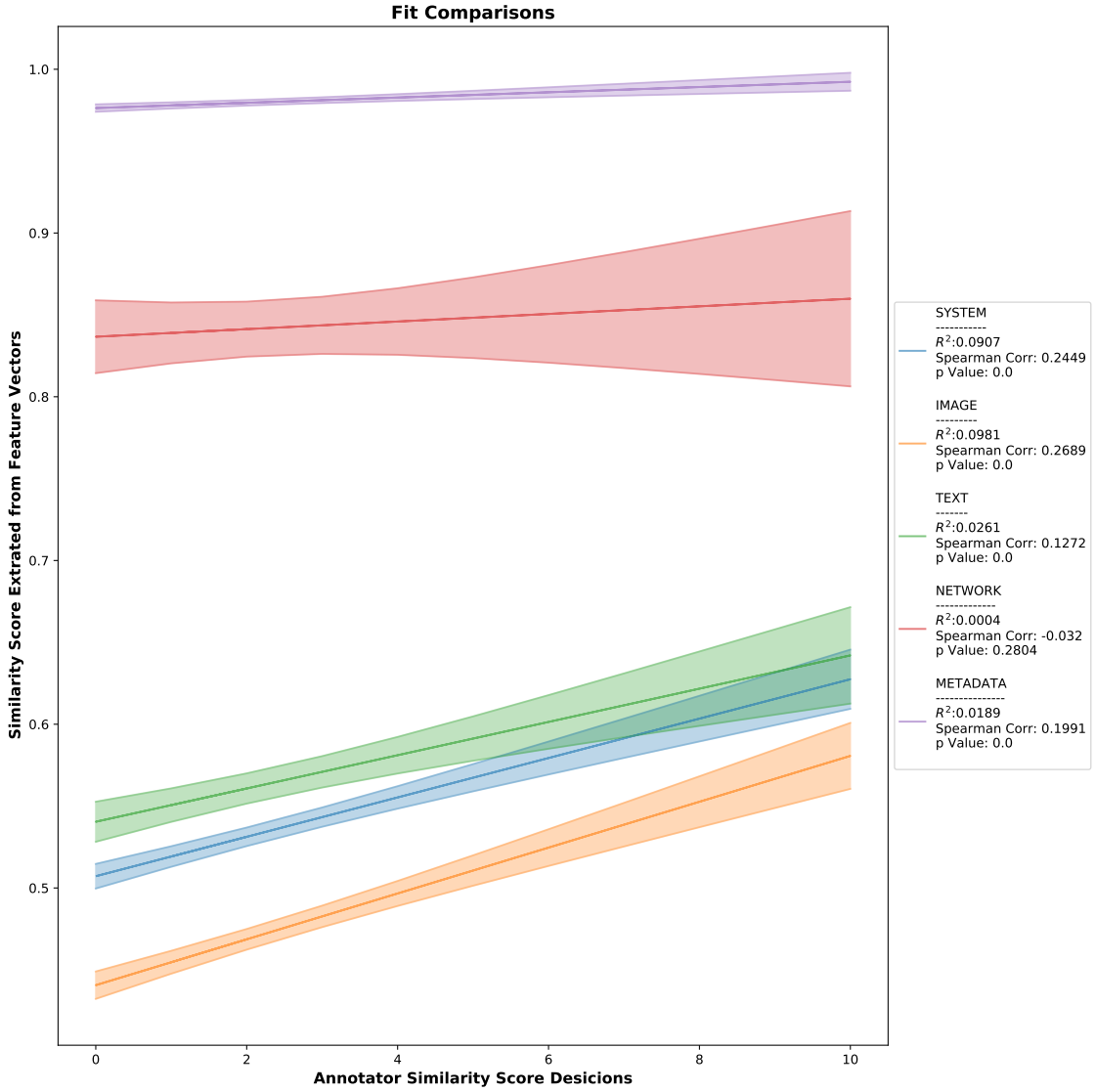


Figure 3.29 **Fit Comparison per Data Type**: As annotator decisions are taken as ground truth, AUC scores of different literature metrics are compared with the newly introduced engagement rate metric. Introduced engagement rate metric appears to be outperforms others when annotators' decisions are taken as ground truth

After regression lines have been fitted into the data for each data type and for the system, Spearman correlation scores are calculated. Generated scores show that when similarity is calculated only using image-based features, the agreement with annotators is highest, with a Spearman correlation of 0.2689. Image-based features are tracked by the system correlation, which is 0.2449. It is shown that network-based similarity calculation does not follow agreement with annotators.

