

TEORÍA DE LA INFORMACIÓN

Proyecto

Carlos Alexis Barrios Bello
zS23000636@estudiantes.uv.mx

Maestría en Inteligencia Artificial

IIIA Instituto de Investigaciones en Inteligencia Artificial
Universidad Veracruzana
Campus Sur, Calle Paseo Lote II, Sección 2a, No 112
Nuevo Xalapa, Xalapa, Ver., México 91097

11 de diciembre de 2024

1. Introducción

A lo largo de este proyecto, se intentará replicar el artículo llamado “*A novel Bayes Model: Hidden Naive Bayes*” escrito por Liangxiao Jiang, Harry Zhang, and Zhihua Cai Jiang et al. (2009). Este artículo crean un nuevo derivado de Naives Bayes donde se mejora la fuerte suposición de la independencia condicional de las características haciendo uso de **padres ocultos**.

En este proyecto se intentará calcular las mismas precisiones de cada base de datos utilizada en el artículo y, además, se implementará un grafo donde se indique qué atributos son padres e hijos de otros atributos. Lo anterior no se implementa en el código del artículo, lo cual puede ayudar en comprender qué atributos son más importantes en cualquier base de datos.

2. Marco Teórico

2.1. Hidden Naive Bayes (HNB)

El modelo Hidden Naive Bayes (HNB), propuesto en el artículo “*A novel Bayes Model: Hidden Naive Bayes*”, es una extensión del modelo Naive

Bayes que incorpora dependencias ocultas entre los atributos. Este modelo mejora la precisión al relajar la fuerte asunción de independencia condicional del Naive Bayes mediante la introducción de **padres ocultos**.

La codificación de este modelo se implementará en Python Van Rossum and Drake (2009) en el IDE de Jupyter Notebook, para el uso de las bases de datos se utilizó Pandas McKinney et al. (2010), para las operaciones se usó Numpy Harris et al. (2020), para las visualizaciones Matplotlib Hunter (2007), para la evaluación de las métricas y aplicar validación cruzada con K-folds se utilizó scikit-learn Pedregosa et al. (2011).

2.2. Cálculo de Probabilidades de Clase ($P(C)$)

El modelo calcula las probabilidades de clase $P(C)$ utilizando **M-estimación**, la cual suaviza los conteos para evitar probabilidades nulas. La probabilidad de una clase C se define como:

$$P(C) = \frac{\text{count}(C) + \frac{\epsilon}{k}}{\text{total samples} + \epsilon} \quad (1)$$

donde ϵ es un factor de suavizado (Laplace) y k es el número de clases. Esta fórmula asegura que incluso las clases con pocos datos tengan una probabilidad no nula.

2.3. Cálculo de Probabilidades Condicionales

Para modelar las relaciones entre los atributos y las clases, el HNB calcula las probabilidades condicionales $P(a_i|C)$ y $P(a_i|a_j, C)$ usando la fórmula de M-estimación:

$$P(a_i|C) = \frac{\text{count}(a_i, C) + \frac{\epsilon}{n_i}}{\text{count}(C) + \epsilon} \quad (2)$$

$$P(a_i|a_j, C) = \frac{\text{count}(a_i, a_j, C) + \frac{\epsilon}{n_{ij}}}{\text{count}(a_j, C) + \epsilon} \quad (3)$$

donde n_i y n_{ij} representan los valores únicos del atributo a_i y el par (a_i, a_j) , respectivamente.

2.4. Cálculo de Información Mutua Condicional

El HNB mide las dependencias entre atributos mediante la **información mutua condicional**:

$$I(A_i; A_j|C) = \sum_{a_i, a_j} P(a_i, a_j|C) \log \frac{P(a_i, a_j|C)}{P(a_i|C)P(a_j|C)} \quad (4)$$

Esto permite identificar qué atributos tienen una fuerte dependencia en presencia de una clase C .

