Demo 12-12-2018

December 12, 2018

Logit, logit rbf black 20 y logit rbf grey 20 con todos los datasets

```
In [1]: %%javascript
        IPython.OutputArea.prototype._should_scroll = function(lines) {
            return false;
        }
```

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

```
In [2]: import warnings
        warnings.filterwarnings('ignore')
```

```
In [3]: from demo_utils.demo0 import Demo0
        #from demo_utils.general import SUPPORTED_DATASETS
```

0.1 Logit | RFF | No PCA

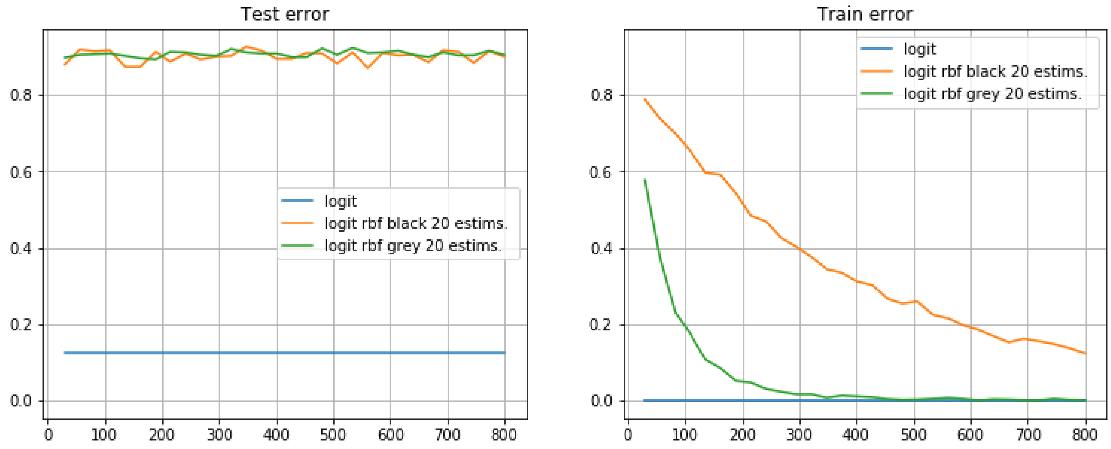
```
In [4]: model1 = {
            'model_name': 'logit',
            'sampler_name': 'identity',
            'box_type': 'none',
            'n_estim': None,
            'pca': False
        }
        model2 = {
            'model_name': 'logit',
            'sampler_name': 'rbf',
            'box_type': 'black',
            'n_estim': 20,
            'pca': False
        }
        model3 = {
            'model_name': 'logit',
            'sampler_name': 'rbf',
            'box_type': 'grey',
            'n_estim': 20,
            'pca': False
        }

        info = {
            'models': [model1, model2, model3],
            'features_range': (30, 800),
            'dts_size': 2000,
            'dts_name': 'mnist'
        }
```

```
In [5]: Demo0().non_interactive(**info)
```

