Diplomatura en Ciencia de Datos, Aprendizaje Automático y sus Aplicaciones

Mentoría Series Temporales

Mentor: Ezequiel Medina

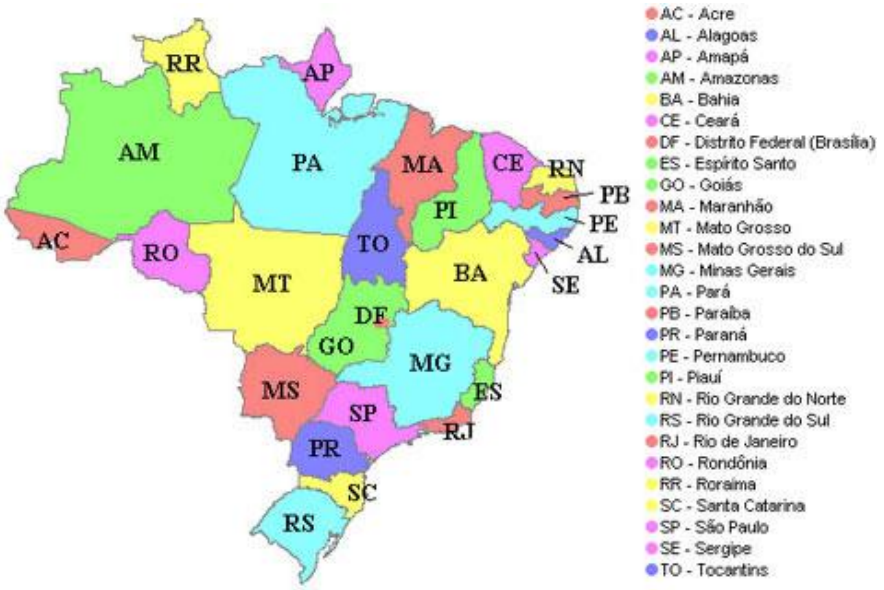
Integrantes: Fernando Mancuso
Isabel Rivadero
Miguel Vargas
Pablo Madoery
Nehuen Montoro

Agenda

1. Descripción del dataset (Isabel y Fernando)
2. Problemática (Nehuen)
3. Modelos y resultados(Fernando)
4. Próximos pasos (Pablo)

Descripción del dataset

Envios de Mercado Libre en Brasil



Descripción del dataset

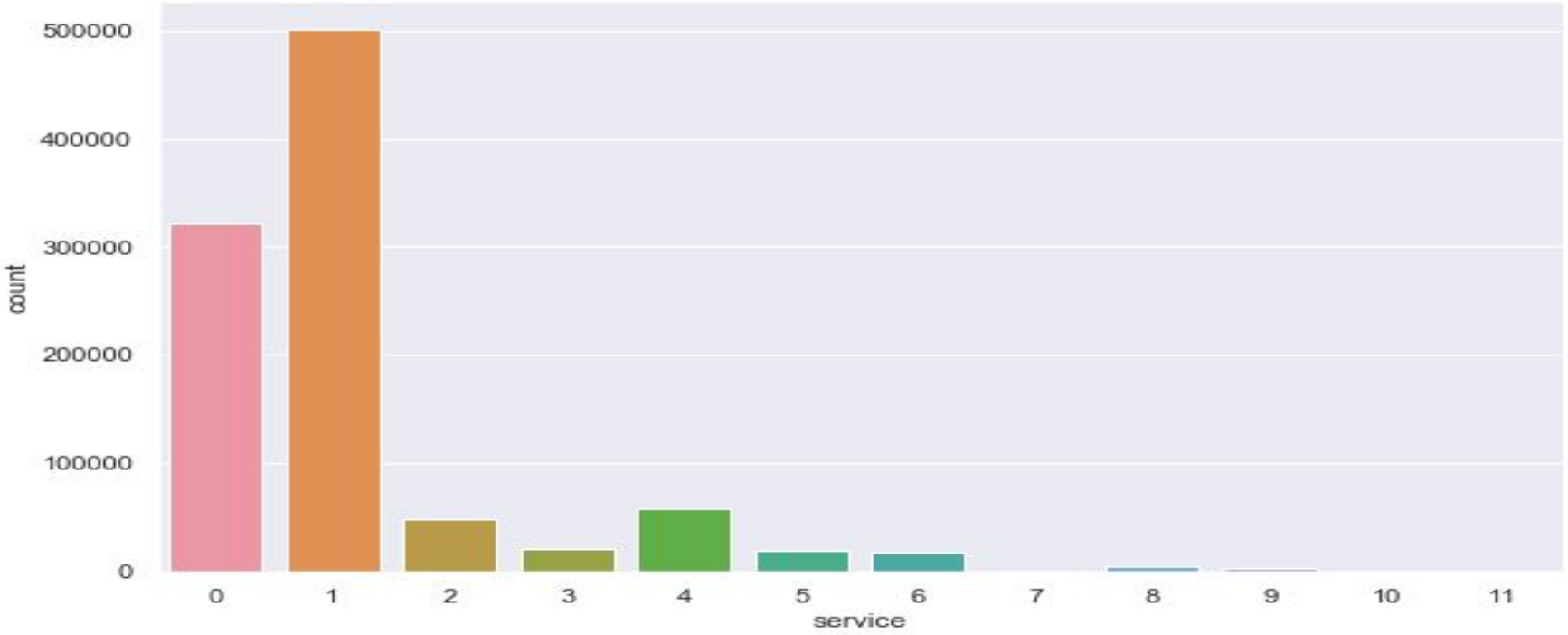
Envios de Mercado Libre en Brasil

	sender_state	sender_zipcode	receiver_state	receiver_zipcode	shipment_type	quantity	service	status	date_created	date_sent	date_visit	target
0	SP	3005	SP	5409	express	1	0	done	2019-03-04 00:00:00	2019-03-05 13:24:00	2019-03-07 18:01:00	2
1	SP	17052	MG	37750	standard	1	1	done	2019-03-19 00:00:00	2019-03-20 14:44:00	2019-03-27 10:21:00	5
2	SP	2033	SP	11040	express	1	0	done	2019-02-18 00:00:00	2019-02-21 15:08:00	2019-02-28 18:19:00	5
3	SP	13900	SP	18500	express	1	0	done	2019-03-09 00:00:00	2019-03-11 15:48:00	2019-03-12 13:33:00	1
4	SP	4361	RS	96810	express	1	0	done	2019-03-08 00:00:00	2019-03-12 08:19:00	2019-03-16 08:24:00	4

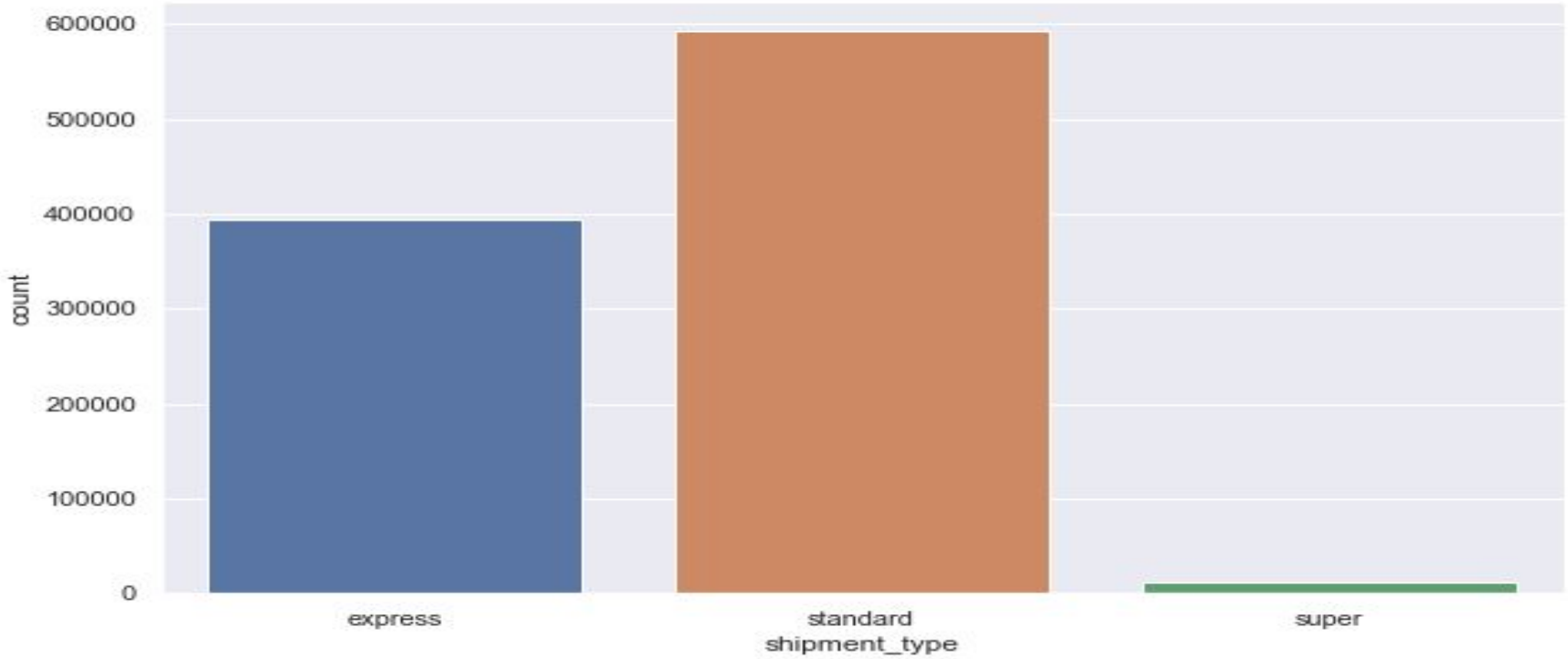
Features

- **service**: Identificador unico de un tipo de servicio de un correo en particular.
- **shipment_type**: Tipo de envío: standard, express, super
- **sender_zipcode:** Código postal de quien envía el paquete (usualmente el vendedor).
- **receiver_zipcode:** Código postal de quien recibe el paquete (usualmente el comprador).
- **sender_state:** Nombre abreviado del estado de quien envía el paquete.
- **receiver_state:** Nombre abreviado del estado de quien recibe el paquete.
- **quantity:** Cantidad de items que tiene dentro el paquete.
- **status:** Estado final del envío.
- **date_created:** Fecha de compra de el o los items.
- **date_sent:** Fecha en que el correo recibe el paquete.
- **date_visit:** Fecha en que el correo entrega el paquete.
- **target:** Cantidad de dias hábiles que tardó el correo en entregar el paquete desde que lo recibe.

