

Phase 4: Development Part 2

PROBLEM STATEMENT:

In this technology you will continue building your project by selecting a machine learning algorithm, training the model, and evaluating its performance.

STEPS INVOLVED IN TRAINING AND EVALUATION:

Feature Extraction:

- Convert the text data into numerical features suitable for machine learning.
- Using word embeddings like Word2Vec, GloVe, or FastText.
- Creating a Bag of Words (BoW) or TF-IDF representation.
- Using pre-trained language models like BERT, GPT-2, or similar models.

Select a Machine Learning Algorithm:

- Choose a suitable machine learning algorithm for sentiment analysis. Common choices include:
- Logistic Regression
- Support Vector Machines
- Naive Bayes
- Deep Learning models like LSTM or CNN
- Transformers (e.g., BERT)

Training the Model:

- Train the selected model on the training data.
- For deep learning models, this involves setting hyperparameters, defining the architecture, and training for a specified number of epochs.

Hyperparameter Tuning:

- Fine-tune hyperparameters such as learning rate, batch size, and model architecture to optimize performance on the validation set.

Model Evaluation:

- Evaluate the model's performance using appropriate metrics. For sentiment analysis, common metrics include accuracy, precision, recall, F1-score, and confusion matrices.
- Visualize the results, if necessary, to gain insights into model behaviour.

Source Code:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.model_selection import StratifiedShuffleSplit
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords

#### Load your dataset, assuming it's in a CSV file
data = pd.read_csv('Tweets.csv')

#### Inspect the first few rows of the dataset
print(data.head())

#### Text cleaning
data['text'] = data['text'].str.lower()
data['text'] = data['text'].str.replace('[^a-zA-Z]', ' ', regex=True)
print("After Text Cleaning:")
print(data['text'].head())

#### Tokenization (using NLTK as an example)
data['tokens'] = data['text'].apply(word_tokenize)
print("After Tokenization:")
print(data['tokens'].head())

#### Remove stop words (using NLTK as an example)
stop_words = set(stopwords.words('english'))
data['tokens'] = data['tokens'].apply(lambda tokens: [word for word in tokens
if word not in stop_words])
print("After Stop Words Removal:")
print(data['tokens'].head())

#### Label encoding (assuming you have 'sentiment' as the label column)
sentiment_mapping = {'negative': 0, 'neutral': 1, 'positive': 2}
data['airline_sentiment'] = data['airline_sentiment'].map(sentiment_mapping)
print("After Label Encoding:")
print(data['airline_sentiment'].head())
X = data['text'] # Input features
y = data['airline_sentiment'] # Target variable

#### Perform Stratified Sampling with a 80-20 split
stratified_split = StratifiedShuffleSplit(n_splits=1, test_size=0.2,
random_state=42)
for train_index, test_index in stratified_split.split(X, y):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
```

```

print("X_train:", X_train.head())
print("y_train:", y_train.head())
print("X_test:", X_test.head())
print("y_test:", y_test.head())

#### Extract TF-IDF features
vectorizer = TfidfVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)

#### Train a logistic regression model
clf = LogisticRegression()
clf.fit(X_train_vectorized, y_train)

#### Evaluate the model on the test set
y_pred = clf.predict(X_test_vectorized)
print(classification_report(y_test, y_pred))

```

Output:

```

PS C:\Users\arunr> cd c:\Users\arunr\Downloads\archive
PS C:\Users\arunr\Downloads\archive> & C:\Users\arunr\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Users\arunr\Downloads\archive/py

```

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	...	tweet_coord	tweet_created	tweet_location	user_timezone
0	570306133677760513	neutral	1.0000	NaN	...	NaN	2015-02-24 11:35:52 -0800	NaN	Eastern Time (US & Canada)
1	570301130888122368	positive	0.3486	NaN	...	NaN	2015-02-24 11:15:59 -0800	NaN	Pacific Time (US & Canada)
2	570301083672813571	neutral	0.6837	NaN	...	NaN	2015-02-24 11:15:48 -0800	Lets Play	Central Time (US & Canada)
3	570301031407624196	negative	1.0000	Bad Flight	...	NaN	2015-02-24 11:15:36 -0800	NaN	Pacific Time (US & Canada)
4	570300817074462722	negative	1.0000	Can't Tell	...	NaN	2015-02-24 11:14:45 -0800	NaN	Pacific Time (US & Canada)

```

[5 rows x 15 columns]
After Text Cleaning:
0    virginamerica what dhepburn said
1    virginamerica plus you ve added commercials t...
2    virginamerica i didn t today must mean i n...
3    virginamerica it s really aggressive to blast...
4    virginamerica and it s a really big bad thing...
Name: text, dtype: object
After Tokenization:
0    [virginamerica, what, dhepburn, said]
1    [virginamerica, plus, you, ve, added, commerci...
2    [virginamerica, i, didn, t, today, must, mean,...
3    [virginamerica, it, s, really, aggressive, to,...
4    [virginamerica, and, it, s, a, really, big, ba...
Name: tokens, dtype: object
After Stop Words Removal:
0    [virginamerica, dhepburn, said]
1    [virginamerica, plus, added, commercials, expe...
2    [virginamerica, today, must, mean, need, take,...
3    [virginamerica, really, aggressive, blast, obn...
4    [virginamerica, really, big, bad, thing]
Name: tokens, dtype: object
After Label Encoding:
0    1
1    2
2    1
3    0
4    0
Name: airline_sentiment, dtype: int64
X_train: 1262    united what would it cost
10772    usairways used get emails pre purchase a...
4204    united no it was flight cancelled flightla...
5491    southwestair not frustrated just an idea gr...
12096    americanair narrowly made standby lots of s...
Name: text, dtype: object

```

```
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
4204 united no it was flight cancelled flightla...
5491 southwestair not frustrated just an idea gr...
12096 americanair narrowly made standby lots of s...
Name: text, dtype: object
y_train: 1262 1
10772 1
4204 0
5491 2
12096 0
Name: airline_sentiment, dtype: int64
X test: 2998 united past
7719 jetblue would you say a delay is more likely ...
8575 jetblue i cheated on you and i m sorry i ll...
618 united disappointed that u didnt honor my ...
11741 usairways the airline is embarrassing itself ...
Name: text, dtype: object
y_test: 2998 1
7719 2
8575 0
618 0
11741 0
Name: airline_sentiment, dtype: int64
C:\Users\arunr\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\sklearn\linear_model\_logistic.py:460: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_i = _check_optimize_result(
precision recall f1-score support

0 0.83 0.94 0.88 1835
1 0.67 0.56 0.61 620
2 0.82 0.56 0.67 473

accuracy 0.80 2928
macro avg 0.77 0.69 0.72 2928
weighted avg 0.79 0.80 0.79 2928
```

Conclusion:

In summary, the trained sentiment analysis model for marketing provides actionable insights into customer sentiments and brand perception. The thorough training, evaluation, and deployment process ensures its reliability for data-driven marketing decisions, with continuous monitoring and documentation supporting its ongoing effectiveness and transparency.