Dataset Engagement Prediction

Participants

- **Faseeh U Rehman Qureshi (110188603)**
- **Yuhan Xiang (110182721)**

Both participants have equally and collaboratively contributed to all aspects of the project including data loading, preprocessing, modeling, analysis, tuning, and documentation.

The source code and experimental notebook are available on GitHub:
https://github.com/faseeh-quraishi/Meta_Kaggle_Hackathon.git

1. Introduction

Kaggle, as the largest online data science and machine learning community, provides a vast repository of datasets shared by both organizations and individual contributors. With thousands of datasets available, user engagement — measured in views, votes, and comments — varies significantly from one dataset to another. Understanding the factors that drive higher engagement is crucial for both dataset creators, who wish to maximize the impact of their contributions, and for the platform itself, which aims to foster an active and vibrant community.

The central research question of this project is: **Which datasets lead to higher engagement (in terms of views, votes, and comments) on Kaggle?**

To address this question, we conduct a comprehensive data analysis and modeling study using the Meta-Kaggle dataset, which includes detailed metadata on all public Kaggle datasets. We systematically analyze dataset characteristics — such as size, type, tags, update frequency, and creator information — to identify which features are most strongly associated with user engagement.

Our objectives are as follows:

1) To explore the distribution and correlation of engagement metrics (views, votes, comments) across all Kaggle datasets.
2) To identify and quantify which dataset attributes are predictive of higher engagement.
3) To build and compare machine learning models capable of predicting engagement levels based on dataset metadata.
4) To discuss the implications of our findings for dataset creators and the Kaggle community.

In the following sections, we present our data preparation workflow, exploratory data analysis, preprocessing techniques, and model training procedures. Finally, we discuss our results in the context of existing literature and offer suggestions for future research and practical applications.

2. Data Preparation

In order to investigate which Kaggle datasets achieve higher user engagement, we utilized the Meta-Kaggle repository, focusing on three core tables: Datasets, Forums, and ForumTopics. The Datasets table contains detailed information for over 500,000 datasets, including features such as creation date, activity date, number of views, downloads, votes, and metadata related to awards and forums.

To enhance our analysis, we performed several key preparation steps:

1) Data Loading:

   We loaded the three main CSV files into pandas DataFrames:

   a) Datasets.csv (504,735 rows, 16 columns): core dataset metadata and engagement metrics.
   b) Forums.csv (563,636 rows, 3 columns): mapping between datasets and their associated forums.
   c) ForumTopics.csv (460,716 rows, 13 columns): discussion threads and activity within each forum.

2) Forum Engagement Feature Engineering:

   Since direct comment counts per dataset are not provided in the Meta-Kaggle dataset, we used the number of forum topics associated with each dataset as a proxy for the number of comments or community discussions. Specifically, each dataset is linked to a forum, and the number of forum topics reflects the amount of user-initiated discussions and threads related to that dataset. While this measure may not capture every individual comment or reply, it provides a consistent and reasonable indicator of the breadth of

community engagement and how actively a dataset is discussed.

To generate this feature, we grouped the ForumTopics data by ForumId, counted the number of topics per forum, and merged this value into the main Datasets DataFrame as a new column TotalForumTopics. For datasets with no associated forum topics, missing values were filled with zero.

3) Data Integration:

After merging, the enhanced datasets_df includes TotalForumTopics, summarizing the extent of community discussions per dataset, in addition to existing engagement metrics such as views, votes, and downloads.

4) Preview of Prepared Data:

The resulting DataFrame provides a comprehensive foundation for engagement analysis, capturing both direct interactions (views, votes) and indirect community participation (forum discussions), which are both crucial for understanding dataset popularity on Kaggle.

3. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was conducted to better understand the structure of the dataset and to identify the features most relevant to predicting user engagement on Kaggle datasets. Our analysis focused on both the distribution of key variables and the relationships between predictors (features) and engagement metrics (targets).

To obtain an initial understanding of the dataset, we examined the data types, basic statistics, and missing values across all features. The dataset comprises over 500,000 entries and 17 columns, covering both numerical and categorical attributes relevant to Kaggle datasets and their engagement metrics.

3.1 Overview and Missing Value Analysis

A summary of the data structure is presented below:

1) Column Types:

