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

esquema de ejemplo:

1. presentar el dataset y posibles problemas o aplicaciones del mismo
2. describir el dataset (lo que aprendimos en el práctico de visualización), mostrando qué cosas pueden ser prometedoras y qué cosas apuntan a problemas (clases desbalanceadas? valores con distribuciones difíciles? valores faltantes? posibles errores?)
3. proponer diferentes formas de atacar estos problemas (mas o menos lo aprendido en el práctico de curación)
4. mostrar resultados de clasificación (prácticos de supervisado)
5. si se tienen, se muestran algunos resultados de clustering. Este punto puede preceder al 4. porque clustering se puede usar como análisis exploratorio de datos, por lo tanto sería previo a clasificación
6. se pueden presentar variantes con diferentes subconjuntos de características, embeddings y, por supuesto, clasificadores distintos
7. se termina con posibles extensiones de este trabajo, o sea, todo lo que habrían querido hacer pero no les dieron los tiempos

Agenda

1. Descripción del dataset
2. Problemática
3. Modelos y resultados
4. Optimización
5. Próximos pasos

Descripción del dataset

Presentar el dataset:

- Mostrar características del dataset..
- Mostrar las distribuciones de las variables…
- etc.. lo que vimos en el práctico de análisis

Descripción del dataset

- Mostrar dónde se encuentran las dificultades con las que se encontraron
- Como resolver estas dificultades..
- etc… Lo que vimos en curación
-

Problemática

- Presentar la problemática que queremos resolver (reducida y completa)
- Mostrar cómo decidimos atacar esa problemática
- Mostrar qué restricciones tenemos por trabajar con series temporales..

Modelos

- Presentar los modelos que se probaron:
  - no supervisados: clustering, knn
  - supervisados: logistic regression, xgboost
  - explicar mínimamente y a grandes razgos como funcionan
-

Modelos

- Mostrar los resultados obtenidos: métricas, matriz de confusión,..
- Mostrar los gráficos que les parezcan relevantes
-

Optimización

- Mostrar como eligieron los hiperparámetros y cuales funcionaron mejor
- Si se puede mostrar algún gráfico o tabla comparativa del proceso de selección

Próximos pasos

- Mostrar que nos faltó para probar:
  - algoritmos clásicos de series temporales
  - redes neuronales
  - embeddings
  - lo que se les ocurra que podrían probar a futuro...
-

