NLP Project

Automated Customers Reviews: sentimental analysis

Introduction

Business Case

Problem Statement

Retail companies receive thousands of text reviews monthly

Manual categorization is time-consuming and costly

Need for automated sentiment classification

Project Goals

✓ Classify reviews as Positive, Negative, or Neutral

✓ Compare Traditional ML vs Deep Learning (Transformers)

✓ Evaluate which approach yields better results

DATASET

HuggingFace IMDB + Local Amazon CSVs

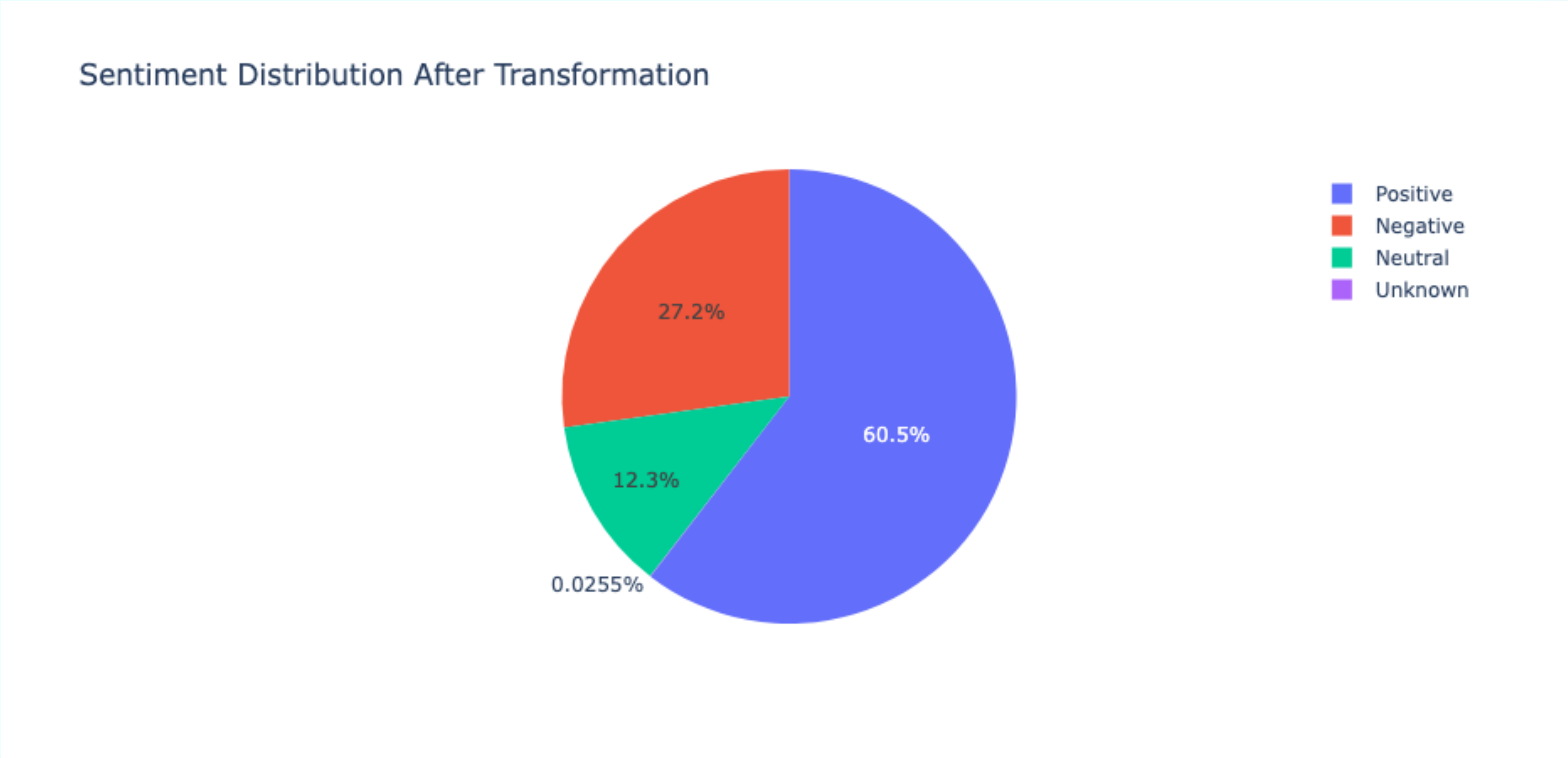
Total reviews after cleaning: 54617

Sentiment distribution:

Positive (5 🌟) *32953*

Negative (1-3 🌟) *14941*

Neutral (4 🌟) *6723*



	reviews.rating	sentiment	count
0	1.0	Negative	673
1	2.0	Negative	12982
2	3.0	Negative	1303
3	4.0	Neutral	6776
4	5.0	Positive	33252

FINAL DATASET PREPARATION

Cleaning and dealing with imbalanced classes

- Nans cleaned (<15)
- Very shorts reviews cleaned (10 characters).
- Removed "unknown" sentiment label.
- Imbalanced dealing strategy:
  - Class weights: balanced (x1.15) for Logistic regression and Linear SVM, balanced subsample and neutral weight x1.15 for Random Forest (🥇).
  - For training only, upsampled neutral to match negative (RandomOverSampler)
  - "Stratify=y".

- Advanced preprocessing with NLTK.
- Tokenisation, stopwords removals, lemmatisation.
- TF-IDF (importance) and count vectorisation (frequency) –not combined.

Train set size: 43692
Test set size: 10923

Train set distribution:

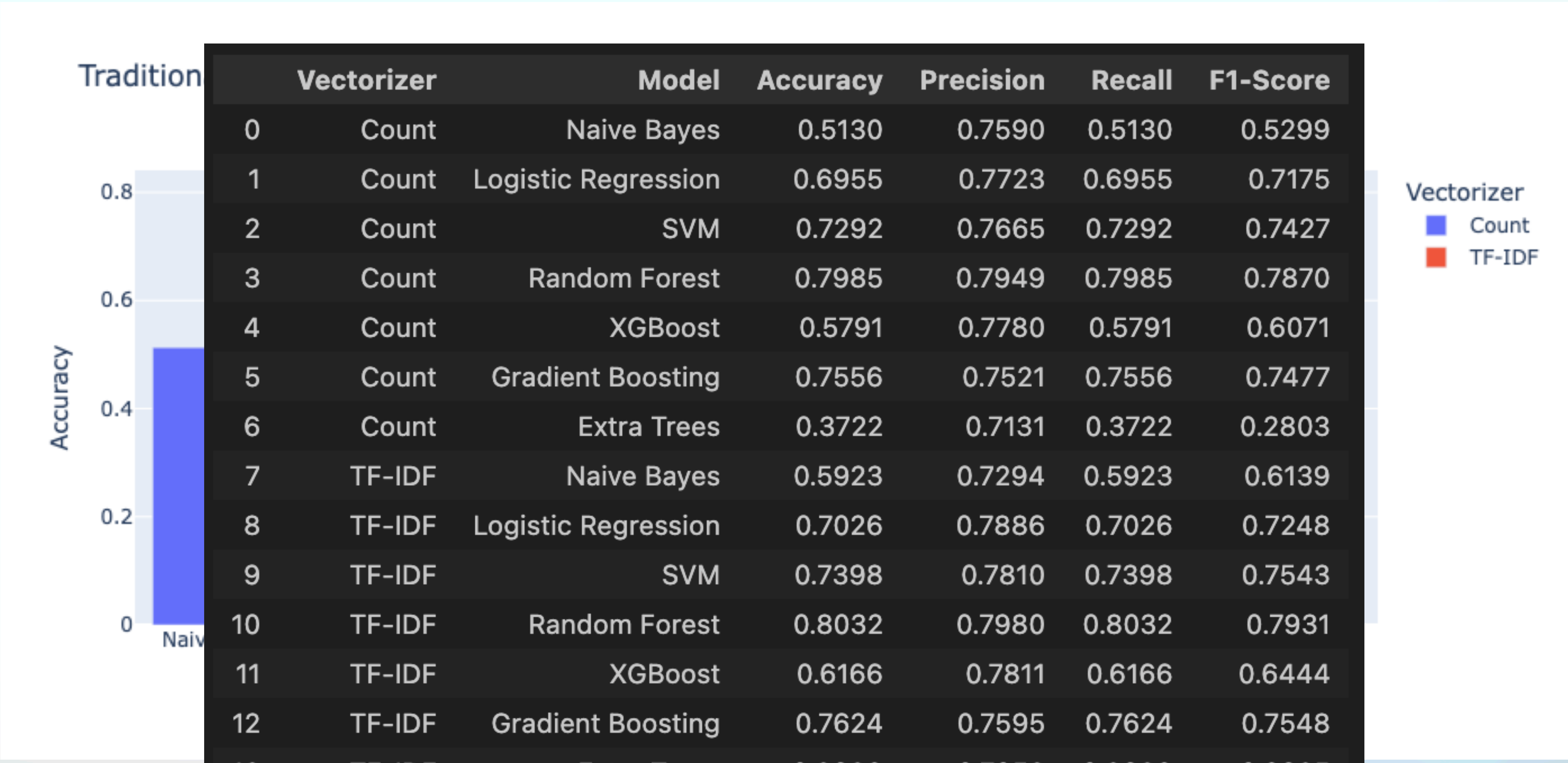
Positive	**26361**
Negative	**11953**
Neutral	**5378**

