Universidad del Norte
Departamento de Matemáticas y Estadística
Profesor: Dr. Lihki Rubio
Parcial de Machine Learning
Alumno:_____

Duración: 6.5 horas Fecha: / /

Instrucciones:

- Está prohibido el uso de celulares y relojes inteligentes. Para el caso de los ejercicios teóricos, a esto se agrega, hojas de formulas, cuadernos y calculadora. Al estudiante sorprendido con este tipo de elementos, le será anulada la evaluación y será reportada al respectivo coordinador como plagio.
- Los ejercicios teóricos deben ser desarrollados de forma tradicional, solo con papel y lápiz. Solo podrá usar su computadora una vez que haya finalizado con el/los ejercicios teóricos seleccionados.
- Los ejercicios computacionales debe realizarlos en su computadora personal. Debe grabar su pantalla desde que usted inicia la solución, y en frente del profesor, finalizar la grabación cuando este listo para realizar la entrega final.
- El profesor encargado, registrará los tiempos de inicio y finalización. Para grabar su pantalla puede usar por ejemplo **ActivePresenter**. En caso de que la duración de la grabación no coincida con las restas entre el tiempo final e inicial registrado por el profesor, la evaluación será anulada y será reportada al respectivo coordinador como plagio.

Ejercicio 1

Regresión Ridge (2 puntos): Consideremos el modelo de regresión lineal

$$y_i = \beta_0 + \beta_1 \cdot x_{i1} + \beta_2 \cdot x_{i2} + \cdots + \beta_p \cdot x_{ip} + \varepsilon_i, \ i = 1, 2, \ldots, n, \tag{1}$$

basado en los datos observados $\{(y_i, x_{i1}, x_{i2}, \ldots, x_{ip}) : \ i = 1, 2, \ldots, n\}$ para la variable respuesta y y p variables predictoras $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)$. Siga los siguientes pasos para obtener la estimación $\hat{\boldsymbol{\beta}}_R$ de la regresión ridge

a) Para evitar problemas de multicolinealidad, considere estandarizar los datos:

$$z_{ij} = (x_{ij} - \overline{x}_j)/s_j$$

siendo $\overline{x}_j, s_j$ la media y la desviacion de la j-ésima característica respectivamente.

b) Reescriba (1) en su forma matricial: $\boldsymbol{y} = \vec{A} + Z\vec{B} + \varepsilon$. ¿Quienes son $\vec{A}, \vec{B}$?.

c) Defina el estimador ridge $S_\lambda(\beta_0^\star, \boldsymbol{\beta}_s)$ y especifique cual es el término de regularización.

d) Resuelva el problema de minimización calculando las siguientes derivadas parciales y resolviendo el sistema resultante. Debe demostrar antes que: $\partial_{\boldsymbol{x}}(\boldsymbol{x}^T A\boldsymbol{x}) = 2A\boldsymbol{x}$

$$\frac{\partial S_\lambda(\beta_0^\star, \boldsymbol{\beta}_s)}{\partial \beta_0^\star} = 0, \quad \frac{\partial S_\lambda(\beta_0^\star, \boldsymbol{\beta}_s)}{\partial \boldsymbol{\beta}_s} = 0. \tag{2}$$

e) Por medio de estandarización demuestre que $\beta_0^\star = 0$ y redefina la formulación matricial del ítem b). Basado en esta representación, defina el operador a minimizar $S_\lambda(\boldsymbol{\beta})$ y calcule el estimador ridge asociado $\hat{\boldsymbol{\beta}}_R$. Demuestre que su valor esperado es $E(\hat{\boldsymbol{\beta}}_R) = (X^T X + \lambda \boldsymbol{I}_p)^{-1} X^T X \boldsymbol{\beta}$.

