|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| logo berri CIFP  CIFP Ciudad Jardín LHII | | | **Curso de Especialización:** Inteligencia artificial y Big Data | | | |
| **Modulo:**  Sistemas de aprendizaje automático  Programación de Inteligencia artificial | | | **Cód. Modulo:**  SAPA  PRIA |
| **Curso:** IABD | | | |
| **1ª Evaluación** | | | **Fecha**: 02/12/2024 | | | |
| **Cód. Prof:** PTI2  **Cód. Prof:** EI15 | | **Profesor:** Ainara Montoya  **Profesor:** Amaia De la cruz | |  | | |
|  | | | | | | |
| **Datos del Alumno** | | | | | | |
| **Nº** | **Nombre: Elena** | | | | **X**  **Firma** | |
| **Apellido 1: Sanz** | | | | |
| **Apellido 2: Espada** | | | | |

**Calificación:**

# EXAMEN

## EJERCICIO 1

Carga el conjunto de datos Iris (el de las flores según la anchura y la altura del pétalo y del sépalo) desde scikit-learn (se ha hecho en algún ejercicio).

Explora los datos visualmente utilizando gráficos: representa la distribución de cada variable (todas las de X e y), diagramas de cajas de cada variable (todas las de X) y la matriz de correlaciones para entender las relaciones entre las características. ¿Qué puedes deducir?

Divide el conjunto de datos en entrenamiento y prueba, estratificando por el tipo de flor.

Prepara los datos mediante una pipeline.

Entrena y evalúa un clasificador Random Forest utilizando validación cruzada. ¿Qué métrica has usado para evaluar el modelo y por qué? ¿Cuál es el mejor y el peor error que consigues?

Optimiza el número de árboles, ¿cuál es el mejor valor?

Evalúa el rendimiento del modelo optimizado en el conjunto de prueba.

Guarda el modelo en un fichero y cárgalo desde un fichero .py que pida 4 datos para una nueva flor (anchura del pétalo, longitud del pétalo, anchura del sépalo y longitud del sépalo) y prediga de que clase es la flor (quiero que me de un nombre, no un número y hay que controlar las excepciones).

## EJERCICIO 2

En este ejercicio, trabajarás con un conjunto de datos de pruebas creados (ejercicio2.csv).

Tu objetivo es construir un modelo de regresión que pueda predecir con precisión la variable dependiente y a partir de las variables independientes x1 y x2.

Deberás:

* Cargar los datos
* Visualizar gráficamente los datos (x1 respecto a y e x2 respecto a y)
* Dividir los datos en conjunto de entrenamiento y pruebas
* Probar diferentes modelos de regresión polinómica, hasta ver cuál es el mejor degree (y explicarlo gráficamente y por medida del error en los conjuntos de entrenamiento y pruebas), no hace falta que uses validación cruzada. ¿Qué medición has usado para evaluar el modelo y por qué? ¿cuál es el mejor degree?
* Probar ahora con Elastic Net, optimizando los hiperparámetros con búsqueda exhaustiva, ¿ha mejorado con esta regularización? ¿cuáles son los mejores datos para los hiperparámetros?

## EJERCICIO 3

Queremos predecir la calidad del vino (quality) a partir del resto de características.

Carga el conjunto de datos winequality.csv, explora los datos y divide el conjunto de datos en entrenamiento y pruebas (10% para las pruebas).

Crea una pipeline para preparar los datos y aplicar un algoritmo de máquinas de vector soporte, entrena el modelo (guarda el tiempo que tardas en entrenar el modelo) y evalúa el modelo en el conjunto de pruebas (no hace falta que pruebes con diferentes hiperparámetros...)

Crea una pipeline para preparar los datos y aplicar PCA que explique al menos el 95% de la varianza, aplica esta pipeline al conjunto de entrenamiento y al conjunto de pruebas.

Muestra cuantas dimensiones tenías al principio y cuantas después de aplicar PCA.

Aplica a los datos obtenido de aplicar PCA el algoritmo de máquinas de vector soporte, entrena el modelo (guarda el tiempo que tardas en entrenar el modelo) y evalúa el modelo en el conjunto de pruebas (no hace falta que pruebes con diferentes hiperparámetros...)

Compara el error y el tiempo empleado en los dos modelos (sin pca y con pca) e interpreta el resultado.

