# PREDICTING POPULARITY ON INSTAGRAM: INTEGRATING DEEP LEARNING, MACHINE LEARNING, AND EXPLAINABLE AI FOR ENHANCED SOCIAL MEDIA ANALYTICS

JAVIER RENE TORRALBA FLORES

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

### student number

[redacted]

### committee

dr. Drew Hendrickson
Hezha Mohammedkhan MSc

### location

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

### date

May 20th, 2024

### word count

8,615

### acknowledgments

I want to thank my parents, Marycarmen de Torralba and Victor Javier Torralba for their unwavering support throughout my university studies. I want to thank my supervisor, Drew Hendrickson, for the guidance throughout the supervision of this thesis. Finally, also want to thank Daniel Fidrmuc for the help and good times at the library working on this project.

# PREDICTING POPULARITY ON INSTAGRAM: INTEGRATING DEEP LEARNING, MACHINE LEARNING, AND EXPLAINABLE AI FOR ENHANCED SOCIAL MEDIA ANALYTICS

JAVIER RENE TORRALBA FLORES

## Abstract

In an era where social media platforms like Instagram influence economic and societal outcomes, accurately predicting the popularity of posts becomes essential. This thesis investigates the potential of integrating deep learning and machine learning models to forecast the popularity of Instagram posts before they are published. This research can be summarised in how integrating deep learning, machine learning, and explainable AI techniques optimizes the prediction and understanding of Instagram post popularity, considering diverse data features and error dynamics. Obtaining a sample of 133,642 posts from 27,893 different users, I used a ResNet50 pre-trained on ImageNet and a BERT model pre-trained on Twitter data to extract image and text features. Different sets of features, including images, text, and user features, are tested on three different algorithms: XGBoost, Light Gradient Boosting Machine (LGBM), and Deep Neural Networks. The best-performing algorithm, LGBM, used all features and achieved an F1 score of about 69%. An error analysis shows that the error rate depends on the predicted probability value made by the model. SHAP values reveal that the number of followers and posts are the most influential in predicting the popularity of a post, yet image and text features can have a stronger predictive value together. Combining the best-performing model, domain knowledge, error analysis, and SHAP values can help Instagram users make data-driven decisions to post the most engaging content.

# 1 DATA SOURCE, ETHICS, CODE, AND TECHNOLOGY STATEMENT

## 1.1 *Source/Code/Ethics/Technology Statement Example*

Data Source: The data, including each Instagram post's images, text, and metadata information, has been acquired from the Kim et al. (2021) through an online request. The original data owner complied with Instagram's policy for data collection. The original data owner also released this dataset to the public upon filing a request stating that this would be used for research purposes. The original owner of the data during and after the completion of this thesis retains ownership of the data. However, the institution is informed about using this data for this thesis and potential research publications. The author created all the figures using the shared data from Kim et al. (2021). The thesis code can be accessed through the GitHub repository by clicking here. The site contains a portfolio of the author's data science and economics research, including all relevant codes for this thesis. Because of GDPR reasons, the data is not made public on this site. Several packages were used for the coding of this thesis, both in R and Python. You can find a list of the packages used here:

- Numpy (Harris et al., 2020)
- Matplotlib (Hunter, 2007)
- Pandas (pandas development team, 2020)
- Scikit Learn (Pedregosa et al., 2018)
- LightGBM (LightGBM, 2023)
- XGBoost (XGBoost, 2022)
- TensorFlow/Keras (Martín Abadi et al., 2015)
- Dplyr (Wickham et al., 2023)
- Tidyverse (Wickham et al., 2019)
- Purrr (Wickham & Henry, 2023)
- Seaborn (Waskom, 2021)
- Data.table (Barrett et al., 2024)
- PIL (Umesh, 2012)
- Transformers (Vaswani et al., 2023)

- roBERTa pre-trained on Twitter data (Barbieri et al., 2020)
- Shap (Lundberg & Lee, 2017a)
- Google Translate API
- ResNet 50 pre-trained on ImageNet (He et al., 2016)

Aside from programming packges, I also used the following software for my thesis:

- Google Collab
- Draw.io

To the best of my knowledge, there is no sign of bias against different groups in terms of representation and inclusiveness in the data. The author produced all the code using platforms like Google, Stack Overflow, and ChatGPT (OpenAI, 2023) for debugging and code optimization. ChatGPT and Grammarly were used to improve the author's original content for paraphrasing, spell-checking, and grammar. No other typesetting tools or services were used.

## 2 INTRODUCTION

Instagram is one of the most popular social media platforms, with over 2 billion monthly active users (Barinka, 2022). It has increasingly become an important site for everyday people who join for entertainment or social purposes. But it has also become a tool for influencers, marketers, and entrepreneurs who seek to commercialize their products or services. According to Dixon (2024), 80% of global marketers use Instagram. There are 1.3 billion photos shared per day on the platform per Aslam (2024).

Given the social media platform's massive reach, users who post on the app seek the highest engagement rate. Any published content must be highly engaging and attract as much positive attention. But how can you ensure you do this consistently? What exactly could drive this positive attention? Several popular sites provide recommendations, but what if instead of relying on intuition and feeling for what could be a popular post, we could take a data-driven approach to efficiently and effectively forecast a popular post on the platform? What if we could make this robust with a thorough error analysis to help the user understand when the model fails? And what if we could use explainable AI techniques to uncover how the model makes decisions? Forecasting popular posts before posting them could be an incredibly useful and powerful tool for influencers and marketers wishing to post the most engaging content on the platform.

Delving into such a topic is beneficial for society. Building an efficient and effective Instagram popularity prediction model could benefit publishers on the app. It would better inform influencers and marketers about different machine-learning methods they could use to forecast popularity. It could also help guide decisions about what posts are most engaging and should be prioritized. The use of a machine learning model, with thorough error analysis and explainable AI to understand how the model makes decisions, would be a great societal contribution to people seeking to create consistently high engagement on the platform.

The literature on this topic is vast. For instance, Riis et al. (2021) and Zhang et al. (2018) use quantitative metrics to predict popularity using likes, engagement rates, or personalized measures of popularity. However, the research community has not yet converged on a universally accepted definition of "popularity" that could discern between high and low engagement rates. Exploring a novel definition of popularity, discerning between high and low engagement, can help the existing literature understand how different measures affect models and predictions. At the same time, a novel definition of popularity in the literature can help creators better grasp the different possibilities available to create powerful machine learning models. No studies in this field have evaluated whether the errors of their models depend on the predicted values outputted by their models. Also, no study on social media popularity predictions has used SHAP values and gone in depth about how these could have individual forecasting decisions. Given the vast literature on this topic, I first identify the gaps on the literature and then present my research questions under section 4.

This thesis is scientifically relevant as it contributes to a field with various definitions of popularity. It also contributes using explainable AI methods to understand how the model makes decisions. The study also presents a unique error analysis showing how the model fails. Combining the model, explainable AI techniques, error analysis, and user domain knowledge can be a powerful data-driven approach to deciding the best posts to publish.

This thesis fills the literature gaps using state-of-the-art modeling techniques, such as ResNet50 for image and BERT for text feature extraction and tests three different machine learning models. It explores the model's error patterns, uses explainable AI techniques to explain how the model decides popularity and provides advice for its users on combining this information to extract the most useful information.

My results show that the best-performing model achieves about 69% accuracy, outperforming the baseline of 50%. Most features, including image, text, and user information, contribute to predicting popularity, with the number of followers and posts being the most influential features.

My work's outline consists of section 3 with a literature review, delving into papers with related themes and methodologies to define state-of-the-art modeling, followed by section 4, which defines the research strategy and research question. After this, section 5 explains the methodology and experimental set-up, defining my data, methods, and modeling techniques. Once this is established, section 6 reviews the results of my model. After this, section 7 discusses the performance, limitations, and future research strategies. Lastly, section 8 concludes.

# 3 RELATED WORK

## 3.1 *Social media popularity prediction literature*

Research about the popularity of general social media and Instagram posts exists in the data science literature and has varied over the years with different purposes. Research ranges from predicting whether a single image will go viral (Ding et al., 2019), predicting the popularity of publications across different sites (J. Chen et al., 2019; Hsu et al., 2019) to predicting the number of likes an Instagram post will get (Gayberi & Oguducu, 2019; Gupta et al., 2020; Zhang et al., 2018), to only using metadata from a post to predict popularity Carta et al. (2020). These studies have been driven to understand what kind of posts are most likely to go viral, have more engagement, or be more popular, ranging from using a combination of text, images, and metadata features. Several have also been motivated to understand what makes a user famous on the platform.

### 3.1.1 *Types of datasets used to predict a post's popularity*

In this section, I explore the types of datasets used for research related to this thesis. Instagram datasets are not easy to access because the social media site does not provide an API for obtaining information at the post level. The data must be scraped over time or accessed through a third party. Ding et al. (2019) aims to predict the virality of an image. For this, they use a dataset with 2.5 million images. There are a total of 110,000 users in their data. Their database includes information on the user, such as the number of followers and posts, and information from the post, such as images, captions, and posting time. Riis et al. (2021) used a dataset with a sample of 1 million posts. They do not specify how many users are present in the sample but highlight that the dataset has information at the user and post level. Zhang et al. (2018) uses images and captions for their studies to predict the popularity of a post. Their data consists of 60,785 posts by 441 users, similar to the other studies, containing information at the user and post level. Lastly, Carta et al. (2020) used only post captions and metadata on the user to make predictions about the popularity of a post. They have 100,000 posts from 2,500 accounts. All in all, datasets for this type of study range from a few thousand observations to millions. This will motivate the data I use under section 5.1.

### 3.1.2 *Definitions of popularity*

Among the published research on predicting popularity, studies have not converged on a definition of popularity. Riis et al. (2021) uses likes as their form of popularity, aiming to predict the number of likes a post will get. As section 5.2 explores, the distribution of a post's likes on social media sites is often highly skewed, with most posts having a small number of likes and a few posts having a very large number of likes. Because of this, some studies have used logarithmic scales to adjust for the high skewness, such as Gayberi and Oguducu (2019) and Riis et al. (2021). Others opt for creating a binary classification, where a metric for popularity is created from other features to define popularity. For instance, Carta et al. (2020) uses a binary variable, where they use the moving average of the number of likes of an account to determine whether the next post they will make will be more popular than the previous set of posts they made. Similarly, Zhang et al., 2018 also uses a binary variable as their target, where the top 25% of a user's posts are classified as popular, and the bottom 25% are classified as unpopular.

The engagement rate is usually defined as likes plus comments over followers. This metric is important for site publishers as it provides a threshold for how much engagement each post generates relative to the account size. Several popular sites and blogs advising marketers and influencers mention that 3.5% or higher are desirable rates (Demeku, 2023; Lewis, 2022).

### 3.1.3 *Image and text feature extraction*

Given the nature of Instagram data, which are images and text, studies have used feature extraction methods to analyze them. ResNet50, ResNet18, YOLOv3, Inception-V3, and EfficientNet-B6 are all examples of pre-trained transfer learning models that other studies have used for image feature extraction (Ding et al., 2019; Kim et al., 2020; Riis et al., 2021; Zhang et al., 2018). These have been used to extract high-level features, context, scenery, or general features from the images.It has been found that the ResNet50 model pre-trained on ImageNet data performs best at getting the most useful feature from Instagram images to make predictions about their popularity, according to Ding et al. (2019) and Riis et al. (2021).

