Practica 2
Clasificación de frijoles mediante tecnicas de machine learning

1st Erick Franco Gaona
Departamento de Estudios Multidisciplinarios
Universidad de Guanajuato
Yuriria, México
e.francogaona@ugto.mx

***Resumen*—Las regresiones se utilizan a menudo para predecir situaciones de la vida real en la industria y la ciencia. Existen diversas técnicas para realizar regresiones como pueden ser regresiones lineales multiples o bosques aleatorios. En este trabajo se presenta un caso de estudio de esas dos técnicas sobre un conjunto de datos público para revisar la diferencia de efectividad entre ambos métodos. Mientras que la regresión lineal multiple obtuvo un 82 % de efectividad, los bosques aleatorios obtuvieron un 88 % a costa de un mayor tiempo de ejecución.**

I. Introducción

La clasificación consta en ordenar u organizar las cosas en un conjunto de categorías o clases. Puedes categorizar ideas, objetos o cualquier tipo de referencia. El concepto de clasificación tiene diferentes vertientes: aprendizaje supervisado, no supervisado, semi supervisado y por refuerzo. El aprendizaje supervisado cuenta con un conocimiento a priori, es decir para la tarea de clasificar un objeto dentro de una categoría o clase se tienen modelos ya clasificados (objetos agrupados que tienen características comunes). En la primera fase se tiene un conjunto de entrenamiento o de aprendizaje (para el diseño del clasificador) y otro llamado de prueba o de validación (para clasificación), estos sirven para construir un modelo o regla general para la clasificación. En la segunda fase se clasifican los objetos o muestras de las que se desconoce la clase a las que pertenecen. A diferencia del aprendizaje supervisado, el aprendizaje no supervisado o clustering no se cuenta con conocimiento a priori, por lo que se tiene un área de entrenamiento disponible para la tarea de clasificación. En este tipo de clasificación contamos con "objetos" o muestras que tiene un conjunto de características, de las que no sabemos a qué clase o categoría pertenece, entonces la finalidad es el descubrimiento de grupos de "objetos" cuyas características afines nos permitan separar las diferentes clases.

En ocasiones, es muy complicado disponer de un conjunto de datos completamente etiquetado. Este tipo de aprendizaje tiene un poco de los dos anteriores. Usando este enfoque, se comienza etiquetando manualmente algunos de los datos. Una vez se tenga una pequeña porción de datos etiquetados, se entrena uno o varios algoritmos de aprendizaje supervisado sobre esa pequeña parte de datos etiquetados y se usan los modelos resultantes del entrenamiento para etiquetar el resto de los comentarios. Finalmente, se entrena un algoritmo de aprendizaje supervisado utilizando como etiquetas las etiquetadas manualmente y las generadas por los modelos anteriores. Por último, el aprendizaje por refuerzo es un método de aprendizaje automático que se basa en recompensar los comportamientos deseados y penalizar los no deseados. Es un aprendizaje que fija objetivos a largo plazo para obtener una recompensa general máxima y lograr una solución óptima. El juego es uno de los campos más utilizados para poner a prueba el aprendizaje por refuerzo. En estos casos, el agente recibe información sobre las reglas del juego y aprende a jugar por sí mismo. En un principio se comporta de manera aleatoria, pero con el tiempo empieza a aprender movimientos más sofisticados. Este tipo de aprendizaje se aplica también en otras áreas como la robótica, la optimización de recursos o sistemas de control.

En este trabajo se cuenta con un dataset público que posee 16 características de 6 diferentes tipos de frijol, por lo que, se utiliza el aprendizaje supervisado. En las siguientes secciones se explica la teoria y ejemplifica mediante el caso de estudio su aplicación.

