Report On

Sentiment Analysis on McDonald’s Stores Reviews

Submitted in partial fulfillment of the requirements of the Course project in Semester VII of Final Year Computer Science and Engineering (Data Science)

by
Kirtika Iyer (19)
Abhay Shukla (57)
Chirag Thanth (50)
Dhanesh Yadav (67)

Supervisor
Prof. Raunak Joshi

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Computer Science and Engineering (Data Science)

(2023-24)

Vidyavardhini's College of Engineering & Technology
Department of Computer Science and Engineering (Data Science)

CERTIFICATE

This is to certify that the project entitled “Sentiment Analysis on McDonald’s Stores Reviews” is a bonafide work of "Kirtika Iyer (19), Abhay Shukla (57),Chirag Thanth (50), Dhanesh Yadav (67)" submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester VII of Final Year Computer Science and engineering (Data Science).

Supervisor

Prof. Raunak Joshi

Dr. Vikas Gupta
Head of Department

Table of Contents

Pg. No

Chapter No	Title	Page No.
1	**Abstract**	**1**
2	**Problem Statement**	**2**
3	**Block Diagram**	**3**
4	**Module Description**	**4**
5	**Brief Description of Software & Hardware requirements**	**5**
6	**Code**	**6**
7	**Result**	**9**
8	**Conclusion**	**10**
9	**References**	**11**

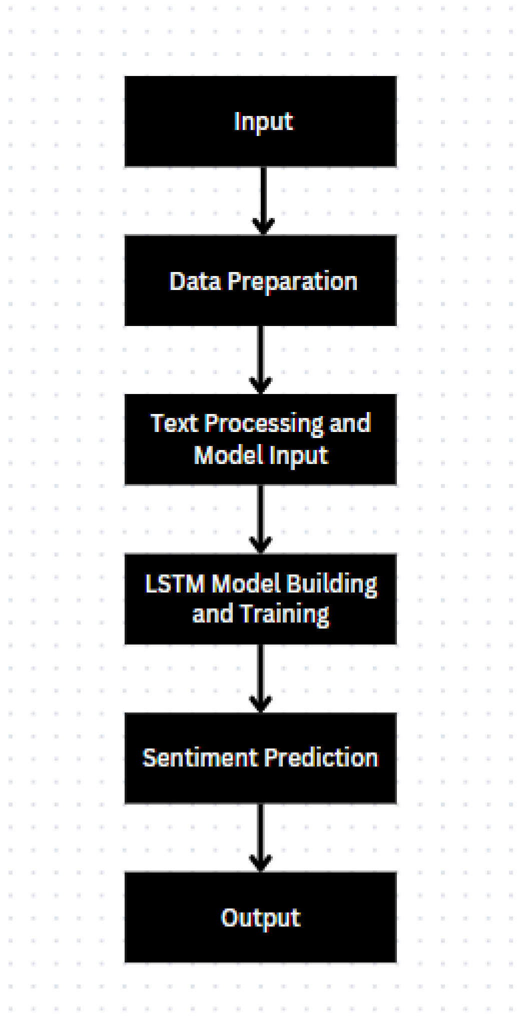
Abstract

This study presents a comprehensive sentiment analysis of customer reviews for McDonald's stores, aiming to gain insights into the perceptions and sentiments of consumers towards the fast-food chain. The vast proliferation of online review platforms has made customer feedback a valuable source of information for businesses to improve their services and products. This research focuses on collecting and analyzing a diverse dataset of reviews from various online platforms, encompassing a wide range of geographic locations and cultural contexts.The study employs natural language processing and machine learning techniques to classify reviews into positive, negative, or neutral sentiments. Text preprocessing, feature engineering, and state-of-the-art sentiment analysis algorithms are utilized to effectively categorize the reviews. Additionally, the research investigates the key factors influencing sentiment by conducting thematic analysis to identify recurring themes and topics within the reviews.

Problem Statement

Sentiment analysis of McDonald's store reviews is crucial for understanding and improving customer satisfaction, identifying areas of strength and weakness, and enhancing overall brand perception. The objective of this project is to develop a comprehensive sentiment analysis system to analyze and interpret customer reviews of McDonald's stores, extracting valuable insights to aid in decision-making and strategic planning.By addressing these challenges and objectives, the sentiment analysis of McDonald's store reviews aims to provide valuable insights, enhance customer satisfaction, and ultimately contribute to the improvement of McDonald's overall brand image and customer experience.

