Data Mining Project Report (Team 10): Sentiment Analysis on Airline Tweets

Yuk Shan Wai (2024895), Jui-Ling Wu (1980658), Hyewon Kang (1980634), Yung Chi Ngai (2028153), Sindi Kallcishta (1979899) and Sueda Sogutlu (1978962)

University of Mannheim, 68131 Mannheim, Germany

Abstract. This project aims to help US airlines learn about customer sentiment by analyzing Twitter posts related to airline experiences. Using sentiment analysis techniques and algorithms such as Multinomial Naive Bayes, Logistic Regression, Random Forest, SVM and BERT, the project categorizes tweets into positive, neutral and negative sentiment, providing actionable insights to improve customer satisfaction.

1 Introduction

The development of technology and the exponential growth of social media and review websites have provided customers with an increasing number of channels for expressing their feelings online. This has created a significant challenge for companies in the current business environment. As the transparency of reviews has increased, companies must take ad-hoc actions to address or resolve customer complaints more effectively and efficiently. As a result, US airlines have turned to Twitter for gathering customer reviews. The real-time updates and global user base of Twitter ensure that airlines receive the most recent and genuine feedback from a diverse audience worldwide.

In the domain of sentiment analytics, the application area revolves around the B2C industry, specifically focusing on understanding and analyzing customer sentiment and feedback regarding the whole customer experience journey. This encompasses a broad range of aspects, including service experiences, customer service interactions, amenities, safety measures, and overall satisfaction levels.

The goal of this project is to assist US airlines in understanding customer sentiment by analyzing Twitter posts related to airline experiences. Sentiment analysis techniques are employed, utilizing algorithms such as Multinomial Naive Bayes, Logistic Regression, Random Forest, SVM, and BERT, to categorize posts into positive, neutral, and negative sentiments. The insights derived from this analysis are provided to airlines, enabling them to predict customer behavior and enhance their services accordingly. This approach empowers airlines to proactively address customer concerns, ultimately leading to improved customer satisfaction.

2 Dataset Profile

The dataset utilized in this project was sourced from Kaggle and comprises Twitter posts addressing the challenges encountered by major U.S. airlines. This collection

of Twitter data was gathered in February 2015, encompassing a total of 14,640 rows and 9 columns.

To facilitate data classification, we employed the LabelEncoder algorithm, which categorized the tweets into three groups with the following numerical values: Negative: 0, Neutral: 1, Positive: 2.

While there are some missing values in the 'negativereason', 'negativereason_confidence', and 'user_timezone' columns, these do not significantly impact the sentiment analysis. The 'negativereason' and 'negativereason_confidence' columns have missing values because not all users expressed negative sentiments. Similarly, the missing values in the 'user_timezone' column do not affect the overall sentiment analysis.

2.1 Target Data Distribution

The dataset is distributed across the sentiment categories as follows and Figure 1:

- Negative tweets: 9,178, Neutral tweets: 3,099, Positive tweets: 2,363

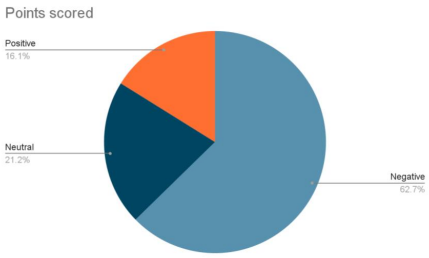

Figure 1: Data distribution structure

2.2 Negative Tweets Reason Distribution

In order to better understand the dataset, we also analyzed the reason for the negative tweets and it has the following reasons: Customer Service Problem, Late Flight, Can't Tell, Canceled Flight, Lost Luggage, Bad Flight, Flight Booking Problems, Flight Attendant Complaints, Long Lines, Damaged Luggage and the distribution is shown in Figure 2.

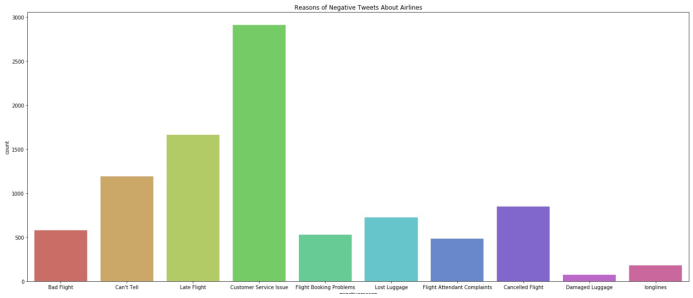

Figure 2: Reasons of Negative Tweets about Airlines

2.3 Dataset Types and Attributes

The dataset types and attributes are listed in Table 1.

