APRENDIZAJE AUTOMÁTICO AVANZADO

INFORME TÉCNICO UNIDAD II – APRENDIZAJE SUPERVISADO

PRESENTADO POR

Daniel José Rueda Lobato

Diego Andrés Valderrama Laverde

Catalina Piedrahita Jaramillo

PROFESOR:

José Lisandro Aguilar Castro

UNIVERSIDAD EAFIT

MEDELLÍN

SEPTIEMBRE 2020

1. OBJETIVO DE LA ITERACIÓN

Analizar diferentes técnicas de aprendizaje supervisado para seleccionar de acuerdo con las medidas de evaluación un modelo que determine si un individuo gana más de 50 mil dólares al año a partir de características demográficas como en el nivel de educación, la edad, el género, la ocupación, etc., con base en datos extraídos del Censo de 1994 de Estados Unidos

2. CONTEXTUALIZACIÓN DEL PROBLEMA

El problema o tarea consiste en utilizar un algoritmo de Aprendizaje automático supervisado para clasificar cuáles individuos ganan más de 50 mil dólares al año y cuáles no a partir de sus características demográficas. La base de datos utilizada proviene de Kaggle y es una extracción del Censo 1994 de Estados Unidos.

3. DISEÑO DEL MODELO

La base de datos es de corte transversal de individuos y cuenta con 15 variables descriptoras, de las cuales 6 son cuantitativas y 9 son cualitativas, y 48 842 registros. La variable objetivo es *income*, que tiene dos etiquetas: una si gana más de 50 mil dólares al año, y otra si no. La Tabla 1 muestra la caracterización de la base de datos.

Para realizar la tarea de clasificación utilizando un algoritmo de aprendizaje automático es necesario antes realizar un proceso de ingeniería de descriptores en la base de datos. La ingeniería de descriptores es importante porque permite eliminar el ruido en los datos, mejorar el rendimiento de los algoritmos de *Machine Learning*, tanto en la velocidad de aprendizaje, como en la exactitud y el costo computacional.

El procedimiento de ingeniería de descriptores empezó con la identificación de datos faltantes o *missing values*. La cantidad de datos faltantes en la base no es significativa, por lo que se optó por hacer una imputación. Específicamente, como los datos perdidos se encontraban en variables categóricas, se imputó la etiqueta más frecuente, es decir, la moda.

