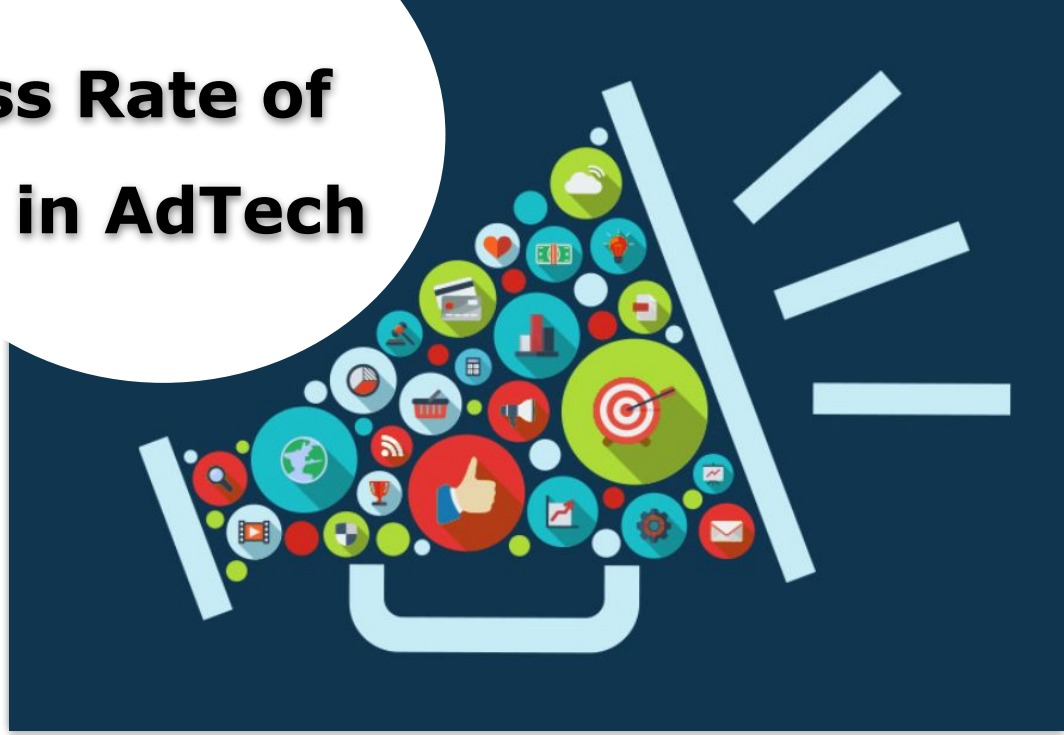
Predicting the Success Rate of Video Advertisements in AdTech

Buse Ozer

September 2020



Advisor: Boris Prodhomme

Tutor: Nacera Bennacer

CentraleSupélec

smart

Agenda

1. Introduction to Digital Advertising
2. General Overview of Advertising Ecosystem
3. Problem Definition
4. Methodology
5. Comparison of Different Models
6. Conclusion
7. Future Work

CentraleSupélec

smart

Introduction to Digital Advertising

Why Digital Display Advertising Is Important ?

High increase in digitalization

High revenue in digital advertising

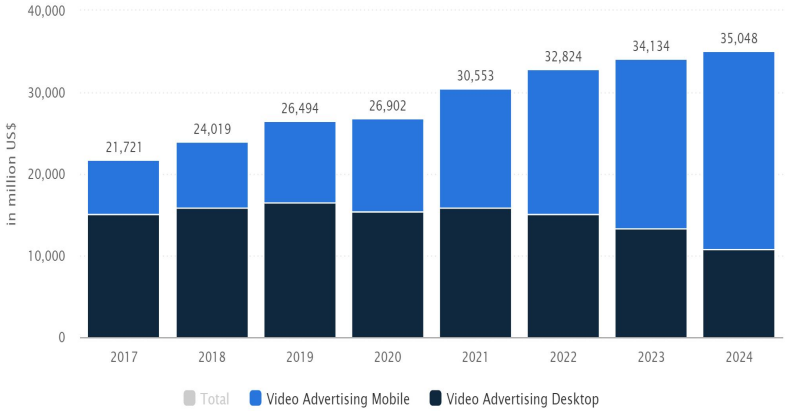
Helps to accelerate digital commitment of users

Why Are Video Advertisements Important ?

Drives more engagement

Leads to more revenue

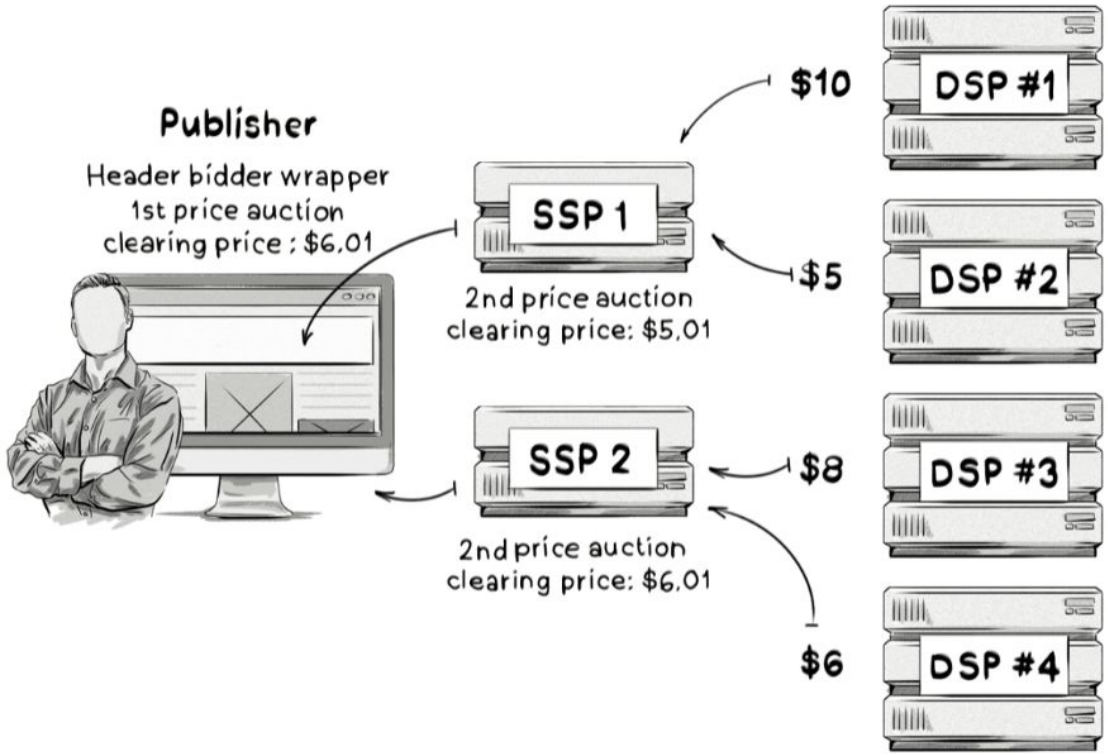
Better user experience



CentraleSupélec

smart

General Overview of Advertising Ecosystem



CentraleSupélec

smart

Problem Definition

Success Rate Definition

Probability of an ad to be delivered

$$\text{Success rate} = \frac{\text{\# of times a video ad displayed}}{\text{\# of times a video ad won an auction}}$$

Why Do We Focus on Video Success Rate ?

Our Display RTB delivery rate > **90 %**

Our Video RTB delivery rate < **30%**

Why Success Rate Optimization Is Important for Smart ?

Each loss opportunity is a direct loss in revenue for us and our publishers

CentraleSupélec

smart

The Video Obstacles

Problems

- Timeouts of media files
- Player and creative incompatibility
- VPAID issues
- User leaves the page before the ad loads
- Connection cut

Solutions

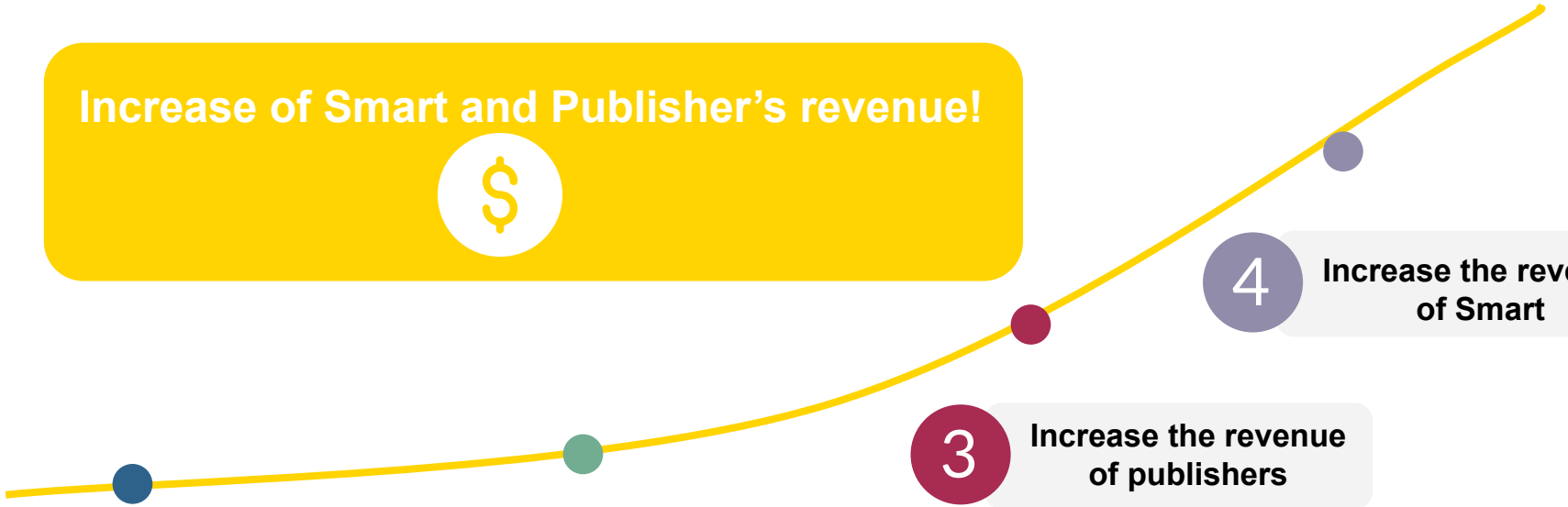
- Ensure that video ads run smoothly
- Predict the delivery rate of a bid
- Highest expected revenue

Bidder	Bid price	Delivery rate (predicted)	Expected revenue
A	$10	10%	?
B	$4	50%	?

smart

The Objective

Increase of Smart and Publisher's revenue!



1. Apply delivery rate optimization
2. Increase the delivery rate of publishers
3. Increase the revenue of publishers
4. Increase the revenue of Smart

CentraleSupélec

smart

Predictive Model

What exactly is to be predicted ?

The probability of an ad to be displayed to an end-user

What is the goal of the predictor ?

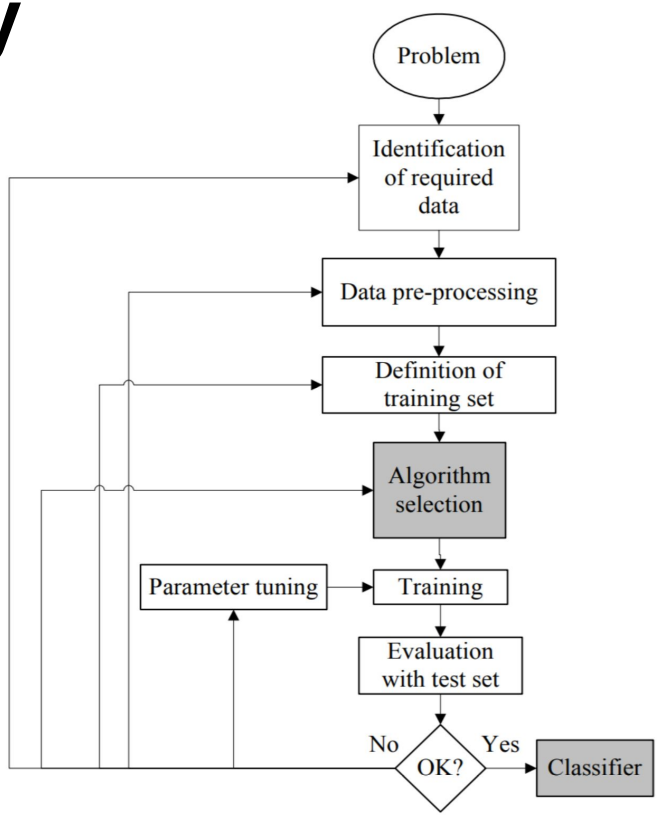
1. Be as **accurate** as possible
2. Have as **a few classification errors** as possible

What is the metric to measure the model performance ?

