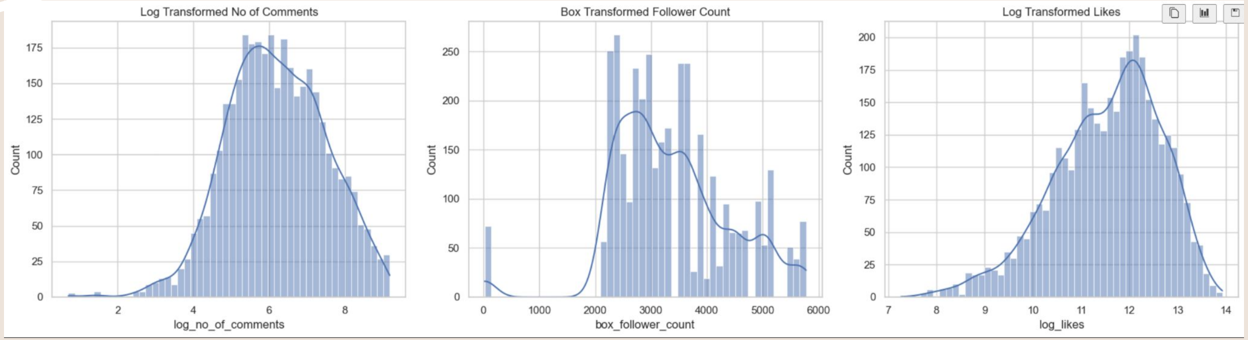
Instagram Likes: Analysis and Prediction

By: Ayan Bhowmick

Exploratory Data Analysis

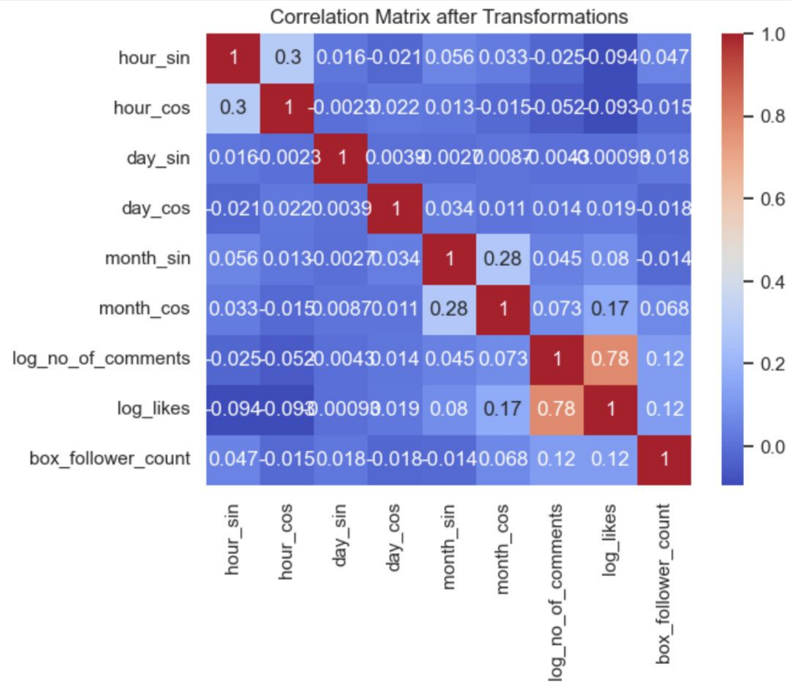
- **Skewness in Variables**
  - A significant skew was observed in several variables.
  - **Action Taken**: Applied log transformations to `likes` and `no_of_comments`; used Box-Cox for `follower_counts` to normalize distributions. Then applied a standard scaler (as I would later be testing SVMs).
- **Outlier Removal**
  - Removed outliers in `no_of_comments` to improve data quality.
- **Correlation Coefficient Matrices**
  - Histograms and correlation matrices are included to illustrate the results.
- **Transformation Results**
  - Correlation coefficient between `likes` and `no_of_comments` increased from **0.69** to **0.78** after transformations.
  - Skewness dropped to < |1|.

