STORE SALES - TIME SERIES FORECASTING

Juan Felipe Vásquez Uribe
Jhon David Ballesteros Vargas
Natalia Polo Peña

PROFESOR:
Raúl Ramos Pollán

CURSO:
Intro a la Inteligencia Artificial para ciencias e Ingeniería

Facultad de Ingeniería
Medellín
Mayo, 2023

Índice

1. Planteamiento del problema **3**
1.1. Métrica de desempeño . . . 3
1.2. Objetivos . . . 3
1.2.1. Objetivo General . . . 3
1.2.2. Objetivos Específicos . . . 3

2. Exploración de Datos **4**

3. Tratamiento de Datos **6**
3.1. Construcción del Dataset de Trabajo . . . 6
3.2. Tratamiento de datos faltantes . . . 6
3.3. Partición de los Datos . . . 7

4. Modelos Supervisados **8**
4.1. Random Forest Regressor . . . 8
4.1.1. Mejores hiperparámetros . . . 8
4.1.2. Desempeño alcanzado . . . 9
4.2. Support Vector Regressor . . . 9
4.2.1. Mejores hiperparámetros . . . 10
4.2.2. Desempeño alcanzado . . . 10
4.3. SARIMAX . . . 11
4.3.1. Mejores hiperparámetros . . . 11
4.3.2. Desempeño alcanzado . . . 12

5. Modelos No supervisado + Supervisado **12**
5.1. PCA+RFR . . . 13
5.1.1. Mejores hiperparámetros . . . 13
5.1.2. Desempeño alcanzado . . . 14
5.2. PCA+SVR . . . 14
5.2.1. Mejores hiperparámetros . . . 15
5.2.2. Desempeño alcanzado . . . 15

6. Curvas de Aprendizaje **15**

7. Retos y consideraciones de despliegue **16**

8. Conclusiones **17**

1. Planteamiento del problema

Pronosticar la demanda de productos es una tarea común para científicos de datos actualmente. En efecto, un pronóstico más preciso de ventas o tendencias de consumo por medio del aprendizaje automático, permite a las empresas tomar mejores decisiones de marketing o estrategias de mercadeo. Permite visualizar de manera mías detallada y óptima las necesidades y tendencias de los clientes en el mercado.

Bajo esta iniciativa el reto de este proyecto consiste en predecir las venta en un periodo de tiempo determinados de varias tiendas de la corporación Favorita, quien es una de las grandes minoristas de la distribución de dulces y golosinas en Ecuador. El reto originalmente surge de una de las competencias de la reconocida plataforma Kaggle: Store Sales - Time Series Forecasting Competition. Aunque el alcance de este proyecto no es participar en la competencia desde esa plataforma fueron obtenidos los datos, métricas de desempeño y guías para la resolución del problema.

1.1. Métrica de desempeño

La métrica de desempeño de la competencia es el error medio es Raíz Error logarítmico cuadrático medio o RMSLE por sus siglas en inglés. Definido como:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\log(1+\hat{y}_i)-\log(1+y_i)\right)^2} \tag{1}$$

Esta misma será la métrica de desempeño que se usará al evaluar las predicciones de los modelos entrenados y ejecutados. Se ha determinado que resultados muy buenos serán $RMSLE < 0{,}01$, resultados aceptable serán entre $0{,}1 < RMSLE < 0{,}5$ y resultados poco aceptables $RMSLE > 0{,}5$

1.2. Objetivos

1.2.1. Objetivo General

Entrenar modelos supervisados y no supervisados para predecir valores futuros en un problema de Series de Tiempo

1.2.2. Objetivos Específicos

- Tener el primer acercamiento a un problema de Machine Learning de predicciones de series de tiempo
- Realziar limpieza y procesamiento de los datos brindados por la competencia
- Encontrar los mejores parámetros, curva de aprendizaje y evaluar desempeño de dos modelos supervisados
- Encontrar los mejores parámetros, curva de aprendizaje y evaluar desempeño de dos combinaciones de modelo supervisado y no supervisado

2. Exploración de Datos

Al momento de obtener la Data de la competencia se poseen varios archivos que serán de utilidad a lo largo de la resolución del problema. La base de datos principal es el archivo denominado `train.csv`. Este posee los registros de las ventas y la cantidad de productos en promoción de cada día, de cada tienda, de cada familia de producto ofrecidos. En la tabla 1 se observa un resumen de la información de esta base de datos.

Stores	54
Families	33
Días analizados	1684
Fecha Inicial	2013-01-01
Fecha Final	2017-08-15
Registros totales	3′000,888

Tabla 1: Resumen información archivo train.csv.

La variable objetivo a predecir son las ventas o `sales` por lo que vale la pena graficar su relación en el tiempo. En la figura 1 se realizó un grafico interactivo donde se visualiza el total de ventas por día. Es importante rescatar que se sigue cierto patrón de ventas a lo largo del año. Esta hipótesis será fundamental en el procesamiento de datos.

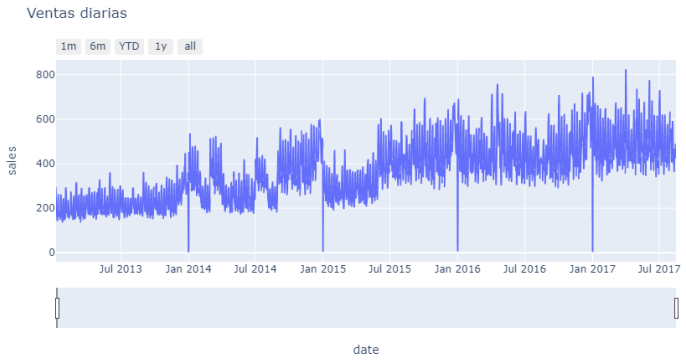


Figura 1: Suma de ventas por día.

