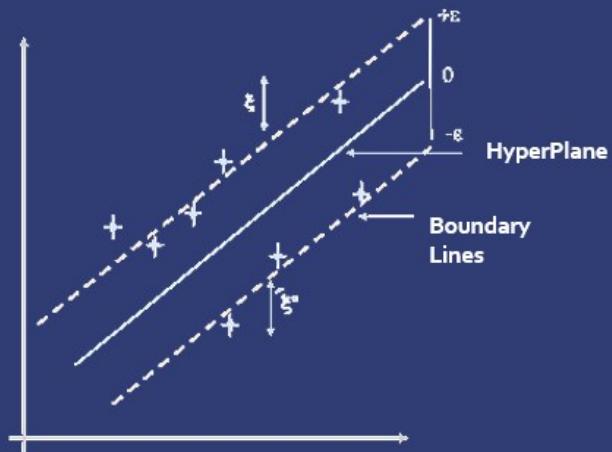


# Proyecto de Máquinas de Vectores

## Soporte para Regresión (SVR)

Máster en Ciberseguridad e Inteligencia de Datos

## Support Vector Regression



Realizado por:  
**Jorge Cabrera Rodríguez**

# Índice general

|   |           |
|---|-----------|
| <b>Índice general</b>                     | <b>I</b>  |
| <b>Índice de figuras</b>                  | <b>1</b>  |
| <b>Índice de cuadros</b>                  | <b>2</b>  |
| <b>1 Introducción</b>                     | <b>3</b>  |
| <b>2 Descripción de conjunto de datos</b> | <b>4</b>  |
| 2.1 Descripción . . . . .                 | 4         |
| 2.2 Limpieza de datos . . . . .           | 5         |
| 2.3 Datos de prueba . . . . .             | 5         |
| <b>3 Implementación de modelos</b>        | <b>6</b>  |
| 3.1 Algoritmos implementados . . . . .    | 6         |
| 3.2 Ajuste de hiperparámetros . . . . .   | 6         |
| 3.3 Métricas de rendimiento . . . . .     | 7         |
| <b>4 Escenarios de regresión</b>          | <b>9</b>  |
| 4.1 Escenario 01 . . . . .                | 9         |
| 4.2 Escenario 02 . . . . .                | 11        |
| 4.3 Escenario 03 . . . . .                | 12        |
| 4.4 Escenario 04 . . . . .                | 13        |
| 4.5 Escenario 05 . . . . .                | 15        |
| 4.6 Escenario 06 . . . . .                | 16        |
| 4.7 Escenario 07 . . . . .                | 17        |
| 4.8 Escenario 08 . . . . .                | 19        |
| 4.9 Escenario 09 . . . . .                | 20        |
| 4.10 Escenario 10 . . . . .               | 21        |
| <b>5 Conclusiones</b>                     | <b>23</b> |

# Índice de figuras

|   |    |
|---|----|
| 4.1 Comparativa de regresión de escenario 01 . . . . .  | 10 |
| 4.2 Comparativa de regresión de escenario 02 . . . . .  | 11 |
| 4.3 Comparativa de regresión de escenario 03 . . . . .  | 13 |
| 4.4 Comparativa de regresión de escenario 04 . . . . .  | 14 |
| 4.5 Comparativa de regresión de escenario 05 . . . . .  | 15 |
| 4.6 Comparativa de regresión de escenario 06 . . . . .  | 17 |
| 4.7 Comparativa de regresión de escenario 07 . . . . .  | 18 |
| 4.8 Comparativa de regresión de escenario 08 . . . . .  | 20 |
| 4.9 Comparativa de regresión de escenario 09 . . . . .  | 21 |
| 4.10 Comparativa de regresión de escenario 10 . . . . . | 22 |

# Índice de cuadros

|      |   |    |
|------|---|----|
| 2.1  | Listado de campos del conjunto de datos . . . . . | 4  |
| 3.1  | Tabla de hiperparámetros por algoritmo . . . . .  | 7  |
| 4.1  | Resultados de escenario 01 . . . . .              | 10 |
| 4.2  | Resultados de escenario 02 . . . . .              | 11 |
| 4.3  | Resultados de escenario 03 . . . . .              | 13 |
| 4.4  | Resultados de escenario 04 . . . . .              | 14 |
| 4.5  | Resultados de escenario 05 . . . . .              | 15 |
| 4.6  | Resultados de escenario 06 . . . . .              | 17 |
| 4.7  | Resultados de escenario 07 . . . . .              | 18 |
| 4.8  | Resultados de escenario 08 . . . . .              | 19 |
| 4.9  | Resultados de escenario 09 . . . . .              | 21 |
| 4.10 | Resultados de escenario 10 . . . . .              | 22 |

# 1 Introducción

En el mercado del transporte de mercancías, la estimación de tiempos de envío de grandes mercancías es un aspecto crucial en la logística de una empresa: Determinar de manera precisa estos tiempos permiten a las compañías distribuidoras optimizar las rutas de mercancía para poder realizar un mayor número de envíos en el menor tiempo posible. Sin embargo, esto depende enteramente de que el estimado de los tiempos de envío sea certero.

El objetivo de este proyecto es diseñar, entrenar y medir el rendimiento de diferentes modelos de regresión que sean capaces de estimar (en la medida de lo posible) los tiempos de envío (*shipping time*) de diferentes transportes de mercancías.

Para ello se han planteado diferentes algoritmos de regresión con diferentes configuraciones de parámetros y operaciones de preprocesados (los denominados escenarios de aquí en adelante).