https://github.com/highheat4/Instagram_Data_Analysis/blob/main/EDA.ipynb



Feature Selection

- **Time Extraction**: From a single large, unscaled timestamp, I extracted:
  - **Month**
  - **Day**
  - **Hour**
- **Cyclical Transformations**: Recognizing the cyclical nature of time:
  - Applied sine and cosine transformations to the hour, allowing for a more accurate representation of proximity (e.g., hour 23 is closer to hour 0 than to hour 12).
- **Weekend Indicator**: Tested significance of a weekend feature, but the correlations with likes were generally low.
- **Correlation Insights**: The highest correlation observed was with the **month** (0.12). Although this is relatively small, I decided to retain it in the dataset to maintain at least one temporal factor. I removed all other time dependencies



https://github.com/highheat4/Instagram_Data_Analysis/blob/main/metadata_prediction.ipynb

Metadata Analysis & Insights

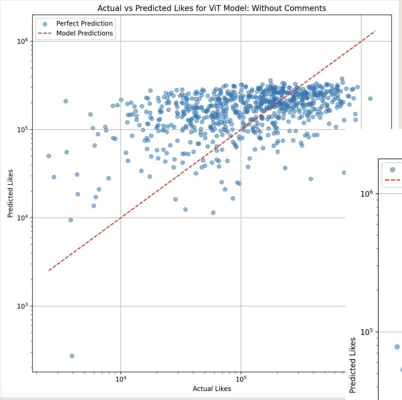
- **Ablation Study on Comments**
  - To predict likes for upcoming posts, the comment count is not available in advance. Therefore, an ablation study was conducted by removing the comment feature.
- **Models Tested**
  - Five models were evaluated for metadata analysis:
    - Linear Regression
    - Random Forest
    - Gradient Boosting
    - XGBoost
    - Support Vector Machines
- **Results Without Image Data**
  - Analysis on metadata alone yielded an R^2 score of **0.888**, indicating strong predictive capability, but not perfect.
- **Impact of Comments on Predictions**
  - Inclusion of comments significantly influences predictions. Without this feature, the highest R^2 score dropped to **0.749**.

https://github.com/highheat4/Instagram_Data_Analysis/blob/main/metadata_prediction.ipynb

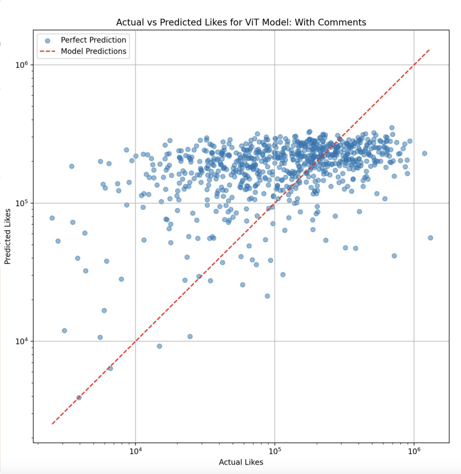
Holistic Approach

- To leverage image data effectively, two main approaches were employed:
  - **Image-Based Models**:
    - Vision Transformers were first fine tuned on a subset of the data, predicting on ‘likes’ directly.
    - Image embeddings were then extracted using the fine tuned ViTs and used as features for other regression models
  - **CLIP Image Classification**:
    - CLIP was used to classify images, and the resulting labels were converted into GloVe word embeddings, which were then added to the feature set.
- Due to the high dimensionality of these embeddings, dimensionality reduction techniques were applied:
  - **PCA** was used for 5 different levels of reduction: 1, 2, 5, 20, 50, 100, and 200 dimensions.
  - This would aid other models in performing regression tasks.
- To avoid potential high collinearity between embeddings extracted by the two models, I opted not to combine them into a single feature set.