A fin de definir el número de lags en ventas a implementar en el modelo, se graficó la correlación entre las ventas de un día y sus días pasados. En la figura 2 se observa que la mayor correlación se encuentra 7 días antes, es decir se encuentran patrones de ventas por semana.

Una segunda base de datos proporcionada es `holidays_events.csv` quien presenta un listado con los días festivos o especiales de Ecuador para las fechas presentadas. Presenta una columna del tipo de Festividad: evento, puente día de trabajo, día feriado o celebración transferida. Otra columna que indica si es una celebración de índole local, regional o nacional.

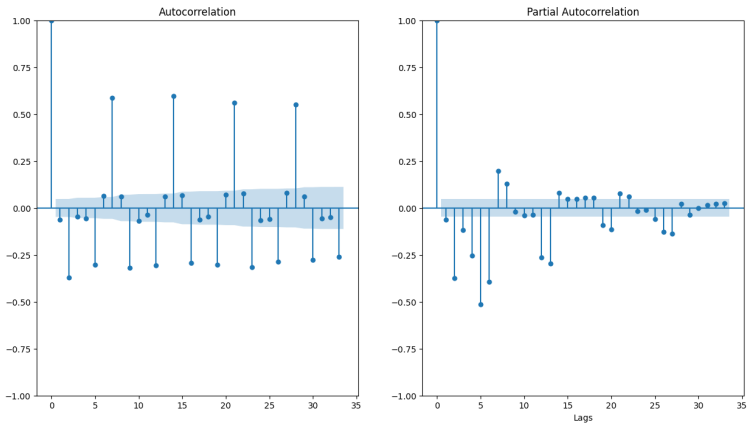
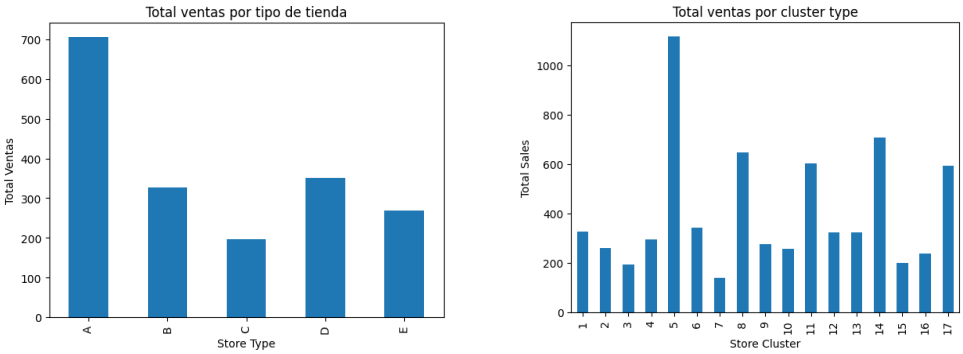


Figura 2: Correlación de ventas con días pasados.

Otra de las bases de datos con la que se cuenta es `stores.csv` . En esta se presenta un listado de todas las tiendas y su clasificación por tipo y clúster según los artículos vendidos. En la figura 3 se observan las ventas que se realizan por el tipo de tienda y el clúster al que pertenecen, deduciendo que pueden ser un factor importante a la hora de predecir las ventas. También se cuenta con el archivo `transactions.csv` que indica el numero de facturas realizadas por día en una tienda. Dada su relación en el tiempo, esta variable será incluida como lag dentro del dataset a formar para el procesamiento de datos.



(a) Ventas por tipo de tienda. (b) Ventas por culster de tienda.

Figura 3: Análisis de ventas en tiendas.

Finalmente se cuenta con la data `oil.csv` que proporciona el precio del barril de petroleo para los días analizados. Este es una archivo incluido en la data proporcionada gracias a la dependencia de Ecuador con la economía petrolera. Se buscó la correlación entre el precio del petroleo y las ventas (figura 4) obteniendo como resultado $-0{,}079$ al ser negativo indica que existe una relación inversa entre la suma de las ventas de un día y el precio del petroleo de ese día. Aunque no es una correlación fuerte, no se descarta totalmente pues es posible (como se demostrará luego) que las ventas no dependen unicamente del precio del día en cuestión sino de días anteriores.

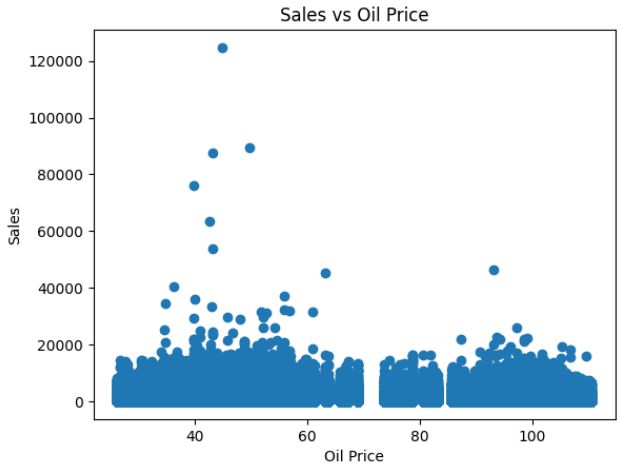


Figura 4: Relación entre las ventas por día y el precio del petroleo .

