Cómo leer esta presentación:

- **Cliente**: slides con contenido enfocado a negocio
- **Técnico**: slides con modelos, métricas, validaciones,…

Dependiendo de la audiencia saltarse las diapositivas rojas;
a menos que se quiera profundizar en el tema.

Uplift modeling for marketing optimization

End-to-end proyect from data exploration to business decision-making

Iván Martínez Ibarburu

11/02/2026

https://github.com/IvanMIbarburu/Uplift_marketing_proyect

Business problem:

- A company creates an ad for a campaign and show it to a big population of potential users (treatment).
- The company expects an increase of the conversion rate after the treatment.
- However… the campaign costs are non-negligible; and not all users are sensible to the ad… or even the product! Even more, there may be users that use to buy the product without the need of the campaign.

Business problem:

- What if there is a way to estimate who of the users increase the conversion rate **BECAUSE** of the ad?

- If could use this “magic” the company would “guess” who to target. And eliminate the costs of targeting those who are not interested in the product, who are insensible to the ad, who are already customers, and focus only on the real potential users.

- This solution translates in one goal: maximize the profit by reducing innecesary costs.

Dataset (experiment)

- Criteo Uplift Prediction Dataset from Kaggle: [https://www.kaggle.com/datasets/arashnic/uplift-modeling]
- Almost 14 million entries
- Randomized experiment
- Treatment and control groups (85% - 15%)
- Highly imbalanced outcome (only 0.29% conversion rate)

Dataset (features)

- f0, f1, f2,… f11: feature values (describe the user's profile)
- treatment: describes if the user belongs to the treatment (1) or the control (0) group
- visit: describes if the user visit (1) or not (0) the website
- conversion: describes if the user converts (1) or not (0)
- exposure: describes if the user was really exposed (1) or not (0)

EDA (sanity checks)

- Total length of the dataset: 13.979.592
- Total length of control group: 2.096.937 (15% is correct)

- Conversion rate of control group: 0.1938%
- Conversion rate of treat group: 0.3089%

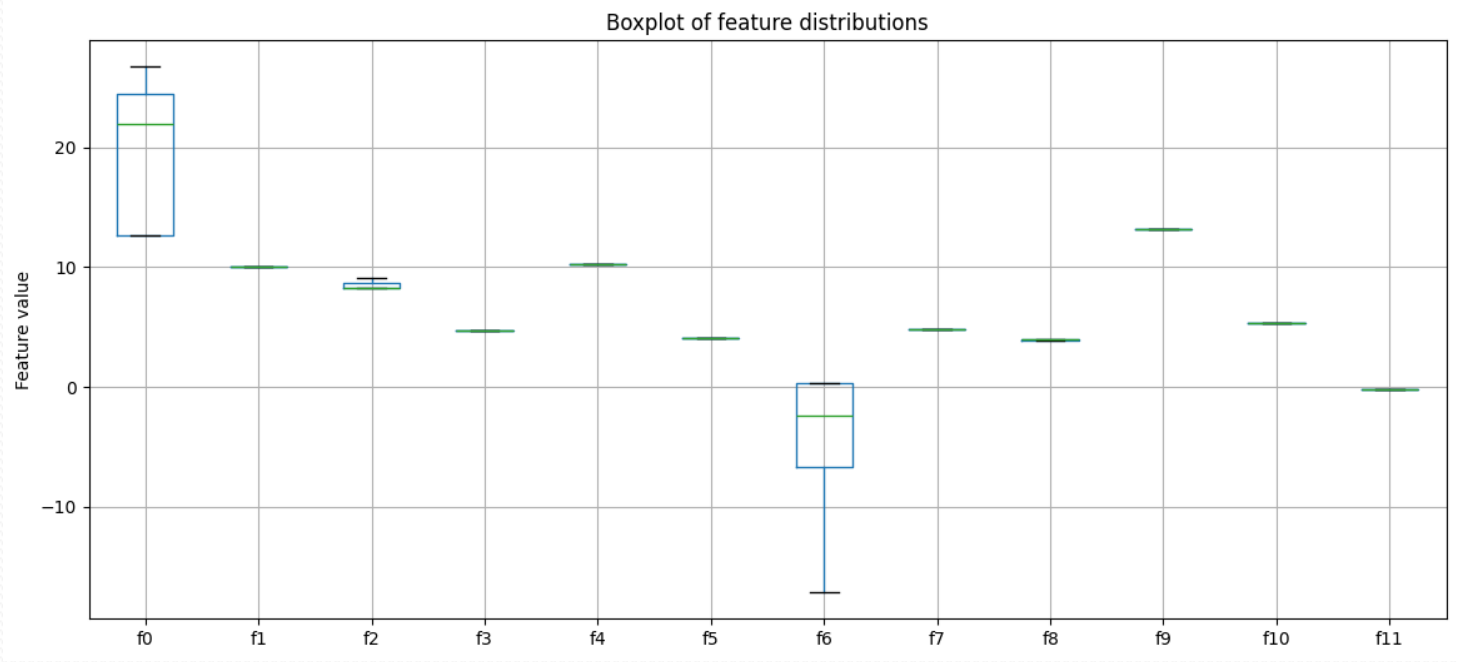
First conclusion: to increase the conversion rate you can simply treat everybody!

- Ratio of conversion: 1.5939
- Average treatment effect (absolute): 0.1152%

- Visit rate of control group: 3.8201%
- Visit rate of treat group: 4.8543%

Entries with no sense: (None)
- Treatment=0 & Exposure=1: 0
- Visit=0 & Conversion=1: 0

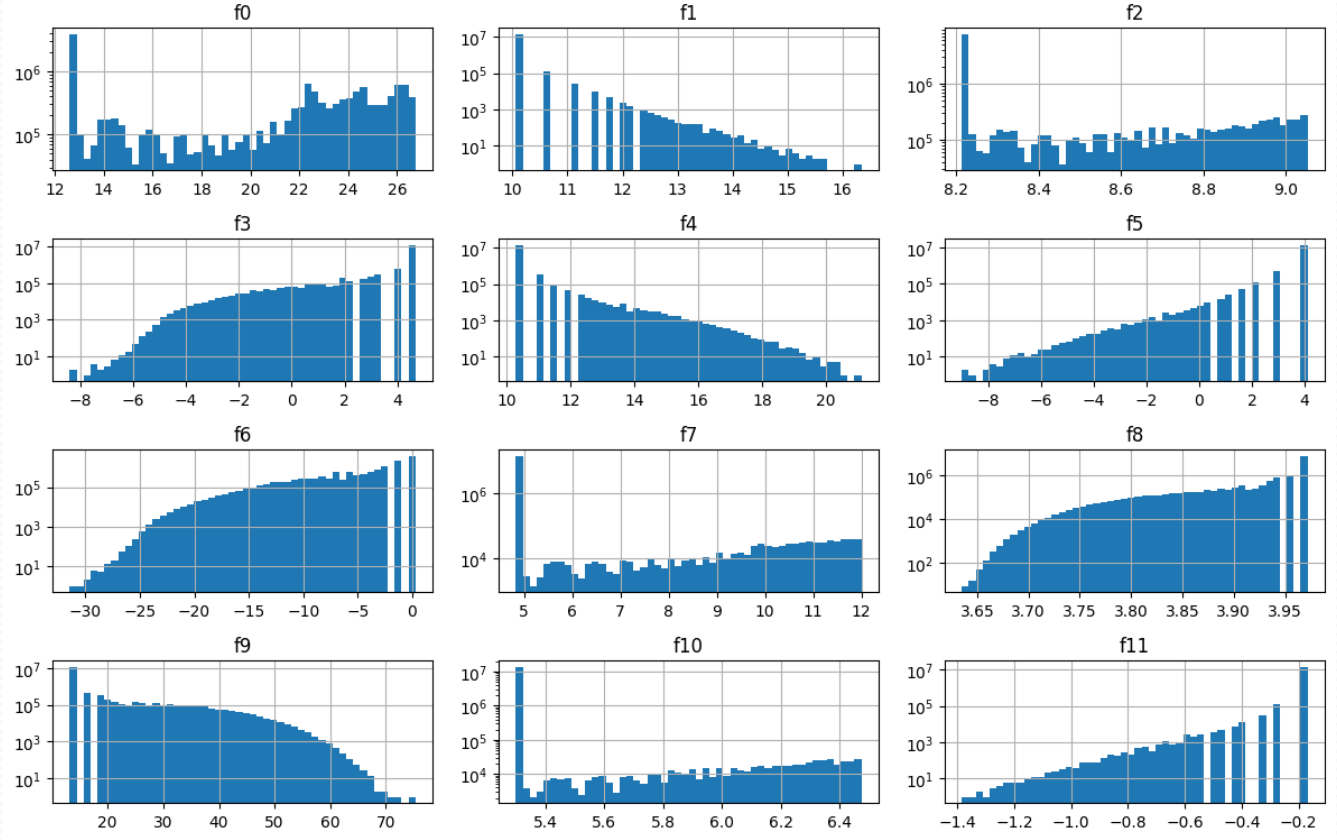
EDA (feature values)



