

**A Consultation Report on Traditional Machine Learning for Shopper
Review Sentiment Analysis**

Deep Patel (2504308)

APPLICATION OF MACHINE LEARNING (ACCA7025)



Table of Contents

Abstract	2
1. Introduction.....	2
2. Related Work.....	3
3. Workflow Methodology	5
3.1 Data Collection and Labelling	5
3.2 Text Pre-processing	5
3.3 Feature Extraction.....	6
3.4 Model Training and Selection.....	6
3.5 Hyperparameter Tuning	6
3.6 Evaluation Metrics.....	7
4. Discussion and Recommendations	8
5. Conclusion.....	9
References	11

Abstract

Customer sentiment analysis can now be considered one of the most significant marketing tools which marketing organisations ought to employ in order to analyse consumer feedback at mass level. The authors in the present report present a detailed discussion on traditional machine learning (ML) solutions that could be implemented to classify sentiments of shopper reviews, and the objective is to suggest a computationally effective and efficient solution to a marketing firm with minimal hardware capabilities. Having done a systematic literature search concerning the subject matter, along with the lexicon-based methods, Naive Bayes, Support Vector Machines (SVM) and the Logistic Regression technique, this report proposes the workflow based on the TF-IDF feature extraction and Logistic Regression as the primary classifier. The proposed pipeline consists of data pre-processing, feature engineering, L2 regularised model training, and cross-validated analysis of the accuracy, F1-score, precision, recall, and ROC-AUC measure. Empirical evidence exists in literature to substantiate this recommendation since it can have a high classification performance with low computation overhead that is appropriate to client limitations. The report concludes that the proposed pipeline is not only efficient in the context of the requirements of the current deployment, but its scale is also geared towards the gradual fulfilment of the requirements of the client as data assets of the client grow.

1. Introduction

The rapid rate at which e-commerce sites grow has produced a new kind of consumer-generated text, as never before, in the shape of product reviews, commentary on social media, and posts on forums. The possibility to determine automatically the sentiment coded in such information, to classify a specific review as positive, negative or neutral is highly commercially significant to marketing organisations [1]. Sentiment analysis is a sub-subdiscipline of Natural Language Processing (NLP) also referred to as opinion mining, which enables organisations to transform unstructured textual data into business intelligence that can be used to make strategic decisions in product development, customer relationship management and brand positioning [2].

This consultation has also been contacted by a marketing firm that requested them to advise on how to implement ML models to automatize the process of sentiment classification of shopper reviews. The need to lower the cost of computation, which is among the fundamental operation constraints, directs the exploration to the less resource-intensive and traditional types of ML approaches rather than resource-consuming deep learning models. This weakness can be attributed by the fact that a growing body of literature has revealed that under the right conditions, classical classifiers can be utilized to perform competitive performance in sentiment classification problems and simultaneously can be computationally infeasible [3].

The use of the classic supervised learning approach is thus not a tradeoff but a well-supported strategic decision to an organisation that is resource conscious.

This report has four objectives: (i) to critically review the available literature on traditional ML approaches to sentiment analysis; (ii) to describe a comprehensive and reproducible ML pipeline that meets the needs of the client; (iii) to suggest a suitable classifier algorithm with empirical evidence in similar use-cases in the literature; and (iv) to specify the evaluation metrics and fine-tuning strategies to achieve reliable and commercially viable classes. The report is structured as follows: Section 2 is a survey of the related work; Section 3 outlines the proposed workflow methodology; Section 4 discusses the proposed workflow and gives a recommendation; finally, Section 5 draws conclusions and provides the implications to the client organisation.

2. Related Work

The history of automated sentiment analysis has been long and well-charted, and the seminal study by Pang and Lee [1] defining sentiment classification as a separate task in NLP and demonstrating the effectiveness of supervised learning to movie review data. They compared their relative performance in Cornell Movie Reviews dataset using Naive Bayes, Maximum Entropy and SVM classifiers and discovered that SVM was always more successful than other two, achieving the accuracy of 83 percent. This result was observed to be highly influential and its most significant finding that discriminative classifiers outperform generative ones in sentiment tasks but not in generative ones has since been replicated and extended to many tasks [2], [5].