3. Tratamiento de Datos

3.1. Construcción del Dataset de Trabajo

Originalmente se poseen las ventas por familia de artículo (33 familias distintas), por tienda (54 tiendas distintas), por día (desde el 2013-01-01 hasta el 2017-08-15) con un total de 3′000,888 de filas. Para cumplir los alcances de este proyecto se analizarán solo la suma de las ventas por tienda, por día, en las mismas fechas, es decir, se dejará a un lado en análisis de las ventas por familia de productos. De esta forma se recoge registro un total de 90,204 filas.

Basado en el análisis y exploración de la sección anterior, se pretende construir un sola base de datos que recoja toda la mayor parte de la información de los 5 archivos proporcionados en la competencia de Kaggle.

En primera instancia se crearon nuevos descriptores relacionados con el tiempo para la ventas, como lo son: día de la semana, año, si es fin de semana, semana del año, mes, si es inicio o fin de mes, si es inicio o fin de año, si es fecha de pago. También se llevó a binario las instancias del tipo de festividad si es local, regional o nacional.

Por otro lado, se crearon lags o retrocesos para las variables de ventas, precio del petroleo y transacciones realizadas en ese día. Para las ventas se tienen en cuenta los días anteriores, para el precio del petroleo y las transacciones se tienen en cuentan los 3 días anteriores. Finalemnte se obtiene un Dataset de trabajo con 33 columnas y 90,204 filas.

3.2. Tratamiento de datos faltantes

En el procesamiento del archivo `oil.csv` se encontró que 43 fechas no contaban con el precio del petroleo, para eso se decidió llenar este valor con el promedio del precio los 3 días anteriores

Por otro lado, una vez construido el Dataset de trabajo, se empezará en análisis de predicciones por tiendas. En este sentido, se evidencia que no se cuenta con registros de

la tienda 52 antes del 2017-04-20. Es posible que esta tienda solo se haya abierto después de la fecha mencionada. Al no tener registros de ventas, promociones o transacciones, el análisis para esta tienda es prácticamente desierto, pues los pocos registros que se tienen no alcanzan a cubrir la cantidad necesaria de datos para un entrenamiento que será usado para predicciones. Para dar frente a esta situación, se comparó las ventas, las promociones y las transacciones de la tienda 52 en las fechas que se tiene registro con las demás tiendas, y se le asignaron los valores de la tienda que presenta mayor similitud. Obteniendo como resultado:

- Remplazar datos faltantes de ventas por datos de tienda 8
- Remplazar datos faltantes de promociones por datos de tienda 50
- Remplazar datos faltantes de transacciones por datos de tienda 34

Como se evidenciará en el transcurso de este informe, esta estrategia dio buenos resultados, al punto de que se obtienen mejores desempeños de predicciones de la tienda 52 (que un principio poseía la mayoría de datos faltantes) que algunas tiendas que poseían toda su data completa

3.3. Partición de los Datos

En la comptencia original de Kaggle se proporcianan datos de train desde el 2013-01-01 hasta el 2017-08-15 y un dataset de test desde el 2017-08-16 al 2017-08-30, desde luego este no posee el valor de las ventas pues es la variable que los competidores deben predecir. Como el objetivo de este proyecto no es la participación en la competencia, solo se tendrán en cuentan los registros de `train.csv` pues es quien posee el total de ventas. De esos registros se decidió, basado en el análisis de ventas a lo largo del año que la fecha de corte para el propio `train` y `validation` es 2017-03-01 (para evitar ambiguedades de lenguaje se seguirá llamando `validation`). De esta forma:

	Inicio	Fin	Filas
train	2013-01-01	2017-03-01	81,216
test	2013-01-02	2017-08-15	9,018

Tabla 2: Dataset train y test del proyecto

Para la evaluación del desempeño de los modelos, se harán análisis y predicciones por tiendas, esto a fin de buscar la facilidad en cuanto al menejo de las series de tiempo por días. De esta manera se filtran los datos por una tienda en especifico se realiza calibración, predicciones, se evalúa desempeño y se construye curva de aprendizaje para un tienda en especifico. Luego se replica el mismo procedimiento a las demás tiendas. Como se demostrará luego, en la mayoría de modelos estudiados esta técnica dio buenos resultados

4. Modelos Supervisados

A fin de predecir las ventas por tienda en las fechas descritas anteriormente, se implementaran 3 módelos supervisados: Random Forest Regressor (RFR), Support Vector Regressor (SVR) y ARIMAX. En cada uno de ellos se empieza el análisis para una sola tienda, encontrando los mejores hiperparámetros, se calibra el modelo, se realizan predicciones y se evalúa el desempeño. El análisis se extiende luego para todas las tiendas.

4.1. Random Forest Regressor

El modelo Random Forest es un modelo de aprendizaje automático que utiliza un conjunto de árboles de decisión para realizar predicciones. El Random Forest se compone de múltiples árboles de decisión, donde cada árbol se entrena de forma independiente. La cantidad de árboles en el conjunto se determina por el usuario. Como parámetros a analizar se tiene `n_estimators` para determinar el número de árboles, `max_depth` para determinar la profundidad del árbol, `min_samples_split` para el número mínimo de muestras necesarias para dividir un nodo interno, `min_samples_leaf` el número mínimo de muestras requeridas para estar en un nodo y `max_features` para el número de características a considerar al buscar la mejor división [1].

