

Group: CSEMP144

Aspect-Based Sentiment Analysis of Hospital Reviews in Bhubaneswar

Synopsis of code:

This is a Streamlit-based web application that performs aspect-based sentiment analysis on hospital reviews in Bhubaneswar. Here's a breakdown of its main functionality:

1. Core Purpose:

- Analyzes hospital reviews by breaking them down into specific aspects (staff, cleanliness, wait time, facilities, cost, communication)
- Performs sentiment analysis using three different methods (TextBlob, VADER, and BERT)
- Visualizes the results through an interactive dashboard

2. Key Components:

- SentimentAnalyzer: Uses three different sentiment analysis models to classify text as Positive, Negative, or Neutral
- AspectDictionary: Maintains predefined keywords for different aspects of hospital service
- ReviewAnalyzer: Processes reviews to extract aspects and perform sentiment analysis
- AspectAnalyzer: Calculates metrics for each aspect across hospitals
- DashboardVisualizer: Creates various visualizations using matplotlib and seaborn

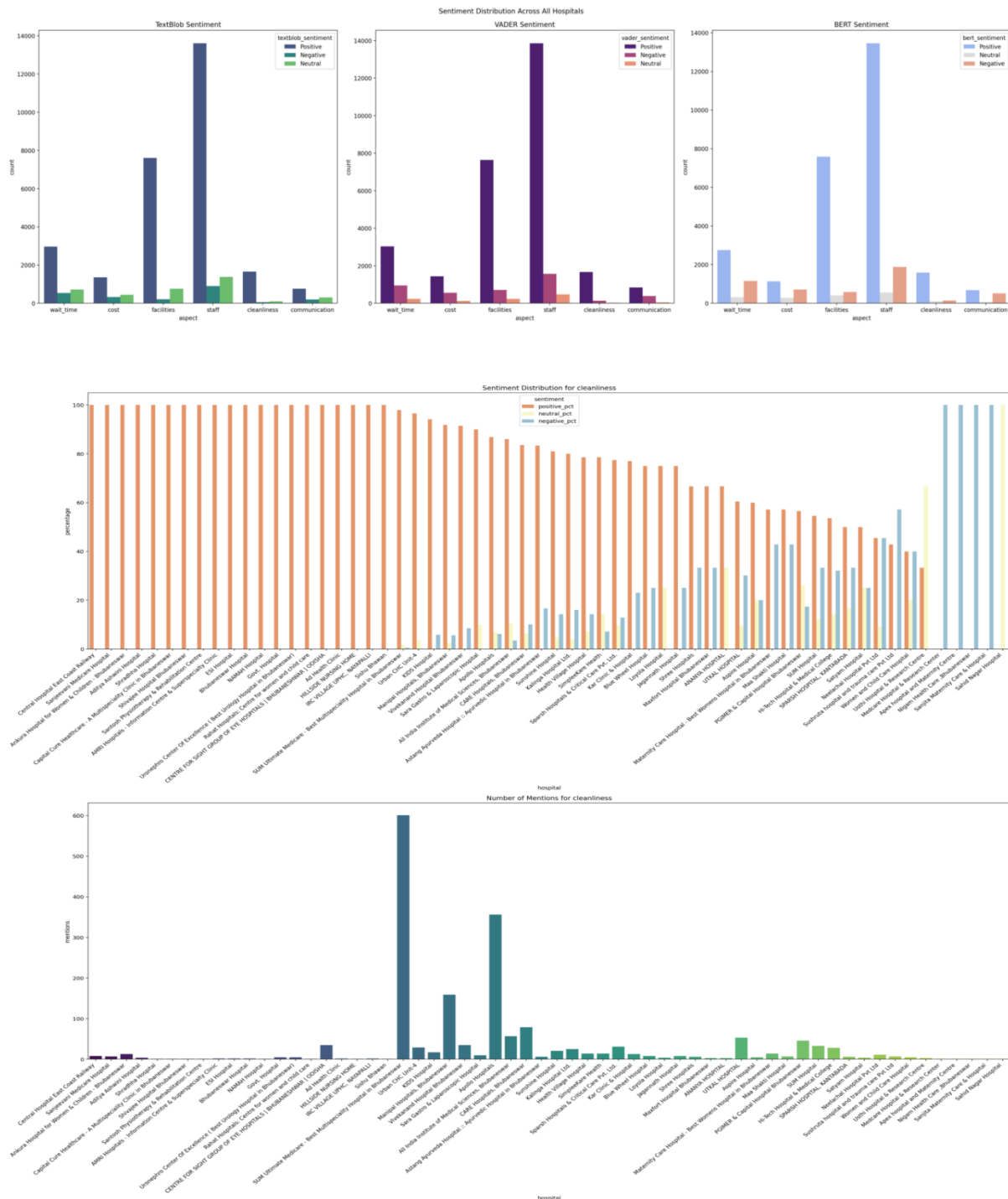
3. Workflow:

