Sentiment Analysis for Product Review

Ketil Jacobsen, Jonas Stokkeland 15.11.2024

https://672631-ml-project-streamlit-test-ddavmf.streamlit.app/

Describe The Problem

SCOPE

The goal of this project is to create a web application that is able to analyze the sentiment of product reviews as either positive or negative. The application helps businesses understand customer opinions by analyzing reviews submitted on their products.

Companies can use this tool to monitor customer satisfaction, identify products with negative feedback quickly and respond to customer needs.

The sentiment analysis application is designed as a self-contained system. The primary components include:

1. **Input Interface**: A web interface built using Streamlit, where users can input reviews for analysis.
2. **Sentiment Analysis Model**: A pre-trained BERT model (distilbert-base-uncased-finetuned-sst-2-english) for classifying the sentiment of text as positive or negative.
3. **Data Storage**: A local JSON file to store reviews and their analyzed sentiments for easy access.

Since the model is pre-trained, it operates as a standalone component, and its behavior is stable unless updates are made to the model itself or the supporting libraries. Changes to the input interface (e.g., adding new features) or data storage (e.g., transitioning from JSON to a database) would not significantly impact the model's functionality but may require adjustments in how data flows through the system. Overall, the system is modular, ensuring minimal interdependence between components.

The sentiment analysis web application is designed to serve multiple business functions and offers value across various teams:

Customer Support Teams: Quickly identify issues or complaints about products, enabling proactive customer service.

Product and Marketing Teams: Analyze customer sentiment to gauge the success of a product and inform marketing strategies or product adjustments.

Business Intelligence: Aggregate sentiment data over time to understand trends and customer loyalty, influencing broader business strategies.

The web application serves as a centralized platform where businesses can submit product reviews for sentiment analysis. They can either enter individual reviews or potentially the application can be expanded so that companies can integrate bulk submissions for a broader overview.

By analyzing recent reviews, businesses can quickly identify negative trends or recurring complaints about specific products. If many reviews are negative, they can take proactive measures to address quality or customer service issues.

The application could also be used regularly to monitor sentiment trends over time. For example, businesses could analyze sentiment on a weekly or monthly basis to see if customer satisfaction is improving or declining, especially after releasing a product update or responding to issues.

Several similar solutions for sentiment analysis already exist in the market. Here are some commonly used alternatives:

1. **Commercial Sentiment Analysis Tools**:
   - **Google Cloud Natural Language API**: Offers pre-built sentiment analysis models that are easy to integrate into applications.
   - **Amazon Comprehend**: AWS's natural language processing service includes sentiment analysis and works well for business use cases.
   - **IBM Watson Natural Language Understanding**: Provides sentiment analysis as part of its broader NLP services.
2. **Social Media Monitoring Tools**:
   - **Sprinklr, Hootsuite, and Brandwatch**: These tools offer sentiment analysis for social media mentions, providing businesses insights into customer sentiment across platforms.
3. **Custom-Built Sentiment Models**:
   - Companies may use custom-trained models or open-source models like BERT, trained or fine-tuned specifically for their own datasets.

Most businesses today use one of the commercial sentiment analysis services or build their own models using machine learning frameworks. However, these solutions can sometimes be costly, especially for small businesses or projects, which is why an open-source approach using a pre-trained BERT model can be a budget-friendly alternative.

The performance of the sentiment analysis web application will be evaluated based on its ability to meet the needs of businesses effectively. This will be achieved through several measurable outcomes:

1. **Accuracy**: The model's sentiment classifications (positive or negative) will be validated against test data, ensuring reliability for decision-making. A minimum classification accuracy of 80% is considered successful.
2. **Response Time**: The application will aim to deliver sentiment analysis results within 2 seconds per review, enabling businesses to process customer feedback in real-time.
3. **Ease of Use**: The simplicity of the interface ensures accessibility for non-technical users. Engagement metrics, such as the frequency of reviews submitted, will indicate how effectively the tool meets user needs.
4. **Adoption of Insights**: Businesses can use the insights generated to identify trends, address customer concerns, and guide product improvements. Success will be measured by how often the results lead to actionable decisions.

These metrics will ensure that the tool provides businesses with fast, accurate, and actionable insights into customer sentiment, thereby aligning with the project's goals.

This project is designed to provide value to multiple groups, each with distinct interests and needs.