Recall that the similarity score in the system is calculated by using the equation 3.2. For each data type, weights were assigned according to the ratios of the data types in the feature space shown in figure 3.11. As weights change, the Spearman correlation of the system in figure 3.29 will also change dramatically. By using this chart, better weights can be assigned since the relation of each data type between annotators' decisions can be easily seen.

We also tried to use embeddings that represent content better than the other embeddings. To find the best 10 content respresent embeddings, we have compared each embedding with others to generate cosine similarity score. Then, a heatmap is generated that the elements are calculated similarity scores. Then, column-wise summation is applied to find the best 10 content represent embeddings. Same evaluation is executed by using these embeddings only and it appears that the Spearman correlation scores are slightly lower than Figure 3.29. To investigate why did it happen, we have checked the internal consistency of shared posts by influencers in figure 3.30.

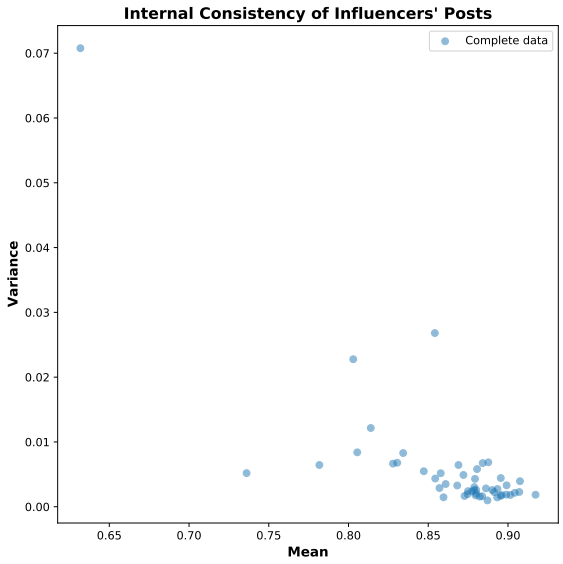


Figure 3.30 **Internal Consistency of Influencers**: Scatter shows the distribution of variances and meanss of embedding cosine similarities per influencer. If an influencer shares similar content most of the time, we expect to see the mean close to 1 and variance close to 0.

X axis shows the mean of cosine similarity scores and y axis shows the variance of them. If an influencer shares consistency, then, mean of it will be close to 1 and variance will be close to 0. Since randomly picked influencers' means are close to 1 and variances are close to 0, we don't need to extract any embeddings since created mean embedding will get influenced by using low amount of embeddings which will result with a lack of information. This will reduce the performance of the model.

4. Discussion & Results

In this thesis, we introduce a new metric that can be used as a new engagement scorer that mainly aims to make estimations regarding organic engagers with the help of an ecology-based population estimation method called capture-recapture. Counting commenters and identifying mutual of them would give a more organic score in terms of engagement. To observe the performance of the introduced metric, we compared its performance with literature-used metrics by taking the decisions of 4 different human annotators as ground truth. The results shown in table 4.1 highlight the comparison of the performances of each of the literature-proven metrics with the introduced metric. Results show that the performance of the introduced metric in terms of identifying influencers that have a high engagement rate outperforms a literature-proven metrics when decisions of 4 annotators are taken as ground truth. The introduced metric can be so reliable that its performance even passed two of the annotators.

Table 4.1 **AUC scores of compared metrics**: Performance comparison of used metrics. Annotator_3 contributes the best performance with AUC score 0.67 while Post number metric shows a lack of performance with an AUC score 0.453. The introduced metric takes 3^{th} place which outperforms all industry used metrics as well as annotator_4 and annotator_2 with AUC score 0.624.

