
Online News Popularity Prediction

The Project

Purpose

Our task was to help online news companies predict the popularity of articles before they are published.

- Increase advertising revenue
- Enhance brand reputation

Context

- Explore data from nearly 40 - thousand online articles
- Target a success metric of 1,500 shares
- Leverage multiple computer models

Problem statement

- Choose the most effective machine learning (ML) model
- Support easy deployment for future cases

Approach

Obtain and Scrub

Clean Data

- Meet necessary assumptions
- Useable formats and labels
- Missing values
- Training and testing data

Select Features

From 59 columns

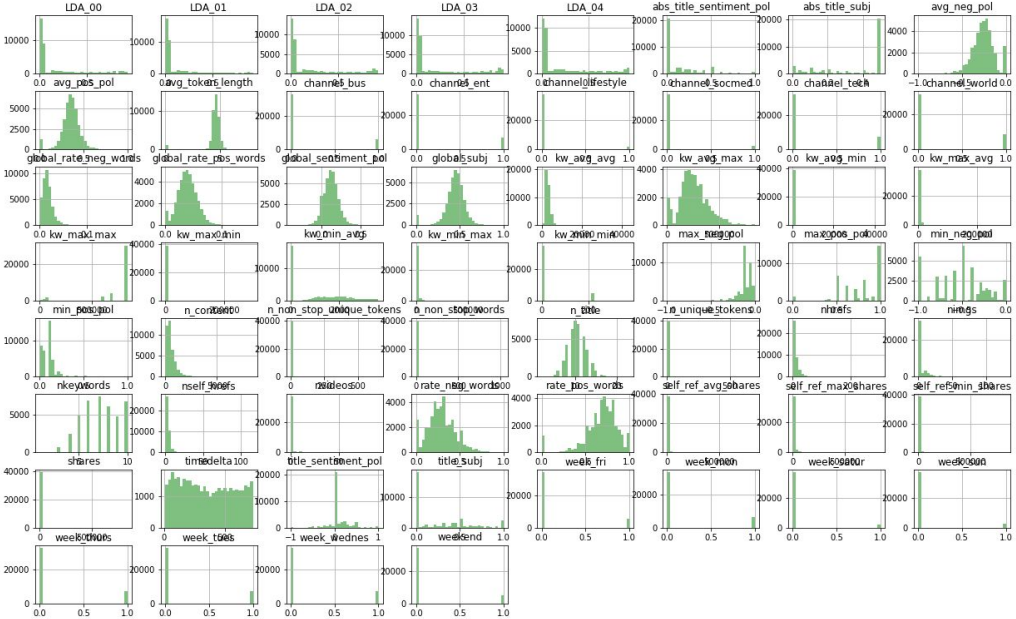
- Which article characteristics are the strongest predictors of success?
- How many do we need for reliable predictability?

Optimize

Increase conversion

Fine tune the details that make it possible for models to reach their greatest potential.

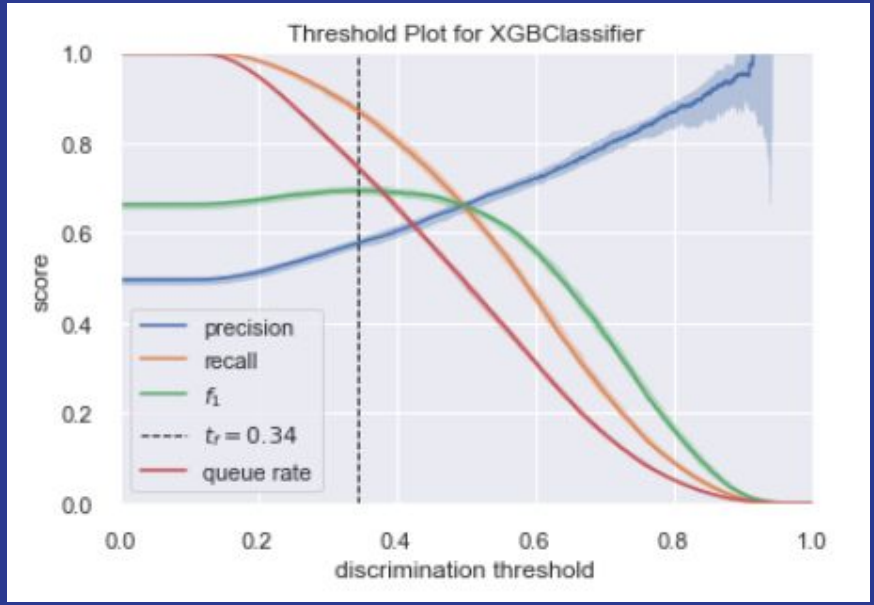
Analyzing 59 Possible Predictors



Machine Learning: Select 12 Most Important

Prioritizing Precision

	precision	recall	f1-score	support
0	0.67	0.66	0.67	5073
1	0.65	0.65	0.65	4838
accuracy			0.66	9911
macro avg	0.66	0.66	0.66	9911
weighted avg	0.66	0.66	0.66	9911



