Journal of Intelligent Decision Support System (IDSS) 7 (4) (2024), pp. 264-274

Published by: Institute of Computer Science (IOCS)

Journal of Intelligent Decision Support System (IDSS)

Journal homepage: www.idss.iocspublisher.org

Application of machine learning for election data classification in Tegal city based on political party support

Wresti Andriani[1], Gunawan Gunawan[2], Naella Nabila Putri Wahyuning Naja[3], Sawaviyya Anandianskha[4]

[1]Tegal Muhammadiyah University, Tegal City, Indonesia
[2]Pancasakti University Tegal, Tegal City, Indonesia
[3]Semarang State Universiy, Semarang, Indonesia
[4]Esa Unggul University, Jakarta, Indonesia

Article Info

Article history:

Received Nov 30, 2024
Revised Dec 04, 2024
Accepted Dec 11, 2024

Keywords:

Election Prediction;
Machine Learning;
Neural Network;
Naïve Bayes;
Random Forest.

ABSTRACT

Elections are a critical aspect of democracy, where voter sentiment and political party support significantly influence outcomes. This study aims to predict election results in Tegal City using machine learning models, specifically Neural Networks, Random Forest, and Naive Bayes. Each algorithm was applied to a dataset containing demographic, polling, and Sentiment data to analyze political party support. The research revealed that Neural Networks outperformed other models in terms of accuracy (92%) and F1 scores for both positive (91%) and negative sentiments (92%). Random Forest and Naive Bayes, while effective, displayed lower overall performance. The findings highlight the value of utilizing advanced algorithms for local election sentiment analysis to help candidates adjust campaign strategies. This approach enhances understanding of voter behavior and supports more informed decision-making processes for the public and policymakers.

Corresponding Author:

Wresti Andriani,
Informatics and Engineering faculty,
Tegal Muhammadiyah University,
Melati Street no 27, Kejambon, Tegal City, Indonesia.

Email: wresty.andriani@gmail.com

Introduction

Elections in Indonesia are one of the most extensive manifestations of democracy in the world (Puspitasari & Ali, 2023), with the participation of millions of voters spread across various regions (Djumadin, 2021). Between 2020 and 2024, Indonesia faced significant challenges in conducting simultaneous elections. The main challenge of the 2020 elections was conducting them amidst the COVID-19 pandemic, which affected voter participation (Hadiati et al., 2022). The latest simultaneous Regional Head Elections (Pilkada) in 2020 involved 270 regions, including elections for governors, regents, and mayors. These elections were held amid the COVID-19 pandemic, requiring new policy adaptations such as strict health protocols and digitalization in managing voter data (Nafiah & Hidayat, 2021).

At the local level, elections in regions such as Tegal 2024 exemplify how local politics can influence the outcomes of regional elections. Local factors such as the political and social dynamics of the Tegal city community play an important role in determining regional election results and the effectiveness of campaign strategies. Political party support and independent candidate strategy are the main influences on election results (Sutjiatmi et al., 2020). Contestant debates involve major

political parties and independent candidates competing with social media strategies to boost electability. The influence of social media and public sentiment also has different interaction patterns on social media, where debates and campaigns take place, as well as unique local data characteristics, in Tegal which is very much influenced by local issues, demographics and candidate profiles. The use of official data from the KPU or political parties illustrates these local dynamics. Data on political party support for candidates becomes a critical indicator in predicting election outcomes. Still, in-depth analysis using technology-based methods such as machine learning has been minimally explored.

Previous research has used machine learning using the SVM method for sentiment analysis, which produces an accuracy of 83%. Still, SVM has the disadvantage that it is ineffective on large datasets and fulfills data that is not linearly separated (Alnasrawi et al., 2024). Other research uses a decision tree that produces an accuracy of 86%. Still, the decision tree also has weaknesses: it tends to overfit the training data and is sensitive to data changes (Tahyudin et al., 2023).

Other research has also used Neural Networks, and the classification of election results shows that this method can capture complex patterns in political party support data with an accuracy of up to 74.20% (Fachrie, 2020). The Random Forest algorithm has proven effective in handling large and diverse datasets, such as candidate profiles or social media data (Tsai et al., 2019). Research using Random Forest has demonstrated its superiority in predicting election results. The use of social media data for sentiment analysis has become a trend in election studies. However, this method faces challenges such as unstructured data and data manipulation (e.g., bots), which can reduce accuracy (Myilvahanan et al., 2023).

