Tarea 1

INSTRUCCIONES

Estas pruebas están basadas en la filosofía de aprender haciendo y la intención es que tratando de realizar los problemas los estudiantes ejerciten e interioricen los conocimientos teóricos que han adquirido mediante el estudio de los textos y los materiales complementarios. Al tratarse de una prueba evaluable se trata de un trabajo individual, es de este modo, pensando los problemas por su cuenta e investigando como lograrán una buena comprensión de la materia.

Entrega: Las soluciones a los problemas han de ser redactados en un documento realizado mediante una aplicación informática que permita incluir las fórmulas y los desarrollos de los problemas. Existen múltiples opciones como el editor de ecuaciones de Microsoft Word Office, o software libre como LibreOffice. También es posible redactarlas empleando LaTeX. Es recomendable este modo ya que ofrece la mejor presentación para textos científicos y su aprendizaje será de enorme utilidad para aquellos que desarrollan una carrera en Ciencias e Ingeniería. Si deciden tomar esta opción en el curso virtual hay un enlace con un excelente tutorial de LaTeX y una plantilla que pueden usar si lo desean. En ningún caso se permitirá la entrega de documentos redactados a mano y escaneados.

A la plataforma debe subirse un archivo .zip .rar .tar o equivalente en el que estén contenidos la memoria en pdf con el formato adecuado y también la fuente: el archivo .doc o .tex o equivalente del que se ha obtenido el pdf. También debe subir el archivo de LiveScript de Matlab.mlx con el que ha realizado los cálculos y exportado a pdf.

Trabajo individual: Se trata de una prueba calificable y por tanto individual. No se permité el intercambio de información entre los estudiantes, que además iría en detrimento de su aprendizaje. El equipo docente, como es su obligación, velará porque esto sea así en la corrección y utilizando las herramientas antiplagio de las que dispone la institución. En caso de ser detectado plagio se verá obligado a actuar al respecto tomando las medidas pertinentes sobre todos los agentes implicados.

1 PROBLEMAS

Esta primera tarea no está enfocada en el tratamiento en detalle de un problema amplio sino en la resolución de varios ejercicios. Por tanto, no es pertinente seguir el esquema en la memoria de presentar una introducción, luego un desarrollo y finalmente unas conclusiones. Debe limitarse a resolver de forma detallada los problemas, obtener los resultados y las representaciones gráficas pedidas y añadir unos comentarios finales sobre la solución que obtiene cuando sean pertinentes.

En las partes que requieren programación la tarea es guiada, de modo que se dan instrucciones detalladas de como se pueden realizar los cálculos para que pueda implementarlos y aprender a programar este tipo de problemas de forma eficiente.

En esta primera práctica el objetivo es que aprenda a generar conjuntos de datos aleatorios y pueda evaluar cuando los resultados de contrastes estadísticos y regresiones pueden ser válidos o no dependiendo de la incertidumbre inherente a las muestras de datos, que aquí puede controlar.

Es también un objetivo fundamental que comprenda las medidas aciertos y fallos a través de su programación en la regresión logística, ya que son de suma importancia porque su uso se extiende a todo problema de clasificación, no necesariamente binaria.

1. 1 Generación de muestras aleatorias ofertas.

Se trata de realizar el ejercicio ya adelantado en el material de la asignatura, para generar un conjunto de muestras aleatorias de los resultados ficticios de una toma de datos de compras en almacenes donde hay diferentes ofertas.

Programar una simulación y ejecutar sobre ella el test ANOVA. Se trata de seguir el texto base, programando un ejercicio similar al realizado en él, en un Live Script de Matlab.

Simulación:

Genere un conjunto de 500 pares de datos. Cada uno de ellos debe pertenecer a uno de los tres grupos siguientes de forma equiprobable: "offer1","offer2" y "nooffer". El grupo de pertenencia es el primer dato de cada par.

El segundo dato es el tamaño de compra: "purchase_amt". Éste toma un determinado valor con distinta probabilidad dependiendo del grupo al que pertenezca.

1.- "offer1" distribución normal de media 80 y desviación típica 30.

2.- "offer2" distribución normal de media 85 y desviación típica 30.

3.- "nooffer" distribución normal de media 40 y desviación típica 30.

Si alguno de los valores obtenidos en las simulaciones es negativo debe ser sustituido por cero.

Análisis: Realice un test ANOVA para determinar si las ofertas tienen efecto al nivel de significancia de 0.05. Use la función de Matlab y programe el cálculo explícito que implica el uso de las ecuaciones 3-4, 3-5 y 3-6 del texto base.

Comente los resultados

1. 2 Generación de muestras para regresión polinomial y ajuste.

Genere una muestra de $n = 1000$ datos aleatorios uniformemente distribuidos entre $x = 0$ y $x = 10$ que serán la variable explicativa x.

Genere la variable y a partir de la siguiente fórmula:

$$y = 10 + x - 5x^2 + \frac{1}{2}x^3 + \varepsilon$$

donde ε es una variable aleatoria con distribuión normal y desviación típica $\sigma = 20$.

Represente los datos y frente a x y observará una muestra aleatoria con cierta tendencia.

Realice tres ajustes a los datos de polinomios que van de grado 1 a 3. Represente los ajustes que obtiene.

