Proyecto final inteligencia artificial

CONSUMO DE ALCHOL EN ESTUDIANTES DE SECUNDARIA

Aaron Santiago Pedraza Cardenas
aaronpedraza@javeriana.edu.co

Brandon Emilio Gonzalez
gonzales.brandon@javeriana.edu.co

ING: Francisco Calderón

Pontificia Universidad Javeriana

24/11/20
Bogota,Colombia

Introducción:

La adolescencia representa la transición de la niñez a la edad adulta, es decir, es un periodo de desarrollo donde la persona adquiere las capacidades físicas y psíquicas que la identificarán como desarrollada, especialmente las sexuales, que le permitirán reproducirse. Al ser una etapa de profundos cambios, está marcada por la inestabilidad y, en la mayoría de las ocasiones, el desconcierto y la confusión de los propios jóvenes ante sus cambios.

Uno de estos cambios por los cuales los jóvenes estan supuestos a pasar es el consumó de alcohol, por lo cual a lo largo de este documento vamos analizar cuales son las características de lo jovenes mas consumidores de alcohol y sus calificaciones en tres diferentes periodos o cortes.

Objetivos:

- Objetivo 1: Realizar un análisis exploratorio de los datos y conclusiones que se puedan sacar de ellos.
- Objetivo 2: Realizar una predicción sobre los valores objetivo (G1, G2 y G3).

Dataset:

El dataset que analizaremos está disponible en el siguiente enlance:
https://www.kaggle.com/uciml/student-alcohol-consumption

Atributos del dataset:

Los atributos para los conjuntos de datos de student-mat.cvs y student-por.cvs son:

- school:escuela del estudiante (binario: 'GP' - Gabriel Pereira o 'MS' - Mousinho da Silveira)
- sex: sexo del estudiante (binario: 'F' - mujer o 'M' - hombre)
- age: edad del estudiante (numérico: de 15 a 22)
- address: tipo de dirección del hogar del estudiante (binario: 'U' - urbano o 'R' - rural)
- famsize: tamaño de la familia (binario: 'LE3' - menor o igual a 3 o 'GT3' - mayor que 3)
- Pstatus: estado de cohabitación de los padres (binario: 'T' viviendo juntos o 'A' - separados)
- Medu: educación de la madre (numérico: 0 - ninguno, 1 - educación primaria (4º grado), 2 - 5º a 9º grado, 3 - educación secundaria o 4 - educación superior
- Fedu: educación del padre (numérico: 0 - ninguno, 1 - educación primaria (4 ° grado), 2 - 5 ° a 9 ° grado, 3 - educación secundaria o 4 - educación superior)
- Mjob: trabajo de la madre (nominal: 'maestra', relacionado con la atención de 'salud', 'servicios' civiles (por ejemplo, administrativo o policial), 'en_casa' u 'otro')

- Fjob: trabajo del padre (nominal: 'maestro', relacionado con el cuidado de la salud, 'servicios' civiles (por ejemplo, administrativo o policial), 'en_home' u 'otro')
- reason: motivo para elegir esta escuela (nominal: cerca de 'casa', 'reputación' de la escuela, preferencia de 'curso' u 'otro')
- guardian: tutor del estudiante (nominal: 'madre', 'padre' u 'otro')
- traveltime: tiempo de viaje de la casa a la escuela (numérico: 1 - 1 hora)
- studytime: tiempo de estudio semanal (numérico: 1 - 10 horas)
- failures: número de fallos de clases anteriores (numérico: n si 1 <= n <3, en caso contrario 4)
- schoolsup: apoyo educativo adicional (binario: sí o no)
- famsup: apoyo educativo familiar (binario: sí o no)
- paid: clases pagas adicionales dentro de la asignatura del curso (matemáticas o portugués) (binario: sí o no)
- activities: actividades extracurriculares (binario: sí o no)
- nursery: asistió a la guardería (binario: sí o no)
- higher: quiere cursar estudios superiores (binario: sí o no)
- Internet: acceso a Internet en casa (binario: sí o no)
- romantic: con una relación romántica (binario: sí o no)
- famrel: calidad de las relaciones familiares (numérico: de 1 muy mala a 5 - excelente)
- freetime:tiempo libre después de la escuela (numérico: de 1 - muy bajo a 5 - muy alto)
- goout :salir con amigos (numérico: de 1 - muy bajo a 5 - muy alto)
- Dalc: consumo de alcohol en la jornada laboral (numérico: de 1 - muy bajo a 5 - muy alto)
- Walc: consumo de alcohol durante el fin de semana (numérico: de 1 - muy bajo a 5 - muy alto)
- health: estado de salud actual (numérico: de 1 - muy malo a 5 - muy bueno)
- absences: número de ausencias escolares (numérico: de 0 a 93)