AUC, ability of a classifier to distinguish between classes

Among all “positive”-“negative” pairs in the dataset compute the proportion of those which are ranked correctly by the evaluated classification algorithm.

Methodology



CentraleSupélec

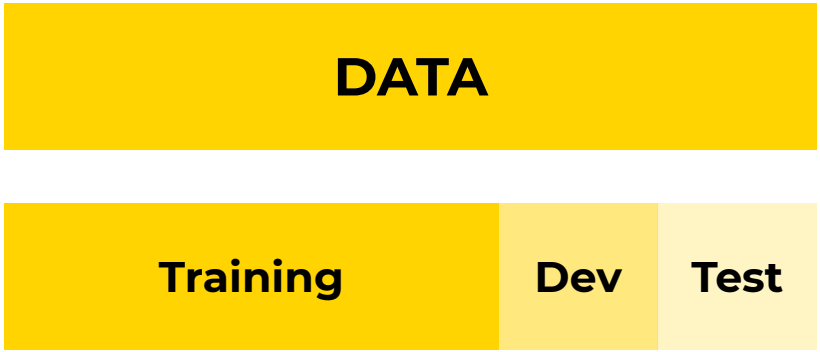
AI Platform

Google Cloud Platform

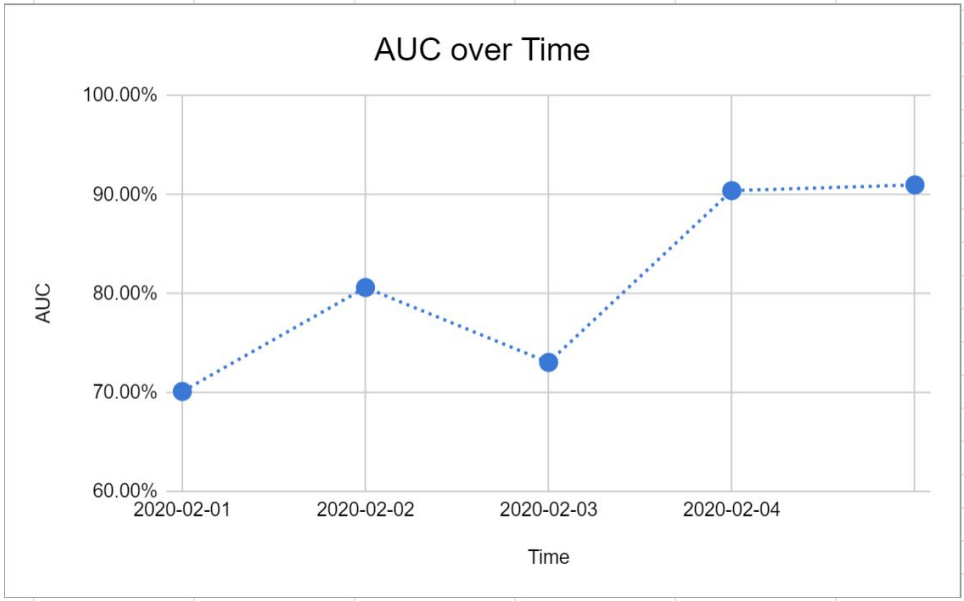
Big Query

smart

Data Set Split



Experiment	Training Set	Dev Set	Test Set
Experiment 1	Day 1	Day 2	Day 3
Experiment 2	Day 2	Day 3	Day 4
Experiment 3	Day 3	Day 4	Day 5
...			
Experiment n	Day n	Day n+1	Day n+2



Comparison of Tree-Based Models

Data Set	Model	LogLoss	Precision	Recall	Accuracy	AUC	F1
2020-02-01	Decision Tree	0.7477	0.8820	0.6223	0.6726	0.7942	0.7297
2020-02-01	Ada Boost	0.7259	0.8798	0.6099	0.6634	0.7705	0.7204
2020-02-01	Random Forest	0.6130	0.8789	0.6176	0.6678	0.7907	0.7254
2020-02-01	XGBClassifier	0.5960	0.8869	0.6242	0.6766	0.8059	0.7327
2020-02-02	Decision Tree	2.4462	0.7590	0.9019	0.7287	0.6389	0.8243
2020-02-02	Ada Boost	0.7950	0.7621	0.8882	0.7247	0.6562	0.8203
2020-02-02	Random Forest	0.4752	0.7496	0.9541	0.7452	0.7186	0.8396
2020-02-02	XGBClassifier	0.4716	0.7517	0.9358	0.7381	0.7330	0.8337
2020-02-03	Decision Tree	0.4975	0.8563	0.8996	0.8241	0.8728	0.8774
2020-02-03	Ada Boost	0.4750	0.8566	0.8960	0.8221	0.8431	0.8759
2020-02-03	Random Forest	0.3438	0.8548	0.9084	0.8285	0.8741	0.8808
2020-02-03	XGBClassifier	0.3081	0.8586	0.9062	0.8306	0.9042	0.8818

CentraleSupélec

smart

Comparison of Tree-Based Models

Model	Log Loss	AUC
Decision Tree	1.2305	0.7686
Ada Boost	0.6653	0.7566
Random Forest	0.4773	0.7945
XGBoost	0.4586	0.8144

Averaged on Multiple Days

Model	Log Loss	Precision	Recall	Accuracy	AUC	F1
Decision Tree	1.2305	0.8324	0.8079	0.7418	0.7686	0.8105
Ada Boost	0.6653	0.8328	0.7980	0.7367	0.7566	0.8055
Random Forest	0.4773	0.8278	0.8267	0.7472	0.7945	0.8153
XGBClassifier	0.4586	0.8324	0.8221	0.7484	0.8144	0.8161

Importance of Freshness of The Model

Objective

Impact of re-calculating the model as frequent as possible

Result

Hourly basis gives **4%** better AUC

The most current data

↓

Model performs better

Before

Trained on Day n-1

Tested on Day n

82% of AUC

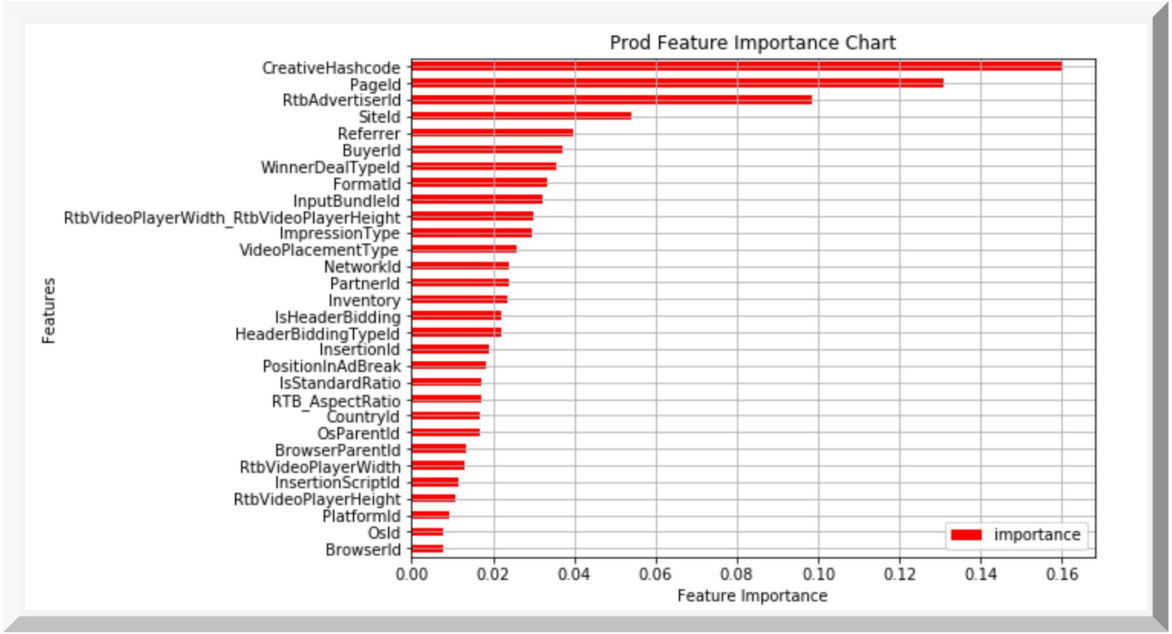
After

Trained on hour n-25, n-23, ..., n-1

Tested on hour n, n+1, n+2, n+3

86% of AUC

Adding New Dimensions



3.21 %

New Features from RTB

PageId	OsParentId
BuyerId	OsId
WinnerDealTypeId	BrowserId
FormatId	RtbVideoPlayerWidth
CountryId	RtbVideoPlayerHeight
InsertionId	VideoPlacementType
BrowserParentId	PositionInAdBreak

