<-img src="img1.jpg" align=”center”>

Previsão de transações bancárias de 200.000 clientes Santander

Projeto criado por **Alessandra Faria Abreu** enquanto ouvia Banda Eva e tomava café
Desafio disponivel no *Kaggle*

```
#permite que os gráficos gerados sejam mostrados na mesma janela
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import datasets, linear_model
from sklearn.model_selection import train_test_split
#import plotly.graph_objs as go
#define o estilo dos gráficos , "ggplot" é um estilo popular em R
plt.style.use('ggplot')
```

```
#@title Default title text
#Realiza a leitura da base (csv) e carrega na mémoria
#Para execução dentro do colab-Google
from google.colab import drive
drive.mount('/content/drive/')
A ='/content/drive/My Drive/Trabalhos Faculdade/TDC/test.csv'
B = '/content/drive/My Drive/Trabalhos Faculdade/TDC/train.csv'
transacoesTrain = pd.read_csv(B)
transacoesTest = pd.read_csv(A)
```

Default title text

```
Drive already mounted at /content/drive/; to attempt to forcibly remount, call drive
```

Exploração das Bases

Visualização da relação de dados da base Train - Dados

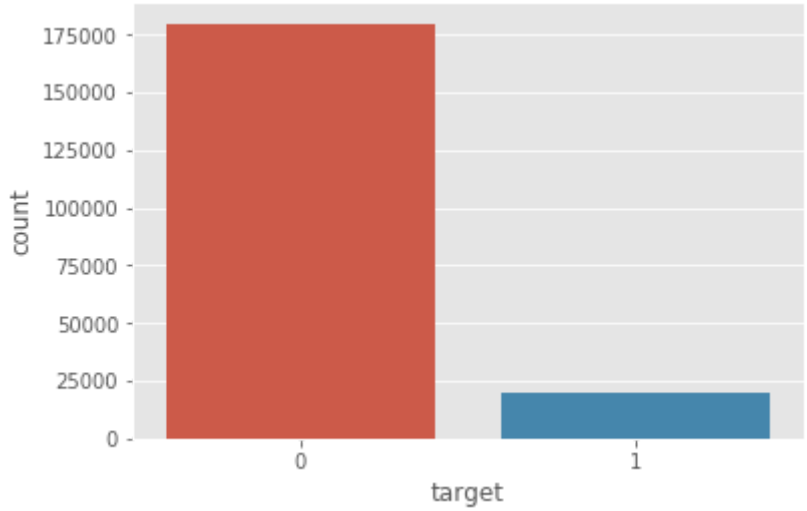
```
#Quantidade de linhas e colunas
transacoesTrain.shape
```

```
(200000, 202)
```

Visualização das Bases - Gráficos

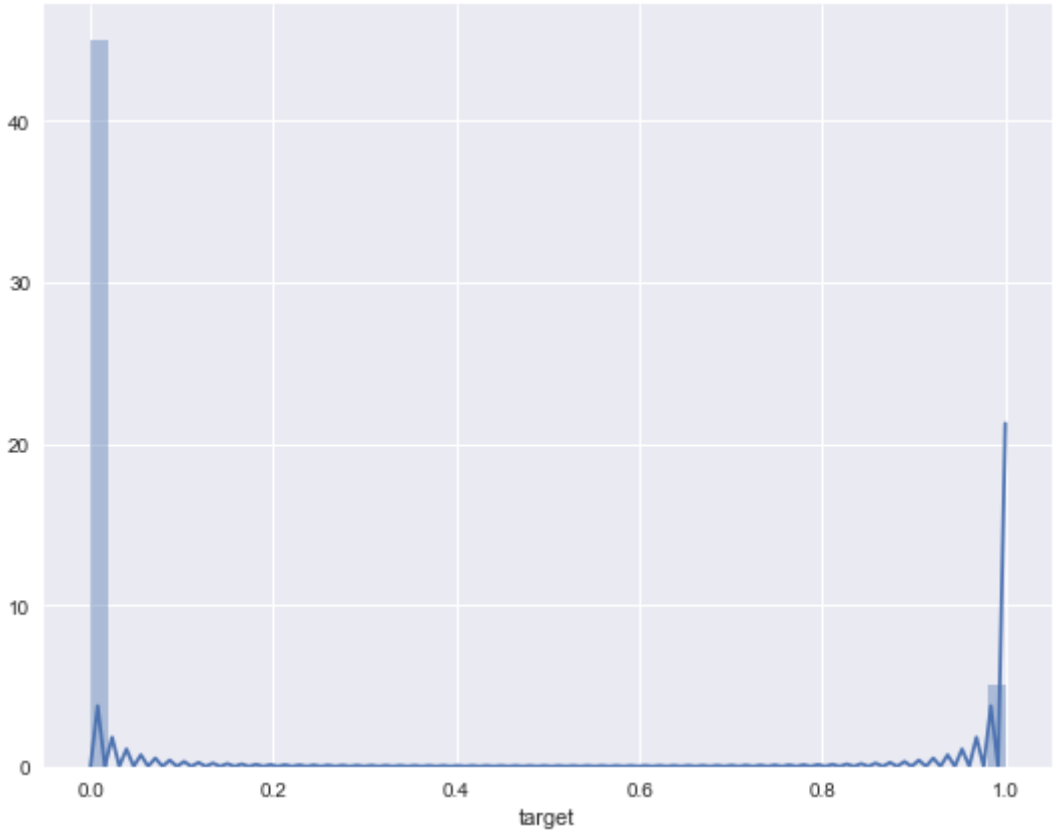
```
1 sns.countplot(transacoesTrain['target'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x110fe58d0>
```




```
1 #Mostra que o target é desbalanceado
2 sns.set(rc={'figure.figsize':(9,7)})
3 sns.distplot(transacoesTrain['target']);
```

```
/anaconda3/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6462: UserWarning: Th
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



Reamostragem

