

Modelado de datos

1. Clasificamos choferes de acuerdo a ...
2. Clasificamos rutas de acuerdo a ...
3. Buscamos el match para la combinación Ruta+Chofer

Fuentes de datos

Fuentes de datos públicas / libres que podés consumir en Argentina

Aquí algunas fuentes útiles + qué tipo de dato podés obtener de cada una.

Ejemplos

- Dirección Nacional de Vialidad (DNV) – Argentina: publican datos de infraestructura vial, rutas nacionales, condiciones de la red vial. [Open Contracting Data Registry](#)
- Instituto Geográfico Nacional de la República Argentina (IGN): mapas base, geodatos de carreteras, pendientes, etc. [Open Data Index](#)
- Portal de datos abiertos de la Ciudad de Buenos Aires: por ejemplo, datos de transporte, tráfico, semáforos, estaciones de servicio. Ejemplos: datasets de semáforos (ubicaciones) [TUMI Datahub+2Redalyc+2](#)
- OpenStreetMap / descarga de geodatos de Argentina: para topografía, rutas, tipo de vía (urbana/interurbana), curvatura de la ruta. [Geo2Day](#)
- Portal “Datos Abiertos” nacional: Gobierno de la Nación Argentina y sus datos abiertos ([datos.gob.ar](#)) que indexan numerosos conjuntos. [Dateno+1](#)

Qué tipos de variables podés obtener de estas fuentes

- Tipo de vía, cantidad de carriles, pendientes, curvaturas → de OSM/IGN.
- Tráfico promedio, posibles congestiones, sensores urbanos → portales de movilidad urbana.
- Infraestructura (semáforos, estaciones de servicio, paradas) → portales locales.
- Accidentes históricos (puede que dependa del organismo provincial o nacional).

- Desvíos / paradas no programadas → quizás mediante telemetría privada o datos internos de tu compañía.

[Data Registry](#) | [Open Contracting Partnership](#)

[Argentina: Dirección Nacional de Vialidad | OCP Data Registry](#)

Download the OCDS data for Argentina: Dirección Nacional de Vialidad. Learn about the data source's coverage, features and quality issues.

Open Data Index

[Open Data Index - Open Knowledge](#)

The state of government open data in Argentina for National Map in 2015

[Welcome - TUMI Datahub](#)

[Traffic Lights - TUMI Datahub](#)

Traffic Lights This dataset falls under the category Individual Transport Traffic Control Systems. It contains the following data: Information on traffic lights in the city, with geographical...

[Map of and GIS data — download OSM vector layers](#)

[Map of Argentina and GIS data — download OSM vector layers](#)

Vector data of Argentina, maps and ready-to-go GIS projects in ESRI Shape, GeoPackage, Geodatabase, GeoJSON, PBF, CSV, TAB

[Dateno Registry and Dataset Search Engine](#)

A next-generation data search service provides fast, comprehensive access to open datasets worldwide, with powerful filters and an API-first architecture for seamless integration.

Fuentes de datos **no públicas / comerciales / internas**

Dado que querés una buena granularidad para “desvíos, paradas no programadas, consumo por km, historial de accidentes”, puede que necesites combinar con fuentes no completamente abiertas:

- Datos de telemetría de los vehículos (GPS, OBD) de tu flota o del proveedor (como el que ya tenés de Scania).
- Datos de aseguradoras / siniestros para “historial de accidentes” si es compatible.

- Datos de tráfico en tiempo real o históricos de empresas como TomTom, que tienen índice de congestión para Argentina. [TomTom](#)
 - Información de rutas específicas de carga o logística que ya tenga la empresa (por ejemplo, duración real vs planificada).
 - Datos privados de sensores instalados en la vía (contador de vehículos, sensores de velocidad) vía convenios con municipios.
-

Paso 1: Clasificación de rutas

Querés entender **qué tan exigente o riesgosa es cada ruta**.

Podés clasificarlas según tres dimensiones principales:

1. Peligrosidad

- Curvas cerradas, pendientes, tipo de terreno.
- Historial de accidentes o eventos bruscos (harsh braking, etc.).
- Tramos urbanos vs rurales, tráfico promedio.