CentraleSupélec

smart

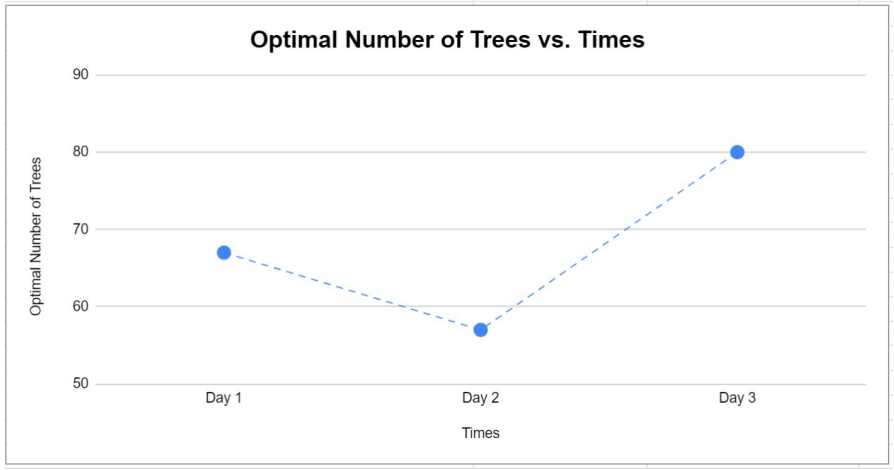
Early Stopping Strategy

Problem

There is a high variance between model complexity over time

Solution

Applying early stopping algorithm in order to control the model complexity



Early Stopping

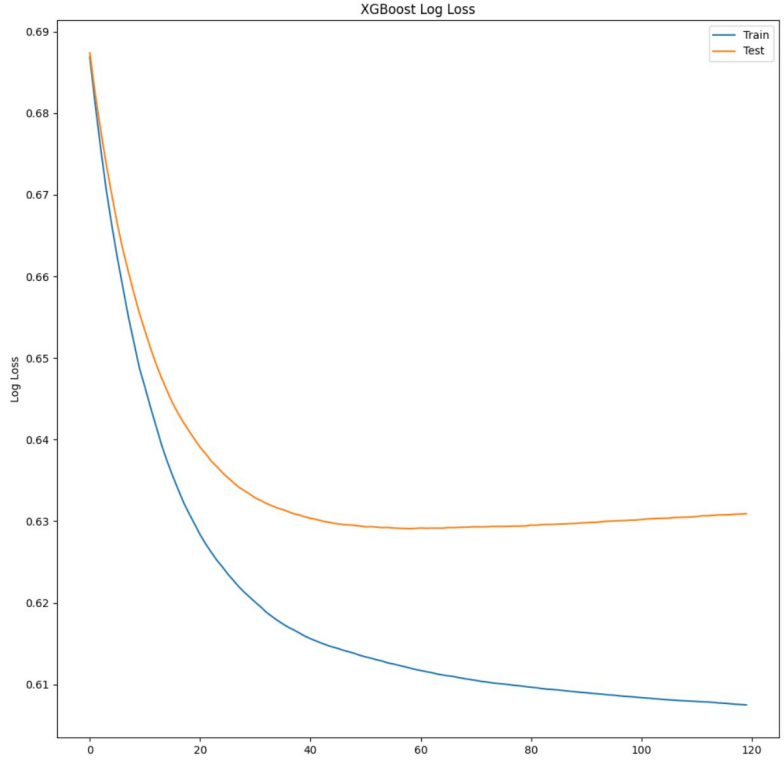
Use training and dev sets

Keeps on training the model until there is no more improvement on dev set

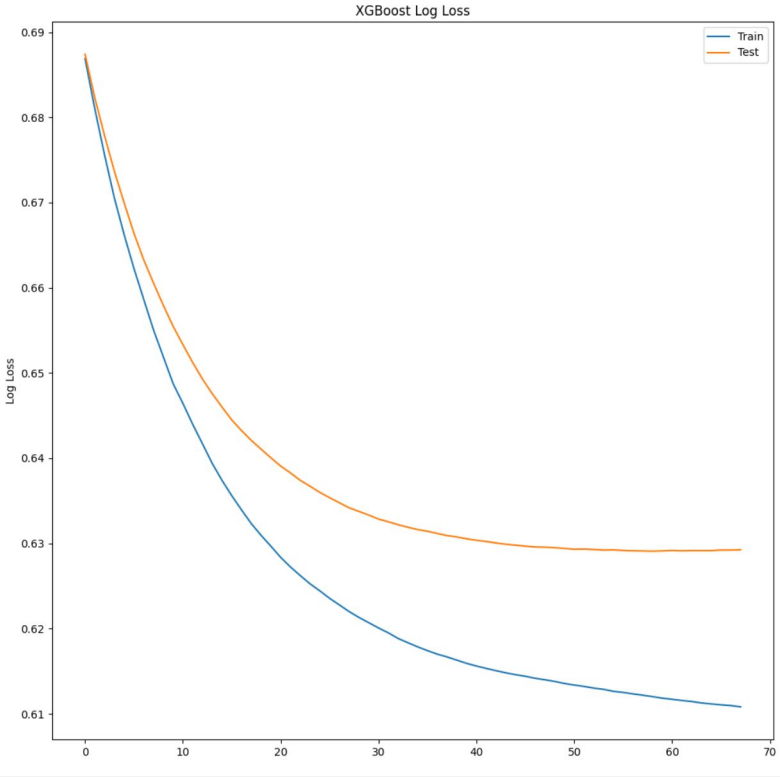
Determine how many trees to be built before overfitting

Prevents Overfitting

Without Early Stopping Algorithm



With Early Stopping Algorithm



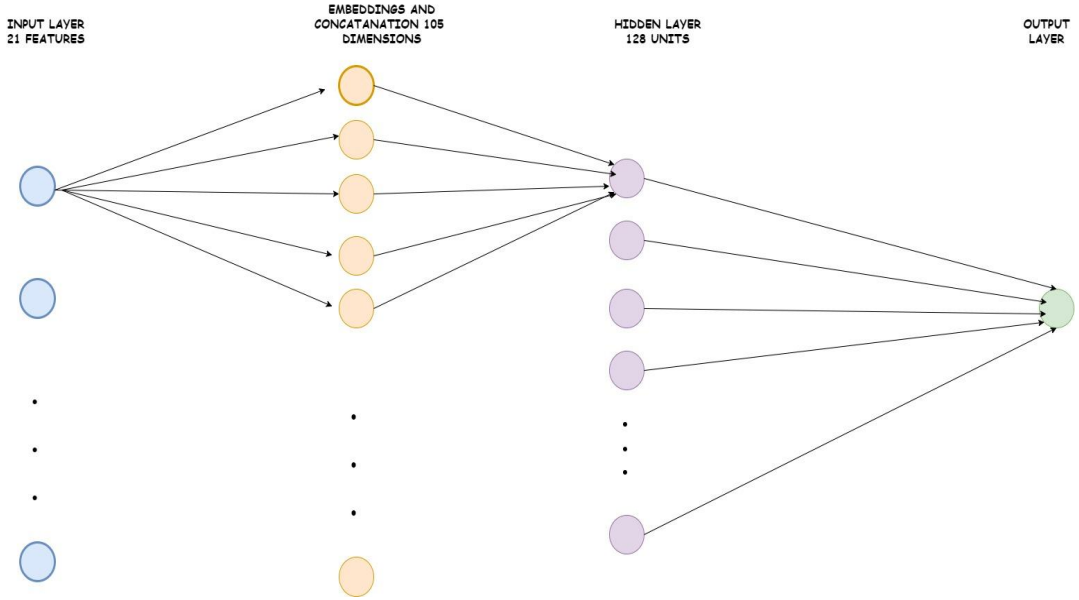
Neural Network Model

Why Neural Network Model?

- Entity Embedding
- Ability to apply continuous learning
  - Ability of a model to learn continuously
  - Progressive and adaptive learning as new data comes in
  - Overcome forgetting of previously seen data
- Transfer Learning
  - The ability to transfer knowledge from one domain to another
  - Train a model on different tasks at the same time
- TensorFlow Serving in production

Neural Network Model

- Embedding layer with 5 dim
- 1 hidden layer with 128 units
- Dropout with 0.3 probability



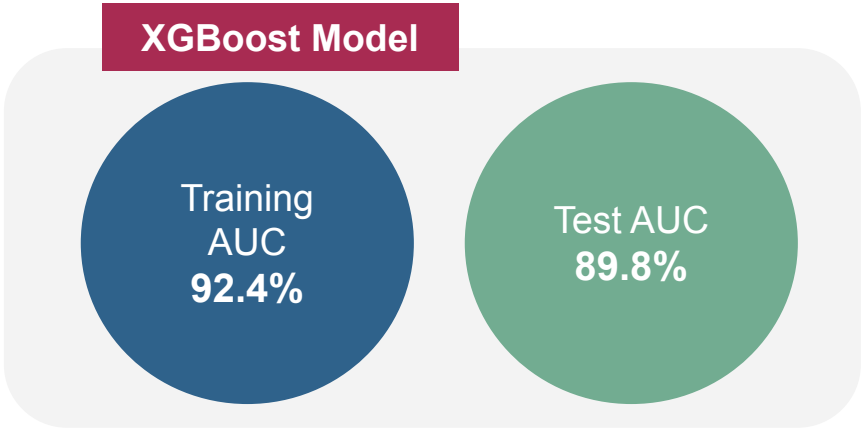
TensorFlow

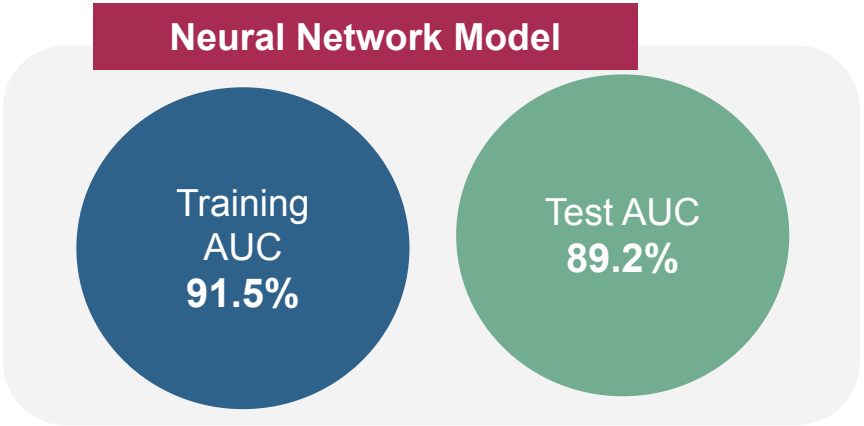
Keras

CentraleSupélec

smart

Final Results





CentraleSupélec

smart

Conclusion

- There is **no trend** in AUC over time
  - Run experiments on multiple days
- XGBoost model **outperforms** Decision Tree, AdaBoost and Random Forest
  - Better AUC, better **control over the model**
  - Scalable and parallelizable
- High variance in data
  - The model has to be **adaptive** and **flexible**
  - Early Stopping mechanism
- XGBoost and Neural Network performs nearly the same
  - Neural Network requires more expertise
- Implementation of Neural Network has some advantages
  - Continuous Learning
  - Serving in production
  - Transfer Learning

smart

Future Work

- Change the weighting strategy of the XGBoost model
  - Weighting on the revenue of samples to focus where the impact is high
- Continuous Learning in Neural Network Model
- Anomaly Detection
  - Detect spaces where delivery late is low
  - Finding the responsible publisher for the anomaly

Thank you !

Predicting the Success Rate of Video Advertisements

Buse Ozer

September 2020

CentraleSupélec

smart

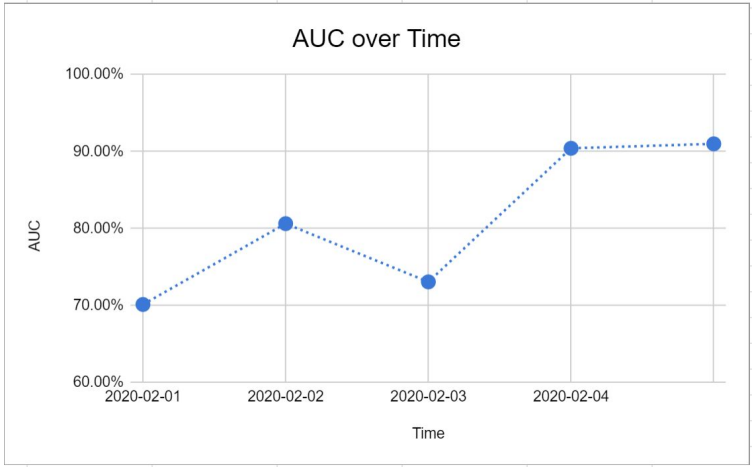
References

- Page 3, https://www.statista.com/outlook/218/100/video-advertising/worldwide
- Page 4, https://clearcode.cc/blog/sequential-auctions-header-bidding-first-price-second-price-auctions/
- Page 9, Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. International Journal of Computer Trends and Technology (IJCTT), 48(3), 128-138.


