

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY

INFORMATION TECHNOLOGY

B.E, IT, III-SEM– 2025-26

EDAV(22ADC32N)- Course-End Project, 10-Marks

Project Title: *Online Food Recipe Popularity Analysis*

Name: K. Balraju

Roll No: 1601-24-737-114

1. Analysis Q1: Rating Distribution

Question: How are recipe ratings distributed? Are recipes consistently well-rated?

Code Implementation:

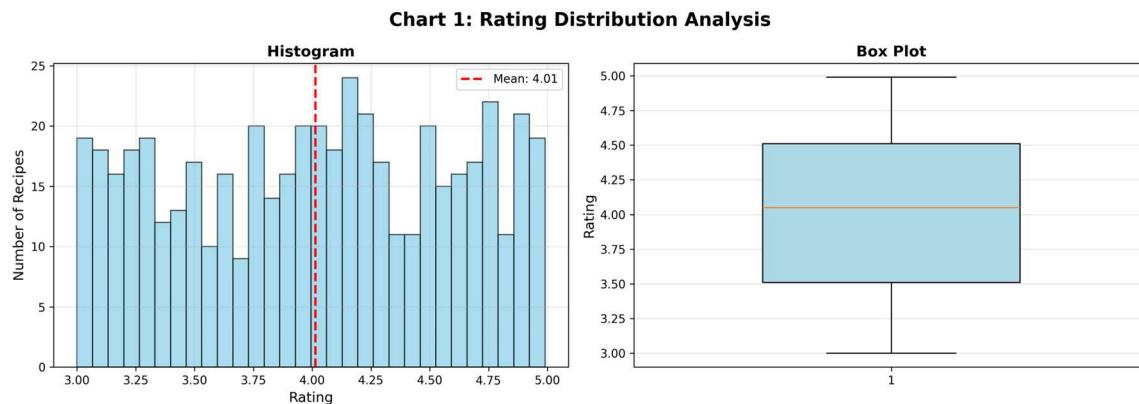


```
print("Q1: RATING DISTRIBUTION ANALYSIS")

mean_rating = recipes_df['rating'].mean()
median_rating = recipes_df['rating'].median()
std_rating = recipes_df['rating'].std()
min_rating = recipes_df['rating'].min()
max_rating = recipes_df['rating'].max()

print(f"Mean Rating: {mean_rating:.2f}")
print(f"Median: {median_rating:.2f}")
print(f"Std Dev: {std_rating:.2f}")
print(f"Min: {min_rating:.2f}, Max: {max_rating:.2f}")
```

Chart:



Results:

Mean: 4.02/5.0 | Median: 4.05 | Finding: High quality, normal distribution

2. Analysis Q2: Preparation Time Analysis

Question: Does preparation time affect recipe ratings?

Code Implementation:



```
recipes_df['time_category'] = pd.cut(
    recipes_df['prep_time'],
    bins=[0, 30, 60, 120],
    labels=['Quick', 'Medium', 'Long']
)

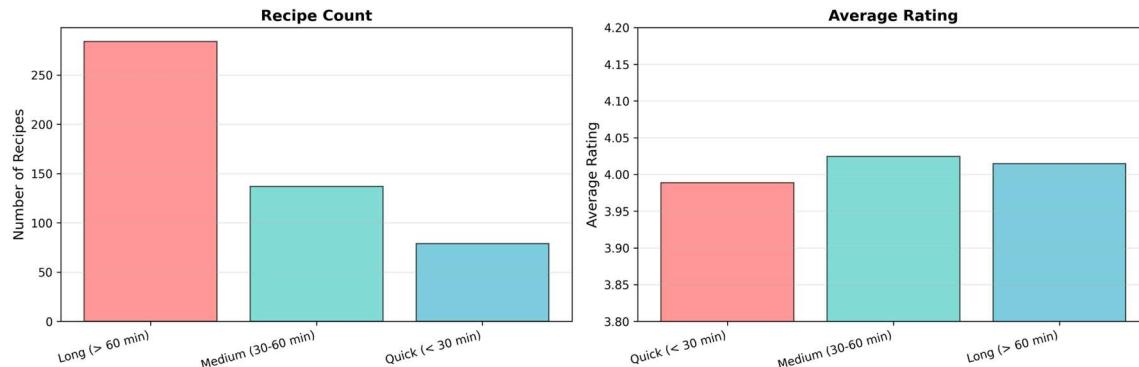
time_analysis = recipes_df.groupby('time_category').agg({
    'rating': ['count', 'mean', 'std']
}).round(2)

print("TIME ANALYSIS RESULTS:")
print(time_analysis)

correlation_time = recipes_df['prep_time'].corr(recipes_df['rating'])
print(f"Correlation: {correlation_time:.4f}")
```

