PRÀCTICA III: Aprendizaje Bayesiàno: Naïve Bayes

Nombre: Oussama Berrouhou Barrouhou

NIU: 1711619

Fecha: 23/12/2024

ÍNDICE

INTRODUCCIÓN

En este proyecto, se ha implementado un sistema de clasificación de tweets mediante una red bayesiana (Naïve Bayes) para determinar si son positivos o negativos. Además, se han analizado distintas estrategias para variar el tamaño del conjunto de entrenamiento y/o el diccionario de palabras, así como se ha tenido en cuenta el efecto de Laplace smoothing. El objetivo principal es observar cómo cambian los resultados (sobre todo la exactitud o como la llamaremos a partir de ahora, accuracy) según cada estrategia y comprender mejor el algoritmo de Naïve Bayes.

SOLUCIÓN PROPUESTA

PARTE C

En este apartado se desarrolla la parte fundamental: implementar una red bayesiana para determinar si los tweets de prueba, es decir test, son positivos o negativos. Se usa la base de datos completa (tanto train como test) y el diccionario obtenido de la parte de entrenamiento.

- Generación de diccionarios (estructura y contenido)

  Para generar el diccionario, se itera sobre todos los tweets de entrenamiento. Para cada tweet, se dividen sus palabras y se actualizan los contadores de frecuencia de cada clase: word_counts_pos (positivo) y word_counts_neg (negativo).
  Con estas frecuencias es posible calcular P(w|c) y, al precomputar estos valores, se obtiene una tabla con las probabilidades condicionales de cada palabra con respecto a cada clase.
  Se añade la opción min_freq para filtrar palabras que aparezcan muy pocas veces.

- Si hay múltiples diccionarios, ¿por qué y cómo?

  Es posible crear varios diccionarios variando min_freq. Esto nos ayuda a comparar el efecto de tener un diccionario más extenso frente a uno más reducido.

También se puede usar un diccionario fijo (cosa que se hace en la estrategia 3) para observar el rendimiento cuando el vocabulario no cambia, mientras el conjunto de train crece o disminuye.

- Justificación del método de validación

  Se ha utilizado k-fold cross-validation para tener una estimación más robusta del accuracy del modelo. Este método reduce la dependencia de una sola división train/test y proporciona una medición más estable del rendimiento, especialmente en datasets grandes para posteriormente poder trabajar con la mejor partición que sería, en este caso, la que mejor accuracy tendría.

- Justificación de la métrica

  Se elige la accuracy porque es una métrica sencilla y adecuada para problemas de clasificación binaria razonablemente equilibrados. Además, puede consultarse la matriz de confusión, pero como métrica principal accuracy es útil para comparar rápidamente los distintos experimentos.

- Resultados y análisis

  Los resultados muestran que la red bayesiana obtiene una accuracy razonable (entre el 66% y el 76% según la estrategia, el dataset y el parámetro min_freq).

  En general, la validación k-fold refleja mayor estabilidad en la medición, y se observan márgenes estrechos entre el mejor y el peor fold.

PARTE B

En este ejercicio, se evalúa la red variando distintos tamaños de train y de diccionario, sin Laplace Smoothing:

Las estrategias a analizar son:

Ampliar el conjunto de train:

Al aumentar el número de tweets de entrenamiento, también crece de forma natural el diccionario, porque se encuentran palabras nuevas.

Fijar el train, pero variar el tamaño del diccionario:

Esto puede hacerse filtrando con distintos valores de min_freq.

Cuanto más alto sea min_freq, más ruido se elimina, pero también se pueden descartar palabras que podrían resultar relevantes.

Mantener el diccionario fijo, pero modificar el train:

Se construye el diccionario con todo el train (o un train completo) y luego se usa solo una parte para entrenar. Esto permite analizar el rendimiento cuando el vocabulario no cambia.

Preguntas a responder:

¿Qué pasa en el primer caso, cuando se amplía el número de tweets y el diccionario crece?

Al añadir más tweets, el modelo suele mejorar o mejor dicho estabilizarse porque se dispone de más datos, pero a veces el exceso de información o ruido puede hacer que el accuracy no mejore indefinidamente.

¿Cómo afecta el tamaño del diccionario?

Un diccionario muy grande por ejemplo, con min_freq=1, puede introducir ruido o muchas palabras infrecuentes. Por otro lado un diccionario muy pequeño con min_freq elevado, puede no tener la diversidad suficiente para distinguir tweets. Suele haber un equilibrio entre un vocabulario extenso y uno demasiado restringido.