AUC Scores of Compared Metrics	
Metric Name	**Score**
Annotator_3	0.67
Annotator_1	0.66
System	0.62
Commenter Num	0.60
Like Num	0.58
Annotator_4	0.56
Followings	0.55
Annotator_2	0.52
Followers	0.51
Post Num	0.45

Another interesting outcome observed from the table 4.1 is the performance of the following number. Although initially looking by eye, following a number has no effect on engagement. As Cresci, Di Pietro, Petrocchi, Spognardi & Tesconi (2015) claimed, accounts that have a lot of followers but an inadequate number of followings may indicate that a large portion of the followers are covered by bots. Thus, as the number of followings increases, the proportion of organic engagers increases, which lets the organic engagement rate increase.

The quality of ground truths is another factor that influences AUC scores. Annotators were chosen among those who frequently use social media platforms, especially Instagram. Since they have no background on the marketing methodologies or identifying engage accounts, the ground truth decisions they create may not be as valuable as expert decisions, which may drive the AUC score of the introduced metric slightly lower than its actual state. Besides, recall that while scraping posts of influencers, we took the first 100 commenters', if there are any, of the first 32 posts' information and used it while calculating capture-recapture scores due to the high computational cost. If we were able to scrape all posts and commenters for an influencer, the calculated capture-recapture score would reflect a better organic engagement estimation for the influencer, which may result in an increase in the AUC score of the introduced metric.

Recall that in the calculation of the introduced metric, a linear regression line was fitted to the data to come up with comparable engagement rates, as the vision is provided in figure 3.12. The fit of the linear regression line will increase as the number of samples increases since the estimation of the expected capture recapture scores calculated by the fit will converge to the real estimation. However, due to challenges in the Instagram scraping operation, such as IP blockades and dynamic and frequent changes on the source of the Instagram HTML page, the collected number of samples is fixed. As we scrape a higher number of samples, the fit of the regression line will converge to reality, which results in better engagement rate calculations.

Beside the proposed engagement rate metric, we have introduced a new way to calculate similarity scores for platforms that contain multi-type data. Since experiments for this thesis were held on Instagram, the used data types are image, text, metadata, and network. To observe the performance of the proposed similarity score calculation, the annotators' observations regarding the content similarity were taken as ground truth and checked against the Spearman correlation scores. The presentation of image-based features to influencers performed slightly better than the system. There may be several causes that led to this outcome. The first reason

Table 4.2 **Spearman Correlation Score Comparison of Used Presentations**: Performance comparison of used presentations Image based presentation of influencer contributes the best performance with Spearman score 0.2689 while System base representation shows a slightly lower performance with Spearman score 0.2449.

Similarity Agreement Correlation Scores of Compared Metrics	
Metric Name	**Spearman Corraletion**
Image	0.269
System	0.245
Metadata	0.199
Text	0.127
Network	0.032

is the behavior of annotators. Since it is easier for humans to make decisions by looking at pictures only, the annotators may have a bias toward image-based content but ignore other reflectors, which ends up with a higher score for image-based representation. Other than that, recall that the introduced similarity score calculation relies on hyperparameter tuning, which effects the dominance of each data type on similarity decisions. While making decisions on these hyperparameters, we used the ratio of each data type on the influencer representations shown in figure 3.11. A better hyperparameter adjustment will cause the system score to pass the image-based representation.

5. Conclusion

Social media platforms such as Instagram consistently grow, becoming a market for brands and firms. The demand and usage of influencer marketing is enormous; influencers are starting to manipulate metrics in the literature by using bots from service providers, which results in lower profits on the firm and brand sides. In this thesis, we come up with a new method to calculate the organic engagement rate, which is the total number of engagements that are provided by real accounts. We also launched a system that suggests similar influencers and has higher engagement rates than the target influencer.

To validate the proposed approaches, 4,527 Instagram account was scraped by using 3 different heuristics. The first heuristic aims to collect accounts without ignoring whether an account is an influencer or not, which provides a negative kurtosis distribution of follower numbers as shown in figure 3.1a. This dataset is used to observe the performance of the capture-recapture method on the Instagram ecosystem. The second heuristic focused on the Turkish influencer market and scraped only Turkish influencers. This dataset is used to evaluate the performance of the proposed influencer recommendation system by conducting a user annotation study. The last heuristic was used to collect influencers that were classified by the experts. The collected influencers belong to this heuristic separated into three groups as food influencers, tech influencers, and fashion influencers. This dataset is used to evaluate the performance of pre-trained models that will be further used to extract image and text based embeddings.