Título

1. Contenido

Descripción del dataset

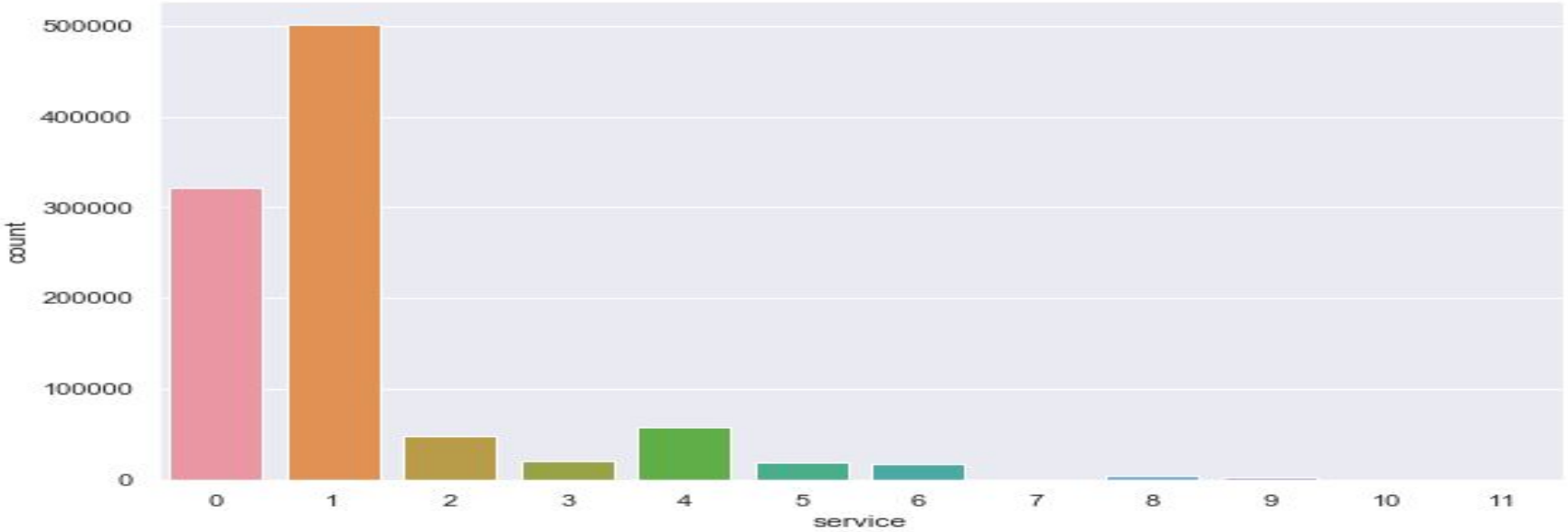
Envios de Mercado Libre en Brasil

	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

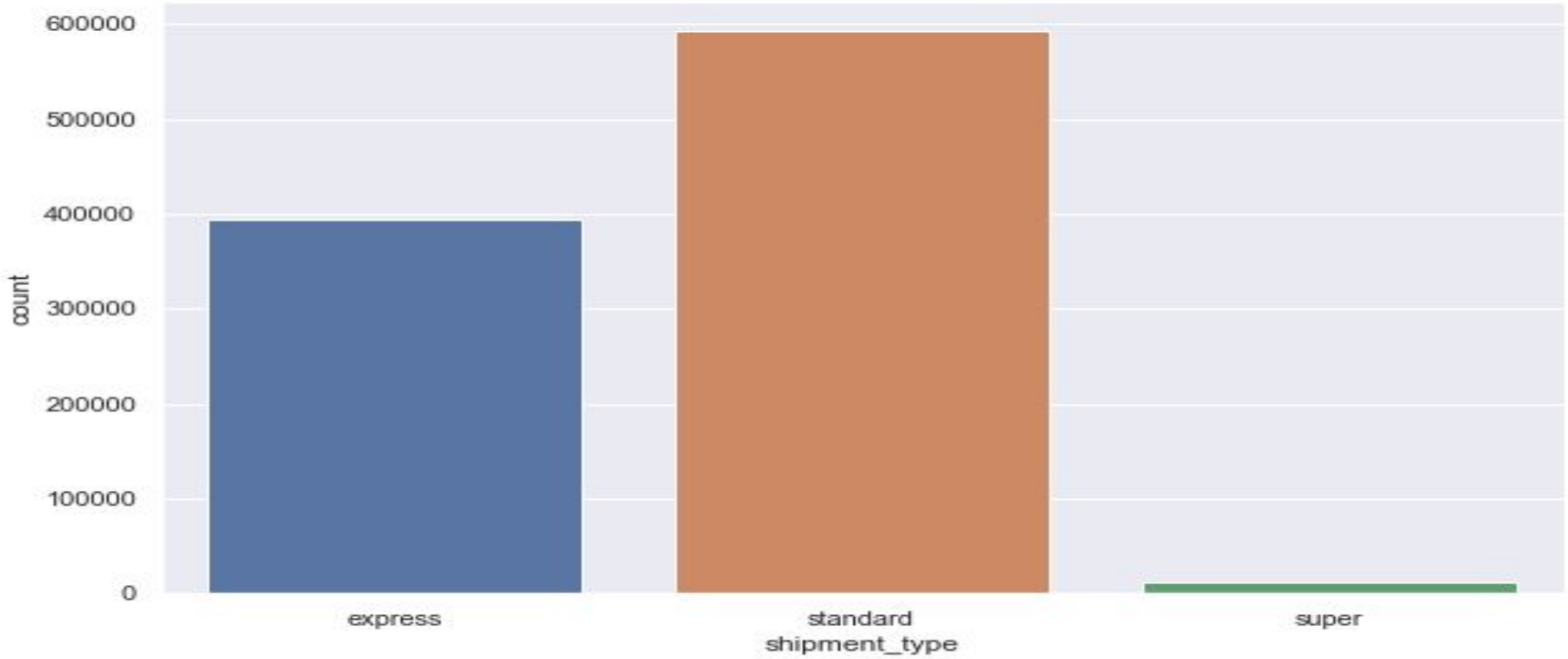
Features

- **service**: Identificador unico de un tipo de servicio de un correo en particular.
- **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.

