**Phase-3**

**predicting customer churn using machine learning to uncover hidden patterns**

**Student Name:** [Enter Your Name]

**Register Number:** [Enter Your Register Number]

**Institution:** [Insert College Name]

**Department:** [Enter Your Department Name]

**Date of Submission:** [Insert Date]

**Github Repository Link:** [Update the project source code to your Github Repository]

# 1. Problem Statement Social media platforms have become a hub of human emotions, opinions, and reactions to real-world events. However, the vast volume of this data is unstructured and hard to interpret without automation. This project aims to decode human emotions from social media conversations using sentiment analysis. It’s a text classification problem where the model categorizes user posts into emotional sentiments (e.g., happy, sad, angry, neutral). This can help businesses understand customer mood, gauge public opinion, and detect emotional trends in real-time.

Customer churn significantly impacts the profitability of telecom companies. Identifying potential churners in advance enables proactive customer retention strategies. The project aims to build a machine learning classification model that predicts whether a customer is likely to churn based on their usage behavior, demographics, and service history.

# 2. Abstract This project focuses on extracting and interpreting human emotions from social media conversations using Natural Language Processing (NLP). The goal is to develop a sentiment analysis system that classifies texts into emotional categories. The data is collected from public APIs like Twitter or Reddit, preprocessed using NLP techniques, and analyzed using machine learning models like Logistic Regression, SVM, or BERT. The project also includes a user-friendly deployment using Streamlit. The outcomes can aid in customer behavior analysis and brand monitoring for businesses.

This project addresses the issue of customer churn in the telecom sector using supervised machine learning. The primary objective is to analyze customer data and uncover patterns that indicate whether a customer is at risk of churning. The dataset was preprocessed to handle missing values and encode categorical data. Several classification algorithms were tested, with Random Forest achieving the best performance. Insights derived from the model help businesses reduce churn by targeting at-risk customers with retention campaigns.

# 3. System Requirements Hardware: Minimum 4 GB RAM Dual-core processor (i5 or equivalent) Software: Python 3.8+ Libraries: pandas, numpy, matplotlib, seaborn, sklearn, nltk, transformers IDE: Google Colab / Jupyter Notebook

 **Hardware:**

* RAM: Minimum 4 GB
* Processor: Intel i3 or higher

 **Software:**

* Python 3.8+
* Jupyter Notebook / Google Colab
* Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, streamlit (for deployment)

# 4. Objectives  Classify user sentiments into predefined emotion categories.  Visualize sentiment trends for better insights.  Deploy an interactive web application to allow users to input text and receive sentiment analysis.  Provide analytics that can support business decisions based on public emotional responses.

 Predict whether a customer will churn or not using historical data.

 Uncover behavioral patterns leading to churn.

 Evaluate and compare multiple classification algorithms.

 Deploy the best-performing model with a simple user interface.

**5. Flowchart of Project Workflow**

1

.Data

Collection

.Data

2

Cleaning &

Preprocessing

3

.Exploratory

Data Analysis

4

.Features

Engineering

5

.Model

Evalution

7

.Insights &

Visualization

8.Deployment

# 6. Dataset Description Source: Kaggle (e.g., Twitter US Airline Sentiment dataset) or via APIs like Tweepy Type: Public Structure: Typically includes fields like tweet ID, user, timestamp, text, sentiment label Preview: df.head()

 **Source:** Kaggle (Telco Customer Churn dataset)

 **Type:** Public

 **Size:** ~7043 rows × 21 columns

 **df.head():** (Insert screenshot of the first few rows)

# 7. Data Preprocessing Removed duplicates and null values Cleaned text (stop words, emojis, URLs) Used tokenization, lemmatization Encoded target labels Example: TF-IDF vectorization used for model input

 **Missing Values:** Handled using mode/mean replacement

 **Duplicates:** Removed using .drop\_duplicates()

 **Encoding:** LabelEncoder for categorical variables

 **Scaling:** StandardScaler for numerical features

# 8. Exploratory Data Analysis (EDA) Visuals: Bar plots of sentiment distribution, word clouds, heatmaps Key insights: More negative tweets than positive Frequent words linked to emotions Tools used: matplotlib, seaborn, wordcloud

 Visualizations: Histograms, heatmaps, pie charts, bar plots

 Key Insights:

* Senior citizens and customers with long tenure are less likely to churn.
* Monthly charges have a strong correlation with churn.

 (Insert screenshots of EDA plots)

# 9. Feature Engineering Text vectorization: TF-IDF, Word2Vec, BERT embeddings Feature selection: Top n-grams, POS tagging Features like sentiment polarity, tweet length used to improve accuracy

*  Created new feature: TotalChargesPerMonth = TotalCharges / tenure
*  Removed features with low importance
*  Used feature importance from Random Forest for selection
*  Applied standardization to improve model performance

# 10. Model Building Models tested: Logistic Regression, SVM, Random Forest, LSTM, BERT Best performing: BERT fine-tuned on sentiment dataset Training and validation logs documented

 Models Used:

* Logistic Regression (baseline)
* Decision Tree
* Random Forest (best accuracy)

 Chosen because:

* Logistic: Simple and interpretable
* Random Forest: Handles non-linear data and reduces overfitting

 (Insert training output screenshots)

# 11. Model Evaluation Metrics: Accuracy, Precision, Recall, F1-Score Visuals: Confusion matrix, ROC curve Example: BERT achieved 90%+ accuracy on validation set

 **Metrics:**

* Accuracy: 82%
* Precision, Recall, F1-Score reported

 **Visuals:**

* Confusion matrix
* ROC curve

 (Insert screenshots of confusion matrix and metrics)

# 12. Deployment Platform: Streamlit Cloud / Hugging Face Spaces URL: [Provide link here] Features: User input box Output: emotion classification Screenshot of UI included

 **Platform:** Streamlit Cloud

 **Deployment Method:** Python-based UI with real-time churn prediction

 **Public Link:** [Insert your Streamlit app link]

 **UI Screenshot:** (Insert screenshot)

 **Sample Prediction:** Input: Gender=Male, MonthlyCharges=75 → Output: Churn = Yes

**13. Source code**

# 14. Future scope Multi-lingual sentiment detection Real-time analysis on trending hashtags Integration with business dashboards (e.g., Power BI)

 Integrate real-time customer data streams for live churn prediction.

 Use deep learning models (e.g., LSTM) to improve sequential behavior prediction.

 Incorporate customer feedback and sentiment analysis from text reviews.

# 13. Team Members and Roles

| **Team Member Name** | **Role** |
| --- | --- |
| Bharath Kumar | Data Preprocessing & Cleaning |
| Sneha Reddy | Exploratory Data Analysis (EDA) & Visualization |
| Arjun Varma | Model Building & Evaluation |
| Priya Sharma | Deployment & Documentation |