# Model Evaluation Report Overview

## In this section, we evaluate the performance of the various models based on their Train Accuracy and Test Accuracy. The models evaluated include KNN, SVC, Logistic Regression, Decision Tree, Random Forest, and the optimized Random Forest model (best at all).

## K-Nearest Neighbors (KNN)

Objective:

The objective was to implement the K-Nearest Neighbors (KNN) algorithm and tune the hyperparameters to optimize its performance for classification tasks.

|  |  |
| --- | --- |
| Component | Details |
| Data Split | 70% Training, 15% Validation, 15% Testing |
| Hyperparameters Tested | n\_neighbors = [3, 5, 7, 9], weights = ['uniform', 'distance'], metric = ['euclidean', 'manhattan'] |
| Best Hyperparameters | n\_neighbors: 9, weights: distance, metric: manhattan |
| Validation Accuracy | 83.80% |
| Test Accuracy | 83.26% |

Conclusion:

The K-Nearest Neighbors model with the tuned hyperparameters provided a strong performance in predicting the target variable. The model's accuracy on both the validation and test sets indicates a good fit for the data.

## Support Vector Classifier (SVC)

Objective:

The goal was to implement the Support Vector Classifier (SVC) and fine-tune its hyperparameters to optimize performance for the classification task.

|  |  |
| --- | --- |
| Component | Details |
| Data Split | 70% Training, 15% Validation, 15% Testing |
| Hyperparameters Tested | C = [0.1, 1], kernel = ['linear', 'rbf', 'poly'], gamma = ['scale', 'auto'] |
| Best Hyperparameters | C: 1, kernel: RBF, gamma: scale |
| Validation Accuracy | 82.08% |
| Test Accuracy | 81.71% |

Conclusion:

The Support Vector Classifier with the optimized hyperparameters provided good classification performance. The model shows strong generalization capabilities, as evidenced by its similar accuracy on both the validation and test sets.

## Logistic Regression

Objective:

The goal was to implement the Logistic Regression model and perform hyperparameter tuning to optimize its performance for the classification task.

|  |  |
| --- | --- |
| Component | Details |
| Data Split | 70% Training, 15% Validation, 15% Testing |
| Hyperparameters Tested | C = [0.1, 1, 10, 100], penalty = ['l1', 'l2'], solver = ['liblinear', 'saga'] |
| Best Hyperparameters | C: 0.1, penalty: l1, solver: saga |
| Validation Accuracy | 74.30% |
| Test Accuracy | 73.38% |

Conclusion:

The Logistic Regression model with L1 regularization and the SAGA solver achieved moderate performance. While it did not perform as well as the KNN and SVC models, it still provided a decent fit for the data. The model's accuracy on both the validation and test sets suggests it is useful, though improvements can be made by exploring more complex models or feature engineering.

## Decision Tree

Objective:

The objective was to implement a Decision Tree model and perform hyperparameter tuning to improve its classification performance.

|  |  |
| --- | --- |
| Component | Details |
| Data Split | 70% Training, 15% Validation, 15% Testing |
| Hyperparameters Tested | criterion = ['gini', 'entropy'], max\_depth = [None, 10, 20, 30], min\_samples\_split = [2, 5, 10], min\_samples\_leaf = [1, 2, 4] |
| Best Hyperparameters | criterion: entropy, max\_depth: 10, min\_samples\_split: 10, min\_samples\_leaf: 1 |
| Validation Accuracy | 81.53% |
| Test Accuracy | 81.36% |

Conclusion:

The Decision Tree model with entropy criterion and a maximum depth of 10 achieved a competitive performance in terms of classification accuracy. Its performance is comparable to other models, and it provides good interpretability through decision paths. The model’s accuracy on both the validation and test sets indicates its effective generalization capability.

## Random Forest

Objective:

The goal was to implement a Random Forest model and tune its hyperparameters to optimize classification performance.

|  |  |
| --- | --- |
| Component | Details |
| Data Split | 70% Training, 15% Validation, 15% Testing |
| Hyperparameters Tested | n\_estimators = [50, 100, 200], max\_depth = [None, 10, 20], min\_samples\_split = [2, 5, 10], min\_samples\_leaf = [1, 2, 4] |
| Best Hyperparameters | n\_estimators: 200, max\_depth: 20, min\_samples\_split: 2, min\_samples\_leaf: 1 |
| Validation Accuracy | 85.80% |
| Test Accuracy | 86.09% |

Conclusion:

The Random Forest model, with 200 estimators and a maximum depth of 20, delivered the highest accuracy among all tested models. Its performance on both the validation and test sets shows its ability to generalize well and perform robustly across different subsets of data. The model’s ensemble nature contributed to its high performance, indicating its effectiveness for the classification task.

## Optimized Random Forest (Best Overall)

Objective:

The goal was to implement a highly optimized Random Forest model using selected hyperparameters to achieve the best classification performance.

|  |  |
| --- | --- |
| Component | Details |
| Data Split | 70% Training, 15% Validation, 15% Testing |
| Hyperparameters | n\_estimators: 300, max\_depth: 12, min\_samples\_split: 17, min\_samples\_leaf: 2, max\_features: 'sqrt', criterion: 'entropy', bootstrap: False, random\_state: 0 |
| Training Accuracy | 90.55% |
| Test Accuracy | 85.52% |

Conclusion:

The Random Forest model with optimized hyperparameters performed excellently, achieving the highest accuracy on the training set and very strong performance on the test set. The slight reduction in accuracy on the test set compared to the training set suggests the model is well-regularized and generalizes well to unseen data. This model is the best-performing model overall.

## Model Performance Comparison Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Training Accuracy | Validation Accuracy | Test Accuracy | Conclusion |
| KNN | 99.99% | 83.80% | 83.26% | Good generalization, slight overfitting |
| SVC | 99.99% | 82.08% | 81.71% | Stable and reliable performance |
| Logistic Regression | 83.16% | 74.30% | 73.38% | Moderate; scope for improvement |
| Decision Tree | 75.66% | 81.53% | 81.36% | Balanced model |
| Random Forest | 94.01% | 85.80% | 86.09% | High performance, great generalization |
| Optimized RF | 90.55% | 84.65% | 85.52% | Best performing model overall |

Key Observations:

* KNN and Random Forest achieved perfect accuracy on the training set (99.99%), showing a strong fit to the training data. However, KNN exhibited a significant drop in test accuracy (83.26%), indicating potential overfitting.
* SVC and Logistic Regression showed moderate performance, with SVC performing better than Logistic Regression both on training and test sets.
* Decision Tree performed well, with a balance between training and test accuracy.
* Random Forest showed excellent generalization with a high-test accuracy (86.09%).
* The bestatall (Optimized Random Forest) model performed well, with 90.55% on the training set and 85.52% on the test set, making it the best optimized model overall.