2.5. Cálculo de Pesos W_{ij}

Los pesos W_{ij} representan la fuerza de la relación entre los atributos A_i y A_j en una clase C , y se calculan como:

$$W_{ij} = \frac{I(A_i; A_j | C)}{\sum_{k \neq i} I(A_i; A_k | C)} \quad (5)$$

El uso de W_{ij} permite modelar dependencias ocultas al asignar **padres ocultos** a cada atributo.

2.6. Padres Ocultos y Visualización

Con base en los pesos W_{ij} , se seleccionan los padres ocultos de cada atributo mediante un umbral:

$$W_{ij} > \text{umbral} \quad (6)$$

El grafo resultante, generado con la biblioteca *networkx*, visualiza las relaciones entre atributos y sus padres ocultos. Los pesos promedio W_{ij} son mostrados como etiquetas en las aristas del grafo para interpretar la importancia de las relaciones.

2.7. Predicción

La probabilidad de un atributo dado sus padres ocultos se calcula como:

$$P(a_i | A_{\text{hp}_i}, C) = \sum_{j \neq i} W_{ij} \cdot P(a_i | a_j, C) \quad (7)$$

Finalmente, las probabilidades combinadas permiten predecir la clase C con máxima probabilidad:

$$\hat{C} = \arg \max_C \left[\log P(C) + \sum_i \log P(a_i | A_{\text{hp}_i}, C) \right] \quad (8)$$

3. Bases de datos

Cuadro 1: Description of Data Sets Used in the Experiments

No.	Dataset	Inst	Ft	Class	Miss. Va	Num. Va
1	audiology	226	70	24	Y	N
2	autos	205	26	7	Y	Y
3	balance-scale	625	5	3	N	Y
4	breast-cancer	286	10	2	Y	N
5	breast-w	699	10	2	N	Y
6	colic.ORIG	368	25	2	Y	Y
7	credit-a	690	16	2	Y	Y
8	credit-g	1000	21	2	N	Y
9	diabetes	768	9	2	N	Y
10	Glass	214	10	7	N	Y
11	heart-c	303	14	5	Y	Y
12	heart-h	294	14	5	N	Y
13	heart-statlog	270	14	2	N	Y
14	hypothyroid	3772	30	4	Y	Y
15	ionosphere	351	35	2	N	Y
16	iris	150	5	3	N	Y
17	kr-vs-kp	3196	37	2	N	N
18	labor	57	17	2	Y	Y
19	letter	20000	17	26	N	Y
20	lymph	148	19	4	N	Y
21	mushroom	8124	23	2	Y	N
22	primary-tumor	339	18	21	Y	N
23	segment	2310	20	7	N	Y
24	soybean	683	36	19	Y	N
25	splice	3190	62	3	N	N
26	vehicle	846	19	4	N	Y
27	vote	435	17	2	Y	N
28	waveform-5000	5000	41	3	N	Y
29	zoo	101	18	7	N	Y

Con respecto al conjunto de base de datos, sólo se pudieron encontrar 29 de 36 aplicados. La base de datos “colic.ORIG” se presentó un problema, las características llamadas “25, 26, 27: type of lesion” tiene una codificación de números, sin embargo, no sigue la codificación como se describe en la

información de la base de datos, no se pudo saber qué tipo de lesión tiene cada instancia, así que se procedió a eliminar esas tres columnas, restando 25 columnas. El dataset “heart-h” no tiene valores faltantes, el artículo se equivocó en ello.

4. Preprocesamiento de los datos

Se siguieron los pasos del mismo preprocesamiento de los datos:

1. Rellenamiento de los datos faltantes con el filtro no supervisado llamado “ReplaceMissingValues” en Weka.
2. Discretizar las columnas numéricas con el filtro “Discretize” en Weka, utilizando 10 bins y como complemento, se usó “UseBinsNumbers” con “True”. Lo anterior para que lo devolviera como un entrada fácil de limpiar.
3. Después de aplicar la discretización, Weka no hace la discretización como comúnmente se hace, que es convertirlo a una clase categórica común, como ponerle números o letras. En realidad, hace algo así: ‘B1of10’. La implementación en python no está hecha para leer algo así, por ello se implementó otra limpieza para sólo conservar el número que está a lado de la B. Esa limpieza fue un ligero script en python.
4. Se removieron atributos inútiles o que le pudieran meter ruido al modelo:
 - colic.ORIG se eliminó ‘Hospital Number’ ya que son valores únicos y no proporcionan información relevante.
 - Ionosphere se eliminó ‘O2’ ya que sólo tiene una clase.
 - Segment se eliminó ‘region-pixel-count’ ya que sólo tiene una clase.
 - Splice se eliminó ‘id’ ya que son puros valores únicos y no proporcionan información relevante.
 - Zoo se eliminó ‘animal’ ya que son puros valores únicos.

