



Machine Learning and Data Science - ENCS5341

Assignment #3

Prepared By: Ali Halayqa - 1201769

Section : 1

Instructor : Dr.Yazan Abu Farha

Mohammad shrateh – 1201369

Section : 3

Instructor : Ismail khater

Date: 26/12/2024

1. Introduction

In this report, we will explore different machine learning techniques. Such as K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machines (SVM), and ensemble techniques such as Boosting and Bagging. The goal of this assignment is to experiment with these methods, compare their performance using classification metrics, such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC.

1.1. K-Nearest Neighbors (KNN)

This algorithm is classified as non-parametric, i.e. it depends on cases and is used in regression and classification. It classifies data points based on the majority class of their nearest neighbors in the feature space, based on several metrics including Euclidean or Manhattan. KNN is considered simple, intuitive and effective for smaller data sets with clear class boundaries. However, it can be computationally expensive, as prediction requires comparing the point to all the training data. It is sensitive to noisy data, irrelevant features, and the choice of k , which affects performance. KNN performs well on low-dimensional data but struggles with the curse of dimensionality.

1.2. Logistic Regression

It is a statistical model used for binary classification, It estimates the probability of inputs belonging to a certain class using a logistic function (**sigmoid**). It creates linear boundaries by modeling the logarithmic probabilities of the outputs as a linear function of the characteristics of the inputs. Logistic regression uses logit loss to optimize the weights and applies L1/L2 regularization to prevent overfitting. It is effective for linearly separable data, but has difficulties with nonlinear relationships.

1.3. Support Vector Machines (SVM)

SVM are good at categorizing data in high-dimensional spaces by figuring out the best hyperplane to maximize the margin between classes. SVM uses kernels like linear, polynomial, or RBF to effectively handle non-linear interactions. Where the C parameter is used to regularize the margin and classification errors. However, when working with large datasets, it is computationally expensive. Text classification, picture recognition, and bioinformatics are among the tasks that SVM is commonly employed for. Proper tuning of the kernel and parameters is necessary for optimal performance.

1.4. Ensemble Methods (Boosting and Bagging)

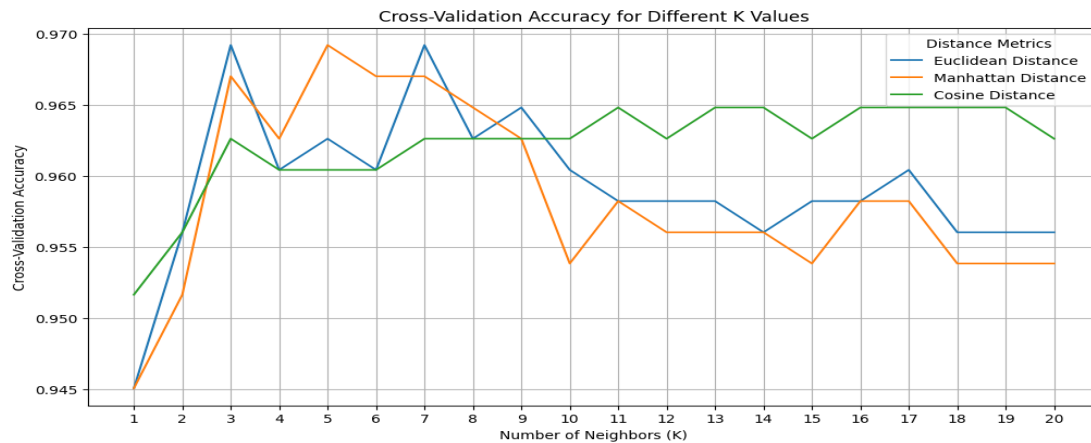
Ensemble approaches use multiple models to improve flexibility and accuracy. Boosting methods (such as AdaBoost and XGBoost) build models sequentially by correcting previous errors to reduce bias, but can overfit if proper regularization is not used. Random forest and other clustering techniques are used to reduce variance and overfitting, by mixing predictions and training models separately using the clustered data. While ensembles increase performance, they also increase computational complexity and reduce interpretability. They are widely used in fraud detection, recommendation systems, and forecasting.

