Tuskegee University

College of Business and Information Science

Department of Computer Science

ISCS.0536—Data Mining & Machine Learning

Instructor: Dr. Chang Xiao

Project Report

#Submitted by:

Sameeruddin Mohammed

Abstract

This project aims to classify U.S. congresspersons as Democrats or Republicans based on their voting records on key issues. Using the U.S. Congressional Voting Records dataset, which contains 435 instances and 16 voting features, machine learning models were trained to predict party affiliation. Three classifiers—Logistic Regression**,** Naive Bayes, and Decision Tree—were applied to the dataset.

The data was preprocessed to handle missing values and encode categorical attributes into numerical form. Models were evaluated using metrics such as accuracy, precision, recall, and F1-score. Logistic Regression emerged as the best-performing model with an accuracy of 96.18%**,** followed by the Decision Tree **(**94.66%**)** and Naive Bayes **(**92.37%**).** Cross-validation further validated these results, confirming Logistic Regression's reliability.

The project demonstrates the effectiveness of machine learning in analyzing voting behavior and predicting political affiliations, providing a foundation for further exploration of political data analysis

Introduction

The purpose of this project is to predict the political party affiliation of U.S. congresspersons (Democrat or Republican) based on their voting records on key legislative issues. Political affiliation often correlates with distinct voting patterns, and understanding these patterns can provide valuable insights into policy alignments, ideological trends, and future voting behaviors.

The dataset used for this project is the U.S. Congressional Voting Records dataset, which consists of voting behavior for 435 congresspersons on 16 key issues, such as education spending, military funding, and healthcare. Each congressperson's votes are recorded as yes, no, or missing (?), and the target variable (Class) represents their political affiliation: 0 for Democrat and 1 for Republican.

This dataset has real-world significance as it reflects the decision-making processes of elected representatives. Predicting political affiliation based on voting patterns can aid in understanding how party ideologies influence policy decisions and can also serve as a foundation for further political analysis and forecasting.

Dataset Overview

The dataset used in this project is the U.S. Congressional Voting Records dataset, which provides voting data for 435 U.S. congresspersons on 16 key legislative issues. Each row in the dataset represents a congressperson, and the columns represent their votes (yes, no, or missing). The primary goal is to predict the political party affiliation of each congressperson based on these voting patterns.

Key Characteristics of the Dataset:

1. Number of Instances (Rows)**:** 435 (each representing one congressperson).
2. Number of Attributes (Columns)**:** 16 features (voting records) and 1 target variable (Class).
3. Target Variable (**Class**)**:**
   - Democrat
   - Republican
4. Features**:**
   - Each feature corresponds to a vote on a specific issue (e.g., handicapped-infants, water-project-cost-sharing, education-spending).
   - Votes are categorical and represented as:
     - **y**: Yes
     - n: No
     - **?** Missing
5. Challenges**:**
   - Missing Values: Some voting records are incomplete and represented as, which were handled during data preprocessing.
   - Categorical Data: Votes are categorical and required conversion to numerical values for compatibility with machine learning models.

Relevance of the Dataset:

This dataset is particularly relevant as it reflects real-world voting behavior in the U.S. Congress. By analyzing voting patterns, this project not only predicts political affiliation but also provides insights into how party ideologies align with specific policy decisions. The dataset's focus on legislative votes makes it an excellent choice for exploring the application of machine learning in political data analysis.

Data Preprocessing

The data preprocessing stage was crucial for preparing the dataset for machine learning models. This involved handling missing values, transforming categorical data into numerical format, and splitting the data into training and testing sets. Below are the key preprocessing steps:

1. Handling Missing Data

- The dataset contained missing values represented as? in the voting records.
- Missing values were replaced with the **mode** (most frequent value) of each column. This ensured:
    - Retention of all instances in the dataset.
    - Consistency in the representation of voting behavior.