Moving on to the extraction of text features, several studies have used models like Latent Dirichlet Allocation (LDA), Long Short Term Memory (LSTM), SentiStrength, or pre-trained BERT to extract features from text (Carta et al., 2020; Kim et al., 2020; Zhang et al., 2018). While feature extraction methods like LSTMs or LDAs provide robust feature extraction methods, pre-trained BERT models strike a balance between time efficiency and relevant feature extraction. Kim et al. (2020) showed this during a study where they classify Instagram posts according to their content category, using a pre-trained BERT model to extract text features.

### 3.1.4 *Algorithms and performance*

After extracting text and image features, all features are concatenated with the rest of the metadata and plugged into machine learning algorithms. Among the used algorithms, but with poor performance, are Decision Trees, Random Forests, and Support Vector machines (Gayberi & Oguducu, 2019; Gupta et al., 2020). Among the better-performing algorithms is Deep Neural Networks, found by Ding et al. (2019), Gayberi and Oguducu (2019), and Zhang et al. (2018). However, the best-performing algorithms in the literature are gradient-boosting algorithms. For example, Hsu et al. (2019) and Riis et al. (2021) used a Light Gradient Boosting algorithm. On the other hand, Carta et al. (2020), J. Chen et al. (2019), Gupta et al. (2020), and Zhang et al. (2018) all used an XGBoost algorithm.

Since the definition of popularity changes throughout papers, ranging from likes, to logarithmically transformed like counts, to binary or multi-class classification of popularity levels, it is hard to directly compare the performance of several of these models since the problem statements can be either classification or regression problems, with different measurement scales.

For regression problems, such as Gayberi and Oguducu (2019), Gupta et al. (2020), Hsu et al. (2019), and Riis et al. (2021), the best performing models are Gradient Boosting Algorithms, more specifically, Extreme Gradient Boosting regression (XGBoost) and Light Gradient Boosting regression (LGBM regression) performed best at predicting the popularity of social media posts. Riis et al. (2021) targeted the variable "like count" with a log transformation for this feature. The best-performing algorithm is LGBM, which obtained an RMSE of 1.157, Spearman's rank correlation of 0.510, and R2 of 0.283. Gupta et al. (2020) targeted the log-normalized like count, and XGBoost performed best with an RMSE of 0.1231 and Spearman's rank correlation of 0.93. Similarly, Hsu et al. (2019) targeted the log-normalized like count, and LGBM showed the best performance; it obtained a Spearman's rank correlation of 0.656 and an MAE of 1.497.

For binary classification problems, Carta et al. (2020), who used a moving average of likes that a user has to predict whether their next post will be popular, found that an XGBoost algorithm proved to be the best performing one, achieving a performance of 67.50% accuracy and an F1 of 65.81%. Similarly, Zhang et al. (2018) also uses a binary classification to define popularity. Their definition of popularity depends on an individual's past information. They use a deep neural network, which gives the best results, showing an accuracy of 71.19%, F-score of 72.58%, recall of 75.45%, and precision of 69.91%.

### 3.2 *Predictive modeling (not popularity-related) with Instagram data*

Other important sources of multimodal predictive modeling have come from the data source for this paper. Kim et al. (2021) have a vast Instagram dataset on influencer posts. They used a deep neural network, image feature extraction from a pre-trained Inception V-3 model, text feature extraction from a pre-trained BERT model, and metadata to predict whether a post had undisclosed advertising.

Similarly, Kim et al. (2020) also uses Instagram data in a multimodal way to classify the category of each influencer (such as Travel, Food, Fashion, etc.). They used the pre-trained model BERT for text and Inception-v3 for images. They use a CNN model to classify influencers. Their model demonstrates a high accuracy of 98.32% in classifying each influencer's category. Finally, the authors release a dataset of 10 million images and metadata from 33,000 accounts.

### 3.3 *State-of-the-art modeling and unanswered questions*

From the review of the relevant literature, it is possible to extract state-of-the-art modeling techniques. Firstly, the best-performing image feature extraction is the ResNet50 model pre-trained on ImageNet, as per Ding et al. (2019), Kim et al. (2020), and Riis et al. (2021). Secondly, the best-performing methods for text feature extraction are LSTM, LDA, or pre-trained BERT models, per Kim et al. (2020) and Zhang et al. (2018). Thirdly, the best-performing machine learning algorithms are Deep Neural Networks, Extreme Gradient Boost, and Light Gradient Boosting Machines, as per Carta et al. (2020), Ding et al. (2019), Gayberi and Oguducu (2019), Gupta et al. (2020), Hsu et al. (2019), Riis et al. (2021), and Zhang et al. (2018).

While the literature has strongly focused on predicting like counts (Gayberi & Oguducu, 2019; Gupta et al., 2020; Hsu et al., 2019; Riis et al., 2021) or a binary prediction adjusted for individual characteristics (Carta et al., 2020; Zhang et al., 2018), nobody has used the engagement rate as a threshold for popularity. Even though the Instagram community places a high emphasis on having high engagement rates (a high number of likes and comments compared to followers), this has never been used as a threshold to define popularity. Researching whether using the engagement rate as a cutoff for popularity would be worthwhile as it is a highly emphasized metric for the industry.

In addition, the use of Explainable AI methods and SHAP values has remained unexplored in the literature. The closest research on this matter is by Carta et al. (2020), who uses feature importance to highlight important features. SHAP values have remained unexplored in this field. Also, papers exploring binary classifications have not explained whether their error rates change depending on the prediction. It is unclear whether the probability predicted by the model to be asssigned into either category affects the error rates of the model. Finally, while studies like Kim et al. (2021) use pre-trained BERT models, it is unclear whether these models would have a worse performance if non-English observations were included in the datasets. All studies have focused on English language predictions; therefore, it would be worthwhile to evaluate whether feeding non-English captions to a BERT model would affect the performance.

## 4 RESEARCH STRATEGY QUESTIONS

Based on the literature gaps found in section 3.3 I aim to answer the following research questions:

- **RQ1: Out of the models tested, which model performs best in predicting the popularity of an Instagram post with the novel definition of popularity?**
  - **Sub-RQ1.1: Do image and text features add predictive value to a user's metadata with this new measure of popularity?** It is worthwhile to perform an ablation study, where image, text, and user features are included or deleted. Other studies, such as Carta et al. (2020), performed similar ablation studies but have done this without images or this new definition. Given the new definition of popularity, studying how different features provide different predictive power would be worthwhile.
  - **Sub-RQ1.2: After extracting features from an English-based pre-trained BERT model on both English and non-English features, does the performance of the algorithm increase if non-English observations are excluded?** No study touches on this matter. Determining whether the model performs worse

with all types of languages included would be worthwhile to ensure the best performance possible without any biases. Especially considering that I use an English-based BERT model. The same test sets will be used.

- **RQ2: Which features are most important, and what is their relationship with predictions?** Using explainable AI techniques in the form of SHAP values, it is possible to understand what features have the biggest influence and how these features interact to make a prediction.
- **RQ3: Do errors of the best-performing model depend on the estimated probability the model gives each prediction?** Based on the literature, studies that have predicted popularity targeting a binary variable, such as Carta et al. (2020) and Zhang et al. (2018), have not studied whether the errors depend on the probability the model gives to make the prediction. Understanding this could help make the best use of these predictive models.

## 5 METHODOLOGY AND EXPERIMENTAL SETUP

This section reviews the data, its preparation, the modeling techniques, error analysis, and explainable AI methods I use. Figure 5 shows a high-level overview of the text discussed below.

### 5.1 *Data construction and description*

I use data provided by Kim et al., 2021. The original dataset is formed of 1,601,074 posts with 38,113 influencers for six years. The data is split into JSON, JPG, and TXT files, which I clean, wrangle, and transform into a final CSV dataset. First, JSON files contain individual posts information. For example, captions (text accompanying a picture on a post), likes, comments, timestamps, sponsorship tags, user tags, a post's ID, and whether the post contains a video. Each JSON file containing this information, henceforth referred to as metadata, is transformed into a row of information in a CSV file. Second, one set of TXT files contains user information, such as name, number of followers and followees (the number of followers a user has), number of posts made, category of the influencer, and personal information such as email addresses and phone numbers. This data is transformed into a CSV file and later merged with the rest of the metatada. Another set of TXT files contains a mapping file to match JPG files with each post. Third, JPG files contain images of each post in the data. These images are turned into features, as discussed in section 5.3.1, to be concatenated with the rest of the data to create a CSV file.

Each row of data is a post with its relevant information, an image, caption, month, day, hour, likes and comments count, user information such as follower and followees count, a username, a sponsorship label, name and username, number of posts made, category, biography, and number of images posted. A user can post more than a single picture on Instagram. For example, a user can post eight pictures in a single post, but my dataset may contain only two. Because of this, it is decided only to keep observations where only one picture is posted. This aims to avoid biases in the data since not all sets of posts in the data include all images. Hence, the final dataset comprises 133,642 posts, each containing a single picture. There are 27,893 users in the data. Based on the literature reviewed under section 3.1.1, it is sensible to assume this thesis uses an appropriately sized dataset compared to similar sizes of Carta et al. (2020) and Zhang et al. (2018).

## 5.2 *Exploratory Data Analysis*

Before modeling, I explore the most relevant parts of the data through descriptive statistics and visualizations. Table 1 gives descriptive statistics on the distribution of the most relevant numerical features. These features have a high standard deviation, mostly from the high skewness of social media data. The intuition behind this is that most social media users do not have many followers or likes, while a few have massive amounts.

Table 1: Descriptive statistics of numerical values in the data

| | Like Count | Comment Count | Followers | Followees | Number of posts |
|---|---:|---:|---:|---:|---:|
| count | 133,642 | 133,642 | 133,642 | 133,642 | 133,642 |
| mean | 3,926 | 75 | 138,656 | 1,582 | 1,269 |
| std | 34,946 | 917 | 1,340,448 | 3,462 | 1,687 |
| min | 0 | 0 | 198 | 0 | 91 |
| 25% | 206 | 7 | 7,811 | 593 | 469 |
| 50% | 612 | 24 | 20,973 | 995 | 846 |
| 75% | 1,720 | 60 | 62,617 | 1,881 | 1,545 |
| max | 3,850,463 | 169,030 | 119,050,781 | 304,758 | 127,520 |

The most relevant issue with the high skewness of the data is the distribution of likes. As mentioned under section 3.1.2, this becomes an issue when predicting the number of likes a post will get. Some solutions to this problem are logarithmic transformations, as done by Gayberi and Oguducu (2019) and Riis et al. (2021). Figure 1 visualizes this problem (the top 5th percentile is cut from the visualization). Steps to counter this issue are taken in section 5.3.3.

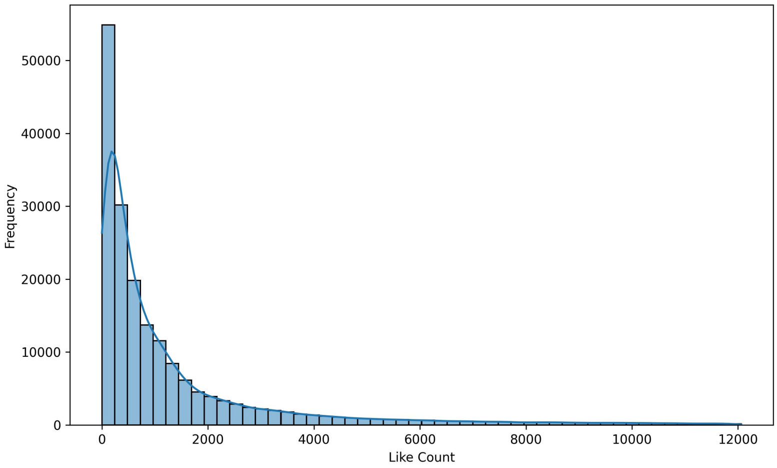

Figure 1: Distribution of likes with the top 5th percentile cut off

Further, figures 2 and 3 provide insight into the time of year and day of the week in which posts are made to give an idea of the timing of the posts. As the visualizations show, the distributions are relatively even.