Con esta investigación se busca buscar qué algoritmo se adecúa más al problema de estimación de tiempos de envío.

## 2 Descripción de conjunto de datos

### 2.1. Descripción

El conjunto de datos de envíos utilizado para entrenar y probar los modelos de regresión ha sido extraído del repositorio público *Shipping Optimization Challenge*, alojado en el sitio web Kaggle.

Este conjunto se encuentra almacenado en el fichero '`train_2.pr.csv`' y cuenta con los siguientes campos:

| Campos                           | Tipo             | Descripción   |
|----------------------------------|------------------|---|
| <code>shipment_id</code>         | Numerico         | ID del envío.   |
| <code>send_timestamp</code>      | <i>Timestamp</i> | Fecha en la que el pedido fue enviado al país de destino (en la zona horaria del país de origen). |
| <code>pick_up_point</code>       | Categórico       | Abreviatura del punto de recogida.  |
| <code>drop_off_point</code>      | Categórico       | Abreviatura del punto de entrega.   |
| <code>source_country</code>      | Categórico       | País desde donde se deben enviar los bienes.  |
| <code>destination_country</code> | Categórico       | País al que se deben enviar los bienes.   |
| <code>freight_cost</code>        | Numerico         | Costo de transporte por kg.   |
| <code>gross_weight</code>        | Numerico         | Peso bruto en kg que se debe enviar.  |
| <code>shipment_charges</code>    | Numerico         | Costo fijo por envío.   |
| <code>shipment_mode</code>       | Categórico       | Método de envío (por ejemplo, aire, mar).   |
| <code>shipping_company</code>    | Categórico       | Empresa de envío candidata.   |
| <code>selected</code>            | Categórico       | Si se seleccionó o no la empresa en <code>shipping_company</code> .                               |
| <code>shipping_time</code>       | Numerico         | La cantidad de tiempo que tardan los bienes en llegar a su destino.                               |

Tabla 2.1: Listado de campos del conjunto de datos

Cuenta con 5114 registros, y tras realizar un análisis previo se ha determinado que ninguno de estos registros cuenta con campos nulos. En este mismo análisis se ha observado que los campos `pick_up_point` y `selected` toman un único valor para todo el conjunto de datos, por lo que se puede determinar que son **campos redundantes**. Además, el campo `shipment_id` muestra valores irrelevantes al ser un identificador generado aleatoriamente, por lo que también es un **campo redundante**.

Además, el campo `drop_off_point` toma para cada registro valores equivalentes al campo `destination_country`. Es decir, siempre que en un registro el campo `drop_off_point` toma un valor  $X$ , el campo `destination_country` toma siempre el valor  $Y$ . Ya que este campo aporta la misma información exacta que `destination_country`, es considerado como **campo redundante**.

Al leer la fuente de datos desde el entorno informático, se ha generado un campo residual `Unnamed:0` que representa la posición de la fila en el fichero. Este campo también es considerado **campo redundante**.

## 2.2. Limpieza de datos

Antes de comenzar el desarrollo del proyecto se han realizado las siguientes operaciones de limpieza de datos:

- Se ha eliminado el campo `pick_up_point` por considerarse redundante.
- Se ha eliminado el campo `drop_off_point` por considerarse redundante.
- Se ha eliminado el campo `selected` por considerarse redundante.
- Se ha eliminado el campo residual `Unnamed:0` por considerarse redundante.

## 2.3. Datos de prueba

Adicionalmente al conjunto de datos de entrenamiento, el repositorio de origen ofrece un fichero denominado `test_2.csv` con el que probar los modelos generados. Este conjunto de datos no cuenta con los tiempos de envío, por lo que no se podrá saber su rendimiento una vez predichos sus tiempos.

## 3 Implementación de modelos

### 3.1. Algoritmos implementados

Para este proyecto se han implementado los siguientes algoritmos de modelos de aprendizaje automático (*Machine Learning*):

- *Support Vector Regressor (SVR)*: Un algoritmo de aprendizaje automático utilizado para problemas de regresión, que busca encontrar la función que mejor se ajuste a un conjunto de datos mediante la construcción de un hiperplano en un espacio de alta dimensión.
- *Decision Tree Regressor*: Un algoritmo de aprendizaje automático que construye un árbol de decisiones a partir de los datos de entrenamiento y lo utiliza para predecir el valor objetivo de nuevas instancias.
- *Random Forest Regressor*: Un método de aprendizaje automático que opera construyendo múltiples árboles de decisión durante el entrenamiento y produciendo la predicción que es la media de las predicciones de los árboles individuales.
- *AdaBoost Regressor*: Un algoritmo de aprendizaje automático que ajusta secuencialmente modelos débiles a los datos, enfocándose en las instancias mal clasificadas en cada iteración para mejorar el rendimiento del modelo.
- *KNeighbors Regressor*: Un método de aprendizaje automático que busca los k vecinos más cercanos a un punto de consulta en el espacio de características y utiliza sus etiquetas para realizar una predicción de regresión.

Estos modelos se entrenarán y probarán para diferentes combinaciones de preprocesados (escenarios) con el objetivo de comprobar en qué situación los modelos ofrecen mejores resultados, así como el modelo específico con mejores métricas de rendimiento.

### 3.2. Ajuste de hiperparámetros

Con el objetivo de optimizar el rendimiento de los algoritmos de regresión, se ha utilizado un método de ajustado de hiperparámetros denominado *Grid Search*. Este método entrena y prueba estimadores utilizando un rango de hiperparámetros definido por el usuario. Al ejecutar *Grid Search* se obtienen los hiperparámetros que maximizan la métrica de rendimiento a utilizar.

