2024/2025 1Q

Lista de Problemas 1

APA

Javier Béjar

Departament de Ciències de la Computació

Grau en Enginyeria Informàtica - UPC

UPC FIB Facultat d'Informàtica de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

DEPARTAMENT DE CIÉNCIES DE LA COMPUTACIÓ

FACULTAT D'INFORMÁTICA DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Primera edición, septiembre 2021

Esta edición, septiembre 2024

Instrucciones:

Para la entrega de grupo debéis elegir un problema del capítulo de problemas de grupo.

Para la entrega individual debéis elegir un problema del capítulo de problemas individuales.

Cada miembro del grupo debe elegir un problema individual diferente.

Debéis hacer la entrega subiendo la solución al racó.

Evaluación:

La nota de esta entrega se calculará como 1/3 de la nota del problema de grupo más 2/3 de la nota del problema individual.

Al realizar el informe correspondiente a los problemas explicad los resultados y las respuestas a las preguntas de la manera que os parezca necesaria. Se valorará más que uséis gráficas u otros elementos para ser más ilustrativos.

Podéis entregar los resultados como un notebook (Colab/Jupyter). Alternativamente, podéis hacer un documento explicando los resultados como un PDF y un archivo python con el código

También, si queréis, podéis poner las respuestas a las preguntas en el notebook, este os permite insertar texto en markdown y en latex.

Aseguraos de que los notebooks mantienen la solución que habéis obtenido, no los entreguéis sin ejecutar.

Objetivos de aprendizaje:

1. Hacer un mínimo análisis exploratorio de un conjunto de datos
2. Hacer el preproceso de un conjunto de datos para usar regresión
3. Saber plantear problemas de regresión sencillos y resolverlos usando diferentes métodos
4. Interpretar los resultados de un problema de regresión

CAPÍTULO 1

Problemas de Grupo

Al resolver el problema explicad bien lo que hacéis, no hacer ningún comentario o hacer comentarios superficiales tendrán una nota más baja.

Tenéis que mostrar que habéis entendido los métodos que estáis aplicando, así que un corta y pega de problemas similares no es suficiente.

1. **Predicción del uso de bicicletas**

   El uso compartido de bicicletas es un servicio proporcionado por cualquier ciudad importante del mundo, por lo que comprender y predecir el comportamiento del sistema es un elemento clave. Vamos a trabajar con el conjunto de datos de bicicletas compartidas del repositorio de conjuntos de datos de UCI que recopila estadísticas agregadas de uso de bicicletas junto con otra información adicional relevante. Se pueden descargar los datos desde aquí `https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset`.

   El objetivo de este problema es predecir cuántas bicicletas se usarán al día siguiente a partir de los datos de días anteriores (usaremos el archivo `day.csv`). Podéis leer en el `Readme.txt` los detalles sobre las variables.

   a) El primer paso es preprocesar y preparar los datos antes de ajustar cualquier modelo. Hay algunas variables que no son útiles para el problema o que no tiene sentido usar. Eliminadlas del conjunto de datos y explicad por qué las eliminái s.

   Necesitaremos obtener variables que nos permitan predecir a partir de la historia del sistema. Tal como están los datos no podemos hacer eso, por lo que necesitaremos un poco de preproceso. La librería `pandas` permite generar una copia de una tabla de datos desplazada una serie de instantes temporales usando el método `shift`. Mirad como funciona y generad una copia de los datos desplazada un día y añadidla como nuevas columnas (fijaos que os saldrán datos perdidos ¿por qué?). Si queremos predecir el futuro habrá una serie de variables que no podemos saber. Partid la tabla de datos en las columnas que usaremos para predecir y las que podríamos predecir a partir de las otras. Fijaos que hay variables del día actual que sabemos, como el día de la semana que es.

Dado que tenemos que predecir el futuro, no podemos partir los datos en entrenamiento y test tal como lo hacemos habitualmente (explicad por qué no podemos hacerlo). Vamos a seleccionar los primeros 500 ejemplos para entrenamiento y el resto para test.

