Sentiment Analysis On Tweets With Machine Learning

Lucas Gomes Dantas

Departamento de Informática e Matemática Aplicada (DIMAp)
Universidade Federal do Rio Grande do Norte (UFRN)

July 16, 2024

Abstract

This article explores the application of machine learning algorithms to sentiment analysis, aiming to identify predominant sentiments in textual data. Utilizing a dataset of English tweets from Kaggle, the study examines the effectiveness of various learning methods. Experimental results provide insights into the best practices for sentiment classification, highlighting the potential of machine learning in developing sentiment analysis systems.

Introduction

This article focuses on sentiment analysis, aiming to classify predominant sentiment in text into sadness, joy, love, anger, fear, and surprise.

Dataset

- Extracted from Kaggle: `https://www.kaggle.com/datasets/nelgiriyewithana/emotions`
- 416,809 instances with 3 attributes: ID, text, and label (0: sadness, 1: joy, 2: love, 3: anger, 4: fear, 5: surprise).

State of the Art

- Transformer Models: BERT, RoBERTa, XLNet, T5, GPT-3.
- RNNs: GRU.
- CNNs for text.
- Word Embeddings: Word2Vec, GloVe, Doc2Vec.
- Neural Network-Based Approaches: Autoencoders.
- Current Research Trends: Transfer Learning, Domain Adaptation, Multimodal Sentiment Analysis, Explainable AI.

Pre-processing

- Text converted to numerical vectors using BERT tokenization.
- Boxplot analysis showed no significant outliers in the dataset.
- Instance reduction with sampling (20% reduction).
- Attribute selection with Decision Tree and PCA.
- The Original Dataset had to be reduced to 24000 instances.

Pre-processing: Boxplot Analysis

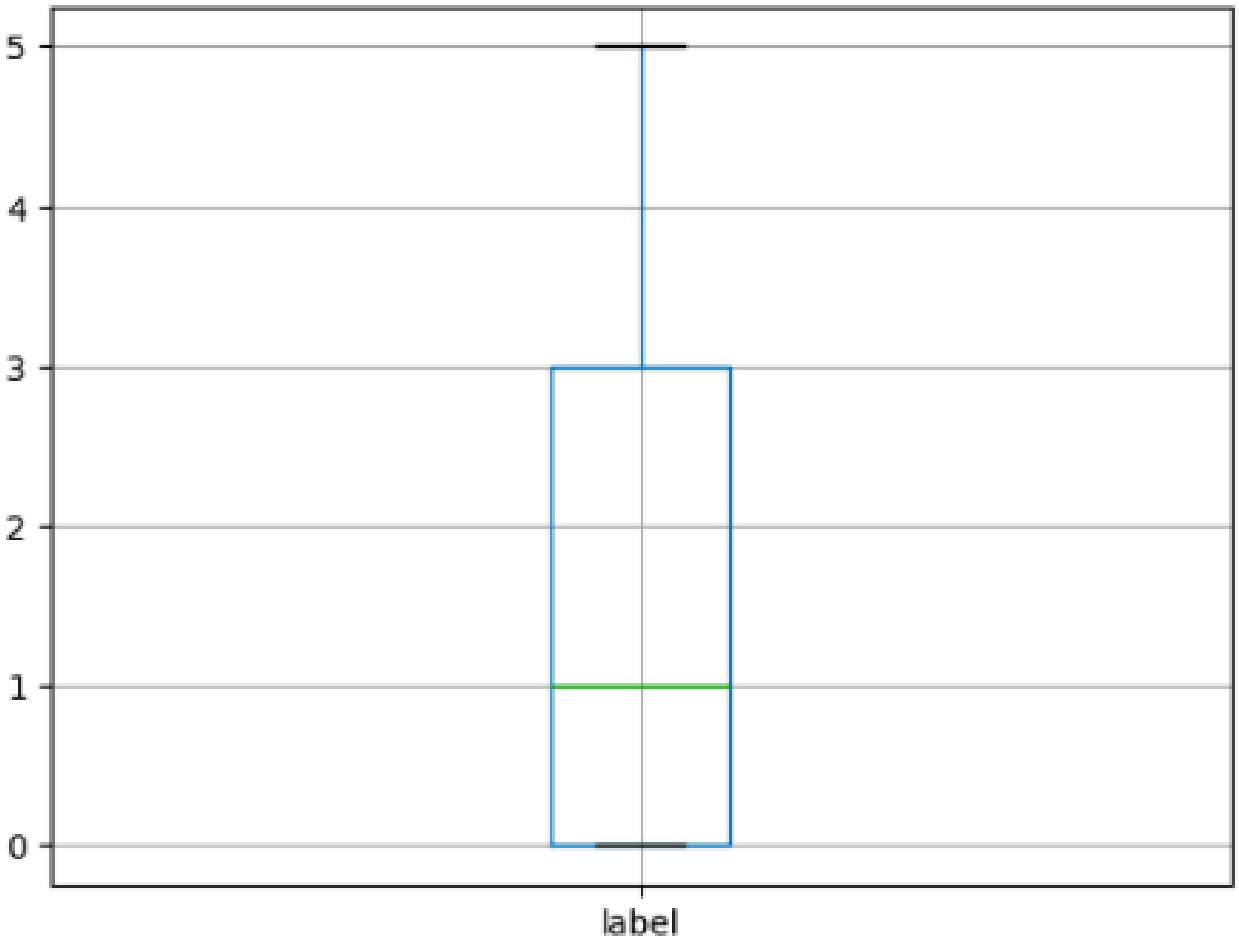

Figure: Boxplot applied on the "label" attribute from the sentiments dataset.