Los rangos de hiperparámetros especificados para cada algoritmo son los siguientes:

| Algoritmo               | Hiperparámetro   | Rango                           |
|-------------------------|------------------|---------------------------------|
| SVR                     | kernel           | poly, rbf, sigmoid              |
|                         | C                | 0.1, 1, 10                      |
|                         | epsilon          | 0.1, 0.2, 0.5                   |
| Decision Tree Regressor | max_depth        | None, 5, 10, 20                 |
|                         | min_sample_split | 2, 5, 10                        |
|                         | min_sample_leaf  | 1, 2, 4                         |
| AdaBoost Regressor      | n_estimators     | 50, 100, 200                    |
|                         | learning_rate    | 0.01, 0.1, 1.0                  |
|                         | loss             | linear, square, exponential     |
| KNeighbors Regressor    | n_estimators     | 5, 10, 20                       |
|                         | weights          | uniform, distance               |
|                         | algorithm        | auto, ball_tree, kd_tree, brute |
|                         | leaf_size        | 30, 50, 100                     |
|                         | p                | 1, 2                            |
| Random Forest Regressor |                  |                                 |

Tabla 3.1: Tabla de hiperparámetros por algoritmo

El algoritmo *Random Forest Regressor* es el único algoritmo listado sin hiperparámetros optimizados, pues durante el desarrollo del proyecto se intentó averiguar sus hiperparámetros ideales mediante *Grid Search*, pero los tiempos de entrenamiento del modelo eran demasiado elevados, y por tanto entorpecían el estudio.

Se decidió utilizar el algoritmo sin optimización de hiperparámetros.

### 3.3. Métricas de rendimiento

La métrica de rendimiento utilizada para medir el rendimiento de los algoritmos es el coeficiente de determinación, comúnmente conocido como  $R^2$ . Esta métrica proporciona una medida de qué tan bien se ajustan los valores predichos por el modelo a los valores observados en los datos. En otras palabras,  $R^2$  indica la proporción de la varianza en la variable dependiente que es predecible a partir de las variables independientes en el modelo. Un valor de  $R^2$  cercano a 1 indica un ajuste perfecto, mientras que un valor cercano a 0 sugiere que el modelo no explica la variabilidad de los datos.

El coeficiente de determinación  $R^2$  se calcula mediante la siguiente fórmula matemática:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Donde  $SS_{res}$  es la suma de los cuadrados de los residuos, es decir, la diferencia entre los valores observados y los valores predichos por el modelo, y  $SS_{tot}$  es la suma total de los cuadrados, es decir, la diferencia entre los valores observados y la media de los valores observados. Un valor de  $R^2$  más cercano a 1 indica un mejor ajuste del modelo a los datos.

## 4 Escenarios de regresión

Se define **escenario** como una secuencia de preprocesados aplicados al conjunto de datos, con el objetivo de tomar estos datos corregidos y alimentar los diferentes modelos de aprendizaje automático.

En el desarrollo de este proyecto se han implementado múltiples **escenarios** diferentes, con el objetivo de observar el impacto de diferentes preprocesados en los resultados finales de regresión.

La creación de escenarios se ha realizado de forma progresiva, por lo que la secuencia de preprocesados de determinados escenarios pueden haberse escogido porque en escenarios anteriores haya habido mejoría al aplicar alguno de los preprocesados de la secuencia.

Para cada uno de estos escenarios se han implementado todos los algoritmos de regresión comentados anteriormente, utilizando el método *Grid Search* para la optimización de hiperparámetros. Para cada algoritmo se han medido también sus tiempos de entrenamiento y predicción, para determinar su eficiencia computacional.

Adicionalmente, en cada escenario se ha generado para cada algoritmo un gráfico representativo de los valores estimados por el regresor frente a los valores reales de las muestras.

### 4.1. Escenario 01

#### 4.1.1. Descripción

El primer escenario generado realizó la secuencia mínima de preprocesados para alimentar un modelo de regresión. El único requerimiento de este tipo de modelos es que las características con las que se entrena sean todas enteramente numéricas, por lo que hizo falta convertir las características categóricas en primer lugar. Para ello, se aplicaron los siguientes preprocesados:

- Se aplicó un proceso de etiquetado (*labeling*) para cada campo. En este etiquetado, cada posible valor categórico ha sido mapeado como un valor numérico, para luego ser sustituido en el conjunto de datos.
- Existe un campo del conjunto de datos expresado en forma de fecha (*timestamp*) que se llama `send_timestamp`. Esta variable temporal ha sido convertida a variable numérica calculando el número de segundos transcurridos desde la **Fecha Epoch** de los sistemas **Unix**.