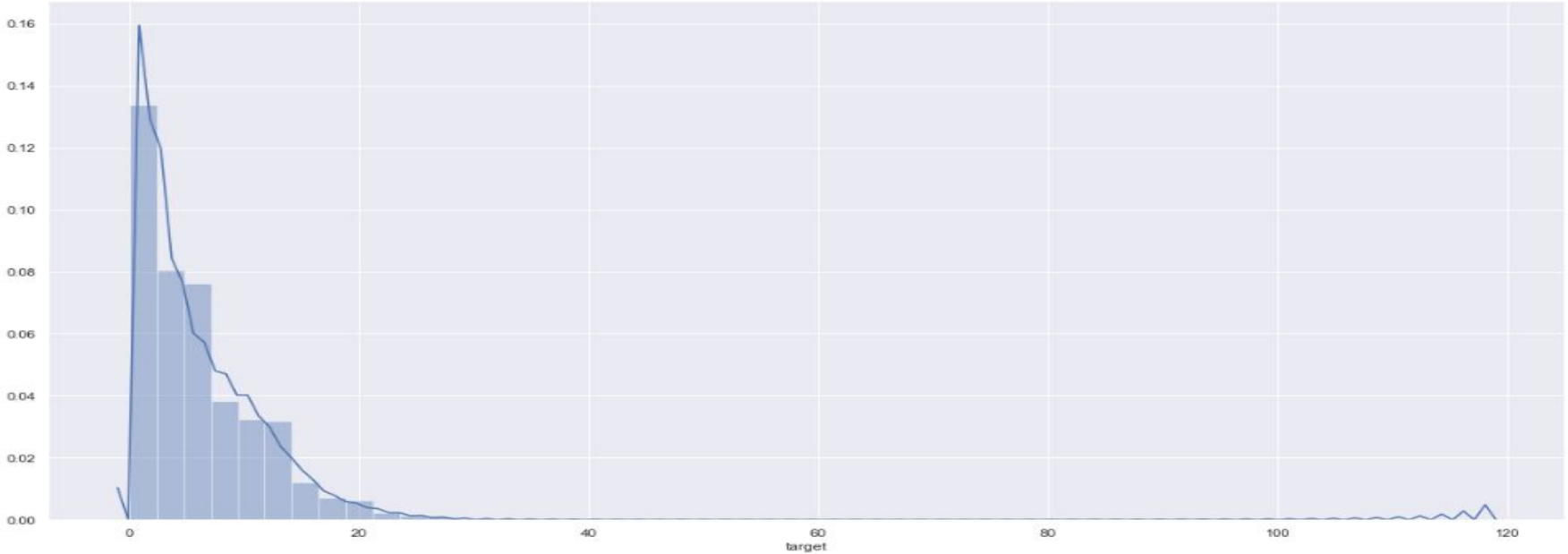
Cantidad de envíos por tipo de servicio



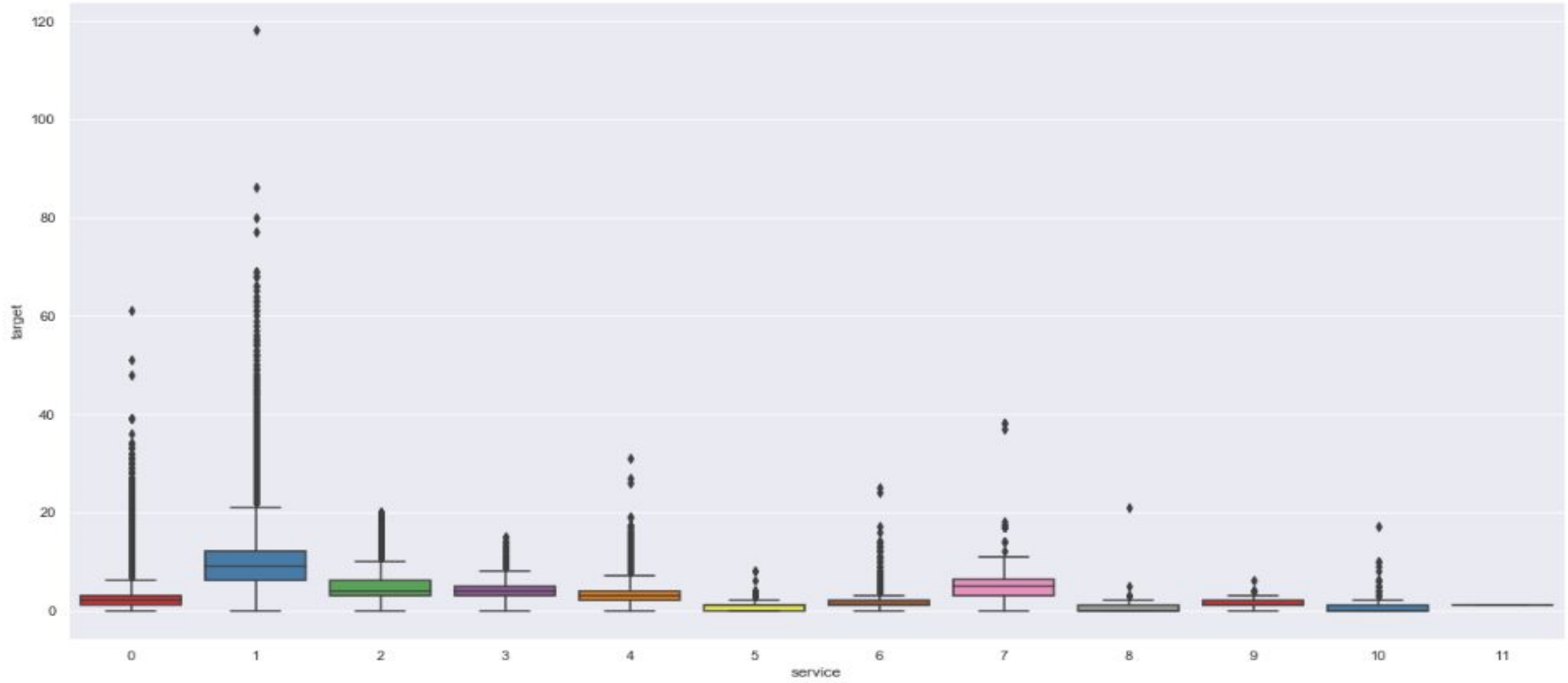
Cantidad de envíos por tipo de envío



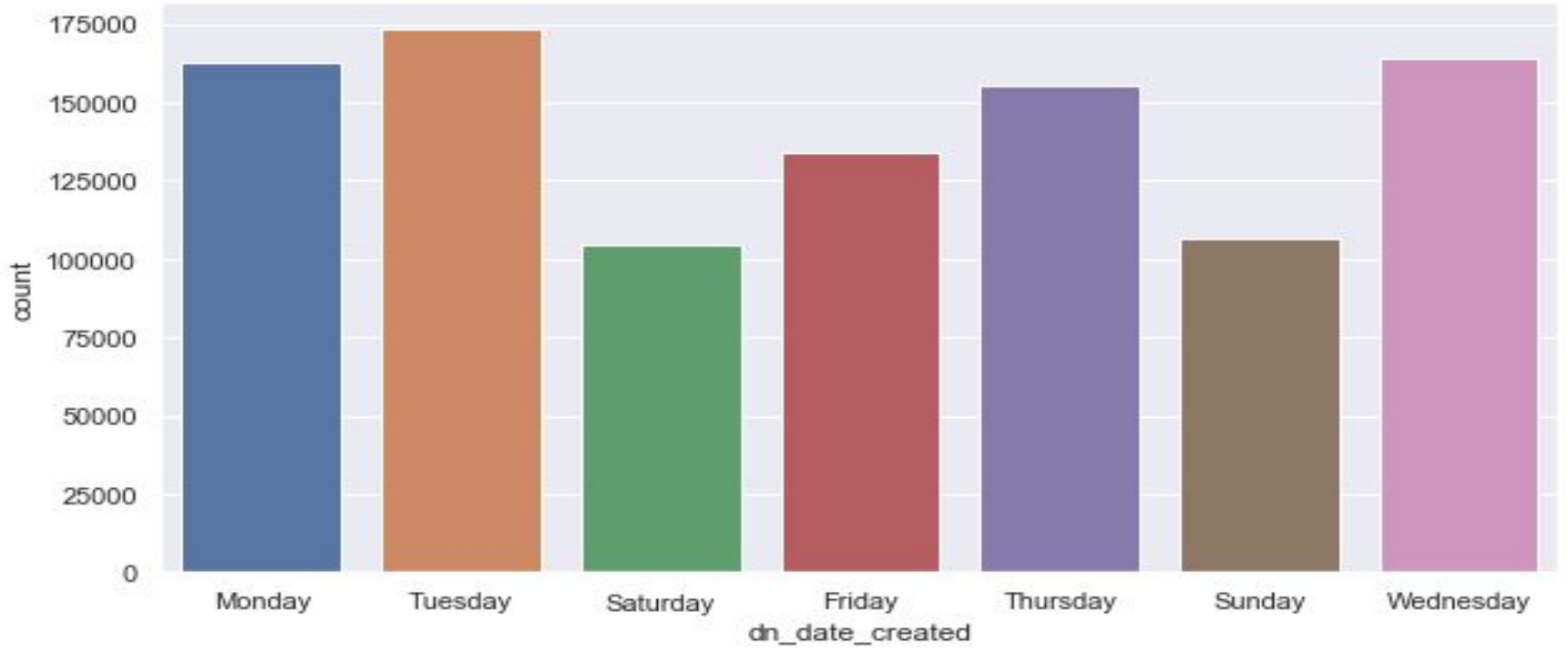
Distribución del target



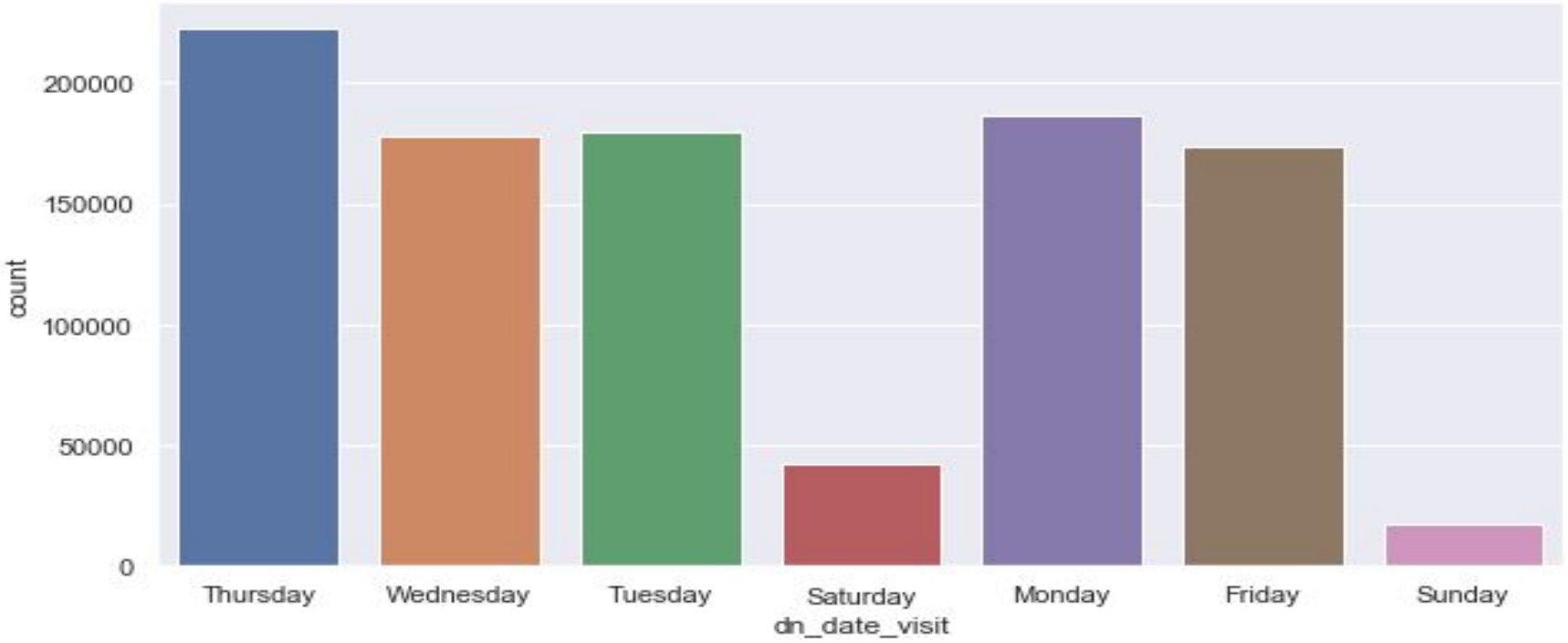
Boxplot de target vs service



Compras según el día

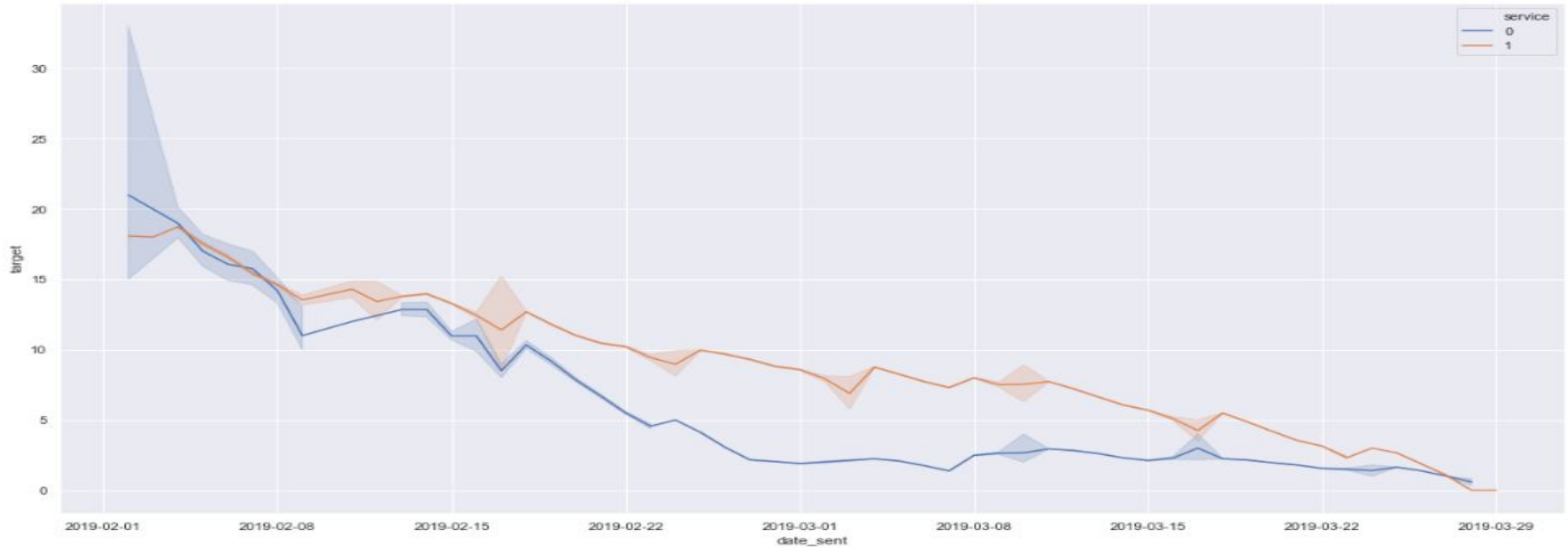


Llegada de paquetes según el día

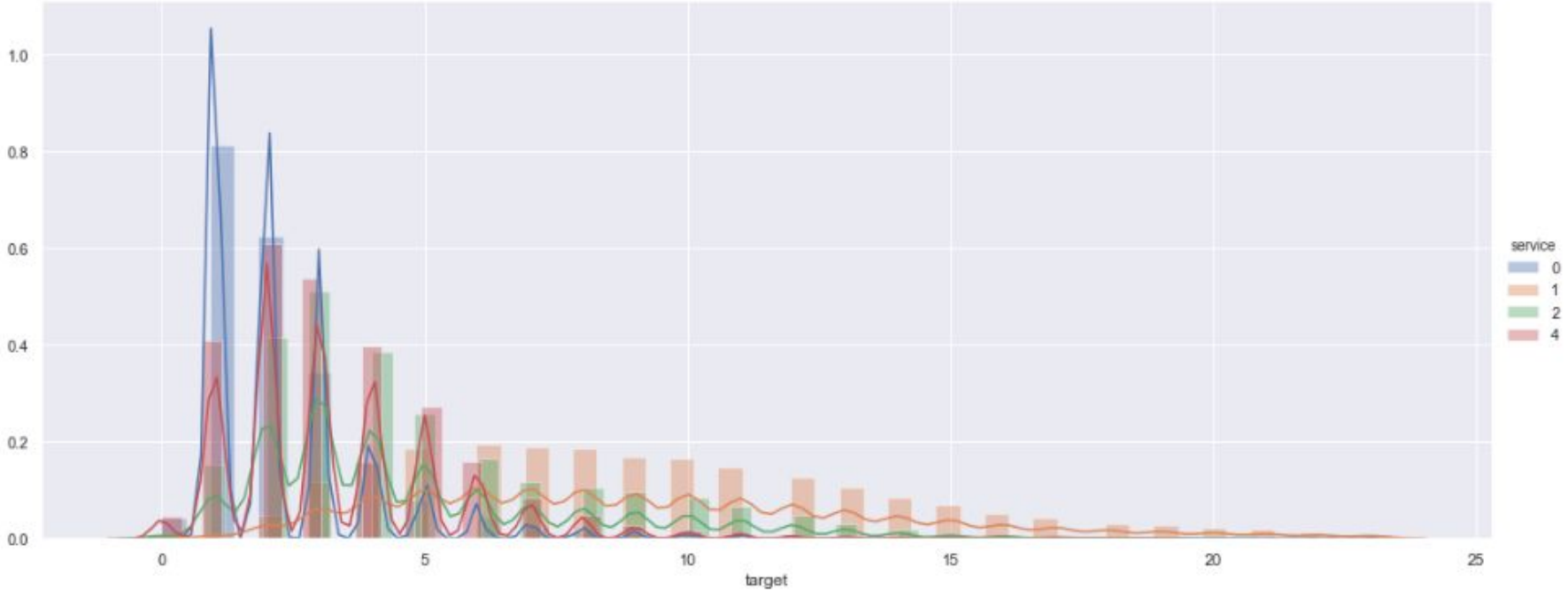


Evolución del target para los servicios 0 y 1

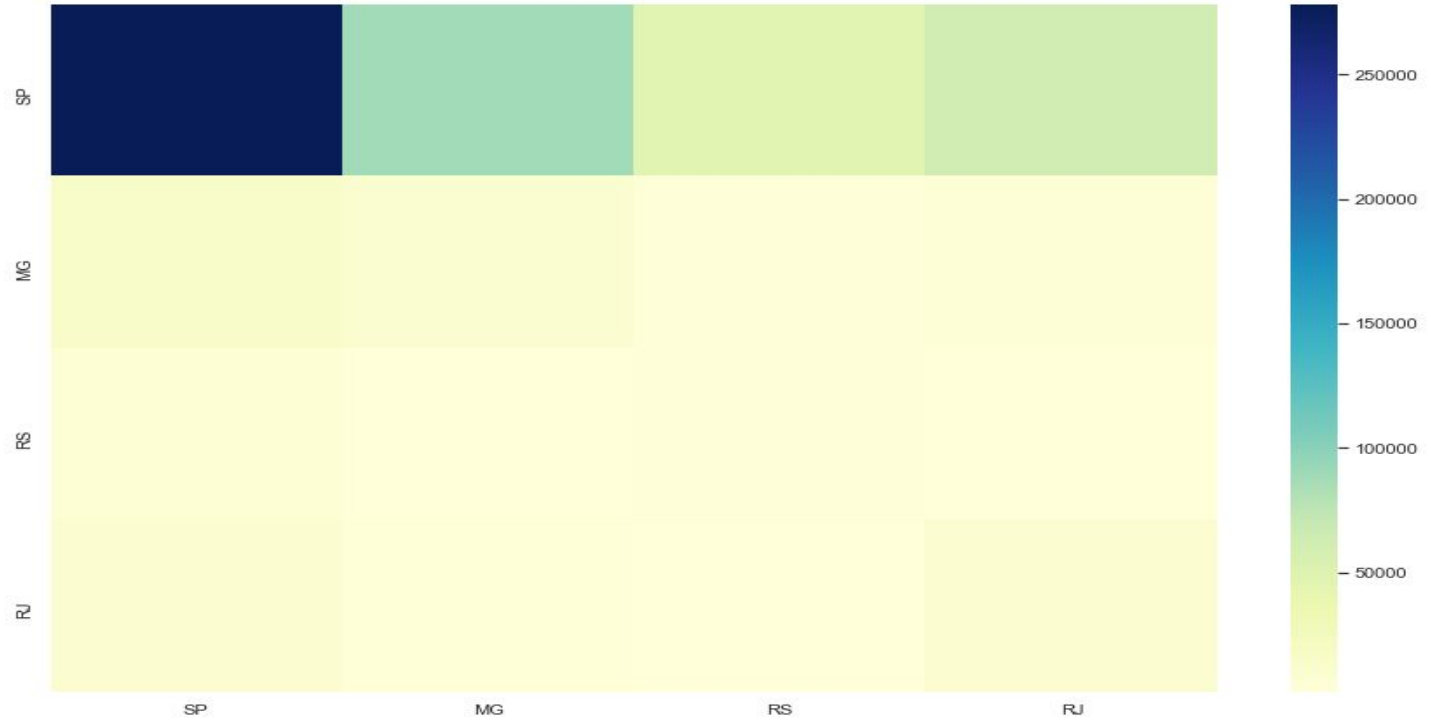
Analizamos como se comportan los servicios 0 y 1(concentran más del 80% de los correos enviados) que llegaron en marzo de 2019



Distribuciones del target para los 4 servicios mas representados:



Heat-map de las rutas de compradores a vendedores



ANÁLISIS Y CURACIÓN DE DATOS

- Verificamos la consistencia de la información