```
df = transacoesTrain.apply(pd.to_numeric, errors='coerce')
df.fillna(0, inplace=True)
x_train, x_test, y_train, y_test = train_test_split(df.drop('target',
                                                    axis=1),
                                                    df['target'],
                                                    test_size=0.3,
                                                    random_state=42)
```

```
# Iremos dividir a base em target 1 e target 0
count_class_0, count_class_1 = transacoesTrain.target.value_counts()

# Divide
df_class_0 = transacoesTrain[transacoesTrain['target'] == 0]
df_class_1 = transacoesTrain[transacoesTrain['target'] == 1]
```

```
INFO:numexpr.utils:NumExpr defaulting to 2 threads.
```

```
df_class_0.shape , df_class_1.shape
```

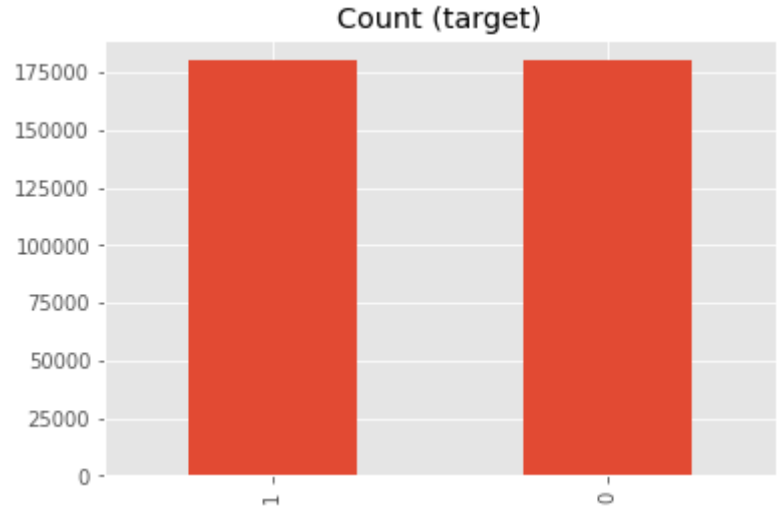
```
((179902, 202), (20098, 202))
```

```
df_class_1_over = df_class_1.sample(count_class_0, replace=True)
df_base_train_balanceada_over = pd.concat([df_class_0, df_class_1_over], axis=0)

print('Random over-sampling:')
print(df_base_train_balanceada_over.target.value_counts())

df_base_train_balanceada_over.target.value_counts().plot(kind='bar', title='Count (targ
```

```
Random over-sampling:
1    179902
0    179902
Name: target, dtype: int64
```



EXECUTANDO O MODELO

