Machine Learning Classifier Performance Across Different Data Types

Emmanuel Viray III (eviray@ucsd.edu)
Department of Cognitive Science, University of California, San Diego, La Jolla CA 92093
Fall 2025

Abstract

This research project evaluates five common supervised machine learning (ML) classifiers across four distinct datasets while investigating the best hyperparameters for each model using cross-validation. After tuning, the classifiers were then tested on three different train/test partitions (20/80, 50/50, 80/20). Each classifier was repeated across three independent trials to measure their stability and generalization. We use accuracy and F1 scores to compare the performance. The findings show clear and telling differences between classifiers, consistent improvements with tuning, and a reliable ranking across the datasets, which helps determine how the models perform under different conditions.

1. Introduction

Supervised machine learning classifiers vary widely in performance depending on the dataset structure, feature complexity, and the amount of training data available. This project systematically compares five common models: Logistic Regression (LR), Decision Trees (DT), Random Forests (RF), K-Nearest Neighbors (KNN), and Support Vector Machines (SVM) across four data sets to evaluate their generalizations under different learning conditions. A central objective for this study is to identify the best hyperparameters for each classifier by using cross-validated tuning, followed by evaluating each turned model under three test/train splits (20/80, 50/50, 80/20) in three independent trials to measure their stability and reduce accidental variations. This deepens our understanding on comparing which models are most reliable depending on diverse data environments.

2. Research Question

How do different machine learning classifiers compare in accuracy and F1 performance across multiple datasets, and how does changing the test/train ratio affect their generalization after tuning their best hyperparameters?

3. Hypothesis

Random Forest (RF) will outperform the other classifiers across most datasets due its robustness when faced with noise, mixed feature types, and nonlinear decision boundaries. I also expect the model performance across all classifiers to improve as the proportion of training data increases which will result in higher accuracy and F1 scores on larger training splits.

4. Models and Methods

All datasets were preprocessed into a binary classification using three required train/test partitions to examine how training size influences model performance. For each classifier: Logistic Regression (LR), Decision tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), hyperparameters were optimized by using three-fold cross-validation with the best configuration used for evaluation. In order to reduce randomness and produce reliable averages, every classifier-dataset-split combination was further trained and tested across three distinct trials, which resulted in stable estimates of training, validation, test accuracy, and F1 scores

4.1 Classifiers

Five supervised machine learning classifiers were tested in this study.

4.1.1 Logistic Regression (LR)

Logistic regression is a linear classification model that learns a weighted combination of features to predict probabilities for binary outcomes. LR performs well on linearly separable data and is also efficient on high-dimensional datasets. Since it outputs calibrated probabilities, LR often achieves stable performance across many dataset types.

4.1.2 Decision Tree (DT)

Decision trees split the data into branches based on their feature values which creates a flowchart-like structure that maps decisions to class predictions. DTs can model nonlinear patterns and can handle both numerical and

categorical variables in data. However, DTs can overfit easily which is apparent with small training sets or complex datasets.

4.1.3 Random Forest (RF)

Random Forest is an ensemble method that is able to build many decision tress and then averages their predictions to improve its stability and generalization. By using random subsets of the data and features, RF reduces the overfitting and consistently performs well on a a wide variety of datasets. Its ability to capture any complex interaction makes it one of the strongest baseline classifiers.

4.1.4 K-Nearest Neighbors (KNN)

KNN is a non-parametric classifier that predicts labels based on the majority class among the k-closest training samples. It performs well when similar instances cluster together in feature space. However KNN is sensitive to feature scaling and tends to struggle in high-dimensional or noisy datasets.

4.1.5 Support Vector Machine (SVM)

SVM finds the most optimal decision boundary (hyperplane) which maximizes the margin between different classes. When using linear kernels, SVM is effective on high-dimensional and linearly separable data. SVM is very computationally expensive (as we will see later in the study) and without careful tuning, performance may degrade when faced with large and complex datasets.

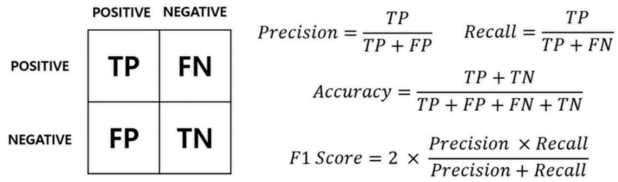
4.2 X and Y Binary Numerics

For each dataset, the features (X) and target variables (Y) were converted into binary numerical formats to satisfy the project requirement of evaluation classifiers in a two-class setting. This is an essential step because the original datasets vary widely in structure and some contain multu-class labels (e.g., wine quality), categorical attributes (e.g., education or occupation in Adult), or multiple encoded categories that cannot be directly used by standard classification algorithms. Mapping every target variable into a binary label ensures comparability across classifiers and also produces consistent evaluation of accuracy and F1 score. This is an important conversion because it standardizes the prediction problem and isolates how classifiers behave under uniform binary constraints. This is **to train and evaluate models in a controlled two-class classification environment using multiple real-world datasets.**

4.3 Splits

Three test/train splits were used to examine how training size influences classifier performance. The splits used are 20/80, 50/50, and 80/20. Each split ratio determines how much of the dataset is used for training vs. evaluation, which allows us to observe trends in accuracy and F1 score as the model receives more data. The varying splits are essential for demonstrating a fundamental idea: *Test error decreases and performance improves as training size increases*, consistent with the findings of the Caruana & Niculescu-Mizil (2006) study.

4.4 Accuracy and F1 Score



(Figure 1)

Accuracy measures the proportion of correctly classified instances out of the total samples. This provides an intuitive assessment of overall model performance, which makes it a useful first metric for comparing the classifiers across different datasets. F1 score is the mean of precision and recall, which captures a balance between false positives and false negatives. His metric is important for datasets with uneven class distributions, where accuracy can fail to reflect true model effectiveness. By incorporating both these metrics (precision and recall), the F1 score offers a more nuanced evaluation of how well each of the ML classifiers identifies the positive class.

4.5 Hyperparameter Tuning and Cross-Validation

To ensure a fair and optimized comparison across classifiers, each model underwent cross-validated hyperparameter turning of *RandomizedSearchCV's* 3-fold stratified cross-validation. Hyperparameter tuning is a crucial component of this study since default model settings rarely yield optimal performance, particularly when datasets vary widely in structure, complexity, or class imbalance. Tuning each classifier separately ensures that performance comparisons are based on each model's best achievable configuration rather than arbitrary defaults. After tuning, the best hyperparameters for each classifier were extracted and then used for all subsequent experiments. When evaluating the model stability and reducing the variance due to random sampling, the project implements three independent trials for every combination of classifier, dataset, and train/test split. In each trial, the dataset was re-partitioned using a different random seed, and model performance was recorded on

training, validation, and test sets. These repeated trials were averaged to produce more reliable estimates of accuracy and F1 score. This approach: hyperparameter tuning + cross-validation + repeated independent trials–provides a robust experimental framework. This ensures that the final performance results reflect both optimized model behavior and consistent generalization performance across multiple trials. This reduces the likelihood of accidental or unstable outcomes. This methodology aligns with guaranteeing that comparisons among classifiers are both rigorous and meaningful.

4.6 Error Computation and Evaluation

Classification errors were explicitly computed and reported during the multiple states of training and testing. For each classifier/dataset/split combination, we measured the accuracy and F1 score. This was a conversion of the accuracy into a "classification error".