  - Los ids son únicos
  - Si no tuviéramos estos índices**:** Fortalecemos con datos de la ruta(zipcodes).
  - No hay valores faltantes

ANÁLISIS Y CURACIÓN DE DATOS

- Fechas inconsistentes:
  - 171 compras que se realizaron después de que fueron enviadas.
  - 168 compras que se realizaron después de que fueron entregadas.
  - 173 filas con datos inconsistentes con respecto a las fechas.

Otras consideraciones

- **Riesgos al imputar datos:** Que el dato genere sesgos
- **Normalización:** Solo features numéricos.
- **Reducción de dimensionalidad:** PCA solo con features numéricos
- **Marcado desbalanceo de clases:** Concentración en pocos estados.

Aplicamos todas las curaciones

1. Eliminamos fechas incorrectas
2. Corregimos targets incorrectos
3. Eliminamos outliers

Boxplot del target

Problemática

El objetivo principal de este trabajo es el de intentar **predecir el tiempo que le llevará a un envío llegar a destino:**

- **Problema de clasificación con dos clases:**
  - **Rápido:** si tarda hasta 3 días hábiles.
  - **Lento**: si tarda más de 3 días hábiles.

- P**roblema de clasificación con 21 clases:**
  - clases de 0 a 19 hábiles
  - una clase que representa a aquellos envíos que tardan más de 19 días.

Problemática

División del dataset en **train/test**

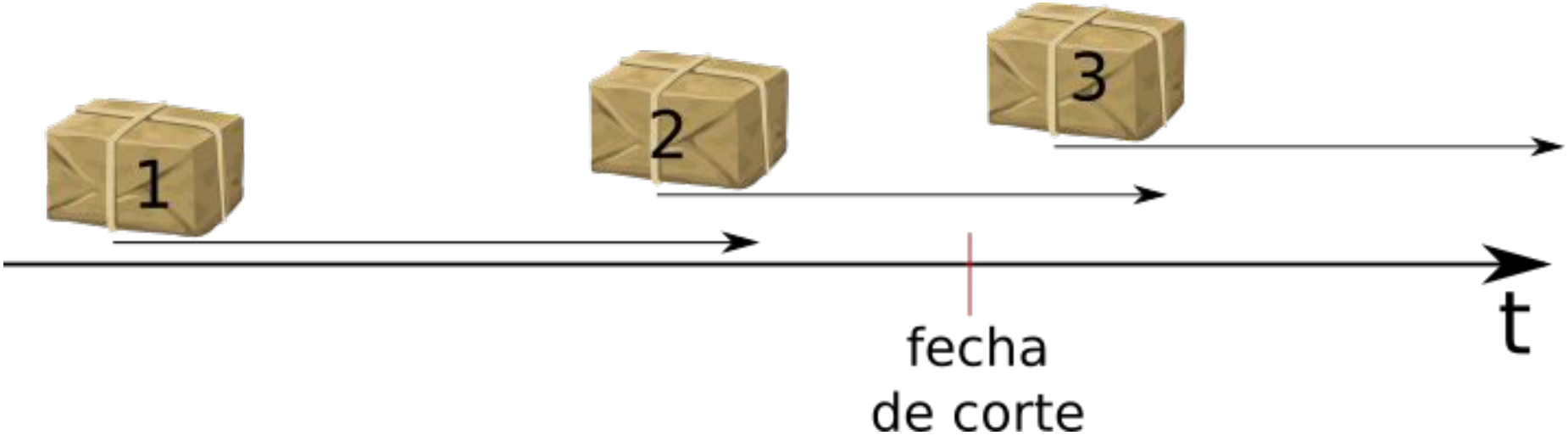
Cada envío tiene tres features que son timestamps:

- **date_created**: es la fecha de la compra en el sitio web.
- **date_sent**: es la fecha en que el envío entra al correo.
- **date_visit**: es la fecha en que el paquete llega a destino.

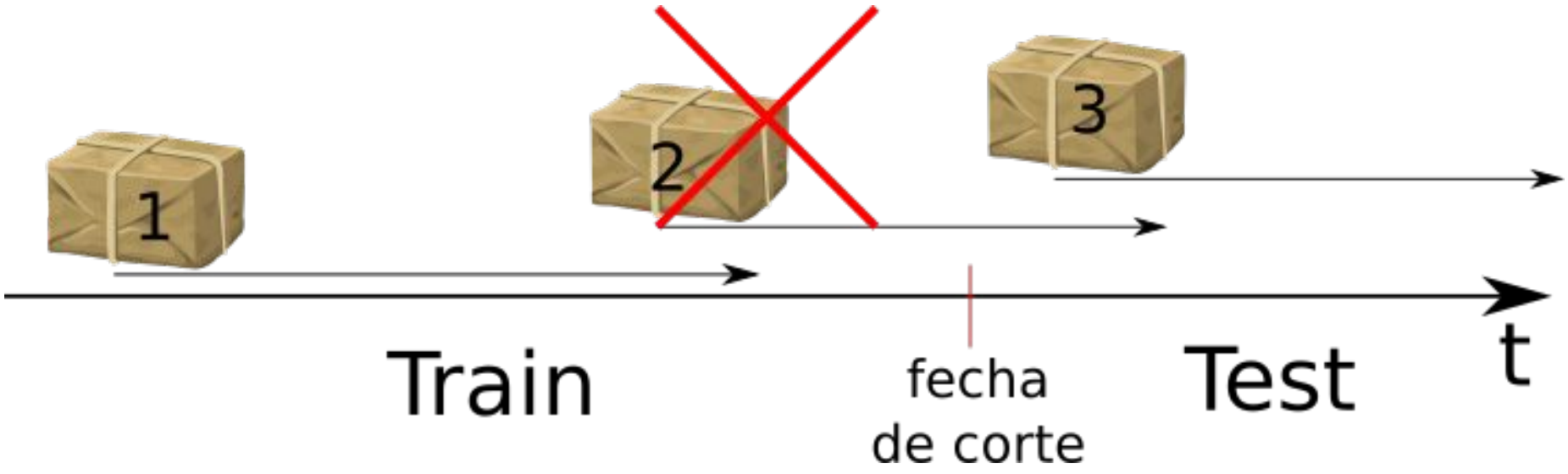
Como queremos caracterizar el comportamiento de los correos las fechas relevantes son date_sent y date_visit y la diferencia entre ellas es el target.

