UNIVERSITY OF
GUELPH

JULY 29, 2024

The Analysis and Predictions of TikTok Trends Using Machine Learning

Department of Computer Science
CIS 4900
Professor Stefan C. Kremer

CREATED BY
Puneet Sandher
Sulakshan Sivakumaran

Abstract

TikTok is an ever-growing social media platform with users creating viral content daily. The platform's algorithm is ever-evolving, intending to deliver viral content tailored to the user, creating trends. The rise of content creation is becoming a mainstream source of income for many individuals and a valuable marketing tool for companies, creating a need to predict the next big trend on TikTok.

TikTrend is an application designed to predict future TikTok trends, by analyzing user engagement features and leveraging machine learning to optimize content creation for digital marketing. Machine learning techniques were utilized to preprocess data including cleaning, scaling, and encoding. The research uses two datasets to provide a comprehensive understanding of engagement metrics and account analytics: one dataset features top TikTok accounts, and the other dataset includes TikTok posts. This data provides insights to assist brands in refining their digital marketing strategies on TikTok. By understanding the core features that cause high user engagement, TikTrend aims to provide digital marketers with insights to optimize their marketing campaigns.

Table of Contents

Phase 1: Data Preprocessing

Phase 2: Model Training

Introduction

Social media is rapidly evolving and predicting trends is crucial to develop successful marketing campaigns and optimize content creation. TikTrend leverages machine learning to develop a tool that provides a comprehensive report of the past, present, and future trends on TikTok. The report aims to provide a detailed analysis of the second phase of TikTrend's development, detailing the team's understanding of machine learning algorithms and the process of developing our model through rigorous training and testing. By leveraging artificial intelligence (AI), TikTrend will be a new tool to optimize the effectiveness of digital marketing by offering strategic insights into current trends and predictive insights.

Research Question: What features of TikTok posts most significantly impact user engagement and how can machine learning be used to optimize content creation for digital marketing?

Hypothesis: Training a machine learning model with comprehensive historical engagement data can identify key features that significantly influence user interaction such as hashtags, engagement metrics (including comments, shares and likes), and account characteristics. For instance, the model will determine patterns between various engagement metrics to determine how to create viral content by looking at hashtags, the publishing time of posts, and TikTok posts' engagement metrics. The models should predict the combinations of these features with low to medium accuracy. The TikTok algorithm is ever-evolving and the model will periodically need new data to identify new trends and maintain optimization.

Overview of Machine Learning

What is Machine Learning?

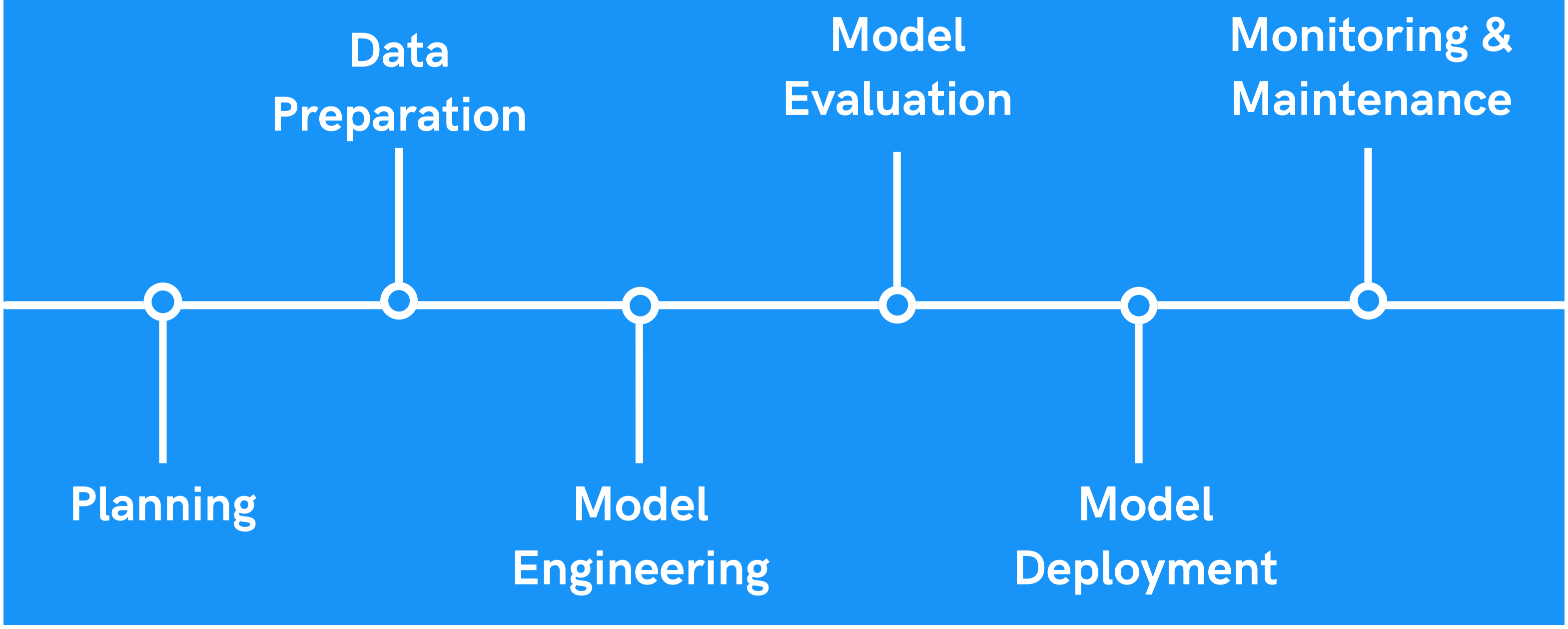
Machine Learning is a subfield of AI that refers to the process of teaching a computer how to perform a task without explicit programming. Rather than programming, datasets are fed into an algorithm to gradually improve the accuracy of its behaviours. In 1959, Arthur Samuel pioneered AI with the objective of developing computer models that imitate how humans learn (Brown, 2021). There are two primary functions of machine learning: classifying data and predicting future outcomes based on historical or provided data (Brown, 2021). Machine learning is used across all industries and integrated into daily activities through applications such as speech recognition, chatbots for customer service, image analysis, personalized recommendation systems, and fraud prevention (IBM, 2023).

There are three types of machine learning: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning involves training algorithms to classify data or predict outcomes using a labelled dataset (IBM, 2023). As input data is fed into the model, the model adjusts its weights until it has been fitted appropriately (IBM, 2023). This type of machine learning is typically used for classification and regression problems. Comparatively, unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled datasets (IBM, 2023). This algorithm can discover hidden patterns or data groupings and determine similarities and differences within the dataset, making it ideal for data analysis, image and pattern recognition (IBM, 2023). Lastly, reinforcement learning is where the algorithm isn't trained using sample data and the model learns using trial and error (IBM, 2023). Reinforcement learning mimics human learning, with software being trained to make decisions that produce the most optimal results.

Machine learning, like any technology, has challenges that require caution and ethical consideration. There are concerns of the impact AI will have on employment, privacy, bias, and accountability. The advancement of AI is shifting job demands towards roles related to AI, which brings the challenge of assisting society with this transition (IBM, 2023).

Bias and discrimination are prevalent in AI and continue to be a concern that has been difficult to overcome as there are biases present in training data, that has led to many instances of inequality (IBM, 2023). Globally, governments are implementing policies to protect data and privacy, which can be difficult to train models as it requires large volumes of data (Forbes Technology Council, 2023). Moreover, these ever-evolving regulations, as well as, jurisdictions with limited regulations puts too much responsibility on companies to uphold ethics and integrity (Forbes Technology Council, 2023). This exposes businesses to legal challenges and the need for transparent practices in the development of AI.

The Machine Learning Life Cycle



Planning

Data Preparation

Model Engineering

Model Evaluation

Model Deployment

Monitoring & Maintenance

The Machine Learning Life Cycle refers to a series of stages involved in developing, deploying, and maintaining machine learning models (Awan, 2022). The ML lifecycle consists of the following six stages:

Stage 1: Planning

The planning stage is where the problem is defined. This involves assessing the scope, success metrics and feasibility of the model (Awan, 2022). This stage is where the idea and design of the model are planned. Key questions such as what is the purpose of the model, and how will the success of the model be defined, are asked.

Stage 2: Data Preparation

This stage involves collecting and preparing the data that will be used to improve the accuracy of the algorithm. Data will be collected and then processed, which is the process of creating features from the data that will be used to train the model.

Stage 3: Model Engineering

Model engineering is where the model is built and trained. The dataset will be separated into a training set and a testing set. The training set will be used to build this model by feeding this data into the algorithm.

Stage 4: Model Evaluation

Model evaluation is where the model is tested. The testing set will be used to validate the accuracy of the model and identify any errors that occur.

Stage 5: Model Deployment

This stage involves deploying the model into a production environment. This model is now ready to make predictions on new data and can be integrated into other applications to be used (Awan, 2022).

Stage 6: Monitoring and Maintenance

The performance of the model is monitored and maintained with changes and updates implemented as needed to maintain and improve the quality of the model.

Project Architecture

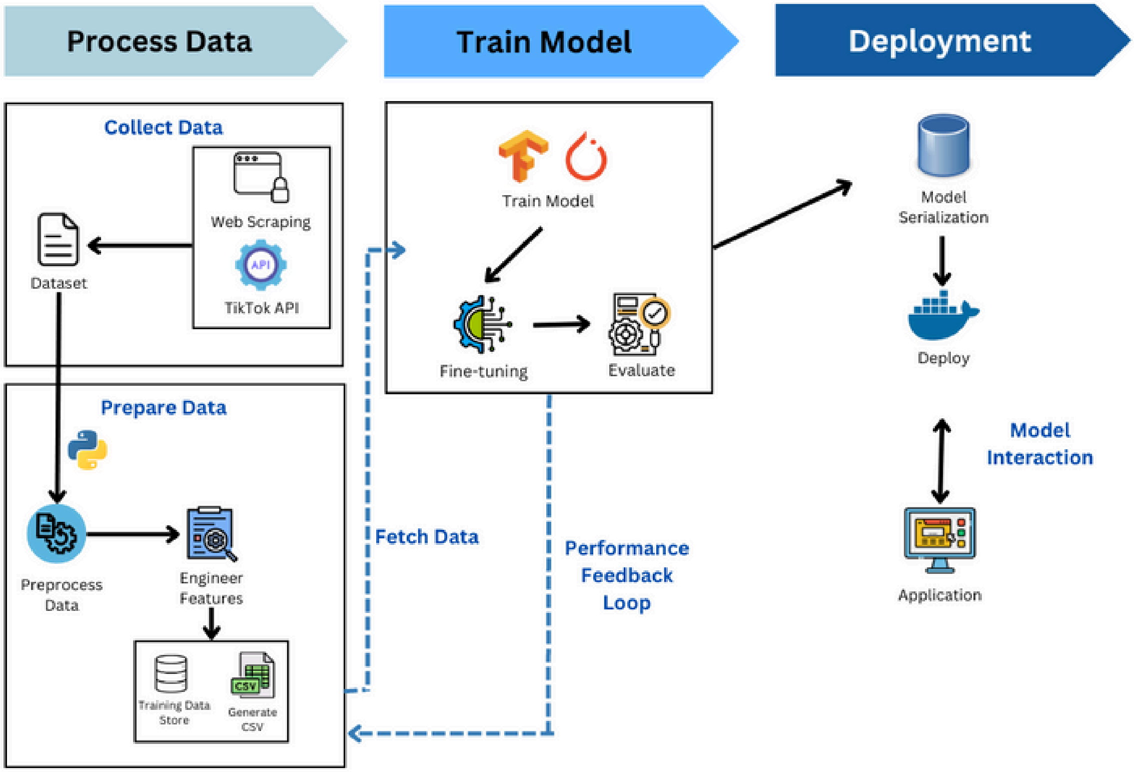


FIGURE 1.0

Figure 1.0 provides a detailed overview of TikTrend's architecture and its three main components: data preprocessing, model training and deployment. This diagram is subject to change, as adjustments may be made to optimize performance and quality.

Data Processing

There are two objectives for data preprocessing: collecting relevant data and preparing the data for training. A public repository had datasets for TikTok posts and top creators that were collected using web scraping and the TikTok API. This data undergoes preprocessing to filter out irrelevant information and transform it to identify engineered features such as timestamps, captions, and user engagement metrics. Furthermore, various data encoding techniques are applied depending on the data, to ensure it is compatible with the machine learning model. To support the development of multiple testing models and efficient data access and retrieval, the data is organized and stored in a structured database and as CSV files.

Model Training

Keras, scikit-learn, and PyTorch are popular frameworks for training machine learning models that support a variety of supervised learning techniques. Each of these tools will be evaluated based on the needs of the models and one tool will be selected for training. After training the model, it will undergo a fine-tuning process to improve accuracy and overall performance. The model's performance is evaluated with the testing data to provide an unbiased assessment, as well as, looking at metrics such as accuracy, precision, mean absolute error for regression and various other metrics. This iterative process of refining the training data and model parameters is based on the performance feedback. Moreover, it is a common practice to develop multiple models to be evaluated against each other to select the most effective one for deployment.

Deployment

A model that meets the predefined acceptance criteria is selected, serialized and exported to be deployed. The serialized model and its dependents are held within a Docker container to ensure usability for all environments. After deployment, the application interface allows users to interact with the model, which retrieves and processes data from the Docker container. This deployment allows for future scalability and flexibility, as the system can adapt to higher workloads and user requirements.

Data Collection

Overview of Data Source

Developing a machine learning model requires a substantial amount of TikTok data to accurately identify trends. This data will be used to train, test, and finetune the model to improve its accuracy. Collecting data from TikTok can be accomplished by leveraging the TikTok API and web scraping, however, this is a complex and time-consuming process. To complete this project by the expected deadline, a public repository on GitHub contained TikTok datasets that were relevant to the model. This repository was created by Ivan Tran and can be found at https://github.com/datares/TikTok_Famous.

The repository contains 2 main sources of data. The first dataset is a collection of data on TikTok videos. This CSV contains analytics on various TikTok reels from a wide range of content creators. Each row of data includes the reel's time of creation, the username of the creator, relevant hashtags, song used, length of video, number of likes, number of shares, number of comments, number of times played, number of account followers, number of total account likes, and number of total videos under this account.

The second dataset is a collection of data on TikTok accounts. This CSV contains analytics on various TikTok content creators rather than their specific posts. Each row of data includes the account username, country of origin, number of followers, number of views, number of likes, percentage of engagement, if it is a brand account, gender, age, ethnicity, if they are famous, the genre of content, and if they are a member of the LGBTQ community.

Volume of Data

Data volume refers to the amount of data available in a given data set. The amount of data needed depends on the problem and a general rule of thumb is that the more data the better (Eugene Dorfman, 2022). A large dataset improves the accuracy of the model as it provides a comprehensive representation to help the model make predictions or classifications more accurately (Stupak, 2024). Moreover, it prevents overfitting where the model learns the noise and patterns from a smaller dataset, reducing the model’s performance (Deepchecks, 2021).

In addition, exposing the model to more scenarios will allow it to effectively learn and understand features, thus capable of handling more complex tasks (Stupak, 2024). Overall, several factors influence the required volume of data, including the type of problem, data quality, accuracy requirements, and the number of features.

The dataset with TikTok posts has 41,702 rows and 13 columns of data and the dataset of the TikTok accounts has 256 rows and 14 columns of data. The amount of data needed for a machine learning model depends on the model's complexity. A model that uses a large amount of layers/nodes or many algorithms will require more data (Eugene Dorfman, 2022). The TikTok datasets would ideally be larger for the model’s purposes, and this will be taken into account when analyzing the model. A dataset is considered sufficient by following the 10 times rule, where a dataset has at least ten times as many data points as there are features in the dataset, to prevent sample bias (Eugene Dorfman, 2022). This rule is followed after the datasets are preprocessed, thus there is sufficient data for this machine learning model. As development progresses, if the model’s accuracy is significantly hindered by the volume of data, additional data collection and data augmentation techniques will be used.

Data Preparation

Overview

Data preparation is a critical stage in the machine learning lifecycle, as it directly influences the performance and accuracy of the model (The Pecan Team, 2023). Using the processed data, the algorithm will learn patterns, relationships and analysis, thus it is a core component of the machine learning process.

The data preparation stage can be broken down into the following steps:

1. Data Collection: Begin by collecting data that is relevant to the purpose of the model. This data can be collected from a variety of sources such as API's, databases, or open/external datasets.
2. Clean Data: Once the data has been collected, the dataset will be cleaned. This can involve removing outliers, handling missing values, and more.
3. Transform Data: The data is converted to fit the requirements of machine learning algorithms such as feature scaling and encoding (The Pecan Team, 2023).
4. Create Training and Testing Set: After the data is transformed, it will be split into two: a training set and a testing set.

Data preparation is an iterative process as data will be adjusted depending on the performance and accuracy of the model.

Data Preparation of TikTrend

Data Cleaning

Data cleaning is a critical step in machine learning in which the raw data is prepared for analysis. This process is important as it directly impacts the efficiency and accuracy of the model. Clean data ensures that the machine learning algorithms can identify patterns and insights resulting in more reliable predictions. The following are the core techniques used to clean both TikTok datasets.

1. Removing Irrelevant Data

Initially, irrelevant data columns that are unnecessary for the model's goals are removed to focus on more relevant features. In the TikTok account dataset, the following columns were removed to focus on numeric engagement data: Usernames, Age, Brand Account, Song, Gender, Famous, Ethnicity, LGBTQ and Rank. The following columns were removed from the TikTok posts dataset: video_length, id, user_name, and song. These columns were removed to simplify the pre-processing phase, allowing for more relevant columns to be focused on. Columns that were dropped will **not** be reintroduced during the development and training of the model.