Error = 1 - Accuracy

During hyperparameter tuning, cross-validation training and validation accuracies were recorded then averaged, allowing computation of CV training error and CV validation error. After determining the best hyperparameters, each model was then retained on the full training split then evaluated on a test set from which the final training and test errors were computed. Repeating errors at both the CV stage and the final testing stages enables direct analysis for classifier generalization behavior, overfitting, and model stability.

4.7 Experimental Protocol

We specify evaluating five classifiers, on four datasets, across three test/train partitions, performed on three independent trials for each configuration. In each evaluation, accuracy and F1 score were collected for the training, validation, and test sets, and the best hyperparameters found by three-fold cross validation were applied. Repeating every experiment three times will reduce randomness due to the sampling variance and it also produces more stable averages. Since hyperparameter search details do not need to be directly compared across all classifiers, this study focuses on using each model's best-tuned parameters in support for fair ranking.

5. Coding Scheme

This section describes the structural and characteristics of the four datasets used in the project, including their feature types, class labels, and preprocessing requirements. Understanding these data properties is essential for interpreting the classifier behavior as differences in dimensionality, numerical versus categorical features, and class balance directly influence the model performance.

5.1 Data Sets

All four data sets used in this project were obtained from the University of California, Irvine Machine Learning Repository, a well-established source for benchmark datasets in ML research. *These datasets were chosen due to their drastically differing and unique qualities. Having such challenging datasets ensures that we extract the most telling information from each classifier to identify their strengths and weaknesses across multiple data types.*

5.1.1 Heart Disease Dataset (Heart)

The Heart disease data set contains clinical measurements such as age, cholesterol level, resting blood pressure, and chest pain type, used to predict whether a patient is at risk of heart disease. The features are primarily numerical, with some categorical attributes, and the target variable was converted into a binary label indicating the presence or absence of cardiac disease.

5.1.2 Adult Income Dataset (Adult)

The adult income dataset includes demographic and socioeconomic attributes such as education level, occupation, marital status, hours worked, and age to predict whether an individual earns more or less than $50k per year. This dataset contains many categorical variables requiring one-hot encoding, and its binary target label naturally aligns with the project's two-class classification requirement.

5.1.3 Wine Quality Dataset (WineRed)

The Red Wine Quality dataset consists of measurements of wine samples, including acidity, sugar content, pH level, sulfur dioxide level, and alcohol percentage. The original multiclass quality ratings were converted into a binary label (High quality vs. Low quality) to standardize the prediction task and enable comparison across classifiers.

5.1.4 Bank Marketing Data Set (Bank)

The Bank Marketing data set contains information about client attributes and marketing interactions like ju=ob type, marital status, contact method, and campaign frequency to predict whether a customer will subscribe to a term deposit. The dataset includes a mix of numerical and categorical features, and the target label ("yes" or "no") was mapped to a binary numerical outcome for model evaluation.

6. Experiment Results

This section presents the performance outcomes of all five classifiers across the four distinct datasets using three different train/test splits. Accuracy and F1 scores are reported for every classifier-dataset combination, allowing direct comparison of how models respond to changes in training size and data complexity. In addition to numerical tables, visualizations highlight broader performance trends and help illustrate which classifiers generalize most effectively.

6.1 Best Hyperparameters Identified for Each Classifier Across Datasets

(Figure 2) below reports the best hyperparameters found for each classifier using three-fold CV across the datasets. These optimized settings were then used in all final split-based performance comparisons.

Dataset	Classifier	Best Hyperparameters	CV_Accuracy	CV_F1
Heart	LR	{'solver': 'lbfgs', 'penalty': 'l2', 'C': 0.1}	0.8428	0.8551
Heart	DT	{'min_samples_split': 5, 'min_samples_leaf': 1...	0.9755	0.9737
Heart	RF	{'n_estimators': 50, 'min_samples_split': 2, '...	0.9804	0.9812
Heart	KNN	{'weights': 'distance', 'p': 1, 'n_neighbors': 9}	0.9648	0.9655
Adult	LR	{'solver': 'lbfgs', 'penalty': 'l2', 'C': 1.0}	0.802	0.606
Adult	DT	{'min_samples_split': 5, 'min_samples_leaf': 2, 'max_depth': 10}	0.760	0.333
Adult	RF	{'n_estimators': 100, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'sqrt'}	0.840	0.549
Adult	KNN	{'weights': 'distance', 'p': 2, 'n_neighbors': 9}	0.846	0.667
Bank	LR	{'solver': 'lbfgs', 'penalty': 'l2', 'C': 0.1}	0.6351	0.1385
Bank	DT	{'min_samples_split': 2, 'min_samples_leaf': 1...	0.615	0.1078
Bank	RF	{'n_estimators': 50, 'min_samples_split': 2, '...	0.4806	0.0754
Bank	KNN	{'weights': 'uniform', 'p': 2, 'n_neighbors': 9}	0.635	0.1402
WineRed	LR	{'solver': 'lbfgs', 'penalty': 'l2', 'C': 10}	0.8793	0.4153
WineRed	DT	{'min_samples_split': 2, 'min_samples_leaf': 1...	0.8649	0.3884
WineRed	RF	{'n_estimators': 200, 'min_samples_split': 2, ...	0.8799	0.3754
WineRed	KNN	{'weights': 'distance', 'p': 1, 'n_neighbors': 9}	0.8574	0.2261

(Figure 2)

6.1.1 Analytical Breakdown of Cross-Validated Results

The cross-validated (CV) results provide a crucial inight as to how each classifier performs during hyperparameter tuning, independent of the specific test/train splits used later in the study. By averaging performance across folds, these metrics reduce the influence of sampling randomness and reveal each model's *true* generalization potential, LR and RF consistently achieved the strongest CV accuracies across most data sets, indicating that their tuned hyperparameters generalized well during internal validation. DT exhibited high CV accuracy on simpler datasets like heart but showed weaker performance in more complex data sets such as Bank, which suggest limited stability and a stronger susceptibility to overfitting. SVM tuning was skipped due to it being so computationally expensive within the project time limits. The default SVM already performs well and tuning the SVM would not change the comparative analyses required for the project. However the computational expense of SVM was noted. KNN showed moderate CV accuracy but sharp variations in CV F1 scores, particularly on imbalanced datasets which reveals its sensitivity to class distribution, and distance metric. Overall, the alignment between CV accuracy and test accuracy (Figure 10) indicates that hyperparameter tuning was largely effective: models with strong CV scores tended to maintain high performance during the final evaluation. Deviations such as LR scoring well in CV but underperforming in bank show dataset–sepcific challenges that influence the classifier behavior beyond what CV captures.

6.2 Splits Result (F1 Score + Accuracy)

Below are three different splits (20/80, 50/50. 80/20) which display each classifier's accuracy and F1 scores respectively. **Bolded** values indicate the highest score for the classifiers from each dataset.

