Reto Atmira Stock Prediction Nevermore

Mario Campos Mocholí y Pere Marco Garcia

Universitat Politècnica de València

macammoc@inf.upv.es, pemargar@inf.upv.es

1 Introducción

La Organización nos reta a desarrollar un modelo de estimación de unidades vendidas para cada uno de los artículos y para cada uno de los días del año, un modelo de regresión. La métrica a minimizar del sistema es el *rMSE*, definida por:

$$rMSE = \frac{\sqrt{\frac{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2}{n}}}{\overline{y}} \tag{1}$$

Para lograr este fin, se proporcionan dos conjuntos de muestras:

- **Modelar**: 4045022 muestras etiquetadas, utilizado para entrenar y evaluar el modelo por el grupo. En adelante, conjunto modelar.
- **Estimar**: 212841 muestras sin etiquetar, utilizado para evaluar el sistema por la Organización. En adelante, conjunto estimar.

El *script* que genera el fichero respuesta es *Nevermore.py*.

2 Dominio del problema

El problema a tratar consiste en garantizar cubrir la demanda que reclaman los clientes de un e-commerce considerando dos condiciones en el proceso. En primer lugar se debe evitar que se produzca rotura en stock (no hay disponibilidad suficiente de un producto). En segundo lugar, se debe intentar estimar con el menor error posible para minimizar la cantidad de producto a almacenar. En base a ello, se pretende generar un modelo que estime el número de unidades vendidas para cada producto cada uno de los días del año.

2.1 Naturaleza de las variables

Cada muestra está formada por 10 variables, 11 en el caso del conjunto modelar. Cada una representa:

- **Fecha**: Momento en el que se produce el registro. Sigue el formato AAAA-MM-DD.
- **Id**: valor único para identificar cada uno de los diferentes productos.
- **Visitas**: Numero de veces que se ha visitado un producto a lo largo de un día concreto. Puede ser inferior al número de ventas del producto en ese mismo día si se ha comprado desde el carrito o añadido a este mediante las recomendaciones.
- **Categoría uno**: Categoría del producto.
- **Categoría dos**: Subcategoría en la que se agrupan los productos tras la categoría uno.
- **Estado**: situación en la que se encuentra el producto. Esta variable toma 3 posibles valores para el dataset Modelar_UH2021: Rotura: no hay stock físico disponible para servir en nuestros almacenes; Tránsito: no hay stock físico en nuestros almacenes, pero está pendiente de entrega inminente desde proveedor; y No Rotura: hay stock físico disponible en nuestros almacenes.

  Para el dataset Estimar_UH2021 todas las variables aparecen con el estado Tránsito o No Rotura.

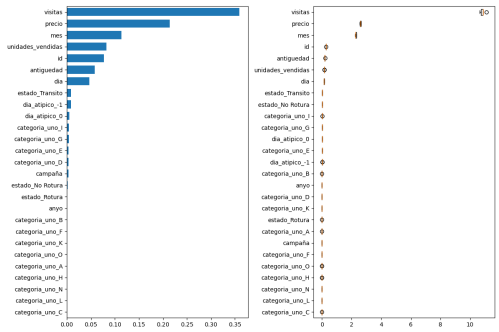

Figura 1. Correlación sobre las ventas

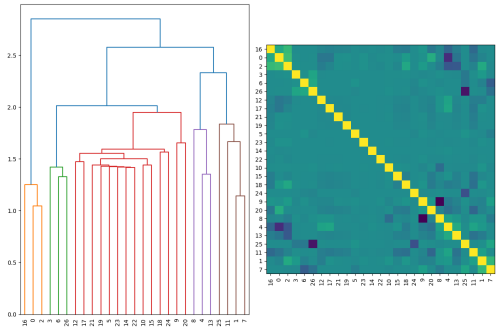

Figura 2. Análisis de correlación

- **Precio**: Precio unitario al que se vende un producto.
- **Día atípico**: toma los valores -1, 0, 1 según nos encontremos en periodo de demanda inferior, igual o superior a lo habitual.
- **Campaña**: Indica si el producto estaba en promoción o no durante alguna de las campañas principales.
- **Antigüedad**: Muestra el número de días desde que el producto se ha añadido al catálogo.
- **Unidades vendidas**: Variable a predecir en el conjunto Estimar_UH202. Indica la cantidad en unidades que se ha vendido de un producto en una fecha determinada.

3 Análisis exploratorio de los datos

En esta sección se realiza el análisis y representación de los datos para entenderlos mejor, observar su composición y tratarlos para mejorar su uso para el modelado.

3.1 Distribución entre ambos conjuntos

Para saber hasta que punto se podía utilizar el dataset proporcionado para modelar en su totalidad para predecir adecuadamente el comportamiento de el conjunto a estimar se han realizado diversas mediciones, en base a las cuales se ha resuelto ciertas consideraciones.