Q&A

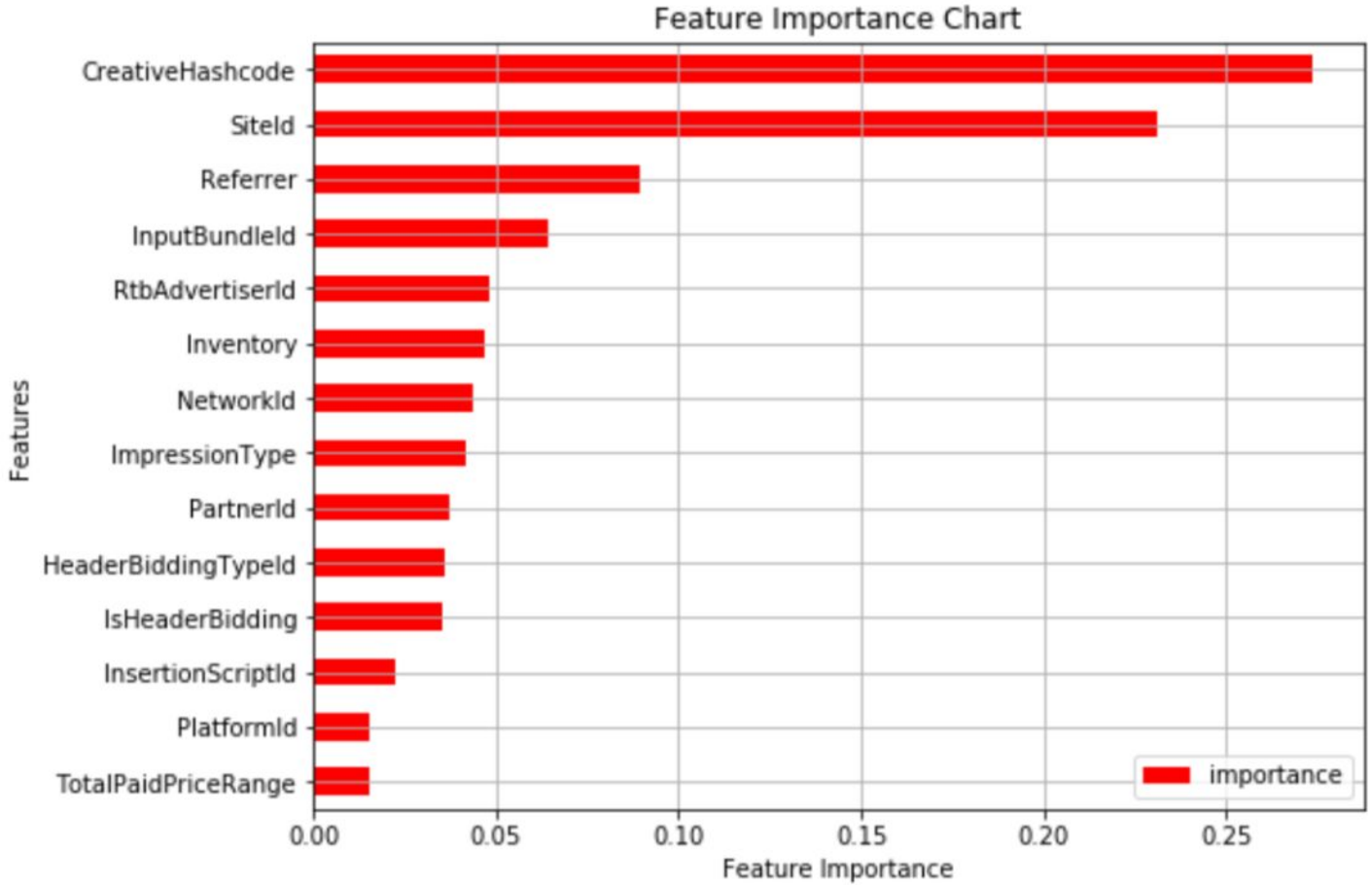
Tuning Hyperparameters of the Model

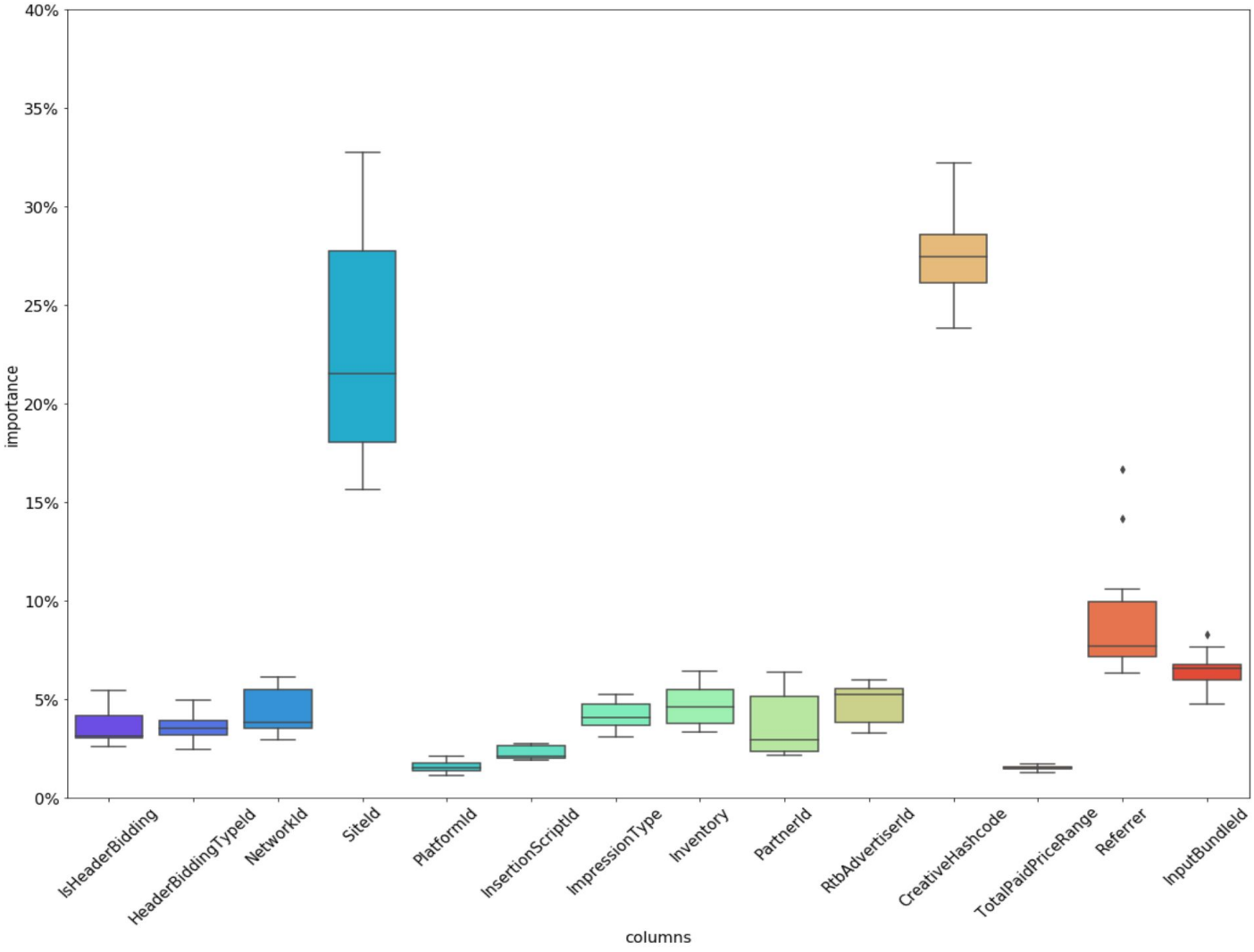
- Grid Search
- Tested on multiple days
- Best results are obtained with the parameters below
  - Max_depth = 20
    - Maximum tree depth for base learners
  - Learning_rate = 0.05
    - Weighting factor for the corrections from existing trees
  - Colsample_bytree = 0.6
    - The percentage of dimensions to built the trees
  - N_estimators = 140
    - Number of trees to use in XGBoost



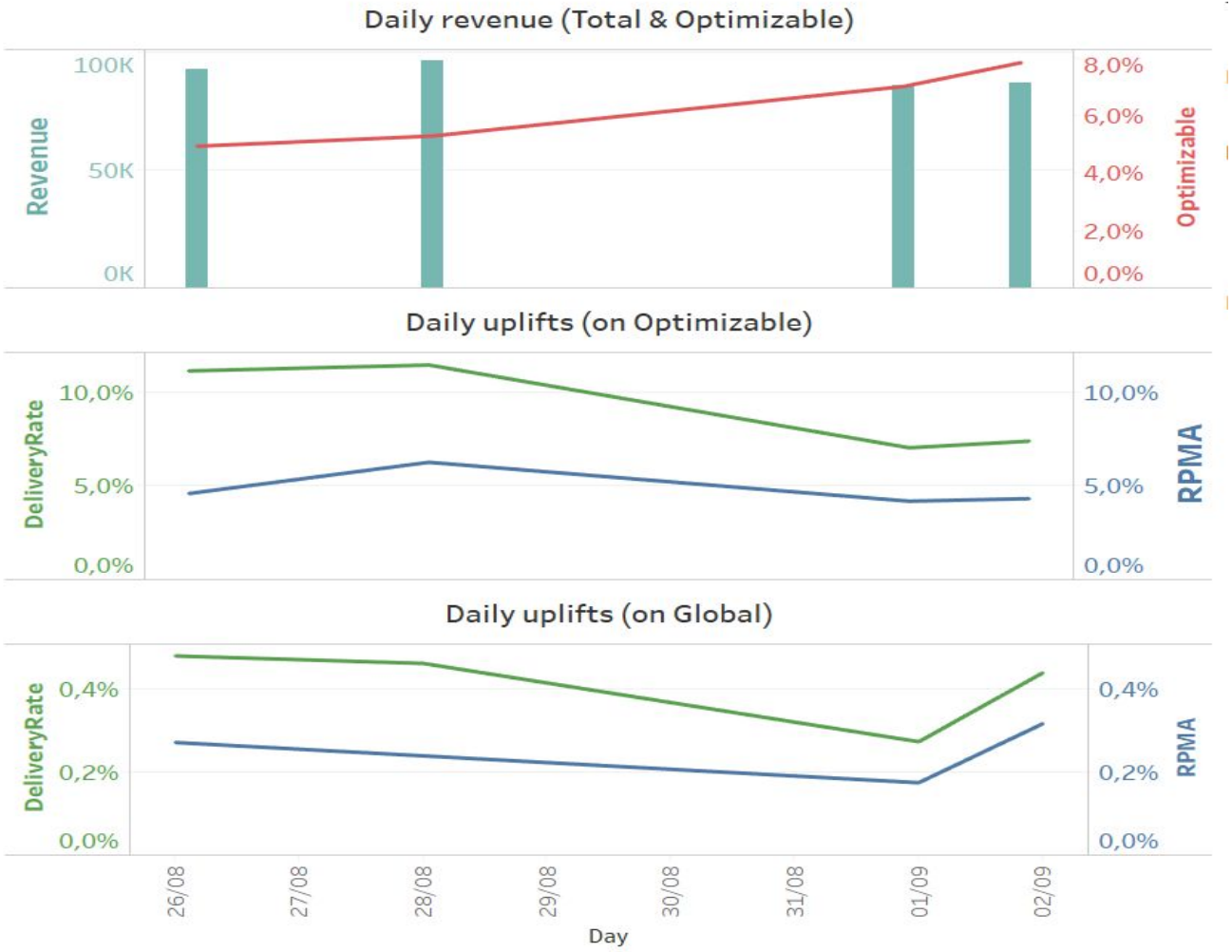
CentraleSupélec

smart


Feature Importance Chart
CreativeHashcode
SiteId
Referrer
InputBundleId
RtbAdvertiserId
Inventory
NetworkId
ImpressionType
PartnerId
HeaderBiddingTypeId
IsHeaderBidding
InsertionScriptId
PlatformId
TotalPaidPriceRange
Features
importance
0.00
0.05
0.10
0.15
0.20
0.25
Feature Importance


40%
35%
30%
25%
20%
15%
10%
5%
0%
importance
IsHeaderBidding
HeaderBiddingTypeId
NetworkId
SiteId
PlatformId
InsertionScriptId
ImpressionType
Inventory
PartnerId
RtbAdvertiserId
CreativeHashcode
TotalPaidPriceRange
Referrer
InputBundleId
columns

smart


Daily revenue (Total & Optimizable)
Revenue
100K
50K
0K
Optimizable
8,0%
6,0%
4,0%
2,0%
0,0%
Daily uplifts (on Optimizable)
DeliveryRate
10,0%
5,0%
0,0%
RPMA
10,0%
5,0%
0,0%
Daily uplifts (on Global)
DeliveryRate
0,4%
0,2%
0,0%
RPMA
0,4%
0,2%
0,0%
26/08
27/08
28/08
29/08
30/08
31/08
01/09
02/09
Day