II. Teoría

II-A. Clasificador bayesiano simple

El teorema de Bayes es utilizado para calcular la probabilidad de un suceso, teniendo información de antemano sobre ese suceso. Es posble calcular la probabilidad de un suceso A, sabiendo además que ese suceso cumple cierta característica que condiciona su probabilidad. El teorema de la probabilidad total hace inferencia sobre un suceso B, a partir de los resultados de los sucesos A. Por su parte, Bayes calcula la probabilidad de A condicionado a B. Para calcular la probabilidad tal como la definió Bayes en este tipo de sucesos, es necesaria la siguiente fórmula:

$$P[A_n/B] = \frac{P[B/A_n] * P[A_n]}{\sum P[B/A_i] * P[A_i]} \quad (1)$$

donde B es el suceso sobre el que se tiene información previa y A_n son los distintos sucesos condicionados. En la parte del numerador se encuentra la probabilidad condicionada, y el denominador la probabilidad total.

Los clasificadores NaiveBayes (NBC por sus siglas en inglés) son algoritmos de aprendizaje automático simples pero potentes. Se basan en la probabilidad condicional y el teorema de Bayes.

II-B. Regresión lineal multiple

En muchas aplicaciones habrá más de un regresor, es decir, más de una variable independiente que ayude a explicar a Y [2]. Por ejemplo, si se tratara de un problema con dos regresores la estructura múltiple de la regresión se podría escribir como

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \quad (2)$$

el modelo de regresión lineal múltiple general se expresa de la siguiente manera

$$Y = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k \quad (3)$$

donde cada coeficiente de regresión β_i se estima por medio de b_i, a partir de los datos muestrales, usando el método de mínimos cuadrados. Si se acomoda en forma matricial se tiene que:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots x_{k1} \\ 1 & x_{12} & \cdots x_{k2} \\ 1 & n & \cdots x_{kn} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix} + \begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix} \quad (4)$$

$$Y = X\beta + \epsilon \quad (5)$$

Para encontrar las β se tiene de forma matricial que

$$\beta = (X^T X)^{-1} X^T Y \quad (6)$$

II-C. Árboles de decisión

Un árbol de decisión es un método de aprendizaje supervisado que predice valores de respuesta aprendiendo reglas de decisión derivadas de características [3]. Se pueden utilizar tanto en contextos de regresión como de clasificación. A diferencia de los modelos lineales, los árboles de decisión pueden capturar interacciones no lineales entre características y objetivos.Los árboles de decisión funcionan dividiendo el espacio de características en múltiples regiones rectangulares simples, divididas por ejes de división paralelos. Para obtener una predicción para una observación en particular, se utiliza la media de las observaciones de entrenamiento en la partición a la que pertenece la nueva observación.

De forma matemática la función para un árbol de regresión se expresa de la siguiente manera:

$$f(x) = \sum_{m=1}^{M} w_m \phi(x; v_m) \quad (7)$$

donde w_m es la respuesta media en una región particular (R_m), v_m representa cómo se divide cada variable en un valor de umbral particular.

De forma visual un árbol de decisión con dos variables de características (X_1 y X_2) y una respuesta numérica "y" se puede observar en la Figura 2. Por otro lado, en la Figura 3 se muestra un subconjunto que contiene el espacio de caracteristicas. El dominio se divide mediante divisiones paralelas de eje, es decir, cada división del dominio se alinea con uno de los ejes de características.

II-D. Bosques aleatorios de decisión

El principal problema con los árboles de decisión es que tienden a sobreajustarse. Puedes crear un árbol que realice regresiones perfectamente con los datos de entrenamiento, pero no en el conjunto de datos de prueba. Para ese problema se crearon los bosques aleatorios de desición. La aplicación repetida del algoritmo de generación de árboles de decisión a los mismos datos con diferentes parámetros produce lo que se denomina un bosque de decisión aleatorio [4]. Este algoritmo es uno de los métodos de pronóstico más eficientes y ampliamente utilizados para big data en la actualidad, promediando múltiples modelos sin ruido ni sesgo para reducir la variación final general.

En la práctica, se construyen diferentes conjuntos de entrenamiento y prueba sobre los mismos datos, la unión de estos árboles de diferentes complejidades y con datos de origen distinto aunque del mismo conjunto resulta un bosque aleatorio. Su principal característica es que produce modelos más robustos que los que se obtienen generando un único árbol de decisión complejo para los mismos datos. El ensamblaje de diferentes modelos (árboles de decisión) produce predicciones más sólidas. Los grupos de árboles de clasificación se combinan y se infiere una predicción a partir de la población de árboles. Mientras haya suficientes árboles en el bosque, hay poco o ningún riesgo de sobreadaptación. Los árboles de decisión también se pueden sobreajustar. Los bosques aleatorios obtienen esto construyendo árboles de diferentes tamaños a partir de subconjuntos y combinando los resultados.

