Nombre: José Arcos Aneas
D.N.I: 74740565-H

Prácticas tema 1

Apartados con el algoritmo k-NN:

A) Experimento para evaluar la incidencia de k: Clasificar los datos de test para k=1,3,5,7,9,11,..,51 usando la distancia L2 (Euclídea). Obtener una gráfica con las 3 curvas (una por clase) mostrando la evolución de los valores de precisión para cada k. Obtener la curva promedio de la precisión de todas las clases para cada valor de k y representarla junto a las otras gráficas. Valorar dichas gráficas dando una posible explicación a su comportamiento.
Código del apartado: k-NN --- apartadoA-kNN

Este experimento sirve para valorar la elección de k. Llegamos a la conclusión después de un numero alto de experimentos que la mejor elección de k depende fundamentalmente del valor de los datos, para valores grandes de k se reduce el llamado efecto ruido en la clasificación pero se crean limites entre clases parecidas. Un buen k puede ser determinado mediante una optimización de uso.

B) Experimento para valorar el error de ajuste: Inspirándose en el código del punto anterior. Obtener las gráficas de los errores de ajuste (porcentaje de falsos positivos + falsos negativos) por cada clase, para valores de k=1,3,5,7,9,11,..,51 cuando se clasifican tanto los datos de entrenamiento como los datos de test. Dibujar las gráficas obtenidas y valorar su comportamiento.
Código del apartado: k-NN----apartadoB-k-NN

Los resultados son valores opuestos a los del apartado anterior ya que se representa el error como (1- precisión).

C) Experimento para valorar la ponderación de los ítems: Repetir el experimento anterior pero ahora ponderando la influencia de los prototipos con el inverso de su distancia al ítem. Comparar los resultados con los del punto anterior explicando las diferencias.
Código del apartado: k-NN----apartadoC-k-NN.

Se considera la contribución de cada vecino de acuerdo a la distancia entre el y el ejemplar a ser clasificado x, dando mayor peso a los vecinos mas cercanos. Se pondera el voto de cada vecino de acuerdo al cuadrado inverso de sus distancias. De esta forma no hay riesgo de permitir a todos los ejemplos de entrenamiento contribuir a la clasificación de x ya que al ser muy distantes no tendrían peso asociado. La desventaja de considerar todos los ejemplos seria su lenta respuesta. Se busca un método local en el que solo los vecinos mas cercanos son considerados. Este método presenta mayor robustez ante los ruidos de datos y suficientemente efectivo en conjunto de datos grandes.

D) Experimento para valorar el tamaño de la muestra de prendizaje: Ahora fijamos el valor k=1 y vamos generando distintas particiones aleatorias de la base de datos con #training=50%,60%,70%,80%,todos-1 del tamaño de la base de datos y el resto para test. Para cada valor de #training repetimos el experimento 10 veces y promediamos los valores de precisión. Dibujar gráficas de los resultados de precisión obtenidos por cada clase para los distintos valores de #training. Hacerlo también para la precisión media total de las clases. Valorar los resultados.
Código del apartado: k-NN----apartadoD--kNN

Para valores altos de training en ocasiones 50%-90% los resultados son en su mayoría satisfactorios en cambio para valores pequeños de training el resultado no es tan alentador. En

ciertas ocasiones un valor alto de training nos lleva a errores, esto se conoce como sobre-entrenamiento. El hecho de que solo tenga a un vecino también es perjudicial para valores que están en la frontera entre las regiones de dos clases. Para el ultimo valor en el que todas las muestas menos una son tomadas como training se presenta en gran parte de los experimentos sobre-entrenamiento.

E) Experimento para valorar la influencia de distancia usada. Repetir el experimento anterior con distancia L1 (Manhatan). Valorar los resultados de precisión obtenidos frente a los obtenidos con la distancia L2 (Euclídea).
Codigo del apartado: k-NN----apartadoE-kNN
La distancia de Manhatan es la distancia entre dos puntos medida sobre ejes a ángulos rectos, es decir; la distancia que se recorrería para llegar de un punto a otro si se siguiera una trayectoria de rejilla (o de cuadrícula), la distancia de Manhattan mide la distancia total entre la solución ideal y cada alternativa dependiendo del problema.
Debemos tener en cuanta que la región que rodea a un valor x a determinar no es un circulo, en el caso de la distancia de Manhatan es un rombo.
Los resultados utilizando esta distancia son mejores las precisión aumenta conforme aumenta el tamaño del training.

F) Experimento para valorar la influencia de la normalización de datos. Normalizar cada una de las variables a media 0 y desviación típica 1. Repetir el experimento (A) con datos sin normalizar y normalizados (usar np.random.seed(0)). Comparar los resultados y explicar las diferencias que se aprecien.
Código del apartado: k-NN----apartadoF-k-NN
Al normalizar o estandarizar los datos se hacen mas correctos y los resultados del clasificador son mas efectivos. Al comparar los datos con los del apartado demostramos que los resultados son mas exactos para el caso en el que los datos están normalizados.

