

## Trabajo Practico Open VINO con Python

### Paso a Paso con Texto (Clasificación de Sentimientos)

#### 1. Instalar OpenVINO

Ejecuta el siguiente comando para instalar OpenVINO y las herramientas necesarias:

```
bash
```

Copiar código

```
pip install openvino-dev[onnx]
```

#### 2. Descargar un modelo preentrenado de texto

Usaremos un modelo compatible con OpenVINO para clasificación de texto. Un modelo típico es el de sentimientos preentrenado en IMDB. Ejecuta:

```
bash
```

Copiar código

```
omz_downloader --name bert-small-uncased-sst2
```

Este modelo clasifica texto en dos categorías:

- **0:** Sentimiento negativo.

- **1: Sentimiento positivo.**

### **3. Preparar un script simple**

Guarda este código como text\_infer.py:

python

Copiar código

```
from openvino.runtime
```

```
import Core import numpy as np
```

```
# Inicializar el motor OpenVINO core = Core()
```

```
# Ruta del modelo preentrenado
```

```
model_path = "intel/bert-small-uncased-sst2/FP16/bert-small-uncased-sst2.xml" compiled_model =  
core.compile_model(model_path, "CPU")
```

```
# Tokenizador básico para texto
```

```
from transformers import BertTokenizer tokenizer =  
BertTokenizer.from_pretrained("bert-base-uncased")
```

```
# Función para procesar el texto
```

```
def preprocess_text(text): tokens = tokenizer(text,  
padding="max_length", max_length=128, truncation=True,  
return_tensors="np") return tokens["input_ids"],  
tokens["attention_mask"]
```

### *# Texto de ejemplo para inferencia*

```
text = "I love using OpenVINO for AI tasks!" input_ids,  
attention_mask = preprocess_text(text)
```

### *# Realizar inferencia*

```
output = compiled_model([input_ids, attention_mask])  
predictions = output[compiled_model.output(0)]
```

### *# Interpretar resultados*

```
class_id = np.argmax(predictions) labels = ["Negative",  
"Positive"] print(f"Texto: {text}") print(f"Predicción:  
{labels[class_id]}")
```

## **4. Ejecutar el script**

Desde la terminal, ejecuta:

```
bash
```

Copiar código

```
python text_infer.py
```

## **5. Resultado**

El script imprimirá algo como:

```
makefile
```

Copiar código

Texto: I love using OpenVINO for AI tasks! Predicción:  
Positive

## Explicación del flujo

1. **Modelo de clasificación de texto:** El modelo toma una oración y predice si es positiva o negativa.
2. **Tokenización:** Usamos BertTokenizer para transformar el texto en números comprensibles por el modelo.
3. **Inferencia con OpenVINO:** Procesamos los datos con OpenVINO y obtenemos las predicciones.
4. **Resultado:** Mostramos si el texto tiene un sentimiento positivo o negativo.

## Ventajas de este ejemplo

- No utiliza imágenes, solo texto.
- Los estudiantes pueden probar fácilmente con sus propias oraciones.
- Introduce conceptos básicos como tokenización y clasificación.