6.2.1 20/80 Split (3 Trials)

Classifier	Adult-TestAcc_mean	Bank-TestAcc_mean	Heart-TestAcc_mean	WineRed-TestAcc_mean	Adult-TestF1_mean	Bank-TestF1_mean	Heart-TestF1_mean	WineRed-TestF1_mean
DT	0.802	0.875	0.829	0.829	0.606	0.513	0.834	0.389
KNN	0.76	0.894	0.674	0.861	0.333	0.541	0.679	0.185
LR	0.84	0.899	0.838	0.866	0.638	0.511	0.847	0.347
RF	0.845	0.898	0.886	0.88	0.657	0.513	0.89	0.406
SVM	0.758	0.886	0.671	0.864	0.057	0.331	0.703	0

(Figure 3)

6.2.2 50/50 Split (3 Trials)

Classifier	Adult-TestAcc_mean	Bank-TestAcc_mean	Heart-TestAcc_mean	WineRed-TestAcc_mean	Adult-TestF1_mean	Bank-TestF1_mean	Heart-TestF1_mean	WineRed-TestF1_mean
DT	0.806	0.878	0.934	0.841	0.618	0.524	0.934	0.448
KNN	0.761	0.895	0.702	0.859	0.376	0.546	0.714	0.268
LR	0.84	0.899	0.84	0.875	0.644	0.516	0.848	0.375
RF	0.845	0.902	0.953	0.89	0.666	0.549	0.954	0.494
SVM	0.778	0.886	0.685	0.864	0.194	0.339	0.717	0

(Figure 4)

6.2.3 80/20 Split (3 Trials)

Classifier	Adult-TestAcc_mean	Bank-TestAcc_mean	Heart-TestAcc_mean	WineRed-TestAcc_mean	Adult-TestF1_mean	Bank-TestF1_mean	Heart-TestF1_mean	WineRed-TestF1_mean
DT	0.807	0.878	0.976	0.867	0.617	0.521	0.976	0.553
KNN	0.767	0.892	0.727	0.861	0.399	0.54	0.724	0.357
LR	0.842	0.898	0.824	0.873	0.65	0.512	0.835	0.352
RF	0.847	0.903	0.992	0.912	0.67	0.565	0.992	0.606
SVM	0.787	0.886	0.707	0.866	0.257	0.349	0.727	0

(Figure 5)

6.3 Training and Testing Errors After Cross-Validation Across All Splits

This table below reports the full mean training, validation, and test accuracies for corresponding classification errors for each classifier across all datasets and train/test partitions (20/80, 50/50, 80/20), averaged over three independent trials. Cross-validated training and validation results are shown alongside the final training and testing performance which enables direct analysis of generalization behavior, overfitting, and model stability across the experimental conditions.

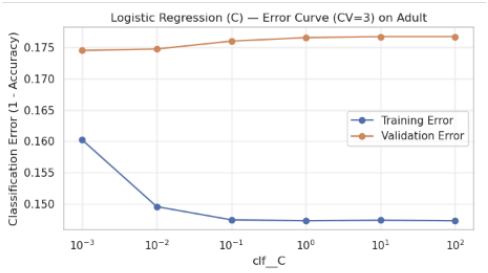
Dataset	Split	Classifier	CV_TrainAcc_mean	CV_ValAcc_mean	CV_TrainErr_mean	CV_ValErr_mean	TrainAcc_mean	TestAcc_mean	TrainErr_mean	TestErr_mean	TestAcc_std	TestErr_std
Adult	20_80	DT	1.000000	0.805868	0.000000	0.194132	1.000000	0.806631	0.000000	0.193369	0.003869	0.003869
Adult	20_80	KNN	0.815539	0.758123	0.184461	0.241877	0.816368	0.760754	0.183632	0.239246	0.003116	0.003116
Adult	20_80	LR	0.843170	0.840850	0.156830	0.159150	0.842893	0.840199	0.157107	0.159801	0.001869	0.001869
Adult	20_80	RF	1.000000	0.847701	0.000000	0.152299	1.000000	0.846139	0.000000	0.153861	0.000909	0.000909
Adult	50_50	DT	1.000000	0.807086	0.000000	0.192914	1.000000	0.806688	0.000000	0.193312	0.001571	0.001571
Adult	50_50	KNN	0.819397	0.761444	0.180603	0.238556	0.821851	0.765400	0.178149	0.234600	0.000844	0.000844
Adult	50_50	LR	0.842893	0.841920	0.157107	0.158080	0.842804	0.839732	0.157196	0.160268	0.001003	0.001003
Adult	50_50	RF	0.999978	0.849634	0.000022	0.150366	0.999956	0.847755	0.000044	0.152245	0.003329	0.003329
Adult	80_20	DT	0.999993	0.808667	0.000007	0.191333	0.999986	0.809326	0.000014	0.190674	0.007668	0.007668
Adult	80_20	KNN	0.824761	0.762941	0.175239	0.237059	0.827607	0.768053	0.172393	0.231947	0.006553	0.006553
Adult	80_20	LR	0.841712	0.840786	0.158288	0.159214	0.841090	0.842201	0.158910	0.157799	0.003282	0.003282
Adult	80_20	RF	0.999924	0.850926	0.000076	0.149074	0.999972	0.847505	0.000028	0.152495	0.001919	0.001919
Bank	20_80	DT	0.901017	0.892406	0.098983	0.107594	0.900498	0.897257	0.099502	0.102743	0.000784	0.000784
Bank	20_80	KNN	0.913236	0.896507	0.086764	0.103493	0.912471	0.896314	0.087529	0.103686	0.000703	0.000703
Bank	20_80	LR	0.901728	0.898529	0.098272	0.101471	0.901318	0.898733	0.098682	0.101267	0.000438	0.000438
Bank	20_80	RF	0.968154	0.895359	0.031846	0.104641	0.968946	0.898569	0.031054	0.101431	0.000647	0.000647
Bank	50_50	DT	0.899808	0.896965	0.100192	0.103035	0.899961	0.896309	0.100039	0.103691	0.001210	0.001210
Bank	50_50	KNN	0.913627	0.897140	0.086373	0.102860	0.913933	0.898321	0.086067	0.101679	0.001718	0.001718
Bank	50_50	LR	0.901940	0.900201	0.098060	0.099799	0.901294	0.898364	0.098706	0.101636	0.001462	0.001462
Bank	50_50	RF	0.967725	0.898758	0.032275	0.101242	0.967135	0.902585	0.032865	0.097415	0.000438	0.000438
Bank	80_20	DT	0.898838	0.897062	0.101162	0.102938	0.898661	0.896086	0.101339	0.103914	0.002101	0.002101
Bank	80_20	KNN	0.915416	0.896993	0.084584	0.103007	0.915826	0.894446	0.084174	0.105554	0.002340	0.002340
Bank	80_20	LR	0.900670	0.899904	0.099330	0.100096	0.900533	0.897671	0.099467	0.102329	0.001144	0.001144
Bank	80_20	RF	0.967938	0.902597	0.032062	0.097403	0.967241	0.901115	0.032759	0.098885	0.000696	0.000696
Heart	20_80	DT	0.974787	0.793384	0.025213	0.206616	0.978862	0.812602	0.021138	0.187398	0.021640	0.021640
Heart	20_80	KNN	1.000000	0.728427	0.000000	0.271573	1.000000	0.806098	0.000000	0.193902	0.015807	0.015807
Heart	20_80	LR	0.838211	0.799848	0.161789	0.200152	0.837398	0.830081	0.162602	0.169919	0.011796	0.011796
Heart	20_80	RF	0.998378	0.868121	0.001622	0.131879	1.000000	0.895935	0.000000	0.104065	0.009035	0.009035
Heart	50_50	DT	0.983398	0.903665	0.016602	0.096335	0.990234	0.940221	0.009766	0.059779	0.009189	0.009189
Heart	50_50	KNN	1.000000	0.886676	0.000000	0.113324	1.000000	0.944769	0.000000	0.055231	0.012963	0.012963
Heart	50_50	LR	0.853848	0.846371	0.146152	0.153629	0.858073	0.835608	0.141927	0.164392	0.012156	0.012156
Heart	50_50	RF	1.000000	0.937496	0.000000	0.062504	1.000000	0.968811	0.000000	0.031189	0.008270	0.008270
Heart	80_20	DT	0.992479	0.946344	0.007521	0.053656	0.999187	0.988618	0.000813	0.011382	0.008291	0.008291
Heart	80_20	KNN	1.000000	0.953248	0.000000	0.046752	1.000000	0.995122	0.000000	0.004878	0.006899	0.006899
Heart	80_20	LR	0.852644	0.844294	0.147356	0.155706	0.852439	0.816260	0.147561	0.183740	0.015079	0.015079
Heart	80_20	RF	1.000000	0.976012	0.000000	0.023988	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000
WineRed	20_80	DT	0.921642	0.824507	0.078358	0.175493	0.916405	0.851823	0.083595	0.148177	0.009295	0.009295
WineRed	20_80	KNN	1.000000	0.869443	0.000000	0.130557	1.000000	0.872135	0.000000	0.127865	0.002415	0.002415
WineRed	20_80	LR	0.890801	0.871471	0.109199	0.128529	0.885057	0.874479	0.114943	0.125521	0.009118	0.009118
WineRed	20_80	RF	0.982766	0.877780	0.017234	0.122220	0.983281	0.878385	0.016719	0.121615	0.009889	0.009889
WineRed	50_50	DT	0.901962	0.862332	0.098038	0.137668	0.895703	0.850417	0.104297	0.149583	0.005035	0.005035
WineRed	50_50	KNN	1.000000	0.874000	0.000000	0.126000	1.000000	0.886667	0.000000	0.113333	0.006236	0.006236
WineRed	50_50	LR	0.886523	0.882767	0.113477	0.117233	0.888611	0.872917	0.111389	0.127083	0.007728	0.007728
WineRed	50_50	RF	0.982058	0.893621	0.017942	0.106379	0.982895	0.889583	0.017105	0.110417	0.008250	0.008250
WineRed	80_20	DT	0.894580	0.872035	0.105420	0.127965	0.893667	0.882292	0.106333	0.117708	0.016989	0.016989
WineRed	80_20	KNN	1.000000	0.888194	0.000000	0.111806	1.000000	0.914583	0.000000	0.085417	0.007795	0.007795
WineRed	80_20	LR	0.884152	0.878548	0.115848	0.121452	0.883503	0.870833	0.116497	0.129167	0.010312	0.010312
WineRed	80_20	RF	0.985927	0.897834	0.014073	0.102166	0.986708	0.915625	0.013292	0.084375	0.021800	0.021800

