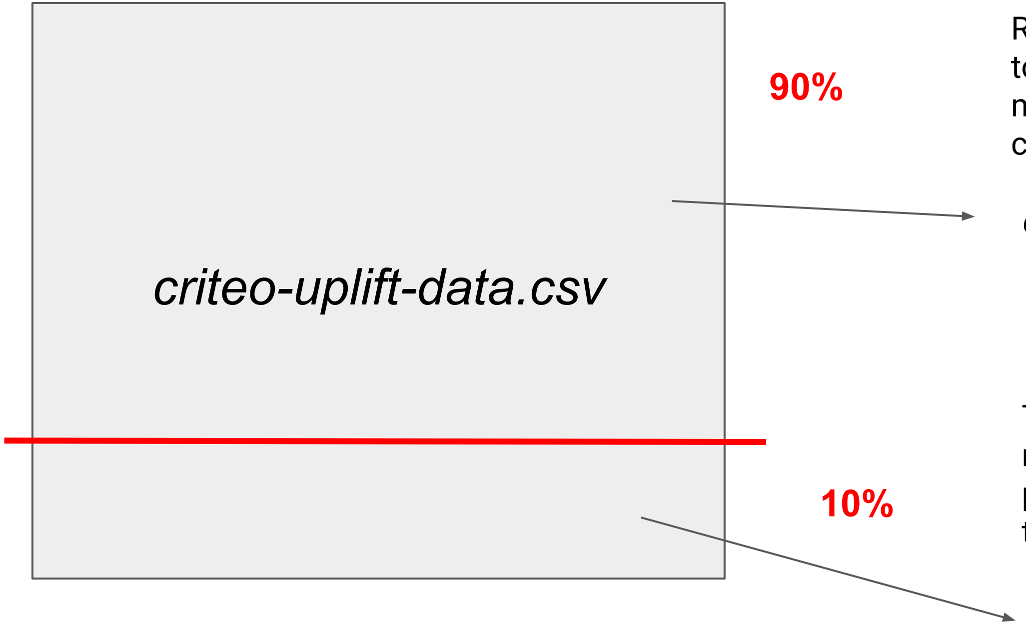
0. Split original data to simulate uplift execution



Represents historical data. It is used to train & validate the propensity model based on exposure + covariates.

criteo-uplift-model.csv

This sample, not included in model-training, represents a new population - the uplift model is user to optimize response vs. spend

criteo-uplift-score.csv

Traditional Uplift Model Setup

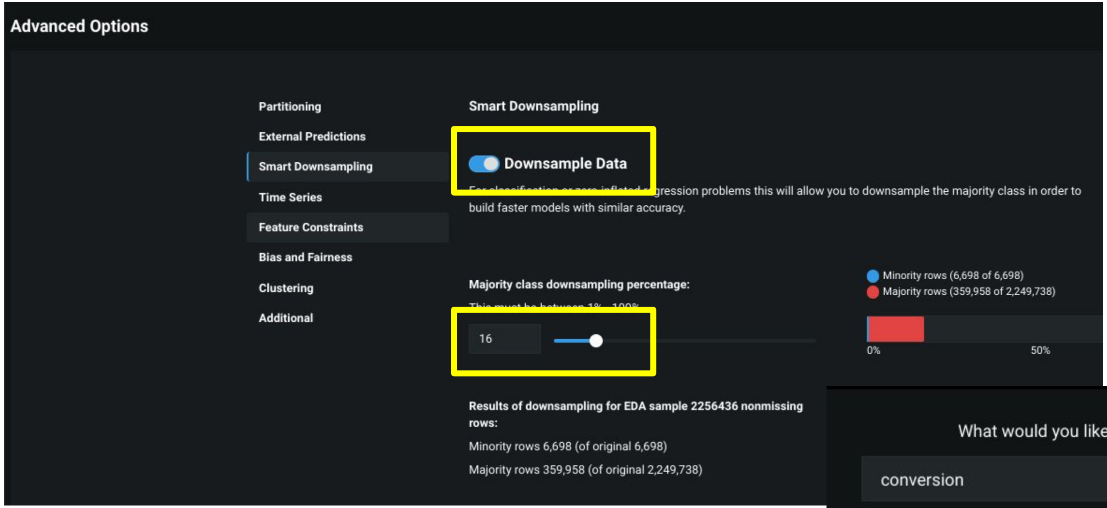
Menu | Search | Feature List: All Features | View Raw Data | + Create feature list | 1-16 of 16