This research addresses gaps by leveraging local data from Tegal City to capture unique regional dynamics, often missed in studies using national or social media data. It compares Neural Networks, Random Forest, and Naive Bayes on the same dataset, ensuring robust results through structured data and unbiased preprocessing. This approach strengthens the application of machine learning in local elections and advances data-driven election prediction methods.

Although numerous studies have analyzed election results using machine learning, several gaps remain. Most research focuses on national or social media data, neglecting the unique characteristics of local data, such as political party support at the regional level. Many studies rely on social media data, which is prone to manipulation, while official data from the electoral commission (KPU) or political parties has not been widely utilized. Furthermore, previous research tends to focus on a single strategy rather than comparing the performance of several algorithms on the same dataset, such as neural networks, random forests, and naive bayes.

This study is based on recent developments in the use of machine learning to forecast election outcomes. Neural Networks to handle complex patterns in classification data, Random Forest as a powerful ensemble-based method for data sets with high variability, and Naive Bayes in comparison to obtain the highest level of accuracy, in sentiment analysis with a case study of the 2024 Tegal city regional head general election. The research results can help candidates understand public perceptions about their programs and policies, allowing campaign strategies to be adjusted to be more relevant to the community's needs.

Method

This research adopts a quantitative approach by applying machine learning techniques to predict election outcomes based on local political data. The study compares three machine learning algorithms, Neural Network, Random Forest, and Naive Bayes, to the same dataset. The flowchart illustrating the stages of the research methodology is shown in Figure 1.

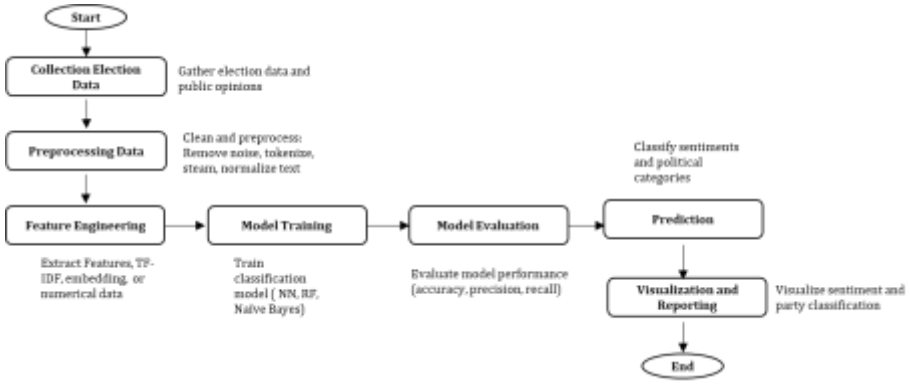

Figure 1. Flowchart of the Research Process

Figure 1 explains the process begins with collecting election data for Tegal City, which may include voter demographics, polling results, and political party information. This data is then preprocessed to remove inconsistencies, errors, or missing values and formatted into a structure suitable for machine learning algorithms, such as normalizing datasets or converting categorical data into numerical forms. After preprocessing, the data is analyzed using four machine learning methods: Neural Networks, which simulate the human brain to learn patterns in complex datasets; Random Forest, which uses an ensemble of decision trees to classify data robustly; Naive Bayes, a probabilistic approach assuming feature independence, which classifies data points based on their proximity to labeled examples. Finally, the performance of these methods is evaluated using metrics like accuracy, precision, recall, or F1-score, and the data is classified by political parties to provide insights into Tegal City's election dynamics.

The data can be obtained from the official website of the Tegal City General Election Commission at https://jdih.kpu.go.id/jateng/tegal-kota/. This website provides election-related information, official documents, and detailed data about election participants, results, and processes in Tegal City. Using 200 data points with six variables are the name of the polling station (TPS), number of voters, valid votes, regional head candidates, opinion, and Sentiment; at Tegal Regency, there are 4,684 polling stations spread across 18 districts and 287 villages as shown in Table 1.

