2022/2023 1Q

Lista de Problemas 2

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 2022

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.

La parte que no es de programación la podéis hacer a mano y escanearla a un archivo **PDF**. Comprobad que **sea legible**.

Para la parte de programación 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:

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

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 diariamente (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. Dividid los datos en conjuntos de entrenamiento y test[1] (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. Estandarizad las variables (calculad el estandarizador a partir de los datos del entrenamiento y luego aplicadlo a los datos 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) La variable a predecir es un conteo. En este caso podría tener sentido utilizar un regresor que use un modelo adecuado para este tipo de datos como el `PoissonRegressor` de scikit-learn. Veréis que tiene un parámetro de regularización `alpha` que necesitaréis ajustar usando validación cruzada. Con el mejor parámetro de regularización según el error de validación cruzada, ajustad el modelo con los datos de entrenamiento y calculad la *calidad* del modelo con los datos de test.

[1] Fijad el parámetro `random_state` en la función `train_test_split` para que los resultados no cambien cuando repitáis el experimento.

d) A veces el modelo que parece adecuado no lo es tanto después de todo. Ajustad una regresión lineal a los datos y calculad la *calidad* del modelo empleando validación cruzada y con los datos de test.

e) Cuando se trata de entender un problema, es útil reducir el número de atributos en el modelo. Ajustad una regresión LASSO (ajustando la regularización usando la versión CV del modelo) y calculad la *calidad* del modelo con los datos de test.

f) Ahora podemos analizar y comparar los resultados:

- Comparad los valores pronosticados con los valores reales de los tres modelos y sus calidades. ¿Qué modelo os parece mejor? ¿Por qué?
- Analizad los pesos de la regresión lineal y LASSO. ¿Qué atributos son más importantes? ¿Hay atributos que no son relevantes?
- Comparad las predicciones de la regresión de Poisson y la regresión de LASSO. ¿Hay alguna diferencia en su comportamiento? ¿Qué creéis que está pasando?

2. **Test de asentamiento de cemento**

Muchas aplicaciones del aprendizaje automático son en problemas donde hay un proceso físico que es difícil de predecir y debe aproximarse. El conjunto de datos de test de asentamiento de cemento (Concrete Slump Test) del repositorio de conjuntos de datos de UCI recopila datos de las características de la composición de cemento para que se puedan predecir ciertas propiedades. Podéis descargar los datos desde aquí `https://archive.ics.uci.edu/ml/datasets/Concrete+Slump+Test`.

El objetivo de este problema es predecir la última variable de salida de las tres predicciones posibles (28 days compressive strength), eliminad las otras dos del conjunto de datos. Podéis leer en la página web del conjunto de datos la descripción de las variables.

a) El primer paso es preprocesar y preparar los datos antes de ajustar cualquier modelo. Eliminad las dos variables de salida del conjunto de datos que no vais a usar. Dividid los datos en conjunto de entrenamiento y test[2] (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. Estandarizad las variables (calculad el estandarizador a partir de los datos de entrenamiento y luego aplicadlo a los datos 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 a los datos y calculad la *calidad* del modelo utilizando validación cruzada y también con los datos de test.

d) Ajustad regresiones Ridge y LASSO a los datos ajustando sus parámetros de regularización (usando la versión CV de los modelos). ¿Son mejores las predicciones de las regresiones regularizadas? ¿Tiene la regularización algún impacto en los pesos?

e) Para algunos dominios, tener pesos negativos para los atributos no tiene interpretación. Un cambio que se puede aplicar a la regresión lineal en ciertos dominios es forzar que todos los pesos sean positivos. Todos los modelos de regresión lineal tienen un parámetro `positive` que se puede configurar para forzar esta restricción. Repetid el ajuste para los tres modelos

[2] Fijad el parámetro `random_state` en la función `train_test_split` para que los resultados no cambien cuando repitáis el experimento.

y comentad los resultados[3]. Representad el gráfico QQ de los residuos de los datos de test (podéis usar `scipy.stats.probplot`). ¿Los residuos siguen una distribución gaussiana?

f) Seleccionad el mejor modelo y calculad una gráfica de efectos (effect plot) utilizando los datos de entrenamiento[4]. Calculad los efectos para el primer ejemplo del conjunto de test, ¿es este ejemplo típico? En otras palabras, ¿las contribuciones se acercan a la contribución media de los datos de entrenamiento?

