TWITTER SENTIMENT ANALYSIS

Presented By:

Team 1

M D Zoran Zeno

Mohammed Riyazullah

G Sai Charan

G Karthik Sai

AIML A

What is Sentiment Analysis

- **Sentiment Analysis** identifies emotions or opinions expressed in text.
- It classifies text as **Positive** or **Negative**
- Commonly used in social media monitoring, product reviews, and feedback systems.Write your agenda point

Problem Statement

- Analyze tweets to determine whether they convey a positive or negative sentiment.
- Use Logistic Regression, a simple yet powerful classification algorithm.
- Evaluate model performance using metrics like accuracy, precision, and recall.

Data Collection

- Dataset: Twitter Sentiment Analysis dataset on Kaggle.
- Contains tweets labeled as Positive (1) or Negative (0).
- Number of samples: ~1.6 million tweets.
- Data includes tweet text, user info, and sentiment label.

Data Preprocessing

- Remove special characters (keeping only A-Z and a-z).
- Convert all text to lowercase.
- Tokenize tweets into words (split).
- Remove stopwords (using NLTK) and perform stemming (using PorterStemmer).
- Convert cleaned text into numerical features using TF-IDF Vectorizer.

Why Logistic Regression?

- Excellent Baseline: Simple, fast, and efficient; provides a strong benchmark before using complex models.
- Highly Interpretable: Easy to understand relationships between features (words) and predictions.
- Ideal for Binary Classification: Statistically suited for Positive/Negative sentiment tasks.
- Efficient to Train: Works well on large datasets (like 1.6M tweets) without heavy computational needs.

Performance Metrics

Accuracy Score

- Logistic Regression: 78.35%
- Naive Bayes: 76.92%
- KNN : 63.19%

Precision

- Logistic Regression: 0.77
- Naive-Bayes: 0
- KNN: 0.58

Recall

- Logistic Regression: 0.80
- Naive-Bayes: 0
- KNN: 0.86

Confusion Matrices

		Logistic Regression	
		Predicted Values	
		Negative	Positive
True Values	Negative	122032	37633
	Positive	30644	128959

		Naïve-Bayes	
		Predicted Values	
		Negative	Positive
True Values	Negative	122778	0
	Positive	36785	0

		KNN	
		Predicted Values	
		Negative	Positive
True Values	Negative	63712	95953
	Positive	21539	138064

Results and Observations

- The Best Performing Model: **Logistic Regression**
- Least Accurate Model: **K Neighbors Classifier**
- Constant Predictor: Naive-Bayes, predicted every tweet as **Negative**
- Naive-Bayes had the lowest Precision and Recall because of **Model Collapse**. The model did not learn anything meaningful from the data. It had learned that the easiest way to get the right output is by predicting **Negative** every single time.
- This could be due to an issue in the cuML library's implementation of Naive-Bayes (it is different from the standard sklearn), since the preprocessing steps in our workflow were correct.