Blitzer, Dredze and Pereira [5] did not limit themselves to single topic corpora or sentiment classification by suggesting Multi-Domain Sentiment Data which is composed of Amazon product reviews of books, DVDs, electronics and kitchen appliances. Their experiment was concerned with the issue of domain adaptation - the tendency of a classifier that has been trained to deal with a specific type of product to perform worse when presented with another type of product - and it can be directly applied to the case of the client, where the products the reviews may refer to can be of various types. The experiment demonstrated that the domain shift effects were greatly minimized with the aid of the carefully selected TF-IDF weight features, i.e. unigrams and bigrams. This is a first-hand guiding principle on the feature engineering strategy that is recommended in this report.

With regard to the significance of feature representation Joachims [3] introduced among the first theoretically oriented studies of SVM in text categorisation, whereby the high-dimensional sparse TF-IDF vectors form the input to SVMs with linear kernels. Joachims argued that dimensionality of text feature spaces is in fact a desirable attribute because it increases the chances of linear separability of linear SVMs. This conceptual understanding forms a principled rationale of the

application of linear classifiers to text-based sentiment problems and has since turned out to be a cornerstone argument in the text classification literature. Kiritchenko, Zhu and Mohammad [7] made similar discoveries of Logistic Regression under similar circumstances further claiming the case of linear models.

One of the earliest and most commonly used methods of text classification, based on the probabilistic model outlined by McCallum and Nigam [12], is Naïve Bayes classifiers. Naive Bayes Multinomial Naive Bayes models, which approximate term frequency counts under the conditional independence assumption when the class label is known, have been shown to be effective at sentiment tasks, despite their simplification that the data is bag-of-words or TF-IDF, when trained on bag-of-words or TF-IDF representations [9]. Naive Bayes and Maximum entropy classifiers to Twitter sentiment data were used by Go, Bhayani and Huang [4], and they found that the working accuracy of Naive Bayes was approximately 81.3 per cent on binary classification and the training time was much lower than other classifiers. This efficiency benefit is especially relevant in the current context, where minimisation of the computational resources is a client requirement.

The Logistic Regression has been severely tested in binary and multi-class as a sentiment classifier. Kiritchenko et al. [7] used L2-regularised Logistic Regression as the core of a feature-based sentiment system of short informal texts, showing that regularisation provides good generalisation in high-dimensional sparse feature spaces. Their system that won SemEval had stressed on the significance of concentration as far as feature engineering that incorporates sentiment lexicon scores, negation management and n-gram features as well as the use of Logistic Regression that will lead to the states of art in the case of traditional classifiers. The practical advantage of the resulting model whereby individual feature coefficients are a direct mapping of sentiment-bearing terms was discovered to be an interpretability advantage over less interpretable models.

Maas et al. [8] presented the IMDB Large Movie Review Dataset that has gained popularity as a benchmark on binary sentiment classification and provided a standard point of cross-study comparisons. They tested it and found that a bag-of-words Logistic Regression model using unigrams and bigrams achieved an 88.89% accuracy, which was a high upper limit of the traditional methods on large-scale binary sentiment tasks. This value is noteworthy in that the eventual improvement of far more complex models on top of well-tuned Logistic Regression is minor in a series of subsequent studies, which argues in favor of the claim that extreme complexity of models beyond Logistic Regression may be unjustified in the current context.

The other unsupervised paradigm, as surveyed by Taboada et al. [10], is lexicon-based in which the documents are rated on sentiment by adding the polarity values of a pre-trained dictionary of polarity, e.g., SentiWordNet. Although such approaches do not need any form of labelled training data - which seems to be a benefit in the context of resource-constrained deployments - their performance was always worse than supervised classifiers in a variety of domains, and the accuracy

was generally 5-15 percentage points worse than SVM or Logistic Regression baselines. The observation provides a clear empirical basis of why supervised learning is a viable option in the present application, particularly since star-rating annotations that are frequently present in conjunction with shopper review datasets can provide a viable and cheap labelling system [2], [5].

Overall, the literature offers a consistent set of findings: of the traditional ML classifiers, the ones that show a consistent high performance in sentiment classification tasks are the Logistic Regression and the Linear SVM, with Naive Bayes being a computationally-efficient alternative with a bit worse performance. Unigram and bigram n-gram TF-IDF comes out as a leading and most validated feature representation strategy in these studies [1], [3], [7]. The results are used directly to inform the workflow and algorithm suggestion that is discussed in the subsequent sections.

