ChatGPT Reviews Analysis Report

Introduction

This report analyzes user reviews of ChatGPT, focusing on sentiment analysis, ratings distribution, keyword trends, and time-based rating fluctuations. The goal is to extract meaningful insights from user feedback.

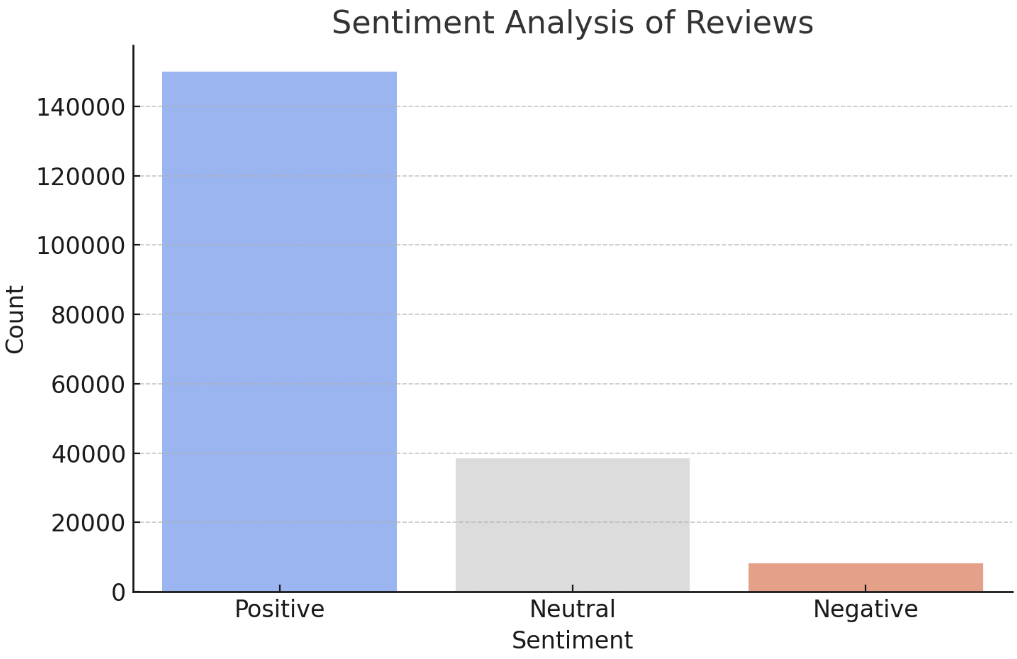
1. Sentiment Analysis

Sentiment analysis categorizes reviews as Positive, Neutral, or Negative based on text sentiment. Most reviews were positive, with a smaller proportion of neutral and negative sentiments.

```
# Sentiment Analysis using TextBlob
from textblob import TextBlob

def get_sentiment(text):
    polarity = TextBlob(text).sentiment.polarity
    if polarity > 0:
        return "Positive"
    elif polarity < 0:
        return "Negative"
    else:
        return "Neutral"

df["Sentiment"] = df["Review"].apply(get_sentiment)
```

