Device Price Classification

Project Structure:

- Python
  - API
  - Data Modeling
- SpringBoot
  - DevicePriceClassification
  - DeviceClassification.postman_collection.json

The work steps are as follows:

1. Data processing and building/training the appropriate model.
2. Building the classification service that utilizes the trained model in the form of an API using FastAPI.
3. Building the application that interacts with the devices using SpringBoot.

To run the project:

1- Download the following dependencies to run the Python API:
   a. Python==3.8.4
   b. Scikit-learn==1.2.1
   c. Joblib==1.2.0
   d. matplotlib==3.6.3
   e. seaborn==0.12.2
   f. fastapi==0.94.1
   g. Jinja2==3.1.2
   h. uvicorn==0.21.1
   i. regex==2022.10.31
   j. gunicorn==21.2.0

2- After downloading the dependencies, go to the path Python/API and click on the run.bat file to execute or run ModelAPI.py.

3- The API can be tested directly after running it through the test interface: Python/API/index.html

4- After running the Python API, you can start the SpringBoot\DevicePriceClassification\Devices-Price-Classification project and test various requests, as the endpoints have been documented in the SpringBoot\DeviceClassification.postman_collection.json file.

Additional information:

1- The process of data preprocessing and training the best model is contained within the file named " Python\Data Modeling \ Data Modeling.ipynb ".

2- It is worth mentioning that the proposed classifier deals only with the top 10 features, which are: selected_features = ['battery_power', 'int_memory', 'mobile_wt', 'n_cores', 'px_height', 'px_width', 'ram', 'sc_h', 'sc_w', 'talk_time']. These details are outlined in the file named " Python\Data Modeling \ Data Modeling.ipynb ".

3- All unclassified test samples have been classified and the results saved in the file named " Python\Data Modeling \ labeled_test_data.csv".

Important Note:

1. **Python API deals with samples in Python\API\testdata.csv**