CRISP-DM Document: Sentiment Analysis of Skin Care Products

1. **Business Understanding**

1.1. Project Overview

This project uses Natural Language Processing (NLP) techniques to evaluate user-generated skincare product reviews, with the objectives of identifying relevant sentiment patterns across distinct customer groups. The dataset compromises hundreds of evaluation, each providing subjective input on product efficacy, which is frequently linked to particular skin features such as tone type.

The primary goal is to extract sentiment (positive, negative, neutral) from free text reviews and detect patterns associated with certain product features, skin types, and brands. The approach includes text preprocessing, tokenization, sentiment labelling and model training with text classification- appropriate machine learning methods. Advance visualization and analysis are utilized to understand how sentiments change among demographic or skin type groupings.

Unlike recommendation systems, which advice things, this NLP research focuses on understanding why certain products are evaluated positively, providing consumers and companies with deeper, data-driven understandings. By uncovering sentiment trends in large scale textual data, our study helps to provide more transparent skincare experiences and data-driven product development.

1.2. Business Problem

In the beauty and skincare market, user evaluations are valuable yet under-utilized source of consumer information. These evaluation frequently include comprehensive personal experience with products, emphasizing their impact on different skin kinds, tones, and condition. However, due to unstructured and subjective nature of the data, companies, researchers and potential customers find it challenging to properly assess sentiment are establish trends across enormous amounts of input.

Most analytics now rely on star rating or keyword mentions, which oversimplify user sentiment and fail to capture complex thoughts like mixed sentiment or conditional satisfaction (for example, “great for dry skin but irritating on sensitive ears”). This lack of granularity may lead to bad product development decisions, unproductive marketing tactics and inadequate customer service.

This project deal with the demand for more understanding into skincare product evaluations by creating an NLP_ powered sentiment analysis system. By using powerful natural processing language techniques to identify and evaluate user sentiment, the system hopes to derive significant pattern that represent real-world product success across abroad user base. The study will also look at links between sentiment and variables like skin tone, skin type and brand to get a better understanding of how various demographics react to skincare products.

1.3. Project Objectives:

- To perform sentiment analysis on customer reviews on products to enhance customer satisfaction.
- To use data visualizations tools to assess product categories and brand popularity to guide companies on future pricing Assess price range across various products to improve affordability of products by customers.
- To detect common keywords and phrases to highlight positive, neutral and negative reviews on products to understand customer satisfaction and dissatisfaction.
- To provide actionable insights in order to improve customer satisfaction across various products, brands and categories.

1.4. Produce Project Plan

The project will follow the CRISP-DM (Cross Industry Standard Process for Data Mining) methodology and is scheduled for completion within **three weeks**. Each week is dedicated to specific phases of the data mining process to ensure focused execution and timely delivery.

1.4.1 Project Timeline

Week	Phases	Key Activities
Week 1	**Business Understanding**, **Data Understanding**, **Data Preparation**	- Define business and data mining objectives - Explore and assess data quality - Clean and preprocess the dataset (e.g., remove nulls, tokenize text, drop redundant columns)
Week 2	**Modeling**, **Evaluation**	- Apply sentiment analysis models (e.g., logistic regression, SVM, or deep learning) - Fine-tune model parameters

Week	Phases	Key Activities
		- Evaluate performance using metrics like accuracy, precision, recall, and F1-score
Week 3	**Deployment**	- Document key insights and results - Visualize findings through charts or dashboards - Prepare final report and presentation for stakeholders

1.4.2 Milestones and Deliverables

- **End of Week 1**: Clean, structured, and well-understood dataset ready for modeling.
- **End of Week 2**: Trained sentiment classification model with performance evaluation.
- **End of Week 3**: Final report, visual summaries, and actionable insights delivered.

2. Data Understanding

2.1 Collect Initial Data

The dataset used in this project is a merged collection of reviews extracted from five CSV files (reviews_0-250_masked.csv to reviews_1250-end_masked.csv). After merging and cleaning, the resulting dataset includes over **200,000 reviews** and the following key features:

- **User-related attributes**: skin_type, skin_tone, eye_color, hair_color
- **Product information**: product_name, brand_name, price_usd, price_category
- **Review content**: review_text, review_title, rating, is_recommended
- **Engagement metrics**: total_feedback_count, total_pos_feedback_count, total_neg_feedback_count

The data was loaded using pandas, and basic inspection steps confirmed a consistent structure across files.

