Machine Learning Report   LEVILLAYER Noam, LAHJAJI Hamid, VIRELIZIER Aubin
April 2025

For the first part (the binary classification) we made a multiple step preprocessing approach:

- Raw version: original tweets, very messy
- Cleaning: removing URLs, mentions, punctuation, numbers + lowercasing
- Lemmatization: turning word into most basic form and get rid of adjectives
- Vectorization: Bag of Words (BOW)

1 Grading Criterion 1: Implementing Algorithms

To evaluate and compare the performance of different algorithms, we applied the same set of explanatory variables (based on the BOW representation) to six models:

- Logistic Regression
- K-Nearest Neighbors,
- Random Forest,
- Neural Networks (MLP),

and two additional models not covered in class: **SVM** and **FinBERT**(deep-learning)
For each model, we report precision, recall, weighted F1-score, training time, and the main hyperparameters used. This comparison allows us to assess the trade-offs between accuracy, complexity, and computational efficiency across models:

MODEL	VECTORIZER	PRECISION	RECALL	F1-SCORE	TIME	HYPERPARAMETERS
Logistic Regression	BOW	0.8796	0.8903	0.8747	<1 sec	default
KNN	BOW	0.8302	0.8541	0.8329	<1 sec	n_neighbors = 1
Random Forest	BOW	0.8786	0.8895	0.8791	15 sec	default
Neural Network (MLP)	BOW	0.8627	0.8692	0.8654	56 sec	max_iter = 300
SVM	BOW	0.8658	0.8743	0.8690	151 sec	default
FinBERT	BertTokenizer	0.8951	0.8977	0.8963	247 sec	default

Table 1: Comparaison compacte des modèles de classification

Among the traditional machine learning models, Random Forest and Logistic Regression achieved the best overall performance, both showing high precision and recall with very fast training times.
Regarding FinBERT it achieved the highest F1-score and excellent precision and recall. Although its training time is longer, this is expected due to the complexity of transformer-based models. It clearly demonstrates the power of a financially pretrained language model for understanding and classifying financial text.

2 Grading Criterion 2: Model Optimization

2.1 Selection of Explanatory Variables

To explore the impact of feature(X) representation on model performance, we selected the Support Vector Machine (SVM) classifier from the previous comparison. We varied the explanatory variables by switching from a Bag of Words (BOW) vectorizer to a TF-IDF vectorizer. This change aims to reduce Test Error by assigning more importance to informative terms. The evolution of the test error resulting from this modification is illustrated in the graph below.

2.2 Selection of Hyperparameters

To analyze the impact of hyperparameter optimization on model performance, we selected two models: Random Forest and Multi-layer Perceptron (MLP). For each model, we tuned key hyperparameters using grid search and evaluated their influence on the test error and weighted F1-Score.
For the MLP model, we observed a slight improvement in performance when changing the solver from the default 'adam' to 'lbfgs'. The 'lbfgs' solver is a quasi-Newton optimization method known to perform better on smaller datasets (our dataset contains around 9k tweets but this is still considered relatively small for training neural networks)
Same thing, we observed a slight improvement in performance when we change parameters. In fact we have adjusted "nestimators", "maxdepth", "minsamplesplit", "minsamplesleaf", "bootstrap" and "classweight".

3 Grading Criterion 3: Selection and Evaluation of the Best Performing Model

- We selected FinBERT as the best performing model due to its results and suitability for financial text classification. Unlike traditional algorithms, FinBERT is based on BERT architecture and specifically pretrained on financial data, giving it a powerful understanding of this domain-specific language.

Instead of using basic vectorizers like Bag of Words or TF-IDF, FinBERT uses BertTokenizer, which creates contextual embeddings (i.e. adding word meaning in context)

With a training time of approximately 4 minutes (on GPU). FinBERT achieved the best metrics among those we have tested:

- Precision: 0.8951 -*When the model predicts a class, it's correct 90% of the time-*
- Recall: 0.8977 -*The model correctly finds 90% of all true cases-*
- Weighted F1-Score: 0.8963 -*Balanced performance across both classes, considering class size. -*
- Accuracy: 0.8977 -*Nearly 90% of tweets were correctly classified overall. -*

The ROC curve below shows an AUC of 0.89, confirms strong separation between classes.

However, performance on the minority class (”Negative”) is weaker, indicating a class imbalance. Future improvements could include balancing the dataset or adjusting class weights. We address that problem in Grading Criterion 4 with the RoBERTa model.

4 Grading Criterion 4: Ability to Learn Coding Independently

We chose a fine-tuned tweet-sentiment-analysis RoBERTa model from Hugging Face, which outputs 0, 1, 2 for negative, positive, or neutral sentiment. This model is ideal for our database. This is based on roberta-base which has 125M parameters.

To handle the unbalanced label distribution in our dataset, we performed stratified cross-validation to ensure each fold maintains the same label proportion as the original dataset, providing a more reliable performance evaluation.

During training, we experimented with various hyperparameter combinations but encountered overfitting at a certain epoch. To address this, we focused on methods to reduce overfitting, especially given the small dataset size.

We employed back-translation, translating text into French and back to English to generate new, unique tweets while preserving original sentiment labels. This approach augmented our dataset size to nearly 19,000 unique tweets, enhancing the diversity and robustness of our training data. We used Hugging Face translation models for this process, which took about 3 to 4 hours.

Back-translation significantly improved model performance. Validation loss was kept under 0.35, and the weighted F1-score, precision, recall, and accuracy increased to approximately 0.94, reaching up to 0.95 in some folds. The ROC curves for each class showed good performance despite the unbalanced label distribution.

MODEL	VECTORIZATION	PRECISION	RECALL	ACCURACY	WEIGHTED F1	TIME
RoBERTa + CV (10 folds)	AutoTokenizer	0.9482	0.9483	0.9483	0.9482	8–10 hrs

Table 2: Performance du modèle RoBERTa avec validation croisée

AUC (Area Under the Curve) Results:

- Class 0 - AUC: 0.93: The model has a high discriminative power for Class 0, indicating it can effectively distinguish between Class 0 and other classes.
- Class 1 - AUC: 0.93: Similarly, the model performs well for Class 1, showing a strong ability to differentiate Class 1 from other classes.
- Class 2 - AUC: 0.92: The model also performs well for Class 2, though slightly lower than Classes 0 and 1, but still indicating good discriminative power.
- Micro-Average AUC: 0.94: The overall performance of the model across all classes is excellent, with a high micro-average AUC, suggesting that the model is generally very effective in distinguishing between all classes.

Performance summary:

- **Precision and Recall:** Both precision and recall are very high—approximately 0.9482 and 0.9483, respectively. This balance suggests that the model is excellent at correctly identifying the positive class (high recall) while keeping the rate of false positives low (high precision). In other words, the classifier is not only sensitive to true positives but also avoids misclassifying negatives as positives.
- **Accuracy and Weighted F1 Score:** The reported accuracy of around 0.9483 provides further evidence of the model's reliability over the entire dataset. Moreover, the close match between the accuracy and the weighted F1 score (approximately 0.9482) implies that performance is consistent across all classes, even in the presence of class imbalances. The weighted F1 score, which takes both precision and recall into account for each class, reinforces that the model maintains a strong balance between these factors.
- **Training Time:** Although the model requires a relatively lengthy training period (8-10 hours), this duration is justifiable given the high quality of the extracted features and the robust performance metrics. The investment in training time is likely reflected in the model's precise and reliable predictions, demonstrating that the complexity and training duration are warranted to achieve these outcomes.