UIDE

Powered by
Arizona State University

DISEÑO DE PROCESOS ETL EN DATA SCIENCE

PRÁCTICA 1

OBJETIVO GENERAL

Este documento describe el proceso ETL (Extract, Transform, Load) aplicado al archivo 'notas_master_data_science.csv'. El objetivo del proceso es extraer los datos de los estudiantes, calcular el promedio de notas, clasificar a los estudiantes en aprobados y reprobados, y presentar los resultados mediante diagramas de barras y gráficos tipo rosco (pie chart).

FASE 1 - EXTRACCIÓN

En esta fase se realiza la lectura del archivo CSV 'notas_master_data_science.csv', el cual contiene 50 registros de estudiantes y 5 columnas correspondientes a las materias del máster en Data Science: Machine Learning, Big Data Analytics, Deep Learning, Data Visualization y Statistics & Probability.

```python
def DownloadFile(uri: str, filename: str, overwrite: bool = False, timeout: int = 20):
    dest = Path(filename).resolve()
    if dest.exists() and dest.is_file() and dest.stat().st_size > 0 and not overwrite:
        print(
            f'✅ Ya existe: "{dest}". No se descarga (use overwrite=True para forzar).'
        )
        return
    if dest.parent and not dest.parent.exists():
        dest.parent.mkdir(parents=True, exist_ok=True)
    print(f'ℹ️ Descargando "{uri}" → "{dest}"')
    try:
        with requests.get(uri, stream=True, timeout=timeout) as resp:
            resp.raise_for_status()
            tmp = dest.with_suffix(dest.suffix + ".part")
            with open(tmp, "wb") as f:
                for chunk in resp.iter_content(chunk_size=1024 * 64):
                    if chunk:  # filtra keep-alive chunks
                        f.write(chunk)
            tmp.replace(dest)
        print(f'✅ Archivo "{dest}" descargado exitosamente.')
    except requests.exceptions.RequestException as e:
        print(f"❌ Error al descargar: {e}")
```

DESCARGA DEL DATASET

En un servidor se encuentra subido el dataset que consiste en el archivo CSV. La función **DownloadFile** permite descargar el archivo de la URL y lo almacena en un directorio específico.

```python
DOWNLOAD_DIR = "Temp"

DATA_FILE_URI = "https://github.com/UIDE-Tareas/5-Diseno-Procesos-ETL-Data-Science-Tarea1/raw/refs/heads/main/Data/NotasMasterDataScience.csv"
DATA_FILENAME = f"{DOWNLOAD_DIR}/NotasMasterNotasMasterDataScience.csv"

ShowTitleBox(
    "DESCARGANDO BASE DE DATOS",
    boxLineStyle=TitleBoxLineStyle.BLOCK,
    color=ConsoleColor.CYAN,
)
DownloadFile(DATA_FILE_URI, DATA_FILENAME, False)
```

PRIMERA VISTA A LOS DATOS

```
ANÁLISIS INICIAL DE DATOS
ℹ INFO Notas Master Data Science ℹ
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 6 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Nombre                  50 non-null     object
 1   Machine Learning        50 non-null     int64
 2   Big Data Analytics      50 non-null     int64
 3   Deep Learning           50 non-null     int64
 4   Data Visualization      50 non-null     int64
 5   Statistics & Probability  50 non-null     int64
dtypes: int64(5), object(1)
memory usage: 2.5+ KB
```

```
ShowTitleBox(
    "ANÁLISIS INICIAL DE DATOS",
    boxLineStyle=TitleBoxLineStyle.BLOCK,
    color=ConsoleColor.CYAN,
)

data = pd.read_csv(DATA_FILENAME)
ShowDfInfo(data, "Notas Master Data Science")
ShowDfHead(data, "Notas Master Data Science", 100)
ShowDfShape(data, "Notas Master Data Science")
```

ℹ Notas Master Data Science: Primeros 10 elementos.

Nombre	Machine Learning	Big Data Analytics	Deep Learning	Data Visualization	Statistics & Probability
Estudiante_1	84	65	30	74	87
Estudiante_2	34	52	56	50	69
Estudiante_3	95	74	40	100	35
Estudiante_4	53	96	54	60	46
Estudiante_5	86	30	65	59	66
Estudiante_6	83	57	51	46	92
Estudiante_7	68	94	10	67	17
Estudiante_8	9	16	8	68	90
Estudiante_9	98	53	66	41	70
Estudiante_10	0	70	77	21	95

ℹ Notas Master Data Science - Tamaño de los datos
50 filas x 6 columnas

Mostramos la información de las columnas.