Pre-processing: Decision Tree

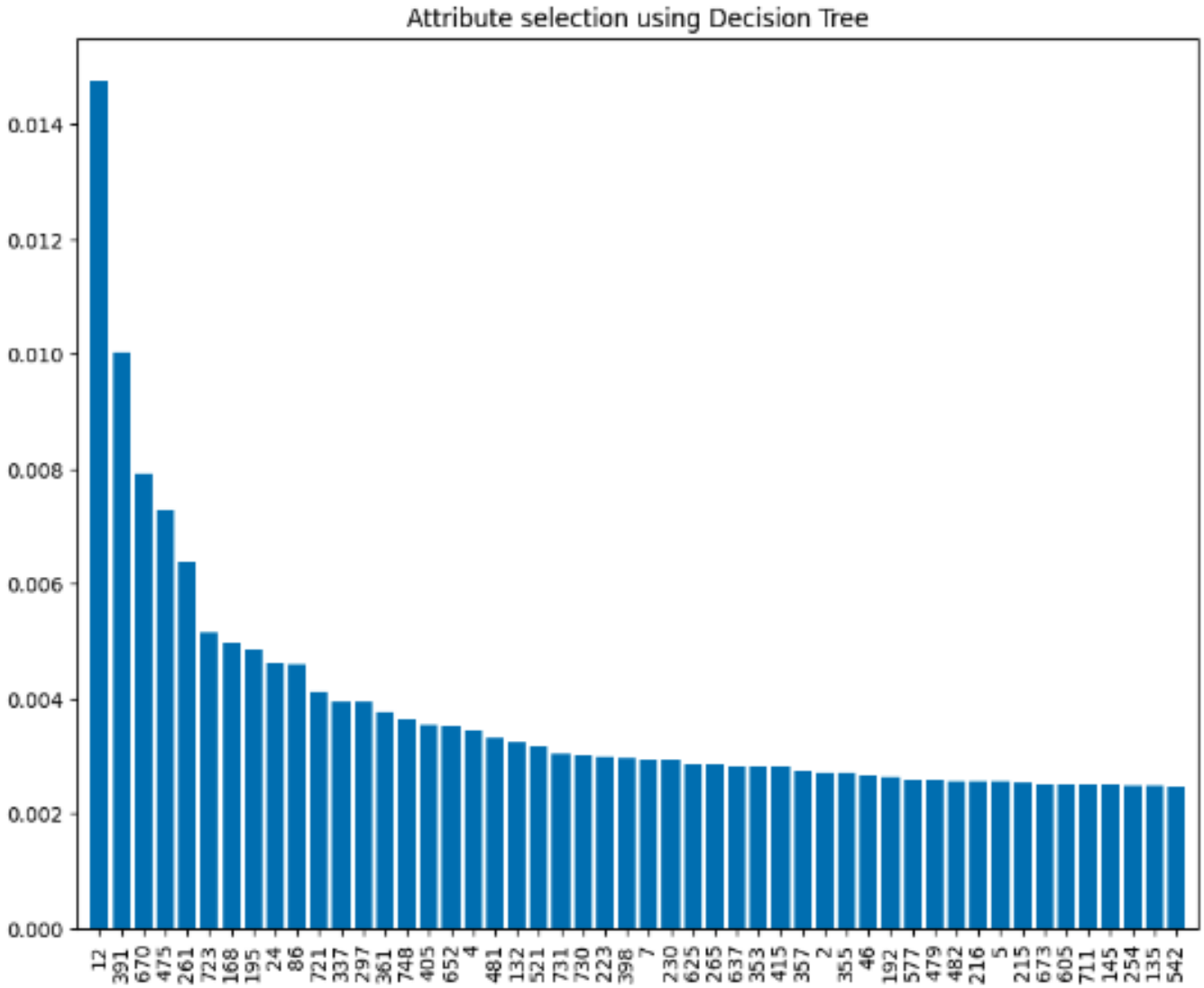

Figure: Attribute selection with Decision Tree

Pre-processing: Scree Plot

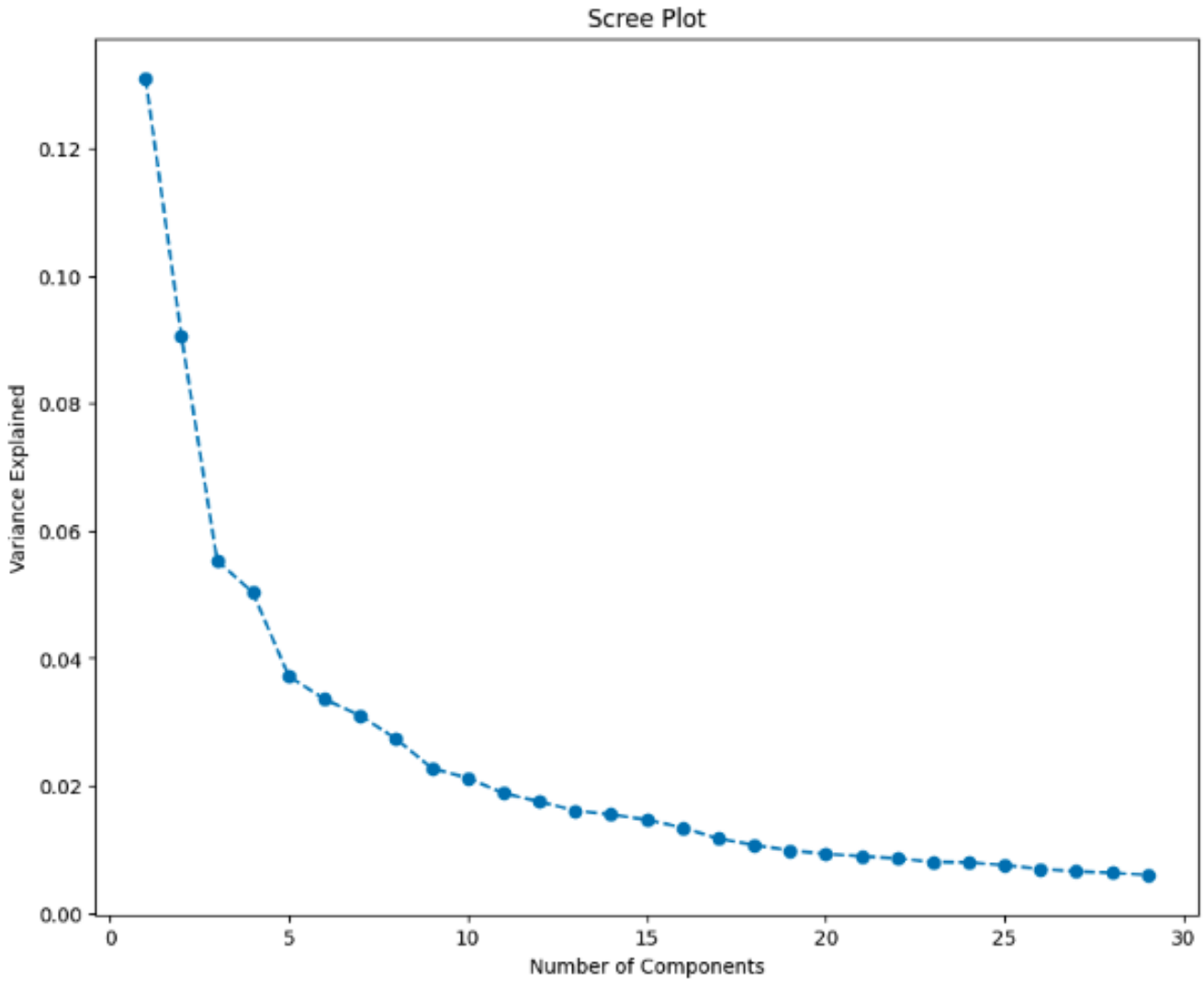

Figure: Scree plot for the PCA components, second execution.

Pre-processing: End State

Dataset	Number of Instances	Number of Attributes
Original Dataset	24,000	769
Reduced Dataset 1	19,200	769
Reduced Dataset 2	24,000	51
Reduced Dataset 3	24,000	30

Table: Summary of dataset reductions and their attributes.

Supervised Learning

- Algorithms: k-NN, Decision Trees, Naive Bayes, Multilayer Perceptron (MLP).
- Experimentation with different parameters and datasets.
- MLP showed the best performance with optimal configurations.

Supervised Learning: kNN

Dataset	Train/Test	1k Acc	3k Acc	5k Acc
Original Dataset	10-fold CV	33.27	34.78	37.14
	70/30	32.86	34.75	36.43
	80/20	33.35	34.70	36.58
	90/10	33.00	35.79	37.70
Reduced Dataset 1	10-fold CV	32.99	34.06	36.44
	70/30	33.21	34.16	35.43
	80/20	33.22	33.90	36.01
	90/10	32.55	33.33	36.04
Reduced Dataset 2	10-fold CV	16.39	16.48	16.39
	70/30	16.36	16.38	15.72
	80/20	16.22	16.66	15.68
	90/10	15.54	15.66	16.29
Reduced Dataset 3	10-fold CV	30.97	32.33	33.99
	70/30	30.63	32.26	33.81
	80/20	30.54	32.43	33.91
	90/10	30.75	32.33	33.29
Average		**28.24**	**29.38**	**30.68**
Standard Deviation		**7.06**	**7.62**	**8.55**

Table 2. kNN accuracy results for the different datasets and methodologies.

Supervised Learning: kNN

Training Strategy	Accuracy
10-fold Cross-Validation	29.60
Hold-out 90/10	29.36
Hold-out 80/20	29.43
Hold-out 70/30	29.33

Table 3. kNN average accuracy results for different training strategies.