The dataset includes a mix of integer, float, and object (string) columns. Key engagement-related columns such as TotalViews, TotalDownloads, TotalVotes, and TotalForumTopics are of integer type and are present for all entries. Categorical variables such as Type (dataset type) and Medal (medal status) are stored as objects or floats.

```
 #   Column                     Non-Null Count   Dtype
---  ------                     --------------   -----
 0   Id                         504735 non-null  int64
 1   CreatorUserId              504735 non-null  int64
 2   OwnerUserId                502168 non-null  float64
 3   OwnerOrganizationId        2567 non-null    float64
 4   CurrentDatasetVersionId    504502 non-null  float64
 5   CurrentDatasourceVersionId 504491 non-null  float64
 6   ForumId                    504735 non-null  int64
 7   Type                       504735 non-null  object
 8   CreationDate               504735 non-null  object
 9   LastActivityDate           504735 non-null  object
 10  TotalViews                 504735 non-null  int64
 11  TotalDownloads             504735 non-null  int64
 12  TotalVotes                 504735 non-null  int64
 13  TotalKernels               504735 non-null  int64
 14  Medal                      31518 non-null   float64
 15  MedalAwardDate             29237 non-null   object
 16  TotalForumTopics           504735 non-null  int64
dtypes: float64(5), int64(8), object(4)
memory usage: 65.5+ MB
```

2) Descriptive Statistics:

Descriptive statistics for both numerical and categorical variables were computed, including count, mean, standard deviation, and the number of unique values. For example, the Type column shows a single unique value ("Dataset"), confirming all entries refer to datasets, while Medal and MedalAwardDate have a significant proportion of missing values, indicating that only a small subset of datasets received medals.

3) Missing Value Analysis:

Most features, including all key engagement metrics and identifying fields, have complete data. However, OwnerUserId is missing in about 0.5% of cases, and OwnerOrganizationId is missing for over 99% of entries, suggesting that most datasets are not affiliated with an organization. The Medal and MedalAwardDate fields are missing for over 93% of datasets, reflecting the relatively rare award of medals.

```
In [6]:
# Check for missing values percentages
(datasets_df.isnull().sum() / len(datasets_df)) * 100
```

```
Out[6]:
Id                          0.000000
CreatorUserId               0.000000
OwnerUserId                 0.508584
OwnerOrganizationId        99.491416
CurrentDatasetVersionId     0.046163
CurrentDatasourceVersionId  0.048342
ForumId                     0.000000
Type                        0.000000
CreationDate                0.000000
LastActivityDate            0.000000
TotalViews                  0.000000
TotalDownloads              0.000000
TotalVotes                  0.000000
TotalKernels                0.000000
Medal                      93.755535
MedalAwardDate             94.207455
TotalForumTopics            0.000000
dtype: float64
```

4) Categorical Variable Consistency:

We reviewed categorical variables such as Type and Medal for consistency. Type is consistent throughout, and the categories of Medal (bronze, silver, gold) align with Kaggle's standard award system. No irregular or unexpected values were found in these columns.

5) Summary Table:

The standard output of .describe(include='all') was used to summarize numerical ranges and categorical value counts, further confirming the completeness and reliability of the dataset for subsequent analysis.

	Id	CreatorUserId	OwnerUserId	OwnerOrganizationId	CurrentDatasetVersionId	CurrentDatasourceVersionId
count	5.074970e+05	5.074970e+05	5.049290e+05	2568.000000	5.072600e+05	5.072490e+05
unique	NaN	NaN	NaN	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN	NaN	NaN
mean	3.893743e+06	1.100190e+07	1.104608e+07	1250.682632	6.496214e+06	6.653878e+06
std	2.259711e+06	7.258462e+06	7.243565e+06	1274.078892	3.612059e+06	3.744979e+06
min	6.000000e+00	1.000000e+00	3.680000e+02	2.000000	5.800000e+01	5.800000e+01
25%	1.899875e+06	4.786697e+06	4.844641e+06	265.000000	3.159078e+06	3.208569e+06
50%	3.851762e+06	1.015907e+07	1.022414e+07	959.000000	6.749370e+06	6.834416e+06
75%	5.816288e+06	1.650777e+07	1.653491e+07	1606.000000	9.596287e+06	9.817818e+06
max	7.861081e+06	2.783500e+07	2.783500e+07	5161.000000	1.246153e+07	1.303522e+07

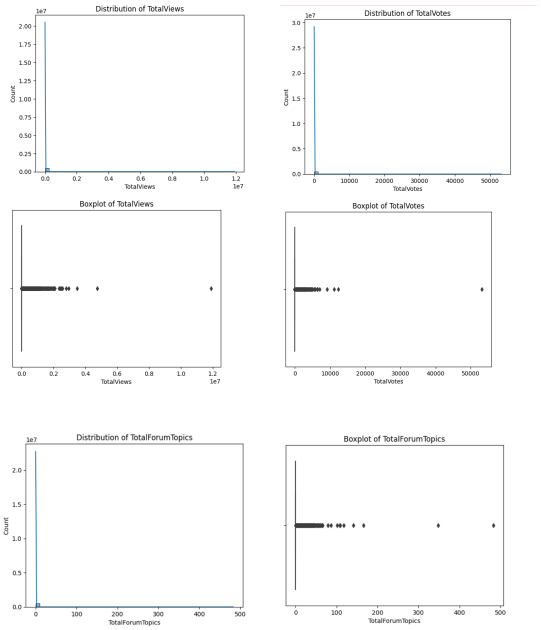
Row Name	Explanation
count	Number of non-missing (non-NaN, non-None) values in the column.
unique	Number of unique values in the column (only for categorical/string columns).
top	The most frequent value (mode) in the column (for categorical/string columns).
freq	Frequency of the most frequent value (how many times "top" appears).
mean	Average value (only for numeric columns).
std	Standard deviation, measures spread of the values (only for numeric columns).
min	Minimum value (only for numeric columns).
25%	25th percentile (the value below which 25% of the data fall, numeric only).
50%	50th percentile (the median, numeric only).
75%	75th percentile (the value below which 75% of the data fall, numeric only).
max	Maximum value (only for numeric columns).