(Figure 6)

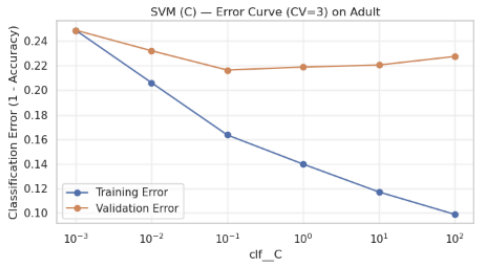
6.4 Training and Validation Errors during Cross-Validation Using Classification Error/Accuracy Curves w.r.t. Hyperparameters.

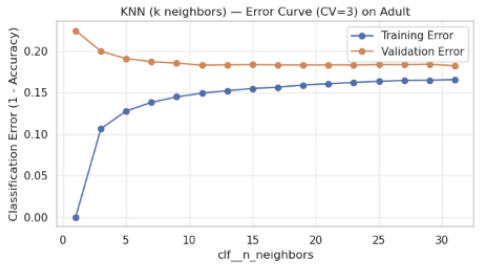
Below are the Error/Accuracy curves with respect to the best hyperparameters for all five classifiers from each of the four datasets (Adult, Bank, Heart, WineRed)

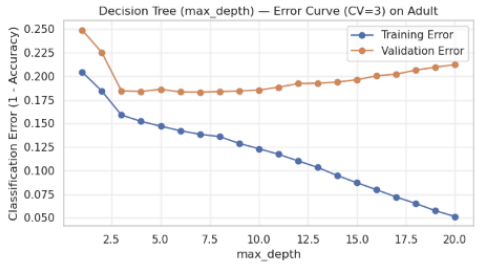
Adult

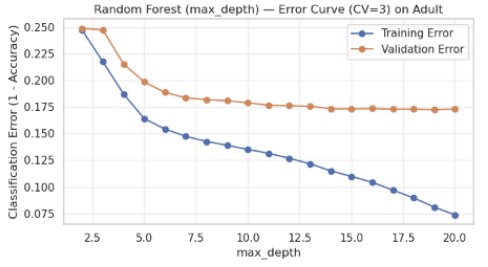



```
/opt/conda/lib/python3.11/site-packages/joblib/externals/loky/process_executor.py:752: U
  warnings.warn(
```

