Pontificia Universidad Católica de Chile
Facultad de Ciencias Económicas y Administrativas

Segundo Semestre 2022

Curso : Machine Learning para Negocios
Sigla : EAA3707
Profesores : M Ignacia Vicuña

Tarea 2

Pregunta 1:

Con los datos `Social_Network_Ads.csv` que utilizamos en clases, se construyeron dos modelos de para predecir la probabilidad de compra de un producto en particular, en función de su edad, salario estimado y género. El primer modelo fue construido usando el algoritmo knn y el segundo SVM. El archivo `predict.csv` contiene la probabilidad predicha de ambos modelos junto con la variable respuesta de la muestra de validación. Existen diferentes métricas para comparar el nivel predictivo de los modelos, y los más usados son el `accurancy` y el `AUC`. A continuación veremos como calcular el AUC y su interpretación.
La curva de ROC (Receiver Operating Characteristic Curve) es una representación gráfica del rendimiento de un clasificador binario. Se crea trazando la tasa de verdaderos positivos (sensitividad) frente a la tasa de falsos positivos (1-especificidad) en varios valores de corte o valores de umbral. El área bajo la curva (AUC), es una métrica de rendimiento que mientras mayor sea el área bajo la curva, mayor será el poder de predicción del modelo.Cuanto más cerca esté el área de 0.5, menos preciso será el modelo, así el poder predictivo de un clasificador, se puede medir utilizando la siguiente regla de decisión:

AUC	calidad ajuste
0.9-1.0	Excelente
0.8 -0.9	Muy buena
0.7-0.8	Buena
0.6 - 0.7	Satisfactoria
0.5 -0.6	Insatisfactoria

Se le pide a Ud. construir la curva de ROC de ambos clasificadores, realizando los siguientes pasos:

1. Modifique el nivel de referencia de la variable respuesta, ya que por defecto R considera el menor valor como clase de referencia. Lo anterior se puede hacer con el siguiente código:

   ```
   datos = datos%>% mutate(Purchased = factor(Purchased,levels=c(1,0)))
   ```

2. Ordene el dataframe de manera ascendente según la probabilidad predicha por el clasificador $h()$.

3. Cree el vector `thresholds` que será igual a los valores predichos utilizando el clasificador $h()$. Elimine los valores repetidos y agregue el valor 0 y 1 si no está considerado.

4. Para cada valor t del vector `thresholds`, calcule la sensitividad y la especificidad del clasificador $h()$ a partir de la siguiente regla de decisión:

   - Las observaciones x donde $h(x) \geq t$ se consideran como positivas, $\hat{y} = 1$.
   - Las observaciones tales que $h(x) < t$ se consideran en la clase negativa, $\hat{y} = 0$.

   Puede ser de utilidad usar la función `specificity()` y `sensitivity()` para el cálculo de la sensitividad y especificidad.

5. Grafique la tasa de falsos positivos (eje x) versus la tasa de verdaderos positivos (eje y) para cada uno de los clasificadores.

6. `Tidymodels` tiene implementado el cáculo de la curva de ROC con la función `roc_curve( )` y el cálculo del área bajo la curva `roc_auc( )`. Utilice ambas funciones y compare con lo obtenido anteriormente.

Pregunta 2:

La base de datos MNIST es una base de datos de dígitos escritos a mano que se utiliza habitualmente para entrenar diversos sistemas de procesamiento de imágenes. El conjunto de datos contiene 60.000 pequeñas imágenes cuadradas de 28×28 pixeles en escala de grises de dígitos simples escritos a mano entre 0 y 9.
El objetivo será construir un clasificador utilizando el algortimo de Support Vector Machine para predecir el valor del dígito en base a las imágenes escritas a mano. Para la ilustración de este ejemplo, utilizaremos una muestra del conjunto de datos. Para ello realice los siguiente:

1. Descargue los datos de la librería "keras"de R, y extraiga una muestra de tamaño $n = 2000$, que será el conjunto de datos que utilizaremos. Debe instalar previamente la librería `keras` y posteriormente ejecutar el siguiente código:

```
library(keras)
mnist =  dataset_mnist()
train_images =  mnist$train$x
train_labels  =  mnist$train$y

set.seed(123)

index = sample(1:60000,2000, replace = FALSE)
sample_images  =  data_images[index,,]
sample_y       =  data_labels[index]
```

2. Utilice el siguiente código para visualizar la primera imagen del data set:

```
plot(as.raster(sample_images[1,,],max=255))
```

   ¿Qué números escrito a mano contiene las primeras 5 imágenes del conjunto de datos `sample_images`?

