MACHINE LEARNING PROJECT

Overview

This dataset includes 3,616 reviews about airline experiences, with the following main columns:

- Rating and Review Details: Includes an overall rating (rating), a brief review headline (header), and detailed content (content).
- Reviewer Information: Data about the author (author), review date (date), and location (place).
- Flight Details: Includes aircraft type (aircraft), traveler type (Traveler type), seat type (seat_type), flight route (route), and date flown (date_flown).
- Service Ratings: Ratings for seat comfort, cabin staff service, food and beverages, ground service, entertainment, and value for money.
- Other Information: Whether the trip was verified (trip_verified) and if the reviewer recommends the airline (recommended).

1. **K-Nearest Neighbors (KNN):**
   - **How it works**: KNN is a simple, instance-based learning algorithm. It classifies a new data point based on the majority label of its nearest neighbors in the feature space (using a distance metric like Euclidean).
   - **Key characteristics**: Non-parametric, sensitive to feature scaling, and computationally expensive for large datasets due to distance calculations.
2. **Naive Bayes:**
   - **How it works**: Based on Bayes' Theorem, it calculates the probability of a class given the feature values, assuming features are independent (hence "naive"). It selects the class with the highest posterior probability.
   - **Key characteristics**: Fast and efficient for text classification and categorical data but performs poorly when features are highly correlated.
3. **Support Vector Machine (SVM):**
   - **How it works**: SVM finds the hyperplane that best separates classes in the feature space by maximizing the margin (distance) between data

points of different classes. It can handle non-linear data using kernel functions.

- **Key characteristics**: Effective for high-dimensional data and works well for small datasets but can be slow and memoryintensive for large datasets.

4. **Decision Tree:**
   - **How it works**: A tree-like model splits the dataset into subsets based on feature values to predict the target variable. It chooses splits that maximize information gain or minimize impurity.
   - **Key characteristics**: Easy to interpret, handles mixed feature types well, but prone to overfitting unless regularized (e.g., using pruning).

Algorithm	Type	Strengths	Weaknesses	Best Used For
KNN	Instancebased	Simple, intuitive, no training phase required	Sensitive to scaling, computational cost	Small datasets, nonlinear relationships
Naive Bayes	Probabilistic	Fast, handles text data well, scalable	Assumes independence of features	Text classification, spam detection
SVM	Discriminative	Effective for high-dimensional spaces	Computationally expensive, requires tuning	Complex, high-dimensional datasets
Decision Tree	Rule-based	Easy to interpret, handles mixed data types	Overfitting, biased with imbalanced data	Simple models, datasets with clear splits

Process

1. Dataset:

This code preprocesses a dataset for machine learning tasks, focusing on classification and regression. Here's a summary:

Dataset Loading and Cleaning: Unnecessary columns are removed, and specific target variables are mapped to binary values for classification.

Feature Separation: Features are divided into numerical and categorical columns for preprocessing. The trip_verified column is kept as a separate feature.

Pipeline Setup: Preprocessing pipelines are defined:

Numerical data is imputed using the mean and scaled.

Categorical data is imputed with the most frequent value and one-hot encoded.

Feature Transformation: All features are processed using the pipelines, and the trip_verified column is appended to the transformed features.

Target Reconstruction: Two separate datasets are created: one for classification (with the recommended column) and one for regression (with the rating column).

Data Balancing: The classification dataset is balanced by downsampling the majority class to match the size of the minority class.

Dataset Splitting: Both the balanced classification dataset and the regression dataset are split into training, validation, and test sets using a 70-15-15 split.

Saving Outputs: The processed datasets, including the balanced and split datasets for classification and regression, are saved for further use.

This process ensures that the data is cleaned, preprocessed, balanced, and ready for machine learning modeling.

<u>KNN Model:</u>

Observations:

Euclidean Metric:

- Performance improves as kkk increases.
- Highest **F1-Score** of **0.94** is achieved at k=10k=10k=10, with strong Precision (0.92) and Recall (0.96).
- Smaller kkk values (e.g., k=1k=1k=1) show slightly lower overall scores, possibly due to higher sensitivity to noise in the data.

Manhattan Metric:

- Similar trend of improving performance as kkk increases.
- Highest **F1-Score** of **0.94** also occurs at k=10k=10k=10, matching the Euclidean metric, with Precision (0.92) and Recall (0.95).

