**Práctico Aprendizaje Supervisado**

**Olariaga Sandra, Pesce Cristian**

**Cuadro comparativo modelos con GridSearchCV**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modelos** | **Conjunto entrenamiento**  **Mejor Modelo y Mejores Parametros** | **Conjunto test**  **Mejor Modelo y Mejores Parametros** | **roc\_auc\_score** | **f1\_score** |
| **Random Forest con GridSearchCV**  GridSearchCV(cv=3, estimator=RandomForestClassifier(), n\_jobs=-1,  param\_grid={'class\_weight': ['balanced', 'balanced\_subsample'],  'criterion': ['gini'], 'min\_samples\_leaf': [1],  'min\_samples\_split': [2, 3],  'n\_estimators': [100, 150], 'random\_state': [0]},  scoring='accuracy', verbose=4) | RandomForestClassifier(class\_weight='balanced\_subsample', n\_estimators=150,  random\_state=0)  Mejores Parametros  {'class\_weight': 'balanced\_subsample', 'criterion': 'gini', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 150, 'random\_state': 0}  Best Random Forest accuracy: 0.72245217792118 | RandomForestClassifier(class\_weight='balanced\_subsample', n\_estimators=150,  random\_state=0)  Mejores Parametros  {'class\_weight': 'balanced\_subsample', 'criterion': 'gini', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 150, 'random\_state': 0}  Best Random Forest accuracy: 0.7447317983646381 | 0.9520628136642012 | 0.9351954910291063 |
| Random Forest con MinMax Scaler  GridSearchCV(cv=3, estimator=RandomForestClassifier(), n\_jobs=-1,  param\_grid={'class\_weight': ['balanced', 'balanced\_subsample'],  'criterion': ['gini'], 'min\_samples\_leaf': [1],  'min\_samples\_split': [2, 3], 'n\_estimators': [100, 150], 'random\_state': [0]}, scoring='accuracy', verbose=4) | Mejor Modelo  RandomForestClassifier(class\_weight='balanced\_subsample', n\_estimators=150,  random\_state=0)  Mejores Parametros  {'class\_weight': 'balanced\_subsample', 'criterion': 'gini', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 150, 'random\_state': 0}  Best Random Forest accuracy: 0.722565875393716 | Mejor Modelo  RandomForestClassifier(class\_weight='balanced', n\_estimators=150,  random\_state=0)  Mejores Parametros  {'class\_weight': 'balanced', 'criterion': 'gini', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 150, 'random\_state': 0}  Best Random Forest accuracy: 0.7449253890009172 | 0.9520905380952714 | 0.6989437521852746 |
| Modelo Linear SVC  Accuracy: 51 |  |  | 0.569954134555689 | 0.5524152896417701 |

De estaq tabla podemos deducir que el mejor modelo es el Random Forest con GridSearchCV por el accuracy y los score de las curvas ROC.

**Otros modelos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modelo** | **Mejores parámetros** | **Mejor Estimador** | **roc\_auc\_score** | **f1\_score** |
| **BayesSearchCV**  BayesSearchCV(cv=3, estimator=RandomForestClassifier(), n\_iter=3, n\_jobs=-1, scoring='accuracy',  search\_spaces={'bootstrap': Categorical(categories=(True, False), prior=None),  'max\_depth': Integer(low=6, high=20, prior='uniform', transform='identity'),  'max\_features': Categorical(categories=('auto', 'sqrt', 'log2'), prior=None),  'min\_samples\_leaf': Integer(low=2, high=10, prior='uniform', transform='identity'),  'min\_samples\_split': Integer(low=2, high=10, prior='uniform', transform='identity'),  'n\_estimators': Integer(low=100, high=500, prior='uniform', transform='identity')},  verbose=4)  **Best Score:** 0.7809787201352079 | OrderedDict([('bootstrap', False),  ('max\_depth', 20),  ('max\_features', 'sqrt'),  ('min\_samples\_leaf', 7),  ('min\_samples\_split', 4),  ('n\_estimators', 165)]) | RandomForestClassifier(bootstrap=False, max\_depth=20, max\_features='sqrt',  min\_samples\_leaf=7, min\_samples\_split=4,  n\_estimators=165) | 0.8704595227789397 | 0.6933247797506192 |
| XGBClassifier  XGBClassifier(alpha=5, base\_score=0.5, booster='gbtree', colsample\_bylevel=1,  colsample\_bynode=1, colsample\_bytree=1, cv=3, gamma=0, gpu\_id=-1, importance\_type='gain', interaction\_constraints='', learning\_rate=0.1, max\_delta\_step=0, max\_depth=10, min\_child\_weight=1, missing=nan, monotone\_constraints='()', n\_estimators=50, n\_jobs=-1, num\_parallel\_tree=1, random\_state=0, reg\_alpha=5, reg\_lambda=1, scale\_pos\_weight=1, scoring='accuracy',  subsample=1, tree\_method='exact', validate\_parameters=1, verbose=3, verbosity=None)  **Boosting accuracy: 0.7846296117079046** |  |  | 0.5172289535103792 | 0.6989437521852746 |

De la tabla anterior podemos deducir que el modelo BayesSearchCV también tiene un buen desempeño según los score de las curvas ROC.