Haced una exploración mínima del conjunto de datos de entrenamiento observando las relaciones entre las variables, especialmente con las variables objetivo. Describid las cosas que hayáis visto que os parezcan interesantes. Estandarizad los datos antes de entrenar los modelos.

b) Dado que hay varias variables que podemos predecir para el siguiente día vamos a escoger `temp`, `hum`, `windspeed` y `cnt`. Ajustad una regresión lineal a los datos y calculad la *calidad* del modelo empleando validación cruzada y con los datos de test. Para hacer la validación cruzada tenéis que usar el método `TimeSeriesSplit` con 5 particiones en el parámetro `cv` de la validación. ¿Por qué no es válida la validación cruzada tradicional? Comentad los resultados obtenidos con cada una de las variables. Calculad el error de validación cruzada y el del test con el *mean absolute error* (tendréis que mirar la documentación de *scikit-learn* para ver como se hace)

c) Probablemente la regularización ayude a obtener mejores resultados. Usad Ridge regression y LASSO para predecir cada variable. Tendréis que usar el mismo método para hacer la validación cruzada. Comentad los resultados obtenidos con cada una de las variables

d) Se nos podría ocurrir que dado que podemos predecir algunas variables para el día siguiente, esta nos podría valer como substituto e introducirla en el modelo para poder predecir mejor la variable `cnt`. Añadid las predicciones de la variable que mejor se predice a los datos y ajustad de nuevo la regresión lineal, la Ridge Regression y el LASSO. Explicad lo que sucede y por qué.

e) Para entender el modelo tenemos que analizar con detalle los resultados. Representad las predicciones del mejor modelo para el test contra los valores reales, analizad el gráfico de los residuos, explicad los resultados. Mirad los pesos que le asigna el LASSO a las variables ¿hay algunas que son descartadas? Eliminad las variables que no considera relevantes LASSO y ajustad de nuevo una regresión lineal. Comparad los resultados.

Ajustad una regresión lineal con todas las variables usando el método `OLS` de la librería `statsmodels` (los pesos serán los mismos) y analizad la significatividad que asigna el método al coeficiente de cada variable. Eliminad las variables que no tienen una significatividad menor que 0.05 y repetid la regresión. Comparad y comentad los resultados de los dos modelos reducidos.

f) Hemos asumido que sabiendo los datos del día anterior era suficiente para predecir los del día siguiente. Ajustad el modelo LASSO a los datos con dos y tres días antes. Comparad los resultados de los modelos entre ellos y con el que solo usa el día anterior. Mirad las variables que descarta LASSO y explicad lo que observéis.

2. **Aire limpio, aire puro**

La contaminación del aire es algo serio en las grandes ciudades como Barcelona, el poder relacionarlo con otras variables puede ayudar a comprender mejor sus fuentes y las circunstancias que le afectan.

El ayuntamiento de Barcelona recolecta diversos datos sobre la ciudad en su portal de datos abiertos[1]. Vamos a trabajar con un extracto de esos datos para los años 2022-2024, eligiendo un subconjunto de variables que son medidas diarias que representan diferentes características que tienen alguna relación con la contaminación como el número de matriculaciones de vehículos, el

[1] https://portaldades.ajuntament.barcelona.cat/

número de personas que llegan a Barcelona desde ciudades cercanas, el precio de la electricidad, el volumen de tránsito e información meteorológica medida en diferentes puntos de la ciudad (temperatura, viento, precipitación). El objetivo es buscar la relación con la cantidad de dióxido de nitrógeno (NO2) medido en l'Eixample.

Podéis obtener estos datos mediante la función `load_BCN_NO2` de la librería `apafib`. Resolved los siguientes apartados ilustrando los resultados de la manera que os parezca más adecuada.

a) Dividid el conjunto de datos en entrenamiento y test (60 %/40 %). Haced una exploración mínima del conjunto de datos de entrenamiento observando las relaciones entre las variables, especialmente con la variable objetivo. Describid las cosas que hayáis visto que os parezcan interesantes. Transformad las variables adecuadamente para poder ajustar un modelo de regresión tanto el conjunto de entrenamiento como el de test.

b) Aplicad algún método de reducción de dimensionalidad a los datos de entrenamiento y comentad lo que se pueda apreciar en la visualización. Pensad en qué podéis representar sobre la transformación.

c) Ajustad una regresión lineal, una regresión Ridge y una regresión LASSO a los datos ¿Os parece suficientemente bueno el resultado? Representad los valores de la variable objetivo para el conjunto de test contra las predicciones y representad los residuos. ¿Qué modelo os parece mejor?

