What features did you consider?

I chose to keep the problem statement strictly a NLP problem. That's why the features I chose were title, text and virality. Virality was calculated based on given formula. To avoid computational power and time the model present is only trained on title.

What model do you use and why?

From the nature of virality, it's pretty clear this is a regression problem. I chose linear regression and lasso regression. It's pretty clear from negative value of r^2 that models fits the data poorly. With more time, I would like to use other linear models from the scikit-learn library.

What was your evaluation metric for this?

I calculate r^2 and explained variance score. This helps us to calculate mean error. In an ideal world both r^2 and explained variance score would be equal to 1 and mean error would be 0.

$R^2 = 1 - [(\text{Sum of Squared Residuals} / n) / \text{Variance}_{y\_actual}]$

$\text{Explained Variance Score} = 1 - [\text{Variance}_{(Y_{predicted} - Y_{actual})} / \text{Variance}_{y\_actual}]$

Variance(Ypredicted - Yactual) = (Sum of Squared Residuals - Mean Error) / n

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

A real-life predictor would also consider factors such as platform, time of posting, user posting and other basic understanding of users. I would use nltk, textblob like libraries to extract more features such as sentiment to experiment. PCA or other dimensionality reduction should also be added once features from text are generated.

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

The code presented is far from being deployed. It's a project on general introduction to NLP and linear models. I would do model and feature extraction from scratch in house so that parameters are tuned perfectly to the maximize optimization, I can also take pretrained models to develop initial features and then develop our own model on it. Deep-learning could also be used to make better models. LSTM with ReLU works almost perfect on small text data. Sentiment analysis also has been proven to work best with BERT and transformers. Moreover, I would also introduce the model to Github, Docker and cloud.

Instructions for running the code:

Python notebook attached should run perfectly fine on any system with python3.7+ from a Jupyter notebook.