Lo primero que se ha sopesado ha sido qué variables podían tener un impacto relevante en el modelo a entrenar si presentaban algún tipo de desbalanceo entre los conjuntos de datos. Como respuesta se han seleccionado las variable "campaña". "estado" y "día atípico". Estas tres variable, según el impacto que pudieran tener en la estimación del modelo, podrían producir errores inesperados en el conjunto a estimar. Además, en el caso de estado, se observa que en el dataset del que se tiene que obtener una respuesta carece de ningún dato con valor "rotura".

Tabla 1. Distribución de variables

Variable	***Modelar***	***Modelar***
Días de campaña	0.45%	1.45%
Etd. no Rotura	83.34%	98.9%
Etd. Transito	1.31%	1.1%
Día atípico -1	3.48%	3.38%
Día atípico 0	86.30%	65.46%
Día atípico 1	10.23%	31.16%

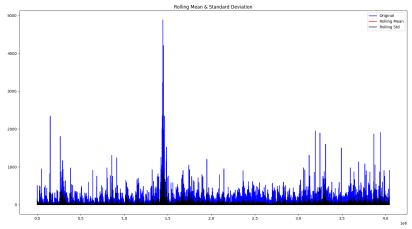

Figura 3. Desviación estándar y media móvil

Los resultados del estudio de la distribución de estos datos se pueden encontrar en la Tabla 1. Observamos que, a pesar de no coincidir, la diferencia de distribución no resulta alarmante en gran medida. En el caso de la variable "estado" se puede observar una gran diferencia entre los dos conjuntos en el Valor "No rotura", pero dicho resultado se debe a la ausencia del valor "Rotura" que hemos mencionado anteriormente. El resultado que más atención requiere de los obtenidos es el aumento de días atípicos "1" en contrapartida del porcentaje del valor "0". Sería de esperar que, si el modelo ha realizado ha aprendido en entornos con la primera distribución, sea propenso a subestimar la demanda que se producirá en el segundo escenario.

Afortunadamente, como se observa en las gráficas que se presentan más adelante, la influencia de estas variables sobre el resultado estimado es aparentemente demasiado bajo para modificarlo de manera relevante.

3.2 Análisis de correlación

Se han aplicado diversas técnicas para visualizar la influencia de las diferentes muestras entre ellas y sobre la variable que se pretende estimar. Para ello se ha realizado dos análisis principales que estudian la correlación entre las variables. En ellos podemos ver claramente que , a pesar de haber creado un gran número de variables, no son las que tienen un mayor impacto, por lo que es de esperar que sus modificaciones no tengan gran influencia en el resultado como se ha comentado anteriormente. También se observa que las variables generadas para la fecha tienen una gran importancia, lo que explica los diferentes resultados obtenidos durante la modificación de dicha variable. A partir de esto hemos concluido que la representación de las variables es adecuada y no se considera indispensable realizar un filtrado de las muestras de entrenamiento.

3.3 Análisis de forecasting

Al tratarse de un problema de predicción de stock, se han realizado análisis tales como ARIMA, descomposición de ondas, Dicker Fuller Test así como la exploración del trend y seasonality. De estos se ha concluido que tratar el dataset como un único producto global, estando realmente compuesto por diferentes productos, da resultados anómalos.

4 Manipulación de las variables

Para solucionar los problemas descritos anteriormente será necesario manipular los datos y adaptarlos para su correcto funcionamiento en los sistemas y obtener mejores métricas. A continuación, se describen todas las transformaciones realizadas.

4.1 Pretratamiento

Lo primero que se debía hacer era transformar las variables cualitativas en valores numéricos que permitieran trabajar con ellos para generar un modelo de regresión. Para ello se ha optado por diferentes procedimiento según el campo a tratar.

- **Fecha**: En el caso de la fecha observamos que incluye siempre la hora fijada en "00:00:00". Esta parte de la variable ha sido eliminado por completo ya que al ser constante no podía dar ningún valor a las estimaciones. A continuación, se ha separado la fecha en tres campos distintos: dia, mes y año. Con esto conseguimos cambiar una variable compleja en tres variables de tipo entero.
- **Categoría_uno**: Para evitar que se relacione una categoría en mayor medida con las categorías vecinas como podría suceder asignando un número a cada valor se ha decidido crear un one-hot encoding para esta variable, sustituyéndola así por 13 variables distintas, una por cada posible valor de la variable categoría_uno.
- **Estado**: En este caso se ha optado por seguir la misma estrategia que en la variable anterior creando 3 nuevas variables para los estados de "Rotura", "No Rotura" y "Transito".

4.2 Depuración

Uno de los factores más importantes que caracteriza este problema de regresión es la gran cantidad de objetos diferentes del que se intenta predecir la demanda. Hemos observado que gran parte de los productos ofrecidos para modelar no se usan en el conjunto a estimar, por lo que esto podría afectar negativamente de forma notable en las predicciones. Para evitar esto hemos optado por realizar una regresión de los objetos de los que realmente se requiere hacer una estimación. En los conjuntos de entrenamiento y test este cambio ha supuesto una notable mejora en la puntuación obtenida con la métrica, motivo por el cual lo hemos aplicado también al modelo final.

