dominant-donor-study

August 27, 2020

[1]:

```
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:80% !important; }</style>"))

# Set to automatically reload.
%load_ext autoreload
%autoreload 2

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

```
<IPython.core.display.HTML object>
```

1 Ransdomization estimation of probability of event

1.1 Using Student's intependent t-test as measure

- Sensitive to change in location - such as average prediction metrics.

[2]:

```
from scipy.stats import ttest_ind
import random
import math

def randomization_t_test(flags, values,  max_iter=1000):
    ref_t = ttest_ind(values[flags], values[~flags], equal_var=False)[0]

    values = values.copy()
    results = []
    for i in range(max_iter):
        random.shuffle(values)
        results.append(ttest_ind(values[flags], values[~flags],␣
 ↪equal_var=False)[0])

    results.sort()

    def find_index(elements, value):
        left, right = 0, len(elements) - 1

        while left <= right:
            middle = (left + right) // 2

            if math.isclose(elements[middle], value):
                return middle
            # Could improve this to give more precise value at lower end.
            elif middle == left or middle == right:
                return (left+right)/2

            if elements[middle] < value:
                left = middle + 1
            elif elements[middle] > value:
                right = middle - 1

    posn = find_index(results, ref_t)

    p_value = (posn+0.5)/max_iter
    p_value = min(p_value, 1-p_value)
    return (ref_t, p_value)
```

2 Dominant donor study - dominant by fraction of borrowed

[3]:

```
import pandas as pd

def test_for_dominant_donor_difference(borrowed_min_val, fraction=0.667,␣
 ↪max_iter=100):

    min_val = str(borrowed_min_val)
    table = pd.
 ↪read_csv("cv-10-fold-all-Tokens-10-fold-CV-min-"+min_val+"-prfa-bydonormin.
 ↪csv")

    donor_flag = table['donor_frac'].ge(fraction)
    table = table.iloc[:,3:]
    print(table.groupby(donor_flag).size())
    print(table.groupby(donor_flag).mean())

    for col in table.columns:
        result = randomization_t_test(donor_flag, table[col], max_iter=max_iter)
        print(f'Var {col}, t={result[0]:.3f}, p-value={result[1]:.4f}')
```

[4]:

```
test_for_dominant_donor_difference(300, fraction=0.667, max_iter=5000)
```

```
donor_frac
False    9
True     8
dtype: int64
             bs_prec  bs_recall     bs_f1    bs_acc   md_prec  md_recall  \
donor_frac
False       0.308444   0.672111  0.389889  0.759444  0.672222   0.585111
True        0.536125   0.739250  0.587500  0.795625  0.785125   0.722375

               md_f1    md_acc   nd_prec  nd_recall     nd_f1    nd_acc
donor_frac
False       0.622444  0.771000  0.690111   0.606889  0.642333  0.784333
True        0.749125  0.835625  0.810375   0.722250  0.759500  0.842625
Var bs_prec, t=2.141, p-value=0.0293
Var bs_recall, t=2.306, p-value=0.0177
Var bs_f1, t=1.973, p-value=0.0336
Var bs_acc, t=1.230, p-value=0.1254
Var md_prec, t=3.781, p-value=0.0017
Var md_recall, t=3.345, p-value=0.0025
Var md_f1, t=3.640, p-value=0.0007
Var md_acc, t=2.625, p-value=0.0127
Var nd_prec, t=4.832, p-value=0.0009
Var nd_recall, t=2.628, p-value=0.0082
Var nd_f1, t=3.535, p-value=0.0037
Var nd_acc, t=2.600, p-value=0.0097
```

[5]:

```
test_for_dominant_donor_difference(200, fraction=0.667, max_iter=5000)
```

```
donor_frac
False    14
True     15
dtype: int64
             bs_prec  bs_recall     bs_f1    bs_acc   md_prec  md_recall  \
donor_frac
False       0.223857   0.523643  0.287857  0.783857  0.646071   0.531643
True        0.438933   0.733467  0.505200  0.837667  0.762067   0.632533

               md_f1    md_acc  nd_prec  nd_recall     nd_f1    nd_acc
donor_frac
False       0.577857  0.777643   0.6740   0.556214  0.602429  0.791214
True        0.684600  0.843733   0.8056   0.644533  0.707600  0.851267
Var bs_prec, t=2.443, p-value=0.0102
Var bs_recall, t=2.414, p-value=0.0112
Var bs_f1, t=2.356, p-value=0.0142
Var bs_acc, t=2.156, p-value=0.0180
Var md_prec, t=3.740, p-value=0.0012
Var md_recall, t=2.199, p-value=0.0185
Var md_f1, t=2.760, p-value=0.0042
Var md_acc, t=3.291, p-value=0.0012
Var nd_prec, t=4.531, p-value=0.0002
Var nd_recall, t=1.811, p-value=0.0399
Var nd_f1, t=2.623, p-value=0.0087
Var nd_acc, t=3.008, p-value=0.0032
```

[6]:

```
test_for_dominant_donor_difference(100, fraction=0.667, max_iter=5000)
```

```
donor_frac
False    17
True     20
dtype: int64
             bs_prec  bs_recall     bs_f1    bs_acc   md_prec  md_recall  \
donor_frac
False       0.192353   0.497706  0.252176  0.801647  0.639294   0.505118
True        0.418000   0.736650  0.490200  0.858100  0.762550   0.599550

               md_f1    md_acc   nd_prec  nd_recall     nd_f1    nd_acc
donor_frac
False       0.558235  0.788824  0.654941   0.513412  0.567294  0.794118
True        0.661950  0.855900  0.787850   0.619400  0.684950  0.865000
Var bs_prec, t=3.101, p-value=0.0012
Var bs_recall, t=3.072, p-value=0.0007
Var bs_f1, t=3.100, p-value=0.0017
Var bs_acc, t=2.386, p-value=0.0097
Var md_prec, t=4.551, p-value=0.0002
Var md_recall, t=2.136, p-value=0.0210
Var md_f1, t=2.783, p-value=0.0047
Var md_acc, t=3.771, p-value=0.0002
Var nd_prec, t=4.280, p-value=0.0002
Var nd_recall, t=2.163, p-value=0.0180
Var nd_f1, t=2.767, p-value=0.0037
Var nd_acc, t=4.215, p-value=0.0002
```

[ ]:

```

```