Asignatura	Datos del alumno	Fecha
Aprendizaje Automático y Minería de Datos	Estudiante: Oscar David Bocanegra	3/03/2025
	Estudiante: Laura Marcela Barona	

Trabajo: Lectura de datos y análisis descriptivo

Objetivos

El objetivo principal de esta actividad es que el alumno sea capaz de realizar una lectura y un análisis descriptivo de los datos proporcionados de forma que el lector entienda que contiene un conjunto de datos.

Descripción de la actividad y pautas de elaboración

La variable respuesta es «cnt» y el enlace que contiene los datos es el siguiente:

Accede al enlace a través del aula virtual o desde la siguiente dirección:
https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset

Pasos a seguir (orientativo):

- Análisis descriptivo de los datos.
- Determinar el conjunto de modelización y el de validación.
- Tratamiento de *missing* (si los hay).
- Correlaciones.
- Distribuciones.
- Gráficos que consideres adecuados.

Criterios de evaluación

- La evaluación y la entrega de actividades se realizará de forma individual.

Asignatura	Datos del alumno	Fecha
Aprendizaje Automático y Minería de Datos	Estudiante: Oscar David Bocanegra	3/03/2025
	Estudiante: Laura Marcela Barona	

- Se puede utilizar Python.
- Se valorarán especialmente los comentarios sobre la relación de las variables predictoras con la variable respuesta.
- Se deberán comentar los resultados obtenidos y el código.

Extensión máxima: 3 páginas, fuente Calibri 12 e interlineado 1,5.

Análisis Descriptivo de los datos

El análisis descriptivo de un DataFrame es una técnica fundamental en el análisis de datos que nos permite entender mejor la distribución y las características de los para esta actividad vamos a realizar el análisis descriptivo de los DataFrames `df_day` y `df2_hour` en donde el objetivo de este análisis es comprender la distribución de los datos del sistema de alquiler de bicicletas y su relación con diversas variables predictoras

Por ende, para facilitar este trabajo vamos a utilizar 2 librerías muy conocidas en el mundo del análisis de datos las cuales son pandas y numpy

DataFrame `df_day`

El DataFrame `df_day` contiene datos diarios sobre el alquiler de bicicletas en donde identificamos algunas de las columnas más relevantes incluyen:

- `instant`: Identificador único de cada registro.
- `dteday`: Fecha del registro.
- `season`: Estación del año (1: Primavera, 2: Verano, 3: Otoño, 4: Invierno).
- `yr`: Año (0: 2011, 1: 2012).
- `mnth`: Mes del año.
- `holiday`: Indica si el día es festivo (0: No, 1: Sí).
- `weekday`: Día de la semana.
- `workingday`: Indica si el día es laboral (0: No, 1: Sí).

- `weathersit`: Condición meteorológica (1: Despejado, 2: Nublado, 3: Lluvia ligera, 4: Lluvia intensa).
- `temp`: Temperatura normalizada.
- `atemp`: Sensación térmica normalizada.
- `hum`: Humedad normalizada.
- `windspeed`: Velocidad del viento normalizada.
- `casual`: Número de usuarios ocasionales.
- `registered`: Número de usuarios registrados.
- `cnt`: Número total de alquileres.

