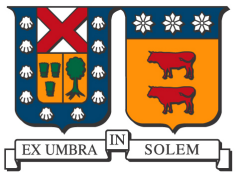

Universidad Técnica Federico Santa María
Departamento de Informática

1er Semestre 2016

Homework I - Machine Learning - INF

Prof.

Autor:
Francisco Mena Toro
francisco.mena.13@sansano.usm.cl
201373504-5
San Joaquín

Autor:
Francisco Perez Casto
francisco.perezca.13@sansano.usm.cl
201373516-9
San Joaquín

1. Introducción

a

2. Supuestos

a

3. Desarrollo

En este informe se presenta principalmente el estudio de diferentes tipos de regresiones, tales como la *Regresión Lineal Ordinaria*, *Regresión de Ridge* y *Regresión de Lasso*. Para poder comparar y ver cual se entrena mejor en base a un conjunto de datos (*training set*), midiendo los errores obteniendo en base a *training set*, *validation set* y *test set*.

Como bien se sabe, un modelo lineal tiene la forma:

$$f(x) = \beta_0 + \beta_1 x^{(1)} + \beta_2 x^{(2)} + ... + \beta_i x^{(N)} \tag{1}$$

Donde β_i, $i = 0, 1, 2, ..., N$ son los parametros asociados a cada caracteristica del modelo lineal.

En esta primera parte se trabaja sobre datos utilizando regresión lineal para predecir valores basados en la data de entrenamiento.

3.1. Regresión Lineal Ordinaria (LSS)

En esta sección se trabaja con un dataset (*prostate-cancer*) [1], donde con determinados atributos/características, se intentará predecir datos futuros mediante un algoritmo que ajusta un modelo de regresión lineal, en base a un conjunto de entrenamiento.

a) Primero se crea el dataset de trabajo, luego se puede observar que en la línea 5 se elimina una columna innecesaria, la cual se llama "*Unnamed: 0*" que entrega la información de enumeración de los datos, la cual viene por defecto en el dataframe *df*, es por esto que se elimina con el método *drop*(). Para el caso de la linea 9, se elimina la columna llamada "*train*" la cual fue extraída y guardada en otra variable anteriormente, además de que no forma parte de las variables del dataset o del target, ya que simboliza un valor de True o False si cada dato corresponde al training set o no.

b) El dataset se conforma de 97 entradas en el input space (χ), con 9 columnas, de las cuales 8 conforman las características de estudio y que predicirán la novena columna (target). Dentro de los valores del dataset existen valores decimales y valores enteros. Existen características que tienen una escala muy diferente a otras, tales como la edad (age), la cual tiene una media aritémica muy distintante a las demás características en el dataset. Por otro lado, existen características que poseen una desviación estandar muy alta comparado con el resto, tal como el porcentaje de Gleason escalado 4 o 5 (pgg45).

c) Es importante normalizar el dataset, para así tener una desviación estándar mas estandarizada y no con valores tan alejados entre sí, como es el caso del porcentaje de Gleason (pgg45) y la edad (age), entre otros. Es conveniente realizar esta operación, puesto que facilita enormemente el manejo de los datos, ya que deja a todos en un mismo rango de valores, por lo que permite realizar un mejor ajuste y evitando ademas problemas relacionados con los limites de representación computacional, además de poder comparar entre sí, al estar en la misma escala.

d) Luego, se realiza un ajuste lineal por mínimos cuadrados básica, para ello primero se necesita saber el tamaño de la data, lo que justamente se hace en el paso 3 a través del metodo shape(). Por otra parte,

el argumento de la función que implementa esta regresión lineal es *fit-intercept*.

Si se define el intercepto como $x^{(0)} = 1$, se puede definir el siguiente modelo lineal en forma matricial:

$$f(x) = \sum_{i=0}^{I} \beta_i x^{(i)} = \beta^T x \tag{2}$$

Donde I es el numero de caracteristicas.