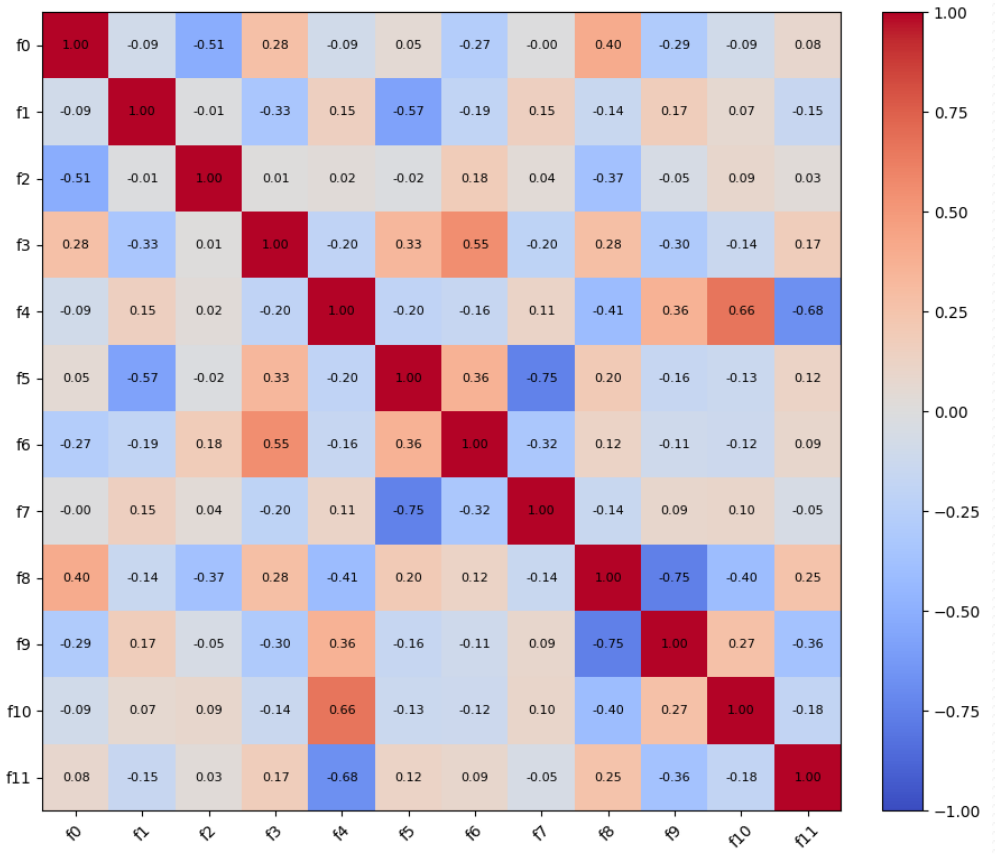
This looks like the result of an encoder…

The outliers are not

Log-distribution of feature values



Correlation matrix



High correlation between some pair of features:

- f5 and f7
- f4 and f11
- f4 and f10
- …

Most of them are moderate.

EDA conclusions:

- The features have heterogeneous ranges and heavy-tailed distributions.
- Several features show a high concentration of values around specific points
- Correlations between features are generally moderate, suggesting some redundancy but no severe multicollinearity issues.

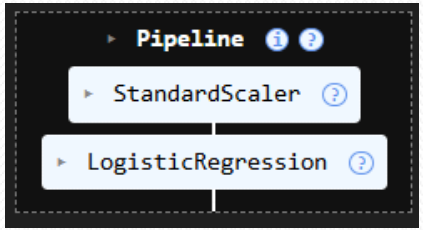
- The feature set appears suitable for modeling using regularized linear models as a first baseline, without the need for feature transformations.

Baseline approach:

- Using the control group (T= 0) let's try to estimate the probability of conversion (Y = 1) given only the feature values (X):

$$P(Y = 1 \mid T = 0, X)$$

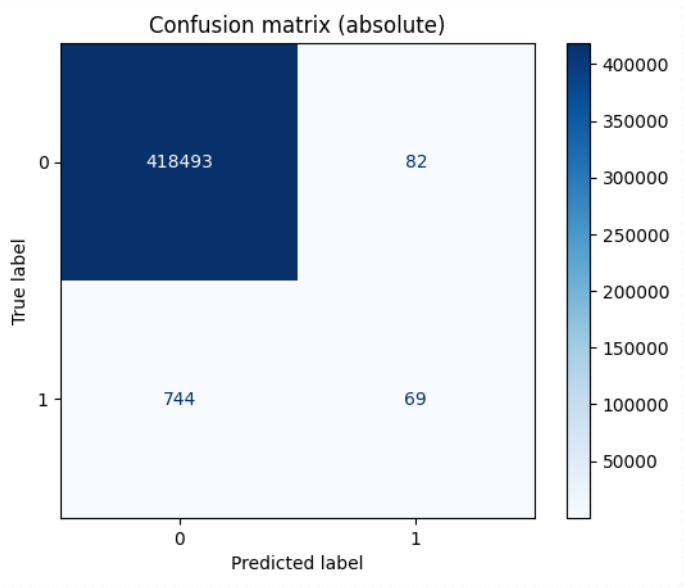
- Let's use a simple model, Logistic Regression:

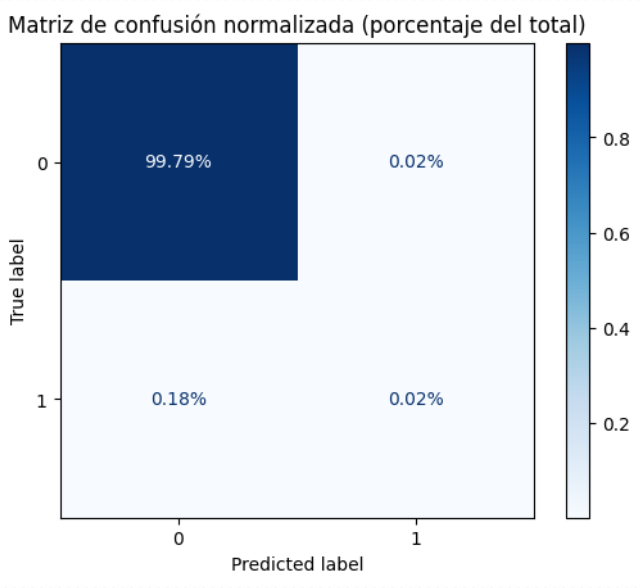


$$p = \frac{1}{1 + e^{f(x)}}$$

$$f(x) = \sum_{i=0}^{k} \beta_i x_i$$

- The logistic model output is the probability of conversion, given a threshold of 0.5:





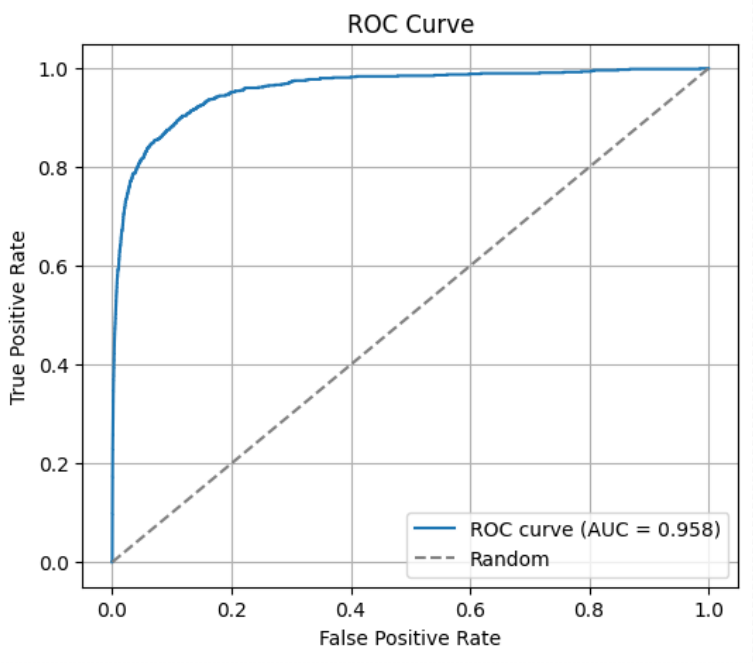
- Accuracy: 0.9980
- Recall: 0.0849

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

GREAT! I have a model that predicts worse than just say “everything is 0”…

ROC-AUC



AUC: 0.9584

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

What really matters for the uplifting model is not “a good prediction” of the conversion; but “**a good sorting**” of the probability of conversion.

That is: we are interested in a good AUC score (the higher the better) because that mean that the probability of converters are higher than no-converters.

