**Deployment on Flask**

Name: Seoyoung Kim

Batch Code: LISUM25

Date: Nov. 2023

Submitted to: https://github.com/syoungk7/Model\_Deployment

**1. Dataset:** Adult income dataset

https://www.kaggle.com/datasets/wenruliu/adult-income-dataset/

Fields: The dataset contains 16 columns

Target filed: Income -- The income is divided into two classes: <= 50K and > 50K

Number of attributes: 14 -- These are the demographics and other features to describe a person

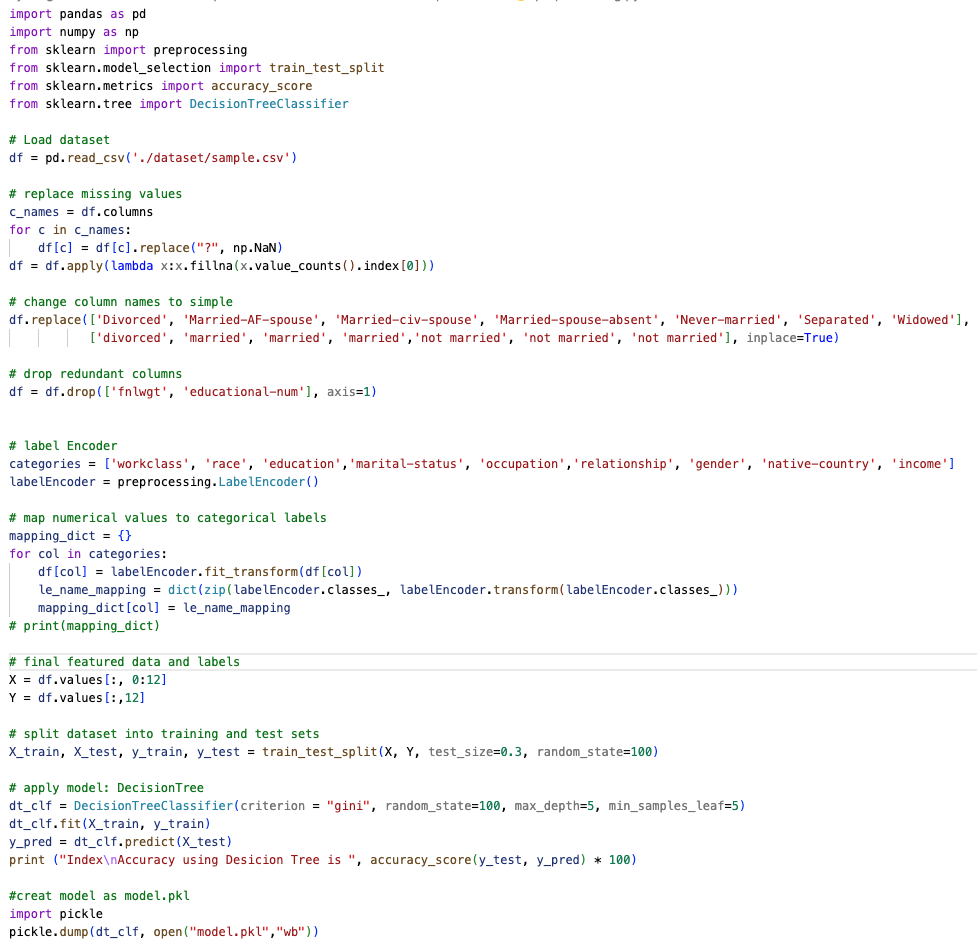
**2. Model used**: Decision Tree using DecisionTreeClassifier()

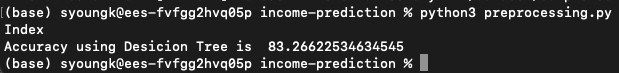
A Decision Tree is a popular machine learning algorithm used for both classification and regression tasks. It works by recursively partitioning the data into subsets based on the most significant attribute at each step.