d) Habréis visto que las variables meteorológicas se toman en diferentes puntos de la ciudad, pero podéis comprobar que las medidas son bastante parecidas (comprobadlo y mostrad que es así adecuadamente). ¿Es posible reducir el número de observatorios a solo uno? ¿Cuál es el mejor? Comprobadlo con el modelo que haya dado el mejor resultado.

e) Otra manera de ver la relevancia de las variables en el modelo es comprobar la significatividad de los coeficientes de regresión. Ajustad una regresión lineal con todas las variables usando el método `OLS` de la librería `statsmodels` (los pesos serán los mismos que en la regresión lineal) y analizad la significatividad que asigna el método al coeficiente de cada variable. Explicad lo que habéis visto ¿hay alguna posible razón para que algunas variables no sean importantes? Eliminad las variables que no son significativas y ajustad de nuevo el mejor modelo. Comentad los resultados.

f) Si representáis las predicciones del mejor modelo contra los valores reales probablemente veréis que no todas las predicciones son homogéneas. Eliminad todas las variables que el modelo `OLS` considera no significativas y usad la función `PolynomialFeatures` de scikit-learn para añadir características al conjunto de datos que correspondan a polinomios de grado 2 para esas variables (usad *interaction_only=True* para que solo se tengan en cuenta las interacciones entre las variables). Ajustad de nuevo una regresión lineal y un modelo LASSO para estas variables y evalúad la calidad de los modelos. Representad las predicciones respecto a los valores reales y comentad lo que observáis.

3. **El precio de las cosas y más**

Asumimos que los precios de las cosas están ligados entre ellos de manera más o menos compleja, eso mueve todo lo que está relacionado con la economía, como por ejemplo la bolsa. A veces hay cosas que pueden estar relacionadas también de manera más o menos coherente y permite descubrir factores desconocidos pueden sorprendernos (o ser simplemente espurios).

El ayuntamiento de Barcelona recolecta diversos datos sobre la ciudad en su portal de datos abiertos[1]. Vamos a trabajar con un extracto de esos datos para los años 2022-2023, eligiendo un subconjunto de variables que tienen que ver con la economía, como el precio de productos de la cesta de la compra o el número de matriculaciones de vehículos y, por lo tanto, tendrán

alguna relación con el índice IBEX. De manera exploratoria añadiremos una serie de variables que no parecen tener relación y que si es el caso no tendrán un peso en el modelo, como son la temperatura medida en diferentes puntos de la ciudad y el nivel de ruido a diferentes horas del día.

Podéis obtener estos datos mediante la función `load_BCN_precios` de la librería `apafib`. Resolved los siguientes apartados ilustrando los resultados de la manera que os parezca más adecuada.

a) Dividid el conjunto de datos en entrenamiento y test (60 %/40 %). Haced una exploración mínima del conjunto de datos de entrenamiento observando las relaciones entre las variables, especialmente con la variable objetivo. Describid las cosas que hayáis visto que os parezcan interesantes. Transformad las variables adecuadamente para poder ajustar un modelo de regresión tanto el conjunto de entrenamiento como el de test

b) Aplicad algún método de reducción de dimensionalidad a los datos de entrenamiento y comentad lo que se pueda apreciar en la visualización. Pensad en qué podéis representar sobre la transformación.

c) Ajustad una regresión lineal, una regresión Ridge y una regresión LASSO a los datos ¿Os parece suficientemente bueno el resultado? Representad los valores de la variable objetivo para el conjunto de test contra las predicciones y representad los residuos. ¿Qué modelo os parece mejor?

d) Comprobad si el modelo LASSO identifica las variables espurias como no significativas. Eliminadlas del modelo y volved a ajustar el mejor modelo que os ha salido ¿Cuál es el mejor? Comentad los resultados.

e) Otra manera de ver la relevancia de las variables en el modelo es comprobar la significatividad de los coeficientes de regresión. Ajustad una regresión lineal con todas las variables usando el método `OLS` de la librería `statsmodels` (los pesos serán los mismos que en la regresión lineal) y analizad la significatividad que asigna el método al coeficiente de cada variable. Explicad lo que habéis visto ¿hay alguna posible razón para que algunas variables no sean importantes? Eliminad las variables que no son significativas y ajustad de nuevo el mejor modelo. Comentad los resultados.

f) A veces las interacciones entre las variables son importantes para obtener un mejor modelo. Partid del conjunto de datos del que habéis quitado las variables no significativas y usad la función `PolynomialFeatures` de scikit-learn para añadir características al conjunto de datos que correspondan a polinomios de grado 2 para esas variables. Ajustad de nuevo una regresión lineal y un modelo Ridge para estas variables y evaluad la calidad de los modelos. Representad las predicciones respecto a los valores reales y comentad lo que observáis.