The goal is a good probability ranking

Baseline conclusions:

- A high ROC-AUC indicates strong ranking power, not necessarily good classification.
- In highly imbalanced conversion problems, it is common to have excellent discrimination without a meaningful decision threshold
- High accuracy only implies that the model is pretty good predicting true negatives
- This baseline does not distinguish between users who benefit from treatment and those who do not

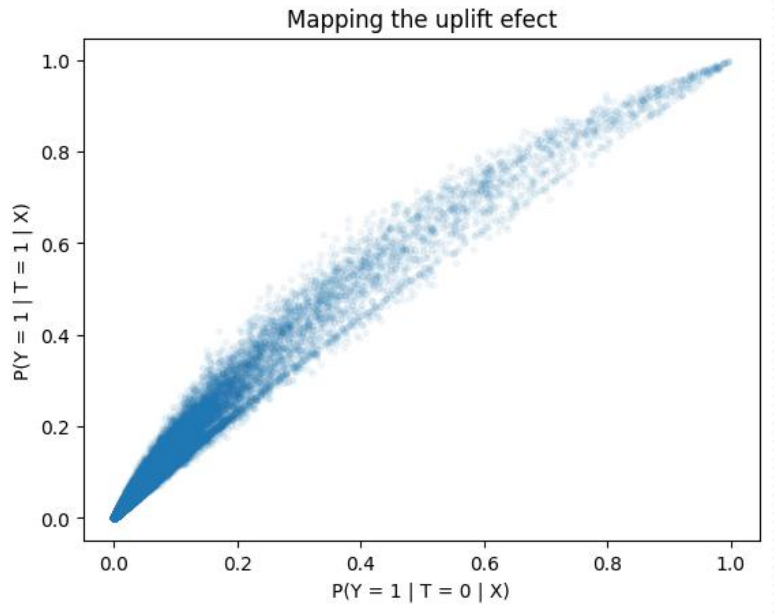
Two model approach (T-learner)

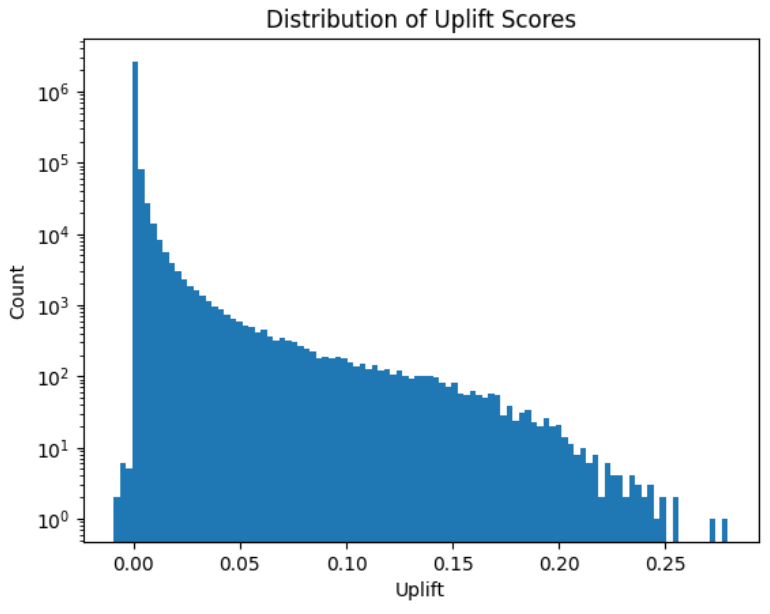
- We are really interested in the **increase of probability of conversion** due to the treatment:

$$\text{Uplift}(X) = P(Y=1 \mid T=1, X) - P(Y=1 \mid T=0, X)$$

Uplift	**High P(Y=1 | T=1, X)**	**Low P(Y=1 | T=1, X)**
High P(Y=1 | T=0, X)	Always convert (waste of resources)	Negative responders (never treat them!)
Low P(Y=1 | T=0, X)	Persuadable (main target)	Never convert (waste of resources)

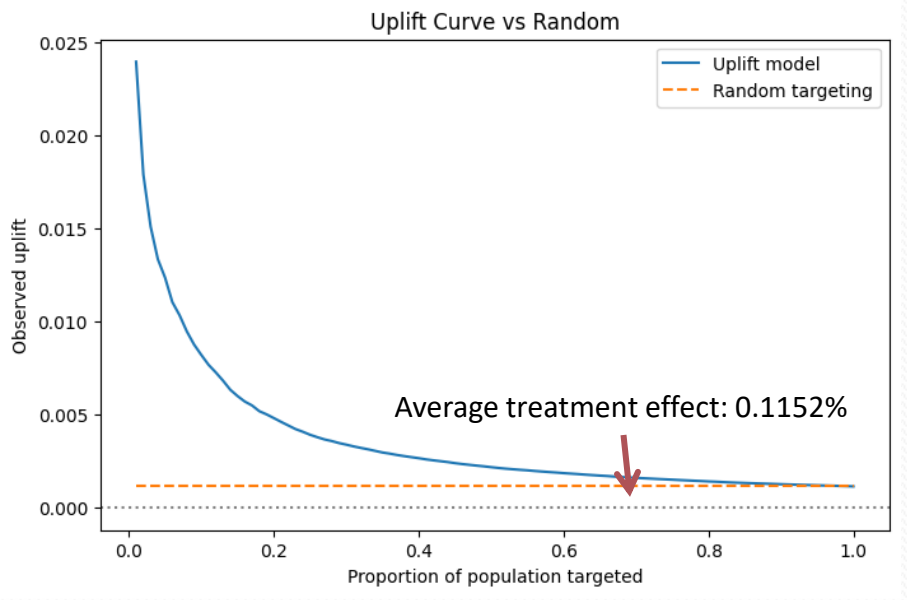
- Create a second model for the treat group: $P(Y = 1 \mid T = 1, X)$
- Using the same pipeline (standarization + logistic regression)
- Train both models with their respectives groups.
- Calculate for the dataset (test) both probabilities of conversion
- Calculate the “uplift score” (the difference between them):
  - Uplift > 0 means the user may be persuadable
  - Uplift = 0 (aprox) means the user may be insensible
  - Uplift < 0 means the user may be a negative responder





This method predicts a lot of insensible users, some persuadable users, and just a few negative responders.

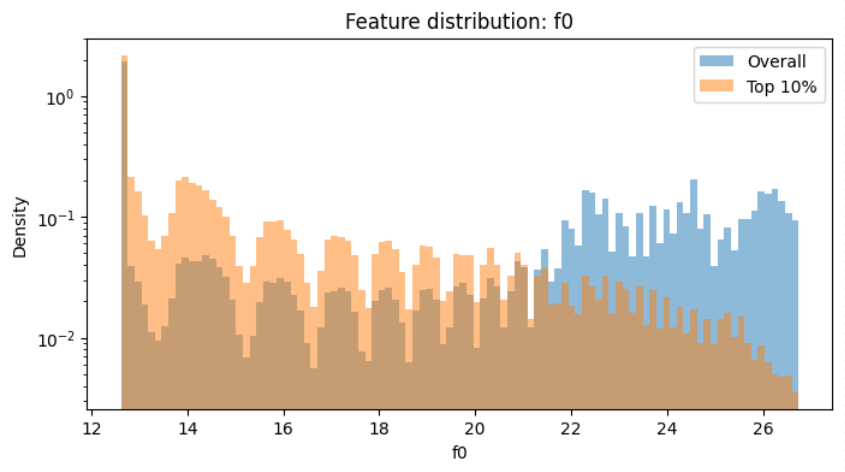
- The main objective of this method is to sort the users by their uplift score.
- Sort the dataframe by their calculated uplift score
- Select different percentiles to measure the real uplift given this sorting (i.e. using the real conversion rate of both groups of the dataset)
- The graph shows that, even with a very simple model, there is a good sorting of the potential uplift and so the targeting of specific users.

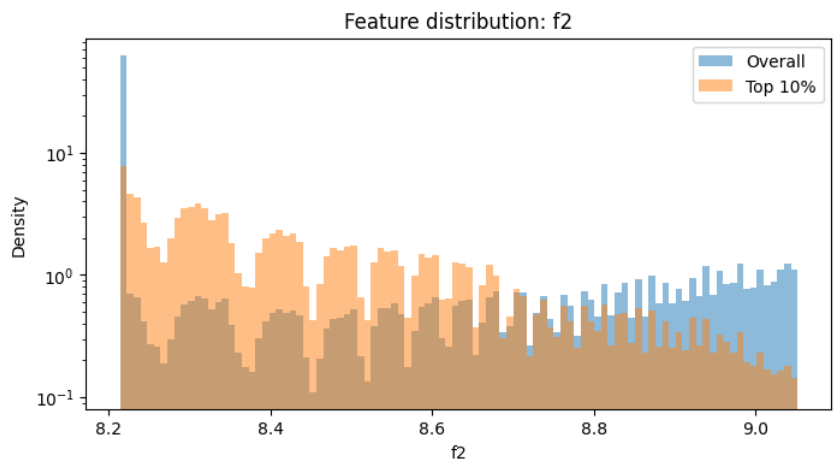


T-learner conclusions

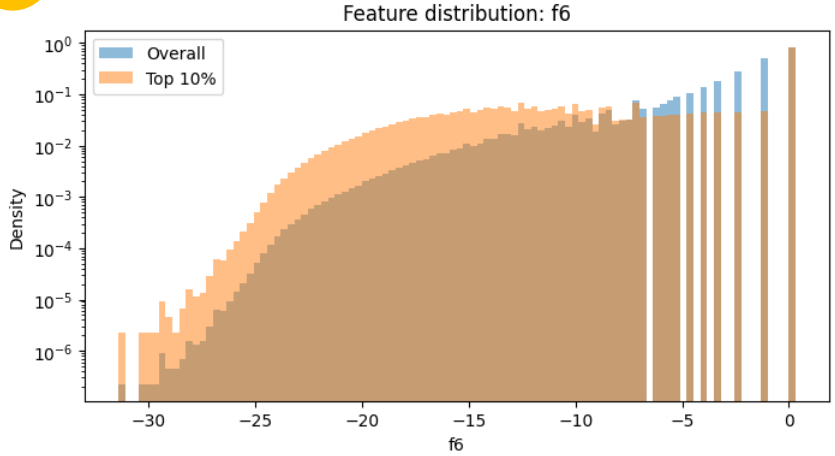
- The top-ranked percentiles deliver substantially higher observed uplift
- The model is able to concentrate **incremental** conversions in a small fraction of users
- Random targeting provides a flat baseline uplift