2.2 Describe Data

Initial inspection and preprocessing revealed the following:

- **Missing Values**: Features like skin_tone, eye_color, and hair_color contained high percentages of missing data. These were imputed using the mode, while is_recommended was filled using its median.
- **Redundant Columns**: Columns such as Unnamed: 0, submission_time, and similar duplicates were dropped.
- **Shape of the Data**: After cleaning, the dataset had **~1,500 rows** and **16 relevant columns**.
- **Ratings**: The rating column ranges from 1 to 5, providing a numerical representation of customer satisfaction.
- **Price Information**: price_usd values were used to generate a new price_category column with binned price ranges (e.g., <$20, $20-50, etc.).

2.3 Explore Data

The following exploratory analyses were conducted:

- **Brand Popularity**: A bar chart of the top 20 most reviewed brands showed dominant brand participation.
- **Skin Tone and Skin Type Distribution**: Horizontal bar charts were created to display the frequency of skin tones and types among reviewers.
- **Rating Distribution**: A count plot revealed that most ratings skewed toward higher values (e.g., 4 or 5 stars).
- **Price Distribution**: Histogram and category plots highlighted that most products fall within the $20–$50 price range.
- **Outliers and Feedback Counts**:
  - Box plots exposed outliers in price_usd and feedback-related columns.
  - These columns were capped at the 95th percentile to prevent modeling distortion.
- **Feedback Summary**: A comparison of positive, negative, and total feedback counts showed significantly more positive interactions.
- **Brand Pricing**: The top 20 most expensive and 20 most affordable brands were visualized via horizontal bar charts.
- **Price Segmentation**: The price_category feature enabled segmented analysis of product pricing.

2.4 Verify Data Quality

Several quality assurance steps were taken:

- **Imputation**: Missing categorical values were imputed appropriately to maintain dataset size.
- **Text Fields**: Records missing review_text or review_title were dropped to preserve analysis validity.
- **Data Integrity**: After imputation and cleaning, a null value check (data.isna().sum()) confirmed there were no remaining missing values in the relevant columns.
- **Consistency Checks**: Distribution plots for each key variable were used to identify and address skewness, extreme outliers, or irregular values.
- **Feedback Columns**: Extreme feedback counts were capped at the 95th percentile to reduce the influence of outliers on modeling.

3. Text Preprocessing and Feature Engineering

3.1 Objective

The goal of preprocessing and feature engineering was to convert raw, user-generated product reviews into a clean, structured, and machine-readable format. This step is crucial to ensure the accuracy and reliability of downstream modeling, especially given the subjective and informal nature of customer reviews.

3.2. Text Preprocessing

Customer reviews were inherently noisy, requiring structured preprocessing to minimize dimensionality while preserving semantic meaning.

- **Lowercasing**: All text was converted to lowercase to standardize inputs and reduce vocabulary sparsity.
- **Punctuation & Special Characters**: Removed using regular expressions to clean up tokens.
- **Stopword Removal**: Generic English stopwords were removed using NLTK's standard list, enhanced with domain-specific terms like "product," "review," etc.
- **Tokenization**: Text was split into meaningful word units using word_tokenize.
- **Lemmatization**: Words were reduced to their root form (e.g., "running" → "run") using spaCy, improving generalization.

3.3 Feature Engineering

Several derived features were engineered to enrich the dataset and boost model performance:

- **Sentiment Labels**: Based on rating values:
  - Ratings > 3 → **Positive**
  - Rating = 3 → **Neutral**
  - Ratings < 3 → **Negative**

  These sentiment classes formed a supplementary categorical label that aligned with user satisfaction.

- **TF-IDF Vectorization**: The processed text was vectorized using Term Frequency–Inverse Document Frequency (TF-IDF), which emphasizes unique and informative terms across reviews. The top 500 terms were retained to ensure meaningful, compact representations.
- **Demographic Flags**: Binary indicators were created for:
  - skin_tone: E.g., light, medium, deep.
  - skin_type: E.g., oily, combination, sensitive.

  These features enabled demographic-aware analysis and potential personalization of insights.

3.4 Dataset Splitting and Resampling

- **Train/Test Split**: The dataset was split into 80% training and 20% testing sets. Stratification ensured that the is_recommended classes were evenly represented across both subsets.
- **Class Imbalance Handling**: SMOTE (Synthetic Minority Oversampling Technique) was applied **only to the training set** to synthetically balance the is_recommended variable, which was initially skewed.

3.5 Tools and Libraries Used

- **Pandas / NumPy**: Data handling and transformations.
- **NLTK / spaCy**: Text processing (tokenization, stopword filtering, lemmatization).
- **Scikit-learn**: TF-IDF vectorization, SMOTE, model input preparation.
- **Matplotlib / Seaborn**: Data visualization and distribution analysis.