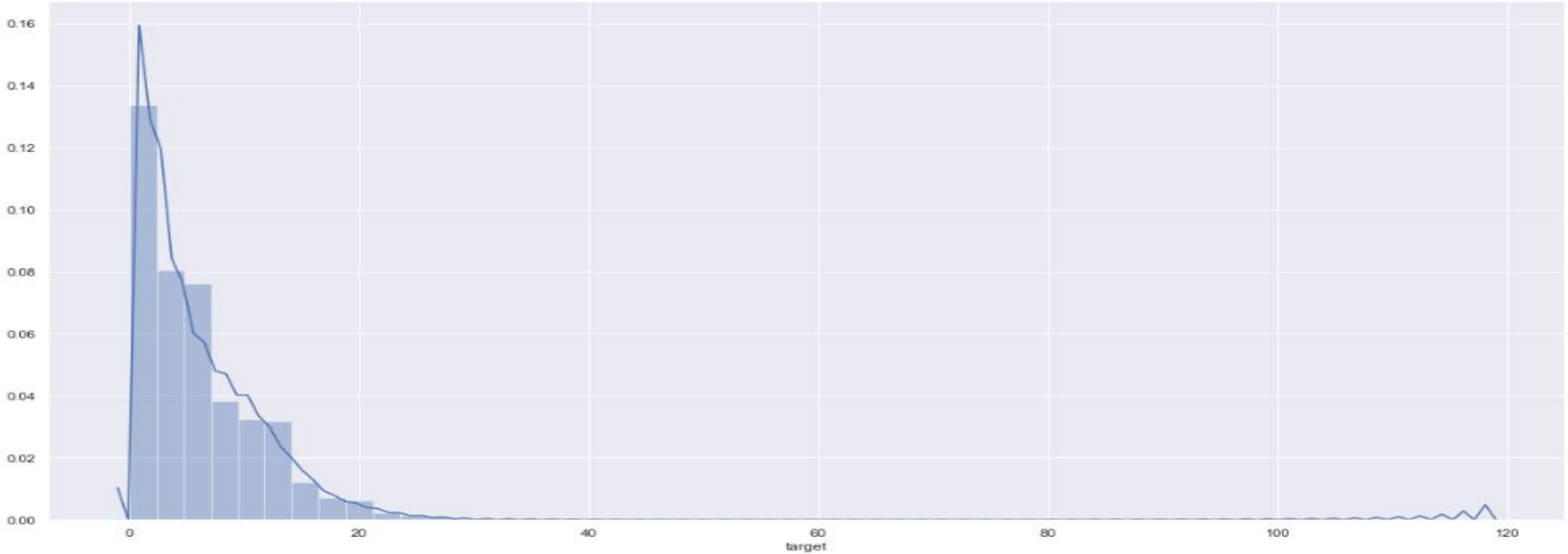
Cantidad de envíos por servicio



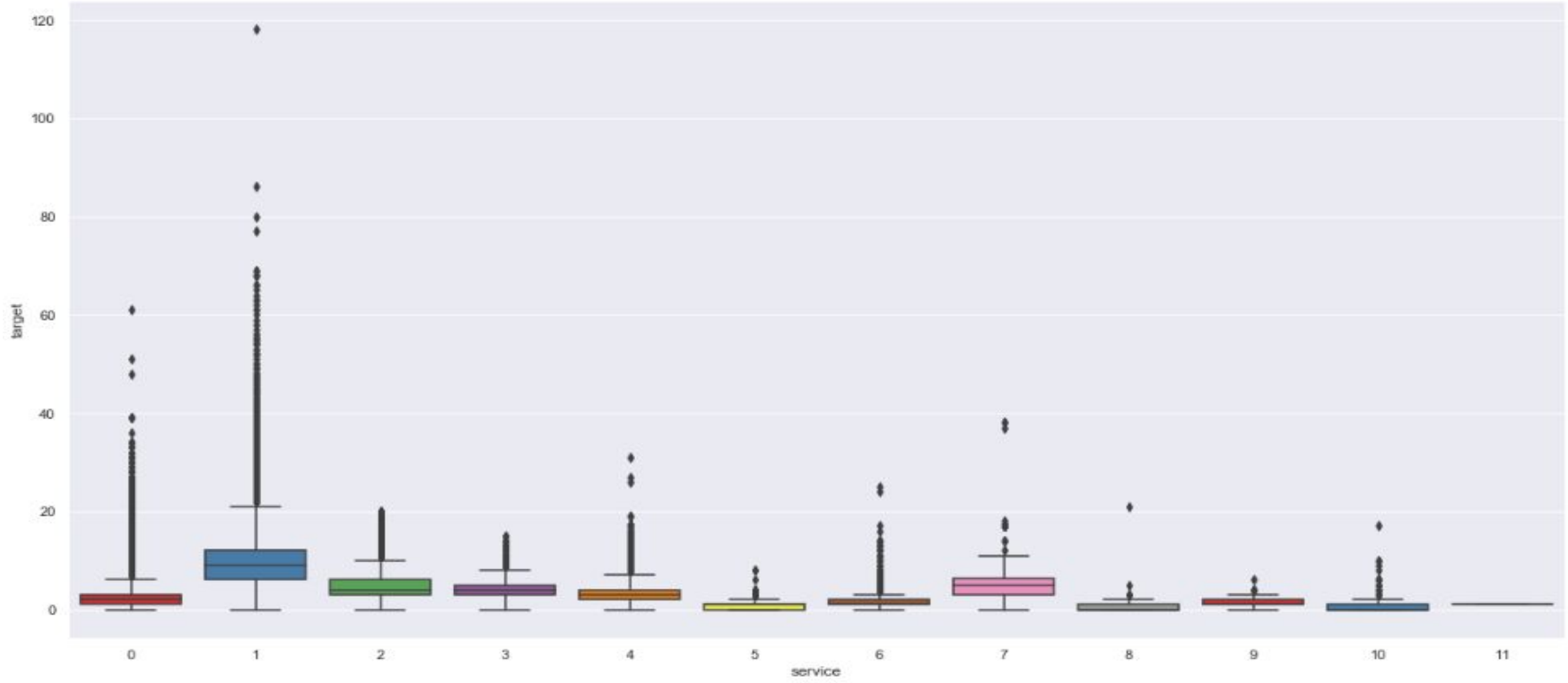
Cantidad de envíos por tipo de envío