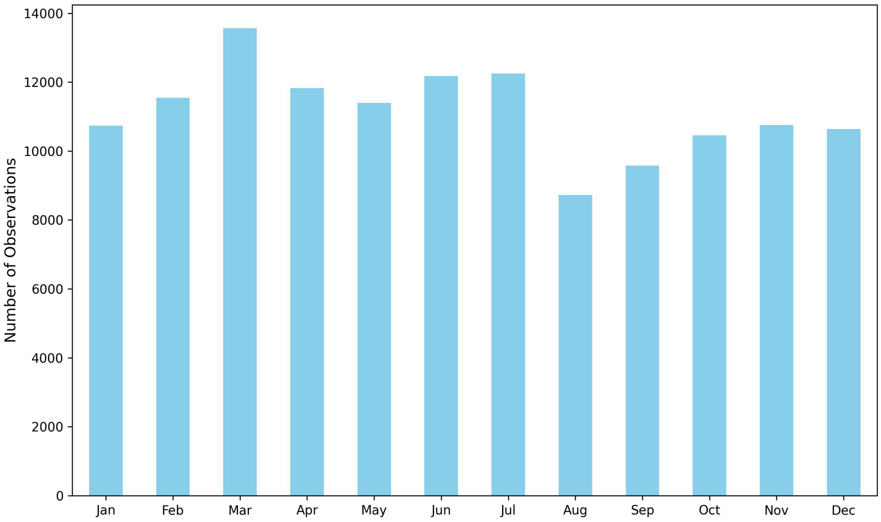

Figure 2: Observations per month

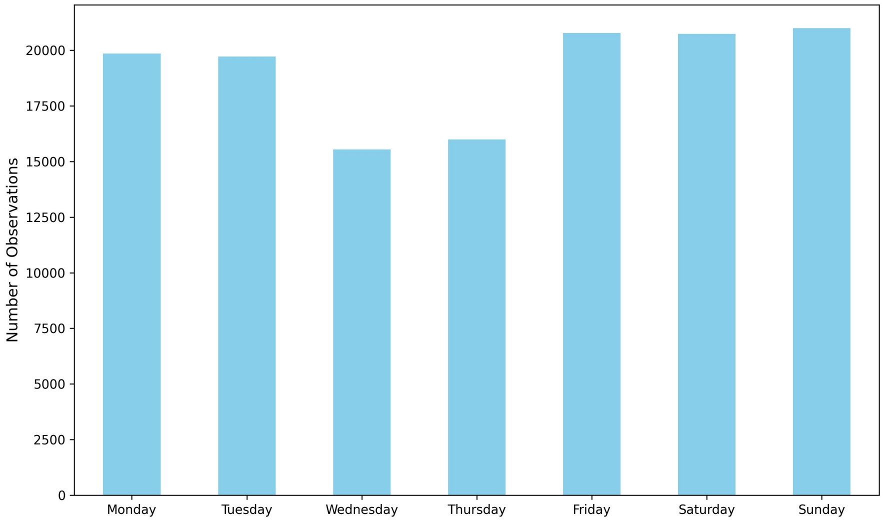

Figure 3: Observations per day

Figure 4 provides insight into the distribution of languages in the sample. There are 113,210 observations in English and 20,432 in a foreign language, predominantly Italian (4,730), French (3,379), and German (3,074).

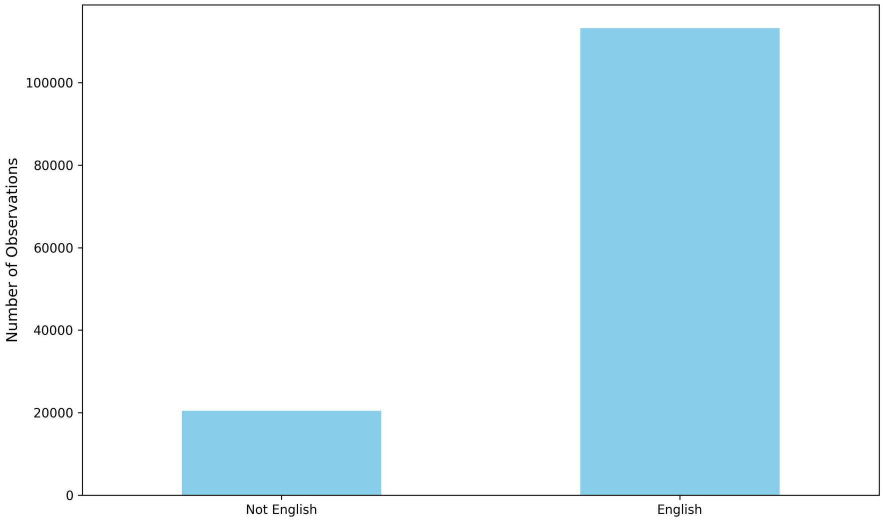

Figure 4: Observations in English vs non English

Because of GDPR concerns, pictures or text from the sample are not displayed in the thesis. Nevertheless, the sample is a random sample from Instagram influencers.

## 5.3 *Data pre-processing*

### 5.3.1 *Image feature extraction*

Taking inspiration from the state-of-the-art modeling discussed in section 3.3, I decide to use ResNet50 pre-trained on ImageNet (He et al., 2016) to extract image features. As previously discussed, this is one of the best models for feature extraction in this field according to (Ding et al., 2019; Riis et al., 2021). 2,048 features are extracted for each image and concatenated to the rest of its respective post's information.

### 5.3.2 *Text feature extraction*

Following the state-of-the-art modeling discussed in section 3.3, a roBERTa model pre-trained on Twitter data (Barbieri et al., 2020) extracts features from the text accompanying each picture. A BERT model uses a Transformer, an attention mechanism that learns contextual relations between words. This type of model captures the context of a word based on its surroundings. BERT models process text through multiple layers. In the model I used, I extracted features by taking the average of the last layer. This allows the extraction of the most abstract representations of the text data and is better at representing semantic information. 748 features are obtained from the captions from the captions which are later concatenated with the rest of the post's metadata.

### 5.3.3 *Dimensionality Reduction and Feature Engineering*

Some features of the dataset are deleted because of potential biasing or irrelevance. Starting with irrelevant features that are deleted, phone, email, URL, owner ID, post ID, JSON files, the dimensions of an image, media preview information. Moving on to features that could bias the algorithm, the username, along with other user identifiers, are deleted from the dataset, such that the data would not make predictions based on a specific user.

Next, feature creation. This includes languages and timestamps. UNIX timestamps are transformed into year, month, and day features. At the same time, not all captions in the dataset are in English. Google Translates API is used to detect the language of each caption. This creates a language feature, which helps create the English versus non-English visualizations under section 5.2.

Finally, sections 2 , 3.3, and 4 hint at a novel definition of popularity. I define a novel definition of popularity using the engagement rate as a threshold to discern between popular and non-popular. First, the *engagement rage* feature is created.

$$Engagement = (\frac{Likes + Comments}{Followers}) \tag{1}$$

This feature gives us the engagement rate of each post in the data. Second, I take the median engagement rate of the distribution of engagement rates and set everything above the median as popular and everything below the median as not popular. As explained in section 5.4, the median engagement rate is taken from each train, validation, and test, and everything above this benchmark is defined as popular. On average, most median engagement rates are around 3.5%, which aligns with what blogs and popular articles say is a very good engagement rate, as discussed under section 3.1.2

### 5.3.4 *One-hot encoding and Feature scaling*

The features year, month, hour, day of the week, and category are all one-hot encoded. These are decided to be one-hot encoded as they are categorical variables that any algorithm should not interpret as numerical or have an ordinal value.

The features, including the number of followers, followees, and posts, are scaled with Scikit Learn's standard scaler (Pedregosa et al., 2018). The features are scaled for several reasons. First, dealing with very skewed data for these features helps mitigate the impact of outliers. Second, it helps the algorithm converge faster since all values are closer. And third, it allows for consistency and equalization across the features, preventing them from disproportionately influencing the model's outcome. The features are scaled after the data split and according to their own set (train, test, validation) to avoid data leakages.

## 5.4 *Data Split and Monte Carlo Cross Validation*

The data is split by users, with the intention of not having the same user show up on either training, validation, or testing simultaneously. This aims to avoid potential data leakage and ensure the model results are generalizable. Taking inspiration from Zhang et al. (2018), I use a split of 80% training, 10% validation, and 10% testing. Splitting by users means that test, train, and validation sets will always vary in size slightly across different seeds of splits, as some users might have more posts in the dataset than others. Most samples are approximately the same size, nevertheless.

Since a popular post is defined as anything above the median engagement rate, as discussed under section 5.3.3, the target variable is defined after each data split to avoid data leakages. Therefore, each train, test, and validation set has its measure of popularity, yet each set had an even split of 50-50 for popularity.

Since the study aims to create a novel baseline with robust and generalizable capabilities, I used Monte Carlo Cross-Validation. This randomly splits the dataset into training and testing sets multiple times, using different seeds to split the data. This randomness introduces variability, allowing a more robust estimate of the model's performance. The split is performed 7 times, with 7 different seeds for each model.

## 5.5 *Baseline*

The baseline is decided by assuming the model would guess the majority class. As discussed under sections 5.3.3 and 5.4, the target variable is defined after each data split. Given that the data is split by the median, the target variable is binary, with each category having an even split of 50-50. This means the majority class is 50%, providing an intuitive baseline. If the model would guess the majority class, it would accurately guess it with a 50% accuracy.

## 5.6 *Algorithms*

As explored and concluded under section 3.3, the state-of-the-art modeling uses Deep Neural Networks, Extreme Gradient Boosting (XGBoost), and Light Gradient Boosting Machines (LightGBM). In this section I discuss the core mechanisms behind these algorithms and why these perform well for predicting the popularity of an Instagram post.

### 5.6.1 *Deep Neural Network*

Deep Neural Networks are powerful machine learning models with multiple layers of interconnected nodes. They can capture complex patterns in high-dimensional data. Each layer of a Neural Network adds complexity and abstraction by learning hierarchical representations (LeCun et al., 2015). Given the depth and complexity of a neural network, it makes it effective at capturing non-linear relations in the data. Combining these characteristics of how a Neural Network works with the multimodal and high-dimensional characteristics of the data representing an Instagram post makes a Neural Network a strong candidate for predicting the popularity of an Instagram post. In table 2, you can find the architecture and hyperparameters chosen for the Neural Network used in this study, inspired by Gayberi and Oguducu (2019).

Table 2: Hyperparameters and Architecture of the Deep Neural Network

| Parameter | Value |
|---|---|
| Input Layer | 64 neurons, ReLU activation |
| Dropout rate | 0.5 |
| Hidden Layer | 32 neurons, ReLU activation |
| Output Layer | 1 neuron, Sigmoid activation |
| Optimizer | Adam |
| Learning Rate | 0.001 |
| Loss Function | Binary Crossentropy |
| Early Stopping Patience | 20 |
| Batch Size | 128 |
| Epochs | 500 |

### 5.6.2 *Light Gradient Boosting Algorithm*

LightGBM is a gradient-boosting ensemble method based on decision trees. It can be used for both classification and regression problems. It is a histogram-based method in which data is bucketed into bins that use the distribution of the histogram. The bins make data predictions instead of each data point (Ke et al., 2017). Given the nature of Instagram data, where you have many dimensions from image and text features, in addition to metadata features, and the complex interactions among all these features, LightGBM offers a robust yet efficient way of training large and high dimensional datasets like the one at hand and making predictions with this data. Table 3 shows the hyperparameters for this algorithm.