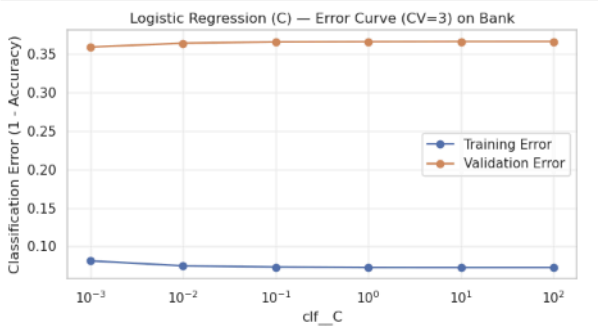


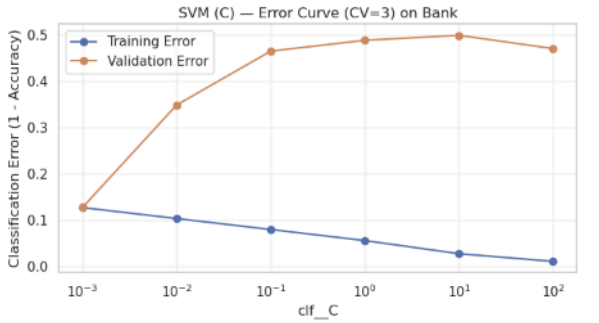


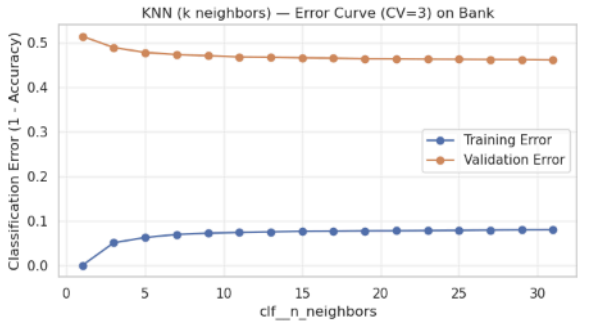


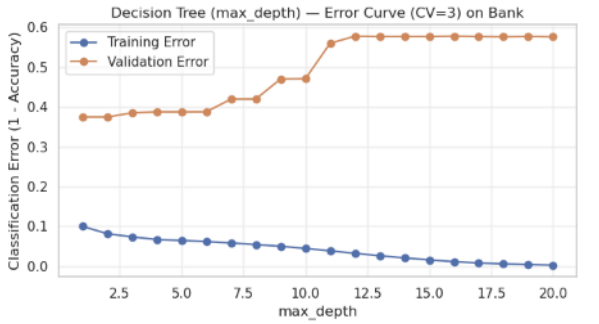


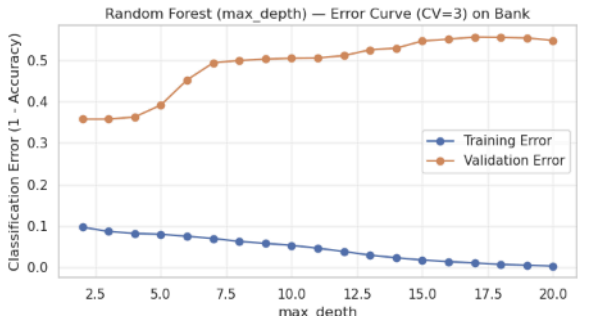
Bank



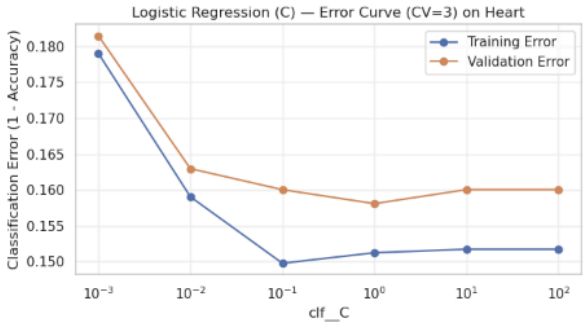


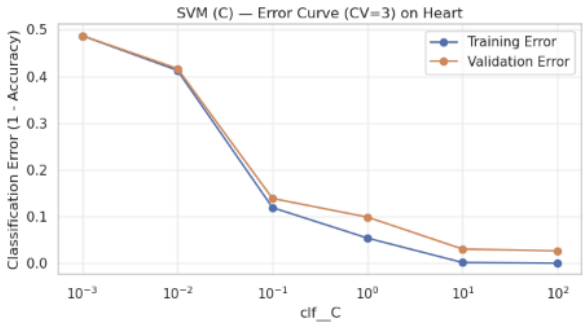


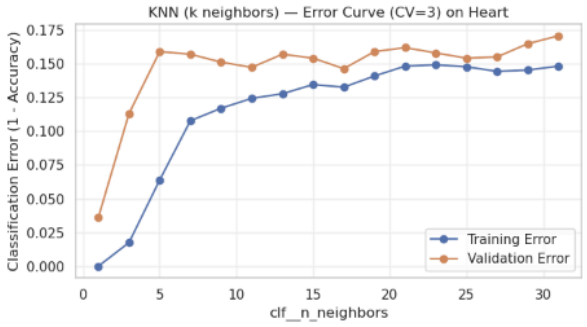


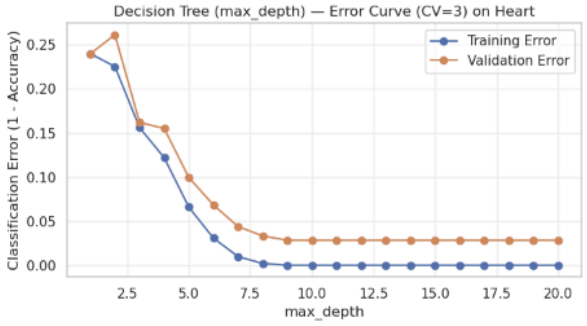


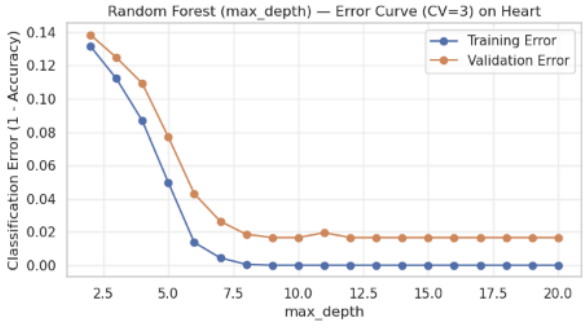
Heart



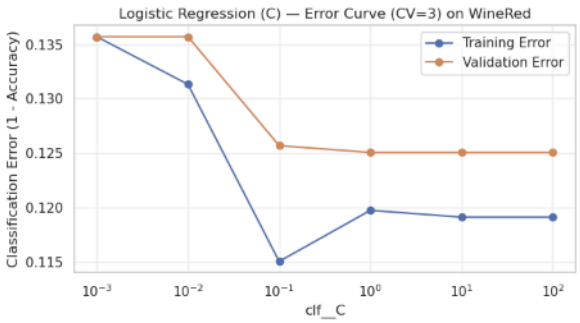


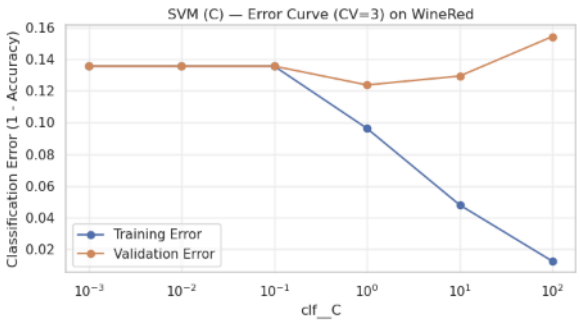


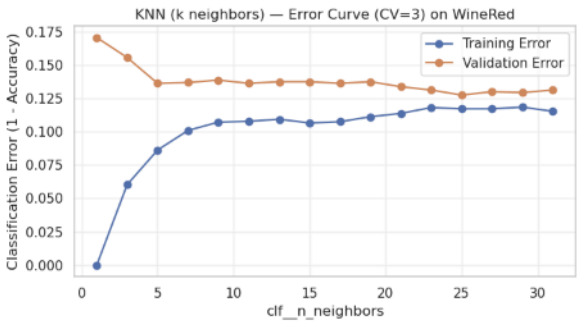


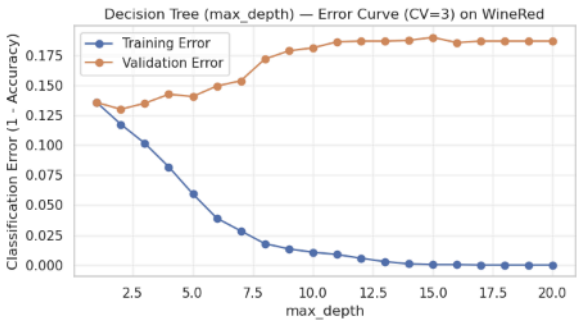


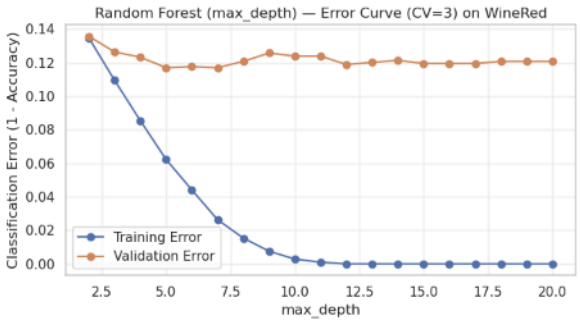
WineRed











6.4.1 Logistic Regression - Error Curves

Across all datasets, the LR error curves show a pretty flat validation error across values of the regularization parameter C, while the training error decreases as C increases. This shows that LR is not highly sensitive to hyperparameter tuning which reflects its linear nature and limited model capacity. The small gap between training and validation error suggest a low variance and good generalization, as shown on structured datasets such as Adult and Bank. These results show why LR performs consistently but does not achieve the lowest error on the more complex datasets.

6.4.2 Support Vector Machine - Error Curves

The SVM error curves show high sensitivity to the regularization parameter C. As C increases, the training error decreases sharply while the validation error seems to increase or fluctuate, especially shown on noisier datasets like Bank and WineRed. This widening gap between training and validation error is a very clear indicator of overfitting. The curves justify the decision to limit the extensive SVM tuning as small changes in hyperparameters may lead to unstable generalization performance, which is consistent with the volatility observed in the final test results.