We compared pre-trained models to determine the best one to be used in embedding extraction by using the models' embeddings to estimate influencers' classes by using the Support Vector Machine algorithm. Although evaluation metrics for both images and text were not distinguishable from each other, models that separate influencer classes better on the UMAP embedding system are used in the system, which are VGG-16 for images and MiniLM-L6 for text. The evaluation of pre-train model performances was conducted image-wise and influencer-wise.

Before the evaluation of the influencer recommendation system, an annotation study is conducted by using 4 annotators to construct a ground truth dataset for pair-wise influencer comparisons. Questions regarding content similarity and engagement rate were asked to annotators, and we collected their responses as a score between 0 and 10 on a linear scale. By using the annotators' responses, common decisions were made by annotators and used them as ground-truth. To check the quality of the responses, analysis based on annotators were made and be ensured.

The performance of the recommendation system was evaluated using ground truths obtained from annotators. The system's performance regarding the evaluation of engagement rate was compared with both literature metrics and annotators. With an AUC score of 0.62, the introduced engagement rate metric was shown to be the best metric among literature metrics and 3^{th} best in all comparisons by eliminating two annotators. Considering the results of annotators and their performance, our systems achieve human-level performance and perform better than other popular measures used in influencer marketing. This shows the effectiveness of the proposed engagement rate calculation. We have also shown the performance of the system regarding identifying content similarity by comparing it with data-type-based representations of influencers. The results show that the system's similarity score calculation is in the top tier, with a Spearman correlation score of 0.2449 with annotator ground truths.

The outcomes of the thesis showed that literature used metrics were insufficient for identifying organic engagements, which points to the need for a new engagement rate metric. By using the capture-recapture methodology inspired by the approach in ecology, organic engagement rates can be identified better from the literature used metrics. Although we used the capture-recapture methodology to come up with an organic engagement rate, it can also be used to generate a score for content similarity by trying to capture mutual commenters among influencers. The influencers that contain the same commenters in their ecosystem will be assumed to be similar since humans behave to follow similar content generally (McPherson et al. (2001)).

BIBLIOGRAPHY

(2019). Engagement rate. `https://sproutsocial.com/glossary/engagement-rate/`.

Abidin, C. (2015). Communicative intimacies: Influencers and percieved interconnectedness.

Abidin, C. (2016). Visibility labour: Engaging with influencers' fashion brands and# ootd advertorial campaigns on instagram. *Media International Australia*, *161*(1), 86–100.

Akpinar, E. & Berger, J. (2017). Valuable virality. *Journal of Marketing Research*, *54*(2), 318–330.

Amos, C., Holmes, G., & Strutton, D. (2008). Exploring the relationship between celebrity endorser effects and advertising effectiveness: A quantitative synthesis of effect size. *International journal of advertising*, *27*(2), 209–234.

Belanche, D., Casaló, L. V., Flavián, M., & Ibáñez-Sánchez, S. (2021). Building influencers' credibility on instagram: Effects on followers' attitudes and behavioral responses toward the influencer. *Journal of Retailing and Consumer Services*, *61*, 102585.

Benevenuto, F., Magno, G., Rodrigues, T., & Almeida, V. (2010). Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, (pp.1̃2).

Bertini, M., Ferracani, A., Papucci, R., & Del Bimbo, A. (2020). Keeping up with the influencers: Improving user recommendation in instagram using visual content. In *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, (pp. 29–34).

Breves, P. L., Liebers, N., Abt, M., & Kunze, A. (2019). The perceived fit between instagram influencers and the endorsed brand: How influencer–brand fit affects source credibility and persuasive effectiveness. *Journal of Advertising Research*, *59*(4), 440–454.

Cao, Q., Yang, X., Yu, J., & Palow, C. (2014). Uncovering large groups of active malicious accounts in online social networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, (pp. 477–488).

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 1251–1258).

Colliander, J. & Dahlén, M. (2011). Following the fashionable friend: The power of social media: Weighing publicity effectiveness of blogs versus online magazines. *Journal of advertising research*, *51*(1), 313–320.

Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2015). Fame for sale: Efficient detection of fake twitter followers. *Decision Support Systems*, *80*, 56–71.

De Veirman, M., Cauberghe, V., & Hudders, L. (2017). Marketing through instagram influencers: the impact of number of followers and product divergence on brand attitude. *International journal of advertising*, *36*(5), 798–828.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A

large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, (pp. 248–255). Ieee.

Evans, N. J., Wojdynski, B. W., & Grubbs Hoy, M. (2019). How sponsorship transparency mitigates negative effects of advertising recognition. *International Journal of Advertising*, *38*(3), 364–382.

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Communications of the ACM*, *59*(7), 96–104.

Guy, I. (2015). Recommender systems handbook, chapter social recommender systems.

Guy, I. (2018). People recommendation on social media. In *Social information access* (pp. 570–623). Springer.

Hao, S. & Feamster, N. (2008). Estimating botnet populations from attack traffic.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Holland, C. (2019). The power of the influencer. *BDJ Team*, *6*(9), 20–21.

Hughes, C., Swaminathan, V., & Brooks, G. (2019). Driving brand engagement through online social influencers: An empirical investigation of sponsored blogging campaigns. *Journal of Marketing*, *83*(5), 78–96.

Jensen Schau, H. & Gilly, M. C. (2003). We are what we post? self-presentation in personal web space. *Journal of consumer research*, *30*(3), 385–404.

Ki, C.-W. Kim, Y.-K. (2019). The mechanism by which social media influencers persuade consumers: The role of consumers' desire to mimic. *Psychology & Marketing*, *36*(10), 905–922.

Knoll, J., Schramm, H., Schallhorn, C., & Wynistorf, S. (2015). Good guy vs. bad guy: the influence of parasocial interactions with media characters on brand placement effects. *International Journal of Advertising*, *34*(5), 720–743.

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, (pp. 591–600).

Le Cren, E. (1965). Some factors regulating the size of populations of freshwater fish: With 3 figures in the text. *Internationale Vereinigung für Theoretische und Angewandte Limnologie: Mitteilungen*, *13*(1), 88–105.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

Leung, F. F., Gu, F. F., Li, Y., Zhang, J. Z., & Palmatier, R. W. (2022). Express: Influencer marketing effectiveness. *Journal of Marketing*, 00222429221102889.

Leung, F. F., Gu, F. F., & Palmatier, R. W. (2022). Online influencer marketing. *Journal of the Academy of Marketing Science*, *50*(2), 226–251.

Lou, C. & Yuan, S. (2019). Influencer marketing: how message value and credibility affect consumer trust of branded content on social media. *Journal of Interactive Advertising*, *19*(1), 58–73.

Lucy, B. (2021). The benefits of nano-influencer marketing. `https://www.origingrowth.co.uk/blog/the-benefits-of-nano-influencer-marketing`.

Manly, B. (1970). A simulation study of animal population estimation using the capture-recapture method. *Journal of Applied Ecology*, 13–39.

McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology*, 415–444.

Pollock, K. H., Nichols, J. D., Brownie, C., & Hines, J. E. (1990). Statistical inference for capture-recapture experiments. *Wildlife monographs*, 3–97.

Sandra, C. (2021). Instagram influencer marketing: 5 reasons to work with micro-influencers. `https://mention.com/en/blog/instagram-influencer-marketing/`.

Schemer, C., Matthes, J., Wirth, W., & Textor, S. (2008). Does "passing the courvoisier" always pay off? positive and negative evaluative conditioning effects of brand placements in music videos. *Psychology & Marketing*, *25*(10), 923–943.

Seering, J., Flores, J. P., Savage, S., & Hammer, J. (2018). The social roles of bots: evaluating impact of bots on discussions in online communities. *Proceedings of the ACM on Human-Computer Interaction*, *2*(CSCW), 1–29.

Sen, I., Aggarwal, A., Mian, S., Singh, S., Kumaraguru, P., & Datta, A. (2018). Worth its weight in likes: Towards detecting fake likes on instagram. In *Proceedings of the 10th ACM conference on web science*, (pp. 205–209).

Silvera, D. H. & Austad, B. (2004). Factors predicting the effectiveness of celebrity endorsement advertisements. *European Journal of marketing*.

