Hyperparameter Sensitivity Trends Across Dataset Types in the UCI Machine Learning Repository

#CourseCode : **CSIT332** Principles of Machine Learning
#Instructor : Prof. Manoranjan Das
#Student Name: Shanay Sheth, Pranav Shinde

1. Introduction

Hyperparameters play a crucial role in shaping the behavior and performance of machine learning models. Even minor adjustments can lead to noticeable changes in Accuracy, F1 Score, and AUC, especially on datasets that are noisy, imbalanced, or contain complex nonlinear patterns.

While many studies only report the *best* model performance, this project focuses on understanding *how stable* the performance is when hyperparameters vary. The goal is to measure **hyperparameter sensitivity** and determine which models are robust and which require precise tuning.

This report analyzes five datasets using four common classification models to determine how dataset characteristics influence model stability. The work includes both **individual hyperparameter analysis** and **full Grid Search variance analysis**.

2. Datasets

Five datasets from the UCI Machine Learning Repository were selected due to their variety in size, noise level, class balance, and feature types:

- **Abalone** – 4177 × 8, noisy, nonlinear, moderately balanced
- **Diabetes** – small, irregular time-series, mixed coded features
- **Mushroom** – 8124 × 22, fully categorical, highly separable
- **Student Dropout & Academic Success** – 4424 × 36, imbalanced, heterogeneous
- **Default of Credit Card Clients** – 30,000 × 23, financial behavioral data, imbalanced

These datasets together cover: numerical, categorical, mixed-type, balanced, imbalanced, and noisy environments.

3. Methodology

3.1 Preprocessing

- Handling missing values
- Encoding categorical variables
- Standardizing numerical features
- Applying balancing techniques when needed (SMOTE or class weights)

3.2 Models Evaluated

- Decision Tree
- K-Nearest Neighbors (KNN)
- Logistic Regression
- Support Vector Machine (SVM)

3.3 Hyperparameter Search

A structured **Grid Search** with 5-fold cross-validation was used.

Metrics measured:

- Accuracy Variance
- F1 Variance
- AUC Variance
- Combined Mean Variance (overall model stability score)

3.4 Individual Hyperparameter Sensitivity

Before performing Grid Search, each model’s hyperparameters were tested independently by varying one at a time while fixing the rest. This helped identify:

- The most impactful hyperparameters
- Which hyperparameters caused instability
- Whether some could be removed from Grid Search

4. Individual Hyperparameter Insights

Some hyperparameters consistently caused high variance across datasets:

- **KNN:** `n_neighbors`
  → Strong influence on performance, especially on imbalanced datasets.
- **SVM:** `gamma`
  → Extremely sensitive in noisy datasets like Abalone.
- **Logistic Regression:** `C`
  → Sensitive when features are not properly scaled.

- **Decision Tree:** `max_depth`
  → Moderate sensitivity but still the most unstable model overall.

These insights supported the need for a combined Grid Search + sensitivity variance analysis.

5. Grid Search Results by Dataset

5.1 Diabetes Dataset

Model	Overall Variance
Decision Tree	0.004
KNN	0.000433
Logistic Regression	0
SVM	0

Interpretation:
Decision Tree shows high instability due to its greedy splitting. Linear models remain stable despite the dataset's irregularities.

5.2 Mushroom Dataset

Model	Overall Variance
Logistic Regression	0
SVM	0
KNN	0
Decision Tree	0.000067

Interpretation:
The dataset is almost perfectly separable, so nearly all models show perfect stability.

5.3 Student Dropout Dataset

Model	Overall Variance
Logistic Regression	0
SVM	0
KNN	0
Decision Tree	0.000067

Interpretation:
Linear and kernel methods handle the structured data well. Decision Tree remains slightly sensitive.

5.4 Abalone Dataset

Model	Overall Variance
Decision Tree	0.004
Logistic Regression	0.000433
KNN	0
SVM	0

Interpretation:
The dataset has nonlinear, noisy biological measurements. Trees overfit; KNN and SVM generalize better.

5.5 Credit Card Default Dataset

Model	Overall Variance
Logistic Regression	0
SVM	0
KNN	0
Decision Tree	0.000048

Interpretation:
Strong correlations (especially repayment history) help stabilize all models except Decision Tree.

6. Cross-Dataset Model Stability Summary

Model	Mean Variance (Lower = More Stable)
SVM	0
Logistic Regression	0.0000867
KNN	0.0000867
Decision Tree	0.0016365

Ranking (Most → Least Stable)

1. **SVM** — perfectly stable
2. **Logistic Regression** — stable and predictable
3. **KNN** — stable but dependent on dataset structure
4. **Decision Tree** — highly sensitive and unstable

Interpretation:
Mathematically constrained models (SVM, LR) are stable even under hyperparameter changes. Decision Trees, due to greedy splitting and high flexibility, fluctuate the most.

7. General Sensitivity Observations

Dataset Characteristics That Increase Sensitivity

- High noise
- Class imbalance
- Nonlinear boundaries
- High dimensionality
- Heterogeneous/mixed features

Patterns Observed

- Large datasets tend to be more stable (Credit Card)
- Highly separable datasets show almost zero sensitivity (Mushroom)
- Noisy/small datasets display large variance (Abalone, Diabetes)

These findings align with both hyperparameter-level and Grid Search–level analyses.

8. Conclusion

This project demonstrates that hyperparameter sensitivity varies both across models and across datasets. The study highlights several key takeaways:

- **SVM is consistently the most stable classifier**, unaffected even by large hyperparameter changes.
- **Logistic Regression provides similarly stable performance**, especially with proper scaling.
- **KNN performs well** but is sensitive to `n_neighbors` and data imbalance.
- **Decision Trees are highly unstable**, with even small changes causing noticeable variance.

From a practical standpoint, this means:

- For real-world, risk-sensitive tasks → prefer SVM or Logistic Regression.
- For exploratory or highly nonlinear tasks → use KNN or Decision Trees but tune carefully.
- Understanding dataset properties is just as important as tuning the model.

.