Preprocesamiento de los datos:

Primero vamos a realizar un preprocesado de los datos, para ello usaremos la librería Pandas. Crearemos un DataFrame a partir de los archivos CSV disponibles y, además, los pre procesaremos para que todos los campos contengan valores numéricos.

```python
[195] import pandas as pd
      import numpy as np
      import matplotlib.pyplot as plt

      #Cargar conjuntos de datos
      math_course_file = '/content/student-mat.csv'
      portuguese_course_file = '/content/student-por.csv'
      separator = ';'
      df1 = pd.read_csv(filepath_or_buffer=math_course_file, sep=separator)
      df2 = pd.read_csv(filepath_or_buffer=portuguese_course_file, sep=separator)

      df = pd.concat([df1, df2], axis=0) #pila vertical
```

```python
[196] #explorar los conjuntos de datos
      print(df1.shape)
      print(df2.shape)
      print(df.shape)
      print(df.columns)

      #cantidad de NAN's
      print(df.info())
```

Fig1. Exploración de datos.

En la figura 1 se observa las líneas de código que encargan convertir lo archivos cvs en dataframe mediante la librería de pandas y así poder realizar una exploración del dataset y conocer las características, además de saber cuales son los atributos que son necesarios convertir en valores numéricos para poder realizar su posterior análisis.

También estas líneas de código permiten observar si hay algún valor NAN, para posteriormente iniciar con el preprocesamiento de algunas columnas para que sus datos sean numéricos y, así, poder analizarlos mejor.

```python
[ ]  #obtener un número relacionado con el campo laboral
     def parse_job_to_binary(x):
         return [i[0] for i in enumerate(['teacher', 'health', 'services', 'at_home', 'other']) if i[1]==x][0]
     def parse_job_to_string(x):
         return [i[1] for i in enumerate(['teacher', 'health', 'services', 'at_home', 'other']) if i[0]==x][0]

     #obtener un número relacionado con el motivo del elegir la escuela
     def parse_reason_to_binary(x):
         return [i[0] for i in enumerate(['home', 'reputation', 'course', 'other']) if i[1]==x][0]

     #obtener un número relacionado con el campo tutor del estudiante
     def parse_guardian_to_binary(x):
         return [i[0] for i in enumerate(['mother', 'father', 'other']) if i[1]==x][0]
```

```python
[ ]  #analizar correctamente todos los campos no son numéricos
     df.school = df.school.apply(lambda x: 0 if x=='GP' else 1) #solo tenemos dos escuelas
     df.address = df.address.apply(lambda x: 0 if x=='U' else 1) #solo tenemos dos direcciones
     df.famsize = df.famsize.apply(lambda x: 0 if x=='LE3' else 1) #tenemos dos tamaños de familia
     df.Pstatus = df.Pstatus.apply(lambda x: 0 if x=='A' else 1) #tenemos estados de convivencia de los padres
     df.Mjob = df.Mjob.apply(lambda x: parse_job_to_binary(x))
     df.Fjob = df.Fjob.apply(lambda x: parse_job_to_binary(x))
     df.reason = df.reason.apply(lambda x: parse_reason_to_binary(x))
     df.guardian = df.guardian.apply(lambda x: parse_guardian_to_binary(x))
     df.sex = df.sex.apply(lambda x: 0 if x=='M' else 1)
     df.schoolsup = df.schoolsup.apply(lambda x: 0 if x=='no' else 1)
     df.famsup = df.famsup.apply(lambda x: 0 if x=='no' else 1)
     df.paid = df.paid.apply(lambda x: 0 if x=='no' else 1)
     df.activities = df.activities.apply(lambda x: 0 if x=='no' else 1)
     df.nursery = df.nursery.apply(lambda x: 0 if x=='no' else 1)
     df.higher = df.higher.apply(lambda x: 0 if x=='no' else 1)
     df.internet = df.internet.apply(lambda x: 0 if x=='no' else 1)
     df.romantic = df.romantic.apply(lambda x: 0 if x=='no' else 1)
     df['G1Pass'] = df.G1.apply(lambda x: 0 if x<10 else 1) #crear una nueva columna para el éxito de la clasificación G1
     df['G2Pass'] = df.G2.apply(lambda x: 0 if x<10 else 1) #crear una nueva columna para el éxito de la clasificación G2
     df['G3Pass'] = df.G3.apply(lambda x: 0 if x<10 else 1) #crear una nueva columna para el éxito final de la clasificación
     df.absences = df.absences.apply(lambda x: 0 if x<10 else 1) #ponemos 1 cuando un estudiante frecuentemente hace ausencias
     df.studytime = df.studytime.apply(lambda x: 0 if x<3 else 1) #ponemos 1 cuando un estudiante estudia mucho con frecuencia
     df.freetime = df.freetime.apply(lambda x: 0 if x<3 else 1) #Ponemos 1 cuando un estudiante frecuentemente tiene mucho tiempo l
```