CAPÍTULO 2

Problemas Individuales

Al resolver el problema explicad bien lo que hacéis, no hacer ningún comentario o hacer comentarios superficiales tendrán una nota más baja.

Tenéis que mostrar que habéis entendido los métodos que estáis aplicando, así que un corta y pega de problemas similares no es suficiente.

Para obtener los datos de algunos de estos problemas necesitaréis instalaros la última versión de la librería `apafib`. La podéis instalar localmente haciendo:

```
pip install --user --upgrade apafib
```

Para usar las funciones de carga de datos solo tenéis que añadir su importación desde la librería, en vuestro script o notebook, por ejemplo

```
from apafib import load_stroke
```

La función por lo general os retornará un `DataFrame` de Pandas con los datos. Si no es así el enunciado explicará que retorna.

Otros problemas necesitarán la librería `ucimlrepo`, la podéis instalar de la misma forma. Esta librería carga los datos usando el método `fetch_ucirepo` que recibe un parámetro `id` que es el identificador del conjunto de datos en el repositorio de datos de UCI.

1. **Frío, Frío, Caliente, Caliente**

   El cambio climático obliga a tener en cuenta las características de las viviendas para optimizar la energía necesaria para enfriarlas o calentarlas. El conjunto de datos *Energy Efficiency*[1] tiene los cálculos de cuanta energía es necesaria para enfriar/calentar un edificio a partir de un conjunto de

[1] Tenéis información sobre sus atributos en https://archive.ics.uci.edu/dataset/242/energy+efficiency.

características. Estos valores se han calculado mediante simulaciones que son más costosas de calcular que un modelo de aprendizaje entrenado con datos. El objetivo es evaluar si un modelo de regresión es suficiente para obtener una buena estimación.

Los datos se pueden cargar usando la librería `ucimlrepo` mediante el método `fetch_ucirepo` usando 242 como valor del parámetro `id`. Esto retorna una estructura desde la que se puede acceder a los datos y sus características.

a) Divide el conjunto de datos en entrenamiento y test (60 %/40 %). Haz una exploración mínima del conjunto de datos de entrenamiento observando las relaciones entre las variables, especialmente con las variables objetivo. Describe las cosas que hayas visto que te parezcan interesantes. Transforma las variables adecuadamente para poder ajustar un modelo de regresión, tanto el conjunto de entrenamiento como el de test. Usaremos la variable `Heating Load` como variable objetivo para ajustar los modelos.

b) Aplica Análisis de Componentes Principales (PCA) al conjunto de entrenamiento, analiza sus resultados. Visualiza los datos usando las dos primeras componentes representando la variable objetivo. ¿Crees que puede haber una relación entre las variables del conjunto de datos y esa variable objetivo? ¿Por qué? Observa el número de componentes necesarios para obtener el 100 % de la variancia de los datos ¿observaste alguna cosa en la matriz de correlación de las variables independientes que permita explicar esto?

c) Ajusta un modelo de regresión lineal, Ridge y LASSO a los datos para predecir la variable objetivo. ¿Te parece suficientemente bueno el resultado? ¿Por qué? Analiza los pesos de los coeficientes de los modelos. Compara las predicciones de los modelos contra los valores reales y explica lo que observes.

d) El conjunto de datos ha sido creado artificialmente y no puede dar una idea de como se comportaría un modelo en una situación real. Algo que sucede frecuentemente es tener valores perdidos en el conjunto de datos. Implementa una función que substituya aleatoriamente un porcentaje de los datos por valores perdidos (`np.nan`). Haz un estudio de la calidad de un modelo LASSO cuando hay un porcentaje de datos perdidos del 10 %, 25 % y 50 % imputando los valores perdidos usando el método `KNNImputer` de la librería `scikit learn` usando 5 vecinos. Deberás añadir los valores perdidos al conjunto de datos y luego partirlo en entrenamiento y test para simular un caso real en el que los datos nuevos también tienen valores perdidos. Repite el experimento para cada porcentaje al menos 10 veces. Compara los resultados y explica que has obtenido.