### 4.1.2. Resultados obtenidos

Las métricas de rendimiento y tiempos de entrenamiento/predicción del **escenario 01** han sido los siguientes:

| Model                 | $R^2$ score | Training time | Prediction time |
|-----------------------|-------------|---------------|-----------------|
| SVR                   | -0.316321   | 0.646481      | 0.170164        |
| KNeighborsRegressor   | 0.020313    | 0.012182      | 0.007037        |
| RandomForestRegressor | 0.513437    | 4.217883      | 0.032037        |
| DecisionTreeRegressor | 0.547034    | 0.014498      | 0.001000        |
| AdaBoostRegressor     | 0.583008    | 0.640691      | 0.010998        |

Tabla 4.1: Resultados de escenario 01

El gráfico de predicciones resultante es el siguiente:

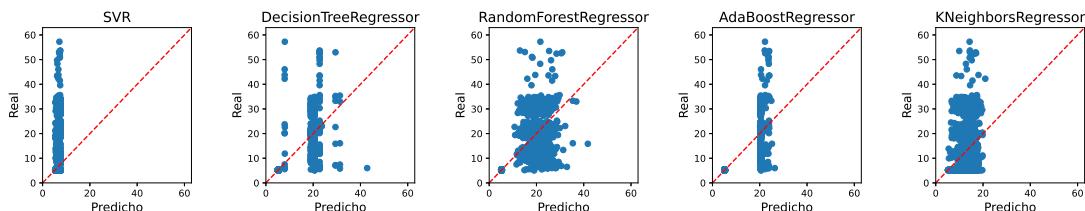


Figura 4.1: Comparativa de regresión de escenario 01

### 4.1.3. Conclusión del escenario

En este primer escenario se han obtenido resultados promedio de coeficiente  $R^2$ , obteniendo la mitad de los algoritmos implementados un valor de  $R^2$  superior a 0.5.

Analizando los tiempos de entrenamiento/predicción se observa que el algoritmo *Random Forest Regressor* tarda varias magnitudes de tiempo más que el resto de algoritmos en entrenarse: Esta diferencia tan significativa podría implicar que el algoritmo no sea viable para implementar en proyectos con conjuntos de datos mucho más masivos.

Destaca el pésimo resultado de los algoritmos *Support Vector Regressor* y *KNeighbors Regressor*, donde el primero obtiene valores negativos y el segundo valores cercanos al 0. Si este comportamiento se mantiene en posteriores escenarios, se podría determinar que estos algoritmos no son viables para el problema de estimación de tiempos.

Se propone tomar los preprocesados de este escenario como base para los próximos, con el añadido de diferentes algoritmos de Escalado de datos para los campos numéricos.

## 4.2. Escenario 02

### 4.2.1. Descripción

El segundo escenario parte de los resultados del **escenario 01**. En este escenario se aplicarán los siguientes preprocesados:

- Etiquetado de campos categóricos (heredado de **escenario 01**).
- Conversión de campo `send_timestamp` a segundos (heredado de **escenario 01**).
- Escalado de campos numéricos mediante `StandardScaler`.

### 4.2.2. Resultados obtenidos

Las métricas de rendimiento y tiempos de entrenamiento/predicción del **escenario 02** han sido los siguientes:

| Model                 | $R^2$ score | Training time | Prediction time |
|-----------------------|-------------|---------------|-----------------|
| SVR                   | -0.038365   | 0.414286      | 0.104517        |
| KNeighborsRegressor   | -0.011375   | 0.007000      | 0.005988        |
| RandomForestRegressor | 0.467232    | 4.372633      | 0.033549        |
| DecisionTreeRegressor | 0.517259    | 0.012000      | 0.001000        |
| AdaBoostRegressor     | 0.555980    | 0.831870      | 0.022994        |

Tabla 4.2: Resultados de escenario 02

El gráfico de predicciones resultante es el siguiente:

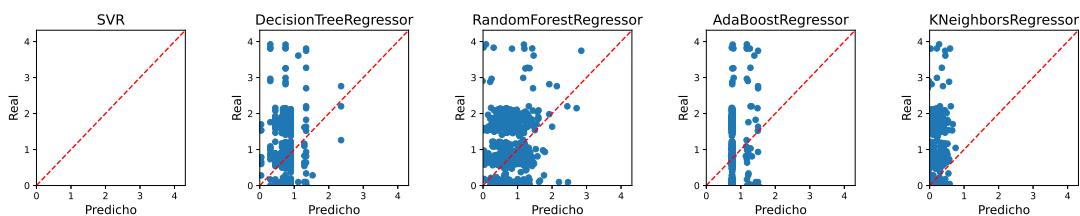


Figura 4.2: Comparativa de regresión de escenario 02

### 4.2.3. Conclusión del escenario

Los resultados del **escenario 02** han sido parecidos a los de su escenario predecesor. Se observa que los tres algoritmos con mejor  $R^2$  han empeorado su funcionamiento frente al **escenario 01**:

- *AdaBoost Regressor* ha pasado de un 0.58 a 0.55.
- *Decision Tree Regressor* ha pasado de un 0.54 a 0.51.
- *Random Forest Regressor* ha pasado de un 0.58 a 0.55.

Sin embargo, el *Support Vector Regressor* obtiene una mejoría notable al alcanzar un  $R^2$  cercano a 0, frente al  $R^2$  negativo del escenario anterior.

Los tres algoritmos comentados anteriormente se mantienen como los mejores regresores hasta el momento.

## 4.3. Escenario 03

### 4.3.1. Descripción

El tercer escenario parte de los resultados del **escenario 02**. En este escenario se aplicarán los siguientes preprocesados:

- Etiquetado de campos categóricos (heredado de **escenario 01**).
- Conversión de campo `send_timestamp` a segundos (heredado de **escenario 01**).
- Escalado de campos numéricos mediante `MinMaxScaler`.