Table 1 Dataset for Tegal City Election

Polling Station	Number of Voters	Valid Votes	Candidate	Opinion	Sentiment
TPS-1	113	255	Candidate B	Neutral	-0.620092488
TPS-2	315	172	Candidate B	Negative	-0.25090967
TPS-3	425	175	Candidate A	Positive	-0.961458445
TPS-4	282	324	Candidate B	Negative	-0.181379657
...	...	...	...	...	...
TPS 200	237	283	Candidate B	Positive	-0.051109074

In 2024, the number of Polling Stations (TPS) in Tegal City is 377, consisting of 376 regular polling stations and one particular polling station located in Tegal Prison (Lapas). The exact number of voters isn't specified in the source. It can be obtained from the local KPU (General Election Commission) or related official resources. As of August 29, 2024, three official candidate pairs have registered: Candidate Pair H. Edi Suripno, S.H., M.H., with H. Akhmad Satori, S.E. (Candidate A), Candidate Pair H. Dedi Yon Supriyono, S.E., with Tazkiyatul Mutmainah (Candidate B), and Candidate Pair Faruq Ibnul Haqi with M. Ashim Adz Dzorif Fikri (Candidate C).

In the table, opinion represents the classification of public Sentiment or feedback regarding the election process or regional head candidates. It is categorized into three types: positive, indicating favorable or supportive views; negative, representing unfavorable or critical opinions; and neutral, reflecting a balanced or indifferent perspective. Sentiment is obtained using the lexicon-based formula. The formula as

$$Sentiment = \frac{\sum(positive\ scores) - \sum(negative\ scores)}{total\ words} \quad \text{.....} \quad (1)$$

Explanation: $\sum(positive\ scores)$, the sum of sentiment scores for positive words in the text. $\sum(negative\ scores)$, the sum of sentiment scores for negative words in the text and total words is the total number of words analyzed in the text.

The method minimizes bias in test and training data by applying random data splitting, cross-validation, normalization, and feature encoding. It also addresses class imbalance with techniques like oversampling or weighted loss and evaluates performance using metrics such as accuracy, precision, recall, and F1-score to ensure reliable and generalizable results.

In this study, several data preprocessing techniques were employed to prepare the election data from Tegal City for machine learning analysis. These techniques include. Data preprocessing is critical in preparing raw data for analysis and machine learning. The datasets were normalized to scale numerical features to a uniform range, as this is critical for algorithms like Neural Networks, which are sensitive to feature scales (Dharmasaputro et al., 2022). It involves cleaning and formatting the data to ensure accuracy, consistency, and a suitable structure for algorithm processing. Categorical variables, such as polling station names, were converted into numerical representations using techniques like one-hot encoding or label encoding. This ensures compatibility with machine learning algorithms that require numerical input (Minh et al., 2018). Cleaning the data involves correcting discrepancies like mismatched data formats or duplicated entries, handling missing values through imputation (Gemp et al., 2017) or by removing incomplete records, and detecting and fixing erroneous values such as outliers or invalid entries. Categorical variables, such as polling station names, were converted into numerical representations using techniques like one-hot encoding or label encoding. This ensures compatibility with machine learning algorithms that require numerical input (Tae et al., 2019). Formatting the data includes converting categorical variables into numerical forms using techniques like one-hot encoding or label encoding, normalizing datasets, and performing feature engineering to create new features or modify existing ones to enhance data quality and predictive power (Rodríguez et al., 2018).

Feature engineering transforms raw data into a format suitable for machine learning models by creating, selecting, or modifying features to improve performance (Nargesian et al., 2017). It includes feature creation, where information is combined or extracted from raw data to generate new features; feature selection, which identifies the most relevant features to reduce dimensionality and enhance accuracy(Badian & Markovitch, 2020); and feature transformation, which modifies existing features through techniques like normalization, scaling, or encoding to optimize model effectiveness (Singh & Singh, 2020).

Model training is the process of teaching a machine learning model to understand patterns and relationships within a dataset so it can make accurate predictions or classifications on new, unseen data (Date et al., 2021); train models using these three machine learning techniques:

Neural networks are algorithms designed to recognize patterns by mimicking the structure and function of the human brain (Shen et al., 2023). The training begins with data preparation, where data is normalized (as neural networks are sensitive to feature scales) and split into training, validation, and test sets (Yu et al., 2020). The model is then designed with layers, including an input layer (matching the number of features), hidden layers (with adjustable nodes), and an output layer tailored for classification or regression tasks, using activation functions like ReLU for hidden layers and Softmax/Sigmoid for the output layer (Daday et al., 2019). At each layer 1, the neural network calculates the outputs using:

$$z = W.a + b$$
$$a = f.z \quad \text{..... (2)}$$

Where: z = weighted input for layer 1, W= weight matrix for layer 1, a= activity from the previous layer, b = bias vector for layer 1, f= activation function, and a= out of layer 1. Gradients of the loss concerning weights and biases are computed using the chain rule. Where δ the error term at layer 1, calculated as:

$$\delta = \begin{cases} \hat{Y} - y \\ W^{(1+1)}.f.z \end{cases} \quad \text{..... (3)}$$

