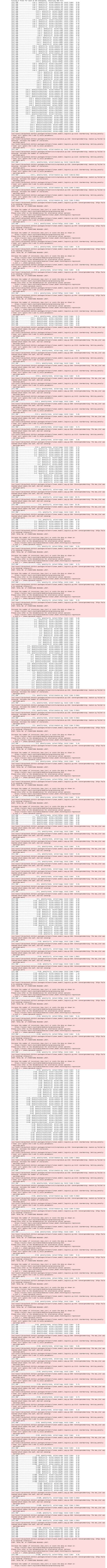
prin Mejor inea	<pre>rt pickle ain_smt = pickle.load(open("saved_feats/ShuffleSplit_X_train_smt", "rb")) st = pickle.load(open("saved_feats/ShuffleSplit_X_test", "rb")) ain_smt = pickle.load(open("saved_feats/ShuffleSplit_y_train_smt", "rb")) st = pickle.load(open("saved_feats/ShuffleSplit_y_test", "rb")) nomialNB m_grid_mnb = {'alpha': [1, 0.1, 0.01, 0.001]} _mnb = GridSearchCV(MultinomialNB(fit_prior=True,</pre>
grid	<pre>_mnb.fit(X_train_smt, y_train_smt) t("Mejor modelo predicho: " + str(grid_mnb.best_estimator_)) modelo predicho: MultinomialNB(alpha=0.001, class_prior=[0.2, 0.5, 0.93]) arSVC m_grid_svc = {'C': [0.1, 1, 10, 100]} _svc = GridSearchCV(LinearSVC(penalty='l1', loss='squared_hinge', dual=False, class_weight={'Negati_param_grid_svc, scoring='accuracy', refit=True, verbose=2) _svc.fit(X_train_smt, y_train_smt) t("Mejor modelo predicho: " + str(grid_svc.best_estimator_))</pre>
Fitti [CV] [CV] [CV] [CV] [CV] [CV] [CV] [CV]	ng 5 folds for each of 4 candidates, totalling 20 fits END
[CV] /usr/ verge war [CV] /usr/ verge war [CV] /usr/ verge war	nings.warn(END
verge war [CV] /usr/ verge war [CV] /usr/ verge war [CV] /usr/ verge war	local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed, increase the number of iterations. nings.warn(END
/usr/verge war [CV] /usr/verge war [CV] Mejor	<pre>Liblinear failed local/lib/python3.10/dist-packages/sklearn/svm/_base.py:1225: ConvergenceWarning: Liblinear failed , increase the number of iterations. nings.warn(END</pre>
para grid grid prin Fitti [CV] [CV] [CV]	<pre>nings.warn(m_grid_svc = {'loss': ['hinge', 'squared_hinge'], 'C': [0.1, 1, 10, 100]} _svc = GridSearchCV(LinearSVC(penalty='l2', dual=False, class_weight={'Negative': 0.2, 'Neutral': 0</pre>
[CV] [CV] [CV] [CV] [CV] [CV] [CV] [CV]	END C=0.1, loss=squared_hinge; total time= 0.7s END C=0.1, loss=squared_hinge; total time= 0.7s END C=0.1, loss=squared_hinge; total time= 0.7s END C=0.1, loss=squared_hinge; total time= 0.1s END
[CV] [CV] [CV] [CV] [CV] [CV] [CV] [CV]	C=10, loss=hinge; total time=
[CV] [CV] [CV] [CV] /usr/ 20 fi The s If th Below 20 fi Trace	END
_scor e Fil s Fil e r Value hen d war /usr/	
war Mejor Ogis para	<pre>nan 0.96947898] nings.warn(modelo predicho: LinearSVC(C=100,</pre>



/usr/local/lib/python3.2 reached which means the warnings.warn([CV] END	C=100, penalty=l2, solver=sag; total time= 3.3s
/usr/local/lib/python3.: reached which means the warnings.warn(C=100, penalty=l2, solver=sag; total time= 3.2s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not converge C=100, penalty=l2, solver=sag; total time= 3.3s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma
/usr/local/lib/python3.2 reached which means the warnings.warn([CV] END/ usr/local/lib/python3.2 reached which means the warnings.warn([CV] END/ usr/local/lib/python3.2 reached which means the warnings.warn(<pre>10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not convergeC=100, penalty=l2, solver=saga; total time= 3.7s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not convergeC=100, penalty=l2, solver=saga; total time= 3.6s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma</pre>
/usr/local/lib/python3.2 reached which means the warnings.warn([CV] END/ usr/local/lib/python3.2 reached which means the warnings.warn(/usr/local/lib/python3.2 ='none' will ignore the warnings.warn([CV] END	<pre>10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not convergeC=100, penalty=l2, solver=saga; total time= 3.7s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not converge 10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parametersC=100, penalty=l2, solver=saga; total time= 3.7s penalty=elasticnet, solver=newton-cg; total time= 0.0s</pre>
[CV] END	penalty=elasticnet, solver=newton-cg; total time= 0.0s 100, penalty=elasticnet, solver=lbfgs; total time= 0.0s penalty=elasticnet, solver=liblinear; total time= 0.0s