1 Una demo genérica

- Dataset: **mnist**
- Size: **2000**



1.1 Logit | RFF | Sí PCA

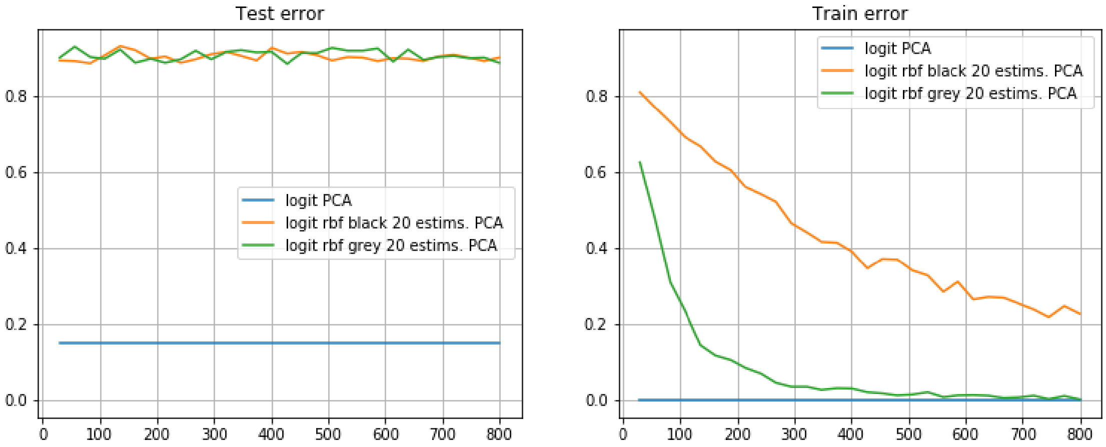
```
In [6]: model1 = {
            'model_name': 'logit',
            'sampler_name': 'identity',
            'box_type': 'none',
            'n_estim': None,
            'pca': True
        }
        model2 = {
            'model_name': 'logit',
            'sampler_name': 'rbf',
            'box_type': 'black',
            'n_estim': 20,
            'pca': True
        }
        model3 = {
            'model_name': 'logit',
            'sampler_name': 'rbf',
            'box_type': 'grey',
            'n_estim': 20,
            'pca': True
        }

        info = {
            'models': [model1, model2, model3],
            'features_range': (30, 800),
            'dts_size': 2000,
            'dts_name': 'mnist'
        }
```

```
In [7]: Demo0().non_interactive(**info)
```

2 Una demo genérica

- Dataset: **mnist**
- Size: **2000**



2.1 Logit | Nystroem | No PCA

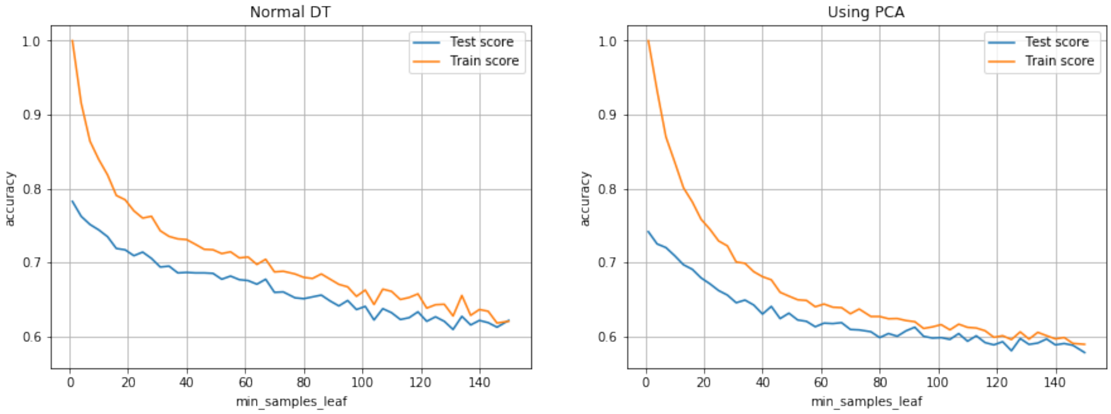
```
In [8]: model1 = {
            'model_name': 'logit',
            'sampler_name': 'identity',
            'box_type': 'none',
            'n_estim': None,
            'pca': False
}
model2 = {
    'model_name': 'logit',
    'sampler_name': 'nystroem',
    'box_type': 'black',
    'n_estim': 20,
    'pca': False
}
model3 = {
    'model_name': 'logit',
    'sampler_name': 'nystroem',
    'box_type': 'grey',
    'n_estim': 20,
    'pca': False
}

info = {
    'models': [model1, model2, model3],
    'features_range': (30, 800),
    'dts_size': 2000,
    'dts_name': 'mnist'
}
```

```
In [9]: Demo0().non_interactive(**info)
```

