Hackthon 6

Hugo Castilho

2018-01-06

Contents

List of Figures

List of Tables

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

1 Exploratory Data Analysis

The dataset has 8164 samples, each property was analysed for insights and anomalies. The following descriptions were inferred using the name of the property in the dataset file and the type and values of the data.

id Integer, identifier for each subject, all values distinct no problems detected.

target Integer, did the subject become unemployed in the following 12 months.

birth date Date (YYYY-MM-DD), date of birth of the subject. The youngest being 2016-02-10 and the oldest 1928-01-09. Frequency plot (see fig. 1 on page 9) shows nothing surprising.

country of origin Categorical, country names come in a variety of inconsistent formats.

domestic relationship type Categorical, (see table 3 on page 6) who the subject lives with. Categories are unclear and ill defined. Moreover it is inconsistent with the *domestic status*, there are a considerable number of entries classified as *domestic relation type–never married* and *domestic status–d* (presumably divorced) (see table 5 on page 6).

domestic status Categorical, marital status or if has married several times (see table 4 on page 6). By elimination category *d* is supposedly divorced.

earned dividends Numerical, monetary amount (currency not specified). Return from distribution of corporate earnings, it's 0 for all samples.

ethnicity Categorical, categories have funny names (see table 6 on page 7).

gender Categorical, all female dataset (see table 7 on page 7).

job type Categorical, current job type of the subject (see table 8 on page 7), government, self employed, item etc...

interest earned Numerical, monetary amount (currency not specified). Returns from loaning money (see fig. 2 on page 10).

monthly work Numerical, number of hours of work per month (see fig. 3 on page 11)

profession Categorical, type of profession (see table 9 on page 7).

school level Categorical, subject level of schooling (see table 10 on page 8).

2 Data pre-processing

This section details any and all data pre-processing before modelling. The first section section 2.1 on the next page explains what was done to convert the dataset to usable, unambiguous types. Afterwards section section 2.2 on the following page details the what was done to select and improve the features feed to the model. Finally section section 2.3 on the next page breaks down the train/test split.

2.1 Data cleaning

To input in the models birth dates where converted to timestamps.

On importing the dataset all countries where converted to the corresponding *ISO 3166–1 alpha-2* representation. Some inputs required special rules. Especially ambiguous was *dr* which represents no country code this was converted to Dominican Republic (DO) even if the race for these inputs suggests it's not (mostly white).

For categorical data, the classes where kept as is and turned into dummy class variables.

2.2 Feature engineering

In the dataset earned dividends and gender do not change, these properties where dropped since they convey no useful information. If new samples include this value this decision will be reconsidered.

2.3 Train/test splitting

The train/test split was done holding out .4 of the data for final validation (train_test_split). Furthermore training was done using a shuffle split (*ShuffleSplitCV*) with .3 for testing.

3 Modelling

To get a feelling of the baseline performance of the models available in the *scikit* package severall where tried with the default parameters. The train/test split is described in section 2.3. The results are in table 1, a the exact parameters passet to the models is present in **??** on page ??

Table 1:

Model	AUC ROC Score
QuadraticDiscriminantAnalysis	0.8053
SVC	0.6698
SGDClassifier	0.6377
KNeighborsClassifier	0.7267
GaussianProcessClassifier	0.7022
GaussianNB	0.6718
DecisionTreeClassifier	0.6729
BaggingClassifier	0.8428
RandomForestClassifier	0.8160
ExtraTreesClassifier	0.8154
AdaBoostClassifier	0.9008
GradientBoostingClassifier	0.9212
VotingClassifier	0.8233

4 Evaluation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

A EDA Tables

Table 2: Country of origin value counts.

u.s.	7330
unknown	126
mexico	111
philippines	60
de	50
puerto rico	39
jamaica	34
cuba	34
el-salvador	30
canada	28
dr	27
gb	22
south	20
italy	18
columbia	17
haiti	17
china	17
vietnam	17
guatemala	16
japan	15
poland	14
peru	11
taiwan	11
thailand	11
fr	9
trinadad/tobago	8
india	7
nicaragua	7
portugal	6
honduras	6
laos	6
ecuador	6
iran	6
ireland	5
us territory	4
hong	4
scotland	3
hungary	3
greece	3
yugoslavia	3
cambodia	2
netherlands	1

Table 3: Domestic relationship type value counts.

not living with family	2919
never married	2063
living with child	1750
has husband	1106
living with extende family	325
has wife	1

Table 4: Domestic status value counts.

single	3662
d	2073
married 2	1170
spouse passed	599
divorce pending	486
married not together	163
married 1	11