3.6 Challenges and Considerations

- **Noise in Language**: Informal grammar, slang, and typos were hard to fully normalize.
- **Vocabulary Sparsity**: TF-IDF helped, but using contextual embeddings (e.g., BERT) could further improve performance.
- **Bias Risk**: Product reviews may reflect individual bias. Care was taken to ensure demographic features didn't introduce discriminatory patterns.

4. Modeling

This phase involved designing and evaluating predictive models to classify the sentiment of skincare product reviews based on both textual and structured features.

4.1 Select Modeling Technique

The sentiment analysis task was approached using supervised learning algorithms. The selected models included:

- **Logistic Regression**: Chosen as a baseline due to its simplicity, speed, and interpretability.
- **Random Forest**: A robust ensemble method that captures non-linear feature interactions well.
- **XGBoost**: An advanced gradient boosting algorithm known for high accuracy and efficiency.
- **LinearSVC (Support Vector Classifier)**: Effective for high-dimensional spaces like TF-IDF vectors, with strong performance in text classification.

All models were selected for their established effectiveness in text classification and their complementary strengths in interpretability, scalability, and robustness.

4.2 Generate Test Design

A structured test design was used to ensure fair model comparison:

- **Train-Test Split**:
  - 80% training, 20% testing
  - Stratified to preserve class distribution in the target variable
- **Feature Engineering**:
  - Used TF-IDF Vectorizer on review text
  - Categorical features (skin_type, skin_tone) were one-hot encoded
  - Numerical features price_usd, rating) were standardized
  - Combined into a single preprocessing pipeline using ColumnTransformer
- **Class Imbalance Handling**:
  - SMOTE (Synthetic Minority Oversampling Technique) was applied on the training data after transformation to address imbalance
- **Model Tuning**:

  - GridSearchCV or RandomizedSearchCV was used for parameter optimization
  - Cross-validation (CV=3 or 5) ensured model generalization

4.3 Build Model

Each model was trained on the SMOTE-balanced, preprocessed dataset. Highlights include:

- **Logistic Regression**:
  - Trained with max_iter=1000
  - Served as a strong and interpretable baseline
  - Provided valuable insights into feature weights
- **Random Forest**:
  - Trained with standard hyperparameters
  - Utilized feature importance for insight
  - Demonstrated strong F1 performance
- **XGBoost**:
  - Tuned using RandomizedSearchCV
  - Key parameters: n_estimators = 100, learning_rate = 0.2 , max_depth =8
  - Delivered the **highest AUC** and F1-score across models
- **LinearSVC**:
  - Grid search over C and max_iter
  - Fast and effective for sparse TF-IDF matrices
  - Showed solid precision and accuracy

4.4 Assess Model

4.4.1 Evaluation Metrics

Models were evaluated using:

- **Accuracy** – Overall correctness
- **Precision & Recall** – Per-class performance
- **F1-Score** – Balance between precision and recall
- **Confusion Matrix** – Insight into classification errors
- **ROC-AUC** – Performance across all classification thresholds

4.4.2 Results Summary

Model	AUC	F1 Score	Strengths
XGBoost	0.99	~0.89	Best overall, excellent recall and precision
Logistic Reg.	~0.98	~0.84	Interpretable, fast baseline
Random Forest	~0.98	~0.85	Robust, handles interactions well
LinearSVC	~0.97	~0.85	Strong in high-dimensional text features

5. Evaluation

5.1 Model Evaluation Method

- Each classifier was trained on a SMOTE-resampled training set and evaluated on the same test set.
- Evaluation included classification report, confusion matrix, ROC_AUC_score.
- XGBoost outperformed other models with the best balance of recall and precision.

5.2 Interpretation and Insights

- **Textual Indicators**: Words such as “hydrating,” “gentle,” and “love” were strong positive signals. Terms like “burn,” “breakout,” and “drying” indicated negative sentiment.
- **Skin-Type Sensitivity**: Sentiment patterns varied by skin type, e.g., some products rated highly for dry skin were criticized by oily skin users.
- **Model Agreement**: All models showed consistent trends, validating the robustness of sentiment predictions.

5.2 Business Relevance and Deployment

Preferred Model: **XGBoost**

- Chosen for its high AUC and F1-score, combined with robust handling of diverse features
- Model and preprocessing pipeline saved via pickle
- Suitable for deployment in recommendation engines, dashboards, or automated review screening tools

6.Deployment

The sentiment classification model is deployed via a **Streamlit web app**. Users can input product reviews, which are vectorized using a saved TF-IDF vectorizer and classified by a pre-trained model. The app displays the predicted sentiment class instantly.

- **Tools**: Streamlit, Joblib, Python
- **Outputs**: Real-time sentiment prediction
- **Deployment**: Local or cloud-ready (Streamlit Cloud, Docker, AWS)