Tabla 1. Caracterización de la base de datos

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   age              48842 non-null  int64
 1   capital-gain     48842 non-null  int64
 2   capital-loss     48842 non-null  int64
 3   hours-per-week   48842 non-null  int64
 4   educational-num  48842 non-null  int64
 5   workclass        48842 non-null  object
 6   fnlwgt           48842 non-null  int64
 7   education        48842 non-null  object
 8   marital-status   48842 non-null  object
 9   occupation       48842 non-null  object
 10  relationship     48842 non-null  object
 11  race             48842 non-null  object
 12  gender           48842 non-null  object
 13  native-country   48842 non-null  object
 14  income           48842 non-null  object
dtypes: int64(6), object(9)
memory usage: 5.6+ MB
```

A continuación, se procedió a realizar un análisis exploratorio de los datos utilizando estadística descriptiva. La Tabla 2 muestra las estadísticas descriptivas para las variables cuantitativas en la base de datos, mientras que la Tabla 3 muestra las frecuencias, entre otras estadísticas de las variables cualitativas. Los descriptores de *capital-gain* y *capital-loss* fueron unidos en un solo descriptor llamado *capital_total*, siendo la diferencia entre ambos.

Tabla 2. Estadísticas descriptivas de las variables cuantitativas

	age	capital-gain	capital-loss	hours-per-week	educational-num
count	48842.000000	48842.000000	48842.000000	48842.000000	48842.000000
mean	38.643585	1079.067626	87.502314	40.422382	10.078089
std	13.710510	7452.019058	403.004552	12.391444	2.570973
min	17.000000	0.000000	0.000000	1.000000	1.000000
25%	28.000000	0.000000	0.000000	40.000000	9.000000
50%	37.000000	0.000000	0.000000	40.000000	10.000000
75%	48.000000	0.000000	0.000000	45.000000	12.000000
max	90.000000	99999.000000	4356.000000	99.000000	16.000000

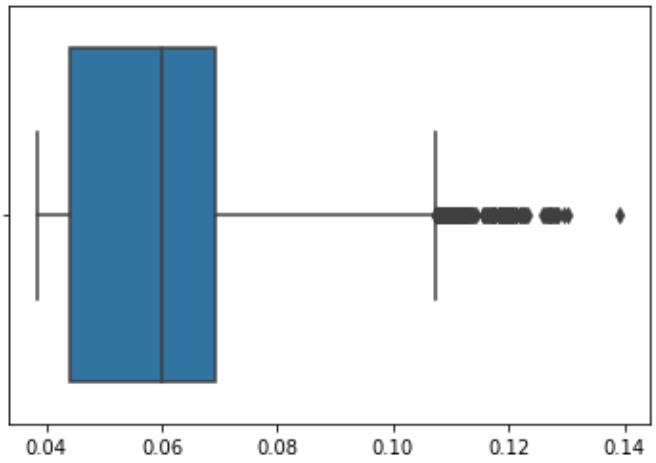
Tabla 3. Estadísticas descriptivas variables cualitativas

	workclass	education	marital-status	occupation	relationship	race	gender	native-country	income
count	48842	48842	48842	48842	48842	48842	48842	48842	48842
unique	8	16	7	14	6	5	2	41	2
top	Private	HS-grad	Married-civ-spouse	Prof-specialty	Husband	White	Male	United-States	<=50K
freq	36705	15784	22379	8981	19716	41762	32650	44689	37155

Para el posterior análisis, los descriptores categóricos fueron codificados de tal forma que se crea una variable dicotómica nueva por cada etiqueta de la variable categórica original, es decir, se realizó lo que se conoce como *One-Hot encoding*. De esta forma, ahora la base de datos cuenta con 104 variables y 48 842 individuos. Por otra parte, las variables cuantitativas fueron estandarizadas para eliminar cualquier efecto de medida que pueda influir en los resultados de las estimaciones.

Seguidamente se analizó la base en búsqueda de valores atípicos, para ello se utilizó la distancia de Gower para medir la distancia de cada individuo al vector centro de los datos; luego se removieron aquellos registros que se encontraran a una distancia de Gower mayor a 1.5 veces el rango intercuartílico. La Gráfica 1 muestra el gráfico de caja y bigotes que muestran los individuos atípicos según la distancia de Gower.

Gráfica 1. Boxplot de las distancias de Gower



Luego se utilizaron 4 técnicas diferentes para realizar la selección de características: Análisis estadístico univariante, Eliminación recursiva de características (RFE), Support Vector Machine (SVM) y una regresión de Lasso con penalización. Con cada una de estas técnicas se obtuvo una cantidad distinta de características.

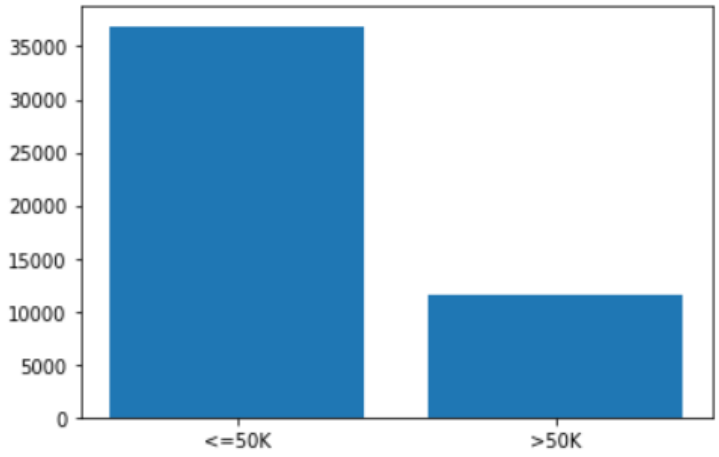
Tabla 4 Cantidad de variables por técnica de selección de características

Técnica	Cantidad de descriptores
Análisis estadístico univariante	80
RFE	52
SVM	91
Regresión Lasso con penalización	51

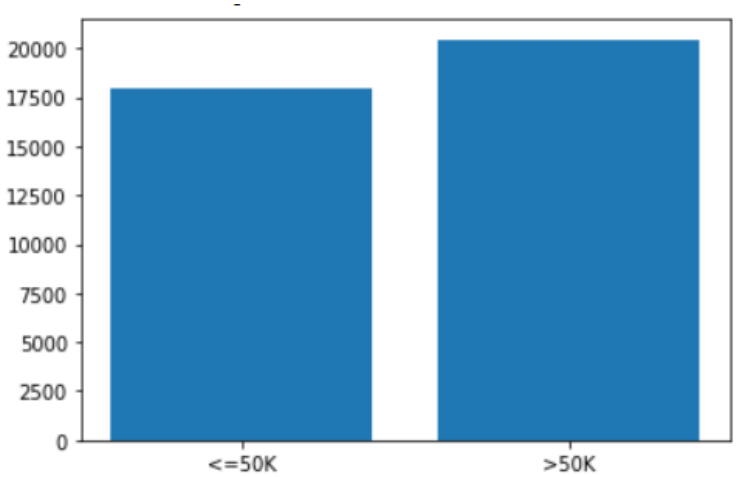
En este punto se dividió la base de datos en una base de entrenamiento y otra de prueba en una proporción de 70 % y 30 %.

A continuación, se llevó a cabo un análisis sobre el balance de clases de la variable objetivo *income* encontrando un desbalance en donde la clase mayoritaria eran a la que pertenecían individuos que ganaban menos de 50 mil dólares al año (véase Gráfica 2). Para balancear la base de datos se utilizó la técnica de SMOTEENN, que consiste en realizar un sobremuestro de la clase minoritaria y luego realizar un submuestreo de la clase mayoritaria. Esta técnica se prefiere por encima del SMOTE, debido a que permite eliminar cierto ruido que puede resultar después del solo sobremuestreo. La Gráfica 3 muestra uno de los resultados luego haberse realizado el balance sobre la base de entrenamiento utilizando SMOTEENN.

Gráfica 2. Clases de la variable income



Gráfica 3. Clases de la variable income después del balance