Este argumento es de gran importancia puesto que indica si se realiza un ajuste en relación a un valor constante, es decir, el llamado intercepto. Como anteriormente ya se añadió una columna con un valor constante (1,0) para el intercepto, no es necesario indicarle a la función que lo realice, por lo que se le da un valor booleano *False*, que indica que los datos ya están normalizados.

e) Se construye una tabla con los pesos y Z-score correspondientes a cada predictor (variable). Las variables que tienen mas peso en el modelo se pueden observar mediante el analisis de los valores de los coeficientes de cada una. Este analisis tambien puede lograrse con la medida Z-score, el cual representa la misma informacion que los coeficientes, pero de una manera estandarizada, puesto que para obtener sus valores se resta el promedio correspondiente a cada variable, y se divide por su desviacion estandar. Como esta medida esta estandarizada como una distribucion normal, esto indica que una significacion del 5 % equivale aproximadamente a 2 desviaciones estandar. Siguiendo el mismo razonamiento, las variables con mas peso, es decir, mas correlacionadas con el target, son las siguientes: Lcavol, Lweight y en menor medida Lbph. Por esto mismo, las variables que no poseen suficiente evidencia para demostrar relacion con el target serian todo el resto de variables.

Atributo	Coeficiente	Std. Error	Z-score
Lcavol	0.5966394	0.1215977	4.9066641
Lweight	0.2723253	0.0918286	2.9655804
Age	-0.145638	0.0973158	-1.496554
Lbph	0.1892731	0.0981576	1.9282562
Svi	0.1794042	0.1186893	1.5115442
Lcp	-0.159118	0.1483895	-1.072302
Gleason	0.1007800	0.1394762	0.7225607
Pgg45	0.1148827	0.1475101	0.7788125
Intercept	2.4000628	0.0862118	27.839132

f) En esta parte se estima el error de predicción del modelo generado por el método de regresión lineal ordinaria por mínimos cuadrados, donde se cuenta con data de test "real" (*test set*).
Se compara el error del test set con el error en cross-validation, donde se divide la data en *k-folds* y donde se utilizan $k - 1$ folds de training set y un fold como validation set. El error se calcula como el error cuadrático medio (mse), obteniendo los errores reales (msetest) y los errores de prueba por *croos-validation* (msecv).

Se obtienen los siguientes resultados:

- Para el error del test set se obtiene un valor de 0.521274.
- Con K = 10 folds el error estimado de prueba con cross-validation 0.757237.
- Con K = 5 folds el error estimado de prueba con cross-validation 0.956514.

Como se puede ver el error en cross-validation es mayor que el error test "real", esto es ya que el error de cross-validation es un estimador del error de test, por lo que no necesariamente sera el mismo. Esto

es ya que cross-validation es dependiente del training set, ya que estima el error de prueba en base a esto, por lo que como la cantidad de datos del training set es baja, no llega a ser representativa por lo que la estimación del error de test no es precisa.

Para el caso de k=5 folds, el error estimado es más grande y mas alejado del error de test "real", esto es por la cantidad de datos en el training set es baja, ya que con 4 folds se tiene un 80 % de los datos del training set original para realizar el modelo lineal, es decir, se tienen menos datos, por lo que la estimación queda muy dependiente de los datos escogidos en el training set. Para el caso de k=10 folds, se tiene 9 folds para realizar el modelo lineal, es decir 90 % de los datos originales del training set. Es por esto que la estimación con k=5 es peor.

j) Se realiza un Q-Q plot sobre el error cuadrático medio (*mse*) en los datos de entrenamiento (*training set*), presentado a continuación:

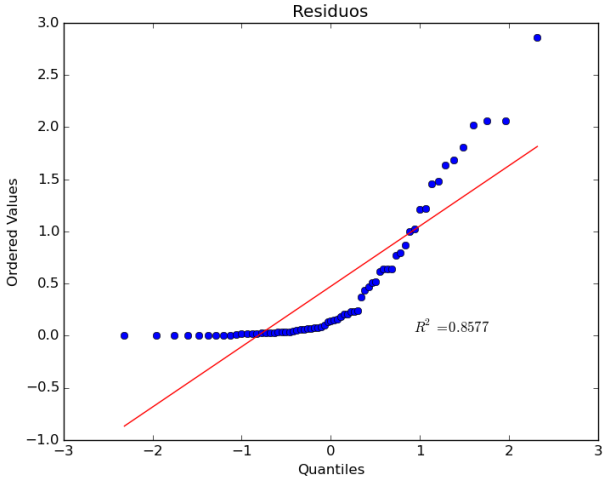

Figura 1: Q-Q plot del error de los datos de entrenamiento

Se puede ver como los datos de error de entrenamiento no siguen una distribución normal, debido a su comportamiento en el gráfico, de manera que no siguen la línea roja, por lo que el supuesto de normalidad en el residuo no vale.
Esta es una forma gráfica de ver si es que los datos siguen una distribución normal o no.

