solution_to_pdf

November 2, 2020

1

1.1 1. ,

:

1)

- approval_rate = sum(can_be_branded) / total_cnt

2)

. (
, -).

bad_car_cancel_rate = bad_car_model_tag_cnt / trips_cancel_cnt

defect_trip_rate = trips_defect_cnt / trips_rated_cnt

defect_rate = bad_car_cancel_rate + α * defect_trip_rate

3)

,

1.2 1:

, . .
, .

[94]:

```
Percent of approved cars: 79.0
Percent of sticked cars: 14.9
```

79% . . 15% .

1.3 2:

, “ ”, . . “ ” . ,
, , .

, (bad_car_model_tag_cnt). ,
, . ,
. , .. , , .
, . bad_car_cancel_rate =
bad_car_model_tag_cnt / trips_cancel_cnt.

(defect_rate = defect_trips_cnt /
trips_rated_cnt). - ,
. , .

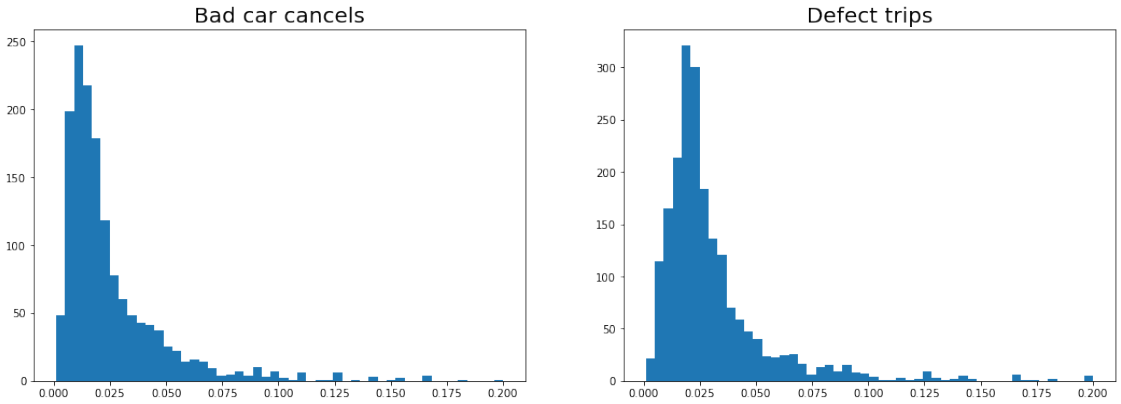
bad_car_cancel_rate + α * defect_rate,

α 0.2-0.5.

, “ ”, .

1.3.1 (,)