El análisis descriptivo de este DataFrame se puede realizar utilizando el método `describe()`, que proporciona estadísticas resumidas como la media, la desviación estándar, los valores mínimos y máximos, y los percentiles.

```
# Análisis descriptivo del dataframe df
print("Análisis descriptivo del dataframe df:")
print(df_day.describe())
```

```
Análisis descriptivo del dataframe df:
          instant      season          yr        mnth     holiday     weekday  \
count  731.000000  731.000000  731.000000  731.000000  731.000000  731.000000
mean   366.000000    2.496580    0.500684    6.519836    0.028728    2.997264
std    211.165812    1.110807    0.500342    3.451913    0.167155    2.004787
min      1.000000    1.000000    0.000000    1.000000    0.000000    0.000000
25%    183.500000    2.000000    0.000000    4.000000    0.000000    1.000000
50%    366.000000    3.000000    1.000000    7.000000    0.000000    3.000000
75%    548.500000    3.000000    1.000000   10.000000    0.000000    5.000000
max    731.000000    4.000000    1.000000   12.000000    1.000000    6.000000

       workingday  weathersit        temp       atemp         hum   windspeed  \
count  731.000000  731.000000  731.000000  731.000000  731.000000  731.000000
mean     0.683995    1.395349    0.495385    0.474354    0.627894    0.190486
std      0.465233    0.544894    0.183051    0.162961    0.142429    0.077498
min      0.000000    1.000000    0.059130    0.079070    0.000000    0.022392
25%      0.000000    1.000000    0.337083    0.337842    0.520000    0.134950
50%      1.000000    1.000000    0.498333    0.486733    0.626667    0.180975
75%      1.000000    2.000000    0.655417    0.608602    0.730209    0.233214
max      1.000000    3.000000    0.861667    0.840896    0.972500    0.507463

            casual   registered          cnt
count   731.000000   731.000000   731.000000
mean    848.176471  3656.172367  4504.348837
std     686.622488  1560.256377  1937.211452
...
25%     315.500000  2497.000000  3152.000000
50%     713.000000  3662.000000  4548.000000
75%    1096.000000  4776.500000  5956.000000
max    3410.000000  6946.000000  8714.000000
```

DataFrame `df2_hour`

El DataFrame `df2_hour` contiene datos horarios sobre el alquiler de bicicletas. Las columnas son similares a las del DataFrame `df_day`, pero incluyen una columna adicional la cual es `hr`: Hora del día (0-23).

Al igual que con `df_day`, el análisis descriptivo de `df2_hour` se puede realizar utilizando el método `describe()`.

```
# Análisis descriptivo del dataframe df2
print("\nAnálisis descriptivo del dataframe df2:")
print(df2_hour.describe())
```

```
Análisis descriptivo del dataframe df2:
            instant        season            yr          mnth            hr  \
count  17379.0000  17379.000000  17379.000000  17379.000000  17379.000000
mean    8690.0000      2.501640      0.502561      6.537775     11.546752
std     5017.0295      1.106918      0.500008      3.438776      6.914405
min        1.0000      1.000000      0.000000      1.000000      0.000000
25%     4345.5000      2.000000      0.000000      4.000000      6.000000
50%     8690.0000      3.000000      1.000000      7.000000     12.000000
75%    13034.5000      3.000000      1.000000     10.000000     18.000000
max    17379.0000      4.000000      1.000000     12.000000     23.000000

             holiday       weekday    workingday    weathersit          temp  \
count  17379.000000  17379.000000  17379.000000  17379.000000  17379.000000
mean       0.028770      3.003683      0.682721      1.425283      0.496987
std        0.167165      2.005771      0.465431      0.639357      0.192556
min        0.000000      0.000000      0.000000      1.000000      0.020000
25%        0.000000      1.000000      0.000000      1.000000      0.340000
50%        0.000000      3.000000      1.000000      1.000000      0.500000
75%        0.000000      5.000000      1.000000      2.000000      0.660000
max        1.000000      6.000000      1.000000      4.000000      1.000000

              atemp           hum     windspeed        casual    registered  \
count  17379.000000  17379.000000  17379.000000  17379.000000  17379.000000
mean       0.475775      0.627229      0.190098     35.676218    153.786869
...
25%       40.000000
50%      142.000000
75%      281.000000
max      977.000000
```

Resultados del Análisis Descriptivo

A continuación, se presentan algunos puntos clave del análisis descriptivo de ambos DataFrames:

- ****Distribución de Alquileres****: La variable `cnt` la cual es la variable solicitada para este trabajo muestra la distribución del número total de alquileres. En `df_day`, la

media y la desviación estándar de `cnt` nos indican la cantidad promedio de alquileres diarios y la variabilidad de estos. En `df2_hour`, la distribución horaria de `cnt` nos permite entender los patrones de uso a lo largo del día.