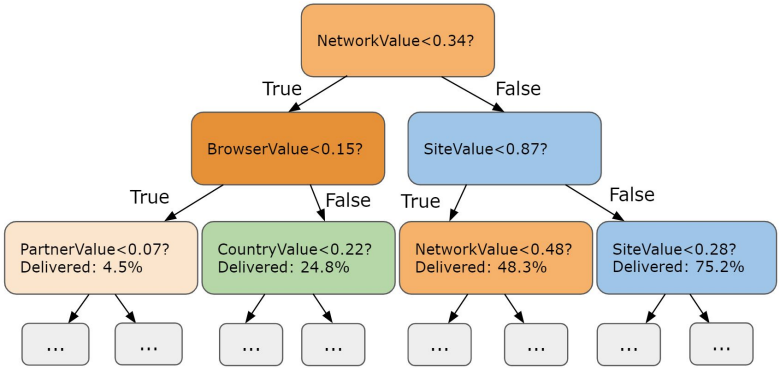
Only Video inventory
- Optimizable auctions = multi-bid auctions
- Optimized auctions = Optimizable and DROS was called
- Optimizable but not optimized = DROS was not called (for A/B testing)

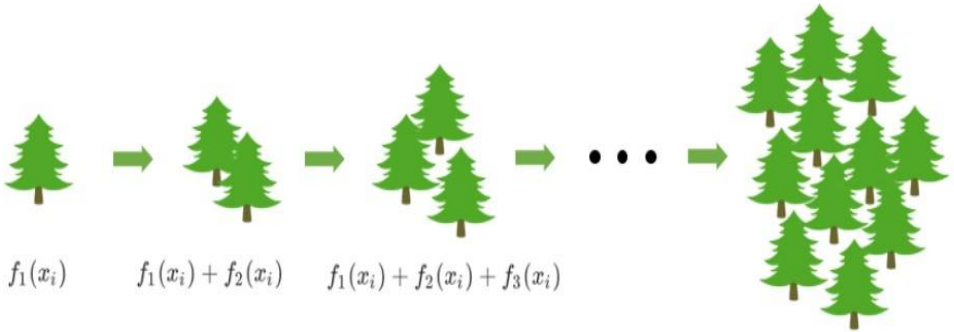
OptimizableRatio (Impressions)	4,8%
OptimizableRatio (Revenue)	6,2%
DeliveryRate (Global)	21,3%
DeliveryRate (Optimizable but not Optimized)	20,7%
DeliveryRate (Optimized)	22,5%
DeliveryRateUplift (on Optimizable) (pts)	1,9pts
DeliveryRateUplift (Global) (pts)	0,1pts
DeliveryRateUplift (on Optimizable) (%)	9,2%
DeliveryRateUplift (Global) (%)	0,4%
RevenuePerThousandWin (Global)	1,07 €
RevenuePerThousandWin (Optimizable but not Optimized)	1,33 €
RevenuePerThousandWin (Optimized)	1,39 €
DailyRevenueUplift (Global)	226 €
RevenueUplift (on Optimizable)	4,9%
RevenueUplift (Global)	0,3%

XGBoost Model

Model

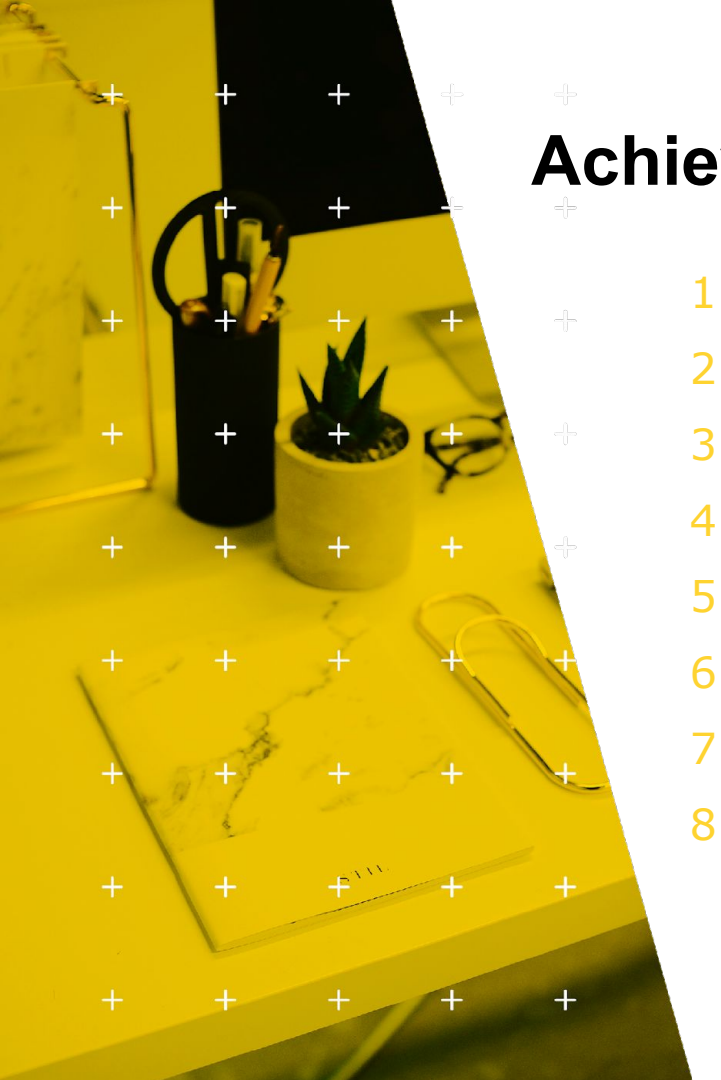
- Tree-based
- Ensemble Learning
- Consists of many simple trees
- Learns mistakes from previous trees
- Builts sequential trees





CentraleSupélec

smart



Achievements

1. Moving all experiments to Google Cloud Platform
2. Importance of Freshness of the Model
3. Adding New RTB dimensions to the model
4. Analyzing Cross Features
5. Tuning Hyperparameters of the Model
6. Early Stopping Algorithm
7. Pushing the new model to production
8. PoC of a new model with Deep Learning

smar+

Click to learn more

1 Moving all Experiments to Google Cloud Platform

Problems

- ❏ Resource limitation
- ❏ Time limitation
- ❏ Everything is in local machines

Solution

Google Cloud Platform


AI Platform


Big Query


GPU

How ?

- ❏ Allocate resources as much as needed
- ❏ Improved Performance
- ❏ Allows collaboration, accessible amongst team members
- ❏ Cloud Billing Account
- ❏ State of the art security

smart

2 Better Process for Evaluating the Model

In production

- Learn from past
- Predict on the present

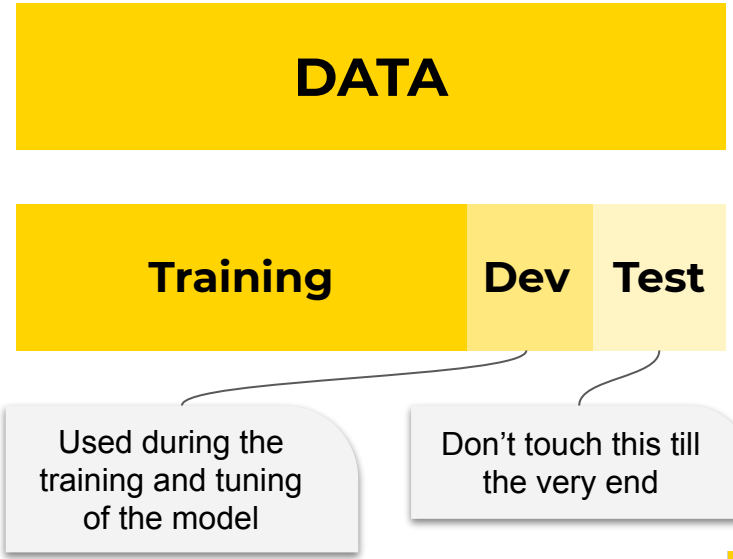
Goal

- Reproduce “in-production” conditions

How ?

- Split historical dataset into
  - Training/Testing
- Time constraint

Data Set Split



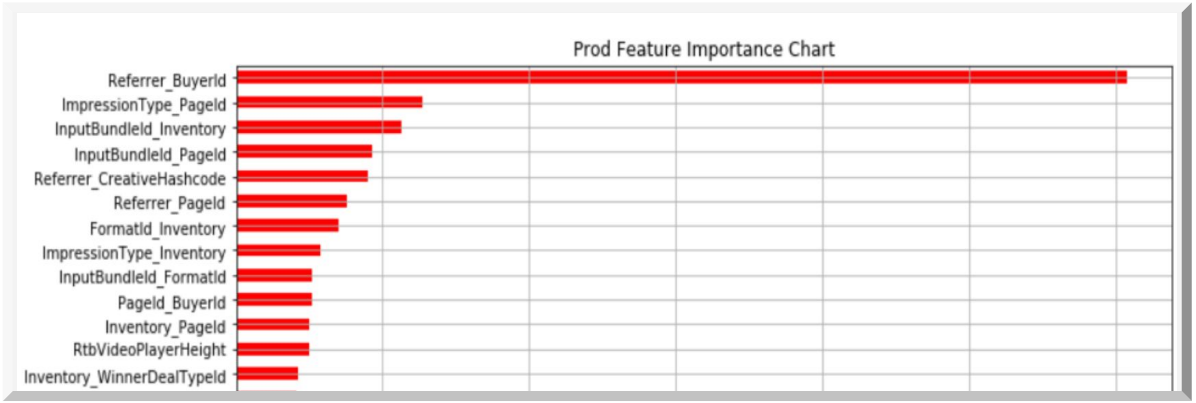
smart

4 Analyzing Cross Features

Click to learn more

Objectives

- See the most important feature combinations
- Find incompatibilities between supply and demand dimensions
- Gain knowledge for anomaly detection