Fig2. Preprocesamiento de los datos.

La figura 2 permite observar los atributos que tenia mas de dos características por lo cual se decide que tomen un valor numérico, como por ejemplo se enumeraron los campos laborares tanto para la madre y el padre de los estudiantes, además de la razón por la cual escogieron la escuela, asimismo se enumeran quien es el acudiente del estudiante.

También los otros atributos que solamente tenia dos características tomar un valor binario, como se observa el segundo código en la figura 2.

Objetivo 1:

Una vez se realiza el preprocesamiento de los datos, se procede a realizar un análisis de los datos y presentación, pero teniendo en cuenta la siguiente ecuación:

$$ALC = \frac{W_{alc} * 2 + D_{alc} * 5}{7}$$

Esta ecuación es tomada del paper [1], donde W_{alc}es el consumo de alcohol los fines de semana y D_{alc} el consumo de alcohol entre semana. El valor de ALC va a permitir clasificar si un estudiante es bebedor obteniendo un $ALC \geq 3$ y no bebedor como $ALC < 3$.

```
#columna de clasificación
df['Alc'] = (df.Walc*2 + df.Dalc*5)/7
df.Alc = df.Alc.apply(lambda x: 0 if x < 3 else 1)
print(df.Alc.describe())
```

```
count    1044.000000
mean        0.113027
std         0.316777
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         1.000000
Name: Alc, dtype: float64
```

Fig3. Clasificación de bebedores y no bebedores.

En la figura 3 se observa la líneas de código que se encargan de clasificar a los estudiantes de como bebedores y no bebedores de acuerdo a su consumo de alcohol de fin semana(W_{alc}) y entre semana(D_{alc}), mediante la condición ya establecida anteriormente.

```
#Porcentaje de bebedores por sexo
print(df.groupby('sex').mean().loc[:,'Alc'])

#Porcentaje de bebedores por edad
print(df.groupby('age').mean().loc[:,'Alc'])

#Porcentaje de bebedores por marca FinalSuccess (G3)
print(df.groupby('G3Pass').mean().loc[:,'Alc'])

#Porcentaje de bebedores por sexo y G3Pass
print(df.groupby(['sex', 'G3Pass']).mean().loc[:,'Alc'])

#Porcentaje de bebedores por edad y G3Pass
print(df.groupby(['age', 'G3Pass']).mean().loc[:,'Alc'])

#Porcentaje de bebedores por campo de tiempo de estudio
print(df.groupby('studytime').mean().loc[:,'Alc'])

#Porcentaje de bebedores por campo de tiempo libre
print(df.groupby('freetime').mean().loc[:,'Alc'])

#Porcentaje de bebedoras por goout
print(df.groupby('goout').mean().loc[:,'Alc'])

#Porcentaje de bebedores por ausencias
print(df.groupby('absences').mean().loc[:,'Alc'])
```

Fig4. Análisis de los estudiantes.

