Índice general

Nota: Para la replicación de los resultados, haremos uso de la función *trainControl*, donde emplearemos validación cruzada con 5 grupos y tres repeticiones. No utilizamos 10 grupos en la validación como es lo usual debido al pequeño número de registros que tenemos.

```
# Para todos los modelos
fitControl <- trainControl(method = "repeatedcv",
                           number = 5, repeats = 3,
                           verboseIter =FALSE )
```

De cara a utilizar optimizar el rendimiento de los algoritmos de regresión, haremos uso de la instrucción *preProcess* de la función *train* para escalar nuestras variables y que estén todas en la misma escala, con el objetivo de obtener mejores métricas y que los modelo minimicen el error al predecir las ventas.

Al hacer la división de los datos para entrenar los modelos, se ha optado por no utilizar un conjunto de datos de validación ya que únicamente tenemos 181 registros y tener datos para validad los modelos supondría tener aún menos registros para entrenarlos, y por tanto, se obtendrían métricas menos precisas.

0.0.0.0.1. Predicción del volumen total de ventas

División de los datos en entrenamiento y testeo

Se toma una partición de 80 % 20 % para los datos de entrenamiento y testeo:

```
set.seed(17)
indices <-
  createDataPartition(VolumenVentas_TOTAL$VENTAS, p = .8, list = FALSE)

# Datos de entrenamiento
DatosEntreamiento_Total <- VolumenVentas_TOTAL[indices,]
# Datos de testeo
DatosTesteo_Total<-VolumenVentas_TOTAL[-indices,]
```

0.0.0.0.1.1. Algoritmo 1: Máquina de vector soporte (SVM)

Hiperparámetros del algoritmo

- Validación cruzada con 5 grupos y tres repeticiones
- Parámetro de costo, C: malla de valores entre 1 y 3. Este parámetro penaliza al modelo por cometer errores. Cuanto mayor sea su valor, menos probable es que el algoritmo realice una predicción errónea.

Modelado

```
# Malla para hiperparámetros
SVMGrid <-  expand.grid(C = seq(1,3, length = 20))

set.seed(17)
modeloSVM_T <- train(VENTAS~.,
                 data = DatosEntreamiento_Total[,-1],
                 method = "svmLinear",
                 trControl=fitControl,
                 preProcess=c("center","scale"),
                 tuneGrid = SVMGrid)
```

Resultados

El modelo tiene un costo $C = 1.2105263$ y nos ofrece las siguientes métricas:

Tabla 1: Métricas del mejor modelo

	C	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
3	1.210526	587.6893	0.6561937	302.248	353.6313	0.2462402	106.1962

Métricas del remuestreo:

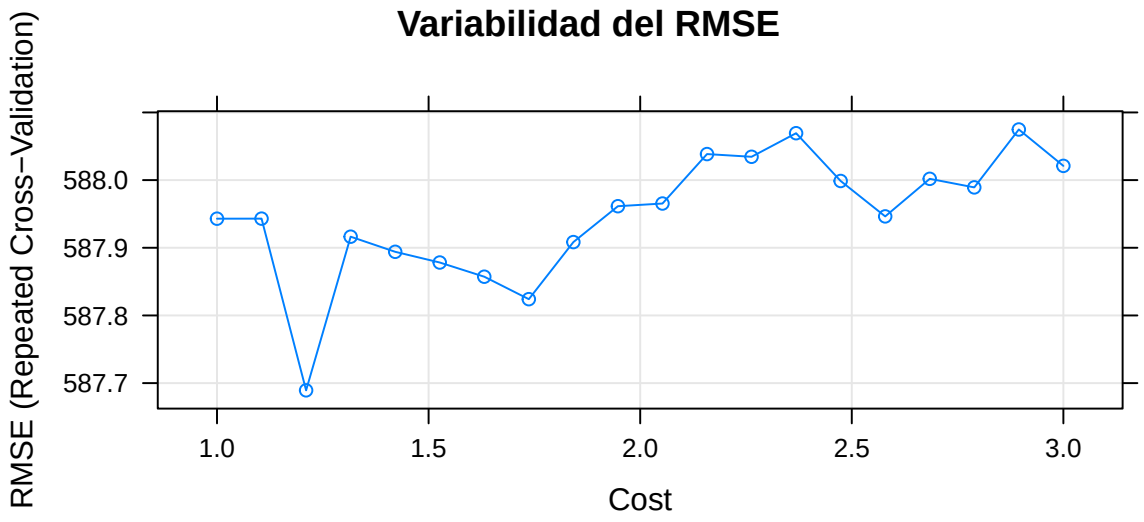
Tabla 2: Métricas en el remuestreo

RMSE	Rsquared	MAE	Resample
1220.4229	0.3006125	481.5539	Fold1.Rep1
291.9557	0.8889093	237.2185	Fold2.Rep1
262.4682	0.8793230	194.5485	Fold2.Rep3
489.6627	0.7161635	292.3244	Fold4.Rep3
394.5036	0.8365604	223.9593	Fold3.Rep1
452.5807	0.7743701	227.9287	Fold5.Rep1
256.4737	0.9247301	209.0976	Fold2.Rep2
657.9852	0.3823591	330.6228	Fold4.Rep2
259.4812	0.9166944	203.5507	Fold1.Rep3
1184.4119	0.3707289	469.2383	Fold3.Rep3
718.4412	0.5383558	357.2759	Fold5.Rep3
640.1730	0.3813119	333.5713	Fold4.Rep1
229.5025	0.9455725	174.8641	Fold1.Rep2
565.4990	0.6451018	321.2116	Fold3.Rep2
1191.7784	0.3421120	476.7550	Fold5.Rep2