Supervised Learning: Decision Tree

Base	Train/Test	md = 3 Acc	md = 5 Acc	md = 7 Acc
Original Dataset	10-fold CV	27.3	29.71	31.25
	70/30	27.77	30.48	30.98
	80/20	26.54	29.91	31.66
	90/10	26.37	28.75	30.70
Reduced Dataset 1	10-fold CV	27.25	30.00	31.45
	70/30	28.12	29.96	31.35
	80/20	28.09	30.54	30.65
	90/10	26.56	31.14	31.45
Reduced Dataset 2	10-fold CV	16.67	16.37	16.64
	70/30	16.08	16.25	15.62
	80/20	16.64	17.29	16.00
	90/10	16.41	16.87	15.87
Reduced Dataset 3	10-fold CV	26.94	28.94	30.82
	70/30	26.58	29.08	30.12
	80/20	25.54	29.02	30.83
	90/10	26.91	28.50	30.00
Average		**24.36**	**26.43**	**27.21**
Standard Deviation		**4.61**	**5.66**	**6.47**

Table 4. Decision Tree accuracy results for the different datasets and methodologies, with varying max depths (md).

Supervised Learning: Decision Tree

Training Strategy	Accuracy
10-fold Cross-Validation	26.11
Hold-out 90/10	25.79
Hold-out 80/20	26.06
Hold-out 70/30	26.03

Table 5. Decision Tree average accuracy results for different training strategies.

Supervised Learning: Naive Bayes

Dataset	Train/Test	Default Acc
Original Dataset	10-fold CV	38.77
	70/30	40.00
	80/20	39.70
	90/10	39.54
Reduced Dataset 1	10-fold CV	38.94
	70/30	40.53
	80/20	40.62
	90/10	40.15
Reduced Dataset 2	10-fold CV	16.57
	70/30	16.61
	80/20	16.95
	90/10	16.75
Reduced Dataset 3	10-fold CV	41.72
	70/30	41.20
	80/20	42.18
	90/10	42.25
Average		**34.53**
Standard Deviation		**10.33**

Table 6. Naive Bayes accuracy results for the different datasets and methodologies.

Learning Strategy	Accuracy
10-fold Cross-Validation	34.00
Hold-out 90/10	34.67
Hold-out 80/20	34.86
Hold-out 70/30	34.59

Table 7. Naive Bayes average accuracy results for different training strategies.

Supervised Learning: MLP

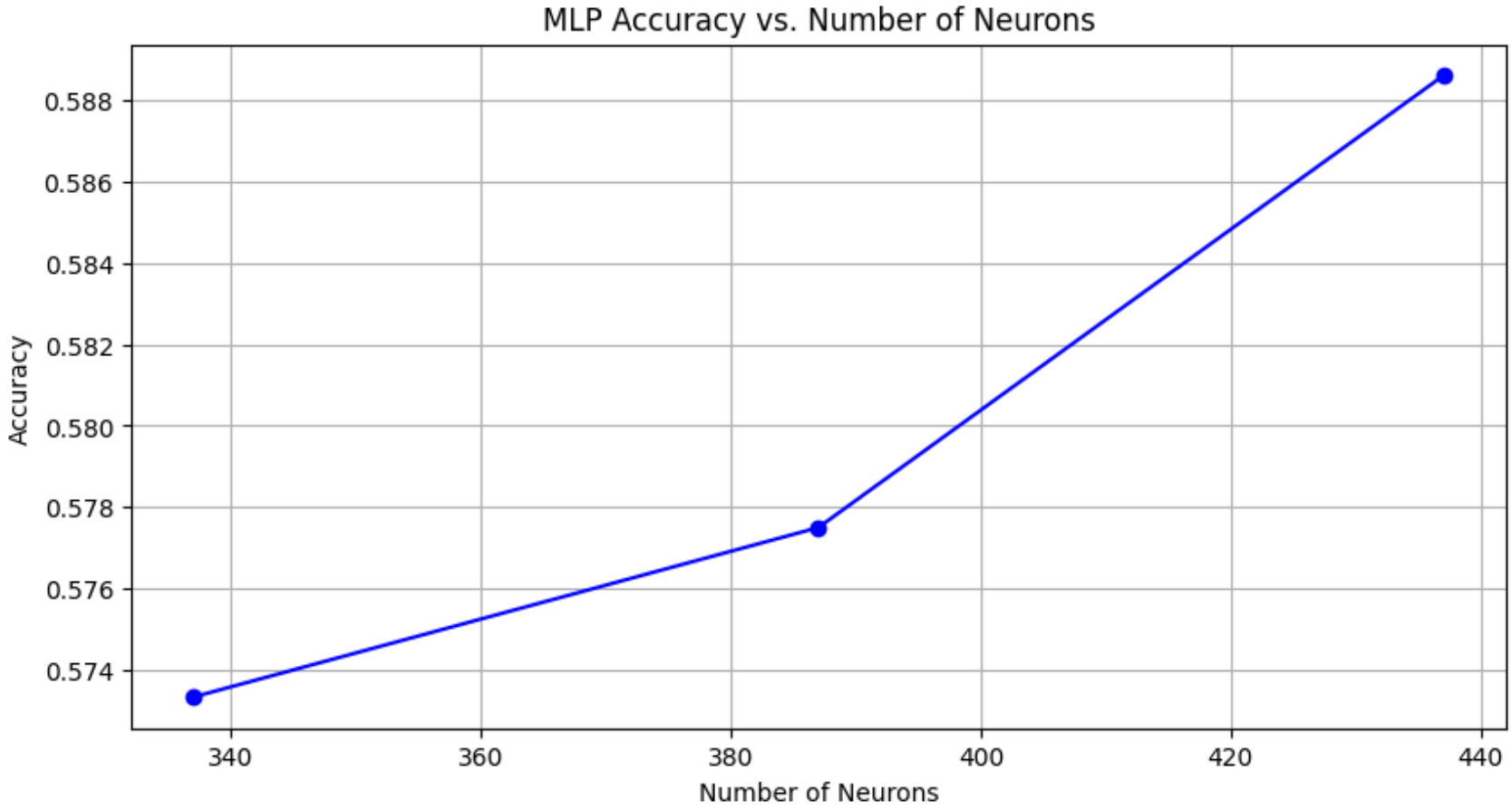

Figure: MLP execution with 70/30 holdout, accuracy vs. number of neurons.

