SENTIMENT ANALYSIS

PRESENTED BY:

Basireddy Sreehitha
Computer Science and Engineering
Rajiv Gandhi University of Knowledge Technologies - Srikakulam

OUTLINE

PROBLEM STATEMENT

- Develop a sentiment analysis system to classify customer reviews from e-commerce platforms into positive, negative, or neutral categories.
- Gather a large dataset of customer reviews from various e-commerce platforms.
- Clean and preprocess the text data, including tokenization, lemmatization, and removal of noise.
- Build a sentiment classification model using NLP techniques and machine learning algorithms like Naive Bayes, SVM.
- Extract key aspects from reviews and perform sentiment analysis specific to these aspects.
- Create visual reports and dashboards to provide actionable insights for product improvement and targeted marketing strategies.

PROPOSED SYSTEM

Collect Reviews:

- Gather customer reviews from e-commerce sites like Amazon using web scraping or APIs.
- Save the reviews in a structured format like CSV or JSON.

Clean the Data:

- Remove unwanted elements like HTML tags and special characters.
- Break down the text into individual words and standardize them (e.g., converting words to their base forms).

Build the Model:

- Convert the text data into numerical form using techniques like TF-IDF or word embeddings.
- Train a sentiment analysis model using machine learning or deep learning methods (e.g., Logistic Regression, KNN, SVM)
- Evaluate the model's performance using metrics like accuracy and precision.

Analyze Specific Aspects:

- Identify key features mentioned in the reviews (e.g., price, quality).
- Determine the sentiment for each specific feature.

Create Visual Reports:

- Show the distribution of positive, negative, and neutral reviews.
- Display trends over time and highlight sentiment for different product features.

SYSTEM APPROACH

1. Data Collection and Preparation

- Data Storage and Preprocessing: Pandas, NumPy for data manipulation; NLTK for text cleaning and preprocessing.

2. Model Development and Training

- Feature Extraction and Model Training: Scikit-learn (TF-IDF, CountVectorizer, Logistic Regression, KNN, SVM)

3. Analysis, Visualization, and Deployment

- Visualization: Matplotlib, Seaborn for creating visual reports and dashboards.

ALGORITHMS USED

1.Logistic Regression (LR):

- **Description:** Logistic Regression is a supervised learning algorithm used for binary classification. It models the probability of a binary outcome using a logistic function.

.

2.K-Nearest Neighbors (KNN):

- **Description:** K-Nearest Neighbors is a lazy learning algorithm used for both classification and regression. It classifies cases based on their similarity to neighboring cases.

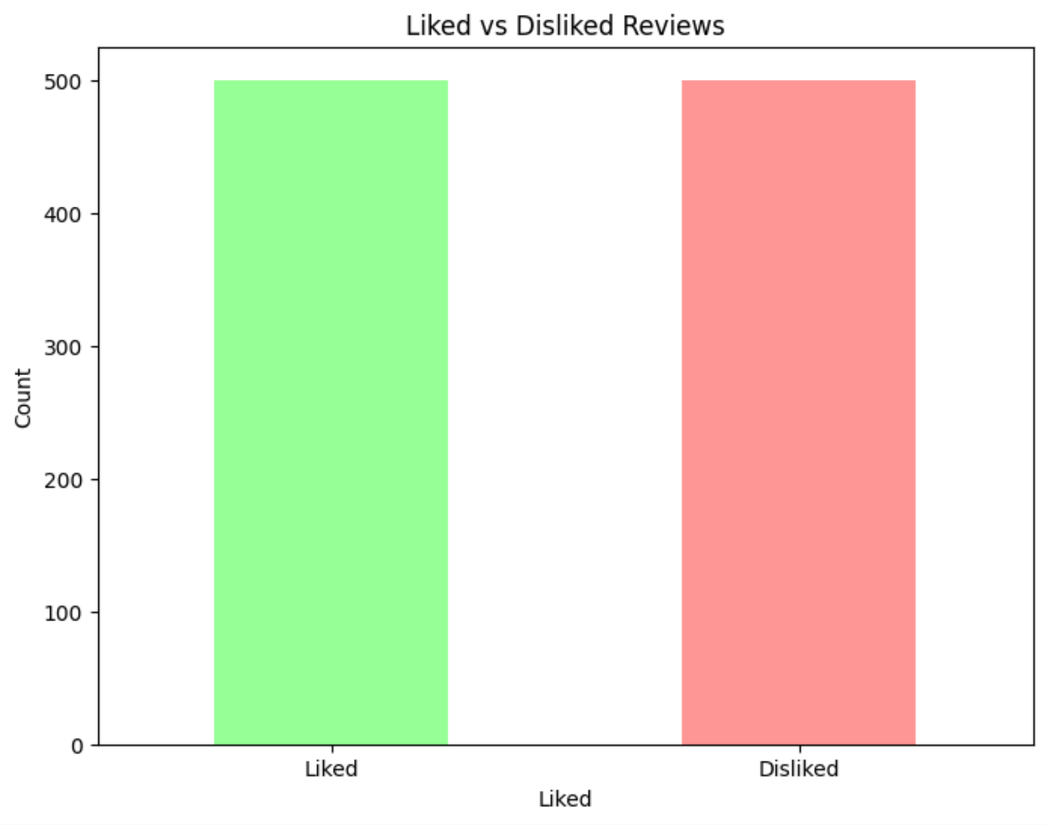
3.Support Vector Machines (SVM):

