**Project Report**

**Name : Mrinal Ravindra Masne**

**Project Title : Emotion Detection in Text Using Machine Learning**

* **Problem state with business use case :**

Problem Statement

In the age of digital communication, vast amounts of text data are generated daily through social media, customer reviews, emails, and other online platforms. This text often contains valuable insights about human emotions that can be critical for businesses, researchers, and policymakers. However, manually analyzing and classifying these emotions is impractical due to the sheer volume of data. Therefore, there is a need for an automated system that can accurately classify emotions in text data using advanced machine learning and natural language processing (NLP) techniques.

Business Use Case

**Emotion Detection in Customer Reviews to Enhance User Experience**

Business Scenario:

A leading e-commerce company receives thousands of customer reviews daily. These reviews contain crucial feedback about products, services, and overall customer satisfaction. However, understanding the nuanced emotions expressed in these reviews is challenging due to their volume and the complexity of human language.

Solution:

Implement an automated emotion detection system that classifies the emotions expressed in customer reviews. This system will use state-of-the-art machine learning and NLP techniques to analyze text and categorize it into different emotion classes such as happiness, anger, sadness, surprise, and more.

Benefits:

* Enhanced Customer Support:

Identify and prioritize negative emotions (e.g., anger, frustration) to address issues promptly.

Improve customer satisfaction by responding to emotional feedback more effectively.

* Product Improvement:

Gather insights on how customers feel about specific products, leading to targeted improvements.

Identify common emotional themes related to product features to guide product development.

* Marketing Strategy:

Tailor marketing campaigns based on the emotional responses of customers.

Use positive emotions (e.g., happiness, excitement) expressed in reviews for promotional material.

* Brand Management:

Monitor brand sentiment and emotional trends over time to manage brand reputation.

Detect and mitigate potential PR crises by identifying spikes in negative emotions.

* Data-Driven Decisions:

Provide management with detailed emotion analysis reports to inform strategic decisions.

Enhance customer relationship management by integrating emotional insights.

Implementation:

1. Collect and preprocess customer reviews from the e-commerce platform.
2. Train a machine learning model using a labeled emotions dataset to classify text into predefined emotion categories.
3. Deploy the model to analyze incoming reviews in real-time and generate emotion-based reports and alerts.

* **Dataset Description:**

Columns

1. **Text:**

Description: This column contains the raw textual data, representing sentences or phrases that convey different emotions.

Data Type: String

Examples:

"I am so happy today!"

"I feel scared about the future."

"He makes me so angry."

1. **Label:**

Description: This column contains numerical values that correspond to different emotion categories. Each number represents a specific emotion as described below.

Data Type: Integer

Emotion Mapping:

0: Sadness

1: Joy

2: Love

3: Anger

4: Fear

5: Surprise

Examples:

0 (Sadness)

1 (Joy)

4 (Fear)

Emotion Categories :

Sadness (0):

Description: Represents feelings of sorrow or unhappiness.

Example: "I feel so down today."

Joy (1):

Description: Represents feelings of happiness and positivity.

Example: "I am so thrilled about the news!"

Love (2):

Description: Represents feelings of affection and care.

Example: "I love spending time with my family."

Anger (3):

Description: Represents feelings of displeasure and hostility.

Example: "I am furious with how things turned out."

Fear (4):

Description: Represents feelings of anxiety and apprehension.

Example: "I am terrified of speaking in public."

Surprise (5):

Description: Represents feelings of astonishment and amazement.

Example: "I was completely shocked by the unexpected visit."

* **Data Preprocessing :**

Preprocessing Steps

* Convert to Lowercase:

Description: All characters in the text are converted to lowercase to ensure uniformity and to avoid case sensitivity issues.

Example: "I am Happy" becomes "i am happy".

* Remove Links:

Description: Any URLs or links present in the text are removed to eliminate irrelevant information.

Example: "Check this out: http://example.com" becomes "Check this out: ".

* Remove Newline Characters:

Description: Newline characters are replaced with spaces to ensure the text is in a single line.

Example: "Hello\nWorld" becomes "Hello World".