El artículo menciona que eliminaron la columna ‘instance name’ de splice, sin embargo no hay ninguna columna con ese nombre, se hace la suposición de que eliminaron ‘id’.

5. No se hizo uso de hacer submuestreos de la base de datos ya que en python el tiempo de ejecución aplicando validación cruzada con k-folds es menor a 15 seg por cada base de datos.

5. Modelo para comparar

Sólo se tiene implementado el modelo de Naive Bayes. El cual se explica así:

5.1. Naive Bayes

Este clasificador se basa en el Teorema de Bayes Marsland (2009), asume que las características o atributos de entrada son condicionalmente independientes entre sí, esta es la ingenuidad del clasificador, dado el resultado de la clase, lo que simplifica en gran medida el cálculo de probabilidades. La fórmula básica del teorema es:

$$P(C_i|x_i) = \frac{P(x_i|C_i)P(C_i)}{P(x_i)} \quad (9)$$

donde $P(C|x)$ es la probabilidad posterior de la clase C dado el vector de características x , $P(x|C)$ es la probabilidad de observar los datos x dado que la clase es C , $P(C)$ es la probabilidad previa de la clase C y $P(x)$ es la probabilidad marginal de observar x .

Como el clasificador asume que cada característica es independiente de las demás, dada la clase, facilita el cálculo $P(x|C)$ como el producto de probabilidades individuales de cada característica:

$$P(x|C) = \prod_k P(x_k|C). \quad (10)$$

Además, el clasificador maximiza la probabilidad posterior (MAP), lo cual se puede simplificar omitiendo el denominador de la ecuación 9 $P(x)$, ya que es constante para todas las clases

$$C^* = \underset{C}{\operatorname{argmax}} P(C) \prod_k P(X_k, C) \quad (11)$$

con C^* para la característica seleccionada.

6. Resultados

Cuadro 2: Comparison of Naive Bayes and Hidden Naive Bayes on Various Data Sets (Accuracy)

DataSet	Naive Bayes		Hidden Naive Bayes	
	Accuracy	\pm Std	Accuracy	\pm Std
audiology	0.783004	0.095021	0.211858	0.091011
auto	0.711429	0.077246	0.731429	0.088229
balance-scale	0.918459	0.29794	0.905658	0.033590
breast-cancer	0.727956	0.069828	0.709852	0.059482
breast-w	0.971346	0.014453	0.972795	0.016287
colic.ORIG	0.785135	0.049991	0.774249	0.052499
credit-a	0.844928	0.047207	0.446377	0.078261
credit-g	0.757000	0.043829	0.752000	0.045563
diabetes	0.759040	0.047081	0.765670	0.036062
Glass	0.583983	0.091345	0.560823	0.109659
heart-c	0.551935	0.112679	0.561935	0.080653
heart-h	0.666552	0.068682	0.649310	0.080153
heart-statlog	0.818519	0.088347	0.803704	0.068393
hypothyroid	0.927093	0.013438	0.922852	0.014100
ionosphere	0.894683	0.042143	0.906111	0.047505
iris	0.940000	0.046667	0.946667	0.083267
kr-vs-kp	0.877040	0.020100	0.477785	0.029046
labor	0.913333	0.114698	0.510000	0.158500
letter	0.707800	0.008542	0.766800	0.009185
lymph	0.810000	0.121227	0.693333	0.202425
mushroom	0.991137	0.003431	0.979445	0.003724
primary-tumor	0.441979	0.086650	0.247683	0.076610
segment	0.907359	0.013159	0.746753	0.028005
soybean	0.926726	0.018774	0.190473	0.046214
splice	0.956740	0.009693	0.927273	0.015849
vehicle	0.609818	0.033371	0.651106	0.052283
vote	0.903436	0.040977	0.901216	0.041210
waveform-5000	0.797000	0.015212	0.801600	0.015041
zoo	0.960000	0.048990	0.401818	0.182987