Ejercicio 2

Regresión de Vectores de Soporte (1.5 puntos): Utilizando la función de pérdida de Huber aproximada o de pérdida lineal ε-insensibe definida por:

$$\mathcal{L}(y, f(\boldsymbol{x})) = \begin{cases} |y - f(\boldsymbol{x})| - \varepsilon & \text{si } |y - f(\boldsymbol{x})| > \varepsilon \\ 0 & \text{si } |y - f(\boldsymbol{x})| \leq \varepsilon \end{cases}, \tag{3}$$

aproxime la tarea de regresión $y = g(\boldsymbol{x}) + \eta_n$, usando un model lineal de la forma $f(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x} + \theta_0$, el cual cuantifica el error de desajuste del modelo,

- Verifique que la tarea de minimización para el correspondiente costo empírico, regularizado por la norma de $\boldsymbol{\theta}$, está proyectado en términos de las variables auxiliares como:

$$\begin{cases} \text{minimizar} & J(\boldsymbol{\theta}, \theta_0, \xi, \tilde{\xi}) = \frac{1}{2}\|\boldsymbol{\theta}\|^2 + C\left(\sum_{n=1}^{N} \xi_n + \sum_{n=1}^{N} \tilde{\xi}_n\right), \\ \text{sujeto a} & y_n - \boldsymbol{\theta}^T \boldsymbol{x}_n - \theta_0 \leq \varepsilon + \tilde{\xi}_n, \quad n = 1, 2, \ldots, N, \\ & \boldsymbol{\theta}^T \boldsymbol{x}_n + \theta_0 - y_n \leq \varepsilon + \xi_n, \quad n = 1, 2, \ldots, N, \\ & \tilde{\xi}_n \geq 0, \quad \tilde{\xi}_n \geq 0, \ n = 1, 2, \ldots, N. \end{cases} \tag{4}$$

- Defina el Lagrangiano $L(\boldsymbol{\theta}, \theta_0, \tilde{\xi}, \xi, \lambda, \mu)$ y calcule los multiplicadores de Lagrange asociados a la tarea de optimización, por medio del cálculo de las siguientes derivadas:

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = \mathbf{0}, \ \frac{\partial L}{\partial \theta_0} = 0, \ \frac{\partial L}{\partial \tilde{\xi}_n} = 0, \ \frac{\partial L}{\partial \xi_n} = 0. \tag{5}$$

- Encuentre la representación dual de Walfe asociada utilizando la estimación $\hat{\boldsymbol{\theta}}$ y reescriba el problema en terminos de los multiplcadores de Lagrange. Tenga en cuenta que $\lambda = \tilde{\lambda}_n - \lambda_n$ y $b = A\boldsymbol{\theta} = \boldsymbol{\theta}^T \boldsymbol{x}_n$, y la representación de Walfe

$$\begin{aligned} \text{minimizar:} \quad & \frac{1}{2}\boldsymbol{\theta}^T \boldsymbol{\theta} - \lambda^T (A\boldsymbol{\theta} - \boldsymbol{b}), \\ \text{sujeto a:} \quad & \theta - A^T \lambda = 0. \end{aligned}$$

- Indique detalladamente, cuales son los multiplicadores correspondientes a los vectores de soporte. Basados en los espacios de Hilbert con kernel reproductor, defina que es el kernel trick. Use el kernel trick verique que la predicción de vectores de soporte está dada por:

$$\hat{y}(\boldsymbol{x}) = \sum_{n=1}^{N_s} (\tilde{\lambda}_n - \tilde{\lambda}_n) \kappa(\boldsymbol{x}, \boldsymbol{x}_n) + \hat{\theta}_0, \quad \text{Prediccion SVR.} \tag{6}$$

Ejercicio 3

- **Multilayer Perceptrón (Backpropagation) (1.5 puntos)**. Sea $(\boldsymbol{y}_n, \boldsymbol{x}_n), n = 1, 2, \ldots, N$, un conjunto de muestras de entrenamiento. Supongamos que la red consta de L capas, $L - 1$ capas ocultas y una capa de salida. Cada capa consta de $k_r, r = 1, 2, \ldots, L$, neuronas, donde $k_0 = l$, y l es la dimensionalidad del espacio de características de entrada. Así, los vectores de salida (objetivo/deseado) son

$$\boldsymbol{y}_n = [y_{n1}, y_{n2}, \ldots, y_{nk_L}]^T \in \mathbb{R}^{K_L}, \quad n = 1, 2, \ldots, N.$$