3. **Millas por galón**

El conjunto de datos *auto-mpg* es un conjunto de datos clásico que tiene las características de varios modelos de automóviles antiguos y tiene como objetivo predecir el millaje por galón de automóviles. Podéis descargar los datos desde aquí https://archive.ics.uci.edu/ml/datasets/auto+mpg.

a) El primer paso es preprocesar y preparar los datos antes de ajustar cualquier modelo. Tendréis que ocuparos de algunos valores perdidos en el conjunto de datos. Dividid los datos en conjuntos de entrenamiento y test[5] (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. Estandarizad las variables (calculad el estandarizador a partir de los datos del entrenamiento y luego aplicadlo a los datos 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 a los datos y calculad la *calidad* del modelo mediante validación cruzada y también con los datos de test. Representad las predicciones contra los valores reales. ¿Hay algo extraño en el gráfico?

d) Algunas veces los valores atípicos en los datos pueden sesgar el resultado y estropear las predicciones. Una solución es usar un modelo de regresión que sea tolerante a valores atípicos como la regresión de Huber. Este modelo ajusta el error absoluto medio en lugar del error cuadrático. Ajustad el `HuberRegressor` a los datos. Tendréis que ajustar los parámetros `epsilon` y `alpha` mediante validación cruzada (leed la documentación para ver qué significan). ¿Es este modelo mejor que el anterior?

e) Tal vez el problema es que la dependencia entre el objetivo y las variables no es lineal. Tenéis en `scikit-learn` un preprocesamiento que puede introducir características calculadas como polinomios de los datos originales (`PolynomialFeatures`). Transformad el problema agregando características que correspondan a polinomios de orden 2 y volved a ajustar la regresión lineal. Observad que esta transformación mantiene los atributos originales y añade todos los productos cruzados de las variables, por lo que también introduce las interacciones entre ellas. ¿Es mejor este modelo? ¿Qué hay del gráfico de las predicciones contra los valores reales?

f) Agregar más atributos al conjunto de datos es una receta para el sobreajuste. Ajustad una regresión Ridge a estos nuevos datos, ajustando la regularización utilizando la versión CV

[3] Para la regresión Ridge, el método `RidgeCV` aún no tiene el parámetro, pero `Ridge` sí lo tiene, por lo que tendréis que ajustar el parámetro de regularización primero con `RidgeCV` y luego utilzarlo con `Ridge`, no es exactamente lo mismo, pero es mas sencillo hacerlo así.

[4] La gráfica de efectos es una gráfica de caja (boxplot) de los valores de cada atributo multiplicados por los pesos del modelo.

[5] Fijad el parámetro `random_state` en la función `train_test_split` para que los resultados no cambien cuando repitáis el experimento.

del modelo. ¿Han cambiado los pesos del modelo? ¿Es mejor este modelo? ¿Alguno de los atributos de los polinomios tiene alguna importancia en el modelo?

CAPÍTULO 2

Problemas Individuales

Para obtener los datos para estos problemas necesitaréis instalaros la última versión de la libreria `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 retornara un `DataFrame` de Pandas con los datos. Si no es así el enunciado explicará que retorna.

1. **¿Que edad tienes?**

   El conjunto de datos *Stroke Prediction Dataset*[1] tiene la descripción de un conjunto de personas a partir de características demográficas, médicas y de hábitos. Usualmente, se pueden predecir diferentes cosas a partir de un conjunto de datos y en este caso vamos a intentar predecir la edad de las personas a partir de sus características.

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

   a) Elimina las variables que no tiene sentido usar. Divide el conjunto de datos en entrenamiento y test (70 %/30 %). 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.

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

b) Aplica Análisis de Componentes Principales (PCA) al conjunto de entrenamiento y visualízalo en 2D representando la variable objetivo. ¿Crees que puede haber una relación entre las variables del conjunto de datos y la variable objetivo? ¿Por qué?

c) Ajusta un modelo de regresión LASSO para predecir la edad de las personas y estima la calidad de la regresión. ¿Te parece suficientemente bueno el resultado? ¿Por qué?

d) Representa los residuos de la regresión y los valores reales contra las predicciones para el conjunto de test. ¿Qué te parece el resultado? ¿Observas algún comportamiento extraño? ¿Cuál puede ser la causa?

2. **La medicina es cara**

El coste de los seguros médicos varía bastante según las circunstancias de cada persona, pero a veces averiguar como se calcula realmente no es tan sencillo. El conjunto de datos `Medical Cost Personal Dataset`[2] tiene la descripción de las características de un grupo de personas y los cargos de su seguro médico. Nos interesa predecir esta última variable (`charges`).