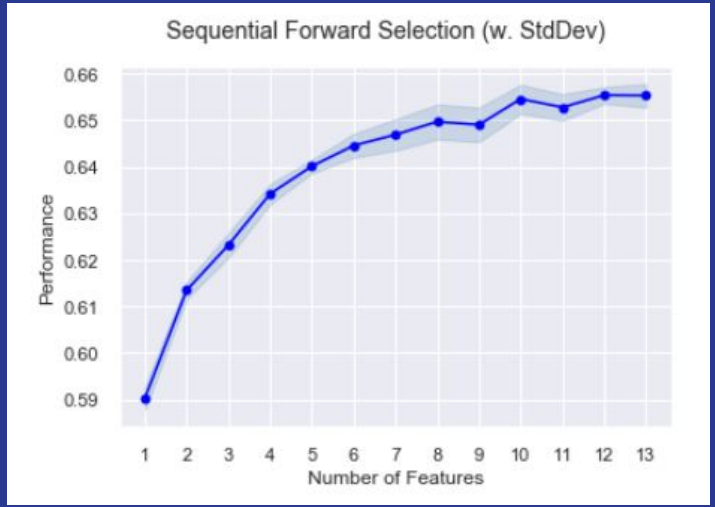
Focus on:

- Increasing Advertiser ROI
- Improving Brand Satisfaction
- Improving User Experience

Optimizing Performance

* A final XGBoost model capable of predicting whether an article will earn at least 1500 shares:
    - 67\% precision
    - 66\% accuracy
* Identification of the following best predictors as determined through forward selection, using the mlxtend library's SequentialFeatureSelector module:

- timedelta	Days between the article publication and the dataset acquisition
- n_unique_tokens	Rate of unique words in the content
- n_non_stop_words	Rate of non-stop words in the content
- channel_ent	Is data channel 'Entertainment'?
- channel_socmed	Is data channel 'Social Media'?
- channel_tech	Is data channel 'Tech'?
- kw_avg_avg	Avg. keyword (avg. shares)
- self_ref_min_shares	Min. shares of referenced articles in Mashable
- self_ref_avg_shares	Avg. shares of referenced articles in Mashable
- weekend	Was the article published on the weekend?
- LDA_02	Closeness to LDA topic 2
- LDA_03	Closeness to LDA topic 3

