prog_datasci_7_analisis_entrega

April 30, 2022

1 Fundamentos de Programación

1.1 PEC 7 - Enunciado

En este Notebook se encontraréis el conjunto de actividades evaluables como PEC de la asignatura. Veréis que cada una de ellas tiene asociada una puntuación, que indica el peso que tiene la actividad sobre la nota final de la PEC. Adicionalmente, hay un ejercicio opcional, que no tiene puntuación dentro de la PEC, pero que se valora al final del semestre de cara a conceder las matrículas de honor y redondear las notas finales. Podréis sacar la máxima nota de la PAC sin necesidad de hacer este ejercicio. El objetivo de este ejercicio es que sirva como pequeño reto para los estudiantes que quieran profundizar en el contenido de la asignatura.

Veréis que todas las actividades de la PEC tienen una etiqueta, que indica los recursos necesarios para llevarla a cabo. Hay tres posibles etiquetas:

- NM **Sólo materiales**: las herramientas necesarias para realizar la actividad se pueden encontrar en los materiales de la asignatura.
- EG **Consulta externa guiada**: la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, pero el enunciado contiene indicaciones de dónde o cómo encontrar la información adicional necesaria para resolver la actividad.
- EI **Consulta externa independiente**: la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, y el enunciado puede no incluir la descripción de dónde o cómo encontrar esta información adicional. Será necesario que el estudiante busque esta información utilizando los recursos que se han explicado en la asignatura.

Es importante notar que estas etiquetas no indican el nivel de dificultad del ejercicio, sino únicamente la necesidad de consulta de documentación externa para su resolución. Además, recordad que las **etiquetas son informativas**, pero podréis consultar referencias externas en cualquier momento (aunque no se indique explícitamente) o puede ser que podáis hacer una actividad sin consultar ningún tipo de documentación. Por ejemplo, para resolver una actividad que sólo requiera los materiales de la asignatura, puedéis consultar referencias externas si queréis, ya sea tanto para ayudaros en la resolución como para ampliar el conocimiento!

En cuanto a la consulta de documentación externa en la resolución de los ejercicios, recordad **citar siempre la bibliografía utilizada** para resolver cada actividad.

1.1.1 Ejercicio 1

Las enfermedades cardiovasculares (ECV) son la principal causa de muerte en todo el mundo y se cobran aproximadamente 18 millones de vidas cada año, lo que representa casi un tercio de todas las muertes en todo el mundo.

La insuficiencia cardíaca es un evento común causado por las ECV.

La mayoría de las enfermedades cardiovasculares se pueden prevenir abordando los factores de riesgo conductuales, como el tabaquismo, la dieta poco saludable y la obesidad, la inactividad física y el consumo nocivo de alcohol, mediante estrategias que abarquen a toda la población.

Las personas con enfermedades cardiovasculares o con alto riesgo cardiovascular (por la presencia de uno o más factores de riesgo como hipertensión, diabetes, hiperlipidemia o enfermedad ya establecida) necesitan una detección y manejo tempranos en los que un modelo de aprendizaje automático puede ser de gran ayuda.

Veréis que en esta PEC usaremos un conjunto de datos sobre insuficiencia cardíaca, que encontraréis en el fichero `heart_failure_clinical_records_dataset.csv`.

Este conjunto de datos contiene 12 características que pueden usarse para predecir la mortalidad por insuficiencia cardíaca. Estas son:

- **age**: Edad del paciente
- **anaemia:** nivel de hemoglobina del paciente (booleano)
- **creatinine_phosphokinase:** Nivel de la enzima CPK en la sangre (mcg/L)
- **diabetes:** si el paciente tiene diabetes (booleano)
- **ejection_fraction:** porcentaje de sangre que sale del corazón en cada contracción
- **high_blood_pressure:** Si el paciente tiene hipertensión (booleano)
- **platelets:** Recuento de plaquetas en sangre (kiloplaquetas/mL)
- **serum_creatinine:** Nivel de creatinina sérica en la sangre (mg/dL)
- **serum_sodium:** Nivel de sodio sérico en la sangre (mEq/L)
- **sex:** Sexo del paciente
- **smoking:** Si el paciente fuma o no (Booleano)
- **time:** Período de seguimiento (días)
- **DEATH_EVENT:** si el paciente falleció durante el período de seguimiento (booleano)

Codificación para los atributos que tienen valores booleanos: - 0 = Negativo (No) - 1 = Positivo (Sí)

Este conjunto de datos fue publicado en:

- Davide Chicco, Giuseppe Jurman: Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making 20, 16 (2020). (enlace)

a) Carga la tabla en un dataframe. Mustra los nombres de las variables. Muestra los primeros 10 elementos y comprueba que las columnas generados son correctas. Recuerda de tener especial cuidado con definir los separadores decimales y los delimitadores de campo correctos.

NM **(0.5 puntos)**

[207]:
```
# Respuesta
```

b) A partir de aquí:

- Muestra el número de observaciones (filas)
- Muestra el número de observaciones vacías por columna.
- Elimina las filas que tengan un elemento no observado en alguna de sus columnas
- Vuelve a mostrar el número de observaciones restante.

NM **(0.5 puntos)**

[210]:
```
# Respuesta
```

c) Calcula:

- El tanto por ciento de pacientes fallecidos durante el período de seguimiento.

