



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

## **Sentiment Analysis on US Airline Reviews**

By

PARIMI NRUTHI SRI NEHA (20MIC0077)

SRI SOWJANYA TATAVARTI (20MIC0089)

NARSEPALLI SREEJA (20MIC0090)

DEVATHA CHANDINI SRI SAI GAYATHRI (20MIC0100)

SUBMITTED TO:

**Prof.Boominathan. P**

## **ABSTRACT:**

Sentiment analysis is nothing, it just recognizes the sentiment behind the text. It is often used by businesses to detect sentiment in social data, product reputation, and understanding customers.

When it comes to decision making, internet is playing a significant role, all around the world. Many people use the blogs, social media, and other online platforms to share their thoughts and views via internet. This results on the internet being filled, with full of relevant and irrelevant information. So, it creates a great challenge of fetching the desired information over the internet by analyzing each document.

Sentiment analysis paves the way on handling this problem at ease. This greatly helps customers in decision making on selection of best fit US Airlines on analyzing the other customer's opinion in online review sites like Skytrax and other micro-blogging sites like Twitter which provides the Aspect level sentiment analysis.

Sentiment analysis is a valuable tool for airlines to gain insights into customer feedback and identify areas for improvement. By analysing customer sentiments towards various aspects of their services such as in-flight experience, customer service, baggage handling, and pricing, airlines can improve the overall customer experience and build customer loyalty.

Sentiment analysis is a type of natural language processing problem that determines the sentiment or emotion of a piece of text. For example, an algorithm could be constructed to classify whether a product's review was positive, neutral, or negative. Natural language processing (NLP) is a field of artificial intelligence that involves computers understanding and processing human language. The goal is often to derive meaning from text. Additionally, there are numerous techniques that can be applied to analyse text and extract meaning. For this project, we will focus on building a recurrent neural network (RNN) to classify the sentiment of tweets about airlines using Keras and a pretrained word embedding.

Social media listening tools can be used to monitor and respond to customer complaints in real-time, demonstrating a commitment to addressing customer concerns. Sentiment analysis can provide airlines with valuable insights into

customer sentiment and opinions, enabling them to make data-driven decisions to improve their services and enhance customer satisfaction.

## **INTRODUCTION:**

Sentiment analysis can be particularly useful for airlines to gain insights into how their customers feel about their services and identify areas for improvement. Airlines can use sentiment analysis to analyse customer feedback from various sources such as social media, surveys, and customer support channels.

By using sentiment analysis, airlines can identify the sentiments of their customers towards various aspects of their services such as in-flight experience, customer service, baggage handling, and pricing. This information can be used to improve the overall customer experience and build customer loyalty.

For example, if the sentiment analysis shows that customers are consistently complaining about the long wait times at the baggage claim area, the airline can take steps to improve their baggage handling process, such as increasing staffing levels, streamlining the baggage handling process, or implementing self-service baggage drop-off systems.

Sentiment analysis can also help airlines to monitor and respond to customer complaints in real-time. By using social media listening tools, airlines can quickly identify negative sentiment towards their services and respond to customer complaints promptly, demonstrating that they value their customers and are committed to addressing their concerns.

Overall, sentiment analysis can provide airlines with valuable insights into the sentiments and opinions of their customers, allowing them to improve their services and enhance the customer experience.

Airline industry is one of the largest and leading industries in the world which enables services to thousands of customers in a single day. Approximately 2,246,000 passengers adopt flights in the United States of America (USA) per day as per the reports provided by Federal Aviation Administration Air-Traffic [FAA]. This project research is focused on the top ten US based airline carriers namely America Airlines, Alaska Airlines, Delta Airlines, JetBlue Airlines, Hawaiian Airlines, SkyWest Airlines, Southwest Airlines, United Airlines, Spirit Airlines and Us Airways are the top ten US based carriers of airline which is taken into account in this paper.

In USA, same geographical area is covered by these airlines during flight which makes it to fall under the primary position for choosing these airlines. Added to this these are the low cost carriers in USA and similar flight fare is identified. Furthermore, great competition is going on among them which force the competitors to create a good competitive edge. Not much more results have been found on the research on airline industry based on aspect of sentiment analysis. This research greatly focuses on bridging the gaps between the customer's views and airlines carriers as a great milestone. Further the implementation of the proposed research occurs in other domains such as entertainment, education, automobiles, etc.

