

## Boosting

Se hará un estudio utilizando el modelo de AdaBoost para regresión introducido por Freund y Schapire en 1995.

TODO Añadir la bibliografía Y. Freund, R. Schapire, “A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting”, 1995.

Para ello se utilizará la función

```
class sklearn.ensemble.AdaBoostRegressor(base_estimator=None,
*, n_estimators=50, learning_rate=1.0,
loss='linear', random_state=None)
```

Bibliografía: - Teoría: <https://scikit-learn.org/stable/modules/ensemble.html#adaboost>

- Implementación: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html>

Los hiperparámetros que ajustaremos y utilizaremos de esta función son los siguientes.

- **base\_estimator** objeto con el estimador base que utilizar, utilizares **DecisionTreeRegressor** inicialos con profundidad máxima de tres. (TODO, justificar, en el guión se nos indica que utilicemos estos.
- **learning\_rate** haremos un estudio de cómo varía en función del learning rate.
- **loss** La función de pérdida para actualizar los pesos del boosting en cada iteración, será la línea
- **n\_estimators** número de estimadores para el cual el boosting termina, en caso de encontrarse un ajuste perfecto pararía antes.

### Estudio preliminar

A partir de los datos normalizados y sin outliers probaremos una serie de parámetros para comprobar cuáles dan los mejores resultados.

Realizaremos estimaciones con el número de estimadores y la tasa de aprendizaje, obteniendo los siguientes resultados:

Mejores parámetros: {'learning\_rate': 0.01, 'n\_estimators': 80} Con una  $R^2$  de: 0.5863897348444905

Parámetros	$R^2$ medio	Desviación típica $R^2$	Ranking	tiempo medio ajuste
n_estimators 100 learning_rate 0.01	0.5850	0.0110	1.0000	3.8879
n_estimators 50 learning_rate 0.1	0.5843	0.0152	2.0000	1.9871
n_estimators 60 learning_rate 0.1	0.5842	0.0173	3.0000	2.2133
n_estimators 80 learning_rate 0.1	0.5813	0.0239	4.0000	2.8120
n_estimators 80 learning_rate 0.01	0.5811	0.0083	5.0000	3.0417
n_estimators 100 learning_rate 0.1	0.5805	0.0237	6.0000	3.4479

Parámetros	$R^2$ medio	Desviación típica $R^2$	Ranking	tiempo medio ajuste
n_estimators 100 learning_rate 0.001	0.5795	0.0107	7.0000	4.7428
n_estimators 50 learning_rate 0.01	0.5794	0.0075	8.0000	2.3126
n_estimators 60 learning_rate 0.001	0.5792	0.0096	9.0000	2.6877
n_estimators 80 learning_rate 0.001	0.5792	0.0122	10.0000	4.4050
n_estimators 60 learning_rate 0.01	0.5788	0.0092	11.0000	2.2067
n_estimators 50 learning_rate 0.001	0.5768	0.0110	12.0000	1.7847
n_estimators 50 learning_rate 1	0.4986	0.0410	13.0000	1.5605
n_estimators 80 learning_rate 1	0.4830	0.0455	14.0000	2.2940
n_estimators 60 learning_rate 1	0.4813	0.0462	15.0000	1.7972
n_estimators 100 learning_rate 1	0.4647	0.0587	16.0000	2.6109

De esto valores observamos que una tasa de aprendizaje de entre 0.01 y 0.1 parece ser la mejor opción.

Analicemos cómo varía entonces si cambiamos el conjunto de entrenamiento:

Table 2: Comparativa de mejores resultados en validación cruzada según el preprocesado de los datos

Datos de entrenamiento	Mejor error	Con parámetros
Sin normalizar con outliers	0.6193	{‘learning_rate’: 0.1, ‘n_estimators’: 50}
Normalizado con outliers	0.6161	{‘learning_rate’: 0.01, ‘n_estimators’: 100}
Sin normalizar sin outliers	0.5851	{‘learning_rate’: 0.01, ‘n_estimators’: 100}
Normalizado con outliers	0.5829	{‘learning_rate’: 0.1, ‘n_estimators’: 50}

Además analizando las distintas pruebas vemos que de manera general se mantiene esa mejora (ver sucesivas tablas de validación cruzada).

Table 3: Validación cruzada para data set sin normalizar con outliers

Parámetros	$R^2$ medio	Desviación típica $R^2$	Ranking	tiempo medio ajuste
n_estimators 50 learning_rate 0.1	0.6193	0.0124	1.0000	1.5586
n_estimators 100 learning_rate 0.01	0.6160	0.0247	2.0000	3.2366
n_estimators 100 learning_rate 0.1	0.6146	0.0126	3.0000	2.7815
n_estimators 50 learning_rate 0.01	0.6102	0.0293	4.0000	1.6474

Table 4: Validación cruzada para data set normalizado con outliers

Parámetros	$R^2$ medio	Desviación típica $R^2$	Ranking	tiempo medio ajuste
n_estimators 100 learning_rate 0.01	0.6161	0.0242	1.0000	3.1191
n_estimators 100 learning_rate 0.1	0.6159	0.0135	2.0000	2.7651
n_estimators 50 learning_rate 0.01	0.6149	0.0276	3.0000	1.5735
n_estimators 50 learning_rate 0.1	0.6131	0.0150	4.0000	1.5489

Table 5: Validación cruzada para data set sin normalizar sin outliers

Parámetros	$R^2$ medio	Desviación típica $R^2$	Ranking	tiempo medio ajuste
n_estimators 100 learning_rate 0.01	0.5851	0.0089	1.0000	3.0453
n_estimators 50 learning_rate 0.01	0.5824	0.0109	2.0000	1.5233
n_estimators 50 learning_rate 0.1	0.5814	0.0188	3.0000	1.5110
n_estimators 100 learning_rate 0.1	0.5768	0.0245	4.0000	2.7195

Table 6: Normalizado con outliers

Parámetros	$R^2$ medio	Desviación típica $R^2$	Ranking	tiempo medio ajuste
n_estimators 50 learning_rate 0.1	0.5829	0.0169	1.0000	1.5142
n_estimators 100 learning_rate 0.01	0.5824	0.0087	2.0000	3.0499
n_estimators 50 learning_rate 0.01	0.5811	0.0107	3.0000	1.5124
n_estimators 100 learning_rate 0.1	0.5801	0.0246	4.0000	2.7327

El hecho de que los mejores sean sin normalizar con outliers y normalizado con outliers; y que ambos tengan errores muy similares nos hace pensar que los coeficientes del modelo juegan un papel importante vamos a probar con transformaciones de los datos y regularización.