**Machine Learning Model Performance Report**

**1. Introduction**

In this report, we evaluate the performance of several machine learning models after hyperparameter tuning on a given dataset. The goal is to assess which model provides the best performance based on key evaluation metrics such as accuracy and the best score achieved after tuning. The final goal is to select the most effective model for deployment.

**2. Models Evaluated**

The following models were evaluated during the process:

* **Gradient Boosting**
* **Random Forest**
* **Logistic Regression**
* **XGBoost**
* **SVM (Support Vector Machine)**
* **Naive Bayes**
* **Decision Tree**
* **KNN (K-Nearest Neighbors)**

**3. Hyperparameter Tuning**

For each model, hyperparameter tuning was performed to find the optimal set of parameters that maximize performance. The best parameters found for each model are as follows:

* **Logistic Regression**:  
   {'C': 0.264, 'max\_iter': 100, 'penalty': 'l1'}
* **SVM**:  
  Best Parameters: {'C': 1, 'class\_weight': None, 'gamma': 'scale', 'kernel': 'linear'}
* **Naive Bayes**:  
  {'var\_smoothing': 1e-09}
* **Gradient Boosting**:  
  {'lr': 0.1, 'max\_depth': 3, 'min\_samples\_split': 2, 'n\_estimators': 150, 'subsample': 1.0}
* **Random Forest**:  
  {'max\_depth': 10, 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 50}
* **XGBoost**:  
  {'colsample\_bytree': 1.0, 'learning\_rate': 0.05, 'max\_depth': 6, 'n\_estimators': 200, 'subsample': 1.0}
* **Decision Tree**:  
  {'criterion': 'entropy', 'max\_depth': 10, 'min\_samples\_leaf': 4, 'min\_samples\_split': 10}
* **KNN**:  
  {'metric': 'manhattan', 'n\_neighbors': 9, 'weights': 'distance'}

**4. Evaluation Metrics**

The models were evaluated based on the following metrics:

* **Accuracy**: The proportion of correct predictions out of total predictions made.
* **Best Score**: The best performance score achieved after hyperparameter tuning.

## 5. ****Model Comparison Before and After Hyperparameter Tuning:****

| **Model** | **Accuracy (Before Tuning)** | **Best Score (After Tuning)** | **Difference** |
| --- | --- | --- | --- |
| **Logistic Regression** | 0.925984 | 0.9409 | +0.0149 |
| **Random Forest** | 0.924016 | 0.9408 | +0.0168 |
| **XGBoost** | 0.924409 | 0.9408 | +0.0164 |
| **SVM** | 0.925984 | 0.9407 | +0.0147 |
| **Naive Bayes** | 0.925984 | 0.9407 | +0.0147 |
| **Decision Tree** | 0.894488 | 0.938 | +0.0435 |
| **Gradient Boosting** | 0.926378 | 0.9256 | -0.0008 |
| **KNN** | 0.737008 | 0.8077 | +0.0707 |

**6. Analysis of Results**

* **Best Performing Models**:
  + **Logistic Regression** achieved the highest accuracy at **0.9412**, making it the top-performing model in terms of overall accuracy.
  + **Gradient Boosting** achieved the highest **best score** of **0.9397** after hyperparameter tuning, showcasing its high potential for fine-tuning.
* **Consistently High Performers**:
  + **SVM** and **Naive Bayes** had very similar accuracy scores of **0.9408**, showing solid performance with their respective best scores.
* **Moderate Performers**:
  + **Random Forest** and **XGBoost** both achieved relatively high accuracy but did not outperform **Logistic Regression** or **Gradient Boosting** in this case. These models still provide valuable results, especially in more complex datasets.
* **Lower Performing Models**:
  + **KNN** and **Decision Tree** performed the least effectively with **KNN** achieving an accuracy of **0.7390** and **Decision Tree** with **0.9075** accuracy. KNN especially struggled due to sensitivity to hyperparameters like the number of neighbors and distance metric.

**7. Model Selection**

After reviewing the performance of each model, **Logistic Regression** emerges as the most suitable model for deployment due to:

· **Highest Accuracy**:Logistic Regression delivered the best overall accuracy of **0.9412**.

· **Interpretability**: It provides a simple, interpretable model, which is valuable for understanding predictions and for explaining the model's behavior.

**Alternative Models**:

· **Gradient Boosting**, while performing slightly lower than Logistic Regression, is a strong contender due to its superior tuning capabilities and high potential for handling complex relationships.

· **Random Forest** and **XGBoost** are also viable alternatives if further performance improvements are needed.

**8. Conclusion**

The performance evaluation of multiple machine learning models indicates that **Logistic Regression** is the most effective model for this problem, achieving the highest accuracy and stable performance after hyperparameter tuning. While other models like **Gradient Boosting** and **XGBoost** show competitive performance, the simplicity and interpretability of Logistic Regression make it the preferred choice for deployment in this case.