III. Resultados

Para realizar la práctica se programó en lenguaje Python (revisar Anexo A) los dos sistemas de regresión. Primero se obtuvo el conjunto de datos publico mpg-auto el cual contiene información de distintos autos que se relacionan con el rendimiento de gasolina como se muestra en la Tabla 1. Para ambos casos se tuvo que hacer un pre-procesamiento de los datos debido a que el archivo disponible no cuenta con información completa para todos los registros. Es importante mencionar que el conjunto de datos contiene registros sobre el

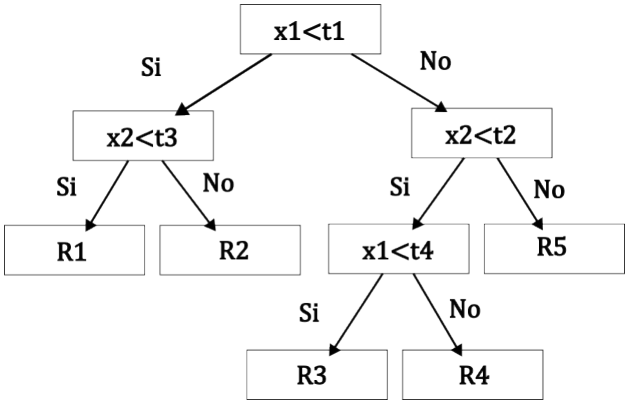

Figura 1. Ejemplo visual de un árbol de decisión.

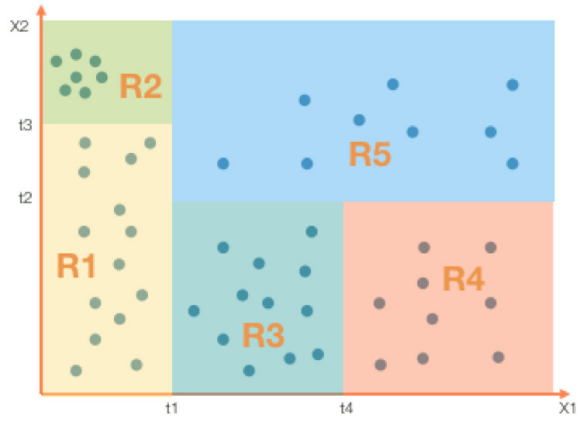

Figura 2. Divisiones de los subconjuntos que realiza un árbol de decisión.

modelo y al ser una variable categorica puede ser convertida a variable dummy. Las variables se normalizaron para quedar en el mismo rango. los datos se dividieron para entrenamiento y pruebas en una proporcion 80/20 respectivamente.

Mediante la técnica de eliminación hacia atras se optimizó la regresión lineal multiple, colocando todas las variables x se fueron eliminando las variables que su valor P superaba el umbral del 5 % para el sistema. En este conjunto de datos en particular solo se eliminó una variable la aceleración que si bien superaba el 5 % de umbral, no lo superó por mucho pues obtuvo 6 % y se pudo dejar en el modelo.

MPG-auto dataset	
Dato	Tipo de dato
Millas por galón	continuo
Cilindros	discreto multivalor
Desplazamiento	continuo
Caballos de fuerza	continua
Peso	continuo
Aceleración	continua
Año del modelo	discreto multivaluado
Origen	discreto multivaluado
Nombre del coche	cadena

Tabla I
CONJUNTO DE DATOS MPG-AUTO

Para los bosques aleatorios el proceso es similar, se ejecuta con el pre-procesamiento antes mencionado. Los resultados obtenidos se midieron mediante las metricas R^2 y R^2 ajustada como se observa en la Tabla 2.