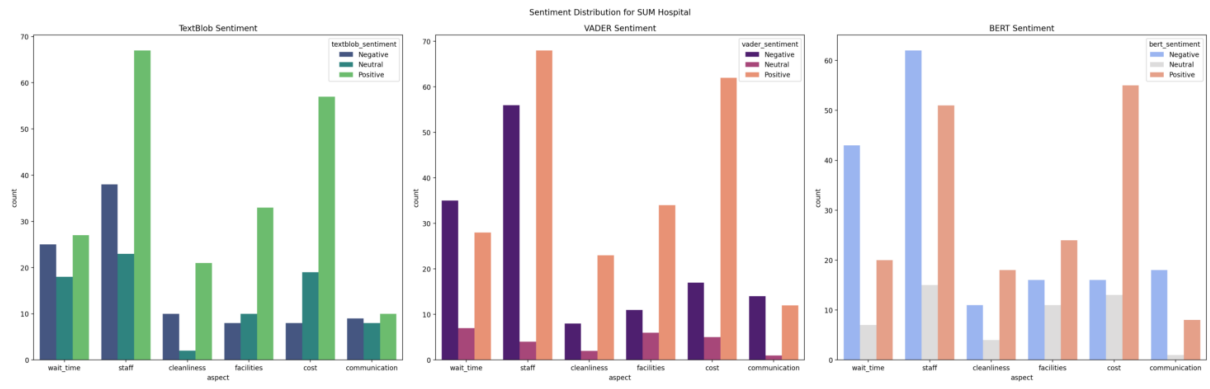
- Users upload a CSV file containing hospital reviews
- **The system processes each review to:**
 1. Identify mentioned aspects using keyword matching
 2. Analyze sentiment using all three models
- **Generates visualizations including:**
 1. Sentiment distribution across aspects
 2. Hospital-specific analysis
 3. Aspect-specific comparisons
 4. Detailed metrics tables

4. Technical Features:

- Uses data classes for structured data handling
- Implements caching and logging for better performance
- Provides progress bars during analysis
- Handles errors gracefully with try-except blocks
- Uses type hints for better code maintenance

Visualization/Figures:





Aspect-Based Sentiment Analysis of Hospital Reviews In Bhubaneswar

Upload your CSV file



Drag and drop file here
Limit 200MB per file • CSV

Browse files



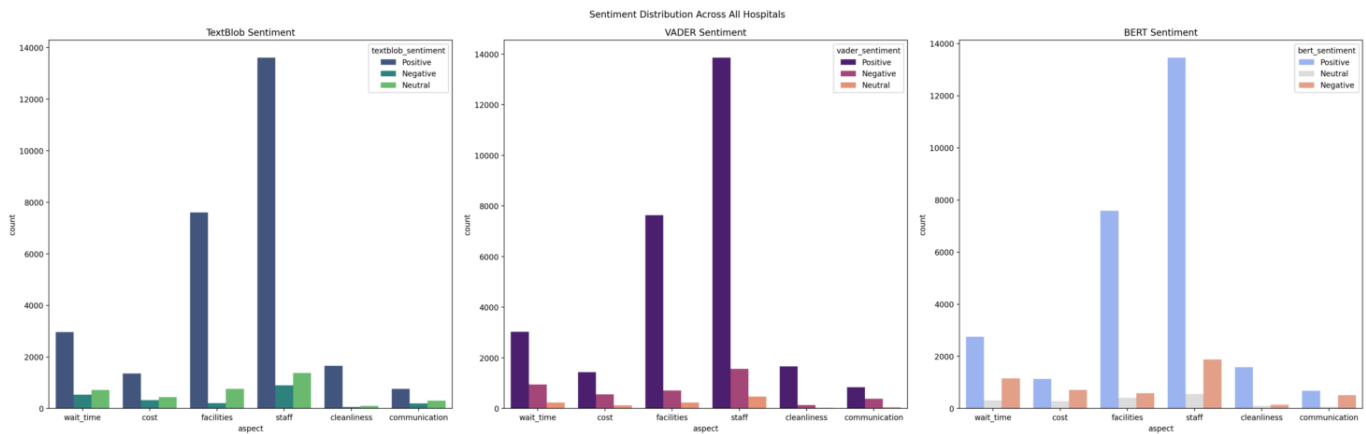
dataset_v2.csv 8.1MB



Data Sample:


	totalScore	text	title
0	4.6	Very bad service don't go their treatment	Sunshine H
1	4.6	Biswajit Mishra	Sunshine H
2	4.6	Don't go to this hospital. Extremely unprofessional staffs. Bad behavior with no sense	Sunshine H
3	4.6	Very Good behavior Doctors and staff ❤️ Giving Five stars for all the services started f	Sunshine H
4	4.6	Room size very small in compare to price. A little bit improvement needed in room se	Sunshine H

