Group 03

TECHLABS/SCHICKLER HACKATHON

Group members:

Artem Dontsov
Nour Abdennebi
Muhammad Affan Qamar

TechLabs

SCHICKLER

Goal

Ranking system for existing articles

- Score is total amount of seconds users were reading the article

Predicition of the score for newly written articles using a machine learning model

Approach

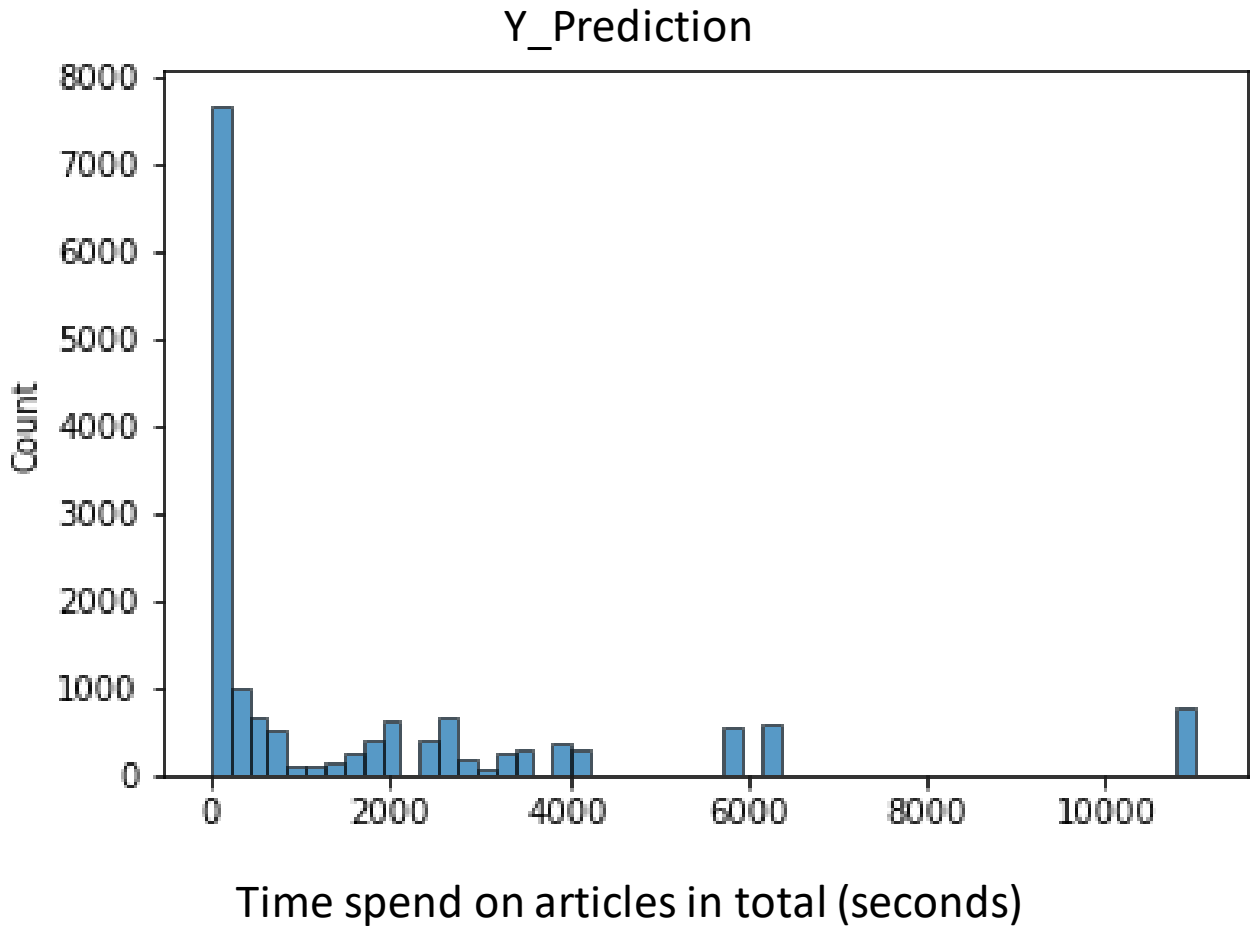
Data Science

- Clean the data
- Get some metrics out of the article full text
  - We used ARI (automated readability index)
- Create a feasible feature set for the machine learning model
  - We used all available features
  - We used a linear support vector machine for categorization