Mostramos las primera filas.

Mostramos el tamaño del dataset.

FASE 2 - TRANSFORMACIÓN

Durante esta etapa se calculará el promedio de cada estudiante considerando las cinco materias. Posteriormente, se clasificará a los estudiantes como 'Aprobados' si su media es igual o superior a 60, y 'Reprobados' en caso contrario.

PROMEDIO DE ESTUDIANTES

Tomamos las columnas excepto la primera que contiene el nombre, luego se calcula el promedio de las materias y se redondea a dos decimales, ese cálculo lo asignamos a una nueva columna llamada Promedio.

Luego se muestran los primeros 10 elementos.

```
ShowTitleBox(
    "CALCULO DE PROMEDIOS",
    boxLineStyle=TitleBoxLineStyle.BLOCK,
    color=ConsoleColor.CYAN,
)
data["Promedio"] = data.iloc[:, 1:].mean(axis=1).round(2)
ShowDfHead(data, "Notas Master Data Science", 10)
```

```
CALCULO DE PROMEDIOS
iNotas Master Data Science: Primeros 10 elementos.
```

Nombre	Machine Learning	Big Data Analytics	Deep Learning	Data Visualization	Statistics & Probability	Promedio
Estudiante_1	84	65	30	74	87	68
Estudiante_2	34	52	56	50	69	52.2
Estudiante_3	95	74	40	100	35	68.8
Estudiante_4	53	96	54	60	46	61.8
Estudiante_5	86	30	65	59	66	61.2
Estudiante_6	83	57	51	46	92	65.8
Estudiante_7	68	94	10	67	17	51.2
Estudiante_8	9	16	8	68	90	38.2
Estudiante_9	98	53	66	41	70	65.6
Estudiante_10	0	70	77	21	95	52.6

OBTENER ESTUDIANTES APROBADOS Y REPROBADOS

```
UMBRAL_APROBADO = 60.0
data["Estado"] = data["Promedio"].apply(lambda x: "Aprobado" if x >= UMBRAL_APROBADO else "Reprobado")
dataAprobados = data[data.Promedio >= UMBRAL_APROBADO]
dataReprobados = data[data.Promedio < UMBRAL_APROBADO]
ShowDfHead(dataAprobados, " Master Data Science - Estudiantes Aprobados ✅", 10)
ShowDfHead(dataReprobados, " Master Data Science - Estudiantes Reprobados ❌", 10)
```

ℹ️ Master Data Science - Estudiantes Aprobados ✅: Primeros 10 elementos.

Nombre	Machine Learning	Big Data Analytics	Deep Learning	Data Visualization	Statistics & Probability	Promedio	Estado
Estudiante_1	84	65	30	74	87	68	Aprobado
Estudiante_3	95	74	40	100	35	68.8	Aprobado
Estudiante_4	53	96	54	60	46	61.8	Aprobado
Estudiante_5	86	30	65	59	66	61.2	Aprobado
Estudiante_6	83	57	51	46	92	65.8	Aprobado
Estudiante_9	98	53	66	41	70	65.6	Aprobado
Estudiante_17	99	83	100	53	100	87	Aprobado
Estudiante_18	92	65	56	38	87	67.6	Aprobado
Estudiante_20	99	79	79	65	21	68.6	Aprobado
Estudiante_26	96	31	54	78	62	64.2	Aprobado

ℹ️ Master Data Science - Estudiantes Reprobados ❌: Primeros 10 elementos.

Nombre	Machine Learning	Big Data Analytics	Deep Learning	Data Visualization	Statistics & Probability	Promedio	Estado
Estudiante_2	34	52	56	50	69	52.2	Reprobado
Estudiante_7	68	94	10	67	17	51.2	Reprobado
Estudiante_8	9	16	8	68	90	38.2	Reprobado
Estudiante_10	0	70	77	21	95	52.6	Reprobado
Estudiante_11	18	65	79	51	51	52.8	Reprobado
Estudiante_12	23	10	78	15	38	32.8	Reprobado
Estudiante_13	19	28	33	22	43	29	Reprobado
Estudiante_14	61	19	56	87	44	53.4	Reprobado
Estudiante_15	15	25	46	17	62	33	Reprobado
Estudiante_16	17	96	70	73	7	52.6	Reprobado

Establecemos una constante para tener un umbral y determinar si un estudiante aprueba o reprueba el curso.

Creamos una columna en la que contiene el valor aprobado o reprobado, de acuerdo al umbral.