Una vez el dataset esta con datos numéricos y los estudiante están clasificados como bebedores y no bebedores, se realiza un pequeño análisis en el cual se revisan porcentajes de bebedores por sexo, edad, notas del ultimo corte, tiempo de estudio, tiempo libre, salidas con amigos y ausencias de clases, este análisis se realiza con las líneas de código que se presentan en al figura 4.

```
sex
0    0.200883
1    0.045685
Name: Alc, dtype: float64
age
15    0.067010
16    0.085409
17    0.122744
18    0.130631
19    0.232143
20    0.111111
21    0.666667
22    1.000000
Name: Alc, dtype: float64
G3Pass
0    0.143478
1    0.104423
Name: Alc, dtype: float64
sex  G3Pass
0    0         0.257143
     1         0.183908
1    0         0.048000
     1         0.045064
Name: Alc, dtype: float64
age  G3Pass
15   0         0.060606
     1         0.068323
16   0         0.122449
     1         0.077586
17   0         0.150000
     1         0.115207
18   0         0.114754
     1         0.136646
19   0         0.217391
     1         0.242424
20   0         1.000000
     1         0.000000
21   0         1.000000
     1         0.500000
22   0         1.000000
Name: Alc, dtype: float64
studytime
0    0.130488
1    0.049107
Name: Alc, dtype: float64
freetime
0    0.089362
1    0.119901
Name: Alc, dtype: float64
goout
1    0.042254
2    0.060484
3    0.074627
4    0.149780
5    0.251534
Name: Alc, dtype: float64
absences
0    0.095398
1    0.215686
Name: Alc, dtype: float64
```

Fig5. Resultados del análisis.

Después de analizar los datos, podemos observar en la figura 5 que hay algunos factores que son influyentes en el consumo de alcohol de los estudiantes:

- Género (Hombres superan a las mujeres con diferencia)
- Edad (Por ejemplo, todos los estudiantes de 22 años son bebedores)
- Las ausencias
- El tiempo de estudio
- El tiempo libre
- Las salidas
- El resultado de las notas (aprobado / suspenso)

```
#porcentaje de aprobados según si son consumidores del alcohol
print(df.groupby(['Alc']).mean().loc[:,'G1Pass'])

#porcentajes de aprobados segun sexo y si consumen alcohol
print(df.groupby(['sex', 'Alc']).mean().loc[:,'G1Pass'])
print(df.groupby(['sex', 'Alc']).mean().loc[:,'G2Pass'])
print(df.groupby(['sex', 'Alc']).mean().loc[:,'G3Pass'])
```

```
Alc
0    0.725702
1    0.618644
Name: G1Pass, dtype: float64
sex  Alc
0    0      0.723757
     1      0.604396
1    0      0.726950
     1      0.666667
Name: G1Pass, dtype: float64
sex  Alc
0    0      0.729282
     1      0.582418
1    0      0.741135
     1      0.666667
Name: G2Pass, dtype: float64
sex  Alc
0    0      0.784530
     1      0.703297
1    0      0.789007
     1      0.777778
Name: G3Pass, dtype: float64
```

Fig6. Análisis. De influencia en el alcohol.

También se realiza un análisis de como el alcohol influyen en las notas de calificaciones para los tres diferentes cortes, este análisis se realiza como se presenta en la figura 6 y su resultado muestra que el consumo de alcohol, reduce la tasa de aprobados en los cortes, además existe un porcentaje mayor de hombres que no consiguen aprobar que son bebedores.

