Clasificación con KNN y Naive Bayes

Marisol Flores y
Luis Miguel García Velázquez
Profesores LTIC
ENES Morelia (UNAM)
Email: mflores@enesmorelia.unam.mx,
luism_garcia@enesmorelia.unam.mx

Ibeth Escobedo Rios
Estudiante
LTIC
ENES Morelia (UNAM)
Email: ibtescobedo@gmail.com

Leonardo Ariel Tapia Figueroa
Estudiante
LTIC
ENES Morelia (UNAM)
Email: leoatapia309@gmail.com

***Resumen*—Una subrama de la Inteligencia Artificial es el aprendizaje supervisado, el cual nos ayua a "predecir las etiquetas de clase categóricas de nuevos registros, con base en observaciones pasadas"[1]. Por eso en este proyecto vamos a ver las diferencias al utilizar Naive Bayes y KNN, dos métodos de clasificación distintos los cuales su objetivo final es predecir si un alumno se graduará o desertará de acuerdo a ciertos factores.**

I. Introducción

En la actualidad cada vez más personas tienen acceso a la educación pero existen diferentes factores ya sea externos o internos que afectan las probabilidades de que un alumno se gradue o no; entre ellos: la nacionalidad, edad, género, inflación en el país que residen, tasa de desempleo y claro rendimiento en la escuela. Analizar este tipo de datos no solo nos da información requerida para clasificar sino también nos puede abrir un panorama más amplio para ubicar ciertos factores que pueden facilitar la deserción escolar. De esta manera se pueden localizar a los alumnos que tienen más riesgo de sufrir un abandono escolar y buscar soluciones o exponer los resultados para que un experto en el tema opine.

Por eso en este proyecto se puede observar la utilización y manipulación de cierto tipo de datos utiizando 2 diferentes métodos, especificamente KNN y Naive Bayes para clasificar.

Junio 19, 2022

II. Materiales y métodos

Los métodos utilizados pertenecen al aprendizaje supervisado, los cuales son algoritmos trabajan con datos "etiquetados", intentado encontrar una función que, dadas las variables de entrada, les asigne la etiqueta de salida adecuada. El algoritmo se entrena con un "histórico" de datos y así "aprende" a asignar la etiqueta de salida adecuada a un nuevo valor, es decir, predice el valor de salida.[2]

Los datos a utilizar se bajaron de una plataforma llamada kaggle con el titulo "Predict Dropout or Academic Success"[3] con más de 4000 registros. Los atributos que contiene este dataset son: Estado civil, Modo de solicitud, Orden de solicitud, Curso, Asistencia diurna/nocturna, Título anterior, Título previo (grado), Nacionalidad, Calificación de la madre, Calificación del padre, Ocupación de la madre, Ocupación del padre, Grado de admisión, Desplazado, Necesidades educativas especiales, Deudor, Tasas de matrícula al día, Género, Titular de la beca, Edad de inscripción, Internacional, Unidades curriculares 1er sem (acreditado), Unidades curriculares 1er sem (matriculados), Unidades curriculares 1er sem (evaluaciones), Unidades curriculares 1er sem (aprobado), Unidades curriculares 1er sem (grado), Unidades curriculares 1er sem (sin evaluaciones), Unidades curriculares 2º sem (acreditado), Unidades curriculares 2º sem (matriculados), Unidades curriculares 2º sem (evaluaciones), Unidades curriculares 2º sem (aprobado), Unidades curriculares 2º sem (grado), Unidades curriculares 2º sem (sin evaluaciones), Tasa de desempleo, Tasa de inflación, PIB, y Target. Lo que vamos a predecir es Target la cual viene dividida en 3 partes que es, graduado, deserto e inscrito, los alumnos que estan inscritos no resultan útiles a la hora de predecir así que se eliminó todas las filas que estén clasificadas de esa manera. Tambien se observo que el curso no servía para la clasificación ya que la distribución está sesgada a la derecha (ver Fig. 1).

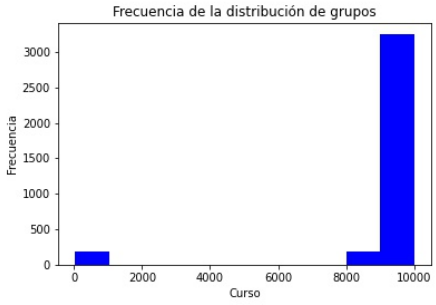

Figura 1. La distribución se inclina a solo un valor lo que deja a 33 en un valor distinto y más de 4000 datos van a caer en un solo valor

Una vez analizados los atributos y tras examinar los demás, no se realizaron más cambios. (ver Fig. 2) Estas fueron las únicas modificaciones que se le hizo al dataset de forma general para el uso de los métodos.