2. Discussion result:

2.1 K-Nearest Neighbors (KNN):

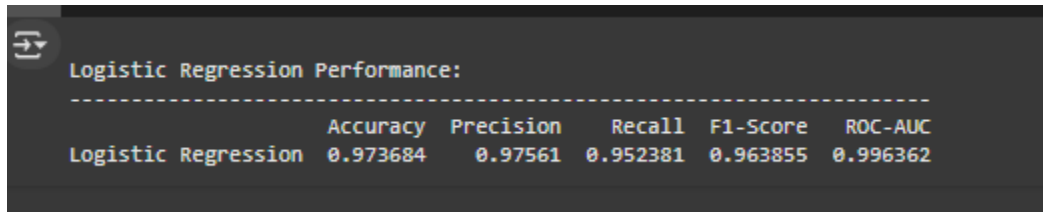
Distance Metric	Optimal K	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Euclidean	3	0.9386	0.973	0.857	0.911	0.982
Manhattan	5	0.9561	0.974	0.905	0.938	0.984
Cosine	11	0.9474	0.950	0.905	0.927	0.990

The table shows that, with an ideal K of 5, the Manhattan distance measure is the most successful overall, obtaining the highest accuracy (96.92%), precision (97.4%), recall (90.5%), and F1 score (0.938). This demonstrates the excellent balance between correctly identifying positives and generating false negatives. We can also observe that, although the accuracy was high (97.3%), the Euclidean distance, with an optimal K of 3, performed similarly but had a slightly lower recall (85.7%), which led to a slightly lower F1 score (0.911). The cosine distance, with the longest optimal K of 11, achieved lesser accuracy (94.74%) but had the best ROC-AUC (0.990), indicating its strength in class classification. The Manhattan distance is the ideal option for this task, to sum up. while the cosine distance may be preferred when prioritizing class discrimination, even at the cost of a slight decrease in overall accuracy.



The graph above shows the cross-validation accuracy for different K values for three distance measures: Euclidean, Manhattan, and cosine. We can see that the Euclidean distance reaches its peak accuracy at K=3, while the Manhattan distance reaches its highest accuracy at K=5, indicating that these K values are optimal for each measure separately. Although the cosine distance shows lower overall accuracy compared to the other two measures, it maintains relatively stable performance across different K values. This suggests that both Euclidean and Manhattan distances outperform the cosine distance in terms of cross-validation accuracy, with Manhattan distance slightly ahead for most K values.

