Using DataRobot with Propensity Score Matching for Quasi-Experimentation

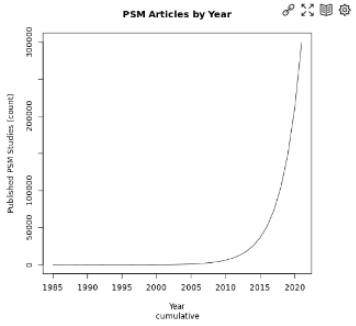
Interpreter: python. **FINISHED** Took 0 millisec. Updated by on May 26 2021, 11:50:00 AM (EDT) (outdated)

Who uses PSM?

- Medical Researchers
- Marketers
- ROI Data Scientists
- Economists
- Anyone estimating treatment effects without a perfect experiment

In 2018 Google Scholar had 127,000 results for "propensity score" AND (match OR matched OR matching). By November 2020, that number was already up to 263,000 - and a high % of the new articles were researching treatments for COVID-19 (Miller, 2020).

Interpreter: md. **FINISHED**
Took 0 millisec. Updated by jason.miller@datarobot.com on May 27 2021, 2:41:40 PM (EDT)



Interpreter: spark.r. **FINISHED**
Took 0 millisec. Updated by on May 25 2021, 11:28:35 AM (EDT)

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 25 2021, 11:36:25 AM (EDT)

Example Scenario

Acme Widgets Co. wants to increase sales and has tried offering a promotion to some customers which offers them a free gift with any purchase. The cost of the promotion was $6,000. Customers were selected from an email sign up list to receive the promotion, but the Data Science department insists that this introduces selection bias since the demographic profile of customers on the email list is systematically different from customers overall - for instance, customers on the list tend to be younger and more frequently located in the United States.

The company needs to estimate the effect of the promotion on spending to calculate ROI and determine if it will continue testing promotions. In the past, the corporate HQ requested the marketing department to use regression models for this purpose. However, marketing has a vested interest in showing high ROI from their campaigns, thus HQ became concerned that marketing analysts were biased towards chosing model specifications that indicate higher ROI. Instead, HQ has requested a quasi-experiment, wherein no modeling is performed involving targeted outcomes. Instead, we estimate the probability of treatment assignment (i.e. the Propensity Score), then use that to create a matched sample. Average Effect of Treatment on the Treated (ATT) is estimated on the matched sample and used to calculate ROI with less potential for bias.

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 26 2021, 11:48:25 AM (EDT)

Step 1: Modeling the Propensity Score

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 26 2021, 11:50:33 AM (EDT)

Load Libaries

Interpreter: md.

```
%python
import datarobot as dr
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
```

Interpreter: python. **FINISHED** Took 0 millisec. Updated by jason.miller@datarobot.com on May 29 2021, 5:41:20 PM (EDT)

Connect to DR

Interpreter: md.

```
# Credentials
MAIN_CREDS = z.getDatasource("JM_DR")
MAIN_TOKEN = MAIN_CREDS["token"]
DEMO_CREDS = z.getDatasource("JM_DR_Demo")
DEMO_TOKEN = DEMO_CREDS["token"]

# Connect
dr.Client(token=MAIN_TOKEN , endpoint='https://app.datarobot.com/api/v2')
```

Interpreter: python. **ABORT** Took 0 millisec. Updated by jason.miller@datarobot.com on May 29 2021, 7:39:42 PM (EDT)

Import Data Set

Interpreter: md.

```
!wget http://zdata/marketing_promotional_campaign.csv
```

```
--2021-05-29 21:41:31--  http://zdata/marketing_promotional_campaign.csv
Resolving zdata (zdata)... 127.0.0.1
Connecting to zdata (zdata)|127.0.0.1|:80... connected.
HTTP request sent, awaiting response...
200 OK
Length: unspecified [text/plain]
Saving to: 'marketing_promotional_campaign.csv'


          marketing     [<=>                ]       0  --.-KB/s
marketing_promotion     [ <=>               ] 272.19K  --.-KB/s    in 0.002s

2021-05-29 21:41:32 (169 MB/s) - 'marketing_promotional_campaign.csv' saved [278719]
```

Interpreter: python. **FINISHED** Took 1 sec 0 millisec. Updated by jason.miller@datarobot.com on May 29 2021, 5:41:32 PM (EDT)

```
# Even with a custom feature list, DR won't completely ignore the target column like we'll want, so we need to drop it from a copy of the data set
data = pd.read_csv('marketing_promotional_campaign.csv')

# drop the target
data.drop('spend', inplace=True, axis=1)
print(data)

# write a CSV of the subset of data
data.to_csv('marketing_promo_no_target.csv')
```

```
     gender        age region  received_promotional_credit
0         1  30.827442     US                          True
```

```
      gender  age region  received_promotional_credit
0          1  30.827442     US                 True
1          1  30.582260     US                 True
2          1  30.042642     US                 True
3          0  29.133073     US                 True
4          0  29.837817     US                 True
...      ...        ...    ...                  ...
5995       1  30.831919     UK                False
5996       1  31.743311     UK                False
5997       1  33.437218     UK                False
5998       1  32.923452     UK                False
5999       1  32.774871     UK                False

[6000 rows x 4 columns]
```