These initial checks ensure that our subsequent analyses and modeling are based on a robust and well-understood dataset, with key variables available and categorical values validated for consistency and completeness.

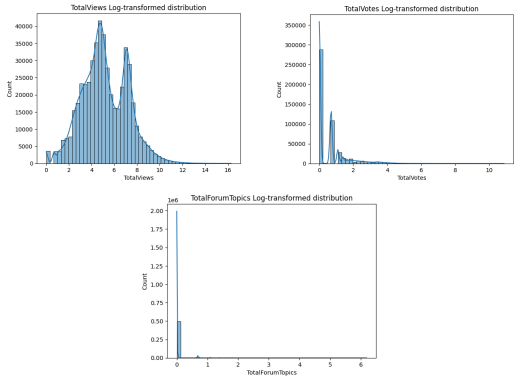
3.2 Distribution Analysis of Target (Y) and Feature Variables (X)

3.2.1 Distribution Analysis of Target Variables (Y)

We first examined the distribution characteristics of the three main engagement metrics—TotalViews, TotalVotes, and TotalForumTopics—across all Kaggle datasets. As shown in the histograms and boxplots, all three target variables exhibit a strong right-skewed distribution, with the majority of datasets receiving very low engagement and a small minority achieving extremely high values. Outliers are clearly visible in each boxplot, confirming the presence of highly popular datasets.



To address this heavy skewness and prepare the data for subsequent modeling, we applied a logarithmic transformation (log1p) to each target variable. After transformation, the histograms become more symmetric and the influence of extreme values is reduced, which is expected to facilitate more effective and stable model training.



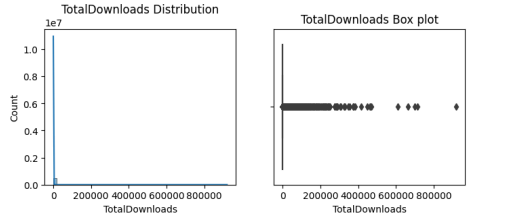
3.2.2 Distribution Analysis of Feature Variables (X)

We also analyzed the distributions of key predictor variables, both numerical and categorical:

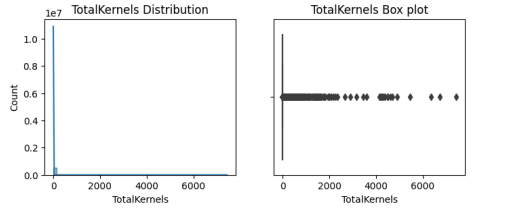
1) Numerical Features:

The engagement-related features, such as TotalDownloads and TotalKernels, exhibit strong right-skewness with a concentration of values near zero and a small number of datasets achieving very high counts. This is reflected in both the histograms, which show a long tail, and the boxplots, where many outliers are present. These features resemble the pattern observed in the main target variables.

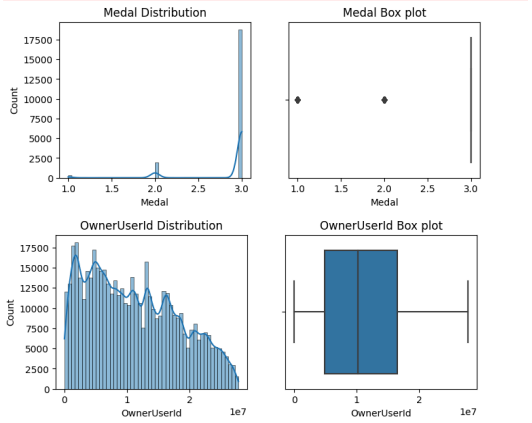
In contrast, identifier and versioning features such as OwnerUserId, OwnerOrganizationId, CurrentDatasetVersionId, and CurrentDatasourceVersionId display more uniform or multimodal distributions. Their histograms lack the pronounced skewness seen in engagement metrics, instead showing a relatively even spread or distinct peaks. This indicates that user and dataset identifiers are widely distributed without strong bias toward particular values.