Name	Type	Explanation
Tweet_id	float64	A unique identifier for each tweet.
Airline_sentiment	object	The sentiment of the tweet (positive, negative, neutral).
Airline_sentiment_confidence	float64	The confidence score of the sentiment classification.
Negativereason	object	The reason for negative sentiment.
Negativereason_confidence	float64	The confidence score of the negative reason classification.
Name	object	The name of the user who posted the tweet.
Retweet_count	int64	The number of times the tweet was retweeted.
text	object	The content of the tweet.
tweet_created	object	The timestamp when the tweet was created.
user_timezone	object	The timezone of the user who posted the tweet.

Table 1: Dataset types and attributes

3 Preprocessing and Mining

3.1 Data Preprocessing

The preprocessing stage is a crucial step in developing an airline sentiment analysis model from Twitter data. This phase involves transforming raw social media text into structured, model-ready data using Python and its powerful libraries like Pandas and Scikit-Learn. The key preprocessing steps include data formatting and type adjustment, text data preprocessing, label and date handling, and additional preparations.

The preprocessing steps we used in our code are the shown below:

- **Convert to lowercase:** This converts all characters in the text to lowercase. It's important for standardizing the text data because NLP models often treat uppercase and lowercase characters differently, which can lead to sparsity in the vocabulary.
- **Remove URLs:** URLs typically don't carry sentiment-bearing information and can be removed.
- **Remove special characters and punctuation:** This regex pattern removes all non-alphanumeric characters (special characters) from the text and replaces them with a space. Punctuation marks are included in this category.
- **Tokenization:** This function tokenizes the text, splitting it into individual words or tokens. This step is crucial for further analysis as it breaks down the text into its basic units.
- **Remove stopwords:** Common words like "and", "the", "is", etc., which don't carry significant sentiment, are removed.
- **Lemmatization:** This step reduces words to their base or root form (lemmas). It helps in standardizing words so that variations of the same word are treated as the same token. For example, "running", "runs", and "ran" would all be converted to "run".

- **Join tokens back into text:** Finally, the preprocessed tokens are joined back together into a single string, separated by spaces. This is the format commonly used for further analysis or modeling.

By systematically processing the data through these steps, we ensure that it is optimally prepared for the modeling stage, enhancing the accuracy and efficiency of the sentiment analysis. This preparation enhances the data quality for when machine learning algorithms are deployed, facilitating a robust model training phase, and ultimately leading to more reliable sentiment analysis outcomes tailored for the dynamic and occasionally chaotic nature of data derived from social media platforms like Twitter.

3.2 Evaluation Setup

After completing the initial preprocessing steps on our dataset, which involved cleaning and preparing the text data, we proceeded to divide it into training and testing sets. Subsequently, we encoded the target variable and converted the text data into numerical features suitable for machine learning model training. To ensure reproducibility, we employed an 80-20 split for training and testing, respectively, with a fixed random state. Following this split, we utilized a TF-IDF vectorizer to transform the text data into numerical representations. Initially, we applied this vectorization process to the preprocessed text data within the training set, generating TF-IDF features. Later, we applied the same vectorization technique to the preprocessed text data within the test set, using the vectorizer previously fitted on the training data. Finally, we utilized this processed data to train and evaluate various sentiment analysis models.

To evaluate the performance of the models we are using in this project we used various matrixes such as accuracy,ROC-AUC, precision-recall curve, confusion matrix and classification report.The results obtained from these evaluations provide insights into how well the classifier performs on the given dataset.