Feature Name	Data Quality	Index	Var Type	Unique	Missing	Mean	Std Dev	Median	Min	Max
exposure		16	Numeric	2	0	0.03	0.17	0	0	1
visit		15	Numeric	2	0	0.05	0.21	0	0	1
conversion		14	Numeric	2	0	0.00	0.05	0	0	1
treatment		13	Numeric	2	0	0.85	0.36	1	0	1
f11	i	12	Numeric	80	0	-0.17	0.02	-0.17	-1.28	-0.17
f10	i	11	Numeric	95,101	0	5.33	0.17	5.30	5.30	6.47

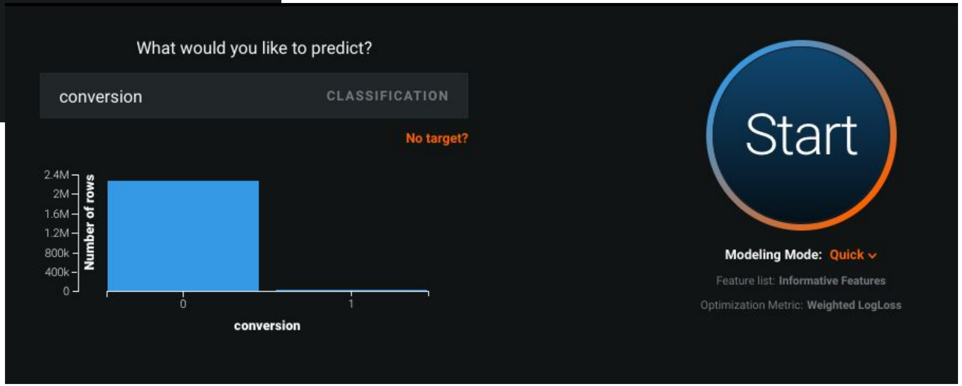
A treatment with multiple levels

Binary target

1. Launch AutoPilot to predict **conversion**



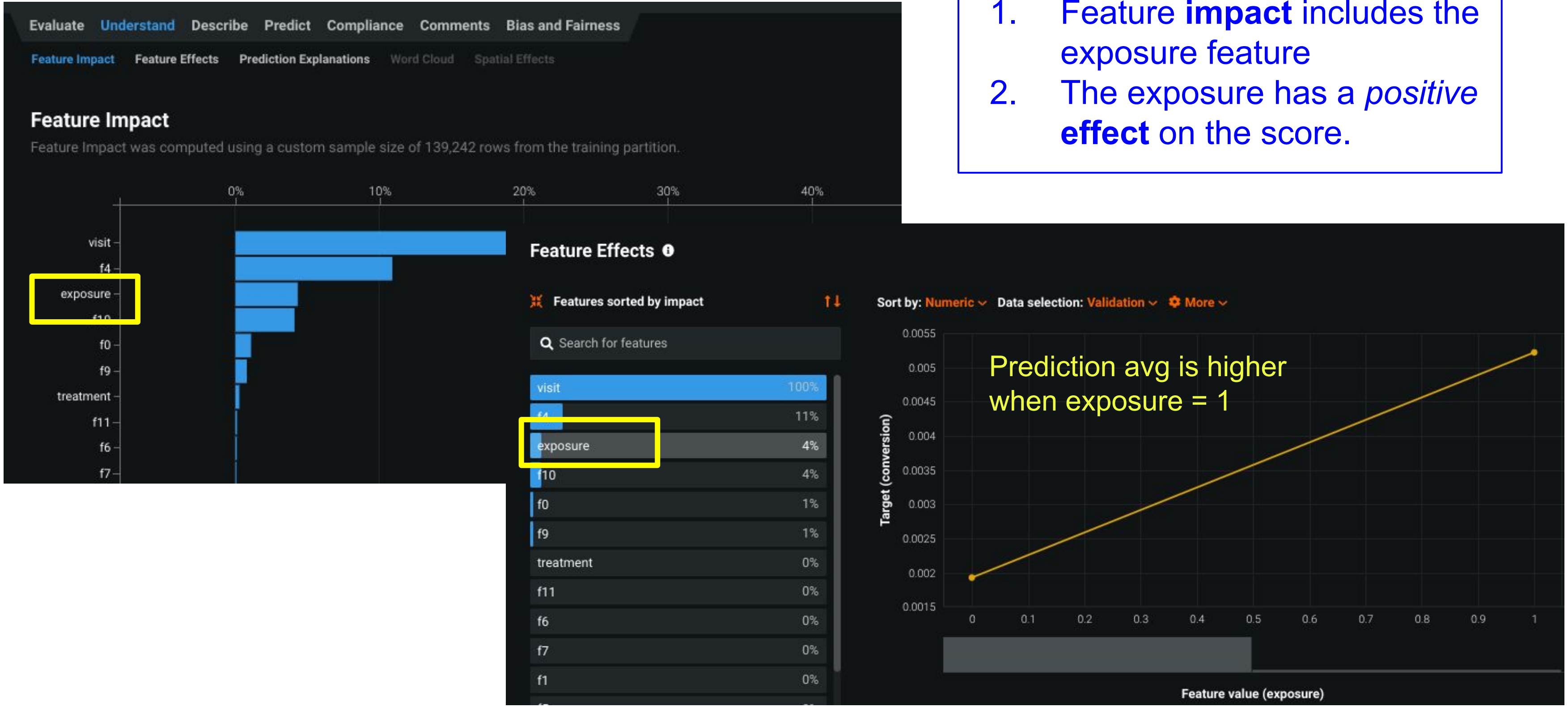
If your target is binary and minority class is very small (<1%), consider ***downsampling***. It's unlikely to improve performance but will definitely speed up training.