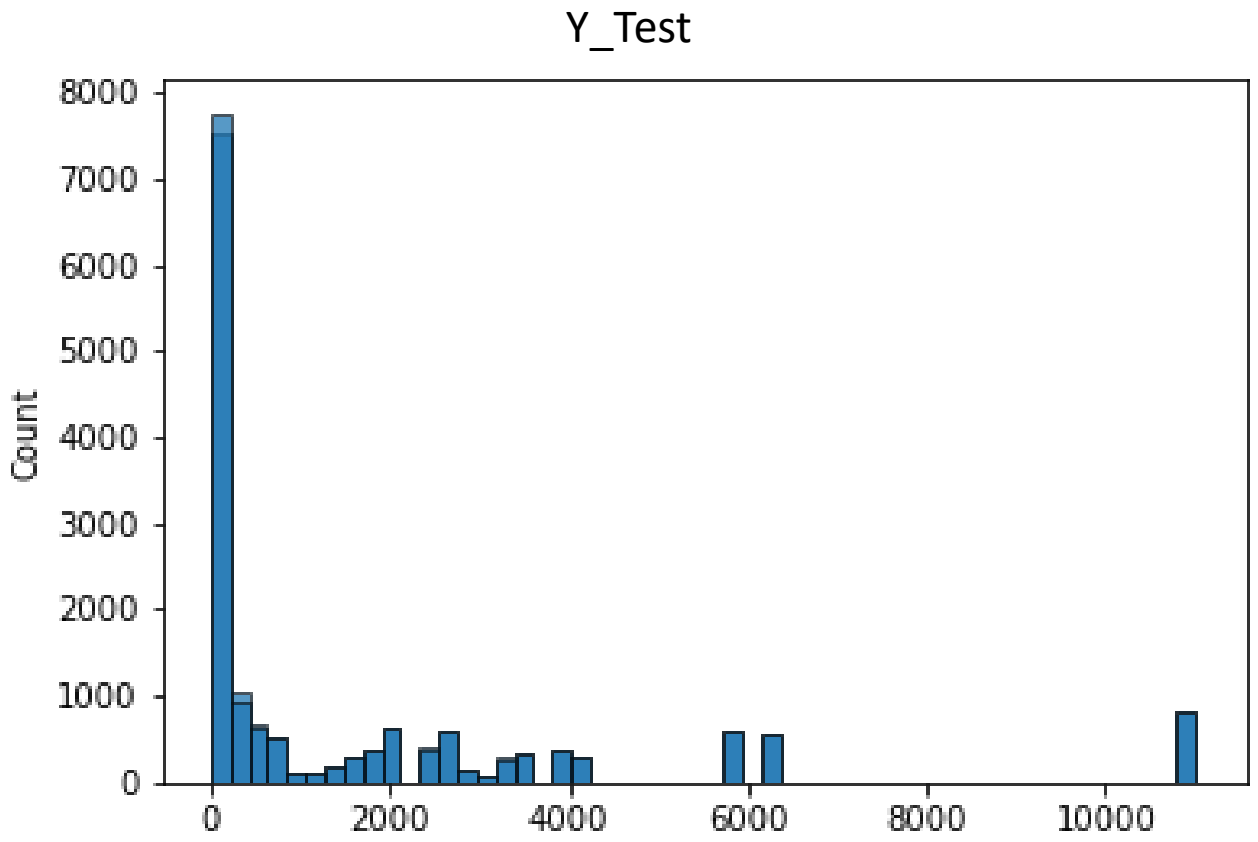
Web development

- Create a frontend prototype to display the score distribution for the article
- Create a form that calls an API to predict the score for a new article

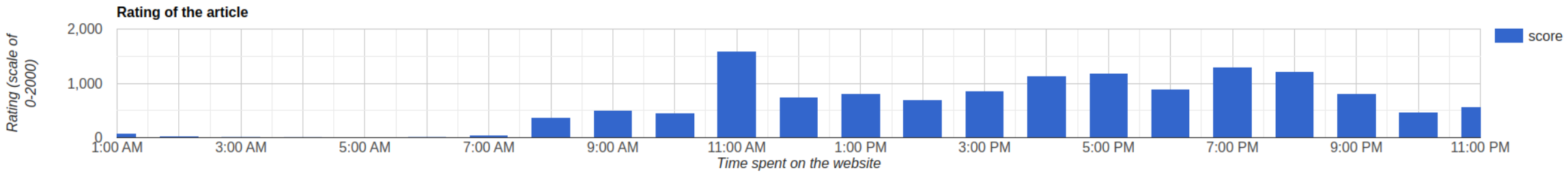
Result /1


Y_Prediction
Count
8000
7000
6000
5000
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
Time spend on articles in total (seconds)