4 Model and Parameter Setting

4.1 Model Selection

Based on the objective and the dataset, we aim to implement various supervised learning models to classify unseen data effectively and achieve the goals of this study. Additionally, the performance of each model will be visualized to understand their effectiveness better. In this section, we list the models implemented in this study:

i. Multinomial Naïve Bayes
A probabilistic classifier based on Bayes' theorem, particularly well-suited for text classification problems like sentiment analysis. It assumes that the features (words) are conditionally independent given the class, which simplifies computation and often performs well with text data (McCallum & Nigam, 1998; Rennie et al., 2003).

ii. Logistic Regression
A linear model for binary classification that can be extended to multiclass problems using techniques like one-vs-rest or multinomial logistic regression. It's straightforward to implement and provides probabilities for class membership, making it useful for sentiment analysis (Cox, 1958).

iii. Random Forest
An ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification tasks. It is robust to overfitting and can handle large datasets with high dimensionality, making it suitable for sentiment analysis tasks (Breiman, 2001; Liaw & Wiener, 2002).

iv. Support Vector Machine (SVM)
A supervised learning model that finds the optimal hyperplane separating different classes in the feature space. SVMs are effective in high-dimensional spaces and are often used for text classification due to their robustness and accuracy (Cortes & Vapnik, 1995; Joachims, 1998).

v. BERT (Bidirectional Encoder Representations from Transformers)
A deep learning model that captures the context of a word from both directions (left-to-right and right-to-left), making it highly effective for various natural language processing tasks, including sentiment analysis. Its ability to understand context at a deep level allows for superior performance compared to traditional models (Devlin et al., 2018; Qiu et al., 2020).

4.2 Hyperparameter Optimization

Hyperparameters are preset parameters that influence a model's performance but are not learned during training. Proper tuning of these parameters can enhance the model's generalization to unseen data. The team used GridSearchCV for hyperparameter tuning; and employed macro averaging for performance evaluation due to the dataset imbalance (Bergstra & Bengio, 2012; Pedregosa et al., 2011).

1. Multinomial Naïve Bayes

   A range of positive values (0.01, 0.1, 0.5, 1.0, 2.0, 5.0, 10.0) is tested as alpha (smoothing parameter) to prevent zero probabilities for unseen words.

   Best parameters: {'alpha': 0.1}

2. Logistic Regression A range of C (regularization parameter) is tested to prevent overfitting by penalizing large coefficients in the model.

   Best parameters: {'C': 10}

3. Random Forest Using GridSearchCV, we explore different combinations of parameter settings, aiming to find the best setup. After training the model on the training data, we identify the best combination of settings and print their F1 score. With these optimal settings, we create a new RandomForestClassifier and train it on the data.

   Best parameters: {'max-depth': None, 'min-samples-leaf': 1, 'min-samples-split': 2, 'n-estimators': 300}

4. Support Vector Machine (SVM) The regularization parameter C balances between maximizing the margin and minimizing classification errors, while the kernel choice influences the decision boundary and model complexity.

   Best parameters: {'C': 1, 'gamma': 'scale', 'kernel': 'linear'}

5. BERT Due to computational constraints, performing hyperparameter optimization directly within the code may not be feasible. However, we can outline some hypothesis steps to guide the selection of hyperparameters. Here are some considerations for each hyperparameter

   - Learning Rate: Controls the step size for updating model parameters. Common values range from 1e-5 to 5e-5.

   - Batch Size: Determines the number of samples processed before updating parameters. Typical values are 16, 32, or 64.

   - Number of Epochs: Defines the complete passes through the training dataset. BERT tends to overfit with too many epochs, so typical values range from 2 to 4.

5 Evaluation and Result

5.1 Model Performance Metrics

Throughout the course of our sentiment analysis study, we thoroughly investigated a number of machine learning algorithms in an effort to determine the best method for categorizing the sentiments that users of Twitter comments express. The classification problem was multi-class because our dataset included three classes: neutral, positive, and negative sentiments.

We used a variety of algorithms, each of which provided a different approach to sentiment classification. Among these algorithms were BERT, Random Forest, Support Vector Machine (SVM), Multinomial Naive Bayes, and Logistic Regression. Each algorithm was selected based on its potential to handle multi-class classification tasks and its suitability for our dataset.

Using the standard sklearn function "train_test_split", we divided our dataset into training and testing sets, allocating 80% of the data for training and 20% for testing, to ensure an unbiased evaluation. With the help of this partitioning technique, we were able to train our models on a sizable enough amount of the data while keeping another for objective assessment.

We gave special attention to evaluation measures that provide information about model performance across all classes, since our classification problem is multi-class and our dataset naturally exhibits class imbalances. Accuracy, precision, recall, F1-score (weighted and macro averages), and ROC-AUC were among the important measures taken into account.