Table 5: Domestic relationship type grouped by domestic status counts.

domestic status	domestic relationship type	count
d	living with child	116
	living with extende family	47
	never married	1006
	not living with family	904
divorce pending	living with child	40
	living with extende family	25
	never married	293
	not living with family	128
married 1	has husband	10
	living with child	1
married 2	has husband	1096
	has wife	1
	living with child	28
	living with extende family	42
	not living with family	3
married not together	living with child	25
	living with extende family	7
	never married	73
	not living with family	58
single	living with child	1530
	living with extende family	174
	never married	449
	not living with family	1509
spouse passed	living with child	10
	living with extende family	30
	never married	242
	not living with family	317

Table 6: Ethnicity value counts.

white and privileged	6523
afro american	1210
asian	262
american indian	88
other	81

Table 7: Gender value counts.

Female	8164

Table 8: Job type value counts.

private	5919
unknown	620
local-gov	618
state-gov	368
self-emp-not-inc	303
federal-gov	236
self-emp-inc	94
without-pay	4
never-worked	2

Table 9: Profession value counts.

secretarial	1949
other	1423
specialist technician	1096
sales	978
C-level	842
unknown	622
mechanic	420
technology support	247
vocational	184
household labor	131
estate employee	108
defense contractor	58
trucking	58
agriculture	48

Table 10: School level value counts.

secondary	2594
entry level college	2165
college graduate	1188
basic vocational	373
some post graduate	355
secondary 11	341
advanced vocational	326
10th	248
secondary-7 through 8	123
secondary 12	106
secondary-9	104
secondary-5 through 6	72
advanced post graduate	61
primary school	58
primary 1 through 4	37
kindergarten	13

B EDA Figures

Figure 1: Date of birth frequency.

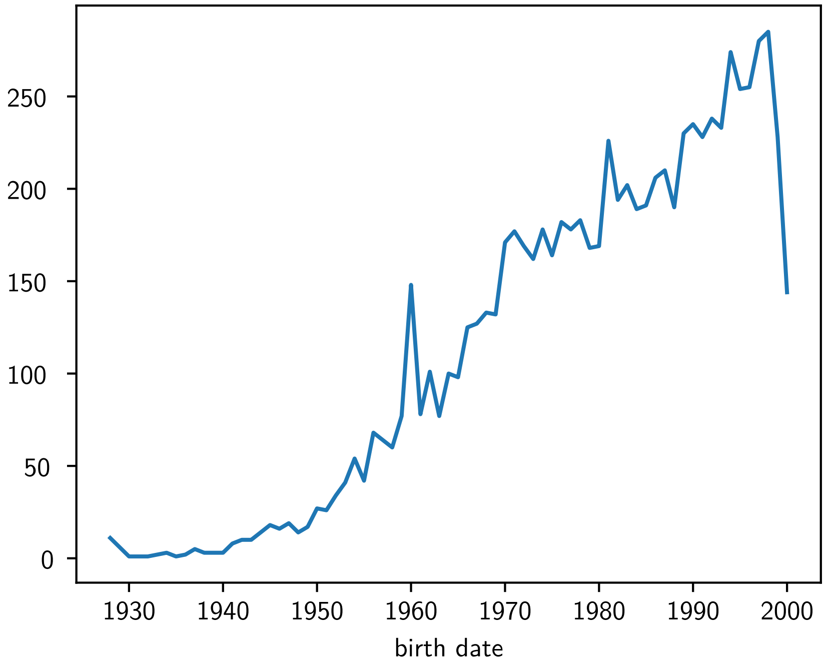


Figure 2: Interest earned frequency (logarithmic scale).