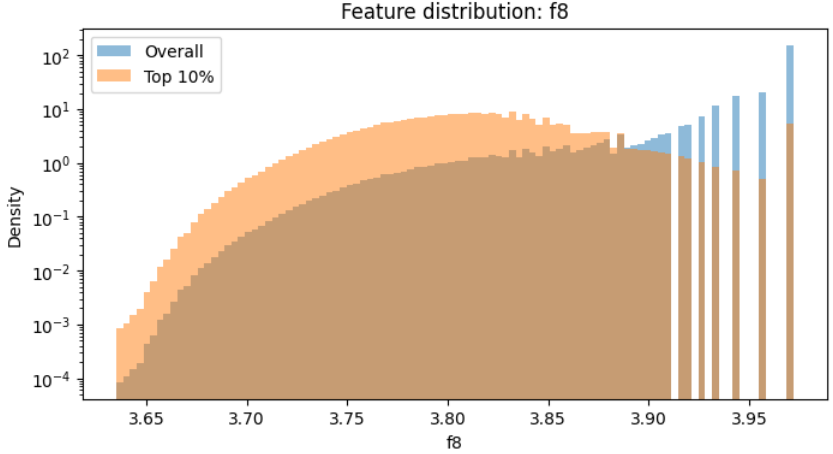
Profile of top 10%

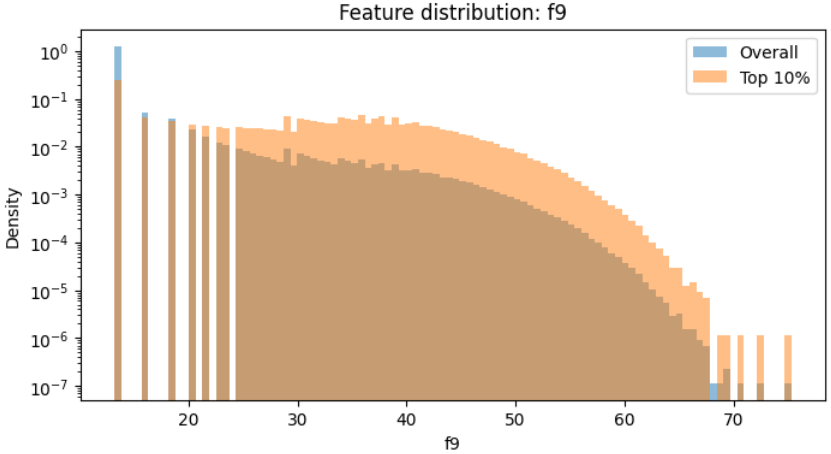


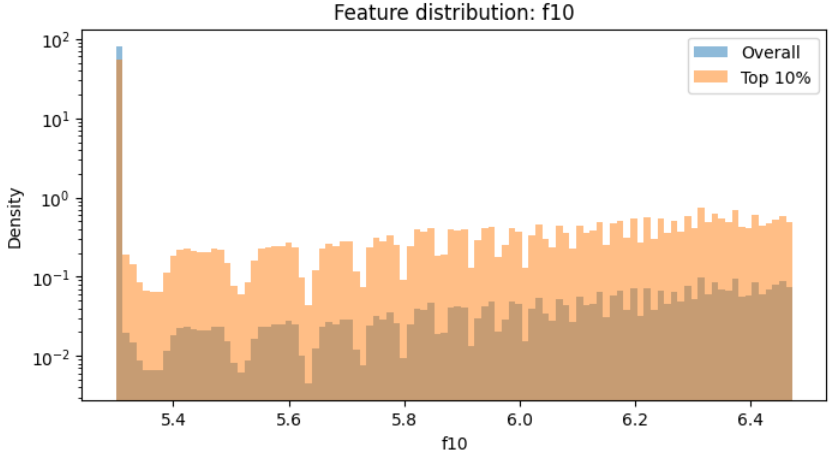


If we could use the decoder to get the real meaning of this profile…


Feature distribution: f6
Overall
Top 10%
Density
f6


Feature distribution: f8
Overall
Top 10%
Density
f8


Feature distribution: f9
Overall
Top 10%
Density
f9


Feature distribution: f10
Overall
Top 10%
Density
f10

Demo: “what-if”

- Using only 10% of the dataset (due to size limits)

- Given the costs per user and the return of conversion: what is the best percentile to target?

$$Maximize:\ \ k\ (Uplift(k) * R\ -\ \ C)$$

$$with:\ k = percentil * Population$$

$$R \equiv return\ of\ conversion$$

$$C \equiv costs\ per\ user$$

https://upliftmarketingproyect-x4xqd2kprx5pzpcs8j3mvm.streamlit.app/

Business Parameters

CPM (€)

3,00

Impressions per user

10

Revenue per conversion (€)

40,00

Population size

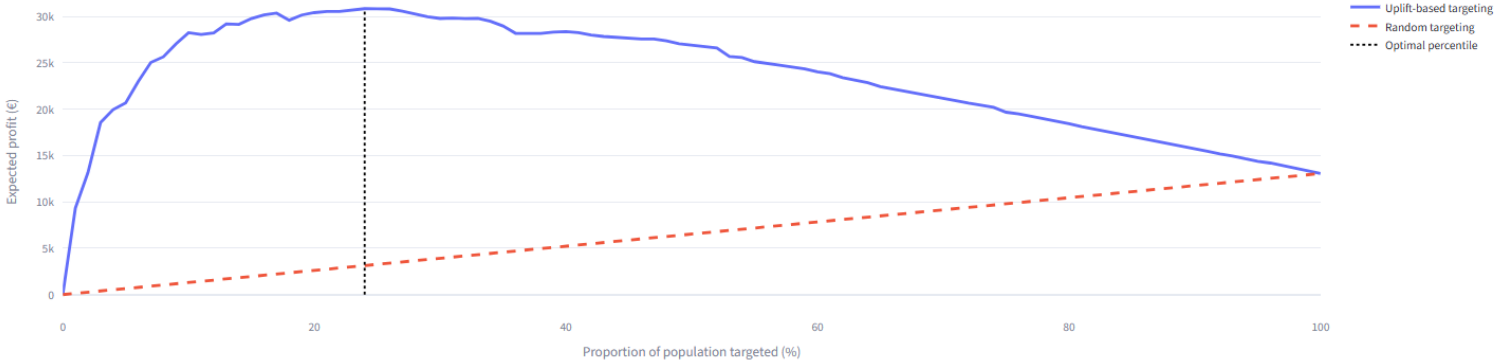
1000000

Uplift Marketing – Profit Optimization Demo

This demo shows how uplift modeling can be used to **optimize targeting decisions** by balancing treatment cost and conversion value.

Baseline (no treatment): 81,697 € (shown as y = 0 in the chart)

Profit vs Targeted Population



Optimal Strategy

Optimal percentile	Profit (uplift)	Profit (random)
24.0%	30,832 €	3,134 €

Conversion rate of control group: 0.2042%

Conversion rate of treat group: 0.3119%

Second iteration

- Try to get real meaning from the features values (for the top10% users)
- Try again the T-learner approach but with different models (XGBoost)
- Try new approaches:
  - S-learner (using the same Logistic Regression with features AND treatment)
  - Class Transformation (predicts Z, a new class defining if the user is a good candidate)
- Add budget constrains into the “what-if” demo
- Add heterogeneous costs? Heterogeneus return of converion?
- Multi-treatment with more than one ad?
- …