4.1.1. Mejores hiperparámetros

La búsqueda de los mejores parametros para este modelo se obtuvo, haciendo la iteración de cada una de las posibles combinaciones definidas y evaluando el desempeño de predicciones con cada uno de ellos. La figura 5 muestra los mejores resultados obtenidos.

n_estimator	max_feature	max_depth	min_samples_split	min_samples_leaf	RMSLE
150	sqrt	12	2	1	0.072115
200	sqrt	12	2	1	0.072241
100	sqrt	12	5	1	0.072406
300	sqrt	15	5	2	0.072616
150	sqrt	12	5	1	0.072712

Figura 5: Desempeños obtenidos en la búsqueda de hiperparametros para el modelo RFR.

De esta forma tenemos que los parámetros que se usarán en la calibración del módelo son

- `n_estimators=150`
- `max_depth=9`
- `max_features='sqrt'`
- `min_samples_split=2`
- `min_samples_leaf=1`

4.1.2. Desempeño alcanzado

La figura 6 muestra la predicciones en ventas obtenidas por el modelo, comparado con las ventas verdaderas que se alcanzaron. La grafica muestra un gran acercamiento y similitud de las perdiciones con la realidad.

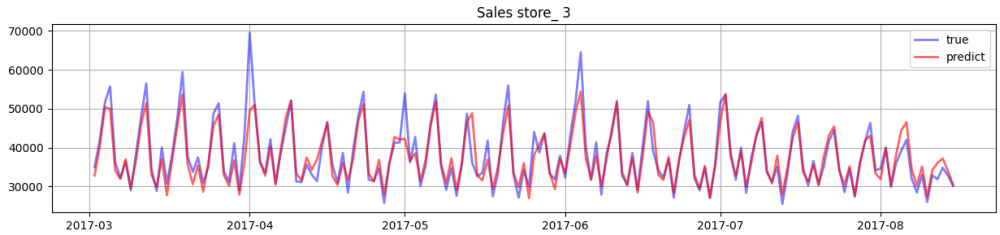


Figura 6: Predicción de ventas del módelo RFR.

Cuando se realza la predicción de ventas para todas las tiendas se obtiene un desempeño promedio de $RMSLE = 0,1197$. Según los criterios establecidos al inicio del proyecto es considerado un desempeño bueno. La figura 7 muestra el RMSLE alcanzado en la predicción de todas las tiendas. Es importante notar que todas las tiendas se comportarón de manera similar, hay poca varianza de los resultados entre ellas

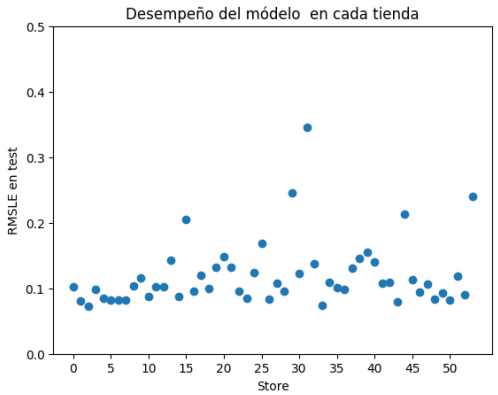


Figura 7: Predicción de ventas del módelo RFR.

4.2. Support Vector Regressor

Antes de aplicar SVR, es común aplicar una transformación a los datos de entrada y salida. Esto puede incluir escalado de características y normalización para asegurar que todas las variables estén en la misma escala y tengan una distribución más adecuada para el modelo. El SVR utiliza un enfoque basado en el kernel para mapear los datos a un espacio de mayor dimensión donde sea más fácil encontrar una función de regresión lineal. El kernel especifica la forma de la función de mapeo y puede ser lineal, polinómico, radial (RBF) u otro tipo. SVR tiene varios hiperparámetros que deben ser ajustados. Estos incluyen el parámetro de regularización C, que controla el equilibrio entre el ajuste de los datos de entrenamiento y la complejidad del modelo, y el parámetro de kernel, que afecta la forma de la función de

regresión. El objetivo del SVR es encontrar una función de regresión que minimice el error de predicción mientras cumple con una tolerancia predefinida [2]

4.2.1. Mejores hiperparámetros

Al igual que en modelo RFR se procedió a iterar hasta encontrar los parámetros donde el error sea mínimo.

C	gamma	epsilon	RMSLE
50	0.001	0.100	0.120633
50	0.001	0.001	0.121202
100	0.001	0.100	0.121454
100	0.001	0.001	0.121477
100	0.001	0.010	0.121508

Figura 8: Desempeños obtenidos en la búsqueda de hiperparametros para el modelo SVR.

De esta forma tenemos que los parámetros que se usarán en la calibración del módelo son

- `C=50`
- `gamma=0.001`
- `epsilon=0.1`

4.2.2. Desempeño alcanzado

La figura 9 muestra la predicciones en ventas obtenidas por el modelo, comparado con las ventas verdaderas que se alcanzaron. Al igual, que en el modelo RFR, la grafica muestra un gran acercamiento y similitud de las perdiciones con la realidad.