2. Processing Null Values in Data Sets

It is necessary to appropriately handle null values as machine learning models will have difficulty training effectively. Many entries in categorical data had null values. As null values are still considered values, these were replaced with a constant. For example, in the TikTok account dataset, all null values were filled in under the Genre column with the string "noGenre." In the TikTok posts dataset, a similar strategy was used with posts with no hashtags and was replaced with "NoHashtag." There was no numerical data that had a null value as this dataset was partially preprocessed.

3. Converting Data Types

Numerical values that were represented as strings, were parsed to float values using Python functions. For example, in the TikTok account dataset, the percentages in the Engagement column were converted to floats. These conversions are done to better process the datasets and improve the model's accuracy for training, testing and overall predictions.

4. Processing Hashtag Strings for Encoding

In the TikTok post dataset, the hashtags were initially a string but were converted to an array of strings. This conversion involved numerous steps. To begin with, before converting the hashtags into vectors, it was necessary to ensure that the data was consistent and properly processed to avoid errors. All strings were transformed to lowercase, and special characters were removed to simplify the data, reduce noise, and these characters were irrelevant to the analysis. Many hashtags were variations or abbreviations of "for you page," and were standardized for consistency. Furthermore, each hashtag was a single string without spaces and needed to be split by word which was accomplished by leveraging the WordSegment library, to prepare them for encoding using Word2Vec. Each word in the hashtag was stored in an array and encoded using Word2Vec which captures word-level relationships but not multi-word relationships for example, the hashtag "MachineLearning" would be transformed to "Machine" and "Learning."

5. Processing Outliers

After assessing the datasets, it was recognized that there were no extreme outliers that would have affected our algorithm. Moreover, the purpose of the analysis is to predict viral content and these outliers provide data to comprehensively understand its characteristics. Outliers could represent when content has really "blown up" or has completely failed to "take off". This could aid the algorithm in identifying viral and non-viral trends and may lead to some interesting final results.

Transformation Techniques

Data transformation techniques are simply techniques that prepare data for the model. For the model, techniques were commonly used to transform categorical values into numerical values.

1. One-Hot Encoding

One of the techniques used was hot encoding. This technique is used on categorical values, where a new column is made for each unique category with zeros and ones being filled in each column (1 representing true and 0 representing false). This was implemented in the Genre column, with each genre being a new category. For example, the genre of "Dancing" became a new column with each user having a 1 if the account is a dancing account or a 0 if not.

2. Cyclic Encoding

The TikTok posts dataset had a created_time column that was encoded using cyclic encoding. Cyclic encoding is used when there are recurring patterns in a dataset that repeat at a fixed interval, specifically in this dataset it is the month, day, hour, and weekday (Lewinson, 2022). Cyclic encoding represents data in a manner where the first point in the cycle lies close to the last point in the cycle, simply showing that the beginning and end of the cycle are close to each other.

All the metrics related to the publishing time have a sine and cosine column to represent how they are mapped to represent the cyclic nature and identify the distance between two points (Lewinson, 2022). The months were encoded with the assumption that there are 31 days in every month, which creates a skew for months that are less than 31 days, however, the purposes of the model do not require fine granularity of the day.

3. Word2Vec Embeddings

Word2Vec is a model used to create vector word embeddings and is especially useful for large datasets. There are two architectures of Word2Vec: continuous bag-of-words (CBOW) and continuous skip-gram model (TensorFlow, 2024). CBOW predicts the target word based on the context and skip-gram predicts the context given a word (TensorFlow, 2024). The TikTok posts dataset has a column of hashtag arrays with the column containing a total of 1289626 hashtags, and Word2Vec embeddings were used to represent the context significance within the dataset.

Word2Vec recommends a vector dimension between 100 and 1000, and a dimension of 200 was selected to create embeddings for the hashtags (Ministry of Justice: Data Science Hub, 2024). This size is sufficient to identify relationships between the hashtags, manage memory effectively and ensure embeddings are detailed and practical for model training (Ministry of Justice: Data Science Hub, 2024). The dimension of the vectors may be adjusted during model training, to optimize accuracy. Vector embeddings offer a powerful method to analyze and understand semantic relationships and clustering within hashtags (Burley, 2023). These vector representations will be an asset to identifying popular TikTok posts as they can be leveraged to identify trends, understand relations, and enhance recommendation systems using semantic tags (TensorFlow, 2024).

4. Scaling

Feature scaling is a technique used to adjust the scale of data, ensuring that each feature is equally considered in the analysis (Bhandari, 2024). Many machine learning algorithms are sensitive to the magnitude of features. Misinterpretations of data due to the scale of data will cause inaccurate predictions (Bhandari, 2024). Both datasets had a variety of columns and engagement metrics of different magnitudes, thus developing an accurate model required appropriate scaling. Both datasets used log scaling and then normalization (min-max scaling). Implementing both scaling techniques reduces skewness in the data, and then normalizes all features to a consistent range. (Bhandari, 2024). Log scaling is applied when the data has a large magnitude with outliers that skew the analysis. This scaling compresses the scale and brings the data closer together (Donia, 2023). Each column in both datasets has significant outliers to represent the viral accounts or posts, thus log scaling was proven to be the most effective. By applying log scaling, the data was normalized which will improve the accuracy of the model as it makes patterns more visible and reduces skewness in data (Donia, 2023). After, min-max scaling was applied as it transformed features to be within the range of zero and one, to eliminate inconsistent scaling (Donia, 2023).

Data Splitting

Data splitting is required to correctly train and assess the accuracy of a model. The dataset will be divided into three subsets: training, testing and validation. A training set is the data that will be used by the model to learn patterns, relationships and features within the data. The testing set is used to evaluate the performance and accuracy of the model on unseen data. The validation set evaluates the model's performance during training to adjust hyper-parameters and prevent over-fitting (Pandian, 2022). The general rule for splitting the dataset is assigning 80 percent for training the model and the remaining 20 percent for testing the model, which is split in half to have a validation and test set (The Pecan Team, 2023). The five-fold cross-validation technique is a robust method for evaluating machine learning models. This validation technique involves dividing the dataset into five folds and training and validating the model five times (Pandian, 2022). For each iteration, a different fold is used as a validation set and the rest are used for training, then the performance is measured using various metrics such as accuracy and precision (Pandian, 2022). Finally, an average of the performance metrics is calculated to provide an estimate of the model's overall performance (Pandian, 2022).

Results

Descriptive Statistics

Table 1.0 - Statistical Summary of Top TikTok Accounts

	Mode	Median	Mean	Standard Deviation
Followers	9,500,000	12,600,000	15,221,094	8,466,966
Views	1,600,000	1,600,000	3,722,951	6,504,690
Likes	1,000,000	276,250	612,181	980,863

Table 1.0 provides a statistical overview, including the mode, median, mean, and standard deviation, of the top TikTok accounts dataset.

Table 1.1 - Statistical Summary of Top TikTok Posts

	Mode	Median	Mean	Standard Deviation
# of Likes	1,100,000	283,650	551,413	685,308
# of Shares	9	460	3,636	13,281
# of Comments	27	2,176	21,303	15,422
# of Views	1,100,000	1,650,000	3,473,976	6,436,514
# of Account Followers	1,000,000	2,500,000	5,141,533	6,150,926
# of Account Total Likes	4,700,000	4,800,000	44,524,320	255,449,188
Year	2020	2020	2019	0.18

Table 1.1 provides a statistical overview, including the mode, median, mean, and standard deviation, of the top TikTok accounts dataset.

Data Visualizations

Top TikTok Accounts Dataset

Note: This graph is based on the original unscaled dataset

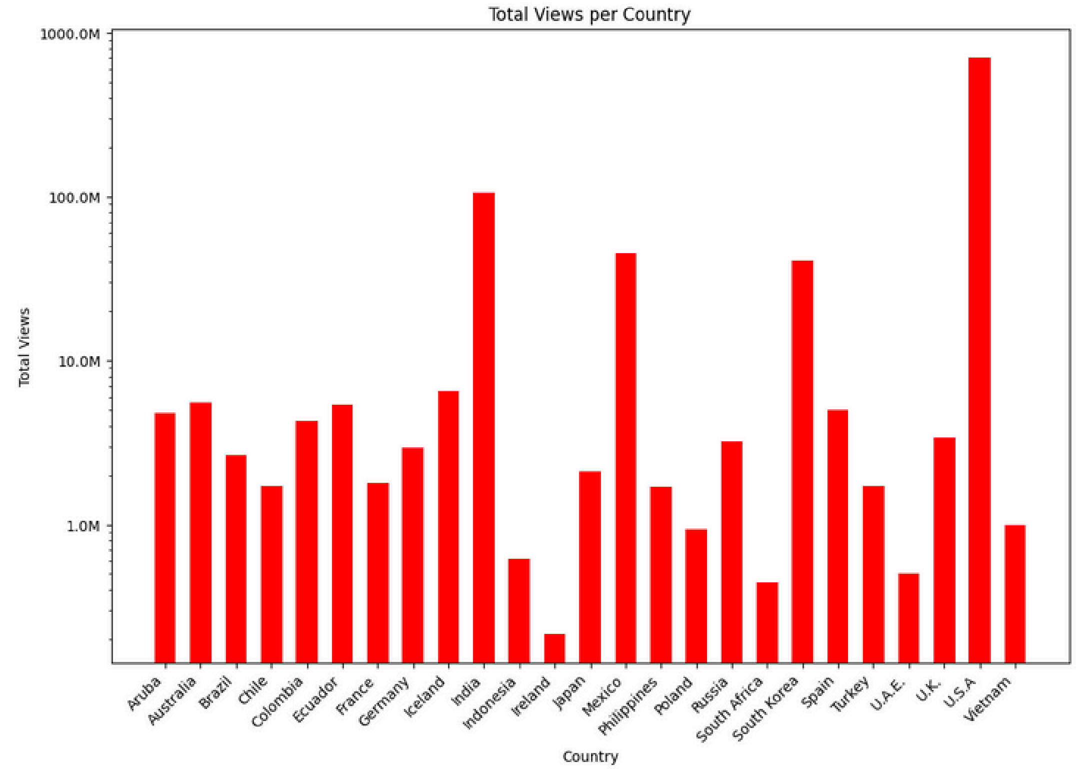


FIGURE 1.1

Figure 1.1 showcases the total amount of TikTok views produced by accounts within their respective countries. Looking at the graph, many notable observations can be made. To begin with, the country that produces the most views within this dataset is the U.S.A., while Ireland is the country that produces the least amount of views. Many countries in Europe have a similar view count, with countries such as Spain, Germany, and Iceland having a view count between 6 and 10 million. India has the second-highest view count within the dataset, with a view count of approximately 100 million. This could be due to its high population with TikTok becoming more relevant around the world. Surprisingly, Mexico has the third highest view count in this dataset which could be due to Spanish being the primary language in the country. With Spanish being one of the most widely spoken languages in the world, Spanish content produced within Mexico can reach and attract a wide range of viewers, increasing the country's view count. Similar to Spain, Aruba has a high view count of close to 10 million views, with the country speaking primarily Dutch and Papiamento. This means that the content made in Aruba attracts all of those who speak either Dutch or Papiamento, attracting a wide range of viewers and increasing the country's view count.

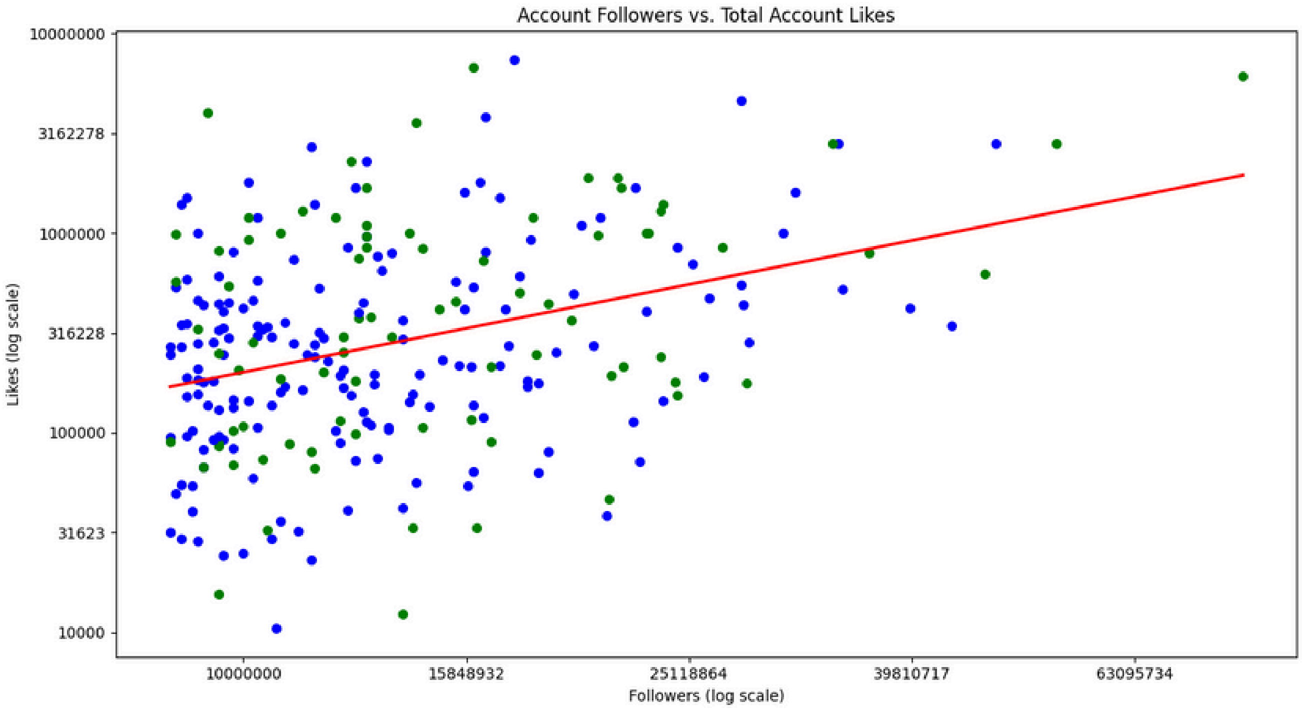


Fig. 1.1. Total TikTok Account Followers Compared To The Total TikTok Account Likes

* All green data points are Accounts with a Genre of "Dancing"

FIGURE 1.2

Figure 1.2 is a scatter plot that illustrates the relationship between a TikTok account's total number of followers and its total number of likes. Looking at the graph, many notable observations can be made. The graph showcases a positive correlation, with the number of followers increasing as the number of likes increases. This can be shown by the trend of the points moving upward from left to right. There is also a cluster of points around 10,000,000, followers, with the points slowly starting to spread out and decrease as the follower count increases. This cluster of points could represent a subcategory of the data. For example, this cluster could be the majority of accounts that fall under the "Dancing" genre, with all those accounts sharing a similar trend in terms of followers and likes. There are a few outliers that go against the trend, such as the point in the top right of the graph. This data point has an extremely high likes and follower count with no other points in its near vicinity.

TikTok Posts Dataset

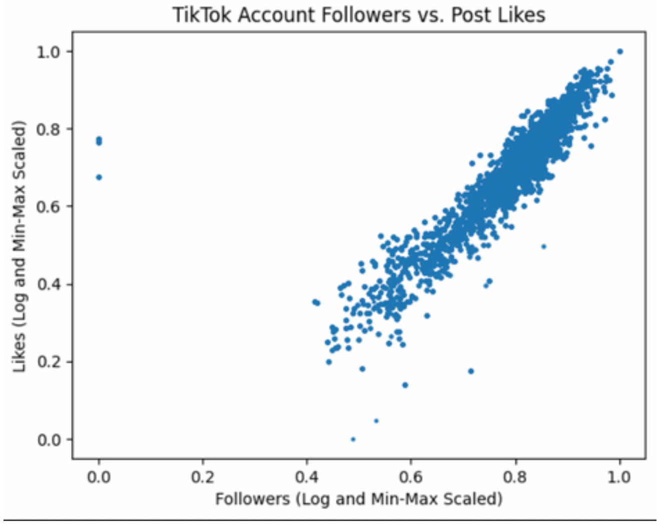


FIGURE 1.3

Figure 1.3 illustrates the relationship between the followers a TikTok account (x-axis) has and the likes received (y-axis) on a post. The data was log-transformed to handle the wide range of values, then the min-max scale where each data point is between 0 and 1. This transformation allows for a better visual representation of the relationship between the number of TikTok followers and likes. There is a strong positive correlation between followers and likes, suggesting that accounts with more followers tend to receive more likes on their posts. The complete TikTok posts dataset, tiktokPostsProcessedData.csv, has a data point with a follower count of 95 receiving over 17000 views on a TikTok post. The clustering along the diagonal suggests a proportional increase in likes and followers. The diagonal becomes narrower at higher follower counts, demonstrating that accounts with a large number of followers receive a more consistent number of likes. There are a few data points on the left, with lower follower counts but receiving a high amount of likes, meaning that these posts could have been viral despite the minimal followers. This demonstrates the potential reach and engagement on TikTok from accounts with a small following.

