Amazon ML Challenge

Team Name : Error 404 Brain not Found

Vaishnavi Prajapati | Ayush Suryavanshi | Soumya Gupta

Machine Learning Model for Entity Value Extraction from Product Images

Project Overview

The goal of the project was to develop a machine learning model capable of extracting specific entity values such as weight, volume, voltage, and dimensions from product images. This helps in fields like healthcare, e-commerce, and content moderation, where precise product information is vital.

Project Scope

- **Problem Statement:** In this hackathon, the goal is to create a machine learning model that extracts entity values from images. This capability is crucial in fields like healthcare, e-commerce, and content moderation, where precise product information is vital. As digital marketplaces expand, many products lack detailed textual descriptions, making it essential to obtain key details directly from images. These images provide important information such as weight, volume, voltage, wattage, dimensions, and many more, which are critical for digital stores.
- **Objective:**
    - Extract enlisted entities (weight, volume, voltage, dimensions, etc.) from product images.
    - Output is the extracted data in a csv file.

Data Details

- **index:** A unique identifier (ID) for the data sample.
- **image_link:** Public URL where the product image is available for download.
- **group_id:** Category code of the product.
- **entity_name:** Product entity name. For example, “item_weight”.
- **entity_value:** Product entity value. For example, “34 gram”.

Output Format:

The output file should be a CSV with 2 columns:

- **index:** The unique identifier (ID) of the data sample.
- **prediction:** A string which should have the following format: “x unit” where x is a float number in standard formatting and unit is one of the allowed units.

Data Preprocessing

Libraries used :
Pandas, Numpy, Scikit-learn, TensorFlow/Keras, Tesseract, Matplotlib/Seaborn

Steps for Data Preprocessing:

- Load the data and inspecting missing values
- Feature Engineering
- Handling unseen labels and standardise numerical features
- Extracting image features and prepare data for model training

Our Approach

Libraries used : Scikit learn

Approach 1: Using LabelEncoder and Handling Unseen Labels

- Label Mapping: We create a dictionary label_mapping from the LabelEncoder's fitted classes.
- Map Known Labels: We apply this mapping to the test dataset. If any group_id in the test set is not found in the train set, it will result in a NaN.
- Handle Unseen Labels: Any missing labels are replaced with -1 (or another placeholder value). You can later decide how to handle this placeholder in your model (e.g., treat it as a separate category).

Approach 2: Use OrdinalEncoder (Alternative to LabelEncoder). The OrdinalEncoder from sklearn.preprocessing has an argument handle_unknown='use_encoded_value' which can encode unseen categories with a special value. This is a more flexible approach for handling unknown labels.

Model Used

Once you've engineered your features, you can experiment with different models to predict entity_value. Some models to consider are:

- **Linear Regression or Ridge/Lasso Regression**
- **Tree-based Models**
- **Deep Learning Models**

Details of Files

- EDA.ipynb: code to preprocess text
- Model Training.ipynb: code for model training
- test_out.csv: output file
- train.csv: to train the model and check the prediction
- xgboost_model.pkl: converting multiple submission file to single submission file