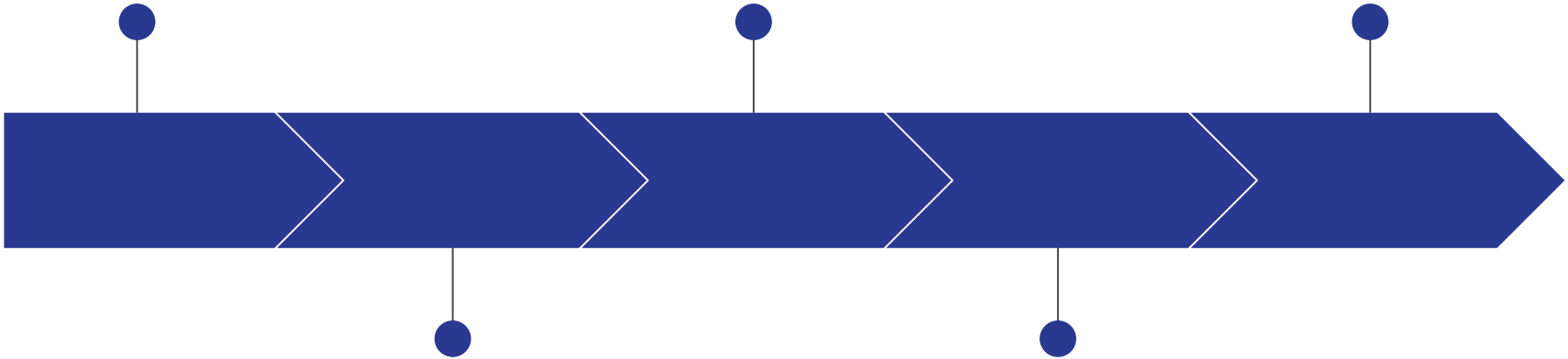


Implementation

Load Existing Data

Apply Saved Model

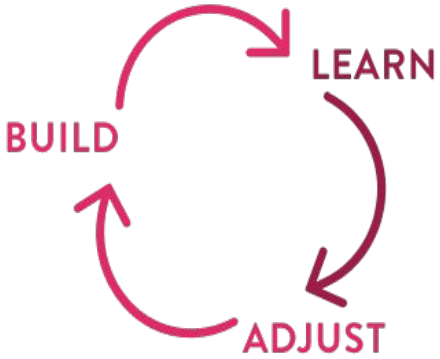
Stakeholder Decisions

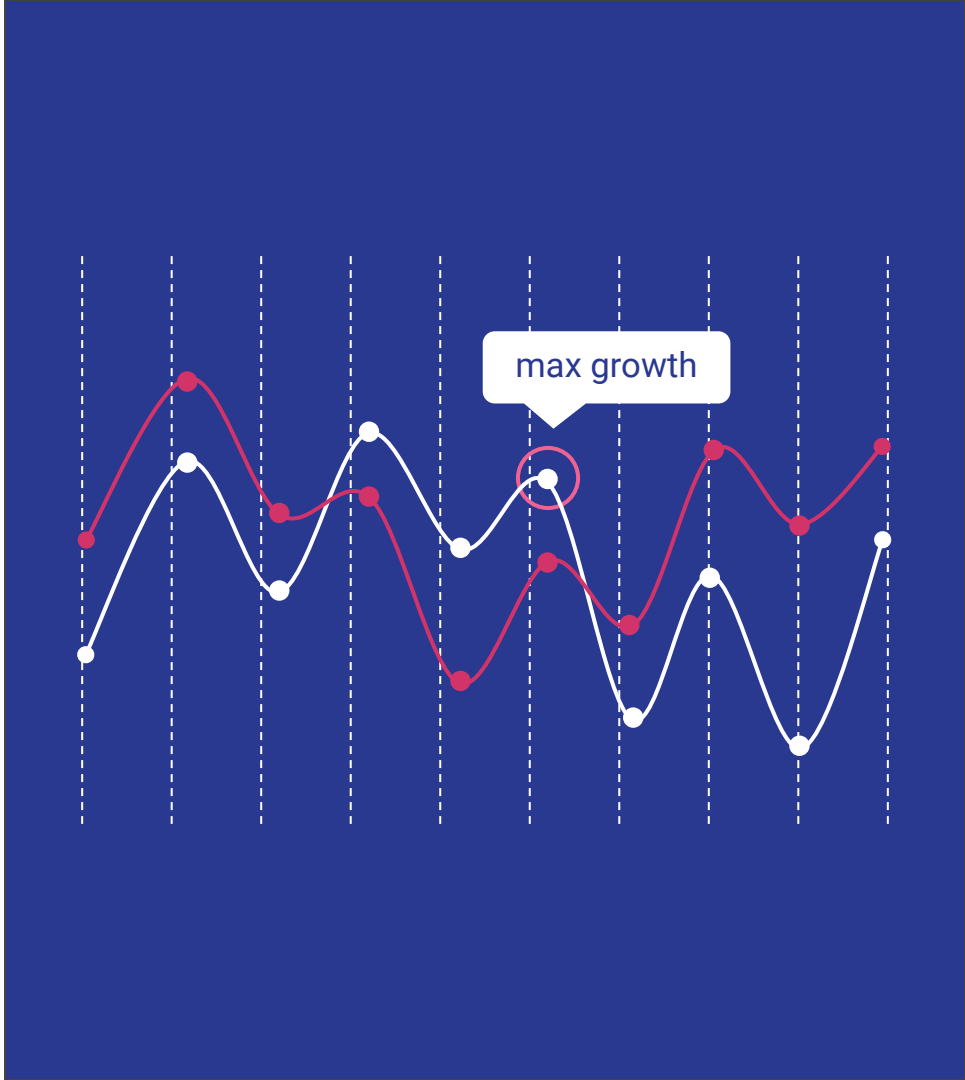


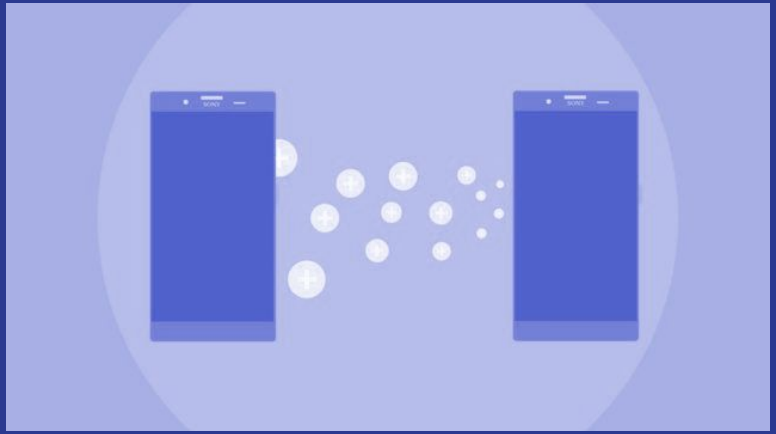
Integrate New Data

Make Predictions for
New Articles

Update and Redeploy for Different Segments







Thank you