3. Cree un datafame que contenga las imágenes y la variable respuesta. Luego utilice la semilla set.sed(3707) y realice una partición al 90 % para los datos de entrenamiento.

4. Ajuste un SVM lineal con margen suave a los datos de entrenamiento. Realice validación cruzada para determinar el valor óptimo del parámetro C. Utilice $k = 5$ folds, grilla aleatoria de 10 valores y seleccione según la métrica `roc_auc`.

5. Ajuste un SVM no lineal utilizando el kernel RBF a los datos de entrenamiento. Realice validación cruzada para determinar el valor óptimo de los parámetros C y σ. Utilice $k = 5$ folds, grilla aleatoria de 10 valores y seleccione según la métrica `roc_auc`.

6. Para cada uno de los ajustes de SVM lineal y no lineal, entregue la matriz de confusión y calcule la tasa de clasificación errónea para el conjunto de prueba. Comente los resultados.

Pregunta 3:

Según la Organización Mundial de la Salud (OMS), los accidentes cerebrovasculares son la segunda causa de muerte en el mundo, responsable de aproximadamente el 11 % del total de fallecimientos. A partir de la base de datos `stroke.csv` se desea construir un modelo para predecir si un paciente tiene probabilidades de sufrir un accidente cerebrovascular, en función de variables predictoras como el género, la edad, tipo de trabajo, índice de masa corporal, hábito de fumar, entre otras variables. La Tabla 1 contiene la descripción de todas las variables que contiene el conjunto de datos.

(a) Inspeccione la estructura de los datos con el comando `str( )` y verifique si las variables numéricas y categóricas están en el formato correcto. De no serlo, modifíquelas a numéricas y factor.

(b) ¿Qué porcentaje de pacientes no revela el status de fumador? ¿Qué sugiere hacer con ese grupo de pacientes para el modelamiento?

(c) Sin excluir los valores missing, separe la data en conjunto de entrenamiento (80 %) y muestra de validación (20 %). Utilice la variable stroke para estratificar el muestreo.

(d) Los árboles de decisión pueden manejar los valores faltantes mediante el uso de predictores sustitutos, considerando la categoría missing como un nivel del factor. Ajuste un árbol de clasificación a la muestra de entrenmiento, considerando los siguientes valores de los hiperparámetros: `cost_complexity = 0`, `tree_depth = 20` , `min_n = 15`, para obtener un árbol "grande".

(e) Grafique el árbol de clasificación obtenido e interprete los resultados. ¿Cuál es la tasa de error del conjunto test?

(f) Realice 10-fold validación para determinar los valores óptimos de los hiperparámetros: `cost_complexity()` y `tree_depth()`. Para ello utilice una grilla regular con el rango de cada hiperparámetro que trae por defecto y utilice `levels = 5`. Seleccione el conjunto de parámetros que tenga mayor área bajo la curva (AUC).

(g) Con los valores encontrados, pode el árbol y calcule la tasa de error del conjunto test. ¿Mejora la poda la tasa de error en el conjunto de pruebas? Grafique el árbol podado e interprete los resultados.

(h) Calcule la importancia de la variable para cada predictor para el árbol podado.

(i) Utilice el enfoque de bagging tree para construir un modelo de clasificación. ¿Qué tasa de error de prueba se obtiene? ¿Qué variables son las más importantes?. Comente los resultados.

(j) Utilice Random Forests para construir un modelo de clasificación. Obtenga el error de la prueba, la importancia y discuta el efecto de mtry en la tasa de error. Comente los resultados.

Tabla 1: Descripción Variables del dataframe stroke

Variable	Description
`id`	unique identifier
`gender`	"Male", "Female" or "Other"
`age`	age of the patient
`hypertension`	0 if the patient doesn't have hypertension, 1 if the patient has hypertension
`heart_disease`	0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
`ever_married`	"No" or "Yes"
`work_type`	"children", "Govt_jov", "Never_worked", "Private" or "Self-employed".
`Residence_type`	"Rural" or "Urban"
`avg_glucose_level`	average glucose level in blood
`bmi`	body mass index
`smoking_status`	"formerly smoked", "never smoked", "smokes" or "Unknown"
`stroke`	1 if the patient had a stroke or 0 if not