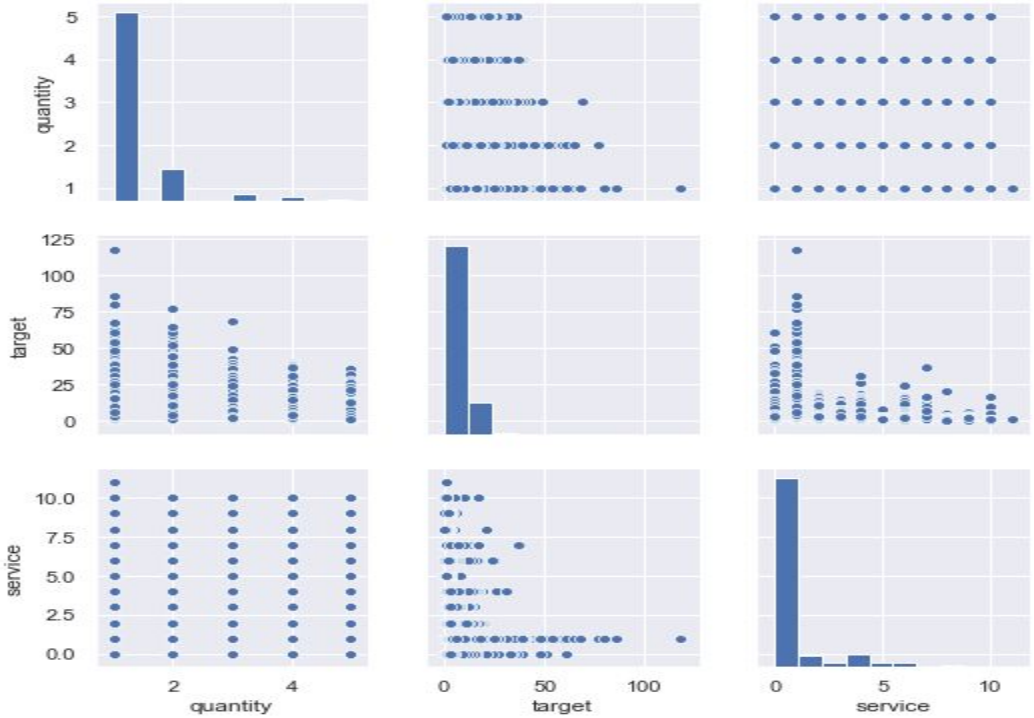
Distribución del target



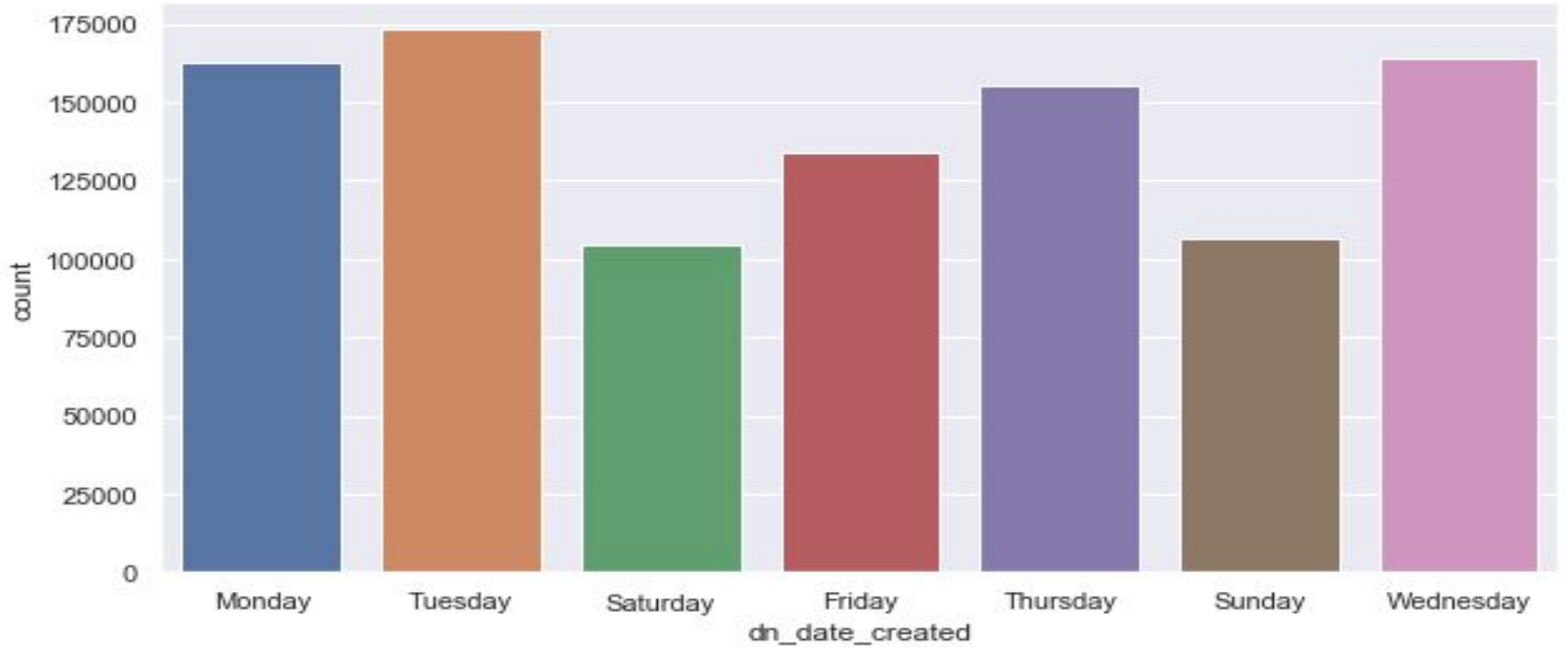
Outliers (target vs service)