Sentiment analysis (SA) tells user whether the information about the product is satisfactory or not before they buy it. Marketers and firms use this analysis data to understand about their products or services in such a way that it can be offered as per the user's requirements. Textual Information retrieval techniques mainly focus on processing, searching, or analyzing the factual data present. Facts have an objective component but, there are some other textual contents which express subjective characteristics. These contents are mainly opinions, sentiments, appraisals, attitudes, and emotions, which form the core of Sentiment Analysis (SA). It offers many challenging opportunities to develop new applications, mainly due to the huge growth of available information on online sources like blogs and social networks. For example, recommendations of items proposed by a recommendation system can be predicted by taking into account considerations such as positive or negative opinions about those items by making use of SA.

## **Background**

### Recurrent Neural Network (RNN)

A RNN allows information from a previous output to be fed as input into the current state. Simply put, we can use previous information to help make a current decision.

### Long Short-Term Memory (LSTM)

Simple RNNs suffer from the vanishing gradient problem which occurs when information from earlier layers disappear as the network becomes deeper.

A LSTM algorithm was created to avoid this problem by allowing the neural network to carry information across multiple time steps. This means it can save important information for later use, preventing gradients from vanishing during the process. Additionally, a LSTM cell can determine what information to remove as well. Therefore, it can learn to recognize an important input and store it for the future while removing unnecessary information.

## **Data**

14640 tweets from 7700 users were analysed. Twitter data was scraped from February of 2015 and contributors were asked to first classify positive, negative, and neutral tweets, followed by categorizing negative reasons (such as "late flight" or "rude service").

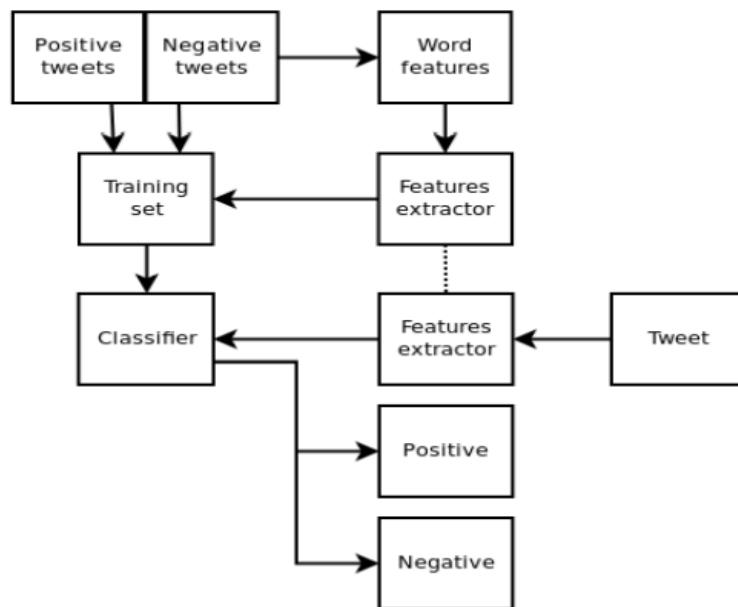
## **ARCHITECTURE:**

NLP models work by finding relationships between the constituent parts of language — for example, the letters, words, and sentences found in a text dataset. NLP architectures use various methods for data pre-processing, feature extraction, and modelling.

Natural language processing (NLP) is the discipline of building machines that can manipulate human language — or data that resembles human language — in the way that it is written, spoken, and organized. It evolved from computational linguistics, which uses computer science to understand the principles of language, but rather than developing theoretical frameworks, NLP is an engineering discipline that seeks to build technology to accomplish useful tasks. NLP can be divided into two overlapping subfields: natural language understanding (NLU), which focuses on semantic analysis or determining the intended meaning of text, and natural language generation (NLG), which focuses on text generation by a machine. NLP is separate from — but often used in conjunction with — speech recognition, which seeks to parse spoken language into words, turning sound into text and vice versa.