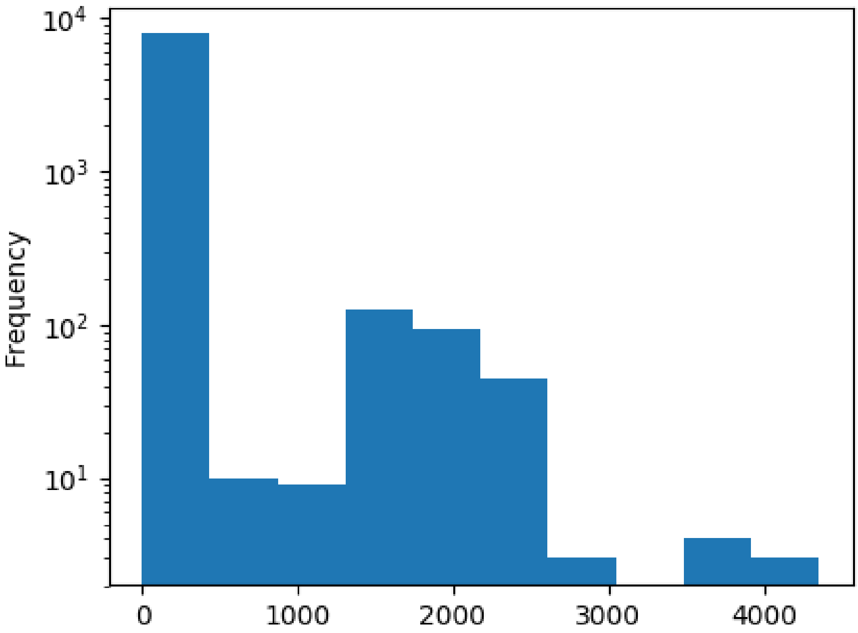
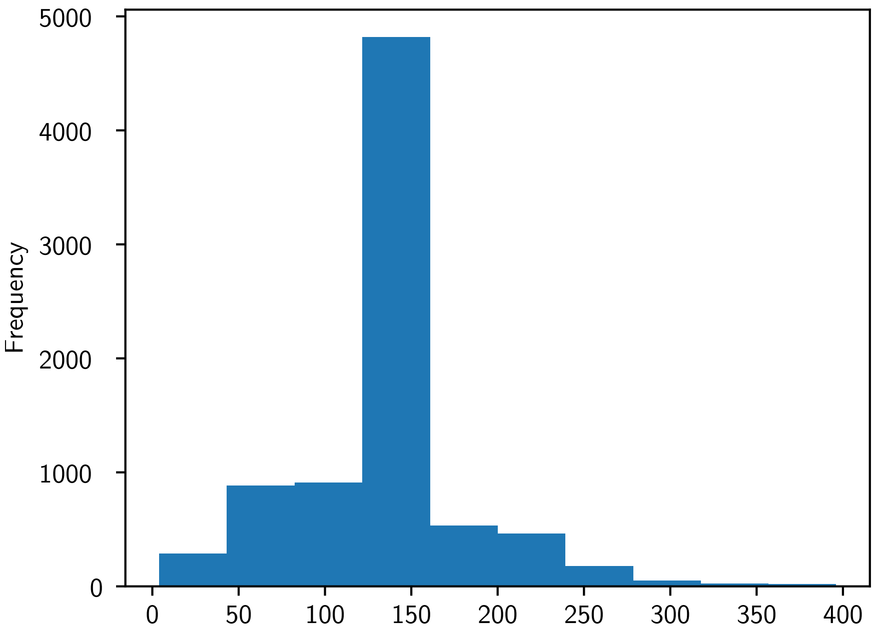


Figure 3: Monthly work frequency.



C Modelling

C.1 Initial model tests params

QuadraticDiscriminantAnalysis

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': QuadraticDiscriminantAnalysis(priors=None, reg_param=0.0,
               store_covariance=False, store_covariances=None, tol=0.0001),
 'estimator__priors': None,
 'estimator__reg_param': 0.0,
 'estimator__store_covariance': False,
 'estimator__store_covariances': None,
 'estimator__tol': 0.0001,
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```

SVC

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False),
 'estimator__C': 1.0,
 'estimator__cache_size': 200,
 'estimator__class_weight': None,
 'estimator__coef0': 0.0,
 'estimator__decision_function_shape': 'ovr',
 'estimator__degree': 3,
 'estimator__gamma': 'auto',
 'estimator__kernel': 'rbf',
 'estimator__max_iter': -1,
 'estimator__probability': False,
 'estimator__random_state': None,
 'estimator__shrinking': True,
 'estimator__tol': 0.001,
 'estimator__verbose': False,
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {'probability': [True]},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```

SGDClassifier

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': SGDClassifier(alpha=0.0001, average=False, class_weight=None, epsilon=0.1,
       eta0=0.0, fit_intercept=True, l1_ratio=0.15,
       learning_rate='optimal', loss='hinge', max_iter=None, n_iter=None,
       n_jobs=1, penalty='l2', power_t=0.5, random_state=None,
       shuffle=True, tol=None, verbose=0, warm_start=False),
 'estimator__alpha': 0.0001,
 'estimator__average': False,
 'estimator__class_weight': None,
 'estimator__epsilon': 0.1,
 'estimator__eta0': 0.0,
 'estimator__fit_intercept': True,
 'estimator__l1_ratio': 0.15,
 'estimator__learning_rate': 'optimal',
 'estimator__loss': 'hinge',
 'estimator__max_iter': None,
 'estimator__n_iter': None,
 'estimator__n_jobs': 1,
 'estimator__penalty': 'l2',
 'estimator__power_t': 0.5,
 'estimator__random_state': None,
 'estimator__shuffle': True,
 'estimator__tol': None,
 'estimator__verbose': 0,
 'estimator__warm_start': False,
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {'loss': ['log']},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```