Table 3: Hyperparameters Used in the LightGBM Model

| Hyperparameter | Value |
|---|---|
| Objective | Binary |
| Metric | Binary error |
| Alpha | 0.9 |
| Learning Rate | 0.1 |
| Number of Boost Rounds | 1000 |
| Early Stopping Rounds | 20 |
| Epoch | 500 |

### 5.6.3 *Extreme Gradient Boosting Algorithm*

Extreme Gradient Boosting (XGBoost) is an advanced implementation of gradient boosting that is efficient and effective at handling large-scale and high-dimensional data. XGBoost improves standard boosting techniques by introducing a more regularized model formalization to control over-fitting, making it highly robust, especially for complex regression and classification tasks. It uses gradient boosting frameworks at its core, building models stage-wise and optimizing arbitrary differentiable loss functions (T. Chen & Guestrin, 2016). In the context of Instagram data, XGBoost excels by effectively managing sparse data from various sources, such as text, images, and metadata. The algorithm can capture complex and non-linear relationships in such data, posing as a strong model to predict the popularity of an Instagram post. Table 4 shows the hyperparameters for the XGBoost algorithm.

Table 4: Hyperparameters Used in the XGBoost Model

| Hyperparameter | Value |
|---|---|
| Objective | Binary |
| Metric | Binary error |
| Alpha | 0.9 |
| Learning Rate | 0.1 |
| Number of Boost Rounds | 1000 |
| Early Stopping Rounds | 20 |
| Epoch | 500 |

Given that I use both LightGBM and XGBoost, it is worth highlighting some key differences. These models mainly differ in their tree-building approaches and handling of data scales. XGBoost builds trees level-wise, which can be more resource-intensive, making it ideal for smaller datasets where fine-grained control over model complexity is crucial. LightGBM, however, uses a leaf-wise growth strategy and histogram-based memory optimization, allowing it to process large datasets more efficiently and faster, making it suitable for scenarios where computational performance is a priority.

### 5.7 *Feature Ablation and Observation Removal*

I perform feature ablation to provide a full picture of each algorithm's performance across different modalities, such as text, images, metadata, and a combination. The goal is to show how removing features impacts a model's performance. This methodology answers sub-RQ1.1.

Moreover, I use the algorithms and feature ablation described above to compare the algorithm's performance between including all features (i.e., all languages included) versus only observations in which only English captions are included. This answers sub-RQ1.2. Additionally, comparing the performance across different algorithms with feature ablation and comparing the performance of English versus non-English features answers RQ1.

## 5.8 *Performance and Error Analysis*

Taking inspiration from Carta et al. (2020), who also uses a binary classification for their research, I use accuracy as the primary performance measure. Additionally, for the best-performing models for each of the three algorithms, I provide a complete picture of their performance (Precision, Recall, F1-Score, Accuracy, Macro Average, and Weighted Average), along with a confusion matrix on one of the testing sets.

The results are shown in a 95% confidence interval to demonstrate their robustness. Given that the Monte Carlo Cross-Validation split is performed under seven different seeds, this provided seven different accuracy measures for the testing sets. Since these are not enough observations to perform a 95% confidence, where you need 30 observations for the Central Limit Theorem to hold, I decide to use a bootstrap confidence interval. Bootstrap confidence intervals are constructed by sampling 10,000 random values, with replacement, from the seven accuracy scores obtained from the Monte Carlo Cross-Validation (Xu & Liang, 2001). After this, the 2.5 and 97.5 percentile samples are chosen from this random sampling, creating robust confidence intervals for the proposed algorithms' performance.

For the error analysis, I use a confusion matrix to show the performance of the best model. I also show a distribution of the predicted probabilities and the error rate for each of the predicted values. This answers RQ 3.

## 5.9 *Explainable AI: SHAP values*

The "SHAPley Additive exPlanations" (SHAP values). SHAP values are a form of explainable AI used to show how different features affect the prediction of results, how the top features influence predictions, and provide intuition for model users to feel at ease. These values can help measure the input features' contribution to individual predictions (Lundberg & Lee, 2017b). I use feature importance, waterfall, and summary plot visualizations to explain how the model makes decisions and answer RQ 2.

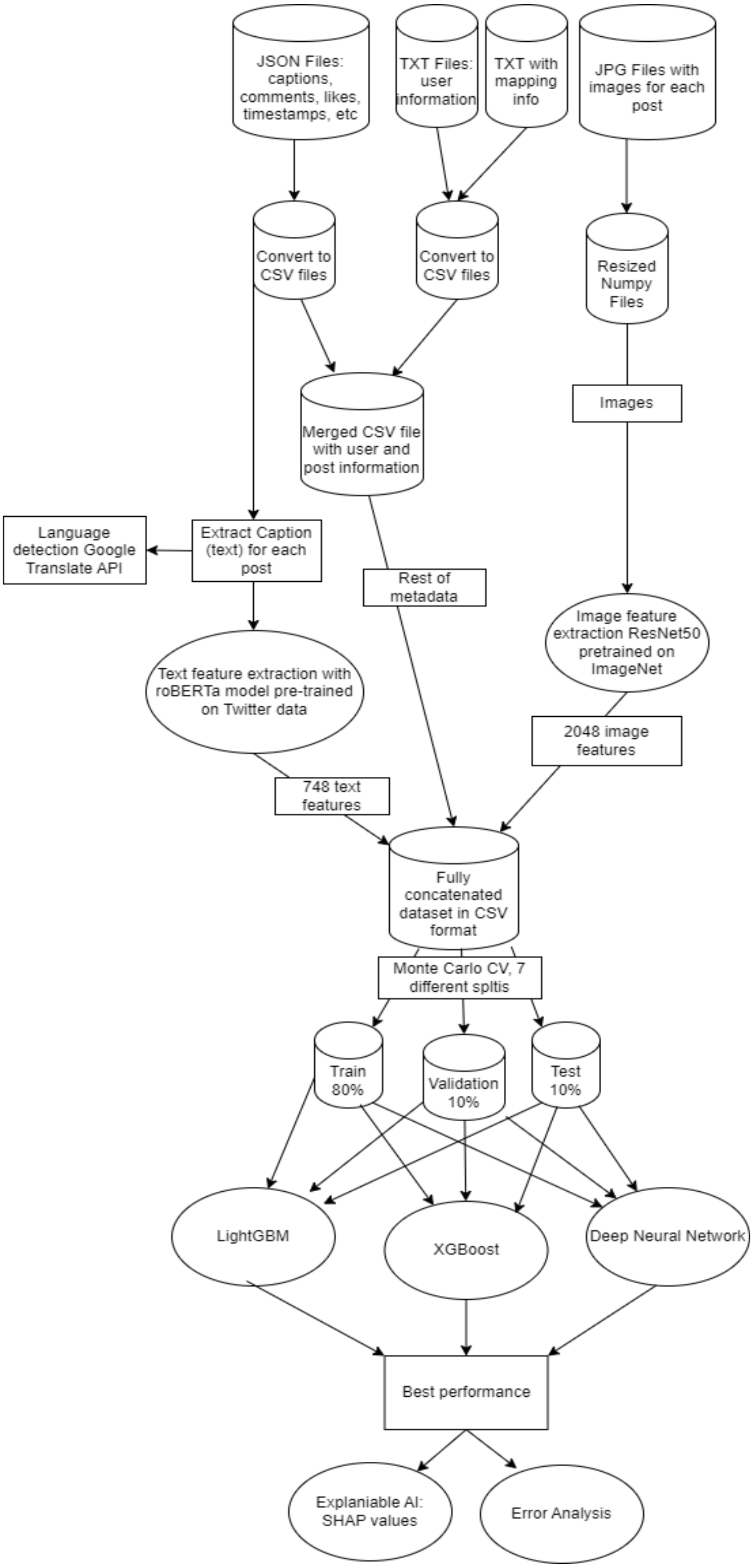
JSON Files: captions, comments, likes, timestamps, etc
TXT Files: user information
TXT with mapping info
JPG Files with images for each post
Convert to CSV files
Convert to CSV files
Resized Numpy Files
Merged CSV file with user and post information
Images
Language detection Google Translate API
Extract Caption (text) for each post
Rest of metadata
Image feature extraction ResNet50 pretrained on ImageNet
Text feature extraction with roBERTa model pre-trained on Twitter data
2048 image features
748 text features
Fully concatenated dataset in CSV format
Monte Carlo CV, 7 different spltis
Train 80%
Validation 10%
Test 10%
LightGBM
XGBoost
Deep Neural Network
Best performance
Explaniable AI: SHAP values
Error Analysis

Figure 5: Research pipeline

# 6 RESULTS

## 6.1 *Regular results*

First, it is important to understand what each row in tables 5 and 7 refer to. 'Images only' refers only to using the features extracted from the ResNet50 model to predict popularity. Similarly, 'Text only' only uses the features extracted from a post's caption with the roBERTa model pretrained on Twitter data. 'Relevant Metadata' refers to only using a user's information on follower and followee count and the number of posts the user has made. The difference between 'Images + Text + Metadata' and 'All information available' is that the former only uses the images, text, and relevant metadata (followers, followees, and number of posts), whereas 'All information available' includes all the information under 'Images + Text + Metadata' plus the month, day of the week, hour, Image count in the post, whether the post includes a video, and whether it is sponsored.

In table 5, you can find the results for the different algorithms tested under section 5. The values are shown in Bootstrap Confidence Intervals and accuracies as discussed under 5.8.

Table 5: Comparison of Accuracies across Different Feature Sets

| Feature Set | XGBoost | LGBM | Deep Neural Network |
|---|---|---|---|
| Baseline | 50% | 50% | 50% |
| Images only | [56.15% - 57.10%] | [55.50% - 57.77%] | [56.07% - 57.94%] |
| Text only | [58.22% - 59.08%] | [58.10% - 59.79%] | [58.96% - 59.85%] |
| Relevant Metadata | [62.58% - 64.19%] | [62.83% - 65.27%] | [61.07% - 63.48%] |
| Images + Text | [59.44% - 60.33%] | [59.83% - 61.19%] | [58.19% - 60.66%] |
| Images + Text + Metadata | [67.50% - 68.52%] | [67.29% - 68.71%] | [63.24% - 65.50%] |
| All information available | [68.45% - 69.35%] | [68.19% - 69.74%] | [63.73% - 66.33%] |

Per table 5, all feature sets predict above the baseline of 50%. Using only images to predict popularity provides a confidence interval of about [55.50% - 57.77%]. Using text to make predictions has a higher accuracy rate than using only images, with no confidence intervals overlapping between 'text only' and 'images only.' Using only 'Relevant Metadata' (followers, followers, and posts count) further increases the accuracy across all algorithms, with Deep Neural Networks (DNN) performing worst with a confidence interval of [61.07% - 63.48%] and LGBM performing best in this category with a confidence interval of [62.83% - 65.27%]. Only using followers, followees, and number of posts outperforms using images or text, hinting that a user's characteristics are more important to predict their popularity above the median than the published content. Using images and text provides higher confidence intervals than only using text or images, yet the accuracy is still close to only using text features. Using Image, text, and followers, followees, and posts features provides the second-best performance, with the best performing algorithm for these feature sets being LGBM with a confidence interval of [67.29% - 68.71%] (though closely overlapping with XGBoost). The best performance across all algorithms can be achieved using all information available. All three algorithms perform best here against all other feature sets. The best performing algorithm is LGBM with a confidence interval of [68.19% - 69.74%] with XGBoost closely overlapping, and DNN having the worst performance of the three with this feature set with a confidence interval of [63.73% - 66.33%]. Table 6 shows all performance metrics for the best-performing model, which is LGBM, using all features. The full performance metrics for the best-performing XGBoost and DNN are found in Appendix A.