Otros datos relevantes que encontramos son las variables meteorológicas las cuales son `temp`, `atemp`, `hum` y `windspeed` proporcionan información sobre las condiciones meteorológicas. Estas variables pueden influir significativamente en el número de alquileres.

Y por ultimo encontramos los usuarios ocasionales vs. registrados en donde las variables `casual` y `registered` nos permiten diferenciar entre usuarios ocasionales y registrados y esto nos da la facilidad de entender el comportamiento de diferentes tipos de usuarios.

Determinar el conjunto de modelización y el de validación.

En este apartado básicamente lo que estamos haciendo es dividir el dataframe en 2 el cual va ser como lo dice el enunciado un apartado de modelización y otro de validación el cual lo establecimos en un 80% modelización y un 20% validación y esto lo hacemos usando la librería de sklearn como lo veremos en la siguiente imagen

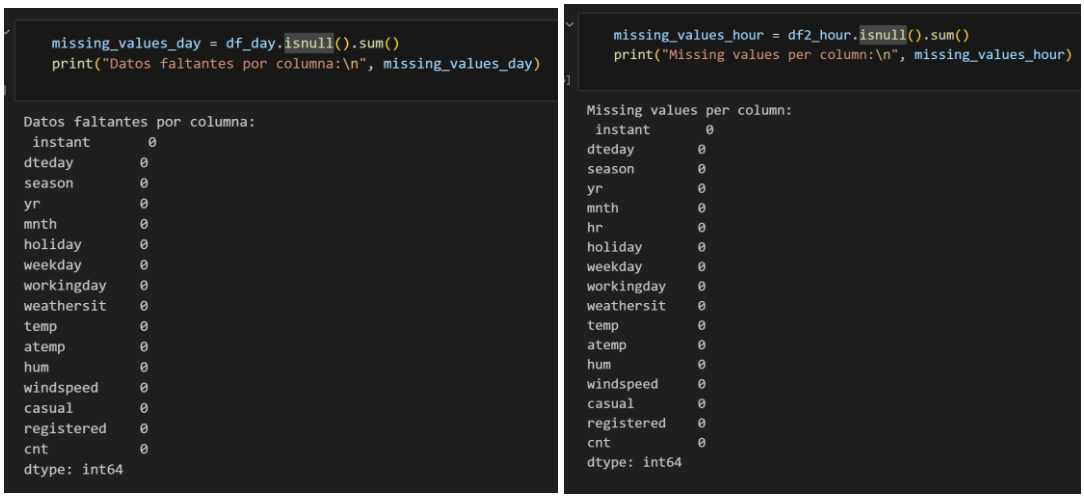
```
from sklearn.model_selection import train_test_split

# Dividir el dataframe df_day en conjuntos de entrenamiento y validación
train_df_day, val_df_day = train_test_split(df_day, test_size=0.2, random_state=42)

# Dividir el dataframe df2_hour en conjuntos de entrenamiento y validación
train_df2_hour, val_df2_hour = train_test_split(df2_hour, test_size=0.2, random_state=42)

# Mostrar el tamaño de los conjuntos
print(f"Tamaño del conjunto de entrenamiento (df_day): {train_df_day.shape}")
print(f"Tamaño del conjunto de validación (df_day): {val_df_day.shape}")
print(f"Tamaño del conjunto de entrenamiento (df2_hour): {train_df2_hour.shape}")
print(f"Tamaño del conjunto de validación (df2_hour): {val_df2_hour.shape}")
```

```
Tamaño del conjunto de entrenamiento (df_day): (584, 16)
Tamaño del conjunto de validación (df_day): (147, 16)
Tamaño del conjunto de entrenamiento (df2_hour): (13903, 17)
Tamaño del conjunto de validación (df2_hour): (3476, 17)
```

Tratamiento de *missing* (si los hay).