	Mother's qualification	Father's qualification	Curricular units 1st sem (credited)	Curricular units 2nd sem (credited)
Mother's qualification	1.000000	0.535140	0.046855	0.042771
Father's qualification	0.535140	1.000000	0.040054	0.042666
Curricular units 1st sem (credited)	0.046855	0.040054	1.000000	0.944811
Curricular units 2nd sem (credited)	0.042771	0.042666	0.944811	1.000000

Figura 2. Se aprecía la correlación de algunos de los principales campos

II-A. KNN (vecinos cercanos)

El algoritmo KNN asume que hay elementos similares demasiado cerca. En otras palabras, las cosas similares están cerca unas de otras. Esto se basa en la idea de distancia o cercanía [4] "Para cada instancia sin clasificar, el algoritmo localiza las k instancias más similares en el conjunto de entrenamiento. La clase predominante entre las k instancias seleccionadas representa la predicción para la clase de la instancia sin clasificar."[5]

Para poder implementar este algoritmo la Dra. Flores sugiere:[5]

- Decidir cuanto vale k
- Para cada instancia sin clasificar:
  - Medir las distancias contra todos los conjuntos de entrenamiento
  - Decidir quienes son la k más cercana
  - Elegir cual clase aparece más veces

Se eligio este método por los tipos de datos que tenemos ya que todos son númericos y pueden ser facilmente representados con distancias, entre menos distancia exista más parecida será con cierta clase.

II-A1. Preparación de los datos: Al utilizar este método es necesario tener todo en la misma escala porque de lo contrario algunos atributos tendrian más influencia que otros, la manera de normalizar fue de la siguiente manera:

$$Z = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

La ecuación 1 normaliza un valor de una lista de valores dando como resultado un valor entre 0 y 1.

II-A2. División de los datos: Los datos se dividiran en x_train, x_test, y_train y y_test para esto se utilizaron una función ya creada de la libreria sklearn. Lo usual es separar entre 70 % y 80 % los datos de entrenamiento y en este caso utilizamos el 80 % para los datos de entrenamiento y 20 % para los datos de validación, de forma aleatoria.

II-A3. Implementación de KNN: De sklearn.neighbors se importa KNeighborsClassifier[6] con esta libreria es fácil hacer la clasificación solo se le pasan los datos ya preparados que fue lo que se hizo anteriormente. Ahora tenemos que asignarle un valor a k por lo cual se hace un análisis del error promedio intentando con diferentes valores de k (ver Fig. 3) como se observa en la figura donde el error es menor es cuando $k = 20$ por lo que implementamos en algoritmo con esa k. Queda de esta manera KNeighborsClassifier(n_neighbors = 20) y despues se ajusta.

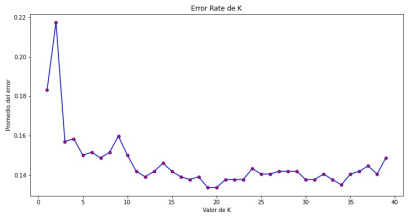

Figura 3. Error promedio al utilizar desde k = 1 hasta k = 40

II-B. Naive Bayes

Este método como su nombre lo indica se basa en el teorema de Bayes 2 [7]

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2}$$

Por lo que los predictores deben ser independientes, se utiliza principalmente para analizar sentimientos, recomendación y spam ya que lo que hace es predecir mediante semejanza y probabilidad. Se eligio este algoritmo porque los datos se separan en dos clases y cada una se obtiene por factores distintos que tienen correlación entre si, lo que hace perfecto a Naive Bayes para predecir si un alumno se acerca más a las caracteristicas de un graduado o de un desertor.

II-B1. Preparación de los datos: Para la preparación correcta de los datos fue necesario normalizar los datos de 0 a 1, para posteriormente poder discretizarlos en 5 categorías cada una con un intérvalo de 0.20 unidades, que se clasificaron con números enteros del 0 al 4. Los datos que se discretizaron fueron aquellos que tienen que ver con una calificación(grade) y la edad. Para normalizarlos se utilizó la misma fórmula que en el método de KNN (1).

