Readme

The attached `virality_predictor` folder contains following files

- `readme.pdf`
- `requirements.txt` - Libraries used in making predictor
- `article_text_features.ipynb` - Features generated from article content (text) & stores it in `data/article_info.csv`
- `virality_predictor.ipynb` - Model and their evaluation
- `Data` - Contains data
  - `shared_articles.csv`
  - `users_interactions.csv`
  - `article_info.csv`

Questions and Answers

What features did you consider?

I have considered following features for this problem:

1. Features extracted from users interaction to articles.

   - Number of View
   - Number of Bookmark
   - Number of Comment created
   - Number of Follow
   - Number of Like

2. Features extracted from articles

   - Language of article (one hot encoded)
   - Content type (one hot encoded)
   - Event type (one hot encoded)

3. Features extracted from article text

   - Numbers of tokens
   - Numbers of unique tokens
   - Average token length
   - N non-stop unique tokens
   - Global subjectivity
   - Avg positive polarity
   - Global sentiment polarity

What model did you use and why?

For this problem, I have used simple and perfect fit **regressions** model.

Starting from simple Linear regression model to more sophisticated and optimized regression models and tested each model with test set. All the model are fine tuned and each model evaluated based on RMSE, MAE, and r-squared scores.

I have used lasso and ridge as both are the regularized model and built on top of linear regression with small changes in penalty.

The XGB and CatBoost are optimized and efficient machine learning libraries. So, It is good practice to fit and test your data on such libraries regressors. Both came up with improvement on ridge and lasso and also uses the functionality of both ridge and lasso.

Following models are chosen:

- Linear regression
- Lasso regression
- Ridge regression
- XGB regression
- CatBoost regression

What was your evaluation metric for this?

As this is a regression problem and the dependent variable is continuous, So, RMSE, MAE, and r-squared metric, is best for such problem to evaluate.
In this project I have used following metric to evaluate the regression model:

- RMSE
- MAE
- R-Squared

What features would you like to add to the model in the future if you had more time?

For future work, I would like to play more with article content, add more features as possible from article content and see how it performs with respect to users activity (Like, comment, bookmark etc.) because for any article to become popular or viral is totally based on the content in it.

So, for future reference following features I would like add below features to model:

- Article title, content and its trends
  - Number of images
  - Platform - website where it published
  - Day - Weekdays or weekend
  - Title subjectivity
  - Title sentiment polarity

- Inlinks and outlinks of article url

- Normalization and Scaling to apply wherever possible.

What other things would you want to try before deploying this model in production.

Before deploying to production level I would like to do following things with model and its output:

- Make the model into classification model, Don't worry it stills predicts the popularity (virality) of articles, but in a category or divide the virality score into categories. For example we divide the vitality score (calculated from vitality equation) into 5 sub category with 5 (most viral) to 1 (less viral) So, we get to know to how popular or viral particular article at very upper level.

- The virality score can be calculated by ensembling of 3-4 different regression model if virality score is continuous.

- A/B testing on different groups and types of users