¿Cómo afecta el tamaño del conjunto de tweets de test?

Si el test se reduce demasiado, la medida de accuracy puede fluctuar mucho debido al bajo número de ejemplos. Con un test más grande se obtiene una métrica más estable, pero se entrena con menos datos.

PARTE A

Este apartado repite lo del B, pero incluyendo Laplace Smoothing para calcular $P(w|c)$. Así, para cada palabra:

$P(w|c)=count(w,c)+\alpha/(total\_words(c)+\alpha\times|V|)$

¿Cómo afecta el Laplace Smoothing?

Laplace Smoothing evita que ciertas palabras con count=0 tengan probabilidad cero, y por lo tanto no se penalice en exceso la clase si aparece una palabra desconocida. En datasets con alta dispersión de palabras, suele mejorar la accuracy porque el modelo no anula completamente la clase ante palabras no vistas. Cuando hay muchos datos,puede ser menor, pero generalmente ayuda a estabilizar el modelo.

PROBLEMAS ENCONTRADOS DURANTE EL PROYECTO

PARTE C

Al implementar la parte C y desarrollarla, me han surgido algunos problemas, los más destacables son:
Datos inconsistentes: Algunos tweets contenían caracteres extraños o carecían de etiqueta y esto se resolvió de una manera bastante fácil como lo es filtrar y usar dropna.
Tamaños muy grandes: Con más de un millón de tweets, se necesitó optimizar o particionar ciertas funciones con librerías debido a un altísimo tiempo a la hora de ejecutar el código.
Con esto, se consiguió pasar de aproximadamente 20 minutos a 40 segundos usando todo el dataset.

PARTE B

Para esta parte, los problemas encontrados durante la implementación ha sido los siguientes:
Diferencias sutiles de accuracy: Al aumentar el conjunto train, no siempre sube la accuracy si el dataset contiene ruido.
Equilibrio de min_freq: Subirlo demasiado puede reducir el vocabulario excesivamente, por lo que mantenerlo bajo puede incluir mucho ruido.

PARTE A

Cualquier error al sumar alpha (α) en el numerador y en el denominador puede causar una bajada inesperada de accuracy y esto mismo es lo que pasó inicialmente en el desarrollo de dicha parte. Finalmente solo hubo que modificar la función adecuándola a la fórmula original.

RESULTADOS

A continuación, se detallan los resultados obtenidos para cada apartado del proyecto, incluyendo las diferentes estrategias implementadas y sus correspondientes accuracies.
Para cada implementación nos basaremos en el siguiente formato que corresponde al del dataset proporcionado:

id	text	date	label
1	is so sad for my apl friend	04/03/2015	0
2	i miss the new moon trail	06/10/2015	0
3	omg it already 730 o	03/04/2015	1
4	omgag im sooo im gunn cry i've been at thi de...	13/11/2015	0
5	i think mi bf is che on me tt	10/08/2015	0

PARTE C

```
Ejemplo de datos:
    id                                                text        date  label
0   1                        is so sad for my apl friend   04/03/2015      0
1   2                          i miss the new moon trail   06/10/2015      0
2   3                               omg it already 730 o   03/04/2015      1
3   4   omgag im sooo im gunn cry i've been at thi de...   13/11/2015      0
4   5                      i think mi bf is che on me tt   10/08/2015      0
Fold 1, Accuracy: 0.6642
Fold 2, Accuracy: 0.6662
Fold 3, Accuracy: 0.6645
Fold 4, Accuracy: 0.6657
Fold 5, Accuracy: 0.6655
Fold 6, Accuracy: 0.6647
Fold 7, Accuracy: 0.6653
Fold 8, Accuracy: 0.6669
Fold 9, Accuracy: 0.6648
Fold 10, Accuracy: 0.6645

Resultados K-Fold (10 folds):
Mejor fold: 8 con accuracy: 0.6669
Accuracy promedio: 0.6652
```

Accuracy promedio: Al ejecutar la implementación del código de la Parte C se obtiene una accuracy promedio de 66.52% en la validación k-fold, lo que indica que el modelo tiene una capacidad moderada para clasificar correctamente los sentimientos de los tweets.

Estabilidad del modelo: Aparte, la variación entre los distintos folds es mínima, oscilando entre 66.42% y 66.69%. Esto sugiere que el modelo es estable y no está excesivamente influenciado por una única partición de los datos.

Mejor fold: Por otro lado, el fold 8 alcanzó la mejor accuracy con un 66.69%, mostrando que en ciertas particiones el modelo puede rendir ligeramente mejor.