Ahora para seleccionar una de las bases obtenidas en la selección de características se procedió a realizar estimaciones con los diferentes modelos supervisados Support Vector Machine (SVM), Naive Bayes, Regresión logística y Random Forest.

En la Tabla 5 se muestran los resultados obtenidos para cada una de las bases en los 4 modelos *Support Vector Machine*, el clasificador *Naïve Bayes*, la regresión logística y el *Random Forest* (RF). De aquí podemos concluir que la selección de descriptores obtenida mediante la regresión de Lasso y RFE en promedio son con las que se logran mejores accurancy tanto con la base de datos de entrenamiento como al validar con la de prueba. Sin embargo, las diferencias son muy pequeñas, es menor a un punto porcentual, teniendo la base de Lasso 1 descriptor menos. En este sentido, buscando un balance entre desempeño promedio de los algoritmos y número de descriptores relevantes se optó por el uso de la base Lasso.

Tabla 5 Resultados selección de descriptores por modelo[1]

		Análisis Estadístico	SVM	Lasso	RFE
SVM	Exactitud entrenamiento	92.64%	92.70%	92.60%	92.50%
	Exactitud modelo	78.37%	77.13%	76.92%	76.93%
Naive Bayes	Exactitud entrenamiento	83.39%	82.73%	88.65%	89.34%
	Exactitud modelo	66.69%	64.68%	73.92%	75.60%
Regresión Logística	Exactitud entrenamiento	92.58%	92.83%	92.71%	92.64%
	Exactitud modelo	78.53%	77.25%	77.17%	77.13%
Random Forest	Exactitud entrenamiento	99.54%	99.41%	99.38%	99.40%
	Exactitud modelo	80.82%	81.02%	80.90%	80.82%

[1]Estos resultados son de referencia, ya que pueden cambiar con cada corrida al cambiar el conjunto de datos de entrenamiento, pero en general los descriptores de Lasso dan los mejores resultados siempre.

La regresión Lasso realiza una selección automática de las características durante el entrenamiento con la base de datos utilizando un término regulador α, escogido óptimamente, y que penaliza los coeficientes que no son relevantes para alcanzar un modelo parsimonioso. De esta forma, solo tienen importancia 51 descriptores para determinar si un individuo gana por encima de 50 mil dólares según la regresión Lasso (véase Tabla 6).

Tabla 6. Caracterización de la base de datos de entrenamiento

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 38696 entries, 0 to 38695
Data columns (total 52 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   age                38696 non-null  float64
 1   hours-per-week     38696 non-null  float64
 2   educational-num    38696 non-null  float64
 3   capital_total      38696 non-null  float64
 4   fnlwgt             38696 non-null  float64
 5   workclass_1        38696 non-null  float64
 6   workclass_3        38696 non-null  float64
 7   workclass_4        38696 non-null  float64
 8   workclass_5        38696 non-null  float64
 9   workclass_6        38696 non-null  float64
 10  education_2        38696 non-null  float64
 11  education_3        38696 non-null  float64
 12  education_4        38696 non-null  float64
 13  education_6        38696 non-null  float64
 14  education_7        38696 non-null  float64
 15  education_9        38696 non-null  float64
 16  education_10       38696 non-null  float64
 17  education_11       38696 non-null  float64
 18  education_12       38696 non-null  float64
 19  education_14       38696 non-null  float64
 20  education_15       38696 non-null  float64
 21  education_16       38696 non-null  float64
 22  marital-status_1   38696 non-null  float64
 23  marital-status_2   38696 non-null  float64
 24  marital-status_4   38696 non-null  float64
 25  marital-status_6   38696 non-null  float64
 26  occupation_1       38696 non-null  float64
 27  occupation_2       38696 non-null  float64
 28  occupation_3       38696 non-null  float64
 29  occupation_4       38696 non-null  float64
 30  occupation_5       38696 non-null  float64
 31  occupation_6       38696 non-null  float64
 32  occupation_8       38696 non-null  float64
 33  occupation_9       38696 non-null  float64
 34  occupation_10      38696 non-null  float64
 35  occupation_12      38696 non-null  float64
 36  occupation_13      38696 non-null  float64
 37  relationship_2     38696 non-null  float64
 38  relationship_3     38696 non-null  float64
 39  relationship_4     38696 non-null  float64
 40  relationship_5     38696 non-null  float64
 41  race_2             38696 non-null  float64
 42  race_5             38696 non-null  float64
 43  gender_1           38696 non-null  float64
 44  native-country_1   38696 non-null  float64
 45  native-country_4   38696 non-null  float64
 46  native-country_8   38696 non-null  float64
 47  native-country_14  38696 non-null  float64
 48  native-country_15  38696 non-null  float64
 49  native-country_21  38696 non-null  float64
 50  native-country_25  38696 non-null  float64
 51  income             38696 non-null  int64
dtypes: float64(51), int64(1)
```