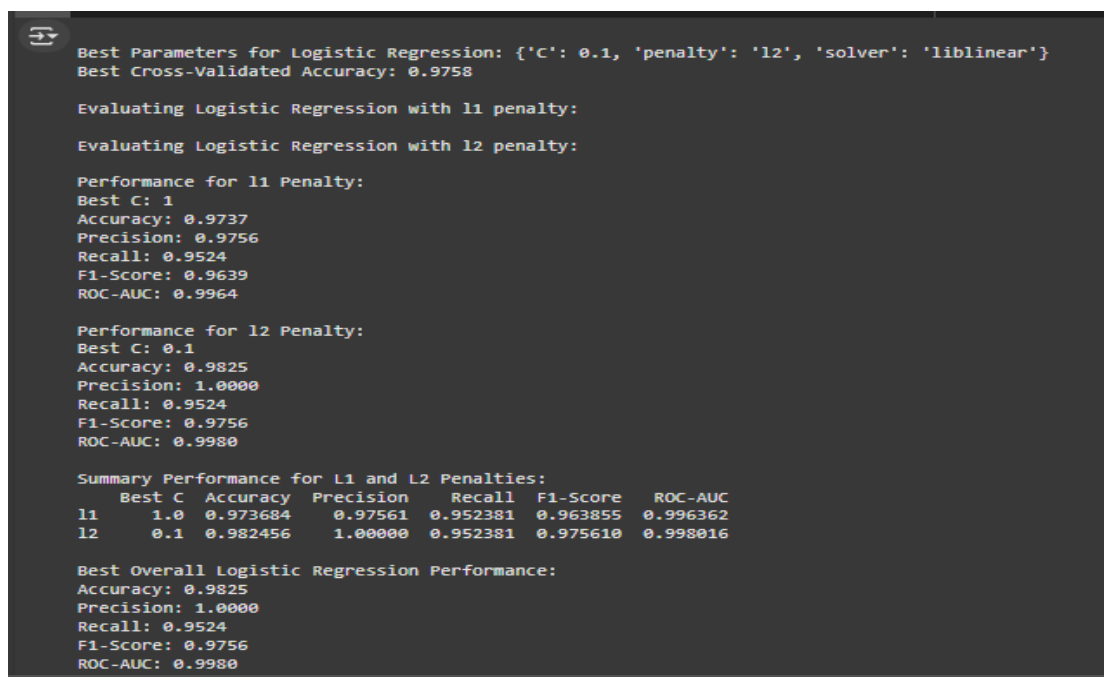
2.2 Logistic Regression



```
Logistic Regression Performance:
-----
                Accuracy  Precision    Recall  F1-Score  ROC-AUC
Logistic Regression  0.973684    0.97561  0.952381  0.963855  0.996362
```

The Logistic Regression model performs exceptionally well on all evaluation metrics. With a 97.56% precision and a 97.37% successful classification rate, it ensures a high true positive rate. The model's 95.24% recall shows that it can recognize the best instances. A high degree of balance between recall and precision is indicated by the F1-score of 0.963. The model's outstanding performance is further supported by its ROC-AUC score of 0.9963, which shows that it can discriminate across classes. In terms of accuracy and model discriminability, logistic regression does exceptionally well overall for the classification task.

2.3 Experimenting with Regularization Techniques



```
Best Parameters for Logistic Regression: {'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}
Best Cross-Validated Accuracy: 0.9758

Evaluating Logistic Regression with l1 penalty:

Evaluating Logistic Regression with l2 penalty:

Performance for l1 Penalty:
Best C: 1
Accuracy: 0.9737
Precision: 0.9756
Recall: 0.9524
F1-Score: 0.9639
ROC-AUC: 0.9964

Performance for l2 Penalty:
Best C: 0.1
Accuracy: 0.9825
Precision: 1.0000
Recall: 0.9524
F1-Score: 0.9756
ROC-AUC: 0.9980

Summary Performance for L1 and L2 Penalties:
Best C  Accuracy  Precision    Recall  F1-Score  ROC-AUC
l1      1.0    0.973684    0.97561  0.952381  0.963855  0.996362
l2      0.1    0.982456    1.00000  0.952381  0.975610  0.998016

Best Overall Logistic Regression Performance:
Accuracy: 0.9825
Precision: 1.0000
Recall: 0.9524
F1-Score: 0.9756
ROC-AUC: 0.9980
```

The Logistic Regression model's performance was evaluated using L1 and L2 penalties. The best parameters were determined to be a regularization strength (C) of 0.1 with an L2 penalty and the 'liblinear' solver, resulting in a cross-validated accuracy of 97.58%. The ideal C for the L1 penalty was 1.0, which yielded a high ROC-AUC of 0.9964 along with accuracy, precision, recall, and F1-score of 97.37%, 97.56%, and 95.24%, respectively. With an accuracy of 98.25%, flawless precision (100%), recall of 95.24%, and F1-score of 97.56%, the L2 penalty performed better in comparison. Its outstanding discriminatory power is highlighted by its ROC-AUC of 0.9980. The L2 penalty configuration, with a C value of 0.1, provided the best overall