2. Encoding Categorical Data

- Voting records were categorical (`y` for Yes, `n` for No, and? for missing) and needed to be converted into numerical values for machine learning models.
- **Label Encoding** was used to transform categorical values:
    - `y` → `1`
    - `n` → `0`
- The target variable (`Class`) was also encoded:
    - `0` for Democrat
    - `1` for Republican

3. Train-Test Split

- The dataset was split into two parts:
    - Training Set: 70% of the data, used to train the models.
    - Testing Set: 30% of the data, used to evaluate model performance.
- This split ensures the models are tested on unseen data, providing a reliable estimate of their generalization capabilities.

Why These Steps?

- Replacing Missing Values: Preserved the dataset's size and avoided information loss while maintaining consistency.
- Encoding Categorical Data: Allowed machine learning algorithms to process the data numerically.
- Train-Test Split: Ensured that the models were evaluated on separate data, reducing overfitting and providing a fair assessment.

Logistic Regression

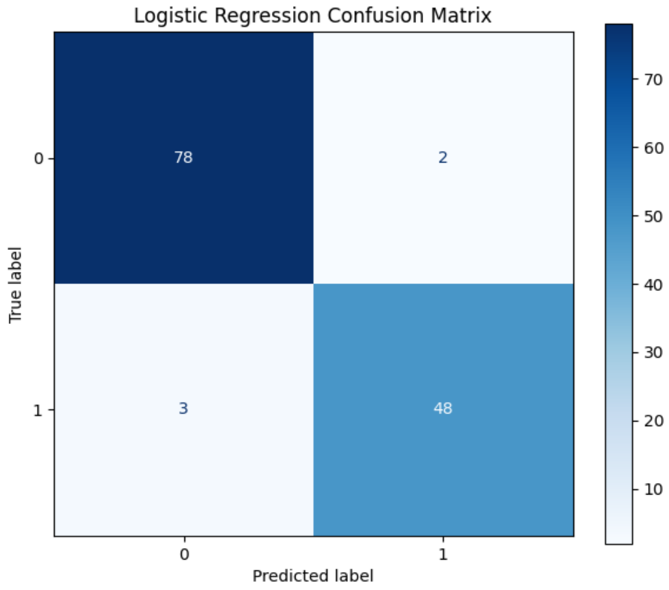
- Logistic Regression is a linear model used for binary classification tasks. It predicts the probability that a given instance belongs to one of the two classes (Democrat or Republican in this case) based on a weighted combination of the input features.

Why Logistic Regression?

- It is simple, efficient, and interpretable, making it a great baseline for binary classification.
- It assumes a linear relationship between the input features and the log-odds of the target variable, which often works well for structured datasets like this one.

Performance Summary:

- **Accuracy**: 96.18%
- **Precision**:
  - Democrat (Class 0): 96%
  - Republican (Class 1): 96%
- **Recall**:
  - Democrat (Class 0): 97%
  - Republican (Class 1): 94%
- **Strengths**:
  - High accuracy and balanced precision/recall for both classes.
  - Easy to implement and interpret.



True Positives (TP) = 48 correct predictions for Republicans

True Negatives (TN) = 78 correct predictions for Democrats

False Positives (FP) = 2 Democrats incorrectly predicted as Republicans

False Negatives (FN) = 3 Republicans incorrectly predicted as Democrats

Naive Bayes

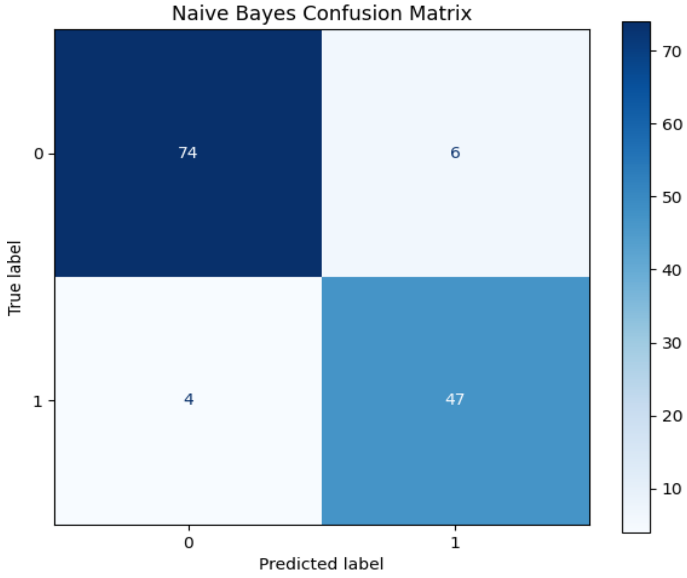
Naive Bayes is a probabilistic classifier based on Bayes' Theorem, with the assumption that features are independent given the class label. Despite the "naive" independence assumption, it often performs well on categorical datasets like this one.