Regresión lineal multiple	Bosques aleatorios
R^2 = 82.54 %	R^2 = 82.27 %
R^2 = 82.54 %	R^2 = 88.22 %

Tabla II
RESULTADOS DEL ENTRENAMIENTO

CONCLUSIONES

Como se pudo observar en los resultados, los bosques aleatorios son más robustos que la regresión lineal multiple aunque requiere más tiempo de procesamiento. Sin embargo, actualmente el poder de computo es cada vez mayor por lo que el tiempo se reduce con el paso de los años mediante el hardware. Durante el entrenamiento no se tomó en cuenta la variable categorica del modelo del auto ya que al intentar convertirla en variable dummy, eran demasiadas variables que no podian ser procesadas. Para trabajo futuro se puede modificar manualmente el conjunto de datos para asegurarse en reducir esa variable y colocar solo la marca del vehiculo. No se realizó en este trabajo ya que el rendimiento obtenido es considerablemente optimo por lo que podría no valer la pena incluir esas variables dummies.

Las regresiones no son 100 % efectivas ya que no siempre se va a ajustar perfectamente a los valores a predecir pero si se puede hacer una aproximación cercana mediante la optimización de parametros de los métodos. Ademas, es bueno revisar de ser posible los datos que se proporcionan para el entrenamiento y eliminar los valores atipicos para evitar errores. El sobreajuste es un problema que trata de evitar los bosques aleatorios y que generalmente cumple su función, pero dependiendo de la aplicación puede optarse por una técnica u otra. A pesar de no ser confiables en su totalidad. las predicciones que es capaz de realizar alguno de estos métodos son lo suficientemente robustos para utilizarse en la industria ajustando el modelo lomejor posible para mejores resultados.

REFERENCIAS

[1] Yang, Q. (2017). Regression. In: Schintler, L., McNeely, C. (eds) Encyclopedia of Big Data. Springer, Cham. https://doi.org/10.1007/978-3-319-32001-4-174-1.
[2] Probabilidad y estadística aplicadas a la ingeniería, Douglas C. Montgomery y George C. Runger. Limusa Wiley, 2002. Segunda edición.
[3] Rokach, L., Maimon, O. (2005). Decision Trees. In: Maimon, O., Rokach, L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA. https://doi.org/10.1007/0-387-25465-X-9.
[4] Cutler, A., Cutler, D.R., Stevens, J.R. (2012). Random Forests. In: Zhang, C., Ma, Y. (eds) Ensemble Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-1-4419-9326-7-5.

IV. ANEXO A

Lectura de conjunto de datos y separandolos en variables dependientes e independientes y eliminación de valores incompletos.

```
cols=["MPG", "cylinders", "displacement", "
    horsepower", "weight", "acceleration", "
    model␣year", "origin"]
dataset=pd.read_csv("auto-mpg.data", na_values="
    ?", comment='\t', sep='␣', skipinitialspace=
    True, names=cols)
X=dataset.iloc[:,1:].values
Y=dataset.iloc[:,0].values
```

Normalizado de datos.

```
sc=StandardScaler()
X=sc.fit_transform(X)
```

Separando el conjunto de datos.

```
X_train, X_test, Y_train, Y_test=
    train_test_split(X,Y,test_size=0.2,
    random_state=0)
```

IV-A. Regresión lineal multiple

Ejecución de la regresión lineal multiple y calculo de las metricas de evaluación.

```
regresion=LinearRegression()
regresion.fit(X_train, Y_train)
ypred=regresion.predict(X_test)

print("R^2␣:␣", r2_score(Y_test, ypred))
print("R^2␣ajustada:␣", 1 - (1-r2_score(Y_test,
    ypred))*(len(Y)-1)/(len(Y)-X.shape[1]-1))
```

IV-B. Bosques aleatorios

Ejecución del bosque aleatorio estableciendo 10 árboles y calculo de las metricas de evaluación.

```
regresion=RandomForestRegressor(n_estimators=10
    ,random_state=0)
regresion.fit(X_train, Y_train)
ypred=regresion.predict(X_test)

print("R^2␣:␣", r2_score(Y_test, ypred))
print("R^2␣ajustada:␣", 1 - (1-r2_score(Y_test,
    ypred))*(len(Y)-1)/(len(Y)-X.shape[1]-1))
```