Teniendo lista la base de datos de entrenamiento, que cuenta con 51 descriptores; la variable objetivo *income*; y 38 399 individuos; y la base de prueba, con 48 descriptores y 14 565 individuos; se procede a revisar en detalle los algoritmos de aprendizaje supervisado para realizar una selección.

La Tabla 7 muestra los resultados de exactitud para cada algoritmo utilizando tanto la base de prueba como la base de entrenamiento.

Tabla 7. Puntaje de exactitud de modelos de clasificación

Algoritmo	Exactitud con la base de prueba	Exactitud con la base de entrenamiento
Support Vector Machine	76.92%	92.60%
Naïve Bayes	73.92%	88.65%
Regresión logística	77.17%	92.71%
Random Forest	**80.90%**	**99.38%**

En general, no parece haber problemas de sobreajuste en ningún modelo, ya que los puntajes de exactitud con la base de entrenamiento y la de prueba no están muy distantes. El modelo que tuvo la exactitud más baja fue el clasificador *Naïve Bayes*, mientras el modelo con el mejor desempeño en exactitud fue el *RF*. Por tal motivo, este algoritmo fue seleccionado entre los propuestos inicialmente para el estudio.

RF es un algoritmo de clasificación de *Machine Learning* que construye muchos árboles de decisión que no están correlacionados para producir predicciones en conjunto que son más precisas que cualquier predicción individual (Yiu, 2019). Para el presente caso, se utiliza una agregación *bootstrap* o *bagging* para cada árbol de decisión, es decir, el modelo construye un árbol de decisión sobre un remuestreo con reemplazo del mismo tamaño de la base de entrenamiento y, por tanto, los árboles serán diferentes; esto se realiza por cada árbol del bosque. De esta forma, se espera que con los repetidos remuestreos los resultados de los diferentes árboles converjan en promedio.