Why Naive Bayes?

- It is computationally efficient and works well with high-dimensional or categorical data.
- It is particularly suitable for problems where the relationships between features are less complex or where quick results are needed.

Performance Summary:

- **Accuracy**: 92.37%
- **Precision**:
  - Democrat (Class 0): 95%
  - Republican (Class 1): 89%
- **Recall**:
  - Democrat (Class 0): 93%
  - Republican (Class 1): 92%
- **Strengths**:
  - Fast and efficient for categorical data.
  - Performs well even with small amounts of data.



True Positives (TP): 47 Correct predictions for Republicans

True Negatives (TN): 74 Correct predictions for Democrats

False Positives (FP): 6 Democrats incorrectly predicted as Republicans

False Negatives (FN): 4 Republicans

Decision Tree

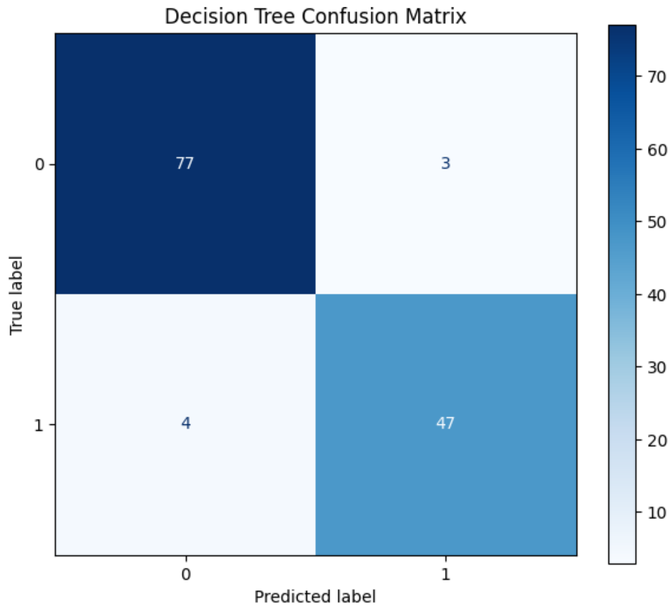
A Decision Tree is a non-linear model that splits data into subsets based on feature values, forming a tree-like structure. Each internal node represents a decision (based on a feature), and each leaf node represents a class label (Democrat or Republican).

Why Decision Tree?

- It handles both numerical and categorical data effectively, making it suitable for this dataset with categorical voting patterns.
- It is highly interpretable, as the tree structure allows us to visualize the decision-making process.

Performance Summary:

- **Accuracy**: 94.66%
- **Precision**:
  - Democrat (Class 0): 95%
  - Republican (Class 1): 94%
- **Recall**:
  - Democrat (Class 0): 96%
  - Republican (Class 1): 92%
- **Strengths**:
  - Can model complex, non-linear relationships.
  - Provides insights into feature importance, revealing which voting issues were most significant in determining political affiliation.
  - Requires minimal data preprocessing.