- El promedio, desviación estándar, mínimo, máximo y cuartiles de cada variable.
- Representa en un barplot la variable `DEATH_EVENT`

NM **(1 punto)**

d) calcula la matriz de correlación entre las variables, y representa la matriz de correlación gráficamente con un mapa de calor o `heatmap`. Qué tres variables aparecen más correlacionadas con `DEATH_EVENT`?

NM **(1 punto)**

1.1.2 Ejercicio 2

a) Realizad un test estadístico ttest para responder a la siguiente pregunta: ¿Los pacientes que no sobreviven tienen unos niveles diferentes de creatinina que los que sobreviven ? **(0.5 puntos)**

Hipótesis Alternativa: Los pacientes que no sobreviven no tienen un cambio diferente de la que muestran los que sobreviven.

Hipótesis Nula: Los pacientes que no sobreviven tienen un cambio diferente de la que muestran los que sobreviven.

b) Una vez realizado el test, visualiza gráficamente con un boxplot la diferencia entre los dos grupos, interpreta los valores resultantes y elabora una respuesta. **(0.5 puntos)**

(1 punto) EG

Nota: Consulta este enlace si quieres ampliar conocimientos sobre el test estadístico ttest. Para realizar el test estadístico automáticamente podéis utilizar el método ttest_ind de `Scipy.stats`

[220]:
```
# Respuesta
```

1.1.3 Ejercicio 3

Vamos a hacer una PCA de 2 componentes de los datos, con las variables:
- age - anaemia - creatinine_phosphokinase - diabetes - ejection_fraction - high_blood_pressure - platelets - serum_creatinine - serum_sodium

Para ello:

a) Antes de ajustar el modelo de PCA, tipifica los datos con `sklearn.preprocessing.StandardScaler`. ¿ Qué hace esta función? Calcula el valor medio y la desviación típica de los datos tipificados después de aplicar la función para cada de las variables.

NM **(0.5 puntos)**

[223]:
```
# Respuesta
```

b) Ajusta el modelo de componentes principales sobre los datos tipificados, limitando el número de componentes principales a 2.

NM **(0.5 puntos)**

[226]:
```
# Respuesta
```

c) Muestra los componentes y la varianza explicada de cada uno de ellos. ¿ Que quieren decir los valores obtenidos?

NM **(0.5 puntos)**

[228]:
```
# Respuesta
```

Vemos que casi con dos componente podemos explicar un 27% de la varianza que tenemos.

d) Representa un scatterplot con las proyecciones de los dos componentes transformados, de forma que el color de cada punto indique la condición `DEATH_EVENT`.

NM **(0.5 puntos)**

[2]:
```
# Respuesta
```

e) Muestra una gráfica donde podamos ver como evoluciona la variabilidad explicada, a medida que vamos incrementando hasta 9 componentes. ¿Cuántos componentes necesitamos para explicar más del 90% de la varianza?

NM **(1 punto)**

[231]:
```
# Respuesta
```

1.1.4 Ejercicio 4

En este ejercicio nos gustaría ver si podemos agrupar los pacientes en grupos diferentes según sus características de una manera `no supervisada`. Con esa finalidad, utilizaremos el método `KMeans` que hemos visto en la teoría.

a) Antes de aplicar el KMeans, normaliza los datos originales utilizando la función `StandardScaler`, quitando de los datos la variable `DEATH_EVENT`

NM **(0.5 puntos)**

[ ]:
```
# Respuesta
```

b) Uno de los problemas de los métodos no supervisados es identificar el número óptimo de clusters. Para poder estimar esta número óptimo, se utiliza frecuentemente `el método de Elbow`. Busca información sobre este método y utilízalo. Propón un número óptimo de clusters

EI **(1 punto)**

[ ]:
```
# Respuesta
```

1.1.5 Ejercicio 5

En los ejercicios resueltos hemos visto como usar la regresión logística, y hemos hablado de la regresión lineal múltiple.

En este ejercicio, tendréis que aplicar la regresión logística para construir un modelo predictivo de la variable `DEATH_EVENT` según las características del conjunto de datos.

a) Dividid los datos ya tipificados en conjuntos de entrenamiento (training) y evaluación (test), con un tamaño de la muestra para test del 40%.

NM **(0,5 punto)**

[ ]:
```
# Respuesta
```

b) A continuación, entrenad el modelo con la porción de datos para el entrenamiento y calculad la precisión del modelo. Escribid una breve explicación que indique lo que significaría obtener tal precisión. ¿ Sería útil para predecir la salida o muerte de un paciente (`DEATH_EVENT`)?

NM **(1 punto)**

[ ]:
```
# Respuesta
```

1.1.6 Ejercicio opcional

En este ejercicio, asumiremos que sólo podemos usar estas variables para predecir la salida (`DEATH_EVENT`) del paciente: `ejection_fraction`,`serum_creatinine`,`age`.

Aplicad un clasificador basado en un árbol de decicisión de un máximo de 2 niveles de profundidad para predecir la salida (`DEATH_EVENT`) utilizando las tres variables, empleando un 50% de las muestras de entrenamiento y el 50% de test. Debéis utilizar la función sklearn.tree.DecisionTreeClassifier.

¿Qué valor de precisión obtenemos en un modelo basado en un árbol de decisión? Representa el árbol de decisión y expórtalo a un archivo PDF.

Nota: Tal vez la función tree de sklearn sea de utilidad.

EI

[3]:
```
# Respuesta
```