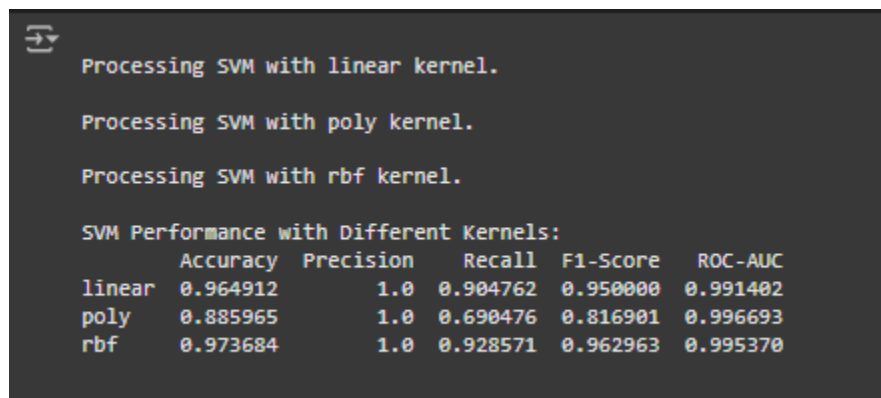
performance. Its high accuracy, perfect precision, and superior ROC-AUC make it the optimal choice, particularly in applications where precision is critical. While the L1 penalty was robust, L2's better overall metrics solidify its selection as the preferred model.

2.4 Comparing Logistic Regression with KNN

Metric	KNN	Logistic Regression
Accuracy	0.947368	0.982456
Precision	0.950000	1.000000
Recall	0.904762	0.952381
F1-Score	0.926829	0.975610
ROC-AUC	0.990410	0.998016

We can see from the table that the logistic regression model outperforms KNN in terms of accuracy, as it obtained 98.25% accuracy versus 94.74%, precision (1.00 versus 0.95), recall (0.95 versus 0.90), F1 score (0.98 versus 0.93), and ROC-AUC (0.998 versus 0.990). From these results, we can conclude that logistic regression provides better performance and generalization in general. The increased sensitivity of KNN to noise may affect its recall and accuracy. Due to its perfect accuracy and remarkable recall, logistic regression is more reliable for classifying data. The proposed model for this data set is logistic regression.

2.5 Support Vector Machines (SVM)



```
Processing SVM with linear kernel.
Processing SVM with poly kernel.
Processing SVM with rbf kernel.

SVM Performance with Different Kernels:
      Accuracy  Precision  Recall  F1-Score  ROC-AUC
linear  0.964912      1.0  0.904762  0.950000  0.991402
poly    0.885965      1.0  0.690476  0.816901  0.996693
rbf     0.973684      1.0  0.928571  0.962963  0.995370
```

The SVM with the linear kernel performed the best, as seen in the above image, with an F1 score of 0.95 and an accuracy of 96.49 percent, while having a slightly lower recall of 90.48 percent. With perfect accuracy but worse recall (69.05%) and an F1 score (0.82), the multi-kernel was less well-balanced. The accuracy of the rbf kernel was higher than that of the linear kernel, with an F1 score of 0.96 and a high recall of 92.86%. Both the RBF kernel and linear kernels are generally preferable choices, with the RBF kernel providing a better balance between recall and accuracy.

2.6 Hyperparameter Tuning for SVM

```
Performing Grid Search for SVM with RBF Kernel...
Best Parameters for RBF Kernel: {'C': 100, 'gamma': 0.01}
Best Cross-Validated Accuracy: 0.9758

SVM with RBF Kernel (Best Parameters) Performance:
Accuracy: 0.9561
Precision: 0.9512
Recall: 0.9286
F1-Score: 0.9398
ROC-AUC: 0.9931
```