Observación general:

Los resultados son mas dependientes de la relevancia de las muestras. Además de la elección de un buen conjunto de entrenamiento que represente de forma correcta las características de las clases que forman el rango de valores a clasificar.

<u>Apartados relacionados con arboles de decisión:</u>

A. Experimento para valorar la influencia de la poda: Realizaremos un experimento aleatorio que simula una situación de las que abordan las técnicas de poda. Considere para este experimento que el 30% de las muestras son de entrenamiento y el 70% de test. Con ello conseguimos que en las muestras de entrenamiento no estén representados todas las regularidades de las población.
1.- Ajustar 100 modelos de árbol a las datos de entrenamiento para un rango de valores de 1 a 8 en el número mínimo de items por hoja (min_samples_leaf). (no fijar semilla en el generador de números aleatorios y reiniciar el generador para cada árbol)
2.- Calcule el error promedio sobre todas las clases de los 100 árboles sobre los datos de entrenamiento y los de test para cada valor de min_samples_leaf y represente las dos curvas obtenidas. Valore si se observa algún patrón de interés en dichas curvas que permita determinar la profundidad más adecuada para el árbol. Si es necesario ejecutar el experimento varias veces y observar las curvas resultantes.
Código de este apartado: Arboles de Decisión: apartadoA-ArbolesDecision

Con los resultados de la gráficas se puede determinar que el 4 es un valor apropiado de profundidad para construir los arboles.
El código explica como se ha realizado.

B. Experimento para valorar la interpretación de los árboles: Fijar el conjunto entrenamiento y estimar un árbol usando min_samples_leaf =8.
Código de este apartado: Arboles de decisión---apartadoB-ArbolDecision.py
a) Escribir en términos de cláusulas (IF..THEN..) el proceso de decisión del árbol para el árbol completo

```
if (X[3]<= 0.80000
        error 0.664489795918
        samples=105
        value=[33. 33. 39.])
        then(error =0.000
                samples=33
                value=[33. 0. 0.])
        if(X[3] <= 1.7500
                error=0.496552777778
                samples=72
                value=[0. 33. 39.])
                if(X[2] <= 4.4500
                        error=0.197530864
                        samples=36
                        value=[0. 32. 4.])
                        then( error=0.0000
                                samples=19
                                value=[0. 19. 0.])
                        if(X[1] <= 2.8500
                                error=0.359861591696
                                samples=17
                                value=[0. 13. 4.])
                        them(error=0.5 samples=8 value=[0. 4. 4.]
                                or
                           error=0.00samples=9 value=[0. 9. 0.]

        if(X[3] <= 1.8500
                error=0.054012345679
                samples=36
                value=[0. 1. 35.])
                then ( error=0.2188 samples=8 value=[0. 1. 7.]
                        or
                     error=0.0000 samples=28 value=[0. 0. 28.])
```

b) Comparar dichas reglas con los obtenidas hasta los nodos de nivel-4 del árbol e identificar posibles redundancias.
No se presentan redundancias hasta los nodos de nivel-4
c) Decir si algunas de las reglas del nivel-8 son susceptibles de simplificación
No he encontrado ninguna simplificación para las reglas de nivel 8.

C. Experimento para valorar los criterios de impureza de un nodo: Ajustar árboles usando el criterio de Gini y el de Ganancia- Información(entropía), dejando el resto de parámetros constantes.

Evaluar y comparar los árboles obtenidos.
Código del apartado: Arboles de Decision---apartadoC-ArbolesDecision.py

Los arboles dibujados con estos criterios presentan diferencia ya que los arboles construidos con el criterio de Gini utilizan los coeficientes de Gini que son medidas de desigualdad, por tanto tendra mayor relevancia la desigualdad de los datos de los elementos dentro de sus diferentes clases que se han utilizado para training. Los arboles construidos por el criterio de la entropía (la cual es inversamente proporcional a la redundancia que podemos tener de una variable) dan mayor importancia a lo bien formadas que están las clases que forman nuestro conjunto de test y training. Los arboles con el criterio de la entropía son balanceados hacia la derecha y presentan un error en la raíz y en los niveles inferiores a ella, mayor que los arboles con el criterio de Gini.
A pesar de eso la precisión es la misma en ambos arboles como se puede comprobar al ejecutar en Spyder el código de este apartado

Bibliografía y fuentes de información para realizar el trabajo de prácticas:

- Wikipedia.
- Código de ayuda para el apartadoA que subió el profesor de la asignatura a la plataforma.
- Distintos documentos de libre copia de universidades españolas de informática y estadistica (como Málaga, Granada, Madrid, Valencia,.....)