GridSearchCV

Índice

In [1]:

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.multiclass import OneVsRestClassifier

import pickle
X_train_smt = pickle.load(open("saved_feats/ShuffleSplit_X_train_smt", "rb"))
X_test = pickle.load(open("saved_feats/ShuffleSplit_X_test", "rb"))
y_train_smt = pickle.load(open("saved_feats/ShuffleSplit_y_train_smt", "rb"))
y_test = pickle.load(open("saved_feats/ShuffleSplit_y_test", "rb"))
```

MultinomialNB

In [2]:

```
param_grid_mnb = {'alpha': [1, 0.1, 0.01, 0.001]}
grid_mnb = GridSearchCV(MultinomialNB(fit_prior=True,
                                      class_prior=[0.2, 0.5, 0.93]), param_grid_mnb)
grid_mnb.fit(X_train_smt, y_train_smt)
print("Mejor modelo predicho: " + str(grid_mnb.best_estimator_))
```

```
Mejor modelo predicho: MultinomialNB(alpha=0.001, class_prior=[0.2, 0.5, 0.93])
```

LinearSVC

In [3]:

```
param_grid_svc = {'C': [0.1, 1, 10, 100]}
grid_svc = GridSearchCV(LinearSVC(penalty='l1', loss='squared_hinge', dual=False, class_weight={'Negative': 0.
                        param_grid_svc, scoring='accuracy', refit=True, verbose=2)
grid_svc.fit(X_train_smt, y_train_smt)
print("Mejor modelo predicho: " + str(grid_svc.best_estimator_))
```

```
Fitting 5 folds for each of 4 candidates, totalling 20 fits
[CV] END ...........................................C=0.1; total time=   2.6s
[CV] END ...........................................C=0.1; total time=   2.6s
[CV] END ...........................................C=0.1; total time=   2.6s
[CV] END ...........................................C=0.1; total time=   2.7s
[CV] END ...........................................C=0.1; total time=   2.8s
[CV] END .............................................C=1; total time=  15.7s
[CV] END .............................................C=1; total time=  16.0s
[CV] END .............................................C=1; total time=  16.6s
[CV] END .............................................C=1; total time=  17.5s
[CV] END .............................................C=1; total time=  18.9s
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed to con
verge, increase the number of iterations.
  warnings.warn(
[CV] END ............................................C=10; total time=  25.4s
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed to con
verge, increase the number of iterations.
  warnings.warn(
[CV] END ............................................C=10; total time=  25.1s
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed to con
verge, increase the number of iterations.
  warnings.warn(
[CV] END ............................................C=10; total time=  24.8s
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed to con
verge, increase the number of iterations.
  warnings.warn(
[CV] END ............................................C=10; total time=  25.1s
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed to con
verge, increase the number of iterations.
  warnings.warn(
[CV] END ............................................C=10; total time=  25.5s
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed to con
verge, increase the number of iterations.
  warnings.warn(
[CV] END ...........................................C=100; total time=  21.6s
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed to con
verge, increase the number of iterations.
  warnings.warn(
[CV] END ...........................................C=100; total time=  21.5s
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed to con
verge, increase the number of iterations.
  warnings.warn(
[CV] END ...........................................C=100; total time=  21.5s
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed to con
verge, increase the number of iterations.
  warnings.warn(
[CV] END ...........................................C=100; total time=  21.6s
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed to con
verge, increase the number of iterations.
  warnings.warn(
[CV] END ...........................................C=100; total time=  22.3s
Mejor modelo predicho: LinearSVC(C=10,
          class_weight={'Negative': 0.2, 'Neutral': 0.5, 'Positive': 0.93},
          dual=False, penalty='l1')
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed to con
verge, increase the number of iterations.
  warnings.warn(
```

In [4]:

```
param_grid_svc = {'loss': ['hinge', 'squared_hinge'], 'C': [0.1, 1, 10, 100]}
grid_svc = GridSearchCV(LinearSVC(penalty='l2', dual=False, class_weight={'Negative': 0.2, 'Neutral': 0.5, 'Po
                        param_grid_svc, scoring='accuracy', refit=True, verbose=2)
grid_svc.fit(X_train_smt, y_train_smt)
print("Mejor modelo predicho: " + str(grid_svc.best_estimator_))
```