Trabajaremos con una versión de este conjunto que podéis obtener mediante la función `load_medical_cost` 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 (70 %/30 %). 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 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.

c) La relación entre las variables del conjunto de datos y la variable objetivo podrían ser no lineal. Usa la función `PolynomialFeatures` de scikit-learn para añadir características al conjunto de datos que correspondan a polinomios de grado 2.

d) Ajusta a estos nuevos datos un modelo de regresión lineal y uno de regresión LASSO y estima la calidad de la regresión. ¿Te parece suficientemente bueno el resultado de los modelos? Representa los residuos y comenta que aparece. ¿Qué modelo te parece mejor? ¿Por qué?

3. **La respuesta está en el viento**

Las series temporales son un mundo aparte en análisis de datos y aprendizaje automático, pero hay modelos sencillos que se pueden adaptar para trabajar con ellas. El principal problema que tienen los datos temporales es que se incumple el que los ejemplos sean independientes entre sí. Eso complica por ejemplo la forma en la que se ha de hacer la validación.

En datos temporales el modelo más sencillo es el denominado auto regresivo (AR). Este sería equivalente a la regresión lineal. Si tenemos una variable temporal, el instante t se puede predecir a partir un número de los instantes anteriores, de manera que:

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

[2] Tenéis información sobre sus atributos en https://www.kaggle.com/datasets/mirichoi0218/insurance.

Donde c es una constante y ϵ_t es ruido gausiano. Se puede ver que se corresponde prácticamente con una regresión lineal usando una ventana p de observaciones anteriores.

El conjunto de datos `Wind Speed Prediction Dataset`[3] 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.

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) Para generar el conjunto de datos primero elimina la variable `DATE` y todas las variables `IND`. El tratamiento de datos perdidos en series temporales es diferente que en otros datos. Dado que los datos no son independientes hemos de rellenar los huecos en la serie temporal de maneras diferentes a las habituales. En este caso puedes usar la función `fillna` de Pandas usando el parámetro `method='ffill'`. Esto substituye los datos perdidos por el valor del último punto en la secuencia con un valor valido.

Para generar los datos, tendrás que obtener ventanas de una cierta longitud `w`. La función de numpy `sliding_window_view` permite obtener una vista de una matriz que corresponde a lo que necesitas. Si obtienes una ventana de longitud `w+1` esa última columna corresponderá al valor a predecir con los `w` elementos de la ventana. Tendrás que generar conjuntos de datos con solo la variable `WIND` y con todas las variables para longitudes de ventana 1, 3, 7 y 10. Fíjate que trabajas con matrices con 2 y 3 dimensiones dependiendo del número de variables que uses, ve con cuidado con las dimensiones y adapta los datos para poder aplicar regresión.

La validación en series temporales no puede usar validación cruzada, piensa cuál es la razón y explícalo en el informe. Tendrás que generar un conjunto de entrenamiento, uno de validación y uno de test. Selecciona los primeros 6000 ejemplos del conjunto de datos, usa los primeros 4000 para entrenamiento y el resto divídelo en validación y test, asegurándote que ninguno de los tres conjuntos comparta ventanas.

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 y LASSO con los diferentes conjuntos de datos y compara su calidad y las características de los modelos. El ajuste de parámetros no lo puedes hacer mediante validación cruzada, has de utilizar el conjunto de datos de validación. Selecciona el mejor modelo.

c) Representa el qqplot de los residuos del mejor modelo y comprueba si son gausianos. Comenta los resultados.

d) Representa la predicción del mejor modelo para una pequeña ventana ($\approx$ 200) de datos de la muestra de test. ¿Crees que la regresión está haciendo una buena aproximación de la serie temporal? ¿Qué características debería cumplir esta serie para que la regresión lineal fuera un buen modelo para predecirla?

4. **Correlación no es causalidad, pero tampoco casualidad**

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 al inicio de la pandemia empezó a recolectar

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

y mostrar diversos datos sobre la ciudad en su portal de datos abiertos[4]. Vamos a trabajar con un extracto de esos datos para el año 2021, eligiendo un subconjunto de variables que corresponden a la evolución de precios de un conjunto de alimentos y el número de vuelos del aeropuerto del Prat que tienen su origen y destino en Europa.