De la anterior tabla se aprecia que en muchas bases de datos NB y HNB son muy similares en cuestión del accuracy, en pocas ocasiones HNB (9/26) le gana a NB, siendo NB el mejor modelo en cuestión de eficiencia y menor complejidad de tiempo al ejecutar dicha implementación. En otros casos, HNB compite muy bien con NB, por milésimas o centésimas pierde.

Sin embargo, se resalta los casos en donde HNB es muy malo, comparado con NB, en accuracy. Se resaltan las bases de datos:

- audiology.
- credit-a.
- kr vs kp.
- labor.
- lymph.
- primary tumor.
- segment.
- soybean.
- zoo.

Se buscaron las semejanzas en estas bases de datos y se encontró la posible causa. Estas tienen en común que sus características (la mayoría, algunas o unas pocas) sean binarias o terciarias. De tal manera que HNB cae en muchos problemas.

1. Los atributos tienen menos valores posibles, lo que puede llevar a un cálculo de probabilidades condicionales menos informativo. Esto hace que las dependencias entre atributos sean menos evidentes y, por lo tanto, los pesos W_{ij} y las relaciones entre padres ocultos sean menos significativos. Las probabilidades condicionales pueden ser dominadas por el suavizado ϵ , lo que genera distribuciones menos diferenciadas entre las clases.
2. Con pocos valores posibles, el término de suavizado puede dominar la probabilidad condicional real, llevándola a valores casi uniformes. Esto reduce la capacidad del modelo para diferenciar entre clases. Las probabilidades condicionales pierden precisión, afectando tanto el cálculo de $P(a_i|C)$ y $P(a_i|a_j, C)$, y por ende, el cálculo de la información mutua y los pesos.

3. Si las dependencias reales entre los atributos son débiles o inexistentes, el modelo podría introducir relaciones artificiales (falsos positivos) debido al cálculo de pesos, especialmente en bases con baja variabilidad. El modelo sobreajusta a estas dependencias ficticias, afectando negativamente el rendimiento en el conjunto de prueba.

Para futuras mejoras, se recomienda hacer un ajuste dinámico de ϵ y realizar un buen algoritmo de discretización a la base de datos.

En el artículo no mencionan el ajuste dinámico de ϵ , sólo dan un valor constante de 1.0, se intentó con ese número, pero el algoritmo daba una peor precisión o simplemente caía en divergencia, intuitivamente, se cambió a un valor menor para que los resultados fueran lo más competitivos con NB.

Con respecto a la métrica CLL (Conditional Log-Likelihood), se obtuvo un resultado muy bueno en comparación con el artículo. Esto se observa en la siguiente tabla.

No se entiende cómo es que obtuvieron esos resultados, parecieran que están normalizados como los de este proyecto, pero multiplicados por algún valor extra. Esto porque, si se calculara CLL de los modelos (NB y HNB) según el artículo, serían muy excéntricos comparados a los del artículo. En esta implementación, CLL se calcula así:

$$\text{CLL}(G \mid D) = -\frac{1}{t} \sum_{i=1}^t \log P_G(c_i \mid \mathbf{a}_i), \quad (12)$$

donde:

- t : Número total de ejemplos en el conjunto de datos,
- \mathbf{a}_i : Vector de atributos del ejemplo,
- c_i : Clase verdadera del ejemplo,
- $P_G(c_i \mid \mathbf{a}_i)$: Probabilidad condicional estimada por el modelo G para la clase verdadera c_i dado el vector de atributos \mathbf{a}_i .