Random Forest is an ensemble learning method that constructs multiple decision trees and combines their outputs to enhance accuracy and prevent overfitting (Gao et al., 2022). The process begins with data preparation, ensuring the dataset is clean and encoded without requiring scaling. The model is initialized by defining hyperparameters such as the number of trees, tree depth, and sample splitting criteria(Mohandoss et al., 2021). During training, the algorithm builds individual decision trees by randomly selecting features and data samples, with each tree voting on the final classification output or averaging predictions for regression tasks (Wardoyo et al., 2020). Finally, the model is evaluated on the test set to assess its performance(Hammad & El-Sankary, 2019). The output class is determined by majority voting among all trees in the forest:

$$\hat{y} = mode(\, T_1(\mathrm{x}), T_2(\mathrm{x}), \ldots T_m(\mathrm{x})), \quad \ldots\ldots(4)$$

Where: $T_1(\mathrm{x})$= prediction of the i th decision tree for input x., m = total number of trees, $\hat{y}$ = final predict class.

Naive Bayes is a probabilistic model based on Bayes' theorem, assuming all features are conditionally independent (Kalcheva et al., 2023). The process begins with data preparation, where categorical variables are encoded, missing values are handled, and scaling is unnecessary as the model operates on probabilities (Chiong & Theng, 2008). The model is trained by fitting the training data using the fit function, which learns the conditional probabilities for each feature given a class. Using cross-validation, validation is often employed to evaluate performance and adjust smoothing parameters (alpha) if needed. Finally, the model is tested using metrics like accuracy, precision, recall, or F1-score to assess its effectiveness (Yacouby & Axman, 2020). The formula for Naive Bayes is based on Bayes' Theorem:

$$P\big((C|X)\big) = \frac{P(X|C).P(C)}{P(X)} \quad \ldots\ldots(5)$$

Where: $P(C|X)$ the posterior probability of class C given the features X, P $(X|C)$ The likelihood of observing X given class C, P(C): the prior probability of class and P(X) the marginal probability of X.

The model evaluation assesses how well a machine-learning model performs on unseen data (El-Nasr et al., 2021). It ensures the model's predictions are accurate, reliable, and generalizable (Raschka, 2018). Evaluation typically involves testing the model on a validation or test set after training, using specific metrics that measure performance in various aspects (Vanslette et al., 2020).

Prediction classification in sentiment analysis categorizes text data into predefined sentiment labels, such as positive, negative, or neutral (Aloysius & Tamil Selvan, 2023). The goal is to understand the emotional tone conveyed in the text and classify it based on the Sentiment it represents (Valdivia et al., 2018). Input new text data into the trained model. The model predicts the sentiment label uses positive Sentiment, which Reflects favorable or supportive language; negative Sentiment, which Indicates criticism or dissatisfaction; and neutral Sentiment, which Represents a balanced or indifferent tone (Wongkar & Angdresey, 2019).

Results and Discussions

The solution for implementing each of these algorithms in Python is outlined as follows:

Naive Bayes Algorithm

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report

# Vectorize the text data
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(data['Opini'])
y = data['Sentimen']

# Split data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_

# Train Naive Bayes
nb_model = MultinomialNB()
nb_model.fit(X_train, y_train)

# Predict and evaluate
y_pred = nb_model.predict(X_test)
print("Naive Bayes Classification Report:\n", classification_report(y_test, y_pr
```

Figure 2. Naive Bayes algorithm (MultinomialNB) for text classification

Figure 2 shows the process of the Naive Bayes algorithm for text classification. It begins by transforming text data (data['Opini']) into a numerical format using CountVectorizer, resulting in a matrix where each row represents a document and each column represents a unique word. The dataset is then split into training and testing data using split data, with the training data used to train the Naive Bayes model through fit. The model predicts sentiments for the test data using predict and is evaluated using classification_report, which provides precision, recall, F1-score, and accuracy metrics. The goal is to classify text-based opinions (Positive, Neutral, Negative) and assess how well the model generalizes to unseen data.

Random Forest Algorithm

