Note:

The option to leave a custom review on the product listing and the website hosting will be incorporated by the next round.
On our homepage, a red border around the product indicates that the product has been flagged as bogus, according to our model.
Our algorithm (including all the ML models) has been trained on real-world datasets and is considerably accurate. (Outputs low review score for authentic amazon products)
The website is optimised to run on devices of all aspect ratios.

Project Pitch

Through this project, we have decided to solve two crucial problems considering factors like scalability and depth of solution.

Problem 1:

Fraudulent and Bogus Reviews:
It is estimated that about 11-15% of reviews on e-commerce websites are fraudulent reviews, and in regards to Amazon, the numbers stand around 40%; they stem from various factors such as bot reviews, review swapping, purchased/incentivized reviews, etc.

To tackle this problem, we have created a scoring system which hinges on the following parameters:

- Verification Check
- An Auto-encoder which detects anomalies in a review
- A Classifier, which is trained on a dataset of labelled reviews.
- Sentiment Analysis
- Miscellaneous parameters like whether a review was tagged as helpful.

Each of these parameters has been assigned a ** Weight ** (which can be changed as per the requirement), and based on this weight, a mathematical formula is calculated, which assigns a score to each review. A higher score signifies a higher tendency of the review to be fake.

Problem 2:

Counterfeit/Fake Product Detection:
The proliferation of counterfeit products poses a significant threat to consumer trust, brand integrity, and fair market competition.

Similar to how we solved Problem 1, we chose specific parameters which define a scoring system. These parameters are:

- Average review score for a particular product which is taken from the previous problem.
- Price reletive to other products of the same category
- Quality of the listing (Readability, richness, detail, etc.)
- Sentiment Analysis

The models have been coded on python and have been integrated with a React.js frontend through Flask. To create the ML and NLP models, sci-kit learn, tensorflow and NLTK has been used.

The **USP** of our product is the extensive analysis giving an accurate score. The project is easy to scale and this can be achieved by using pre-trained models for image analysis, using a better dataset (considering database constraints).

File Types:

- Main.py: Contains the actual machine learning models and all the data frame used in the models.
- transfromer_pipeline.py: This contains a prebuilt transformer-pipeline not currently used due to token constraints. Added to demonstrate scalability of project.
- backend: Contains our backend server.
- website/amazon: Contains our frontend files.
- Csv files:
  - a) Classifer_dataset.csv: CSV file used to train the classifier.(See problem 1, paramter 3 in main.py) (source: https://osf.io/3vds7)
  - b) final_dataset.csv: Main Dataset used for obtaining the reviews.
  - c) product.csv: Main dataset used for obtaining products. (derived from final_dataset.csv)
  - d) Finalfrontenddb.csv: Dataset used for making the backend node.js server

How to use:

- Ensure that node.js and npm are installed and can be used from the cmd/terminal.
- Clone the repository and navigate to the project location in the cmd/terminal.
- To run the Python script (assuming you have Python installed), in cmd/terminal run ***pip install -r requirements.txt.***
- In a new terminal, navigate to the backend folder and *run* ***node server.js.***
- From the counterfeit_Detection folder, navigate to the amazon folder (***cd ./website/amazon***) and in terminal run ***npm start.***