[CV] END .0	penalty=elasticnet, solver=liblinear; total time= 0.0s C=100, penalty=elasticnet, solver=sag; total time= 0.0s =100, penalty=elasticnet, solver=saga; total time= 0.0s =100/dist-packages/sklearn/utils/optimize.py:210: ConvergenceWarning: newton-cg
warnings.warn(/usr/local/lib/python3.2 ='none' will ignore the warnings.warn([CV] END	10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parameters =100, penalty=none, solver=newton-cg; total time=191.8min 10/dist-packages/sklearn/utils/optimize.py:210: ConvergenceWarning: newton-cg
/usr/local/lib/python3.2 converge. Increase the marnings.warn(/usr/local/lib/python3.2 ='none' will ignore the warnings.warn([CV] END	<pre>10/dist-packages/sklearn/utils/optimize.py:210: ConvergenceWarning: newton-cg number of iterations. 10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parameters C=100, penalty=none, solver=newton-cg; total time= 8.7min 10/dist-packages/sklearn/utils/optimize.py:210: ConvergenceWarning: newton-cg number of iterations. 10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting</pre>
warnings.warn([CV] END	<pre>10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parameters C=100, penalty=none, solver=newton-cg; total time= 9.5min 10/dist-packages/sklearn/linear_model/_logistic.py:444: ConvergenceWarning: l):</pre>
Increase the number of a https://scikit-learn please also refer to the https://scikit-learn n_iter_i = _check_opt:/usr/local/lib/python3.a='none' will ignore the warnings.warn([CV] END/usr/local/lib/python3.a	iterations (max_iter) or scale the data as shown in: n.org/stable/modules/preprocessing.html e documentation for alternative solver options: n.org/stable/modules/linear_model.html#logistic-regression imize_result(10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parametersC=100, penalty=none, solver=lbfgs; total time= 6.1s 10/dist-packages/sklearn/linear_model/_logistic.py:444: ConvergenceWarning: l
d to converge (status=1) STOP: TOTAL NO. of ITER/ Increase the number of in https://scikit-lear/ Please also refer to the https://scikit-lear/ n_iter_i = _check_opt://usr/local/lib/python3.1 ='none' will ignore the warnings.warn([CV] END): ATIONS REACHED LIMIT. iterations (max_iter) or scale the data as shown in: n.org/stable/modules/preprocessing.html e documentation for alternative solver options: n.org/stable/modules/linear_model.html#logistic-regression imize_result(10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parametersC=100, penalty=none, solver=lbfgs; total time= 5.3s
d to converge (status=1) STOP: TOTAL NO. of ITER/ Increase the number of in https://scikit-lear/ Please also refer to the https://scikit-lear/ n_iter_i = _check_opt://usr/local/lib/python3.: ='none' will ignore the warnings.warn(ATIONS REACHED LIMIT. iterations (max_iter) or scale the data as shown in: n.org/stable/modules/preprocessing.html e documentation for alternative solver options: n.org/stable/modules/linear_model.html#logistic-regression imize_result(10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parameters
/usr/local/lib/python3.7 /usr/local/lib/python3.7 d to converge (status=1) STOP: TOTAL NO. of ITER/ Increase the number of : https://scikit-leary Please also refer to the https://scikit-leary n_iter_i = _check_opt://usr/local/lib/python3.3	ATIONS REACHED LIMIT. iterations (max_iter) or scale the data as shown in: n.org/stable/modules/preprocessing.html e documentation for alternative solver options: n.org/stable/modules/linear_model.html#logistic-regression
/usr/local/lib/python3.2 d to converge (status=1) GTOP: TOTAL NO. of ITER/ Increase the number of a https://scikit-leary Please also refer to the https://scikit-leary n_iter_i = _check_opta /usr/local/lib/python3.2 e'none' will ignore the	ATIONS REACHED LIMIT. iterations (max_iter) or scale the data as shown in: n.org/stable/modules/preprocessing.html e documentation for alternative solver options: n.org/stable/modules/linear_model.html#logistic-regression
CV] END	C=100, penalty=none, solver=lbfgs; total time= 5.5s C=100, penalty=none, solver=liblinear; total time= 0.0s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not converge 10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parameters