Supervised Learning: MLP

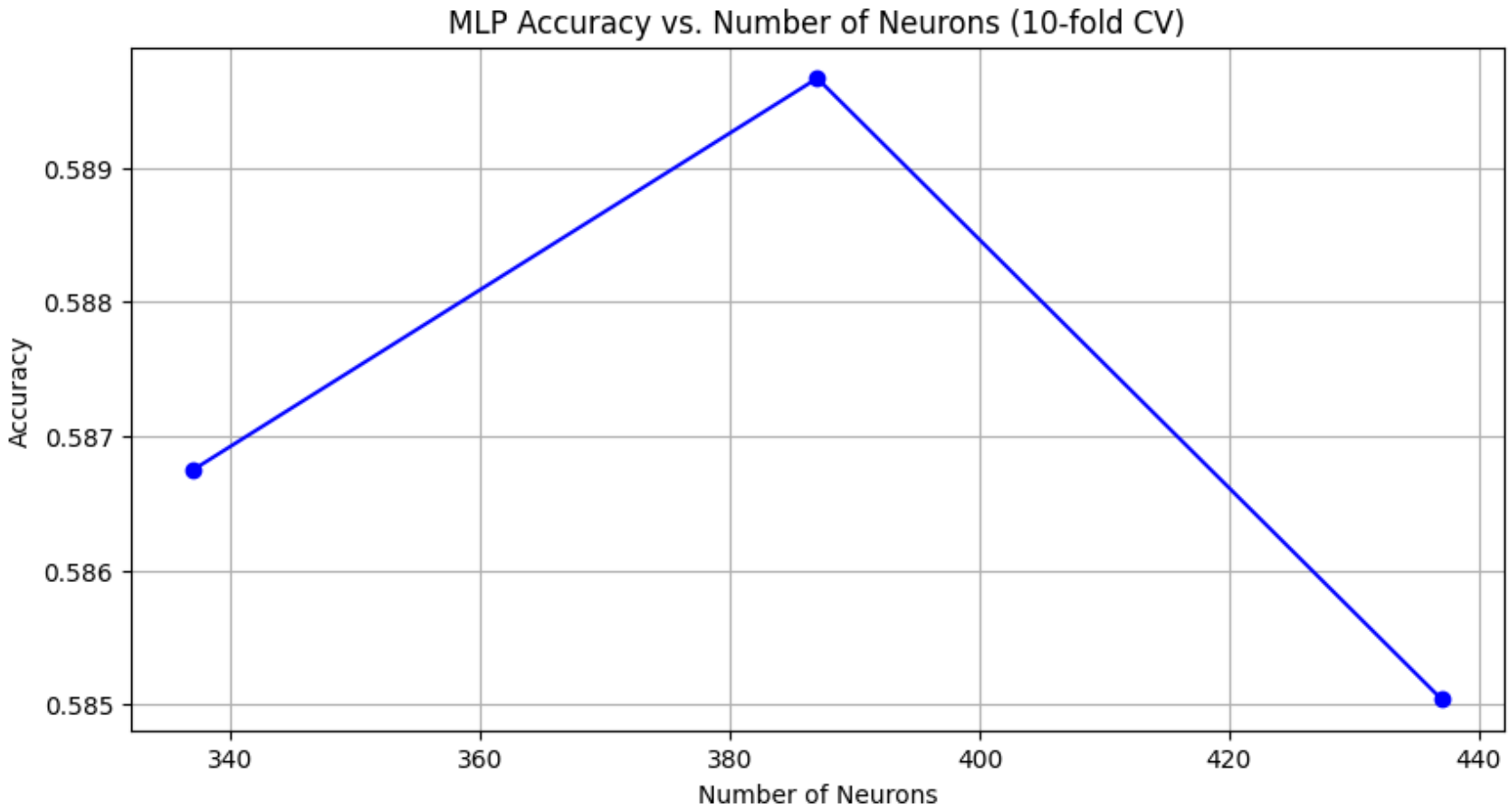

Figure: MLP execution with 10-fold cross-validation, accuracy vs. number of neurons.

Supervised Learning: MLP

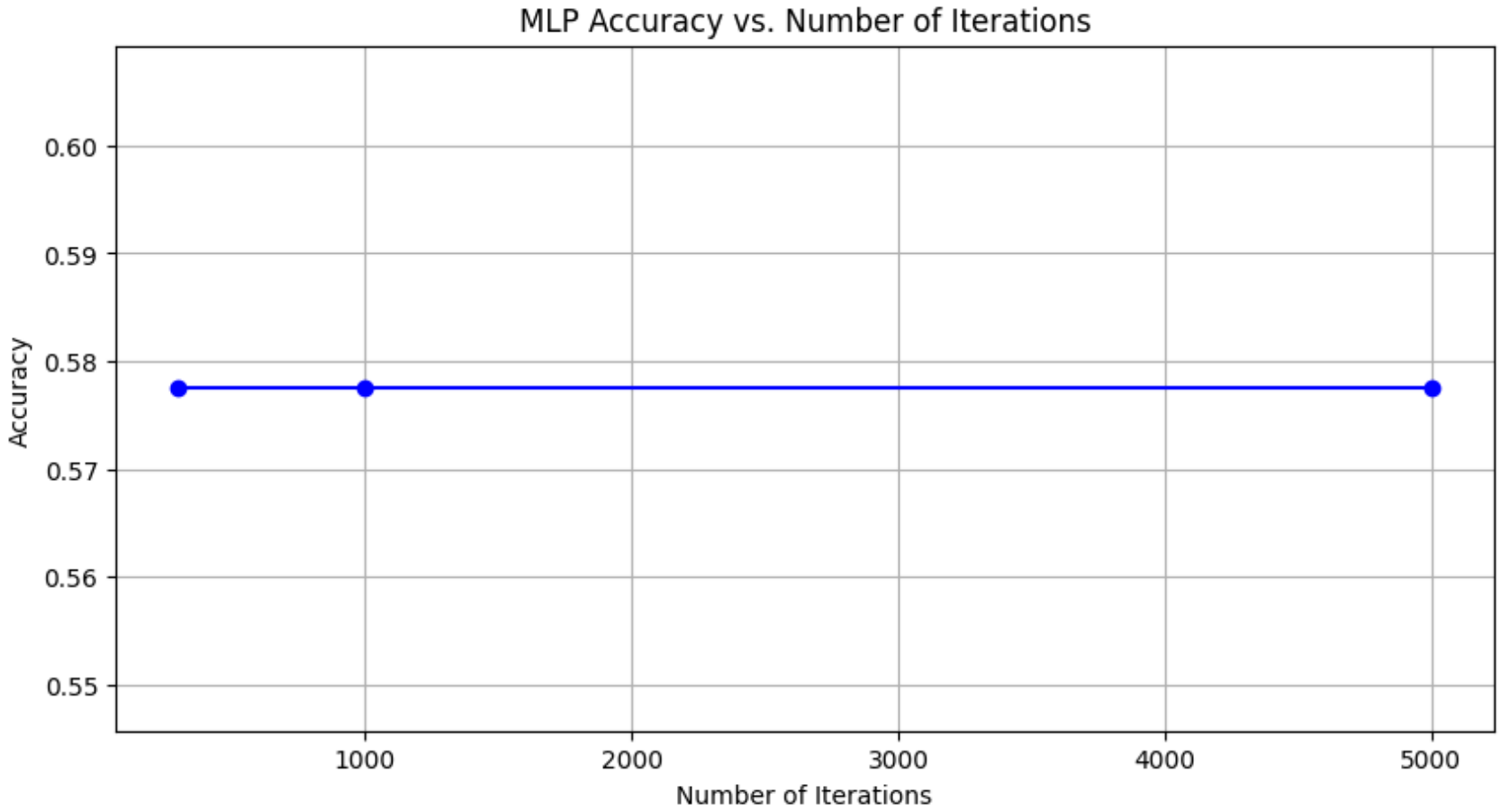

Figure: MLP execution with 70/30 holdout, varying the amount of iterations for 437 neurons.