KNeighborsClassifier

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=5, p=2,
          weights='uniform'),
 'estimator__algorithm': 'auto',
 'estimator__leaf_size': 30,
 'estimator__metric': 'minkowski',
 'estimator__metric_params': None,
 'estimator__n_jobs': 1,
 'estimator__n_neighbors': 5,
 'estimator__p': 2,
 'estimator__weights': 'uniform',
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```

GaussianProcessClassifier

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': GaussianProcessClassifier(copy_X_train=True, kernel=None,
             max_iter_predict=100, multi_class='one_vs_rest', n_jobs=1,
             n_restarts_optimizer=0, optimizer='fmin_l_bfgs_b',
             random_state=None, warm_start=False),
 'estimator__copy_X_train': True,
 'estimator__kernel': None,
 'estimator__max_iter_predict': 100,
 'estimator__multi_class': 'one_vs_rest',
 'estimator__n_jobs': 1,
 'estimator__n_restarts_optimizer': 0,
 'estimator__optimizer': 'fmin_l_bfgs_b',
 'estimator__random_state': None,
 'estimator__warm_start': False,
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```

GaussianNB

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': GaussianNB(priors=None),
 'estimator__priors': None,
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```

DecisionTreeClassifier

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best'),
 'estimator__class_weight': None,
 'estimator__criterion': 'gini',
 'estimator__max_depth': None,
 'estimator__max_features': None,
 'estimator__max_leaf_nodes': None,
 'estimator__min_impurity_decrease': 0.0,
 'estimator__min_impurity_split': None,
 'estimator__min_samples_leaf': 1,
 'estimator__min_samples_split': 2,
 'estimator__min_weight_fraction_leaf': 0.0,
 'estimator__presort': False,
 'estimator__random_state': None,
 'estimator__splitter': 'best',
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```

BaggingClassifier

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': BaggingClassifier(base_estimator=None, bootstrap=True,
         bootstrap_features=False, max_features=1.0, max_samples=1.0,
         n_estimators=10, n_jobs=1, oob_score=False, random_state=None,
         verbose=0, warm_start=False),
 'estimator__base_estimator': None,
 'estimator__bootstrap': True,
 'estimator__bootstrap_features': False,
 'estimator__max_features': 1.0,
 'estimator__max_samples': 1.0,
 'estimator__n_estimators': 10,
 'estimator__n_jobs': 1,
 'estimator__oob_score': False,
 'estimator__random_state': None,
 'estimator__verbose': 0,
 'estimator__warm_start': False,
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```

RandomForestClassifier

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False),
 'estimator__bootstrap': True,
 'estimator__class_weight': None,
 'estimator__criterion': 'gini',
 'estimator__max_depth': None,
 'estimator__max_features': 'auto',
 'estimator__max_leaf_nodes': None,
 'estimator__min_impurity_decrease': 0.0,
 'estimator__min_impurity_split': None,
 'estimator__min_samples_leaf': 1,
 'estimator__min_samples_split': 2,
 'estimator__min_weight_fraction_leaf': 0.0,
 'estimator__n_estimators': 10,
 'estimator__n_jobs': 1,
 'estimator__oob_score': False,
 'estimator__random_state': None,
 'estimator__verbose': 0,
 'estimator__warm_start': False,
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```

ExtraTreesClassifier

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': ExtraTreesClassifier(bootstrap=False, class_weight=None, criterion='gini',
           max_depth=None, max_features='auto', max_leaf_nodes=None,
           min_impurity_decrease=0.0, min_impurity_split=None,
           min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
           oob_score=False, random_state=None, verbose=0, warm_start=False),
 'estimator__bootstrap': False,
 'estimator__class_weight': None,
 'estimator__criterion': 'gini',
 'estimator__max_depth': None,
 'estimator__max_features': 'auto',
 'estimator__max_leaf_nodes': None,
 'estimator__min_impurity_decrease': 0.0,
 'estimator__min_impurity_split': None,
 'estimator__min_samples_leaf': 1,
 'estimator__min_samples_split': 2,
 'estimator__min_weight_fraction_leaf': 0.0,
 'estimator__n_estimators': 10,
 'estimator__n_jobs': 1,
 'estimator__oob_score': False,
 'estimator__random_state': None,
 'estimator__verbose': 0,
 'estimator__warm_start': False,
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```

