10/12/2024

# Sentiment Analysis Project on Amazon Product Reviews

## Innovators "Team"

Presented By:
Ahmed Mohamed Essam
Omar Tamer Salah El Din
Mostafa Mohamed Amin
Ahmed Tarek El Sayed Ghazy

# Table of Contents

# Project Responsibilities

- Mostafa Mohamed Amin Exploratory Data Analysis (EDA)
- Ahmed Tarek El Sayed Ghazy  Comparison of Machine Learning Models for Text Classification
- Omar Tamer Salah El Din Attention-Based NLP model.
- Ahmed Mohamed Essam Sentiment Analysis with DistilBERT on Azure and MLflow for Tracking and Model Lifecycle

# Introduction

This report presents a comprehensive analysis of sentiment in Amazon product reviews. The dataset comprises reviews spanning 18 years, including ~35 million reviews up to March 2013. The primary objective is to classify reviews into positive or negative sentiments and derive insights from the data.

# Data Collection and Preprocessing

## Data Overview

The Amazon reviews polarity dataset is constructed by categorizing review scores 1 and 2 as negative, and 4 and 5 as positive. Reviews with a score of 3 are ignored. The dataset includes:

- Polarity: 1 for negative and 2 for positive
- Title: Review heading
- Text: Review body

Each class has 1,800,000 training samples and 200,000 testing samples.

## Data Cleaning

- **Missing Data Check**: No missing values were found in the dataset, ensuring it is ready for analysis without requiring imputation or removal of data points.
- **Text Preprocessing**:
  - Text Cleaning: Removed special characters, punctuation, symbols, and irrelevant numbers.
  - Standardization: Converted text to lowercase and removed extra spaces.
  - Tokenization: Segmented text into individual words (tokens).
  - Optional Steps: Removing stop words, stemming or lemmatization, and creating n-grams.

# Exploratory Data Analysis (EDA)

## Basic Statistical Summary:

- Mean Rating: Provided an understanding of general customer satisfaction.
- Median Rating: Indicated whether the data is skewed.
- Mode Rating: Showed the most frequently given rating.

## Sentiment Analysis (Polarity Calculation):

- Sentiment Analysis with VADER: Calculated sentiment scores (positive, negative, neutral, and compound) for each review.
- Classifying Sentiments: Used compound scores to categorize sentiments into 'Positive', 'Negative', or 'Neutral'.

## Sentiment Distribution:

- Positive Sentiment Dominance: Majority of reviews are positive (1,421 counts).
- Neutral Sentiment: Few reviews are neutral (54 counts).
- Negative Sentiment: A small portion of reviews are negative (122 counts).

## Word Count and Length Analysis:

- Word Count: Number of words in each review.
- Character Count: Number of characters in each review.

## Word Cloud of Most Common Words:

- Common Words: Visualized frequently used words, excluding stop words.

## Distribution of Ratings:

- Histogram: Visualized the distribution of ratings, indicating polarized opinions.

## Outlier Detection using Cosine Similarity:

- Outcome: Identified reviews with very little similarity to others as outliers.

## Correlation Matrix:

- Features: Rating, Polarity, Word Count.
- Insight: High positive correlation between polarity and rating, no strong correlation between word count and rating.

## Summary of Key Insights

- Overall Sentiment: Majority of customers had a positive sentiment.
- Rating Polarization: Ratings were polarized, indicating varying customer expectations or inconsistencies in product quality.
- Common Themes: Frequently mentioned aspects of the product provided insights into key features and common concerns.
- Outliers and Extreme Cases: Identified unique experiences not reflected in the broader dataset.
- Correlations: Strong correlation between rating and polarity, no strong correlation between word count and rating.

# Comparison of Machine Learning Models for Text Classification

## Data Preprocessing

- Data Loading and Memory Usage: Loaded train and test datasets.
- Resampling: Balanced training dataset by downsampling.
- Text Cleaning and Lemmatization: Cleaned text data and removed stop words.
- Feature Engineering: Applied TF-IDF vectorization.

## Model Training and Evaluation

- Logistic Regression: Accuracy of 85%.
- Support Vector Machine (SVM): Best model with an accuracy of 86%.
- Random Forest: Accuracy of 83%.
- Gradient Boosting: Accuracy of 79%.
- AdaBoost: Accuracy of 81%.
- Extra Trees: Accuracy of 84%.
- XGBoost: Accuracy of 82%.
- CatBoost: Accuracy of 82%.
- Multinomial Naive Bayes: Accuracy of 83%.
- Voting Classifier (Ensemble): Accuracy of 84%.

## Hyperparameter Tuning

- Random Forest: Best parameters achieved a score of 82.83%.

## Visualization

- Confusion Matrices: Visualized performance of each model.
- Word Clouds: Visualized most frequent words for positive and negative classes.

## Model Comparison

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Logistic Regression | 85% | 0.85 | 0.85 | 0.85 |
| SVM | 86% | 0.86 | 0.86 | 0.86 |
| Random Forest | 83% | 0.83 | 0.83 | 0.83 |
| Gradient Boosting | 79% | 0.79 | 0.79 | 0.79 |
| AdaBoost | 81% | 0.81 | 0.81 | 0.81 |
| Extra Trees | 84% | 0.84 | 0.84 | 0.84 |
| XGBoost | 82% | 0.82 | 0.82 | 0.82 |
| CatBoost | 82% | 0.82 | 0.82 | 0.82 |
| Multinomial Naive Bayes | 83% | 0.83 | 0.83 | 0.83 |
| Voting Classifier | 84% | 0.84 | 0.84 | 0.84 |

## Best Model

- Support Vector Machine (SVM): Emerged as the best model with an accuracy of 86%.