Table 6: Classification Performance Metrics for LGBM

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Low | 0.67 | 0.70 | 0.68 | 6648 |
| High | 0.68 | 0.66 | 0.67 | 6648 |
| **Accuracy** | | | **0.68** | **13296** |
| **Macro Avg** | **0.68** | **0.68** | **0.68** | **13296** |
| **Weighted Avg** | **0.68** | **0.68** | **0.68** | **13296** |

## 6.2 *Results for posts only made in English*

The results for this section include models trained and evaluated only in English observations. All non-English observations are removed. For completeness and comparison, I include features that should (mostly) not be affected by language, such as images and relevant metadata.

Table 7: Observations just in English: Comparison of Algorithms across Different Feature Sets

| Feature Set | XGBoost | LGBM | Deep Neural Network |
|---|---|---|---|
| Baseline | 50% | 50% | 50% |
| Images only | [56.02% - 56.66%] | [56.54% - 57.35%] | [56.03% - 56.31%] |
| Text only | [58.33% - 59.08%] | [58.01% - 59.00%] | [58.14% - 60.24%] |
| Relevant Metadata | [64.07% - 65.68%] | [64.15% - 65.81%] | [62.70% - 64.45%] |
| Images + Text | [59.08% - 60.27%] | [59.52% - 60.51%] | [58.69% - 59.33%] |
| Images + Text + Metadata | [68.31% - 69.14%] | [67.84% - 69.06%] | [64.48% - 65.11%] |
| All information available | [69.07% - 70.18%] | [68.51% - 70.62%] | [64.02% - 65.42%] |

The results in table 7 are similar to those in table 5. One would expect that using only the text features to predict the popularity level would increase its performance by using only observations in English, but across all features, all confidence intervals in table 7 and 5 overlap. While table 8 shows better performance than table 6, the confidence intervals between all languages and only English still overlap. Therefore, it can be concluded that there is no difference between using only English observations and using all languages in the sample.

Table 8: Classification Performance For LGBM only English

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Low | 0.70 | 0.71 | 0.70 | 5816 |
| High | 0.70 | 0.69 | 0.70 | 5816 |
| **Accuracy** | | | **0.70** | **11632** |
| **Macro Avg** | **0.70** | **0.70** | **0.70** | **11632** |
| **Weighted Avg** | **0.70** | **0.70** | **0.70** | **11632** |

## 6.3 *Error Analysis*

Figure 6 shows the confusion matrix for the best-performing models under table 5, the Light Gradient Boosting Machine (LGBM) using all features. Under Appendix B, you can also find the confusion matrices and the rest of the visualizations of this section for the best-performing XGBoost and Deep Neural Network models.

At first glance, the model can discern between low and high popularity at a relatively balanced rate. The model is more accurate at predicting low-popularity images than high-popularity images. However, it is worth noting that this might differ across data splits while still staying among the confidence intervals shown in table 5.

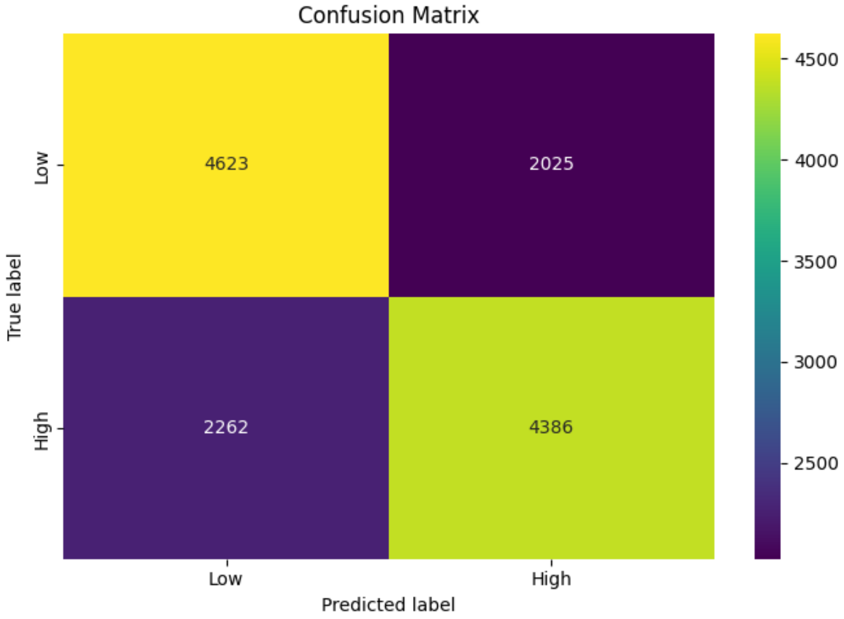

Figure 6: LGBM Confusion Matrix

A model like LGBM that predicts a binary variable does not explicitly predict a binary outcome but rather a probability of the input's potential outcome. The model creates a continuous probability between 0 and 1, indicating the potential outcome of an input. Figure 7 shows the distribution of predicted probability values by the LGBM model. The confusion matrix above defines as popular any value that the LGBM model predicted above 0.5 as high popularity and everything below 0.5 as low popularity.

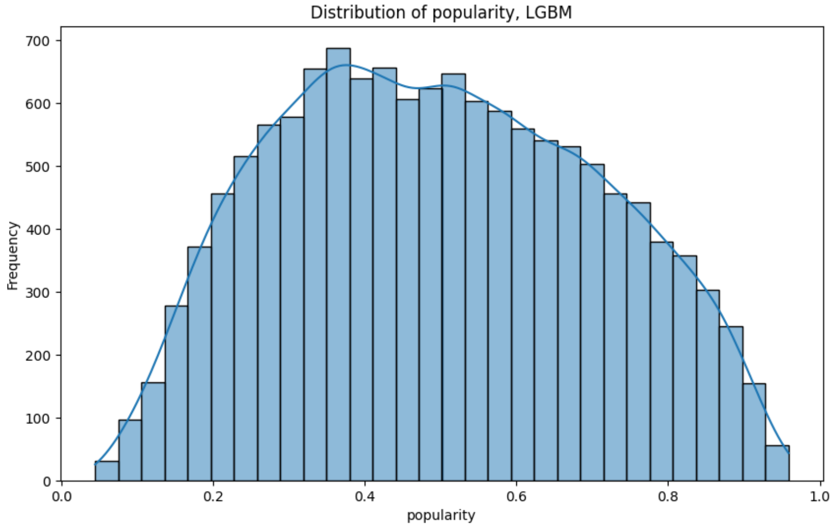

Figure 7: LGBM Predicted Values Distribution

Having knowledge of the distribution of predicted probabilities from the best-performing LGBM model, we can construct the error rate for each of these bins of probabilities. Figure 8 shows the error rate per each predicted value. Intuitively, values around the threshold of popularity (0.5) are the ones which the model has the highest error rate for (around 45% error rate), whereas the closer the probability is to 0 or 1, the lower the error rate is, up to the point where at the tails of the probabilities, the error rate is about 5%. This shows that the error rates depend on the predicted probability the model assigns.

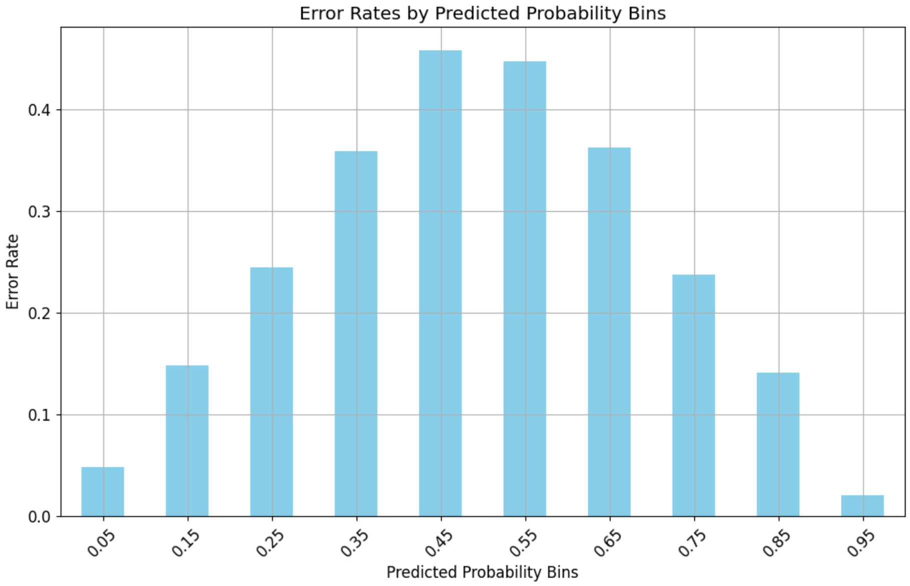

Figure 8: LGBM Error Rate per predicted value

### 6.4 *Explainable AI: SHAP values*

Ahead, I explore the results from the SHAP values from the best performing model, LGBM using all available information (all languages). The feature importance plot is represented by Figure 9. The features on the y-axis are ordered with the most predictive value at the top of the figure. Per this figure, the number of posts and followers derive the model's most predictive power. Many text and image features do not have a strong individual predictive power but do as a whole. There are 2,048 image features and 748 for text.

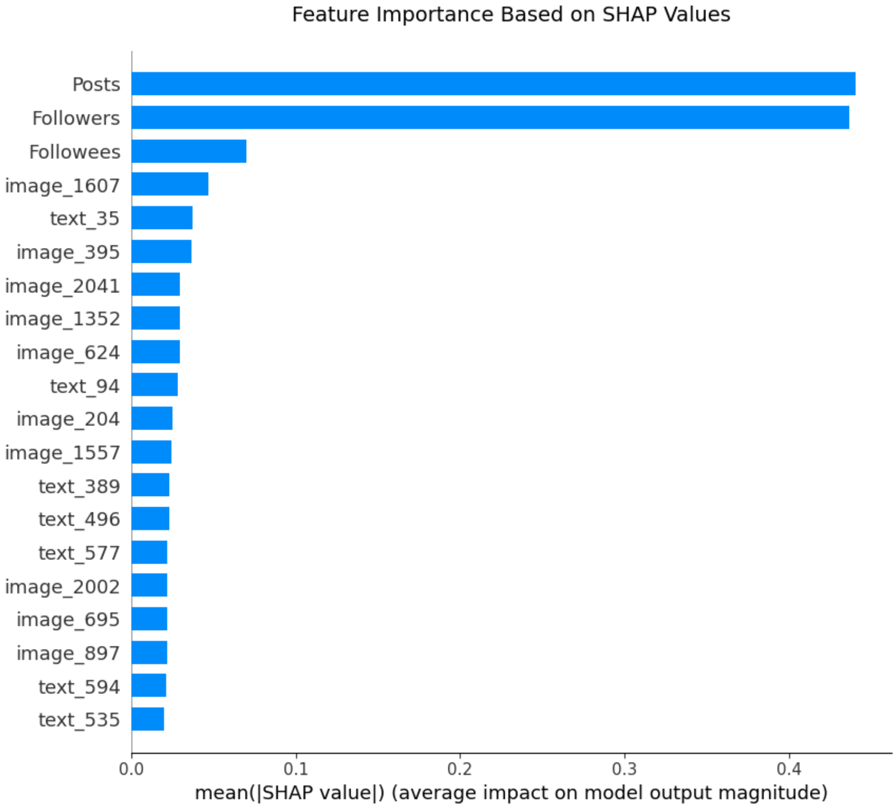

Figure 9: Feature Importnace

A waterfall plot helps understand how the makes predictions for individual posts (Lundberg & Lee, 2017a). For instance, Figure 10 shows the SHAP values and how these influence a prediction. The X-axis represents log-odd values, which can be transformed into probabilities. On the X-Axis, the value E[f(x)] is the baseline value, with a log-odds value of 0.008, which translates to a probability of 50%. At the top right is the individual log-odds value for this prediction, which is 1.205. This translates to a predicted popularity probability of 76.94%. A value for followers of -0.099 (remember that this feature is scaled) increases the probability that the model will predict a popular picture. Similarly, fewer posts also contribute to a higher probability of predicting popularity. Further, combining images and text features also increases the probability of predicting popularity but to a smaller level.