3. Workflow Methodology

The suggested ML pipeline will be built according to the typical supervised learning cycle explained by Jurafsky and Martin [9] adjusted to the needs of sentiment classification of shopper reviews. This pipeline consists of six sequential stages: (i) collection and labelling of data; (ii) text pre-processing; (iii) feature extraction; (iv) model training and comparative selection; (v) hyperparameter tuning and (vi) evaluation. The pipeline has a schematic overview in Figure 1 of the Appendix. There were three primary steps of data collection and labelling. The data will be collected and labelled using the 3.1 Data Collection and Labelling method.

3.1 Data Collection and Labelling

This begins with a labeled set of shopper reviews acquired. In practice, textual review sites usually co-rate textual reviews with some form of star rating (e.g., 1-5) that they may use as proxy labels to classify sentiment with minimal manual labels [2]. A typical heuristic is to treat a review with a rating of 4 or 5 as positive and a review with a rating of 1 or 2 as negative, with 3-star reviews either ignored, or used as a third category called neutral in ternary systems [5]. Publicly available benchmarks like the Amazon Product Reviews corpus and the IMDB Large Movie Review Dataset [8] would be a good source of well-labeled corpora that can be prototyped and then combined with the proprietary review data of the client in order to develop the first model.

3.2 Text Pre-processing

The raw review text should go through a standardised pre-processing sequence to de-noise and normalise the input representation. The following steps are performed in the following order (i) lower-casing all the tokens to get vocabulary uniformity; (ii) removal of HTML entities, URLs, punctuation characters and special characters; (iii) tokenisation, the process of splitting text into single word tokens;

(iv) stop-word elimination, the removal of the high frequency, functional words, such as the, is, and so forth, which carry little sentiment signal [9 Taboada et al. [10] showed that this straightforward shortcut is significant in enhancing the performance of classifiers as it maintains the semantic inversion that is brought about by negation. All the pre-processing steps are carried out with the help of the NLTK and scikit-learn libraries [6].

3.3 Feature Extraction

The information in the form of text undergoes pre-processing and is then transformed into a numerical feature data that can be used in the ML algorithms. The main feature extraction mechanism that will be used is weighting mechanism which is Term Frequency-Inverse Document Frequency (TF-IDF) [3]. TF-IDF places more emphasis on the terms within a particular document that are frequent but are also rare in corpus, and so concentrate on discriminative sentiment-carrying vocabulary but prevent frequent and uninformative words. The range of unigrams and bigrams (n-gram range: [1, 2]) are used to produce the feature matrix, two-word sequences in which two-word sequences are empirically shown to improve sentiment classification by representing shorter syntactic patterns, such as not good or highly recommended [1], [5]. The feature space size is kept at 50,000 features selected by decreasing TF-IDF score to manage memory usage and is in the computational reach of the client [6].

3.4 Model Training and Selection

Stratified 5-fold cross-validation is used to divide the data to obtain a balanced representation of the classes of all the training and validation folds [9]. The process is chosen over a simple train-test split since it yields a more statistically robust and less prone to variance estimation of generalisation performance, which is particularly true in the case of a moderately-sized dataset. This cross-validation plan trains and evaluates three candidate classifiers: (1) Multinomial Naive Bayes, which is also a computationally minimal baseline; (2) Linear SVM with a hinge loss (implemented with scikit-learns LinearSVC); and (3) Logistic Regression with L2 regularisation. Pedregosa et al. [6] proposed that comparative method that empirically identifies the best performing model on the target evaluation metrics and makes a final choice and implementation.

3.5 Hyperparameter Tuning

After determining the right algorithm through comparative analysis, the hyperparameter optimisation is done with the assistance of the Grid Search with cross validation (GridSearchCV). In the case of Logistic Regression, the main hyperparameter is the regularisation strength C, which determines the trade-off between model complexity and minimisation of the training error: the smaller the C, the stronger the regularisation and the lower is the overfitting, whereas the larger C is, the closer the model fits the training data [7]. A logarithmic grid of search in C values [0.001, 0.1, 0.1, 1, 10, 100], is considered and the one that gives the highest

mean cross-validated macro-F1 score across the validation folds is chosen. With the TF-IDF vectoriser, the max features (max_features) and n-gram range are like tunable parameters and are added to the search grid [6]

