***Project Plan***

**1. Initial Project Setup**

**Goal:** Create the project directory structure, set up version control with GitHub, and establish a working environment.

**Steps:**

1. **Create Project Directory:**
   * Use the following structure:
   * ecommerce-product-tool/
   * │
   * ├── data/
   * │ ├── raw/ # Raw scraped data from Alibaba and resale platforms
   * │ ├── processed/ # Cleaned and merged datasets
   * │ ├── results/ # Final datasets and analysis results
   * │
   * ├── notebooks/
   * │ ├── 1\_data\_collection.ipynb # Data scraping scripts
   * │ ├── 2\_data\_engineering.ipynb # Cleaning and preprocessing
   * │ ├── 3\_modeling.ipynb # Clustering, regression, sentiment analysis
   * │ ├── 4\_visualization.ipynb # Creating visualizations and summaries
   * │
   * ├── scripts/
   * │ ├── scrape\_alibaba.py # Script for Alibaba scraping
   * │ ├── scrape\_resale.py # Script for resale platform scraping
   * │ ├── data\_cleaning.py # Data cleaning and preprocessing script
   * │ ├── clustering.py # Supplier clustering model
   * │ ├── profitability\_model.py # Profitability prediction script
   * │ ├── sentiment\_analysis.py # Sentiment analysis script
   * │ ├── run\_pipeline.py # Main pipeline script to automate everything
   * │
   * ├── tests/
   * │ ├── test\_scraping.py # Unit tests for scraping
   * │ ├── test\_data\_cleaning.py # Unit tests for data preprocessing
   * │ ├── test\_models.py # Unit tests for models
   * │
   * ├── results/
   * │ ├── insights.pdf # Final report or summary
   * │ ├── visualizations/ # Generated plots and charts
   * │
   * ├── .gitignore # Ignore unnecessary files
   * ├── README.md # Project overview and instructions
   * ├── requirements.txt # Python dependencies
   * └── LICENSE # Project license (optional)
2. **Initialize Git Repository:**
   * Run the following commands:
   * cd ecommerce-product-tool/
   * git init
   * git add .
   * git commit -m "Initial project setup"
3. **Link with GitHub:**
   * Create a new repository on GitHub.
   * Link it to your local project:
   * git remote add origin <your-repo-url>
   * git branch -M main
   * git push -u origin main
4. **Create a Virtual Environment:**
   * Set up an isolated environment:
   * python -m venv env
   * source env/bin/activate # Use `env\Scripts\activate` on Windows
   * Install dependencies:
   * pip install pandas numpy scikit-learn matplotlib seaborn nltk scrapy selenium
   * pip freeze > requirements.txt

**2. Data Collection**

**Goal:** Scrape product and supplier data from Alibaba and resale platform data from Amazon or Jumia.

**Steps:**

1. **Scrape Alibaba Data:**
   * Write a script (scripts/scrape\_alibaba.py) to collect:
     + Product Name, Price Range, MOQ, Supplier Rating, Delivery Time, Verified Status.
   * Save the raw data to data/raw/alibaba\_data.csv.
2. **Scrape Resale Data:**
   * Write a script (scripts/scrape\_resale.py) to collect:
     + Product Name, Resale Price, Average Rating, Total Reviews, Customer Feedback.
   * Save the raw data to data/raw/resale\_data.csv.
3. **Commit Changes:**
   * Push the updated scripts and data:
   * git add scripts/scrape\_alibaba.py scripts/scrape\_resale.py data/raw/
   * git commit -m "Add data collection scripts and raw data"
   * git push

**3. Data Cleaning and Preprocessing**

**Goal:** Standardize, clean, and merge data for further analysis.

**Steps:**

1. **Clean and Standardize Data:**
   * Write a script (scripts/data\_cleaning.py) to:
     + Standardize price ranges to averages.
     + Normalize supplier details (e.g., delivery time as integers).
     + Handle missing or inconsistent data.
2. **Merge Datasets:**
   * Combine Alibaba and resale platform data into a single dataset.
   * Add calculated fields like Total Cost and Profit Margin.
3. **Save Processed Data:**
   * Save the cleaned dataset to data/processed/processed\_data.csv.
4. **Commit Changes:**
   * Push the cleaning script and processed data:
   * git add scripts/data\_cleaning.py data/processed/
   * git commit -m "Add data cleaning and merging script"
   * git push

**4. Modeling and Analysis**

**Goal:** Apply data science techniques to extract insights.

**Steps:**

1. **Clustering:**
   * Write a script (scripts/clustering.py) to group suppliers using K-Means clustering.
   * Save cluster labels to data/results/supplier\_clusters.csv.
2. **Profitability Prediction:**
   * Write a script (scripts/profitability\_model.py) to train a regression model that predicts profit margins.
   * Save predictions to data/results/profit\_predictions.csv.
3. **Sentiment Analysis:**
   * Write a script (scripts/sentiment\_analysis.py) to:
     + Analyze customer reviews for sentiment.
     + Save sentiment scores and keywords to data/results/sentiment\_analysis.csv.
4. **Commit Changes:**
   * Push analysis scripts and results:
   * git add scripts/ data/results/
   * git commit -m "Add clustering, profitability, and sentiment analysis"
   * git push

**5. Visualization and Reporting**

**Goal:** Create visualizations and compile insights into a report.

**Steps:**

1. **Generate Visualizations:**
   * Create charts and plots using a script (scripts/visualization.py):
     + Bar charts for profitability.
     + Scatter plots for supplier reliability vs. cost.
     + Word clouds for customer feedback.
2. **Compile Insights:**
   * Write a final report summarizing recommendations and findings.
   * Save it as results/insights.pdf.
3. **Commit Changes:**
   * Push visualization scripts and reports:
   * git add scripts/visualization.py results/
   * git commit -m "Add visualizations and final report"
   * git push

**6. Automation**

**Goal:** Combine all processes into a single pipeline script.

**Steps:**

1. **Write Pipeline Script:**
   * Create scripts/run\_pipeline.py to automate:
     + Data collection.
     + Cleaning and merging.
     + Modeling and analysis.
     + Visualization and reporting.
2. **Run the Pipeline:**
   * Test the pipeline to ensure it works end-to-end:
   * python scripts/run\_pipeline.py
3. **Commit Changes:**
   * Push the pipeline script:
   * git add scripts/run\_pipeline.py
   * git commit -m "Add automated pipeline script"
   * git push

**Final Deliverables:**

1. **Cleaned Data:** Structured datasets in data/processed/.
2. **Analysis Results:** Predictions, clusters, and sentiment scores in data/results/.
3. **Visualizations:** Charts and plots in results/visualizations/.
4. **Report:** Insights and recommendations in results/insights.pdf.
5. **Automated Pipeline:** Script to reproduce the entire process.

Let me know if you'd like to discuss any part further!