En el artículo no mencionan ese factor de normalización que se usa, y si no lo usará los resultados serían muy grandes (negativamente hablando) de tal forma que no se pueden comparar con el artículo.

Cuadro 3: Comparison of Naive Bayes and Hidden Naive Bayes on Various Data Sets (CLL)

DataSet	Naive Bayes		Hidden Naive Bayes	
	CLL	\mp Std	CLL	\mp Std
audiology	-4.589968	3.169176	-27.601409	2.239432
autos	-5.482678	4.017147	-4.914831	2.537918
balance-scale	-0.502711	0.068797	-0.415859	0.096474
breast-cancer	-0.769257	0.182076	-0.885163	0.158620
breast-w	-0.423097	0.248077	-0.470242	0.284006
colic.ORIG	-1.219625	0.601340	-1.608423	0.735268
credit-a	-0.502109	0.212548	-5.412888	0.750367
credit-g	-0.576274	0.110687	-0.667832	0.148421
diabetes	-0.617889	0.156877	-0.754237	0.156078
Glass	-2.501738	1.167282	-6.257457	2.626289
heart-c	-2.703893	1.152018	-2.788506	1.091700
heart-h	-1.975290	0.832507	-4.663027	1.288760
heart-statlog	-0.734645	0.341309	-0.867538	0.303130
hypothyroid	-0.306540	0.111902	-2.780682	0.508209
ionosphere	-1.323399	0.898979	-1.482938	0.833056
iris	-0.433393	0.579923	-0.465269	0.686224
kr-vs-kp	-0.290597	0.019321	-12.860705	2.214328
labor	-0.448308	0.988335	-7.117861	3.374967
letter	-1.268837	0.045029	-1.209778	0.063885
lymph	-1.783899	2.246274	-2.826875	2.142246
mushroom	-0.022368	0.011774	-0.072628	0.015618
primary-tumor	-4.592137	1.462966	-24.346133	2.621209
segment	-0.434802	0.097671	-2.508987	0.673723
soybean	-0.399293	0.115535	-27.344166	1.271741
splice	-0.163656	0.035299	-0.351659	0.097646
vehicle	-2.067615	0.324553	-2.448110	0.467883
vote	-0.631731	0.264229	-1.037814	0.436564
waveform-5000	-0.783393	0.092561	-1.076816	0.131733
zoo	-0.495264	0.758878	-20.435765	6.515563

CLL fue la peor métrica para HNB, teniendo sólo 3 resultados que le ganaron a NB, mientras que NB dominó completamente esta métrica con 23/26.

6.1. Diagramas

A continuación se mostrarán algunos diagramas y sus especificaciones de cuántos papás e hijos tiene cada característica de la base de datos empleada. A continuación, se da el diagrama de Glass con los pesos arriba de 0.15:

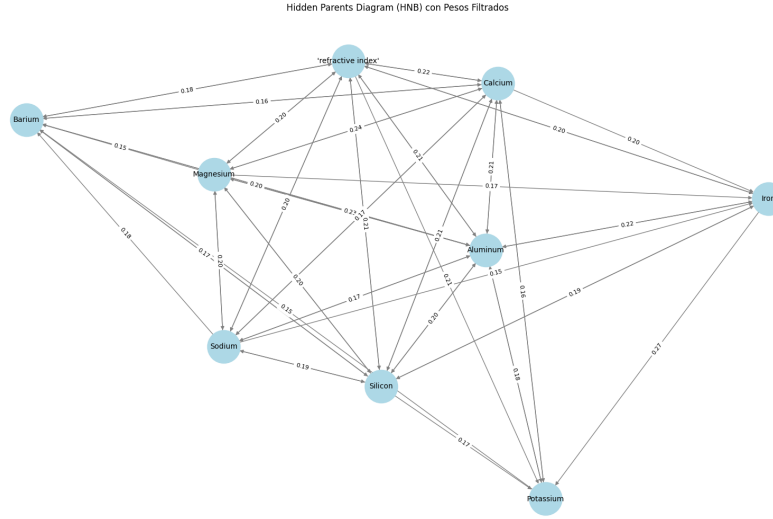


Figura 1: Relaciones entre cada atributo de Glass con $W_{ij} > 0.15$

Para mejor análisis se deja la cantidad de padres e hijos que tiene cada característica significativa de la base de datos.