Para una correcta división del dataset hay que establecer una fecha y conservar para train todos aquellos envíos que hayan llegado antes de esa fecha y para test aquellos que sean enviados a partir de esa fecha.

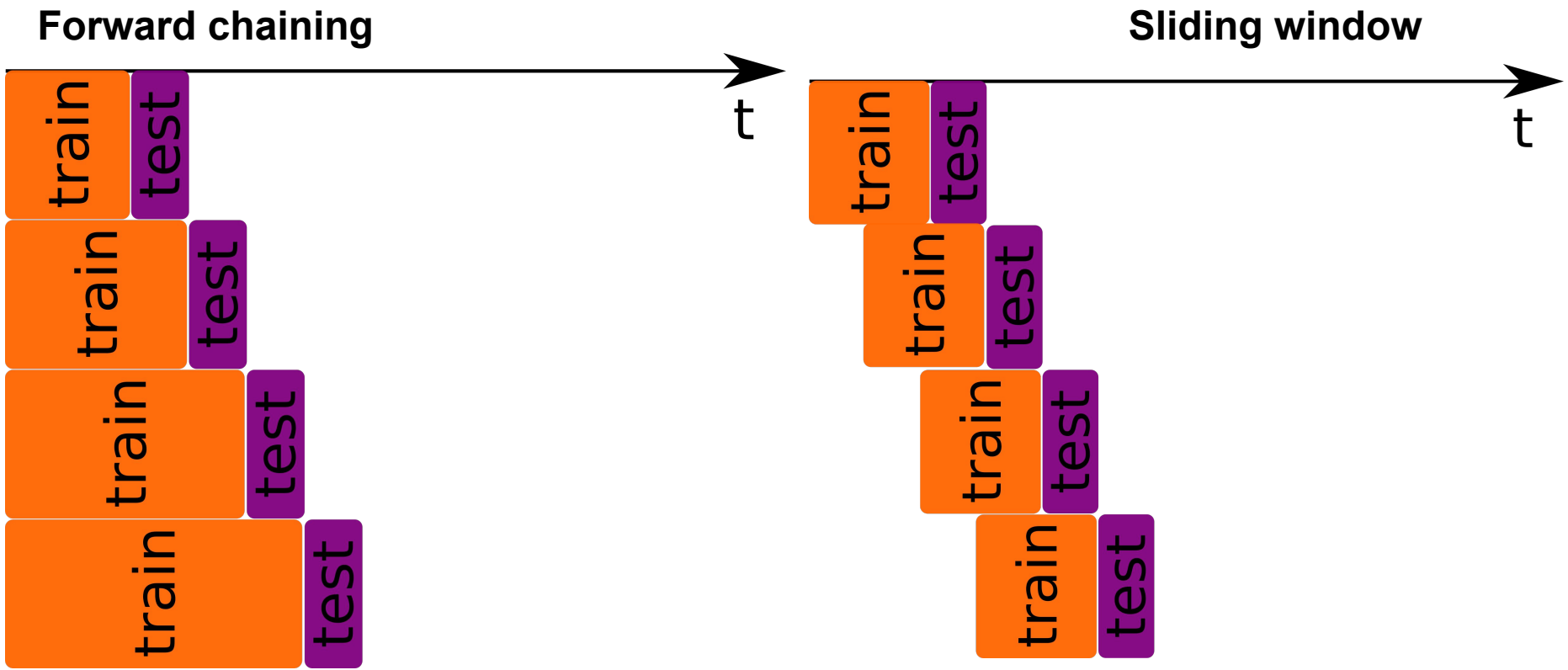
Problemática



Problemática



Problemática

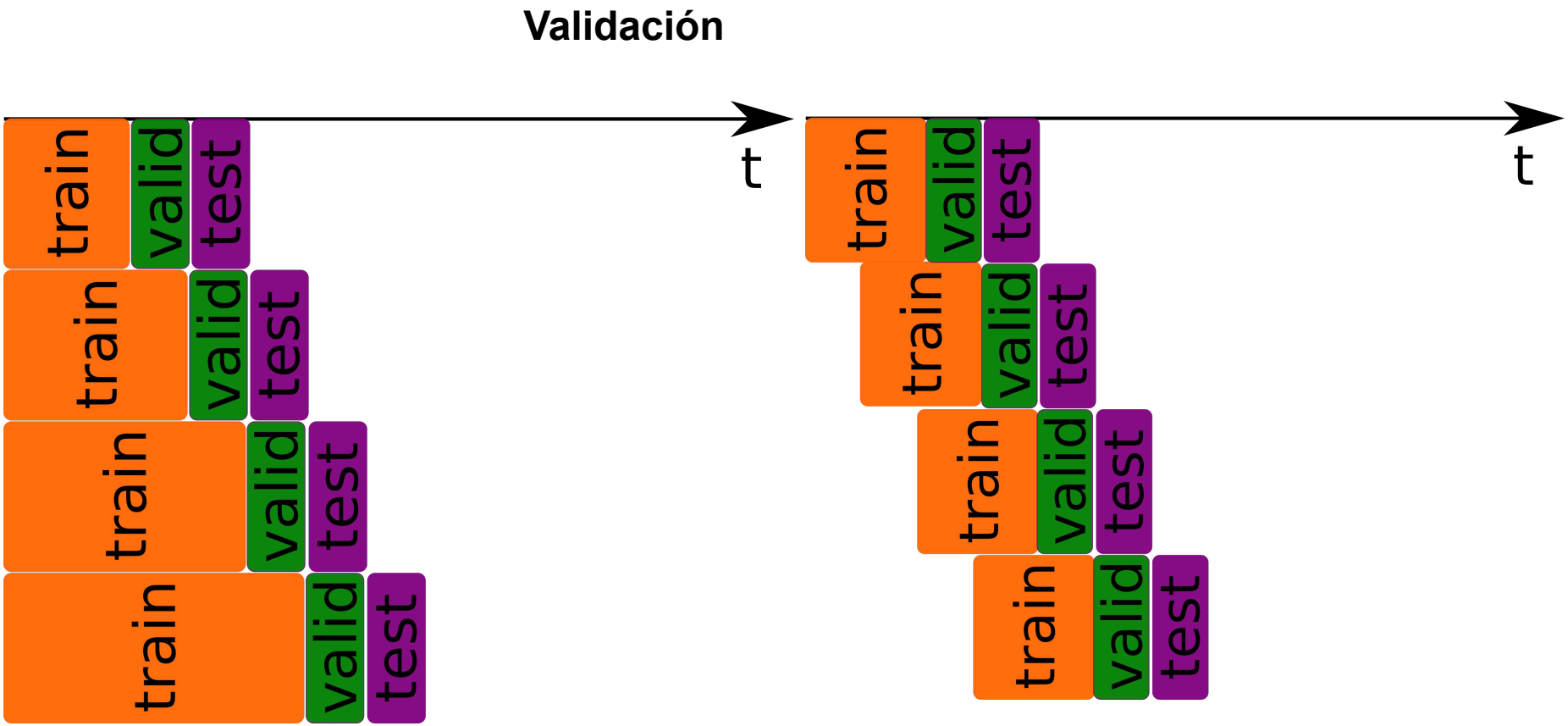


Problemática

¿Cómo Validamos?

Problemática

Problemática



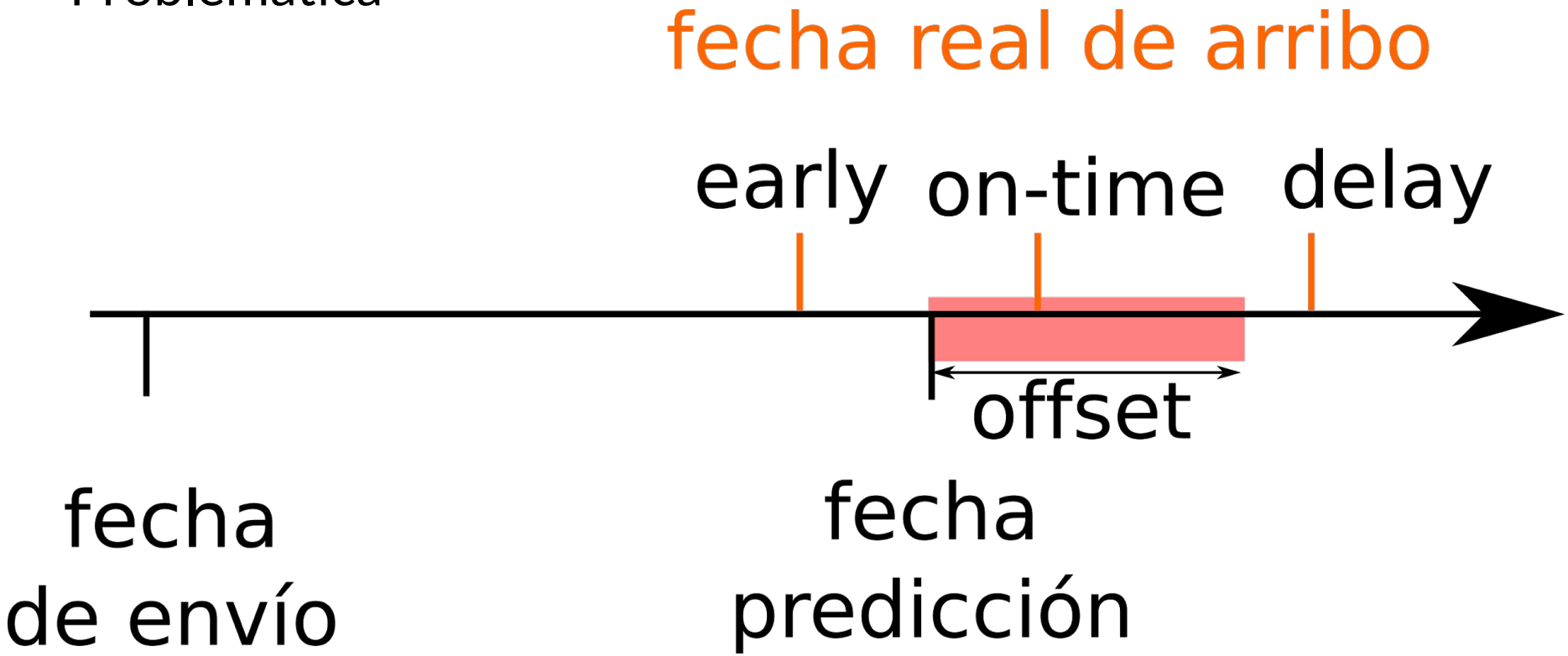
Problemática

Definición de **métricas**:

- Rápido/Lento:
  - Utilizamos **accuracy** como métrica

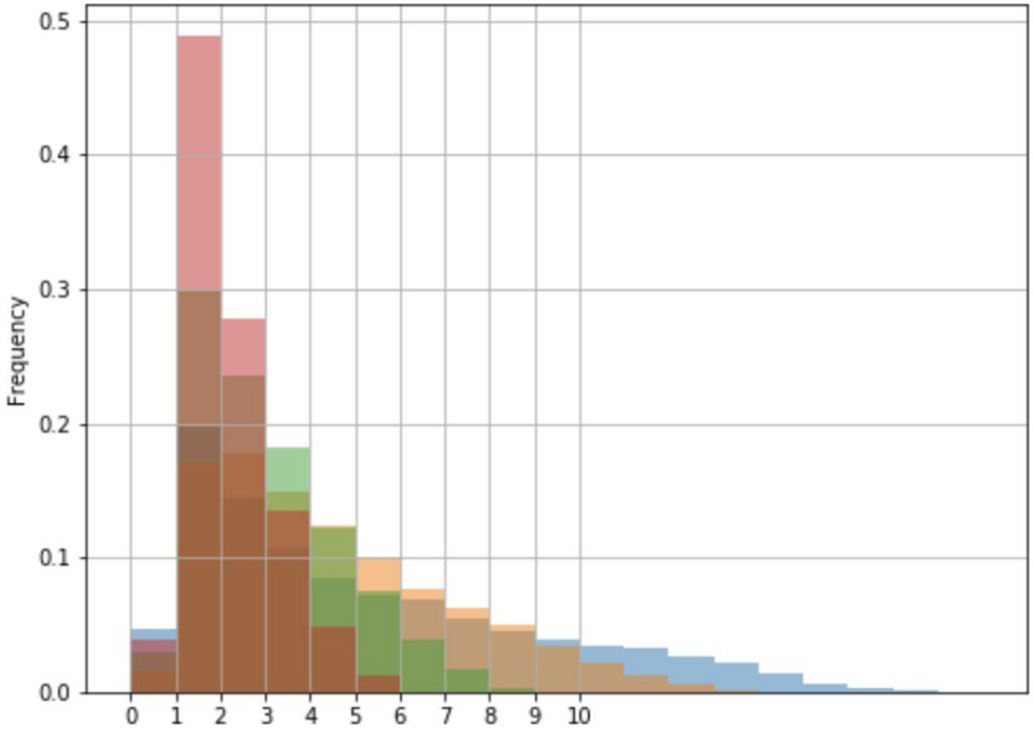
- Para el caso con 21 clases :
  - **on_time**: nos dice si el valor real de la etiqueta está entre el valor predicho y un margen al que llamaremos **offset**.
  - **delay**: nos dice si el valor real de la etiqueta está por encima del valor predicho más el offset.
  - **early**: nos dice si el valor real de la etiqueta está por debajo del valor predicho

Problemática



Problemática

Datos Censurados / Truncados



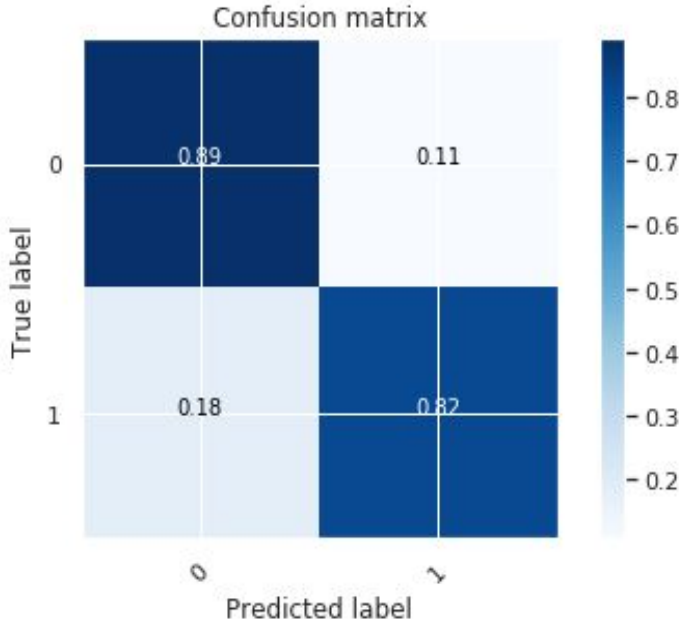
Introducción al Aprendizaje Automático

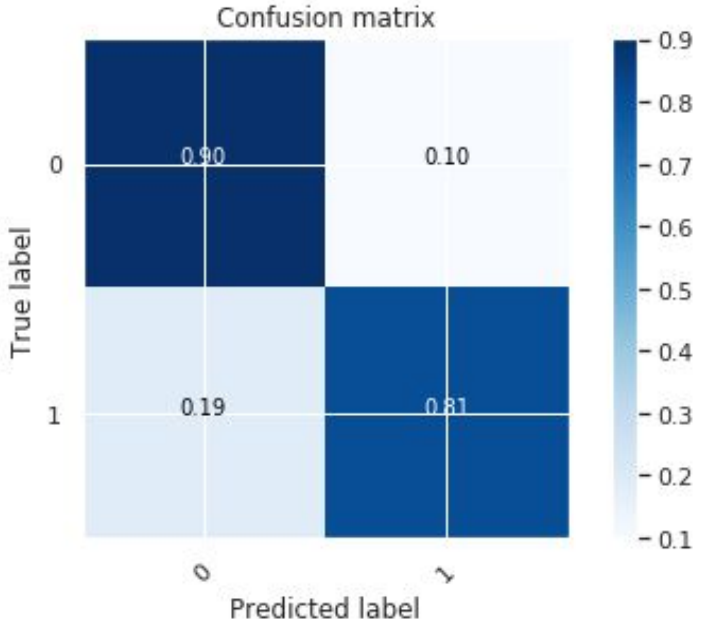
Problema de clasificación binaria

- Para salvar las rutas poco representadas, implementamos una codificación para los features sender_zipcode y receiver_zipcode: recortamos ultimo digito para reducir la granularidad de los zipcodes.

- Seleccionamos un conjunto de features numericos para entrenar modelos de machine learning y mediante las correlaciones de spearman, kendall y pearson determinamos cuales features guardan relacion y estos son los que usamos para entrenar

(RLineal y Logistica)Estandarizando los features y re entrenando:

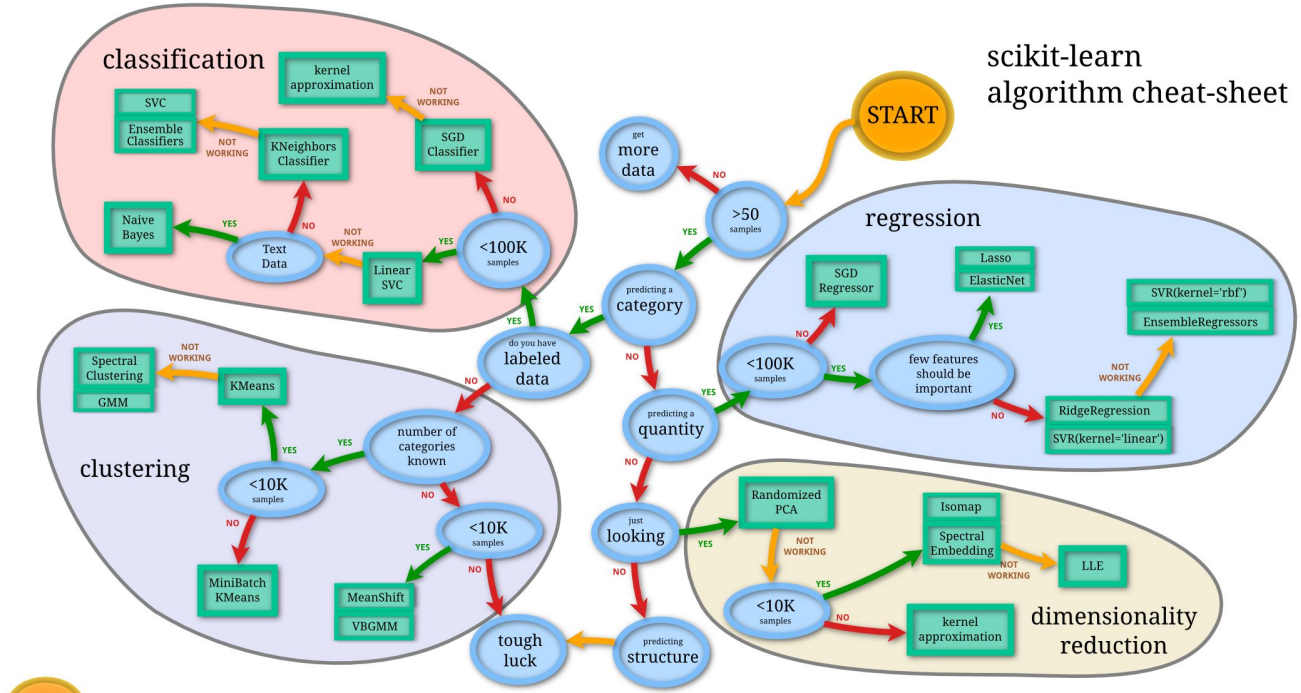




El mejor método para clasificarlos fue LogisticRegression.