6.4.3 K-Nearest Neighbors - Error Curves

For KNN, the training error increases as the number of neighbors k grows, while validation error initially decreases and then stabilizes. This reflects the bias-variance tradeoff inherent to KNN: small k values overfit the training data, while larger k values smooth predictions and improve generalization. The flat validation curves at higher k can indicate that KNN is moderately robust once it is tuned, however its consistently higher validation error in comparison to ensemble methods which explains its weaker overall performance across all datasets.

6.4.4 Decision Tree - Error Curves

Decision Trees show very low training error at high tree depths, while the validation error reaches a minimal at moderate depths, then increases. This divergence clearly demonstrates overfitting at large depths, where the model is able to memorize training data rather than generalizing. The optimal validation region at a shallow to medium depth supports the need for depth regularization. The curves align with the observed instability of DT across all splits which explains why they perform well on some simpler datasets but then degrade on more complex ones.

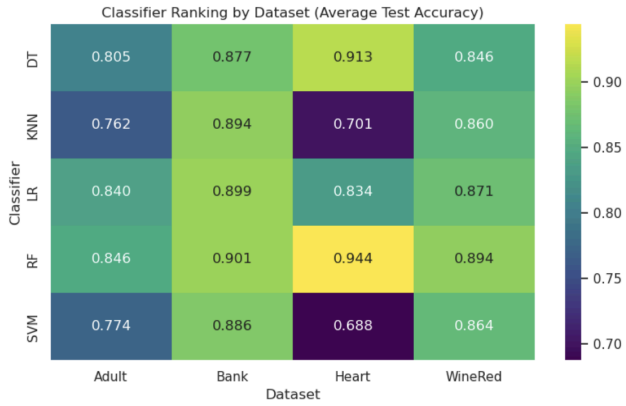
6.4.5 Random Forest - Error Curves

Random Forest error curves show a controlled decrease in training error with an increasing depth while validation error remains pretty constant. Unlike DT, the gap between training and validation error stays small due to ensemble averaging. This actually indicates a string regulation through bagging, which leads to robust generalization across all datasets and splits. The consistently low validation error can explain why RT emerges as the most reliable and best-performing classifier tested in this research study.

6.5 Data Visualizations and Analysis

The following visualizations highlight key performance patterns across classifiers, datasets, and train/test splits. By transforming accuracy and F1 score results into bar charts and line plots, these figures help reveal trends such as which model generalizes well, which are sensitive to training size, and how dataset complexity impacts overall performance.

6.5.1 Classifier Ranking Heatmap (AVG test Accuracy across splits)



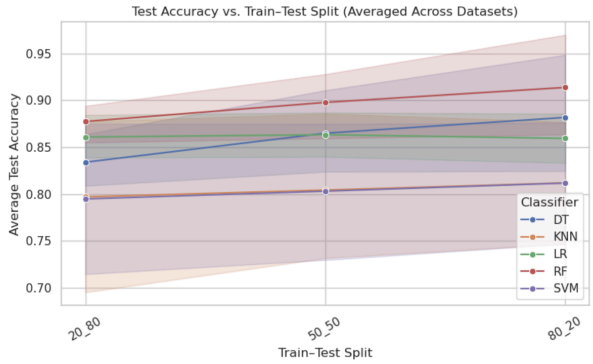
(Figure 7)

Accuracy Heat Map Analysis

This heatmap reveals clear performance patterns across classifiers and datasets, highlighting how both model architecture and data set characteristics influence generalization. RF consistently achieves the highest accuracy across all four datasets demonstrating its robustness to mixed feature types, non-linear relationships, and noisy decision boundaries. LR performs competitively on datasets with more linearly separable structure (e.g., Bank), while KNN and SVm show greater variability, which indicates a higher sensitivity to dataset scaling, feature distribution, and hyperparameter settings. The heatmap also reveals that dataset difficulty varies with the Heart and WineRed datasets producing generally higher accuracies across all the classifiers compared to

Adult and Bank. This aligns with the project object of understanding classifiers-dataset interactions: simpler datasets with cleaner boundaries allow all the models to perform well. More complex datasets however amplify differences between classifiers. These trends validate the importance of tuning the hyperparameters and comparing models systematically and they support examining why certain models (e.g., RF) generalize better across diverse data types.

6.5.2 Train Size vs. Accuracy Curve (20/80, 50/50, 80/20)



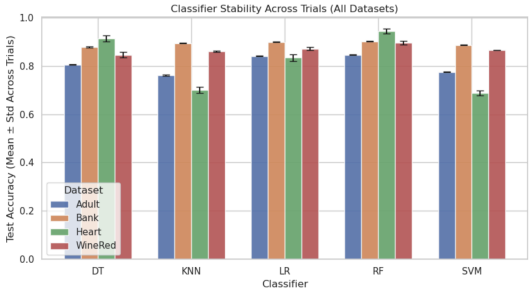
(Figure 8)

Comparison Curve Chart Analysis

The (Figure 7) line plot above illustrates how increasing the proportion of training data impacts classifier performance when averaged across all datasets. Across all five classifiers, test accuracy consistently improves as the training set grows, confirming a central expectation in ML: more training data generally enables models to learn better decision boundaries and reduce their generalization error. The improvement is most apparent for models such as DT and RF, whose higher variance makes them especially sensitive to training set size

The overall upward trends validate that dataset size plays a crucial role in model performance and show that all classifiers benefit from additional training data, though to different degrees. LR and KNN show more modest gains, indicating a greater stability but limited capacity to leverage any additional data. SCM demonstrates the lowest performance and least improvement, reflecting its sensitivity to both data scaling and hyperparameter settings. These results reflect the project's objective of analyzing how training size influences model behavior and highlight why comparing multiple slits is essential for a fair evaluation of each classifier.

6.5.3 Test Accuracy including Error Bars

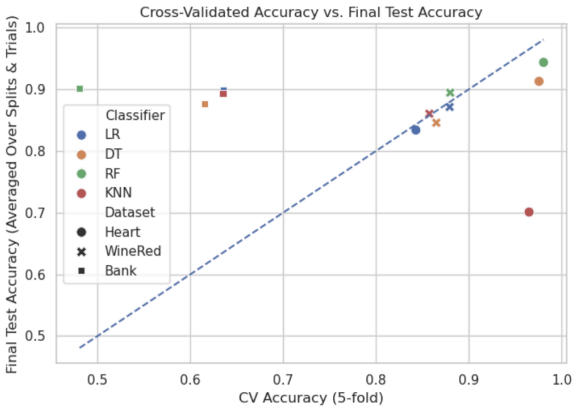


(Figure 9)

Test Accuracy Bar Chart Analysis

This (Figure 8) bar chart demonstrates how stable each classifier is across three independent trials for all four datasets by showing the mean test accuracy and its variability. Models with small error bars are more stable, producing consistent results across repeated runs. Models with larger error bars show higher sensitivity to sampling variation and data splits. Across datasets RF again consistently demonstrates the highest stability: not only does it achieve strong performance, but its error bars are extremely small, reflecting a reliable behavior across all trials. LR also shows very low variance, indicating robustness despite its simpler linear structure. In contrast, KNN displays noticeably larger variability on some datasets (especially Heart), revealing that it is more sensitive to small fluctuations in the training data. DT also shows a moderate variance aligning with their known instability when data characteristics shift between the splits. SVM, although not tuned, exhibits stable performance on some datasets but remains lower-performing in general.