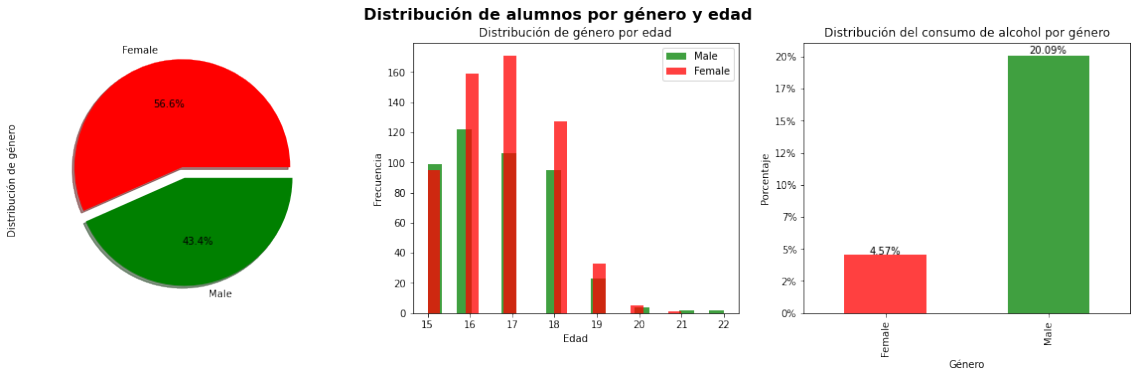

Fig7 Resultados del análisis.

La figura 7 muestra las graficas de un análisis de la distribución de genero y edad del dataset, además una grafica en la que muestra que el consumo de alcohol esta mas presente en hombres con un 20.09% que en mujeres con un 4.57%. También el 56.6% de los estudiantes son mujeres, y la mayoría de estudiantes están entre los 16,17 y 18 años.

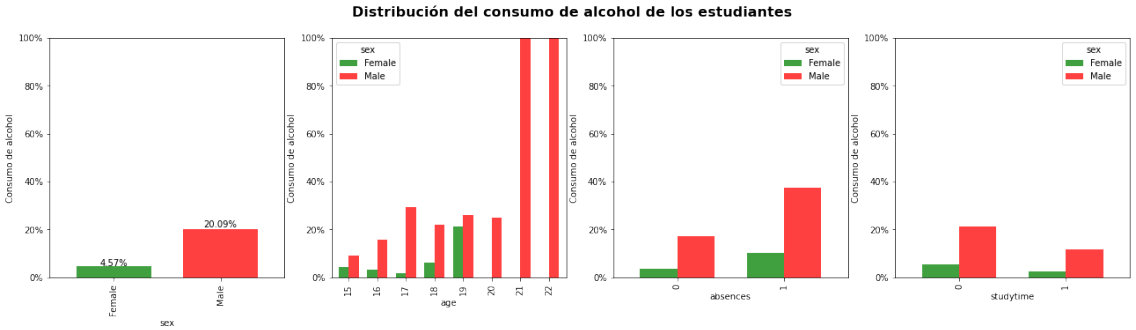

Fig8 Distribución de consumo de alcohol.

En la figura 8, se observa lo siguiente:

- Los chicos (20,09%) consumen mucho más alcohol que las chicas (4,57%)
- El consumo de alcohol se dispara en los chicos mayores de 16 años
- Las ausencias a clase influyen en el aumento del consumo de alcohol
- El consumo de alcohol también se ve influenciado si no se dedica tiempo a estudiar
- Los estudiantes mayores de 21 hombres son bebedores.

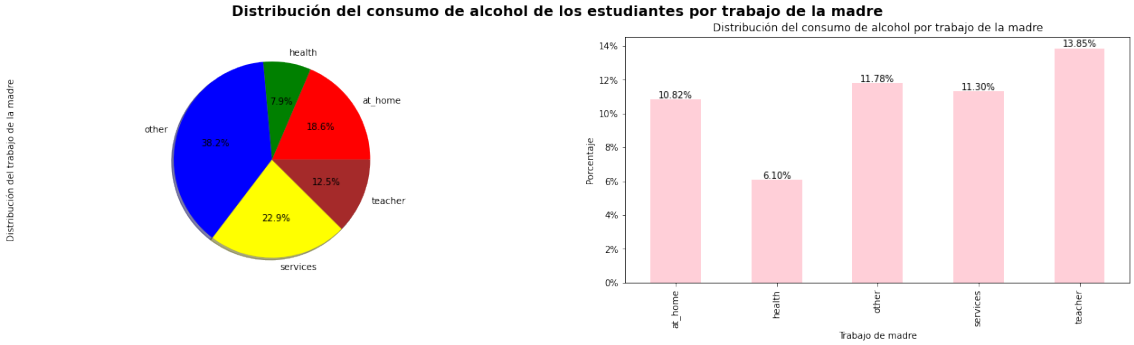

Fig9 Distribución de consumo de alcohol por trabajo de la madre.