3.6 Evaluation Metrics

A complete set of metrics to the classification tasks is used to evaluate model performance. Accuracy- the ratio of the correct items- is an overall measure of the performance which can be misleading in the case of imbalance between the classes [9]. According to this, Precision (the percentage of predicted positives that are really positive), Recall (the percentage of actual positives that is actually sentineled) and F1-score (the harmonic mean of Precision and Recall) are computed on a class-by-class basis and shown as macro-averages. ROC-AUC (Receiver Operating Characteristic Area Under the Curve) is a threshold-free measure of the discriminability of a classifier and is particularly handy when comparing the performance of multiple classifiers [6]. Confusion matrix is also analyzed to identify systematic misclassification schemes; such systematic misclassification can be the global tendency to misclassify negative reviews as positive; this would be of great commercial significance, and would have to be highlighted to the client. The suggested evaluation metrics along with their interpretations are summarised in Table 1.

Table 1: Recommended Evaluation Metrics for Sentiment Classifier Assessment

Metric	Definition	Relevance to Task
Accuracy	Proportion of all instances correctly classified.	Broad performance indicator; use with caution if classes are imbalanced.
Precision	$TP / (TP + FP)$: fraction of positive predictions that are correct.	Minimises false positives; important if acting on positive reviews.
Recall	$TP / (TP + FN)$: fraction of actual positives correctly retrieved.	Minimises missed negative reviews; critical for customer service alerts.
Macro F1	Harmonic mean of Precision and Recall, averaged equally across classes.	Primary recommended metric; balances performance across both classes.
ROC-AUC	Area under the Receiver Operating Characteristic curve.	Threshold-independent comparison metric across classifier variants.
Confusion Matrix	Cross-tabulation of predicted vs. actual class labels.	Reveals systematic misclassification patterns for targeted improvement.

4. Discussion and Recommendations

Based on the analysis of the literature and workflow analysis in the previous sections, the optimal sentiment classifier to apply in the analysis system of the shopper reviews by the client would be Logistic Regression with L2 regularisation, and TF-IDF feature extraction using unigrams and bigrams. Three overlapping lines of empirical evidence support this recommendation.

Firstly, the Logistic Regression has been as effective as or more so than the SVM and the Naive Bayes in binary sentiment classification problems in point-to-point comparative analysis of different benchmark datasets [1], [7]. Model coefficients of a trained Logistic Regression classifier can be directly interpreted: positive coefficients indicate terms most likely to predict positive sentiment, whereas negative coefficients indicate negative sentiment predictors. This practical benefit of transparency is significant as compared to less interpretable designs, as marketing analysts can audit and verify model behaviour without expert ML knowledge. The interpretability criterion is one that is particularly welcome during the phase of a young company when the commercial importance of the stakeholders in the trust to the ML system is at stake.

Second, in high-dimensional, sparse feature space (that Logistic Regression inherently generalises to in sparse feature space, which is the case in TF-IDF-based text representations [3], [7]) it is also sparse. In comparison, Naive Bayes is based on the conditional independence assumption, which is frequently breached in natural language data; its accuracy has been demonstrated to level off at smaller scales compared to Logistic Regression since increasing scale and vocabulary size [1]. Linear SVM is also a competitor which must not be eliminated as a second competitor in the experimental comparison, as the differences between the marginal performance of it and that of Logistic Regression may be conditional on the nature of the client review corpus under use.

Third, training and inference of Logistic Regression is much cheaper than the cost of training a kernel-based SVM that is quadratically or worse as the number of training cases increases [3]. As the scikit-learn [6] implementation is optimised, training the Logistic Regression on tens of thousands of reviews can be done in seconds on commodity hardware, and entirely within the constraints of the computing power of the client. The complete pipeline, including pre-processing, TF-IDF vectorisation and training a model, can be performed on a regular laptop without GPU acceleration, including just installing typical Python packages (scikit-learn, NLTK, pandas). Table 2 provides a relative summary of the three candidate algorithms considered.

