COLOMBIA POTENCIA DE LA VIDA

TIC

Despliegue del Modelo

TALENTO TECH

AZ | PROYECTOS EDUCATIVOS

UTP Universidad Tecnológica de Pereira

La actividad Despliegue del Modelo se centra en la transición del desarrollo de modelos de inteligencia artificial hacia su implementación en entornos de producción. Abordaremos estrategias para empaquetar el modelo de manera eficiente y prepararlo para su despliegue. Además, exploraremos cómo integrar estos modelos en aplicaciones web, lo que nos permitirá utilizarlos de manera práctica en diversos escenarios del mundo real.

Estos subtemas nos equiparán con las habilidades necesarias para llevar nuestros modelos desde el laboratorio hasta el entorno de producción, facilitando su uso y aplicación en situaciones reales.

Despliegue del Modelo

El despliegue del modelo es el proceso crucial de llevar un modelo de inteligencia artificial desde su fase de desarrollo hasta su implementación en un entorno de producción. A continuación, se detalla este proceso, incluyendo el empaquetado del modelo y su integración en aplicaciones web.

1. Empaquetado del Modelo

El empaquetado del modelo implica preparar el modelo para su implementación en un entorno de producción. Algunas estrategias comunes incluyen:

Serialización del Modelo: Convertir el modelo entrenado en un formato que pueda ser guardado en disco, como pickle, HDF5 o formatos específicos de bibliotecas de aprendizaje automático como TensorFlow o PyTorch.	**Gestión de Dependencias:** Asegurarse de que todas las dependencias y bibliotecas necesarias para ejecutar el modelo estén disponibles en el entorno de producción. Se pueden utilizar herramientas como pip o conda para gestionar las dependencias.	**Pruebas y Validación:** Realizar pruebas exhaustivas del modelo empaquetado para garantizar que funcione correctamente en el entorno de producción y produzca resultados precisos y consistentes.

2. Integración en Aplicaciones Web

La integración de modelos de aprendizaje automático en aplicaciones web permite su uso práctico y accesible a través de interfaces de usuario. Algunas consideraciones importantes incluyen:

Diseño de la API: Crear una API (Interfaz de Programación de Aplicaciones) que permita a la aplicación web comunicarse con el modelo. Esto puede implicar el diseño de puntos finales (endpoints) para enviar datos al modelo y recibir predicciones.

Desarrollo Frontend y Backend: Desarrollar tanto el frontend (interfaz de usuario) como el backend (servidor) de la aplicación web para interactuar con la API del modelo. Esto puede involucrar el uso de frameworks como Flask o Django en el backend y HTML, CSS y JavaScript en el frontend.

COLOMBIA POTENCIA DE LA VIDA

TIC

Escalabilidad y Rendimiento: Asegurar que la aplicación web pueda manejar múltiples solicitudes de manera eficiente y que el modelo responda rápidamente a las consultas. Esto puede requerir técnicas de escalabilidad y optimización de código.

Seguridad: Implementar medidas de seguridad para proteger la integridad y confidencialidad de los datos transmitidos entre la aplicación web y el modelo, así como para prevenir ataques maliciosos.

Ejercicio 1: Empaquetado del Modelo

```python
# Ejercicio 1: Empaquetado del Modelo
from sklearn.datasets import load_iris
from sklearn.ensemble import RandomForestClassifier
import joblib

# Cargar el conjunto de datos
iris = load_iris()
X, y = iris.data, iris.target

# Entrenar el modelo
model = RandomForestClassifier()
model.fit(X, y)
```

```python
# Guardar el modelo entrenado en disco
joblib.dump(model, 'modelo_entrenado.pkl')

# Cargar el modelo desde el disco
loaded_model = joblib.load('modelo_entrenado.pkl')

# Realizar predicciones con el modelo cargado
predictions = loaded_model.predict(X[:5])
print("Predicciones:", predictions)
```

Ejercicio 2: Gestión de Dependencias

Para este ejercicio, crea un archivo llamado **requirements.txt** que contenga las dependencias necesarias, por ejemplo:

```
scikit-learn==0.24.2
joblib==1.0.1
```

Luego, ejecuta el siguiente comando en tu terminal para instalar las dependencias:

```
pip install -r requirements.txt
```