Análisis de Importancia de Nodos:

- Sodium - Padres (in_degree): 5, Hijos (out_degree): 7
- 'refractive index' - Padres (in_degree): 7, Hijos (out_degree): 8
- Aluminum - Padres (in_degree): 8, Hijos (out_degree): 8
- Silicon - Padres (in_degree): 7, Hijos (out_degree): 8
- Calcium - Padres (in_degree): 7, Hijos (out_degree): 8
- Magnesium - Padres (in_degree): 5, Hijos (out_degree): 7
- Iron - Padres (in_degree): 6, Hijos (out_degree): 4
- Barium - Padres (in_degree): 7, Hijos (out_degree): 4

- Potassium - Padres (in_degree): 5, Hijos (out_degree): 3

Otra base de datos que forma un diagrama muy estético, esto por la cantidad de atributos que tiene es el de Diabetes:

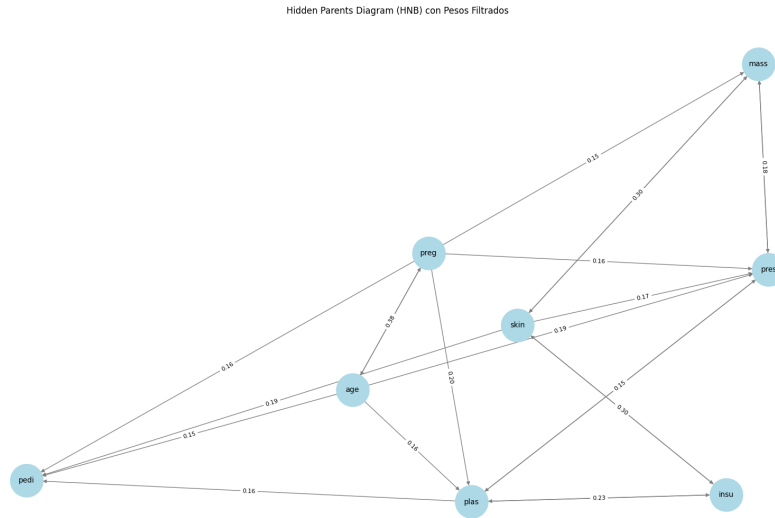


Figura 2: Relaciones entre cada atributo de Diabetes con $W_{ij} > 0.15$

Para mejor análisis se deja la cantidad de padres e hijos que tiene cada característica significativa de la base de datos.

Análisis de Importancia de Nodos:

- age - Padres (in_degree): 1, Hijos (out_degree): 4
- preg - Padres (in_degree): 1, Hijos (out_degree): 5
- pres - Padres (in_degree): 5, Hijos (out_degree): 2
- plas - Padres (in_degree): 4, Hijos (out_degree): 3
- insu - Padres (in_degree): 2, Hijos (out_degree): 2
- skin - Padres (in_degree): 2, Hijos (out_degree): 4
- mass - Padres (in_degree): 3, Hijos (out_degree): 2
- pedi - Padres (in_degree): 4, Hijos (out_degree): 0

Los diagramas se pueden generar para cualquier base de datos, un problema sería la cantidad de características que tenga esa base de datos. En este escrito se escogieron base de datos con pocas características (menos de 10 atributos). Ahora se verá una base de datos que tenga más de 10 atributos.