TRADITIONAL ML MODELS

Training and Evaluation

Best performing model: RANDOM FOREST with TF-IDF vectoriser.

Overall Accuracy: 0.8032
Overall Precision: 0.7980
Overall Recall: 0.8032
Overall F1-Score: 0.7931



	Vectorizer	Model	Accuracy	Precision	Recall	F1-Score
0	Count	Naive Bayes	0.5130	0.7590	0.5130	0.5299
1	Count	Logistic Regression	0.6955	0.7723	0.6955	0.7175
2	Count	SVM	0.7292	0.7665	0.7292	0.7427
3	Count	Random Forest	0.7985	0.7949	0.7985	0.7870
4	Count	XGBoost	0.5791	0.7780	0.5791	0.6071
5	Count	Gradient Boosting	0.7556	0.7521	0.7556	0.7477
6	Count	Extra Trees	0.3722	0.7131	0.3722	0.2803
7	TF-IDF	Naive Bayes	0.5923	0.7294	0.5923	0.6139
8	TF-IDF	Logistic Regression	0.7026	0.7886	0.7026	0.7248
9	TF-IDF	SVM	0.7398	0.7810	0.7398	0.7543
10	TF-IDF	Random Forest	0.8032	0.7980	0.8032	0.7931
11	TF-IDF	XGBoost	0.6166	0.7811	0.6166	0.6444
12	TF-IDF	Gradient Boosting	0.7624	0.7595	0.7624	0.7548
13	TF-IDF	Extra Trees	0.3689	0.7053	0.3689	0.2635

HYPER-PARAMETRING

Two examples

- Grid Search for the best parameters (TD-IDF vectorised dataset).
- RF:
  Too much false positives for Positives (still a 10% of the true positives).
  A good precision of true positives.
- XGBoost:
  Worse global scores.
  Best precision (but for neutral)

	Negative	Neutral	Positive
Negative	2256	75	657
Neutral	14	634	697
Positive	442	436	5712

Random Forest

	Negative	Neutral	Positive
Negative	2279	453	256
Neutral	5	1313	27
Positive	772	3789	2029

