Problem Statement

ML Challenge 2025 Problem Statement

Smart Product Pricing Challenge

In e-commerce, determining the optimal price point for products is crucial for marketplace success and customer satisfaction. Your challenge is to develop an ML solution that analyzes product details and predict the price of the product. The relationship between product attributes and pricing is complex - with factors like brand, specifications, product quantity directly influence pricing. Your task is to build a model that can analyze these product details holistically and suggest an optimal price.

Data Description:

The dataset consists of the following columns:

1. sample_id: A unique identifier for the input sample
2. catalog_content: Text field containing title, product description and an Item Pack Quantity(IPQ) concatenated.
3. image_link: Public URL where the product image is available for download. Example link - https://m.media-amazon.com/images/I/71XfHPR36-L.jpg (https://m.media-amazon.com/images/I/71XfHPR36-L.jpg) To download images, use the download_images function from src/utils.py. See sample code in src/test.ipynb.
4. price: Price of the product (Target variable - only available in training data)

Dataset Details:

Training Dataset: 75k products with complete product details and prices
Test Set: 75k products for final evaluation

Output Format:

The output file should be a CSV with 2 columns:

1. sample_id: The unique identifier of the data sample. Note the ID should match the test record sample_id.
2. price: A float value representing the predicted price of the product.

Note: Make sure to output a prediction for all sample IDs. If you have less/more number of output samples in the output file as compared to test.csv, your output won't be evaluated.

File Descriptions:

Source files

1. src/utils.py: Contains helper functions for downloading images from the image_link. You may need to retry a few times to download all images due to possible throttling issues.

2. sample_code.py: Sample dummy code that can generate an output file in the given format. Usage of this file is optional.

Dataset files

1. dataset/train.csv: Training file with labels (price).
2. dataset/test.csv: Test file without output labels (price). Generate predictions using your model/solution on this file's
   data and format the output file to match sample_test_out.csv
3. dataset/sample_test.csv: Sample test input file.
4. dataset/sample_test_out.csv: Sample outputs for sample_test.csv. The output for test.csv must be formatted in the
   exact same way. Note: The predictions in the file might not be correct
   Constraints:

1. You will be provided with a sample output file. Format your output to match the sample output file exactly.
2. Predicted prices must be positive float values.
3. Final model should be a MIT/Apache 2.0 License model and up to 8 Billion parameters.
   Evaluation Criteria:

Submissions are evaluated using Symmetric Mean Absolute Percentage Error (SMAPE): A statistical measure that expresses the relative difference between predicted and actual values as a percentage, while treating positive and negative errors equally.

Formula:

$$\text{SMAPE} = (1/n) * \Sigma |\text{predicted\_price} - \text{actual\_price}| / ((|\text{actual\_price}| + |\text{predicted\_price}|)/2)$$

Example: If actual price = $100 and predicted price = $120

$$\text{SMAPE} = |100-120| / ((|100| + |120|)/2) * 100\% = 18.18\%$$

Note: SMAPE is bounded between 0% and 200%. Lower values indicate better performance.

Leaderboard Information:

Public Leaderboard: During the challenge, rankings will be based on 25K samples from the test set to provide real-time feedback on your model's performance.

Final Rankings: The final decision will be based on performance on the complete 75K test set along with provided documentation of the proposed approach by the teams.

Submission Requirements:

1. Upload a test_out.csv file in the Portal with the exact same formatting as sample_test_out.csv
2. All participating teams must also provide a 1-page document describing:
   - Methodology used
   - Model architecture/algorithms selected
   - Feature engineering techniques applied
   - Any other relevant information about the approach
   - Note: A sample template for this documentation is provided in Documentation_template.md

Academic Integrity and Fair Play:

⚠ STRICTLY PROHIBITED: External Price Lookup

Participants are STRICTLY NOT ALLOWED to obtain prices from the internet, external databases, or any sources outside the provided dataset. This includes but is not limited to:

- Web scraping product prices from e-commerce websites
- Using APIs to fetch current market prices
- Manual price lookup from online sources
- Using any external pricing databases or services

Enforcement:

- All submitted approaches, methodologies, and code pipelines will be thoroughly reviewed and verified
- Any evidence of external price lookup or data augmentation from internet sources will result in immediate disqualification

Fair Play: This challenge is designed to test your machine learning and data science skills using only the provided training data. External price lookup defeats the purpose of the challenge.

Tips for Success:

- Consider both textual features (catalog_content) and visual features (product images)
- Explore feature engineering techniques for text and image data
- Consider ensemble methods combining different model types
- Pay attention to outliers and data preprocessing