Observando los resultados, llegamos lo siguiente:

- El modelo consigue explicar un 65.62 % de la variabilidad total del volumen de ventas para los datos de entrenamiento
- El error cuadrático medio es de 588 unidades
- Respecto al remuestreo en la validación cruzada, vemos que las métricas tienen mucha variabilidad, en algunas ocasiones se obtiene un R^2 por encima de 0.9 y en otras de 0.3. Ocurre lo mismo para el error cuadrático medio y el error medio absoluto, indicando que el modelo no es muy robusto y por tanto, las predicciones serán poco fiables.

En el gráfico mostrado a continuación, se puede observar la variabilidad del error cuadrático medio en función del valor de costo:



0.0.0.0.1.2. Algoritmo 2: K-Nearest Neighbor Regression (KNN)

Hiperparámetros del algoritmo

- Validación cruzada con 5 grupos y tres repeticiones
- Número de vecinos, k: malla para 3,5,7 y 9

Modelado

```
# Malla para hiperparámetros
KNNGrid <-  expand.grid(k = seq(3,9, by=2))

set.seed(17)
modeloKNN_T <- train(VENTAS~.,
                     data = DatosEntreamiento_Total[,-1],
                     method = "knn",
                     trControl=fitControl,
                     preProcess=c("center","scale"),
                     tuneGrid = KNNGrid)
```

Resultados

El con $K = 3$ vecinos es el que nos proporciona mejores métricas:

Tabla 3: Métricas del mejor modelo

k	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
3	846.7828	0.3436966	540.7302	285.1381	0.1598935	118.3129

Métricas del remuestreo

Tabla 4: Métricas en el remuestreo

RMSE	Rsquared	MAE	Resample
1404.6704	0.1056963	631.8736	Fold1.Rep1
749.0636	0.3333724	488.4946	Fold2.Rep1
715.9994	0.4581447	497.8966	Fold3.Rep1
1277.4472	0.2913174	627.4253	Fold3.Rep3
1413.4015	0.0842595	836.8736	Fold5.Rep2
623.3366	0.5333188	453.2874	Fold2.Rep2
580.8543	0.4750718	412.3214	Fold4.Rep1
902.7923	0.1422233	638.6071	Fold4.Rep3
796.0439	0.2502584	615.2644	Fold1.Rep3
624.4946	0.5588357	423.7000	Fold3.Rep2
797.2842	0.3553866	546.4762	Fold5.Rep1
804.9407	0.4371745	476.7586	Fold5.Rep3
562.9675	0.4646278	400.2111	Fold2.Rep3
764.6453	0.1821050	604.5446	Fold4.Rep2
683.8007	0.4836566	457.2184	Fold1.Rep2

Observando los resultados, llegamos lo siguiente:

- El modelo consigue explicar un 34.37 % de la variabilidad total del volumen de ventas para los datos de entrenamiento
- El error cuadrático medio es de 588 unidades
- Respecto al remuestreo en la validación cruzada, observamos que las métricas varían menos, pero en general son bastante pobres, llegando a obtener un R^2 mayor que 0.5 en varias ocasiones.

0.0.0.0.1.3. Algoritmo 3: Extreme Gradient Boosting (XGBoost)

Hiperparámetros del algoritmo

- Validación cruzada con 5 grupos y tres repeticiones
- Número de pruebas de hiperparametrización (tune length): 5

Modelado

```
set.seed(17)
modeloXGB_T <- train(VENTAS~.,
                     data = DatosEntreamiento_Total[,-1],
                     method = "xgbTree",
                     trControl=fitControl,
                     preProcess=c("center","scale"),
                     tuneLength=5,
                     verbosity=0)
```

Resultados

El modelo que nos ofrece mejores métricas tiene la siguiente configuración:

Tabla 5: Métricas del mejor modelo

	nrounds	max_depth	eta	gamma	colsample_bytree	min_child_weight	subsample
16	50	1	0.3	0	0.6	1	0.875

Tabla 6: Métricas del mejor modelo

	eta	max_depth	gamma	colsample_bytree	min_child_weight	subsample	nrounds	R
16	0.3	1	0	0.6	1	0.875	50	704

Métricas del remuestreo:

Tabla 7: Métricas en el remuestreo

RMSE	Rsquared	MAE	Resample
416.1331	0.9108932	286.9915	Fold5.Rep3
476.0265	0.7648500	392.3498	Fold5.Rep1
602.7303	0.5498719	404.4526	Fold2.Rep1
499.5408	0.6833817	319.4494	Fold2.Rep2
1163.4245	0.3813007	500.0996	Fold1.Rep1
1250.4217	0.3267276	657.1214	Fold3.Rep3
1263.8394	0.2744380	611.5463	Fold5.Rep2
655.3023	0.3285951	453.6916	Fold4.Rep1
517.2751	0.6892034	403.7047	Fold3.Rep2
545.6449	0.6827815	445.6331	Fold1.Rep2
661.9835	0.5478931	468.9270	Fold3.Rep1
571.9472	0.5113387	460.8252	Fold4.Rep2
625.9301	0.5378184	500.1739	Fold1.Rep3
698.6595	0.3128300	511.6066	Fold2.Rep3
624.3663	0.5427556	464.9745	Fold4.Rep3

Con estos resultados:

- El mejor modelo consigue explicar un 53.63 % de la variabilidad total del volumen de ventas para los datos de entrenamiento
- El error cuadrático medio es de 705 unidades
- Respecto al remuestreo en la validación cruzada, vuelve a ocurrir como con el primer modelo, existe mucha variabilidad en las métricas. En ocasiones el modelo predice muy bien el volumen de ventas y en otros casos lo hace bastante mal. El coeficiente de determinación varía entre un valor de 0.274438 y 0.9108932, por lo que las predicciones no tienen ninguna fiabilidad.

0.0.0.0.1.4. Prueba de los modelos en los datos de testeo y elección del modelo final

Configuramos los tres modelos con los mejores hiperparámetros y mostramos a continuación una tabla con el coeficiente de determinación y el error cuadrático medio de los tres modelos para poder seleccionar un modelo óptimo que aplicar a los datos de testeo:

Modelo	RMSE	R2
SVM	587.6893	0.6561937
KNN	846.7828	0.3436966
XGBoost	462.5869	0.5400056

Observando la tabla, el modelo seleccionado para predecir el volumen total de ventas en los datos de testeo es el modelo XGBoost, ya que a pesar de tener un valor peor del coeficiente de determinación, es un modelo más robusto que el resto y por tanto las métricas serán más fiables y además tiene menor valor del RMSE.

Predicción del volumen total de ventas para el conjunto de datos test

Tabla 8: XGBoost

Fecha	Predicción	Valor real	Error absoluto en la predicción
2020-08-03	2373	2408	35
2020-08-05	681	1638	957
2020-08-15	892	398	494
2020-08-18	2082	1947	135
2020-08-19	2033	1437	596
2020-08-20	988	1460	472
2020-08-22	2342	1983	359
2020-08-23	-90	155	245
2020-08-24	2494	2294	200
2020-08-27	2310	1766	544
2020-09-05	2422	2325	97
2020-09-10	2260	2050	210
2020-09-15	2153	2218	65
2020-09-17	2301	1969	332
2020-09-28	2494	2608	114
2020-10-03	2863	2968	105
2020-10-08	2408	2126	282
2020-10-18	230	192	38
2020-10-19	2413	2248	165
2020-10-22	1016	1811	795
2020-10-25	230	258	28
2020-10-28	2473	2386	87
2020-11-04	2073	2307	234
2020-11-10	1792	2000	208
2020-11-16	2414	2397	17
2020-12-12	2542	2860	318
2020-12-14	1101	2143	1042
2020-12-17	2260	6969	4709
2020-12-21	2494	2103	391
2020-12-29	2112	2770	658
2020-12-30	2430	2637	207
2021-01-09	2812	2546	266
2021-01-11	2683	2492	191
2021-01-25	2505	2407	98
2021-01-26	2700	2094	606
2021-01-29	2272	2621	349

Nota: se ha obtenido en una ocasión una predicción de ventas negativo.

Las métricas obtenidas trás entrenar el modelo en los datos de testeo son las siguientes:

	Métricas
RMSE	880.6117981
Rsquared	0.3699773
MAE	434.6944444

El modelo *extreme gradient boosting* explica un 37 % de la variabilidad total del volumen de ventas en los datos de testeo. Esta métrica ha empeorado, indicando que el modelo no ha sabido generalizar bien con datos nuevos. El RMSE tiene un valor demasiado alto para el volumen de ventas diario.

0.0.0.0.2. Predicción del volumen de ventas del producto con calcio

División de los datos en entrenamiento y testeo

Tomamos una partición de 80 % 20 % para los datos de entrenamiento y testeo:

```
set.seed(17)
indices <-
  createDataPartition(VolumenVentas_CALCIO$VENTAS, p = .8, list = FALSE)

# Datos de entrenamiento
DatosEntreamiento_Calcio <- VolumenVentas_CALCIO[indices,]
# Datos de testeo
DatosTesteo_Calcio<-VolumenVentas_CALCIO[-indices,]
```

0.0.0.0.2.1. Algoritmo 1: Máquina de vector soporte (SVM)

Hiperparámetros del algoritmo

- Validación cruzada con 5 grupos y tres repeticiones
- Parámetro de costo, C: malla de valores entre 1 y 3. Este parámetro penaliza al modelo por cometer errores. Cuanto mayor sea su valor, menos probable es que el algoritmo realice una predicción errónea.