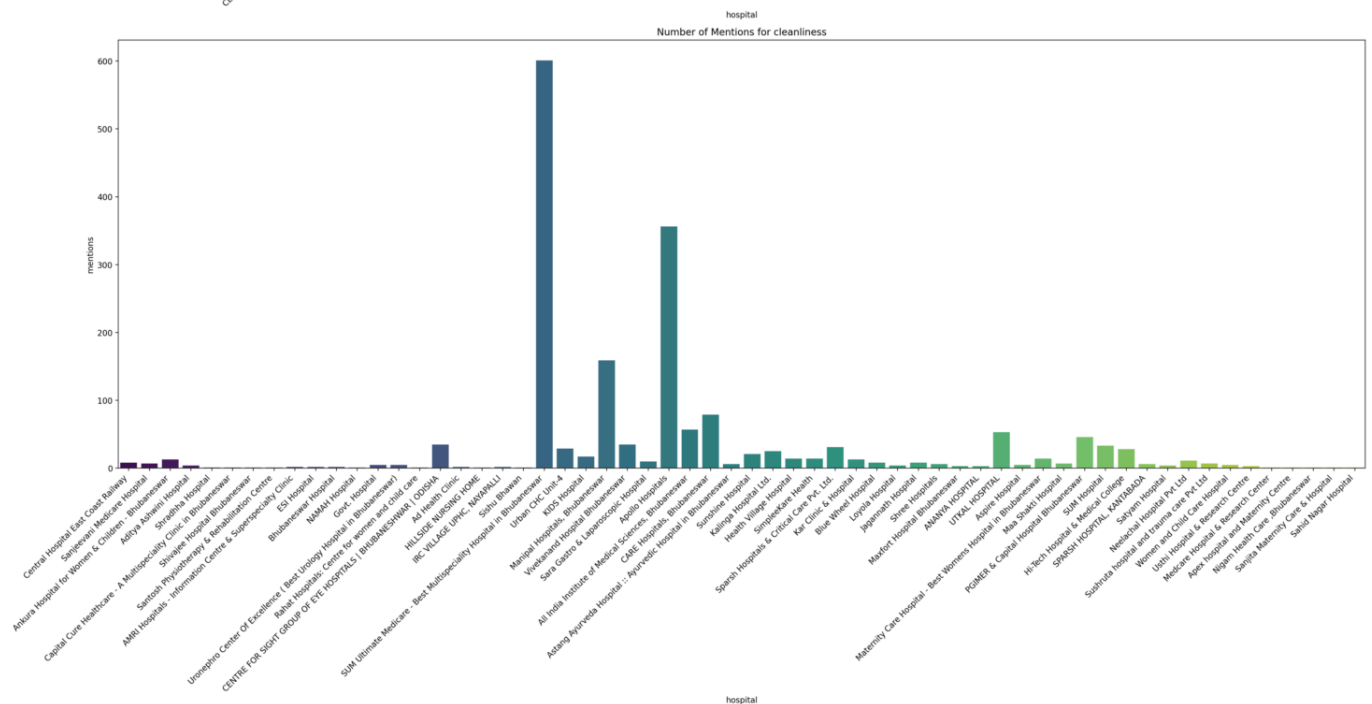
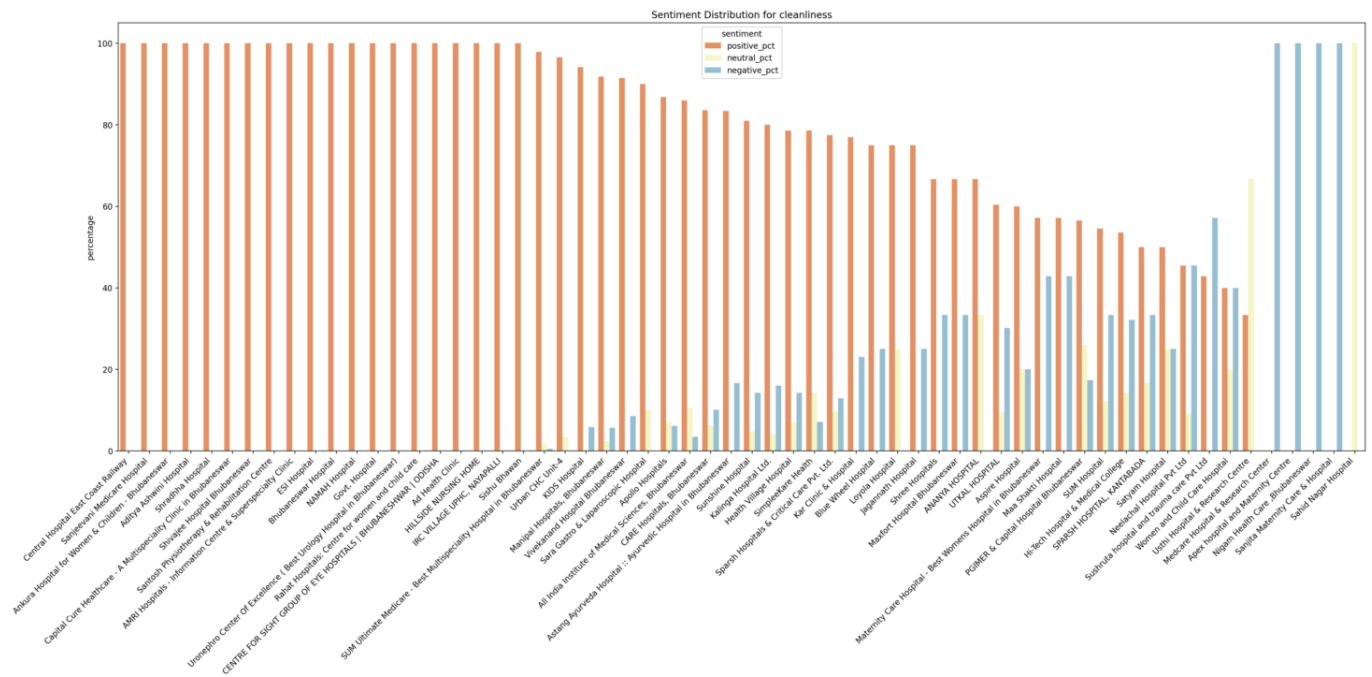
Overall Sentiment Distribution by Aspect