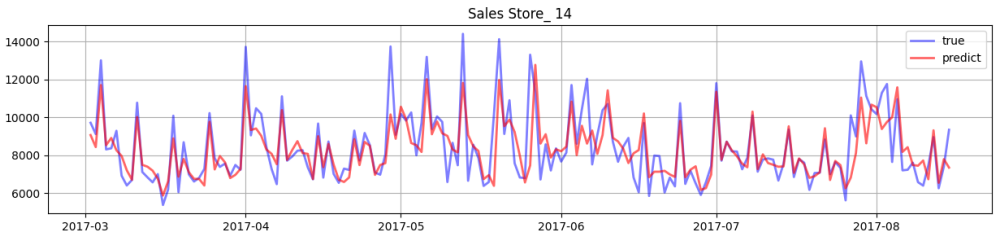


Figura 9: Predicción de ventas del módelo SVR.

Cuando se realza la predicción de ventas para todas las tiendas se obtiene un desempeño promedio de $RMSLE = 0,1311$. Según los criterios establecidos al inicio del proyecto es considerado un desempeño bueno. La figura 10 muestra el RMSLE alcanzado en la predicción de todas las tiendas. Es importante notar que todas las tiendas se comportarón de manera similar, hay poca varianza de los resultados entre ellas. El desempeño obtenido por por RFR en contraparte con el SVR son muy similares, siendo ligeramente mejor el primero.

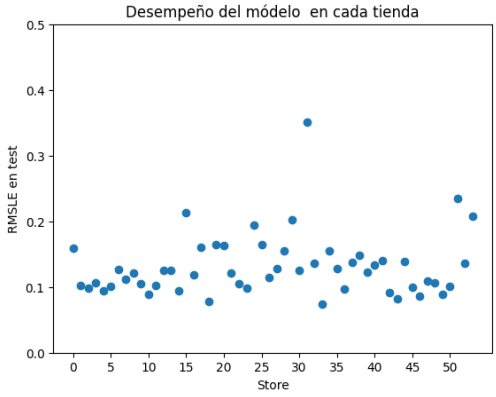


Figura 10: Predicción de ventas del módelo SVR.

4.3. SARIMAX

Seasonal AutoRegressive Integrated Moving Average with exogenous variables es un modelo estadístico utilizado para el análisis y la predicción de series de tiempo. Combina componentes ARIMA (AutoRegressive Integrated Moving Average) con componentes de estacionalidad y permite la inclusión de variables exógenas. En muchas series de tiempo, hay patrones estacionales que se repiten a intervalos regulares (por ejemplo, patrones mensuales o estacionales). El modelo SARIMAX incorpora términos de estacionalidad para modelar y ajustar estos patrones estacionales.

La inclusión de variables exógenas, también conocidas como variables predictoras o variables regresoras son características externas que pueden influir en la serie de tiempo y ayudar a mejorar las predicciones. Para este dataset, se trabajó de dos formas con este modelo: la primera, incluyendo variables exógenas (todas las variables del Data Frame incluidos los lags), y la segunda sin hacer uso de variables exógenas.

4.3.1. Mejores hiperparámetros

El parámetro "p"se refiere al orden del componente autorregresivo (AR) del modelo. Representa la cantidad de pasos de tiempo anteriores que se tienen en cuenta al predecir el valor actual de la serie de tiempo. Un valor de "p"mayor indica que se están considerando más pasos de tiempo anteriores en la predicción.

El parámetro "d"se refiere al orden de diferenciación (diferenciación integrada) del componente integrado del modelo. La diferenciación se utiliza para transformar una serie de tiempo no estacionaria en una serie estacionaria. Un valor de "d"mayor indica que se están aplicando más diferencias a la serie de tiempo para lograr la estacionariedad.

El parámetro "q"se refiere al orden del componente de media móvil (MA) del modelo. Representa la cantidad de pasos de tiempo anteriores que se tienen en cuenta al predecir el valor actual de la serie de tiempo, pero basándose en los errores residuales del modelo. Un valor de "q"mayor indica que se están considerando más pasos de tiempo anteriores en función de los errores residuales [3].

Para la búsqueda de estos parametros se uso la función `auto_arima ()` Obteniendo que los mejores parametros para todo el dataset son los mostrados en la figura 11. `p=7`, `d=1` `q=1`

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	1502
Model:	SARIMAX(7, 1, 1)	**Log Likelihood**	-13011.493
Date:	Sun, 28 May 2023	**AIC**	26040.986
Time:	03:12:34	**BIC**	26088.811
Sample:	0	**HQIC**	26058.802
	- 1502		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.2946	0.145	-2.034	0.042	-0.579	-0.011
ar.L2	-0.3739	0.103	-3.636	0.000	-0.575	-0.172
ar.L3	-0.3985	0.102	-3.893	0.000	-0.599	-0.198
ar.L4	-0.3929	0.109	-3.615	0.000	-0.606	-0.180
ar.L5	-0.3630	0.108	-3.352	0.001	-0.575	-0.151
ar.L6	-0.2205	0.099	-2.224	0.026	-0.415	-0.026
ar.L7	0.3240	0.078	4.163	0.000	0.171	0.477
ma.L1	-0.4024	0.144	-2.786	0.005	-0.686	-0.119
sigma2	1.987e+06	3.3e+04	60.185	0.000	1.92e+06	2.05e+06

Ljung-Box (L1) (Q):	0.07	**Jarque-Bera (JB):**	22687.81
Prob(Q):	0.79	**Prob(JB):**	0.00
Heteroskedasticity (H):	4.70	**Skew:**	-0.89
Prob(H) (two-sided):	0.00	**Kurtosis:**	21.96

Figura 11: Desempeños obtenidos en la búsqueda de hiperparámetros para el modelo RFR.