Para este caso se confirmó que ambos datasets contienen valores completos, esto quiere decir que están sin datos faltantes esto lo hacemos usando el siguiente

comando el cual es algo que nos ofrece la librería de pandas para comprobar que datos faltantes tenemos o no como lo es en este caso.



```
missing_values_day = df_day.isnull().sum()
print("Datos faltantes por columna:\n", missing_values_day)

Datos faltantes por columna:
 instant       0
dteday         0
season         0
yr             0
mnth           0
holiday        0
weekday        0
workingday     0
weathersit     0
temp           0
atemp          0
hum            0
windspeed      0
casual         0
registered     0
cnt            0
dtype: int64
```

```
missing_values_hour = df2_hour.isnull().sum()
print("Missing values per column:\n", missing_values_hour)

Missing values per column:
 instant       0
dteday         0
season         0
yr             0
mnth           0
hr             0
holiday        0
weekday        0
workingday     0
weathersit     0
temp           0
atemp          0
hum            0
windspeed      0
casual         0
registered     0
cnt            0
dtype: int64
```

Correlaciones

Para este caso se calcularon las correlaciones entre las variables y la variable respuesta cnt (cantidad de bicicletas alquiladas) en donde se destacan lo resultados de que la mayor correlación positiva se encuentra con registered (usuarios registrados), con valores de 0.945 en datos diarios y 0.972 en datos horarios y que la temperatura (temp y atemp) presenta una correlación moderada con cnt, alrededor de 0.63 en datos diarios y 0.40 en datos horarios y en donde la humedad (hum) y la velocidad del viento (windspeed) tienen correlaciones negativas con cnt, indicando que condiciones climáticas adversas afectan el uso del servicio.

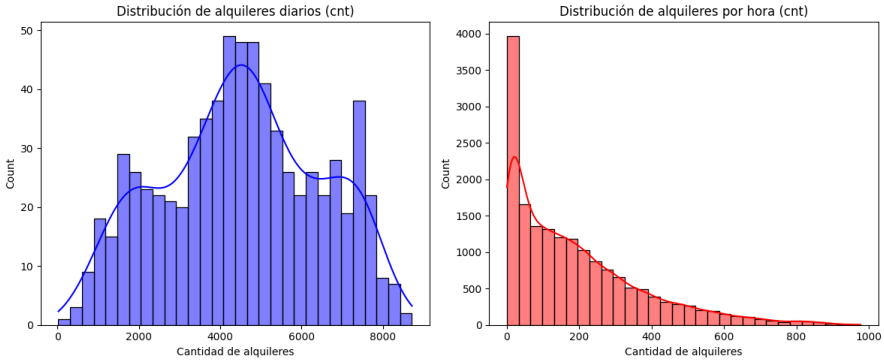
Distribuciones.

Para este caso se analizaron las distribuciones de la variable cnt en ambos conjuntos de datos:

En los datos diarios, la distribución muestra una tendencia a valores altos, lo que sugiere una alta demanda en días específicos.

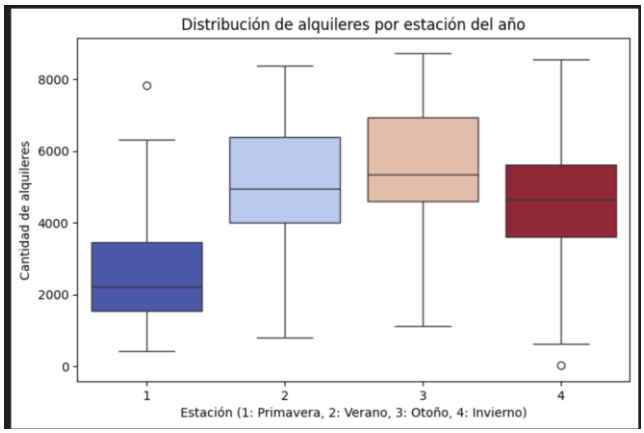
En los datos horarios, la distribución es más dispersa, con muchos valores bajos, reflejando la variabilidad de la demanda a lo largo del día. Y además de esto lo podemos observar de la siguiente manera