Aspect-Specific Analysis

cleanliness

Positive Sentiment % 



Detailed Metrics by Hospital and Aspect

	hospital	aspect	mentions	pe
140	Satyam Hospital	staff	9	44
317	Nigam Health Care ,Bhubaneswar	staff	9	44
40	CARE Hospitals, Bhubaneswar	wait_time	258	43
78	Usthi Hospital & Research Centre	staff	44	43
171	Sushruta hospital and trauma care Pvt Ltd	cleanliness	7	43
170	Sushruta hospital and trauma care Pvt Ltd	facilities	7	43
5	Sunshine Hospital	communication	21	43
124	Blue Wheel Hospital	wait_time	21	43
146	Hi-Tech Hospital & Medical College	staff	71	43
200	Aspire Hospital	staff	19	43
148	Hi-Tech Hospital & Medical College	wait time	31	43

Overall Hospital Ratings

	Hospital	Average Rating	Number of Reviews	Standard Deviation
0	100 Bedded MCH	4.4000	3	0.0000
1	ABHYANGA HEALTH	4.9000	19	0.0000
2	AIIMS Emergency	4.1000	21	0.0000
3	AMRI Hospitals - Information Centre & Superspecialty Clinic	3.9000	49	0.0000
4	ANANYA HOSPITAL	3.3000	155	0.0000
5	Aastha Hospital Health care service Opd Icu Ambulance Medical Emergency Diabetic Gastro Hospital Pathology	5.0000	5	0.0000
6	Ad Health Clinic	4.9000	27	0.0000
7	Aditya Ashwini Hospital	4.8000	521	0.0000
8	All India Institute of Medical Sciences, Bhubaneswar	4.3000	1028	0.0000
9	Ananta Jyot Eye Hospital	4.4000	6	0.0000
10	Ankura Hospital for Women & Children - Bhubaneswar	4.8000	493	0.0000
11	Apex hospital and Maternity Centre	2.9000	3	0.0000