Best Model:

The **Euclidean metric with k=10k=10k=10** is identified as the bestperforming model because it achieves the highest F1-Score (0.940.940.94), which balances Precision (0.92) and Recall (0.96). Although the Manhattan metric also achieves an F1-Score of 0.94 at k=10k=10k=10, the choice may lean towards Euclidean due to its slightly better Recall.

Precision:

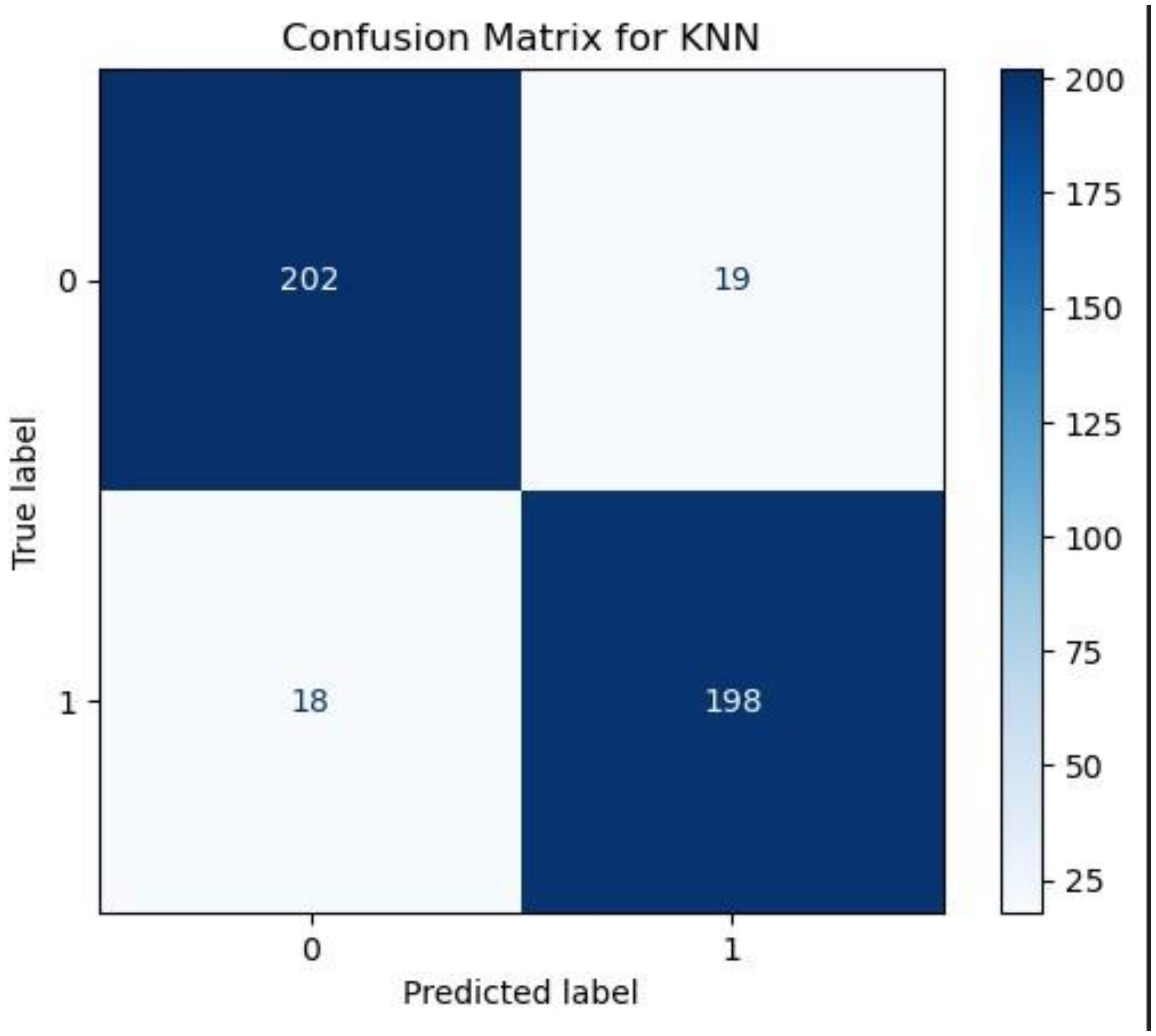
- The proportion of correctly predicted positive instances out of all instances predicted as positive.
- Class 0: Precision=202202+19≈0.92\text{Precision} = \frac{202}{202+19} \approx 0.92Precision=202+19202≈0.92
- Class 1: Precision=198198+18≈0.91\text{Precision} = \frac{198}{198+18} \approx 0.91Precision=198+18198≈0.91 **Recall**:
- The proportion of correctly predicted positive instances out of all actual positive instances.
- Class 0: Recall=202202+18≈0.91\text{Recall} = \frac{202}{202+18} \approx 0.91Recall=202+18202≈0.91
- Class 1: Recall=198198+19≈0.92\text{Recall} = \frac{198}{198+19} \approx 0.92Recall=198+19198≈0.92 **F1-Score**:
- The harmonic mean of precision and recall, balancing both metrics.
- Class 0: F1-Score=2×Precision×RecallPrecision+Recall≈0.92\text{F1Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \approx 0.92F1Score=2×Precision+RecallPrecision×Recall≈0.92
- Class 1: F1-Score=2×Precision×RecallPrecision+Recall≈0.91\text{F1Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \approx 0.91F1Score=2×Precision+RecallPrecision×Recall≈0.91