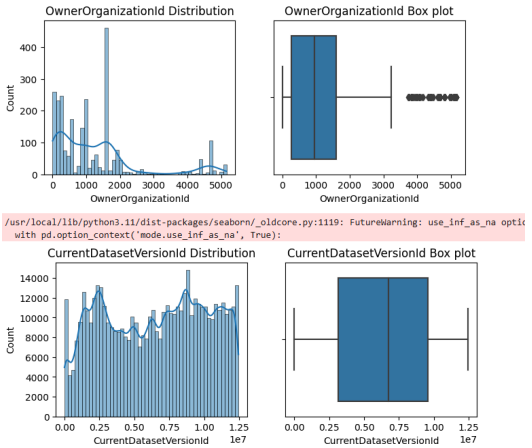

```
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na optic
  with pd.option_context('mode.use_inf_as_na', True):
```

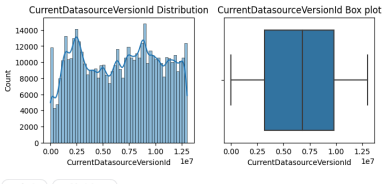



```
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na optic
  with pd.option_context('mode.use_inf_as_na', True):
```




```
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na optic
  with pd.option_context('mode.use_inf_as_na', True):
```

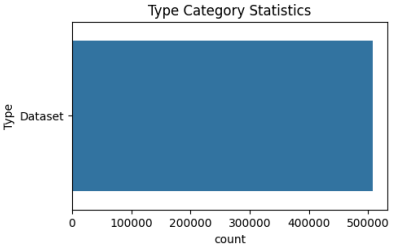




2) Categorical Features:

The Type column is categorical and remains constant ("Dataset") for all entries, as confirmed by the bar chart and value counts.

```
Type Number of each category:
 Type
Dataset    507497
Name: count, dtype: int64
```



3) Missing Value and Unique Value Analysis:

We calculated the missing value counts and the number of unique values for each selected feature. While most features are nearly complete, fields like Medal, OwnerOrganizationId exhibit a large proportion of missing data, consistent with the fact that most datasets are uploaded by individual users rather than organizations.

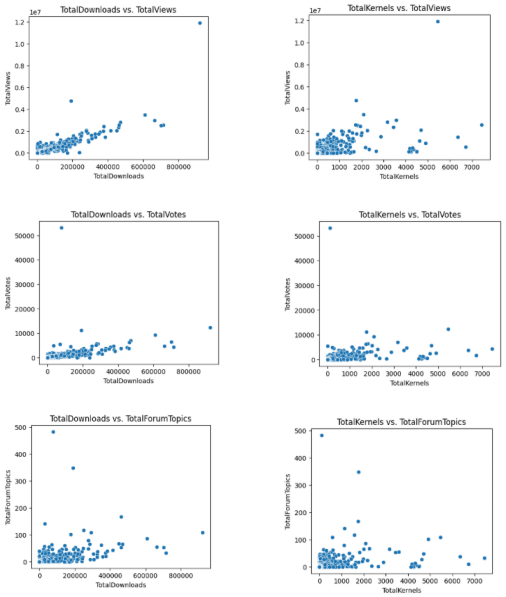
```
Missing values for features:
 TotalDownloads                  0
TotalKernels                     0
Medal                       486380
OwnerUserId                   2568
OwnerOrganizationId         504929
CurrentDatasetVersionId        237
CurrentDatasourceVersionId     248
Type                             0
dtype: int64

TotalDownloads Number of unique values: 7588
TotalKernels Number of unique values: 487
Medal Number of unique values: 3
OwnerUserId Number of unique values: 187182
OwnerOrganizationId Number of unique values: 387
CurrentDatasetVersionId Number of unique values: 507260
CurrentDatasourceVersionId Number of unique values: 507249
Type Number of unique values: 1
```

3.3 Feature Correlation and Extreme Value Analysis

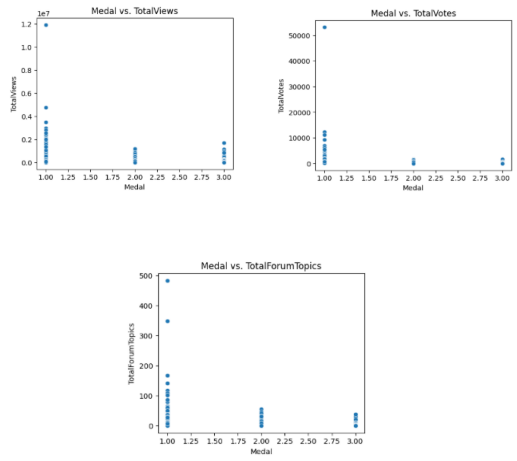
3.3.1 Relationship between Features and Targets (X vs Y)

To assess the relationships between key features and engagement metrics, we visualized scatter plots and boxplots for all major predictors against each target variable (TotalViews, TotalVotes, TotalForumTopics).