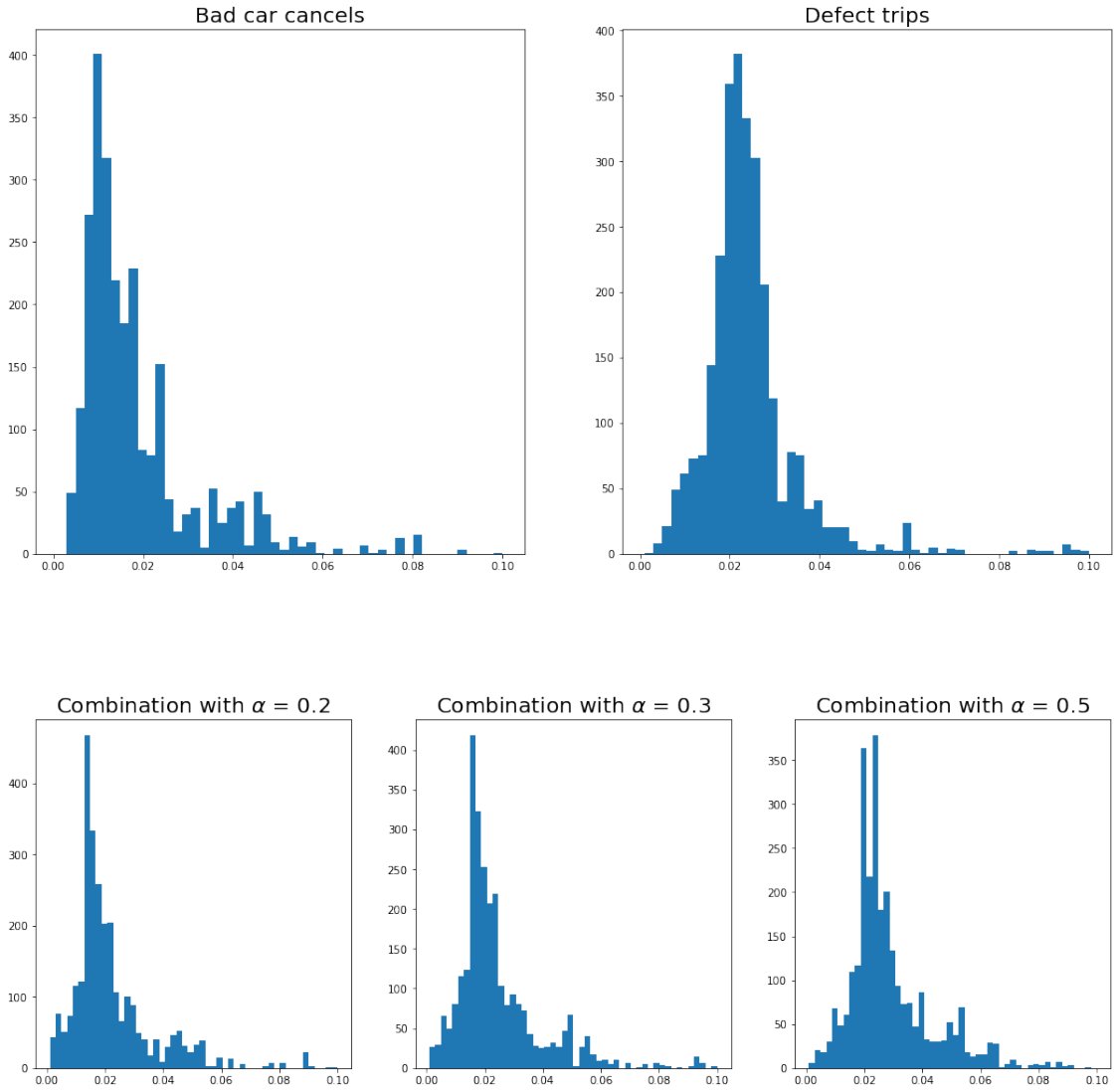
[143]:



, .

1.3.2

[191]:



$\alpha = 0.2$

1.4

:

`defect_rate` n . n, trade-off

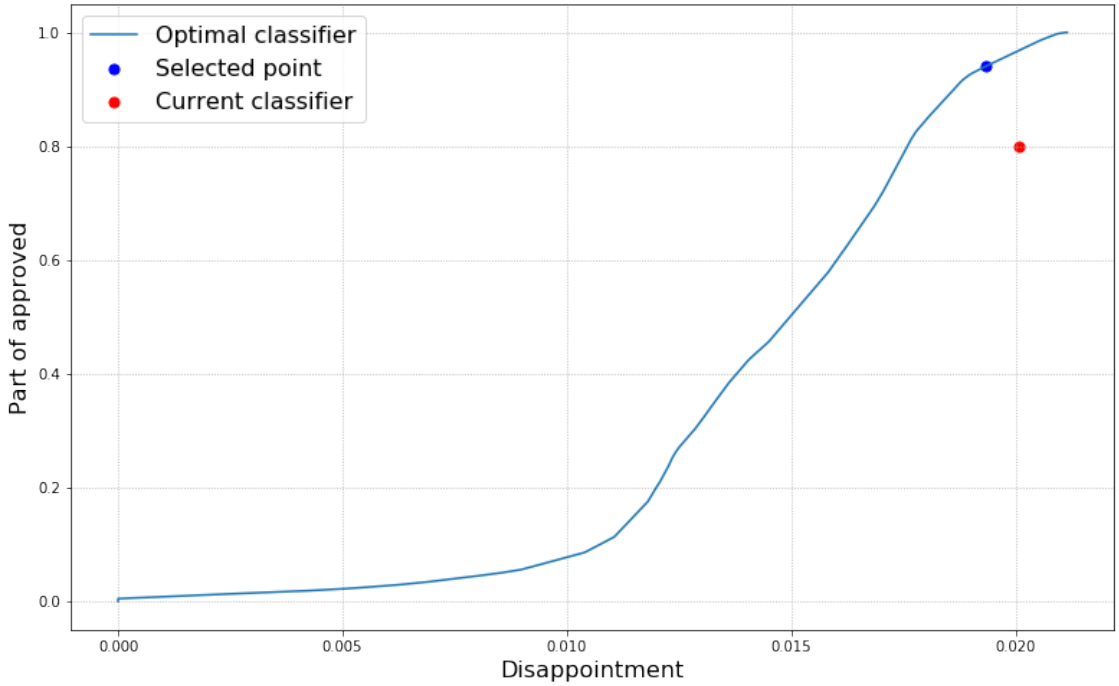
.