Support:

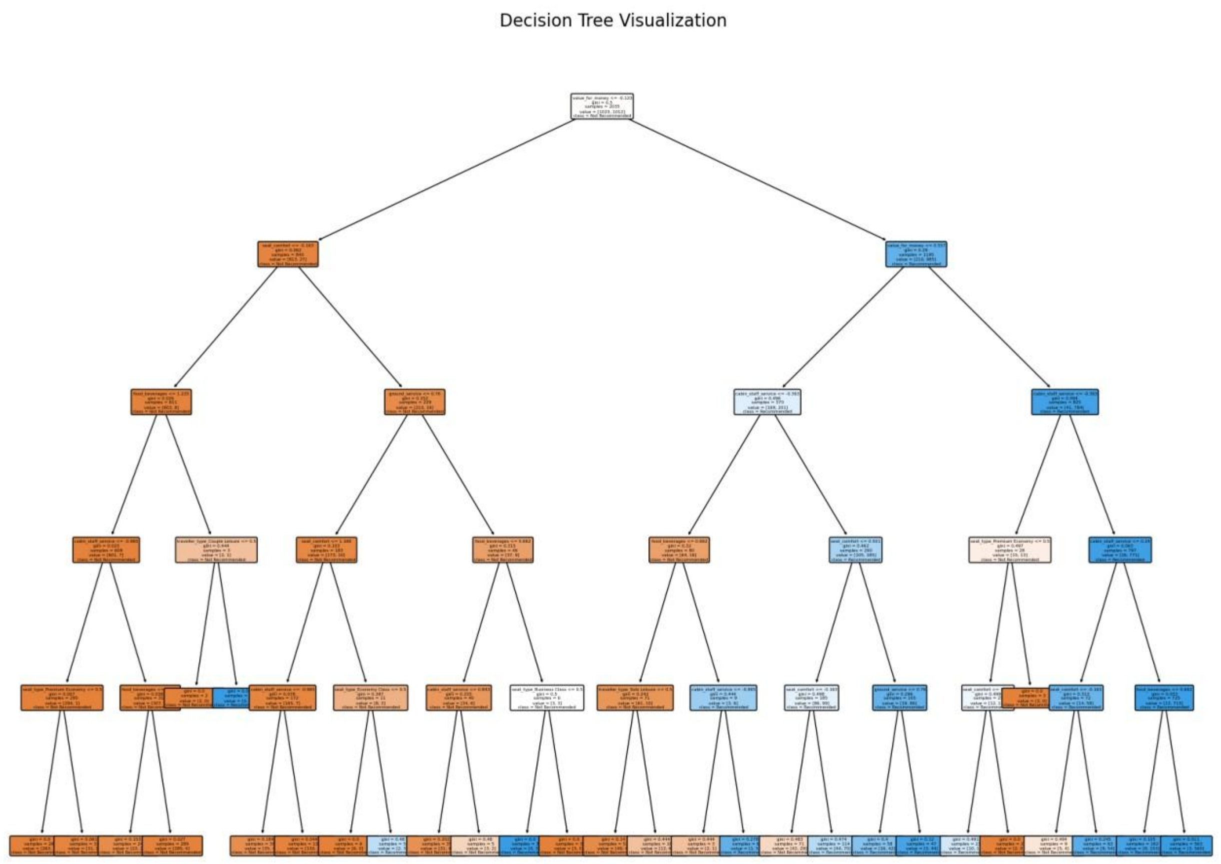
- The number of actual instances for each class in the test set.