En el kmeans solo usamos las rutas y el accuracy nos dió alrededor de 0.65, esto nos dice que la ruta influye mucho en el tiempo que tarda en llegar un paquete. Sin embargo, en las regresiones usamos además otros features y nos dieron mejores métricas lo que podríamos interpretar como que quizás el kmeans andaría mejor con mas features

Modelos


scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
NO
Text Data
NOT WORKING
Linear SVC
YES
<100K samples
NO
get more data
NO
>50 samples
YES
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
YES
NO
<100K samples
YES
few features should be important
NOT WORKING
NO
RidgeRegression
SVR(kernel='linear')
YES
predicting a category
YES
NO
do you have labeled data
YES
NO
predicting a quantity
YES
NO
clustering
Spectral Clustering
GMM
NOT WORKING
KMeans
YES
number of categories known
YES
NO
<10K samples
NO
MiniBatch KMeans
<10K samples
YES
NO
MeanShift
VBGMM
just looking
YES
NO
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
tough luck
predicting structure
Back
scikit learn

Modelo basado en árboles de decisión (supervisado)

- Agregamos el clasificador XGBoostClassifier como estimador final. Entrenamos el modelo, predecimos el conjunto de test y calculamos las métricas ontime, delay y early, sin ventana

```
{'ontimesv': 0.5394927152785127,
 'delaysv': 0.22330405333897768,
 'earlysv': 0.23720323138250962}
```

Modelo basado en vecinos cercanos (no supervisado / semi supervisado)

- Agregamos el clasificador KNeighborsClassifier como estimador final. Calculamos las métricas ontime , delay y early , sin ventana.

```
{'ontimesv': 0.5231065015698311,
 'delaysv': 0.21339118778001198,
 'earlysv': 0.263502310650157}
```

Modelo basado en regresión:

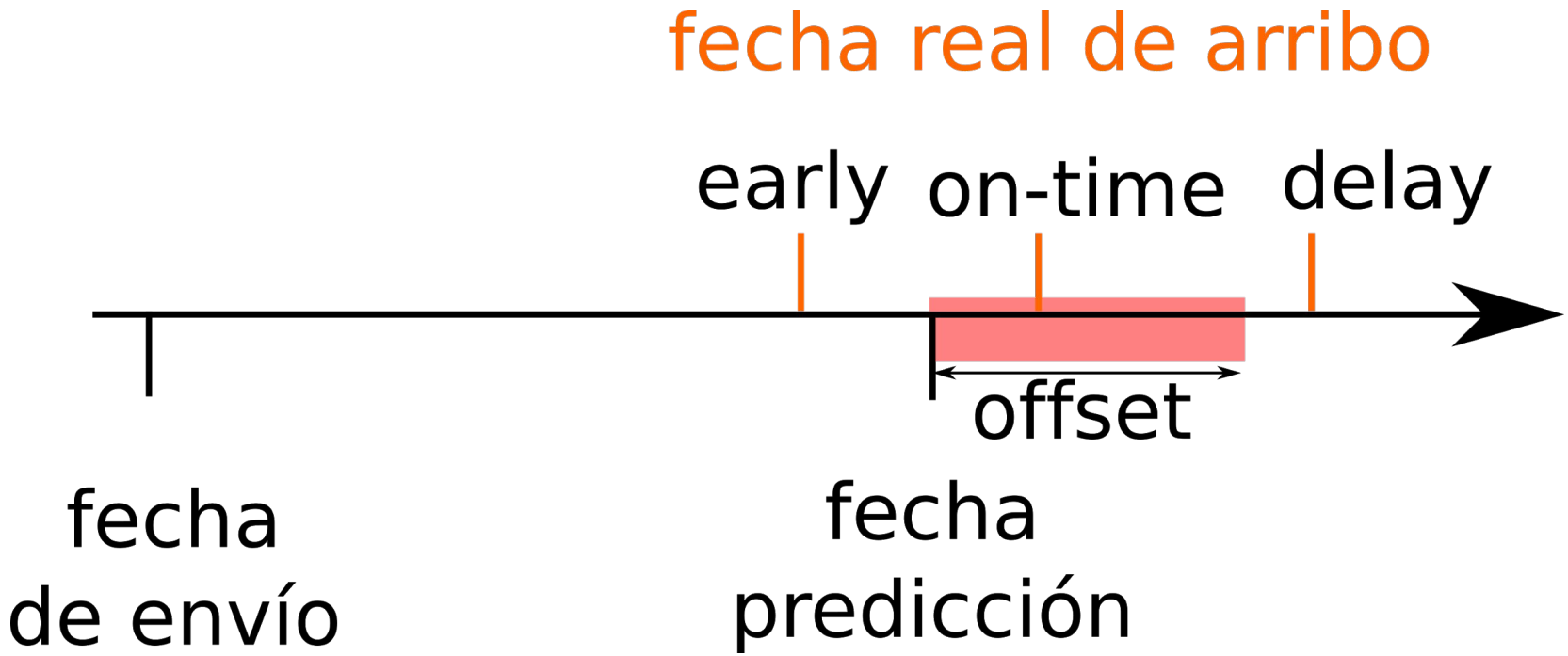
- Agregamos LogisticR como estimador final, calculamos métricas ontime , delay y early , sin ventana.

```
{'ontimesv': 0.4793805340953187,
 'delaysv': 0.37123857903834623,
 'earlysv': 0.14938088686633505}
```

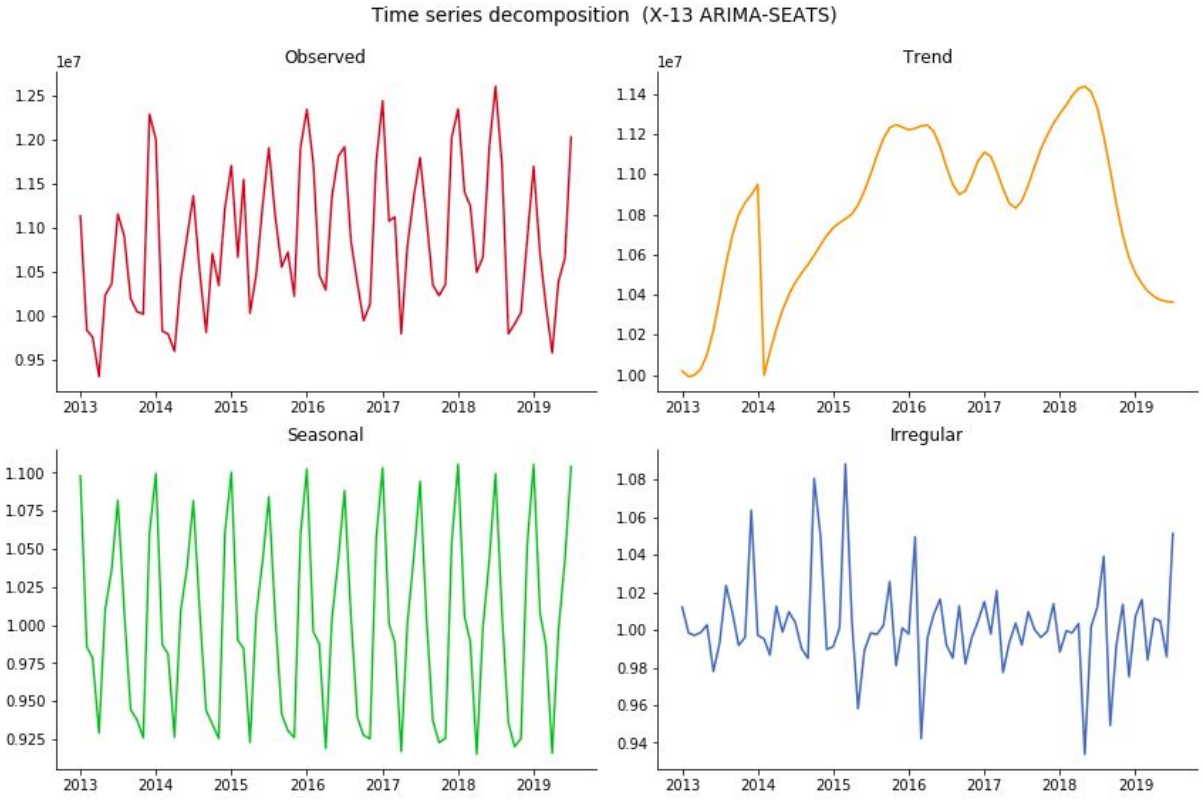
Elegimos Logistic Regression porque da informacion de la probabilidad que tiene cada etiqueta en cada prediccion que se hizo lo que puede ser util para elegir el offset. Además da informacion de que tan segura es posible estar de la prediccion hecha.