The key concepts are:

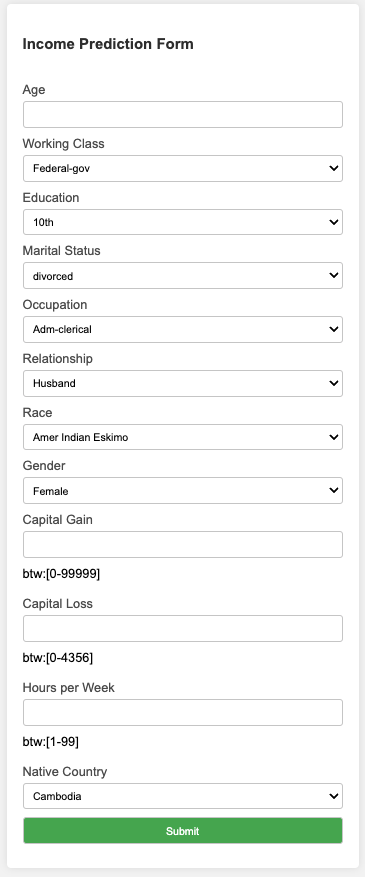
* **Nodes**: Decision Trees consist of nodes, which represent features or attributes in the dataset.
* **Root Node**: The topmost node in a Decision Tree is called the root node. It represents the entire dataset and is split into subsets based on a selected attribute.
* **Decision Nodes** (Internal Nodes): Nodes that follow the root node are decision nodes. They represent a decision or a test on an attribute, leading to different branches.
* **Leaves** (Terminal Nodes): The end nodes of a Decision Tree are called leaves or terminal nodes. They represent the final output or decision, such as a class label in classification or a numerical value in regression.
* **Branches**: The edges connecting nodes represent the outcome of a decision. The tree structure is formed by these branches.
* **Splitting**: At each decision node, the dataset is split into subsets based on a specific feature or attribute. The goal is to create homogeneous subsets that are more predictable.
* **Decision Criteria**: The criteria for splitting nodes are determined based on metrics like Gini impurity (for classification) or mean squared error (for regression). These metrics quantify the purity or homogeneity of the subsets.
* **Pruning**: Decision Trees can be prone to overfitting, capturing noise in the data. Pruning involves removing unnecessary branches to improve the model's generalization to new, unseen data.

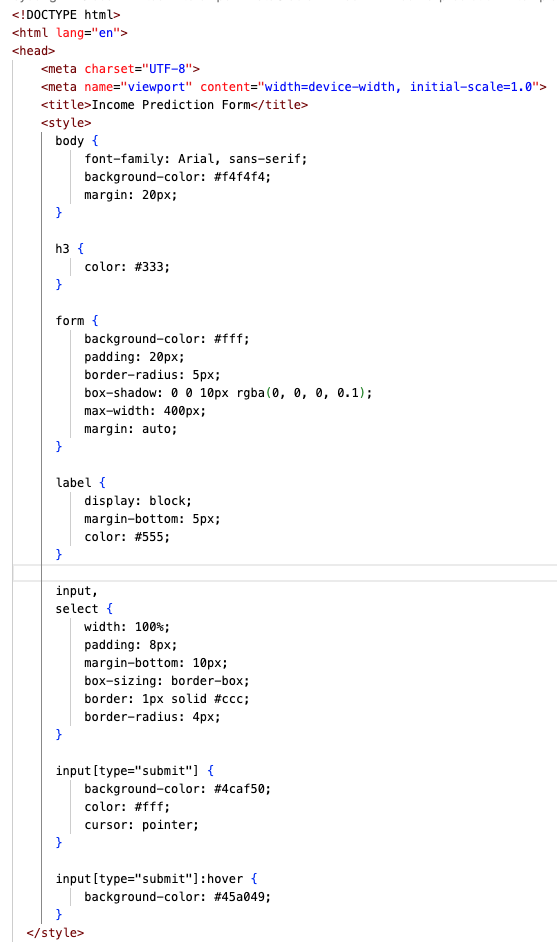
**3. Model build**: preprocessing.py

(reference: https://www.geeksforgeeks.org/deploy-machine-learning-model-using-flask/)



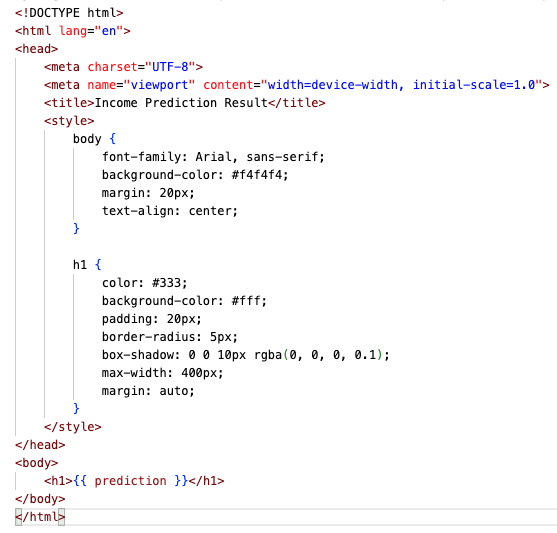
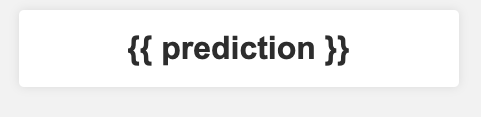
**4. HTML files for web deployment**: index.html and result.html