Supervised Learning: MLP

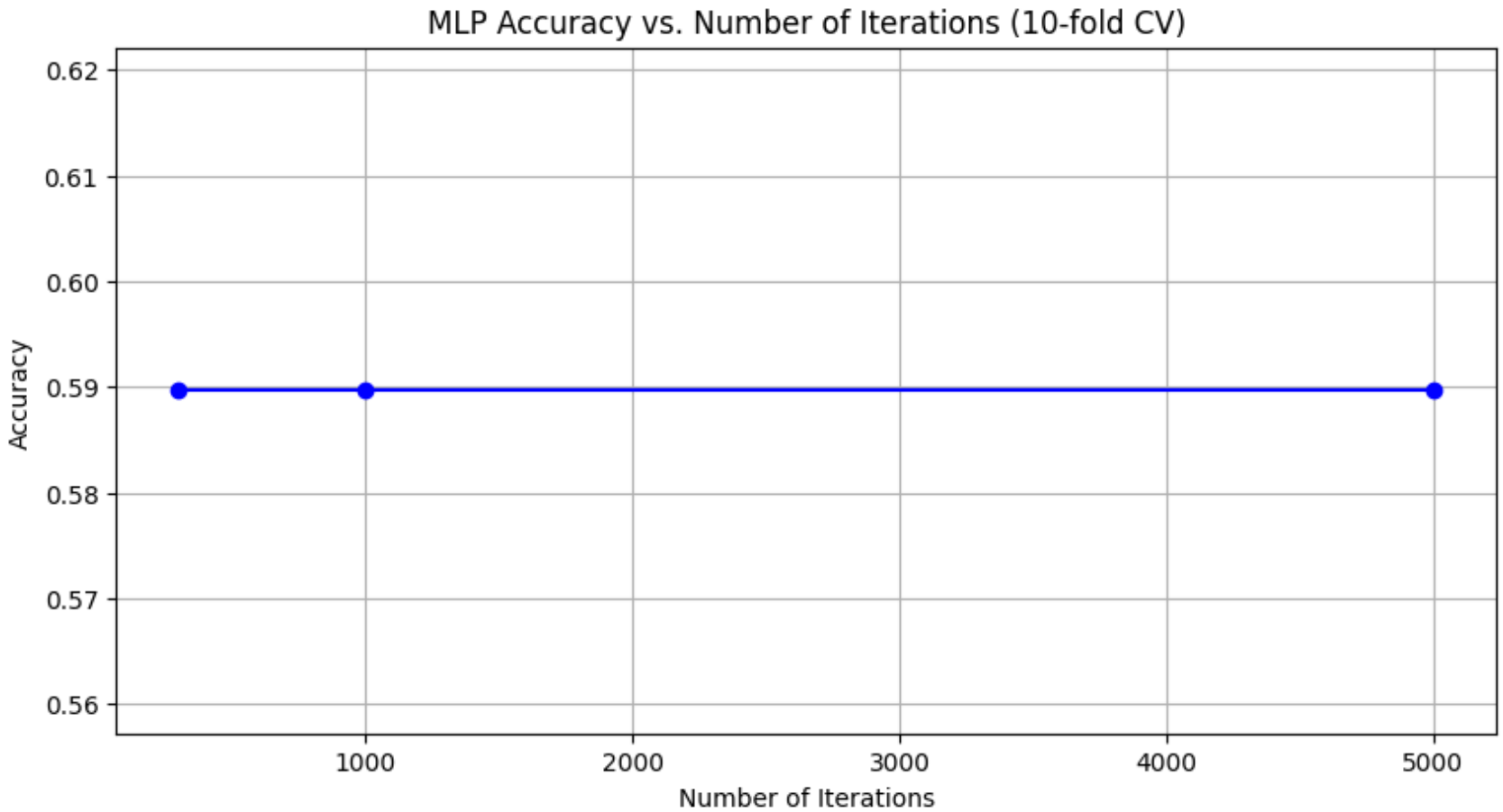

Figure: MLP execution with 10-fold cross-validation, varying the amount of iterations for 387 neurons.

Supervised Learning: MLP

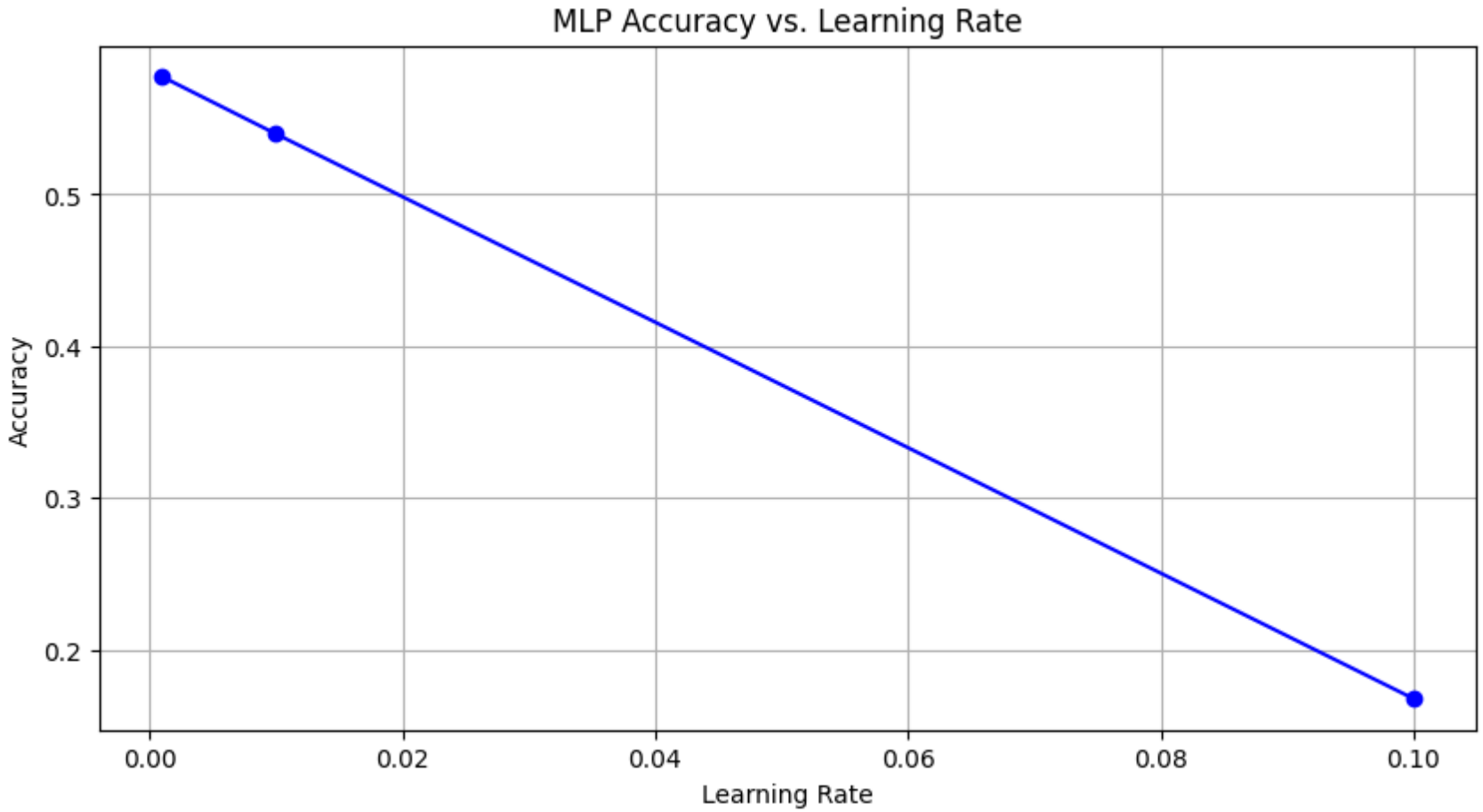

Figure: MLP execution with 70/30 holdout, 437 neurons, 300 iterations, varying the learning rate.