# Attention-Based NLP model.

## Files and Descriptions:

- app.py: Streamlit web application for text classification.
- main.py: Command-line interface for text classification.
- model_handler.py: Loads pretrained DistilBERT model and handles text classification.
- predictor.py: Used for batch prediction and evaluation.

# Sentiment Analysis with DistilBERT on Azure

## Utilizing the Pretrained DistilBERT Model from AzureML Registry:

- Model: distilbert-base-uncased-finetuned-sst-2-english.
- Deployment: Low-code deployment with Azure Machine Learning Studio.
- MLflow-Ready Artifacts Tracking: Automatic tracking of model artifacts, environment configuration, and scoring script.
- Autoscaling for Deployment and Inference: Metric-based scaling, infrastructure options, and manual scaling.

## Azure Features vs. On-Premise

- Ease of Deployment: Quick and minimal configuration with Azure.
- Scalability: Dynamic autoscaling options with Azure.
- Maintenance and Updates: Seamless maintenance and updates with Azure.

# MLflow for Tracking and Model Lifecycle

The emergence of MLflow has significantly enhanced the efficiency of managing and tracking machine learning workflows. This document focuses on using MLflow in machine learning projects, leveraging various machine learning models to monitor, log, and evaluate experimentations.

## MLflow Main Features

- **Tracking**: MLflow's tracking component to log and query experiments, including code, data, configuration, and results.
- **Projects**: A standard format for packaging reusable data science code.
- **Registry**: A central repository to manage the lifecycle of ML models.
- **Models**: MLflow’s model component facilitates managing and deploying machine learning models.

## Definitions and Main Functionalities

- **Experiments**: Group of related ML runs.
- **Runs**: Instances of model training processes, including data, configuration, and results.
- **Model Parameters**: Hyperparameters used during the training process.
- **Model Metrics**: Performance indicators like accuracy, precision, recall, and F1-score.
- **System Metrics**: Resource usage details such as CPU and memory consumption.
- **Artifacts**: Files generated by a run, including models, plots, and logs.
- **Models Registry**: A centralized model store to keep track of versions, tags, aliases, and descriptions for deployed models.

## MLflow Benefits

- **Reproducibility**: Ensures experiments are repeatable.
- **Experimentation Management**: Simplifies tracking and comparison of multiple runs.
- **Model Lifecycle Management**: Facilitates model versioning and deployment.
- **Integration**: Seamlessly integrates with various platforms and tools like Azure ML, Kubernetes, and more.

## Integrating MLflow with Azure Machine Learning Studio

Integrating MLflow with Azure Machine Learning Studio enhances collaboration and deployment capabilities. It can log experiments in Azure ML, access them via MLflow, and leverage Azure’s robust infrastructure for training and deployment. This integration allows for seamless transitions from experimentation to production, ensuring scalability and reliability.

## Summary

This project utilizes MLflow to streamline the sentiment analysis workflow. By leveraging MLflow's tracking, projects, registry, and model management features, it can effectively monitor, log, and manage various machine learning models.

# Conclusion

The sentiment analysis project on Amazon product reviews conducted by Team Innovators has successfully achieved its objectives, providing valuable insights into customer sentiments and preferences. By utilizing a comprehensive dataset of reviews spanning 18 years, we effectively classified reviews into positive and negative sentiments, offering a clear understanding of customer satisfaction and product performance.

## Key Achievements:

1. Data Preparation and Cleaning: Ensured the dataset was free of missing values and performed extensive text preprocessing, including cleaning, tokenization, and standardization, establishing a robust foundation for accurate sentiment analysis.
2. Sentiment Analysis: Employed the VADER sentiment analysis tool to calculate sentiment scores and classify reviews into positive, negative, and neutral categories. The analysis revealed a predominance of positive sentiments, indicating general customer satisfaction.
3. Statistical and Visual Insights: Conducted basic statistical summaries, word count analysis, word clouds, and distribution plots to identify key themes and patterns in customer reviews. The polarized distribution of ratings highlighted varying customer expectations and product quality.
4. Machine Learning Model Comparison: Evaluated multiple machine learning models, including Logistic Regression, SVM, Random Forest, Gradient Boosting, and others. The Support Vector Machine (SVM) with an RBF kernel emerged as the best-performing model with an accuracy of 86%.
5. Deployment and Application: Developed a sentiment analysis application using a pretrained DistilBERT model, demonstrating the efficiency of Azure Machine Learning Studio for model deployment, autoscaling, and maintenance.

## Insights and Recommendations:

- Overall Sentiment: The majority of reviews were positive, aligning with high average ratings, suggesting general customer satisfaction.
- Rating Polarization: The polarized distribution of ratings indicates intense customer opinions, either highly positive or negative, reflecting varying expectations or product quality inconsistencies.
- Common Themes: Frequently mentioned words and themes provided insights into customer priorities and concerns, useful for improving product descriptions and addressing common issues.
- Outliers and Extreme Cases: Identified outlier reviews to understand unique customer experiences not captured through average sentiment or rating scores.

- Model Deployment: The SVM model with an RBF kernel is recommended for deployment due to its superior performance. The ensemble Voting Classifier also showed promising results and can be considered as an alternative.

# Future Work:

- Hyperparameter Tuning: Further tuning of model hyperparameters to enhance performance.
- Feature Engineering: Exploration of additional feature engineering techniques to improve model accuracy.
- Continuous Monitoring: Implementation of a system for continuous monitoring and periodic retraining of the model with new data to ensure robustness and relevance.