3 Una demo genérica

- Dataset: **mnist**
- Size: **2000**



3.1 Logit | Nystroem | Sí PCA

```
In [10]: model1 = {
             'model_name': 'logit',
             'sampler_name': 'identity',
             'box_type': 'none',
             'n_estim': None,
             'pca': True
         }
         model2 = {
             'model_name': 'logit',
             'sampler_name': 'nystroem',
             'box_type': 'black',
             'n_estim': 20,
             'pca': True
         }
         model3 = {
             'model_name': 'logit',
             'sampler_name': 'nystroem',
             'box_type': 'grey',
             'n_estim': 20,
             'pca': True
         }

         info = {
             'models': [model1, model2, model3],
             'features_range': (30, 800),
             'dts_size': 2000,
             'dts_name': 'mnist'
         }
```

```
In [11]: Demo0().non_interactive(**info)
```

4 Una demo genérica

```
    ---------------------------------------------------------------------------

    LinAlgError                               Traceback (most recent call last)

    <ipython-input-11-0429739dc21c> in <module>
----> 1 Demo0().non_interactive(**info)


    ~/git/TFG/code/notebooks/python/demo_utils/generic_demo.py in non_interactive(self, **a
     40     def non_interactive(self, **argw):
     41         display(md(self.desc))
---> 42         train_scores, test_scores = self.run_demo(**argw)
```

```
     43         fig = self.get_generic_graph_from_scores(train_scores, test_scores)
     44         # fig.suptitle(self.title)


    ~/git/TFG/code/notebooks/python/demo_utils/demo0.py in run_demo(self, dts_name, dts_siz
    234                 # train_score y test_score son diccionarios
    235                 train_score, test_score =\
--> 236                     get_sampling_model_scores(clf, dataset, features)
    237                 lab = self.get_label(model_name, sampler_name, box_type,
    238                                      n_estim, pca)


    ~/git/TFG/code/notebooks/python/demo_utils/learning.py in get_sampling_model_scores(cl
    277         else:
    278             clf.set_params(base_estimator__sampler__n_components=f)
--> 279         clf.fit(data_train, target_train)
    280         train_score = clf.score(data_train, target_train)
    281         test_score = clf.score(data_test, target_test)


    ~/.local/lib/python3.6/site-packages/sklearn/pipeline.py in fit(self, X, y, **fit_param
    263             This estimator
    264         """
--> 265         Xt, fit_params = self._fit(X, y, **fit_params)
    266         if self._final_estimator is not None:
    267             self._final_estimator.fit(Xt, y, **fit_params)


    ~/.local/lib/python3.6/site-packages/sklearn/pipeline.py in _fit(self, X, y, **fit_para
    228                 Xt, fitted_transformer = fit_transform_one_cached(
    229                     cloned_transformer, Xt, y, None,
--> 230                     **fit_params_steps[name])
    231                 # Replace the transformer of the step with the fitted
    232                 # transformer. This is necessary when loading the transformer


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/memory.py in __call__(sel
    327 
    328     def __call__(self, *args, **kwargs):
--> 329         return self.func(*args, **kwargs)
    330 
    331     def call_and_shelve(self, *args, **kwargs):


    ~/.local/lib/python3.6/site-packages/sklearn/pipeline.py in _fit_transform_one(transfo
    612 def _fit_transform_one(transformer, X, y, weight, **fit_params):
    613     if hasattr(transformer, 'fit_transform'):
--> 614         res = transformer.fit_transform(X, y, **fit_params)
```

```
    615     else:
    616         res = transformer.fit(X, y, **fit_params).transform(X)


    ~/.local/lib/python3.6/site-packages/sklearn/decomposition/pca.py in fit_transform(sel
    357
    358         """
--> 359         U, S, V = self._fit(X)
    360         U = U[:, :self.n_components_]
    361


    ~/.local/lib/python3.6/site-packages/sklearn/decomposition/pca.py in _fit(self, X)
    404         # Call different fits for either full or truncated SVD
    405         if self._fit_svd_solver == 'full':
--> 406             return self._fit_full(X, n_components)
    407         elif self._fit_svd_solver in ['arpack', 'randomized']:
    408             return self._fit_truncated(X, n_components, self._fit_svd_solver)


    ~/.local/lib/python3.6/site-packages/sklearn/decomposition/pca.py in _fit_full(self, X
    435         X -= self.mean_
    436
--> 437         U, S, V = linalg.svd(X, full_matrices=False)
    438         # flip eigenvectors' sign to enforce deterministic output
    439         U, V = svd_flip(U, V)


    ~/.local/lib/python3.6/site-packages/scipy/linalg/decomp_svd.py in svd(a, full_matrices
    130
    131     if info > 0:
--> 132         raise LinAlgError("SVD did not converge")
    133     if info < 0:
    134         raise ValueError('illegal value in %d-th argument of internal gesdd'


    LinAlgError: SVD did not converge
```