2. **You give me fever**

Al diseñar dispositivos electrónicos de medida se pueden usar multitud de sensores que pueden aumentar la precisión, pero eso los hace más caros. Una posibilidad es analizar la necesidad de cada sensor obteniendo un modelo que permita ver la influencia en la medida que nos interesa y determinar qué sensores son realmente importantes. El conjunto de datos *Infrared Thermography Temperature*[2] tiene un conjunto de variables que se usan para determinar de dos modos la temperatura de pacientes. El objetivo es estudiar qué sensores realmente son necesarios para obtener estos valores.

Los datos se pueden cargar usando la librería `ucimlrepo` mediante el método `fetch_ucirepo` usando 925 como valor del parámetro `id`. Esto retorna una estructura desde la que se puede acceder a los datos y sus características.

[2]Tenéis información sobre sus atributos en https://archive.ics.uci.edu/dataset/925/infrared+thermography+temperature+dataset

a) Haremos primero una limpieza de los datos antes de partirlos en entrenamiento y test. Verás que hay una variable que tiene datos perdidos y valores extremos. Encuentra la variable analizando/visualizando el conjunto de datos y elimina los ejemplos problemáticos. Verás que hay varias variables categóricas, transfórmalas adecuadamente y comenta cualquier cosa que parezca fuera de lo usual en sus valores.

Divide el conjunto de datos en entrenamiento y test (70 %/30 %). Usaremos como variable objetivo `aveOralF`. Haz una exploración mínima del conjunto de datos de entrenamiento observando las relaciones entre las variables, especialmente con la variable objetivo. Describe las cosas que hayas visto que te parezcan interesantes. Transforma las variables adecuadamente para poder ajustar un modelo de regresión tanto el conjunto de entrenamiento como el de test.

b) Ajusta un modelo de regresión lineal, Ridge y LASSO para predecir la variable objetivo y estima la calidad de la regresión. ¿Te parece suficientemente bueno el resultado? Representa los residuos y comenta que aparece. Representa las predicciones respecto a los valores reales y comenta lo que se observa. Analiza qué variables usa LASSO para hacer las predicciones.

c) Selecciona del conjunto de datos las variables a las que LASSO les asigna un peso. Usa la función `PolynomialFeatures` de scikit-learn para añadir características al conjunto de datos que correspondan a polinomios de grado 2 para esas variables (usa *interaction_only=True* para que solo se tengan en cuenta las interacciones entre las variables). Ajusta de nuevo una regresión lineal y un modelo LASSO para estas variables y evalúa la calidad del modelo. Representa los residuos y comenta que aparece. Representa las predicciones respecto a los valores reales y los que predecía el mejor modelo del apartado anterior. Comenta lo que se observa.

d) La regresión cuantil[3] es un modelo que permite estimar el intervalo de predicción de una regresión. La regresión se realiza para que las predicciones estén por debajo de un cuantil de probabilidad determinado. Haciendo la regresión a diferentes cuantiles podemos tener el intervalo de valores posibles para una predicción. El cuantil 0.5 corresponde a la media de la predicción de la regresión. Ajusta una regresión cuantil para los cuantiles 0.1, 0.5 y 0.9 para los datos del apartado anterior. Tendrás que ajustar el peso de la regularización de esta regresión para obtener el mejor modelo. Selecciona un subconjunto al azar de ejemplos y represéntalos contra los valores reales la predicción de los tres modelos (tres gráficas una encima de la otra). Comenta los resultados.

3. **Blowing in the wind**

Una forma sencilla de predecir series temporales es utilizar regresión lineal sobre un número de instantes temporales de la serie. Estaremos prediciendo el futuro en función de las observaciones pasadas en una ventana de tiempo, de manera que:

$$f(x_t) = c + \left[\sum_{i=1}^{p} w_{t-i} \cdot x_{t-i}\right] + \epsilon_t$$

Donde c es una constante y ϵ_t es ruido gausiano. Este modelo es denominado auto regresivo (AR).

El conjunto de datos `Wind Speed Prediction Dataset`[4] tiene mediciones de diferentes variables tomadas por una estación meteorológica durante 15 años. El objetivo es predecir el valor de la variable `WIND` usando ventanas de datos pasados.