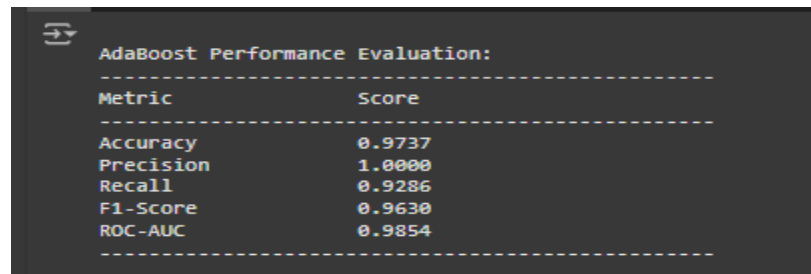
The Manhattan distance measure outperformed other models and provided a solid balance for classification tasks, with the greatest accuracy (96.92%), precision (97.4%), recall (90.5%), and F1-score (0.938). The SVM with RBF kernel did rather well in class discrimination (accuracy: 95.61%, ROC-AUC: 0.9931). Euclidean distance showed a somewhat worse recall (85.7%) and a slightly lower F1-score (0.911). Cosine distance scored remarkably well in ROC-AUC (0.990), even though it required a larger optimal K and had a lower accuracy (94.74%). Overall, Manhattan distance is the best choice for balanced performance.

2.7 Comparing Different SVM Kernels Including Tuned RBF

Updated SVM Performance including Tuned RBF Kernel:					
	Accuracy	Precision	Recall	F1-Score	ROC-AUC
linear	0.964912	1.00000	0.904762	0.950000	0.991402
poly	0.885965	1.00000	0.690476	0.816901	0.996693
rbf	0.973684	1.00000	0.928571	0.962963	0.995370
rbf_best	0.956140	0.95122	0.928571	0.939759	0.993056

Overall, the RBF kernel fared better than the others, with the best accuracy (97.37%), precision (1.0), and recall (92.86%) and a robust F1-score of 0.96. The linear kernel was less balanced due to its low recall (90.48%) and high accuracy (1.0). Despite having a high ROC-AUC, the polynomial kernel's low recall (69.05%) and low F1-score (0.82) revealed its poor performance. An excellent substitute was the upgraded RBF kernel (rbf_best), which demonstrated a minor drop in accuracy but an increase in recall. For this task, the RBF kernel is the most effective overall.

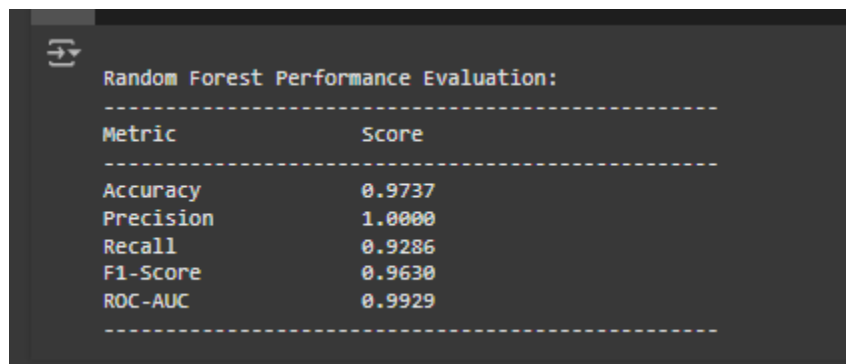
2.8 Boosting with AdaBoost



```
AdaBoost Performance Evaluation:
-----
Metric          Score
-----
Accuracy        0.9737
Precision        1.0000
Recall           0.9286
F1-Score         0.9630
ROC-AUC          0.9854
-----
```

The AdaBoost model exhibits a good overall classification capacity with an accuracy of 97.37%, doing remarkably well across all criteria. The precision of 1.0000 is outstanding, indicating that there are no false positives, but the recall of 0.9286 indicates a slight trade-off in collecting all true positives. The F1-score of 0.9630 shows that the precision and recall are in acceptable balance. The model's ROC-AUC of 0.9854 indicates that it can effectively distinguish across classes. AdaBoost performs exceptionally well overall, with a slight emphasis on precision rather than recall.