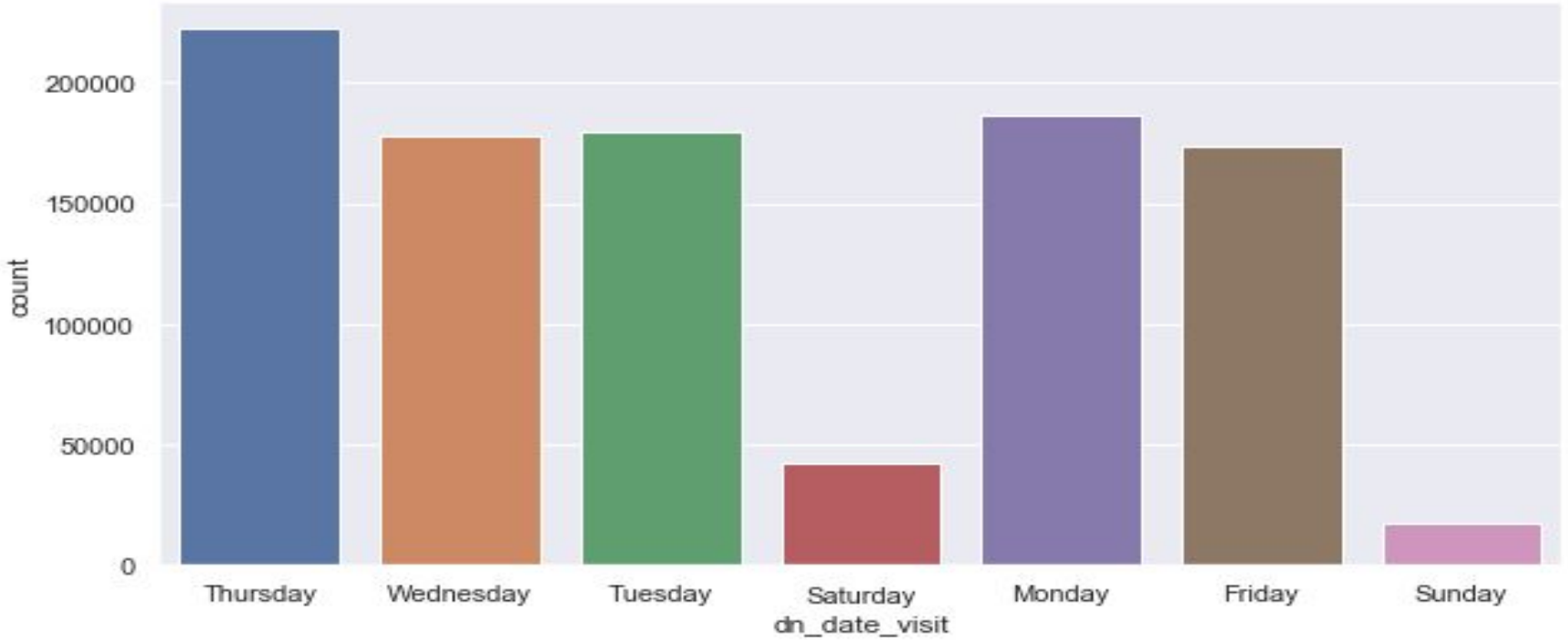
No existen variables correlacionadas



Fines de semana

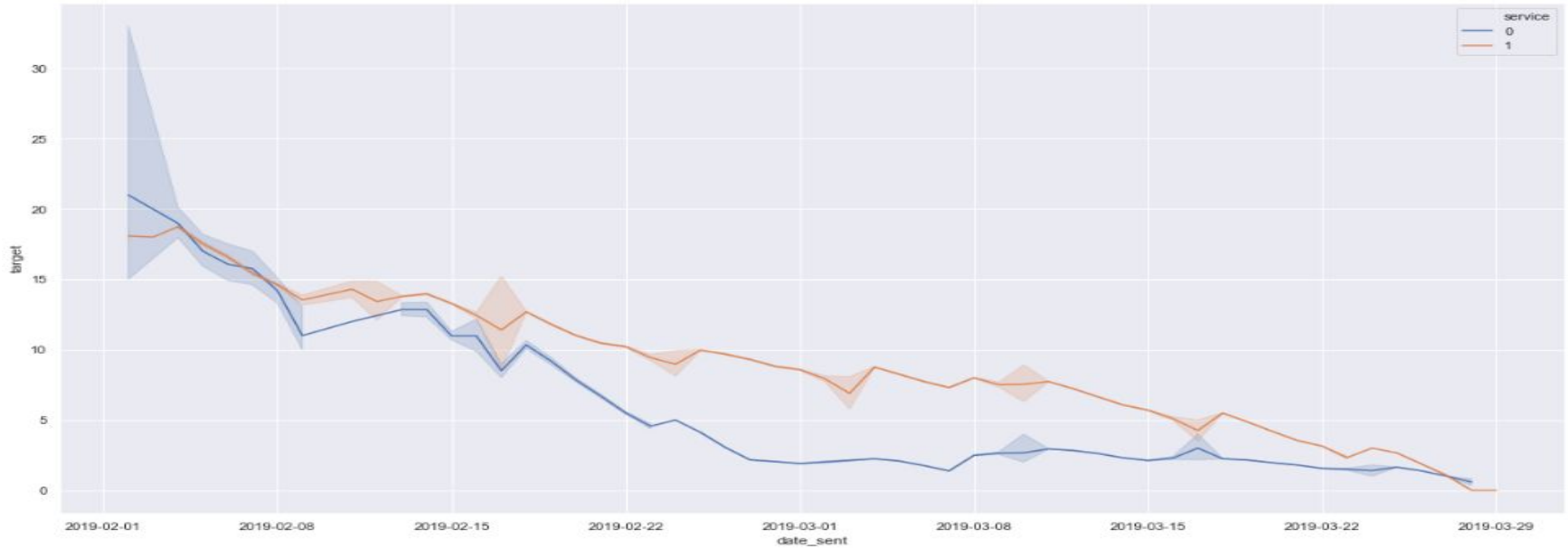


Paquetes que llegan por día

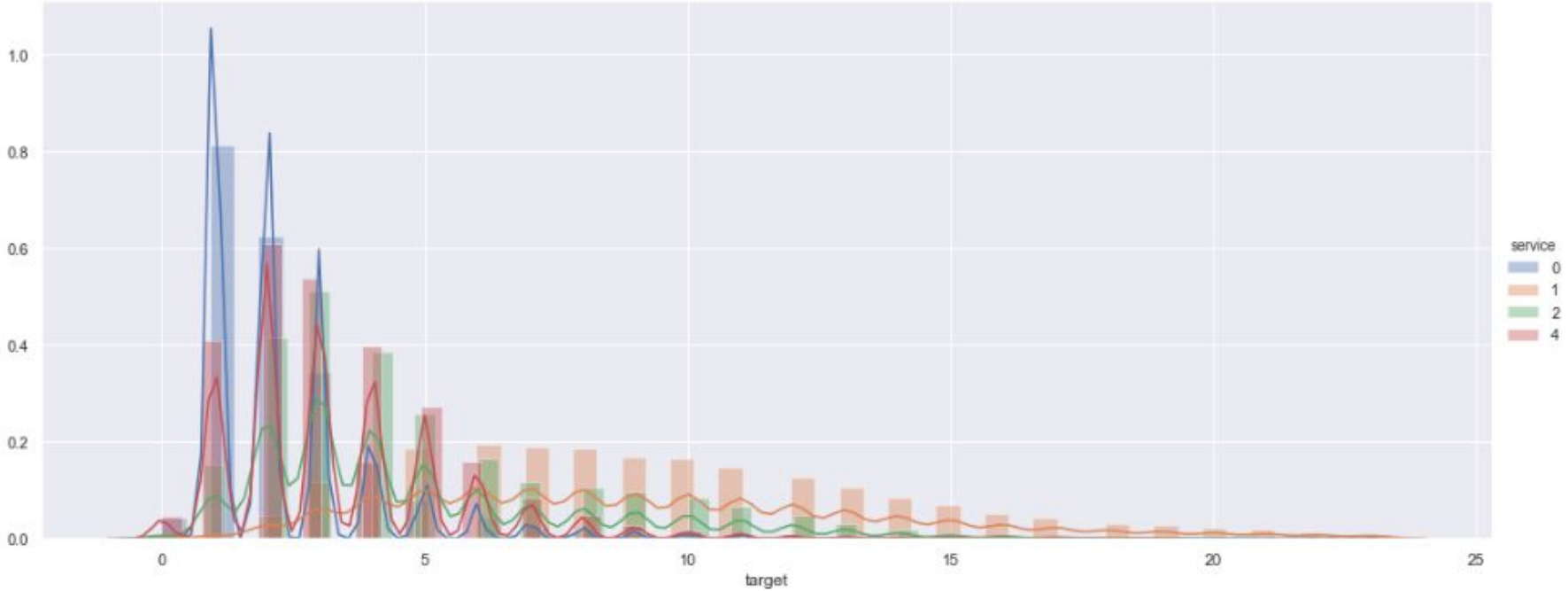


Distribuciones de target para los servicios 0 y 1