```
Fitting 5 folds for each of 8 candidates, totalling 40 fits
[CV] END ...............................C=0.1, loss=hinge; total time=   0.1s
[CV] END ...............................C=0.1, loss=hinge; total time=   0.1s
[CV] END ...............................C=0.1, loss=hinge; total time=   0.1s
[CV] END ...............................C=0.1, loss=hinge; total time=   0.1s
[CV] END ...............................C=0.1, loss=hinge; total time=   0.1s
[CV] END .......................C=0.1, loss=squared_hinge; total time=   0.7s
[CV] END .......................C=0.1, loss=squared_hinge; total time=   0.7s
[CV] END .......................C=0.1, loss=squared_hinge; total time=   0.7s
[CV] END .......................C=0.1, loss=squared_hinge; total time=   0.7s
[CV] END .......................C=0.1, loss=squared_hinge; total time=   0.7s
[CV] END .................................C=1, loss=hinge; total time=   0.1s
[CV] END .................................C=1, loss=hinge; total time=   0.1s
[CV] END .................................C=1, loss=hinge; total time=   0.1s
[CV] END .................................C=1, loss=hinge; total time=   0.1s
[CV] END .................................C=1, loss=hinge; total time=   0.1s
[CV] END .........................C=1, loss=squared_hinge; total time=   1.1s
[CV] END .........................C=1, loss=squared_hinge; total time=   1.3s
[CV] END .........................C=1, loss=squared_hinge; total time=   1.3s
[CV] END .........................C=1, loss=squared_hinge; total time=   1.1s
[CV] END .........................C=1, loss=squared_hinge; total time=   1.2s
[CV] END ................................C=10, loss=hinge; total time=   0.1s
[CV] END ................................C=10, loss=hinge; total time=   0.1s
[CV] END ................................C=10, loss=hinge; total time=   0.1s
[CV] END ................................C=10, loss=hinge; total time=   0.1s
[CV] END ................................C=10, loss=hinge; total time=   0.1s
[CV] END ........................C=10, loss=squared_hinge; total time=   2.5s
[CV] END ........................C=10, loss=squared_hinge; total time=   2.6s
[CV] END ........................C=10, loss=squared_hinge; total time=   2.6s
[CV] END ........................C=10, loss=squared_hinge; total time=   2.6s
[CV] END ........................C=10, loss=squared_hinge; total time=   3.3s
[CV] END ...............................C=100, loss=hinge; total time=   0.1s
[CV] END ...............................C=100, loss=hinge; total time=   0.1s
[CV] END ...............................C=100, loss=hinge; total time=   0.1s
[CV] END ...............................C=100, loss=hinge; total time=   0.1s
[CV] END ...............................C=100, loss=hinge; total time=   0.1s
[CV] END .......................C=100, loss=squared_hinge; total time=   7.2s
[CV] END .......................C=100, loss=squared_hinge; total time=  10.7s
[CV] END .......................C=100, loss=squared_hinge; total time=   7.3s
[CV] END .......................C=100, loss=squared_hinge; total time=   7.9s
[CV] END .......................C=100, loss=squared_hinge; total time=   4.8s
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py:378: FitFailedWarning:
20 fits failed out of a total of 40.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
20 fits failed with the following error:
Traceback (most recent call last):
  File "/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py", line 686, in _fit_and
_score
    estimator.fit(X_train, y_train, **fit_params)
  File "/usr/local/lib/python3.10/dist-packages/sklearn/svm/_classes.py", line 257, in fit
    self.coef_, self.intercept_, n_iter_ = _fit_liblinear(
  File "/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py", line 1204, in _fit_liblinear
    solver_type = _get_liblinear_solver_type(multi_class, penalty, loss, dual)
  File "/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py", line 1043, in _get_liblinear_solver_typ
e
    raise ValueError(
ValueError: Unsupported set of arguments: The combination of penalty='l2' and loss='hinge' are not supported w
hen dual=False, Parameters: penalty='l2', loss='hinge', dual=False

  warnings.warn(some_fits_failed_message, FitFailedWarning)
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.py:953: UserWarning: One or more of th
e test scores are non-finite: [        nan 0.92701578        nan 0.95865372        nan 0.96052022
        nan 0.96947898]
  warnings.warn(
Mejor modelo predicho: LinearSVC(C=100,
          class_weight={'Negative': 0.2, 'Neutral': 0.5, 'Positive': 0.93},
          dual=False)
```

Logistic Regression

In [10]:

```
param_grid_logit = {'penalty': ['l1', 'l2', 'elasticnet', 'none'], 'C': [
    0.1, 1, 10, 100], 'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']}
grid_logit = GridSearchCV(LogisticRegression(class_weight={'Negative': 0.2, 'Neutral': 0.5, 'Positive': 0.93}
                          param_grid_logit, verbose=2)
grid_logit.fit(X_train_smt, y_train_smt)
```