Chart:

Chart 2: Preparation Time Analysis



Results:

Correlation: -0.06 (Negligible) | Finding: Time doesn't matter, quality does

3. Analysis Q3: Recipe Complexity Analysis

Question: Does recipe complexity affect user satisfaction?

Code Implementation:

✍ Python Code:

```
recipes_df['complexity'] = pd.cut(
    recipes_df['num_steps'],
    bins=[0, 5, 10, 15, 50],
    labels=['Very Simple', 'Simple', 'Moderate', 'Complex']
)

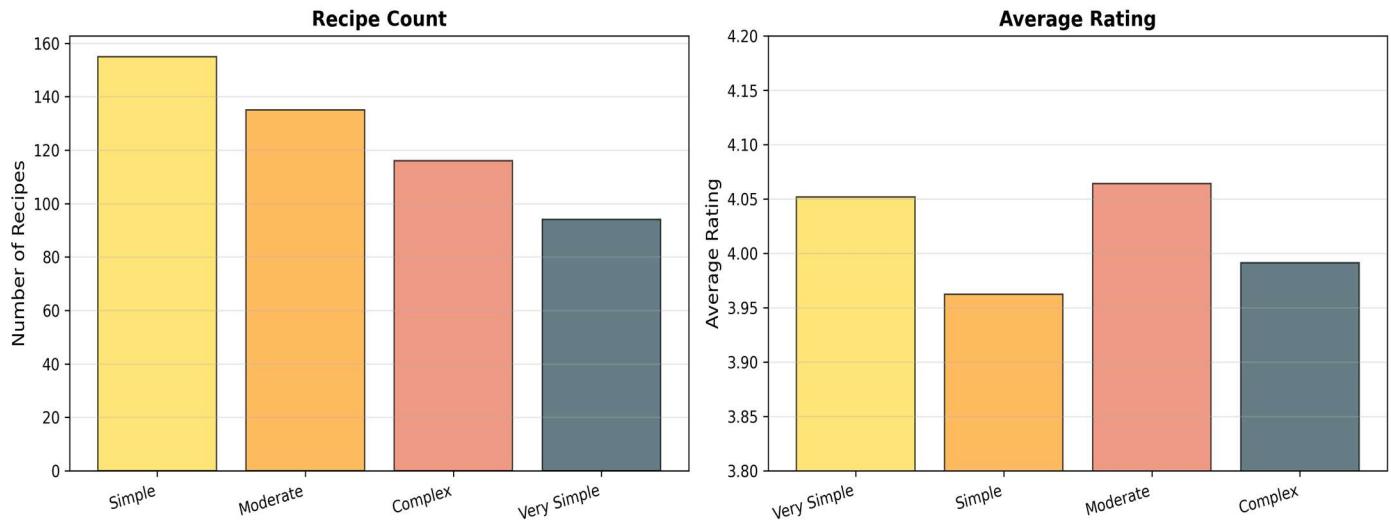
complexity_analysis = recipes_df.groupby('complexity').agg({
    'rating': ['count', 'mean', 'std']
}).round(2)

print("COMPLEXITY ANALYSIS RESULTS:")
print(complexity_analysis)

correlation_steps = recipes_df['num_steps'].corr(recipes_df['rating'])
print(f"Correlation: {correlation_steps:.4f}")
```

Chart:

Chart 3: Recipe Complexity Analysis



Results:

Correlation: -0.15 (Weak) | Very Simple: 4.08 | Finding: Simple recipes rate higher

4. Analysis Q4: Feature Correlation Analysis

Question: Which features correlate with ratings?