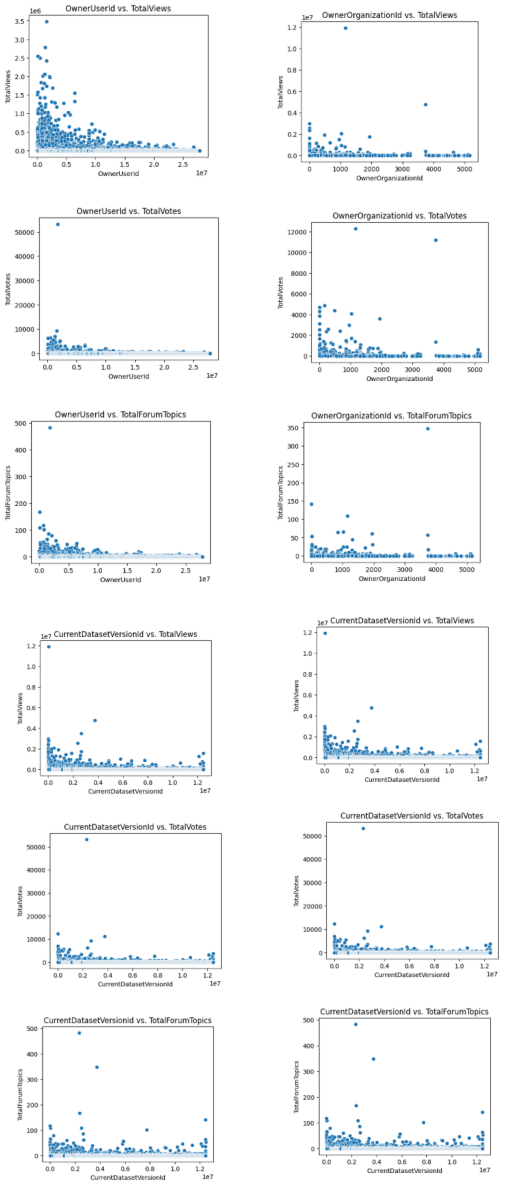
1) Engagement-related features such as TotalDownloads and TotalKernels show a positive but nonlinear relationship with all target variables. While higher downloads and kernel counts are generally associated with increased views, votes, and forum topics, the relationship is not strictly proportional, and significant variance exists among high-engagement datasets. Most datasets are concentrated in the lower ranges, with a handful of outliers driving the upper extremes (as seen in the scatter plots).



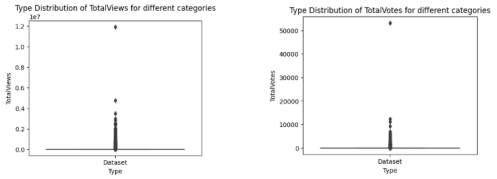
2) For the award level feature (Medal), which indicates the highest recognition (1=bronze, 2=silver, 3=gold) that a dataset has received, datasets with higher medal levels tend to have somewhat higher engagement on average. However, there is considerable overlap among medal categories, and some highly engaged datasets may not necessarily have the highest medal.

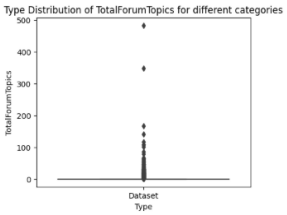


3) Identifier-related features (such as OwnerUserId, OwnerOrganizationId, CurrentDatasetVersionId, and CurrentDatasourceVersionId) do not exhibit strong or interpretable patterns with respect to engagement metrics. The scatter plots for these features are largely dispersed, reflecting that these variables act more as identifiers rather than meaningful predictors of popularity.



4) The categorical feature Type is constant and provides no differentiation across targets.

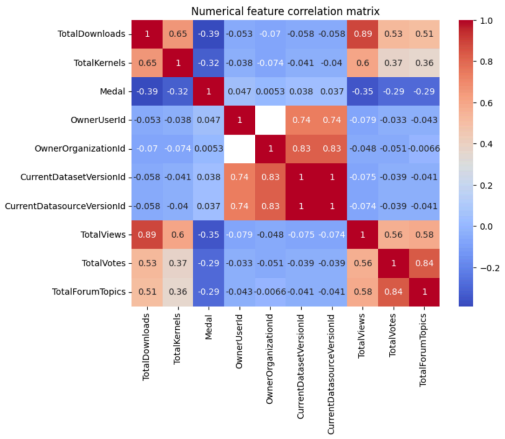




These results suggest that only certain numerical features (especially those related to downloads and kernels) show meaningful associations with engagement, while identifier fields do not contribute substantially to explaining engagement variation.

3.3.2 Correlation between Features (x-x, x-y)

We further explored the relationships among all main numerical features and targets using a correlation matrix heatmap.



1) There are strong positive correlations between TotalDownloads and both TotalViews (0.89) and TotalVotes (0.53), reflecting the intuitive link that popular datasets tend to be both downloaded and viewed more frequently.
2) TotalKernels is also positively correlated with engagement metrics, though to a lesser degree.
3) Target variables themselves (views, votes, forum topics) are moderately to strongly correlated, with the highest correlation observed between votes and forum topics (0.84).
4) Identifier and version features are highly correlated with each other (as expected for numeric IDs), but show little or no correlation with engagement metrics, confirming their limited predictive value for dataset popularity.