Table 2: Comparative Summary of Candidate Classifiers for Sentiment Analysis

Classifier	Typical Accuracy*	Computational Cost	Interpretability
Logistic Regression	~85–89%	Low (seconds)	High – direct coefficient inspection
Linear SVM	~84–88%	Low–Medium	Moderate – weight vector inspection
Naïve Bayes (MNB)	~79–83%	Very Low	Moderate – feature log-probabilities
Lexicon-based (baseline)	~70–78%	Very Low	High – rule-based

* Accuracy ranges are indicative figures drawn from benchmarks cited in this report [1], [4], [7], [8].

In terms of the evaluation strategy, it is suggested that the client should use the macro-averaged F1-score as the main performance measure since it equalizes performance on positive and negative classes regardless of the classes frequency distribution [9]. Having a macro-F1 score of more than 0.85 on a held-out test set would represent a robust baseline to initial deployment, which is comparable to other studies reported in similar studies [7], [8]. The last hold-out evaluation set should be a stratified 80/20 train-test split, only used once all the hyperparameter tuning has been done, to give a fair estimate of production performance. This protocol avoids data leakage, which is a sensitive issue in implemented ML systems [6].

With respect to continuous operational management, sentiment distributions can also change over time as the language used by consumers changes and new types of products are created- a process called dataset drift or concept drift [2]. It is therefore recommended that the client establishes some form of monitoring to ensure that the performance of the classifier on the incoming review data is tracked every month and retraining is performed when the macro-F1 declines to a certain predetermined value (e.g. 0.80). The ease of retraining because of the lightweight of the proposed pipeline is also an advantage of the traditional ML method over more complex architectures since the former is simple and cheap to train.

5. Conclusion

The current report has provided an overview of classical machine learning methods of sentiment analysis and suggested a real-world, computationally effective pipeline which can be applied to the business needs of a marketing organization that wishes to categorize reviews of shoppers. The work section thereof has demonstrated that the so-called supervised classifiers, in particular, the Logistic Regression and the Linear SVM, are always better than unsupervised lexicon-based system, and match much more complicated structures at a very small

fraction of the cost. Naive bayes was found to be a feasible and effective baseline, but its performance limit is typically less than that of the suggested classifier.

The six-step pipeline (data labelling using star-rating heuristics, text pre-processing including negation processing and then extracting features using TF-IDF with n-grams of unigrams and bigrams), stratified cross-validated training, Grid Search hyperparameter optimisation, and multi-metric evaluation is founded on best practice in the literature on ML and can be directly implemented using standard and open-source Python packages. L2 regularisation Logistic Regression has been suggested as the best classifier due to its empirically proven trade off between predictive accuracy, model interpretability, scalability and low computational footprint.

The excessive effects of this recommendation do not just stop at immediate deployment. As the amount of data, and the computational capacity of the organisation, increases, the same pipeline can be scaled out over time, e.g. by adding sentiment lexicon features, part-of-speech tags or domain-specific vocabulary etc. without necessarily rearchitecturally re-designing the pipeline. The framework is thus not only suitable to the present requirements but is set to be able to adapt with the strategic course of the company. Periodic retraining of classifiers, and periodic monitoring of their performance as new review data becomes available will be needed in the long run performance of the classifiers, but will be fully compatible with the low-overhead design ethos that is the subject of this recommendation.

References

- [1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, no. 1–2, pp. 1–135, 2008.
- [2] B. Liu, *Sentiment Analysis and Opinion Mining*. San Rafael, CA, USA: Morgan & Claypool Publishers, 2012.
- [3] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. 10th Eur. Conf. Mach. Learn. (ECML)*, Chemnitz, Germany, 1998, pp. 137–142.
- [4] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *Stanford Univ., Stanford, CA, USA, Tech. Rep. CS224N*, 2009.
- [5] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proc. 45th Annu. Meet. Assoc. Comput. Linguistics*, Prague, Czech Republic, 2007, pp. 440–447.
- [6] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [7] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *J. Artif. Intell. Res.*, vol. 50, pp. 723–762, 2014.
- [8] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meet. Assoc. Comput. Linguistics*, Portland, OR, USA, 2011, pp. 142–150.
- [9] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. (draft). 2023. [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/>
- [10] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Comput. Linguistics*, vol. 37, no. 2, pp. 267–307, 2011.
- [11] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [12] A. McCallum and K. Nigam, "A comparison of event models for Naive Bayes text classification," in *Proc. AAAI-98 Workshop Learn. Text Categorization*, Madison, WI, USA, 1998, pp. 41–48.