. ,

[298]:
```
plt.figure(figsize=(13, 8))
plt.grid(ls=':')
plt.xlabel('Disappointment', fontsize=16)
```

```
plt.ylabel('Part of approved', fontsize=16)
metric1 = (metric_values[:, 0] * metric_values[:, 1]).cumsum() / metric_values[:
 ↪, 1].cumsum()
metric2 = metric_values[:, 1].cumsum()
plt.plot(metric1, metric2, label='Optimal classifier')
i = 550
plt.scatter([metric1[i]], [metric2[i]],
            color='b', s=50, label='Selected point')
plt.scatter([disappointment_rate_of_approved], [part_of_approved],
            color='r', s=50, label='Current classifier')
plt.legend(fontsize=16)
```

[298]: <matplotlib.legend.Legend at 0x7f1b15da97b8>

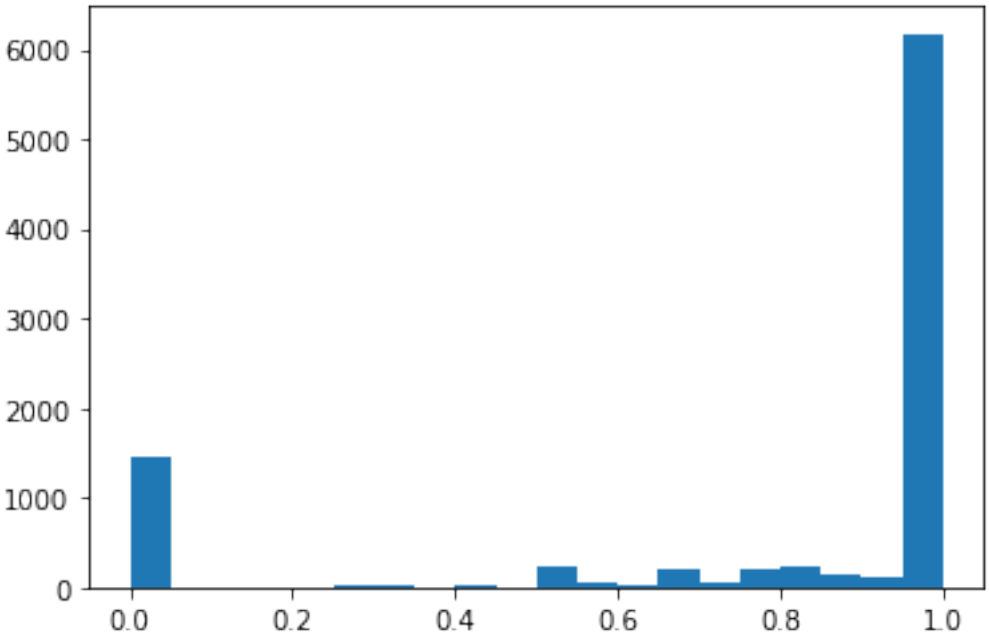


1.5

9000 . 71% (. . ,).

,

[257]:



- 0 1, . . .

[258]:

```
Total part of approved: 0.798
```

. , , . , .

, , , defect_rate.

[301]:

```
Total part of approved with ours: 0.889
```

, , .

, , .

1.6

, , , defect_rate (1.8%) .

, , , . ,
, `defect_rate`.

[ ]:

```
```