A Systematic Comparison of Machine Learning Classifiers Across Diverse Datasets and Training Set Sizes

Dhanashree Kulkarni [1]

Abstract

This project evaluates the performance of three classifiers on three different datasets across 3 training set sizes. Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine models were trained and tested on each dataset using three train to test ratios (20/80, 50/50, and 80/20) to examine differences in performance across datasets, and the effect of training size on model accuracy, precision, and recall. Across nearly all datasets and partitions, Random Forest achieved the highest overall performance. The changes in evaluation metrics for each classifier as training size increases is also analyzed.

1. Introduction

Machine learning classification has diverse applications, including medical diagnosis, customer behavior modeling, etc. While deep learning has gained prominence, classical machine learning algorithms remain important due to their interpretability, computational efficiency, and robust performance across various domains.

This project examines the behavior of four classifiers in a binary classification exercise. Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine classifiers were trained on 3 different datasets. By evaluating all four models under identical conditions, this project aims to provide a clear comparison of their strengths, weaknesses, and consistency across domains and sizes of training data.

2. Methods

2.1. Datasets

The three datasets that were chosen for this study were all selected from Kaggle, which is an online community for Data Science and Machine Learning enthusiasts.

The first dataset is the Stroke Prediction dataset which contains about 5,000 rows of data. This dataset contains 10 features that can be used to predict the occurrence of a stroke for an individual based on their medical history and demographics. This dataset is extremely imbalanced because most of the people observed did not experience a stroke. There are about 4,800 entries of people who did not experience a stroke whereas there are only about 250 entries of people who experienced one.

The second dataset is the U.S. Accidents dataset (2016-2023). This dataset contains countrywide car accident data. There are approximately 7.7 million observations in this dataset. For each accident, there are a total of 46 features that have been recorded. This project uses this dataset to predict the severity of a car accident using the 22 most clear features to predict an accident. Due to computational capacity constraints, 25,000 observations were randomly sampled from this dataset. The severity of an accident is the target in this project. Severity is measured on a scale of 1 to 4 with 1 indicating the least impact on traffic. This is expected to indicate the length and severity of the accident itself. For this project, accidents with severity levels of 1 or 2 are considered low severity, and those measured as 3 or 4 are considered high severity. There are about 20,000 low severity accidents and about 5,000 high severity accidents.

The last dataset is the Telco Customer Churn dataset that contains 20 features that can be used to predict if a customer discontinues services from Telco. It contains about 7,000 rows of data. There are about 5,000 entries of customers who were retained by the company, and about 2,000 entries of customers that were lost.

2.2. Preprocessing

Preprocessing the datasets involves improving the quality of the data in order to achieve better model performance and optimize computational cost. For this project, the five main steps taken to preprocess the data are:

1. Splitting data into training and testing data.
   The datasets are split into training and test sets using the train_test_split() function from scikit learn. The proportion of test data is varied in each partition. We perform three train-test splits for each dataset: 80/20, 50/50, and 20/80.

2. Imputing missing numerical values.

Missing numerical entries were imputed using the median of the corresponding feature. The median was determined to be the optimal choice for imputing missing values. However, other metrics such as mean and mode can easily be applied by changing the parameters for a general imputation function defined in the code. The function used to achieve this is pd.fillna from the pandas library.

3. Imputing missing categorical values.
   Missing categorical entries were filled by adding a new "missing" category to represent them. Rows with missing values were not deleted in order to avoid unnecessary loss of data, especially because missing entries would affect only a few of the features.

4. Encoding categorical features.
   The categorical features of the dataset were encoded using One Hot Encoding. This creates new columns for each category in a given feature. The presence or absence of the category is represented using 1s and 0s respectively. One Hot Encoding is a popular method for preprocessing data and ensures that categorical data is handled effectively by machine learning models. The OneHotEncoder class from scikit learn was used to achieve this.

It is important to note that imputation of missing numerical and categorical entries, encoding of categorical features, and standardizing of numerical features are done after the splitting into training and test data to avoid data leakage.

2.3. Classifiers

Four classifiers were evaluated in this project.

1. Logistic Regression.
   Logistic Regression is a simple and interpretable linear classifier. During grid search, the important hyperparameters examined were the regularization strength (C) and the penalty type, including L1, L2, or elastic-net.

2. Support Vector Machine.
   The SVM is a margin based model. The important parameters examined during grid search are the kernel type (rbf, linear), and gamma.