True Positives (TP): 47 Correct predictions for Republicans

True Negatives (TN): 77 Correct predictions for Democrats

False Positives (FP): 3 Democrats incorrectly predicted as Republicans

False Negatives (FN): 4 Republicans incorrectly predicted as Democrats

Evaluation: Cross-Validation

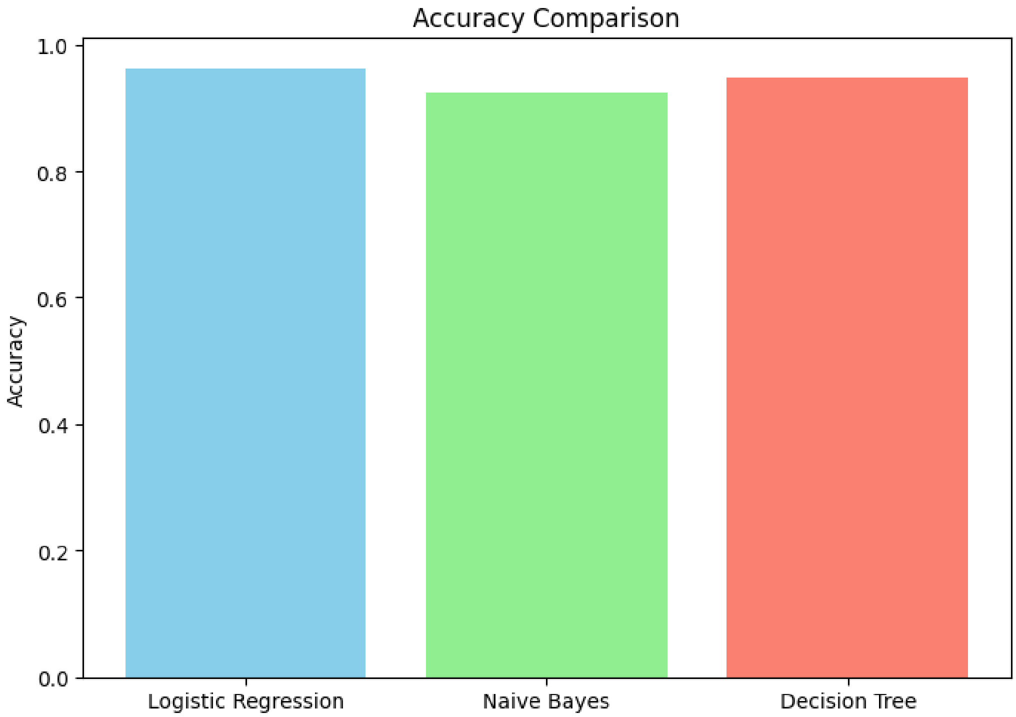
Purpose of Cross-Validation:

Cross-validation is an evaluation technique used to assess how well a model generalizes to unseen data. In this project, **5-**fold cross-validation was performed for all three classifiers. The dataset was divided into five equal parts, with the model trained on four parts and tested on the remaining part. This process was repeated five times, and the average performance was calculated.

Cross-Validation Results:

- **Logistic Regression**:
  - Accuracy scores across 5 folds**:** [97.70%, 95.40%, 90.80%, 100%, 94.30%]
  - Mean Accuracy**: 95.63%**
  - Logistic Regression demonstrated consistent and high accuracy, confirming its reliability for this dataset.
- **Naive Bayes**:
  - Mean Accuracy**: 90.11%**
  - Naive Bayes performed efficiently but was less accurate than Logistic Regression and Decision Tree, particularly due to its independence assumption.
- **Decision Tree**:
  - Mean Accuracy**: 93.79%**
  - Decision Tree showed strong performance and good generalization, slightly outperforming Naive Bayes but falling short of Logistic Regression.