```
!apt-get install swig -y
```

```
!pip install Cython numpy

# sometimes you have to run the next command twice on colab
# I haven't figured out why
!pip install auto-sklearn
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  swig3.0
Suggested packages:
  swig-doc swig-examples swig3.0-examples swig3.0-doc
The following NEW packages will be installed:
  swig swig3.0
0 upgraded, 2 newly installed, 0 to remove and 39 not upgraded.
Need to get 1,100 kB of archives.
After this operation, 5,822 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/universe amd64 swig3.0 amd64 3.0.12-
Get:2 http://archive.ubuntu.com/ubuntu bionic/universe amd64 swig amd64 3.0.12-1 [
Fetched 1,100 kB in 1s (766 kB/s)
Selecting previously unselected package swig3.0.
(Reading database ... 160837 files and directories currently installed.)
Preparing to unpack .../swig3.0_3.0.12-1_amd64.deb ...
Unpacking swig3.0 (3.0.12-1) ...
Selecting previously unselected package swig.
Preparing to unpack .../swig_3.0.12-1_amd64.deb ...
Unpacking swig (3.0.12-1) ...
Setting up swig3.0 (3.0.12-1) ...
Setting up swig (3.0.12-1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Requirement already satisfied: Cython in /usr/local/lib/python3.7/dist-packages (0
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (1.
Collecting auto-sklearn
  Downloading https://files.pythonhosted.org/packages/a5/1b/9249f7d4498cbdb0130352
     |████████████████████████████████| 6.1MB 5.9MB/s
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: numpy>=1.9.0 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: scipy>=0.14.1 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (f
Collecting scikit-learn<0.25.0,>=0.24.0
  Downloading https://files.pythonhosted.org/packages/a8/eb/a48f25c967526b66d5f1fa
     |████████████████████████████████| 22.3MB 58.0MB/s
Requirement already satisfied: dask in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: distributed>=2.2.0 in /usr/local/lib/python3.7/dist
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: pandas>=1.0 in /usr/local/lib/python3.7/dist-packag
Collecting liac-arff
  Downloading https://files.pythonhosted.org/packages/6e/43/73944aa5ad2b3185c0f0ba
Collecting ConfigSpace<0.5,>=0.4.14
  Downloading https://files.pythonhosted.org/packages/3e/19/726bcb262949ec28c71ba5
     |████████████████████████████████| 4.2MB 30.6MB/s
Collecting pynisher>=0.6.3
  Downloading https://files.pythonhosted.org/packages/8d/39/edac9acf3bd245ecf47515
Collecting pyrfr<0.9,>=0.8.1
  Downloading https://files.pythonhosted.org/packages/8b/1a/56b630c949e942d12f4ad5
     |████████████████████████████████| 4.0MB 33.1MB/s
Collecting smac<0.14,>=0.13.1
  Downloading https://files.pythonhosted.org/packages/ef/f2/8ea040eaa2253a3606472b
     |████████████████████████████████| 266kB 45.1MB/s
```

```
Collecting threadpoolctl>=2.0.0
  Downloading https://files.pythonhosted.org/packages/c6/e8/c216b9b60cbba4642d3ca1
Requirement already satisfied: fsspec>=0.6.0 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: toolz>=0.8.2 in /usr/local/lib/python3.7/dist-packa
```

```
#!pip install dask distributed
!pip install dask[complete] distributed --upgrade

```

```
from scipy import stats
import sklearn.datasets
from sklearn.metrics import accuracy_score
from sklearn import svm
import sklearn.model_selection
import sklearn.datasets
import sklearn.metrics
import autosklearn.classification
```

```
df = df_base_train_balanceada_over.apply(pd.to_numeric, errors='coerce')
df.fillna(0, inplace=True)
x_train, x_test, y_train, y_test = train_test_split(df.drop('target',
                                                    axis=1),
                                                    df['target'],
                                                    test_size=0.3,
                                                    random_state=42)
```

```
# configure auto-sklearn
automl = autosklearn.classification.AutoSklearnClassifier(
          time_left_for_this_task=18000, # execute o auto-sklearn por no máximo x segun
          per_run_time_limit=10800, #, gastar no máximo Y segundos para cada modelo de
          include_preprocessors=["no_preprocessing"],
          delete_output_folder_after_terminate=False,
          delete_tmp_folder_after_terminate=False
          )
```

```
# train model(s)
automl.fit(x_train, y_train)

# evaluate
y_hat = automl.predict(x_test)
test_acc = sklearn.metrics.accuracy_score(y_test, y_hat)
test_report = sklearn.metrics.classification_report(y_test, y_hat)
test_matrix = sklearn.metrics.confusion_matrix(y_test, y_hat)
print("-------------------------------------------------------------------------")
print("Test Accuracy score {0}".format(test_acc))
print("-------------------------------------------------------------------------")
print("Test Report score {0}".format(test_report))
print("-------------------------------------------------------------------------")
print("Test Confusion Matrix {0}".format(test_matrix))
print("-------------------------------------------------------------------------")
print(automl.sprint_statistics())
print("-------------------------------------------------------------------------")
```

```
print(automl.show_models())
print("--------------------------------------------------------------------------------")
print(automl.get_models_with_weights())
```

```
[WARNING] [2021-07-16 19:46:50,136:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0

[WARNING] [2021-07-16 19:46:50,136:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,137:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,137:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,137:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,137:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,137:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,137:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,137:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,137:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,140:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,140:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,140:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,140:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,140:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,140:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,140:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,140:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,140:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,140:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,141:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,141:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,141:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,141:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,141:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,141:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,141:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,141:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,141:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,141:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,141:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,142:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,142:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,142:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,142:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,142:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,142:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,142:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,142:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,142:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,142:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,142:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,142:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,143:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,143:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,143:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,143:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,143:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,143:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,143:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,143:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,144:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,145:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,145:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,145:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,146:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
[WARNING] [2021-07-16 19:46:50,146:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
```