3. Decision Tree.
   Decision Tree is a non-linear model that is able to capture interactions between multiple features. The important hyperparameters that were examined were maximum depth, minimum samples per split, and criterion (gini or entropy).

4. Random Forest.
   The Random Forest model is an ensemble of decision trees and is known for its strong performance in prediction tasks. The important hyperparameters that were examined during grid search were number of trees, max depth, and max features.

2.4. Grid Search and Model Training

Grid search and cross-validation were applied to all four classifiers. Grid search systematically tested different combinations of hyperparameters for each model. A 5-fold cross-validation was performed on the training data to obtain robust performance estimates, ensuring that the results were not overly dependent on a single train-test split. The best-performing hyperparameters were selected, and the mean evaluation metrics across all folds were recorded, providing a reliable estimate of model performance. These average metrics were then used to compare the classifiers.

2.5. Evaluation Metrics

The classifiers were evaluated using average accuracy, precision, recall, and F1-score across the cross-validation folds. These metrics were recorded for each classifier and for each partition of a given dataset, allowing for a consistent comparison of the models across different data partitions. Since accuracy can be misleading for imbalanced datasets, precision, recall, and F1-score were included to provide a more comprehensive assessment of classifier performance.

3. Experiments

3.1. Experimental Setup

This project was performed using Python 3.13.7 and utilized classes and functions from many python libraries such as scikit learn, pandas, and numpy. For each of the three datasets, we create three different partitions of training and testing data. The train to test ratios we include are 80/20, 50/50, and 20/80. We run all four classifiers on the three different partitions, one dataset after another.

3.2. Limitations on Experimental Setup

The main limitation of the experimental setup is the size of datasets being used. Since this project is being run on local computers, training for very large datasets can take several hours, and could also be taxing on the hardware. The largest dataset being used is of size (2500, 24), and the SVM and random forest models take about 140 minutes and 60 minutes respectively to complete training for this dataset. Quality of the data, and the topic of the dataset was prioritized while choosing medium sized datasets.

4. Results

4.1. Stroke Prediction Dataset

The results of the classifiers on the three partitions of the stroke prediction dataset are displayed in the tables below.

4.1.1. 80/20 Split

Model	Accuracy	F1	Precision	Recall
Logistic Regression	0.735816	0.124451	0.790896	0.215040
Support Vector Machine	0.731657	0.124613	0.806970	0.215876
Decision Tree	0.821179	0.127702	0.498720	0.202907
Random Forest	0.852498	0.141778	0.444523	0.214790

Table 1. Classifier performance on 80/20 split of Stroke Prediction Dataset.

For the 80/20 split, the random forest classifier performed best overall with highest scores for accuracy, F-1 score, and recall. The decision tree model was very slightly better in precision. The second best classifier overall was the decision tree, followed by the SVM and logistic regression classifiers.

4.1.2. 50/50 Split

Model	Accuracy	F1	Precision	Recall
Logistic Regression	0.738160	0.126183	0.775000	0.216989
Support Vector Machine	0.728376	0.126568	0.808333	0.218802
Decision Tree	0.864188	0.113503	0.275000	0.160432
Random Forest	0.856751	0.163746	0.491667	0.245381

Table 2. Classifier performance on 50/50 split of Stroke Prediction Dataset.

For the 50/50 split on the stroke prediction dataset, random forest was again the best performing classifier. Its performance was very slightly worse than its performance on the 80/20 split. The decision tree, SVM, and logistic regression classifiers were much closer in their performances for this split. It is interesting to note that the decision tree had much worse precision, recall, and F-1 score compared to the previous split, and was outperformed by the SVM classifier in these metrics.

4.1.3. 20/80 Split

Model	Accuracy	F1	Precision	Recall
Logistic Regression	0.743572	0.117050	0.766667	0.202482
Support Vector Machine	0.844433	0.157509	0.625000	0.251221
Decision Tree	0.876738	0.119386	0.300000	0.169890
Random Forest	0.894328	0.183838	0.458333	0.261880

Table 3. Classifier performance on 20/80 split of Stroke Prediction Dataset.

The performance of the classifiers for the 20/80 split goes against our expectation because the performance increases slightly compared to the previous partitions.
This is an important result that can be directly related to the imbalance in the dataset with a disproportionately large number of people not experiencing a stroke.