	Hospital	Average Rating	Number of Reviews	Standard Deviation
12	Apollo Hospitals	4.7000	15960	0.0000
13	Aspire Hospital	3.9000	42	0.0000
14	Astang Ayurveda Hospital :: Ayurvedic Hospital in Bhubaneswar	4.2000	131	0.0000
15	BMC Dispensary	4.0000	1	<NA>
16	Baidyanath Memorial Hospital	3.9000	22	0.0000
17	Bhubaneswar Hospital	4.0000	17	0.0000
18	Bhubaneswar Hospital & Ultrasound Centre	4.8000	2	0.0000
19	Bhubaneswar Patia	2.0000	1	<NA>
20	Bhubaneswar Super speciality Clinic	4.6000	7	0.0000
21	Blue Wheel Hospital	4.4000	361	0.0000
22	CARE Hospitals, Bhubaneswar	4.3000	2191	0.0000
23	CENTRE FOR SIGHT GROUP OF EYE HOSPITALS BHUBANESHWAR ODISHA	4.9000	1017	0.0000
24	CGHS Hospital	3.8000	21	0.0000
25	Capital Cure Healthcare - A Multispeciality Clinic in Bhubaneswar	4.7000	33	0.0000
26	Capital Hospital CT Scan and X-RAY,BBSR	4.7000	2	0.0000
27	Central Hospital East Coast Railway	4.2000	53	0.0000
28	City Health Care & Medicines	5.0000	1	<NA>
29	Composite Hospital ..CRPF.. BBSR	4.5000	4	0.0000
30	Dr Sitikantha Nanda	5.0000	6	0.0000
31	Dr. Sharmili Sinha - Best Hospital in BHUBANESWAR	4.1000	1	<NA>
32	Dr.M.Jagatjit, Consultant Orthopaedics , Blue Wheel Hospital	5.0000	9	0.0000
33	ESI DISPENSARY HOSPITAL	5.0000	1	<NA>
34	ESI Hospital	3.9000	76	0.0000
35	Ebullience Clinics	4.4000	3	0.0000
36	Global Hospital Information Center	4.0000	1	<NA>
37	Govt. Hospital	4.4000	23	0.0000

	Hospital	Average Rating	Number of Reviews	Standard Deviation
38	HCG Panda Cancer DayCare Hospital	4.3000	26	0.0000
39	HILLSIDE NURSING HOME	4.0000	32	0.0000
40	Health Village Hospital	4.5000	375	0.0000
41	Hemalata Cancer Institute hospital	3.4000	3	0.0000
42	Hi-Tech Hospital & Medical College	3.7000	442	0.0000
43	Hospi Home's Baidyanath Hospital- Geriatric Multi Specialty Hospital	5.0000	17	0.0000
44	Hospital Bhubaneswar Main Govt	4.2000	2	0.0000
45	IRC VILLAGE UPHC, NAYAPALLI	4.6000	15	0.0000
46	Jagannath Hospital	3.4000	286	0.0000
47	KIDS Hospital	4.4000	365	0.0000
48	Kalinga Hospital Ltd.	3.9000	945	0.0000
49	Kar Clinic & Hospital	3.8000	502	0.0000
50	Loyola Hospital	3.9000	24	0.0000
51	MAMATA HOSPITAL	4.8000	2	0.0000
52	Maa Shakti Hospital	3.4000	144	0.0000
53	Makneel Hospital	5.0000	1	<NA>
54	Manipal Hospitals, Bhubaneswar	4.8000	9561	0.0000
55	Maternity Care Hospital - Best Womens Hospital in Bhubaneswar	3.6000	117	0.0000
56	Maxfort Hospital Bhubaneswar	3.4000	38	0.0000
57	Medcare Hospital & Research Center	3.7000	29	0.0000
58	Municipal Hospital	4.0200	10	0.3600
59	NAMAH Hospital	4.0000	21	0.0000
60	Narayani Hospital	5.0000	6	0.0000
61	Neelachal Hospital Pvt Ltd	4.2000	208	0.0000
62	Nephrology Department ,Blue Wheel Hospital	5.0000	2	0.0000
63	Nigam Health Care ,Bhubaneswar	4.0000	14	0.0000
64	Now MANIPAL. HOSPITALS	3.9000	6	0.0000

	Hospital	Average Rating	Number of Reviews	Standard Deviation
65	Odisha Multispeciality Hospital And Trauma Care Centre Private Limited	5.0000	1	<NA>
66	Omm Hospital, Patrapada, Bhubaneswar	5.0000	7	0.0000
67	PGIMER & Capital Hospital Bhubaneswar	4.0000	483	0.0000
68	Padma Hospital	2.9000	46	0.0000
69	Power Hospital	4.2000	17	0.0000
70	Rahat Hospitals: Centre for women and child care	4.9000	6	0.0000
71	Regional Govt. Hospital, Unit-VIII	4.5000	8	0.0000
72	S S M Hospital	3.5000	14	0.0000
73	SIMS HOSPITAL	3.4000	3	0.0000
74	SPARSH HOSPITAL, KANTABADA	4.1000	129	0.0000
75	SUM Hospital	3.7000	735	0.0000
76	SUM Ultimate Medicare - Best Multispeciality Hospital in Bhubaneswar	4.7000	6650	0.0000
77	Sahid Nagar Hospital	4.5000	6	0.0000
78	Sanjeevani Medicare Hospital	4.0000	130	0.0000
79	Sanjita Maternity Care & Hospital	2.8000	3	0.0000
80	Santosh Memorial hospital	2.3000	1	<NA>
81	Santosh Physiotherapy & Rehabilitation Centre	3.8000	43	0.0000
82	Sara Gastro & Laparoscopic Hospital	4.4000	62	0.0000
83	Satayu Hospital & Diabetic Research Centre	4.1000	3	0.0000
84	Satyam Hospital	3.9000	20	0.0000
85	Shivajee Hospital Bhubaneswar	4.6000	11	0.0000
86	Shradhha Hospital	4.5000	4	0.0000
87	Shree Hospitals	3.7000	404	0.0000
88	SimpleeKare Health	4.5000	276	0.0000
89	Sishu Bhawan	4.2000	107	0.0000
90	Sparsh Hospitals & Critical Care Pvt. Ltd.	4.2000	1126	0.0000
91	Sunflower Hospital	4.0000	3	0.0000