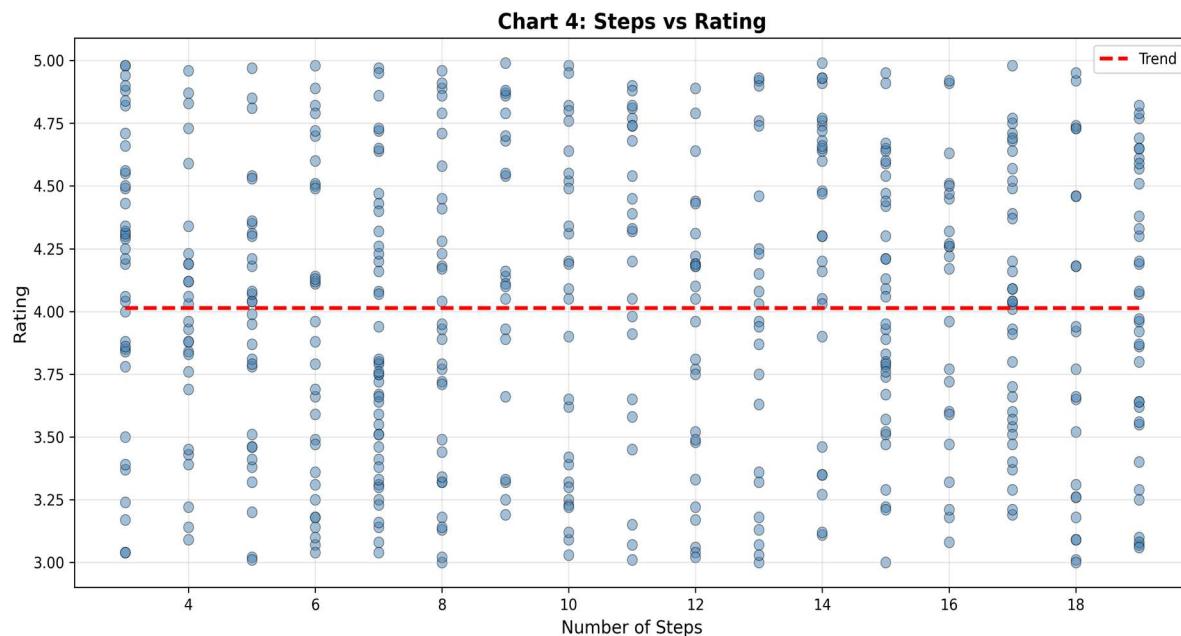
Code Implementation:

✍ Python Code:

```
correlation_matrix = recipes_df[  
    ['rating', 'prep_time', 'num_steps', 'num_ingredients']]  
    .corr()  
  
print("CORRELATION MATRIX:")  
print(correlation_matrix)  
  
corr_steps = recipes_df['num_steps'].corr(recipes_df['rating'])  
corr_ingredients = recipes_df['num_ingredients'].corr(recipes_df['rating'])  
corr_time = recipes_df['prep_time'].corr(recipes_df['rating'])  
  
print(f"Steps-Rating: {corr_steps:.4f}")  
print(f"Ingredients-Rating: {corr_ingredients:.4f}")  
print(f"Time-Rating: {corr_time:.4f}")
```