La decisión del investigador al momento de utilizar el *RF* es sobre cuál es el número de árboles de decisión que el algoritmo debe construir. Al respecto, Oshiro, Perez y Baranauskas (2012) sugieren, basados en experimentos, un rango entre 64 y 128 árboles en un *RF*, ya que se obtiene un buen balance entre AUC, tiempo de procesamiento y uso de memoria. Por ello se realizaron pruebas con diferentes números de árboles y se llegó a la conclusión de que estimar un modelo *RF* con 3 árboles no disminuye grandemente la exactitud del algoritmo ya que las ganancias en exactitud con un mayor número de árboles no eran significativas, además el tiempo de procesamiento con 3 árboles es menor. El puntaje de exactitud para el *RF* con 3 árboles son los que se encuentran en la Tabla 7.

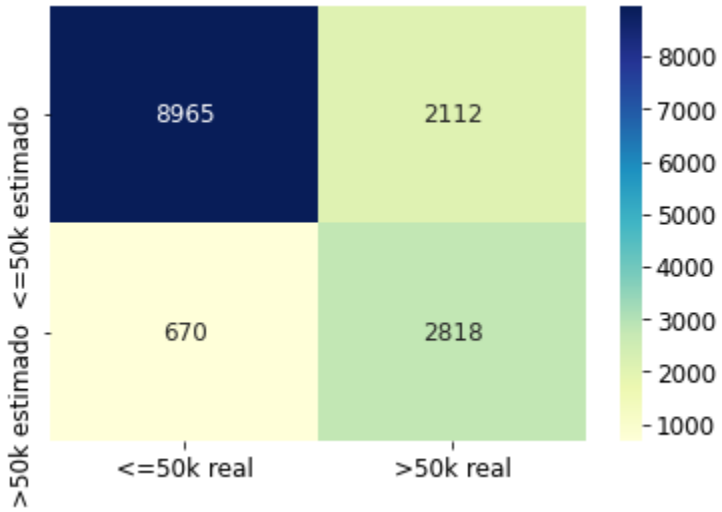
4. PRUEBAS

Habiendo entrenado el modelo *RF* con 3 árboles de decisión, se obtuvieron las estimaciones sobre la base de prueba. El puntaje de exactitud con la base de prueba es de 80.9 %, que es mayor que la probabilidad del 76.05 % de encontrar la clase más frecuente en los datos de prueba; por lo que el algoritmo presenta un mejor desempeño en comparación.

Una vez el algoritmo clasifica los registros entre individuos que ganan más de 50 mil dólares al año y los que no, se puede realizar una comparación con la clasificación de la variable objetivo *income* a través de una matriz de confusión. La matriz de confusión de la Gráfica 4 muestra un número relativamente grande de 8 965 individuos que fueron clasificados correctamente como individuos que ganan menos de 50 mil dólares al año, mientras que 2 818 fueron clasificados correctamente

como individuos que ganan más de esa cantidad. Por otro lado, 2 112 individuos fueron clasificados incorrectamente como que ganan menos de 50 mil dólares al año y 670 fueron clasificados incorrectamente como individuos que ganan más.

Gráfica 4. Matriz de confusión



A partir de los datos de la matriz de confusión se obtiene un reporte de clasificación (véase Tabla 8) que muestra algunas métricas para evaluar el algoritmo de clasificación: la precisión, la sensibilidad (*recall*) y el puntaje F1. La precisión indica que 93 % de los individuos que ganan menos de 50 mil dólares al año fueron clasificados correctamente, mientras que el 57 % de los individuos que ganan más, fueron exactamente clasificados. La sensibilidad sugiere que 81 % de los que el algoritmo clasificó como que ganaban menos de 50 mil dólares al año fueron clasificados adecuadamente, y 81 % de los clasificados como que ganaban más fueron clasificados de manera acertada. El puntaje F1 es una medida que engloba la precisión y la sensibilidad, y entre más cerca esté de 1 indica un mejor desempeño del algoritmo. Los puntajes F1 fueron de 87 % y 67 % para la clasificación de individuos que ganan menos de 50 mil dólares al año y los que ganan más, respectivamente.