Long Short-Term Memory is an advanced version of recurrent neural network (RNN) architecture that was designed to model chronological sequences and their long-range dependencies more precisely than conventional RNNs. The

major highlights include the interior design of a basic LSTM cell, the variations brought into the LSTM architecture, and few applications of LSTMs that are highly in demand. It also makes a comparison between LSTMs and GRUs. The article concludes with a list of disadvantages of the LSTM network and a brief introduction of the upcoming attention-based models that are swiftly replacing LSTMs in the real world.



**Fig.1. Sentiment Analysis Architecture**

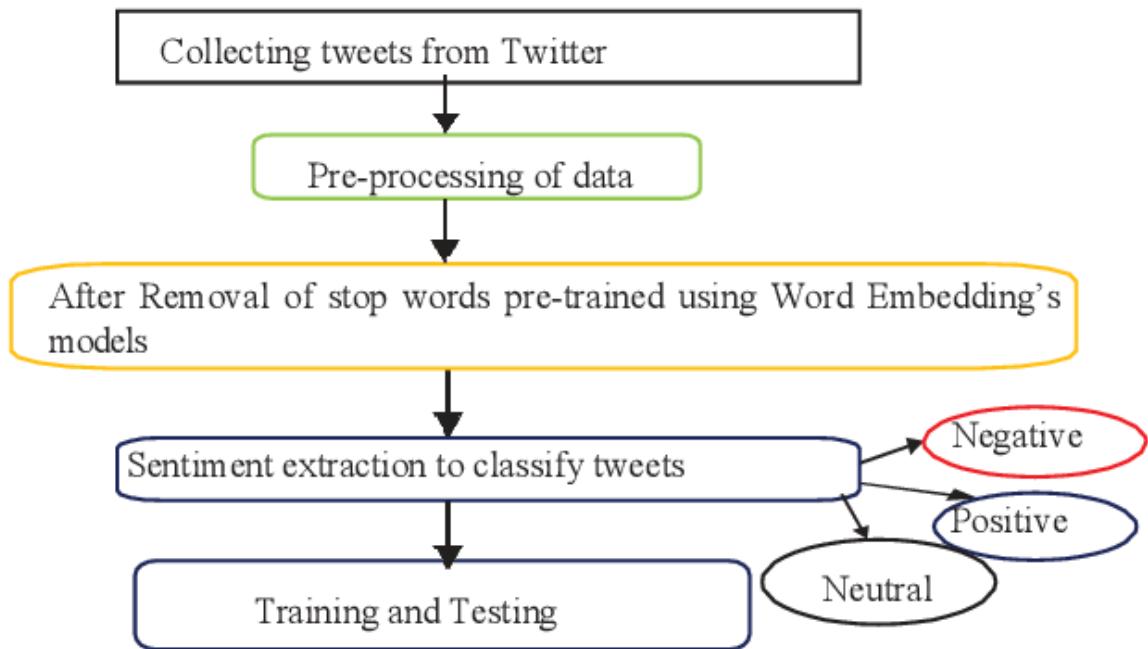
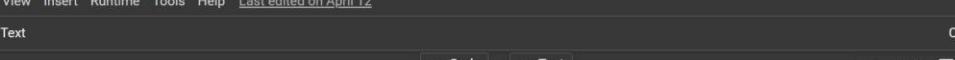


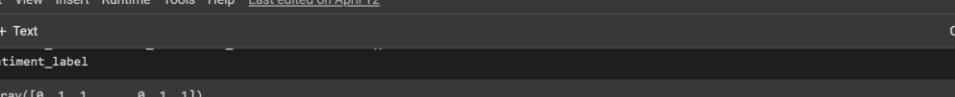
Fig. 2. Flow diagram of data

## SINPPETS:



The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** "Sentiment Analysis.ipynb" with a star icon.
- Menu Bar:** File, Edit, View, Insert, Runtime, Tools, Help. A note "Last edited on April 12" is displayed.
- Toolbar:** Comment, Share, Settings, and a "P" icon.
- Code Cell:** Contains Python code for sentiment analysis. The code imports pandas, matplotlib.pyplot, Tokenizer, pad\_sequences, Sequential, LSTM, Dense, Dropout, SpatialDropout1D, and Embedding from tensorflow.keras. It also reads a CSV file named "Tweets.csv".
- Cell Buttons:** + Code, + Text, and other standard notebook controls.



