

Sentiment Analysis Using Logistic Regression

Overview

Sentiment analysis is a natural language processing (NLP) technique used to determine the emotional tone behind textual data. It is commonly applied to understand opinions in reviews, social media, and other text sources. In this project, we implement sentiment analysis using a logistic regression model to classify text as expressing either positive or negative sentiments.

Why Use Logistic Regression for Sentiment Analysis?

Logistic regression is a widely used algorithm for binary classification tasks due to its simplicity and effectiveness. It models the probability of a binary outcome based on one or more predictor variables. In the context of sentiment analysis, logistic regression can effectively distinguish between positive and negative sentiments by learning the relationship between textual features and sentiment labels.

Prerequisites

Before running the code, ensure you have the following installed:

- Python 3.x
- scikit-learn
- NumPy

You can install the required libraries using pip:

```
pip install scikit-learn numpy
```

Files Included

- `sentiment_analysis.py` : Contains the implementation of the sentiment analysis model using logistic regression.
-

Code Description

The code performs the following steps:

1. Import Necessary Libraries:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

- `CountVectorizer` : Converts text data into a matrix of token counts.
- `LogisticRegression` : Implements the logistic regression algorithm.
- `train_test_split` : Splits the dataset into training and testing sets.

2. Prepare the Dataset:

```
data = ["I love this product", "This is bad", "Amazing experience", "Worst purchase"]
labels = [1, 0, 1, 0] # 1: Positive, 0: Negative
```

- `data` : A list of text samples.
- `labels` : Corresponding sentiment labels (1 for positive, 0 for negative).

3. Convert Text to Numerical Data:

```
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(data)
```

- `CountVectorizer` transforms the text data into a numerical format suitable for model training.

4. Split the Data:

```
X_train, X_test, y_train, y_test = train_test_split(X, labels, test_size=0.25)
```

- The dataset is split into training and testing sets, with 25% of the data reserved for testing.

5. Train the Logistic Regression Model:

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

- A logistic regression model is instantiated and trained on the training data.

6. Make Predictions:

```
test_text = ["I hate this product"]
test_vector = vectorizer.transform(test_text)
prediction = model.predict(test_vector)
print("Predicted Sentiment:", "Positive" if prediction[0] == 1 else "Negative")
```

- The model predicts the sentiment of new text input.

Expected Outputs

For the provided test input `"I hate this product"`, the model outputs:

```
Predicted Sentiment: Negative
```

Use Cases

- **Product Reviews:** Analyzing customer feedback to determine overall satisfaction.
 - **Social Media Monitoring:** Gauging public sentiment on platforms like Twitter or Facebook.
 - **Market Research:** Understanding consumer opinions on products or services.
-

Advantages

- **Simplicity:** Logistic regression is easy to implement and interpret.
 - **Efficiency:** Suitable for large datasets with high-dimensional features.
 - **Baseline Performance:** Provides a strong baseline for binary classification tasks.
-

Future Enhancements

- **Data Expansion:** Incorporate a larger and more diverse dataset to improve model robustness.
 - **Feature Engineering:** Utilize advanced techniques like TF-IDF or word embeddings for better feature representation.
 - **Model Comparison:** Explore other classification algorithms such as Support Vector Machines or Neural Networks for potential performance gains.
-

References

- [Sentiment Analysis with Logistic Regression: A Hands-On Approach](#)
 - [Sentiment Analysis with Logistic Regression - Towards AI](#)
 - [Sentiment Analysis using Logistic Regression - De La Salle University](#)
-