4.1 LinearSVC | RFF | No PCA

```
In [12]: model1 = {
             'model_name': 'linear_svc',
             'sampler_name': 'identity',
             'box_type': 'none',
             'n_estim': None,
             'pca': False
          }
```

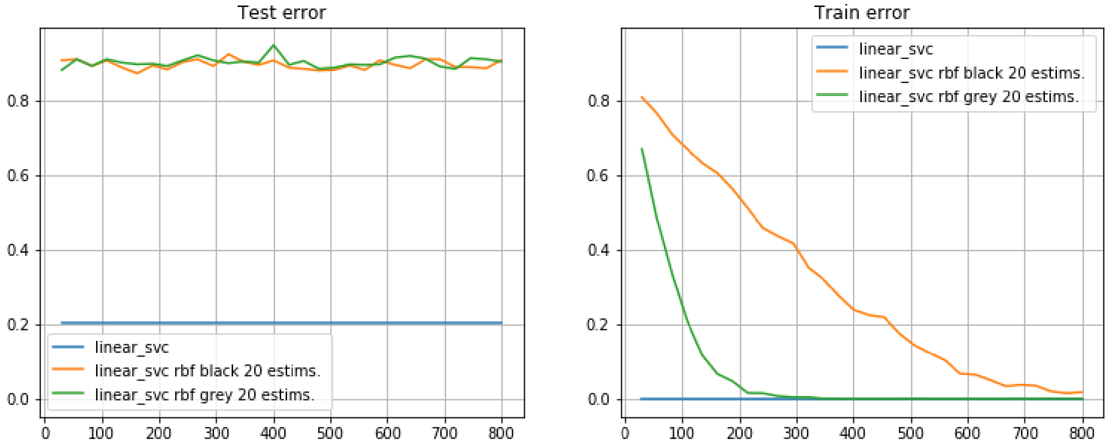
```
model2 = {
    'model_name': 'linear_svc',
    'sampler_name': 'rbf',
    'box_type': 'black',
    'n_estim': 20,
    'pca': False
}
model3 = {
    'model_name': 'linear_svc',
    'sampler_name': 'rbf',
    'box_type': 'grey',
    'n_estim': 20,
    'pca': False
}

info = {
    'models': [model1, model2, model3],
    'features_range': (30, 800),
    'dts_size': 2000,
    'dts_name': 'mnist'
}
```

```
In [13]: Demo0().non_interactive(**info)
```