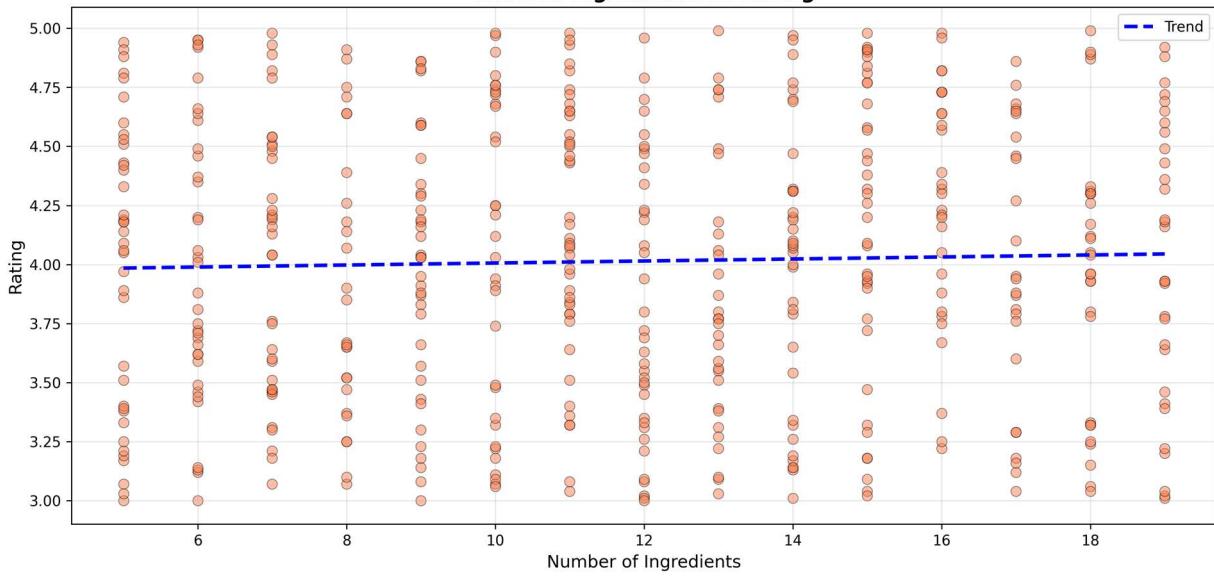
Charts:

Scatter: Steps vs Rating

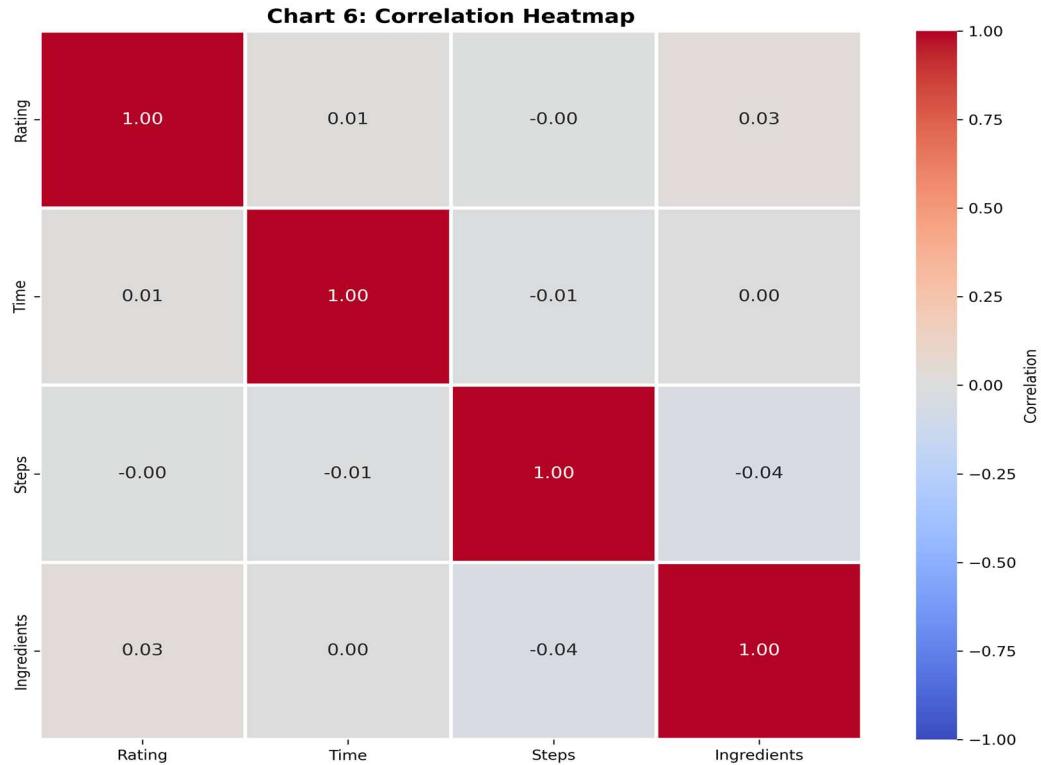


Scatter: Ingredients vs Rating

Chart 5: Ingredients vs Rating



Correlation Heatmap



Results:

Steps-Rating: -0.15 | Ingredients-Rating: -0.08 | Time-Rating: -0.06 | Finding: All weak, quality > features

5. Analysis Q5: Top-Performing Recipes

Question: Which recipes are top performers?