smart

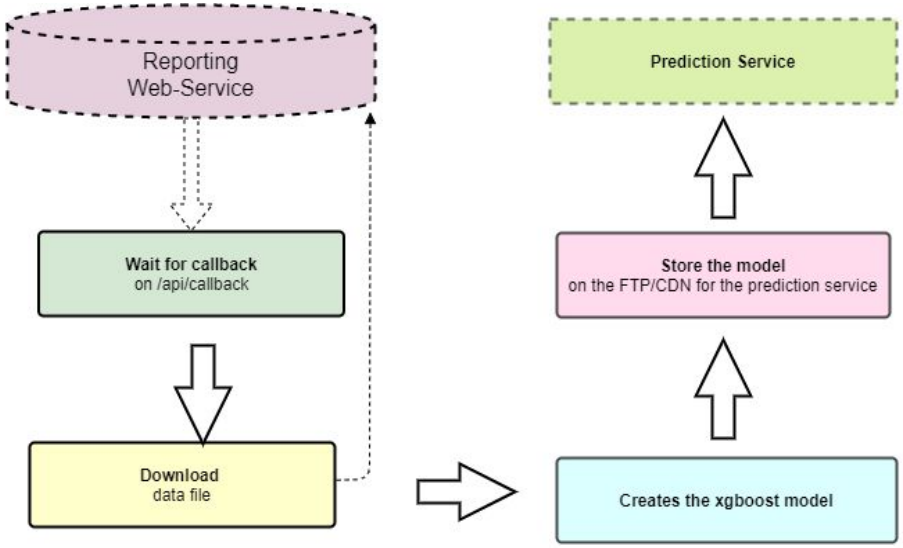
7 Pushing the new model to production

Delivery Rate Learning Service

Deployed the model in production

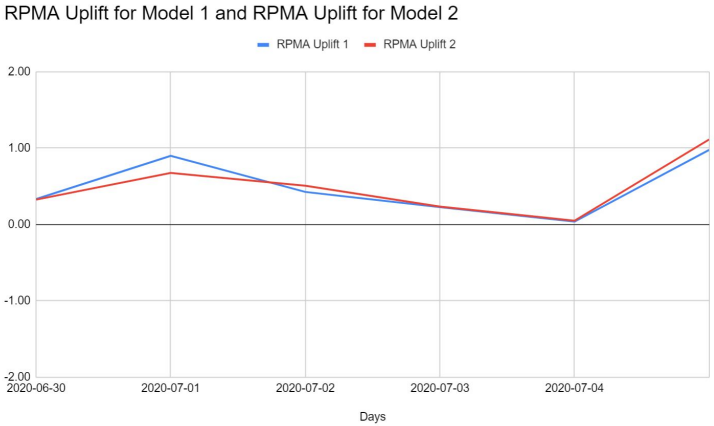
Scheduled a new reporting for model B

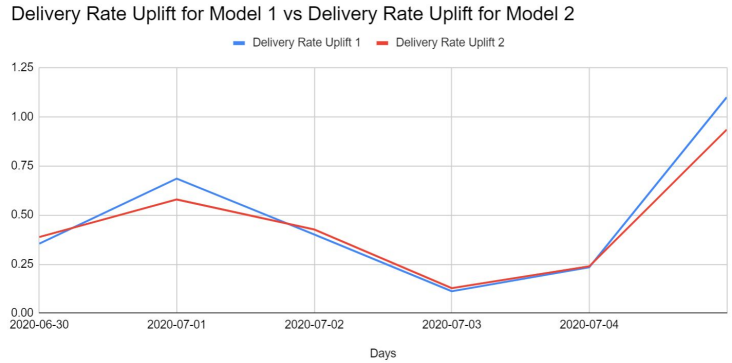
Created new configuration for prod 2



python Flask scikit learn XGBoost TC +ableau smart

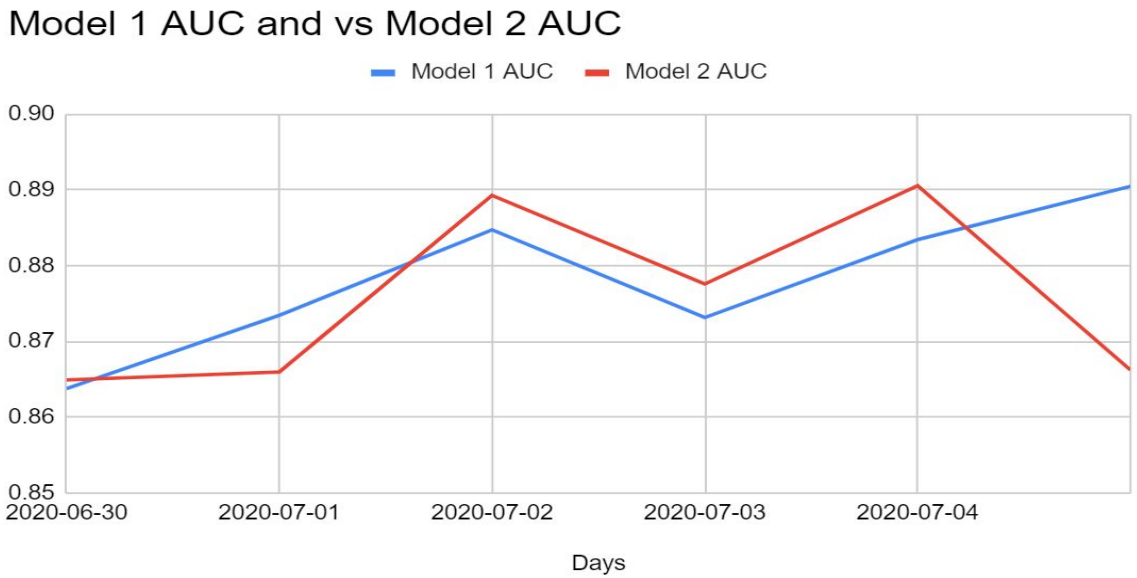
The First Results



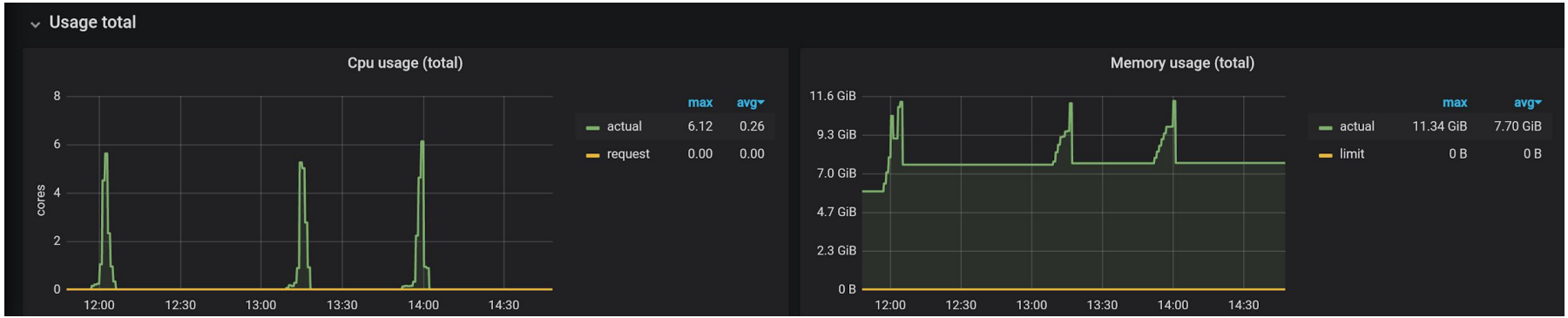


- 06-30
  - Activation of the new model, same configuration
- 07-01
  - Making experiments
    - Adding new dimensions
    - Changes of parameters of the model
    - Testing different conf on reporting scheduling
      - Model B is unstable
      - Learn less, less precise
- 07-02, 07-03, 07-04
  - No changes, model B is stable
  - Better precision
  - No uplift when there is no anomalies
- 07-05
  - Tests for memory leak
    - Model B is unstable
    - Learn less, less precise

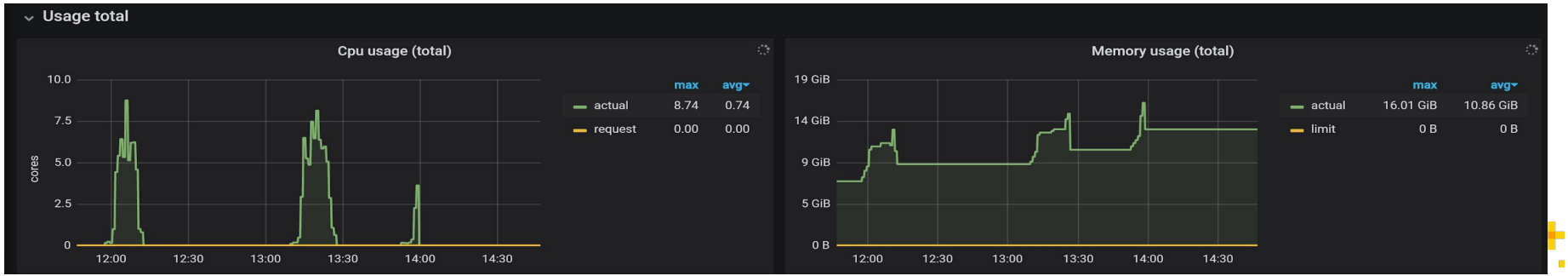
The First Results



Model 1



Model 2



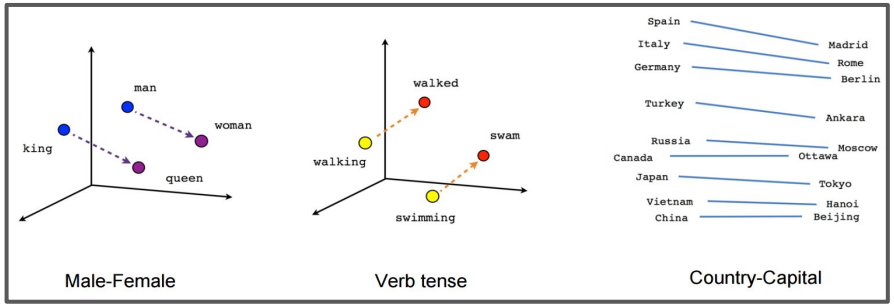
8

PoC of a new model with Deep Learning

Handle Categorical Data with Deep Learning

- One Hot Encoding
  - Sparse way of representing
- Target Encoding
  - Replacing with the mean of the target
- Embeddings
  - Convert discrete values to a vector of continuous values

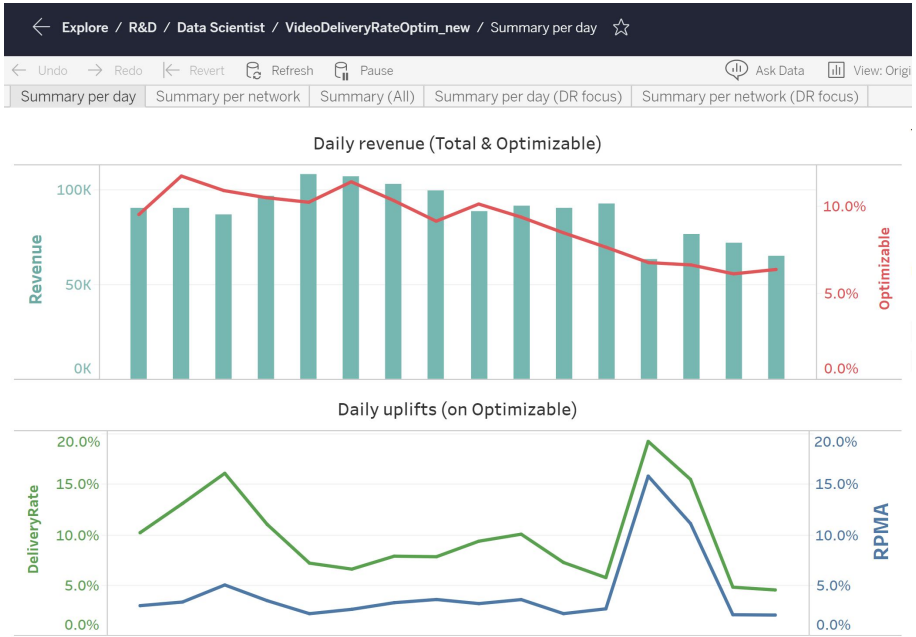
puppy	[0.9, 1.0, 0.0]
dog	[1.0, 0.2, 0.0]
kitten	[0.0, 1.0, 0.9]
cat	[0.0, 0.2, 1.0]

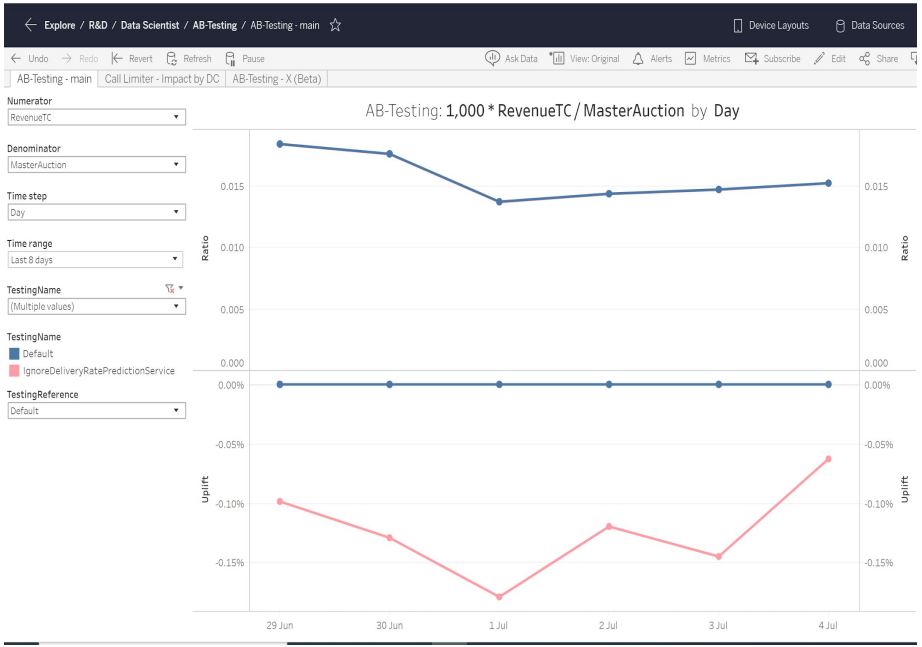


Video Optim - Next Steps

- Monitoring revenue uplift
  - Investigate results, find where DROS has big uplift / downlift
- Comparing model A and model B
- Anomaly detection workflow
  - Find the network responsible
  - Get the responsible dimensions (CreativeHashCode, Referrer)
- Exploration of VAST data + add important VAST dimensions to the model
- Change weighting strategy of the model
  - *Revenue* and *OptimizableRatio*
  - Focus where the impact is high
- Implementing continuous learning for deep learning model
- Industrialization of Deep Learning Model