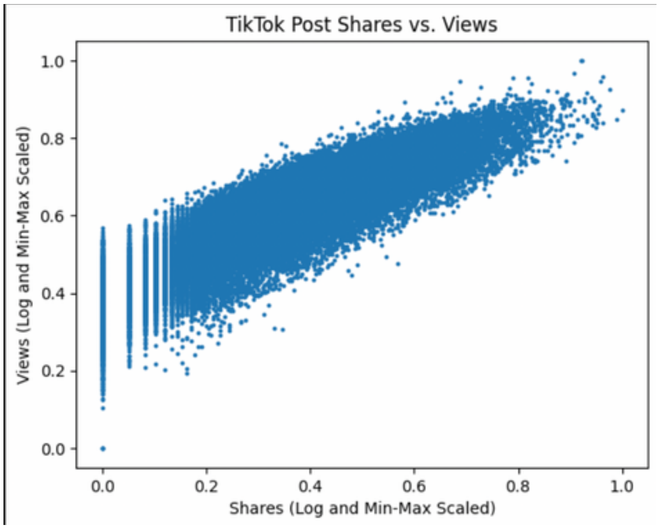


FIGURE 1.4

This figure is a visual demonstration of the relationship between a TikTok post's shares (x-axis) and views (y-axis). The data was log-transformed to handle the wide range of values, then the min-max scale transformed each data point between 0 and 1. This allows for a clear visual representation of the relationship between the number of shares and views a TikTok post garners. The scatter plot indicates that posts that are more frequently shared receive more views, overall the plot has a strong positive correlation. There is a proportional increase in shares and views as there is significant clustering along the diagonal line. Moreover, the plot demonstrates that posts can still achieve a significant number of views even if they are not widely shared. The complete TikTok posts dataset, tiktokPostsProcessedData.csv, has a data point with 30 shared and garnering over 20000 views on a TikTok post. There is a small subset of posts in the top right of the graph with high engagement levels, demonstrating this content may have gone viral.

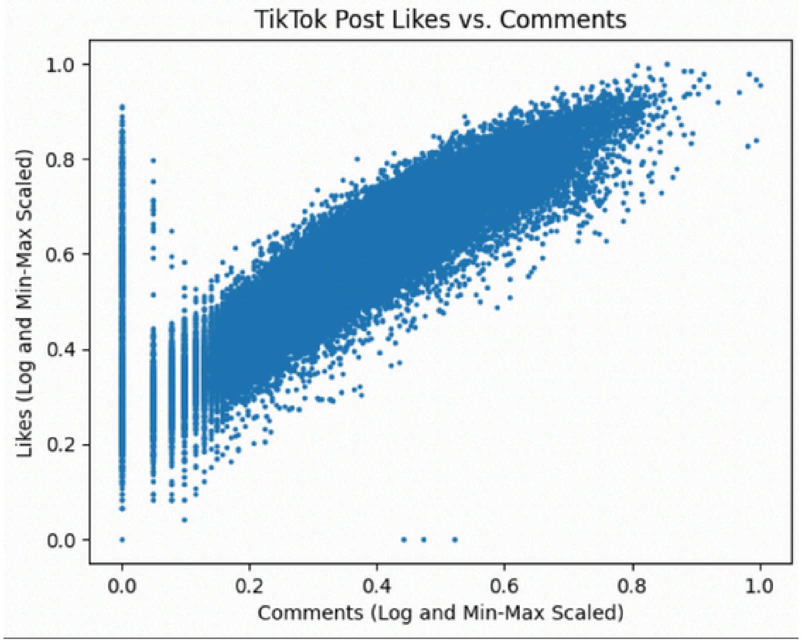


FIGURE 1.5

The scatter plot illustrates the relationship between the number of likes and comments on TikTok posts. Each point on the graph represents a TikTok post, with the x-axis displaying the comments and the likes on the y-axis. The data was log-transformed to handle the wide range of values, then the min-max scale transformed each data point between 0 and 1. This allows for a clear visual representation of the relationship between the number of comments and likes a TikTok post garners. There is a positive correlation between likes and comments, demonstrating that posts with higher comment counts tend to receive more likes. The clustering along the diagonal suggests a proportional increase in likes and comments. Moreover, there are a significant amount of data points at the lower end, indicating posts with little engagement in both likes and comments. There is a small subset of data points that gain high levels of likes and comments, most likely indicating that the post went viral.

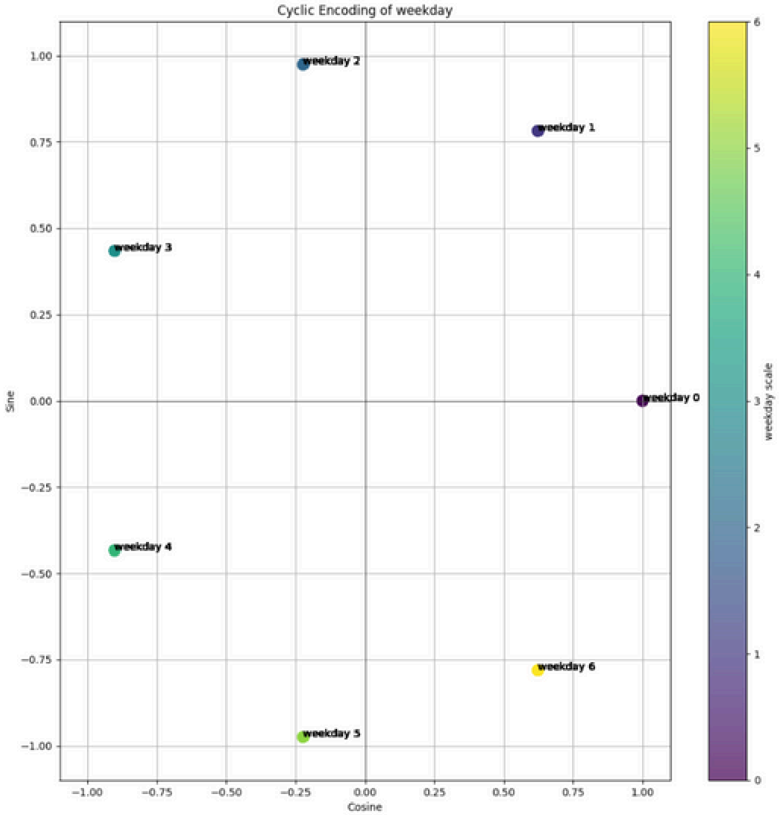


FIGURE 1.6

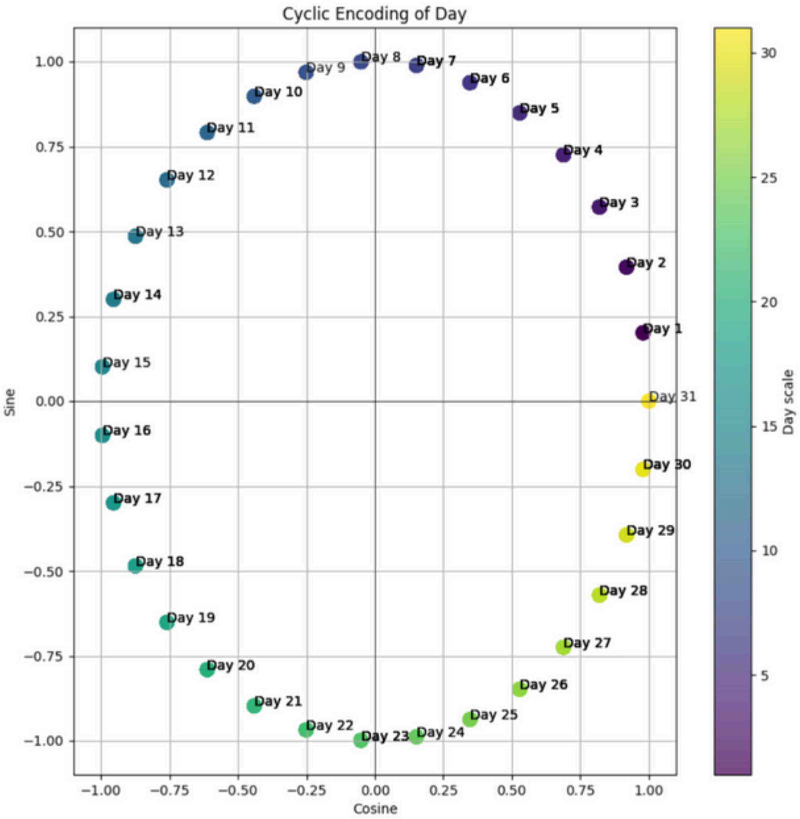


FIGURE 1.7

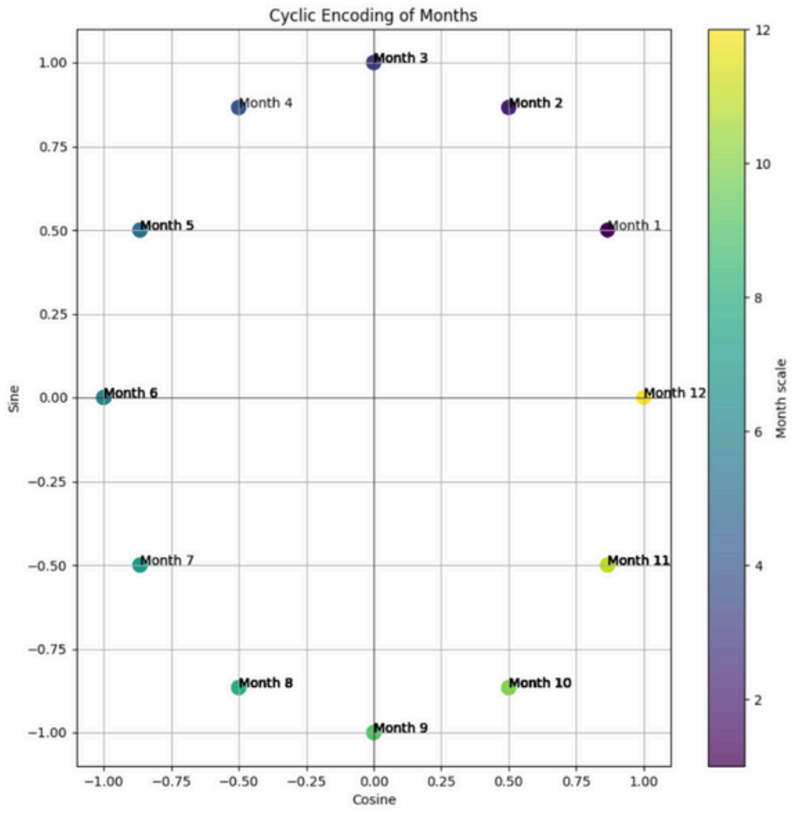


FIGURE I.8

Figures 1.6, 1.7, and 1.8 are visual representations of the cyclic encoding of a TikTok post's publishing date, specifically the month, day and day of the week. In these figures, cosine is represented on the x-axis and the sine function on the y-axis forms a circle for each time unit. This circular representation demonstrates the cyclical nature of time, as seen in the graphs where the points referring to the beginning and end of each cycle are close to each other for example, month one and month 12.

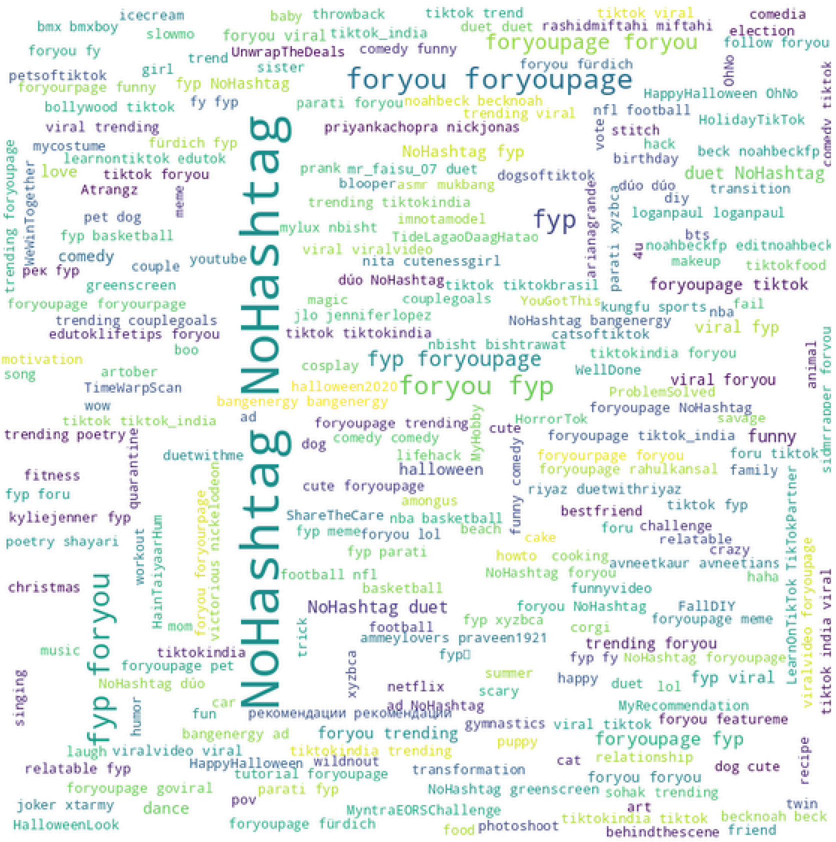


FIGURE 1.9

This word cloud in Figure 1.9 showcases the most popular hashtags in the TikTok posts dataset. The size and frequency of the words indicate high frequency and represent popular trends within the dataset. The diagram was created without consolidating similar hashtags to demonstrate the variations of the same hashtag. For example, the hashtag "for you page" appears in many different forms. Furthermore, the hashtags highlight the geographical diversity of the posts in the dataset, with many hashtags in different languages or references to specific countries. Overall, the figure provides insights into how content is being made and how well users are engaged.

Top TikTok Account Dataset - Before Preprocessing

TABLE 1.2

	Account #1	Account #2	Account #3	Account #4	Account #5
Rank	1	26	134	224	225
Username	@charlidamelio	@tiktok_india	@linhbarbie	@vivekkeshari1	@rahimabram
Country	U.S.A	India	Vietnam	India	Russia
Followers	78.9m	24.2m	12.3m	9.2m	8.6m
Views	38.3m	2.3m	1.0m	540.6k	1.6m
Likes	6.1m	153.2k	168.5k	82.3k	271.2k
Engagement	16.60%	6.90%	17%	16%	17%
Brand Account	0	1	0	0	0
Gender	Female	null	Female	Male	Male
Age	16	null	null	24	22
Ethnicity	White	null	Southeast Asian	South Asian	White
Famous	0	1	0	0	1
Genre	Dancing, Lipsyncing, Lifestyle	null	Lipsyncing, Acting	Motivational Speaking	Lipsyncing, Promotion, Lifestyle
LGBTQ	0	null	0	0	0

Top TikTok Account Dataset - After Preprocessing *

TABLE 1.3

	Account #1	Account #2	Account #3	Account #4	Account #5
Rank	1	26	134	224	225
Followers	1	0.4668	0.1614	0.0304	0
Views	0.9483	0.5011	0.3686	0.2708	0.4433
Likes	0.9706	0.4087	0.4232	0.3140	0.4958
Engagement	0.4203	0.0899	0.4333	0.4007	0.4333
Dancing	1	0	0	0	0
Lipsyncing	1	0	1	0	1
Lifestyle	1	0	0	0	1
Acting	0	0	1	0	0
Motivational Speaking	0	0	0	1	0
Promotion	0	0	0	0	1
noGenre	0	1	0	0	0
Country_U.S.A	1	0	0	0	0
Country_India	0	1	0	1	0
Country_Vietnam	0	0	1	0	0
Country_Russia	0	0	0	0	1

***Not all columns are shown in the above table. Complete dataset is available in outputAccountDataset.csv**

Analysis of Processed Data

Final Preprocessed Data

The TikTok account dataset has a total of 257 data points and the TikTok posts dataset has 41,703 data points. Five distinct data points in each dataset are showcased before and after preprocessing as illustrative examples to demonstrate the effects of preprocessing and patterns in the datasets.

Top TikTok Account Dataset

The TikTok account data set was preprocessed using numerous data preparation techniques. The final preprocessed dataset is contained in the file "preprocessedAccountData.csv." To take a deeper look into the effects of preprocessing, there are two files of five data points each, that showcase the dataset before and after preprocessing. File "inputAccountDataset.csv" includes five data points before preprocessing, and file "outputAccountDataset.csv" shows those five points after preprocessing. These data points can also be found in Table 1.2 and Table 1.3. Below is a breakdown of the significance of the five data points after preprocessing.

To begin with, in the first data point, the number of followers is 1.0, showing that this TikTok account has the most followers in the dataset. This is because, when scaling the data using min-max scaling, numbers are scaled into a range from 0-1, with 0 being the smallest number in the dataset and 1 being the largest number in the dataset. This data point also contains three genres, which is a high number of genres for a single account.

Moving on to the second data point, this account has a genre of "noGenre". In the original dataset, this account had no genre listed and was read as a null value by the model. The "noGenre" genre was created and set for null values during preprocessing. Only two accounts in the entire dataset were listed as "noGenre".

Looking at the country column under the third data point, this account is located in Vietnam, with this being the only account made and run in Vietnam in the entire dataset. Looking at a study conducted in Vietnam in 2003, "over 81 percent of Gen Z respondents confirmed using TikTok" (Statista Research Department, 2024), showcasing the popularity of the platform.

Unlike the previous data point, the fourth data point is located in India. This was a frequent country in the dataset, with a total of 60 accounts being from India. This account also had a single genre of "Motivational Speaking". This genre was quite uncommon, with only three accounts having this genre listed in the dataset.

Finally, taking a look at the final data point, this account was located in Russia. Despite Russia's high population, only three accounts in this dataset were found to be made in Russia. However, this may not be a correct indicator of the popularity of TikTok in Russia due to the size of the dataset. The number of followers in Russia is 0.0, meaning that this account has the smallest follower count in the dataset. Looking at the original dataset, it is clear that this account has a follower count of 8.6 million.