* Remove Words Containing Numbers:

Description: Words that contain numbers are removed to discard non-textual information.

Example: "I have 2 dogs" becomes "I have dogs".

* Remove Extra Spaces:

Description: Multiple consecutive spaces are replaced with a single space to normalize the spacing in the text.

Example: "Hello World" becomes "Hello World".

* Remove Special Characters:

Description: All special characters and punctuation marks are removed to retain only alphabetic characters.

Example: "Hello, World!" becomes "Hello World".

* Tokenize the Text:

Description: The text is split into individual words (tokens) using tokenization.

Example: "Hello World" becomes ["Hello", "World"].

* Remove Stopwords:

Description: Commonly used words (stopwords) that do not contribute significant meaning are removed.

Example: "I am happy" becomes ["happy"].

* Stopwords List: A predefined list of stopwords from the NLTK library is used.
* Stemming and Lemmatization:

Stemming:

Description: Words are reduced to their root form using the Porter Stemmer to ensure uniformity.

Example: "running" becomes "run".

Lemmatization:

Description: Words are transformed into their base form using the WordNet Lemmatizer to ensure accurate representation of words.

Example: "running" becomes "run".

* **Modelling Approach :**

1. Data Splitting

To ensure the model's performance is generalizable to unseen data, we split the dataset into training and testing sets.

Training Set: Used to train the machine learning model.

Testing Set: Used to evaluate the model's performance on unseen data.

2. Feature Extraction

Since machine learning models require numerical input, we converted the text data into numerical features using the Term Frequency-Inverse Document Frequency (TF-IDF) method.

TF-IDF Vectorization: Converts the text into a matrix of TF-IDF features, which reflect the importance of words in the documents.

3. Model Selection

We experimented with several machine learning algorithms to identify the best-performing model for our classification task. The models considered included:

* Logistic Regression
* Support Vector Machine (SVM)
* Random Forest

4. Model Training

Each selected model was trained on the training dataset.

5. Model Evaluation

The trained models were evaluated on the testing dataset using various performance metrics to assess their accuracy and effectiveness in emotion classification.

Accuracy: The percentage of correctly classified instances.

Precision, Recall, F1-Score: These metrics provide a more detailed performance evaluation, especially for imbalanced datasets.

6. Final Model Selection

The model with the best performance on the evaluation metrics was selected as the final model for emotion classification. This model was then used to predict emotions in new, unseen text data.

* **Models Used:**

1. Logistic Regression

Description:

Logistic Regression is a linear model commonly used for binary classification problems. It estimates the probability that a given input belongs to a certain class and is particularly effective for problems where the relationship between the input features and the class labels is approximately linear.

Application:

We applied Logistic Regression to classify the text data into six emotion categories.

The model was trained using the TF-IDF features extracted from the text data.

2. Support Vector Machine (SVM)

Description:

SVM is a powerful classification algorithm that works well for both linear and non-linear data. It finds the hyperplane that best separates the data points of different classes by maximizing the margin between them. SVM is particularly effective in high-dimensional spaces.

Application:

We utilized SVM with a linear kernel for emotion classification.

The model was trained on the TF-IDF features to capture the relationship between words and emotions.

3. Random Forest

Description:

Random Forest is an ensemble learning method that combines multiple decision trees to improve classification accuracy. It reduces overfitting and improves generalization by averaging the results of individual trees.

Application:

We applied Random Forest to classify emotions in the text data.

The model was trained using the TF-IDF features.

* **Modelling Results :**

1. Logistic Regression

Accuracy:  82.46%

2. Support Vector Machine (SVM)

Accuracy: 82.18%

3. Random Forest

Accuracy: 85.09%

* **Conclusion**

Based on the evaluation metrics, Random Forest was identified as the best-performing model for emotion classification in text data. It not only achieved the highest accuracy but also provided robust and reliable predictions across all emotion categories. Therefore, we selected Random Forest as the final model for our emotion detection system. This model will be used in production to classify emotions from customer reviews, enhancing our ability to derive actionable insights and improve customer experience.