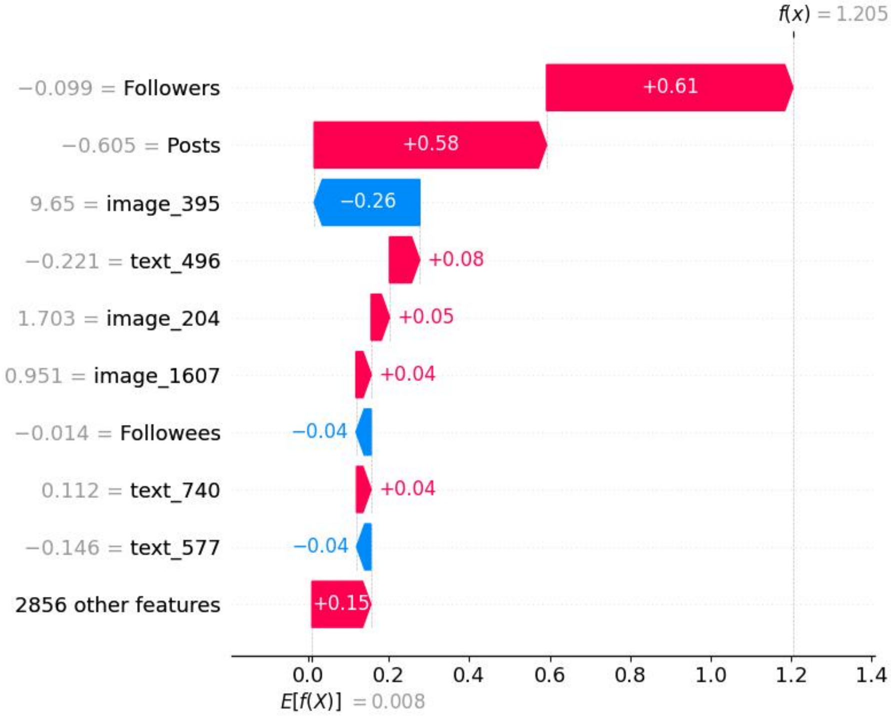

Figure 10: SHAP: waterfall for the first observation of the test set

Another example using a waterfall plot can be observed in Figure 11. In this case, this post's predicted probability of popularity is 28.45% (log odds of -0.922). Similar to Figure 10, their relatively low number of followers contributes to a higher probability of popularity for the prediction, but the combination of text and image features drastically pushes down the log odds and, therefore, the probability of this post being predicted as popular.

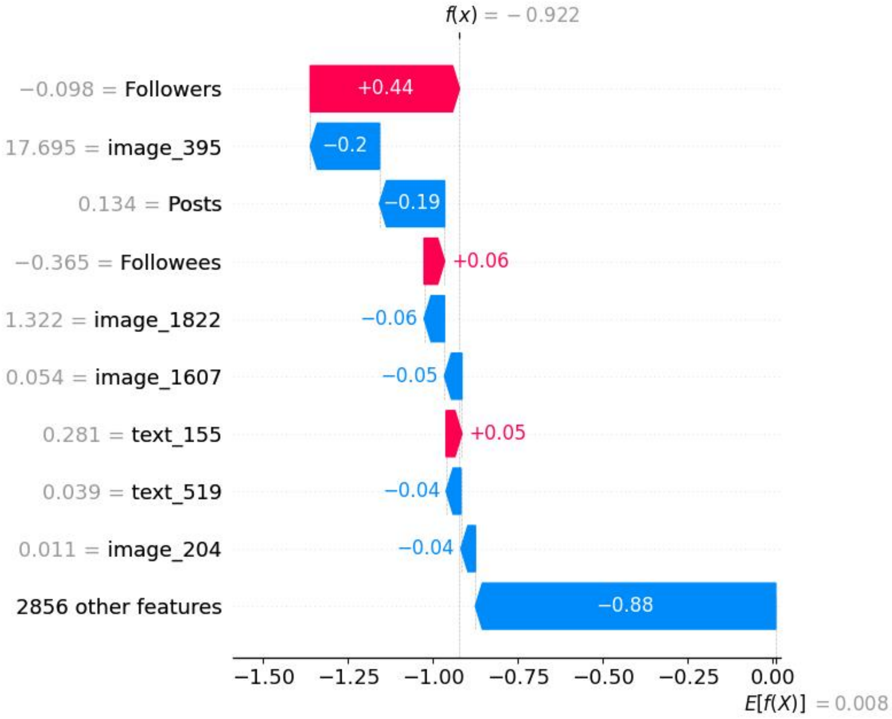

Figure 11: SHAP: waterfall for the fifth observation of the test set

Figure 12 is a summary plot, a dot chart that visualizes the directionality impact of features. Along the y-axis, features are ordered according to importance, with the most important features at the top. The x-axis represents the contribution to the model's score, with 0 in the middle, and small dots represent each prediction. Dots to the left of 0 indicate a lower probability of the model predicting high popularity, and dots to the right of 0 represent a higher probability of predicting high popularity. Blue colors represent lower feature values, and red represents higher ones. For instance, the number of posts ('Posts') is the most important feature, with higher values leading to more negative predictions (less popular) and vice-versa. The second most important feature is the number of followers, for which higher values contribute to a less popular post, whereas fewer followers contribute to a more popular prediction. While this might seem counter-intuitive, it is important to remember the definition of popularity: likes plus comments over followers. It is more likely that a user with several million followers does not have as much engagement with their posts as someone with fewer but more active followers. Since there are more than 2,800 features, not all of them are shown in this visualization, but image and text features appear to have subtler individual impacts, suggesting potential interaction effects that enhance their predictive power when combined (Lundberg & Lee, 2017a).

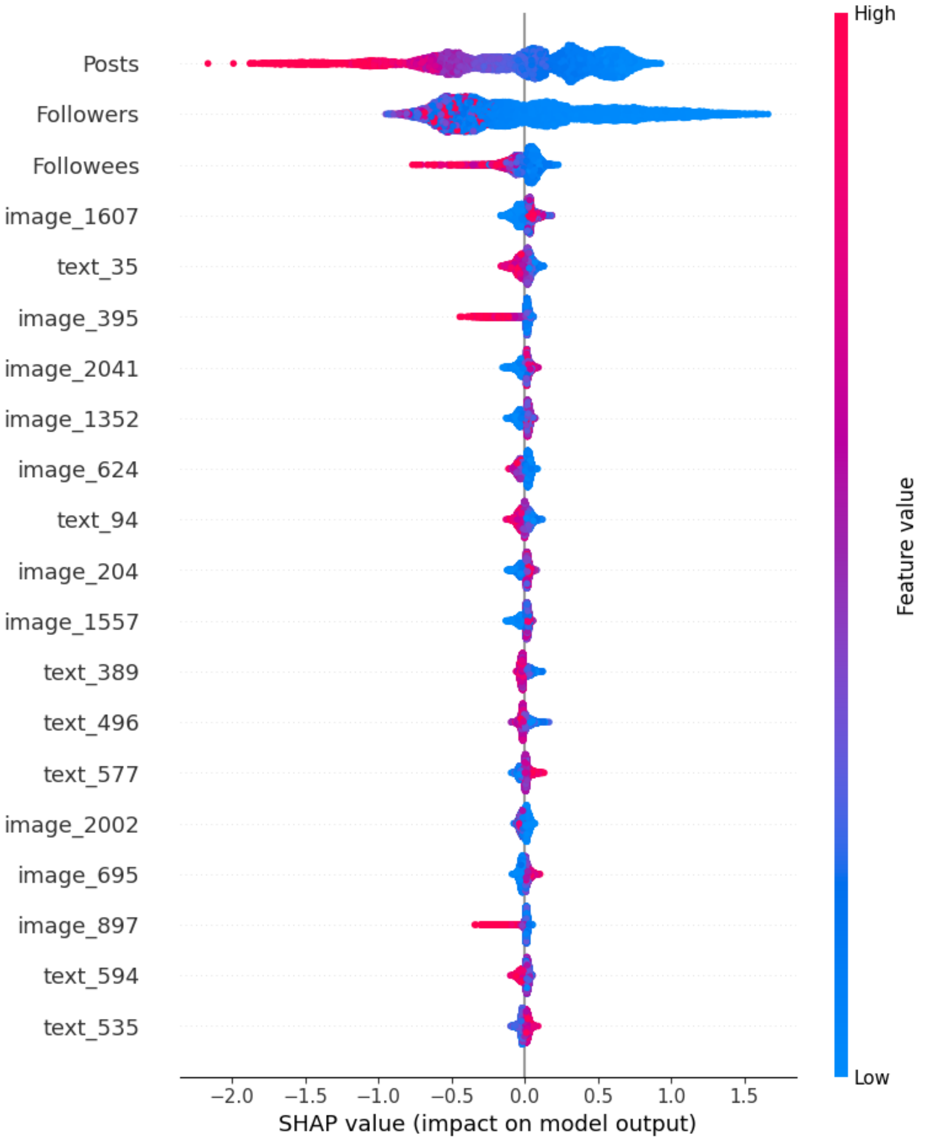

Figure 12: SHAP Values Summary Plot

To summarise, the number of posts and followers has the most predictive value according to SHAP values. Individual image and text features have small predictive contributions, but these combinations greatly help the model make sense of the data. On average, lower counts of posts and followers help predict higher popularity (higher engagement rate), while high numbers of posts and followers help predict less popular posts.

# 7 DISCUSSION

## 7.1 *Summary and discussion of results*

Firstly, I touch on the comparison of the performances of different algorithms. Out of the three tested models, LightGBM and XGBoost both outperform Deep Neural Networks (DNN). Using bootstrap confidence intervals, I provide a robust range of potential accuracies for each model. The best-performing models, including all available features describing a post for a user before making a post, have an accuracy of about 69%, with a low range of 68% and a high range of 70%. The model's performance worsens by removing features and keeping only image, text, metadata, or a combination of these, yet it still performs above the baseline. Including only the most relevant metadata of a post can already predict the popularity of a post to levels in the lower range of 62% and the high range of 65%, suggesting that a user's characteristics already provide plenty of predictive power. As other literature has recorded, for example, Carta et al. (2020), Gayberi and Oguducu (2019), Gupta et al. (2020), Hsu et al. (2019), and Zhang et al. (2018) demonstrated that LightGBM and/or XGBoost performed best at predicting the popularity of posts. While using different measurement scales and methodologies, the average accuracy of 69% of the best model is close to the performance of Carta et al. (2020) at 67.50% and Zhang et al. (2018) at 71.19%.