2. Complejidad operativa

- Cantidad de paradas planificadas.
- Frecuencia de desvíos o desvíos históricos (routeDeviationMeters).
- Duración total y condiciones del recorrido (día/noche, clima, etc.).

3. Eficiencia logística

- Nivel de cumplimiento con el plan (plannedVsActualTime).
- Consumo promedio de combustible por km histórico en esa ruta.
- Densidad de tráfico o tiempo medio detenido.

 Ejemplo de categorías:

- Ruta sencilla / urbana / baja peligrosidad
- Ruta mixta / interurbana / media peligrosidad
- Ruta compleja / montaña / alta peligrosidad

Paso 2: Clasificación de choferes

Usás tus métricas de comportamiento (`DriverBehaviorFeatures`) para crear **perfiles de conducción**:

1. Nivel de seguridad

- Basado en frenadas bruscas, aceleraciones extremas, y exceso de velocidad.
- Podés generar un *Safety Score* del 0 al 100.

2. Eficiencia energética

- Basado en consumo por km, uso de control crucero e idle time.
- Choferes que logran mantener bajo consumo → eficientes.

3. Cumplimiento y confiabilidad

- Basado en desvíos, paradas no programadas y puntualidad.
- Choferes que siguen rutas y horarios → confiables.

4. Factor de ajuste

- Basado en cantidad de km's ya hechos

👉 Ejemplo de clasificación final:

- **Tipo A – Conductor Seguro y Eficiente**
- **Tipo B – Conductor Regular**
- **Tipo C – Conductor Riesgoso / Ineficiente**

También podrías usar un modelo de clustering (K-Means) para que el sistema encuentre automáticamente estos grupos.

⚙️ Paso 3: Match Ruta + Chofer

El objetivo es asignar el chofer **más adecuado para cada tipo de ruta**, maximizando seguridad y eficiencia.

Podés calcular un *Match Score* (por ejemplo, de 0 a 1) considerando:

1. Compatibilidad de riesgo

- Choferes seguros (baja tasa de eventos bruscos) para rutas de alta peligrosidad.
- Choferes más experimentados asignados a rutas complejas.

2. Eficiencia esperada

- Choferes con alto *efficiency score* para rutas largas o con mucho tráfico.
- Evitar choferes con alto *idleTimePercentage* en trayectos urbanos.

3. Historial de cumplimiento

- Choferes con buen *plannedVsActualTime* y baja desviación histórica para rutas con tiempos críticos.
- Penalizar choferes con desvíos frecuentes o paradas no planificadas.

👉 Fórmula simplificada:

$$\text{MatchScore} = w_1 \times (1 - | \text{Ruta.Peligrosidad} - \text{Chofer.Seguridad} |) + w_2 \times (1 - | \text{Ruta.EficienciaRequerida} - \text{Chofer.Eficiencia} |) + w_3 \times (1 - | \text{Ruta.Complejidad} - \text{Chofer.Confiabilidad} |)$$
$$\text{MatchScore} = w_1 \times (1 - | \text{Ruta.Peligrosidad} - \text{Chofer.Seguridad} |) + w_2 \times (1 - | \text{Ruta.EficienciaRequerida} -$$

$$\text{Chofer.Eficiencia} + w_3 \times (1 - |\text{Ruta.Complejidad} - \text{Chofer.Confiabilidad}|) \text{MatchScore} = w_1 \times (1 - |\text{Ruta.Peligrosidad} - \text{Chofer.Seguridad}|) + w_2 \times (1 - |\text{Ruta.EficienciaRequerida} - \text{Chofer.Eficiencia}|) + w_3 \times (1 - |\text{Ruta.Complejidad} - \text{Chofer.Confiabilidad}|)$$

Donde los **W** son pesos (por ejemplo: **0.5, 0.3, 0.2**).