```
[WARNING] [2021-07-16 19:46:50,146:Client-AutoMLSMBO(1)::8ad4eb2e-e66e-11eb-828e-0
```

Métricas

```
1 X_train.shape, y_train.shape ,X_test.shape, y_test.shape
```

```
((140000, 200), (140000,), (60000, 200), (60000,))
```

```
1 #Importar o módulo de métricas scikit-learn para o cálculo de precisão
2 from sklearn import metrics
3 # Acuracy , o quanto a arvore bateu
4 metrics.accuracy_score(y_test, y_pred)
```

```
0.8996333333333333
```

```
1 # Para saber quão bem o modelo irá generalizar
2 #ou seja, ela serve para saber se o modelo será efetivo ao receber um dado que ele nunc
3 clf.score(X_train, y_train)
```

```
1.0
```

Matriz de Confusão

```
1 #O Método confusion_matrix retorna a matriz de confusão
2 from sklearn.metrics import confusion_matrix
3 confusion_matrix(y_test,clf.predict(X_test))
4
```

```
array([[53976,     0],
       [ 6022,     2]])
```

```
1 pd.crosstab(y_test,clf.predict(X_test),rownames=['Real'],colnames=['Predito'],margins=T
```

Predito	0	1	All
Real			
0	53976	0	53976
1	6022	2	6024
All	59998	2	60000

```
1
```

```
------------------------------------------------------------------------
Test Accuracy score 0.9102666246688036
------------------------------------------------------------------------
Test Report score               precision    recall  f1-score   support

           0       0.92      0.90      0.91     53982
           1       0.90      0.92      0.91     53960

    accuracy                           0.91    107942
   macro avg       0.91      0.91      0.91    107942
weighted avg       0.91      0.91      0.91    107942

------------------------------------------------------------------------
Test Confusion Matrix [[48649  5333]
 [ 4353 49607]]
------------------------------------------------------------------------
auto-sklearn results:
  Dataset name: 29b090ba-e012-11eb-81fa-0242ac1c0002
  Metric: accuracy
  Best validation score: 0.825949
  Number of target algorithm runs: 54
  Number of successful target algorithm runs: 12
  Number of crashed target algorithm runs: 6
  Number of target algorithms that exceeded the time limit: 5
  Number of target algorithms that exceeded the memory limit: 31

------------------------------------------------------------------------
[(0.220000, SimpleClassificationPipeline({'balancing:strategy': 'weighting', 'classifier:__choice_
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})),
(0.180000, SimpleClassificationPipeline({'balancing:strategy': 'weighting', 'classifier:__choice__
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})),
(0.140000, SimpleClassificationPipeline({'balancing:strategy': 'none', 'classifier:__choice__': 'p
dataset_properties={
  'task': 1,
  'sparse': False,
```

```
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})),
(0.120000, SimpleClassificationPipeline({'balancing:strategy': 'weighting', 'classifier:__choice__
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})),
(0.100000, SimpleClassificationPipeline({'balancing:strategy': 'none', 'classifier:__choice__': 'c
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})),
(0.100000, SimpleClassificationPipeline({'balancing:strategy': 'none', 'classifier:__choice__': 'c
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})),
(0.080000, SimpleClassificationPipeline({'balancing:strategy': 'none', 'classifier:__choice__': 'c
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})),
(0.040000, SimpleClassificationPipeline({'balancing:strategy': 'weighting', 'classifier:__choice__
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})),
(0.020000, SimpleClassificationPipeline({'balancing:strategy': 'none', 'classifier:__choice__': 'c
dataset_properties={
  'task': 1,
```

```
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})),
]
--------------------------------------------------------------------------------
[(0.22, SimpleClassificationPipeline({'balancing:strategy': 'weighting', 'classifier:__choice__':
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})), (0.18, SimpleClassificationPipeline({'balancing:strategy': 'weighting', 'cla
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})), (0.14, SimpleClassificationPipeline({'balancing:strategy': 'none', 'classifi
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})), (0.12, SimpleClassificationPipeline({'balancing:strategy': 'weighting', 'cla
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})), (0.1, SimpleClassificationPipeline({'balancing:strategy': 'none', 'classifie
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})), (0.1, SimpleClassificationPipeline({'balancing:strategy': 'none', 'classifie
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
```

```
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})), (0.08, SimpleClassificationPipeline({'balancing:strategy': 'none', 'classif
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})), (0.04, SimpleClassificationPipeline({'balancing:strategy': 'weighting', 'cla
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False})), (0.02, SimpleClassificationPipeline({'balancing:strategy': 'none', 'classif
dataset_properties={
  'task': 1,
  'sparse': False,
  'multilabel': False,
  'multiclass': False,
  'target_type': 'classification',
  'signed': False}))]
```