La figura 9 indica que las madres que tiene un trabajo como profesora son tienen el índice mas alto en tener hijos que consumen alcohol y las madres en trabajar en servicios de salud tiene el menor porcentaje de hijos que consuman alcohol.

Objetivo 2:

Es importante mencionar que se crearon 3 nuevas columnas para poder predecir si un estudiante ha aprobado el G1, G2 y G3, estos campos son: **G1Pass, G2Pass y G3Pass.** También para la predicción de las notas, se opta por utilizar diferentes clasificadores como KNN, arboles de decisión y Naive bayes. Donde arboles decisión se toma como referencia del paper 1, donde los árboles de decisión están compuestos por varios SI-ENTONCES en cascada. Cuando el algoritmo crea un árbol de decisión, necesita decidir qué atributos están involucrados en la fase de división. Existe un índice llamado criterio de división para este propósito. KNIME propone ganancia de información e impureza de Gini. La impureza de Gini es una medida de la frecuencia con la que un elemento elegido al azar del conjunto se etiquetaría incorrectamente si se etiquetara al azar de acuerdo con la distribución de etiquetas en el subconjunto y la ganancia de información es una medida basada en la entropía.[1]

Y los clasificadores KNN, y Naive bayes se toman como referencia de las clases de inteligencia artificial, en la cuales hubo ejemplos de como implementar estos clasificadores de acuerdo con la predicción que se desee realiza.

```
from sklearn.model_selection import StratifiedShuffleSplit
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
import random

#dividir el conjunto de datos en un 60% para entrenamiento y 40% para pruebas
def predict(train, labels, grades, algorithm = 'knn'):
    stratified_data = StratifiedShuffleSplit(n_splits=3, test_size=0.4, random_state=random.randint(0, 999))
    for i, j in stratified_data.split(train, labels):
        X_train = train.iloc[i]
        y_train = labels.iloc[i]
        X_test = train.iloc[j]
        y_test = labels.iloc[j]

    if algorithm == 'knn':
        model = KNeighborsClassifier()
    elif algorithm == 'dt':
        model = DecisionTreeClassifier()
    elif algorithm == 'LR':
        model = LogisticRegression()
    elif algorithm == "NB":
        model = RandomForestClassifier()


    model.fit(train, labels)
    pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, pred)
    print("Results for " + grades + " (algorithm = " + algorithm + "):")
    print(classification_report(y_test, pred, target_names=["No-pass", "Pass"]))
    print("Acuracy: %0.2f%%" % (accuracy*100) + "\n")
    print("Confusion matrix " + grades + ":")
    print(confusion_matrix(y_test, pred))
    print("\n")


#delete some columns
train = df.drop(['Walc','Dalc','G1','G2','G3','G1Pass','G2Pass','G3Pass'], axis = 1)

"""imprimir los resultados finales"""

#con kNN
predict(train, df['G1Pass'], "G1", algorithm='knn')
predict(train, df['G2Pass'], "G2", algorithm='knn')
predict(train, df['G3Pass'], "G3", algorithm='knn')

#con Decision Tree
predict(train, df['G1Pass'], "G1", algorithm='dt')
predict(train, df['G2Pass'], "G2", algorithm='dt')
predict(train, df['G3Pass'], "G3", algorithm='dt')

#con Naive_bayes
predict(train, df['G1Pass'], "G1", algorithm='NB')
predict(train, df['G2Pass'], "G2", algorithm='NB')
predict(train, df['G3Pass'], "G3", algorithm='NB')
```

Fig10 Predicción de las notas de los cortes.

La figura 10, presenta las líneas de código de que permiten realizar las predicciones de las notas de los estudiantes, en el cual primero se parte en dividir el data set en 60% para el entrenamiento y un 40% para el conjunto de pruebas o test, también se decide por imprimir la matriz de confusión de cada algoritmo de predicción.