Tableau Dashboards



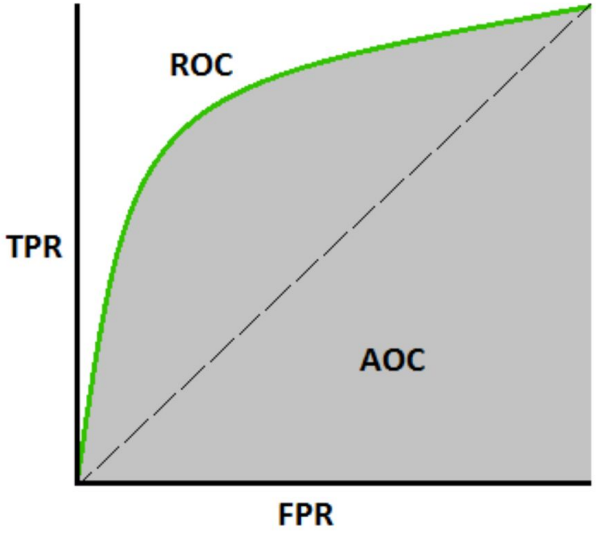


CentraleSupélec

smart

Area Under the Curve

ROC (**Receiver Operating Characteristics**)

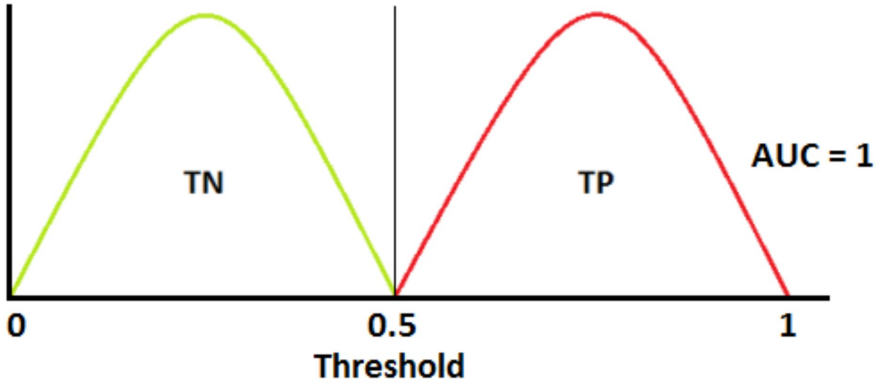


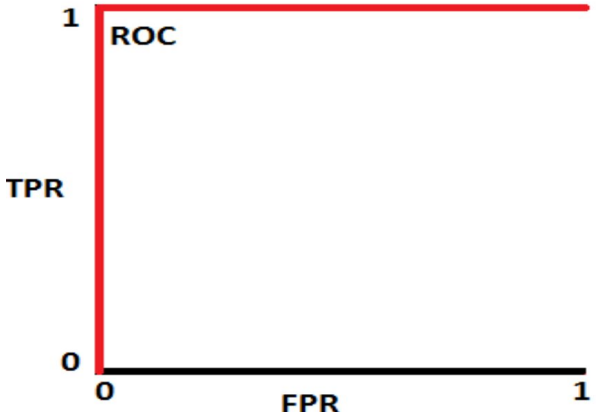
$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

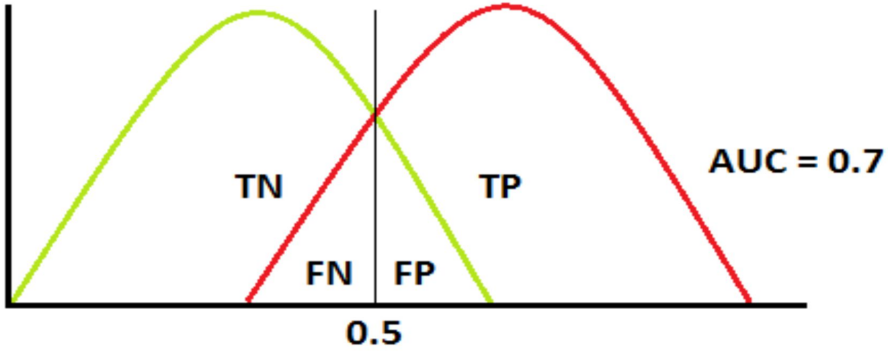
$$\text{FPR} = 1 - \text{Specificity} = \frac{FP}{TN + FP}$$

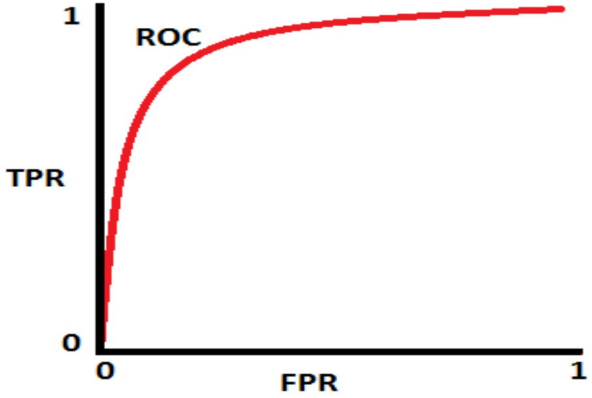
ROC

curve of probability









XGBOOST LOSS FUNCTION

$$\left[\sum_{i=1}^{n} L(y_i, p_i)\right] + \frac{1}{2}\lambda O_{value}^2$$

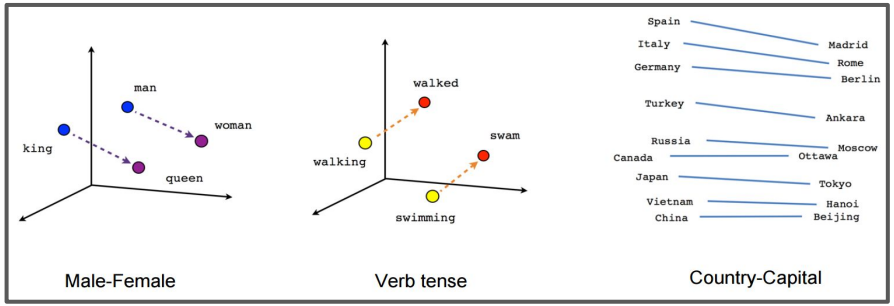
8

PoC of a new model with Deep Learning

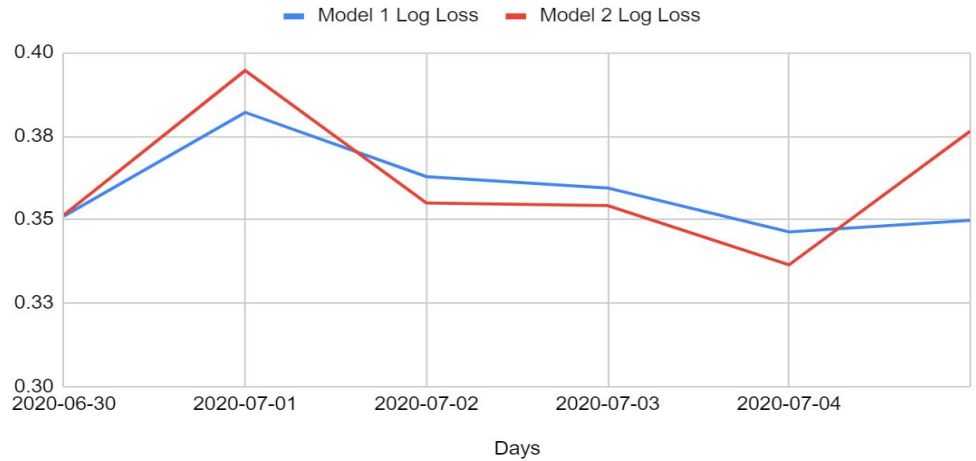
Handle Categorical Data with Deep Learning

- One Hot Encoding
  - Sparse way of representing
- Target Encoding
  - Replacing with the mean of the target
- Embeddings
  - Convert discrete values to a vector of continuous values

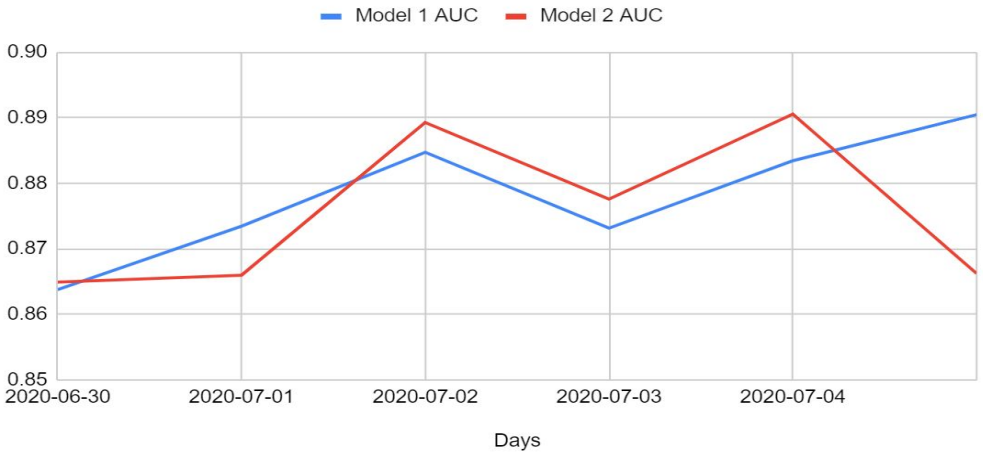
puppy	[0.9, 1.0, 0.0]
dog	[1.0, 0.2, 0.0]
kitten	[0.0, 1.0, 0.9]
cat	[0.0, 0.2, 1.0]



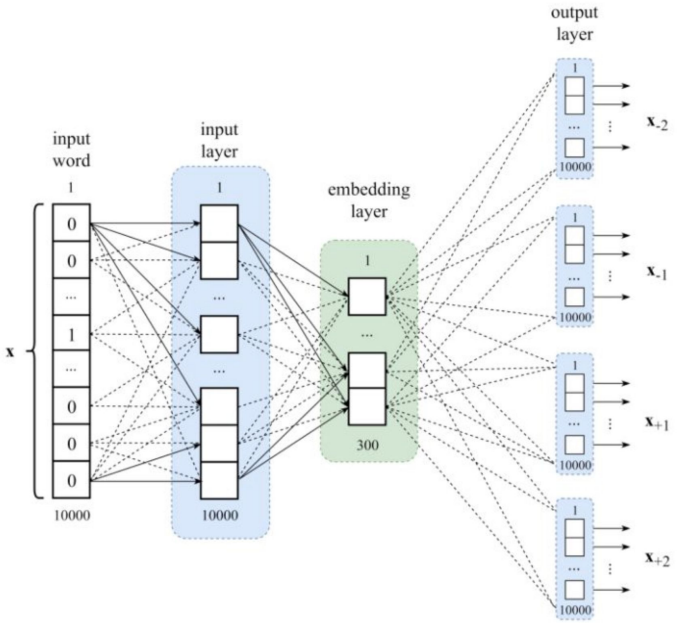
Model 1 Log Loss vs Model 2 Log Loss



Model 1 AUC and vs Model 2 AUC



Embedding Layer



Moving all experiments to Google Cloud Platform

GCP

- Allocate resource as much as needed
- Flexible
- Easy to use

Google AI Platform

- Run ML code on top of TensorFlow, Scikit-learn and XGBoost
- Build ML pipelines