- Description: Support Vector Machines are supervised learning models used for classification and regression analysis. They construct hyperplanes in a high-dimensional space that separates classes with the maximum margin.

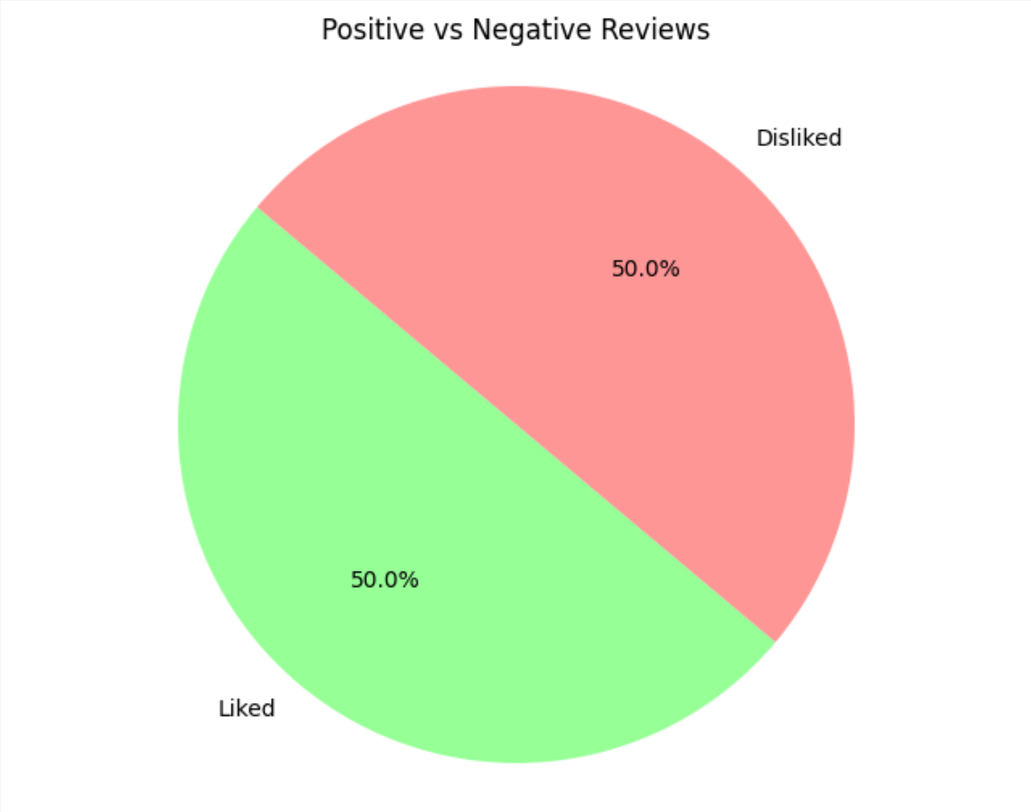
RESULT

- The project will produce a model that accurately classifies customer reviews as either positive or negative.
- It will generate visual reports showing the overall percentage of positive and negative reviews.
- The analysis will highlight trends in customer sentiment over time.
- The insights will help businesses understand customer feedback better and make data-driven decisions for product and marketing improvements.