II-B2. División de los datos: Al igual que la división de datos que se hizo con KNN se hará aquí, dejando 80 % de datos para el entrenamiento y 20 % para la validación.

II-B3. Implementar Naive Bayes: La implementación que se hizo fue con suavizamiento Laplace con $\lambda = 1$ esto porque la probabilidad condicional de la aparición de un determinado valor propio puede ser 0 y esto afectaría a los resultados. El algoritmo que se utilizó fue hecho de forma casera [7] donde se creó una función la cual recibe como parametros de entrada X_train, X_test y y_train devolviendo y_pred que es la predicción hecha.

III. Experimentos y resultados

III-A. Analisis de los resultados

Para cada uno de los métodos utilizados se obtuvo un reporte similar por lo que se pueden analizar cada uno y despues compararlos.

Para comprender un poco mejor lo que se esta mostrando en las tablas I y II:

- macro avg = promedio de la media no ponderada por etiqueta
- weighted = promedio de la media ponderada por soporte por etiqueta
- Medir las distancias contra todos los conjuntos de entrenamiento
- f1-score = es la media armónica de precisión y recuperación
- support = es el número de muestras de la respuesta verdadera que se encuentran en esa clase

Tenemos el reporte de los resultados de KNN (ver cuadro I) la cual obtuvo un desempeño bueno donde se obtuvo un peor resultado fue en recall al tratar de predecir los graduados.

	Precision	recall	f1-score	support
Dropout	0.91	0.71	0.80	270
Graduate	0.85	0.96	0.90	456
macro avg	0.88	084	0.85	726
weighted avg	0.87	0.87	0.86	726

Cuadro I
REPORTE DE KNN

Por otro lado el reporte de los resultados de la utilización de Naive Bayes (ver cuadro II) la cual obtuvo un desempeño muy bueno y los resultados son uniformes no se dispersan tanto.

	Precision	recall	f1-score	support
Dropout	0.87	0.84	0.85	280
Graduate	0.90	0.92	0.91	446
macro avg	0.88	088	0.88	726
weighted avg	0.89	0.89	0.89	726

Cuadro II
REPORTE DE NAIVE BAYES

Los resultados tienen sentido ya que como vimos la correlación es buena para este tipo de clasificaciones (ver Fig. 4).

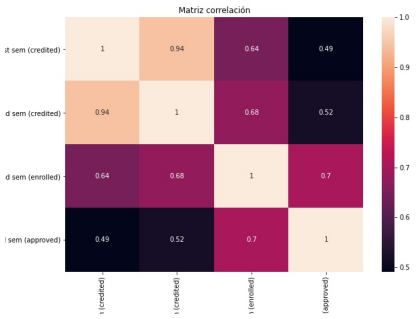

Figura 4. Se aprecía la correlación de algunos de los principales atributos

III-B. Matriz de confusión y accuracy

La matriz de confusión nos ayuda a apreciar la distribución de como clasifico el algoritmo.La ecuación para la presición es $P = \frac{TP}{TP+FP}$ y para la sensibilidad es $R = \frac{TP}{TP+FN}$.

Comenzando a analizar la matriz obtenida de Knn (ver matriz 3) como se puede observar se obtuvo muchos más predicciones erroneas en Falsos positivos por lo que este algoritmo esta inclinado a tener más presición que sensibilidad. (ver Fig. 5)

$$\begin{bmatrix} 193 & 77 \\ 20 & 436 \end{bmatrix} \tag{3}$$

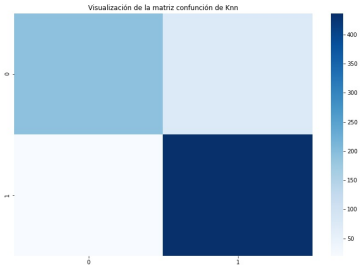

Figura 5. Matriz de confusión KNN

Acuraccy KNN: 0.87

La matriz de confusión de Naive Bayes se distribuyen los errores entre FP y FN además es un poco menor el error (ver matriz 4) a simple vista de la matriz se considera como un resultado aceptable (ver Fig. 6)