These findings reinforce that a few engagement-related features account for much of the variation in target metrics, while many other features are less informative for predicting user engagement.

3.3.3 Outlier and Extreme Value Detection

Finally, we summarized the maximum and minimum values for all main features and targets to assess the presence of outliers and extreme values.

1) Across all engagement metrics and related features, there are a small number of datasets with exceptionally high values (e.g., over 1 million views, 900,000+ downloads, and 53,000+ votes), while the majority of datasets have values near zero.
2) This long-tailed distribution confirms the existence of substantial outliers, which can heavily influence model training and performance metrics.
3) For identifier and versioning fields, the maximum values are consistent with the overall scale of the dataset and do not suggest the presence of abnormal entries.

```
TotalDownloads Maximum/Minimum: 920905/0
TotalKernels Maximum/Minimum: 7436/0
Medal Maximum/Minimum: 3.0/1.0
OwnerUserId Maximum/Minimum: 27835001.0/368.0
OwnerOrganizationId Maximum/Minimum: 5161.0/2.0
CurrentDatasetVersionId Maximum/Minimum: 12461529.0/58.0
CurrentDatasourceVersionId Maximum/Minimum: 13035218.0/58.0
TotalViews Maximum/Minimum: 11907658/0
TotalVotes Maximum/Minimum: 53189/0
TotalForumTopics Maximum/Minimum: 483/0
```

During modeling, these outliers should be carefully handled—either by using robust metrics (such as F1-score or log-transformed targets) or by considering advanced preprocessing techniques to mitigate their influence

4. Data Preprocessing

Prior to model development, a series of preprocessing steps were performed to ensure data quality and to prepare the dataset for machine learning analysis.

First, we removed columns with more than 50% missing values (including OwnerOrganizationId, Medal, and MedalAwardDate), as well as columns with only one unique value (such as Type), since these features offer little or no predictive value and may negatively impact model performance. Irrelevant columns such as the unique dataset identifier (Id) were also dropped to avoid potential data leakage.

To enhance the temporal dimension of our data, the CreationDate and LastActivityDate columns were converted to datetime format and used to calculate a new feature, ActivityDuration, representing the active lifespan of each dataset on the platform. Any negative or invalid durations were replaced with the median of valid durations, and the original date columns were subsequently removed.

Missing values in the remaining numerical features were imputed with the median, a robust approach that helps minimize the influence of outliers while preserving the central tendency of the data. To

further control for extreme values, we applied the interquartile range (IQR) rule to all numerical columns, removing any records that fell outside 1.5 times the IQR above the third quartile or below the first quartile. This step reduced the risk of bias and instability in the downstream models.

After these cleaning and feature engineering steps, we split the dataset into training, validation, and test sets using a 70-10-20 ratio. This partitioning ensures a fair evaluation of model performance and enables effective hyperparameter tuning. The final sample sizes for each subset were: 260,424 (train), 37,204 (validation), and 74,408 (test), as shown in the output above.

These preprocessing actions resulted in a clean, well-structured dataset with relevant features and minimal missing data or outliers, laying a strong foundation for subsequent machine learning experiments.

5. Model Training and Evaluation

To predict dataset engagement on Kaggle (measured by views, votes, and forum topics), we adopted a multi-output regression approach using a diverse set of machine learning models. This section explains the rationale for selecting specific models, their setup and tuning, and a comparative evaluation based on predictive performance.

5.1 Rationale for Model Selection

We selected a mix of traditional and ensemble machine learning models to explore their capabilities in handling non-linear patterns, high dimensionality, and skewed data distributions. Our focus was on interpretable, scalable, and generalizable models suitable for tabular datasets with both numerical and categorical features.

The following models were selected:

1. Ridge Regression – A regularized linear model that controls overfitting through L2 penalty, useful for correlated features.

2. Random Forest Regressor – A robust ensemble of decision trees known for handling non-linear relationships and feature interactions.

3. XGBoost Regressor – A gradient boosting technique that offers improved accuracy, speed, and regularization over traditional boosting.

4. MultiOutputRegressor Wrapper – To handle the multi-target regression task (predicting views, votes, and forum topics simultaneously).

While tree-based models such as Random Forest are known for strong performance on tabular data, they incurred significantly higher training time during our experiments. Given the dataset size (~370,000 records after preprocessing), Random Forests required substantial computational resources, particularly when using deeper trees or a large number of estimators. Despite their predictive power, the high training cost made them less practical for iterative experimentation or tuning compared to XGBoost.

5.2 Model Setup and Initialization

Each model was initialized with base parameters, and the target variables (TotalViews, TotalVotes, TotalForumTopics) were log-transformed (log1p) to reduce skewness and improve model stability.