Modelado

```
# Malla para hiperparámetros
# SVMGrid <-  expand.grid(C = seq(1,3, length = 20))

set.seed(17)
modeloSVM_C <- train(VENTAS~.,
                 data = DatosEntreamiento_Calcio[,-1],
                 method = "svmLinear",
                 trControl=fitControl,
                 preProcess=c("center","scale"),
                 tuneGrid = SVMGrid)
```

Resultados

El modelo con un costo $C = 2.5789474$ es el que proporciona mejores métricas:

Tabla 9: Métricas del mejor modelo

	C	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
16	2.578947	248.195	0.7074759	150.598	87.34702	0.1902624	44.66165

Métricas del remuestreo:

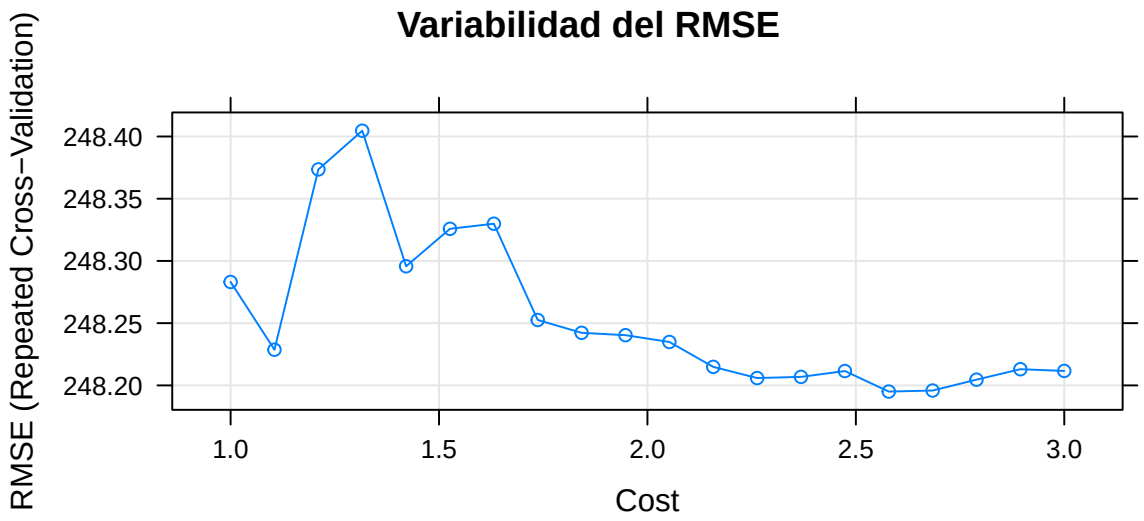
Tabla 10: Métricas en el remuestreo

RMSE	Rsquared	MAE	Resample
240.9431	0.7528666	153.28714	Fold1.Rep1
257.0387	0.7416743	144.19956	Fold2.Rep1
269.7712	0.6841806	129.43102	Fold2.Rep3
381.1070	0.4206277	235.75784	Fold4.Rep3
270.1446	0.6633964	136.77486	Fold3.Rep1
126.3598	0.9291042	101.35360	Fold5.Rep1
128.2822	0.9326806	108.14283	Fold2.Rep2
366.0838	0.3287901	217.53158	Fold4.Rep2
128.7852	0.9151760	99.38765	Fold1.Rep3
301.9259	0.6274851	180.33924	Fold3.Rep3
136.9206	0.9282832	107.51334	Fold5.Rep3
361.4197	0.4686916	221.85848	Fold4.Rep1
292.6800	0.6331110	160.02718	Fold1.Rep2
228.2525	0.7868135	135.70777	Fold3.Rep2
233.2111	0.7992582	127.65854	Fold5.Rep2

Observando los resultados, llegamos lo siguiente:

- El modelo consigue explicar un 70.75 % de la variabilidad total del volumen de ventas para los datos de entrenamiento, siendo por tanto el mejor modelo obtenido hasta el momento
- El error cuadrático medio es de 248 unidades
- Respecto al remuestreo en la validación cruzada, la variabilidad no es tan evidente como para los otros modelos, pero si es considerable.

En el gráfico mostrado a continuación, se muestra la variabilidad del error cuadrático medio en función del valor de costo:



Del gráfico podemos concluir que a medida que el costo es mayor, el error cuadrático medio disminuye considerable.