- Sea $\boldsymbol{\theta}_j^r$ denota el vector de los pesos sinápsis asociados a la j-th neurona de la r-th capa, con

$j = 1, 2, \ldots, k_r$ y $r = 1, 2, \ldots, L$, donde el término de sesgo se incluye en $\boldsymbol{\theta}_j^r$, es decir,

$$\boldsymbol{\theta}_j^r := [\theta_{j0}^r, \theta_{j1}^r, \ldots, \theta_{jk_{r-1}}^r]^T. \tag{7}$$

- El paso iterativo básico para el esquema de gradiente descendiente se escribe como

$$\boldsymbol{\theta}_j^r(\text{new}) = \boldsymbol{\theta}_j^r(old) + \Delta\boldsymbol{\theta}_j^r, \quad \text{donde} \quad \Delta\boldsymbol{\theta}_j^r := -\mu \left.\frac{\partial J}{\partial \boldsymbol{\theta}_j^r}\right|_{\boldsymbol{\theta}_j^r(old)},$$

  μ es el tamaño de paso definido por el usuario y J denota la función de coste.

- Considere la salida y_k^r de la neurona k en la capa r. Entonces tenemos

$$y_k^r = f(\boldsymbol{\theta}_k^{r^T} \boldsymbol{y}^{r-1}), \quad k = 1, 2, \ldots, k_r,$$

  donde $\boldsymbol{y}^{r-1}$ es el vector (ampliado) que comprende todas las salidas de la capa anterior, $r-1$, y f denota la no-linealidad.

- El algoritmo *backpropagation* considera la salida de la j-th neurona de la última capa, dada por

$$\hat{y}_j := y_j^L = f(\boldsymbol{\theta}_j^{L^T} \boldsymbol{y}^{L-1}).$$

- Para la derivación detallada del algoritmo *backpropagation*, adoptamos la función de pérdida del error cuadrático

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} J_n(\boldsymbol{\theta}) \quad \text{y} \quad J_n(\boldsymbol{\theta}) = \frac{1}{2}\sum_{k=1}^{k_L} (\hat{y}_{nk} - y_{nk})^2,$$

  donde $\hat{y}_{nk},\ k = 1, 2, \ldots, k_L$, son las estimaciones proporcionadas en los correspondientes nodos de salida de la red. Las consideraremos como los elementos de un vector correspondiente, $\hat{\boldsymbol{y}}_n$.

- Sea z_{nj}^r la salida del combinador lineal de la j-th neurona en la capa r en el instante de tiempo n, cuando se aplica el patrón $\boldsymbol{x}_n$ en los nodos de entrada. Entonces podemos escribir

$$z_{nj}^r = \sum_{m=1}^{k_{r-1}} \theta_{jm}^r y_{nm}^{r-1} + \theta_{j0}^r = \sum_{m=0}^{k_{r-1}} \theta_{jm}^r y_{nm}^{r-1} = \boldsymbol{\theta}_j^{r^T} \boldsymbol{y}_n^{r-1},$$

  donde $\boldsymbol{y}_n^{r-1} := [1, y_{n1}^{r-1}, \ldots, y_{nk_{r-1}}^{r-1}]^T$, y $y_{n0}^r \equiv 1,\ \forall\ r, n$ y $\boldsymbol{\theta}_j^r$ ha sido definido en la Ecuación 7.

- *Demuestre que*

$$\Delta\boldsymbol{\theta}_j^r = -\mu \sum_{n=1}^{N} \delta_{nj}^r \boldsymbol{y}_n^{r-1}, \quad r = 1, 2, \ldots, L, \text{ donde } \delta_{nj}^r := \frac{\partial J_n}{\partial z_{nj}^r}.$$

- *Demuestre que*, para $r = L$, usando la función de pérdida $J_n = \frac{1}{2}\sum_{k=1}^{k_L} \left(f(z_{nk}^L) - y_{nk}\right)^2$ se tiene que

$$\delta_{nj}^L = (\hat{y}_{nj} - y_{nj}) f'(z_{nj}^L) = e_{nj} f'(z_{nj}^L), \quad j = 1, 2, \ldots, k_L,$$

  donde f' denota la derivada de f y e_{nj} es el error asociado con el j-th output en el tiempo n.