TikTok Posts Dataset - Before Preprocessing

TABLE 1.4

	Post #1	Post #2	Post #3	Post #4	Post #5
ID	6892519502127320322	6882945622962375942	6861708313474731270	6892508421405281542	6892537706161769729
Time Posted	1604789755	1602560714	1597615969	1604787205	1604793994
Username	robertdowneyjnrofficial	jackblack	vindieselbrasileiro	chars…alien	tiktokforyouvids
Hashtags	['summer', 'avengers', 'ohnanana', 'robertdowneyjnr', 'rdj', 'foryou', 'viral', 'foru', 'water', 'stark']	[]	['viral', 'diadossolteiros', 'resort', 'challenge', 'fy', 'geracsotiktok', 'velozesefuriosos', 'brasil', 'brasilbemsertanejo', 'domingo', 'parceria']	['fypシ foryoupage', 'fyptiktok', 'fyp', 'fypps', 'fypforyourpage', 'fyppppppppppppp', 'fyppage', 'fyppls', 'fypforyou', 'fypforyoupageシ', 'fypシ']	[]
Song	Capone - Oh No	original sound	som original	original sound	original sound
Video Length	9	8	19	8	13
# of Likes	2698	3900000	16800	708	966
# of Shares	4	59100	75	23	1
# of Comments	230	72300	302	48	8
# of Views	17300	17700000	338300	3542	4438
# of Account Followers	651100	3600000	1700000	26200	3700000
# of Total Account Likes	144000	16200000	4200000	73800	116800000
# of Account's Posts	3	16	22	600	2377

TikTok Posts Dataset - After Preprocessing *

TABLE 1.5

	Post #1	Post #2	Post #3	Post #4	Post #5
month sin	-0.5001	-0.8660	-0.8660	-0.5000	-0.5000
month cos	0.8660	0.5000	-0.5000	0.8660	0.8660
day sin	0.9885	0.4853	-0.1012	0.9885	0.9987
day cos	0.1514	-0.8743	-0.9949	0.1514	-0.05064
hour sin	-0.5001	0.7071	-0.5000	-0.5000	0.0
hour cos	0.8660	0.7071	0.8660	0.8660	1.0
weekday sin	-0.9749	0.7818	-0.7818	-0.9749	-0.7818
weekday cos	-0.2225	0.6234	0.6235	-0.2225	0.6235
Vector Hashtags	[1.66072451e-05 -1.41368764e-04 …	[-5.13711529e-06 -5.52049547e-04 …	[6.39662145e-06 -3.54825514e-04 …	[-1.54113456e-06 -1.65614860e-04 …	[-5.13711529e-06 -5.52049547e-04 …
# of Likes	0.1552	1.0	0.3675	0.0	0.0360
# of Shares	0.0890	0.9999	0.3534	0.2414	0.0
# of Followers	0.6490	0.9945	0.8429	0.0	1.0
# of Comments	0.3609	1.0000	0.39109	0.1885	0.0
# of Views	0.1862	0.9999	0.5353	0.0	0.0265
Total Account Likes	0.0907	0.7318	0.5486	0.0	0.9999
Post Year	0.0	0.0	0.0	0.0	0.0

***Not all columns are shown in the above table. Complete dataset is available in testTikTokPostOutput.csv**

Top TikTok Posts Dataset

The TikTok posts dataset had various types of data that were preprocessed using data preprocessing techniques. Table 1.4 provides a snapshot of the diverse content and engagement strategies used on the platform with five distinct data points, as shown in the file "testTikTokPostInput.csv." Table 1.5 displays these data points post-preprocessing, as shown in the file "testTikTokPostOutput.csv." The complete preprocessed dataset is contained in the file "tiktokPostsPreprocessedData.csv."

The first post is a promotional TikTok from Robert Downey Jr. for his latest Avengers movie. This example represents a common marketing strategy used by celebrities and studios to promote projects, using hashtags relevant to the movie and actor, as well as trending music in the background. Moreover, many posts have music in the background and others have the original audio from the video.

The second post is from another celebrity, with high engagement but a lack of hashtags. Posts without hashtags are processed with the string "noHashtag." This is an example of a common celebrity account with strong engagement metrics despite minimal posts and a lack of hashtags demonstrating a celebrity's strong influence on the platform.

The third post is from a TikTok account tailored for Brazilians with hashtags in Portuguese and using hashtags like "geracsotiktok." A common hashtag pattern is a specific interest followed by "tiktok" to push the content to the For You Page of users that enjoy this content.

The fourth post demonstrates a common strategy of using the hashtag "for you page" with different variations, which is a common technique to push content on users' For You Page. These variations are processed from its original meaning which is "For You Page."

The fifth post is from an account dedicated to publishing videos commonly found on users' For You Page, however, does not use hashtags to increase reach. Despite this, the account has strong account engagement metrics although the post is not garnering the same attraction.

Overall, the five highlighted data points in Table 1.4 and 1.5 represent a unique and recurring characteristic in the dataset and offer insights into the current strategies on TikTok for marketing, promotion and high engagement.

Challenges

The main challenge in the data preprocessing phase was securing a large and relevant dataset and encoding strings. To begin, finding or curating a large dataset to train a machine learning model that coincides with the research question is difficult. Given the project's time constraints, it was impractical to compile a dataset using the TikTok API. Moreover, there were limitations to the API of how much and how often data could be collected, instead, an existing dataset of TikTok data was used. Additionally, due to the limitations of the API and time constraints, the model is not periodically updated with the most current TikTok data, potentially impacting the ability to provide accurate analytics as the TikTok algorithm evolves. Another significant challenge was encoding hashtags, which had many unique and frequent hashtags. Many hashtags share similar meanings but with abbreviations or spelling variations which adds complexity to encoding. To address this challenge, the hashtags were encoded using word2vec embeddings. The accuracy and effectiveness of these embeddings will be evaluated during the model training phase.

Next Steps

The next phase of the project involved training a model using the preprocessed datasets. Initially, simpler models were trained and complexity was gradually increased with models that closely align with the research objectives. This strategy tested the quality of the dataset and required adjusting of the processing methods of features to improve accuracy. Developing an accurate model that fulfilled the specified acceptance criteria required many iterations and various algorithms to optimize the model's performance such as Random Forest Regression, Linear Regression and Decision Tree Regression. The final model was somewhat capable of identifying trends and features that make a viral post and high engagement levels. Moreover, the models predicted multiple features such as determining the genre by country. This model provided insights to assist organizations in creating a strategic marketing campaign on TikTok.

Conclusion

In conclusion, the preprocessing phase of the machine learning lifecycle is complete. This phase was critical to the success of the model as quality foundational datasets are needed to develop an accurate machine-learning model. This stage involved developing a theoretical knowledge of machine learning and preprocessing a dataset specifically using cleaning, scaling and encoding data techniques for various types of data. Many challenges arose, notably the limitations of the TikTok API, which was mitigated through the use of an existing dataset. Significant progress has been made in the development of TikTrend to assist companies in optimizing their digital marketing campaigns and creating content that achieves high user engagement. TikTok is a global platform that garners millions of creators and viewers, with thousands of ongoing trends with record-breaking levels of engagement. By leveraging machine learning, TikTrend aims to accurately predict these future trends.

Phase 2 of TikTrend:

Model Development

Machine Learning Platforms

There is a wide selection of open-source machine learning tools available, each with unique features. Three tools were considered for developing TikTrend: Keras, PyTorch, and scikit-learn. Keras was developed by Google engineer François Chollet and has a focus on modern deep learning, providing an approachable and highly productive interface for solving machine learning problems (TensorFlow, 2019). PyTorch was developed by Facebook and is known for its dynamic computation graph, quick prototyping, and flexibility (Pykes, 2023). Scikit-learn is a traditional machine-learning library and specializes in classical machine-learning algorithms such as regression, clustering, and classification (Loobuyck, 2020).

There are two ideal tools for developing TikTrend: PyTorch and scikit-learn. PyTorch was considered because of its dynamic nature, flexibility, and strong adoption in the research community (Pykes, 2023). PyTorch has features to compute graphs dynamically, allowing for modifications of the model during runtime, which is an asset for experimentation (Loobuyck, 2020). Moreover, PyTorch has a user-friendly API, concise commands and comprehensive resources available (Loobuyck, 2020). Despite these advantages, scikit-learn was selected for TikTrend as the project only required the use of classical machine learning algorithms, which scikit-learn is known for. Scikit-learn has a straightforward interface allowing for rapid prototyping, an asset given the project's time constraints.

Machine Learning Techniques

This section introduces various machine learning concepts and algorithms applied in the development of TikTrend. It provides an overview of the techniques used to analyze and predict trends on TikTok.

Classification and Regression

Classification problems involve predicting categorical labels based on input data, resulting in a discrete value or category (Keita, 2022). Regression problems involve predicting numerical values based on input data, resulting in a continuous output (Keita, 2022).

Linear Regression

Linear regression is an algorithm that provides a linear relationship between independent and dependent variables (IBM, 2024). This is also known as a linear model. This algorithm helps explain the relationship between input and output variables. In linear regression, the dependent variable changes based on the independent variable, with the purpose of the regression model being to predict the dependent variable's value (IBM, 2024). A single independent variable is referred to as simple linear regression, while multiple independent variables are referred to as multiple linear regression (Brownlee, 2023). Linear regression is best suited when at least two variables are available in the data and the data is numerical for example, market forecasting, scientific analysis and portfolio management (IBM, 2024).

Linear regression is represented by a linear equation, y = mx + b. This equation can be broken down into its singular variables, with y representing the variable being predicted (dependent variable), x representing the variable that is being used to make predictions (independent variable), m representing the change in y for a single unit change in x (the slope of the line), and b representing the value of y when x is 0 (y-intercept). In higher dimensions, the line is called a plane or a hyper-plane when there is more than one input x (Brownlee, 2023).

Decision Trees

Decision trees are a type of supervised learning algorithm that is commonly used in machine learning to model and predict outcomes based on input data, specifically for classification and regression tasks (GeeksforGeeks, 2024). This algorithm is extremely effective due to its ability to lay out all the possible outcomes for a given problem (Coursera, 2023).

Decision trees are hierarchical models that consist of a root node, branches, internal nodes, and leaf nodes (Saini, 2024). The tree starts with the root node, where the entire dataset starts dividing based on various features or conditions (Saini, 2024). The nodes that are created from the initial split are known as decision nodes, as they represent intermediate decisions or conditions within the tree (Saini, 2024). A specific path of decisions is known as a branch, with the final node in a branch being a leaf node, a node where further splitting is not possible (Saini, 2024).

Random Forest

Random forest is a machine-learning algorithm that combines the output of multiple decision trees to reach a single result (IBM, 2021). This algorithm uses the bagging technique, where a random sample of data in a training set is selected with replacement and the individual data points can be chosen more than once (IBM, 2021). Then, each subset of data is used to train a different tree (IBM, 2021). Additionally, the forest algorithm utilizes feature randomness, which generates a random subset of features with low correlation among decision trees (IBM, 2021). Both techniques are combined to create an uncorrelated forest of decision trees, which can be used to solve regression or classification problems (IBM, 2021).

Gradient Boosting Regressor

Gradient boosting regressor is typically used for a tabular dataset to identify nonlinear relationships between features and target variables of a model (Masui, 2022). This technique sequentially builds decision trees, where each tree corrects the error of the predecessor, thus creating multiple weak models and combining them to create a strong predictive model (Masui, 2022).

Support Vector Regression (SVR)

Support vector regression (SVR) is a type of support vector machine (SVM) that is used to predict continuous outputs (Sethi, 2020). SVR is effective for handling complex linear and non-linear relationships by using different kernel functions (Sethi, 2020). This machine-learning algorithm identifies a hyperplane in a high-dimensional space that fits the data point within the defined margin of tolerance, which maximizes the distance between the hyperplane and the closest data point and minimizes error (Sethi, 2020).

eXtreme Gradient Boosting

eXtreme Gradient Boosting (XGBoost) is a robust machine learning algorithm known for its speed and performance (Nvidia, 2024). This model builds decision trees, where each tree corrects the errors of the previous ones, optimizing both accuracy and computational efficiency (Nvidia, 2024). XGBoost is used for its scalability and effectiveness in handling large datasets and complex models (Nvidia, 2024).

Models

This section provides an overview of the six machine learning models developed to analyze TikTok posts and accounts, to understand the factors that drive engagement. Each model highlights a critical feature in the top TikTok accounts or TikTok posts datasets, to identify valuable insights for strategic content creation and advertising. Three models focus on the top TikTok accounts, while the other three analyze TikTok posts. Each model was trained using the five-fold cross-validation method and the training results of the top machine learning algorithms for each model are analyzed.

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
random_state=42)
```

Top TikTok Account Dataset

All below models were developed using five-fold cross-validation and a training test split of 80/20, with 80 percent of the data used for training and 20 percent for testing. A random state of 42 is used to control the shuffling applied to the data when running our model, allowing for consistent results each time the model is run.

Model 1.0 - TikTok Account Location by Country

The purpose of model 1.0 is to predict the location of a TikTok account by analyzing engagement metrics. These metrics are the number of followers, number of views, number of likes, and account engagement. Model 1.0 was created to understand the impact of target audiences based on their residing country. Figure 1.1 showcases the impact that an account’s country of residence has on their respective view count.

Two machine learning algorithms were used to build the best possible model: linear regression and random forest classifier.

The linear regression model was used with no specific parameters specified, simply just a default model. The model showed strong performance levels under linear regression which was surprising as this was a classification problem. Linear regression is typically unsuitable for classification as it is ideal for working with continuous values whereas classification problems require discrete values (Kumar, 2021).

```
model = LinearRegression()
```

The random forest classifier was optimized using Grid Search CV to find the most optimal parameters. The code snippet below showcases the input parameters that produced the best results. To begin with, the random state controls the shuffling applied to the data when running our model against testing and training datasets (Modasiya, 2022). This produces the same result for any one integer value passed in, meaning that each time the model is run, the result will always be the same as long as the random state value is the same (Modasiya, 2022). Maximum depth is defined as the longest path between the root node and the leaf node (Saxena, 2023). This means that depth of each tree can be determined (Saxena, 2023). Minimum sample splits define the minimum required number of observations in any given node to split it (Saxena, 2023). By increasing the value of the min_sample_split, it reduces the number of splits that happen in the decision tree, preventing overfitting (Saxena, 2023). The minimum sample leaf specifies the minimum number of samples that should be present in the leaf node after splitting a node (Saxena, 2023). This model showed decent predictive performances which was expected as the random forest classifier performs best when used for classification problems.

```
model = RandomForestClassifier(random_state=42, max_depth=5, min_samples_split=5,
min_samples_leaf=3)
```

Model 1.1 - TikTok Account # of Followers

Account followers are a key metric on social media platforms, representing the number of people who follow an account to stay updated on its content. On TikTok, this indicates the audience size and engagement level with the account's content. The purpose of model 1.1 is to predict the number of followers of a TikTok account by analyzing engagement metrics of the number of total account views, number of total account likes and account engagement.

Two machine learning algorithms were trialled to build the best possible model: linear regression and random forest regressor.

Similar to model 1.0, the linear regression model was used with no specific parameters specified, simply just a default model. The model did not show strong performance levels under linear regression, which was unexpected due to the algorithm being suitable for the problem set. Linear regression is best suited when the data is numerical which was the case for both our x and y. The poor performance could have occurred due to various reasons. An in-depth analysis of the model performance can be found in the **Training Results** section.

```
model = LinearRegression()
```

Random forest regressor was trialled for the second version of model 1.1, with the algorithm being optimized using Grid Search CV to find the most optimal parameters. The most optimal values were a maximum depth of five, a minimum sample split of three, and a minimum sample leaf of five. Random forest regressor was used in this scenario as a numerical value was being predicted based on various numerical input data. This model did not perform exceedingly well, producing similar results to the linear regression model.

```
model = RandomForestClassifier(random_state=42, max_depth=5, min_samples_split=3,
min_samples_leaf=5)
```

Model 1.2 - TikTok Account Genre

All TikTok accounts have a designated genre in which their content revolves around. Accounts can contain videos on topics such as dancing, comedy and motivational speaking. Model 1.2 was developed to predict the genre of a TikTik account based on engagement metrics of number of account followers, number of total account views, number of total account likes and account engagement.

Two machine learning algorithms were trialled to build the best possible model: linear regression and random forest regressor.

This linear regression model was used with no specific parameters specified, and this model did not show strong performance levels under this algorithm. This is expected as linear regression does not work well with discrete values.

```
model = LinearRegression()
```

A random forest classifier algorithm was used for model 1.2 due to its strengths with classification problems. Grid Search CV was used to find the most optimal parameters. The most optimal values were a maximum depth of five, a minimum sample split of five, and a minimum sample leaf of three.

```
randomForest = RandomForestClassifier(random_state=42, max_depth=5,
min_samples_split=5, min_samples_leaf=3)
```

Top TikTok Post Dataset

Model 1.3 - TikTok Post # of Views

The purpose of the machine learning model is to predict the number of views a TikTok post would garner by analyzing other engagement metrics specifically, the number of likes, shares and followers a TikTok account had. This model was made to address one of the most critical needs of marketing executives which is high content visibility. Figure 1.4 demonstrates that there is a linear relationship between the number of views and shares, thus initial viewership is necessary to extend outreach.

Three machine learning algorithms demonstrated strong results for this problem: linear regression, random forest regression and support vector regression.

The linear regression model was optimized using Randomized Search CV to identify the best-fit intercept, which was found to be true. This result was expected as there is a baseline level of views influenced by the engagement metrics.

