



INSTITUTO POLITECNICO NACIONAL “IPN”



ESCUELA SUPERIOR DE COMPUTO “ESCOM”

LANDEROS CORTES PEDRO JONAS

MACHINE LEARNING

PRACTICA 1

INDICE

Introducción	2
Desarrollo.....	3
Flujo del programa	3
Ejecución del programa.....	3
Explicación de código	5
Anexos	6
Conclusiones	7

INDICE DE FIGURAS

FIGURE 1. INTERACCION INICIAL CON EL USUARIO	4
FIGURE 2. FRUTA NO ENCONTRADA	4
FIGURE 3. FRUTA ENCONTRADA	4
FIGURE 4. MULTIPLES MATCH DE FRUTAS	5

Introducción

En el campo de machine learning, los sistemas aprenden a partir de datos para mejorar su capacidad de tomar decisiones o realizar predicciones sin la necesidad de ser programados explícitamente en cada caso. Esta práctica implementa un programa basado en esa idea que funciona para intentar predecir distintas frutas. El objetivo es que a través de una serie de preguntas estructuradas sobre características como lo son el color, si tiene semillas, o si se pela el sistema intente adivinar la fruta que el usuario tiene en mente.

El funcionamiento se apoya en una base de datos (CSV) en donde se almacenan frutas y sus atributos, cada que el sistema acierta se incrementa el conteo de apariciones de esa fruta, lo cual permite priorizarla en futuras predicciones. Cuando la fruta no es identificada el programa entra en un modo de aprendizaje, solicita el nombre de la fruta al usuario y guarda sus características dentro de la base de datos. Este mecanismo

emula el proceso de retroalimentación de los modelos de aprendizaje automático en donde los errores se convierten en oportunidades de mejora.

Desarrollo

Flujo del programa

El flujo general de ejecución del programa se puede describir en tres fases:

1. **Interacción inicial con el usuario:** El sistema solicita al usuario que piense en una fruta, luego plantea preguntas específicas sobre sus características. Las respuestas pueden ser afirmativas negativas o desconocidas, lo que permite cierto nivel de flexibilidad en el proceso de identificación.
2. **Filtrado:** Con base en las respuestas del usuario, el programa filtra la base de datos para obtener un conjunto de frutas candidatas que cumplen con las condiciones dadas, en caso de haber múltiples coincidencias, se ordenan de acuerdo con el número de apariciones (cuantas veces han sido acertadas previamente) y por orden alfabético en caso de un nuevo empate.
3. **Aprendizaje:** Si el sistema adivina correctamente la fruta se incremente el contador de apariciones para reforzar la importancia de esa opción en el futuro, pero si no logra acertar se solicita el nombre de la fruta al usuario y registra sus características, enriqueciendo así la base de datos, este mecanismo de retroalimentación emula un aprendizaje supervisado básico: a partir de ejemplos nuevos el programa mejora su capacidad de predicción.

Los valores exactos para considerar de cada fruta son los siguientes:

- Nombre
- Color
- Critica
- Tropical
- Con semillas
- Se pella
- Dulce
- Apariciones

Ejecución del programa

A continuación, se muestran ejemplos de la ejecución del programa:

Interacción inicial

```
== Akinator de Frutas (multivalor) ==
Usando base: frutas_multivalor.csv
Piensa en una FRUTA.
Responde con s / n / ? (desconocido). Escribe 'regresar' para volver a la pregunta anterior.
Para el color, escribe el color (o ?). Ej: amarilla, amarillo, azul, beige, cafe, marron, morado, naranja, negro, rojo, rosa, verde

Color principal (minusculas, o '?' si no sabes): |
```

Figure 1. Interaccion inicial con el usuario

Fruta no encontrada

```
Color principal (minusculas, o '?' si no sabes): ?
¿Es citrica? (s/n/?)
¿Es tropical? (s/n/?)
¿Tiene semillas? (s/n/?)
¿Se pela para comer? (s/n/?)
¿Es dulce? (s/n/?)

No acerte. Vamos a aprender esta fruta (solo necesito el nombre).
Nombre: |
```

Figure 2. Fruta no encontrada

Fruta encontrada

```
Color principal (minusculas, o '?' si no sabes): amarillo
¿Es citrica? (s/n/?)
¿Es tropical? (s/n/?)
¿Tiene semillas? (s/n/?)
¿Se pela para comer? (s/n/?)
¿Es dulce? (s/n/?)
¿Es platano? (s/n)
¡Adivine!
¿Otra vez? (s/n) |
```

Figure 3. Fruta encontrada

Varios match de mismas características

```
Color principal (minusculas, o '?' si no sabes): morado
¿Es citrica? (s/n/?)
¿Es tropical? (s/n/?)
¿Tiene semillas? (s/n/?)
¿Se pela para comer? (s/n/?)
¿Es dulce? (s/n/?)
¿Es uva? (s/n)
¿Es ciruela? (s/n)
¡Adivine!
¿Otra vez? (s/n)
```

Figure 4. Multiples match de frutas

Explicación de código

El programa se estructura en varios bloques que permiten organizar sus funcionalidades de manera clara.

En primer lugar, se definen las constantes y el esquema de datos. Se establece la lista de columnas que debe tener el archivo CSV, incluyendo atributos como el nombre, color y características booleanas, además del campo apariciones que cuenta cuántas veces se ha acertado una fruta. También se definen conjuntos de palabras que representan respuestas afirmativas, negativas o comandos para retroceder.