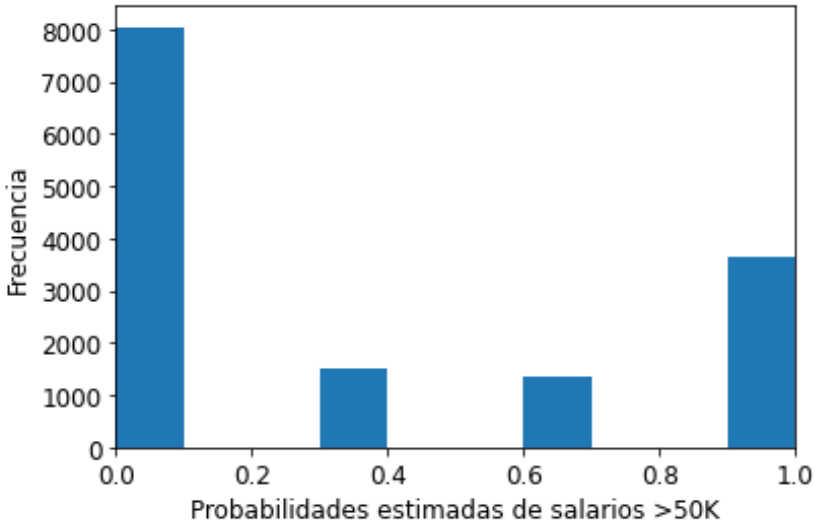
Tabla 8. Reporte de clasificación

	precision	recall	f1-score	support
0	0.93	0.81	0.87	11077
1	0.57	0.81	0.67	3488
accuracy			0.81	14565
macro avg	0.75	0.81	0.77	14565
weighted avg	0.84	0.81	0.82	14565

La Gráfica 5 muestra el histograma de las probabilidades de ganar un salario mayor a 50 mil dólares al año para cada individuo de la base de datos de prueba estimadas por el algoritmo *RF*. Se puede observar que alrededor de 8 000 individuos tienen una probabilidad entre 0 % y 10 % de ganar más de 50 mil dólares, cerca de 2 900 individuos se encuentran entre 30 % y 40 %, y entre 60 % y 70 %, mientras que en torno a 3 600 individuos tienen una probabilidad entre 90 % y 100 %. Es evidente

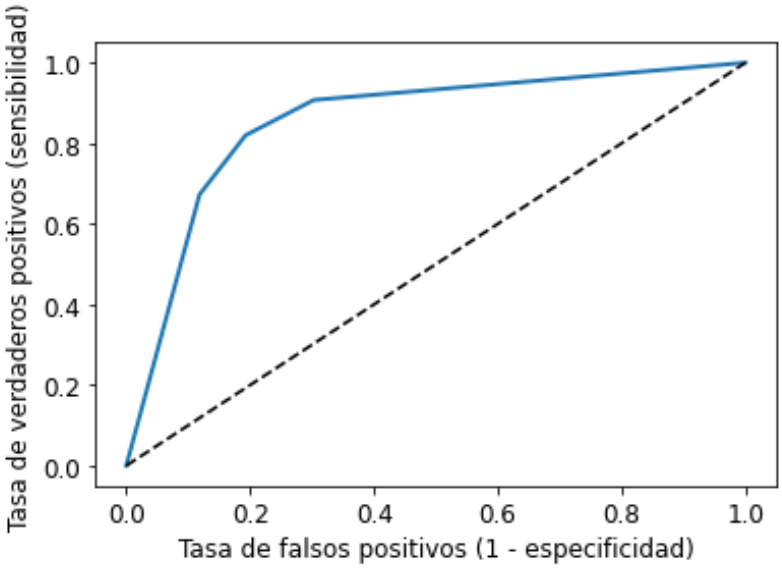
que un pequeño grupo de individuos tiene una probabilidad alta de ganar más de 50 mil dólares al año.

Gráfica 5. Histograma de las probabilidades estimadas de salarios mayores a 50 mil dólares



A continuación, se realiza el análisis de la curva ROC de la Gráfica 6. La curva ROC relaciona la sensibilidad del modelo, es decir, la tasa de verdaderos positivos, con la tasa de falsos positivos, o sea, 1 menos el puntaje de especificidad. La idea detrás de la curva ROC es que esta debe encontrarse lo más alejada posible por encima de la bisectriz, indicando que el algoritmo clasifica correctamente a los individuos que ganan más de 50 mil dólares al año con una probabilidad mayor que lo que sería clasificarlos incorrectamente.

