| DATOS PERSONALES | FIRMA |
| --- | --- |
| |  |  | | --- | --- | | Nombre: JUAN CARLOS | DNI: 1103332134 | |  |
| Apellidos: JARAMILLO JARAMILLO |

| ESTUDIO | ASIGNATURA | CONVOCATORIA |
| --- | --- | --- |
| MÁSTER UNIVERSITARIO EN INGENIERÍA MATEMÁTICA Y COMPUTACIÓN (PLAN 2016) | 4391020006.- TÉCNICAS MULTIVARIANTES | Ordinaria Número periodo 1823 |

| FECHA | MODELO | CIUDAD DEL EXAMEN |
| --- | --- | --- |
| 14-16/01/2022 | Modelo - C |  |

| Etiqueta identificativa |
| --- |
|  |

**INSTRUCCIONES GENERALES**

1. Ten disponible tu documentación oficial para identificarte, en el caso de que se te solicite.
2. Si tu examen consta de una parte tipo test, indica las respuestas en la plantilla según las características de este.
3. Debes contestar en el documento adjunto, respetando en todo momento el espaciado indicado para cada pregunta. Si este es en formato digital, los márgenes, el interlineado, fuente y tamaño de letra vienen dados por defecto y no deben modificarse. En cualquier caso, asegúrate de que la presentación es suficientemente clara y legible. Entrega toda la documentación relativa al examen, revisando con detenimiento que los archivos o documentos son los correctos. El envío de archivos erróneos o un envío incompleto supondrá una calificación de “no presentado”.
4. Durante el examen y en la corrección por parte del docente, se aplicará el Reglamento de Evaluación Académica de UNIR que regula las consecuencias derivadas de las posibles irregularidades y prácticas académicas incorrectas con relación al plagio y uso inadecuado de materiales y recursos.
5. No está permitido el uso de Internet ni ningún tipo de comunicación con otra persona.Durante todo el examen tu teléfono móvil debe estar en modo avión.
6. La parte principal de cada pregunta consiste en interpretar y comentar los resultados obtenidos. Si te limitas a hacer los cálculos no vas a poder superar el examen.
7. Es fundamental que las respuestas estén debidamente redactadas, de forma clara y precisa y sin faltas de ortografía.
8. Para hacer el examen puedes utilizar los apuntes del curso y los scripts que hayas preparado y Python para hacer los cálculos.

**Puntuación**

**Preguntas**

* Puntuación máxima 10.00 puntos

El examen constará de un ejercicio práctico (8,5 puntos) y una pregunta teórica (1,5 puntos). Los enunciados están en la página 14 y el espacio para responder el examen está entre las práginas 4 y 13.  
  
**1.** Pregunta

Estimada profesora, le pido mil disculpas no pude cargar el csv que cree en excel, no se que paso pero no pude cargarlo en spyder, con todo le adjunto los archivos de excel, cvs y codigos.

Lo que si puedo comentar de acuerdo a las actividades realizadas, es que de los modelos desarrollados debemos escoger el que tenga el menor error cuadrático medio.

Tamien puedo comentar que en el modelo Lasso y Ridge, cuanto mas grande sea el alpha, los coeficientes disminuyen y el modelo se vueleve mas robusto, por eso elegimos alpha=10 000.}

Mientras que en la regresión elástica debemos ir rpobado valores de l1\_ratio, bajos cercanos a cero para acrecarse a la regresion Ridge y valores cercanos a 1 para ajustarse a la regresion Lasso.

#Importamos todas las librerias que vamos a utilizar

import matplotlib.pyplot as plt

from sklearn.datasets import make\_regression

import os, sys

import pandas as pd

import os

import tarfile

import urllib.request

import numpy as np

from pandas.core.common import flatten

from plotnine import \*

from array import \*

import scipy.stats as stats

import math

import matplotlib as mpl

import statsmodels.api as sm

from sklearn import linear\_model

from sklearn.metrics import mean\_squared\_error, r2\_score

import statsmodels.formula.api as smf

import statsmodels.stats.api as sms

dataf\_x\_y = pd.read\_csv("datos\_ok.csv")

print(dataf\_x\_y)

#dataf\_y=pd.DataFrame(Y,columns=['y'])

dataf\_x\_y.describe()

dataf\_x\_y.info()

# dividir los histogramas numericos------------------------------

fig, axes = plt.subplots(nrows = 5, ncols = 3, figsize = (30,30))

# definir nombres ejes--------------------------------------------

# primera fila

axes[0,0].set\_xlabel("x1")

axes[0,0].set\_ylabel("y")

axes[0,1].set\_xlabel("x2")

axes[0,1].set\_ylabel("y")

axes[0,2].set\_xlabel("x3")

axes[0,2].set\_ylabel("y")

# segunda fila

axes[1,0].set\_xlabel("x4")

axes[1,0].set\_ylabel("y")

axes[1,1].set\_xlabel("x5")

axes[1,1].set\_ylabel("y")

axes[1,2].set\_xlabel("x6")

axes[1,2].set\_ylabel("y")

# tercera fila

axes[2,0].set\_xlabel("x7")

axes[2,0].set\_ylabel("y")

axes[2,1].set\_xlabel("x8")

axes[2,1].set\_ylabel("y")

axes[2,2].set\_xlabel("x9")

axes[2,2].set\_ylabel("y")

# cuarta fila

axes[3,0].set\_xlabel("x10")

axes[3,0].set\_ylabel("y")

axes[3,1].set\_xlabel("x11")

axes[3,1].set\_ylabel("y")

axes[3,2].set\_xlabel("x12")

axes[3,2].set\_ylabel("y")

