| DATOS PERSONALES | FIRMA |
| --- | --- |
| |  |  | | --- | --- | | Nombre: | DNI: | |  |
| Apellidos: |

| 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.

Librerias

import os, sys

# cargar librerias-----------------------------------------------

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 matplotlib.pyplot as plt

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

1. **Division del data set**

**#Ingresamos la las columas x1, x2, x3, x4, x16**

**X=[[-0.21,1.59,0.45,1.06,1.14],**

**[-0.25,-0.01,2.67,-0.69,-2.68],**

**[-0.63,1.56,-0.05,-0.25,1.61],**

**[0.65,1.43,-0.17,1.50,1.60],**

**[0.15,-0.28,-0.81,-0.55,0.53],**

**[0.89,0.07,0.13,0.49,-0.06],**

**[0.16,0.60,1.29,-0.26,-0.70],**

**[-0.09,-0.17,0.43,-0.06,-0.59],**

**[0.78,-0.60,-1.26,-0.03,0.66],**

**[-0.43,-0.76,-0.68,0.76,-0.08]]**

**#Iingresamos Y**

**Y=[[92.55],[123.16],[75.20],[84.69],[-8.76],[106.57],[72.97],[-20.51],[-43.37],[-122.46]]**

**df\_X=pd.DataFrame(X)**

**df\_Y=pd.DataFrame(Y)**

**df\_X.describe()**

**Imagen de la pantalla de una computadora

Descripción generada automáticamente con confianza baja**

**df\_Y.describe()**

**Texto

Descripción generada automáticamente**

**Particiones**

**def particiones(target, dataset, test\_part):**

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

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

**test\_indices = mezclar\_indices[:test\_part\_size]**

**train\_indices = mezclar\_indices[test\_part\_size:]**

**return dataset.iloc[train\_indices], dataset.iloc[test\_indices], target.iloc[train\_indices], target.iloc[test\_indices]**

**# Usar funcion particiones con test\_part 1-200/220 = 20/220---------------------**

**X\_train, X\_test, Y\_train, Y\_test = particiones(df\_Y, df\_X, 0.2)**

**print(X\_train.info())**

**print(X\_train.describe())**

**Imagen que contiene texto, placa, sostener, tabla

Descripción generada automáticamente**

**Nos quedarian 8 entradas de entrenamiento**

**print(Y\_train.info())**

**print(Y\_train.describe())**

**Imagen que contiene texto, calle

Descripción generada automáticamente**

**Renombramos las columnas**

**X\_train.columns=['x\_train\_1',**

**'x\_train\_2',**

**'x\_train\_3',**

**'x\_train\_4',**

**'x\_train\_5'**

**]**

**Y\_train.columns=['y\_train']**

**X\_test.columns=['x\_test\_1',**

**'x\_test\_2',**

**'x\_test\_3',**

**'x\_test\_4',**

**'x\_test\_5'**

**]**

**Y\_test.columns=['y\_test']**

1. **Metodo de Ridge**

**from sklearn.linear\_model import Ridge**

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

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

**modelo\_ridge=ridge\_reg.fit(X\_train, Y\_train)**

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

**# intercepto**

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

**# coeficientes de regresion**

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

**y\_pred\_ridge=modelo\_ridge.predict(X\_test)**

**print(y\_pred\_ridge)**

**print(Y\_test)**

**Los resultados obtenidos fueron:**

**El valor del intercepto fue: 10.73**

**Los coeficientes encontrados fueron: 47.73407885 48.73297248 43.72390262 -16.08085336 4.65224313**

**Al utilizar los valores de prueba hacemos la comparacion**

|  |  |
| --- | --- |
| **Valores ajustados** | **Valores de prueba** |
| **104.93** | **72.97** |
| **86.13** | **92.55** |

Podemos ver que el ajuste no ha sido del todo bueno, dado que hay una distancia importante entre la preduccion y los datos reales. Esto puede deberse a la existencia de outliers en el conjunto de datos

**Metodo de Lasso**

**from sklearn.linear\_model import Lasso**

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

**lasso\_reg = Lasso(alpha = 1)**

**modelo\_lasso=lasso\_reg.fit(X\_train, Y\_train)**

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

**# intercepto**

**print("El intercepto es ",lasso\_reg.intercept\_)**

**# coeficientes de regresion**

**print("Los coeficientes del metodo de Lasso son ",lasso\_reg.coef\_)**

**y\_pred\_lasso=modelo\_lasso.predict(X\_test)**

**print("Datos ajustados", y\_pred\_lasso)**

**print("Datos Reales de prueba",Y\_test)**

**Los resultados obtenidos fueron:**

**El valor del intercepto fue: 7.03**

**Los coeficientes encontrados fueron: 74.49030168 64.57308759 42.99387878 -27.94286798 -0.**

|  |  |
| --- | --- |
| **Valores ajustados** | **Valores de prueba** |
| **120.42** | **72.97** |
| **83.79** | **92.55** |

La prediccion bajo este metodo es peor que la anterior , esto pues tambien puede suceder por que la cantidad de datos de entrenamiento no es muy grande y esto hace que la preduccion no se tan precisa

Red Elastica

from sklearn.linear\_model import ElasticNetCV

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

red\_elastica = ElasticNetCV(l1\_ratio = [0.1, 0.4,0.5,0.6,0.7,0.9,0.95,0.99], \

n\_alphas=100, alphas=[0.01,0.05,0.1,0.11])

red\_elastica.fit(X\_train, Y\_train)

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

# intercepto

print("El intercepto es",red\_elastica.intercept\_)

# coeficientes de regresion

print("Los coeficientes encontrados fueron", red\_elastica.coef\_)

param3=np.insert(red\_elastica.coef\_,0,red\_elastica.intercept\_)

##Validacion cruzada

print("El valor optimo de r es: \n",red\_elastica.l1\_ratio\_)

print("El valor optimo de alfa es: \n",red\_elastica.alpha\_)

y\_pre\_elastica=red\_elastica.predict(X\_test)

print("Datos ajustados ",y\_pre\_elastica)

print("Datos reales de prueba ",Y\_test)

Los resultados obtenidos fueron:

Intercepto: 7.8

Coeficientes: 69.17129525 57.254606 47.89688543 -26.4351446 5.61638567

Validacion cruzada:

Valor optimo de r: 0.7

Valor optimo de alfa es: 0.1

**Al utilizar los valores de prueba hacemos la comparacion**

|  |  |
| --- | --- |
| **Valores ajustados** | **Valores de prueba** |
| **117.95** | **72.97** |
| **84.25** | **92.55** |

De nuevo encontramos una distancia importante entre los valores predichos y los reales de prueba.

Error cuadratico medio

Ridge

El Error Cuadratico Medio es de: 8.818061191105034

Lasso

El Error Cuadratico Medio es de: 8.818061191105034

Red Elasitca

El Error Cuadratico Medio es de: 8.818061191105034

Los tres modelos se comportaron de formas similares