Conclusión: La cross-validation confirma que el modelo de Naïve Bayes implementado funciona de manera consistente en diferentes divisiones del conjunto de datos, proporcionando una estimación fiable de su rendimiento general.

PARTE B

Para esta parte analizaremos cada una de las estrategias:

ESTRATEGIA 1: AMPLIAR TAMAÑO DE ENTRENAMIENTO (DICCIONARIO VARIABLE)

```
Estrategia 1: Aumentar tamaño de entrenamiento (diccionario variable)
Tamaño Train: 20000, accuracy: 0.5915
Tamaño Train: 40000, accuracy: 0.5641
Tamaño Train: 60000, accuracy: 0.5598
Tamaño Train: 80000, accuracy: 0.5528
```

Tendencia del accuracy: Vemos que el accuracy disminuye al incrementar el tamaño del conjunto de entrenamiento. Esto puede deberse a que al añadir más datos se incorporan más palabras que aportan ruido o información menos relevante, dificultando la generalización del modelo.

Impacto en el diccionario: Otro concepto a tener en cuenta, es que al aumentar el conjunto de entrenamiento, el diccionario también crece incorporando más palabras que pueden ser menos informativas o más redundantes.

Conclusión: Ampliar el conjunto de entrenamiento sin controlar el tamaño del diccionario puede resultar en una disminución del accuracy, posiblemente debido a la introducción de más ruido en los datos.

ESTRATEGIA 2: MODIFICAR EL TAMAÑO DEL DICCIONARIO MANTENIENDO FIJO EL DE ENTRENAMIENTO

```
Estrategia 2: Modificar tamaño del diccionario
Frecuencia mínima: 1, accuracy: 0.6626
Frecuencia mínima: 3, accuracy: 0.6238
Frecuencia mínima: 5, accuracy: 0.6075
Frecuencia mínima: 10, accuracy: 0.5873
```

Relación entre min_freq y accuracy: A medida que aumenta min_freq, la accuracy del modelo disminuye. Esto se debe a que al incrementar min_freq se eliminan palabras menos frecuentes del diccionario, lo que puede llevar a perder términos que aportan información relevante para la clasificación.

Impacto en el diccionario: Con min_freq=1, el diccionario incluye todas las palabras, lo que proporciona una cobertura completa pero también puede introducir ruido. Al aumentar min_freq se reduce el tamaño del diccionario, eliminando palabras que aparecen muy pocas veces.

Conclusión: Mantener un diccionario más amplio (menor min_freq) favorece una mayor accuracy, ya que incluye más información, aunque con el riesgo de introducir palabras menos informativas e útiles.

ESTRATEGIA 3: AMPLIAR TAMAÑO DE ENTRENAMIENTO (DICCIONARIO FIJO)

```
Estrategia 3: Modificar tamaño del conjunto de entrenamiento (diccionario fijo)
Tamaño Train: 20000, accuracy: 0.5473
Tamaño Train: 40000, accuracy: 0.5631
Tamaño Train: 60000, accuracy: 0.5724
Tamaño Train: 80000, accuracy: 0.5797
```

Tendencia del accuracy: El accuracy aumenta al incrementar el tamaño del conjunto de entrenamiento cuando el diccionario se mantiene fijo. Esto indica que con más datos proporcionados, el modelo puede aprender mejor las probabilidades condicionales sin introducir ruido adicional.
Impacto en el diccionario: Al mantener el diccionario fijo se evita la inclusión de palabras nuevas que podrían ser ruidosas, permitiendo de tal manera que el modelo se enfoque en un conjunto de palabras estable y relevante.
Conclusión: Mantener un diccionario fijo y aumentar el conjunto de entrenamiento mejora el accuracy del modelo, ya que se dispone de más información sobre las mismas palabras, fortaleciendo las probabilidades condicionales.

PARTE A

ESTRATEGIA 1: AMPLIAR TAMAÑO DE ENTRENAMIENTO CON LAPLACE SMOOTHING

```
Estrategia 1: Aumentar tamaño de entrenamiento con Laplace Smoothing
Tamaño Train: 20000, accuracy: 0.7522
Tamaño Train: 40000, accuracy: 0.7508
Tamaño Train: 60000, accuracy: 0.7614
Tamaño Train: 80000, accuracy: 0.7616
```

Impacto de Laplace Smoothing: Al aplicar Laplace smoothing, se mejora significativamente la exactitud del modelo en comparación con las estrategias sin el suavizado. Las accuracies en esta estrategia son superiores (entre 75.22% y 76.16%) frente a las obtenidas sin suavizado.