# quinta fila

axes[4,0].set\_xlabel("x13")

axes[4,0].set\_ylabel("y")

axes[4,1].set\_xlabel("x14")

axes[4,1].set\_ylabel("y")

dataf\_x\_y.hist(bins = 50, ax = axes);

plt.xlabel("x");

plt.ylabel("residuos");

plt.show();

plt2=plt.figure(figsize=(3,3))

ax=plt2.add\_subplot(111, title='Variable respuesta')

ax.hist(Y) #aqui ponemos la variable respuesta Y

ax.grid

#Partición de la muestra con 20/220 (aprox 0.09090909090909090909) datos de test

np.random.seed(22233322)

# definir la funcion particiones.

#8 para entrenamiento, 2 para validación.

def particiones(dataset, valida\_part):

valida\_part\_size = int(len(dataset) \* valida\_part)

mezclar\_indices = np.random.permutation(len(dataset))

valida\_indices = mezclar\_indices[:valida\_part\_size]

entrena\_indices = mezclar\_indices[valida\_part\_size:]

return dataset.iloc[entrena\_indices], dataset.iloc[valida\_indices]

# usar funcion particiones con valida\_part 0.2---------------------

entrena\_set, valida\_set = particiones(dataf\_x\_y, 0.2)

# comprobar longitudes de los set de entrenamiento y de validación-----

print(round(len(dataf\_x\_y) \* 0.8, 1)) #longitud del dataframe dataf\_x\_y (10 datos\*0.8)=8 datos

print(len(entrena\_set)) #Longitud del set de entrenamiento

print(len(dataf\_x\_y) \* 0.2) #longitud del dataframe dataf\_x\_y (10 datos\*0.2)=2 datos

print(len(valida\_set)) #Longitud del set de validación.

#Describimos el conjunto de datos

print(dataf\_x\_y.info())

print(dataf\_x\_y.describe())

print(entrena\_set.info())

print(valida\_set.info())

print(entrena\_set.describe())

print(valida\_set.describe())

# separar variable respuesta del dataset de entrenamiento y de validación (pag. 22 tema 5)-------------------------

respuesta\_entrena = entrena\_set["y"].copy()

entrena\_set\_sin\_y = entrena\_set.drop("y", axis=1)

respuesta\_valida = valida\_set["y"].copy()

valida\_set\_sin\_y = valida\_set.drop("y", axis=1)

#Regresión Ridge

from sklearn.linear\_model import Ridge

# ajustar el modelo----------------------------------------------

ridge\_reg\_b = Ridge(alpha = 1e4, solver = "auto")

ridge\_reg\_b.fit(entrena\_set\_sin\_y, respuesta\_entrena)

# obtener coeficientes del modelo--------------------------------

# intercepto

print(ridge\_reg\_b.intercept\_)

# coeficientes de regresion

print(ridge\_reg\_b.coef\_)

#Calcular el error cuadrático medio

y\_pred\_Ridge = e\_net.predict(valida\_set\_sin\_y)

residuos\_Ridge = respuesta\_valida - y\_pred\_Ridge

ecm\_Ridge = mean\_squared\_error(respuesta\_valida, y\_pred\_RE)

#Regresión Lassso

# importar clase-------------------------------------------------

from sklearn.linear\_model import Lasso

# ajustar el modelo----------------------------------------------

lasso\_reg = Lasso(alpha = 1e4)

lasso\_reg.fit(entrena\_set\_sin\_y, respuesta\_entrena)

# obtener coeficientes del modelo--------------------------------

# intercepto

print(lasso\_reg.intercept\_)

# coeficientes de regresion

print(lasso\_reg.coef\_)

#Calcular el error cuadrático medio

y\_pred\_Lasso = e\_net.predict(valida\_set\_sin\_y)

residuos\_Lasso = respuesta\_valida - y\_pred\_Lasso

ecm\_Lasso = mean\_squared\_error(respuesta\_valida, y\_pred\_RE)

#Red elastica

# semilla para que los resultados sean los mismos----------------

np.random.seed(3)

# importar clase-------------------------------------------------

from sklearn.linear\_model import ElasticNetCV

# ajustar el modelo----------------------------------------------

e\_net = ElasticNetCV(cv = 10, l1\_ratio = 0.1)

e\_net.fit(entrena\_set\_sin\_y, respuesta\_entrena)

# obtener coeficientes del modelo--------------------------------

# intercepto

print(e\_net.intercept\_)

# coeficientes de regresion

print(e\_net.coef\_)

# semilla para que los resultados sean los mismos----------------

np.random.seed(3)

# importar clase-------------------------------------------------

from sklearn.linear\_model import ElasticNetCV

# ajustar el modelo----------------------------------------------

e\_net\_b = ElasticNetCV(cv = 10, l1\_ratio = 0.95)

e\_net\_b.fit(entrena\_set\_sin\_y, respuesta\_entrena)

# obtener coeficientes del modelo--------------------------------

# intercepto

print(e\_net\_b.intercept\_)

# coeficientes de regresion

print(e\_net\_b.coef\_)

#Calcular el error cuadrático medio

y\_pred\_Red = e\_net.predict(valida\_set\_sin\_y)

residuos\_Red = respuesta\_valida - y\_pred\_Red

ecm\_Red = mean\_squared\_error(respuesta\_valida, y\_pred\_Red)

 (Responder en 10 caras)

-