index.html





result.html



**5. Model deployment using Flask**

(reference: https://phoenixnap.com/kb/install-flask)

script.py



**Step 1**: Install virtualenv

pip install virtualenv

**Step 2**: Create an Environment

1. Make a separate directory for your project <income-prediction>
2. Move into the directory
3. Within the directory, create the virtual environment for Flask. When you create the environment, a new folder appears in your project directory with the environment’s name.
4. Create an Environment in Linux and MacOS: python3 -m venv venv

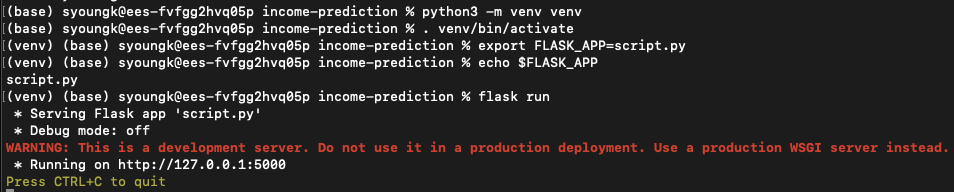
**Step 3**: Activate the Environment

Activate the virtual environment in Linux and MacOS with:. venv/bin/activate

**Step 4**: Install Flask

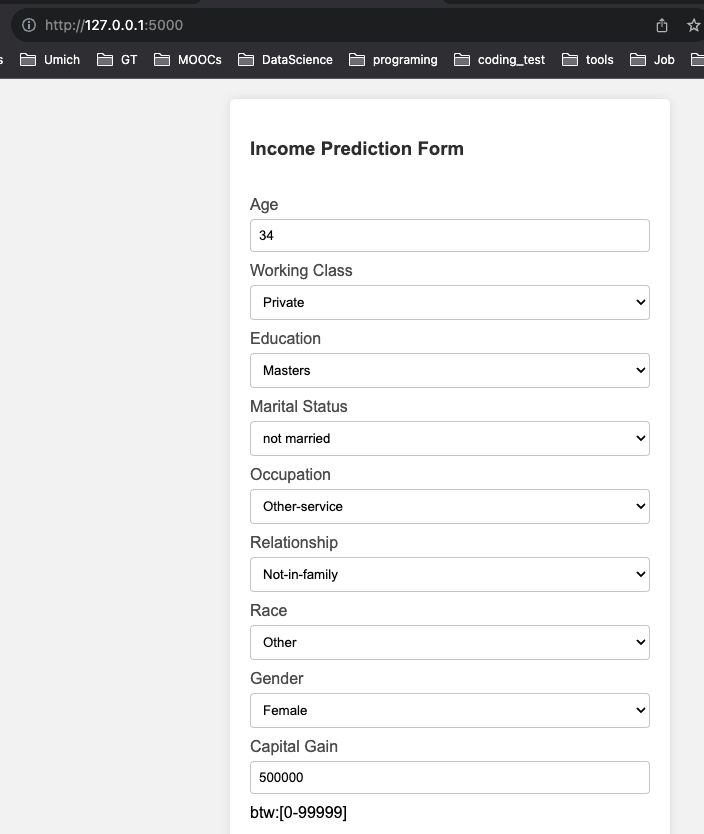
Install Flask within the activated environment using pip: pip install Flask

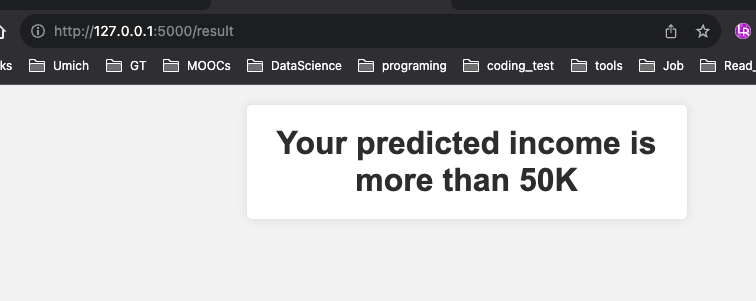
**Step 5**: Development

1. Using the console, navigate to the project folder using the cd command.
2. Set the FLASK\_APP environment variable: export FLASK\_APP=script.py
3. Run the Flask application with: flask run
4. Copy and paste the address into the browser to see the project running:

**6. Result**

**Input form**



**Prediction from Model**