4.2. Classifier Performance vs Training Size for Stroke Prediction Dataset.

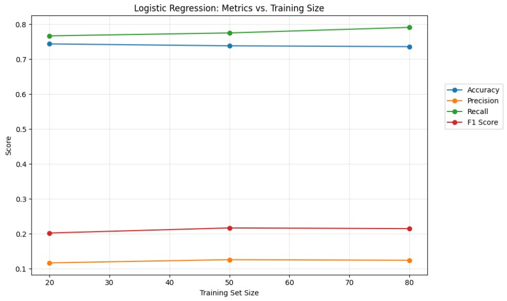

Figure 1. Logistic Regression Performance vs Training size for Stroke Prediction Dataset.

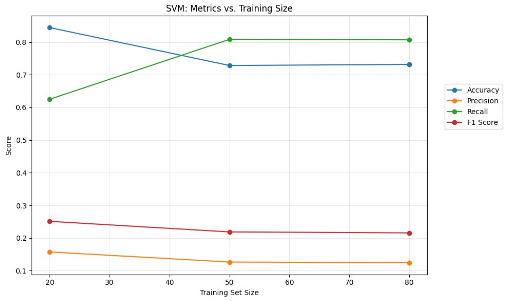

Figure 2. SVM Performance vs Training size for Stroke Prediction Dataset.

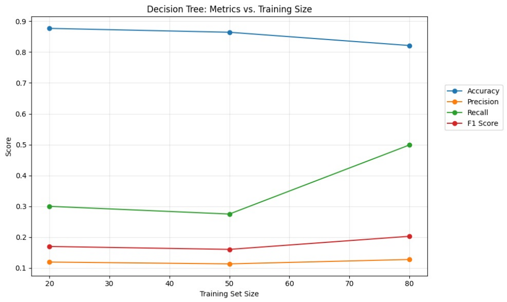

Figure 3. Decision Tree Performance vs Training size for Stroke Prediction Dataset.

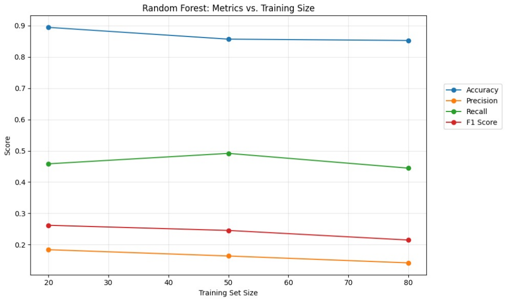

Figure 4. Random Forest Performance vs Training size for Stroke Prediction Dataset.

4.3. US Accidents Dataset

The results of the classifiers on the three partitions of the accident dataset are displayed in the tables below.

4.3.1. 80/20 Split

Model	Accuracy	F1	Precision	Recall
Logistic Regression	0.63040	0.297933	0.652377	0.409042
Support Vector Machine	0.68170	0.338087	0.651357	0.445116
Decision Tree	0.70785	0.374826	0.733480	0.496066
Random Forest	0.72495	0.391783	0.728633	0.509486

Table 4. Classifier performance on 80/20 split of Accident Severity Prediction Dataset.

For the 80/20 split of the US Accident dataset, random forest performs the best compared to other classifiers. But the performance of the classifiers are quite similar to each other for most of the evaluation metrics. Decision tree is the second best classifier for this split.

4.3.2. 50/50 Split

Model	Accuracy	F1	Precision	Recall
Logistic Regression	0.63208	0.297318	0.649170	0.407803
Support Vector Machine	0.67400	0.325366	0.624550	0.427791
Decision Tree	0.69080	0.357669	0.734527	0.480737
Random Forest	0.72536	0.389945	0.723851	0.506819

Table 5. Classifier performance on 50/50 split of Accident Severity Prediction Dataset.

For the 50/50 split on the US Accident dataset, we see very similar performances to the previous split. Overall, majority of the models perform very slightly worse than in the previous split.

4.3.3. 20/80 Split

Model	Accuracy	F1	Precision	Recall
Logistic Regression	0.6436	0.297413	0.639735	0.405912
Support Vector Machine	0.6602	0.310313	0.638672	0.417463
Decision Tree	0.7022	0.356987	0.697548	0.471555
Random Forest	0.7290	0.384588	0.704888	0.497234

Table 6. Classifier performance on 20/80 split of Accident Severity Prediction Dataset.