Code:

```
import pandas as pd
import nltk
from textblob import TextBlob
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from transformers import pipeline
import streamlit as st
import matplotlib.pyplot as plt
import seaborn as sns
from typing import List, Dict, Optional, Tuple
from dataclasses import dataclass
from pathlib import Path
import logging
from functools import lru_cache

# Configure logging
logging.basicConfig(
    level=logging.INFO, format="%(asctime)s - %(levelname)s - %(message)s"
)
logger = logging.getLogger(__name__)

@dataclass
class SentimentResult:
    """Data class to store sentiment analysis results"""

    textblob: str
    vader: str
    bert: str

class AspectDictionary:
    """Manages aspect-related keywords and operations"""

    ASPECTS = {
        "staff": [
            "staff",
            "doctor",
            "nurse",
            "receptionist",
            "physician",
            "specialist",
            "attendant",
            "caretaker",
            "surgeon",
        ],
    }
```

```
        "therapist",
        "clinician",
        "technician",
        "assistant",
        "medical team",
    ],
    "cleanliness": [
        "clean",
        "dirty",
        "hygiene",
        "sanitary",
        "sanitation",
        "neat",
        "messy",
        "filthy",
        "tidy",
        "spotless",
        "dusty",
        "orderly",
        "sterile",
        "disinfection",
        "germs",
    ],
    "wait_time": [
        "wait",
        "time",
        "delay",
        "long",
        "quick",
        "queue",
        "waiting",
        "hours",
        "minutes",
        "fast",
        "slow",
        "prompt",
        "timely",
        "lag",
        "late",
        "speed",
        "duration",
        "hold",
    ],
    "facilities": [
        "facility",
        "room",
        "equipment",
        "bed",
    ],

```

```
    "resources",
    "infrastructure",
    "building",
    "furniture",
    "amenities",
    "technology",
    "device",
    "tools",
    "environment",
    "setup",
    "labs",
    "cafeteria",
    "restroom",
    "parking",
    "accessibility",
],
"cost": [
    "cost",
    "price",
    "expensive",
    "affordable",
    "billing",
    "charges",
    "insurance",
    "payment",
    "fees",
    "overpriced",
    "inexpensive",
    "discount",
    "expense",
    "rates",
    "coverage",
    "deductible",
],
"communication": [
    "communication",
    "information",
    "explain",
    "informed",
    "clarity",
    "questions",
    "answers",
    "understand",
    "details",
    "update",
    "report",
    "feedback",
    "interaction",
```

```

        "discussion",
        "notify",
        "guidance",
        "instructions",
    ],
}

@classmethod
def get_aspects(cls) -> Dict[str, List[str]]:
    return cls.ASPECTS

class SentimentAnalyzer:
    """Handles multiple sentiment analysis methods"""

    def __init__(self):
        self.vader_analyzer = SentimentIntensityAnalyzer()
        self.bert_analyzer = pipeline(
            "sentiment-analysis",
            model="nlptown/bert-base-multilingual-uncased-sentiment",
        )
        self._download_nltk_data()

    @staticmethod
    def _download_nltk_data():
        """Download required NLTK data"""
        try:
            nltk.data.find("tokenizers/punkt")
            nltk.data.find("taggers/averaged_perceptron_tagger")
        except LookupError:
            nltk.download("punkt")
            nltk.download("averaged_perceptron_tagger")

    @staticmethod
    def _analyze_textblob(text: str) -> str:
        """Analyze sentiment using TextBlob"""
        polarity = TextBlob(text).sentiment.polarity
        return (
            "Positive"
            if polarity > 0.1
            else "Negative" if polarity < -0.1 else "Neutral"
        )

    def _analyze_vader(self, text: str) -> str:
        """Analyze sentiment using VADER"""
        scores = self.vader_analyzer.polarity_scores(text)
        return (
            "Positive"
            if scores["compound"] > 0.05

