ROADMAP

Mining Sentiments: Analysing Customer Feedback with ML.

**1. Project Planning**

* **Define Objectives**:
  + Classify product reviews into positive, negative, or neutral sentiments.
  + Refine sentiment classification accuracy using multiple algorithms.
  + Group similar reviews using clustering for actionable insights.
* **Choose a Dataset**:
  + Example Datasets:
    - **IMDB Reviews Dataset** (available on Kaggle).
    - **Amazon Product Reviews** dataset.
    - **Yelp Reviews Dataset** for product or service-related feedback.

**2. Data Collection and Preparation**

* **Collect Data**:
  + Gather datasets containing text reviews with corresponding sentiment labels (positive, negative, neutral).
* **Data Pre-processing**:
  + Clean text data:
    - Remove punctuation, stop words, and special characters.
    - Convert text to lowercase.
    - Tokenize and lemmatize/stem words.
  + Create word vectors using:
    - **Bag of Words (BoW)**.
    - **TF-IDF (Term Frequency-Inverse Document Frequency)**.
  + Split Dataset:
    - Training Set: 70%.
    - Testing Set: 30%.

**3. Exploratory Data Analysis (EDA)**

* **Visualizations**:
  + Word clouds for most frequent positive and negative words.
  + Sentiment distribution using bar charts.
* **Key Insights**:
  + Identify patterns in text data (e.g., commonly used adjectives for positive/negative reviews).

**4. Model Development**

**Naive Bayes:**

* **Goal**: Classify reviews into positive, negative, or neutral sentiments quickly and effectively.
* **Implementation**:
  + Train the model using word vectors as features and sentiment labels as targets.
  + Evaluate using accuracy, precision, recall, and F1-score.

**Support Vector Machine (SVM):**

* **Goal**: Improve sentiment classification with better boundary separation.
* **Implementation**:
  + Train using TF-IDF features for refined classification.
  + Use **kernel functions** (linear or RBF) for enhanced performance.
  + Tune hyperparameters (C, gamma) using grid search.

**K-Means Clustering:**

* **Goal**: Group reviews into clusters based on similarity (e.g., highly positive, highly negative, or neutral).
* **Implementation**:
  + Apply clustering on TF-IDF vectors.
  + Evaluate clustering using **silhouette score**.
  + Label clusters manually or using heuristics if needed.

**5. Model Evaluation**

* **Naive Bayes & SVM**:
  + Use confusion matrix, accuracy, and classification reports for evaluation.
  + Plot ROC-AUC curve to compare model performances.
* **K-Means**:
  + Analyse cluster centroids to understand the themes of grouped reviews.
  + Visualize clusters using PCA or t-SNE for dimensionality reduction.

**6. Insights and Interpretations**

* **Sentiment Trends**:
  + Highlight the percentage of positive, negative, and neutral reviews.
* **Feature Importance**:
  + Identify words contributing most to positive and negative sentiments.
* **Cluster Insights**:
  + Group reviews based on similar themes (e.g., product quality, customer service).

**7. Deployment**

* Develop a simple tool or app for sentiment analysis:
  + Input: A product review text.
  + Output:
    - Sentiment classification (positive, negative, neutral).
    - Keywords driving the sentiment.
    - Associated cluster (e.g., quality, delivery, support).

**8. Documentation and Reporting**

* Create a detailed report including:
  + Problem statement and objectives.
  + Dataset details.
  + Data preprocessing steps.
  + Algorithm performances and comparisons.
  + Key insights and visualizations.

**Tools and Libraries**

* **Languages**: Python
* **Libraries**:
  + pandas, numpy for data manipulation.
  + nltk, spacy for text preprocessing.
  + sklearn for ML models and evaluation.
  + matplotlib, seaborn for visualizations.
  + flask or streamlit for deployment.