Filtramos los aprobado y reprobados.

Mostramos los primeros 10 elementos de cada grupo.

OBTENER LOS PROMEDIOS DE LAS MATERIAS

```
promediosMateria = data.iloc[:, 1:-2].mean(axis=0).round(2)
dataPromediosMateria = pandas.DataFrame({
    "Materia": promediosMateria.index,
    "Promedio": promediosMateria.values
})
ShowDfHead(dataPromediosMateria, " Master Data Science - Promedios por materia", 10)
```

ℹ️ Master Data Science - Promedios por materia: Primeros 10 elementos.

Materia	Promedio
Machine Learning	50.14
Big Data Analytics	44.32
Deep Learning	55.08
Data Visualization	50.78
Statistics & Probability	57.7

Seleccionamos todas las filas y las columnas exceptuando la primera que es el nombre y las 2 últimas que contienen el estado que es categórica, el promedio de, estudiante. Obtenemos el promedio en el eje X.

Creamos un nuevo DataFrame con la información de la materia y el promedio.

Mostramos las primeras filas de este nuevo set de datos.

FASE 3 - CARGA Y VISUALIZACIÓN

Los resultados se presentarán mediante gráficos de barras que muestran la distribución de notas por materia y un diagrama tipo rosco que ilustra el porcentaje de aprobados y reprobados. Estas visualizaciones permiten analizar rápidamente el desempeño general de la cohorte.

```
HOST = "localhost"
PORT = 7374
app = Dash(__name__, external_stylesheets=[dbc.themes.QUARTZ])
print(f"Iniciando Dashboard Host:{HOST}, Port:{PORT}...")
app.title = "Master Data Science - Análisis de Notas"
```

```
figPastel = px.pie(
    data,
    names="Estado",
    title="Distribución de Aprobados vs Reprobados",
    color="Estado",
    color_discrete_map={"Aprobado": "#00cc96", "Reprobado": "#ef553b"},
    hole=0.3,
)
figPastel.update_layout(template="plotly_dark")


figBarras = px.bar(
    dataPromediosMateria,
    x="Materia",
    y="Promedio",
    title="Promedio de Calificaciones por Materia",
    text="Promedio",
    color="Promedio",
    color_continuous_scale="Blues",
)
figBarras.update_traces(texttemplate="%{text:.2f}", textposition="outside")
figBarras.update_layout(template="plotly_dark", xaxis_title="Materia", yaxis_title="Promedio")
```

VISUALIZACIÓN

Utilizando la lib dash creamos un dashboard para mostrar los gráficos.

Dash crea un servidor web que se ejecuta y sirve el sitio web que permite visualizar el dashboard.

Creamos las figuras para los promedios de las materias.

VISUALIZACIÓN

Creamos una función adicional para mostrar el mejor y peor estudiante en una tarjeta KPI.

```python
def KpiCard(title, value, subtitle):
    return dbc.Card(
        dbc.CardBody(
            [
                html.H6(title, className="text-muted"),
                html.H2(value, className="mb-1"),
                html.P(subtitle, className="text-muted mb-0"),
            ]
        ),
        className="h-100 text-center",
    )

kpiMejorEstudiante = KpiCard("🏆 Mejor Estudiante", f"{mejorEstudiante['Promedio']:.2f}", mejorEstudiante["Nombre"])
kpiPeorEstudiante = KpiCard("📉 Peor Estudiante", f"{peorEstudiante['Promedio']:.2f}", peorEstudiante["Nombre"])
```

```python
app.layout = dbc.Container(
    [
        dbc.NavbarSimple(
            brand="Dashboard de Calificaciones – Análisis de Notas",
            color="primary",
            dark=True,
            className="mb-4",
        ),

        dbc.Row(
            [
                dbc.Col(kpiMejorEstudiante, md=6),
                dbc.Col(kpiPeorEstudiante, md=6),
            ],
            className="g-4",
        ),

        dbc.Row(
            [
                dbc.Col(
                    dbc.Card(
                        dbc.CardBody([
                            html.H5("Distribución de Aprobados vs Reprobados"),
                            dcc.Graph(figure=figPastel)
                        ])
                    ),
                    md=6
                ),
                dbc.Col(
                    dbc.Card(
                        dbc.CardBody([
                            html.H5("Promedio de Calificaciones por Materia"),
                            dcc.Graph(figure=figBarras)
                        ])
                    ),
                    md=6
                ),
            ],
            className="g-4",
        ),
    ],
    fluid=True,
)
```

