# Uber Reviews Sentiment Analysis Using Forecasting

**202418009[1], 202418011[2], 202418048[3]**

202418009@daiict.ac.in
202418011@daiict.ac.in
202418048@daiict.ac.in

## Project Definition

**Forecast Sentiment Based on Textual Review Content**

**What to Predict:** Predict future sentiment trends by analyzing the evolution of textual sentiment in the content column over time.

**Why It's Useful:** Textual sentiment can reveal nuanced user feelings (e.g., ”Very convenient” vs. ”Good”) that raw scores might miss, helping Uber understand specific pain points or strengths.

## Abstract

The goal of this research is to Forecast Sentiment Based on Textual Review Content. Objective of this model is to Predict future sentiment trends by analyzing the evolution of textual sentiment in the content column over time. Textual sentiment can reveal nuanced user feelings (e.g., ”Very convenient” vs. ”Good”) that raw scores might miss, helping Uber understand specific pain points or strengths. This paper uses Time series analysis with deep learning methods to predict the sentiment over time.

## Literature Survey

**Sentiment analysis: Sentiment Analysis: Review of Methods and Models**

Sentiment analysis, or opinion mining, is a natural language processing (NLP) subfield that identifies the emotional tone in text, classifying it as positive, negative, or neutral. It has become vital for analyzing user-generated content from social media, e-commerce, and reviews, aiding businesses, researchers, and policymakers in understanding public opinion. This review covers the evolution of sentiment classification methods, from rule-based to deep learning approaches.

### 1. Rule-Based and Lexicon-Based Methods

Early sentiment analysis used rule-based and lexicon-based methods, relying on predefined word lists to assign sentiment scores (e.g., ”happy” as positive, ”sad” as negative). Techniques like tokenization and negation handling improve accuracy, with VADER being a key example tailored for social media, including emoticons and slang. These methods are simple but struggle with context, sarcasm, and domain-specific terms.

### 2. Machine Learning-Based Methods

Machine learning (ML) transformed sentiment classification by learning from labeled data. Key algorithms include:

- Naive Bayes: A probabilistic model efficient for short texts like tweets, though less effective with nuanced sentiments.
- Support Vector Machines (SVM): Robust for text classification, especially with TF-IDF, excelling in review analysis.
- Logistic Regression: Simple and interpretable, it performs well on balanced datasets.
- K-Nearest Neighbors (KNN): Effective for small datasets but scales poorly with larger ones.

Feature engineering (e.g., bag-of-words, n-grams) is essential, though ML methods may miss semantic relationships and rely on data quality.

### 3. Deep Learning-Based Models

Deep learning advanced sentiment classification with neural networks capturing complex patterns. Key models include:

- Recurrent Neural Networks (RNNs): Suitable for sequential text, with LSTMs and GRUs improving long-term dependency capture, ideal for sentence-level analysis.
- Convolutional Neural Networks (CNNs): Adapted from image processing, they efficiently extract features from short texts like posts.
- Transformers and Attention Mechanisms: BERT and similar models use bidirectional context and attention to excel in tasks like aspect-based analysis, outperforming earlier methods on datasets like IMDB reviews.
- Pre-trained Language Models: RoBERTa, GPT, and XLNet, fine-tuned from large corpora, deliver top performance but demand significant computational power.

**Time Series Analysis:**

### 1. Traditional Time Series Models

Traditional models lay the groundwork for forecasting and have been adapted for sentiment data:

- Autoregressive Integrated Moving Average (ARIMA): ARIMA forecasts using past data and errors, applied to Weibo sentiment with LDA clustering to model sentiment shifts, effectively capturing trends.
- Exponential Smoothing (ETS): ETS prioritizes recent data, used with news sentiment to forecast stock prices, showing slight accuracy gains tied to immediate behavioral shifts.
- Seasonal Decomposition: Breaks time series into components, applied to Twitter sentiment to detect recurring opinion patterns, aiding behavioral cycle analysis.

These models are efficient and interpretable but falter with non-linear or complex sentiment-behavior dynamics.