Block Diagram



Module Description

Module 1: Data Preparation

Input: McDonald's store reviews dataset
Preprocess the dataset, including loading the data, performing sentiment analysis with VADER, and splitting it into training and testing sets.
Output: Preprocessed data (X_train, X_test, y_train, y_test)

Module 2: Text Processing and Model Input

Input: Preprocessed data from Module 1
Tokenization: Convert text reviews into sequences of numerical values.
Padding: Ensure all sequences are of the same length.
Encoding: Encode sentiment labels into numerical values.
Output: Tokenized and padded sequences (X_train_pad, X_test_pad) and encoded labels (y_train_encoded, y_test_encoded)

Module 3: LSTM Model Building and Training

Input: Tokenized and padded sequences and encoded labels
Build an LSTM model with layers for embedding, LSTM, and dense (output).
Compile the model with loss, optimizer, and metrics.
Train the model on the training data.
Output: Trained LSTM model.

Module 4: Sentiment Prediction

Input: Trained LSTM model, tokenized and padded review sequence
Predict sentiment for new reviews using the trained model.
Output: Predicted sentiment label (Positive, Negative, or Neutral)

Brief description of software and hardware requirements

Software Requirements

Python: The entire project is implemented in Python. Make sure you have Python installed. You can use Python 3.x.

Python Libraries: Install the necessary Python libraries, including:
pandas
numpy
matplotlib
seaborn
nltk
scikit-learn
tensorflow (for deep learning)
nltk's VADER sentiment analysis tool

Jupyter Notebook or IDE

Hardware Requirements

CPU: While basic training and inference can be done on most CPUs, for more complex models and larger datasets, a more powerful CPU is beneficial. A multi-core processor will speed up training times.

GPU (Optional): For deep learning tasks, especially training large models, having a GPU can significantly speed up the process. Nvidia GPUs with CUDA support are commonly used for this purpose.

Memory (RAM): Having an ample amount of RAM is crucial for working with large datasets. A minimum of 8GB is recommended, but 16GB or more is better, especially when working with deep learning models.

Storage: We require sufficient storage space for your dataset, code, and model checkpoints. SSDs are faster for data access and model saving/loading.

Operating System: The code can run on Windows, macOS, or Linux. Choose an OS that you are comfortable with.

Internet Connection: To download libraries, models, and datasets, you'll need an internet connection. Additionally, if you plan to deploy the model, an internet connection might be required for model updates and data retrieval.

Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense
from tensorflow.keras.optimizers import Adam
# Load the data
mcd = pd.read_csv('./McDonald_s_Reviews.csv', encoding='latin-1')
# Sentiment analysis using VADER
sia = SentimentIntensityAnalyzer()
mcd['compound_score']=mcd['review'].apply(lambda x:
sia.polarity_scores(x)['compound'])
mcd['sentiment'] = mcd['compound_score'].apply(lambda score: 'Positive' if score >=
0.05 else 'Negative' if score <= -0.05 else 'Neutral')
# Split the data
X = mcd['review']
y = mcd['sentiment']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Tokenization and Padding
max_words = 1000
max_len = 100
tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(X_train)
X_train_seq = tokenizer.texts_to_sequences(X_train)
X_test_seq = tokenizer.texts_to_sequences(X_test)
X_train_pad = pad_sequences(X_train_seq, maxlen=max_len)
X_test_pad = pad_sequences(X_test_seq, maxlen=max_len)
# Encoding the labels
label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(y_train)
y_test_encoded = label_encoder.transform(y_test)
# Build the LSTM model
model = Sequential()
model.add(Embedding(input_dim=max_words,output_dim=128, input_length=max_len))
model.add(LSTM(128))
model.add(Dense(3, activation='softmax'))
# Compile the model
model.compile(loss='sparse_categorical_crossentropy',
optimizer=Adam(learning_rate=0.001), metrics=['accuracy'])
```

```
# Train the model
model.fit(X_train_pad, y_train_encoded, validation_data=(X_test_pad, y_test_encoded),
epochs=5, batch_size=64)
# Evaluate the model
y_pred = model.predict(X_test_pad)
y_pred_labels = [label_encoder.classes_[np.argmax(pred)] for pred in y_pred]
accuracy = accuracy_score(y_test, y_pred_labels)
print("Accuracy:", accuracy)
print("Classification Report:")
print(classification_report(y_test, y_pred_labels))
# Function to predict sentiment using the trained model
def predict_sentiment(review):
    review_seq = tokenizer.texts_to_sequences([review])
    review_pad = pad_sequences(review_seq, maxlen=max_len)
    sentiment_prob = model.predict(review_pad)
    sentiment_label = label_encoder.classes_[np.argmax(sentiment_prob)]
    return sentiment_label
# Test the model with new reviews
new_review = "This restaurant has excellent service and delicious food."
predicted_sentiment = predict_sentiment(new_review)
print("Predicted sentiment:", predicted_sentiment)
new_review2 = "This restaurant sucks."
predicted_sentiment = predict_sentiment(new_review2)
print("Predicted sentiment:", predicted_sentiment)
new_review3 = "This is dull"
predicted_sentiment = predict_sentiment(new_review3)
print("Predicted sentiment:", predicted_sentiment)
```

Result

Model Training:

```
Epoch 1/5
341/341 [==============================] - 84s 237ms/step - loss: 0.5680 - accuracy: 0.7659 - val_loss: 0.3905 - val_accuracy: 0.8550
Epoch 2/5
341/341 [==============================] - 91s 268ms/step - loss: 0.3439 - accuracy: 0.8788 - val_loss: 0.3571 - val_accuracy: 0.8747
Epoch 3/5
341/341 [==============================] - 84s 246ms/step - loss: 0.3130 - accuracy: 0.8896 - val_loss: 0.3530 - val_accuracy: 0.8730
Epoch 4/5
341/341 [==============================] - 81s 238ms/step - loss: 0.2883 - accuracy: 0.8977 - val_loss: 0.3404 - val_accuracy: 0.8802
Epoch 5/5
341/341 [==============================] - 79s 231ms/step - loss: 0.2658 - accuracy: 0.9052 - val_loss: 0.3392 - val_accuracy: 0.8879
<keras.src.callbacks.History at 0x7a27b07c9b10>
```

Model Evaluation:

```
171/171 [==============================] - 11s 61ms/step
Accuracy: 0.8878693338227198
Classification Report:
              precision    recall  f1-score   support

    Negative       0.84      0.85      0.85      1525
     Neutral       0.81      0.90      0.85      1046
    Positive       0.94      0.90      0.92      2878

    accuracy                           0.89      5449
   macro avg       0.87      0.88      0.87      5449
weighted avg       0.89      0.89      0.89      5449
```

Predictions:

```
new_review = "This restaurant has excellent service and delicious food."
predicted_sentiment = predict_sentiment(new_review)
print("Predicted sentiment:", predicted_sentiment)
```

```
1/1 [==============================] - 0s 33ms/step
Predicted sentiment: Positive
```

```
new_review2 = "This restaurant sucks."
predicted_sentiment = predict_sentiment(new_review2)
print("Predicted sentiment:", predicted_sentiment)
```

```
1/1 [==============================] - 0s 38ms/step
Predicted sentiment: Negative
```

```
new_review3 = "This is dull"
predicted_sentiment = predict_sentiment(new_review3)
print("Predicted sentiment:", predicted_sentiment)
```

```
1/1 [==============================] - 0s 63ms/step
Predicted sentiment: Neutral
```

Conclusion

The code performs sentiment analysis on Mc Donald’s Store reviews using an LSTM (Long Short Term Memory) model through the following four stages:Data Preparation, Text Processing and Model Input, LSTM (Long Short Term Memory) Model Building and Training, Sentiment Prediction.In conclusion, the code successfully demonstrates the implementation of a sentiment analysis project using an LSTM (Long Short Term Memory) model. It pre-processes the data, trains an LSTM (Long Short Term Memory) model, and provides a way to predict sentiments for new reviews. The accuracy and classification report help assess the model's performance. This project can be used to classify sentiments in various applications, such as customer reviews, feedback analysis, and more.

References

- Rose, S., Sreejith, R., & Senthil, S. (2019). Social media data analytics to improve the customer services: the case of fast-food companies. *International Journal of Recent Technology and Engineering*, *8*(2), 6359-6366.
- Hariguna, T., Sukmana, H. T., & Kim, J. I. (2020). Survey opinion using sentiment analysis. *J. Appl. Data Sci*, *1*(1), 35-40.
- HaqNawaz, T. A., Al-laith, A., Ahmad, I., Tharanidharan, S., & Nazar, S. K. A. SENTIMENTAL ANALYSIS OF SOCIAL MEDIA TO FIND OUT CUSTOMER OPINION.
- Kang, D., & Park, Y. (2014). based measurement of customer satisfaction in mobile service: Sentiment analysis and VIKOR approach. *Expert Systems with Applications*, *41*(4), 1041-1050.
- Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, *56*(4), 82-89.