

Realización del deploy

Para esta aplicación se utiliza Flask para crear una interfaz web cargando el archivo de datos para luego mostrar los resultados después de presionar el botón de cálculo. Para lograr esto, se necesitan seguir estos pasos:

1. Instalar Flask: Se lo puede hacer utilizando el siguiente comando en la terminal o en el símbolo del sistema:

```
bashCopy code
pip install Flask
```

2. Crear el archivo de la aplicación Flask: Crear un nuevo archivo en VS Code y nombrarlo, por ejemplo, `app.py`. Aquí es donde se escribirá el código para la aplicación web utilizando Flask.
3. Importar los módulos necesarios: En el archivo `app.py`, importar los módulos necesarios, incluyendo Flask, pandas, numpy y los módulos relacionados con el modelo de regresión de árbol de decisión y el cálculo de incremento de producción.
4. Crear la aplicación Flask: Definir la aplicación Flask utilizando `Flask(__name__)`.
5. Crear las rutas y vistas: Definir las rutas y vistas para las páginas web de carga y cálculo. Esto incluirá una página con un formulario de carga de archivos y otra página que muestre los resultados después de realizar el cálculo.
6. Cargar el archivo CSV: En la vista de carga, permite al usuario seleccionar un archivo CSV y cargarlo en un DataFrame utilizando pandas.
7. Realizar el cálculo: En la vista de cálculo, realizar el cálculo de promedio anual y el modelo de regresión de árbol de decisión tal como lo hiciste en el código original.
8. Mostrar los resultados: Después de realizar el cálculo, muestra los resultados en la vista correspondiente.

Aquí está el código modificado del archivo `app.py` que sigue estos pasos:

```
pythonCopy code
import pandas as pd
import numpy as np
from flask import Flask, render_template, request

from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
```

```

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        # Obtener el archivo CSV cargado por el usuario
        archivo = request.files['file']
        df = pd.read_csv(archivo, encoding="latin-1")

        # Realizar el cálculo y obtener los resultados
        resultados = calcular_resultados(df)
        return render_template('resultados.html', resultados=resultados)
    return render_template('index.html')

def calcular_resultados(df):
    # Cálculo de promedio anual
    average_values = df.groupby('year')['data_value'].mean()

    # Resto del código para el cálculo y procesamiento de los datos
    # ... (como lo hiciste anteriormente)

    return resultados

if __name__ == '__main__':
    app.run(debug=True)

```

Además, se necesitar crear las plantillas HTML para la interfaz web. Crear una carpeta llamada `templates` en el mismo directorio que el archivo `app.py`, y dentro de esta carpeta, crear dos archivos llamados `index.html` y `resultados.html`. Estos archivos contendrán el código HTML para las páginas de carga y cálculo, respectivamente.

Aquí hay una versión simple de las plantillas HTML:

`index.html`:

```

htmlCopy code
<!DOCTYPE html>
<html>
<head>
    <title>Carga de Archivo CSV</title>
</head>
<body>
    <h1>Cargar Archivo CSV</h1>
    <form method="post" enctype="multipart/form-data">
        <input type="file" name="file" accept=".csv" required>
        <input type="submit" value="Cargar">
    </form>
</body>
</html>

```

resultados.html:

htmlCopy code

```
<!DOCTYPE html>
<html>
<head>
  <title>Resultados del Cálculo</title>
</head>
<body>
  <h1>Resultados del Cálculo</h1>
  <p>Incremento anual de data_value:</p>
  {% for year, incremento_info in resultados.items() %}
    <p>Año {{ year }}: Incremento: {{ incremento_info.incremento }},
Porcentaje: {{ incremento_info.incremento_porcentaje }}%</p>
  {% endfor %}
  <!-- Mostrar otros resultados aquí -->
</body>
</html>
```

Con estos cambios, ahora se puedes ejecutar la aplicación. Ejecutar `app.py` y acceder a `http://127.0.0.1:5000/` en un navegador para cargar el archivo CSV y realizar el cálculo. Los resultados se mostrarán en la página de resultados después de presionar el botón de cálculo. Se puede personalizarla según las necesidades. Se pueden agregar más funcionalidades y vistas según se requiera.