```

```

        else "Negative" if scores["compound"] < -0.05 else
"Neutral"
    )

    def _analyze_bert(self, text: str) -> str:
        """Analyze sentiment using BERT"""
        result = self.bert_analyzer(text[:512])[0]
        label = int(result["label"].split()[0])
        if label <= 2:
            return "Negative"
        elif label == 3:
            return "Neutral"
        return "Positive"

    def analyze_text(self, text: str) -> SentimentResult:
        """Analyze text using all sentiment analysis methods"""
        return SentimentResult(
            textblob=self._analyze_textblob(text),
            vader=self._analyze_vader(text),
            bert=self._analyze_bert(text),
        )

class ReviewAnalyzer:
    """Handles the analysis of hospital reviews"""

    def __init__(self):
        self.sentiment_analyzer = SentimentAnalyzer()
        self.aspect_dict = AspectDictionary.get_aspects()

    @staticmethod
    def load_data(filepath: Path) -> pd.DataFrame:
        """Load and preprocess the review data"""
        try:
            df = pd.read_csv(filepath)
            required_columns = ["title", "text", "totalScore"]

            if not all(col in df.columns for col in required_columns):
                raise ValueError(f"Missing required columns:
{required_columns}")

            df = df.dropna(subset=required_columns)
            return df

        except Exception as e:
            logger.error(f"Error loading data: {str(e)}")
            raise

    def extract_aspects(self, text: str) -> List[str]:

```

```

        """Extract aspects from review text"""
        aspects = set()
        tokens = nltk.word_tokenize(text.lower())

        for word in tokens:
            for aspect, keywords in self.aspect_dict.items():
                if word in keywords:
                    aspects.add(aspect)

        return list(aspects)

def analyze_reviews(self, df: pd.DataFrame) -> pd.DataFrame:
    """Analyze aspects and sentiments in reviews"""
    results = []
    total_rows = len(df)

    progress_bar = st.progress(0)

    for idx, (_, row) in enumerate(df.iterrows()):
        aspects = self.extract_aspects(row["text"])
        sentiments =
self.sentiment_analyzer.analyze_text(row["text"])

        for aspect in aspects:
            results.append(
                {
                    "hospital": row["title"],
                    "totalScore": row["totalScore"],
                    "aspect": aspect,
                    "textblob_sentiment": sentiments.textblob,
                    "vader_sentiment": sentiments.vader,
                    "bert_sentiment": sentiments.bert,
                }
            )

        progress_bar.progress((idx + 1) / total_rows)

    return pd.DataFrame(results)

class AspectAnalyzer:
    """Handles aspect-specific analysis and metrics"""

    @staticmethod
    def calculate_aspect_metrics(df: pd.DataFrame) -> pd.DataFrame:
        """Calculate various metrics for each aspect by hospital"""
        metrics = []

        for hospital in df["hospital"].unique():

```



```

        hospital_data = df[df["hospital"] == hospital]

        for aspect in hospital_data["aspect"].unique():
            aspect_data = hospital_data[hospital_data["aspect"] ==
aspect]

            # Calculate sentiment percentages
            total_mentions = len(aspect_data)
            positive_pct = (
                aspect_data["bert_sentiment"] == "Positive"
            ).mean() * 100
            negative_pct = (
                aspect_data["bert_sentiment"] == "Negative"
            ).mean() * 100
            neutral_pct = (aspect_data["bert_sentiment"] ==
"Neutral").mean() * 100

            # Calculate average score for this aspect
            avg_score = aspect_data["totalScore"].mean()

            metrics.append(
                {
                    "hospital": hospital,
                    "aspect": aspect,
                    "mentions": total_mentions,
                    "positive_pct": positive_pct,
                    "negative_pct": negative_pct,
                    "neutral_pct": neutral_pct,
                    "avg_score": avg_score,
                }
            )

        return pd.DataFrame(metrics)

class DashboardVisualizer:
    """Handles the visualization of analysis results"""

    @staticmethod
    def plot_sentiment_distribution(aspect_df: pd.DataFrame, title:
str) -> None:
        """Plot sentiment distribution across different models"""
        fig, axs = plt.subplots(1, 3, figsize=(25, 8))

        sns.countplot(
            data=aspect_df,
            x="aspect",
            hue="textblob_sentiment",
            ax=axs[0],

```

```

        palette="viridis",
    )
    axs[0].set_title("TextBlob Sentiment")

    sns.countplot(
        data=aspect_df,
        x="aspect",
        hue="vader_sentiment",
        ax=axs[1],
        palette="magma",
    )
    axs[1].set_title("VADER Sentiment")

    sns.countplot(
        data=aspect_df,
        x="aspect",
        hue="bert_sentiment",
        ax=axs[2],
        palette="coolwarm",
    )
    axs[2].set_title("BERT Sentiment")

    plt.suptitle(title)
    plt.tight_layout()
    st.pyplot(fig)

    @staticmethod
    def plot_aspect_comparison(
        metrics_df: pd.DataFrame, aspect: str, sort_by: str
    ) -> None:
        """Plot comparison of hospitals for a specific aspect"""
        aspect_data = metrics_df[metrics_df["aspect"] ==
aspect].sort_values(
            sort_by, ascending=False
        )

        fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(24, 24))

        # Plot sentiment distribution
        sentiment_data = pd.melt(
            aspect_data,
            id_vars=["hospital"],
            value_vars=["positive_pct", "neutral_pct", "negative_pct"],
            var_name="sentiment",
            value_name="percentage",
        )

        sns.barplot(