Supervised Learning: MLP

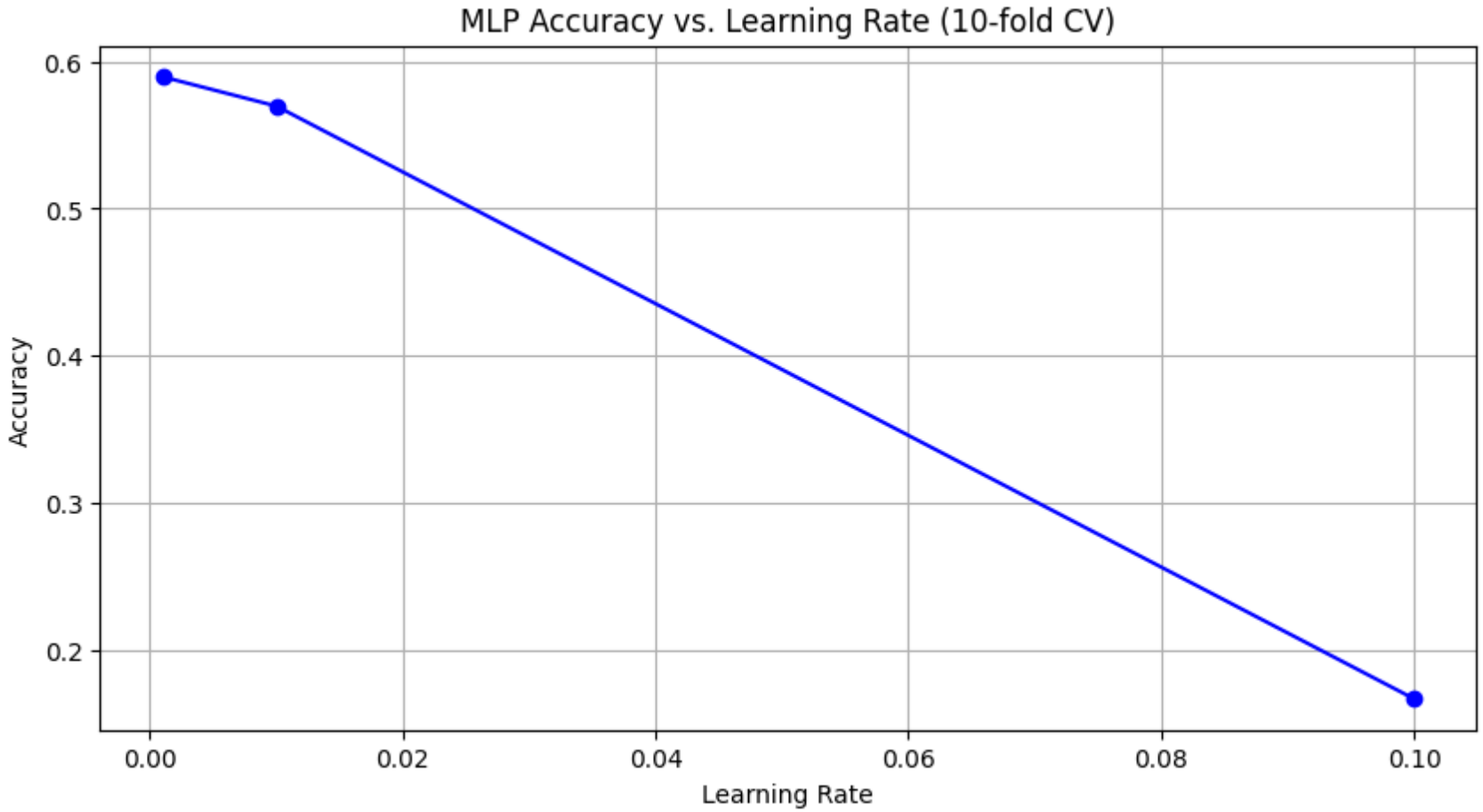

Figure: MLP execution with 10-fold cross-validation, 387 neurons, 300 iterations, varying the learning rate.

Supervised Learning: MLP

After completing these experiments, the best configuration found so far was using the following parameters:

- Number of neurons: 387
- Maximum number of iterations: 300
- Learning rate: 0.001
- Best results found through 10-fold cross-validation

Supervised Learning: MLP

Dataset	MLP Accuracy
Original Dataset	58.96
Reduced Dataset 1	58.26
Reduced Dataset 2	16.49
Reduced Dataset 3	38.97

Table 8. MLP accuracy with the local optimal configuration found.

Supervised Learning: MLP with GridSearch

The execution of GridSearch returned the following configuration as the best set of parameters:

- Neurons: 387
- Learning rate: 0.01
- Maximum iterations: 300
- Random State: 50
- Solver: SGD

Supervised Learning: MLP with GridSearch

Dataset	MLP Accuracy (GridSearch)
Original Dataset	59.95
Reduced Dataset 1	57.50
Reduced Dataset 2	16.61
Reduced Dataset 3	40.19

Table: MLP Accuracy for Different Datasets (GridSearch)

Supervised Learning: MLP with GridSearch

The parameters found by GridSearch were similar to those found during the progression of experiments. The notable differences are in the learning rate found by GridSearch, which was 0.01 (versus 0.001) and the solver, which was SGD (versus ADAM, which was the default in other executions).
The differences in accuracy were:

- Original Dataset: GridSearch presented an improvement of 0.99% in accuracy.
- Reduced Dataset 1: GridSearch presented a decrease of 0.76% in accuracy.
- Reduced Dataset 2: GridSearch presented an improvement of 0.12% in accuracy.
- Reduced Dataset 3: GridSearch presented an improvement of 1.22% in accuracy.

Overall, the accuracy increased, even if slightly, through the use of these parameters found with GridSearch.

Supervised Learning: Results

It was observed that, as the number of attributes present in each instance decreased, the accuracy also decreased. This is due to the complexity of the task of sentiment analysis and the loss of representation of important data due to the reduction of attributes.

Dataset	k-NN	Decision Tree (DT)	Naive Bayes (NB)	MLP
Original Dataset	35.03	29.28	39.50	58.96
Reduced Dataset 1	34.28	29.71	40.06	58.26
Reduced Dataset 2	16.15	16.39	16.72	16.49
Reduced Dataset 3	32.27	28.61	41.84	38.97
Overall Average	**29.43**	**25.99**	**34.53**	**43.17**

Table 10. Average accuracies of k-NN, Decision Tree, Naive Bayes and MLP models on all datasets.

Unsupervised Learning

- Algorithms: k-Means, Hierarchical Clustering.
- DB and Silhouette indices used for validation.
- k-Means showed slightly better performance.