0.0.0.0.2.2. Algoritmo 2: K-Nearest Neighbor Regression (KNN)

Hiperparámetros del algoritmo

- Validación cruzada con 5 grupos y tres repeticiones
- Número de vecinos, k: malla para 3,5,7 y 9

Modelado

```
# Malla para hiperparámetros
# KNNGrid <-  expand.grid(k = seq(3,9, by=2))

set.seed(17)
modeloKNN_C <- train(VENTAS~.,
                 data = DatosEntreamiento_Calcio[,-1],
                 method = "knn",
                 trControl=fitControl,
                 preProcess=c("center","scale"),
                 tuneGrid = KNNGrid)
```

Resultados

El modelo que nos ofrece mejores métricas utiliza 3 vecinos, $K = 3$:

Tabla 11: Métricas del mejor modelo

k	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
3	451.9872	0.2839112	318.5748	71.43604	0.1648509	58.45925

Métricas del remuestreo:

Tabla 12: Métricas en el remuestreo

RMSE	Rsquared	MAE	Resample
572.3235	0.0315764	416.3851	Fold1.Rep1
500.3352	0.1307151	349.1220	Fold2.Rep1
451.5164	0.3330137	277.1500	Fold3.Rep1
553.1287	0.0651621	409.0067	Fold3.Rep3
385.4063	0.4734255	273.7167	Fold5.Rep2
537.6554	0.3046121	400.4350	Fold2.Rep2
474.8857	0.1482071	329.3592	Fold4.Rep1
435.5522	0.3269406	334.0804	Fold4.Rep3
330.7681	0.4613159	243.7298	Fold1.Rep3
474.3070	0.3334474	341.9268	Fold3.Rep2
348.6624	0.5887024	222.8306	Fold5.Rep1
378.0183	0.4506251	273.1962	Fold5.Rep3
449.1638	0.2875217	296.2782	Fold2.Rep3
430.7202	0.1305150	296.2417	Fold4.Rep2
457.3643	0.1928885	315.1644	Fold1.Rep2

Observando los resultados, llegamos lo siguiente:

- El mejor modelo consigue explicar un 28.39 % de la variabilidad total del volumen de ventas para los datos de entrenamiento
- El error cuadrático medio del mejor modelo es de 248 unidades
- Respecto al remuestreo en la validación cruzada, observamos que las métricas no presentan una gran variabilidad, pero son bastante pobres, llegando a obtener un $R^2 = 0.47$ en una ocasión, aunque también se obtienen valores menores que 0.1 en bastantes repeticiones.

0.0.0.0.2.3. Algoritmo 3: Extreme Gradient Boosting (XGBoost)

Hiperparámetros del algoritmo

- Validación cruzada con 5 grupos y tres repeticiones
- Número de pruebas de hiperparametrización (tune length): 5

Modelado

```
set.seed(17)
modeloXGB_C <- train(VENTAS~.,
                     data = DatosEntreamiento_Calcio[,-1],
                     method = "xgbTree",
                     trControl=fitControl,
                     preProcess=c("center","scale"),
                     tuneLength=5,
                     verbosity=0)
```

Resultados

El modelo que nos ofrece mejores métricas tiene la siguiente configuración:

Tabla 13: Parámetros del mejor modelo

	nrounds	max_depth	eta	gamma	colsample_bytree	min_child_weight	subsample
285	250	1	0.4	0	0.8	1	0.625

Tabla 14: Métricas del mejor modelo

	eta	max_depth	gamma	colsample_bytree	min_child_weight	subsample	nrounds	I
285	0.4	1	0	0.8	1	0.625	250	249

Métricas del remuestreo:

Tabla 15: Métricas en el remuestreo

RMSE	Rsquared	MAE	Resample
167.9423	0.8803742	128.2799	Fold5.Rep1
285.6977	0.6485156	202.9127	Fold1.Rep2
162.4105	0.8561303	137.4504	Fold1.Rep3
242.3533	0.7596276	187.8843	Fold3.Rep2
193.3144	0.8668377	134.4847	Fold5.Rep3
345.6360	0.5017274	223.1577	Fold4.Rep3
319.9133	0.5578115	210.5213	Fold4.Rep1
187.5055	0.8596916	138.8387	Fold2.Rep2
320.7237	0.4598667	205.1731	Fold4.Rep2
242.7884	0.7248521	165.4077	Fold3.Rep1
216.9473	0.8193424	153.9177	Fold5.Rep2
252.1706	0.7163478	152.5975	Fold2.Rep3
254.6381	0.7335432	184.5502	Fold1.Rep1
258.0996	0.7225360	175.3519	Fold3.Rep3
287.3594	0.6787976	205.3636	Fold2.Rep1

Observando los resultados, llegamos lo siguiente:

- El mejor modelo consigue explicar un 71.91 % de la variabilidad total del volumen de ventas para los datos de entrenamiento
- El error cuadrático medio es de 249 unidades
- Respecto al remuestreo en la validación cruzada, se trata de un modelo bastante robusto con métricas que no oscilan tanto como en los demás modelos. El coeficiente de determinación varía entre un valor de 0.4598667 y 0.8803742, por lo que las predicciones serán más fiables que en el resto de modelos.

0.0.0.0.2.4. Prueba de los modelos en los datos de testeo y elección del modelo final

Configuramos los tres modelos con los mejores hiperparámetros y mostramos a continuación una tabla con el coeficiente de determinación y el error cuadrático medio de los tres modelos para poder seleccionar un modelo óptimo que aplicar a los datos de testeo:

Modelo	RMSE	R2
SVM	248.1950	0.7074759
KNN	451.9872	0.2839112
XGBoost	173.6163	0.7178589

Observando la tabla, el modelo seleccionado para predecir el volumen de ventas para el producto con calcio en los datos de testeo es de nuevo el arbol de regresión XGBoost, ya que ambas métricas se obtienen mejores resultados.