Secondly, I test whether the algorithm would perform better if only English features are to be included. As previously mentioned, no literature has touched on the popularity of social media posts and how languages might play a role. The roBERTa pre-trained model in this study is for English text. Therefore, it is sensible to predict an increase in accuracy if non-English observations are not included. However, this is not the case, as the accuracy for predicting popularity across all combinations of feature sets is virtually the same as keeping all languages in the sample. All confidence intervals between using only English observations and using all languages overlapped, showing that dropping non-English observations did not affect the model's predictive power. Therefore, it is concluded that including foreign languages in the feature extraction process for this pre-trained roBERTa model on Twitter data does not affect the predictive power of this model.

Thirdly, the error analysis is a useful tool for potential model users and fills an unexplored gap in the literature. While the global accuracy and F1 scores are about 68% for the best-performing model, this model does not have the same error rate for all predictions it makes. If the model predicts a probability of popularity closer to 0.5, it is more likely that the model will be wrong. This is intuitive, as the threshold for defining the binary classification is 0.5. As the model is more unsure about the popularity of a post, it is more likely to assign a value closer to 0.5, increasing the chance of an incorrect prediction. However, the error rate drops as the predicted probability approaches 0 or 1 (i.e., the model being more certain of the post being popular or not). The error rate for predicted probabilities above 0.9 or below 0.1 drops to under 10%. Having this knowledge can significantly increase the value of this algorithm. For example, picture a user who is fairly confident about the image and text of a post they would like to make but would like to forecast its popularity. If the model would give the user a value of 0.48, this would mean that the model predicts the post not to be popular. However, because it is known that the error rate for this model for values close to 0.5 is quite high, the user could safely discard the prediction and make the post. However, if the model had predicted a value of 0.03, then the user might want to reconsider and understand what is causing the model to forecast this prediction, as the error rate for this type of value is quite low.

Lastly, SHAP values reveal the number of posts and followers are the most important features. After this, individual text and image features have smaller predictive values, but there are 2,048 image features and 748 text features. The SHAP values show that a higher number of posts by a user hints at a less popular prediction, and the smaller number of followers a user has leads to a more popular prediction. At first, this might seem counter-intuitive. One would imagine more followers being more popular. However, popularity is defined as likes and comments over the number of followers (as equation 1 shows). Users with fewer followers likely have more engaging followers, prompting these to have more popular posts. As shown in section 6.3 and figures 10 and 11, waterfall visualizations for SHAP values can provide useful information about individual predictions. Combining domain knowledge, the prediction by the model, the error analysis, and the SHAP waterfall visualizations can be a robust and data-driven approach to deciding whether to make a post. For example, a user makes a prediction with the model. Using the predicted probability, the user knows the error rate for this probability value. With the SHAP waterfall visualizations, the user can know whether the image or text features are affecting the prediction or if other features that the user cannot change are determining the prediction. Combining the model prediction, the error rates, SHAP values, and domain knowledge can be a useful tool to predict whether a user should make a post.

## 7.2 *Societal Impact*

There are several contributions this model brings to society, though two shine the brightest. First, this model provides insights for better creation of popularity prediction. It uses a novel specification of popularity. It also demonstrates that the language in which the BERT model processes the data does not affect the model's performance. This model also shows that image and text features add predictive value to a user's information. These are novel aspects that have not been explored in the literature, but they can help further enhance and add knowledge to this type of prediction in the future.

Second, the combination of domain knowledge, the model, awareness of its error patterns, and the use of SHAP values, more specifically, waterfall plots, can provide a powerful tool for forecasting the popularity of a post on Instagram. This can be useful for users like marketers, influencers, social media managers, small entrepreneurs, and more. An example is the small entrepreneur and social media manager "Cavi Productions" (@caviproduction). They specialize in taking pictures of small to medium-sized businesses, like restaurants or entrepreneurs, handling their social media sites, creating captions for these posts, and generating organic engagement on different sites. Cavi Productions handles up to 10 different accounts, each representing a small restaurant or retail store that wants someone to post on social media for them. An entrepreneurial project like this could benefit from forecasting the popularity of each post since they need to make so many of these for different accounts. This shows that deciding what to publish on Instagram can be a data-driven approach to attempt to achieve a higher engagement rate on the platform.

## 7.3 *Limitations and future directions*

This model possesses limitations that are important for both users and future directions the study could take. First, the data used to train the model contains only one picture per post, while in reality, an Instagram post may have more than one picture. It would be worthwhile to research whether posts with multiple pictures have different predictive values for popularity than using only posts with a single image like this study.

Second, while this thesis uses state-of-the-art modeling for image and text feature extraction, it would be worthwhile to test out even more of these feature extracting methods, like using an Inception-V3, or EfficientNet-B6 like Ding et al. (2019) did to test whether these feature extraction methods would have a better performance with the novel specification of popularity proposed here. This could also be the case for text feature extraction, using more robust methods like Zhang et al. (2018) using an LSTM or LDA.

Third, exploring other explainable AI techniques that could explain image and text characteristics that cause higher or lower engagement rates would be a useful research avenue. While SHAP values tell us how individual features contribute to the model, the abstract nature of image and text features makes it difficult to have an intuition of how different image features might affect the model's predictive values.

And fourthly, it would be worthwhile to replicate this model with less heterogeneous data. As section 5.2 showed, there can be quite a wide range in type of users, with some users having many more followers than others. Having a more homogenous mix of users, like users with only a certain range of followers could help the model pick up more nuanced details of images and types of captions that each user makes to be able to understand at a deeper level how these users could post content that will be more engaging for their follower base.

# 8 CONCLUSION

This thesis uses Instagram data at the post level to predict its popularity using images, text, and user information. By creating and understanding the model, it is possible to answer the research questions posed under section 4. Ahead, I answer each of them one by one.

- **RQ1: Out of the models tested, which model performs best in predicting the popularity of an Instagram post with the novel definition of popularity?**

The best-performing models, as shown under Table 5, are both the LightGBM (LGBM) and XGBoosst models when feeding them all information about a post available to a user publishing it. They perform [68.19%—69.74%] in a bootstrap confidence interval. This aligns with what other literature Carta et al. (2020) and Zhang et al. (2018) found to be the best model.

- **Sub-RQ1.1: Do image and text features add predictive value to a user's metadata with this new measure of popularity?**

Yes, adding images and text to a user's information improves accuracy. For example, the LightGBM model, including only user information, provided a confidence interval of [62.83% - 65.27%] while including images, text, user information, plus additional information of the post, produced an accuracy of [68.19% - 69.74%].

- **Sub-RQ1.2: After extracting features from an English-based pre-trained BERT model on both English and non-English features, does the performance of the algorithm increase if non-English observations are excluded?**

No, the performance of any model is not improved by removing text features. You can observe this by comparing the results in tables 5 and 6.2. While the performances in table 6.2 are slightly better when using all features, when predicting popularity only with text, the performance between using only English observations and non-English observations is almost identical, hinting at language not having an effect in making accurate predictions, even if the BERT model used is for the English language.

- **RQ2: Which features are most important, and what is their relationship with predictions?**

SHAP values reveal that the number of posts and followers are the most influential predictors of popularity. Fewer posts and followers are linked to more engaging posts, which increases the likelihood of predicting a post's popularity. Additionally, individual text and image features have small effects on prediction, but they contribute to the prediction of popularity as a whole.

- **RQ3: Do errors of the best-performing model depend on the estimated probability the model gives each prediction?**

The best-performing model has, on average, an error rate of about 31%, but the error rate is not the same across all predictions. A model that outputs a binary classification predicts a probability between 0 and 1 to assign it to a class. For the best model, the closer the prediction is to 0.5, the higher the error rate, at about 45%. The closer the prediction is to 0 or 1, the lower the error rate is, going as low as 5% for predictions above 0.95 or below 0.05. Therefore, the errors of the best-performing model depend on the estimated probability the model gives to each prediction.

## REFERENCES

Aslam, S. (2024, February). https://www.omnicoreagency.com/instagram-statistics/#:~:text=Around%201.3%20billion%20photos%20are,use%20explore%20page%20every%20month.

Barbieri, F., Camacho-Collados, J., Espinosa Anke, L., & Neves, L. (2020). TweetEval: Unified benchmark and comparative evaluation for tweet classification. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1644–1650. https://doi.org/10.18653/v1/2020.findings-emnlp.148

Barinka, A. (2022). Meta's instagram users reach 2 billion, closing in on facebook [Accessed: 2024-02-16]. https://www.bloomberg.com/news/articles/2022-10-26/meta-s-instagram-users-reach-2-billion-closing-in-on-facebook?embedded-checkout=true

Barrett, T., Dowle, M., Srinivasan, A., Gorecki, J., Chirico, M., & Hocking, T. (2024). *Data.table: Extension of 'data.frame'* [R package version 1.15.99, https://Rdatatable.gitlab.io/data.table, https://github.com/Rdatatable/data.table]. https://r-datatable.com

Carta, S., Podda, A. S., Recupero, D. R., Saia, R., & Usai, G. (2020). Popularity prediction of instagram posts. *Information*, *11*(9). https://doi.org/10.3390/info11090453

Chen, J., Liang, D., Zhu, Z., Zhou, X., Ye, Z., & Mo, X. (2019). Social media popularity prediction based on visual-textual features with xgboost. *Proceedings of the 27th ACM International Conference on Multimedia*, 2692–2696.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Demeku, A. (2023, July). Instagram engagement rates: Everything you need to know. https://later.com/blog/instagram-engagement-rate/

Ding, K., Ma, K., & Wang, S. (2019). Intrinsic image popularity assessment. *ACM International Conference on Multimedia*, 1979–1987.

Dixon, S. J. (2024, April). Topic: Instagram. https://www.statista.com/topics/1882/instagram/#topicOverview

Gayberi, M., & Oguducu, S. G. (2019). Popularity prediction of posts in social networks based on user, post and image features. *Proceedings of the 11th International Conference on Management of Digital EcoSystems*, 9–15.

Gupta, V., Jung, K., & Yoo, S.-C. (2020). Exploring the power of multimodal features for predicting the popularity of social media image in a tourist destination. *Multimodal Technologies and Interaction*, *4*(3), 64.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hsu, C.-C., Kang, L.-W., Lee, C.-Y., Lee, J.-Y., Zhang, Z.-X., & Wu, S.-M. (2019). Popularity prediction of social media based on multi-modal feature mining. *Proceedings of the 27th ACM International Conference on Multimedia*, 2687–2691.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

Kim, S., Jiang, J.-Y., Nakada, M., Han, J., & Wang, W. (2020). Multimodal post attentive profiling for influencer marketing. *Proceedings of The Web Conference 2020*, 2878–2884.

Kim, S., Jiang, J.-Y., & Wang, W. (2021). Discovering undisclosed paid partnership on social media via aspect-attentive sponsored post learning. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 319–327.

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

Lewis, D. (2022, August). What is a good engagement rate on instagram? https://scrunch.com/blog/what-is-a-good-engagement-rate-on-instagram

LightGBM, L. (2023). Welcome to lightgbm's documentation! https://lightgbm.readthedocs.io/en/stable/

Lundberg, S. M., & Lee, S.-I. (2017a). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30* (pp. 4765–4774). Curran Associates, Inc. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

Lundberg, S. M., & Lee, S.-I. (2017b). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, ... Xiaoqiang Zheng. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems [Software available from tensorflow.org]. https://www.tensorflow.org/

OpenAI. (2023). Chatgpt [Accessed: 2024-05-2024].

pandas development team, T. (2020, February). *Pandas-dev/pandas: Pandas* (Version latest). Zenodo. https://doi.org/10.5281/zenodo.3509134

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2018). Scikit-learn: Machine learning in python.