4.3.2. Desempeño alcanzado

La figura 12 representa las predicciones en ventas alcanzadas por el modelo SARIMAX. Es un modelo que sigue la tendencia regular de las ventas

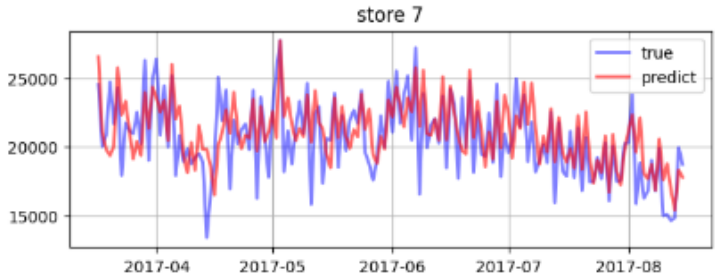


Figura 12: Predicción de ventas del módelo SARIMAX.

La figura 13 muestra la distribución del desempeño alcanzado por el modelo SARIMAX en la predicción de ventas de cada una de las tiendas. El Desemepeño promedio para este modelo fue de $RMSLE = 0,1356$, este es un valor casi identico al obtenido al modelo SVR

5. Modelos No supervisado + Supervisado

A modo de evaluación de la combinación de algortimo supervisado con no supervisado se pretende estudiar el desempeño del modelo PCA + RFR y PCA + SVR.

La combinación del modelo PCA (Análisis de Componentes Principales) con un modelo supervisado puede ser útil en ciertos casos, especialmente cuando se trabaja con conjuntos de datos de alta dimensionalidad. Después de aplicar PCA, se pueden seleccionar un número específico de componentes principales que retengan la mayor parte de la varianza en los datos. Esto ayuda a reducir la dimensionalidad del conjunto de datos y eliminar características

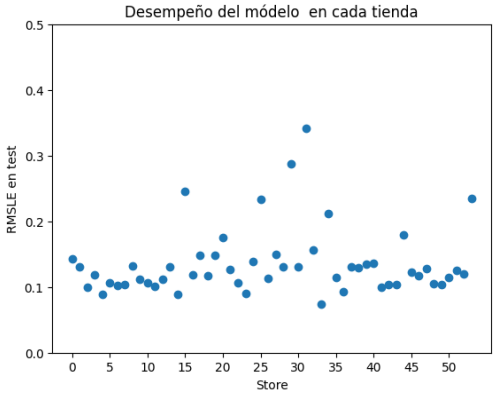


Figura 13: Desempeño en predicción de ventas del módelo SARIMAX.

redundantes o menos informativas.

De igual forma, al evaluar un híbrido de PCA con un modelo supervisado puede proporcionar varios beneficios. Al reducir la dimensionalidad del conjunto de datos, PCA puede ayudar a evitar el sobreajuste y mejorar la eficiencia computacional al reducir el número de características. Además, al eliminar características redundantes, PCA puede ayudar a mejorar la interpretación y comprensión de los resultados.

5.1. PCA+RFR

Una vez que se ha realizado la reducción de dimensionalidad con PCA y se ha seleccionado el número deseado de componentes principales, el conjunto de datos reducido se utiliza para entrenar el modelo Random Forest. Este modelo se entrena en las nuevas características generadas por PCA en lugar de las características originales.Después de entrenar el modelo Random Forest, se puede utilizar para realizar predicciones en nuevos datos, siguiendo el mismo proceso que se utilizaría con Random Forest convencional.

5.1.1. Mejores hiperparámetros

El primer paso consiste en encontrar el número de componente a los que quiere ser reducida la matriz de datos ingresados para calibrar el modelo. Usando las funciones incluida en la clase `sklearn.decomposition.PCA` se encontró que el número óptimo de componentes a reducir es 3. De esta manera, la matriz incial de datos para una tienda que contenía 33 columnas quedo reducida a 3 columnas

El siguiente paso consiste en la búsqueda de los mejores parámetros del modelo Random Forest, en este apartado se hizo uso de la clase `sklearn.model_selection.GridSearchCV` donde se obtuvo que los mejores parámetros para la predicción de todas las tiendas es:

- `n_estimators=100`
- `max_depth=6`
- `min_samples_split=2`

- `min_samples_leaf=1`

5.1.2. Desempeño alcanzado

La figura 14 muestra la predicciones en ventas obtenidas por el modelo, comparado con las ventas verdaderas que se alcanzaron. La grafica muestra un gran acercamiento y similitud de las perdiciones con la realidad.

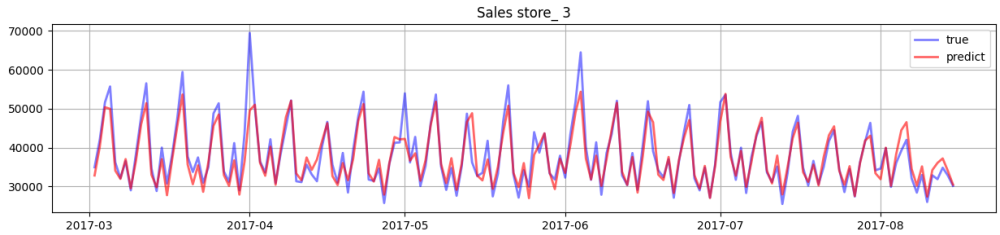


Figura 14: Predicción de ventas del módelo PCA+ RFR.

