PSM Demo #2: Logistic vs ML Profit Comparison

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 29 2021, 5:04:56 PM (EDT)

Scenario

Propensity Score Matching (PSM) is a 2-step technique for quasi-experimentation. Traditionally, a logitistic regression on treatment is estimated in the first step, without reference to the true target variable. In the 2nd step experimental statistics such as Average Effect of Treatment on the Treated (ATT) are estimated on a matched and pruned sample.

This demo illustrates the value of replacing the 1st step's logistic regression with Machine Learning (ML) via DataRobot. A modified copy of the Lending Club data set is used, wherein executives want to know the effect of a certain 'treatment' (marketing outreach, called "Tr" in the data) on the number of referrals received from each existing customer ("num_referrals"). Treatment can not be truly randomized, thus PSM is used to reduce bias in selection as well as model specification.

The expected (average) value of a referral is $900. There is an 800 dollar cost per customer associated with treatment. There are 9,824 customers in the data set, of whom 293 were included in the pilot of the marketing campaign (i.e. 293 treated). During the Data Generation Process (DGP) the treatment effect was defined as 0.75 (75% of 1 referral).

For an introduction to Quasi-Experiments and PSM with DataRobot check out Demo #1:

https://app.zepl.com/ODFHKV0LJ/notebooks/e1be0bcb11264260bede11649f0795ec

Interpreter: md. **FINISHED** Took 0 millisec. Updated by jason.miller@datarobot.com on May 31 2021, 10:09:00 PM (EDT)

Import Raw Data and Connect to DataRobot

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 31 2021, 9:17:56 PM (EDT)

Load Libaries & Connect to DR

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 29 2021, 5:35:27 PM (EDT)

```
%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 31 2021, 9:19:52 PM (EDT)

```
# 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=DEMO_TOKEN , endpoint='https://app.datarobot.com/api/v2')
```

```
<datarobot.rest.RESTClientObject at 0x7fcde400e790>
```

Interpreter: python. **FINISHED** Took 2 sec 0 millisec. Updated by jason.miller@datarobot.com on May 31 2021, 9:19:55 PM (EDT)

Import the Raw Data

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 29 2021, 5:35:27 PM (EDT)

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

```
--2021-06-01 01:20:00--  http://zdata/lending_club_psm_jm.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: 'lending_club_psm_jm.csv.2'


          lending_c     [<=>                 ]       0  --.-KB/s
lending_club_psm_jm     [ <=>                ] 297.17K  --.-KB/s    in 0.001s

2021-06-01 01:20:00 (321 MB/s) - 'lending_club_psm_jm.csv.2' saved [304305]
```

Interpreter: python. **FINISHED** Took 0 millisec. Updated by jason.miller@datarobot.com on May 31 2021, 9:20:00 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('lending_club_psm_jm.csv')

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

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

```
     home_ownership  annual_inc pymnt_plan addr_state  total_acc  Tr
0              RENT     50000.0          n         NC          9   0
1              RENT     34200.0          n         IL         31   0
2              RENT     30000.0          n         FL          8   0
3          MORTGAGE     51000.0          n         NY         23   0
4              RENT     44004.0          n         CA         32   0
...             ...         ...        ...        ...        ...  ..
9889           RENT     49008.0          n         TX         16   0
9890           RENT     50000.0          n         SC          9   0
9891            OWN     83000.0          n         CA         45   1
9892            OWN    100000.0          n         FL         33   1
9893           RENT     62000.0          n         MD         21   0

[9894 rows x 6 columns]
```

Interpreter: python. **FINISHED** Took 0 millisec. Updated by jason.miller@datarobot.com on May 31 2021, 9:20:14 PM (EDT)

```
%r
model.data   <- read.csv("lending_club_psm_no_target_jm.csv",stringsAsFactors = T)
model.data$X <- NULL # drop rowid

summary(model.data)
```

```
home_ownership   annual_inc     pymnt_plan   addr_state     total_acc
 MORTGAGE:4404   Min.   :  2000   n:9892     CA     :1728   Min.   : 1.0
 OTHER   :  33   1st Qu.: 40000   y:   2     NY     : 953   1st Qu.:13.0
 OWN     : 765   Median : 58000              FL     : 701   Median :20.0
 RENT    :4692   Mean   : 68157              TX     : 696   Mean   :22.1
                 3rd Qu.: 82000              NJ     : 479   3rd Qu.:29.0
```