For the 20/80 split, on a higher level, majority of the models perform very slightly worse on majority of the evaluation metrics compared to the previous splits.
Since this dataset is much less imbalanced than Stroke Prediction dataset, model performance seems to be more aligned with our expectation that lesser training data is associated with lower performance.

4.4. Classifier Performance vs Training Size for US Accidents Dataset.

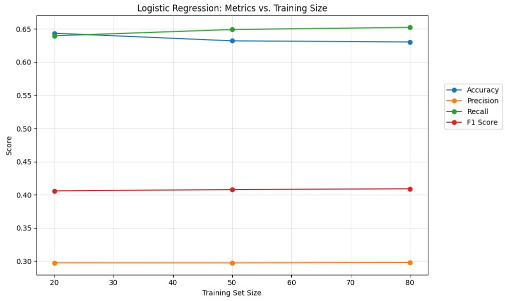

Figure 5. Logistic Regression Performance vs Training size for US Accidents Dataset.

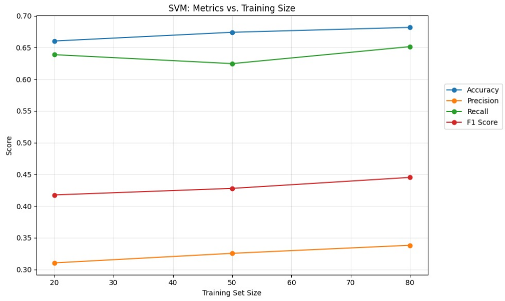

Figure 6. SVM Performance vs Training size for US Accidents Dataset.

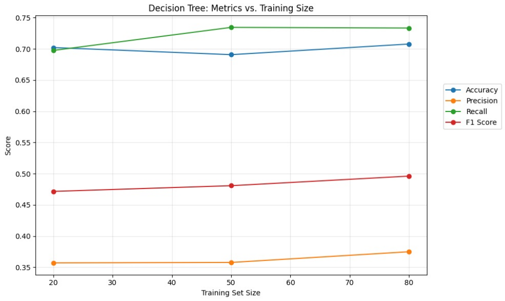

Figure 7. Decision Tree Performance vs Training size for US Accidents Dataset.

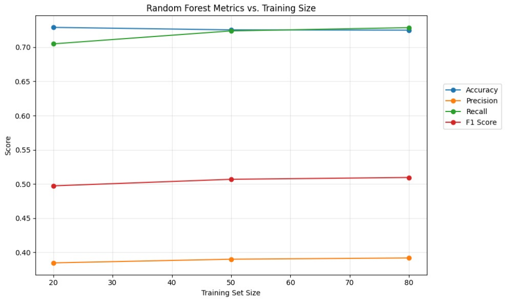

Figure 8. Random Forest Performance vs Training size for US Accidents Dataset.

4.5. Telco Churn Dataset

The results of the classifiers on the three partitions of the Telco Churn prediction dataset are displayed in the tables below.

4.5.1. 80/20 Split

Model	Accuracy	F1	Precision	Recall
Logistic Regression	0.741396	0.508422	0.797463	0.620928
Support Vector Machine	0.746898	0.515766	0.780087	0.620840
Decision Tree	0.731278	0.496821	0.785400	0.608171
Random Forest	0.760387	0.534497	0.758698	0.627148

Table 7. Classifier performance on 80/20 split of Telco Churn Prediction Dataset.

4.5.2. 50/50 Split

Model	Accuracy	F1	Precision	Recall
Logistic Regression	0.730472	0.491697	0.787632	0.605213
Support Vector Machine	0.731324	0.492683	0.768120	0.599823
Decision Tree	0.689862	0.449334	0.778960	0.569100
Random Forest	0.755466	0.523700	0.750787	0.616813

Table 8. Classifier performance on 50/50 split of Telco Churn Prediction Dataset.

4.6. 20/80 Split

Model	Accuracy	F1	Precision	Recall
Logistic Regression	0.741219	0.508217	0.797463	0.620777
Support Vector Machine	0.746898	0.515766	0.780087	0.620840
Decision Tree	0.731278	0.496821	0.785400	0.608171
Random Forest	0.766422	0.544040	0.741982	0.627762

Table 9. Classifier performance on 20/80 split of Telco Churn Prediction Dataset.