### 4.3.2. Resultados obtenidos

Las métricas de rendimiento y tiempos de entrenamiento/predicción del **escenario 03** han sido los siguientes:

| Model                 | $R^2$ score | Training time | Prediction time |
|-----------------------|-------------|---------------|-----------------|
| SVR                   | -0.047125   | 0.464571      | 0.467453        |
| KNeighborsRegressor   | -0.023687   | 0.007001      | 0.005998        |
| RandomForestRegressor | 0.524206    | 4.697995      | 0.031038        |
| DecisionTreeRegressor | 0.583340    | 0.015960      | 0.001002        |
| AdaBoostRegressor     | 0.596477    | 0.733636      | 0.023594        |

Tabla 4.3: Resultados de escenario 03

El gráfico de predicciones resultante es el siguiente:

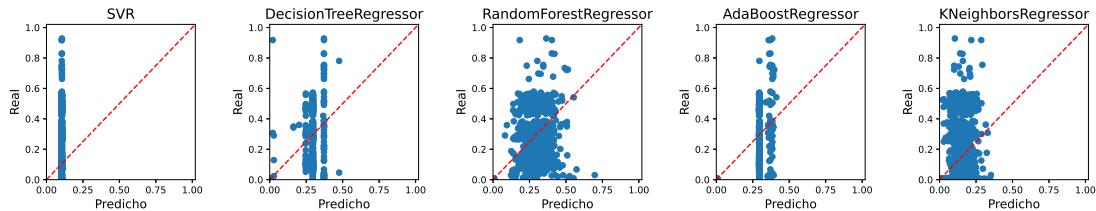


Figura 4.3: Comparativa de regresión de escenario 03

### 4.3.3. Conclusión del escenario

En este escenario se obtiene una pequeña mejoría en los resultados respecto al **escenario 01** (incremento de 0,02 de  $R^2$  en todos los modelos), manteniendo unos tiempos de entrenamiento y predicción constantes.

Los algoritmos *Support Vector Regressor* y *KNeighbors Regressor* siguen manteniéndose como inviables hasta este escenario.

## 4.4. Escenario 04

### 4.4.1. Descripción

El cuarto escenario parte de los resultados del **escenario 03**. En este escenario se aplicarán los siguientes preprocesados:

- Etiquetado de campos categóricos (heredado de **escenario 01**).
- Conversión de campo `send_timestamp` a segundos (heredado de **escenario 01**).

- Escalado de campos numéricos mediante `RobustScaler`.

#### 4.4.2. Resultados obtenidos

Las métricas de rendimiento y tiempos de entrenamiento/predicción del **escenario 04** han sido los siguientes:

| Model                 | $R^2$ score | Training time | Prediction time |
|-----------------------|-------------|---------------|-----------------|
| SVR                   | 0.001261    | 0.209716      | 0.062515        |
| KNeighborsRegressor   | 0.021716    | 0.003999      | 0.005999        |
| RandomForestRegressor | 0.500073    | 3.633811      | 0.027987        |
| DecisionTreeRegressor | 0.512756    | 0.010000      | 0.001010        |
| AdaBoostRegressor     | 0.571256    | 0.299844      | 0.011000        |

Tabla 4.4: Resultados de escenario 04

El gráfico de predicciones resultante es el siguiente:

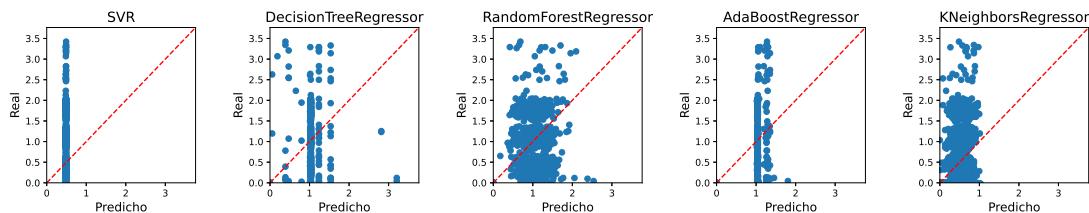


Figura 4.4: Comparativa de regresión de escenario 04

#### 4.4.3. Conclusión del escenario

Este escenario mantiene unos resultados similares a sus predecesores, con variaciones de  $R^2$  marginales. El valor más destacable es el  $R^2$  del *Support Vector Regressor*, pues se trata del primer escenario en el que toma valor positivo.

## 4.5. Escenario 05

### 4.5.1. Descripción

El quinto escenario parte de los resultados del escenario 04. En este escenario se aplicarán los siguientes preprocesados:

- Etiquetado de campos categóricos (heredado de **escenario 01**).
- Conversión de campo `send_timestamp` a segundos (heredado de **escenario 01**).
- Escalado de campos numéricos mediante `QuantileTransformer`.

### 4.5.2. Resultados obtenidos

Las métricas de rendimiento y tiempos de entrenamiento/predicción del **escenario 05** han sido los siguientes:

| Model                 | $R^2$ score | Training time | Prediction time |
|-----------------------|-------------|---------------|-----------------|
| KNeighborsRegressor   | 0.006006    | 0.016998      | 0.006033        |
| SVR                   | 0.026127    | 0.281775      | 0.105560        |
| RandomForestRegressor | 0.714554    | 5.024667      | 0.035048        |
| DecisionTreeRegressor | 0.734811    | 0.011999      | 0.000998        |
| AdaBoostRegressor     | 0.749276    | 1.595908      | 0.051583        |

Tabla 4.5: Resultados de escenario 05

El gráfico de predicciones resultante es el siguiente:

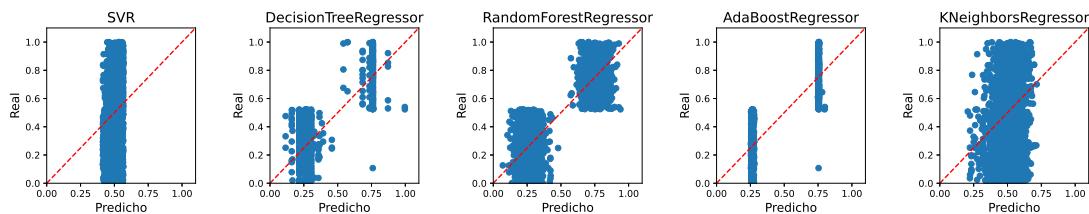


Figura 4.5: Comparativa de regresión de escenario 05

### 4.5.3. Conclusión del escenario

Este es el primer escenario en el que se observa una mejoría notable en el valor  $R^2$ , aumentando los tres mejores algoritmos de regresión hasta el momento en 0.2 unidades. Este escenario coloca al algoritmo *AdaBoost Regressor* como el mejor algoritmo de regresión hasta el momento.

La mejoría general de  $R^2$  también implica aumentos en los tiempos de entrenamiento: El algoritmo *AdaBoost* ha alcanzado 1.59 segundos de entrenamiento, un aumento debido seguramente a la naturaleza de los datos. Este incremento de tiempos favorece al segundo mejor algoritmo de regresión, el *Decision Tree Regressor*, pues alcanza resultados similares con tiempos de entrenamiento y validación abismalmente inferiores.

Se observa como los algoritmos *KNeighbors Regressor* y *Support Vector Regressor* siguen manifestando resultados pésimos de  $R^2$ .

## 4.6. Escenario 06

### 4.6.1. Descripción

Habiendo observado la mejoría del **escenario 05**, en el **escenario 06** se ha decidido modificar los preprocesados aplicados con la esperanza de obtener resultados variados. Los preprocesados son los siguientes:

- Etiquetado de campos categóricicos (heredado de **escenario 01**).
- Segmentación de campo `send_timestamp` en múltiples campos temporales que dependan de la fecha original, como puedan ser `year`, `month`, `dayofweek` o `dayofyear`.

### 4.6.2. Resultados obtenidos

Las métricas de rendimiento y tiempos de entrenamiento/predicción del **escenario 06** han sido los siguientes:

| Model                 | $R^2$ score | Training time | Prediction time |
|-----------------------|-------------|---------------|-----------------|
| SVR                   | -0.248382   | 0.891390      | 0.585317        |
| KNeighborsRegressor   | 0.067268    | 0.011016      | 0.017999        |
| RandomForestRegressor | 0.573800    | 7.269513      | 0.035532        |
| DecisionTreeRegressor | 0.582850    | 0.014557      | 0.001998        |
| AdaBoostRegressor     | 0.602407    | 0.743121      | 0.026997        |

Tabla 4.6: Resultados de escenario 06

El gráfico de predicciones resultante es el siguiente:

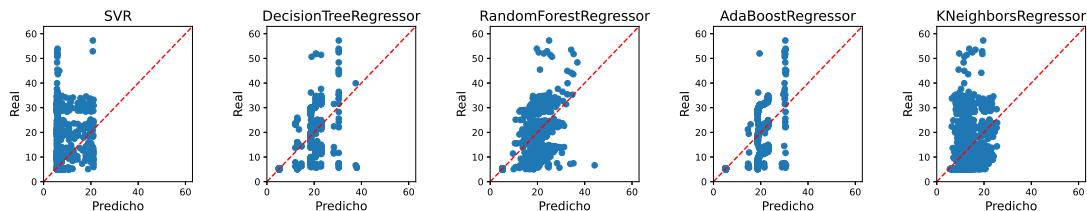


Figura 4.6: Comparativa de regresión de escenario 06

### 4.6.3. Conclusión del escenario

En este escenario se vuelven a obtener resultados similares a los de los primeros escenarios, con una ligera mejoría de coeficiente  $R^2$  en los tres mejores algoritmos (aumento de 0.03 respecto al escenario 2).

El aumento de variables del conjunto parece haber provocado que el algoritmo *Random Forest Regressor* aumente su tiempo de entrenamiento aproximadamente 2 segundos. Este aumento de tiempo podría perjudicar posibles implementaciones del algoritmo en escenarios reales con conjuntos de datos más grandes.

## 4.7. Escenario 07

### 4.7.1. Descripción

El séptimo escenario realiza las siguientes operaciones de preprocesado:

- Etiquetado de campos categóricos (heredado de **escenario 01**).

- Segmentación de campo `send_timestamp` en múltiples campos (heredado de **escenario 06**).
- Escalado de valores numéricos mediante `QuantileTransformer` (heredado de **escenario 05**).

### 4.7.2. Resultados obtenidos

Las métricas de rendimiento y tiempos de entrenamiento/predicción del **escenario 07** han sido los siguientes:

| Model                 | $R^2$ score | Training time | Prediction time |
|-----------------------|-------------|---------------|-----------------|
| SVR                   | 0.135108    | 0.538066      | 0.158312        |
| KNeighborsRegressor   | 0.264823    | 0.012081      | 0.028323        |
| RandomForestRegressor | 0.739366    | 6.125848      | 0.037087        |
| DecisionTreeRegressor | 0.744398    | 0.020655      | 0.002002        |
| AdaBoostRegressor     | 0.750044    | 1.298414      | 0.025342        |

Tabla 4.7: Resultados de escenario 07

El gráfico de predicciones resultante es el siguiente:

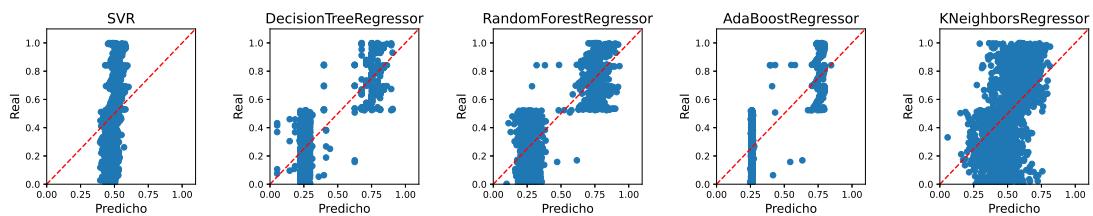


Figura 4.7: Comparativa de regresión de escenario 07

### 4.7.3. Conclusión del escenario

Al aplicar este escenario se vuelven a observar valores de  $R^2$  superiores a 0.7 en los tres mejores algoritmos:

- *AdaBoost Regressor*
- *Decision Tree Regressor*

- *Random Forest Regressor*

Este escenario ha producido hasta el momento los  $R^2$  más elevados para el *Support Vector Regressor* y el *KNeighbors Regressor*, pese a que ninguno alcanza el 0.3.

Los tiempos de entrenamiento y predicción se mantienen constantes.

## 4.8. Escenario 08

### 4.8.1. Descripción

El octavo escenario aplica los mismos preprocesados que el **escenario 07**:

- Etiquetado de campos categóricos (heredado de **escenario 01**).
- Segmentación de campo `send_timestamp` en múltiples campos (heredado de **escenario 06**).
- Escalado de valores numéricos mediante `QuantileTransformer` (heredado de **escenario 05**).

La mayor diferencia radica en que en este escenario eliminamos el campo de minutos, por considerarse potencialmente irrelevante al tener una granularidad demasiado pequeña.

### 4.8.2. Resultados obtenidos

Las métricas de rendimiento y tiempos de entrenamiento/predicción del **escenario 08** han sido los siguientes:

| Model                 | $R^2$ score | Training time | Prediction time |
|-----------------------|-------------|---------------|-----------------|
| SVR                   | 0.134701    | 0.491569      | 0.142532        |
| KNeighborsRegressor   | 0.245336    | 0.002001      | 0.049006        |
| RandomForestRegressor | 0.735003    | 5.769983      | 0.034516        |
| AdaBoostRegressor     | 0.747912    | 0.392227      | 0.014999        |
| DecisionTreeRegressor | 0.750602    | 0.013138      | 0.000999        |

Tabla 4.8: Resultados de escenario 08

El gráfico de predicciones resultante es el siguiente:

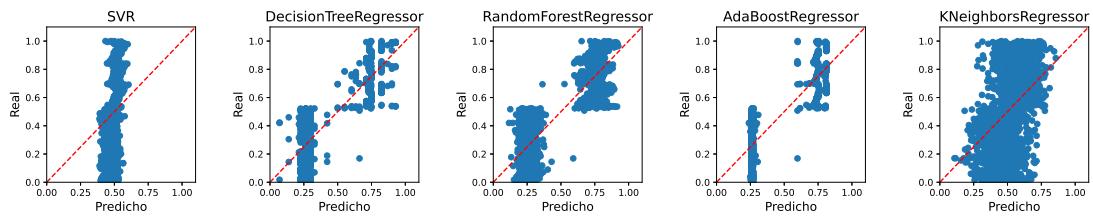


Figura 4.8: Comparativa de regresión de escenario 08

### 4.8.3. Conclusión del escenario

Los resultados del escenario son muy similares a los extraídos en el **escenario 07**, por lo que no se puede extraer ninguna conclusión positiva de haber eliminado el campo en cuestión. Se puede determinar que la eliminación del campo relacionado con los minutos de un envío no parece afectar al modelo.

## 4.9. Escenario 09

### 4.9.1. Descripción

El noveno escenario aplica los siguientes preprocesados:

- Etiquetado de campos categóricos (heredado de **escenario 01**).
- Segmentación de campo `send_timestamp` en múltiples campos (heredado de **escenario 06**).
- Escalado de valores numéricos mediante `QuantileTransformer` (heredado de **escenario 05**).
- Extracción de campos menos relevantes mediante *Recursive Feature Elimination (RFE)*.

### 4.9.2. Resultados obtenidos

Las métricas de rendimiento y tiempos de entrenamiento/predicción del **escenario 09** han sido los siguientes:

| Model                 | $R^2$ score | Training time | Prediction time |
|-----------------------|-------------|---------------|-----------------|
| KNeighborsRegressor   | 0.200449    | 0.005006      | 0.010034        |
| SVR                   | 0.591240    | 0.463433      | 0.376839        |
| RandomForestRegressor | 0.715988    | 3.947413      | 0.038667        |
| DecisionTreeRegressor | 0.736614    | 0.010008      | 0.001128        |
| AdaBoostRegressor     | 0.745412    | 1.506342      | 0.054661        |

Tabla 4.9: Resultados de escenario 09

El gráfico de predicciones resultante es el siguiente:

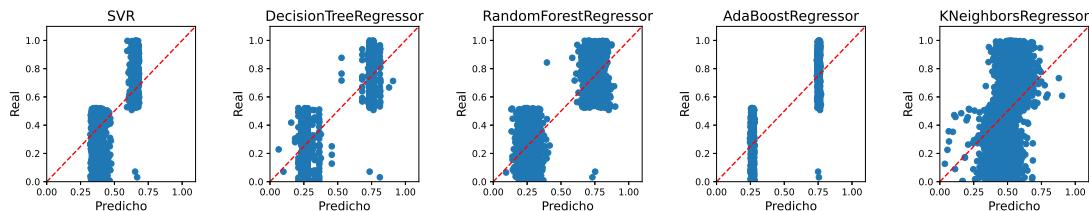


Figura 4.9: Comparativa de regresión de escenario 09

### 4.9.3. Conclusión del escenario

Realizando la extracción de características se observa una mejoría significativa en los resultados del *Support Vector Regressor*: Se trata del primer escenario en el que alcanza un valor de  $R^2$  de 0.4.