/usr/local/lib/python3.2 reached which means the warnings.warn(/usr/local/lib/python3.2 ='none' will ignore the warnings.warn([CV] END/ /usr/local/lib/python3.2 reached which means the warnings.warn(C=100, penalty=none, solver=sag; total time= 3.8s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not converge 10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parameters C=100, penalty=none, solver=sag; total time= 3.6s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not converge
e'none' will ignore the warnings.warn([CV] END	<pre>10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parametersC=100, penalty=none, solver=sag; total time= 3.4s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not converge 10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parametersC=100, penalty=none, solver=sag; total time= 3.7s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma</pre>
warnings.warn('usr/local/lib/python3.2 e'none' will ignore the warnings.warn([CV] END	<pre>coef_ did not converge 10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parametersC=100, penalty=none, solver=sag; total time= 3.9s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not converge 10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parameters</pre>
[CV] END	C=100, penalty=none, solver=saga; total time= 3.8s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not converge 10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parameters C=100, penalty=none, solver=saga; total time= 3.7s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not converge 10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting
e'none' will ignore the warnings.warn([CV] END	C and l1_ratio parameters C=100, penalty=none, solver=saga; total time= 3.8s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma coef_ did not converge 10/dist-packages/sklearn/linear_model/_logistic.py:1113: UserWarning: Setting C and l1_ratio parameters C=100, penalty=none, solver=saga; total time= 3.8s 10/dist-packages/sklearn/linear_model/_sag.py:350: ConvergenceWarning: The ma
warnings.warn('usr/local/lib/python3.2 80 fits failed out of a The score on these train If these failures are no Below are more details a 20 fits failed with the Traceback (most recent of File "/usr/local/lib/p	n-test partitions for these parameters will be set to nan. ot expected, you can try to debug them by setting error_score='raise'. about the failures:
File "/usr/local/lib/p solver = _check_solv File "/usr/local/lib/p raise ValueError(/alueError: Solver newton 20 fits failed with the raceback (most recent of	· · · · · · · · · · · · · · · · · · ·
estimator.fit(X_tra: File "/usr/local/lib/ solver = _check_solv File "/usr/local/lib/ raise ValueError(/alueError: Solver lbfg: 20 fits failed with the Traceback (most recent of File "/usr/local/lib/ _score	call last): python3.10/dist-packages/sklearn/model_selection/_validation.py", line 686, i
File "/usr/local/lib/p solver = _check_solv File "/usr/local/lib/p raise ValueError(/alueError: Solver sag s 20 fits failed with the Fraceback (most recent of File "/usr/local/lib/p _score	· · · · · · · · · · · · · · · · · · ·
File "/usr/local/lib/p solver = _check_solv File "/usr/local/lib/p raise ValueError(/alueError: Solver newton) 20 fits failed with the fraceback (most recent of File "/usr/local/lib/p score	<pre>python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 1091, in fi ver(self.solver, self.penalty, self.dual) python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 61, in _che on-cg supports only 'l2' or 'none' penalties, got elasticnet penalty</pre>
File "/usr/local/lib/p solver = _check_solv File "/usr/local/lib/p raise ValueError('alueError: Solver lbfg: 0 fits failed with the raceback (most recent of File "/usr/local/lib/p score	<pre>python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 1091, in fi ver(self.solver, self.penalty, self.dual) python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 61, in _che s supports only 'l2' or 'none' penalties, got elasticnet penalty. following error:</pre>