A screenshot of a Jupyter Notebook interface. The title bar says "Sentiment Analysis.ipynb". The menu bar includes File, Edit, View, Insert, Runtime, Tools, Help, and a timestamp "Last edited on April 12". On the right, there are buttons for Comment, Share, and settings. Below the menu, there are buttons for "Code" and "Text". The main area shows Python code for sentiment analysis:

```
+ Code + Text

sentiment_label
(array([0, 1, 1, ..., 0, 1, 1]),
Index(['positive', 'negative'], dtype='object'))

[ ] tweet = tweet_df.text.values
tokenizer = Tokenizer(num_words=5000)
tokenizer.fit_on_texts(tweet)
vocab_size = len(tokenizer.word_index) + 1
encoded_docs = tokenizer.texts_to_sequences(tweet)
padded_sequence = pad_sequences(encoded_docs, maxlen=200)

[ ] print(tokenizer.word_index)

{'to': 1, 'the': 2, 'i': 3, 'a': 4, 'united': 5, 'you': 6, 'for': 7, 'flight': 8, 'and': 9, 'on': 10, 'my': 11, 'usairways': 12, 'americanair': 13, 'is': 14}
```

Sentiment Analysis.ipynb

```
embedding_vector_length = 32
model = Sequential()
model.add(Embedding(vocab_size, embedding_vector_length, input_length=200))
model.add(SpatialDropout1D(0.25))
model.add(LSTM(50, dropout=0.5, recurrent_dropout=0.5))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
```

Model: "sequential\_3"

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 200, 32)	423488
spatial_dropout1d_1 (SpatialDropout1D)	(None, 200, 32)	0
lstm_1 (LSTM)	(None, 50)	16600
dropout_1 (Dropout)	(None, 50)	0
dense_1 (Dense)	(None, 1)	51

Sentiment Analysis.ipynb

```
File Edit View Insert Runtime Tools Help Saving...
```

[17] history = model.fit(padded\_sequence,sentiment\_label[0],validation\_split=0.2, epochs=5, batch\_size=32)

```
Epoch 1/5
289/289 [=====] - 116s 401ms/step - loss: 0.0973 - accuracy: 0.9660 - val_loss: 0.1773 - val_accuracy: 0.9389
Epoch 2/5
289/289 [=====] - 99s 344ms/step - loss: 0.0848 - accuracy: 0.9680 - val_loss: 0.2025 - val_accuracy: 0.9437
Epoch 3/5
289/289 [=====] - 99s 343ms/step - loss: 0.0766 - accuracy: 0.9738 - val_loss: 0.2046 - val_accuracy: 0.9428
Epoch 4/5
289/289 [=====] - 97s 337ms/step - loss: 0.0768 - accuracy: 0.9701 - val_loss: 0.1960 - val_accuracy: 0.9329
Epoch 5/5
289/289 [=====] - 100s 347ms/step - loss: 0.0675 - accuracy: 0.9755 - val_loss: 0.2488 - val_accuracy: 0.9381
```

Sentiment Analysis.ipynb

```
File Edit View Insert Runtime Tools Help All changes saved
```

+ Code + Text

```
[ ] def predict_sentiment(text):
    tw = tokenizer.texts_to_sequences([text])
    tw = pad_sequences(tw,maxlen=200)
    prediction = int(model.predict(tw).round().item())
    print("Predicted label: ", sentiment_label[1][prediction])
```

```
[ ] test_sentence1 = "I enjoyed my journey on this flight."
predict_sentiment(test_sentence1)

test_sentence2 = "This is the worst flight experience of my life!"
predict_sentiment(test_sentence2)
```

```
1/1 [=====] - 0s 43ms/step
Predicted label: positive
1/1 [=====] - 0s 42ms/step
Predicted label: negative
```