```
                3rd Qu.: 82000              NJ     : 479   3rd Qu.:29.0
                Max.   :900000              VA     : 389   Max.   :90.0
                                            (Other):4948
      Tr
 Min.   :0.00000
 1st Qu.:0.00000
 Median :0.00000
 Mean   :0.02961
 3rd Qu.:0.00000
 Max.   :1.00000
```

Interpreter: spark.r. **FINISHED** Took 0 millisec. Updated by jason.miller@datarobot.com on May 31 2021, 9:21:10 PM (EDT)

Traditional Logistic Regression Propensity Score Model

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 29 2021, 9:18:54 PM (EDT)

```
%r

ps <- glm("Tr ~ annual_inc + pymnt_plan + total_acc + home_ownership", data = model.data, family = binomial)
summary(ps)

saveRDS(ps, "logitistic.RDS")

Call:
glm(formula = “Tr ~ annual_inc + pymnt_plan + total_acc + home_ownership”,
    family = binomial, data = model.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7003  -0.1031  -0.0827  -0.0676   3.7301

Coefficients:
                    Estimate Std. Error z value Pr(&gt;|z|)
(Intercept)        -7.488e+00  3.413e-01 -21.937  &lt; 2e-16 ***
annual_inc          1.017e-05  1.046e-06   9.719  &lt; 2e-16 ***
pymnt_plany        -1.166e+01  1.690e+03  -0.007  0.99450
total_acc           2.833e-02  6.049e-03   4.684 2.81e-06 ***
home_ownershipOTHER -1.036e+01  4.138e+02  -0.025  0.98003
home_ownershipOWN    5.509e+00  2.818e-01  19.549  &lt; 2e-16 ***

home_ownershipRENT   1.003e+00  3.240e-01   3.097  0.00196 **

Signif. codes:  0 '**' 0.001 '' 0.01 '' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2639.7  on 9893  degrees of freedom


Residual deviance: 1410.0  on 9887  degrees of freedom
AIC: 1424
```

Interpreter: spark.r. **FINISHED** Took 5 sec 0 millisec. Updated by on May 31 2021, 9:22:33 PM (EDT) (outdated)

```
%r

# Score the data
model.data$score <- predict(ps, type = "r")

summary(model.data)

# Write out the scored file
write.csv(model.data, "logit_scored_file_no_target_jm.csv", row.names = F)

home_ownership   annual_inc     pymnt_plan   addr_state     total_acc
 MORTGAGE:4404   Min.   :  2000   n:9892     CA     :1728   Min.   : 1.0
 OTHER   :  33   1st Qu.: 40000   y:   2     NY     : 953   1st Qu.:13.0
 OWN     : 765   Median : 58000              FL     : 701   Median :20.0
 RENT    :4692   Mean   : 68157              TX     : 696   Mean   :22.1
                 3rd Qu.: 82000              NJ     : 479   3rd Qu.:29.0
                 Max.   :900000              VA     : 389   Max.   :90.0
                                             (Other):4948
      Tr              score
 Min.   :0.00000   Min.   :0.000000
 1st Qu.:0.00000   1st Qu.:0.002421
 Median :0.00000   Median :0.003561
 Mean   :0.02961   Mean   :0.029614
 3rd Qu.:0.00000   3rd Qu.:0.005686
 Max.   :1.00000   Max.   :0.994504
```

Interpreter: spark.r. **FINISHED** Took 0 millisec. Updated by on May 31 2021, 9:22:40 PM (EDT)

Now let’s estimate the same PS model using DataRobot. We’ll try matching using both sets of scores and compare results.

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 29 2021, 5:33:23 PM (EDT)

DataRobot Propensity Score Modeling

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 29 2021, 5:24:23 PM (EDT)

Add the Data to AI Catalog & Start a New Project

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 29 2021, 5:36:53 PM (EDT)

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

Interpreter: python. **FINISHED** Took 44 sec 0 millisec. Updated by jason.miller@datarobot.com on May 31 2021, 7:45:08 PM (EDT)