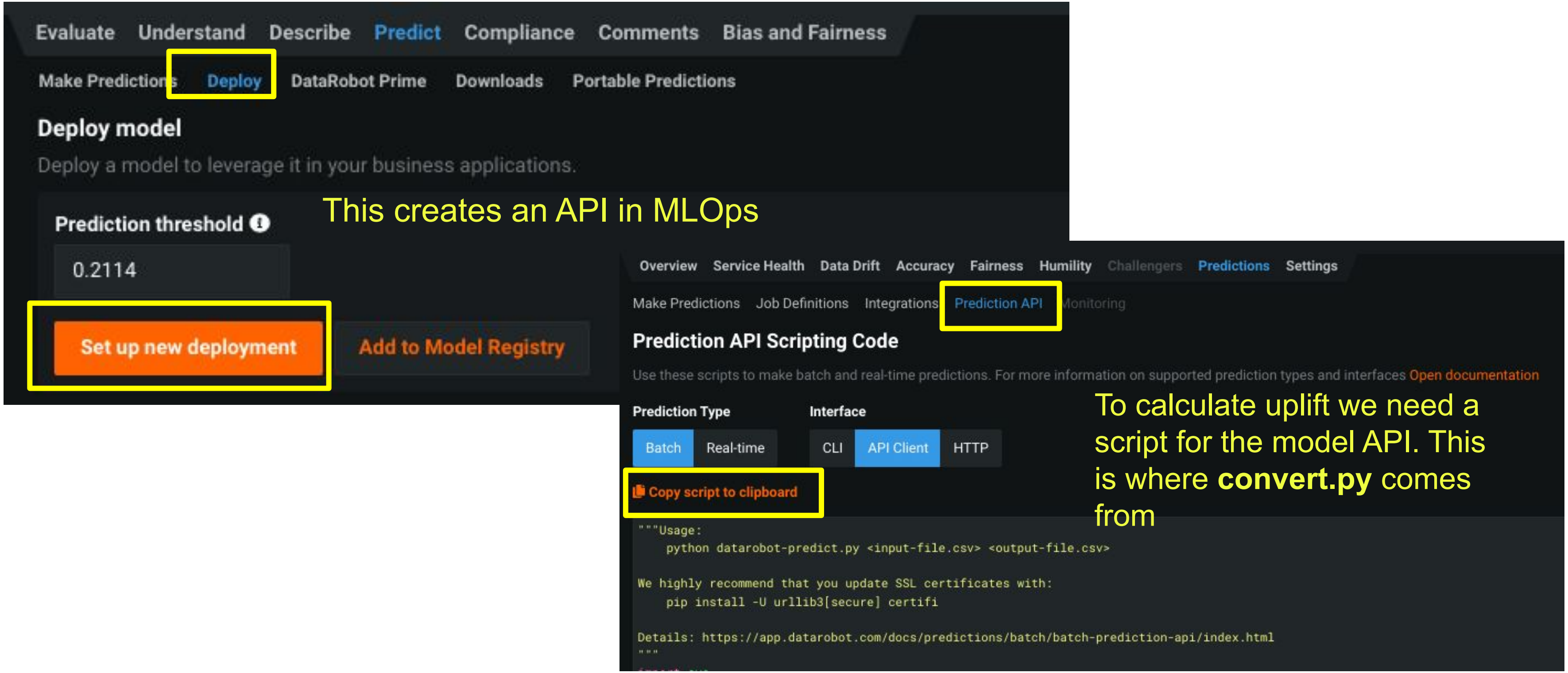
2. Confirm AutoPilot Results

Verify the following:

1. Feature **impact** includes the exposure feature
2. The exposure has a *positive* **effect** on the score.



3. Deploy the Model to generate uplift scores



4a. Prepare score file and calculate uplift

Use the API to create **propensity*** and **response*** scores.

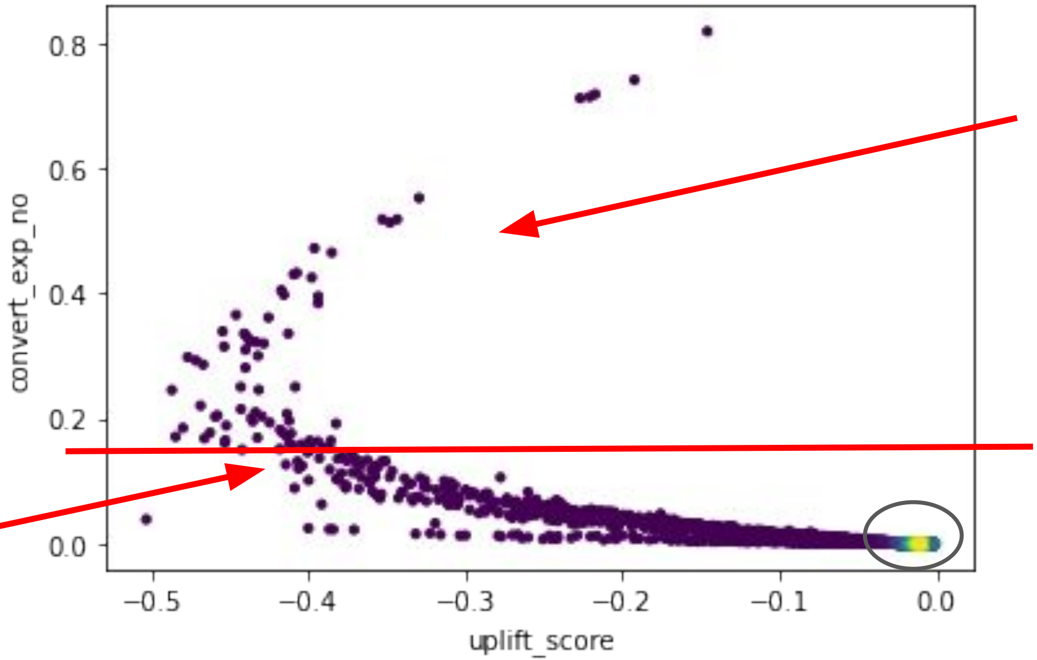
```
audience = pd.read_csv('./customers.csv')

audience['exposure']=0

audience.to_csv(path_or_buf='./audience')
os.system("./convert.py audience
wo_exp.csv")

audience['exposure']=1

audience.to_csv(path_or_buf='./audience')
os.system("./convert.py audience
with_exp.csv")
```

Difference of prop. and resp = **uplift score**.

```
noexp =
pd.read_csv('./wo_exp.csv').rename(columns={'con
version_1_PREDICTION': 'convert_exp_no'})

exp =
pd.read_csv('./with_exp.csv').rename(columns={'c
onversion_1_PREDICTION': 'convert_exp_yes'})

scores =
pd.concat([noexp[['convert_exp_no']],exp[['conve
rt_exp_yes']]],axis=1)

scores['uplift_score']=scores['convert_exp_no']-
scores['convert_exp_yes']

scores.plot.scatter(x = 'uplift_score', y =
'convert_exp_no');
```

** Propensity = Pr(purch. | no intervention). Response = Pr(purch. | intervention)*

4b. Uplift Scatter plot



These customers will respond regardless of exposure (**sure things** + **sleeping dogs**)

These are the **persuadables** (where you should spend ad money starting at top-left)

Note: this is a density plot. Most of the scores appear in this small oval

Appendix: Uplift open-source package

The DataRobot approach to the automation of uplift modeling is based on the Causal ML methodology (originally developed at uber).