Algorithm	**Train Score**		**Test Score**	
	Accuracy	**ROC-AUC**	**Accuracy**	**ROC-AUC**
Multinomial Naïve Bayes	0.887	0.978	0.745	0.872
Logistic Regression	0.964	0.992	0.777	0.883
Random Forest	0.995	0.999	0.762	0.867
SVM	0.899	0.968	0.786	0.892
BERT	0.969	0.61	0.81	1

Table 2: Evaluation of models on train and test data

5.2 Result

Algorithm		Precision	Recall	F1-Score
Multinomial Naïve Bayes	Class 0	0.76	0.96	0.85
	Class 1	0.68	0.35	0.46
	Class 2	0.73	0.44	0.55
	Macro Avg	**0.72**	**0.58**	**0.62**
	Weighted Avg	**0.74**	**0.75**	**0.72**
Logistic Regression	Class 0	0.84	0.9	0.87
	Class 1	0.6	0.55	0.57
	Class 2	0.73	0.62	0.67
	Macro Avg	**0.72**	**0.69**	**0.7**
	Weighted Avg	**0.77**	**0.78**	**0.77**
Random Forest	Class 0	0.78	0.94	0.85
	Class 1	0.64	0.43	0.51
	Class 2	0.78	0.51	0.62
	Macro Avg	**0.73**	**0.63**	**0.66**
	Weighted Avg	**0.75**	**0.76**	**0.74**
SVM	Class 0	0.82	0.92	0.87
	Class 1	0.65	0.52	0.58
	Class 2	0.76	0.61	0.68
	Macro Avg	**0.74**	**0.68**	**0.71**
	Weighted Avg	**0.78**	**0.79**	**0.78**
BERT	Class 0	0.88	0.89	0.89
	Class 1	0.66	0.52	0.58
	Class 2	0.67	0.82	0.74
	Macro Avg	**0.74**	**0.74**	**0.73**
	Weighted Avg	**0.8**	**0.81**	**0.8**

Table 3: Evaluation on models with each class on test data
(Class 0 - Negative, Class 1 - Neutral, Class 2 - Positive)

BERT was found to regularly beat competing algorithms across numerous assessment metrics after a thorough evaluation. The most advanced language representation model, BERT, showed remarkable capacity to convey the nuanced emotions found in Twitter comments.

With an astounding AUC-ROC of 90% on the test dataset, BERT proved to be the best algorithm in distinguishing between the classes. Furthermore, BERT demonstrated its ability to achieve a balanced performance across all sentiment classes with a weighted average of 0.8 for its F1-score.

As a result, our sentiment analysis project demonstrated how remarkably effective BERT is at identifying the emotions represented in Twitter comments. BERT is the algorithm of choice for sentiment classification tasks because of its superior performance across a range of evaluation criteria, such as accuracy, F1-score, and ROC-AUC. This is especially true in situations when there are imbalances between classes and several classes of sentiment. In order to obtain nuanced and precise sentiment classification findings, this emphasizes how crucial it is to use sophisticated language representation models in sentiment analysis applications.

6 Error Analysis

6.1 Observations and Initial Insights

A few types of common errors in the model's performance shown below:

False Negatives (FN):
For example, sentiments that were expressed negatively in an indirect way such as "The flight was as expected, nothing special" would often be classified wrongly as neutral sentiments rather than negative ones. However, it tends to be a negative sentiment. There were 70 negative sentiments that the model did not capture out of 1000 possible negative sentiments, having a 7% FN rate.

Underperforming Classes:
The imbalance among classes in the training data leads to a bias towards the majority class (negative), resulting in notably poorer performance for the neutral and positive classes, as depicted in Table 3. Potential remedies for this issue involve adjusting class weights, a technique we employed by using weighted averages during model construction, or generating synthetic data.

6.2 Examples and Observations:

A statement like "The flight was as expected, nothing special" was often misclassified as neutral instead of negative. These errors commonly occurred as the model struggled with mixed feelings or understated dissatisfaction Feedback such as "Not too bad I guess, could be better" and "It works, but I'm not impressed with the performance" highlighted the dissatisfaction from customers, which the model frequently overlooked.

False Positives (FP): Positive sentiments were occasionally over-identified.

A statement such as "Great price but the product broke the first day" was wrongly labeled as positive just because of the phrase "Great price." This gave a misleading impression of higher customer satisfaction than reality.

Positive words sometimes caused the system to label the sentence as positive incorrectly, leading to incorrect categorizations.

Sarcasm or irony like "Oh great, another software update that fixes nothing" was misinterpreted as genuine praise on literally meaning. This misinterpretation was a significant source of false positives.