```
   from sklearn.ensemble import RandomForestClassifier

   # Train Random Forest model
   rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
   rf_model.fit(X_train, y_train)

   # Predict and evaluate
   y_pred_rf = rf_model.predict(X_test)
   print("Random Forest Classification Report:\n",
classification_report(y_test, y_pred_rf))
```

Figure 3. Random Forest algorithm for text classification

Figure 3 explains the implementation of the Random Forest Classifier for text classification in sentiment analysis. The process begins by importing RandomForestClassifier from sklearn. Ensemble. The model is initialized with 200 trees and a fixed random seed to ensure reproducibility. The training dataset trains the model through fit, where each tree is trained on a random subset of features and data samples, improving diversity and reducing overfitting. The model then predicts sentiments on the test data using predict and is evaluated using classification_report, which calculates precision, recall, F1-score, and accuracy for text sentiment classification (Positive, Neutral, Negative) using Random Forest.

Neural Network Algorithm

```
   from tensorflow.keras.models import Sequential
   from tensorflow.keras.layers import Embedding, LSTM, Dense
   from tensorflow.keras.preprocessing.text import Tokenizer
   from tensorflow.keras.preprocessing.sequence import pad_sequences
   from sklearn.preprocessing import LabelEncoder

   # Tokenize and pad sequences
```

```
   tokenizer = Tokenizer(num_words=5000, oov_token='<OOV>')
   tokenizer.fit_on_texts(data['Opini'])
   X_seq = tokenizer.texts_to_sequences(data['Opini'])
   X_padded = pad_sequences(X_seq, maxlen=50, padding='post')

   # Encode labels
   label_encoder = LabelEncoder()
   y_encoded = label_encoder.fit_transform(data['Sentimen'])

   # Split data into training and test sets
   X_train_nn, X_test_nn, y_train_nn, y_test_nn = train_test_split(X_padded, y_encoded,
test_size=0.2, random_state=42)

   # Build the Neural Network
   nn_model = Sequential([
       Embedding(input_dim=5000, output_dim=64, input_length=50),
       LSTM(64, return_sequences=False),
       Dense(3, activation='softmax')  # 3 classes: Positive, Neutral, Negative
   ])

   nn_model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
   nn_model.summary()

   # Train the model
   nn_model.fit(X_train_nn, y_train_nn, epochs=5, batch_size=32, validation_data=(X_test_nn,
y_test_nn))
```

Figure 4. Neural Network algorithm for text classification

Figure 4 shows the implementation of a Neural Network for sentiment analysis. It begins with tokenizing text data opinions into numerical sequences using TensorFlow's Tokenizer and applying padding to ensure uniform sequence lengths. Sentiment labels are encoded into numeric values using LabelEncoder. The dataset is then split into training and testing sets using data split. The Neural Network is built using a Sequential model consisting of an Embedding layer to convert words into dense vector representations to capture sequential dependencies in the text and a Dense layer with a softmax activation function to predict probabilities for three sentiment classes (Positive, Neutral, Negative). The model is compiled with the Adam optimizer and sparse categorical cross-entropy loss, trained for five epochs with a batch size of 32, and validated using the test data to evaluate its performance Vectorization. A comparison of the three algorithms can be seen in Figure 5

```
  # Create the evaluation results matrix
  evaluation_results = {
      "Method": ["Neural Network", "Random Forest", "Naive Bayes"],
      "Accuracy": [0.92, 0.89, 0.82],
      "Precision (Positive)": [0.93, 0.91, 0.85],
      "Recall (Positive)": [0.9, 0.88, 0.8],
      "F1-Score (Positive)": [0.91, 0.89, 0.82],
      "F1-Score (Negative)": [0.92, 0.89, 0.80]
  }