Comparison



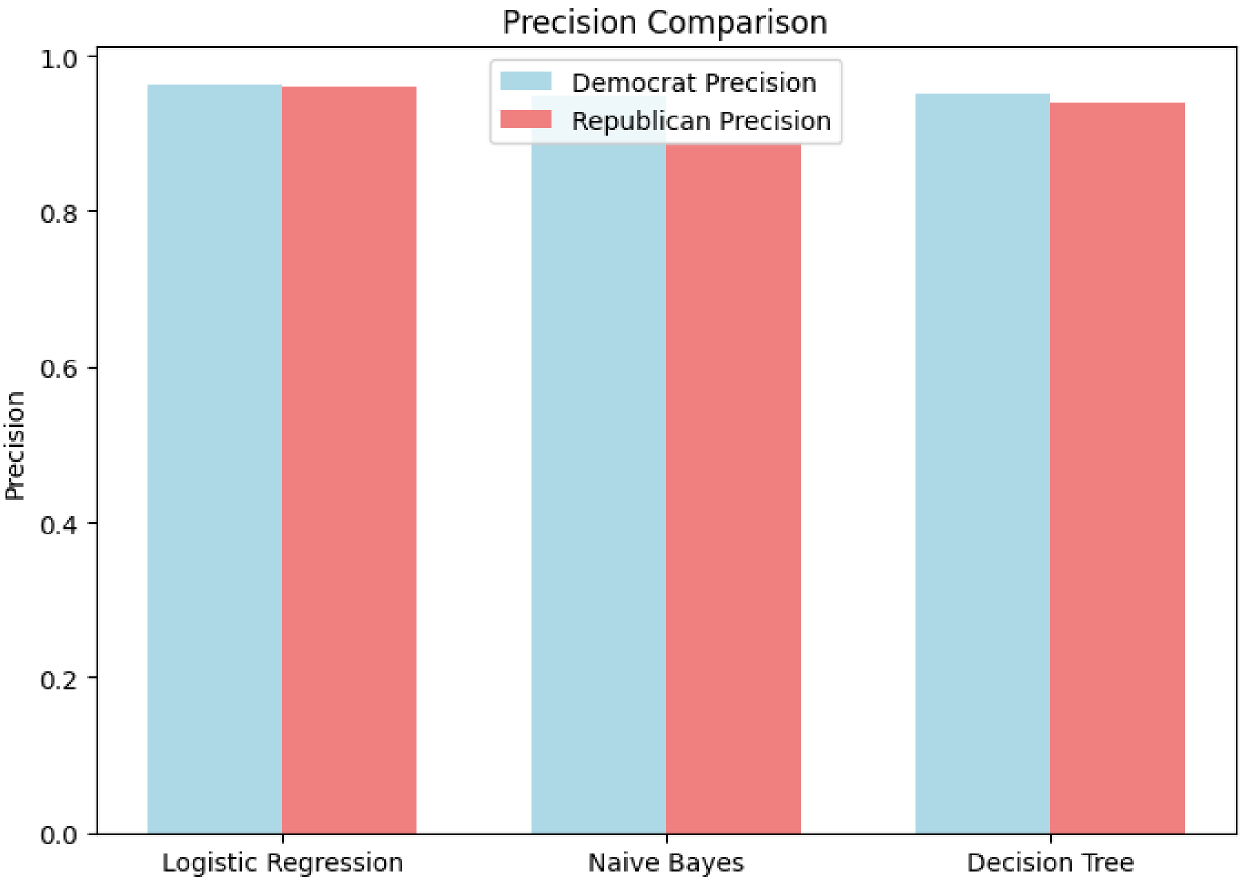
Accuracy

Logistic Regression: **96.18%**

Naive Bayes: **92.37%**

Decision Tree: **94.66%**

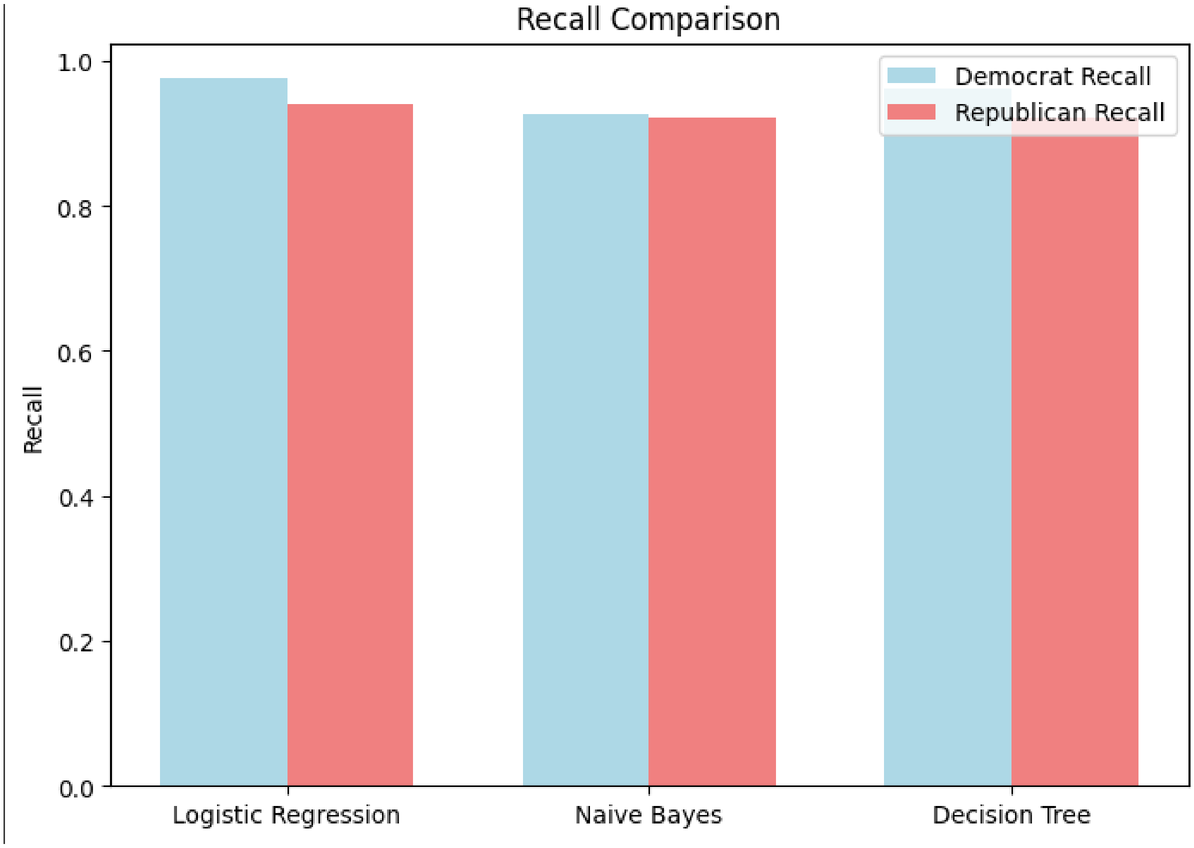
Comparison



Precision

Logistic Regression:
Class 0: **96%**, Class 1: **96%**
Naive Bayes:
Class 0: **95%,** Class 1: **89%**
Decision Tree:
Class 0: **95%,** Class 1: **94%**