Unsupervised Learning: k-Means

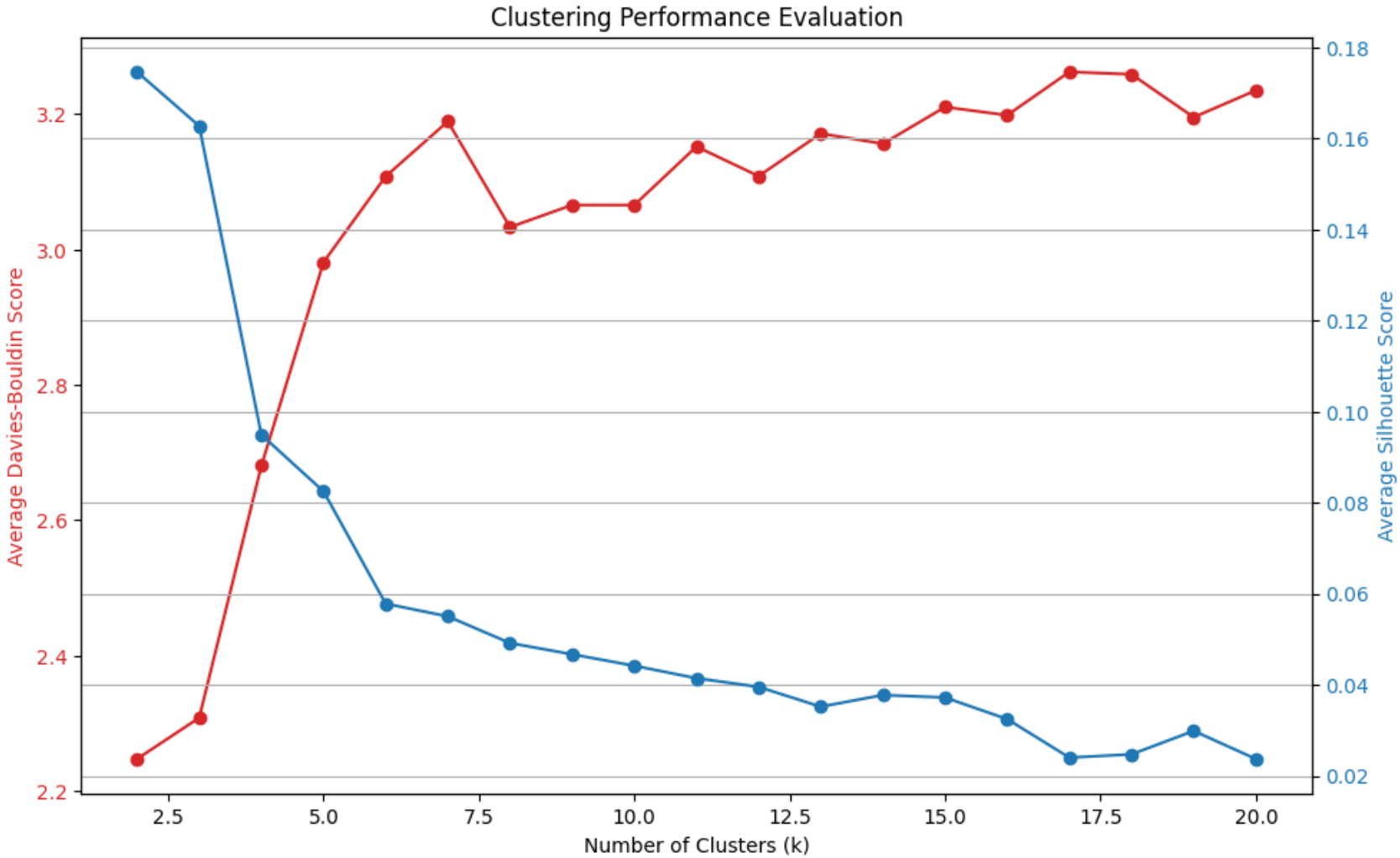

Figure: k-Means Performance

Unsupervised Learning: Hierarchical Clustering

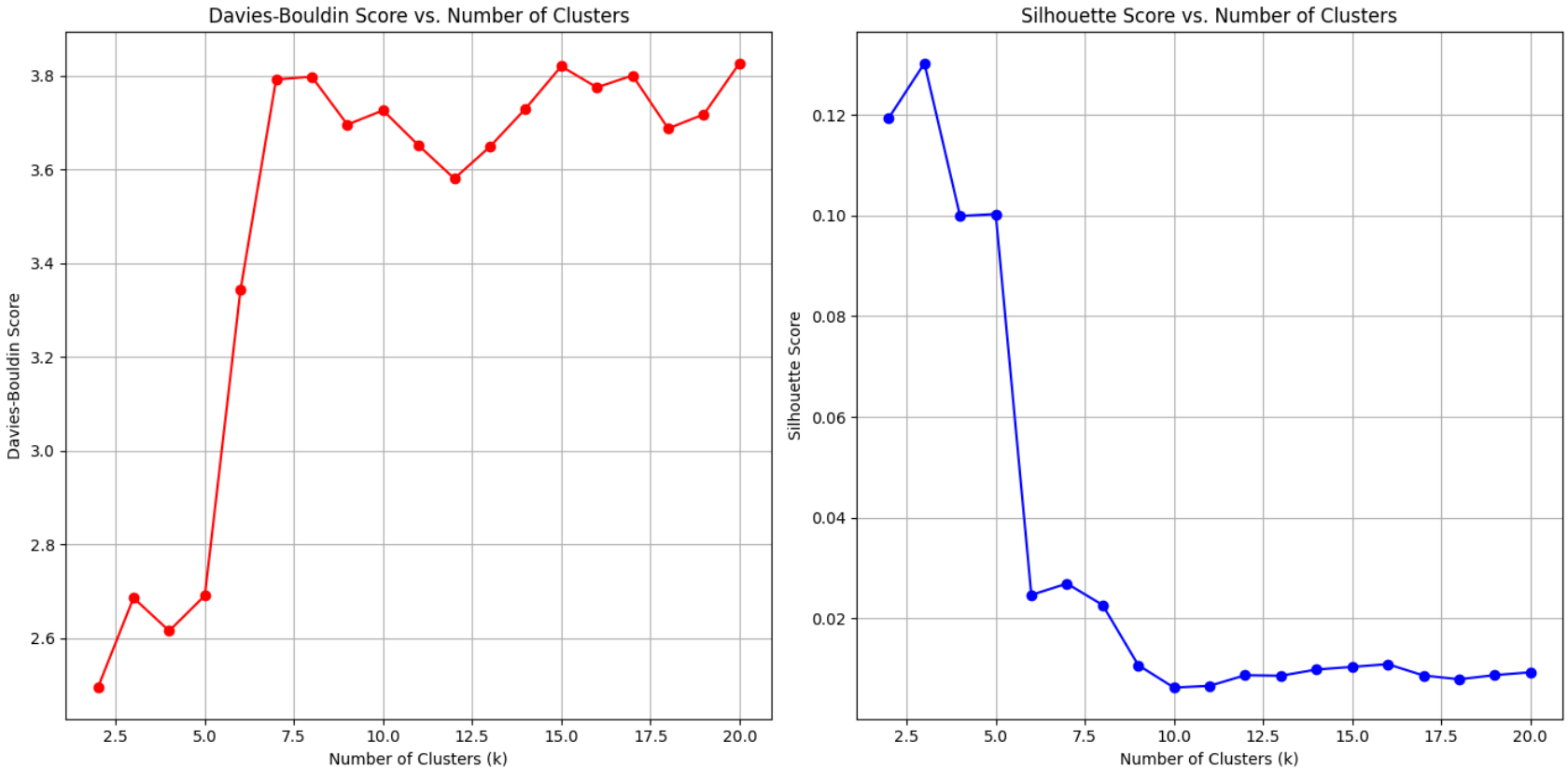

Figure: Hierarchical Clustering with Davies-Bouldin and Silhouette Scores.

Unsupervised Learning: k-Means

After running both algorithms again, each using `k = 2`, the DB and Silhouette indices were calculated. k-Means showed superior performance in terms of both compactness and cluster separation compared to the Hierarchical method. This is evidenced by the higher Silhouette index obtained by k-Means, indicating that the clusters are relatively more compact and well-separated than those formed by hierarchical clustering. Additionally, the lower Davies-Bouldin index for k-Means suggests that the clusters in this method are less internally dispersed and more distinct from each other compared to the hierarchical method, even though in practice they are not significantly different.

Ensemble Learning

- Algorithms: Bagging, Boosting, Random Forest, Stacking.
- Bagging with default and modified configurations.
- Boosting showed mixed performance.
- Random Forest with n_estimators set to 100 showed good accuracy.
- Stacking provided the best performance overall.

Ensemble Learning: Results

Method	AD	kNN	NB	MLP
Bagging Default	36.17	37.80125	39.51875	62.43875
Bagging Max Features 0.3	35.37875	39.17875	39.27375	60.6575
Bagging Max Features 0.5	35.935	38.52875	39.2775	61.30875
Bagging Max Features 0.8	35.67375	38.0925	39.385	61.72875
Boosting Default	31.81375	36.975	39.5425	61.19625
Random Forest	38.265	37.895	37.88	37.54
Stacking	60.1	60.705	61.235	59.885

Table 22. Average accuracy scores for ensemble algorithms on the dataset

Statistical Test

- Friedman test showed significant differences in supervised algorithms' performance.
- Nemenyi post-hoc test revealed no significant differences between pairs of algorithms.
- Wilcoxon test for unsupervised learning showed no significant differences.