El resto de algoritmos mantienen resultados similares frente a escenarios anteriores.

## 4.10. Escenario 10

### 4.10.1. Descripción

Como escenario final, se han implementado los siguientes preprocesados para observar su rendimiento:

- *OneHot Encoding* de las variables categóricas.
- Segmentación de campo `send_timestamp` en múltiples campos (heredado de escenario 06).

- Escalado de valores numéricos mediante `QuantileTransformer` (heredado de escenario 05).

#### 4.10.2. Resultados obtenidos

Las métricas de rendimiento y tiempos de entrenamiento/predicción del **escenario 10** han sido los siguientes:

| Model                 | $R^2$ score | Training time | Prediction time |
|-----------------------|-------------|---------------|-----------------|
| SVR                   | 0.110332    | 0.444454      | 0.139508        |
| KNeighborsRegressor   | 0.371332    | 0.001999      | 0.041007        |
| RandomForestRegressor | 0.726136    | 5.799738      | 0.030998        |
| AdaBoostRegressor     | 0.741017    | 0.789322      | 0.024515        |
| DecisionTreeRegressor | 0.747163    | 0.012001      | 0.001001        |

Tabla 4.10: Resultados de escenario 10

El gráfico de predicciones resultante es el siguiente:

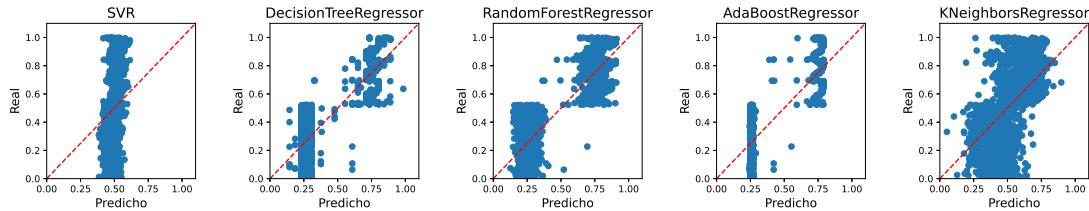


Figura 4.10: Comparativa de regresión de escenario 10

#### 4.10.3. Conclusión del escenario

Este escenario es el que mejores resultados globales obtiene, pues ninguno de los algoritmos implementados tiene un  $R^2$  inferior a 0.35. Además de eso, el *Decision Tree Regressor* se ha colocado por primera vez como el algoritmo con mejores resultados de  $R^2$ .

## 5 Conclusiones

Tras un análisis exhaustivo y una evaluación detallada de varios modelos de regresión para el conjunto de datos en consideración, se han extraído conclusiones significativas que pueden orientar las decisiones futuras en el proceso de modelado.

En primer lugar, se observó que *Adaboost Regressor* destacó como el modelo más prometedor, demostrando los mejores resultados generales en términos de rendimiento predictivo. Sus métricas de evaluación superaron consistentemente a las de los otros modelos considerados, lo que sugiere que podría ser una opción sólida para la tarea de regresión.

A pesar de que *Adaboost Regressor* lideró en términos de rendimiento, el análisis también reveló la viabilidad de *Decision Tree Regressor*. Aunque este modelo no logró alcanzar los niveles de rendimiento de *Adaboost Regressor*, mostró un desempeño respetable y requirió menos tiempo tanto para el entrenamiento como para la predicción. Esta eficiencia computacional podría considerarse ventajosa en ciertos contextos de implementación.

Sin embargo, no todos los modelos evaluados resultaron viables para la tarea en cuestión. Tanto *SVR* como *KNeighbors Regressor* exhibieron métricas de evaluación que sugieren su inviabilidad en el conjunto de datos específico. Estos resultados resaltan la importancia de una selección cuidadosa del modelo en función de las características y la naturaleza de los datos.

Además, durante el análisis de preprocesamiento de datos, se identificaron estrategias que mejoraron el rendimiento de los modelos. En particular, se observó que el normalizado por Cuantiles (*Quantile Transformer*) mostró ser una técnica beneficiosa que puede considerarse en futuros trabajos de modelado para mejorar la capacidad predictiva de los modelos.

Por último, se encontró que el *Onehot Encoding*, una técnica comúnmente utilizada para codificar variables categóricas, desempeñó un papel crucial en el rendimiento de los modelos. Esta observación subraya la importancia de abordar adecuadamente las variables categóricas en el proceso de modelado para obtener resultados más precisos y confiables.

En resumen, estas conclusiones enfatizan la importancia de una selección cuidadosa del modelo y las técnicas de preprocesamiento de datos en el proceso de modelado de regresión. Los hallazgos obtenidos proporcionan una guía valiosa para futuros trabajos de investigación y desarrollo en este campo, con el objetivo de mejorar aún más la capacidad predictiva y la precisión de los modelos.