5 Una demo genérica

- Dataset: **mnist**
- Size: **2000**



5.1 LinearSVC | RFF | Sí PCA

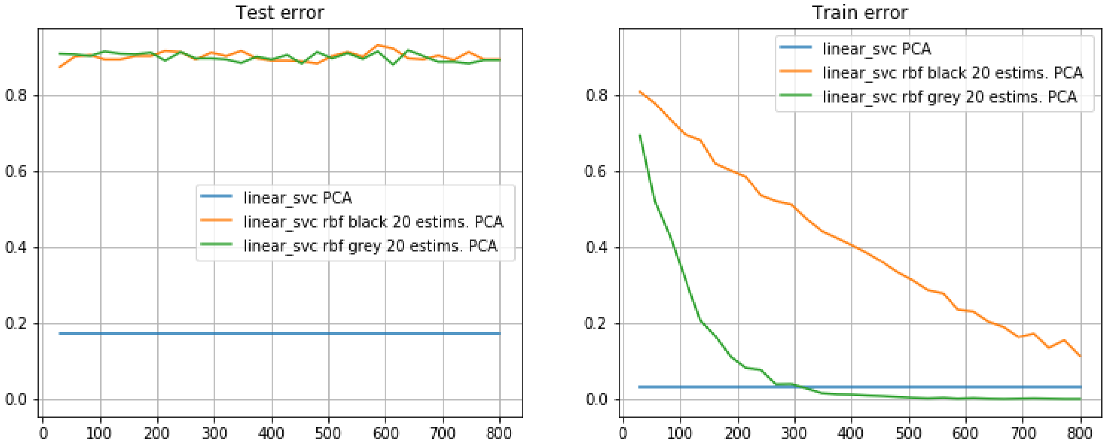
```
In [14]: model1 = {
             'model_name': 'linear_svc',
             'sampler_name': 'identity',
             'box_type': 'none',
             'n_estim': None,
             'pca': True
         }
         model2 = {
             'model_name': 'linear_svc',
             'sampler_name': 'rbf',
             'box_type': 'black',
             'n_estim': 20,
             'pca': True
         }
         model3 = {
             'model_name': 'linear_svc',
             'sampler_name': 'rbf',
             'box_type': 'grey',
             'n_estim': 20,
             'pca': True
         }

         info = {
             'models': [model1, model2, model3],
             'features_range': (30, 800),
             'dts_size': 2000,
             'dts_name': 'mnist'
         }
```

```
In [15]: Demo0().non_interactive(**info)
```

6 Una demo genérica

- Dataset: **mnist**
- Size: **2000**



6.1 LinearSVC | Nystroem | No PCA

```
In [16]: model1 = {
             'model_name': 'linear_svc',
             'sampler_name': 'identity',
             'box_type': 'none',
             'n_estim': None,
             'pca': False
         }
         model2 = {
             'model_name': 'linear_svc',
             'sampler_name': 'nystroem',
             'box_type': 'black',
             'n_estim': 20,
             'pca': False
         }
         model3 = {
             'model_name': 'linear_svc',
             'sampler_name': 'nystroem',
             'box_type': 'grey',
             'n_estim': 20,
             'pca': False
         }

         info = {
             'models': [model1, model2, model3],
             'features_range': (30, 800),
             'dts_size': 2000,
             'dts_name': 'mnist'
         }
```

```
In [17]: Demo0().non_interactive(**info)
```