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Valsesia, F., Proserpio, D., & Nunes, J. C. (2020). The positive effect of not following others on social media. *Journal of Marketing Research*, *57*(6), 1152–1168.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wirth, K., Menchen-Trevino, E., & Moore, R. T. (2019). Bots by topic: exploring differences in bot activity by conversation topic. In *Proceedings of the 10th International Conference on Social Media and Society*, (pp. 77–82).

Zazat, N. (2017). Social media experiment reveals how easy it is to create fake instagram accounts and make money from them. `https://www.independent.co.uk/tech/social-media-experiment-fake-instagram-accounts-make-money-influencer-star-blogger-mediakix-a7887836.html`.

APPENDIX A

Data Structure

Social media platforms such as Instagram provide extensive information that might be challenging to organize and analyze systematically. Since the main concern of this project is to create a product that can be used in live streaming, the system that is created should not rely on a single dataset that will be updated as new influencers are added. Otherwise, any unrecognized corruption on file while adding new influencers results in a system failure. In addition to this, it would not be efficient to read one huge file at a time every time. Because of these reasons, we have decided to store the information for each influencer in a different folder that contains every piece of information regarding the influencer. In this case, it is easier for a model to work with streaming data since it gets each influencer-wise information one by one. As a result, the operation of reading influencers is independent of one another, and if an influencer is corrupt, it can be easily ignored while making suggestions. Of course, we can utilize relational or unstructured database systems, but our initial efforts prefer simplicity over time-consuming engineering efforts.

Figure A.1 shows the directory tree construction of the dataset. As it can be seen, each influencer is separated into folders, and there are sub-folders in each influencer tab to make it more organized. `raw` folder contains files related to the profile page, posts, comment-based information for each scraped post, a batch of follower and following information, etc. Each folder type is built differently than the others. For example, the folder that contains information regarding the profile page (which is named as `influencer1.json` in Figure A.1) holds the complete dictionary as it is stored in Instagram. It is also the same for post-related files such as file `postId1.json` in Figure A.1. However, the construction of `commenters.json` is as follows:

Figure A.1 **Scraped Data Structure**: General look to the dataset directory tree before having data pre-processing

Listing A.1 `commenter.json` structure

```
{
"postId1":{"commenterUsername1":["Comment1", "Comment2", ..., "CommentN"],
           "commenterUsername2":[...],
           ...,
           "commneterUsernameM":[...]},
"postId2":{...},
...,
"postIdK":{...}
}
```

As it can be seen from A.1, commenter-related information is stored using the nested dictionary concept, where the post id is the first key and the value is another dictionary. The inner dictionary contains information regarding the post only. In this scenario, keys are the commenters that made comments on the given postid, and values are lists that contain the comments that the reference user made. Because

it is possible for a user to enter more than one comment, comments are stored in lists. Because we only scrape 100 comments, the number of comments N cannot exceed 100. For the same reason, the commenter number M can also not exceed the number 1000. The post id number K represents a maximum of 32 posts.

Listing A.2 `follower.json/following.json` structure

```
{
"accountName": {"is_private": bool,
                "follower": ["followerUsername1", "followerUsername2",
                    ..., "followerUsernameL"]}
}
```

The structures of `follower.json` and `following.json` are identical. They both contain a dictionary that holds only one key, which is the username of the influencer that is found in the target folder, as it is visualized in Listing A.2. Inside of inner dictionary, there are 2 keys which are `is_private` and `follower`. `is_private` holds an information whether an account is private or not. The value can only be either `True` or `False` where `True` shows that the account is private and `False` shows that it is public. Note that the system that has been developed does not generate a score for private accounts. `follower` key holds a list value regarding the follower's username. Total follower number L cannot exceed 32.

Images for each scraped post are stored in a different folder called `images`. This folder contains only post images with `.jpg` extension and named by the postId.

The structure that was shown in Figure A.1 shows only the state of the folder after the scraping operation for influencers is finished. The folder is expanded after new files are added with pre-processing, as can be seen in Figure A.2. There are files that have identical names from the `raw` folder. These are the simplified versions of the corresponding folders, which are faster to read. The structure of the `influencer.json` can be seen in the listing A.3. It contains relevant information such as the number of followers, followings, category type, etc. These information are later be used while creating a pipeline for the validation of the system and performance comparison.