Interpreter: python. **FINISHED** Took 0 millisec. Updated by jason.miller@datarobot.com on May 29 2021, 5:41:37 PM (EDT)

```
# send the new data set to AI Catalog
dataset = dr.Dataset.create_from_file(file_path='marketing_promo_no_target.csv')
```

Interpreter: python. **ABORT**

Create a Project

Interpreter: md.

```
project = dr.Project.create_from_dataset(dataset.id, project_name='Zepl NB: Matching for Quasi Experimentation (v2.2.1)')
```

Interpreter: python. **FINISHED** Took 24 sec 0 millisec. Updated by jason.miller@datarobot.com on May 29 2021, 6:39:35 PM (EDT)

Set Target to Treatment Assignment, Add Feature List *That Omits the True Target*, and Start Running Autopilot

Interpreter: md.

```
# Special note:
#
#   Using a feature list to omit the target turned out to be inefficient since DR still tries models with all informative features. Since we dropped the target
#       ahead of time we no longer strictly need a custom feature list in this step.


# Set target, feature list, and run the project
project.set_target(target='received_promotional_credit', # this is our treatment vs control flag (received_promotional_credit)
                   metric='LogLoss',
                   mode=dr.AUTOPILOT_MODE.FULL_AUTO)

featurelist = project.create_featurelist('Omit the Target for Matching', ['gender', 'age', 'region', 'received_promotional_credit'])
project.set_worker_count(-1)
project.start_autopilot(featurelist.id) # optional: add wait for autopilot to complete
```

Interpreter: python. **FINISHED** Took 1 min 0 millisec. Updated by jason.miller@datarobot.com on May 26 2021, 4:36:44 PM (EDT)

Cross-Validate All Models

Interpreter: md.

```
lb = project.get_models()

for model in lb:
    try:
        model.cross_validate()
    except:
        pass
```

Interpreter: python. **FINISHED** Took 2 min 13 sec 0 millisec. Updated by jason.miller@datarobot.com on May 26 2021, 4:43:44 PM (EDT)

Unlock Holdouts

Interpreter: md.

```
project.unlock_holdout()
```

Interpreter: python. **ABORT** Took 0 millisec. Updated by jason.miller@datarobot.com on May 29 2021, 7:39:42 PM (EDT)

Get the Best Model and Learn About it

Interpreter: md.

```
# Resume (if notebook has been stopped):
project = dr.Project.get('60aeac5beb87822a16af5148')
# To do: make this more programmatic
model = dr.Model.get(project=project.id,
                     model_id="60aeb7c03d594b70a8a5fcfa")
```

Interpreter: python. **ABORT** Took 0 millisec. Updated by jason.miller@datarobot.com on May 27 2021, 12:26:00 PM (EDT)

```
#Get Feature Impact
feature_impact = model.get_or_request_feature_impact()

#Save feature impact in pandas dataframe
fi_df = pd.DataFrame(feature_impact)

#Dispaly using Zepl's built in Vizualizations
z.show(fi_df)
```

ZEPL ZEPPELIN

Download

Index	redundantWith	featureName	impactNormalized	impactUnnormalized
0	None	age	1.0	0.46530869428797283
1	None	region	0.05652326216690192	0.026300765315777908
2	None	gender	0.013882363226476364	0.006459584306543087

Interpreter: python. **FINISHED** Took 1 sec 0 millisec. Updated by jason.miller@datarobot.com on May 27 2021, 10:24:27 AM (EDT)

```
prediction_server = dr.PredictionServer.list()[0]

deployment = dr.Deployment.create_from_learning_model(
    model.id, label='Propensity Scoring Deployment', description='Deployment for Zepl NB Matching for Quasi-experimentation v2.2',
    default_prediction_server_id=prediction_server.id)
deployment
```

```
Deployment(Propensity Scoring Deployment)
```

Interpreter: python. **FINISHED** Took 5 sec 0 millisec. Updated by on May 27 2021, 10:24:40 AM (EDT)

Propensity Scoring

Interpreter: md.

```
dr.BatchPredictionJob.score_to_file(
    deployment.id,
    'marketing_promo_no_target.csv',
    './scored_results.csv')
```

Interpreter: python. **ABORT** Took 0 millisec. Updated by jason.miller@datarobot.com on May 27 2021, 12:26:01 PM (EDT)