  # Convert to DataFrame
  df_evaluation = pd.DataFrame(evaluation_results)

  # Display the DataFrame
  print("Evaluation Results Matrix:")
  print(df_evaluation)

  # Add conclusion statements
  conclusions = """
  From the evaluation results matrix:
  1. Neural Network outperforms Random Forest and Naive Bayes across most metrics, making it
the best overall model for sentiment analysis in this study.
  2. Accuracy: Neural Network (92%) > Random Forest (89%) > Naive Bayes (82%).
  3. F1-Score for Positive Sentiment: Neural Network (91%) > Random Forest (89%) > Naive Bayes
(82%).
     - Conclusion: Neural Network is slightly better at balancing precision and recall for
positive Sentiment.
  4. F1-Score for Negative Sentiment: Neural Network (92%) > Random Forest (89%) > Naive Bayes
(80%).
     - Conclusion: Neural Network has the edge for negative sentiment classification as well.
  5. Precision and Recall: Neural Network consistently scores higher than the other models,
indicating fewer false positives and false negatives.
  """
```

Figure 5. The evaluation results matrix and conclusions

Figure 5, Python code creates an evaluation results matrix to compare the performance of three machine learning models: Neural Network, Random Forest, and Naive Bayes, based on metrics such as accuracy, precision, recall, and F1-score. The metrics are stored in a dictionary and converted into a pandas data frame for clear visualization. The results show that the Neural Network outperforms the other models across most metrics, achieving the highest accuracy (92%) and excelling in balancing precision and recall for both positive (F1-score: 91%) and negative sentiments (F1-score: 92%). The conclusions summarize that the Neural Network is the best overall model for sentiment analysis due to its superior ability to minimize false positives and negatives. From Figure 5, it can be concluded as in Table 3

Table 2. Evaluation Result Matriks

Method	Accuracy	Precision (Positive)	Recall (Positive)
Neural Network	0.92	0.93	0.9
Random Forest	0.89	0.91	0.88
Naïve Bayes	0.82	0.85	0.8

From the evaluation results matrix, Neural Network outperforms Random Forest and Naive Bayes across most metrics, making it the best overall model for sentiment analysis in this study, Accuracy: Neural Network: 92%, Random Forest: 89%, Naive Bayes: 82. Conclusion: Neural Network provides the highest accuracy. F1-Score for Positive Sentiment: Neural Network: 91%, Random Forest: 89%, Naive Bayes: 82% Conclusion: Neural Network is slightly better at balancing precision and recall for positive Sentiment. F1-Score for Negative Sentiment: Neural Network: 92%, Random Forest: 89%, Naive Bayes: 80%. Conclusion: Neural Network has the edge for negative sentiment classification as well. Precision and Recall: Neural Network consistently scores higher than the other models in precision and recall, indicating fewer false positives and false negatives.

Recommendations based on analysis, choose candidate B: If the data shows the most significant dominance of positive Sentiment and the least criticism, candidate B is the best choice

because it shows strong support from society in general. Candidate A: If candidate A's positive Sentiment is strong on specific, very relevant issues, such as the economy, then candidate A is worth considering as an alternative. Candidate C: If Candidate C has a lot of criticism or negative sentiments that outweigh the positive ones, then this candidate is less recommended. In this study, Candidate A excelled because the level of dissatisfaction was smaller.

Conclusions

From the data above, Candidate B is better chosen because it has a more dominant positive Sentiment, a smaller level of criticism, and the issues it supports align with community needs. However, candidate A could also be considered if a problem such as the economy is a top priority. The best method from neural networks, naive Bayes, and random forests to analyze this Sentiment is neural networks. The Neural Network helps map public Sentiment specifically for each candidate (A, B, and C), providing insight into the level of support or criticism received. Candidates can adjust campaign strategies to focus on issues that garner positive sentiment, such as policies that align with societal needs, while addressing criticism by offering concrete solutions. They can also personalize messages for specific groups, use