```

```

        data=sentiment_data,
        x="hospital",
        y="percentage",
        hue="sentiment",
        ax=ax1,
        palette="RdYlBu",
    )
    ax1.set_title(f"Sentiment Distribution for {aspect}")
    ax1.set_xticklabels(ax1.get_xticklabels(), rotation=45,
ha="right")

    # Plot mention counts
    sns.barplot(
        data=aspect_data, x="hospital", y="mentions", ax=ax2,
palette="viridis"
    )
    ax2.set_title(f"Number of Mentions for {aspect}")
    ax2.set_xticklabels(ax2.get_xticklabels(), rotation=45,
ha="right")

    plt.tight_layout()
    st.pyplot(fig)

def main():
    """Main application function"""
    st.title("Aspect-Based Sentiment Analysis of Hospital Reviews In
Bhubaneswar")

    uploaded_file = st.file_uploader("Upload your CSV file",
type="csv")

    if uploaded_file is None:
        st.info("Please upload a CSV file to begin analysis")
        return

    try:
        analyzer = ReviewAnalyzer()
        df = analyzer.load_data(uploaded_file)

        st.write("Data Sample:", df.head())

        with st.spinner("Analyzing reviews..."):
            aspect_df = analyzer.analyze_reviews(df)
            aspect_metrics =
AspectAnalyzer.calculate_aspect_metrics(aspect_df)

            visualizer = DashboardVisualizer()

```

```

# Overall sentiment distribution
st.subheader("Overall Sentiment Distribution by Aspect")
visualizer.plot_sentiment_distribution(
    aspect_df, "Sentiment Distribution Across All Hospitals"
)

# Aspect-specific analysis
st.subheader("Aspect-Specific Analysis")

col1, col2 = st.columns(2)

with col1:
    selected_aspect = st.selectbox(
        "Select an aspect to analyze",
        sorted(aspect_df["aspect"].unique())
    )

    with col2:
        sort_metric = st.selectbox(
            "Sort hospitals by",
            [
                "positive_pct",
                "negative_pct",
                "neutral_pct",
                "mentions",
                "avg_score",
            ],
            format_func=lambda x: {
                "positive_pct": "Positive Sentiment %",
                "negative_pct": "Negative Sentiment %",
                "neutral_pct": "Neutral Sentiment %",
                "mentions": "Number of Mentions",
                "avg_score": "Average Score",
            }[x],
        )

    visualizer.plot_aspect_comparison(aspect_metrics,
selected_aspect, sort_metric)

# Detailed metrics table
st.subheader("Detailed Metrics by Hospital and Aspect")

metrics_view = aspect_metrics.copy()
metrics_view["positive_pct"] = (
    metrics_view["positive_pct"].round(1).astype(str) + "%"
)
metrics_view["negative_pct"] = (
    metrics_view["negative_pct"].round(1).astype(str) + "%"
)

```

```

    )
    metrics_view["neutral_pct"] = (
        metrics_view["neutral_pct"].round(1).astype(str) + "%"
    )
    metrics_view["avg_score"] = metrics_view["avg_score"].round(2)

    st.dataframe(
        metrics_view.sort_values([sort_metric], ascending=False),
        use_container_width=True,
    )

    # Average ratings
    st.subheader("Overall Hospital Ratings")
    avg_rating = (
        df.groupby("title")["totalScore"]
        .agg(["mean", "count", "std"])
        .round(2)
        .reset_index()
    )
    avg_rating.columns = [
        "Hospital",
        "Average Rating",
        "Number of Reviews",
        "Standard Deviation",
    ]
    st.table(avg_rating)

    # Hospital-specific analysis
    st.subheader("Hospital-Specific Analysis")
    selected_hospital = st.selectbox("Select a hospital",
df["title"].unique())

    hospital_data = aspect_df[aspect_df["hospital"] ==
selected_hospital]
    visualizer.plot_sentiment_distribution(
        hospital_data, f"Sentiment Distribution for
{selected_hospital}")
    )

    except Exception as e:
        st.error(f"An error occurred: {str(e)}")
        logger.error(f"Application error: {str(e)}", exc_info=True)

if __name__ == "__main__":
    main()

```