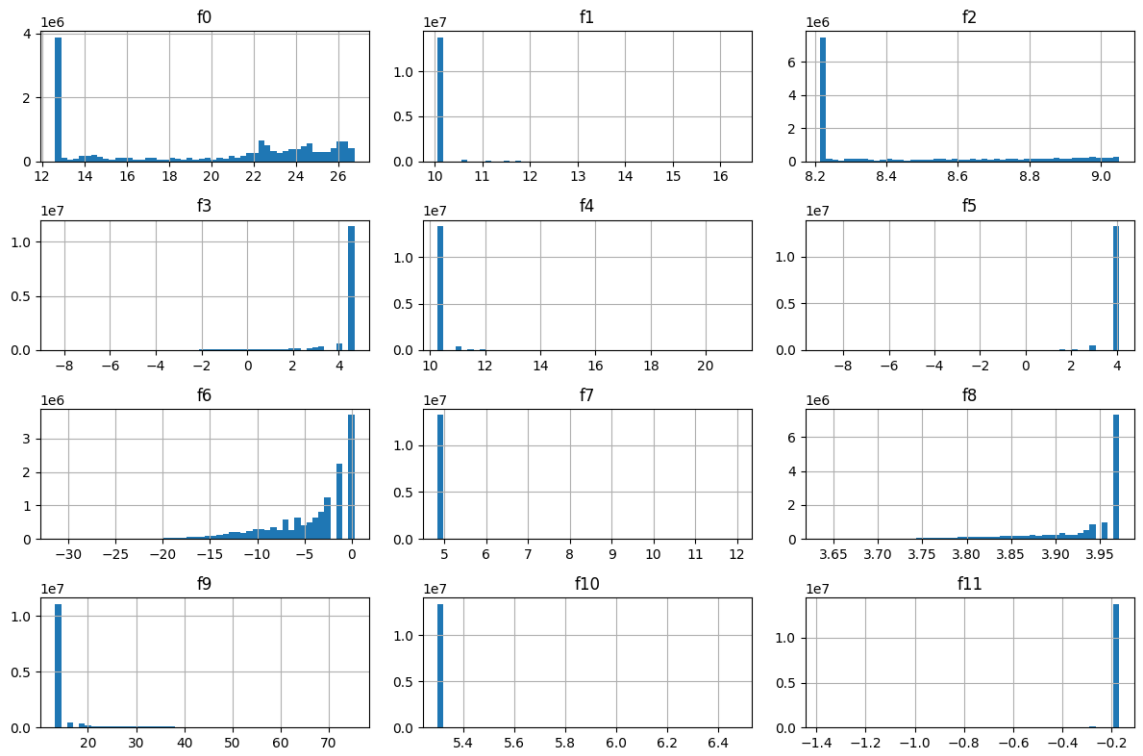
Thanks for your attention

Appendix

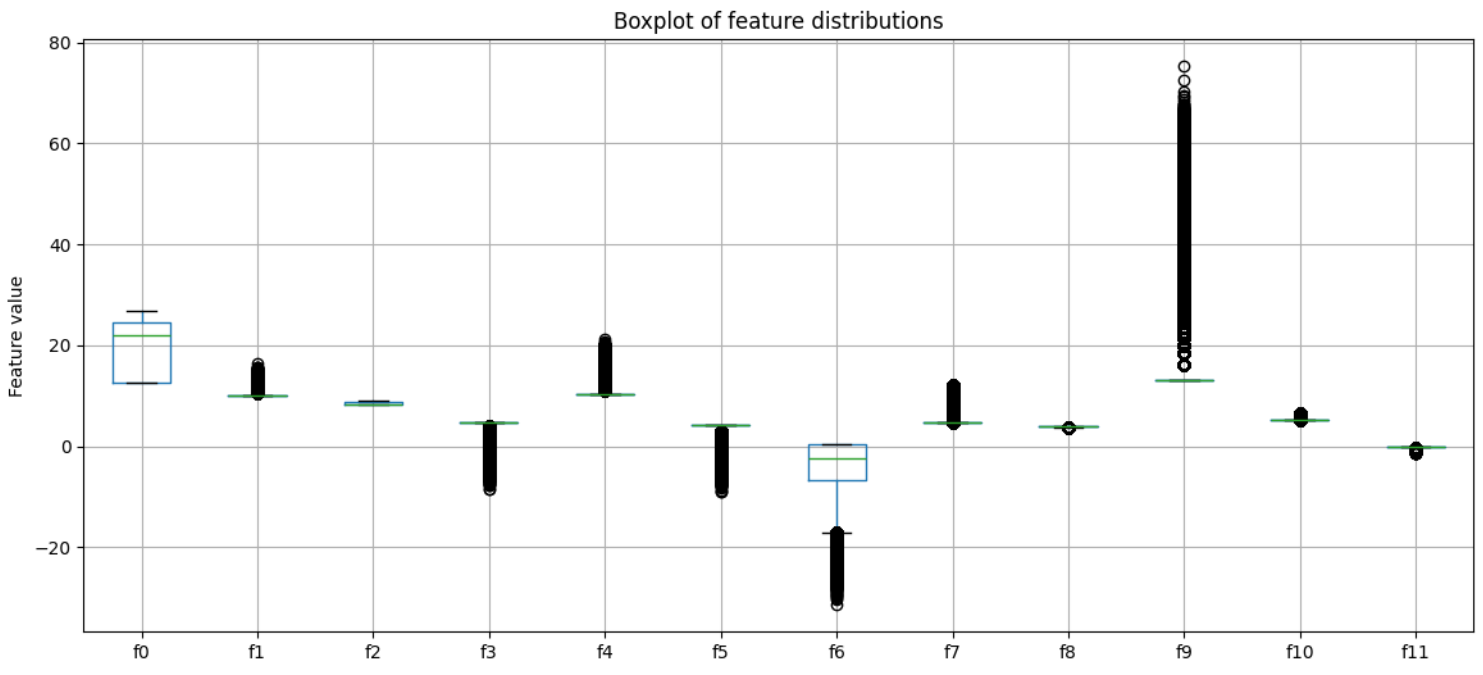
Dataset description

	count	mean	std	min	25%	50%	75%	max
f0	13979592.0	19.620297	5.377464	12.616365	12.616365	21.923413	24.436459	26.745255
f1	13979592.0	10.069977	0.104756	10.059654	10.059654	10.059654	10.059654	16.344187
f2	13979592.0	8.446582	0.299316	8.214383	8.214383	8.214383	8.723335	9.051962
f3	13979592.0	4.178923	1.336645	-8.398387	4.679882	4.679882	4.679882	4.679882
f4	13979592.0	10.338837	0.343308	10.280525	10.280525	10.280525	10.280525	21.123508
f5	13979592.0	4.028513	0.431097	-9.011892	4.115453	4.115453	4.115453	4.115453
f6	13979592.0	-4.155356	4.577914	-31.429784	-6.699321	-2.411115	0.294443	0.294443
f7	13979592.0	5.101765	1.205248	4.833815	4.833815	4.833815	4.833815	11.998401
f8	13979592.0	3.933581	0.056660	3.635107	3.910792	3.971858	3.971858	3.971858
f9	13979592.0	16.027638	7.018975	13.190056	13.190056	13.190056	13.190056	75.295017
f10	13979592.0	5.333396	0.168229	5.300375	5.300375	5.300375	5.300375	6.473917
f11	13979592.0	-0.170967	0.022833	-1.383941	-0.168679	-0.168679	-0.168679	-0.168679
treatment	13979592.0	0.850000	0.357071	0.000000	1.000000	1.000000	1.000000	1.000000
conversion	13979592.0	0.002917	0.053927	0.000000	0.000000	0.000000	0.000000	1.000000
visit	13979592.0	0.046992	0.211622	0.000000	0.000000	0.000000	0.000000	1.000000
exposure	13979592.0	0.030631	0.172316	0.000000	0.000000	0.000000	0.000000	1.000000