Próximos Pasos

Mejorar Ventanas de predicción y considerar otras métricas



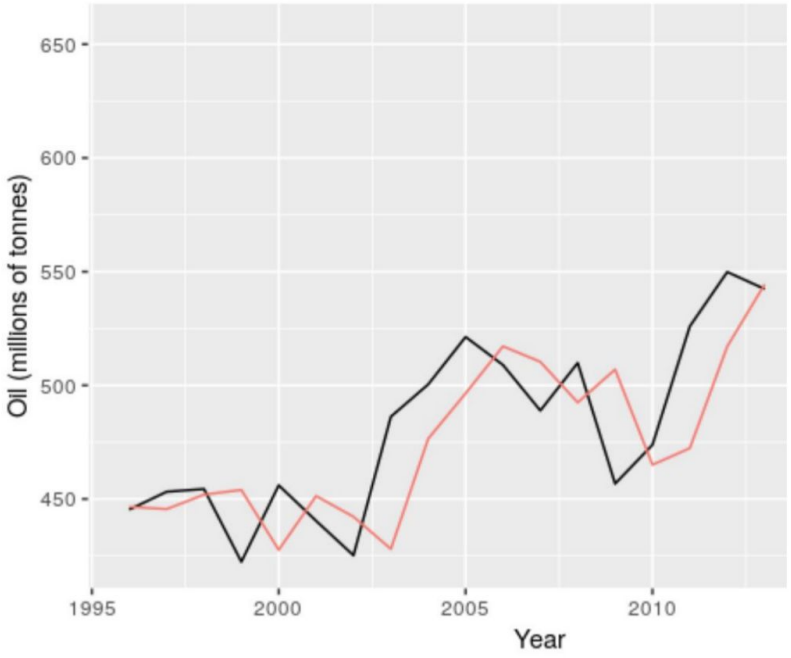
En la materia optativa **Series Temporales** vimos que el **análisis** suele ser **diferente** a los métodos que utilizamos en materias anteriores:



Exponential smoothing

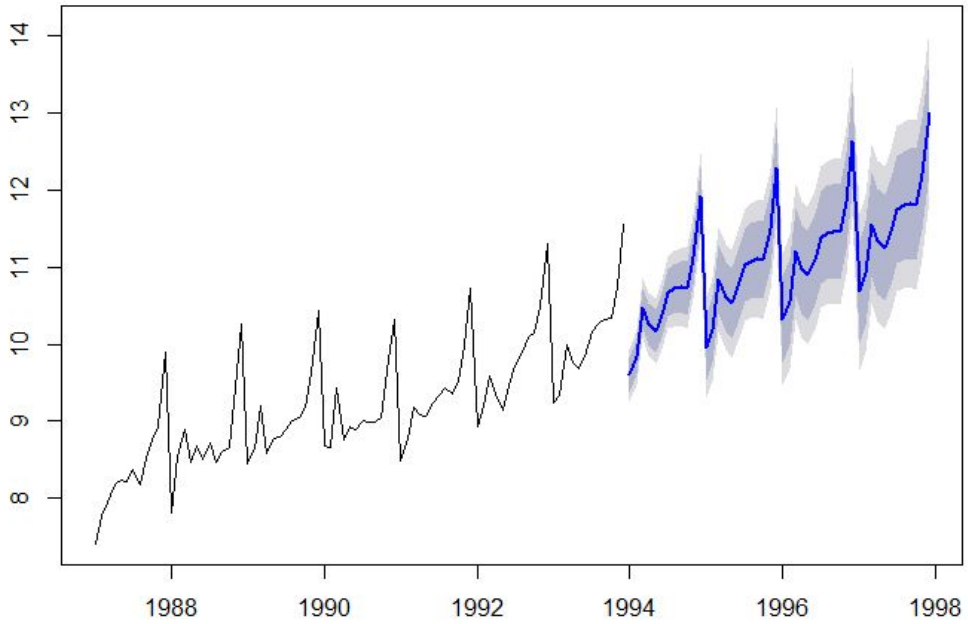
las predicciones son promedios pesados de observaciones pasadas,
donde los pesos decaen exponencialmente cuando nos movemos al pasado.

Se suele usar cuando los datos no tienen patrones de tendencia ni estacionales claros



Holt Winters

incorpora el manejo de la tendencia y agrega correcciones de patrones estacionales en los datos



ARIMA (p,d,q): intentan describir la autocorrelación en los datos.

Suelen ser complementarios a exponential smoothing.

No considera estaciones.

p: orden de los términos de la autocorrelación.

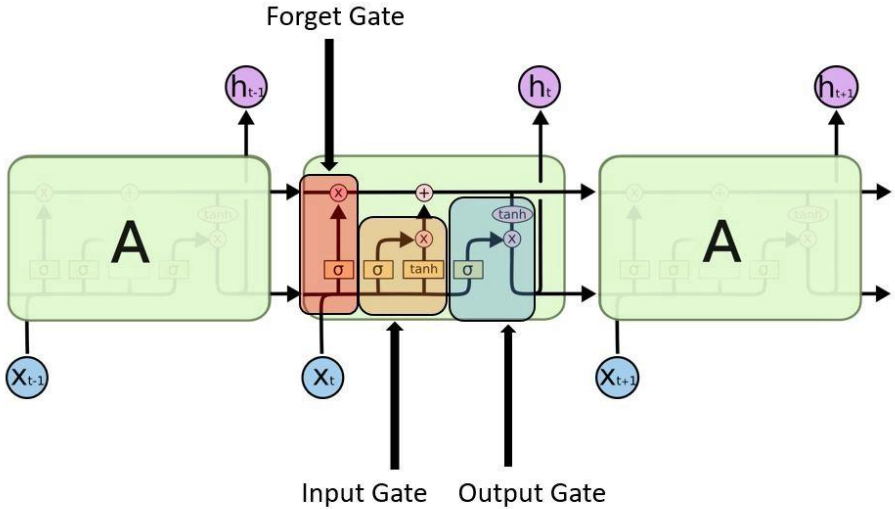
q: orden de los términos de medias móviles

d: orden de diferenciación para alcanzar la estacionalidad

Seasonal ARIMA: considera estaciones.

Long Short Term Memory (LSTM)

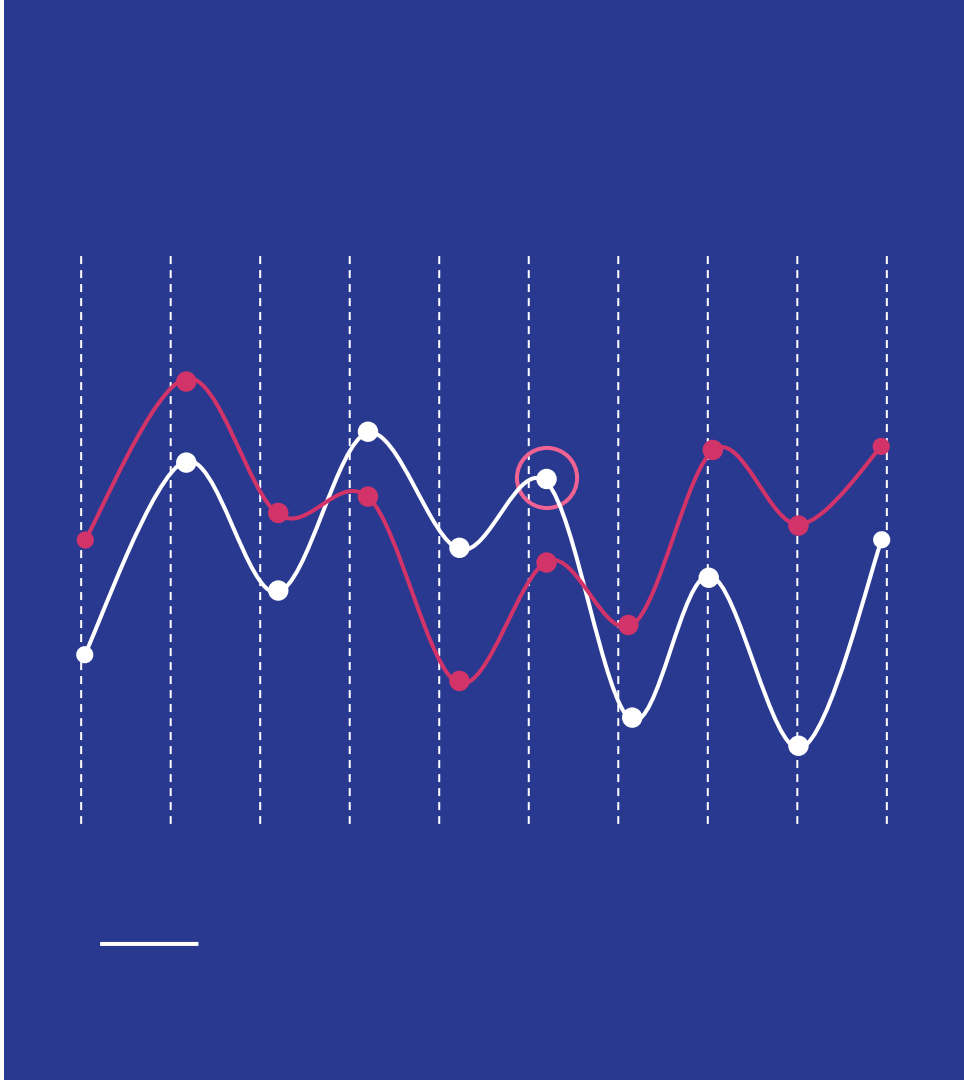
versión modificada redes neuronales recurrentes capaces de “recordar” datos pasados



Agregar nuevos features: google maps: distancias, rutas, etc

Realizar heat-maps con los resultados de distintos modelos

Gracias!!!



Backup