[3] Está implementada en `scikit learn` como `QuantileRegressor`.

[4] Tenéis información sobre sus atributos en https://www.kaggle.com/datasets/fedesoriano/wind-speed-prediction-dataset.

Trabajaremos con una versión de este conjunto que podéis obtener mediante la función `load_wind_prediction` de la librería `apafib`. Resuelve los siguientes apartados ilustrando los resultados de la manera que te parezca más adecuada.

a) Primero haremos una limpieza de los datos. Verás que existen datos perdidos en alguna de las variables. Dado que son series temporales una manera sencilla de hacer la imputación es usar el valor del último instante válido de la serie. En este caso puedes usar la función `fillna` de Pandas usando el parámetro `method='ffill'`. Hay tres variables `IND` que no están descritas en el conjunto de datos, no sabiendo lo que significan lo mejor es eliminarlas. Tienes también la variable `DATE`, que de por sí no tiene mucha utilidad, pero podemos pensar que saber el mes del año podría ser útil para predecir el viento. Averigua como transformar con Pandas esa variable en formato `datetime` y como extraer el mes. Una vez obtenido ese valor puedes deshacerte de la variable.

Para generar los datos necesitaremos ventanas temporales de cierta longitud. `Pandas` permite generar una copia de una tabla de datos desplazada una serie de instantes temporales usando el método `shift`. Genera tres conjuntos de datos con longitud de ventana 2, 4 y 6 de manera que puedas predecir el viento de un dia a partir de las variables del día anterior, tres días y cinco días. Tendrás que descartar todas las columnas que corresponden al último instante temporal de manera que quede solo la variable `WIND` que es la que has de predecir. Elimina todos los valores perdidos que ha generado esta transformación.

La validación en series temporales no puede usar validación cruzada. Tendrás que generar un conjunto de entrenamiento, uno de validación y uno de test para evaluar los modelos. Parte el conjunto de datos de manera que tengas un 70 % de entrenamiento, 15 % de validación y 15 % de test. Elimina las últimas filas del conjunto de entrenamiento y validación para que no haya instantes temporales compartidos entre las particiones.

b) La calidad de un modelo en series temporales se puede medir de diferentes maneras. Para este caso, usa el error absoluto medio (MAE). Esto tiene la ventaja de que el error esta en las unidades de la variable respuesta, en este caso *m/s*. Entrena regresiones lineales, Ridge y LASSO con los diferentes conjuntos de datos y compara su calidad y las características de los modelos. El ajuste de hiperparámetros no lo puedes hacer mediante validación cruzada, has de utilizar el conjunto de datos de validación. Selecciona el mejor modelo adecuadamente. Analiza los pesos de la regresión LASSO ¿es el mes relevante para la predicción? ¿Cómo se comportan los pesos respecto a la longitud de la ventana? Representa el qqplot de los residuos del mejor modelo y comprueba si son gausianos. Comenta los resultados.

c) Al predecir el viento nos interesa saber la incertidumbre de la predicción. La regresión cuantil[5] es un modelo que permite estimar el intervalo de predicción de una regresión. La regresión se realiza para que las predicciones estén por debajo de un cuantil de probabilidad determinado. Haciendo la regresión a diferentes cuantiles podemos tener el intervalo de valores posibles para una predicción. El cuantil 0.5 corresponde a la media de la predicción de la regresión. Ajusta una regresión cuantil para los cuantiles 0.1, 0.5 y 0.9. Tendrás que ajustar el peso de la regularización de esta regresión para obtener el mejor modelo usando la muestra de validación. Selecciona los primeros y últimos 100 instantes de la serie de test y representa las predicciones de los diferentes cuantiles. Calcula la media y varianza de la diferencia entre la predicción del cuantil 0.1 y 0.9 para estos dos intervalos para ver si hay una diferencia. Comenta los resultados.

Representa la predicción del mejor modelo para una pequeña ventana ($\approx$ 100) de datos de la muestra de test. ¿Crees que la regresión está haciendo una buena aproximación de la

[5] Está implementada en `scikit learn` como `QuantileRegressor`.

serie temporal? ¿Qué características debería cumplir esta serie para que la regresión lineal fuera un buen modelo para predecirla?