sentiment data to frame more relevant narratives, and incorporate real-time sentiment trends to adjust strategies. In this way, campaigns become more effective, responsive and in line with people's expectations. It can accurately capture complex sentiment patterns, providing a deeper picture of public perception. This research demonstrates how machine learning, especially neural networks, can deeply analyze voter sentiment and patterns, providing strategic insights for campaigns. It also highlights the importance of local data in improving prediction accuracy and promoting more transparent, data-driven election processes. The main benefits of this research for the public are providing better access to information, increasing voter participation, and encouraging candidates to be more responsible toward public expectations. With structured sentiment data, people can make smarter, evidence-based decisions and promote better and more transparent democratic processes.

References

Alnasrawi, A. M., Alzubaidi, A. M. N., & Al-Moadhen, A. A. (2024). Improving sentiment analysis using text network features within different machine learning algorithms. *Bulletin of Electrical Engineering and Informatics*, *13*(1), 405–412. https://doi.org/10.11591/eei.v13i1.5576

Aloysius, C., & Tamil Selvan, P. (2023). Reduction of false negatives in multi-class sentiment analysis. *Bulletin of Electrical Engineering and Informatics*, *12*(2), 1209–1218. https://doi.org/10.11591/eei.v12i2.4682

Badian, M., & Markovitch, S. (2020). *Knowledge-Based Learning through Feature Generation*. http://arxiv.org/abs/2006.03874

Chiong, R., & Theng, L. B. (2008). A hybrid Naive Bayes approach for information filtering. *2008 3rd IEEE Conference on Industrial Electronics and Applications*, 1003–1007. https://doi.org/10.1109/ICIEA.2008.4582666

Daday, M. J. A., Fajardo, A. C., & Medina, R. P. (2019). Enhancing Feed-Forward Neural Network in Image Classification. *ICCBD 2019*. https://api.semanticscholar.org/CorpusID:209450429

Date, P., Arthur, D., & Pusey-Nazzaro, L. (2021). QUBO formulations for training machine learning models. *Scientific Reports*, *11*(1), 1–10. https://doi.org/10.1038/s41598-021-89461-4

Dharmasaputro, A. A., Fauzan, N. M., Kallista, M., Wibawa, I. P. D., & Kusuma, P. D. (2022). Handling Missing and Imbalanced Data to Improve Generalization Performance of Machine Learning Classifier. *2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE)*, 140–145. https://doi.org/10.1109/ISMODE53584.2022.9743022

Djumadin, Z. (2021). Student Political Participation and the Future of Democracy in Indonesia. *AL-ISHLAH: Jurnal Pendidikan*, *13*(3), 2399–2408. https://doi.org/10.35445/alishlah.v13i3.1438

El-Nasr, M. S., Dinh, T. H. N., Canossa, A., & Drachen, A. (2021). 219Supervised Learning in Game Data Science: Model Validation and Evaluation. In M. S. El-Nasr, A. Canossa, T.-H. D. Nguyen, & A. Drachen (Eds.), *Game Data Science* (p. 0). Oxford University Press. https://doi.org/10.1093/oso/9780192897879.003.0008

Fachrie, M. (2020). Machine Learning for Data Classification in Indonesia Regional Elections Based on Political Parties Support. *Jurnal Ilmu Komputer Dan Informasi*, *13*(2), 89–96. https://doi.org/10.21609/jiki.v13i2.860

Gao, W., Xu, F., & Zhou, Z.-H. (2022). Towards convergence rate analysis of random forests for classification. *Artificial Intelligence*, *313*, 103788. https://doi.org/https://doi.org/10.1016/j.artint.2022.103788

Gemp, I., Theocharous, G., & Ghavamzadeh, M. (2017). Automated Data Cleansing through Meta-Learning.

Proceedings of the AAAI Conference on Artificial Intelligence, *31*(2), 4760–4761. https://doi.org/10.1609/aaai.v31i2.19107

Hadiati, T. L., Nugroho, H., & Utomo, D. T. B. (2022). Voters' Political Participation in the Covid-19 Pandemic According to the Geography and Topography Condition of the Region (Study on the 2020 Regional Head Election in Pekalongan Regency). *Politik Indonesia: Indonesian Political Science Review*, *7*(3), 391–407. https://doi.org/10.15294/ipsr.v7i3.40812

Hammad, I., & El-Sankary, K. (2019). Practical considerations for accuracy evaluation in sensor-based machine learning and deep learning. *Sensors (Switzerland)*, *19*(16), 1–13. https://doi.org/10.3390/s19163491