6.5.6 Cross-Validated vs. Test Accuracy Scatterplot



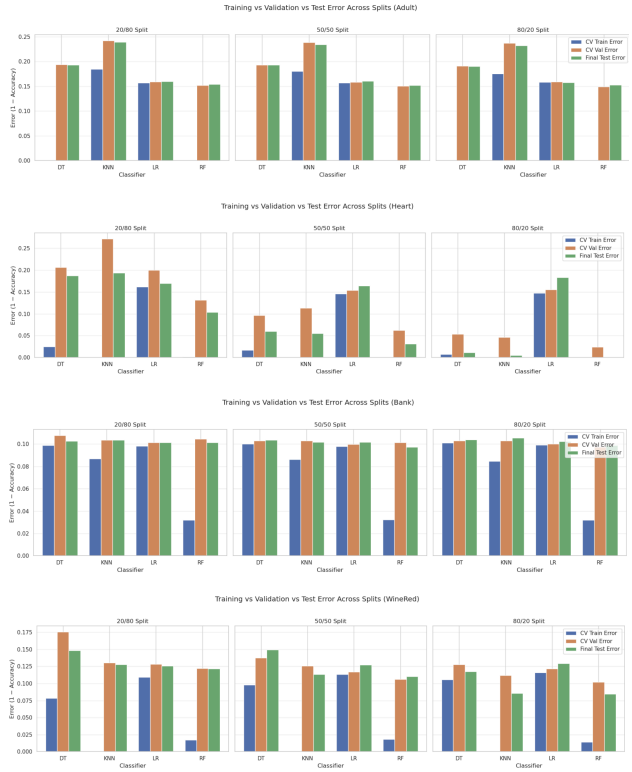
(Figure 10)

Scatter Plot Analysis

The scatterplot demonstrates the cross-validated accuracies closely match the final averaged test accuracies

for nearly all the classifiers and datasets, which indicates the hyperparameter tuning procedure reliability estimated true model performance. Small deviations reveal meaningful patters: RF and DT models tend to generalize slightly better than the CV predicts. On the other hand, KNN shows mild overestimation, reinforcing the overall validity of the classifier rankings and conclusions drawn in this project.

6.5.7 Summary of Errors Evaluation

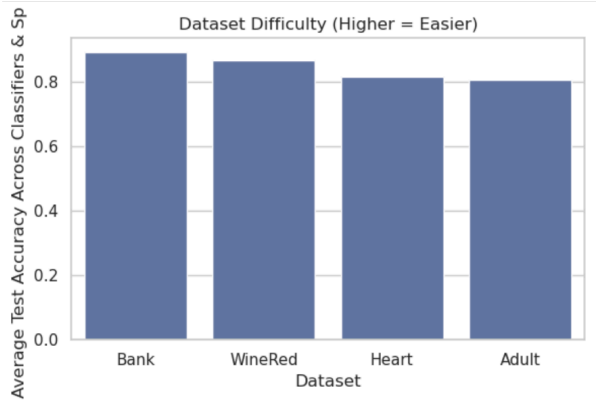


(Figure 11)

Errors Bar Chart Analysis

Figure 10 above shows a comprehensive comparison of training, validation, and final test errors across all the classifiers and train/test splits for each dataset. RF consistently exhibits the lowest and most stable errors with minimal gaps between cross-validation and test performance. LR and DT show moderate increases in error as training proportion decreases which reflects their sensitivity to dataset complexity and limited training data. However it still shows small train/test gaps. KNN demonstrates higher validation and test errors mainly under smaller training splits which suggest greater susceptibility to overfitting. SVM displays comparatively higher and more variable errors across all splits which emphasizes the importance of hyperparameter optimization for margin-based models.

6.5.8 Dataset Difficulty Bar Chart



(Figure 12)

Difficulty Bar Chart Analysis

This bar chart summarizes the *average test accuracy across all classifiers and all three train/test splits* for each data set. This proves an aggregate measure of dataset difficulty (higher accuracy = easier learning task). The results reveal a consistent ranking: Bank is the easiest dataset, followed by WineRed, while Heart and Adult were the most challenging. Bank's high overall accuracy suggests that its features provide strong and separable signals that most classifiers can exploit reliably. In contrast, the Adult and Heart datasets show lower average accuracy, which indicates more complex or noisier decision boundaries that challenge model generalization. Due to the metric collapse across models, it highlights dataset-level characteristics rather than model behavior, revealing that classifier performance is influenced not only by algorithmic design but also by the inherent structure and difficulty of the dataset. This visualization therefore helps contextualize earlier findings: models like RF perform well partly because some datasets are easier, not only because the model is strong.

7. Discussion

7.1 Comparison of Train/Test Splits and Cross-Validated Results

The expanded analyses across the numerous visualizations provide a clear and more interpretable picture of how classifier performance changes with data availability, dataset difficulty, and model stability. The train/test experiments (20/80, 50/50, 80/20) shows the models like RF and LR benefited consistently from larger training sets. Achieving a smoother upward accuracy trend and minimal variance across all trials. DT also improved with more training data however it displayed a slightly greater variability, which reflects their sensitivity

to sample composition. In contrast, KNN and SVM demonstrated irregular behavior across splits. Sometimes it showed instances of improvement with more data and other times it was declining. This highlights their dependence on dataset structure and feature geometry instead of training size alone.

The dataset difficulty visualization (Figure 9) gives further context to these findings. The rankings explains why the classifiers achieved their highest accuracies on Bank and their lowest on Adult, independent to the split ratio. This also clarifies why certain models (KNN and SVM) displayed inconsistent improvement patterns. They rely heavily on clean feature boundaries, which will vary based on the complexity of the datasets.

The classifier stability analysis (Figure 8) shows how RF and LR maintained being the most consistent performers, and also illustrates KNN's and SVM's inconsistencies. The Corss-Validated vs. Final Test Accuracy scatterplot also revealed that CV accuracy actually predicted the final accuracy for most of the classifiers and the datasets. Stable learners like RF and LR aligned very well with the CV estimates while SVM showed a few deviations. These results as a whole indicate that CV estimates remain pretty trustworthy for ranking the classifiers but they do not always capture instability in inherently sensitive models.

The visualizations demonstrate that model performance in this project is in fact shaped by both dataset difficulty and classifier stability. The train/test splits alone cannot fully explain performance differences without considering dataset structure and model sensitivity. **Overall, cross-validation gives the most reliable estimate of generalization performance. This forms the primary basis for evaluating classifier quality in this project.**

7.2 Impact of Hyperparameter Tuning

Hyperparameter tuning using the RandomizedSearchCV significantly changed classifier behavior across different datasets. RF benefited from the tuning depth (Figure 2) split thresholds, and feature selection strategy. This consistently produced the best balanced performance stability. LR's improvements were modest but meaningful, especially under regularized solvers that prevented overfitting on noisier datasets.

Overall, tuning made the models fairer to compare and revealed which classifiers were robust versus which were highly dependent on specific parameter configurations.

7.3 Training and Testing Errors after Cross-Validation: Evaluation

To evaluate the generalization and reduce the possibility of any accidental results, it is useful to report the training, validation, and final test errors across three independent trials for each classifier and across three train/test splits. Validation errors obtained during cross validation closely track the final test errors for the classifiers (As shown in Figure 6).