Sentiment Analysis.ipynb

```
[ ] pd.crosstab(df['airline'], df['negativereason'])

<table border="1">
|  | negativereason | Bad Flight | Can't Tell | Cancelled Flight | Customer Service Issue | Damaged Luggage | Flight Attendant Complaints | Flight Booking Problems | Late Flight | Lost Luggage | longlines |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| airline |  |  |  |  |  |  |  |  |  |  |  |
| American | 87 | 198 | 246 | 768 | 12 | 87 | 130 | 249 | 149 | 34 |  |
| Delta | 64 | 186 | 51 | 199 | 11 | 60 | 44 | 269 | 57 | 14 |  |
| Southwest | 90 | 159 | 162 | 391 | 14 | 38 | 61 | 152 | 90 | 29 |  |
| US Airways | 104 | 246 | 189 | 811 | 11 | 123 | 122 | 453 | 154 | 50 |  |
| United | 216 | 379 | 181 | 681 | 22 | 168 | 144 | 525 | 269 | 48 |  |
| Virgin America | 19 | 22 | 18 | 60 | 4 | 5 | 28 | 17 | 5 | 3 |  |



Negative reason per airline
```

Sentiment Analysis.ipynb

```
[ ] the same i.e. Customer Service Issue followed by Late Flight.

[ ] import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
sia = SentimentIntensityAnalyzer()
def get_sentiment_scores(review):
    scores = sia.polarity_scores(review)
    return scores['neg'], scores['neu'], scores['pos']
def get_rating(neg, neu, pos):
    if pos > neg and pos > neu:
        return 5
    elif pos > neg and pos == neu:
        return 4
    elif neu > neg and neu > pos:
        return 3
    elif neg > pos and neg > neu:
        return 1
    else:
        return 2

rating_df = pd.DataFrame(df['text'].apply(get_sentiment_scores).tolist(),
columns=['negative', 'neutral', 'positive'])
```

Sentiment Analysis.ipynb

```
[ ] return 2

[ ] rating_df = pd.DataFrame(df['text'].apply(get_sentiment_scores).tolist(),
columns=['negative', 'neutral', 'positive'])

rating_df['rating'] = rating_df.apply(lambda row: get_rating(row['negative'], row['neutral'], row['positive']), axis=1)

# add the sentiment labels to the new dataframe
rating_df['sentiment'] = df['airline_sentiment']

# print the average ratings for each sentiment label
print("Average ratings by label:")
print(rating_df.groupby('sentiment')['rating'].mean())

Average ratings by label:
sentiment
negative    2.988124
neutral     3.058083
positive    3.384257
Name: rating, dtype: float64
```

Sentiment Analysis.ipynb

```
import numpy as np
from tensorflow.keras.models import load_model

# Define the rating function
def get_rating(prediction):
    if prediction >= 0.8:
        return 'Positive'
    elif prediction < 0.8 and prediction >= 0.5:
        return 'Neutral'
    else:
        return 'Negative'

# Load the test data
test_data = pd.read_csv('Tweets.csv')

# Preprocess the test data
test_tweet = test_data['text'].values
tokenizer = Tokenizer(num_words=5000)
tokenizer.fit_on_texts(test_tweet)
vocab_size = len(tokenizer.word_index) + 1
encoded_docs = tokenizer.texts_to_sequences(test_tweet)
```

Sentiment Analysis.ipynb

```
padded_sequence = pad_sequences(encoded_docs, maxlen=200)

# Make predictions
predictions = model.predict(padded_sequence)

# Get the rating for each prediction
ratings = np.vectorize(get_rating)(predictions)
# Add the ratings to the test data
test_data['rating'] = ratings

# Print the test data with the ratings
print(test_data)

458/458 [=====] - 20s 43ms/step
      tweet_id airline sentiment airline_sentiment_confidence \
0      570306133677760513     neutral          1.0000
1      570301130888122368   positive          0.3486
2      570301083672813571     neutral          0.6837
3      570301031407624196   negative          1.0000
4      570300817074462722   negative          1.0000
...       ...
14635  569587686496825344   positive          0.3487
14636  569587371693355008   negative          1.0000
```