### 2. Machine Learning-Based Models

Machine learning (ML) improves forecasting by handling non-linearities and multivariate data, ideal for sentiment:

- Support Vector Regression (SVR): SVR enhances stock price predictions using Twitter sentiment, modeling investor behavior.
- Random Forests: Forecasts cryptocurrency prices with social media sentiment, revealing collective emotional impacts on markets.
- Gradient Boosting: XGBoost, applied to e-commerce trends in Marketing 5.0, boosts consumer behavior predictions by combining sentiment and transaction data.

### 3. Deep Learning-Based Models

Deep learning transforms forecasting for sentiment and behavior with complex dependency modeling:

- Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM): LSTMs with news and Twitter sentiment predict stock volatility (e.g., -0.7 correlation) and cryptocurrency trends, capturing investor behavior.
- Convolutional Neural Networks (CNNs): CNNs with news sentiment improve short-term financial forecasts by detecting behavioral shifts like panic selling.
- Transformers: Outperform LSTMs in cryptocurrency forecasting using tweet sentiment, leveraging attention for volatile behavior prediction; BERT enhances aspect-based analysis.
- Generative Adversarial Networks (GANs): GANs generate synthetic data, showing potential for sentiment evolution modeling, though accuracy gains are dataset-specific.

## Gaps in Current Research

Sentiment analysis integrated with time series analysis has advanced forecasting in domains like finance and public opinion, yet significant gaps persist. Current methods struggle with contextual nuances like sarcasm, often relying on static lexicons or models that miss evolving language patterns, while temporal dependency modeling remains limited, with traditional approaches like ARIMA unable to capture non-linear sentiment shifts and even advanced deep learning models lacking intraday granularity. Multimodal integration is underdeveloped, ignoring non-textual cues like images or audio that could enrich time series predictions. Scalability is a challenge, with deep learning's computational demands clashing with real-time needs, and simpler models lacking sophistication.

## Links Used for This Research

- A Survey on Sentiment Analysis Methods, Applications, and Challenges - Springer: `https://link.springer.com/article/10.1007/s10462-022-10166-2`
- Sentiment Analysis Algorithms and Applications: A Survey - ScienceDirect: `https://www.sciencedirect.com/science/article/pii/S2090447914000550`
- A Comprehensive Survey on Sentiment Analysis Techniques - IJTech: `https://ijtech.eng.ui.ac.id/article/view/11111`
- A Multi-Method Survey on the Use of Sentiment Analysis in Multivariate Financial Time Series Forecasting - MDPI: `https://www.mdpi.com/2076-3417/13/3/1447`
- Time Series Forecasting Using Transformers with Sentiment Analysis on Financial Data - Archivo Digital UPM: `https://oa.upm.es/78566/`

## Dataset Selection

**Identification:**

**Uber Customer Reviews Dataset:** `https://www.kaggle.com/datasets/kanchana1990/uber-customer-reviews-dataset-2024`

**Contains:**

- Username: The reviewer's username is anonymized for privacy.
- Content: Captures the detailed review text where users share their experiences and opinions.
- Score: A numerical rating (1–5) indicating the user's satisfaction level with the product or service.
- ThumbsUpCount: Represents the number of likes or upvotes a review has received, showing how helpful it is to others.
- ReviewCreatedVersion: Specifies the app version when the review was written, helping track feedback across updates.
- At: A timestamp indicating the exact date and time when the review was posted.
- ReplyContent: Contains the developer's response to the review, addressing user concerns or feedback.
- RepliedAt: Records the date and time when the developer replied, useful for analyzing response times.
- AppVersion: The app version string associated with the review, helping track sentiment changes across different releases.

**Justification:**

This dataset contains over 12,000 customer reviews of the Uber app collected from the Google Play Store. The reviews provide insights into user experiences, including ratings, feedback on services, and developer responses. The data is cleaned and anonymized to ensure privacy compliance and ethical usage. It serves as a valuable resource for sentiment analysis, natural language processing (NLP) with time series.

**Preprocessing:**

- Text normalization: Convert all text to lowercase for consistency and remove extra spaces, special characters, emojis.
- Feature Extraction: Extract year, month, and day from At to analyze time-based trends.
- Remove Unnecessary Data: Exclude user images as they are not relevant to the analysis.
- Sentiment Analysis: Use NLP tools like VADER or TextBlob to extract sentiment from review text.
- Categorization: Optionally, classify reviews based on app version for deeper insights.