```
randomSearch = RandomizedSearchCV(pipeline, param_distributions=param_dist,
n_iter=10, cv=5)
```

There were many iterations of the random forest regression model and the most accurate model had a maximum depth of 10 and a random state of 100. These parameters prevented overfitting while capturing the interactions with the independent and dependent variables.

```
randomForest = RandomForestRegressor(max_depth = 10, random_state=100)
```

Support vector regression (SVR) demonstrated strong predictive capabilities, specifically using the default radial basis function (RBF) kernel. This kernel is effective in handling non-linear relationships within the data.

```
svr = SVR(kernel='rbf')
```

Model 1.4 - TikTok Post Hashtags

Many factors drive engagement on TikTok posts, and predicting the popularity of content can provide invaluable insights for marketers and content creators. Model 1.4 leverages the hashtags and number of comments of a post to predict the number of likes a post garners. The hashtags were transformed into vector embeddings using Word2Vec refer to page 15.

It is critical to select machine learning algorithms that can effectively train on vector embeddings. The ability to handle high-dimensional data, model non-linear relationships, and manage outliers is essential for achieving a high-performing model. Two machine learning algorithms demonstrated promising results for this model: random forest regression and eXtreme gradient boosting (XGBoost) algorithm for regression.

Random forest regression was selected for this model as this algorithm can handle complex, non-linear relationships between hashtags, comments and likes. Moreover, this algorithm is well-suited against outliers and less sensitive to overfitting, thus can provide reliable predictions. Specific parameters were selected for the number of decision trees, the depth of a tree, the minimum number of samples, the minimum number of samples for a leaf node and the seed for the random number generator. Collectively, these parameters produced strong results and reduced overfitting.

```
randomForest = RandomForestRegressor(n_estimators=200, max_depth=15,
min_samples_split=10, min_samples_leaf=4, random_state=42)
```

XGBoost regression was selected because it works well with vector embeddings as an input feature due to its ability to handle high-dimensional data and non-linear relationships (Brownlee, 2021). The following parameter values were selected as they produced the best results: number of decision trees, the depth of the tree, step size of each boosting iteration, the subsample, fraction of features to be randomly sampled for each tree, alpha, and the seed for the random number generator.

```
xgb = XGBRegressor(n_estimators=500, max_depth=5, learning_rate=0.01,
subsample=0.8, colsample_bytree=0.8, alpha=0.1, random_state=100)
```

Model 1.5 - Predicting Engagement Metrics Based on Time Published

Many social media platforms, including TikTok, have specific times of the day, week and month where posts garner maximum engagement (Strapagiel, 2024). The purpose of this model is to identify the optimal times to publish TikTok posts to receive a high number of likes. This model analyzes the day of the week, month and the number of views the TikTok has received to predict the number of likes a post will receive. These insights will assist in posting TikToks strategically to maximize reach and impact.

Three machine learning algorithms produced the best results for this model: linear regression, random forest regression, and gradient boosting regression.

Linear regression will identify the linear relationship between the input features and a continuous target variable. This will provide a baseline to understand the linear correlations between hashtags and the number of likes and comments. No parameters were specified and the default model was used.

```
linearRegression = LinearRegression()
```

Random forest regression was selected, and the following parameters were used to produce strong results: the number of decision trees, the depth of each tree, the minimum number of samples required to split a node, the minimum number of samples required for a leaf node, and the seed for the random number generator.

```
randomForest = RandomForestRegressor(n_estimators=100, max_depth=10,
min_samples_split=5, min_samples_leaf=2, random_state=42)
```

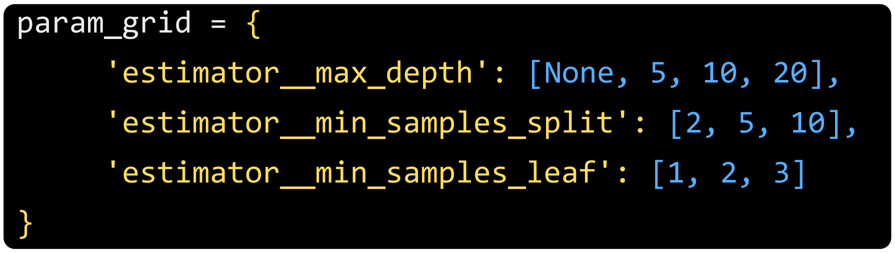
The gradient boosting regressor was selected and produced strong results because of its ability to reduce overfitting, handle outliers, and its ability to offer insight into feature importance. The following parameters customtized as it produced the best results: the number of decision trees, the depth of the tree, the step size of each boosting iteration, and the seed for the random number generator.

```
gradientBoost = GradientBoostingRegressor(n_estimators=100, max_depth=3,
learning_rate=0.1, random_state=42)
```

Model Optimization

Model tuning is the process of configuring a model's input parameters to guide its learning, leading to the most accurate and effective model (Shah, 2021). Once an algorithm has been selected, model accuracies will differ based on these input parameters. These parameters can decrease, increase or have no effect on model statistics.

Multiple algorithm variations were tested throughout the model training process using grid search. Grid search is an algorithm that takes in multiple variations of input parameters and exhaustively tries all input parameter combinations (Shah, 2021). The model performance for each possible combination is evaluated and the best model is chosen and returned. For example, Grid Search CV was used to find the most optimal model for model 1.0.