```
project = dr.Project.create_from_dataset(dataset.id, project_name='Zepl NB: PSM Demo 2 (v0.7)')
```

Interpreter: python. **FINISHED** Took 22 sec 0 millisec. Updated by jason.miller@datarobot.com on May 30 2021, 9:29:01 PM (EDT)

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

Interpreter: python. **FINISHED**

```
featurelist = project.create_featurelist('Omit the Target for Matching', ['home_ownership', 'annual_inc',  'addr_state', 'Tr', 'pymnt_plan'])
project.set_worker_count(-1)
project.start_autopilot(featurelist.id) # optional: add wait for autopilot to complete
```

Interpreter: python. **FINISHED** Took 2 sec 0 millisec. Updated by on May 30 2021, 9:29:59 PM (EDT)

Cross-Validate All Models

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 30 2021, 5:23:18 PM (EDT)

```
lb = project.get_models()

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

Interpreter: python. **FINISHED** Took 1 sec 0 millisec. Updated by on May 30 2021, 9:30:00 PM (EDT)

Unlock Holdouts

Interpreter: md. **FINISHED**
Took 0 millisec. Updated by on May 29 2021, 5:46:40 PM (EDT)

```
project.unlock_holdout()
```

```
Project(Zepl NB: PSM Demo 2 (v0.5))
```

Interpreter: python. **FINISHED** Took 2 sec 0 millisec. Updated by on May 30 2021, 9:30:02 PM (EDT)

Get the Best Model and Learn About It

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 29 2021, 5:46:47 PM (EDT)

```
# Resume (if notebook has been stopped):
project = dr.Project.get('60b43bc75ef7eff1f9040497')

# To do: make this more programmatic
model = dr.Model.get(project=project.id,
                     model_id="60b440383485bc0bde069677")
```

Interpreter: python. **ABORT** Took 0 millisec. Updated by jason.miller@datarobot.com on May 31 2021, 11:11:54 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	home_ownership	1.0	0.4272574435290796
1	None	annual_inc	0.45020166655047994	0.1923520131228892

Interpreter: python. **FINISHED** Took 2 sec 0 millisec. Updated by on May 30 2021, 10:22:01 PM (EDT)

Deploy

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 29 2021, 6:42:33 PM (EDT)

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

deployment = dr.Deployment.create_from_learning_model(
    model.id, label='PSM Demo 2 Deployment', description='Deployment for Zepl NB PSM Demo #2 (v0.6)',
    default_prediction_server_id=prediction_server.id)
deployment # 60b43e2bfd805d815fa89f73
```

```
Deployment(PSM Demo 2 Deployment)
```

Interpreter: python. **FINISHED** Took 4 sec 0 millisec. Updated by on May 30 2021, 10:22:24 PM (EDT)

Propensity Scoring

Interpreter: md. **FINISHED**
Took 0 millisec. Updated by on May 29 2021, 6:53:24 PM (EDT)

```
deploymentid = '60b4485d659b85d960180b32'
dr.BatchPredictionJob.score_to_file(
    deploymentid,
    'lending_club_psm_no_target_jm.csv',
    './scored_results_60b4485d659b85d960180b32.csv')
```

Interpreter: python. **ABORT** Took 0 millisec. Updated by jason.miller@datarobot.com on May 31 2021, 11:11:54 PM (EDT)

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

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 29 2021, 6:56:16 PM (EDT)

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 29 2021, 6:56:16 PM (EDT)

```
%r
#Load R Libaries
if(!require(pacman)) install.packages("pacman"); require(pacman)
p_load(caret, data.table, e1071, Matching)
```

Interpreter: spark.r. **FINISHED** Took 3 min 5 sec 0 millisec. Updated by jason.miller@datarobot.com on May 31 2021, 9:06:14 PM (EDT)