3.2. Selección de Atributos

Utilizando el mismo dataset (*prostate-cancer*) se realiza un estudio sobre seleccionar los atributos/características esenciales, es decir, que afecten mas el resultado de la variable que se desea predecir (target). Se realizan dos métodos para analizar el impacto que tiene cada atributo sobre el target, una es mediante FSS en donde se va seleccionando cada atributo, uno por uno, siendo esa variable la mejor en ese momento. Otra es mediante BSS, la cual va eliminando un atributo en cada iteración, siendo ese atributo el que menos influencia tiene en el resultado de la variable target.

a) FSS parte de un modelo sin atributos (variables) y agrega uno a la vez, eligiendo la mejor localmente. Para este caso la selección fue en el orden siguiente:

i	Variable seleccionada	MSE	variables
1	Lcavol	0.876172	2
2	Lweight	0.752606	3
3	Lbph	0.748883	4
4	Svi	0.746635	5
5	Pgg45	0.748007	6
6	Lcp	0.734094	7
7	Age	0.726706	8
8	Gleason	0.757237	9

Se ve que el número de variables parte en 2, esto es ya que el intercepto es considerado como la primera variable (β_0) del modelo lineal por lo que debe ir en un principio en cada modelo que se haga.

Donde se puede ver como las primeras características en escoger son *Lcavol* y *Lweight*, corroborando el análisis anterior hecho con el Z-score de las variables(atributos) en el modelo, por lo que se puede ver que FSS dice que estas dos primeras características son las mas relevantes para predecir el target. Se ve como la última característica en ser escogida es *Gleason*, por lo que se puede decir que según FSS esta característica tiene la menor influencia sobre el target.

Se realiza un gráfico con los errores de entrenamiento y de test, sobre cada modelo que se construyó con cada selección de las características descritas en la tabla superior. Mostrando a continuación:

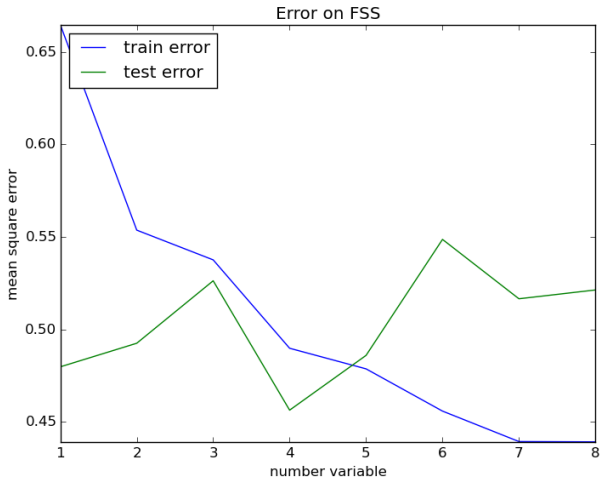

Figura 2: errores en FSS en regresión lineal ordinaria

Se puede ver como el error de entrenamiento disminuye a medida que se escogen mas variables, esto es ya que el algoritmo FSS funciona de esta manera, trabajando sobre el training set y escogiendo lo que es mas óptimo para este, donde mientras mas variables se seleccionan más se puede predecir del target, ya que el error va disminuyendo.
Para el caso del error de test, este varía aleatoriamente, ya que no existen métodos para predecir este error, por lo que las decisiones que va tomando FSS no entregan información sobre el error de test, sino que realiza operaciones sobre el training set, esperando que estas afecten al test set.
Se puede ver que para el caso en que el numero de variables se acerca a la cantidad maxima del modelo, el error de entrenamiento disminuye, incluso llegando a ser menor que el error de prueba. Este caso se conoce como *overfitting*, donde el modelo se sobre-ajusta a los datos de entrenamiento, prediciendo que el error disminuye, siendo que en verdad este se mantiene igual.

b) BSS parte de un modelo completo (todas las características), eliminando una característica a la vez. Para este caso, el orden de eliminar cada atributo fue el siguiente:

i	Variable seleccionada	MSE	variables
1	Gleason	0.726706	9
2	Age	0.734094	8
3	Lcp	0.748007	7
4	Pgg45	0.746635	6
5	Svi	0.748883	5
6	Lbph	0.752606	4
7	Lweight	0.876172	3
8	Lcavol	1.795596	2