- *Demuestre que*, para $r < L$, debido a la dependencia sucesiva entre las capas, el valor de z_{nj}^{r-1} influye en todos los valores $z_{nk}^r,\ k = 1, 2, \ldots, k_r$, de la capa siguiente. Empleando la regla de la cadena, para $r = L, L-1, \ldots, 2$, se tiene que

$$\delta_{nj}^{r-1} = \left(\sum_{k=1}^{k_r} \delta_{nk}^r \theta_{kj}^r\right) f'(z_{nj}^{r-1}) := e_{nj}^{r-1} f'(z_{nj}^{r-1})$$

- Dado que f es la función *sigmoid*, $f(z) = 1/(1 + \exp(-az))$, demuestre que

$$f'(z) = a f(z)(1 - f(z)).$$

Ejercicio 4

Análisis Exploratorio de Datos (1.5 puntos). Dado los siguientes conjuntos de datos: Wind Speed y Fraud Detection, realizar un análisis exploratorio de datos el cual incluya lo siguiente:

- Descripción de tipos de variables, *reducción de nombres extensos en columnas*, calcular *número de observaciones, media, desviación estándar, mínimo, máximo, cuartiles*, realizar conteo de *datos faltantes y su porcentaje*, *histograma o diagrama de barras para la variable respuesta e independientes según corresponda*, seleccionar un mínimo de 4 variables independientes. Análisis de *simetría, datos atípicos y dispersión, etc,...* por medio de boxplot(). *Análisis bivariado*. Trazado de scatterplot() y regplot() para un mínimo de 4 pares de variables explicativas. En cada figura *agregar un análisis y descripción*. Para el conjunto de datos de *detección de fraude* hacer un merge entre tablas basado en *TransactionID*. Para esto, debe usar la función merge(). Esto es: pd.merge(train_transaction, train_identity, on='TransactionID', how='left'). Haga lo mismo para el *conjunto de prueba, el cual debe usar para evaluar el modelo final.*

- Según corresponda, realizar imputación de datos faltantes con la mediana (ver impute()).Realizar reducción de dimensionalidad por medio de eliminiación de columnas altamente correlacionadas usando *Variance Inflation Factor (VIF)*. Para esto se recomienda usar la siguiente librería variance_inflation_factor(). Un $VIF \geq 5$ indica alta multicolinealidad entre la correspondiente variable independiente y las demás variables. *Recomendación: Eliminar una columna a la vez. Aquella con el máximo* $VIF \geq 5$. *Luego, para el nuevo pandas, calcular nuevamente VIF e identificar nuevas columnas con* $VIF \geq 5$ *máximo, y así sucesivamente hasta obtener solo valores de* $VIF < 5$. Según corresponda, variables categóricas deben previamente codificarse usando por ejemplo OneHotEncoder(). Pueden mantener las variables categóricas antes de la codificación previa al entrenamiento del modelo y *reducir multicolinealidad usando la prueba* chi2_contingency().

Ejercicio 5

Modelos de Clasificación (1.5 puntos). Considere el conjunto de datos Fraud Detection. Implemente la versión de clasificación para cada uno de los modelos estudiados en clases, y prediga la variable respuesta *isFraud*. Construir una tabla de error que contenga las métricas usuales de clasificación: *precission, recall,* f_1*-score, AUC*. Además, agregue *matrices de confusión* (ver confusion_matrix) y *curvas ROC* (ver plot_roc). Puede utilizar la librería GridSearchCV y Pipeline para evaluar cada modelo. Verifique que la validación cruzada seleccionada es la adecuada, y justifíquelo. *Utilice la métrica AUC, para seleccionar el mejor modelo de clasificación (maximizar AUC)*. Los resultados deben estar registrados en una tabla de error (ver Tabla 1) que resuma cada score obtenido por modelo implementado.