Ruta	Peligrosidad	Complejidad	Eficiencia requerida	Chofer	Seguridad	Eficiencia	Confiabilidad	Match (%)
Ruta A	Alta	Alta	Media	Juan	Alta	Alta	Media	92%
Ruta B	Media	Media	Alta	Pedro	Media	Alta	Alta	88%
Ruta C	Baja	Baja	Alta	Carlos	Baja	Alta	Media	65%

Machete:

Data → ETL (clasificar datos en base a reglas) (como clasifico choferes y rutas?) → Tabla de datos clasificados (Dataset para entrenar)

Rutas

Categorización de datos

- Cantidad de destinos → Complejidad operativa del viaje (Numerico)
- Distancias totales → Complejidad operativa del viaje (Numerico)
- Historico de rechazos y derivaciones → Complejidad operativa del viaje (Numerico) (Cantidad de reachazos)
- ETA planificado y calculado (cumplimiento) → Eficiencia logística (Fórmula → Numerico) (real - planificado de todas las veces que se recorrió la ruta)
- Historicos de desvios (de kms sobre el planificado) → Eficiencia logística (Numerico) (Sumatoria de todos los desvios o normalizar con promedios)
- Historico de averías en ruta → Peligrosidad del viaje / Complejidad operativa del viaje (no candidato)
- --- 🙌 datubis de terubi ---

- Margen de ganancia → costo del viaje (nafta, etc) / cuanto cobramos el viaje → **Eficiencia logística** (Formula → Numerico)
- Factor de ruta (cantidad de autovia, ruta nacional, etc que tiene el viaje total) → **Complejidad operativa del viaje** (Formula → Numerico) (polinomial)
- Validar OSRM

1. `distancia_total_km` (Estático, OSRM)

****Descripción:****

Es la variable más básica. Mide la longitud total del tramo en kilómetros. Rutas más largas implican, por defecto, mayor consumo de combustible, más tiempo de exposición al riesgo y mayor desgaste del vehículo. *

****Cálculo:****

1. Al hacer la consulta al servicio `route` de OSRM, obtienes una respuesta JSON.

2. Extrae el valor de `routes[0].distance`.

3. ****Importante:**** OSRM devuelve esta distancia en ****metros****. Debes dividirla por 1000. *

****Fórmula:**** `(routes[0].distance) / 1000` ---

2. `velocidad_promedio_kmh` (Estático, OSRM) *

****Descripción:**** Esta es una variable crucial que actúa como un ****proxy del tipo de vía****. No mide la velocidad máxima, sino el promedio general.

* ****Velocidad alta** (ej. 90 km/h):**

Sugiere que la ruta es mayormente autopista o ruta nacional despejada (Baja complejidad de conducción).

* ****Velocidad baja** (ej. 40 km/h):**

Sugiere mucho tramo urbano, caminos de montaña, o rutas en mal estado (Alta complejidad). *

****Cálculo:****

1. De la misma respuesta de OSRM, obtienes `distance` (en metros) y `duration` (en segundos). 2. Debes convertir ambas para obtener km/h. * ****Fórmula:****

`(routes[0].distance / 1000) / (routes[0].duration / 3600)` --

-

3. `numero_pasos_navegacion` (Estático, OSRM) * **Descripción:** Mide la "complejidad de navegación". Una ruta de 300 km que es "todo recto" es muy simple (pocos pasos). Una ruta de 50 km que incluye 45 giros, rotondas y salidas es muy compleja para el conductor y aumenta la probabilidad de errores. * **Cálculo:**

1. Debes hacer la consulta a OSRM con el parámetro `steps=true` (ej.

`.../route/v1/driving/coords?steps=true`). 2. Esto te devolverá un *array* (lista) llamado `steps` dentro de `routes[0].legs[0]`. 3. Simplemente cuenta cuántos elementos hay en esa lista. * **Fórmula:**

`len(routes[0].legs[0].steps)` (en pseudocódigo/Python) ---

4. `indice_cumplimiento_promedio` (Histórico, BBDD) * **Descripción:**

Mide la **predictibilidad** de la ruta. Compara lo que *planificaste* contra lo que *realmente sucedió*. Un valor cercano a 1 es perfecto. Un valor de 1.5 significa que la ruta toma, en promedio, un 50% más de tiempo de lo esperado (alta complejidad, probablemente por tráfico, demoras, etc.). * **Cálculo:** 1. Desde tu BBDD, agrupa todos los viajes históricos por