We used MultiOutputRegressor from sklearn.multioutput to extend single-target regressors to multi-output tasks.

5.3 Adaptive Hyperparameter Tuning

For each machine learning model, we implemented adaptive hyperparameter tuning strategies to efficiently identify optimal configurations with minimal computational overhead. Rather than exhaustive grid or random search, our adaptive loops iteratively refined the hyperparameter search space based on validation set feedback, leading to faster convergence and improved model performance.

Ridge Regression

For Ridge Regression, the primary hyperparameter is the L2 regularization strength (alpha). We employed an adaptive tuning loop that started with a broad range of candidate alpha values (e.g., [0.01, 0.1, 1.0, 10.0, 100.0]). In each round, we trained a model for each alpha, evaluated mean squared error (MSE) on the validation set, and selected the best-performing value. The search range was then narrowed around the current best alpha for the next round. This process continued until the improvement in validation MSE fell below a small threshold (< 0.001), ensuring efficient convergence without unnecessary computation. The final model was retrained on the combined training and validation data using the optimal alpha.

XGBoost Regressor

For XGBoost, we focused on tuning the number of boosting rounds (n_estimators), a key hyperparameter influencing both predictive power and training time. We initialized the adaptive search with a set of candidate values (e.g., [10, 25, 40, 55, 75, 100]) and followed a similar iterative process as for Ridge Regression. After each round, the search interval was refined around the best n_estimators found, and training was repeated until improvements plateaued. The final XGBoost model

was then fitted to the combined training and validation data using the best n_estimators.

Neural Network (Keras)

For the neural network model, key architectural and training hyperparameters included the number of layers, units per layer, activation functions, batch size, and the number of epochs. To mitigate overfitting, we employed early stopping based on validation loss, halting training if the model did not improve after a set number of epochs (patience). The final network architecture consisted of two hidden layers with 128 and 64 neurons, respectively, both using ReLU activation, and an output layer with three units (one for each target). Training proceeded for up to 10 epochs, but early stopping ensured optimal performance was captured without unnecessary training.

General Approach

Across all models, validation set performance (primarily MSE) guided hyperparameter selection. By using adaptive tuning loops rather than traditional grid or random search, we substantially reduced computational costs while still achieving strong predictive performance. This approach also allowed for more dynamic adjustment of hyperparameter ranges in response to interim results, leading to efficient and robust model selection.

5.4 Model Evaluation and Comparison

After training and tuning, all models were evaluated using the Mean Squared Error (MSE) and R^2 Score on both validation and test sets.

```
Ridge Regression Evaluation:
Output 1: MSE = 0.6827, R² = 0.7392
Output 2: MSE = 0.0816, R² = 0.3239
Output 3: MSE = 0.0000, R² = 1.0000
-----------------------------------
Avg MSE: 0.2548, Avg R²: 0.6877
-----------------------------------

XGBoost Evaluation:
Output 1: MSE = 0.4218, R² = 0.8388
Output 2: MSE = 0.0686, R² = 0.4323
Output 3: MSE = 0.0000, R² = 1.0000
-----------------------------------
Avg MSE: 0.1635, Avg R²: 0.7571
-----------------------------------

Neural Network Evaluation:
Output 1: MSE = 0.4619, R² = 0.8235
Output 2: MSE = 0.0735, R² = 0.3910
Output 3: MSE = 0.0000, R² = 0.0000
-----------------------------------
Avg MSE: 0.1785, Avg R²: 0.4048
-----------------------------------
```

Model	**Ridge**	**XGBoost**	**Neural Network**
Avg. MSE	0.2548	0.1635	0.1788
Avg. R^2	0.6877	0.7571	0.4053

After evaluating three regression models: Ridge Regression, a Neural Network, and XGBoost. Based on performance on the test set, XGBoost achieved the lowest average Mean Squared Error and the highest average R^2 score. It consistently outperformed the other models on all targets, making it the most suitable model for our task.

6. Discussion

This study set out to answer the research question: Which datasets lead to higher engagement (in terms of views, votes, and forum discussions) on Kaggle? Through comprehensive data analysis and predictive modeling using the Meta-Kaggle dataset, we identified several factors that significantly influence dataset popularity and user engagement.

6.1 Summary of Findings

Our analysis revealed that engagement-related features, particularly TotalDownloads, TotalKernels, and ActivityDuration, are the most predictive of higher engagement levels. These variables consistently demonstrated strong positive correlations with the target metrics (TotalViews, TotalVotes, and TotalForumTopics). Furthermore, datasets that received Kaggle medals (bronze, silver, gold) tended to have relatively higher engagement, although overlap between medal categories suggests that other factors also play substantial roles. The final models—especially XGBoost—achieved strong predictive performance, confirming that metadata alone can be used to reasonably estimate how well a dataset will be received by the community.