Modelo	*precission*	*recall*	f_1*-score*	*AUC*
K-NN	···	···	···	···
Ridge	···	···	···	···
Lasso	···	···	···	···
Naive Bayes	···	···	···	···
XGBoost	···	···	···	···
SVM	···	···	···	···
MLP	···	···	···	···

Cuadro 1: Modelo de clasificación para detección de fraude.

Ejercicio 6

Modelos de Regresión (2 puntos). Considere el conjunto de datos Wind Speed. Implemente la versión de regresión de cada uno de los modelos estudiados en clases, para *predecir velocidad del viento horaria (VENTO, VELOCIDADE HORARIA (m/s))* en el conjunto de datos suministrado. Construir una *tabla de error* con las métricas usuales de regresión, *MAPE, RMSE, R2* (ver Table 2). Realice *particiones de entrenamiento y validación*, con base en lo descrito en la Figura 1. Estas particiones siguen la tendencia de la *velocidad del viento*. *Utilice la métrica RMSE* en la evaluación y validación,

para seleccionar el mejor modelo de regresión. El *pliegue de validación en cada partición, debe estar siempre ubicado en el porcentaje final de cada partición*, debido a que el tiempo es fundamental en dichas predicciones.

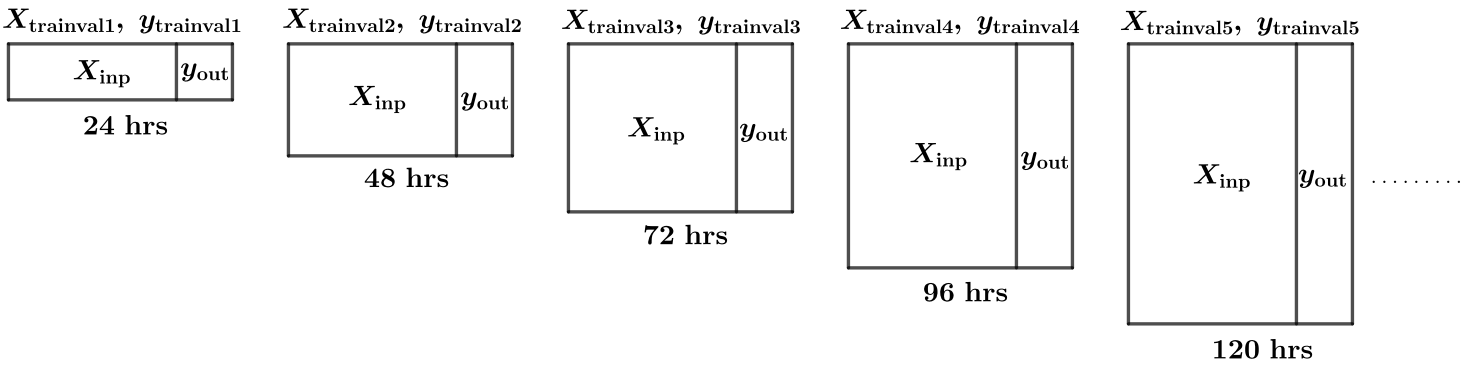

Figura 1: Particiones de entrenamiento y validación. Modelo de regresión.

Modelo	*MAPE*	*RMSE*	R^2
K-NN	···	···	···
Ridge	···	···	···
Lasso	···	···	···
XGBoost	···	···	···
SVM	···	···	···
MLP	···	···	···

Cuadro 2: Modelo de regresión para velocidad del viento.

Diccionario de variables

Detección de fraude

Los datos proceden de transacciones reales de *comercio electrónico de Vesta* y contienen una amplia gama de características, desde el tipo de dispositivo hasta las características del producto. El objetivo principal es mejorar la eficacia de las alertas de transacciones fraudulentas para millones de personas en todo el mundo, ayudando a cientos de miles de empresas a reducir sus pérdidas por fraude y aumentar sus ingresos. Y, por supuesto, ahorrará a muchas personas la molestia de los falsos positivos.