Base de datos Mushroom (23 atributos):

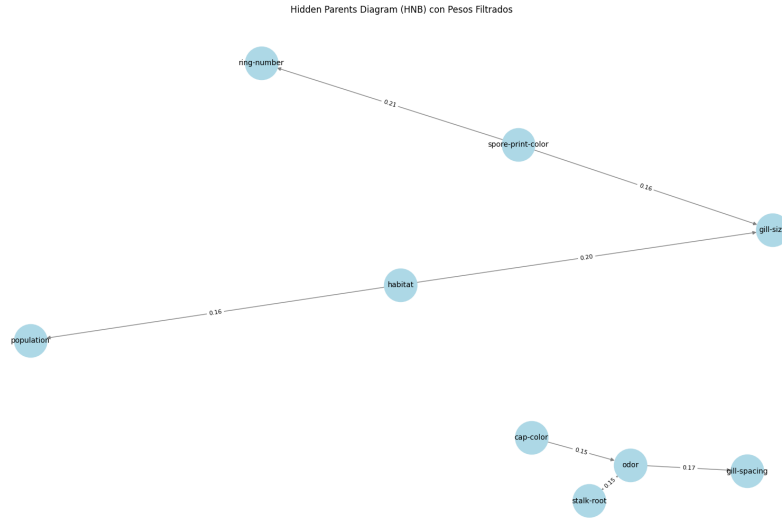


Figura 3: Relaciones entre cada atributo de Mushroom con $W_{ij} > 0.15$

Para mejor análisis se deja la cantidad de padres e hijos que tiene cada característica significativa de la base de datos.

Análisis de Importancia de Nodos:

- cap-color - Padres (in_degree): 0, Hijos (out_degree): 1
- odor - Padres (in_degree): 2, Hijos (out_degree): 2
- stalk-root - Padres (in_degree): 1, Hijos (out_degree): 1
- habitat - Padres (in_degree): 0, Hijos (out_degree): 2
- gill-size - Padres (in_degree): 2, Hijos (out_degree): 0
- spore-print-color - Padres (in_degree): 0, Hijos (out_degree): 2
- ring-number - Padres (in_degree): 1, Hijos (out_degree): 0

- population - Padres (in_degree): 1, Hijos (out_degree): 0
- gill-spacing - Padres (in_degree): 1, Hijos (out_degree): 0

A pesar de tener más atributos que las anteriores bases de datos, sólo muy pocos cumplen la condición de que sus pesos sean mayores a 0.15, lo cual resalta la importancia entre ellos.

7. Conclusiones

En este proyecto se evaluó la implementación del modelo Hidden Naive Bayes (HNB) y su comparación con el Naive Bayes tradicional (NB) en términos de precisión y log-verosimilitud condicional (CLL) en múltiples conjuntos de datos. Aunque el HNB introduce dependencias ocultas para mejorar la precisión en ciertas bases de datos, los resultados demuestran que en la mayoría de los casos, el modelo NB sigue siendo superior debido a su simplicidad y eficiencia computacional.

En el análisis se identificó que el HNB tiene un desempeño inferior en bases de datos con atributos mayormente binarios o terciarios, lo que sugiere que su cálculo de probabilidades condicionales y pesos se ve afectado negativamente en estos contextos. Esto se debe a la limitada diferenciación de distribuciones en estas características, lo que conduce a un menor rendimiento.

A pesar de estas limitaciones, en este proyecto se destacó la utilidad de visualizar las relaciones entre atributos mediante grafos de padres ocultos, lo que ofrece una herramienta adicional para interpretar la importancia relativa de los atributos en cada base de datos.

Para mejorar el desempeño del HNB, se recomienda explorar ajustes dinámicos del factor de suavizado (ϵ) y optimizar los algoritmos de discretización. Estas mejoras podrían mitigar los problemas observados y permitir un uso más robusto del modelo en una variedad más amplia de conjuntos de datos.

En conclusión, aunque el HNB tiene potencial para capturar dependencias complejas entre atributos, su implementación y aplicación requieren ajustes cuidadosos para superar las limitaciones observadas en este trabajo. Lo anterior aplicado a diferentes bases de datos.

Referencias

- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.
- Jiang, L., Zhang, H., and Cai, Z. (2009). A novel bayes model: Hidden naive bayes. *IEEE Transactions on Knowledge and Data Engineering*, 21(10):1361–1371.
- Marsland, S. (2009). *Machine Learning - An Algorithmic Perspective*. Chapman and Hall / CRC machine learning and pattern recognition series. CRC Press.
- McKinney, W. et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.