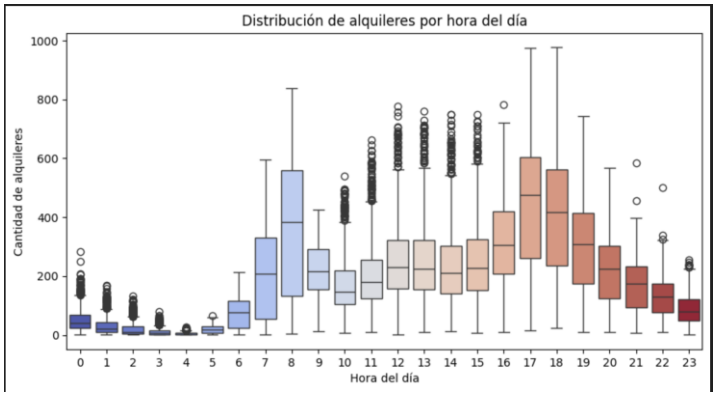
Asignatura	Datos del alumno	Fecha
Aprendizaje Automático y Minería de Datos	Estudiante: Oscar David Bocanegra Estudiante: Laura Marcela Barona	3/03/2025

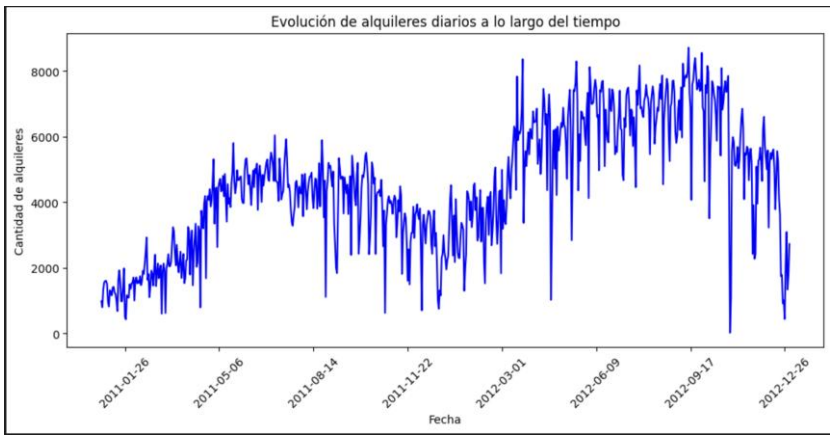


Gráficos que consideres adecuados.

Para este caso usamos 3 gráficos el cual el primero hace referencia al alquiler por estación: Se observa un mayor número de alquileres en verano y otoño, mientras que en invierno la demanda es menor, otro para alquileres por hora del día: Se identifican dos picos principales: 8-9 AM y 5-7 PM, lo que sugiere un uso ligado a horarios laborales y escolares y por último utilizamos uno para la evolución de alquileres diarios a lo largo del tiempo.







Análisis de Resultados

A partir de los gráficos y cálculos realizados, se pueden extraer algunas conclusiones como las siguientes conclusiones:

Gran porcentaje de alquiler de los usuarios registrados: La gran mayoría de los alquileres provienen de usuarios registrados, lo que indica una dependencia del servicio en usuarios recurrentes de parte de ellos.

Efecto del clima: Condiciones favorables (temperatura adecuada, baja humedad y poco viento) fomentan el uso de las bicicletas de lo contrario este uso disminuye.

Comportamiento estacional y diario: Hay un patrón cíclico en los alquileres, con mayor uso en meses cálidos y horarios de traslado.

Oportunidades para optimización: Con base en estos datos, podrían implementarse estrategias como incentivos en invierno o ajustes en la disponibilidad de bicicletas según la demanda horaria.