Trained ViT Direct Prediction



R squared = 0.12



R squared = 0.0932

https://github.com/highheat4/Instagram_Data_Analysis/blob/main/core/vit_train.py

- Training on Transformed Data:
  - Ran ViT training on transformed data.
  - Observed that performance worsened with more training.
  - R squared reached -0.85 at some point.
- Ensuring Transformation Issues:
  - To verify that the issue wasn't with the transformations (as ViT requires another transformation for non-image data), ran multiple epochs on untransformed data.
  - Highest R squared achieved: 0.12 without comments, 0.0932 with comments

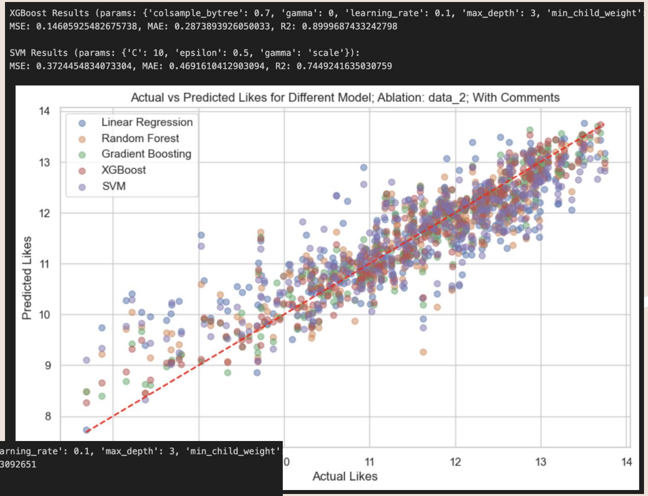
Trained ViT Embeddings + Metadata

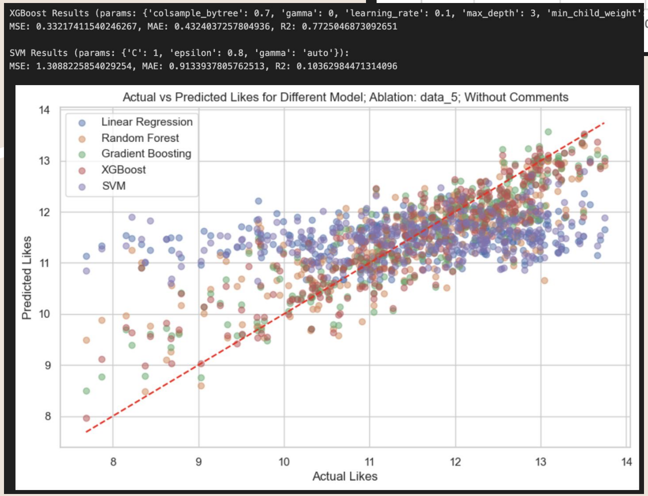
- Model Embedding Extraction:
  - Extracted embeddings from fine-tuned model
- Dimensionality Reduction:
  - Performed Principal Component Analysis (PCA) on embeddings
- Model Training:
  - Utilized same models as in Metadata Analysis
- Results:
  - [Insert key findings here]

CLIP -> GloVe + Metadata

- Image Classification:
  - Used AI-generated list of Instagram-relevant image categories
  - Classified images into these categories using CLIP
- Embedding Processing:
  - Converted categories to GloVe word embeddings
  - Applied PCA for dimensionality reduction
- Model Training:
  - Utilized same models as in Metadata Analysis
- Feature Integration:
  - Combined original metadata with PCA-reduced embeddings
- Result: Slightly better performance on both ablations

https://github.com/highheat4/Instagram_Data_Analysis/blob/main/core/clipglove_embedding_extract.py





https://github.com/highheat4/Instagram_Data_Analysis/blob/main/core/word_and_md_pred.ipynb

Results and Summary

- Key Findings:
  - Image data marginally improves like count prediction
  - Transformer models alone struggle with accurate predictions
- Implications:
  - Combined approach may yield best performance
  - Observed improvements could be due to random chance
- Limitations:
  - Performance boosts are small
  - Multiple ablations may influence results
- Best Results: CLIP -> GloVe + Metadata
  - With comments, XGB w dimensions reduced to 2; $R^2 = .900$
  - W/o comments, XGB w dimensions reduced to 5; $R^2 = .772$

Revisions

Looking back, I believe I could have done outlier detection better than I did to remove outliers in more dimensions than just one