COLOMBIA POTENCIA DE LA VIDA

TIC

Adquisición de Datos

TALENTO TECH | AZ PROYECTOS EDUCATIVOS | UTP Universidad Tecnológica de Pereira

Adquisición de Datos

Este tema titulado Adquisición de Datos, explora los pilares fundamentales para obtener datos de manera efectiva en proyectos de inteligencia artificial. Comienza con un análisis exhaustivo de diversas fuentes de datos, desde conjuntos de datos preexistentes hasta archivos CSV, bases de datos y APIs.

Se abordarán las consideraciones clave para seleccionar la fuente de datos más apropiada, adaptándola a las necesidades específicas de cada proyecto. Luego, nos enfocaremos en la práctica de cargar datos utilizando las funciones de Scikit-Learn, explorando ejemplos concretos que incluirán técnicas para una visualización inicial de datos.

Finalmente, destacaremos la importancia crucial de la calidad y relevancia de los datos, iniciando una discusión sobre estrategias para evaluar y mejorar continuamente la calidad de los datos adquiridos. Esta unidad proporcionará una base sólida para la construcción de modelos de inteligencia artificial respaldados por datos confiables y pertinentes.

Introducción a Diversas Fuentes de Datos

En inteligencia artificial, la selección adecuada de fuentes de datos es el primer paso hacia la construcción de modelos efectivos. En esta sección, se detallarán diversas fuentes de datos que actúan como materia prima para nuestros proyectos. Comenzaremos explorando conjuntos de datos preexistentes, que pueden provenir de repositorios públicos o privados, brindando un punto de partida valioso para una variedad de aplicaciones.

Estudiaremos la riqueza de información contenida en archivos CSV, una forma común y accesible de estructurar datos tabulares. Además, abordaremos las complejidades y posibilidades que ofrecen las bases de datos, destacando su capacidad para manejar grandes volúmenes de datos de manera eficiente. No olvidaremos las API (Interfaces de Programación de Aplicaciones), que nos permiten acceder a datos de plataformas externas de manera programática.

1. Conjuntos de Datos Preexistentes:

Definición: Conjuntos de datos recopilados previamente y disponibles para su uso. Pueden provenir de diversas fuentes, como investigaciones académicas, competiciones en línea, o repositorios especializados.

Ejemplos: MNIST para reconocimiento de dígitos, CIFAR-10 para clasificación de imágenes, IMDB para análisis de sentimientos en textos, entre otros.

Ventajas: Ahorro de tiempo en la recopilación de datos y validación de resultados previos.

2. Archivos CSV:

Definición: Archivos de texto plano que contienen datos organizados en filas y columnas, separados por comas. Son una forma común de almacenar datos tabulares.

Ejemplos: Datos de ventas, registros de usuarios, información financiera.

Ventajas: Facilidad de lectura y escritura, compatibilidad con una variedad de herramientas y lenguajes de programación.

3. Bases de Datos:

Definición: Almacenes estructurados de datos que permiten la recuperación eficiente de información. Pueden ser relacionales o NoSQL, dependiendo de la estructura de datos.

Ejemplos: MySQL, PostgreSQL, MongoDB.

Ventajas: Escalabilidad, integridad de datos, soporte para consultas complejas.

4. APIs (Interfaces de Programación de Aplicaciones):

Definición: Conjunto de reglas que permiten a las aplicaciones comunicarse entre sí. En el contexto de la IA, se utilizan para acceder a datos en tiempo real.

Ejemplos: Twitter API para datos de redes sociales, penWeatherMap API para datos meteorológicos.

Ventajas: Acceso a información actualizada en tiempo real, interacción con plataformas externas.

Consideraciones importantes

1. Calidad de los Datos: La precisión y consistencia de los datos son fundamentales para el rendimiento del modelo. Realizar una exploración y limpieza cuidadosa es esencial.

2. Privacidad y Ética: Al trabajar con datos, es crucial garantizar el cumplimiento de normativas de privacidad y ética en el manejo de la información.

Ejemplos de Fuentes de Datos en Inteligencia Artificial

Los siguientes ejemplos **NO son ejecutables**, sólo muestran las diferentes formas en las que nos podemos encontrar las bases de datos:

1. Conjuntos de Datos Preexistentes:

a) Para una base de datos en una URL particular:

```
import requests

url = "URL_DEL_CONJUNTO_DE_DATOS"

response = requests.get(url)

with open("dataset.zip", "wb") as file:

    file.write(response.content)
```

b) Para la Exploración de Datos:

```
import pandas as pd

mnist_data = pd.read_csv("mnist_dataset.zip")

print(mnist_data.head())
```

2. Archivos CSV:

a) Manipulación de Datos CSV:

```
import pandas as pd

data = pd.read_csv("data.csv")

print(sales_data.describe())
```

3. Bases de Datos:

a) Conexión a una Base de Datos:

```
from sqlalchemy import create_engine

# Conexión a una base de datos SQLite

engine = create_engine('sqlite:///nombre_base_de_datos.db')

# Recuperar una muestra de datos desde una tabla

query = "SELECT * FROM nombre_tabla LIMIT 5;"

result = engine.execute(query)

for row in result:

    print(row)
```

b) Consultas SQL:

```
from sqlalchemy import create_engine

engine = create_engine('sqlite:///nombre_base_de_datos.db')

# Consulta para obtener la cantidad total de registros

total_query = "SELECT COUNT(*) FROM nombre_tabla;"

total_records = engine.execute(total_query).scalar()

print(f"Total de registros: {total_records}")

# Consulta con filtro

filter_query = "SELECT * FROM nombre_tabla WHERE columna =

'valor';"

filtered_data = pd.read_sql_query(filter_query, engine)

print(filtered_data.head())
```

4. APIs (Interfaces de Programación de Aplicaciones):

a) Acceso a una API Pública:

```
import requests

url = "URL_DE_LA_API"

response = requests.get(url)

data = response.json()

relevant_info = data["informacion_relevante"]

print(relevant_info)
```

b) Gestión de Autenticación:

```
import requests

api_key = "TU_CLAVE_DE_API"

headers = {"Authorization": f"Bearer {api_key}"}

url = "URL_DE_LA_API_CON_AUTENTICACION"

response = requests.get(url, headers=headers)

data = response.json()

print(data)
```