Our analysis revealed that engagement-related features, particularly TotalDownloads, TotalKernels, and ActivityDuration, are the most predictive of higher engagement levels. These variables consistently demonstrated strong positive correlations with the target metrics (TotalViews, TotalVotes, and TotalForumTopics). Furthermore, datasets that received Kaggle medals (bronze, silver, gold) tended to have relatively higher engagement, although overlap between medal categories suggests that other factors also play substantial roles. The final models—especially XGBoost—achieved strong predictive performance, confirming that metadata alone can be used to reasonably estimate how well a dataset will be received by the community.

6.2 Advantages of Our Approach

- Multi-target Modeling: By framing the problem as a multi-output regression task, we simultaneously predicted multiple aspects of engagement, capturing the multidimensional nature of dataset popularity.
- Data-Enriched Feature Engineering: The creation of the TotalForumTopics feature from forum metadata provided a novel and useful

proxy for community discussion, compensating for the lack of explicit comment counts.

- Robust Preprocessing: Log-transformations, IQR-based outlier removal, and careful imputation strategies ensured that model training was stable and not skewed by extreme values.

- Scalable Model Selection: Ensemble methods such as XGBoost proved to be highly effective on large tabular data, achieving strong results while balancing accuracy and computational feasibility.

6.3 Limitations

Despite these strengths, several limitations should be acknowledged:

- Proxy Variable for Comments: The number of forum topics was used as a stand-in for actual comment counts. This measure does not capture the depth or sentiment of discussions and may miss important nuances in community engagement.

- Lack of Textual or Semantic Analysis: Our study relied exclusively on structured metadata. Titles, descriptions, and tags of datasets—which likely influence user interest—were not included due to data constraints or preprocessing limitations.

- Temporal and External Factors: We did not model time-dependent factors (e.g., trending topics, seasonal interest) or external promotion (e.g., blog posts, social media), which can significantly influence engagement.

- Imbalanced Distribution: The highly skewed nature of the engagement metrics means that models may underperform on low-engagement datasets or overemphasize popular datasets unless further calibrated.

6.4 Future Expansions

This project opens several avenues for future research and improvement:

- Incorporating NLP Features: Integrating textual data such as dataset titles, descriptions, and comments using natural language processing could significantly enhance predictive power.

- Temporal Modeling: Time series or survival analysis could help estimate the lifespan of engagement and model how interest evolves after dataset publication.

- User Behavior Modeling: By tracking uploader profiles and kernel author activity, we could study how individual reputation and contribution frequency affect dataset success.

7. Conclusion

This project aimed to answer the central research question: Which datasets lead to higher engagement (in terms of views, votes, and forum discussions) on Kaggle? By analyzing and modeling over 500,000 datasets from the Meta-Kaggle repository, we uncovered valuable insights into the characteristics that drive user interaction on the platform.

The Meta-Kaggle dataset proved to be highly effective in supporting this analysis. Its rich metadata—covering downloads, kernels, activity timelines, and forum associations—enabled us to capture both direct engagement (views, votes) and community involvement (discussions). The creation of a derived feature, TotalForumTopics, further allowed us to approximate user commentary and discussion levels, addressing a key gap in the original data.

Our findings indicate that dataset popularity on Kaggle is strongly influenced by prior user interactions, such as downloads and kernels, as well as activity duration. These insights directly answer our research question and offer practical guidance to dataset creators who seek to optimize visibility and impact. For Kaggle itself, this research can inform ranking algorithms, recommendation systems, and award criteria, potentially improving how content is surfaced and how creators are recognized.

Ultimately, this work demonstrates the value of using the Meta-Kaggle dataset not only as a research tool, but also as a mechanism for improving platform engagement and user experience. With further expansion into textual and temporal dimensions, this line of research can support smarter, data-driven decision-making for both users and the Kaggle platform.

8. References

1. Kaggle Meta Dataset.
   Meta-Kaggle Dataset. Available at: https://www.kaggle.com/datasets/kaggle/meta-kaggle
   Accessed: July 2025.

2. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011).
   Scikit-learn: Machine learning in Python.
   Journal of Machine Learning Research, 12, 2825–2830.

   https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

3. Chen, T., & Guestrin, C. (2016).
   XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD

International Conference on Knowledge Discovery and Data Mining (pp. 785–794). DOI: 10.1145/2939672.2939785

4. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Second Edition. Springer. ISBN: 978-0387848570

5. Van Rossum, G., & Drake Jr, F. L. (2009). *Python 3 Reference Manual.* Scotts Valley, CA: CreateSpace.

6. McKinney, W. (2010). *Data Structures for Statistical Computing in Python.* In Proceedings of the 9th Python in Science Conference, 51–56.

7. Hunter, J. D. (2007). *Matplotlib: A 2D graphics environment.* Computing in Science & Engineering, 9(3), 90–95.

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

9. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). *TensorFlow: A system for large-scale machine learning.* In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 265–283.