4 .Con el uso de librerías realiza en Python los mismos preprocesamiento del punto 3.

onehotencoder, labelencoder, discretización y normalización.

codigo(onehotencoder)

```
import pandas as pd
from sklearn.preprocessing import OneHotEncoder

# Leer el archivo CSV completo
data = pd.read_csv('/content/Drive/MyDrive/datos/examencovid354.csv')

# Instanciar el OneHotEncoder
preprocesamiento = OneHotEncoder()

# Seleccionar todas las columnas
# Aplicar OneHotEncoder a todas las columnas (si tienes categóricas y numéricas)
# Para este ejemplo, seleccionamos todas las columnas del archivo CSV
listaDatos = data.values.tolist()

# Ajustar el OneHotEncoder a los datos del CSV
preprocesamiento.fit(listaDatos)

# Ver las categorías detectadas por OneHotEncoder
print("Categorías detectadas por OneHotEncoder:")
print(preprocesamiento.categories_)

# Transformar los datos a OneHotEncoding y convertir el resultado en una matriz
resultado = preprocesamiento.transform(listaDatos).toarray()

# Mostrar los resultados del OneHotEncoder aplicados a todas las filas del CSV
print("Resultado de la transformación OneHotEncoder:")
print(resultado)

# Si quieres ver un resumen o las primeras filas del resultado transformado
print("Primeras  filas transformadas:")
print(resultado[:5])
```

Corrida

```
          school, relatives and friends ],
Resultado de la transformación OneHotEnc
[[1. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
Primeras  filas transformadas:
[[1. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

Codigo(labelencoder)

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Leer el archivo CSV completo
data = pd.read_csv('/content/Drive/MyDrive/datos/examencovid354.csv')

# Verificar si se ha leído el archivo correctamente
print(f"El archivo tiene {data.shape[0]} filas y {data.shape[1]}
columnas.")

# Revisar si hay columnas con valores faltantes
print("\nNúmero de valores faltantes por columna:")
print(data.isnull().sum())

# Inicializar LabelEncoder
label_encoder = LabelEncoder()

# Aplicar LabelEncoder a todas las columnas que sean de tipo 'object'
o 'category'
for columna in data.columns:
    if data[columna].dtype == 'object' or data[columna].dtype ==
```

```
'category':
        # Llenar valores NaN con una cadena vacía o algún valor
adecuado para no tener errores
        data[columna] = data[columna].fillna('desconocido')

        # Aplicar LabelEncoder y crear una nueva columna con el
sufijo '_encoded'
        data[columna + '_encoded'] =
label_encoder.fit_transform(data[columna].astype(str))

        # Imprimir los primeros valores de la codificación realizada
        print(f"\nColumna: {columna}")
        print(data[[columna, columna + '_encoded']].head())
    else:
        print(f"La columna {columna} no es de tipo texto, se omite la
codificación.")

# Mostrar las primeras filas del DataFrame con las nuevas columnas
codificadas
print("\nDataFrame con las columnas codificadas:")




print(data.head(180))

# Mostrar el total de filas
print(f"Total de filas leídas: {data.shape[0]}")
```

corrida

```
Columna: Time spent on TV
  Time spent on TV  Time spent on TV_encoded
0                1                         9
1                0                         1
2                0                         1
3                0                         1
4                1                         9
La columna Number of meals per day no es de tipo texto, se omite la codificación.

Columna: Change in your weight
  Change in your weight  Change in your weight_encoded
0             Increased                             1
1             Decreased                             0
2       Remain Constant                             2
3             Decreased                             0
4       Remain Constant                             2

Columna: Health issue during lockdown
  Health issue during lockdown  Health issue during lockdown_encoded
0                           NO                                    0
1                           NO                                    0
2                           NO                                    0
3                           NO                                    0
4                           NO                                    0
```