Predicción del volumen total de ventas para el conjunto de datos test

Tabla 16: SVM

Fecha	Predicción	Valor real	Error absoluto en la predicción
2020-08-05	746	820	74
2020-08-10	1189	1063	126
2020-08-14	1094	1080	14
2020-08-17	901	1255	354
2020-08-19	977	805	172
2020-08-21	799	873	74
2020-08-23	-44	52	96
2020-08-24	1195	1227	32
2020-08-27	1042	847	195
2020-08-30	-44	86	130
2020-09-02	1181	1173	8
2020-09-05	1281	1157	124
2020-09-17	1235	1045	190
2020-10-08	1198	1007	191
2020-10-09	1280	1150	130
2020-10-13	1223	1604	381
2020-10-15	1257	4643	3386
2020-10-20	1181	891	290
2020-10-22	930	898	32
2020-10-23	1272	1203	69
2020-10-25	141	134	7
2020-10-30	1278	1224	54
2020-11-02	1106	1280	174
2020-11-04	952	1134	182
2020-11-10	1207	954	253
2020-11-20	1293	1055	238
2020-12-12	1370	1487	117
2020-12-17	1213	3979	2766
2020-12-21	1368	1067	301
2020-12-29	1239	1535	296
2020-12-30	1244	1452	208
2021-01-12	939	1443	504
2021-01-15	1009	1509	500
2021-01-21	956	1120	164
2021-01-30	1347	1564	217

Nota: se ha obtenido en una ocasión una predicción de ventas negativa.

Las métricas obtenidas trás entrenar el modelo en los datos de testeo son las siguientes:

	Métricas
RMSE	769.0996034
Rsquared	0.2305839
MAE	344.2571429

El modelo *XGBoost* explica un 23.06 % de la variabilidad total del volumen de ventas en los datos de testeo. Esta métrica ha empeorado considerablemente con respecto al entrenamiento en los datos de testeo, indicando que el modelo no ha sabido generalizar bien con datos nuevos. El RMSE tiene un valor demasiado alto para el volumen de ventas diario.

0.0.0.0.3. Predicción del volumen de ventas del producto sin calcio

División de los datos en entrenamiento y testeo

Se ha tomado una partición de 80 % 20 % para los datos de entrenamiento y testeo:

```
set.seed(17)
indices <-
  createDataPartition(VolumenVentas_SIN_CALCIO$VENTAS, p = .8, list = FALSE)

# Datos de entrenamiento
DatosEntreamiento_SinCalcio <- VolumenVentas_SIN_CALCIO[indices,]
# Datos de testeo
DatosTesteo_SinCalcio<-VolumenVentas_SIN_CALCIO[-indices,]
```

0.0.0.0.3.1. Algoritmo 1: Máquina de vector soporte (SVM)

Hiperparámetros del algoritmo

- Validación cruzada con 5 grupos y tres repeticiones
- Parámetro de costo, C: malla de valores entre 1 y 3. Este parámetro penaliza al modelo por cometer errores. Cuanto mayor sea su valor, menos probable es que el algoritmo realice una predicción errónea.

Modelado

```
# Malla para hiperparámetros
# SVMGrid <-  expand.grid(C = seq(1,3, length = 20))

set.seed(17)
modeloSVM_SC <- train(VENTAS~.,
                 data = DatosEntreamiento_SinCalcio[,-1],
                 method = "svmLinear",
                 trControl=fitControl,
                 preProcess=c("center","scale"),
                 tuneGrid = SVMGrid)
```

Resultados

El modelo que nos ofrece mejores métricas tiene un costo, $C = 3$:

Tabla 17: Métricas del mejor modelo

	C	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
20	3	317.4709	0.5775744	169.7522	144.2538	0.2323743	50.21514

Métricas del remuestreo:

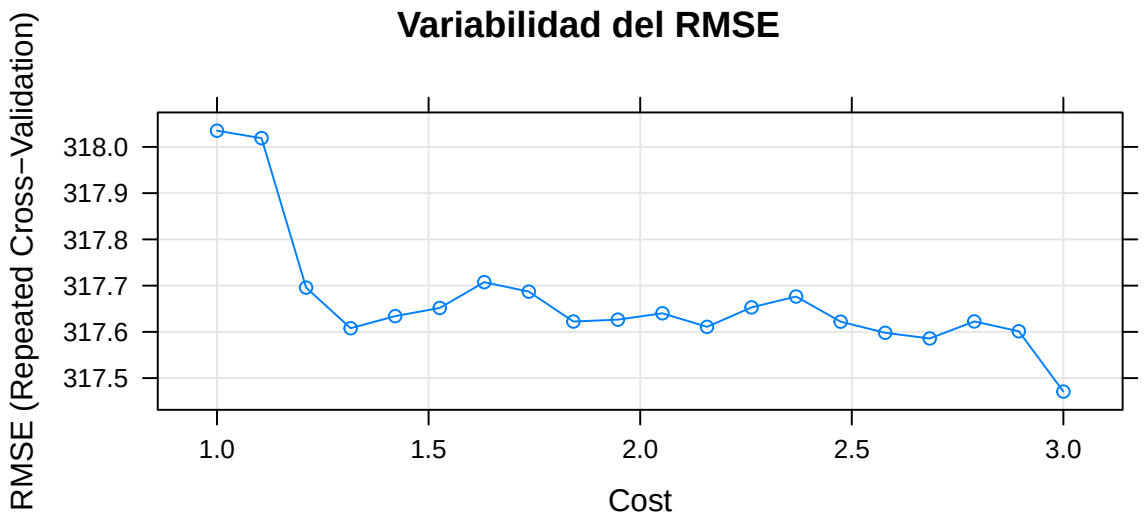
Tabla 18: Métricas en el remuestreo

RMSE	Rsquared	MAE	Resample
516.9227	0.3062750	240.50110	Fold1.Rep1
111.6519	0.9419220	89.52072	Fold2.Rep1
224.9224	0.6438748	144.34168	Fold4.Rep1
259.2031	0.6346006	150.51927	Fold1.Rep2
455.2018	0.3836770	219.40587	Fold3.Rep2
250.6692	0.6641082	173.56098	Fold5.Rep2
296.2303	0.6291907	173.05033	Fold2.Rep3
206.0004	0.7923622	134.12053	Fold4.Rep3
217.3812	0.7448938	127.38547	Fold3.Rep1
478.5237	0.2795861	240.73049	Fold5.Rep1
114.7170	0.9190959	96.70108	Fold2.Rep2
496.9208	0.3411204	226.56249	Fold4.Rep2
453.7582	0.2727344	184.61141	Fold1.Rep3
213.3679	0.7350984	131.10814	Fold3.Rep3
466.5929	0.3750762	214.16352	Fold5.Rep3

Observando los resultados, llegamos lo siguiente:

- El modelo consigue explicar un 57.76 % de la variabilidad total del volumen de ventas para los datos de entrenamiento
- El error cuadrático medio es de 317 unidades
- Respecto al remuestreo en la validación cruzada, la variabilidad no es tan evidente como para los otros modelos, pero si es considerable.

En el gráfico mostrado a continuación, se muestra la variabilidad del error cuadrático medio en función del valor de costo:



0.0.0.0.3.2. Algoritmo 2: K-Nearest Neighbor Regression (KNN)

Hiperparámetros del algoritmo

- Validación cruzada con 5 grupos y tres repeticiones
- Número de vecinos, k: malla para 3,5,7 y 9

Modelado

```
# Malla para hiperparámetros
# KNNGrid <-  expand.grid(k = seq(3,9, by=2))

set.seed(17)
modeloKNN_SC <- train(VENTAS~.,
                 data = DatosEntreamiento_SinCalcio[,-1],
                 method = "knn",
                 trControl=fitControl,
                 preProcess=c("center","scale"),
                 tuneGrid = KNNGrid)
```

Resultados

El modelo que nos ofrece mejores métricas utiliza 3 vecinos, $K = 3$:

Tabla 19: Métricas del mejor modelo

k	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
3	478.3983	0.2393137	335.4121	103.1098	0.145849	64.68044

Métricas del remuestreo:

Tabla 20: Métricas en el remuestreo

RMSE	Rsquared	MAE	Resample
520.3555	0.3069627	280.2695	Fold1.Rep1
469.8273	0.2183906	340.2513	Fold2.Rep1
456.4045	0.2802971	347.9046	Fold3.Rep1
369.5701	0.3165743	263.2586	Fold3.Rep3
385.5325	0.3937653	272.9431	Fold5.Rep2
264.8136	0.6328042	203.3146	Fold2.Rep2
396.7408	0.1524737	303.6970	Fold4.Rep1
409.5578	0.3035520	301.4383	Fold4.Rep3
607.3783	0.0303997	403.6034	Fold1.Rep3
563.0386	0.1621227	385.9111	Fold3.Rep2
609.7963	0.1073408	415.9643	Fold5.Rep1
563.1325	0.2090226	384.7632	Fold5.Rep3
515.4785	0.1773736	377.5083	Fold2.Rep3
610.4333	0.0909551	427.9613	Fold4.Rep2
433.9143	0.2076708	322.3931	Fold1.Rep2

Observando los resultados, llegamos lo siguiente:

- El mejor modelo consigue explicar un 23.93 % de la variabilidad total del volumen de ventas para los datos de entrenamiento
- El error cuadrático medio del mejor modelo es de 3 unidades
- Respecto al remuestreo en la validación cruzada, observamos que las métricas no presentan una gran variabilidad, pero son bastante pobres, llegando a obtener un R^2 mayor que 0.60 en una ocasión, aunque también se obtienen valores menores que 0.1 en bastantes iteracciones.