4.7. Classifier Performance vs Training Size for Telco Customer Churn Dataset.

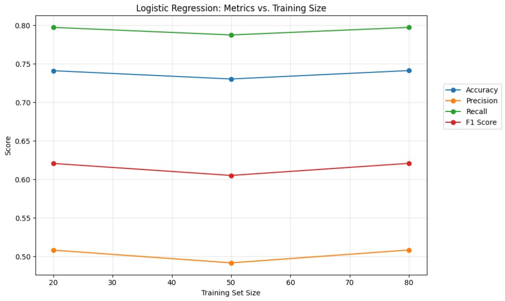

Figure 9. Logistic Regression Performance vs Training size for Telco Customer Churn Dataset.

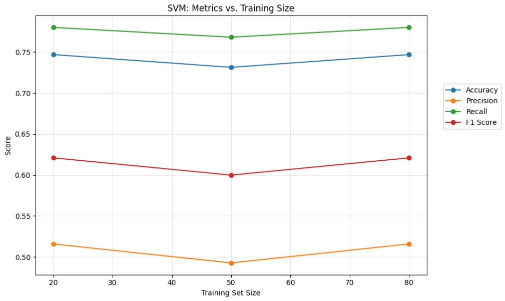

Figure 10. Support Vector Machine Performance vs Training size for Telco Customer Churn Dataset.

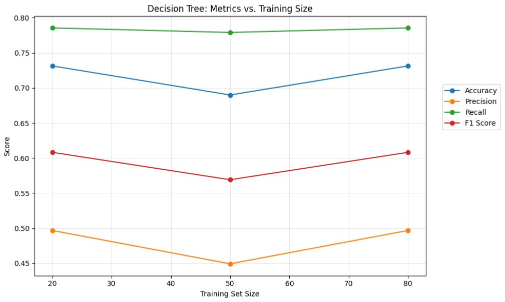

Figure 11. Decision Tree Performance vs Training size for Telco Customer Churn Dataset.

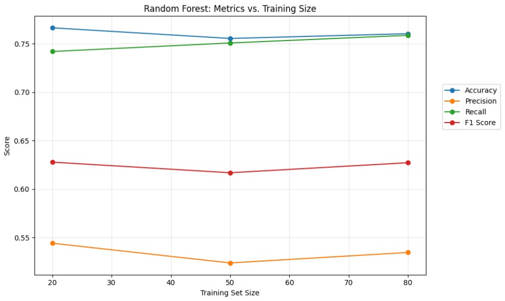

Figure 12. Random Forest Performance vs Training size for Telco Customer Churn Dataset.

4.8. Observations Across Datasets

Across the three datasets, we see that the models perform differently and sometimes showed overfitting as well. The nature of the datasets themselves could also be contributing to the differences in model performance across the datasets. It is also interesting to notice that while some of the metrics improve for all classifiers as the training size increases, some other metrics decrease. A much deeper exploration of these results could provide more insight on why these changes are observed.

5. Conclusion

Random Forest was the best performing classifier across all datasets and partitions. Decision Tree was second best overall, followed by the SVM and logistic regression classifier. For more balanced datasets such as the US Accidents dataset and Telco Churn dataset, a smaller training sample was associated with lower scores for majority of the metrics observed. However, for the Stroke Prediction dataset, which was a highly imbalanced dataset, the effect of training size was less apparent, and went against the expectation that smaller training samples would lead to diminished performance. For the imbalanced dataset too, Random Forest and Decision Tree remained to be the stronger classifiers.

The main limitations of this project are the size of datasets, and the extreme imbalance in one of the datasets, although it added information about how classifiers perform differently for such a dataset.

This project can be extended in several ways including increasing the size of the datasets, and including more complex datasets with a larger number of features. Further, deep learning models can be included which may perform significantly better on capturing non-linear relationships in datasets. Also, this project compared models on only 4 metrics – accuracy, precision, recall, and F-1 score. Evaluating classifiers on many more evaluation metrics could offer more insight into the performance of classifiers in real-world settings.

6. References

Caruana, R., Niculescu-Mizil, A. (2006, June). An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning (pp. 161-168).

GeeksforGeeks. (2025). Comprehensive guide to classification models in scikit-learn. https://www.geeksforgeeks.org/machine-learning/comprehensive-guide-to-classification-models-in-scikit-learn/

GeeksforGeeks. (2025). Implementing decision tree classifiers with scikit-learn. https://www.geeksforgeeks.org/building-and-implementing-decision-tree-classifiers-with-scikit-learn-a-comprehensive-guide/