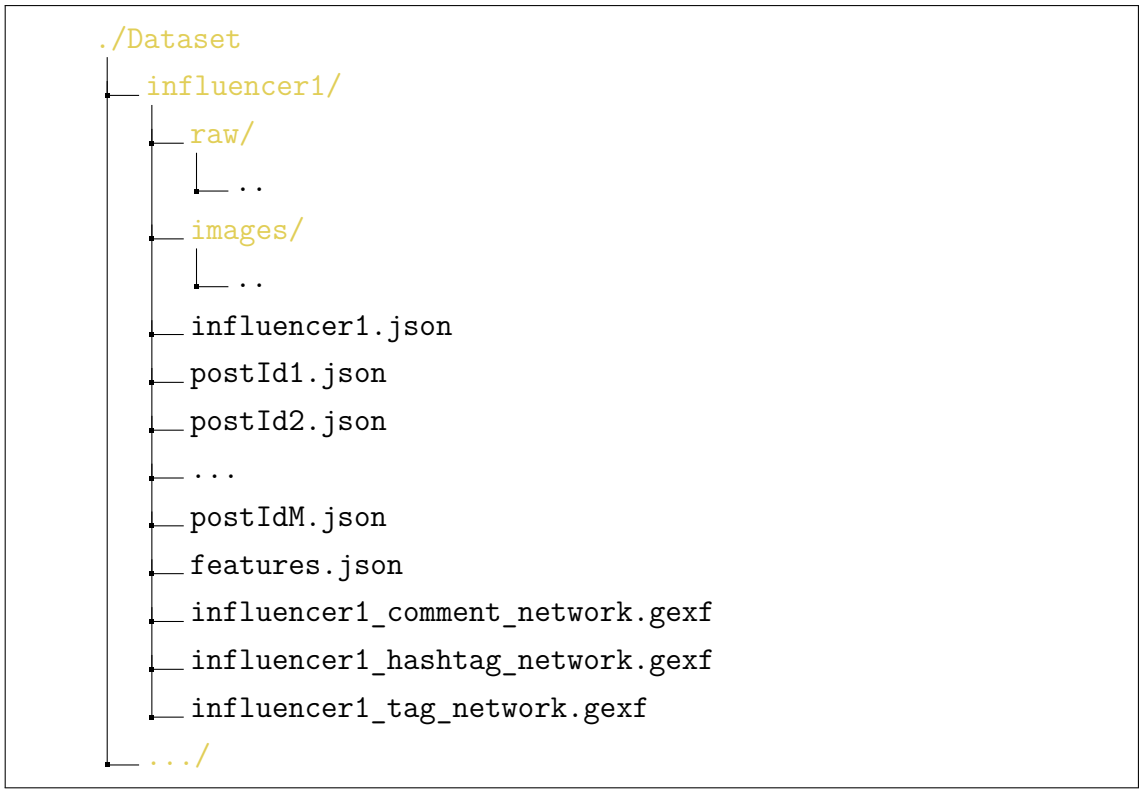


Figure A.2 **Directory Structure after Preprocessing**: The structure of the dataset Folder after preprocessing is finished.

Listing A.3 `influencer1.json` structure

```
{
"id": int,
"posts": int,
"followers": int,
"following": int,
"full_name": str,
"verified": bool,
"category_type": str }
```

Listing A.4 `influencer1.json` structure

```
{
"upperdata": {
    "post_id": str,
    "shortcode": str,
    "is_video": bool,
    "like_number": int,
    "timestamp": int,
    "text": str,
    "comment_number": int},
"tags":{
    "TAGGED_USERNAME1":{
        "full_name": str,
        "id": int:
        "is_verified": bool,
        "username": str
        },
    "TAGGED_USERNAME2": {...},
    ...,
    "TAGGED_USERNAMEN": {}
    }
"comments":{
    "COMMENTER_USERNAME1": {
        "text": ["comment1", "comment2", ..., "commentK"],
        "user_info":{"username": str}
        },
    "COMMENTER_USERNAME2": {...},
    ...,
    "COMMENTER_USERNAMEM": {...}
    }
}
```