Gráfica 6. Curva ROC del Random Forest para predecir salarios



Una forma de analizar la curva ROC es midiendo el área bajo ella o AUC. El cual indicaría la probabilidad del algoritmo de clasificar correctamente a un individuo escogido aleatoriamente. En

este caso, el AUC del *RF* es de 0.855, es decir, hay una probabilidad de 85.5 % de que un individuo escogido aleatoriamente sea clasificado correctamente como un individuo que gana más de 50 mil dólares al año. Dado que el valor de AUC está entre 0.75 y 0.9, se puede decir que el desempeño de clasificación del algoritmo de *RF* es bueno. Asimismo, se calculó el AUC promedio de una validación cruzada utilizando 10 grupos, obteniendo un valor de 0.96, lo que sugiere una agregación muy buena por parte del algoritmo.

5. ANÁLISIS DE RESULTADOS

Este trabajo buscó determinar si un individuo gana más de 50 mil dólares al año a partir de características demográficas como en el nivel de educación, la edad, el género, la ocupación, etc. Se utilizó una base de datos proveniente de Kaggle, que es una extracción del Censo 1994 de Estados Unidos. Se realizó la ingeniería de características y se seleccionó los descriptores relevantes utilizando la técnica de regresión Lasso. La base fue dividida para realizar el entrenamiento y la prueba del algoritmo de aprendizaje supervisado para hacer la clasificación. En primera instancia, se entrenaron cuatro algoritmos: *SVM*, *Naïve Bayes*, regresión logística y *RF*. *RF* fue el algoritmo con mayor puntaje de exactitud, a saber, en las diferentes ejecuciones siempre estuvo alrededor del 80% o más y predice mejor que solo predecir la ocurrencia de la clase más frecuente en la variable objetivo con probabilidad de 76.05 % (número de registros de la clase más frecuente sobre el total de registros).

La matriz de confusión de la Gráfica 4 y el reporte de clasificación de la Tabla 8 evidencian un buen desempeño del clasificador, mostrando que el 93 % de los individuos que ganan menos de 50 mil dólares al año fueron clasificados correctamente y que 57 % de los individuos que ganan más de esa cantidad fueron exactamente clasificados. Por otro lado, del total de individuos clasificados como que devengan menos de la cantidad establecida, 81 % fueron clasificados correctamente; porcentaje cerca del 81 % del total de individuos clasificados como que tienen ingresos mayores a 50 mil dólares al año fueron clasificados acertadamente.

Analizando el histograma de probabilidad de tener salarios mayores a 50 mil dólares al año de la Gráfica 5, se hace evidente la distribución bimodal. El 55.25 % de los individuos tiene una baja probabilidad de tener salarios altos, mientras que el 25.12 % tienen altas probabilidades de recibir un salario elevado. Los individuos restantes tienen probabilidades bajas y medias.

Por otra parte, la curva ROC y su área debajo indican que el 85.5 % de individuos escogidos aleatoriamente serán clasificados correctamente; en otras palabras, 86 de cada 100 personas seleccionadas al azar serán clasificadas exactamente, lo cual sugiere un desempeño bueno. Dividiendo la base de entrenamiento en 10 grupos y calculando el AUC sobre 9 grupos iteradamente descartando cada vez un grupo diferente, se obtienen 10 valores de AUC, y su promedio es otra medida de desempeño. Para este estudio, el AUC promedio es de 96.4 %, lo que sugiere un muy buen desempeño del algoritmo de *RF*.

En conclusión, tras haber comparado diferentes técnicas de aprendizaje supervisado, haber escogido y evaluado el modelo de *RF* utilizando diferentes métricas de desempeño, se puede concluir que el algoritmo de RF puede determinar acertadamente si un individuo tiene un salario mayor que 50 mil dólares al año o no.

REFERENCIAS

Oshiro, T., Perez, P., & Baranauskas, J. (2012). How many trees in a random forest? *In International workshop on machine learning and data mining in pattern recognition*, 154-168.

Yiu, T. (2019). *Understanding Random Forest*. Obtenido de Towards Data Science: https://towardsdatascience.com/understanding-random-forest-58381e0602d2