XGBoost

TRANSFORMER APPROACH

HuggingFace Transformer

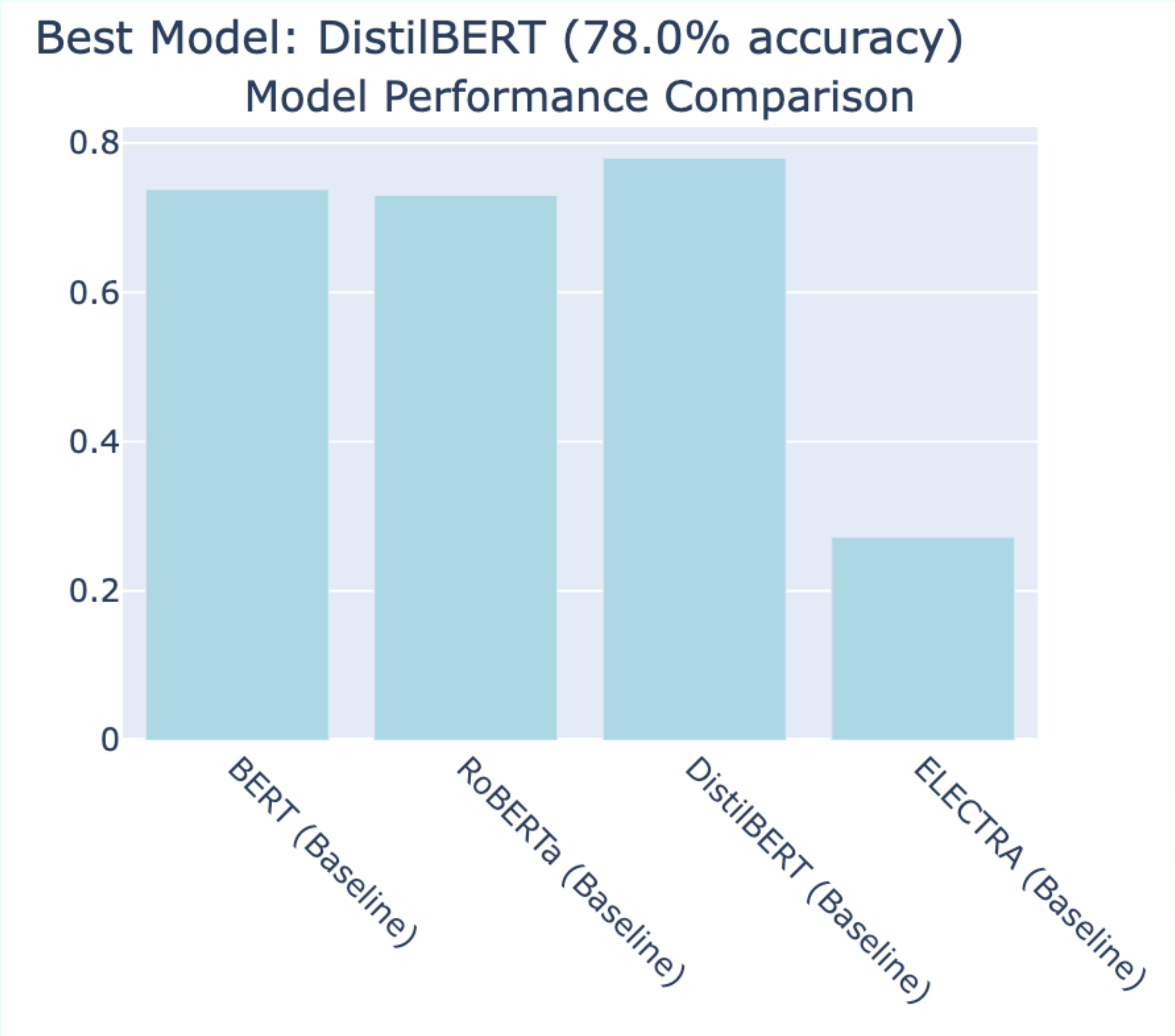
Models Evaluated

1. **BERT**

2. **RoBERTa** - Optimised BERT variant

3. **DistilBERT** - 60% smaller, 97% performance

4. **ELECTRA** - Efficient pre-training



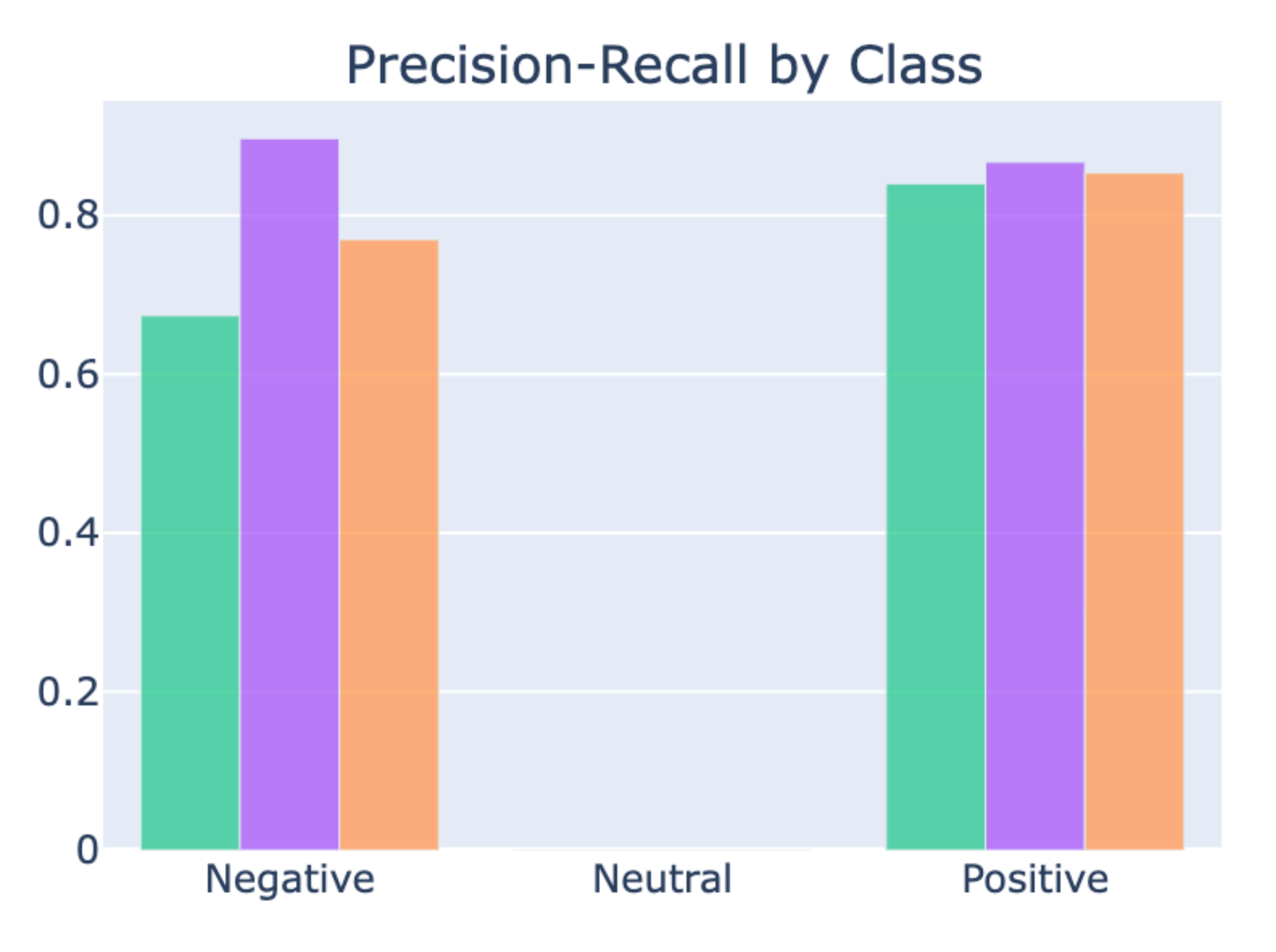
	Model	Type	Accuracy	Precision	Recall	F1-Score
0	BERT	Baseline (Pre-trained)	0.738	0.7594	0.738	0.7472
1	RoBERTa	Baseline (Pre-trained)	0.730	0.7432	0.730	0.7337
2	DistilBERT	Baseline (Pre-trained)	0.780	0.7025	0.780	0.7368
3	ELECTRA	Baseline (Pre-trained)	0.272	0.0740	0.272	0.1163