Result /2


Y_Test
Count
8000
7000
6000
5000
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
Time spend on articles in total (seconds)

Result /3



Submit Form

Text

Paywall

True False Submit

Result /4

Stored the trained model, so we can reload it every time the API starts without needing to retrain it

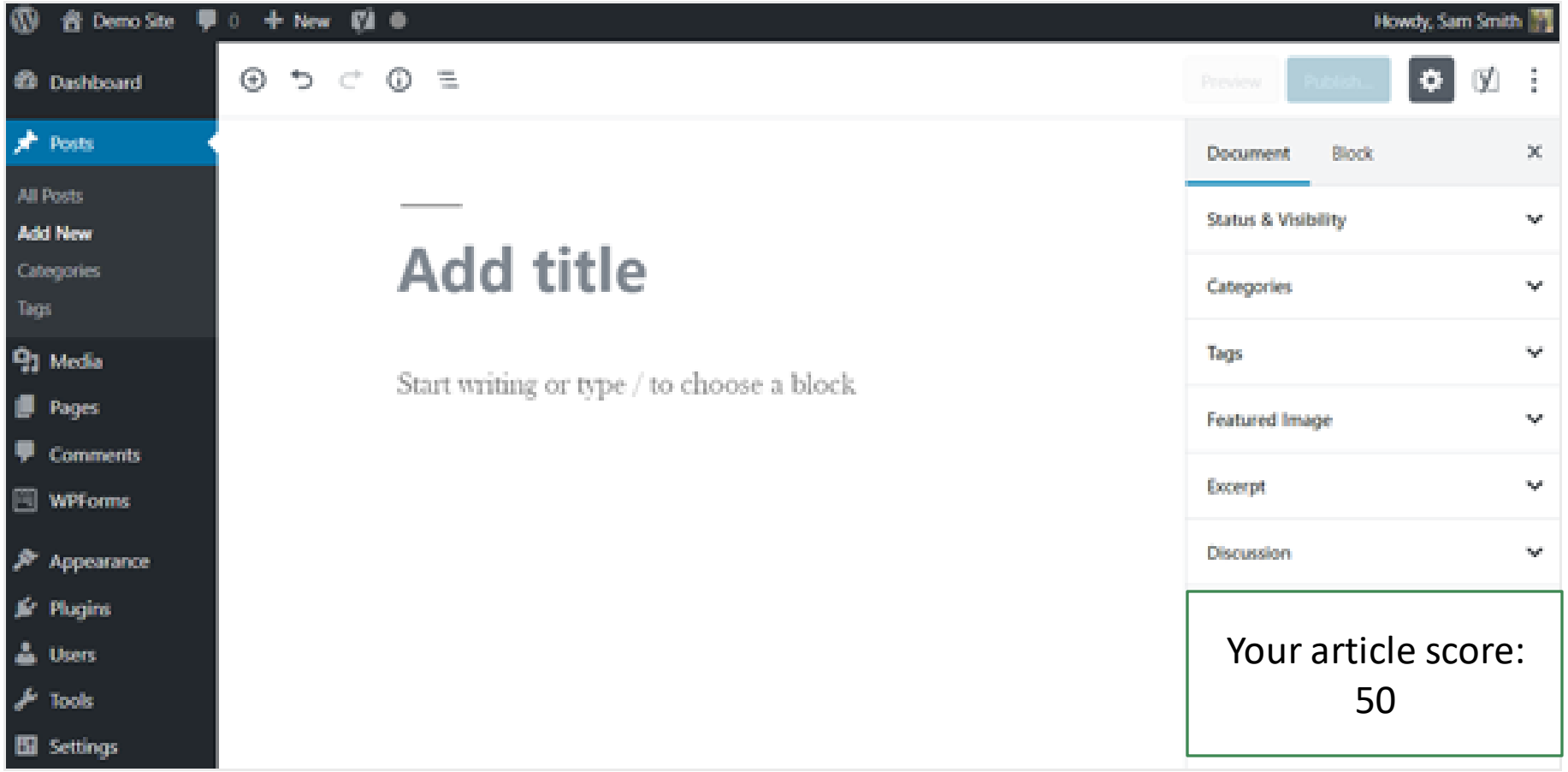
Python API using FastAPI

Result /5 - Possible errors

Used dataset has not all categorical values (for example portal_id was not complete)

Using only a batch of the page view resulted in many scores being zero, as the graphs show

Example Business Case



Next Steps

Advanced feature selection

Training the model with the complete dataset

Take into consideration the time when the content was published

Add sentiment analysis into API, currently all sentiment result values are assumed as 0