BigQuery

- Serverless, highly scalable, and cost-effective cloud data warehouse

smart

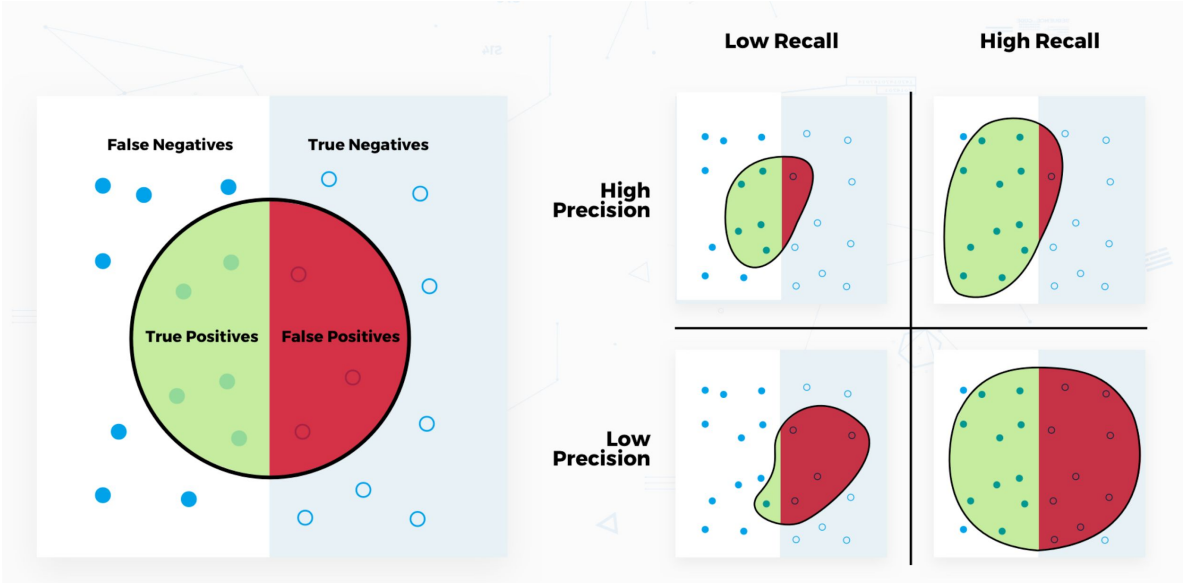
Switching Prediction Target from Success to Failure

Before

Predicting Success

After

Predicting Failure



ITX4

12 features;

IsHeaderBidding;HeaderBiddingTypeId;NetworkId;

SiteId;PlatformId;InsertionScriptId;ImpressionType;

Inventory;PartnerId;CreativeHashcode;

Referrer;InputBundleId

ITX5

19 features;

IsHeaderBidding;HeaderBiddingTypeId;NetworkId;

SiteId;PlatformId;InsertionScriptId;ImpressionType;

Inventory;PartnerId;CreativeHashcode;Referrer;

InputBundleId;InsertionId;FormatId;SmartBuyerId;

VideoPlacementType;PageId;OsParentId;

BrowserParentId

Perfumation feature importance

CreativeHashcode	0.083310
PageId	0.041163
Referrer	0.025063
SmartBuyerId	0.023372
Inventory	0.019067
InsertionId	0.018772
PlatformId	0.018174
NetworkId	0.016077
BrowserParentId	0.011999
OsParentId	0.011998
PartnerId	0.011587
SiteId	0.007714
InputBundleId	0.005112
InsertionScriptId	0.002813
IsHeaderBidding	0.002100
VideoPlacementType	0.001938
FormatId	0.001512
HeaderBiddingTypeId	0.001163

Gain feature importance

PageId	0.159561
CreativeHashcode	0.118734
Referrer	0.098011
SiteId	0.091464
VideoPlacementType	0.083967
NetworkId	0.063915
FormatId	0.055364
SmartBuyerId	0.048019
IsHeaderBidding	0.039537
InputBundleId	0.038564
PartnerId	0.037207
Inventory	0.030380
HeaderBiddingTypeId	0.027295
ImpressionType	0.027182
InsertionId	0.024226
InsertionScriptId	0.021150
PlatformId	0.012486
BrowserParentId	0.012018
OsParentId	0.010920

Target Encoding

is applying additive smoothing. The intuition behind this approach is smoothing the

computed mean by the global mean. For instance, in our case, the target is the delivery

rate, and we are interested in encoding the networkId column. Assuming that there is a

new networkId that is contained by only three samples in the data set and it gives 0.9

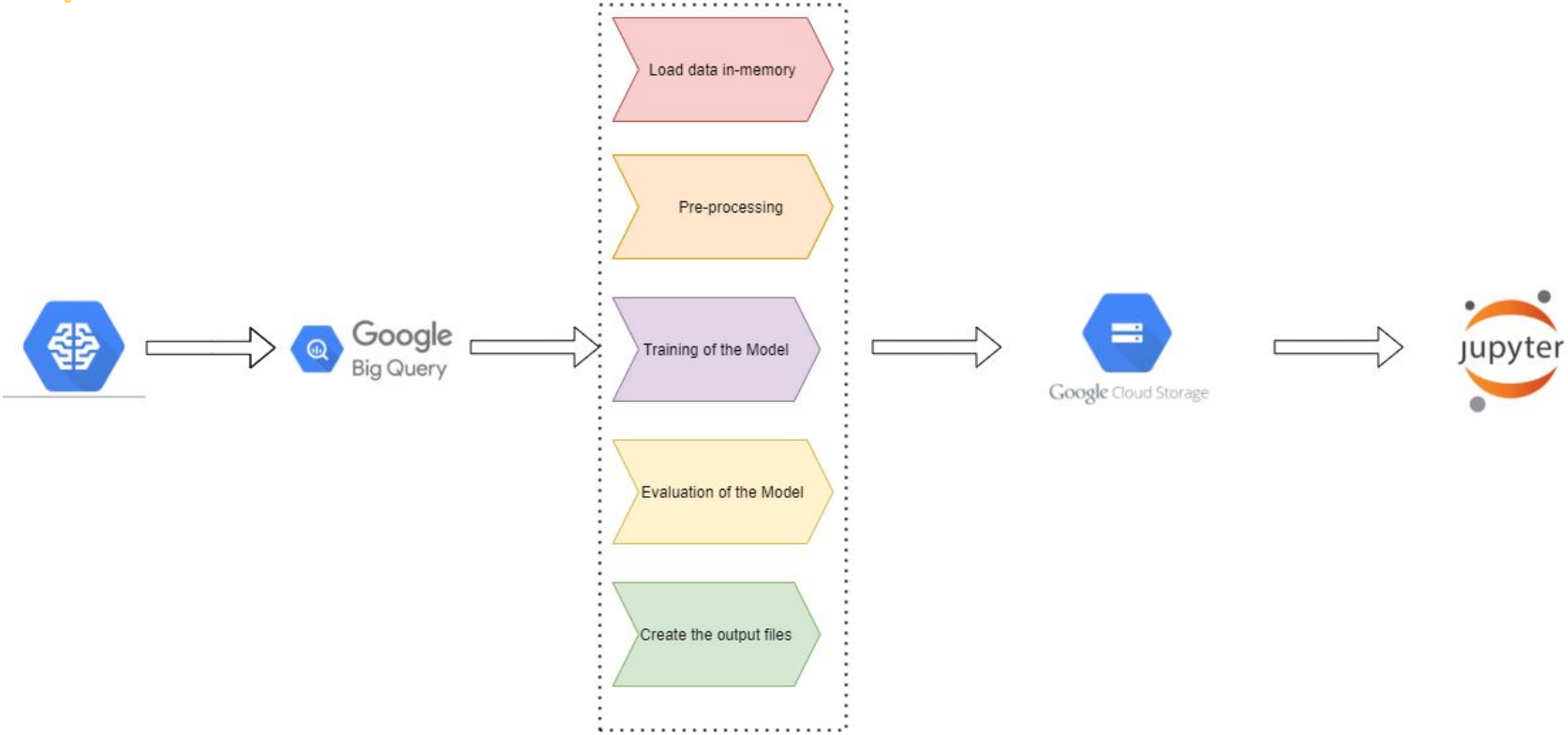
of target mean for the new networkId whereas the target mean of all networkIds hover

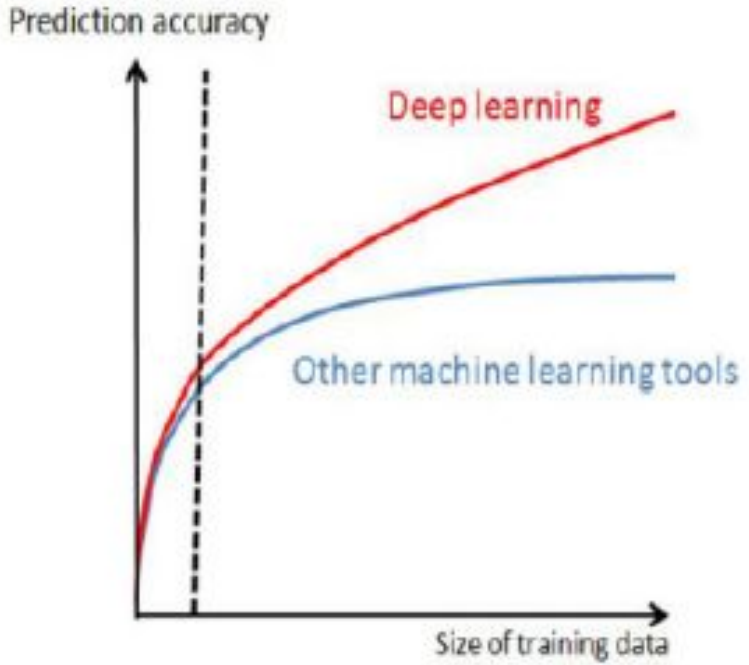
around 0.3. Therefore, it is better not to rely on only three values and to smooth the

average by including the average delivery rate over all the networkIds.

Embedding Layer

it limits the number of columns per category, and it is able to capture similarities between similar variables. It performs well in many benchmarks. However, it requires more tuning compared to the previous methods therefore it takes more time to implement, as explained in the next paragraph.


Load data in-memory
Pre-processing
Training of the Model
Evaluation of the Model
Create the output files
Google
Big Query
Google Cloud Storage
jupyter


Prediction accuracy
Deep learning
Other machine learning tools
Size of training data

XGBoost scalable and parallelizable

- Split Finding Algorithms(Basic Exact Greedy Algorithm)
- Approximate Algorithm:
  XGBoost uses approximation on the split points
  - Quantile approach
    - Get the same accuracy as the exact greedy algorithm
    - Benefit of being distributable
- Sparsity Aware Split Finding
- Column Block for Parallel Learning
  - Block structure helps to optimize the computation complexity of split finding
  - Sort the data by column, in a compressed format
  - Different blocks can be distributed across machines
- Cache-Aware Access(non-continuous)

TensorFlow Serving in Production

- Can serve multiple models, or multiple versions of the same model simultaneously
- Exposes both gRPC as well as HTTP/2 inference endpoints
- Allows deployment of new model versions without changing any client code
- Supports many *servables*: Tensorflow models, embeddings, vocabularies, feature transformations and even non-Tensorflow-based machine learning models