Riis, C., Kowalczyk, D., & Hansen, L. (2021). On the limits to multi-modal popularity prediction on instagram: A new robust, efficient and explainable baseline. *Proceedings of the 13th International Conference on Agents and Artificial Intelligence*. https://doi.org/10.5220/0010377112001209

Umesh, P. (2012). Image processing in python. *CSI Communications*, *23*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). Attention is all you need.

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., ... Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *Dplyr: A grammar of data manipulation* [R package version 1.1.4, https://github.com/tidyverse/dplyr]. https://dplyr.tidyverse.org

Wickham, H., & Henry, L. (2023). *Purrr: Functional programming tools* [R package version 1.0.2, https://github.com/tidyverse/purrr]. https://purrr.tidyverse.org/

XGBoost. (2022). Xgboost documentation. https://xgboost.readthedocs.io/en/stable/index.html

Xu, Q.-S., & Liang, Y.-Z. (2001). Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, *56*(1), 1–11.

Zhang, Z., Chen, T., Zhou, Z., Li, J., & Luo, J. (2018). How to become instagram famous: Post popularity prediction with dual-attention.

*2018 IEEE International Conference on Big Data (Big Data)*, 2383–2392. https://api.semanticscholar.org/CorpusID:52821231

# APPENDIX A

Table 9: Classification Performance Metrics for DNN

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Low | 0.64 | 0.62 | 0.63 | 6619 |
| High | 0.63 | 0.65 | 0.64 | 6619 |
| **Accuracy** | | | **0.64** | **13238** |
| **Macro Avg** | **0.64** | **0.64** | **0.64** | **13238** |
| **Weighted Avg** | **0.64** | **0.64** | **0.64** | **13238** |

Table 10: Classification Performance Metrics for XGBoost

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Low | 0.67 | 0.71 | 0.69 | 6336 |
| High | 0.70 | 0.65 | 0.67 | 6336 |
| **Accuracy** | | | **0.68** | **12672** |
| **Macro Avg** | **0.68** | **0.68** | **0.68** | **12672** |
| **Weighted Avg** | **0.68** | **0.68** | **0.68** | **12672** |

# 9 APPENDIX B

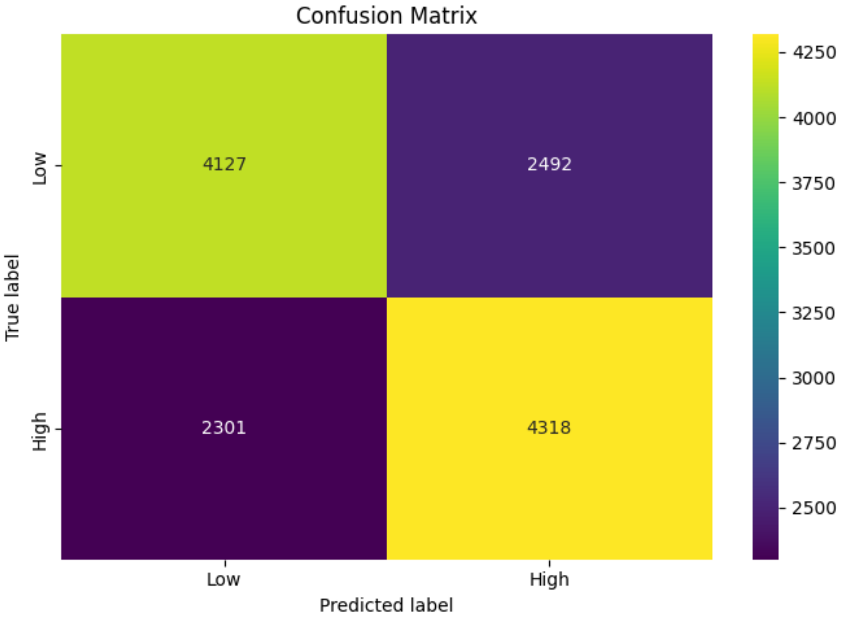

Figure 13: DNN Confusion Matrix

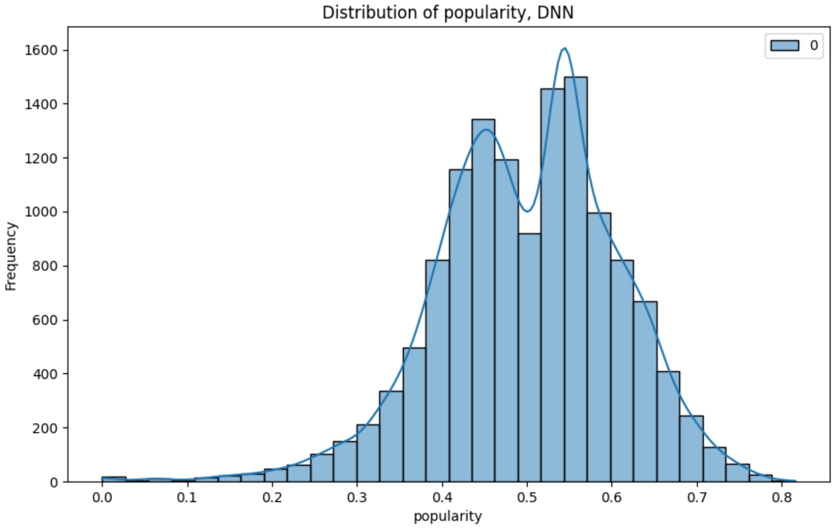

Figure 14: DNN Predicted Values Distribution

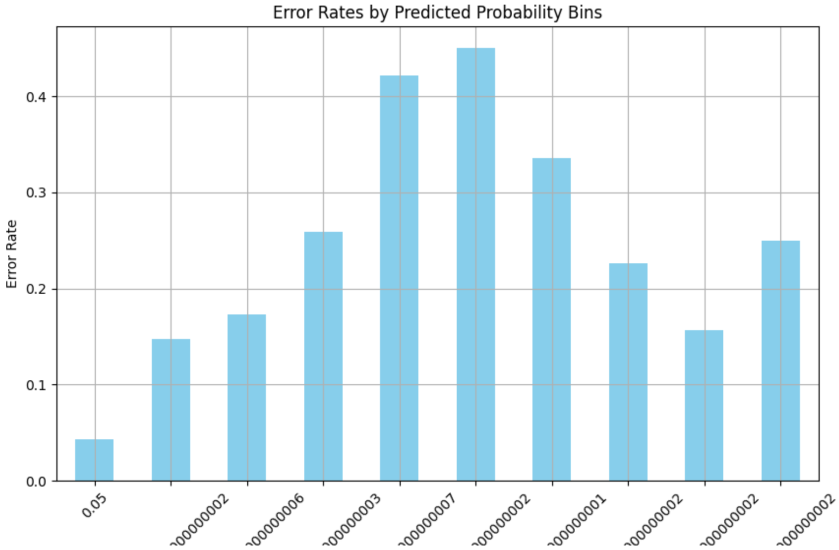

Figure 15: DNN Error Rate per predicted value

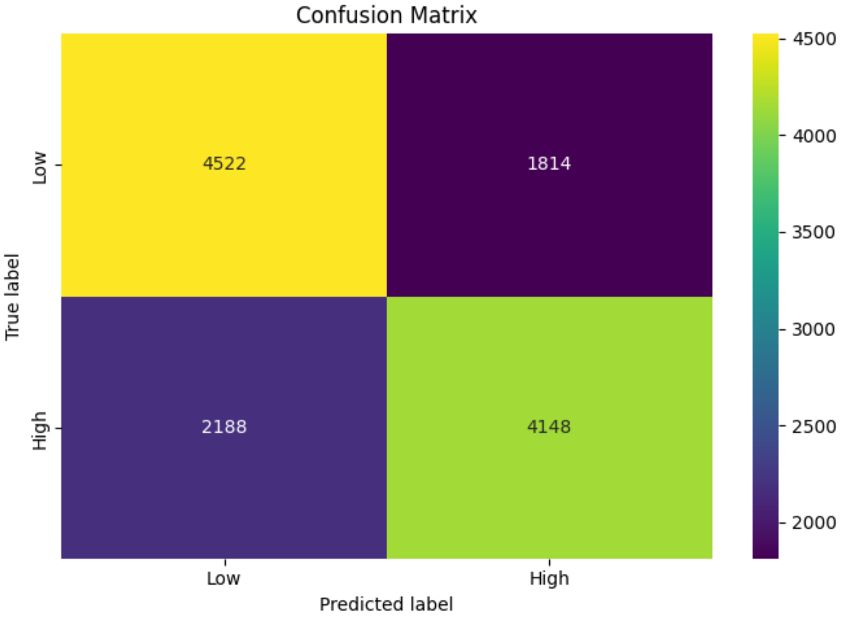

Figure 16: XGBoost Confusion Matrix

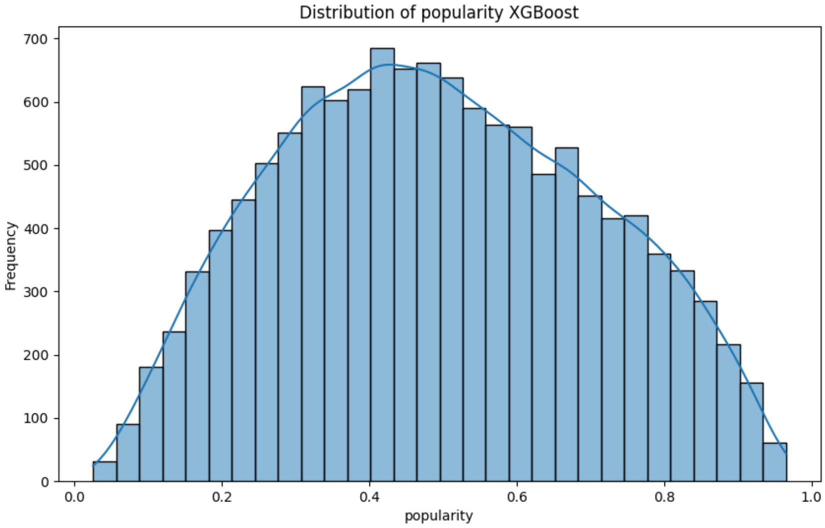

Figure 17: XGBoost Predicted Values Distribution

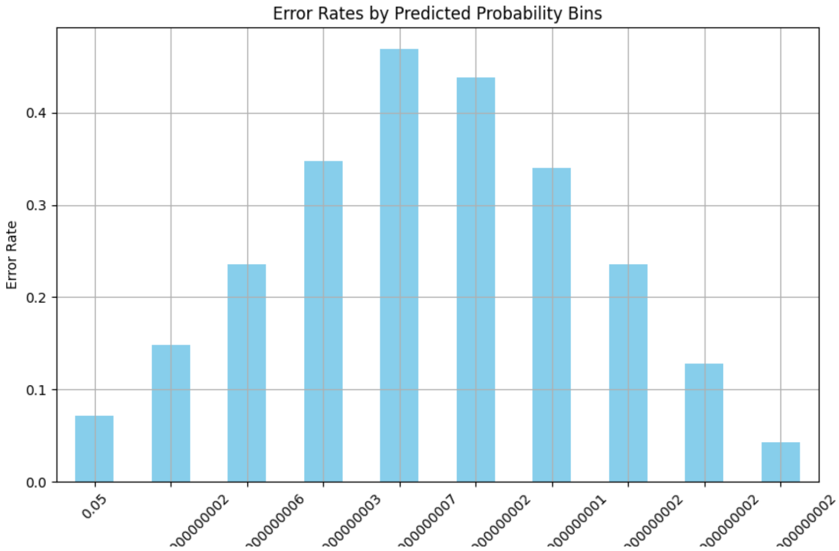

Figure 18: XGBoost Error Rate per predicted value