AdaBoostClassifier

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None,
          learning_rate=1.0, n_estimators=50, random_state=None),
 'estimator__algorithm': 'SAMME.R',
 'estimator__base_estimator': None,
 'estimator__learning_rate': 1.0,
 'estimator__n_estimators': 50,
 'estimator__random_state': None,
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```

GradientBoostingClassifier

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': GradientBoostingClassifier(criterion='friedman_mse', init=None,
              learning_rate=0.1, loss='deviance', max_depth=3,
              max_features=None, max_leaf_nodes=None,
              min_impurity_decrease=0.0, min_impurity_split=None,
              min_samples_leaf=1, min_samples_split=2,
              min_weight_fraction_leaf=0.0, n_estimators=100,
              presort='auto', random_state=None, subsample=1.0, verbose=0,
              warm_start=False),
 'estimator__criterion': 'friedman_mse',
 'estimator__init': None,
 'estimator__learning_rate': 0.1,
 'estimator__loss': 'deviance',
 'estimator__max_depth': 3,
 'estimator__max_features': None,
 'estimator__max_leaf_nodes': None,
 'estimator__min_impurity_decrease': 0.0,
 'estimator__min_impurity_split': None,
 'estimator__min_samples_leaf': 1,
 'estimator__min_samples_split': 2,
 'estimator__min_weight_fraction_leaf': 0.0,
 'estimator__n_estimators': 100,
 'estimator__presort': 'auto',
 'estimator__random_state': None,
 'estimator__subsample': 1.0,
 'estimator__verbose': 0,
 'estimator__warm_start': False,
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```

VotingClassifier

```
{'cv': ShuffleSplit(n_splits=5, random_state=1, test_size=0.3, train_size=None),
 'error_score': 'raise',
 'estimator': VotingClassifier(estimators=[('clf1', LogisticRegression(C=1.0, class_weight
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)), ('clf2', Ran...se, random_state=None, verbose=0,
            warm_start=False)), ('clf3', GaussianNB(priors=None))],
         flatten_transform=None, n_jobs=1, voting='soft', weights=None),
 'estimator__clf1': LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False),
 'estimator__clf1__C': 1.0,
 'estimator__clf1__class_weight': None,
 'estimator__clf1__dual': False,
 'estimator__clf1__fit_intercept': True,
 'estimator__clf1__intercept_scaling': 1,
 'estimator__clf1__max_iter': 100,
 'estimator__clf1__multi_class': 'ovr',
 'estimator__clf1__n_jobs': 1,
 'estimator__clf1__penalty': 'l2',
 'estimator__clf1__random_state': None,
 'estimator__clf1__solver': 'liblinear',
 'estimator__clf1__tol': 0.0001,
 'estimator__clf1__verbose': 0,
 'estimator__clf1__warm_start': False,
 'estimator__clf2': RandomForestClassifier(bootstrap=True, class_weight=None, criterion='g
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False),
 'estimator__clf2__bootstrap': True,
 'estimator__clf2__class_weight': None,
 'estimator__clf2__criterion': 'gini',
 'estimator__clf2__max_depth': None,
 'estimator__clf2__max_features': 'auto',
 'estimator__clf2__max_leaf_nodes': None,
 'estimator__clf2__min_impurity_decrease': 0.0,
 'estimator__clf2__min_impurity_split': None,
 'estimator__clf2__min_samples_leaf': 1,
 'estimator__clf2__min_samples_split': 2,
 'estimator__clf2__min_weight_fraction_leaf': 0.0,
 'estimator__clf2__n_estimators': 10,
 'estimator__clf2__n_jobs': 1,
 'estimator__clf2__oob_score': False,
 'estimator__clf2__random_state': None,
 'estimator__clf2__verbose': 0,
 'estimator__clf2__warm_start': False,
 'estimator__clf3': GaussianNB(priors=None),
 'estimator__clf3__priors': None,
 'estimator__estimators': [('clf1',
                            LogisticRegression(C=1.0, class_weight=None, dual=False, fit_i
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)),
                           ('clf2',
                            RandomForestClassifier(bootstrap=True, class_weight=None, crit
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)),
                           ('clf3', GaussianNB(priors=None))],
 'estimator__flatten_transform': None,
 'estimator__n_jobs': 1,
 'estimator__voting': 'soft',
 'estimator__weights': None,
 'fit_params': None,
 'iid': True,
 'n_jobs': 3,
 'param_grid': {},
 'pre_dispatch': '2*n_jobs',
 'refit': True,
 'return_train_score': 'warn',
 'scoring': 'roc_auc',
 'verbose': 0}
```