Puedes obtener estos datos mediante la función `load_BCN_vuelos` 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 (80 %/20 %). 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) Aplica Análisis de Componentes Principales (PCA) al conjunto de entrenamiento y visualízalo en 2D representando la variable objetivo. ¿Crees que puede haber una relación entre las variables del conjunto de datos y la variable objetivo? ¿Por qué?

c) Ajusta una regresión lineal 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 el qqplot. ¿Qué modelo te parece mejor?

d) Las cosas más extrañas suceden entre variables. Usa la función `PolynomialFeatures` de scikit-learn para añadir características al conjunto de datos que correspondan a polinomios de grado 2. Vuelve a ajustar la regresión lineal y la regresión LASSO. ¿Han mejorado los modelos? Vistos los resultados y el análisis inicial que has hecho de las variables ¿Te aventurarías a explicar por qué aparece esta relación entre las variables?

5. **¿Que trae a los Británicos a Barcelona?**

Ok, todo el mundo sabe lo que trae a los Británicos a Barcelona, pero ¿realmente es eso? El portal de datos abiertos del ayuntamiento de Barcelona nos ofrece la oportunidad de averiguarlo. El ayuntamiento de Barcelona al inicio de la pandemia empezó a recolectar y mostrar diversos datos sobre la ciudad[4]. Vamos a trabajar con un extracto de esos datos para el año 2021, con un subconjunto de variables que hemos elegido según nuestro criterio *experto* para desentrañar este misterio. El objetivo es aproximar el número de visitantes diarios de ciudadanos del Reino Unido a Barcelona.

Puedes obtener estos datos mediante la función `load_BCN_UK` 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 (80 %/20 %). 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) Aplica Análisis de Componentes Principales (PCA) al conjunto de entrenamiento y visualízalo en 2D representando la variable objetivo. ¿Crees que puede haber una relación entre las variables del conjunto de datos y la variable objetivo? ¿Por qué?

c) 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 el qqplot. ¿Qué modelo te parece mejor?

[4] https://dades.ajuntament.barcelona.cat/la-ciutat-al-dia/

d) La regresión LASSO nos ha indicado qué variables aparentemente no tienen influencia en nuestros amigos Británicos, pero todavía podríamos reducir algo más el misterio. La regresión lineal y la LASSO tienen el parámetro booleano `positive` que obliga que los coeficientes sean todos positivos. Repite el ajuste de estos dos modelos con este parámetro a cierto ¿Ha afectado mucho a la calidad del modelo? Decide de las variables que han quedado cuál es la que tiene menos sentido, elimínala del conjunto de datos y ajusta un modelo de regresión lineal sin restricciones. ¿Ha afectado mucho a la calidad del modelo? Comenta el resultado.

6. **Barcelona motor del IBEX**

La predicción bursátil es un problema complejo, pero a veces se pueden observar relaciones con variables que aparentemente no deberían influenciar. El portal de datos abiertos del ayuntamiento de Barcelona recoge informaciones diarias sobre la ciudad[4] y esto nos ofrece la oportunidad de averiguar si lo que pasa en Barcelona tiene alguna influencia en el mercado del IBEX. Vamos a trabajar con un extracto de esos datos para el año 2021, con un subconjunto de variables que hemos elegido según nuestro criterio *experto* desentrañar esa influencia. El objetivo es aproximar el valor de la cotización del IBEX a partir de las otras variables.

Puedes obtener estos datos mediante la función `load_BCN_IBEX` 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 (80 %/20 %). 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) Aplica Análisis de Componentes Principales (PCA) al conjunto de entrenamiento y visualízalo en 2D representando la variable objetivo. ¿Crees que puede haber una relación entre las variables del conjunto de datos y la variable objetivo? ¿Por qué?

c) 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 el qqplot. ¿Qué modelo te parece mejor? ¿Tienen sentido las variables con más peso que aparecen en los modelos para la variable que queremos predecir? Elimina las variables que tienen menos peso en los modelos del conjunto de datos y reajusta el modelo de regresión lineal ¿Cómo ha cambiado el peso de las variables que quedan?

d) Al ser un problema complejo, igual hay interacciones entre variables que explican mejor la variable objetivo. Usa la función `PolynomialFeatures` de scikit-learn para añadir al conjunto de datos original características que correspondan a polinomios de grado 2. Vuelve a ajustar la regresión Ridge y la regresión LASSO. ¿Han mejorado los modelos? Fíjate en las variables a las que LASSO no les ha dado un peso 0. ¿Se corresponden con interacciones entre variables?