7 Una demo genérica

```
    ---------------------------------------------------------------------------

    LinAlgError                               Traceback (most recent call last)

    <ipython-input-17-0429739dc21c> in <module>
----> 1 Demo0().non_interactive(**info)


    ~/git/TFG/code/notebooks/python/demo_utils/generic_demo.py in non_interactive(self, **a
     40     def non_interactive(self, **argw):
     41         display(md(self.desc))
---> 42         train_scores, test_scores = self.run_demo(**argw)
     43         fig = self.get_generic_graph_from_scores(train_scores, test_scores)
     44         # fig.suptitle(self.title)


    ~/git/TFG/code/notebooks/python/demo_utils/demo0.py in run_demo(self, dts_name, dts_siz
    234                 # train_score y test_score son diccionarios
    235                 train_score, test_score =\
--> 236                     get_sampling_model_scores(clf, dataset, features)
    237                 lab = self.get_label(model_name, sampler_name, box_type,
    238                                      n_estim, pca)


    ~/git/TFG/code/notebooks/python/demo_utils/learning.py in get_sampling_model_scores(cl
    277         else:
    278             clf.set_params(base_estimator__sampler__n_components=f)
--> 279         clf.fit(data_train, target_train)
    280         train_score = clf.score(data_train, target_train)
    281         test_score = clf.score(data_test, target_test)


    ~/.local/lib/python3.6/site-packages/sklearn/ensemble/bagging.py in fit(self, X, y, sam
    242         self : object
    243         """
--> 244         return self._fit(X, y, self.max_samples, sample_weight=sample_weight)
    245 
    246     def _fit(self, X, y, max_samples=None, max_depth=None, sample_weight=None):


    ~/.local/lib/python3.6/site-packages/sklearn/ensemble/bagging.py in _fit(self, X, y, ma
    372                 total_n_estimators,
    373                 verbose=self.verbose)
--> 374             for i in range(n_jobs))
    375 
    376         # Reduce


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in __call__(
    981             # remaining jobs.
    982             self._iterating = False
--> 983             if self.dispatch_one_batch(iterator):
    984                 self._iterating = self._original_iterator is not None
    985


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in dispatch_
    823                 return False
    824             else:
--> 825                 self._dispatch(tasks)
    826                 return True
    827


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in _dispatch
    780         with self._lock:
    781             job_idx = len(self._jobs)
--> 782             job = self._backend.apply_async(batch, callback=cb)
    783             # A job can complete so quickly than its callback is
    784             # called before we get here, causing self._jobs to


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/_parallel_backends.py in
    180     def apply_async(self, func, callback=None):
    181         """Schedule a func to be run"""
--> 182         result = ImmediateResult(func)
    183         if callback:
    184             callback(result)


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/_parallel_backends.py in
    543         # Don't delay the application, to avoid keeping the input
    544         # arguments in memory
--> 545         self.results = batch()
    546
    547     def get(self):


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in __call__(
    259         with parallel_backend(self._backend):
    260             return [func(*args, **kwargs)
--> 261                     for func, args, kwargs in self.items]
    262
    263     def __len__(self):


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in <listcomp
    259         with parallel_backend(self._backend):
    260             return [func(*args, **kwargs)
--> 261                     for func, args, kwargs in self.items]
    262 
    263     def __len__(self):


    ~/.local/lib/python3.6/site-packages/sklearn/ensemble/bagging.py in _parallel_build_es
    112 
    113         else:
--> 114             estimator.fit((X[indices])[:, features], y[indices])
    115 
    116         estimators.append(estimator)


    ~/.local/lib/python3.6/site-packages/sklearn/pipeline.py in fit(self, X, y, **fit_para
    263             This estimator
    264         """
--> 265         Xt, fit_params = self._fit(X, y, **fit_params)
    266         if self._final_estimator is not None:
    267             self._final_estimator.fit(Xt, y, **fit_params)


    ~/.local/lib/python3.6/site-packages/sklearn/pipeline.py in _fit(self, X, y, **fit_par
    228                 Xt, fitted_transformer = fit_transform_one_cached(
    229                     cloned_transformer, Xt, y, None,
--> 230                     **fit_params_steps[name])
    231                 # Replace the transformer of the step with the fitted
    232                 # transformer. This is necessary when loading the transformer


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/memory.py in __call__(se
    327 
    328     def __call__(self, *args, **kwargs):
--> 329         return self.func(*args, **kwargs)
    330 
    331     def call_and_shelve(self, *args, **kwargs):


    ~/.local/lib/python3.6/site-packages/sklearn/pipeline.py in _fit_transform_one(transfo
    612 def _fit_transform_one(transformer, X, y, weight, **fit_params):
    613     if hasattr(transformer, 'fit_transform'):
--> 614         res = transformer.fit_transform(X, y, **fit_params)
    615     else:
```

```
    616         res = transformer.fit(X, y, **fit_params).transform(X)


    ~/.local/lib/python3.6/site-packages/sklearn/base.py in fit_transform(self, X, y, **fi
    463         else:
    464             # fit method of arity 2 (supervised transformation)
--> 465             return self.fit(X, y, **fit_params).transform(X)
    466
    467


    ~/.local/lib/python3.6/site-packages/sklearn/kernel_approximation.py in fit(self, X, y
    575
    576         # sqrt of kernel matrix on basis vectors
--> 577         U, S, V = svd(basis_kernel)
    578         S = np.maximum(S, 1e-12)
    579         self.normalization_ = np.dot(U / np.sqrt(S), V)


    ~/.local/lib/python3.6/site-packages/scipy/linalg/decomp_svd.py in svd(a, full_matrices
    130
    131     if info > 0:
--> 132         raise LinAlgError("SVD did not converge")
    133     if info < 0:
    134         raise ValueError('illegal value in %d-th argument of internal gesdd'


    LinAlgError: SVD did not converge
```