- *TransactionDT*: Intervalo de tiempo a partir de una fecha y hora de referencia
- *TransactionAMT*: Importe del pago de la transacción en USD
- *ProductCD*: Código de producto, el producto de cada transacción
- *card1 - card6*: Información de la tarjeta de pago, como tipo de tarjeta, categoría de tarjeta, banco emisor, país, etc.
- *addr*: Dirección
- *dist*: Distancia
- *P_ and (R_) emaildomain*: Dominio de correo electrónico del comprador y del destinatario
- *C1-C14*: Recuento, cuántas direcciones se encuentran asociadas a la tarjeta de pago, etc. El significado real está codificado.

- *D1-D15*: Intervalo de tiempo, como los días transcurridos entre la transacción anterior, etc.
- *M1-M9*: Coinciden, como los nombres en la tarjeta y la dirección, etc.
- *Vxxx*: Vesta ofrece una gran variedad de funciones, como la clasificación, el recuento y otras relaciones entre entidades.
- *DeviceType*: Codificada. Información de identidad o conexión de red (IP, ISP, Proxy, etc) o firma digital
- *DeviceInfo*: Codificada. Información de identidad o conexión de red (IP, ISP, Proxy, etc) o firma digital
- *id_12 - id_38*: Codificada. Información de identidad o conexión de red (IP, ISP, Proxy, etc) o firma digital

Velocidad del viento

El pronostico de la velocidad del viento es fundamental, sobre todo por sus implicaciones en: *seguridad en la aviación y la navegación, generación de energía eólica, agricultura, construcción, meteorología, recreación y deporte*. Los datos suministrados, reportan diferentes mediciones que pueden explicar y permitir realizar la predicción de la velocidad del viento. Suponga que las *mediciones presentadas, son obtenidas cada 24 hrs* (ver Wind Speed). Además, suponga que desea *pronosticar, cual será la la velocidad del viento, durante las próximas 24 hrs, fuera de la muestra.* El objetivo principal es, *identificar que cantidad de energía eólica se puede generar durante este tiempo (24 hrs), para posteriormente, poder comercializarla* a empresas que producen por ejemplo hidrógeno verde.

- *HORA (UTC)*: Hora
- *VENTO, DIREï¿½ï¿½O HORARIA (gr) (ï¿½ (gr))*: Dirección del viento horaria
- *VENTO, VELOCIDADE HORARIA (m/s)*: Velocidad horario del viento (m/s)
- *UMIDADE REL. MAX. NA HORA ANT. (AUT) (%)*: Humedad rel. máx. hora anterior (AUT) (%)
- *UMIDADE REL. MIN. NA HORA ANT. (AUT) (%)*: Humedad rel. mín. hora anterior (AUT) (%)
- *TEMPERATURA Mï¿½XIMA NA HORA ANT. (AUT) (ï¿½C)*: Temperatura máx. hora anterior (AUT) (°C)
- *TEMPERATURA Mï¿½NIMA NA HORA ANT. (AUT) (ï¿½C)*: Temperatura mín. hora anterior (AUT) (°C)
- *UMIDADE RELATIVA DO AR, HORARIA (%)*: Humedad relativa horaria (%)
- *PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO, HORARIA (mB)*: Presión atmosférica a nivel de estación, horaria (mB)
- *PRECIPITAï¿½ï¿½O TOTAL, HORï¿½RIO (mm)*: Precipitación total por hora (mm)
- *VENTO, RAJADA MAXIMA (m/s)*: Máxima ráfaga de viento (m/s)
- *PRESSï¿½O ATMOSFERICA MAX.NA HORA ANT. (AUT) (mB)*: Presión atmosférica máx. hora anterior (AUT) (mB)
- *PRESSï¿½O ATMOSFERICA MIN.NA HORA ANT. (AUT) (mB)*: Presión atmosférica mín. hora anterior (AUT) (mB)