4. **Visitad los museos de Barcelona**

   Barcelona es una ciudad llena de museos, el predecir la afluencia de visitantes puede ser una buena manera de planificar su gestión. El ayuntamiento de Barcelona recolecta diversos datos sobre la ciudad en su portal de datos abiertos[6]. Vamos a trabajar con un extracto de esos datos para los años 2022-2024, eligiendo un subconjunto de variables que son medias o acumulados que representan diferentes sectores, como los barcos que hacen escala en la ciudad, los destinos de los aviones del Prat, la llegada de personas desde diferentes procedencias o la temperatura de la ciudad en diferentes puntos. También tenemos el número de entradas vendidas en los museos de Barcelona en la semana y queremos predecir las entradas de la semana que viene (`EntradasMuseos-Museos+1`)

   Puedes obtener estos datos mediante la función `load_BCN_Museos` de la librería `apafib`. Resuelve los siguientes apartados ilustrando los resultados de la manera que te parezca más adecuada.

   a) Divide el conjunto de datos en entrenamiento y test (60 %/40 %). Haz una exploración mínima del conjunto de datos de entrenamiento observando las relaciones entre las variables, especialmente con la variable objetivo. Describe las cosas que hayas visto que te parezcan interesantes. Transforma las variables adecuadamente para poder ajustar un modelo de regresión tanto el conjunto de entrenamiento como el de test.

   b) Ajusta una regresión lineal, una regresión Ridge y una regresión LASSO a los datos ¿Te parece suficientemente bueno el resultado? Representa los valores de la variable objetivo para el conjunto de test contra las predicciones y representa los residuos. ¿Qué modelo te parece mejor?

   c) El predecir directamente el número de entradas puede ser complicado al ser un valor de gran magnitud. Transforma la variable utilizando el logaritmo y ajusta de nuevo los modelos. Representa los valores de la variable objetivo para el conjunto de test contra las predicciones y representa los residuos. ¿Qué modelo de entre todos te parece mejor? Elige otras variables que correspondan a una magnitud grande y transfórmalas también usando el logaritmo. Ajusta el mejor modelo que tienes y comprueba si hay alguna mejora.

   d) Hay veces que algunos ejemplos tienen mucha influencia en la regresión. La regresión de Huber (`HuberRegressor`) permite aplicar un peso que reduce la influencia de los ejemplos más alejados. Este modelo tiene además una regularización como LASSO. Ajusta adecuadamente estos dos hiperparámetros, aplicando este modelo de regresión a la transformación de los datos que mejor resultado haya obtenido. Comenta los resultados.

5. **Calentando el ambiente**

   En el mundo de los datos aparecen correlaciones espurias de vez en cuando que ocultan relaciones con terceras variables que desconocemos. Por ejemplo, la venta diaria de helados está correlacionada con el número de ahogamientos en piscinas. Obviamente, dejar de vender helados no salvará vidas. El ayuntamiento de Barcelona recolecta diversos datos sobre la ciudad en su portal de datos abiertos[6]. Vamos a trabajar con un extracto de esos datos para el año 2022-2023, eligiendo un subconjunto de variables que corresponden a la llegada diaria de personas a Barcelona desde distintas procedencias (cercana, media, lejana) y su relación con la temperatura de la ciudad. Básicamente, comprobaremos que la variación de temperatura que sentimos en la

[6] https://portaldades.ajuntament.barcelona.cat/

ciudad se debe a la hospitalidad de los barceloneses con sus visitantes. Esa temperatura se verá registrada en las mediciones de la estación meteorológica de zona universitaria[7].

Puedes obtener estos datos mediante la función `load_BCN_calor` de la librería `apafib`. Resuelve los siguientes apartados ilustrando los resultados de la manera que te parezca más adecuada.

a) Divide el conjunto de datos en entrenamiento y test (60 %/40 %). Haz una exploración mínima del conjunto de datos de entrenamiento observando las relaciones entre las variables, especialmente con la variable objetivo. Describe las cosas que hayas visto que te parezcan interesantes. Transforma las variables adecuadamente para poder ajustar un modelo de regresión tanto para el conjunto de entrenamiento como para el de test.

b) Ajusta una regresión lineal, una regresión Ridge y una regresión LASSO a los datos ¿Te parece suficientemente bueno el resultado? Representa los valores de la variable objetivo para el conjunto de test contra las predicciones y observa los residuos y el qqplot. ¿Qué modelo te parece mejor? ¿Hay alguna variable que haya descartado LASSO?