1. **Primary Stakeholders**:
    - **Business Owners and Managers**: They rely on insights from the application to monitor customer satisfaction and identify areas for improvement in their products or services. This group benefits from data-driven decision-making enabled by the sentiment analysis tool.
    - **Customer Support Teams**: These stakeholders use the application to prioritize and address negative reviews quickly, ensuring improved customer satisfaction and retention.
    - **Product and Marketing Teams**: Insights from the application help these stakeholders understand customer preferences and tailor their strategies to enhance product offerings or marketing campaigns.
2. **Secondary Stakeholders**:
    - **Customers**: While not direct users of the application, they indirectly benefit as their feedback is analyzed and used to improve the products and services they use.

By considering the diverse needs of these stakeholders, the project aims to deliver a tool that provides actionable insights, fosters customer satisfaction, and drives continuous improvement for businesses

The successful completion of this project requires a combination of computational resources, tools, and personnel.

1. **Computational Resources:**
    - Pre-trained Model: The project utilizes a pre-trained BERT model, which can run efficiently on a CPU. A GPU is not strictly necessary but could improve processing speed during testing or deployment.

- Hosting Platform: The application will be deployed on a platform such as Streamlit Community Cloud, which provides free hosting for web applications and sufficient computational power for lightweight models like the one used here.

2. **Software Tools:**
   - Development Environment: Python and a virtual environment to manage dependencies, along with libraries like streamlit, transformers, and matplotlib.
   - Version Control: GitHub for version control and collaboration.
   - Text Editor or IDE: Tools such as Visual Studio Code for coding.
3. **Personnel:**
   - Team Members: A small team of 1-3 members with knowledge of Python programming, basic machine learning, and web development.

These resources ensure the project can be developed, tested, and deployed efficiently, meeting the expectations of all stakeholders.

METRICS

The success of this project will be evaluated through a combination of business and technical metrics, ensuring that the application meets its intended objectives effectively.

Minimum Business Metric Performance:

The application's primary business objective is to provide businesses with reliable insights into customer sentiment. For the project to be considered a success, the sentiment analysis system must achieve:

- **Accuracy**: At least 80% accuracy in classifying reviews as positive or negative, ensuring businesses can rely on the insights to make informed decisions.
- **Ease of Use**: The application should provide a user-friendly interface that allows non-technical users to analyze reviews efficiently.
- **Timeliness**: Sentiment results should be returned within 2 seconds per review to enable real-time decision-making.

Machine Learning Metrics:

To assess the system's performance, the following machine learning metrics will be used:

- **Accuracy**: The percentage of reviews correctly classified as positive or negative. This ensures the model aligns with the real-world sentiment of customer feedback.
- **Precision and Recall**: Precision will measure how well the model avoids false positives, while recall will ensure it correctly identifies negative reviews. Both are critical for maintaining trust in the analysis results.

Software Metrics:

The application’s usability and efficiency will be evaluated through:

- **Latency**: The time taken to process a single review and return a result. A latency of under 2 seconds ensures the app is responsive and suitable for real-time use.
- **Throughput**: The number of reviews the system can process in a given time. While designed for individual use, scalability will be considered for handling larger datasets if needed.
- **System Reliability**: Ensuring the app handles multiple inputs without crashing or producing errors.

Connection to Business Objectives:

These metrics directly support the business goals of improving customer feedback analysis and decision-making:

- High **accuracy** ensures businesses trust the results, leading to actionable insights.
- Low **latency** supports real-time responsiveness, enabling faster reactions to trends.
- A user-friendly interface ensures high engagement and adoption by non-technical users.

By achieving these metrics, the system will meet its goal of providing a reliable, efficient, and accessible sentiment analysis tool for businesses.

DATA

The data used for this project consists of customer product reviews, which are text-based and contain natural language expressions of sentiment. These reviews serve as the input data for the sentiment analysis application.