The confusion matrix provided ([1705, 142, 42], [231, 292, 57], [98, 56, 305]) indicates several false positives and negatives, which are the main reasons for the low accuracy rates.

6.3 Challenges Encountered

- Data Complexity: The inherent complexity and subtleties of human language pose significant challenges for sentiment analysis. Cultural diversity and context-specific nuances often elude AI, leading to issues like false negatives and false positives. These complexities make it difficult for models to accurately interpret the true meaning behind certain phrases.

- Sarcasm and Mixed Sentiments: Detecting sarcasm and mixed sentiments is particularly challenging, as it requires more than just a literal interpretation of the text. The nuanced nature of sarcasm often results in misclassification, demanding more sophisticated models capable of understanding context deeply.

- Model Limitations: Although the Random Forest Classifier (RFC) is highly adaptable and robust, it struggles to detect subtle language cues. It is also prone to overfitting, particularly when applied to unseen data. This limitation affects its generalization performance.

- Computational Limitation: The use of complex algorithms like Support Vector Machine (SVM) and BERT introduces significant computational demands, making the training process time-consuming. To mitigate this, we reduced the parameter search space and used fewer folds in cross-validation. Despite these adjustments, computational efficiency remains a challenge.

6.4 Potential Improvements

6.4.1 Enhancing Feature Engineering

By installing Incorporate advanced NLP techniques like part-of-speech tagging and context-aware sentiment analysis, which can be used to improve detection of subtle linguistic cues and enhance overall contextual understanding.

6.4.2 Model Parameter Tuning

By adjusting the number of decision trees and their depth,therefore we can optimize the RFC's balance between bias and variance, reducing overfitting and improving generalization.

6.4.3 Model Training

By regularly updating the training data with new, annotated examples, particularly those that highlight previous errors.It is therefore to continuous refinement of the model's algorithms to enhance adaptability to new patterns and wordings with inner meaning.

6.4.4 Addressing Imbalanced Data

By using data augmentation techniques to generate more balanced datasets, such as synthetic data generation or oversampling minority classes. In addition, implementing cost-sensitive learning algorithms that assign higher penalties for misclassifying underrepresented classes.

7 Conclusion

In this project, our sentiment analysis on airline tweets aimed to assist US airlines in understanding customer sentiment by employing various machine learning algorithms. Through the application of various machine learning algorithms including Multinomial Naïve Bayes, Logistic Regression, Random Forest, Support Vector Machine (SVM), and BERT, we conducted a comprehensive sentiment analysis on a dataset comprising tweets about major U.S. airlines. We preprocessed the data, optimized model parameters, and evaluated model performance using a range of metrics

including accuracy, precision, recall, F1-score, and ROC-AUC. Through rigorous evaluation and analysis, we found that BERT consistently outperformed other algorithms demonstrating superior accuracy and balanced performance across sentiment classes.

However, our evaluation also highlighted challenges and areas for improvement. We observed common challenges such as data complexity, sarcasm detection, and computational limitations. The imbalance among sentiment classes in the training data also posed challenges, leading to biased predictions favoring the majority class. To address these challenges and further enhance the performance of sentiment analysis models, we proposed several potential improvements like enhancing feature engineering, model parameter tuning, model training and addressing imbalanced data.

Overall, our project underscores the importance of leveraging advanced machine learning techniques, particularly BERT, in sentiment analysis tasks to accurately capture and understand customer sentiments expressed on social media platforms like Twitter. By understanding customer sentiments more accurately, US airlines can proactively address issues, enhance customer satisfaction, and ultimately improve their overall service quality and reputation in the highly competitive airline industry. As the field of sentiment analysis continues to evolve, integrating sophisticated algorithms and refining methodologies will be key to unlocking deeper insights and driving continuous improvement in customer experience management.

References

[1] McCallum, A., & Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. In AAAI-98 workshop on learning for text categorization (pp. 41-48).

[2] Rennie, J. D. M., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In ICML (Vol. 3, pp. 616-623).

[3] Cox, D. R. (1958). The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), 20(2), 215-232.

[4] Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

[5] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.

[6] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20, 273-297.

[7] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Springer, Berlin, Heidelberg.

[8] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

[9] Qiu, X., Sun, T., Xu, Y., Shao, Y., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. Science China Technological Sciences, 63, 1872-1897.

[10] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(2).

[11] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.