## EJERCICIO 4

Una empresa de coches ha sacado un nuevo modelo al mercado. Le ha preguntado a una red social quién ha comprado el producto, recopilando los datos de sexo, edad y salario de cada uno de ellos. Ahora queremos construir un modelo que nos permita determinar con estos atributos si la persona comprará el producto o no.

Los datos están en compras.csv

Recopilación y recolección de los datos:

* Obtener los datos
* Explorar los datos con exhaustividad: datos estadísticos, gráficos, correlaciones... e indicar las deducciones a las que llegas
* Dividir los datos en conjunto de entrenamiento y pruebas
* Preparar los datos

Algunas pistas a tener en cuenta (\*\* esto no quiere decir que sólo haya que hacer esto\*\*):

* No usar las características que no sirven
* Tratar los valores nulos (aunque ahora no haya)
* Edad: primeramente convertir en categórica poniendo las siguientes etiquetas y luego aplicar el cambio que procede en las variables categóricas (es para aplicar los diferentes tipos de transformaciones que hay, no porque tenga mucho sentido).
  + 0-20 --> Rango1
  + 20-30 --> Rango 2
  + 30-40 --> Rango 3
  + 40-50 --> Rango 4
  + 50- --> Rango 5
* Salario: aplicar logaritmo (aunque no tenga una cola larga a la derecha) y luego estandarizar

## CRITERIOS DE EVALUACIÓN

|  |  |
| --- | --- |
| **EJERCICIO 1** | **PUNTUACIÓN** |
| **Cargar datos** | 0,15 |
| **Gráficos de distribución de las variables** | 0,15 |
| **Diagramas de caja** | 0,15 |
| **Matriz de correlaciones** | 0,15 |
| **Deducciones** | 0,25 |
| **Dividir en entrenamiento y pruebas estratificando** | 0,15 |
| **Pipeline para preparar los datos** | 0,15 |
| **Pipeline para el modelo** | 0,15 |
| **Validación cruzada** | 0,25 |
| **Explicar medida de validación elegida** | 0,20 |
| **Mejor y peor error** | 0,15 |
| **Optimizar modelo** | 0,25 |
| **Mejor hiperparámetro** | 0,15 |
| **Evaluar en el conjunto de test** | 0,15 |
| **Guardar el modelo** | 0,15 |
| **Importar el modelo** | 0,15 |
| **Predecir nuevos datos desde fichero .py** | 0,15 |
| **Controlar las excepciones** | 0,10 |
|  | **3** |

|  |  |
| --- | --- |
| **EJERCICIO 2** | **PUNTUACIÓN** |
| **Cargar datos** | 0,20 |
| **Gráficos** | 0,20 |
| **Dividir datos en entrenamiento y pruebas** | 0,10 |
| **Pipeline para los modelos de regresión polinomial** | 0,25 |
| **Gráfica para ver el mejor degree** | 0,50 |
| **Errores de entrenamiento y prueba para el mejor degree** | 0,25 |
| **Explicación medida del error elegida** | 0,25 |
| **Pipeline para elastic net** | 0,20 |
| **Definir valores para la optimización de los hiperparámetros** | 0,25 |
| **Búsqueda exhaustiva** | 0,20 |
| **Mejores hiperparámetros** | 0,15 |
| **Evaluación en entrenamiento y prueba para ver si ha mejorado** | 0,15 |
|  | **2,7** |

|  |  |
| --- | --- |
| **EJERCICIO 3** | **PUNTUACIÓN** |
| **Cargar, explorar y dividir los datos** | 0,20 |
| **Pipeline y entrenar modelo sin pca** | 0,25 |
| **Guardar tiempo de entrenamiento** | 0,20 |
| **Evaluar en el conjunto de pruebas** | 0,15 |
| **Pipeline para pca y aplicarlo en entrenamiento y pruebas** | 0,30 |
| **Dimensiones sin pca y con pca** | 0,20 |
| **Entrenar modelo en los datos con pca** | 0,25 |
| **Guardar tiempo de entrenamiento** | 0,20 |
| **Evaluar en el conjunto de pruebas** | 0,15 |
| **Comparar e interpretar los datos** | 0,40 |
|  | **2,3** |

|  |  |
| --- | --- |
| **EJERCICIO 4** | **PUNTUACIÓN** |
| **Obtener los datos** | 0,20 |
| **Explorar los datos** | 0,60 |
| **Dividir los datos en conjunto de entrenamiento y pruebas** | 0,10 |
| **Preparar los datos** | 1,10 |
|  | **2** |