Inspect Model Performance Using ML Practices (not normally done in PSM)

Interpreter: md.

```
%r
install.packages("caret")
install.packages("e1071")
install.packages("Matching")

require(caret)
require(data.table)
require(e1071)
require(Matching)

data   <- fread("marketing_promotional_campaign.csv")
scores <- fread("scored_results.csv")
df     <- cbind(data, scores)
```

Interpreter: spark.r. **FINISHED** Took 50 sec 0 millisec. Updated by jason.miller@datarobot.com on May 27 2021, 10:28:45 AM (EDT) (outdated)

```
%r
#table(df$TC_PREDICTION, df$TC)
#table(df$TC_Treatment_PREDICTION > .4, df$TC)
caret::confusionMatrix(as.factor(df$received_promotional_credit_PREDICTION), as.factor(df$received_promotional_credit)) #
```

```
Confusion Matrix and Statistics

      Reference


Prediction FALSE TRUE
     FALSE  3429  637
     TRUE    571 1363

           Accuracy : 0.7987
             95% CI : (0.7883, 0.8087)
No Information Rate : 0.6667
P-Value [Acc &gt; NIR] : &lt; 2e-16

              Kappa : 0.5432


Mcnemar's Test P-Value : 0.06146

        Sensitivity : 0.8572
        Specificity : 0.6815
     Pos Pred Value : 0.8433
     Neg Pred Value : 0.7048
         Prevalence : 0.6667
     Detection Rate : 0.5715


Detection Prevalence : 0.6777
      Balanced Accuracy : 0.7694
```

Interpreter: spark.r. **FINISHED** Took 0 millisec. Updated by jason.miller@datarobot.com on May 26 2021, 5:26:01 PM (EDT)

Step 2: Matching, Pruning & Quasi-experimentation

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 26 2021, 11:50:57 AM (EDT)

Conduct the Propensity Score Matching and Estimate Average Effect of Treatment on the Treated

Interpreter: md.

```
%r
#head(df)
df$Tr <- ifelse(df$received_promotional_credit == T, 1, 0)

# PSM with 1-to-1 matching, allowing ties, replacement, and estimating ATT
effect       <- Match(Y             = df$spend,
                      Tr            = df$Tr,
                      X             = df$received_promotional_credit_True_PREDICTION, # Propensity Score from DR
                      estimand      = "ATT",
                      M             = 2, # Controls 1-to-1 vs many-to-1
                      ties          = T,
                      replace       = T,
                      caliper       = 0.01,
                      CommonSupport = T
)

summary(effect) # True treatment effect is 10(%)
```

```
Estimate…  10.086
AI SE……  0.12861
```

```
AI SE.....  0.12861
T-stat.....  78.424
p.val.....  &lt; 2.22e-16

Original number of observations..............  5063
Original number of treated obs...............  1979
Matched number of observations...............  1851
Matched number of observations  (unweighted).  8602

Caliper (SDs)........................................  0.01
Number of obs dropped by 'exact' or 'caliper'  128
```

Interpreter: spark.r. **FINISHED** Took 2 sec 0 millisec. Updated by jason.miller@datarobot.com on May 27 2021, 11:25:40 AM (EDT)

Covariate Balance Before and After Matching & Pruning

Interpreter: md.

```
%r
MatchBalance(as.formula(Tr ~ age + gender + region), data=df, nboots=250, match.out = effect)
```

```
***** (V1) age *****
                       Before Matching       After Matching
mean treatment........     29.999            30.009
mean control.........     31.999            31.086
std mean diff.........    -199.74            -107.4

mean raw eQQ diff.....      1.999            1.1316
med  raw eQQ diff.....     1.9453            1.0871
max  raw eQQ diff.....          3            2.5017

mean eCDF diff.......    0.37289           0.26474
med  eCDF diff.......    0.40725           0.28528
max  eCDF diff.......      0.601           0.43455

var ratio (Tr/Co).....    0.50065           0.91062
T-test p-value........ &lt; 2.22e-16        &lt; 2.22e-16
KS Bootstrap p-value.. &lt; 2.22e-16        &lt; 2.22e-16
KS Naive p-value..... &lt; 2.22e-16        &lt; 2.22e-16
KS Statistic.........      0.601          0.43455

***** (V2) gender *****
                       Before Matching       After Matching
mean treatment........     0.4875           0.48406
mean control.........    0.74375           0.48582
std mean diff.........    -51.253          -0.35177

mean raw eQQ diff.....      0.256          0.022437
med  raw eQQ diff.....          0                 0
max  raw eQQ diff.....          1                 1
```

Interpreter: spark.r. **FINISHED** Took 10 sec 0 millisec. Updated by on May 27 2021, 11:25:58 AM (EDT)