- Class 0: 221
- Class 1: 216 **Overall Metrics**:
- **Accuracy**: The proportion of correctly predicted instances out of all instances. Accuracy=202+198437≈0.92\text{Accuracy} = \frac{202+198}{437} \approx 0.92Accuracy=437202+198≈0.92
- **Macro Average**: Unweighted average of Precision, Recall, and F1-Score across classes.

Weighted Average: Average of Precision, Recall, and F1-Score, weighted by the number of instances in each class.



Decision Tree:



Precision:

- The proportion of correctly predicted instances for a class out of all instances predicted as that class.
- **Class 0**: Precision=203203+18≈0.91\text{Precision} = \frac{203}{203+18} \approx 0.91Precision=203+18203≈0.91
- **Class 1**: Precision=195195+21≈0.92\text{Precision} = \frac{195}{195+21} \approx 0.92Precision=195+21195≈0.92 **Recall**:
- The proportion of correctly predicted instances for a class out of all actual instances of that class.
- **Class 0**: Recall=203203+18≈0.92\text{Recall} = \frac{203}{203+18} \approx 0.92Recall=203+18203≈0.92
- **Class 1**: Recall=195195+21≈0.90\text{Recall} = \frac{195}{195+21} \approx 0.90Recall=195+21195≈0.90 **F1-Score**:
- The harmonic mean of precision and recall, balancing both metrics.

- **Class 0**: F1-Score=2×Precision×RecallPrecision+Recall≈0.91\text{F1Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \approx 0.91F1Score=2×Precision+RecallPrecision×Recall≈0.91
- **Class 1**: F1-Score=2×Precision×RecallPrecision+Recall≈0.91\text{F1Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \approx

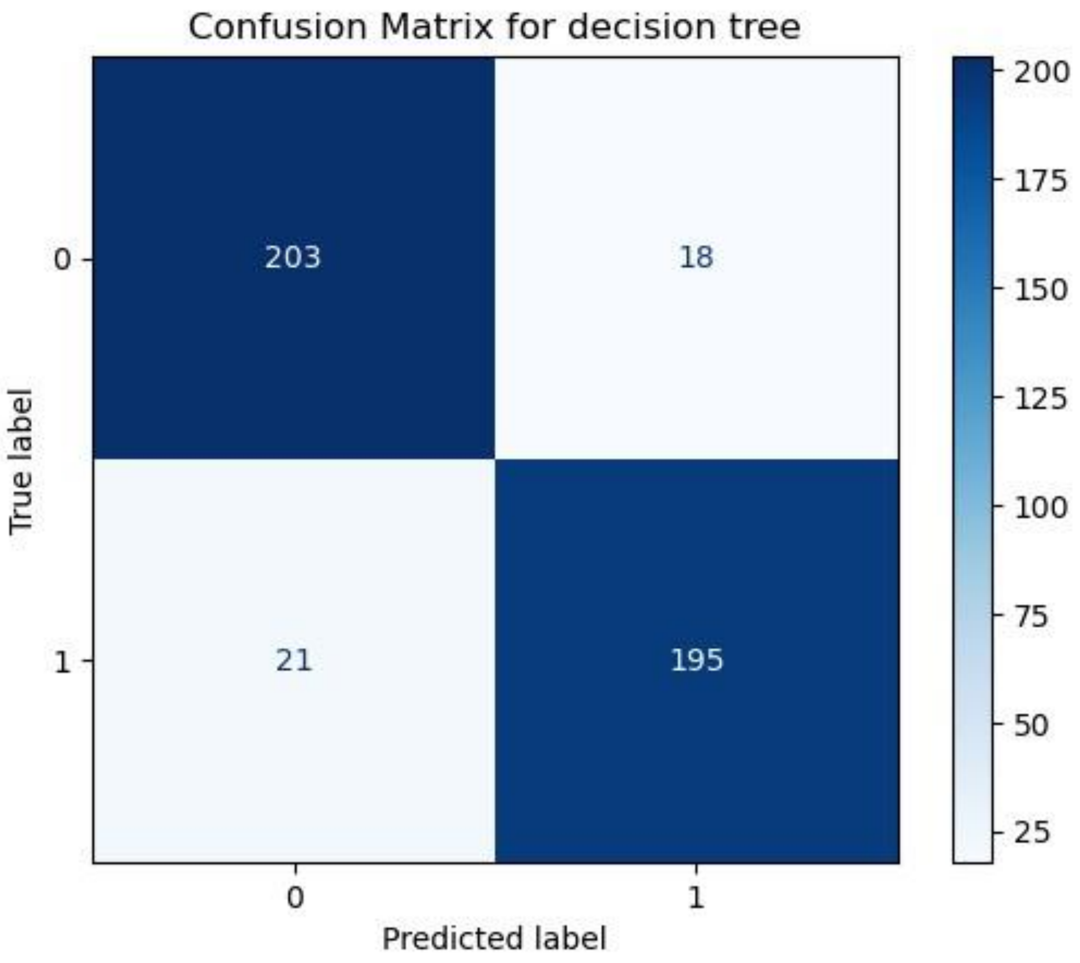
0.91F1Score=2×Precision+RecallPrecision×Recall≈0.91 **Support**:

- The number of actual instances for each class in the test set:
  - Class 0: 221 ○

  Class 1: 216 **Overall**

Metrics:

- **Accuracy**: The proportion of correctly predicted instances out of all instances: Accuracy=203+195437≈0.91\text{Accuracy} = \frac{203+195}{437} \approx 0.91Accuracy=437203+195≈0.91
- **Macro Average**: Unweighted average of Precision, Recall, and F1-Score across classes.
- **Weighted Average**: Average of Precision, Recall, and F1-Score, weighted by the number of instances in each class.



Confusion Matrix for decision tree
True label
Predicted label
0
1
0
1
203
18
21
195
200
175
150
125
100
75
50
25

Naive Bayes

Metrics for Each Class:

1. **Class 0**:
   - **Precision**: Precision=True PositivesTrue Positives+False Positives\text{Preci sion} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}Precision=True Positives+False PositivesTrue Positives
     - Out of all instances predicted as Class 0, 91% were actually Class 0.
   - **Recall**: Recall=True PositivesTrue Positives+False Negatives\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}Recall=True Positives+False NegativesTrue Positives
     - Out of all actual instances of Class 0, 89% were correctly predicted as Class 0. ○ **F1-Score**: The harmonic mean of precision and recall.
     - F1-Score for Class 0 is 0.90, indicating balanced performance.
2. **Class 1**:
   - **Precision**: 89% of the instances predicted as Class 1 were correctly classified.
   - **Recall**: 91% of all actual Class 1 instances were correctly predicted as Class 1.
   - **F1-Score**: The F1-Score for Class 1 is also 0.90.

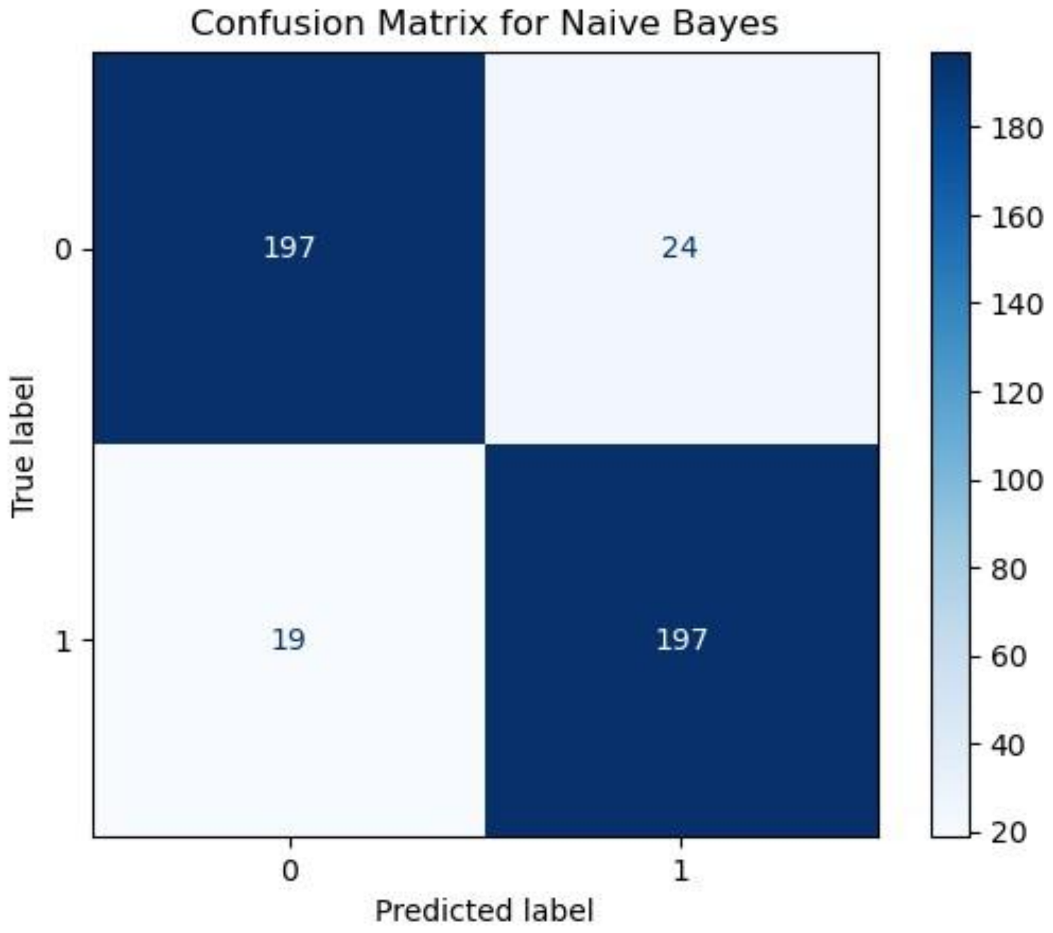
Overall Metrics:

1. **Accuracy**:
   - Accuracy=Correct PredictionsTotal Instances\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Instances}}Accuracy=Total InstancesCorrect Predictions ○ The

   model correctly predicted 90% of all instances in the test set.
2. **Macro Average**:

- The unweighted average of Precision, Recall, and F1-Score across the two classes.
- Macro Avg Precision, Recall, F1-Score=0.90\text{Macro Avg Precision, Recall, F1-Score} = 0.90Macro Avg Precision, Recall, F1-Score=0.90

3. **Weighted Average**:
   - The weighted average of Precision, Recall, and F1-Score, considering the number of instances in each class (support). ○ Also equal to 0.90, indicating consistent performance across both classes.



SVM

Metrics for Each Class:

1. **Class 0**:
   - **Precision**:
     - Precision=True PositivesTrue Positives+False Positives\text

{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}Precision=True Positives+False PositivesTrue Pos itives

 Out of all instances predicted as Class 0, 94% were correct.

- **Recall**:

   Recall=True PositivesTrue Positives+False Negatives\text{R ecall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}Recall=True Positives+False NegativesTrue Posit ives

   Out of all actual instances of Class 0, 91% were correctly identified. o **F1-Score**:

   The harmonic mean of precision and recall, balancing the two metrics.

   F1-Score for Class 0 is 0.92, indicating strong and balanced performance.

2. **Class 1**:

   - **Precision**:

      Out of all instances predicted as Class 1, 92% were correct.

   - **Recall**:

      94% of all actual instances of Class 1 were correctly predicted.

   - **F1-Score**:

      The F1-Score for Class 1 is 0.93, slightly higher than Class 0.

Overall Metrics:

1. **Accuracy**:

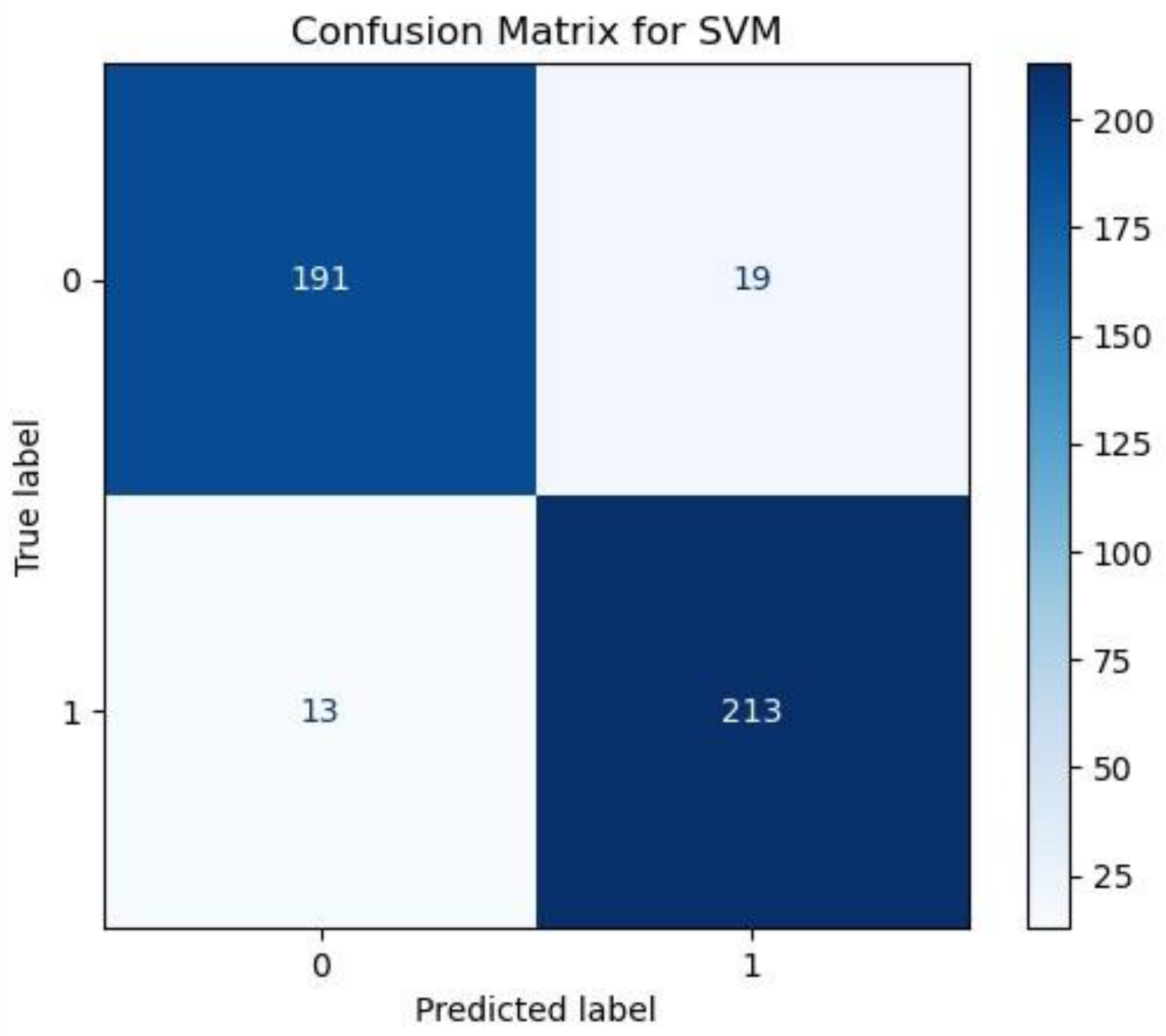
   - Accuracy=Correct PredictionsTotal Instances\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Instances}}Accuracy=Total InstancesCorrect Predictions

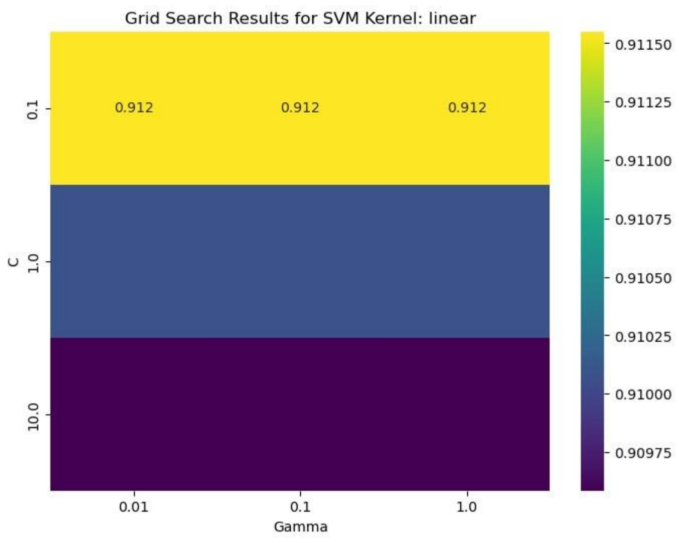
   - The model correctly classified 93% of all instances in the validation set.

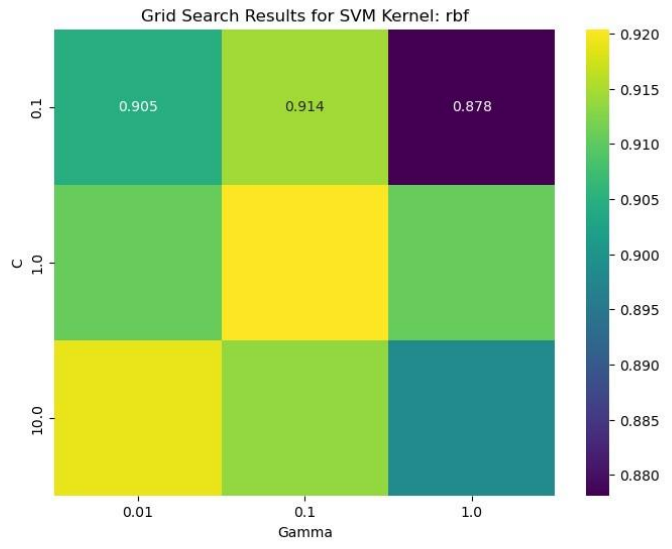
2. **Macro Average**:

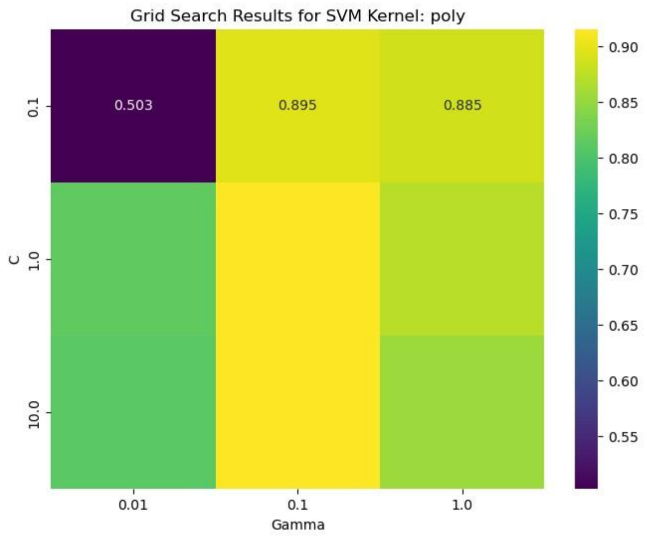
- The unweighted average of Precision, Recall, and F1-Score across the two classes:
  - Macro Avg Precision, Recall, F1-Score=0.93\text{Macro Avg Precision, Recall, F1-Score} = 0.93Macro Avg Precision, Recall, F1-Score=0.93
  - This shows that the model performs similarly for both classes without bias toward one.

3. **Weighted Average**:
   - The weighted average of Precision, Recall, and F1-Score, considering the number of instances (support) in each class:
     - Weighted Avg Precision, Recall, F1-Score=0.93\text{Weighted Avg Precision, Recall, F1-Score} = 0.93Weighted Avg Precision, Recall, F1-Score=0.93
     - This confirms balanced performance given the slight difference in class distributions (Class 0 has 210 instances; Class 1 has 226).


Confusion Matrix for SVM
191
19
13
213
True label
Predicted label
0
1
0
1
200
175
150
125
100
75
50
25







Metrics for Each Class:

1. **Class 0**:

- **Precision**:
  - Precision=True PositivesTrue Positives+False Positives\text {Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}Precision=True Positives+False PositivesTrue Pos itives
  - Out of all instances predicted as Class 0, 94% were correct. o

    Recall:
  - Recall=True PositivesTrue Positives+False Negatives\text{R ecall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}Recall=True Positives+False NegativesTrue Posit ives
  - Out of all actual instances of Class 0, 91% were correctly identified. o **F1-Score**:
  - The harmonic mean of precision and recall, balancing the two metrics.
  - F1-Score for Class 0 is 0.92, indicating strong and balanced performance.

2. **Class 1**:
   - **Precision**:
     - Out of all instances predicted as Class 1, 92% were correct.
   - **Recall**:
     - 94% of all actual instances of Class 1 were correctly predicted.
   - **F1-Score**:
     - The F1-Score for Class 1 is 0.93, slightly higher than Class 0.

Overall Metrics:

1. **Accuracy**:
   - Accuracy=Correct PredictionsTotal Instances\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Instances}}Accuracy=Total InstancesCorrect Predictions
   - The model correctly classified 93% of all instances in the validation set.

2. **Macro Average**:

    - The unweighted average of Precision, Recall, and F1-Score across the two classes:

        - Macro Avg Precision, Recall, F1-Score=0.93\text{Macro Avg Precision, Recall, F1-Score} = 0.93Macro Avg Precision, Recall, F1-Score=0.93

        - This shows that the model performs similarly for both classes without bias toward one.

3. **Weighted Average**:

    - The weighted average of Precision, Recall, and F1-Score, considering the number of instances (support) in each class:

        - Weighted Avg Precision, Recall, F1-Score=0.93\text{Weighted Avg Precision, Recall, F1-Score} = 0.93Weighted Avg Precision, Recall, F1-Score=0.93

        - This confirms balanced performance given the slight difference in class distributions (Class 0 has 210 instances; Class 1 has 226).