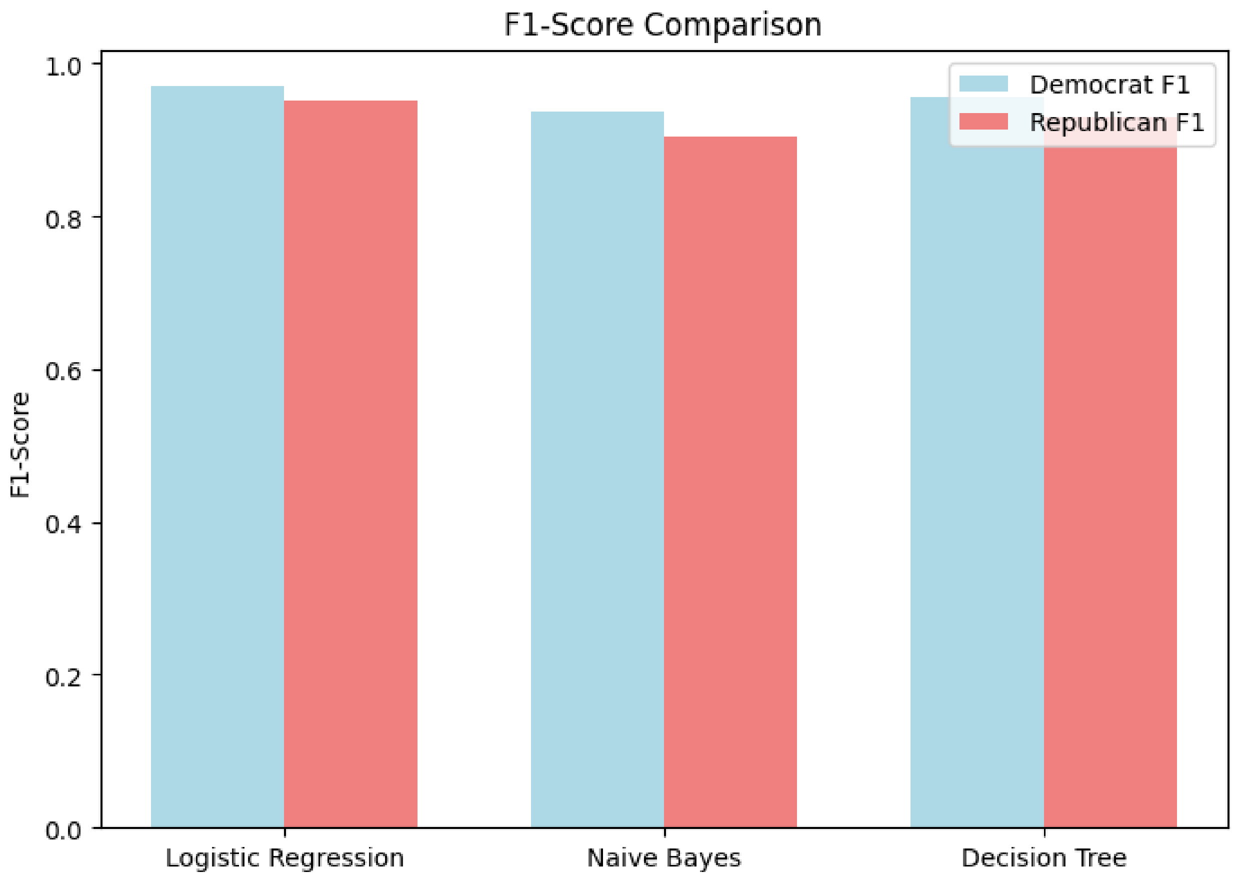
Comparison



Recall

Logistic Regression:
Class 0: **97%,** Class 1: **94%**
Naive Bayes:
Class 0: **93%,** Class 1: **92%**
Decision Tree:
Class 0: **96%,** Class 1: **92%**

Comparison



F1-Score

Logistic Regression:
Class 0: **97%**, Class 1: **95%**
Naive Bayes:
Class 0: **94%,** Class 1: **90%**
Decision Tree:
Class 0: **96%,** Class 1: **93%**

Conclusions

Best Classifier: Logistic Regression

Logistic Regression outperformed Naive Bayes and Decision Tree in terms of accuracy (**96.18%**) and achieved consistently high precision, recall, and F1-scores for both Democrats and Republicans.

Naive Bayes:

> While simpler and faster, Naive Bayes had slightly lower accuracy (**92.37%**) and struggled with precision for Republicans.

Decision Tree:

> Decision Tree performed well with an accuracy of **94.66%** and offered valuable insights into feature importance. However, it was slightly less accurate than Logistic Regression.

References:

https://www.google.com/search?q=naive+bayes+&sca_esv=0baf5c9e671f57b4&rlz=1C1VDKB_enUS1126US1126&sxsrf=ADLYWIKMVMawo7bzFqQP7p7rDQWrIQJfhg%3A1733773997414&ei=rUpXZ9r8GIfcwN4Pl77a2QI&ved=0ahUKEwiah_K5u5uKAxUHLtAFHRefNisQ4dUDCA8&uact=5&oq=naive+bayes+&gs_lp=Egxnd3Mtd2l6LXNlcnAiDG5haXZlIGJheWVzIDIKECMYgAQYJxiKBTINEAAYgAQYsQMYQxiKBTIKEAAYgAQYQxiKBTIIEAAYgAQYsQMyChAAGIAEGBQYhwIyChAAGIAEGEMYigUyChAAGIAEGEMYigUyBRAAGIAEMgUQABiABDIKEAAYgAQYQxiKBUicCFD6Alj6AnABeAGQAQCYAWSgAWSqAQMwLjG4AQPIAQD4AQGYAgKgAm_CAgoQABiwAxjWBBhHwgINEAAYgAQYsAMYQxiKBZgDAIgGAZAGCpIHAzEuMaAHiwc&sclient=gws-wiz-serp

https://www.ibm.com/topics/naive-bayes#:~:text=Na%C3%AFve%20Bayes%20is%20part%20of,important%20to%20differentiate%20between%20classes.

https://www.geeksforgeeks.org/decision-tree/