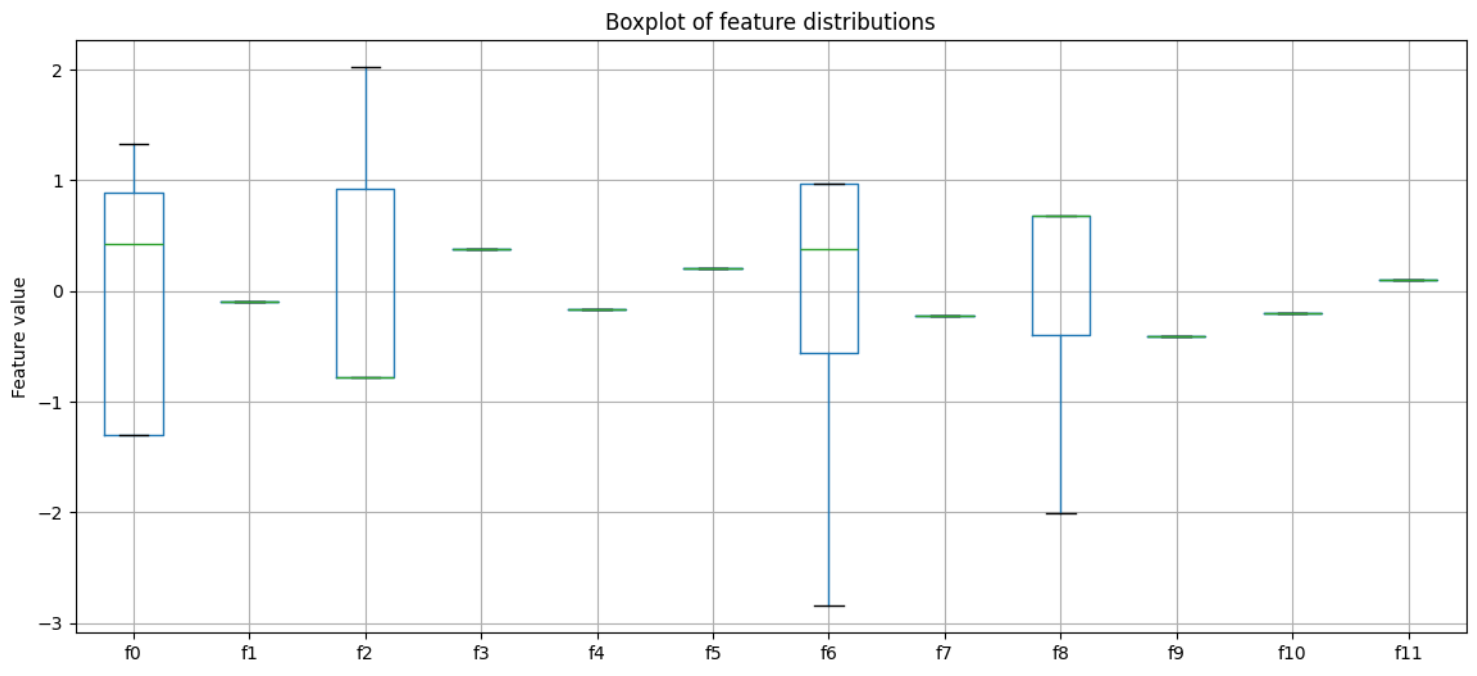
Original histograms of features



Feature boxplot complete



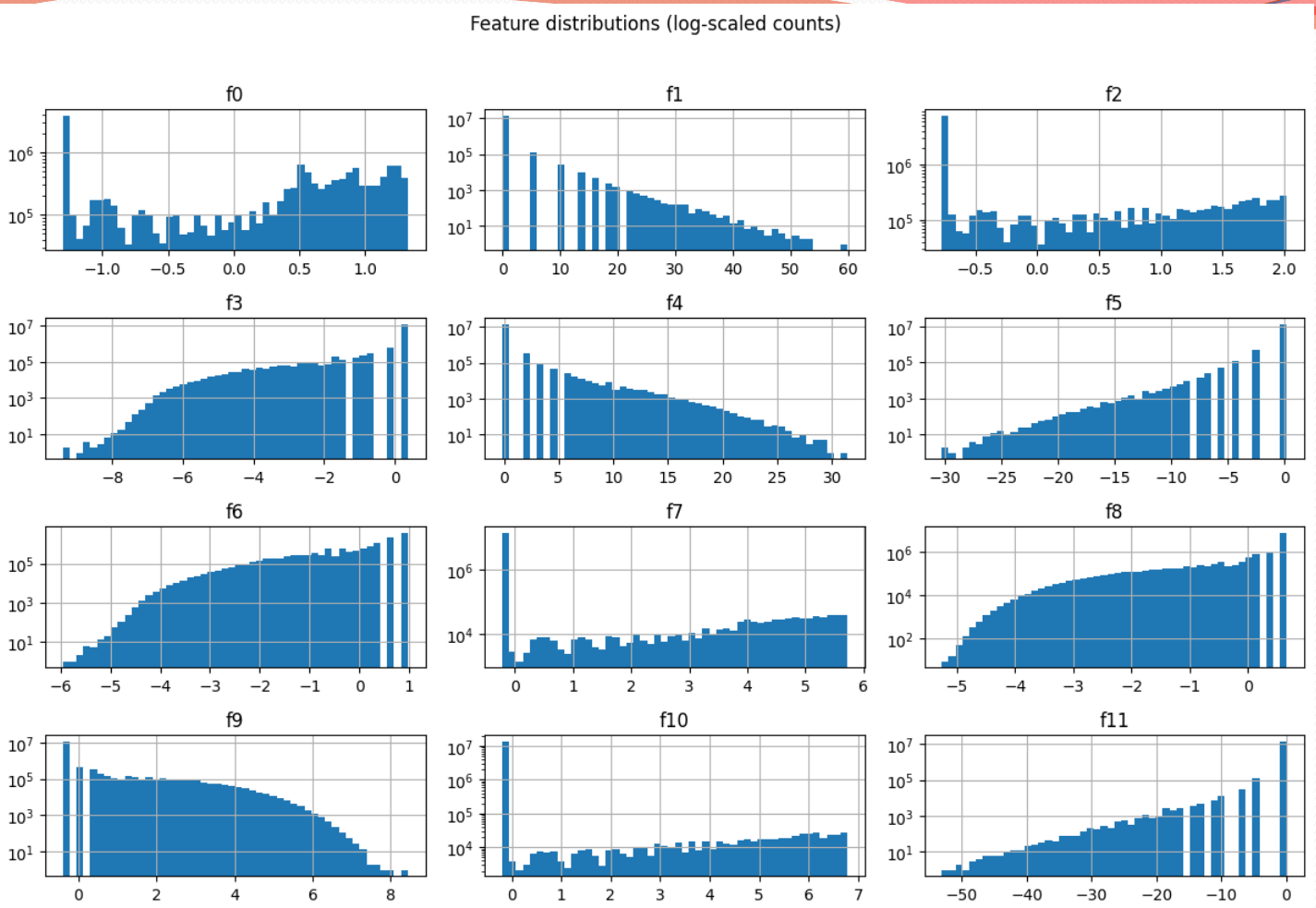
Standarized features



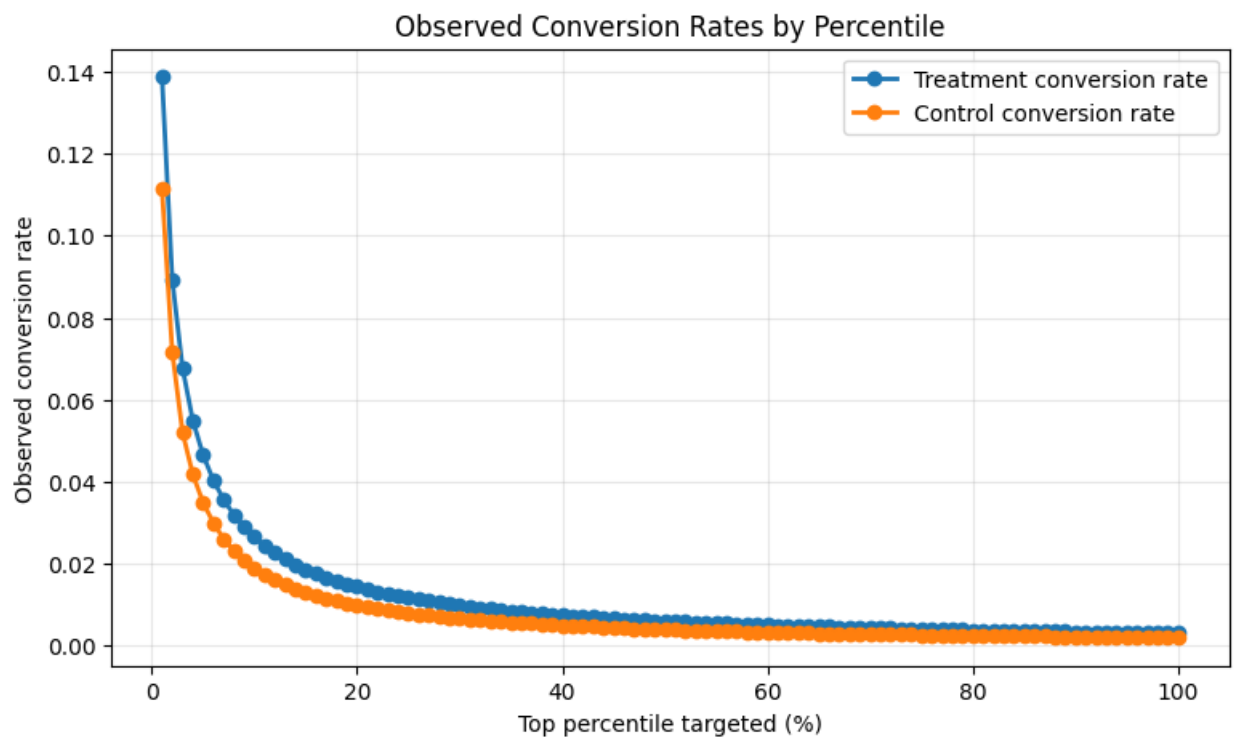
Without plotting the outliers

Standarized features

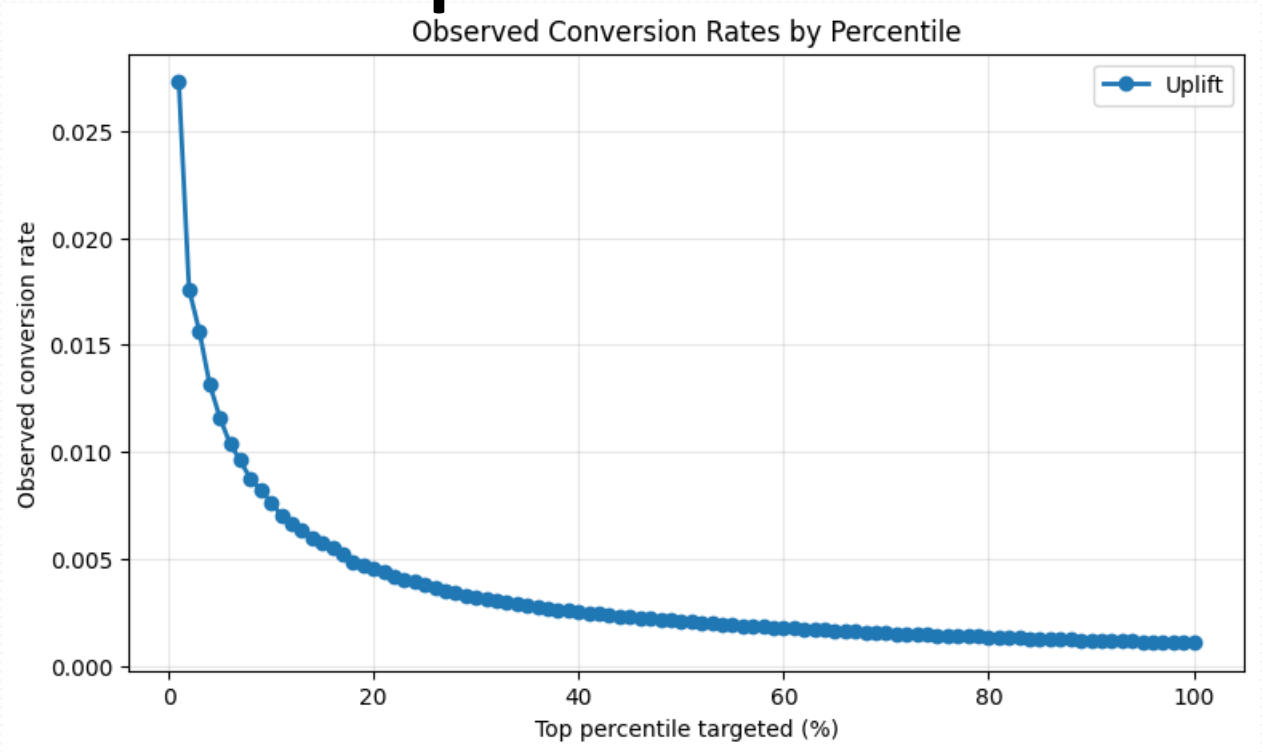
	0.01	0.25	0.50	0.75	0.99
f0	-1.302460	-1.302460	0.428290	0.895619	1.311507
f1	-0.098534	-0.098534	-0.098534	-0.098534	5.818642
f2	-0.775767	-0.775767	-0.775767	0.924618	1.992321
f3	-4.423128	0.374788	0.374788	0.374788	0.374788
f4	-0.169853	-0.169853	-0.169853	-0.169853	4.760884
f5	-4.169837	0.201673	0.201673	0.201673	0.201673
f6	-2.975573	-0.555704	0.381012	0.972014	0.972014
f7	-0.222320	-0.222320	-0.222320	-0.222320	5.293782
f8	-3.211772	-0.402207	0.675564	0.675564	0.675564
f9	-0.404273	-0.404273	-0.404273	-0.404273	4.112567
f10	-0.196284	-0.196284	-0.196284	-0.196284	5.967292
f11	-4.221258	0.100208	0.100208	0.100208	0.100208