Cuando se realza la predicción de ventas para todas las tiendas se obtiene un desempeño promedio de $RMSLE = 0,1779$. Notar que aunque sigue siendo un buen resultado, es un desempeño ligeramente mayor al obtenido en el modelo RFR. La figura 15 muestra el RMSLE alcanzado en la predicción de todas las tiendas. Es importante notar que todas las tiendas se comportaron de manera similar, hay poca varianza de los resultados entre ellas

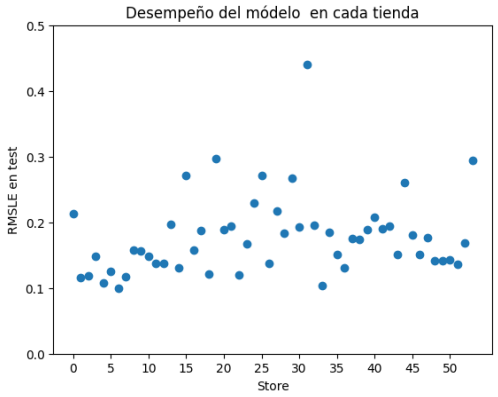


Figura 15: Predicción de ventas del módelo PCA+RFR.

5.2. PCA+SVR

En esta combinación, primero se estandarizan los datos, luego se entrena con PCA y se determina el número de componentes. Posteriormente, se transforman los datos estandarizados para llevarlos al número de columnas encontrados en el segundo paso. Se continua con la calibración utilizando el SVR con los datos estandarizados del paso anterior, se realizan predicciones y por último se realiza el inverso del primer paso para llevar los datos al rango original.

5.2.1. Mejores hiperparámetros

Se procede de manera análoga al caso anterior buscando el numero de componentes óptimos en los que debe ser transformado. Para este caso se encontró que debe ser transformado a 15 columnas.

El siguiente paso consiste en la búsqueda de los mejores hiperparámetros del modelo SVR, procediendo de manera analoga a los anteriores se encontró que los parametros a usar en la calibración del modelo son:

- `C=100`
- `gamma=0.001`
- `epsilon=0.001`

5.2.2. Desempeño alcanzado

La figura 16 muestra la predicciones en ventas obtenidas por el modelo, comparado con las ventas verdaderas que se alcanzaron.

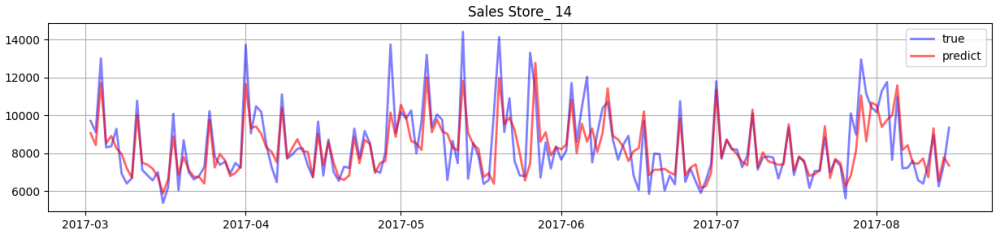


Figura 16: Predicción de ventas del módelo PCA + SVR.

Cuando se realza la predicción de ventas para todas las tiendas se obtiene un desempeño promedio de $RMSLE = 0,4608$. Notar que este modelo muestra el peor desempeño de todos los adquiridos. La figura 17 muestra el RMSLE alcanzado en la predicción de todas las tiendas. A diferencia de todos los modelos estudiados hasta el momento, se videncia variaciones significativas entre el desempeño de unas tiendas a otras. En especifico hay 4 tiendas que obtuvieron un RMSLE mauor a 1, lo que indica que seguramente los parametros elegidos para todas las tiendas, no aplican o no son los ideales para estas tiendas de malas predicciones.

6. Curvas de Aprendizaje

La figura 18 muestra curvas de aprendizaje para una tienda aleatoria de cada uno de los modelos estudiados. Esta curva se realizó para cada una de las tiendas en cada modelo, mostrando similitudes en los comportamiento de una tienda a otra, Por ejemplo los modelos SVR, PCA+SVR y SARIMAX muestran la curva de aprendizaje deseada donde hay una convergencia entre el desempeño de train y test. Sin mebargo también es de notar que existe bias en estas curvas debido a la gran dispariedad entre el desempeño de train y test cuando

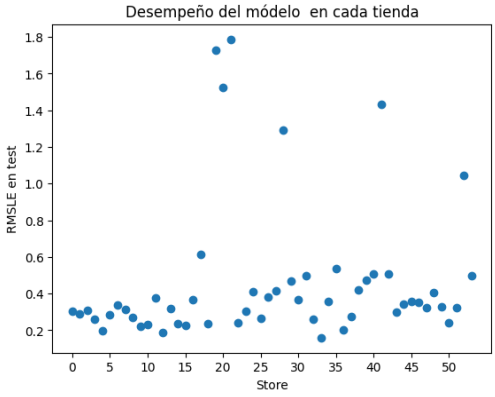
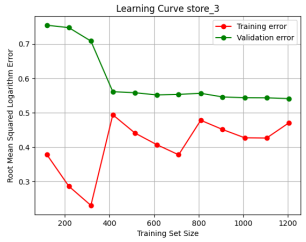
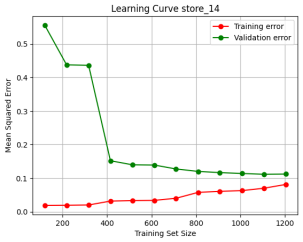


Figura 17: Predicción de ventas del módelo PCA+SVR.