7.1 LinearSVC | Nystroem | Sí PCA

```
In [18]: model1 = {
             'model_name': 'linear_svc',
             'sampler_name': 'identity',
             'box_type': 'none',
             'n_estim': None,
             'pca': True
         }
         model2 = {
             'model_name': 'linear_svc',
             'sampler_name': 'nystroem',
             'box_type': 'black',
             'n_estim': 20,
             'pca': True
         }
         model3 = {
             'model_name': 'linear_svc',
             'sampler_name': 'nystroem',
             'box_type': 'grey',
             'n_estim': 20,
             'pca': True
         }

         info = {
             'models': [model1, model2, model3],
             'features_range': (30, 800),
             'dts_size': 2000,
             'dts_name': 'mnist'
         }
```

```
In [19]: Demo0().non_interactive(**info)
```

8 Una demo genérica

```
    ---------------------------------------------------------------------------

    LinAlgError                               Traceback (most recent call last)

    <ipython-input-19-0429739dc21c> in <module>
----> 1 Demo0().non_interactive(**info)


    ~/git/TFG/code/notebooks/python/demo_utils/generic_demo.py in non_interactive(self, **a
     40     def non_interactive(self, **argw):
     41         display(md(self.desc))
---> 42         train_scores, test_scores = self.run_demo(**argw)
     43         fig = self.get_generic_graph_from_scores(train_scores, test_scores)
     44         # fig.suptitle(self.title)


    ~/git/TFG/code/notebooks/python/demo_utils/demo0.py in run_demo(self, dts_name, dts_siz
    234                 # train_score y test_score son diccionarios
    235                 train_score, test_score =\
--> 236                     get_sampling_model_scores(clf, dataset, features)
    237                 lab = self.get_label(model_name, sampler_name, box_type,
    238                                      n_estim, pca)


    ~/git/TFG/code/notebooks/python/demo_utils/learning.py in get_sampling_model_scores(cl
    277         else:
    278             clf.set_params(base_estimator__sampler__n_components=f)
--> 279         clf.fit(data_train, target_train)
    280         train_score = clf.score(data_train, target_train)
    281         test_score = clf.score(data_test, target_test)


    ~/.local/lib/python3.6/site-packages/sklearn/ensemble/bagging.py in fit(self, X, y, sam
    242         self : object
    243         """
--> 244         return self._fit(X, y, self.max_samples, sample_weight=sample_weight)
    245
    246     def _fit(self, X, y, max_samples=None, max_depth=None, sample_weight=None):


    ~/.local/lib/python3.6/site-packages/sklearn/ensemble/bagging.py in _fit(self, X, y, ma
    372                 total_n_estimators,
    373                 verbose=self.verbose)
--> 374             for i in range(n_jobs))
    375
    376         # Reduce


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in __call__(s
    981             # remaining jobs.
    982             self._iterating = False
--> 983             if self.dispatch_one_batch(iterator):
    984                 self._iterating = self._original_iterator is not None
    985


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in dispatch_o
    823                 return False
    824             else:
--> 825                 self._dispatch(tasks)
    826                 return True
    827


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in _dispatch
    780         with self._lock:
    781             job_idx = len(self._jobs)
--> 782             job = self._backend.apply_async(batch, callback=cb)
    783             # A job can complete so quickly than its callback is
    784             # called before we get here, causing self._jobs to