File "/usr/local/lib/p solver = _check_solv File "/usr/local/lib/p raise ValueError(/alueError: Only 'saga' 20 fits failed with the raceback (most recent of File "/usr/local/lib/p score estimator.fit(X_train	<pre>python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 1091, in fi ver(self.solver, self.penalty, self.dual) python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 71, in _che solver supports elasticnet penalty, got solver=liblinear. following error: call last): python3.10/dist-packages/sklearn/model_selection/_validation.py", line 686, i in, y_train, **fit_params)</pre>
solver = _check_solver File "/usr/local/lib/post raise ValueError(//alueError: Solver sages)	
raise ValueError(/alueError: l1_ratio mus 20 fits failed with the Fraceback (most recent of File "/usr/local/lib/p _score estimator.fit(X_trainely) solver = _check_solv	call last): python3.10/dist-packages/sklearn/model_selection/_validation.py", line 686, i in, y_train, **fit_params) python3.10/dist-packages/sklearn/linear_model/_logistic.py", line 1091, in fi ver(self.solver, self.penalty, self.dual)
raise ValueError("per ValueError: penalty='non warnings.warn(some_fir Vusr/local/lib/python3.2 e test scores are non-fir 0.68307724 0.73038657 0 nan nan 0.95917937 0.95885084	0.68301154 0.68307724 nan nan nan 0.94738497 0.95806237 nan nan nan 0.9044452 nan 0.87615824 0.89133662 0.87579685 0.8758297
0.95806237 nan 0 0.95640322 nan 0 0.94869906 0.94866621 nan 0.94738497 0	0.95929435 0.95878513 nan nan 0.95484267 0.94869906 0.94554512 0.94623503 nan nan nan nan 0.95806237 nan 0.95911366 0.95880155 0.95962289 nan 0.95824305 0.95747097 0.95738884 0.95712601 nan nan nan 0.94738497 0.95806237 nan
<pre>{'C': 100, 'penalty': ' LogisticRegression(C=100</pre>	<pre>s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre {'C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre {'C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre {'C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre {'C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' ogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre {'C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre {'C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' ogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' ogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pressor print("Mejor modelo pressor penas pena	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pressor print("Mejor modelo pressor penas pena	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pressor print("Mejor modelo pressor penas pena	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' logisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pressor print("Mejor modelo pressor penas pena	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pressor print("Mejor modelo pressor penas pena	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pressor print("Mejor modelo pressor penas pena	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pressor print("Mejor modelo pressor penas pena	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre {'C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' ogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre {'C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' ogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre {'C': 100, 'penalty': ' LogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pre ['C': 100, 'penalty': ' ogisticRegression(C=100 class pena param_grid_logit = {'pe 0.1, 1, 10, 100], ' grid_logit = GridSearch grid_logit.fit(X_train_	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pressor print("Mejor modelo pressor penas pena	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>
print("Mejor modelo pressor print("Mejor modelo pressor penas pena	<pre>l1', 'solver': 'liblinear'} 0, s_weight={'Negative': 0.2, 'Neutral': 0.5,</pre>