0.0.0.0.3.3. Algoritmo 3: Extreme Gradient Boosting (XGBoost)

Hiperparámetros del algoritmo

- Validación cruzada con 5 grupos y tres repeticiones
- Número de pruebas de hiperparametrización (tune length): 5

Modelado

```
set.seed(17)
modeloXGB_SC <- train(VENTAS~.,
                 data = DatosEntreamiento_SinCalcio[,-1],
                 method = "xgbTree",
                 trControl=fitControl,
                 preProcess=c("center","scale"),
                 tuneLength=5,
                 verbosity=0)
```

Resultados

El modelo que nos ofrece mejores métricas tiene la siguiente configuración:

Tabla 21: Métricas del mejor modelo

	nrounds	max_depth	eta	gamma	colsample_bytree	min_child_weight	subsample
29	200	1	0.3	0	0.8	1	0.5

Tabla 22: Métricas del mejor modelo

	eta	max_depth	gamma	colsample_bytree	min_child_weight	subsample	nrounds	R
29	0.3	1	0	0.8	1	0.5	200	350.

Métricas del remuestreo:

Tabla 23: Métricas en el remuestreo

RMSE	Rsquared	MAE	Resample
441.0882	0.4458448	246.9654	Fold5.Rep3
236.6804	0.6849787	180.4792	Fold2.Rep2
293.3100	0.5771612	225.6992	Fold1.Rep2
280.3913	0.6758052	236.6343	Fold2.Rep3
482.4470	0.4197410	260.9571	Fold1.Rep1
323.9682	0.4277819	244.1710	Fold3.Rep3
304.4729	0.5477748	250.2880	Fold2.Rep1
242.2772	0.5970592	202.6494	Fold4.Rep1
445.8052	0.4189784	309.2103	Fold3.Rep2
270.9050	0.6837504	193.7147	Fold3.Rep1
475.0891	0.4055751	242.9997	Fold4.Rep2
239.8452	0.6885826	188.3545	Fold5.Rep2
453.2530	0.2705498	222.0091	Fold1.Rep3
438.4911	0.3937193	290.9016	Fold5.Rep1
335.2732	0.5024921	258.8112	Fold4.Rep3

Observando los resultados, llegamos lo siguiente:

- El mejor modelo consigue explicar un 51.6 % de la variabilidad total del volumen de ventas para los datos de entrenamiento
- El error cuadrático medio es de 351 unidades
- Respecto al remuestreo en la validación cruzada, se trata de un modelo robusto, ya que las métricas no oscilan tanto como en el resto de modelos. El coeficiente de determinación varía entre un valor de 0.2705498 y 0.6885826 por lo que las predicciones serán algo más robustas, a pesar de no tener un valor de R^2 especialmente elevado.

0.0.0.0.3.4. Prueba de los modelos en los datos de testeo y elección del modelo final

Configuramos los tres modelos con los mejores hiperparámetros y mostramos a continuación una tabla con el coeficiente de determinación y el error cuadrático medio de los tres modelos para poder seleccionar un modelo óptimo que aplicar a los datos de testeo:

Modelo	RMSE	R2
SVM	317.4709	0.5775744
KNN	478.3983	0.2393137
XGBoost	234.7513	0.5104031

Observando la tabla, el modelo seleccionado para predecir el volumen de ventas para el producto con sin calcio en los datos de testeo es el modelo XGBoost, ya que a pesar de obtener un valor del coeficiente de correlación ligeramente peor, se trata de un modelo más robusto, y por tanto las métricas serán más fiables. Además, el valor del RMSE es mejor.

Predicción del volumen total de ventas para el conjunto de datos test

Tabla 24: XGBoost

Fecha	Predicción	Valor real	Error absoluto en la predicción
2020-08-01	1122	1067	55
2020-08-03	927	1069	142
2020-08-05	865	818	47
2020-08-16	-43	135	178
2020-08-20	870	673	197
2020-08-21	978	766	212
2020-08-24	929	1067	138
2020-08-25	855	977	122
2020-08-26	857	931	74
2020-08-28	977	766	211
2020-08-30	-43	98	141
2020-08-31	928	1169	241
2020-09-18	1116	995	121
2020-09-26	1262	1250	12
2020-09-28	1067	1253	186
2020-09-29	992	929	63
2020-10-02	1167	1173	6
2020-10-09	1163	1095	68
2020-10-21	1056	795	261
2020-10-27	1040	1173	133
2020-11-01	122	85	37
2020-11-06	1142	1031	111
2020-11-08	122	72	50
2020-11-09	1070	1145	75
2020-11-12	1014	978	36
2020-11-28	1346	1461	115
2020-12-04	1129	1213	84
2020-12-18	1129	971	158
2020-12-19	1278	1255	23
2020-12-23	1009	1312	303
2020-12-24	1020	984	36
2021-01-05	1000	1433	433
2021-01-07	1028	1193	165
2021-01-22	1123	1154	31
2021-01-26	1000	934	66
2021-01-29	1205	1294	89

Nota: se ha obtenido en dos ocasiiones una predicción de ventas negativa.

Las métricas obtenidas trás entrenar el modelo en los datos de testeo son las siguientes:

	Métricas
RMSE	152.432499
Rsquared	0.822966
MAE	122.777778

El modelo *XGBOost* explica un 82.3 % de la variabilidad total del volumen de ventas en los datos de testeo. El resultado obtenido es bastante óptimo y el es el mejor de todo el modelado. El modelo ha sabido generalizar bastante bien con datos nuevos.