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/_parallel_backends.py in
    180     def apply_async(self, func, callback=None):
    181         """Schedule a func to be run"""
--> 182         result = ImmediateResult(func)
    183         if callback:
```

```
    184             callback(result)


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/_parallel_backends.py in
    543         # Don't delay the application, to avoid keeping the input
    544         # arguments in memory
--> 545         self.results = batch()
    546 
    547     def get(self):


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in __call__(s
    259         with parallel_backend(self._backend):
    260             return [func(*args, **kwargs)
--> 261                     for func, args, kwargs in self.items]
    262 
    263     def __len__(self):


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/parallel.py in <listcomp>
    259         with parallel_backend(self._backend):
    260             return [func(*args, **kwargs)
--> 261                     for func, args, kwargs in self.items]
    262 
    263     def __len__(self):


    ~/.local/lib/python3.6/site-packages/sklearn/ensemble/bagging.py in _parallel_build_es
    112 
    113         else:
--> 114             estimator.fit((X[indices])[:, features], y[indices])
    115 
    116         estimators.append(estimator)


    ~/.local/lib/python3.6/site-packages/sklearn/pipeline.py in fit(self, X, y, **fit_param
    263             This estimator
    264         """
--> 265         Xt, fit_params = self._fit(X, y, **fit_params)
    266         if self._final_estimator is not None:
    267             self._final_estimator.fit(Xt, y, **fit_params)


    ~/.local/lib/python3.6/site-packages/sklearn/pipeline.py in _fit(self, X, y, **fit_para
    228                 Xt, fitted_transformer = fit_transform_one_cached(
    229                     cloned_transformer, Xt, y, None,
--> 230                     **fit_params_steps[name])
    231                 # Replace the transformer of the step with the fitted
    232                 # transformer. This is necessary when loading the transformer


    ~/.local/lib/python3.6/site-packages/sklearn/externals/joblib/memory.py in __call__(sel
    327
    328     def __call__(self, *args, **kwargs):
--> 329         return self.func(*args, **kwargs)
    330
    331     def call_and_shelve(self, *args, **kwargs):


    ~/.local/lib/python3.6/site-packages/sklearn/pipeline.py in _fit_transform_one(transfo
    612 def _fit_transform_one(transformer, X, y, weight, **fit_params):
    613     if hasattr(transformer, 'fit_transform'):
--> 614         res = transformer.fit_transform(X, y, **fit_params)
    615     else:
    616         res = transformer.fit(X, y, **fit_params).transform(X)


    ~/.local/lib/python3.6/site-packages/sklearn/base.py in fit_transform(self, X, y, **fi
    463         else:
    464             # fit method of arity 2 (supervised transformation)
--> 465             return self.fit(X, y, **fit_params).transform(X)
    466
    467


    ~/.local/lib/python3.6/site-packages/sklearn/kernel_approximation.py in fit(self, X, y
    575
    576         # sqrt of kernel matrix on basis vectors
--> 577         U, S, V = svd(basis_kernel)
    578         S = np.maximum(S, 1e-12)
    579         self.normalization_ = np.dot(U / np.sqrt(S), V)


    ~/.local/lib/python3.6/site-packages/scipy/linalg/decomp_svd.py in svd(a, full_matrice
    130
    131     if info > 0:
--> 132         raise LinAlgError("SVD did not converge")
    133     if info < 0:
    134         raise ValueError('illegal value in %d-th argument of internal gesdd'


    LinAlgError: SVD did not converge
```