Luego aparecen las funciones de normalización, que sirven para unificar la información de entrada. Estas funciones eliminan acentos, convierten a minúsculas y transforman valores como “sí”, “yes” o “1” en True, y “no” o “0” en False. De la misma manera, se normalizan los colores para evitar problemas de escritura y se procesan celdas que contienen varios valores separados por distintos símbolos (|, ,, ;, /). Gracias a este proceso, las entradas del usuario y los datos del CSV quedan en un formato estándar, lo que facilita las comparaciones y evita errores.

En la parte de representación interna, cada fila del CSV se convierte en un diccionario de conjuntos. Esto permite que una fruta pueda tener varias opciones de color o valores booleanos ambiguos. Por ejemplo, un limón puede estar tanto en la categoría “verde” como en “amarillo”. Existen funciones para fusionar registros repetidos y para volver a transformar esos datos en una fila lista para guardarse en el CSV.

El bloque de entrada y salida de datos se encarga de leer el archivo CSV, verificar que tenga las columnas correctas, fusionar duplicados y guardar los cambios ordenados alfabéticamente. También incluye la lógica para añadir nuevas frutas o actualizar las existentes.

El motor de preguntas es el encargado de interactuar con el usuario. Presenta las preguntas en un orden fijo (color, cítrica, tropical, con semillas, se pela y dulce) y permite regresar a una pregunta anterior en caso de error. Cada respuesta escrita por el usuario se normaliza y se traduce en valores consistentes (True, False o None).

Una vez obtenidas las respuestas, el motor de búsqueda filtra las frutas candidatas que coincidan con esas características. Estas se ordenan priorizando las que tienen más apariciones y, en caso de empate, por orden alfabético. El programa pregunta al usuario si alguna de esas frutas es la correcta y, si acierta, incrementa su contador de apariciones.

Si no se logra adivinar la fruta, entra en el modo de aprendizaje. En este caso, solicita al usuario el nombre de la fruta y la agrega al CSV con las características introducidas, ampliando así la base de conocimiento del sistema.

Finalmente, el programa incluye un bloque de métricas y evaluación, que permite simular su rendimiento calculando el porcentaje de aciertos y el número promedio de preguntas necesarias para adivinar correctamente.

El archivo principal define la función main, que controla el flujo general. Al iniciar, revisa si existe el CSV, permite ejecutar solo la simulación de métricas si se pasa el parámetro --simular, y en el modo normal ejecuta el ciclo de juego hasta que el usuario decida terminar.

Anexos

Ejemplo representativo de la base de datos (CSV) con algunas muestras de distintas frutas

nombre	color	citrica	tropical	con_semilla	se_pela	dulce	apariciones
acerola	rojo	TRUE	TRUE	TRUE	FALSE	TRUE	0
aguacate	negro verde	FALSE	FALSE	TRUE	True False	TRUE	21
albaricoque	naranja	FALSE	FALSE	TRUE	FALSE	TRUE	0
arandano	azul	FALSE	FALSE	TRUE	FALSE	TRUE	0
arandano	rojo	FALSE	FALSE	TRUE	FALSE	FALSE	0
banana	amarillo	FALSE	TRUE	FALSE	TRUE	TRUE	0
bergamota	verde	TRUE	FALSE	TRUE	TRUE	FALSE	0
cana	verde	FALSE	TRUE	FALSE	TRUE	TRUE	0
caqui	naranja	FALSE	FALSE	TRUE	FALSE	TRUE	0
carambola	amarillo	FALSE	TRUE	TRUE	FALSE	TRUE	0
cereza	rojo	FALSE	FALSE	TRUE	FALSE	TRUE	0
chirimoya	verde	FALSE	TRUE	TRUE	TRUE	TRUE	0
ciruela	morado	FALSE	FALSE	TRUE	FALSE	TRUE	1
ciruela	roja rojo	FALSE	FALSE	TRUE	FALSE	TRUE	0
clementina	naranja	TRUE	FALSE	FALSE	TRUE	TRUE	0
coco	cafe marron	FALSE	TRUE	True False	TRUE	True False	10
durazno	naranja	FALSE	True False	TRUE	FALSE	TRUE	1
durian	verde	FALSE	TRUE	TRUE	TRUE	TRUE	0
framboesa	rojo	FALSE	FALSE	TRUE	FALSE	TRUE	0
fresa	rojo	True False	True False	TRUE	FALSE	True False	23
granada	rojo	FALSE	FALSE	TRUE	TRUE	TRUE	0
granadilla	naranja	FALSE	TRUE	TRUE	TRUE	TRUE	0
grosella	rojo	FALSE	FALSE	TRUE	FALSE	FALSE	0
grosella	negra morado	FALSE	FALSE	TRUE	FALSE	FALSE	0

Conclusiones

La práctica permitió aplicar conceptos básicos de machine learning de una manera sencilla y práctica, utilizando un programa que funciona como un “Akinator de frutas”. A través de la interacción con el usuario, el sistema fue capaz de adivinar una fruta a partir de preguntas sobre sus características, apoyándose en una base de datos que se enriquece progresivamente. Cada acierto incrementa la frecuencia de apariciones y cada error se convierte en una oportunidad de aprendizaje al añadir nueva información.

En conclusión, la actividad resultó útil para entender la importancia de los datos en los sistemas inteligentes y cómo incluso un enfoque sencillo puede reflejar el comportamiento adaptativo característico del aprendizaje automático.