1. **Type of Data and Source**:
   - The data consists of textual product reviews that describe customers' opinions on various products.
   - For testing and demonstration purposes, manually created reviews (positive and negative) will be used, ensuring a balanced representation of sentiments.
   - If necessary, publicly available datasets like the **IMDB Reviews Dataset** or **Amazon Product Reviews** could be used for additional validation or expansion of the test set.
2. **Data Availability and Requirements**:
   - Currently, manually generated reviews are used to test the application. This provides enough data to demonstrate the functionality of the pre-trained sentiment analysis model.
   - Since a pre-trained model (BERT) is used, there is no immediate need for large-scale training data.
3. **Supervised Learning and Labels**:
   - The sentiment labels (positive or negative) are already embedded in the pre-trained BERT model, which has been fine-tuned on the **SST-2 dataset**. This dataset contains human-annotated reviews, ensuring high accuracy and consistency in the labels.
4. **Ensuring Label Accuracy and Consistency**:
   - Pre-trained models like BERT ensure high-quality and consistent labels since they are fine-tuned on professionally curated datasets.
5. **Privacy and Ethical Considerations**:
   - If real customer reviews are used, care must be taken to anonymize any personally identifiable information to protect user privacy.
   - Only publicly available reviews or synthetic data will be used, ensuring compliance with ethical and legal standards for data usage.
6. **Data Representation for the Model**:
   - The text reviews are directly processed by the BERT model, which converts them into embeddings (numerical representations of text) suitable for sentiment analysis.
   - No additional feature engineering or scaling is required, as the pre-trained model handles text tokenization, encoding, and analysis internally.

By leveraging pre-trained models and manually curated test data, this project minimizes the complexity of data collection and preparation while maintaining high accuracy and ethical standards.

MODELING

The project leverages pre-trained models for sentiment analysis to deliver accurate and efficient classification of customer reviews. The chosen model, **DistilBERT** (distilbert-base-uncased-fine tuned-sst-2-english), is a lightweight, fine-tuned version of **BERT** specifically designed for binary sentiment classification. This approach eliminates the need for extensive model training and provides reliable performance out of the box.

DistilBERT, developed by Hugging Face, is a distilled version of BERT, retaining approximately 95% of BERT's performance while being more efficient in terms of speed and resource usage. This model is highly suitable for the project due to its state-of-the-art accuracy and its capability to process raw text input without additional feature engineering. On the **Stanford Sentiment Treebank (SST-2)** dataset, the original BERT model achieves an accuracy of 92.7%, as documented by (Devlin et al., 2019). Given that DistilBERT retains over 95% of BERT's performance on tasks within the GLUE benchmark (which includes SST-2), we can infer an approximate accuracy range of 90-92% for DistilBERT on sentiment classification tasks.

For baseline performance, we consider random guessing, which provides a 50% accuracy baseline for a binary classification task(https://ieeexplore.ieee.org/document/7984735). In comparison, human-level performance on the SST-2 dataset exceeds 90% accuracy, providing a benchmark that confirms the reliability of DistilBERT's performance.

While simpler models like logistic regression or Naive Bayes could be explored for baseline comparisons, the use of a pre-trained transformer model negates the need for training or fine-tuning simpler models for production. DistilBERT's high performance and efficiency make it an ideal choice for this sentiment analysis application, providing a robust benchmark for evaluating the app.

DEPLOYMENT

The model has been deployed using Streamlit’s integrated deployment system. Predictions will be used to generate reports summarizing overall sentiment trends, helping businesses better understand audience opinions. To monitor and maintain the model, we’ll collect user feedback to identify prediction errors, track any recurring misclassifications, and periodically check for newer or improved versions of the model, allowing us to swap in an updated model as needed without requiring full retraining.

REFERENCES

Hugging Face Documentation on DistilBERT

- Provides insights into the development and performance retention of DistilBERT, highlighting its efficiency compared to BERT.
- Hugging Face, “DistilBERT Model Documentation.” Available at: https://huggingface.co/docs/transformers/model_doc/distilbert

Original BERT Research Paper

- Establishes BERT’s performance on the SST-2 dataset, achieving 92.7% accuracy. This serves as a benchmark for understanding DistilBERT’s inferred performance on similar tasks.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805. Available at: https://arxiv.org/abs/1810.04805

Baseline Accuracy in Binary Classification

- Discusses the concept of using random guessing as a baseline in binary classification tasks, relevant for setting performance expectations.
- Schein, A.I., et al. (2005). "A Comparison of Prediction Performance in Binary Classification Tasks." IEEE. Available at: https://ieeexplore.ieee.org/document/7984735

Streamlit Documentation:

- Offers insights into building interactive web applications, which has guided the implementation of the user interface.
- URL: https://docs.streamlit.io/