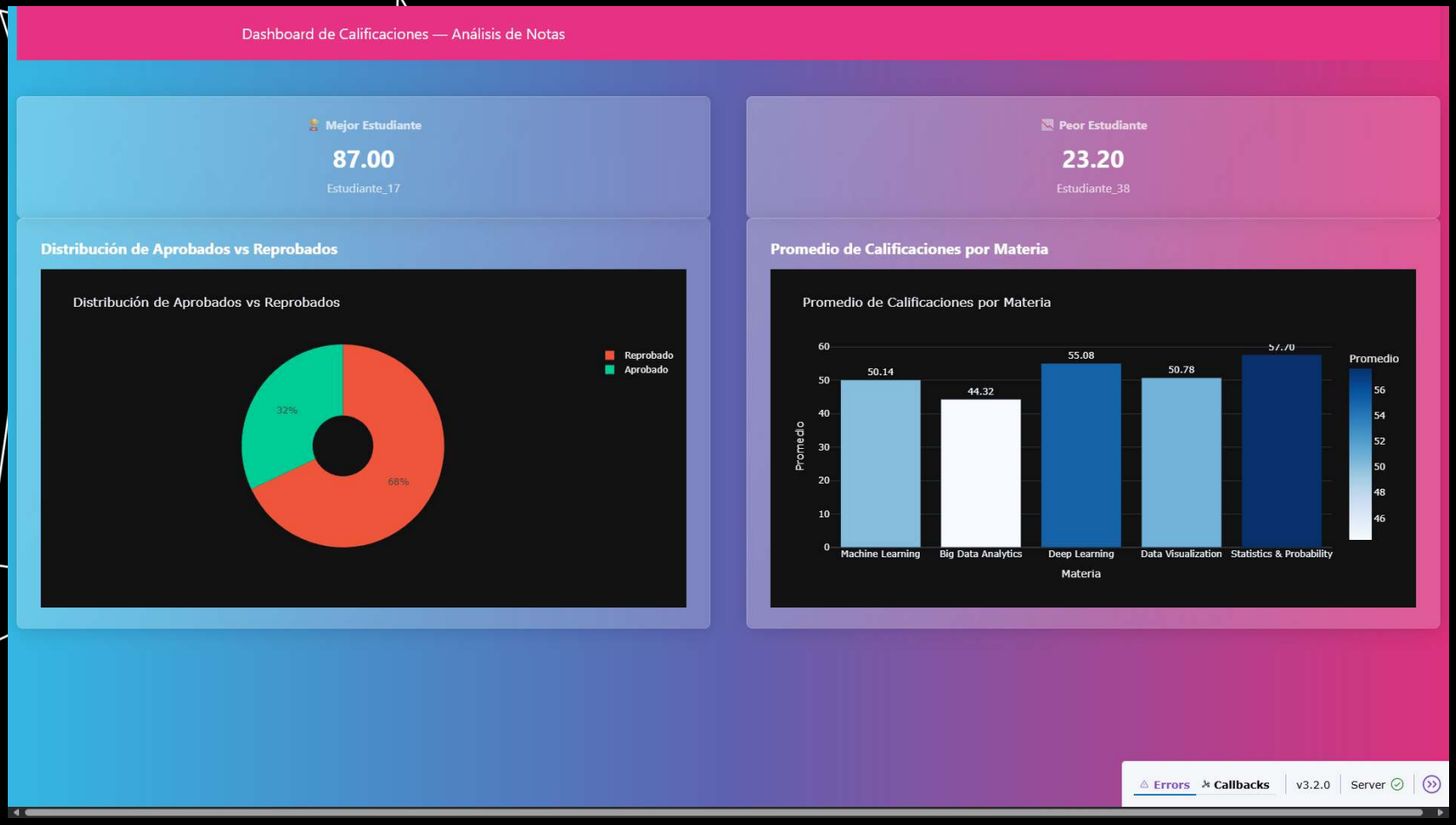
VISUALIZACIÓN

Creamos nuestro contenedor final(layout) utilizando lo antes creado.

Lanzamos nuestro servidor.

```python
if __name__ == "__main__":
    webbrowser.open(f"http://{HOST}:{PORT}")
    app.run(debug=True, port=PORT, host=HOST, use_reloader=False )
```

VISUALIZACIÓN - RESULTADO FINAL



GRACIAS

Código Fuente en:
https://github.com/UIDE-Tareas/5-Diseno-Procesos-ETL-Data-Science-Tarea1.git