$$\begin{bmatrix} 234 & 46 \\ 36 & 410 \end{bmatrix} \tag{4}$$

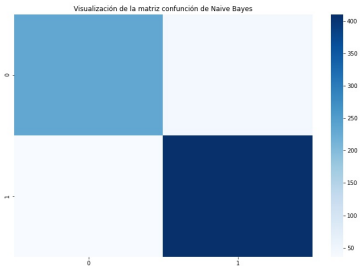

Figura 6. Matriz de confusión Naive Bayes

Accuracy Naive Bayes: 0.89

III-C. Comparación

Por un lado tenemos un algoritmo ya implementado (knn) y por otro un algoritmo hecho caseramente desde 0, así que se podria decir que por obvias razones quien tuviera mejor resultado sería el algoritmo ya hecho con sus propias librerias, pero en este caso no es solo eso lo que influye a los resultados sino tambien la forma de los datos y la manera en la que funciona cada metodo. En cuanto accuracy casí quedaron iguales pero obtuvo más Naive Bayes por una diferencia de 0.02 parece poco pero a gran escala se puede reflejar de una

manera más grave. Las diferencias en la matriz de confusión son muy visibles ya que utilizando Knn se tiene mas error en recall, miestras que en precision tienen casí el mismo error.

Una manera rapida de visualizar lo que anteriormente mencione es viendo el (Cuadro III) en el cual se coloco de manera seguida los resultados para ser mejor observados.

	Knn Precision	Naive Precision	Knn recall	Naive recall	Knn f1-score	Naive f1-score	Knn support	Naive support
Dropout	0.91	0.87	0.71	0.84	0.80	0.85	270	280
Graduate	0.85	0.90	0.96	0.92	0.90	0.91	465	446
macro avg	0.88	088	0.84	0.88	0.85	0.88	726	726
weighted avg	0.87	0.89	0.87	0.89	0.86	0.89	726	726

Cuadro III
TABLA PARA VIZUALIZAR LAS DIFERENCIAS

IV. CONCLUSIONES

En conclusión si se tuviera que elegir un método para decir que es mejor en este conjunto de datos se llegaría a decir que es Naive Bayes porque aquí lo que nos interesa es que la mayor cantidad de alumnos se graduen y si nos sale en el resultado que un alumno si se va a graduar y al último no se graduan no se pudo actuar a tiempo y son alumnos descuidados por eso aqui nos importa mas la sensibilidad que la presición, sumando que tambien obtuvo más accuracy. Los dos metodos funcionaron bien, se puede predecir el abandono escolar y con esto se puede buscar técnicas para prevenirlo.

AGRADECIMIENTO

Los autores queremos agradecer a nuestros profesores Marisol Flores, Luis Miguel García y a nuestro ayudante de clase Javier Navarro, que fueron parte fundamental en la obtención de los conocimientos que llevó a la realización de este estudio.

REFERENCIAS

[1] Rodriguez Galindo, D. (2021, March 28). Aprendizaje Supervisado. CII.IA. https://www.ciiia.mx/noticiasciiia/aprendizaje-supervisado-1

[2] Simeone O. (2018). Very Brief Introduction to Machine Learning With Applications to Communication Systems. IEEE Transactions on Cognitive Communications and Networking, 4(4), 648–664. https://doi.org/10.1109/tccn.2018.2881442

[3] Predict Dropout or Academic Success. (2022, 5 junio). Kaggle. https://www.kaggle.com/datasets/ankanhore545/dropout-or-academic-success

[4] Harrison, O. (2019, 14 julio). Machine Learning Basics with the K-Nearest Neighbors Algorithm. Medium. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

[5] Flores, M. (2022, 23 marzo). k Nearest Neighbors (kNN) [Diapositivas]. s/p. https://drive.google.com/file/d/1No91NjXCtpedIA1F7z9KPabWoIoYwt6O/view

[6] Nearest Neighbors. Scikit-Learn. https://scikit-learn.org/stable/modules/neighbors.html

[7] Roman, V. (2021, 9 diciembre). Algoritmos Naive Bayes: Fundamentos e Implementación. Medium. https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fudamentos-e-implementaci %C3 %B3n-4bcb24b307f

[8] Escobedo, I. (2022, 19 junio). Prediccion KNN Naive-Bayes graduados desertados: Mediante algoritmos de clasificación se busca predecir si un alumno se graduara o desertara. GitHub. https://github.com/IbtIbeth/Prediccion_KNN_Naive-Bayes_graduados_desertados