c) Una manera de ver la relevancia de las variables en el modelo es comprobar la significatividad de los coeficientes de regresión. Ajusta una regresión lineal con todas las variables usando el método `OLS` de la librería `statsmodels` (los pesos seran los mismos que en la regresión lineal, compruébalo) y analiza la significatividad que asigna el método al coeficiente de cada variable. ¿Hay procedencias que son irrelevantes para el cambio de temperatura? Podemos ver la importancia de las procedencias a partir de la magnitud de los coeficientes de regresión y su sentido ¿quién contribuye más al aumento de temperatura? ¿De dónde vienen los que refrescan el ambiente?

d) Podríamos decir que la hospitalidad barcelonesa va más allá y es incluso probablemente no lineal. Usa la función `PolynomialFeatures` de scikit-learn para añadir características al conjunto de datos que correspondan a polinomios de grado 2. Ajusta de nuevo una regresión lineal y un modelo LASSO para estas variables y evalúa la calidad de los modelos. Representa las predicciones respecto a los valores reales, compara con los resultados de los modelos anteriores y comenta lo que observes.

6. **Cuidado con donde aparcas**

El conducir por grandes ciudades tiene emparejado el caos y el no saber donde dejar el coche. Los ayuntamientos no dejan pasar ni una y están prestos a multar la menor infracción. El ayuntamiento de Barcelona recolecta diversos datos sobre la ciudad en su portal de datos abiertos[6] y quiere estudiar cuáles son las circunstancias que influyen en el número de infracciones que se realizan en la ciudad. Vamos a trabajar con un extracto de esos datos para los años 2022-2023, eligiendo un subconjunto de variables relacionadas con el tráfico y sus circunstancias (y una variable extra adicional, por si acaso). Tenemos la contaminación medida en la estación meteorológica de l'Eixample (NO2), la contaminación acústica a diferentes horas del día, el tránsito en diferentes franjas horarias, la temperatura del día y el número de personas de otros municipios cercanos que vinieron a Barcelona. El objetivo es estimar el número de sanciones que se ponen en el día.

Puedes obtener estos datos mediante la función `load_BCN_sanciones` de la librería `apafib`. Resuelve los siguientes apartados ilustrando los resultados de la manera que te parezca más adecuada.

a) Divide el conjunto de datos en entrenamiento y test (60 %/40 %). Haz una exploración mínima del conjunto de datos de entrenamiento observando las relaciones entre las variables,

[7] Obviamente hay una explicación a este fenómeno, pero claramente se pueden sacar conclusiones equivocadas con cualquier conjunto de datos, hay que aplicar el sentido común.

especialmente con la variable objetivo. Describe las cosas que hayas visto que te parezcan interesantes. Transforma las variables adecuadamente para poder ajustar un modelo de regresión tanto para el conjunto de entrenamiento como para el de test.

b) Ajusta una regresión lineal, una regresión Ridge y una regresión LASSO a los datos ¿Te parece suficientemente bueno el resultado? Representa los valores de la variable objetivo para el conjunto de test contra las predicciones y observa los residuos y el qqplot. ¿Qué modelo te parece mejor? ¿Hay alguna variable que haya descartado LASSO?

c) Otra manera de ver la relevancia de las variables en el modelo es comprobar la significatividad de los coeficientes de regresión. Ajustad una regresión lineal con todas las variables usando el método `OLS` de la librería `statsmodels` (los pesos serán los mismos que los de la regresión lineal) y analiza la significatividad que asigna el método al coeficiente de cada variable. Explica lo que has visto. ¿Hay alguna posible razón para que algunas variables no sean importantes? Elimina las variables que no son significativas y ajusta de nuevo el mejor modelo. Comenta los resultados.

d) Siempre es posible que haya interacciones entre las variables que expliquen cosas que no obtengamos con las variables individuales. Usa la función `PolynomialFeatures` de scikit-learn para añadir características al conjunto de datos que correspondan a polinomios de grado 2. Ajusta de nuevo una regresión lineal, Ridge y LASSO para estas variables y evalúa la calidad de los modelos. Representa las predicciones respecto a los valores reales y comenta lo que observes. Compara las predicciones con las del mejor modelo con las variables originales.