Documentation: https://causalml.readthedocs.io/en/latest/about.html
Youtube pres: https://www.youtube.com/watch?v=2J9j7peWQgI
Python package: https://github.com/uber/causalml
White paper: https://arxiv.org/abs/2002.11631

CausalML

Goal of Uplift Modeling

Uplift model estimates heterogeneous treatment effects with ML algorithms

Conditional average treatment effect: **CATE** = E [Y| Intervention, X] - E [Y | No Intervention, X]

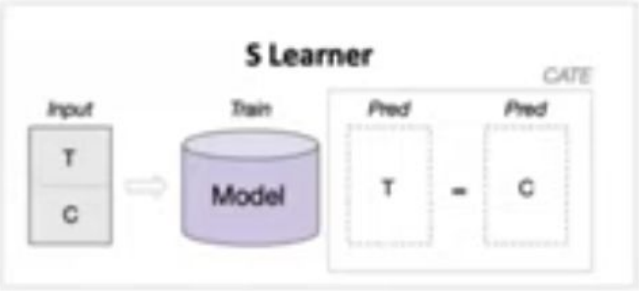
Sure things & Lost causes – Will behave the same no matter what you do. Including them as a target in the model is okay, but will make our targeting inefficient.

Sleeping dogs - These people are turned off by your intervention. Definitely don't include them, ideally you would even downrank them.

Persuadable - This is the population you actually care about because they exhibit the ideal behavior *because* you intervened. Ideally you uprank them as much as possible

Source: https://www.youtube.com/watch?v=2J9j7peWQgI

S Meta-Learner for Uplift Modeling



Procedure

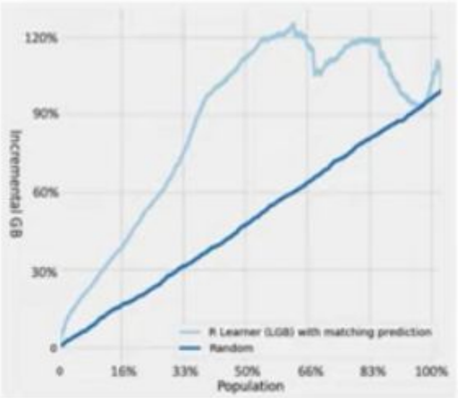
1. Create a binary feature is_treatment, indicating whether a user is from the treatment group
2. Train a single (S) model
3. For all users, set is_treatment to 1 and calculate $yhat_{is\_treatment=1}$
4. For all users, set is_treatment to 0 and calculate $yhat_{is\_treatment=0}$
5. CATE = $yhat_{is\_treatment=0}$ - $yhat_{is\_treatment=1}$

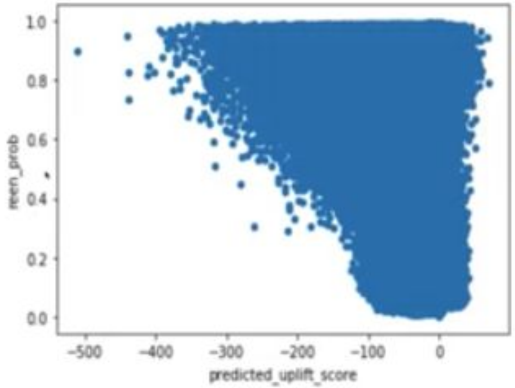
Source: https://www.youtube.com/watch?v=2J9j7peWQgI

Optimization with Uplift Model

Uplift Modeling @ Uber

- Maintaining > 100% campaign incremental values while targeting only ~ **50%** population by identify the persuadable users
- Propensity score and uplift score are weakly negatively correlated . While propensity score model is easier to build in general, it cannot substitute the uplift model.





Source: https://www.youtube.com/watch?v=2J9j7peWQgI