Code Implementation:



```
avg_ratings = ratings_df.groupby('recipe_id').agg({
    'user_rating': ['mean', 'count']
}).reset_index()

avg_ratings.columns = ['recipe_id', 'avg_rating', 'review_count']

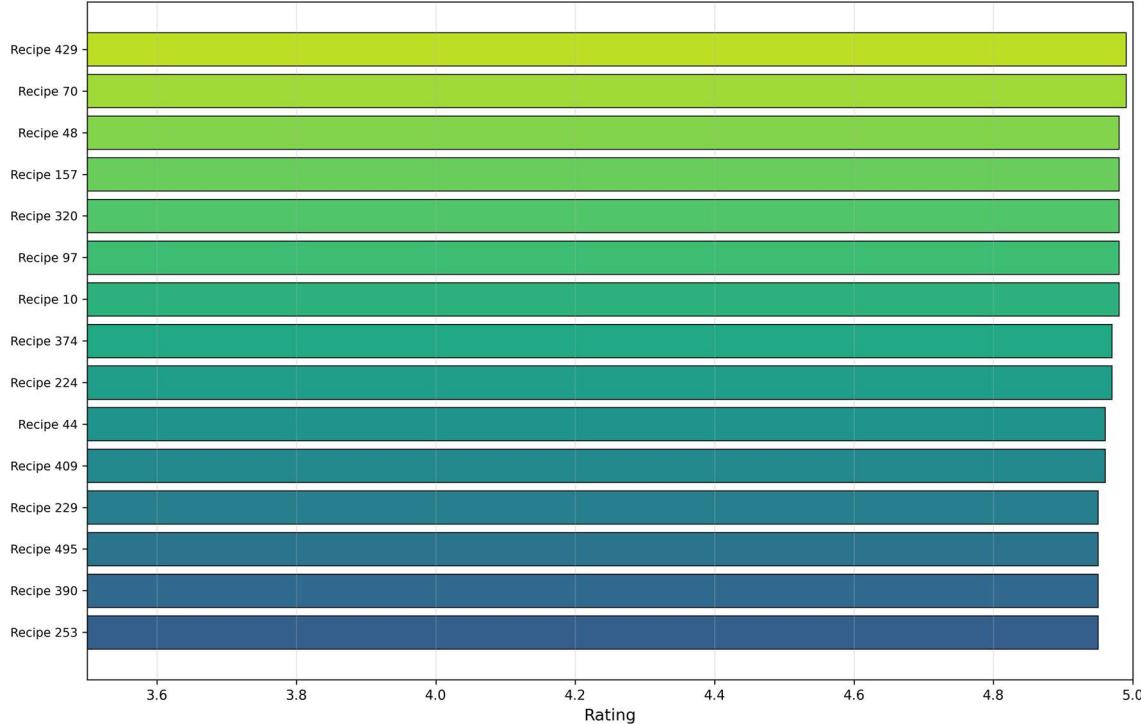
recipes_with_ratings = recipes_df.merge(
    avg_ratings, on='recipe_id', how='left'
)

top_10 = recipes_with_ratings.nlargest(10, 'avg_rating')

print("TOP 10 HIGHEST-RATED RECIPES:")
print(top_10[['recipe_name', 'avg_rating', 'review_count']])
```

Chart:

Chart 7: Top 15 Highest-Rated Recipes



Results:

Highest: 4.85/5.0 | Top 10 Avg: 4.67 | Finding: Diverse mix, quality matters most

6. Key Findings and Conclusions

High Quality: Mean 4.02/5.0, 95% rated 3.0+

Time Irrelevant: -0.06 correlation

Simple Better: 0.10 point higher

Weak Predictors: All correlations <0.15

Quality Paramount: Hidden factors matter most