Represente los residuos frente a las predicciones de los modelos.

Recoja los resultados de cada ajuste, parámetros, indicadores de si son significativos o no, etc.

Comente los resultados.

1. 3 Generación de muestras para regresión multivariante y ajuste.

Genere una muestra de $n = 1000$ datos aleatorios uniformemente distribuidos entre $x_1 = 0$ y $x_1 = 10$.

Genere una muestra de $n = 1000$ datos aleatorios uniformemente distribuidos entre $x_2 = 10$ y $x_2 = 20$.

Genere una muestra de $n = 1000$ datos aleatorios uniformemente distribuidos entre $x_3 = 5$ y $x_3 = 10$.

Estas muestras serán la terna de variables explicativas

$$(x_1, x_2, x_3)$$

Cada una de ellas debe ser un vector columna de tamaño 1000 en Matlab. No deben ser ordenadas mediante la función sort.

A partir de ellas genere la variable explicada y mediante la siguiente fórmula:

$$y = 3 + 2x_1 + x_2 - x_3 + \varepsilon$$

donde ε es una variable aleatoria con distribuión normal y desviación típica $\sigma = 1$.

1.- Realice un ajuste de regresión lineal. Debe correr el programa varias veces y observar como los resultados de la regresión (parámetros, intervalos, etc) cambian. Esto es debido a que cada vez partimos de una muestra aleatoria diferente tanto en las variables explicativas como en la explicada a pesar de que su relación funcional se mantenga. Esto es lo que puede suceder en las tomas de datos.

Comente lo que observe.

2.- Realice ahora varias ejecuciones variando la desviación típica de la variable aleatoria ε desde un valor de cero hasta valores altos para ver que sucede con los resultados.

Comente las tendencias que observe.

3.- Realice finalmente una ejecución con el valor de la desviación típica $\sigma = 1$ y represente los residuos frente a la variable predicha y el gráfico qqplot.

Recoja los resultados de este último ajuste, parámetros, indicadores de si son significativos o no, etc.

Comente los resultados.

1. 4 Ejercicio cálculo tpr y fpr

Programar un código que obtenga tpr y fpr

https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Es posible definir cuatro cantidades importantes en la evaluación de todo problema cuya variable respuesta es dicotómica o categórica. En el caso más sencillo de la variable dicotómica nuestro modelo ofrecerá un vector de scores o predicciones que es la probabilidad de que tengamos el valor 1. La repuesta deseable es una variable dicotómica con valores exclusivamente 0 o 1 para ello se utiliza un valor umbral "threshold" por debajo del cual convertimos a nuestro valor predico en un cero y por encima en un 1, obteniendo la respuesta puramente dicotómica. Esta predicción, que depende del umbral debe compararse con la variable respuesta real que conocemos. Para cada caso podemos tener

cuatro resultados:

1.- Respuesta 1 y predicción 1. Verdadero positivo, predicción correcta de un positivo. True positive.

2.- Respuesta 1 y predicción 0. Falso negativo, predicción errónea de un negativo cuando debía ser un positivo. False negative.

3.- Respuesta 0 y predicción 1. Falso positivo, predicción errónea de un positivo cuando debía ser negativo. False positive.

4.- Respuesta 0 y predicción 0. Verdadero negativo, predicción correcta de un negativo. True negative.

Entender como obtener y evaluar el resultado en cada uno de los casos es importante ya que es general para cualquier modelo de variable dicotómica, no únicamente para la regresión logística.

Es posible haciendo uso de las capacidades de Matlab de manejo de matrices obtener estos resultados sin la necesidad de programar ningún bucle for o estructura de decisión if.

Posibles pasos a seguir para programarlo (el estudiante puede adoptar otros y seguir su propia estrategia si así lo desea)

1.- Obtener el vector de predicciones o scores en forma de vector columna con valores en el intervalo [0,1]

2.- Crea un vector fila con diferentes valores de umbral equiespaciados. Estos irán de 0 a 1. Vector thresholds.

3.- Crear una matriz en la que se compare cada valor de score con cada valor de threshold de modo tenga valores 1 o 0 según si superan o no el umbral.

4.- Obtener matrices truepositive, truenegative, falsepositive y falsenegative comparando cada uno de los vectores columna, que corresponden a un valor de threshold con el vector de datos de la variable respuesta.

5.- Obtener para cada valor de threshold el número total de cada una de las cuatro cantidades antes indicadas. Se puede usar la suma en las columnas.

6.- Se puede comprobar que para cada valor de threshold la suma de las cuatro será igual al número total de casos.

7.- Calcular los valores de tpr y fpr y representarlos para el caso que nos ocupa.

Obténganse las curvas ROC y TPR/FPR que para el conjunto de datos Churn.

Comente los resultados.

Es preciso programarlo sin hacer uso de la función integrada en Matlab aunque si lo desearan podrían utilizarla para comprobar sus resutados teniendo presente que la salida de este comando es algo más compleja y difícil de interpretar:

https://www.mathworks.com/help/deeplearning/ref/roc.html

NOTA: Matlab puede hacer directamente todas las comparaciones (u otras operaciones) posibles entre un vector columna y un vector fila obteniéndose una matriz con ellas.