

RTB AD PRICE PREDICTIONS POSTER

Shay Bozo (ID. 301166278), Edan Hauon (ID. 305249187)

IDC, Israel

Abstract

Fyber main product act as an ad exchange platform. When we get an 'ad request' we enrich the request with more data and send the bid request with a minimum floor price to the relevant ad networks DSPs companies that want to run their ads. The purpose of the project is to predict the responses of the ad networks DSPs

The Problem in More Details

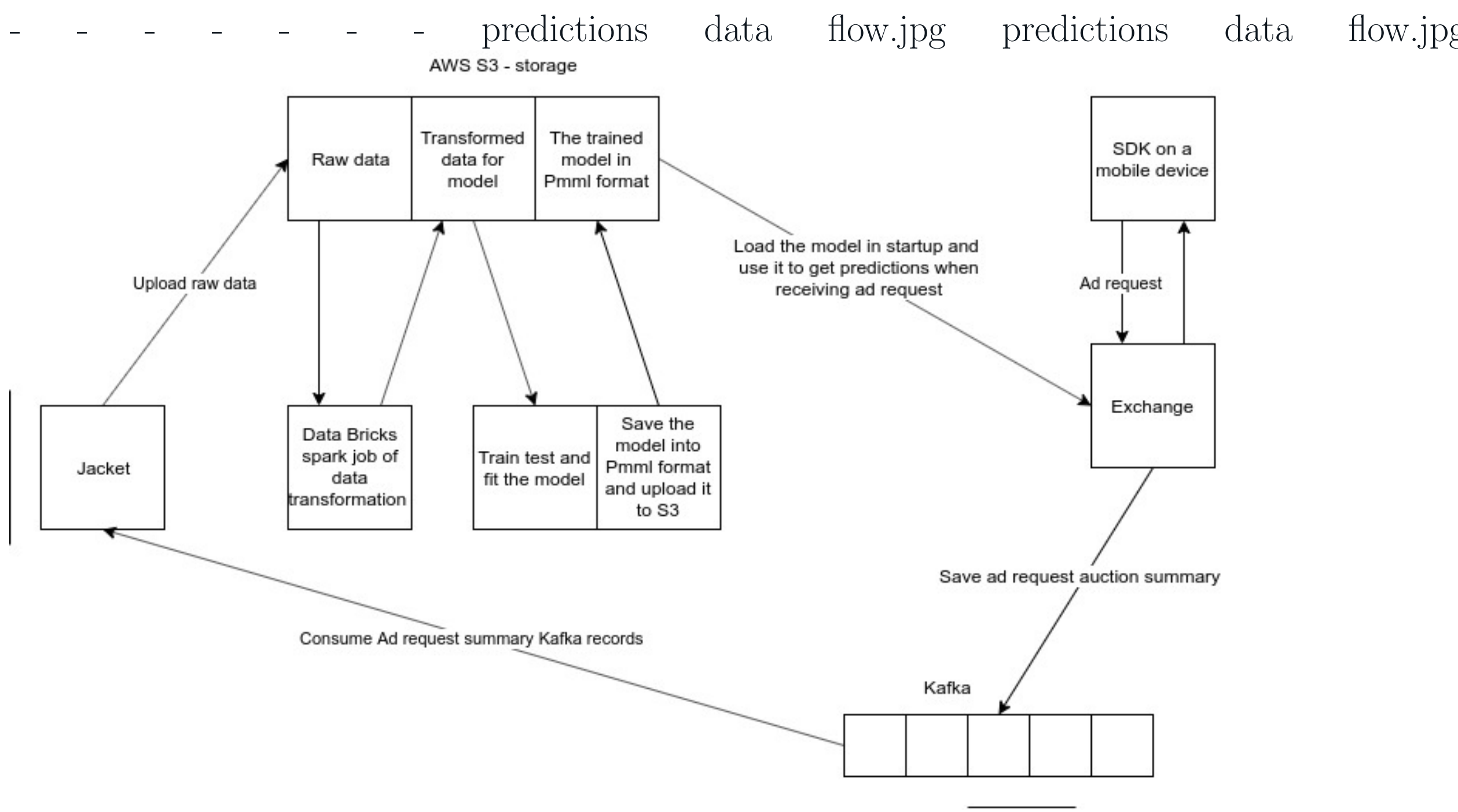
When an ad request received by the Viper web service and after I enriched it with all the relevant data I want to be able to predict in real time what will be the response of the ad network in both response status(bid, no bid, rejected, etc) and in cases of a bid also the bid price. With this prediction I can now perform a number of actions such as bid optimization - not sending the request in cases the prediction is for 'no bid' or changing the minimum auction floor price information that is about to be sent in order to get a better prediction

Method

- During the run of the Viper web service after every auction I will collect all relevant data and send it to Kafka
- From Kafka a consumer service will read and upload the raw data to S3
- A daily Spark job on DataBricks will filter, transform and encode the raw data and will save it back to S3 ready to be used
- Another daily job will train and test a DNN model using mainly TensorFlow
- The model will be export and saved back to S3 in Pmml format
- The Viper web service will load the model and use it in run time

Results

- The first job of data transformation takes 1h to run and saves two files 1 for train and 1 for test to S3 to be used later
- The second job of building a model takes 1/2h to run and saves the model to S3 in a Pmml format
- The Viper web service load the model and gets predictions in real time
- Please see other numeric predictions results in the report and the appendixes



Conclusions

- If you want to process a lot of information on a daily bases some parallelism must be done in order to get to a reasonable run time
- There is no way today of using parallelism in the training and testing phase - this sets a very hard limit on the number of records that you can process on a daily bases and your model should be build in a way that recognize it
- Most of the time should be spent on working on the data and understanding what are the important features and if you need to add new ones
- In order to get to a reasonable data processing running time it's important to reduce all the not important features that don't significantly reduce the predictions errors
- There are many tools and libraries to process and train data but most of them don't work together and many work around were needed.
- In some business ceases there is much importance to time series and there is no specific way of doing it today - I broke the date to 6 different features