Feature distributions (log-scaled counts)
f0
f1
f2
f3
f4
f5
f6
f7
f8
f9
f10
f11

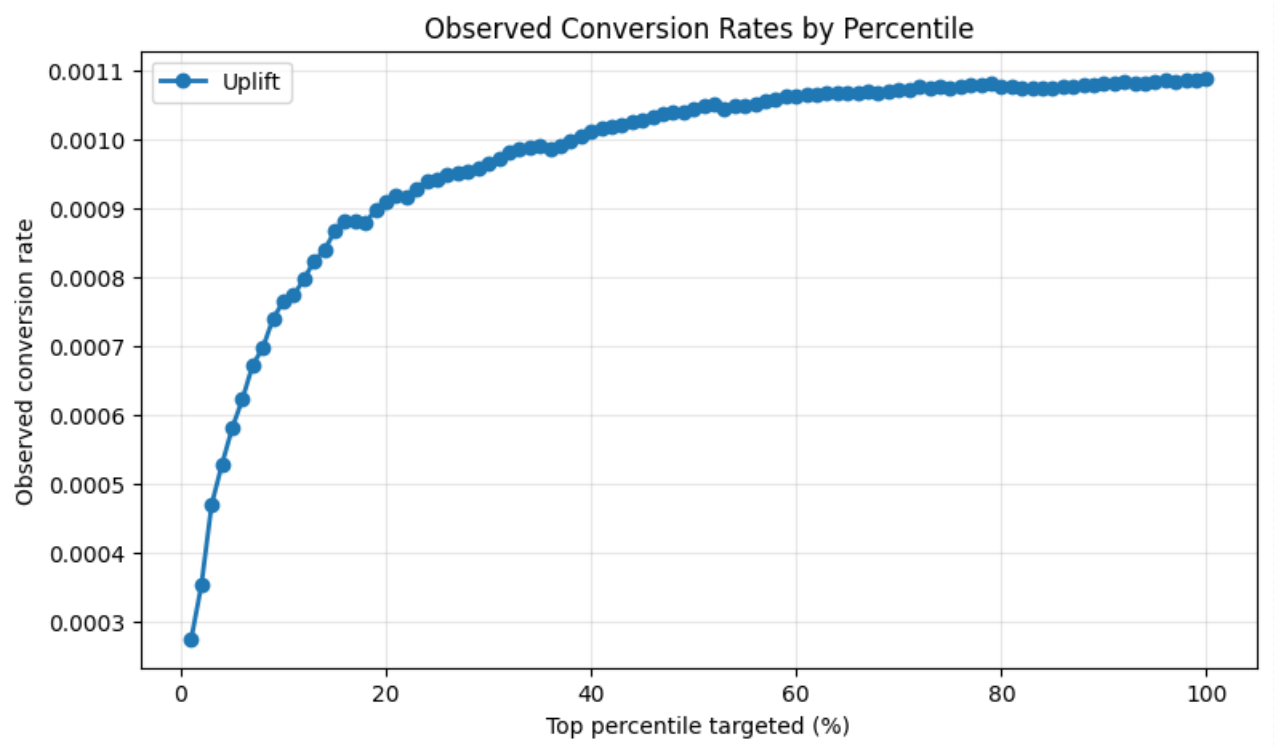
T-learner models



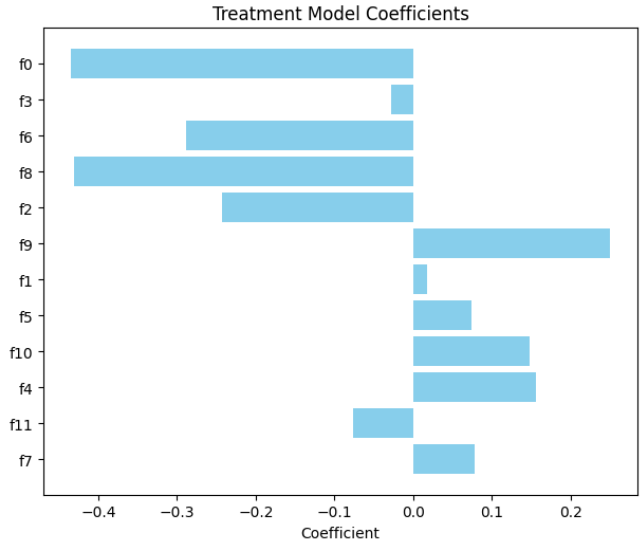
T-learner uplift

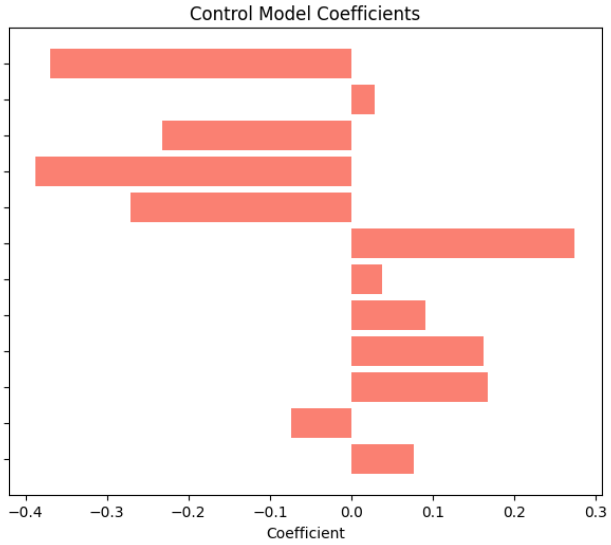


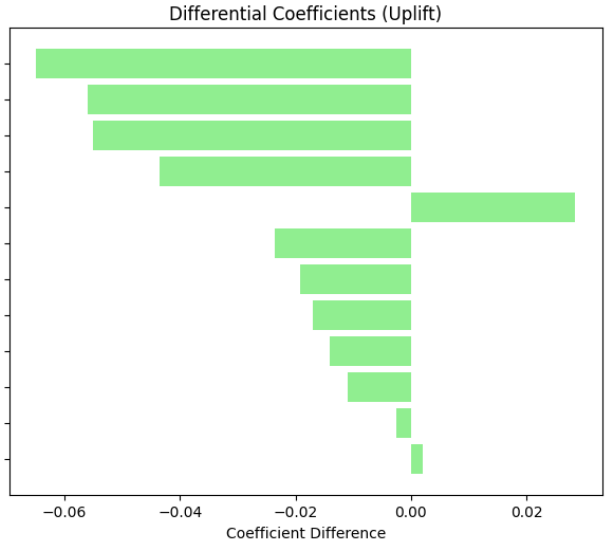
Uplift * percentiles effect



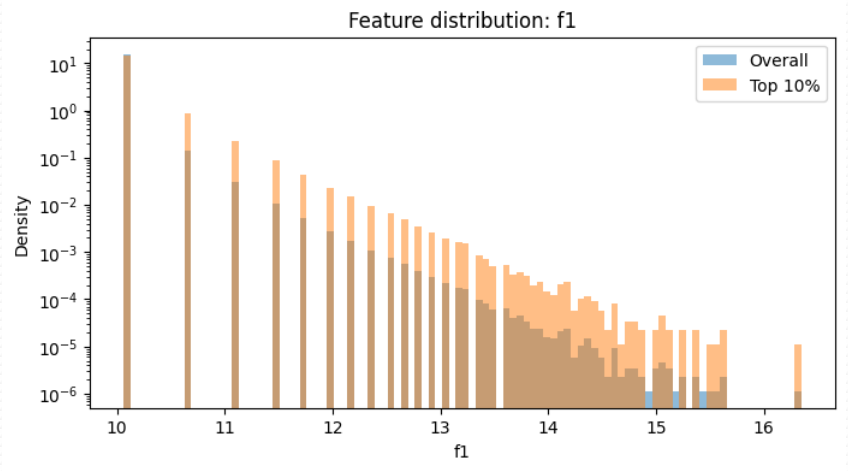
Coefficients

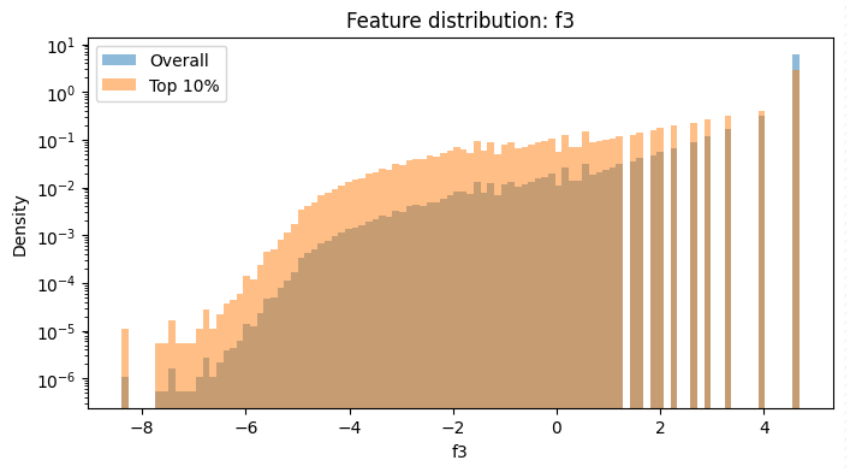


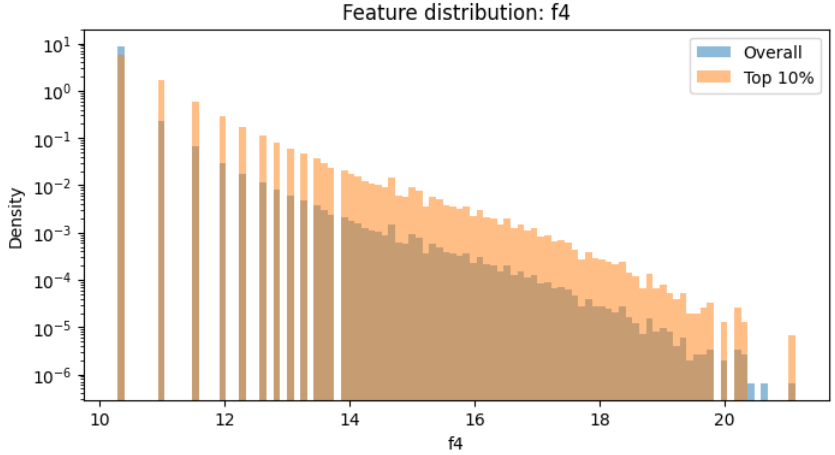


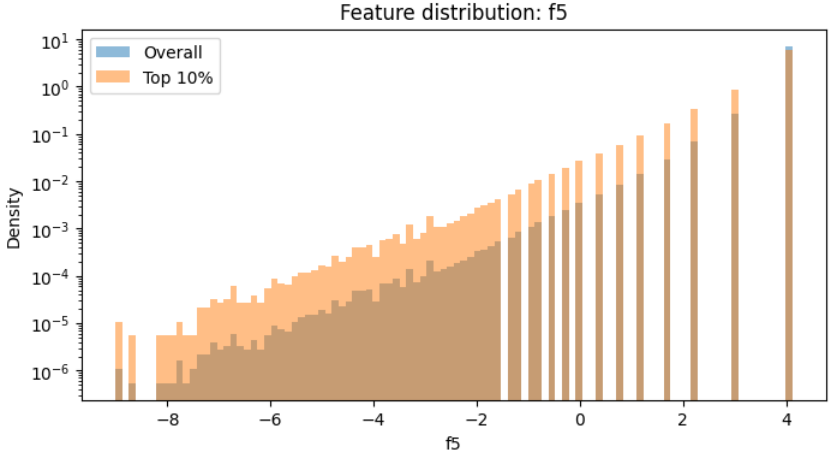


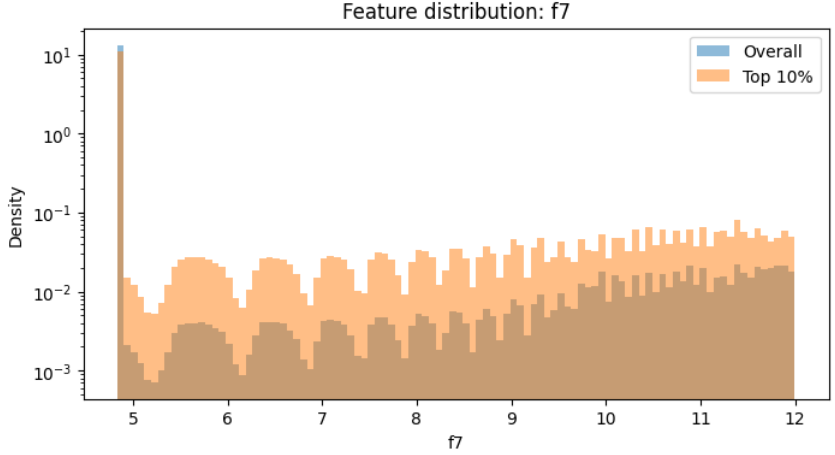
Remain top 10% features

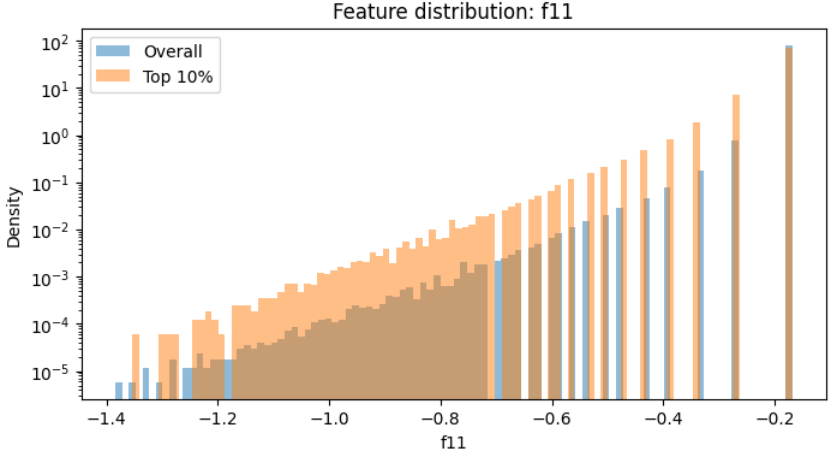





Feature distribution: f4
Overall
Top 10%
Density
f4


Feature distribution: f5
Overall
Top 10%
Density
f5


Feature distribution: f7
Overall
Top 10%
Density
f7


Feature distribution: f11
Overall
Top 10%
Density
f11

S-learner

P(Y = 1 , X)
X : features + treatment

- Use only one model instead of two (T-learner)
- Predict conversion using features and treatment
- Calculate the uplift score
- Sort the data using the uplift score
- Calculate the real uplift between both groups per percentiles

Class Transformation

P(Z = 1 , X)
X : features

- Create a new target “Z” to predict, “if treat that user is better than no treat”
- Create a model (XGBoost) that predicts “Z” using only the feature values
- Sort by predicted Z score
- Calculate the real uplift between both groups per percentiles

Y \ T	0	1
0	Z = 1 (convert if treated?)	Z = 0 (waste of resources)
1	Z = 0 (no need to treat)	Z = 1 (the treatment worked)