```
%r
data     <- fread("lending_club_psm_jm.csv")
drscores <- fread("scored_results_60b4485d659b85d960180b32.csv")
lgscores <- fread("logit_scored_file_no_target_jm.csv")
df       <- cbind(data, drscores)

df$dr_prediction    <- df$Tr_1_PREDICTION
df$logit_prediction <- lgscores$score
head(df)
```

```
home_ownership annual_inc pymnt_plan addr_state total_acc Tr num_referrals
1:           RENT      50000          n         NC         9  0             2
2:           RENT      34200          n         IL        31  0             0
3:           RENT      30000          n         FL         8  0             1
4:       MORTGAGE      51000          n         NY        23  0             2
5:           RENT      44004          n         CA        32  0             1
6:           RENT      65500          n         GA        15  0             0
   Tr_1_PREDICTION Tr_0_PREDICTION Tr_PREDICTION THRESHOLD POSITIVE_CLASS
1:     0.004929825       0.9950702             0       0.5              1
2:     0.004929825       0.9950702             0       0.5              1
3:     0.005745331       0.9942547             0       0.5              1
4:     0.006737336       0.9932627             0       0.5              1
5:     0.004929825       0.9950702             0       0.5              1
6:     0.004929825       0.9950702             0       0.5              1
   dr_prediction logit_prediction
1:   0.004929825      0.003266169
2:   0.004929825      0.005177878
3:   0.005745331      0.002592495
4:   0.006737336      0.001801189
5:   0.004929825      0.005880834
6:   0.004929825      0.004526462
```

Interpreter: spark.r. **FINISHED** Took 0 millisec. Updated by jason.miller@datarobot.com on May 31 2021, 9:22:51 PM (EDT)

Matching with Both Traditional & DR Scores

Interpreter: md. **FINISHED** Took 0 millisec. Updated by jason.miller@datarobot.com on May 29 2021, 9:33:35 PM (EDT)

```
%r

# Logistic Regression PSM
logit        <- Match(Y             = df$num_referrals,
                      Tr            = df$Tr,
                      X             = df$logit_prediction, # Propensity Score from Logistic Reg.
                      estimand      = "ATT",
                      M             = 1, # Controls 1-to-1 vs many-to-1
                      ties          = F,
                      replace       = T,
                      caliper       = 0.01,
                      CommonSupport = T
)

summary(logit)
```

```
Estimate...  0.91358
SE.........  0.1513
T-stat.....  6.0381
p.val......  1.5594e-09

Original number of observations..............  9810
Original number of treated obs...............  292
Matched number of observations...............  162
Matched number of observations  (unweighted).  162

Caliper (SDs)........................................  0.01
Number of obs dropped by 'exact' or 'caliper'  130
```

Interpreter: spark.r. **FINISHED** Took 0 millisec. Updated by on May 30 2021, 10:26:26 PM (EDT)

```
%r
# DataRobot PSM
summary(df$dr_prediction[df$Tr==1])
summary(df$dr_prediction[df$Tr==0])

drps         <- Match(Y             = df$num_referrals,
                      Tr            = df$Tr,
                      X             = df$dr_prediction, # Propensity Score from DR
                      estimand      = "ATT",
                      M             = 1, # Controls 1-to-1 vs many-to-1
                      ties          = T,
                      replace       = F,
                      caliper       = 2
)


# DataRobot
summary(drps)
```

```
Min.  1st Qu.   Median     Mean  3rd Qu.     Max.


0.004441 0.990940 0.990940 0.822851 0.990940 0.990940
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.001349 0.004930 0.005745 0.005562 0.006737 0.012601

Estimate...  0.78
SE.........  0.27465
T-stat.....  2.84
p.val......  0.0045115

Original number of observations..............  9894
Original number of treated obs...............  293
Matched number of observations...............  50
Matched number of observations  (unweighted).  50

Caliper (SDs)........................................  2
Number of obs dropped by 'exact' or 'caliper'  243
```

Interpreter: spark.r. **FINISHED** Took 0 millisec. Updated by jason.miller@datarobot.com on May 31 2021, 9:27:46 PM (EDT)

Business Interpretation

Interpreter: md. **FINISHED** Took 0 millisec. Updated by on May 29 2021, 9:38:04 PM (EDT)

Assigning Dollar Values