4.3 Feature Manipulation

Una vez trasladados los datos aun formato más manejable y apto para su uso en el entrenamiento de un modelo, se ha estudiado el comportamiento de los resultados según cambios en las variables a utilizar. Los campos sobre los que se han efectuado pruebas concluyentes son los siguientes:

- **Fecha**: La primera modificación planteada para la fecha fue la de añadir una variable que representara el día del año en el que se hacía el registro, sustituyendo a las variables de día y mes. Otra alternativa que se ha estudiado ha sido la de incluir una variable adicional que indicara el día de la semana de la fecha en concreto para poder aprovechar algún patrón de compra relacionado con la semana. Los resultados mostraron que la única que mejoraba era aquella en la que, además de las variables "dia", "mes" y "anyo", se añadía la variable "dia_anyo". Sin embargo, al poder producir redundancia y sobre-ajuste, hemos optado por no implementar esta variable.
- **Categoría_dos**: Durante la manipulación de los archivos observamos que algunos productos no disponían de segunda categoría. Por este motivo y la poca relevancia presentada en los análisis en esta variable hemos optado por no considerarla a la hora de realizar los modelos.
- **Antigüedad**: A diferencia del campo anterior, la variable Antigüedad presenta una relevancia notable, por lo que a los datos sin esta variable se las ha asignado un 0 para poder manipular los datos adecuadamente.

5 Implementación del modelo

Tras el anterior análisis de los datos podemos afirmar, tal como indicaba la Organización, que los grandes problemas del reto son los expuestos en la introducción de este documento Por ello, el sistema a desarrollar deberá contar con la flexibilidad y robustez necesaria para clasificar correctamente pese a estos inconvenientes.

Tabla 2. *rMSE* aproximado de las técnicas

Técnica	***rMSE***
SVM	7
Neural network	13
Isotonic regressor	8
Logistic regressor	8
GBoosting	3
Random Forests	2
Kernel Ridge Cluster	7
MLP Regressor	5

5.1 Antecedentes

Se han probado diversas técnicas de regresión de las que cabe destacar las Support Vector Machine, redes neuronales desarrolladas mediante TensorFlow, varios regresores de la libreria Sklearn como el Isotonic regressor, el Logistic regressor y el Kernel Ridge Cluster. También se ha probado el Multi-layer Perceptron regressor. Finalmente también se ha probado los random forest de XGBoosting, pero se obtienen resultados ligeramente peores a los random forests de Sklearn.

En la anterior entrega se probó a entrenar con el dataset entero mediante un random forest, para la entrega final se ha realizado un cambio sunstancial.

5.2 Modelo elegido

De entre todas las técnicas analizadas destacan dos de ellas: *Random Forests* y *Gradient Boosting*. La primera es la elección final. Más concretamente se ha utilizado el regresor basado en *Random Forests* de Sklearn.

Se ha utilizado este ya que es el que aporta mejores resultados al reto y, con una posterior optimización de hiperparámetros, se han conseguido mejores métricas.

Tras realizar diversos análisis de forecasting de los datos, se ha obtenido que el dataset esta compuesto por diversos productos, cada uno con su propio periodo, frecuencia y estacionalidad. Por lo tanto se ha optado por generar un modelo por cada producto y finalmente evaluar el modelo global con la media de todos los modelos. Para ello se ha descompuesto el dataset modelar y estimar por el campo "id", guardando cada modelo en un diccionario cuya clave es el id y valor el modelo. Posteriormente, a la hora de generar el fichero respuesta, se llama a cada modelo para generar una respuesta por cada muestra.

5.3 Evaluación del modelo

Pese que la métrica a optimizar es la propia de la Organización, hemos buscado la minimización del *MSE* ya que la mayoría de librerías de regresión solo utilizan MSE o MAE para entrenar modelos. Esta métrica se define como:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2}{n} \tag{2}$$

Las métricas obtenidas para las diferentes versiones del sistema son las que se muestran en la Tabla 2 y tal como se observa los procesamientos mejoran ligeramente todas las métricas. Se ha realizado *cross validation* para reducir el riesgo de *overfitting*. La métrica obtenida a priori es de aproximadamente 1.5 mejorando la de la primera entrega que era de aproximadamente 2.

6 Conclusiones

Tras los análisis descritos y otros experimentos realizados que no han podido ser descritos debido a la brevedad del documento, finalmente se ha obtenido el fichero respuesta del equipo mediante el script *Nevermore.py* sobre el conjunto *Estimar* tras ser tratado por el archivo *input.py* con parámetro activo de *drop* y usando un modelo entrenado implementando las mejores estrategias encontradas descritas en este documento.