Chat Sentiment Analysis using CatBoost

1. Introduction

With the rise of digital communication, chat-based platforms like WhatsApp, Instagram, customer service bots, and social media DMs have become vital sources of user sentiment. These chat texts are often short, spontaneous, filled with emojis, abbreviations, and non-standard grammar, making traditional sentiment analysis a challenge.

To overcome this, we focused on building a robust sentiment analysis system that could classify messages as **Positive**, **Negative**, or **Neutral**. After testing multiple algorithms, we chose **CatBoost**, a high-performance gradient boosting algorithm developed by Yandex. It excelled in both accuracy and adaptability for real-world, informal chat data.

---

2. Literature Review

Sentiment analysis has evolved through several generations of techniques:

- **Lexicon-Based Models**: VADER and SentiWordNet offered simple rule-based analysis using sentiment dictionaries but lacked context sensitivity.
- **Traditional ML Models**: Logistic Regression, Naive Bayes, and SVM used TF-IDF features but struggled with sarcasm, slang, and informal language.
- **Ensemble Models**: XGBoost, LightGBM, and Random Forest improved results using multiple learners but required more tuning and preprocessing.
- **Deep Learning Models**: LSTM, GRU, and BERT captured context and long-term dependencies but were resource-heavy and harder to interpret.

- **CatBoost**: Combines the best of both worlds—powerful boosting with native handling of text and categorical data, fast training, and great out-of-the-box performance.

---

3. Approaches to Sentiment Analysis

Approach	Context Aware	Feature Engineering	Accuracy	Example Models
Lexicon-Based	No	Manual	Low	VADER, SentiWordNet
Rule-Based	No	Manual	Low–Medium	Regex, Custom Rules
Traditional ML	Partially	Required	Medium	SVM, Logistic Regression
Ensemble Models	Yes	Minimal	High	CatBoost, XGBoost
Deep Learning	Yes	Minimal	Very High	LSTM, BERT, RoBERTa

---

4. Why CatBoost?

CatBoost stands out due to its balance of performance, flexibility, and efficiency. Here's a detailed breakdown:

4.1 Why This Algorithm?

- Delivered over **85% accuracy** on our dataset
- Performed well on **TF-IDF features** with **minimal preprocessing**
- Naturally handles categorical and text data without transformation
- Resistant to overfitting using **ordered boosting**

4.2 Why Not Others?

Algorithm	Limitation
Logistic Regression	Linear model; lower accuracy (~78%)
Naive Bayes	Too simplistic; misclassifies neutral sentiments
SVM	Resource-heavy and sensitive to high-dimensional vectors
XGBoost	Excellent but slower; more hyperparameter tuning needed
Random Forest	Decent but lacks performance with sparse chat-style features

4.3 Pros and Cons of CatBoost

Pros:

- Supports categorical and text inputs natively
- High accuracy out-of-the-box
- Minimal preprocessing required
- Handles missing values and noisy data well
- Built-in visualization and interpretability tools

Cons:

- Slightly slower than simpler models (e.g., Logistic Regression)
- Not as widely used as XGBoost in some communities

---

5. Methodology

5.1 Data Collection

- Chat messages labeled into **Positive (140)**, **Neutral (30)**, and **Negative (130)** categories

- Included emojis and casual phrases typical of modern conversations

5.2 Data Preprocessing

- Converted emojis to text using `emoji.demojize()`
- Removed punctuation and special symbols
- Applied **TF-IDF Vectorization** to convert text into numerical features

5.3 Model Development

- Used **CatBoostClassifier** from the CatBoost library
- Parameters: iterations = 300, depth = 6, learning_rate = 0.1
- Train-test split: 80:20
- Evaluated using accuracy, precision, recall, F1-score

---

6. Evaluation Metrics

Metric	Value (Average)
Accuracy	85.7%
Precision	86.3%
Recall	85.0%
F1-Score	85.6%

CatBoost showed excellent balance in handling all classes equally, which is crucial when the dataset includes subtle neutral tones and emoji-based sentiment.

---

7. Benefits of Using CatBoost

- Outperforms traditional ML on sparse and messy text

- Learns non-linear relationships in chat text
- Provides interpretable results through feature importance
- Works well with emojis and noisy tokens
- Quick implementation and minimal tuning

---

8. Deployment and Real-time Use

- The trained model can be deployed in real-time systems like chatbots or moderation platforms
- New messages can be cleaned and transformed using the same preprocessing pipeline
- Prediction is fast, making it suitable for real-time monitoring

---

9. Future Enhancements

- Upgrade from TF-IDF to **contextual embeddings** (e.g., BERT, USE)
- Add **emoji sentiment scores** as input features
- Use **SHAP** for detailed model interpretability
- Enable **multi-language support**
- Incorporate feedback loops for continuous learning

---

10. Conclusion

CatBoost offers a reliable and scalable approach to sentiment analysis, especially for informal, emoji-rich, and short texts like chats. With minimal preprocessing and strong accuracy, it stands out among both traditional and deep learning models. This makes it a top choice for real-world sentiment monitoring in modern communication systems.

In our project, CatBoost helped us balance performance, efficiency, and explainability—demonstrating how traditional ML techniques can still thrive when used smartly in the right context.