Tendencia de accuracy: El accuracy se mantiene estable o mejora ligeramente al incrementar el tamaño del conjunto de entrenamiento sobre todo en los tamaños más grandes lo que significa que el suavizado ayuda a manejar mejor las palabras poco frecuentes o desconocidas.
Conclusión: El uso de Laplace smoothing mejora la capacidad del modelo para manejar palabras con baja frecuencia, evitando probabilidades cero y permitiendo una mejor generalización cuando se incrementa el tamaño del conjunto de entrenamiento.

ESTRATEGIA 2: VARIAR TAMAÑO DE DICCIONARIO CON LAPLACE SMOOTHING (TRAIN FIJO)

```
Estrategia 2: Variar tamaño del diccionario con Laplace Smoothing (train fijo)
Frecuencia mínima: 1, accuracy: 0.7597
Frecuencia mínima: 3, accuracy: 0.7514
Frecuencia mínima: 5, accuracy: 0.7462
```

Relación entre min_freq y accuracy con Laplace Smoothing: Al igual que sin suavizado, al incrementar min_freq y reducir el tamaño del diccionario el accuracy disminuye. Sin embargo, con Laplace smoothing, el impacto negativo es menos pronunciado ya que el suavizado rebaja el efecto de eliminar palabras potencialmente informativas.
Impacto en el diccionario: Al aumentar min_freq se reduce el diccionario eliminando palabras menos frecuentes. Con Laplace smoothing, el modelo maneja mejor las palabras desconocidas manteniendo una buena accuracy incluso con diccionarios más reducidos.
Conclusión: Aunque el accuracy nuevamente disminuye al aumentar min_freq, Laplace smoothing ayuda a mantener una exactitud relativamente alta, mejorando la robustez y firmeza del modelo frente a palabras desconocidas.

ESTRATEGIA 3: VARIAR TAMAÑO DE ENTRENAMIENTO CON LAPLACE SMOOTHING (DICCIONARIO FIJO)

```
Estrategia 3: Mantener diccionario fijo y variar tamaño de entrenamiento con Laplace Smoothing
Tamaño Train: 20000, accuracy: 0.7406
Tamaño Train: 40000, accuracy: 0.7533
Tamaño Train: 60000, accuracy: 0.7562
Tamaño Train: 80000, accuracy: 0.7597
```

Impacto de Laplace Smoothing: Con el suavizado, EL accuracy mejora consistentemente al incrementar el tamaño del conjunto de entrenamiento, alcanzando una exactitud de hasta 75.97%.
Tendencia del accuracy: El accuracy aumenta progresivamente con más datos indicando que el modelo se beneficia y mejora de un mayor conjunto de entrenamiento sin añadir ruido al mantener el diccionario fijo.
Conclusión: Mantener un diccionario fijo y aumentar el tamaño del conjunto de entrenamiento junto con Laplace smoothing mejora significativamente la accuracy del modelo, mostrando la efectividad del suavizado para manejar palabras raras y optimizar el aprendizaje.

CONCLUSIÓN

Como conclusión, podemos sacar ciertas vistas gracias al proyecto, algunas de las cuales con como los siguientes puntos a mencionar.
El modelo Naïve Bayes es una herramienta sencilla pero efectiva para la clasificación de sentimientos en tweets. A pesar de su simplicidad logra una accuracy razonable, especialmente cuando se aplica Laplace Smoothing.
Por lo que se puede decir que el suavizado de Laplace mejora significativamente la accuracy del modelo al evitar probabilidades cero para palabras no vistas durante el entrenamiento como ya he mencionado. Esto resulta especialmente beneficioso cuando se manejan vocabularios grandes y conjuntos de datos extensos
Por otra parte, mencionar el impacto del tamaño del conjunto de entrenamiento y del diccionario.Ya que sin suavizado, aumentar el tamaño del train puede introducir ruido y disminuir la exactitud. Con suavizado, la exactitud mejora conforme se incrementa el conjunto de entrenamiento. También, reducir el diccionario mediante min_freq disminuye la exactitud, pero Laplace smoothing mitiga este. Incrementar el

tamaño del conjunto de entrenamiento con un diccionario fijo y Laplace Smoothing mejora la exactitud, mostrando una mejor capacidad de generalización del modelo. Por último, la utilización de k-fold cross-validation proporciona una estimación más fiable y estable de la accuracy del modelo, lo que es crucial para evaluar su rendimiento de manera objetiva.