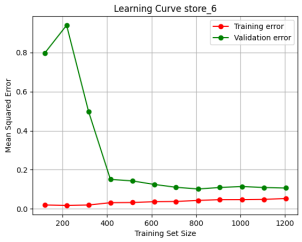
hay pocos datos de entrenamiento. Este comportamiento puede deberse princiapalemnte a que los datos se encuentran demasiado mezclados o que el modelo es demasido simple para el problema en cuestión



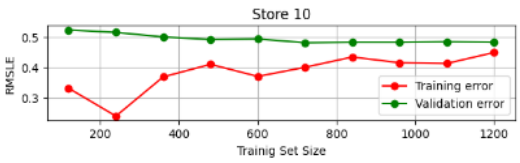
(a) Modelo RF.



(b) Modelo SVR.



(c) Modelo PCA+ SVR.



(d) Modelo PCA+SVR.

(e) Modelo SARIMAX.

Figura 18: Curva de Aprendizaje para los modelos analizados.

7. Retos y consideraciones de despliegue

El primer reto al que se enfrentó el equipo fue la construcción del dataset de trabajo. Las bases de Datos proporcionadas por la competencia eran ajenas unas de ellas, por lo que era tarea del analista construir una Base que agrupara todos los datos, en primera instancia no se quiso trabajar con la totalidad de registros de ventas por días, por tiendas, por familia de producto pues requería un nivel de experiencia y dominio mayor. Al intentar trabajar unicamente con la suma de ventas por días se identificó que se obtiene un dataset con pocas filas (alrededor de 1500) lo cual no cumpliría con los requerimientos del proyecto. También fue una gran tarea la construcción de las demás columnas que no venían por defecto en los

datos proporcionadas, se construyeron a base de la intución y análisis de los datos.

El segundo reto vino de la mano a la falta de experiencia en entrenar modelos para la predicción de problemas con series de tiempo. Al ser el primer proyecto de Machine Learning de todos los integrantes del grupo, se vio en un primer momento una curva de aprendizaje lenta, estando en muchas ocasiones desorientados en cómo encaminar el proyecto a fin de cumplir los objetivos.

Otro de las dificultades o limitaciones fue la capacidad de computo de las máquinas en las que se desatolló estos modelos, realzar tareas como construir curvas de aprendizaje para más de 80.000 instancias requiere un costo computacional grande que se vio reflejado en largos tiempos de ejecución de celdas de código. Por ejemplo a lo largo de todo el proyecto , nos enfrentamos a retos como adquirir conocimientos de paralelismo en programación para la obtención de mejores resultados.

Para un entorno productivo, el modelo sería de ayuda para el área de aprovisionamiento y marketing de la compañía , donde el equipo interno obtenga resultados semanales o mensuales donde los resultados de las predicciones de las ventas le sirvan al equipo para tomar decisiones sobre la compra y aprovisionamiento de cada tienda según sus ventas estimadas , adicionalmente serviría para el estudio de marketing de aquellas tiendas donde se reporten mas ventas y crear estrategias para aquellas tiendas que tienen pocas ventas. En un entorno productivo este modelo de predicción permite a las compañías tener información relevante para toma de decisiones que puedan ayudar a reducir costos operacionales y sobre abastecimiento de artículos en las diferentes tiendas

8. Conclusiones

En este informe se ha abordado el tratamiento de datos de un dataset de ventas de tiendas, aplicando modelos supervisados y no supervisados. El objetivo principal ha sido analizar y predecir las ventas de cada una de las tiendas , así como identificar patrones y comportamientos de ventas.

Al aplicar modelos supervisados se ha logrado predecir las ventas futuras de una tienda , los modelos no supervisados por sí solos no ofrecen una herramienta para la predicción de series de tiempo , mientras que la combinación de modelos no supervisados + supervisados no se encontró una mejora para la predicción e identificación de patrones de ventas. En definitiva, no se observó una mejoría con el uso de combinación de modelos supervisados y no supervisados, en comparación con los resultados obtenidos con un solo modelo.

Es importante destacar también, que con el análisis de las curvas de aprendizajes obtenidas con cada modelo se pudo obtener información sobre cada uno de los modelos de aprendizaje automático, por lo que, fueron útiles para evaluar el sesgo y la varianza de cada uno de ellos.

Por último, se puede concluir que los hiperparametros de todo el dataset no siempre pueden ser los mejores para un conjunto de datos específicos, en este caso, para una sola tienda.

Referencias

[1] Random Forest Classifier sklearn.ensemble.RandomForestClassifier — documentación de scikit-learn 1.2.2.

[2] Support Vector Regression sklearn.svm.SVR — documentación de scikit-learn 1.2.2

[3] SARIMAX https://alkaline-ml.com/pmdarima/tips_and_tricks.html

[4] Pronóstico de series temporales con ARIMA, SARIMA Y SARIMAX Pronóstico de Series Temporales con ARIMA, SARIMA y SARIMAX — por Brendan Artley — Hacia la ciencia de datos (towardsdatascience.com)