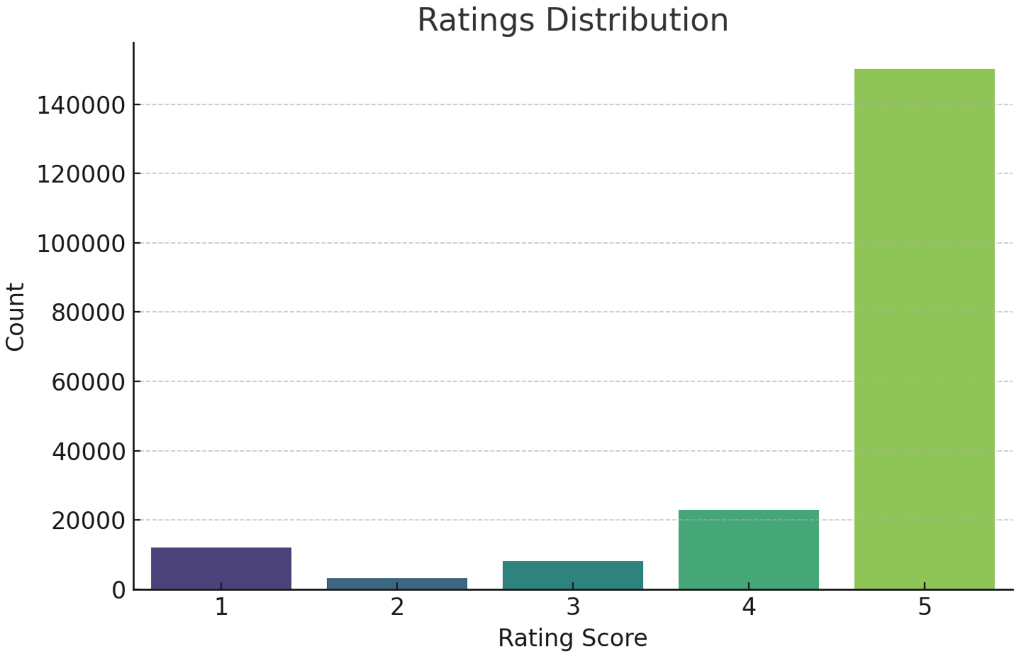


2. Ratings Distribution

The majority of users rated ChatGPT with 5 stars, indicating strong satisfaction. Fewer users provided lower ratings, suggesting a generally positive reception.

```
# Plot Ratings Distribution
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))
sns.countplot(data=df, x="Ratings", palette="viridis")
plt.title("Ratings Distribution")
plt.xlabel("Rating Score")
plt.ylabel("Count")
plt.show()
```



3. Keyword Analysis (Word Cloud)

A word cloud was generated to highlight the most commonly used words in reviews. Frequent words such as good, great, and app indicate positive feedback.

```
# Generate Word Cloud
from wordcloud import WordCloud

text = " ".join(df["Review"].dropna())
wordcloud        =        WordCloud(width=800,        height=400,        background_color="white",
colormap="coolwarm").generate(text)

plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

app
good
answer
helpful
use
love
best app
thank
best
good app
question
give
chat gpt
useful
help
chatgpt
amazing
nice app
great
need
work
thing
lot
google
great app
information
student
response
feature
everything
knowledge
time
know
much
used
nice
easy
awesome
working
think
really
cool
free
perfect
learning
well
understand
ask
find
study
app good
helpful app
one
now
make
even
provide
sometime
world
issue
good good
best ai
life
better
problem
will
using
try
keep
star
ai
far
still
say
super
bad
application
update
word
day
amazing app
good nice
anything
future
alway
voice
people
error
download
tool
assignment
wonderful
way
ai app
homework
conversation
fast
option
studies
ok
text
friend
useful app
made
experience
writing
gpt

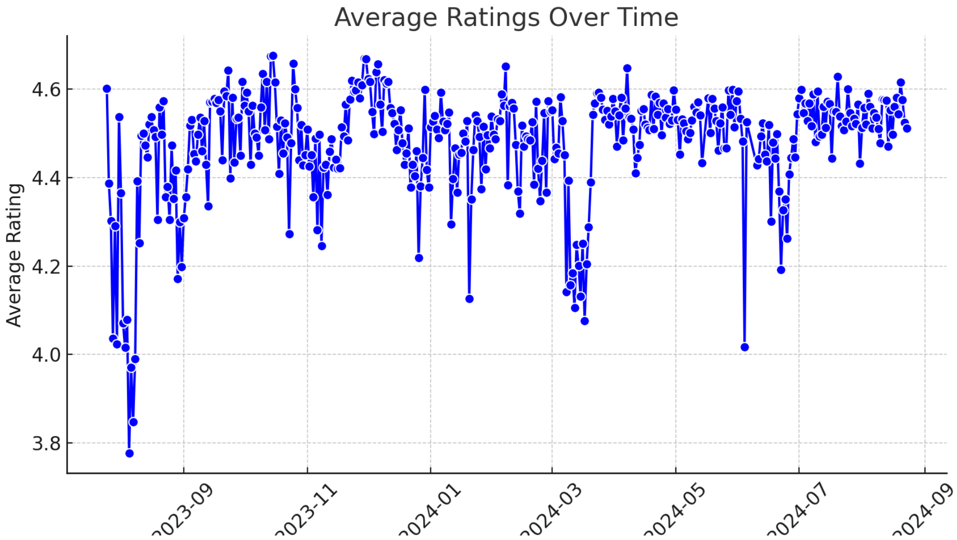
4. Time-Based Trends

The trend analysis tracks average ratings over time. Ratings remain generally stable, but fluctuations might reflect user experience variations.

```
# Convert Review Date to datetime format
df["Review Date"] = pd.to_datetime(df["Review Date"])

# Group by date and calculate average rating
ratings_over_time = df.groupby(df["Review Date"].dt.date)["Ratings"].mean()

# Plot Ratings Over Time
plt.figure(figsize=(10, 5))
sns.lineplot(x=ratings_over_time.index, y=ratings_over_time.values, marker="o", color="blue")
plt.title("Average Ratings Over Time")
plt.xlabel("Date")
plt.ylabel("Average Rating")
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```


Average Ratings Over Time
Average Rating