Para esta tabla el número de variables (última columna) son las variables presentes en el modelo, es

por esto que va disminuyendo, ya que en cada iteración va disminuyendo este número.
Se puede ver como la primera variable en ser eliminada fue la característica *Gleason*, analogo al caso de FSS donde esa fue la última característica en ser seleccionada, por lo que se puede decir que según BSS esta característica es la que menos peso tiene al predecir la variable target. Se ve como la últimas características en ser eliminadas fue *Lcavol* y *Lweight*, analogo al caso de FSS y entregando más argumentos a favor de que estas son las características mas relevantes del modelo.

A continuación se presenta un gráfico con los errores de entrenamiento y de test, para cada modelo formado con cada eliminación de cada característica. Mostrando a continuación:

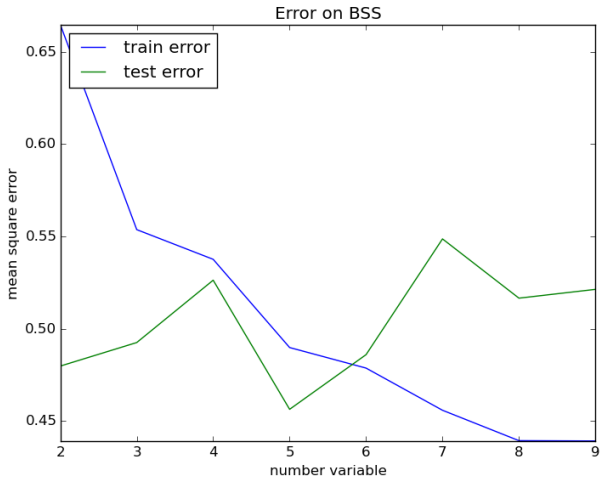

Figura 3: error BSS en regresión lineal ordinaria

Gráfico similar al caso del algoritmo FSS, esto ocurre ya que estos algoritmos trabajan sobre el training set y uno es análogo a otro, ya que ambos llegan al mismo resultado, es decir, al mismo orden de importancia de las caracteristicas. A medida que el modelo tiene mas características existe una precisión mayor al predecir el target en el training set, pero para el test set no es posible predecir o disminuir este error con la selección.

3.3. Regularización

En esta parte del informe se tratara el tema de la regularización, en donde se castigan los coeficientes altos del modelo lineal, regularizando estos y dándoles un límite de valor para tomar y ver como afecta esto al modelo. Para eso se harán distintas comparaciones entre los errores de test y entrenamiento, para distintos algoritmos de aprendizaje donde cada uno tiene un método de regularización diferente, incluyendo diferentes rangos en la regularización.

a) En primer lugar, se estudia la regularizacion del método de "*Ridge Regression*" el cual regulariza los atributos o variables, para obtener un modelo lineal mas preciso y asi poder analizar la data de mejor forma y obtener comparaciones entre estas variables.
En el código entregado, en la tercera línea se puede observar que se elimina la columna *intercept* con el metodo drop(), puesto que ya no se necesita, ya que la regularización es sobre los atributos/características del modelo y el intercepto no es modificado.
Ademas, en la novena línea se ejecuta el método de Ridge a través de la factorización svd, con el parámetro *fit intercept* asignado como *True*, puesto que como la columna fue eliminada, el mismo método se encarga de calcular el intercepto.

A continuacion, se presenta el gráfico de la regularización:

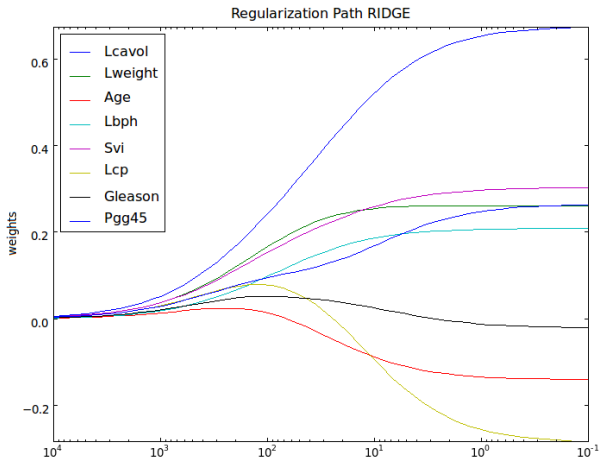

Figura 4: Gráfico de regularización de Ridge.