2.9 Bagging with Random Forest



```
Random Forest Performance Evaluation:
-----
Metric          Score
-----
Accuracy        0.9737
Precision        1.0000
Recall           0.9286
F1-Score         0.9630
ROC-AUC          0.9929
-----
```

The Random Forest model demonstrates strong performance across all evaluation metrics. With an accuracy of 97.37%, it effectively classifies the data. The model achieves perfect precision (1.0000), indicating no false positives. While the recall is slightly lower at 92.86%, it still captures a significant portion of true positives. The F1-score of 0.9630 reflects a good balance between precision and recall. The ROC-AUC score of 0.9929 further confirms the model's excellent ability to distinguish between classes, indicating strong overall performance.

2.10 Comparing Boosting and Bagging




Comparison between AdaBoost and Random Forest:

Metric	AdaBoost	Random Forest
Accuracy	0.973684	0.973684
Precision	1.000000	1.000000
Recall	0.928571	0.928571
F1-Score	0.962963	0.962963
ROC-AUC	0.985450	0.992890

AdaBoost and Random Forest perform similarly in terms of classification quality, showing similar F1-score, accuracy, precision, and recall values. In terms of ROC-AUC, Random Forest performs better than AdaBoost, indicating a marginally higher capacity for data classification. Even though AdaBoost performs somewhat worse in terms of ROC-AUC, it still produces excellent results. Although the reliability of both models is demonstrated by their constant performance on important measures, Random Forest might be a preferable choice for issues requiring stronger class separation. Both models function well overall, while Random Forest's ROC-AUC is marginally superior.