Codigo(discretización)

```python
import pandas as pd
from sklearn.preprocessing import KBinsDiscretizer

# Leer el archivo CSV completo
data = pd.read_csv('/content/Drive/MyDrive/datos/examencovid354.csv')

# Seleccionar solo las columnas numéricas del archivo
columnas_numericas = data.select_dtypes(include=['float64', 'int64'])

# Instanciar el KBinsDiscretizer
# n_bins indica el número de intervalos en los que dividir los datos
# encode define el método de codificación de los intervalos
# strategy define la estrategia para los límites de los intervalos
('uniform', 'quantile', 'kmeans')
discretizer = KBinsDiscretizer(n_bins=5, encode='ordinal',
strategy='uniform')

# Ajustar y transformar las columnas numéricas con KBinsDiscretizer
resultado_discretizado =
discretizer.fit_transform(columnas_numericas)
```

```python
# Crear un nuevo DataFrame con los resultados discretizados
df_discretizado = pd.DataFrame(resultado_discretizado,
columns=columnas_numericas.columns)

# Mostrar los resultados de la discretización
print("Datos discretizados:")
print(df_discretizado)

# Si deseas ver solo las primeras 5 filas del resultado discretizado
print("Primeras 5 filas discretizadas:")
print(df_discretizado.head())
```

Corrida

```
Datos discretizados:
      Age of Subject  Time spent on Online Class  Time spent on self study  \
0                1.0                         1.0                       1.0
1                1.0                         0.0                       0.0
2                1.0                         3.0                       0.0
3                1.0                         1.0                       0.0
4                1.0                         1.0                       0.0
...              ...                         ...                       ...
1177             0.0                         1.0                       1.0
1178             0.0                         3.0                       1.0
1179             0.0                         2.0                       0.0
1180             0.0                         2.0                       0.0
1181             0.0                         2.0                       0.0

      Time spent on fitness  Time spent on sleep  Time spent on social media  \
0                       0.0                  1.0                        1.0
1                       2.0                  2.0                        1.0
2                       0.0                  0.0                        1.0
3                       1.0                  0.0                        2.0
4                       1.0                  1.0                        1.0
...                     ...                  ...                        ...
1177                    1.0                  1.0                        0.0
1178                    1.0                  2.0                        0.0
1179                    0.0                  1.0                        1.0
1180                    1.0                  1.0                        0.0
1181                    0.0                  1.0                        0.0
```

Codigo(normalización)

```python
import pandas as pd
from sklearn.preprocessing import Normalizer
```

```
# Leer el archivo CSV completo
data = pd.read_csv('/content/Drive/MyDrive/datos/examencovid354.csv')

# Seleccionar solo las columnas numéricas del archivo
# Puedes ajustar si sabes qué columnas deben ser normalizadas
columnas_numericas = data.select_dtypes(include=['float64', 'int64'])

# Instanciar el Normalizer
preprocesamiento = Normalizer(norm='l2')

# Ajustar y transformar las columnas numéricas con Normalizer
resultado_normalizado =
preprocesamiento.fit_transform(columnas_numericas)

# Crear un nuevo DataFrame con los resultados normalizados
df_normalizado = pd.DataFrame(resultado_normalizado,
columns=columnas_numericas.columns)

# Mostrar los resultados de la normalización
print("Datos normalizados:")
print(df_normalizado)

# Si deseas ver solo las primeras 5 filas del resultado normalizado
print("Primeras 5 filas normalizadas:")
print(df_normalizado.head())
```

Corrida

```
Datos normalizados:
      Age of Subject  Time spent on Online Class  Time spent on self study  \
0           0.907909                    0.086468                  0.172935
1           0.885044                    0.000000                  0.000000
2           0.888231                    0.310881                  0.133235
3           0.909091                    0.136364                  0.090909
4           0.896258                    0.128037                  0.128037
...              ...                         ...                       ...
1177        0.768221                    0.192055                  0.256074
1178        0.751559                    0.322097                  0.214731
1179        0.785001                    0.241539                  0.000000
1180        0.789437                    0.281942                  0.197359
1181        0.810524                    0.311740                  0.124696

      Time spent on fitness  Time spent on sleep  Time spent on social media  \
0                  0.000000             0.302636                    0.129701
1                  0.084290             0.421450                    0.126435
2                  0.000000             0.266469                    0.088823
3                  0.045455             0.272727                    0.227273
4                  0.042679             0.341432                    0.128037
...                     ...                  ...                         ...
1177               0.064018             0.512148                    0.064018
1178               0.053683             0.483145                    0.053683
1179               0.030192             0.483077                    0.181154
1180               0.056388             0.451107                    0.028194
1181               0.031174             0.436436                    0.062348

      Number of meals per day
0                    0.172935
1                    0.126435
2                    0.133235
3                    0.136364
```

Link Google colab

https://colab.research.google.com/drive/1R60up3Db-VQ5wdCEmZv7fXqeVigWugOi?usp=sharing