`id_tramo`. 2. Calcula el tiempo promedio *real* y el tiempo promedio *planificado* para ese tramo. * **Fórmula (SQL):** $\text{AVG}(\text{tiempo_real_minutos}) /$

$\text{AVG}(\text{tiempo_planificado_minutos})$ ---

5. `variabilidad_tiempo_min` (Histórico, BBDD) * **Descripción:** Mide la

volatilidad o **consistencia** del tramo. Una ruta simple (ej. autopista sin tráfico) siempre tarda casi lo mismo (baja variabilidad). Una ruta compleja (ej. cruce de ciudad, paso fronterizo) tiene tiempos muy dispares (alta variabilidad). Esta es una métrica de complejidad excelente. *

Cálculo: 1. Agrupa todos los viajes históricos por `id_tramo`. 2. Calcula la **desviación estándar** (Standard Deviation) de los `tiempos_reales` de viaje. *

Fórmula (SQL): $\text{STDDEV}(\text{tiempo_real_minutos})$ ---

6. `tasa_averias_normalizada` (Histórico, BBDD) * **Descripción:** Mide el

riesgo físico o la "dureza" de la ruta sobre el vehículo. Normaliza la cantidad de averías por la cantidad de viajes, para que un tramo muy transitado no parezca injustamente más complejo. Una tasa alta sugiere mal estado del camino, topografía exigente o alto riesgo. * **Cálculo:**

1. Agrupa todos los viajes por `id_tramo`. 2. Cuenta el número total de viajes que tuvieron una avería y divídelo por el número total de viajes realizados en ese tramo. (Necesitarás una columna tipo booleano o 0/1 que indique `tuvo_averia`). * **Fórmula (SQL):**

$\text{SUM}(\text{columna_tuvo_averia}) * 1.0 / \text{COUNT}(\text{id_viaje_total})$ ---

7. `eficiencia_combustible_lt_km` (Histórico, BBDD) * **Descripción:** Es un proxy fantástico para el **esfuerzo del motor**. Un consumo alto (muchos litros por km) indica alta complejidad: * **Topografía:** Muchas subidas. * **Estado de la vía:** Camino de ripio o en mal estado (más resistencia). * **Tráfico:** Constante "frenar y acelerar" en tramos urbanos. * **Cálculo:** 1. Para cada viaje en tu BBDD, idealmente ya tienes `litros_consumidos` y `km_recorridos`. 2. Calcula el ratio `lt/km` para cada viaje. 3. Agrupa por `id_tramo` y calcula el promedio de ese ratio. * **Fórmula (SQL):** `AVG(litros_consumidos / km_recorridos)`

Choferes

- Cumplimiento de despacho → **Confiabilidad y cumplimiento**
- Historicos de desvios (de kms sobre el planificado) → **Eficiencia logística y energética (Numerico)** (Sumatoria de todos los desvios o normalizar con promedios)
- Historico de averías en ruta → Peligrosidad del viaje / Complejidad operativa del viaje (no candidato)
- Historico de paradas no permitidas (cantidad de eventos en los viajes) → **Confiabilidad y cumplimiento**
- Km conducidos (para nivelar los km que le son asignados a cada chofer, si A y B tienen puntaje 100 y 99 pero B tiene 200000km menos recorridos, corresponderia B, tiene que entrar tambien en el puntaje) → equiparar kilometros **Factor de ajuste**
- ETA planificado y calculado (cumplimiento) → **Eficiencia logística** (Fórmula → Numerico) (real - planificado)
- Ausencias (no programadas) → **Confiabilidad y cumplimiento (cantidad)** (Numerico)
- --- 🙌 datubis de terubi ---
- Historico de huella de carbono **Eficiencia logística y energética (Numerico)** (Sumatoria de todos los desvios o normalizar con promedios)
- Agresividad de conducción **Nivel de seguridad - Numérica**