```
param_grid = {
     'estimator__max_depth': [None, 5, 10, 20],
     'estimator__min_samples_split': [2, 5, 10],
     'estimator__min_samples_leaf': [1, 2, 3]
}
```

The above array of possible parameters was given to each possible input value, with grid search trialling every possible combination. The highest accuracy for model 1.0 occurred with the parameters of max_depth of 5, min_samples_split of 5, and min_samples_leaf of 3.

Models were also optimized and tuned by varying the number of features passed in as independent variables. Adding more features that contain relevant data can improve model accuracy as it provides the model with more information to learn from. On the other hand, adding too many features may cause overfitting, and having too few features may cause underfitting. Experimenting with the number of features passed in and the quality of data given showcased how model accuracy can have a wide range of possible values.

Training Visualizations

Top TikTok Account Dataset

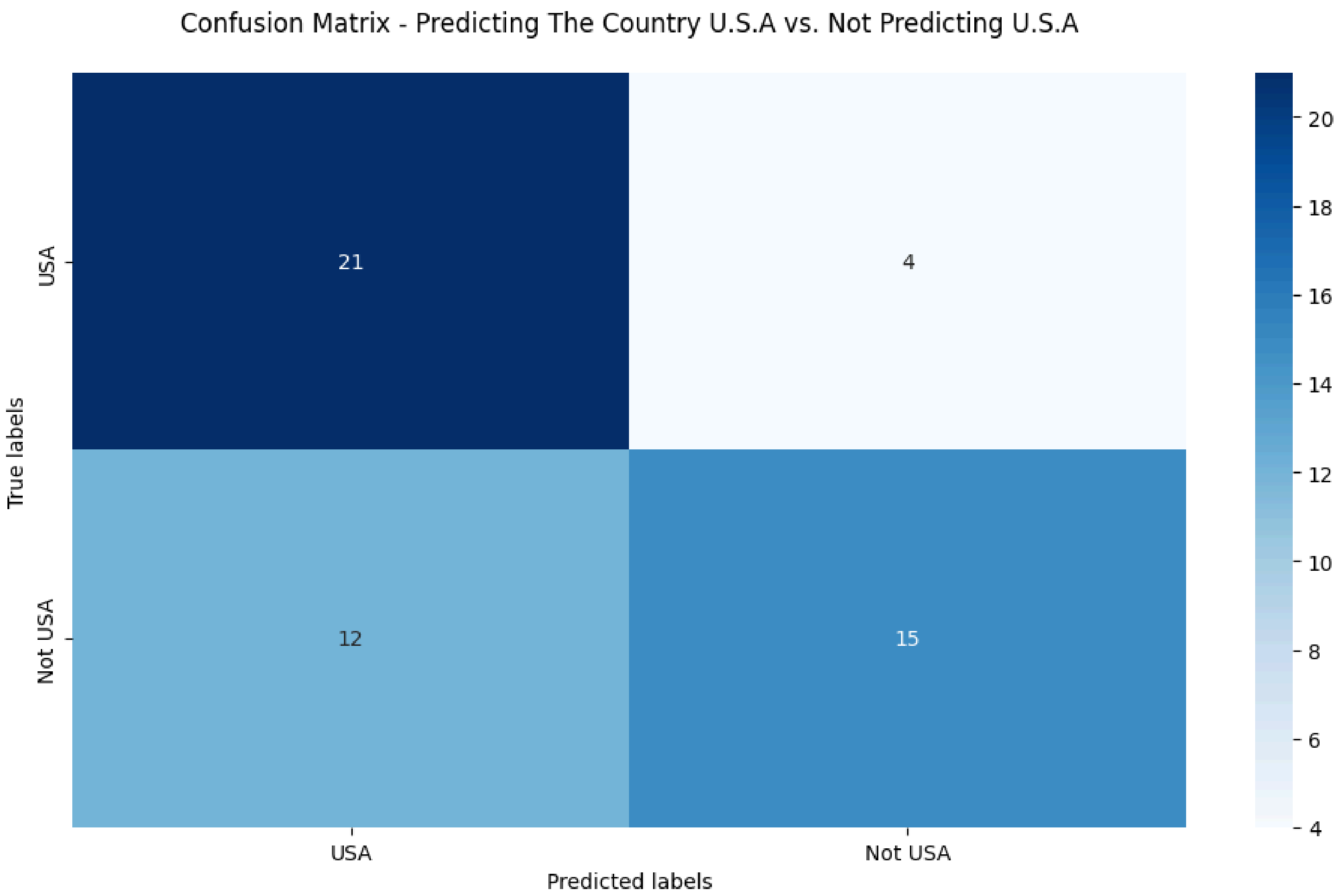


Fig. 2.0. A confusion matrix displaying model 1.0's ability to predict the country U.S.A

FIGURE 2.0

Figure 2.0 is a confusion matrix for model 1.0. A confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes (Suresh, 2020). The matrix compares the actual target values with those predicted by the machine learning model (Suresh, 2020). It displays the number of correct and incorrect predictions made by a classifier, showcasing the performance and accuracy of the model (Suresh, 2020).

Figure 2.0 is a matrix that showcases the ability to predict the country U.S.A, with the target classes being that the country is U.S.A or that the country is not U.S.A. The matrix is split into four quadrants with the top left being true positive, bottom left being false positive, top right being false negative, and bottom right being true negative. The following chart indicates the meaning of each quadrant and their values for model 1.0:

Table 2.0 - Confusion Matrix Figure 2.0 Analysis

Quadrant	Value	Analysis
True Positive (TP)	21	Based on Figure 2.0, Model 1.0 predicted that the country was U.S.A on 21 instances and the country was indeed U.S.A.
True Negative (TN)	15	Based on Figure 2.0, Model 1.0 predicted that the country was **not** U.S.A in 15 instances and the country **was correctly not** U.S.A.
False Positive (FP)	12	Based on Figure 2.0, Model 1.0 predicted that the country was U.S.A on 12 instances however, the country was **not U.S.A**. This is known as a Type I Error (Suresh, 2020).
False Negative (FN)	4	Based on Figure 2.0, Model 1.0 predicted that the country was **not** U.S.A on 4 instances however, the country **was indeed** U.S.A. This is known as a Type II Error (Suresh, 2020).

The model performance can be evaluated through the following statistics.

ACCURACY

TP + TN / TP + TN + FP + FN
= Correct Predictions / Total Predictions
= 21 + 15 / 21 + 15 + 12 + 4
= 36 / 52
= 0.69
= 69%

Accuracy is a measure of how often the classifier makes the correct prediction (Suresh, 2020). The accuracy shows that model 1.0 is currently predicting the country U.S.A correctly approximately 69 percent of the time

PRECISION

TP / TP + FP
= Predictions Actually Positive / Total Predicted Positive
= 21 / 21 + 12
= 21 / 33
= 0.63
= 63%

Precision is a measure of correctness that is achieved in true prediction. It essentially states how many predictions are actually positive out of all the total positive predicted (Suresh, 2020). The precision metric shows that when model 1.0 predicts the country U.S.A, it is correct about 63% of the time.

RECALL

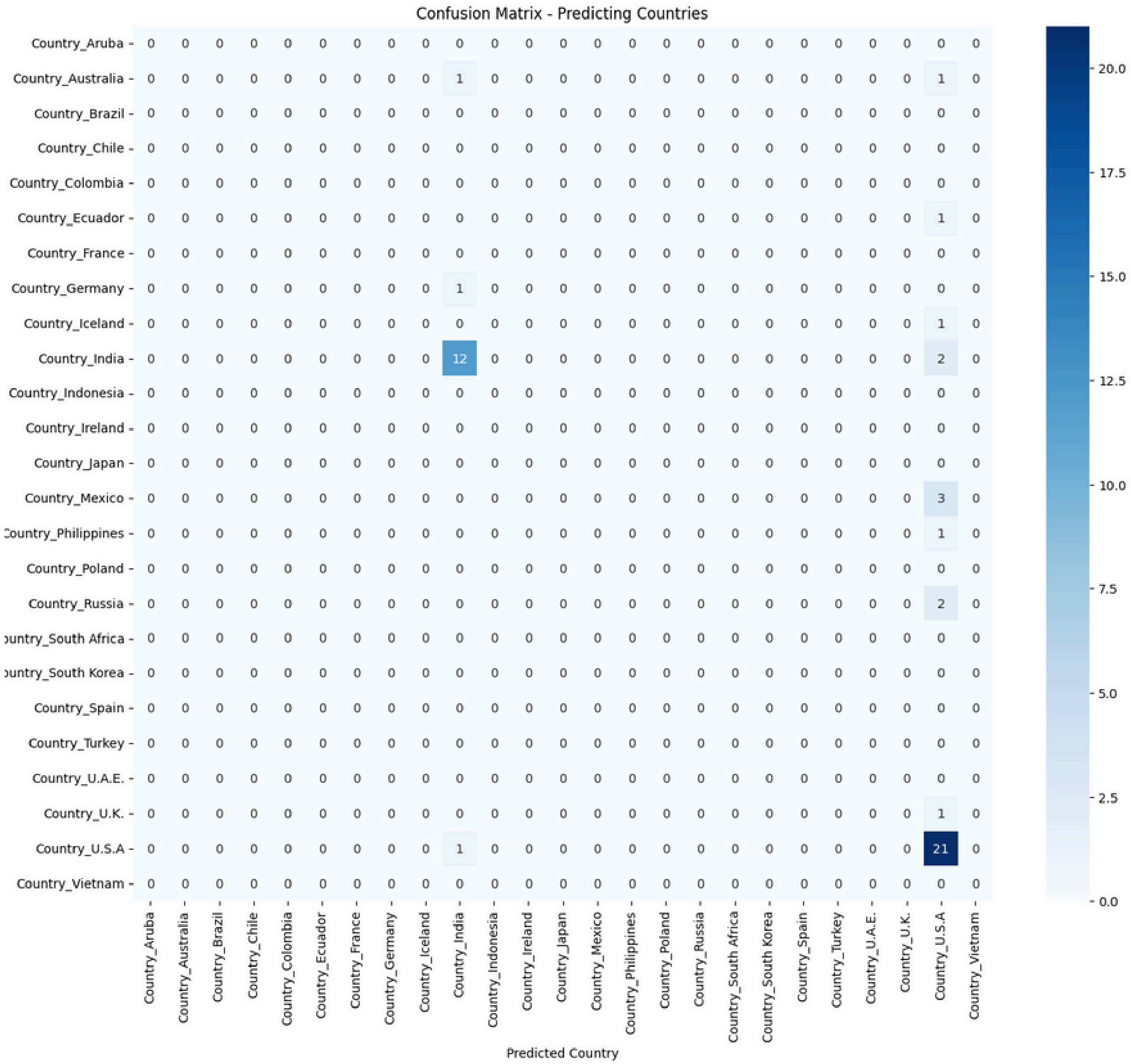
TP / TP + FN
= Predictions Actually Positive / Total Actual Positive
= 21 / 21 + 4
= 0.84
= 84%

Recall is a measure of actual observations which are predicted correctly (Suresh, 2020). The recall metric shows that model 1.0 correctly identified 84 percent of the actual U.S.A. instances.

F1-SCORE

2 * (Precision * Recall / Precision + Recall)
= 2 * (0.63 * 0.84 / 0.63 + 0.84)
= 2 * (0.36)
= 0.72

F1-Score is the harmonic mean of precision and recall (Suresh, 2020). This metric will always fall between 0 and 1 (Suresh, 2020). The F1-Score of 0.72 showcases model 1.0’s prediction accuracy. The model can accurately make predictions most of the time but there is still room for improvement.



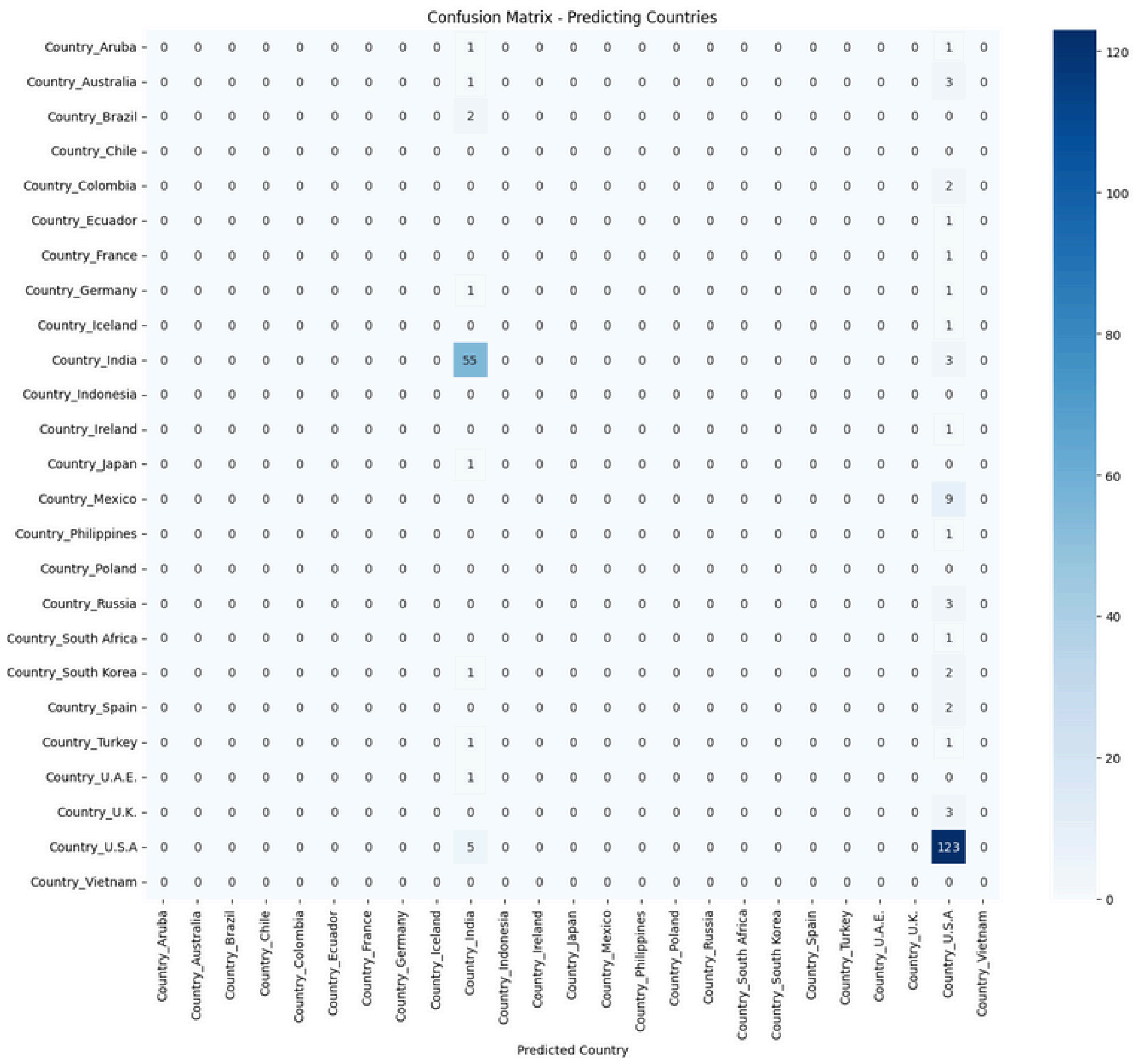
Confusion Matrix - Predicting Countries

	Country_Aruba	Country_Australia	Country_Brazil	Country_Chile	Country_Colombia	Country_Ecuador	Country_France	Country_Germany	Country_Iceland	Country_India	Country_Indonesia	Country_Ireland	Country_Japan	Country_Mexico	Country_Philippines	Country_Poland	Country_Russia	Country_South Africa	Country_South Korea	Country_Spain	Country_Turkey	Country_U.A.E.	Country_U.K.	Country_U.S.A	Country_Vietnam
Country_Aruba	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Australia	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_Brazil	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Chile	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Colombia	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Ecuador	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_France	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Germany	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Iceland	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_India	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
Country_Indonesia	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Ireland	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Japan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Mexico	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
Country_Philippines	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_Poland	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Russia	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
ountry_South Africa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ountry_South Korea	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Spain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Turkey	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_U.A.E.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_U.K.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_U.S.A	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0
Country_Vietnam	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 2.1. A confusion matrix displaying model 1.0's ability to predict various countries on a small subset of data

FIGURE 2.1

Figure 2.1 is a confusion matrix for model 1.0 that showcases its ability to predict a country on a test dataset, which is a small subset of data from the original dataset. Figure 2.1 showcases the accuracy of our model's predictions. Taking a look at the middle diagonal, from top left to bottom right, there were 12 instances where the country India was predicted correctly and 21 instances where the country U.S.A. was predicted correctly. There were multiple instances where the U.S.A. and India were both predicted incorrectly. For example, the U.S.A. was predicted by model 1.0 when the country was Russia in 2 instances. Many columns are filled with only 0's as these countries were not in the test dataset. Looking at the original account dataset, we can see that the majority of the data is from the U.S.A. and India, which is most likely why our matrix is displaying values from only those 2 countries. Our predictions may be biased due to the difference in the range of country values.



	Country_Aruba	Country_Australia	Country_Brazil	Country_Chile	Country_Colombia	Country_Ecuador	Country_France	Country_Germany	Country_Iceland	Country_India	Country_Indonesia	Country_Ireland	Country_Japan	Country_Mexico	Country_Philippines	Country_Poland	Country_Russia	Country_South Africa	Country_South Korea	Country_Spain	Country_Turkey	Country_U.A.E.	Country_U.K.	Country_U.S.A	Country_Vietnam
Country_Aruba	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_Australia	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
Country_Brazil	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Chile	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Colombia	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
Country_Ecuador	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_France	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_Germany	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_Iceland	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_India	0	0	0	0	0	0	0	0	0	55	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
Country_Indonesia	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Ireland	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_Japan	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Mexico	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0
Country_Philippines	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_Poland	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_Russia	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
Country_South Africa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_South Korea	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
Country_Spain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
Country_Turkey	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Country_U.A.E.	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Country_U.K.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
Country_U.S.A	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	123	0
Country_Vietnam	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 2.2. A confusion matrix displaying model 1.0's ability to predict various countries on the training subset

FIGURE 2.2

Figure 2.2 is a confusion matrix for model 1.0 that showcases its ability to predict a country on a training dataset. This subset is significantly larger than the set of data used for Figure 2.1, with almost every country having at least one data point in their respective row. There is a pattern with the model where it only predicts two countries: U.S.A. and India. No other countries are predicted for, with the U.S.A and India being predicted incorrectly on several occasions. For example, the model predicts the country U.S.A when the country is Mexico on 9 instances. The model predicts the country India in 2 instances when the country is Brazil. Taking a look at the middle diagonal, from top left to bottom right, there were 55 instances where the country India was predicted correctly and 123 instances where the country U.S.A. was predicted correctly. While the number of correct predictions for these countries heavily outweighs the number of incorrect predictions, Figure 2.2 showcases the model's inability to predict any other country. This problem may have occurred due to the original dataset being mainly made up of accounts from the U.S.A. and India, with 61 accounts being from India and 137 accounts being from the U.S.A.

Training Results

Top TikTok Account Dataset

MODEL 1.0

Both algorithms produced a similar accuracy, showing that model 1.0 is currently predicting the country correctly about 61 to 63 percent of the time. Linear regression produced a significantly better precision score than the random forest classifier, showing that the linear regression model predicts a country correctly 79 percent of the time when it predicts a positive class. Linear regression also produced a significantly higher recall score of 61 percent compared to the random forest classifier's score of 6 percent, with the recall score demonstrating the number of observations of positive classes that are predicted as positive. The F1-score is the harmonic mean of precision and recall and is a more accurate representation of a model's accuracy, with linear regression once again having a higher score of approximately 57 percent against the random forest classifier's 6 percent. Linear regression is the most effective algorithm for this model as it makes the most accurate predictions on a TikTok account's country of origin.

TABLE 2.1 - Performance Results Of Model 1.0

Metric	Linear Regression	Random Forest Classifier
Accuracy	0.6154	0.6346
Precision	0.7975	0.0575
Recall	0.6154	0.0679
F1-Score	0.5744	0.0621

Table 2.1 provides insight into the performance of Model 1.0 under the linear regression and random forest classifier algorithms. The following metrics were used to showcase the ability of the model under a classification problem: accuracy, precision, recall and F1-score.

TABLE 2.2 - Cross Validation Accuracy Scores

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
Accuracy	0.8077	0.6078	0.5686	0.4902	0.5294

Table 2.2 showcases the accuracies created by five-fold cross-validation through each iteration. The five accuracies that are calculated for each iteration showcase a wide range of possible accuracies of model performance against various unique subsets of data. Five-fold cross-validation is used on all models.

MODEL 1.1

MSE represents the average of the squared difference between the original and predicted values in the data set (Chugh, 2024), with both algorithms producing low MSEs indicating that few errors are being made. R-squared on the other hand is a statistical measure of how well the regression line approximates the actual data (Newcastle University, 2023). Random forest regression produced a slightly higher R^2 showcasing that this algorithm approximates the actual data better than linear regression. RMSE is simply the square root of the MSE, with both algorithms producing almost the same value. MAE is the average of the absolute difference between the actual and predicted values in the dataset (Chugh, 2024). It is essentially a measure of the average size of the mistakes in a collection of predictions (Deepchecks, 2024), with both algorithms producing a low score of approximately 0.15. When comparing both algorithms, random forest regression outperforms linear regression slightly and can be considered the stronger algorithm for this model.

TABLE 2.3 - Performance Results of Model 1.1

Metric (Mean, Std Dev)	Linear Regression	Random Forest Classifier
Mean Square Error	(0.0318, 0.0318)	(0.0293, 0.0286)
R-squared	(0.1514, 0.0508)	(0.2189, 0.2826)
Root Mean Square Error	(0.1785, 0.1784)	(0.1713, 0.1690)