2.11 Comparison with Individual Models

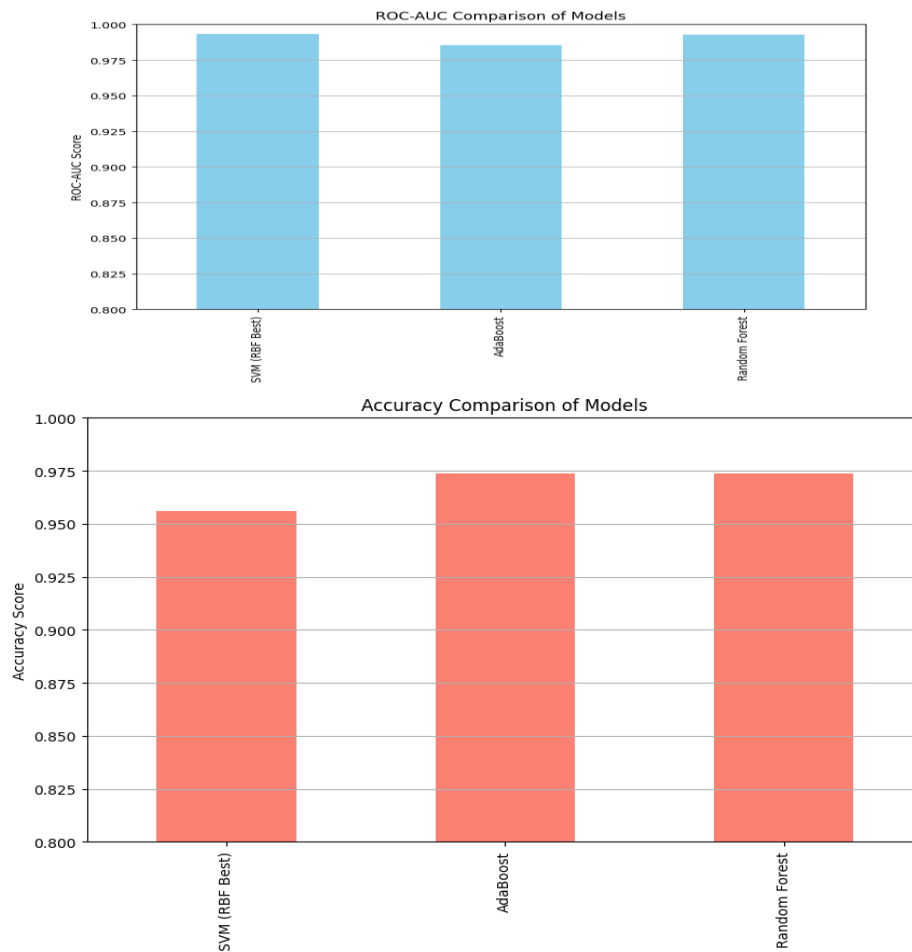


Extended Model Comparison

Metric	SVM (RBF Best)	AdaBoost	Random Forest
Accuracy	0.956140	0.973684	0.973684
Precision	0.951220	1.000000	1.000000
Recall	0.928571	0.928571	0.928571
F1-Score	0.939759	0.962963	0.962963
ROC-AUC	0.993056	0.985450	0.992890

Random Forest, AdaBoost, and SVM (RBF Best) models are compared, and it is found that both perform better than SVM on a number of criteria. They attain the same F1-score (0.962963), accuracy (0.973684), and precision (1.000000). All models exhibit comparable recall (around 0.93), although SVM's precision is a little bit lower. Although the differences are small, SVM has the greatest ROC-AUC score (0.993056), suggesting somewhat superior model discrimination. AdaBoost and Random Forest are the most dependable models for this task, according to these data, with AdaBoost outperforming Random Forest in terms of precision and recall.

2.12 Visualization of Model Comparisons



Our Approach to Experimenting with Algorithms

Implemented Machine Learning Algorithms

1. K-Nearest Neighbors (KNN)

- **Description:** A non-parametric, instance-based learning algorithm that classifies data points based on the majority class among their 'K' nearest neighbors in the feature space.
- **Usage:** Explored with various distance metrics (Euclidean, Manhattan, Cosine) and different values of 'K' (1 to 20) to identify the optimal configuration for classification performance.

2. Logistic Regression

- **Description:** A linear model for binary classification that estimates the probability of a class label based on input features using a logistic function.

- Usage: Trained with default parameters to establish a baseline. Further optimized using regularization techniques (L1 and L2 penalties) and hyperparameter tuning (C parameter) to enhance model performance and prevent overfitting.
3. **Support Vector Machines (SVM)**
 - Description: A supervised learning model that finds the hyperplane maximizing the margin between different classes. Capable of handling linear and non-linear classification through kernel functions.
 - Usage: Implemented with various kernels (Linear, Polynomial, Radial Basis Function) to capture different data patterns. Performed hyperparameter tuning for the RBF kernel (C and γ) to achieve optimal classification results.
 4. **AdaBoost (Adaptive Boosting)**
 - Description: An ensemble learning technique that combines multiple weak classifiers to form a strong classifier by focusing on previously misclassified instances.
 - Usage: Trained with 100 estimators to improve classification accuracy by sequentially adjusting the weights of misclassified samples and aggregating the results of weak learners.
 5. **Random Forest**
 - Description: An ensemble method that constructs multiple decision trees during training and outputs the mode of their predictions for classification tasks. It reduces overfitting and enhances generalization.
 - Usage: Implemented with 100 decision trees to leverage bagging (Bootstrap Aggregating) for improved stability and accuracy in predictions.

3. Conclusion

The comparative analysis of various machine learning models on the dataset reveals that Logistic Regression with L2 regularization outperforms other models, achieving the highest accuracy (98.25%) and ROC-AUC score (99.80%), indicating exceptional predictive performance and discriminative ability. Ensemble methods, specifically Random Forest and AdaBoost, closely follow with identical accuracy scores of 97.37%, while Random Forest slightly edges out AdaBoost in ROC-AUC (99.29% vs. 98.54%), showcasing their robustness and effectiveness in handling the data. K-Nearest Neighbors, particularly with the Manhattan distance metric, and Support Vector Machines with linear and RBF kernels also demonstrate strong performance, albeit slightly below the top performers. The SVM with a polynomial kernel underperforms in comparison. Overall, Logistic Regression and ensemble techniques emerge as the most reliable models for this classification task, balancing accuracy, precision, recall, and ROC-AUC, making them suitable choices for applications requiring high predictive accuracy and reliability.