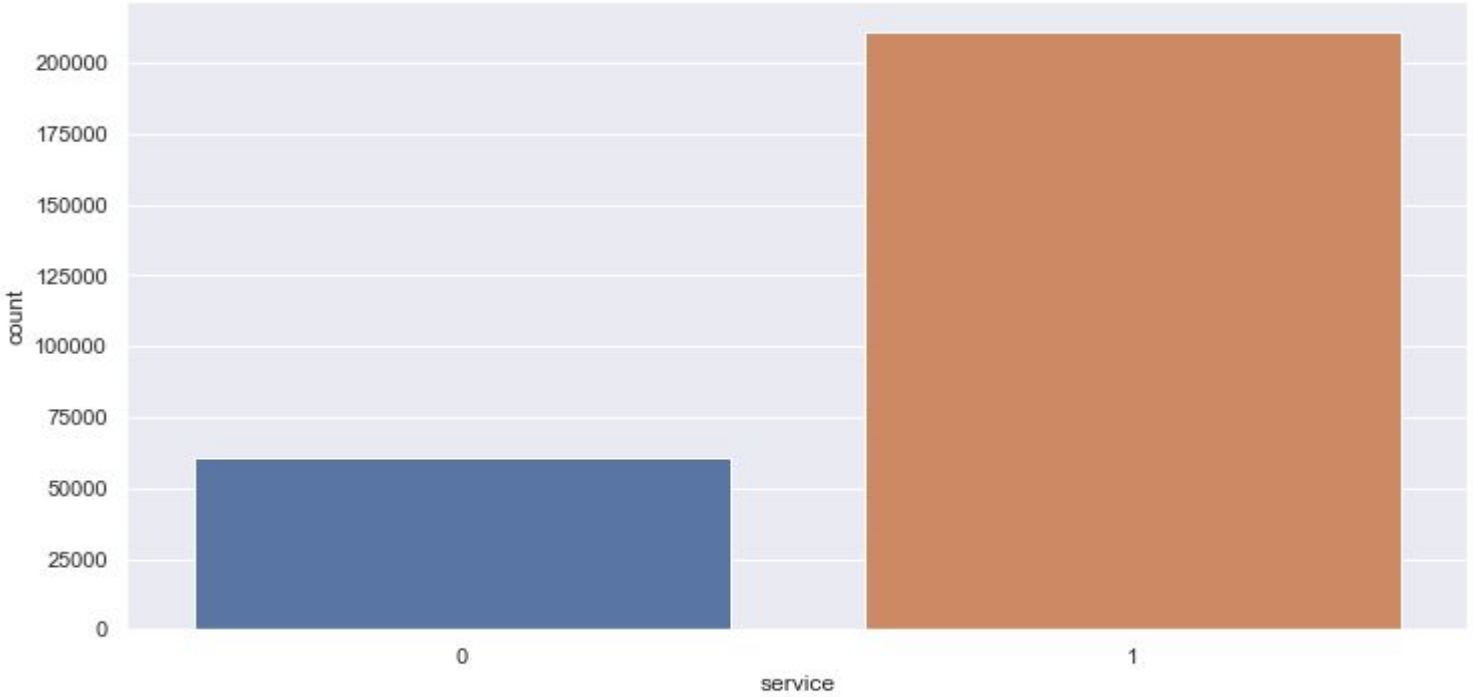
Analizamos como se comportan los servicios 0 y 1(concentran mas del 80% de los correos enviados) entre Febrero y Abril de 2019



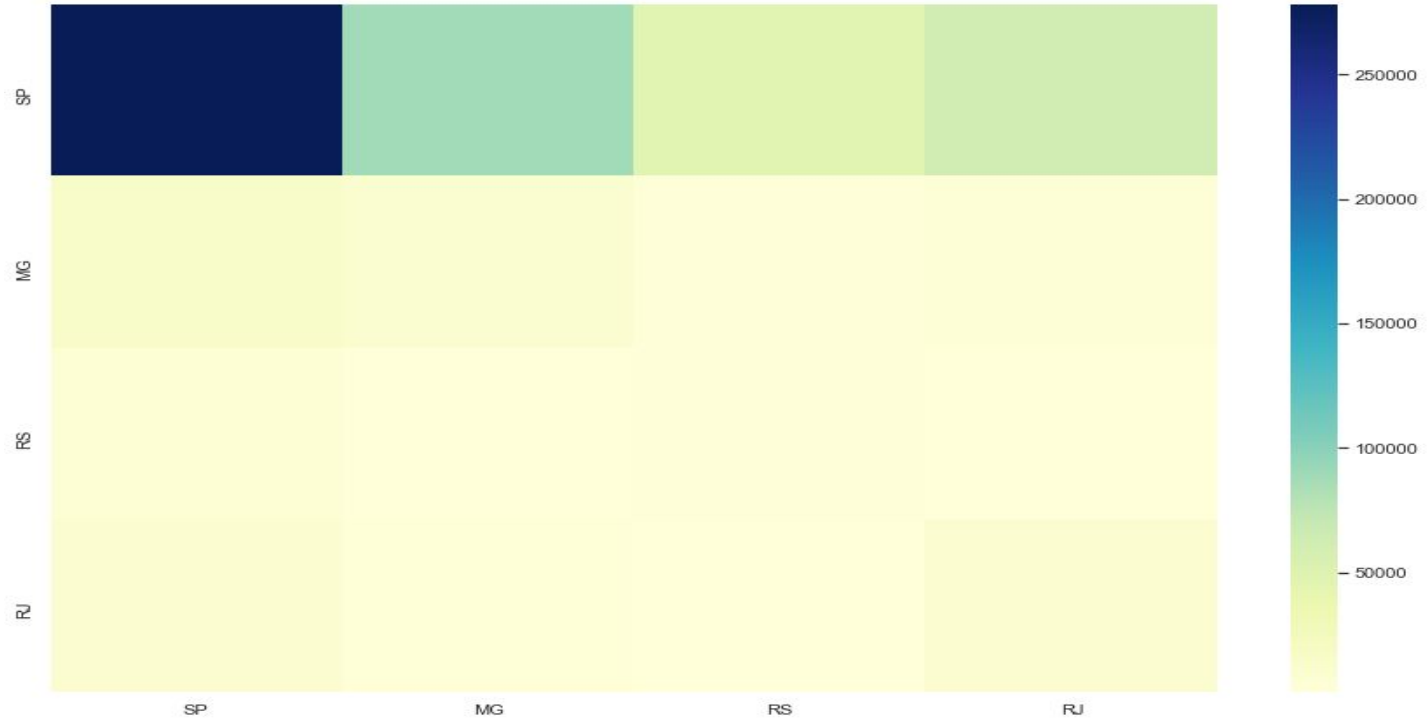
Distribuciones del target para 4 servicios:



Los servicios fuera de 'SP'



Heat-map de los zipcodes de vendedores y compradores



De interés

- Los registros de los correos durante Octubre, Noviembre y Diciembre pertenecen solamente a los servicios 0 y 1 que fueron los que enviaron aproximadamente el 80 % de los correos A partir de Enero de 2019 se registran correos enviados por otros servicios. En general los tiempos de demora para todos los servicios se fueron reduciendo desde Octubre de 2018 a Marzo de 2019.
- La cantidad de items por paquete no esta relacionada con el target.
- No pudimos observar correlacion entre variables cuantitativas
- Entre los distintos servicios existen características en común sería interesante poder analizar las posibles causas.

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

- Datos inconsistentes: vemos que los datos de cada feature pertenecen a su dominio.
  - Estados: analizamos si existe algún valor inusual
  - Datos de tipo datetime: filas con 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
- Aplicar normalización: Solo se pueden normalizar features numéricos.
- Reducción de dimensionalidad: A efectos de reducir su varianza, se puede acotar el rango de valores posibles que toma el feature utilizando PCA solo con features numéricos

Aplicamos todas las curaciones

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

Boxplot del target

Problemática

- Presentar la problemática que queremos resolver (reducida y completa)
- Mostrar cómo decidimos atacar esa problemática
- Mostrar qué restricciones tenemos por trabajar con series temporales..

INTRODUCCION AL APRENDIZAJE AUTOMATICO

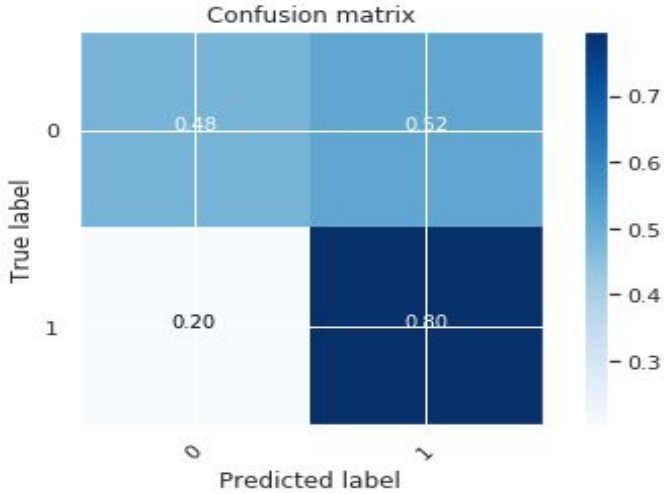
Diferenciamos entre envíos rápidos(si llega antes de 3 días hábiles) y lentos si llega después, esto reduce los problemas de clasificación a 2 clases.

- 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

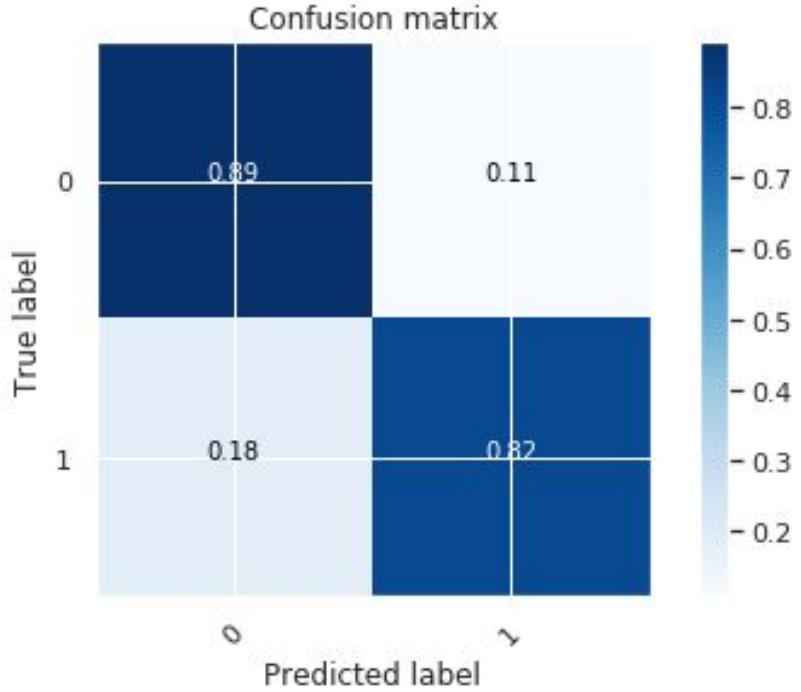
- Clusterizamos los envíos basados únicamente en las rutas utilizando KMeans