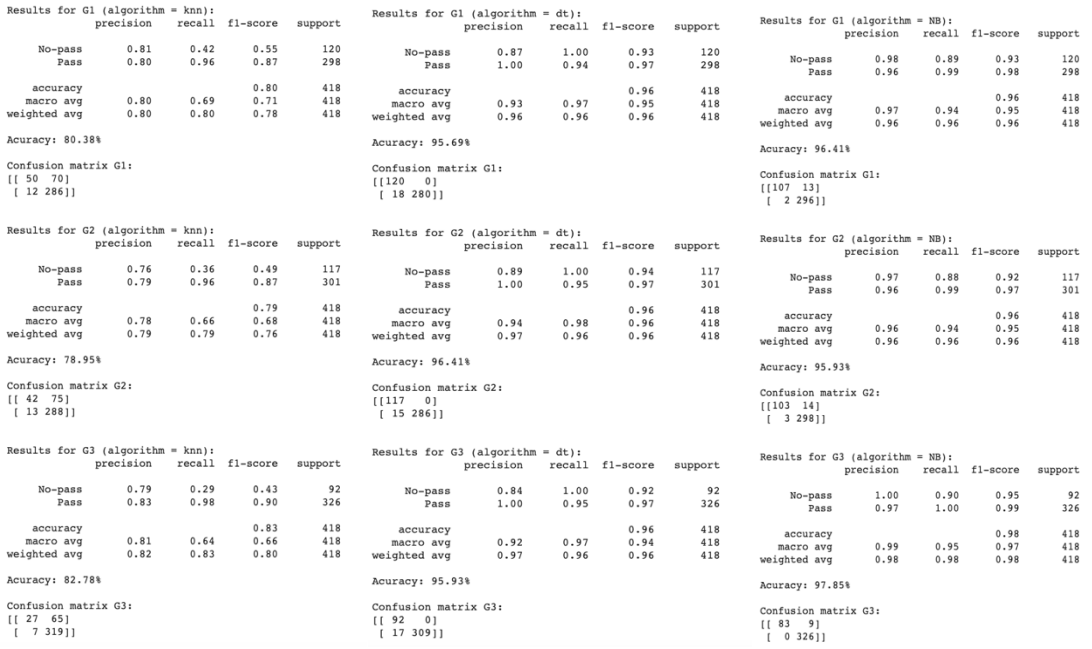

Fig10 Resultados de las predicciones.

.

Sea han habilitado la predicción con diferentes algoritmos y estos son los resultados obtenidos:

- Decision Tree: G1 accuracy: 96.89%, G2 accuracy: 96.65%, G3 accuracy: 97.37%.
- kNN: G1 accuracy: 80.86%, G2 accuracy: 82.06%, G3 accuracy: 81.58%.
- GaussianNB: G1 accuracy: 96.41%, G2 accuracy: 95.93%, G3 accuracy: 97.85%.

Los mejores resultados se consiguen con Decision Tree. Podemos fijarnos también en la matriz de confusión donde observamos que la sensibilidad y la sensibilidad están muy parejos. Además, en general, el número de falsos positivos es muy reducido (lo cual nos interesa, preferiremos predecir con mayor precisión los «aprobados» en comparación con los «suspensos»).

Conclusiones:

- El análisis de datos arrojó que los jóvenes hombres que consumen alcohol tienen un porcentaje de aprobación menor que los hombres que no beben.

- El consumo de alcohol en hombres es mayor que el consumo en mujeres, lo que genera una mayor pérdida de las asignaturas en el género masculino. Por otra parte, en el caso de los hombres la tendencia es que al aumentar la edad el mayor es la probabilidad de que consuma alcohol

- Los mejores resultados se consiguen usando el algoritmo de Decision tree presentando una exactitud promedio de 96.97% frente a un 96.73% del GausisanNB y el 81.5% del KNN.

- La matriz de confusión fue la herramienta que permitió evaluar de manera más eficiente los algoritmos para encontrar el mejor métodos

- Una posible mejora orientada al trabajo futuro sería tratar de ajustar (tuning) los clasificadores para mejorar la accuracy.

Referencias:

1. University of Minho, Cortez, P. C., & Silva, A. (2018, abril). *USING DATA MINING TO PREDICT SECONDARY SCHOOL STUDENT PERFORMANCE*. University of Minho.