Statistical Test: Supervised Learning

The Friedman test was applied to these results, yielding a test statistic of 9.90 and a p-value of 0.0194. Since the p-value is less than the significance level of 0.05, the null hypothesis can be rejected and conclude that there are significant differences in the performance of the algorithms.

Algorithm	k-NN	Decision Tree	Naive Bayes	MLP
k-NN	1.0000	0.9000	0.0656	0.8240
Decision Tree	0.9000	1.0000	0.0656	0.8240
Naive Bayes	0.0656	0.0656	1.0000	0.3549
MLP	0.8240	0.8240	0.3549	1.0000

Table 20. P-values from the Nemenyi post-hoc test

Statistical Test: Supervised Learning

From the results, it is observed that none of the p-values are less than 0.05, indicating no significant differences between any pairs of algorithms. Despite the initial significant differences detected by the Friedman test, the Nemenyi post-hoc test shows that these differences are not significant when comparing each pair of algorithms individ- ually.

Statistical Test: Unsupervised Learning

There are no significant differences in the performance of the k-Means and Hierarchical Clustering algorithms based on these metrics.

Metric	**Wilcoxon Statistic**	**p-value**
DB Index	0.0	1.0
Silhouette Score	0.0	1.0

Table: Wilcoxon signed-rank test results for the DB Index and Silhouette Score

Statistical Test: Ensemble Learning

The Friedman test was conducted to compare the accuracy scores of the ensemble algorithms. The p-value is greater than the significance level of 0.05. Therefore, there are no significant differences in the performance of the ensemble algorithms based on the accuracy scores.
To further confirm this, the Nemenyi post-hoc test was later conducted.

Test	**Statistic**
Friedman test statistic	6.60
p-value	0.0858

Table: Friedman test results

Statistical Test: Ensemble Learning

	0	1	2	3	4	5	6
0	1.00000	0.900000	0.9	0.9	0.900000	0.829610	0.900000
1	0.90000	1.000000	0.9	0.9	0.900000	0.900000	0.637136
2	0.90000	0.900000	1.0	0.9	0.900000	0.900000	0.900000
3	0.90000	0.900000	0.9	1.0	0.900000	0.900000	0.900000
4	0.90000	0.900000	0.9	0.9	1.000000	0.900000	0.540899
5	0.82961	0.900000	0.9	0.9	0.900000	1.000000	0.440184
6	0.90000	0.637136	0.9	0.9	0.540899	0.440184	1.000000

Table 24. Nemenyi post-hoc test results

Conclusion

- Data reduction was beneficial.
- MLP was the best supervised method.
- Clustering algorithms did not show significant improvements.
- Ensemble classifiers, particularly Stacking, improved performance.
- Statistical tests confirmed the findings.

References I

Kanwal Ahmed, Muhammad Imran Nadeem, Dun Li, Zhiyun Zheng, Yazeed Yasin Ghadi, Muhammad Assam, and Heba G. Mohamed.
Exploiting stacked autoencoders for improved sentiment analysis.
Applied Sciences, 12(23), 2022.

Ali Areshey and Hassan Mathkour.
Transfer learning for sentiment classification using bidirectional encoder representations from transformers (bert) model.
Sensors, 23(11), 2023.

Christopher M. Bishop.
Pattern Recognition and Machine Learning.
Springer, 2006.

References II


David M Blei, Andrew Y Ng, and Michael I Jordan.
Latent dirichlet allocation.
Journal of Machine Learning Research, 3:993–1022, 2003.


R. Boulic and O. Renault.
3d hierarchies for animation.
In Nadia Magnenat-Thalmann and Daniel Thalmann, editors, *New Trends in Animation and Visualization*. John Wiley & Sons ltd., 1991.


Leo Breiman.
Random forests.
Machine Learning, 45(1):5–32, 2001.

References III


Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al.
Language models are few-shot learners.
arXiv preprint arXiv:2005.14165, 2020.


Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al.
Universal sentence encoder.
In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, 2018.

References IV


Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio.
Learning phrase representations using rnn encoder-decoder for statistical machine translation.
arXiv preprint arXiv:1406.1078, 2014.


Thomas M. Cover and Peter E. Hart.
The nearest neighbor problem.
IEEE Transactions on Information Theory, 13(1):21–27, 1967.


Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova.
Bert: Pre-training of deep bidirectional transformers for language understanding.
NAACL-HLT, 1(2019):4171–4186, 2019.

References V


TG Dietterich.
Ensemble methods in machine learning.
Multiple Classifier Systems: First International Workshop, MCS 2000, Lecture Notes in Computer Science, pages 1–15, 01 2000.


Arwa Diwali, Kawther Saeedi, Kia Dashtipour, Mandar Gogate, Erik Cambria, and Amir Hussain.
Sentiment analysis meets explainable artificial intelligence: A survey on explainable sentiment analysis.
IEEE Transactions on Affective Computing, pages 1–12, 2023.


Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu.
A density-based algorithm for discovering clusters in large spatial databases with noise.

References VI

In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. AAAI Press, 1996.


Felix A Gers, Jürgen Schmidhuber, and Fred Cummins.
Learning to forget: Continual prediction with lstm.
Neural Computation, 12(10):2451–2471, 1999.


Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
Deep Learning.
MIT Press, 2016.


Yoon Kim.
Convolutional neural networks for sentence classification.
In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.

References VII


Diederik P Kingma and Max Welling.
Auto-encoding variational bayes.
arXiv preprint arXiv:1312.6114, 2014.


Donald E. Knuth.
The TEX Book.
Addison-Wesley, 15th edition, 1984.


Songning Lai, Xifeng Hu, Haoxuan Xu, Zhaoxia Ren, and Zhi Liu.
Multimodal sentiment analysis: A survey, 2023.


Quoc Le and Tomas Mikolov.
Distributed representations of sentences and documents.
In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196, 2014.

References VIII


Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov.
Roberta: A robustly optimized bert pretraining approach.
arXiv preprint arXiv:1907.11692, 2019.


Yishu Miao, Lei Yu, and Phil Blunsom.
Neural variational inference for text processing.
arXiv preprint arXiv:1511.06038, 2016.


Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean.
Efficient estimation of word representations in vector space.
arXiv preprint arXiv:1301.3781, 2013.

References IX

Samreen Naeem, Aqib Ali, Sania Anam, and Munawar Ahmed.

An unsupervised machine learning algorithms: Comprehensive review.
IJCDS Journal, 13:911–921, 04 2023.

Vladimir Nasteski.
An overview of the supervised machine learning methods.
HORIZONS.B, 4:51–62, 12 2017.

Jeffrey Pennington, Richard Socher, and Christopher D Manning.
Glove: Global vectors for word representation.
In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

References X


J. Ross Quinlan.
Induction of decision trees.
Machine Learning, 1(1):81–106, 1986.


Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu.
Exploring the limits of transfer learning with a unified text-to-text transformer.
Journal of Machine Learning Research, 21(140):1–67, 2020.


Irina Rish.
An empirical study of the naive bayes classifier.
In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, volume 3, pages 41–46. IBM, 2001.

References XI


Mohammad Rostami and Aram Galstyan.
Domain adaptation for sentiment analysis using increased intraclass separation.
CoRR, abs/2107.01598, 2021.


David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams.
Learning Internal Representations by Error Propagation, volume 1.
MIT Press, 1986.


A. Smith and B. Jones.
On the complexity of computing.
In A. B. Smith-Jones, editor, *Advances in Computer Science*, pages 555–566. Publishing Press, 1999.

References XII


David H. Wolpert.
Stacked generalization.
Neural Networks, 5(2):241–259, 1992.


Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le.
XInet: Generalized autoregressive pretraining for language understanding.
arXiv preprint arXiv:1906.08237, 2019.