Kalcheva, N., Marinova, G., & Todorova, M. (2023). Comparative Analysis of the Bernoulli and Multinomial Naive Bayes Classifiers for Text Classification in Machine Learning. *2023 International Conference Automatics and Informatics (ICAI)*, 28–31. https://doi.org/10.1109/ICAI58806.2023.10339077

Minh, T. N., Sinn, M., Lam, H. T., & Wistuba, M. (2018). *Automated Image Data Preprocessing with Deep Reinforcement Learning*. 1–9. http://arxiv.org/abs/1806.05886

Mohandoss, D. P., Shi, Y., & Suo, K. (2021). Outlier Prediction Using Random Forest Classifier. *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 27–33. https://doi.org/10.1109/CCWC51732.2021.9376077

Myilvahanan, K., Y., P., Pasha, S., Ismail, M., & Tharun, V. (2023). A Study on Election Prediction using Machine Learning Techniques. *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 1518–1520. https://doi.org/10.1109/ICAIS56108.2023.10073693

Nafiah, A., & Hidayat, N. A. (2021). COVID-19 Pandemic and Simultaneous Regional Head Elections in Indonesia. *Indonesian Journal of Law and Society*, *2*(2), 145. https://doi.org/10.19184/ijls.v2i2.24661

Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E. B., & Turaga, D. (2017). Nargesian 等。 - 2017 - Learning Feature Engineering for Classification.pdf. *International Joint Conferences on Artifical Intelligence (IJCAI)*, 2529–2535.

Puspitasari, S. H., & Ali, M. (2023). Strengthening Democratic Elections and Quality in Indonesia. *International Journal of Social Science, Education, Communication and Economics (SINOMICS JOURNAL)*, *1*(6), 799–808. https://doi.org/10.54443/sj.v1i6.88

Raschka, S. (2018). *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. http://arxiv.org/abs/1811.12808

Rodríguez, P., Bautista, M. A., Gonzàlez, J., & Escalera, S. (2018). Beyond one-hot encoding: Lower dimensional target embedding. *Image and Vision Computing*, *75*, 21–31. https://doi.org/10.1016/j.imavis.2018.04.004

Shen, K., Guo, J., Tan, X., Tang, S., Wang, R., & Bian, J. (2023). *A Study on ReLU and Softmax in Transformer*. http://arxiv.org/abs/2302.06461

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524. https://doi.org/https://doi.org/10.1016/j.asoc.2019.105524

Sutjiatmi, S., Akta Padma Eldo, D. H., & Zainudin, A. (2020). Public Perception Regarding Money Politics in General Election 2019 (Compartive Study on Tegal City and Tegal Regency). *CosmoGov*, *6*(1), 61. https://doi.org/10.24198/cosmogov.v6i1.26632

Tae, K. H., Roh, Y., Oh, Y. H., Kim, H., & Whang, S. E. (2019). Data cleaning for accurate, fair, and robust models: A big data - AI integration approach. *Proceedings of the ACM SIGMOD International Conference on Management of Data*. https://doi.org/10.1145/3329486.3329493

Tahyudin, I., Hananto, A. R., Rahayu, S. A., Anjani, R. M., & Nurhopipah, A. (2023). Sentiment Analysis Model Development on E-Money Service Complaints. *TEM Journal*, *12*(4), 2050–2055. https://doi.org/10.18421/TEM124-15

Tsai, M.-H., Wang, Y., Kwak, M., & Rigole, N. (2019). A Machine Learning Based Strategy for Election Result Prediction. *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 1408–1410. https://doi.org/10.1109/CSCI49370.2019.00263

Valdivia, A., Luzón, M. V., Cambria, E., & Herrera, F. (2018). Consensus vote models for detecting and filtering neutrality in sentiment analysis. *Information Fusion*, *44*, 126–135. https://doi.org/https://doi.org/10.1016/j.inffus.2018.03.007

Vanslette, K., Tohme, T., & Youcef-Toumi, K. (2020). A general model validation and testing tool. *Reliability Engineering & System Safety*, *195*, 106684. https://doi.org/https://doi.org/10.1016/j.ress.2019.106684

Wardoyo, R., Musdholifah, A., Pradipta, G. A., & Sanjaya, I. N. H. (2020). Weighted Majority Voting by Statistical Performance Analysis on Ensemble Multiclassifier. *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 1–8. https://doi.org/10.1109/ICIC50835.2020.9288552

Wongkar, M., & Angdresey, A. (2019). Sentiment Analysis Using Naive Bayes Algorithm Of The Data Crawler: Twitter. *2019 Fourth International Conference on Informatics and Computing (ICIC)*, 1–5. https://doi.org/10.1109/ICIC47613.2019.8985884

Yacouby, R., & Axman, D. (2020). *Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models*. 79–91. https://doi.org/10.18653/v1/2020.eval4nlp-1.9

Yu, L., Hu, Y., Xie, X., Lin, Y., & Hong, W. (2020). Complex-Valued Full Convolutional Neural Network for SAR Target

Classification. *IEEE Geoscience and Remote Sensing Letters*, *17*(10), 1752–1756. https://doi.org/10.1109/LGRS.2019.2953892