Mean Absolute Error	(0.1578, 0.0842)	(0.1504, 0.0827)

MODEL 1.2

Looking at the accuracy, both algorithms produced extremely low percentages of approximately 3 percent, showing overall very weak predictive ability. Precision scores were average, with the random forest classifier predicting a genre correctly 52 percent of the time when predicting a positive class, compared to linear regression's 40 percent. Both algorithms produced poor recall scores with the random forest classifier having the higher score by approximately 11 percent. The F1-score of both models were poor, demonstrating the overall poor performance of both models. In conclusion, both models did not perform well and were unable to learn and understand the TikTok account dataset.

TABLE 2.4 - Performance Results for Model 1.2

Metric	Linear Regression	Random Forest Classifier
Accuracy	0.0385	0.0384
Precision	0.4091	0.5263
Recall	0.0865	0.1923
F1-Score	0.1429	0.2817

Table 2.4 demonstrates the performance of Model 1.2 under the linear regression and random forest classifier algorithms, with the following metrics being used to showcase the ability of the model under a classification problem: accuracy, precision, recall and F1-score.

Top TikTok Account Dataset

MODEL 1.3

Model 1.3 was successfully trained using three machine-learning algorithms: linear regression, random forest regression, and support vector regression. Each algorithm was evaluated and a comprehensive overview of the results is in Table 2.5. After evaluating the performance of all the models, random forest regression demonstrated the most consistent and accurate performance. Figure 2.3 and Figure 2.5 demonstrated strong results of MSE and RMSE, in addition to achieving the highest R^2 as seen in Figure 2.4. Figure 2.6 demonstrates the MAE and confirms random forest regression's precision, as it consistently produced the lowest absolute errors.

Although support vector regression showed competitive results with a close second in most metrics, and linear regression performed reliably, random forest regression stood out as the best-performing model overall due to its balance of low error rates and consistent performance. This result is unsurprising as random forest regression is best for complex interactions which is the case for predicting the number of views a TikTok post would garner.

TABLE 2.5 - Performance Results Of Model 1.3

Metric (Mean, Std Dev)	Linear Regression	Random Forest Classifier	Support Vector Regression
Mean Square Error	(0.2302, 0.0066)	(0.2128, 0.0034)	(0.2253, 0.0091)
R-squared	(0.9609, 0.0009)	(0.9639, 0.00053)	(0.9618, 0.0009)
Root Mean Square Error	(0.4797, 0.0069)	(0.4613, 0.0037)	(0.4745, 0.0094)
Mean Absolute Error	(0.3708, 0.0044)	(0.3556, 0.0036)	(0.3578, 0.0048)

Table 2.5 is a performance comparison of linear regression, random forest regression, and support vector regression (SVR) based on key metrics. The table presents the mean and standard deviation for mean squared error (MSE), R-squared (R^2), root mean squared error (RMSE), and mean absolute error (MAE).

FIGURE 2.3

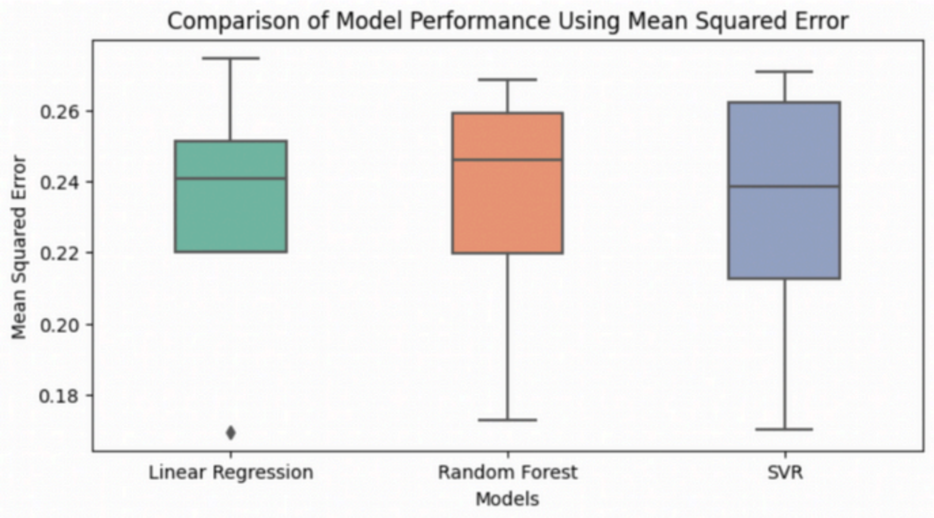


Figure 2.3 presents a box plot of the MSE of the machine learning algorithms used for Model 1.3. Linear regression has the lowest median and variability which indicates minimal errors and consistent performance. There is an outlier that suggests that this model performs better in certain cases. Random Forest performs reasonably well, however, SVR has the highest variability and the median indicates less reliable performance.

FIGURE 2.4

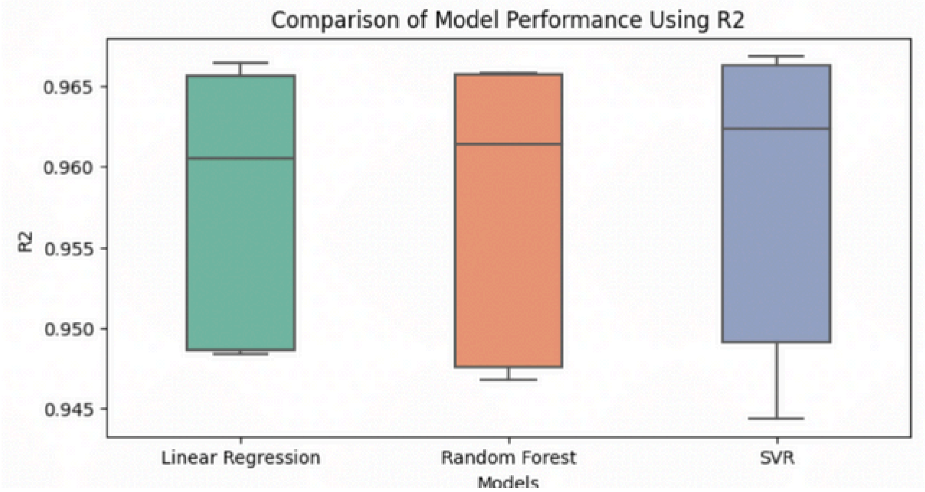


Figure 2.4 is a box plot of the R2 of the machine learning algorithms used for Model 1.3. Linear regression has a high performance in terms of R2, with a high median and moderate consistency. Random forest shows consistent performance with a slightly lower median R^2 but less variability. SVR has the lowest median R^2 and the highest variability in performance. Overall, linear regression and random forest both perform well, random forest is the most consistent model and SVR shows the least reliable performance in explaining the variance in the data.

FIGURE 2.5

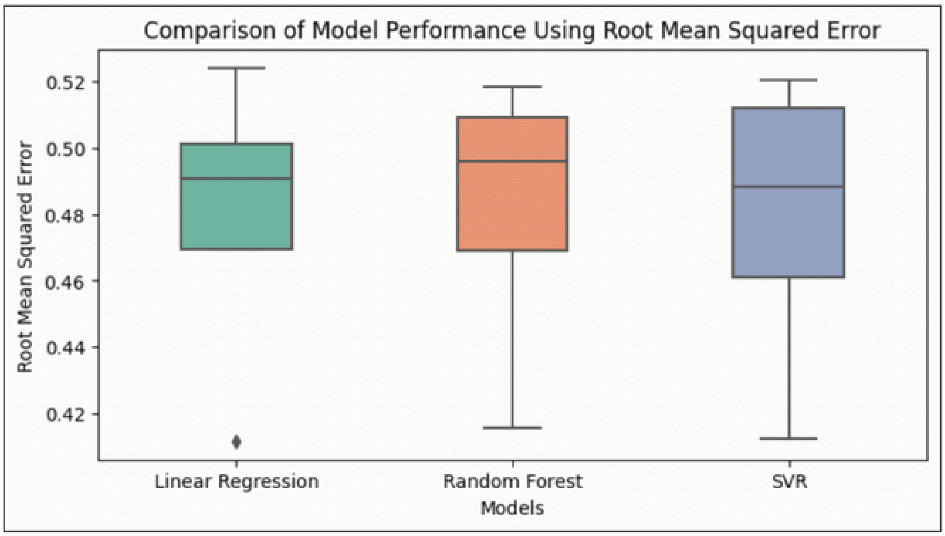


Figure 2.5 presents a box plot of the RMSE of the machine learning algorithms used for Model 1.3. Linear regression is the best model as it has the lowest median and consistent results. Moreover, Linear regression has an outlier suggesting that in some cases there is higher accuracy. Random forest shows slightly higher variability but still performs reasonably well. SVR has the highest variability and the highest median RMSE, indicating less reliable performance.

FIGURE 2.6

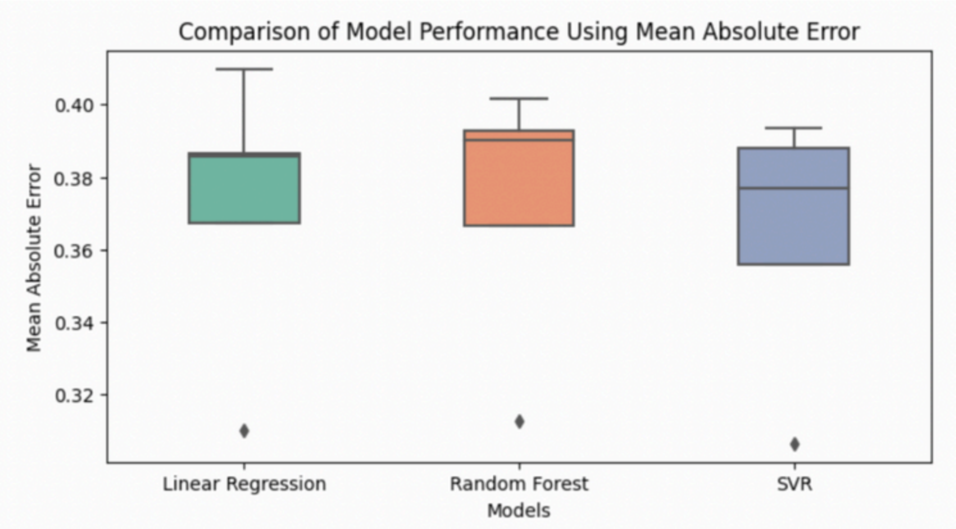


Figure 2.6 presents a box plot of the MAE of the machine learning algorithms used for Model 1.3. SVR has a strong performance in terms of MAE, with the lowest median and consistent results. Linear regression performs well with a slightly higher median MAE and consistent performance. Random forest performs similarly to linear regression but with slightly more variability. Overall, SVR is the best model in this comparison for minimizing mean absolute error and providing consistent performance.

MODEL 1.4

The performance of random forest regression and Gradient Boosting Regression models was assessed using several key metrics: mean squared error (MSE), R-squared (R^2), root mean squared error (RMSE), and mean absolute error (MAE). Table 2.6 provides an overview of the results of each model against the criterion. As demonstrated in Figure Figure 2.7 and Figure 2.9, gradient boosting regression had the overall best performance and accuracy. Both models had high R^2 scores as demonstrated in Figure 2.8, but gradient-boosting regression had a slightly higher score demonstrating effectiveness in understanding data variance. Figure 2.10 visualizes the MAE for both algorithms with gradient boosting regression having a lower score, thus having a higher precision. While both algorithms performed well, gradient boosting regression consistently outperformed it across all metrics and is the most effective model. By leveraging gradient boosting regression the number of likes of a TikTok post can be predicted using the number of views and hashtags used.

TABLE 2.6 - Performance Results Of Model 1.4

Metric (Mean, Std Dev)	Random Forest Regression	Gradient Boosting Regression
Mean Square Error	(0.0039, 0.0001)	(0.0038, 0.0002)
R-squared	(0.8099, 0.0234)	(0.8141,0.0216)
Root Mean Square Error	(0.0622, 0.0011)	(0.0615, 0.0012)
Mean Absolute Error	(0.0454, 0.0013)	(0.0449, 0.0014)

Table 2.6 is a performance comparison of random forest regression, and gradient boosting regression (SVR) based on key metrics. The table presents the mean and standard deviation for mean squared error (MSE), R-squared (R^2), root mean squared error (RMSE), and mean absolute error (MAE).

FIGURE 2.7

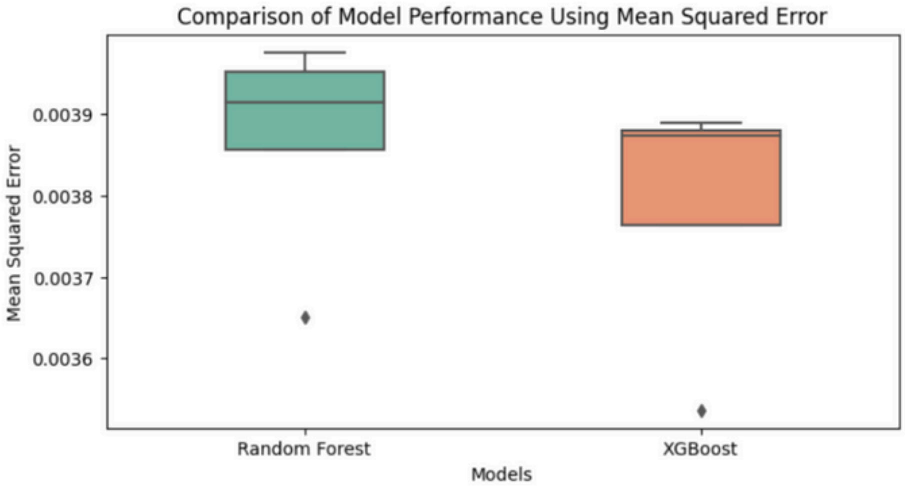


Figure 2.7 is a box plot of the MSE for each machine-learning algorithm of Model 1.4. Both models have outliers indicating that there is potential for low MSE in certain cases. The XGBoost regressor has a lower median than random forest and thus, has a better overall performance.

FIGURE 2.8

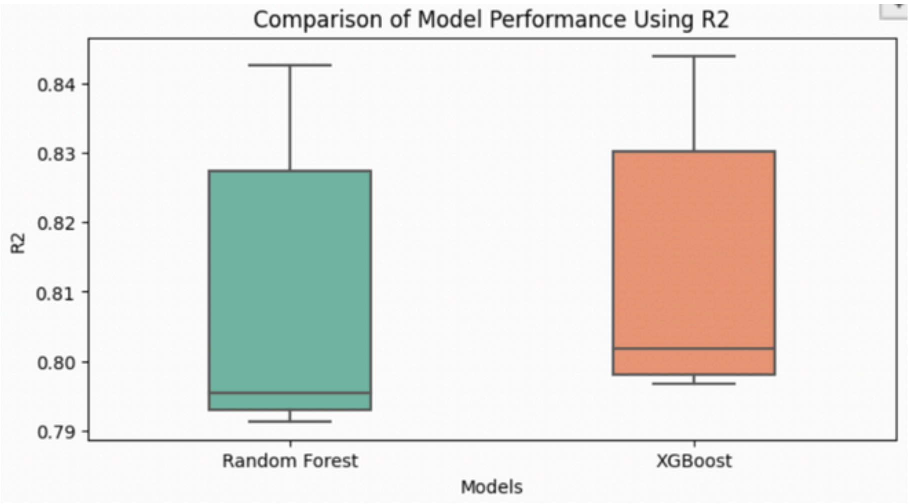


Figure 2.8 presents a box plot of the R2 for each machine-learning algorithm experimented for Model 1.4. Both algorithms have a wide range of R2 values, however, XGBoost has less variability. Overall, the XGBoost regressor performs better in terms of R2 due to its slightly higher median and more consistent results.

FIGURE 2.9

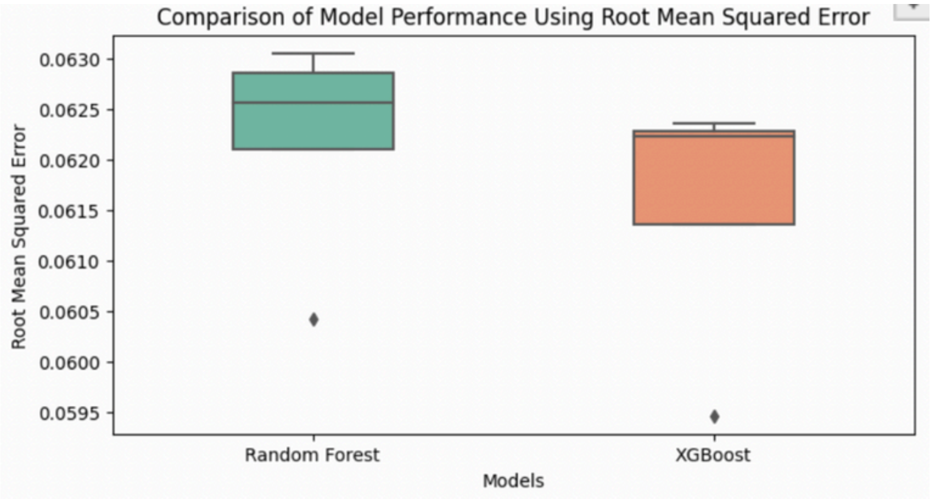


Figure 2.9 is a box plot of the RMSE for each machine-learning algorithm of Model 1.4. XGBoost regressor has a slightly lower median than random forest, indicating marginally better performance. Moreover, the XGBoost regressor has a smaller variability of RMSE, indicating more consistent results.

FIGURE 2.10

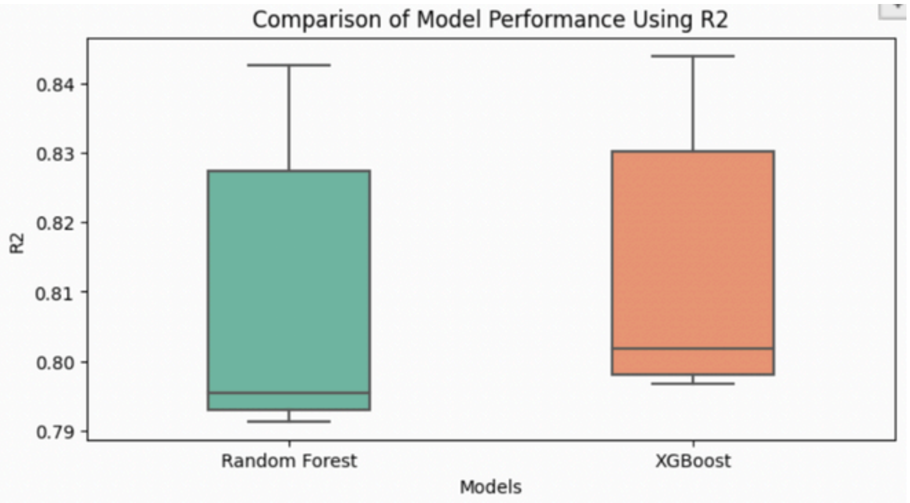


Figure 2.10 is a box plot of the MAE for random forest and XGBoost regressor of Model 1.4. XGBoost has a slightly lower median and less variability, indicating more consistent performance. XGBoost regressor has two outliers above and below the box indicating instances of better and worse performance. Random forest has an outlier below, indicating certain cases of better performance, however, it has a higher variability of MAE values. Overall, the XGBoost regressor has better MAE results.

Analysis of Hashtag Word2Vec Model

It was difficult to train accurate and strong models because of the quality and accuracy of the hashtag embeddings. The Word2Vec model of the hashtags of TikTok posts has a vocabulary size of 6413, however, a strong and accurate model requires a vast vocabulary. Table 2.7 provides an example of the Word2Vec model's performance in identifying the similarity between two common words found in the hashtags. Based on these results, the model's accuracy can be significantly improved as the similarity scores are poor. Simple hashtags were tested and the similarity scores should be higher for words such as funny and comedy. A comprehensive review of the Word2Vec model's ability to understand and identify semantic relationships is demonstrated in Table 2.8, where the model finds the most similar words in its vocabulary based on input. Table 2.8 indicates that some words perform better than others, but there were still many words that were unrelated or did not make sense, such as the results for funny. Many of these incorrect results had high similarity scores, further demonstrating that significant improvement is necessary to create an accurate model that meets the purpose of Model 1.4. The results of Model 1.4 are heavily impacted by the quality of the hashtag embeddings, which will be improved by expanding the vocabulary by using a larger dataset of hashtags and fine-tuning the Word2Vec model.

TABLE 2.7 - Word2Vec Similarity Score between Two Hashtags

Hashtag 1	Hashtag 2	Similarity Score
fitness	workout	0.7532
funny	comedy	0.6745
pet	dog	0.8486

Table 2.7 consists of two hashtags and their corresponding similarity score, which is calculated by measuring the cosine similarity between their word vectors. The cosine similarity is a value between -1 and 1 to indicate how similar the vectors are, where high similarity scores indicate the hashtags are semantically related and low scores have less in common semantically. This figure demonstrates the accuracy and effectiveness of the hashtag Word2Vec model in capturing word similarities and representing word semantics.

TABLE 2.8 - Word2Vec Similarity Score with All Hashtags in Dataset

Word	Similar Words (word, similarity score)
funny	('recipes', 0.9126) ('youtube', 0.9112) ('strange', 0.9111) ('tips', 0.9055)
food	('fashion', 0.9471) ('arab', 0.9436) ('chef', 0.9423) ('foodie', 0.9342)
love	('friend', 0.8731) ('couple', 0.8344) ('ns', 0.8299) ('boy', 0.8253)
fashion	('food', 0.9471) ('partner', 0.9469) ('chef', 0.9423) ('arab', 0.9404)

Table 2.8 illustrates the performance of the hashtag vector embeddings using the Word2Vec model. The table provides a list of similar words along with their similarity scores when a common word is given as input, demonstrating the model's accuracy in identifying and ranking related terms based on their vector representations. Each word has at least one relevant word with a high similarity score, many of the similar words are inaccurate. Different vector sizes for the Word2Vec model were tested, but they all had similar results, so the original vector size of 200 was used. These results may be due to the small size of the dataset.

MODEL 1.5

Model 1.5 was trained with the following machine learning algorithms: linear regression, random forest regression, and Gradient Boosting Regression. Table 2.9 provides a comprehensive evaluation of each model using Mean Squared Error (MSE), R-squared (R^2), Root Mean Squared Error (RMSE), and Mean Absolute Error. Figures 2.11 and 2.13 demonstrate that gradient boosting regression had the best overall performance with the lowest MSE and RMSE, indicating its accuracy in error minimization. This model achieved the highest R^2 as seen in Figure 2.12 indicating its effectiveness in explaining data variance. Figure 2.14 indicates that gradient boosting regression has the lowest MAE, thus it has high precision. Linear regression and random forest regression performed similarly, gradient boosting regression consistently outperformed them across all metrics, thus it’s the most effective model for this scenario.

TABLE 2.9 - Performance Results Of Model 1.5

Metric (Mean, Std Dev)	Linear Regression	Random Forest Classifier	Gradient Boosting Regression
Mean Square Error	(0.2323, 0.0346)	(0.2326, 0.0336)	(0.2253, 0.0333)
R-squared	(0.9602, 0.0078)	(0.9601, 0.0077)	(0.9614, 0.0075)
Root Mean Square Error	(0.4809, 0.0365)	(0.4812, 0.0355)	(0.4737, 0.0357)
Mean Absolute Error	(0.3723, 0.0286)	(0.3722, 0.0274)	(0.3669, 0.0275)

Table 2.9 provides an overview of the model training results for each machine learning algorithm for Model 1.5. Each algorithm is evaluated by the mean and standard deviation of the following metrics: mean square error, R2, root mean square error and mean absolute error. Based on this data, linear regression and random forest regression have similar performance metrics, with linear regression having a slightly lower mean in RMSE and MSE compared to random forest regression.

FIGURE 2.11

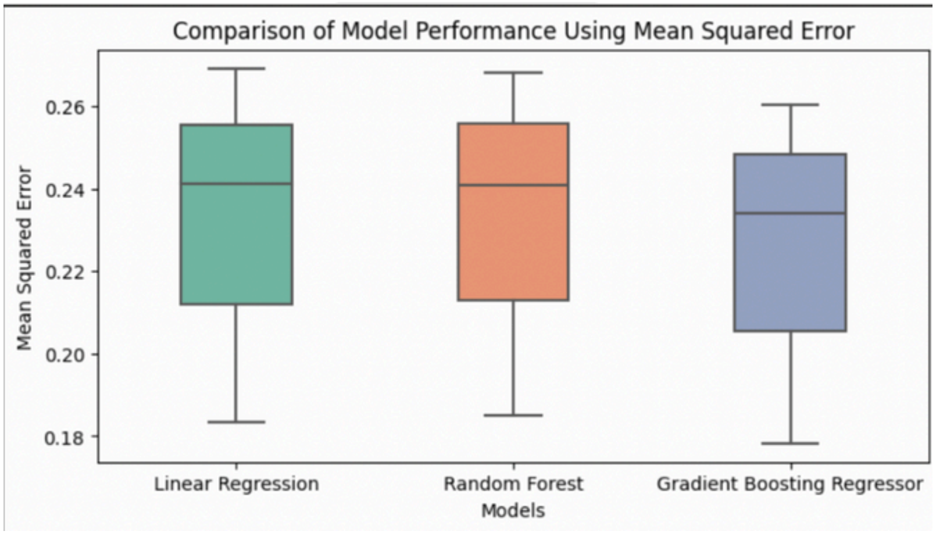


Figure 2.11 is a box plot of the MSE for each machine-learning algorithm of Model 1.5. The MSE of each model is very similar. The Gradient Boosting Regressor has the lowest median and the smallest range, indicating strong and consistent performance. Overall, each algorithm performed similarly, however, the gradient-boosting regressor outperforms the others.

FIGURE 2.12

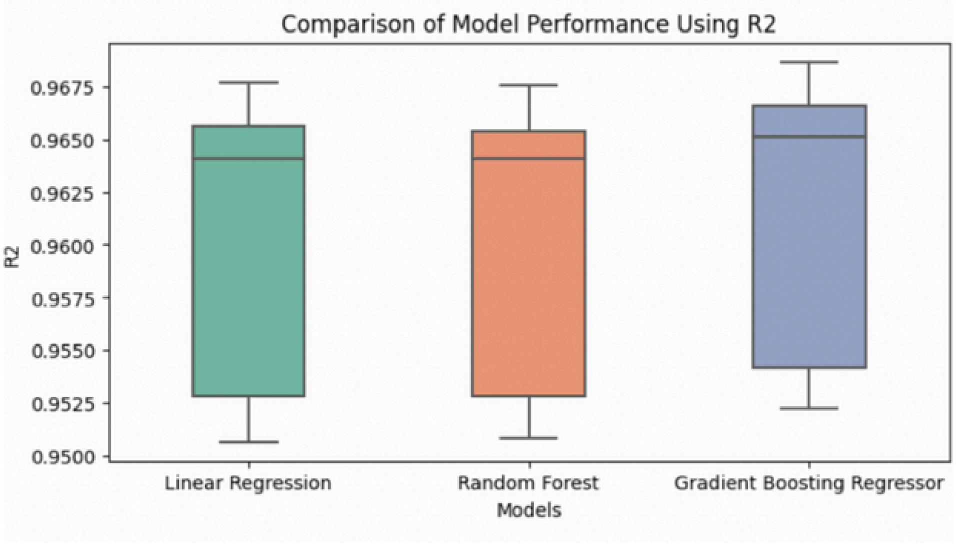