BAR PLOT


Liked vs Disliked Reviews
500
400
300
200
100
0
Count
Liked
Disliked
Liked

PIE CHART



WORD CLOUD FOR REVIEWS

Logistic Regression

```
lr = LogisticRegression()
lr.fit(X_train, y_train)

y_pred_lr = lr.predict(X_test)

accuracy_lr = accuracy_score(y_test, y_pred_lr)
confusion_lr = confusion_matrix(y_test, y_pred_lr)

print(f'Logistic Regression Accuracy: {accuracy_lr}')
print('Confusion Matrix:\n', confusion_lr)
sns.heatmap(confusion_lr, annot=True, fmt='d', cmap='Blues')
plt.title('Logistic Regression Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

Output:

Logistic Regression Accuracy: 0.77
Confusion Matrix:
[[83 13]
[33 71]]

SVM

```
svm = SVC()
svm.fit(X_train, y_train)
y_pred_svm = svm.predict(X_test)

accuracy_svm = accuracy_score(y_test, y_pred_svm)
confusion_svm = confusion_matrix(y_test, y_pred_svm)

print(f'SVM Accuracy: {accuracy_svm}')
print('Confusion Matrix:\n', confusion_svm)

sns.heatmap(confusion_svm, annot=True, fmt='d', cmap='Reds')
plt.title('SVM Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

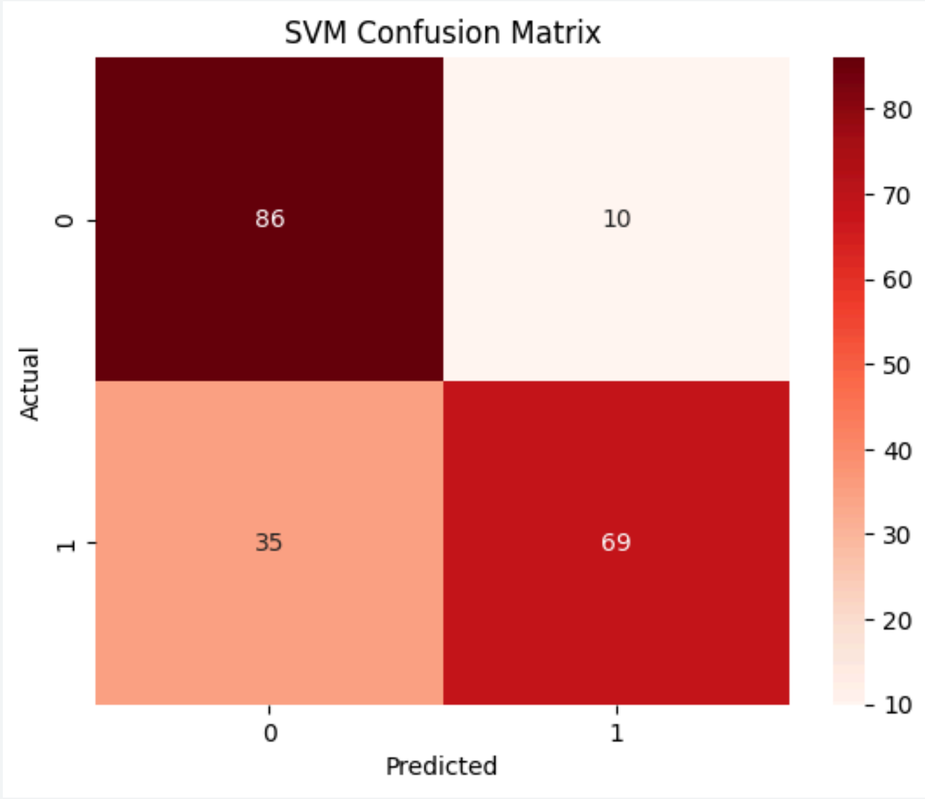
Output:

SVM Accuracy: 0.775

Confusion Matrix:

[[86 10]

[35 69]]

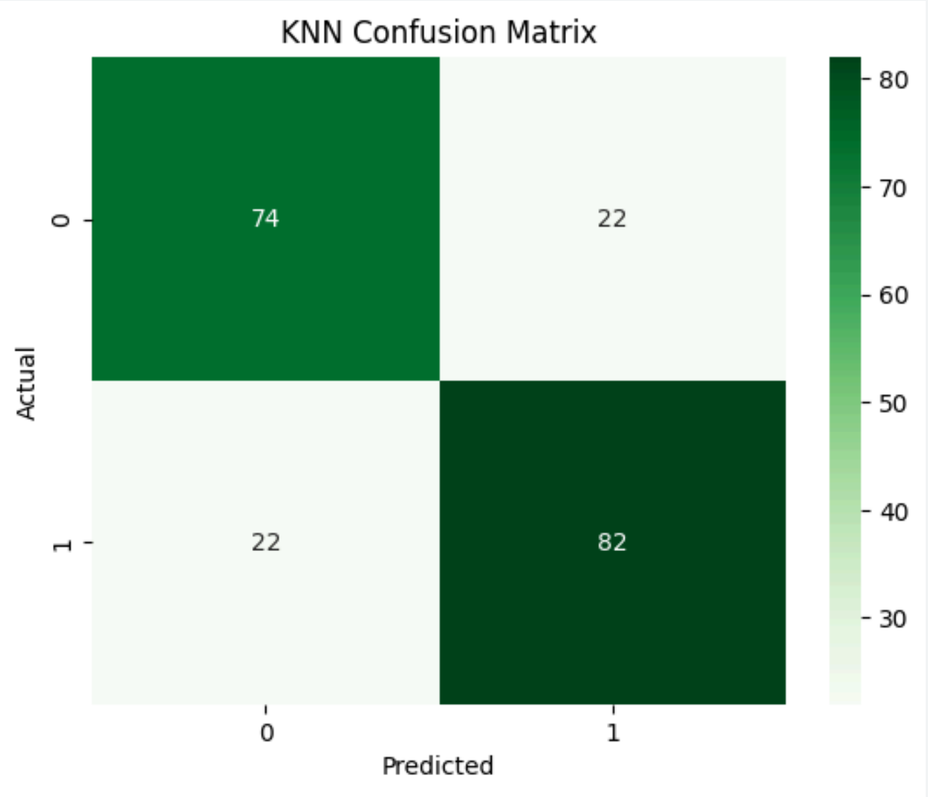


KNN

```
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

y_pred_knn = knn.predict(X_test)

accuracy_knn = accuracy_score(y_test, y_pred_knn)
confusion_knn = confusion_matrix(y_test, y_pred_knn)
print(f'KNN Accuracy: {accuracy_knn}')
print('Confusion Matrix:\n', confusion_knn)
sns.heatmap(confusion_knn, annot=True, fmt='d', cmap='Greens')
plt.title('KNN Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```



Output:

KNN Accuracy: 0.78

Confusion Matrix:

[[74 22]

[22 82]]

CONCLUSION

In our sentiment analysis project, we evaluated Logistic Regression, K-Nearest Neighbors, and Support Vector Machines, achieving accuracies of approximately 77%, 78%, and 77.5%, respectively. These results indicate that all three algorithms perform reasonably well for classifying reviews as positive or negative. KNN showed a slight edge in accuracy, highlighting its suitability for capturing local data patterns. Moving forward, optimizing parameters and exploring ensemble methods could further enhance classification performance. Overall, our study demonstrates the effectiveness of these algorithms in sentiment analysis tasks.

FUTURE SCOPE

Using Many Languages:

- Make sure the program can work with reviews in different languages.
- This would help more people understand what customers think.

Using Deep Learning:

- Try more advanced models that can understand the context of sentences.
- These models can make more accurate predictions.

REFERENCES

https://ieeexplore.ieee.org/document/9512387

"Opinion Mining and Sentiment Analysis" by Bo Pang and Lillian Lee - ResearchGate website

https://www.cambridge.org/9781108486378

THANK YOU