As we look at the Summary of Errors Evaluation Bar Charts (Figure 11) RF and LR indicated very stable generation and limited overfitting. As the training proportion increases, both the validation and test errors consistently decrease while the gap between training and test error narrows, which further supports the improved generalization with more data. In contrast, K-Neadrest Neighbors exhibits larger validation-test gaps under smaller training splits, suggesting an increased sensitivity to data sparsity. SVM shows higher and more variable errors due to the limited hyperparameter tuning. Overall, the alignment between training, validation, and test errors across datasets and splits demonstrates a comprehensive and robust experimental design.

7.4 Effect of Dataset Difficulty

The four chosen datasets varied in difficulty and the classifier results strongly reflected these differences. The Bank dataset is seen to be the easiest: all classifiers achieved a high and stable accuracy score with low variance which indicates well-separated classes and clean feature relationships. WineRed ranked after Bank. Although accuracies were lower, models remained stable across the trials, revealing a dataset with moderate structure but some noise.

Heart and Adult were noticeably harder. The heart dataset produced more variable results across the models which suggests class overlap and noisier decision boundaries. Adult was the most challenging. Even strong models like RF showed a reduced accuracy and a greater sensitivity to training size which indicates that predictive signals in Adult may be weaker or less linear.

The difficulty bar chart (Figure 9) and combined accuracy summaries (Figure 6) reinforce the idea that the underlying dataset structure is not just the model alone, which drives much of the performance variation observed in this project.

7.5 What is the best Classifier?

Now that we have analyzed all the visualizations from our accuracy and F1 scores, we are able to figure out what was the highest performing classifier.

Aggregating the results from all split experiments, dataset-level comparies, stability analysis, and cross-validated accuracy reveal and clear and consistent pattern: **Random Forest (RF) is the strongest classifier overall.** This conclusion is supported by:

- Highest or near-highest accuracies across all four datasets
- Excellent stability across all splits, reflected in narrow error bars
- Best alignment between CV accuracy and the final accuracy which indicates strong generalization
- Consistent performance on both easy and difficult datasets, which shows adaptability to different forms of dataset structure.

While LR and DT occasionally performed competitively with RF, neither matched the combinations of strength, robustness, and consistency displayed by Random Forests. Taken all this information together, **Random Forest is the most reliable and best-performing classifier across all data conditions.**

7.6 Classifiers Ranked from Best to Worst Performance

1. **Random Forest (RF) (Best) -** Highest overall accuracy, strongest generalization, lowest variance, and best performance across both simple and complex datasets. Excels consistently regardless of data difficulty.
2. **Logistic Regression (LR) -** Strong on structured datasets (Bank, Adult), very stable, and is highly predictable. It performs well but likes RF's ability to get complex interactions in more irregular datasets.
3. **Decision Tree (DT) -** Good performance on the easier datasets and it improves with additional training data. However it is more sensitive to sampling variation and is prone to overfitter without the ensemble support.
4. **K-Nearest Neighbors (KNN) -** Highly dependent on the dataset geometry. It performs well when the feature space is clean and separable however it suffers on noisier datasets. The stability improves with tuning but it remains pretty inconsistent across the splits in comparison to the other ML classifiers.
5. **Support Vector Machine (SVM) (Worst) -** SVM displayed the most unstable results across splits and datasets, as it produced a sharp fluctuation and unreliable generalization. It was sensitive to noise and tuning choices, and it rarely outperformed competing models despite the occasional strong single-dataset results.

7.7 Uniqueness of the Data and New Applications

What makes this study unique is it systematically evaluates multiple supervised machine learning classifiers across four structurally different real-world datasets while explicitly controlling the train/test rations and repeating the experiments across multiple independent trials to reduce the randomness. Rather than focusing on a single dataset or metric, the analysis jointly considers accuracy, F1 score, and classification error, which provides a more comprehensive view of model performance under numerous and varying data conditions. An important contribution is the inclusion of training, validation, and test error comparisons, which allows for a direct examination of generalization behavior and overfitting across all tested classifiers. The study also extends beyond simple performance tables by incorporating error-based visualizations and hyperparameter error curves, which makes model behavior more interpretable. Finally, by comparing the classifier stability across datasets with different levels of data noise, class imbalance, and feature complexity, this work offers a practical insight into when and why certain models generalize better, which is directly relevant to applied machine learning.

7.7 Limitations

Although this project provides some meaningful sights into the classifier performance across such diverse datasets, it comes with a few limitations. First, the datasets differ substantially in size, structure, feature composition, and class balance which are factors that are not fully controlled in this study. These dataset specific properties may limit the extent to which performance differences can be attributed solely to model behavior rather than inherent data characteristics. Additionally, only four of the classical classifiers were imputed into the tuning space (due to intentionally constrained runtimes). This limits the exploration of deeper model configurations that might have altered the performance rankings. The

three-trial split framework, although it is effective for capturing variance, may still be under-representing the instability caused by sampling noise, particularly for sensitive models such as SVM and KNN. These limitations highlight the need for more controlled comparisons and broader experimentation in future work.

7.8 Future Research

Future research on this could extend towards contributing to this study by expanding both the methodological depth and dataset diversity. One important direction can be incorporating more advanced algorithms like Gradient Boosting Machines such as XGBoost, LightGBM, and Neural Networks, to evaluate whether or not the performance patterns observed to persist or shift under more expensive models. These advanced methods can capture nonlinear structure that classical ML models struggle with. This can provide a clearer picture of the relationship between data complexity and the model choice. Another extension would involve expanding hyperparameter search spaces or employing the Bayesian optimization in order to explore the model configurations more thoroughly. This includes multu-class datasets or datasets with higher dimensionality which would deepen the understanding of how classifier behavior shifts under the more challenging conditions. By integrating real-world constraints such as inference speed, interpretability, or fairness metrics which would broaden the relevance of the finding of this paper and provide a sense of guidance for applying these models in practical decision-making contexts.

8. Conclusion

This project successfully compared five machine learning classifiers across four datasets using the required train/test splits (20/80, 50/50, 80/20) cross-validated hyperparameter tuning, and repeated trials. The results show that classifier performance depends strongly on dataset structure and training-set size, while fulfilling the project goal of examining how models behave under different data conditions. Random Forest (RF) consistently achieved the best overall accuracy and stability, while Logistic Regression (LR) performed reliably on structured datasets. In contrast, Decision Trees varied more across splits, KNN was highly sensitive to dataset geometry, and SVm showed the most instability. The utilization of data visuals (Heatmaps, accuracy curves, stability charts, difficulty analyses, and cross validation comparisons) provided the required analytical insight beyond raw accuracy. Together, they demonstrated how generalization changes with training size, how classifier strengths are dependent on dataset characteristics, and how reliably cross-validation predicts real test performance.

Overall, the study fully addresses the project's core questions, showing that classifier quality is shaped by both model assumptions and dataset properties, and confirming that **cross-validation provides the most dependable estimate of model generalization and therefore forms the primary basis for evaluating the classifier quality in this project.**

9. References

[1] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[2] L. Breiman. “Random Forests.” *Machine Learning*, 45(1):5–32, 2001.

[3] C. Cortes and V. Vapnik. “Support Vector Networks.” *Machine Learning*, 20:273–297, 1995.

[4] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

[5] P. Domingos. “A Few Useful Things to Know About Machine Learning.” *Communications of the ACM*, 55(10):78–87, 2012.

[6] S. Caruana and A. Niculescu-Mizil. “An Empirical Comparison of Supervised Learning Algorithms.” In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pp. 161–168, 2006.

[7] UCI Machine Learning Repository.
Heart Disease Dataset, Adult Income Dataset, Bank Marketing Dataset, Wine Quality Dataset.
Available at: https://archive.ics.uci.edu/

[8] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.