Figure 2.12 presents a box plot of the R2 for each machine-learning algorithm of Model 1.5. The medians of linear regression and random forest are very similar, but with linear regression being slightly higher. However, the gradient-boosting regressor has the highest median R2 value. Linear regression has the highest variability in the R2 values, while gradient boosting has a smaller range, indicating more consistent performance. Overall, each algorithm performed similarly, the gradient boosting regressor performed, as it has the highest median and most consistent performance.

FIGURE 2.13

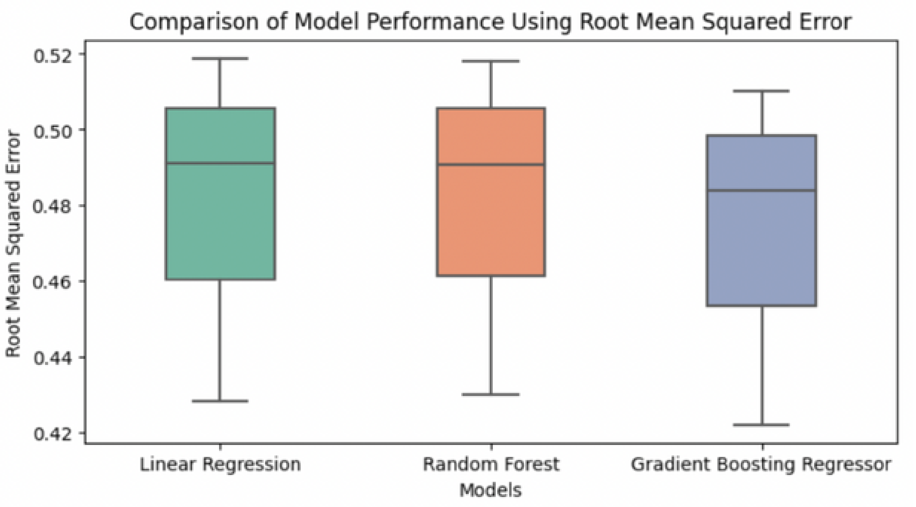


Figure 2.13 presents a box plot of the RSME for each machine-learning algorithm of Model 1.5. Random forest has a moderate range of RMSE and a slightly higher median than gradient boosting regressor. Linear regression has the highest variability and median than the other models. Gradient boosting regressor has the best performance in terms of RMSE as it has the lowest median and most consistent results.

FIGURE 2.14

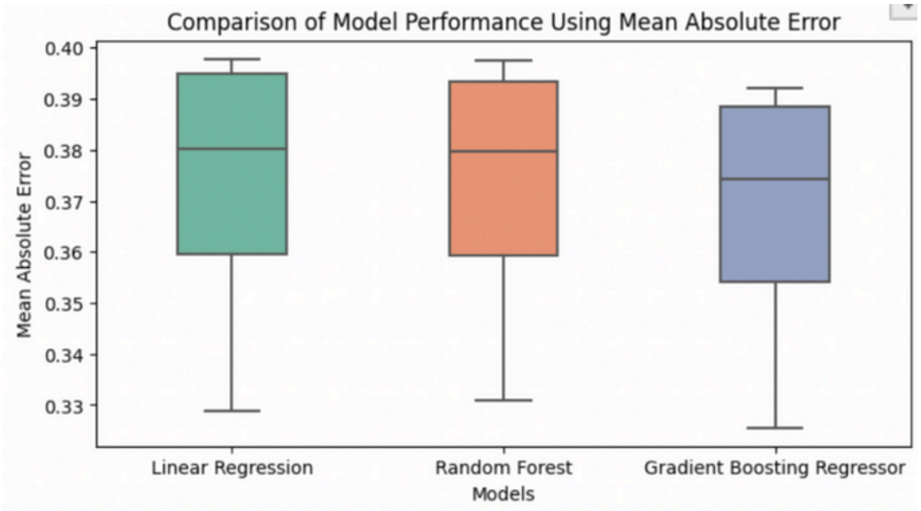


Figure 2.14 is a box plot of the MAE for each machine-learning algorithm of Model 1.5. Random forest performs well but has a slightly higher MAE and variability similar to linear regression. The gradient-boosting regressor has the best results as it has the lowest median and most consistent results.

Challenges

Throughout the second phase of TikTrend, numerous challenges were encountered while training and testing the models. To begin with, the main challenge discovered was poor model performance. Many models had average or sub-par performances after optimization which will lead to unreliable predictions. Models 1.1 and 1.2 showcased poor statistics as illustrated in the Training Results section. The models' performance was poor due to the size of the dataset. For example, the TikTok account dataset was extremely small, with only 256 rows of data. The data size may have limited our model's ability to learn leading to sub-par accuracies. Additionally, the quality of our dataset may have negatively impacted our model's performance, with the majority of our dataset being accounts from India and the U.S.A. This most likely resulted in overfitting with the dataset overly predicting for India and the U.S.A, as shown by Figures 2.1 and 2.2.

There were many challenges in the development of Model 1.4 due to the vector embeddings of the hashtags. These vector embeddings were developed using Word2Vec and the performance of the Word2Vec model was suboptimal in identifying similarities and relationships between words. The hashtags underwent multiple rounds of preprocessing by attempting different parameters on the Word2Vec model to improve the vector embeddings, however, the accuracy improvements were minimal and the original embeddings were used. This performance of the Word2Vec model is most likely due to the model's small vocabulary size and a larger vocabulary would yield stronger results. Furthermore, due to time constraints, more advanced and robust machine learning tools such as PyTorch and TensorFlow were not utilized, limiting the implementation to scikit-learn.

Future Plans

In the third phase of TikTrend, the focus will be on presenting the results and predictions of the models in a comprehensive report. This report will serve as a guide to demonstrate the capabilities and accuracy of the model using examples and results to potential users. The goal is to provide a comprehensive understanding of the models' performance and how they can be utilized effectively. In the next phase, the models will be deployed as a web application with an emphasis on a user-friendly interface to provide users with easy access to TikTrend. Future expansions of TikTrend include incorporating new TikTok data into the models weekly, improving the models' accuracy and providing a more comprehensive trend analysis by including additional engagement metrics.

Conclusion

To conclude, the model engineering and evaluation stages of the machine learning life cycle are complete. Developing machine learning models involved training and evaluating models, using various techniques to ensure the best accuracy in predicting dynamic trends on TikTok. There were many challenges in training the models because of the size of the datasets and time restrictions, resulting in some models having poor accuracy. Despite these challenges, the evaluation results were promising and demonstrated that the models have the potential to perform reliably. Training the models was a success and showcases the potential of TikTrend to revolutionize the digital marketing industry. Continuous refinement will make TikTrend an essential digital marketing tool, driving informed and strategic marketing campaigns.

Reflection

The development of TikTrend involved the first four stages of the machine learning lifecycle: planning, data preparation, model engineering and model evaluation. Each stage presents its own set of challenges, some of which were resolved swiftly and others required more time and adapting.

In the planning stage, the scope, goal, purpose and hypothesis of TikTrend were defined. Given the time constraints, the scope of TikTrend was too ambitious and a smaller scope would have produced a higher-quality model. This stage was executed smoothly, however, conducting preliminary research on how data would be collected, would have reduced the need for scope adjustments.

The data preparation stage was where the data was collected, which was the most challenging and time-consuming phase of the project. This stage included experimenting with APIs, web scrapers and searching for relevant datasets online. Collecting data was a difficult and time-consuming process because there were specific requirements for the data and the data needed to be large enough to train, to produce a high-quality dataset. Due to this challenge, an open-source dataset was selected, while suitable, it was slightly smaller than ideal and posed a threat to the model's performance.

The third stage of the machine learning lifecycle, model engineering, focused on the training and development of TikTrend's models. This phase required extensive research into various machine-learning algorithms and tools. Due to the difficulties in the data preparation stage, there was not enough time to allocate for the team to explore and utilize other model training tools, thus scikit-learn was used by default. This stage involved experimenting with various finetuning techniques and machine learning algorithms to create an optimized model.

The final stage completed was the model evaluation stage. This stage was executed seamlessly with minimal challenges with various tables and visualizations developed to present the findings. Unfortunately, the performance of most models was subpar, due to the size of the datasets, highlighting the importance of an appropriate amount of data for machine learning.

Reflection

Overall, the two greatest challenges faced during the development of TikTrend were finding an appropriate dataset that met the project’s requirements and managing the tight time constraints. The difficulties with finding an appropriate dataset resulted in less time available for the proceeding stages causing the scope to be reduced. The planning stage should have been approached with an understanding of the available data, which would have mitigated many challenges and allowed for a more seamless development process. This experience has highlighted the need for adaptability when dealing with unforeseen challenges, such as time constraints and data limitations. These insights will guide future projects, ensuring that they are more effectively planned and executed.

References

Awan, A. A. (2022, October). *The Machine Learning Life Cycle Explained*. Www.datacamp.com. https://www.datacamp.com/blog/machine-learning-lifecycle-explained

Bhandari, A. (2020, April 3). *Feature Scaling | Standardization Vs Normalization*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

Brown, S. (2021, April 21). *Machine learning, explained*. MIT Sloan; MIT Sloan School of Management. https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

Brownlee, J. (2021, March 11). *XGBoost for Regression*. Machine Learning Mastery. https://machinelearningmastery.com/xgboost-for-regression/

Burley, D. (2023, February 22). *Word2Vec Explained: How Computers Learned to Talk Like We Do!* AI Search Blog. https://www.coveo.com/blog/word2vec-explained/

Chugh, A. (2020, December 8). *MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better?* Medium. https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e

Coursera. (2023, November 29). *Decision Trees in Machine Learning: Two Types (+ Examples)*. Coursera. https://www.coursera.org/articles/decision-tree-machine-learning

Deepchecks. (2024a, May 27). *What is Mean Absolute Error?* Deepchecks. https://deepchecks.com/glossary/mean-absolute-error/#:~:text=Mean%20Absolute%20Error%20(MAE)%20is

Deepchecks. (2024b, June 12). *What is Overfitting in Machine Learning*. Deepchecks. https://deepchecks.com/glossary/overfitting-in-machine-learning/

Donia, O. (2023, March 9). *Data Scaling and Normalization: A Guide for Data Scientists*. Medium. https://generativeai.pub/data-scaling-and-normalization-a-guide-for-data-scientists-d6f9fdfa7b2d

Eugene Dorfman. (2022, March 24). *How Much Data Is Required for Machine Learning?* PostIndustria. https://postindustria.com/how-much-data-is-required-for-machine-learning/

Forbes Technology Council . (2023, July 28). *AI & Machine Learning: Identifying Opportunities & Challenges*. Councils.forbes.com. https://councils.forbes.com/blog/ai-and-machine-learning

GeeksforGeeks. (2024, March 15). *Decision Tree in Machine Learning*. GeeksforGeeks. https://www.geeksforgeeks.org/decision-tree-introduction-example/

IBM. (2021, October 20). *What is Random Forest?* Www.ibm.com. https://www.ibm.com/topics/random-forest

IBM. (2023). *What Is Machine Learning?* IBM. https://www.ibm.com/topics/machine-learning

IBM. (2024). *About Linear Regression*. Www.ibm.com; IBM. https://www.ibm.com/topics/linear-regression

Jason Brownlee. (2016, March 24). *Linear Regression for Machine Learning*. Machine Learning Mastery. https://machinelearningmastery.com/linear-regression-for-machine-learning/

Keita, Z. (2022, September 21). *Classification in Machine Learning: A Guide for Beginners*. DataCamp. https://www.datacamp.com/blog/classification-machine-learning

Kumar, A. (2021, February 5). *Why Linear Regression is not suitable for classification?* Analytics Vidhya. https://medium.com/analytics-vidhya/why-linear-regression-is-not-suitable-for-classification-cd724dd61cb8#:~:text=There%20are%20two%20things%20that

Lewinson, E. (2022, February 17). *Three Approaches to Encoding Time Information as Features for ML Models*. NVIDIA Technical Blog. https://developer.nvidia.com/blog/three-approaches-to-encoding-time-information-as-features-for-ml-models/

Loobuyck, U. (2020, May 28). *Scikit-learn, TensorFlow, PyTorch, Keras… but where to begin?* Towards Data Science. https://towardsdatascience.com/scikit-learn-tensorflow-pytorch-keras-but-where-to-begin-9b499e2547d0

Masui, T. (2022, February 12). *All You Need to Know about Gradient Boosting Algorithm − Part 1. Regression*. Medium. https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502

Ministry of Justice: Data Science Hub. (2024). *Neural Network Models*. GitHub. https://github.com/moj-analytical-services/NLP-guidance/blob/master/NNmodels.md

Modasiya, K. (2022, November 28). *What the heck is random_state?* Medium. https://kishanmodasiya.medium.com/what-the-heck-is-random-state-24a7a8389f3d

Narasimhan, K. A. (2021, January 21). *Why Linear Regression is not Suitable for Classification?* Analytics Vidhya.

https://medium.com/analytics-vidhya/why-linear-regression-is-not-suitable-for-classification-cd724dd61cb8#:~:text=There%20are%20two%20things%20that

Newcastle University. (2023). *Numeracy, Maths and Statistics - Academic Skills Kit*. Www.ncl.ac.uk. https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/coefficient-of-determination-r-squared.html#:~:text=The%20coefficient%20of%20determination%2C%20or

Nvidia. (2024, July 29). *What is XGBoost?* NVIDIA Data Science Glossary. https://www.nvidia.com/en-us/glossary/xgboost/

Pandian, S. (2022, February 17). *K-Fold Cross Validation Technique and its Essentials*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/

Pykes, K. (2023, December). *8 of The Most Popular Machine Learning Tools*. datacamp. https://www.datacamp.com/blog/most-popular-machine-learning-tools

Saini, A. (2021, August 29). *Decision Tree Algorithm - A Complete Guide*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

Saxena, S. (2023, March 12). *Random Forest Hyperparameter Tuning in Python | Machine learning*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/

Sethi, A. (2020, March 27). *Support Vector Regression In Machine Learning*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/

Shah, R. (2021, June 23). *GridSearchCV |Tune Hyperparameters with GridSearchCV*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/

Statista Research Department. (2023). *Vietnam: TikTok penetration rate by generation 2023*. Statista. https://www.statista.com/statistics/1395248/vietnam-tiktok-penetration-rate-by-generation/#:~:text=Based%20on%20a%20survey%20conducted

Strapagiel, L. (2024, June 6). *How to Find the Best Time to Post on TikTok in 2024 - Shopify Canada*. Shopify. https://www.shopify.com/ca/blog/best-time-to-post-on-tikok

Stupak, T. (2024, January 12). *How Much Data Is Required To Train ML Models in 2024?* Akkio. https://www.akkio.com/post/how-much-data-is-required-to-train-ml#:~:text=The%2010%20times%20rule%20is

Suresh, A. (2020, November 20). *What is a confusion matrix?* Medium. https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

TensorFlow. (2019). *Keras | TensorFlow Core | TensorFlow*. TensorFlow. https://www.tensorflow.org/guide/keras

TensorFlow. (2024, June 15). *word2vec*. TensorFlow. https://www.tensorflow.org/text/tutorials/word2vec

Yifat, R. (2023, November 21). *Data Preparation for Machine Learning: The Ultimate Guide to Doing It Right*. Pecan AI. https://www.pecan.ai/blog/data-preparation-for-machine-learning/#:~:text=Data%20preparation%20for%20machine%20learning%20is%20the%20process%20of%20cleaning