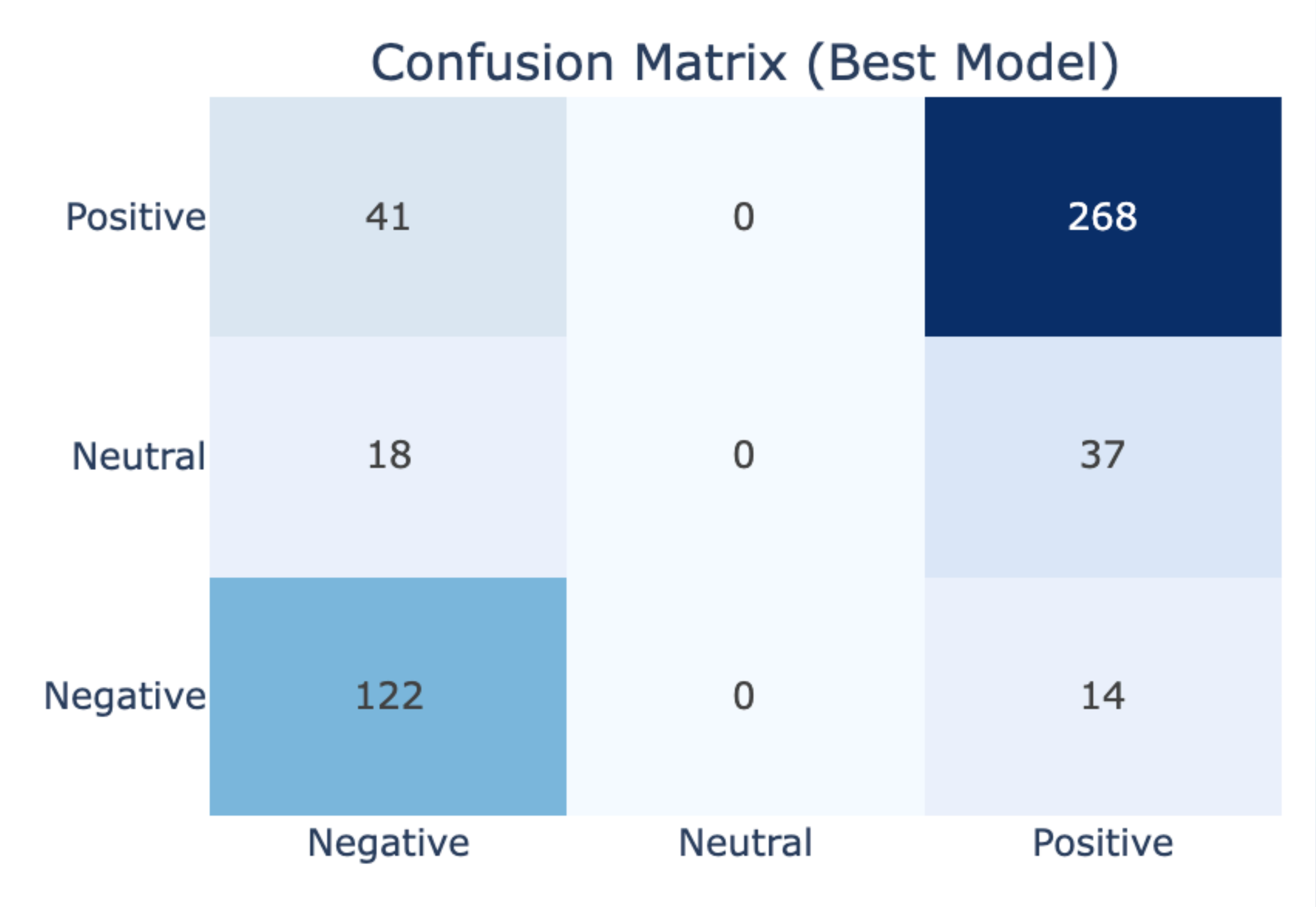
TRANSFORMER MODELS

Evaluation Dashboard

- DistilBert results:

```
🏆 BEST PERFORMING TRANSFORMER MODEL:
   🥇 Model: DistilBERT (Baseline (Pre-trained))
   📊 Accuracy: 0.7800 (78.0%)
   📊 F1-Score: 0.7368
   📊 Precision: 0.7025
   📊 Recall: 0.7800
```





FINAL ANALYSIS

What to change and what to keep

- Replace the alternative dataset: instead of IMDB, I would consider SemEval or another.

- Fine-tuning for a better optimisation.

- Class imbalanced: oversampling with SMOTE or threshold adjustment (better decision gap for the minority class).

thank you