El gráfico representa el peso que tiene cada variable o característica (eje y) en el modelo general por cada alpha (eje x) estudiado en el rango [10^4,10^{-1}]. La forma del grafico se atribuye a la restriccion cuadratica que presenta la regresion:

$$\sum_{i=1}^{I} \beta_i^2 \leq s \tag{3}$$

Se puede observar del gráfico que las características que poseen un peso superior a las demas características son *Lcavol* y *Lweight*, puesto que al ir disminuyendo el valor de alpha, los pesos de estas variables aumentan mas que las otras. También se puede ver que *svi* empieza a tener mas peso que la variable *Lweight* solamente al disminuir el alpha a un valor inferior a 10^-1. Esto apoya las afirmaciones anteriores de los métodos FSS, BSS y Z-score con un nivel de significancia del 5 %, obteniendo los mismos resultados, es decir, las variables *Lcavol* y *Lweight* poseen mas incidencia en la decision (target).

b) Por otro lado, se puede hacer el mismo análisis para la regularización con el método de "*Lasso*", regularizando los atributos de una manera mas estricta que "*Ridge*", ya que tiene una penalización mas alta para los atributos, siendo más riguroso a la hora de decidir cual característica tendrá mas peso en el modelo.

Para esto se obtiene el siguiente gráfico:

El gráfico representa el peso que tiene cada variable o característica (eje y) en el modelo general por cada alpha (eje x) estudiado en el rango [10^1,10^{-2}]. La forma del gráfico se atribuye a la restricción lineal siguiente:

$$\sum_{i=1}^{I} |\beta| \leq s \tag{4}$$

Se puede observar del gráfico, que posee una forma mucho menos "suave" que con el metodo de *Ridge*, incluso llevando los pesos de cada variable a cero luego de aumentar el valor de alpha más allá de 10^0.

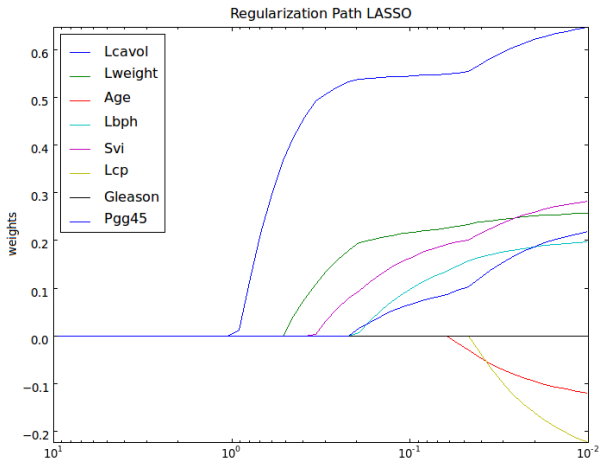

Figura 5: Gráfico de regularización de Lasso

Esto concuerda con lo esperado, puesto que se sabe de antemano que *Lasso* posee un nivel de penalizacion mucho mas riguroso, y por lo tanto, a medida que los valores de alpha disminuyen la diferencia entre los pesos de las variables es mas notoria.

c) Para este caso se visualiza gráficamente cómo se relaciona el error de prueba y de entrenamiento con la penalización (alpha), mostrando a continuación:

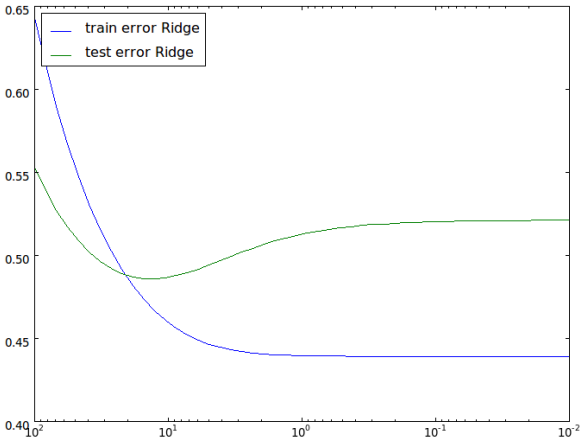

Figura 6

Si se realiza un analisis en conjunto con la ecuacion de Ridge (β^{ridge}), se observa que para alphas pequeños ($\alpha \sim 0$) la penalizacion es baja, por lo tanto, los valores de β_i no estan tan limitados, obteniendo valores similares a las del modelo de regresion lineal ordinaria. A medida que aumenta el valor de alpha, y con esto la penalizacion, los valores de β_i se ven mas restringidos, traduciendose a una ecuacion en donde las caracteristicas no van a tener tanta influencia sobre el target, dependiendo en gran medida del intercepto.
La forma cuadrática del grafico nuevamente se atribuye a la ecuacion ((3)).