Matriz de Confusión

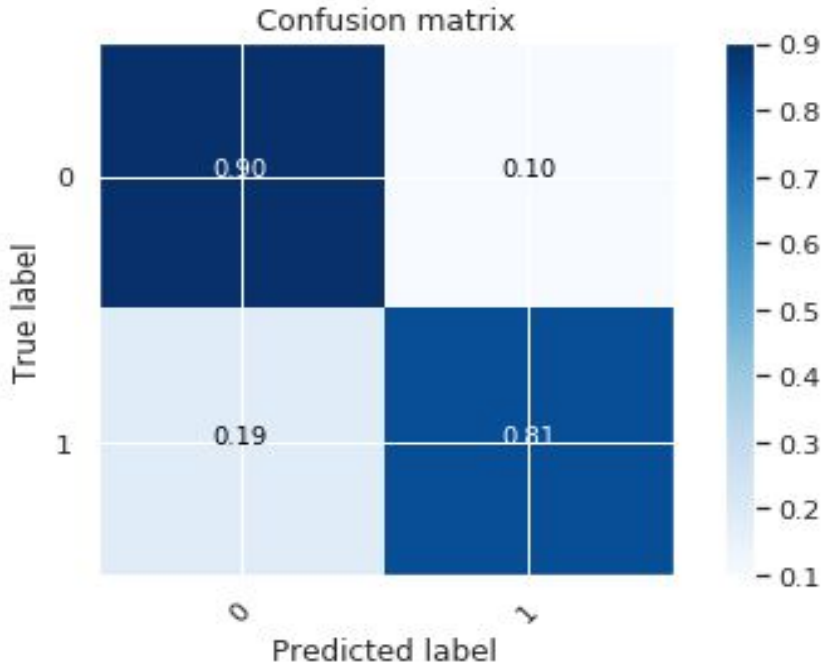


Utilizando el metodo Elbow vemos que a partir de 3 clusters las distancias no tienen un cambio significativo por lo tanto este es el numero optimo de clusters

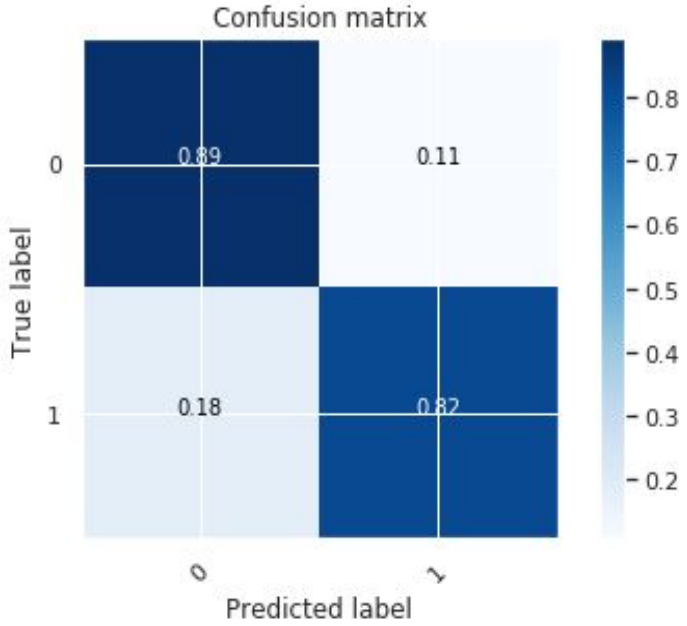
Modelos lineales: modelo basado en regresión lineal

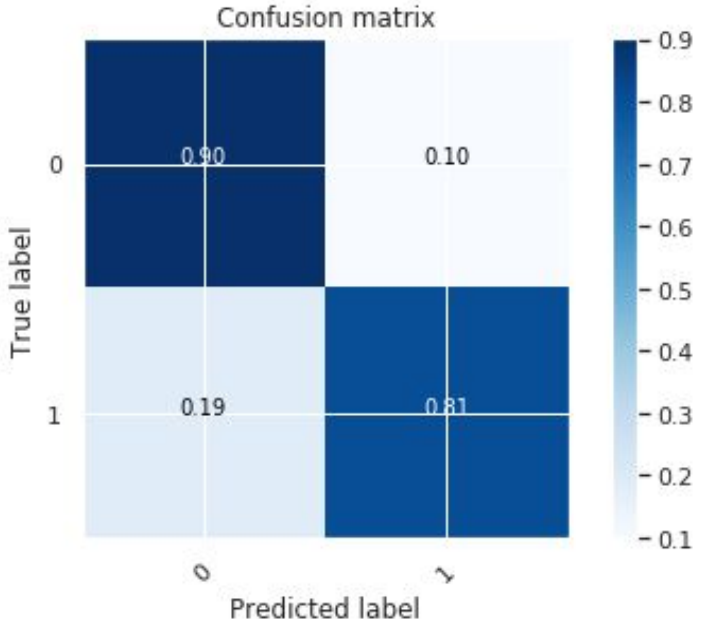


Modelo basado en regresión logística



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





El mejor método para clasificarlos fue LogisticRegression.

En el kmeans solo usamos las rutas y las metricas nos dieron 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 ademas otros features y nos dieron mejores metricas lo que podriamos interpretar como que quizas el kmeans andaría mejor con mas features

Aprendizaje supervisado y no supervisado

- Utilizamos el target tratando de predecir solo 21 clases para simplificar. Asignamos todo target mayor que 20 a 20
- Agregamos el concepto de ventana de predicción formada por dos componentes, un speed (cantidad de días hábiles predichos), más un offset (margen de error de la predicción). Las predicciones son de la forma: (speed, offset).
- Solo utilizamos los features de las rutas: zipcodes y service
- Diseñamos un pipeline con las siguientes transformaciones:
  - Recortamos el último dígito de los zip codes
  - Normalizamos los features
  - Proyectamos los features utilizando PCA, manteniendo 3 componentes.
  - Agregamos un clasificador

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.

Ventanas de predicción:

- Construimos un offset para mejorar las predicciones, de forma que tenga avg_offset menor o igual a 1

Viendo las probabilidades de cada etiqueta tomamos el maximo valor y si es menor al 1 por ciento le asignamos un offset de 3, si esta entre 1 y 10 por ciento le asignamos un offset de 2 y si esta entre 10 por ciento y 30 por ciento le asignamos un offset de 1 y finalmente, le asignamos un cero de offset al resto.

- Construimos un offset que mejore las métricas de los modelos y que tenga un avg_offset menor o igual que 2.5

Viendo las probabilidades de cada etiqueta tomamos el maximo valor y si es menor al 20 por ciento le asignamos un offset de 3, si esta entre 20 y 30 por ciento le asignamos un offset de 2 y si esta entre 30 por ciento y 40 por ciento le asignamos un offset de 1 y finalmente, le asignamos un cero de offset al resto

Próximos pasos

- Algoritmos clásicos de series temporales
- Redes neuronales
- Modelos bayesianos
- Embeddings

Gracias!!!