Este metodo de Ridge tiene un objetivo similar que los metodos de seleccion de caracteristicas (FSS y BSS), pero en lugar de eliminar las caracteristicas del modelo estas son "amortiguadas" asignandoles un coeficiente (peso) bastante bajo.

Se puede observar ademas, que para un alpha apropiado el modelo de Ridge es mejor que el de minimos cuadrados, debido a la relacion entre los errores de entrenamiento y de prueba. REVISAR

d) Analogo al caso anterior, se ve como afecta la penalización de la variable *alpha* para los distintos atributos:

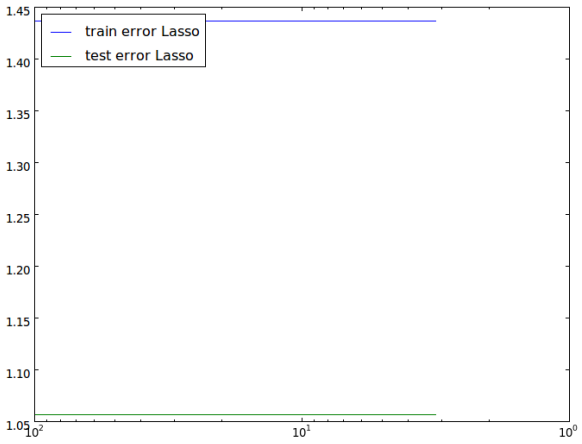

Figura 7

En contraste a la ecuacion de Lasso se conoce que esta es bastante restrictiva con los coeficientes (β_i). Esto es visualizado en el grafico, ya que para los valores de alpha que se presentan los coeficientes ya han sido asignados a cero (ver grafico 5), por lo que los errores que se presentan son de un modelo sin variables, es decir, un modelo constante, y de ahi la forma del grafico. Se puede ver que para valores pequeños de alpha estos ya generan que el modelo no tenga variables ($\beta_i = 0 \forall i$). De esto se desprende que el modelo a utilizar va a depender del training set, es decir, si se desea nivelar el peso de ciertas caracteristicas de una forma mas fuerte que Ridge, es conveniente usar Lasso.

e) Para conocer qué valor de *alpha* es el óptimo en los distintos metodos presentados (Lasso y Ridge). Se utiliza la tecnica *cross-validation* con el fin de probar distintos alphas y medir el error con los *k-folds*, para luego determinar el alpha con menor error cuadratico medio de todos los cross-validation.

Para los distintos modelos se obtuvieron los siguientes resultados:

- Metodo de Ridge: $\hat{\alpha} = 2{,}33$, con un error de mse(cv): 0,75
- Metodo de Lasso: $\hat{\alpha} = 0{,}10$, con un error de mse(cv): 0,87

Al comparar ambos metodos, es facil visualizar que el metodo de Ridge posee un error cuadratico medio menor que el de Lasso, esto nos quiere decir que el valor de alpha mas seguro en este caso seria de 2,33.

3.4. Predicción de utilidades de películas

En esta parte de la, se trabaja con un dataset que contiene datos sobre peliculas, con el objetivo de hacer un estudio sobre estas y poder predecir las utilidades que se obtendran del estreno de cada una en USA.

- a) En primer lugar, se cuenta con una matriz de gran tamaño, la cual contiene muchos datos nulos, por lo que la estructura en la que se presenta esta matriz es en un formato sparse (disperso), es decir, los valores no nulos se encuentran alejados unos de otros. Mantener este formato es de gran utilidad, ya que se ahorra espacio en el disco, ademas de proveer eficiencia en operaciones aritmeticas. Se conoce que los modelos de regresion lineal utilizan operaciones matriz vector, las cuales son costosas computacionalmente y dificultarian de gran manera el estudio de los datos.
- b)

4. Anexo

Lo demás adjuntado en archivos .py

Referencias

[1] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.